

**SIEMENS**

*Ingenuity for life*

24/7

Industry Online Support

Home

# Projektierung von Meldungen und Alarmen in WinCC (TIA Portal) - Erweiterung mit S7-1200/S7-1500

WinCC V14 SP1

<https://support.industry.siemens.com/cs/ww/de/view/62121503>

Siemens  
Industry  
Online  
Support



# Rechtliche Hinweise

## Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.

Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

## Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.

Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

## Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Ergänzend gelten die Siemens Nutzungsbedingungen (<https://support.industry.siemens.com>).

## Securityhinweise

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 nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter: <https://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates 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: <http://www.siemens.com/industrialsecurity>.

# Inhaltsverzeichnis

<b>Rechtliche Hinweise .....</b>	<b>2</b>
<b>1 Einführung.....</b>	<b>4</b>
1.1 Überblick.....	4
1.2 Funktionsweise.....	5
1.3 Abgrenzung .....	5
1.4 Verwendete Komponenten.....	6
<b>2 Projektierung von Meldungen in WinCC Basic/Comfort/Advanced .....</b>	<b>7</b>
2.1 Projektierung von benutzerdefinierten Meldungen .....	7
2.2 Projektierung von systemdefinierten Meldungen .....	8
2.2.1 Projektierung von Systemmeldungen (S7-1200, S7-1500).....	8
2.2.2 Projektierung von CPU-Systemdiagnosemeldungen (S7-1500).....	12
2.3 Projektierung von Steuerungsmeldungen .....	15
2.3.1 Projektierung von Program_Alarm (S7-1500) .....	15
2.3.2 Projektierung von Get_AlarmState (S7-1500) .....	22
2.3.3 Projektierung von Gen_UsrMsg (S7-1200, S7-1500) .....	26
2.4 Einsatz von Meldeklassen.....	31
2.5 Einsatz von Meldegruppen.....	32
2.6 Das Quittierkonzept.....	33
<b>3 Projektierung von Meldungen in WinCC Professional .....</b>	<b>35</b>
3.1 Projektierung von benutzerdefinierten Meldungen .....	35
3.2 Projektierung von systemdefinierten Meldungen .....	35
3.2.1 Projektierung von Systemmeldungen (S7-1200, S7-1500).....	35
3.2.2 Projektierung von CPU-Systemdiagnosemeldungen (S7-1500).....	41
3.3 Projektierung von Steuerungsmeldungen .....	43
3.3.1 Projektierung von Program_Alarm (S7-1500) .....	43
3.3.2 Projektierung von Get_AlarmState (S7-1500) .....	45
3.3.3 Projektierung von Gen_UsrMsg (S7-1200, S7-1500) .....	45
3.4 Einsatz von Meldeklassen.....	46
3.5 Einsatz von Meldegruppen.....	47
3.6 Das Quittierkonzept.....	48
3.6.1 Meldung ohne Quittierung .....	48
3.6.2 Meldung mit Einfachquittierung .....	48
3.6.3 Meldung mit Zweifachquittierung .....	49
3.6.4 Allgemeine Informationen zu Quittierkonzepten .....	51
<b>4 Wissenswertes .....</b>	<b>52</b>
4.1 Grundlagen.....	52
4.1.1 Übersicht über die Meldeverfahren .....	52
4.1.2 Benutzerdefinierte Meldungen .....	53
4.1.3 Systemdefinierte Meldungen.....	54
4.1.4 Verfügbarkeit der Meldeverfahren.....	55
<b>5 Anhang.....</b>	<b>56</b>
5.1 Service und Support.....	56
5.2 Links und Literatur .....	57
5.3 Änderungsdokumentation .....	58



# 1 Einführung

## 1.1 Überblick

### Motivation

In dieser Dokumentation wird die Projektierung von Meldungen und Alarmen in WinCC (TIA Portal) im Zusammenhang mit den Steuerungsfamilien S7-1200/S7-1500 beschrieben.

### Applikative Umsetzung

Ausgangsbasis hierfür ist die Dokumentation „Projektierung von Meldungen und Alarmen in WinCC (TIA Portal)“ in Zusammenhang mit den Steuerungsfamilien S7-300/S7-400 dieses Anwendungsbeispiels.

Bei den Grundlagen sowie in ausgewählten Kapiteln wird sich explizit auf diese Dokumentation bezogen, da es sich um den gleichen Inhalt und Projektierungsumfang handelt.

Das Anwendungsbeispiel stellt eine Ergänzung zum Handbuch und der Online-Hilfe des TIA Portals dar.

### Vorausgesetzte Kenntnisse

Grundlegende Kenntnisse zu WinCC (TIA Portal) werden vorausgesetzt.

### Hinweis

Grundlagen zu WinCC werden im SITRAIN-Kurs „SIMATIC WinCC maschinennah im TIA Portal“ und „SIMATIC WinCC SCADA im TIA Portal“ vermittelt.

- [SIMATIC WinCC maschinennah im TIA Portal \(de\)](#)
- [SIMATIC WinCC on the machine level in the TIA Portal \(en\)](#)
  
- [SIMATIC WinCC SCADA im TIA Portal \(de\)](#)
- [SIMATIC WinCC SCADA in the TIA Portal \(en\)](#)

Grundlagen zu STEP 7 werden im SITRAIN-Kurs „SIMATIC Programmieren 1 im TIA Portal“ und „SIMATIC S7-1200 Basiskurs“ vermittelt.

- [SIMATIC Programmieren 1 im TIA Portal \(de\)](#)
- [SIMATIC programming 1 in the TIA Portal \(en\)](#)
  
- [SIMATIC S7-1200 Basiskurs \(de\)](#)
- [SIMATIC S7-1200 basic course \(en\)](#)

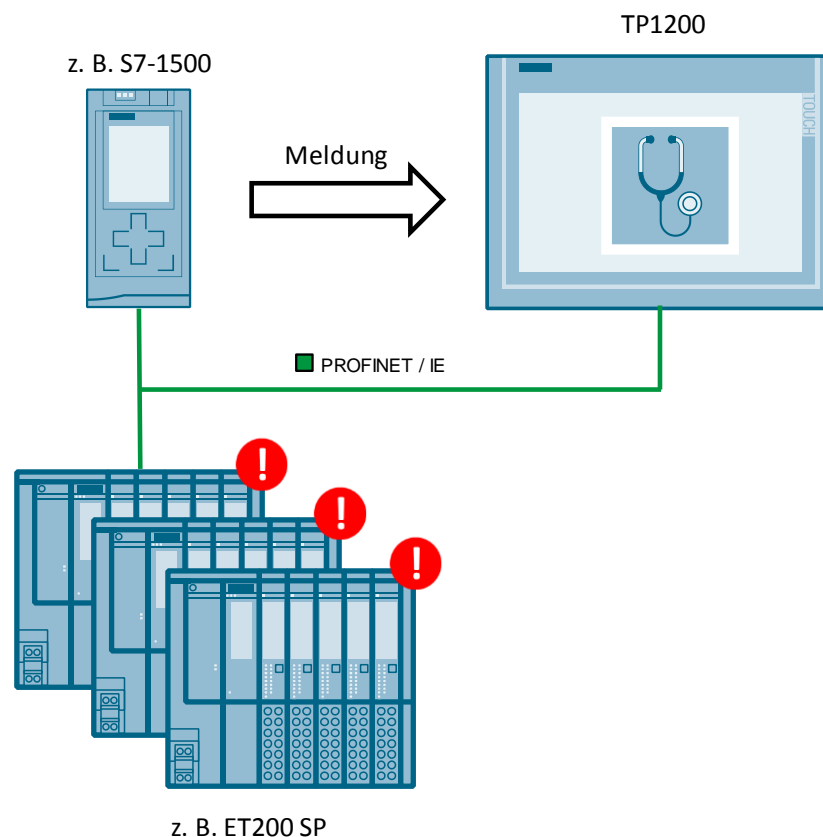
## 1.2 Funktionsweise

Im laufenden Prozess einer Anlage ist es unerlässlich, Informationen über Betriebszustände, Störungen und einzelne Prozesse visuell an einem HMI Bediengerät auszugeben. Das WinCC (TIA Portal) Meldesystem bietet in Zusammenhang mit den SIMATIC Controllern S7-1200/S7-1500 für jede dieser Informationen ein entsprechendes Meldeverfahren an.

In dieser Applikation erhalten Sie:

- Einen Überblick über die verschiedenen Meldeverfahren in WinCC
- Unterstützung bei der Auswahl des richtigen Meldeverfahrens für Ihren Anwendungsfall und Ihre vorhandene Hardware (S7-1200/S7-1500)
- Detaillierte Projektierungsanweisungen der verschiedenen Arten von Meldungen in WinCC (TIA Portal) und STEP 7 (TIA Portal)

Abbildung 1-1



## 1.3 Abgrenzung

In diesem Anwendungsbeispiel werden Meldungen und Alarmer in WinCC (TIA Portal) in Zusammenhang mit den Steuerungsfamilien S7-1200/S7-1500 erläutert.

Weiterführende Informationen zu den Meldeverfahren mit den Steuerungen S7-300/S7-400 finden Sie im zweiten PDF-Dokument zu diesem Anwendungsbeispiel.

## 1.4 Verwendete Komponenten

Die Applikation wurde mit nachfolgenden Komponenten erstellt:

### Hardware-Komponenten

Tabelle 1-1

Komponente	Anz.	Artikelnummer	Hinweis
SIMATIC CPU 1513-1 PN	1	6ES7513-1AL01-0AB0	Alternativ kann auch jede andere CPU aus der Familie S7-1500 verwendet werden, es muss lediglich ein Gerätetausch in der Projektierung durchgeführt werden.
Memory Card 24 MByte	1	6ES7954-8LF02-0AA0	
SIMATIC HMI TP1200 Comfort	1	6AV2124-0MC01-0AX0	Alternativ kann auch jedes andere Comfort Panel verwendet werden (Gerätetausch notwendig).

**Hinweis** Beim Einsatz einer CPU aus der Familie S7-1200 ist der Umfang der möglichen Meldungsarten gegenüber einer CPU aus der Familie S7-1500 stark eingeschränkt.

### Software-Komponenten

Tabelle 1-2

Komponente	Anz.	Artikelnummer
STEP 7 Professional V14 SP1 (TIA Portal)	1	6ES7822-1..04-..
WinCC Professional V14 SP1 (TIA Portal)	1	6AV210-....4-0
WinCC Runtime Professional V14 SP1 (TIA Portal)	1	6AV2105-....4-0

## 2 Projektierung von Meldungen in WinCC Basic/Comfort/Advanced

In diesem Abschnitt wird als Ergänzung zum Handbuch (Kapitel [„Mit Meldungen arbeiten“](#)) auf die Projektierung von systemdefinierten Meldungen und Steuerungsmeldungen eingegangen. Die Projektierung von benutzerdefinierten Meldungen ist mit aufgeführt.

### Benötigte Softwarekomponenten

- WinCC Basic/Comfort/Advanced V14 SP1 (TIA Portal)
- STEP 7 Professional V14 SP1 (TIA Portal)

### Voraussetzung

Es existiert bereits ein WinCC (TIA Portal) Projekt mit einer angelegten Verbindung zwischen dem Bediengerät und der Steuerung.

### 2.1 Projektierung von benutzerdefinierten Meldungen

Die Projektierung von benutzerdefinierten Meldungen (Bitmeldungen und Analogmeldungen) ist bei WinCC Basic/Comfort/Advanced unabhängig von der eingesetzten Steuerung. Eine ausführliche Beschreibung der Projektierung von benutzerdefinierten Meldungen finden Sie im Anwendungsbeispiel [„Projektierung von Meldungen und Alarmer in WinCC \(TIA Portal\)“](#). Außerdem finden Sie im Systemhandbuch von WinCC Advanced V14 SP1 unter den Kapiteln [„Bitmeldungen projektieren“](#) und [„Analogmeldungen projektieren“](#) ebenfalls eine ausführliche Beschreibung.

## 2.2 Projektierung von systemdefinierten Meldungen

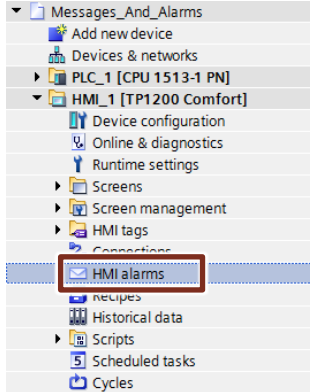
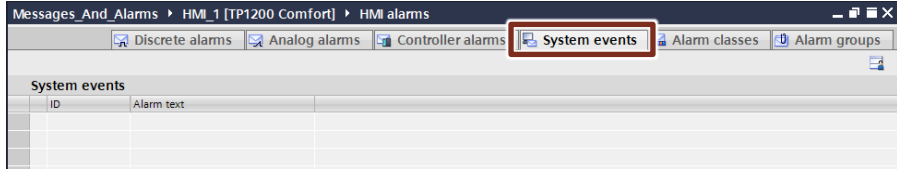
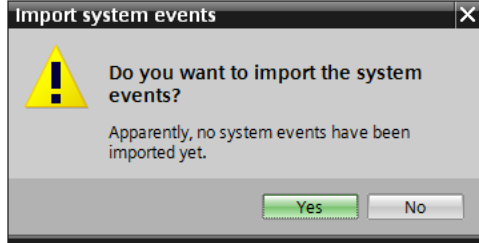
### 2.2.1 Projektierung von Systemmeldungen (S7-1200, S7-1500)

Systemmeldungen sind im HMI standardmäßig für verschiedene Sprachen hinterlegt. Wenn Sie die Systemmeldungen in weitere Sprachen übersetzen wollen, müssen die Texte zuerst in das Projekt importiert werden.

#### Importieren von Systemmeldungen

Das Importieren der Systemmeldungen ist nur bei Projekten notwendig, die neu erstellt wurden, bzw. wenn die Systemmeldungen noch nicht importiert wurden.

Tabelle 2-1

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihres angelegten Bediengerätes die „HMI-Meldungen“.</p> 
2.	<p>Öffnen Sie die Registerkarte „Systemmeldungen“.</p> 
3.	<p>Bestätigen Sie den nachfolgenden Dialog, um die „Systemmeldungen“ zu importieren.</p> 
4.	<p>Das Importieren der Systemmeldungen ist hiermit abgeschlossen.</p>



### Anzeige der Systemmeldungen in Runtime auf dem Bediengerät

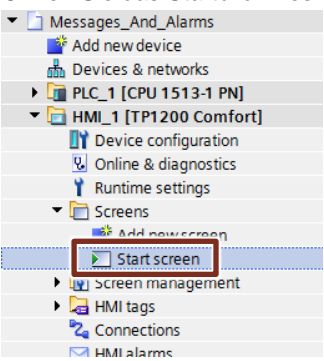
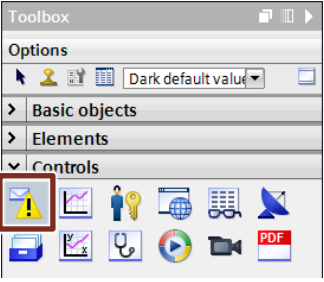
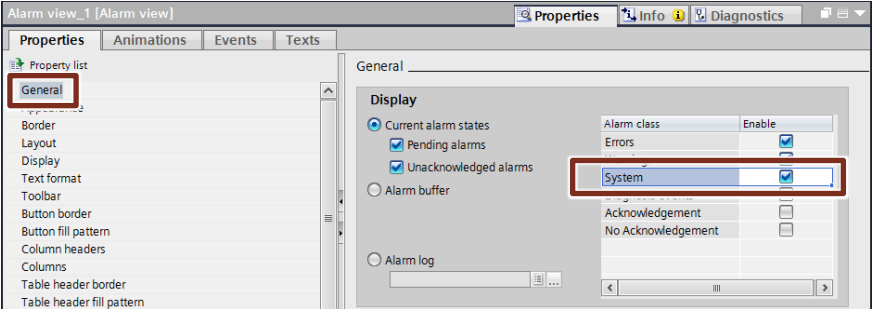
In der nachfolgenden Tabelle finden Sie die Anleitung zur Projektierung einer Meldeanzeige. Hierbei handelt es sich um ein statisches Element (immer sichtbar), welches Sie fest in ein Bild projektieren.

#### Hinweis

Neben einer statischen Meldeanzeige, besteht die Möglichkeit, sich Meldungen dynamisch mittels eines Meldefensters ausgeben zu lassen. Dieses wird im „Globalen Bild“ des HMI's projektiert und erscheint nur, wenn eine Meldung auftritt.

Weitere Informationen finden Sie im Handbuch von WinCC unter dem Kapitel [„Meldefenster projektieren“](#).

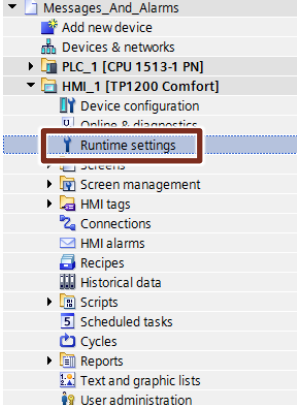
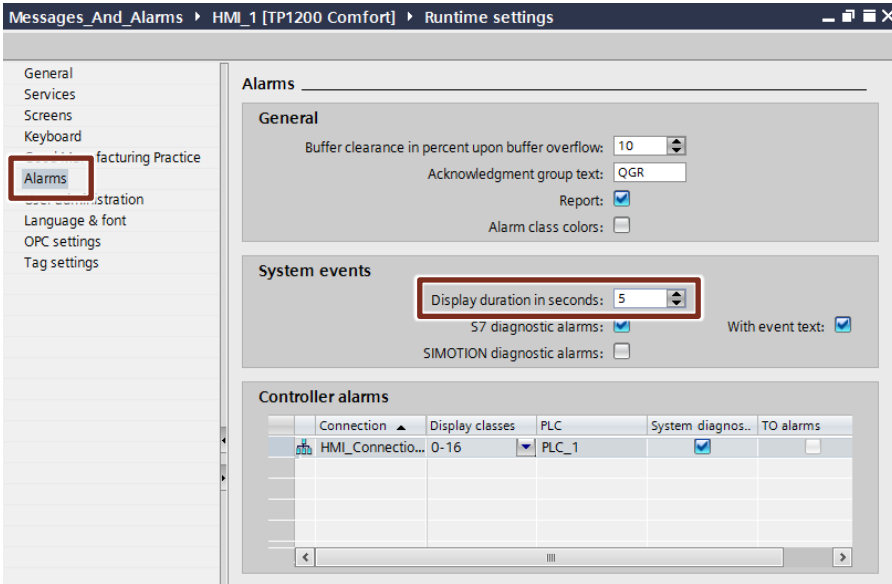
Tabelle 2-2

Nr.	Aktion
1.	<p>Öffnen Sie das Startbild ihres Bediengerätes.</p> 
2.	<p>Wählen Sie aus der Task Card „Werkzeuge“ unter den Controls die „Meldeanzeige“ aus</p> 
3.	<p>Ziehen Sie die „Meldeanzeige“ per Drag&amp;Drop in Ihr Startbild und passen Sie diese auf die gewünschte Größe an.</p>
4.	<p>Wählen Sie unter den Eigenschaften der Meldeanzeige „System“ an, um Systemmeldungen auf dem Bediengerät zur Anzeige zu bringen.</p> 

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Basic/Comfort/Advanced unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

**Anzeigedauer für Systemmeldungen definieren**

Tabelle 2-3

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihres angelegten Bediengerätes die „Runtime-Einstellungen“.</p> 
2.	<p>Öffnen Sie das Menü „Meldungen“.</p>  <p>Geben Sie unter „Systemmeldungen &gt; Anzeigedauer in Sekunden“ eine Anzeigedauer für die Systemmeldungen auf dem Bediengerät ein.</p> <p><b>Hinweis</b> Sollen die Systemmeldungen permanent anstehen, wählen Sie eine Anzeigedauer von „0“ Sekunden.</p>
3.	<p>Die Einstellungen zur Anzeigedauer von Systemmeldungen sind damit abgeschlossen.</p>

### Meldetexte für Systemmeldungen ändern

Die Meldetexte der Systemmeldungen können bei Bedarf geändert bzw. angepasst werden. Die dazugehörigen Meldenummern sind nicht änderbar.

