SIEMENS

Application description • 03/2014

Use of the Write command

SINUMERIK 828D, 840D sl

http://support.automation.siemens.com/WW/view/en/85090492

Warranty and liability

Note

The application examples are non-binding and do not claim to be complete in terms of configuration and equipment or to take account of any other contingencies. The application examples do not represent specific customer solutions; they are intended only as support for typical tasks. The user has sole responsibility for ensuring correct operation of the products described. These application examples do not exempt the user from their due diligence obligation with regard to application, installation, operation and maintenance. By using these application examples, you agree that Siemens cannot be made liable for possible damage beyond the mentioned liability clause. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any differences between the suggestions made in these application examples and other Siemens publications such as catalogs, the contents of the other document(s) take priority.

We do not provide a warranty for any of the information contained in this document.

We accept no liability for any damage or loss caused by the examples, information, programs, planning data or performance data described in this application example, irrespective of the legal basis for claims arising from such damage or loss, unless liability is mandatory. For example, according to the product liability law, in cases of malfeasance, gross negligence, due to endangerment of life, body or health, due to assumption of a guarantee for a product's characteristics of state, due to malicious concealment of a defect or due to violation of basic contractual obligations. Any compensation for violation of basic contractual obligations, however, shall be limited to the foreseeable damage or loss which is typically envisaged in contracts unless there has been gross negligence or unless liability is mandatory due to endangerment of life, body, or health. Any change to the burden of proof to your disadvantage is not covered hereby.

Any form of duplication of these application examples or excerpts hereof is not permitted without the express consent of Siemens Industry Sector.

Security notes

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, devices, and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

To ensure that Siemens products and solutions are operated securely, suitable preventive measures (e.g. cell protection concept) must be taken and each component must be integrated into a state-of-the-art holistic industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity

To receive information about product updates on a regular basis, register for our product newsletter. For more information, visit http://support.automation.siemens.com

Table of contents

Warranty and liability 2

1 Task 4

1.1 Overview 4

2 "WRITE" high-level language element 5

2.1 Function 5
2.2 Requirements 5
2.3 Code syntax 5
2.3.1 Errors that occur when using the "WRITE" high-level language element 6
2.4 Typical use of the Write command 7
2.5 Further examples 10
2.5.1 Write command into the passive file system without absolute path data 10
2.5.2 Write command into the passive file system with absolute path data 10

3 Notes 11

3.1 File 11
3.1.1 File name 11

4 Contact 12

5 History 12

1 Task

1.1 Overview

Production with CNC machine tools requires a large quantity of measurement data and parameters to guarantee the quality and verify the production in accordance with the legal requirements. This applies particularly to the medical technology (production of implants) and aircraft construction sectors. Both sectors are confronted with very high demands on quality and the accountability for the manufactured parts over decades.

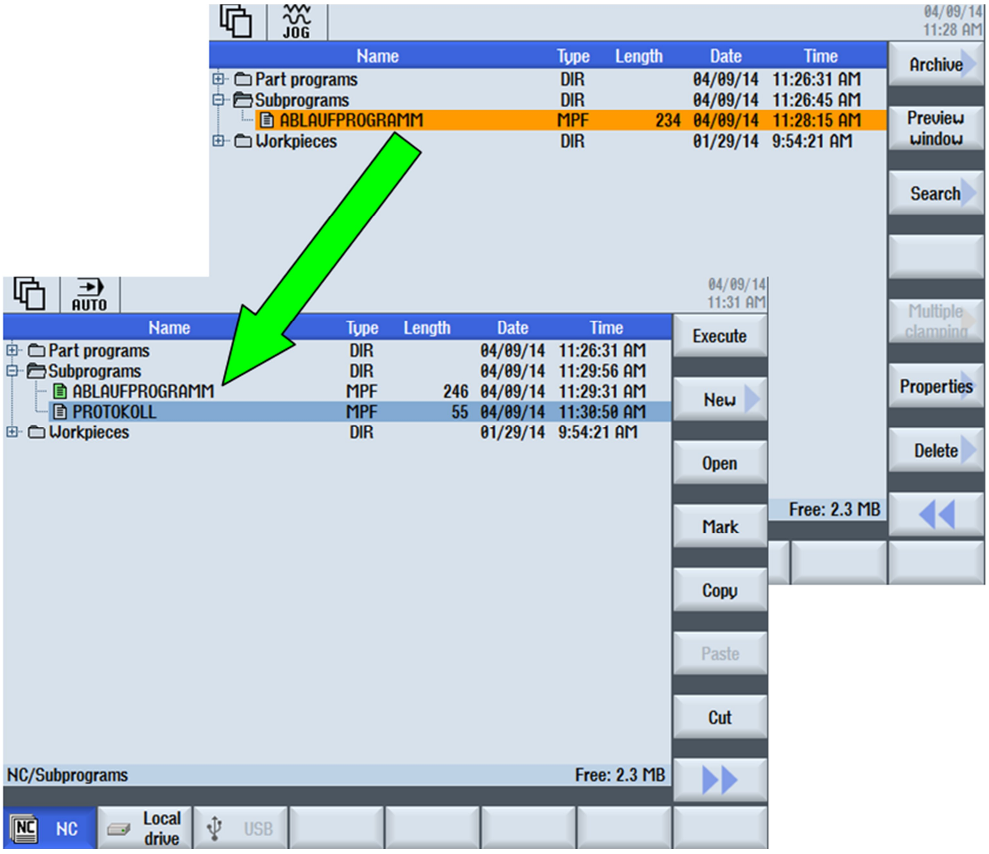
In order to prevent the loss of machining parameters during the production process, the "WRITE" high-level language element can be used to archive the machining parameters, for example, to safeguard the production. The archiving is performed through the automatic creation of a log file in the NC of the machine.

