SIEMENS

SINUMERIK

SINUMERIK 828D Werkzeugverwaltung

Funktionshandbuch

Einleitung 1

Grundlegende Sicherheitshinweise 2

Grundlagen 3

Maschinendaten für die Werkzeugverwaltung 4

NC-PLC-Schnittstelle der Werkzeugverwaltung 5

PLC Programmbausteine 6

Beispiel: Beladen / Entladen 7

Beispiel: Handwerkzeuge wechseln 8

Anwender-Beispiel für Drehmaschine 9

Anwender-Beispiel für Fräsmaschine 10

NC-Programmierung 11

SINUMERIK Operate Bedienoberfläche 12

Anbindung Codeträger - Tool Ident Connection (Option) 13

Anhang A

Gültig für

Steuerung
SINUMERIK 828

CNC-Software Version 4.95

07/2021
6FC5397-7JP40-0AA0

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

⚠ **GEFAHR**
bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠ **WARNUNG**
bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠ **VORSICHT**
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

⚠ **WARNUNG**
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

Dokumentbestellnummer: 6FC5397-7JP40-0AA0
Ⓟ 07/2021 Änderungen vorbehalten

Copyright © Siemens AG 2019 - 2021.
Alle Rechte vorbehalten

Inhaltsverzeichnis

1 Einleitung 9

1.1 Über SINUMERIK 9

1.2 Über diese Dokumentation 9

1.3 Dokumentation im Internet 10

1.3.1 Dokumentationsübersicht SINUMERIK-Bedienkomponenten 10

1.4 Feedback zur technischen Dokumentation 10

1.5 mySupport-Dokumentation 10

1.6 Service und Support 11

1.7 Wichtige Produktinformationen 13

2 Grundlegende Sicherheitshinweise 15

2.1 Allgemeine Sicherheitshinweise 15

2.2 Gewährleistung und Haftung für Applikationsbeispiele 15

2.3 Security-Hinweise 15

3 Grundlagen 17

3.1 Werkzeugverwaltung - Übersicht 17

3.2 Struktur der Werkzeugverwaltung 18

3.3 Komponenten der Werkzeugverwaltung 20

3.4 Multitool 23

3.5 Handwerkzeuge 24

4 Maschinendaten für die Werkzeugverwaltung 27

5 NC-PLC-Schnittstelle der Werkzeugverwaltung 31

5.1 Werkzeug umsetzen, entladen, beladen, Magazin positionieren 33

5.2 Werkzeugwechsel 40

5.3 Transferschritt- und Quittungsschritt-Tabellen 49

6 PLC Programmbausteine 53

6.1 Quittungsprozess 53

6.2 Quittungsarten 53

6.3 Quittungszustände 55

6.4 Schritt-Tabellen projektieren 59

6.5 Quittungsschritte projektieren 62

6.6 PLC-Programm anpassen 63

6.7 Auskunft zum Magazinplatz ... 65
6.8 PI Dienst: TMMVTL ... 67
7 Beispiel: Beladen / Entladen ... 69
8 Beispiel: Handwerkzeuge wechseln ... 71
9 Anwender-Beispiel für Drehmaschine ... 75
9.1 Beispiel: Drehmaschine mit Revolvermagazin (MAG_CONF_MPF) ... 75
9.2 Beispiel: Quittungsschritte (Drehmaschine) ... 80
9.3 Beispiel: Werkzeugwechselzyklus für Drehmaschine ... 81
9.4 Beispiel: Satzsuchlauf und MCALL ... 83
9.5 Beispiel: Drehmaschine mit Gegenspindel ... 84
9.6 Beispiel: Test auf leere Zwischenspeicher ... 84
9.7 Beispiel: Werkzeug aus einem Zwischenspeicher ins Magazin transportieren ... 85
9.8 Beispiel: Auftrag "Werkzeugwechsel vorbereiten" wiederholen ... 85
10 Anwender-Beispiel für Fräsmaschine ... 87
10.1 Beispiel: Fräsmaschine mit Kettenmagazin und Doppelgreifer (MAG_CONF_MPF) ... 87
10.2 Ablaufdiagramm: Werkzeugwechsel ... 92
10.3 Beispiel: Quittungsschritte (Fräsmaschine) ... 101
10.4 Beispiel: Werkzeugwechselzyklus für Fräsmaschine ... 102
11 NC-Programmierung ... 105
11.1 Übersicht der BTSS und Systemvariablen ... 105
11.2 Verwendung der Magazindaten ... 107
11.3 Verwendung der Werkzeug- und Schneidendaten ... 109
11.4 Schneidendaten ... 112
11.4.1 Schneidendaten ... 112
11.4.2 Schneidenparameter $TC_DP ... 112
11.4.3 Anwender-Schneidendaten $TC_DPC ... 114
11.4.4 Schneidenbezogene Werkzeugüberwachung $TC_MOP ... 115
11.4.5 Anwender-Schneidenüberwachung $TC_MOPC ... 116
11.4.6 Einsatzortabhängige Korrekturen fein (Summenkorrekturen) $TC_SCP ... 117
11.4.7 Ortsabhängige Korrekturen grob (Einrichtekorrekturen) $TC_ECP ... 117
11.5 Werkzeugdaten ... 118
11.5.1 Übersicht ... 118
11.5.2 Werkzeugbezogene Daten $TC_TP ... 118
11.5.3 Werkzeugbezogene Schleifdaten $TC_TPG ... 124
11.5.4 Werkzeugbezogene Anwenderdaten $TC_TPC ... 125
11.6 Magazindaten ... 126
11.6.1 Übersicht Magazindaten ... 126
11.6.2 Magazinbeschreibungsdaten $TC_MAP ... 126
11.6.3 Magazin-Anwenderdaten $TC_MAPC ... 129
11.6.4 Magazinplatzdaten $TC_MPP ... 129

11.6.5 Magazinplatz-Anwenderdaten $TC_MPPC ... 133
11.6.6 Magazinplatztyphierarchie $TC_MPTH ... 134
11.6.7 Abstand zur Wechselstelle $TC_MDP ... 135
11.6.8 Magazinbausteine $TC_MAMP ... 138
11.6.9 Zuordnung von Zwischenspeichern zu Spindeln $TC_MLSR ... 145

11.7 Multitooldaten ... 147
11.7.1 $TC_MTP - Multitooldaten ... 147
11.7.2 $TC_MTPP - Multitool Platzdaten ... 157
11.7.3 Multitoolbezogene Anwenderdaten ... 159
11.7.4 Multitoolplatzbezogene Anwenderdaten ... 159

11.8 Adapterdaten $TC_ADPT ... 160

11.9 Freie Anwendervariablen ... 161

11.10 NC-Sprachbefehle ... 162
11.10.1 CHKDNO - Prüfung der Eindeutigkeit der D-Nummer ... 162
11.10.2 CHKDM - Prüfung der Eindeutigkeit innerhalb eines Magazins ... 163
11.10.3 GETACTTD - Ermittlung der T-Nummer zu einer eindeutigen D-Nummer ... 164
11.10.4 GETDNO - D-Nummer auslesen ... 165
11.10.5 SETDNO - D-Nummer setzen bzw. ändern ... 166
11.10.6 DZERO - D-Nummern ungültig setzen ... 167
11.10.7 DELDL - Additive Korrekturen löschen ... 167
11.10.8 NEWT - Neues Werkzeug anlegen ... 168
11.10.9 NEWMT Neues Multitool anlegen ... 169
11.10.10 DELT Werkzeug löschen ... 170
11.10.11 DELMT - Multitool löschen ... 171
11.10.12 $TC_MTPN - Löschen eines oder aller Multitools ... 172
11.10.13 GETT - T-Nummer lesen ... 172
11.10.14 SETPIECE - Stückzahlzähler dekrementieren ... 174
11.10.15 GETSELT - Lesen der angewählten T-Nummer ... 176
11.10.16 GETEXET - Lesen der eingewechselten T-Nummer ... 179
11.10.17 $P_MTHSDC - Master-Toolholder bezüglich der D-Korrekturanwahl ... 182
11.10.18 $P_TH_OF_D - Master Toolholder bezüglich der aktuellen D-Korrektur ... 184
11.10.19 GETACTT - Lesen der aktiven internen T-Nummmer ... 185
11.10.20 SETMS - Masterspindel setzen ... 186
11.10.21 SETMTH Masterwerkzeughalternummer setzen ... 187
11.10.22 POSM - Magazin positionieren ... 190
11.10.23 POSMT - Multitool auf WZ-Halter auf Platznummer positionieren ... 192
11.10.24 MVTOOL - Sprachbefehl zum Bewegen eines Werkzeugs ... 194
11.10.25 SETTA - Werkzeug aus Verschleißverbund aktiv setzen ... 197
11.10.26 SETTIA - Werkzeug aus Verschleißverbund inaktiv setzen ... 199
11.10.27 RESETMON - Sprachbefehl zur Sollwertaktivierung ... 201
11.10.28 DELTC - Lösche Werkzeug-Trägerdatensatz ... 205
11.10.29 TCA - Werkzeug-Anwahl/Werzeugwechsel unabhängig vom Status des Werkzeugs ... 206
11.10.30 TCI - Wechsle Werkzeug aus Zwischenspeicher in das Magazin ... 209
11.10.31 GETFREELOC - Suche Leerplatz ... 211
11.10.32 DELMLRES - Lösche den Platzzustand "reserviert für WZ im Zwischenspeicher" ... 218
11.10.33 DELMLOWNER - Lösche Eigentümermagazinplatz des Werkzeugs ... 219
11.10.34 $P_USEKT - Werkzeugwechsel nur mit Werkzeugen der Untergruppe ... 221
11.10.35 TOOLGNT/TOOLGT - Werkzeuggruppen ... 224
11.10.36 $P_TMNOIS - ist Nummer T-Nummer, Magazinnummer oder MT-Nummer ... 225
11.10.37 $P_TOOLEXIST - Existenz eines Werkzeugs feststellen ... 226

11.10.38 $A_TOOLMN - Magazin-Nr. vom Werkzeug lesen ... 227
11.10.39 $P_MTOOLN / $P_MTOOLMT- Anzahl Multitools / MT-Nummer ermitteln ... 228
11.10.40 $P_MTOOLNT / $P_MTOOLT - Anzahl der Werkzeuge im Multitool ... 229
11.10.41 $A_TOOLMLN - Magazinplatz-Nr. von Werkzeug lesen ... 231
11.10.42 $P_TOOLND - Schneidenanzahl von Werkzeug lesen ... 231
11.10.43 $A_MONIFACT - Faktor für Standzeitüberwachung ... 232
11.10.44 $AC_MONMIN - Faktor für die Werkzeugsuche ... 233
11.10.45 $P_TOOLNG - Anzahl Werkzeuggruppen ... 236
11.10.46 $A_MYMN / $A_MYMLN - Eigentümermagazin/platz des Werkzeugs ... 237
11.10.47 $A_MYMTN / $A_MYMTLN - $A_TOOLMTN / $A_TOOLMTLN - Werkzeuge im Multitool ... 238
11.10.48 $P_TOOLNT / $P_TOOLT - T-Nummern ... 239
11.10.49 $P_TOOLD - D-Nummern ... 240
11.10.50 $P_TOOLNDL - Anzahl definierter DL-Korrekturen ... 241
11.10.51 $A_USEDND - Stückzahlzählung ... 242
11.10.52 $A_USEDT - Stückzahlzählung ... 243
11.10.53 $A_USEDD - Stückzahlzählung ... 245
11.10.54 $P_MAGN / $P_MAG - Magazine ... 246
11.10.55 $P_MAGNDIS / $P_MAGDISS / $P_MAGDISL - Magazindistanztabellen ... 247
11.10.56 $P_MAGNS / $P_MAGS - Werkzeughalter ... 248
11.10.57 $P_MAGNREL / $P_MAGREL - zugeordnete Zwischenspeicher ... 249
11.10.58 Beispiel zu den Magazinkonfigurations-Systemvariablen ... 250
11.10.59 $P_MAGNH / $P_MAGNHLT / $P_MAGHLT - Platztyphierarchien ... 252
11.10.60 $P_MAGNA / $P_MAGA - Werkzeug-Adapter ... 255
11.10.61 Weitere Sprachbefehle ... 256
11.10.62 Variablen für Unterprogrammersetzungstechnik ... 264
11.10.63 Variablen für WZ-Wechsel in Synchronaktion ... 265

11.11 Festlegungen bei der Programmierung von Daten ... 266
11.11.1 Werkzeugwechsel ... 266
11.11.2 Schneidenanwahl ... 267
11.11.3 Werkzeugübernahme aus Programmtest ... 268

11.12 Programmierung T=Platznummer ... 268

11.13 Mehrere Revolver mit "T=Platznummer" aufrufen ... 270

11.14 Programmierbeispiele ... 271

11.15 Übersicht der übrigen BTSS-Bausteine der WZV ... 271
11.15.1 Magazindaten, Verzeichnis ... 271
11.15.2 Werkzeugdaten, Verzeichnis ... 272
11.15.3 Parametrierung, Rückgabeparameter _N_TMGETT, _N_TSEARC ... 272
11.15.4 Arbeitskorrekturen ... 273
11.15.5 PI-Dienste und Sprachbefehle für WZV ... 274

12 SINUMERIK Operate Bedienoberfläche ... 281

12.1 Bedienoberfläche konfigurieren ... 281
12.1.1 Allgemeine Einstellungen ... 284
12.1.2 Werkzeuglisten konfigurieren ... 290
12.1.2.1 Kennungen der Werkzeugparameter ... 294
12.1.2.2 Kennungen der Schneidenparameter ... 297
12.1.2.3 Kennungen der Überwachungsparameter ... 300
12.1.2.4 Kennungen der Schleifparameter ... 301
12.1.2.5 Kennungen der Magazinplatzparameter ... 301
12.1.2.6 Kennungen der Magazinplatzadapterparameter ... 302

12.1.2.7 Kennungen der Multitoolparameter ... 305
12.1.2.8 Kennungen der Multitoolplatzparameter ... 306
12.1.2.9 Kennungen der Winkelkopfparameter ... 307
12.1.2.10 Kennungen der Winkelkopfplatzparameter ... 309
12.1.3 Listenparameter konfigurieren ... 309
12.1.4 Liste der Werkzeugtypen ... 313
12.1.5 Werkzeugtypen konfigurieren ... 315
12.1.6 Fenster "Weitere Daten" konfigurieren ... 318
12.1.7 Fenster "Neues Werkzeug - Favoriten" konfigurieren ... 320
12.1.8 Fenster "Neues Werkzeug" konfigurieren ... 320
12.1.9 Fenster "Neues Multitool" konfigurieren ... 321
12.1.10 Fenster "Neuer Winkelkopf" konfigurieren ... 322
12.1.11 Fenster "Neues Werkzeug von Codeträger" konfigurieren ... 322
12.1.12 Fenster "Neues Werkzeug aus Datei" konfigurieren ... 322
12.1.13 Fenster "Neues Werkzeug aus TTD-Datei" konfigurieren ... 323
12.1.14 Standardwerte für neue Werkzeuge konfigurieren ... 323
12.1.15 Fenster "Details" konfigurieren ... 328
12.1.16 Namen für Magazine und Magazinplätze vergeben ... 330
12.1.17 Fenster "Tooltips für Multitoolplätze vergeben" konfigurieren ... 332
12.1.18 Namen für Magazinplatztypen vergeben ... 332
12.1.19 Zuordnung von Magazinen und Magazinplätzen zu Kanälen ... 334
12.1.20 Zuordnung von Magazinen und Magazinplätzen zu Koordinatensystemen ... 336
12.1.21 Sperren des Softkeys in Abhängigkeit von Filtern ... 337
12.1.22 Fenster "Auswahl der Beladestelle" konfigurieren ... 338
12.1.23 Kühlmittel und werkzeugspezifische Funktionen ... 339
12.1.24 Werkzeugwechselgrund beim Reaktivieren ... 342
12.1.25 Codeträgeranbindung konfigurieren ... 346
12.1.26 Speichern in Datei konfigurieren ... 349
12.1.27 Anlegen aus TTD-Datei konfigurieren ... 351

12.2 OEM-Texte anlegen ... 352
12.2.1 Kennungen der Standardtexte ... 353
12.2.2 Beispiele für OEM-Texte ... 355

12.3 Beispiele ... 358
12.3.1 Beispiel: OEM-Werkzeugliste konfigurieren ... 358
12.3.1.1 Anpassung der Konfigurationsdatei ... 358
12.3.1.2 Anpassung der Kundentextdatei ... 360
12.3.2 Beispiel: Magazinplatztypen mit Namen konfigurieren ... 362
12.3.2.1 Anpassung der Konfigurationsdatei ... 362
12.3.2.2 Anpassung der Kundentextdatei ... 363

12.4 Arbeiten mit zwei Werkzeugträgern ... 366

13 Anbindung Codeträger - Tool Ident Connection (Option) ... 369

13.1 Installation ... 369
13.1.1 Systemvoraussetzungen ... 369
13.1.2 Installation von Tool Ident Connection ... 370
13.1.2.1 Aktivieren von Tool Ident Connection ... 370
13.1.2.2 Installation Wkonvert-Wizard ... 371

13.2 Konfiguration ... 376
13.2.1 Konfiguration der Werkzeuglisten ... 376
13.2.2 TDIIdentCfg.xml ... 376

13.2.3 Schreib-/Lesegerät an der PLC ... 379
13.2.4 Datenkonvertierung ... 379
13.2.4.1 Übersicht ... 379
13.2.4.2 Anwenden des Wkonvert-Wizard ... 381
13.2.4.3 Meldungen des Wkonvert-Wizard ... 382
13.2.5 Konvertierungsdateien wkonvert.txt und DefToolDat.txt ... 383
13.2.5.1 Aufbau der Beschreibungsdatei wkonvert.txt ... 383
13.2.5.2 Beispiele für Beschreibungsdateien wkonvert.txt ... 388
13.2.5.3 Werkzeugdaten-Initialisierungsdatei DefToolDat.txt ... 391
13.2.6 Konvertierungsvorschrift auf Basis der MCIS-C Skriptsprache ... 393
13.2.6.1 Aufbau der Konvertierungsvorschrift ... 393
13.2.6.2 Zugriff auf Werkzeugdaten ... 395
13.2.6.3 Hilfsfunktionen für Zahlencodierungen ... 397
13.2.6.4 Konfiguration von ToolSpec.xml ... 402
13.2.6.5 Fehlerbehandlung ... 403
13.2.6.6 Hilfsfunktionen in Kovertierungsvorschriften ... 404
13.2.7 Beispiele ... 404

13.3 Kommando-Codes ... 408
13.3.1 Funktionsumfang der PLC-Schnittstelle (K-Codes) ... 408
13.3.2 Parameterbeschreibung ... 409
13.3.3 Beladen vom Codeträger (K-Code = 0) ... 409
13.3.4 Entladen auf den Codeträger (K-Code = 1) ... 410
13.3.5 Werkzeug beladen aus der Werkzeugliste (K-Code = 7) ... 412
13.3.6 Werkzeug entladen in die Werkzeugliste (K-Code = 8) ... 413
13.3.7 Leerplatzsuche (K-Code = 9) ... 414
13.3.8 Werkzeug im NC löschen (K-Code = 10) ... 415
13.3.9 Codeträger aktualisieren (K-Code = 14) ... 416
13.3.10 Lesen der Daten vom Codeträger (K-Code = 15) ... 417
13.3.11 Fehlercodes ... 418

13.4 Beispiele ... 423
13.4.1 Beladen vom Codeträger über PLC-Schnittstelle ... 424
13.4.2 Entladen auf den Codeträger über PLC- Schnittstelle ... 427

A Anhang ... 429

A.1 Liste der Abkürzungen ... 429

Index ... 431

Einleitung 1

1.1 Über SINUMERIK

Von einfachen CNC-Standardmaschinen über standardisierte Maschinen, bis hin zu modularen Premium-Maschinenkonzepten – die CNC-Steuerungen SINUMERIK bieten für jedes Maschinenkonzept die passende Lösung. Ob Einzelteil- oder Massenfertigung, einfache oder komplexe Werkstücke – SINUMERIK ist die hochproduktive Automatisierungslösung durchgängig für alle Fertigungsbereiche – vom Muster- und Werkzeugbau über den Formenbau bis zur Großserienfertigung.

Für weitere Informationen besuchen Sie die Internetseite zu SINUMERIK (https://www.siemens.de/sinumerik).

1.2 Über diese Dokumentation

Standardumfang

In der vorliegenden Dokumentation ist die Funktionalität des Standardumfangs beschrieben. Dieser kann vom Umfang der Funktionalitäten des gelieferten Systems abweichen. Die Funktionalitäten des gelieferten Systems entnehmen Sie ausschließlich den Bestellunterlagen.

Im System können weitere, in dieser Dokumentation nicht erläuterte Funktionen ablauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei der Neulieferung bzw. im Servicefall.

Diese Dokumentation kann aus Gründen der Übersichtlichkeit nicht sämtliche Detailinformationen zu allen Typen des Produkts enthalten. Ferner kann diese Dokumentation nicht jeden möglichen Fall der Aufstellung, des Betriebs und der Instandhaltung berücksichtigen

Durch den Maschinenhersteller vorgenommene Ergänzungen oder Änderungen am Produkt dokumentiert der Maschinenhersteller.

Webseiten Dritter

Dieses Dokument kann Hyperlinks auf Webseiten Dritter enthalten. Siemens übernimmt für die Inhalte dieser Webseiten weder eine Verantwortung noch macht Siemens sich diese Webseiten und ihre Inhalte zu eigen. Siemens kontrolliert nicht die Informationen auf diesen Webseiten und ist auch nicht für die dort bereitgehaltenen Inhalte und Informationen verantwortlich. Das Risiko für deren Nutzung trägt der Nutzer.

1.3 Dokumentation im Internet

1.3.1 Dokumentationsübersicht SINUMERIK-Bedienkomponenten

Eine umfangreiche Dokumentation zu SINUMERIK-Bedienkomponenten finden Sie unter Dokumentationsübersicht SINUMERIK-Bedienkomponenten (https://support.industry.siemens.com/cs/ww/de/view/109783841).

Sie haben die Möglichkeit, die Dokumente anzuzeigen oder im PDF- und HTML5-Format herunterzuladen.

Die Dokumentation ist in folgende Kategorien unterteilt:

- Bedientafeln
- Maschinensteuertafeln
- Machine Pushbutton Panel
- Handheld Unit/Mini-Handgeräte
- Weitere Bedienkomponenten

Einen Überblick über die wichtigsten Dokumente, Beiträge und Links zum Thema "SINUMERIK" finden Sie unter SINUMERIK Übersicht-Themenseite (https://support.industry.siemens.com/cs/document/109766201/sinumerik-ein-%C3%BCberblick-der-wichtigsten-dokumente-und-links?lc=de-ww).

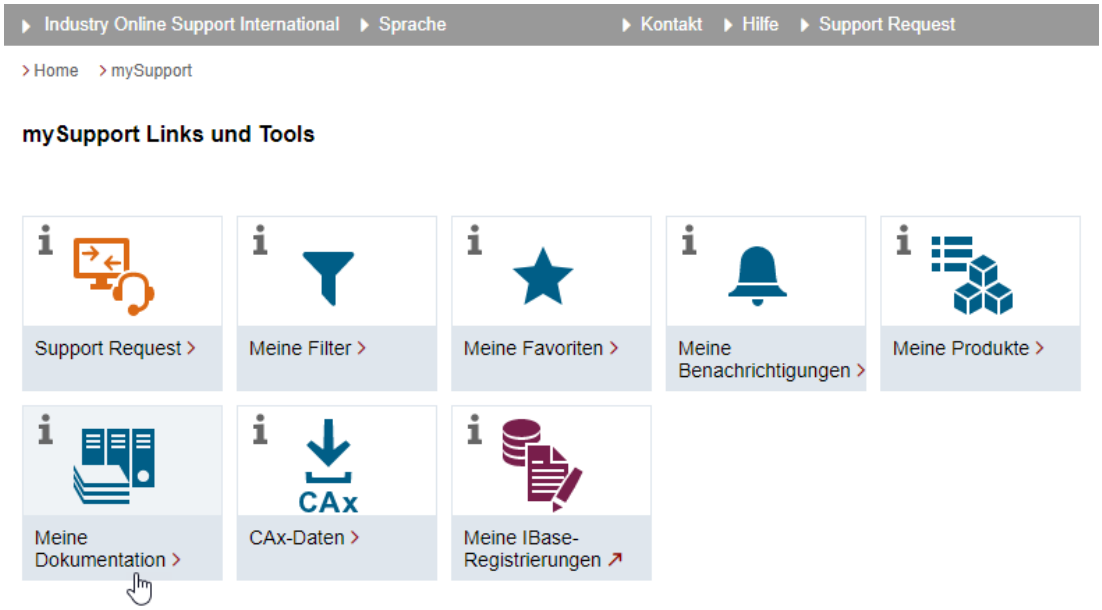
1.4 Feedback zur technischen Dokumentation

Bei Fragen, Anregungen oder Korrekturen zu der im Siemens Industry Online Support veröffentlichten technischen Dokumentation nutzen Sie den Link "Feedback senden" am Ende eines Beitrags.

1.5 mySupport-Dokumentation

Mit dem webbasierten System "mySupport-Dokumentation" können Sie Ihre Dokumentation auf Basis der Siemens-Inhalte individuell zusammenstellen und für die eigene Maschinendokumentation anpassen.

Sie starten die Anwendung über die Kachel "Meine Dokumentation" auf der mySupport-Startseite (https://support.industry.siemens.com/cs/ww/de/my):



Der Export des konfigurierten Handbuchs ist im RTF-, PDF- oder XML-Format möglich.

Hinweis

Siemens-Inhalte, die die Anwendung mySupport-Dokumentation unterstützen, erkennen Sie am Vorhandensein des Links "Konfigurieren".

1.6 Service und Support

Product Support

Weitere Informationen zum Produkt finden Sie im Internet:

Product support (https://support.industry.siemens.com/cs/ww/de/)

Unter dieser Adresse finden Sie Folgendes:

- Aktuelle Produkt-Informationen (Produktmitteilungen)
- FAQ (häufig gestellte Fragen)
- Handbücher
- Downloads
- Newsletter mit den neuesten Informationen zu Ihren Produkten
- Forum zum weltweiten Informations- und Erfahrungsaustausch für Anwender und Spezialisten
- Ansprechpartner vor Ort über unsere Ansprechpartner-Datenbank (→ "Kontakt")
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile und vieles mehr (→ "Services")

Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter der Adresse (https://support.industry.siemens.com/cs/ww/de/sc/4868) im Bereich "Kontakt".

Um eine technische Frage zu stellen, nutzen Sie das Online-Formular im Bereich "Support Request".

Training

Unter folgender Adresse (https://www.siemens.de/sitrain) finden Sie Informationen zu SITRAIN. SITRAIN bietet Trainingsangebote für Siemens-Produkte, Systeme und Lösungen der Antriebs- und Automatisierungstechnik.

Siemens-Support für unterwegs



Mit der preisgekrönten App "Siemens Industry Online Support" haben Sie jederzeit und überall Zugang zu über 300.000 Dokumenten der Siemens Industry-Produkte. Die App unterstützt Sie unter anderem in folgenden Einsatzfeldern:

- Lösen von Problemen bei einer Projektumsetzung
- Fehlerbehebung bei Störungen
- Erweiterung oder Neuplanung einer Anlage

Außerdem haben Sie Zugang zum Technical Forum und weiteren Beiträgen, die von unseren Experten für Sie erstellt werden:

- FAQs
- Anwendungsbeispiele
- Handbücher
- Zertifikate
- Produktmitteilungen und viele andere

Die App "Siemens Industry Online Support" ist für Apple iOS und Android verfügbar.

Data-Matrix-Code auf dem Typenschild

Der Data-Matrix-Code auf dem Typenschild beinhaltet die spezifischen Daten des Geräts. Dieser Code kann mit jedem Smartphone eingelesen werden, über die Mobile App "Industry Online Support" können damit technische Informationen zum entsprechenden Gerät angezeigt werden.

1.7 Wichtige Produktinformationen

Verwendung von OpenSSL

Dieses Produkt kann folgende Software enthalten:

- Software, die durch das OpenSSL-Projekt für die Nutzung innerhalb des OpenSSL-Toolkits entwickelt wurde
- Von Eric Young erstellte kryptografische Software
- Von Eric Young entwickelte Software

Weitere Informationen finden Sie im Internet:

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

Einhaltung der Datenschutz-Grundverordnung

Siemens beachtet die Grundsätze des Datenschutzes, insbesondere die Gebote der Datenminimierung (privacy by design).

Für dieses Produkt bedeutet das:

Das Produkt verarbeitet oder speichert keine personenbezogenen Daten, lediglich technische Funktionsdaten (z. B. Zeitstempel). Verknüpft der Anwender diese Daten mit anderen Daten (z. B. Schichtplänen) oder speichert er personenbezogene Daten auf dem gleichen Medium (z. B. Festplatte) und stellt so einen Personenbezug her, hat er die Einhaltung der datenschutzrechtlichen Vorgaben selbst sicherzustellen.

Grundlegende Sicherheitshinweise 2

2.1 Allgemeine Sicherheitshinweise

⚠ WARNUNG

Lebensgefahr bei Nichtbeachtung von Sicherheitshinweisen und Restrisiken

Bei Nichtbeachtung der Sicherheitshinweise und Restrisiken in der zugehörigen Hardware-Dokumentation können Unfälle mit schweren Verletzungen oder Tod auftreten.

- Halten Sie die Sicherheitshinweise der Hardware-Dokumentation ein.
- Berücksichtigen Sie bei der Risikobeurteilung die Restrisiken.

⚠ WARNUNG

Fehlfunktionen der Maschine infolge fehlerhafter oder veränderter Parametrierung

Durch fehlerhafte oder veränderte Parametrierung können Fehlfunktionen an Maschinen auftreten, die zu Körperverletzungen oder Tod führen können.

- Schützen Sie die Parametrierung vor unbefugtem Zugriff.
- Beherrschen Sie mögliche Fehlfunktionen durch geeignete Maßnahmen, z. B. NOT-HALT oder NOT-AUS.

2.2 Gewährleistung und Haftung für Applikationsbeispiele

Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen.

Als Anwender sind Sie für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung.

2.3 Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z.B. Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter:
https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity)

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter:
https://www.siemens.com/industrialsecurity (https://new.siemens.com/global/en/products/services/cert.html#Subscriptions)

Weitere Informationen finden Sie im Internet:

Projektierungshandbuch Industrial Security (https://support.industry.siemens.com/cs/ww/de/view/108862708)

⚠ WARNUNG

Unsichere Betriebszustände durch Manipulation der Software

Manipulationen der Software, z. B. Viren, Trojaner oder Würmer, können unsichere Betriebszustände in Ihrer Anlage verursachen, die zu Tod, schwerer Körperverletzung und zu Sachschäden führen können.

- Halten Sie die Software aktuell.
- Integrieren Sie die Automatisierungs- und Antriebskomponenten in ein ganzheitliches Industrial Security-Konzept der Anlage oder Maschine nach dem aktuellen Stand der Technik.
- Berücksichtigen Sie bei Ihrem ganzheitlichen Industrial Security-Konzept alle eingesetzten Produkte.
- Schützen Sie die Dateien in Wechselspeichermedien vor Schadsoftware durch entsprechende Schutzmaßnahmen, z. B. Virenscanner.
- Prüfen Sie beim Abschluss der Inbetriebnahme alle security-relevanten Einstellungen.

Grundlagen 3

3.1 Werkzeugverwaltung - Übersicht

Werkzeugverwaltung (WZV)

Die Funktion Werkzeugverwaltung (WZV) stellt sicher, dass an der Maschine zu jeder Zeit das richtige Werkzeug am richtigen Platz ist.

Maschine, Magazine, Beladestellen und Werkzeugzwischenspeicher (z. B. Spindeln, Greifer) bilden ein konkretes Anlagensystem, in welchem die Werkzeuge aufbewahrt und transportiert werden. Die Werkzeugverwaltung informiert die NC stets über den aktuellen Standort der Werkzeuge und protokolliert die durch Teileprogramm, PLC oder SINUMERIK Operate ausgelösten Werkzeugbewegungen.

Bei Inbetriebnahme der Werkzeugverwaltung wird die Architektur des konkreten Anlagensystems in der Steuerung abgebildet. Der "Aufenthaltsort" eines Werkzeugs wird in der Steuerung in der Form von Wertepaaren (Magazinnummer und Platznummer) beschrieben.

Hinweis

Lieferumfang

Die Werkzeugverwaltung ist im Lieferumfang für alle Steuerungen enthalten.

- Die Funktion "Ersatzwerkzeuge für Werkzeugverwaltung" (6FC5800-0AM78-0YB0) ist eine lizenzpflichtige Option.
- Die Funktion "Überwachung auf max. WZ-Drehzahl/Beschleunigung" (6FC5800-0AS08-0YB0) ist eine lizenzpflichtige Option.

Ersatzwerkzeuge

Wird die Option "Ersatzwerkzeuge für Werkzeugverwaltung" verwendet, kann man Ersatzwerkzeuge erzeugen und benutzen.

Werkzeuge mit dem gleichen Werkzeugnamen bilden eine Werkzeuggruppe und unterscheiden sich in der Duplo-Nummer. Die maximale Anzahl von Ersatzwerkzeugen einer Werkzeuggruppe kann über das MD17500 $MN_MAXNUM_REPLACEMENT_TOOLS eingestellt werden. Im Teileprogramm wird der Name der Werkzeuggruppe programmiert. Enthält die Werkzeuggruppe mehrere Werkzeuge, kommt in der Regel das Werkzeug mit dem Status "aktiv" zum Einsatz. Wird dieses gesperrt, so kommt eines der anderen Werkzeuge aus der gleichen Werkzeuggruppe zum Einsatz.

Weitere Informationen

Eine umfassende Beschreibung der Werkzeug- und Magazinparameter sowie der Datenstruktur finden Sie in:

- SINUMERIK 828D Listenhandbuch Maschinendaten (https://support.industry.siemens.com/cs/document/109766724)
- SINUMERIK 828D Listenhandbuch NC-Variablen und Nahtstellensignale (https://support.industry.siemens.com/cs/document/109766724)
- Die Bedienoberfläche der Werkzeugverwaltung ist beschrieben in: Inbetriebnahmehandbuch IM9 SINUMERIK Operate (https://support.industry.siemens.com/cs/document/109766724)

Siehe auch

SIOS_840Dsl_FH_WZV (https://support.industry.siemens.com/cs/document/109763234)

3.2 Struktur der Werkzeugverwaltung

Funktionsstruktur

Die Software-Komponenten der Steuerung haben bei der Werkzeugverwaltung folgende Aufgaben:

- **NC:**
  Die Werkzeugverwaltung verwaltet die Plätze der Magazine. Diese Plätze können frei, mit Werkzeugen bestückt oder durch übergroße Werkzeuge auf Nachbarplätzen belegt sein. Freie Plätze können mit weiteren Werkzeugen beladen werden. Über die Werkzeugverwaltung ist für den Maschinenhersteller eine optimale Verwaltung der Werkzeuge und Magazinplätze gegeben. Mit der Magazinverwaltung stehen erweiterte Funktionen wie Beladen, Entladen, Positionieren von Werkzeugen zur Verfügung. Außerdem gibt es Suchfunktionen von Werkzeugen, Magazinplätzen und Suchstrategien für Ersatzwerkzeuge.
  Bei den Überwachungsfunktionen werden bei Ablauf der aktiven Überwachung Werkzeuge gesperrt und nicht mehr verwendet. Für die weitere Bearbeitung wird - sofern vorhanden - automatisch ein nicht gesperrtes gleichartiges Werkzeug (Ersatzwerkzeug) verwendet.
  - Bereich TO: Alle Daten, die für die Werkzeugverwaltung notwendig sind (Werkzeug- und Magazindaten), liegen in der NC im Bereich TO.
  - TO-Einheit (TOA): Für jeden Kanal kann maximal eine eigene TO-Einheit, als Untermenge des Bereichs TO der NC, eingerichtet werden. Eine TO-Einheit stellt einen unabhängigen Bereich der Werkzeugverwaltung dar. Es existiert keine Verbindung zwischen den TO-Einheiten. Einer TO-Einheit können mehrere Kanäle zugeordnet werden. Ein Kanal kann aber nicht mehreren TO-Einheiten zugeordnet werden. Zweikanalige Systeme sind so voreingestellt, dass jedem Kanal eine eigene TO-Einheit zugeordnet ist (MD28085 $MC_MM_LINK_TOA_UNIT).

- **PLC:**
  - Werkzeugwechsel durchführen
  - Werkzeuge im Magazin bewegen
  - Greifersteuerung
  - gegebenenfalls Magazinsteuerung
  - Sicherheitsverriegelungen
  - Bereitstellung der Struktur der Werkzeugbewegungen in Transferschritt-Tabellen
  - Quittung der Werkzeugbewegungen mit Quittungsschritt-Tabellen
- **SINUMERIK Operate:**
  - Werkzeugdatenanzeige, Ein-/Ausgabe
  - Magazindatenanzeige, Ein-/Ausgabe
  - Umsetzdialog Beladen/Entladen

PLC-Programm

Das PLC-Programm führt die Aufträge der Werkzeugverwaltung aus und quittiert alle Positionsänderungen von Werkzeugen (und Magazinen). Die Überwachung und Verhinderung von möglichen Kollisionen ist alleinige Aufgabe des PLC-Programms, z. B.:

- Mehrere Spindeln benutzen das gleiche Magazin.
- Die Wege von gleichzeitig anstehenden Aufträgen kreuzen sich.
- Solange sich ein großes Werkzeug im Shifter befindet, darf die Kette nicht bewegt werden.

PLC-Firmware

Funktionen der PLC-Firmware:

- Aufträge der Werkzeugverwaltung an das PLC-Programm übergeben
- Quittungen des PLC-Programm an die Werkzeugverwaltung übermitteln
- Rückmeldungen zu jeder Quittung (Quittung fehlerhaft mit Fehlernummer oder Quittung OK) an das PLC-Programm übergeben
- Zusätzlich: Auftragszustand registrieren

3.3 Komponenten der Werkzeugverwaltung

Werkzeugliste, Magazine, Magazinliste

Es können Revolver- und Kettenmagazine verwaltet werden. Andere Magazinarten werden auf diese abgebildet. Beladestellen oder Beladestationen sind als Magazinart für das Be- und Entladen zu verwenden. Ein Magazin-Zwischenspeicher fasst alle weiteren Plätze zusammen, auf denen Werkzeuge platziert werden können (Spindel, Greifer, ...).

Voreinstellung bei Werkzeugmagazinen:

Die Anzahl der Magazine, die die NC verwalten kann, ist vom System vorgegeben:

	SW24x	**SW26x**	**SW28x**
Anzahl der Werkzeuge	128	256	768
Anzahl der Magazine	3 (1 reales Magazin)	3 (1 reales Magazin)	4 (2 reale Magazine)
Anzahl der Magazinplätze	84	84	260
Anzahl der Werkzeugschneiden	256	512	1536

Da immer ein Zwischenspeicher und eine Beladestelle vorhanden sein müssen, können maximal zwei reale Magazine verwaltet werden:

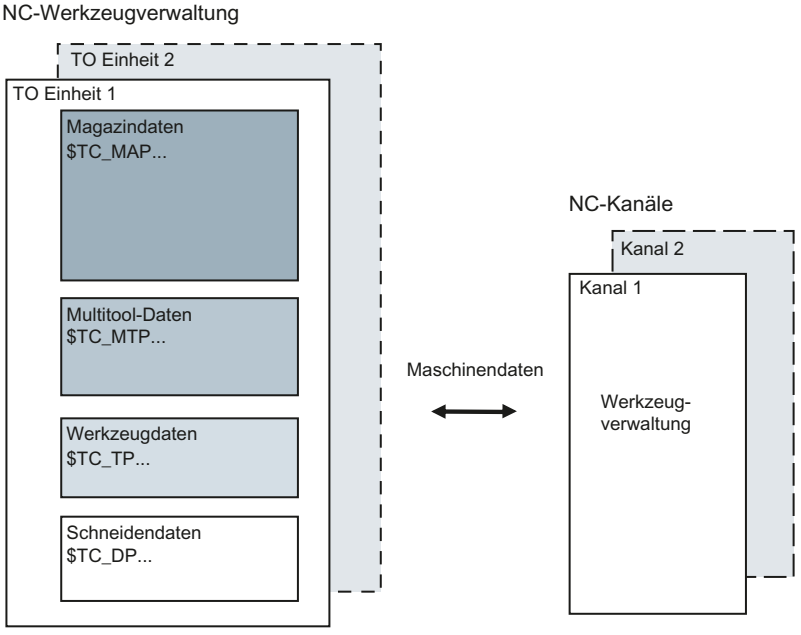


Bild 3-1 Zuordnung TO Einheit zum NC-Kanal

Technologie Drehen: Sonderfall bei SW28x

Anzahl Magazine = 6, davon 1 reales Magazin pro Kanal, wenn Kanal 1 die TO-Einheit 1 und Kanal 2 die TO-Einheit 2 verwendet (Voreinstellung).

Verwenden beide Kanäle gemeinsam die TO-Einheit 1, können bis zu 4 reale Magazine eingerichtet werden.

Magazine

Vom System werden für alle Plätze im Magazin Informationen zur Verfügung gestellt, die Inhalt und Zustand der Plätze beschreiben.

Die Position eines Werkzeugs wird durch einen Bezeichner für das Magazin und einen Bezeichner für den Platz beschrieben. Magazine haben dabei Bezeichner und Nummer, Magazinplätze nur eine Nummer. In einem realen Magazin (Kette, Revolver usw.) ist die Position des Werkzeugs durch die bei der Inbetriebnahme vergebene Magazinnummer und den Platz in diesem Magazin gekennzeichnet.

Beispiel: Die T-Nummer des Werkzeugs des Magazinplatzes 7 aus Magazin 1: $TC_MPP6[1,7]

Werkzeugliste

Die Werkzeugliste enthält alle der NC bekannten Werkzeuge. Das sind die Werkzeuge im Magazin und die entladenen Werkzeuge, deren Daten vorgehalten werden sollen. Die Werkzeugverwaltung arbeitet mit den beladenen Werkzeugen aus der Werkzeugliste.

Magazinliste

Die Magazinliste ist ein platzorientiertes Abbild des Werkzeugmagazins, der Greifer und der Spindel. Die Werkzeugverwaltung arbeitet nur mit den Werkzeugen aus der Magazinliste. Beim Werkzeugwechsel können aber auch zusätzliche Werkzeuge ohne Magazinzuordnung ausgewählt werden. Das Werkzeug muss von Hand in die Maschine eingesetzt werden und nach der Bearbeitung wieder von Hand entnommen werden (Handwerkzeug).

Belademagazin

Das Belademagazin ist das 1. interne Magazin und wird mit der Magazinnummer 9999 geführt. Im Belademagazin sind Beladestellen für das Beladen und Entladen von Werkzeugen enthalten. Bei der Vergabe der Plätze gibt es eine feste Belegung, alle anderen Plätze sind frei vorgebbar. Bei der festen Belegung handelt es sich um den Platz 1 innerhalb des Belademagazins. Der Platz 1 ist für das Beladen/Entladen in alle Spindeln/Werkzeughalter reserviert.

Weiterhin werden alle Positionieraufträge und Umspeicheraufträge zu beliebigen Plätzen (nicht Beladestellen) über Platz 1 abgewickelt. Die genannten Aufträge, die sich auf einen bestimmten Beladeplatz beziehen, werden an der Schnittstelle dieses Beladeplatzes ausgegeben. Die Zuordnung der Beladeplätze zu Magazinen erfolgt bei der Inbetriebnahme ($TC_MDP1). Eine Beladestelle ist ein offener Zugang zum Magazin, an dem das Werkzeug manuell direkt in das Magazin ein- oder auswechselt wird.

Zwischenspeicher

Beim Zwischenspeicher handelt es sich um das 2. interne Magazin. Unter Zwischenspeicher sind Spindel, Werkzeughalter, Greifer, Lader und Übergabeplatz zusammengefasst. Die Zwischenspeicher werden unter der Magazinnummer 9998 geführt. Jedem Zwischenspeicher-Element wird ein eindeutiger Platz zugeordnet. Die Vergabe der Platznummer kann beliebig erfolgen. Es wird empfohlen, alle Spindeln oder Werkzeughalter ab der Nummer 1 aufwärts zu nummerieren. Die Zuordnung zu realen Magazinen oder zwischen Spindel/Werkzeughalter zu anderen Zwischenspeichern wird bei der Inbetriebnahme vorgenommen ($TC_MDP2, $TC_MLSR).

Kettenmagazine

Die Einstellung im MD22550: $MC_TOOL_CHANGE_MODE darf für diese Magazinarten nur den Wert 1 haben.

Bei Kettenmagazinen sind in der Regel zusätzliche Zwischenspeicher zum Transport zwischen Magazin und Spindel vorhanden. In diesen zusätzlichen Zwischenspeichern können sich temporär Werkzeuge befinden.

Bezeichnung der Zwischenspeicher und Beladestellen:

Magazin	Platz	Bedeutung
1	xx	reales Magazin 1 (Kette, Teller, Fläche), Position xx
9998	1	Spindel
9998	2	Greifer
9998	3	Greifer
9998	4	Toolboy
9998	5	Shifter
9999	1	Beladestelle für Spindel, Handwerkzeug
9999	2	Beladestelle für Magazin

Revolvermagazin

Die Einstellung von MD22550 $MC_TOOL_CHANGE_MODE hat üblicherweise den Wert 0.

Bei Revolvermagazinen sind keine zusätzlichen Zwischenspeicher vorhanden um einen Transport von Magazin nach Spindel durchzuführen. Die Werkzeuge bei Revolvermagazinen werden physikalisch nicht in die Spindel transportiert. Sie werden in eine definierte Lage durch Verdrehen des Revolvers bewegt, um die Bearbeitung mit einem Werkzeug durchzuführen. Der Transport in die Spindel oder den Werkzeughalter wird nur datenmäßig durchgeführt. Ein Transport des Werkzeugs nach dem Zwischenspeicher 9998/1 (Spindel) dient dazu der Werkzeugverwaltung mitzuteilen das der Revolver mit dem angeforderten Werkzeug zur Bearbeitungsposition gedreht wurde.

Der Programmierbefehl T = Bezeichner löst den Werkzeugwechsel aus. Alternativ kann auch T = Platz programmiert werden. Bei T = Platz ist es zulässig, dass sich kein Werkzeug auf diesem Platz befindet.

Bezeichnung der Zwischenspeicher und Beladestellen:

Magazin	Platz	Bedeutung
1	xx	reales Magazin 1 (Revolver), Position xx
9998	1	Werkzeughalter
9999	1	Beladestelle für Werkzeughalter, Handwerkzeug

Falls auch für den Revolver in MD22550 $MC_TOOL_CHANGE_MODE der Wert 1 eingestellt ist, gilt die Aussage wie bei Kettenmagazinen.

Nebenplatzbetrachtung

Eine Nebenplatzbetrachtung ist bei der Verwendung von übergroßen Werkzeugen von Bedeutung. Bei der Leerplatzsuche (Beladen, Wechsel) werden dann im Magazinplatz Parameter $TC_MPP4 die Bits 4 ... 11 ausgewertet (Halbplatz belegt/reserviert).

3.4 Multitool

Multitool in der Technologie Drehen

Ein Multitool ist ein Mini-Revolver, der in der Technologie Drehen mit einer Anzahl von Werkzeugen wie ein Werkzeug in ein Magazin beladen und von einem Magazin entladen werden. Die T-Anwahl im Teileprogramm erkennt und prüft für die Werkzeugsuche auch die Werkzeuge eines Multitools.

Kurzbeschreibung**:**

- Mit Hilfe eines Multitools haben Sie die Möglichkeit, mehr als ein Werkzeug auf einem Magazinplatz aufzunehmen.
- Ein Multitool besitzt eine definierte Anzahl von Plätzen zur Aufnahme von Werkzeugen.
- Das Multitool wird in Bezug auf Beladen und Entladen ins Magazin wie ein Werkzeug behandelt.
- Das Multitool wird als fertiges Gebilde betrachtet, das von Hand bestückt wird und als ganzes Be- und Entladen oder transportiert wird. Bestücken eines Multitools mit Werkzeugen oder die Entnahme von Werkzeugen aus einem Multitool erzeugen keinen Auftrag der Werkzeugverwaltung an die PLC-Schnittstelle.
- Das Multitool kennt die Parameter "Anzahl Plätze" und "Art der Abstandsbestimmung der Plätze".
- Die Art des Abstandes beim Mulititool kann über die Platznummer oder über den Winkel des Platzes definiert werden.
- Die Multitool-Namen entstammen demselben Namensraum wie die Werkzeugnamen und Magazinnamen.
- Unterschiedliche Multitools müssen unterschiedliche Namen haben, es wird keine Duplo-Nummer definiert.

Die Funktionalität hat Auswirkungen auf folgende Schnittstellen:

- NC-Sprache (Definition der Daten für Werkzeuganwahl, Werkzeugwechsel, Werkzeugbewegungen)
- NC-Variablen (Definition der Daten und Visualisierung)
- NC/PLC-Schnittstelle

Die Beschreibung der Programmierung sowie der Multitool-Daten finden Sie in: SINUMERIK 840D sl Funktionshandbuch "Werkzeugverwaltung (https://support.industry.siemens.com/cs/document/109752352)"

3.5 Handwerkzeuge

Handwerkzeuge

Handwerkzeuge sind Werkzeuge, deren Daten vollständig in der NC vorhanden sind, die aber nicht in das Magazin beladen werden. Ob Handwerkzeuge in der Anlage zulässig sind wird in MD22562 $MC_TOOL_CHANGE_ERROR_MODE Bit 1 eingestellt. Das automatisch ausgewählte Werkzeug muss von Hand in die Maschine eingesetzt und nach der Bearbeitung wieder von Hand entnommen werden.

Zu den Handwerkzeugen zählen folgende Arten von Werkzeugen:

- Übergroße Werkzeuge
- Werkzeuge, die nicht im Magazin aufbewahrt werden können.
- Werkzeuge, die nicht über das Greifersystem der Anlage laufen dürfen.

Verantwortung des Bedieners

Der Maschinenbediener hat dafür zu sorgen,

- dass der Datensatz des Werkzeugs, das er auf die Spindel steckt, sich in der NC befindet.
- dass er zu dem in der NC abgelegten Datensatz das passende Werkzeug auf die Spindel bringt.

⚠ VORSICHT
Sicherheit der Maschine Für die Einhaltung der Sicherheitsvorschriften muss der Maschinenbediener über das PLC-Programm sorgen.

Eindeutige Kennzeichnung von Handwerkzeugen

Ein Handwerkzeug bekommt eine eindeutige Kennung durch $TC_TP8[T_Nr] Bit 15 = 1. Ist dieser Zustand gesetzt, wird für das Werkzeug kein Leerplatz in einem Magazin gesucht und dieses (Alt–)Werkzeug beim Werkzeugwechsel direkt auf den virtuellen Magazinplatz 9999/1 transportiert. Im Werkzeugwechsel-Unterprogramm kann das Erkennen der Kennzeichnung mit einem bitweisem UND erfolgen:

IF $TC_TP8[_TOOL_IN_SPINDLE] B_AND 'B1000000000000000' GOTOF HANDTOOL

Setzen des Zustandes "Handwerkzeug":

Der Zustand "Handwerkzeug" eines Werkzeuges wird gesetzt, wenn folgende Bedingungen erfüllt sind:

- Das Maschinendatum ist gesetzt: MD22562 $MC_TOOL_CHANGE_ERROR_MODE Bit 1 = 1
- Im Rahmen eines Werkzeugwechsels wurde eine nicht beladene Wechselstelle angewählt.

Zurücksetzen des Zustandes "Handwerkzeug":

Der Zustand "Handwerkzeug" eines Werkzeuges wird in folgenden Fällen zurückgesetzt:

- Das Werkzeug wird entladen, d.h. das Werkzeug hat keinen aktuellen Magazinplatz mehr. Dies kann insbesondere erfolgen durch:
  - Das Auswechseln des Altwerkzeuges im Rahmen des Werkzeugwechsels.
  - Den Transport des Handwerkzeuges auf eine Beladestelle (Entladen über Bedienung am SINUMERIK Operate).
  - Das Kommando zur Anwahl oder zum Einwechseln des Handwerkzeuges wird abgebrochen und das Werkzeug wurde noch nicht auf einen Magazinplatz bewegt.
- Das Werkzeug wurde auf einen Platz eines realen Magazins oder des Beladem agazin gebracht.

Hinweis

Alarme für Handwerkzeuge

Sie erhalten zu einem Werkzeugwechsel, bei dem ein Handwerkzeug beteiligt ist, immer einen Alarm (17212, 17214 oder 17216). Der Alarm wird durch die Werkzeugwechsel-Quittung des PLC-Programms gelöscht.

Die Alarme sind ausblendbar: MD11415 $MN_SUPPRESS_ALARM_MASK_2 Bit 16 = 1

Werkzeug über SINUMERIK Operate laden

Ein Werkzeug, das z. B. über SINUMERIK Operate auf die Spindel geladen wurde, ist kein Handwerkzeug und wird infolge dessen mit T0/M06 im Magazin abgelegt.

Maschinendaten für die Werkzeugverwaltung

4

Voreinstellung Maschinendaten

Folgende Maschinendaten für die Werkzeugverwaltung sind bereits voreingestellt oder werden beim Hochlauf mit "NCK default data" gesetzt.

Technologie Drehen:

Nummer	Name	Voreinstellung
MD10717	$MN_T_NO_FCT_CYCLE_NAME	TCHANGE
MD17500	$MN_MAXNUM_REPLACEMENT_TOOLS	1
MD20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1
MD20310	$MC_TOOL_MANAGEMENT_MASK	181400F
MD22550	$MC_TOOL_CHANGE_MODE	0
MD22560	$MC_TOOL_CHANGE_MCODE	6
MD22562	$MC_TOOL_CHANGE_ERROR_MODE	0 (für Handwerkzeug)

Technologie Fräsen:

Nummer	Name	Voreinstellung
MD10715[0]	$MN_M_NO_FCT_CYCLE	6
MD10716[0]	$MN_M_NO_FCT_CYCLE_NAME	L6
MD17500	$MN_MAXNUM_REPLACEMENT_TOOLS	1
MD20124	$MC_TOOL_MANAGEMENT_TOOLHOLDER	1
MD20310	$MC_TOOL_MANAGEMENT_MASK	180400F
MD22550	$MC_TOOL_CHANGE_MODE	1
MD22560	$MC_TOOL_CHANGE_MCODE	206
MD22562	$MC_TOOL_CHANGE_ERROR_MODE	0 (für Handwerkzeug)

MD20270: $MC_CUTTING_EDGE_DEFAULT

Grundstellung der Werkzeugschneide ohne Programmierung (DWORD): Wird nach einem Werkzeugwechsel keine Schneide programmiert, so wird die Schneidennummer verwendet, die in $MC_CUTTING_EDGE_DEFAULT eingestellt ist.

MD20270: $MC_CUTTING_EDGE_DEFAULT	
> 0	Nummer der Schneide, die mit M206 angewählt wird. Die Schneidenanwahl erfolgt auch wenn anschließend eine D-Programmierung erfolgt.
= 1	Voreinstellung
= 0	Nach einem Werkzeugwechsel ist zunächst keine Schneide aktiv. Eine vor dem Werkzeugwechsel aktive Werkzeugkorrektur wird abgewählt (entspricht D0!). Die Schneidenanwahl erfolgt erst bei D-Programmierung.

MD20270: $MC_CUTTING_EDGE_DEFAULT	
= -1	Schneidennummer des alten Werkzeugs gilt auch für das neue Werkzeug
= -2	Schneidenkorrektur des alten Werkzeugs bleibt weiterhin aktiv bis D programmiert wird.

Die Einstellung des MD20270 hat Einfluss auf die Satzaufbereitung der NC. Um zu verhindern, dass mit Aufruf des Wechselbefehls ein Vorlauf-Stopp bis zur Quittung des Werkzeugwechsels eintritt, führen Sie NC-Funktionen ohne Werkzeugkorrektur im Werkzeugwechsel-Unterprogramm aus, zum Beispiel Achsen verfahren oder die Ausgabe von Hilfsfunktionen.

Beispiel: MD20270: $MC_CUTTING_EDGE_DEFAULT= 0 oder = -2

Nach dem Aufruf M206 zum Werkzeugwechsel können die Achsen sofort weiterfahren, ohne auf die Wechselquittung zu warten und Verfahrsätze ohne Werkzeugkorrektur ausführen. Erst im Satz mit einer Korrekturanwahl (D-Nr.) wird angehalten bis der Werkzeugwechsel von der PLC als beendet gemeldet wurde.

Teileprogramm:

```
N10 T="Bohrer18"          ; Werkzeugwechselvorbereitung
N20 M6                    ; Aufruf des Werkzeugwechsel-Unterprogramms
Werkzeugwechsel-Unterprogramm L6:
N10 D0                    ; Korrekturabwahl
N20 M206                  ; Werkzeugwechsel
N40 Y150 M79              ; Maschinenachsen verfahren
N50 G01 D1 X10            ; Einschalten der Werkzeugkorrektur.
                          ; Prüfung, ob der Werkzeugwechsel erfolgt ist.
                          ; Der Vorlauf-Stopp wird solange aufrechterhalten bis die
                          Werkzeugwechsel-Vorbereitung abgeschlossen wurde. Der
                          Hauptlauf wartet bei N50 (D1) bis der Werkzeugwechsel
                          durchgeführt und quittiert ist.
```

MD20310: $MC_TOOL_MANAGEMENT_MASK

Aktivierung der Werkzeugverwaltungsfunktionen:

MD20310: $MC_TOOL_MANAGEMENT_MASK		
Bit 0	= 1	Werkzeugverwaltung aktiv: Die Funktionen der Werkzeugverwaltung sind für den aktuellen Kanal frei geschaltet.
Bit 1	= 1	Überwachungsfunktionen der Werkzeugverwaltung aktiv: Die Funktionen für die Überwachung der Werkzeuge (Standzeit und Stückzahl) werden frei geschaltet.
Bit 2	= 1	OEM-Funktionen aktiv
Bit 3	= 1	Nebenplatzbetrachtung aktiv
Bit 14	= 1	Werkzeug- und Korrekturanwahl entsprechend den Einstellungen in: MD20110 $MC_RESET_MODE_MASK MD20112 $MC_START_MODE_MASK
Bit 23	= 1	Bei Korrekturanwahl keine Synchronisation mit Hauptlauf.

MD20310: $MC_TOOL_MANAGEMENT_MASK		
Bit 24	= 1	Mit einem asynchronen Transfer mit Quittungsstatus 201 können Sie ein Werkzeug auf einen für ein anderes Werkzeug mit "reserviert für Werkzeug aus Zwischenspeicher" reservierten Platz bewegen. Diese Platzreservierung wird dazu vor der Ausführung der Bewegung entfernt ("reserviert für neu zu beladenes Werkzeug " (Bitwert="H8") bleibt wirksam).
zusätzlich nur bei T-Version:		
Bit 16	= 1	T-Platznummer ist aktiv.

Befindet sich ein gesperrtes Werkzeug auf dem programmiertem Platz, wird bei gesetzter Option "Ersatzwerkzeuge für Werkzeugverwaltung" der Platz eines (wenn vorhandenen) Schwesterwerkzeugs als Auftrag von der Werkzeugverwaltung ausgegeben.

MD52270: $MCS_TM_FUNCTION_MASK

Funktionsmaske Werkzeugverwaltung

MD52270: $MCS_TM_FUNCTION_MASK	
Bit 0:	Werkzeug anlegen auf Magazinplatz nicht zugelassen. Werkzeuge können nur außerhalb des Magazins angelegt werden.
Bit 1:	Be-/Entladesperre, wenn die Maschine nicht im Reset-Zustand ist. Werkzeuge können nur be-/entladen werden, wenn der entsprechende Kanal im Reset-Zustand ist.
Bit 2:	Be-/Entladesperre bei NOT-HALT. Werkzeuge können nur be-/entladen werden, wenn NOT-HALT nicht aktiv ist
Bit 3:	Werkzeug in/aus Spindel be-/entladen gesperrt. Werkzeuge können nicht in die Spindel geladen oder aus der Spindel entladen werden.
Bit 4:	Beladen erfolgt direkt in die Spindel. Das Beladen von Werkzeugen erfolgt ausschließlich direkt in die Spindel.
Bit 5:	reserviert
Bit 6:	reserviert
Bit 7:	Werkzeug über die T-Nummer anlegen. Beim Werkzeug anlegen muss die T-Nummer des Werkzeugs eingegeben werden.
Bit 8:	Werkzeug umsetzen ausblenden. Die Funktion "Werkzeug umsetzen" wird in der Bedienoberfläche ausgeblendet.
Bit 9:	Magazin positionieren ausblenden. Die Funktion "Magazin positionieren" wird in der Bedienoberfläche ausgeblendet.
Bit 10:	Werkzeug reaktivieren mit Magazin positionieren. Vor dem Reaktivieren wird das Werkzeug auf die Beladestelle positioniert.
Bit 11:	Werkzeug reaktivieren in allen Überwachungsarten. Beim Reaktivieren eines Werkzeugs werden alle in der NC freigegebenen Überwachungsarten zu diesem Werkzeug reaktiviert. D.h. auch die Überwachungsarten, die für das jeweilige Werkzeug nicht eingestellt sind, sondern nur im Hintergrund liegen.
Bit 12:	Werkzeug reaktivieren ausblenden. Die Funktion "Werkzeug reaktivieren" wird in der Bedienoberfläche ausgeblendet.

Voreinstellung für Technologie Drehen

Für die Technologie Drehen bestimmen folgende Maschinendaten die Funktion:

- **MD22550: $MC_TOOL_CHANGE_MODE = 0**
  Einstellung für ein Revolvermagazin: das neue Werkzeug wird mit der T-Funktion sofort eingewechselt. Es wird kein zusätzlicher M-Befehl verwendet. Es wird nicht zwischen Werkzeugwechsel vorbereiten und Werkzeugwechsel durchführen unterschieden.
  Die Funktion "Handwerkzeuge" ist für diesen Fall nicht freigegeben.
- **MD20310: $MC_TOOL_MANAGEMENT_MASK = 81400F (Bit 16=1)**
  Mit Bit 16 wird die Programmierart des Werkzeugs eingestellt:
  T = "x" mit x als Werkzeug-Bezeichner
  Tx, mit x als Platznummer des Magazins, mit dessen Werkzeug gearbeitet wird.
  Bei aktivierter Funktion wird mit T1 anstelle des Werkzeugs mit dem Bezeichner "1" das Werkzeug auf dem Platznummer 1 angewählt. Dann wird der Bezeichner des Werkzeugs auf diesem Platz ermittelt (z. B. "SCHLICHTER"). Es wird so verfahren als wäre T="SCHLICHTER" programmiert worden.
  Bei T = Platznummer ist es zulässig, dass sich kein Werkzeug auf diesem Platz befindet.
- **MD10717: $MN_T_NO_FCT_CYCLE_NAME = TCHANGE**
  Name des Werkzeugwechselzyklus für T-Funktions-Ersetzung.
  Eine Beschreibung finden Sie im Kapitel Beispiel: Werkzeugwechselzyklus für Drehmaschine (Seite 81)

ISO-Dialekte in der Technologie Drehen:

Voreinstellung für die Programmierung der Werkzeugkorrektur:

MD10888	$MN_EXTERN_DIGITS_TOOL_NO = 0
MD10889	$MN_EXTERN_DIGITS_OFFSET_NO = 2
MD10890	$MN_EXTERN_TOOLPROG_MODE = 4

Eine Programmierung von T101 und T0101 führt zum gleichen Ergebnis: T1 H01

Weitere Informationen finden Sie in: SINUMERIK 828D/840D sl Funktionshandbuch ISO-Dialekte (https://support.industry.siemens.com/cs/document/65371498)

- Für ISO-Mode (G291) gilt die Beschreibung "Substitution durch einen Ersetzungszyklus bei einem Werkzeuganwahlsatz".
- Für die Programmierung der Werkzeugkorrektur gilt die Beschreibung "Werkzeugwechsel und Werkzeugkorrekturen".

Magazin konfigurieren

Nach einer Änderung der Magazinkonfiguration ist ein Power-On Reset notwendig. Die geänderte Konfiguration wird erst nach einem Neustart der NC angezeigt.

Beispiele:

Kapitel Konfiguration eines Kettenmagazins mit einem Doppelgreifer (Seite 87)

Kapitel Konfiguration eines Revolvermagazins: (Seite 75) Das Programm ist auf der Toolbox für SINUMERIK 828D im Verzeichnis "Examples" enthalten.

NC-PLC-Schnittstelle der Werkzeugverwaltung

5

Übersicht

Die Werkzeugverwaltung erhält Aufträge zur Vorbereitung und Durchführung eines Werkzeugwechsels (T-Befehl, M06), einer Werkzeugbewegung (MVTOOL) oder einer Magazinpositionierung (POSM) vom Teileprogramm oder von SINUMERIK Operate. Aus diesen Aufträgen ermittelt die Werkzeugverwaltung die notwendigen Platzwechsel der Werkzeuge und beauftragt damit die PLC.

Die NC-/PLC-Schnittstelle stellt getrennte Datenbausteine für Entladen, Beladen, Umsetzen und Magazin positionieren einerseits und Werkzeugwechsel andererseits zur Verfügung.

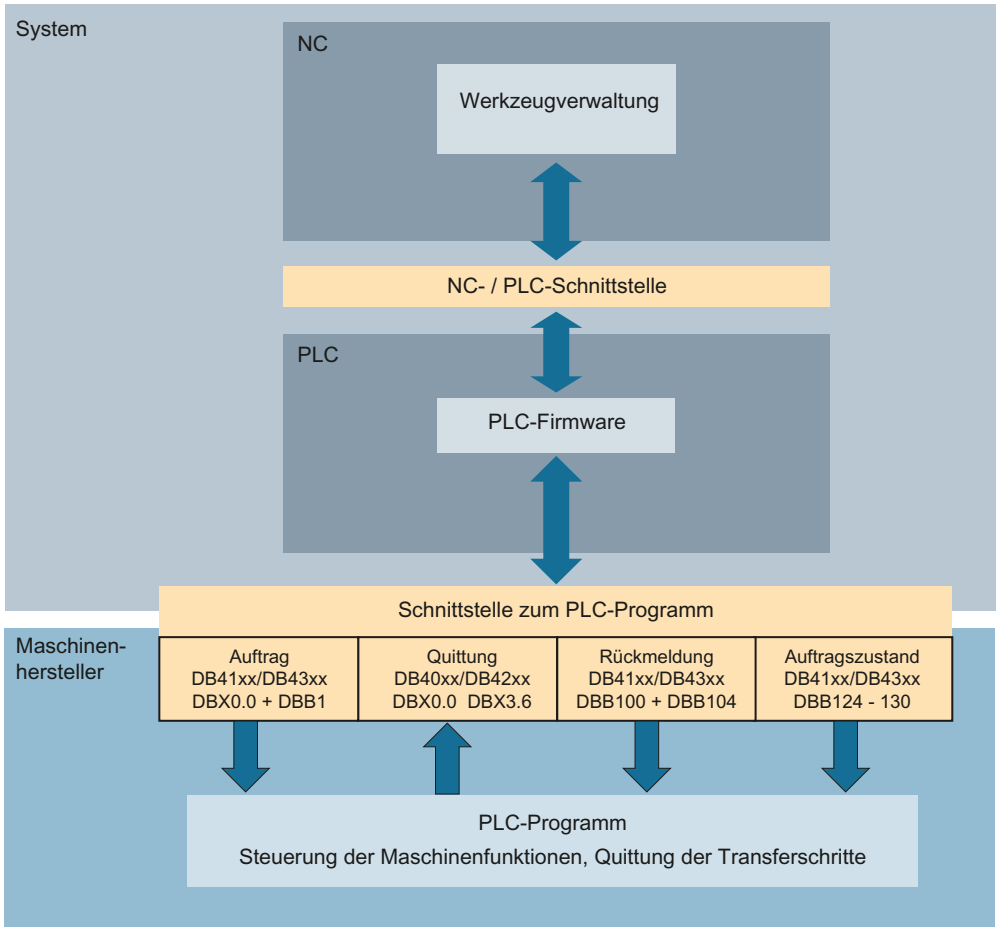


Bild 5-1 Schnittstellen der Werkzeugverwaltung

Beladen/Entladen und Umsetzen

Beladen/Entladen und Umsetzen: Werkzeugbewegung (MVTOOL) oder Magazinpositionierung (POSM) von Teileprogramm oder SINUMERIK Operate

xx: Index der Beladestelle

y: Index der TO-Einheit

- 40xx y000 Quittungen vom PLC-Programm zum Beladen/Entladen und Umsetzen, Magazin positionieren
- 41xx y000 Aufträge an Werkzeugverwaltung zum Beladen/Entladen und Umsetzen, Magazin positionieren
- 41xx y100 Rückmeldung von Werkzeugverwaltung
- 41xx y120 Auftragszustand

Werkzeugwechsel

Werkzeugwechsel (T-Befehl, M06)

xx: Index der Spindel

y: Index der TO-Einheit

- 42xx y000 Quittungen vom PLC-Programm zum Werkzeugwechsel vorbereiten und durchführen
- 43xx y000 Aufträge an Werkzeugverwaltung zum Werkzeugwechsel vorbereiten und durchführen
- 43xx y100 Rückmeldung von Werkzeugverwaltung
- 43xx y120 Auftragszustand

Transferschritt- und Quittungsschritt-Tabellen:

- 9900 0000 konstante Transferschritt-Tabelle (projektierbar)
- 9901 0000 variable Transferschritt-Tabelle (projektierbar und änderbar)
- 9902 0000 Quittungsschritt-Tabelle (projektierbar)

Beispiel:

Für jede TO-Einheit steht ein eigener Datenbereich in den jeweiligen Datenbausteinen zur Verfügung:

- Werkzeughalter 1 in TO-Einheit 1: DB4x00.0xxx
- Werkzeughalter 2 in TO-Einheit 1: DB4x01.0xxx
- Werkzeughalter 1 in TO-Einheit 2: DB4x00.1xxx
- Werkzeughalter 2 in TO-Einheit 2: DB4x01.1xxx

5.1 Werkzeug umsetzen, entladen, beladen, Magazin positionieren

Werkzeug- und Magazinbewegungen

Pro Beladestelle gibt es je eine Schnittstelle für:

- **Aufträge** zum Beladen, Entladen, Umsetzen von Werkzeug (MVTOOL) und zum Magazinpositionieren (POSM).
  Meldung an das PLC-Programm:
  - Auftrag aktiv
  - Auftragsspezifikation
  - Auftragsbeschreibung

  Die Aufträge erscheinen in der Schnittstelle der Beladestelle, von der aus ein Werkzeug eingewechselt oder wohin ein Werkzeug ausgewechselt werden soll.
- **Quittungen** des PLC-Programms
  Alle Quittungen zu einem Auftrag müssen in der Schnittstelle der gleichen Beladestelle erfolgen. Quittungsfehler sind ebenfalls in dieser Schnittstelle zurückzusetzen.
- **Rückmeldung** der Werkzeugverwaltung an das PLC-Programm
  Meldung an das PLC-Programm:
  - Zustand der Quittung
  - Fehlerstatus
  - Abbild der Quittungsbits
- **Auftragszustand**
  Ausgewählte Daten der letzten Zwischen- oder Ende-Quittung werden gespeichert. Diese Daten werden von der PLC-Firmware für die nächste Quittung an die Werkzeugverwaltung benötigt und sind zu Diagnosezwecken lesbar. Vom PLC-Programm können diese Daten für das Wiederaufsetzen nach einem Abbruch (z. B. Reset während eines Werkzeugwechsels) genutzt werden.

Alle Quittungen zu einem Auftrag müssen in der Schnittstelle der gleichen Beladestelle erfolgen.

Nahtstellensignal	Bedeutung
DB40xx.DBXy000.0 – DBXy003.6	PLC-Programm: Quittungen zum Werkzeug Be-/Entladen, Umsetzen oder zum Positionieren des Magazins
DB40xx.DBXy009.0	PLC-Programm: Rücksetzen der Meldung "Quittung fehlerhaft" (DB41xx.DBX100.1) und der Diagnose-Informationen in der Rückmeldeschnittstelle
DB41xx.DBXy000.0	Werkzeugverwaltung: Auftrag zum Werkzeug Be-/Entladen, Umsetzen oder zum Positionieren des Magazins
DB41xx.DBBy001	Werkzeugverwaltung: Auftragsspezifikation
DB41xx.DBWy006 – DBWy034	Auftragsbeschreibung

Nahtstellensignal	Bedeutung
DB41xx.DBXy100:0 DB41xx.DBXy100.1	Positive Rückmeldung: Quittungszustand, Quittung OK, 1 PLC-Zyklus anstehend Negative Rückmeldung: Quittungszustand, Quittung fehlerhaft, statisch anstehend
DB41xx.DBBy104	Werkzeugverwaltung: Rückmeldung Fehlerstatus
DB41xx.DBXy108.0 - DB41xx.DBXy111.6	Abbild der Quittungen zum Werkzeug beladen, entladen, umsetzen oder zum Magazin positionieren. Dieses Abbild gehört zur positiven oder negativen Rückmeldung und ist so lange wie diese gültig.
DB41xx.DBWy124 – DBWy130	Auftragszustand

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Beladestelle

Aufträge: Schnittstellenbeschreibung

DB4100...41xx	Signale von Werkzeugverwaltung [r]							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy000	Multitool: Daten im Multitool-Bereich							Auftrag ist noch aktiv.
DBBy001		Multitool: Positionieren ist NC-Aktivität	Multitool: Positionieren	Positionieren ist NC-Aktivität	Positionieren	Umsetzen	Entladen	Beladen
DBBy002	reserviert							
DBBy003	reserviert							
DBBy004	reserviert							
DBBy005	reserviert							
DBWy006	Quell-Magazinnummer (INT)							
DBWy008	Quell-Platznummer (INT)							
DBWy010	Ziel-Magazinnummer (INT)							
DBWy012	Ziel-Platznummer (INT)							
DBWy014: HMI → PLC								Beladen/ Entladen ohne Magazinbewegung
DBWy016								
...								
DBWy056	Multitool: Typ (INT)							
DBWy058	Multitool: Platzanzahl (INT)							
DBWy060	Multitool: Werkzeugabstand (FLOAT)							
DBWy062								
DBWy064	Multitool: Nummer (INT)							

DB4100...41xx	Signale von Werkzeugverwaltung [r]
DBWy066	Multitool: Platznummer (INT)
DBWy068	Multitool: Spindelnummer/Werkzeughalternummer (INT)

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Beladestelle

Aufträge: Signalbeschreibung

Signalbeschreibung:

- Auftrag:
  Die Schnittstelle enthält einen Auftrag. Die Auftragsbearbeitung wurde noch nicht mit einer Ende-Quittung abgeschlossen. Nach Übertragung der Ende-Quittung an die Werkzeugverwaltung wird dieses Signal rückgesetzt.
- Multitool: Daten im Multitool-Bereich
  Die Schnittstelle enthält Informationen zum im Auftrag angegebenen Multitool.
- Beladen:
  Über die Beladestelle des Parameters "Quellplatz" soll der Magazinplatz des Parameters "Zielplatz" mit einem Werkzeug beladen werden.
- Entladen:
  Das Werkzeug im Magazinplatz des Parameters "Quellplatz" soll nach der Entladestelle des Parameters "Zielplatz" entladen werden.
- Umsetzen:
  Das Werkzeug im Magazinplatz des Parameters "Quellplatz" soll nach dem Magazinplatz des Parameters "Zielplatz" umgesetzt werden.
- Positionieren:
  Der Magazinplatz des Parameters "Quellplatz" soll an die Wechsel-/Be-/Entladestelle des Parameters "Zielplatz" positioniert werden. Das Werkzeug bleibt in seinem Magazinplatz.
- Positionieren ist NC-Aktivität:
  Der Positionierauftrag kommt vom Teileprogramm.
- Multitool: Positionieren
  Das Werkzeug im Multitool wird in die Bearbeitungsposition bewegt. Das Multitool bleibt im Werkzeughalter/Spindel.
- Multitool: Positionieren ist NC-Aktivität:
  Der Positionierauftrag kommt vom Teileprogramm.
- Quellplatz:
  Magazin- und Platznummer eines Werkzeugs das fortbewegt oder an eine Wechsel- oder Beladstelle positioniert werden soll.
- Zielplatz:
  Magazin- und Platznummer wohin ein Werkzeug bewegt oder ein Magazinplatz positioniert werden soll.

- Beladen/Entladen ohne Magazinbewegung:
  Das Signal wird bei Einlesen von Rüstdaten oder der Funktion "Alle Werkzeuge entladen" zusätzlich zu den Aufträgen für Be- oder Entladen vom HMI gesetzt. Das Signal kann im PLC-Programm zur Verriegelung unerwünschter Magazinbewegungen verwendet werden (z. B. nur Totalquittung setzen, keine weitere Aktion). Das Signal wird mit gültiger Ende-Quittung des Auftrags an die Werkzeugverwaltung vom System zurückgesetzt.
- Mutlitool: Typ:
  Art der Abstandskodierung: 1 = Platznummer, 2 = Länge, 3 = Winkel
- Multitool: Platzanzahl
  Anzahl der im Multitool vorhandenen Werkzeugplätze
- Multitool: Werkzeugabstand
  Wert für die Abstandskodierung: Platznummer des Werkzeugs im Multitool oder Winkel oder Abstand des Werkzeugs innerhalb des Multitools
- Multitool: Nummer
  T-Nummer des Multitools
- Multitool: Platznummer
  Fortlaufend nummerierter Platz im Multitool
- Multitool: Spindelnummer / Werkzeughalternummer
  Zum Multitool gehörige Spindelnummer / Werkzeughalternummer

Quittungen: Schnittstellenbeschreibung

DB4000...40xx	**Signale an Werkzeugverwaltung [r/w]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy000	Quittungs-schritt 7	Quittungs-schritt 6	Quittungs-schritt 5	Quittungs-schritt 4	Quittungs-schritt 3	Quittungs-schritt 2	Quittungs-schritt 1	Total-Quit-tung
DBBy001	Quittungs-schritt 15	Quittungs-schritt 14	Quittungs-schritt 13	Quittungs-schritt 12	Quittungs-schritt 11	Quittungs-schritt 10	Quittungs-schritt 9	Quittungs-schritt 8
DBBy002	Quittungs-schritt 23	Quittungs-schritt 22	Quittungs-schritt 21	Quittungs-schritt 20	Quittungs-schritt 19	Quittungs-schritt 18	Quittungs-schritt 17	Quittungs-schritt 16
DBBy003	reserviert	Quittungs-schritt 30	Quittungs-schritt 29	Quittungs-schritt 28	Quittungs-schritt 27	Quittungs-schritt 26	Quittungs-schritt 25	Quittungs-schritt 24
DBWy004	Multitool: Platznummer (INT)							
DBBy006	reserviert							
DBBy007	reserviert							
DBBy008	reserviert							
DBBy009							Auftrags-quittung	Rückset-zen Quit-tungsfehler
DBWy010 • • •	reserviert							
DBWy026	reserviert							
DBBy028	Quittungs-schritt 38	Quittungs-schritt 37	Quittungs-schritt 36	Quittungs-schritt 35	Quittungs-schritt 34	Quittungs-schritt 33	Quittungs-schritt 32	Quittungs-schritt 31
DBBy029	Quittungs-schritt 46	Quittungs-schritt 45	Quittungs-schritt 44	Quittungs-schritt 43	Quittungs-schritt 42	Quittungs-schritt 41	Quittungs-schritt 40	Quittungs-schritt 39

DB4000...40xx	Signale an Werkzeugverwaltung [r/w]							
DBBy030	Quittungs-schritt 54	Quittungs-schritt 53	Quittungs-schritt 52	Quittungs-schritt 51	Quittungs-schritt 50	Quittungs-schritt 49	Quittungs-schritt 48	Quittungs-schritt 47
DBBy031	reserviert	reserviert	Quittungs-schritt 60	Quittungs-schritt 59	Quittungs-schritt 58	Quittungs-schritt 57	Quittungs-schritt 56	Quittungs-schritt 55

xx: Beladestelle

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

Quittungen: Signalbeschreibung

Signalbeschreibung:

- Total-Quittung:
  Bei einer 0/1-Flanke wird zum aktuellen Auftrag die Ende-Quittung mit Status 99 gesendet (Auftrag komplett erledigt, alle Ziel-Positionen sind erreicht). Solange das Signal ansteht, dürfen keine Änderungen an den Daten dieser Schnittstelle vorgenommen werden!
  Nach der Übertragung der Quittung an die Werkzeugverwaltung wird dieses Signal von der PLC-Firmware zurückgesetzt.
- Quittungsschritt 1...60:
  Bei einer 0/1-Flanke wird der zugehörige Quittungsschritt aus der Quittungsschritt-Tabelle an die Werkzeugverwaltung gesendet. Solange das Signal ansteht, dürfen keine Änderungen an den Daten dieser Schnittstelle und der variablen Transferschritt-Tabelle vorgenommen werden!
  Nach der Übertragung der Quittung an die Werkzeugverwaltung wird dieses Signal von der PLC-Firmware zurückgesetzt.
- Multitool: Platznummer
  Das PLC-Programm meldet die aktive Platznummer an die Werkzeugverwaltung der NC zurück.
- Rücksetzen Quittungsfehler:
  Rücksetzen der Meldung "Quittung fehlerhaft" (DB41xx.DBX100.1) und der Diagnose-Informationen in der Rückmeldeschnittstelle.
- Auftragsquittung:
  Dieses Bit wirkt nur bei Totalquittung DB4000.DBX0.0 = 1
  DB4000.DBX9.1 = 0 ⇒ Auftragsquittung mit Status 99
  DB4000.DBX9.1 = 1 ⇒ Auftragsquittung mit Status 3

Rückmeldungen: Schnittstellenbeschreibung

DB4100...41xx	Signale von Werkzeugverwaltung [r]							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy100							Quittung fehlerhaft	Quittung OK
DBBy101	reserviert							
DBWy102	reserviert							
DBWy104	Fehlerstatus (WORD)							
DBBy106	reserviert							
DBBy107	reserviert							

DB4100...41xx	**Signale von Werkzeugverwaltung [r]**							
DBBy108	Quittungs-schritt 7	Quittungs-schritt 6	Quittungs-schritt 5	Quittungs-schritt 4	Quittungs-schritt 3	Quittungs-schritt 2	Quittungs-schritt 1	Quittungs-schritt 0
DBBy109	Quittungs-schritt 15	Quittungs-schritt 14	Quittungs-schritt 13	Quittungs-schritt 12	Quittungs-schritt 11	Quittungs-schritt 10	Quittungs-schritt 9	Quittungs-schritt 8
DBBy110	Quittungs-schritt 22	Quittungs-schritt 22	Quittungs-schritt 21	Quittungs-schritt 20	Quittungs-schritt 19	Quittungs-schritt 18	Quittungs-schritt 17	Quittungs-schritt 16
DBBy111	reserviert	Quittungs-schritt 30	Quittungs-schritt 29	Quittungs-schritt 28	Quittungs-schritt 27	Quittungs-schritt 26	Quittungs-schritt 25	Quittungs-schritt 24
DBWy112	Multitool: Platznummer (INT)							
• • •								
DBBy152	Quittungs-schritt 38	Quittungs-schritt 37	Quittungs-schritt 36	Quittungs-schritt 35	Quittungs-schritt 34	Quittungs-schritt 33	Quittungs-schritt 32	Quittungs-schritt 31
DBBy153	Quittungs-schritt 46	Quittungs-schritt 45	Quittungs-schritt 44	Quittungs-schritt 43	Quittungs-schritt 42	Quittungs-schritt 41	Quittungs-schritt 40	Quittungs-schritt 39
DBBy154	Quittungs-schritt 54	Quittungs-schritt 53	Quittungs-schritt 52	Quittungs-schritt 51	Quittungs-schritt 50	Quittungs-schritt 49	Quittungs-schritt 48	Quittungs-schritt 47
DBBy155	reserviert	reserviert	Quittungs-schritt 60	Quittungs-schritt 59	Quittungs-schritt 58	Quittungs-schritt 57	Quittungs-schritt 56	Quittungs-schritt 55

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Beladestelle

Rückmeldungen: Signalbeschreibung

Signalbeschreibung:

- **Quittung OK (DB41xx.DBXy100.0):**
  Die Quittung des PLC-Programm (Bereich DB40xx.DBB0 bis DBB3) wurde fehlerfrei an die Werkzeugverwaltung übertragen. Dieses Signal wird nach einem PLC-Takt rückgesetzt.
- **Quittung fehlerhaft (DB41xx.DBXy100.1):**
  Negative Rückmeldung Quittungszustand. Die Quittung des PLC-Programm (Bereich DB40xx.DBB0 bis DBB3) ist fehlerhaft. Die Fehlerursache wird in "Fehlerstatus" mitgeteilt. Das Bit "Quittung fehlerhaft" wird auch gesetzt, wenn die Quittung von der PLC-Firmware als fehlerfrei akzeptiert wurde und die Werkzeugverwaltung einen Fehler im quittierten Werkzeugtransfer erkennt und diesen meldet (z. B. wenn der Zielplatz des Werkzeugtransfers belegt ist).
  Fehler, die direkt von der Werkzeugverwaltung in der NC erkannt werden, ohne dass vom PLC-Programm eine Quittung gegeben wurde, führen nicht zum Setzen des Bits 100.1.
  Liegt ein Fehler vor, der die Weitergabe der Quittung an die Werkzeugverwaltung verhindert (Fehlerstatus 1 bis 7), wird der Fehler nur an der Nahtstelle der Werkzeugverwaltung und nicht von der NC ausgegeben (kein NC Alarm!).
  Bei Bedarf melden Sie diese Fehler mit einem Hersteller-PLC-Alarm durch das PLC-Programm.
  Dieses Signal bleibt bis zum Quittieren des Fehlers statisch anstehen (Setzen des Bits "Rücksetzen Quittungsfehler" DB40xx.DBX9.0 im PLC-Programm). Die Nahtstelle in DB40xx.DBB0 bis DBB3 ist bei anstehendem Bit "Quittung fehlerhaft" gesperrt. Eingehende Quittungsbits werden von der PLC-Firmware nicht ausgewertet und mit dem Setzen des Bits "Rücksetzen Quittungsfehler" gelöscht.
- **Multitool: Platznummer**
  Multitool: Platznummer, die das PLC-Programm als aktive Platznummer beim Quittieren in die Quittungsschnittstelle eingetragen hat.

Fehlerstatus

Bei einem Fehler steht in DB41xx.DBBy104 eine Diagnosenummer ungleich Null mit folgender Bedeutung:

Diagnosenummer	Bedeutung
0	kein Fehler
1	mehrere Quittungssignale gleichzeitig
2	Quittung ohne Auftrag
3	ungültige Transferschritt-Nummer
4	für eine Positionsangabe fehlt der Auftrag
5	Status erlaubt keine Platzänderung (Quittungsstatus 0 wurde verwendet)
7	Ein unerlaubter Quittungsstatus wurde verwendet
andere Werte:	Die Zahl entspricht der Fehlermeldung der Werkzeugverwaltung in der NC, die durch diesen Transfer verursacht wurde.

Der Fehlerstatus wird mit dem Quittieren des Fehlers auf Null zurückgesetzt.

Abbild der Quittungen (DB41xx.DBBy108 bis DBBy111)

Die vom PLC-Programm zuletzt gesetzten Quittungen (DB40xx.DBB0 bis DBB3) werden hier von der PLC-Firmware zusammen mit den Bits "Quittung OK" oder "Quittung fehlerhaft" gesetzt und rückgesetzt. Diese statisch anstehenden Bits dienen im Fehlerfall zur Diagnose, welcher Quittungsschritt den Fehler ausgelöst hat. Werden im PLC-Programm mehrere Quittungsbits gesetzt, werden diese auch in das Abbild eingetragen.

Auftragszustand

DB4100...41xx	Signale von Werkzeugverwaltung [r]							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWy120	reserviert							
DBWy122	reserviert							
DBWy124	aktuelle Magazinnummer des Werkzeugs (INT)							
DBWy126	aktuelle Platznummer des Werkzeugs (INT)							
DBWy128	Ziel-Magazinnummer des Werkzeugs (INT)							
DBWy130	Ziel-Platznummer des Werkzeugs (INT)							
DBWy132	reserviert							
DBWy134	reserviert							
DBWy136	reserviert							
DBWy138	reserviert							
DBWy140	Multitool-Typ (INT)							
DBWy142	Multitool-Platzanzahl (INT)							
DBWy144	Multitool-Werkzeugabstand (FLOAT)							
DBWy146								
DBWy148	Multitool-Platznummer (INT)							

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Beladestelle

5.2 Werkzeugwechsel

Nahtstellenbeschreibung

Pro Werkzeughalter/Spindel gibt es je eine Schnittstelle für:

- Aufträge zur Vorbereitung und Durchführung des Werkzeugwechsels.
  Meldung an das PLC-Programm: Auftrag aktiv; Auftragsspezifikation und Auftragsbeschreibung.
  Die Aufträge erscheinen in der Schnittstelle des Werkzeughalters (der Spindel), woher ein Werkzeug eingewechselt oder aus dem ein Werkzeug ausgewechselt werden soll.
- Quittungen des PLC-Programms
  Alle Quittungen zu einem Auftrag müssen in der Schnittstelle des gleichen Werkzeughalters (der gleichen Spindel) erfolgen. Quittungsfehler sind ebenfalls in dieser Schnittstelle zurückzusetzen.

- Rückmeldung der Werkzeugverwaltung an das PLC-Programm
  Meldung an das PLC-Programm: Zustand der Quittung, Fehlerstatus, Abbild der Quittungsbits.
- Auftragszustand
  Ausgewählte Daten der letzten Zwischen- oder Ende-Quittung werden gespeichert. Diese Daten werden von der PLC-Firmware für die nächste Quittung an die Werkzeugverwaltung benötigt und sind zu Diagnosezwecken lesbar. Vom PLC-Programm können diese Daten für das Wiederaufsetzen nach einem Abbruch (z. B. Reset während eines Werkzeugwechsels) genutzt werden.

Ende-Quittung für Werkzeugwechsel

Für "Werkzeugwechsel vorbereiten" und "Werkzeugwechsel durchführen" kann es eine gemeinsame (Werkzeugwechsel durch T-Befehl bei Revolver) oder getrennte (Txx und M206 in verschiedenen Sätzen, Voreinstellung Fräsen) Ende-Quittungen geben. Die Ende-Quittung auf den Auftrag zum Werkzeugwechsel vorbereiten, lässt bei entsprechender MD-Einstellung den Vorlauf der NC weiterlaufen.

Die Maschinendaten mit denen das Verhalten von Satzvorlauf, Hauptlauf und das unterschiedliche Quittungsverhalten sind in Kapitel Maschinendaten für die Werkzeugverwaltung (Seite 27) beschrieben.

Mit der Ende-Quittung auf den Auftrag "Werkzeugwechsel durchführen" kann der Hauptlauf der NC fortgesetzt werden. Diese Ende-Quittungen sollten deshalb so früh wie möglich erfolgen. Das kann bedeuten, dass die Ende-Quittung erfolgt, bevor das alte Werkzeug im Magazin ist (z. B. das neue Werkzeug befindet sich in der Spindel, das alte Werkzeug im Toolboy). Die restlichen Schritte des alte Werkzeugs bis ins Magazin müssen dann asynchron mitgeteilt werden. Dazu ist die gleiche Schnittstelle wie für die synchronen Quittungen zu verwenden.

Nahtstellensignal	Bedeutung
DB42xx.DBXy000.0 – DBXy003.6	PLC-Programm: Quittungen zum Werkzeugwechsel vorbereiten und durchführen
DB42xx.DBXy009.0	PLC-Programm: Rücksetzen der Meldung "Quittung fehlerhaft" (DB43xx.DBX100.1) und der Diagnose-Informationen in der Rückmeldungsschnittstelle
DB43xx.DBXy000.0	Werkzeugverwaltung: Auftrag zum "Werkzeugwechsel vorbereiten" und "Werkzeugwechsel durchführen"
DB43xx.DBBy001	Werkzeugverwaltung Auftragsspezifikation
DB43xx.DBWy006 – DBWy034	Auftragsbeschreibung
DB43xx.DBXy100.0 DB43xx.DBXy100.1	Positive Rückmeldung Quittungszustand, Quittung OK, 1 PLC-Zyklus anstehend Negative Rückmeldung Quittungszustand, Quittung fehlerhaft, statisch anstehend
DB43xx.DBXy100.0 DB43xx.DBXy100.1	Positive Rückmeldung Quittungszustand, 1 PLC-Zyklus anstehend Negative Rückmeldung Quittungszustand, statisch anstehend.
DB43xx.DBXy100.0	Werkzeugverwaltung Rückmeldung Quittungszustand
DB43xx.DBBy104	Werkzeugverwaltung Rückmeldung Fehlerstatus

Nahtstellensignal	Bedeutung
DB43xx.DBXy108.0 - DB43xx.DBXy111.6	Abbild der Quittungen zum Werkzeugwechsel: Dieses Abbild gehört zur positiven oder negativen Rückmeldung und ist ebenso lange gültig.
DB43xx.DBWy124 – DBWy138	Auftragszustand

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Index der Spindel / Werkzeughalter

Aufträge: Schnittstellenbeschreibung

DB4300...43xx	Signale von Werkzeugverwaltung [r]							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy000	Multitool: Daten im Multitool-Bereich	--	--	--	--	--	--	Auftrag ist noch aktiv.
DBBy001	Werkzeug verbleibt in der Spindel	Handwerkzeug auswechseln	Handwerkzeug einwechseln	kein altes Werkzeug	T0	Wechsel vorbereiten	Wechsel durchführen (Anstoß: M06)	festplatzcodiert
DBBy002	reserviert							
DBBy003	reserviert							
DBBy004	reserviert							
DBBy005	reserviert							
DBWy006	Quell-Magazinnummer für neues Werkzeug (INT)							
DBWy008	Quell-Platznummer für neues Werkzeug (INT)							
DBWy010	reserviert							
DBWy012	reserviert							
DBWy014	reserviert							
DBWy016	reserviert							
DBWy018	Ziel-Magazinnummer für altes Werkzeug (INT)							
DBWy020	Ziel-Platznummer für altes Werkzeug (INT)							
DBWy022	Platztyp (INT)							
DBWy024	Größe links (INT)							
DBWy026	Größe rechts (INT)							
DBWy028	reserviert							
DBWy030	reserviert							
DBBy032	Werkzeugstatus für neues Werkzeug:							
	Werkzeug war im Einsatz	Werkzeug festplatzcodiert	Werkzeug im Wechsel	Vorwarngrenze erreicht	Werkzeug vermessen	Werkzeug gesperrt	Werkzeug freigegeben	aktives Werkzeug
DBBy033	Werkzeugstatus für neues Werkzeug:							
	Handwerkzeug	1:1 Wechsel	--	Stammwerkzeug	zu beladen	zu entladen	gesperrt	Kennung für Werkzeuge im Zwischenspeicher

DB4300...43xx	Signale von Werkzeugverwaltung [r]
DBWy034	Neues Werkzeug: Interne T-Nummer der NC (INT)
DBWy036	reserviert
DBWy038	reserviert
DBWy040	reserviert
DBWy042	reserviert
DBWy044	Freiparameter 1 (DWORD)
DBWy046	
DBWy048	Freiparameter 2 (DWORD)
DBWy050	
DBWy052	Freiparameter 3 (DWORD)
DBWy054	
DBWy056	Multitool: Typ (INT)
DBWy058	Multitool: Platzanzahl (INT)
DBWy060	Multitool: Werkzeugabstand (FLOAT)
DBWy062	
DBWy064	Multitool: Nummer (Quelle) neues Werkzeug (INT)
DBWy066	Multitool: Platznummer (Quelle) neues Werkzeug (INT)
DBWy068	Multitool: Nummer (Ziel) altes Werkzeug (INT)
DBWy070	Multitool: Platznummer (Ziel) altes Werkzeug (INT)
DBWy072	Platztyp (INT)
DBWy074	Größe links (INT)
DBWy076	Größe rechts (INT)
DBWy078	Größe oben (INT)
DBWy080	Größe unten (INT)
DBWy082	Werkzeugstatus für neues Werkzeug (INT)
DBWy084	T-Nummer des Werkzeugs (INT)
DBWy086	Spindelnummer / Werkzeughalternummer (INT)
DBWy088	Multitool: ursprüngliche Magazinnummer (INT)
DBWy090	Multitool: ursprüngliche Platznummer (INT)

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Werkzeughalter

Aufträge: Signalbeschreibung

Signalbeschreibung:

- Auftrag:
  Die Schnittstelle enthält einen Auftrag. Die Auftragsbearbeitung wurde noch nicht mit einer Ende-Quittung abgeschlossen. Nach Übertragung der Ende-Quittung an die Werkzeugverwaltung wird dieses Signal rückgesetzt.
- Multitool: Daten im Multitool-Bereich
  Die Schnittstelle enthält Informationen zum im Auftrag angegebenen Multitool..
- Festplatz-codiert: Das neue Werkzeug ist festplatz-codiert.

- Werkzeugwechsel durchführen:
  Das neue Werkzeug ist in den Werkzeughalter / die Spindel einzuwechseln. Das alte Werkzeug ist auf einen Magazinplatz zurückzubringen. Dieser Auftrag erfordert immer eine Ende-Quittung.
- Werkzeugwechsel vorbereiten:
  Neues Werkzeug bereitstellen. Magazinplatz evtl. für altes Werkzeug an die Wechselstelle positionieren. Dieser Auftrag erfordert einzeln eine Ende-Quittung, bei gleichzeitigem Auftrag "Wechsel durchführen" entfällt die Ende-Quittung auf die Vorbereitung.
- T0: T0 wurde programmiert (Werkzeughalter/Spindel leer fahren).
- kein altes Werkzeug:
  Werkzeugwechsel in den bisher leeren Werkzeughalter/Spindel.
- Handwerkzeug einwechseln:
  Ein Handwerkzeug ist einzuwechseln. Welches Werkzeug eingewechselt werden soll, wird am SINUMERIK Operate angezeigt.
- Handwerkzeug auswechseln:
  Das Werkzeug ist über Handbedienung auszuwechseln.
- Werkzeug verbleibt in Spindel:
  Das Bit wird gesetzt bei einem Wechsel von Werkzeughalter → Spindel nach Werkzeughalter → Spindel. Auslöser kann z. B. Reset- Startmode oder Satzsuchlauf sein.
- Quellplatz neues Werkzeug:
  Magazin- und Platznummer, woher das neue Werkzeug kommt (meist ein Platz in einem realen Magazin).
- Zielplatz altes Werkzeug:
  Magazin- und Platznummer, wohin das alte Werkzeug zu transportieren ist (meist ein Platz in einem realen Magazin).
- Ursprung neues Werkzeug:
  - Interne T-Nummer: Interne T-Nummer des neuen Werkzeugs
  - Werkzeugstatus: Werkzeugstatus des neuen Werkzeugs
  - Platztyp: Platztyp des neuen Werkzeugs
  - Größe: Größe (rechts, links) des neuen Werkzeugs
- Freie Parameter: 3 freie Parameter, die vom Teileprogramm an das PLC-Programm übergegeben werden.
- Mutlitool: Typ:
  Art der Abstandskodierung: 1 = Platznummer, 2 = Länge, 3 = Winkel
- Multitool: Platzanzahl
  Anzahl der im Multitool vorhandenen Werkzeugplätze
- Multitool: Werkzeugabstand
  Wert für die Abstandskodierung: Platznummer des Werkzeugs im Multitool oder Winkel oder Abstand des Werkzeugs innerhalb des Multitools
- Multitool: Nummer (Quelle) / Multitool: Platznummer (Quelle) neues Werkzeug
  Magazinnummer / Platznummer, von der das neue Multitool kommt (meist ein Platz in einem realen Magazin).

- Multitool: Nummer (Ziel) / Multitool: Platznummer (Ziel) altes Werkzeug Magazinnummer / Platznummer, wohin das alte Multitool zu transportieren ist (meist ein Platz in einem realen Magazin).
- Platztyp: Platztyp des neuen Werkzeugs
- Größe links Platzbedarf für das Multitool im Magazin nach links
- Größe rechts: Platzbedarf für das Multitool im Magazin nach rechts
- Größe oben: Platzbedarf für das Multitool im Magazin nach oben
- Größe unten: Platzbedarf für das Multitool im Magazin nach oben
- Werkzeugstatus
- T-Nummer des Werkzeugs
- Spindelnummer / Werkzeughalternummer
- Multitool: ursprüngliche Magazinnummer
- Multitool: ursprüngliche Platznummer

Quittungen: Schnittstellenbeschreibung

DB4200...42xx	**Signale an Werkzeugverwaltung [r/w]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy000	Quittungs-schritt 7	Quittungs-schritt 6	Quittungs-schritt 5	Quittungs-schritt 4	Quit-tungs-schritt 3	Quittungs-schritt 2	Quittungs-schritt 1	Total-Quit-tung
DBBy001	Quittungs-schritt 15	Quittungs-schritt 14	Quittungs-schritt 13	Quittungs-schritt 12	Quit-tungs-schritt 11	Quittungs-schritt 10	Quittungs-schritt 9	Quittungs-schritt 8
DBBy002	Quittungs-schritt 23	Quittungs-schritt 22	Quittungs-schritt 21	Quittungs-schritt 20	Quit-tungs-schritt 19	Quittungs-schritt 18	Quittungs-schritt 17	Quittungs-schritt 16
DBBy003	reserviert	Quittungs-schritt 30	Quittungs-schritt 29	Quittungs-schritt 28	Quit-tungs-schritt 27	Quittungs-schritt 26	Quittungs-schritt 25	Quittungs-schritt 24
DBWy004	Multitool: Platznummer (INT)							
DBBy006	reserviert							
DBBy007	reserviert							
DBBy008	reserviert							
DBBy009							Auftrags-quittung	Rücksetzen Quittungs-fehler
DBWy010 • • •	reserviert							
DBWy026	reserviert							
DBBy028	Quittungs-schritt 38	Quittungs-schritt 37	Quittungs-schritt 36	Quittungs-schritt 35	Quit-tungs-schritt 34	Quittungs-schritt 33	Quittungs-schritt 32	Quittungs-schritt 31

DB4200...42xx	**Signale an Werkzeugverwaltung [r/w]**							
DBBy029	Quittungs-schritt 46	Quittungs-schritt 45	Quittungs-schritt 44	Quittungs-schritt 43	Quit-tungs-schritt 42	Quittungs-schritt 41	Quittungs-schritt 40	Quittungs-schritt 39
DBBy030	Quittungs-schritt 54	Quittungs-schritt 53	Quittungs-schritt 52	Quittungs-schritt 51	Quit-tungs-schritt 50	Quittungs-schritt 49	Quittungs-schritt 48	Quittungs-schritt 47
DBBy031	reserviert	reserviert	Quittungs-schritt 60	Quittungs-schritt 59	Quit-tungs-schritt 58	Quittungs-schritt 57	Quittungs-schritt 56	Quittungs-schritt 55

xx: Beladestelle

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

Quittungen: Signalbeschreibung

Signalbeschreibung:

- Total-Quittung:
  Bei einer 0/1-Flanke wird zum aktuellen Auftrag die Ende-Quittung mit Status 99 gesendet (Auftrag komplett erledigt, alle Ziel-Positionen sind erreicht). Solange das Signal ansteht, dürfen keine Änderungen an den Daten dieser Schnittstelle vorgenommen werden!
  Nach der Übertragung der Quittung an die Werkzeugverwaltung wird dieses Signal von der PLC-Firmware zurückgesetzt.
- Quittungsschritt 1...60:
  Bei einer 0/1-Flanke wird der zugehörige Quittungsschritt aus der Quittungsschritt-Tabelle an die Werkzeugverwaltung gesendet. Solange das Signal ansteht, dürfen keine Änderungen an den Daten dieser Schnittstelle und der variablen Transferschritt-Tabelle vorgenommen werden!
  Nach der Übertragung der Quittung an die Werkzeugverwaltung wird dieses Signal von der PLC-Firmware zurückgesetzt.
- Multitool: Platznummer
  Das PLC-Programm meldet die aktive Platznummer an die Werkzeugverwaltung der NC zurück.
- Rücksetzen Quittungsfehler:
  Rücksetzen der Meldung Quittung fehlerhaft (DB43xx.DBX100.1) und der Diagnose-Informationen in der Rückmeldeschnittstelle.
- Auftragsquittung:
  Dieses Bit wirkt nur bei Totalquittung DB4200.DBX0.0 = 1
  DB4200.DBX9.1 = 0 ⇒ Auftragsquittung mit Status 99
  DB4200.DBX9.1 = 1 ⇒ Auftragsquittung mit Status 3

Rückmeldungen: Schnittstellenbeschreibung

DB4300...43xx	**Signale von Werkzeugverwaltung [r]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBy100							Quittung fehlerhaft	Quittung OK
DBBy101	reserviert							

DB4300...43xx	**Signale von Werkzeugverwaltung [r]**							
DBWy102	reserviert							
DBWy104	Fehlerstatus (WORD)							
DBWy106	reserviert							
DBBy108	Quittungs-schritt 7	Quittungs-schritt 6	Quittungs-schritt 5	Quittungs-schritt 4	Quittungs-schritt 3	Quittungs-schritt 2	Quittungs-schritt 1	Quittungs-schritt 0
DBBy109	Quittungs-schritt 15	Quittungs-schritt 14	Quittungs-schritt 13	Quittungs-schritt 12	Quittungs-schritt 11	Quittungs-schritt 10	Quittungs-schritt 9	Quittungs-schritt 8
DBBy110	Quittungs-schritt 23	Quittungs-schritt 22	Quittungs-schritt 21	Quittungs-schritt 20	Quittungs-schritt 19	Quittungs-schritt 18	Quittungs-schritt 17	Quittungs-schritt 16
DBBy111	reserviert	Quittungs-schritt 30	Quittungs-schritt 29	Quittungs-schritt 28	Quittungs-schritt 27	Quittungs-schritt 26	Quittungs-schritt 25	Quittungs-schritt 24
DBBy112	Multitool: Platznummer (INT)							
• • •								
DBBy152	Quittungs-schritt 38	Quittungs-schritt 37	Quittungs-schritt 36	Quittungs-schritt 35	Quittungs-schritt 34	Quittungs-schritt 33	Quittungs-schritt 32	Quittungs-schritt 31
DBBy153	Quittungs-schritt 46	Quittungs-schritt 45	Quittungs-schritt 44	Quittungs-schritt 43	Quittungs-schritt 42	Quittungs-schritt 41	Quittungs-schritt 40	Quittungs-schritt 39
DBBy154	Quittungs-schritt 54	Quittungs-schritt 53	Quittungs-schritt 52	Quittungs-schritt 51	Quittungs-schritt 50	Quittungs-schritt 49	Quittungs-schritt 48	Quittungs-schritt 47
DBBy155	reserviert	reserviert	Quittungs-schritt 60	Quittungs-schritt 59	Quittungs-schritt 58	Quittungs-schritt 57	Quittungs-schritt 56	Quittungs-schritt 55

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Werkzeughalter

Rückmeldungen: Signalbeschreibung

Signalbeschreibung:

- **Quittung OK (DB43xx.DBX100.0): Positive Rückmeldung Quittungszustand.**
  Die Quittung des PLC-Programm (Bereich DB42xx.DBB0 bis DBB3) wurde fehlerfrei an die Werkzeugverwaltung übertragen. Dieses Signal wird nach einem PLC-Takt rückgesetzt.
- **Quittung fehlerhaft (DB43xx.DBX100.1): Negative Rückmeldung Quittungszustand.**
  Die Quittung des PLC-Programm (Bereich DB42xx.DBB0 bis DBB3) ist fehlerhaft. Die Fehlerursache wird in "Fehlerstatus" mitgeteilt.
  Das Bit "Quittung fehlerhaft" wird auch gesetzt wenn die Quittung von der PLC-Firmware als fehlerfrei akzeptiert wurde und die Werkzeugverwaltung einen Fehler im quittierten Werkzeugtransfer erkennt und diesen meldet (zum Beispiel wenn der Zielplatz des Werkzeugtransfers belegt ist).
  Fehler, die direkt von der Werkzeugverwaltung in der NC erkannt werden, ohne dass vom PLC-Programm eine Quittung gegeben wurde, führen nicht zum Setzen des Bits 100.1.
  Liegt ein Fehler vor, der die Weitergabe der Quittung an die Werkzeugverwaltung verhindert (Fehlerstatus 1 bis 7), wird der Fehler nur an der Schnittstelle der Werkzeugverwaltung und nicht von der NC ausgegeben (kein NC Alarm!).
  Bei Bedarf meldet der Maschinenhersteller diese Fehler, in dem das PLC-Programm einen PLC-Alarm auslöst.
  Dieses Signal bleibt bis zum Quittieren des Fehlers statisch anstehen (Setzen des Bits "Rücksetzen Quittungsfehler" DB4200.DBX9.0 durch den Hersteller). Die NC-/PLC-Schnittstelle in DB42xx.DBB0 bis DBB3 ist bei anstehendem Bit "Quittung fehlerhaft" gesperrt. Eingehende Quittungsbits werden von der PLC-Firmware nicht ausgewertet und ebenfalls mit dem Setzen des Bits "Rücksetzen Quittungsfehler" gelöscht.
- **Multitool: Platznummer**
  Multitool: Platznummer, die das PLC-Programm als aktive Platznummer beim Quittieren in die Quittungsschnittstelle eingetragen hat.

Fehlerstatus

Bei einem Fehler steht in DB43xx.DBB104 eine Diagnosenummer ungleich Null mit folgender Bedeutung:

Diagnosenummer	Bedeutung
0	kein Fehler
1	mehrere Quittungssignale gleichzeitig
2	Quittung ohne Auftrag
3	ungültige Transferschritt-Nummer
4	für eine Positionsangabe fehlt der Auftrag
5	Status erlaubt keine Platzänderung (Quittungsstatus 0 wurde verwendet)
7	Ein unerlaubter Quittungsstatus wurde verwendet
andere Werte:	Die Zahl entspricht der Fehlermeldung der Werkzeugverwaltung in der NC, die durch diesen Transfer verursacht wurde.

Der Fehlerstatus wird mit dem Quittieren des Fehlers zurückgesetzt.

Abbild der Quittungen (DB43xx.DBB108 bis DBB111)

Die vom PLC-Programm zuletzt gesetzten Quittungen (DB42xx.DBB0 bis DBB3) werden hier von der PLC-Firmware zusammen mit den Bits "Quittung OK" oder "Quittung fehlerhaft" gesetzt und rückgesetzt. Diese statisch anstehenden Bits dienen im Fehlerfall zur Diagnose, welcher Quittungsschritt den Fehler ausgelöst hat. Wurden im PLC-Programm mehrere Quittungsbits gesetzt, werden auch diese in das Abbild eingetragen.

Auftragszustand

DB4300 ... 43xx	Signale von Werkzeugverwaltung [r]							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWy120	reserviert							
DBWy122	reserviert							
DBWy124	aktuelle Magazinnummer für neues Werkzeug (INT)							
DBWy126	aktuelle Platznummer für neues Werkzeug (INT)							
DBWy128	Ziel-Magazinnummer für neues Werkzeug (INT)							
DBWy130	Ziel-Platznummer für neues Werkzeug (INT)							
DBWy132	aktuelle Magazinnummer für altes Werkzeug (INT)							
DBWy134	aktuelle Platznummer für altes Werkzeug (INT)							
DBWy136	Ziel-Magazinnummer für altes Werkzeug (INT)							
DBWy138	Ziel-Platznummer für altes Werkzeug (INT)							
DBWy140	Multitool: Typ (INT)							
DBWy142	Multitool: Platzanzahl (INT)							
DBWy144	Multitool: Werkzeugabstand (FLOAT)							
DBWy146								
DBWy148	Multitool: Platznummer (INT)							

y [= 0 oder 1]: TO-Einheit 1 oder TO-Einheit 2

xx: Werkzeughalter

5.3 Transferschritt- und Quittungsschritt-Tabellen

Projektierbare Schritt-Tabellen

In den Datenbausteinen TM_CTS (DB9900), TM_VTS (DB9901) und TM_ACK (DB9902) stehen projektierbare Tabellen zur Verfügung, mit denen die Abläufe der Werkzeugbewegungen beschrieben werden.

Nahtstellensignal	Name	Bedeutung
DB9900	TM_CTS	konstante Transferschritt-Tabelle (projektierbar)
DB9901	TM_VTS	variable Transferschritt-Tabelle (projektierbar und von PLC-Programm beschreibbar)
DB9902	TM_ACK	Quittungsschritt-Tabelle (projektierbar)

Die Datenbausteine DB40xx, 41xx, 42xx und 43xx sind Systembausteine und werden automatisch von der Steuerung angelegt.

Die Datenbausteine DB9900, DB9901 und DB9902 werden vom Programming Tool unter Bibliotheken/Spezielle Datenbausteine zur Verfügung gestellt. Die Bausteine sind noch nicht mit den notwendigen Daten gefüllt und müssen in das PLC-Projekt kopiert und bearbeitet werden.

Transferschritt-Tabellen

Die einzelnen Werkzeugbewegungen werden als Transferschritte Werkzeug von Magazinplatz x/y nach Magazinplatz m/n definiert. Mit diesen Transferschritten können Quittungsschritte definiert werden. DB9900 enthält fest projektierte Transferschritte (konstante Transferschritt-Tabelle). DB9901 kann vom PLC-Programm verändert werden, zum Beispiel für die Quittung von Zwischenschritten wie Magazinpositionen bei der Werkzeugwechselvorbereitung (variable Transferschritt-Tabelle).

DB9900	**Konstante Transferschritt-Tabelle [r]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW0	Transferschritt 1 Quell-Magazinnummer (INT)							
DBW2	Transferschritt 1 Quell-Platznummer (INT)							
DBW4	Transferschritt 1 Ziel-Magazinnummer (INT)							
DBW6	Transferschritt 1 Ziel-Platznummer (INT)							
DBW8	Transferschritt 2 Quell-Magazinnummer (INT)							
DBW10	Transferschritt 2 Quell-Platznummer (INT)							
DBW12	Transferschritt 2 Ziel-Magazinnummer (INT)							
DBW14	Transferschritt 2 Ziel-Platznummer (INT)							
	...							
DBW504	Transferschritt 64 Quell-Magazinnummer (INT)							
DBW506	Transferschritt 64 Quell-Platznummer (INT)							
DBW508	Transferschritt 64 Ziel-Magazinnummer (INT)							
DBW510	Transferschritt 64 Ziel-Platznummer (INT)							

DB9901	**Variable Transferschritt-Tabelle [rw]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW0	Transferschritt 101 Quell-Magazinnummer (INT)							
DBW2	Transferschritt 101 Quell-Platznummer (INT)							
DBW4	Transferschritt 101 Ziel-Magazinnummer (INT)							
DBW6	Transferschritt 101 Ziel-Platznummer (INT)							
DBB 8	Transferschritt 102 Quell-Magazinnummer (INT)							
DBW10	Transferschritt 102 Quell-Platznummer (INT)							
DBW12	Transferschritt 102 Ziel-Magazinnummer (INT)							
DBW14	Transferschritt 102 Ziel-Platznummer (INT)							
	...							
DBW504	Transferschritt 164 Quell-Magazinnummer (INT)							
DBW506	Transferschritt 164 Quell-Platznummer (INT)							

DB9901	**Variable Transferschritt-Tabelle [rw]**
DBW508	Transferschritt 164 Ziel-Magazinnummer (INT)
DBW510	Transferschritt 164 Ziel-Platznummer (INT)

Quittungsschritt-Tabelle

Jeder Eintrag indiziert zwei Transferschritte (für Neu- und Altwerkzeug) und gibt den erreichten zugehörigen Status an. Die Quittungsschritt-Tabelle in DB9902 wird gemeinsam für Quittungen auf der Nahtstelle der Beladestelle als auch auf der Nahtstelle des Werkzeughalters verwendet.

DB9902	**Quittungsschritt-Tabelle [r]**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Quittungsschritt 1 Transferschritt für neues Werkzeug (BYTE)							
DBB1	Quittungsschritt 1 Transferschritt für altes Werkzeug (BYTE)							
DBB2	Quittungsschritt 1 Quittungsstatus (BYTE)							
DBB3	Quittungsschritt 1 reserviert							
DBB4	Quittungsschritt 2 Transferschritt für neues Werkzeug (BYTE)							
DBB5	Quittungsschritt 2 Transferschritt für altes Werkzeug (BYTE)							
DBB6	Quittungsschritt 2 Quittungsstatus (BYTE)							
DBB7	Quittungsschritt 2 reserviert							
	...							
DBB236	Quittungsschritt 60 Transferschritt für neues Werkzeug (BYTE)							
DBB237	Quittungsschritt 60 Transferschritt für altes Werkzeug (BYTE)							
DBB238	Quittungsschritt 60 Quittungsstatus (BYTE)							
DBB239	Quittungsschritt 60 reserviert							

Siehe auch

PLC Programmbausteine (Seite 53)

PLC Programmbausteine 6

6.1 Quittungsprozess

Informationen an die Werkzeugverwaltung

Die Werkzeugverwaltung erwartet Quittungen auf ihre Aufträge, um die realen Werkzeugpositionen stets zu verfolgen und mitzuführen. Pro Auftrag ist mindestens eine Quittung erforderlich. Für viele Anwendungen ist dies ausreichend.

Die Quittung erfolgt entweder über die in DB9902 definierte Tabelle oder nach vollständig abgeschlossenem Auftrag der Werkzeugverwaltung in einem Schritt mit der Total-Quittung (DB40xx/42xx DBX0.0) durch eine 0/1-Flanke (setzen) des entsprechenden Bits in der Anwender-Nahtstelle.

Solange das Quittungssignal ansteht, dürfen keine Änderungen an den Daten dieser Schnittstelle vorgenommen werden! Nach der Übertragung der Quittung an die Werkzeugverwaltung wird dieses Signal von der PLC-Firmware zurückgesetzt. Das Rücksetzen kann unter Umständen nach mehreren PLC-Takten erfolgen.

Wird die Werkzeugverwaltung auch über Zwischenpositionen informiert, ergeben sich weitere Vorteile:

- **Information über Zwischenpositionen:**
  Sind der Werkzeugverwaltung alle Zwischenpositionen von Werkzeugen bekannt, kann die Belegung des Zwischenspeichermagazins auch rückgefragt werden. Dies erleichtert die Hochlaufstrategie nach dem Aus- und Einschalten oder nach einem Kommando-Abbruch (z. B. durch Reset). Wird ein Werkzeug, das gerade ausgewechselt wird, sofort wieder benötigt, kann es ohne erst im Magazin abgelegt zu werden schon von einem Zwischenspeicherplatz wieder in die Spindel eingewechselt werden.
- **Information über Magazinpositionen:**
  Ist der Werkzeugverwaltung bekannt, welcher Magazinplatz sich an einer Übergabestelle (Wechselstelle zur Spindel, Beladestelle) befindet, kann sie für die Leerplatzsuche oder bei der Wahl des neuen Werkzeugs die kürzesten Wege im Magazin ermitteln. Bei Aufträgen kann die Werkzeugverwaltung die Magazinposition meist aus Zwischenquittungen (z. B. Werkzeug-Transfer zwischen realem Magazin und Zwischenspeicher) oder der Ende-Quittung (z. B. Auftrag "Magazin positionieren" abgeschlossen) erkennen. Erfolgt die Positionierung des Magazins durch das PLC-Programm selbst (z. B. durch Maschinentasten) ohne Auftrag von der Werkzeugverwaltung, muss dies der Werkzeugverwaltung über asynchrone Meldungen mitgeteilt werden.

6.2 Quittungsarten

Werkzeug- und Magazinbewegungen

Die Werkzeugverwaltung unterscheidet zwischen synchroner Quittung und asynchroner auftragsunabhängiger Mitteilung.

Synchrone Quittung

- Quittung von Zwischenschritten eines Auftrags (Werkzeugverwaltung registriert aktuelle Positionsänderungen von Werkzeugen, Teileprogramm muss noch warten)
  Mit einer Zwischenquittung wird die Werkzeugverwaltung über Zwischenschritte zu einem Auftrag informiert. Bei Zwischenquittungen interessiert nur die Zielposition des Zwischenschrittes. Die Quellposition ist aus dem Auftrag oder der letzten Zwischenquittung bekannt. Beim Werkzeugwechsel können auch zwei Werkzeuge (neues und altes) gleichzeitig quittiert werden. Zwischenquittungen sind nur **vor** der Ende-Quittung möglich.
- Ende-Quittung zu einem Auftrag (Teileprogramm kann fortgesetzt werden)
  Zu jedem Auftrag ist eine Ende-Quittung erforderlich. Die Ende-Quittung erlaubt die Fortsetzung des Teileprogramms und gibt die Auftragsschnittstelle für neue Aufträge frei. Sie sollte so früh wie möglich erfolgen (z. B. sobald sich das neue Werkzeug in der Spindel befindet und keine Kollision mehr zu erwarten ist). Weitere Schritte nach der Ende-Quittung zu einem Auftrag (z. B. den Rückweg des alten Werkzeugs ins Magazin) können der Werkzeugverwaltung asynchron mitgeteilt werden.

Asynchrone auftragsunabhängige Mitteilung

Positionsänderung eines Werkzeugs oder Magazins ("asynchrone Mitteilung", z. B. wenn die PLC eine Werkzeugposition durch Maschinensteuertafel-Bedienung ohne Werkzeugwechselauftrag verändert.)

Mit einer asynchronen Mitteilung kann die Werkzeugverwaltung unabhängig von einem Auftrag über eine Werkzeug- oder Magazinbewegung informiert werden. Eine asynchrone Mitteilung muss immer Quellposition (von) **und** Zielposition (nach) enthalten.

Werkzeugbewegungen innerhalb eines Magazins (Werkzeug umsetzen) können nur mit tatsächlich belegten Magazinplätzen durchgeführt werden. Leertransfers sind nicht zulässig. In einer Mitteilung können zwei asynchrone Transfers realisiert werden. In diesem Fall muss die Nahtstelle für Werkzeugwechsel DB42xx benutzt werden.

Wirkung der Quittungen

Wirkung der Quittungen auf den Auftrag und das Teileprogramm:

- Zwischenquittung und Ende-Quittung erfolgen synchron zum Auftrag.
  - Das Teileprogramm muss warten.
  - Es kann noch kein neuer Auftrag kommen.
- Mitteilung eines asynchronen Transfers:
  - Das Teileprogramm läuft weiter.
  - Die Mitteilung ist völlig unabhängig von irgendeinem Auftrag.

6.3 Quittungszustände

Quittungszustände und ihre Bedeutung:

Der Status, der die jeweilige Quittungsart kennzeichnet, ist in der nachfolgenden Tabelle angegeben:

Quittung		Bedeutung
Synchrone Ende-Quittung	**1**	Auftrag an angegebener Position beendet: Die Werkzeuge befinden sich auf den angegebenen Positionen. Das Teileprogramm kann fortgesetzt werden.
	3	Auftrag abgebrochen: Der Auftrag wird abgebrochen, bis dahin quittierte Positionsänderungen von Werkzeugen werden beibehalten. Das Abbruch-Kommando selbst löst keine Positionsquittungen oder -änderungen in der Werkzeugverwaltung aus.
	7	Auftrag "Werkzeugwechsel vorbereiten" wiederholen: Der Werkzeugwechsel wurde vorher eine neue Werkzeugposition mitgeteilt. Der Auftrag "Werkzeugwechsel vorbereiten" soll unter Berücksichtigung dieser Position neu berechnet werden. Nur zulässig bei noch nicht quittiertem Vorbereitungskommando.
	99	Total-Quittung: Auftrag beendet, alle Positionen erreicht. Alle beteiligten Werkzeuge befinden sich auf den im Auftrag vorgegebenen Positionen. Das Teileprogramm kann fortgesetzt werden. Alle Zielpositionen aus dem Auftrag sind erreicht.
Synchrone Zwischenquittung	**105**	Zwischenposition für Werkzeug: Die Werkzeuge wurden von der im Auftrag angegebenen Quellposition oder von der zuletzt quittierten Zwischenposition auf die angegebene Zielposition bewegt.
Asynchronen Transfer mitteilen	**201**	Werkzeugbewegung mitteilen: Das Werkzeug wurde von der Quellposition auf die angegebene Zielposition bewegt. Bei Bewegung von einem Platz im realen Magazin auf einen Zwischenspeicherplatz, wird der Quellplatz für das Werkzeug reserviert. Beachten Sie hierzu auch MD20310, Bit 24 (Seite 27).
	204	Magazinposition mitteilen: Der Magazinplatz befindet sich an der zum angegebenen Zielplatz gehörenden Wechsel-/Belade-/Entladestelle.

