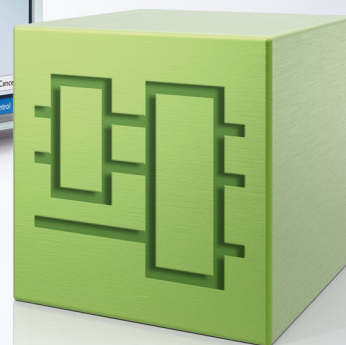
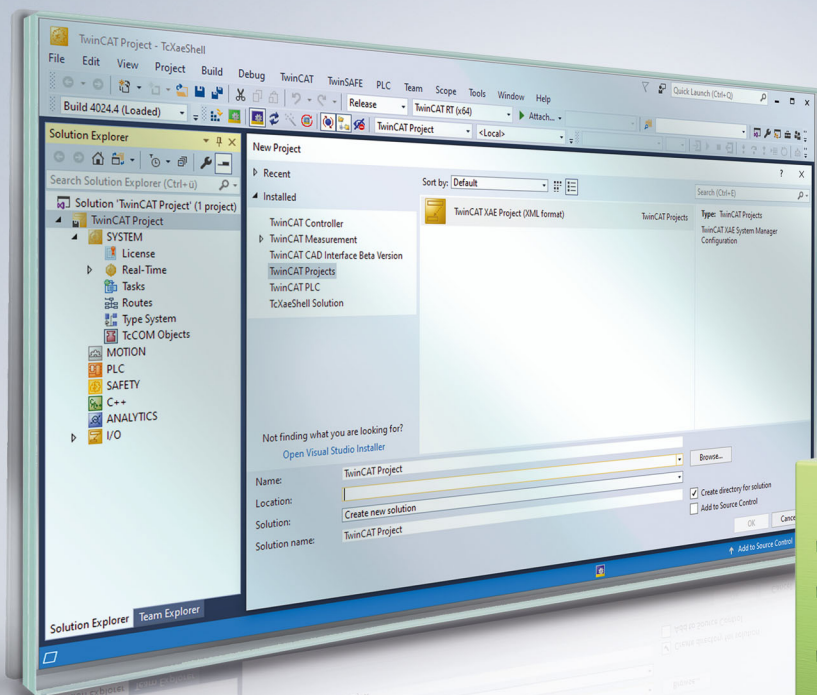


BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2_SystemCX



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Funktionsbausteine	9
3.1	[veraltet]	9
3.1.1	FB_CxGetDeviceIdentification	9
3.1.2	FB_CxGetDeviceIdentificationEx	10
3.1.3	FB_CX1010SetWatchdog	11
3.1.4	FB_CX1020SetWatchdog	11
3.1.5	FB_CX1030SetWatchdog	12
3.2	FB_CXProfiler	13
3.3	FB_CXReadKBusCycleUpdateTime	14
3.4	FB_CXReadKBusError	15
3.5	FB_CXSetTextDisplay	16
3.6	FB_CXSetTextDisplayUSB	18
3.7	FB_CXGetTextDisplayUSB.....	20
3.8	FB_CXSimpleUps	22
3.9	FB_CX5010SetWatchdog	24
3.10	FB_CX5020SetWatchdog	25
3.11	FB_CX7000_LED_ERR	26
3.12	FB_CX7000_LED_WD.....	27
3.13	FB_CX7080_LED_ERR	28
3.14	FB_CX7080_LED_WD.....	29
3.15	FB_CX70xx_RW_EEPROM	31
3.16	FB_CX70xx_ResetOnBoardIO	32
4	Funktionen.....	34
4.1	[veraltet]	34
4.1.1	F_GetVersionTcSystemCX.....	34
4.1.2	F_GetVersionTcSystemCX1000	34
4.1.3	F_GetVersionTcSystemCX1010	34
4.1.4	F_GetVersionTcSystemCX1020	35
4.1.5	F_GetVersionTcSystemCX1030	35
4.1.6	F_GetVersionTcSystemCX5010	35
4.1.7	F_GetVersionTcSystemCX5020	36
4.1.8	F_GetVersionTcSystemCX9000	36
4.1.9	F_GetVersionTcSystemCX9010	36
4.1.10	F_CXSubTimeStamp	37
4.1.11	F_CX1000SetWatchdog	37
4.1.12	F_CX9000SetWatchdog	38
4.1.13	F_CX9010SetWatchdog	39
4.2	F_CXNaviSwitch	39

4.3	F_CXNaviSwitchUSB	40
4.4	F_CX81xx_ADDRESS	41
4.5	F_CX8180_LED_ERR	42
4.6	F_CX8180_LED_WD	42
4.7	F_CX8190_LED_ERR	43
4.8	F_CX8190_LED_WD	43
5	Datentypen	45
5.1	[veraltet]	45
5.1.1	ST_CxDeviceIdentification	45
5.1.2	ST_CxDeviceIdentificationEx	45
5.2	ST_CX_ProfilerStruct	46
5.3	E_CX81x0_LED	47
6	Globale Konstanten	48
6.1	Bibliotheksversion	48

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie [hier](#).

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den [RSS Feed](#).

2 Übersicht

In dieser Bibliothek sind Funktionen und Funktionsbausteine enthalten, die Features auf den Geräten der Embedded PC CX Linie nutzen.

Funktionsbausteine

Name	Beschreibung
FB_CXProfiler [► 13]	Laufzeitmessung von SPS-Code über den CPU-Counter
FB_CXSetTextDisplay [► 16]	Ansteuerung des zweizeiligen Displays des CX1100
FB_CXSetTextDisplayUSB [► 18]	Schreiben und Löschen von Zeilen auf dem zweizeiligen Display des CX 2100 oder der Klemme EL6090.
FB_CXGetTextDisplayUSB [► 20]	Lesen von Zeilen auf dem zweizeiligen Display des CX 2100 oder der Klemme EL6090.
FB_CXSimpleUps [► 22]	Ansteuerung der USV CX1190-UPS (Gerätename CX1100-0900, CX1100-0910, CX1100-0920)
FB_CX5010SetWatchdog [► 24]	Aktiviert einen Hardware-Watchdog auf dem CX5010.
FB_CX5020SetWatchdog [► 25]	Aktiviert einen Hardware-Watchdog auf dem CX5020.

Funktionen

Name	Beschreibung
F_CXSubTimeStamp [► 37]	Berechnet 64bit Subtraktion (Zeit A [100ns] - Zeit B [100ns]) als Ergebnis in μ s, nur bei Differenzen zwischen 0 und 4294967295 μ s, siehe Link.
F_CXNaviSwitch [► 39]	Mit dieser Funktion wird der Wert des CX1100-Navigationsschalters in einen Enum gewandelt.
F_CXNaviSwitchUSB [► 40]	Mit dieser Funktion wird der Wert des CX2100-Navigationsschalter in einen Enum- Wert gewandelt.

3 Funktionsbausteine

3.1 [veraltet]

3.1.1 FB_CxGetDeviceIdentification



Mit dem Funktionsbaustein FB_CxGetDeviceIdentification können Geräteidentifikationsdaten des CX ausgelesen werden.



Veraltete Funktionalität

- Verwenden Sie den FB_GetDeviceIdentificationEx aus der Tc2_Uilities Bibliothek.

Eingänge

```
VAR_INPUT
    bExecute : BOOL;
    tTimeout : TIME;
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Mit der steigenden Flanke wird das Kommando ausgeführt.
tTimeout	TIME	Gibt die Zeit bis zum Abbrechen der Funktion an.

Ausgänge

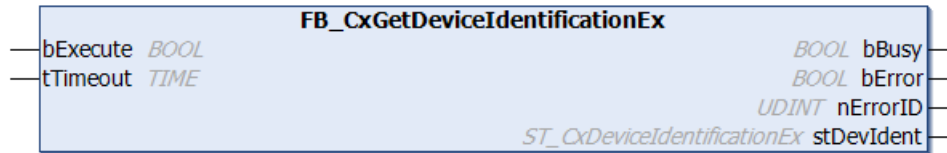
```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrorID : UDINT;
    stDevIdent : ST_CxDeviceIdentification;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Die Daten werden aus dem CX ausgelesen. Nach fehlerfreier Ausführung stehen bei bBusy = FALSE die Daten in der Struktur stDevIdent.
bError	BOOL	Wird TRUE, sobald ein Fehler eintritt.
nErrorID	UDINT	Liefert bei einem gesetzten bError-Ausgang die Fehlernummer.
stDevIdent	ST_CxDeviceIdentification	Enthält die gelesenen Gerätedaten. (Typ: ST_CxDeviceIdentification [► 45])

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7 :TC TR x86, WEC7: TC CE7 ARMV7)	Tc2_SystemCX

3.1.2 FB_CxGetDeviceIdentificationEx



Mit dem Funktionsbaustein FB_CxGetDeviceIdentificationEx können Geräteidentifikationsdaten des CX ausgelesen werden. Der Funktionsbaustein ist die Erweiterung des Funktionsbausteins FB_CxGetDeviceIdentification. Die gelesenen Gerätedaten werden in der Variablen stDevIdent vom Typ ST_CxDeviceIdentificationEx [► 45] gespeichert.

Veraltete Funktionalität

- Verwenden Sie den FB_GetDeviceIdentificationEx aus der Tc2_Uilities Bibliothek.

Eingänge

```

VAR_INPUT
  bExecute : BOOL;
  tTimeout : TIME;
END_VAR
  
```

Name	Typ	Beschreibung
bExecute	BOOL	Mit steigender Flanke wird das Kommando ausgeführt.
tTimeout	TIME	Gibt die Zeit bis zum Abbrechen der Funktion an.

Ausgänge

```

VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrorId : UDINT;
  stDevIdent : ST_CxDeviceIdentificationEx;
END_VAR
  
```

Name	Typ	Beschreibung
bBusy	BOOL	Die Daten werden aus dem CX ausgelesen. Nach fehlerfreier Ausführung stehen bei bBusy = FALSE die Daten in der Struktur stDevIdent.
bError	BOOL	Wird TRUE, sobald ein Fehler eintritt.
nErrorID	UDINT	Liefert bei einem gesetzten bError- Ausgang die Fehlernummer.
stDevIdent	ST_CxDeviceIdentificationEx	Enthält die gelesenen Gerätedaten (Typ: ST_CxDeviceIdentificationEx [► 45])

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7 :TC TR x86, WEC7: TC CE7 ARMV7)	Tc2_SystemCX

3.1.3 FB_CX1010SetWatchdog



Der Funktionsbaustein FB_CX1010SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX1010. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann minimal 2 Sekunden und maximal 255 Sekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktionsbausteininstanz zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX1010 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Umlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX1010 nach Ablauf der tTimeOut-Zeit kommt!

Eingänge

```
VAR_INPUT
    tTimeOut : TIME;
    bEnable  : BOOL;
END_VAR
```

Name	Typ	Beschreibung
tTimeOut	TIME	Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.
bEnable	BOOL	Aktivieren bzw. Deaktivieren des Watchdogs.

