

Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: TcSystemCX



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Funktionen	9
3.1	F_CXSubTimeStamp	9
3.2	F_CXNaviSwitch	9
3.3	F_GetVersionTcSystemCX	10
4	Funktionsbausteine	11
4.1	FB_CxGetDeviceIdentification	11
4.2	FB_CXProfiler	11
4.3	FB_CXSetTextDisplay	12
4.4	FB_CXSimpleUps	14
5	Datentypen	16
5.1	ST_CX_DeviceIdentification	16
5.2	ST_CX_ProfilerStruct	16

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte 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 weiter entwickelt.

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.

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!

Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

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.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.



Tipps oder Fingerzeige

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

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 in unserem <https://www.beckhoff.de/secguide>.

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 unter <https://www.beckhoff.de/secinfo>.

2 Übersicht

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

Funktionen

Name	Beschreibung
F_CXSubTimeStamp [► 9]	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 [► 9]	Mit dieser Funktion wird der Wert des CX1100-Navigationsschalters in einen Enum gewandelt.
F_GetVersionTcSystemCX [► 10]	Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

Funktionsbausteine

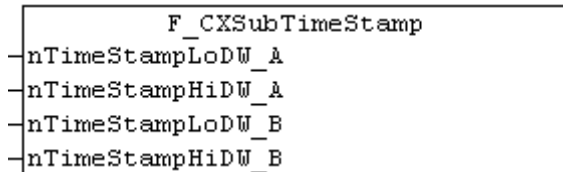
Name	Beschreibung
FB_CxGetDeviceIdentification [► 11]	Auslesen der Geräteidentifikation des CX-Gerätes (siehe Anforderungen)
FB_CXProfiler [► 11]	Laufzeitmessung von SPS-Code über den CPU-Counter
FB_CXSetTextDisplay [► 12]	Ansteuerung des zweizeiligen Displays des CX1100
FB_CXSimpleUps [► 14]	Ansteuerung der USV CX1190-UPS (Gerätename CX1100-0900, CX1100-0910, CX1100-0920)

Voraussetzungen

Komponente	Version
TwinCAT auf dem Entwicklungsrechner	2.9 Build 959 oder höher
CX1000-Windows CE-Image	1.75 oder höher; FB_CxGetDeviceIdentification [► 11] ab 2.15
CX1000-Windows XP-Image	1.14 oder höher; FB_CxGetDeviceIdentification [► 11] höher als 1.33
CX1020-Windows CE-Image	2.04 oder höher; FB_CxGetDeviceIdentification [► 11] ab 2.15
CX1020-Windows XP-Image	1.30 oder höher; FB_CxGetDeviceIdentification [► 11] höher als 1.33

3 Funktionen

3.1 F_CXSubTimeStamp



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
```

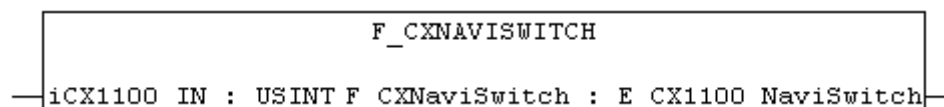
nTimeStampLoDW_A: Untere 32bit des Zeitstempels A.

nTimeStampHiDW_A: Obere 32bit des Zeitstempels A.

nTimeStampLoDW_B: Untere 32bit des Zeitstempels B.

nTimeStampHiDW_B: Obere 32bit des Zeitstempels B.

3.2 F_CXNaviSwitch



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

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).

3.3 F_GetVersionTcSystemCX