Übersicht über ausgewertete Tabellen-Parameter

Quittungsstatus		1	3	7	99	105	201	204
Neues Werkzeug	Transferschritt	x	-	-	-	x	x	x
	• von Magazin	-	-	-	-	-	xx	xx
	• von Platz	-	-	-	-	-	xx	xx
	• nach Magazin	xx	-	-	-	xx	xx	zz
	• nach Platz	xx	-	-	-	xx	xx	zz

Quittungsstatus		**1**	**3**	**7**	**99**	**105**	**201**	**204**
Altes Werkzeug	Transferschritt	x	-	-	-	x	x	-
	• von Magazin	-	-	-	-	-	xx	-
	• von Platz	-	-	-	-	-	xx	-
	• nach Magazin	xx	-	-	-	xx	xx	-
	• nach Platz	xx	-	-	-	xx	xx	-

Legende:

- Datum nicht relevant

x: Nummer (1...n) des Transferschrittes aus der Transferschritt-Tabelle

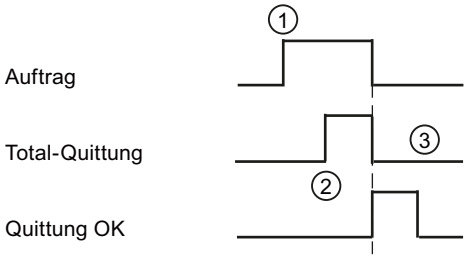
xx: Magazinnummer, Platznummer des Werkzeugs

zz: Magazinnummer, Platznummer der Belade-/Entlade- oder Wechselstelle

Die Statusübersicht gibt folgende Informationen aus:

- Die Status 1, 6, 105, 201 und 204 sind in der Quittungsschritt-Tabelle entsprechend ihrer Bedeutung zusammen mit Transferschritten zu sinnvollen Quittungsschritten zusammenzusetzen.
- Wird Status 1 mit beiden Transferschrittnummern = 0 kodiert, dient dieser Quittungsschritt als Ende-Quittung des bis zu diesem Zeitpunkt durch Zwischenquittungen erreichten Standes.
- Wurde ein Werkzeug aus einem realen Magazin in einen Zwischenspeicher bewegt (Umsetzen, MVTOOL), wird bei Quittung mit Status 6 oder Total-Quittung der Quellplatz für dieses Werkzeug reserviert ($TC_MPP4 Bit 1 und Bit 2). Das Verhalten ist gleich wie bei der Entnahme des Werkzeugs aus dem Magazin bei Werkzeugwechsel. Mit Status 1 erfolgt bei Umsetzen oder MVTOOL keine Reservierung des Quellplatzes.
- Die Status 3 und 7 brauchen in der Quittungsschritt-Tabelle nur einmal kodiert zu werden, da keine Transferschritte ausgewertet werden.
- Status 99 braucht nicht kodiert werden, er wird durch das Bit der "Total-Quittung" vorgegeben.

Typischer Ablauf eines beliebigen Auftrags mit Total-Quittung



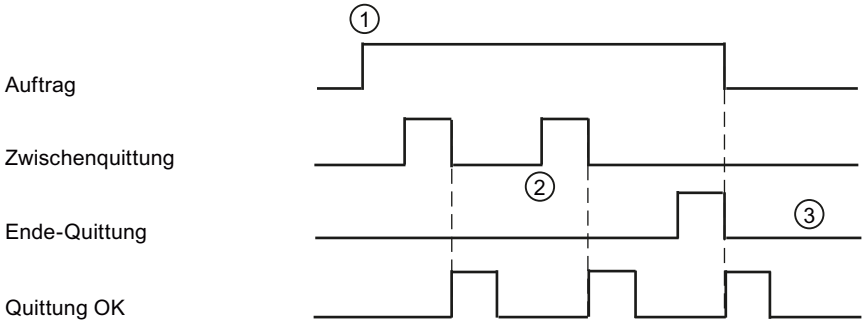
Signalbeschreibung:

① Das PLC-Programm erkennt an der 0/1 Flanke des Signals DB43xx.DBX0.0 (Auftrag), dass ein neuer Auftrag von der Werkzeugverwaltung übergeben wurde.

② Das PLC-Programm setzt das Quittungssignal in DB42xx.DBX0.0 (Total-Quittung). Mit Auslösen der 0/1 Flanke beginnt die PLC-Firmware mit der Übertragung der Quittung an die Werkzeugverwaltung.

③ Nach erfolgreicher Übertragung der Quittung an die Werkzeugverwaltung wird von der PLC-Firmware das Signal Quittung OK einen PLC Takt auf 1 gesetzt und gleichzeitig das Signal Auftrag sowie das Quittungsbit auf 0 zurückgesetzt.

Typischer Ablauf eines beliebigen Auftrags mit Zwischenquittung und Ende-Quittung



Signalbeschreibung:

① Das PLC-Programm erkennt an der 0/1 Flanke des Signals DB43xx.DBX0.0 (Auftrag), dass ein neuer Auftrag von der Werkzeugverwaltung übergeben wurde.

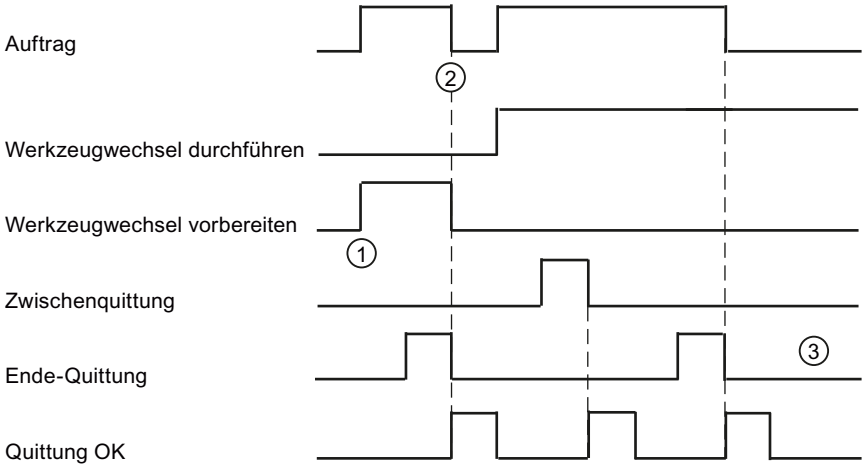
② Das PLC-Programm quittiert die in den DB9900, DB9901 und DB9902 projektierten Transferschritte mit Quittungsstatus 105. Die Positionen der Werkzeuge werden anhand der Transferschritte der Quittungen von der Werkzeugverwaltung aktualisiert.

③ Das PLC-Programm quittiert mit Quittungsstatus 1 die Ausführung des Auftrags. Nach erfolgreicher Übertragung der Quittung an die WZV wird von der PLC-Firmware das Signal Quittung OK einen PLC-Takt auf 1 gesetzt und gleichzeitig das Signal Auftrag sowie das Quittungsbit auf 0 zurückgesetzt.

Ablauf der Werkzeugverwaltung mit Satz-Splitting (Maschinendaten-Einstellung Fräsen)

```
Txx M6 ;
```

Mit M6 wird das Programm L6 aufgerufen (Grundeinstellung)



Signalbeschreibung:

① Das PLC-Programm bekommt einen neuen Auftrag. Der Auftrag "Werkzeugwechsel vorbereiten" und der Auftrag "Werkzeugwechsel durchführen" kommen nacheinander. Txx und M206 wurden in getrennten NC Sätzen programmiert. In der Nahtstelle in DB43xx steht nur der Auftrag "Werkzeugwechsel vorbereiten". Der Auftrag "Werkzeugwechsel durchführen" wird erst nach Ende-Quittung des Auftrags für das Vorbereiten ausgegeben.

② Mit Quittung des Auftrags "Werkzeugwechsel vorbereiten" wird das Bit DB43xx.DBX0.0 (Auftrag) zurückgesetzt. Falls der Wechselbefehl (M206) bereits den Hauptlauf der NC durchlaufen hat, wird anschließend sofort der neue Auftrag an der Nahtstelle ausgegeben.

③ Der Auftrag "Werkzeugwechsel durchführen" wird als normaler Auftrag quittiert. Es wird die Ende-Quittung mit OK zurück gemeldet und gleichzeitig das Bit für den Auftrag zurückgesetzt.

Die Beschreibung des Auftrags ("Werkzeugwechsel durchführen" und "Werkzeugwechsel vorbereiten") wird nicht zurückgesetzt. Byte 1 von DB43xx wird erst mit dem nächsten Auftrag neu geschrieben.

Hinweis

MD20270, MD20310:

Das Verhalten der Nahtstelle in DB43xx.DBB1 sowie der NC Satzverarbeitung wird mit der Einstellung in MD20270: $MC_CUTTING_EDGE_DEFAULT und MD20310: $MC_TOOL_MANAGEMENT_MASK Bit 5, 6, 7 und 8 beeinflusst.

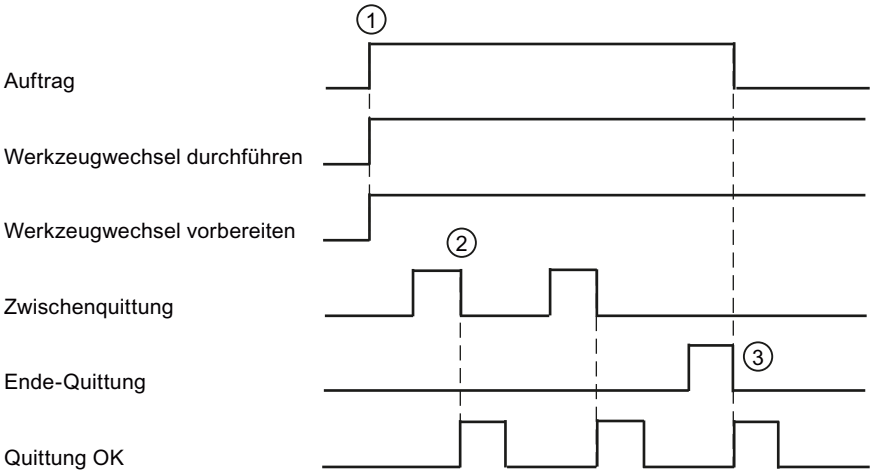
Der hier beschriebene Ablauf entspricht der Voreinstellung des Maschinendatums.

Ablauf der Werkzeugverwaltung ohne Satz-Splitting (Maschinendaten-Einstellung Fräsen)

```
Txx Myy ;
```

Myy ist die Einstellung aus MD22560: $MC_TOOL_CHANGE_M_CODE

Diese Art der Programmierung wird nicht empfohlen!



Signalbeschreibung:

① Das PLC-Programm bekommt einen neuen Auftrag. Der Auftrag "Werkzeugwechsel vorbereiten" und der Auftrag "Werkzeugwechsel durchführen" kommen gemeinsam. Txx und M206 wurden in einem NC Satz programmiert.

② Es werden mehrere Zwischenschritte quittiert. Der Zustand des Auftrags bleibt unverändert. Die Positionen der Werkzeuge werden anhand der Transferschritte der Quittungen von der Werkzeugverwaltung aktualisiert.

③ Es wird die Ende-Quittung mit OK zurückgemeldet und gleichzeitig das Bit für den Auftrag zurückgesetzt.

Die Beschreibung des Auftrags ("Werkzeugwechsel durchführen" und "Werkzeugwechsel vorbereiten") wird nicht zurückgesetzt. Byte 1 von DB43xx wird erst mit dem nächsten Auftrag neu geschrieben.

Hinweis

MD20310: $MC_TOOL_MANAGEMENT_MASK

Das Verhalten der Nahtstelle in DB43xx.DBB1 wird mit der Einstellung in MD20310 Bit 10 beeinflusst.

Der hier beschriebene Ablauf entspricht der Voreinstellung des Maschinendatums.

6.4 Schritt-Tabellen projektieren

Schritt-Tabellen projektieren

Die Transferschritt-Tabellen (TM_CTS im DB9900 und TM_VTS im DB9901) und die Quittungsschritt-Tabelle (TM_ACK im DB9902) finden Sie im Programming Tool unter "Bibliotheken" → "Spezielle Datenbausteine". Mit einem Doppelklick werden die Bausteine in das Projekt kopiert.

Die Struktur der Datenbausteine ist fest vorgegeben. Da die Datenbausteine noch nicht mit den notwendigen Daten befüllt sind, müssen sie im Programming Tool über den Menübefehl "Ansicht" → "Datenbaustein" bearbeitet werden. Die Projektierung der konstanten Tabellen

(TM_CTS im DB9900 und TM_ACK im DB9902) erfolgt durch Schreiben der Anfangswerte der Datenbausteine im Programming Tool.

Zusammen mit dem PLC-Programm werden die Anfangswerte der Datenbausteine in die Steuerung geladen. Geänderte Anfangswerte werden erst nach einem Neustart der PLC aktiv.

Projektierung der Transferschritte

Änderungen von Werkzeug- und Magazinpositionen durch das PLC-Programm müssen der Werkzeugverwaltung mitgeteilt werden. Als Hilfsmittel dient dazu eine Tabelle aller mechanischen Einzelbewegungen, die quittiert werden sollen. Die Tabelle enthält für jeden Werkzeug-Transfer jeweils die Start- und Zielposition des Werkzeugs oder für die Positionierung eines Magazinplatzes an eine Übergabestelle (Wechsel-, Belade- oder Entladestelle) die Magazinposition und die Bezeichnung der Übergabestelle.

- Die Transferschritte 1 ... 64 sind fest in TM_CTS (DB9900) projektiert und nur durch Neuladen änderbar.
- Die Transferschritte 101 ... 164 in TM_VTS (DB9901) können vom PLC-Programm ganz oder teilweise überschrieben werden (z. B. durch das Eintragen des aktuellen Magazinplatzes).

Codierung für Position aus Auftrag

Die Plätze realer Magazine werden in der konstanten Transferschritttabelle nicht mit ihren tatsächlichen Werten (z. B. 1/14 für Magazin 1 Platz 14) sondern mit symbolischen Werten (0/1) oder (0/2) bezeichnet. Andernfalls würde die Transferschritttabelle bei großen Magazinen riesige Ausmaße annehmen.

Diese symbolischen Werte haben folgende Bedeutung:

Magazin / Platz	Bedeutung
(0 / 1)	Die Quellposition des einzigen oder neuen Werkzeugs aus dem Auftrag soll benutzt werden.
(0 / 2)	Die Zielposition des alten Werkzeugs aus dem Auftrag soll benutzt werden.
(0 / 3)	Die Zielposition des einzigen oder neuen Werkzeugs aus dem Auftrag soll benutzt werden.

Diese symbolische Notationsform kann nur für synchrone Zwischenquittungen und Ende-Quittungen benutzt werden, da asynchrone Mitteilungen keinen Auftrag als Bezug haben.

Beispiel: Konstante Transferschritt-Tabelle

Transferschritt	Adresse DB9900	Name	Anfangswert	Kommentar
1	0.0	SrcMag_1	0	Quell-Magazinnummer des Transferschritts
	2.0	SrcPos_1	1	Quell-Positionsnummer des Transferschritts
	4.0	DstMag_1	0	Ziel-Magazinnummer des Transferschritts
	6.0	DstPos_1	1	Ziel-Positionsnummer des Transferschritts

Transferschritt	Adresse DB9900	Name	Anfangswert	Kommentar
2	8.0	SrcMag_2	0	Quell-Magazinnummer des Transferschritts
	10.0	SrcPos_2	1	Quell-Positionsnummer des Transferschritts
	12.0	DstMag_2	9998	Ziel-Magazinnummer des Transferschritts
	14.0	DstPos_2	2	Ziel-Positionsnummer des Transferschritts

Beispiel als komplette Schritte

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
1	0	1	0	1	Werkzeug vorbereiten: Magazin wird zur Wechselstelle des neuen Werkzeugs positioniert.
2	0	1	9998	2	Werkzeugwechsel: Werkzeug von Magazin nach Greifer 1
3	9998	1	9998	3	Werkzeugwechsel: Werkzeug von Spindel nach Greifer 2
4	9998	2	9998	1	Werkzeugwechsel: Werkzeug von Greifer 1 nach Spindel
5	9998	3	0	2	Werkzeugwechsel: Werkzeug von Greifer 2 nach Magazin

Beispiel: Variable Transferschritt Tabelle

Transferschritt	Adresse DB9901	Name	Anfangswert	Kommentar
101	0.0	SrcMag_101	1	Quell-Magazinnummer des Transferschritts
	2.0	SrcPos_101	0	Quell-Positionsnummer des Transferschritts
	4.0	DstMag_101	9998	Ziel-Magazinnummer des Transferschritts
	6.0	DstPos_101	1	Ziel-Positionsnummer des Transferschritts
102	8.0	SrcMag_102	1	Quell-Magazinnummer des Transferschritts
	10.0	SrcPos_102	0	Quell-Positionsnummer des Transferschritts
	12.0	DstMag_102	9998	Ziel-Magazinnummer des Transferschritts
	14.0	DstPos_102	2	Ziel-Positionsnummer des Transferschritts

Beispiel als komplette Schritte

Transfer-schritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
101	1	0	9998	1	"Werkzeugwechsel vorbereiten": Magazinplatz wird zur Wechselstelle positioniert. Die Quellposition muss vom PLC-Programm eingetragen werden.
102	1	0	9998	2	"Werkzeugwechsel vorbereiten": Werkzeug von Magazin nach Zwischenspeicher. Die Quellposition muss vom PLC-Programm eingetragen werden.

6.5 Quittungsschritte projektieren

Projektierung der Quittungsschritte

Zum Quittieren von Werkzeug und Magazinbewegungen stellt die PLC 31 Quittungsschritte bereit, die über entsprechende Bits auf der Anwendernahtstelle zu aktivieren sind. Die Daten dieser Quittungsschritte (mit Ausnahme des Sonderfalls Quittungsschritt 9: Total-Quittung) werden in der Quittungsschritt-Tabelle TM_ACK (DB9902) hinterlegt. Zu einem Quittungsschritt werden die Transferschritte (Nummer des Transferschritts aus der Transferschritt-Tabelle) für altes und neues Werkzeug und ein Quittungsstatus zusammengefasst.

Wichtig ist es, in diese Tabelle einen Quittungsschritt mit dem Status 3 aufzunehmen, damit eventuelle Fehler zurückgesetzt werden können. Die Nummern für den Transfer in diesem Quittungsschritt sind 0.

Sonderbedeutung für Transferschritt 0

Es werden nur die dem Quittungsschritt zugeordneten Transferschritte ausgeführt. Ist nur ein oder kein Transferschritt zugeordnet wird für das Werkzeug mit Transferschritt = 0 kein Werkzeugtransfer ausgeführt. Das Werkzeug ist nicht vorhanden oder verbleibt auf seinem bisherigen Platz.

Beispiel: Quittungsschritt-Tabelle

Quittungs-schritt	Adresse DB9902	Name	Anfangs-wert	Kommentar
1	0.0	TsNewT_1	0	Transferschritt-Nummer des neuen Werkzeugs
	1.0	TsOldT_1	0	Transferschritt-Nummer des alten Werkzeugs
	2.0	State_1	3	Status an NC

Quittungs-schritt	Adresse DB9902	Name	Anfangs-wert	Kommentar
2	4.0	TsNewT_2	1	Transferschritt-Nummer des neuen Werkzeugs
	5.0	TsOldT_2	0	Transferschritt-Nummer des alten Werkzeugs
	6.0	State_2	1	Status an NC

Beispiel als komplette Schritte

Quittungs-schritt	Transferschritt		Quittungs-status	Bemerkung
	neues Werkzeug	altes Werkzeug		
1	0	0	3	Kommando "Auftrag abbrechen"
2	2	0	1	Werkzeug vorbereiten: Werkzeug von Magazin (Wechselstelle) in den Greifer 1 übernommen.

Die Quittung erfolgt durch Setzen des entsprechenden Bits in der Anwendernahtstelle:

- DB40xx für Beladen/Entladen, Umsetzen oder Magazin positionieren
- DB42xx für "Werkzeugwechsel vorbereiten" und "Werkzeugwechsel durchführen"

Das Quittungsbit wird von der PLC-Firmware einen PLC-Zyklus nach der Verarbeitung zurückgesetzt.

Im gleichen Datenbaustein, in dem quittiert wurde, wird für einen PLC Zyklus die Rückmeldung in Bit 100.0 (Quittung OK) oder als statisches Signal Bit 100.1 (Quittung fehlerhaft), bei fehlerhafter Quittung in Byte 104 der Fehlerstatus und in den Bytes DBB108 bis DBB111 die zuletzt gesetzten Quittungsbits ausgegeben. Diese statisch anstehenden Bits dienen im Fehlerfall zur Diagnose, welcher Quittungsschritt den Fehler ausgelöst hat. Werden im PLC-Programm mehrere Quittungsbits gesetzt, werden diese auch in das Abbild eingetragen. Das Rücksetzen des Fehlerstatus erfolgt mit dem Quittieren des Fehlers durch den Anwender in DB40xx.DBX9.0 oder DB42xx.DBX9.0.

6.6 PLC-Programm anpassen

PLC-Programm anpassen

Die Steuerung der mechanischen Vorgänge, Überwachung und Verhinderung von möglichen Kollisionen und die Quittung der Positionsänderungen der Werkzeuge sind Aufgabe des PLC-Programms.

Siehe auch

Im PLC-Projekt auf der Toolbox CD finden Sie Beispiele für Transferschritte und deren Quittung für eine Drehmaschine mit Revolvermagazin und eine Fräsmaschine mit Kettenmagazin und Doppelgreifer:

- Anwender-Beispiel für Fräsmaschine (Seite 87)
- Anwender-Beispiel für Drehmaschine (Seite 75)

Diese Bausteine sind Funktionsbeispiele für die Quittung verschiedener Aufträge der Werkzeugverwaltung.

Aufträge quittieren

Viele Werkzeugbewegungen oder Aufträge der Werkzeugverwaltung können ohne vorherige Zwischenquittung direkt mit Total-Quittung an Bit 0.0 der NC-/PLC-Schnittstelle quittiert werden.

Zum Beispiel:

- Drehungen eines Revolvermagazins
- Beladen/Entladen (nur bei Systemen ohne zusätzliche Zwischenspeicher wie Handling-Systeme, Lader usw.)
- Wechsel von Handwerkzeugen
- Magazin positionieren

Regeln beim Quittieren

Sind Zwischenschritte sinnvoll, müssen einige Regeln beim Quittieren eingehalten werden:

Vom PLC-Programm muss sichergestellt sein, dass alle Quittungen ordnungsgemäß an die Werkzeugverwaltung übermittelt werden.

- Es darf immer nur ein Quittungssignal an die Werkzeugverwaltung gesendet werden.
- Synchrone Quittungen sind nur bei anstehendem Auftrag zulässig.
- Es dürfen nur gültige Transferschritt-Nummern verwendet werden (1 - 64, 100 - 164). Bei asynchronen Mitteilungen ist bei Status 201 mindestens ein Transferschritt, bei Status 204 ein Transferschritt für das Neuwerkzeug anzugeben.
- Codierte Positionen in den Transferschritten dürfen nur für synchrone Quittungen und nur mit den Werten 0/1, 0/2 oder 0/3 verwendet werden.
- Es dürfen keine unerlaubten Quittungsstatus verwendet werden.
- Magazin positionieren mit Auftrag darf nur eine synchrone Quittung (Ende-Quittung) erhalten. Zwischenpositionen müssen mit asynchronen Mitteilungen an die Werkzeugverwaltung gemeldet werden
- Quittungssignale müssen vom PLC-Programm zurückgesetzt werden. Nach Setzen eines Quittungsbits darf die NC-/PLC-Schnittstelle bis zur Rückmeldung in DB41xx/DB43xx.DBB100 nicht verändert werden!
- Asynchrone Mitteilungen mit zwei Transferschritten müssen auf der Schnittstelle für Werkzeugwechsel (DB42xx) quittiert werden.

6.7 Auskunft zum Magazinplatz

Übersicht

Über die Schnittstelle "NC-Dienste" des DB1200 können Sie bis zu 8 NC-Variablen in einem Auftrag zu lesen.

Variable $TC_MPP2 mit Index 7: Platztyp

Parametrierung:

Variable aus NC lesen	Adresse	Signal	Gültige Werte
Auftrag	DB1200.DBX0.0	Start	0/1
	DB1200.DBX0.1	Variable schreiben	0
	DB1200.DBB1	Anzahl der Variablen	1 ... 8
Parameter	DB120x.DBW1000	Variablen Index	7
	DB120x.DBW1002	Platznummer	1 ... 31999
	DB120x.DBW1004	Magazinnummer	1 ... 9999
Ergebnis	DB1200.DBX2000.0	Auftrag beendet	0/1
	DB1200.DBX2000.1	Fehler im Auftrag	0/1
	DB120x.DBX3000.0	Variable ist gültig	0/1
	DB120x.DBB3001	Zugriffsergebnis	0/3/5/10
	DB120x.DBW3004	Daten von NC-Variable	n

Legende:

n > 0: Platztyp für virtuellen Platz

n = 0: "match all" (Zwischenspeicher)

n = 9999: undefiniert (kein virtueller Platz)

Variable $TC_MPP4 mit Index 8: Platzzustand

Parametrierung:

Variable aus NC lesen	Adresse	Signal	Gültige Werte
Auftrag	DB1200.DBX0.0	Start	0/1
	DB1200.DBX0.1	Variable schreiben	0
	DB1200.DBB1	Anzahl der Variablen	1 ... 8
Parameter	DB120x.DBW1000	Variablen Index	8
	DB120x.DBW1002	Platznummer	1 ... 31999
	DB120x.DBW1004	Magazinnummer	1 ... 9999

Variable aus NC lesen	Adresse	Signal	Gültige Werte
Ergebnis	DB1200.DBX2000.0	Auftrag beendet	0/1
	DB1200.DBX2000.1	Fehler im Auftrag	0/1
	DB120x.DBX3000.0	Variable ist gültig	0/1
	DB120x.DBB3001	Zugriffsergebnis	0/3/5/10
	DB120x.DBW3004	Daten von NC-Variable	n

Legende:
n = 1 gesperrt
n = 2 frei (<> belegt)
n = 4 reserviert für Werkzeug im Zwischenspeicher
n = 8 reserviert für zu beladendes Werkzeug
n = 16 belegt im linken Halbplatz
n = 32 belegt im rechten Halbplatz
n = 64 belegt im oberen Halbplatz
n = 128 belegt im unteren Halbplatz

Variable $TC_MPP6 mit Index 9: T-Nummer des Werkzeugs auf diesem Platz

Parametrierung:

Variable aus NC lesen	Adresse	Signal	Gültige Werte
Auftrag	DB1200.DBX0.0	Start	0/1
	DB1200.DBX0.1	Variable schreiben	0
	DB1200.DBB1	Anzahl der Variablen	1 ... 8
Parameter	DB120x.DBW1000	Variablen Index	9
	DB120x.DBW1002	Platznummer	1 ... 31999
	DB120x.DBW1004	Magazinnummer	1 ... 9999
Ergebnis	DB1200.DBX2000.0	Auftrag beendet	0/1
	DB1200.DBX2000.1	Fehler im Auftrag	0/1
	DB120x.DBX3000.0	Variable ist gültig	0/1
	DB120x.DBB3001	Zugriffsergebnis	0/3/5/10
	DB120x.DBW3004	Daten von NC-Variable	n

n = T-Nummer des Werkzeugs auf parametriertem Platz

Fehlerfall (für alle Variablen)

Im Fehlerfall bleibt DB120x.DBX3000.0 = 0 und es erfolgt ein Eintrag im Zugriffsergebnis:

Werte in DB120x.DBB3001	
0	kein Fehler
3	Objektzugriff nicht erlaubt
5	ungültige Adresse
10	Objekt nicht existent

6.8 PI Dienst: TMMVTL

Funktion

Mit dem Programm-Instanz Dienst TMMVTL ist es von PLC aus möglich, einen Auftrag zum Umsetzen eines Werkzeugs anzustoßen. Nach fehlerfreiem "PI Start" führt die Werkzeugverwaltung für das Werkzeug auf dem angegebenen Quellplatz eine Leerplatzsuche im Zielmagazin durch. Anschließend erhält die PLC einen Auftrag zum Umsetzen des Werkzeugs (Anwendernahtstelle DB41xx.DBB0).

Parametrierung

PI Dienst im NC-Bereich starten:

PI Dienst	Adresse	Signal	Gültige Werte
Auftrag	DB1200.DBX4000.0	Start	0/1
	DB1200.DBB4001	PI Index	5
Parameter	DB1200.DBW4004	Werkzeugnummer (interne T-Nummer)	1 ... 31999
	DB1200.DBW4006	Quell-Platznummer	1 ... 31999
	DB1200.DBW4008	Quell-Magazinnummer	1 ... 31999
	DB1200.DBW4010	Ziel-Platznummer	-1
	DB1200.DBW4012	Ziel-Magazinnummer	1 ... 32000
Ergebnis	DB1200.DBX5000.0	Auftrag beendet	0/1
	DB1200.DBX5000.1	Fehler im Auftrag	0/1

Das Werkzeug kann wahlweise über T-Nummer oder über Platz- und Magazinnummer angegeben werden. Eine nicht benutzte Spezifikation trägt den Wert -1.

Mit Parameter Ziel-Platznummer =-1 wird entsprechend der eingestellten Suchstrategie im gesamten Magazin ein Leerplatz für das Werkzeug gesucht. Ist ein Zielplatz angegeben, wird geprüft, ob der Platz mit der angegebenen Ziel-Platznummer für das Werkzeug frei und geeignet ist.

Bei Ziel-Magazinnummer = -1 wird für Werkzeuge auf einem Zwischenspeicher entsprechend der sich aus $TC_MDP2 ergebenden Zuordnung gesucht.

Beispiele:

- Bei Verwendung von Zwischenspeichern zum Rücktransport des Werkzeugs (zum Beispiel Toolboy und/oder Shifter) kann im Zuge des asynchronen Rücktransports eine explizite Leerplatzsuche im Magazin notwendig sein. In diesem Fall muss sich die PLC nicht den Ursprungsplatz merken, dieser PI Dienst sucht nach einem geeigneten Platz.
- Aus einem Hintergrundmagazin soll ein Werkzeug in das Vordergrundmagazin geholt werden.

Beispiel: Beladen / Entladen

7

Programmierung

Zum Beladen werden Werkzeuge direkt in das Magazin oder die Spindel gesteckt, zum Entladen direkt aus dem Magazin entnommen. Es reicht in der Regel eine einmalige Quittung des Bedieners oder des PLC-Programms (Werkzeughalter ist wieder geschlossen) als Meldung, dass der Vorgang beendet ist. Die Projektierung eines Transferschritts ist nicht notwendig. Es kann die Total-Quittung in DB40xx.DBX0.0. gesetzt werden.

Quittung an Werkzeugverwaltung:

Quittungsschritt	Quittungsbit	Transferschritt neues Werkzeug	Transferschritt altes Werkzeug	Status
xxx	DB4000.DBX0.0	--	--	(99)

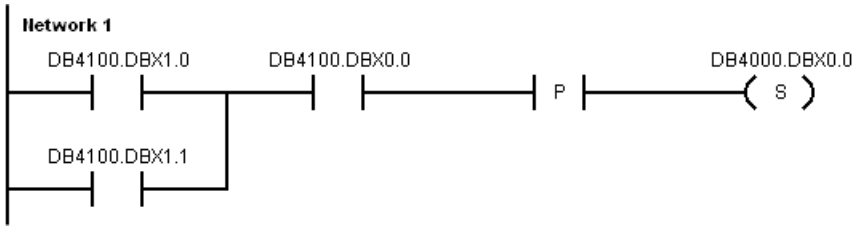


Bild 7-1 Programmierung im PLC-Programm

Beladevorgänge über Handling-Systeme oder der Transport eines Werkzeugs von der Spindel ins Magazin können über weitere asynchrone Mitteilungen erfolgen.

Hinweis

Handwerkzeuge

Bitte beachten Sie die Besonderheiten beim Einsatz von Handwerkzeugen (Seite 24).

Ein Werkzeug das in die Spindel oder ein Magazin beladen wurde ist kein Handwerkzeug. Infolge dessen wird es mit T0/M06 im Magazin abgelegt.

In den folgenden Abschnitten sind weitere Abläufe eines Beladevorgangs beschrieben.

Beladen über die Spindel mit Vorwahl eines Magazinplatzes

Sie können ein neues Werkzeug direkt auf dem gewünschten freien Magazinplatz anlegen oder ein Werkzeug aus der Werkzeugliste, das sich nicht im Magazin befindet, mit dem Dialog "Beladen" auf einen Magazinplatz bringen.

1. Hier wird immer ein Auftrag der Werkzeugverwaltung an der Nahtstelle für die Beladestelle ausgegeben. Dieser Auftrag muss quittiert werden.
2. Danach wechseln Sie das Werkzeug, das sich noch nicht im Magazin befindet, mit Txx M6 oder über einen asynchronen Transfer in die Spindel.
3. Stecken Sie das Werkzeug danach von Hand in die Spindel und legen es mit T0 M6 in das Magazin ab.

Die Werkzeuge können übergroß oder festplatz-codiert sein.

Beladen über die Spindel ohne Vorwahl eines Magazinplatzes

Sie können ein neues Werkzeug direkt auf der Spindel anlegen oder ein Werkzeug aus der Werkzeugliste, das sich nicht im Magazin befindet, mit dem Dialog "Beladen" auf die Spindel bringen.

1. Hier wird immer ein Auftrag der Werkzeugverwaltung an der Nahtstelle für die Beladestelle ausgegeben. Dieser Auftrag muss quittiert werden.
2. Stecken Sie das Werkzeug danach von Hand in die Spindel und legen es mit T0 M6 in das Magazin ab. Von der Werkzeugverwaltung wird ein freier Platz ausgewählt auf dem das Werkzeug abgelegt werden kann.

Die Werkzeuge können übergroß oder festplatz-codiert sein.

Beladen direkt in das Magazin

Positionieren Sie den gewünschten Magazinplatz zur Beladestelle. Sie können ein neues Werkzeug direkt auf dem gewünschten freien Magazinplatz anlegen oder ein Werkzeug aus der Werkzeugliste, das sich nicht im Magazin befindet, mit dem Dialog "Beladen" auf den gewählten Magazinplatz bringen.

1. Hier wird immer ein Auftrag der Werkzeugverwaltung an der Nahtstelle für die Beladestelle ausgegeben. Dieser Auftrag muss quittiert werden.
2. Stecken Sie jetzt das Werkzeug in das Magazin.

Für diesen Ablauf gibt es keine Einschränkungen oder Randbedingungen.

Beispiel: Handwerkzeuge wechseln

8

Programmierung

In MD22562: $MC_TOOL_CHANGE_ERROR_MODE Bit 1=1 ist vom NC-Teileprogramm zusätzliche Werkzeuge ohne Magazinzuordnung ausgewählt worden. Das ausgewählte Werkzeug muss von Hand in die Maschine eingesetzt und nach der Bearbeitung wieder von Hand entnommen werden ("Handwerkzeuge").

Der Bediener hat dafür zu sorgen, dass der Datensatz des Werkzeugs, das er auf die Spindel steckt, sich in der NC befindet, oder dass er zu dem in der NC abgelegten Datensatz das passende Werkzeug auf die Spindel bringt.

ACHTUNG
Sicherheitsvorschriften bei Handwerkzeugen
Die Einhaltung der Sicherheitsvorschriften ist im PLC-Programm abzusichern.

Dem PLC-Programm wird bei einem Auftrag zu einem Werkzeugwechsel mit DB43xx.DBX1.5 und DBX1.6 mitgeteilt, ob ein Handwerkzeug beteiligt ist. Der Bediener erhält die Aufforderung, den Werkzeugwechsel durchzuführen:

17212 "Kanal %1: Werkzeugverwaltung: Handwerkzeug %3, Duplonr. %2 einwechseln auf Werkzeughalter %4"

17214 "Kanal %1: Werkzeugverwaltung: Handwerkzeug %3 von Spindel/Werkzeughalter %2 entnehmen"

Die Alarme werden durch die Quittung der PLC nach Werkzeugwechsel zurückgesetzt.

Ausgangsposition 1

Handwerkzeug in der Spindel soll gegen anderes Handwerkzeug gewechselt werden.

Auftrag von Werkzeugverwaltung an PLC-Programm (Werkzeugwechsel):

DB4300.DBX0.0, DBX1.2, DBX1.5 und DBX1.6 ("Werkzeugwechsel vorbereiten")

DB43xx.DBW6	Quell-Magazinnummer	9999
DB43xx.DBW8	Quell-Platznummer	1
DB43xx.DBW10	Ziel-Magazinnummer	9999
DB43xx.DBW12	Ziel-Platznummer	1

Quittung "Werkzeugwechsel vorbereiten":

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	0	0	1

Damit steht auf der Nahtstelle: DB4300.DBX0.0 /1.1, DBX1.5 und DBX1.6 (durchführen")

Der Auftrag bleibt sonst unverändert, die Werkzeuge sind noch auf den Ausgangspositionen. Zur Quittung der Entnahme des alten Werkzeugs kann ein Zwischenschritt eingebaut werden.

Synchrone Quittung

Das Werkzeug befindet sich nicht mehr in der Spindel.

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
6 DB9900.DBW40	9998	1	9999	1	Werkzeug aus Spindel entnommen

Quittungsschritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	0	6	105

Nach der Zwischenquittung der leeren Spindel wird nach Einsetzten des neuen Werkzeugs in die Spindel der Wechsel mit einer Total-Quittung beendet:

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
	DB4200.DBX0.0			(99)

Ausgangsposition 2

Handwerkzeug ist in der Spindel und ein Werkzeug aus dem Magazin soll eingewechselt werden.

Auftrag von Werkzeugverwaltung an PLC-Programm (Werkzeugwechsel):

DB4300.DBX0.0, DBX1.2 und DBX1.6 ("Werkzeugwechsel vorbereiten")

DB43xx.DBW6	Quell-Magazinnummer	1
DB43xx.DBW8	Quell-Platznummer	6
DB43xx.DBW10	Ziel-Magazinnummer	9999
DB43xx.DBW12	Ziel-Platznummer	1

Quittung "Werkzeugwechsel vorbereiten":

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	0	0	1

Damit steht auf der Nahtstelle: DB4300.DBX0.0 /1.1 und 1.6 ("Werkzeugwechsel durchführen")

Der Auftrag bleibt sonst unverändert, die Werkzeuge sind noch auf den Ausgangspositionen.

Synchrone Quittung

Das alte Werkzeug befindet sich nicht mehr in der Spindel.

Transferschritt	Von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
6 DB9900.DBW40	9998	1	9999	1	Asynchrone Mitteilung, Werkzeug von Spindel entladen

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	0	6	105

Die Spindel ist jetzt leer, das alte Werkzeug ist außerhalb des Magazins.

Nächster Schritt

Synchrone Quittung Neuwerkzeug nach Greifer 1

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
3 DB9900.DBW16	0	1	9998	2	Neuwerkzeug nach Greifer 1

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	3	0	105

Auftrag ist unverändert.

Nächster Schritt

Synchrone Quittung neues Werkzeug von Greifer 1 nach Spindel

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
4 DB9900.DBW24	9998	2	9998	1	Neues Werkzeug von Greifer 1 nach Spindel

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	4	0	105

Damit sind die Werkzeugbewegungen abgeschlossen.

Ende-Quittung

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
xxx	DB4200.DBXx.x	0	0	1

Der Schritt neues Werkzeug von Greifer 1 nach Spindel kann entfallen und durch eine Total-Quittung ersetzt werden. Damit wird der Werkzeugverwaltung ebenfalls mitgeteilt, dass alle Werkzeuge auf ihren Zielpositionen sind.

Quittungs-schritt	Quittungsbit	Transferschritt neues WZ	Transferschritt altes WZ	Status
	DB4200.DBX0.0			(99)

Anwender-Beispiel für Drehmaschine 9

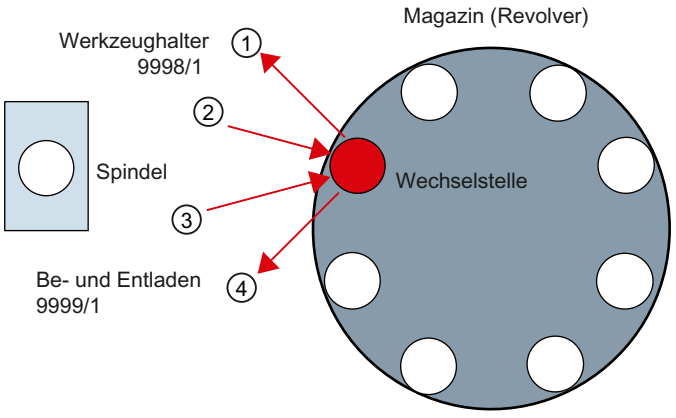
9.1 Beispiel: Drehmaschine mit Revolvermagazin (MAG_CONF_MPF)

Beispiel-Datei

Das Programm zur Magazinkonfiguration finden Sie auf der Toolbox.

Das Programm kann in die Steuerung eingelesen werden und ist an die konkrete Anlage anzupassen.

Konfiguration



1 ... 4 Transferschritte

Bild 9-1 Drehmaschine mit Revolvermagazin

Beschreibung des Programms

Zu Beginn werden alle alten Magazindefinitionen und Werkzeuge gelöscht. Im weiteren Programmablauf werden alle Magazine und Zwischenspeicher durch Schreiben der Magazinparameter neu erzeugt und definiert.

In N70 kann die Suchstrategie für Werkzeuge und Magazinplätze festgelegt werden.

Für Revolvermagazine ist es sinnvoll alle Plätze festplatz-codiert zu definieren. Für Magazintyp 3 wird in N320 $TC_MAP3[NUM_MAG] = 81 (Bit 6 = 1) eingestellt.

Die Plätze des Kettenmagazins werden ab N430 bis N500 eingerichtet. Magazin Platztyp = 0 bedeutet, dass der Magazinplatz mit Werkzeugen unterschiedlicher Platztypen beladen werden kann.

Ab N520 werden die Zwischenspeicher eingestellt.

Ab N920 wird die Zuordnung der Zwischenspeicher zur Spindel/Werkzeughalter und dem Magazin vorgenommen.

Nachdem das Programm zur Magazinkonfiguration durchlaufen ist, führen Sie einen Neustart der NC durch (NCK Reset).

Siehe auch

Eine genaue Beschreibung der verwendeten Parameter finden Sie im Funktionshandbuch Werkzeugverwaltung der SINUMERIK 840D sl.

Beispiel MAG_CONF_MPF

1. Anlagenkonfiguration:
   - 1 Revolvermagazin mit 8 Plätzen (in N40 einstellbar)
   - 1 Beladestelle
   - 3 Zwischenspeicherplätze (in N50 einstellbar, ab N540 Zuordnungen)
2. Teileprogramm:

```
;MAG_CONF_MPF
N10 def int NUM_MAG,MAG_TYPE, LOCATIONS, PLA-
CE, NUM_BUFFER, NUM_LOAD, PLACE_SEARCH
;
N20 NUM_MAG = 1                                ;Nummer des  Magazins
N30 MAG_TYPE = 3                               ;Magazinart (1: Kette, 3: Revolver,
                                               5: Flächenmagazin)
N40 LOCATIONS = 8                              ;Anzahl der Magazinplätze
N50 NUM_BUFFER = 1                             ;Anzahl der Zwischenspeicher (Spin-
                                               del, Greifer)
N60 NUM_LOAD = 1                               ; Anzahl der Beladestellen
N70 PLACE_SEARCH = 257                         ; Art der Suchstrategie
;= 257 Bit13=0 kein Tausch Altwerkzeug auf Platz von Neuwerkzeug
;Einstellung für Pickup Magazin
;=12289 Bit13=1 Tausch Altwerkzeug auf Platz von Neuwerkzeug
;Einstellung für Kettenmagazin
N80;
N90;
;Parameter prüfen
N100 STOPRE
N110 if ((NUM_MAG==0)or(LOCATIONS==0))
N120  Err1:STOPRE
N130  MSG("Wrong Parameter --> Cancel")
N140  G04 F4
N150  STOPRE
N160  M0
N170  GOTOB Err1
N180 endif
```

```
N190; Magazin Konfiguration
N200;
N210;
N220; Lösche alte Daten wenn, Magazin 1 angelegt wird
N230 if NUM_MAG ==1
N240 $TC_MAP1[0]=0                              ; lösche Magazine
N250 $TC_DP1[0,0]=0                             ; lösche Werkzeuge
N260 STOPRE
N270 endif
; Konfiguration
;
N280 $TC_MAMP2= PLACE_SEARCH                    ; Art der Suchstrategie
;
; Magazin
; Magazin einrichten
N290 $TC_MAP1[NUM_MAG]= MAG_TYPE
N300 $TC_MAP2[NUM_MAG]="MAGAZIN"<<NUM_MAG
N310 if MAG_TYPE == 3
N320 $TC_MAP3[NUM_MAG]=81                       ; Magazinzustand, alle Plätze fest-
                                                platz-codiert bei Revolvermagazin
N330 else
N340 $TC_MAP3[NUM_MAG]=17                       ; Magazinzustand
N350 endif
N360 $TC_MAP4[NUM_MAG]=-1
N370 $TC_MAP5[NUM_MAG]=-1
N380 $TC_MAP6[NUM_MAG]=1                        ; Anzahl Zeilen des Magazins
N390 $TC_MAP8[NUM_MAG]=0
N400 $TC_MAP9[NUM_MAG]=0
N410 $TC_MAP7[NUM_MAG]=LOCATIONS                ; Anzahl Magazinplätze
N420 $TC_MAP10[NUM_MAG]=PLACE_SEARCH
;
; Magazinplätze
N430 for PLACE=1 to LOCATIONS
N440   STOPRE
N450   $TC_MPP1[NUM_MAG,PLACE]=1                ; Platzart
N460   $TC_MPP2[NUM_MAG,PLACE]=0                ; Platzytp,0 mit jedem WZ Platztyp
                                                verträglich
N470   $TC_MPP3[NUM_MAG,PLACE]=1                ; Nebenplatzbetrachtung ein (aus wä-
                                                re 0)
N480   $TC_MPP4[NUM_MAG,PLACE]=2                ; Platzzustand
N490   $TC_MPP5[NUM_MAG,PLACE]=PLACE            ; Platzartindex
N500 endfor
N510 STOPRE
;
N520; Definition Zwischenspeichermagazin (immer Nummer 9998)
```

```
;
N530 $TC_MAP1[9998]=7                          ; Magazinart 7: Zwischenspeicher
N540 $TC_MAP2[9998]="BUFFER"<<NUM_MAG
N550 $TC_MAP3[9998]=17                         ; Magazinzustand
N560 $TC_MAP6[9998]=1                          ; Anzahl Zeilen
N570 $TC_MAP7[9998]=NUM_BUFFER                 ; Anzahl Plätze
;
; Plätze des Zwischenspeichers
; Spindel
N580 $TC_MPP1[9998,1]=2                        ; Platzart (hier Spindel)
N590 $TC_MPP2[9998,1]=0                        ; Platztyp (hier immer 0)
N600 $TC_MPP3[9998,1]=0                        ; Nebenplatzbetrachtung aus
N610 $TC_MPP4[9998,1]=2                        ; Platzzustand
N620 $TC_MPP5[9998,1]=1                        ; Platzartindex
;
N630; Greifer
N640 FOR PLACE=2 to NUM_BUFFER
N650     STOPRE
N660     $TC_MPP1[9998,PLACE]=3                ;(hier Greifer)
N670     $TC_MPP2[9998,PLACE]=0                ;(hier immer 0)
N680     $TC_MPP3[9998,PLACE]=0                ;Nebenplatzbetrachtung aus
N690     $TC_MPP4[9998,PLACE]=2                ;Platzzustand
N700     $TC_MPP5[9998,PLACE]=PLACE            ;Platzartindex
N710 endfor
N720 STOPRE
;
;
N730; Definition Belademagazin (immer Nummer 9999)
;
N740 $TC_MAP1[9999]=9                          ; Magazinart 9: Belademagazin
N750 $TC_MAP2[9999]="BELADEMAGAZIN"<<NUM_MAG
N760 $TC_MAP3[9999]=17                         ; Magazinzustand
N770 $TC_MAP4[9999]=-1
N780 $TC_MAP5[9999]=-1
N790 $TC_MAP6[9999]=1                          ; Anzahl Zeilen
N800 $TC_MAP7[9999]=NUM_LOAD                   ; Anzahl Plätze
N810 STOPRE;
;
N820; Belademagazinplätze
;
N830 for PLACE=1 to NUM_LOAD
N840     STOPRE
N850     $TC_MPP1[9999,PLACE]=7                ; Platzart Beladestelle
N860     $TC_MPP2[9999,PLACE]=0                ; Platztyp (hier immer 0)
```

```
N870      $TC_MPP3[9999,PLACE]=0                  ; Nebenplatzbetrachtung aus
N880      $TC_MPP4[9999,PLACE]=2                  ; Platzzustand
N890      $TC_MPP5[9999,PLACE]=PLACE              ; Platzartindex
N900 endfor
N910 STOPRE
;
;
N920; Offsets (Abstände) ; Abstände zum Magazin
;
; Zwischenspeicher
N930 for PLACE=1 to NUM_BUFFER
N940  $TC_MDP2[1,PLACE]=0
N950 endfor
N960 STOPRE
;
; Beladestellen
N970 for PLACE=1 to NUM_LOAD
N980    stopre
N990    $TC_MDP1[1,PLACE]=0
N1000 endfor

N1010 M30                                         ; Ende
```

Anzeige in der Bediensoftware

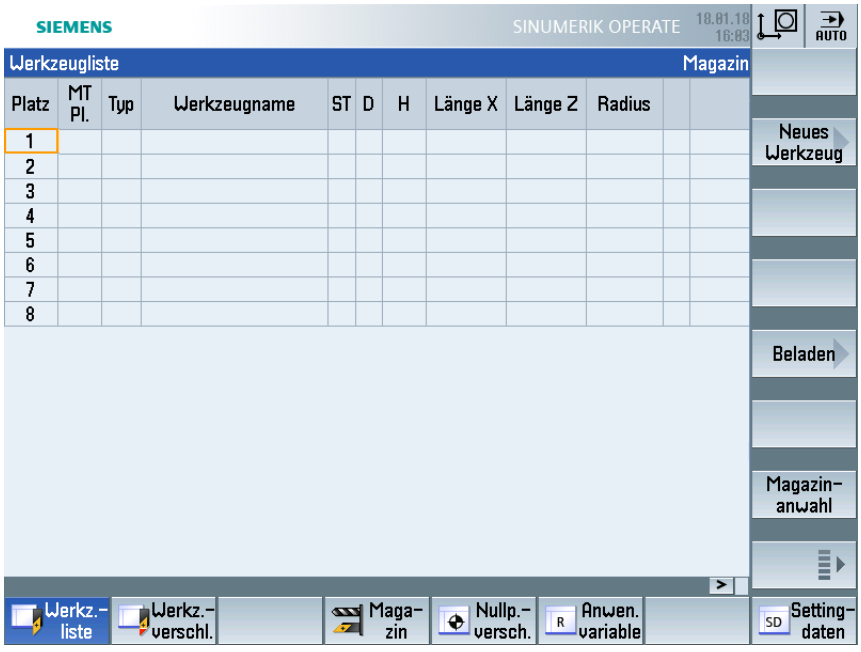


Bild 9-2 Werkzeugliste Drehmaschine

9.2 Beispiel: Quittungsschritte (Drehmaschine)

Quittungsschritte

In der Regel sind die mechanischen Abläufe an einer Drehmaschine einfacher als an einer Fräsmaschine. Bei der im vorherigen Kapitel beschriebenen Konfiguration ohne zusätzlichen Zwischenspeicher können die Werkzeugwechsel nach den mechanischen Bewegungen ohne Transferschritte quittiert werden. Das PLC-Programm muss in den Auftragsschnittstellen eingehende Aufträge erkennen und die mechanischen Bewegungen ausführen.

Total-Quittung an Werkzeugverwaltung:

Quittungsschritt	Quittungsbit	Transferschritt neues Werkzeug	Transferschritt altes Werkzeug	Status
--	DB4000.DBX0.0	--	--	(99)
--	DB4200.DBX0.0	--	--	(99)

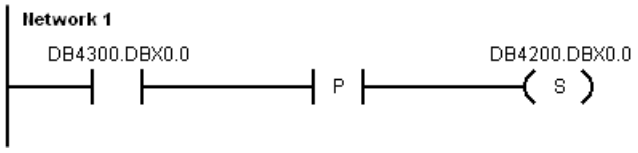


Bild 9-3 Programmierung im PLC-Programm

Für Magazinbewegungen ohne Auftrag von der Werkzeugverwaltung kann eine asynchrone Mitteilung genutzt werden.

Quittung an Werkzeugverwaltung:

Quittungsschritt	Quittungsbit	Transferschritt neues Werkzeug	Transferschritt altes Werkzeug	Status
xxx	DB4000.DBXx.x	101	0	204

Transferschritt	Quelle		Ziel		Bemerkung
	Magazin	Platz	Magazin	Platz	
101 DB9901.DBW0	1	**n**	9998	1	Der variable Platz im Magazin 1 wurde an die Wechselstelle zur Spindel positioniert.

n ist hier die tatsächliche vom PLC-Programm in die variable Transfertabelle einzutragende Platznummer (n ≠ 0).

9.3 Beispiel: Werkzeugwechselzyklus für Drehmaschine

Übergabe-Variablen des T-Ersetzungszyklus:

Voraussetzungen:

- Die Option "Ersatzwerkzeuge für Werkzeugverwaltung" ist nicht aktiv.
- Jede Magazinbewegung wurde der Werkzeugverwaltung mitgeteilt.

In MD10717 $MN_T_NO_FCT_CYCLE_NAME (Name des Zyklus, der anstelle der T-Funktion ausgeführt werden soll z. B. "TCHANGE") stellen Sie ein, dass mit Aufruf des T-Befehls ein Zyklus durchlaufen wird. Der Zyklus wird bei jedem T-Aufruf durchlaufen, unabhängig davon ob ein neues Werkzeug oder das bereits aktive Werkzeug aufgerufen wird. In diesem Zyklus kann die Position des Revolvers auf das programmierte Werkzeug gestellt werden (POSM). Das ist notwendig wenn nach einer Werkzeuganwahl der Revolver von Hand positioniert und das Werkzeug nicht abgewählt wurde. In diesem Fall übergibt die NC keinen neuen Auftrag für "Werkzeugwechsel durchführen" an die NC/PLC-Nahtstelle.

Programmierung

Im folgenden Programmierbeispiel wird mit POSM ein Auftrag zum Positionieren des Magazins an der Nahtstelle ausgegeben. Während der Ausführung der Magazinpositionierung muss die Satzabarbeitung der NC vom PLC-Programm gesteuert werden. In den meisten Fällen wird das Setzen der Einlese- oder Vorschubsperre während dieser Zeit sinnvoll sein. Sobald die im Auftrag angegebene Sollposition erreicht ist (d.h. die asynchrone Mitteilung der Magazinposition), wird der Auftrag mit Total-Quittung beendet.

```
PROC L6 SAVE SBLOF DISPLOF
  IF $C_T_PROG==1                    ; T ist numerisch
    IF $C_T==0                       ; T=0
      T=0
    ENDIF
    IF $C_T>0
      IF $C_T<=$TC_MAP7[1]           ; Magazinplatz existiert?
      POSM($C_T)                     ; Magazin positionieren
      ENDIF
      T=$C_T                         ; T-Programmierung Platznummer
    ENDIF
  ENDIF

  IF $C_TS_PROG==1                   ; T ist Bezeichner
    _TNO_NEW=GETT($C_TS,1)           ; Abfrage der T-Nummer
    IF _TNO_NEW>0                    ; T-Nummer existiert?
      _TL_NEW=$A_MYMLN[_TNO_NEW]     ; Abfrage Platznummer
    ENDIF
    IF _TL_NEW>0                     ; Werkzeug ist im Magazin?
      POSM(_TL_NEW)                  ; Magazin positionieren
```

```
    ENDIF
    T=$C_TS                          ; T-Programmierung ohne Adresserweiterung
  ENDIF
M17
```

Werkzeugaufruf bei Werkzeug messen

Für spezielle Anwendungen wie z. B. Messen in JOG ist es notwendig, ein bestimmtes Werkzeug (z. B. Fraeser_15, Duplo-Nummer 2), unabhängig von seinem Status auf die Spindel oder den Werkzeughalter einzuwechseln, wenn zum Beispiel ein Werkzeug im Einsatz war und von der Werkzeugüberwachung gesperrt wurde. Jetzt soll es vor der erneuten Freigabe ausgemessen werden.Für den Werkzeugaufruf unabhängig vom Status und mit Auswahl von Duplo-Nummer und Werkzeughalternummer steht der NC-Befehl TCA zur Verfügung und wird bei Werkzeugmessen in JOG benutzt. Der NC-Befehl TCA("Werkzeugname", Duplo-Nummer, Werkzeughalter-Nummer) verhält sich bezüglich der Alarm- und Kommandoausgabe an die PLC analog zum T-Befehl.

Randbedingungen

Bei aktiver T-Funktionsersetzung (= Voreinstellung bei Technologie Drehen) sind folgende Randbedingungen zu beachten:

- Der in MD10717 $MN_T_NO_FCT_CYCLE_NAME definierte und bei T-Aufruf durchlaufene Zyklus wird bei dem Befehl TCA nicht gestartet.
  Um sicherzustellen dass bei der Programmierung von TCA auch ein Werkzeugwechselzyklus gestartet werden kann ist der Sprachbefehl TCA in _TCA umdefiniert (Voreinstellung NC).
- TCA kann mit einem Ersetzungszyklus substituiert werden. Der Name des Ersetzungszyklus wird in das MD15710 $MN_TCA_CYCLE_NAME eingetragen. Dieser Ersetzungszyklus wird aus dem Zyklus der umdefinierten TCA-Funktion heraus aufgerufen (TCA.SPF aus der Toolbox).
- Die T- und TCA-Ersetzung können in einem gemeinsamen Ersetzungszyklus realisiert werden.

Hinweis

Die Korrekturanwahl entsprechend MD20270 $MC_CUTTING_EDGE_DEFAULT wirkt analog zum T-Befehl. TCA und D dürfen nicht in einem Satz programmiert werden.

Auf der Toolbox befindet sich ein Beispielzyklus für die Ersetzung beider Werkzeugaufrufe in einem gemeinsamen Ersetzungszyklus.

Das folgende Beispiel beschreibt den Ablauf:

```
PROC TCA(STRING[64] _TOOL_NAME,INT _DUPLO,INT _TH_NO)
                                         ;VERSION: 01.00.07. Oct 22, 2009
                                         ;CHANGE: 01.00.07. Oct 22, 2009
TCA(_TOOL_NAME,_DUPLO,_TH_NO)            ;Werkzeugaufruf an NC
```

```
; fügen Sie hier die Maschinenfunktionen für den Wechsel ein
M17
```

Die Funktion steht unabhängig von der Werkzeugverwaltung zur Verfügung; siehe Kapitel "Ersetzung von Funktionen durch Unterprogramme" im Funktionshandbuch Basisfunktionen (K1) (https://support.industry.siemens.com/cs/document/109766724).

9.4 Beispiel: Satzsuchlauf und MCALL

Voraussetzung

Im Werkzeugwechsel-Unterprogramm ist für Satzsuchlauf über modal aufgerufene Zyklen (z. B. Bohrzyklen) dafür Sorge zu tragen, dass der modale Aufruf erhalten bleibt und sicherzustellen, dass das richtige Werkzeug zum Einsatz kommt.

Weitere Informationen

Inbetriebnahmehandbuch SINUMERIK Operate, (https://support.industry.siemens.com/cs/document/109766724) "Technologische Zyklen für Bohren"

Nach einem Satzsuchlauf mit Berechnung wird im Suchlauf-ASUP das Werkzeugwechsel-Unterprogramm (L6) aufgerufen. Dort darf der modale Aufruf nur abgewählt werden, wenn L6 nicht aus dem ASUP heraus aufgerufen wurde.

Programmierbeispiel

```
L6
N100 IF $P_MC == TRUE   ;MCALL ist aktiv
N110   IF $AC_ASUP <> 0  ; Aufruf erfolgt über Suchlauf-ASUP oder
Überspeichern → MCALL nicht abwählen
N140   ELSE
N150    MCALL
N160   ENDIF
N170 ENDIF
```

Der Maschinehersteller muss ebenfalls sicherstellen, ob ein realer Werkzeugwechsel mit Achsbewegungen und ein modaler Zyklenaufruf zulässig sind. Unter Umständen muss bei programmiertem Werkzeug nach Satzsuchlauf ungleich dem Werkzeug, das sich in der Spindel befindet, der Werkzeugwechsel verhindert werden.

9.5 Beispiel: Drehmaschine mit Gegenspindel

Magazinkonfiguration

In der Magazinkonfiguration erhält das Magazin doppelt so viele Plätze wie real vorhanden, z. B. bei einem Revolver mit 12 Plätzen werden 24 Plätze eingerichtet. Platz 1-12 für die Hauptspindel, Platz 13-24 für die Gegenspindel.

Vom PLC-Programm wird das Magazin so positioniert, dass zum Beispiel für Platz 1 und Platz 13 die gleiche Position angefahren wird. Jeder reale Magazinplatz entspricht damit einem virtuellen Magazinplatz für die Hauptspindel und einem virtuellen Magazinplatz für die Gegenspindel.

9.6 Beispiel: Test auf leere Zwischenspeicher

Vorgehensweise

Lesen der T-Nummer eines Werkzeuges in Greifer 1 und Greifer 2:

1. Tragen Sie im PLC-Programm die Parameter in DB1200 ein.
2. Setzen Sie in DB1200.DBX0.0 den Start zum Lesen der Platzzustände.

Wurde der Auftrag erfolgreich durchgeführt, erhalten Sie die Ergebnisse ab DB1200.DBB3000:

Variable aus NC lesen	Adresse	Signal	Werte
Parameter	DB1200.DBW1000	Variablen Index	9
	DB1200.DBW1002	Platznummer	2
	DB1200.DBW1004	Magazinnummer	9998
	DB1201.DBW1000	Variablen Index	9
	DB1201.DBW1002	Platznummer	3
	DB1201.DBW1004	Magazinnummer	9998
Auftrag	DB1200.DBX0.1	Variable schreiben	0
	DB1200.DBX0.2	PI-Dienst	0
	DB1200.DBB1	Anzahl der Variablen	2
	DB1200.DBX0.0	Start	→ 1
Ergebnis	DB1200.DBB2000.0	Auftrag beendet	1
	DB1200.DBX2000.1	Fehler im Auftrag	0
	DB1200.DBX3000.0	Variable ist gültig	1
	DB1200.DBB3001	Zugriffsergebnis	0
	DB1200.DBW3004	Daten von NC-Variable	n
	DB1201.DBX3000.0	Variable ist gültig	1
	DB1201.DBB3001	Zugriffsergebnis	n
	DB1201.DBW3004	Daten von NC-Variable	0

9.7 Beispiel: Werkzeug aus einem Zwischenspeicher ins Magazin transportieren

Vorgehensweise

Ein Werkzeug aus einem Zwischenspeicher (z. B. Greifer) soll in das Magazin abgelegt werden. Mit dem PI-Dienst TMMVTL wird die Leerplatzsuche für das Werkzeug aus Greifer 1 (Magazin 9998, Platz 2) ausgeführt und ein Auftrag zum Umsetzen des Werkzeugs erzeugt.

Tragen Sie im PLC-Programm die Parameter in DB1200 ein und setzen Sie in DB1200.DBX4000.0 das Start-Signal des PI-Dienstes.

PI-Dienste im NC-Bereich starten	Adresse	Signal	Werte
Parameter	DB1200.DBW4004	Werkzeugnummer	0
	DB1200.DBW4006	Quell-Platznummer	2
	DB1200.DBW4008	Quell-Magazinnummer	9998
	DB1201.DBW4010	Ziel-Platznummer	-1
	DB1201.DBW4012	Ziel-Magazinnummer	1
Auftrag	DB1200.DBB4001	PI-Index	5
	DB1200.DBX4000.0	Start	→ 1
Ergebnis	DB1200.DBB5000.0	Auftrag beendet	1
	DB1200.DBX5000.1	Fehler im Auftrag	0

Sie erhalten in DB410x.DBX0.0 einen Auftrag zum Umsetzen des Werkzeuges aus dem Zwischenspeicher. Der Zielplatz in Magazin 1 steht in DB4100.DBW12. Das PLC-Programm kann damit den notwendigen Ablauf ausführen.

9.8 Beispiel: Auftrag "Werkzeugwechsel vorbereiten" wiederholen

Ablauf: Auftrag wiederholen

Bei einer Fräsmaschine mit Toolboy und Shifter können sich Mitteilung des asynchronen Transfers und Auftrag kreuzen:

- Für das alte Werkzeug wurde mit der Ende-Quittung als Zielposition der Toolboy an die Werkzeugverwaltung quittiert.
- Das Teileprogramm wird mit dem neuen Werkzeug fortgesetzt und benötigt nach sehr kurzer Zeit wieder das vorherige (alte) Werkzeug.
- Die Werkzeugverwaltung generiert den nächsten Vorbereitungsauftrag zum Werkzeugwechsel mit dem Toolboy als Quellposition für das dann neue Werkzeug.
- In der gleichen Zeit wurde vom PLC-Programm das Werkzeug aus dem Toolboy in den Shifter übernommen.

- Das PLC-Programm teilt asynchron die Werkzeugbewegung vom Toolboy in den Shifter mit und beginnt mit dem Weitertransport ins Magazin.
- Im nächsten Takt erkennt das PLC-Programm den neuen Auftrag, das Werkzeug vom Toolboy in die Spindel einzuwechseln.

Im Toolboy befindet sich kein Werkzeug mehr. Das PLC-Programm muss diesen Fall erkennen (Überwachung: enthält Toolboy oder Shifter ein Werkzeug?). Es kann jetzt den Rücktransport des Werkzeugs ins Magazin abbrechen. Von der Werkzeugverwaltung kann dann eine Auftragswiederholung (Status 7) angefordert werden:

- Die Werkzeugverwaltung hat inzwischen die Mitteilung erhalten, dass sich das Werkzeug im Shifter befindet.
- Die Werkzeugverwaltung generiert einen neuen Vorbereitungsauftrag mit der Quellposition Shifter für das neue Werkzeug.

Hinweis

Die Quittung zur Wiederholung eines Auftrags "Werkzeugwechsel vorbereiten" darf nur **vor** der Ende-Quittung des Auftrags "Werkzeugwechsel vorbereiten" erfolgen!

Quittung an Werkzeugverwaltung:

Quittungsschritt	Quittungsbit	Transferschritt neues Werkzeug	Transferschritt altes Werkzeug	Status
xxx	DB4000.DBXx.x	0	0	7

Anwender-Beispiel für Fräsmaschine 10

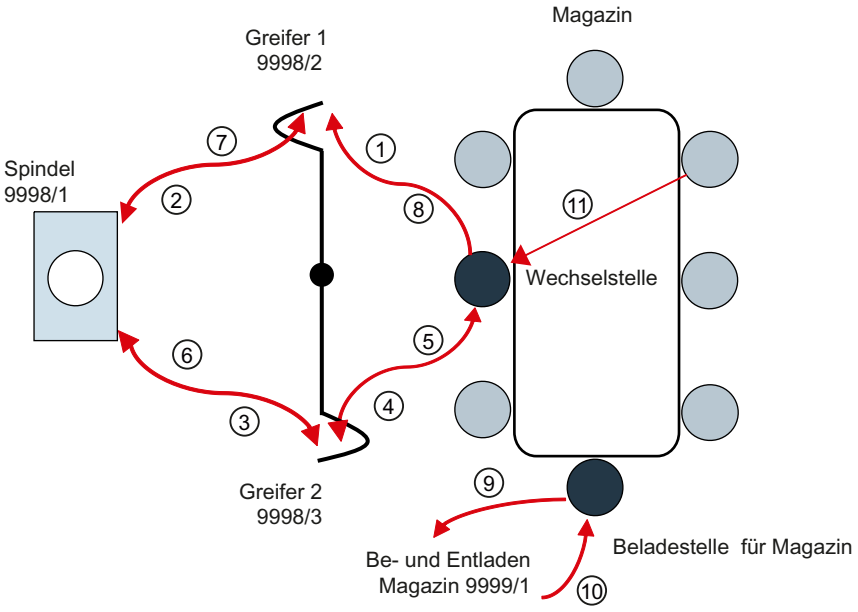
10.1 Beispiel: Fräsmaschine mit Kettenmagazin und Doppelgreifer (MAG_CONF_MPF)

Beispiel-Datei

Das Programm zur Magazinkonfiguration finden Sie auf der Toolbox.

Das Programm kann in die Steuerung eingelesen werden und ist an die konkrete Anlage anzupassen.

Konfiguration



1 ... 11 Transferschritte

Bild 10-1 Fräsmaschine mit Kettenmagazin

Bezeichnung der Zwischenspeicher und Beladestellen:

Magazin	Platz	Bedeutung
1	xx	reales Magazin (Kette, Teller, Fläche), Position xx
9998	1	Spindel
9998	2	Greifer 1
9998	3	Greifer 2
9999	1	Beladestelle für Magazin

Beschreibung des Programms

Zu Beginn werden alle alten Magazindefinitionen und Werkzeuge gelöscht. Im weiteren Programmablauf werden alle Magazine und Zwischenspeicher durch Schreiben der Magazinparameter neu erzeugt und definiert.

In N70 kann die Suchstrategie für Werkzeuge und für Magazinplätze ausgewählt werden. Hier wird festgelegt ob ein Werkzeug bei Wechsel aus der Spindel direkt auf den Platz des Neuwerkzeugs abgelegt werden kann. Damit ist der Werkzeugwechsel in einem mechanischen Ablauf möglich, es können so kürzere Wechselzeiten erreicht werden. An Pickup-Magazinen ist dieser Ablauf nicht möglich.

Die Plätze des Kettenmagazins werden ab N430 bis N500 eingerichtet. Magazin Platztyp = 0 bedeutet das der Magazinplatz mit Werkzeugen unterschiedlicher Platztypen beladen werden kann.

Ab N520 werden die Zwischenspeicher eingestellt. Sind zusätzliche Zwischenspeicher vorhanden (Toolboy, Shifter ...) ist die Anzahl in N50 zu ändern.

Analog dazu ist bei zusätzlich vorhandenen Beladestellen in N60 zu verfahren.

Ab N920 wird die Zuordnung der Zwischenspeicher zur Spindel/Werkzeughalter und dem Magazin vorgenommen.

Nachdem das Programm zur Magazinkonfiguration durchlaufen ist, führen Sie einen Neustart der NC durch: Reset (po).

Siehe auch

Weitere Informationen:

- Die Konfiguration eines Tellermagazins mit Festplatz-Codierung erfolgt wie die Konfiguration einer Revolvermagazins: Beispiel: Drehmaschine mit Revolvermagazin (MAG_CONF_MPF) (Seite 75)
- Eine genaue Beschreibung der verwendeten Parameter finden Sie im Funktionshandbuch Werkzeugverwaltung der SINUMERIK 840D sl.

Beispiel MAG_CONF_MPF

1. Anlagenkonfiguration:
   - 1 Kettenmagazin mit 8 Plätzen (in N40 einstellbar)
   - 1 Beladestelle
   - 3 Zwischenspeicherplätze (in N50 einstellbar, ab N540 Zuordnungen)
2. Teileprogramm:

```
;MAG_CONF_MPF
N10 def int NUM_MAG,MAG_TYPE, LOCATIONS, PLA-
CE, NUM_BUFFER, NUM_LOAD, PLACE_SEARCH
;
N20 NUM_MAG = 1                                ;Nummer des  Magazins
```

```
N30 MAG_TYPE = 1                                ;Magazinart (1: Kette, 3: Revolver,
                                                5: Flächenmagazin)
N40 LOCATIONS = 8                               ;Anzahl der Magazinplätze
N50 NUM_BUFFER = 3                              ;Anzahl der Zwischenspeicher (Spin-
                                                del, Greifer)
N60 NUM_LOAD = 1                                ; Anzahl der Beladestellen
N70 PLACE_SEARCH = 12289                        ; Art der Suchstrategie
;= 257 Bit13=0 kein Tausch Altwerkzeug auf Platz von Neuwerkzeug
;Einstellung für Pickup Magazin
;=12289 Bit13=1 Tausch Altwerkzeug auf Platz von Neuwerkzeug
;Einstellung für Kettenmagazin
N80;
N90;
;Parameter prüfen
N100 STOPRE
N110 if ((NUM_MAG==0)or(LOCATIONS==0))
N120  Err1:STOPRE
N130  MSG("Wrong Parameter --> Cancel")
N140  G04 F4
N150  STOPRE
N160  M0
N170  GOTOB Err1
N180 endif
N190; Magazin Konfiguration
N200;
N210;
N220; Lösche alte Daten wenn, Magazin 1 angelegt wird
N230 if NUM_MAG ==1
N240 $TC_MAP1[0]=0                              ; lösche Magazine
N250 $TC_DP1[0,0]=0                             ; lösche Werkzeuge
N260 STOPRE
N270 endif
; Konfiguration
;
N280 $TC_MAMP2= PLACE_SEARCH                    ; Art der Suchstrategie
;
; Magazin
; Magazin einrichten
N290 $TC_MAP1[NUM_MAG]= MAG_TYPE
N300 $TC_MAP2[NUM_MAG]="MAGAZIN"<<NUM_MAG
N310 if MAG_TYPE == 3
N320 $TC_MAP3[NUM_MAG]=81                       ; Magazinzustand, alle Plätze fest-
                                                platz-codiert bei Revolvermagazin
N330 else
N340 $TC_MAP3[NUM_MAG]=17                       ; Magazinzustand
```

```
N350 endif
N360 $TC_MAP4[NUM_MAG]=-1
N370 $TC_MAP5[NUM_MAG]=-1
N380 $TC_MAP6[NUM_MAG]=1                          ; Anzahl Zeilen des Magazins
N390 $TC_MAP8[NUM_MAG]=0
N400 $TC_MAP9[NUM_MAG]=0
N410 $TC_MAP7[NUM_MAG]=LOCATIONS                  ; Anzahl Magazinplätze
N420 $TC_MAP10[NUM_MAG]=PLACE_SEARCH
;
; Magazinplätze
N430 for PLACE=1 to LOCATIONS
N440   STOPRE
N450   $TC_MPP1[NUM_MAG,PLACE]=1                  ; Platzart
N460   $TC_MPP2[NUM_MAG,PLACE]=0                  ; Platzytp,0 mit jedem WZ Platztyp
                                                  verträglich
N470   $TC_MPP3[NUM_MAG,PLACE]=1                  ; Nebenplatzbetrachtung ein (aus wä-
                                                  re 0)
N480   $TC_MPP4[NUM_MAG,PLACE]=2                  ; Platzzustand
N490   $TC_MPP5[NUM_MAG,PLACE]=PLACE               ; Platzartindex
N500 endfor
N510 STOPRE
;
N520; Definition Zwischenspeichermagazin (immer Nummer 9998)
;
N530 $TC_MAP1[9998]=7                             ; Magazinart: 7: Zwischenspeicher
N540 $TC_MAP2[9998]="BUFFER"<<NUM_MAG
N550 $TC_MAP3[9998]=17                            ; Magazinzustand
N560 $TC_MAP6[9998]=1                             ; Anzahl Zeilen
N570 $TC_MAP7[9998]=NUM_BUFFER                    ; Anzahl Plätze
;
; Plätze des Zwischenspeichers
; Spindel
N580 $TC_MPP1[9998,1]=2                           ; Platzart (hier Spindel)
N590 $TC_MPP2[9998,1]=0                           ; Platztyp (hier immer 0)
N600 $TC_MPP3[9998,1]=0                           ; Nebenplatzbetrachtung aus
N610 $TC_MPP4[9998,1]=2                           ; Platzzustand
N620 $TC_MPP5[9998,1]=1                           ; Platzartindex
;
N630; Greifer
N640 FOR PLACE=2 to NUM_BUFFER
N650     STOPRE
N660     $TC_MPP1[9998,PLACE]=3                   ;(hier Greifer)
N670     $TC_MPP2[9998,PLACE]=0                   ;(hier immer 0)
N680     $TC_MPP3[9998,PLACE]=0                   ;Nebenplatzbetrachtung aus
N690     $TC_MPP4[9998,PLACE]=2                   ;Platzzustand
```

```
N700     $TC_MPP5[9998,PLACE]=PLACE             ;Platzartindex
N710 endfor
N720 STOPRE
;
;
N730; Definition Beladem agazin (immer Nummer 9999)
;
N740 $TC_MAP1[9999]=9                           ; Magazinart: 9: Beladem agazin
N750 $TC_MAP2[9999]="BELADEMAGAZIN"<<NUM_MAG
N760 $TC_MAP3[9999]=17                          ; Magazinzustand
N770 $TC_MAP4[9999]=-1
N780 $TC_MAP5[9999]=-1
N790 $TC_MAP6[9999]=1                           ; Anzahl Zeilen
N800 $TC_MAP7[9999]=NUM_LOAD                    ; Anzahl Plätze
N810 STOPRE;
;
N820; Beladem agazinplätze
;
N830 for PLACE=1 to NUM_LOAD
N840     STOPRE
N850     $TC_MPP1[9999,PLACE]=7                 ; Platzart Beladestelle
N860     $TC_MPP2[9999,PLACE]=0                 ; Platztyp (hier immer 0)
N870     $TC_MPP3[9999,PLACE]=0                 ; Nebenplatzbetrachtung aus
N880     $TC_MPP4[9999,PLACE]=2                 ; Platzzustand
N890     $TC_MPP5[9999,PLACE]=PLACE             ; Platzartindex
N900 endfor
N910 STOPRE
;
;
N920; Offsets (Abstände) ; Abstände zum Magazin
;
; Zwischenspeicher
N930 for PLACE=1 to NUM_BUFFER
N940  $TC_MDP2[1,PLACE]=0
N950 endfor
N960 STOPRE
;
; Beladestellen
N970 for PLACE=1 to NUM_LOAD
N980    stopre
N990    $TC_MDP1[1,PLACE]=0
N1000 endfor
N1010 M30                                       ; Ende
```

Anzeige in der Bediensoftware

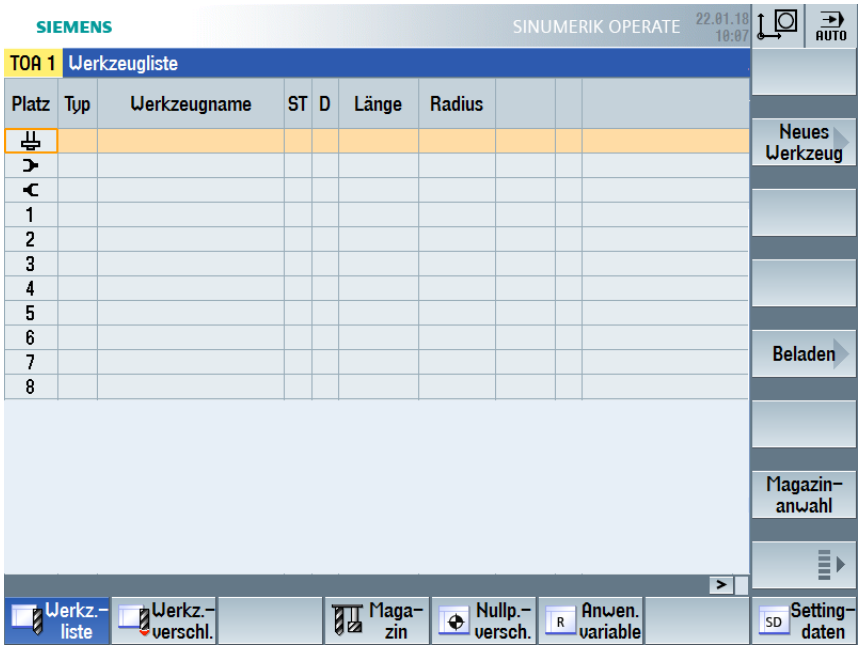


Bild 10-2 Werkzeugliste Fräsmaschine

10.2 Ablaufdiagramm: Werkzeugwechsel

Programmablauf Werkzeugwechsel (PLC)

Der hier beschriebene Ablauf beschreibt den Wechsel zwischen Magazin und Spindel. Der Wechsel von Handwerkzeugen sowie Be- und Entladen sind nicht berücksichtigt. Diese Abläufe sind beschrieben in:

- Beispiel: Beladen / Entladen (Seite 69)
- Beispiel: Handwerkzeuge wechseln (Seite 71)

Die Voreinstellung der Maschinendaten ist so gewählt, dass bei Programmierung mit dem T-Befehl an der Nahtstelle der Auftrag für "Werkzeugwechsel vorbereiten" ausgelöst wird:

```
N10 T = "Werkzeugname" M6
```

Der Satzvorlauf wird nicht unterbrochen. Gleichzeitig wird mit M6 das Werkzeugwechsel-Unterprogramm gestartet (L6). Sobald der Auftrag für "Werkzeugwechsel vorbereiten" quittiert wurde und im Werkzeugwechsel Unterprogramm M-Code für Werkzeugwechsel ausgegeben erreicht ist, wird an der Nahtstelle der Auftrag "Werkzeugwechsel durchführen" ausgegeben (Satz-Splitting).

Einem Werkzeugwechselbefehl (M206) muss immer ein Befehl zum Werkzeugwechsel vorbereiten vorangehen. Ein Wechselbefehl ohne vorherigen Auftrag "Werkzeugwechsel vorbereiten" löst keinen Auftrag der Werkzeugverwaltung aus.

Im Programmablauf finden Sie den Ausdruck 1:1 Wechsel. Das bedeutet, dass der Werkzeugwechsel in einem Zyklus durchgeführt wird. Das Werkzeug aus der Spindel

(Altwerkzeug) wird auf dem Magazinplatz des Neuwerkzeugs abgelegt. Es ist keine zusätzliche Magazinpositionierung notwendig. In diesem Fall ist im Auftrag der Werkzeugverwaltung der Zielplatz des Altwerkzeugs gleich dem Quellplatz des Neuwerkzeugs (DB43xx.DBW6 und DBW8 sind gleich DBW18 und DBW20).

Ein 1:1 Wechsel ist nicht möglich bei:

- unterschiedlichen Platztypen der Werkzeuge
- unterschiedlichen Werkzeuggrößen
- Festplatzwerkzeugen

Mit Programmierung von T0 im NC-Teileprogramm wird ein Werkzeugwechsel ohne Neuwerkzeug angestoßen. Es wird nur das Spindelwerkzeug ins Magazin transportiert (Spindel leer fahren).

① ... ⑭ bezeichnen die Schritte im folgenden Ablaufdiagramm.

Sie können anhand der Schritte die unterschiedlichen Quittungsmöglichkeiten sehen. Es kommen sowohl auftragsbezogene Quittungen als auch asynchrone Mitteilungen zum Einsatz. Die programmierten Transferschritte entnehmen Sie bitte der Tabelle in Kapitel Beispiel: Quittungsschritte (Fräsmaschine) (Seite 101)

Ablaufdiagramm "Werkzeugwechsel"

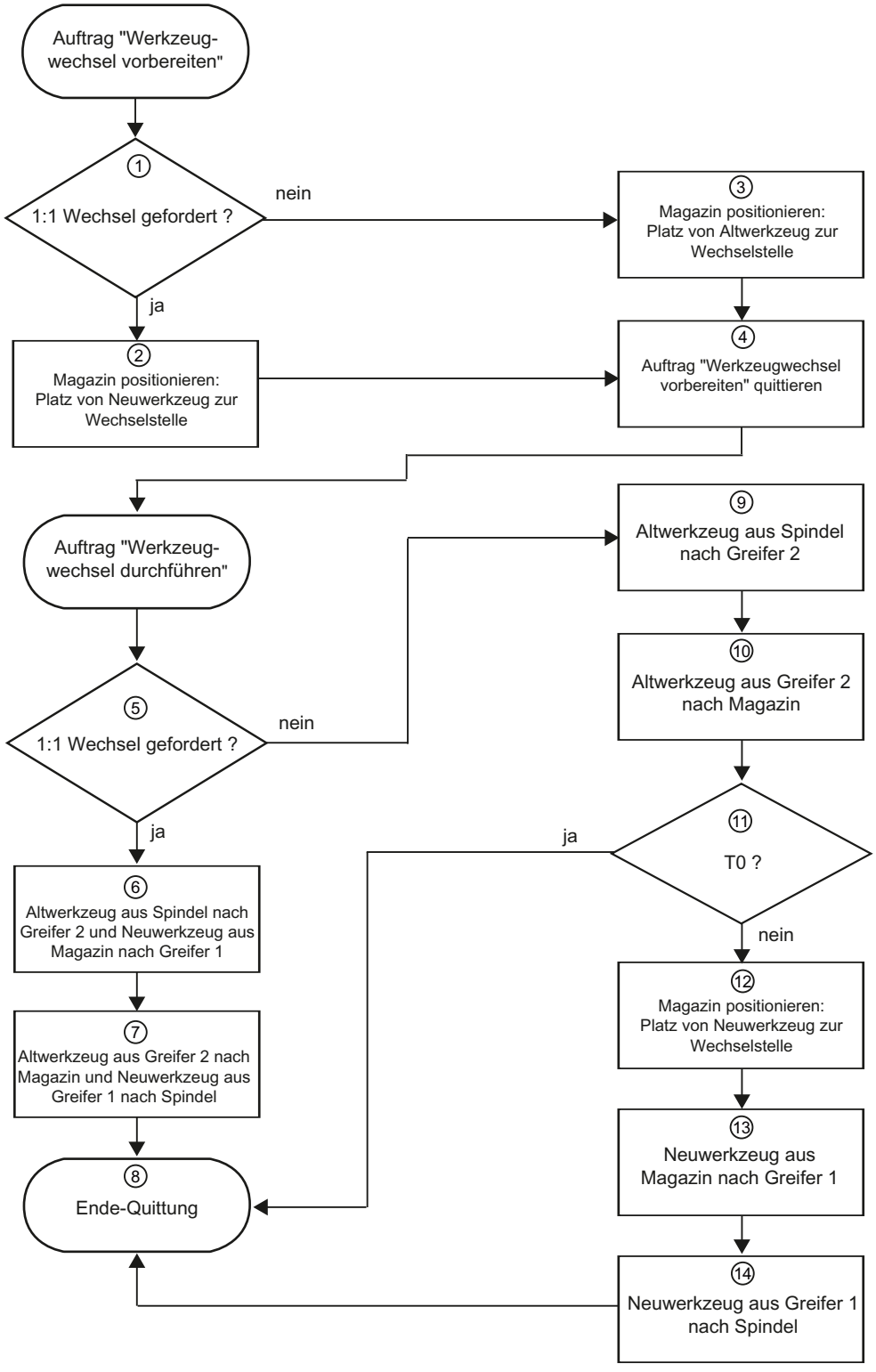


Bild 10-3 Ablaufdiagramm

Beschreibung des Ablaufs:

- NC-Teileprogramm:
  T-Befehl oder T-Befehl mit gleichzeitigem WZW-Aufruf (M6)
- Nahtstellensignale:
  Auftrag von WZV: DB43xx.DBX0.0 (Auftragsbit) und
  DB43xx.DBB1 (Kommandobits): Wechsel vorbereiten (DB43xx.DBX1.2)
- Magazinnummer und Platznummer der zu bewegenden Werkzeuge:
  DB43xx.DBW6 bis DBW20: Quellplatz neues Werkzeug, Zielplatz altes Werkzeug

Schritt 1: 1:1-Wechsel gefordert

Abfrage: Quellplatz neues Werkzeug == Zielplatz altes Werkzeug

- Normalfall: Es befindet sich ein Werkzeug im Werkzeughalter und es wird ein neues Werkzeug angefordert. Beide Werkzeuge haben den gleichen Platztyp und gleiche Werkzeuggröße in der Magazinliste, die Werkzeuge sind nicht festplatzcodiert.
  Das alte Werkzeug wird in direktem Wechsel auf dem Magazinplatz des neuen Werkzeugs abgelegt (1:1-Wechsel). Ist kein altes Werkzeug im Werkzeughalter (DB4300.DBX1.4), erfolgt der gleiche Ablauf. Das Magazin wird in diesem Fall das neue Werkzeug zur Wechselstelle positionieren.
  → weiter mit Schritt 2

Abfrage: Quellplatz neues Werkzeug >< Zielplatz altes Werkzeug, Sonderfälle:

- Über DB43xx.DBX1.3 wird von der Werkzeugverwaltung mitgeteilt, ob das aktive Werkzeug aus dem Werkzeughalter entfernt werden soll (T0). In diesem Fall wird das Magazin den Ablageplatz des alten Werkzeugs (das noch in der Spindel ist) zur Wechselstelle positionieren. Der Magazinplatz für das Altes Werkzeug wird auch dann zur Wechselstelle positioniert, wenn das alte Werkzeug nicht auf dem Platz des neuen Werkzeugs abgelegt werden kann (1:1-Wechsel nicht möglich). Ursache können unterschiedliche Platztypen oder Werkzeuggrößen sein, oder es sind festplatzcodierte Werkzeuge beteiligt. In diesem Fall wird der Werkzeugwechsel in zwei Schritten durchgeführt. Zuerst wird das alte Werkzeug in das Magazin abgelegt und danach das neue Werkzeug zur Spindel transportiert.
  → weiter mit Schritt 3

Schritt 2: Magazin positionieren, Platz von neuem Werkzeug zur Wechselstelle

← vorheriger Schritt: Schritt 1

Die Magazinbewegung kann z. B. über eine von der PLC-gesteuerte NC-Rundachse erfolgen. Die Bewegung sollte an die Werkzeugverwaltung gemeldet werden. Die Position des Magazins wird damit an der Bedienoberfläche in den Ansichten der Werkzeug- und Magazinliste aktualisiert.

Quittung an WZV:

Von DB4300.DBW8 (Platznummer für neues Werkzeug - Quelle) wird die Zielposition gelesen und nach DB9901.DBW2 (Transferschritt 101) geschrieben. Bei Magazin-Koinzidenz an der Zielposition wird der Schritt asynchron quittiert:

Quittungsschritt	Quittungsbit	Transferschritt neues Werkzeug	Transferschritt altes Werkzeug	Status
4	DB4200.DBX0.4	101	0	204
Der variable Platz im Magazin 1 wurde an die Wechselstelle zur Spindel positioniert.				

Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
101 DB9901.DBW0	1	n	9998	1
n ist hier die tatsächliche Platznummer (n ≠ 0), die vom PLC-Programm in die variable Transfertabelle eingetragen wird.				

→ weiter mit Schritt 4

Schritt 3: Magazin positionieren, Platz von altem Werkzeug zur Wechselstelle

← vorheriger Schritt: Schritt 1

Analog zu Schritt 2, jedoch wird der Zielplatz des Magazins hier von DB4300.DBW20 (Platznummer für altes Werkzeug - Ziel) bezogen.

Quittung an WZV:

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
4	DB4200.DBX0.4	101	0	204
Der variable Platz im Magazin 1 wurde an die Wechselstelle zur Spindel positioniert.				

Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
101 DB9901.DBW0	1	n	9998	1

n ist hier die tatsächliche Platznummer (n ≠ 0), die vom PLC-Programm in die variable Transfertabelle eingetragen wird.

→ weiter mit Schritt 4

Schritt 4: Auftrag Werkzeugwechsel vorbereiten und quittieren

← vorheriger Schritt: Schritt 2 oder Schritt 3

Quittung an WZV:

- NC Programm:
  M206 löst den Auftrag für WZW durchführen aus
- Nahtstellensignale:
  Auftrag von WZV: DB43xx.DBX0.0 (Auftragsbit)
  DB43xx.DBB1 (Kommandobits): Wechsel durchführen (DB43xx.DBX1.1)
- Magazinnummer und Platznummer der zu bewegenden Werkzeuge:
  DB43xx.DBW6 bis DBW20: Quellplatz neues Werkzeug, Zielplatz altes Werkzeug

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
1	DB4200.DBX0.1	0	0	1
An vielen Systemen ist damit die Vorbereitung des Werkzeugwechsels abgeschlossen.				

→ weiter mit Schritt 5

Schritt 5: 1:1 Wechsel gefordert

← vorheriger Schritt: kein vorheriger Schritt, Einsprung in die Schrittkette bei Auftrag "Werkzeugverwaltung durchführen"

Analog zu Schritt 1 wird geprüft, ob ein direkter Wechsel oder ein Wechsel in zwei Schritten durchgeführt wird:

- 1:1 Wechsel möglich: **→ weiter mit Schritt 6**
- 1:1 Wechsel nicht möglich: **→ weiter mit Schritt 9**

Schritt 6: Altes Werkzeug aus Spindel nach Greifer 2 und neues Werkzeug aus Magazin nach Greifer 1

← vorheriger Schritt: Schritt 5

Das PLC-Programm steuert die Maschinenfunktionen, mit denen die Greiferbewegungen, Werkzeugklemmung usw. stattfinden. Sind die mechanischen Bewegungen abgeschlossen und im PLC-Programm quittiert, erfolgt die Quittung der Werkzeugbewegungen an die Werkzeugverwaltung.

Quittung an WZV:

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
5	DB4200.DBX0.5	1	2	105

Neues Werkzeug von Magazin nach Greifer 1 → Schritt 6 oder 13

Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
1 DB9900.DBW0	0	1	9998	2

Altes Werkzeug von Spindel nach Greifer 2 → Schritt 6 oder 9

Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
2 DB9900.DBW8	9998	1	9998	3

→ weiter mit Schritt 7

Schritt 7: Altes Werkzeug aus Greifer 2 nach Magazin und neues Werkzeug aus Greifer 1 nach Spindel

← vorheriger Schritt: Schritt 6

Das PLC-Programm steuert die Maschinenfunktionen, für die Greiferbewegungen, Werkzeugklemmung usw. Sind die mechanischen Bewegungen abgeschlossen und im PLC-Programm quittiert, erfolgt die Quittung der Werkzeugbewegungen an die Werkzeugverwaltung.

Quittung an WZV:

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
7	DB4200.DBX0.7	3	4	105

Neues Werkzeug von Greifer 1 nach Spindel → Schritt 7 oder 14

Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
3 DB9900.DBW16	9998	2	9998	1
Altes Werkzeug von Greifer 2 nach Magazin → Schritt 7 oder 10				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
4 DB9900.DBW24	9998	3	0	2

→ weiter mit Schritt 8

Schritt 8: Ende-Quittung

← vorheriger Schritt: Schritt 7 oder Schritt 14

Mit Werkzeugwechsler in Grundstellung oder einem Zustand, in dem die Maschine die Bearbeitung fortsetzen kann, erfolgt die Ende-Quittung. Es sind hier eventuell noch mechanische Bewegungen durchzuführen, bevor der Werkzeugwechsel beendet werden kann.

Quittung an WZV:

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
1	DB4200.DBX0.1	0	0	1

→ weiter mit Schritt 9

Schritt 9: Altes Werkzeug aus Spindel nach Greifer 2

← vorheriger Schritt: Schritt 5

Das PLC-Programm steuert die Maschinenfunktionen, mit denen die Greiferbewegungen, Werkzeugklemmung usw. stattfinden. Sind die mechanischen Bewegungen abgeschlossen und im PLC-Programm quittiert, erfolgt die Quittung der Werkzeugbewegungen an die Werkzeugverwaltung.

Quittung an WZV:

Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
8	DB4200.DBX1.0	0	2	105
Altes Werkzeug von Spindel nach Greifer 2 → Schritt 6 oder 9				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
2 DB9900.DBW8	9998	1	9998	3

→ weiter mit Schritt 10

Schritt 10: Altes Werkzeug aus Greifer 2 nach Magazin

← vorheriger Schritt: Schritt 9

Das PLC-Programm steuert die Maschinenfunktionen, mit denen die Greiferbewegungen, Werkzeugklemmung usw. stattfinden. Sind die mechanischen Bewegungen abgeschlossen und im PLC-Programm quittiert, erfolgt die Quittung der Werkzeugbewegungen an die Werkzeugverwaltung. Quittung an WZV:				
Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
9	DB4200.DBX1.1	0	4	105
Altes Werkzeug von Greifer 2 nach Magazin → Schritt 7 oder 10				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
4 DB9900.DBW24	9998	3	0	2

→ weiter mit Schritt 11

Schritt 11: T0 ?

← vorheriger Schritt: Schritt 10

Abfrage: ist im Werkzeugwechselauftrag T0 gesetzt? → DB43xx.DBX1.3

- Soll nur der Werkzeughalter leer gefahren werden, kann der Werkzeugwechsel beendet werden. **→ weiter mit Schritt 8**
- Soll ein neues Werkzeug zum Werkzeugträger gebracht werden? **→ weiter mit Schritt 12**

Schritt 12: Magazin positionieren Platz von neuem Werkzeug zur Wechselstelle

← vorheriger Schritt: Schritt 11

Ablauf wie Schritt 2 Quittung an WZV:				
Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
4	DB4200.DBX0.4	101	0	204
Der variable Platz im Magazin 1 wurde an die Wechselstelle zur Spindel positioniert.				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
101 DB9901.DBW0	1	n	9998	1

n ist hier die tatsächliche Platznummer (n ≠ 0), die vom PLC-Programm in die variable Transfertabelle eingetragen wird.

→ weiter mit Schritt 13

Schritt 13: Neues Werkzeug aus Magazin nach Greifer 1

← vorheriger Schritt: Schritt 12

Das PLC-Programm steuert die Maschinenfunktionen, mit denen die Greiferbewegungen, Werkzeugklemmung usw. stattfinden. Quittung an WZV:				
Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
10	DB4200.DBX1.2	1	0	105
Neues Werkzeug von Magazin nach Greifer 1 → Schritt 6 oder 13				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
1 DB9900.DBW0	0	1	9998	2

→ weiter mit Schritt 14

Schritt 14: Neues Werkzeug aus Greifer 1 nach Spindel

← vorheriger Schritt: Schritt 13

Das PLC-Programm steuert die Maschinenfunktionen, mit denen die Greiferbewegungen, Werkzeugklemmung usw. stattfinden. Sind die mechanischen Bewegungen abgeschlossen und im PLC-Programm quittiert, erfolgt die Quittung der Werkzeugbewegungen an die Werkzeugverwaltung. Der Werkzeugwechsel kann beendet werden. Quittung an WZV:				
Quittungsschritt	**Quittungsbit**	**Transferschritt neues Werkzeug**	**Transferschritt altes Werkzeug**	**Status**
11	DB4200.DBX1.3	3	0	105
Neues Werkzeug von Greifer 1 nach Spindel → Schritt 7 oder 14				
Transferschritt	**von**		**nach**	
	Magazin	**Platz**	**Magazin**	**Platz**
3 DB9900.DBW16	9998	2	9998	1

→ weiter mit Schritt 8

10.3 Beispiel: Quittungsschritte (Fräsmaschine)

Konstante Transferschritt-Tabelle

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
1 DB9900.DBW0	0	1	9998	2	Neues Werkzeug von Magazin nach Greifer 1 Schritt ⑥ oder ⑬
2 DB9900.DBW8	9998		9998	3	Schritt ⑥ oder ⑨
3 DB9900.DBW16	9998		9998	1	Schritt ⑦ oder ⑩
4 DB9900.DBW24	9998	3	0	2	Schritt ⑦ oder ⑭
5 DB9900.DBW32	0	2	9998	1	Ablageplatz des alten Werkzeugs
6 DB9900.DBW40	0	1	9998	1	Magazinplatz mit dem neuen Werkzeug zur Wechselstelle Schritt ② oder ⑫
7 DB9900.DBW48	--	--	--	--	

Variable Transferschritt-Tabelle

Transferschritt	von		nach		Bemerkung
	Magazin	Platz	Magazin	Platz	
101 DB9901.DBW0	1	n	9998	1	Der variable Platz im Magazin 1 wurde an die Wechselstelle zur Spindel positioniert.
102 DB9901.DBW8	--	--	--	--	

n: ist hier die tatsächliche, vom PLC-Programm in die variable Transfertabelle einzutragende Platznummer ($n \neq 0$).

Quittungsschritt-Tabelle

Quittungsschritt	Transferschritt		Quittungs-status	Bemerkung
	Altes Werkzeug	Neues Werkzeug		
1 DB9902.DBW0	0	0	1	Ende-Quittung, Schritt ④ und ⑧
2 DB9902.DBW4	0	0	3	Auftrag abbrechen

Quittungsschritt	Transferschritt		Quittungs-status	Bemerkung
	Altes Werk-zeug	Neues Werkzeug		
3 DB9902.DBW8	0	0	105	Zwischenquittung bei Folgeauf-trag, Schritt ④
4 DB9902.DBW12	101	0	204	Variabler Magazinplatz zur Wech-selstelle
5 DB9902.DBW16	1	2	105	Zwischenquittung Schritt ⑥
6 DB9902.DBW20	0	5	105	Zwischenquittung Schritt ③
7 DB9902.DBW24	3	4	105	Zwischenquittung Schritt ⑦
8 DB9902.DBW28	0	2	105	Zwischenquittung Schritt ⑨
9 DB9902.DBW32	0	4	105	Zwischenquittung Schritt ⑩
10 DB9902.DBW36	1	0	105	Zwischenquittung Schritt ⑬
11 DB9902.DBW40	3	0	105	Zwischenquittung Schritt ⑭
12 DB9902.DBW44	--	--	--	

Hinweis: Die Schrittnummern ① ... ⑭ beziehen sich auf das Ablaufdiagramm im Kapitel Ablaufdiagramm: Werkzeugwechsel (Seite 92)

10.4 Beispiel: Werkzeugwechselzyklus für Fräsmaschine

Programmbeispiel

```
PROC L6 SAVE DISPLOF
;---------------------------------------------------------------
; Beispiel Werkzeugwechselzyklus für Maschinenhersteller
;---------------------------------------------------------------
DEF INT _WZ_IN_SP,_WZ_VOR
DEF REAL _SPP= ...                        ; Spindelposition
;
  IF(NOT $P_SEARCH)                       ; wenn kein Satzsuchlauf
    _WZ_IN_SP=$TC_MPP6[9998,1]            ; Werkzeug in der Spindel
    GETSELT(_WZ_VOR)                      ; vorher angewähltes Werkzeug
;
    IF(_WZ_IN_SP<>_WZ_VOR)                ; wenn anderes Werkzeug
```

```
      SPOS=_SPP                          ; Spindel positionieren
      G0                                 ; Werkzeugwechselposition anfahren
      G75 Z=0
      WAITS(1)
    ENDIF
  ELSE
  ENDIF
;
; Werkzeug einwechseln: Werkzeugverwaltung und PLC
M206
M17
;-------------------------------------------------------------------
; ENDE
;-------------------------------------------------------------------
```

NC-Programmierung 11

11.1 Übersicht der BTSS und Systemvariablen

Grundlagen

Alle Daten, die für die Werkzeugverwaltung notwendig sind (um ein Magazin zu definieren oder ein Werkzeug zu beladen...), liegen in der NC. Die Daten können über Teileprogramme mit Systemvariablen und über PLC mit FB2 und FB3 gelesen bzw. geschrieben werden. Der Anwender (Maschinenhersteller) sollte bei der Projektierung der Maschine prüfen, ob es günstiger ist, die Daten der WZV in der PLC, NC oder in einem ASUP zu lesen bzw. zu schreiben.

Auf die Systemvariablen kann in der Regel lesend und schreibend zugegriffen werden.

Gegebenenfalls ist bei Verwendung von Sprachbefehlen die Programmierung des Befehls "STOPRE" erforderlich.

Die $TC-Variablen erzeugen keinen Vorlaufstopp.

Der Bezeichner eines Werkzeugs darf aus folgenden Zeichen bestehen:

a...z

A...Z

0...9

+ - _ . ,

Groß- und Kleinbuchstaben gelten als unterschiedliche Buchstaben.

Hinweis

Zusätzliche Informationen zu den BTSS-Variablen sind in der Hilfedatei zum NC-VAR-Selektor enthalten.

Ausführen der Befehle

Durch das erstmalige Schreiben von Daten für Werkzeuge oder Magazine werden die entsprechenden Objekte mit ihren Default-Daten angelegt. Beispiele dafür sind:

- $TC_TP1 legt ein Werkzeug an.
- $TC_MAP1[x] legt ein Magazin an.

Hinweis

Nicht freigeschaltete/aktive Funktionen

Werden Daten von nicht freigegebenen Funktionen geschrieben, wird KEIN Alarm ausgegeben und der Schreibbefehl wird nicht ausgeführt.

Beim Versuch Daten von nicht aktiven Funktionen zu lesen, wird ein Alarm ausgegeben.

Übersicht

Das folgende Bild zeigt eine Übersicht aller Schneiden-, Werkzeug- und Magazindaten ($TC_...) bei aktiver Werkzeugverwaltung.

Bemerkung:

Die dargestellte Reihenfolge der Systemvariablen entspricht der BTSS-Reihenfolge bezüglich der Nummerierung.

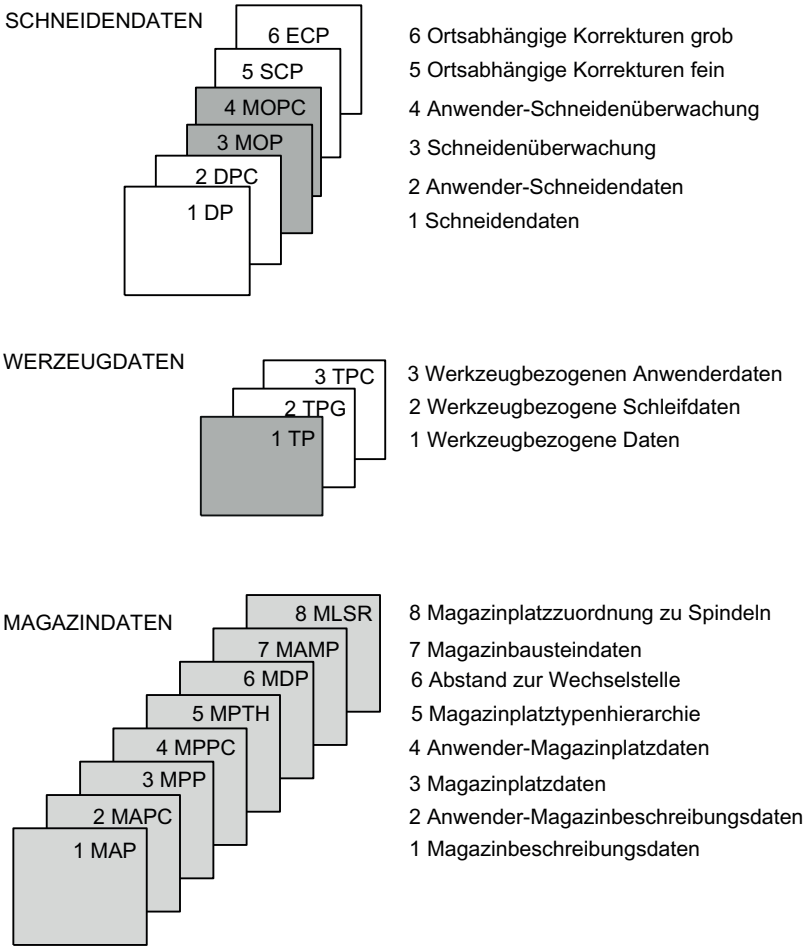


Bild 11-1 Übersicht Schneiden-, Werkzeug- und Magazindaten

Die Bezeichner (DP,...TP,...MOP,...) sind aus der NC-Sprache entlehnt. Sie sind Namensbestandteile der Systemvariablen $TC_DP,...

Hinweis

Alle grauen Datenfelder sind nur bei aktiver Werkzeugverwaltung vorhanden.

Dunkelgraue Datenfelder sind auch ohne WZV, aber mit Überwachungsfunktion vorhanden.

Weiße Datenfelder sind auch ohne aktive WZV vorhanden.

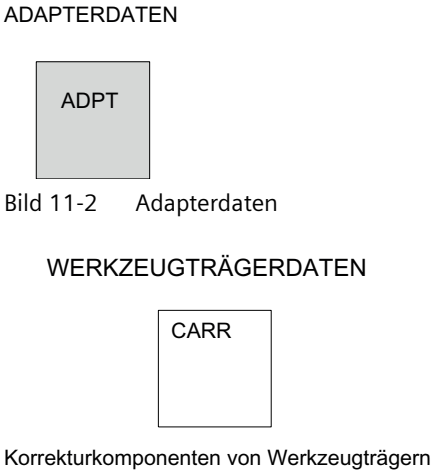


Bild 11-2 Adapterdaten

Bild 11-3 Werkzeugträgerdaten

11.2 Verwendung der Magazindaten

Reihenfolge der Datendefinition

Durch den Vorgang "Werkzeug einem Magazinplatz zuordnen" wird eine Abhängigkeit zwischen den Werkzeug-Daten und den Magazin-/Magazinplatzdaten geschaffen.

Beispiel:

Das Werkzeug enthält den Magazinplatztyp, für den es vorgesehen ist. Der Magazinplatz enthält den eigenen Magazinplatztyp. Wenn nun das Werkzeug dem Magazinplatz zugeordnet wurde, so kann im Allgemeinen der Platztyp nicht mehr verändert werden, da dies zu Inkonsistenzen führen würde.

Daraus ergibt sich die Forderung, dass Werkzeuge, Magazine über einen speziellen Vorgang in die Steuerung geladen werden, und während der Verarbeitung die Struktur bestimmenden Definitionen nicht mehr geändert werden dürfen (das sind z. B. Magazindimension, Magazinplatztyp, Duplo-Nr., Werkzeug-Name, ...). Das sind nicht: Schneidendaten, Magazinplatz-, Werkzeug-Zustand, ...

Verwendung von Magazindaten

Aktion	Programm-Befehl	Beschreibung
Neues Magazin anlegen	`$TC_MAPx[y] = Wert;`	Wert <>0, y = Magazin-Nr. eines noch nicht angelegten Magazins
Löschen eines Magazins	`$TC_MAP1[y] = 0;`	Die Daten des Magazins und seiner Magazinplätze, sowie eventuell definierte Abstände zu Wechselstellen werden gelöscht. Der damit verbundene Speicher wird frei gegeben.
Löschen eines Magazins und der darin enthaltenen Werkzeuge	`$TC_MAP6[y] = 0;`	Die Daten des Magazins und seiner Magazinplätze, sowie eventuell definierte Abstände zu Wechselstellen werden gelöscht. Enthaltene Werkzeuge werden mitgelöscht. Der damit verbundene Speicher wird frei gegeben
Löschen aller Magazine	`$TC_MAP1[0] = 0;`	Alle Daten aller Magazine der angewählten TO-Bereichseinheit werden gelöscht und der zugehörige Speicher wird frei gegeben. Der Magazindatenbaustein ist anschließend leer.
Neuen Magazinplatz anlegen	`$TC_MPPx[y,z] = Wert;`	Wert <>0, y = Platznummer noch nicht vorhanden. Vor dem Anlegen der Daten des ersten Platzes muss das zugehörige Magazin definiert worden sein. Wenn der erste Parameter des ersten anzulegenden Magazinplatzes beschrieben wird, dann werden entsprechend den Angaben für die Zeilen- und Spaltenzahl des Magazins alle dem Magazin zugehörigen Magazinplätze mit ihren Vorbelegungswerten angelegt.
Setzen der Magazinplatztypenhierarchie	`$TC_MPTHx[y] = Wert;`	
Setzen der Magazindistanzen (Abstand zur Wechselstelle)	`$TC_MPTHx[y] = Wert;`	
Löschen der Magazindistanzen (Abstand zur Wechselstelle)	`$TC_MDPx[y,0] = 0`	Lösche alle definierten Distanzen des Magazins mit der Nummer "y". D. h., das Magazin wird beim Werkzeugsuchen und Leerplatzsuchen nicht mehr "gesehen".
	`$TC_MDPx[0,0] = 0;`	Lösche alle definierten Distanzen aller Magazine der TO-Einheit.
Löschen der Zuordnungen des Zwischenspeichers zu Spindeln	`$TC_MLSR[x,0] = 0;`	Lösche alle definierten Zuordnungen eines Zwischenspeicher-Platzes mit der Nummer "x". D. h. der Platz "x" wird beim Werkzeugsuchen nicht mehr "gesehen".
	`$TC_MLSR[0,0] = 0;`	Lösche alle definierten Zuordnungen von Zwischenspeichern der TO-Einheit zu Spindeln.
Setzen der Magazinbausteindaten	`$TC_MAMPx = Wert;`	

Daten laden

Wegen der Kopplung von Werkzeugen mit Magazinen, über den Magazinplatzparameter $TC_MPP6 gilt folgende Vorschrift für sinnvolles Definieren von Werkzeugen und Magazinen:

1. Werkzeugdaten laden
2. Magazindaten laden
3. Die $TC_MPP6 Parameter laden (⇒ setzt Werkzeug auf Magazinplatz)

Bei der Datensicherung wird diese Reihenfolge eingehalten.

Die Schleifdaten eines Werkzeugs können erst beschrieben werden, wenn zuvor mindestens für eine Schneide der Werkzeugtyp = "Schleifwerkzeug" festgelegt wurde.

Die Distanzparameter ($TC_MDPx) und der Zwischenspeicherzuordnungsparameter ($TC_MLSR) können erst beschrieben werden, nachdem die Magazine und ihre Plätze definiert worden sind.

Daten löschen

Ein Werkzeug kann nicht gelöscht werden, solange es in einem Magazin enthalten ist. Das heißt, dass die Reihenfolge beim Löschen folgende ist:

1. Magazindaten löschen (damit werden Werkzeuge aus dem Magazin ausgetragen) bzw. Werkzeug explizit aus dem Magazin austragen.
2. Werkzeugdaten löschen

Weiterhin kann ein Magazin nicht gelöscht werden, wenn es den Zustand $TC_MAP3[i] = 8 (Bewegen ist aktiv) hat. Der Löschbefehl für alle Magazine wird abgelehnt, wenn auch nur **ein** Magazin das Löschen verhindert.

Hinweis

Soll ein einzelnes Werkzeug gelöscht werden, so muss es zuerst durch einen Entladevorgang aus dem Magazinplatz entfernt werden und kann erst dann gelöscht werden.

Aktuell angewählte Werkzeuge können nicht gelöscht werden! Durch die Programmierung von T0 vor Ende des Teileprogramms kann unabhängig von Maschinendaten-Einstellungen (siehe MD zur Anwahl von Werkzeugen über das Programmende hinaus) sichergestellt werden, dass nach Beenden des Teileprogramms kein Werkzeug mehr angewählt ist.

11.3 Verwendung der Werkzeug- und Schneidendaten

Übersicht

Wenn ein Parameter einer nicht existierenden Schneide, eines Werkzeugs oder Magazins beschrieben wird, werden die Schneide, das Werkzeug bzw. das Magazin neu angelegt.

Hinweis

Beim Anlegen des Werkzeugs werden auch alle schneidenspezifischen Daten der Schneide D1 mit angelegt. (DP, DPC, MOP, MOPC mit "0" vorbesetzt.) Die schleifspezifischen Werkzeugdaten ($TC_TG1...) werden erst angelegt, sobald einer der Werkzeugtypen ($TC_DP1) 400-499 für irgendeine Schneide des Werkzeugs programmiert wird.

Verwendung der Werkzeug- und Schneidendaten mit Werkzeugverwaltung

Magazinverwaltung und Überwachung sind aktiv (WZMG und WZMO)

Aktion	Programm-Befehl	Beschreibung

Anlegen eines Werkzeugs	`T_NR = NEWT("Werkzeugbezeichner", Duplonummer)`
	oder
	`$TC_TP1[y] = Duplonummer;`
	`$TC_TP2[y] = "Werkzeugbezeichner"`
		`y = T-Nummer`

Anlegen einer Schneide	`$TC_DPx[y,z] = Wert`	Schneide D=z anlegen, wenn D=z noch nicht existiert!
		`y = T-Nummer`
		`z = D-Nummer`

Setzen von Werkzeugdaten	`$TC_TPx[y] = Wert;`
	oder
	`$TC_TPx[GETT("BOHRER",DUPLO_NR)] = Wert;`
	oder
	`$TC_TPCx[y] = Wert;`
	`$TC_TGx[y] = Wert;`
		`y = T-Nummer`
		Schreiben der werkzeugbezogenen Anwenderdaten
		Schreiben der werkzeugbezogenen Schleifdaten

Setzen der Daten einer Werkzeugschneide	`$TC_DPx[y,z] = Wert`
	`$TC_DPCx[y,z] = Wert`
	`$TC_MOPx[y,z] = Wer`
	`$TC_MOPCx[y,z] = Wert`
		Schreiben der Korrekturdaten
		Schreiben der schneidenbezogenen Anwenderdaten
		Schreiben der schneidenbezogenen Überwachungsdaten
		Schreiben der CC (OEM) Schneidenüberwachungsdaten
		`y = T-Nummer`
		`z = D-Nummer`

Löschen der Werkzeug- und Schneidendaten	`$TC_TP1[0,0];`	Beim Löschen der Werkzeuge müssen auch die Einträge der Platzdaten korrigiert werden.

Löschen eines Werkzeugs	`$TC_TP1[y] = 0;`
	oder
	`$TC_TP1[GETT("Werkzeugbezeichner", Duplonummer)] = 0;`
	oder
	`DELT["Werkzeugbezeichner", Duplonummer]`
		Es werden alle werkzeugbezogenen Daten auf "0" gesetzt (Anwenderdaten, Hierarchiedaten, ...). Beim Löschen eines Werkzeugs müssen auch die Einträge der Platzdaten korrigiert werden.

Verwendung der Werkzeug- und Schneidendaten ohne Werkzeugverwaltung

Aktion	Programm-Befehl	Beschreibung

Anlegen eines Werkzeugs	`$TC_DPx[y,z] = Wert;`
	
	`$TC_TP2[y] = "Nummer"`
		Werkzeug T anlegen, wenn T noch nicht existiert!
		`y = T-Nummer`
		`z = D-Nummer`

Anlegen einer Schneide	`$TC_DPx[y,z] = Wert`	Schneide D=z anlegen, wenn D=z noch nicht existiert!
		`y = T-Nummer`
		`z = D-Nummer`

Setzen der Daten einer Werkzeugschneide	`$TC_DPx[y,z] = Wert`
	`$TC_DPCx[y,z] = Wert`
	`$TC_MOPx[y,z] = Wer`
	`$TC_MOPCx[y,z] = Wert`
		Schreiben der Korrekturdaten
		Schreiben der schneidenbezogenen Anwenderdaten
		Schreiben der schneidenbezogenen Überwachungsdaten
		Schreiben der CC (OEM) Schneidenüberwachungsdaten
		`y = T-Nummer`
		`z = D-Nummer`

Löschen der Werkzeug- und Schneidendaten	`$TC_DP1[0,0] = 0;`	Alle Werkzeuge des Kanals werden gelöscht, der Speicher wird frei gegeben.

Löschen der Daten

Beim Löschen wird der Speicherbereich gelöscht und automatisch wieder frei gegeben.

Ein Werkzeug kann nur gelöscht werden, wenn es nicht aktiv (Korrektur bestimmend) ist und sich nicht in einem Magazin befindet, d. h., es muss ausgewechselt bzw. entladen sein.

Hinweis

Wird die Werkzeugverwaltung benutzt, muss darauf geachtet werden, dass es für das zu löschende Werkzeug keine Zuordnung zu einem Magazinplatz gibt ($TC_MPP6). Diese Zuordnung muss vor dem Werkzeuglöschvorgang gelöscht werden.

Die schleifspezifischen Werkzeugdaten ($TC_TG1,...) werden angelegt, sobald einer der Werkzeugtypen ($TC_DP1) 400-499 für irgendeine Schneide des Werkzeugs programmiert wird.

Wird der Werkzeug-Typ vom aktuellen Wert aus dem Bereich 400-499 auf einen Wert außerhalb dieses Bereichs gesetzt, so wird der Speicher für die Schleifdaten wieder frei gegeben; d. h., die schleifspezifischen Daten sind damit verloren.

11.4 Schneidendaten

11.4.1 Schneidendaten

Schneidendaten

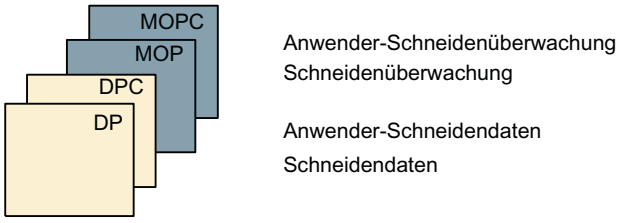


Bild 11-4 Übersicht Schneidendaten

Diese Daten sind für jede angelegte Schneide (D1- D12) vorhanden. Bei der WZV kommen zu den Geometrie- und Anwenderdaten noch die optionalen Überwachungsdaten der Schneiden dazu.

Werden Schneiden über HMI angelegt, wird die D-Nr. ab 1 beginnend gezählt. Werden Schneiden über ein NC-Programm angelegt, so ist es möglich, die D-Nr. mit Lücken zu programmieren, z. B. D1, D3, D6.

11.4.2 Schneidenparameter $TC_DP

$TC_DPx[t,D]

Schneidenparameter für Geometrie, Technologie und Werkzeugtyp.

Parameter:

- x: Parameter 1...25
  Der Maximalwert für x wird in der BTSS-Variablen "numCuttEdgeParams" im Baustein Y angezeigt.
- t: T-Nummer 1...32000
- D: Schneidennummer 1...12 bzw. D-Nummer

Pro Werkzeugschneide können abhängig vom Werkzeugtyp bis zu 25 Schneidenparameter programmiert werden.