Ausgänge

```
VAR_OUTPUT
    bEnabled : BOOL;
    bError   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bEnabled	BOOL	TRUE = Watchdog ist aktiv, FALSE = Watchdog ist nicht aktiv.
bError	BOOL	Fehler beim Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.1.4 FB_CX1020SetWatchdog



Der Funktionsbaustein FB_CX1020SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX1020. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann minimal 2 Sekunden und maximal 255 Sekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktionsbausteininstanz zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX1020 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS
Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX1020 nach Ablauf der tTimeOut-Zeit kommt!

Eingänge

```

VAR_INPUT
    tTimeOut : TIME;
    bEnable   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
tTimeOut	TIME	Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.
bEnable	BOOL	Aktivieren bzw. Deaktivieren des Watchdogs.

Ausgänge

```

VAR_OUTPUT
    bEnabled : BOOL;
    bError   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bEnabled	BOOL	TRUE = Watchdog ist aktiv, FALSE = Watchdog ist nicht aktiv.
bError	BOOL	Fehler beim Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.1.5 FB_CX1030SetWatchdog



Der Funktionsbaustein FB_CX1030SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX1030. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann minimal 2 Sekunden sein und maximal 255 Sekunden.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktionsbausteininstanz zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX1030 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX1030 nach Ablauf der tTimeOut-Zeit kommt!

Eingänge

```
VAR_INPUT
    tTimeOut : TIME;
    bEnable   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
tTimeOut	TIME	Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.
bEnable	BOOL	Aktivieren bzw. Deaktivieren des Watchdogs.

Ausgänge

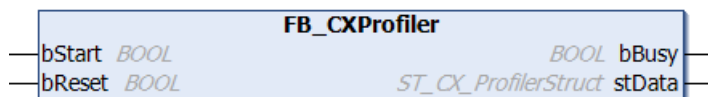
```
VAR_OUTPUT
    bEnabled : BOOL;
    bError   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bEnabled	BOOL	TRUE = Watchdog ist aktiv, FALSE = Watchdog ist nicht aktiv.
bError	BOOL	Fehler beim Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.2 FB_CXProfiler



Mit dem Funktionsbaustein FB_CXProfiler können unter Microsoft Windows CE Messungen der Ausführungszeit vom SPS-Code durchgeführt werden.



Für andere Betriebssysteme beachten Sie den Funktionsbaustein Profiler in der [Tc2 Utilities](#) Bibliothek.

Intern wird eine Instanz des GETCPUOUNTER-Funktionsbausteines aufgerufen. Über eine steigende Flanke an dem bStart-Eingang wird der Messvorgang gestartet und bei einer fallenden Flanke gestoppt. Die Messungen werden intern ausgewertet und stehen am stData-Ausgang in einer Struktur vom Typ [ST_CX_ProfilerStruct](#) [▶ 46] für weitere Verarbeitung zur Verfügung. Neben der aktuellen, minimalen und maximalen Ausführungszeit wird von dem Funktionsbaustein die mittlere Ausführungszeit der letzten 100 Messungen berechnet. Die gemessenen Zeiten werden in Mikrosekunden ausgegeben. Die Ausgangsvariable [stData.dwMeasureCycle](#) [▶ 46] gibt Auskunft über die gesamte Anzahl der bereits durchgeführten Messungen. Um die Ausführungszeit von einem bestimmten Programmabschnitt der SPS zu messen, muss der Messvorgang über eine steigende START-Flanke am Anfang des zu messenden Programmabschnitts gestartet und am Ende über eine fallende START-Flanke gestoppt werden. Alle Werte am DATA-Ausgang können zurückgesetzt werden, wenn gleichzeitig mit der steigenden Flanke am START auch eine steigende Flanke am RESET-Eingang erzeugt wird. Die bereits ermittelten Messwerte der DATA-Struktur werden dann ungültig und aus den folgenden Aufrufen des Funktionsbausteins neu berechnet.

Bemerkung:

Die ermittelten Zeiten können von den tatsächlichen Werten abweichen, da bereits für die Aufrufe des GETCPUOUNTER-Funktionsbausteins einige Zeit benötigt wird. Diese Zeit ist rechnerabhängig und in den ermittelten Zeiten enthalten. Taskunterbrechungen z.B. durch die NC werden nicht erkannt und führen zu größeren Messzeiten.

 **Eingänge**

```
VAR_INPUT
    bStart : BOOL;
    bReset : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bStart	BOOL	Über eine positive Flanke an diesem Eingang wird die Messung der Ausführungszeit gestartet. Über eine negative Flanke an diesem Eingang wird die Messung gestoppt und die aktuelle, minimale, maximale und mittlere Ausführungszeit neu berechnet. Die Variable <code>stData.dwMeasureCycle</code> [▶ 46] wird dabei inkrementiert.
bReset	BOOL	Wenn gleichzeitig mit einer steigenden Flanke am START-Eingang auch eine steigende Flanke an diesem Eingang erzeugt wurde, dann werden alle Variablen am DATA-Ausgang zurückgesetzt. Die alten Werte für die aktuelle, minimale, maximale und mittlere Ausführungszeit werden dabei zurückgesetzt und für die nächsten Messungen neu berechnet.

 **Ausgänge**

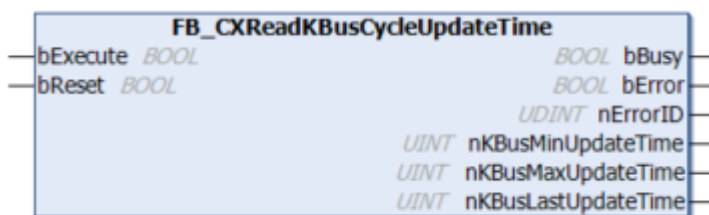
```
VAR_OUTPUT
    bBusy : BOOL;
    stData : ST_CX_ProfilerStruct;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Beim Starten des Messvorgangs wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Zeitmessung abgeschlossen wurde. Nachdem der bBusy-Ausgang zurückgesetzt wurde, stehen die aktuellen Zeiten am stData-Ausgang zur Verfügung.
stData	ST_CX_ProfilerStruct	Struktur vom Typ <code>ST_CX_ProfilerStruct</code> [▶ 46] mit den gemessenen Zeiten in [µs].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1	CX (WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_SystemCX

3.3 FB_CXReadKBusCycleUpdateTime



Der Baustein kann für alle CX verwendet werden, die direkt mit K-Bus Klemmen betrieben werden. Der Baustein ermittelt die Update-Zeit des K-Busses (K-Bus Laufzeit) mit Min.- und Max.-Werten. Die Min- und Max.-Werte können zurückgesetzt werden.

Der K-Bus läuft tasksynchron zum SPS-Programm. Wenn das SPS-Programm fertig ist, wird der K-Bus gestartet, die Ausgänge werden geschrieben und die Eingänge gelesen. Das bedeutet, ein kompletter Zyklus besteht immer aus der SPS-Laufzeit plus der K-Bus Update-Zeit.

- Ist die Summe beider Werte kleiner als die eingestellte Task-Zykluszeit, läuft Ihr System synchron und damit optimal.
- Ist die Summe größer als ihre Task-Zykluszeit, läuft ihr System nicht mehr in Echtzeit. Wir empfehlen, diesen Zustand zu vermeiden.
Sie beheben ihn, indem Sie die Task-Zeit anpassen, Ihr SPS-Programm überarbeiten oder den K-Bus verkleinern.

Eingänge

```
FUNCTION_BLOCK FB_CXReadKBusCycleUpdateTime
VAR_INPUT
    bExecute      : BOOL;           // rising edge triggers read process
    bReset        : BOOL;           // set TRUE to reset the min. and max. values
END_VAR
```

Name	Type	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
bReset	BOOL	Reset der Min.- und Max.-Werte

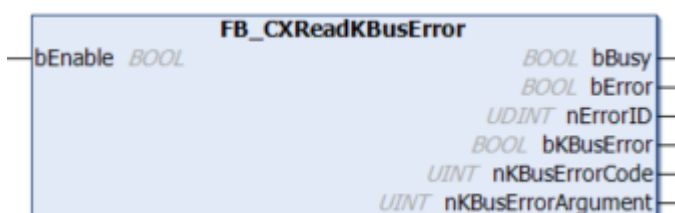
Ausgänge

```
VAR_OUTPUT
    bBusy         : BOOL;           // FB is in process
    bError        : BOOL;           // FB has an error
    nErrorID      : UDINT;          // ADS Fehler Code
    nKBusMinUpdateTime : UINT;       // min. cycle update time in [µs]
    nKBusMaxUpdateTime : UINT;       // max. cycle update time in [µs]
    nKBusLastUpdateTime : UINT;      // last cycle update time in [µs]
END_VAR
```

Name	Type	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code
nKBusMinUpdateTime	UINT	Minimale Update-Zeit in [µs] des K-Bus
nKBusMaxUpdateTime	UINT	Maximale Update-Zeit in [µs] des K-Bus
nKBusLastUpdateTime	UINT	Letzte Update-Zeit in [µs] des K-Bus

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.22	CX (x86, x64, ARM)	Tc2_SystemCX (System) >= 3.4.7.0

3.4 FB_CXReadKBusError



Der Baustein ermöglicht das Auslesen weiterer Informationen zu einem K-Bus Fehler, um ein genaueres Fehlerbild zu erhalten und eine bessere Diagnose zu ermöglichen.

Die Voraussetzung dafür ist, dass Sie direkt an Ihrem CX die K-Bus Klemmen verwenden.

Beispiel: Wenn Sie die K-Bus Verlängerung verwenden und das Kabel der K-Bus Verlängerung gezogen wird, dann wird Ihnen eine K-Bus Unterbrechung als `nErrorCode = 4` und die Position, an der der K-Bus unterbrochen worden ist, als `nErrorArgument = Position` angezeigt.

Weitere Informationen zu möglichen K-Bus Fehlern am Beispiel des CX7000:

<https://infosys.beckhoff.com/content/1031/cx7000/9948355595.html?id=6787792405096234356>

 **Eingänge**

Name	Type	Beschreibung
bEnable	BOOL	Verknüpfen sie diese Variable mit Bit 0 vom K-Bus State, um den Leseprozess automatisch zu aktivieren im Falle eines K-Bus Fehlers. Der K-Bus State ist eine Word Variable. Um das Bit 0 Ihrer verknüpften Variablen auszumaskieren, können Sie <code><VariableName>.0</code> verwenden.

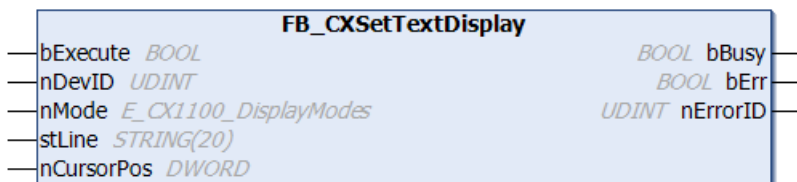
 **Ausgänge**

Name	Type	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code
bKBusError	BOOL	Der K-Bus hat einen Fehler. Nähere Informationen werden an den Ausgängen <code>nKBusErrorCode</code> und <code>nKBusErrorArgument</code> ausgegeben.
nKBusErrorCode	UINT	K-Bus Error Code
nKBusErrorArgument	UINT	K-Bus Error Argument

Ist der Fehler behoben, kann über den Funktionsbaustein `IOF_DeviceReset` (aus der SPS Bibliothek `Tc2_IoFunctions`) der K-Bus neu gestartet werden.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.22	CX (x86, x64, ARM)	Tc2_SystemCX (System) >= 3.4.7.0

3.5 FB_CXSetTextDisplay



Mit dem Funktionsbaustein `FB_CXSetTextDisplay` können Meldungen auf dem zweizeiligen Display des CX1100 ausgegeben werden.

 **Eingänge**