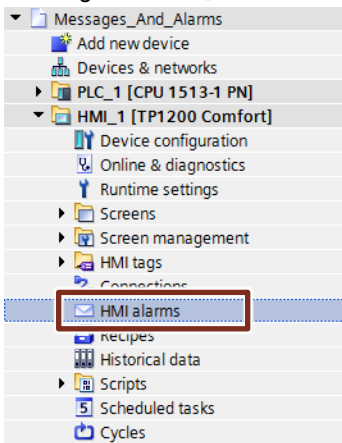
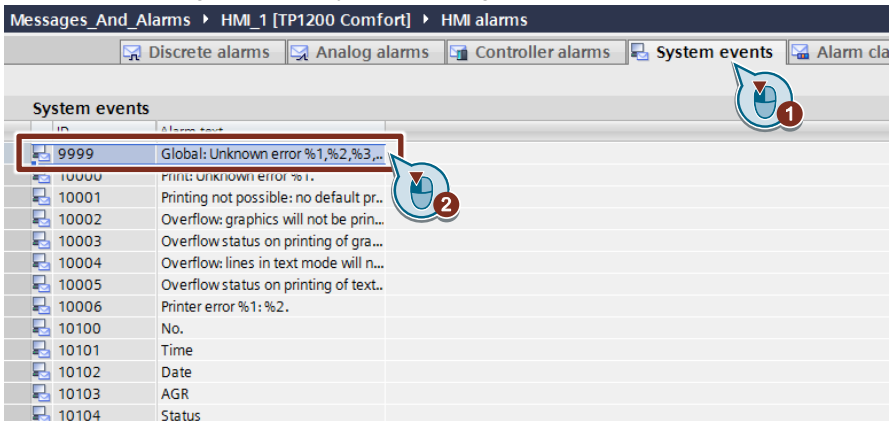
#### Hinweis

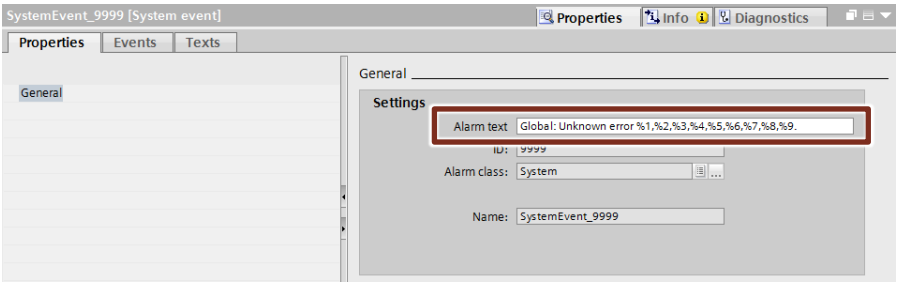
Bei der Änderung von Systemmeldungen erfolgt eine Veränderung von klar definierten Meldungen, wodurch es unter Umständen zu Fehlinterpretationen kommen kann.

Die importierten Meldungen sind Bestandteile des Handbuches und der Online-Hilfe.

Eine Änderung hat zur Folge, dass die geänderten Meldungen nicht mehr mit der Dokumentation übereinstimmen.

Tabelle 2-4

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihres angelegten Bediengerätes die „HMI-Meldungen“.</p> 
2.	<p>Öffnen Sie die Registerkarte „Systemmeldungen“.</p>  <p>Markieren Sie die Systemmeldung, deren Meldetext geändert werden soll.</p>

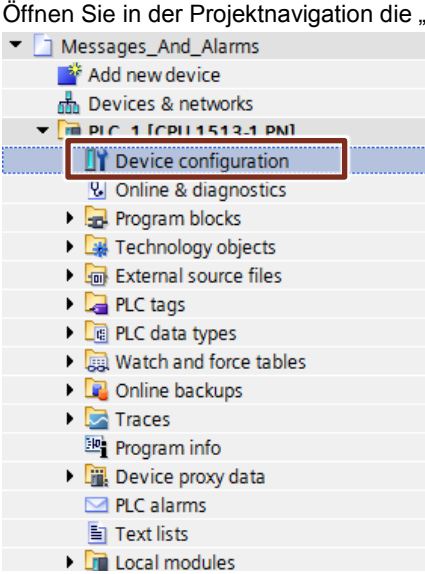
Nr.	Aktion
3.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Eigenschaften &gt; Allgemein“.</p>  <p>Ändern Sie unter „Meldetext“ den Meldetext der Systemmeldung.</p> <p><b>Hinweis</b> Beim Ändern eines Meldetextes darf die Anzahl der Platzhalter nicht verändert werden. Ein Platzhalter ist z. B. %1</p>
4.	Die Änderung des Meldetextes für die Systemmeldung ist damit abgeschlossen.

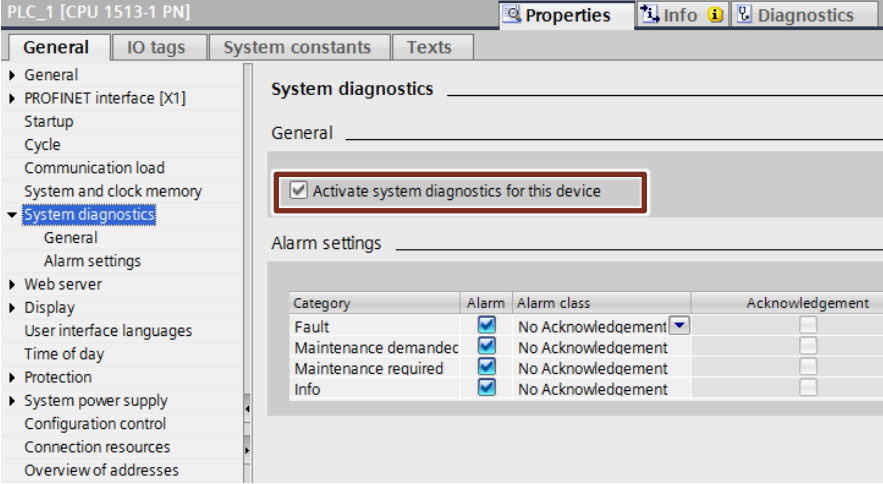
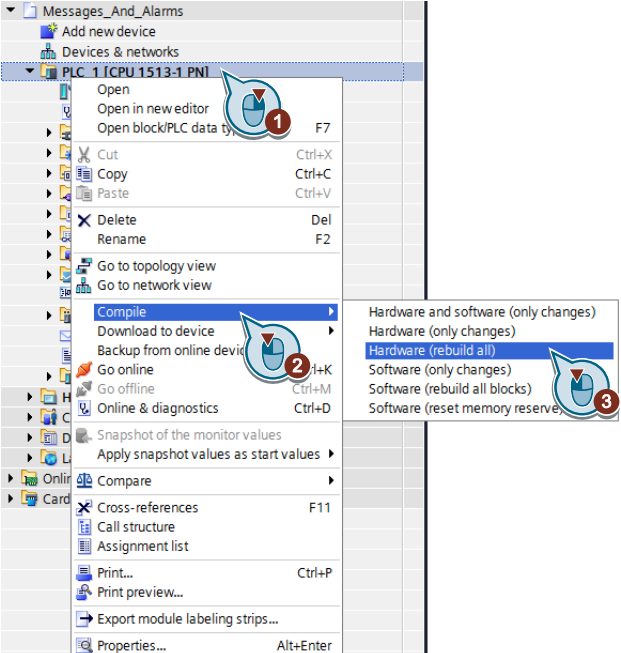
### 2.2.2 Projektierung von CPU-Systemdiagnosemeldungen (S7-1500)

Im folgenden Abschnitt ist die Projektierung der Anzeige von Systemdiagnosemeldungen einer S7-1500 CPU auf einem Bediengerät beschrieben.

#### Einstellungen in STEP 7

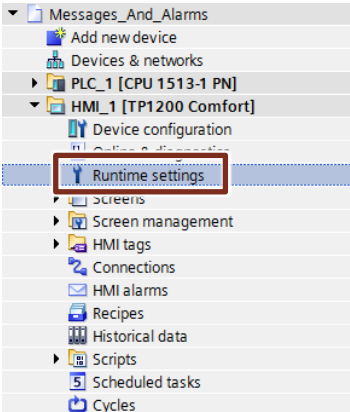
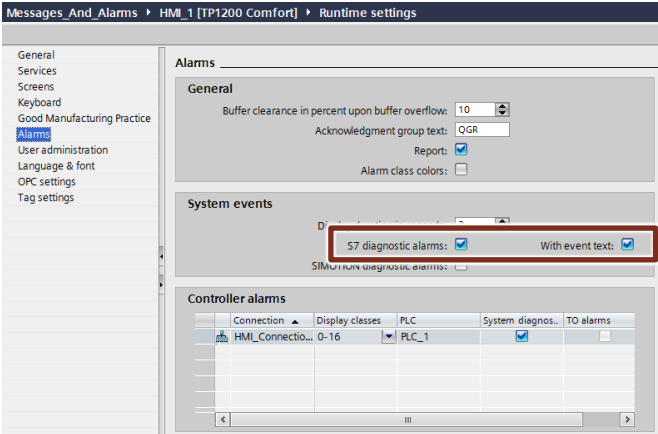
Tabelle 2-5

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation die „Gerätekonfiguration“ ihrer CPU.</p> 
2.	Markieren Sie in der Gerätesicht die CPU auf dem Baugruppenträger.

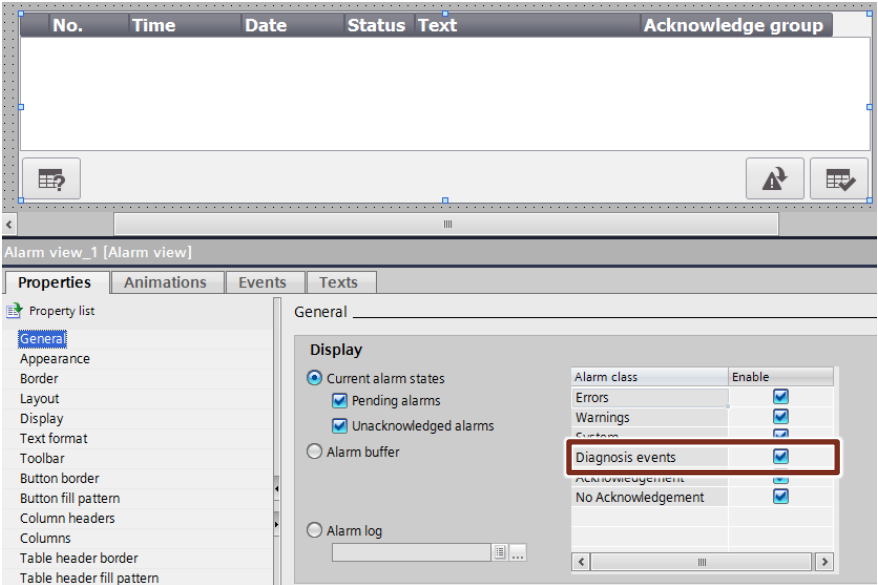
Nr.	Aktion																				
3.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Allgemein &gt; Systemdiagnose“.</p> <p>Das Optionskästchen „Systemdiagnose für dieses Gerät aktivieren“ ist standardmäßig aktiviert und kann nicht deaktiviert werden.</p>  <table border="1" data-bbox="742 683 1356 795"> <thead> <tr> <th>Category</th> <th>Alarm</th> <th>Alarm class</th> <th>Acknowledgement</th> </tr> </thead> <tbody> <tr> <td>Fault</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Maintenance demand</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Maintenance required</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Info</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p><b>Optional:</b></p> <p>Je nach Bedarf können Sie im unteren Bereich des Inspektorfensters unter „Meldungseinstellungen“ die Meldungen und Meldeklassen der jeweiligen Kategorien anpassen.</p>	Category	Alarm	Alarm class	Acknowledgement	Fault	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Maintenance demand	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Maintenance required	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Info	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>
Category	Alarm	Alarm class	Acknowledgement																		
Fault	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Maintenance demand	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Maintenance required	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Info	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
4.	<p>Klicken Sie mit der rechten Maustaste auf die CPU in der Projektnavigation. Wählen Sie im Kontextmenü „Übersetzen &gt; Hardware (komplett übersetzen)“ aus.</p> 																				
5.	Die Einstellungen in STEP 7 sind hiermit abgeschlossen.																				

**Einstellungen in WinCC Basic/Comfort/Advanced**

Tabelle 2-6

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation die „Runtime-Einstellungen“ Ihres Bediengerätes.</p> 
2.	<p>Wählen Sie unter dem Punkt „Meldungen &gt; Systemmeldungen“ das Optionskästchen „S7 Diagnosemeldungen (Nummer)“.</p>  <p>Um neben der Meldenummer den dazugehörigen Meldetext anzuzeigen muss das Optionskästchen „Zeige Meldetext“ zusätzlich aktiviert werden.</p>
3.	<p>Legen Sie im nächsten Schritt eine „Meldeanzeige“ in einem Bild ihres Bediengerätes an.</p>



Nr.	Aktion
4.	<p>Aktivieren Sie unter den Eigenschaften der Meldeanzeige unter „Eigenschaften &gt; Allgemein“ die Meldeklasse „Diagnosis events“ um Systemdiagnosemeldungen in WinCC zur Anzeige zu bringen.</p>  <p><b>Hinweis</b> Je nach Alarmklasse der CPU (wie z. B. in <a href="#">Tabelle 2-5</a> im Schritt 3) kann es zusätzlich notwendig sein, die Meldeklassen „Acknowledgement“ bzw. „No Acknowledgement“ zu aktivieren.</p>
5.	<p>Übertragen Sie das Projekt in das Bediengerät. Die Einstellungen in WinCC sind damit abgeschlossen.</p>

© Siemens AG 2018. All rights reserved.

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Basic/Comfort/Advanced unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

**Hinweis** Informationen und Möglichkeiten der Systemdiagnose finden Sie im Anwendungsbeispiel [„Systemdiagnose mit S7-1500 und TIA Portal“](#).

## 2.3 Projektierung von Steuerungsmeldungen

### 2.3.1 Projektierung von Program\_Alarm (S7-1500)

Die Anweisung „Program\_Alarm: Programmmeldung mit Begleitwerten erzeugen“ überwacht ein Signal und generiert bei einem Signalwechsel am Parameter SIG von 0 auf 1 eine kommende Programmmeldung. Bei einem Signalwechsel von 1 auf 0 wird eine gehende Programmmeldung erzeugt. Die Auswahl ob die Meldung quittierpflichtig ist oder nicht, hängt dabei von der projektierten Meldeklasse ab. Standardmäßig muss der Baustein nicht quittiert werden.

An die Programmmeldung können Sie bis zu zehn Begleitwerte anhängen. Jede Meldung, sowohl kommende als auch gehende, werden dabei mit einem Zeitstempel versehen.

**Parameter des Meldebausteins Program\_Alarm**

Tabelle 2-7

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SIG	Input	BOOL	E, A, M, D, L, T, Z oder Konstante	Das zu überwachende Signal. <ul style="list-style-type: none"> <li>Positive Signalfanke: Eine kommende Programmmeldung wird generiert</li> <li>Negative Signalfanke: Eine gehende Programmmeldung wird generiert</li> </ul>
TIMESTAMP	Input	LDT	M, D, L oder Konstante	Dieser Parameter wird dazu verwendet, einen Alarm mit einem Zeitstempel zu versehen, der z. B. von einem dezentral gestempelten Eingangssignal stammt. Der Zeitwert ist immer in Systemzeit (d. h. UTC) anzugeben, da diese Zeit bei der anlagenweiten Uhrzeitsynchronisation verwendet wird. <ul style="list-style-type: none"> <li>"Nicht verschaltet" bedeutet, dass die Systemzeit der CPU beim Signalwechsel als Zeitstempel für den Alarm verwendet wird (Voreinstellung).</li> <li>Die Übergabe einer Systemzeit bedeutet, dass diese beim Signalwechsel als Zeitstempel für den Alarm verwendet wird.</li> </ul> Hinweis: Falls ein Alarm mit einer Lokalzeit gestempelt werden soll, so muss ein Konvertierungsbaustein Vorgeschaltet werden, der die Lokalzeit auf die Systemzeit umrechnet. Nur dadurch ist garantiert, dass der Zeitstempel in der Alarmanzeige richtig dargestellt wird.
SD_i	Input	VARIANT	E, A, M, D, L	i-ter Begleitwert ( $1 \leq i \leq 10$ ) Als Begleitwerte können Sie Binärzahlen, Ganzzahlen, Gleitpunktzahlen oder Zeichenfolgen verwenden.
Error	Output	BOOL	E, A, M, D, L	Zustandsparameter Error Error = TRUE bedeutet, dass bei der Bearbeitung ein Fehler aufgetreten ist. Die mögliche Fehlerursache wird am Parameter Status angezeigt.
Status	Output	WORD	E, A, M, D, L	Zustandsparameter Status Anzeige der Fehlerinformation (siehe "Parameter Error und Status").

Weitere Informationen zum Baustein „Program\_Alarm“ finden Sie im Systemhandbuch von STEP 7 unter dem Themenpunkt „[Program Alarm: Programmmeldung mit Begleitwerten erzeugen](#)“ oder dem Anwendungsbeispiel „[Diagnose im Anwenderprogramm mit S7-1500](#)“.

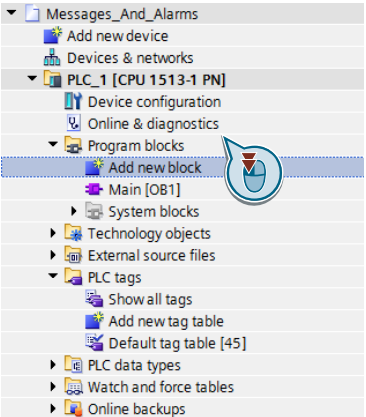
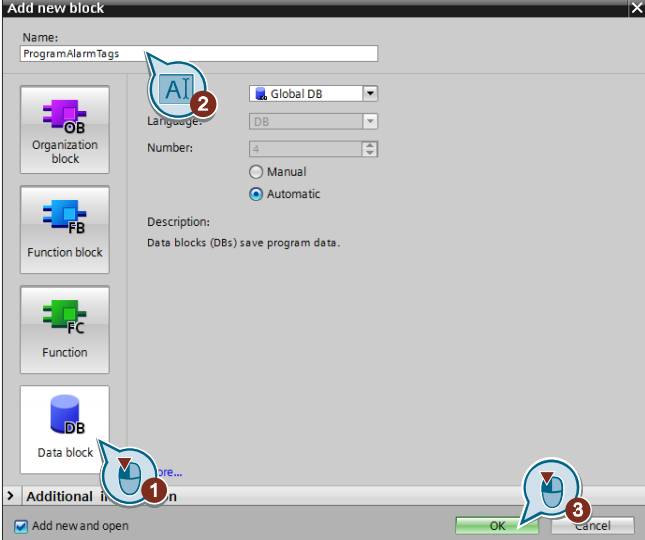
### Einstellung in STEP 7

Die folgende Tabelle zeigt die notwendigen Projektierungsschritte, mit denen Sie eine Programmmeldung in STEP 7 erstellen.