Weitere Informationen: Funktionshandbuch Werkzeuge, Werkzeugkorrektur

BTSS-Baustein TO

Adressberechnungen:

- Zeile = (D - 1) * "numCuttEdgeParams" + Parameternummer
- Spalte = T-Nummer

Werkzeugkorrektur-Parameter (Systemvariablen)					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorb.[2)]**
$TC_DP1	INT	Werkzeugtyp	edgeData	REAL	9999
$TC_DP2	Double	Schneidenlage*	edgeData	REAL	0
$TC_DP3	Double	Geometrie Länge 1	edgeData	REAL	0
$TC_DP4	Double	Geometrie Länge 2	edgeData	REAL	0
$TC_DP5	Double	Geometrie Länge 3	edgeData	REAL	0
$TC_DP6	Double	Geometrie Radius	edgeData	REAL	0
$TC_DP7	Double	Geometrie - Eckenradius (WZ-Typ 700; Nutsäge)	edgeData	REAL	0
$TC_DP8[1)]	Double	Geometrie - Länge 4 (WZ-Typ 700; Nutsäge)	edgeData	REAL	0
$TC_DP9[1)]	Double	Geometrie Länge 5	edgeData	REAL	0
$TC_DP10[1)]	Double	Geometrie - Winkel 1	edgeData	REAL	0
$TC_DP11[1)]	Double	Geometrie - Winkel 2 für kegelige Fräswerkzeuge	edgeData	REAL	0
$TC_DP12	Double	Verschleiß - Länge 1	edgeData	REAL	0
$TC_DP13	Double	Verschleiß - Länge 2	edgeData	REAL	0
$TC_DP14	Double	Verschleiß - Länge 3	edgeData	REAL	0
$TC_DP15	Double	Verschleiß - Radius	edgeData	REAL	0
$TC_DP16	Double	Verschleiß - Nutbreit b/Verrundungsradius	edgeData	REAL	0
$TC_DP17	Double	Verschleiß - Überstand k	edgeData	REAL	0
$TC_DP18	Double	Verschleiß - Länge 5	edgeData	REAL	0
$TC_DP19	Double	Verschleiß - Winkel 1	edgeData	REAL	0
$TC_DP20	Double	Verschleiß - Winkel 2 für kegelige Fräswerkzeuge	edgeData	REAL	0
$TC_DP21	Double	Adapter - Länge 1	edgeData	REAL	0
$TC_DP22	Double	Adapter - Länge 2	edgeData	REAL	0
$TC_DP23	Double	Adapter - Länge 3	edgeData	REAL	0
$TC_DP24	Double	Freiwinkel	edgeData	REAL	0
$TC_DP25	Double	**ManualTurn**: Schnittgeschwindigkeit **ShopMill**: Zustandswert von WZ des Typs 1xx und 2xx.	edgeData	REAL	0
$TC_DPCE [t,d]	INT	Systemvariable eines Korrekturdatensatzes mit T=t und D=d, der die Schneidennummer CE enthält. (eindeutige D-Nr. oder auch freie Vergabe von D-Nr. zur Schneidennr.) Wertebereich erlaubter Schneidennummern: 1, 2, 3 ... MD18106.	-	-	0
$TC_DPH [t,d]	INT	H-Parameter (Y / extraCuttEdgeParams), Bit0=1	-	-	0
$TC_DPV	Double	Werkzeugschneidenorientierung	-	-	-

Werkzeugkorrektur-Parameter (Systemvariablen)					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorb.[2)]**
$TC_DPV3		L1-Komponente der Werkzeugschneidenorientierung	-	-	-
$TC_DPV4		L2-Komponente Werkzeugschneidenorientierung	-	-	-
$TC_DPV5		L3-Komponente Werkzeugschneidenorientierung	-	-	-
$TC_DPNT	INT	Zähnezahl, Schneidenparameter 34	-	-	-
$TC_DPROT	Double	Werkzeugspannwinkel bei Drehwerkzeugen	-	-	-
$TC_DPVNx	Double	Komponente der Orientierungsnormalen	-	-	-
1) Diese Daten haben, abhängig vom Werkzeugtyp, unterschiedliche Bedeutungen.					

$TC_DP11

$TC_DP11 enthält die Kennung für die Hauptbearbeitungsrichtung wie sie von Siemens-Zyklus 950 definiert und benötigt wird. $TC_DP11 nimmt eine Zwischenstellung von Werkzeug-OEM-Parameter und NC-Systemvariable ein:

- $TC_DP11 ist Werkzeug-OEM-Parameter insofern, als NC den Wert inhaltlich nicht auswertet.
- $TC_DP11 ist NC-Systemvariable insofern, als NC beim Zugriff über $P_ADT[n], mit n = 11, die Werte 1, 2, 3 und 4 der Werkzeug-Adaptertransformation unterwirft, falls WZMG und Unterfunktion "Werkzeug-Adapter" aktiv sind. Diese Systemparametereigenschaft tritt auch beim analogen BTSS-Baustein TOT zu Tage.

11.4.3 Anwender-Schneidendaten $TC_DPC

$TC_DPCx[t,D]

Pro Werkzeugschneide können bis zu 64 zusätzliche Schneidenparameter programmiert werden. Einstellen mit MD18096 MM_NUM_CC_TOA_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2=1 setzen)

X = Parameter 1...64

T = T-Nummer 1...32000

D = Schneidennummer 1..12

D = D-Nummer

BTSS-Baustein TUE/TUO

Berechnung der Zeile: (d-1)*numCuttEdgeParams_tu+Parameternr.

Berechnung der Spalte: T-Nummer

Anwenderbezogene Schneidendaten					
NC-Bezeichner	Typ	Bezeichnung	BTSS-Variable	Typ	Vorbelegung
$TC_DPC1	Double	CC_Schneidenparameter1	edgeData	REAL	0
...	Double	...	edgeData	REAL	0
$TC_DPC64	Double	CC_Schneidenparameter64	edgeData	REAL	0

Hinweis

Die Daten werden in der Werkzeugverwaltung angezeigt. Hier könnte z. B. die "max. Schnittgeschwindigkeit" abgelegt werden, die dann im Teileprogramm ausgewertet wird.

11.4.4 Schneidenbezogene Werkzeugüberwachung $TC_MOP

$TC_MOPx[t,D]

Die Werkzeugschneiden werden nach Standzeit, Stückzahl und/oder Verschleiß überwacht.

X = Parameter 1...15

T = T_Nummer 1...32000

D = Schneidennummer 1...12

D = D-Nummer

BTSS-Baustein TS

Berechnung der Zeile: (d-1)*numCuttEdgeParams_ts+Parameternr.

Berechnung der Spalte: T-Nummer

Überwachungsdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbe-le-gung**
$TC_MOP1	Double	Vorwarngrenze Standzeit in min	data	REAL	0
$TC_MOP2	Double	Rest-Standzeit in Minuten	data	REAL	0
$TC_MOP3	INT	Vorwarngrenze Stückzahl	data	REAL	0
$TC_MOP4	INT	Rest-Stückzahl	data	REAL	0
$TC_MOP11	Double	Standzeit Sollwert	data	REAL	0
$TC_MOP13	INT	Stückzahl Sollwert	data	REAL	0
$TC_MOP5	Double	Verschleißvorwarngrenze - oder auch ortsabhängige Korrektur fein-Vorwarngrenze	data	REAL	0

Überwachungsdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbe-legung**
$TC_MOP6	Double	Verschleißistwert - oder ortsabhängige Korrektur fein-Istwert	data	REAL	0
$TC_MOP15	Double	Verschleißsollwert - oder ortsabhängige Korrektur fein-Sollwert	data	REAL	0

11.4.5 Anwender-Schneidenüberwachung $TC_MOPC

$TC_MOPCx[t,D]

Anwenderdaten Werkzeugüberwachung (schneidenbezogen)

Pro Werkzeugschneide können bis zu 64 (ab V4.7) zusätzliche Werkzeugüberwachungsparameter programmiert werden. Einstellung mit MD18098 MM_NUM_CC_MON_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

X = Parameter 1....64

T = T_Nummer 1....32000

D = Schneidennummer 1....12

D = D-Nummer

BTSS-Baustein TUS

Berechnung der Zeile: (d-1)*numCuttEdgeParams_tus+Parameternr.

Berechnung der Spalte: T-Nummer

Anwenderdaten Werkzeugüberwachung (schneidenbezogen)					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variablen**	**Typ**	**Vorbe-legung**
$TC_MOPC1	Int	CC-Überwachungsparameter	userdata	REAL	0
...	Int	...	userdata	REAL	0
$TC_MOPC64	Int	CC-Überwachungsparameter	userdata	REAL	0

11.4.6 Einsatzortabhängige Korrekturen fein (Summenkorrekturen) $TC_SCP

$TC_SCPx[t,D]

Einsatzortsabhängige Korrekturen fein (häufig wird auch der Begriff Summenkorrekturen verwendet) setzen sich aus den Fehlergrößen zusammen, die die Abweichung des Werkstücks vom Sollmaß ausmachen. Die Parameter der einsatzortsabhängigen Korrekturen beziehen sich auf die geometrischen Daten einer Schneide. DL: Location dependent; Korrekturen bezüglich des jeweiligen Einsatzorts.

X = Parameter für DL=1...DL=6

T = T-Nummer 1...32000

D = Schneidennummer 1...12

D = D-Nummer

BTSS-Baustein TOS, TOST

Berechnung der Zeile: (d-1)*(maxnumEdgeSC*numParams_SC) + ((EdgeSC_1)*numParams_SC)+Parameternr.

Berechnung der Spalte: T-Nummer

Ortsabhängige Korrekturen				
Name	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**
$TC_SCPx				
x = 13-21	Double	aktivierbar mit DL=1	edgeSCData	REAL
x = 23-31	Double	aktivierbar mit DL=2	edgeSCData	REAL
x = 33-41	Double	aktivierbar mit DL=3	edgeSCData	REAL
x = 43-51	Double	aktivierbar mit DL=4	edgeSCData	REAL
x = 53-61	Double	aktivierbar mit DL=5	edgeSCData	REAL
x = 63-71	Double	aktivierbar mit DL=6	edgeSCData	REAL
		Transformierte ortsabhängige Korrekturen fein, Baustein TOST	edgeSCData	REAL

11.4.7 Ortsabhängige Korrekturen grob (Einrichtekorrekturen) $TC_ECP

$TC_ECPx[t,D]

Ortsabhängige Korrekturen grob (auch Einrichtekorrekturen) können vom Einrichter vor der Bearbeitung eingestellt werden (siehe auch $TC_SCP (Seite 117)).

X = Parameter für DL=1...DL=6

T = T-Nummer 1...32000

D =Schneidennummer 1...12

D = D-Nummer

BTSS-Baustein TOE, TOET

Berechnung der Zeile: (d-1)*(maxnumEdge_SC*numParams_SC)+
((EdgeSC-1)*numParams_SC)+Parameternr.

Berechnung der Spalte: T-Nummer

Einrichtekorrekturen				
Name	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**
$TC_ECPx			edgeECData	REAL
x = 13-21	Double	aktivierbar mit DL=1	edgeECData	REAL
x = 23-31	Double	aktivierbar mit DL=2	edgeECData	REAL
x = 33-41	Double	aktivierbar mit DL=3	edgeECData	REAL
x = 43-51	Double	aktivierbar mit DL=4	edgeECData	REAL
x = 53-61	Double	aktivierbar mit DL=5	edgeECData	REAL
x = 63-71	Double	aktivierbar mit DL=6	edgeECData	REAL
		Transformierte Einrichtekorrekturen, Baustein TOET	edgeECData	REAL

11.5 Werkzeugdaten

11.5.1 Übersicht

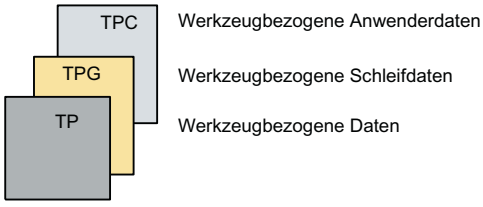


Bild 11-5 Übersicht Werkzeugdaten

11.5.2 Werkzeugbezogene Daten $TC_TP

$TC_TPx[t]

Allgemeine Werkzeugdaten

Diese Daten beschreiben das Werkzeug im Magazin.

Programmierung der allgemeinen Werkzeugdaten bei Werkzeugverwaltung

x: = Parameter 1...11

t: = T-Nummer 1...32000

BTSS-Baustein TD

Berechnung der Zeile: T-Nummer

Berechnung der Spalte: entfällt

Werkzeugbezogene Daten WZV					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_TP2	String	Werkzeugbezeichner	toolIdent	String	"T-Nr."
$TC_TP1	INT	Duplo-Nummer	duploNo	WORD	T-Nr.
$TC_TP3	INT	Größe nach links	toolsize_left	WORD	1
$TC_TP4	INT	Größe nach rechts	toolsize_right	WORD	1
$TC_TP5	INT	Größe nach oben	toolsize_upper	WORD	1
$TC_TP6	INT	Größe nach unten	toolsize_down	WORD	1
$TC_TP7	INT	Magazinplatztyp	toolplace_spec	WORD	9999
$TC_TP8	INT	Status Wert 0 nicht freigegeben Bit 0 aktives Werkzeug Bit 1 freigegeben Bit 2 gesperrt Bit 3 vermessen Bit 4 Vorwarngrenze erreicht Bit 5 WZ im Wechsel Bit 6 festplatzcodiert Bit 7 WZ war im Einsatz Bit 8 Kennung für Werkzeuge im Zwischenspeicher Bit 9=1 ignoriere Gesperrtzustand Bit 9=0 nicht ignorieren Bit 10 zu entladen Bit 11 zu beladen Bit 12 reserviert Bit 13 reserviert Bit 14 1:1-Tausch Bit 15 Handwerkzeug Bit 17 Werkzeug ist auf einem gesperrten Magazinplatz	toolState	WORD	0=nicht freigegeben
$TC_TP9	INT	Art der Werkzeugüberwachung Wert 0 keine WZ-Überwachung Bit 0 Standzeit Bit 1 Stückzahl Bit 2 Verschleißüberwachung aktiv Bit 3 Verschleißüberwachung, Ortsabhängige Korrektur fein aktiv	toolMon	WORD	0
$TC_TP10	INT	Nummer für ein Ersatzwerkzeug.	toolSearch	WORD	0

Werkzeugbezogene Daten WZV					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_TP11	INT	Werkzeuginfo: Damit können WZ-Gruppen in Untergruppen aufgeteilt werden. Werkzeug-Anwahl nur mit Werkzeugen der Untergruppe	toolInfo	Integer	0
$A_TOOLMN	INT	Magazinzuordnung Werkzeug	toolInMag	WORD	
$A_TOOLMLN	INT	Platzzuordnung Werkzeug	toolInPlace	WORD	
$P_TOOLND	INT	Anzahl Schneiden	numCuttEdges	WORD	
-		AdapterNr. Zuordnung	adaptNo	WORD	
$A_MYMN	INT	EigentümerMagazin des Werkzeugs	toolMyMag	WORD	
$A_MYMLN	INT	EigentümerMagazinplatz des Werkzeugs	toolMyPlace	WORD	
$TC_TP_PROTA	string	Name des dreidimensionalen Schutzbereichs für das Werkzeug oder Name der Datei, die die Beschreibung des Schutzbereichs für das WZ enthält			
$TC_TP_MAX_VELO	real	Maximale Drehzahl des Werkzeugs, wenn der Wert > 0 ist. Wenn keine Drehzahlgrenze definiert ist (=0), dann findet keine Überwachung statt	toolMaxVelo		
$TC_TP_MAX_ACC	real	Maximale Beschleunigung des Werkzeugs, wenn der Wert > 0 ist. Wenn keine Beschleunigungsgrenze definiert ist (=0), findet keine Überwachung statt	toolMaxAcc		

$TC_TP1 und $TC_TP2

So wie die T-Nr. zur eindeutigen Identifizierung eines Werkzeuges ausreicht, so ist ein Werkzeug ebenso eindeutig durch seine Duplo-Nr. und seinen WZ-Namen (Bezeichner) bestimmt.

Innerhalb einer TO-Einheit dürfen daher nur Namen mit unterschiedlicher Duplo-Nr. enthalten sein. Die Schreibvorgänge von $TC_TP1 und $TC_TP2 werden dahingehend überwacht und bei Kollisionen abgelehnt.

$TC_TP3 bis TP6

Größe nach Halbplätzen:

Die Größe 1 bedeutet, dass das Werkzeug genau den eigenen Magazinplatz vollständig belegt. Wenn die Größe in einer Richtung um 1 höher wird, wird noch ein halber Nebenplatz in dieser Richtung für das Werkzeug reserviert.

Die maximal programmierbare Größe ist 11. Das bedeutet, dass höchstens fünf ganze Nebenplätze in einer Richtung vom Werkzeug reserviert werden können.

Die Werkzeuggröße (Halbplätze) kann nur geschrieben bzw. geändert werden, wenn das Werkzeug keinen Eigentümerplatz hat. Dieses trifft auf folgende Situationen zu:

- Das Werkzeug ist noch nicht beladen
- Während des Beladevorgangs, solange das Werkzeug noch nicht auf seinem Zielplatz in einem realen Magazin ist
- Wenn sich das Werkzeug auf einem Platz des Zwischenspeicher- oder Belademagazins befindet und zuvor der Eigentümerplatz sowie die Reservierung im Magazin gelöscht wurde (ausschließlich mit den Sprachbefehlen DELMRES und DELMOWNER).

$TC_TP7

Der Magazinplatztyp kann nur geschrieben bzw. geändert werden, wenn das Werkzeug keinen Eigentümerplatz hat. Dieses trifft auf folgende Situationen zu:

- Das Werkzeug ist noch nicht beladen
- Während des Beladevorgangs, solange das Werkzeug noch nicht auf seinem Zielplatz in einem realen Magazin ist
- Wenn sich das Werkzeug auf einem Platz des Zwischenspeicher- oder Belademagazins befindet und zuvor der Eigentümerplatz sowie die Reservierung im Magazin gelöscht wurde (ausschließlich mit den Sprachbefehlen DELMRES und DELMOWNER).

$TC_TP8

Die Beschreibung des Werkzeugzustandes erfolgt über die Systemvariable $TC_TP8. Die Systemvariable ist bitcodiert. Damit ist jedem Bit dieses Datums ein bestimmter Zustand des Werkzeuges zugeordnet.

Ein Werkzeug muss den Zustand **Bit 1=1** ("freigegeben") haben, damit es im Rahmen eines programmierten Werkzeugwechsels für die Bearbeitung auf den Werkzeughalter eingewechselt werden kann.

Ein Werkzeug, das auf den Werkzeughalter (Spindel, ...) eingewechselt wird, wird von der NC bei der Werkzeuganwahl auf den Zustand **Bit 0** ("aktiv") gesetzt.

Ein Werkzeug kann nicht eingewechselt werden, wenn es den Zustand **Bit 2=1** ("gesperrt") hat. Der Zustand wird von der Werkzeug-Überwachungsfunktion automatisch gesetzt, wenn ein Überwachungswert mindestens einer Schneide den Grenzwert erreicht. Bei der Erzeugung von INIT-Sätzen (siehe MD 20110 und 20112) wird bzw. kann der Zustand Bit 2=4 des Werkzeugs auf dem Werkzeughalter ignoriert werden. Auch die PLC hat die Möglichkeit NC zu veranlassen, den Zustand bei der Werkzeuganwahl zu ignorieren.

Der Zustand **Bit 4=1** ("Vorwarngrenze") hat vorzugsweise informativen Charakter. Das Werkzeug ist mit diesem Zustand weiterhin einwechselbar.

Der Zustand **Bit 7=1** ("war im Einsatz") wird von NC gesetzt, wenn das Werkzeug von einem Magazinplatz des Typs Spindel bzw. Werkzeughalter entfernt wird.

Der Werkzeugzustand **Bit 5=1** (="W"= befindet sich im Wechsel) wird im gepufferten Hochlauf von der Software stets zurückgesetzt. Ein Werkzeug erhält/verliert diesen Zustand im Rahmen eines programmierten Werkzeugwechsels.

Es gilt: Alle am Werkzeugwechsel beteiligten Werkzeuge (neu und alt) erhalten mit der Werkzeuganwahl den Status Bit 5=1. Mit der Endequittierung des jeweiligen Werkzeugbefehls wird der Zustand wieder zurückgesetzt.

Speziell gilt:

Mit Endequittierung des PLC-Kommandos 2 (Programmierung der T Adresse mit $MC_TOOL_CHANGE_MODE=1) wird der Zustand "W" des alten Werkzeugs zurückgesetzt.

Mit Endequittierung der PLC-Kommandos 3, 4, 5
(Programmierung von M06 in einem Satz mit $MC_TOOL_CHANGE_MODE=1, T, M06 in einem Satz mit $MC_TOOL_CHANGE_MODE=1
T-Adresse mit $MC_TOOL_CHANGE_MODE=0)
wird der Zustand Bit 5=1 des alten und des neuen Werkzeugs zurückgesetzt.

Werkzeuge, die sich im Zwischenspeicher befinden, sind für einen neu programmierten Werkzeugbefehl auch einsatzfähig, wenn sie den Zustand Bit 5=1 (im Wechsel) haben.

Werkzeuge, die sich im realen Magazin befinden und diesen Zustand haben, sind in Abhängigkeit des Bits 21 des MD 20310 einsatzfähig bzw. nicht einsatzfähig für einen konkurrierenden Werkzeug-Wechselbefehl einer anderen Spindel.

Bei der Werkzeuganwahl im Rahmen von Satzsuchlauf und bei der Initsatzgenerierung wird der Zustand Bit 5=1 generell nicht beachtet.

Bei RESET wird in den Werkzeugen, die zu dem Zeitpunkt an einem Werkzeugwechsel beteiligt sind, der Zustand zurückgesetzt.

Der Zustand Bit 5=1 wird bei der Werkzeuganwahl eines Handwerkzeugs nicht ausgewertet.

Der Werkzeugzustand **Bit 8=1** ("Rücktransport") sorgt beim nächsten Werkzeugwechsel dafür, dass ein Werkzeug, welches sich auf einem Zwischenspeicher befindet und nicht für den nächsten Bearbeitungseinsatz bestimmt ist, wieder in das reale Magazin zurück transportiert wird.

Bit 9 ignoriere Gesperrtzustand.

Sitzt dieses Bit wird der Gesperrtzustand dieses Werkzeugs ignoriert. D. h. das gesperrte Werkzeug kann (Abhängig von der Suchstrategie) zum Einsatz kommen.

Dieser Zustand wirkt unabhängig von dem PLC-Nahtstellensignal:

"WZ-Sperre unwirksam" (DB21.DBx29.7).

Zustand Bit 11 (zu beladen)

Bit 11 wird bei Werkzeugen gesetzt, die sich nicht auf einem Magazin befinden und beladen werden sollen. Es gelten folgende Festlegungen:

- Der Zustand bleibt bei Power On erhalten.
- Er wird bei der Datensicherung übernommen und beim Wiedereinspielen in den NC erneut geschrieben.
- Beim Zuordnen eines Werkzeuges zu einem realen Magazin wird vom NC der WZ-Zustand zurückgesetzt (gilt bei Plätzen der Platzart 1, also nicht bei internen Magazinen wie Beladem agazin, Zwischenspeichermagazin etc.).

Bit	Wert	Bedeutung
11	0	"nicht zu beladen"
	1	"zu beladen"

Zustand Bit 10 (zu entladen)

Bit 10 wird bei Werkzeugen gesetzt, die sich in einem Magazin befinden und entladen werden sollen. Es gelten folgende Festlegungen:

- Der Zustand bleibt bei Power On erhalten
- Er wird bei der Datensicherung übernommen und beim Wiedereinspielen in den NC erneut geschrieben.
- Beim Entladen des Werkzeuges über einen Entladeplatz wird vom NC der WZ-Zustand zurückgesetzt.

Bit	Wert	Bedeutung
10	0	"nicht zu entladen"
	1	"zu entladen"

Bei $MC_TOOL_CHANGE_ERROR_MODE, Bit 9 = 1:

Ein WZ, das den Zustand 'zu entladen' hat ($TC_TP8, Bit 10 = 1), wird wie ein gesperrtes Werkzeug behandelt, d.h. es wird bei der WZ-Suche nicht mehr berücksichtigt, und ist im Rahmen einer T-Programmierung nicht mehr anwählbar.

Zustand Bit 12 (reserviert)

Bit 12 ist reserviert.

Zustand Bit 14 (WZ für 1:1-Tausch markiert)

NC-interner Zustand, der anzeigt (bei eingestellter Suchstrategie 1:1-Tausch), dass Neu- und Altwerkzeug 1:1 getauscht werden.

Hinweis

Vorsicht bei "manuellem" Ändern des Werkzeugzustandes über BTSS währen!d der Bearbeitung. Dieses könnte interne Zustandsänderungen durch den NC rückgängig machen und Fehlerbearbeitungen zur Folge haben.

Zustand Bit 17 (Werkzeug ist auf einem gesperrten Magazinplatz)

Werkzeuge, die auf einem nachträglich gesperrten Magazinplatz sitzen, bekommen mit $TC_TP8 Bit 17=1 die Kennung, dass sie auf einen gesperrten Magazinplatz sitzen. Mit der Anweisung $TC_MPP6 lassen sich so gekennzeichnete Werkzeuge auf gesperrte Magazinplätze bzw. gesperrte Nebenplätze setzen. Übergroße Werkzeuge, die über den Rand des Magazins hinaus ragen, werden nicht berücksichtigt.

$TC_TP9

Wird mit $TC_TP9 eine Überwachungsart für das Werkzeug aktiviert, so werden die aktuellen Überwachungsparameter ausgewertet und gegebenenfalls der Werkzeugzustand auf "gesperrt" oder "Vorwarngrenze erreicht" gesetzt. Eine bestehende Werkzeugsperre wird hingegen nicht aufgehoben. Auch nicht dann, wenn die Überwachungsfunktion für dieses Werkzeug ausgeschaltet wird.

$TC_TP11

Werkzeug-Untergruppen

Die Systemvariable ist bitcodiert. Es werden nur die Bits 0...3 ausgewertet. Eine WZ-Gruppe (gleicher Bezeichner, verschiedene Duplo-Nr.) kann dadurch in maximal 4 Untergruppen aufgeteilt werden. Dabei kann ein Werkzeug auch in mehreren Untergruppen enthalten sein.

Ist kein Bit gesetzt, also $TC_TP11[x]=0 ist das gleichbedeutend mit "alle Bits gesetzt", d. h., das WZ gehört zu allen definierten Untergruppen.

Auswahl der WZ-Untergruppe

1. Mit dem Sprachbefehl **$P_USEKT** (UseKindofTool)
   (nur möglich, wenn nicht mit der Einstellung T=Platz gearbeitet wird)
   Es können bei der WZ-Suche nur Werkzeuge gefunden werden, die in der Systemvariablen $TC_TP11 eines dieser Bits sitzen haben. Damit ist es möglich, so genannte "Technologiegruppen" zu bilden, Werkzeuge mit gleichem Bezeichner zu differenzieren und gezielt für die Bearbeitung freizugeben.
   Beispiel 1:
   $P_USEKT=4
   D. h. es werden nur WZ berücksichtigt, die im $TC_TP11 das Bit 2 sitzen haben oder
   Beispiel 2:
   $P_USEKT=9
   D. h., es werden nur WZ berücksichtigt, die im $TC_TP11 das Bit 3 oder 0 sitzen haben
2. Durch Programmierung eines Werkzeugs
   Bei der Funktion **T=Platz** wird $P_USEKT mit jedem WZ-Wechsel automatisch gesetzt und zwar auf den $TC_TP11-Wert des eingewechselten Werkzeugs.
   Beispiel: T3 M06
   Der Bit-Wert des $TC_TP11 von T3 ist jetzt gültig (wird in "USEKT übernommen").
   Beim Übergang auf ein Ersatz-WZ (und nur da) werden nur solche Werkzeuge berücksichtigt, die in der Systemvariablen $TC_TP11 eines dieser Bits gesetzt haben.

11.5.3 Werkzeugbezogene Schleifdaten $TC_TPG

$TC_TPGx[t]

Technologiespezifische Schleifdaten

Die Vorbelegung der Schleifdaten erfolgt mit "0". Werkzeuge mit **Werkzeugtyp 400 bis 499** sind immer **Schleifwerkzeuge**, d. h., haben zusätzlich diese Daten, die auch zusätzlich Speicherplatz belegen. Wird ein Typ von 400-499 auf einen Typ außerhalb dieses Bereiches geändert, so

verliert ein solches Werkzeug seine schleifspezifischen Daten - der zugehörige Speicher wird dabei wieder freigegeben und kann für andere Werkzeuge benutzt werden.

x: = Parameter 1....9

t: = T-Nummer 1...32000

BTSS-Baustein TG

Berechnung der Zeile: T-Nummer

Berechnung der Spalte: Entfällt

Werkzeugbezogene Schleifdaten				
Name	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$TC_TPG 1	INT	Spindelnummer	spinNoDress	REAL
$TC_TPG 2	INT	Verkettungsvorschrift	conntectPar	REAL
$TC_TPG 3	Double	minimaler Scheibenradius	minToolDia	REAL
$TC_TPG 4	Double	minimale Scheibenbreite	minToolWide	REAL
$TC_TPG 5	Double	aktuelle Scheibenbreite	actToolWide	REAL
$TC_TPG 6	Double	maximale Drehzahl	maxRotSpeed	REAL
$TC_TPG 7	Double	maximale Umfangsgeschwindigkeit	maxTipSpeed	REAL
$TC_TPG 8	Double	Neigungswinkel schräge Scheibe	inclAngle	REAL
$TC_TPG 9	INT	Parameternummer für Radiusberechnung	paramNrCCV	REAL

11.5.4 Werkzeugbezogene Anwenderdaten $TC_TPC

$TC_TPCx[t]

Anwenderbezogene Werkzeugdaten

Pro Werkzeug können zusätzlich 64 (ab V4.7) werkzeugspezifische Parameter angelegt werden. Einstellen mit MD18094 MM_CC_TDA_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

x: = Parameter 1...64

t: = T-Nummer 1...32000

BTSS-Baustein TU/TUD

Berechnung der Zeile: T-Nummer

Berechnung der Spalte: Parameternummer

Werkzeugbezogene Daten OEM-Anwender				
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$ TC_TPC1	Double		data	REAL
...	Double		data	REAL
$TC_TPC64	Double		data	REAL

Hinweis

Die Daten werden in der Werkzeugverwaltung angezeigt. Hier können z. B. zusätzliche Werkzeugstatus abgelegt werden.

11.6 Magazindaten

11.6.1 Übersicht Magazindaten

Magazindaten

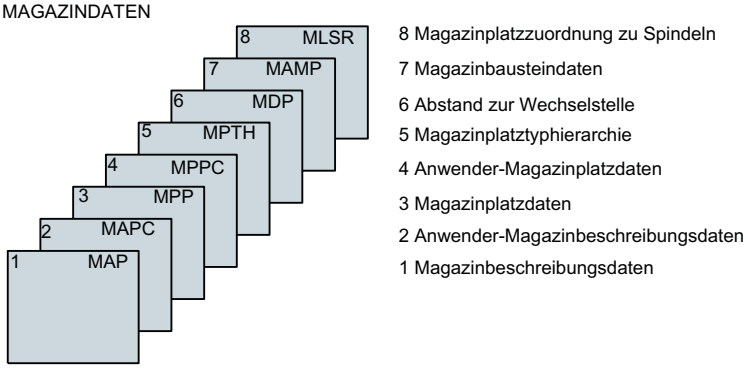


Bild 11-6 Übersicht Magazindaten

11.6.2 Magazinbeschreibungsdaten $TC_MAP

$TC_MAPx[n]

Magazinbeschreibungsdaten

Diese Daten kennzeichnen das reale Magazin:

x: = Parameter 1...10

n: = Magazinnummer 1...3200

BTSS-Baustein TM

Berechnung der Zeile: Magazinnummer

Berechnung der Spalte: Entfällt

Magazinbeschreibungsdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
		Magazinnummer	magNo	WORD	0
$TC_MAP2	String	Bezeichner des Magazins	magIdent	String	" "
$TC_MAP1	INT	Art des Magazins **1** = Kette **3** = Revolver **5** = Flächenmagazin **7** = WZ-Zwischenspeichermagazin **9** = Belademagazine	magKind	WORD	0
$TC_MAP3	INT	Zustand des Magazins Bit 0: aktives Magazin Bit 1: gesperrt Bit 2: Magazin in Beladeposition Bit 3: WZ-Bewegen ist aktiv Bit 4: Magazin bzw. WZ darf bewegt werden. Zum Beladen freigegeben Bit 6: Magazin ist festplatzcodiert, d.h. Werkzeuge in diesem Magazin werden wie festplatzcodierte Werkzeuge behandelt Bit 8: Randplatz darf nicht überlappt werden links Bit 9: Randplatz darf nicht überlappt werden rechts Bit 10: Randplatz darf nicht überlappt werden oben Bit 11: Randplatz darf nicht überlappt werden unten	magState	WORD	2
$TC_MAP4	INT	reserviert			-1
$TC_MAP5	INT	reserviert			-1
$TC_MAP6	INT	Anzahl Zeilen (nur Flächenmagazin)	magDim	WORD	1
$TC_MAP7	INT	Anzahl Spalten	-	-	
-	-	Anzahl Plätze des Magazins, entspricht TC_MAP6*TC_MAP7	magNrPlaces	WORD	0
$TC_MAP8	INT	Aktuelle Magazinposition bezogen auf die Wechselposition	magActPlace	WORD	0
-			magCmd	WORD	
-			magCmdState	WORD	
-			magCmdPar1	WORD	
-			magCmdPar2	WORD	
$TC_MAP9	INT	Aktuelle Verschleißverbundnummer	magWearCompoundNo	DINT	0

Magazinbeschreibungsdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbele-gung**
$TC_MAP10 (Bit 0...7)	INT	Aktuelle WZ-Suchstrategien des Magazins (siehe $TC_MAMP2)	magTool-SearchStrat	WORD	0
$TC_MAP10 (Bit 8...15)	INT	Aktuelle Leerplatzsuchstrategie des Magazins	magPlace-SearchStrat	WORD	0

$TC_MAP3

Der Magazinzustand **Bit 3** (Werkzeugbewegen ist aktiv) wird im gepufferten Hochlaufen der Software stets zurückgesetzt.

Ein Magazin, das den Zustand "Werkzeugbewegen ist aktiv" hat, kann nicht gelöscht werden.

In einem Magazin mit dem Zustand "gesperrt" erfolgt keine Leerplatzsuche. Wird für die Leerplatzsuche explizit ein gesperrtes Magazin vorgegeben, so wird mit einem Fehler abgebrochen.

Ein Werkzeug, das sich in einem "gesperrten" Magazin befindet, kann nicht in die Spindel bzw. den WZ-Halter eingewechselt werden.

Überlappen der Randplätze (Bit 8...11=1)

Bei Defaulteinstellung (Bit 8 ...11 = 0) überragt ein übergroßes Werkzeug die Magazinrandplätze. Muss das verhindert werden (z. B. auf Grund der mechanischen Gegebenheiten an der Maschine), kann eines oder mehrere dieser Bits gesetzt werden.

Beispiel: Bits 8 und 9 sind gesetzt, Bits 10 und 11 sind nicht gesetzt.

Das bedeutet, dass ein übergroßes Werkzeug das Magazin (typisch ein Flächenmagazin) an seiner linken und rechten Seite nicht überlappen darf, aber ein übergroßes Werkzeug das Magazin an seiner Ober- und Unterseite überlappen/überragen darf.

Dabei gilt:

Links und oben bzgl. eines Bezugsplatzes ist dort, wo sich die Magazinplätze mit den kleineren Magazinnummern befinden.

Rechts und unten bzgl. eines Bezugsplatzes ist dort, wo sich die Magazinplätze mit den größeren Magazinplatznummern befinden.

$TC_MAP8

Die aktuelle Magazinposition $TC_MAP8 wird von NC bei jeder Magazinbewegung aufgefrischt.

Nach Laden der Magazinkonfiguration hat die Variable $TC_MAP8 den Wert Null. Der Positionswert ist die Nummer des Magazinplatzes, der sich an der Nullpunktposition des Magazins befindet. Die Magazinposition kann als maximalen Wert die Nummer der Anzahl der Magazinplätze im Magazin haben. Größere und negative Werte werden abgelehnt.

$TC_MAP10

Magazinspezifische Werkzeugsuche

Die Bit-Einstellungen entsprechen exakt der Systemvariablen $TC_MAMP2.

Für Zwischenspeichermagazine gilt grundsätzlich die Defaulteinstellung "0", d. h., es wird vorwärts, ab dem 1. Magazinplatz gesucht.

11.6.3 Magazin-Anwenderdaten $TC_MAPC

$TC_MAPCx[n]

Magazin-Anwenderdaten

Pro Magazin können zusätzlich bis zu 64 (ab V4.7) Anwenderdaten angelegt werden. Einstellung im MD18090 MM_NUM_CC_MAGAZINE_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

x: = Parameter 1...64

n: = Magazin-Nummer 1...32000

BTSS-Baustein TUM

Berechnung der Zeile: Parameternummer

Berechnung der Spalte: Magazinnummer

Magazinbeschreibungsdaten OEM-Anwender				
Name	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$TC_MAPC1			userData	DINT
...			userData	DINT
$TC_MAPC64			userData	DINT

11.6.4 Magazinplatzdaten $TC_MPP

$TC_MPPx[n,m]

Magazinplatzdaten

Diese Daten beschreiben den Magazinplatz

x: = Parameter 1..7

n: = Physikalische Magazinnummer 1...32000

m: = Physikalische Platznummer 1...1500

Der maximale Wert von x ist in der BTSS-Variablen numMagPlaceParams im Baustein Y enthalten.

BTSS-Baustein TP

Berechnung der Zeile: (magazinPlatzNr-1)*numMagPlaceParams+Parameternr.

Berechnung der Spalte: Magazinnummer

Magazinplatzdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbe-legung**
$TC_MPP1	INT	Platzart 1 = Magazinplatz 2 = Spindel, Werkzeughalter 3 = Greifer 4 = Lader 5 = Übergabeplatz 6 = Beladestation 7 = Beladestelle	placeData	WORD	0
$TC_MPP2	INT	Platztyp > 0: Platztyp für virtuellen Platz = 0: jedes WZ passt auf diesen Platz 9999: nicht definiert	placeData	WORD	9999
$TC_MPP3	BO-OL	Nebenplatzbetrachtung ein/aus	placeData	WORD	FALSE
$TC_MPP4	INT	Platzzustand Bit 0: gesperrt Bit 1: frei zur Aufnahme eines Werkzeugs (belegt) Bit 2: reserviert für Werkzeug aus Zwischenspeicher Bit 3: reserviert für neu zu beladendes Werkzeug Bit 4: belegt im linken Halbplatz Bit 5: belegt im rechten Halbplatz Bit 6: belegt im oberen Halbplatz Bit 7: belegt im unteren Halbplatz Bit 8: linker Halbplatz reserviert Bit 9: rechter Halbplatz reserviert Bit 10:oberer Halbplatz reserviert Bit 11: unterer Halbplatz reserviert Bit 12: Verschleißverbund gesperrt Bit 13: Überlappung erlaubt Gesperrte Magazinplätze können durch übergroße WZ überlappt werden. Nur möglich bei aktiver Nebenplatzbetrachtung (siehe auch MD 17520).	placeData	WORD	1
-		Referenz phys. Magazin (r.o.)	placeData	WORD	0
$TC_MPP5	INT	Platzartindex (Nummerierung einer Platzart) oder Verschleißverbundnummer	placeData	WORD	0
$TC_MPP6	INT	T-Nr. des Werkzeugs auf diesem Platz	placeData	WORD	0
$TC_MPP7	INT	Nr. des Adapters auf Mag.Platz	placeData	WORD	0

Magazinplatzdaten Werkzeugverwaltung					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_MPP_SP	INT	Für WZ-Drehzahl und Beschleunigungsüberwachung Nur relevant, wenn $MC_TOOLHOLDER_MANAGEMENT > 0. Spindel-Nr., die mit dem Werkzeughalter verbunden ist	placeData		
$TC_MPP66	INT	T-Nr. des Werkzeugs, für das die Kennung "reserviert für WZ im Zwischenspeicher" sitzt. Wird ein Werkzeug im Rahmen des Wechsels vom Magazin in die Spindel transportiert, wird der Magazinplatz reserviert (Kennung "reserviert für WZ im Zwischenspeicher"), im $TC_MPP6 wird die T-Nr. dieses WZ eingetragen, und das Werkzeug hat in den Variabeln $A_MYMN/MYMLN diesen Platz eingetragen.	placeData		

Schreiben von Magazinplatzdaten

Besonderheiten beim Schreiben von Magazinplatzdaten:

Beim erstmaligen Beschreiben einer der $TC_MPP... - Parameter werden alle durch die Magazinparameter definierten Magazinplätze mit ihren Defaultwerten angelegt (der Speicher für die Plätze wird damit "verbraucht"). D. h., zu diesem Zeitpunkt muss die Definition des Magazins ($TC_MAP... - Parameter) erfolgt sein.

$TC_MPP1 (Platzart)

Auf Magazinen, die nicht von der Art "intern" sind ($TC_MAP1 = 7 oder = 9), dürfen nur Magazinplätze von der Art "Magazinplatz" ($TC_MPP1 = 1) definiert werden.

Zu den Platzarten:

- 1 = Magazinplatz
  Auf realen Magazinen dürfen nur Plätze vom Typ "1" definiert werden.
- 2 = Spindel/Toolholder
- 3 = Greifer
- 4 = Lader
- 5 = Übergabeplatz
  Die Unterscheidung Greifer/Lader/Übergabeplatz ist für HMI Applikationen gedacht. NC trifft hier keine Unterscheidung, Lader und Übergabeplatz werden wie Greifer behandelt.
- 6 = Beladestation
  Nach dem WZ bewegen auf diesen Platz bleibt das WZ dort. Es kann nur durch explizite Bedienhandlung von dort entfernt (entladen) werden.
- 7 = Beladestelle

Wird ein WZ aus dem Magazin oder Zwischenspeicher auf diesen Platz bewegt, so wird das WZ nach der PLC-Quittung dieses Bewegekommandos automatisch von diesem Platz entfernt.

Es ist zu beachten, dass es beim Schreiben von Platzzustand und Nummer des Werkzeugs auf diesem Platz folgende Abhängigkeiten zu $TC_MPP2 bis $TC_MPP4 gibt, die während des Schreibvorgangs geprüft werden:

- Enthält der Platz bereits ein Werkzeug, ist der zu schreibenden Platztyp mit dem Platztyp des Werkzeugs zu prüfen.
- Zustand "frei" darf nur geschrieben werden, wenn keiner der "belegt"-Zustände gesetzt ist und wenn kein Werkzeug auf dem Platz ist.
- Zustand "gesperrt" kann, unabhängig von den Zuständen, gesetzt werden.
- Ist kein Werkzeug enthalten, dann ist der Zustand "frei" automatisch aktiv; d. h., es kann durch NC-Programm oder PLC, HMI nicht der Zustand "nicht frei" gesetzt werden.
- "Belegt"-Zustände können nur von NC im Rahmen der Nebenplatzbetrachtung gesetzt werden; d. h. beim Schreiben durch NC-Programm oder PLC, HMI werden diese Zustände ignoriert.
- Zustand "reserviert für WZ aus Zwischenspeicher" wird beim Werkzeugwechsel von NC bei Entnahme des WZ aus dem realen Magazin gesetzt. Dieser Platz ist damit für andere WZ als dem entnommenen nicht als "frei" erkennbar.
- Die Zustände "reserviert für WZ aus Zwischenspeicher" und "reserviert für neu zu beladenes WZ" **eines Platzes** werden automatisch zurückgesetzt, wenn ein WZ auf diesen Platz gesetzt wird.
- Die Zustände "reserviert für WZ aus Zwischenspeicher" und "reserviert für neu zu beladenes WZ" eines **realen Magazinplatzes** werden automatisch zurückgesetzt, wenn ein WZ von diesem Platz auf einen Platz im Be-/Entlademagazin gesetzt wird.
- Zustand "reserviert für WZ aus Zwischenspeicher" wird bei der Leeplatzsuche dann zurückgesetzt, wenn das WZ, für das die Leerplatzsuche erfolgt, einen anderen als den bisherigen realen Magazinplatz zugewiesen bekommt. Der neu gefundene Leerplatz erhält stattdessen den Zustand "reserviert für WZ aus Zwischenspeicher" und wird neuer Eigentümer des WZ, für das die Suche erfolgte.

Der Magazinplatzzustand "reserviert für zu beladendes Werkzeug" wird beim Wiedereinschalten der Steuerung stets zurückgesetzt. Bei aktiver Nebenplatzbetrachtung werden auch die entsprechenden Reservierungen der Nebenplätze mitberücksichtigt.

Nur wenn der Benutzer die Magazindefinition direkt auf NC-Programmebene vornehmen will, muss er sich mit diesen Regeln auseinandersetzen. Die Datensicherung erfolgt derart, dass die Regeln beim Einspielen der Daten nach NC eingehalten werden.

$TC_MPP5 (Platzartindex)

Dieses Datum enthält bei Magazinplätzen von der Art "Spindel" ($TC_MPP1) die Spindelnummer, die dadurch der Werkzeugverwaltung bekannt gemacht wird.

Der Wert kann nicht verändert werden für Platzart = 1 ($TC_MPP1; d. h., für alle Plätze der internen Magazine), wenn ein Werkzeug auf dem Platz ist.

$TC_MPP6 (T-Nr)

- Die Werkzeuge können erst auf die Magazinplätze gesetzt werden, wenn sowohl das Werkzeug als auch das Magazin samt Magazinplätzen definiert ist.

Das Werkzeug darf höchstens auf einem Magazinplatz enthalten sein!

Vorgehensweise:

Zunächst wird versucht, das zur T-Nr. gehörende Werkzeug zu finden.

- Ist es bereits definiert, dann wird versucht, es - mit den nötigen Prüfungen - dem Magazinplatz hinzuzufügen.
- Ist es noch nicht definiert, liegt ein Fehler vor.

Prüfungen:

- Das zu plazierende Werkzeug muss vom Typ her zum Typ des Platzes passen. Ist der Typ zum Schreibzeitpunkt noch nicht explizit gesetzt (Vorbesetzung = 9999 = "nicht definiert"), wird das Werkzeug nicht plaziert.
- Der Zustand des Platzes muss "frei" und darf nicht "gesperrt" sein.
- Wenn der Wert T-Nr.=0 programmiert wird, bedeutet dies, dass das vorhandene Werkzeug vom Magazinplatz entfernt wird.
  Achtung: $TC_MPP6 = 0 ändert auch den Zustand des Platzes: Ein Werkzeug kann nur auf einen Magazinplatz kommen, wenn der Platz nicht bereits ein Werkzeug enthält. Das alte Werkzeug muss gegebenenfalls mit $TC_MPP6 = 0 erst entfernt werden.

Hinweis

Aufgrund dieser Abhängigkeit der einzelnen Daten ist es zwingend erforderlich, die T-Nr. des Werkzeugs als letztes Datum einer Magazinkonfiguration zu schreiben. Hält man sich nicht an diese Reihenfolge, so werden eventuell Voreinstellwerte gesetzt, die zu unerwünschten Daten führen können.

11.6.5 Magazinplatz-Anwenderdaten $TC_MPPC

$TC_MPPCx[n,m]

Magazinplatz-Anwenderdaten

Pro Magazin können zusätzlich bis zu 64 (ab V4.7) Anwenderdaten angelegt werden.
Einstellung für Parameteranzahl im MD18092 $MN_MM_NUM_CC_MAGLOC_PARAM und Freigabe mit MD18080 $MN_MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

x: = Parameter 1...64

n: = Magazin-Nummer 1...30

m: = Magazinplatz-Nummer 1...32000

BTSS-Baustein TUP

Berechnung der Zeile: (m-1)*numMagLocParams_u+Parameternr.

Berechnung der Spalte: Magazinnummer

Magazinplatzdaten OEM-Anwender					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_MPPC1	INT		userplaceData	DINT	0
...	INT		userplaceData	DINT	0
$TC_MPPC64	INT		userplaceData	DINT	0

11.6.6 Magazinplatztyphierarchie $TC_MPTH

$TC_MPTH[n,m]

Magazinplatztypenhierarchie

Die Platztypen können durch Programmierung dieser Systemvariablen in eine Hierarchie gebracht werden.

n: = Index der Hierarchie, von 0...$MN_MM_MAX_NUM_OF_HIERARCHIES

m: = Index innerhalb der Hierarchie n, Platztyp 0...$MN_MM_MAX_HIERARCHY_ENTRIES

Magazinplatztypen siehe auch $TC_TP7 und $TC_MPP2.

BTSS-Baustein TT

Berechnung der Zeile: Nummer des Platztyps+1

Berechnung der Spalte: Nummer der Platzhierarchie+1

Magazindaten: Magazinplatztypenhierarchie					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_MPTH[n,m]	INT	Platztyphierarchisierung n: Hierarchie 0...$MN_MM_MAX_NUM_OF_HIERARCHIES - 1 m: Platztyp 0...$MN_MM_MAX_HIERARCHY_ENTRIES - 1	placeType	WORD	9999

Soll ein Werkzeug in das Magazin eingewechselt werden, so entscheidet der Platztyp, welche Plätze zur Wahl stehen, d. h., $TC_TP7 und $TC_MPP2 müssen definiert sein.

Ist der Platztyp des Werkzeugs Teil einer Platztyp-Hierarchie, so wird die Platzvergabe gemäß dieser Hierarchie vorgenommen.

In einer TO-Bereichseinheit können mehrere solcher Hierarchien angelegt werden. Bei konventioneller Hierarchie darf aber ein Platztyp nur in eine Hierarchie eingetragen werden.

Beispiel

Ein Kettenmagazin soll in 6 Platztypen unterteilt werden und folgende Hierarchie definiert werden (die Magazin-Nr. ist "1", die Nummern der Platztypen sind willkürlich gewählt).

Platztyp_124 < Platztyp_3 < Platztyp_15 < Platztyp_1080 < Platztyp_5 <Platztyp_18

Definitionen:

Magazin:

$TC_MPP2[Magazinnr, Platz]

$TC_MPP2[1,1...6] = 124

$TC_MPP2[1,7...12] = 3

$TC_MPP2[1,13...18] = 15

$TC_MPP2[1,19...24] = 1080

$TC_MPP2[1,25...30] = 5

$TC_MPP2[1,31...36] = 18

Hierarchie:

$TC_MPTH[0,0] = 124

$TC_MPTH[0,1] = 3

$TC_MPTH[0,2] = 15

$TC_MPTH[0,3] = 1080

$TC_MPTH[0,4] = 5

$TC_MPTH[0,5] = 18

Wird ein Werkzeug des Typs_15 ($TC_TP) beladen, so wird es vorzugsweise auf den Plätzen 13...18 abgelegt. Ist keiner dieser Plätze frei, so wird die Leerplatzsuche, entsprechend der Hierarchie bei Plätzen des Typs_1080 fortgesetzt.

11.6.7 Abstand zur Wechselstelle $TC_MDP

$TC_MDPx[n,m]

Abstand vom Magazinnullpunkt

$TC_MDPx[n,m]=Wert

x: = 1: Belademagazin: Beladestellen, Beladestation (1. int. Mag.)

x: = 2: Zwischenspeichermagazin: Spindel, Greifer,..(2. int. Mag.)

n: = Magazin-Nr. des realen Magazins

m: = Platz-Nr. des internen Magazins (Beladestelle,..).

Wert: = Distanz in Anzahl Plätze

BTSS-Baustein TPM

Berechnung der Zeile:

(PlatzNr-1)*numPlaceMulti*numPlaceMultiParams+ParameterNr

Berechnung der Spalte: Magazinnummer

Magazindaten: Abstand zur Wechselstelle				
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$TC_MDP1	INT	Abstand zur Wechselstelle des Magazins n zum Platz m des 1. internen Magazins (Beladem agazin, 9999)	multiPlace	WORD
$TC_MDP2	INT	Abstand zur Wechselstelle des Magazins n zum Platz m des 2. internen Magazins (Zwischenspeichermagazin, 9998)	multiPlace	WORD

Beschreibung

Magazinposition

Beim Werkzeugwechsel, Beladen und Entladen benötigt man die aktuelle Magazinposition. Diese Position bezieht sich auf den vom Maschinenhersteller festgelegten Magazin-Nullpunkt. In der Regel liegt dieser an der Wechselstelle.

Bei der Initialisierung muss die Nummer des Platzes am Magazin-Nullpunkt angegeben werden. Ansonsten wird angenommen, dass sich der nicht existierende Platz 0 an der Wechselposition befindet.

Wird das Magazin durch einen Auftrag verfahren, so wird die aktuelle Position entsprechend verändert. Der NC hat keine Kenntnis, um wie viel Plätze das Magazin verfährt, er weiß aber die Ziele der entsprechenden Kommandos. Durch die Festlegung, welchen Abstand ein Objekt (z. B. Spindel 2) von der Wechselstelle hat, ist der NC in der Lage, die aktuelle Position zu aktualisieren.

Anmerkung:

Der Wert der Distanz und die aktuelle Magazinposition werden auch für Flächenmagazine ausgewertet.

Bei der Leerplatzsuche und der Werkzeugsuche wird bei den Suchstrategien, die sich auf die aktuelle Magazinposition beziehen, die in der Systemvariablen $TC_MAP8 enthaltene Position jeweils auf die Wechselstelle, Beladestelle umgerechnet, von der ausgehend die Suche erfolgt. Bei Suchaufträgen wird NC-intern immer mit angegeben, bezogen, auf welche Wechselstelle, Beladestelle gesucht werden soll.

Zuordnung

Durch die Definition von $TC_MDPx[n,m] wird eine Verbindung oder Zuordnung (Distanzbeziehung) hergestellt, zwischen dem realen Magazin und den Zwischenspeichern bzw. den Beladestellen. So kann das HMI ausschließlich zugeordnete Zwischenspeicherplätze und Beladestellen anzeigen. Ein Werkzeugtransport im Rahmen einer t-Vorbereitung oder eines Wechsels kann nur über zugeordnete Zwischenspeicher erfolgen (Ausnahme: asynchrone Transfers). Ebenso erfolgt die Werkzeugsuche, bei T-Vorbereitung und Wechsel, nur in denen zur Spindel bzw. Toolholder zugeordneten Magazinen.

Mit dem Maschinendatum $MN_MM_NUM_LOCS_WITH_DISTANCE kann die maximale Anzahl dieser Distanzbeziehungen festgelegt werden.

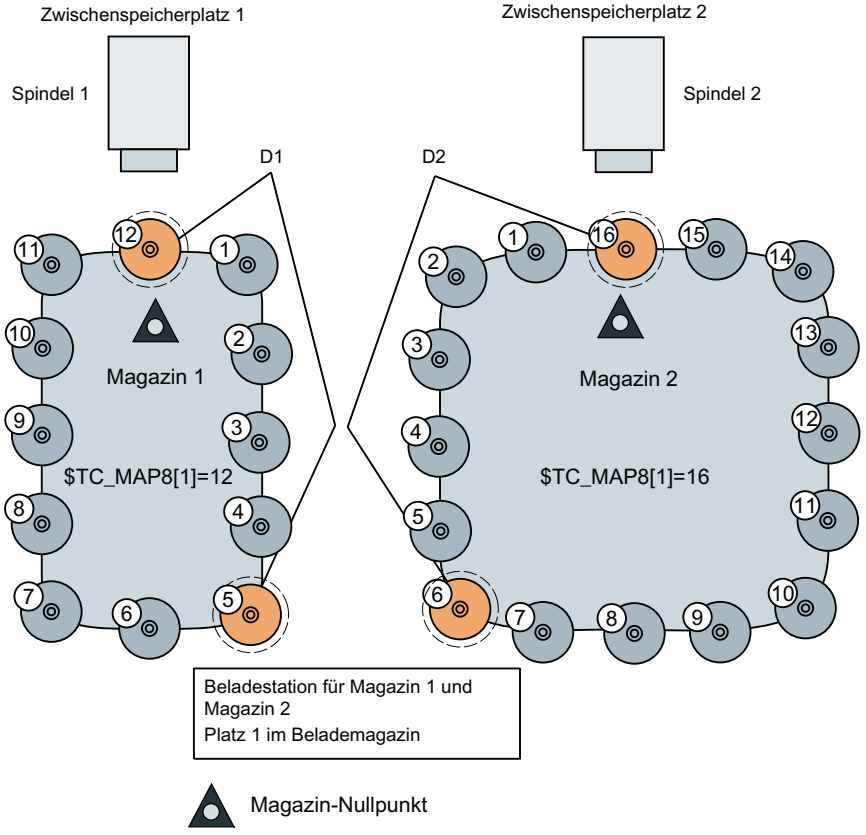
Beispiel: 1. TO-Einheit, 1 Magazin, 1x Spindel, Doppelgreifer, 2x Beladestelle. Das macht insgesamt 5 Distanzbeziehungen (3x $TC_MDP2 für die Zuordnung des realen Magazins zu den

3 Zwischenspeichern, 2x $TC_MDP1 für die Zuordnung des realen Magazins zu den beiden Beladestellen).

Hinweis

Mit dem Befehl $TC_MDP2[n,m]=9999 kann eine Distanzbeziehung aufgelöst werden.

Beispiel:



D1 = Abstand vom Magazinnullpunkt zur Beladestation -> 5 Plätze = $TC_MDP1[1,1]

D2 = Abstand vom Magazinnullpunkt zur Beladestation -> 6 Plätze = $TC_MDP1[2,1]

Bild 11-7 Abstand zur Wechselstelle $TC_MDPx[y,z]=Wert

Normalerweise ist der Magazin-Nullpunkt die Wechselstelle der Spindel. Somit gilt folgende Aussage:

- Wenn Platz 1 an der Nullpunktpositon steht, ist die aktuelle Magazinposition = 1 ($TC_MAP8[1]).

Beispiele für die Programmierung des Abstandes zur Nullpunktposition:

$TC_MDP1[1,1] = 5	Abstand von Platz 1 der Beladestation zur Nullpunktposition von Magazin 1
$TC_MDP1[2,1] = 6	Abstand desselben Platzes zur Nullpunktposition von Magazin 2

$TC_MDP2[1,1] = 0	Abstand von Platz 1 des 2. internen Magazins zur Nullpunktposition von Magazin 1
$TC_MDP2[2,2] = 0	Abstand von Platz 2 des 2. internen Magazins zur Nullpunktposition von Magazin 2

11.6.8 Magazinbausteine $TC_MAMP

$TC_MAMPx

Magazinbausteindaten

x: = Parameter 1, 2, 3

BTSS-Baustein TMC

Berechnung der Zeile: Entfällt

Berechnung der Spalte: Entfällt

Magazinbausteindaten, Magazin-Kontrollblock					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbe-legung**
$TC_MAMP1	String	Bezeichner des Konfigurations-magazins	magCBIdent	String	" "
		Nummer des BeladeMagazins	magBLMag	WORD	
		Nummer des Zwischenspeicher-magazins	magZWMag	WORD	

Magazinbausteindaten, Magazin-Kontrollblock					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_MAMP2	INT	Art der WZ-Suche (Bit 0...7) und Art der Leerplatzsuche (Bit 8...16) Bit 0=0: Defaultstrategie Nimm das erste verfügbare WZ das in der WZ-Gruppe gefunden wird. Suche zuerst in dem Magazin, aus dem der letzte Wechsel erfolgte. Bit 0=1: Wähle das "aktive" WZ im Magazin des zuvor gewechselten WZ, sonst suche das Ersatzwerkzeug mit kleinster Duplonummer. Falls in diesem Magazin kein WZ gefunden wird, wird die Suche in den anderen verbundenen Magazinen fortgesetzt. Bit 1: Suche das nächste Ersatzwerkzeug aus, das die kürzeste Entfernung von der aktuellen Magazinposition hat. Bit 2: Wähle das "aktive" WZ, sonst Ersatzwerkzeug mit der kleinsten in $TC_TP10 enthaltenen Nummer (funktioniert nicht bei Handwerkzeugen). Bit 3: Suche das Werkzeug in der Gruppe, mit dem **kleinsten Istwert** der überwachten Größe. Bit 4: Suche das Werkzeug in der Gruppe, mit dem **größten Istwert** der überwachten Größe.	magSearch		0
$TC_MAMP2	INT	Bit 5: reserviert Bit 6: Suche vorrangig im aktuell betrachteten Magazin (wirkt nur in Verbindung mit Bit 7=1) Bit 7=0: Beginn der WZ-Suche im Magazin, aus dem das zuletzt gewechselte WZ stammt Bit 7=1: Beginn der Suche immer im 1. Magazin der Distanztabelle Bit 7= 1 + Bit 0=1 oder Bit 2=1, falls kein "aktivesWZ" im Magazin gefunden wird, dann wird - falls vorhanden - das aktive WZ aus einem anderen, mit dem WZ-Halter verbundenen Magazinen angewählt	magSearch		0

Magazinbausteindaten, Magazin-Kontrollblock					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbe-legung**
$TC_MAMP2		Bit 8 Vorwärts-Suche bei erster Platz-nummer beginnend Bit 9 Vorwärts-Suche bei aktueller Ma-gazinposition beginnend Bit 10 Rückwärts-Suche bei letzter Platznr. beginnend			
$TC_MAMP2		Bit 11 Rückwärts-Suche bei aktueller Magazinposition beginnend Bit 12 symmetrische Suche bei aktuel-ler Magazinposition beginnend Bit 13 (1:1-Tausch) Verhalten Falls der 1:1-Tausch nicht mög-lich ist, gilt die Leerplatzsuchstra-tegie "symmetrisch". Der 1:1-Tausch wirkt zusätzlich zu anderen eingestellten Such-strategien. Wenn möglich wird der 1:1-Tausch vorrangig behan-delt. Bit 14 = 0 Suche vorrangig in den einzelnen Magazinen. Wenn kein mögli-cher Platz gefunden wird, suche im nächsten Magazin nach einen freien Platz für das Werkzeug. Bit 14 = 1 Suche in allen Magazinen nach dem besten Platz entsprechend der Hierarchie für das Werkzeug.			

Magazinbausteindaten, Magazin-Kontrollblock					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
		Bit 15 = 0 (Konventioneller Hierarchieart): Bei dieser Art wird der Platztyp des suchenden Werkzeuges in der Tabelle der Systemvariablen $TC_MPTH gesucht. Wird der Platztyp gefunden, wird diese Hierarchie genommen und von dieser Stufe bis zum Ende ausgewertet. Bit 15 = 1 (Alternative Hierarchieart): Für die ausgezeichneten Platztypen 1, ..., $MN_MM_MAX_NUM_OF_HIERARCHIES können Platztyphierarchien definiert werden. Die Hierarchie für Platztyp 1 wird durch $TC_MPTH[0,n] definiert, für den Platztyp 2 wird durch $TC_MPTH[1,n] definiert und so weiter. (n: Index innerhalb einer Hierarchie). Es können bei dieser Einstellung ein Platztyp in verschiedenen Hierarchien definiert sein.			
$TC_MAMP2	INT	Bit16: Die Mini-Hierarchie, die nur aus dem Platztyp selbst und dem Platztyp 0 besteht, hebt die Hierarchie-Betrachtung auf. In diesem Fall wird bei der Leerplatzsuche nicht zwischen dem passenden Platztyp ($TC_TP7 == $TC_MPP2) und dem allgemeinen Platztyp "0" des Magazinplatzes unterschieden.			
$TC_MAMP3	INT	Behandlung der Werkzeuge in einem Verschleißverbund (Bit 0...7) Suchstrategien für Verschleißverbände (Bit 8...15) Bit 0=0: Beim Aktivsetzen eines Verschleißverbundes bleibt der Zustand der WZe unverändert Bit 0=1: Beim Aktivsetzen eines Verschleißverbundes wird der Zustand der Werkzeuge verändert. Aus jeder WZ-Gruppe wird ein WZ aktiv gesetzt	modeWearGroup	WORD	0

Magazinbausteindaten, Magazin-Kontrollblock					
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-Variable**	**Typ**	**Vorbelegung**
$TC_MAMP3		Bit 1=0: Beim Sperren eines Verschleißverbundes bleibt der Zustand der WZe unverändert Bit 1=1: Beim Sperren eines Verschleißverbundes wird der Zustand der WZe verändert Bit 2...7 reserviert Suchstrategie für nächsten Verschleißverbund Bit 8=0: finde den nächstmöglichen Verschleißverbund Bit 8=1: finde den Verschleißverbund mit der nächsthöheren aktivierbaren Verbundnummer Bit 9...11 reserviert			
$TC_MAMP3		Suchstrategie innerhalb der WZ-Gruppe für das aktiv zu setzende WZ (Sprachbefehl SETTA bzw. PI_SETTST) Bit 12=0: kleinste mögliche Duplonummer Bit 12=1: kleinste mögliche Magazinplatznummer Bit 13=1: kleinste in $TC_TP10 (Ersatzreihenfolge) enthaltene Nr.			

Zusammenwirken der Bits für die Werkzeugsuchstrategie

Folgende Strategien beginnen die Suche immer im ersten Magazin der Distanztabelle.

$TC_MAMP2 =		**Bedeutung**
Bit	**Hex.-Wert**	
7 + 0	'H80'	sonst gleich wie Wert 0
7 + 0	'H81'	Sonst gleich wie Wert 1 - falls allerdings kein "aktives" WZ im Magazin gefunden wird, dann wird - falls vorhanden - das "aktive'"WZ aus einem anderen mit dem WZ-Halter verbundenen Magazin angewählt. Ist kein WZ mit dem Zustand "aktiv" verfügbar, so wird das Ersatz-WZ mit der kleinsten Duplo-Nummer gesucht.
7 + 6 + 0	'HC1'	Sonst gleich wie Wert 1 - falls allerdings kein "aktives" WZ im Magazin gefunden wird, dann wird im gleichen Magazin nach einem einsetzbaren Werkzeug gesucht, falls nicht vorhanden wird in den anderen mit dem WZ-Halter verbundenen Magazinen gesucht.
7 + 1 7 + 6 + 1	'H82' 'HC2'	Sonst gleich wie Bit 1 =1 ('H2')

$TC_MAMP2 =		Bedeutung
7 + 2	'H84'	Sonst gleich wie Bit 2 =1 ('H4') - falls allerdings kein "aktives" WZ im Magazin gefunden wird, dann wird - falls vorhanden - das 'aktive' WZ aus einem anderen mit dem WZ-Halter verbundenen Magazin angewählt. Ist kein WZ mit dem Zustand "aktiv" verfügbar, so wird das Ersatz-WZ mit der kleinsten Nummer in $TC_TP10 gesucht.
7 + 6 + 2	'HC4'	Sonst gleich wie Bit 2 =1 ('H4') - falls allerdings kein "aktives" WZ im Magazin gefunden wird, dann wird im gleichen Magazin nach einem einsetzbaren Werkzeug gesucht, falls vorhanden wird in den anderen mit dem WZ-Halter verbundenen Magazinen nach dem Ersatz-WZ mit der kleinsten Nummer in $TC_TP10 gesucht.
7 + 3 7 + 6 + 3	'H88' 'HC8'	Sonst gleich wie Bit 3 = 1 ('H8')
7 + 4 7 + 6 + 4	'H90' 'HD0'	Sonst gleich wie Bit 4 = 1 ('H10')

Suchvorgang für $TC_MAMP2, Bit 0=1, Bit 6=0, Bit 7=1

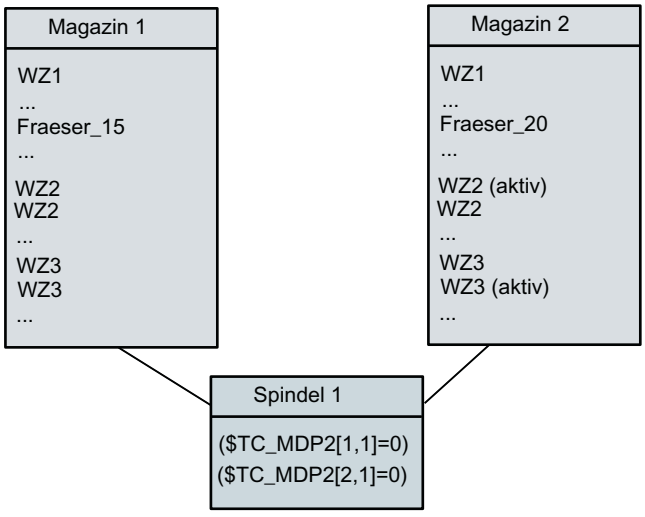
Innerhalb der Magazine wird gemäß der eingestellten WZ-Suchstrategie (durch den Parameter $TC_MAMP2 definiert) nach dem passenden Werkzeug gesucht. Die Suche gilt als erfolgreich, wenn ein Werkzeug gemäß der Suchstrategie gefunden wurde.

Suchvorgang für $TC_MAMP2, Bit 0=1, Bit 6=1, Bit 7=1

Innerhalb der Magazine wird zunächst gemäß der eingestellten WZ-Suchstrategie (durch den Parameter $TC_MAMP2 definiert) nach dem passenden Werkzeug gesucht. Die Suche gilt als erfolgreich, wenn ein Werkzeug gemäß der Suchstrategie gefunden wurde bzw. alle Werkzeuge der Gruppe geprüft wurden und kein Werkzeug gemäß der Suchstrategie gefunden wurde und sich trotzdem ein einsetzbares unter den durchsuchten Werkzeugen befindet.

Wird nach Strategie Bit 2=1 (Einsatzreihenfolge $TC_TP10) gesucht, gilt dasselbe Prinzip. Das Kriterium ist ebenfalls "aktive WZ" und dann anstelle der Duplo-Nr. die Nummer in $TC_TP10.

Beispiel zur Verdeutlichung der Wirkungsweise von Bit 6 und Bit 7



Annahme:

- Alle Werkzeuge sind einsatzfähig
- Grundeinstellung der Suchstrategie ist Bit 0=1 - suche das aktive Werkzeug, gibt es kein aktives, nimm das mit der kleinsten Duplo-Nr.
- Spindel_1 ist mit beiden Magazinen verbunden, das Magazin_1 ist das Erste in der Distanztabelle

Beispiel_1:

Bit 6 = 0
Bit 7 = 0

T="Fraeser_15" M06	Das Werkzeug "Fraeser_15" gibt es nur einmal, es wird im Magzin_1 gefunden und eingewechselt
...	
T="WZ3" M06	Aus dem Magazin_1 erfolgte der letzte Wechsel. Also wird zuerst im Magazin_1 gesucht. Dort gibt es kein aktives "WZ3", aber 2 einsatzfähige Werkzeuge dieser Gruppe. Es wird das mit kleinster Duplo-Nr. ausgewählt. Das Magazin_2 wird nicht mehr betrachtet.

Beispiel_2:

Bit 6 = 0
Bit 7 = 1

T="Fraeser_20" M06	Das Werkzeug "Fraeser_20" gibt es nur einmal, es wird im Magzin_2 gefunden und eingewechselt
...	
T="WZ3" M06	Aus dem Magazin_2 erfolgte der letzte Wechsel. Aufgrund von Bit 7 = 1 beginnt die WZ-Suche jedoch im Magazin_1 Diese Einstellung bedeutet aber auch, dass die Suchstrategie "aktives Werkzeug" Vorrang vor der magazinspezifischen Betrachtung hat. Das bedeutet: Im Magazin_1 wird kein aktives Werkzeug aus der Gruppe "WZ3" gefunden, also wird die Suche im nächsten Magazin (entsprechend der Reihenfolge in der Distanztabelle) fortgesetzt. Dort gibt es ein aktives "WZ3" - das wird ausgewählt.

Beispiel_3:

Bit 6 = 1
Bit 7 = 1

T="Fraeser_20" M06	Das Werkzeug "Fraeser_20" gibt es nur einmal, es wird im Magzin_2 gefunden und eingewechselt
...	
T="WZ3" M06	Aus dem Magazin_2 erfolgte der letzte Wechsel. Aufgrund von Bit 7 = 1 beginnt die WZ-Suche im Magazin_1 Dort gibt es jedoch kein aktives Werkzeug dieser Gruppe. Aufgrund von Bit 6 = 1 (betrachtet vorrangig das aktuelle Magazin) wird jetzt ein einsatzfähiges "WZ3" im Magazin_1 (das ist das für diese Suche das aktuelle Magazin) gesucht und gefunden.

$TC_MAMP2

Bei Software-Ständen kleiner 2.5 erfolgt die Werkzeugsuche grundsätzlich magazinspezifisch, beginnend in dem Magazin aus dem der letzte Wechsel erfolgt ist.

Mit dem Bit 7 gibt es eine weitere Einstellmöglichkeit der Werkzeugsuche.

Bit 7 = 1

Die Suche beginnt immer im 1. Magazin der Distanztabelle. Ist mit Bit 0 oder Bit 2 die Suche nach dem aktiven Werkzeug eingestellt, so gilt, dass das aktive Werkzeug über alle mit der Spindel verbundenen Magazine gesucht wird. Erst wenn in allen diesen Magazinen kein aktives Werkzeug gefunden wird, beginnt die Suche nach einem Schwesternwerkzeug.

Hinweis

Die Werkzeug-Reihenfolge in einer WZ-Gruppe ist nicht definiert (z. B. aufsteigende Duplo-Nr.). D. h., wird nach der Default-Strategie (MAMP2=0) gesucht, wird ein beliebiges, einsatzfähiges Werkzeug gefunden.

11.6.9 Zuordnung von Zwischenspeichern zu Spindeln $TC_MLSR

$TC_MLSR[x,y]

Zuordnung von Zwischenspeicherplätzen zu Spindeln - $TC_MLSR[x,y]

x: = Platz-Nr. im Zwischenspeicher 1... 32000

y: = Platz-Nr. der Spindel im Zwischenspeichermagazin 1... 32000

BTSS-Baustein entfällt

Berechnung der Zeile: Entfällt

Berechnung der Spalte: Entfällt

NC-Bezeichner	Typ	Bezeichnung	BTSS-Variable	Typ	Vorbelegung
$TC_MLSR[x,y]	INT	Systemvariable zur Zuordnung von Magazinplätzen des ZWS-Mag. zur Spindel	-	-	0

Beschreibung

Diese Zuordnung ist für die Werkzeugsuche wichtig. Die Suche erfolgt immer, von der anfordernden Spindel aus gesehen, zuerst in den zugeordneten Zwischenspeichern, dann in den zugeordneten Magazinen. D. h., ein Werkzeug, das im Zwischenspeicher sitzt, kann nur gefunden werden, wenn über dem Parameter $TC_MLSR eine Zuordnung zur Spindel besteht.

Die Anzahl möglicher Verbindungen Spindel zu Zwischenspeicher, kann über das Maschinendatum $MN_MM_DIST_REL_PER_MAGLOC eingestellt werden.

Beispiel: 1. TO-Einheit, 1 Magazin, 1x Spindel, Doppelgreifer, 2x Beladestelle. Über $TC_MLSR[2,1]=0 und $TC_MLSR[3,1]=0 wird der Doppelgreifer mit der Spindel verbunden. Der korrekte Wert für das MD ist "2".

Die Reihenfolge der Programmierung ist maßgebend für die Reihenfolge beim automatischen WZ-Rücktransport.

Hinweis

Der Wert der Systemvariablen wird inhaltlich nicht ausgewertet. Die Zuordnung wird über die Angabe der Indices x und y festgelegt. Um über das Teileprogramm zu prüfen, ob eine bestimmte Zuordnung besteht, muss eine Leseoperation den Wert Null liefern.

Hinweis

Einer Spindel können nicht mehr als 16 Magazine bzw. Zwischenspeicherplätze zugeordnet werden.

Magazindistanz zum Zwischenspeicher über WZ-Halter/Spindel

$TC_MDP2 und $TC_MLSR setzen die Zwischenspeicherplätze und Magazine in Beziehung zueinander.

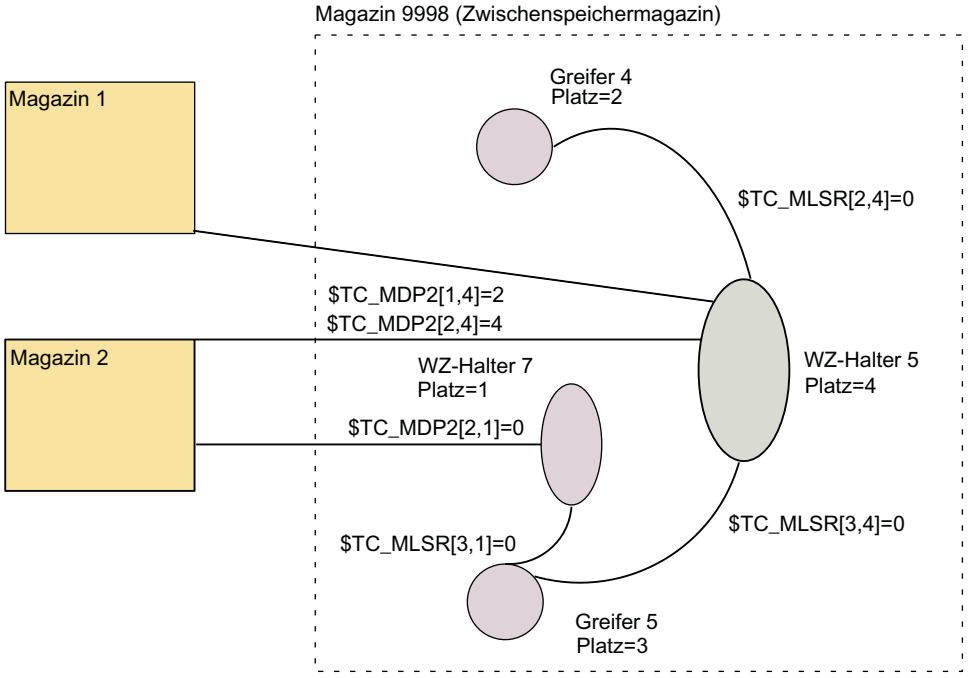


Bild 11-8 Magazindistanz zum Zwischenspeicher

Konfiguration

Es sind zwei Magazine mit den Nummern 1 und 2 definiert.

Im Zwischenspeicher 9998 sind vier Plätze 1, 2, 3 und 4 definiert; zwei WZ-Halter 5, 7 und zwei Greifer 4, 5.

```
$TC_MPP1[9998,1] = 2          ;Platzart = Spindel bzw. WZ-Halter
$TC_MPP5[9998,1] = 7          ;WZ-Halternr. = 7
$TC_MPP1[9998,2] = 3          ;Platzart = Greifer
$TC_MPP5[9998,2] = 4          ;Greifernr. = 4
$TC_MPP1[9998,3] = 3          ;Platzart = Greifer
$TC_MPP5[9998,3] = 5          ;Greifernr. = 5
$TC_MPP1[9998,4] = 2          ;Platzart = Spindel bzw. WZ-Halter
$TC_MPP5[9998,4] = 5          ;WZ-Halternr. = 5
```

Die beiden Greifer sind mit dem WZ-Halter 5 durch $TC_MLSR verbunden. Sie benötigen keine eigene Distanzdefinition zu den Magazinen. Über den WZ-Halter 5 sind sie mit den dort definierten Distanzbeziehungen mit den Magazinen verbunden. Es ist aber möglich, eigene Distanzbeziehungen für die Greifer zu definieren.

WZ-Halter 5 ist mit den beiden Magazinen durch $TC_MDP2 verbunden.

WZ-Halter 7 ist nur mit Magazin 2 verbunden; ihm ist der Greifer 5 zugeordnet.

11.7 Multitooldaten

11.7.1 $TC_MTP - Multitooldaten

Übersicht

Ein Multitool ist eine Zusammenfassung mehrerer Werkzeuge zu einer Einheit. Die folgenden Parameter erlauben es, das Multitool so zu beschreiben, dass es sich in die Systematik der Funktion WZMG einfügt.

Alle Parameter der folgenden Tabelle haben keine physikalische Einheit. Das voreingestellte Zugriffsrecht ist "Schlüsselschalterstellung 0", sowohl für Zugriffe aus dem NC-Programm als auch für Zugriffe über BTSS. Mit dem Befehl REDEF können Zugriffsrechte eingeschränkt werden.

`$TC_MTPx[y]`

x: Parameter 1 ...8, ...PROTA

y: = Multitool-Nummer MT = 1 ...32000

NC-Bezeichner	Beschreibung	Format	Vorbelegung, Grenzwerte
$TC_MTPN	Anzahl Plätze und als Löschbefehl definiert	INT	2 2-$MN_MAX_TOOLS_PER_MULTITOOL
$TC_MTP2	Bezeichner	STRING	MT-Nummer
$TC_MTP3	Größe nach links in Halbplätzen (= zur kleineren Platznr. hin)	INT	1 1 - 11

NC-Bezeichner	Beschreibung	Format	Vorbelegung, Grenzwerte
$TC_MTP4	Größe nach rechts in Halbplätzen (= zur größeren Platznr. hin)	INT	1 1 - 11
$TC_MTP5	Größe nach oben in Halbplätzen (= zur kleineren Platznr. hin)	INT	1 1 - 11
$TC_MTP6	Größe nach unten in Halbplätzen (= zur größeren Platznr. hin)	INT	1 1 - 11
$TC_MTP7	Magazinplatztyp	INT	9999 (nicht definiert) 0 - 32000
$TC_MTP8	Zustand; bitcodiert	INT	0 (nicht frei gegeben) 0 - "0x3FFFF"
$TC_MTP_POS	MT-Position. Es gilt: Position 1 ist definiert als "Platz 1 in Bearbeitungsposition"	INT	0 (nicht definiert) 0-$MN_MAX_TOOLS_PER_MULTITOOL
$TC_MTP_KD	Art der Abstandskodierung	INT	0 (nicht definiert) 1 (= Platznummer) 2 (= Länge) 3 (= Winkel)
$TC_MTP_PROTA	Name des Schutzbereichs bzw. Name der Datei, die die Beschreibung des Schutzbereiches enthält.	STRING	""(= nicht definiert)

$TC_MTPN - Anzahl MT-Plätze

Das Multitool hat keine Duplonummer wie das Werkzeug. Der Parameter $TC_MTPN enthält die Anzahl an Plätzen, die das Multitool zur Aufnahme von Werkzeugen bietet. Zusätzlich wird durch Programmieren von $TC_MTPN[mtNr] = 0 das Multitool gelöscht. Nach erfolgter Erzeugung der MT-Plätze kann der Parameter nicht mehr verändert werden.

Ein Multitool ist bzgl. seiner Plätze einem Kettenmagazin verwandt, d. h., wenn es WZ-Suchstrategien oder Leerplatzsuchstrategien gibt, deren Ergebnisse von der Art des Magazins abhängen, kann sich die Eigenschaft wie bei einem Kettenmagazin bemerkbar machen.

$TC_MTP2 - MT-Name

Multitools können mit Namen versehen werden, sodass ein Multitool eindeutig bestimmbar durch seinen Namen, oder durch seine Nummer ist. Die Regeln zur Namensvergebung sind die gleichen, wie sie auch für den Werkzeugnamen gelten. Multitoolnamen müssen von Werkzeugnamen und Magazinnamen innerhalb einer TO-Einheit verschieden sein.

$TC_MTTP3 $TC_MTP6 - MT-Größe

Die Größenangabe des Multitools ist analog der Größenangabe eines Werkzeugs und dient dazu, für das Multitool beim Beladen in das Magazin (Leerplatzsuche) bei aktiver Nebenplatzbetrachtung einen optimal geeigneten Magazinplatz zu finden.

Die Regeln zum Ändern der Multitoolgröße entsprechen den Regeln zum Ändern der WZ–Größe, d.h. die MT-Größe eines Multitools kann verändert werden, solange das MT noch keinem Eigentümerplatz zugeordnet ist.

$TC_MTP7

Der Magazinplatztyp des Multitools ist anlog dem Magazinplatztyp eines einfachen Werkzeugs und wird für die Leerplatzsuche, Leerplatzprüfung benötigt (Beladevorgänge in das Magazin, WZ–Wechsel). Die erlaubten Werte und die Regeln zum Ändern des Datums entsprechen denen, die für das korrespondierende Datum im WZ gelten. Der Wert kann nicht geändert werden, wenn das Multitool im Magazin beladen ist.

Multitools, die in Beladem agazinen bzw. Zwischenspeichermagazinen enthalten sind und die noch keinen Eigentümerplatz haben (d. h. die logisch als unbeladen gelten – $A_MYMN=0, $A_MYMLN=0), darf der Magazinplatztyp verändert werden. Der Magazinplatztyp eines Multitools kann verändert werden, solange das Multitool noch keinem Eigentümerplatz zugeordnet ist.

$TC_MTP8 - MT-Zustand

Der Zustand eines Multitools entspricht dem Zustand eines einzelnen Werkzeugs. Die definierten Zustände sind die gleichen wie eines einzelnen Werkzeugs wie unter $TC_TP definiert ist.

Es gelten folgende weitere Festlegungen:

Bit	Bedeutung	Kommentar
0	aktiv	Reserviert Multitools haben den Zustand "aktiv" nicht definiert. Nur darin enthaltene Werkzeuge können den Zustand einnehmen.
1	frei gegeben	Bestückte Multitools haben diesen Zustand, damit darin ein Werkzeug angewählt werden kann.
2	gesperrt	Multitool kann gesperrt werden. Unabhängig vom Zustand der darin enthaltenen Werkzeuge kann damit die WZ-Anwahl dieser Werkzeuge verhindert werden. Das PLC-Signal "ignoriere Gesperrt-Zustand" ignoriert auch den Gesperrt-Zustand des Multitools. Der Werkzeug-Wechselbefehl TCA für ein Werkzeug ignoriert den Gesperrt-Zustand des Multitools.
3	vermessen	Reserviert Multitools können nicht vermessen werden. Nur darin enthaltene Werkzeuge können den Zustand einnehmen.
4	Vorwarngrenze erreicht	Reserviert Multitools können nicht überwacht werden. Nur darin enthaltene Werkzeuge können den Zustand einnehmen.

Bit	Bedeutung	Kommentar
5	befindet sich im Wechsel	Multitool während der WZ-Anwahl vor konkurrierender weiterer WZ-Anwahl / MT-Bewegung schützen. Damit ist es nicht möglich, ein WZ innerhalb des MT anzuwählen, solange dieser Zustand wegen der Anwahl eines WZ im MT gesetzt ist. Falls der Zustand gesetzt ist, ist eine weitere WZ-Anwahl für ein WZ aus dem MT erlaubt, falls diese Anwahl bzgl. desselben WZ-Halters erfolgt, wie die Anwahl, die den Zustand des MT gesetzt hat. Es liegt dann keine konkurrierende WZ-Anwahl vor, sondern eine ersetzende. Falls der Zustand gesetzt ist, hat ein WZ in diesem MT ebenfalls diesen Zustand gesetzt. Der Zustand wird zusammen mit dem des Werkzeugs wieder rückgesetzt. Bit 21 von $TC_TOOL_MANAGEMENT_MASK bezieht sich auf das WZ und das MT. Bei POWER ON wird der Zustand zurückgesetzt.
6	festplatzcodiert	Multitools können festplatzcodiert sein
7	war im Einsatz	Multitools haben den Zustand analog des Werkzeugs definiert. Nach dem Auswechseln eines Werkzeugs vom WZ-Halter erhalten das MT und das Alt-WZ den Zustand.
8	autom. Rücktransport	Multitools verhalten sich bzgl. des WZ-Transports wie einfache Werkzeuge. $TC_TOOL_MANAGEMENT_MASK, Bit 15 bezieht sich auch auf diesen Zustand.
9	ignoriere gesperrt	Der Zustand wird intern gesetzt und rückgesetzt. Die Regeln sind analog den Regeln für das WZ.
10	MT ist zu entladen	Multitool kann für das Entladen markiert werden
11	MT ist zu beladen	Multitool kann für das Beladen markiert werden
12	reserviert	reserviert
13	reserviert	reserviert
14	Für 1:1-Tausch markiert	Multitool ist für den bevorstehenden 1:1 Tausch markiert. Es gelten die Regeln wie für das WZ.
15	Handwerkzeug	Das Multitool wird als Handwerkzeug verwendet. Der Zustand wird für das MT im Rahmen der WZ-Anwahl gesetzt und wieder rückgesetzt. Anzuwählendes WZ und dessen MT ändern den Zustand synchron. Die an der WZ-Anwahl nicht beteiligten Werkzeuge erfahren keine Zustandsänderungen.
16	MT wird gesperrt, falls ein WZ im MT gesperrt wird	Falls ein im MT enthaltenes WZ den Zustandsübergang nach "gesperrt" gemacht hat, nimmt auch das MT diesen Zustand ein (Bit 2). Nachdem ein WZ den Statusübergang nach "nicht gesperrt" gemacht hat, und es anschließend kein WZ mehr im Multitool gibt, das den Zustand "gesperrt" hat, verliert auch das MT selbst den Zustand "gesperrt". Das Bestücken/Herauslösen eines gesperrten / nicht gesperrten Werkzeugs in das MT bzw. aus dem MT kann den MT-Zustand "gesperrt/nicht gesperrt" ändern. Beispiel: Bestücken in einem gesperrten WZ entspricht einem MT-Zustandsübergang nach "gesperrt".

Hinweis

Reservierte bzw. nicht definierte MT-Zustandsbits können nicht beschrieben werden. Das Schreiben eines solchen Bits führt zum Alarm 17050 "Unerlaubter Wert".

Abhängigkeiten von Werkzeug- und Multitool-Zuständen

Werkzeuge, die im Multitool bestückt sind, können den Zustand des Multitools beeinflussen und das MT kann bei MT-Transportvorgängen den WZ-Status der bestückten Werkzeuge beeinflussen.

Bit	Bedeutung	Kommentar
1	frei gegeben	Werkzeuge und Multitools werden unabhängig voneinander vom Anwender mit dem Status versehen. (Es können frei gegebene Werkzeuge in einem nicht frei gegebenen Multitool bestückt sein. Es können nicht frei gegebene Werkzeuge in einem frei gegebenen MT bestückt sein.)
2	gesperrt	Die Änderung dieses Zustands wird vom WZ aus veranlasst im Rahmen der WZ–Überwachung und des Befehls RESETMON. Wie der Zustand der bestückten Werkzeuge den des Multitools beeinflusst, zeigen die Beispiele 1a, 1b, 2a, 2b am Ende dieser Tabelle. Anwenderspezifische Änderungen des MT–Zustands werden nicht auf die bestückten Werkzeuge übertragen.
5	befindet sich im Wechsel	Der Zustand wird für das MT im Rahmen der WZ–Anwahl gesetzt und wieder rückgesetzt. Anzuwählendes WZ und dessen MT ändern den Zustand synchron. Die an der WZ–Anwahl nicht beteiligten Werkzeuge erfahren dabei keine Zustandsänderung. Bei POWER ON wird der Zustand stets zurückgesetzt.
6	festplatzcodiert	Im MT bestückte Werkzeuge sind implizit festplatzcodiert, d. h., der Zustand hat für Werkzeuge im MT keine Bedeutung. Für das MT selbst ist der Zustand definiert, hat aber keinen Einfluss auf die bestückten Werkzeuge. Eine Zustandsänderung für das WZ hat keine Auswirkung auf den Zustand des Multitools.
7	war im Einsatz	Der Zustand wird im Rahmen des WZ–Wechsels für das MT und das am Wechsel beteiligte WZ im MT automatisch gesetzt. Nicht am WZ–Wechsel beteiligte Werkzeuge im MT ändern den Zustand nicht. Für das Rücksetzen ist der Anwender verantwortlich. Verliert das Letzte der im MT enthaltenen Werkzeuge diesen Zustand, dann verliert auch das MT diesen Zustand. Löscht der Anwender den Zustand des Multitools, so wird der Zustand auch für alle enthaltenen Werkzeuge gelöscht. Setzt der Anwender diesen MT-Zustand, so hat das keine Auswirkungen auf die im MT bestückten Werkzeuge.
8	autom. Rücktransport	Der Zustand wird automatisch beim WZ–Wechsel nur für das MT gesetzt und rückgesetzt, nicht aber für die im MT enthaltenen Werkzeuge.
9	ignoriere gesperrt	Der Zustand wird automatisch gesetzt bzw. rückgesetzt. Der Zustand wird für das einzuwechselnde WZ gesetzt bzw. rückgesetzt und für das MT, in dem dieses WZ bestückt ist. Der zustand bleibt für die nicht am Wechsel beteiligten anderen Werkzeuge unverändert.
10	MT ist zu entladen	Mit dem Entladen aus dem Magazin verliert das MT und die darin bestückten Werkzeuge diesen Zustand. Wird ein WZ, das diesen Zustand gesetzt hat, aus dem MT entfernt, wird der Zustand im entfernten WZ zurückgesetzt.
11	MT ist zu beladen	Mit dem Beladen auf einen Magazinplatz verlieren das MT und die in ihm bestückten Werkzeuge diesen Zustand. Wird ein WZ, das diesen Zustand gesetzt hat, in ein MT bestückt, dann wird der Zustand des Werkzeugs: – nicht geändert, falls das MT den Zustand gesetzt hat – zurückgesetzt, falls das MT den Zustand nicht gesetzt hat
12	Stamm-WZ	Der Zustand wird vom Anwender vergeben. Es gibt keine Beeinflussung des WZ–Zustands auf das MT und umgekehrt.

Bit	Bedeutung	Kommentar
14	für 1:1-Tausch markiert	Für im MT bestückte Werkzeuge wird dieser Zustand nicht benötigt. Er wird automatisch im Rahmen der Leerplatzsuche nur für das MT gesetzt und nach Ende des Kommandos wieder rückgesetzt.
16	MT wird gesperrt, falls ein WZ im MT gesperrt wird	Siehe die Beispiele 1a, 1b, 2a, 2b am Ende dieser Tabelle. Achtung Falls zum Zeitpunkt des Änderns dieses Zustands Werkzeuge im MT bestückt sind, dann gilt: Die Änderung dieses Zustands bewirkt ein Prüfen des Zustands der bestückten Werkzeuge und ein entsprechendes Setzen/Rücksetzen des MT-Zustands "gesperrt". Achtung Das Bestücken/Entfernen eines Werkzeugs in das MT / vom MT bewirkt nach Abschluss des Vorgangs ein Prüfen des Zustands der im MT bestückten WZ-Zustände und ein entsprechendes Setzen/Rücksetzen des MT-Zustands "gesperrt". Siehe Beispiele 4a, 4b.

Beispiel 1a:

Bit 2 = 0 und **Bit 16 = 1**

Es sind drei nicht gesperrte und frei gegebene Werkzeuge im MT enthalten. Das MT selbst ist ebenfalls nicht gesperrt und frei gegeben. Bit 16 = 1 ist gesetzt.

Wenn der Zustand "gesperrt" eines beliebigen dieser Werkzeuge im MT gesetzt wird (manuell oder automatisch), wird zusätzlich der Zustand "gesperrt" im MT gesetzt, obwohl noch zwei Werkzeuge den Zustand "gesperrt" nicht gesetzt haben (d. h. noch einsatzfähig sind).

Beispiel 1b:

Bit 2 = 1 und **Bit 16 = 1**

Es sind drei gesperrte und freigegebene Werkzeuge im MT enthalten. Das MT selbst ist ebenfalls gesperrt und frei gegeben. Bit 16 = 1 ist gesetzt.

Wenn der Zustand "gesperrt" von zweien der drei Werkzeuge im MT gelöscht wird, dann bleibt der Zustand "gesperrt" des Multitools selbst unverändert. Wird dann für das letzte WZ mit dem Zustand "gesperrt' der Zustand "gesperrt" gelöscht (manuell oder automatisch), wird zusätzlich der Zustand "gesperrt" im MT gelöscht.

Beispiel 2a:

Bit 2 = 0 und **Bit 16 = 0**

Es sind drei nicht gesperrte und einsatzfähige Werkzeuge im MT enthalten. Das MT selbst sei ebenfalls nicht gesperrt und frei gegeben. Bit 16 = 0 ist gesetzt.

Wenn der Zustand "gesperrt" von zweien der drei Werkzeuge im MT gesetzt wird (manuell oder automatisch), dann bleibt der Zustandswert des Bits 2 im MT unverändert. Somit kann das letzte "ungesperrte" WZ des Multitools weiterhin eingewechselt werden. Erst wenn auch dieses WZ als Letztes den Zustand "gesperrt" einnimmt, wird auch im MT der Zustand "gesperrt" gesetzt.

Beispiel 2b:

Bit 2 = 1 und **Bit 16 = 0**

Es sind drei gesperrte und einsatzfähige Werkzeuge im MT enthalten. Das MT selbst sei ebenfalls gesperrt und frei gegeben. Bit 16 = 0 ist gesetzt.

Wenn der Zustand "gesperrt" eines Werkzeugs im MT gelöscht wird (manuell oder automatisch), dann wird zusätzlich der Zustand "gesperrt" im MT gelöscht, unabhängig davon wie der Zustand der anderen Werkzeuge im MT ist.

Bit 16 = 0 ist z. B. vorteilhaft, wenn das MT nur gleichartige Werkzeuge enthält. Alle Werkzeuge können dann bis zum Erreichen ihrer jeweiligen Überwachungsgrenze eingesetzt werden.

Bit 16 = 1 ist z. B. vorteilhaft, wenn ein MT eingewechselt werden soll und dessen nicht gleichartige Werkzeuge verschiedene Bearbeitungen vornehmen sollen. Wenn eines dieser Werkzeuge nun wegen der Überwachung gesperrt wird, dann muss ein anderes MT eingewechselt werden, das wieder einen kompletten einsatzfähigen WZ–Satz für diese Bearbeitung enthält.

In den folgenden beiden Bildern wird noch einmal eine grafische Darstellung der "gesperrt" Zustandskopplung dargestellt:

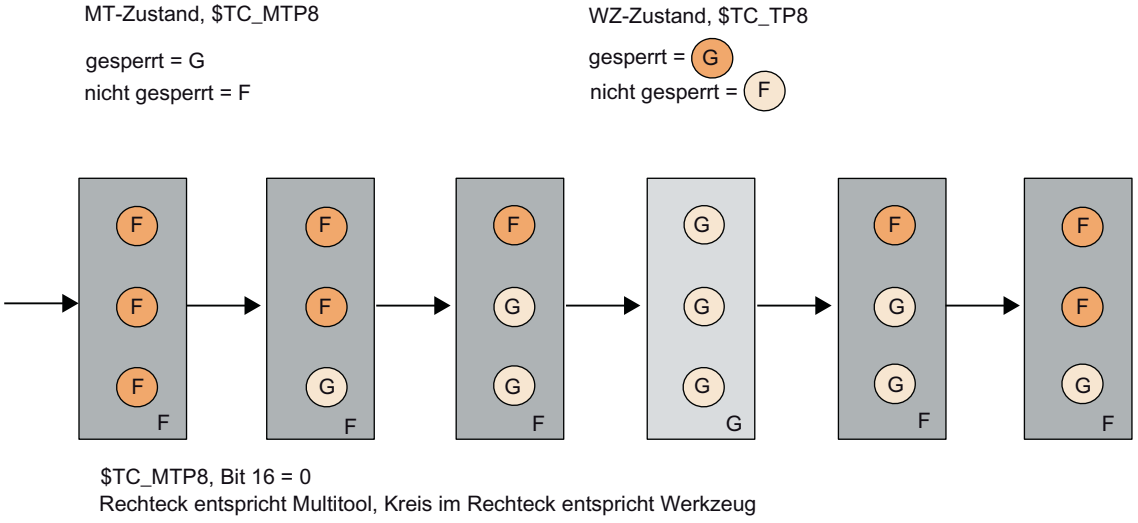


Bild 11-9 Automatische Zustandsänderung des Multitools in Abhängigkeit des Zustands der bestückten Werkzeuge und des Bit 16 von $TC_MTP8

Beispiel 3a:

Bit 16 = 0:

Wird im oberen Bild Bit 16 von 0 nach 1 zu dem Zeitpunkt geändert, zu dem der Zustand vorliegt wie er im von links fünften MT-Zustandsbild zu sehen ist (MT mit F), dann ändert sich der Zustand des MTs von F nach G, da nach der Bitwertänderung mindestens ein WZ mit dem Zustand G bestückt ist und daher das MT auch den Zustand G einnimmt. Mit Änderung des Bitwerts von 0 nach eins geht dieses Bild in folgendes über.

Beispiel 4a:

Bit 16 = 0:

Wird im oberen Bild im von links dritten MT-Zustandsbild (MT mit F) das Werkzeug mit F vom Multitool entfernt, ändert sich der Zustand des MTs von F nach G, da nach der WZ-Entfernung nur noch zwei Werkzeuge mit Zustand G bestückt sind und daher das MT auch den Zustand G einnimmt.

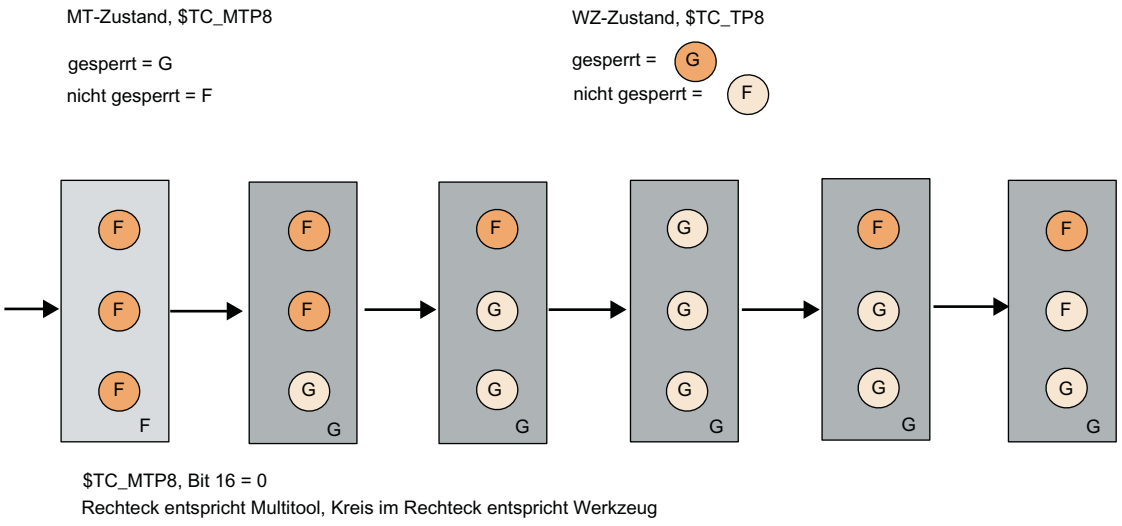


Bild 11-10 Automatische Zustandsänderung des Multitools in Abhängigkeit des Zustands der bestückten Werkzeuge und des Bits 16 von $TC_MTP8

Beispiel 4b:

Bit 16 = 1:

Wird im oberen Bild im von links zweiten MT-Zustandsbild (MT mit G) das WZ mit G vom Multitool entfernt, ändert sich der Zustand des MTs von G nach F, da nach der WZ-Entfernung nur noch zwei Werkzeuge mit Zustand F bestückt sind und daher das MT auch den Zustand F einnimmt.

$TC_MTP_POS - MT-Position

Die Position des Multitools ist die MT–Platznummer des Platzes, der in Bearbeitungsposition ist, und typisch das bearbeitende (aktive) WZ enthält; siehe nächstes Bild.

Mit der Ende-Quittierung (PLC-Statuswert=1) eines WZ-Wechselbefehls für ein WZ im MT wird die MT-Position auf den Wert der MT-Platznummer des MT–Platzes gesetzt, der das eingewechselte WZ enthält. Die anderen PLC-Statuswerte (z. B. Status_105) ändern den MT–Positionswert nicht.

Mit der Ende-Quittierung (PLC-Statuswert=1) eines MT-Positionierbefehls (POSMT, _N_POSMT) wird die MT-Position auf den Wert der programmierten MT-Platznummer gesetzt. Die anderen PLC-Statuswerte sind für den MT-Positionierbefehl nicht erlaubt und werden mit Alarm 6405 "falsche Quittierungsdaten" abgewiesen.

Mit einem asynchronen Transfer kann die PLC das Multitool positionieren.

Wird ein an PLC ausgegebener Befehl nicht mit "Ende" quittiert (z. B. Abbruch, Status_3), so erfolgt auch kein Aktualisieren der MT-Position in NC (selbst wenn die Positionierung mechanisch bereits erfolgt ist).

Die MT–Position eines Multitools, das nicht auf einem WZ-Halter ist, hat keine weitere Bedeutung. Ausnahme ist die Programmierung T="Magazin–Platznummer", die typisch für Revolver verwendet wird.

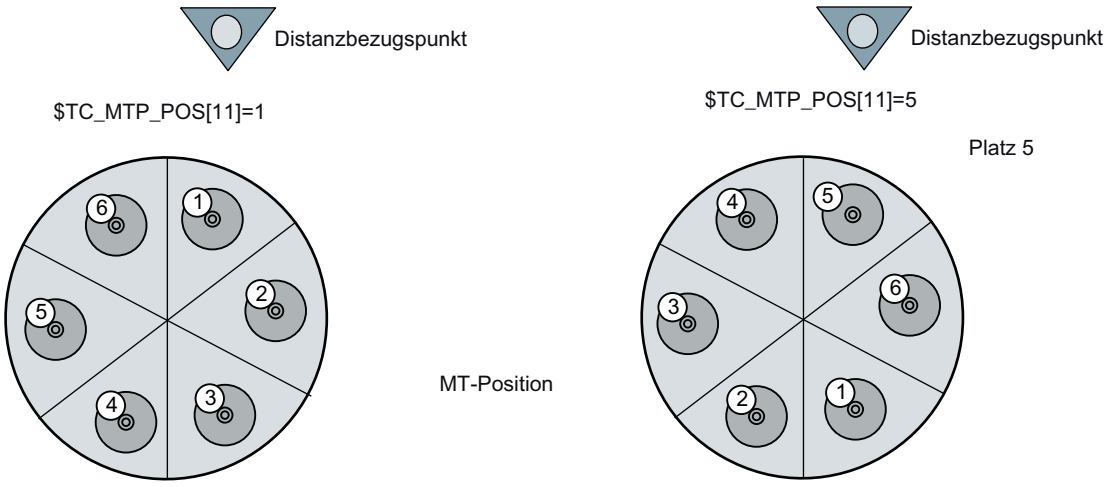


Bild 11-11 Der Platz an der Bearbeitungsposition bestimmt die MT-Position

Beispiel 1:

Ein MT wird mit einem WZ-Bewegen Befehl (z. B. MVTOOL) auf den Werkzeughalter bewegt. Die Parameter werden gemäß der aktuellen MT-Position gesetzt. Mit der Ende-Quittierung des MT-Bewegebefehls kann PLC dem NC mitteilen, ob und wie die Position des Multitools zu aktualisieren ist.

- Der FC6 wird parametriert mit MultitoolLoc=0. NC nimmt keine Positionssynchronisierung vor, die mechanische Position kann von der datentechnischen abweichen.
- Der FC6 wird parametriert mit MultitoolLoc=1 (gültige MT-Platznummer). Das bedeutet, dass NC die MT-Position auf die quittierte Position synchronisiert.

Erfolgt anschließend eine WZ-Wechselprogrammierung für ein WZ dieses MTs auf dem WZ-Halter, so wird ein WZ-Wechselkommando an PLC ausgegeben mit dem die MT-Positionierungsdaten an PLC gesandt werden. Die MT-Position wird mit der Ende-Quittierung des WZ–Wechselbefehls auf den Wert aktualisiert, der zum einzuwechselnden WZ gehört. PLC muss die MT-Positionierung zwingend ausgeführt haben, bevor die Ende-Quittierung an NC gegeben wird. PLC darf in den Quittierungsdaten die von NC vorgegebene MT-Position bei WZ-Anwahl, WZ-Wechsel Kommandos nicht abändern. Der Versuch wird mit Alarm 6405 "Befehl hat ungültigen PLC-Quittungsparameter "abgelehnt.

Beispiel 2:

Ein MT wird mit einem WZ–Bewegen Befehl (z. B. MVTOOL) auf den WZ–Halter bewegt. Wie in Beispiel 1 liegt es an der PLC-Applikation wie mit der MT-Positionierung/-Synchronisierung verfahren wird.

Wird anschließend RESET ausgeführt, so wird – falls RESET mit "aktiviere WZ auf dem Master-WZ-Halter" konfiguriert ist – ein WZ–Wechselkommando an PLC ausgegeben mit dem die Daten des zu aktivierenden Werkzeugs und die MT–Positionierungsdaten an PLC gesandt werden. Bei RESET wird in NC das WZ als zu aktivierend bestimmt, das sich auf dem MT-Platz befindet, den die an PLC ausgegebene MT-Position anzeigt. D. h. in diesem Beispiel hängt es von der PLC-Logik beim WZ-Bewegen auf den Master-WZ-Halter ab, welches WZ bei RESET aktiviert wird. Soll ein anderes WZ aktiviert werden, so muss die Applikation dieses bestimmen, indem z. B. vorher mit dem NC-Programmbefehl POSMT (oder analogem PI _N_POSMT durch HMI oder PLC) das MT entsprechend positioniert wird. Es ist auch möglich, die MT-Position in NC vor RESET direkt auf den gewünschten Wert zu setzen (durch Programmierung des Parameters $TC_MTP8 bzw.

durch Schreiben der Position über BTSS (durch HMI oder PLC), oder die PLC positioniert das Multitool mit einem asynchronen Transfer).

$TC_MTP_KD - Art der Abstandskodierung für die MT-Plätze

Die Platzabstände (**K**indOf**D**istance) innerhalb des Multitools können auf verschiedene Art definiert werden:

$TC_MTP_KD[y] = 1	Die Abstandskodierung erfolgt über die MT-Platznummer. Die Platznummer wird beim Erzeugen der MT-Plätze automatisch vergeben. $TC_MTPPL, $TC_MTPPA sind nicht schreibbar, außer mit dem Wert 0.0.
$TC_MTP_KD[y] = 2	Die Abstandskodierung erfolgt über Längenangaben. $TC_MTPPL ist schreibbar, $TC_MTPPA ist nicht schreibbar, außer $TC_MTPPA[...]=0.0.
$TC_MTP_KD[y] = 3	Die Abstandskodierung erfolgt über Winkelangaben. $TC_MTPPA ist schreibbar. $TC_MTPPL ist nicht schreibbar, außer $TC_MTPPL[...]=0.0.

Die Art der Abstandskodierung kann nur verändert werden, solange noch keine MT-Plätze erzeugt worden sind. Andernfalls wird Alarm 6438 "Inkonsistente Datenänderung ist nicht erlaubt" erzeugt.

Beispiel:

Wenn zur Abstandskodierung die Platznummer festgelegt worden ist ($TC_MTP_KD[y] = 1), dann führt das Schreiben von $TC_MTPPL, $TC_MTPPA mit anderen Werten als 0.0 zum Alarm 6464 "Befehl kann für die aktuelle Multitool-Abstandskodierung nicht programmiert werden".

Beispiel:

Das Multitool wird definiert. Es ist die Anzahl MT-Plätze zwar definiert, aber noch nicht erzeugt worden.

```
$TC_MTP_KD[200] = 1     ;späteres Schreiben von $TC_MTPPL bzw. $TC_MTPPA führt
                        ;zum Alarm "Befehl kann für die definierte ;Multitoolabstandsko-
                        dierung nicht programmiert werden"
$TC_MTP_KD[300] = 2     ;späteres Schreiben von $TC_MTPPL ist erlaubt.
                        ;späteres Schreiben von $TC_MTPPA führt zum Alarm
                        ;"Befehl kann für die definierte Multitoolabstandskodierung
                        ;nicht programmiert werden"
```

Nun werden die MT-Plätze erzeugt z. B. durch Beschreiben des Systemparameters:

```
$TC_MTPPL[300, 1] = 30.0      ; alle MT-Plätze werden mit dem Befehl implizit erzeugt
```

Danach werde versucht, die Distanzart zu ändern (die zuvor mit $TC_MTP_KD[300] = 2 definiert wurde):

```
$TC_MTPPL[300, 1] = 3   ;Nach dem die MT-Plätze erzeugt worden sind, darf die Art
                        ;der Abstandskodierung nicht mehr geändert werden:
                        ;Alarm "Inkonsistente Datenänderung nicht erlaubt"
```

11.7.2 $TC_MTPP - Multitool Platzdaten

Übersicht

Das voreingestellte Zugriffsrecht der folgenden Parameter ist "Schlüsselschalterstellung 0" (=jeder darf zugreifen), sowohl für Zugriffe aus dem NC-Programm als auch für Zugriffe über BTSS. Mit dem Befehl REDEF können Zugriffsrechte eingeschränkt werden.

$TC_MTPPx[y, z]

x:= 2, 4, 6, 7, L, A (siehe Tabelle)

y: = Multitool–Nummer MT=1...32000

z: = n-ter Platz im Multitool; 1,...$MN_MAX_TOOLS_PER_MULTITOOL

Die Parameternummern x sind an die Werte der Magazinplatzparameter angelehnt.

Die Nummern x der Parameter sind an die Nummern der Magazinplatz–Parameter $TC_MPP1,... angelehnt.

NC-Bezeichner	Beschreibung	Format	Physik. Einheit	Vorbelegung Grenzwerte
$TC_MTPP2	Multitoolplatztyp (entspricht der Bedeutung des Magazinplatztyps)	INT	keine	0 (=jedes WZ passt auf den Platz) 0 - 32000
$TC_MTPP4	MT-Platzzustand	INT	keine	0 (=jedes WZ passt auf den Platz) 0 - 3
$TC_MTPP6	T-Nummer des enthaltenen Werkzeugs	INT	keine	0 0 - 32000
$TC_MTPP7	Reserviert für Magazinplatzdaten			
$TC_MTPPL	Abstand vom Referenzplatz; Einheit "Länge" (L steht für Length/Länge)	REAL	Länge [mm/inch]	0 0 - 1000
$TC_MTPPA	Abstand vom Referenzplatz; Einheit "Winkel" (A steht für Angle/Winkel)	REAL	Winkel [Grad]	0 0 - 360

$TC_MTPP2 - MT-Platztyp

Der Platztyp eines zu bestückenden Werkzeugs muss mit dem Platztyp des MT-Platzes übereinstimmen. Die Prüfung wird bei der Bestückung des Multitools durch Schreiben des Parameters $TC_MTPP6 vorgenommen.

Der Platztyp darf nur geändert werden, wenn der neue Wert mit dem Wert des Platztyps eines eventuell enthaltenen Werkzeugs verträglich ist.

Der Vorbelegungswert ist so gewählt (jedes WZ passt auf den MT-Platz), dass, falls die Funktion "Platztypprüfung" in der konkreten Applikation nicht benötigt wird für den WZ-Bestückungsvorgang, der Parameter nicht weiter betrachtet werden muss; d. h. per Vorbelegung schon passend ist.

$TC_MTPP4 - MT-Platzzustand

Es sind folgende Platzzustände definiert (Bitnummern und Definition an den Zustand des Magazinplatzes angelehnt; $TC_MPP4):

0 gesperrt

1 frei zur Aufnahme eines Werkzeugs

$TC_MTPP6 - T-Nummer des bestückten Werkzeugs

Die Plätze des Multitools können mit Werkzeugen bestückt werden. Das Bestücken ist nur mithilfe des Parameters $TC_MTPP6 definiert. Insbesondere ist kein Bestückvorgang unter Beteiligung von PLC definiert (Der BTSS PI-Dienst _N_TMMVTL und der NC-Sprachbefehl MVTOOL sind ausschließlich für Be-/Entladevorgänge von Werkzeugen und Multitools in Magazine bzw. aus Magazinen vorbehalten. Mit $TC_MTPP6[mtNr, plNr] = T-Nummer findet die Bestückung statt. Mit $TC_MTPP6[mtNr, plNr] = 0 wird ein WZ aus dem Multitool entfernt.

Das Multitool kann Werkzeuge mit gleichen oder unterschiedlichen Namen enthalten. Jedes WZ passt mit dem Vorbelegungswert von $TC_MTPP2 auf jeden Platz eines Multitools. Allgemein wird angenommen, dass ein Multitool ein mechanisches Gebilde ist, dessen korrekte Bestückung allein vom Anwender zu verantworten ist.

Falls für der MT-Zustand, $TC_MTP8, Bit 16 = 1 gesetzt ist, ist Folgendes zu beachten:

Beim Bestücken eines Werkzeugs in ein MT bzw. beim Entfernen eines WZs aus einem MT kann sich der MT Zustand "gesperrt"/"nicht gesperrt" ändern. Dabei gilt:

- Nach dem Bestücken bestimmen die Zustände der Werkzeuge im MT, ob und wie sich der Zustand des Multitools automatisch ändert.

Beim Entfernen eines WZs aus einem MT gilt:

- Nach dem Entfernen des Werkzeugs bestimmen die Zustände der verbleibenden Werkzeuge im MT, ob und wie sich der Zustand des MTs automatisch ändert.

Formal können Werkzeuge zu jedem Zeitpunkt in ein MT bestückt werden, oder aus einem MT entfernt werden. Es liegt in der Verantwortung des Anwenders, dies im Einklang mit der Applikation zu tun. Beispielsweise muss der Anwender wissen, was es bedeutet, ein auf einen WZ-Halter eingewechseltes MT vom WZ-Halter zu entfernen, oder ein aktives WZ aus einem MT zu entfernen.

Ein WZ kann entweder einem Magazin, oder einem MT zugeordnet werden; nicht aber gleichzeitig einem Magazin und einem MT. Ein derartiger Versuch wird mit Alarm 17260 "Illegale Magazinplatzdefinition" abgelehnt.

Beim Bestücken eines Werkzeugs auf ein MT wird der Magazinplatztyp von WZ und MT-Platz auf Verträglichkeit geprüft, falls der Platztyp des MT-Platzes ungleich dem Wert "jedes WZ passt auf den Platz" ist (das ist zugleich der Vorbelegungswert). Ein WZ kann nur auf einen MT-Platz bestückt werden, wenn der Zustand des MT-Platzes den Wert "frei" hat.

Hinweis

Multitools haben keine Funktion Nebenplatzbetrachtung für übergroße Werkzeuge. Das bedeutet, dass beim Bestücken eines Werkzeugs auf einen MT-Platz die WZ-Größenangaben im WZ nicht ausgewertet werden.

$TC_MTPPL, $TC_MTPPA - Platzabstand

Der Abstand der Werkzeuge innerhalb des Multitools wird alternativ mit einem dieser Parameter definiert. Welcher Parameter für das konkrete Multitool zu definieren ist, wird über den MT-Parameter $TC_MTP_KD bestimmt. Die Änderungsregeln für das Datum sind die, die auch für die Multitoolgrößen definiert sind. Für die Abstandskodierung "MT-Platznummer" wird kein Systemparameter gebraucht. Beim Anlegen der Plätze für ein MT werden die Plätze mit den Nummern 1, 2, .. automatisch versehen.

11.7.3 Multitoolbezogene Anwenderdaten

$TC_MTPCx[t]

Anwenderbezogene Multitool-Werkzeugdaten

Pro Multitool können zusätzlich 64 multitoolspezifische Parameter angelegt werden. Einstellen mit MD18192 MM_NUM_CC_MULTITOOL_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

x: = Parameter 1...64

t: = T-Nummer 1...32000

BTSS-Baustein MTUD

Berechnung der Zeile: T-Nummer

Berechnung der Spalte: Parameternummer

Werkzeugbezogene Daten OEM-Anwender				
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$ TC_MTPC1	Double		data	REAL
...	Double		data	REAL
$TC_MTPC64	Double		data	REAL

Hinweis

Die Daten werden in der Werkzeugverwaltung angezeigt. Hier können z. B. zusätzliche Werkzeugstatus abgelegt werden.

11.7.4 Multitoolplatzbezogene Anwenderdaten

$TC_MTPPCx[t]

Anwenderbezogene Multitool-Werkzeugdaten

Pro Multitool können zusätzlich 64 multitoolplatzspezifische Parameter angelegt werden. Einstellen mit MD18194 $MN_MM_NUM_CC_MTLOC_PARAM und Freigabe mit MD18080 MM_TOOL_MANAGEMENT_MASK (Bit 2 setzen)

x: = Parameter 1...64

t: = T-Nummer 1...32000

BTSS-Baustein MTUP

Berechnung der Zeile: T-Nummer

Berechnung der Spalte: Parameternummer

Werkzeugbezogene Daten OEM-Anwender				
NC-Bezeichner	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$ TC_MTPPC1	Double		data	REAL
...	Double		data	REAL
$TC_MTPPC64	Double		data	REAL

Hinweis

Die Daten werden in der Werkzeugverwaltung angezeigt. Hier können z. B. zusätzliche Werkzeugstatus abgelegt werden.

11.8 Adapterdaten $TC_ADPT

$TC_ADPTx[n]

Falls das Maschinendatum $MN_MM_NUM_TOOL_ADAPTER einen Wert = -1 oder > 0 hat, werden die Adapterdaten über folgende Variablen definiert, gelöscht, gelesen, geschrieben

x: = Parameter 1...3, T

n: = Nummer des Adapters

Hinweis

BTSS-Baustein AD

Berechnung der Zeile: Länge 1, 2, 3 = Zeile 1, 2, 3, Transformation = Zeile 4

Berechnung der Spalte: Adapternummer

Adapterdaten				
Name	**Typ**	**Bezeichnung**	**BTSS-VAR**	**Typ**
$TC_ADPT1	Double	Adaptergeometrie: Länge 1	adaptData	REAL
$TC_ADPT2	Double	Adaptergeometrie: Länge 2	adaptData	REAL
$TC_ADPT3	Double	Adaptergeometrie: Länge 3	adaptData	REAL
$TC_ADPTT[n]	Double	Adaptertransformations-Nummer	adaptData	REAL

Die Adaptergeometriewerte wirken auf die Geometriewerte der Schneide analog wie die Systemvariablen $TC_DP 21, $TC_DP 22, $TC_DP 23. Die Parameter sind nur bei aktiver Werkzeugverwaltung verfügbar.

Bei der Adaptertransformation sind die Transformationsnummern 1 bis 8 möglich. Der Parameter ist nur bei aktiver Werkzeugverwaltung verfügbar.

$TC_MPP7[m,p]: Nummer des dem Magazinplatz zugeordneten Adapters

Wert=0 : kein Adapter dem Platz zugeordnet

Wert>0 : Nummer des zugeordneten Magazins

11.9 Freie Anwendervariablen

Freie Parameter

Dem Anwender stehen mit diesen programmierbaren Variablen drei freie Parameter zur Verfügung. Diese Systemvariablen werden über die Anwender-Nahtstelle mit dem T-Anwahlsignal und dem Wechselbefehl zur PLC übertragen. Hiermit steht dem Anwender eine Möglichkeit offen, zusätzliche Information bezüglich der Werkzeugverwaltung zur PLC zu bringen. Die Parameter sind vom NC-Programm aus lese- und schreibbar. Sie sind nicht gepuffert und werden bei Reset oder Programmende auf 0 gesetzt.

$P_VDITCP[x]

x: = Parameter 0, 1, 2

NC-Bezeichner	Beschreibung	Format
$TC_VDITCP[0]	WZV-VDI freier Parameter 0	int
$TC_VDITCP[1]	WZV-VDI freier Parameter 1	int
$TC_VDITCP[2]	WZV-VDI freier Parameter 2	int

Programmierung

Die Parameter können beliebig im NC-Programm programmiert werden. Die Ausgabe an PLC erfolgt aber immer in Verbindung mit dem in Folgenden programmierten Werkzeug-Vorbereitungs- oder Wechselkommandos.

Beispiel:

```
T= "WZ1"
$P_VDITCP[0] = 1
M06
$P_VDITCP[0] = 2
T= "WZ2"
```

Mit der Kommandoausgabe von T="WZ2" an PLC wird genau der Wert = 2 an PLC mit ausgegeben und nicht auch der Wert 1 bei Ausgabe des M06-Kommandos an PLC.

Die Ausgabe des programmierten Wertes erfolgt auch mit der Programmierung von M6, d. h., die Ausgabe kann nun auch mit der Kommandonummer 3 erfolgen, sofern $MC_CHANGE_MODE=1 eingestellt ist.

11.10 NC-Sprachbefehle

11.10.1 CHKDNO - Prüfung der Eindeutigkeit der D-Nummer

Bei aktiver Werkzeugverwaltung prüft die Funktion `CHKDNO` die Eindeutigkeit der D-Nummer. Unter Eindeutigkeit der D-Nummer wird hier (keine Ersatzwerkzeuge) verstanden, dass die D-Nummern aller in der TO-Einheit definierten Werkzeuge genau einmal auftreten dürfen. Die D-Nummern in der TO-Einheit sind eindeutig und absolut.

Syntax

```
State = CHKDNO(T1, T2,D)
```

Bedeutung

`CHKDNO`	Die Funktion liefert als Rückgabewert den Wert von `state` zurück	
	Datentyp:	Bool
`T1`	T-Nummer von Werkzeug 1 (optional)	
	Datentyp:	INT
`T2`	T-Nummer von Werkzeug 2 (optional)	
	Datentyp:	INT
`D`	D-Nummer (optional); Ohne Parameter D werden alle D-Nummern der genannten Werkzeuge geprüft.	
	Datentyp:	INT
Rückgabewert		

state	state zeigt an, ob eine D-Nummernkollision auftritt	
	Daten-typ:	BOOL
	FALSE = D-Nummernkollision liegt vor	
	TRUE = D-Nummern sind eindeutig	

D-Nummern von Ersatzwerkzeugen

Mit aktiver Werkzeugverwaltung kann man Ersatzwerkzeuge definieren und benutzen. Das Bearbeitungs-Teileprogramm gibt in der Regel keinen Hinweis darauf, ob Ersatzwerkzeuge vorhanden sind. Das Bearbeitungsprogramm spricht Werkzeuge im Allgemeinen mit T="Bezeichner" an. (Die Programmierung T="Platznummer" wird intern wieder auf T="Bezeichner" zurückgeführt). Das Programm enthält ansonsten nur noch die eigentliche Programmierung der Korrektur (die D-Nummer). Aus diesem Grund müssen die D-Nummern von Werkzeug und Ersatzwerkzeugen dieselben sein.