```
VAR_INPUT
    bExecute      : BOOL;
    nDevID       : UDINT;
    nMode        : E_CX1100_DisplayModes;
```



```

    stLine      : STRING(20);
    nCursorPos  : DWORD;
END_VAR

```

Name	Typ	Beschreibung
bExecute	BOOL	Mit steigender Flanke wird das Kommando ausgeführt.
nDevID	UDINT	Device ID des CX1100-Gerätes.
nMode	E_CX1100_DisplayModes	Modeumschaltung
stLine	STRING	String mit 20 Zeichen. Dieser String wird mit dem entsprechenden Kommando im Display angezeigt.
nCursorPos	DWORD	Cursorposition. Ab dieser Stelle im Display wird der String geschrieben.

 **Ausgänge**

```

VAR_OUTPUT
    bBusy      : BOOL;
    bErr       : BOOL;
    nErrorID   : UDINT;
END_VAR

```

Name	Typ	Beschreibung
bBusy	BOOL	Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.
bErr	BOOL	Wird TRUE, sobald ein Fehler eintritt.
nErrorID	UDINT	Liefert bei einem gesetzten bError-Ausgang die Fehlernummer.

E_CX1000_DisplayModes :

```

E_CX1000_DisplayModes : (
    e_CX1100_DisplayNoAction := 0,
    e_CX1100_DisplayOn := 1,
    e_CX1100_DisplayOff,
    e_CX1100_CursorOn,
    e_CX1100_CursorOff,
    e_CX1100_CursorBlinkOn,
    e_CX1100_CursorBlinkOff,
    e_CX1100_BackLightOn,
    e_CX1100_BackLightOff,
    e_CX1100_ClearDisplay,
    e_CX1100_WriteLine1,
    e_CX1100_WriteLine2
);

```

- e_CX1100_DisplayNoAction:** Keine Aktion.
- e_CX1100_DisplayOn:** Einschalten des Displays.
- e_CX1100_DisplayOff:** Ausschalten des Displays.
- e_CX1100_CursorOn:** Einschalten des Cursors.
- e_CX1100_CursorOff:** Ausschalten des Cursors.
- e_CX1100_CursorBlinkOn:** Einschalten des Cursors blinkend.
- e_CX1100_CursorBlinkOff:** Ausschalten des Cursors blinkend.
- e_CX1100_BackLightOn:** Einschalten der Hintergrundbeleuchtung.
- e_CX1100_BackLightOff:** Ausschalten der Hintergrundbeleuchtung.
- e_CX1100_ClearDisplay:** Löschen des Bildschirminhaltes.
- e_CX1100_WriteLine1:** Schreiben der ersten Zeile.
- e_CX1100_WriteLine2:** Schreiben der zweiten Zeile.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.6 FB_CXSetTextDisplayUSB

Der Funktionsbaustein wird sowohl für den CX2100 als auch für die Klemme EL6090 eingesetzt. Beim Aufruf des Funktionsbausteins muss nur die passende NetID und Portnummer für das entsprechende Gerät vergeben werden.

	nPort:	sNetID
CX2100	Wird in TwinCAT auf der ESB Device Registerkarte angezeigt.	Ist die NetID vom PC oder wird leer (' ') gelassen.
EL6090	Ist die EtherCAT-Adresse der Klemme	Ist die EtherCAT-Adresse vom EtherCAT-Master.



Mit dem Funktionsbaustein FB_CXSetTextDisplayUSB werden Meldungen auf dem zweizeiligen Display geschrieben und gelöscht. Der Cursor wird auf dem Display gesteuert, indem er ein- und ausgeschaltet wird oder zum Blinken gebracht wird. Außerdem wird die Hintergrundbeleuchtung mit dem Funktionsbaustein eingeschaltet oder ausgeschaltet.

 **Eingänge**

```

VAR_INPUT
    bExecute      : BOOL;
    sNetID        : T_AmsNetID;
    nPort         : T_AmsPort;
    eMode         : E_CX2100_DisplayModesWr;
    sLine1        : STRING(80);
    sLine2        : STRING(80);
    nCursorPosX   : USINT;
    nCursorPosY   : USINT;
END_VAR
    
```

Name	Typ	Beschreibung
bExecute	BOOL	Mit steigender Flanke wird das Kommando ausgeführt.
sNetID	T_AmsNetID	AMSNetID des Gerätes
nPort	T_AmsPort	AMSPort des Gerätes
eMode	E_CX2100_DisplayModesRd	Mode-Umschaltung
sLine1	STRING	String mit 80 Zeichen. Dieser String wird mit dem entsprechenden Kommando im Display in der ersten Zeile angezeigt. Bei Strings mit mehr als 16 Zeichen, wird der Text als Lauftext angezeigt.
sLine2	STRING	String mit 80 Zeichen. Dieser String wird mit dem entsprechenden Kommando im Display in der zweiten Zeile angezeigt. Bei Strings mit mehr als 16 Zeichen, wird der Text als Lauftext angezeigt.
nCursorPosX	USINT	Cursorposition auf der X-Achse. Ab dieser Stelle im Display wird der String geschrieben.
nCursorPosY	USINT	Cursorposition auf der Y-Achse. Ab dieser Stelle im Display wird der String geschrieben.

 **Ausgänge**

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorID   : UDINT;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.
bError	BOOL	Wird TRUE, sobald ein Fehler eintritt.
nErrorID	UDINT	Liefert bei einem gesetzten bError-Ausgang die Fehlernummer.

E_CX2100_DisplayModesWr:

```
E_CX2100_DisplayModesWr : (
    eCX2100_DisplayNoActionWr := 0,
    eCX2100_CursorOn,
    eCX2100_CursorOff,
    eCX2100_CursorBlinkOn,
    eCX2100_CursorBlinkOff,
    eCX2100_BackLightOn,
    eCX2100_BackLightOff,
    eCX2100_ClearDisplay,
    eCX2100_WriteLine1,
    eCX2100_WriteLine2,
    eCX2100_WriteLines,
    eCX2100_CursorPosX,
    eCX2100_CursorPosY,
    eCX2100_CursorPosXY
);
```

eCX2100_DisplayNoActionWr: Keine Aktion.

eCX2100_CursorOn: Einschalten des Cursors.

eCX2100_CursorOff: Ausschalten des Cursors.

eCX2100_CursorBlinkOn: Einschalten des Cursors blinkend.

eCX2100_CursorBlinkOff: Ausschalten des Cursors blinkend.

eCX2100_BackLightOn: Einschalten der Hintergrundbeleuchtung.

eCX2100_BackLightOff: Ausschalten der Hintergrundbeleuchtung.

eCX2100_ClearDisplay: Löschen des Bildschirminhalts.

eCX2100_WriteLine1: Schreiben der ersten Zeile.

eCX2100_WriteLine2: Schreiben der zweiten Zeile.

eCX2100_WriteLines: Schreiben der Zeilen.

eCX2100_CursorPosX: Cursor Position auf der X-Achse.

eCX2100_CursorPosY: Cursor Position auf der Y-Achse.

eCX2100_CursorPosXY: Cursor Position auf der XY-Achse.

Voraussetzungen bei Verwendung der EL6090

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/ BSD: TC RT x64, TC OS ARMT2)	Tc2_SystemCX

Voraussetzungen bei Verwendung des CX2100

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, TC/ BSD: TC RT x64)	Tc2_SystemCX

3.7 FB_CXGetTextDisplayUSB

Der Funktionsbaustein wird sowohl für den CX2100 als auch für die Klemme EL6090 eingesetzt. Beim Aufruf des Funktionsbausteins muss nur die passende NetID und Portnummer für das entsprechende Gerät vergeben werden.

	nPort:	sNetID
CX2100	Wird in TwinCAT auf der ESB Device Registerkarte angezeigt.	Ist die NetID vom PC oder wird leer (' ') gelassen.
EL6090	Ist die EtherCAT-Adresse der Klemme	Ist die EtherCAT-Adresse vom EtherCAT-Master.



Mit dem Funktionsbaustein FB_CXGetTextDisplayUSB werden die Zeilen auf dem Display gelesen. Außerdem wird der Cursorstatus gelesen, also ob der Cursor eingeschaltet oder ausgeschaltet ist und ob der Cursor blinkt. Der Funktionsbaustein liefert auch Informationen darüber, ob die Hintergrundbeleuchtung ein- oder ausgeschaltet ist.

 **Eingänge**

```
VAR_INPUT
  bExecute      : BOOL;
  sNetID        : T_AmsNetID;
  nPort         : T_AmsPort;
  eMode         : E_CX2100_DisplayModesRd;
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Mit steigender Flanke wird das Kommando ausgeführt.
sNetID	T_AmsNetID	AMSNetID des Gerätes
nPort	T_AmsPort	AMSPort des Gerätes
eMode	E_CX2100_DisplayModesRd	Mode-Umschaltung

 **Ausgänge**

```
VAR_OUTPUT
  bBusy         : BOOL;
  bError        : BOOL;
  nErrorID      : UDINT;
  sLine1        : STRING(80);
  sLine2        : STRING(80);
  nCursorPosX   : USINT;
  nCursorPosY   : USINT;
  nCursorMode   : USINT;
  nBacklight    : USINT;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.
bError	BOOL	Wird TRUE, sobald ein Fehler eintritt.
nErrorID	UDINT	Liefert bei einem gesetzten bError-Ausgang die Fehlernummer.
sLine1	STRING	Dieser String wird mit dem entsprechenden Kommando gelesen.
sLine2	STRING	Dieser String wird mit dem entsprechenden Kommando gelesen.
nCursorPosX	USINT	Position des Cursors auf der X-Achse
nCursorPosY	USINT	Position des Cursors auf der Y-Achse
nCursorMode	USINT	Modus des Cursors
nBacklight	USINT	Hintergrundbeleuchtung

E_CX2100_DisplayModesRd:

```
E_CX2100_DisplayModesRd : (
  eCX2100_DisplayNoActionRd := 0,
  eCX2100_ReadCursorInfo,
  eCX2100_ReadBackLight,
  eCX2100_ReadLine1,
  eCX2100_ReadLine2,
  eCX2100_ReadLines
);
```

eCX2100_DisplayNoActionRd: Keine Aktion.

eCX2100_ReadCursorInfo: Werte über den Cursor lesen.

eCX2100_ReadBackLight: Werte der Hintergrundbeleuchtung lesen.

eCX2100_ReadLine1: Werte aus der ersten Zeile lesen.

eCX2100_ReadLine2: Werte aus der zweiten Zeile lesen.

eCX2100_ReadLines: Werte aus Zeilen lesen.

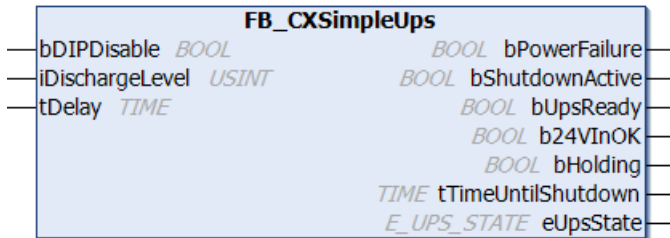
Voraussetzungen bei Verwendung des CX2100

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, TC/BSD: TC RT x64)	Tc2_SystemCX

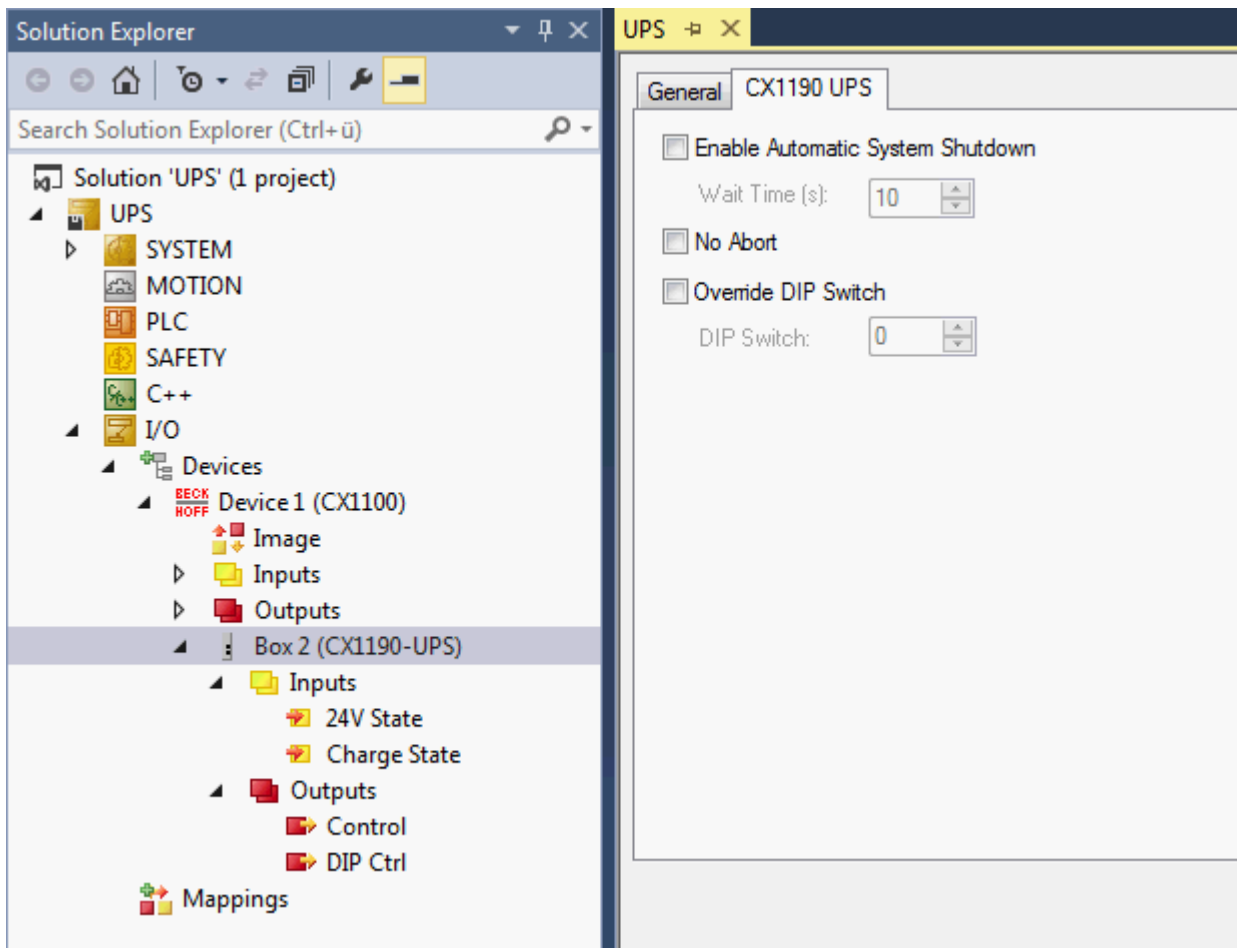
Voraussetzungen bei Verwendung der EL6090

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_SystemCX

3.8 FB_CXSimpleUps



Der Funktionsbaustein FB_CXSimpleUps kann auf dem CX1000 oder CX1020 verwendet werden, um die USV CX1190-UPS aus der SPS anzusteuern. In diesem Fall müssen im TwinCAT System Manager die USV-Einstellungen deaktiviert sein.



Eingänge

```
VAR_INPUT
    bDIPDisable      : BOOL;
    iDischargeLevel  : USINT;
    tDelay           : TIME;
END_VAR
```

Name	Typ	Beschreibung
bDIPDisable	BOOL	Wenn bDIPDisable = TRUE, dann wird die Stellung des Ladezustandsschalters an der USV ignoriert und stattdessen iDischargeLevel verwendet.
iDischargeLevel	USINT	Ausschaltsschwelle für Entladung: 0 = 100% (Maximale Entladung), 9 = 90%, 8 = 80%, ..., 2 = 20%, 1 = 10% (Minimale Entladung).
tDelay	TIME	Haltezeit, bevor der Shutdown durchgeführt wird. Sie wird benutzt, um kurze Spannungsausfälle (bis zu 10s) zu überbrücken. Nach Überschreiten der Haltezeit wird die Halteperiode abgebrochen. Intern wartet der FB für 2.5s. Ist bis dahin die Spannung zurückgekehrt, dann kehrt der FB zum normalen Operation zurück, anderenfalls wird das System heruntergefahren. Kehrt die Spannung während oder nach dem Herunterfahren zurück, rebootet der CX automatisch nach dem Entladen und Wiederaufladen der USV.

Ausgänge

```
VAR_OUTPUT
    bPowerFailure      : BOOL;
    bShutdownActive    : BOOL;
    bUpsReady          : BOOL;
    b24VInOK          : BOOL;
    bHolding           : BOOL;
```

```
tTimeUntilShutdown : TIME;
eUpsState           : E_UPS_STATE;
END_VAR
```

Name	Typ	Beschreibung
bPowerFailure	BOOL	Wird TRUE, wenn ein Spannungsausfall der Versorgungsspannung erkannt wird, wird FALSE, wenn die Eingangsspannung wiederkehrt.
bShutdownActive	BOOL	Wird TRUE, sobald der Stop bzw. Shutdown ausgeführt wird.
bUpsReady	BOOL	Wird TRUE, sobald die 24V-Eingangsspannung bereitsteht.
b24VinOK	BOOL	Wird TRUE, sobald die USV die Ausgangsspannung bereitstellt.
bHolding	BOOL	Wird TRUE, wenn ein Spannungsausfall der Versorgungsspannung erkannt wird und die Haltezeit noch nicht abgelaufen ist.
tTimeUntilShutdown	TIME	Zeigt die verbleibende Haltezeit bis zum Shutdown an.
eUpsState	E_UPS_STATE	Zeigt den Status der USV an [UNDEF CHARGING CHARGED DISCHARGE DISCHARGE_RESTART OUTPUT_OFF OVERLOAD].

Konfigurationsvariablen

```
VAR_CONFIG
  Ii24VState   AT %I* : BYTE;
  IiChargeState AT %I* : USINT;
  QiControl    AT %Q* : BYTE;
  QiDipControl AT %Q* : USINT;
END_VAR
```

Name	Typ	Beschreibung
Ii24VState	BYTE	Muss mit Eingang '24V State' gelinkt sein, siehe Bild oben.
IiChargeState	USINT	Muss mit Eingang 'Charge State' gelinkt sein, siehe Bild oben.
QiControl	BYTE	Muss mit Ausgang 'Contol' gelinkt sein, siehe Bild oben.
QiDipControl	USINT	Muss mit Ausgang 'DIP Ctrl' gelinkt sein, siehe Bild oben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX



Für andere CX und PC beachten Sie bitte den Funktionsbaustein FB_S_SUPS_BAPI in der Tc2_SUPS Bibliothek.

3.9 FB_CX5010SetWatchdog



Der Funktionsbaustein FB_CX5010SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX5010. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann minimal 2 Sekunden sein und maximal 255 Sekunden.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktionsbausteininstanz zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX5010 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stopp, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX5010 nach Ablauf der tTimeOut-Zeit kommt!

Eingänge

```
VAR_INPUT
    tTimeOut : TIME;
    bEnable  : BOOL;
END_VAR
```

Name	Typ	Beschreibung
tTimeOut	TIME	Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.
bEnable	BOOL	Aktivieren bzw. Deaktivieren des Watchdogs.

Ausgänge

```
VAR_OUTPUT
    bEnabled : BOOL;
    bError   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bEnabled	BOOL	TRUE = Watchdog ist aktiv, FALSE = Watchdog ist nicht aktiv.
bError	BOOL	Fehler beim Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.10 FB_CX5020SetWatchdog



Der Funktionsbaustein FB_CX5020SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX5020. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann minimal 2 Sekunden und maximal 255 Sekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktionsbausteininstanz zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX5020 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stopp, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX5020 nach Ablauf der tTimeOut-Zeit kommt!

Eingänge

```
VAR_INPUT
  tTimeout : TIME;
  bEnable  : BOOL;
END_VAR
```

Name	Typ	Beschreibung
tTimeout	TIME	Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.
bEnable	BOOL	Aktivieren bzw. Deaktivieren des Watchdogs.