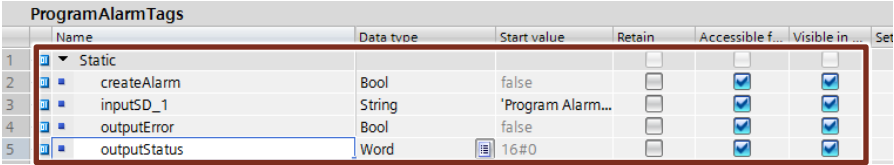
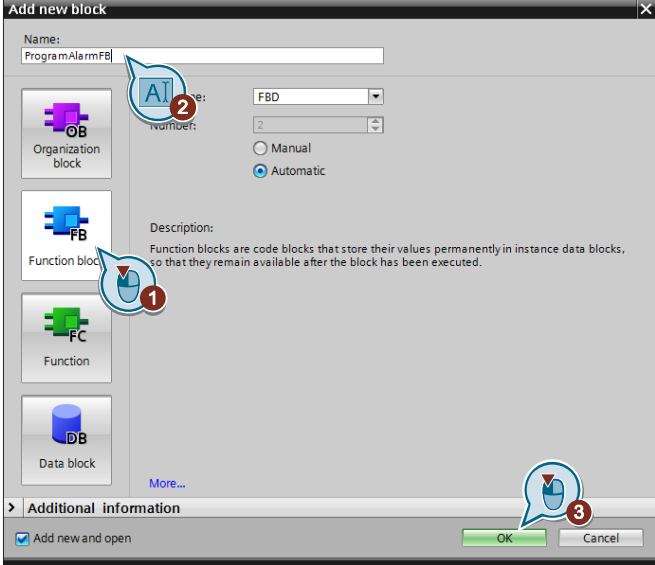
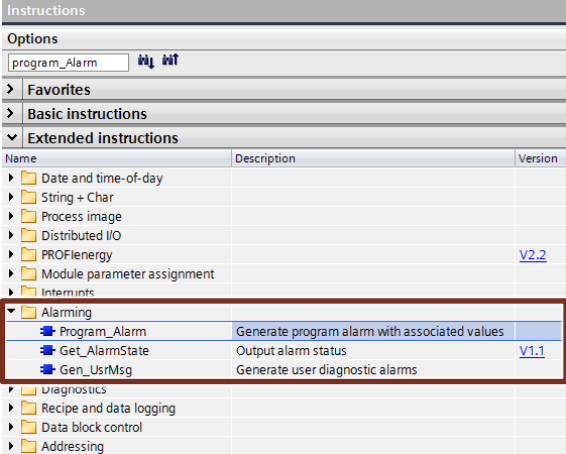
#### Hinweis

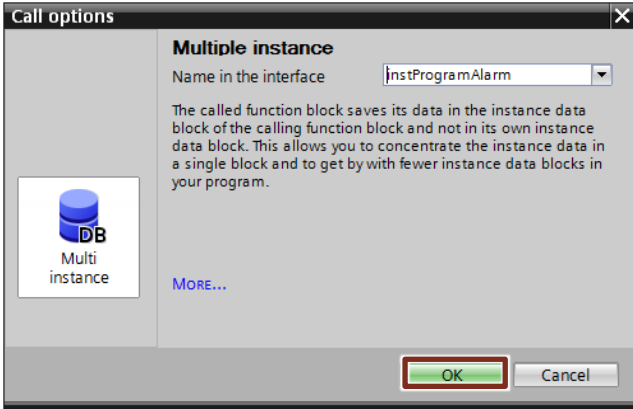
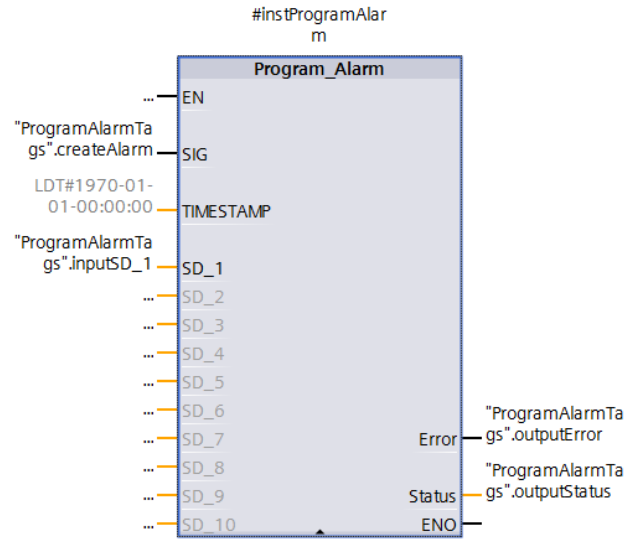
Die STEP 7-Projektierungsschritte für den Meldebaustein „Program\_Alarm“ sind sowohl für die Visualisierungs-Versionen WinCC Basic/Comfort/Advanced als auch für WinCC Professional gültig.

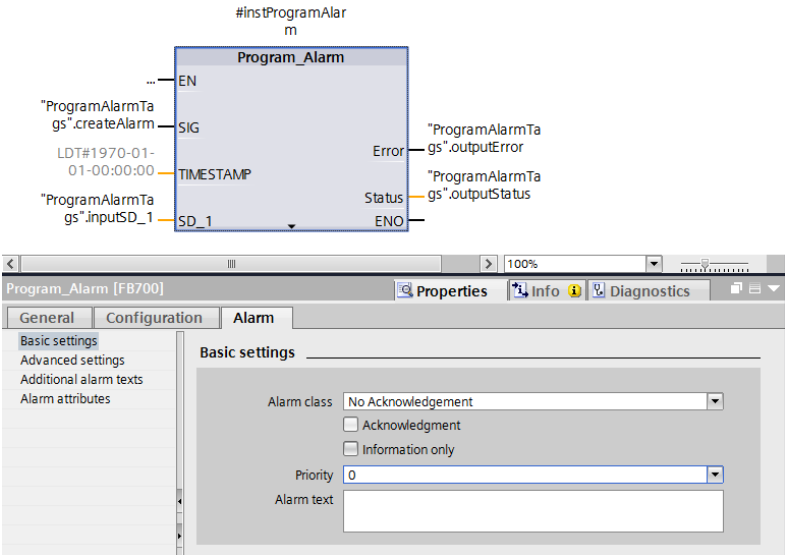
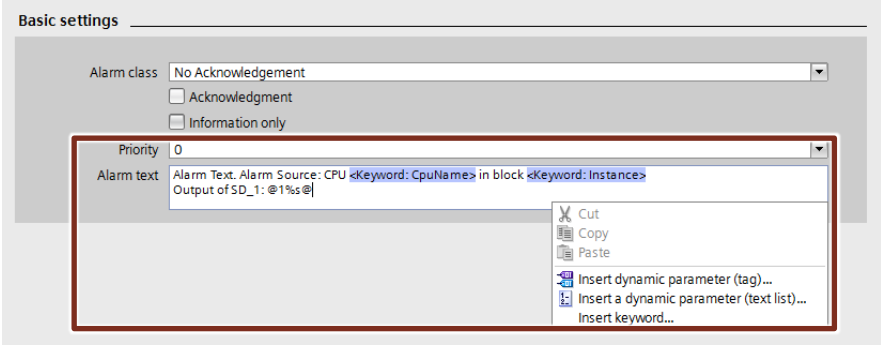
Tabelle 2-8

Nr.	Aktion
1.	<p>Zur Ablage und Überwachung der Werte wird zunächst ein globaler Datenbaustein angelegt.</p> <p>Doppelklicken Sie hierzu unter der angelegten CPU auf „Programmbausteine &gt;Neuen Baustein hinzufügen“.</p> 
2.	<p>Wählen Sie aus dem Dialog „Datenbaustein“ aus und ändern falls notwendig den Namen des Bausteins. Bestätigen Sie anschließend mit „OK“.</p> 

## 2 Projektierung von Meldungen in WinCC Basic/Comfort/Advanced

Nr.	Aktion																																																
3.	<p>Fügen Sie in dem angelegten Datenbaustein vier Variablen mit den entsprechenden Datentypen ein, um die Anweisung „Program_Alarm“ verschalten zu können.</p>  <table border="1" data-bbox="469 327 1366 490"> <caption>ProgramAlarmTags</caption> <thead> <tr> <th></th> <th>Name</th> <th>Data type</th> <th>Start value</th> <th>Retain</th> <th>Accessible f...</th> <th>Visible in ...</th> <th>Set</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>createAlarm</td> <td>Bool</td> <td>false</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>3</td> <td>inputSD_1</td> <td>String</td> <td>'Program Alarm...</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>4</td> <td>outputError</td> <td>Bool</td> <td>false</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>5</td> <td>outputStatus</td> <td>Word</td> <td>16#0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>		Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Set	1	Static							2	createAlarm	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	inputSD_1	String	'Program Alarm...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	outputError	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	outputStatus	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Set																																										
1	Static																																																
2	createAlarm	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																										
3	inputSD_1	String	'Program Alarm...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																										
4	outputError	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																										
5	outputStatus	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																										
4.	<p>Legen Sie, ähnlich wie in den ersten beiden Schritten beschrieben, einen Funktionsbaustein an, um die Anweisung „Program_Alarm“ aufrufen zu können.</p> 																																																
5.	<p>Öffnen Sie unter der Task Card „Anweisungen &gt; Erweiterte Anweisungen“ den Ordner „Meldungen“.</p>  <table border="1" data-bbox="469 1352 1037 1657"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Version</th> </tr> </thead> <tbody> <tr> <td>Program_Alarm</td> <td>Generate program alarm with associated values</td> <td></td> </tr> <tr> <td>Get_AlarmState</td> <td>Output alarm status</td> <td>V1.1</td> </tr> <tr> <td>Gen_UsrMsg</td> <td>Generate user diagnostic alarms</td> <td></td> </tr> </tbody> </table>	Name	Description	Version	Program_Alarm	Generate program alarm with associated values		Get_AlarmState	Output alarm status	V1.1	Gen_UsrMsg	Generate user diagnostic alarms																																					
Name	Description	Version																																															
Program_Alarm	Generate program alarm with associated values																																																
Get_AlarmState	Output alarm status	V1.1																																															
Gen_UsrMsg	Generate user diagnostic alarms																																																

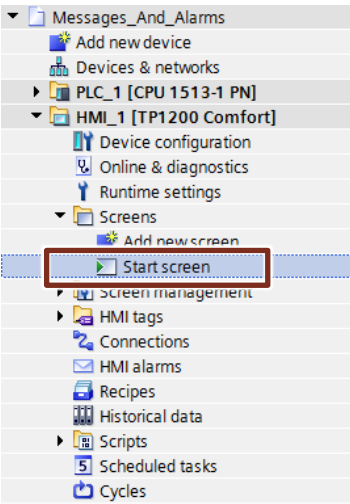
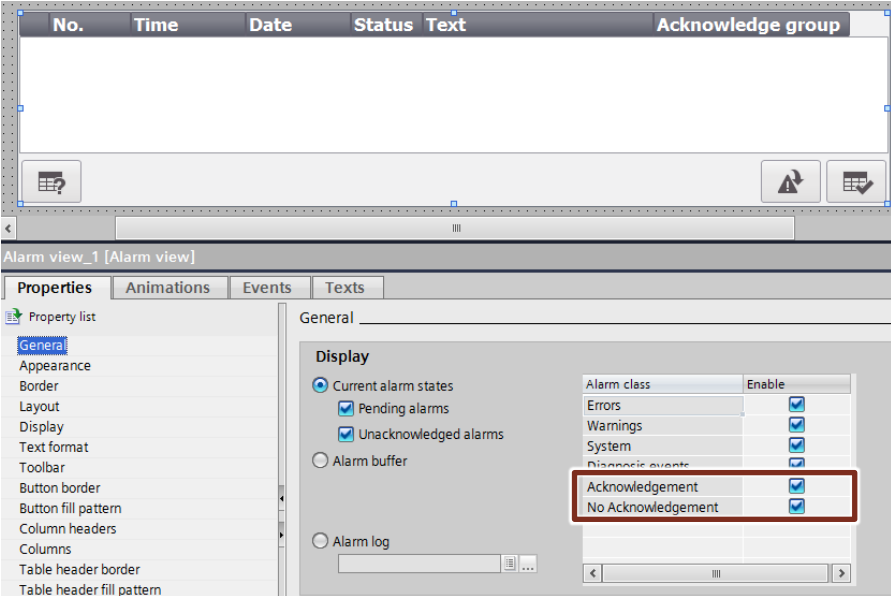
Nr.	Aktion
6.	<p>Ziehen Sie den Meldebaustein „Program_Alarm“ per Drag&amp;Drop in ein leeres Netzwerk des Funktionsbausteins. Bestätigen Sie die „Aufrufoptionen“ mit „OK“.</p> 
7.	<p>Verschalten Sie den eingefügten Baustein „Program_Alarm“ mit den Variablen aus dem Datenbaustein.</p> 

Nr.	Aktion
8.	<p>Um die Meldungen zu bearbeiten, klicken Sie unter dem Register „Eigenschaften“ auf Meldungen. Hier können Sie:</p> <ul style="list-style-type: none"> <li>• Die Meldeklasse, Priorität und den Meldetext bearbeiten (Grundeinstellungen),</li> <li>• Anzeigeklasse, Gruppen-ID und Protokollierung wählen (Erweiterte Einstellungen) und</li> <li>• Zusätzliche Meldetexte definieren.</li> </ul>  <p>The top image shows a Ladder Logic (LAD) diagram of the 'Program_Alarm' block. It is connected to a variable '#InstProgramAlarm' and has several inputs and outputs: 'EN' (Enable), 'SIG' (Signal) from 'ProgramAlarmTags.createAlarm', 'TIMESTAMP' from 'LDT#1970-01-01-00:00:00', 'SD_1' (Status Data) from 'ProgramAlarmTags.inputSD_1', 'Error' (Output Error), 'Status' (Output Status) to 'ProgramAlarmTags.outputStatus', and 'ENO' (Enable Output).</p> <p>The bottom image shows the 'Properties' window for the 'Program_Alarm' block. The 'Alarm' tab is active, showing 'Basic settings'. The 'Alarm class' is set to 'No Acknowledgement'. There are checkboxes for 'Acknowledgment' and 'Information only'. The 'Priority' is set to '0'. The 'Alarm text' field is empty.</p>
9.	<p>Wenn Sie im Eingabefenster der Meldetexte mit der rechten Maustaste klicken, wählen Sie aus dem Kontextmenü verschiedene Optionen aus, um die Meldetexte genauer zu spezifizieren.</p>  <p>The image shows a close-up of the 'Alarm text' input field. The text inside is 'Alarm Text. Alarm Source: CPU &lt;Keyword: CpuNames&gt; in block &lt;Keyword: Instance&gt; Output of SD_1: @1%s@'. A right-click context menu is open over the text, showing options: 'Cut', 'Copy', 'Paste', 'Insert dynamic parameter (tag)...', 'Insert a dynamic parameter (text list)...', and 'Insert keyword...'. A red box highlights the input field and the menu.</p> <p><b>Hinweis</b>          Durch die Zeichenfolge „@1%s@“ wird der Wert des Parameter SD_1 ausgelesen und als Zeichenkette ausgegeben.</p>



**Einstellungen in WinCC Basic/Comfort/Advanced**

Tabelle 2-9

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation das Startbild ihres Bediengerätes.</p> 
2.	<p>Legen Sie eine Meldeanzeige an.</p>
3.	<p>Aktivieren Sie unter den Eigenschaften der Meldeanzeige unter „Eigenschaften &gt; Allgemein“ die Meldeklasse „Acknowledgement“ und „No Acknowledgement“ um Programmierungen in WinCC zur Anzeige zu bringen.</p> 
4.	<p>Übertragen Sie das Projekt auf ihr Bediengerät. Die Einstellungen in WinCC sind damit abgeschlossen.</p>

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Basic/Comfort/Advanced unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

### 2.3.2 Projektierung von Get\_AlarmState (S7-1500)

Mit der Anweisung „Meldungszustand ausgeben“ (Get\_AlarmState) geben Sie den Meldungszustand einer Programmmeldung aus.

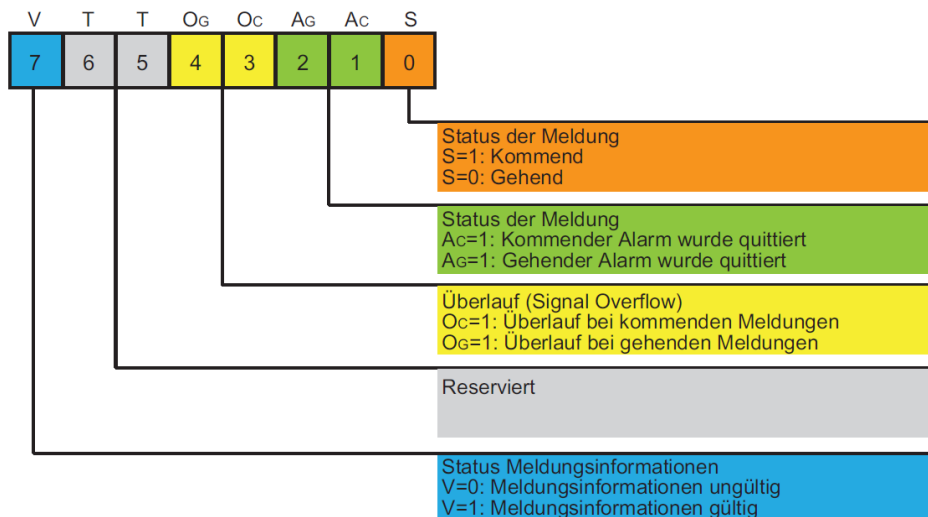
**Hinweis** Die Ausgabe des Meldungszustandes bezieht sich jeweils auf eine Programmmeldung, die über die Anweisung „Program\_Alarm“ erstellt wurde.

Welche Programmmeldung ausgegeben werden soll, wird über den Eingangsparameter „Alarm“ festgelegt. Hier geben Sie den Instanz-DB der Anweisung „Program\_Alarm“ an, dessen Meldungszustand Sie ausgeben möchten.

Über den Ausgangsparameter „AlarmState“ wird der Meldungszustand in einem Byte ausgegeben.

Die einzelnen Bits des Ausgangsparameters „AlarmState“ bedeuten hierbei:

Abbildung 2-1



Weitere Informationen zur Anweisung „Get\_AlarmState“ finden Sie im Systemhandbuch von STEP 7 unter dem Themenpunkt [„Get\\_AlarmState: Meldungszustand ausgeben“](#).