Beispiel

Aktives Werkzeug und Ersatzwerkzeuge für T="Bohrer_5mm"

- T-Nr. = 10 mit den D-Nummern 1, 2, 3 (aktiv)
- T-Nr. = 11 mit den D-Nummern 1, 2, 3 (Ersatz)
- T-Nr. = 12 mit den D-Nummern 1, 2, 3 (Ersatz)

Aktives Werkzeug und Ersatzwerkzeuge für T="Bohrer_3mm"

- T-Nr. = 20 mit den D-Nummern 1, 2, 3 (aktiv)
- T-Nr. = 21 mit den D-Nummern 1, 2, 3 (Ersatz)
- T-Nr. = 22 mit den D-Nummern 1, 2, 3 (Ersatz)

CHKDNO - ohne Angabe von Parametern - ermittelt im oben genannten Beispiel eine Kollision der D-Nummern 1, 2 und 3 von "Bohrer_5mm" mit den D-Nummern 1, 2 und 3 von "Bohrer_3mm", aber nicht zwischen den D-Nummern von aktiven und Ersatzwerkzeugen.

Die auftretenden Kollisionen werden einzeln als Alarme angezeigt, wie z. B.:

- "Kanal 1 D-Nummer 1 bei Werkzeug-T-Nr. 10 und 20 definiert"
- "Kanal 1 D-Nummer 1 bei Werkzeug-T-Nr. 10 und 21 definiert"

Bei ungültiger Parametrierung (genannte T- bzw. D-Nummer ist im Kanal nicht definiert), wird ebenfalls state = FALSE zurückgegeben.

Falls gilt: MAX_CUTTING_EDGE_NO ≤ MAX_CUTTING_EDGE_PER_TOOL liefert CHKDNO unabhängig von der Parametrierung immer den Zustand TRUE.

11.10.2 CHKDM - Prüfung der Eindeutigkeit innerhalb eines Magazins

Bei aktiver Werkzeugverwaltung prüft der Befehl CHKDM bestehende Daten in NC auf D-Nummerneindeutigkeit innerhalb eines oder mehrerer Magazine. Die Funktionalität entspricht der von CHKDNO. Die Parameter sind optional.

Syntax

```
state = CHKDM(Magnr, Dnr, WZ-Halternr)
```

Bedeutung

CHKDM	Überprüft die Eindeutigkeit einer D-Nummer innerhalb eines Magazins	
	Datentyp:	Bool
Magnr	Magazinnummer des zu prüfenden Magazins	
	Datentyp:	INT
	Weglassen des Parameters bzw. Programmierung mit dem Wert=0 bedeutet, dass die Werkzeuge aller der in Wz-Haltenr genannten Spindelnr. bzw. WZ-Halternr. verbundenen Magazine geprüft werden.	
Dnr	D-Nummer, gegen die geprüft wird	
	Datentyp:	INT
	Weglassen des Parameters bzw. Programmierung mit dem Wert=0 bedeutet, dass alle D-Nummern des genannten Magazins auf Eindeutigkeit geprüft werden.	
Wz-Halternr	Gibt an, bezüglich welcher Spindelnummer bzw. Werkzeughalternummer die Magazine geprüft werden	
	Einheit	INT
	Weglassen des Parameters bedeutet, dass sich die Magazine für die Prüfung aus der Distanztabelle des Spindelplatzes für die Masterspindel oder den Mastertoolholder ergeben.	
Rückgabewert		
state	Ergebnis der Prüfung	
	Datentyp	Bool
	TRUE	Geprüfte D-Nummern sind eindeutig
	FALSE	Prüfung nicht in Ordnung

11.10.3 GETACTTD - Ermittlung der T-Nummer zu einer eindeutigen D-Nummer

Bei aktiver Werkzeugverwaltung (z. B. bei Messzyklenprogrammen) liefert der Befehl GETACTTD, ausgehend von einer D-Nummer, die zugehörige T-Nummer des in der WZ-Gruppe aktiven Werkzeugs.

Syntax

```
status = GETACTTD (Tnr, Dnr)
```

Bedeutung

GETACTTD	liefert die T-Nummer des in der Werkzeug-Gruppe aktiven Werkzeugs	
	Datentyp:	INT

Dnr	D-Nummer, für die die T-Nummer gesucht werden soll. Es findet keine Prüfung der D-Nummer auf Eindeutigkeit statt.	
	Datentyp:	INT
	Wenn in verschiedenen Werkzeuggruppen derselben TO-Einheit gleiche D-Nummern definiert sind, wird die T-Nummer der ersten gefundenen Werkzeuggruppe ermittelt, deren Werkzeuge die genannte Nummer enthalten.	
Tnr	Gefundene T-Nummer	
	Datentyp:	INT
Rückgabewert		
status	Ergebnis der Suche	
	Datentyp:	INT
	0	T-Nummer gefunden, Tnr erhält den Wert
	-1	Zur angegebenen D-Nummer existiert keine T-Nummer, Tnr erhält den Wert 0.
	-2	D-Nummer ist nicht eindeutig, Tnr erhält den Wert der ersten ermittelten D-Nummer.
	-3	In der Werkzeuggruppe gibt es kein Werkzeug mit dem gewünschten Status und der angegebenen D-Nummer, Tnr erhält den Wert 0.
	-4	In der Werkzeuggruppe gibt es mehrere Werkzeuge mit dem gewünschten Status und der gesuchten D-Nummer; Tnr enthält den Wert des ersten gefundenen Werkzeuges mit der gewünschten D-Nummer.
	-5	Funktion konnte aus anderen Gründen nicht ausgeführt werden.

11.10.4 GETDNO - D-Nummer auslesen

Die Funktion GETDNO liest die Korrekturnummer d eines Werkzeugs mit der T-Nummer t und der Schneide ce aus.

Syntax

```
d = GETDNO(t, ce)
```

Bedeutung

GETDNO	Liest die Korrekturnummer d aus	
	Datentyp:	INT
t	T-Nummer des Werkzeugs	
	Datentyp:	INT
ce	Schneide ce des Werkzeugs	
	Datentyp:	INT
Rückgabewert		

d	Rückgabewert Korrekturnummer d	
	Datentyp:	INT
	Sind t oder ce Parameter, zu denen kein Datensatz existiert, wird d=0 zurückgegeben. Die Syntaxregeln verletzenden Parameter erzeugen einen Alarm.	

Abhängigkeiten

- Der Befehl ist nur verfügbar, wenn $MN_MAX_CUTTING_EDGE_NO > $MN_MAX_CUTTING_EDGE_PER_TOOL.
- $MN_MAX_CUTTING_EDGE_NO ≤ $MN_MAX_CUTTING_EDGE_PER_TOOL liefert GETDNO d=ce als D-Nummer.

11.10.5 SETDNO - D-Nummer setzen bzw. ändern

Mit der Funktion SETDNO kann die Korrekturnummer d der Schneide ce des Werkzeugs t gesetzt bzw. geändert werden.

Syntax

```
state = SETDNO(t, ce, d)
```

Bedeutung

SETDNO	Funktion zum Bearbeiten der Korrekurnummer d der Schneide ce des Werkzeugs t	
	Datentyp:	INT
t	T-Nummer des Werkzeugs	
	Datentyp:	INT
ce	Schneide ce des Werkzeugs	
	Datentyp:	INT
d	Korrekturnummer d	
	Datentyp:	INT
	Sind t oder ce Parameter, zu denen kein Datensatz existiert, wird state = FALSE zurückgegeben. Die Syntaxregeln verletzenden Parameter erzeugen einen Alarm.	
Rückgabewert		
state	Status der Funktion	
	Datentyp:	INT
	FALSE (0)	Bearbeiten war nicht erfolgreich
	TRUE (1)	Bearbeiten war erfolgreich

Abhängigkeiten

t, ce, d müssen > 0 angegeben werden, d=0 kann nicht gesetzt werden.

Siehe auch

DZERO - D-Nummern ungültig setzen (Seite 167)

11.10.6 DZERO - D-Nummern ungültig setzen

Die Funktion `DZERO` kennzeichnet alle D-Nummer einer TO-Einheit als ungültig. Der Befehl dient zur Unterstützung während des Umrüstens. So gekennzeichnete Korrekturdatensätze werden nicht mehr vom Sprachbefehl `CHKDNO` geprüft. Um sie wieder zugänglich zu machen, müssen die D-Nummern wieder mit `SETDNO` gesetzt werden, siehe auch SETDNO - D-Nummer setzen bzw. ändern (Seite 166).

Syntax

```
DZERO
```

11.10.7 DELDL - Additive Korrekturen löschen

Mit der Funktion `DELDL` werden additive Korrekturen für die Schneide eines Werkzeugs gelöscht (Freigabe von Speicher). Dabei werden sowohl die festgelegten Verschleißwerte als auch die Einrichtewerte gelöscht.

Syntax

```
status = DELDL(t, d)
```

Bedeutung

`DELDL`	Löscht alle Summenkorrekturen der Schneide `d` des Werkzeugs `t`	
	Datentyp:	INT
`t`	Nummer des Werkzeugs (optional); wird `t` und `d` nicht angegeben, werden alle Summenkorrekturen aller Schneiden aller Werkzeuge der TO-Einheit gelöscht.	
	Datentyp:	INT
`d`	Nummer der Schneide (optional); wird `d` nicht angegeben, werden alle Summenkorrekturen aller Schneiden des Werkzeugs t gelöscht.	
	Datentyp:	INT
Rückgabewert		
`status`	Ergebnis des Löschens	
	Datentyp:	INT
	0	Das Löschen wurde erfolgreich durchgeführt
	-1	Das Löschen wurde nicht durchgeführt (wenn die Parametrierung genau eine Schneide bezeichnet) oder das Löschen erfolgte nicht vollständig (wenn die Parametrierung mehrere Schneiden bezeichnet)

11.10.8 NEWT - Neues Werkzeug anlegen

Mit der Funktion NEWT wird ein neues Werkzeug ohne Angabe einer T-Nummer angelegt.

Die Funktion liefert als Rückgabe die automatisch erzeugte T-Nummer, mit der das Werkzeug anschließend adressiert werden kann. Das neue Werkzeug erhält automatisch eine Schneide mit CE-Nummer. = 1 und D-Nummer = 1. Alle Korrekturen sind mit 0 vorbesetzt. Eine Änderung der CE-Nummer kann anschließend über die Funktion SETDNO erfolgen, siehe SETDNO - D-Nummer setzen bzw. ändern (Seite 166).

Schlägt das Anlegen des neuen Werkzeugs fehl, wird ein Alarm ausgegeben.

Syntax

```
tnr = NEWT ("WZ", Duplonr.)
```

Bedeutung

NEWT	Legt ein neues Werkzeug mit dem Namen "WZ" und einer Duplo-Nr. Duplonr. an und liefert als Rückgabewert die T-Nummer tnr des neuen Werkzeugs.	
	Datentyp:	INT
WZ	Bezeichnung des Werkzeugs	
	Datentyp:	STRING
Duplonr.	Duplonummer des Werkzeugs (optional); bei Nichtangabe wird Duplo-Nr. = alte Duplo-Nr. +1 angewendet.	
	Datentyp:	INT
Rückgabewert		
tnr	T-Nummer des neuen Werkzeugs	
	Datentyp:	INT

Beispiele

Beispiel 1: Werkeug neu anlegen mit der Funktion NEWT und den CE-/D-Nummern 2, 47

Programmcode	**Kommentar**
def int tnr	
tnr = NEWT("Stahl", 111)	;WZ mit Ident/Duplonr.="Stahl"/111, T-Nr.=tnr=1 im Beispiel und einer Schneide CE=1, D=1 wird erzeugt
	;die Schneide soll nach CE=2, D=47 geändert werden
$TC_DPCE[tnr, 1]=2	:umbenennen der CE-Nummer
SETDNO(tnr,2,47)	;umbenennen der D-Nummer
	;die übrigen Daten des Werkzeugs / der Schneide zuweisen

Beispiel 2: Werkzeug "Stahl"/111, T-Nr.=tnr=1 anlegen mit $TC... und den CE-Nummern= 2, 4 (es wird angenommen, dass T-Nr.=1 noch nicht existiert

Programmcode	Kommentar
`$TC_TP1[1] = 111`	`;Werkzeug mit T-Nr.=1 neu anlegen, Duplonr.=111`
`$TC_TP2[1] = "Stahl"`	`;Werkzeug-Ident="Stahl" zuweisen`
`$TC_TPCE[1,47] = 2`	`;Korrektur D=47 neu erzeugen, CE-Nr.=2 zuweisen`
	`;die übrigen Daten des Werkzeugs / der Schneide zuweisen`

Die Funktion dient zum Anlegen von Werkzeugen in einem Beladeprogramm (Beladezyklus).

11.10.9 NEWMT Neues Multitool anlegen

Die Funktion `NEWMT` ist mit WZMG verfügbar.

Mit der Funktion `NEWMT` wird ein neues Multitool mit dem angegebenen Namen `"name"` und der MT–Platzanzahl = `"Platzanzahl"` erzeugt.

Es wird ein implizites STOPRE nach Satzende veranlasst.

Meldungen:

- Ist die Funktion "Mehrere Werkzeuge auf Magazinplatz" nicht aktiviert, so wird Alarm 6436 "Befehl kann nicht programmiert werden. Funktion ist nicht aktiviert" erzeugt.
- Ist der Parameter `name` schon in Verwendung, so wird Alarm 14020 "unerlaubter Parameterwert" erzeugt.
- Ist der Parameter `Platzanzahl` außerhalb des erlaubten Bereichs, so wird Alarm 14020 "unerlaubter Parameterwert" erzeugt.
- Wenn die Funktion `NEWMT` mit Alarm abgebrochen wird, dann wird für `mtNr` der Wert "0" zurückgegeben.

Syntax

```
mtNr = NEWMT(name, Platzanzahl)
```

Bedeutung

`NEWMT`	Legt ein neues Multitool mit der Bezeichnung `name` und der Platzanzahl `Platzanzahl` an.	
	Datentyp:	INT
`name`	Bezeichnung des Multitools	
	Datentyp:	STRING
`Platzanzahl`	Anzahl der Multitool-Plätze	
	Datentyp:	INT
Rückgabewert		

`mtNr`	MT-Nummer des Multitools; Die MT-Nummer wird dabei automatisch erzeugt und als Ergebniswert zurückgegeben.	
	Datentyp:	INT

Beispiele

Es gibt bereits ein Werkzeug mit dem Namen "A" und der T-Nummer 1 und ein Magazin mit dem Namen "B" und der Nummer 2. Nun wird Folgendes programmiert:

Programmcode	**Kommentar**
`def int mtNr`	`;`
`mtNr = NEWMT("C", 2)`	`;`

Der Befehl wird erfolgreich ausgeführt, `mtNr` hat den Wert 3. Das Multitool wurde für 2 Plätze erzeugt. Die Plätze wurden aber noch nicht erzeugt. Ein Aufruf mit dem Namen "A" oder "B" hätte zum Alarm geführt, da die Namen schon für ein Werkzeug und ein Magazin vergeben sind.

Nach der Erzeugung mit `NEWMT` können die weiteren Parameter des Multitools definiert werden, z. B.:

Programmcode	**Kommentar**
`$TC_MTP7[ 3 ] = 12`	`; Multitool kann auf Magazinplatz mit dieser Platztypnummer beladen werden`
`$TC_MTP_POS[ 3 ] = 2`	`; Positionswert = Platz 2`
`$TC_MTP_KD[ 3 ] = 3`	`; winkelcodierter Abstand`
`$TC_MTPPA[ 3, 1 ] = 0.0`	`; mit dieser Schreiboperation werden die beiden Plätze erzeugt`
`$TC_MTPPA[ 3, 2 ] = 180.0`	`;`
`$TC_MTP8[ 3 ] = 2`	`; freigegeben (Multitooldefinition ist abgeschlossen)`

Die beiden Parameter $TC_MTPN und $TC_MTP2 wurden bereits mit dem Befehl NEWMT definiert. Die Parameter $TC_MTP3,..., 6 wurden nicht programmiert. Es sind die Vorbelegungswerte wirksam.

11.10.10 DELT Werkzeug löschen

Mit der Funktion `DELT` wird durch Angabe von WZ-Bezeichner und Duplonummer ein Werkzeug gelöscht. Es können nur entladene Werkzeuge gelöscht werden.

Es werden alle werkzeugbezogenen Daten auf 0 gesetzt (Anwenderdaten, Hierarchiedaten, ...).

Syntax

```
DELT("WZ", DUPLO_NR)
```

Bedeutung

DELT	Löscht das angegeben Werkzeug	
	Datentyp:	-
WZ	Bezeichnung des Werkzeugs	
	Datentyp:	STRING
DUPLO_NR	Duplo-Nummer des Werkzeugs	
	Datentyp:	INT
Rückgabewert	-	

Beispiele

Funktion dient zum Löschen von Werkzeugen im Teileprogramm.

Programmcode	**Kommentar**
DELT("BOHRER", DUPLO_NR)	;
	;

11.10.11 DELMT - Multitool löschen

Die Funktion DELMT ist mit WZMG verfügbar.

Mit der Funktion DELMT wird ein Multitool gelöscht.

Das Löschen ist nur möglich, wenn das Multitool nicht in einem Magazin enthalten ist (entladen ist) und keines der im Multitool enthaltenen Werkzeuge, das bezüglich der Korrekturanwahl aktive Werkzeug ist. Falls zum Löschzeitpunkt noch Werkzeuge im Multitool enthalten sind, werden diese vor dem Löschvorgang automatisch aus dem Multitool ausgetragen. Die Werkzeuge werden dabei nicht gelöscht.

Wird ein Name angegeben, zu dem kein Multitool definiert ist, so wird der Befehl mit Alarm 17220 "Werkzeug existiert nicht" abgelehnt.

Hinweis

Nachdem das Multitool entladen wurde, sind auch die darin enthaltenen Werkzeuge entladen. Nachdem das Multitool gelöscht ist, sind die darin enthaltenen Werkzeuge wieder verfügbar, um in ein Magazin zu geladen werden, oder um in ein anderes Multitool bestückt zu werden.

Syntax

```
DELMT (name)
```

Bedeutung

`DELMT`	Löscht das Multitool mit der Bezeichnung `name`	
`name`	Bezeichnung des Multitools	
	Datentyp:	STRING

11.10.12 $TC_MTPN - Löschen eines oder aller Multitools

In Analogie zu $TC_TP1[TNr] = 0 ist es möglich mit $TC_MTPN einzelne oder alle Multitools zu löschen. Der Löschbefehl ist wie folgt definiert:

- $TC_MTPN[MTNr] = 0 ; lösche Multitool mit der Nummer MTNr > 0
- $TC_MTPN[0] = 0 ; lösche alle Multitools

Für MTNr > 0 wird das Multitool mit der Nummer MTNr gelöscht. Für MTNr = 0 werden alle Multitools gelöscht. Es gelten dieselben Regeln, wie sie für den Befehl `DELMT` definiert sind, siehe auch DELMT - Multitool löschen (Seite 171).

Syntax

```
$TC_MTPN[MTNr]
```

Bedeutung

`$TC_MTPN[MTNr]`	Liefert die Multitoolnummer oder löscht ein Multitool	
	Datentyp:	INT
`MTNr`	Nummer des Multitools	
	Datentyp:	INT
	Wertebereich:	0 < x < SLMAXTOOLNUMBER

Hinweis

Die Löschbefehle zum Löschen aller Werkzeuge ($TC_TP1[0] = 0 und $TC_DP1[0,0] = 0) löschen auch alle Multitools.

Nur mit dem Löschen des Multitools werden auch die MT-Plätze gelöscht.

11.10.13 GETT - T-Nummer lesen

Die Funktion GETT gibt anhand des Werkzeugbezeichners und dessen Duplonummer die dazu gehörige T-Nummer als Rückgabewert zurück.

Der Befehl kann für Werkzeuge sowie für Multitools verwendet werden. Bei Multitools kann keine Duplonummer programmiert werden.

Kann der Werkzeugbezeichner bzw. die Duplonummer keinem Werkzeug zugeordnet werden, so wird der Wert -1 zurückgeliefert.

Die Funktion findet z. B. Anwendung beim Nachladen von Werkzeugen über Teileprogramm.

Syntax

```
tNo = GETT("WZ", DUPLO_NR)
```

Bedeutung

GETT	Gibt die T-Nummer eines Werkzeugs oder Multitools zurück	
	Datentyp:	INT
WZ	Bezeichnung des Werkzeugs	
	Datentyp:	STRING
DUPLO_NR	Duplo-Nr. des Werkzeugs (optional)	
	Datentyp:	INT
	Ist die Duplo-Nr. nicht vorgegeben, wird die T-Nummer eines beliebigen Werkzeugs aus der Gruppe der Werkzeuge mit dem angegebenen Bezeichner oder die Nummer eines Multitools zurückgegeben (innerhalb einer WZ-Gruppe ist die Reihenfolge nicht definiert).	
Rückgabewert		
tNo	T-Nummer des Werkzeugs	
	Datentyp:	INT
	> 0	Werkzeug- oder Multitoolnummer zum programmierten Namen
	-1	Name ist weder Werkzeug- noch Multitoolname
	-2	Es wurde ein Multitoolname mit Duplonummer programmiert

Beispiele

Ermittle T-Nummer für Bohrer mit Duplonummer.

Programmcode	**Kommentar**
`R10=GETT("BOHRER", DUPLO_NR)`	`;in R10 steht die T-Nummer`
`$TC_TPx,[GETT("BOHRER",DUPLO_NR)]=Wert`	`;Schreiben der werkzeugbezogenen Daten`

11.10.14 SETPIECE - Stückzahlzähler dekrementieren

Mit der Funktion `SETPIECE` werden die Stückzahl-Überwachungsdaten der an dem Bearbeitungsprozess beteiligten Werkzeuge aktualisiert. Es werden alle Werkzeuge erfasst, die seit der letzten Aktivierung von `SETPIECE` eingewechselt wurden. Die Funktion dient in der Regel zur Programmierung am Ende des NC-Teileprogramms zum Dekrementieren der Stückzahl aller Werkzeuge, die an der Stückzahl-Überwachung beteiligt sind.

Hinweis

Der Befehl wirkt im Satzsuchlauf (mit/ohne Berechnung) nicht. Mit dem Wert = 0 für die Stückzahl wird die interne Tabelle gemerkter Werkzeuge/Schneiden gelöscht.

Hinweis

SETPIECE arbeitet "blind" die Tabelle ab. D. h., jede darin eingetragene Schneide wird erfasst. Dabei ist es unerheblich wo das WZ zum Zeitpunkt der SETPIECE-Programmierung sitzt, ob es im Magazin abgelegt wurde, auf der Spindel verbleibt oder auch entladen wurde.

Ist eingestellt, dass das Werkzeug über Reset hinaus aktiv bleibt und beim Programmstart befindet sich ein Werkzeug auf der Spindel, so erfolgt intern eine Anwahl genau dieses Werkzeugs und damit ein Eintrag in die Setpiece-Tabelle.

Syntax

```
SETPIECE(x,y)
```

Bedeutung

`SETPIECE`	Aktualisiert die Stückzahl-Überwachungsdaten	
	Datentyp:	-
`x := 0 ... 32000`	Wert, um den dekrementiert wird	
	Datentyp:	INT
`y := 0...8`	Spindelindex; der Wert 0 bedeutet Index der Hauptspindel (muss nicht programmiert werden)	
	Datentyp:	INT

Beispiele

Programmcode	**Kommentar**
`SETPIECE(1);`	`Werkstückzähler der Hauptspindel wird um 1 dekrementiert`
`SETPIECE(1,1);`	`Werkstückzähler der Spindelnr. bzw. WZ-Halternr. 1 wird um 1 dekrementiert`
`SETPIECE(4,2);`	`Werkstückzähler der Spindelnr. bzw. WZ-Halternr. 2 wird um 4 dekrementiert`

Beispiele für SETPIECE mit Wechselbefehl M06

Für ein Werkstück (Programm) sollen die beteiligten Werkzeuge um den Wert 1 dekrementiert werden.

Programmcode	Kommentar
T1	;T1 wird vorangewählt (bzgl. der Hauptspindel)
M06	;T1 wird gewechselt
D1	;D1 wird aktiv
T2	;T2 wird vorangewählt
:	;Bearbeitungsprogramm
:	
M06	;T2 wird gewechselt
D1	;D1 von T2 wird aktiv
T3	;T3 wird vorangewählt
:	;Bearbeitungsprogramm
:	
:	
M06	
T0	;Vorbereitung zum Leerräumen der Spindel
:	
M06	;Spindel leerräumen
SETPIECE(1)	;SETPIECE auf alle Werkzeuge
M30	

Je Werkzeug soll eine Dekrementierung erfolgen

In diesem Beispiel sollen die Werkzeuge T1, T2 und T3 ein Programm bearbeiten. Alle drei Werkzeuge sind Stückzahl überwacht. Es soll erreicht werden, dass Werkzeug T1 um den Wert 1, T2 um den Wert 2 dekrementiert und T3 nicht dekrementiert wird.

Dafür ist folgende Programmierung erforderlich:

Programmcode	Kommentar
N500 T1	
N600 M06	
N700 D1	;Mit der Korrekturanwahl wird das eingewechselte Werkzeug in den SETPIECE Speicher aufgenommen
N900 T2	;Vorbereitung nächstes Werkzeug
	;Bearbeitungsbefehl
:	
N1000 setpiece(1)	;SETPIECE wirkt auf T1, Setpiece Speicher wird gelöscht
N1100 M06	
N1200 D1	
N1400 T3	
:	;Bearbeitungsbefehle
:	
N1500 setpiece(2)	;Wirkt nur auf T2

Programmcode	**Kommentar**
N1600 M06	
N1700 D1	
:	;Bearbeitungsbefehle
:	
N1800 setpiece(0)	;Wirkt nur auf T3, keine Dekrementierung
N1900 T0	
N2000 M06	
N2100 D0	
N2300 M30	

11.10.15 GETSELT - Lesen der angewählten T-Nummer

Die Funktion GETSELT ist mit WZMO und WZMG verfügbar.

Der Funktion GETSELT liefert die T-Nummer des angewählten Werkzeug bezüglich der einzelnen Werkzeughalter bzw. Spindeln aus der Sicht des NC-Programms und dient dazu um zwischen der T-Programmierung und der Schneidenanwahl auf die Korrketurdaten des angewählten Werkzeugs zuzugreifen.

Die angewählte T-Nummer steht im Vorlauf bzw. dem NC-Programm erst nach der nächsten Schneidenanwahl zur Verfügung. Die Funktion GETSELT hat die Aufgabe, die Synchronisation mit dem Hauptlauf früher herzustellen.

ACHTUNG
Alarmverzögerung Falls eine Alarmverzögerung aktiv ist ($MC_TOOL_CHANGE_ERROR_MODE, Bit 0 = 1) und ein Alarm in der WZ-Vorbereitung auftritt, der nach M06 verzögert werden soll und vor der Programmierung von M06 GETSELT programmiert wird, dann hat die T-Nummer, die von GETSELT gefunden wird, den Wert = –1.

Syntax

```
GETSELT (Tno, th, ssl)
```

Bedeutung

GETSELT	Liest die angewählte T-Nummer aus	
	Datentyp:	-
Tno	T-Nummer des Werkzeugs	
	Datentyp:	INT
	= 0	Es gibt keine angewählte T-Nummer bzw. das Werkzeug wurde zwischenzeitlich gelöscht (tritt nur bei ssl="S" auf).
	= -1	Wenn die Alarmverzögerung aktiv ist und zwischen T- und M06-Programmierung GETSELT aufgerufen wird.

`th= 0,1, ..., ≤ 20`	Nummer der Spindel bzw. des Werkzeughalters (optional). Wird `th` nicht angegeben, wirkt dies wie th=0.	
	Datentyp:	INT
	• Wenn ssl="" bzw. nicht gesetzt ist, bezieht sich der Befehl auf die momentan wirksame Hauptspindel / wirksamen Haupt-Werkzeughalter • Wenn ssl="S" ist, bezieht sich der Befehl während des Satzsuchlaufs bzw. Testbetriebs auf den Werkzeughalter, der vor dem Satzsuchlauf die D-Korrektur bestimmt	
`ssl`	Satzsuchlauf (optional)	
	Datentyp	CHAR
	= ""	Funktion wie bisher, d. h., der aktuelle Zustand nach NC-Programm
	= "S"	Satzsuchlauf: Es wird das angewählte Werkzeug vor dem Satzsuchlauf bzw. Testbetrieb geliefert

Ist "`ssl`" gesetzt (="S" für "searchrun"), liefert der Sprachbefehl GETSELT während des Satzsuchlaufs bzw. Testbetriebs die T-Nummer des angewählten Werkzeuges auf dem angegebenen Werkzeughalters **vor dem Satzsuchlauf** bzw. **vor dem Testbetriebs**. Nach dem Ende des Satzsuchlaufs bzw. des Testbetriebs liefert der Sprachbefehl GETSELT nach der ersten T-Programmierung für den angegebenen Werkzeughalter den Wert unabhängig von ssl.

Nach Programmende/-abbruch werden die T-Nummern aller Werkzeughalter, die über GETSELT geliefert werden, auf den Wert = 0 gesetzt. Für den Master-Werkzeughalter wird anschließend der Wert entsprechend den Einstellungen in $MC_RESET_MODE_MASK gesetzt, oder nach erneutem Programmstart entsprechend den Einstellungen in $MC_START_MODE_MASK.

Die Funktion macht bei Bedarf einen impliziten Vorlaufstop, falls der Hauptlauf den WZ-Vorbereitungs-Befehl noch nicht bekommen hat.

BTSS-Variable

Die Werte, die die Funktion "GETSELT" liefert, stehen in der BTSS–Variable "toolHolderData" des Bausteins C/S zur Verfügung.

Mit dem Parameter-Nr. 2 bekommt man den Wert von `GETSELT(tNo, Th)`.

Mit dem Parameter-Nr. 4 bekommt man den Wert von `GETSELT(tNo, Th, "S")`.

Die BTSS–Variable steht in allen Varianten der Werkzeugverwaltung zur Verfügung.

Beispiel (1)

Für den programmierten Wert der WZ-Halternummer = 3 gibt es in der Magazinverwaltung keinen WZ-Halterplatz**.**

Programmcode	**Kommentar**
`N110 GETSELT ( tNo, 3 )`	`; tNo = 0 ist das Ergebnis.`

Beispiel (2)

Für den programmierten WZ-Halter mit der Nummer = 4 ist keine WZ-Vorbereitung programmiert und es ist kein WZ eingewechselt.

Programmcode	Kommentar
N110 GETSELT (tNo, 4)	; tNo = 0 ist das Ergebnis.

Beispiel (3)

Es gibt ein Werkzeug mit Name “gesperrt” und der T-Nummer = 5.

Programmcode	Kommentar
def int tNo	
N100 T=”gesperrt”	; Alarm ’kein einsatzfaehiges WZ’ wird ; verzoegert bis zugehoeriges M06 programmiert ; wird, bzw. bis erneut T programmiert wird.
N110 GETSELT (tNo)	; tNo = –1 ist das Ergebnis
N120 M06	; der verzoegerte Alarm aus N100 wird ausgegeben

BTSS:

Die BTSS-Variable "progTNumber" im Variablenbaustein C/S zeigt in diesem Fall die zuvor programmierte T-Nummer an, bzw. den Wert = 0, falls vor der fehlerhaften T-Anwahl noch keine andere T-Anwahl programmiert wurde.

Beispiel (4)

Im Reset-Zustand ist das Werkzeug mit der T-Nummer=5 in der Hauptspindel eingewechselt. Es wird ein Satzsuchlauf auf den Satz “N100” im folgenden NC-Programm gestartet.

Programmcode	Kommentar
def int tNo, tNo1	
N70 T7	
N80 GETSELT(tNo)	; tNo = 7
N90 GETSELT(tNo1,,”S”)	; tNo1=5
N100 X1 G0	; Zielsatz
N110 GETSELT(tNo)	; tNo = 7
N120 GETSELT(tNo1,,”S”)	; tNo1=5, da noch kein neues T nach dem Ziel- ; satz programmiert wurde
N130 T8	
N140 GETSELT(tNo)	; tNo = 8
N150 GETSELT(tNo1,,”S”)	; tNo1=8, da ein neues T programmiert wurde

BTSS:

Die BTSS-Variable "progTNumberSSL" im Variablenbaustein C/S zeigt in diesem Fall die zuvor programmierte T-Nummer an, bzw. den Wert = 0, falls vor der fehlerhaften T-Anwahl noch keine andere T-Anwahl programmiert wurde.

Siehe auch Kap. Satzsuchlauf (SSL) in Verbindung mit aktiver Werkzeugverwaltung

11.10.16 GETEXET - Lesen der eingewechselten T-Nummer

Die Funktion `GETEXET` ist mit WZMO und WZMG verfügbar.

Die Funktion GETEXET ist speziell für den Satzsuchlauf gedacht. Sie wird analog zu GETSELT parametriert und liefert die T-Nummer des, aus Sicht des NC-Programms, aktiven Werkzeugs.

Syntax

```
GETEXET(Tno, th, ssl)
```

Bedeutung

`GETEXET`	Liest die angewählte T-Nummer aus	
	Datentyp:	-
`Tno`	T-Nummer des Werkzeugs	
	Datentyp:	INT
	= 0	Es gibt keine eingewechselte T-Nummer bzw. das Werkzeug wurde zwischenzeitlich gelöscht (tritt nur auf, wenn ssl="S").
`th= 0,1, ..., ≤ 20`	Nummer der Spindel bzw. des Werkzeughalters (optional). Wird `th` nicht angegeben, bezieht sich GETEXET auf die aktuelle Masterspindel.	
	Datentyp:	INT
	= 0:	Wenn ssl= "" bzw. nicht gesetzt ist, bezieht sich der Befehl auf die momentan wirksame Hauptspindel/Werkzeughalter. Wenn ssl= "S" ist, bezieht sich der Befehl auf den Werkzeughalter, der vor dem Satzsuchlauf die D-Korrektur bestimmt.
`ssl`	Satzsuchlauf (optional)	
	Datentyp	CHAR
	= "":	Funktion wie bisher, d. h., der aktuelle Zustand nach NC-Programm
	= "S":	Satzsuchlauf-Funktion: Es wird das eingewechselte Werkzeug vor dem Satzsuchlauf bzw. Testbetrieb geliefert.

Ist "ssl" gesetzt (="S" für "searchrun"), liefert der Sprachbefehl GETEXT während des Satzsuchlaufs bzw. Testbetriebs die T-Nummer des eingewechselten Werkzeugs auf dem angegebenen Werkzeughalter **vor dem Satzsuchlauf** bzw. **vor dem Testbetrieb**. Nach dem Ende des Satzsuchlaufs bzw. Testbetriebs liefert der Sprachbefehl GETEXT nach der ersten Programmierung eines Werkzeugwechsels für den angegebenen Werkzeughalter den Wert unabhängig von ssl.

Die Funktion macht bei Bedarf einen impliziten Vorlaufstopp, wenn der Hauptlauf den Wechselbefehl noch nicht bekommen hat.

Nach Programmende/-abbruch werden die "getexet"-T-Nummern aller WZ-Halter auf den Wert = 0 gesetzt. Für den Master-WZ-Halter wird anschließend der Wert entsprechend den

Einstellungen in $MC_RESET_MODE_MASK gesetzt, oder nach erneutem Programmstart entsprechend den Einstellungen in $MC_START_MODE_MASK.

Hinweis

Falls der Werkzeugwechsel allein mit T programmiert wird ($MC_TOOL_CHANGE_MODE), führen GETSELT und GETEXET zur selben T-Nummer.

Falls der WZ–Wechsel mit T (Anwahl) und M06 (Wechsel) programmiert werden, so sind die beiden Befehle inhaltlich unterschiedlich.

BTSS-Variable

Die Werte, die die Funktion "GETEXET" liefert, stehen in der BTSS–Variable "toolHolderData" des Bausteins C/S zur Verfügung.

Mit der Parameter-Nr. 3 bekommt man den Wert von `GETEXET(tNo, Th)`.

Mit der Parameter-Nr. 5 bekommt man den Wert von `GETEXET(tNo, Th, "S")`.

Die BTSS-Variable steht in allen Varianten der Werkzeugverwaltung zur Verfügung.

Beispiele (1)

Es sind drei Werkzeuge definiert mit Tnr. = 5 / Name = "WZ5" , Tnr. = 7 / Name = "WZ7" und Tnr. = 9 / Name = "WZ9".

Der WZ–Wechsel wird mit T + M6 programmiert.

Es sind zwei WZ-Halter definiert. WZ-Halter 2 ist der Master-WZ-Halter.

Zu Beginn ist weder ein WZ auf den WZ-Haltern, noch ist ein WZ aktiv oder programmiert:

Programmcode	**Kommentar**
`def int tNr1, tNr2`	
`T="WZ5"  ; WZ-Anwahl für Master-WZ-Halter`	
`getselt ( tNr2, 2 )`	`;tNr2 = 5 vorangewaehltes WZ fuer Master-WZ-Halter`
`getselt ( tNr1, 1 )`	`;tNr1 = 0`
`getexet ( tNr2, 2 )`	`;tNr2 = 0 kein aktives WZ fuer Master-WZ-Halter`
`getexet ( tNr1, 1 )`	`;tNr1 = 0`
`M6  ; WZ-Wechsel für Master-WZ-Halter`	
`getselt ( tNr2, 2 )`	`;tNr2 = 5`
`getexet ( tNr2, 2 )`	`;tNr2 = 5`
`T="WZ9"  ; WZ-Anwahl für Master-WZ-Halter`	
`getselt ( tNr2, 2 )`	`;tNr2 = 9`
`getexet ( tNr2, 2 )`	`;tNr2 = 5`
`T1="WZ7"  ; WZ-Anwahl für Neben-WZ-Halter`	

Programmcode	Kommentar
getselt (tNr2, 2)	;tNr2 = 9 vorangewaehltes WZ fuer Master-WZ-Halter
getselt (tNr1, 1)	;tNr1 = 7 vorangewaehltes WZ fuer Neben-WZ-Halter
M1=6 ; WZ-Wechsel für Neben-WZ-Halter	
getexet (tNr1, 1)	;tNr1 = 7 aktives WZ fuer Neben-WZ-Halter

Beispiel (2)

Im Reset–Zustand ist das Werkzeug mit der T-Nummer=5 in der Hauptspindel eingewechselt. Es wird ein Satzsuchlauf auf den Satz "N100" im folgenden NC–Programm gestartet.

Programmcode	Kommentar
def int tNo, tNo1	
N70 T7 M6	
N75 T8	
N80 GETEXET(tNo)	;tNo =7
N90 GETSELT(tNo1,,"S")	;tNo1=5
N100 X1 G0	;Zielsatz
N110 GETEXET(tNo)	;tNo =7
N120 GETEXET(tNo1,,"S")	;tNo1=5, da noch kein neues WZ nach dem Ziel-;satz programmiert wurde
N130 M6	
N140 GETEXET(tNo)	;tNo=8
N150 GETEXET(tNo1,,"S")	;tNo1=8, da ein neues WZ eingewechselt wurde

BTSS:
Die BTSS-Variable "progTNumberSSL" im Variablenbaustein C/S zeigt in diesem Fall die zuvor programmierte T-Nummer an, bzw. den Wert = 0, falls vor der fehlerhaften T-Anwahl noch keine andere T-Anwah- programmiert wurde.

Gebrauch von $P_TOOLNO - Getexet

Mit aktiver Werkzeugverwaltung sollte generell mit GETEXET gearbeitet werden.

Falls man sich an folgende Vorgaben hält, kann auch $P_TOOLNO verwendet werden:

Nach der Programmierung von D liefert $P_TOOLNO immer den korrekten Wert der aktiven T-Nummer.

Programmcode	Kommentar
T2 M6 D1	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
r1 = $P_TOOLNO	;r1 = 2

Für $MC_CUTTING_EDGE_DEFAULT=-1, >0 liefert $P_TOOLNO immer den korrekten Wert der aktiven T-Nummer.

Programmcode	Kommentar
T2 M6	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
r1 = $P_TOOLNO	;r1 = 2

Ausnahmen gibt es für $MC_CUTTING_EDGE_DEFAULT=-2 und 0.

a) $MC_CUTTING_EDGE_DEFAULT = 0

Programmcode	Kommentar
T2 M6	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
r1 = $P_TOOLNO	;!!!r1 = ? - die T-Nr. wird erst im Hauptlauf bestimmt. ;!!!D.h. sie steht hier i.a. noch nicht zur Verfügung.
def int tNo	
T2 M6	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
GETEXET(tNo, $PMTHSDC)	;sichere Ermittlung der aktiven Tnr. tNo = 2

b) $MC_CUTTING_EDGE_DEFAULT = -2

Programmcode	Kommentar
N10 T2 M6	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
r1 = $P_TOOLNO	;!!!r1 = 22 - falls T-Nr.=22 das vor N10 aktives WZ war
def int tNo	
N10 T2 M6	;WZ-Gruppe "2" hat genau ein WZ mit Tnr. = 2
GETEXET(tNo, $P_MTHSDC)	;sichere Ermittlung der einzuwechselnden aktiven Tnr. ;tNo = 2 - Die Tnr. wird erst mit der naechsten ;Programmierung von D>0 zur aktiven Tnr.

D. h., für $MC_CUTTING_EDGE_DEFAULT=-2 können sowohl $P_TOOLNO als auch GETEXET benutzt werden, und zwar in verschiedener Bedeutung.

Siehe auch Kap. Satzsuchlauf (SSL) in Verbindung mit aktiver Werkzeugverwaltung.

11.10.17 $P_MTHSDC - Master-Toolholder bezüglich der D-Korrekturanwahl

Die Systemvariable $P_MTHSDC steht mit WZMG zur Verfügung.

Die Systemvariable $P_MTHSDC (**M**aster **t**ool **h**older regarding selected **D**–Correction) liefert die Nummer des WZ–Halters/der Spindel auf dem der letzte WZW auf einen Master-Toolholder/-Spindel stattfand. D.h. das ist der WZ–Halter, auf dem das WZ sitzt, das die nächste D–Korrektur bestimmt und damit aktives WZ wird (wenn nicht bereits vorher dieses Werkzeug aktiv war).

Syntax

```
$P_MTHSDC
```

Bedeutung

$P_MTHSDC	Liefert die Nummer des Werzeughalters/der Spindel	
	Datentyp:	INT
Rückgabewert		
	> 0	Erfolgreicher Lesezugriff
	= 0	Kein Werkzeughalter bzw. Spindel verfügbar
	-1	WZMG nicht verfügbar

Beispiele

Einstellungen:

$MC_CUTTING_EDGE_DEFAULT = -2,
d. h., mit M06 bleibt die alte WZ-Korrektur weiter aktiv, falls dazu nicht explizit D programmiert wird.

$MN_TOOL_CHANGE_MODE = 0

$MC_TOOL_MANAGEMENT_TOOLHOLDER = 2

```
Programmcode                Kommentar
N10 SETMTH ( 1)
N20 T="Wz5"                 ;T-Nr. = 5 ist das einzige WZ in der WZ-Gruppe
                            ;WZ-Wechsel auf aktuellen Mastertoolholder (WZ-Halter 1)
N30 D5                      ;Korrektur des zuletzt auf einen Mastertoolholder gewech-
                            ;selten WZs (hier t-Nr. 5)
N100 SETMTH ( 2)            ;Wechsel des Mastertoolholders
N110 T="Wz7"                ;T-Nr. = 7 ist das einzige WZ in der WZ-Gruppe
                            ;WZ-Wechsel auf aktuellen Mastertoolholder (WZ-Halter 2)
N120 SETMTH ( 1)            ;Wechsel des Mastertoolholders
...                         ;weitere Sätze ohne T, M6, D
N130 D5                     ;Korrektur mit D5 des Werkzeugs, das zuletzt auf einen
                            ;Mastertoolholder eingewechselt wurde (hier T=7 auf Tool-
                            ;holder 2). Erst jetzt wird ein neues aktives WZ festge-
                            legt
```

Damit nach einem Satzsuchlauf auf einen Satz zwischen N120 und N130 die D-Anwahl in N130 sich auf das richtige WZ bezieht (hier "Wz7"), muss der WZW für "Wz7" auf Toolholder 2 als Master-Toolholder erfolgen und als letzter WZW erfolgen. Dies ist über eventuelle ASUP-–Programme sicherzustellen. Mit den bisherigen zur Verfügung stehenden Sprachbefehlen und Variablen kann diese Information nicht ermittelt werden. Die Variable enthält die Information, auf welchen WZ–Halter der letzte WZW durchgeführt wurde, als dieser Master–WZ–Halter war.

Im obigen Fall enthält die Variable $P_MTHSDC den Wert 2, denn auf diesen WZ–Halter wurde der letzte Wechsel auf einen Master-WZ-Halter gemacht. Mit Hilfe von GETEXET(TNo,2) bzw. if ($P_MTHSDC > 0) GETEXET(TNo,$P_MTHSDC) kann das Werkzeug ermittelt werden, das auf diesen WZ–Halter gehört, damit eine korrekte Programmfortsetzung möglich ist.

$P_MTHSDC = 0 bedeutet, dass aktuell kein WZ-Halter definiert ist, der nach einer D-Programmierung das aktive WZ hat.

11.10.18 $P_TH_OF_D - Master Toolholder bezüglich der aktuellen D-Korrektur

Die Systemvariable $P_TH_OF_D ist mit WZMO und WZMG verfügbar.

Die Systemvariable $P_TH_OF_D (Tool holder of active D-correction) liefert die Nummer des WZ-Halters / der Spindel, auf der das Werkzeug sitzt bzw. saß, das die aktive D-Korrektur bestimmt.

Syntax

```
$P_TH_OF_D
```

Bedeutung

$P_TH_OF_D	Liefert die Nummer des Werkzeughalters/Spindel	
	Datentyp:	INT
	> 0	erfolgreicher Lesezugriff
	0	Kein Werkzeughalter verfügbar
	-1	Funktion nicht verfügbar

Beispiele

Einstellungen:

$MC_CUTTINGEDGE_DEFAULT = -2, d. h., mit M06 bleibt die alte WZ-Korrektur weiter aktiv, falls nicht explizit D programmiert wird.

$MN_TOOLCHANGE_MODE = 0

$MC_TOOL_MANAGEMENT_TOOLHOLDER = 2

Programmcode	**Kommentar**
N00	; ;kein Werzeug aktiv → $P_TH_OF_D =
N10 SETMTH(1)	
N20 T="Wz5"	;T-Nr. = 5 ist das einzige WZ in der WZ-Gruppe;WZ-Wechsel auf aktuellen Master-Toolholder (WZ-Halter 1)
N30 D5	;Korrektur ist vom Werkzeug, das auf dem Werkzeughalter 1;sitzt → $P_TH_OF_D = 1
N100 SETMTH(2)	;Wechsel des Master-Toolholders
N110 T="Wz7"	;T-Nr. = 7 ist das einzige WZ in der WZ-Gruppe ;WZ-Wechsel auf aktuellen Master-Toolholder (WZ-Halter 2)
N120 SETMTH(1)	;Wechsel des Master-Toolholders
...	;weitere Sätze ohne T, M6, D
N130 D5	;Korrektur mit D5 des Werkzeugs, das zuletzt auf einen;Master-Toolholder eingewechselt wurde (hier T=7 auf Tool-;holder 2) → $P_TH_OF_D = 2

In einem ProgEvent-Programm soll nach einem Satzsuchlauf bezüglich der Werkzeuge und Werkzeugkorrektur genau die Situation hergestellt werden, die an dieser Stelle ohne Satzsuchlauf wäre.

Nach einem Satzsuchlauf auf einen Satz zwischen N100 und N130 wird bestimmt, wie das gerade aktive "D5" aus dem Satz N30 Zustande kommt. Dazu muss erkannt werden, dass "D5" von dem Werkzeug auf dem Werkzeughalter 1 kommt. Der letzte Werkzeugwechsel auf einen Haupt-Werkzeughalter erfolgte im Satz N20 auf den Werkzeughalter 1. Diese Information liefert die Systemvariable $P_TH_OF_D.

Die Variable $P_TH_OF_D gibt an, auf welchen Werkzeughalter sich das Werkzeug befindet, das die **gerade aktive D-Korrektur** bestimmt.

Die Variable $P_MTHSDC gibt an, auf welchen Werkzeughalter sich das Werkzeug befindet, das bei der **nächsten D-Programmierung** die Korrektur bestimmt.

$P_TH_OF_D = 0 bedeutet, dass aktuell kein Werkzeughalter definiert ist, auf den sich die aktuelle Werkzeugkorrektur "D" bezieht. Damit ist auch das aktive Werkzeug T=0 und die aktive D-Nr. =0.

11.10.19 GETACTT - Lesen der aktiven internen T-Nummmer

Die Funktion GETACTT liefert die T-Nummer des Werkzeugs mit dem Status "aktiv" und "war im Einsatz" über den Parameter "TNr" aus einer Werkzeuggruppe mit dem Bezeichner "name" (ein WZ wird "aktiv", unmittelbar, bevor es in den WZ-Halter eingewechselt wird)

Hinweis

Mit GETACCT kann ein Werkzeug, das im Ersteinsatz auf der Spindel sitzt, nicht erfasst werden.

Die WZ-Reihenfolge in einer Gruppe ist nicht definiert. D. h., GETACCT wird ein beliebiges WZ der Gruppe lesen, bei dem die Status-Bits "aktiv" und "war im Einsatz" sitzen.

Syntax

```
status=GETACTT(Tno,name)
```

Bedeutung

GETACTT	Liefert die T-Nummer des aktiven Werkzeugs	
	Datentyp:	INT
Tno	T-Nummer des aktiven Werkzeugs	
	Datentyp:	INT
name	Name der Werkzeuggruppe	
	Datentyp:	STRING
Rückgabewert		
status	Zeigt den Status Auslesens an	

	Datentyp:	INT
	0	Erfolgreich durchgeführt; Tnr. enthält gewünschten Wert
	-1	Zum angegebenen Bezeichner existiert kein WZ; TNr. enthält den Wert = 0
	-2	In der WZ-Gruppe gibt es kein WZ mit dem gewünschten Status; TNr. enthält den Wert = 0
	-3	In der WZ-Gruppe gibt es mehrere Werkzeuge mit dem gewünschten Status; Tnr. enthält den Wert des ersten WZs mit dem gewünschten Zustand

GETACTT kann mehrdeutig sein! Es ist immer denkbar, dass in einer WZ-Gruppe mehrere WZe denselben Status haben. Der Befehl wird nur dann sinnvoll funktionieren, wenn der Anwender dafür sorgt, dass er in der WZ-Gruppe genau ein WZ mit dem gewünschten Status hat.

Der Befehl veranlasst keine Hauptlaufsynchronisation. Evtl. muss vor dem Aufruf STOPRE eingegeben werden.

Beispiel

Die WZ-Gruppe "Bohrer" hat drei WZe mit den Duplonummern 1,2,3 und den T-Nummern 1,2,3:

```
Programmcode                          Kommentar
def int Tno, status                   ;in der WZ-Gruppe "Bohrer"sei zunächst kein aktives WZ
status=GETACTT(Tno, "Bohrer")         ;status=-2, Tno=0
T="Bohrer"                            ;Vorbereitung setzt WZ-Status auf "aktiv"
status=GETACTT(Tno, "Bohrer")         ;status=0, Tno=0
                                      ;das WZ ist zwar aktiv, die Kennung "war im Einsatz" sitzt aber nicht
M06                                   ;Wechsel
T="Hugo"                              ;Vorbereitung
status=GETACTT(Tno, "Bohrer")         ;status=-2, Tno=0
                                      ;das WZ ist zwar aktiv, die Kennung "war im Einsatz" sitzt aber immer noch nicht
M06                                   ;Wechsel
status=GETACTT(Tno, "Bohrer")         ;status=0, Tno=1
                                      ;Leseauftrag wird durchgeführt
                                      ;das WZ "Bohrer" hat jetzt, durch das Auswechseln, den Status "war im Einsatz" der Status "aktiv" sitzt unverändert
```

11.10.20 SETMS - Masterspindel setzen

Verfügbar mit WZMO und WZMG.

Die Funktion `SETMS` erklärt die unter `n` angegebene Spindel zur Masterspindel. Die Definition als Masterspindel ist auch über ein Maschinendatum möglich.

Mit SETMS ohne Spindelangabe wird auf die im Maschinendatum festgelegte Spindel zurückgeschaltet.

Syntax

```
SETMS(n)
```

Bedeutung

SETMS	Erklärt die unter n angegebene Spindel zur Masterspindel	
	Datentyp:	-
n	Nummer der Spindel	
	Datentyp:	INT

Reset-Verhalten

Die programmierten Werte von SETMS können über Programmende/RESET/START hinweg aktiv bleiben.

11.10.21 SETMTH Masterwerkzeughalternummer setzen

Die Funktion SETMTH ist verfügbar mit WZMG.

Mit der Funktion SETMTH wird der Werkzeughalte mit der Nummer n zum Masterwerkzeughalter werden.

Über das Maschinendatum **MD20124 TOOL_MANAGEMENT_TOOLHOLDER** kann eingestellt werden, ob statt einer Spindelnummer eine Werkzeughalternummer vergeben wird, um den Einsatzort eines einzuwechselnden Werkzeuges festzulegen. Der Sprachbefehl ist nur sinnvoll einzusetzen, wenn das MD > 0 ist.

Hinweis

SETMTH ändert nicht das aktive Werkzeug. Erst der anschließende programmierte Werkzeugwechsel kann die neue Festlegung in Bezug auf den Master-WZ-Halter berücksichtigen.

Syntax

```
SETMTH(n)
```

Bedeutung

SETMSH	Erklärt den unter n angegebenen Werkzeughalter zum Masterwerkzeughalter	
	Datentyp:	-

n	Nummer des Werkzeughalters	
	Datentyp:	INT

Beispiel (1)

Programmierbeispiel

Programmcode	**Kommentar**
T="Fraeser" M06	;Keine Adresserweiterung programmiert → der Master-WZ-Halter ist gemeint; d.h. WZ-Halter 1 (Wert des Maschinendatums TOOL_MANAGEMENT_TOOLHOLDER). Der Werkzeugwechsel erfolgt auf den Zwischenspeicherplatz mit $TC_MPP5=1. Die Bahn wird mit den Korrekturen des Werkzeugs korrigiert.
...	
T2="Bohrer" ..M2=6	;Adresserweiterung für den Neben-WZ-Halter wurde programmiert. Der Werkzeugwechsel erfolgt auf den Zwischenspeicherplatz 2. Die Bahn wird nicht korrigiert.
...	
SETMTH (2)	;erkläre WZ-Halter 2 zum Master-WZ-Halter
T="Fraeser_2""M06	;keine Adresserweiterung programmiert → der Master-WZ-Halter ist gemeint; d. h., WZ-Halter 2. Der Werkzeugwechsel erfolgt auf den Zwischenspeicherplatz 2. Die Bahn wird mit der Werkzeugkorrektur korrigiert.
...	
T1="Bohrer_1" M1=6	;Adresserweiterung für die Neben-WZ-Halter wurde programmiert. Der Werkzeugwechsel erfolgt auf den Zwischenspeicherplatz mit $TC_MPP5=1. Die Bahn wird nicht korrigiert.
...	
SETMTH	;erkläre den durch TOOL_MANAGEMENT_TOOLHOLDER gegebenen WZ-Halter zum Master-WZ-Halter
T="Fraeser_3" M06	;keine Adresserweiterung programmiert → der Master-WZ-Halter ist gemeint; d. h., WZ-Halter 1 (Wert des MD TOOL_MANAGEMENT_TOOLHOLDER). Der Werkzeugwechsel erfolgt auf den Zwischenspeicherplatz 1. Die Bahn wird mit der Werkzeugkorrektur korrigiert.

Beispiel (1)

Es gilt:

```
$MC_RESET_MODE_MASK = "H18041"
$MC_SPIND_DEF_MASTER_SPIND = 1
```

```
$MC_TOOL_MANAGEMENT_TOOLHOLDER = 2
```

Nach Programmende/RESET bleibt sowohl die aktive Werkzeugkorrektur aktiv, als auch die programmierten Werte von SETMTH und SETMS. Weiterhin erfolgt der Werkzeugwechsel nicht auf die Spindel, sondern auf den Werkzeughalter.

Programmcode	Kommentar
T="Bohrer" M06 D2	;Werkzeugwechsel auf Master-WZ-Halter=2
SETMS(3)	;Neue Master-Spindel=3
SETMTH(1)	;Neuer Master-WZ-Halter=1
T="Fraeser" M06 D1	;Werkzeugwechsel auf Master-WZ-Halter=1
M17	

Nach Programmende bzw. RESET gilt Folgendes:

- Spindelnr. = 3 die Master-Spindel
- WZ-Halternr. =1 der Master-WZ-Halter
- WZ = "Fraeser" mit Korrektur D1 bestimmt die Bahnkorrektur

Nach Power On kommen die Einstellungen der Maschinendaten zum Tragen:

- Spindelnr. = 1 ist die Master-Spindel
- WZ-Halternr. = 2 ist der Master-WZ-Halter

Die Werkzeugkorrektur ergibt sich aus der kleinsten D-Nummer des Werkzeugs, das sich auf dem Master-WZ-Halter befindet; d. h., das ist T="Bohrer" mit D1

(In der Annahme, dass das Werkzeug zwei D-Korrekturen D1, D2 hat).

Beispiel (2)

Es gilt

```
$MC_RESET_MODE_MASK = "H41"
$MC_SPIND_DEF_MASTER_SPIND = 1
$MC_TOOL_MANAGEMENT_TOOLHOLDER = 0
```

Nach Programmende/RESET bleibt sowohl die aktive Werkzeugkorrektur aktiv als auch der programmierte Wert SETMS. Weiterhin erfolgt der Werkzeugwechsel auf die Spindel, die damit zum Werkzeughalter wird.

Programmcode	Kommentar
T="Bohrer" M06 D2	;WZ-Wechsel auf Master-WZ-Halter=1
SETMS(3)	;Neue Master-Spindel = Master-WZ-Halter=3
T="Fraeser" M06 D1	;WZ-Wechsel auf Master-WZ-Halter=3
M17	

Nach Programmende bzw. RESET gilt Folgendes:

- Spindelnr. = 1 ist die Master-Spindel
- WZ = "Fraeser" mit Korrektur D1 (das auf der Spindel mit Nr. = 3 ist) bestimmt die Bahnkorrektur

Nach Power On kommen die Einstellungen der Maschinendaten zum Tragen:

- Spindelnr. = 1 ist die Master-Spindel / der Master-WZ-Halter.
- Die Werkzeugkorrektur ergibt sich aus der kleinsten D-Nummer des Werkzeugs, das sich auf dem Master-WZ-Halter befindet, das ist T="Bohrer" mit D1 (in der Annahme, dass das Werkzeug zwei D-Korrekturen D1, D2 hat).

Reset-Verhalten

Die programmierten Werte von SETMTH können über Programmende/RESET/START hinweg aktiv bleiben.

11.10.22 POSM - Magazin positionieren

Die Funktion POSM positioniert ein Magazin zu einem bestimmten Platz eines internen Magazins (z. B. Spindel/WZ-Halter, Belademagazin), unabhängig von der momentanen Belegung des Platzes und vom Status des enthaltenen Werkzeuges. Der Sprachbefehl deckt Teile des BTSS-PI-Dienstes _N_TMPOSM ab.

Hinweis

Der Sprachbefehl POSM(...) wird, ohne eine Quittierung von der PLC abzuwarten, beendet.

Syntax

```
POSM (p, m, ip, im)
```

Bedeutung

POSM	Positioniert ein Magazin zu einem bestimmten Platz	
	Datentyp:	-
p	Platznummer, auf die positioniert werden soll	
	Datentyp:	INT
m	Magazinnummer des Magazins, das bewegt werden soll (optional). Wird m nicht angegeben, bezieht sich die Platznummer auf das Magazin, das als Erstes in der Distanztabelle zum genannten internen Platz enthalten ist. Das Magazin (Nummer m) muss über eine Distanzbeziehung mit dem ausgewählten Belade- bzw. Zwischenspeichermagazinplatz verbunden sein. Bei Angabe falscher Parameter werden Alarme erzeugt (z. B. bei nicht definierten Platznummern).	
	Datentyp:	INT
ip	Platznummer des genannten internen Magazins (Spindelplatz, Belademagazin usw.) Der Parameter ist optional. Wird ip nicht angegeben, bezieht sich der Positioniervorgang auf den Hauptspindelplatz bzw. den Hauptwerkzeughalterplatz.	
	Datentyp:	INT

im	Magazinnummer des internen Magazins bezogen auf die Platznummer ip, zu der das Magazin bewegt werden soll. Ein internes Magazin ist entweder ein Belade- oder Zwischenmagazin. Der Parameter ist optional. Wird im nicht angegeben, bezieht sich der Befehl auf das Zwischenspeichermagazin.	
	Datentyp	INT

Beispiel

Vorgegebene Konfiguration:

- Magazin (Magazinnummer = 1),
- Spindel (Zwischenspeichermagazin = 9998, Platz 1),
- Belademagazin (Belademagazin = 9999, Platz 2).

Programmcode	Kommentar
N100 POSM(4, 1, 1, 9998)	; Es wird vom Magazin 1 der Platz Nummer 4 zur Spindel verfahren.
N100 POSM(4, 1, 1, 9999)	; Befehl für das Verfahren zum Belademagazin

Beispiel mit Ergebnisprüfung

Vorgegeben ist ein Magazin, wie es im folgenden Bild dargestellt ist.

Es soll der Platz 12 an die Wechselstelle positioniert und das Programm erst fortgesetzt werden, wenn das Positionieren erfolgreich beendet wurde (einfachster Fall mit nur einem Magazin und einer definierten Wechselstelle).

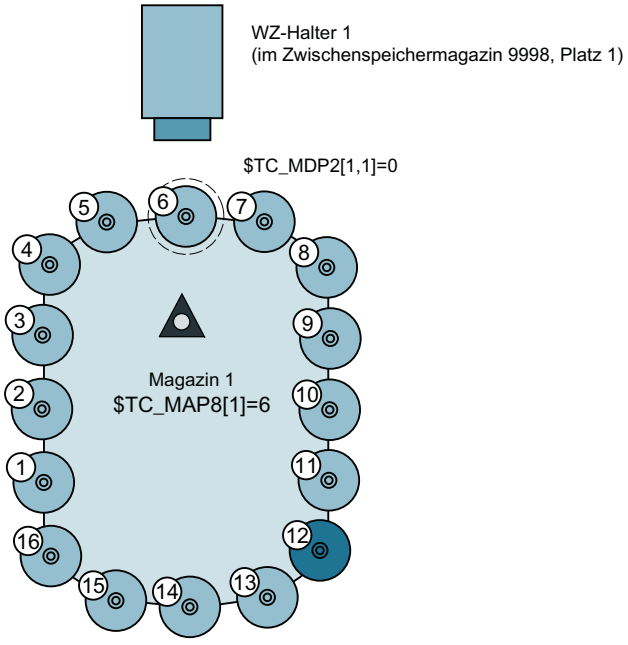


Bild 11-12 Magazinpositionierung mit Ergebnisprüfung des Positioniervorgangs

Der Magazinnullpunkt befindet sich hier im Platz vor dem WZ-Halter 1. Er wird definiert durch die Systemvariable $TC_MDP2. Der WZ-Halter 1 ist der Masterspindel des Kanals zugeordnet.

Programmcode	Kommentar
N100 POSM(12)	;positioniert Platz 12 zur Wechselstelle, die nicht programmierten Parameter werden intern zu POSM (12, 1, 1, 9998) ergänzt
N200 warten:	
N300 G4 F1	;den Verhältnissen an der Maschine entsprechende passende Wartezeit (eventuell Aussprung nötig, falls auf misslungene Positionierung reagiert werden soll)
N400 if ($TC_MAP8[1] <> 12)	;goto warten
	;nachdem POSM(12) ausgeführt ist, muss die aktuelle Magazinposition gleich 12 sein.

Multitool

Die Funktion POSM bleibt in seiner Definition unverändert. Es wird der Magazinplatz programmiert, auf den positioniert werden soll; unabhängig davon, ob der Platz leer ist, ein Werkzeug oder ein Multitool enthält.

Der PI-Dienst _N_TMPOSM bietet verschiedene Möglichkeiten zur Programmierung der Magazinposition, z. B. kann man das WZ über seine T-Nummer oder seinen Namen und Duplonummer programmieren. Sitzt dieses Werkzeug auf einem Magazinplatz, dann wird auf diesen Magazinplatz positioniert. Sitzt das WZ in einem Multitool, das seinerseits auf einem Magazinplatz sitzt, dann wird auf diesen Magazinplatz positioniert. Wird statt einer T–Nummer eine MT–Nummer programmiert und das Multitool sitzt auf einem Magazinplatz, wird auf diesen Magazinplatz positioniert.

11.10.23 POSMT - Multitool auf WZ-Halter auf Platznummer positionieren

Die Funktion POSMT ist für WZMG verfügbar.

Die Funktion POSMT positioniert ein Multitool, das sich auf einem Werkzeughalter befindet, auf die programmierte MT-Platznummer. Es darf zum Zeitpunkt der Ausführung des Befehls keine WZ-Korrektur für den programmierten WZ-Halter aktiv sein, d. h., D0 muss zuvor ausgeführt worden sein.

Der vollständige Befehl lautet: POSMT(state, MTlocno, THno)

Hinweis

Eine Korrekturanwahl bzw. eine Abwahl einer möglicherweise aktiven WZ-Korrektur ist mit dem Befehl nicht verbunden.

Der Befehl löst ein implizites STOPRE aus.

Syntax

```
state = POSMT(state, MTlocno, THno)
```

Bedeutung

POSMT	Positioniert ein Multitool auf eine programmierte MT-Platznummer	
	Datentyp:	INT
MTlocnox	MT-Platznummer des MTs, das sich auf dem programmierten WZ-Halter befindet.	
	Datentyp:	INT
THno	WZ-Halternummer, auf dem sich das zu Positionierende MT befindet. Der Parameter ist optional. Wird THno nicht programmiert, wird automatisch die Nummer des Maste-WZ-Halters verwendet.	
	Datentyp:	INT
Rückgabewert		
state	Erfolgsstatus des Befehls	
	Datentyp:	INT
	0	Erfolgreiche Ausführung des Befehls. Kommando an PLC erfolgreich beendet. (PLC Quittierung kann aber noch ausstehen).
	-1	Befehl kann nicht benutzt werden, weil WZMG und/oder Funktion Multitool nicht aktiv ist.
	-2	Funktion wird nicht ausgeführt wegen Satzsuchlauf, Programmtest.
	-3	MT kann nicht positioniert werden, weil eine WZ-Korrektur bzgl. des programmierten WZ-Halters (noch) aktiv ist.
	-4	MT kann nicht positioniert werden, weil programmierter WZ-Halter kein MT enthält, sondern ein WZ.
	-5	MT kann nicht positioniert werden, weil der programmierte WZ-Halter weder ein MT noch ein WZ enthält.
	-6	MTlocno hat ungültigen Wert.
	-7	THno hat ungültigen Wert.

Beispiel 1

Das Werkzeug mit dem Namen "WZ1" / Duplonummer = 5 bzw. der T-Nummer = 33 ist im Multitool 555 auf MT-Platz = 2 bestückt. Das MT ist mit 6 Plätzen erzeugt worden.

Programmcode	**Kommentar**
`DEF INT state = 0`	`;`
`T="WZ1" M06 D1`	
	`;PLC wechselt das MT 555 auf den WZ-Halter 1 und positioniert auf MT-Platz 2`
	`;PLC quittiert den Wechsel, Achsbewegungen werden programmiert …`

Programmcode	Kommentar
D0	;WZ-Korrektur abwählen
POSMT(state, 5, 1)	
	;positioniere das MT auf den Platz 5
	;das bisherige aktive WZ mit der T-Nummer 33 ist immer noch aktiv (obwohl es wegen des POSMT-Befehls nicht mehr in der Bearbeitungsposition ist)
M17	

Weiterhin ist konfiguriert "aktiviere WZ des WZ-Halters und mache dessen Korrektur D1 aktiv".

Mit RESET (Programmende) wird nun ein WZ-Wechselkommando für PLC erzeugt, das folgende Anfangsdaten hat:

- bringe Multitool von WZ-Halter 1 von WZ-Halter nach WZ-Halter 1 (d.h. das MT sitzt schon auf dem WZ-Halter)
- positioniere das Multitool auf Platz 5

In NC wird nach dem Eingang der PLC-Ende-Quittierung veranlasst:

- aktiviere das WZ mit der T-Nummer 34 und wähle dessen Korrektur D1 an

Das Multitool bleibt auf dem WZ-Halter sitzen und es ist kein Werkzeug und keine D-Korrektur aktiv.

Beispiel (2)

Wenn in Beispiel 1 `POSMT` vor Programmende nicht programmiert wird, sondern nur die MT-Position in NC mit $TC_MTP8 auf den Wert 5 gesetzt wird, würde das im Initsatz erzeugte Kommando an PLC diese MT-Position vorgeben und PLC müsste das MT gemäß der Positionsvorgabe positionieren. PLC darf bei der Quittierung die von NC im Initsatz vorgegebene MT-Position nicht abändern.

11.10.24 MVTOOL - Sprachbefehl zum Bewegen eines Werkzeugs

Die Funktion `MVTOOL` ist für WZMG verfügbar.

Die Funktion MVTOOL be- und entlädt Werkzeuge allein durch NC-Programmierung. Ebenso kann damit ein Werkzeug von einem beliebigen Magazinplatz zu einem beliebig anderen Magazinplatz transportiert werden.

Es muss zwingend ein Werkzeug auf dem Ausgangsmagazinplatz sitzen.

Der Sprachbefehl erzeugt keinen Alarm.

Ob MVTOOL korrekt ausgeführt wurde oder ob ein Fehler auftrat, muss über den Rückgabewert des Parameters "state" geprüft werden.

Der Befehl ist auch auf Multitools anwendbar.

Syntax

```
MVTOOL (state, magFrom, locFrom, magTo, locTo)
```

Bedeutung

MVTOOL	Be- und entlädt Werkzeuge	
	Datentyp:	-
magFrom	Magazinnummer des Magazins, in dem sich das zu bewegende Werkzeug befindet	
	Datentyp:	INT
locFrom	Platznummer, auf dem sich das zu bewegende Werkzeug befindet	
	Datentyp:	INT
magTo	Zielmagazinnummer nach der das Werkzeug bewegt werden soll. Das kann ein Beladem agazin, Zwischenspeichermagazin, oder ein anderes reales Magazin sein. Der Parameter ist optional. Wird magTo nicht programmiert, wird der Wert von magFrom verwendet.	
	Datentyp:	INT
locTo	Zielplatznummer nach der das Werkzeug bewegt werden soll. Der Parameter ist optional. Wird locTo nicht programmiert bzw. wenn der Wert=0 programmiert wird, wird im Magazin magTo - falls dieses ein reales Magazin ist - eine Leerplatzsuche durchgeführt. Im internen Magazin ist keine Leerplatzsuche möglich. Ist magTo die Nummer eines internen Magazins und locTo ist nicht programmiert bzw. mit dem Wert=0 programmiert, dann wird der Status=-8 gesetzt.	
	Datentyp:	INT
state	Erfolgsstatus des Befehls	
	Datentyp:	INT
	0	erfolgreiche Ausführung (PLC-Quittierung kann aber noch ausstehen)
	-1	Befehl kann nicht benutzt werden, weil WZMG nicht aktiv ist
	-2	Funktion wird nicht ausgeführt wegen Satzsuchlauf, Programmtest
	-3	WZ kann nicht bewegt werden (weil z. B. WZ-Zustand " sich im Wechsel" gesetzt ist)
	-4	kein Werkzeug auf dem Ausgangsplatz
	-5	magFrom hat ungültigen Wert
	-6	locFrom hat ungültigen Wert
	-7	magTo hat ungültigen Wert
	-8	locTo hat ungültigen Wert
	-9	keine Distanzbeziehung definiert (falls genau ein Magazin ein internes Magazin ist)
	-10	kein Leerplatz gefunden (falls Parameter locTo nicht programmiert ist)
	-11	Zielplatz ist für das Werkzeug nicht frei
	-12	der Parameter locTo muss programmiert werden, da magTo ein internes Magazin ist

Hinweis

Der Befehl MVTOOL reserviert grundsätzlich den Zielplatz mit "reserviert für zu beladenes WZ".

Der Befehl setzt den WZ-Zustand des zu bewegenden WZs auf "WZ befindet sich im Wechsel", Bitwert "H20" von der Befehlsaufbereitung bis zur Befehlsbeendigung (Quittung mit Status 1 oder 3).

Befehlsbeendigung meint die erfolgreiche Beendigung mit PLC-Status = 1 bzw. 10, oder den Abbruch des Befehls mit PLC-Status = 3.

Wird ein Werkzeug von einem realen Magazin in ein internes Magazin (oder umgekehrt) bewegt, muss eine entsprechende Magazindistanzbeziehung definiert sein.

Falls die programmierten Parameter ungültig sind oder auf dem Ausgangsmagazinplatz kein Werkzeug sitzt, oder falls der programmierte Zielplatz durch das Werkzeug nicht belegbar ist, oder falls bei nicht programmiertem Parameter locTo die Leerplatzsuche keinen Platz gefunden hat, oder falls eine benötigte Magazindistanzbeziehung nicht definiert ist, werden über state entsprechende Fehlercodes zurückgegeben.

Eine Abwahl einer aktiven WZ-Korrektur ist mit dem Befehl nicht verbunden.

PLC

Der Sprachbefehl erzeugt in NC das Kommando für PLC, gibt das Kommando im Rahmen der Abarbeitung des aktiven Satzes an PLC aus. Der aktive Satz gilt erst als abgearbeitet, wenn die Ende-Quittierung des Kommandos durch PLC vorliegt. Das kann der PLC-Status = 1, 3 oder 5 sein. Erst danach kann ein neuer Satz zur Abarbeitung eingewechselt werden.

Beispiele

Es besteht folgende Konfiguration:

- Magazin (Magazinnr. = 5, Plätze 1, ...10),
- eine Spindel (Zwischenspeichermagazinnr. = 9998, Platz 1),
- eine Beladestelle (Beladem agazinnr. = 9999, Platz 1)

Das Magazin ist per Distanzbeziehung (siehe $TC_MDP1/$TC_MDP2) mit der Spindel und der Beladestelle verbunden.

Beispiel 1

Das Werkzeug vom Beladeplatz 9999/1 soll in das Magazin 5 beladen werden. Dazu wird Folgendes programmiert:

Programmcode	Kommentar
`def int state`	
`$TC_MPP6[9999, 1] =123`	`;setze WZ mit interner T-Nr.=123 auf Beladeplatz`
`MVTOOL(state, 9999, 1, 5)`	`;ein passender Leerplatz wird in Magazin 5 gesucht`
`;Falls das Werkzeug genau auf Platz 7 beladen werden soll:`	

Programmcode	Kommentar
MVTOOL(state, 9999, 1, 5, 7)	; Platz 7 wird vor dem Belegen leerplatz-geprüft

Beispiel 2

Das Werkzeug mit der T-Nr.=123 soll vom Beladeplatz auf Spindel 1 (Magazin-Nr.=9998, Platz-Nr.=1) beladen werden:

Programmcode	Kommentar
$TC_MPP6[9999, 1] =123	
MVTOOL(state, 9999, 1, 9998, 1)	

Beispiel 3

Bei gleicher Konfiguration soll ein auf Platz 7 beladenes Werkzeug auf einen anderen passenden Platz im selben Magazin gebracht werden.

Programmcode	Kommentar
MVTOOL(state, 5, 7, 5)	
;oder auf Platz 3 im selben Magazin	
MVTOOL(state, 5, 7, 5, 3)	
;auf irgendeinem Platz im Magazin 11	
MVTOOL(state, 5, 7, 11)	

11.10.25 SETTA - Werkzeug aus Verschleißverbund aktiv setzen

Die Prozedur SETTA setzt alle nicht gesperrten Werkzeuge aus dem gewünschten Verschleißverbund aktiv. Aus einer Werkzeuggruppe wird dabei jedoch maximal ein Werkzeug, bezogen auf eine Spindel bzw. WZ-Halter, aktiv.

Sind zum Aufrufzeitpunkt keine Verschleißverbünde definiert, bzw. ist die Funktion "Verschleißverbund" über Maschinendatum nicht aktiviert, so kann der Befehl trotzdem benutzt werden. Die Aufruf-Parameter müssen dann entsprechend gesetzt werden.

Treffen die Auswahlkriterien von SETTA auf mehrere Werkzeuge innerhalb einer Werkzeuggruppe zu, kommt das Auswahlkriterium $TC_MAMP3, Bit 12-13 zum Tragen. Wenn es vor dem Befehl schon ein aktives WZ gibt, das die Kriterien von SETTA erfüllt, wird aus der Werkzeuggruppe kein weiteres WZ (bezogen auf die gleiche Spindel) aktiv gesetzt.

Wird SETTA nicht parametriert, bezieht sich das Aktivsetzen auf alle beladenen, frei gegebenen und nicht gesperrten Werkzeuge im TO-Bereich.

Syntax

```
SETTA (STATUS, MNR, VNR, USEKT)
```

Bedeutung

SETTA	Setzt alle nicht gesperrten Werkzeuge aus dem Verschleißverbund aktiv	
	Datentyp:	-
MNR	Magazinnummer (optional)	
	Datentyp:	INT
	0	bzw. Nichtangabe des Parameters bedeutet, dass sich das Aktivsetzen auf alle Magazine bezieht - unabhängig davon, ob eine Distanzbeziehung zu einem Spindelplatz besteht.
	-1	Alle Magazine mit einer Distanzbeziehung zu einer Spindel bzw. einem Toolholder.
	-2	Jeder Spindel bzw. jedem Toolholder wird in seinem Magazin ein aktives WZ aus jeder im Magazin vertretenen WZ-Gruppe gesetzt. (entspricht dem mehrfachen Aufruf von SETTA mit den konkreten beteiligten Magazinnummern). Damit kann es z. B. 2 Werkzeughalter geben, denen je 1 Magazin zugeordnet ist. Die Werkzeuge einer Gruppe können beliebig auf die beiden Magazine verteilt werden. Sind Werkzeuge einer Gruppe auf beide Magazine enthalten, so führt das dazu, dass aus der WZ-Gruppe zwei Werkzeuge den Zustand aktiv erhalten.
Werden mehrere Magazine angesprochen (MNR ≤ 0), werden die Magazine als Einheit betrachtet und es wird ein Werkzeug aus der Werkzeuggruppe über alle betrachteten Magazine aktiv gesetzt. Gesperrte Magazine ($TC_MAP3, Bit 1=1) werden dabei nicht betrachtet.		
VNR	Verschleißverbund Nummer (optional)	
	Datentyp:	INT
	0	Das Aktivsetzen bezieht sich auf die Magzinplätze, die keinem Verschleißverbund zugeordnet sind.
	-1	Aktiver Verschleißverbund ($TC_MAP9). Gibt es in keinem der genannten Magazine einen aktiven Verschleißverbund, dann wird der Status -1 zurückgegeben und keine Zustandsänderung der Werkzeuge durchgeführt.
	-2	bzw. Nichtangabe des Parameters bedeutet, dass sich das Aktivsetzen auf das ganze Magazin erstreckt. Insbesondere wird der Parameter VNR nicht angegeben bzw. = -2 gesetzt, wenn nicht mit der Funktion "Verschleißverbund" gearbeitet wird.
USEKT	Werkzeuguntergruppe (optional)	
	Datentyp:	INT
	0	bzw. Nichtangabe des Parameters bedeutet, alle Werkzeuge der Gruppe werden betrachtet.
	-1	Der aktuell programmierte Wert von $P_USEKT wird verwendet.

STATUS	Rückgabeparameter (optional)	
	Datentyp:	REF INT
	0	Prozedur wurde korrekt ausgeführt.
	1	Prozedur wurde durchgeführt, es gibt aber noch ein anderes aktives Werkzeug in einer Gruppe, z. B. ein entladenes Werkzeug.
	-1	Prozedur wurde nicht ausgeführt, da kein aktiver Verschleißverbund in den ausgewählten Magazinen vorhanden ist.
	-2	Prozedur wurde nicht ausgeführt, da die programmierte Verschleißverbundnummer nicht existiert.
	-3	Prozedur wurde nicht ausgeführt, da die programmierte Magazinnummer nicht existiert.
	-4	Prozedur wurde nicht ausgeführt, da die Funktion "Verschleißbund" nicht freigegeben ist (MN_TOOL_MANAGEMENT_MASK).
	-5	Prozedur wurde aus anderen Gründen nicht ausgeführt.

Für das durch SETTA zu aktivierende Werkzeug kann eine Suchstrategie im Parameter $TC_MAMP3 eingestellt werden.

Bit 12 = 0	kleinste mögliche Duplonummer (default)
Bit 12 = 1	kleinster möglicher Magazinplatz
Bit 13 = 1	kleinste mögliche im Parameter $TC_TP10 (Einsatzreihenfolge) enthaltende Nummer

Hinweis

Für die Prozedur SETTA muss zwingend der Verschleißverbund eingestellt sein.

Multitool

Wird SETTA für ein Multitool programmiert statt für ein Magazin, wird folgender Alarm erzeugt: Alarm 6462 "[Kanal %1:] Satz %2 Befehl %' kann nur für Magazine programmiert werden. '%4' bezeichnet kein Magazin.

Siehe auch

SETTIA - Werkzeug aus Verschleißverbund inaktiv setzen (Seite 199)

11.10.26 SETTIA - Werkzeug aus Verschleißverbund inaktiv setzen

Die Prozedur SETTIA setzt in allen Werkzeugen den Status "Aktiv" zurück. Durch die Parametrierung des Sprachbefehls kann das Magazin- oder Verschleißverbundspezifisch erfolgen.

Hinweis

Für die Prozedur SETTIA muss zwingend der Verschleißverbund eingestellt sein.

Syntax

```
SETTIA(STATUS, MNR, VNR,USEKT)
```

Bedeutung

SETTIA:	Setzt alle nicht gesperrten Werkzeuge aus dem Verschleißverbund inaktiv	
	Datentyp:	INT
MNR	Magazinnummer (optional)	
	Datentyp:	INT
	0	Das Inaktivsetzen bezieht sich auf alle Magazine, ungeachtet einer Zuordnung zu einer Spindel. In diesem Fall werden auch die Werkzeuge im Zwischenspeicher einschließlich Toolholder betrachtet.
	> 0	Magazinnummer, in der das Inaktivsetzen erfolgen soll. Werkzeuge dieses Magazins, die im Zwischenspeicher sind, werden nicht betrachtet. D. h., werden diese Werkzeuge in das Magazin zurückgewechselt, haben sie nach wie vor ihren "aktiv" Status.
	-1	Alle Magazine mit einer Distanzbeziehung zu einer Spindel bzw. einem Toolholder.
VNR	Verschleißverbund Nummer (optional)	
	0	Das Inaktivsetzen bezieht sich auf alle Werkzeuge, die keinem Verschleißverbund zugeordnet sind. Ist kein Verschleißverbund definiert, bezieht sich das Inaktivsetzen auf alle Werkzeuge im Magazin.
	> 0	Verschleißverbund Nummer, in dem das Aktivsetzen erfolgen soll.
	-1	Aktiver Verschleißverbund ($TC_MAP9).
USEKT	Werkzeuguntergruppe (optional)	
	0	Alle Werkzeuge der Gruppe werden betrachtet.
	> 0	Es werden die Werkzeuge betrachtet, bei denen im Parameter $TC_TP11 ein Bit des unter USEKT programmierten Wertes sitzt.
	-1	Der aktuell programmierte Wert von USEKT wird verwendet.
STATUS	Rückgabeparameter (optional)	
	Datentyp:	REF INT
	0	Prozedur konnte korrekt ausgeführt werden.
	-1	Prozedur wurde nicht ausgeführt, da kein aktiver Verschleißverbund in den ausgewählten Magazinen vorhanden ist.
	-2	Prozedur wurde nicht ausgeführt, da die programmierte Verschleißverbund-Nummer nicht existiert.
	-3	Prozedur wurde nicht ausgeführt, da die programmierte Magazinnummer nicht existiert.
	-4	Prozedur wurde nicht ausgeführt, da die Funktion "Verschleißbund" nicht freigegeben ist (MN_TOOL_MANAGEMENT_MASK).
	-5	Prozedur wurde nicht ausgeführt.

Für das durch SETTIA zu aktivierende Werkzeug kann im Parameter $TC_MAMP3 eine Suchstrategie eingestellt werden.

Multitool

Wird SETTIA für ein Multitool programmiert statt für ein Magazin, wird folgender Alarm erzeugt: Alarm 6462 "[Kanal %1:] Satz %2 Befehl %3' kann nur für Magazine programmiert werden. '%4' bezeichnet kein Magazin.

Siehe auch

SETTA - Werkzeug aus Verschleißverbund aktiv setzen (Seite 197)

11.10.27 RESETMON - Sprachbefehl zur Sollwertaktivierung

Die Prozedur RESETMON setzt den Istwert des Werkzeugs auf den Sollwert.

Es werden die ursprünglichen Maximalwerte, wie z. B. Sollstandzeit und Sollstückzahl, reaktiviert und Werkzeug-Zustände wie "Werkzeug gesperrt" (Vorwarngrenze erreicht) zurückgesetzt.

Der Befehl wird für Werkzeuge sowie für Multitools verwendet.

Syntax

```
RESETMON (state, t, d, mon, resetStates)
```

Bedeutung

RESETMON	Setzt den Istwert des Werkzeugs auf den Sollwert	
	Datentyp:	...
state	Rückgabeparameter, der folgende Werte annehmen kann:	
	Datentyp:	REF INT
	0	Befehl erfolgreich ausgeführt
	-1	Die Schneide mit der genannten D-Nummer d existiert nicht.
	-2	Das Werkzeug mit der genannten T-Nummer t existiert nicht.
	-3	Das genannte Werkzeug hat keine definierte Überwachungsfunktion. Dieser Status ist nur möglich, wenn t explizit benannt wurde.
	-4	Überwachungsfunktion ist im NC nicht aktiv geschaltet, d. h., der Befehl kam nicht zur Ausführung.
t	Interne T-Nummer	
	Datentyp:	INT
	t = 0	Es werden alle Werkzeuge behandelt.
	t > 0	Es wird genau dieses Werkzeug behandelt.
	t < 0	Es wird der absolute Betrag von t gebildet und alle Schwester-Werkzeuge dieses Werkzeuges behandelt.

`d`	Die D-Nummer des Werkzeuges (optional) Wenn der Parameter nicht angegeben oder mit 0 belegt wird, werden alle D-Nummern bzw. alle Schneiden des Werkzeuges behandelt.	
	Datentyp:	INT
	d > 0	Der Befehl bezieht sich genau auf die genannte D-Nummer.
`mon`	Bitcodierter Parameter (optional) Wenn der Parameter nicht angegeben oder mit 0 belegt wird, werden alle Istwerte der für das Werkzeug aktiven Überwachungen der bezeichneten Schneide (n) auf die Sollwerte gesetzt.	
	mon > 0	Es wird genau der Istwert der genannten Überwachungsart behandelt. Mögliche Werte sind die positiven Werte des Systemparameters **$TC_TP9** (1, 2, 4, 8) bzw. die entsprechenden Bitkombinationen bei Aktivierung mehrerer Überwachungsarten.
	mon < 0	Es wird genau der Istwert im "Betrag von mon" genannten Überwachungsart behandelt. Es erfolgt keine Einschränkung durch die Werte der Systemvariablen $TC_TP9. Damit können auch Werte nicht aktivierter Überwachungsarten zurückgesetzt werden. Insbesondere ist auch das gleichzeitige Zurücksetzen der Istwerte von Verschleiß- und Summenkorrektur-Überwachungswerten möglich.
`resetStates`	Bitcodierter Parameter (optional)	
	Bit 0	WZ-Status "aktiv" wird gelöscht
	Bit 1	WZ-Status "freigegeben" wird gesetzt
	Bit 2	WZ-Status "gesperrt" wird rückgesetzt, wenn • a) die Überwachungsdaten das zulassen • b) der Parameter "mon" entsprechend gesetzt ist
	Bit 3	WZ-Status "vermessen" wird gesetzt
	Bit 4	WZ-Status "Vorwarngrenze" wird rückgesetzt, wenn • c) die Überwachungsdaten das zulassen • d) der Parameter "mon" entsprechend gesetzt ist
	Bit 5	nicht erlaubt
	Bit 6	nicht erlaubt
	Bit 7	WZ-Status "war im Einsatz" wird gelöscht
	Bit 8	nicht erlaubt
	Bit 9	nicht erlaubt
	Bit 10	WZ-Status "zu entladen" wird gelöscht
	Bit 11	nicht erlaubt
	Bit 12	nicht erlaubt

Mit dem Parameter `resetStates` kann zusätzlich zu den Überwachungsparametern auch gezielt der WZ-Status verändert werden. Die Bitcodierung von `resetStates` entspricht dem Parameter des WZ-Status $TC_TP8[x].

Wird dieser Parameter nicht angegeben, wird auf das Maschinendatum $MN_TOOL_RESETMON_MASK zugegriffen. Die Bitcodierung dieses Datums ist identisch mit der des Parameters "resetStates". Beim analogen PI-Dienst, PI_TRESMO wirkt ebenfalls das Maschinendatum.

RESETMON für Multitools

Multitool

Für ein Multitool werden die Parameter von RESETMON erweitert.

Syntax für Multitools

```
RESETMON (state, MTno, Dno, mon, resetStates)
```

Bedeutung der Parameter bei Multitools

`state`	Rückgabeparameter, der folgende Werte annehmen kann:		
	Datentyp:		REF INT
	0		Befehl erfolgreich ausgeführt
	-1		Die Schneide mit der genannten Nummer `Dno` existiert nicht
	-2		Das Multitool mit der genannten Nummer `MTno` existiert nicht
`MTno`	Interne T-Nummer		
	Datentyp:		INT
	t = 0		Es werden alle Werkzeuge behandelt.
	t > 0		Es wird genau dieses Werkzeug behandelt.
	t < 0		Es wird der absolute Betrag von t gebildet und alle Schwester-Werkzeuge dieses Werkzeuges behandelt.
Dno	Die D-Nummer des Werkzeuges (optional) Wenn der Parameter nicht angegeben oder mit 0 belegt wird, werden alle D-Nummern bzw. alle Schneiden des Werkzeuges behandelt.		
	Datentyp:		INT
	Dno = 0		Ist der Parameter nicht angegeben oder wird das Werkzeug mit 0 programmiert, werden alle Schneiden der Werkzeuge im Multitool behandelt.
	Dno > 0		Der Befehl bezieht sich genau auf die Schneiden der Werkzeuge im Multitool mit der genannten D-Nummer.
`mon`	mon bezieht auf die Werkzeuge im Multitool - siehe oben		
	Datentyp:		INT
`resetStates`	`resetStates` bezieht sich zum einen auf Zustände des Einzelwerkzeugs, die mit dem Rücksetzen auf die Sollwerte zusätzlich geändert werden; aber hier für diese spezielle Programmierung auch zusätzlich auf die entsprechenden MT-Zustände, die geändert werden, wenn mindestens der gleiche Zustand in einem der Werkzeuge des Multitoolss geändert wurde.		
	Bit	Wert	Bedeutung

	1 =	0	Multitool-Status "freigegeben" bleibt unverändert
		1	Multitool-Status "freigegeben" wird gesetzt
	2 =	0	Multitool-Status "gesperrt" bleibt unverändert
		1	Multitool-Status "gesperrt" wird gelöscht, wenn Überwachungsdaten dies zulassen und der 4. Parameter entsprechend gesetzt ist

Hinweis

Wenn der optionale Parameter `resetstates` nicht programmiert wird, dann gilt als impliziter Wert der Wert des MD $MN_TOOL_RESETMON_MASK.

Für den Parameter T-Nummer des korrespondierenden PI-Dienst _N_TRESMO kann analog die MT-Nummer programmiert werden.

Das zur Parametrierung des Befehls RESETMON verfügbare MD $MN_TOOL_RESETMON_MASK, mit der Beschreibung "Verhalten der Werkzeugdaten bei RESETMON" behält seine Wirkung auf die Werkzeuge - auch wenn sie in einem Multitool sind und mit RESETMON zurückgesetzt werden.

Die wählbaren Zustandsänderungen wirken sich auch auf das Multitool aus, falls dieses programmiert worden ist und den entsprechenden Zustand definiert hat.

		WZ	MT
Bit 0	Status "aktiv" löschen	X	-
Bit 1	Status "freigegeben" setzen	X	X
Bit 2	Status "gesperrt" bedingt löschen, wenn Überwachungsdaten dies zulassen	X	X
Bit 3	Status "vermessen" setzen	X	-
Bit 4	Status "Vorwarngrenze" bedingt löschen, wenn Überwachungsdaten dies zulassen	X	-
Bit 7	Status "war im Einsatz" löschen	X	-
Bit 10	Status "zu entladen" löschen	x	x

Beispiel Multitool (1)

Der Zeitüberwachungsistwert (4. Parameter = 1) der Schneiden mit der D-Nummer 2 (3. Parameter) aller Werkzeuge, die im MT mit der Nummer 500 bestückt sind, wird auf den Sollwert zurückgesetzt. Dabei wird den Zustandswert der Werkzeuge, deren Zeitüberwachungswert geändert worden ist, gemäß den Angaben des letzten Parameters geändert.

Da 500 die Nummer eines Multitools ist, wird zusätzlich der Zustand dieses Multitools gemäß den Angaben des letzten Parameters geändert. Dabei ist zu beachten, dass nur die programmierten Bits 1, 2,10 für die MT-Zustandswertänderung Bedeutung haben.

Über den Referenzparameter "state" wird der Erfolg des Befehls mitgeteilt.

```
int state=0

RESETMON(state, 500, 2, 1, "H49F")
```

Das MT 500 aus dem Beispiel ist mit 3 Werkzeugen bestückt. Jedes WZ hat zwei Schneiden mit D-Nr.=1 und 2. Zwei sind zeitüberwacht und wegen des Zeitlimits in den Schneiden mit der D-

Nummer = 2 gesperrt worden, das dritte Werkzeug ist stückzahlüberwacht und ist ebenfalls gesperrt, weil in der Schneide mit der D-Nummer = 2 das Stückzahllimit erreicht worden ist. Nun werden mit RESETMON die beiden zeitüberwachten Werkzeuge rückgesetzt, der WZ-Zustand "gesperrt" aufgehoben und der MT-Zustand "gesperrt" ebenfalls aufgehoben (obwohl das stückzahlüberwachte WZ weiterhin gesperrt ist). Die anderen programmierten WZ- und MT-Zustände außer "gesperrt" werden der Definition gemäß geändert.

Hätte man programmiert:

```
RESETMON(state, 500, 2, 0, "H49F")
```

wären alle Überwachungsistwerte aller im MT bestückten Werkzeuge auf ihre Sollwerte gesetzt worden (4. Parameter = 0) und alle Werkzeuge hätten den Zustand "gesperrt" rückgesetzt und das MT hätte seinen Zustand "gesperrt" rückgesetzt. Die weiteren programmierten Zustände wären auch geändert worden.

Beispiel Multitool (2)

Die Istwerte der Zeitüberwachung aller Schneiden und aller Werkzeuge werden auf die Sollwerte der Zeitüberwachung gesetzt.

Der Zustand der Werkzeuge werden gemäß der Programmierung (letzter Parameter = "H405"; setze Zustand "gesperrt" bedingt zurück und setzte Zustände "aktiv" und "zu entladen" unbedingt zurück) geändert. Geändert wird auch den Zustand von Multitools gemäß der Programmierung (den MT-Zustand "gesperrt" bedingt, den Zustand "zu entladen" unbedingt).

```
int state=0
RESETMON(state, 0, 0, 1, "H405")
```

11.10.28 DELTC - Lösche Werkzeug-Trägerdatensatz

Die Funktion "Orientierbarer Werkzeugträger" muss aktiv sein. Die Funktion kann den Funktionen WZMO und WZMG additiv überlagert werden.

Die Funktion `DELTC` löscht einen Werkzeug-Trägerdatensatz.

Syntax

```
DELTC(n,m)
```

Bedeutung

`DELTC`	Löscht einen Werkzeug-Trägerdatensatz	
	Datentyp:	-
`n`	Erste Nummer des Werkzeug-Trägerdaten-Bereichs, dessen Werte auf Null gesetzt werden sollen (opttional). Wenn `n` nicht angegeben wird, werden alle Werkzeug-Trägerdatensätze, beginnend von der kleinsten bis zur größten Nummer auf Null gesetzt	
	Datentyp:	INT

m	Letzte Nummer des Werkzeug-Trägerdaten-Bereichs, dessen Werte auf Null gesetzt werden sollen (optional). Wenn m nicht angegeben wird, dann wird genau der durch n genannte Werkzeug-Trägerdatensatz auf Null gesetzt. Ist m größer als die größte Nummer eines Werkzeug-Trägerdatensatzes in diesem Kanal, so werden die Datensätze bis zur größten Nummer auf Null gesetzt.	
	Datentyp:	INT

Die Werkzeug-Trägerdatensätze werden durch die Systemvariablen $TC_CARRx definiert. Zum Nullsetzen aller Datensätze gibt es die Systemvariable $TC_CARR1[0]. Mit DELTC kann nun ein Werkzeug-Trägerdaten-Nummernbereich von n bis m auf Null gesetzt werden.

Insbesondere ist DELTC inhaltsgleich mit $TC_CARR1[0]=0= setze alle Datensätze auf Null.

Die Parameter n, m müssen mit Werten größer als Null programmiert werden. Andere Werte führen zu einem Alarm.

Der Parameter n muss kleiner als m sein. Andere Werte führen zu einem Alarm.

Außerdem muss n im Bereich der erlaubten Werkzeug-Trägerdatennummern liegen.

Der gewählte Nummernbereich muss den Nummernbereich der auf dem Kanal vorhandenen Werkzeug-Trägerdatensätze enthalten. Anderenfalls wird die Programmierung mit Alarm abgelehnt.

Ist die Funktion "Werkzeug-Trägerdaten" nicht aktiviert ($MN_MM_NUM_TOOL_CARRIER 0 0) erzeugt DELTC ebenfalls einen Alarm.

Beispiel

In der TO-Einheit sind 14 Werkzeug-Trägersätze mit den Nummern 1 bis 14 definiert.

Programmcode	**Kommentar**
DELTC(5,8)	;setze die Werte der Datensätze 5, 6, 7, 8 auf Null
DELTC(5,20)	;setze die Werte der Datensätze 5, 6, 7, ...,14 auf Null
DELTC(9)	;setze die Werte des Datensatzes 9 auf Null
DELTC()	;setze die Werte der Datensätze 1, ..., 14 auf Null
DELTC(0,1)	;Fehler → Alarm - n, m müssen größer als Null sein
DELTC(0,-2)	;Fehler → Alarm - n, m müssen größer als Null sein
DELTC(0)	;Fehler → Alarm - n muss größer als Null sein
DELTC(15,20)	;Fehler → Alarm - n darf maximal 14 sein
DELTC(20)	;Fehler → Alarm - n darf maximal 14 sein

11.10.29 TCA - Werkzeug-Anwahl/Werzeugwechsel unabhängig vom Status des Werkzeugs

Die Funktion TCA ist für WZMO und WZMG verfügbar.

Die Funktion TCA wechselt ein Werkzeug unabhängig von seinem "gesperrt" oder "zu entladen" Status ein.

Für gewisse Abläufe (z. B. Messzyklen) ist es nötig, ein bestimmtes Werkzeug - unabhängig von seinem Status (z. B. ein gesperrtes Werkzeug) auf die Spindel bzw. den Toolholder einzuwechseln.

Syntax

```
TCA("WZ-Name", Duplonr., Toolholder-Nr.)
```

Bedeutung

`TCA`	Wechselt ein Werkzeug unabhängig von seinem Status ein.	
	Datentyp:	-
`WZ-Name`	Bezeichner des einzuwechselnden Werkzeugs	
	Datentyp:	STRING
`Duplonr.`	Duplonummer des einzuwechselnden Werkzeugs (optional). Wird `Duplonr.` nicht angegeben, wird das Werkzeug mit der kleinsten Duplonummer eingewechselt.	
	Datentyp:	INT
`Toolholder-Nr.`	Werkzeughalter bzw. Spindel, auf die der Wechsel erfolgen soll (optional). Wird `Toolholder-Nr.` nicht angegeben, bezieht sich der Wechsel auf die aktuell eingestellte oder programmierte Masterspindel bzw. den Mastertoolholder. Für WZMO ohne WZMG gilt: Parameter entspricht der Adressextension des T-Befehls. (Die Einstellung des MD $MC_T_M_ADDRERSS_EXT_IS_SPINO ist zu berücksichtigen.)	
	Datentyp:	INT

TCA verhält sich bezüglich der Alarm- und Kommandoausgabe analog zum T-Befehl.

Ist weder WZMG noch WZMO aktiv, wird ein Alarm erzeugt.

Auftretende Alarme bei der Programmierung sind analog zu den Alarmen bei T-Programmierung zu behandeln.

Hinweis

Die Korrekturanwahl, entsprechend $MC_CUTTING_EDGE_DEFAULT wirkt analog zum T-Befehl. TCA und D dürfen nicht in einem Satz programmiert werden.

Beispiele

1. Vorbereitung und Wechsel mit T-Befehl (d. h. $MC_TOOL_CHANGE_MODE=0)
   Konstellation 1x Revolver, 1x Toolholder
   Es gibt 2 Werkzeuge mit dem Bezeichner "Schlichter" und den Duplonummern 1 und 2.
   ```
   TCA("Schlichter", 1,1)
   ```
   Das Werkzeug "Schlichter" mit der Duplonummer 1 wird auf den Toolholder 1 eingewechselt.
   Bei der angenommenen Maschinenkonstellation führt folgende Programmierung zum selben Ergebnis:
   ```
   TCA("Schlichter")
   ```
   Die Duplonummer ist nicht angegeben, das bedeutet, dass das Werkzeug mit der kleinsten Duplonummer gewechselt wird, und das ist Duplo "1".
   Die Toolholder-Nr. ist nicht angegeben. Daher bezieht sich der Wechsel auf den aktuellen Mastertoolholder und das ist "1".

2. Wechsel mit M06 ($MC_TOOL_CHANGE_MODE=1)
   Konstellation: 1x Kettenmagazin, 2x Spindel, Spindel_1 ist die Masterspindel. Es gibt 4 beladene Werkzeuge, "FRAESER_20MM", mit den Duplonummern 4, 5, 8 und 15.
   "FRAESER_20MM", Duplo "8" wurde gesperrt und muss vermessen werden. Das Messen geschieht auf der Spindel 2.
   ```
   TCA("FRAESER_20MM",8,2)
   M2=6
   ```
   Das Werkzeug "FRAESER_20MM", Duplo "8" wird für die Spindel "2" vorbereitet und gewechselt.
   Hier würde nun folgende Programmierung zu einem anderen Ergebnis führen:
   ```
   TCA("FRAESER_20MM")
   M06
   ```
   Vorbereitet wird das Werkzeug "FRAESER_20MM", Duplo "4" (kleinste Duplonummer) für die Spindel "1" (das ist die Masterspindel) und mit M06 gewechselt.

Hinweis

Gegenüber dem T-Befehl gelten für TCA folgende Besonderheiten:

- TCA und D können nicht in einem Satz programmiert werden.
- TCA setzt die eingestellten Suchstrategien ($TC_MAMP2 und/oder $TC_MAP10) außer Kraft und ignoriert die programmierten Werte von $P_USEKT.
- Das Werkzeug muss den Status "freigegeben" haben.
- Die PLC-Nahtstellensignale "Übergang auf neues Ersatz-WZ" und "letztes Ersatz-WZ der Gruppe" werden nicht gesetzt.

PLC

Die PLC darf ein mit "TCA" vorbereitetes Werkzeug nicht ablehnen. Eine Quittierung mit Status_2, Status_7 oder Status_103 ist nicht zulässig.

Achtung: Auf der Nahtstelle gibt es derzeit kein Kriterium, ob ein WZ abgelehnt werden darf oder nicht.

Wird mit dieser Funktion gearbeitet, so muss das durch eine Zusatzkennung der PLC mitgeteilt werden.

Programmcode	Kommentar
$TC_VDITCP[2]=101	("101" - Kennung, dass die PLC nicht ablehnen darf)

Programmcode	Kommentar
TCA("Fraeser",1)	

Definitionsergänzung falls das programmierte Werkzeug ein Multitool ist

Mit der Funktion TCA kann man ein bestimmtes Werkzeug mit "WZ-Name" / "Duplonummer" unabhängig vom WZ-Zustand auf den WZ-Halter wechseln. Ist dieses Werkzeug in einem Multitool, wird der Zustand des Multitools ebenso wie der Zustand des Werkzeugs für den Befehl ignoriert. Insbesondere kann damit ein Werkzeug aus einem nicht einsatzfähigen Multitool angewählt bzw. ausgewechselt werden.

Hinweis

Der Befehl kann nur für Werkzeuge - auch solche, die in einem Multitool enthalten sind - und nicht für ein Multitool programmiert werden. Ist der programmierte Name z. B. ein MT-Name, wird der Alarm 6460 "[Kanal %1:] Satz %2 Befehl %3 kann nur für Werkzeuge programmiert werden. %4 bezeichnet kein Werkzeug." erzeugt.

Hinweis

Ein Werkzeug von einem gesperrtem Magazin oder Magazinplatz kann auch mit TCA nicht in die Spindel bzw. den WZ-Halter eingewechselt werden.

11.10.30 TCI - Wechsle Werkzeug aus Zwischenspeicher in das Magazin

Die Funktion TCI ist für WZMG verfügbar.

Die Funktion TCI wechselt die Werkzeuge der Zwischenspeicherplätze in das Magazin zurück. Werkzeughalterplätze sind davon allerdings ausgenommen. Anwendungen ergeben sich in der Regel bei komplexen Maschinen mit mehreren Greifern.

Die nötige Leerplatzsuche erfolgt wie bei einem programmierten Werkzeugwechsel mit T.

Hinweis

TCI kann nicht zusammen mit M06 in einem NC-Satz programmiert werden. Werkzeugwechsel Vorbereitung und Ausführung werden in einem Befehlsschritt ausgeführt.

Der Befehl TCI kann nicht substituiert (T-Funktionsersetzung) werden.

TCI(locNo, WZ-Halternr.)

locNo	Nummer des Zwischenspeichers, dessen Werkzeug in das Magazin zurückgewechselt werden soll. Da die locNo nicht die Platznummer eines Werkzeughalters sein kann, hat das Rückwechseln keine Auswirkung auf die aktive Werkzeugkorrektur.
WZ-Halternr.	Gibt die Nummer des Werkzeughalters an, von dem die Entsorgung des Werkzeugs erfolgen soll. Der Parameter ist optional. Wird er nicht angegeben, wird automatisch der aktuelle Master-Werkzeughalter gewählt.

Wird eine ungültige Platznummer programmiert, wird der Alarm 6403 erzeugt.

Syntax

```
TCI(locNo, WZ-Halternr.)
```

Bedeutung

TCI	Wechselt die Werkzeuge der Zwischenspeicherplätze in das Magazin zurück.	
	Datentyp:	-
locNo	Nummer des Zwischenspeichers, dessen Werkzeug in das Magazin zurückgewechselt werden soll. Da die locNo nicht die Platznummer eines Werkzeughalters sein kann, hat das Rückwechseln keine Auswirkung auf die aktive Werkzeugkorrektur.	
	Datentyp:	INT
WZ-Halternr.	Gibt die Nummer des Werkzeughalters an, von dem die Entsorgung des Werkzeugs erfolgen soll (optional). Wird Wz-Halternr. nicht angegeben, wird automatisch der aktuelle Master-Werkzeughalter gewählt.	
	Datentyp:	INT

Wird eine ungültige Platznummer programmiert, wird der Alarm 6403 erzeugt.

Die Platznummer locNo ist ungültig,

- wenn locNo auf einen Werkzeughalter / eine Spindel zeigt (Alarm 6450)
- wenn locNo auf einen nicht definierten Zwischenspeicherplatz zeigt (Alarm 6403)
- wenn locNo nicht mit dem programmierten Werkzeughalter bzw. dem Master-Werkzeughalter durch $TC_MLSR verbunden ist (Alarm 6454)
- wenn weder für den Zwischenspeicherplatz locNo noch für den Werkzeughalter / die Spindel eine Distanztabelle definiert ist (Alarm 6454)

Ist kein Zwischenspeichermagazin definiert, wird der Alarm 6451 erzeugt.

Ist der angegebene Werkzeughalter nicht definiert, wird der Alarm 6452 erzeugt.

Ist WZMG nicht aktiv, so wird der Alarm 6431 erzeugt.

Um TCI erfolgreich zu programmieren, ist es notwendig, dass die genannte Platznummer locNo über $TC_MLSR dem Werkzeughalter zugeordnet ist. Die Leerplatzsuche erfolgt in den Magazinen, die in der Distanztabelle (definiert durch $TC_MDP2) des Zwischenplatzspeichers locNo oder des Werkzeughalters definiert sind. Haben sowohl der Zwischenplatzspeicher locNo als auch der Werkzeughalter eine Distanztabelle, so wird die Distanztabelle des Zwischenplatzspeichers verwendet. Haben beide keine Distanztabelle, wird der Alarm 6454 erzeugt.

Hinweis

Der Befehl TCI enthält als Parameter die Platznummer eines Platzes (Greifer, Lader, Übergabeplatz) des Zwischenspeichermagazins. Um diesen Sprachbefehl in eigenen Zyklenprogrammen zu verwenden, kann die Platznummer über die Systemvariablen $P_MAGNREL, $P_MAGREL in Erfahrung gebracht werden.

Beispiele

Es besteht folgende Magazinkonfiguration:

- Magazin 1

Im Zwischenspeichermagazin mit 5 Plätzen ist definiert:

- Spindel 2 (Platz 1) mit Greifern 1 und 2 (Plätze 3 und 4 mit der Spindel über $TC_MLSR[3,1]=0 und $TC_MLSR[4,1]=0 verbunden)
- Spindel 1 (Platz 2) mit Greifer 3 (Platz 5 mit der Spindel über $TC_MLSR[5,2]=0 verbunden)

Programmiert wird:

Programmcode	Kommentar
`TCI(2)`	`;erzeugt Alarm 6450`
`TCI(5)`	`;wechselt das Werkzeug von Platz 5 (Greifer 3) zurück in das Magazin`
`TCI(9)`	`;Alarm 6403 (Zwischenspeicher hat nur die Nummer 1 bis 5)`

Der Anwender bestimmt durch seine Programmierung die Reihenfolge der Entsorgung der Zwischenspeicherplätze.

PLC

TCI wird in der PLC wie bei Programmierung von T0 M06 abgewickelt.

Multitool

Der Befehl TCI wechselt auch Multitools aus internen Magazinen in das Magazin.

11.10.31 GETFREELOC - Suche Leerplatz

Die Funktion `GETFREELOC` ist für WZMG verfügbar.

Die Funktion `GETFREELOC` sucht für ein gegebenes Werkzeug einen Leerplatz in den Magazinen, die dem genannten Beladeplatz oder dem genannten Spindel-/Werkzeughalter-Platz durch einen Eintrag in der Distanztabelle zugeordnet ist. Als Suchstrategie wird die durch $TC_MAMP2 bzw. $TC_MAP10 eingestellte Strategie verwendet. Die Suchstrategie kann programmiert werden.

Definierte Platztyphierarchien werden bei der Leerplatzsuche genauso berücksichtigt wie bei der Leerplatzsuche durch PI-Dienst oder beim programmierten Werkzeugwechsel.

Falls weniger als drei Parameter programmiert werden, wird der Alarm 14020 erzeugt.

Hinweis

GETFREELOC reserviert den gefundenen Leerplatz nur dann, wenn der optionale Parameter `withReserv` mit "L" oder "S" programmiert wird.

GETFREELOC(magNo&, locNo&, T-Nr., refMag, refLoc, withReserv)

Suche/Prüfe Leerplatz für genanntes Werkzeug bzgl. genannten Belade-/ Zwischenspeichermagazin und Platznummer. Reserviere den gefundenen Platz entsprechend der Programmierung des Parameters withReserv.

Syntax

```
GETFREELOC(magNo, locNo, T-Nr., refMag, refLoc, withReserv)
```

Bedeutung

GETFREELOC	Sucht für ein gegebenes Werkzeug einen Leerplatz in den Magazinen	
	Datentyp:	-

`mgNo`	Magazinnummer des Magazins; Der Parameter ist sowohl Eingabe- als auch Ergebnisparameter.	
	Datentyp:	REF INT
	Eingabewert	
	> 0	Magazinnummer des Magazins, in dem die Suche erfolgen soll. Dabei ist zu beachten, dass nur Magazinnummern gültig sind, die vom programmierten Referenzplatz (Parameter `refMag`/`refLoc`) aus erreichbar sind.
	0	Keine Vorgabe des Magazins, in dem gesucht werden soll. Suche beginnt in dem Magazin, das entsprechend der eingestellten Suchstrategie das Erste ist.
	Ausgabewert	
	> 0	Magazinnummer des Magazins in dem der Leerplatz gefunden wurde.
	0	Falls kein Leerplatz gefunden wird.
	-1	WZMG nicht aktiv
	-2	Ungültige Magazinnummer angegeben
	-3	Ungültige Magazinplatznummer angegeben. Die Platznummer gilt auch als ungültig, wenn die Magazinnummer ungültig ist.
	-4	Ungültige T-Nummer `tNo` angegeben.
	-5	Ungültiger Buchstabe für "`refMag`".
	-6	Falls `refMag` = = "S", ungültige WZ-Halternummer `refLoc` angegeben. Falls `regMag` = = "L", ungültige Beladeplatznummer `refLoc` angegeben.
	-7	Parameter "`withReserv`" hat nicht erlaubten Wert.
	-8	Parameter `withReserv` = "reserviert für neu zu beladenes WZ" kann nicht für Werkzeuge programmiert werden, die sich im Zwischenspeicher befinden (unabhängig davon, wie der Wert des Eigentümermagazins dieses Werkzeugs ist).
	-9	Parameter `withReserv` = "reserviert für WZ im Zwischenspeicher" ist nur möglich für Werkzeuge, die sich im Zwischenspeicher befinden.
	-10	Der Wert des Parameters "`withReserv`" verträgt sich nicht mit dem programmierten Wert des Parameters `refMag`.
	-11	Erster Parameter hat den Wert einer Multitoolnummer (d. h. Leerplatzsuche im MT). Es dürfen nur drei Parameter programmiert werden.
	-99	Sonstiges (unerwartetes) Problem in der Programmierung.

locNo	Magazinnummer des Platzes; Der Parameter ist sowohl Eingabe- als auch Ergebnisparameter	
	Datentyp:	REF INT
	Eingabewert	
	> 0	Magazinplatznummer des Platzes, der für die Aufnahme des angegebenen Werkzeugs geprüft werden soll. Falls magNo=0 programmiert ist, wird ein Wert locNo > 0 ignoriert.
	0	Keine Vorgabe des Magazinplatzes. Suche beginnt mit dem Platz, der entsprechend der eingestellten Suchstrategie der Erste ist.
	Ergebniswert	
	> 0	Magazinplatznummer des gefundenen bzw. geprüften Leerplatzes.
	0	Falls kein Leerplatz gefunden wird.
	-1, -2, ..., -99	Bedeutungsgleich mit den Fehlerstatuswerten des Parameters magNo.
tNo	T-Nummer des Werkzeugs, für das ein Leerplatz gesucht wird. Der gesuchte Platz muss für die Werkzeuggröße und für den im Werkzeug definierten Magazinplatztyp geeignet sein. Wird eine ungültige T-Nummer programmiert, dann geben die Parameter magNo, locNo jeweils den Wert -4 zurück.	
refMag	Referenzmagazin bzgl. dem die Leerplatzsuche erfolgen soll (optional).	
	Datentyp:	CHAR
	"S" = Zwischenspeichermagazin	
	"L" = Beladem agazin	
	"-" = kein Referenzmagazin. Wird verwendet, wenn konkret ein Magazin angegeben werden soll.	
	Wird ein Wert ungleich "S", "L"programmiert, dann geben die Parameter magNo, locNo jeweils den Wert -5 zurück. Ist das genannte Referenzmagazin noch nicht definiert, dann geben die Parameter magNo, locNo ebenfalls den Wert -5 zurück.	
refLoc	Spindelnummer/Werkzeughalternummer;	
	Datentyp:	INT
	Ist refMag gleich "S", wird hier die Spindelnummer/Werkzeughalternummer angegeben bzgl. der die Leerplatzsuche erfolgen soll. Wird eine ungültige WZ-Halternummer programmiert, dann geben die Parameter magNo, locNo jeweils den Wert -6 zurück. Ist refMag gleich "L", wird hier die Nummer des Platzes im Beladem agazin angegeben, bzgl. dessen der Leerplatz gesucht werden soll. Wird eine ungültige Nummer programmiert, dann geben die Parameter magNo, locNo jeweils den Wert -6 zurück. Der Parameter ist optional. Falls er nicht programmiert wird, wird für refMag = "S" die Suche bzgl. des Master-Werkzeughalters durchgeführt. Für refMag = "L" wird die Suche bzgl. der Platznummer = 1 im Beladem agazin durchgeführt. Für refMag = "-" wird der Parameter nicht berücksichtigt.	

withReserv	Reserviert den gefundenen/geprüften Leerplatz im realen Magazin.	
	Datentyp:	CHAR
	"L" = reserviert den gefundenen/geprüften Leerplatz im realen Magazin mit "reserviert für zu beladenes WZ" (das ist im Systemparameter $TC_MPP4 der Bitwert "H8). Die Reservierung wird nur gesetzt, falls das programmierte Werkzeug noch nicht beladen ist, d. h., die Eigentümeradresse des Werkzeugs Null ist. Andernfalls wird der Status auf den Wert = -8 gesetzt.	
	"S" = Falls das Werkzeug zur programmierten T-Nummer bereits auf einem Platz im Zwischenspeicher ist: reserviere den gefundenen/geprüften Leerplatz im realen Magazin mit "reserviert für WZ im Zwischenspeicher" (das ist im Systemparameter $TC_MPP4 der Bitwert "H4"; üblicherweise automatisch gesetzt, wenn der WZ-Wechsel mit T / M06 programmiert wird). Falls die Eigentümermagazinadresse dieses Werkzeugs = 0 ist und das Werkzeug sich auf irgendeinem Magazinplatz befindet (d. h., aktuelle Magazinadresse des Werkzeugs ungleich 0): setze zusätzlich diese Eigentümeradresse auf den Wert des gefundenen/geprüften Leerplatzes. Falls das Werkzeug bereits eine Eigentümermagazinadresse hat, dann wird diese durch die neue Magazinadresse ersetzt, falls die neue eine andere ist als die alte. Falls der Platz der alten Eigentümermagazinadresse reserviert ist, wird die Reservierung für diesen Platz aufgehoben. Falls für das programmierte Werkzeug bereits ein Platz reserviert ist, wird diese Reservierung aufgehoben, bevor die neue Reservierung gesetzt wird."-" = lässt den Resevierungszustand des gefundenen/geprüften Leerplatzes unverändert. Der Parameter ist optional. Falls er nicht programmiert ist, so gilt "-" als programmiert, d. h., der Reserviertzustand bleibt unverändert.	

Hinweis

Sind mehrere Parameter falsch, so hängt der Wert von magNo, locNo davon ab, welcher Parameter von NC zuerst geprüft wird.

Wird für ein festcodiertes Werkzeug ein Leerplatz gesucht, liefert GETFREELOC immer den Eigentümerplatz dieses Werkzeugs.

Plätze interner Magazine (Beladem agazin, Zwischenspeichermagazin) können nicht reserviert werden.

Ist WZMG nicht aktiv, so wird ein Alarm erzeugt.

Beispiele

Es gibt zwei Magazine mit den Magazinnummern 1 und 2, die jeweils 5 Plätze haben. Außerdem gibt es zwei Spindeln (Spindelnummer 1 und 2 mit den Platznummern 1 und 2) und einen Greifer (Platznummer 3, Spindelnummer 1 zugeordnet) im Zwischespeichermagazin und zwei Beladeplätze (Platznummer 1 und 2) im Beladem agazin.

Beide Magazine sind mit den Plätzen des Zwischenspeichers und den Plätzen des Beladem agazins verbunden (über die Distanzbeziehungen). Die folgenden Beispiele sind unabhängig voneinander zu verstehen.