Ausgänge

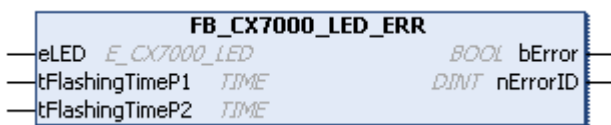
```
VAR_OUTPUT
  bEnabled : BOOL;
  bError   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
bEnabled	BOOL	TRUE = Watchdog ist aktiv, FALSE = Watchdog ist nicht aktiv.
bError	BOOL	Fehler beim Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

3.11 FB_CX7000_LED_ERR



Der Baustein erlaubt das Verwenden der ERR LED auf dem CX7000. Der Baustein ist bei seinem Aufruf sofort aktiv und steuert die ERR LED über den Mode an.

Um Zustände des SPS-Programms, der Kommunikation oder andere Hinweise nach außen hin sichtbar zu machen, kann die ERR LED des CX7000 verwendet werden.

Die ERR LED ist zweifarbig ausgeführt, rot und grün. Sind beide Farben eingeschaltet, leuchtet die LED gelb. Sie können die LED entweder einschalten oder blinken lassen.

Anwenderspezifische Funktion der LEDs

Durch die anwenderspezifische Nutzbarkeit der LEDs kann der Beckhoff Support die Bedeutung eines Blink-Codes nicht kennen und den Kunden nicht unterstützen.

- Dokumentieren Sie die Funktion der LEDs für Ihre Kunden.

Eingänge

```
VAR_INPUT
  bEnable : BOOL; // set TRUE to enable LED handling; Reset in order to re
set error
  eLED : E_CX7000_LED; // LED flashing mode
  tFlashingTimeP1 : TIME:=T#250MS; // Flashing Time >=200ms first pulse
  tFlashingTimeP2 : TIME:=T#250MS; // Flashing Time >=200ms second pulse
END_VAR
```

Name	TYP	Beschreibung
bEnable	BOOL	Der Baustein steuert die LED an sobald und solange der Eingang TRUE ist.
eLED	E_CX7000_LED	LED Mode
tFlashingTimeP1	TIME	Zeit für den ersten Puls (>= 200 ms)
tFlashingTimeP2	TIME	Zeit für den zweiten Puls (>= 200 ms)

Ausgänge

```
VAR_OUTPUT
  bError      : BOOL;          // error flag
  nErrorID    : UDINT;        (* ADS Error ID. If nErrorID=DEVICE_SRVNOTSUPP probably the image
  version need to be updated to support this feature. *)
END_VAR
```

Name	Typ	Beschreibung
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiel: DEVICE_SRVNOTSUPP: Die Image-Version des CX7000 unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.

Beispiel:

```
VAR
  BK9000_BoxState AT %I* : WORD;
  fbErrorLED : FB_CX7000_LED_ERR;
END_VAR

IF BK9000_BoxState=0 THEN
  fbErrorLED.eLED :=E_CX7000_LED.LED_flashing_GREEN_OFF;
ELSE
  fbErrorLED.eLED :=E_CX7000_LED.LED_flashing_RED_OFF;
END_IF

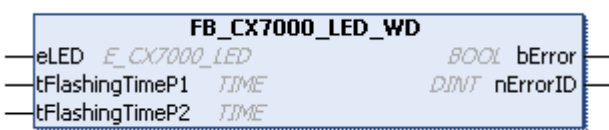
fbErrorLED(
  bEnable := TRUE,
  tFlashingTimeP1 := ,
  tFlashingTimeP2 := ,
  bError => ,
  nErrorID => );
```

HINWEIS

Baustein nur für CX7000 verwendbar
Der Baustein kann und darf nur für den CX7000 verwendet werden.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX7000	Tc2_SystemCX (System) >= 3.4.8.0

3.12 FB_CX7000_LED_WD



Der Baustein erlaubt das Verwenden der WD LED auf dem CX7000. Der Baustein ist sofort mit dem Aufruf aktiv und steuert die WD LED über den Mode an.

Sie können die WD LED des CX7000 verwenden, um die Zustände des SPS Programms, der Kommunikation oder andere Hinweise nach außen hin sichtbar zu machen. Die WD LED ist zweifarbig ausgeführt, rot und grün. Sind beide Farben eingeschaltet, leuchtet die LED gelb. Sie können die LED einschalten und/oder blinken lassen.

● Anwenderspezifische Funktion der LEDs

i Durch die anwenderspezifische Nutzbarkeit der LEDs kann der Beckhoff Support die Bedeutung eines Blink-Codes nicht kennen und den Kunden nicht unterstützen.

- Dokumentieren Sie die Funktion der LEDs für Ihre Kunden.

📌 Eingänge

```
VAR_INPUT
  bEnable      : BOOL;           // set TRUE to enable LED handling; Reset in order to re
set error
  eLED         : E_CX7000_LED;   // LED flashing mode
  tFlashingTimeP1 : TIME:=T#250MS; // Flashing Time >=200ms first pulse
  tFlashingTimeP2 : TIME:=T#250MS; // Flashing Time >=200ms second pulse
END_VAR
```

Name	Type	Beschreibung
bEnable	BOOL	Der Baustein steuert die LED an sobald und solange der Eingang TRUE ist.
eLED	E_CX7000_LED	LED Mode
tFlashingTimeP1	TIME	Zeit für den ersten Puls (>= 200 ms)
tFlashingTimeP2	TIME	Zeit für den zweiten Puls (>= 200 ms)

📌 Ausgänge

```
VAR_OUTPUT
  bError      : BOOL;           // error flag
  nErrorID    : UDINT;         (* ADS Error ID. If nErrorID=DEVICE_SRVNOTSUPP probably the image v
ersion need to be updated to support this feature. *)
END_VAR
```

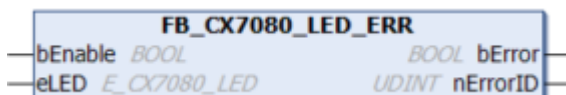
Name	Type	Beschreibung
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiel: DEVICE_SRVNOTSUPP: Die Image-Version des CX7000 unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.

HINWEIS

Baustein nur für CX7000 verwendbar
Der Baustein kann und darf nur für den CX7000 verwendet werden.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX7000	Tc2_SystemCX (System) >= 3.4.8.0

3.13 FB_CX7080_LED_ERR



Sie können die WD/ERR LEDs des CX7080 verwenden, um Zustände des SPS Programms sowie der Kommunikation oder andere Hinweise nach außen hin sichtbar zu machen.

Der Baustein ermöglicht das Setzen der ERR LED des CX7080. Die LED ist zweifarbig ausgeführt, rot und grün. Sind beide Farben eingeschaltet, leuchtet die LED gelb. Ihnen stehen somit drei Farben zur Verfügung. Sie können die LED einschalten und/oder blinken lassen.

Um die LED zu nutzen, muss der entsprechende COM Port in der Konfiguration enthalten sein. Die RS485 Schnittstelle ist mit der ERR LED verknüpft.

Anwenderspezifische Funktion der LEDs

i Durch die anwenderspezifische Nutzbarkeit der LEDs kann der Beckhoff Support die Bedeutung eines Blink-Codes nicht kennen und den Kunden nicht unterstützen.

- Dokumentieren Sie die Funktion der LEDs für Ihre Kunden.

Eingänge

```
VAR_INPUT
  bEnable      : BOOL;           // set TRUE to enable LED handling; Reset in order to reset error
  eLED         : E_CX7080_LED;   // LED flashing mode
END_VAR
```

Name	Typ	Beschreibung
bEnable	BOOL	Der Baustein steuert die LED an, sobald und solange der Eingang TRUE ist.
eLED	E_CX7080_LED	LED Mode

Ausgänge

```
VAR_OUTPUT
  bError       : BOOL;           // error flag
  nErrorID     : UDINT;         (* ADS Error ID.
If nErrorID=DEVICE_NOTFOUND probably the COM port is not set in the TC config.
If nErrorID=DEVICE_SRVNOTSUPP probably the image version need to be updated to support this feature.
*)
END_VAR
```

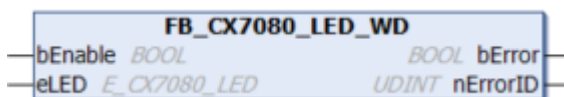
Name	Typ	Beschreibung
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_NOTFOUND: Vermutlich ist der COM Port nicht in der TwinCAT System Konfiguration enthalten. DEVICE_SRVNOTSUPP: Die Image-Version des CX7080 unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.

HINWEIS

Baustein nur für CX7080 verwendbar
Der Baustein kann und darf nur für den CX7080 verwendet werden.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX7080	Tc2_SystemCX (System) >= 3.4.8.0

3.14 FB_CX7080_LED_WD



Sie können die WD/ERR LEDs des CX7080 verwenden, um Zustände des SPS-Programms sowie der Kommunikation oder andere Hinweise nach außen hin sichtbar zu machen.

Der Baustein ermöglicht das Setzen der WD LED des CX7080. Die LED ist zweifarbig ausgeführt, rot und grün. Sind beide Farben eingeschaltet, leuchtet die LED gelb. Ihnen stehen somit drei Farben zur Verfügung. Sie können die LED einschalten und/oder blinken lassen.

Um die LED zu nutzen, muss der entsprechende COM Port in der Konfiguration enthalten sein. Die RS232 Schnittstelle ist mit der WD LED verknüpft.

● Anwenderspezifische Funktion der LEDs

i Durch die anwenderspezifische Nutzbarkeit der LEDs kann der Beckhoff Support die Bedeutung eines Blink-Codes nicht kennen und den Kunden nicht unterstützen.

- Dokumentieren Sie die Funktion der LEDs für Ihre Kunden.

🔌 Eingänge

```
VAR_INPUT
  bEnable      : BOOL;           // set TRUE to enable LED handling; Reset in order to reset error
  eLED         : E_CX7080_LED;   // LED flashing mode
END_VAR
```

Name	Typ	Beschreibung
bEnable	BOOL	Der Baustein steuert die LED an, sobald und solange der Eingang TRUE ist.
eLED	E_CX7080_LED	LED Mode

🔌 Ausgänge

```
VAR_OUTPUT
  bError       : BOOL;           // error flag
  nErrorID     : UDINT;         (* ADS Error ID.
  If nErrorID=DEVICE_NOTFOUND probably the COM port is not set in the TC config.
  If nErrorID=DEVICE_SRVNOTSUPP probably the image version need to be updated to support this feature.
  *)
END_VAR
```

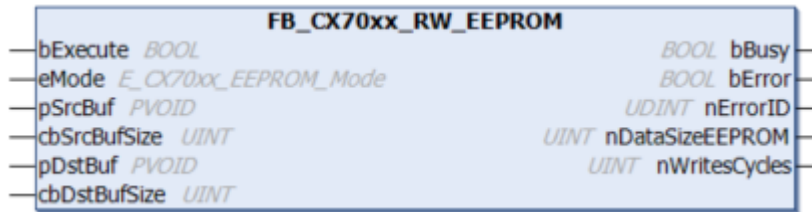
Name	Typ	Beschreibung
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_NOTFOUND: Vermutlich ist der COM Port nicht in der TwinCAT System Konfiguration enthalten. DEVICE_SRVNOTSUPP: Die Image-Version des CX7080 unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.

HINWEIS

Baustein nur für CX7080 verwendbar
Der Baustein kann und darf nur für den CX7080 verwendet werden.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX7080	Tc2_SystemCX (System) >= 3.4.8.0

3.15 FB_CX70xx_RW_EEPROM



Der Baustein erlaubt das Schreiben von maximal 120 Byte in das EEPROM (Hardware) des CX70xx. Das EEPROM darf maximal 200-mal beschrieben werden. Der Speicher ist für das einmalige Schreiben gedacht.

Dieser Funktionsbaustein kann dafür verwendet werden, den CX70xx zu personalisieren. Das heißt, im einfachsten Fall schreiben Sie Ihre Firmenkennung in das EEPROM. Beim Start des CX70xx Programms lesen Sie den Inhalt des Speichers aus. Ist dieser zum Beispiel leer, können Sie das Programm nicht weiter ausführen, da es sich nicht mehr um Ihren originalen CX70xx handelt, den Sie programmiert haben.

Sollten Sie einen CX70xx gegen ein neues Gerät austauschen wollen, muss das EEPROM erneut von Ihnen geschrieben werden.

Eingänge

```
VAR_INPUT
  bExecute      : BOOL;           // rising edge triggers process with selected mode
  eMode         : E_CX70xx_EEPROM_Mode; // select RW mode
  pSrcBuf       : PVOID;         // pointer to WRITE EEPROM data buffer
  cbSrcBufSize  : UINT;          // size of WRITE EEPROM data buffer (max.120 Bytes)
  pDstBuf       : PVOID;         // pointer to READ EEPROM data buffer
  cbDstBufSize  : UINT;          // max.size of READ EEPROM data buffer (max.120 Bytes)
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
eMode	E_CX70xx_EEPROM_Mode	ReadOnly: EEPROM Lesen WriteOnly: EEPROM Schreiben WriteAndRead: EEPROM Schreiben und im Anschluss Lesen
pSrcBuf	PVOID	Pointer auf den Datenpuffer, der geschrieben werden soll.
cbSrcBufLen	UINT	Länge der zu schreibenden Daten (maximal 120 Bytes)
pDstBuf	PVOID	Pointer auf den Datenpuffer, in den der Inhalt des EEPROM kopiert werden soll.
cbDstBufLen	UINT	Länge der zu lesenden Daten. (maximal 120 Bytes) Beim Lesen muss die Längeninformation größer gleich der im EEPROM enthalten Daten sein.