## Parameter des Bausteins Get\_AlarmState

Tabelle 2-10

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
Alarm	Input	ALARM_BASE	D	<p>Instanz der Anweisung "Programmmeldung mit Begleitwerten erzeugen".</p> <ul style="list-style-type: none"> <li>Alarm.MessageType = Alarm_AP, dann hat das Bit Ac entweder den Signalzustand 0 oder 1 und das Bit Ag hat den Signalzustand 1 <ul style="list-style-type: none"> <li>nicht aktiv: 0x86 (1000 0110)</li> <li>aktiv/nicht quittiert: 0x85 (1000 0101)</li> <li>aktiv/quittiert: 0x87 (1000 0111)</li> <li>gegangen/nicht quittiert: 0x84 (10000100)</li> </ul> </li> <li>Alarm.MessageType = Notify_AP, dann hat das Bit Ac entweder den Signalzustand 0 oder 1 und das Bit Ag hat den Signalzustand 1 <ul style="list-style-type: none"> <li>nicht aktiv: 0x86 (1000 0110)</li> <li>aktiv: 0x85 (10000101)</li> </ul> </li> <li>Alarm.MessageType = Inforeport_AP, dann haben die Bits Ac und Ag beide den Signalzustand 1 <ul style="list-style-type: none"> <li>Statisch: 0x86 (10000110)</li> </ul> </li> </ul> <p>Wenn der Alarm nicht aktiv ist oder es sich um einen Inforeport handelt, dann hat das Bit S den Signalzustand 0.</p>
AlarmState	Output	BYTE	E, A, M, D, L	Status der Meldung als Bitfeld
Error	Output	BOOL	E, A, M, D, L	<p>Zustandsparameter</p> <ul style="list-style-type: none"> <li>0: kein Fehler</li> <li>1: Bei der Ausführung der Anweisung ist ein Fehler aufgetreten.</li> </ul> <p>Detailinformationen werden über den Parameter STATUS ausgegeben.</p>

## 2 Projektierung von Meldungen in WinCC Basic/Comfort/Advanced

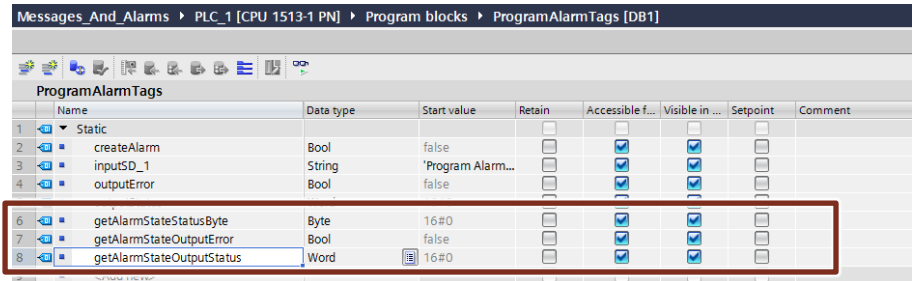
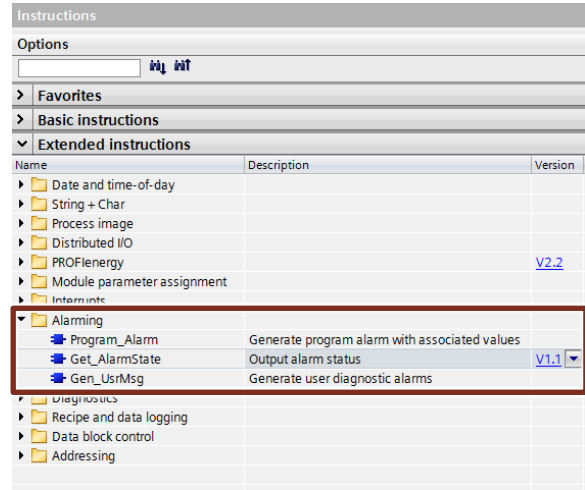
Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
STATUS	Output	WORD	E, A, M, D, L	Zustandsparameter Der Parameter ist nur einen Aufruf lang gesetzt. Zur Anzeige des Status sollten Sie deshalb den STATUS in einen freien Datenbereich kopieren.

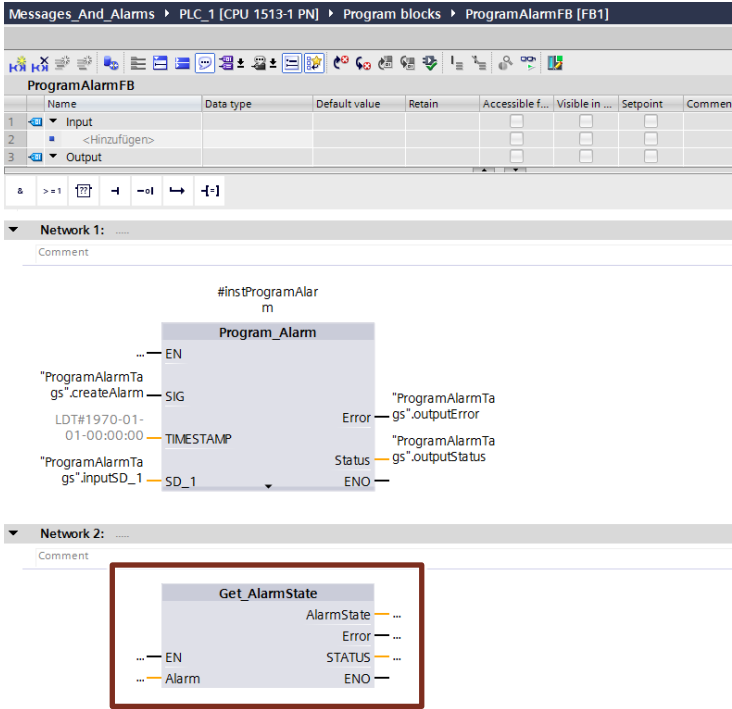
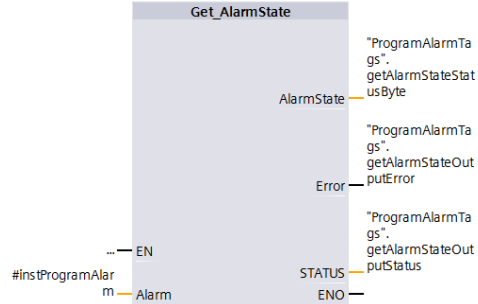
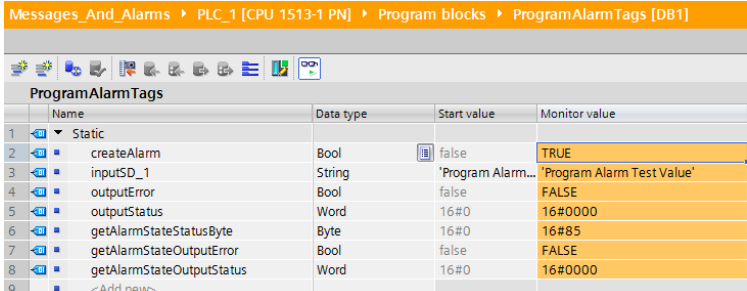
### Projektierung in STEP 7

Die nachfolgende Tabelle zeigt Ihnen wie Sie den Meldungsstatus einer Programmmeldung ausgeben können. Um diese projektieren zu können, ist es notwendig, dass bereits eine Programmmeldung (wie in Kapitel [2.3.1](#)) erzeugt wurde.

**Hinweis** Die Projektierungsschritte für die Anweisung „Get\_AlarmState“ sind sowohl für die Visualisierungs-Versionen WinCC Basic/Comfort/Advanced als auch für WinCC Professional gültig.

Tabelle 2-11

Nr.	Aktion																																																																
1.	<p>Öffnen Sie den Datenbaustein „ProgramAlarmTags“ (siehe <a href="#">Tabelle 2-8</a>) und fügen Sie drei Variablen mit den zugehörigen Datentypen ein.</p>  <table border="1"> <thead> <tr> <th>Name</th> <th>Data type</th> <th>Start value</th> <th>Retain</th> <th>Accessible f...</th> <th>Visible in ...</th> <th>Setpoint</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>Static</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>createAlarm</td> <td>Bool</td> <td>false</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>inputSD_1</td> <td>String</td> <td>'Program Alarm...</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>outputError</td> <td>Bool</td> <td>false</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>getAlarmStateStatusByte</td> <td>Byte</td> <td>16#0</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>getAlarmStateOutputError</td> <td>Bool</td> <td>false</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>getAlarmStateOutputStatus</td> <td>Word</td> <td>16#0</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> </tbody> </table>	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment	Static								createAlarm	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		inputSD_1	String	'Program Alarm...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		outputError	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		getAlarmStateStatusByte	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		getAlarmStateOutputError	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		getAlarmStateOutputStatus	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment																																																										
Static																																																																	
createAlarm	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
inputSD_1	String	'Program Alarm...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
outputError	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
getAlarmStateStatusByte	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
getAlarmStateOutputError	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
getAlarmStateOutputStatus	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																											
2.	<p>Öffnen sie unter der Task Card „Anweisungen &gt; Erweiterte Anweisungen“ den Ordner „Meldungen“.</p>  <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Version</th> </tr> </thead> <tbody> <tr> <td>Program_Alarm</td> <td>Generate program alarm with associated values</td> <td></td> </tr> <tr> <td>Get_AlarmState</td> <td>Output alarm status</td> <td>V1.1</td> </tr> <tr> <td>Gen_UsrMsg</td> <td>Generate user diagnostic alarms</td> <td></td> </tr> </tbody> </table>	Name	Description	Version	Program_Alarm	Generate program alarm with associated values		Get_AlarmState	Output alarm status	V1.1	Gen_UsrMsg	Generate user diagnostic alarms																																																					
Name	Description	Version																																																															
Program_Alarm	Generate program alarm with associated values																																																																
Get_AlarmState	Output alarm status	V1.1																																																															
Gen_UsrMsg	Generate user diagnostic alarms																																																																

Nr.	Aktion
3.	<p>Ziehen Sie den Baustein „Get_AlarmState“ per Drag&amp;Drop in ein leeres Netzwerk des Funktionsbausteins, in dem der Baustein für die Programmmeldung(Programm_Alarm) aufgerufen wird.</p>  <p>The screenshot shows the 'ProgramAlarmFB' function block configuration. In 'Network 1', the 'Program_Alarm' block is called with inputs: EN, SIG (from 'ProgramAlarmTags.createAlarm'), LDT#1970-01-01-00:00:00 (TIMESTAMP), and SD_1 (from 'ProgramAlarmTags.inputSD_1'). It has outputs: outputError, outputStatus, and ENO. In 'Network 2', the 'Get_AlarmState' block is highlighted with a red box, showing its inputs: EN and Alarm, and its outputs: AlarmState, Error, STATUS, and ENO.</p>
4.	<p>Verknüpfen Sie den Eingangsparameter „Alarm“ mit der Instanz der Anweisung „Program_Alarm“. Anschließend verschalten Sie die Ausgangsparameter des Bausteins mit den drei angelegten Variablen des Datenbausteins.</p>  <p>The screenshot shows the 'Get_AlarmState' block with its 'Alarm' input connected to the 'Alarm' output of the '#instProgramAlarm' block. The 'AlarmState' output is connected to 'ProgramAlarmTags.getAlarmStateStatusByte', 'Error' to 'ProgramAlarmTags.putError', and 'STATUS' to 'ProgramAlarmTags.putStatus'.</p>
5.	Laden Sie die Projektierung in Ihre CPU.
6.	<p>Sobald nun im Projekt die Anweisung „Program_Alarm“ einen Signalzustand „True“ am Eingangsparameter SIG erhält, wird die Programmmeldung ausgegeben. Am Ausgang „AlarmState“ der Anweisung „Get_AlarmState“ wird daraufhin die Meldung entsprechende der Meldeklasse angezeigt.</p>  <p>The screenshot shows the 'ProgramAlarmTags' data block with the following values: createAlarm (Bool, false, TRUE), inputSD_1 (String, 'Program Alarm...', 'Program Alarm Test Value'), outputError (Bool, false, FALSE), outputStatus (Word, 16#0, 16#0000), getAlarmStateStatusByte (Byte, 16#0, 16#85), getAlarmStateOutputError (Bool, false, FALSE), and getAlarmStateOutputStatus (Word, 16#0, 16#0000).</p>

### 2.3.3 Projektierung von Gen\_UsrMsg (S7-1200, S7-1500)

Mit der Anweisung „Gen\_UsrMsg“ erzeugen Sie eine Meldung, die in den Diagnosepuffer eingetragen wird. Über den Parameter Mode können Sie wählen, ob eine kommende, oder eine gehende Meldung erzeugt werden soll:

- Mode = 1: Erzeugung einer kommenden Meldung
- Mode = 2: Erzeugung einer gehenden Meldung

Den Inhalt der Meldung definieren Sie über Textlisten und die entsprechenden Textlisten-Einträge. Über die Parameter „TextListID“ und „TextID“ können Sie den entsprechenden Eintrag auswählen, den Sie in den Diagnosepuffer schreiben wollen. In den Textlisten können Sie dabei auch Begleitwerte definieren, welche dann im Diagnosepuffer mit angezeigt werden sollen.

Weitere Informationen zur Anweisung „Gen\_UsrMsg“ finden Sie im Systemhandbuch von STEP 7 unter dem Themenpunkt [„Gen\\_UsrMsg: Anwenderdiagnosemeldung erzeugen“](#).

**Hinweis** Die Anweisung „Anwenderdiagnosemeldung erzeugen“(GEN\_UsrMsg) ist erst ab STEP 7 V13 SP1 verfügbar.

#### Parameter des Bausteins Get\_AlarmState

Tabelle 2-12

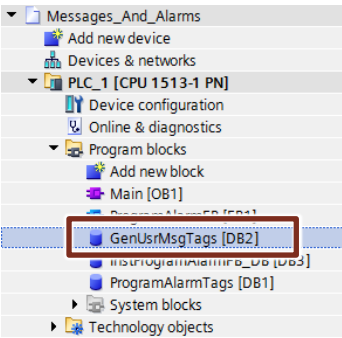
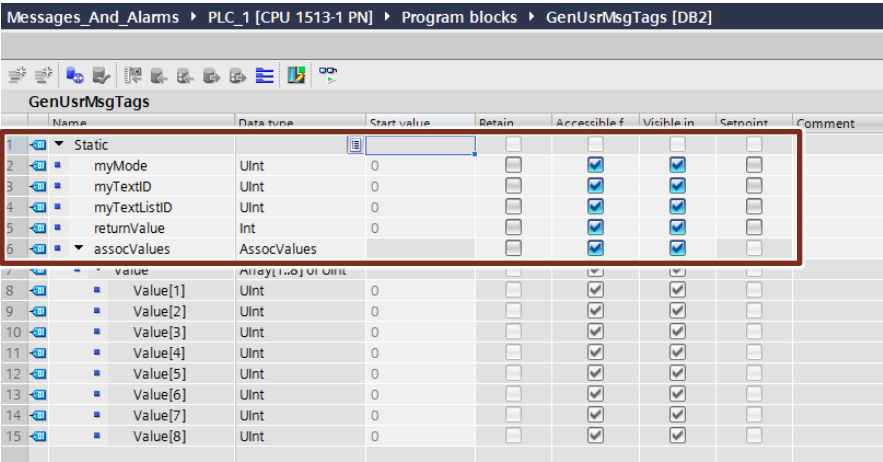
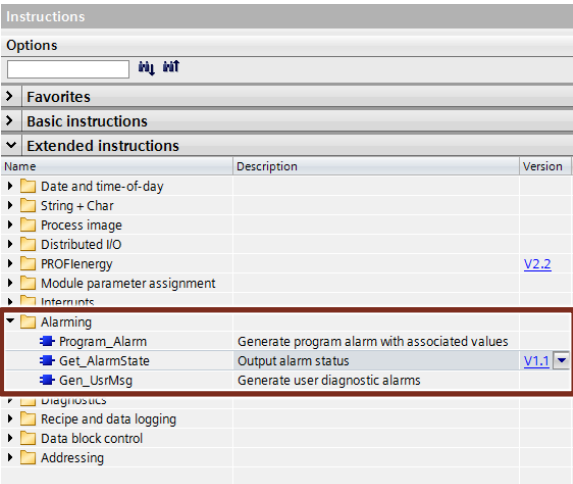
Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
Mode	Input	UInt	E, A, M, D, L, oder Konstante	Parameter zur Auswahl des Status der Meldung <ul style="list-style-type: none"> <li>• 1: kommende Meldung</li> <li>• 2: gehende Meldung</li> </ul>
TextID	Input	UInt	E, A, M, D, L, oder Konstante	ID des Textlisten-Eintrags, welcher für den Meldetext verwendet werden soll.
TextListID	Input	UInt	E, A, M, D, L, oder Konstante	ID der Textliste, die Textlisten-Einträge enthält.
Ret_Val	Return	Int	E, A, M, D, L	Fehlercode der Anweisung.
AssocValues	InOut	Variant	D, L	Zeiger auf den Systemdatentyp AssocValues, über den Sie die Begleitwerte definieren.

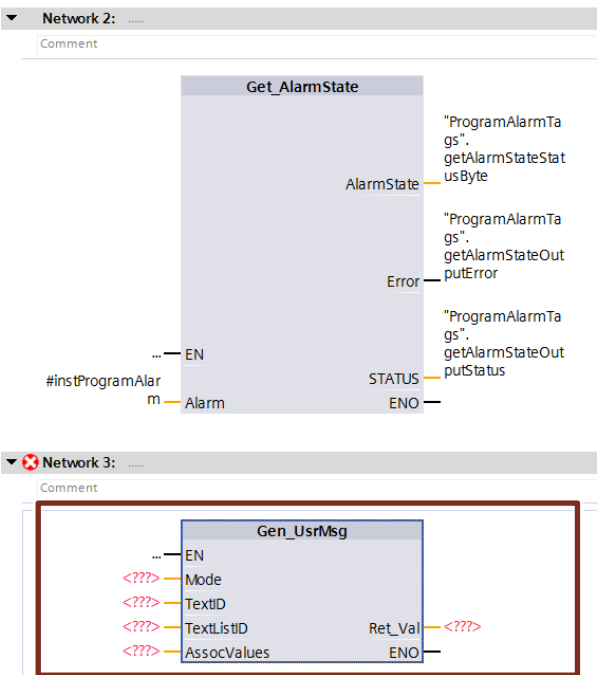
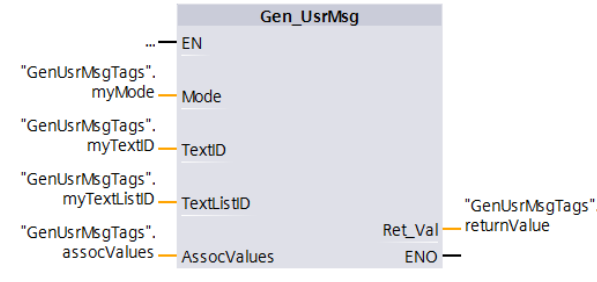
#### Projektierung in STEP 7

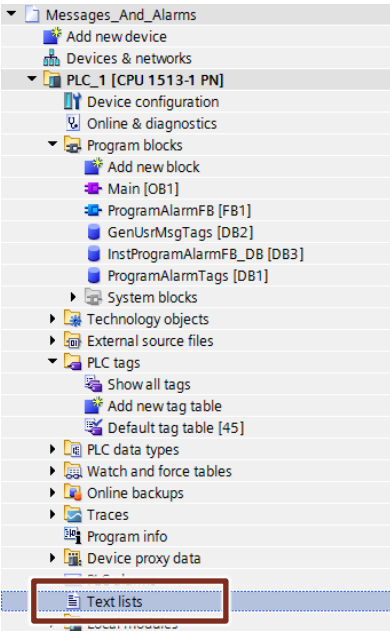
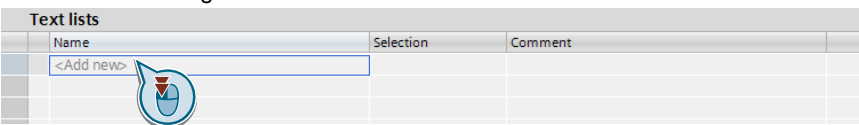
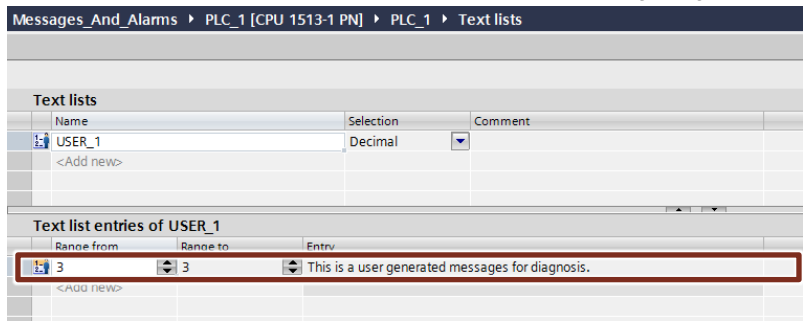
**Hinweis** Die Projektierungsschritte für die Anweisung „Get\_AlarmState“ sind sowohl für die Visualisierungs-Versionen WinCC Basic/Comfort/Advanced als auch für WinCC Professional gültig.