Leerplatzsuche/-reservierung für Beladen über Beladeplatz

```
def int magNo=0, locNo=0

def int tNo=44     ;WZ mit T-Nr. = 44 ist definiert

GETFREELOC  ( magNo, locNo, tNo, "L", 2 )
     ; für das definierte WZ mit T-Nr. = 44 wird bzgl. des Beladeplatzes 2 ein Leerplatz gesucht.
     ; Der geeignete Leerplatz ist im Magazin 2, Platz 5. Die Referenzen magNo/locNo geben
     ; die Werte 2/5 zurück.
     ; Der Zustand des Magazinplatzes wird dabei nucht verändert.

def int magNo=0, locNo=0

def int tNo=44     ;WZ mit T-Nr. = 44 ist definiert

GETFREELOC  ( magNo, locNo, tNo, "L", 2, "L" )
     ; für das definierte WZ mit T-Nr. = 44 wird bzgl. des Beladeplatzes 2 ein Leerplatz gesucht.
     ; Der geeignete Leerplatz ist im Magazin 2, Platz 5. Die Referenzen magNo/locNo geben
     ; die Werte 2/5 zurück.
     ; Der Platz 2/5 wird reserviert "für zu beladendes WZ".

def int magNo=0, locNo=0

def int tNo=44      ;WZ mit T-Nr. = 44 ist definiert

GETFREELOC  ( magNo, locNo, tNo, "L", 2, "S" )
     ; für das definierte WZ mit T-Nr. = 44 wird bzgl. des Beladeplatzes 2 ein Leerplatz gesucht.
     ; Die Programmierung ist unzulässig, da für refMag = "L" die Programmierung
     ; withReserv = "S" nicht definiert
     ; Die Referenzen magNo/locNo geben die Statuswerte = -10  /-10 = der Wert des
     ; Parameters "withReserv" verträgt sich nicht mit dem Wert des
     ; Parameters "refMag" zurück

def int magNo=0, locNo=0
def int tNo = 44      ;WZ mit T-Nr. = 44 ist definiert

GETFREELOC  ( magNo, locNo, tNo, "L", 2, "e" )
     ; für das definierte WZ mit T-Nr. = 44 wird bzgl. des WZ-Halters mit Platznr. = 1 ein
     ;Leerplatz  gesucht.
     ; Die Programmierung ist unzulässig, da der Parameter "withReserv" mit falschem Wert
     ; programmiert ist.
     ; Die Referenzen magNo/locNo ; geben die Statuswerte = -7 / -7 = Parameter "withReserv"
     ;  hat nicht erlaubten Wert zurück.
```

Leerplatzsuche/-reservierung für Beladen über WZ-Halter/Spindel

```
def int magNo = 0, locNo = 0

def int tNo = 4   ;WZ mit T-Nr. = 4 ist definiert

GETFREELOC  ( magNo, locNo, tNo, "S" )
```

bzw. funktional gleichbedeutend

```
GETFREELOC  ( magNo, locNo, tNo, "S", "-" )
     ; für das definierte WZ mit T-Nr.  = 4 wird bzgl. der Masterspindel ein Leerplatz gesucht.
     ; Der geeignete Leerplatz ist im Magazin 1, Platz 3. Die Referenzen
     ; magNo/locNo geben die Werte 1/3 zurück.
     ; Der Zustand des Magazinplates wird dabei nicht verändert.

def int magNo = 2, locNo = 0
```

```
def int tNo = 44

; WZ mit T-Nr. = 44 ist definiert, ist auf dem WZ-Halter und hat noch keinen
; Eigentümerplatz im Magazin (gilt damit als noch nicht beladen),
; oder hat schon einen Eigentümerplatz in einem Magazin (beladen)

GETFREELOC( magNo, locNo, tNo, "S", 1, "L" )
     ; für das Werkzeug mit der T-Nr. 44 wird bzgl. des WZ-Halters mit Platznr. 1 ein
     ; ein Leeplatz gesucht. Die Programmierung ist unzulässig, da versucht wird, die
     ; Reservierung "reserviert für neu zu beladenes WZ" für ein WZ im Zwischenspeicher
     ; zu setzen. Die Referenzen magNo/locNo geben die Statuswerte -8 / -8 = "Parameterwert
    : kann nicht für WZe programmiert werden, die sich im Zwischenspeicher befinden" zurück.

def int magNo, locNo

def int tNo = 44  ; WZ mit T-Nr. = 44 ist definiert, ist auf dem WZ-Halter

GETFREELOC( magNo, locNo, tNo, "S", 1, "S" )
     ; es gilt dieselbe Ausgangsposition wie bei der Programmierung des vorstehenden
     ; Befehls. Der geeignete  Leerplatz ist im Magazin 2, Platz 5. Die Referenzen magNo/locNo
     ; geben die Werte 2/5 zurück. Der Platz 2/5 wird reserviert "für WZ im Zwischenspeichere".

def int magNo = 0, locNo = 0

def int tNo = 44
      ; WZ mit T-Nr. = 44 ist definiert und ist im Greifer enthalten, der über $TC_MLSR mit dem
      ; WZ-Halter des Platzes mit der Nummer 1 verbunden ist

GETFREELOC( magNo, locNo, tNo, "S", 3, "S" )
     ; für das definierte WZ mit T-Nr.  = 44 wird bzgl. des WZ-Halters mit Platnr. 1 ein Leerplatz
     ; gesucht. Der geeignete Leerplatz ist im Magazin 2, Platz 5. Die Referenzen magNo/locNo
     ; geben die Werte 2/5 zurück.
     ; Der Magazinplatz 2/5 wird reserviert "für WZ im Zwischenspeicher".
     ; (Bei NC Hochlauf bleibt dieser Zustand erhalten, falls das reservierende WZ zu dem
     ; Zeitpunkt noch im Zwischenspeicher ist)

def int magNo = 0, locNo = 0

def int tNo = 44  ; WZ mit T-Nr. = 44 ist definiert und nicht im Zwischenspeicher

GETFREELOC( magNo, locNo, tNo, "S", 2, "S" )
    ; für das definierte WZ mit T-Nr.  = 44 wird bzgl. des Zwischenspeicherplatzes 2 ein Leerplatz
     ; gesucht.
     ; Die Programmierung ist unzulässig, da versucht wird, die Reservierung "reserviert für
     : WZ im Zwischenspeicher" für eine WZ-Suche des Zwischenspeichers zu setzen.
     ; Die Referenzen magNo/locNo geben die Statuswerte -9 / -9 = "Parameterwert ist nur
     ; möglich für WZe, die sich im Zwischenspeicher befinden" zurück.

def int magNo = 0, locNo = 0

def int tNo = 44  ; WZ mit T-Nr. = 44 ist definiert, aber (noch) nicht beladen.

GETFREELOC( magNo, locNo, tNo, "S", 1, "S" )
     ; für das definierte WZ mit T-Nr.  = 44 wird bzgl. des WZ-Halters mit Platznr. 1
     ; ein Leerplatz gesucht.
     ; Der geeignete Leerplatz ist im Magazin 2, Platz 5.
     ; Da das WZ aber nicht beladen werden kann,
     ; kann der letzte Parameter nicht mit dem Wert "S" programmiert werden.
```

```
    ; Die referenzen magNo/locNo geben die Statuswerte -8 / -8 = "Parameterwert kann nicht
    ; für WZe programmiert werden, die sich im Zwischenspeicher befinden" zurück.
def int magNo = 0, locNo = 0
def int tNo = 44  ; WZ mit T-Nr. = 44 ist definiert, aber (noch) nicht beladen.
GETFREELOC( magNo, locNo, tNo, "S", 1, "L" )
    ; für das definierte WZ mit T-Nr. = 44 wird bzgl. des WZ-Halters mit Platznr. 1 ein Leerplatz
    ; gesucht. Der geeignete Leerplatz ist im Magazin 2, Platz 5. Die Referenzen
    ; magNo/locNo geben die Werte 2/5 zurück.
```

11.10.32 DELMLRES - Lösche den Platzzustand "reserviert für WZ im Zwischenspeicher"

Die Funktion `DELMLRES` ist für WZMG verfügbar.

Die Funktion `DELMLRES` hebt die Platzreservierung der Art `resType` des Platzes mit der Nummer `magNo` auf.

Syntax

```
state = DELMLRES (magNo, locNo, resType)
```

Bedeutung

`delmlres`	Hebt Platzreservierung auf	
	Datentyp:	INT
`magNo`	Magazinnummer des Magazins, in dem der Platz enthalten ist.	
	Datentyp:	INT
	Einheit:	
	Wertebereich:	*MIN* < x < *MAX*
`locNo`	Magazinplatznummer, des Platzes, dessen Reservierung gelöscht werden soll.	
	Datentyp:	INT
`resType`	Typ der Reservierung, die gelöscht werden soll (optional) • "L" Reservierung "für zu beladenes Werkzeug" ("L "= "Tool to be Loaded") • "S" Reservierung "für Werkzeug im Zwischenspeicher" ("S" = Tool in Spindle" bzw. allgemein WZ im Zwischenspeicher) Wenn `resType` nicht programmiert wird, dann wird die Reservierung gelöscht, die gesetzt ist - entweder "L" oder "S". Wenn keine Reservierung gesetzt ist, dann hat der Befehl keine Auswirkung.	
Rückgabewert		
`state`	Ergebnis der Befehlsausführung	
	Datentyp:	INT
	0	Befehl erfolgreich ausgeführt (auch wenn der Platz nicht reserviert war).
	-1	WZMG ist nicht aktiv
	-2	Ungültige Magazinnummer angegeben

	-3	Ungültige Magazinplatznummer angegeben
	-4	Ungültiger Buchstabe für `resType`

Beispiel 1

Für das auf Greifer 1 (Magazinnr. = 9998, Platnr. = 4) befindliche übergroße Werkzeug soll die Platzreservierung "für Werkzeug im Zwischenspeicher" gelöscht werden.

Programmcode	**Kommentar**
`def int tNo, magNo, locNo`	`;`
`tNo = $TC_MPP6[ 9998, 4 ]`	`;`
`magNo = $A_MYMN[ tNo ]`	`; Eigentuemermagazin`
`locNo = $A_MYLN[ tNo ]`	`; Eigentuemermagazinplatz, der ; reserviert wurde`
`state = DELMLRES ( magNo, locNo, "s" )`	`; gleichbedeutend waere "DELMLRES(magNo, locNo)"`
`if (state < 0 ) gotof fehler`	
`;Falls sichergestellt werden soll, dass der Platz auch für das gedachte einzuwechselnde Werkzeug reserviert ist, kann man zusätzlich prüfen:`	
`if ( $TC_MPP66[ magNo, locNo ] GOTOF FEHLER`	

Beispiel 2

Programmiertes Werkzeug Beladen über WZ-Halter (Magazinnr. 9998, Platznr. = 2). Der Beladevorgang wird mit den Befehlen GETFREELOC, MVTOOL programmiert.

Programmcode	**Kommentar**
`def int tNo, magNo, locNo`	`;`
`tNo = 3`	`; dieses WZ soll beladen werden`
`GETFREELOC (magNo, locNo, tNo, "S", 2, "L")`	`; Leerplatzsuche mit Reservierung`
`if ( (magNo > 0) ans (locNo > 0)`	
`  and ( "irgend etwas ist nicht wie gewuenscht") )`	
`  if ( state < 0 ) GOTOF FEHLER`	
`endif`	
`MVTOOL( state, 9998, 2, magNo, locNo )`	

11.10.33 DELMLOWNER - Lösche Eigentümermagazinplatz des Werkzeugs

Die Funktion `DELMLOWNER` ist für WZMG verfügbar.

Die Funktion `DELMLOWNER` löscht das Werkzeug oder das Multitool im Zwischenspeichermagazin den Eigentümermagazinplatz.

Syntax

```
state = DELMLOWNER(t)
```

Bedeutung

DELMLOWNER>:	Löscht das Werkzeug oder das Multitool im Zwischenspeichermagazin den Eigentümermagazinplatz	
	Datentyp:	INT
t	T-Nummer des Werkzeugs, in dem der Eigentümerplatz gelöscht wird	
	Datentyp:	INT
Rückgabewert		
state	Ergebnis der Befehlsausführung	
	0	Befehl erfolgreich ausgeführt.
	-1	WZMG ist nicht aktiv
	-2	WZ mit der T-Nr. = t existiert nicht - auch für t=0
	-3	das WZ ist nicht beladen (hat keinen Eigentümerplatz)
	-4	das WZ ist nicht im Zwischenspeichermagazin

Beispiele

Falls der Eigentümerplatz für dieses Werkzeug reserviert ist, dann wird implizit auch die Platzreservierung "für Werkzeug im Zwischenspeicher" gelöscht.

Die Systemparameter $A_MYMN [t] und $A_MYMLN [t] lesen nach dem Löschvorgang den Wert = 0.

Programmcode	**Kommentar**
`def int tNo`	`;`
`tNo = GETT( name, duplo )`	`; falls nur Name und Duplonummer des WZs bekannt sind`
`tNo = $TC_MPP&[ magNo, locNo ]`	`; falls nur der Magazinplatz bekannt ist, auf dem das WZ enthalten ist`
`tNo = $TC_MPP&&[ magNo, locNo ]`	`; falls nur der magazinplatz bekannt ist, dessen Reservierung gelöscht werden soll`

Multitool

Die Funktion DELMLOWNER ist auch für Multitools verfügbar.

```
state = DELMLOWNER(INT MTno)
```

- Lösche im Multitool mit der Nummer MTno im Zwischenspeichermagazin den Eigentümermagazinplatz
- Lösche damit auch den Eigentümerplatz der im Multitool enthaltenen Werkzeuge mit den Nummern Tnoi; i=1, ..., $P_MTOOLNT.

Die Systemparameter $A_MYMN [MTno] / $A_MYMLN [MTno] und $A_MYMN [Tnoi] / $A_MYMLN [Tnoi] lesen nach dem Löschvorgang den Wert = 0.

Hinweis

Die Systemparameter $A_TOOLMTN [Tnoi], $A_TOOLMTLN [Tnoi] und $A_MYMTN [Tnoi], $A_MYMTLN [Tnoi] lesen nach dem Löschvorgang weiterhin den Wert des Multitools, MT-Platzes auf dem, das WZ mit der T-Nummer Tnoi bestückt ist.

Hinweis

Der Befehl kann nicht programmiert werden mit der T–Nummer eines Werkzeugs, das in einem Multitool bestückt ist. Dieser Programmierung wird mit state = -5 abgewiesen.

11.10.34 $P_USEKT - Werkzeugwechsel nur mit Werkzeugen der Untergruppe

Übersicht

Diese Funktion ist für WZMO und WZMG verfügbar.

Mit diesem Befehl kann eine Untermenge der Werkzeuge einer WZ-Gruppe ausgewählt werden, die bei den anschließenden Werkzeugwechseln berücksichtigt werden (Werkzeug-Technologiegruppe).

Die Untergruppen werden mit der Systemvariablen $TC_TP11[t] eingestellt.

Name	**$P_USEKT**			
Bedeutung	$P_USEKT ist ein bitcodierter Wert. Es sind nur die Bits 0-3 von inhaltlicher Bedeutung. Alle Werkzeuge, deren Parameter $TC_TP11 eines der Bits von $P_USEKT gesetzt haben, stehen den folgenden Werkzeugwechseln zur Verfügung. Der Wert 0 bedeutet "alle Bits sind gesetzt". Gibt es in einer Werkzeuggruppe, für die ein Werkzeugwechsel programmiert wurde, kein solches Werkzeug. so wird ein Alarm erzeugt.			
Datentyp	INT			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	x	x	-	-
Impliziter Vorlaufstopp	-	-		

Mit dem Maschinendatenbit $MC_RESET_MODE_MASK, Bit 20 kann eingestellt werden, ob nach Reset bzw. Programmende der zuletzt programmierte Wert von $P_USEKT erhalten bleibt oder ob $P_USEKT auf den in $MC_USEKT_RESET_VALUE eingetragenen Wert gesetzt wird.

Verhalten bei NC-Reset

Zur genaueren Definition des Verhaltens bei NC-Reset und NC-Start wird ein weiteres Bit in $MC_RESET_MODE_MASK definiert:

Mit dem neu definierten Bit 20 = 1 wird in Verbindung mit dem Bit 0 = 1 des gleichen Maschinendatums festgelegt, dass der programmierte $P_USEKT-Wert bei NC-Reset erhalten bleibt.

In den anderen drei möglichen Kombinationen von Bit 0 und Bit 20 wird der Wert aus dem MD $MC_USEKT_RESET_VALUE genommen.

Verhalten bei NC-Start

Es bleibt der zuletzt eingestellte $P_USEKT-Wert wirksam. Da $P_USEKT nur durch ein NC-Programm geändert werden kann, wirkt der $P_USEKT, der beim letzten NC-Reset eingestellt wurde.

Es ist möglich, beispielsweise durch Verwendung von ASUP-Programmen bei Programmstart (ProgEvents), einen gewünschten Wert n in $P_USEKT zu programmieren.

Falls der WZ-Wechsel mit der Funktion T="Platz" programmiert wird, so kann der Befehl $P_USEKTebenfalls verwendet werden. Es ist dann allerdings nötig, dass das WZ auf dem programmierten Magazinplatz im Parameter $TC_TP11 mindestens ein Bit des programmierten $P_USEKT-Wertes gesetzt hat. Andernfalls wird ein WZ eines anderen Magazinplatzes angewählt. Bei dieser Einstellung gilt weiterhin: Bei NC–Start ist der Wert von $P_USEKT = 0 vorbelegt, unabhängig von $MC_RESET_MODE_MASK und $MC_USEKT_RESET_VALUE.

Hinweis

Die Systemvariable $TC_TP11 wurde bisher in NC nicht ausgewertet. Der Wert wird automatisch mit 0 vorbelegt. Bei bestehenden Datensätzen sollte geprüft werden, ob die enthaltenen Werte geeignet sind.

Die Programmierung $P_USEKT = 0 hat die Bedeutung, dass alle Werkzeuge der Werkzeuggruppe bei der Werkzeuganwahl betrachtet werden.

Der Wert $TC_TP11[t] = 0 hat die Bedeutung "das Werkzeug gehört allen definierten Werkzeuggruppen an". Damit ist die Kompatibilität zu bestehenden Datensätzen gegeben.

Wird mit der Funktion T=Platz gearbeitet, kann $P_USEKT **nicht** programmiert werden.

Übersicht über das Verhalten bei den verschiedenen Einstellungen

Der Wert von `$MN_USEKT_RESET_VALUE` ist 8.

Der Werte von `$P_USEKT` ergibt dabei bei folgenden Programmen

1. Programm:

```
N10 $P_USEKT=2
N20 M00
N30 M02
M02
```

2. Programm:

```
N100 R0=$P_USEKT
```

Einstellung von $MC_RESET_MODE_MASK	Bit 0 = 0 Bit 20 = 0	Bit 0 = 1 Bit 20 = 0	Bit 0 = 0 Bit 20 = 1	Bit 0 = 1 Bit 20 = 1
Aktionsreihenfolge				
Nach Hochlauf	8	8	8	8
Bei "N20 M00" (nach P_USEKT setzen)	2	2	2	2
Nach Programmende	8	8	8	2
Bei "N100 M00" (nach Programmstart)	8	8	8	2

Durch die Bitcodierung ist es möglich, dass ein WZ mehreren WZ-Untergruppen angehören kann. Durch Konfiguration von NC werden maximal 4 verschiedene WZ-Untergruppen vorgesehen; d.h., es werden nur die Bits 0, 1, 2, 3 berücksichtigt – sowohl bei der Programmierung von $P_USEKT als auch bei $TC_TP11. Werkzeuge mit $TC_TP11[t] = 0 zählen dabei nicht als WZ-Untergruppe.

Beispiel

Die WZ-Gruppe "Fraeser_25" besteht aus 4 Werkzeugen.

(es gilt: Tool_Change_Mode=1)

```
Fraeser_25       Duplo 1          T_Nr. 1          TP11=1
Fraeser_25       Duplo 2          T_Nr. 2          TP11=2
Fraeser_25       Duplo 3          T_Nr. 3          TP11=4
Fraeser_25       Duplo 4          T_Nr. 4          TP11=8
%MPF
...
T="Fraeser_25"

M06              Jedes WZ dieser Gruppe kann eingewechselt werden, da noch keine Selektion getroffen wurde. Es entscheidet die eingestellte Suchstrategie.
...
$P_USEKT=2
...
T="Fraeser_25"
M06              "Fraeser_25", Duplo 2 eingewechselt
...
$P_USEKT=9
...
T="Fraeser_25"
M06              "Fraeser_25", Duplo 1 oder Duplo 4 wird eingewechselt (abhängig von der eingestellten Suchstrategie)
...
$P_USEKT=0
```

```
...
T="Fraeser_25"
M06
```
Jedes WZ dieser Gruppe kann eingewechselt werden, da mit USEKT=0 die Selektion aufgehoben wurde. Es entscheidet die eingestellte Suchstrategie.

```
...
$P_USEKT=15
...
T="Fraeser_25"
M06
```
Jedes WZ dieser Gruppe kann eingewechselt werden, da alle Bits gesetzt sind. Es entscheidet die eingestellte Suchstrategie.

T=Platz, automatische Werkzeug-Selektion

Zunächst wird versucht das Werkzeug des programmierten Magazinplatzes einzuwechseln, unabhängig vom Wert in $TC_TP11. Mit Anwahl des Werkzeugs wird der TP11-Wert dieses Werkzeugs in $P_USEKT übernommen und gilt als programmiert. D.h., es können nur Werkzeuge dieser Untergruppe angewählt werden.

Ist das Werkzeug auf dem programmierten Platz jedoch nicht einsatzfähig, wird der $TC_TP11-Wert des Werkzeugs auf dem programmierten Magazinplatz betrachtet, um zum Ersatzwerkzeug zu gelangen. Ersatzwerkzeuge können nur solche Werkzeuge sein, die in $TC_TP11 eines der Bits des gesperrten Werkzeugs haben.

11.10.35 TOOLGNT/TOOLGT - Werkzeuggruppen

Die Funktionen `TOOLGNT` und `TOLLGT` sind für WZMO und WZMG verfügbar.

`TOOLGNT` bedeutet verkürzt: **TOOLG**roup**N**umberOf**T**ools - Anzahl der Werkzeuge der Gruppe.

`TOOLGT` bedeutet verkürzt: **TOOLG**roup**T**oolNumber - T-Nr. des i-ten Werkzeugs der Werkzeuggruppe mit i = 1, ...TOOLNTG.

Die Sprachbefehle ermöglichen es, Auskunft über die Werkzeuge in der Werkzeug-Gruppe zu erhalten.

TOOLGNT("Bezeichner")

TOOLGT("Bezeichner", i)

Syntax

```
result = TOOLGNT("Bezeichner")
result = TOOLGT("Bezeichner", i)
```

Bedeutung

`TOOLGNT`	Anzahl der Werkzeuge der Gruppe	
`TOOLGT`	T-Nummer des i-ten Werkzeugs der Werkzeuggruppe	
	Datentyp:	INT

Bezeichner	Name des Werkzeugs	
	Datentyp:	STRING
i	Werkzeuggruppe; i = 1 TOOL0TNG	
	Datentyp	STRING
Rückgabewert		
result	Ergebniswert	
	Datentyp:	INT
	> 0	erfolgreicher Lesezugriff
	0	i ist ein Wert außerhalb des erlaubten Bereichs
	-1	weder Funktion WZMG noch Funktion WZMO aktiv
	-2	zum Namen "Bezeichner" existiert kein Werkzeug

Die Reihenfolge der Werkzeuge in der Werkzeug-Gruppe wird von NC festgelegt und ändert sich im Laufe der programmierten Werkzeug-Wechsel mit Werkzeugen aus dieser Gruppe.

Beispiel

Es gibt die Werkzeuge "Bohrer_6mm"/Duplo_1, "Bohrer_6mm"/Duplo_2 und "Bohrer_6mm"/Duplo_3.

Zuerst wird die Anzahl Werkzeuge der Gruppe "Bohrer_6mm" ausgelesen.

```
Programmcode                                    Kommentar
R1=TOOLGNT("Bohrer_6mm")                        ; R1=3
;Dann wird die T-Nummer dieser Werkzeuge ermittelt.
R11=TOOLGT("Bohrer_6mm,1))
R12=TOOLGT("Bohrer_6mm,2)
R13=TOOLGT("Bohrer_6mm,3)
```

11.10.36 $P_TMNOIS - ist Nummer T-Nummer, Magazinnummer oder MT-Nummer

Die Systemvariable $P_TMNOIS ist mit WZMO und WZMG verfügbar.

Die Systemvariable $P_TMNOIS liefert die Nummerneigenschaft zurück. Da sowohl die Einzel–Werkzeuge, Magazine als auch die Multitools zur Nummerierung das Nummernband 1–32000 benutzen, muss unterschieden werden können, was sich hinter einer Nummer verbirgt. Das leistet dieser Systemparameter.

Syntax

```
result = $P_TMNOIS[ nr ]
```

Bedeutung

$P_TMNOIS	Liefert zurück, welche Nummerneigenschaft die angegebene Nummer hat.	
	Datentyp:	INT

nr	Nummer, die untersucht werden soll	
	Datentyp:	INT
Rückgabewert		
result	Eingeschaft der Nummer	
	Datentyp:	INT
	3	nr ist die Nummer eines definierten Werkzeugs und die Nummer eines definierten Magazins
	2	nr ist die Magazinnummer eines definierten Magazins
	1	nr ist die T-Nummer eines definierten Werkzeugs
	0	nr ist die MT-Nummer eines definierten Multitools
	-3	ungültige Nummer. nr ist weder die Nummer eines Werkzeugs oder eines Magazins, noch die eines Multitools.

Beispiele

Überprüft was die Nummer darstellt, z. B. eine T-Nummer, MT-Nummer oder eine ungültige Nummer

```
Programmcode                    Kommentar
def int result                  ;
def int nr=4711                 ;
```

11.10.37 $P_TOOLEXIST - Existenz eines Werkzeugs feststellen

Die Systemvariable $P_TOOLEXIST liefert zu einer T-Nummer die Information, ob dazu ein Werkzeug existiert.

$P_TOOLEXIST wird mit der internen T-Nummer parametriert. Rückgabe ist true oder false. Ist die T-Nr. die eines Multitools liefert $P_TOOLEXIST=false, ist die T-Nr. die eines existierenden Werkzeugs liefert $P_TOOLEXIST=true. Alternativ kann mit $P_TMNOIS (Seite 225) gearbeitet werden.

Syntax

```
$P_TOOLEXIST[t]
```

Bedeutung

$P_TOOLEXIST	Liefert zurück, ob für eine bestimmte Nummer ein Werkzeug existiert.	
	Datentyp:	BOOL
	TRUE	Werkzeug mit dieser Nummer existiert
	FALSE	Werkzeug mit dieser Nummer existiert nicht.
t	T-Nummer des Werkzeugs	
	Datentyp:	INT

Die beiden Befehle stehen jeweils in allen Funktionsausprägungen der WZV zur Verfügung. Es gilt weiterhin, dass der Ergebniswert True von $P_TOOLEXIST dem Ergebniswert 1 von $P_TMNOIS entspricht. Für die folgenden Beispiele gilt:

Beispiele

Beispiel 1: Ein neues WZ soll erzeugt werden (mit einer T-Nummer, die nicht mit der eines Multitools oder mit der eines bereits existierenden Werkzeugs übereinstimmt).

Programmcode	Kommentar
N1 DEF INT no=2	; zu prüfende Nummer
N5 if (($P_TMNOIS[no] != 0) and ($P_TMNOIS[no] != 1))	;
N6 $TC_TP2[no] = "WZ_2"	;neues WZ erzeugen
N7 endif	

Beispiel 2: Beispiel 2: Es wird ein neues MT erzeugt (mit einer MT-Nummer, die nicht mit der eines bereits existierenden Multitools, oder mit der eines bereits existierenden Werkzeugs, oder mit der eines bereits existierenden Magazins übereinstimmt):

Programmcode	Kommentar
N1 DEF INT no=2	; zu prüfende Nummer
N5 if (($P_TMNOIS[no] != 0) and ($P_TMNOIS[no] != 1) and ($P_TMNOIS[no] != 2))	;
N6 $TC_MTPN[no] = 5	; neues MT mit 5 Plätzen erzeugen
N7 endif	

Beispiel 3: Eine explizite Nummer soll darauf geprüft werden, ob sie zu einem definierten WZ gehört:

Programmcode	Kommentar
N1 DEF INT no=2	; zu prüfende Nummer
N5 if ($P_TOOLEXIST[no] == true)	;
N6 $TC_TP9[no] = 1	; definieren der Art der WZ-Überwachung
N7 endif	

11.10.38 $A_TOOLMN - Magazin-Nr. vom Werkzeug lesen

Die Systemvariable $A_TOOLMN gibt die Magazinnummer des Werkzeugs mit der T-Nummer t zurück.

Die Abkürzung TOOLMN bedeutet "**toolm**agazine **n**umber".

Syntax

```
$A_TOOLMN[t]
```

Bedeutung

`$A_TOOLMN`:	Liefert die Magazinnummer zurück	
	Datentyp:	INT
	0	Das Werkzeug ist keinem Magazin zugeordnet.
	-1	Die Funktion Werkzeugverwaltung ist nicht aktiv.
	-2	Es existiert kein Werkzeug mit der angegebenen T-Nummer.
	Liegt die T-Nummer außerhalb des Wertebereichs, wird ein Alarm ausgegeben.	
`t`	T-Nummer des Werkzeugs	
	Datentyp:	INT

11.10.39 $P_MTOOLN / $P_MTOOLMT- Anzahl Multitools / MT-Nummer ermitteln

Die Systemvariablen $P_MTOOLN bzw. $P_MTOOLMT sind mit WZMG verfügbar.

Einen Überblick über die definierten Multitools geben die folgenden Systemvariablen.

Syntax

```
result = $P_MTOOLN
result = $P_MTOOLMT[i]
```

Bedeutung

`$P_MTOOLN`	Liefert die Anzahl der Multitools in diesem Kanal zurück.	
	Datentyp:	INT
Rückgabewert		
`result`	Ergebniswert	
	> 0	Anzahl der definierten Multitools
	0	Kein Multitool definiert
	-1	Funktion WZMG nicht aktiv
	-2	Funktion Multitool nicht aktiv
`$P_MTOOLMT`	Gibt die Multitool-Nummer des i-ten Multitools zurück	
	Datentyp:	INT
`i`	Nummer des Multitools	
Rückgabewert		
`result`	Multitoolnummer des Multitools	
	Datentyp:	INT

Beispiel

Programmcode	Kommentar
def int anzahl = 0	
def int i = 0	
def string[32] mtName	
anzahl = $P_MTOOLN	
for i = 1 to anzahl	
r[i] = $P_MTOOLMT[i]	;schreibe alle Mt-Nummern der definierten Multitools :in aufeinanderfolgende R-Parameter
endfor	
mtName = $P_MTP2[$P_MTOOLMT[i]]	;schreibe Name des letzten Multitools nach mtName

11.10.40 $P_MTOOLNT / $P_MTOOLT - Anzahl der Werkzeuge im Multitool

Die Systemvariablen $P_MTOOLNT und $P_MTOOLT sind mit WZMG verfügbar.

Hinweis

Siehe hierzu die analogen WZ–spezifischen Systemparameter $P_TOOLNT, $P_TOOLT.

Die Systemvariable $P_MTOOLNT liefert die Anzahl der Werkzeuge im Multitool zurück.

Die Systemvariable $P_MTOOLT liefert die T-Nummer des i-ten Werkzeugs des Multitools zurück.

Syntax

```
result = $P_MTOOLNT[mt]
result = $P_MTOOLT[mt, i]
```

Bedeutung

$P_MTOOLNT	Liefert die Anzahl der Werkzeuge im Multitool zurück	
	Datentyp:	INT
mt	Multitoolnummer	
	Datentyp:	INT
Rückgabewert		

result	Ergebniswert	
	Datentyp:	INT
	> 0	Anzahl der Werkzeuge im Multitool
	0	kein WZ im Multitool mit der Nummer mt enthalten
	-1	Funktion WZMG nicht aktiv
	-2	Funktion Multitool nicht aktiv
	-3	mt ist nicht die Nummer eines definierten Multitools
$P_MTOOLT[mt, i]	Liefert die T-Nummer des i-ten Werkzeugs	
	Datentyp:	INT
mt	Multitoolnummer	
	Datentyp:	INT
i	Index der Multitoolwerkzeuge mit 0 < i < $P_MTOOLNT	
	Datentyp:	INT
Rückgabewert		
result	Ergebniswert	
	Datentyp:	INT
	> 0	Nummer des i–ten Werkzeugs im Multitool
	0	i ist ein Wert außerhalb des erlaubten Bereichs
	-1	Funktion WZMG nicht aktiv
	-2	Funktion Multitool nicht aktiv
	-3	mt ist nicht die Nummer eines definierten Multitools

Beispiel

Multitool mit Nummer 500 hat 6 Plätze. Auf den Plätzen 1, 2, 5 sind die 3 Werkzeuge mit den T–Nummern 11, 22, 33 bestückt.

```
Programmcode                    Kommentar
def int anzahl = 0
def int i = 0
anzahl = $P_MTOOLNT[500]        ;Anzahl der Werkzeuge ist 3
for i = 1 to anzahl             ;Schleife über 3 Werkzeuge und nicht über 6 MT-Plätze
r[i] = $P_MTOOLT [500, i]       ;schreibe die T-Nummern der bestückten Werkzeuge in
                                aufeinanderfolgende R-Parameter
                                ;r1=11, r1=22, r3=33
endfor
```

11.10.41 $A_TOOLMLN - Magazinplatz-Nr. von Werkzeug lesen

Anmerkung

TOOLMLN bedeutet verkürzt = "**toolm**agazine **l**ocation **n**umber".

Name	**$A_TOOLMLN[t]**			
Bedeutung	Gibt die Magazinplatznummer des Werkzeugs mit der T-Nr.=t zurück. Ist das nicht einem Magazin zugeordnet, so wird 0 zurückgegeben. Ist die Funktion Werkzeugverwaltung nicht aktiv, so wird -1 zurückgegeben. Existiert kein WZ mit der T-Nr.=t, so wird -2 zurückgegeben. Einen Alarm erhält man, wenn der Wertebereich für die T-Nummer verletzt wird.			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	Der Index gibt die T-Nummer an			1-32000
Zugriff	Lesen im Teileprogramm	Schreiben im Teileproramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	x	-	x	-
Impliziter Vorlaufstopp	x	-		

Anmerkung: Es ist nicht möglich, dass gilt $A_TOOLMLN[t]==0 und $A_TOOLMN[t]>0, oder $A_TOOLMLN[t]>0 und $A_TOOLMN[t]==0.

11.10.42 $P_TOOLND - Schneidenanzahl von Werkzeug lesen

Anmerkung

TOOLND bedeutet verkürzt = "**toolln**umber of **D**s".

Name	**$P_TOOLND[t]**			
Bedeutung	Gibt die Anzahl von Schneiden zurück des Werkzeugs mit der T-Nr.=t. Ein WZ hat immer mindestens eine Schneide. Standard: Existiert kein WZ mit der T-Nr.=t, so wird -1 zurückgegeben. Der Wert 0 wird als Indexfehler abgelehnt.			
Datentyp	INT			
Wertebereich	Standard: 1 - 9 1 - "Maschinendatenwert für die maximale Anzahl von D-Nummern"			
Indizes	Bedeutung			Wertebereich
	Der Index gibt die T-Nummer an			1-32000
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	x	-	-	-
Impliziter Vorlaufstopp	-	-		

11.10.43 $A_MONIFACT - Faktor für Standzeitüberwachung

Übersicht

Sollen verschiedene Werkstückmaterialien mit demselben Werkzeug bearbeitet werden, kann es erforderlich sein, die Zeitintervalle für die Überwachung zu verlängern bzw. zu verkürzen, um den unterschiedlich starken Werkzeugverschleiß zu erfassen. Der Faktor wird vor Einsatz des Werkzeugs entsprechend gesetzt. Die Schreiboperation wirkt hauptlaufsynchron.

Dafür wurde ein kanalspezifischer Parameter definiert, mit dem das aktuelle Zeitmaß multipliziert wird.

Über den Wert = 0 kann die Zeitüberwachung aller auf dem Kanal zum Einsatz kommende Werkzeuge über Teileprogramm ausgeschaltet werden.

Name	**$A_MONIFACT**			
Bedeutung	Nur von Bedeutung bei aktiver Zeitüberwachung in der Werkzeugverwaltung. Faktor zur Beeinflussung des Zeitmaßes beim Zählen der Zeit von zeitüberwachten Werkzeugen. Werte < 1 und > 0 verlangsamen die Zeitzählung (Die Uhr "läuft langsamer"). Werte > 1 beschleunigen die Zeitzählung (Uhr "läuft schneller"). Der Wert = 1 ist nach Steuerungshochlauf, nach Reset, nach M30 wirksam (Voreinstellung) und entspricht der Echtzeit. Der Wert 0 ist ebenfalls erlaubt und bewirkt ein Aussetzen der Zeitzählung aller zeitüberwachten Werkzeuge, die auf diesem Kanal in einer zeitüberwachten Spindel zum Einsatz kommen. Anmerkung: Mit negativen Werten kann man die Überwachungszeit "rückwärts laufen" lassen.			
Datentyp	REAL			
Wertebereich	Wertebereich des Typs REAL			
Indizes	Bedeutung			Wertebereich
				-
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	x	x	x	x
Impliziter Vorlaufstopp	-	x		

Standzeitzähler auf dem Monitor

Bei entsprechender Einstellung der Systemvariablen $A_MONIFACT kann der Standzeitzähler auf dem Monitor mit einer anderen Geschwindigkeit laufen als die Echtzeit. Dabei bleiben die Zeitwerte des BTSS-Bausteines TS und die NC-internen Zeitwerte erhalten wie bisher. Beide sind Echtzeitwerte.

BTSS lesen und schreiben:

Die Zeitwerte werden unabhängig vom aktuellen Wert von $A_MONIFACT ohne Änderung übertragen.

Hinweis

Standzeitwerte am HMI

Am HMI werden die Echtzeitwerte angezeigt, nicht die umgerechneten Standzeiten.

Beispiel

Vorgegeben sind die aktuellen Werte (Einheiten in Echtzeit).

Sollstandzeit: 10 Minuten

Iststandzeit: 2 Minuten - in **einer Minute** wird die Vorwarngrenze erreicht

Vorwarngrenze: 1 Minute

Am Bildschirm werden die Werte 10, 2, 1 angezeigt.

Im Teileprogramm ist **$A_MONIFACT = 2** programmiert (Uhr läuft schneller). Auf dem Monitor bleiben die Sollstandzeit, die Iststandzeit und Vorwarngrenze weiterhin in Echtzeit.

Sollstandzeit: 10 Minuten

Iststandzeit: 2 Minuten - aber in einer **halben Minute** wird die Vorwarngrenze erreicht, weil jetzt die Uhr zweimal schneller läuft.

Vorwarngrenze: 1 Minute

11.10.44 $AC_MONMIN - Faktor für die Werkzeugsuche

Allgemein

Über die Variable $AC_MOMIN wird folgendes festgelegt:

Betrachte nur die Werkzeuge, deren Istwert mindestens um einen Faktor $AC_MONMIN(0, ...1) des Sollwerts vom Grenzwert entfernt sind.

Definition kleinster/größter Istwert

Absolute kleinste/größte **Istwerte** werden bei der Werkzeug-Suche gemäß der Werkzeug-Such-Strategien "suche nach Werkzeug mit kleinstem/größtem Istwert" genau dann verwendet, wenn **alle Werkzeuge einer Werkzeuggruppe dieselbe Überwachungsart** (mittels $TC_TP9) definiert haben.

D.h. also alle Werkzeuge der Werkzeuggruppe sind entweder zeitüberwacht, oder stückzahlüberwacht, oder verschleiß- bzw. alternativ summenkorrekturüberwacht.

Relative kleinste/größte **Istwerte** werden bei der Werkzeug-Suche gemäß der WZ-Such-Strategien "suche nach WZ mit kleinstem/größtem Istwert" genau dann verwendet, wenn die **Werkzeuge einer Werkzeuggruppe unterschiedliche Werkzeug-Überwachungsarten** in $TC_TP9 definiert haben.

D.h. ein Werkzeug kann zeit-, das andere WZ kann stückzahlüberwacht sein. Ein drittes könnte sowohl Verschleiß- als auch zeitüberwacht sein.

Kleinster/größter Istwert bei genau einer Überwachungsart

Hierbei handelt es sich um die Standardanwendung.

Hier entspricht dem kleinsten/größten Istwert der Werkzeuge der Werkzeuggruppe jeweils der kleinste/größte Istwert der überwachten Größe ($TC_MOP2, $TC_MOP4, $TC_MOP6 für Zeit, Stückzahl, Verschleiß bzw. Summenkorrektur).

Beispiel:

Es ist eine Werkzeuggruppe "WZ1" definiert. Es gilt z.B. $TC_MAMP2="H108" - kleinster Istwert:

Duplonr. $TC_TP1	Ist-Wert $TC_MOP2	Soll-Wert $TC_MOP11	absolut kleinster Istwert = $TC_MOP2
1	9	10	9
2	8	10	8
3	6	6	6 kleinster Istwert in der Werkzeuggruppe

Damit wird die Reihenfolge der Werkzeuge für den Einsatz: Duplonr.= 3 → 2 → 1.

Kleinster/größter Istwert bei mehreren parallelen Überwachungsarten

Man kann Werkzeuge einer Werkzeuggruppe auf verschiedene Arten überwachen.

Oder es können für ein Werkzeug verschiedene Arten der Werkzeug-Überwachung festgelegt werden. Diese Situationen werden von NC erkannt und speziell behandelt:

Die Definition des kleinsten/größten Istwerts wird für diese Fälle ermittelt über den Quotienten der Division von Ist- und Sollwert; d.h.

Quotient (Q)= Istwert / Sollwert

Den **kleinsten Istwert** der Werkzeuge in der Werkzeuggruppe hat das Werkzeug mit dem kleinsten Quotienten.

Den **größten Istwert** der Werkzeuge in der Werkzeuggruppe hat das Werkzeug mit dem größten Quotienten.

Beispiel 1:

Werkzeuggruppe "Fraeser" hat zwei Werkzeuge mit T-Nr. =1 und 2 mit je einer Schneide D1.

T1 hat Zeitüberwachung aktiv; $TC_TP9[1]=1.

T2 hat Stückzahlüberwachung aktiv; $TC_TP9[2]=2.

Q(T1) = $TC_MOP2[1,1] / $TC_MOP11[1,1] ist = 0,5

Q(T2) = $TC_MOP4[2,1] / $TC_MOP13[2,1] ist = 0,9

D.h. T1 hat den kleineren Istwert.

Beispiel 2:

Es ist eine Werkzeuggruppe "WZ1" definiert. Es gilt z.B. $TC_MAMP2="H108" - kleinster Istwert:

Duplonr. $TC_TP1	Ist-Wert $TC_MOP2	Soll-Wert $TC_MOP11	absolut kleinster Istwert = $TC_MOP2
1	9	10	0.9
2	8	10	0.8 kleinster Istwert
3	6	6	1

Damit wird die Reihenfolge der Werkzeuge für den Einsatz: Duplonr.=2 → 1 → 3.

$AC_MONMIN

Für den Istwert, der hier gegen den mit dem Faktor $AC_MONMIN versehenen Sollwert geprüft wird, gilt die im obigen gemachte Definition bzgl. des Istwerts.

Beim absoluten Istwert-Vergleich wird geprüft (am Beispiel der Zeitüberwachung):

$TC_MOP2 ≥ $AC_MONMIN * $TC_MOP11.

Das ist das Kriterium für die Einsetzbarkeit des Werkzeugs.

Beim relativen Istwert-Vergleich wird geprüft (am Beispiel der Zeitüberwachung):

$TC_MOP2 / $TC_MOP11 ≤ $AC_MONMIN

Das ist das Kriterium für die Einsetzbarkeit des Werkzeugs.

Das Ergebnis ist jeweils dasselbe.

Hinweis

Der kleinste der Istwerte (sowohl absolut als auch relativ) der Schneiden eines Werkzeugs wird für den Vergleich mit den Istwerten der anderen Werkzeuge verwendet.

Name	**$AC_MONMIN**			
Bedeutung	Nur bei aktiver Werkzeugverwaltung Gibt den Faktor an für die Werkzeugsuchstrategie "Betrachte nur die Werkzeuge, deren Istwert mindestens $AC_MONMIN* Sollwert vom Grenzwert entfernt ist". Der programmierte Wert wird ignoriert, falls bei der Werkzeugsuche der Werkzeugzustand "gesperrt" ignoriert werden soll, was entweder durch Befehl TCA, PLC-Signal oder Maschinendaten für Start/Reset veranlasst werden kann. Siehe dazu auch die Systemvariablen $TC_MOPx, $TC_MAMP2.			
Datentyp	REAL			
Wertebereich	0-1			
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	X	X	X
Impliziter Vorlaufstopp	-	X	-	

Randbedingungen

Falls verschiedene Arten der Werkzeug-Überwachung für die Werkzeuge einer Werkzeuggruppe gewählt werden, muss entschieden werden, ob es für die konkrete Anwendung einen Sinn ergibt, etwa mit der Werkzeug-Suchstrategie "suche Werkzeug mit kleinstem bzw. größtem Istwert" in dieser Werkzeuggruppe zu arbeiten.

Ähnliches gilt, falls man mit mehrschneidigen Werkzeugen arbeiten will. Man muss sich überlegen, ob die Werkzeug-Suchstrategie "suche WZ mit kleinstem bzw. größtem Istwert" in dieser Werkzeuggruppe sinnvoll anzuwenden ist.

Hinweis

Ebenso wie bei den übrigen Werkzeug-Suchstrategien wird das einsatzfähige Werkzeug, das sich zum Zeitpunkt der Werkzeugsuche auf der Spindel oder in einem der zugeordneten Zwischenspeicher befindet, bevorzugt eingesetzt; d.h. die Werkzeug-Suchstrategie kommt nicht zum Tragen.

Das PLC-Signal "WZ nicht sperren" setzt die Werkzeug-Suchstrategie gemäß $AC_MONMIN außer Kraft.

Aktivierung

Damit die Werkzeug-Überwachungsspezifischen Werkzeug-Suchstrategien wirksam werden können, muss

- die Unterfunktion "Werkzeug-Überwachungsfunktion" innerhalb der Werkzeugverwaltungs-Funktion aktiv sein,
- müssen die Schneiden der Werkzeuge entsprechende Werte der Überwachungsparameter gesetzt haben ($TC_MOP1, ...)
- muss das entsprechende Werkzeug für die Überwachung aktiviert sein (Systemvariable $TC_TP9)
- $AC_MONMIN kann zusätzlich im Teileprogramm programmiert werden. Der programmierte Wert kommt sinnvoll zum Tragen, falls Punkte 1, 2, 3 erfüllt sind.

11.10.45 $P_TOOLNG - Anzahl Werkzeuggruppen

Diese Funktion ist für WZMO und WZMG verfügbar.

Name	**$P_TOOLNG**	
Bedeutung	Anzahl definierter Werkzeuggruppen, die dem Kanal zugeordnet sind. > 0: erfolgreicher Lesezugriff 0: keine Werkzeuggruppe definiert (Werkzeuggruppe wird durch Schreiben des Werkzeugnamens definiert) -1: weder Funktion WZMG noch WZMO aktiv	
Datentyp	INT	
Wertebereich	1-32000	
Indizes	Bedeutung	Wertebereich

Name	$P_TOOLNG			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

11.10.46 $A_MYMN / $A_MYMLN - Eigentümermagazin/platz des Werkzeugs

Diese Funktion ist für WZMG verfügbar.

In den Systemvariablen $A_TOOLMN und $A_TOOLMLN wurde definiert, auf welchem Magazin/Magazinplatz sich das dabei angegebene Werkzeug aktuell befindet. Dabei kann es sich sowohl um ein reales als auch ein internes Magazin handeln.

Die beiden Systemvariablen $A_MYMN und $A_MYMLN geben den Magazin/Magazinplatz an (nur reales Magazin), auf den das dabei angegebene Werkzeug beladen wurde bzw. von dem ein in einem internen Magazin befindliches Werkzeug eingewechselt wurde.

Name	$A_MYMN[t] / $A_MYMLN[t]			
Bedeutung	Verwendung: $A_MYMN[t] - Nummer des Eigentümermagazins des Werkzeugs mit der T-Nr. = t > 0: Werkzeug ist beladen 0: Werkzeug ist nicht beladen -1: Funktion WZMG ist nicht aktiv -2: Werkzeug mit der T-Nr. = t existiert nicht - auch für t=0 $A_MYMLN[t] - Nummer des Eigentümermagazinplatzes des Werkzeugs mit der T-Nr. = t > 0: Werkzeug ist beladen 0: Werkzeug ist nicht beladen -1: Funktion WZMG ist nicht aktiv -2: Werkzeug mit der T-Nr. = t existiert nicht - auch für t=0			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	X	-
Impliziter Vorlaufstopp	-	-	-	

1. Für nicht beladene Werkzeuge gilt:
   $A_MYMN = $A_MYMLN = $A_TOOLMN = $A_TOOLMLN = 0
2. Für eingewechselte Handwerkzeuge bzw. neu auf den Werkzeughalter beladene Werkzeuge gelten:
   $A_MYMN = $A_MYMLN = 0
   $A_TOOLMN 1 = 0, $A_TOOLMLN 1 = 0
3. Für Werkzeuge, die beladen und nicht in einem internen Magazin sind, gilt:
   $A_MYMN = $A_TOOLMN > 0
   $A_MYMLN = $A_TOOLMLN >0
   Für festplatzcodierte Werkzeuge in Zwischenspeichern geben die beiden Parameter an, wohin das jeweilige Werkzeug beim Rücktransport in das Magazin gebracht wird.

11.10.47 $A_MYMTN / $A_MYMTLN - $A_TOOLMTN / $A_TOOLMTLN - Werkzeuge im Multitool

Die Systemfunktionen `$A_MYMTN / $A_MYMTLN` bzw. inhaltsgleich `$A_TOOLMTN / $A_TOOLMTLN` geben an, in welchem Multitool und auf welchem Platz im Multitool das Werkzeug sich befindet.

Es gilt:

```
$A_MYMTN = $A_TOOLMTN
```

und

```
$A_MYMTLN = $A_TOOLMTLN.
```

Damit kommt zum Ausdruck, dass das Werkzeug fix im MT verbleibt.

Syntax

```
result = $A_MYMTN [t]
result = $A_TOOLMTN [t]
result = $A_MYMTLN [t]
result = $A_TOOLMTLN [t]
```

Bedeutung

`$A_MYMTN` `$A_TOOLMTN`	Liefert die Multitoolnummer des Eigentümermultitools des Werkzeugs mit der T-Nummer t.	
	Datentyp:	INT
t	T-Nummer des Werkzeugs	
	Datentyp:	INT
	WZ-Tnr. t = 1, ...,32000 `MYMultiToolNumber`	
	WZ-Tnr. t = 1, ..., 32000 `TOOLisOnMultiToolNumber`	
Rückgabewert		

result	Ergebniswert	
	Datentyp:	INT
	>0	Das WZ mit der T-Nummer t befindet sich Multitool mit der Nummer result
	0	Das WZ mit der Tnr.= t ist nicht in einem Multitool
	-1	Funktion WZMG nicht aktiv
	-2	Funktion Multitool nicht aktiv
	-3	ist keine WZ-T-Nr. (außerhalb des Bereichs, nicht definiert, oder es ist eine MT-Nr. oder eine Magazin-Nr.)
$A_MYMTLN $A_TOOLMTLN	Multitoolnummer von Werkzeug t.	
	Datentyp	INT
t	T-Nummer des Werkzeuges	
	Datentyp:	INT
	WZ-Tnr. t = 1, ...,32000 MYMultiToolLoactionNumber	
	WZ-Tnr. t = 1, ...,32000 TOOLisOnMultiToolLoactionNumber	
Rückgabewert		
result	Ergebniswert	
	>0	Das WZ mit der T-Nummer t befindet sich auf dem Platz mit der Nummer result innerhalb eines Multitools
	0	Das WZ mit der T-Nummer = t ist nicht in einem Multitool
	-1	Funktion WZMG nicht aktiv
	-2	Funktion Multitool nicht aktiv
	-3	ist keine WZ–T–Nr. (außerhalb des Bereichs, nicht definiert, oder es ist eine MT-Nr. oder eine Magazin-Nr.)

11.10.48 $P_TOOLNT / $P_TOOLT - T-Nummern

Diese Funktion ist für WZMG und WZMO verfügbar.

Name	**$P_TOOLNT / $P_TOOLT[i]**	
Bedeutung	Diese Systemvariablen ermöglichen einen Überblick über die in NC definierten Werkzeuge. $P_TOOLNT Anzahl definierter Werkzeuge, die dem Kanal zugeordnet sind. > 0:erfolgreicher Lesezugriff 0: kein Werkzeug definiert $P_TOOLT[i] i-te Werkzeugnummer T > 0: T-Nummer 0: i ist ein Wert außerhalb des erlaubten Bereichs	
Datentyp	INT	
Wertebereich	1-32000	
Indizes	Bedeutung	Wertebereich

Name	**$P_TOOLNT / $P_TOOLT[i]**			
	N = Anzahl der Werkzeuge, die dem Kanal zugeordnet sind i = i-te T-Nr.; i=1, ..., $P_TOOLNT			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Die Systemvariable $P_TOOLT liefert für den Index i=1 den Wert 1, falls mindestens eine D-Korrektur definiert ist, für andere Werte von i den Wert 0.

11.10.49 $P_TOOLD - D-Nummern

Diese Funktion ist für WZMG und WZMO verfügbar.

Name	**$P_TOOLND / $P_TOOLD[t,i]**			
Bedeutung	Ermittlung der definierten D-Nummern eines Werkzeugs. Der Befehl kann generell programmiert werden. i-te Werkzeug-Korrekturnummer D des Werkzeugs mit der T-Nr. = t > 0: D-Nummer 0: i ist ein Wert außerhalb des erlaubten Bereichs -2: t ist der Wert eines nicht definierten Werkzeugs			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	t = T-Nummer i = i-te T-Nr.; i=1, ..., $P_TOOLND			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

11.10.50 $P_TOOLNDL - Anzahl definierter DL-Korrekturen

Diese Funktion ist für WZMG und WZMO verfügbar. Die Funktion "Summenkorrektur" muss per MD aktiviert worden sein.

Name	**$P_TOOLNDL[t,d]**			
Bedeutung	Ermittlung der Anzahl der definierten DL-Nummern einer D-Korrektur. Der Befehl kann generell programmiert werden. Anzahl von DL-Korrekturen der D-Korrektur gegeben durch T-Nr. = t, D-Nr. = d > 0: Anzahl DL-Korrekturen 0: keine DL-Korrektur für diese D-Korrektur -1: Summenkorrekturfunktion nicht aktiv -2: t ist der Wert eines nicht definierten Werkzeugs -3: d ist der Wert einer nicht definierten D-Korrektur			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	t = T-Nummer d = D-Nummer			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

11.10.51 $A_USEDND - Stückzahlzählung

Diese Funktion ist für WZMO verfügbar.

Name	**$A_USEDND[s]**			
Bedeutung	Anzahl, der seit der letzten Stückzahlzählung auf Werkzeughalter s benutzten verschiedenen Schneiden, inklusive der momentan auf s aktiven Schneide. Jedes eingesetzte Werkzeug ist mindestens einmal enthalten. Der Index s bedeutet: *WZMG + WZMO* Spindelnummer/Werkzeug-Halternummer s = 0 bedeutet, dass der momentan aktive Master-Werkzeughalter ausgewählt wird. *WZMO ohne WZMG aktiv* a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s wird nicht ausgewertet. Es ist nicht möglich, die Werkstückzählung getrennt nach Werkzeughaltern vorzunehmen. b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: Spindelnummer. s=0 bedeutet, dass die momentan aktive Master-Spindel ausgewählt wird. > 0: Anzahl der eingesetzten Schneiden 0: seit letzter Stückzahlzählung keine Werkzeuge mehr verwendet -1: WZMO ist nicht aktiv -2: s ist der Wert eines nicht definierten Werkzeughalters			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	s = 1, ..., MAXNUM_AXES_PER_CHAN s = 0 bezeichnet den Master-Werkzeughalter			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel siehe "$A_USEDT - Stückzahlzählung (Seite 243)".

11.10.52 $A_USEDT - Stückzahlzählung

Übersicht

Diese Funktion ist für WZMO verfügbar.

Name	**$A_USEDT[i,s]**			
Bedeutung	T-Nummer des Werkzeugs der i-ten Schneide, die seit der letzten Stückzahlzählung auf Werkzeughalter s zum Einsatz gekommen ist bzw. noch im Einsatz ist. Beispiel: Für i=$A_USEDND erhält man die T-Nummer zur ersten Schneide bzw. D-Korrektur, die nach der letzten Stückzahlzählung auf dem programmierten WZ-Halter s angewählt wurde. Der Index s bedeutet: *WZMG + WZMO* Spindelnummer/Werkzeug-Halternummer s = 0 bedeutet, dass der momentan aktive Master-Werkzeughalter ausgewählt wird. *WZMO ohne WZMG aktiv* a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s wird nicht ausgewertet. Es ist nicht möglich, die Werkstückzählung getrennt nach Werkzeughaltern vorzunehmen. b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: Spindelnummer. s=0 bedeutet, dass die momentan aktive Master-Spindel ausgewählt wird. >0: T-Nummer (kann auch mehrfach enthalten sein, falls verschiedene D-Korrekturen des Werkzeugs im Einsatz waren 0: seit letzter Stückzahlzählung keine Schneiden mehr verwendet -1: WZMO ist nicht aktiv -2: s ist der Wert eines nicht definierten Werkzeughalters			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	S = 1, ..., MAXNUM_AXES_PER_CHAN I = 1, ..., $A_USEDND			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel

Es sind zwei Werkzeughalter mit den Nummern 1 und 2 definiert. Werkzeug-Halternummer 1 ist der Master-Werkzeughalter. Auf Werkzeughalter 1 kamen bisher die 3 Werkzeuge mit den T-Nummern 10, 20, 30 zum Einsatz; auf Werkzeughalter 2 ein Werkzeug mit der T-Nummer 666. Jedes Werkzeug hat nur die Korrekturen D1 definiert.

In dem Zustand läuft folgender Programmteil:

```
def int n1, n2, i, tNo
n1 = $A_USEDND[1]          ;n1 = 3 inhaltsgleich wäre gewesen
                           $A_USEDND[0]
n2 = $A_USEDND[2]          ;n2 = 1
for i = 1 to n1
tNo = $A_USEDT[1,i]
MSG ("an Werkstückbearbeitung beteiligte T-Nr. =" << tNo
endfor
                           ;die Schleife zeigt die T-Nummer 10, 20,
                           30 an
T2=0                       ;in $MC_TOOL_MANAGEMENT_MASK seien die Bits
                           7, 8, 19 zur Synchronisation gesetzt.
                           (Automatische Einlesesperre bis
                           Werkzeugwechsel mit "Ende" quittiert ist.)
setpiece(5,2)
if (n2 == 1) tNo = $A_USEDT[1,1]
                           ;setzt tNo auf den Wert 0. Seit Ermittlung
                           von n2 wurde setpiece programmiert. Damit
                           wurde die Liste der eingesetzten Werkzeuge
                           gelöscht und zum genannten Index1 gibt es
                           momentan keinen Eintrag in der Liste der
                           eingesetzten Werkzeuge.
```

11.10.53 $A_USEDD - Stückzahlzählung

Diese Funktion ist für WZMO verfügbar.

Name	**$A_USEDD[i,s]**			
Bedeutung	D-Nummer der i-ten Schneide, die seit der letzten Stückzahlzählung auf Werkzeughalter s zum Einsatz gekommen ist bzw. noch im Einsatz ist. Beispiel: Für i=$A_USEDND erhält man die D-Nummer zur ersten Schneide bzw. D-Korrektur, die nach der letzten Stückzahlzählung auf dem programmierten WZ-Halter s angewählt wurde. Der Index s bedeutet: *WZMG + WZMO* Spindelnummer/Werkzeug-Halternummer s = 0 bedeutet, dass der momentan aktive Master-Werkzeughalter ausgewählt wird. *WZMO ohne WZMG aktiv* a) $MC_T_M_ADDRESS_EXT_IS_SPINO = FALSE: s wird nicht ausgewertet. Es ist nicht möglich, die Werkstückzählung getrennt nach Werkzeughaltern vorzunehmen. b) $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE: Spindelnummer. s=0 bedeutet, dass die momentan aktive Master-Spindel ausgewählt wird. >0: D-Nummer 0: seit letzter Stückzahlzählung keine Werkzeuge mehr verwendet -1: WZMO ist nicht aktiv -2: s ist der Wert eines nicht definierten Werkzeughalters			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	S = 1, ..., MAXNUM_AXES_PER_CHAN I = 1, ..., $A_USEDND			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

11.10.54 $P_MAGN / $P_MAG - Magazine

Diese Funktion ist für WZMG verfügbar.

Name	**$P_MAGN / $P_MAG[i]**			
Bedeutung	$P_MAGN Anzahl definierter Magazine, die dem Kanal zugeordnet sind. > 0: erfolgreicher Lesezugriff 0: keine Magazine definiert -1:WZMG ist nicht aktiv $P_MAG i-te Magazinnummer > 0: erfolgreicher Lesezugriff 0: i ist außerhalb des erlaubten Bereichs -1: WZMG ist nicht aktiv			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	i = 1, ..., $P_MAGN			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel siehe "Beispiel zu den Magazinkonfigurations-Systemvariablen (Seite 250)".

11.10.55 $P_MAGNDIS / $P_MAGDISS / $P_MAGDISL - Magazindistanztabellen

Diese Funktion ist für WZMG verfügbar.

Name	**$P_MAGNDIS[n,m] / $P_MAGDISS[l,i] / $P_MAGDISL[l,i]**			
Bedeutung	$P_MAGNDIS**[n,m]** Anzahl der Magazine, die mit dem Platz m des internen Magazins n verbunden sind > 0: erfolgreicher Lesezugriff 0: kein Magazin ist mit dem Zwischenspeicherplatz verbunden -1: WZMG ist nicht aktiv -2: n ist nicht die Nummer eines internen Magazins -3: m ist nicht die Nummer eines internen Magazinplatzes $P_MAGDISS[l,i] Nummer des i-ten Magazins, das mit dem Platz l des Zwischenspeichermagazins verbunden ist. > 0: erfolgreicher Lesezugriff 0: i ist außerhalb des erlaubten Bereichs -1: WZMG ist nicht aktiv -2: m ist nicht die Nummer eines Zwischenspeicherplatzes -3: kein Zwischenspeicherplatz definiert $P_MAGDISL[l,i] Nummer des i-ten Magazins, das mit dem Platz l des Beladem agazins verbunden ist. > 0: erfolgreicher Lesezugriff 0: i ist außerhalb des erlaubten Bereichs -1: WZMG ist nicht aktiv -2: m ist nicht die Nummer eines Beladem agazinplatzes -3: kein Zwischenspeicherplatz definiert			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	i = 1, ..., $P_MAGNDIS			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiele siehe "Beispiel zu den Magazinkonfigurations-Systemvariablen (Seite 250)".

11.10.56 $P_MAGNS / $P_MAGS - Werkzeughalter

Diese Funktion ist für WZMG verfügbar.

Name	**$P_MAGNS / $P_MAGS[n]**			
Bedeutung	$P_MAGNS Anzahl der Spindelplätze / Werkzeug-Halterplätze im Zwischenspeicher, der dem Kanal zugeordnet ist. > 0: erfolgreicher Lesezugriff 0: keine Spindelplätze definiert -1: WZMG ist nicht aktiv -2: kein Zwischenspeichermagazin definiert $P_MAGS[n] n-te Nummer der Spindel / des Werkzeughalters im Zwischenspeicher > 0: erfolgreicher Lesezugriff 0: n ist außerhalb des erlaubten Bereichs -1: WZMG ist nicht aktiv -3: kein Zwischenspeichermagazin definiert			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	n = 1, ..., max. Werkzeughalternummer			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel siehe "Beispiel zu den Magazinkonfigurations-Systemvariablen (Seite 250)".

11.10.57 $P_MAGNREL / $P_MAGREL - zugeordnete Zwischenspeicher

Diese Funktion ist für WZMG verfügbar.

Name	**$P_MAGNREL[n] / $P_MAGREL[n,m]**			
Bedeutung	$P_MAGNREL[n] Anzahl der der Spindelnr./Werkzeug-Halternr. zugeordneten Zwischenspeicher > 0: erfolgreicher Lesezugriff 0: Spindelplatz hat keinen Zwischenspeicherplatz zugeordnet -1: WZMG ist nicht aktiv -2: n ist nicht die Nummer eines Spindelplatzes -3: kein Zwischenspeichermagazin definiert $P_MAGREL[n,m] m-te Zwischenspeichernummer der n-ten Spindelnummer / Werkzeughalternummer > 0: erfolgreicher Lesezugriff 0: m ist außerhalb des erlaubten Bereichs -1: WZMG ist nicht aktiv -2: n ist nicht die Nummer eines Spindelplatzes -3: kein Zwischenspeichermagazin definiert			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	m = 1, ..., $P_MAGNREL n = 1, ..., max. Werkzeughalternummer			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel siehe "Beispiel zu den Magazinkonfigurations-Systemvariablen (Seite 250)".

11.10.58 Beispiel zu den Magazinkonfigurations-Systemvariablen

Vorgaben

Die im folgenden Beispiel gewählte Magazinkonfiguration ist gewählt. Mit dem Lesen der hier beschriebenen Systemvariablen kann man sich die Information über die aktuelle Magazinkonfiguration beschaffen.

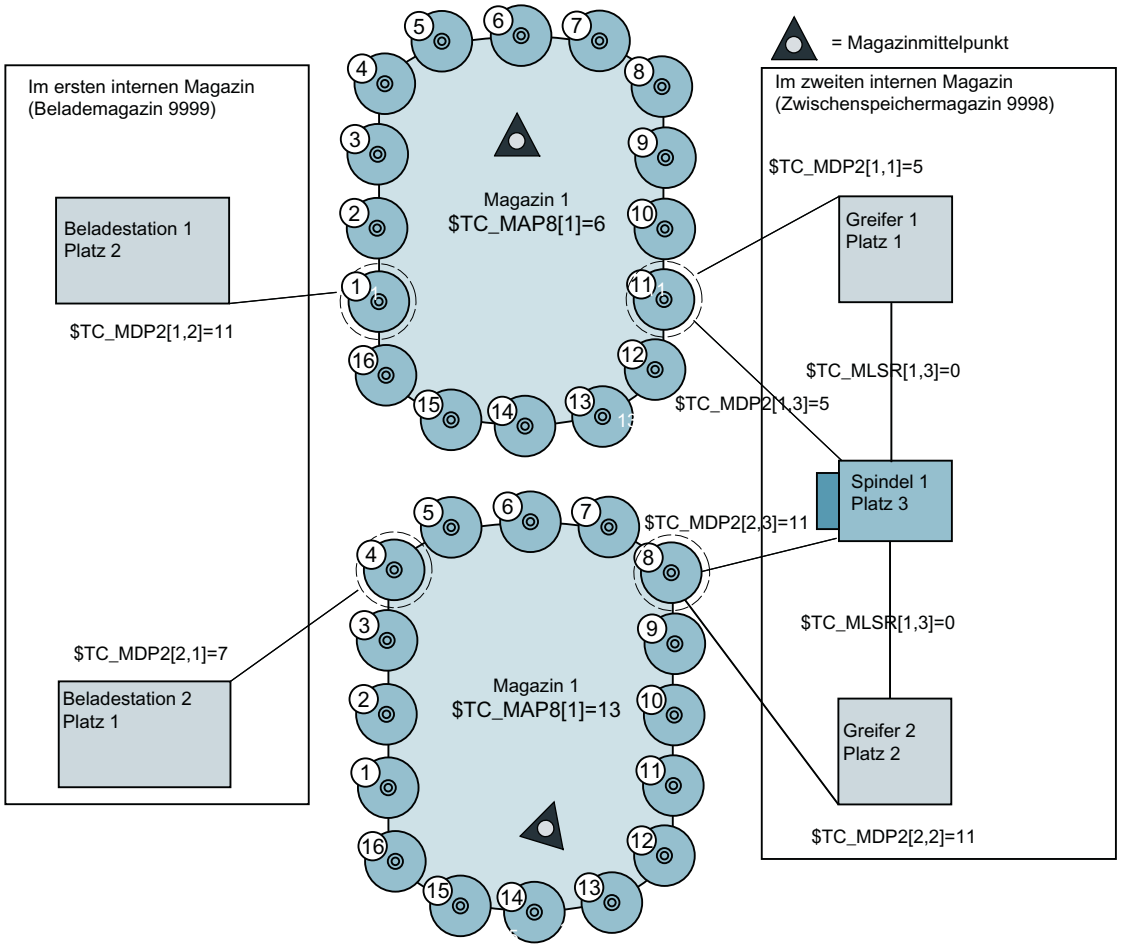


Bild 11-13 Magazinkonfiguration

```
N10 def int noOfMag=0, noOfLoc=0, noOfDist=0, noOfRel=0,
noOfSpindles=0, spindeleNo=0
N20 def int i=0
; Gesamtzahl der definierten Magazine

N100 noOfMag = $P_MAGN     ;noOfMag erhält den Wert =4 - 2reale
                           Magazine
                           ;1, 2+2 interne Magazine 9998, 9999
```

; zeige alle Magazinnummern an

```
N200 for i=1 to noOfMag
N220 MDG ("Magazinnr.="<<$P_MAG[i])
                          ;zeige die Nummern 1, 2, 9998, 9999 an
N240 endfor
```

; Gesamtanzahl der definierten Magazinplätze

```
N300 for i=1 to noOfMag
N320 noOfLoc=noOfLoc + $TC_MAP7[$P_MAG[i]]
N340 endfor               ;noOfLoc enthält nun den Wert 16+16+3+2=37
```

; Anzahl der mit Spindel1 verbundenen Magazine

```
N400 noOfDist=$P_MAGNDIS[9998,3]
                          ;; noOfDist erhält den Wert=2 - Mag.1, 2 sind mit dem Spindelplatz verbunden
```

; zeige die mit Spindel 1 (=Platz 3) verbundenen Magazinnummern

```
N500 for i=1 to noOfDist
N520 MSG ("Magazinnr.=" << $P_MAGDISS[ i ] )
                          ;zeige die Nummern 1, 2 an
N540 endfor
```

; Anzahl der mit Beladestation 2 verbundenen Magazine

```
N410 noOfDist = $P_MAGNDIS[9999,1]
                          ;noOfDist erhält den Wert=1 - Mag. 2 ist
                          mit der Beladestation 2 verbunden
```

; zeige die mit Beladestation 2 (=Platz 1) verbundenen Magazinnummern

```
N510 for i=1 to noOfDist
N530 MSG ("Magazinnr.="<< $P_MAGDISL[i] )
                          ;zeige die Nummer 2 an
N550 endfor
```

; Gesamtanzahl der definierten Spindeln

```
N600 noOfSpindles=        ;noOfSpindles enthält den Wert = 1
$P_MAGNS                  ;- es ist ein Spindelplatz definiert
```

; zeige die Nummern der in der Magazinkonfiguration definierten Spindeln an

```
N620 for i=1 to noOfSpindles
N640 MSG ("Magazinnr.="<< $P_MAGS[i])
                          ;zeige die Nummer 1 an
N660 endfor
```

; Gesamtanzahl der der Spindel 1 zugeordneten Zwischenspeicherplätze

(=Greifer im Beispiel)

```
N700 noOfRel=$P_MAGNREL[1]
                          ;noOfRel enthält den Wert=2 der Spindel
                          sind die Greifer 1 und 2 zugeordnet
```

; zeige die Nummern der in der Magazinkonfiguration definierten Greifer der Spindelnr. 1 an

```
N720 for i=1 to noOfRel
N740 MSG ("Magazinnr.="<< $P_MAGREL[1,i] )
                          ;zeige die Nummern 1, 2 an
N760 endfor
```

11.10.59 $P_MAGNH / $P_MAGNHLT / $P_MAGHLT - Platztyphierarchien

Übersicht

Diese Funktion ist für WZMG verfügbar.

Die Systemvariablen liefern folgende Informationen:

- Anzahl der Hierarchien
- Anzahl der Magazinplatztypen in jeder Hierarchie

Name	**$P_MAGNH / $P_MAGNHLT[n] / $P_MAGHLTn,m]**
Bedeutung	$P_MAGNH Anzahl definierter Magazinplatztyp-Hierarchien, die dem Kanal zugeordnet sind. > 0; erfolgreicher Lesezugriff 0; es sind keine Platztyp-Hierarchien definiert. -1; WZMG ist nicht aktiv $P_MAGNHLT[n] Anzahl der definierten Platztypen in der n-ten definierten Hierarchie > 0; erfolgreicher Lesezugriff 0; die Hierarchie "n" ist außerhalb des definierten Bereichs. -1; WZMG ist nicht aktiv. -2; n ist außerhalb des definierten Bereichs. $P_MAGHLTn,m] m-ter Platztyp der Hierarchie n > 0; erfolgreicher Lesezugriff -1; WZMG ist nicht aktiv. -2; die Hierarchie n hat keine definierten Platztypen. -3; m ist außerhalb des definierten Bereichs.
Datentyp	INT
Wertebereich	1-32000

Name	$P_MAGNH / $P_MAGNHLT[n] / $P_MAGHLTn,m]			
Indizes	Bedeutung			Wertebereich
	n = 1, ..., $P_MAGNH m = 1, ..., $P_MAGNHLT			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Beispiel

Es sind folgende drei Hierarchien definiert:

Hierarchie 1:

```
$TC_MPTH[0,0] = 12
$TC_MPTH[0,1] = 88
$TC_MPTH[0,2] = 9999
$TC_MPTH[0,3] = 9999
$TC_MPTH[0,4] = 9999
```

Hierarchie 2:

```
$TC_MPTH[1,0] = 55
$TC_MPTH[1,1] = 808
$TC_MPTH[1,2] = 45
$TC_MPTH[1,3] = 9999
$TC_MPTH[1,4] = 9999
```

Hierarchie 3:

```
$TC_MPTH[2,0] = 87
$TC_MPTH[2,1] = 21
$TC_MPTH[2,2] = 3
$TC_MPTH[2,3] = 9999
$TC_MPTH[2,4] = 62
```

Hierarchie 4:

```
$TC_MPTH[3,0] = 9999
$TC_MPTH[3,1] = 9999
$TC_MPTH[3,2] = 9999
$TC_MPTH[3,3] = 9999
$TC_MPTH[3,4] = 9999
```

Hierarchie 5:

```
$TC_MPTH[4,0] = 99
$TC_MPTH[4,1] = 124
$TC_MPTH[4,2] = 32
$TC_MPTH[4,3] = 7
$TC_MPTH[4,4] = 9999
```

Das folgende Beispiel liefert, *wie viele Hierarchien* insgesamt definiert sind und *wie viele Magazinplatztypen in jeder Hierarchie* enthalten sind.

```
N10 def int noOfH, noOfTypes[8], locTypeNo
N100 noOfH=$P_MAGNH                ;noOfH erhält den Wert = 5
N220 for i=1 to noOfH
N240 noOfTypes[i - 1]=$P_MAGNHLT[ i ]
                                   ;setzt im Array die Werte 2, 3,
                                   5, 0 ,4
N260 endfor
```

Das folgende Beispiel liefert, *welche Magazinplatztypen* in der 2. Hierarchie definiert sind.

```
N310 for i=1 to noOfTypes[1]
N340 MSG ("Magazinnr.="<<$P_MAGHLT[2, i])
                                   ;zeigt die Werte 55, 808, 45
N330 endfor
```

Das folgende Beispiel liefert, welche Platztypen in der 3. Hierarchie definiert sind.

```
N350 for i=1 to
noOfTypes[2]
N360 MSG ("Magazinnr.="<<$P_MAGHLT[3, i])
                        ;zeigt die Werte 87, 21, 3, 9999, 62
N370 endfor
```

Das folgende Beispiel liefert, welche Platztypen in der 5. Hierarchie definiert sind.

```
N400 for i=1 to noOfTypes[4]
N410 MSG ("Magazinnr.="<<$P_MAGHLT[5, i])
                                   ;zeigt die Werte 99, 124, 32, 7
N420 endfor
```

Das folgende Beispiel liefert den Platztyp einer nicht definierten Hierarchiestufe (über das Ende von Hierarchie 2 hinaus).

```
N500 MSG ( " Magazinnr. = " << $P_MAGHLT[ 2, 4 ] )
                            ; zeigt den Wert –3
```

11.10.60 $P_MAGNA / $P_MAGA - Werkzeug-Adapter

Diese Funktion ist für WZMG verfügbar.

Name	$P_MAGNA / $P_MAGA[i]			
Bedeutung	$P_MAGNA Anzahl definierter Adapter, die dem Kanal zugeordnet sind. > 0 erfolgreicher Lesezugriff 0 keine Adapter definiert -1 Funktion "Adapter" bzw. WZMG ist nicht aktiv $P_MAGA[i] i-te Adapternummer > 0 erfolgreicher Lesezugriff 0 i ist außerhalb des definierten Bereichs -1 Funktion "Adapter" bzw. WZMG ist nicht aktiv			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	i = 1, ..., $P_MAGNA			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

$P_MAGNA1, $P_MAGNA2

Name	$P_MAGNA1			
Bedeutung	$P_MAGNA1 Anzahl der definierten Adaptern vom Typ 1 in der TO-Einheit des Kanals. • >0 erfolgreicher Lesezugriff • 0 keine Adapter vom Typ 1 definiert • -1 Funktion 'Adapter' bzw. WZMG ist nicht aktiv BTSS-Baustein= T/TMV			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	i = 1, ..., $P_MAGNA1			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_MAGNA 2**			
Bedeutung	$P_MAGNA2 Anzahl der definierten Adaptern vom Typ 2 in der TO-Einheit des Kanals. • >0 erfolgreicher Lesezugriff • 0 keine Adapter vom Typ 2 definiert • -1 Funktion 'Adapter' bzw. WZMG ist nicht aktiv BTSS-Baustein= T/TMV			
Datentyp	INT			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
	i = 1, ..., $P_MAGNA2			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

11.10.61 Weitere Sprachbefehle

Name	**$P_TOOLNO**			
Bedeutung	Aktive WZ-Nummer T0 bis T32000. Der Befehl sollte allgemein bei aktiver Magazinverwaltung nicht verwendet werden. Bei aktiver Magazinverwaltung sollte stattdessen GETTEXET verwendet werden. Für $MC_CUTTING_EDGE_DEFAULT=0 kann die falsche T-Nummer ermittelt werden. Falls die Programmierung nach einer Programmierung D > 0 erfolgt, ist sie ebenfalls sicher			
Datentyp	Integer			
Wertebereich	1-32000			
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TOOLP**		
Bedeutung	Zuletzt programmierte WZ-Nummer Befehl ist bei WZMO verfügbar. Er ist analog zum WZMG-spezifischen Befehl GETSELT.		
Datentyp	Integer		
Wertebereich	1-32000		
Indizes	Bedeutung		Wertebereich

Name	**$P_TOOLP**			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TOOL**			
Bedeutung	Aktive Werkzeugschneide (Dx)			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_DLNO**			
Bedeutung	Aktive Summenkorrekturnummer DL=0-DL=max; max=Wert von $MN_MM_MAX_SUMCORR_PER_CUTTEDGE $P_DLNO ist analog den bereits bestehenden Parametern $P_TOOL, $P_TOOLNP - den aktiven D- und T-Nummern.			
Datentyp	Integer			
Wertebereich	0-6			
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-		

Name	**$P_TOOLL[n]**			
Bedeutung	Aktive WZ Gesamtlänge; n = 1...3 `$P_TOOLL[1]` bis `$P_TOOLL[3]` entsprechen den Werten in `$TC_DP3[]` bis `$TC_DP3[]` inklusiv aktiven Werkzeugkorrekturen unabhängig von der Ebenenanwahl und unabhängig von Settingdaten wie `$SC_TOOL_LENGTH_CONST` u.a.			
Datentyp	REAL			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.

Name	**$P_TOOLL[n]**			
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TOOLR**			
Bedeutung	Aktiver Radius			
Datentyp	REAL			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TC**			
Bedeutung	Aktiver Werkzeugträger			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TCANG[n]**			
Bedeutung	Aktiver Winkel einer WZ-Trägerachse; n = 1-2			
Datentyp	REAL			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_TCDIFF[n]**
Bedeutung	Differenz zwischen berechnetem und verwendetem Winkel einer WZ-Trägerachse bei Rasterung (Hirth-Verzahnung) des Winkels
Datentyp	REAL
Wertebereich	

Name	**$P_TCDIFF[n]**			
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_AD[n]**			
Bedeutung	Aktive WZ-Korrektur; n = 1...31 n=1-25 $TC_DP1 bis $TC_DP25 n=26 $TC_DPCE (optional) n=27 $TC_DPH (optional) n=28 $TC_DPV (optional) n=29 $TC_DPV3 (optional) n=30 $TC_DPV4 (optional) n=31 $TC_DPV5 (optional)			
Datentyp	DOUBLE			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	X	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$P_ADT[n]**			
Bedeutung	$P_ADT[n] - transformierte Daten des aktiven Werkzeugs Liefert beim Lesen der Korrekturparameter transformierte Werte der der Werkzeug-Adapter-Transformation unterliegenden Parameter, falls sich das aktive Werkzeug auf einem Werkzeug-Adapter befindet. n=1-25 $TC_DP1 bis $TC_DP25 n=26 $TC_DPCE (optional) n=27 $TC_DPH (optional) n=28 $TC_DPV (optional) n=29 $TC_DPV3 (optional) n=30 $TC_DPV4 (optional) n=31 $TC_DPV5 (optional)			
Datentyp	DOUBLE			
Wertebereich				
Indizes	Bedeutung			Wertebereich
	n: Parameternummer 1 bis 31			
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.

Name	**$P_ADT[n]**			
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$AC_MSNUM**			
Bedeutung	Masterspindel, Rückgabewert 0: keine Spindel vorhanden 1...n: Nummer der Masterspindel			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	X	-
Impliziter Vorlaufstopp	X	-	-	

Name	**$P_MSNUM**			
Bedeutung	Masterspindel 0: keine Spindel vorhanden 1...n: Nummer der Masterspindel			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	

Name	**$AC_MTHNUM**			
Bedeutung	Master Toolholder Wert=0 kein Master-Werkzeughalter definiert Wert>0 Nummer des Master-Werkzeughalters			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	X	-
Impliziter Vorlaufstopp	X	-	X	-

Name	**$P_MTHNUM**			
Bedeutung	Master Toolholder Wert=0 kein Master-Werkzeughalter definiert Wert>0 Nummer des Master-Werkzeughalters			
Datentyp	Integer			
Wertebereich				
Indizes	Bedeutung			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-		

Name	**$TC_TP_MAX_VELO**			
Bedeutung	Maximale Drehzahl des Werkzeugs			
Datentyp	REAL			
Wertebereich	[0, DBL_MAX]			
Indizes	Interne T-Nummer			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	X	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$TC_TP_MAX_ACC**			
Bedeutung	Maximale Drehbeschleunigung des Werkzeugs			
Datentyp	REAL			
Wertebereich	[0, DBL_MAX]			
Indizes	Interne T-Nummer			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	X	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$TC_MPP_SP_MAX_ACC**			
Bedeutung	Nur relevant, wenn $MC_TOOLHOLDER_MANAGEMENT>0: Spindel-Nr., die mit dem Werkzeughalter verbunden ist.			
Datentyp	Integer			
Wertebereich	[0, MAXNUM_SPIND_PER_CHAN]			
Indizes	Magazin-Nr., Magazin-Platz-Nr.			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.

Name	**$TC_MPP_SP_MAX_ACC**			
	X	X	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$P_MTHSDC**			
Bedeutung	Master-WZ-Halternr. bzw. Masterspindelnr. bzgl. derer das aktive WZ für die nächste D-Korrekturanwahl bestimmt wird. > 0 erfolgreicher Lesezugriff 0 Kein Master-WZ-Halter bzw. keine Masterspindel verfügbar. Die nächste D-Korrektur arbeitet mit T0. -1 WZMG nicht verfügbar			
Datentyp	Integer			
Wertebereich	[-1, MAXNUM_AXES_PER_CHAN]			
Indizes	Magazin-Nr., Magazin-Platz-Nr.			Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$P_TH_OF_D**			
Bedeutung	Werkzeughalter bzw. Spindel, auf der das aktive Werkzeug sitzt, das die aktive D-Korrektur enthält. > 0 erfolgreicher Lesezugriff 0 Kein Werkzeughalter bzw. Spindel als Bezug verfügbar, weil z. B. keine D-Korrektur aktiv ist. -1 Funktion ist nicht verfügbar Wenn als BTSS-Variable gelesen, gilt dies für den Zustand im aktuellen Hauptlauf-Satz			
Datentyp	Integer			
Wertebereich	[-1, MAXNUM_AXES_PER_CHAN]			
Indizes				Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$P_MTHNUM_BEFORE_SEARCH**			
Bedeutung	Master-Toolholder bzw. Masterspindel bevor der Satzsuchlauf bzw. Testbetrieb begonnen wurde. >0: erfolgreicher Lesezugriff 0: Kein Werkzeughalter bzw. Spindel als Bezug verfügbar. weil z. B. keine D-Korrektur aktiv ist. -1: Funktion ist nicht verfügbar Wenn der Satzsuchlauf bzw. Testbetrieb beendet ist, enthält diese Variable ab der nächsten D-Programmierung den gleichen Wert wie $P_MTHNUM. Wenn als BTSS-Variable gelesen, gilt dies für den Zustand im aktuellen Hauptlauf-Satz.			
Datentyp	Integer			
Wertebereich	[-1, MAXNUM_AXES_PER_CHAN]			
Indizes				Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$P_D_BEFORE_SEARCH**			
Bedeutung	Die aktive D-Korrektur bevor der Satzsuchlauf bzw. Testbetrieb begonnen wurde. >0: erfolgreicher Lesezugriff 0: Kein Werkzeughalter bzw. Spindel als Bezug verfügbar, weil z. B. keine D-Korrektur aktiv war bzw. ist. -1: Funktion ist nicht verfügbar Wenn der Satzsuchlauf bzw. Testbetrieb beendet ist, enthält diese Variable ab der nächsten D-Programmierung den gleichen Wert wie $P_TOOL. Wenn als BTSS-Varaible gelesen, gilt dies für den Zustand im aktuellen Hauptlauf-Satz.			
Datentyp	Integer			
Wertebereich	[-1, 32000]			
Indizes				Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	-

Name	**$P_DL_BEFORE_SEARCH**			
Bedeutung	Die aktive DL-Korrektur bevor der Satzsuchlauf bzw. Testbetrieb begonnen wurde. >0: erfolgreicher Lesezugriff 0: Kein Werkzeughalter bzw. Spindel als Bezug verfügbar, weil z. B. keine D-Korrektur aktiv war bzw. ist -1: Funktion ist nicht verfügbar Wenn der Satzsuchlauf bzw. Testbetrieb beendet ist, enthält diese Variable ab der nächsten D- bzw. DL-Programmierung den gleichen Wert wie $P_DLNO. Wenn als BTSS-Varaible gelesen, gilt dies für den Zustand im aktuellen Hauptlauf-Satz.			
Datentyp	Integer			
Wertebereich	[-1, 6]			
Indizes				Wertebereich
Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronakt.
	X	-	-	-
Impliziter Vorlaufstopp	-	-	-	-

11.10.62 Variablen für Unterprogrammersetzungstechnik

WZV-Sprachbefehl	**Funktionen**
`$C_T`	Wert der programmierten Adresse T für Zyklenparametrierung und T-Funktionsersetzung
`$C_T_PROG`	Adresse T ist in einem Satz mit Zyklenaufruf oder T-Funktionsersetzung programmiert
`$C_TS`	Liefert bei T- oder TCA-Ersetzung den programmierten Werkzeugbezeichner (`$C_TS_PROG == TRUE`).
`$C_TS_PROG`	Liefert TRUE, wenn bei der T- oder TCA-Ersetzung ein Werkzeugbezeichner programmiert wurde
`$C_TE`	Adresserweiterung für Adresse T bei Unterprogrammaufruf per T-Funktion
`$C_D`	Wert der programmierten Adresse D im ISO-Mode für Zyklenparametrierung
`$C_D_PROG`	Adresse ist in einem Satz mit Zyklenaufruf programmiert
`$C_DL`	Wert der programmierten Adresse DL (additive Werkzeugkorrektur) bei einem Unterprogrammaufruf per M/T-Funktionsersetzung
`$C_DL_PROG`	Abfrage, ob bei einem Unterprogrammaufruf per M/T-Funktionsersetzung die Adresse DL (additive Werkzeugkorrektur) programmiert wurde
`$C_TCA`	Liefert TRUE, wenn die TCA-Ersetzung aktiv ist
`$C_DUPLO_PROG`	Liefert TRUE., wenn die Duplo-Nummer bei der TCA-Ersetzung programmiert wurde
`$C_DUPLO`	Liefert den Wert der programmierten Duplo-Nummer bei TCA-Ersetzung (`$C_DUPLO_PROG == TRUE`)

WZV-Sprachbefehl	Funktionen
$C_THNO_PROG	Liefert TRUE., wenn die Toolholder/Spindel-Nummer bei der TCA-Ersetzung programmiert wurde
$C_THNO	Liefert den Wert der programmierten Toolholder/Spindel-Nummer bei TCA-Ersetzung ($C_THNO_PROG == TRUE)

11.10.63 Variablen für WZ-Wechsel in Synchronaktion

WZV-Sprachbefehl	Funktionen
$AC_TC_FCT	Kommandonummer -1: Zum Lesezeitpunkt ist kein Kommando der WZV aktiv 1: Bewegen (Beladen, Entladen,...) 2: Wechsel vorbereiten 3: Wechsel EIN 4: Wechsel EIN (Revolver, ohne M06) 5: Wechsel vorbereiten und Wechsel EIN (mit M06)
$AC_TC_STATUS	Quittierstatus von PLC FC8/FC6
$AC_TC_THNO	Nummer des Werkzeughalters oder der Spindel auf den das neue Werkzeug eingewechselt werden soll
$AC_TC_TNO	interne T-Nummer des einzuwechselnden Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_MFN	Quell-Magazinnummer des neues Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_LFN	Quell-Platznummer des neuen Werkzeugs 0: es gibt keine neues Werkzeug
$AC_TC_MTN	Ziel-Magazinnummer des neuen Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_LTN	Ziel-Platznummer des neuen Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_MMYN	Eigentümer-Magazin des neuen Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_LMYN	Eigentümerplatz-Magazin des neuen Werkzeugs 0: es gibt kein neues Werkzeug
$AC_TC_MFO	Quell-Magazinnummer des alten Werkzeugs 0: es gibt kein altes Werkzeug
$AC_TC_LFO	Quell-Platznummer des alten Werkzeugs 0: es gibt kein altes Werkzeug
$AC_TC_MTO	Ziel-Magazinnummer des alten Werkzeugs 0: es gibt kein altes Werkzeug
$AC_TC_LTO	Ziel-Platznummer des alten Werkzeugs 0: es gibt kein altes Werkzeug
$AC_TC_CMDT	Triggervariable auf die Kommandoausgabe von NC Wird für einen IPO-Takt gesetzt, wenn NC ein neues Kommando ausgibt.

WZV-Sprachbefehl	Funktionen
$AC_TC_CMDC	Counter für die Kommandoausgabe der NC Mit jeder NC-Kommandoausgabe wird diese Variable um 1 hochgezählt. Kann auch geschrieben (Null-setzen) werden.
$AC_TC_ACKT	Triggervariable auf das Kommando von PLC Wird für einen IPO-Takt gesetzt, wenn die PLC ein Kommando an die NC ausgibt. Kommandoquittung oder eigenständige Mitteilung (asynchroner Transfer).
$AC_TC_ACKC	Counter für Kommando der PLC Mit jedem PLC-Kommando wird diese Variable um 1 hochgezählt. Kann auch geschrieben (Null-setzen) werden.

Zugriff	Lesen im Teileprogramm	Schreiben im Teileprogramm	Lesen in Synchronaktion	Schreiben in Synchronaktion
	X	-	X	-
Impliziter Vorlaufstopp	X	-		

11.11 Festlegungen bei der Programmierung von Daten

11.11.1 Werkzeugwechsel

Programmierung der Werkzeuganwahl

Der Werkzeugwechsel teilt sich in 2 unterschiedliche Schritte auf:

1. Werkzeugwechsel-Vorbereitung
2. Werkzeugwechsel-Ausführung

Bei der NC-Programmierung können die Schritte 1-2 getrennt bzw. gemeinsam programmiert werden (siehe MD22550 TOOL_CHANGE_MODE).

Beispiele

- Werkzeugwechsel in einem Schritt: (Revolver)
  - Tx; Bereitstellen des neuen Werkzeuges "x" und Werkzeugwechsel durchführen.
- Werkzeugwechsel in zwei Schritten:
  - Tx; Werkzeugwechsel-Vorbereitung (Anwahl des Werkzeugs)
  - M06; Werkzeugwechsel-Ausführung

Hinweis

Bei aktiver Werkzeugverwaltung kann die Auswahl eines Werkzeugs nur über einen Werkzeugbezeichner (Namen) erfolgen. Wird nun eine T-Nummer programmiert, so wird die Nummer als Bezeichner (Name) verwendet. Dann muss das Werkzeug beim Beladen die T-Nummer als Namen bekommen.

- Werkzeugwechsel mit Bezeichner:
  - T="BOHRER"; Es wird ein Werkzeug mit dem Bezeichner "BOHRER" gesucht.
- Werkzeugwechsel mit Nummer als Bezeichner:
  - T="123"; Es wird ein Werkzeug mit dem Bezeichner "123" gesucht. Alternativ kann auch T123 programmiert werden.

11.11.2 Schneidenanwahl

Schneidenanwahl nach Werkzeugwechsel

Am Ende eines Werkzeugwechsels gibt es folgende Anwahlmöglichkeiten für die Werkzeugschneide:

1. Die Korrekturnummer D wird programmiert.
2. Die Korrekturnummer D wird nicht programmiert und durch MD20270 $MC_CUTTING_EDGE_DEFAULT vorgegeben:
   = 0: Nach M06 erfolgt keine automatische Schneidenanwahl
   > 0: Nummer der Schneide, die nach M06 angewählt ist.
   = -1 Die Schneiden-Nr. des alten Werkzeuges bleibt erhalten und wird nach M06 auch für das neue Werkzeug angewählt
   = -2 Korrektur des alten WZ bleibt erhalten und wird nach M06 auch für das neue Werkzeug angewählt.

Beispiele:

Werkzeugvorwahl mit folgender Schneidenanwahl

Die Schneidenanwahl bezieht sich immer auf das Werkzeug, das durch M06 eingewechselt wird.

`T1 M06`	Werkzeugwechsel - kein D programmiert, daher Korrekturanwahl gemäß MD 20270
`T5`	Werkzeugvorwahl
X .. Y.. Z...	Arbeiten mit T1 und der Korrektur aus MD 20270
`D2`	Korrektur D2 von T1 !!!
`M06`	Werkzeugwechsel; T5 wird eingewechselt - Korrekturanwahl gemäß MD 20270
`T1`	Werkzeugvorwahl
`X.., Y...`	Arbeiten mit T5 und der Korrektur aus MD 20270

Bei der Programmierung der Werkzeugbefehle wird zwischen der Programmierung für eine Hauptspindel bzw. Nebenspindel unterschieden. Nur die Werkzeugkorrekturwerte des Hauptspindelwerkzeugs werden von der Geometrie berücksichtigt, da pro Kanal immer nur mit einer aktiven Korrektur gearbeitet werden kann. Die Verarbeitung der Werkzeugbefehle für eine Nebenspindel hat nur für die Signalausgabe zur PLC und für die Funktion GETSELT(...) Bewandtnis.

Spindel-Nr. 2 = Hauptspindel:

```
T2 = "BOHRER"
M2 = 06
T1 = "FRAESER"      Werkzeuganwahl für Nebenspindel
M1 = 06             Werkzeugwechsel in die Nebenspindel
D1                  Schneidenanwahl von "Bohrer" (Hauptspindel)
```

Spindel-Nr. 2 = Hauptspindel:

```
T2 = "BOHRER"       Anwahl eines Werkzeugs für die Hauptspindel.
                    Alternativ könnte auch T="Bohrer" angegeben werden.
T1 = x;             Anwahl eines Werkzeugs für eine Nebenspindel
M2 = 06             Werkzeugwechsel
                    Alternativ könnte auch M06 angegeben werden
D1                  Schneidenanwahl eines Werkzeugs mit dem Bezeichner "BOHRER"
```

11.11.3 Werkzeugübernahme aus Programmtest

Mit dem MD20110 $MC_RESET_MODE_MASK, **Bit 3** kann eingestellt werden, dass das aktive Werkzeug und die WZ-Korrektur folgendermaßen übernommen werden:

- Bit 3= 1; aus dem zuletzt beendeten Testprogramm im Testbetrieb.
  Oder
- Bit 3 = 0; aus dem zuletzt beendeten Programm vor Einschalten des Programmtests.

Voraussetzung: Bei MD20110 müssen die Bits 0 und 6 gesetzt sein.

$P_ISTEST

Über die Systemvariable $P_ISTEST kann aus dem Teileprogramm geprüft werden, ob ein Programmtest aktiv ist. Die Systemvariable liefert bei aktivem Programmtest den Wert TRUE.

11.12 Programmierung T=Platznummer

Die Funktion "Programmierung T=Platznummer" ist nur bei aktiver WZV verfügbar. Diese Art der Programmierung ist nicht nur für Revolver, sondern für alle Magazinarten möglich.

Über das Maschinendatum MD20310 $TC_TOOL_MANAGEMENT_MASK wird die Programmierart zur Werkzeuganwahl eingestellt:

- MD20310 $TC_TOOL_MANAGEMENT_MASK, Bit 16=0 (Default-Einstellung) - Programmieren über T = "Bezeichner".
  - T = "x" mit x als Werkzeugbezeichner, z. B. "Bohrer"
- MD20310 $TC_TOOL_MANAGEMENT_MASK, Bit 16=1 - Programmieren über T = "Platznummer".
  - Tx, mit x als Platznummer des Magazins, mit dessen Werkzeug gearbeitet wird, z. B. Platznummer 4

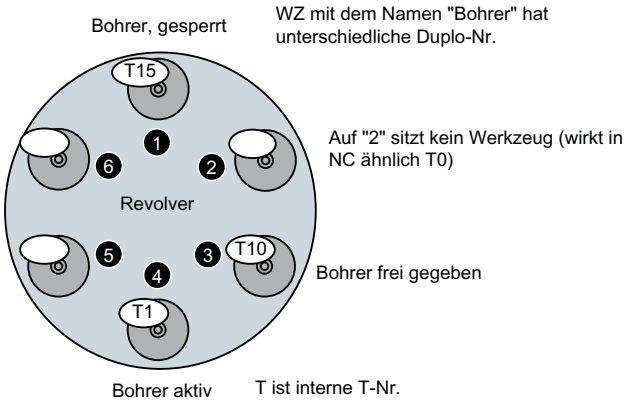


Bild 11-14 Programmieren von T=Platznummer

Erläuterung zur Grafik:

Bei aktivierter Funktion wird mit T1 anstelle des Werkzeugs mit dem Bezeichner "1" das Werkzeug auf dem Platz Nummer 1 angewählt. Dabei wird auf das erste mit dem WZ-Halter verbundene Magazin zugegriffen. Dann wird der Bezeichner des Werkzeugs auf diesem Platz ermittelt ("Bohrer").

Im Weiteren wird so verfahren, als wäre T="Bohrer" programmiert worden. Beim Wechselvorgang wird erst ermittelt, welches der drei Werkzeuge aus der Gruppe "Bohrer" eingewechselt werden soll.

Die eingestellte WZ-Suchstrategie wird berücksichtigt:

- Mit der Strategie "Nimm das erste verfügbare WZ aus der Gruppe" wird T10 von Platz 3 eingewechselt ($TC_MAMP2 Bit0 = 0).
- Mit der Strategie "Nimm das erste WZ mit dem Status *aktiv* aus der Gruppe" wird T1 eingewechselt ($TC_MAMP2 Bit0 = 1).
  Dabei wird vorausgesetzt, dass das Werkzeug T1 auch das Ersatzwerkzeug mit der kleinsten Duplonummer ist. Ist das nicht Fall, dann wird für $TC_MAMP2 Bit0 = 1 beim Werkzeugwechsel das Werkzeug mit der kleinsten Duplo-Nummer eingewechselt.

Die Suchstrategie wird über die Systemvariablen $TC_MAP10 (Aktuelle Suchstrategie des Magazins) bzw. $TC_MAMP2 (Art der Suchstrategie) eingestellt. Als Vorabeinstellung verwendet $TC_MAP10 die Werte von $TC_MAMP2.

T15 auf Platznr. 1 kann nicht verwendet werden, da es gesperrt ist.

Befindet sich auf dem programmierten Platz kein Werkzeug, so wird kein Alarm erzeugt. Der Wechsel wird wie gewohnt ausgegeben, mit T-Nr.=0. Damit kann z. B. der Revolver auf einen leeren Platz positioniert werden.

Sind dem WZ-Halter mehr als ein Magazin zugeordnet, bezieht sich die programmierte Platznummer auf das Magazin, das als Erstes in der Abstandstabelle definiert ist.

Liegen die Werkzeuge der WZ-Gruppe in verschiedenen Magazinen des Werkzeughalters, wird bei der Suche so verfahren wie im Standard der Werkzeugverwaltung.

Hinweis

Bei T=Platz kann alternativ auch T="Bohrer" programmiert werden.

T=1; Werkzeug

T="Bohrer"; Werkzeug mit Bezeichner Bohrer

11.13 Mehrere Revolver mit "T=Platznummer" aufrufen

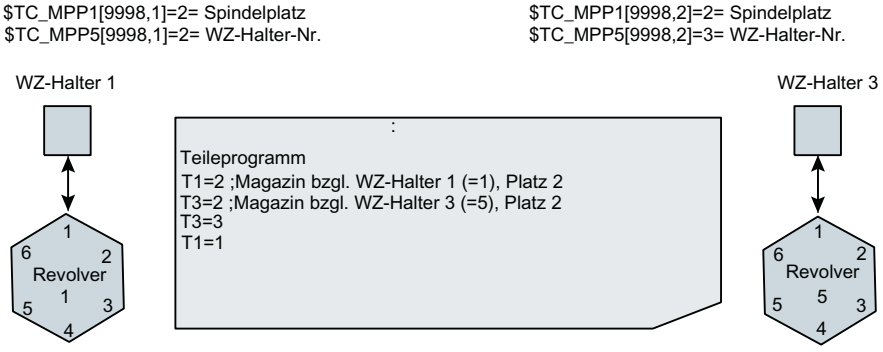


Bild 11-15 T=Platznummer als Funktion der WZV bei Drehmaschinen

Mit der Programmiermöglichkeit "T=Platznummer" und mehreren Magazinen kann in einem Kanal bzw. einer TO-Einheit gearbeitet werden.

- NC-Adresse T kann mit Adresserweiterung T1= ... programmiert werden.
- WZV interpretiert dies dann als Spindel-Nummer bzw. als Werkzeughalternummer.
- T ohne Adresserweiterung bezieht sich dann auf die Hauptspindel.

11.14 Programmierbeispiele

Aktion	Programmbefehl	Beschreibung
Werkzeug anlegen	DEF INT DUPLO_NR DEF INT T_NR DUPLO_NR = 7 T_NR=NEWT("BOHRER",DUPLO_NR)	Neues Werkzeug namens Bohrer mit der Duplo-Nr.= 7 anlegen. Die automatisch erzeugte T-Nr. wird in "T_NR" abgelegt.
	T_NR = GETT("BOHRER", DUPLO_NR) oder $TC_TP2[1] = "BOHRER" ; $TC_TP1[1] = DUPLO_NR	Ermitteln der T-Nummer des bereits angelegten Werkzeugs "BOHRER""mit der Duplo-Nr. 7. Hierbei wird allerdings die T-Nr. durch die Programmierung vorgegeben.
Werkzeugdaten lesen/schreiben	$TC_DP1[GETT("BOHRER", DUPLO_NR), 2] = 210	Werkzeugtyp schreiben für die 2. Schneide des Werkzeugs "Bohrer"/ DUPLO_NR
	$TC_DP1[T_NR, 2] = 210	Werkzeugtyp schreiben für die 2. Schneide des Werkzeugs " T-Nummer"
Werkzeug anwählen	T="BOHRER" oder: T=GETT("BOHRER", DUPLO_NR) oder Tx	Gibt es mehrere Werkzeuge mit diesem Bezeichner, so wird die T-Nr. des erstmöglichen dieser Werkzeuge zurückgegeben. Ermittelt T-Nummer für "BOHRER" mit Duplonummer = DUPLO_NR und wählt dieses an. Aufruf mit T-Nr. z. B. T1,T2,T3,....
Werkzeug löschen	$TC_TP1[T_NR,0]=0 oder DELT ("BOHRER", DUPLO_NR) $TC_TP1[GETT("BOHRER"),0]=0 oder alternativ: DELT("BOHRER")	Werkzeug mit T_NR wird gelöscht Werkzeug "BOHRER", DUPLO_NR wird gelöscht

11.15 Übersicht der übrigen BTSS-Bausteine der WZV

11.15.1 Magazindaten, Verzeichnis

BTSS-Baustein TMV

Berechnung der Zeile: Magazinnummer, wenn ein Feld[ ] vorhanden

Berechnung der Spalte: entfällt

NC-Bezeichner	Bezeichnung	BTSS-Variable	Typ
keine	Anzahl Magazine	`numActMags`	WORD
	Nummer des Magazins	`magVNo[ ]`	WORD
	Bezeichner des Magazins	`magVIdent[ ]`	String
	Anzahl der Adapter	`numActAdapter`	WORD
	Nummer des Adapters	`adaptVNo`	WORD
	Adapter-Typ	`AdaptVType`	WORD

11.15.2 Werkzeugdaten, Verzeichnis

BTSS-Baustein TV

Berechnung der Zeile: laufende Nr. der Werkzeuge, wenn [ ] vorhanden

Berechnung der Spalte: entfällt

NC-Bezeichner	Bezeichnung	BTSS-Variable	Typ
keine	Duplo-Nummer	`nrDuplo[ ]`	WORD
	Anzahl Schneiden	`numCuttEdges[ ]`	WORD
	Anzahl Werkzeuge im Bereich TO	`numTools`	WORD
	Letzte vergebene T-Nummer für WZV	`TnumWZV`	WORD
	Werkzeug-Bezeichner	`toolIdent[ ]`	String
	aktuelles Magazin	`toolInMag[ ]`	WORD
	aktueller Platz	`toolInPlace[ ]`	WORD
	T-Nummer	`toolNo[ ]`	WORD
	Anzahl Werkzeuggruppen	`numToolsGroups`	WORD

11.15.3 Parametrierung, Rückgabeparameter _N_TMGETT, _N_TSEARC

BTSS-Baustein TF

Berechnung der Zeile: siehe Tabelle

Berechnung der Spalte: entfällt

Bezeichnung	BTSS-Variable	Berechnung der Zeile	Typ
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TD	`parDataTD`	Parameterindex von Baustein TD	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TO	`parDataTO`	Parameterindex von Baustein TO	DOUBLE
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TD	`parDataToolIdentTD`	Parameterindex von Baustein TD	String

Bezeichnung	BTSS-Variable	Berechnung der Zeile	Typ
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TS	parDataTS	Parameterindex von Baustein TS	DOUBLE
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TU	parDataTU	Parameterindex von Baustein TU	DOUBLE
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TUE	parDataTUE	Parameterindex von Baustein TUE	DOUBLE
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TUS	parDataTUS	Parameterindex von Baustein TUS	DOUBLE
Maske für Suchkriterium des PI TSEARCH (BTSS-Baustein TD)	parMasksTD	Parameterindex von Baustein TD	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TO	parMasksTO	Parameterindex von Baustein TO	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TS	parMasksTS	Parameterindex von Baustein TS	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TU	parMasksTU	Parameterindex von Baustein TU	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TUE	parMasksTUE	Parameterindex von Baustein TUE	WORD
Vergleichswert für PI TSEARCH von Variablen des BTSS-Bausteins TUS	parMasksTUS	Parameterindex von Baustein TUS	WORD
D-Nummer der verwendeten Schneide	resultCuttingEdgeNrUsed	Anzahl Schneiden des WZ	WORD
Anzahl der benutzten Schneiden	resultNrOfCutEdgesUsed	1	WORD
Rückgabe: gefundene Werkzeuge	resultNrOfTools	1	WORD
Rückgabe: T-Nummern der gefundenen Werkzeuge	resultToolNr[ ]	1... resultNrOfTools	WORD
T-Nummer der verwendeten Schneide	resultToolNrUsed	Anzahl Schneiden des WZ	WORD

11.15.4 Arbeitskorrekturen

BTSS-Baustein AEV

Berechnung der Zeile: Schneidennummer, wenn [ ] vorhanden

Berechnung der Spalte: entfällt

NC-Bezeichner	Bezeichnung	BTSS-VAR	Typ
keine	Anzahl D-Nummern im Baustein	numActDEdges	WORD
	D-Nummern	Dno[...]	WORD
	interne T-Nummer	toolNo[...]	WORD
	Schneidennummer	cuttEdgeNo[...]	WORD
	Werkzeugbezeichner	toolIdent[...]	STRING

NC-Bezeichner	Bezeichnung	BTSS-VAR	Typ
	Duplo-Nummer	`duploNo[...]`	WORD
	Magazin	`toolInMag[...]`	WORD
	Platz	`toolInPlace[...]`	WORD

11.15.5 PI-Dienste und Sprachbefehle für WZV

Übersicht

Mit dem FB4 (PI_SERV) bzw. FB7 können Programminstanz-Dienste (PI-Dienste) im NC-Bereich gestartet werden. Durch Anforderung über den PI-Dienst wird im NC ein Programmteil abgearbeitet, das eine bestimmte Funktion ausübt (z. B. Leerplatzsuche in einem Magazin bei Werkzeugverwaltung).

PI-Dienst	Funktionen	NC-Sprachbefehl
`MMCSEM`	Semaphoren für verschiedene PI-Dienste	
`DELETO`	Werkzeug löschen	`DELT("WZ", Duplo)`
`DELECE`	Löschen einer Werkzeugschneide	`$TC_DP1[t,d]=0`
`CREATO`	Werkzeug erzeugen	`NEWT("WZ", Duplo)`

`CRTOCE`	WZ erzeugen mit Angabe Schneidennr.	`$TC_DPx[t,D]`
		`$TC_DPCx[t,D]`
		`$TC_DPCSx[t,D]`
		`$TC_MOPx[t,D]`
		`$TC_MOPCx[t,D]`

`TMCRTO`	Werkzeug anlegen	`$TC_TPx[t]`
`TMCRTC` (nicht in PLC verfügbar)	Werkzeug anlegen mit Angabe Schneidennr.	`$TC_DPx[t,d]`
`CREACE`	Werkzeugschneide anlegen	`$TC_DP[t,d]=Wert`
`CRCEDN`	Lege neue Schneide an	`$TC_DPx[t,d]`
`TMFDPL`	Leerplatzsuche zum Beladen	`GETFREELOC(magNo, locNo, T-Nr., refMag, refLoc, withReserv)`
`TMMVTL`	Magazinplatz zum Beladen bereitstellen, Werkzeug entladen	
`TMPCIT`	Inkrementwert für Stückzahlzähler setzen Stückzahl dekrementieren um y	`SETPIECE(SpinNo,y)`
`TMPOSM`	Magazinplatz oder Werkzeug positionieren	`POSM(p,m,ip,im)`
`TMFPBP`	Leerplatzsuche nach Eigenschaften	
`TSEARC`	Komplexes Suchen über Suchmasken	Anwenderzyklenprogramm
`TMRASS`	Rücksetzen des Aktiv-Status	
`TMGETT`	Best. T-Nummer zum vorgegebenen WZ-Bezeichner mit Duplonr.	`GETT("WZ", Duplo)`
	Lesen der vorgewählten T-Nummer	`GETSELT(SpinNo)`

PI-Dienst	Funktionen	NC-Sprachbefehl
CHKDNO (nicht in PLC verfügbar) TMCHKD (nicht in PLC verfügbar)	Prüfe die Eindeutigkeit der D-Nummern der Werkzeugdaten der TO-Einheit, die dem ausführenden Kanal zugeordnet ist. Die Parameter t1, t2, d sind optional.	status=CHKDNO (t1,t2,d)
DZERO (nicht in PLC verfügbar)	Setze die D-Nummern aller Werkzeuge der dem Kanal zugeordneten TO-Einheit auf "ungültig" Auf BTSS werden solche D-Nummern mit dem Wert 0 angezeigt. NC-intern wird die ungültige D-Nummer durch Besetzen der D-Nummer mit dem Wert "alte D-Nummer"+32000 erzeugt.	DZERO
	Gib zur Korrekturnr. D=d die zugehörige interne T-Nr.=t des WZ an. Aus der WZ-Gruppe wird das WZ entnommen, das den Zustand "aktiv" und "war im Einsatz" hat.	status=GETACTTD (t,d)
	Gib die D-Nr. zum WZ t und dessen Schneide ce	d=GETDNO(t,ce)
	Setze die D-Nr. des WZ t und dessen Schneide ce auf den Wert d	status=SETDNO (t,ce,d)
	Lesen der aktiven T-Nr. und Status	status=GETACTT (Tno,"WZ")
	Löschbefehl für alle Einsatzortsabhängigen/Einrichtekorrekturen einer Schneide, bzw. eines WZ, wenn d nicht angegeben wird	status=DELDL(t,d)
SETTST (nicht in PLC verfügbar)	Setze Werkzeugstatus auf "aktiv"	SETTA(Stat,m,vnr)
SETTST (nicht in PLC verfügbar)	Setze Werkzeugstatus auf "nicht aktiv"	SETTIAStat,m,vnr)
CHKDM (nicht in PLC verfügbar)	Prüfung der eindeutigen D-Nr. im Magazin; m=Magazin	status=CHKDM(m)
	Werkzeug-Halternr. setzen (h=Halternr.)	SETMTH(h)
	Masterspindel setzen (s=Spindelnr.)	SETMS(s)
TRESMO	Standzeit-/Stückzahl-/Verschleiß-Sollwertaktivierung	RESETMON
TMAWCO (nicht in PLC verfügbar)	Aktivsetzen eines Verschleißverbundes	$TC_MAP9
	Ortsabhängige Korrekturen grob	$TC_EPx[t,d]
	Technologische Schleifdaten	$TC_TPGx[t]
	Lesen der eingewechselten T-Nummer	GETEXET
	Sprachbefehl zum Bewegen eines Werkzeugs	MVTOOL
	Lösche Werkzeug-Trägersatz	DELTC
	Werkzeugabwahl/Werkzeugwechsel unabhängig vom Status des Werkzeugs	TCA
	Lösche den Platzzustand "reserviert für WZ im Zwischenspeicher"	DELMLRES
	Lösche Eigentümermagazinplatz des WZ	DELMLOWNER

NC-Sprachbefehle

Mit den folgenden Sprachbefehlen werden NC-Zustände gelesen.

Funktionen	NC-Sprachbefehl
Aktive WZ-Nr. T	$P_TOOLNO
Zuletzt programmierte WZ-Nr. (ohne Magazinverwaltung)	$P_TOOLP
Aktive Werkzeugschneide	$P_TOOL

Funktionen	NC-Sprachbefehl
Aktive Werkzeuggesamtlänge; n=1-3	$P_TOOLL[n]
Aktiver Werkzeugträger	$P_TC
Aktiver Winkel einer WZ Trägerachse; n=1-2	$P_TCANG[n]
Differenz zwischen berechneten und verwendeten Winkel einer Werkzeugträgerachse bei Hirth-Verzahnung des Winkels	$P_TCDIFF[n]
Aktiver Werkzeugradius; n=1-2	$P_TOOLR
Anzahl Schneiden für WZ t; t=T-Nummer 1 32000	$P_TOOLND[t]
Existiert Werkzeug mit der T-Nummer t	$P_TOOLEXIST[t]
Aktive WZ Korrekturen, n=1...31	$P_AD[n]
Aktive WZ-Korrekturen transformiert; n=1-31	$P_ADT[n]
Aktive Summenkorrekturnummer	$P_DLNO
T-Nr. des Werkzeugs (numerisch)	$C_T
WZ-Bezeichner (string)	$C_TS
Bool-Variable, zeigt an, ob in $C_T ein T-Wort zur Verfügung steht	$C_T_PROG
Parameter für Werkzeugname als String	$C_TS_PROG
Adresserweiterung des T-Wortes	$C_TE
Programmierte D-Nr.	$C_D
Bool-Variable, zeigt an, ob in $C_D eine Korrektur-Nr. zur Verfügung steht	$C_D_PROG
Programmierte Summen-/Einrichtekorrektur	$C_DL
Bool-Variable, zeigt an, ob in $C_DL eine Korrektur-Nr. zur Verfügung steht	$C_DL_PROG
Kommandonummer, spezifiziert den gewünschten Vorgang: 1: Bewegen (Be-/Entladen, Umsetzen...); 2: Wechsel vorbereiten; 3: Wechsel Ein; 4: Wechsel Ein (Revolver, ohne M06); 5: Wechsel vorbereiten und Wechsel Ein (mit M06); -1 Zum Lesezeitpunkt ist kein Kommando der WZV aktiv	$AC_TC_FCT
Quittierstatus von PLC FC8/FC6	$AC_TC_STATUS
Toolholder oder Spindelnummer auf den das neue Werkzeug eingewechselt werden soll	$AC_TC_THNO
Quell-Magazinnummer des neuen Werkzeugs	$AC_TC_MFN
Quell-Platznummer des neuen Werkzeugs	$AC_TC_LFN
Ziel-Magazinnummer des neuen Werkzeugs	$AC_TC_MTN
Ziel-Platznummer des neuen Werkzeugs	$AC_TC_LTN
Quell-Magazinnummer des alten (auszuwechselnden) Werkzeugs	$AC_TC_MFO
Quell-Platznummer des alten (auszuwechselnden) Werkzeugs	$AC_TC_LFO
Ziel-Magazinnummer des alten (auszuwechselnden) Werkzeugs	$AC_TC_MTO
Ziel-Platznummer des alten (auszuwechselnden) Werkzeugs	$AC_TC_LTO
Gibt die Nummer der aktuellen Masterspindel zurück	$AC_MSNUM
Gibt die Nummer der Masterspindel zurück	$P_MSNUM
Gibt die Nummer des Master-WZ-Halters zurück	$AC_MTHNUM

Funktionen	NC-Sprachbefehl
Triggervariable auf Kommandoausgabe der NC (wird für einen IPO gesetzt)	$AC_TC_CMDT
Triggervariable auf Quittierung der PLC (wird für einen IPO gesetzt)	$AC_TC_ACKT
Zähler der Kommandoausgabe	$AC_TC_CMDC
Zähler der Quittungen	$AC_TC_ACKC
Eigentümermagazin des Neu-Werkzeugs	$AC_TC_MMYN
Eigentümerplatz des Neu-Werkzeugs	$AC_TC_LMYN
Gibt die Nummer des Master-WZ-Halters zurück	$P_MTHNUM
Magazinnummer des Werkzeugs t	$A_TOOLMN[t]
Magazinplatz des Werkzeugs t	$A_TOOLMLN[t]
Nummer des Eigentümermagazins	$A_MYMN
Nummer des Eigentümermagazinplatzes	$A_MYMLN
Anzahl definierter Magazine, die dem Kanal zugeordnet sind	$P_MAGN
Anzahl definierter Magazine, i-te Magazinnummer	$P_MAG[i]
Anzahl definierter Adapter, die dem Kanal zugeordnet sind	$P_MAGNA
Anzahl definierter Adapter, i-te Adapternummer	$P_MAGA[i]
Anzahl der Magazine n, die mit dem Platz n verbunden sind	$P_MAGNDIS[m,n]
Nummer des i-ten Magazins, das mit dem Platz l des Zwischenspeichermagazins verbunden ist	$P_MAGNDISS[l,i]
Nummer des i-ten Magazins, das mit dem Platz l des Belademagazins verbunden ist	$P_MAGDISL[l,i]
Anzahl definierter Magazinplatztyp-Hierarchien	$P_MAGNH
Anzahl der definierten Platztypen in der n-ten definierten Hierarchie	$P_MAGNHLT[n]
m-ter Platztyp der Hierarchie n	$P_MAGHLT[n,m]
Anzahl der der Spindelnummern, Werkzeug-Halternummern n zugeordneten Zwischenspeicher	$P_MAGNREL[n]
m-te Zwischenspeichernummer der n-ten Spindelnummer, Werkzeug-Halternummer	$P_MAGREL[n,m]
Anzahl der Spindelplätze, Werkzeug-Halterplätze im Zwischenspeichermagazin, die dem Kanal zugeordnet sind	$P_MAGNS
n-te Nummer der Spindel / des Werkzeughalters im Zwischenspeicher	$P_MAGS[n]
Ermittlung der definierten D-Nummer eines Werkzeugs	$P_TOOLD
Existenz eines Werkzeugs feststellen	$P_TOOLEXIST
Anzahl von DL-Korrekturen der D-Korrektur gegeben durch T-Nummer T und D-Nummer D	$P_TOOLNDL[t,d]
Anzahl definierter Werkzeuggruppen, die dem Kanal zugeordnet sind	$P_TOOLNG
Anzahl definierter Werkzeuge, die dem Kanal zugeordnet sind	$P_TOOLNT
i-te Werkzeugnummer T	$P_TOOLT[i]
Alle Werkzeuge, deren Parameter $TC_TP11 eines der Bits von $P_USEKT gesetzt hat, stehen den folgenden Werkzeugwechseln zur Verfügung. Der Wert Null ist inhaltsgleich mit "alle Bits sind gesetzt"	$P_USEKT, $TC_TP11
Faktor für die Werkzeugsuche	$AC_MONMIN
Stückzahlzählung	$A_USEDND
Stückzahlzählung	$A_USEDT

Funktionen	NC-Sprachbefehl
Stückzahlzählung	$A_USEDD
Faktor für Standzeitüberwachung lesen	$A_MONIFACT

Multitool- und Werkzeugangaben in PI-Diensten

Folgende PI-Dienste beziehen sich auf Werkzeuge und Multitools:

PI-Dienst	Bedeutung
TMFDPL	Leerplatzsuche für genanntes WZ bzw. Multitool zum Beladen bzw. Bestücken. Die Leerplatzsuche im MT ist nur sinnvoll für nicht beladene Werkzeuge, d. h., im Rahmen des Bestückungsvorgangs von Werkzeugen in das MT.
TMFPBP	Leerplatzsuche für genanntes WZ bzw. Multitool. Der Dienst sucht unter den in der Parametrierung angegebenen Magazin(en) einen Leerplatz, welcher den angegebenen Kriterien "Werkzeuggröße" und "Platztyp" genügt.
TMMVTL	Werkzeug bzw. Multitool beladen oder entladen. Es ist nicht möglich, Werkzeuge eines Multitools für den Bewegebefehl zu programmieren. Es ist ebenfalls nicht möglich, mit dem Befehl ein WZ in ein MT zu bewegen.
TMGETT	T-Nummer des Werkzeugs mit vorgegebenem WZ-Bezeichner/Duplonummer bzw. MT-Nummer des Multitools mit vorgegebenem Bezeichner (Multitool: die Duplonummer wird dabei mit dem Parameterwert 00000 programmiert)
TRESMO	Rücksetzen der Überwachungswerte. Falls eine MT-Nummer programmiert wird, werden genau die Werkzeuge des Multitools behandelt.
TMPOSM	Magazin auf vorgegebenen Magazinplatz, vorgegebenes Werkzeug bzw. Multitool positionieren.

Folgende PI-Dienste gibt es funktional für Werkzeuge und Multitools aber mit unterschiedlichen Namen:

PI-Dienst	Bedeutung
Erzeugen	
CREATO	WZ mit vorgegebener T-Nummer erzeugen
CRTOCE	WZ mit vorgegebener T- und D-Nummer erzeugen
TMCRTO	WZ mit vorgegebenem Bezeichner/Duplonummer erzeugen
TMCRTC	WZ mit vorgegebenem Bezeichner/Duplonummer und D-Nummer erzeugen
Löschen	
DELETO	Werkzeug mit vorgegebener T-Nummer löschen
TMDLTO	Werkzeug mit vorgegebenem Bezeichner/Duplonummer löschen

Hinweis

Wenn diese WZ-spezifischen PI-Dienste mit MT-spezifischem Namen, MT-Nummer gerufen werden, wird der jeweilige PI mit Fehlerstatus abgebrochen.

Folgende PI-Dienste sind funktional nur für Werkzeuge definiert (d. h. für Multitools gibt es keine analogen PI-Dienste):

PI-Dienst	Bedeutung
CHEKDM	Prüfung der D-Nummern auf Eindeutigkeit anstoßen
CHKDNO	Prüfen, ob die Werkzeuge eindeutige D-Nummer haben
DZERO	Alle D-Nummern ungültig setzen bei Funktion "eindeutige D-Nummer"
CRCEDN	Neue WZ-Schneide mit Vorgabe der D-Nummer erzeugen
CREACE	Werkzeugschneide erzeugen
DELECE	Löschen einer Werkzeugschneide
SETTST	Werkzeug aus Werkzeuggruppe aktivieren (WZ-Zustand "aktiv" setzen)
TMRASS	Rücksetzen des Aktiv-Status
TMAWCO	Aktiven Verschleißverbund in einem Magazin setzen
TMPCIT	Inkrementwert für Stückzahlzähler setzen
TSEARC	Komplexes Suchen über Suchmasken

Folgende PI-Dienste sind für Adapter definiert:

PI-Dienst	Bedeutung
TMCRAD	Adapter mit gegebenem Typ erzeugen
TMDLAD	Adapter mit gegebener Nummer löschen

Hinweis

Der PI-Dienst betrachtet beim Suchvorgang nur Werkzeuge, auch solche, die in einem Multitool enthalten sind, d. h. der PI-Dienst betrachtet keine Multitools. Es wird kein PI-Dienst "komplexe Multitool-Suche" definiert.

Wenn die WZ-spezifischen PI-Dienste mit MT-spezifischem Namen, MT-Nummer gerufen werden, wird der PI-Dienst mit Fehlerstatus abgebrochen.