The following examples describe the safe and clear use of the "WRITE" high-level language element for the user.

Overview of the creation of a log file

The following diagram provides an overview of the method of operation of the "WRITE" high-level language element.

Fig. 1-1 Automatic creation of a log file using the "WRITE" high-level language element



Name	Type	Length	Date	Time
Part programs	DIR		04/09/14	11:26:31 AM
Subprograms	DIR		04/09/14	11:26:45 AM
ABLAUFPROGRAMM	MPF	234	04/09/14	11:28:15 AM
Workpieces	DIR		01/29/14	9:54:21 AM

Name	Type	Length	Date	Time
Part programs	DIR		04/09/14	11:26:31 AM
Subprograms	DIR		04/09/14	11:29:56 AM
ABLAUFPROGRAMM	MPF	246	04/09/14	11:29:31 AM
PROTOKOLL	MPF	55	04/09/14	11:30:50 AM
Workpieces	DIR		01/29/14	9:54:21 AM

NC/Subprograms Free: 2.3 MB

2 "WRITE" high-level language element

2.1 Function

The task of the "WRITE" high-level language element is to write blocks or data from NC programs to a log file. This is used, for example, to safeguard the production or for logging the measurement data. During the currently running NC program, the "WRITE" high-level language element reads out the specified data and writes it automatically to the log file of the machine NC. If a log file does not exist, it is generated in MPF format. An existing log file is automatically appended during the program execution. The created files are freely accessible and can be read, changed and deleted by all users of the machine.

2.2 Requirements

The "WRITE" high-level language element enables users to store blocks from part programs in files in the passive file system. The length of the stored log files is limited. If the maximum length is exceeded, an error message (Fig. 2-1) is output and the block or data is not saved. If there is sufficient free memory, a new file can be created. The length of the log file can be changed in the following machine data:

- MD11420 $MN_LEN_PROTOCOL_FILE

2.3 Code syntax

The WRITE command is structured in accordance with the following template:

```
DEF INT <error>

WRITE(<error>,“<file name“,“<block/data>“)
```

Table 2-1 Explanation of the existing parts of the WRITE command

Command	Description
WRITE	Command for appending a block or data to the end of a specified file.
<error>	Variable for returning the error value.
<file name>	The name of the file in the passive file system in which the specified block or specified data is to be added. (Additional explanations can be found in Section 3.1.1, File name.)
<block/data>:	The block or data to be added to the specified file.

2.3.1 Errors that occur when using the "WRITE" high-level language element

The <ERROR> variable in the syntax code of the "WRITE" high-level language element does not generate an NCK alarm. The variable is used for the evaluation in the program for a specific reaction.

The following file identifiers are permitted:

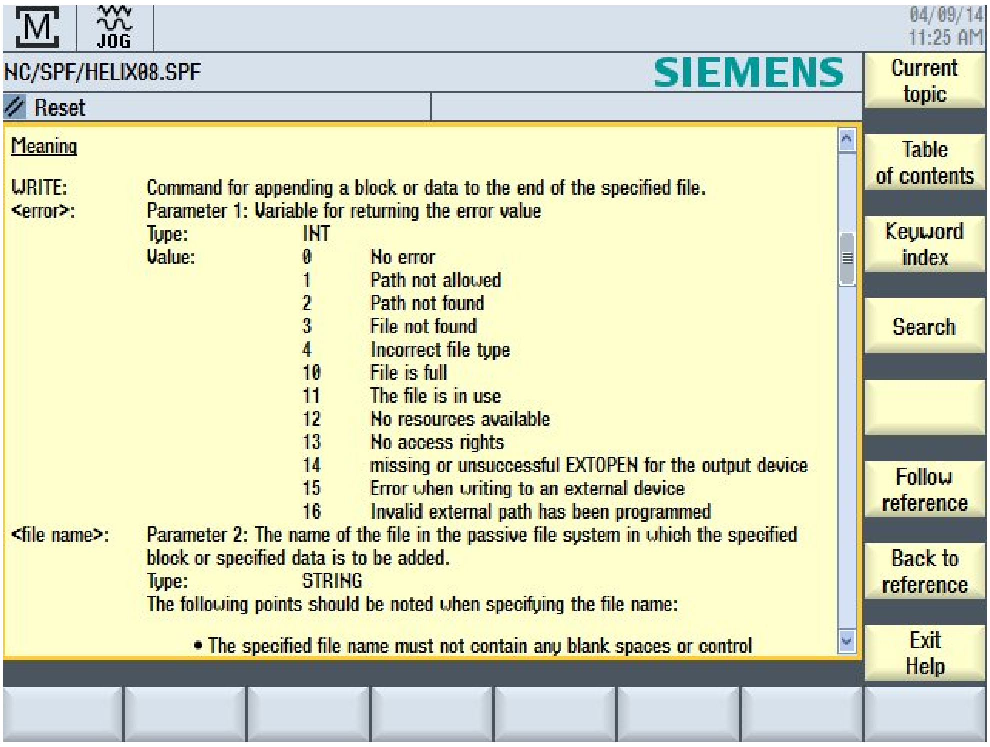
- "_MPF"
- "_SPF"
- "_CYC"

The following characters are permitted in file names:

- 1 to 0
- A to Z
- Underscore "_"

The decimal point and other "special characters" are not permitted in file names. Values from 1 to 16 to read out possible errors are listed in the following.

Fig. 2-1 Errors that occur with the Write command in the help menu of the machine



NC/SPF/HELIX08.SPF

Reset

Meaning

```
WRITE:       Command for appending a block or data to the end of the specified file.
<error>:     Parameter 1: Variable for returning the error value
             Type:        INT
             Value:       0     No error
                          1     Path not allowed
                          2     Path not found
                          3     File not found
                          4     Incorrect file type
                          10    File is full
                          11    The file is in use
                          12    No resources available
                          13    No access rights
                          14    missing or unsuccessful EXTOPEN for the output device
                          15    Error when writing to an external device
                          16    Invalid external path has been programmed
<file name>: Parameter 2: The name of the file in the passive file system in which the specified
             block or specified data is to be added.
             Type:        STRING
             The following points should be noted when specifying the file name:

                 • The specified file name must not contain any blank spaces or control
```

Further information on the "WRITE" high-level language element to be noted can be found in Section 3, Notes.

2.4 Typical use of the Write command

The SINUMERIK control provides numerous measuring cycles for the measurement of workpieces and tools not only during setup mode, but also for measurement during the machining process. The process measurement is used for the quality assurance during production and for the logging of measured values.

Setpoint and actual value of a single-point measurement in the current example are read from the existing "ABLAUFPROGRAMM.SPF" file and a "PROTOKOLL.MPF" log file is written (Fig. 2-2 and Fig.2-3).

Note Press the "HELP" key to find a list of the OVR variables and their meaning at "Result parameters".