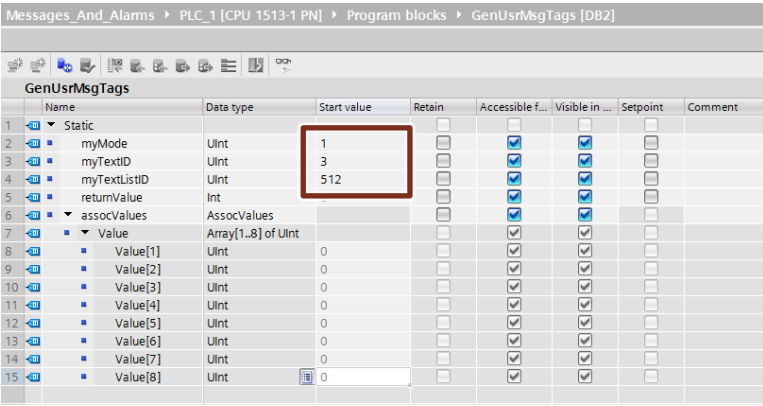


Tabelle 2-13

Nr.	Aktion																																																																																																																								
1.	<p>Zur Ablage und Überwachung der Werte wird zunächst ein weiterer globaler Datenbaustein angelegt.</p> 																																																																																																																								
2.	<p>Legen Sie in dem Datenbaustein „GenUsrMsgTags“ fünf Variablen mit den dazugehörigen Datentypen an.</p>  <table border="1" data-bbox="470 772 1356 1232"> <thead> <tr> <th>Name</th> <th>Data type</th> <th>Start value</th> <th>Retain</th> <th>Accessible f</th> <th>Visible in</th> <th>Setpoint</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>Static</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>myMode</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>myTextID</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>myTextListID</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>returnValue</td> <td>Int</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>assocValues</td> <td>AssocValues</td> <td></td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[1]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[2]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[3]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[4]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[5]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[6]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[7]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Value[8]</td> <td>UInt</td> <td>0</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> </tbody> </table> <p><b>Hinweis</b> Der Datentyp „AssocValues“ ist in STEP 7 standardmäßig als Systemdatentyp hinterlegt. Um diesen Datentyp einer Variablen zuzuweisen, tragen Sie in die Spalte Datentyp „AssocValues“ ein und bestätigen mit der Eingabetaste.</p>	Name	Data type	Start value	Retain	Accessible f	Visible in	Setpoint	Comment	Static								myMode	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			myTextID	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			myTextListID	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			returnValue	Int	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			assocValues	AssocValues			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[1]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[2]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[3]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[4]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[5]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[6]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[7]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Value[8]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Name	Data type	Start value	Retain	Accessible f	Visible in	Setpoint	Comment																																																																																																																		
Static																																																																																																																									
myMode	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
myTextID	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
myTextListID	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
returnValue	Int	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
assocValues	AssocValues			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[1]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[2]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[3]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[4]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[5]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[6]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[7]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
Value[8]	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																																																																																																				
3.	<p>Öffnen Sie unter der Task Card „Anweisungen &gt; Erweiterte Anweisungen“ den Ordner „Meldungen“.</p> 																																																																																																																								

Nr.	Aktion
4.	<p>Ziehen Sie Baustein „Gen_UsrMsg“ per Drag&amp;Drop in ein leeres Netzwerk des Funktionsbausteins, in dem der Baustein für die Programmmeldung aufgerufen wird.</p>  <p>The screenshot shows two networks in a ladder logic editor. Network 2 contains a function block 'Get_AlarmState'. It has two inputs: 'EN' and 'Alarm' (connected to '#InstProgramAlarm'). It has three outputs: 'AlarmState' (connected to '"ProgramAlarmTags".getAlarmStateStatusByte'), 'Error' (connected to '"ProgramAlarmTags".getAlarmStateOutputError'), and 'STATUS' (connected to '"ProgramAlarmTags".getAlarmStateOutputStatus'). Network 3 contains a data block 'Gen_UsrMsg' highlighted with a red box. It has four inputs: 'EN', 'Mode' (connected to '&lt;???'&gt;), 'TextID' (connected to '&lt;???'&gt;), and 'TextListID' (connected to '&lt;???'&gt;). It has two outputs: 'Ret_Val' (connected to '&lt;???'&gt;') and 'ENO'.</p>
5.	<p>Verknüpfen Sie den Eingangsparameter „Alarm“ mit der Instanz der Anweisung „Program_Alarm“. Anschließend verschalten Sie Ausgangsparametern des Bausteins mit den drei angelegten Variablen des Datenbausteins.</p>  <p>The screenshot shows the 'Gen_UsrMsg' data block with its inputs and outputs fully connected to variables. The inputs are: 'EN', 'Mode' (connected to '"GenUsrMsgTags".myMode'), 'TextID' (connected to '"GenUsrMsgTags".myTextID'), and 'TextListID' (connected to '"GenUsrMsgTags".myTextListID'). The outputs are: 'Ret_Val' (connected to '"GenUsrMsgTags".returnValue') and 'ENO'.</p>

Nr.	Aktion
6.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihrer angelegten CPU die „Textlisten“.</p> 
7.	<p>Klicken Sie im Arbeitsbereich der Textliste doppelt auf „Hinzufügen“, um eine neue Textliste hinzuzufügen.</p> 
8.	<p>Fügen Sie der Textliste einen neuen Textlisteneintrag hinzu. Geben Sie hierbei zunächst den Bereich an und anschließend den Text der angezeigt werden soll.</p>  <p><b>Hinweis</b> Die ID der Textliste können Sie sich anzeigen lassen, indem Sie mit der rechten Maustaste auf den Tabellenkopf klicken und „Alle Spalten anzeigen“ auswählen.</p>

Nr.	Aktion																																																																																																																																
9.	<p>Ergänzen Sie in dem Datenbaustein die Werte der Text ID und Textlisten ID der erstellten Textliste.</p>  <p>Messages_And_Alarms &gt; PLC_1 [CPU 1513-1 PN] &gt; Program blocks &gt; GenUsrMsgTags [DB2]</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Data type</th> <th>Start value</th> <th>Retain</th> <th>Accessible f...</th> <th>Visible in ...</th> <th>Setpoint</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>myMode</td> <td>UInt</td> <td>1</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>3</td> <td>myTextID</td> <td>UInt</td> <td>3</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>4</td> <td>myTextListID</td> <td>UInt</td> <td>512</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>5</td> <td>returnValue</td> <td>Int</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>6</td> <td>assocValues</td> <td>AssocValues</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>7</td> <td>Value</td> <td>Array[1..8] of UInt</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>8</td> <td>Value[1]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>9</td> <td>Value[2]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>10</td> <td>Value[3]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>11</td> <td>Value[4]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>12</td> <td>Value[5]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>13</td> <td>Value[6]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>14</td> <td>Value[7]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td>15</td> <td>Value[8]</td> <td>UInt</td> <td>0</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> </tbody> </table>	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment	1	Static							2	myMode	UInt	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3	myTextID	UInt	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		4	myTextListID	UInt	512	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		5	returnValue	Int		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		6	assocValues	AssocValues		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		7	Value	Array[1..8] of UInt		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		8	Value[1]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		9	Value[2]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		10	Value[3]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		11	Value[4]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		12	Value[5]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		13	Value[6]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		14	Value[7]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		15	Value[8]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment																																																																																																																										
1	Static																																																																																																																																
2	myMode	UInt	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
3	myTextID	UInt	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
4	myTextListID	UInt	512	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
5	returnValue	Int		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
6	assocValues	AssocValues		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
7	Value	Array[1..8] of UInt		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
8	Value[1]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
9	Value[2]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
10	Value[3]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
11	Value[4]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
12	Value[5]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
13	Value[6]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
14	Value[7]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
15	Value[8]	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																																																																																											
10.	Laden Sie die Projektierung in Ihre CPU.																																																																																																																																

Die erzeugte Anwenderdiagnosemeldung können Sie sich im Diagnosepuffer der angelegten S7-1500 CPU ausgeben lassen.

**Hinweis**

Den Diagnosepuffer der CPU können Sie sich auch über den Webserver oder dem HMI-Control „System-Diagnoseanzeige“ auf einem Panel ausgeben lassen.

Weiter Informationen finden Sie in dem Anwendungsbeispiel [„System-Diagnoseanzeige unter Verwendung von WinCC \(TIA Portal\) und SIMATIC Comfort Panels“](#) oder dem FAQ: [„Wie kann der Diagnosepuffer einer SIMATIC CPU mit integriertem Webserver an einem SIMATIC Panel angezeigt werden?“](#)

## 2.4 Einsatz von Meldeklassen

Während des Betriebs einer Anlage treten Meldungen unterschiedlicher Prioritäten auf. Um diese Meldungen zu ordnen, teilt man diese in Abhängigkeit der Wichtigkeit in verschiedene Meldeklassen ein. Der Bediener der Anlage kann dadurch erkennen, welche Meldungen eine höhere Priorität und Wichtigkeit haben und vorrangig behandelt werden müssen.

### Funktion der Meldeklassen

Beim Anlegen von neuen Meldungen muss jede einer Meldeklasse zugeordnet werden.

Die Meldeklasse definiert dabei:

- Die Darstellung
- Das Quittierkonzept (Einfachquittierung, Zweifachquittierung, zentraler Melder)
- Die Archivierung der Meldung

### Hinweis

Der Funktionsumfang der Meldeklasse ist abhängig vom projektierten Bediengerät.

### Arten von Meldeklassen

Folgende Arten von Meldeklassen stehen in WinCC (TIA Portal) zur Verfügung.

- **Vordefinierte Meldeklassen**

Für jedes Bediengerät finden Sie im Projektbaum unter „HMI-Meldungen“ in der Registerkarte „Meldeklassen“ vordefinierte Meldeklassen. Diese Meldeklassen können nicht gelöscht und nur bedingt bearbeitet werden.

- **Benutzerdefinierte Meldeklassen**

Für jedes Bediengerät können im Projektbaum unter „HMI-Meldungen“ in der Registerkarte „Meldeklassen“ weitere Meldeklassen angelegt werden. In diesen Meldeklassen kann die Darstellung und das Quittierkonzept der dazugehörigen Meldungen individuell projektiert werden.

- **Projektweite Meldeklassen (nicht bei Basic Panels)**

Projektweite Meldeklassen können bediengeräteübergreifend im Projekt verwendet werden. Im Projektbaum unter „Gemeinsame Daten > Meldeklassen“ werden diese Meldeklassen angezeigt und neue projektweite Meldeklassen projektiert.

## 2.5 Einsatz von Meldegruppen

Während des Betriebs einer Anlage treten Meldungen auf die verschiedene Prozesse und Bereiche betreffen. Um diese Meldungen zu gliedern (z. B. nach einem Anlagenteil) und eine Übersichtlichkeit zu schaffen, kann man Meldungen zu Meldegruppen hinzufügen.

### Funktionsumfang von Meldegruppen in WinCC Basic/Comfort/Advanced

In WinCC Basic/Comfort/Advanced kann man mit Hilfe der Meldegruppen Anlagenteile separat betrachten und die dazugehörigen Meldungen bei Bedarf gemeinsam quittieren. Es können nur quittierpflichtige Meldungen einer Meldegruppe zugeordnet werden, unabhängig von der verwendeten Meldeklasse.

#### Beispiel

Sind mehrere quittierpflichtige Meldungen einer Meldegruppe zugeordnet, werden beim Quittieren einer dieser Meldungen alle Meldungen dieser Meldegruppe quittiert.

### Verwendungszweck

Der Einsatz von Meldegruppen ist bei folgenden Meldungen zu empfehlen:

- Störmeldungen mit gleicher Ursache
- Gleichartige Meldungen
- Meldungen aus einem Anlagenteil (z. B. Presse\_1)
- Meldungen die Teil eines Prozesses sind (z. B. der Temperaturüberwachung)



## 2.6 Das Quittierkonzept

Das Quittierkonzept legen Sie für eine Meldeklasse fest. Alle Meldungen die Bestandteil dieser Meldeklasse sind, werden dann nach diesem Konzept quittiert.

In WinCC Basic/Compact/Advanced gibt es folgende Quittierkonzepte:

- Meldung ohne Quittierung
- Meldung mit Einfachquittierung

Im folgenden Verlauf des Kapitels werden die Quittierkonzepte näher erläutert.

### 2.6.1 Meldung ohne Quittierung

Bei einer Meldung ohne Quittierung muss der Bediener nicht zwingend auf die Meldung reagieren. Die Meldung kommt und geht, ohne dass eine Quittierung erforderlich ist.

### 2.6.2 Meldung mit Einfachquittierung

Im Laufe des Betriebs einer Anlage können Meldungen auftreten, die für den Bediener offensichtlich erkennbar sein müssen. Durch die Projektierung einer Quittierung für eine Meldung steht diese Meldung so lange an, bis der Bediener diese quittiert hat.

**Hinweis** Meldungen über kritische und gefährliche Zustände im Prozess müssen quittierpflichtig sein.

#### Allgemeine Definition

Wird eine quittierpflichtige Meldung durch einen Bediener quittiert, bestätigt der Bediener damit die Bearbeitung bzw. Behebung des Ereignisses, das die Meldung ausgelöst hat.

Nach Bedarf kann die Quittierung der Meldung protokolliert und archiviert werden.

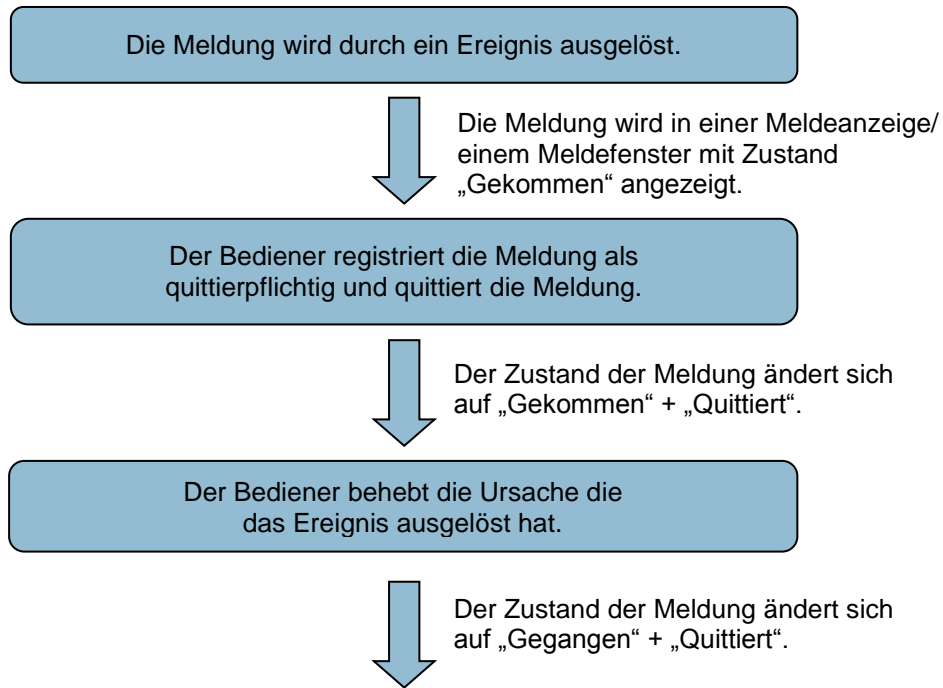
**Hinweis** Die Möglichkeit der Protokollierung und Archivierung von Meldungen ist abhängig vom Bediengerät.

#### Möglichkeiten der Quittierung

Über folgende Möglichkeiten kann die Quittierung einer Meldung in Runtime erfolgen:

- Manuelle Quittierung durch einen berechtigten Bediener am Bediengerät
- Automatische Quittierung durch das System ohne Aktion eines Bedieners über:
  - Variablen
  - Steuerung
  - Systemfunktionen in Funktionslisten
  - Systemfunktionen in Skripten

### Quittierung durch einen Bediener am Bediengerät



In Abhängigkeit von der Projektierung kann ein Bediener in Runtime eine Meldung auf folgende Art quittieren:

- Über die Schaltfläche „Quittieren“ einer Meldeanzeige

**Hinweis** Die Schaltfläche muss vorher unter den Eigenschaften der „Meldeanzeige > Symbolleiste > Schaltflächen“, aktiviert werden.

- Über Funktionstasten und projektierte Schaltflächen mit projektierte Funktion „MeldeanzeigeQuittiereMeldung“ in Bildern
- Über die Quittiertaste <ACK> an einem Bediengerät mit Tastenfront

**Hinweis** Kritische Meldungen sollten nur von berechtigten Bedienern quittiert werden. Versehen Sie aus diesem Grund alle Bedien- und Anzeigeobjekte zum Quittieren einer Meldung mit einer entsprechenden Bedienberechtigung.

### Quittierung durch die Steuerung

Neben dem Quittieren einer Meldung am Bediengerät kann die Quittierung auch über das Steuerungsprogramm erfolgen. Die Quittierung erfolgt dann über die PLC-Quittiervariable, die an jeder quittierpflichtigen Meldung projektiert werden kann. Weitere Informationen zur Projektierung der Quittiervariable können Sie dem Kapitel „Projektierung von Bitmeldungen“ entnehmen.

### Quittierung mehrerer Meldungen zusammen

Um mehrere Meldungen zusammen zu quittieren, müssen diese Meldungen der gleichen Meldegruppe zugeordnet werden. Wird dann eine Meldung aus dieser Meldegruppe quittiert, werden dadurch alle anderen Meldungen aus dieser Meldegruppe quittiert. Es ist dadurch nicht mehr nötig, jede Meldung separat zu quittieren.

Weitere Informationen zu Meldegruppen können Sie dem Kapitel „Einsatz von Meldegruppen“ entnehmen.

## 3 Projektierung von Meldungen in WinCC Professional

In diesem Abschnitt, wird als Ergänzung zum Handbuch (Kapitel [„Arbeiten mit Meldungen“](#)) im Wesentlichen auf die Projektierung von Systemdefinierten Meldungen und Steuerungsmeldungen eingegangen. Die Projektierung von benutzerdefinierten Meldungen ist mit aufgeführt.

### Benötigte Softwarekomponenten

- STEP 7 Professional V14 SP1 (TIA Portal)
- WinCC Professional V14 SP1 (TIA Portal)
- WinCC Runtime Professional V14 SP1 (TIA Portal)

### Voraussetzung

Es existiert bereits ein WinCC (TIA Portal) Projekt mit einer angelegten HMI-Verbindung zwischen der Runtime Professional und der Steuerung.

### 3.1 Projektierung von benutzerdefinierten Meldungen

Die Projektierung von benutzerdefinierten Meldungen (Bitmeldungen, Analogmeldungen und Anwendermeldungen) sind bei WinCC Professional unabhängig von der eingesetzten Steuerung. Eine ausführliche Beschreibung der Projektierung von benutzerdefinierten Meldungen finden Sie im Anwendungsbeispiel [„Projektierung von Meldungen und Alarmer in WinCC \(TIA Portal\)“](#). Außerdem finden Sie im Systemhandbuch von WinCC Professional V14 SP1 unter den Kapiteln [„Projektieren von Bitmeldungen“](#) und [„Projektieren von Analogmeldungen“](#) ebenfalls eine ausführliche Beschreibung.