SINUMERIK Operate Bedienoberfläche 12

12.1 Bedienoberfläche konfigurieren

Konfigurierbare Einstellungen

Folgende Einstellungen sind u. a. konfigurierbar:

- Allgemeine Einstellungen
- Werkzeuglisten konfigurieren
- Listenparameter konfigurieren
- Werkzeugtypen konfigurieren
- Fenster "Weitere Daten" konfigurieren
- Fenster "Neues Werkzeug - Favoriten" konfigurieren
- Fenster "Neues Werkzeug" konfigurieren
- Fenster "Neues Multitool" konfigurieren
- Fenster "Neuer Winkelkopf" konfigurieren
- Fenster "Neues Werkzeug von Codeträger" konfigurieren
- Fenster "Neues Werkzeug aus Datei" konfigurieren
- Standardwerte für neue Werkzeuge konfigurieren
- Fenster "Details" konfigurieren
- Name für Magazine und Magazinplätze vergeben
- Fenster "Tooltips für Multitoolplätze vergeben" konfigurieren
- Name für Magazinplatztypen vergeben
- Zuordnung von Magazinen und Magazinplätzen zu Kanälen
- Zuordnung von Magazinen und Magazinplätzen zu Koordinatensystemen
- Sperren des Softkeys in Abhängigkeit von Filtern
- Fenster "Auswahl der Beladestelle" konfigurieren
- Kühlmittel und werkzeugspezifische Funktionen
- Werkzeugwechselgrund beim Reaktivieren
- Codeträgeranbindung konfigurieren
- Speichern auf Datei konfigurieren

Wie Sie diese Anpassungen konfigurieren, wird in den nachfolgenden Kapiteln erläutert.

Vorgehensweise

1. Öffnen Sie das Verzeichnis: /**siemens**/sinumerik/hmi/template/cfg.
2. Kopieren Sie die Vorlagendatei "oem_sltmlistconfig.xml".
3. Kopieren Sie die Datei ins Verzeichnis /**oem**/sinumerik/hmi/cfg
   ODER: /**user**/sinumerik/hmi/cfg.
4. Ändern Sie den Dateinamen entsprechend der eingestellten Technologie:

Dateiname	MD52200 Technologie	MD52201 Technologieer-weiterung
sltmlistconfig.xml	Technologie Fräsen	
sltmturninglistconfig.xml	Technologie Drehen	
sltmmillturnlistconfig.xml	Technologie Fräsen	Technologie Drehen
sltmcirculargrindinglistconfig.xml	Technologie Rundschleifen	
sltmsurfacegrindinglistconfig.xml	Technologie Flachschleifen	
sltmplclistconfig.xml	PLC-Werkzeugverwaltung "TRANSLINE 2000"	

5. Entfernen Sie die Kommentare des Beispiels in der Vorlage, um Ihre spezifische Konfiguration zu erstellen.

Konfigurationsbeispiele

In der Vorlage oem_sltmlistconfig.xml sind Beispiele zu allen möglichen Konfigurationen enthalten, die in den nachfolgenden Kapiteln im Einzelnen beschrieben sind:

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
- <CONFIGURATION>
<!--  ************************************************************-->
<!--  enter your configuration behind this comment -->
<!--  ************************************************************ -->


 Konfigurationsbeispiele


<!--  ************************************************************ -->
<!--  enter your configuration ahead this comment -->
<!--  ************************************************************ -->
 </CONFIGURATION>
```

Tragen Sie Ihre Konfiguration unter Verwendung folgender xml-Bezeichner ein:

Bezeichner <tag>	Für folgende Einstellung:
<SETTINGS>	Allgemeine Einstellungen (Seite 284)
<LISTCONFIGURATION>	Werkzeuglisten konfigurieren (Seite 290)
<PARAMETERCONFIGURATION>	Listenparameter konfigurieren (Seite 309)
<TOOLTYPECONFIGURATION>	Werkzeugtypen konfigurieren (Seite 315)
<MOREDATACONFIGURATION>	Fenster "Weitere Daten" konfigurieren (Seite 318)

Bezeichner <tag>	Für folgende Einstellung:
<NEWTOOLFAVORITECONFIGURATION>	Fenster "Neues Werkzeug - Favoriten" konfigurieren (Seite 320)
<NEWTOOLCONFIGURATION>	Fenster "Neues Werkzeug" konfigurieren (Seite 320)
<NEWMULTITOOLCONFIGURATION>	Fenster "Neues Multitool" konfigurieren (Seite 321)
<NEWCROSSHEADCONFIGURATION>	AUTOHOTSPOT
<NEWTOOLIDENTCONNECTIONCONFIGURATION>	Fenster "Neues Werkzeug von Codeträger" konfigurieren (Seite 322)
<NEWTOOLSAVETOOLDATACONFIGURATION>	Fenster "Neues Werkzeug aus Datei" konfigurieren (Seite 322)
<NEWTOOLDEFAULTVALUECONFIGURATION>	Standardwerte für neue Werkzeuge konfigurieren (Seite 323)
<DETAILSCONFIGURATION>	Fenster "Details" konfigurieren (Seite 328)
<MAGAZINEPLACENAMECONFIGURATION>	Namen für Magazine und Magazinplätze vergeben (Seite 330)
<MULTITOOLPLACETOOLTIPCONFIGURATION>	Fenster "Tooltips für Multitoolplätze vergeben" konfigurieren (Seite 332)
<PLACETYPECONFIGURATION>	Namen für Magazinplatztypen vergeben (Seite 332)
<CHANNELMAGAZINEASSIGNMENT>	Zuordnung von Magazinen und Magazinplätzen zu Kanälen (Seite 334)
<COORDSYSMAGAZINEASSIGNMENT>	Zuordnung von Magazinen und Magazinplätzen zu Koordinatensystemen (Seite 336)
<FILTERCONFIGURATION>	Sperren des Softkeys in Abhängigkeit von Filtern (Seite 337)
<LOADSTATIONSHANDLEALLTOOLS>	Fenster "Auswahl der Beladestelle" konfigurieren (Seite 338)
<TOOLCHANGEREASON>	Werkzeugwechselgrund beim Reaktivieren (Seite 342)
<IDENTCONNECTIONCONFIGURATION>	Codeträgeranbindung konfigurieren (Seite 346)
<SAVETOOLDATACONFIGURATION>	Speichern in Datei konfigurieren (Seite 349)

12.1.1 Allgemeine Einstellungen

Tag <SETTINGS>

Im Tag <SETTINGS> werden die allgemeinen Einstellungen für die Bedienoberfläche der Werkzeugverwaltung hinterlegt.

Einträge	Bedeutung
ReactivateWithMagPos	true - Reaktivieren mit Positionierung des Werkzeugs auf die Beladestelle des Werkzeugmagazins. false - Standardeinstellung, Reaktivieren ohne Positionierung des Werkzeugmagazins.
ReactivateAllMonitorModes	true - Es werden die Istwerte aller in der NC eingestellten Überwachungsarten zurückgesetzt. false - Standardeinstellung, Istwert der aktiven Überwachungsart wird zurückgesetzt.
ReactivateEnabled	true - Standardeinstellung, Reaktivieren ist frei gegeben. false - Reaktivieren ist gesperrt.
CreateNewToolDialog	true - Fenster "Neues Werkzeug" wird eingeblendet. In diesem Fenster können Sie vor dem tatsächlichen Anlegen des Werkzeugs den Werkzeugnamen und die Werkzeuggröße (Werkzeug übergroß) eingeben. Weitere Eingabemöglichkeiten können über die Projektierung des Fensters "Neues Werkzeug" festgelegt werden. false - Standardeinstellung, Fenster "Neues Werkzeug" wird nicht eingeblendet. Das Werkzeug wird nach der Eingabe des Namens direkt in der Werkzeugliste angelegt.
CreateNewMultiToolDialog	true - Standardeinstellung, Fenster "Neues Multitool" wird eingeblendet. In diesem Fenster können Sie vor dem tatsächlichen Anlegen des Multitools Folgendes konfigurieren: • Multitoolnamen • Anzahl der Multitoolplätze • Art der Abstandeingabe • Wert des Platzabstands pro Multitoolplatz false - Fenster "Neues Multitool" wird nicht eingeblendet. Das Multitool wird nach der Eingabe des Namens direkt in der Werkzeugliste angelegt.
CreateNewCrossHeadDialog	true - Fenster "Neuer Winkelkopf" wird eingeblendet. In diesem Fenster können Sie vor dem tatsächlichen Anlegen des Winkelkopfes den Winkelkopfnamen und die Winkelkopfgröße (Winkelkopf übergroß) eingeben. Weitere Eingabemöglichkeiten können über die Projektierung des Fensters "Neuer Winkelkopf" festgelegt werden. false - Standardeinstellung, Fenster "Neuer Winkelkopf" wird nicht eingeblendet. Der Winkelkopf wird nach der Eingabe des Namens direkt in der Werkzeugliste angelegt

Einträge	Bedeutung
MagazineSelectionDialog	true - Es wird ein Dialog angeboten, in dem der Zwischenspeicher, die Werkzeugmagazine und der NC-Speicher angezeigt werden. Jedes dieser Elemente trägt eine Markierung, mit der das entsprechende Element aus der Listendarstellung aus- bzw. eingeblendet werden kann. Mit dem Softkey "Gehe zu" kann ein Element in der Listendarstellung direkt angewählt werden. false - Kein Dialog zur Magazinanwahl. Es wird zwischen dem Zwischenspeicher, den Werkzeugmagazinen und dem NC-Speicher umgeschaltet. Fehlt der Eintrag MagazineSelectionDialog im Tag `<Settings>`, so wird bei einer Magazinkonfiguration mit mehr als einem Werkzeugmagazin (Systemmagazine ausgeschlossen) automatisch der Dialog angeboten. Wenn nur ein Werkzeugmagazin existiert, wird die Umschalt-Funktion verwendet.
NewToolFavoritesOnly	Bezieht sich auf die Funktion "Neues Werkzeug". true - es werden nur Werkzeugtypen angeboten, die als Favoriten konfiguriert sind. false - Standardeinstellung, es werden alle Werkzeugtypen angeboten.
SortationInFirstEtcLevel	true - Die Funktion Sortieren wird in die erste Softkeyebene gelegt. Die Funktion "Weiteres" entfällt. false - Standardeinstellung, Sortierung in allen Softkeyebenen.
ToolBufferOnceOnTop	true - Standardeinstellung, Zwischenspeicher wird am Anfang der Liste dargestellt. false - Zwischenspeicher wird pro Magazin dargestellt. Dabei werden nur die Plätze angezeigt, die dem Magazin zugeordnet sind.
UnloadToolFromLoadPlaceEnabled	true - Standardeinstellung. Ein Werkzeug auf der Beladestelle kann mit dem Softkey "Entladen" von der Beladestelle entfernt werden. false - Der Softkey "Entladen" wird bei einem Werkzeug auf der Beladestelle nicht angezeigt.
CreateNewToolWithCuttPos9Enabled	true - Schneidenlage 9 wird in der Auswahl der Schneidenlagen angeboten. false - Standardeinstellung, Schneidenlage 9 wird in der Auswahl der Schneidenlagen nicht angeboten.
AutoHNumberDisplayEnabled	Spalte des Schneidenparameters "IsoHNoDPH - ISO H-Nummer" in Abhängigkeit von der Maschinenkonfiguration. true - Standardeinstellung, Spalte wird automatisch eingeblendet. false - Spalte wird nicht automatisch eingeblendet.
AutoUniqueDNumberDisplayEnabled	Spalte des Schneidenparameters "UniqueDNo" in Abhängigkeit von der Maschinenkonfiguration. true - Standardeinstellung, Spalte wird automatisch eingeblendet. false - Spalte wird nicht automatisch eingeblendet.

Einträge	Bedeutung
AccessLevelDeleteAllDNos	Folgender Eintrag legt fest, ab welcher Zugriffsstufe die Funktion "alle D-Nummern löschen" angeboten wird. Standardeinstellung: Zugriffsstufe 3 (Anwender) - nur relevant, wenn die Einstellung "eindeutige D-Nummer" gesetzt ist.
MagazineNumberOfDCheckAndActivate	Standardeinstellung: -2 - Bei der Funktion "Überprüfung auf eindeutige D-Nummer und aktivieren des Werkzeugs" werden die Werkzeuge aller Magazine betrachtet. >0 - Bei der oben genannten Funktion werden nur die Werkzeuge des genannten Magazins betrachtet.
AutoFixedPlaceDisplayEnabled	Spalte des Werkzeugparameters "ToolStateFixed" - (Werkzeugstatus - Werkzeug ist festplatzcodiert) wirkt in Abhängigkeit von der Magazinkonfiguration. true - Standardeinstellung, Spalte wird automatisch ausgeblendet, wenn alle konfigurierten Magazine festplatzcodiert sind. false - Spalte wird nicht automatisch ausgeblendet.
MagPlaceAdapterDisplayEnabled	Spalten der Magazinplatzadapterparameter werden im Fenster "Magazin" angezeigt. false - Standardeinstellung, Parameter werden nicht angezeigt.
AdaptTransDataViewEnabled	Die magazinplatzadaptertransformierte Darstellung der Schneidenparameter wird freigeschaltet. Sie kann im Einstellungsdialog der Werkzeuglisten ein- und ausgeschaltet werden. false - Standardeinstellung, die magazinplatzadaptertransformierte Darstellung der Schneidenparameter wird nicht freigeschaltet.
ActiveCompensationsEnabled	Die Funktion "Arbeitskorrekturen" wird freigeschaltet. Diese Funktion ist abhängig von der Maschinenkonfiguration. Eindeutige D-Nummer, Magazinplatzadapter und Magazinverwaltung muss gesetzt sein. false - Standardeinstellung, die Funktion Arbeitskorrekturen wird nicht freigeschaltet.
M2NLockEnabled	Sperren einer M zu N-Umschaltung während eine Funktion der Werkzeugverwaltung ausgeführt wird. true - Standardeinstellung, Umschaltung wird gesperrt.
SDToolLengthInListEnabled	true - Standardeinstellung, Auswertung der Settingdaten SD42940, SD42942 und SD42950 bei der Anzeige der Achsnamen in den Werkzeuglisten. false - Es findet keine Auswertung der Settingdaten bei der Anzeige der Achsnamen statt.
DisableMultiTools	true - Obwohl in der NC die Funktion Multitools eingerichtet ist, werden diese in den Werkzeuglisten nicht angeboten. false - Standardeinstellung, wenn in der NC die Funktion Multitools eingerichtet ist, werden diese in den Werkzeuglisten auch angeboten.

Einträge	Bedeutung
CursorPositionInfoEnabled	true - Informationen zum Werkzeug, auf dem sich der Cursor befindet, werden auf die SlCap-lokale Variable "/Hmi/TMHMI-CurData" geschrieben. false - Standardeinstellung, es werden keine Informationen zum Werkzeug, auf dem sich der Cursor befindet, auf einer Schnittstelle für den OA bereitgestellt.
EnableMonAndBasisGrindingToolTypes	true - Umschaltung der Schleifwerkzeugtypen in Bezug auf "mit/ohne schleifspezifische Werkzeugüberwachung", "mit/ohne Berücksichtigung des Basismaßes bei der Berechnung des Scheibenradius" freigegeben. false - Standardeinstellung, Umschaltung nicht freigegeben.
ErrorMessageWithoutProgressBar	true - Fehlermeldungen werden ohne Fortschrittanzeige ausgegeben und müssen über Softkey quittiert werden. false - Standardeinstellung, Fehlermeldung wird nach Ablauf der Fortschrittsanzeige automatisch geschlossen.
ColumnZoomByResolution	true - Die Spaltenbreite wird mit der Auflösung des HMI angepasst. false - Standardeinstellung, die Spaltenbreite wird an die Schriftgröße im HMI angepasst. Durch diese Einstellung können bei größeren Auflösungen mehr Spalten nebeneinander ohne Scrollen dargestellt werden.
AccessLevelReadDetailsAllParameter	Folgender Eintrag legt fest, ab welcher Zugriffsstufe die Funktion "Weitere Details" im Dialog "Details" angeboten wird. Standardeinstellung: Zugriffsstufe 7 (Schlüsselschalterstellung 0) - kann durch die Einstellung in MD51198 überlagert werden
FixNumberOfMultiToolPlaces	Multitools werden mit einer festen Anzahl von Multitoolplätzen angelegt. Es erfolgt keine Nachfrage beim Anlegen der Multitools.
FixKindOfDistInMultiTools	Multitools werden mit einer festen Art der Abstandeingabe der Multitoolplätze angelegt. Es erfolgt keine Nachfrage beim Anlegen der Multitools. Möglicher Eingabewert: 3 - Abstand der Multitoolplätze als Winkel.
DlDisplayMode	• DisplayDistributed - Standardeinstellung, die Einrichtekorrekturen EC werden in der Werkzeugliste und die Summenkorrekturen SC werden in der Werkzeugverschleißliste angezeigt. • EcInWearList - die Einrichtekorrekturen EC werden in der Werkzeugverschleißliste angezeigt und die Summenkorrekturen SC werden nicht angezeigt.

Einträge	Bedeutung
CombinedToolMonitoring	Legt die Arten der kombinierten Überwachungsarten fest, die im Toggler angeboten werden: • Time/Count - Standzeit- und Stückzahlüberwachung • Time/Wear - Standzeit- und Verschleißüberwachung • Count/Wear - Stückzahl- und Verschleißüberwachung • Time/WearDlSc - Standzeit- und Verschleißüberwachung für Summen- und Einrichtekorrekturen • Count/WearDlSc - Stückzahl- und Verschleißüberwachung für Summen- und Einrichtekorrekturen • Time/Count/Wear - Standzeit-, Stückzahl- und Verschleißüberwachung • Time/Count/WearDlSc - Standzeit-, Stückzahl- und Verschleißüberwachung für Summen- und Einrichtekorrekturen
FiltersCombinedWithOr	Legt die Filterkriterien fest, die im Filterfenster durch ein logisches ODER verknüpft werden. Die Standardeinstellung ist für alle Filterkriterien ein logisches UND. Die möglichen Einträge sind: • "StandBy" - für einsatzbereite Werkzeuge • "Active" - für Werkzeuge mit Aktivkennung • "Locked" - für gesperrte Werkzeuge • "Prewarn" - für die Vorwarngrenze • "RemainingPieces" - für die Reststückzahl • "RemainingTime" - für die Reststandzeit • "MarkedToUnload" - für die Entladekennung • "MarkedToLoad" - für die Beladekennung
AddToGrindingSpindles	Legt die Spindelnummern fest, die bei der Auswahl der Schleifspindeln im Parameter "GrindingSpindleNo" ($TC_TPG1) angeboten werden sollen. Die Nummern der Schleifspindeln müssen durch ein Leerzeichen voneinander getrennt werden, z. B. -1 8 9. Standardanzeige: reale NC-Spindeln und 0
HideGrindingSpindles	Legt die Spindelnummern der realen NC-Spindeln fest, die bei der Auswahl der Schleifspindeln im Parameter "GrindingSpindleNo" ($TC_TPG1) nicht angeboten werden sollen. Die Nummern der Schleifspindeln müssen durch ein Leerzeichen voneinander getrennt werden, z. B. 1 3.
MagazineMoveMessage	true - Meldung "Magazinbewegung läuft" bzw. "Magazinbewegung beendet" wird angezeigt. false - Standardeinstellung, Meldung wird nicht angezeigt.
ToolLoadErrorMessage	Bezieht sich auf die Ausgabe der Fehlermeldung aufgrund einer negativen PLC. true - Standardeinstellung, Fehlermeldung wird angezeigt. false - Fehlermeldung wird nicht angezeigt.

Die folgenden Einstellungen beziehen sich auf Markierungen in den Listen:

true - Markierung wird angezeigt.

false - Markierung wird nicht angezeigt (Standardeinstellung).

Falls mehrere Markierungen auf einen Platz zutreffen, so wird die wichtigste Information angezeigt. Wenn z. B. ein Werkzeug gesperrt ist und gleichzeitig bei diesem Werkzeug die "Werkzeuglage" und die "Bezugsrichtung Halterwinkel" nicht zusammenpassen, erscheint das rote Kreuz für gesperrt. Das gelbe Dreieck für den Konflikt ist nicht zu sehen.

Einträge	Bedeutung
ShowSymbolActivePlace	Markierung des Magazinplatzes auf der Bearbeitungsposition. true - Standardeinstellung gilt für den Magazintyp "Revolver". false - Einstellung für alle anderen Magazintypen.
ShowSymbolActiveMultiToolPlace	Markierung des Multitoolplatzes auf der Bearbeitungsposition. Anzeige erfolgt nur, wenn sich das Multitool in der Spindel oder dem Werkzeughalter befindet. true - Standardeinstellung.
ShowSymbolLoadPlace	Markierung des Magazinplatzes auf der Beladestelle. true - Standardeinstellung gilt für alle Magazintypen (außer Revolver ist true). false - Einstellung für den Magazintyp "Revolver".
ShowSymbolChangeInSpindlePlace	Markierung des Magazinplatzes auf der Wechselstelle. true - Standardeinstellung gilt für alle Magazintypen, (außer Revolver ist true). false - Einstellung für den Magazintyp "Revolver".
ShowSymbolActiveTool	Markierung des aktiven Werkzeugs. false - Standardeinstellung.
ShowSymbolProgrammedTool	Markierung des Folgewerkzeugs. true - Standardeinstellung.
ShowSymbolActiveDNo	Markierung der aktiven Schneide. false - Standardeinstellung.
ShowSymbolActiveDNoInTool	Markierung des aktiven Werkzeugs durch die Markierung der aktiven Schneide. true - Standardeinstellung für Konfiguration ohne Magazinverwaltung. false - Einstellung für Konfiguration mit Magazinverwaltung.
ShowSymbolAcitveDlNo	Markierung der aktiven Dl-Nummer. true - Standardeinstellung.
ShowConflictPositionAndDirection	Markierung eines Konflikts, Werkzeuglage und Schnittrichtung passen nicht zusammen. true - Standardeinstellung.
ShowConflictActiveToolWarning	Markierung eines Konflikts, Magazintyp Revolver und das aktive Werkzeug befindet sich nicht auf der Bearbeitungsposition. true - Standardeinstellung.

Beispiel

```
<SETTINGS>
     <ReactivateWithMagPos value="false" type="bool" />
     <MagazineMoveMessage value="false" type="bool" />
     <CreateNewToolDialog value="true" type="bool" />
     <MagazineSelectionDialog value="false" type="bool" />
     <AccessLevelWriteDetailsAllParameter value="4" type="int" />
</SETTINGS>
```

12.1.2 Werkzeuglisten konfigurieren

Tag <LISTCONFIGURATION>

Im Tag <LISTCONFIGURATION> befinden sich die Einträge zur Konfiguration folgender Listenansichten:

- Fenster "Werkzeugliste", Kennung `<SlTmToollistForm>`, siehe 1. Beispiel.
- Fenster "Werkzeugverschleiß", Kennung `<SlTmToolwearForm>`
- Fenster "Magazin", Kennung `<SlTmToolmagazinForm>`
- Fenster "Werkzeugdaten OEM", Kennung `<SlTmTooloemForm>`
  Dieses Fenster ist in der Standardkonfiguration ausgeblendet. Das Fenster wird erst angezeigt, wenn Sie es mit `<Enabled>` frei schalten, siehe 3. Beispiel.
- Fenster "Werkzeugauswahl", Kennung `<SlTmToolselectionForm>`: Dieses Fenster kann in den Programmeingabemasken aufgeblendet werden, um ein Werkzeug auszuwählen.
- Fenster "Werkzeuge für Programm ... " Kennung `<SlTmTTDForm>`: Dieses Fenster wird verwendet um die zu einem Programm gespeicherten Werkzeuge (Werkzeugbedarfsdatei *.TTD) anzuzeigen.

Eintrag	Bedeutung
Enabled	true - Fenster wird angezeigt. false - Fenster ist ausgeblendet.
CAPTION	Festlegung des Listennamens.
Text	Wird als Kennung/Item unterhalb der "CAPTION" verwendet um den Text der Überschrift festzulegen.
COLUMNX	Spaltenauswahl, X steht für die Spaltennummer.
TOOLTYPE_ALL	Die Einstellung der Spalte gilt für alle Werkzeugtypen.
TOOLTYPE_ALL_XXX	Die Einstellung der Spalte gilt für alle Werkzeugtypen aus einer Gruppe. XXX steht für die Gruppennummer der Werkzeugtypen. Als Gruppennummer wird die volle Hunderterstelle verwendet, z. B. TOOLTYPE_ALL_100 oder TOOLTYPE_ALL_200 usw..
TOOLTYPE_XXX	Die Einstellung der Spalte gilt für einen bestimmten Werkzeugtyp. XXX steht für die Nummer des Werkzeugtyps.

Eintrag	Bedeutung
EMPTY_MAG_PLACE	Die Einstellung der Spalte gilt für einen leeren Magazinplatz. Leere Magazinplätze treten in einer nach Magazin sortierten Liste auf.
Item	Kennung des Listenparameters. Siehe Beispiel: Mit der Kennung "`MagPlaceType`" wird die Spalte angelegt, in der der Magazinplatztyp angezeigt wird. Kennungen der Werkzeugparameter (Seite 294) Kennungen der Schneidenparameter (Seite 297) Kennungen der Überwachungsparameter (Seite 300) Kennungen der Magazinplatzparameter (Seite 301)
DELETE_COLUMN	Spalte löschen.
INSERT_COLUMN	Spalte einfügen.

Hinweis

Die Einträge wirken unabhängig ihrer vorliegenden Programmierung in folgender Reihenfolge:

1. COLUMNX
2. DELETE_COLUMN
3. INSERT_COLUMN

Hinweis

Der Eintrag Enabled gilt nicht für das Fenster "Werkzeugauswahl". Die anderen Einträge wirken.

Beispiel 1: Magazinliste konfigurieren - Magazinplatz und Werkzeugplatztyp

Hinweis

Die Spalten der Listenkonfiguration lassen sich nicht in allen Fällen aus der Darstellung in der Oberfläche entnehmen, da einige Spalten automatisch eingefügt werden und daher nicht in die Zählung der Konfiguration eingehen.

Variante 1

Spalte 10 und Spalte 11 werden überschrieben. Wenn Spalte 10 und Spalte 11 nicht existiert, werden die neuen Spalten an die vorhandenen Spalten angefügt.

```
<LISTCONFIGURATION>
    <!-- **** Magazinliste *** -->
    <SlTmToolmagazinForm>
      <!-- **** Spalte 10 *** -->
      <COLUMN10>
        <!-- **** fuer alle Werkzeugtypen *** -->
        <TOOLTYPE_ALL>
          <!-- **** Magazinplatztyp anzeigen *** -->
```

```
            <Item value="MagPlaceType" type="QString" />
          </TOOLTYPE_ALL>
        </COLUMN10>
        <!-- **** Spalte 11 *** -->
        <COLUMN11>
          <!-- **** fuer alle Werkzeugtypen *** -->
          <TOOLTYPE_ALL>
            <!-- **** Werkzeugplatztyp anzeigen *** -->
            <Item value="ToolPlaceSpec" type="QString" />
          </TOOLTYPE_ALL>
          <!-- **** fuer alle Multitools *** -->
          <TOOLTYPE_9997>
            <!-- **** fuer alle Multitoolplatztyp anzeigen *** -->
            <Item value="MultiToolPlaceSpec" type="QString" />
          </TOOLTYPE_9997>
        </COLUMN11>
      </SlTmToolmagazinForm>
</LISTCONFIGURATION>
```

Variante 2

Die neuen Spalten werden in der Standardkonfiguration zwischen den Spalten "Magazinplatz gesperrt" und "Werkzeug übergross" eingefügt.

```
<LISTCONFIGURATION>
      <!-- **** Magazinliste *** -->
      <SlTmToolmagazinForm>
        <!-- **** Spalte eingefuegen *** -->
        <INSERT_COLUMN>
          <!-- **** Spalte 7 *** -->
          <COLUMN7>
            <!-- **** fuer alle Werkzeugtypen *** -->
            <TOOLTYPE_ALL>
              <!-- **** Magazinplatztyp anzeigen *** -->
              <Item value="MagPlaceType" type="QString" />
            </TOOLTYPE_ALL>
          </COLUMN7>
          <!-- **** Spalte 8 *** -->
          <COLUMN8>
            <!-- **** fuer alle Werkzeugtypen *** -->
            <TOOLTYPE_ALL>
              <!-- **** Werkzeugplatztyp anzeigen *** -->
              <Item value="ToolPlaceSpec" type="QString" />
            </TOOLTYPE_ALL>
            <!-- **** fuer alle Multitools *** -->
```

```
          <TOOLTYPE_9997>
            <!-- **** fuer alle Multitoolplatztyp anzeigen *** -->
            <Item value="MultiToolPlaceSpec" type="QString" />
          </TOOLTYPE_9997>
        </COLUMN8>
      </INSERT_COLUMN>
    </SlTmToolmagazinForm>
</LISTCONFIGURATION>
```

Beispiel 2: Magazinliste konfigurieren - Spalte "Werkzeug übergross" wird gelöscht

Hinweis

Die Spalten der Listenkonfiguration lassen sich nicht in allen Fällen aus der Darstellung in der Oberfläche entnehmen, da einige Spalten automatisch eingefügt werden und daher nicht in die Zählung der Konfiguration eingehen.

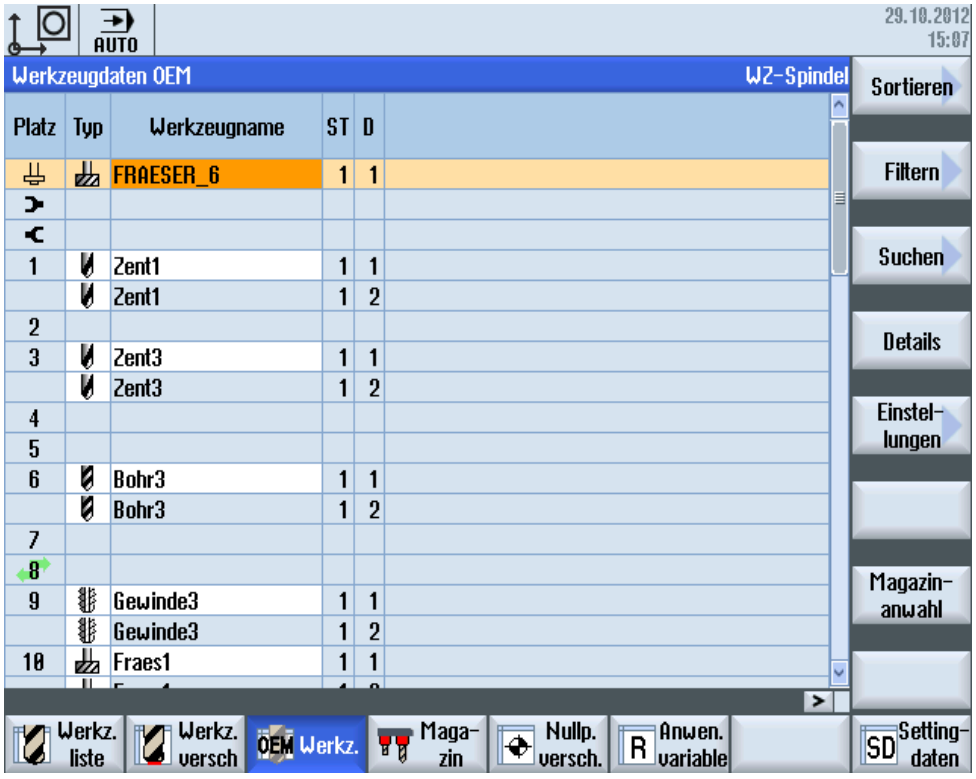
```
<LISTCONFIGURATION>
    <!-- **** Magazinliste *** -->
    <SlTmToolmagazinForm>
      <!-- **** Spalte 7 loeschen *** -->
      <DELETE_COLUMN>
        <COLUMN7 />
      </DELETE_COLUMN>
    </SlTmToolmagazinForm>
</LISTCONFIGURATION>
```

Beispiel 3: Liste Werkzeugdaten OEM freischalten

```
<SlTmTooloemForm>
      <Enabled value="true" type="bool" />
</SlTmTooloemForm>
```

Ergebnis

- Das Fenster "Werkzeugdaten OEM" wird freigeschaltet. Es zeigt automatisch die in der Abbildung dargestellten Spalten.
- Der neue Softkey "OEM Werkz." erscheint und ist bedienbar.



12.1.2.1 Kennungen der Werkzeugparameter

Werkzeugparameter

Für die Werkzeugparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaft	Systemvariable *)
Empty	Leeres Feld	-
ToolNo	Nummer des Werkzeugs	-
AdaptNo	Adapter Nummer Zuordnung	$TC_MPP7
DuploNo	Duplonummer	$TC_TP1
DuploNoRW	Duplonummer, lesender und schreibender Zugriff	$TC_TP1
NumCuttEdges	Anzahl Schneiden des Werkzeuges	$P_TOOLND
ToolIdent	Werkzeugname	$TC_TP2
ToolIdentRO	Werkzeugname, nur lesender Zugriff	$TC_TP2
ToolInfo	Werkzeuginformation	$TC_TP11
ToolInMag	Magazin, in dem sich das Werkzeug befindet	$A_TOOLMN

Kennung	Parameter bzw. Eigenschaft	Systemvariable *)
ToolInPlace	Magazinplatz, auf dem sich das Werkzeug befindet	$A_TOOLMLN
ToolInMagInPlace	Magazinnummer / Magazinplatz	$A_TOOLMN / $A_TOOL_MLN
ToolMonTCW	Art der Werkzeugüberwachung, Standzeit, Stückzahl und Verschleiß. Ist die Verschleißüberwachung über Maschinendatum nicht freigegeben, so wird dieser Parameter wie ToolMonTC behandelt.	$TC_TP9
ToolMonTC	Art der Werkzeugüberwachung, Standzeit und Stückzahl	$TC_TP9
ToolPlaceSpec	Platztyp	$TC_TP7
ToolPlaceSpecIdent	Platztyp, wenn er als Text angezeigt werden soll. Voraussetzung dafür ist, dass Texte für den Platztyp konfiguriert wurden.	$TC_TP7
ToolSearch	Art der Werkzeugsuche für Ersatzwerkzeuge	$TC_TP10
ToolMyMag	Eigentümer Magazin des Werkzeuges	$A_MYMN
ToolMyPlace	Eigentümer Magazinplatz des Werkzeuges	$A_MYMLN
ToolSizeLeft	Werkzeuggröße links in Halbplätzen	$TC_TP3
ToolSizeRight	Werkzeuggröße rechts in Halbplätzen	$TC_TP4
ToolSizeUpper	Werkzeuggröße oben in Halbplätzen	$TC_TP5
ToolSizeDown	Werkzeuggröße unten in Halbplätzen	$TC_TP6
ToolOverSize	Werkzeuggröße als feste Einstellung - zwei Halbplätze links, zwei Halbplätze rechts, ein Halbplatz oben, ein Halbplatz unten	$TC_TP3 - $TC_TP6
ToolState	Werkzeugstatus in hexadezimaler Schreibweise	$TC_TP8
ToolStateActiv	Werkzeugstatus - Werkzeug aktiv	$TC_TP8, Bit0
ToolStateEnabled	Werkzeugstatus - Werkzeug freigegeben	$TC_TP8, Bit1
ToolStateLocked	Werkzeugstatus - Werkzeug gesperrt	$TC_TP8, Bit2
ToolStateLockedRO	Werkzeugstatus - Werkzeug gesperrt, nur lesender Zugriff	$TC_TP8, Bit2
ToolStateMeasured	Werkzeugstatus - Werkzeug vermessen	$TC_TP8, Bit3
ToolStatePrewarn	Werkzeugstatus - Werkzeug hat Vorwarngrenze erreicht	$TC_TP8, Bit4
ToolStatePrewarnRO	Werkzeugstatus - Werkzeug hat Vorwarngrenze erreicht, nur lesender Zugriff	$TC_TP8, Bit4
ToolStateInChange	Werkzeugstatus - Werkzeug befindet sich im Wechsel	$TC_TP8, Bit5
ToolStateFixed	Werkzeugstatus - Werkzeug festplatzcodiert	$TC_TP8, Bit6
ToolStateUsed	Werkzeugstatus - Werkzeug war im Einsatz	$TC_TP8, Bit7
ToolStateAutoReturn	Werkzeugstatus - automatischer Rücktransport	$TC_TP8
ToolStateIgnoreLocked	Werkzeugstatus - gesperrt nicht beachten	$TC_TP8, Bit9
ToolStateMarkedToUnload	Werkzeugstatus - Werkzeug ist markiert zum Entladen	$TC_TP8, Bit10
ToolStateMarkedToLoad	Werkzeugstatus - Werkzeug ist markiert zum Beladen	$TC_TP8, Bit11

Kennung	Parameter bzw. Eigenschaft	Systemvariable *)
ToolStatePermanent	Werkzeugstatus – Werkzeug ist ein Stammwerkzeug	$TC_TP8, Bit12
ToolState1To1Exchange	Werkzeugstatus - eins zu eins Tausch	$TC_TP8, Bit14
ToolStateHandTool	Werkzeugstatus Handwerkzeug	$TC_TP8, Bit15
ToolProtAreaFile	Dateiname der Werkzeugbeschreibung, die zur Bestimmung des Schutzbereichs erforderlich ist	$TC_TP_PROTA
ToolMaxVelo	Maximaldrehzahl	$TC_TP_MAX_VELO
ToolMaxAcc	Maximalbeschleunigung	$TC_TP_MAX_ACC
ToolInMultiTool	Multitool, in dem sich das Werkzeug befindet	$A_TOOLMTN
ToolInCrossHead	Winkelkopf, in dem sich das Werkzeug befindet	$A_TOOLMTN
ToolInMultiToolPlace	Multitoolplatz, auf dem sich das Werkzeug befindet	$A_TOOLMTLN
ToolInCrossHeadPlace	Winkelkopfplatz, auf dem sich das Werkzeug befindet	$A_TOOLMTLN
ToolMyMultiTool	Eigentümer Multitool des Werkzeuges	$A_MYMTN
ToolMyCrossHead	Eigentümer Winkelkopf des Werkzeugs	$A_MYMTN
MultiToolPlaceAdaptNo	Adapter Nummer Zuordnung bezüglich des Multitool-/Winkelkopfplatzes	$TC_MTPP7
ToolMyMultiToolPlace	Eigentümer Multitoolplatz des Werkzeuges	$A_MYMTLN
ToolMyCrossHeadPlace	Eigentümer Winkelkopfplatz des Werkzeugs	$A_MYMTLN
ToolAlarmIsExtended	PLC Werkzeugstatus - Funktion "Verlängerter Alarm" aktiv	-
ToolAlarmLimit	PLC Werkzeugstatus - Alarmgrenze erreicht	-
ToolExtAlarmLimit	PLC Werkzeugstatus - Grenze "Verlängerter Alarm" erreicht	-
ToolUser_1, ... ToolUser_64	OEM-Werkzeugparameter 1 bis 64	$TC_TPC1, ... $TC_TPC64
ToolAppl_1, ... ToolAppl_10	Siemens Applikationswerkzeugparameter 1 bis 10	$TC_TPCS1, ... $TC_TPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

Hinweis

Werkzeugname "ToolIdent"

Die maximale Namenslänge von Werkzeugnamen beträgt 31 ASCII Zeichen.

Bei asiatischen Zeichen oder Unicode Zeichen verringert sich die Zeichenanzahl.

Folgende Sonderzeichen sind nicht zulässig: | # "

12.1.2.2 Kennungen der Schneidenparameter

Für die Schneidenparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
EdgeNo	D-Nummer	-
EdgeNoIdx	Schneidennummer	-
ToolType	Werkzeugtyp, Schneidenparameter 1	$TC_DP1
CuttEdgePos	Schneidenlage, Schneidenparameter 2	$TC_DP2
GeoLength1	Länge 1, Schneidenparameter 3	$TC_DP3
GeoLengthGeoAx1	Länge Geometrieachse 1, Schneidenparameter 3, z. B. Länge X	$TC_DP3
GeoLengthGeoAx1DiscDiameter	Schleifscheibendurchmesser in Geometrieachse 1, Schneidenparameter 3	$TC_DP3
GeoLengthGeoAx1DresserDiameter	Abrichterdurchmesser in Geometrieachse 1, Schneidenparameter 3	$TC_DP3
GeoLength	Länge Schneidenparameter 3	$TC_DP3
GeoLength2	Länge 2, Schneidenparameter 4	$TC_DP4
GeoLengthGeoAx3	Länge Geometrieachse 3, Schneidenparameter 4, z. B. Länge Z	$TC_DP4
GeoLengthGeoAx3DiscDiameter	Schleifscheibendurchmesser in Geometrieachse 3, Schneidenparameter 4	$TC_DP4
GeoLengthGeoAx3DresserDiameter	Abrichterdurchmesser in Geometrieachse 3, Schneidenparameter 4	$TC_DP4
GeoLength3	Länge 3, Schneidenparameter 5	$TC_DP5
GeoLengthGeoAx2	Länge Geometrieachse 2, Schneidenparameter 5, z. B. Länge Y	$TC_DP5
GeoLengthGeoAx2DiscDiameter	Schleifscheibendurchmesser in Geometrieachse 2, Schneidenparameter 5	$TC_DP5
GeoLengthGeoAx2DresserDiameter	Abrichterdurchmesser in Geometrieachse 2, Schneidenparameter 5	$TC_DP5
GeoRadius	Radius, Schneidenparameter 6	$TC_DP6
GeoOutsideRadius714	Außenradius für den Sterntaster, Werkzeugtyp 714, Schneidenparameter 6	$TC_DP6
GeoCuttEdgeRadius	Schneidenradius, Schneidenparameter 6	$TC_DP6
GeoCornerRadius	Eckenradius, Schneidenparameter 7	$TC_DP7
GeoOutsideRadius	Außenradius, Schneidenparameter 7	$TC_DP7
GeoBoomLength	Auslegerlänge, Schneidenparameter 7	$TC_DP7
GeoSlotWidth	Nutbreite z. B. für die Nutsäge, Schneidenparameter 7	$TC_DP7
GeoRadius714	Kugelradius für den Sterntaster, Werkzeugtyp 714, Schneidenparameter 7	$TC_DP7
GeoLength4	Länge 4, Schneidenparameter 8	$TC_DP8
PlateLength	Plattenlänge, Schneidenparameter 8	$TC_DP8
GeoProjection	Überstand z. B. für die Nutsäge, Schneidenparameter 8	$TC_DP8

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
GeoLength5	Länge 5, Schneidenparameter 9	$TC_DP9
GeoWidth	Breite, Schneidenparameter 9	$TC_DP9
PlateWidth	Plattenbreite, Schneidenparameter 9	$TC_DP9
GeoPitch	Gewindesteigung, Schneidenparameter 9	$TC_DP9
BoreRadius	Bohrradius, Schneidenparameter 9	$TC_DP9
TopConeDiameter	Oberer Kegeldurchmesser für die Werkzeugtypen 155, 156 und 157, Schneidenparameter 9	$TC_DP9
HolderAngle	Halterwinkel, Schneidenparameter 10	$TC_DP10
CorrectionAngle	Korrekturwinkel zum Ausrichten des Tasters, Schneidenparameter 10	$TC_DP10
GeoAngle2	Winkel 2, Schneidenparameter 11	$TC_DP11
AngleConicalMillTool	Winkel kegeliger Fräswerkzeuge, Schneidenparameter 11	$TC_DP11
CuttDirection	Bezugsrichtung Halterwinkel, Schneidenparameter 11	$TC_DP11
WearLength1	Verschleißlänge 1, Schneidenparameter 12	$TC_DP12
WearLengthGeoAx1	Verschleißlänge Geometrieachse 1, Schneidenparameter 12, z. B. Δ Länge X	$TC_DP12
WearLength	Verschleißlänge, Schneidenparameter 12	$TC_DP12
WearLength2	Verschleißlänge 2, Schneidenparameter 13	$TC_DP13
WearLengthGeoAx3	Verschleißlänge Geometrieachse 3, Schneidenparameter 13, z. B. Δ Länge Z	$TC_DP13
WearLength3	Verschleißlänge 3, Schneidenparameter 14	$TC_DP14
WearLengthGeoAx2	Verschleißlänge Geometrieachse 2, Schneidenparameter 14, z. B. Δ Länge Y	$TC_DP14
WearRadius	Verschleißradius, Schneidenparameter 15	$TC_DP15
WearOutsideRadius714	Verschleiß Außenradius für den Sterntaster, Werkzeugtyp 714, Schneidenparameter 15	$TC_DP15
WearCuttEdgeRadius	Verschleiß Schneidenradius, Schneidenparameter 15	$TC_DP15
WearCornerRadius	Verschleiß Eckenradius, Schneidenparameter 16	$TC_DP16
WearSlotWidth	Verschleiß Nutbreite z. B. für die Nutsäge, Schneidenparameter 16	$TC_DP16
WearLength4	Verschleißlänge 4, Schneidenparameter 17	$TC_DP17
WearProjection	Verschleiß Überstand z. B. für die Nutsäge, Schneidenparameter 17	$TC_DP17
WearLength5	Verschleißlänge 5, Schneidenparameter 18	$TC_DP18
WearAngle1	Verschleiß Winkel 1, Schneidenparameter 19	$TC_DP19

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
WearAngle2	Verschleiß Winkel 2, Schneidenparameter 20	$TC_DP20
AdaptLength1	Adapterlänge 1, Schneidenparameter 21	$TC_DP21
AdaptLengthGeoAx1	Adapterlänge Geometrieachse 1, Schneidenparameter 21, z. B. Adapter Länge X	$TC_DP21
AdaptLength	Adapterlänge, Schneidenparameter 21	$TC_DP21
AdaptLength2	Adapterlänge 2, Schneidenparameter 22	$TC_DP22
AdaptLengthGeoAx3	Adapterlänge Geometrieachse 3, Schneidenparameter 22, z. B. Adapter Länge Z	$TC_DP22
AdaptLength3	Adapterlänge 3, Schneidenparameter 23	$TC_DP23
AdaptLengthGeoAx2	Adapterlänge Geometrieachse 2, Schneidenparameter 23, z. B. Adapter Länge Y	$TC_DP23
ReliefAngle	Freiwinkel, Schneidenparameter 24	$TC_DP24
PlateAngle	Plattenwinkel, Kombination aus Schneidenparameter 24 und Schneidenparameter 10	$TC_DP24 und $TC_DP10
NoseAngle	Spitzenwinkel, Schneidenparameter 24	$TC_DP24
CuttRate	Schnittgeschwindigkeit, Schneidenparameter 25	$TC_DP25
SpindleDirection	Bei rotierenden Werkzeugen Spindeldrehrichtung der Werkzeugspindel. Bei Drehwerkzeugen Spindeldrehrichtung der Hauptspindel	$TC_DP25 Bit8 und Bit9
Coolant1	Kühlmittel 1	$TC_DP25 Bit10
Coolant2	Kühlmittel 2	$TC_DP25 Bit11
MFunction1	M-Funktion 1	$TC_DP25 Bit0
MFunction2	M-Funktion 2	$TC_DP25 Bit1
MFunction3	M-Funktion 3	$TC_DP25 Bit2
MFunction4	M-Funktion 4	$TC_DP25 Bit3
IsoHNoDPH	ISO H-Nummer	$TC_DPH
OrientNo	Schneidenorientierung	$TC_DPV
OrientV1	Schneidenorientierung Vektor 1	$TC_DPV3
OrientV2	Schneidenorientierung Vektor 2	$TC_DPV4
OrientV3	Schneidenorientierung Vektor 3	$TC_DPV5
OrientVGeoAx1	Schneidenorientierung Vektor Geometrieachse 1, Schneidenorientierungsparameter 3, z. B. Vektor X	$TC_DPV3
OrientVGeoAx3	Schneidenorientierung Vektor Geometrieachse 3, Schneidenorientierungsparameter 4, z. B. Vektor Z	$TC_DPV4
OrientVGeoAx2	Schneidenorientierung Vektor Geometrieachse 2, Schneidenorientierungsparameter 5, z. B. Vektor Y	$TC_DPV5
TeethCount	Zähnezahl, Schneidenparameter 34	$TC_DPNT
BasicRotationAngle	Einspannwinkel des Werkzeugs, Schneidenparameter 35	$TC_DPROT

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
UniqueDNo	Eindeutige D-Nummer	$TC_DPCE
EdgeUser_1, ..., EdgeUser_64	OEM-Schneidenparameter 1 bis 64	$TC_DPC1, ... $TC_DPC64
EdgeAppl_1, ..., EdgeAppl_10x	Siemens Applikationsschneidenparameter 1 bis 10	$TC_DPCS1, ... $TC_DPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.3 Kennungen der Überwachungsparameter

Für die Überwachungsparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
SupWarning	Vorwarngrenze, unter Berücksichtigung der aktuellen Überwachungsart	-
SupRemaining	Istwert, unter Berücksichtigung der aktuellen Überwachungsart	-
SupDesired	Sollwert, unter Berücksichtigung der aktuellen Überwachungsart	-
SupWarningTime	Vorwarngrenze Standzeit	$TC_MOP1
SupRemainingTime	Istwert Standzeit	$TC_MOP2
SupWarningPieces	Vorwarngrenze Stückzahl	$TC_MOP3
SupRemainingPieces	Istwert Stückzahl	$TC_MOP4
SupDesiredTime	Sollwert Standzeit	$TC_MOP11
SupDesiredPieces	Sollwert Stückzahl	$TC_MOP13
SupWarningWear	Vorwarngrenze Verschleiß	$TC_MOP5
SupRemainingWear	Istwert Verschleiß	$TC_MOP6
SupDesiredWear	Sollwert Verschleiß	$TC_MOP15
SupExtendedAlarm	PLC Werkzeugverwaltung: Verlängerter Alarm	-
SupWarningPiecesIncremental	PLC Werkzeugverwaltung: Vorwarngrenze Stückzahl	$TC_MOP13 - $TC_MOP3
SupActualPieces	PLC Werkzeugverwaltung: Istwert Stückzahl	$TC_MOP13 - $TC_MOP4
EdgeSupUser_1, ... EdgeSupUser_64	OEM-Schneidenüberwachungsparameter 1 bis 64	$TC_MOPC1, ... $TC_MOPC64
EdgeSupAppl_1, ... EdgeSupAppl_10	Siemens Applikationsschneidenüberwachungsparameter 1 bis 10	$TC_MOPCS1, ... $TC_MOPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.4 Kennungen der Schleifparameter

Für die Schleifparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung / Parameter	Bedeutung	Systemvariable *)
GrindingSpindleNo	Spindelnummer	$TC_TPG1
GrindingConnectionRule	Verkettungsvorschrift	$TC_TPG2
MinimalDiscRadius	minimaler Scheibenradius	$TC_TPG3
ActualDiscRadius	aktueller Scheibenradius (nur lesender Zugriff)	-
MinimalDiscWidth	minimale Scheibenbreite	$TC_TPG4
ActualDiscWidth	aktuelle Scheibenbreite	$TC_TPG5
MaximalDiscSpeed	maximale Scheibendrehzahl	$TC_TPG6
MaximalDiscPeripheralVelocity	maximale Scheibenumfangsgeschwindigkeit	$TC_TPG7
AngleBevelDisc	Winkel der schrägen Scheibe	$TC_TPG8
ParamForRadiusCalculation	Parameter für die Radiusberechnung	$TC_TPG9
DressingPath	Verzeichnispfad auf des Abrichtprogramm	$TC_TPG_DRSPATH
DressingProgram	Abrichtprogrammname	$TC_TPG_DRSPROG

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

Folgende Parameter sind den Schneiden des Schleifwerkzeugs zugeordnet:

Kennung / Parameter	Bedeutung
GrindingMonitoring	schleifspezifische Werkzeugüberwachung
GrindingUseBaseLength	Berücksichtigung des Basismaßes bei der Berechnung des Scheibenradius

12.1.2.5 Kennungen der Magazinplatzparameter

Für die Magazinplatzparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceKind	Platzart	$TC_MPP1
MagPlaceType	Platztyp	$TC_MPP2
MagPlaceTypeIdent	Platztyp als Text	$TC_MPP2
MagPlaceTNo	Werkzeugnummer des Werkzeugs auf diesem Platz	$TC_MPP6
MagPlaceWatchNeighbour	Nebenplatzbetrachtung	$TC_MPP3
MagPlaceStateLocked	Magazinplatz gesperrt	$TC_MPP4, Bit0
MagPlaceStateEmpty	Magazinplatz frei	$TC_MPP4, Bit1
MagPlaceStateResInterMag	Reserviert für Werkzeug im Zwischenspeicher	$TC_MPP4, Bit2
MagPlaceStateResLoadTool	Reserviert für zu beladendes Werkzeug	$TC_MPP4, Bit3
MagPlaceStateOccupiedLeft	Magazinplatz belegt, linker Halbplatz	$TC_MPP4, Bit4

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceStateOccupiedRight	Magazinplatz belegt, rechter Halbplatz	$TC_MPP4, Bit5
MagPlaceStateOccupiedUpper	Magazinplatz belegt, oberer Halbplatz	$TC_MPP4, Bit6
MagPlaceStateOccupiedDown	Magazinplatz belegt, unterer Halbplatz	$TC_MPP4, Bit7
MagPlaceStateReservedLeft	Magazinplatz reserviert, linker Halbplatz	$TC_MPP4, Bit8
MagPlaceStateReservedRight	Magazinplatz reserviert, rechter Halbplatz	$TC_MPP4, Bit9
MagPlaceStateReservedUpper	Magazinplatz reserviert, oberer Halbplatz	$TC_MPP4, Bit10
MagPlaceStateReservedDown	Magazinplatz reserviert, unterer Halbplatz	$TC_MPP4, Bit11
MagPlaceMagazineNo	Magazinnummer	-
MagPlaceTypeIdx	Artindex	$TC_MPP5
MagPlaceWearGroup	Nummer Verschleißverbund	$TC_MPP5
MagPlaceAdaptNo	Adapternummer	$TC_MPP7
MagPlaceNo	Magazinplatznummer	-
MagNoMagPlaceNo	Magazinnummer / Magazinplatznummer	-
MagPlaceUser_1, ... MagPlaceUser_64	OEM-Magazinplatzparameter 1 bis 64	$TC_MPPC1, ... $TC_MPPC64
MagPlaceAppl_1, ... MagPlaceAppl_10	Siemens Magazinplatzparameter 1 bis 10	$TC_MPPCS1, ... $TC_MPPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.6 Kennungen der Magazinplatzadapterparameter

Die Magazinplatzadapterparameter wurden neu organisiert. Die bisherigen Magazinplatzadapterparameter stehen nach wie vor zur Verfügung (siehe 1. Generation = Typ 1). Sie sollten aber in Zukunft nicht mehr verwendet werden und durch die Magazinplatzadapterparameter (2. Generation = Typ 2) ersetzt werden."

Für die Magazinplatzadapterparameter (1. Generation) werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterLength1	Magazinplatzadapterlänge 1	$TC_ADPT1
MagPlaceAdapterLengthGeoAx1	Magazinplatzadapterlänge Geometrieachse 1	$TC_ADPT1
MagPlaceAdapterLength2	Magazinplatzadapterlänge 2	$TC_ADPT2
MagPlaceAdapterLengthGeoAx3	Magazinplatzadapterlänge Geometrieachse 3	$TC_ADPT2
MagPlaceAdapterLength3	Magazinplatzadapterlänge 3	$TC_ADPT3
MagPlaceAdapterLengthGeoAx2	Magazinplatzadapterlänge Geometrieachse 2	$TC_ADPT3
MagPlaceAdapterTrafoNo	Magazinplatzadaptertransformationsnummer	$TC_ADPTT

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

Für die Magazinplatzadapterparameter (2. Generation) werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterNo	Nummer des Magazinplatzadapters	-
MagPlaceAdapterType	Typ des Magazinplatzadapters: - 1 Adapter für Magazinplätze oder - 2 Adapter für Winkelkopf-/Multitoolplätze -	$TC_ADPT_TYPE[n]
MagPlaceAdapterT1Length1	Länge 1 - Typ 1	$TC_ADPT1[n]
MagPlaceAdapterT1LengthGeoAx1	Länge Geometrieachse 1 - Typ 1	$TC_ADPT1[n]
MagPlaceAdapterT1Length2	Länge 2 - Typ 1	$TC_ADPT2[n]
MagPlaceAdapterT1LengthGeoAx3	Länge Geometrieachse 3 - Typ 1	$TC_ADPT2[n]
MagPlaceAdapterT1Length3	Länge 3 - Typ 1	$TC_ADPT3[n]
MagPlaceAdapterT1LengthGeoAx2	Länge Geometrieachse 2 - Typ 1	$TC_ADPT3[n]
MagPlaceAdapterT1TrafoNo	Transformationsnummer Typ 1	$TC_ADPTT[n]

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

1. Element des Winkelkopfadapters

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterL1_1	Offset – 1. Element, 1. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 0, 0]
MagPlaceAdapterL1GeoAx1	Offset – 1. Element, Geometrieachse 1 - Typ 2	$TC_ADPT_OFF[n, 0, 0]
MagPlaceAdapterL1_2	Offset – 1. Element, 2. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 0, 1]
MagPlaceAdapterL1GeoAx3	Offset – 1. Element, Geometrieachse 3 - Typ 2	$TC_ADPT_OFF[n, 0, 1]
MagPlaceAdapterL1_3	Offset – 1. Element, 3. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 0, 2]
MagPlaceAdapterL1GeoAx2	Offset – 1. Element, Geometrieachse 2 - Typ 2	$TC_ADPT_OFF[n, 0, 2]
MagPlaceAdapterDir1_1	Orientierung – 1. Rundachse, 1. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 0]
MagPlaceAdapterDir1GeoAx1	Orientierung – 1. Rundachse, Geometrieachse 1 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 0]
MagPlaceAdapterDir1_2	Orientierung – 1. Rundachse, 2. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 1]
MagPlaceAdapterDir1GeoAx3	Orientierung – 1. Rundachse, Geometrieachse 3 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 1]
MagPlaceAdapterDir1_3	Orientierung – 1. Rundachse, 3. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 2
MagPlaceAdapterDir1GeoAx2	Orientierung – 1. Rundachse, Geometrieachse 2 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 0, 2]
MagPlaceAdapterAng1	Winkel 1 - Typ 2	$TC_ADPT_ANG[n, 0]
CrossHeadAng1	Winkel 1 am Winkelkopf - Typ 2	$TC_ADPT_ANG[n, 0]
MagPlaceAdapterAngConst1	Winkel-Offset 1 - Typ 2	$TC_ADPT_ANG_CONST[n, 0]

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

2. Element des Winkelkopfadapters

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterL2_1	Offset – 2. Element, 1. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 1, 0]
MagPlaceAdapterL2GeoAx1	Offset – 2. Element, Geometrieachse 1 - Typ 2	$TC_ADPT_OFF[n, 1, 0]
MagPlaceAdapterL2_2	Offset – 2. Element, 2. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 1, 1]
MagPlaceAdapterL2GeoAx3	Offset – 2. Element, Geometrieachse 3 - Typ 2	$TC_ADPT_OFF[n, 1, 1]
MagPlaceAdapterL2_3	Offset – 2. Element, 3. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 1, 2]
MagPlaceAdapterL2GeoAx2	Offset – 2. Element, Geometrieachse 2 - Typ 2	$TC_ADPT_OFF[n, 1, 2]
MagPlaceAdapterDir2_1	Orientierung – 2. Rundachse, 1. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 0]
MagPlaceAdapterDir2GeoAx1	Orientierung – 2. Rundachse, Geometrieachse 1 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 0]
MagPlaceAdapterDir2_2	Orientierung – 2. Rundachse, 2. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 1]
MagPlaceAdapterDir2GeoAx3	Orientierung – 2. Rundachse, Geometrieachse 3 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 1]
MagPlaceAdapterDir2_3	Orientierung – 2. Rundachse, 3. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 2
MagPlaceAdapterDir2GeoAx2	Orientierung – 2. Rundachse, Geometrieachse 2 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 1, 2]
MagPlaceAdapterAng2	Winkel 2 - Typ 2	$TC_ADPT_ANG[n, 1]
CrossHeadAng2	Winkel 2 am Winkelkopf - Typ 2	$TC_ADPT_ANG[n, 1]
MagPlaceAdapterAngConst2	Winkel-Offset 2 - Typ 2	$TC_ADPT_ANG_CONST[n, 1]

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

3. Element des Winkelkopfadapters

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterL3_1	Offset – 3. Element, 1. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 2, 0]
MagPlaceAdapterL3GeoAx1	Offset – 3. Element, Geometrieachse 1 - Typ 2	$TC_ADPT_OFF[n, 2, 0]
MagPlaceAdapterL3_2	Offset – 3. Element, 2. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 2, 1]
MagPlaceAdapterL3GeoAx3	Offset – 3. Element, Geometrieachse 3 - Typ 2	$TC_ADPT_OFF[n, 2, 1]
MagPlaceAdapterL3_3	Offset – 3. Element, 3. Vektorkomponente - Typ 2	$TC_ADPT_OFF[n, 2, 2]
MagPlaceAdapterL3GeoAx2	Offset – 3. Element, Geometrieachse 2 - Typ 2	$TC_ADPT_OFF[n, 2, 2]
MagPlaceAdapterDir3_1	Orientierung – 3. Rundachse, 1. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 0]
MagPlaceAdapterDir3GeoAx1	Orientierung – 3. Rundachse, Geometrieachse 1 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 0]
MagPlaceAdapterDir3_2	Orientierung – 3. Rundachse, 2. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 1]
MagPlaceAdapterDir3GeoAx3	Orientierung – 3. Rundachse, Geometrieachse 3 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 1]
MagPlaceAdapterDir3_3	Orientierung – 3. Rundachse, 3. Vektorkomponente - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 2
MagPlaceAdapterDir3GeoAx2	Orientierung – 3. Rundachse, Geometrieachse 2 - Typ 2	Typ 2 - $TC_ADPT_DIR[n, 2, 2]

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MagPlaceAdapterAng3	Winkel 3 - Typ 2	$TC_ADPT_ANG[n, 2]
MagPlaceAdapterAngConst3	Winkel-Offset 3 - Typ 2	$TC_ADPT_ANG_CONST[n, 2]

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.7 Kennungen der Multitoolparameter

Für die Multitoolparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MultiToolNo	Nummer des Multitools	-
MultiToolNumberOfPlaces	Anzahl der Multitoolplätze	$TC_MTPN
MultiToolNumberOfPlacesRO	Anzahl der Multitoolplätze, nur lesender Zugriff	$TC_MTPN
MultiToolIdent	Name des Multitools	$TC_MTP2
MultiToolInMag	Magazin, in dem sich das Multitool befindet	$A_TOOLMN
MultiToolInPlace	Magazinplatz, auf dem sich das Multitool befindet	$A_TOOLMLN
MultiToolInMagInPlace	Magazinnummer / Magazinplatz	$A_TOOLMN / $A_TOOLMLN
MultiToolMyMag	Eigentümer Magazin des Multitools	$A_MYMN
MultiToolMyPlace	Eigentümer Magazinplatz des Multitools	$A_MYMLN
MultiToolPlaceSpec	Platztyp	$TC_MTP7
MultiToolPlaceSpecIdent	Platztyp als Text	$TC_MTP7
MultiToolSizeLeft	Multitoolgröße links in Halbplätzen	$TC_MTP3
MultiToolSizeRight	Multitoolgröße rechts in Halbplätzen	$TC_MTP4
MultiToolSizeUpper	Multitoolgröße oben in Halbplätzen	$TC_MTP5
MultiToolSizeDown	Multitoolgröße unten in Halbplätzen	$TC_MTP6
MultiToolOverSize	Multitoolgröße als feste Einstellung - zwei Halbplätze links, zwei Halbplätze rechts, ein Halbplatz oben, ein Halbplatz unten	$TC_MTP3 - $TC_MTP6
MultiToolPosition	Multitoolposition	$TC_MTP_POS
MultiToolProtAreaFile	Dateiname der Multitoolbeschreibung, die zur Bestimmung des Schutzbereichs erforderlich ist	$TC_MTP_PROTA
MultiToolKindOfDist	Art der Abstandseingabe der Multitoolplätze	$TC_MTP_KD
MultiToolKindOfDistRO	Art der Abstandseingabe der Multitoolplätze, nur lesender Zugriff	$TC_MTP_KD
MultiToolDistAngle	Art der Abstandseingabe der Multitoolplätze als Winkel	$TC_MTP_KD
MultiToolDistAngleRO	Art der Abstandseingabe der Multitoolplätze als Winkel, nur lesender Zugriff	$TC_MTP_KD

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MultiToolState	Multitoolstatus in hexadezimaler Schreibweise	$TC_MTP8
MultiToolStateEnabled	Multitoolstatus - Multitool freigegeben	$TC_MTP8, Bit1
MultiToolStateLocked	Multitoolstatus - Multitool gesperrt	$TC_MTP8, Bit2
MultiToolStateLockedRO	Multitoolstatus - Multitool gesperrt, nur lesender Zugriff	$TC_MTP8, Bit2
MultiToolStateInChange	Multitoolstatus - Multitool befindet sich im Wechsel	$TC_MTP8, Bit5
MultiToolStateFixed	Multitoolstatus - Multitool festplatzcodiert	$TC_MTP8, Bit6
MultiToolStateUsed	Multitoolstatus - Multitool war im Einsatz	$TC_MTP8, Bit7
MultiToolStateAutoReturn	Multitoolstatus - automatischer Rücktransport	$TC_MTP8, Bit8
MultiToolStateIgnoreLocked	Multitoolstatus - gesperrt nicht beachten	$TC_MTP8, Bit9
MultiToolStateMarkedToUnload	Multitoolstatus - Multitool ist markiert zum Entladen	$TC_MTP8, Bit10
MultiToolStateMarkedToLoad	Multitoolstatus - Multitool ist markiert zum Beladen	$TC_MTP8, Bit11
MultiToolStatePermanent	Multitoolstatus - Multitool ist ein Stammwerkzeug	$TC_MTP8, Bit12
MultiToolState1To1Exchange	Multitoolstatus - eins zu eins Tausch	$TC_MTP8, Bit14
MultiToolStateHandTool	Multitoolstatus - Handwerkzeug	$TC_MTP8, Bit15
MultiToolStateLockMtIfToolLock	Multitoolstatus - Multitool wird gesperrt falls ein Werkzeug im Multitool gesperrt wird	$TC_MTP8, Bit16
MultiToolUser_1, ... MultiToolUser_64	OEM-Multitoolparameter 1 bis 64	$TC_MTPC1, ... $TC_MTPC64
MultiToolAppl_1, ... MultiToolAppl_10	Siemens Applikations Multitoolparameter 1 bis 10	$TC_MTPCS1, ... $TC_MTPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.8 Kennungen der Multitoolplatzparameter

Für die Multitoolplatzparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MultiToolDist	Multitoolplatzabstand unter Berücksichtigung der aktuellen Art der Abstandseingabe des Multitools	-
MultiToolPlaceDistLength	Multitoolplatzabstand Länge	$TC_MTPPL
MultiToolPlaceDistAngle	Multitoolplatzabstand Winkel	$TC_MTPPA
MultiToolPlaceType	Platztyp	$TC_MTPP2
MultiToolPlaceTypeIdent	Platztyp als Text	$TC_MTPP2
MultiToolPlaceState	Multitoolplatzstatus in hexadezimaler Schreibweise	$TC_MTPP4

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
MultiToolPlaceStateLocked	Multitoolplatz gesperrt	$TC_MTPP4, Bit0
MultiToolPlaceStateEmpty	Multitoolplatz frei	$TC_MTPP4, Bit1
MultiToolPlaceTNo	Werkzeugnummer des Werkzeugs auf diesem Multitoolplatz	$TC_MTPP6
MultiToolPlaceAdaptNo	Adapternummer	$TC_MTPP7
MultiToolPlaceNo	Multitoolplatznummer	-
MultiToolPlaceUser_1, ... MultiToolPlaceUser_64	OEM-Multitoolplatzparameter 1 bis 64	$TC_MTPPC1, ... $TC_MTPPC64
MultiToolPlaceAppl_1, ... MultiToolPlaceAppl_10	Siemens Applikations Multitoolplatzparameter 1 bis 10	$TC_MTPPCS1, ... $TC_MTPPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.9 Kennungen der Winkelkopfparameter

Für die Winkelkopfparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
CrossHeadNo	Nummer des Winkelkopfs	-
CrossHeadNumberOfPlaces	Anzahl der Winkelkopfplätze	$TC_MTPN
CrossHeadNumberOfPlacesRO	Anzahl der Winkelkopfplätze, nur lesender Zugriff	$TC_MTPN
CrossHeadIdent	Name des Winkelkopfs	$TC_MTP2
CrossHeadInMag	Magazin, in dem sich der Winkelkopf befindet	$A_TOOLMN
CrossHeadInPlace	Magazinplatz, auf dem sich der Winkelkopf befindet	$A_TOOLMLN
CrossHeadInMagInPlace	Magazinnummer / Magazinplatz	$A_TOOLMN / $A_TOOLMLN
CrossHeadMyMag	Eigentümer Magazin des Winkelkopfs	$A_MYMN
CrossHeadMyPlace	Eigentümer Magazinplatz des Winkelkopfs	$A_MYMLN
CrossHeadPlaceSpec	Platztyp	$TC_MTP7
CrossHeadPlaceSpecIdent	Platztyp als Text	$TC_MTP7
CrossHeadSizeLeft	Winkelkopfgröße links in Halbplätzen	$TC_MTP3
CrossHeadSizeRight	Winkelkopfgröße rechts in Halbplätzen	$TC_MTP4
CrossHeadSizeUpper	Winkelkopfgröße oben in Halbplätzen	$TC_MTP5
CrossHeadSizeDown	Winkelkopfgröße unten in Halbplätzen	$TC_MTP6
CrossHeadOverSize	Winkelkopfgröße als feste Einstellung - zwei Halbplätze links, zwei Halbplätze rechts, ein Halbplatz oben, ein Halbplatz unten	$TC_MTP3 - $TC_MTP6
CrossHeadPosition	Winkelkopfposition	$TC_MTP_POS
CrossHeadProtAreaFile	Dateiname der Winkelkopfbeschreibung, die zur Bestimmung des Schutzbereichs erforderlich ist	$TC_MTP_PROTA

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
CrossHeadKindOfDist	Art der Abstandseingabe der Winkelkopfplätze	$TC_MTP_KD
CrossHeadKindOfDistRO	Art der Abstandseingabe der Winkelkopfplätze, nur lesender Zugriff	$TC_MTP_KD
CrossHeadDistAngle	Art der Abstandseingabe der Winkelkopfplätze als Winkel	$TC_MTP_KD
CrossHeadDistAngleRO	Art der Abstandseingabe der Winkelkopfplätze als Winkel, nur lesender Zugriff	$TC_MTP_KD
CrossHeadState	Winkelkopfstatus in hexadezimaler Schreibweise	$TC_MTP8
CrossHeadStateEnabled	Winkelkopfstatus – Winkelkopf frei gegeben	$TC_MTP8, Bit1
CrossHeadStateLocked	Winkelkopfstatus – Winkelkopf gesperrt	$TC_MTP8, Bit2
CrossHeadStateLockedRO	Winkelkopfstatus – Winkelkopf gesperrt, nur lesender Zugriff	$TC_MTP8, Bit2
CrossHeadStateInChange	Winkelkopfstatus – Winkelkopf befindet sich im Wechsel	$TC_MTP8, Bit5
CrossHeadStateFixed	Winkelkopfstatus – Winkelkopf festplatzcodiert	$TC_MTP8, Bit6
CrossHeadStateUsed	Winkelkopfstatus – Winkelkopf war im Einsatz	$TC_MTP8, Bit7
CrossHeadStateAutoReturn	Winkelkopfstatus – automatischer Rücktransport	$TC_MTP8, Bit8
CrossHeadStateIgnoreLocked	Winkelkopfstatus – gesperrt nicht beachten	$TC_MTP8, Bit9
CrossHeadStateMarkedToUnload	Winkelkopfstatus – Winkelkopf ist markiert zum Entladen	$TC_MTP8, Bit10
CrossHeadStateMarkedToLoad	Winkelkopfstatus – Winkelkopf ist markiert zum Beladen	$TC_MTP8, Bit11
CrossHeadStatePermanent	Winkelkopfstatus – Winkelkopf ist ein Stammwerkzeug	$TC_MTP8, Bit12
CrossHeadState1To1Exchange	Winkelkopfstatus – eins zu eins Tausch	$TC_MTP8, Bit14
CrossHeadStateHandTool	Winkelkopfstatus – Handwerkzeug	$TC_MTP8, Bit15
CrossHeadStateLockMtIfToolLock	Winkelkopfstatus – Winkelkopf wird gesperrt falls ein Werkzeug im Winkelkopf gesperrt	$TC_MTP8, Bit16
CrossHeadUser_1, ... CrossHeadUser_64	OEM-Winkelkopfparameter 1 bis 64	$TC_MTPC1, ... $TC_MTPC64
CrossHeadAppl_1, ... CrossHeadAppl_10	Siemens Applikations Winkelkopfparameter 1 bis 10	$TC_MTPCS1, ... $TC_MTPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.2.10 Kennungen der Winkelkopfplatzparameter

Für die Winkelkopfplatzparameter werden folgende Kennungen in der Konfigurationsdatei verwendet:

Kennung	Parameter bzw. Eigenschaften	Systemvariable *)
CrossHeadDist	Winkelkopfplatzabstand unter Berücksichtigung der aktuellen Art der Abstandseingabe des Winkelkopfs	-
CrossHeadPlaceDistLength	Winkelkopfplatzabstand Länge	$TC_MTPPL
CrossHeadPlaceDistAngle	Winkelkopfplatzabstand Winkel	$TC_MTPPA
CrossHeadPlaceType	Platztyp	$TC_MTPP2
CrossHeadPlaceTypeIdent	Platztyp als Text	$TC_MTPP2
CrossHeadPlaceState	Winkelkopfplatzstatus in hexadezimaler Schreibweise	$TC_MTPP4
CrossHeadPlaceStateLocked	Winkelkopfplatz gesperrt	$TC_MTPP4, Bit0
CrossHeadPlaceStateEmpty	Winkelkopfplatz frei	$TC_MTPP4, Bit1
CrossHeadPlaceTNo	Werkzeugnummer des Werkzeugs auf diesem Winkelkopfplatz	$TC_MTPP6
CrossHeadPlaceAdaptNo	Adapternummer	$TC_MTPP7
CrossHeadPlaceNo	Winkelkopfplatznummer	-
CrossHeadPlaceUser_1, ... CrossHeadPlaceUser_64	OEM-Winkelkopfplatzparameter 1 bis 64	$TC_MTPPC1, ... $TC_MTPPC64
CrossHeadPlaceAppl_1, ... CrossHeadPlaceAppl_10	Siemens Applikations Winkelkopfplatzparameter 1 bis	$TC_MTPPCS1, ... $TC_MTPPCS10

*) Parameter entspricht der Systemvariablen bzw. die Systemvariable bildet die Grundlage zu diesem Parameter.

12.1.3 Listenparameter konfigurieren

Tag <PARAMETERCONFIGURATION>

Im Tag <PARAMETERCONFIGURATION> haben Sie folgende Möglichkeiten:

1. Listenparameter ändern.
2. Neuen Listenparameter auf Basis eines vorhandenen Parameters anlegen.

Listenparameter verändern

Sie können alle im System erfassten Listenparameter verändern.

Kennungen der Werkzeugparameter (Seite 294)

Kennungen der Schneidenparameter (Seite 297)

Kennungen der Überwachungsparameter (Seite 300)

Kennungen der Schleifparameter (Seite 301)

Kennungen der Magazinplatzparameter (Seite 301)
Kennungen der Magazinplatzadapterparameter (Seite 302)
Kennungen der Multitoolparameter (Seite 305)
Kennungen der Multitoolplatzparameter (Seite 306)

AUTOHOTSPOT

AUTOHOTSPOT

Einträge	Bedeutung	
HeadLine	Spaltenüberschrift. Eingegebener Text wird als Überschrift angezeigt. Beispiel: Text "Radius" wird als Überschrift angezeigt (Standard).	
ToolTip	Text, der im Tooltip angezeigt wird. Beispiel: Text "Geometrie Radius" wird im Tooltip angezeigt (Standard).	
ShortText	Text, wenn der Parameter im Fenster "Weitere Daten" angezeigt wird. Beispiel: Text "Rad." wird im Fenster "Weitere Daten" angezeigt (Standard).	
DetailsText	Text, wenn der Parameter im Fenster "Details" angezeigt wird. Beispiel: Text "Radius" wird im Fenster "Details" angezeigt (Standard).	
Width	Spaltenbreite in Pixel bezogen auf eine Auflösung 640x480. Siehe nachfolgendes Beispiel: Die Standard-Spaltenbreite wird auf 53 Pixel verändert.	
DisplayMode	Werte, mit dem der Parameter angezeigt wird. Siehe nachfolgendes Beispiel: Die Standardeinstellung wird in "DoubleMode" geändert. Es können folgende weiteren Werte angenommen werden:	
	AnyMode	Alle Zeichen
	IntegerMode	Ganzzahlige Werte
	UnsignedIntegerMode	Ganzzahlige Werte ohne Vorzeichen
	DoubleMode	Werte mit Nachkommastellen. Festlegung der Anzahl der Nachkommastellen im "DecimalPlaces".
	UnsignedDoubleMode	Werte mit Nachkommastellen ohne Vorzeichen Festlegung der Anzahl der Nachkommastellen in "DecimalPlaces"
	Length	Längenangabe
	Angle	Winkelangabe
	LinearFeedPerTime	Linearvorschub mm/Minute
	LinearFeedPerRevolution	Linearvorschub mm/Umdrehung
	LinearFeedPerTooth	Linearvorschub mm/Zahn
	RevolutionSpeed	Drehzahl
	ConstantCuttingSpeed	Konstante Schnittgeschwindigkeit
DecimalPlaces	Anzahl der Nachkommastellen, wenn bei DisplayMode "DoubleMode" oder "UnsignedDoubleMode" angewählt ist. Siehe Beispiel: 2 Nachkommastellen sind eingetragen.	

Einträge	Bedeutung	
ItemType	Feldtypen, in denen ein Wert dargestellt wird.	
	TextField	Ein-/Ausgabefeld für Werte und Texte
	TextFieldReadOnly	Ausgabefeld für Werte und Texte
	CheckBox	Ein-/Ausgabefeld für Zustände
	CheckBoxReadOnly	Ausgabefeld für Zustände
BitMask	Bitmaske zur Anzeige eines Bits aus einem Wert. Die Bitmaske wird als Integerwert eingetragen. Bit 0 → 1, Bit 1 → 2, Bit 2 → 4	
AccessLevel	Wert für die Zugriffsstufe, die für diesen Parameter in der Bedienoberfläche verwendet wird. Wertebereich 1 bis 7	
UpperLimit	Obere Eingabegrenze für diesen Parameter in der Bedienoberfläche. Gilt nur für Zahleneingabeparameter.	
LowerLimit	Untere Eingabegrenze für diesen Parameter in der Bedienoberfläche. Gilt nur für Zahleneingabeparameter.	
Attribute	Besondere Eigenschaft des Parameters	
	EmptyInMultiTool	Parameter wird auf einem Multitool- bzw. Winkelkopfplatz ausgeblendet

Beispiel

Im nachfolgenden Beispiel wird der Parameter `<GeoRadius>` verwendet.

Geben Sie nur die geänderten Daten an.

```
<PARAMETERCONFIGURATION>
     <GeoRadius>
          <Width value="53" type="int" />
          <DisplayMode value="DoubleMode" type="QString" />
          <DecimalPlaces value="2" type="int" />
     </GeoRadius>
</PARAMETERCONFIGURATION>
```

Neue Listenparameter auf Basis eines Vorhandenen konfigurieren

Vergeben Sie einen neuen Parameternamen und tragen Sie nur die geänderten Daten ein:

Einträge	Bedeutung
	Neuen Parameternamen vergeben Siehe 1. Beispiel: Text "`NewGeoRadius`"
Base	Name des Parameters, auf dem der neue Parameter basiert. Siehe 1. Beispiel: Parameter "`GeoRadius`" wird als Vorlage verwendet.
	Tragen Sie auch hier nur die geänderten Daten ein. Alle anderen Daten werden vom vorhandenen Parameter übernommen. Siehe 1. Beispiel: Geänderte Spaltenbreite auf 46 Pixel. Anzahl der Nachkommastellen auf 1 geändert.

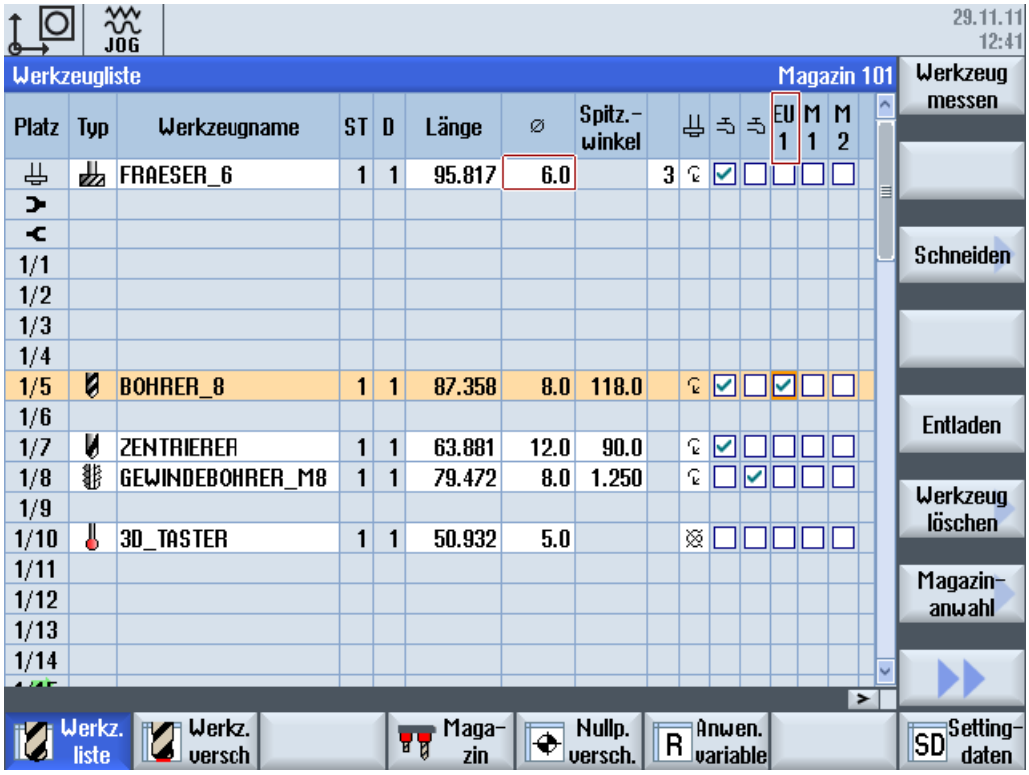
1. Beispiel

```
<PARAMETERCONFIGURATION>
    <NewGeoRadius>
        <Base value="GeoRadius" type="QString" />
        <Width value="46" type="int" />
        <DisplayMode value="DoubleMode" type="QString" />
        <DecimalPlaces value="1" type="int" />
    </NewGeoRadius>
</PARAMETERCONFIGURATION>
```

2. Beispiel

Einträge	Bedeutung
	Neuen Schneidenparameternamen vergeben Im nachfolgenden Beispiel lautet der Name "EdgeUser_1_Bit0".
Base	Name des Parameters, auf dem der neue Parameter basiert. Im Beispiel wird der Parameter "EdgeUser_1" als Vorlage verwendet.
	Tragen Sie auch hier nur die geänderten Daten ein. Alle anderen Daten werden vom vorhandenen Schneidenparameter übernommen.
	Folgende Einträge werden geändert: Feldtyp: Ein-/ und Ausgabefeld für Zustände Werteausgabe: alle Zeichen Bitmaske: Bit 0 Spaltenbreite wird auf 17 Pixel geändert Überschrift nach "TM_HL_EDGE_USER_1_Bit0" geändert Text im Tooltip nach "TM_TT_EDGE_USER_1_Bit0" geändert

```
<PARAMETERCONFIGURATION>
    <EdgeUser_1_Bit0>
        <Base value="EdgeUser_1" type="QString" />
        <ItemType value="CheckBox" type="QString" />
        <DisplayMode value="AnyMode" type="QString" />
        <BitMask value="1" type="int" />
        <Width value="17" type="int" />
        <HeadLine value="TM_HL_EDGE_USER_1_Bit0" type="QString" />
        <ToolTip value="TM_TT_EDGE_USER_1_Bit0" type="QString" />
    </EdgeUser_1_Bit0>
</PARAMETERCONFIGURATION>
```



12.1.4 Liste der Werkzeugtypen

Aufschlüsselung der Werkzeugtypen für Fräswerkzeuge

Gruppe mit Typ 1xy (Fräser):

100	Fräswerkzeug nach CLDATA (DIN 66215)
110	Kugelkopffräser (zylindrischer Gesenkfräser)
111	Kugelkopffräser (kegeliger Gesenkfräser)
120	Schaftfräser (ohne Eckenverrundung)
121	Schaftfräser (mit Eckenverrundung)
130	Winkelkopffräser (ohne Eckenverrundung)
131	Winkelkopffräser (mit Eckenverrundung)
140	Planfräser
145	Gewindefräser
150	Scheibenfräser
151	Säge
155	Kegelstumpffräser (ohne Eckenverrundung)
156	Kegelstumpffräser (mit Eckenverrundung)
157	Kegeliger Gesenkfräser
160	Bohrgewindefräser

Aufschlüsselung der Werkzeugtypen für Bohrer

Gruppe Typ 2xy (Bohrer):

200	Spiralbohrer
205	Vollbohrer
210	Bohrstange
220	Zentrierbohrer
230	Spitzsenker
231	Flachsenker
240	Gewindebohrer Regelgewinde
241	Gewindebohrer Feingewinde
242	Gewindebohrer Withworth-Gewinde
250	Reibahle

Aufschlüsselung der Werkzeugtypen für Schleifwerkzeuge

Gruppe Typ 4xy (Schleifwerkzeuge):

400	Umfang-Schleifscheibe
410	Plan-Schleifscheibe
490	Abrichter
494	Abrichtrolle für das Rundschleifen
495	Abrichtrolle für das Flachschleifen
496	Abrichtrad für das Rundschleifen
497	Abrichtrad für das Flachschleifen

Die Schleifwerkzeugtypen 401, 402, 403 sowie 411, 412, 413 werden in der Auswahl der Werkzeugtypen nicht angeboten. Die schleifspezifische Werkzeugüberwachung und die Berücksichtigung des Basismaßes bei der Berechnung des Scheibenradius wird über eigene Parameter eingestellt. Diese Parameter wirken auf die Werkzeugtypen in der gewohnten Form.

Aufschlüsselung der Werkzeugtypen für Drehwerkzeuge

Gruppe Typ 5xy (Drehwerkzeuge):

500	Schruppstahl
510	Schlichtstahl
520	Einstechstahl
530	Abstechstahl
540	Gewindestahl
550	Formstahl
560	Drehbohrer (ECOCUT)
580	Orientierter Messtaster
585	Kalibrierwerkzeug

Aufschlüsselung der Werkzeugtypen für Sonderwerkzeuge

Gruppe Typ 7xy (Sonderwerkzeuge):

700	Nutsäge
710	3D-Messtaster
711	Kantentaster
712	Monotaster
713	L-Taster
714	Sterntaster
725	Kalibrierwerkzeug
730	Anschlag
731	Pinole
732	Lünette
900	Hilfswerkzeuge

Aufschlüsselung der Werkzeugtypen für Winkelkopf und Multitool

Bei den Werkzeugtypen für Winkelkopf und Multitool handelt es sich nicht um echte Werkzeugtypen im Sinne der NCK. Die Kennungen dienen lediglich als Hilfsmittel um den Winkelkopf und das Multitool bei der Konfiguration der Werkzeuglisten ansprechen zu können.

9994	Winkelkopf
9997	Multitool

12.1.5 Werkzeugtypen konfigurieren

Tag <TOOLTYPECONFIGURATION>

Im Tag <TOOLTYPECONFIGURATION> befinden sich die Einträge zur Konfiguration der Werkzeugtypen:

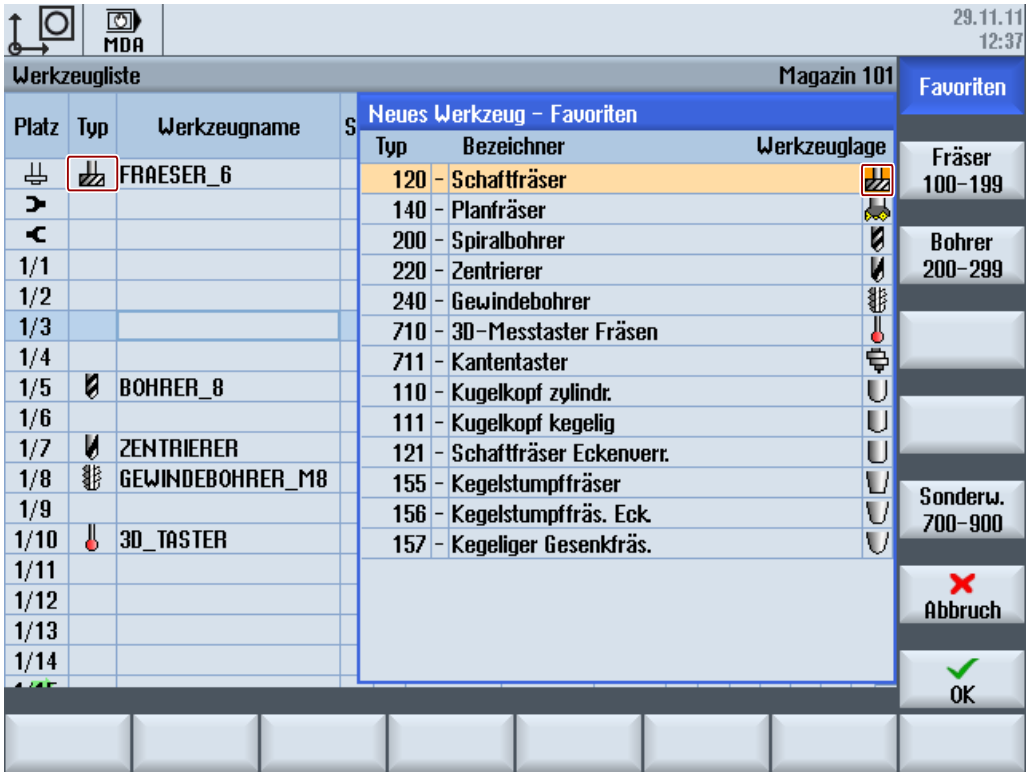
Einträge	Bedeutung
TOOLTYPE_XXX	XXX steht für die Nummer des Werkzeugtyps. Die Zuordnung von Werkzeugtyp und Nummer entnehmen Sie dem Kapitel "Liste der Werkzeugtypen (Seite 313)".
Tooltype	Nummer (XXX) des Werkzeugtyps

Einträge	Bedeutung
Name	Textkennung des Werkzeugnamens. Werkzeugname wird in folgenden Fenstern angezeigt: • "Neues Werkzeug - Favoriten" • "Neues Werkzeug - Fräser 100-199" • "Neues Werkzeug - Bohrer 200-299" • "Neues Werkzeug - Schleifwerkzeug 400-499" • "Neues Werkzeug - Sonder- und Hilfswerkzeuge 700-900" Siehe Beispiel: Für die Textkennung "`TM_PAR_SHANK_END_CUTTER`" erscheint der Name "Schaftfräser" auf der Bedienoberfläche.
Shortname	Textkennung des Werkzeugnamens. Werkzeugname wird in der Werkzeugliste angezeigt. Siehe Beispiel: Für die Textkennung "`TM_PPTT_SHANK_END_CUTTER`" erscheint der Name "FRAESER" auf der Bedienoberfläche.
Tooltip	Textkennung des Werkzeugnamens. Werkzeugname wird im Tooltip angezeigt. Siehe Beispiel: Für die Textkennung "`TM_TTTT_SHANK_END_CUTTER`" erscheint der Name "Schaftfräser" auf der Bedienoberfläche.
Icon9 -für die Technologie Fräsen	Symbol, das den Werkzeugtyp darstellt. Die Symbole werden in folgenden Fenstern angezeigt: • "Werkzeugliste" in der Spalte "Typ" • "Neues Werkzeug - Favoriten" in der Spalte "Werkzeuglage" • "Neues Werkzeug - Fräser 100-199" in der Spalte "Werkzeuglage" • "Neues Werkzeug - Bohrer 200-299" in der Spalte "Werkzeuglage" • "Neues Werkzeug - Sonderw. 700-900 in der Spalte "Werkzeuglage" Die Symbole sind im Format ".png". Sie werden entsprechend der Bildauflösung in folgenden Verzeichnissen abgelegt: /**oem**/sinumerik/hmi/ico/ico640, oder ico800, oder ico1024 /**user**/sinumerik/hmi/ico/ico640, oder ico800, oder ico1024
IconX - für die Technologien Drehen und Schleifen	Symbol, das den Werkzeugtyp darstellt (siehe Icon9 - für die Technologie Fräsen) Besonderheit - in den Technologien Drehen und Schleifen werden Werkzeuglagen unterstützt. Das X steht für die Werkzeuglage, die mit dem Icon dargestellt wird. Die Lage 9 ist eine undefinierte Lage und wird mit einem Kreuz dargestellt.
Icon9_2	Symbol, das den Werkzeugtyp in der Schneidenlage 9 darstellt. Es wird verwendet, wenn die Schneidenlage 9 als gültige Schneidenlage behandelt werden soll. (siehe <SETTINGS> - CreateNewToolWithCuttPos9Enabled, Kapitel "Allgemeine Einstellungen (Seite 284)")
Iconorder	Reihenfolge, in der die Lagen des Werkzeugs, dargestellt durch Symbole, umgeschaltet werden.
DisableInNewToolDialog	Mit diesem Eintrag wird der betroffene Werkzeugtyp beim Anlegen neuer Werkzeuge nicht angeboten. Trifft die Werkzeugliste auf ein bereits vorhandenes Werkzeug mit diesem Typ, dann wird es entsprechend angezeigt.

Einträge	Bedeutung
DELETE_TOOLTYPE	Alle vordefinierten Werkzeugtypen, die in den Werkzeuglisten nicht unterstützt werden sollen.
GrindingTooltypes	Alle Werkzeugtypen, die als Schleifwerkzeugtypen verwendet werden sollen.

Beispiel 1

```
<TOOLTYPECONFIGURATION>
     <TOOLTYPE_120>
       <Tooltype value="120" type="uint" />
       <Name value="TM_PAR_SHANK_END_CUTTER" type="QString" />
       <Shortname value="TM_PPTT_SHANK_END_CUTTER" type="QString" />
       <Tooltip value="TM_TTTT_SHANK_END_CUTTER" type="QString" />
       <Icon9 value="to_poly_shank_end_cutter_down.png" type="QString" />
     </TOOLTYPE_120>
</TOOLTYPECONFIGURATION>
```



Beispiel 2

```
<TOOLTYPECONFIGURATION>
     <DELETE_TOOLTYPE>
          <TOOLTYPE_110 />
```

```
    </DELETE_TOOLTYPE>
</TOOLTYPECONFIGURATION>
```

12.1.6 Fenster "Weitere Daten" konfigurieren

Tag <MOREDATACONFIGURATION>

Im Tag <MOREDATACONFIGURATION> befinden sich die Einträge zur Konfiguration des Fensters "Weitere Daten". Zu jedem Werkzeugtyp können in mehreren Zeilen und Spalten verschiedene weitere Daten angezeigt werden. Jedes anzuzeigende Datum wird durch den Eintrag eines Listenparameters spezifiziert:

Kennungen der Werkzeugparameter (Seite 294)

Kennungen der Schneidenparameter (Seite 297)

Kennungen der Überwachungsparameter (Seite 300)

Im Fenster erscheinen dann eine Kurzbeschreibung des Parameters (ShortText) und daneben der Wert selber. Listenparameter konfigurieren (Seite 309).

Wird ein weiterer Beschreibungstext im Fenster "Weitere Daten" benötigt, so kann auch ein Text als Eintrag in eine Zeile/Spalte eingetragen werden. Dieser Text nimmt im Fenster die gleiche Breite ein, wie die Kurzbeschreibung und der Wert eines Parameters zusammen.

Es können beliebig viele Zeilen und Spalten erstellt werden. Ab einer bestimmten Größe erhält das Fenster einen Scrollbalken.

Einträge	Bedeutung
TOOLTYPE_XXX	Nummer des Werkzeugtyps. Siehe Beispiel: Werkzeugtyp 111= Kugelkopf (kegeliger Gesenkfräser).
ROWX_COLY	Jeweilige ZeileX_SpalteY. Steht hinter der ZeileX und der SpalteY kein Eintrag, bleibt dieses Feld leer. Siehe Beispiel: Die erste Zeile ist im Fenster "Weitere Daten" leer.
Item	Folgende Einträge sind möglich: • Kennung des Parameters • Text Siehe Beispiel: In der zweiten Zeile erscheint die Überschrift "Eckenradius" im Fenster. In der dritten Zeile erscheinen der Text "Radius" und ein Eingabefeld für den Wert im Fenster. Die vierte Zeile ist leer.
TOOLTYPE_ALL_XXX	Die Einstellung gilt für alle Werkzeugtypen aus einer Gruppe. XXX steht für die Gruppennummer der Werkzeugtypen. Als Gruppennummer wird die volle Hunderterstelle verwendet, z. B. TOOLTYPE_ALL_100 oder TOOL-TYPE_ALL_200 usw. (siehe Beispiel).
WITHOUT_MORE_DATA	Werkzeugtypen einer Werkzeuggruppe, für die keine "weiteren Daten" angezeigt werden sollen, können mit dieser Kennung ausgeschlossen werden (siehe Beispiel).

Beispiel 1

```
<MOREDATACONFIGURATION>
     <TOOLTYPE_111>
        <ROW1_COL1>
        </ROW1_COL1>
        <ROW2_COL1>
           <Item value="TM_DGL_CORNER_RADIUS" type="QString" />
        </ROW2_COL1>
        <ROW3_COL1>
           <Item value="GeoCornerRadius" type="QString" />
        </ROW3_COL1>
        <ROW4_COL2>
        </ROW4_COL2>
     </TOOLTYPE_111>
</MOREDATACONFIGURATION>
```

Beispiel 2

```
<MOREDATACONFIGURATION>
     <TOOLTYPE_ALL_500>
        <ROW1_COL1>
           <Item value="TM_DGL_LENGTH" type="QString" />
        </ROW1_COL1>
        <ROW2_COL1>
           <Item value="GeoLengthGeoAx3" type="QString" />
        </ROW2_COL1>
     ...
     </TOOLTYPE_ALL_500>
     <WITHOUT_MORE_DATA>
        <TOOLTYPE_580 />
     </WITHOUT_MORE_DATA>
</MOREDATACONFIGURATION>
```

12.1.7 Fenster "Neues Werkzeug - Favoriten" konfigurieren

Tag <NEWTOOLFAVORITECONFIGURATION>

Im Tag <NEWTOOLFAVORITECONFIGURATION> legen Sie die Werkzeugtypen fest, die am häufigsten verwendet werden. Die festgelegten Werkzeugtypen werden im Fenster "Neues Werkzeug - Favoriten" angezeigt.

Einträge	Bedeutung
StaticTooltypes	Nummern der Werkzeugtypen. Geben Sie die einzelnen Nummern jeweils durch ein Leerzeichen getrennt ein. Die Zuordnung von Werkzeugtyp und Nummer entnehmen Sie dem Kapitel: Liste der Werkzeugtypen (Seite 313)
AdditionalStaticTooltypes	Nummern der Werkzeugtypen, die unterhalb der „StaticTooltypes“ ergänzend angezeigt werden sollen.

Beispiel

```
<NEWTOOLFAVORITECONFIGURATION>
    <StaticTooltypes value="120 140 200 220 240 710 711" type="QString"/>
    <AdditionalStaticTooltypes value="110 111 121 155 156 157 9994 9997"
    type="QString"/>
</NEWTOOLFAVORITECONFIGURATION>
```

12.1.8 Fenster "Neues Werkzeug" konfigurieren

Tag <NEWTOOLCONFIGURATION>

Den Dialog "Neues Werkzeug" verwenden Sie, um ein neues Werkzeug bereits vor dem Anlegen bzw. vor dem Beladen auf einen Magazinplatz mit bestimmten Werkzeugdaten zu versorgen. Dies sind üblicherweise die Werkzeuggröße und/oder der Werkzeugplatztyp, da diese Parameter am beladenen Werkzeug nicht mehr geändert werden können. Es ist sinnvoll zusätzlich den Werkzeugtyp und den Werkzeugnamen in diesem Fenster anzuzeigen.

Im Tag <NEWTOOLCONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neues Werkzeug" angezeigt werden sollen.

Hinweis

Die Anzahl der möglichen Parameter ist durch die Größe des Fensters begrenzt.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Werkzeugparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Werkzeugparameters entnehmen Sie bitte dem Kapitel: Kennungen der Werkzeugparameter (Seite 294)

Beispiel

```
<NEWTOOLCONFIGURATION>
    <!-- Das Standardfeld "Werkzeug übergroß" wird entfernt und durch die Felder -->
    <!-- "Werkzeugplatztyp", "Werkzeuggröße links" und "Werkzeuggröße rechts" -->
    <!-- ersetzt. -->
    <Item2 value="ToolPlaceSpec" type="QString"/>
    <Item3 value="ToolSizeLeft" type="QString"/>
    <Item4 value="ToolSizeRight" type="QString"/>
</NEWTOOLCONFIGURATION>
```

12.1.9 Fenster "Neues Multitool" konfigurieren

Tag <NEWMULTITOOLCONFIGURATION>

Den Dialog "Neues Multitool" verwenden Sie, um ein neues Multitool bereits vor dem Anlegen bzw. vor dem Beladen auf einen Magazinplatz mit bestimmten Multitooldaten zu versorgen. Dies sind üblicherweise die Anzahl der Multitoolplätze und die Art der Abstandsbeziehung der Plätze, da diese Parameter nach dem Anlegen des Multitools nicht mehr geändert werden können. Bei der Multitoolgröße und/oder dem Multitoolplatztyp verhält es sich wie bei der Werkzeuggröße und dem Werkzeugplatztyp. Auch diese Parameter können am beladenen Multitool nicht mehr geändert werden. Es ist sinnvoll zusätzlich den Multitoolnamen in diesem Fenster anzuzeigen.

Im Tag <NEWMULTITOOLCONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neues Multitool" angezeigt werden sollen.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Multitoolparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Multitoolparameters entnehmen Sie bitte dem Kapitel: Kennungen der Multitoolparameter (Seite 305)

12.1.10 Fenster "Neuer Winkelkopf" konfigurieren

Tag <NEWCROSSHEADCONFIGURATION>

Den Dialog "Neuer Winkelkopf" verwenden Sie, um einen neuen Winkelkopf bereits vor dem Anlegen bzw. vor dem Beladen auf einen Magazinplatz mit bestimmten Daten zu versorgen. Dies sind üblicherweise die Winkelkopfgröße und/oder der Winkelkopfplatztyp, da diese Parameter am beladenen Winkelkopf nicht mehr geändert werden können. Es ist sinnvoll zusätzlich den Winkelkopfnamen in diesem Fenster anzuzeigen.

Im Tag <NEWCROSSHEADCONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neuer Winkelkopf" angezeigt werden sollen.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Winkelkopfparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Winkelkopfparameters entnehmen Sie bitte dem Kapitel: Kennungen der Winkelkopfparameter (Seite 307)

12.1.11 Fenster "Neues Werkzeug von Codeträger" konfigurieren

Tag <NEWTOOLIDENTCONNECTIONCONFIGURATION>

Den Dialog "Neues Werkzeug von Codeträger" verwenden Sie, um ein neues Werkzeug vor dem Anlegen zu identifizieren. Daher ist es sinnvoll den Werkzeugtyp und den Werkzeugnamen in diesem Fenster anzuzeigen. Zusätzlich gelten die gleichen Regeln wie bei der Konfiguration des Fensters "Neues Werkzeug" Kapitel "Fenster "Neues Werkzeug" konfigurieren (Seite 320)".

Im Tag <NEWTOOLIDENTCONNECTIONCONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neues Werkzeug von Codeträger" angezeigt werden sollen.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Werkzeugparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Werkzeugparameters entnehmen Sie bitte dem Kapitel: Kennungen der Werkzeugparameter (Seite 294)

12.1.12 Fenster "Neues Werkzeug aus Datei" konfigurieren

Tag <NEWTOOLSAVETOOLDATACONFIGURATION>

Den Dialog "Neues Werkzeug aus Datei" verwenden Sie, um ein neues Werkzeug vor dem Anlegen zu identifizieren. Daher ist es sinnvoll den Werkzeugtyp und den Werkzeugnamen in diesem Fenster anzuzeigen. Zusätzlich gelten die gleichen Regeln wie bei der Konfiguration des Fensters "Neues Werkzeug" Kapitel "Fenster "Neues Werkzeug" konfigurieren (Seite 320)".

Im Tag <NEWTOOLSAVETOOLDATACONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neues Werkzeug aus Datei" angezeigt werden sollen.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Werkzeugparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Werkzeugparameters entnehmen Sie bitte dem Kapitel: Kennungen der Werkzeugparameter (Seite 294)

12.1.13 Fenster "Neues Werkzeug aus TTD-Datei" konfigurieren

Tag <NEWTOOLTTDCREATETOOLCONFIGURATION>

Den Dialog "Neues Werkzeug aus TTD-Datei" verwenden Sie, um ein neues Werkzeug bereits vor dem Anlegen bzw. vor dem Beladen auf einen Magazinplatz mit bestimmten Werkzeugdaten zu versorgen. Dies sind üblicherweise die Werkzeuggröße und/oder der Werkzeugplatztyp, da diese Parameter am beladenen Werkzeug nicht mehr geändert werden können. Es ist sinnvoll zusätzlich den Werkzeugtyp und den Werkzeugnamen in diesem Fenster anzuzeigen.

Im Tag <NEWTOOLTTDCREATETOOLCONFIGURATION> legen Sie die Parameter fest, die im Dialog "Neues Werkzeug aus TTD-Datei" angezeigt werden sollen.

Einträge	Bedeutung
ItemX	Eintrag der Kennung des Werkzeugparameters. Mit "X" wird die Reihenfolge festgelegt, in der die Parameter im Fenster angezeigt werden. Die Kennung des Werkzeugparameters entnehmen Sie bitte dem Kapitel: Kennungen der Werkzeugparameter (Seite 294)

12.1.14 Standardwerte für neue Werkzeuge konfigurieren

Tag <NEWTOOLDEFAULTVALUECONFIGURATION>

Im Tag <NEWTOOLDEFAULTVALUECONFIGURATION> legen sie Standardwerte für Werkzeuge fest, die Sie über die Bedienoberfläche anlegen. Diese Werte können abhängig vom Werkzeugtyp und falls erforderlich auch abhängig von der Schneidenlage konfiguriert werden:

Einträge	Bedeutung
TOOLTYPE_ALL	Die Einstellung des Parameters gilt für alle Werkzeugtypen.
TOOLTYPE_XXX	Die Einstellung des Parameters gilt für einen bestimmten Werkzeugtyp. XXX steht für die Nummer des Werkzeugtyps.
CUTTEDGEPOS_XXX	Die Einstellung des Parameters gilt für eine bestimmte Schneidenlage. XXX steht für die Nummer der Schneidenlage.
Default	Standardwert des Parameters.

Beispiel

```
<NEWTOOLDEFAULTVALUECONFIGURATION>
    <!-- Geometrielänge 1 aller Werkzeugtypen auf 70.0 initialisieren -->
    <TOOLTYPE_ALL>
        <GeoLengthGeoAx1>
            <Default value="70.0" type="double" />
        </GeoLengthGeoAx1>
    </TOOLTYPE_ALL>
    <!-- Spitzenwinkel des Spiralbohrers auf 120.0 initialisieren -->
    <TOOLTYPE_200>
        <NoseAngle>
            <Default value="120.0" type="double" />
        </NoseAngle>
    </TOOLTYPE_200>
    <TOOLTYPE_500>
        <!-- Plattenwinkel des Schruppers auf 82.0 initialisieren -->
            <PlateAngle>
                <Default value="82.0" type="double" />
            </PlateAngle>
        <!-- Halterwinkel des Schruppers abh. von der Schneidenlage initialisieren -->
        <HolderAngle>
                <CUTTEDGEPOS_1>
                    <Default value="93.0" type="double" />
                </CUTTEDGEPOS_1>
                <CUTTEDGEPOS_2>
                    <Default value="93.0" type="double" />
                </CUTTEDGEPOS_2>
                <CUTTEDGEPOS_3>
                    <Default value="93.0" type="double" />
                </CUTTEDGEPOS_3>
                <CUTTEDGEPOS_4>
                    <Default value="93.0" type="double" />
                </CUTTEDGEPOS_4>
                <CUTTEDGEPOS_5>
                    <Default value="49.0" type="double" />
                </CUTTEDGEPOS_5>
                <CUTTEDGEPOS_6>
                    <Default value="49.0" type="double" />
                </CUTTEDGEPOS_6>
                <CUTTEDGEPOS_7>
                    <Default value="49.0" type="double" />
                </CUTTEDGEPOS_7>
                <CUTTEDGEPOS_8>
                    <Default value="49.0" type="double" />
```

```
            </CUTTEDGEPOS_8>
        </HolderAngle>
    </TOOLTYPE_500>
    <!-- Kuehlmittel 2 des Einstechers auf eingeschaltet initialisieren -->
    <TOOLTYPE_520>
        <Coolant2>
            <Default value="true" type="bool" />
        </Coolant2>
    </TOOLTYPE_520>
</NEWTOOLDEFAULTVALUECONFIGURATION>
```

Standardwerte beim Anlegen neuer Werkzeuge

Systemvariable		**Standardwert**
Werkzeuggröße		
Größe nach links	$TC_TP3	1
Größe nach rechts	$TC_TP4	1
Größe nach oben	$TC_TP5	1
Größe nach unten	$TC_TP6	1
Magazinplatztyp des Werkzeugs		
außerhalb des Magazins angelegt	$TC_TP7	1
auf einem Magazinplatz angelegt	$TC_TP7	Platztyp des Magazinplatzes
Werkzeugstatus		
freigegeben	$TC_TP8	2

Abhängig vom Werkzeugtyp ergeben sich folgende Kombinationsmöglichkeiten zwischen Schneidenlage und Schnittrichtung. Der Bediener wählt eine Kombination beim Anlegen des neuen Werkzeugs im Fenster "Neues Werkzeug - Favoriten" aus.

Werkzeugtyp $TC_DP1	**Standardwerte**	
	Schneidenlage $TC_DP2	**Schnittrichtung $TC_DP11**
Typ 500 Schrupper	1	4
Typ 510 Schlichter	2	3
Typ 560 Drehbohrer	3	3
	4	4
	5	1
	6	3
	7	1
	8	3

Werkzeugtyp $TC_DP1	**Standardwerte**	
	Schneidenlage $TC_DP2	**Schnittrichtung $TC_DP11**
Typ 520 Einstecher	1	2
Typ 530 Abstecher	1	4
Schleifwerkzeugtypen	4	4
	3	1
	4	1
	2	3
	3	3
	2	2
Typ 540 Gewindestahl	5	2
	5	1
	8	3
	8	4
	6	3
	6	4
	7	2
	7	1

Wenn die Option ShopMill/ShopTurn gesetzt ist, wird die **Spindeldrehrichtung** vorbelegt:

Werkzeugtyp $TC_DP1	**Standardwert $TC_DP25**	**Spindeldrehrichtung**
Messwerkzeuge		
Typ 580/585/710/712/713/725	Bit 8 = 0 Bit 9 = 0	Spindelstopp
Drehwerkzeuge in Abhängigkeit von SD54215 TM_FUNCTION_MASK_SET Bit 1		
SD54215 Bit 1 = 0	Bit 8 = 1 Bit 9 = 0	Spindel rechts
SD54215 Bit 1 = 1	Bit 8 = 0 Bit 9 = 1	Spindel links
Sonderwerkzeuge		
Typ 730/731/732	Bit 8 = 0 Bit 9 = 0	Spindelstopp
alle anderen Werkzeutypen		
-	Bit 8 = 1 Bit 9 = 0	Spindel rechts

Werkzeugtyp $TC_DP1	**Standardwert $TC_DP25 für Schaltrichtung**
Typ 712 Monotaster	Bit 17 = 1 Bit 20 = 1
Typ 713 L-Taster	Bit 17 = 1 Bit 21 = 1
Typ 714 Sterntaster	Bit 16 = 1 Bit 17 = 1 Bit 18 = 1 Bit 19 = 1

Werkzeugtyp $TC_DP1	**Standardwert $TC_DP24 für Spitzenwinkel**
Typ 200 Spiralbohrer	118
Typ 220 Zentrierer	90

An der Bedienoberfläche erscheint nicht der Freiwinkel sondern der Plattenwinkel:
Plattenwinkel = 180 - Halterwinkel - Freiwinkel

Werkzeugtyp $TC_DP1	**Standardwerte**		
	Schneidenlage $TC_DP2	**Halterwinkel $TC_DP10**	**Freiwinkel $TC_DP24**
Typ 500 Schrupper	1 bis 4	95	5
	5 bis 8	50	50
Typ 510 Schlichter	1 bis 4	93	32
	5 bis 8	62.5	62.5
Typ 560 Drehbohrer	-	90	2

Werkzeugtyp $TC_DP1	**Maßeinheit**	**Standardwert $TC_DP8 für Plattenlänge**
Typ 500 Schrupper	metrisch	11
Typ 510 Schlichter	inch	0.4

Für alle Schleifwerkzeuge gilt für den Parameter zur **Berechnung des Radius**:
$TC_TPG9 = 3 (entspricht Länge 1)

12.1.15 Fenster "Details" konfigurieren

Tag <DETAILSCONFIGURATION>

Im Tag <DETAILSCONFIGURATION> können Sie Einstellungen am Fenster "Details" vornehmen:

Einträge	Bedeutung	
ShowToolNumber	Anzeige der Werkzeugnummer im rechten oberen Feld des Fensters "Details".	
UseAxisNameInLength	Anzeige der Werkzeuglängen mit Achsbezeichnern z. B. Länge X statt Länge 1. Wird keine Angabe gemacht, so wird die Einstellung aus den Werkzeuglisten übernommen.	
ShowBaseLength	Anzeige der Basislänge im Fenster "Details" (AdaptLength1, AdaptLength2 und AdaptLength3). Wird keine Angabe gemacht, so wird die Einstellung aus den Werkzeuglisten übernommen.	
ShowYAxis	Anzeige der Y-Achse im Fenster "Details". Wird keine Angabe gemacht, so wird die Einstellung aus den Werkzeuglisten übernommen.	
AutoDisplayAxisEnabled	`true`	Standardeinstellung Die Auswahl der angezeigten Werkzeuglängen ist an die Anzeige der Werkzeuglängen in der Werkzeugliste angelehnt.
	`false`	Es werden mehrere Komponenten der Werkzeuglänge angezeigt, auch wenn in der Werkzeugliste nur eine Werkzeuglänge angezeigt wird.
OrderOfAxis	Anzeigereihenfolge der Achsen für alle Werkzeugtypen im Fenster "Details" > "Schneidendaten". Die möglichen Werte sind XYZ, XZY, YXZ, YZX, ZXY und ZYX.	

Beispiel 1

```
<DETAILSCONFIGURATION>
    <ShowToolNumber value="true" type="bool" />
</DETAILSCONFIGURATION>
```

Beispiel 2

```
<DETAILSCONFIGURATION>
    <OrderOfAxis value="ZYX" type="QString" />
</DETAILSCONFIGURATION>
```

Für alle Werkzeuge ist die Anzeigereihenfolge der Achsen im Detaildialog ZYX.

Beispiel 3

Wenn der Eintrag `OrderOfAxis` als Tag verwendet wird, kann die Anzeigereihenfolge der Achsen Werkzeugtyp-spezifisch festgelegt werden.

```
<DETAILSCONFIGURATION>
    <OrderOfAxis>
      <TOOLTYPE_130>
        <Item value="ZYX" type="QString" />
      </TOOLTYPE_130>
      <TOOLTYPE_131>
        <Item value="ZYX" type="QString" />
      </TOOLTYPE_131>
    </OrderOfAxis>
</DETAILSCONFIGURATION>
```

Für Winkelköpfe (Typ 130 und 131) ist die Anzeigereihenfolge der Achsen im Detaildialog ZYX.

Beispiel 4

```
<DETAILSCONFIGURATION>
    <OrderOfAxis>
      <TOOLTYPE_130>
        <Item value="ZYX" type="QString" />
      </TOOLTYPE_130>
      <TOOLTYPE_131>
        <Item value="ZYX" type="QString" />
      </TOOLTYPE_131>
      <TOOLTYPE_ALL_500>
        <Item value="ZXY" type="QString" />
      </TOOLTYPE_ALL_500>
      <TOOLTYPE_ALL>
        <Item value="XYZ" type="QString" />
      </TOOLTYPE_ALL>
    </OrderOfAxis>
</DETAILSCONFIGURATION>
```

Für Winkelköpfe (Typ 130 und 131) ist die Anzeigereihenfolge der Achsen im Detailsdialog ZYX.

Für alle Drehwerkzeug (Typ 500 und ff.) ist die Anzeigereihenfolge der Achsen im Detailsdialog ZXY.

Für allen restlichen Werkzeuge ist die Anzeigereihenfolge der Achsen im Detailsdialog XYZ.

12.1.16 Namen für Magazine und Magazinplätze vergeben

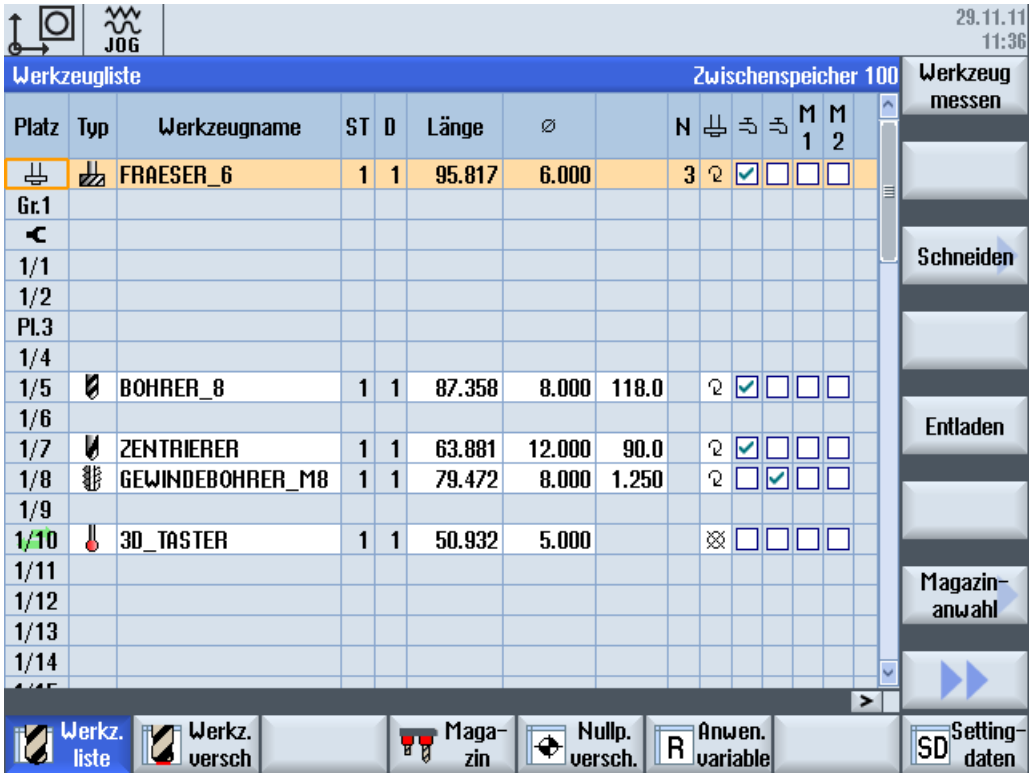
Tag <MAGAZINEPLACENAMECONFIGURATION>

Im Tag <MAGAZINEPLACENAMECONFIGURATION> befinden sich die Einträge zur Vergabe von eigenen Magazinplatznamen. Magazinplätze können mit Texten angezeigt werden. Statt der Kennung, z. B. 1/5 für erstes Magazin, Platz fünf, wird dann in den Werkzeuglisten der projektierte Text ausgegeben.

Einträge	Bedeutung
MAGAZINE_XXX	XXX steht für die Nummer des Magazins.
PLACE_XXX	XXX steht für die Nummer des Platzes.
Name	Textkennung des Magazinplatzes.

Beispiel

```
<MAGAZINEPLACENAMECONFIGURATION>
    <MAGAZINE_9998>
        <PLACE_2>
            <Name value="TM_SPECIAL_MAG_PLACE_1" type="QString" />
        </PLACE_2>
    </MAGAZINE_9998>
    <MAGAZINE_1>
        <PLACE_3>
            <Name value="TM_SPECIAL_MAG_PLACE_2" type="QString" />
        </PLACE_3>
    </MAGAZINE_1>
</MAGAZINEPLACENAMECONFIGURATION>
```



Name für Magazine

Auch für Magazine können eigene Texte vergeben werden. Der Text oder die Textkennung wird in die Magazinkonfigurationsdatei (ini-Datei) der NC eingetragen. Die Systemvariable für den Magazinnamen ist $TC_MAP2[magNo]. Der Name des Magazins wird oben rechts in den Werkzeuglisten angezeigt.

Beispiel

Das Magazin 1 soll den Namen "Hauptmagazin" tragen.