Content of the ABLAUFPROGRAMM.SPF file

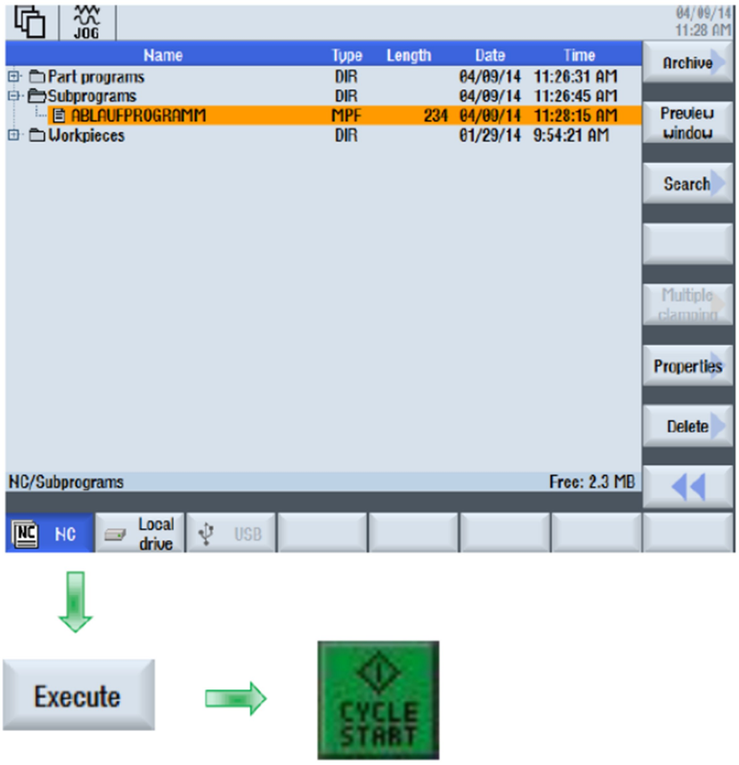
```
DEF INT _ERROR

G17 G54 G90

T="3D_TASTER"
M6

G0 X23 Y0
G0 Z5

CYCLE978(100,10008,,1,0,20,5,3,2,1,"",,0,1.01,1.01,-1.01,0.34,1,0,,1,1)

WRITE(_ERROR,"_N_LOG","SETPOINT Z="<<_OVR[3])
WRITE(_ERROR,"_N_LOG","ACTUAL VALUE Z="<<_OVR[7])
WRITE(_ERROR,"_N_LOG","DIFFERENCE="<<_OVR[19])
M30
```

Table 2-2 Explanation of the programming example

No.	Programming command	Description
1.	DEF INT_ERROR	Definition of error variables
2.	G17 G54 G90	Starting point
3.	T="3D_PROBE"	Selecting the probe
4.	M6	Loading the 3D probe
5.	G0 X23 Y0 G0 Z5	Starting point of the measurement
6.	CYCLE978(100,10008,,1,0,20,5,3,2,1,"",,0,1.01,1.01,-1.01,0.34,1,0,,1,1)	Measuring process
7.	WRITE(_ERROR,"_N_LOG","SETPOINT Z="<<_OVR[3])	Writing of the setpoint in the log file
8.	WRITE(_ERROR,"_N_LOG","ACTUAL VALUE Z="<<_OVR[7])	Writing of the actual value in the log file
9.	WRITE(_ERROR,"_N_LOG","DIFFERENCE="<<_OVR[19])	Writing of the difference between the actual value and

No.	Programming command	Description
		setpoint in the log file
10.	OVR[3])	Offset vector I1$TC_CARR3[n], Z component
11.	OVR[7])	Rotary axis vector V1$TC_CARR7[n], X component
12.	OVR[19])	Offset vector I14$TC_CARR19[n], Y component
13.	M30	End of program

Fig. 2-2 Created sequential program file in the NC



Name	Type	Length	Date	Time
Part programs	DIR		04/09/14	11:26:31 AM
Subprograms	DIR		04/09/14	11:26:45 AM
ABLAUFPROGRAMM	MPF	234	04/09/14	11:28:15 AM
Workpieces	DIR		01/29/14	9:54:21 AM

NC/Subprograms

Free: 2.3 MB

04/09/14
11:28 AM

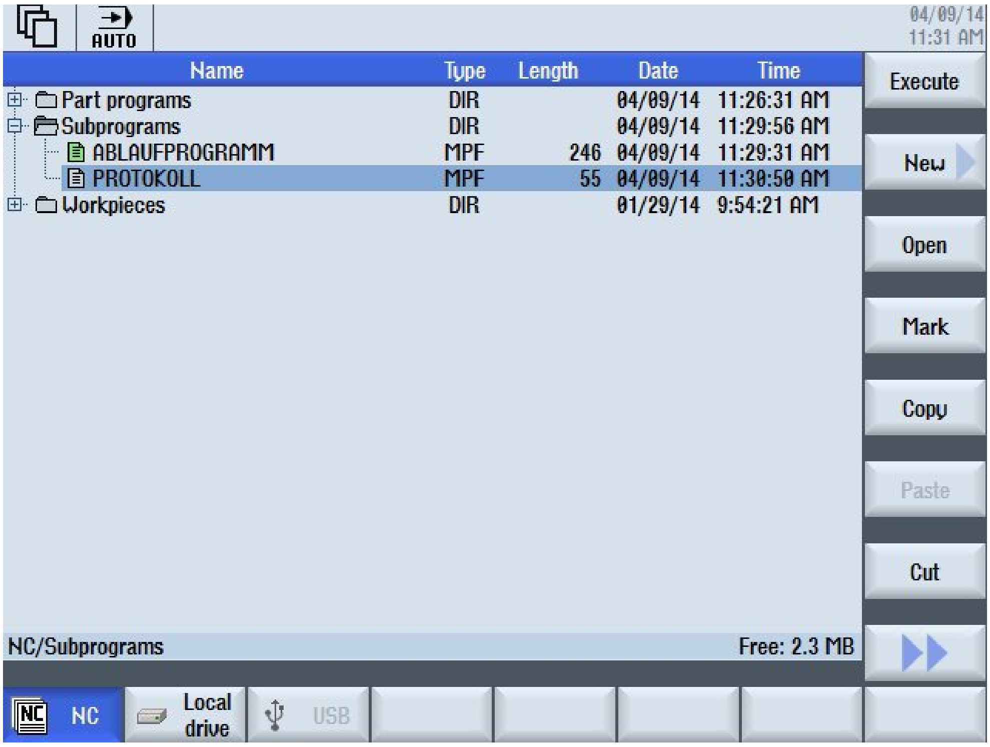
After "Selection" and processing of the sequential program by pressing the "CYCLE START" button (Fig.2-2), the log file (PROTOKOLL.MPF) was created automatically in the same folder. The log file has the following content:

Content of the "PROTOKOLL.MPF" log file

```
SETPOINT Z=0
ACTUAL VALUE Z=0.01300811768
DIFFERENCE=0.01300811768
```

Note The "CYCLE START" button can be pressed here without problem because this does not result in an axis movement of the machine. Only the sequential program is generated.

Fig. 2-3 Sequential program with created log file in the NC



Name	Type	Length	Date	Time
Part programs	DIR		04/09/14	11:26:31 AM
Subprograms	DIR		04/09/14	11:29:56 AM
ABLAUFPROGRAMM	MPF	246	04/09/14	11:29:31 AM
PROTOKOLL	MPF	55	04/09/14	11:30:50 AM
Workpieces	DIR		01/29/14	9:54:21 AM

2.5 Further examples

The following examples show the options and rules for using the Write command. Explanations of the individual programming steps are listed at the end of the section.

2.5.1 Write command into the passive file system without absolute path data

```
N10 DEF INT ERROR
N20 WRITE(ERROR,"PROT","LOG FROM 7.2.97")
N30 IF ERROR
N40 MSG ("Error with WRITE command")
M50 M0
M60 ENDIF
```

2.5.2 Write command into the passive file system with absolute path data

```
….
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","LOG FROM 7.2.97")
…
```

Table 2-3 Explanation of the programming examples

No.	Programming command	Description
1.	DEF INT_ERROR	Definition of error variables
2.	WRITE(ERROR,"PROT","LOG FROM 7.2.97")	Write the "LOG FROM 7.2.97" text to the _N_PROT_MPF file
3.	IF ERROR	Error evaluation
4.	MSG ("Error with WRITE command")	Message when an error occurs
6.	M30	End of program

3 Notes

3.1 File

- If a file be written by means of the WRITE command does not exist in the NC, it is newly created.
- The storage location is the static NC memory. In the case of SINUMERIK 840D sl this is the CompactFlash Card. Compared to the SINUMERIK 840D, this increases the runtime of the WRITE command by approx. 75 ms.
- If a file with the same name exists on the hard disk, it is overwritten after the file is closed (in the NC). The name can be changed in the NC in the "Services" operating area using the "Properties" softkey.

3.1.1 File name

The following points should be noted when specifying the file name:

- The specified file name must not contain any blanks or control characters (characters with ASCII code <= 32), otherwise the "WRITE" high-level element outputs error code 1 "Path not allowed".
- The file name can be specified with path data and file identifier.
  - Path data must be absolute, i.e. start with "/". If a path is not specified, the file is saved to the current directory.
  - If the file name does not contain a domain identifier ("_N_"), it is added accordingly. If the fourth-last character of the file name is an underscore "_", the next three characters are interpreted as the file identifier. In order to be able to use the same file name for all file commands, e.g. via a STRING type variable, only the _SPF and _MPF file identifiers must be used. If there is no "_MPF" or "_SPF" identifier, the file name is automatically augmented with _MPF.
- The file name length can be up to 32 bytes, the path length up to 128 bytes.

4 Contact

Siemens AG

Industry Sector
I DT MC MTS APC2
Frauenauracher Strasse 80
D - 91056 Erlangen, Germany
E-mail: MC-MTS-APC-Tech-Team.i-dt@siemens.com

5 History

Table 5-1

Version	Date	Revision
V1.0	03/2014	First Edition