### 3.2 Projektierung von systemdefinierten Meldungen

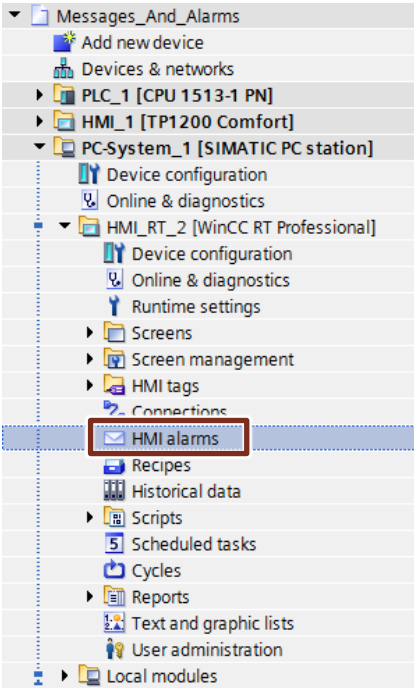
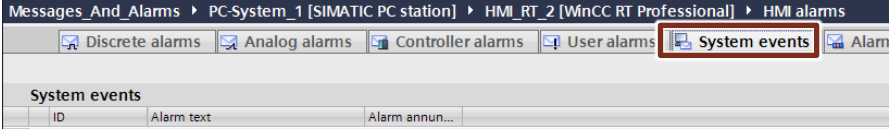
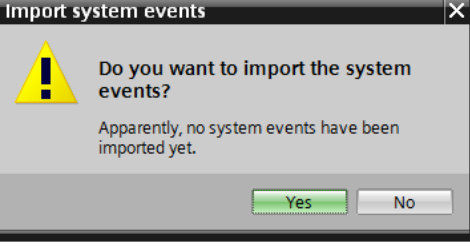
#### 3.2.1 Projektierung von Systemmeldungen (S7-1200, S7-1500)

Systemmeldungen sind im HMI standardmäßig für verschiedene Sprachen hinterlegt. Importieren Sie die Texte zuerst in das Projekt, wenn Sie die Systemmeldungen in weitere Sprachen übersetzen wollen.

#### Importieren von Systemmeldungen

Das Importieren der Systemmeldungen ist nur bei Projekten notwendig, die neu erstellt wurden bzw. wenn die Systemmeldungen noch nicht importiert wurden.

Tabelle 3-1

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihres angelegten Bediengerätes die „HMI-Meldungen“.</p> 
2.	<p>Öffnen Sie die Registerkarte „Systemmeldungen“.</p> 
3.	<p>Bestätigen Sie den nachfolgend erscheinenden Dialog mit „OK“.</p> 
4.	<p>Die Projektierung der Systemmeldungen ist damit abgeschlossen.</p>

### Anzeige der CPU-Systemdiagnosemeldungen in Runtime

Zur Anzeige der CPU-Systemdiagnosemeldungen muss in Ihrem Projekt eine Meldeanzeige projektiert sein.

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Professional unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

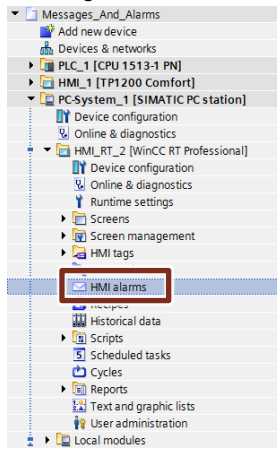
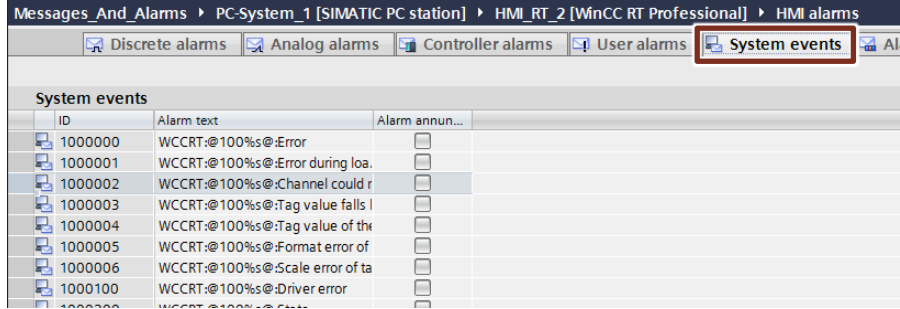
### Meldetexte der Systemmeldungen ändern

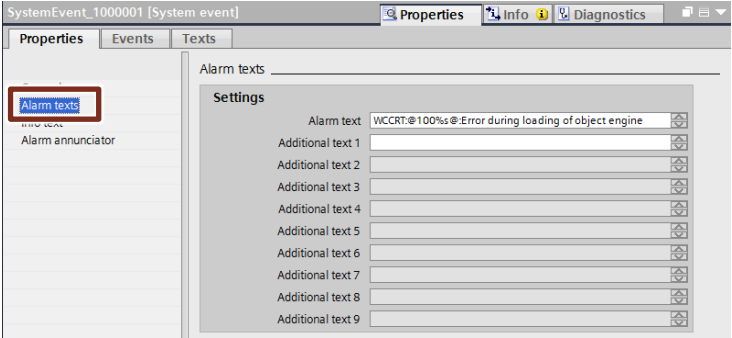
Die Meldetexte der Systemmeldungen können bei Bedarf geändert bzw. angepasst werden. Die dazugehörigen Meldenummern sind nicht änderbar.

#### Hinweis

Bei der Änderung von Systemmeldungen erfolgt eine Veränderung von klar definierten Meldungen, wodurch es unter Umständen zu Fehlinterpretationen kommen kann. Die importierten Meldungen sind Bestandteile des Handbuches und der Online-Hilfe. Eine Änderung hat zur Folge, dass die geänderten Meldungen nicht mehr mit der Dokumentation übereinstimmen.

Tabelle 3-2

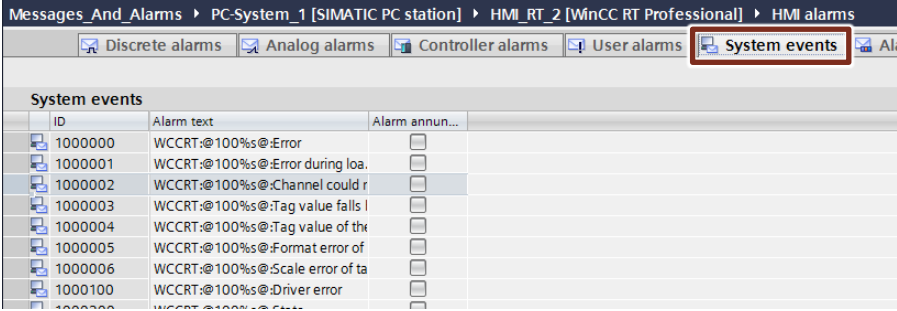
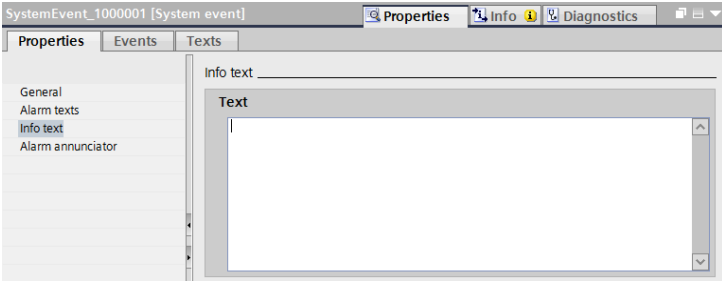
Nr.	Aktion																											
1.	<p>Öffnen Sie in der Projektnavigation unter dem Ordner ihres angelegten Bediengerätes die „HMI-Meldungen“.</p> 																											
2.	<p>Öffnen Sie die Registerkarte „Systemmeldungen“.</p>  <table border="1" data-bbox="464 1332 1367 1556"> <thead> <tr> <th>ID</th> <th>Alarm text</th> <th>Alarm annun...</th> </tr> </thead> <tbody> <tr> <td>1000000</td> <td>WCCRT:@100%:@:Error</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000001</td> <td>WCCRT:@100%:@:Error during loa.</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000002</td> <td>WCCRT:@100%:@:Channel could r</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000003</td> <td>WCCRT:@100%:@:Tag value falls l</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000004</td> <td>WCCRT:@100%:@:Tag value of the</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000005</td> <td>WCCRT:@100%:@:Format error of</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000006</td> <td>WCCRT:@100%:@:Scale error of ta</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000100</td> <td>WCCRT:@100%:@:Driver error</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>Markieren Sie die Systemmeldung deren Meldetext geändert werden soll.</p>	ID	Alarm text	Alarm annun...	1000000	WCCRT:@100%:@:Error	<input type="checkbox"/>	1000001	WCCRT:@100%:@:Error during loa.	<input type="checkbox"/>	1000002	WCCRT:@100%:@:Channel could r	<input type="checkbox"/>	1000003	WCCRT:@100%:@:Tag value falls l	<input type="checkbox"/>	1000004	WCCRT:@100%:@:Tag value of the	<input type="checkbox"/>	1000005	WCCRT:@100%:@:Format error of	<input type="checkbox"/>	1000006	WCCRT:@100%:@:Scale error of ta	<input type="checkbox"/>	1000100	WCCRT:@100%:@:Driver error	<input type="checkbox"/>
ID	Alarm text	Alarm annun...																										
1000000	WCCRT:@100%:@:Error	<input type="checkbox"/>																										
1000001	WCCRT:@100%:@:Error during loa.	<input type="checkbox"/>																										
1000002	WCCRT:@100%:@:Channel could r	<input type="checkbox"/>																										
1000003	WCCRT:@100%:@:Tag value falls l	<input type="checkbox"/>																										
1000004	WCCRT:@100%:@:Tag value of the	<input type="checkbox"/>																										
1000005	WCCRT:@100%:@:Format error of	<input type="checkbox"/>																										
1000006	WCCRT:@100%:@:Scale error of ta	<input type="checkbox"/>																										
1000100	WCCRT:@100%:@:Driver error	<input type="checkbox"/>																										

Nr.	Aktion
3.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Eigenschaften &gt; Meldetexte“.</p>  <p>Ändern Sie unter „Meldetext“ den Meldetext der Systemmeldung.</p> <ul style="list-style-type: none"> <li>• Optional können unter „Zusatztexte“ weitere Texte hinzugefügt werden die die Systemmeldung betreffen.</li> </ul> <p><b>Hinweis</b> Löschen Sie unter keinen Umständen die Platzhalter z. B. @100%#@ in den Systemmeldungen.</p>
4.	Die Änderung des Meldetextes für die Systemmeldung ist damit abgeschlossen.

**Infotext hinzufügen**

Mit einem Infotext können Sie dem Bediener der Anlage weitere Informationen und Handlungshinweise zusätzlich zum Meldetext zur Verfügung stellen.

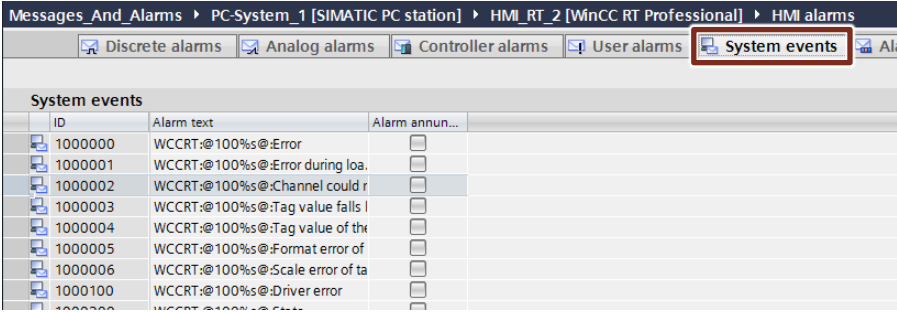
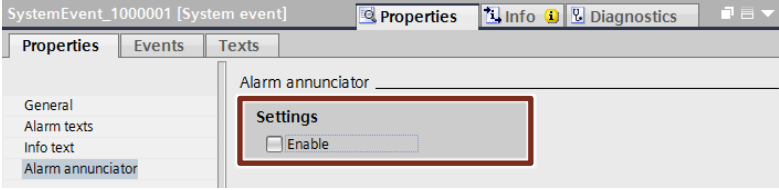
Tabelle 3-3

Nr.	Aktion
1.	<p>Markieren Sie die Systemmeldung in der Registerkarte „Systemmeldungen“.</p> 
2.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Eigenschaften &gt; Infotext“.</p>  <p>Geben Sie unter „Text“ den gewünschten Text ein, der als Infotext zu dieser Systemmeldung erscheinen soll. Über die entsprechende Schaltfläche in der Meldeanzeige kann zu dieser Meldung dann der Infotext aufgerufen werden.</p> <p><b>Hinweis</b> Der Infotext darf maximal 255 Zeichen lang sein.</p>
3.	<p>Die Projektierung des Tooltips für die Systemmeldung ist damit abgeschlossen.</p>

**Zentraler Melder**

Durch die Aktivierung des „Zentralen Melders“ kann eine Meldung zusätzlich zur Meldeanzeige durch ein optisches bzw. akustisches Signal in der Anlage dargestellt werden.

Tabelle 3-4

Nr.	Aktion																											
1.	<p>Markieren Sie die „Systemmeldung“ in der Registerkarte „HMI-Meldungen“.</p>  <table border="1" data-bbox="469 613 1369 835"> <thead> <tr> <th>ID</th> <th>Alarm text</th> <th>Alarm annun...</th> </tr> </thead> <tbody> <tr> <td>1000000</td> <td>WCCRT:@100%\$@:Error</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000001</td> <td>WCCRT:@100%\$@:Error during loa.</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000002</td> <td>WCCRT:@100%\$@:Channel could r</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000003</td> <td>WCCRT:@100%\$@:Tag value falls l</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000004</td> <td>WCCRT:@100%\$@:Tag value of th</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000005</td> <td>WCCRT:@100%\$@:Format error of</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000006</td> <td>WCCRT:@100%\$@:Scale error of ta</td> <td><input type="checkbox"/></td> </tr> <tr> <td>1000100</td> <td>WCCRT:@100%\$@:Driver error</td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	ID	Alarm text	Alarm annun...	1000000	WCCRT:@100%\$@:Error	<input type="checkbox"/>	1000001	WCCRT:@100%\$@:Error during loa.	<input type="checkbox"/>	1000002	WCCRT:@100%\$@:Channel could r	<input type="checkbox"/>	1000003	WCCRT:@100%\$@:Tag value falls l	<input type="checkbox"/>	1000004	WCCRT:@100%\$@:Tag value of th	<input type="checkbox"/>	1000005	WCCRT:@100%\$@:Format error of	<input type="checkbox"/>	1000006	WCCRT:@100%\$@:Scale error of ta	<input type="checkbox"/>	1000100	WCCRT:@100%\$@:Driver error	<input type="checkbox"/>
ID	Alarm text	Alarm annun...																										
1000000	WCCRT:@100%\$@:Error	<input type="checkbox"/>																										
1000001	WCCRT:@100%\$@:Error during loa.	<input type="checkbox"/>																										
1000002	WCCRT:@100%\$@:Channel could r	<input type="checkbox"/>																										
1000003	WCCRT:@100%\$@:Tag value falls l	<input type="checkbox"/>																										
1000004	WCCRT:@100%\$@:Tag value of th	<input type="checkbox"/>																										
1000005	WCCRT:@100%\$@:Format error of	<input type="checkbox"/>																										
1000006	WCCRT:@100%\$@:Scale error of ta	<input type="checkbox"/>																										
1000100	WCCRT:@100%\$@:Driver error	<input type="checkbox"/>																										
2.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Eigenschaften &gt; Zentraler Melder“.</p>  <p>Aktivieren Sie unter den Einstellungen des „Zentralen Melders“ das Optionsfeld.</p>																											
3.	<p>Die Projektierung des Zentralen Melders für die Systemmeldung ist hiermit abgeschlossen.</p>																											

Weitere Informationen zum Zentralen Melder finden Sie im Kapitel [3.6.4](#).



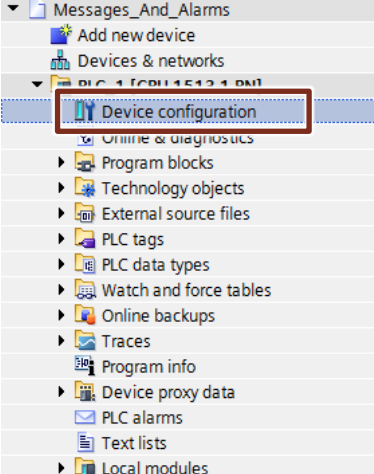
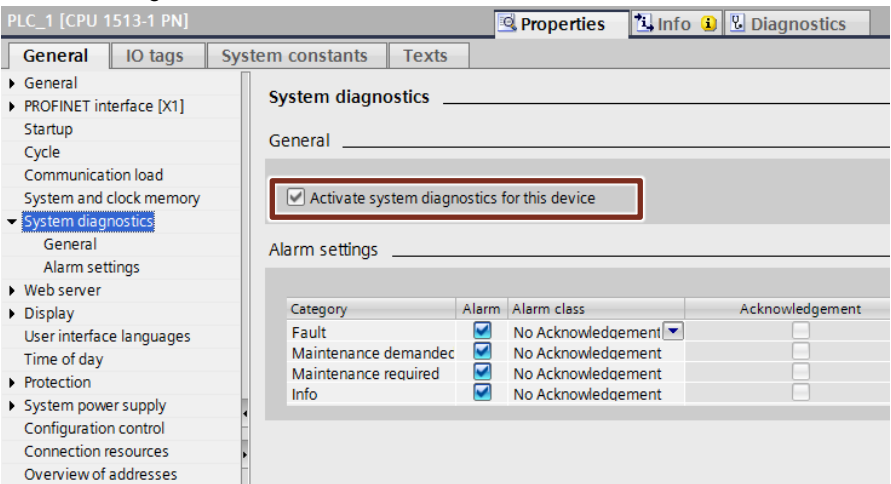
### 3.2.2 Projektierung von CPU-Systemdiagnosemeldungen (S7-1500)

Im folgenden Abschnitt ist die Projektierung der Anzeige von Systemdiagnosemeldungen einer CPU auf einem Bediengerät unter Verwendung folgender Komponenten beschrieben:

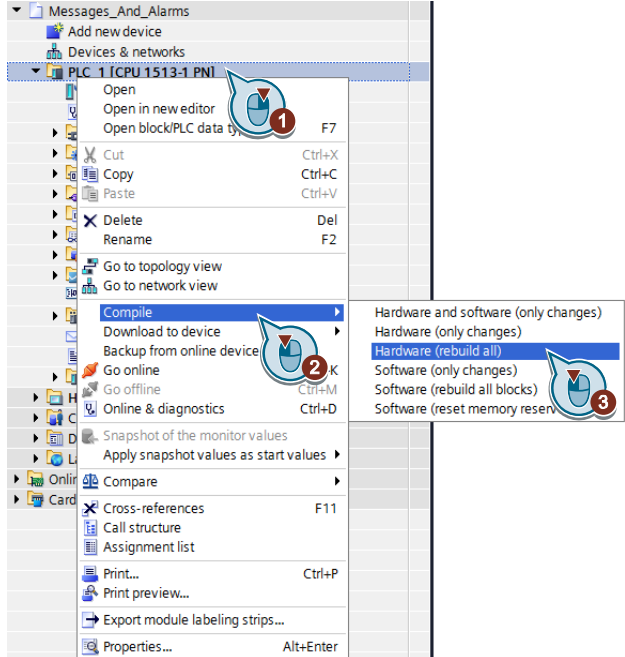
- CPU 1513-1PN
- WinCC Runtime Professional

#### Einstellungen in STEP 7

Tabelle 3-5

Nr.	Aktion																				
1.	<p>Öffnen Sie in der Projektnavigation die „Gerätekongfiguration“ ihrer CPU.</p> 																				
2.	<p>Markieren Sie in der Gerätesicht die CPU auf dem Baugruppenträger.</p>																				
3.	<p>Öffnen Sie im Inspektorfenster das Register „Eigenschaften &gt; Allgemein &gt; Systemdiagnose“.</p> <p>Das Optionskästchen „Systemdiagnose für dieses Gerät aktivieren“ ist standardmäßig aktiviert und kann nicht deaktiviert werden.</p>  <p><b>Optional</b></p> <p>Je nach Bedarf können Sie im unteren Bereich des Inspektorfensters unter „Meldungseinstellungen“ die Meldungen und Meldeklassen der jeweiligen Kategorien anpassen.</p> <table border="1" data-bbox="734 1579 1356 1713"> <thead> <tr> <th>Category</th> <th>Alarm</th> <th>Alarm class</th> <th>Acknowledgement</th> </tr> </thead> <tbody> <tr> <td>Fault</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Maintenance demand</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Maintenance required</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Info</td> <td><input checked="" type="checkbox"/></td> <td>No Acknowledgement</td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	Category	Alarm	Alarm class	Acknowledgement	Fault	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Maintenance demand	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Maintenance required	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>	Info	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>
Category	Alarm	Alarm class	Acknowledgement																		
Fault	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Maintenance demand	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Maintenance required	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		
Info	<input checked="" type="checkbox"/>	No Acknowledgement	<input type="checkbox"/>																		

### 3 Projektierung von Meldungen in WinCC Professional

Nr.	Aktion
4.	<p>Klicken Sie mit der rechten Maustaste auf die CPU in der Projektnavigation. Wählen Sie im Kontextmenü „Übersetzen &gt; Hardware (komplett übersetzen)“ aus.</p> 
5.	Die Einstellungen in STEP 7 sind hiermit abgeschlossen.