```
$TC_MAP2[1]="Hauptmagazin"
```

Soll der Text "Hauptmagazin" sprachabhängig ausgegeben werden, so müssen sprachabhängige Texte für die Textkennung "Hauptmagazin" angelegt werden.

Siehe auch

Die Vorgehensweise zum Anlegen sprachabhängiger Texte, entnehmen Sie folgendem Kapitel:

OEM-Texte anlegen (Seite 352)

12.1.17 Fenster "Tooltips für Multitoolplätze vergeben" konfigurieren

Tag <MULTITOOLPLACETOOLTIPCONFIGURATION>

Im Tag <MULTITOOLPLACETOOLTIPCONFIGURATION> befinden sich die Einträge zur Vergabe von Tooltips für Multitoolplätze. Dies ist besonders hilfreich wenn an einer Drehmaschine alle verwendeten Multitools zwei Plätze haben und eine feste Zuordnung der Multitoolplätze zur Haupt- und zur Gegenspindel besteht, z. B. Platz 1 - Hauptspindel und Platz 2 - Gegenspindel. Dann könnte der Tooltip für Platz 1 "Werkzeughalter für Hauptspindel" lauten.

Einträge	Bedeutung
PLACE_XXX	XXX steht für die Nummer des Multitoolplatzes.
ToolTip	Textkennung des Magazinplatzes.

Beispiel

```
<MULTITOOLPLACETOOLTIPCONFIGURATION>
        <PLACE_1>
            <ToolTip value="TM_HOLDER_OF_MAIN_SPINDLE" type="QString" />
        </PLACE_1>
        <PLACE_2>
            <ToolTip value="TM_HOLDER_OF_COUNTER_SPINDLE" type="QString" />
        </PLACE_2>
</MULTITOOLPLACETOOLTIPCONFIGURATION>
```

Siehe auch

Die Vorgehensweise zum Anlegen sprachabhängiger Texte, entnehmen Sie folgendem Kapitel:

OEM-Texte anlegen (Seite 352)

12.1.18 Namen für Magazinplatztypen vergeben

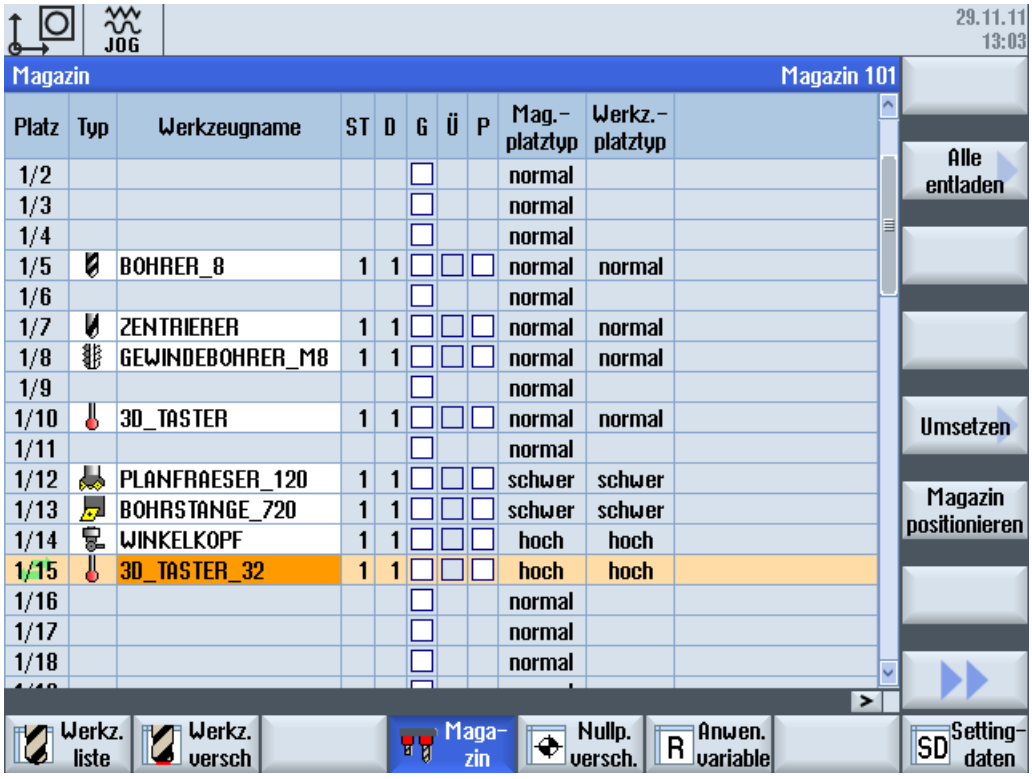
Tag <PLACETYPECONFIGURATION>

Im Tag <PLACETYPECONFIGURATION> befinden sich die Einträge zur Vergabe von eigenen Platztypenbezeichnern. Magazine und Magazinplatztypen können mit Texten in den Werkzeuglisten angezeigt werden. Statt der Kennung 2 für den Platztyp 2 wird dann in den Werkzeuglisten der projektierte Text ausgegeben.

Einträge	Bedeutung
PLACETYPEXXX	XXX steht für die Nummer des Platztyps.
Text	Textkennung des Platztyps.
Tooltip	Tooltip Textkennung des Platztyps.

Beispiel

```
<PLACETYPECONFIGURATION>
     <PLACETYPE1>
        <Text value="TM_PLACETYPE_1" type="QString" />
        <Tooltip value="TM_TT_PLACETYPE_1" type="QString" />
     </PLACETYPE1>
     <PLACETYPE2>
        <Text value="TM_PLACETYPE_2" type="QString" />
        <Tooltip value="TM_TT_PLACETYPE_2" type="QString" />
     </PLACETYPE2>
     <PLACETYPE3>
        <Text value="TM_PLACETYPE_3" type="QString" />
        <Tooltip value="TM_TT_PLACETYPE_3" type="QString" />
     </PLACETYPE3>
</PLACETYPECONFIGURATION>
```



Siehe auch

Die Vorgehensweise zum Anlegen sprachabhängiger Texte, entnehmen Sie folgendem Kapitel:

OEM-Texte anlegen (Seite 352)

12.1.19 Zuordnung von Magazinen und Magazinplätzen zu Kanälen

Voraussetzung

Technologie Drehen oder Technologie Fräsen und Technologieerweiterung Drehen ist eingestellt: Name der zugehörigen Konfigurationsdatei sltmturninglistconfig.xml bzw. sltmmillturnlistconfig.xml.

Zuordnung zwischen Magazin bzw. Magazinplätzen und Kanal

Mit dieser Einstellung können Magazine und Magazinplätze Kanälen zugeordnet werden. Dies ist dann sinnvoll, wenn die Maschine so konfiguriert ist, dass einem Werkzeugbereich (TOA-Bereich) mehrere Kanäle zugeordnet sind. Kommt jedes Magazin dann nur in einem Kanal zum Einsatz, so hat man hier die Möglichkeit, diese Zuordnung zwischen Magazin bzw. Magazinplätzen und Kanal festzuhalten.

Als Folge dieser Konfiguration werden die Werkzeugsymbole in Bezug auf ihre Lage in Abhängigkeit zum Kanal ausgegeben. Jedem Kanal kann über das Kanal-Maschinendatum MD52000 MCS_DISP_COORDINATE_SYSTEM eine Lage zugewiesen werden. Befindet sich ein Werkzeug auf einem Magazinplatz der über diese Konfiguration einem Kanal zugeordnet ist, so wird das Werkzeugsymbol in Abhängigkeit zu dieser eingestellten Lage ausgegeben.

Diese Einstellmöglichkeit ist nur in der Technologie Drehen und in der Technologie Fräsen mit Technologieerweiterung Drehen verfügbar. Sie kann insbesondere dann hilfreich sein, wenn an einer mehrkanaligen Maschine ein Magazin vor und ein anderes Magazin hinter der Drehmitte eingesetzt wird. So können die Werkzeuge in der für den Bediener sichtbaren Lage in die Werkzeugliste eingetragen werden.

Die Zuordnung von Magazinplätzen zu Kanälen ist dann hilfreich, wenn statt zwei Magazinen zwei Werkzeughalter/Werkzeugspindeln existieren, die ihre Werkzeuge aus einem Magazin beziehen. Ein Werkzeughalter wird vor und der andere wird hinter der Drehmitte eingesetzt, so dass sich unterschiedliche Sichten auf das Werkzeug ergeben.

Tag <CHANNELMAGAZINEASSIGNMENT>

Im Tag <CHANNELMAGAZINEASSIGNMENT> befinden sich die Einträge zur Zuordnung von Magazinen bzw. Magazinplätzen zu Kanälen.

Einträge	Bedeutung
CHANNELNO_XXX	XXX steht für die Nummer des Kanals.
Magazines	Nummer der Magazine, die dem Kanal zugeordnet werden.
MAGAZINE_XXX	Nummer des Magazins, dessen Plätze dem Kanal zugeordnet werden sollen.
Places	Nummer der Magazinplätze, die dem Kanal zugeordnet werden sollen.

Beispiel 1

Es soll folgende Zuordnung von Magazin zu Kanal gelten:

Magazin 1-3	Kanal 1
Magazin 4	Kanal 2

Magazin 5	Kanal 3
Magazin 6	Kanal 4

```
<CHANNELMAGAZINEASSIGNMENT>
   <CHANNELNO_1>
      <Magazines value="1 2 3" type="QString" />
   </CHANNELNO_1>
   <CHANNELNO_2>
      <Magazines value="4" type="QString" />
   </CHANNELNO_2>
   <CHANNELNO_3>
      <Magazines value="5" type="QString" />
   </CHANNELNO_3>
   <CHANNELNO_4>
      <Magazines value="6" type="QString" />
   </CHANNELNO_4>
</CHANNELMAGAZINEASSIGNMENT>
```

Beispiel 2

Magazinplatz 1 des Zwischenspeichers soll dem Kanal 1 zugeordnet werden.

Magazinplatz 2 des Zwischenspeichers soll dem Kanal 2 zugeordnet werden.

```
<CHANNELMAGAZINEASSIGNMENT>
   <CHANNELNO_1>
      <MAGAZINE_9998>
         <Places value="1" type="QString" />
      </MAGAZINE_9998>
   </CHANNELNO_1>
   <CHANNELNO_2>
      <MAGAZINE_9998>
         <Places value="2" type="QString" />
      </MAGAZINE_9998>
   </CHANNELNO_2>
</CHANNELMAGAZINEASSIGNMENT>
```

12.1.20 Zuordnung von Magazinen und Magazinplätzen zu Koordinatensystemen

Voraussetzung

Technologie Drehen oder Technologie Fräsen und Technologieerweiterung Drehen ist eingestellt: Name der zugehörigen Konfigurationsdatei sltmturninglistconfig.xml bzw. sltmmillturnlistconfig.xml.

Zuordnung zwischen Magazin bzw. Magazinplätzen und Koordinatensystem

Mit dieser Einstellung können Magazine und Magazinplätze Koordinatensystemen zugeordnet werden. Dies ist dann sinnvoll, wenn die Maschine so aufgebaut ist, dass ein Magazin bzw. ein Magazinplatz nicht zur Lage des im Kanal eingestellten Koordinatensystems passt. Hier hat man die Möglichkeit, diese Zuordnung zwischen Magazin bzw. Magazinplatz und Koordinatensystem festzulegen.

Als Folge dieser Konfiguration werden die Werkzeugsymbole in Bezug auf ihre Lage in Abhängigkeit zum Koordinatensystem ausgegeben.

Diese Einstellmöglichkeit ist nur in der Technologie Drehen und in der Technologie Fräsen mit Technologieerweiterung Drehen verfügbar.

Tag <COORDSYSMAGAZINEASSIGNMENT>

Im Tag <COORDSYSMAGAZINEASSIGNMENT> befinden sich die Einträge zur Zuordnung von Magazinen bzw. Magazinplätzen zu Koordinatensystemen.

Einträge	Bedeutung
COORDSYSNO_XXX	XXX steht für die Nummer des Koordinatensystems.
Magazines	Nummer der Magazine, die dem Kanal zugeordnet werden.
MAGAZINE_XXX	Nummer des Magazins, dessen Plätze dem Koordinatensystem zugeordnet werden sollen.
Places	Nummer der Magazinplätze, die dem Koordinatensystem zugeordnet werden sollen.

Beispiel

Es soll folgende Zuordnung von Magazinen bzw. Magazinplätzen zu Koordinatensystemen gelten:

Magazin 1,3 und 4	Koordinatensystem 34
Magazin 2	Koordinatensystem 33
Magazin 9998 Platz 2	Koordinatensystem 33

```
<COORDSYSMAGAZINEASSIGNMENT>
   <COORDSYSNO_34>
      <Magazines value="1 3 4" type="QString" />
   </COORDSYSNO_34>
```

```
    <COORDSYSNO_33>
        <Magazines value="2" type="QString" />
        <MAGAZINE_9998>
            <Places value="2" type="QString" />
        </MAGAZINE_9998>
    </COORDSYSNO_33>
</COORDSYSMAGAZINEASSIGNMENT>
```

12.1.21 Sperren des Softkeys in Abhängigkeit von Filtern

Tag <FILTERCONFIGURATION> in Kombination mit Tag <SOFTKEYLOCK>

Einige Softkeys können in Abhängigkeit zu Filtereinstellungen ausgeblendet werden. Z. B. wird der Softkey "Neues Werkzeug" ausgeblendet, wenn der Filter "nur gesperrte Werkzeuge" gesetzt ist. Dies ist sinnvoll, weil das neue Werkzeug nicht als gesperrtes Werkzeug angelegt wird und somit direkt aus der Anzeige weggefiltert wird. Das bedeutet, das neue Werkzeug würde nach dem Anlegen nicht angezeigt.

Softkeykennungen des Softkeys, die ausgeblendet werden können:

Einträge	Bedeutung
NewTool	Neues Werkzeug
Edges	Schneiden

Kennungen für Filtereinstellungen, auf die reagiert werden können:

Einträge	Bedeutung
FirstEdge	Filter "nur erste Schneide", Standardeinstellung "Edges"
StandBy	Filter "nur einsatzbereite Werkzeuge", Standardeinstellung "NewTool"
Active	Filter "nur Werkzeuge mit Aktivkennung", Standardeinstellung "NewTool"
Locked	Filter "nur gesperrte Werkzeuge", Standardeinstellung "NewTool"
Prewarn	Filter "nur Werkzeuge mit Vorwarngrenze erreicht", Standardeinstellung "NewTool"
RemainingPieces	Filter "nur Werkzeuge mit Reststückzahl", Standardeinstellung "NewTool Edges"
RemainingTime	Filter "nur Werkzeuge mit Reststandzeit", Standardeinstellung "NewTool Edges"
MarkedToUnload	Filter "nur Werkzeug mit Entladekennung", Standardeinstellung "NewTool Edges"
MarkedToLoad	Filter "nur Werkzeuge mit Beladekennung", Standardeinstellung "NewTool Edges"

Beispiel

Die Softkeys „Neues Werkzeug“ und „Schneiden“ sollen ausgeblendet werden, wenn der Filter „nur gesperrte Werkzeuge“ eingeschaltet ist.

```
<FILTERCONFIGURATION>
   <SOFTKEYLOCK>
      <Locked value="NewTool Edges" type="QString" />
   </SOFTKEYLOCK>
</FILTERCONFIGURATION>
```

12.1.22 Fenster "Auswahl der Beladestelle" konfigurieren

Tag <LOADSTATIONSHANDLEALLTOOLS>

Für die Funktionen "Alle löschen", "Alle entladen" und "Alle beladen" muss eine Beladestelle angegeben werden, falls mehrere Beladestellen in der Magazinkonfiguration der NC konfiguriert sind. Die Angabe der Beladestelle erfolgt über einen Dialog "Auswahl der Beladestelle" oder über die Konfiguration von Beladestellen mit dem Tag <LOADSTATIONSHANDLEALLTOOLS>.

Einträge	Bedeutung
DialogEnabled	Fenster "Auswahl der Beladestelle" anbieten, Standardeinstellung "true"
MAGAZINE_XXX	Nummer des Magazins, für das eine Beladestelle zugeordnet werden soll.
LoadStation	Beladestelle, die dem Magazin zugeordnet werden soll.

Beispiel

```
<LOADSTATIONSHANDLEALLTOOLS>
   <DialogEnabled value="true" type="bool" />
   <MAGAZINE_1>
      <LoadStation value="2" type="uint" />
   </MAGAZINE_1>
   <MAGAZINE_2>
      <LoadStation value="1" type="uint" />
   </MAGAZINE_2>
</LOADSTATIONSHANDLEALLTOOLS>
```

12.1.23 Kühlmittel und werkzeugspezifische Funktionen

Kühlmittel (ShopMill / ShopTurn) zuordnen

In der Werkzeugverwaltung von ShopMill / ShopTurn, können Sie jedem Werkzeug Kühlmittel und werkzeugspezifische Funktionen zuordnen. Diese werden beim Einwechseln des Werkzeugs aktiviert. Die Zuordnung der Kühlmittel zu den entsprechenden Maschinenfunktionen (M-Funktionen) nehmen Sie über folgende Maschinendaten vor:

MD52230 $MCS_M_CODE_ALL_COOLANTS_OFF	M-Code für alle Kühlmittel AUS
MD52231 $MCS_M_CODE_COOLANT_1_ON	M-Code für Kühlmittel 1 EIN
MD52232 $MCS_M_CODE_COOLANT_2_ON	M-Code für Kühlmittel 2 EIN
MD52233 $MCS_M_CODE_COOLANT_1_AND_2_ON	M-Code für beide Kühlmittel EIN

Werkzeugspezifische Funktionen (ShopMill)

Die werkzeugspezifischen Funktionen 1 ... 4 sind für weitere M-Funktionen gedacht, die zu einem Werkzeug aktiviert werden können. Z. B. drittes Kühlmittel, Überwachungen von Drehzahl, Werkzeugbruch usw.

Für das Ein- oder Ausschalten der werkzeugspezifischen Funktionen können maximal 8 M-Funktionen programmiert werden. Die Definition geschieht über folgende Maschinendaten:

MD52281 $MCS_TOOL_MCODE_FUNC_ON[0...3]	M-Code für werkzeugspezifische Funktion EIN
= - 1	

MD52282 $MCS_TOOL_MCODE_FUNC_OFF[0...3]	M-Code für werkzeugspezifische Funktion AUS
= - 1	

Die M-Funktionen für die werkzeugspezifischen Funktionen werden von den ShopMill-Zyklen beim Werkzeugwechsel ausgeführt. Ist der Haken im Feld gesetzt, wird der entsprechende M-Code aus MD52281 generiert. Ist der Haken nicht gesetzt, so wird der entsprechende M-Code aus MD52282 generiert.

Wenn Sie keine zusätzlichen M-Befehle automatisch beim Werkzeugwechsel ausführen möchten, dann setzen Sie für die M-Befehle dieser werkzeugspezifischen Funktion die zugehörigen Maschinendaten auf -1. Die dazugehörenden Felder in der Werkzeugliste werden damit ebenfalls ausgeblendet.

Die Maschinendaten sind mit -1 vorbelegt, so dass für die entsprechenden Funktionen keine M-Werte ausgegeben werden.

Technischer Ablauf

Die M-Funktionen für die werkzeugspezifischen Funktionen werden von den ShopMill-Zyklen in einem Satz generiert (maximal 4 M–Befehle sind möglich).

Die Anzahl und auch die Reihenfolge der M-Befehle hängen von den Einstellungen der Maschinendaten MD52281 bis MD52282 und von der Programmierung ab.

Wenn M–Funktionen kleiner 100 verwendet werden, dann hat die Anzahl und die Reihenfolge der M-Befehle keine Bedeutung. Sie stehen in der Anwender-PLC im dekodierten Bereich (DB21.DBB194 bis DB21.DBB206) zur Verfügung, unabhängig von der Anzahl und der Reihenfolge bei der Programmierung.

Wenn jedoch M–Funktionen ab 100 verwendet werden, dann decodieren Sie diese selber in der Anwender-PLC. Berücksichtigen Sie hierbei die Anzahl und die Reihenfolge (M–Funktion 1 bis M–Funktion 4).

Betriebsart JOG

In der Betriebsart JOG werden keine Kühlmittel und werkzeugspezifischen Funktionen an die Nahtstelle ausgegeben. Bei einem Werkzeugwechsel in der Betriebsart JOG können diese Funktionen vom Bediener über Tasten auf der Maschinensteuertafel ausgelöst werden (Realisierung über PLC-Anwenderprogramm).

Beispiel: Einstellungen von Maschinendaten

MD52281 $MCS_TOOL_MCODE_FUNC_ON[0]	= 90
MD52281 $MCS_TOOL_MCODE_FUNC_ON[1]	= 92
MD52281 $MCS_TOOL_MCODE_FUNC_ON[2]	= 94
MD52281 $MCS_TOOL_MCODE_FUNC_ON[3]	= 96

MD52282 $MCS_TOOL_MCODE_FUNC_OFF[0]	= 91
MD52282 $MCS_TOOL_MCODE_FUNC_OFF[1]	= - 1
MD52282 $MCS_TOOL_MCODE_FUNC_OFF[2]	= 95
MD52282 $MCS_TOOL_MCODE_FUNC_OFF[3]	= 97

Daraus folgt, dass in der Werkzeugliste die Felder 1 bis 4 der werkzeugspezifischen Funktionen angezeigt werden.

Folgende Felder werden aktiviert:

Werkzeugspezifische Funktion 1: ein
Werkzeugspezifische Funktion 2: aus
Werkzeugspezifische Funktion 3: aus
Werkzeugspezifische Funktion 4: aus

Von den ShopMill-Zyklen werden beim Werkzeugwechsel die M-Funktionen M90, M95 und M97 generiert.

Die Texte für "Kühlmittel 1/2 an/aus" und "WZ spez. Fkt. 1, ..." können Sie ändern (siehe Kapitel Kennungen der Standardtexte (Seite 353)). Damit die Texte zwischen Werkzeugliste und Bereich Programm Maschinenfunktionen zusammen passen, beachten Sie bitte das folgende Kapitel.

Anwenderspezifische Texte für das Fenster "Maschinenfunktionen" anlegen

Wenn Sie eine zusätzliche Möglichkeit benötigen Kühlmittel und werkzeugspezifische Funktionen ein- oder auszuschalten, dann können Sie dies im Bedienbereich "Programm" mit Hilfe des Eingabefensters "Maschinenfunktionen" programmieren. Die Texte und Tooltips für die werkzeugspezifischen Funktionen, die in diesem Fenster verwendet werden, können Sie anlegen.

In der Datei "slstepforms_xxx.ts" definieren Sie die Texte für die M-Funktionen.

Wenn Sie die Texte in weiteren Sprachen benötigen, müssen Sie für jede Sprache eine eigene Datei anlegen. Die Dateien unterscheiden sich im Dateinamen lediglich durch die Sprachkennung "xxx". Verwenden Sie nur die vorgegebenen Sprachkennungen.

Weitere Informationen
Inbetriebnahmehandbuch SINUMERIK Operate; Unterstützte Sprachen

Datei "slstepforms_xxx.ts"

Tag	Bedeutung
source	Textkennung für Label: SlStepLabels Es können maximal bis zu 4 M-Funktionen definiert werden: T_LAB_USER_MACHINE_FUNC_1 T_LAB_USER_MACHINE_FUNC_2 T_LAB_USER_MACHINE_FUNC_3 T_LAB_USER_MACHINE_FUNC_4
	Textkennung für Tooltip: SlStepToolTip Für jede Maschinenfunktion x (1 - 4) gibt es 3 Tooltips: T_TT_USER_MACHINE_FUNC_x: standardmäßiger Tooltip, falls die Maschinenfunktion nicht ausgewählt wurde. T_TT_USER_MACHINE_FUNC_x _ON: Tooltip bei aktivierter Maschinenfunktion T_TT_USER_MACHINE_FUNC_x_OFF: Tooltip bei deaktivierter Maschinenfunktion
translation	Text, der auf der Bedienoberfläche angezeigt wird. Labeltexte: Sie können Sie nur einzeiligen Text eingeben. Der Eintrag "lines" entfällt. Tooltip: Sie können mehrzeiligen Text eingeben. Ein Zeilenumbruch wird durch die Zeichen "%n" gesetzt.
chars	Textlänge Die maximale Textlänge ist auf 30 Zeichen festgelegt.
lines	Zeilenanzahl Die maximale Zeilenanzahl beträgt 3.
remark	Eigener Kommentar - Eintrag wird auf der Bedienoberfläche nicht angezeigt.

Vorgehensweise

1. Sie können sich die Musterdatei "oem_slstepforms_deu.ts" aus folgendem Verzeichnis kopieren: /**siemens**/sinumerik/hmi/template/lng.
2. Legen Sie die Datei im Verzeichnis /**oem**/sinumerik/hmi/lng bzw. /**user**/sinumerik/hmi/lng ab bzw. an.

3. Ändern Sie den Dateinamen in "slstepforms_deu.ts".
   Wenn Sie die Texte für weitere Sprachen benötigen, muss für jede Sprache eine eigene Datei angelegt werden. Speichern Sie die Datei mit der entsprechenden Sprachkennung im Dateinamen ab.
4. Öffnen Sie die Datei und legen Sie jeweils im Bereich <message> und </message> die Texte fest.
5. Starten Sie das System neu.
   Damit die Texte zur Programmlaufzeit angezeigt werden, muss die Datei in ein Binärformat konvertiert werden. Diese Konvertierung erfolgt nur im Hochlauf.

Beispiel einer Datei "slstepforms_deu.ts"

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS><TS>
<context>
   <name>SlStepLabels</name>
   <message>
      <source>T_LAB_USER_MACHINE_FUNC_1</source>
      <translation>WZ spez. Funktion 1</translation>
      <chars>20</chars>
   </message>
<context>
   <name>SlStepToolTip</name>
   <!-- tooltips of user machine function 1-->
   <message>
      <source>T_TT_USER_MACHINE_FUNC_1</source>
      <translation>Anwender%nMaschinenfunktionen 1</translation>
      <chars>25</chars>
      <lines>3</lines>
   </message>
</context
</TS>
```

12.1.24 Werkzeugwechselgrund beim Reaktivieren

Funktion

Sie haben die Möglichkeit beim Reaktivieren eines Werkzeugs den Werkzeugwechselgrund zu erfassen, um ihn z. B. für eine Statistik zu verwenden. Nach Drücken des Softkeys "Reaktivieren", wird das Auswahlfenster "Werkzeugwechselgrund" geöffnet. In einem Kontrollkästchen können Sie den Werkzeugwechselgrund anwählen. Der Werkzeugwechsel wird in einen zu konfigurierenden Werkzeugparameter oder in eine lokale Variable des SlCap-Services eingetragen.

Werkzeugwechselgründe

Die Werkzeugwechselgründe können Sie beliebig anpassen oder erweitern. Für die Projektierung stellt die Bedien-Software folgende vordefinierte Texte zur Verfügung:

Textkennung	Text
TM_DGL_DESIRED_PIECES_ELAPSED	Soll-Stückzahl abgelaufen
TM_DGL_TOOL_BREAKAGE	Werkzeugbruch
TM_DGL_CUTTING_EDGE_DAMAGE	Schneidenbruch
TM_DGL_PREMATURE_WEAR	Vorzeitiger Verschleiß
TM_DGL_TYPE_REEQUIPPING	Typumrüstung
TM_DGL_TOOL_TRIAL	Werkzeugversuch
TM_DGL_PREVENTIVE_CHANGE	Vorbeugender Wechsel

Tag <TOOLCHANGEREASON>

Im Tag <TOOLCHANGEREASON> konfigurieren Sie die Funktion:

Parameter	Bezeichnung
Enabled	true - Auswahlfenster "Werkzeugwechselgrund" wird beim Reaktivieren eingeblendet.
	false - Standardwert, Auswahlfenster wird nicht angezeigt.
OnlyOneReasonAllowed	Es darf nur ein Werkzeugwechselgrund ausgewählt werden.
UseCapLocalParameter	Der Werkzeugwechselgrund wird in eine lokale Variable des SlCap-Services eingetragen.
REASONXXX	XXX steht für eine interne Nummer. Die Nummern für den Werkzeugwechselgrund müssen eindeutig sein.
Text	Textkennung des Werkzeugwechselgrunds.
Parameter	Parameterkennung des Parameters, in dem eingetragen wird, ob der Werkzeugwechselgrund angewählt wurde.
BitMask	Bitmaske für den Eintrag des Wechselgrunds in die lokale Variable des SlCap-Services. Die Bitmaske wird als Integerwert eingetragen. Bit 0 → 1, Bit 1 → 2, Bit 2 → 4

Lokale Variable des SlCap-Services

Die Kennung der lokalen Variablen lautet "/Hmi/ToolEdgeReplacementReason". Die Variable ist vom Typ String.

In der folgenden Tabelle werden die Argumente und deren Bedeutung aufgelistet:

Argument	Bedeutung
datetime	Datum und Uhrzeit des Ereignisses
containerNo	Nummer des Verwaltungsbereichs
toaNo	Nummer des To-Bereichs
toolNo	Werkzeugnummer
edgeNo	Schneidennummer
toolIdent	Werkzeugname

Argument	Bedeutung
toolDuplo	Schwesterwerkzeugnummer
ncuName	Name der NCU
reactivationReason	Werkzeugwechselgrund
monitoringMode	Überwachungsart
remainingBefore	Reststückzahl/-standzeit vor dem Reaktivieren

Beispiel 1

Folgendes Beispiel verwendet die in der Bedien-Software bereits vorbereiteten Texte für den Werkzeugwechselgrund. Die Ablage des Werkzeugwechselgrunds erfolgt in diesem Beispiel in noch zu konfigurierenden Parametern, z. B. das OEM-Werkzeugdatum 1, Bit 1 und folgende.

```
<TOOLCHANGEREASON>
   <Enabled value="true" type="bool" />
     <REASON1>
       <Text value="TM_DGL_DESIRED_PIECES_ELAPSED" type="QString" />
       <Parameter value="ToolUser_1_Bit1" type="QString" />
     </REASON1>
     <REASON2>
       <Text value="TM_DGL_TOOL_BREAKAGE" type="QString" />
       <Parameter value="ToolUser_1_Bit2" type="QString" />
     </REASON2>
     <REASON3>
       <Text value="TM_DGL_CUTTING_EDGE_DAMAGE" type="QString" />
       <Parameter value="ToolUser_1_Bit3" type="QString" />
     </REASON3>
     <REASON4>
       <Text value="TM_DGL_PREMATURE_WEAR" type="QString" />
       <Parameter value="ToolUser_1_Bit4" type="QString" />
     </REASON4>
     <REASON5>
       <Text value="TM_DGL_TYPE_REEQUIPPING" type="QString" />
       <Parameter value="ToolUser_1_Bit5" type="QString" />
     </REASON5>
     <REASON6>
       <Text value="TM_DGL_TOOL_TRIAL" type="QString" />
       <Parameter value="ToolUser_1_Bit6" type="QString" />
     </REASON6>
     <REASON7>
       <Text value="TM_DGL_PREVENTIVE_CHANGE" type="QString" />
       <Parameter value="ToolUser_1_Bit7" type="QString" />
     </REASON7>
</TOOLCHANGEREASON>
```

Beispiel 2

Auch in diesem Beispiel werden die vorbereiteten Texte für den Wechselgrund verwendet. Es wird nur ein Wechselgrund als gültige Eingabe zugelassen. Die Ablage des Werkzeugwechselgrunds erfolgt in die lokale Variable des SlCap-Services.

```
<TOOLCHANGEREASON>
   <Enabled value="true" type="bool" />
     <OnlyOneReasonAllowed value="true" type="bool" />
     <UseCapLocalParameter value="true" type="bool" />
     <REASON1>
       <Text value="TM_DGL_DESIRED_PIECES_ELAPSED" type="QString" />
       <BitMask value="1" type="int" />
     </REASON1>
     <REASON2>
       <Text value="TM_DGL_TOOL_BREAKAGE" type="QString" />
       <BitMask value="2" type="int" />
     </REASON2>
     <REASON3>
       <Text value="TM_DGL_CUTTING_EDGE_DAMAGE" type="QString" />
       <BitMask value="4" type="int" />
     </REASON3>
     <REASON4>
       <Text value="TM_DGL_PREMATURE_WEAR" type="QString" />
       <BitMask value="8" type="int" />
     </REASON4>
     <REASON5>
       <Text value="TM_DGL_TYPE_REEQUIPPING" type="QString" />
       <BitMask value="16" type="int" />
     </REASON5>
     <REASON6>
       <Text value="TM_DGL_TOOL_TRIAL" type="QString" />
       <BitMask value="32" type="int" />
     </REASON6>
     <REASON7>
       <Text value="TM_DGL_PREVENTIVE_CHANGE" type="QString" />
       <BitMask value="64" type="int" />
     </REASON7>
</TOOLCHANGEREASON>
```

Siehe auch

Für die Konfiguration von neuen Parametern siehe Listenparameter konfigurieren (Seite 309).

12.1.25 Codeträgeranbindung konfigurieren

Voraussetzung



Software-Option
Für diese Funktion benötigen Sie die Option "Tool Ident Connection".

Tag <IDENTCONNECTIONCONFIGURATION>

Im Tag <IDENTCONNECTIONCONFIGURATION> befinden sich die Einträge zur Konfiguration der Codeträgeranbindung in die Werkzeugliste der Bedienoberfläche. Die Codeträgeranbindung erfolgt über TDI Ident Connection.

Dazu stehen folgende Funktionen in der Werkzeugliste zur Verfügung:

1. Anlegen von Werkzeugen vom Codeträger:

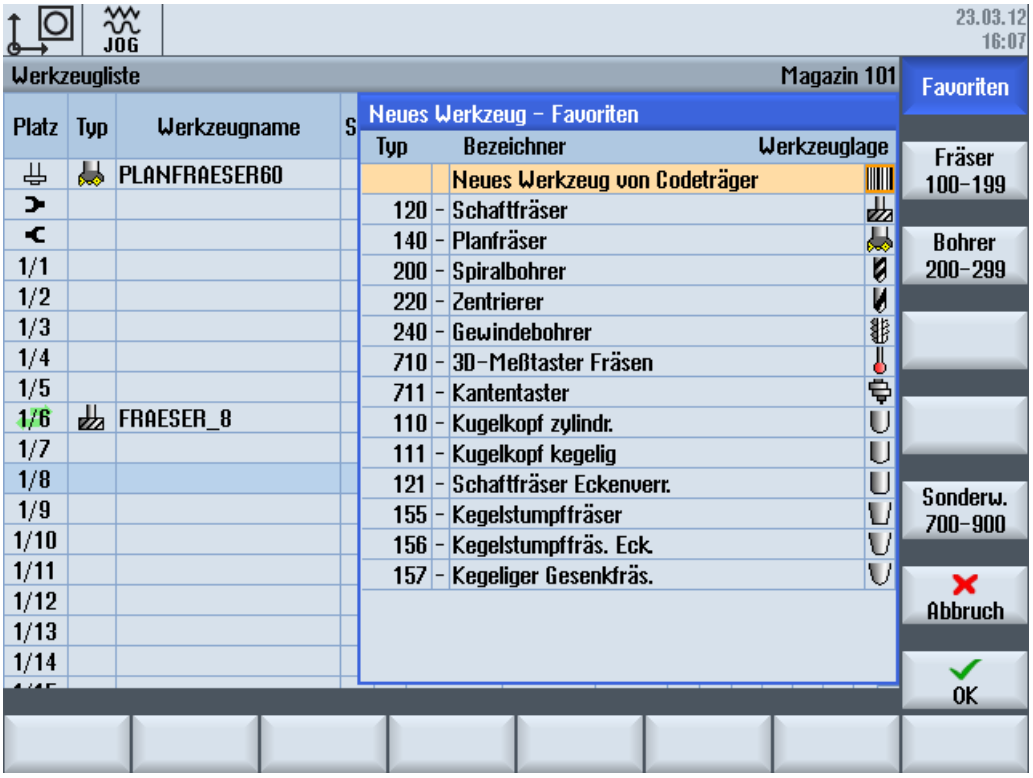


Bild 12-1 Neues Werkzeug von Codeträger

   Es werden die Daten des Werkzeugs vom Codeträger gelesen und im Fenster "Neues Werkzeug" angezeigt. Mit OK wird das Werkzeug dann in der NC angelegt und mit den Codeträgerdaten initialisiert.

2. Entladen von Werkzeugen auf Codeträger:
   Das Werkzeug wird entladen und die Daten des Werkzeugs werden auf den Codeträger geschrieben. Über eine Einstellung (siehe unten) kann das Werkzeug auch gelöscht werden.

3. Löschen des Werkzeugs auf Codeträger:
   Das Werkzeug wird entladen und die Daten des Werkzeugs werden auf den Codeträger geschrieben. Danach wird das Werkzeug aus der NC gelöscht. Über eine Einstellung (siehe unten) kann der Softkey "Auf Codeträger" beim Werkzeug löschen auch ausgeblendet werden.

Folgende Einstellungen können Sie vornehmen:

Einträge	Bedeutung
Enabled	true - Codeträgeranbindung wird angeboten. false - Codeträgeranbindung wird nicht angeboten.
ShowInternalMessage	true - Es werden zusätzlich interne Meldungen zur Fehleranalyse angezeigt. false - Standardeinstellung, es werden keine internen Fehlermeldungen angezeigt.

Einträge	Bedeutung
UnloadWithoutIdentconnection	true - Beim Entladen wird die Funktion "Auf Codeträger" nicht angeboten. Sie wird nur beim "Werkzeug löschen" angezeigt. false - Standardeinstellung, beim Entladen wird die Funktion "Auf Codeträger" angeboten.
UnloadWithDelete	true - Beim Entladen auf Codeträger wird das Werkzeug gelöscht. Die Funktion "Auf Codeträger" wird beim "Werkzeug löschen" nicht angeboten. false - Standardeinstellung, beim Entladen auf Codeträger wird das Werkzeug nur entladen und bleibt im NC-Speicher erhalten.
OnlyInLoadStation	true - Standardeinstellung, einerseits können nur beladene Werkzeuge "auf Codeträger" gespeichert werden, andererseits können neue Werkzeuge "von Codeträger" nur direkt ins Magazin angelegt werden. false - Werkzeuge können unabhängig vom Magazinplatz "von Codeträger" angelegt bzw. "auf Codeträger" gelöscht werden.
CreateNewToolWithDialog	true - Standardeinstellung, vor dem Anlegen des Werkzeugs von Codeträger werden die Werkzeugdaten im Fenster "Neues Werkzeug" angezeigt. false - Das Werkzeug wird direkt in der Werkzeugliste angelegt.
ForceSisterToolNumber	true - Die von den Codeträgerdaten vorgegebene Schwesterwerkzeug-Nummer wird beim Anlegen des Werkzeugs beibehalten. false - Standardeinstellung, die Schwesterwerkzeug-Nummer wird frei vergeben.
UseDefaultData	true - Das neue Werkzeug wird mit den konfigurierten Defaultdaten initialisiert bevor die Codeträgerdaten übernommen werden (siehe Tag <NEWTOOLDEFAULTVALUECONFIGURATION> Kapitel "Standardwerte für neue Werkzeuge konfigurieren (Seite 323)"). false - Standardeinstellung, das neue Werkzeug wird nicht mit den konfigurierten Defaultdaten initialisiert.
OrderFirstWriteThanUnload	true - Beim Entladen/Löschen werden zuerst die Daten auf den Codeträger geschrieben und anschließend wird das Werkzeug entladen bzw. gelöscht. false - Standardeinstellung, beim Entladen/Löschen wird zuerst das Werkzeug entladen, anschließend werden die Daten auf den Codeträger geschrieben und falls angefordert wird das Werkzeug danach gelöscht.

Beispiel

```
<IDENTCONNECTIONCONFIGURATION>
      <Enabled value="true" type="bool" />
      <ShowInternalMessage value="false" type="bool" />
      <UnloadWithDelete value="false" type="bool" />
      <OnlyInLoadStation value="true" type="bool" />
      <CreateNewToolWithDialog value="true" type="bool" />
</IDENTCONNECTIONCONFIGURATION>
```

Weitere Informationen

Weitere Informationen zur Werkzeugverwaltung mit Codeträger finden Sie unter Anbindung Codeträger - Tool Ident Connection (Option) (Seite 369).

12.1.26 Speichern in Datei konfigurieren

Tag <SAVETOOLDATACONFIGURATION>

Im Tag <SAVETOOLDATACONFIGURATION> befinden sich die Einträge zur Konfiguration der Funktion "Werkzeuge in Datei speichern" bzw. "Werkzeuge von Datei anlegen".

Dazu stehen folgende Funktionen in der Werkzeugliste zur Verfügung:

1. Anlegen von Werkzeugen aus Datei:

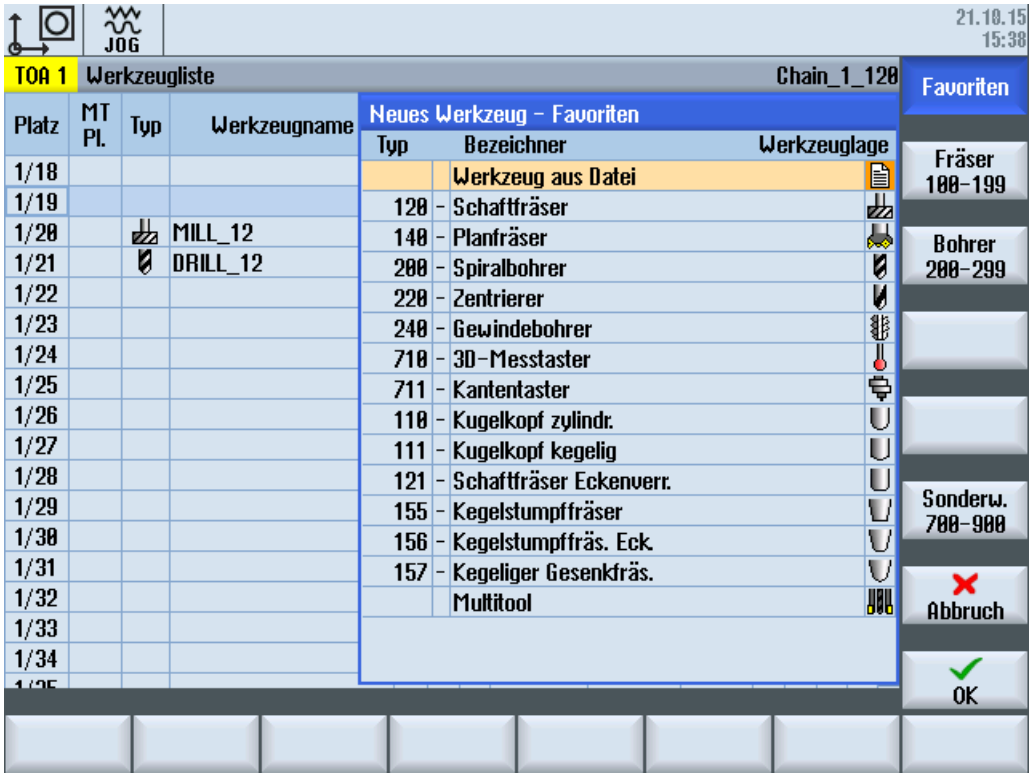


Bild 12-2 Werkzeug aus Datei

Es werden die Daten des Werkzeugs aus einer Datei gelesen und im Fenster "Neues Werkzeug" angezeigt. Mit OK wird das Werkzeug dann in der NC angelegt und mit den Daten aus der Datei initialisiert.

2. Entladen von Werkzeugen in eine Datei:
   Das Werkzeug wird entladen und die Daten des Werkzeugs werden in eine Datei geschrieben. Über eine Einstellung (siehe unten) kann das Werkzeug auch gelöscht werden.
3. Löschen des Werkzeugs in eine Datei:
   Das Werkzeug wird entladen und die Daten des Werkzeugs werden in eine Datei geschrieben. Danach wird das Werkzeug aus der NC gelöscht. Über eine Einstellung (siehe unten) kann der Softkey "In Datei" beim Werkzeug löschen auch ausgeblendet werden.

Folgende Einstellungen können Sie vornehmen:

Einträge	Bedeutung
Enabled	true - Werkzeug von/in Datei wird angeboten. false - Werkzeug von/in Datei wird nicht angeboten.
UnloadWithDelete	true - Beim Entladen in Datei wird das Werkzeug gelöscht. Die Funktion "In Datei" wird beim "Werkzeug löschen" nicht angeboten. false - Standardeinstellung, beim Entladen in Datei wird das Werkzeug nur entladen und bleibt im NC-Speicher erhalten.

Einträge	Bedeutung
CreateNewToolWithDialog	true - Standardeinstellung, vor dem Anlegen des Werkzeugs von Datei werden die Werkzeugdaten im Fenster "Neues Werkzeug" angezeigt. false - Das Werkzeug wird direkt in der Werkzeugliste angelegt.
ForceSisterToolNumber	true - Die von der Datei vorgegebene Schwesterwerkzeug-Nummer wird beim Anlegen des Werkzeugs beibehalten. false - Standardeinstellung, die Schwesterwerkzeug-Nummer wird frei vergeben.
UseDefaultData	true - Das neue Werkzeug wird mit den konfigurierten Defaultdaten initialisiert bevor die Daten von der Datei übernommen werden (siehe Tag <NEWTOOLDEFAULTVALUECONFIGURATION> Kapitel "Standardwerte für neue Werkzeuge konfigurieren (Seite 323)"). false - Standardeinstellung, das neue Werkzeug wird nicht mit den konfigurierten Defaultdaten initialisiert.

12.1.27 Anlegen aus TTD-Datei konfigurieren

Tag <TTDCREATETOOLCONFIGURATION>

Im Tag <TTDCREATETOOLCONFIGURATION> befinden sich die Einträge zur Konfiguration der Funktion „Werkzeuge von TTD-Datei anlegen“.

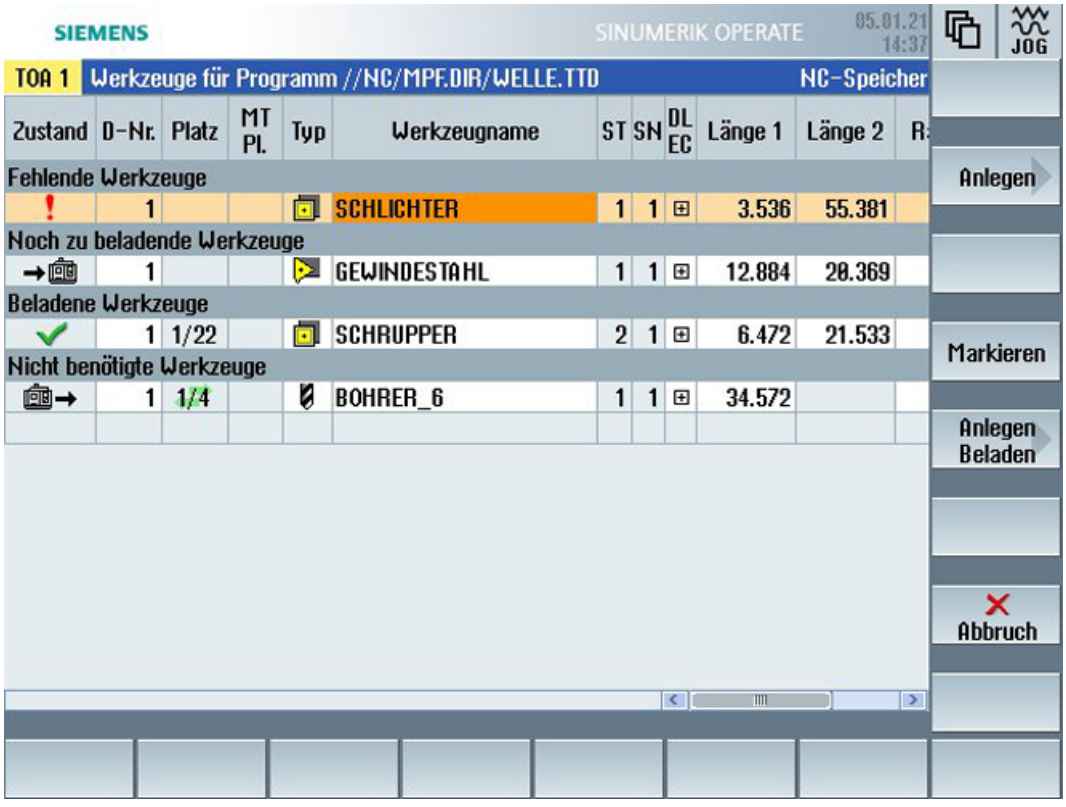


Bild 12-3 Werkzeug aus TTD-Datei

Folgende Einstellungen können Sie vornehmen:

Einträge	Bedeutung
CreateNewToolWithDialog	true - Vor dem Anlegen des Werkzeugs von TTD-Datei werden die Werkzeugdaten im Fenster "Neues Werkzeug aus TTD-Datei" angezeigt false - Standardeinstellung, das Werkzeug wird direkt in der Werkzeugliste angelegt.
ForceSisterToolNumber	true - Die von der Datei vorgegebene Schwesterwerkzeug-Nummer wird beim Anlegen des Werkzeugs beibehalten. false - Standardeinstellung, die Schwesterwerkzeug-Nummer wird frei vergeben.
UseDefaultData	true - Das neue Werkzeug wird mit den konfigurierten Defaultdaten initialisiert, bevor die Daten von der Datei übernommen werden (siehe Tag <NEWTOOLDEFAULTVALUECONFIGURATION> "Standardwerte für neue Werkzeuge konfigurieren (Seite 323)") false - Standardeinstellung, das neue Werkzeug wird nicht mit den konfigurierten Defaultdaten initialisiert.

12.2 OEM-Texte anlegen

Übersicht

Dieses Kapitel beschreibt das Vorgehen, wie Sie OEM-Texte in der jeweiligen Landessprache anlegen.

Die Texte werden jeweils in der Datei "sltmlistdialog_xxx.ts" angelegt.

Für jede Sprache muss eine eigene Datei angelegt werden. Die Dateien unterscheiden sich im Dateinamen durch die Sprachkennung. Setzen Sie für "xxx" die entsprechende Sprachkennung ein.

Weitere Informationen
Inbetriebnahmehandbuch SINUMERIK Operate; Unterstützte Sprachen

Texte anlegen

Alle OEM-Texte aus den vorangegangenen Kapiteln wie z. B. der Tooltip können sprachabhängig gemacht werden, indem der OEM-Text als Textkennung unter dem Tag "source" verwendet wird. Die Übersetzung des Textes in die Landessprache wird unter dem Tag "translation" eingetragen."

Einträge	Bedeutung
source	Textkennung Siehe 1. Beispiel: einzeiliger Text mit neuer Textkennung "MY_NEW_TEXT". Siehe 2. Beispiel: zweizeiliger Text mit neuer Textkennung "MY_NEW_TEXT_2_LINES".
translation	Text, der auf der Bedienoberfläche angezeigt wird. Sie können einzeiligen und mehrzeiligen Text eingeben. Ein Zeilenumbruch wird durch "%n" gesetzt. Siehe 1. Beispiel: Einzeiliger Text "Mein neuer Text" wird angezeigt. Siehe 2. Beispiel: Zweizeiliger Text "Mein neuer - Text" wird angezeigt.

Einträge	Bedeutung
chars	Textlänge Siehe 1. Beispiel: Textlänge ist auf 30 Zeichen festgelegt. Siehe 2. Beispiel: Textlänge ist auf 10 Zeichen pro Zeile festgelegt.
lines	Zeilenanzahl Siehe 2. Beispiel: Die Anzahl der Zeilen beträgt 2.
remark	Eigener Kommentar - wird nicht angezeigt.

Vorgehensweise

1. Sie können sich die Musterdatei "oem_sltmlistdialog_deu.ts" aus folgendem Verzeichnis kopieren: /**siemens**/sinumerik/hmi/template/lng.
2. Legen Sie die Datei im Verzeichnis /**oem**/sinumerik/hmi/lng oder /**user**/sinumerik/hmi/lng ab.
3. Ändern Sie den Dateinamen in "sltmlistdialog_deu.ts".
   Wenn Sie die Texte für weitere Sprachen anlegen möchten, muss für jede Sprache eine eigene Datei angelegt werden. Speichern Sie die Datei mit der entsprechenden Sprachkennung im Dateinamen ab.
4. Öffnen Sie die Datei und legen Sie im Bereich <message> und </message> die Texte fest.
5. Starten Sie das System neu.
   Damit die Texte zur Programmlaufzeit angezeigt werden, muss die Datei in ein Binärformat konvertiert werden. Diese Konvertierung erfolgt nur im Hochlauf.

12.2.1 Kennungen der Standardtexte

Kennungen der Standardtexte

Mit Hilfe der Kennungen können Sie die Standardtexte von OEM-Parametern, Kühlmitteln und werkzeugspezifischen Funktionen ändern. Dies betrifft die Spaltenüberschrift in den Werkzeuglisten, den Tooltip und falls vorhanden auch die Bezeichnung in den Detailbildern.

Nachfolgend erhalten Sie eine Auflistung der Textkennungen:

OEM-Werkzeugparameter

`"ToolUser_1"` bis `"ToolUser_10"`

`"ToolAppl_1"` bis `"ToolAppl_10"`

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
`TM_HL_TOOL_USER_1` bis `...10`	`TM_TT_TOOL_USER_1` bis `...10`
`TM_HL_TOOL_APPL_1` bis `...10`	`TM_TT_TOOL_APPL_1` bis `...10`

Textkennungen in Werkzeugdetails	Textkennungen in Werkzeugdetails - alle Parameter
TM_HL_TOOL_USER_1_DETAILS bis ...10_DETAILS	TM_HL_TOOL_USER_1_LONG bis ...10_LONG
TM_HL_TOOL_APPL_1_DETAILS bis ...10_DETAILS	TM_HL_TOOL_APPL_1_LONG bis ...10_LONG

OEM-Schneidenparameter

"EdgeUser_1" bis "EdgeUser_10"

"EdgeAppl_1" bis "EdgeAppl_10"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
TM_HL_EDGE_USER_1 bis ...10	TM_TT_EDGE_USER_1 bis ...10
TM_HL_EDGE_APPL_1 bis ...10	TM_TT_EDGE_APPL_1 bis ...10

Textkennungen in Werkzeugdetails	Textkennungen in Werkzeugdetails - alle Parameter
TM_HL_EDGE_USER_1_DETAILS bis ...10_DETAILS	TM_HL_EDGE_USER_1_LONG bis ...10_LONG
TM_HL_EDGE_APPL_1_DETAILS bis ...10_DETAILS	TM_HL_EDGE_APPL_1_LONG bis ...10_LONG

OEM-Überwachungsparameter

"EdgeSupUser_1" bis "EdgeSupUser_10"

"EdgeSupAppl_1" bis "EdgeSupAppl_10"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
TM_HL_EDGE_SUPUSER_1 bis ...10	TM_TT_EDGE_SUPUSER_1 bis ...10
TM_HL_EDGE_SUPAPPL_1 bis ...10	TM_TT_EDGE_SUPAPPL_1 bis ...10

Textkennungen in Werkzeugdetails	Textkennungen in Werkzeugdetails - alle Parameter
TM_HL_EDGE_SUPUSER_1_DETAILS bis ...10_DETAILS	TM_HL_EDGE_SUPUSER_1_LONG bis ...10_LONG
TM_HL_EDGE_SUPAPPL_1_DETAILS bis ...10_DETAILS	TM_HL_EDGE_SUPAPPL_1_LONG bis ...10_LONG

OEM-Multitoolparameter

"MultiToolUser_1" bis "MultiToolUser_10"

"MultiToolAppl_1" bis "MultiToolAppl_10"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
TM_HL_MULTITOOL_USER_1 bis ...10	TM_TT_MULTITOOL_USER_1 bis ...10
TM_HL_MULTITOOL_APPL_1 bis ...10	TM_TT_MULTITOOL_APPL_1 bis ...10

Textkennungen in Werkzeugdetails	Textkennungen in Werkzeugdetails - alle Parameter
TM_HL_MULTITOOL_USER_1_DETAILS bis ...10_DETAILS	TM_HL_MULTITOOL_USER_1_LONG bis ...10_LONG
TM_HL_MULTITOOL_APPL_1_DETAILS bis ...10_DETAILS	TM_HL_MULTITOOL_APPL_1_LONG bis ...10_LONG

OEM-Magazinplatzparameter

"MagPlaceUser_1" bis "MagPlaceUser_10"

"MagPlaceAppl_1" bis "MagPlaceAppl_10"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
TM_HL_MAGPLACE_USER_1 bis ...10	TM_TT_MAGPLACE_USER_1 bis ...10
TM_HL_MAGPLACE_APPL_1 bis ...10	TM_TT_MAGPLACE_APPL_1 bis ...10

OEM-Multitoolplatzparameter

"MultiToolPlaceUser_1" bis "MultiToolPlaceUser_10"

"MultiToolPlaceAppl_1" bis "MultiToolPlaceAppl_10"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
TM_HL_MTPLACE_USER_1 bis ...10	TM_TT_MTPLACE_USER_1 bis ...10
TM_HL_MTPLACE_APPL_1 bis ...10	TM_TT_MTPLACE_APPL_1 bis ...10

Kühlmittel und werkzeugspezifische Funktionen

"Coolant1" und "Coolant2"

"MFunction1" bis "MFunction4"

Textkennung der Spaltenüberschrift	Textkennung des Tooltips
Keine Spalte	TM_TT_STATE_COOL_1 und ...2
TM_HL_MFCT1 bis ...4	TM_TT_MFCT1 bis ...4

12.2.2 Beispiele für OEM-Texte

Beispiel 1 mit einzeiligem und zweizeiligem Text

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>
<!--    ********************************************************** -->
<!--    Definition of OEM-Texts
<!--    ********************************************************** -->
```

```
<context>
   <name>SlTmListForm</name>
<!--      ************************************************************ -->
<!--      enter your text behind this comment -->
<!--      ************************************************************ -->
<!--      ************************************************************ -->
<!--      1st example of a single-line text -->
<!--      ************************************************************ -->
<!--   <message>
        <source>MY_NEW_TEXT</source>
        <translation>Mein neuer Text</translation>
        <chars>30</chars>
      </message>-->
<!--      ************************************************************ -->
<!--      2nd example of a double spaced text -->
<!--      ************************************************************ -->
<!--    <message>
        <source>MY_NEW_TEXT_2_LINES</source>
        <translation>Mein neuer%n Text</translation>
        <remark>Mein Kommentar zum Text</remark>
        <chars>10</chars>
        <lines>2</lines>
      </message> -->
<!--      ************************************************************ -->
<!--      enter your text ahead this comment -->
<!--      ************************************************************ -->
</context>
</TS>
```

Beispiel 2

Im folgenden Beispiel aus Kapitel Listenparameter konfigurieren (Seite 309) wird ein neuer Parameter konfiguriert:

```
<PARAMETERCONFIGURATION>
     <EdgeUser_1_Bit0>
        <Base value="EdgeUser_1" type="QString" />
        <ItemType value="CheckBox" type="QString" />
        <DisplayMode value="AnyMode" type="QString" />
        <BitMask value="1" type="int" />
        <Width value="17" type="int" />
        <HeadLine value="TM_HL_EDGE_USER_1_Bit0" type="QString" />
        <ToolTip value="TM_TT_EDGE_USER_1_Bit0" type="QString" />
     </EdgeUser_1_Bit0>
</PARAMETERCONFIGURATION>
```

Die beiden Texte unter HeadLine und ToolTip sollen sprachabhängig ausgegeben werden. Dazu müssen die beiden Textkennungen in die Datei sltmlistdialog_xxx.ts aufgenommen werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>
<!--     ********************************************************** -->
<!--     Definition of OEM-Texts
<!--     ********************************************************** -->
<context>
   <name>SlTmListForm</name>
<!--     ********************************************************** -->
<!--     enter your text behind this comment -->
<!--     ********************************************************** -->
<!--     ********************************************************** -->
<!--     Text fuer HeadLine uebersetzen -->
<!--     ********************************************************** -->
      <message>
        <source>TM_HL_EDGE_USER_1_Bit0</source>
        <translation>Schweres %n Werkzeug</translation>
        <chars>10</chars>
        <lines>2</lines>
      </message>
<!--     ********************************************************** -->
<!--     Text fuer Tooltip uebersetzen -->
<!--     ********************************************************** -->
      <message>
        <source>TM_TT_EDGE_USER_1_Bit0</source>
        <translation>Schweres Werkzeug</translation>
        <chars>20</chars>
      </message>
<!--     ********************************************************** -->
<!--     enter your text ahead this comment -->
<!--     ********************************************************** -->
</context>
</TS>
```

12.3 Beispiele

12.3.1 Beispiel: OEM-Werkzeugliste konfigurieren

12.3.1.1 Anpassung der Konfigurationsdatei

Der folgende Auszug zeigt den Inhalt der Konfigurationsdatei:

- Technologie Fräsen: sltmlistconfig.xml
- Technologie Drehen: sltmturninglistconfig.xml

Die Datei in diesem Beispiel wird in das Verzeichnis "user/sinumerik/hmi/cfg" abgelegt.

Dieses Beispiel beschreibt eine Konfiguration der OEM-Werkzeugliste.

- In der Spalte 6 wird der OEM-Werkzeugparameter 1 angezeigt.
- In der Spalte 7 wird ein eigener neuer Parameter "Kühlmittel 3" eingefügt, der Bit0 aus dem OEM-Werkzeugparameter 2 anzeigt.
- In der Spalte 8 wird der OEM-Schneidenparameter 1 angezeigt.

Beispiel

Datei "sltmlistconfig.xml" bzw. "sltmturninglistconfig.xml":

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
   <!-- **** Kennung fuer die Konfiguration der Listen *** -->
   <LISTCONFIGURATION>
     <!-- **** OEM - Werkzeugliste *** -->
     <SlTmTooloemForm>
       <!-- **** freischalten der OEM - Werkzeugliste *** -->
       <Enabled value="true" type="bool" />
       <!-- **** Die Spalten 1 bis 5 sind bereits vom Standard belegt. *** -->
       <!-- **** Spalte 6 *** -->
       <COLUMN6>
         <!-- **** fuer alle Werkzeugtypen *** -->
         <TOOLTYPE_ALL>
           <!-- **** OEM-Werkzeugparameter 1 anzeigen *** -->
           <Item value="ToolUser_1" type="QString" />
         </TOOLTYPE_ALL>
       </COLUMN6>
       <!-- **** Spalte 7 *** -->
       <COLUMN7>
         <!-- **** fuer alle Werkzeugtypen *** -->
         <TOOLTYPE_ALL>
```

```
            <!-- **** vorrangiges Werkzeug, eigener Parameter *** -->
            <!-- **** OEM-Werkzeugparameter 2 Bit0 anzeigen *** -->
            <Item value="PriorTool" type="QString" />
          </TOOLTYPE_ALL>
        </COLUMN7>
        <!-- **** Spalte 8 *** -->
        <COLUMN8>
          <!-- **** fuer alle Werkzeugtypen *** -->
          <TOOLTYPE_ALL>
            <!-- **** OEM-Schneidenparameter 1 anzeigen *** -->
            <Item value="EdgeUser_1" type="QString" />
          </TOOLTYPE_ALL>
        </COLUMN8>
      </SlTmTooloemForm>
    </LISTCONFIGURATION>
    <!-- **** Kennung fuer die Konfiguration eigener Parameter *** -->
    <PARAMETERCONFIGURATION>
      <!-- **** OEM-Werkzeugparameter 1 aendern *** -->
      <ToolUser_1>
        <!-- **** Spaltenbreite *** -->
        <Width value="36" type="int" />
        <!-- **** Typaenderung in unsigned integer *** -->
        <DisplayMode value="UnsignedIntegerMode" type="QString" />
      </ToolUser_1>
      <!-- **** Kennung des eigenen Parameters *** -->
      <!-- **** vorrangiges Werkzeug *** -->
      <PriorTool>
        <!-- ** neuer Parameter hat als Grundlage den OEM-Werkzeugparameter 2**-->
        <Base value="ToolUser_2" type="QString" />
        <!-- **** Bit 0 wird ausgewertet *** -->
        <BitMask value="1" type="int" />
        <!-- **** Textkennung der Ueberschrift *** -->
        <HeadLine value="TMO_HL_PRIOR_TOOL" type="QString" />
        <!-- **** Textkennung des Tooltips *** -->
        <ToolTip value="TMO_TT_PRIOR_TOOL" type="QString" />
        <!-- **** Spaltenbreite *** -->
        <!-- **** empfohlene Breiten 640x480 - 19 Pixel *** -->
        <!-- **** 800x600 - 21, 1024x768 - 17 *** -->
        <Width value="19" type="int" />
        <!-- **** Anzeigeitem des Parameters *** -->
        <ItemType value="CheckBox" type="QString" />
      </PriorTool>
    </PARAMETERCONFIGURATION>
</CONFIGURATION>
```

12.3.1.2 Anpassung der Kundentextdatei

Der folgende Auszug zeigt den Inhalt der Kundentextdatei "sltmlistdialog_deu.ts".

Die Datei in diesem Beispiel wird in das Verzeichnis "user/sinumerik/hmi/lng" abgelegt.

Beispiel

Datei "sltmlistdialog_deu.ts":

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>

<!-- ************************************************************ -->
<!-- Definition of OEM-Texts                                      -->
<!-- ************************************************************ -->
<context>
     <name>SlTmListForm</name>

<!-- ************************************************************ -->
<!-- enter your text behind this comment                          -->
<!-- ************************************************************ -->

        <!-- **** Ueberschrift OEM-Werkzeugparameter 1 *** -->
        <!-- **** Gewicht  *** -->
        <message>
          <!-- **** Textkennung *** -->
          <source>TM_HL_TOOL_USER_1</source>
          <!-- **** Text, der ausgegeben wird *** -->
          <translation>Ge-%nwicht</translation>
          <!-- **** Kommentar, nur fuer den Uebersetzer *** -->
          <remark>vorrangiges Werkzeug</remark>
          <!-- **** maximale Zeilenlaenge in Zeichen, *** -->
          <!-- **** nur fuer den Uebersetzer *** -->
          <chars>7</chars>
          <!-- **** Anzahl der moeglichen Zeilen *** -->
          <!-- **** nur fuer den Uebersetzer *** -->
          <lines>2</lines>
        </message>
        <!-- **** Tooltip OEM-Werkzeugparameter 1 *** -->
        <!-- **** Gewicht  *** -->
        <message>
          <source>TM_TT_TOOL_USER_1</source>
          <translation>Gewicht des Werkzeugs</translation>
          <chars>35</chars>
          <lines>2</lines>
```

```
    </message>
    <!-- **** Ueberschrift fuer eigenen Parameter *** -->
    <!-- **** vorrangiges Werkzeug  *** -->
    <message>
      <source>TMO_HL_PRIOR_TOOL</source>
      <translation>V</translation>
      <remark>vorrangiges Werkzeug</remark>
      <chars>1</chars>
      <lines>2</lines>
    </message>
    <!-- **** Tooltip fuer eigenen Parameter *** -->
    <!-- **** vorrangiges Werkzeug  *** -->
    <message>
      <source>TMO_TT_PRIOR_TOOL</source>
      <translation>vorrangiges Werkzeug,%nWerkzeug bevorzugt verwenden</
      translation>
      <chars>35</chars>
      <lines>3</lines>
    </message>
    <!-- **** Ueberschrift OEM-Schneidenparameter 1 *** -->
    <!-- **** maximale Werkzeugbreite  *** -->
    <message>
      <source>TM_HL_EDGE_USER_1</source>
      <translation>maximale%nBreite</translation>
      <chars>8</chars>
      <lines>2</lines>
    </message>
    <!-- **** Tooltip OEM-Schneidenparameter 1 *** -->
    <!-- **** maximale Werkzeugbreite  *** -->
    <message>
      <source>TM_TT_EDGE_USER_1</source>
      <translation>maximale Breite%ndes Werkzeugs%nüber alles</
      translation>
      <chars>35</chars>
      <lines>2</lines>
    </message>

<!-- ************************************************************ -->
<!-- enter your text ahead this comment                           -->
<!-- ************************************************************ -->

</context>
</TS>
```

12.3.2 Beispiel: Magazinplatztypen mit Namen konfigurieren

12.3.2.1 Anpassung der Konfigurationsdatei

Der folgende Auszug zeigt den Inhalt der Konfigurationsdatei:

- Technologie Fräsen: sltmlistconfig.xml
- Technologie Drehen: sltmturninglistconfig.xml

Die Datei in diesem Beispiel wird in das Verzeichnis "user/sinumerik/hmi/cfg" abgelegt.

Dieses Beispiel beschreibt eine Konfiguration der Platztypen in der Magazinliste.

Der Magazinplatztyp und der Werkzeugplatztyp sollen in diesem Beispiel als Togglefeld mit Klartexten angezeigt werden.

Im Bereich "LISTCONFIGURATION" wird die Zuordnung der Parameter zu den Spalten der Liste getroffen. In der Spalte 10 wird der Magazinplatztyp angezeigt. In der Spalte 11 wird der Werkzeugplatztyp angezeigt.

Im Bereich "PLACETYPECONFIGURATION" werden die Texte den Platztypen zugeordnet. Die Nummer des Platztyps ist die Ziffer am Ende der Kennung "PLACETYPE".
Es besteht pro Platztyp die Möglichkeit, die Textkennung für einen Toggleboxeintrag und eine Textkennung für einen Tooltip zu hinterlegen.

Beispiel

Datei "sltmlistconfig.xml" bzw. "sltmturninglistconfig.xml":

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
   <!-- **** Konfiguration der Listen *** -->
   <LISTCONFIGURATION>
     <!-- **** Magazinliste *** -->
     <SlTmToolmagazinForm>
       <!-- **** Spalte 10 *** -->
       <COLUMN10>
         <!-- **** fuer alle Werkzeugtypen *** -->
         <TOOLTYPE_ALL>
           <!-- **** Magazinplatztyp anzeigen *** -->
           <Item value="MagPlaceTypeIdent" type="QString" />
         </TOOLTYPE_ALL>
       </COLUMN10>
       <!-- **** Spalte 11 *** -->
       <COLUMN11>
         <!-- **** fuer alle Werkzeugtypen *** -->
         <TOOLTYPE_ALL>
           <!-- **** Werkzeugplatztyp anzeigen *** -->
           <Item value="ToolPlaceSpecIdent" type="QString" />
         </TOOLTYPE_ALL>
```

```
        <!-- **** fuer alle Multitools *** -->
        <TOOLTYPE_9997>
          <!-- **** fuer alle Multitoolplatztyp anzeigen *** -->
          <Item value="MultiToolPlaceSpecIdent" type="QString" />
        </TOOLTYPE_9997>
      </COLUMN11>
    </SlTmToolmagazinForm>
  </LISTCONFIGURATION>
  <!-- **** Konfiguration der Magazinplatztypen *** -->
  <PLACETYPECONFIGURATION>
    <!-- **** Magazinplatztyp 1 *** -->
    <PLACETYPE1>
      <!-- **** Textkennung des Toggleboxeintrags *** -->
      <Text value="TMMP_ITEM_NORMAL_TOOL" type="QString" />
      <!-- **** Textkennung des Tooltips *** -->
      <Tooltip value="TMMP_TT_NORMAL_TOOL" type="QString" />
    </PLACETYPE1>
    <!-- **** Magazinplatztyp 2 *** -->
    <PLACETYPE2>
      <Text value="TMMP_ITEM_HEAVY_TOOL" type="QString" />
      <Tooltip value="TMMP_TT_HEAVY_TOOL" type="QString" />
    </PLACETYPE2>
    <!-- **** Magazinplatztyp 3 *** -->
    <PLACETYPE3>
      <Text value="TMMP_ITEM_BIG_TOOL" type="QString" />
      <Tooltip value="TMMP_TT_BIG_TOOL" type="QString" />
    </PLACETYPE3>
    <!-- **** Magazinplatztyp 4 *** -->
    <PLACETYPE4>
      <Text value="TMMP_ITEM_SENSITIVE_TOOL" type="QString" />
      <Tooltip value="TMMP_TT_SENSITIVE_TOOL" type="QString" />
    </PLACETYPE4>
  </PLACETYPECONFIGURATION>
</CONFIGURATION>
```

12.3.2.2 Anpassung der Kundentextdatei

Der folgende Auszug zeigt den Inhalt der Kundentextdatei "sltmlistdialog_deu.ts".

Die Datei in diesem Beispiel wird in das Verzeichnis "user/sinumerik/hmi/lng" abgelegt.

Beispiel

Datei "sltmlistdialog_deu.ts":

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>

<!-- ************************************************************ -->
<!-- Definition of OEM-Texts -->
<!-- ************************************************************ -->
<context>
     <name>SlTmListForm</name>

<!-- ************************************************************ -->
<!-- enter your text behind this comment -->
<!-- ************************************************************ -->
     <!--  **** Item Magazinplatztyp 1 *** -->
     <!--  **** normales Werkzeug *** -->
     <message>
       <!-- **** Textkennung *** -->
       <source>TMMP_ITEM_NORMAL_TOOL</source>
       <!-- **** Text, der ausgegeben wird *** -->
       <translation>normal</translation>
       <!-- **** maximale Zeilenlaenge in Zeichen, *** -->
       <!-- **** nur fuer den Uebersetzer *** -->
       <chars>9</chars>
     </message>
     <!-- **** Tooltip Magazinplatztyp 1 *** -->
     <!-- **** normales Werkzeug *** -->
     <message>
       <source>TMMP_TT_NORMAL_TOOL</source>
       <translation>normales Werkzeug,%nkeine Einschränkungen</translation>
       <chars>35</chars>
       <lines>2</lines>
     </message>
     <!-- **** Item Magazinplatztyp 2 *** -->
     <!-- **** schweres Werkzeug *** -->
     <message>
       <source>TMMP_ITEM_HEAVY_TOOL</source>
       <translation>schwer</translation>
       <chars>9</chars>
     </message>
     <!-- **** Tooltip Magazinplatztyp 2 *** -->
     <!-- **** schweres Werkzeug *** -->
     <message>
```

```
      <source>TMMP_TT_HEAVY_TOOL</source>
      <translation>schweres Werkzeug</translation>
      <chars>35</chars>
      <lines>2</lines>
    </message>
    <!-- **** Item Magazinplatztyp 3 *** -->
    <!-- **** grosses Werkzeug *** -->
    <message>
      <source>TMMP_TT_BIG_TOOL</source>
      <translation>groß</translation>
      <chars>9</chars>
    </message>
    <!-- **** Tooltip Magazinplatztyp 3 *** -->
    <!-- **** grosses Werkzeug *** -->
    <message>
      <source>TMMP_TT_BIG_TOOL</source>
      <translation>großes Werkzeug</translation>
      <chars>35</chars>
      <lines>2</lines>
    </message>
    <!-- **** Item Magazinplatztyp 4 *** -->
    <!-- **** empfindliches Werkzeug *** -->
    <message>
      <source>TMMP_ITEM_SENSITIVE_TOOL</source>
      <translation>empfindl.</translation>
      <chars>9</chars>
    </message>
    <!-- **** Tooltip Magazinplatztyp 4 *** -->
    <!-- **** empfindliches Werkzeug *** -->
    <message>
      <source>TMMP_TT_SENSITIVE_TOOL</source>
      <translation>Achtung: Werkzeug kann%nleicht zerstört werden</
      translation>
      <chars>35</chars>
      <lines>2</lines>
    </message>

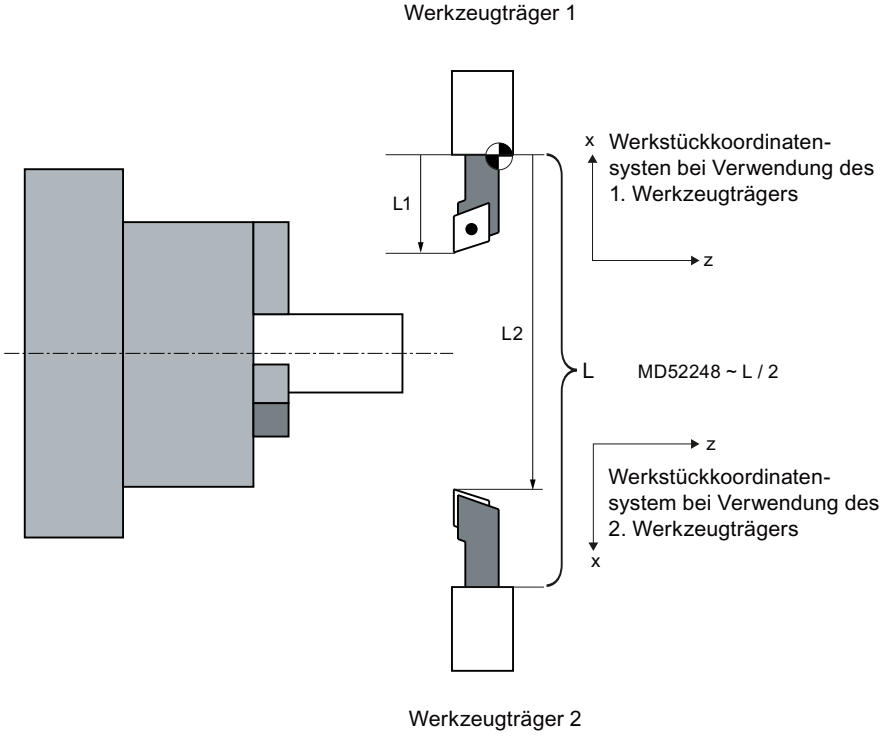
<!-- ************************************************************ -->
<!-- enter your text ahead this comment -->
<!-- ************************************************************ -->

</context>
</TS>
```

12.4 Arbeiten mit zwei Werkzeugträgern

Es kann an einer Drehmaschine mit zwei Werkzeugträgern, die beide auf einer X-Achse montiert sind, gearbeitet werden. Die Werkzeugträger können Revolver, Multifix oder eine Kombination aus beiden sein. Da beide Werkzeugaufnahmen auf derselben Achse montiert sind, kann immer nur mit einem Werkzeug gearbeitet werden. Das Werkstück befindet sich zwischen den beiden Werkzeugaufnahmen.

Alle Werkzeuge haben denselben Bezugspunkt (am Werkzeugträger 1), unabhängig davon auf welchem Werkzeugträger sie sich befinden. Daher sind die Werkzeuglängen vom zweiten Werkzeugträger immer länger als die Werkzeuglängen vom ersten Werkzeugträger. Die Hauptbearbeitung erfolgt in negativer X-Achsrichtung.



Werkzeugträgerdefinition

Über das MD52248 REV_2_BORDER_TOOL_LENGTH wird eine Werkzeuglängengrenze X definiert, damit die Werkzeuge automatisch den beiden Werkzeugträgern zugeordnet werden können:

Werkzeuglänge X < MD52248: Werkzeug gehört zum 1. Werkzeugträger
Werkzeuglänge X ≥ MD52248: Werkzeug gehört zum 2. Werkzeugträger

Weitere Parametrierungen

Für die verwendeten Spiegelungen der X-, Y-Achse und Verschiebungen der C-Achse müssen folgende Maschinendaten konfiguriert werden:

MD24006 $MC_CHSFRAME_RESET_MASK	Aktive Systemframes nach RESET
Bit 4 = 1	Systemframe Werkstückbezug bleibt nach RESET aktiv

MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK	Löschen von Systemframes nach RESET
Bit 4 = 0	Systemframe Werkstückbezug nach RESET nicht löschen

MD28082 $MC_MM_SYSTEM_FRAME_MASK	Systemframes (SRAM) einrichten
Bit 4 = 1	Werkstückbezug ($P_WPFRAME)

MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK	Datenhaltung Systemframes (SRAM) einrichten
Bit 4 = 1	Werkstückbezug ($P_WPFRAME)

Wegen der Spiegelung des Koordinatensystems müssen folgende Maschinen- bzw. Settingdaten gesetzt werden:

MD21202 $MC_LIFTFAST_WITH_MIRROR	Schnellabheben mit Spiegeln
= 1	Bei der Bestimmung der Abheberichtung wird bei aktiver Spiegelung der Kontur auch die Abheberichtung gespiegelt. Die Spiegelung der Abheberichtung bezieht sich nur auf die Richtungskomponenten senkrecht zur Werkzeugrichtung.

SD42900 $SC_MIRROR_TOOL_LENGTH	Vorzeichenwechsel Werkzeuglänge beim Spiegeln
= 0	Die Vorzeichen der Werkzeuglängenkomponenten sind gegenüber Systemframes mit Spiegelungen unabhängig.

SD42910 $SC_MIRROR_TOOL_WEAR	Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln
= 1	Durch Vorzeicheninvertierung werden die Verschleißwerte der Werkzeuglängenkomponenten, deren zugehörige Achsen gespiegelt sind, gespiegelt.

Einstellungen bei ShopTurn-Programmen

Bei ShopTurn werden MD21202, SD42900 und SD42910 für die notwendigen Spiegelungen automatisch eingestellt.

Einstellungen bei G-Code-Programmen

Wenn die entsprechenden Funktionen des zweiten Werkzeugträgers mit Spiegelung der X-Achse auch für G-Code-Programme und für Werkzeugaufrufe über die T,S,M-Maske wirksam werden sollen, müssen weitere Einstellungen vorgenommen werden:

MD10717 $MN_T_NO_FCT_CYCLE_NAME="F_T_REV2"

MD15710 $MN_TCA_CYCLE_NAME="F_T_REV2"

Hinweis

Die Einstellungen von MD10717 und MD15710 sind nicht nutzbar, wenn alte G-Code-Programme für den zweiten Werkzeugträger ohne Spiegelung der X-Achse verfügbar sind und unverändert weiter verwendet werden sollen.

Wenn die Transformation TRANSMIT eingerichtet ist, können bei entsprechender Programmierung die notwendigen Spiegelungen auch automatisch aktiviert werden. Dazu muss der TRANSMIT-Befehl durch ein Unterprogramm (z. B. im Herstellerverzeichnis) ersetzt werden.

```
MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[0]="TRANSMIT"
MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[1]="TRANSM"
```

Unterprogramm TRANSMIT.SPF:

```
PROC TRANSMIT(INT_NR)
F_T_REV2
TRANSM(_NR)
F_T_REV2
RET
```

Hinweis

In der Statusanzeige wird nun anstatt TRANSMIT nur TRANSM angezeigt.

Randbedingungen

- Das Kegeldrehen ist bei "Arbeiten mit zwei Werkzeugträgern" nicht freigegeben.
- Die B-Achsfunktion (Kinematik) ist bei "Arbeiten mit zwei Werkzeugträgern" ebenso nicht freigegeben.

Anbindung Codeträger - Tool Ident Connection (Option)

13

13.1 Installation

13.1.1 Systemvoraussetzungen

Tool Ident Connection ist für das Zielsystem SINUMERIK 828D mit SINUMERIK Operate vorgesehen.

Hinweis

Tool Ident Connectionl ist nur in Verbindung mit der Bedien-Software SINUMERIK Operate ab SW 4.9 SP4 verfügbar.

PLC-Schnittstelle

AUTOHOTSPOT

Kommando-Codes (Seite 408)

Werkzeugverwaltung

Zur Verwendung der Option Tool Ident Connection ist eine aktive Magazinverwaltung erforderlich.

DB1900

Der PLC-Datenbaustein DB1900 ist ab V4.9 SP4 über die PLC-Schnittstelle für "Tool Ident Connectionl" verfügbar, siehe AUTOHOTSPOT.

Hardware-Konfiguration für Lese-/Schreibgeräte

Die Schreib-/Lesegeräte für die Werkzeugdaten werden als PROFINET IO-Geräte an die SINUMERIK 828D angeschlossen. Die Hardwarekonfiguration ist in SDBs hinterlegt. Für die SINUMERIK 828D ist die Busbelegung für verschiedene Standardkonfigurationen vordefiniert und in vorgefertigten SDBs eingetragen.

Lese-/Schreibkopf von Balluff verwenden

Bei Verwendung eines Lese-/Schreibkopfs der Firma Balluff muss der Sonder-SDB wie ein Software-Update installiert werden. Der Sonder-SDB ist in Pridanet unter "SINUMERIK 828D Software" zu finden.

Wie Sie mit Balluff Komponenten umgehen, finden Sie in folgendem FAQ (SIOS) beschrieben:

FAQ-Nummer: 109797865

Titel: SINUMERIK 828: Welche Balluff Komponenten sind für Tool Ident freigegeben?

13.1.2 Installation von Tool Ident Connection

Hinweis

Vor dem Einrichten wird empfohlen ein Backup des Systems anzulegen.

Gemäß den Systemvoraussetzungen müssen zuvor, wenn notwendig, die Zusatzkomponenten installiert werden.

Das Speichermedium für Softwareapplikationen ist eine CompactFlash Card mit fester Verzeichnisstruktur. Die Applikation "Tool Ident Connection" wird innerhalb dieser Struktur im Verzeichnis "addon" abgelegt.

Hardware-Konfiguration für Lese-/Schreibgeräte

Die Schreib-/Lesegeräte für die Werkzeugdaten werden als PROFINET IO-Geräte an die SINUMERIK 828D angeschlossen. Die Hardwarekonfiguration ist in SDBs hinterlegt. Für die SINUMERIK 828D ist die Busbelegung für verschiedene Standardkonfigurationen vordefiniert und in vorgefertigten SDBs eingetragen. Diese SDBs werden mit der Steuerung ausgeliefert.

Für Balluff-Geräte muss der SDB installiert werden.

13.1.2.1 Aktivieren von Tool Ident Connection

Konfigurationsdateien

Zur Konfiguration von Tool Ident Connection stehen folgende Dateien zur Verfügung:

Standard Konfigurationsdateien	Verzeichnis: CompactFlash Card
tdiidentcfg.xml	/siemens/sinumerik/hmi/template\toolidentconnection
toolSpec.xml	
wkonvert.mcx	
DefToolDat.txt	
wkonvert.mcc	
systemconfiguration.ini	
oem_sltmlistconfig.xml	

Einrichtung der Option Tool Ident Connection

Zum Einrichten der Software muss in der "systemconfiguration.ini" in der Sektion [services] folgender Eintrag getätigt werden:

```
SVCxxx=name:=SlMcisTdiIdentConnectionService,
implementation:=slmcistdiidentconnectionservice.SlMcisTdiIdentConnectionService,
process:=SlHmiHost1,
```

Hinweis

Das `xxx` in `SVCxxx` muss durch die freie Nummer eines Services ersetzt werden. Es dürfen nur Nummern ab 200 aufwärts verwendet werden.

Für Anpassungen der Funktion Tool Ident Connection Konfiguration müssen die systemconfiguration.ini und soweit sie angepasst wurden auch die Konfigurationsdateien (aus dem oben genannten Verzeichnis) in folgendes Verzeichnis kopiert werden.

- CompactFlash Card: /addon/sinumerik/hmi/cfg

Die Datei oem_sltmlistconfig.xml muss dabei, abhängig von der Technologie, umbenannt werden. Dies wird im Kapitel "Bedienoberfläche konfigurieren" erklärt.

Eine ausführliche Beschreibung der Konfigurationsmöglichkeiten der oem_sltmlistconfig.xml befindet sich in Bedienoberfläche konfigurieren (Seite 281).

Die Dateien wkonvert.txt, DefToolDat.txt und wkonvert.mcc sind Konfigurationsdateien. Sie müssen nach jeder Änderung durch den wkonvert-Wizard compiliert werden. Das Resultat des Compile-Vorgangs ist eine wkoonvert.mcx. Diese muss in das entsprechende Zielverzeichnis kopiert werden, damit die Einstellungen wirksam werden.

13.1.2.2 Installation Wkonvert-Wizard

Zum Erstellen der benötigten Konvertierungsvorschriften für Tool Ident Connection steht die PC-Applikation Wkonvert-Wizard zur Verfügung.

Um den Wizard nutzen zu können, muss die Applikation auf einen Projektierungs-Rechner installiert werden.

Installationsverlauf

Zur Installation des Wkonvert-Wizard muss die setup.exe, die sich auf der SINUMERIK CNC Software DVD befindet, ausgeführt werden.



Bild 13-1 Sprachauswahl-Dialog

Nach der Auswahl der Installationssprache folgen die einzelnen Installationsschritte.

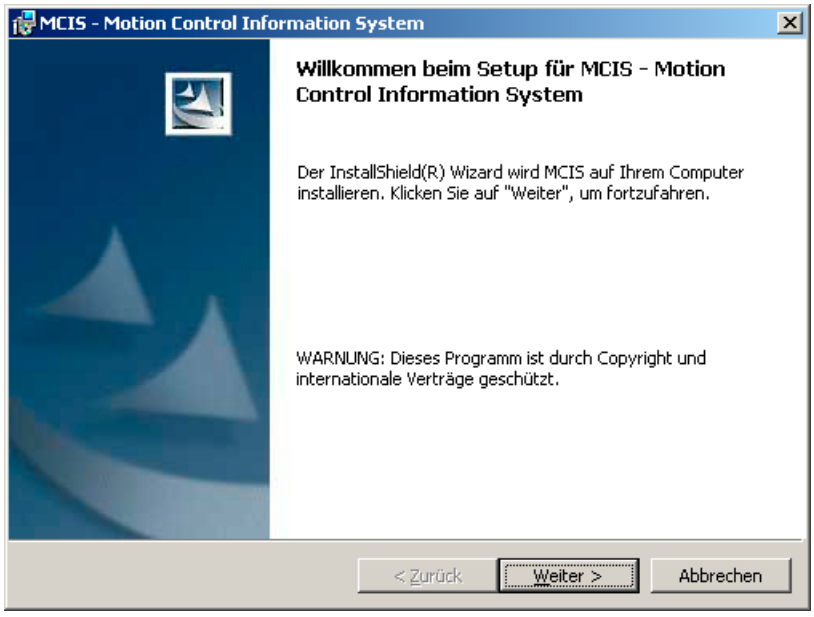


Bild 13-2 Setup vorbereiten

Nach dem Lesen der Lizenzvereinbarung müssen diese akzeptiert werden. Wenn Sie mit der Lizenzvereinbarung nicht einverstanden sind, wird das Setup abgebrochen.



Bild 13-3 Lizenzvereinbarungs-Dialog

Nach der Bestätigung der Lizenzvereinbarung werden Sie nach Benutzername und Organisation gefragt.

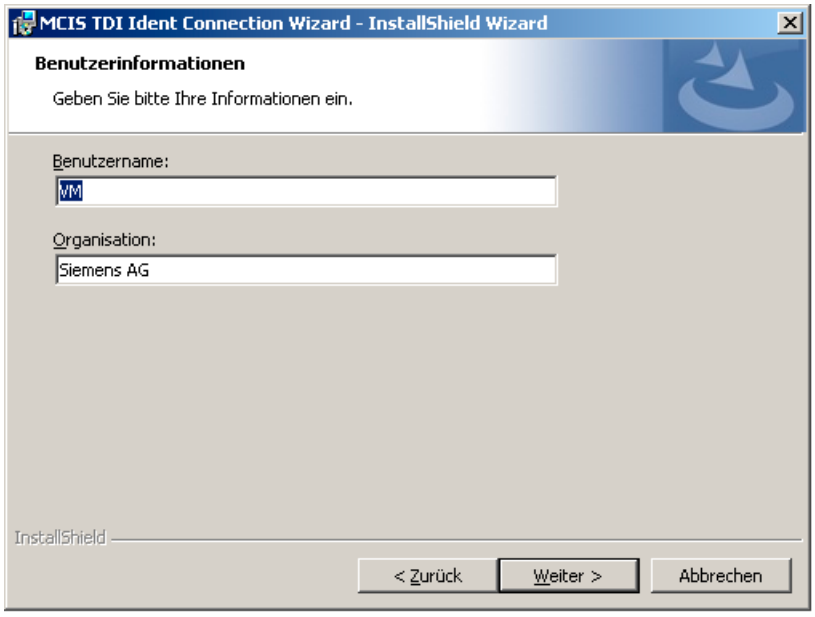


Bild 13-4 Benutzerinformations-Dialog

Anschließend haben Sie die Wahl zwischen einer vollständigen und einer benutzerdefinierten Installation.



Bild 13-5 Setuptyp-Dialog

Bei einer benutzerdefinierten Installation haben Sie die Möglichkeit das Zielverzeichnis der Applikation zu bestimmen.

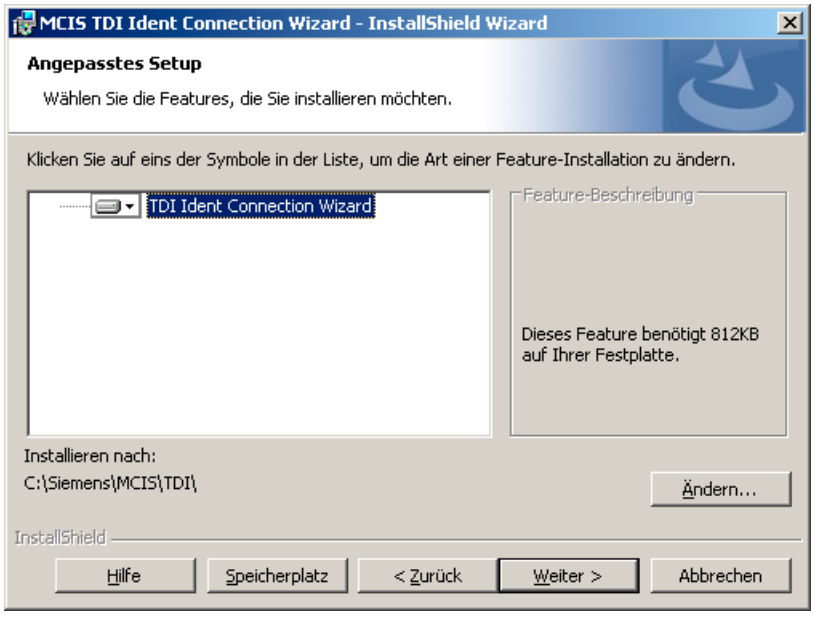


Bild 13-6 Angepasster Setup-Dialog

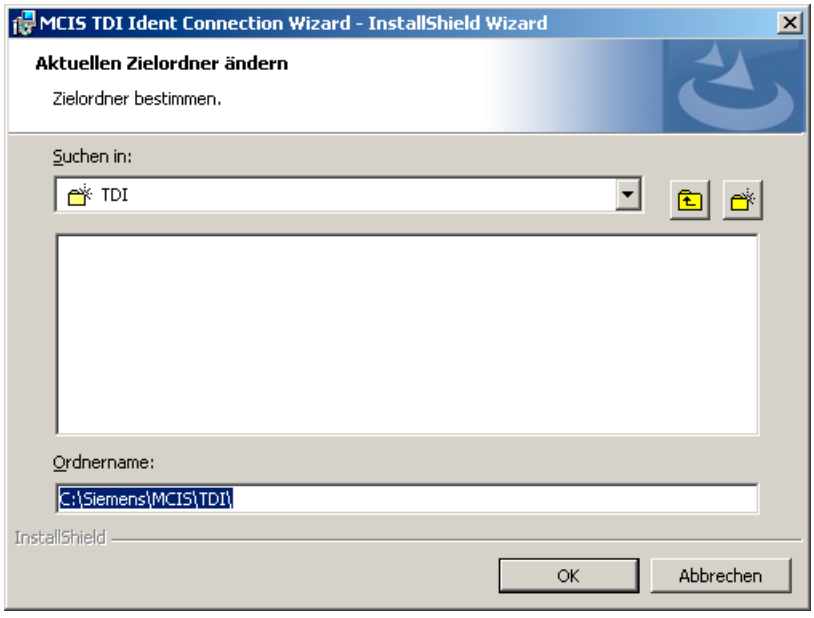


Bild 13-7 Zielordnerauswahl-Dialog

Haben Sie eine vollständige Installation gewählt, klicken Sie auf "Weiter" oder Sie klicken im Komponentenauswahl-Dialog auf "Weiter", kann mit der Installation begonnen werden.

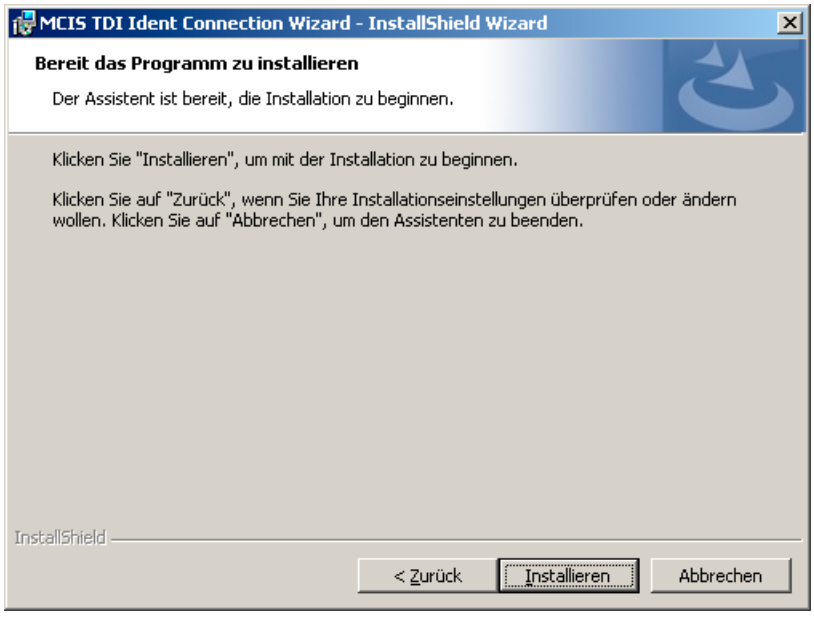


Bild 13-8 Installations-Dialog

Klicken Sie nun auf "Installieren" um die Installation zu starten.

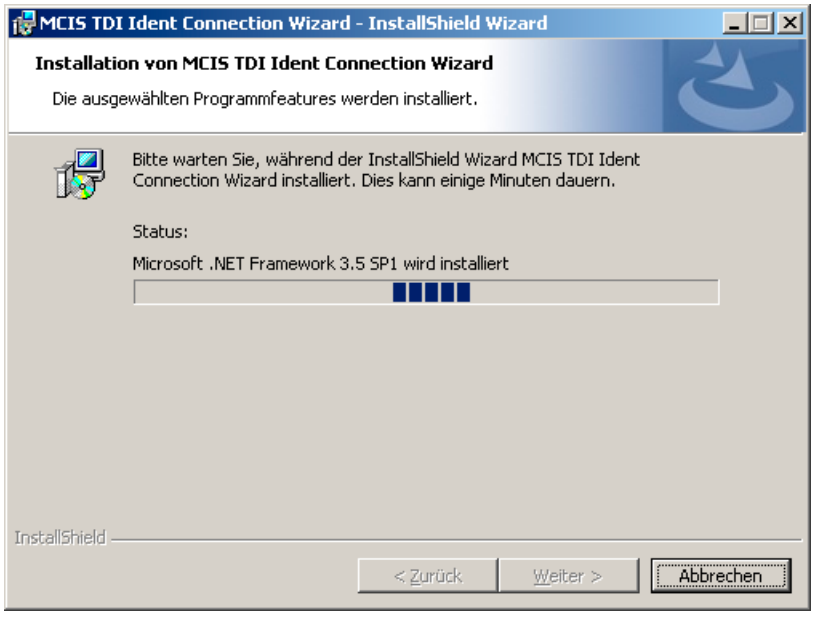


Bild 13-9 Installations-Dialog

Bei Bedarf installiert das Setup die Windows-Komponenten ".NET Framework" und "Windows Installer 3.0".

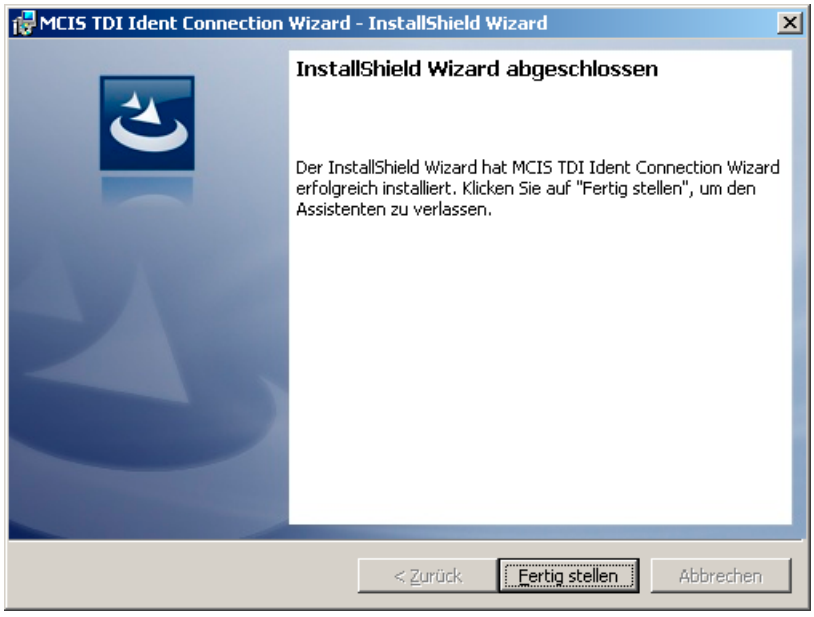


Bild 13-10 Installationsabschluss

Wenn der Setupvorgang erfolgreich durchlaufen wurde, kann der Wkonvert-Wizard über **Start > Programme > MCIS > TDI > TDI Ident Connection Wizard** gestartet werden.

13.2 Konfiguration

13.2.1 Konfiguration der Werkzeuglisten

Tag <IDENTCONNECTIONCONFIGURATION>

Im Tag <IDENTCONNECTIONCONFIGURATION> befinden sich die Einträge zur Konfiguration der Codeträgeranbindung in die Werkzeugliste der Bedienoberfläche. Die Codeträgeranbindung erfolgt über Tool Ident Connection.

Weitere Informationen dazu finden Sie unter Codeträgeranbindung konfigurieren.

13.2.2 TDIIdentCfg.xml

Zur Aktivierung der Software muss in der "systemconfiguration.ini" in der Sektion `[services]` folgender Eintrag getätigt werden:

```
[services]
SVCxxx=name:=SIMcisTdiIdentConnectionService,
implementation:=slmcistdiidentconnectionservice.SIMcisTdiIdentConnectionService,
```

```
process:=SIHmiHost1, timeout:=5000, shutdowntime:=5000
```

Hinweis

xxx in SVCxxx muss durch die freie Nummer eines Services ersetzt werden.

Tool Ident Connection wird über eine auf xml basierende Konfigurationsdatei parametriert. Über diese Konfigurationsdatei wird die Einbindung diverser Lese-/Schreibgeräte über verschiedene Kommunikationstechnologien ermöglicht und parametriert.

Aufbau

Die Konfigurationsdatei ist wie folgt aufgebaut:

```
<?xml version="1.0" encoding="UTP-8"?>
<!-- SIEMENS AG 2010 - GER I IA&DT WEST TSS 7C APC MT - ZN Köln - All rights reserved -->
<!-- Configuration-File for MCIS TDI Ident Connection sl -->
<!-- E-Mail @ kulik.michael@siemens.com -->
<!-- Do not manipulate this file -->
<Configuration>
      <Logging>
          <ErrorFile size="3" enable="0">/var/tmp/TDIIdentLog.log</ErrorFile>
          <TraceFile size="3" enable="0">/var/tmp/TDIIdentTrace.log</TraceFile>
      <Logging>
...
    <GlobalParam>
      <InterfaceDB>1900</InterfaceDB>
      <InterfaceOffset>3110</InterfaceOffset>
      <DisableInterface>false</DisableInterface>
      <TimeOut>1000</TimeOut>
      <InputDB>1900</InputDB>
      <InputOffset>3000</InputOffset>
      <OutputDB>1900</OutputDB>
      <OutputOffset>3058</OutputOffset>
    </GlobalParam>
    <!-- Parametrierung des ParamTM-Interface -->
    <ParamTMInterface enable="1">
      <InterfaceDB>1900</InterfaceDB>
      <InterfaceOffset>3116</InterfaceOffset>
    </ParamTMInterface>
...
    <!--Parametrierung des Schreib-/Lesegeraets -->
    <!--  Type = 1 -> PLC
          Type = 2 -> USB
          Type = 3 -> RJ45 -->
      <Device Type="1">
        <Units>1</Units>
        <Data Unit="1">
          <DataDB>9915</DataDB>
          <Offset>0</Offset>
          <MaxData>1000</MaxData>
          <EOT>0*2F2F</EOT>
      </Data>
    </Device>
</Configuration>
```

Beschreibung

Hinweis

Zur Parametrierung und Konfiguration von Tool Ident Connection dürfen nur die in der nachfolgenden Tabelle beschriebenen Einträge geändert/benutzt werden.

Erstellen Sie immer ein Backup vor einer Änderung.

Die Konfigurationsdatei kann mit einem Texteditor, z. B. Notepad, geändert werden. Sie ist in verschiedene Knoten unterteilt, die unterschiedliche Einträge beinhalten. Die Sektionen und ihre Einträge werden in folgender Tabelle beschrieben.

Abschnitt	Parameter	Bedeutung
GlobalParam	InterfaceDB	Datenbausteinangabe für die PLC-Schnittstelle
	InfaceOffset	Offset für die PLC-Schnittstelle
	TimeOut	Ausführungs-Überwachungszeit
	DisableInterface	De-/Aktivierung der PLC-Schnittstelle
	InputDB	Datenbaustein für Übergabeparameter
	InputOffset	Offset für die Übergabeparameter
	OutputDB	Datenbaustein für die Rückgabeparameter
	OutputOffset	Offset für die Rückgabeparameter
Logic	mcxPath	Datei + Pfadangabe zur projektierten mcx-Datei
	specPath	Datei + Pfadangabe zur ToolSpec.xml
MergeMode	LoadPlaceNo	Default Beladestelle
	UnloadPlaceNo	Default Entladestelle
Device Type="1" -> PLC	Units	Anzahl der verwendeten Leseköpfe
	DataBlock	Datenbaustein des Lesekopfs
	Offset	Offset der Daten des Lesekopfes
	MaxDataBytes	Maximale Daten des Lesekopfs in Byte
	EOT	Ende der Zeichenkette des Lesekopfes

Sollte die Ausführung der PLC-Dienste für Tool Ident Connection wegen Zeitüberschreitung abgebrochen werden, kann über den Eintrag "TimeOut" eine längere Zeit festgelegt werden. Eine Zeitüberschreitung kann z. B. auftreten, wenn das Beladen eines Werkzeugs vom Bediener nicht innerhalb des parametrierten Zeitfensters quittiert wird.

Mit dem Parameter "MergeMode" können die Default Be- und Entladeplatznummer des Beladeplatzmagazins mit den Einträgen von "LoadPlaceNo" und "UnloadPlaceNo" geändert werden.

Die Node "Device" enthält das Attribut "Type", hier wird der Typ des verwendeten Lesegeräts hinterlegt. Das Attribut bestimmt die Einträge unterhalb dieses Bereiches. Die Parameter des Bereiches enthalten die Konfigurationsdaten des Schreibe-/Lesegeräts.

Übersicht des Attributes "Type" des Parameters "Device":

Type	Bezeichnung
1	PLC-Lesegerät → Anschluss via PROFINET IO

13.2.3 Schreib-/Lesegerät an der PLC

Konfiguration PLC-Anbindung

In der Konfiguration für Tool Ident Connection muss für ein PLC-Schreib-/Lesegerät das Attribut "Type" auf die Wertigkeit "1" gesetzt werden. Folgende Parameter müssen unterhalb der Node mit gültigen Werten versorgt werden:

Parameter	Datatyp	Wertebereich
Units	int	1
DataBlock	int	0-32767
Offset	int	0-32767
MaxDataBytes	int	0-32767
EOT	string	10 Zeichen

Auszug "Device" aus einer Konfigurationsdatei (TDIIdentCfg.xml) mit PLC-Anbindung:

```
<Device Type="1">
        <Units>1</Units>
        <Data Unit="1">
              <DataDB>9915</DataDB>
              <Offset>0</Offset>
              <MaxData>1000</MaxData>
              <EOT>0*2F2F</EOT>
        </Data>
</Device>
```

13.2.4 Datenkonvertierung

13.2.4.1 Übersicht

Auf dem Codeträger liegen die Werkzeugdaten als Abfolge von Bytes vor. Das PLC-Anwenderprogramm schreibt die Codeträgerdaten in den zuvor definierten Datenbaustein (TDIIdentcfg.xml). Die Datenkonvertierung greift auf den Datenbaustein zu und erzeugt den Werkzeugdatensatz und umgekehrt.

Nachfolgendes Bild stellt das Schema zur Projektierung der Konvertierungsvorschriften für Tool Ident Connection dar:

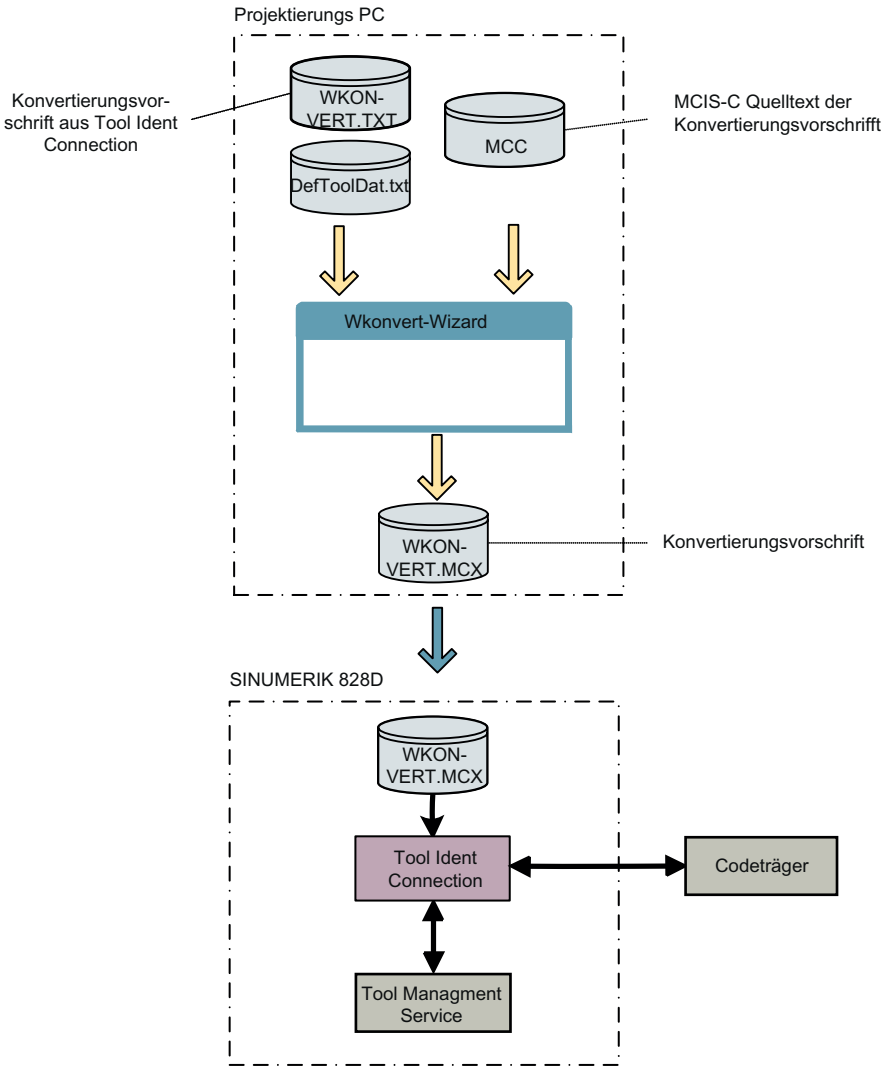


Bild 13-11 Projektierung der Konvertierungsvorschriften

Konvertierungsvorschrift erstellen

Als Teil der Inbetriebnahme von Tool Ident Connection muss auf einen Projektierungsrechner mit der Applikation Wkonvert-Wizard eine Konvertierungsdatei angelegt werden. Dabei wird die Binärdatei wkonvert.mcx erstellt Diese beinhaltet die Konvertierungsvorschrift und Defaultwerkzeugdaten.

Folgende Konvertierungsvorschriften können verarbeitet werden:

- Konvertierungsvorschrift aus Tool IdentConnection
  Dateien: wkonvert.txt und DefToolDat.txt
  Nutzen: Gewährleistung der Kompatibilität mit früheren Versionen von Tool Ident Connection.

  Hinweis

  Der Aufbau der Datei wkonvert.txt wird in Kapitel Aufbau der Beschreibungsdatei wkonvert.txt (Seite 383) beschrieben.

  Der Aufbau der Datei DefToolDat.txt wird in Kapitel Werkzeugdaten-Initialisierungsdatei DefToolDat.txt (Seite 391) behandelt.

- Konvertierungsvorschrift in Form von Quelltext in MCIS-C Skriptsprache.
  Nutzen: erweiterte Funktionalitäten zur Konvertierung.

  Hinweis

  Falls Sie eine kundenspezifische Konvertierungsvorschrift für Tool Ident Connection durch Siemens erstellen lassen wollen, kontaktieren Sie bitte Ihren Siemens-Vertriebspartner.

13.2.4.2 Anwenden des Wkonvert-Wizard

Nach der Installation kann der Wkonvert-Wizard über

Start > Programme > MCIS > TDI > TDI Wkonvert-Wizard

gestartet werden.

Nach dem Ausführen der Applikation erscheint die Grundmaske dieser Applikation:

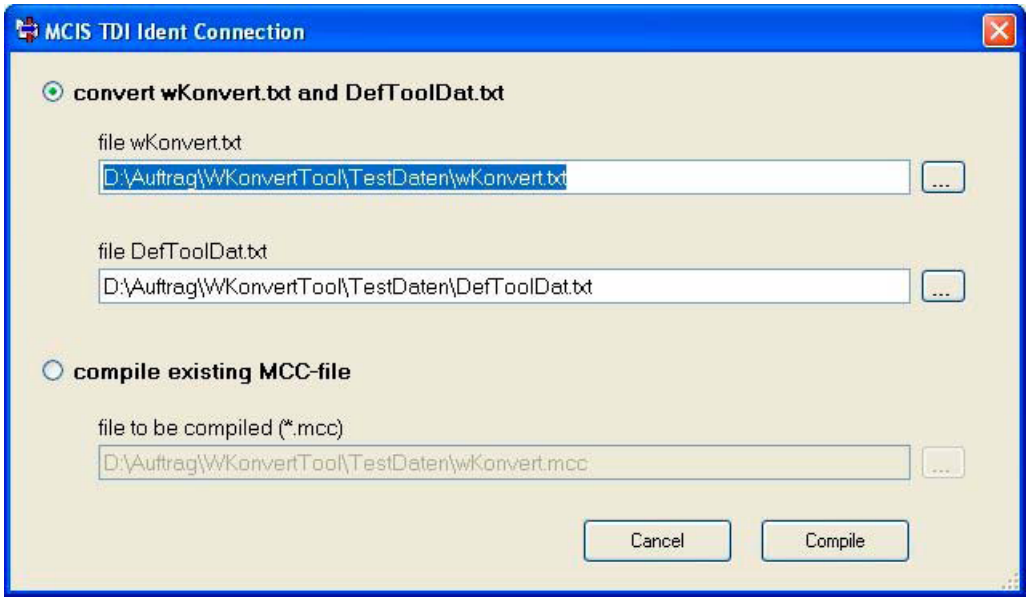


Bild 13-12 Wkonvert-Wizard Maske

Im der Grundmaske der Applikation kann zwischen folgenden Alternativen ausgewählt werden:

- "convert wKonvert.txt and DefToolDat.txt"
  Konvertierungsvorschrift aus Tool IdentConnection (Dateien: wkonvert.txt und DefToolDat.txt) erstellen.
- "compile existing MCC-file"
  Konvertierungsvorschrift in Form vom Quelltext in MCIS-C Skriptsprache erstellen.

Das Zielverzeichnis zur Ausgabe der Ergebnisdatei wkonvert.mcx entspricht dem Verzeichnis, in dem sich die Datei wkonvert.txt befindet.

Hinweis

Nach dem Erstellen der Konvertierungsvorschrift muss die mcx-Datei in das Verzeichnis: "/addon/sinumerik/appl/" kopiert werden. Damit die neue Konvertierungsvorschrift wirksam wird, ist ein Neustart der Bedien-Software erforderlich.

13.2.4.3 Meldungen des Wkonvert-Wizard

In diesem Abschnitt werden Meldungen der Applikation Wkonvert-Wizard beschrieben, die beim Interpretieren der wkonvert.txt und DefToolDat.txt oder der MCIS-C Scripte auftreten können.

Sämtliche Meldungen werden im Abschlussdialog angezeigt.



Bild 13-13 Abschlussdialog

Bei Fehlermeldungen werden die Zeile sowie die Quell-Datei, in welcher der Fehler aufgetreten ist, sowie weitergehende Informationen zum Fehlerbild in einer scrollbaren Textbox angezeigt. Zur Fehlerbehebung muss der aufgetretene Fehler in der Quelldatei behoben werden.

Nachfolgende Tabelle listet mögliche Fehlermeldungen auf:

ErrorCode	ErrorMessage	Bedeutung
200	File <abcd> could not be read!	Datei: DefToolDat.txt; Datei kann nicht gelesen werden. Es sind weitere Informationen angegeben.
201	Unknown data type.	Datei: DefToolDat.txt; einem Item ist kein Wert zugewiesen.
202	Line has no value.	Datei: DefToolDat.txt; "=" fehlt.
203	Invalid line format.	Datei: DefToolDat.txt; "[" oder "]" fehlt.
100	File <abcd> could not be read!	Datei: wKonvert.txt; Datei kann nicht gelesen werden. Es sind weitere Informationen angegeben

ErrorCode	ErrorMessage	Bedeutung
101	Error in value ‚DataLen'	Datei: wKonvert.txt; Fehler bei der Angabe des Wertes ‚DataLen'.
102	New block before BlockEnd	Datei: wKonvert.txt; Kennung BlockEnd fehlt.
103	Missing block numer <xx>	Datei: wKonvert.txt; Block <xx> fehlt.
104	Line is empty	Datei: wKonvert.txt; die angegebene Zeile ist leer.
105	No item number defined	Datei: wKonvert.txt; das Item hat keine Nummer.
110	Missing item: act=<x>, last=<y>.	Datei: wKonvert.txt; Nummerierung der Items ist falsch.
111	Finishing block number <xx>	Datei: wKonvert.txt; Blocknummer ist falsch.

Nach einem erfolgreichen Übersetzungsvorgang der Quelldatei(en) wird nachfolgende Meldung ausgegeben:



Bild 13-14 MessageBox

13.2.5 Konvertierungsdateien wkonvert.txt und DefToolDat.txt

Für Tool Ident Connection werden die Konvertierungsvorschriften für die Dateien Wkonvert.txt und DefToolDat.txt verwendet. Zusätzlich gibt es die Codeträger-Datenformate: REAL und DOUBLE.

13.2.5.1 Aufbau der Beschreibungsdatei wkonvert.txt

Die Daten auf dem Codeträger sind in einer bestimmten Reihenfolge hinterlegt. Damit die Werkzeugverwaltung diesen Datenstrom lesen bzw. schreiben kann, gibt es eine Konvertierungsvorschrift in Form einer Beschreibungsdatei.

Diese Beschreibungsdatei besteht aus genau festgelegten Werkzeug- und Schneiden-Dialogdaten. Nur diese Dialogdaten können von der Werkzeugverwaltung auch tatsächlich verarbeitet werden. Alle anderen Daten auf dem Codeträger dürfen keiner Dialogvariablen zugeordnet werden; sie werden dann von der Standardversion nicht verarbeitet.

Tool Ident Connectiol bietet die Möglichkeit, optionale Software-Komponenten anzubinden. Diese Komponenten realisieren die Anpassung dieser Codeträger-Sonderdaten an den SINUMERIK-Standard. So können sowohl kundenspezifische Codeträger-Formate von der Werkzeugverwaltung gelesen und geschrieben werden, als auch eine besondere Logik in den Transportabläufen der WZ-Daten realisiert werden (z.B. Berücksichtigung der Schwesterwerkzeuge in der NC, Anlegen von zusätzlichen Schneiden, u.s.w.).

Die Beschreibungsdatei kann als ASCII-Datei mit einem Standard-Texteditor erstellt werden. Diese Beschreibungsdatei heißt wkonvert.txt. Sie stellt das exakte Abbild der Daten dar, die sich auf dem Codeträger befinden.

Werkzeug–Dialogdaten

Die Werkzeug-Dialogdaten sind folgendermaßen definiert:

Tabelle 13-1 Werkzeug-Dialogdaten

Dialog–Variable	Datentyp	Bezeichnung	Zuordnung $TC...
T1	String	Werkzeugname, max. 32 Stellen	$TC_TP2
T2	Integer	Duplonummer	$TC_TP1
T3	Integer	Anzahl Schneiden	$P_TOOLND[tnr] tnr=Werkzeugnummer
T4	Integer	Werkzeuggröße links in Halbplätzen	$TC_TP3
T5	Integer	Werkzeuggröße rechts in Halbplätzen	$TC_TP4
T6	Integer	Werkzeuggröße oben in Halbplätzen	$TC_TP5
T7	Integer	Werkzeuggröße unten in Halbplätzen	$TC_TP6
T8	String	Magazinplatztyp	$TC_TP7 [1)]
T9	Integer	Werkzeugstatus	$TC_TP8
T10	Integer	Art der Werkzeug–Überwachung	$TC_TP9
T11	Integer	Art der WZ–Suche	$TC_TP11
T12 [2)]	Integer	Magazinplatztyp	$TC_TP7

1) Der Zeichenstring, der dort abgelegt wird, ist ein steuerungsinterner Platztyp, der dem Wert in $TC_TP7 zugeordnet ist. Dieser Text wird über Inbetriebnahme WZV festgelegt und ist in der Datenbank abgelegt.

2) Diese Zahl entspricht dem String aus T8 und ist alternativ zu T8 zu verwenden.

Schneiden-Dialogdaten

Tabelle 13-2 Schneiden-Dialogdaten

Dialog–Variable	Datentyp	Bezeichnung	Zuordnung $TC...
C1	Integer	Subtyp	$TC_DP1
C4	Integer	Schneidenlage	$TC_DP2
		Geometrie-Werkzeuglängenkorrektur	
C5	Double	Länge 1	$TC_DP3
C6	Double	Länge 2	$TC_DP4
C7	Double	Länge 3	$TC_DP5
		Geometrie-Werkzeugradiuskorrektur	
C8	Double	Länge 1	$TC_DP8
C9	Double	Länge 2	$TC_DP9
C10	Double	Radius 1	$TC_DP6
C11	Double	Radius 2	$TC_DP7
C12	Double	Winkel 1	$TC_DP10

Dialog–Variable	Datentyp	Bezeichnung	Zuordnung $TC...
C13	Double	Winkel 2	$TC_DP11
		Verschleiß-Werkzeuglängenkorrektur	
C14	Double	Länge 1	$TC_DP12
C15	Double	Länge 2	$TC_DP13
C16	Double	Länge 3	$TC_DP14
		Verschleiß-Werkzeugradiuskorrektur	
C17	Double	Länge 1	$TC_DP17
C18	Double	Länge 2	$TC_DP18
C19	Double	Radius 1	$TC_DP15
C20	Double	Radius 2	$TC_DP16
C21	Double	Winkel 1	$TC_DP19
C22	Double	Winkel 2	$TC_DP20
		Basis-/Adaptermaß-Werkzeuglängenkorrektur	
C23	Double	Basis-Länge 1	$TC_DP21
C24	Double	Basis-Länge 2	$TC_DP22
C25	Double	Basis-Länge 3	$TC_DP23
C26	Double	Freischneidwinkel	$TC_DP24
C27	Integer	Überkopfeinsatz	$TC_DP25
C28	Integer	Schneidennummer - für Adressierung der Variablen	
C29 *)	Integer	Standzeit in Minuten	$TC_MOP2
C30 *)	Integer	Vorwarngrenze Standzeit in Minuten	$TC_MOP1
C31 *)	Integer	Stückzahl	$TC_MOP4
C32 *)	Integer	Vorwarngrenze Stückzahl	$TC_MOP3
C33	Double	Soll Standzeit in Minuten	$TC_MOP11
C34 *)	Integer	Soll-Stückzahl	$TC_MOP13
C35	Double	Vorwarngrenze Verschleiß	$TC_MOP5
C36	Double	Verschleiß	$TC_MOP6
C37	Double	Soll-Verschleiß	$TC_MOP15
C38 *)	Double	Standzeit in Minuten	$TC_MOP2
C39 *)	Double	Vorwarngrenze Standzeit in Minuten	$TC_MOP1
C40 *)	Double	Stückzahl	$TC_MOP4
C41 *)	Double	Vorwarngrenze Stückzahl	$TC_MOP3
C42 *)	Double	Soll-Stückzahl	$TC_MOP13
A1 - A10	Double/String	Anwender-Werkzeugdaten	$TC_TPCx
U1 - U10	Double/String	Anwender-Schneidendaten	$TC_DPCx
S1 - S10	Double/String	Anwender-Überwachungsdaten	$TC_MOPCx

*) C38 und C39 können nur alternativ zu C29 und C30 verwendet werden; C40, C41 und C42 können nur alternativ zu C31, C32 und C34 verwendet werden.

Diejenigen Daten auf dem Codeträger, die für die Werkzeugverwaltung nicht relevant sind, aber beim Schreiben auf den Codeträger nicht verändert werden dürfen, werden in der Beschreibungsdatei mit dem Strich (-) anstelle der Dialogvariable bezeichnet (Datenlücken).