Ausgänge

```
VAR_OUTPUT
  bBusy         : BOOL;           // FB is working
  bError        : BOOL;           // FB has an Error
  nErrorID      : UDINT;          (* Error Code
  If nErrorID=DEVICE_INVALIDACCESS the EEPROM write cycles reached max. value.
  If nErrorID=DEVICE_INVALIDPARM the given pointer parameter is invalid/null.
  If nErrorID=DEVICE_INVALIDSIZE the given buffer size is too small or too big.
  If nErrorID=DEVICE_SRVNOTSUPP probably the image version need to be updated to support this feature. *)
  nDataSizeEEPROM : UINT;          // current size of (read) EEPROM data in bytes (max.120 Bytes)
  nWritesCycles  : UINT;          // already performed EEPROM write cycles (maximum possible = 200)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_INVALIDACCESS: Die EEPROM Schreibzyklen haben den maximalen Wert erreicht. Das EEPROM kann nicht erneut beschrieben werden. DEVICE_INVALIDPARG: Die zugewiesenen Pointer sind ungültig/ NULL. DEVICE_INVALIDSIZE: Die zugewiesene Puffergröße ist zu klein oder zu groß. DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.
nDataSizeEEPROM	UINT	Aktuelle Größe in Bytes der gelesenen EEPROM Daten
nWritesCycles	UINT	Anzahl der noch zur Verfügung stehenden Schreibvorgänge

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

3.16 FB_CX70xx_ResetOnBoardIO



Der Baustein erlaubt es, einen Reset vom OnBoard-IO des Embedded-PC CX70xx auszuführen.

Typischer Anwendungsfall ist nach einem Fehler in der Kommunikation zu den OnBoard-IOs (CX7028). Zu einem solchen Fehler kommt es, wenn die Spannungsversorgung (Up) der OnBoard-IOs unterbrochen wird.

HINWEIS

Zustand der IOs

Ausgänge, die im Prozessabbild noch gesetzt sind, werden nach einem Reset sofort wieder eingeschaltet.

Weitere Details zum OnBoard-IO finden Sie in der [Dokumentation des Embedded-PC CX70xx](#).

Eingänge

```
VAR_INPUT
    bExecute      : BOOL;           // rising edge triggers process
    sNetId        : T_AmsNetID;    // AMS Net ID of the OnBoard IOs
    tTimeout      : TIME := DEFAULT_ADS_TIMEOUT; // maximum time allowed for execution of this ADS command
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
sNetId	T_AmsNetID	AMS Net ID der OnBoard-IOs
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

 **Ausgänge**

```
VAR_OUTPUT
  bBusy      : BOOL;          // FB is working
  bError     : BOOL;          // FB has an Error
  nErrorID   : UDINT;        (* Error Code. If nErrorID=DEVICE_SRVNOTSUPP probably the image versio
n need to be updated to support this feature. *)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=47912) ist notwendig.

Beispiel:

```
FUNCTION_BLOCK FB_Test_ResetOnboardIO
VAR
  AMSNetID   : T_AmsNetIdArr; // link to the AMS Net ID of the OnBoard IOs
  State      : WORD;          // link to the State of the OnBoard IOs
  bReset     : BOOL;          // if Ready to Reset you can reset the OnBoard IOs
  fbReset    : FB_CX70xx_ResetOnBoardIO;
END_VAR

IF State<>8 AND NOT State.8 AND State.4 THEN // if OnBoard IO device signals an error and is not OP
but present
  bReset := TRUE;
ELSE
  bReset := FALSE;
END_IF

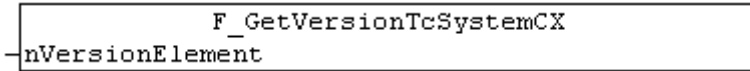
IF NOT fbReset.bBusy AND bReset THEN
  fbReset(bExecute:=TRUE, sNetId:=F_CreateAmsNetId(AMSNetID));
ELSE
  fbReset(bExecute:=FALSE);
END_IF
```

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

4 Funktionen

4.1 [veraltet]

4.1.1 F_GetVersionTcSystemCX



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

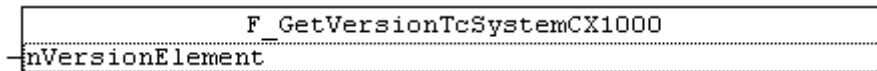
FUNCTION F_GetVersionTcSystemCX : UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.2 F_GetVersionTcSystemCX1000



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

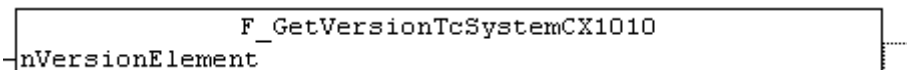
FUNCTION F_GetVersionTcSystemCX1000 : UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.3 F_GetVersionTcSystemCX1010



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcSystemCX1010 : UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.4 F_GetVersionTcSystemCX1020

```
          F_GetVersionTcSystemCX1020
-----nVersionElement
```

Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcSystemCX1020 : UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.5 F_GetVersionTcSystemCX1030

```
          F_GetVersionTcSystemCX1030
-----nVersionElement
```

Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcSystemCX1030 : UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.6 F_GetVersionTcSystemCX5010

```
          F_GetVersionTcSystemCX5010
-----nVersionElement
```

Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcSystemCX5010 : UINT

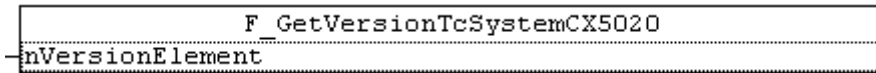
```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;

- 2 : minor number;
- 3 : revision number;

4.1.7 F_GetVersionTcSystemCX5020



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

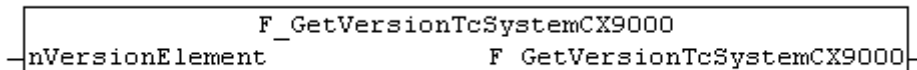
FUNCTION F_GetVersionTcSystemCX5020 : UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.8 F_GetVersionTcSystemCX9000



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

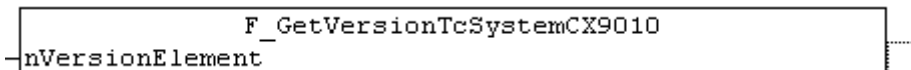
FUNCTION F_GetVersionTcSystemCX9000 : UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.9 F_GetVersionTcSystemCX9010



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcSystemCX9010 : UINT

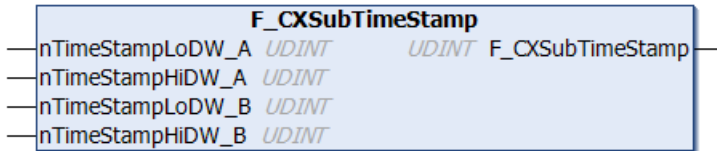
```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

4.1.10 F_CXSubTimeStamp

i Für die Subtraktion bietet TwinCAT3.1 auch 64bit-Datentypen (LINT/ULINT, LWORD) die direkt ausgeführt werden können (A-B) bzw. es bestehen TC2-kompatiblen Funktionen für 64bit-Operationen, die alternativ zur F_CXSubTimeStamp verwendet werden sollten.



Die Funktion F_CXSubTimeStamp führt eine 64bit-Subtraktion Zeitstempel A - Zeitstempel B durch und wandelt das Ergebnis in μs um. Die notwendigen 64bit-Zeitstempel mit einer 100ns-Auflösung können mit dem Funktionsbaustein GETCPUCOUNTER aus dem System ausgelesen werden.

Ist die Differenz zwischen dem Zeitstempel A und dem Zeitstempel B negativ oder größer als 4294967295us, dann wird der Maximalwert 4294967295us zurückgeliefert, das entspricht 71 Minuten, 34 Sekunden, 967 Millisekunden und 295 Mikrosekunden. In diesen Fällen kann die Funktion UInt64Sub64() der TcUtilities.lib verwendet werden, um eine vollständige 64-Bit-Subtraktion mit 64-Bit-Ergebnis in [100ns] durchzuführen.

FUNCTION F_CXSubTimeStamp : UDINT

```
VAR_INPUT
  nTimeStampLoDW_A : UDINT; (* 2*32 bit time stamp A: low DWORD *)
  nTimeStampHiDW_A : UDINT; (* 2*32 bit time stamp A: high DWORD *)
  nTimeStampLoDW_B : UDINT; (* 2*32 bit time stamp B: low DWORD *)
  nTimeStampHiDW_B : UDINT; (* 2*32 bit time stamp B: high DWORD *)
END_VAR
```

nTimeStampLoDW_A: Untere 32bit des Zeitstempels A.

nTimeStampHiDW_A: Obere 32bit des Zeitstempels A.

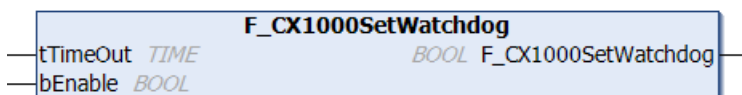
nTimeStampLoDW_B: Untere 32bit des Zeitstempels B.

nTimeStampHiDW_B: Obere 32bit des Zeitstempels B.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7 :TC TR x86, WEC7: TC CE7 ARMV7)	Tc2_SystemCX

4.1.11 F_CX1000SetWatchdog



Die Funktion F_CX1000SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX1000. Der Watchdog wird über bEnable = TRUE und die tTimeout-Zeit aktiviert. Die tTimeoutzeit kann minimal mehrere SPS-Taskzyklen (mehrfaches der Aufrufzeit der Funktion F_CX1000SetWatchdog) sein und maximal 65s und 535ms.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktion zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX1000 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = 0 deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX1000 nach Ablauf der Timeoutzeit kommt!

FUNCTION F_CX1000SetWatchdog: BOOL

```
VAR_INPUT
  tTimeout : TIME;
  bEnable  : BOOL;
END_VAR
```

tTimeOut: Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.

bEnable: Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

4.1.12 F_CX9000SetWatchdog



Die Funktion F_CX9000SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX9000. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann maximal 65 Sekunden und 535 Millisekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktion zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX9000 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX9000 nach Ablauf der tTimeOut-Zeit kommt!

FUNCTION F_CX9000SetWatchdog: BOOL