#### Anzeige der CPU-Systemdiagnosemeldungen in Runtime

Zur Anzeige der CPU-Systemdiagnosemeldungen muss in Ihrem Projekt eine Meldeanzeige projektiert sein.

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Professional unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

### 3.3 Projektierung von Steuerungsmeldungen

#### 3.3.1 Projektierung von Program\_Alarm (S7-1500)

Die Anweisung „Program\_Alarm: Programmmeldung mit Begleitwerten erzeugen“ überwacht ein Signal und generiert bei einem Signalwechsel am Parameter SIG von 0 auf 1 eine kommende Programmmeldung.

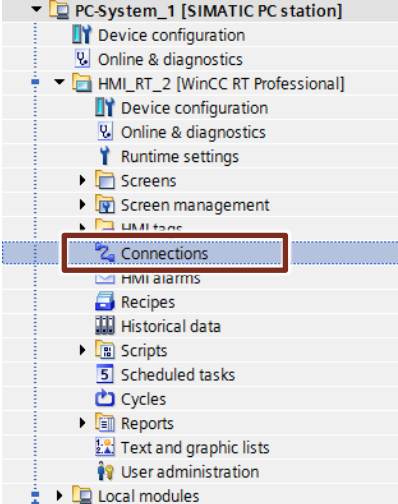
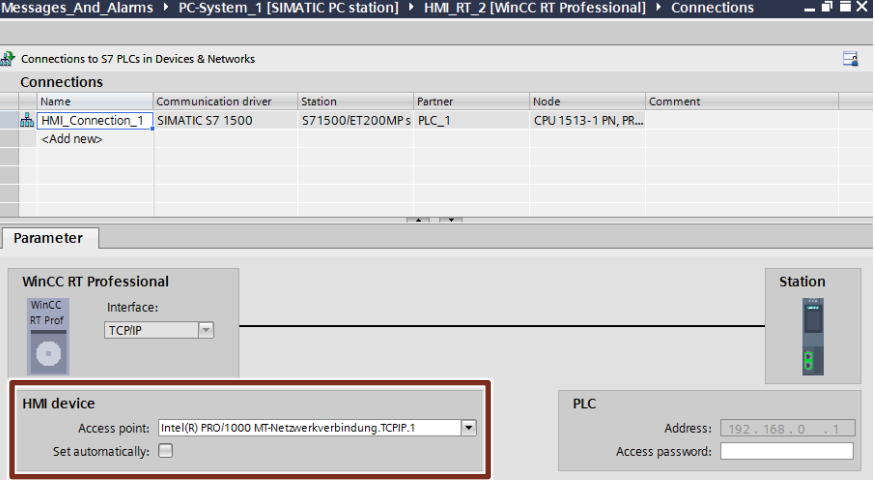
Ausführliche Informationen zur Funktionsweise und den einzelnen Parametern des Meldebausteins sind bereits im Kapitel [2.3.1](#) dieses Dokuments beschrieben.

#### Einstellungen in STEP 7

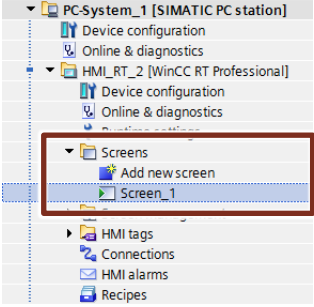
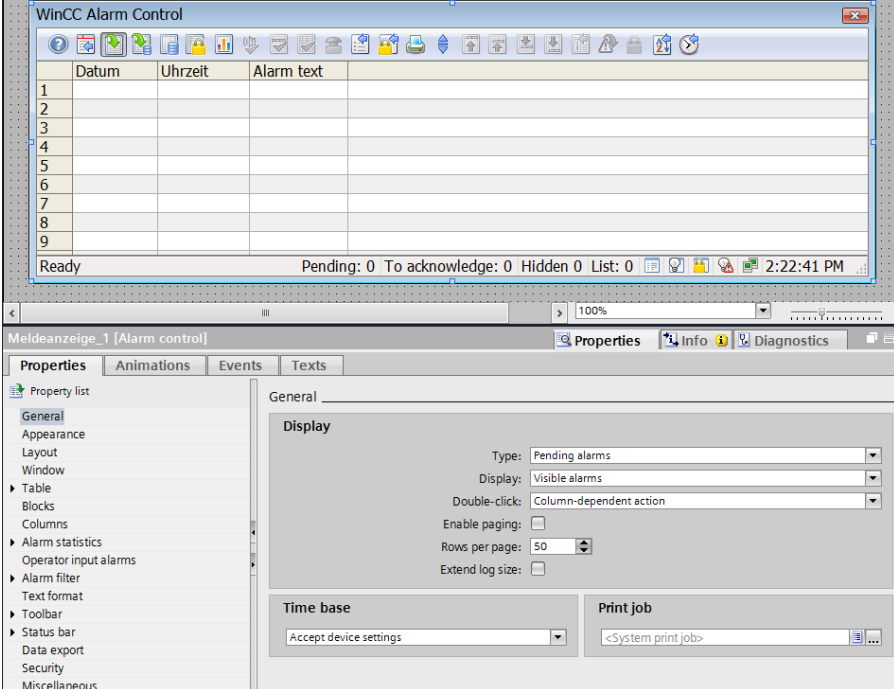
Die notwendigen Projektierungsschritte, welche in STEP 7 für den Meldebaustein „Program\_Alarm“ notwendig sind, finden Sie im Kapitel [2.3.1](#) dieses Dokuments.

#### Einstellungen in WinCC Professional

Tabelle 3-6

Nr.	Aktion
1.	<p>Öffnen Sie in der Projektnavigation den Ordner der angelegten PC-Station und wählen „Verbindungen“ aus.</p> 
2.	<p>Wählen Sie als Zugangspunkt des Bediengeräts Ihre verwendete Netzwerkkarte aus der Klappliste aus und deaktivieren Sie das Optionsfeld „Automatisch setzen“.</p> 

### 3 Projektierung von Meldungen in WinCC Professional

Nr.	Aktion
3.	<p>Öffnen Sie unter der angelegten PC-Station den Ordner „Bilder“ und fügen Sie ein neues Bild hinzu.</p> 
4.	Legen Sie im „Bild_1“ eine Meldeanzeige an.
5.	Starten Sie die Runtime Professional.
6.	<p><b>Optional</b> Die Darstellung und Eigenschaften der Meldeanzeige können Sie über das Register „Eigenschaften“ und die Bereichsnavigation individuell einstellen.</p> 
7.	Die Projektierung in WinCC Professional ist hiermit abgeschlossen.

Weitere Informationen zur Projektierung und Konfiguration einer Meldeanzeige finden Sie im Systemhandbuch von WinCC Professional unter dem Themenpunkt [„Meldeanzeige projektieren“](#).

### 3.3.2 Projektierung von Get\_AlarmState (S7-1500)

Mit der Anweisung „Meldungszustand ausgeben“ (Get\_AlarmState) geben Sie den Meldungszustand einer Programmmeldung aus.

Eine ausführliche Beschreibung dieser Anweisung erfolgte bereits unter dem Thema „[Projektierung von Meldungen in WinCC Basic/Comfort/Advanced](#)“ im Unterkapitel „[Projektierung von Get\\_AlarmState](#)“. Diese Projektierungsschritte können Sie auch für WinCC Professional übernehmen.

### 3.3.3 Projektierung von Gen\_UsrMsg (S7-1200, S7-1500)

Die Anweisung „Gen\_UsrMsg“ erzeugt eine Meldung, die in den Diagnosepuffer eingetragen wird.

Diese Anweisung wurde ebenfalls unter dem vorangegangenen Kapitel „[Projektierung von Meldungen in WinCC Basic/Comfort/Advanced](#)“ behandelt. Eine ausführliche Schritt-für-Schritt Anleitung dieser Anweisung finden Sie im Unterkapitel „[Projektierung von Gen\\_UsrMsg](#)“.

## 3.4 Einsatz von Meldeklassen

Während des Betriebs einer Anlage treten Meldungen unterschiedlicher Prioritäten auf. Um diese Meldungen zu ordnen, teilt man diese in Abhängigkeit der Wichtigkeit in verschiedene Meldeklassen ein. Der Bediener der Anlage kann dadurch erkennen, welche Meldungen eine höhere Priorität und Wichtigkeit haben und vorrangig behandelt werden müssen.

### Funktion der Meldeklassen

Beim Anlegen von neuen Meldungen muss jede Meldung einer Meldeklasse zugeordnet werden.

Die Meldeklasse definiert dabei:

- Die Darstellung
- Das Quittierkonzept (Einfachquittierung, Zweifachquittierung, zentraler Melder)
- Die Archivierung der Meldung

### Hinweis

Der Funktionsumfang der Meldeklasse ist abhängig vom projektierten Bediengerät.

### Arten von Meldeklassen

Folgende Arten von Meldeklassen stehen in WinCC (TIA Portal) zur Verfügung.

- **Vordefinierte Meldeklassen**

Für jedes Bediengerät finden Sie im Projektbaum unter „HMI-Meldungen“ in der Registerkarte „Meldeklassen“ vordefinierte Meldeklassen. Diese Meldeklassen können nicht gelöscht und nur bedingt bearbeitet werden.

- **Benutzerdefinierte Meldeklassen**

Für jedes Bediengerät können Sie im Projektbaum unter „HMI-Meldungen“ in der Registerkarte „Meldeklassen“ weitere Meldeklassen anlegen. In diesen Meldeklassen kann die Darstellung und das Quittierkonzept der dazugehörigen Meldungen individuell projektiert werden.

- **Projektweite Meldeklassen**

Projektweite Meldeklassen können Sie bediengeräteübergreifend im Projekt verwenden. Im Projektbaum unter „Gemeinsame Daten > Meldeklassen“ werden diese Meldeklassen angezeigt und neue projektweite Meldeklassen projektiert.

## 3.5 Einsatz von Meldegruppen

Während des Betriebs einer Anlage treten Meldungen auf, die verschiedene Prozesse und Bereiche betreffen. Um diese Meldungen zu gliedern (z. B. nach Anlagenteilen) und eine Übersichtlichkeit zu schaffen, kann man Meldungen zu Meldegruppen hinzufügen.

### Arten von Meldegruppen

Folgende Arten von Meldegruppen stehen in WinCC (TIA Portal) zur Verfügung.

- **Benutzerdefinierte Meldegruppen**

Benutzerdefinierte Meldegruppen werden nach Bedarf angelegt. Diese Meldegruppen enthalten quittierpflichtige Meldungen und können weitere untergeordnete Meldegruppen enthalten. Untergeordnete Meldegruppen können hierarchisch mit bis zu fünf Untergruppen aufgebaut werden.

- **Meldegruppen aus Meldeklassen**

Für jede vordefinierte Meldeklasse ist in WinCC Professional bereits eine Meldegruppe angelegt. Diese Meldegruppen nennt man Klassengruppen. Alle Meldungen dieser Meldeklasse sind auch in der dazugehörigen Klassengruppe enthalten.

### Funktionsumfang von Meldegruppen in WinCC Professional

In WinCC Professional enthält eine Meldegruppe mehrere Variablen. Diese Variablen sprechen die Eigenschaften aller Meldungen an, die dieser Meldegruppe zugeordnet sind. Werden Einstellungen in einer Meldeklasse vorgenommen werden die Einstellungen in der dazugehörigen Meldegruppe aktualisiert.

### Verwendungszweck

Der Einsatz von Meldegruppen wird empfohlen um:

- Meldungen mit gleicher Störungsursache zusammenzufassen
- Gleichartige Meldungen zusammenzufassen
- Meldungen aus einem Anlagenteil zu beobachten und zu sortieren
- Meldungen, die Teil eines Prozesses sind, zu beobachten und zu sortieren
- Meldungen eines Anlagenteils gemeinsam zu bearbeiten (z. B. Quittieren, Sperren, Anzeige unterdrücken)
- Zustände von Anlagenteilen zu visualisieren
- Prozesse der Anlage zu unterbinden (z. B. durch das Quittieren von Meldungen)

## 3.6 Das Quittierkonzept

Das Quittierkonzept legen Sie für eine Meldeklasse fest. Alle Meldungen die Bestandteil dieser Meldeklasse sind, werden dann nach diesem Konzept quittiert.

In WinCC Professional gibt es grundsätzlich folgende Quittierkonzepte, die in weitere Unterkonzepte unterteilt sind:

- Meldung ohne Quittierung
- Meldung mit Einfachquittierung
- Meldung mit Zweifachquittierung

Im folgenden Verlauf des Kapitels werden diese Quittierkonzepte näher erläutert.

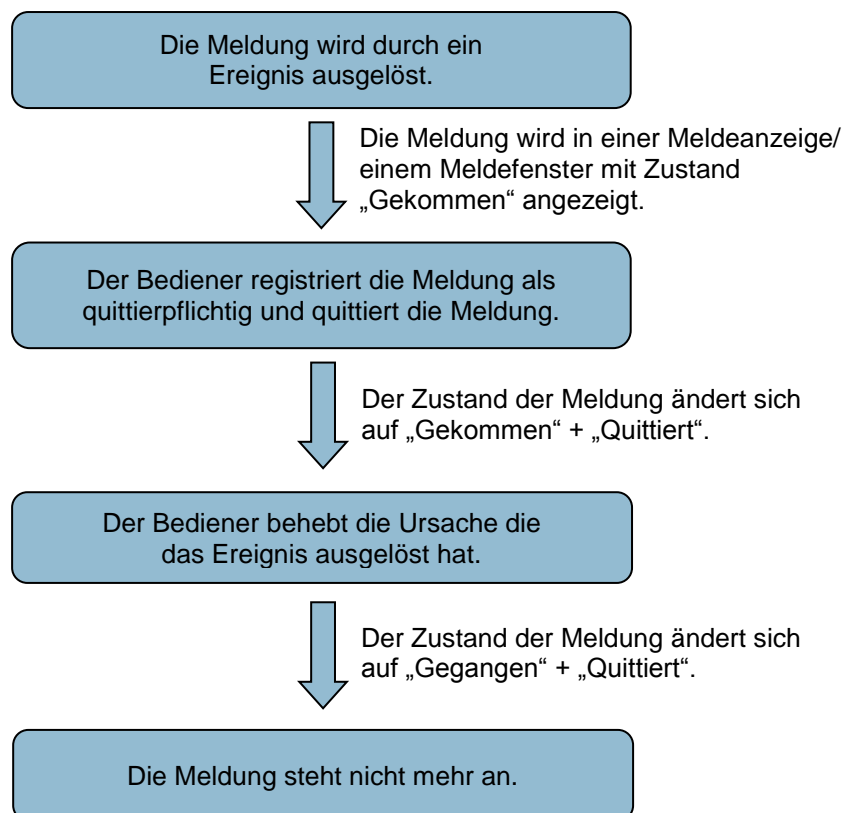
### 3.6.1 Meldung ohne Quittierung

Bei einer Meldung ohne Quittierung muss der Bediener nicht zwingend auf die Meldung reagieren. Die Meldung „kommt“ und „geht“, ohne dass eine Quittierung erforderlich ist.

#### Meldung ohne Zustand „Gegangen“ ohne Quittierung

Solange das Ereignis ansteht, das die Meldung ausgelöst hat, steht die Meldung weiterhin an. Ist das Ereignis, das die Meldung ausgelöst hat, „gegangen“, wird die Meldung nicht mehr angezeigt. Die Meldung wird archiviert und muss dabei nicht quittiert werden.

### 3.6.2 Meldung mit Einfachquittierung





Eine Meldung mit Einfachquittierung erfordert eine Quittierung, sobald das Ereignis, das die Meldung auslöst, eingetreten ist. Die Meldung steht solange an, bis der Bediener die Meldung quittiert hat.

#### Meldung ohne Zustand „Gegangen“ mit Quittierung

Die Meldung steht an, solange das Ereignis, das die Meldung ausgelöst hat, ansteht, und die Meldung nicht quittiert wurde. Ist das Ereignis „gegangen“ und die Meldung quittiert, wird die Meldung nicht mehr angezeigt. Die Meldung wird archiviert.

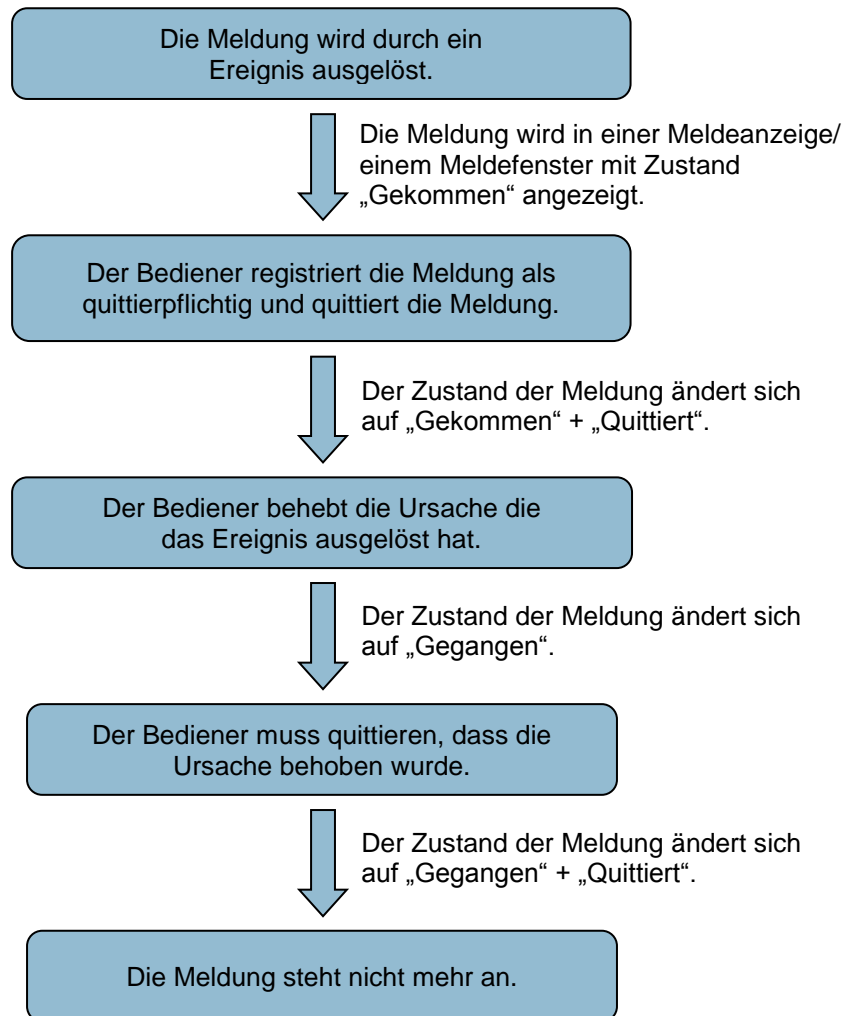
#### Erstwertmeldung mit Blinken und Einfachquittierung

Eine Erstwertmeldung ist eine Meldung innerhalb einer Meldeklasse. Die Meldung, deren Zustand sich seit dem letzten Quittieren als erstes ändert, wird durch Blinken im Meldefenster hervorgehoben. Die Meldung wird dann als Erstwertmeldung bezeichnet.