Datentypen

Die Datentypen der Dialogvariablen sind festlegt:

- Integer: Wertebereich -32768 bis +32767
- Double: Fließkomma doppelt genau
- String: Zeichenfolge aus ASCII–Zeichen

Schlüsselworte

Die Codeträger-Beschreibungsdatei ist zeilenweise aufgebaut, wobei jede Zeile durch eines der folgenden Schlüsselworte eingeleitet wird:

- Hochkomma
  Das Hochkomma (') ist ein Kommentarbeginn. Die nachfolgenden Zeichen bis Zeilenende werden überlesen. Beispiel:
  ```
  ' Dies ist ein Kommentar
  ```
- Datalen
  `DATALEN=CONST | VARIABLE 0x<trennzeichen>`
  Die nachfolgenden Daten haben konstante (CONST) oder variable (VARIABLE) Datenlänge. Daten mit variabler Länge werden mit 0x<trennzeichen> abgeschlossen. Beispiel:
  ```
  DATALEN=VARIABLE 0x0A ' variable Datenlaenge, Trennzeichen LF
  ```
- Item
  `Item<n>=<line>`
  `<n>` := laufende Nummer des Codeträgerdatums, lückenlos aufsteigend ab 1
  `<line>` := <(max.) Länge in Bytes> `<codeträgerdatenformat> <dialogvariable>`
  `<codeträgerdatenformat>` : siehe Tabelle Codeträger-Datenformate
  `<dialogvariable>` : Zuordnung zur Dialogvariable, siehe Tabelle Schneiden-Dialogdaten
  Beispiel:
  ```
  Item1 32 ASCII T3 ' Werkzeug-Bezeichner
  Item2 3 BCD T2 ' Duplonummer
  ```
- BItem
  `BItem<n>=<line>`
  `<n>` := laufende Nummer des Codeträgerdatums, lückenlos aufsteigend ab 1
  `<line>` := analog `Item<n>`
  Konvertierungsvorschrift für Codeträgerdatum <n> innerhalb eines Blocks. Wird dem Codeträgerdatum ein Werkzeug-Dialogdatum T<n> zugeordnet, wird dem Dialogdatum der erste Wert des Codeträgerdatums im Block zugewiesen.
  Beispiel:
  ```
  BItem1 1 BCD C1 ' Subtyp
  BItem2 1 BCD C4 ' Schneidenlage
  ```

- Block
  `Block<n> <wiederholvorschrift>`
  `<n>` := laufende Nummer des Blocks, lückenlos aufsteigend ab 1
  `<wiederholvorschrift>` := * `Item<n>`
  Es folgt ein Block von Daten `BItem<n>` (bis Schlüsselwort `End_Block<n>`), die entsprechend `<wiederholvorschrift>` auf dem Codeträger abgelegt sind/werden.
  Bei `Block <n> * Item <n>` muss Item `<n>` vor Block `<n>` definiert sein.
  Beispiel:
  ```
  Block1 * Item6 ’ Block1 entspr. dem Wert von Item6 wiederholen
  End_Block<n>
  ```
- End_Block
  `End_Block<n>`
  `<n>` := laufende Nummer des Blocks
  Endekennung für einen mit "Block" definierten Datenblock.

Codeträger-Datenformate - <codeträgerdatenformat>

Es werden folgende Codetäger-Datenformate (vergl. `Item` / `BItem`) unterstützt:

Tabelle 13-3 Codeträger–Datenformate

Datenformat	Erläuterung	Wertebereich
ASCII	ASCII-Zeichensatz	-
INT	16 Bit Integer (Intel-Format)	-32768 bis +32767
BCD	Binär codierte Dezimalzahl (ggf. mit Vorzeichen und Dezimalpunkt): nicht relevante Dekaden werden linksbündig mit 0 vorbesetzt.	-
REAL	16 Bit Gleitkommazahl (Intel-Format): Genauigkeit 7 Stellen	$\pm 1{,}5\times10^{-45}$ bis $\pm\ 3{,}4\times10^{38}$
DOUBLE	32 Bit Gleitkommazahl (Intel-Format): Genauigkeit 15-16 Stellen	$\pm 5{,}0\times10^{-324}$ bis $\pm 1{,}7\times10^{308}$
FPX2	Bit Integer (SINUMERIK): niederwertiges Byte an höherwertiger Adresse (umgekehrt wie Intel–Format)	-32768 bis +32767

Zuordnung zwischen Codeträgerdaten und Dialogdaten - <dialogvariable>

Die Konvertierungsvorschrift für `Item<n>` bzw. `BItem<n>` enthält u. a. die Zuordnung zu keiner/einer/mehrerer Dialogvariablen, ggf. mit einer Umsetzungsvorschrift, die in diesem Abschnitt näher erläutert wird.

```
<dialogvariable> :=   <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)]
                      [,<dvar3>[&<dvar4>]=(<uv>)]
                      [,dvarN>[=(<uv>)]
<dvar> :=             T<index> | C<index> | _
                      T = Werkzeugdatum (tool data),
                      C = Schneidendatum (cutting edge data)
                      index = Index innerhalb Werkzeug-/Schneiden-Dialogdaten
                      – = keine Zuordnung zu einer Dialogvariablen
<dvar1>&<dvar2>=      <uv>
                      Umsetzvorschrift gilt für <dvar1> und <dvar2>
```

`uv :=` `<Tetn>`
Tetn := n. Tetrade in Bytefolge
Byte1, = Tet1 und Tet2
Byte2, = Tet3 und Tet4

Beispiel für die Aufteilung der Tetraden der Codeträgervariablen (im BCD-Format) auf Dialogvariable:

```
T4=(Tet1), T5=(Tet2), T6=(Tet3), T7=(Tet4)
```

Hat die Codeträgervariable z. B. den Wert 0x1234, erhält die Dialogvariable T4 den Wert 1, die Dialogvariable T7 den Wert 4.

13.2.5.2 Beispiele für Beschreibungsdateien wkonvert.txt

Der Name der Datei ist nach der Installation "wkonvert.txt".

Tabelle 13-4 Wkonvert.txt Items

Codeträger-Variable	Länge (Bytes)	Daten-format	Dialog-Va-riable	Kommerntar
Item1	32	ASCII	T1	’ Bezeichner
Item2	3	BCD	T2	’ Duplo
Item3	2	BCD	T4=(Tet1), T5=(Tet2), T6=(Tet3), T7=(Tet4)	’ WZ-Groesse: links, rechts, oben, unten
Item4	32	ASCII	T8	’ Platztyp
Item5	2	BCD	T9	’ Status
Item6	1	BCD	T3	’ Anz. Schneiden
Item7	1	BCD	T10	' Art der WZ-Ueberwachung
Item8	1	BCD	T11	' Art der WZ-Ueberwachung
’ Schneidendaten Block1 * Item6				
BItem1	2	BCD	C1	’ Subtyp, Typ
BItem2	3	BCD	C4	’ Schneidenlage
' WZ-Laengen-Korrektur				
BItem3	4	BCD	C5	' Länge 1
BItem4	4	BCD	C5	' Länge 2
BItem5	4	BCD	C7	' Länge 3
' WZ-Radius-Korrektur				
BItem6	4	BCD	C8	' Länge 1
BItem7	4	BCD	C9	' Länge 2
BItem8	4	BCD	C10	' Radius 1
BItem9	4	BCD	C11	' Radius 2
BItem10	4	BCD	C12	' Winkel 1
BItem11	4	BCD	C13	' Winkel 2
' Verschleiss-Laengenkorrektur				

Codeträger-Variable	Länge (Bytes)	Daten-format	Dialog-Variable	Kommerntar
BItem12	4	BCD	C14	' Länge 1
BItem13	4	BCD	C15	' Länge 2
BItem14	4	BCD	C16	' Länge 3
' Verschleiss-Radius-Korrektur				
BItem15	4	BCD	C17	' Länge 1
BItem16	4	BCD	C18	' Länge 2
BItem17	4	BCD	C19	' Radius 1
BItem18	4	BCD	C20	' Radius 2
BItem19	4	BCD	C21	' Winkel 1
BItem20	4	BCD	C22	' Winkel 2
' Basismass-Laengenkorrektur				
BItem21	4	BCD	C23	' Basis-Länge 1
BItem22	4	BCD	C24	' Basis-Länge 2
BItem23	4	BCD	C25	' Basis-Länge 3
BItem24	4	BCD	C26	' Freischneidewinkel
BItem25	1	BCD	C27	' Ueberkopfeinsatz
BItem26	2	BCD	C29	' Standzeit in Minuten
BItem27	2	BCD	C30	' Vorwarngrenze Standzeit
BItem28	2	BCD	C33	' Soll-Standzeit in Minuten
BItem29	2	BCD	C31	' Stueckzahl
BItem30	2	BCD	C32	' Vorwarngrenze Stueckzahl
BItem31	2	BCD	C32	' Soll-Stueckzahl
End_Block1				

Beispiel mit Werkzeug- und Schneiden-OEM-Daten sowie Datenlücken:

Tabelle 13-5 OEM Daten für die Wkonvert.txt

Codeträger-Variable	Länge (Bytes)	Daten-format	Dialog-Variable	Kommerntar
Item1	32	ASCII	T1	' Bezeichner
Item2	3	BCD	T2	' Duplo
Item3	2	BCD	T4=(Tet1), T5=(Tet2), T6=(Tet3), T7=(Tet4)	' WZ-Groesse: links, rechts, oben, unten
Item4	32	ASCII	T8	' Platztyp
Item5	2	BCD	T9	' Status
Item6	1	BCD	T3	' Anz. Schneiden
Item7	1	BCD	T10	' Art der WZ-Ueberwachung
Item8	1	BCD	T11	' Art der WZ-Ueberwachung
Item9	5	ASCII	-	' Werkzeug-Datenluecke
Item10	4	BCD	A1	' Werkzeug OEM-Datum 1

Codeträger-Variable	Länge (Bytes)	Daten-format	Dialog-Variable	Kommerntar
' Schneidendaten Block1 * Item6				
BItem1	2	BCD	C1	' Subtyp, Typ
BItem2	1	BCD	C4	' Schneidenlage
' WZ-Laengen-Korrektur				
BItem3	4	BCD	C5	' Länge 1
BItem4	4	BCD	C5	' Länge 2
BItem5	4	BCD	C7	' Länge 3
' WZ-Radius-Korrektur				
BItem6	4	BCD	C8	' Länge 1
BItem7	4	BCD	C9	' Länge 2
BItem8	4	BCD	C10	' Radius 1
BItem9	4	BCD	C11	' Radius 2
BItem10	4	BCD	C12	' Winkel 1
BItem11	4	BCD	C13	' Winkel 2
' Verschleiss-Laengenkorrektur				
BItem12	4	BCD	C14	' Länge 1
BItem13	4	BCD	C15	' Länge 2
BItem14	4	BCD	C16	' Länge 3
' Verschleiss-Radius-Korrektur				
BItem15	4	BCD	C17	' Länge 1
BItem16	4	BCD	C18	' Länge 2
BItem17	4	BCD	C19	' Radius 1
BItem18	4	BCD	C20	' Radius 2
BItem19	4	BCD	C21	' Winkel 1
BItem20	4	BCD	C22	' Winkel 2
' Basismass-Laengenkorrektur				
BItem21	4	BCD	C23	' Basis-Länge 1
BItem22	4	BCD	C24	' Basis-Länge 2
BItem23	4	BCD	C25	' Basis-Länge 3
BItem24	4	BCD	C26	' Freischneidewinkel
BItem25	1	BCD	C27	' Ueberkopfeinsatz
BItem26	2	BCD	C29	' Standzeit in Minuten
BItem27	2	BCD	C30	' Vorwarngrenze Standzeit
BItem28	2	BCD	C33	' Soll-Standzeit in Minuten
BItem29	2	BCD	C31	' Stueckzahl
BItem30	2	BCD	C32	' Vorwarngrenze Stueckzahl
BItem31	2	BCD	C32	' Soll-Stueckzahl
BItem32	4	BCD	U1	' Schneiden-OEM Datum 1
BItem33	3	BCD	-	' Schneiden-Datenluecke
End_Block1				

13.2.5.3 Werkzeugdaten-Initialisierungsdatei DefToolDat.txt

Mit den in der Datei "DefToolDat.txt" enthaltenen Default-Daten werden bei den Diensten von Tool Ident Connection diejenigen Werkzeugdaten initialisiert, die in der Datenquelle nicht vorhanden sind. Diese Situation kann beim Lesen vom Codeträger aus einer Import-Datei oder vom Fertigungsleitrechner auftreten. In dieser Datei befinden sich die Daten eines Default-Werkzeugs mit einer Schneide in der bekannten NC-$-Syntax.

Der Inhalt von "DefToolDat.txt" muss beim Projektieren von Tool ident Connection vom Anwender angepasst werden. Durch das Entfernen der Kommentarzeichen vor den entsprechenden optionalen Werkzeugdaten, soll ihre Anzahl an die vorhandene NC-Konfiguration angepasst werden. Auch diesen Werkzeugdaten in der Datei zugewiesenen Werte können nach den Anforderungen des Anwenders modifiziert werden.

Beispiel für eine Beschreibungsdatei DefToolDat.txt

```
; Beispiel eines Werkzeug-Defaultdaten-Files.
; ACHTUNG: Bestimmte Datenwerte können in diesem Beispiel nicht
sinnvoll sein!
;
;Tool Data
$TC_TP1[1]=1                    ; Duplo Number
$TC_TP2[1]=1                    ; Tool Name
$TC_TP3[1]=1                    ; Tool size left
$TC_TP4[1]=1                    ; Tool size right
$TC_TP5[1]=1                    ; Tool size up
$TC_TP6[1]=1                    ; Tool size low
$TC_TP7[1]=1                    ; Mag. location type
$TC_TP8[1]=2                    ; Tool Status
$TC_TP9[1]=0                    ; Watchdog type
$TC_TP10[1=0                    ;
;
;OEM Tool Data
;$TC_TPC1[1]=1.0
;$TC_TPC2[1]=2.0
;$TC_TPC3[1]=3.0
;$TC_TPC4[1]=4.0
;$TC_TPC5[1]=5.0
;$TC_TPC6[1]=6.0
;$TC_TPC7[1]=7.0
;$TC_TPC8[1]=8.0
;$TC_TPC9[1]=9.0
;$TC_TPC10[1]=10.0
;
;Cutting Edge Data
```

```
$TC_DP1[1,1]=120              ; Tool Sub-Type
$TC_DP2[1,1]=0                ; Cutting Edge Orientation
$TC_DP3[1,1]=0.0              ; Geom Length 1
$TC_DP4[1,1]=0.0              ; Geom Length 2
$TC_DP5[1,1]=0.0              ; Geom Length 3
$TC_DP6[1,1]=0.0              ; Geom Radius 1
$TC_DP7[1,1]=0.0              ; Geom Radius 2
$TC_DP8[1,1]=0.0              ; Geom-Radius Length 1
$TC_DP9[1,1]=0.0              ; Geom-Radius Length 2
$TC_DP10[1,1]=0.0             ; Geom-Radius Angle 1
$TC_DP11[1,1]=0.0             ; Geom-Radius Angle 2
$TC_DP12[1,1]=0.0             ; Wear Length 1
$TC_DP13[1,1]=0.0             ; Wear Length 2
$TC_DP14[1,1]=0.0             ; Wear Length 3
$TC_DP15[1,1]=0.0             ; Wear Radius 1
$TC_DP16[1,1]=0.0             ; Wear Radius 2
$TC_DP17[1,1]=0.0             ; Wear-Radius Length 1
$TC_DP18[1,1]=0.0             ; Wear-Radius Length 2
$TC_DP19[1,1]=0.0             ; Wear-Radius Angle 1
$TC_DP20[1,1]=0.0             ; Wear-Radius Angle 2
$TC_DP21[1,1]=0.0             ; Base Length 1
$TC_DP22[1,1]=0.0             ; Base Length 2
$TC_DP23[1,1]=0.0             ; Base Length 3
$TC_DP24[1,1]=0.0             ; Tool clearance angle
$TC_DP25[1,1]=0               ; Use of tool reversed
;
;OEM Cutting Edge Data
;$TC_DPC1[1,1]=0.0
;$TC_DPC2[1,1]=20.0
;$TC_DPC3[1,1]=30.0
;$TC_DPC4[1,1]=40.0
;$TC_DPC5[1,1]=50.0
;$TC_DPC6[1,1]=60.0
;$TC_DPC7[1,1]=70.0
;$TC_DPC8[1,1]=80.0
;$TC_DPC9[1,1]=90.0
;$TC_DPC10[1,1]=100.0
;
;Edge monitoring Data
$TC_MOP1[1,1]=0.0             ; Prewarning limit [min]
$TC_MOP2[1,1]=0.0             ; DownTime [min]
$TC_MOP3[1,1]=0               ; Prewarning limit [part]
$TC_MOP4[1,1]=0               ; Part countdown
$TC_MOP5[1,1]=0               ; Wear prewarning limit
$TC_MOP6[1,1]=0               ; Wear
$TC_MOP11[1,1]=0              ; Target service life [min]
$TC_MOP13[1,1]=0              ; Target number of parts
$TC_MOP15[1,1]=0              ; Target wear
```

```
;OEM Edge monitoring Data
;$TC_MOPC1[1,1]=0
;$TC_MOPC2[1,1]=0
;$TC_MOPC3[1,1]=0
;$TC_MOPC4[1,1]=0
;$TC_MOPC5[1,1]=0
;$TC_MOPC6[1,1]=0
;$TC_MOPC7[1,1]=0
;$TC_MOPC8[1,1]=0
;$TC_MOPC9[1,1]=0
;$TC_MOPC10[1,1]=0
;
;Ortsabhängige Schneiden-Summenkorrekturen
;$TC_SCP13[1,1]=0.0
;$TC_SCP14[1,1]=0.0
;$TC_SCP15[1,1]=0.0
;$TC_SCP16[1,1]=0.0
;$TC_SCP17[1,1]=0.0
;$TC_SCP18[1,1]=0.0
;$TC_SCP19[1,1]=0.0
;$TC_SCP20[1,1]=0.0
;$TC_SCP21[1,1]=0.0
M17
```

13.2.6 Konvertierungsvorschrift auf Basis der MCIS-C Skriptsprache

Tool Ident Connection bietet die Möglichkeit auf Basis der MCIS-C Skriptsprache Konvertierungsvorschiften für die Codeträger zu erstellen. Durch die Skriptsprache erhält der Anwender zusätzliche Funktionen zum Interpretieren der Daten auf dem Codeträger.

Durch die Verwendung von Kontrollstrukturen (z. B. if) in Verbindung mit Vergleichsoperatoren (z. B.: >, <) können auch komplexere Codeträger-Formate verarbeitet werden.

13.2.6.1 Aufbau der Konvertierungsvorschrift

Die Konvertierungsvorschrift wird als ASCII-Datei mit der Erweiterung MCC (MCIS-C) erstellt. Die Datei hat folgenden Aufbau:

```
Zeile        Programmcode
1            _VersionInfo := "Sample for WKONVERT-Logic";
2
3            TmTool Tool = new TmTool(); // Tool-Object
4            unsigned char ToolArray[0]; // Array-Object
5
6            void DecodeToolData()
7            {
8                        Tool.TC_TP2 = readASCII( ToolArray, 2, 10 );
9                        ...
10           }
11
```

```
12          void EncodeToolData()
13          {
14                    writeASCII( ToolArray, 2, 10, Tool.TC_TP2 );
15                    ...
16          }
```

Zeile 1: Mit dem Konstrukt _VersionInfo kann der Anwender innerhalb der Konvertierungsvorschrift eigene Versions-Information hinterlegen.

Zeile 3: In dieser Zeile wird ein TmTool-Objekt mit dem Namen Tool definiert. Unter Verwendung dieses Objektes kann auf die einzelnen Werkzeug-/Schneidendaten eines Werkzeugs zugegriffen werden.

Zeile 4: In dieser Zeile wird ein Array-Objekt mit dem Namen ToolArray definiert. Unter Verwendung dieses Objektes kann auf die Daten auf dem Codeträger zugegriffen werden (lesend bzw. schreibend). Die Größe des Datenträgers wird zunächst mit 0 vorbelegt. Die tatsächliche Größe des Codeträgers wird zur Laufzeit ermittelt, und kann mit dem Konstrukt: ToolArray.ArraySize() gelesen werden. Die Zugriffe auf die einzelnen Bytes des Codeträgers erfolgen mit dem Konstrukt: ToolArray[offset], wobei der Wert offset die Werte von 0 bis ToolArray.ArraySize()-1 annehmen darf.

Außer einem direkten Zugriff auf die einzelnen Bytes des Codeträgers können zusätzliche Hilfsfunktionen verwendet werden. Diese Hilfsfunktionen sind immer dann notwendig, wenn Daten in einer bestimmter Kodierung gelesen bzw. geschrieben werden sollen (z. B. readBCD() / writeBCD()).

Zeile 6: Die Funktion mit dem Namen DecodeToolData() wird immer dann vom Tool Ident Connection aufgerufen, wenn Daten vom Codeträger gelesen werden. Zum Zeitpunkt des Aufrufs dieser Funktion enthält das Array-Objekt: ToolArray, wo die gesamten Daten des Codeträgers hinterlegt sind. Das Werkzeug-Objekt: Tool enthält zunächst keine Daten und wird innerhalb dieser Funktionen mit Daten "gefüllt".

Zeile 8: Diese Zeile stellt exemplarisch das Vorgehen innerhalb der DecodeToolData()-Funktion dar. Es wird hier der Werkzeugbezeichner - unter Verwendung der Hilfsfunktion readASCII() – aus den Daten des Codeträgers gelesen und dem Datenelement TC_TP2 des Werkzeugdatensatzes zugewiesen.

Zeile 12: Die Funktion mit dem Namen EncodeToolData() wird immer dann vom Tool Ident Connection aufgerufen, wenn Daten auf den Codeträger geschrieben werden. Zum Zeitpunkt des Aufrufs dieser Funktion enthält das Array-Objekt: ToolArray die aktuellen Daten aus dem Codeträger (vor jedem Schreiben auf Codeträger wird dieser zunächst gelesen). Das Tool-Objekt enthält Daten des Werkzeugs, die auf den Codeträger geschrieben werden sollen.

Zeile 14 Diese Zeile stellt exemplarisch das Vorgehen innerhalb der EncodeToolData()-Funktion dar. Es wird hier der Werkzeugbezeichner TC_TP2 - unter Verwendung der Hilfsfunktion writeASCII() – auf den Codeträger geschrieben. Diese Zeile stellt exemplarisch das Vorgehen innerhalb der EncodeToolData()-Funktion dar. Es wird hier der Werkzeugbezeichner TC_TP2 - unter Verwendung der Hilfsfunktion writeASCII() – auf den Codeträger geschrieben.

Die verwendeten Bezeichner: Tool, ToolArray, DecodeToolData, EncodeToolData dürfen **nicht** verändert werden.

13.2.6.2 Zugriff auf Werkzeugdaten

Innerhalb der Konvertierungsvorschrift kann über das Tool-Objekt lesend und schreibend auf einzelne Werkzeugdaten zugegriffen werden. In diesem Abschnitt werden alle zur Verfügung stehenden Datenelemente aufgeführt. Die hierbei verwendeten Bezeichner orientieren sich an die der Werkzeugverwaltung (z. B. Tool.TC_TP2 - Werkzeugbezeichner). Die Bedeutung der einzelnen Werkzeugdaten ist der Beschreibung der Werkzeugverwaltung im Kapitel "NC-Programmierung (Seite 105)" zu entnehmen. Der Zugriff auf die schneidenbezogenen Daten erfolgt unter der Angabe der Schneidennummer (z. B. Tool.TC_DP2[2] – Schneidenlage der Schneide mit der Schneidennummer 2).

Bezeichner	Bezeichnung	Datentyp MCIC-C
Werkzeugbezogene Daten		
TC_TP1	Duplo-Nummer	long
TC_TP2	Werkzeugbezeichner	string
TC_TP3	Größe nach links	long
TC_TP4	Größe nach rechts	long
TC_TP5	Größe nach oben	long
TC_TP6	Größe nach unten	long
TC_TP7	Magazinplatztyp	long
TC_TP8	Status	long
TC_TP9	Art der Werkzeugüberwachung	long
TC_TP10	Ersatzwechsel-Strategie	long
TC_TP11	Werkzeuginfo	long
TC_TP_PROTA	Name für den Schutzbereich	string
TC_TP_MAX_VELO	Maximale Drehzahl des Werkzeugs	double
TC_TP_MAX_ACC	Maximale Beschleunigung des Werkzeugs	double
Werkzeugbezogene Anwenderdaten		
TC_TPC1	OEM-Anwenderdaten 1	double
...	...	...
TC_TPC64	OEM-Anwenderdaten 64 (als Double)	double
TC_TPC1Str	OEM-Anwenderdaten 1 (als String)	string
...	...	...
TC_TPC64Str	OEM-Anwenderdaten 64 (als String)	double
Schneidenparameter		
TC_DP1 [EdgeNo]	Werkzeugtyp	long
TC_DP2 [EdgeNo]	Schneidenlage	double
TC_DP3 [EdgeNo]	Geometrie Länge 1	double
TC_DP4 [EdgeNo]	Geometrie Länge 2	double
TC_DP5 [EdgeNo]	Geometrie Länge 3	double
TC_DP6 [EdgeNo]	Geometrie Radius	double
TC_DP7 [EdgeNo]	Geometrie - Eckenradius	double
TC_DP8 [EdgeNo]	Geometrie Länge 4	double
TC_DP9 [EdgeNo]	Geometrie Länge 5	double
TC_DP10 [EdgeNo]	Geometrie - Winkel 1	double

Bezeichner	Bezeichnung	Datentyp MCIC-C
TC_DP11 [EdgeNo]	Geometrie - Winkel 2	double
TC_DP12 [EdgeNo]	Verschleiß - Länge 1	double
TC_DP13 [EdgeNo]	Verschleiß - Länge 2	double
TC_DP14 [EdgeNo]	Verschleiß - Länge 3	double
TC_DP15 [EdgeNo]	Verschleiß - Radius	double
TC_DP16 [EdgeNo]	Verschleiß - Verrundungsradius	double
TC_DP17 [EdgeNo]	Verschleiß - Überstand	double
TC_DP18 [EdgeNo]	Verschleiß - Länge 5	double
TC_DP19 [EdgeNo]	Verschleiß - Winkel 1	double
TC_DP20 [EdgeNo]	Verschleiß - Winkel 2	double
TC_DP22 [EdgeNo]	Adapter - Länge 1	double
TC_DP23 [EdgeNo]	Adapter - Länge 2	double
TC_DP24 [EdgeNo]	Adapter - Länge 3	double
TC_DP25 [EdgeNo]	Freiwinkel	double
TC_DPCE [EdgeNo]	Schneidennummer CE	long
TC_DPH [EdgeNo]	H-Parameter	double
TC_DPV [EdgeNo]	Werkzeugschneidenorientierung	double
TC_DPV3 [EdgeNo]	L1-Komponente der Werkzeugschneidenorientierung	double
TC_DPV4 [EdgeNo]	L2-Komponente der Werkzeugschneidenorientierung	double
TC_DPV5 [EdgeNo]	L3-Komponente der Werkzeugschneidenorientierung	double
TC_DPVN3	L1-Komponente der Orientierungsnormalen	double
TC_DPVN4	L2-Komponente der Orientierungsnormalen	double
TC_DPVN5	L3-Komponente der Orientierungsnormalen	double
TC_DPNT	Anzahl der Zähne dieser Schneide	long
TC_DPROT	Grunddrehwinkel der Schneide Hinweis: Für die Zugriffe auf die Variable TC_DPROT wird der TDI-WKonvert-Assistent ab Version 02.03.00.03	double
Schneiden-Zusatzinformationen		
TC_DPC1 [EdgeNo]	OEM-Schneidenanwenderdaten1	double
...	...	...
TC_DPC10 [EdgeNo]	OEM-Schneidenanwenderdaten10	double
Schneidenbezogene Werkzeugüberwachung		
TC_MOP1 [EdgeNo]	Vorwarngrenze Standzeit	double
TC_MOP2 [EdgeNo]	Rest-Standzeit	double
TC_MOP3 [EdgeNo]	Vorwarngrenze Stückzahl	long
TC_MOP4 [EdgeNo]	Rest-Stückzahl	long
TC_MOP11 [EdgeNo]	Standzeit Sollwert	double
TC_MOP13 [EdgeNo]	Stückzahl Sollwert	long
TC_MOP5 [EdgeNo]	Verschleißvorwarngrenze	double
TC_MOP6 [EdgeNo]	Verschleißistwert	double
TC_MOP15 [EdgeNo]	Verschleißsollwert	double
Anwender-Schneidenüberwachung		

Bezeichner	Bezeichnung	Datentyp MCIC-C
TC_MOPC1	OEM-Anwender-Schneidenüberwachung[1]	long
...		...
TC_MOPC10	OEM-Anwender-Schneidenüberwachung[10]	long
Hilfsattribute		
EDGE_NUM	Anzahl Schneiden (Read-Only	long
EDGE_NO [EdgeIdx]	Schneidennummer (Read-Only) Liefert die Schneidennummer einer Schneide anhand des Schneiden-Index. Der Schneiden-Index ist ein Wert zwischen 1 und Tool.EDGE_NUM . Beispiel: Werkzeug mit drei Schneiden:1, 3, 5 Tool.EDGE_NUM liefert 3 Tool.EDGE_NO[1] liefert 1 Tool.EDGE_NO[2] liefert 3 Tool.EDGE_NO[3] liefert 5 Tool.TC_DP2[5] Schneidenlage der Schneide mit der Schneidennummer 5.	einzelne

13.2.6.3 Hilfsfunktionen für Zahlencodierungen

Die Werkzeugdaten auf dem Codeträger können unter Verwendung von verschiedenen Kodierungen abgelegt sein. Beim Lesen und Schreiben dieser Daten innerhalb von MCIS-C stehen folgende Hilfsfunktionen zur Verfügung:

Hinweis

Syntax

Alle diese Funktionen haben immer als ersten Parameter, das Array-Objekt: array und einen Offset innerhalb dieses Arrays als zweiten Parameter. Der Offset kann die Werte zwischen 0 und ToolArray.ArraySize()-1 annehmen. Bei Datentypen mit einer variablen Länge wird zusätzlich die Länge der Daten auf dem Codeträger übergeben, (z. B. readASCII(array, offset, length)).

readASCII() / writeASCII()

```
string      readASCII ( array, offset, length )
            writeASCII ( array, offset, length, string [,precison] )
```

Die Funktionen erlauben das Schreiben und Lesen von ASCII-Strings. Beim Schreiben wird der String "rechtsbündig" ausgerichtet. Ist der String, der geschrieben wird, länger als der Parameter length, dann werden überzählige Zeichen abgeschnitten. Ist der String kürzer, dann wird der geschriebene Bereich mit Leerzeichen aufgefüllt. Beim Lesen werden führende und abschließende Leerzeichen ignoriert. Die Funktion writeASCII() enthält einen optionalen Parameter "precison", mit dem die Anzahl der Nachkommastellen vorgegeben werden kann. Wird dieser Parameter nicht verwendet, dann wird die in der SINUMERIK projektierte Nachkommenstellenanzahl verwendet.

Beispiele:

```
writeASCII ( ToolArray, 0, 5, "ABC" );
readASCII ( ToolArray, 0, 5 ); => "ABC"
```

Hex value	41	42	43	20	20											
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
writeASCII ( ToolArray, 0, 5, "ABCDEFGH" );
readASCII ( ToolArray, 0, 5 ); => "ABCDE"
```

Hex value	41	42	43	44	45											
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
writeASCII ( ToolArray, 0, 5, 3.94 );
writeASCII ( ToolArray, 5, 5, 3.94, 0 );
writeASCII ( ToolArray, 10, 5, 3.94, 1 );
readASCII ( ToolArray, 0, 5 ); => "3.94"
readASCII ( ToolArray, 5, 5 ); => "00004"
readASCII ( ToolArray, 10, 5 ); => "003.9"
```

Hex value	30	33	2E	39	34	30	30	30	30	34	30	30	33	2E	39	
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readBCD() / writeBCD()

```
double      readBCD ( array, offset, length )
            writeBCD ( array, offset, length, val [,precison] )
```

Die Funktionen erlauben das Schreiben und Lesen von BCD Zahlen.

Beispiele:

```
writeBCD ( ToolArray, 0, 5, 3.11 );
readBCD ( ToolArray, 0, 5 ); => "3.11"
writeBCD ( ToolArray, 5, 5, -47.11 );
readBCD ( ToolArray, 0, 5 ); => "-47.11"
writeBCD ( ToolArray, 10, 5, 4711 );
readBCD ( ToolArray, 10, 5 ); => "4711"
```

Hex value	B0	00	00	3E	11	D0	00	04	7E	11	B0	00	00	47	11	
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
writeBCD ( ToolArray, 0, 5, -3.1234 );
readBCD ( ToolArray, 0, 5 ); => "-3.12"
writeBCD ( ToolArray, 5, 5, 1.2 );
readBCD ( ToolArray, 5, 5 ); => "1.2"
```

Hex value	D0	00	00	3E	32	B0	00	04	1E	20						
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
writeBCD ( ToolArray, 0, 4, 3.94 );
readBCD ( ToolArray, 0, 4 ); => "3.94"
writeBCD ( ToolArray, 4, 4, 3.94, 0 );
readBCD ( ToolArray, 4, 4 ); => "4"
writeBCD ( ToolArray, 8, 4, 3.94, 1 );
readBCD ( ToolArray, 8, 4 ); => "3.9"
writeBCD ( ToolArray, 12, 4, 3.94, 2 );
readBCD ( ToolArray, 12, 4 ); => "3.94"
```

Hex value	B0	00	3E	94	B0	00	00	04	B0	00	03	E9	B0	00	3E	94
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readINT() / writeINT()

```
int         readINT ( array, offset )
            writeINT ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 16-Bit Zahlen mit Vorzeichen. Die Daten werden im S7-Format abgelegt (Big-Endian).

Beispiel:

```
writeINT ( ToolArray, 0, 1 );
readINT ( ToolArray, 0, ); => "1"
writeINT ( ToolArray, 2, 4711 ); // hex: 1267
readINT ( ToolArray, 2 ); => "4711"
writeINT ( ToolArray, 4, -300 ); // hex: FED4
readINT ( ToolArray, 4 ); => "-300"
writeINT ( ToolArray, 6, -1 ); // hex: FFFF
readINT ( ToolArray, 6 ); => "-1"
```

Hex value	00	01	12	67	FE	D4	FF	FF								
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readDINT() / writeDINT()

```
long        readDINT ( array, offset )
            writeDINT ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 32-Bit Zahlen mit Vorzeichen. Die Daten werden im S7-Format abgelegt (Big-Endian).

Beispiel:

```
writeDINT ( ToolArray, 0, 1234 ); // hex: 4D2
readDINT ( ToolArray, 0, ); => "1234"
writeDINT ( ToolArray, 4, -1234 ); // hex: FFFFFB2E
readDINT ( ToolArray, 4 ); => "-1234"
```

Hex value	00	00	04	D2	FF	FF	FB	2E								
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readWORD() / writeWORD()

```
long        readWORD ( array, offset )
            writeWORD ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 16-Bit Zahlen ohne Vorzeichen. Die Daten werden im S7-Format abgelegt (Big-Endian).

Beispiel:

```
writeWORD ( ToolArray, 0, 1 ); // hex: 1
readWORD ( ToolArray, 0, ); => "1"
writeWORD ( ToolArray, 2, 50000 ); // hex: C350
readWORD ( ToolArray, 2, ); => "50000"
```

Hex value	00	01	C3	50												
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readDWORD() / writeDWORD()

```
unsigned long         readDWORD ( array, offset )
                      writeDWORD ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 32-Bit Zahlen ohne Vorzeichen. Die Daten werden im S7-Format abgelegt (Big-Endian).

Beispiel:

```
writeDWORD ( ToolArray, 0, 12345678 ); // hex: 075BCD15
readDWORD ( ToolArray, 0, ); => "12345678"
writeDWORD ( ToolArray, 4, 987654321 ); // hex: 3ADE68B1
```

```
readDWORD ( ToolArray, 4, ); => "987654321"
```

Hex value	07	5B	CD	15	3A	DE	68	B1								
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

readREAL() / writeREAL()

```
float       readREAL ( array, offset )
            writeREAL ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 32-Bit Fließkommazahlen. Die Daten werden im S7-Format abgelegt (REAL).

Beispiel:

```
writeREAL ( ToolArray, 0, 3.11f );
readREAL ( ToolArray, 0, ); => "3.11"
writeREAL ( ToolArray, 4, 08.15f );
readREAL ( ToolArray, 4, ); => "08.15"
```

Hex value	40	47	0A	3D	41	02	66	66								
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Hinweis

Die Schreibweise 3.11f sorgt dafür, dass die Konstante 3.11 als 32-Bit-Fließkommazahl (float) interpretiert wird. Als Default wird 64-Bit (double) verwendet.

readDOUBLE() / writeDOUBLE()

```
double      readDOUBLE ( array, offset )
            writeDOUBLE ( array, offset, val )
```

Die Funktionen erlauben das Schreiben und Lesen von 64-Bit Fließkommazahlen. Die Daten entsprechen dem IEEE 754-Standard.

Beispiel:

```
writeDOUBLE ( ToolArray, 0, 3.11 );
readDOUBLE ( ToolArray, 0, ); => "3.11"
writeDOUBLE ( ToolArray, 8, 08.15 );
readDOUBLE ( ToolArray, 8, ); => "08.15"
```

Hex value	40	08	E1	47	AE	14	7A	E1	40	20	4C	CC	CC	CC	CC	CD
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ReverseByteOrder ()

```
ReverseByteOrder ( val )
```

Die Funktionen xxxINT(), xxxDINT(), xxxWORD() und xxxDWORD() lesen und schreiben Werte im Big-Endian Format. Dies entspricht den S7-Formaten. Sollten Werte im Little-Endian Format gelesen/geschrieben werden, dann kann die Funktion ReverseByteOrder() hierfür verwendet werden.

Beispiel:

```
writeINT ( ToolArray, 0, 1 );
writeINT ( ToolArray, 2, ReverseByteOrder(1) );
ReverseByteOrder( readINT( ToolArray, 2 ) ); => "1"
writeDINT ( ToolArray, 4, 4711 );
writeDINT ( ToolArray, 8, ReverseByteOrder(4711L) );
ReverseByteOrder( readDINT( ToolArray, 8 ) ); => "4711"
```

Hex value	00	01	01	00	00	00	12	67	67	12	00	00				
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Hinweis

Die Schreibweise 4711L sorgt dafür, dass die Konstante 4711 als 32-Bit Konstante interpretiert wird. Als Default wird 16-Bit (int) verwendet.

13.2.6.4 Konfiguration von ToolSpec.xml

PlaceTypeToString () / PlaceTypeFromString ()

Der Platztyp eines Werkzeugs (TC_TP7) wird in SINUMERIK Operate ausschließlich als ein numerischer Wert geführt. Im HM wurde für den Platztyp ein String-Wert verwaltet. Die Zuordnung zwischen dem numerischen Wert und dem String-Wert erfolgt in der Datenbank WZACCESS.MDB.

Um die Kompatibilität von Codeträgerformaten zwischen HMI-Advanced und SINUMERIK Operate zu gewährleisten, wird in Tool Ident Connection die Konfigurationsdatei "ToolSpec.xml" eingeführt. Diese Datei muss auf der Steuerung im selben Verzeichnis wie die mcx-Datei abgelegt werden.

Beispiel 1:

```
<!--nachfolgende Datenstruktur wird dazu benutzt die PlatzTyp-
Information ($TC_TP7) einem numerischen Wert in ein String-Wert zu
überführen bzw. umgekehrt. Umsetzung von Text → Code wird der
Stringvergleich casesensitive durchgeführt-->
<ToolSpecConversion>
   <ToolSpec code="1" text="normal"/>
   <ToolSpec code="2" text="gross"/>
   <ToolSpec code="3" text="klein"/>
```

```
</ToolSpecConversion>
```

Mit den Hilfsfunktionen `PlaceTypeToString()` und `PlaceTypeFromString()` kann innerhalb der Konvertierungsvorschrift auf die ToolSpec.xml zugegriffen werden.

```
PlaceTypeToString ( NumType )
PlaceTypeFromString ( StrType )
```

Beispiel 2:

```
PlaceTypeToString ( 1 ); ⇒ "normal"
PlaceTypeToString ( 2 ); ⇒ "gross"
PlaceTypeToString ( 3 ); ⇒ "klein"

PlaceTypeFromString ("normal" ); ⇒ 1
PlaceTypeFromString ("gross" );  ⇒ 2
PlaceTypeFromString ("klein" );  ⇒ 3
```

13.2.6.5 Fehlerbehandlung

Fehler behandeln

Innerhalb einer Konvertierungsvorschrift auf Basis der MCIS-C Skriptsprache ist es möglich eine Prüfung der Werkzeugdaten durchzuführen.

Sollte die Prüfung der Werkzeugdaten negativ ausfallen (z. B.: zu großer Werkzeugdurchmesser, falscher Werkzeugtyp, etc.),

dann kann der Beladevorgang mit einer Fehlermeldung abgebrochen werden.

Die Fehlermeldung wird in dem Feld „Error_Code" in der PLC-Schnittstelle bereitgestellt.

(Siehe auch AUTOHOTSPOT)

Für den hier beschreiben Anwendungsfall kann der Anwender Error_Code aus dem Wertebereich 0x80 - 0xFF verwenden.

Siehe auch Fehlercodes (Seite 418) für die Fehlergruppe 0B: Wkonvertfehler-Gruppe im Anhang.

Um innerhalb der Konvertierungsvorschrift die hier beschriebene Fehlerbehandlung auszulösen, muss die Konvertierungsvorschrift die Deklaration der Error_Code Variable erweitert werden.

Beispiel

```
Variable                  Error_Code         = new Variable<int>();     // Error-Code
TmTool                    Tool               = new TmTool();            // Werkzeug-Objekt
unsigned char            ToolArray[0];                              // Daten des Datenträgers

void DecodeToolData()
{
  …
  int EdgeCount = readBCD( ToolArray, 71, 1 );
  if( EdgeCount > 4 )
  {
     Error_Code = 0xFF;
     return;
   }
   …
}
```

13.2.6.6 Hilfsfunktionen in Kovertierungsvorschriften

Standard C++ Typkonvertierungen

Neben den Befehlen der Skriptsprache zur Erstellung von Konvertierungsvorschriften, stehen weitere Standard C++-Befehle zur Verfügung. Folgende Befehle wandeln beispielsweise Strings in numerische Datentypen (Double) um:

- atof
- _atof_l
- _wtof
- _wtof_l

Eine detaillierte Beschreibung dieser Befehle finden Sie unter anderem in der Microsoft Knowledge Base.

13.2.7 Beispiele

Übersicht

Nachfolgende Sourcecode-Beispiele sind auf Basis der Skriptsprache MCIS-C zur Erstellung von Konvertierungsbeispielen für Tool Ident Connection. Diese Beispiele sind auf dem Datenträger vom Wkonvert-Wizard enthalten.

SampleWkonvert.mcc

```
_VersionInfo := "SampleWkonvert.mcc: Beispiel für WKONWERT-Logik";

TmTool    Tool = new TmTool();       // Werkzeug-Objekt
unsigned char ToolArray[0];          // Daten des Datenträgers

void DecodeToolData()
{

   Tool.TC_TP1   = readWORD( ToolArray, 0 );                          // Duplonummer          Byte 00-01
   Tool.TC_TP2   = readASCII( ToolArray, 2, 10);                      // Werkzeugname         Byte 02-11
   Tool.TC_TP3   = ToolArray[12];                                     // Größe nach links     Byte 12
   Tool.TC_TP4   = ToolArray[13];                                     // Größe nach rechts    Byte 13
   Tool.TC_TP5   = ToolArray[14];                                     // Größe nach oben      Byte 14
   Tool.TC_TP6   = ToolArray[15];                                     // Größe nach unten     Byte 15
   Tool.TC_TP7   = readINT(ToolArray, 16 );                           // Platz-Typ            Byte 16-17

   int EdgeCount = readINT(ToolArray, 18 );                           // Anzahl Scheiden      Byte 18-19

   int EdgeIndex;                                                     // Laufvariable für Schneiden

   for ( EdgeIndex = 1 ; EdgeIndex <= EdgeCount ; EdgeIndex++ )
   {
      int EdgeOffset    = 20 + (EdgeIndex-1) * 24;          // Offset zum Beginn der jeweiligen Schneide
                                                            // 24: Größe der Scheidendaten

      int EdgeNo             = readINT ( ToolArray, EdgeOffset +  0 );     // Schneiden-Nummer  Byte 20-21
      Tool.TC_DP1[EdgeNo]    = readINT ( ToolArray, EdgeOffset +  2 );     // Werkzeugtyp       Byte 22-23
      Tool.TC_DP2[EdgeNo]    = readREAL( ToolArray, EdgeOffset +  4 );     // Schneidenlage     Byte 24-27
      Tool.TC_DP3[EdgeNo]    = readREAL( ToolArray, EdgeOffset +  8 );     // Geometrie Länge 1 Byte 28-31
      Tool.TC_DP4[EdgeNo]    = readREAL( ToolArray, EdgeOffset + 12 );     // Geometrie Länge 2 Byte 32-35
      Tool.TC_DP5[EdgeNo]    = readREAL( ToolArray, EdgeOffset + 16 );     // Geometrie Länge 3 Byte 36-39
      Tool.TC_DP6[EdgeNo]    = readREAL( ToolArray, EdgeOffset + 20 );     // Geometrie Radius  Byte 40-43
   }
}

void EncodeToolData()
{
   writeWORD ( ToolArray, 0,     Tool.TC_TP1 );                       // Duplonummer          Byte 00-01
   writeASCII( ToolArray, 2, 10, Tool.TC_TP2 );                       // Werkzeugname         Byte 02-11

   ToolArray[12]     = Tool.TC_TP3;                                   // Größe nach links     Byte 12
   ToolArray[13]     = Tool.TC_TP4;                                   // Größe nach rechts    Byte 13
   ToolArray[14]     = Tool.TC_TP5;                                   // Größe nach oben      Byte 14
   ToolArray[15]     = Tool.TC_TP6;                                   // Größe nach unten     Byte 15

   writeINT  ( ToolArray, 16,    Tool.TC_TP7 );                       // Platz-Typ            Byte 16-17

   int EdgeCount = Tool.EDGE_NUM;

   writeINT    ( ToolArray, 18, EdgeCount );                          // Anzahl Schneiden     Byte 18-19

   int EdgeIndex;

   for ( EdgeIndex = 1 ; EdgeIndex <= EdgeCount ; EdgeIndex++ )
```

```
  {
    int EdgeNo     = Tool.EDGE_NO[EdgeIndex];                    // Schneiden-Nummer
    int EdgeOffset  = 20 + (EdgeIndex-1) * 24;                    // Offset zum Beginn der jeweiligen Schneide
                                                                  // 24: Größe der Scheidendaten

    writeINT     ( ToolArray, EdgeOffset +  0, EdgeNo );          // Schneiden-Nummer     Byte 20-21
    writeINT      ( ToolArray, EdgeOffset +  2, Tool.TC_DP1[EdgeNo] );  // Werkzeugtyp          Byte 22-23
    writeREAL   ( ToolArray, EdgeOffset +  4, Tool.TC_DP2[EdgeNo] );  // Schneidenlage        Byte 24-27
    writeREAL   ( ToolArray, EdgeOffset +  8, Tool.TC_DP3[EdgeNo] );  // Geometrie Länge 1    Byte 28-31
    writeREAL   ( ToolArray, EdgeOffset + 12, Tool.TC_DP4[EdgeNo] ); // Geometrie Länge 2    Byte 32-35
    writeREAL   ( ToolArray, EdgeOffset + 16, Tool.TC_DP5[EdgeNo] ); // Geometrie Länge 3    Byte 36-39
    writeREAL   ( ToolArray, EdgeOffset + 20, Tool.TC_DP6[EdgeNo] ); // Geometrie Radius     Byte 40-43
  }
}
```

SampleToolSize.mcc

```
_VersionInfo := "SampleToolSize.mcc: Beispiel für WKONWERT-Logik";

TmTool Tool = new TmTool();       // Werkzeug-Objekt
unsigned char ToolArray[0];       // Daten des Datenträgers

void DecodeToolData()
{

   Tool.TC_TP1   = readWORD( ToolArray, 0 );                    // Duplonummer      Byte 00-01
   Tool.TC_TP2   = readASCII( ToolArray, 2, 10);                // Werkzeugname     Byte 02-11
   Tool.TC_TP7   = readINT(ToolArray, 12 );                     // Platz-Typ        Byte 12-13

   int EdgeCount = readINT(ToolArray, 14 );                     // Anzahl Scheiden  Byte 14-15

   int EdgeIndex;                                               // Laufvariable für Schneiden
   int ToolSize = 1;                                            // Werkzeuggröße in Halbplätzen

   for ( EdgeIndex = 1 ; EdgeIndex <= EdgeCount ; EdgeIndex++ )
   {
      int EdgeOffset    = 16 + (EdgeIndex-1) * 24;              // Offset zum Beginn der jeweiligen Schneide
                                                                // 24: Größe der Scheidendaten

      int EdgeNo        = readINT ( ToolArray, EdgeOffset +  0 );  // Schneiden-Nummer Byte 16-17

      Tool.TC_DP1[EdgeNo]   = readINT ( ToolArray, EdgeOffset +  2 );  // Werkzeugtyp      Byte 18-19
      Tool.TC_DP2[EdgeNo]   = readREAL( ToolArray, EdgeOffset +  4 );  // Schneidenlage    Byte 20-23
      Tool.TC_DP3[EdgeNo]   = readREAL( ToolArray, EdgeOffset +  8 );  // Geometrie Länge 1  Byte 24-27
      Tool.TC_DP4[EdgeNo]   = readREAL( ToolArray, EdgeOffset + 12 );  // Geometrie Länge 2  Byte 28-31
      Tool.TC_DP5[EdgeNo]   = readREAL( ToolArray, EdgeOffset + 16 );  // Geometrie Länge 3  Byte 32-35
      Tool.TC_DP6[EdgeNo]   = readREAL( ToolArray, EdgeOffset + 20 );  // Geometrie Radius   Byte 36-39

      double WerkzeugRadius = Tool.TC_DP6[EdgeNo];

      if      ( WerkzeugRadius >  30.0 && ToolSize < 4 )    ToolSize = 4;
      else if ( WerkzeugRadius >= 20.0 && ToolSize < 3 )    ToolSize = 3;
      else if ( WerkzeugRadius >= 10.0 && ToolSize < 2 )    ToolSize = 2;
   }

   Tool.TC_TP3    = ToolSize;                                  // Größe nach links
   Tool.TC_TP4    = ToolSize;                                  // Größe nach rechts
   Tool.TC_TP5    = ToolSize;                                  // Größe nach oben
   Tool.TC_TP6    = ToolSize;                                  // Größe nach unten
}

void EncodeToolData()
{
   writeWORD   ( ToolArray, 0,    Tool.TC_TP1 );                // Duplonummer      Byte 00-01
   writeASCII  ( ToolArray, 2, 10, Tool.TC_TP2 );               // Werkzeugname     Byte 02-11
   writeINT    ( ToolArray, 12,   Tool.TC_TP7 );                // Platz-Typ        Byte 12-13

   int EdgeCount = Tool.EDGE_NUM;

   writeINT    ( ToolArray, 14,   EdgeCount );                  // Anzahl Schneiden Byte 14-15
```

```
    int EdgeIndex;

    for ( EdgeIndex = 1 ; EdgeIndex <= EdgeCount ; EdgeIndex++ )
    {
      int EdgeNo          = Tool.EDGE_NO[EdgeIndex] ;                    // Schneiden-Nummer
      int EdgeOffset   = 16 + (EdgeIndex-1) * 24;                         // Offset zum Beginn der jeweiligen Schneide
                                                                          // 24: Größe der Scheidendaten

      writeINT ( ToolArray, EdgeOffset +  0, EdgeNo );                    // Schneiden-Nummer  Byte 16-17
      writeINT ( ToolArray, EdgeOffset +  2, Tool.TC_DP1[EdgeNo] );       // Werkzeugtyp       Byte 18-19
      writeREAL( ToolArray, EdgeOffset +  4, Tool.TC_DP2[EdgeNo] );       // Schneidenlage     Byte 20-23
      writeREAL( ToolArray, EdgeOffset +  8, Tool.TC_DP3[EdgeNo] );       // Geometrie Länge 1 Byte 24-27
      writeREAL( ToolArray, EdgeOffset + 12, Tool.TC_DP4[EdgeNo] );       // Geometrie Länge 2 Byte 28-31
      writeREAL( ToolArray, EdgeOffset + 16, Tool.TC_DP5[EdgeNo] );       // Geometrie Länge 3 Byte 32-35
      writeREAL( ToolArray, EdgeOffset + 20, Tool.TC_DP6[EdgeNo] );       // Geometrie Radius  Byte 36-39
    }
}
```

13.3 Kommando-Codes

13.3.1 Funktionsumfang der PLC-Schnittstelle (K-Codes)

Funktionsumfang der PLC-Schnittstelle

Die PLC-Schnittstelle hat einen fest definierten Funktionsumfang. Jeder/m definierten Funktion/Dienst ist ein eindeutiger Kommando-Code (K-Code) zugeordnet. Abhängig von dem auszuführenden Kommando müssen vor dem Setzen des Request-Bits verschiedene Übergabeparameter in der PLC beschrieben werden.

Über die PLC-Schnittstelle können folgende Funktionen über Tool Ident Connection ausgeführt werden:

K-Code	Beschreibung
0	Beladen vom Codeträger (K-Code = 0) (Seite 409)
1	Entladen auf den Codeträger (K-Code = 1) (Seite 410)
7	Werkzeug beladen aus der Werkzeugliste (K-Code = 7) (Seite 412)
8	Werkzeug entladen in die Werkzeugliste (K-Code = 8) (Seite 413)
9	Leerplatzsuche (K-Code = 9) (Seite 414)
10	Werkzeug im NC löschen (K-Code = 10) (Seite 415)
14	Codeträger aktualisieren (K-Code = 14) (Seite 416)
15	Lesen der Daten vom Codeträger (K-Code = 15) (Seite 417)

In den nachfolgenden Abschnitten werden detailliert die einzelnen Funktionen von Tool Ident Connection aufgeführt. Für jeden Dienst sind die optionalen und Pflicht-Übergabeparameter aufgelistet.

Des Weiteren erhält man eine Übersicht der Rückgabeparameter, die Tool Ident Connection nach erfolgreicher Durchführung eines Kommando-Codes liefert.

13.3.2 Parameterbeschreibung

Übergabeparameter für die Dienste

In der folgenden Tabelle sind die Begriffe der Übergabeparameter beschrieben:

Parameter	Beschreibung
ActiveEnable	Übernahme des WZ-Statusbits aus der Datenbank in den NC (TRUE) / Ausblenden des Statusbits während der Übernahme (FALSE)
Channel	Kanalnummer
DeleteTool	Werkzeug nach dem Entladen löschen (TRUE) / nicht löschen (FALSE)
Duplo	Duplonummer
FileIndex	Index der Datei von der importiert bzw. in die exportiert werden soll. Der Dateiname, der durch den Index erweitert wird, kann in der INI-Datei vorgegeben werden.
Ident	Werkzeugname
LoadingPlace	Nummer des Be-/Entladeplatzes im Beladem agazin
LoadTool	Werkzeug nach dem Lesen beladen (TRUE) / nicht beladen (FALSE)
Magazine	Magazinnummer
MagazinePlace	Magazinplatznummer des Werkzeugs
MagazinePlaceType	Magazinplatztyp
NCU_Index	Index der NCU (Parameter wird derzeit nicht genutzt)
Override	Exportdatei überschreiben (TRUE) / erweitern (FALSE)
Subtype	Subtyp bzw. Typ
TNumber	T-Nummer
ToolSize_Down	Werkzeuggröße unten in Halbplätzen
ToolSize_Left	Werkzeuggröße links in Halbplätzen
ToolSize_Right	Werkzeuggröße rechts in Halbplätzen
ToolSize_Upper	Werkzeuggröße oben in Halbplätzen
ToolState	Werkzeugstatus
Unit	Codeträger-Einheit (Lese-/Schreibkopf), von der gelesen bzw. auf die geschrieben werden soll. Je nach Gerät: 1...4 .

13.3.3 Beladen vom Codeträger (K-Code = 0)

Liest die Werkzeugdaten vom Codeträger, dessen Einheit mit "Unit" angegeben wird. Danach wird das Werkzeug angelegt, wenn es noch nicht im NC vorhanden sein sollte. Außerdem wird es beladen, wenn "LoadTool" gesetzt wurde. Dazu wird die Nummer des Beladeplatzes im Beladem agazin aus "LoadingPlace" verwendet, wenn "LoadingPlace" > 0 ist. Ist aber "LoadingPlace" = 0 gesetzt, wird der Wert aus der Node "LoadPlaceNo" in der Konfigurationsdatei als Beladeplatznummer benutzt.

Die Kanal-Nummer (Channel) und die Nummer des Lesekopfes (Unit) müssen angegeben werden.

Werden die Parameter "Magazine" mit ">0" und "MagazinePlace" mit "0" belegt, wird versucht das Werkzeug auf einen passenden Leerplatz in dem angegebenen Magazin zu beladen.

Sind die Werte für "Magazine" und "MagazinePlace" > 0, wird versucht, das Werkzeug auf diesen Magazinplatz zu beladen.

In allen anderen Fällen wird das Werkzeug in dem Magazin beladen, in dem ein Leerplatz gefunden wird.

Tabelle 13-6 PLC Parameter für "Beladen vom Codeträger"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3040	Magazin Platz	INT	-1...32000	Magazinplatz Nummer des WZ
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBB3049	Loading Place	BYTE	1...127	Nummer des Be-/Entladeplatzes
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
DBB3052	Unit	BYTE	1...127	Nummer des Codeträgers
DBX3056.2	LoadTool	BOOL	True, False	WZ nach Anlegen beladen
Rückgabeparameter				
DBW3058	Subtype	INT	-1...32000	Subtype des Werkzeuges
DBW3060	Duplo	INT	-1...32000	Duplo-Nummer des WZ
DBW3096	T-Nummer	INT	-1...32000	T-Nummer des WZ
DBW3098	ToolState	INT	-1...32000	Werkzeugstatus
DBW3100	Magazine Platztype	INT	-1...32000	Magazinplatztyp des Werkzeugs
DBB3102	ToolSize_Upper	BYTE	1...127	WZ-Größe oben in Halbplätzen
DBB3103	ToolSize_Down	BYTE	1...127	WZ-Größe unten in Halbplätzen
DBB3104	ToolSize_Left	BYTE	1...127	WZ-Größe links in Halbplätzen
DBB3105	ToolSize_Right	BYTE	1...127	WZ-Größe rechts in Halbplätzen
DBW3106	Magazine	INT	0...32000	Magazin-Nummer des WZ
DBW3108	Magazine Place	INT	0...32000	Magazinplatz-Nummer des WZ

13.3.4 Entladen auf den Codeträger (K-Code = 1)

Entlädt das Werkzeug, welches anhand der Übergabeparameter parametriert wurde. Die Kanal-Nummer und die Nummer des Scheibenkopfes müssen angegeben werden. Wenn die Parameter "Magazine" und "MagazinePlace > 0" sind, wird versucht das Werkzeug zu entladen, das sich auf diesem Platz befindet. Sonst wird das Werkzeug mit den Parametern "Duplo" und "Ident" oder über den Parameter "TNumber" ausgewählt.

Dabei wird die Nummer des Entladeplatzes im Beladem agazin aus "LoadingPlace" verwendet, wenn "LoadingPlace > 0" ist. Ist "LoadingPlace = 0" gesetzt, wird der Wert aus der Node "UnloadPlaceNo" in der Konfigurationsdatei als Entladeplatznummer (Beladestelle) benutzt.

Wenn "DeleteTool" gesetzt ist, wird das Werkzeug nach dem Entladevorgang auch aus dem NC gelöscht.

Tabelle 13-7 PLC Parameter für "Entladen auf den Codeträger"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	0...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBW3040	MagazinePlace	INT	0...32000	Magazinplatz
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBB3049	Loading Place	BYTE	1...127	Nummer des Be-/Entladeplatzes
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
DBB3052	Unit	BYTE	1...127	Nummer des Codeträgers
DBX3056.3	DeleteTool	BOOL	TRUE, FALSE	WZ nach Entladen löschen
Rückgabeparameter				
keine				

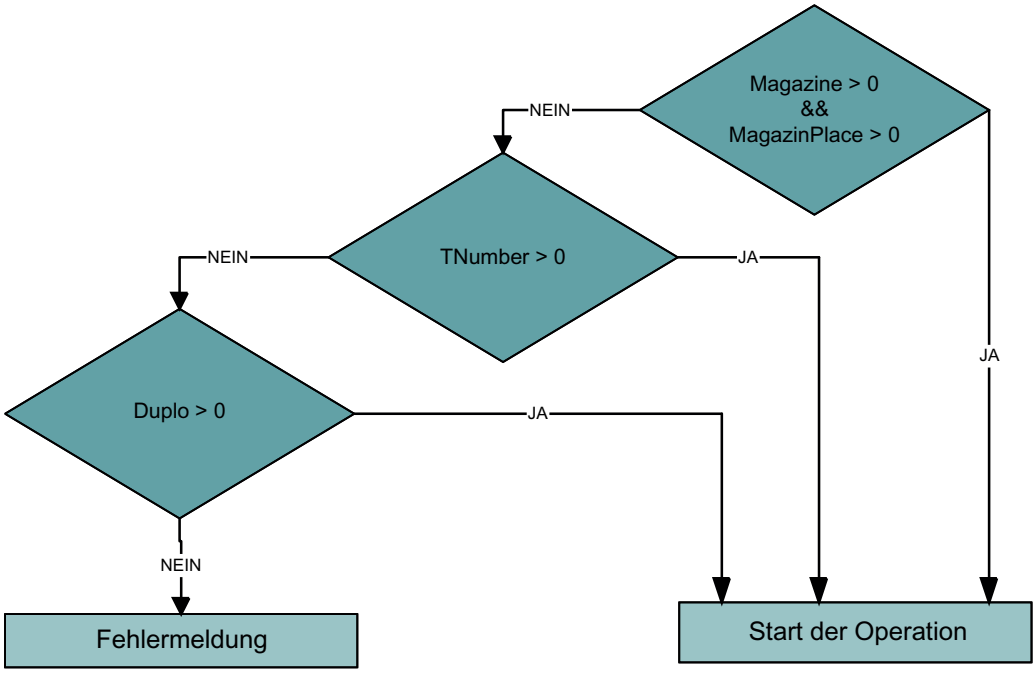


Bild 13-15 Ablaufplan zur Identifizierung von Werkzeugen

13.3.5 Werkzeug beladen aus der Werkzeugliste (K-Code = 7)

Belädt ein Werkzeug aus der Werkzeugliste, welches anhand der Übergabeparameter parametriert wurde. Das Werkzeug kann mit den Parametern "Duplo" und "Ident" oder über den Parameter "TNumber" ausgewählt werden. Zum Beladen wird die Nummer des Beladeplatzes im Beladem agazin aus "LoadingPlace" verwendet, wenn "LoadingPlace" > 0 ist. Ist aber "LoadingPlace" = 0 gesetzt, wird der Wert aus der Node "LoadPlaceNo" in der Konfigurationsdatei als Beladeplatznummer benutzt. Die Kanal-Nummer muß für die Operation angegeben werden.

Werden die Parameter "Magazine" mit ">0" und "MagazinePlace" mit "0" belegt, wird versucht das Werkzeug auf einen passenden Leerplatz in dem angegebenen Magazin zu beladen.

Sind die Werte für "Magazine" und "MagazinePlace" > 0, wird versucht, das Werkzeug auf diesen Magazinplatz zu beladen. In allen anderen Fällen wird das Werkzeug in dem Magazin beladen, in dem ein Leerplatz gefunden wird.

Tabelle 13-8 PLC Parameter für "Werkzeug beladen - Werkzeugliste"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	1...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBW3040	MagazinePlace	INT	0...32000	Magazinplatz-Nummer
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBB3049	Loading Place	BYTE	1...127	Nummer des Be-/Entladeplatzes
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
Rückgabeparameter				
keine				

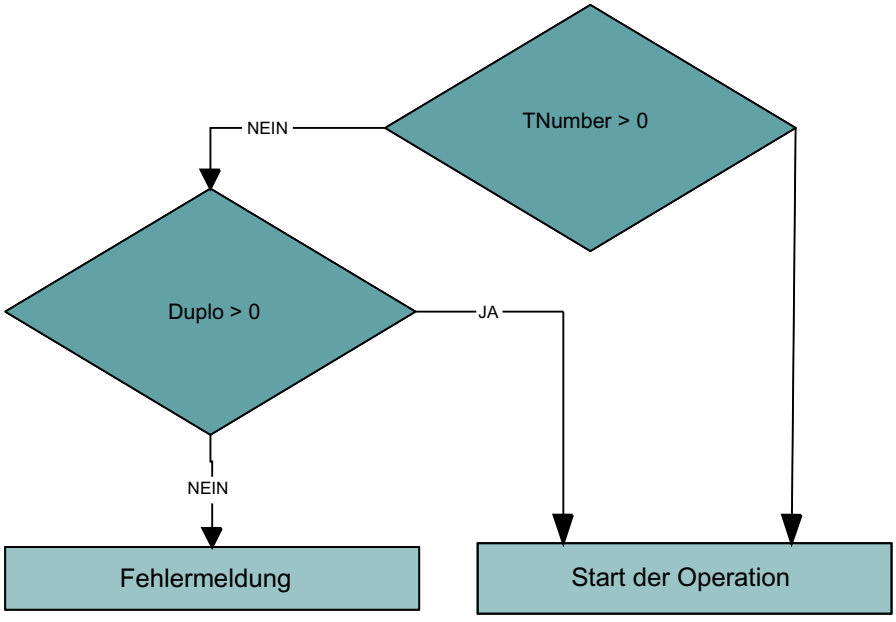


Bild 13-16 Ablaufplan zur Identifizierung von Werkzeugen

13.3.6 Werkzeug entladen in die Werkzeugliste (K-Code = 8)

Entlädt das Werkzeug in die Werkzeugliste, welches anhand der Übergabeparameter parametriert wurde. Die Kanal-Nummer muss für diese Operation angegeben werden.

Wenn die Parameter "Magazine" und "MagazinePlace > 0" sind, wird versucht das Werkzeug zu entladen, das sich auf diesem Platz befindet. Sonst wird das Werkzeug mit den Parametern "Duplo" und "Ident" oder über den Parameter "TNumber" ausgewählt.

Dabei wird die Nummer des Entladeplatzes im Beladeplatzin aus "LoadingPlace" verwendet, wenn "LoadingPlace > 0" ist. Ist "LoadingPlace = 0" gesetzt, wird der Wert aus der Node "UnloadPlaceNo" in der Konfigurationsdatei als Entladeplatznummer (Beladestelle) benutzt.

Wenn "DeleteTool" gesetzt ist, wird das Werkzeug nach dem Entladevorgang auch aus dem NC gelöscht.

Tabelle 13-9 PLC Parameter für "Werkzeug entladen - Werkzeugliste"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	0...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBB3049	LoadingPlace	BYTE	0...127	Nummer der Beladestelle
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
DBW3040	MagazinPlace	INT	0...32000	Magazinplatz
DBX3056.3	DeleteTool	BOOL	True, False	WZ nach dem Entladen löschen
Rückgabeparameter				
keine				

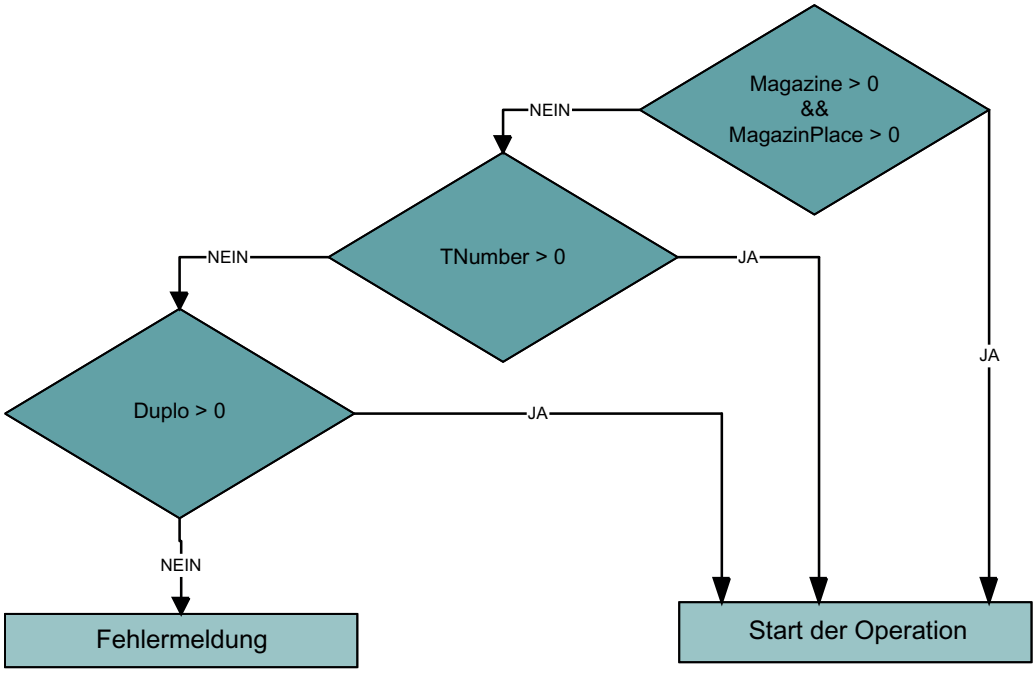


Bild 13-17 Ablaufplan zur Identifizierung von Werkzeugen

13.3.7 Leerplatzsuche (K-Code = 9)

Für das mit "TNumber" oder mit "Ident" und "Duplo" spezifizierte Werkzeug wird eine Leerplatzsuche oder Zielplatzprüfung durchgeführt. Bei "TNumber" = 0 wird ein Leerplatz für das durch "Ident" und "Duplo" angegebene Werkzeug gesucht, bei "TNumber" > 0 mittels T-Nummer.

Für die Durchführung der Funktion ist zwingend ein gültiger Wert für den Parameter "Channel" zu setzen. Anhand der angegebenen Beladestelle, wird versucht, ein Leerplatz in den Magazinen zu finden, mit denen die Beladestelle verbunden ist. Dazu wird die Nummer der Beladestelle im Beladem agazin aus "LoadingPlace" verwendet. Ist aber "LoadingPlace" = 0 gesetzt, wird der Wert aus der Node "LoadPlaceNo" in der Konfigurationsdatei als Beladeplatznummer benutzt.

Ist "Magazine" =0 wird ein Leerplatz in einem beliebigen Magazin gesucht, ist eine Magazin-Nummer angegeben wird in dem entsprechenden Magazin gesucht. Wird zusätzlich ein Magazinplatz >0 angegeben, führt die Funktion eine Zielplatzprüfung durch. Dies ist bei festplatzkodierten Werkzeugen erforderlich.

Tabelle 13-10 PLC Parameter für "Leerplatzsuche"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	0...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBB3049	LoadingPlace	BYTE	0...127	Nummer der Beladestelle
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
DBW3040	MagazinPlace	INT	0...32000	Magazinplatz
Rückgabeparameter				
DBW3058	Subtype	INT	-1...32000	Subtype des Werkzeuges
DBW3060	Duplo	INT	-1...32000	Duplo Nummer des WZ
DBW3096	T-Nummer	INT	-1...32000	T-Nummer des WZ
DBW3098	ToolState	INT	-1...32000	Werkzeugstatus
DBW3100	Magazine Platztype	INT	-1...32000	Magazinplatztyp des Werkzeugs
DBB3102	ToolSize_Upper	BYTE	1...127	WZ-Größe oben in Halbplätzen
DBB3103	ToolSize_Down	BYTE	1...127	WZ-Größe unten in Halbplätzen
DBB3104	ToolSize_Left	BYTE	1...127	WZ-Größe links in Halbplätzen
DBB3105	ToolSize_Right	BYTE	1...127	WZ-Größe rechts in Halbplätzen
DBW3106	Magazine	INT	0...32000	Magazin-Nummer des WZ
DBW3108	MagazinePlace	INT	0...32000	Magazinplatz-Nummer des WZ

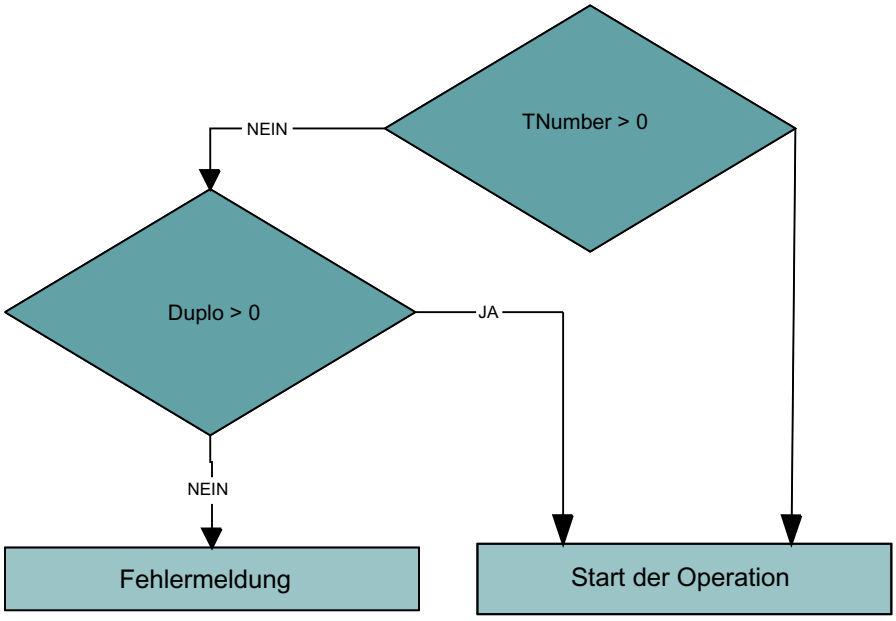


Bild 13-18 Ablaufplan zur Identifizierung von Werkzeugen

13.3.8 Werkzeug im NC löschen (K-Code = 10)

Löscht das mit "TNumber" oder mit "Ident" und "Duplo" angegebene Werkzeug im NC, wenn dieses nicht beladen ist.

Bei der Angabe einer T-Nummer > "0", wird versucht, das Werkzeug mit dieser Nummer im NC zu löschen.

Wenn "TNumber = 0" gesetzt ist, wird versucht, das Werkzeug mit der angegebenen "Ident" und "Duplo" im NC zu löschen.

Tabelle 13-11 PLC Parameter für "Werkzeug im NC löschen"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	0...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBB3048	Channel	BYTE	1...127	TO-Bereich
Rückgabeparameter				
keine				

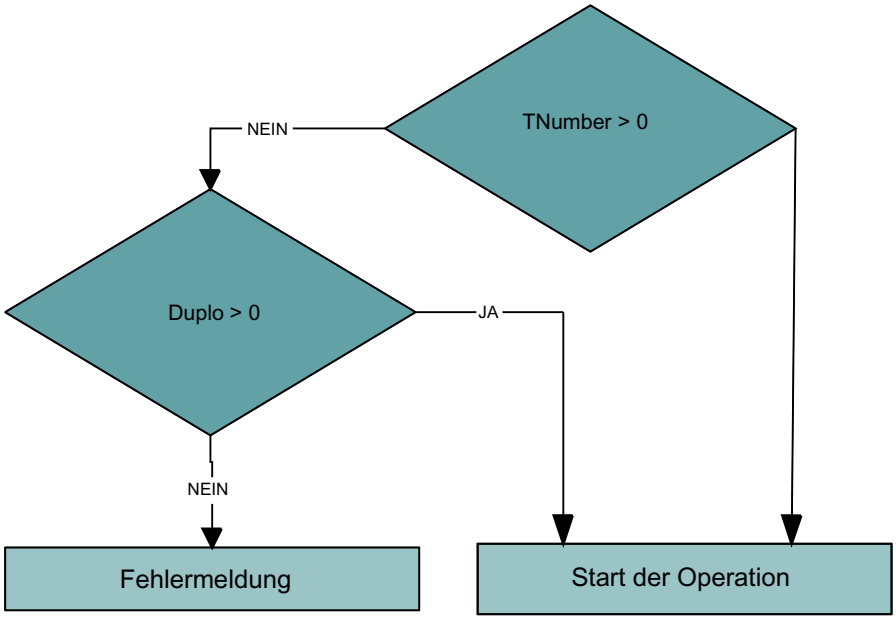


Bild 13-19 Ablaufplan zur Identifizierung von Werkzeugen

13.3.9 Codeträger aktualisieren (K-Code = 14)

Aktualisiert den Codeträger mit den Daten des Werkzeugs, welches anhand der Übergabeparameter parametriert wurde.

Die Kanal-Nummer und die Nummer des Lesekopfes müssen angegeben werden.

Wenn die Parameter "Magazine" und "MagazinePlace > 0" sind, wird versucht die Daten des Werkzeugs, das sich auf diesem Platz befindet, auf den Codeträger zu schreiben. Sonst wird das Werkzeug mit den Parametern "Duplo" und "Ident" oder "TNumber" ausgewählt.

Tabelle 13-12 PLC Parameter für "Codeträger aktualisieren"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBW3002	Duplo	INT	0...32000	Duplo-Nummer
DBW3038	TNumber	INT	0...32000	T-Nummer
DBB3048	Channel	BYTE	1...127	TO-Bereich
DBW3050	Magazine	INT	0...32000	Magazin-Nummer
DBW3040	MagazinPlace	INT	0...32000	Magazinplatz
DBB3052	Unit	BYTE	1...127	Nummer der Codeträgers
Rückgabeparameter				
keine				

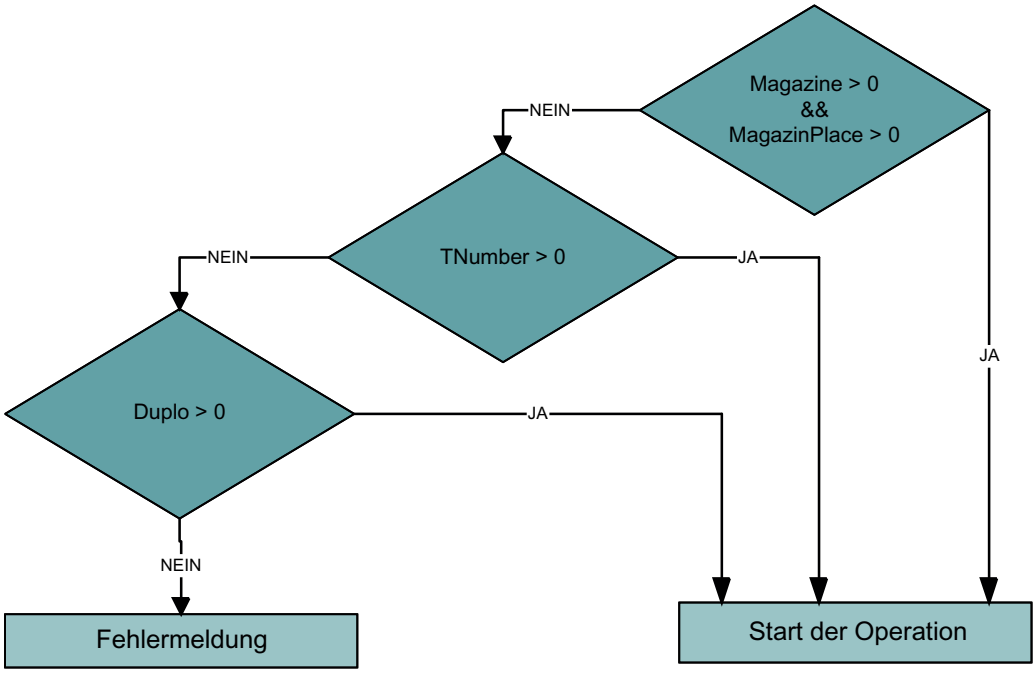


Bild 13-20 Ablaufplan zur Identifizierung von Werkzeugen

13.3.10 Lesen der Daten vom Codeträger (K-Code = 15)

Lesen der Werkzeugdaten vom Codeträger und schreiben der Rückgabeparameter in die PLC. Der Codeträger wird über "Unit" identifiziert.

Damit ist es z.B. möglich von einem Codechip auf dem nur die "Duplo"-Nummer und der Werkzeugbezeichner gespeichert sind, diese zwei Daten zu lesen. Danach kann man dann z.B. die Dienste "Be-/Entladen aus der Werkzeugliste" aufrufen.

Damit können auch kostengünstige "readonly" Codechips verwendet werden.

Tabelle 13-13 PLC Parameter für "Lesen der Daten vom Codeträger"

PLC-Datum	Bezeichner	Typ	Wertebereich	Bedeutung
Übergabeparameter				
DBB3052	Unit	BYTE	1...127	Nummer des Lesekopfs
Rückgabeparameter				
DBW3058	Subtype	INT	-1...32000	Subtype des Werkzeuges
DBW3060	Duplo	INT	-1...32000	Duplo Nummer des WZ
DBW3098	ToolState	INT	-1...32000	Werkzeugstatus
DBW3100	Magazine Platztype	INT	-1...32000	Magazinplatztyp des Werkzeugs
DBB3102	ToolSize_Upper	BYTE	1...127	WZ-Größe oben in Halbplätzen
DBB3103	ToolSize_Down	BYTE	1...127	WZ-Größe unten in Halbplätzen
DBB3104	ToolSize_Left	BYTE	1...127	WZ-Größe links in Halbplätzen
DBB3105	ToolSize_Right	BYTE	1...127	WZ-Größe rechts in Halbplätzen

13.3.11 Fehlercodes

Die Fehlercodes, die im PLC-Byte "K_CodeError" bzw. PLC-Wort "Error_Code" der PLC-Nahtstelle abgelegt werden, liegen im hexadezimalen Format vor und weisen auf die Stelle hin, an der die Ausführung des Auftrags mit einem Fehler abgebrochen wurde.

Sollte der Fehlercode dezimal dargestellt werden, muss dieser in das hexadezimale Format konvertiert werden, um die Fehlerursache aus den Tabellen A-1 bis A-8 zu ermitteln.

Der Fehlercode in "K_CodeError" identifiziert den Kommandocode, bei dem der Fehler aufgetreten ist.

Tabelle 13-14 Kommandocodes des Auftrags im Fehlerfall

K_Code-Error	Dienst
00	Kein Fehler
01	Beladen vom Codeträger
02	Entladen auf den Codeträger
03	
04	
05	Beladen vom Fertigungsleitrechner
06	Entladen in den Fertigungsleitrechner
07	
08	Werkzeug beladen
09	Werkzeug entladen
0A	Leerplatzsuche
0B	Werkzeug im NC löschen
0C	
0D	
0E	
0F	Codeträger aktualisieren
10	Lesen der Daten vom Codeträger
11	
...	Reserve

Die Fehlercodes "Error_Code" setzen sich aus einem High-Byte und einem Low-Byte zusammen, die eine unterschiedliche Bedeutung haben. Das High-Byte (XX) identifiziert die Fehlerartgruppe, in der der Fehler aufgetreten ist. Anhand des Low-Byte (YY) wird die Fehlerursache spezifiziert, die für den Abbruch des Auftrags verantwortlich ist.

Format von Error_Code

XX	YY

Die Inhalte von XX = 00 und YY = 00 bedeuten, dass kein Fehler aufgetreten ist.

XX = 01: Allgemeinfehler-Gruppe

Tabelle 13-15 Low-Byte (YY) des Allgemeinfehlercodes

	Fehlerursache
01	Fehler beim Lesen der Parameter aus DB1900
02	Fehler beim Aufruf des internen Dienstes
03	Zeitüberschreitung bei der Ausführung des Dienstes
04	Es wurden noch keine Werkzeugdaten ermittelt
05	Fehler beim Ausführen des internen Dienstes
...	Reserve
	Fehler in Übergabeparameter
10	Subtype
11	Duplo
12	Ident
13	TNumber
14	MagazinePlace
15	MagazinePlaceType
16	ToolSize_Upper
17	ToolSize_Down
18	ToolSize_Left
19	ToolSize_Right
1A	Channel
1B	Magazine
1C	Unit
1D	NCU_Index
1E	FileIndex
1F	ActiveEnable
20	Override
21	LoadTool
22	DeleteTool
23	Länge von Ident
...	Reserve

XX = 02: Codeträgerfehler-Gruppe

Tabelle 13-16 Low-Byte (YY) des Codeträgerfehlercodes

	Fehlerursache
01	Kein Codeträger vorhanden
02	Fehler beim Codeträger lesen
03	Lesen abgebrochen, da Codeträger entfernt wurde
04	Fehler beim Codeträger beschreiben
05	Schreiben abgebrochen, da Codeträger entfernt wurde

	Fehlerursache
0C	Kommandostring unbekannt
0D	Anzahl Byte unzulässig
0E	Nicht-BCD-Zeichen in empfangenen Schreibdaten gefunden
17	Ungültige Daten
18	Gerät nicht vorhanden
19	Timeout
1A	Hardware Fehler
1B	Ungültige Kopfnumber
1C	Lesekopf nicht angeschlossen
1E	Vom Gerät kann nur gelesen werden
1F	Allgemeiner Lesefehler
20	Allgemeiner Schreibfehler
28	Unbekannter Fehler
68	Kein Datenträger erkannt (Lesen/Schreiben, erste Seite)
69	Kein Datenträger erkannt (Lesen/Schreiben Folgeseite)
6D	Kein Datenträger erkannt (Schreiben, erste Seiten)
6E	Kein Datenträger erkannt (Lesen vor Schreiben)
6F	Kein Datenträger erkannt (Schreiben letzte Seite)
73	Nicht-BCD-Zeichen in empfangenen Schreibdaten gefunden
74	BCD-Format lesen: Nicht-BCD-Zeichen im Datenträger gefunden
8D	Fehlerhafter Zugriff auf Datenträger
8E	Daten im Datenträger sind unvollstaendig

XX = 05: NC-Fehler-Gruppe

Tabelle 13-17 Low-Byte (YY) des NC-Fehlercodes

	Fehlerursache
00	Verbindungsaufnahme zur gefordeten NCU fehlgeschlagen
05	Lesen der NC-Konfiguration fehlgeschlagen
0A	Eine ungültige Kanal-Nummer wurde im Auftrag angegeben
3C	Eine ungültige Magazin-Nummer wurde im Auftrag angegeben
3E	Eine ungültige Magazinplatz-Nummer wurde im Auftrag angegeben
3F	Eine ungültige Schneiden-OEM-Überwachungsparameter-Nummer wurde im Auftrag angegeben
41	Der MMC-Semaphore für das Werkzeug-Suchen/-Anlegen schon belegt
46	Das mit Ident und Duplo angegebene Werkzeug wurde nicht im NC gefunden
47	Das mit T-Nummer angegebene Werkzeug wurde nicht im NC gefunden
48	Keine Werkzeuge wurden im angegebenen Magazin gefunden
49	Auf dem angegebenen Magazinplatz ist kein Werkzeug beladen
4A	Das mit Ident und Duplo angegebene Werkzeug konnte nicht eindeutig im NC ermittelt werden
4B	Anzahl der gefundenen Werkzeuge konnte nicht im NC gelesen werden

	Fehlerursache
50	Das mit Ident und Duplo angegebene Werkzeug wurde nicht im NC angelegt
55	Das angegebene Werkzeug ist im Magazin bereits beladen
5A	Die WZ-Schneide mit der angegebenen Nummer ist nicht im NC angelegt worden
5B	Die WZ-Schneide mit der angegebenen Nummer ist nicht im NC gelöscht worden
5C	Die angegebene Schneide ist nicht im NC vorhanden
5D	Die angegebene WZ-Schneidennummer ist nicht gültig
5E	Die angegebene Schneide ist schon im NC vorhanden
5F	Werkzeug-Zustand konnte nicht im NC gelesen werden
64	Der MMC-Semaphore für die Leerplatzsuche schon belegt
69	Ein Magazin-Leerplatz konnte im NC beim Beladen nicht gefunden werden
6E	Überprüfung des angegebenen Magazin-Platzes zum Beladen negativ quittiert
6F	Der Werkzeug-Löschvorgang wurde mit Fehler beendet
70	Die Leerplatzsuche wurde mit Fehler beendet
71	Das Anlegen eine Werkzeuges wurde mit Fehler beendet
72	Operation zum Umsetzen von Werkzeugen ist fehlgeschlagen
73	Der MMC-Semaphore für das Werkzeug-Be-/Entladen schon belegt
74	Alarmverlängerung wurde mit Fehler beendet
78	Das Magazin gesperrt oder nicht zum Beladen freigegeben
7D	Der Werkzeug-Beladevorgang wurde mit Fehler beendet
82	Die Magazin-Nummer des Werkzeugs konnte nicht im NC gelesen werden
87	Das angegebene Werkzeug ist nicht im NC beladen
8C	Das angegebene Werkzeug ist im anderen Magazin beladen
8E	Das angegebene Werkzeug ist auf anderem Magazinplatz beladen
91	Magazin-Zustand konnte nicht im NC gelesen werden
96	Das Magazin nicht in Beladeposition oder nicht zum Entladen freigegeben
9B	Der Werkzeug-Entladevorgang wurde mit Fehler beendet
A0	Das Werkzeug konnte nicht im NC gelöscht werden
A5	Das Werkzeug ist im NC aktiv! Das Werkzeug konnte nicht im NC gelöscht werden
D2	Das angegebene Werkzeug ist im NC beladen! Das Werkzeug wurde nicht im DB gelöscht

XX = 0B: Wkonvertfehler-Gruppe

Tabelle 13-18 Low-Byte (YY) des Filefehlercodes

	Fehlerursache
01	1401 Fehler beim öffnen der MCX-Datei
02	1402 Fehler beim Lesen der MCX-Datei
03	1403 Fehler beim Lesen der MCX-Datei
04	1404 Fehler beim Lesen der MCX-Datei
05	1405 Fehler beim Lesen der MCX-Datei
06	1406 Fehler beim Lesen der MCX-Datei
07	1410 Fehler beim Lesen der MCX-Datei
08	1628 Systemfehler

	Fehlerursache
09	1801 Systemfehler
0A	1803 Systemfehler
0B	1804 Systemfehler
0C	1805 Falsche Version der MCX-Datei
0D	1806 MCX-Datei enthält nicht unterstützte Funktionen
0E	1807 Systemfehler
0F	1808 Fehler beim Lesen der MCX-Datei
10	1809 Systemfehler
11	1810 Systemfehler
12	2401 Systemfehler
13	2402 Systemfehler
14	2501 Systemfehler
15	2502 Systemfehler
16	2503 Systemfehler
17	2701 Systemfehler
18	2702 Systemfehler
19	2703 Indexfehler beim Zugriff auf Array
1A	2801 Systemfehler ConvertFunctions::CheckAccess()
1B	2802 Systemfehler
1C	2803 Fehler im Zusammenhang mit Konvertierungsfunktionen readXXX()/writeXXX(). Basis Type des Arrary-Parameter muss den 'unsigned char' sein.
1D	2804 Fehler in der Funktion writeBCD(). Die Länge der Daten (val) überschreitet die Größe des Zielbereiches.
1E	2805 Fehler in der Funktion PlaceTypeFromString(). Verwendeter ToolTypeText ist in der ToolSpec.xml nicht definiert.
1F	2806 Fehler in der Funktion PlaceTypeToString(). Verwendeter ToolTypeCode ist in der ToolSpec.xml nicht definiert.
20	2807 Fehler in der Funktion writeREAL(). Datenverlust durch Rundung.
21	2901 Systemfehler
22	2903 Systemfehler
23	2904 Systemfehler
24	2905 Systemfehler
25	2908 Systemfehler
26	2909 Systemfehler
27	2910 Systemfehler
28	2912 Systemfehler
29	2913 Fehler beim Zugriff auf SlTmService. setVal()
2A	2914 Fehler beim Zugriff auf SlTmService. setVal()
2B	2915 Fehler beim Zugriff auf SlTmService. setVal()
2C	2917 Fehler beim Zugriff auf SlTmService. rData().val()
2D	2918 Fehler beim Zugriff auf SlTmService. rData().val()
2E	2919 Fehler beim Zugriff auf SlTmService. setVal()
2F	2921 Fehler beim Zugriff auf SlTmService. val()
30	2922 Systemfehler

	Fehlerursache
31	2923 Fehler beim Zugriff auf SlTmService. val()
32	2924 Systemfehler
33	7001 Systemfehler
34	7002 Die Konvertierungsvorschrift muss ein Objekt mit dem Namen 'Tool' enthalten.
35	7003 Das 'Tool'-Objekt muss den Type 'TmTool' haben.
36	7004 Die Konvertierungsvorschrift muss ein Objekt mit dem Namen 'ToolArray' enthalten.
37	7005 Das 'ToolArray'-Objekt muss ein Array sein.
38	7006 Das 'ToolArray'-Objekt muss ein Array mit dem Basistype 'unsigned char' sein.
39	7007 wurde die Funktion 'DecodeToolData' innerhalb der Konvertierungsvorschrift nicht gefunden.
3A	7008 Das Objekt 'DecodeToolData' muss eine Funktion sein.
3B	7009 Es wurde die Funktion 'EncodeToolData' innerhalb der Konvertierungsvorschrift nicht gefunden.
3C	7010 Das Objekt 'EncodeToolData' muss eine Funktion sein.
3D	7011 Systemfehler
3E	7012 Systemfehler
3F	7013 Systemfehler
40	7014 Systemfehler
41	7015 Fehler beim Lesen der ToolSpec.xml'
42	7016 Fehler beim Interpretieren der ToolSpec.xml
43	7017 der ToolSpec.xml. Das Dokument-Element sollte den Namen: "ToolSpecConversion" tragen.
44	7018 Formatfehler in der ToolSpec.xml. Attribute code="" bzw. text="" sind nicht enthalten.
45	Systemfehler der Datenkonvertierung
80 - FF	Benutzerdefinierte Fehler aus der Konvertierungsvorschrift (siehe Fehlerbehandlung (Seite 403))

13.4 Beispiele

Hinweis

Beipielprojekt

Ein Beispielprojekt für das Be- und Entladen auf den Codeträger über die PLC-Schnittstelle wird mit der Toolbox ausgeliefert.

13.4.1 Beladen vom Codeträger über PLC-Schnittstelle

In diesem Beispiel werden von einem Codeträger die Daten des imaginären Werkzeugs "WZ_Test13" gelesen, das Werkzeug im NC angelegt und beladen. Das Werkzeug soll im Magazin "1" der Standard-NCU im Kanal "1" beladen werden. Der Vorgang soll über den Beladeplatz "1" des Beladem agazins abgewickelt werden.

Die für diesen Dienst erforderlichen Parameter werden in den folgenden Tabellen dargestellt.

Vor dem Ausführen des Dienstes wird von der PLC der Wert "0" in das Kommandocode-Byte (K_Code) der PLC-Schnittstelle im Datenbaustein DB1900 geschrieben. Danach wird dieser Dienst durch Setzen des "Request"-Bits gestartet. Wenn das "Active"-Bit zurückgesetzt und das "Done"-Bit gesetzt wurden, kann das "Fehler"-Byte von der PLC ausgewertet werden. Anschließend muss das "Request"-Bit wieder zurückgesetzt werden.

Tabelle 13-19 Übergabeparameter für "Beladen vom Codeträger"

PLC-Datum		**Bezeichner**	**Wert**
DBW	3040	MagazinePlace	0
DBB	3048	Channel	1
DBB	3049	LoadingPlace	1
DBW	3050	Magazine	1
DBB	3052	Unit	1
DBB	3053	NCU_Index	0
DBX	3056.2	LoadTool	TRUE

Tabelle 13-20 Rückgabeparameter von Tool Ident connection für "Beladen vom Codeträger"

PLC-Datum		**Bezeichner**	**Wert**
DBW	3058	Subtype	120
DBW	3060	Duplo	1
DBW	3096	TNumber	289
DBW	3098	ToolState	14
DBW	3100	MagazinePlaceType	1
DBB	3102	ToolSize_Upper	1
DBB	3103	ToolSize_Down	1
DBB	3104	ToolSize_Left	1
DBB	3105	ToolSize_Right	1
DBW	3106	Magazine	1
DBW	3108	MagazinePlace	1

Im nachfolgenden Ablaufplan wird der Ablauf der einzelnen Operationen verdeutlicht:

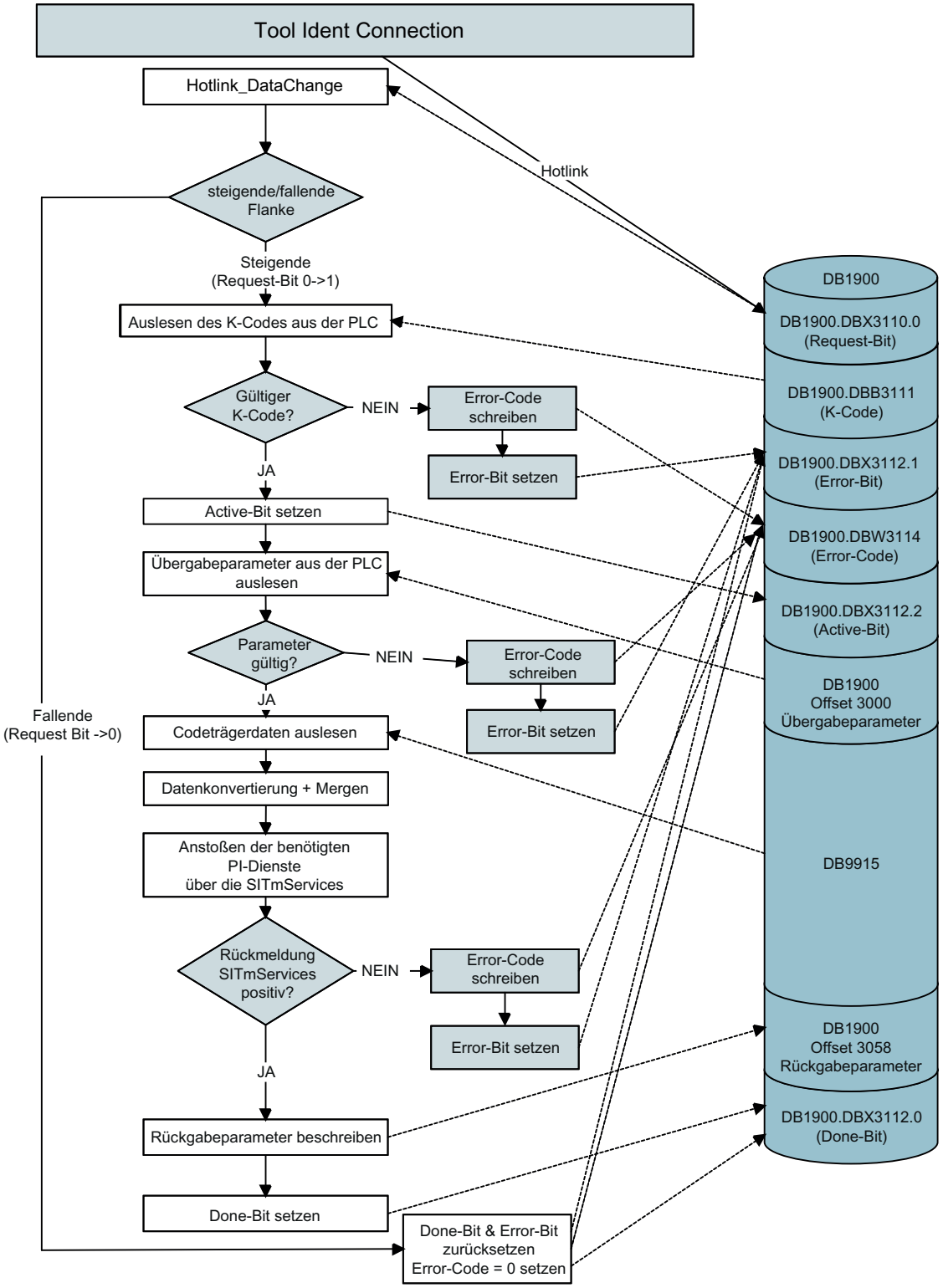
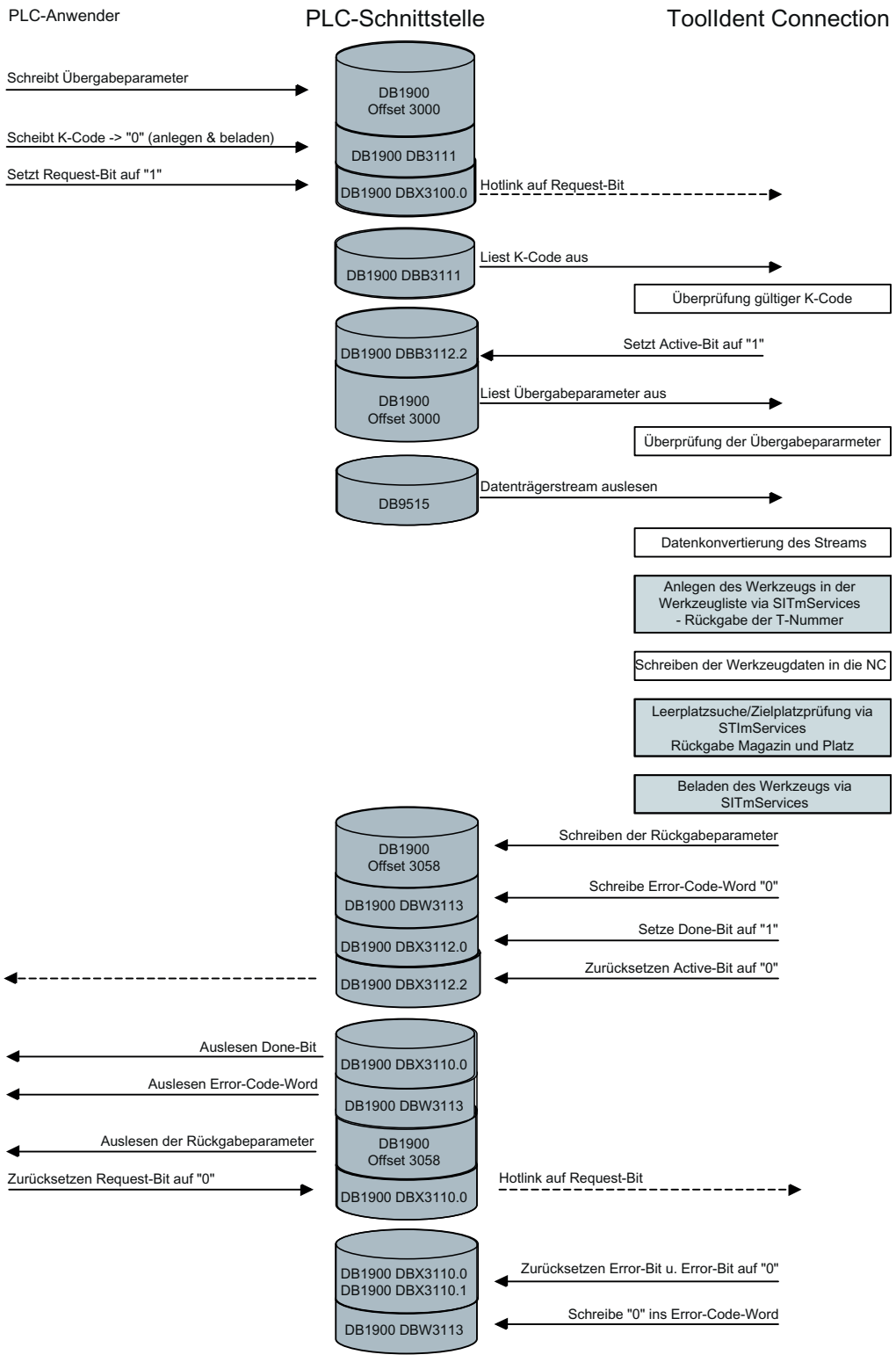


Bild 13-21 Ablaufplan: Beladen vom Codeträger über die PLC-Schnittstelle


PLC-Anwender
PLC-Schnittstelle
ToolIdent Connection
Schreibt Übergabeparameter
Scheibt K-Code -> "0" (anlegen & beladen)
Setzt Request-Bit auf "1"
DB1900
Offset 3000
DB1900 DB3111
DB1900 DBX3100.0
Hotlink auf Request-Bit
DB1900 DBB3111
Liest K-Code aus
Überprüfung gültiger K-Code
DB1900 DBB3112.2
Setzt Active-Bit auf "1"
DB1900
Offset 3000
Liest Übergabeparameter aus
Überprüfung der Übergabeparameter
DB9515
Datenträgerstream auslesen
Datenkonvertierung des Streams
Anlegen des Werkzeugs in der Werkzeugliste via SITmServices - Rückgabe der T-Nummer
Schreiben der Werkzeugdaten in die NC
Leerplatzsuche/Zielplatzprüfung via STImServices Rückgabe Magazin und Platz
Beladen des Werkzeugs via SITmServices
DB1900
Offset 3058
DB1900 DBW3113
DB1900 DBX3112.0
DB1900 DBX3112.2
Schreiben der Rückgabeparameter
Schreibe Error-Code-Word "0"
Setze Done-Bit auf "1"
Zurücksetzen Active-Bit auf "0"
Auslesen Done-Bit
Auslesen Error-Code-Word
Auslesen der Rückgabeparameter
Zurücksetzen Request-Bit auf "0"
DB1900 DBX3110.0
DB1900 DBW3113
DB1900
Offset 3058
DB1900 DBX3110.0
Hotlink auf Request-Bit
DB1900 DBX3110.0
DB1900 DBX3110.1
DB1900 DBW3113
Zurücksetzen Error-Bit u. Error-Bit auf "0"
Schreibe "0" ins Error-Code-Word

13.4.2 Entladen auf den Codeträger über PLC- Schnittstelle

Das im vorangegangenen Beispiel beladene Werkzeug soll entladen und die aktuellen Daten auf den Codeträger geschrieben werden. Nach dem Entladen wird das Werkzeug im NC gelöscht. Der Vorgang soll über den Entladeplatz "2" des Beladem agazins abgewickelt werden.

Die hierfür erforderlichen Übergabeparameter sind in der folgenden Tabelle dargestellt. Dieser Dienst liefert keine Parameter zurück, wodurch keine neuen Werte in den Rückgabeparametern der PLC-Schnittstelle stehen.

Vor der Ausführung des Dienstes muss der Kommandocode "1" in das entsprechende PLC-Byte (K-Code) geschrieben werden. Der Start des Dienstes und die weitere Vorgehensweise sind äquivalent zu der Vorgehensweise in Beladen vom Codeträger über PLC-Schnittstelle (Seite 424).

Tabelle 13-21 Übergabeparameter für "Entladen vom Codeträger"

PLC-Datum		**Bezeichner**	**Wert**
DBW	3002	Duplo	1
DBW	3040	MagazinePlace	1
DBB	3048	Channel	1
DBB	3049	LoadingPlace	2
DBW	3050	Magazine	1
DBB	3052	Unit	1
DBB	3053	NCU_Index	0
DBX	3056.3	DeleteTool	TRUE

Im nachfolgenden Ablaufplan wird der Ablauf der einzelnen Operationen verdeutlicht:

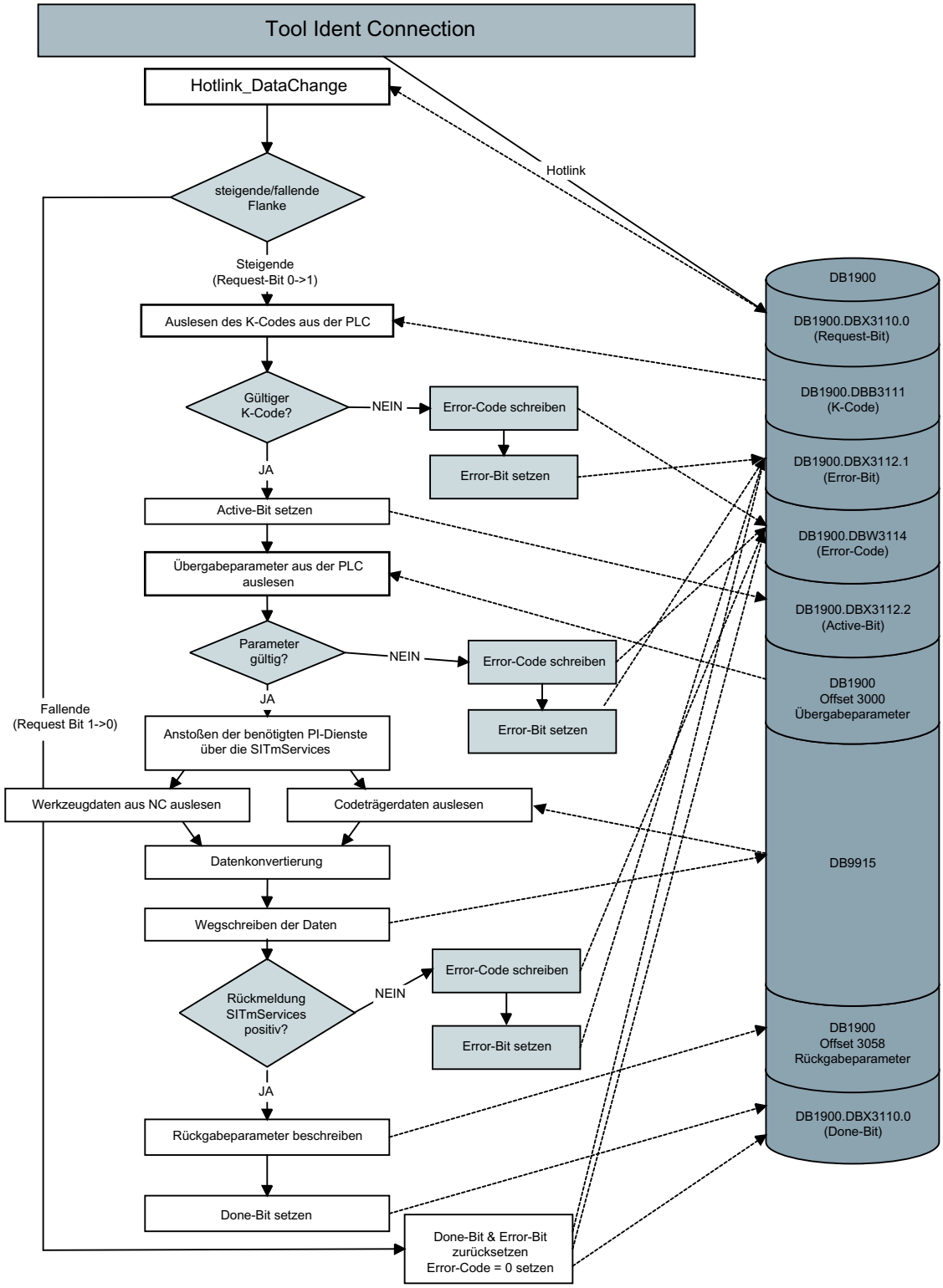


Bild 13-22 Ablaufplan: Entladen vom Codeträger über die PLC-Schnittstelle

Anhang

A

A.1 Liste der Abkürzungen

ASUP	Asynchrones Unterprogramm
BTSS	Bedientafelschnittstelle
CC	Compilezyklus oder OEM oder Anwenderbereich
CUTOM	Cutterradiuscompensation: Werkzeugradiuskorrektur
DB	Datenbaustein in der PLC
DBB	Datenbausteinbyte in der PLC
DBW	Datenbausteinwort in der PLC
DBX	Datenbausteinbit in der PLC
DDE	Dynamic Data Exchange: Dynamischer Datenaustausch
DW	Datenwort
ENC	Encoder: Istwertgeber
EPROM	Erasable Programmable Read Only Memory: Löschbarer, elektrisch programmierbarer Lesespeicher)
FB	Funktionsbaustein
FC	Function Call: Funktionsbaustein in der PLC
GUD	Global User Data: Globale Anwenderdaten
HEX	Kurzbezeichnung für hexadezimale Zahl
HMI	Human Machine Interface
HSA	Hauptspindelantrieb
IBN	Inbetriebnahme
INC	Increment: Schrittmaß
INI	Initializing Data: Initialisierungsdaten
IPC	Industrie PC
ISO-Code	Spezieller Lochstreifencode, Lochanzahl pro Zeichen stets gerade
K1 ... K4	Kanal 1 bis Kanal 4
K-Bus	Kommunikationsbus
MD	Maschinendatum
MDA	Manual Data Automatic: Handeingabe
MKS	Maschinenkoordinatensystem
NCK	Numerical Control Kernel
OA	Open Architecture
OB	Organisationsbaustein in der PLC
OEM	Original Equipment Manufacturer: Hersteller, dessen Produkte unter fremden Firmennamen verkauft werden
OP	Operation Panel: Bedieneinrichtung
PI	Program Invocation: Programmierinstanz
PLC	Programmable Locic Control: Anpasssteuerung

SPS	Speicherprogrammierbare Steuerung
TCA	ToolChangeAbsolut
TCI	ToolChangeIntermediateLocation
TO	Tool Offset: Werkzeugkorrektur
TOA	Tool Offset Active: Kennzeichnung (Dateityp) für Werkzeugkorrekturen
TOOLGNT	ToolGroupNumber OfTools
TOOLGT	TOOLGroupToolNumber
USEKT	UserKindOfTools
VDI	Virtual device interface: Virtuelle Schnittstelle
V.24	Serielle Schnittstelle (Definitionder Austauschleitungen zwischen DEE und DÜE)
WKS	Werkstückkoordinatensystem
WZ	Werkzeug
WLK	Werkzeuglängenkorektur
WRK	Werkzeugradiuskorrektur
WZK	Werkzeugkorrektur
WZBF	Werkzeuggrundfunktion (Basisfunktion)
WZMO	Werkzeug-Überwachungsfunktion bzw. -Monitor
WZMG	Werkzeug-Magazinverwaltung
WZV	Werkzeugverwaltung

Index

$

$A_MONIFACT, 232
$A_MYMLN, 237
$A_MYMN, 237
$A_TOOLMLN, 231
$A_TOOLMN, 227
$A_USEDD, 245
$A_USEDND, 242
$A_USEDT, 243
$AC_MONMIN, 233
$AC_MSNUM, 260, 276
$AC_MTHNUM, 260, 276
$MC_TOOL_CHANGE_ERROR_MODE (MD22562), 24, 71
$MC_TOOL_CHANGE_MODE (MD22550), 22
$P_AD[n], 259
$P_ADT[n], 259
$P_D_BEFORE_SEARCH, 263
$P_DL_BEFORE_SEARCH, 264
$P_DLNO, 257
$P_ISTEST, 268
$P_MAGA, 255
$P_MAGDISL, 247
$P_MAGDISS, 247
$P_MAGHLT, 252
$P_MAGNA, 255
$P_MAGNA1, 255
$P_MAGNA2, 255
$P_MAGNDIS, 247
$P_MAGNH, 252
$P_MAGNHLT, 252
$P_MAGNREL, 249
$P_MAGNS, 248
$P_MAGREL, 249
$P_MAGS, 248
$P_MSNUM, 260, 276
$P_MTHNUM, 261, 277
$P_MTHNUM_BEFORE_SEARCH, 263
$P_MTHSDC, 262
$P_TC, 258
$P_TCANG[n], 258
$P_TCDIFF[n], 258, 259
$P_TH_OF_D, 262
$P_TOOL, 257
$P_TOOLD, 240
$P_TOOLL[n], 257, 258
$P_TOOLND, 231
$P_TOOLNDL, 241
$P_TOOLNG, 236
$P_TOOLNO, 256
$P_TOOLNT, 239
$P_TOOLP, 256, 257
$P_TOOLR, 258
$P_TOOLT, 239
$P_USEKT, $TC_TP11, 221
$P_VDITCP[x], 161
$TC_DPCx[t,D], 114
$TC_DPx[t,D], 112
$TC_MAMPx [n], 138
$TC_MAP10, 128
$TC_MAP3, 128
$TC_MAP8, 128
$TC_MAPCx[n], 129
$TC_MDPx[n,m], 135
$TC_MLSR[x,y], 145
$TC_MOPCx[t,D], 116
$TC_MOPx[t,D], 115
$TC_MPP1, 131
$TC_MPP5, 132
$TC_MPP6, 133
$TC_MPPCx[n,m], 133
$TC_MPPx[n,m], 129
$TC_MPTH[n,m], 134
$TC_MTPCx[t], 159
$TC_MTPPCx[t], 159
$TC_SCPx[t,D], 117
$TC_TP1 und $TC_TP2, 120
$TC_TP3 bis TP 6, 120
$TC_TP7, 121
$TC_TP8, 121
$TC_TPCx[t], 125
$TC_TPGx[t], 124
$TC_TPx[t], 118

A

Anwenderschnittstelle, 31
App "Siemens Industry Online Support", 12
Auftragszustand, 49

B

Beispiel
  Drehmaschine, 75
  Fräsmaschine, 87

Handwerkzeug wechseln (1), 71
Handwerkzeug wechseln (2), 72
Teileprogramm Drehen, 76
Teileprogramm Fräsen, 88
Belademagazin, 21
BTSS, 105
BTSS-Baustein AD, 160
BTSS-Baustein C/S, 177, 180
BTSS-Baustein MTUD, 159
BTSS-Baustein MTUP, 160
BTSS-Baustein TD, 119
BTSS-Baustein TG, 125
BTSS-Baustein TM, 127
BTSS-Baustein TMC, 138, 145
BTSS-Baustein TMV, 272
BTSS-Baustein TO, 112, 115
BTSS-Baustein TOE, TOET, 118
BTSS-Baustein TOS, 117
BTSS-Baustein TP, 130
BTSS-Baustein TPM, 135
BTSS-Baustein TS, 115
BTSS-Baustein TT, 134
BTSS-Baustein TU, 125
BTSS-Baustein TUM, 129
BTSS-Baustein TUP, 133
BTSS-Baustein TUS, 116

C

CHKDM, 163
Codetäger-Datenformate, 387
Codeträger-Beschreibungsdatei, 386
Codeträgerdaten
Datenkonvertierung, 379
Codeträgerdaten, Dialogdaten
Zuordnung, 387
CRCEDN, 274
CREACE, 274
CREATO, 274
CRTOCE, 274

D

Data-Matrix-Code, 13
Datenschutz-Grundverordnung, 13
DefToolDat.txt, 370, 391
DELDL, 167
DELECE, 274
DELETO, 274
DELMLOWNER, 219
DELMLRES, 218
DELMT, 171
DELT, 170
DELTC, 205
D-Nummern von Ersatzwerkzeugen, 163
DZERO, 167

E

Einrichtekorrekturen, 117
Einsatzortsabhängige Korrekturen, 117
Ende-Quittung, 54, 55
Ersatzwerkzeuge, 17

F

Feedback senden, 10

G

G–Code-Programm, 368
GETACTT, 185
GETACTTD, 164
GETEXET, 179
GETFREELOC, 211
GETSELT, 176
GETT, 172

H

Handwerkzeuge, 24

I

IDENTCONNECTIONCONFIGURATION, 376

K

Kennung
Kühlmittel, 355
Magazinplatzadapterparameter, 302
Magazinplatzparameter, 301
Multitoolparameter, 305
Multitoolplatzparameter, 306
OEM-Magazinplatzparameter, 355
OEM-Multitoolparameter, 354
OEM-Schneidenparameter, 354
OEM-Überwachungsparameter, 354
OEM-Werkzeugparameter, 353
Schleifparameter, 301

Schneidenparameter, 297
Überwachungsparameter, 300
Werkzeugparameter, 294
werkzeugspezifische Funktionen, 355
Winkelkopfparameter, 307
Winkelkopfplatzparameter, 309
Kettenmagazin, 22
Kühlmittel
Kennung, 355
zuordnen, 339

L

Listenparameter
ändern, 309
neu anlegen, 311

M

Magazin, 21, 60
Konfiguration, 84
Magazinliste, 21
Magazinplatzadapterparameter
Kennungen, 302
Magazinplatzparameter
Kennungen, 301
OEM-Kennung, 355
Maschinenfunktion, Texte festlegen, 341
MCIS-C Skriptsprache, 393
MD10717
$MN_T_NO_FCT_CYCLE_NAME, 30
MD22550
TOOL_CHANGE_MODE, 22
MD22562
TOOL_CHANGE_ERROR_MODE, 24, 71
MD52270
TM_FUNCTION_MASK, 29
Messen in JOG, 82
Mitteilung, asynchron, 54
MMCSEM, 274
Multitool, 23
Multitoolparameter
Kennungen, 305
OEM-Kennung, 354
Multitoolplatzparameter
Kennungen, 306
MVTOOL, 194
mySupport-Dokumentation, 10

N

NC-Variablen, 65
Nebenplatz, 23
NEWMT, 169
NEWT, 168

O

oem_sltmlistconfig.xml, 370
OpenSSL, 13
Ortsabhängige Korrekturen grob, 117

P

PI Dienst, 67
PI-Dienste, 274
PLC-Firmware, 19
PLC-Programm
Aufträge quittieren, 63
WZV, 19
POSM, 190
POSM (Multitool), 192
POSMT, 192
Product Support, 11
Programmierung der Werkzeuganwahl, 266
Programmierung von Daten
Werkzeug- und Schneidendaten, 109

Q

Quittung
Prozess, 53
Regeln, 64
Status, 55
synchron, 54
WZV, 36
Quittungsschritt, 62
Quittungsschritt-Tabelle, 51

R

RESETMON, 201
RESETMON (Multitool), 203
Revolvermagazin, 22
Rückmeldung, 37

S

Schleifparameter
Kennungen, 301
Schneidenanwahl nach Werkzeugwechsel, 267
Schneidenparameter
Kennungen, 297
OEM-Kennung, 354
SETDNO, 166
SETMS, 186
SETMTH, 187
SETPIECE, 174
SETTA (Multitool), 199
SETTIA, 199
SETTIA (Multitool), 201
ShopMill
werkzeugspezifische Funktionen, 339
Siemens Industry Online Support
App, 12
SINUMERIK, 9
Standardumfang, 9
systemconfiguration.ini, 370, 376
Systemvariablen, 105

T

T=Platz, automatische Werkzeug-Selektion, 224
T=Platznummer, 268
TCA, 206
TCI, 209
TDIdentCfg.xml, 377
tdiidentcfg.xml, 370
Technical Support, 12
TMCRTC, 274
TMCRTO, 274
TMFDPL, 274
TMFPBP, 274
TMGETT, 272, 274
TMMVTL, 274
TMMVTL (PI Dienst), 67
TMPCIT, 274
TMPOSM, 274
TMRASS, 274
Tool Ident Connection
Konfiguration, 377
PLC-Anbindung, 379
TOOLGNT, 224
TOOLGT, 224
ToolSize.mcc, 407
toolSpec.xml, 370
ToolSpec.xml
Konfiguration, 402
Total-Quittung, 55
Training, 12
Transferschritt, 60
Transferschritt-Tabelle
konstante, 50
variable, 50, 51
TRANSMIT-Befehl, 368
TSEARC, 272, 274

U

Überwachungsparameter
Kennungen, 300
OEM-Kennung, 354
Unterprogrammersetzungstechnik, 264

V

Variable
Platztyp, 65
Platzzustand, 65
T-Nummer, 66

W

Webseiten Dritter, 9
Werkzeugaufruf, 82
Werkzeugbezogene Daten, 118
Werkzeugliste, 21
Listenansichten, 290
Werkzeugparameter
Kennungen, 294
OEM-Kennung, 353
Werkzeugspezifische Funktionen
Kennung, 355
Werkzeugträger, 366
Werkzeugtypen
Bohrer, 314
Drehwerkzeuge, 314
Fräser, 313
Schleifwerkzeuge, 314
Sonderwerkzeuge, 315
Werkzeugverwaltung, 17
PLC,TRANSLINE 2000, 282
Satz-Splitting, 92
Technologie Drehen, 282
Technologie Drehen auf Fräsmaschine, 282
Technologie Fräsen, 282
Technologie Schleifen, 282

Werkzeugwechsel, 40, 92, 93
    Ende-Quittung, 41
    Programmierung, 266
Werkzeugwechselgrund konfigurieren, 342
Winkelkopfparameter
    Kennungen, 307
Winkelkopfplatzparameter
    Kennungen, 309
wkonvert.mcc, 370, 405
wkonvert.mcx, 370
wkonvert.txt
    Beschreibungsdatei, 384
Wkonvert-Wizard
    Installation, 371

Z

Zwischenquittung, 54, 55
Zwischenspeicher, 21