```
VAR_INPUT
  tTimeout : TIME;
  bEnable  : BOOL;
END_VAR
```

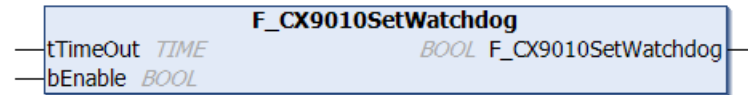
tTimeOut: Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.

bEnable: Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (ARM)	Tc2_SystemCX

4.1.13 F_CX9010SetWatchdog



Die Funktion F_CX9010SetWatchdog aktiviert einen Hardware-Watchdog auf dem CX9000. Der Watchdog wird über bEnable = TRUE und die tTimeOut-Zeit aktiviert. Die tTimeOut-Zeit kann maximal 65 Sekunden und 535 Millisekunden sein.

Wenn der Watchdog einmal aktiviert wurde, muss die Funktion zyklisch in kürzeren Abständen aufgerufen werden als tTimeOut, da bei Ablauf der tTimeOut-Zeit der CX9010 automatisch einen Neustart durchführt. Der Watchdog kann daher dafür eingesetzt werden, um Systeme automatisch neu zu booten, die in eine Endlosschleife gelaufen sind bzw. bei denen die PLC steht.

Der Watchdog kann über bEnable = FALSE bzw. die tTimeOut-Zeit = T#0s deaktiviert werden.

HINWEIS

Der Watchdog muss vor der Verwendung von Breakpoints, einem SPS-Reset bzw. Urlöschen und vor einem TwinCAT Stop, einem Wechsel in den Konfig-Mode oder dem Aktivieren der Konfiguration deaktiviert werden, da es sonst unmittelbar zum Reboot des CX9010 nach Ablauf der tTimeOut-Zeit kommt!

FUNCTION F_CX9010SetWatchdog: BOOL

```
VAR_INPUT
    tTimeOut : TIME;
    bEnable  : BOOL;
END_VAR
```

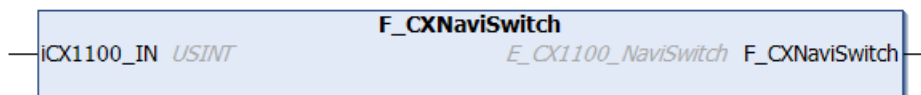
tTimeOut: Watchdogzeit, nach deren Ablauf ein Neustart durchgeführt wird.

bEnable: Aktivieren bzw. Deaktivieren des Watchdogs.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (ARM)	Tc2_SystemCX

4.2 F_CXNaviSwitch



Die Funktion F_CXNaviSwitch wandelt den Wert des CX1100-Navigationsschalters in einen Enum-Wert vom Typ E_CX1100_NaviSwitch um.

FUNCTION F_CXNaviSwitch: E_CX1100_NaviSwitch

```
VAR_INPUT
    iCX1100_IN : USINT
END_VAR
```

E_CX1100_NaviSwitch: Wert des CX1100 Input 'IN'

Enum E_CX1100_NaviSwitch

```

TYPE E_CX1100_NaviSwitch : (
  e_CX1100_NaviSwitch_IDLE := 0,
  e_CX1100_NaviSwitch_MIDDLE := 16,

  (* clockwise in 45 degree steps *)
  e_CX1100_NaviSwitch_TOP := 1,
  e_CX1100_NaviSwitch_TOPRIGHT := 9,
  e_CX1100_NaviSwitch_RIGHT := 8,
  e_CX1100_NaviSwitch_BOTTOMRIGHT := 10,
  e_CX1100_NaviSwitch_BOTTOM := 2,
  e_CX1100_NaviSwitch_BOTTOMLEFT := 6,
  e_CX1100_NaviSwitch_LEFT := 4,
  e_CX1100_NaviSwitch_TOPLEFT := 5,

  (* clockwise in 45 degree steps with middle switch pressed *)
  e_CX1100_NaviSwitch_MIDDLE_TOP := 17,
  e_CX1100_NaviSwitch_MIDDLE_TOPRIGHT := 25,
  e_CX1100_NaviSwitch_MIDDLE_RIGHT := 24,
  e_CX1100_NaviSwitch_MIDDLE_BOTTOMRIGHT := 26,
  e_CX1100_NaviSwitch_MIDDLE_BOTTOM := 18,
  e_CX1100_NaviSwitch_MIDDLE_BOTTOMLEFT := 22,
  e_CX1100_NaviSwitch_MIDDLE_LEFT := 20,
  e_CX1100_NaviSwitch_MIDDLE_TOPLEFT := 21
)
END_VAR

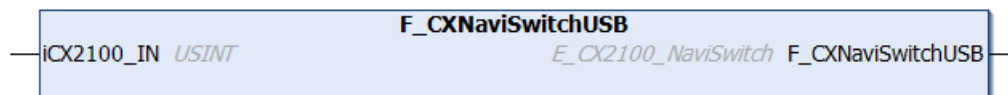
```

Andere Werte als im Enum definiert (z.B. 11) werden als "*** INVALID: Wert ***" im Online Mode angezeigt (z.B. "*** INVALID: 11 ***"). Die Funktion F_CXNaviSwitch liefert dann den ungültigen Wert als Rückgabewert (z.B. 11).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

4.3 F_CXNaviSwitchUSB



Die Funktion F_CXNaviSwitchUSB wandelt den Wert des CX2100- Navigationsschalters oder einer EL6090 in einen Enum- Wert vom Typ E_CX2100_NaviSwitch.

FUNCTION F_CXNaviSwitchUSB: E_CX2100_NaviSwitch

```

VAR_INPUT
  icx2100_IN : USINT;
END_VAR

```

icx2100_IN: Wert des CX2100 Input ,IN'

Enum E_CX2100_NaviSwitch

```

TYPE E_CX2100_NaviSwitch : (
  e_CX2100_NaviSwitch_IDLE := 0,
  e_CX2100_NaviSwitch_MIDDLE := 15,
  e_CX2100_NaviSwitch_ENTER := 16,
  e_CX2100_NaviSwitch_ENTER_MIDDLE := 31,

  (* clockwise in 45 degree steps, 1-2 switches on *)
  e_CX2100_NaviSwitch_TOP := 1,
  e_CX2100_NaviSwitch_TOPRIGHT := 9,
  e_CX2100_NaviSwitch_RIGHT := 8,
  e_CX2100_NaviSwitch_BOTTOMRIGHT := 10,
  e_CX2100_NaviSwitch_BOTTOM := 2,
  e_CX2100_NaviSwitch_BOTTOMLEFT := 6,
  e_CX2100_NaviSwitch_LEFT := 4,
  e_CX2100_NaviSwitch_TOPLEFT := 5,

  (* clockwise in 90 degree, 3 switches on*)
  e_CX2100_NaviSwitch_TOPLEFTRIGHT := 13,

```



```
e_CX2100_NaviSwitch_RIGHTTOPBOTTOM := 11,
e_CX2100_NaviSwitch_BOTTOMLEFTRIGHT := 14,
e_CX2100_NaviSwitch_LEFTTOPBOTTOM := 7,

(* clockwise in 45 degree steps with enter switch pressed, 1-2 switches on *)
e_CX2100_NaviSwitch_ENTER_TOP := 17,
e_CX2100_NaviSwitch_ENTER_TOPRIGHT := 25,
e_CX2100_NaviSwitch_ENTER_RIGHT := 24,
e_CX2100_NaviSwitch_ENTER_BOTTOMRIGHT := 26,
e_CX2100_NaviSwitch_ENTER_BOTTOM := 18,
e_CX2100_NaviSwitch_ENTER_BOTTOMLEFT := 22,
e_CX2100_NaviSwitch_ENTER_LEFT := 20,
e_CX2100_NaviSwitch_ENTER_TOPLEFT := 21

(* clockwise in 90 degree steps with enter switch pressed, 3 switches on *)
e_CX2100_NaviSwitch_ENTER_TOPLEFTRIGHT := 29,
e_CX2100_NaviSwitch_ENTER_RIGHTTOPBOTTOM := 27,
e_CX2100_NaviSwitch_ENTER_BOTTOMLEFTRIGHT := 30,
e_CX2100_NaviSwitch_ENTER_LEFTTOPBOTTOM := 23,
)
```

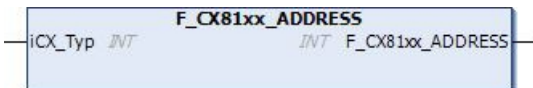
Voraussetzungen bei Verwendung des CX2100

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, TC/ BSD: TC RT x64)	Tc2_SystemCX

Voraussetzungen bei Verwendung der EL6090

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/ BSD: TC RT x64, TC OS ARMT2)	Tc2_SystemCX

4.4 F_CX81xx_ADDRESS



Diese Funktion liest die Stellung des Adresswahlschalters beim CXxxxx aus. Eine mögliche Anwendung ist, dass Sie abhängig von der Schalterstellung verschiedene Programmteile in der SPS aktivieren können.

Diese Funktion liest die Stellung des DIP-Schalters beim CXxxxx aus. Eine mögliche Anwendung ist, dass Sie abhängig von der Schalterstellung verschiedene Programmteile in der SPS aktivieren können.

VAR_INPUT

```
VAR_INPUT
  iCX_Typ : INT; (* Use product code without 'CX' e.g.: CX8180 -> 8180 *)
END_VAR
```

VAR_OUTPUT

```
F_CX80xx_ADDRESS : INT;
```

F_CX80xx_ADDRESS : -1, nicht implementierter CX, Adresse des Schalters

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Hardware	Einzubindende SPS Bibliotheken
TwinCAT v3.1 Build 4022.30	ARM	CXxxxx	Tc2_SystemCX

4.5 F_CX8180_LED_ERR



Da der CX8180 verschiedene Protokolle unterstützt, werden die beiden LEDs WD und ERR auf dem CX8180 nicht durch die Firmware genutzt. Dadurch hat der Anwender die Möglichkeit, eigene Diagnosemeldungen zu erstellen. Mit den LEDs können Sie zum Beispiel anzeigen, ob der CX8180 eine IP-Adresse vom DHCP Server erhalten hat oder ob sich Teilnehmer im Datenaustausch befinden.

Die Funktion F_CX8180_LED_ERR steuert die ERR-LED auf dem CX8180 an. Hier können verschiedene Farben und Blinkmodi benutzt werden. Mögliche Farben für die LEDs sind rot und grün.

VAR_INPUT

```

VAR_INPUT
  eMode      : E_CX81x0_LED;
END_VAR
  
```

eMode: Art und Weise, wie die LED leuchtet, siehe auch [E_CX81x0_LED \[► 47\]](#).

Return value