#### Neuwertmeldung mit Blinken und Einfachquittierung

Neuwertmeldungen sind Meldungen innerhalb einer Meldeklasse. Die Meldungen, deren Zustände sich seit dem letzten Quittieren geändert haben, werden durch Blinken im Meldefenster hervorgehoben.

#### 3.6.3 Meldung mit Zweifachquittierung



Eine Meldung mit Zweifachquittierung erfordert eine Quittierung, sobald das meldeauslösende Ereignis eingetreten ist, sowie eine weitere Quittierung, wenn das Ereignis nicht mehr ansteht. Die Meldung bleibt anstehen bis der Bediener die Meldung zweifach quittiert hat.

#### **Neuwertmeldung mit Blinken und Zweifachquittierung**

Neuwertmeldungen sind Meldungen innerhalb einer Meldeklasse. Die Meldungen, deren Zustände sich seit dem letzten Quittieren geändert haben, werden durch blinken im Meldefenster hervorgehoben. Diese Meldungen müssen bei den Zuständen „Kommen“ und „Gehen“ quittiert werden.

### 3.6.4 Allgemeine Informationen zu Quittierkonzepten

#### Quittierung mehrerer Meldungen zusammen

Die Quittierung von mehreren Meldungen zusammen kann in WinCC Professional über eine Sammelquittierung oder eine Meldegruppe erfolgen.

- **Sammelquittierung über die Meldeanzeige**  
Über die Schaltfläche „Sammelquittieren“ einer Meldeanzeige lassen sich alle sichtbaren, anstehenden und quittierpflichtigen Meldungen mit einem mal quittieren.

**Hinweis** Meldungen mit aktivierter Einzelquittierung lassen sich nicht über die Sammelquittierung quittieren.

- **Quittierung über Meldegruppen**  
Um mehrere quittierpflichtige Meldungen zusammen zu quittieren, müssen diese Meldungen der gleichen Meldegruppe zugeordnet werden. Wird dann eine Meldung aus dieser Meldegruppe quittiert, werden dadurch auch alle anderen Meldungen aus dieser Meldegruppe quittiert. Es ist dadurch nicht mehr nötig, jede Meldung separat zu quittieren.

Weitere Informationen zu Meldegruppen finden Sie in Kapitel [3.5](#).

#### Zentraler Melder

Wurde bei einer quittierpflichtigen Meldung der zentrale Melder projiziert, gibt es folgende Möglichkeiten, den zentralen Melder zu quittieren:

- Der Bediener quittiert den zentralen Melder zusammen mit der quittierpflichtigen Meldung.
- Der Bediener quittiert den zentralen Melder über die Schaltfläche „Quittierung Zentraler Melder“ an einer Meldeanzeige.
- Die Quittierung des zentralen Melders erfolgt über eine Variable.

Die Einstellungen zur Quittierung des zentralen Melders erfolgt in den Einstellungen der Meldeklasse. Diese Einstellungen gelten separat für jede Meldeklasse mit Quittierpflicht.

#### Not-Quittierung

Über die Schaltfläche „Not-Quittieren“ einer Meldeanzeige lassen sich quittierpflichtige Meldungen im Notfall direkt anhand der Meldenummer quittieren.

**Hinweis** Das Quittierbit wird auch an die Steuerung gesendet, wenn die mit der entsprechenden Meldenummer angegebene Meldung nicht ansteht.

Verwenden Sie die Not-Quittierung nur im Notfall.

**Hinweis** Weitere Informationen über das Quittieren von Meldungen in WinCC Professional finden Sie unter folgendem Link:

<http://support.automation.siemens.com/WW/view/de/55622122>

# 4 Wissenswertes

## 4.1 Grundlagen

### 4.1.1 Übersicht über die Meldeverfahren

#### Einleitung

Das Meldesystem in WinCC (TIA Portal) ermöglicht die Anzeige und die Aufzeichnung von Betriebszuständen und Störungen, die in einer Anlage oder auf einem Bediengerät anstehen oder auftreten können.

#### Überblick über das Meldesystem

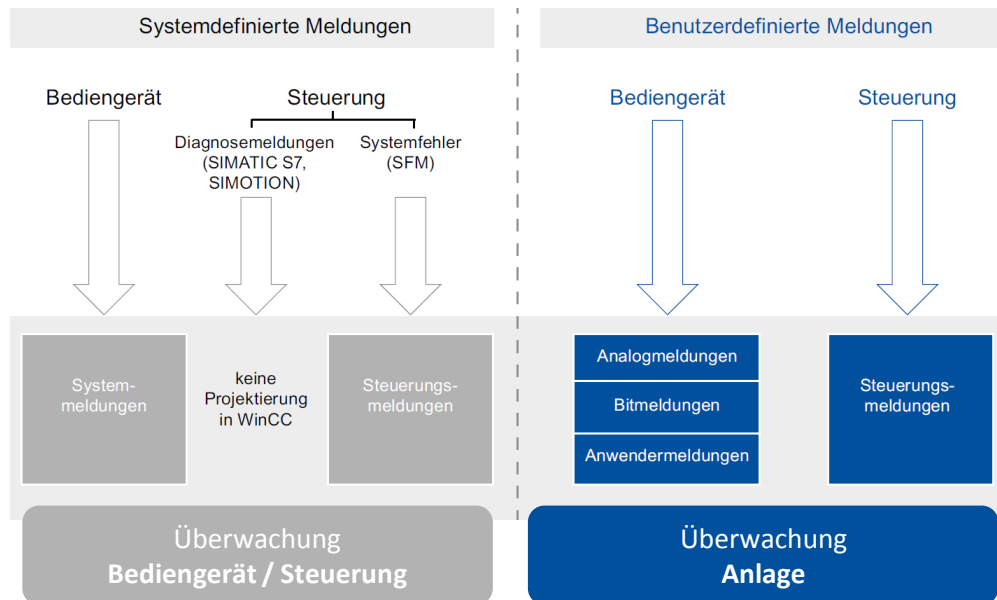
Das Meldesystem verarbeitet verschiedene Meldeverfahren des Bediengerätes und der Steuerung. Die Meldeverfahren werden in Systemdefinierte und Benutzerdefinierte Meldungen unterteilt:

- Systemdefinierte Meldungen dienen zur Überwachung des Bediengerätes und der Steuerung.
- Benutzerdefinierte Meldungen dienen zur Überwachung der Anlage.

Die erfassten Meldeereignisse werden am Bediengerät angezeigt. Durch einen gezielten Zugriff auf die Meldungen und ergänzende Informationen zu den einzelnen Meldungen wird eine zügige Störungslokalisierung gewährleistet. Stillstandzeiten werden dabei verringert oder ganz vermieden.

Das folgende Bild zeigt den Aufbau des Meldesystems in WinCC (TIA Portal):

Abbildung 4-1



**Hinweis** Steuerungsmeldungen und Anwendermeldungen müssen vom Bediengerät unterstützt werden, siehe hierzu auch [Tabelle 4-2](#) auf Seite 55.

## 4.1.2 Benutzerdefinierte Meldungen

Die benutzerdefinierten Meldeverfahren dienen zur Überwachung des Anlagenprozesses. Dabei sind diese nach der Art der Information, die zum Auslösen der Meldung benötigt werden, bezeichnet.

Benutzerdefinierte Meldeverfahren bestehen aus folgenden Meldungen:

- Analogmeldungen
- Bitmeldungen
- Steuerungsmeldungen
- Anwendermeldungen

### Analogmeldungen

Eine Analogmeldung zeigt Grenzwertverletzungen einer Variablen im laufenden Betrieb an. Ausgelöst wird eine Analogmeldung bei der Über-/ Unterschreitung eines vorher definierten Grenzwertes.

### Bitmeldungen

Eine Bitmeldung zeigt Zustandsänderungen im laufenden Betrieb an. Ausgelöst wird eine Bitmeldung bei einem bestimmten Wert (Bit) einer Variablen.

### Steuerungsmeldungen

Steuerungsmeldungen zeigen Zustandswerte der Steuerung im laufenden Betrieb an.

### Anwendermeldungen

Eine Anwendermeldung überwacht Bedienaktionen in der WinCC Runtime Professional im laufenden Betrieb. Ausgelöst werden Anwendermeldungen über das Triggern der Meldenummer. Eine Anwendermeldung kann z. B. folgende Informationen enthalten:

- Art und Inhalt der quittierten Meldung
- Zeitpunkt der Quittierung
- Bediener
- Datum

### Hinweis Geräteabhängigkeit

Steuerungsmeldungen und Anwendermeldungen stehen nicht für alle Bediengeräte zur Verfügung.

### Unterstützte Bausteine

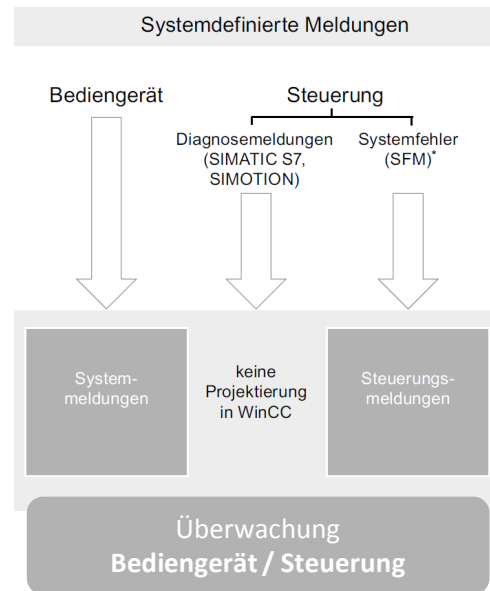
- Program\_Alarm (Programmmeldung mit Begleitwerten erzeugen, nur S7-1500)
- Get\_AlarmState (Meldungszustand ausgeben, nur S7-1500)
- Gen\_UsrMsg (Anwenderdiagnosemeldungen erzeugen, S7-1200 und S7-1500).

### 4.1.3 Systemdefinierte Meldungen

Das systemdefinierte Meldeverfahren dient der Überwachung des Bediengerätes oder der Steuerung. Das Meldeverfahren besteht dabei aus den Meldungen:

- Systemdefinierte Steuerungsmeldungen
- Systemmeldungen

Abbildung 4-2



**Hinweis** Die Systemfehler (SFM) sind nur in Verbindung mit einer S7-300/400 zu verwenden. Bei einer S7-1200/S7-1500 werden die Systemfehler über die integrierte Systemdiagnose gemeldet.

#### Systemdefinierte Steuerungsmeldungen

Eine systemdefinierte Steuerungsmeldung dient der Überwachung von Zuständen und Ereignissen einer SIMATIC S7 Steuerung. Auf einem Bediengerät lassen sich dabei Diagnosemeldungen einer SIMATIC S7-1500 Steuerung anzeigen.

**Hinweis** Die Systemdiagnose ist in der Steuerungsfamilie S7-1500 integriert und kann über den Diagnosepuffer oder der Meldeanzeige am HMI ausgelesen werden.

#### Systemmeldungen

Eine Systemmeldung wird am Bediengerät ausgegeben und dient der Überwachung interner Zustände eines Bediengerätes oder einer Steuerung im laufenden Betrieb. Systemmeldungen geben Auskunft über den Status des Systems und zeigen zum Beispiel Kommunikationsfehler zwischen einem Bediengerät und einer Steuerung an.

#### 4.1.4 Verfügbarkeit der Meldeverfahren

In diesem Kapitel finden Sie verschiedene Übersichten zur Verfügbarkeit der Meldeverfahren.

##### Übersicht Bediengeräte und unterstützte Meldetypen bei S7-1200

In der nachfolgenden Tabelle ist die Verfügbarkeit der unterschiedlichen Meldetypen in Abhängigkeit vom eingesetzten Bediengerät für die Steuerungsfamilie S7-1200 dargestellt.

Tabelle 4-1

Bediengerät	Analog-meldungen	Bit-Meldungen	Steuerungs-meldungen	Anwender-meldungen	Diagnose-meldungen	System-meldungen
Basic Panel	X	X	-	-	-	X
Mobile Panels	X	X	X	-	-	X
Comfort Panels	X	X	X	-	-	X
WinCC RT Advanced	X	X	X	-	-	X
WinCC RT Professional	X	X	X	-	-	X

##### Übersicht Bediengeräte und unterstützte Meldetypen bei S7-1500

In der nachfolgenden Tabelle ist die Verfügbarkeit der unterschiedlichen Meldetypen in Abhängigkeit vom eingesetzten Bediengerät für die Steuerungsfamilie S7-1500 dargestellt.

Tabelle 4-2

Bediengerät	Analog-meldungen	Bit-Meldungen	Steuerungs-meldungen	Anwender-meldungen	Diagnose-meldungen	System-meldungen
Basic Panel	X	X	-	-	-	X
Mobile Panels	X	X	X	-	X	X
Comfort Panels	X	X	X	-	X	X
WinCC RT Advanced	X	X	X	-	X	X
WinCC RT Professional	X	X	X	X	X	X

##### Übersicht Meldebausteine S7-1500

Die nachfolgende Tabelle zeigt die Meldebausteine für Steuerungsmeldungen einer S7-1500.

Tabelle 4-3

Meldebaustein	FB/FC	Quittierung	Kanäle zu überwachende Signale	Begleitwerte
Program_Alarm	FB700	Je nach Meldeklasse	1	Bis zu 10

##### Hinweis

Die Anzahl der projektierbaren Meldebausteine ist abhängig von der eingesetzten SIMATIC S7-1500 Steuerung. Genauere Informationen finden Sie im jeweiligen Gerätehandbuch der SIMATIC Steuerung.

## 5 Anhang

### 5.1 Service und Support

#### Industry Online Support

Sie haben Fragen oder brauchen Unterstützung?

Über den Industry Online Support greifen Sie rund um die Uhr auf das gesamte Service und Support Know-how sowie auf unsere Dienstleistungen zu.

Der Industry Online Support ist die zentrale Adresse für Informationen zu unseren Produkten, Lösungen und Services.

Produktinformationen, Handbücher, Downloads, FAQs und Anwendungsbeispiele – alle Informationen sind mit wenigen Mausklicks erreichbar:

<https://support.industry.siemens.com>

#### Technical Support

Der Technical Support von Siemens Industry unterstützt Sie schnell und kompetent bei allen technischen Anfragen mit einer Vielzahl maßgeschneiderter Angebote – von der Basisunterstützung bis hin zu individuellen Supportverträgen.

Anfragen an den Technical Support stellen Sie per Web-Formular:

[www.siemens.de/industry/supportrequest](http://www.siemens.de/industry/supportrequest)

#### SITRAIN – Training for Industry

Mit unseren weltweit verfügbaren Trainings für unsere Produkte und Lösungen unterstützen wir Sie mit innovativen Lernmethoden.

Mehr zu den angebotenen Trainings und Kursen sowie deren Standorte und Termine erfahren Sie unter:

[www.siemens.de/sitrain](http://www.siemens.de/sitrain)

#### Hinweis

Grundlagen zu WinCC werden im SITRAIN-Kurs „SIMATIC WinCC maschinennah im TIA Portal“ und „SIMATIC WinCC SCADA im TIA Portal“ vermittelt.

- [SIMATIC WinCC maschinennah im TIA Portal \(de\)](#)
- [SIMATIC WinCC on the machine level in the TIA Portal \(en\)](#)
- [SIMATIC WinCC SCADA im TIA Portal \(de\)](#)
- [SIMATIC WinCC SCADA in the TIA Portal \(en\)](#)

Grundlagen zu STEP 7 werden im SITRAIN-Kurs „SIMATIC Programmieren 1 im TIA Portal“ und „SIMATIC S7-1200 Basiskurs“ vermittelt.

- [SIMATIC Programmieren 1 im TIA Portal \(de\)](#)
- [SIMATIC programming 1 in the TIA Portal \(en\)](#)
- [SIMATIC S7-1200 Basiskurs \(de\)](#)
- [SIMATIC S7-1200 basic course \(en\)](#)



## Serviceangebot

Unser Serviceangebot umfasst folgendes:

- Plant Data Services
- Ersatzteilservices
- Reparaturservices
- Vor-Ort und Instandhaltungsservices
- Retrofit- und Modernisierungsservices
- Serviceprogramme und Verträge

Ausführliche Informationen zu unserem Serviceangebot finden Sie im Servicekatalog:

<https://support.industry.siemens.com/cs/sc>

## Industry Online Support App

Mit der App "Siemens Industry Online Support" erhalten Sie auch unterwegs die optimale Unterstützung. Die App ist für Apple iOS, Android und Windows Phone verfügbar:

<https://support.industry.siemens.com/cs/ww/de/sc/2067>

## 5.2 Links und Literatur

Tabelle 5-1

	Thema
\1\	Siemens Industry Online Support <a href="https://support.industry.siemens.com">https://support.industry.siemens.com</a>
\2\	Downloadseite des Beitrages <a href="https://support.industry.siemens.com/cs/ww/de/view/62121503">https://support.industry.siemens.com/cs/ww/de/view/62121503</a>
\3\	Systemhandbuch STEP 7 Professional V14 SP1 <a href="https://support.industry.siemens.com/cs/ww/de/view/109747136">https://support.industry.siemens.com/cs/ww/de/view/109747136</a>
\4\	Systemhandbuch WinCC Advanced V14.0 SP1 <a href="https://support.industry.siemens.com/cs/ww/de/view/109747174">https://support.industry.siemens.com/cs/ww/de/view/109747174</a>
\5\	Systemhandbuch WinCC Professional V14.0 SP1 <a href="https://support.industry.siemens.com/cs/ww/de/view/109747178">https://support.industry.siemens.com/cs/ww/de/view/109747178</a>
\6\	Anwendungsbeispiel: System-Diagnoseanzeige unter Verwendung von WinCC (TIA Portal) und SIMATIC Comfort Panels. <a href="https://support.industry.siemens.com/cs/ww/de/view/61910954">https://support.industry.siemens.com/cs/ww/de/view/61910954</a>
\7\	Anwendungsbeispiel: Systemdiagnose mit S7-1500 und TIA Portal <a href="https://support.industry.siemens.com/cs/ww/de/view/68011497">https://support.industry.siemens.com/cs/ww/de/view/68011497</a>
\8\	FAQ: Wie kann der Diagnosepuffer einer SIMATIC CPU mit integriertem Webserver an einem SIMATIC Panel angezeigt werden? <a href="https://support.industry.siemens.com/cs/ww/de/view/59601288">https://support.industry.siemens.com/cs/ww/de/view/59601288</a>
\9\	FAQ: Wie können Alarm-Meldungen ab WinCC Runtime Professional V11 quittiert werden? <a href="https://support.industry.siemens.com/cs/ww/de/view/55622122">https://support.industry.siemens.com/cs/ww/de/view/55622122</a>

## 5.3 Änderungsdokumentation

Tabelle 5-2

Version	Datum	Änderung
V1.0	07/2016	Erste Ausgabe
V2.0	05/2018	Steuerungsfamilie S7-1200 aufgenommen