```

F_F_CX8180_LED_ERR      : INT;
  
```

F_CX8180_LED_ERR: -1, nicht implementierter Blink Code, 0 OK

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Hardware	Einzubindende SPS Bibliotheken
TwinCAT v3.1 Build 4022.30	ARM	CXxxxx	Tc2_SystemCX

4.6 F_CX8180_LED_WD



Da CX8180 verschiedene Protokolle unterstützt, werden die beiden LEDs WD und ERR auf dem CX8180 nicht durch die Firmware genutzt. Dadurch hat der Anwender die Möglichkeit, eigene Diagnosemeldungen zu erstellen. Mit den LEDs können Sie zum Beispiel anzeigen, ob der CX8180 eine IP-Adresse vom DHCP Server erhalten hat oder ob sich Teilnehmer im Datenaustausch befinden.

Die Funktion F_CX8180_LED_WD steuert die WD-LED auf dem CX8180 an. Hier können verschiedene Farben und Blinkmodi benutzt werden. Mögliche Farben für die LEDs sind rot und grün.

VAR_INPUT

```

VAR_INPUT
  eMode      : E_CX81x0_LED;
END_VAR
  
```

eMode: Art und Weise wie die LED leuchtet, siehe auch [E_CX81x0_LED \[► 47\]](#).

Return value

```

F_CX8180_LED_WD      : INT;
  
```

F_CX8180_LED_WD: -1, nicht implementierter Blink Code, 0 OK

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Hardware	Einzubindende SPS Bibliotheken
TwinCAT v3.1 Build 4022.30	ARM	CXxxxx	Tc2_SystemCX

4.7 F_CX8190_LED_ERR



Da der CX8190 verschiedene Protokolle unterstützt, werden die beiden LEDs WD und ERR auf dem CX8190 nicht durch die Firmware genutzt. Dadurch hat der Anwender die Möglichkeit, eigene Diagnosemeldungen zu erstellen. Mit den LEDs können Sie zum Beispiel anzeigen, ob der CX8190 eine IP-Adresse vom DHCP Server erhalten hat oder ob sich Teilnehmer im Datenaustausch befinden.

Die Funktion F_CX8190_LED_ERR steuert die ERR-LED auf dem CX8190 an. Hier können verschiedene Farben und Blinkmodi benutzt werden. Mögliche Farben für die LEDs sind rot und grün.

VAR_INPUT

```
VAR_INPUT
    eMode      : E_CX81x0_LED;
END_VAR
```

eMode: Art und Weise wie die LED leuchtet, siehe auch [E_CX81x0_LED \[▶ 47\]](#).

Return value

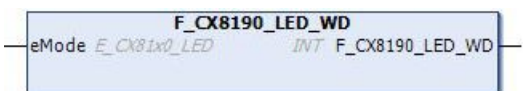
```
F_F_CX8190_LED_ERR : INT;
```

F_CX8190_LED_ERR: -1, nicht implementierter Blink Code, 0 OK

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Hardware	Einzubindende SPS Bibliotheken
TwinCAT v3.1 Build 4022.30	ARM	CXxxxx	Tc2_SystemCX

4.8 F_CX8190_LED_WD



Da CX8190 verschiedene Protokolle unterstützt, werden die beiden LEDs WD und ERR auf dem CX8190 nicht durch die Firmware genutzt. Dadurch hat der Anwender die Möglichkeit, eigene Diagnosemeldungen zu erstellen. Mit den LEDs können Sie zum Beispiel anzeigen, ob der CX8190 eine IP-Adresse vom DHCP Server erhalten hat oder ob sich Teilnehmer im Datenaustausch befinden.

Die Funktion F_CX8190_LED_WD steuert die WD-LED auf dem CX8190 an. Hier können verschiedene Farben und Blinkmodi benutzt werden. Mögliche Farben für die LEDs sind rot und grün.

VAR_INPUT

```
VAR_INPUT
    eMode      : E_CX81x0_LED;
END_VAR
```

eMode: Art und Weise wie die LED leuchtet, siehe auch [E_CX81x0_LED \[▶ 47\]](#).

Return value

```
F_CX8190_LED_WD : INT;
```

F_CX8190_LED_WD: -1, nicht implementierter Blink Code, 0 OK

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Hardware	Einzubindende SPS Bibliotheken
TwinCAT v3.1 Build 4022.30	ARM	CXxxxx	Tc2_SystemCX

5 Datentypen

5.1 [veraltet]

5.1.1 ST_CxDeviceIdentification

```

TYPE ST_CxDeviceIdentification :
STRUCT
  strTargetType      : STRING(20);
  strHardwareModel   : STRING(10);
  strHardwareSerialNo : STRING(12);
  strHardwareVersion : STRING(4);
  strHardwareDate    : STRING(10);
  strHardwareCPU     : STRING(10);
  strImageDevice     : STRING(20);
  strImageVersion    : STRING(10);
  strImageLevel      : STRING(10);
  strImageOsName     : STRING(20);
  strImageOsVersion  : STRING(8);
  strTwinCATVersion  : STRING(4);
  strTwinCATRevision : STRING(4);
  strTwinCATBuild    : STRING(8);
  strTwinCATLevel    : STRING(20);
  strAmsNetId       : STRING(23);
END_STRUCT
END_TYPE
    
```

strTargetType: Typ des Zielsystems, z.B. 'CX1000-CE', ...

strHardwareModel: Hardware-Modell, z.B. '1001'

strHardwareSerialNo: Hardware-Seriennummer, z.B. '123'

strHardwareVersion: Hardware-Version, z.B. '1.7'

strHardwareDate: Hardware-Produktionsdatum, z.B. '18.8.06'

strHardwareCPU: Hardware-CPU-Architektur, z.B. 'INTELx86', 'ARM', 'UNKNOWN' oder '' (leerer String)

strImageDevice: Software-Plattform, z.B. 'CX1000', ...

strImageVersion: Version der Software-Plattform, z.B. '2.15'

strImageLevel: Level der Software-Plattform, z.B. 'HMI'

strImageOsName: Name des Betriebssystems, z.B. 'Windows CE'

strImageOsVersion: Version des Betriebssystems, z.B. '5.0'

strTwinCATVersion: TwinCAT Version, z.B. bei TwinCAT 2.10.1307: '2'

strTwinCATRevision: TwinCAT Reversion, z.B. bei TwinCAT 2.10.1307: '10'

strTwinCATBuild: TwinCAT Build, z.B. bei TwinCAT 2.10.1307: '1307'

strTwinCATLevel: Registrierter TwinCAT Level, z.B. 'PLC', 'NC-PTP', 'NC-I', ...

strAmsNetId: TwinCAT AMS-NetID, z.B. '5.0.252.31.1.1'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

5.1.2 ST_CxDeviceIdentificationEx

```

TYPE ST_CxDeviceIdentificationEx :
STRUCT
  strTargetType      : STRING(30);
    
```

```

strHardwareModel      : STRING(16);
strHardwareSerialNo  : STRING(16);
strHardwareVersion   : STRING(8);
strHardwareDate      : STRING(12);
strHardwareCPU       : STRING(20);
strImageDevice       : STRING(48);
strImageVersion      : STRING(32);
strImageLevel        : STRING(32);
strImageOsName       : STRING(48);
strImageOsVersion    : STRING(8);
strTwinCATVersion    : STRING(4);
strTwinCATRevision   : STRING(4);
strTwinCATBuild      : STRING(8);
strTwinCATLevel      : STRING(20);
strAmsNetId          : T_AMSNetId;
END_STRUCT
END_TYPE

```

strTargeType: Typ des Zielsystems, z.B. 'CX1000-CE', ...

strHardwareModel: Hardware- Modell, z.B. '1001'

strHardwareSerialNo: Hardware- Seriennummer, z.B. '123'

strHardwareVersion: Hardware- Version, z.B. '1.7'

strHardwareDate: Hardware- Produktionsdatum, z.B. '18.8.06'

strHardwareCPU: Hardware- CPU- Architektur, z.B. 'INTELx86', 'ARM', 'UNKNOWN' oder '' (leerer String)

strImageDevice: Software-Plattform, z.B. 'CX1000', ...

strImageVersion: Version der Software-Plattform, z.B. '2.15'

strImageLevel: Level der Software-Plattform, z.B. 'HMI'

strImageOsName: Name des Betriebssystems, z.B. 'Windows CE'

strImageOsVersion: Version des Betriebssystems, z.B. '5.0'

strTwinCATVersion: TwinCAT Version, z.B. bei TwinCAT 2.10.1307: '2'

strTwinCATRevision: TwinCAT Reversion, z.B. bei TwinCAT 2.10.1307: '10'

strTwinCATBuild: TwinCAT Build, z.B. bei TwinCAT 2.10.1307: '1307'

strTwinCATLevel: Registrierter TwinCAT Level, z.B. 'PLC', 'NC-PTP', 'NC-I', ...

strAmsNetId: TwinCAT AMS-NetID, z.B. '5.0.252.31.1.1'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

5.2 ST_CX_ProfilerStruct

```

TYPE ST_CX_ProfilerStruct :
STRUCT
  dwLastExecTime      : DWORD;
  dwMinExecTime       : DWORD;
  dwMaxExecTime       : DWORD;
  dwAverageExecTime   : DWORD;
  dwMeasureCycle      : DWORD;
END_STRUCT
END_TYPE

```

dwLastExecTime: Der letzte gemessene Wert der Ausführungszeit in [µs]

dwMinExecTime: Die minimale Ausführungszeit in [µs]

dwMaxExecTime: Die maximale Ausführungszeit in [µs]

dwAverageExecTime: Die mittlere Ausführungszeit der 100 letzten Messungen in [µs]

dwMeasureCycle: Anzahl der bereits durchgeführten Messungen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.0	CX (x86)	Tc2_SystemCX

5.3 E_CX81x0_LED

```
TYPE E_CX81x0_LED : (  
    eCX81x0_LED_GREEN_OFF,  
    eCX81x0_LED_GREEN_ON,  
    eCX81x0_LED_GREEN_FLASHING_Quick,  
    eCX81x0_LED_GREEN_FLASHING_200ms,  
    eCX81x0_LED_GREEN_FLASHING_500ms,  
    eCX81x0_LED_GREEN_FLASHING_Pulse,  
    eCX81x0_LED_RED_OFF:=20,  
    eCX81x0_LED_RED_ON,  
    eCX81x0_LED_RED_FLASHING_Quick,  
    eCX81x0_LED_RED_FLASHING_200ms,  
    eCX81x0_LED_RED_FLASHING_500ms,  
    eCX81x0_LED_RED_FLASHING_Pulse,  
    eCX81x0_LED_GREEN_RED_OFF:=100,  
    eCX81x0_LED_GREEN_RED_FLASHING_200ms,  
    eCX81x0_LED_GREEN_RED_FLASHING_500ms  
);  
END_TYPE
```

6 Globale Konstanten

6.1 Bibliotheksversion

Alle Bibliotheken haben eine bestimmte Version. Diese Version ist u. a. im SPS-Bibliotheks-Repository zu sehen. Eine globale Konstante enthält die Information über die Bibliotheksversion:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_SystemCX : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_SystemCX: Versionsinformation der Tc2_SystemCX-Bibliothek (Typ: ST_LibVersion)

Um zu sehen, ob die Version, die Sie haben auch die Version ist, die Sie brauchen, benutzen Sie die Funktion F_CmpLibVersion (definiert in der Tc2_System-Bibliothek).



Alle anderen Möglichkeiten Bibliotheksversionen zu vergleichen, die Sie von TwinCAT 2 kennen, sind veraltet!

Mehr Informationen:
www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