```

F_GetVersionTcSystemCX
nVersionElement

```

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:

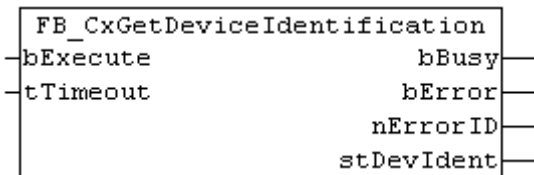
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	PC (i386)	TcSystemCX.Lib

4 Funktionsbausteine

4.1 FB_CxGetDeviceIdentification



Mit dem Funktionsbaustein FB_CxGetDeviceIdentification können Geräteidentifikationsdaten des CX ausgelesen werden.

VAR_INPUT

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

bExecute: Mit der steigende Flanke wird das Kommando ausgeführt.

tTimeout: Gibt die Zeit bis zum Abbrechen der Funktion an.

VAR_OUTPUT

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

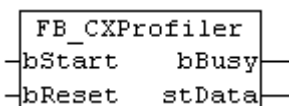
bBusy: Die Daten werden aus dem CX ausgelesen. Nach fehlerfreier Ausführung stehen bei **bBusy = FALSE** die Daten in der Struktur **stDevIdent**.

bError: Wird TRUE, sobald ein Fehler eintritt.

nErrorID: Liefert bei einem gesetzten **bError**-Ausgang die Fehlernummer.

stDevIdent: Enthält die gelesenen Gerätedaten. Siehe [ST_CxDeviceIdentification](#) [► 16].

4.2 FB_CXProfiler



Mit dem Funktionsbaustein FB_CXProfiler können unter Microsoft Windows CE Messungen der Ausführungszeit vom SPS-Code durchgeführt werden. Bei Verwendung von Microsoft Windows XP embedded empfiehlt sich die Verwendung des genaueren Funktionsbausteins Profiler aus der [TcUtilities.lib](#).

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](#) [► 16] 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](#) [► 16] 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.



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 ist in den ermittelten Zeiten enthalten. Taskunterbrechungen z. B. durch die NC werden nicht erkannt und führen zu größeren Messzeiten.

VAR_INPUT

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

bStart: 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 stData.dwMeasureCycle wird dabei inkrementiert.

bReset: 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.

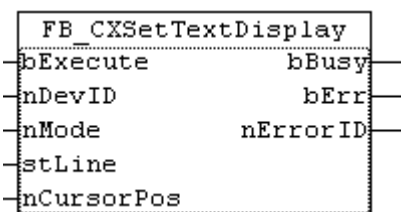
VAR_OUTPUT

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

bBusy: 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: Struktur vom Typ [ST_CX_ProfilerStruct](#) [► 16] mit den gemessenden Zeiten in [µs].

4.3 FB_CXSetTextDisplay



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

VAR_INPUT

```
VAR_INPUT
  bExecute     : BOOL;
  nDevID       : UDINT;
  nMode        : E_CX1100_DisplayModes;
  stLine       : STRING(20);
  nCursorPos   : DWORD;
END_VAR
```

bExecute: Mit der steigende Flanke wird das Kommando ausgeführt.

nDevID: Device ID des CX1100-Gerätes.

nMode: Modeumschaltung (s. Enumeration)

stLine: String mit 20 Zeichen. Dieser String wird mit dem entsprechenden Kommando im Display angezeigt.

nCursorPos: Cursorposition. Ab dieser Stelle im Display wird der String geschrieben.

VAR_OUTPUT

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

bBusy: Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bErr: Wird TRUE, sobald ein Fehler eintritt.

nErrorID: 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.

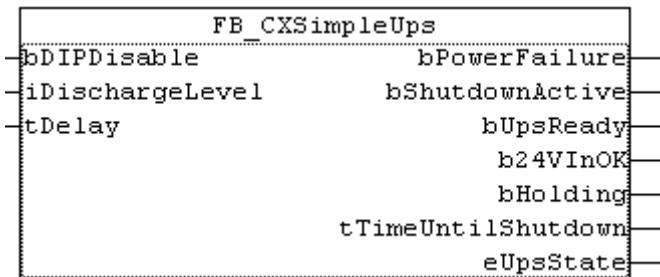
e_CX1100_BackLightOff: Ausschalten der Hintergrundbeleuchtung.

e_CX1100_ClearDisplay: Löschen des Bildschirminhaltes.

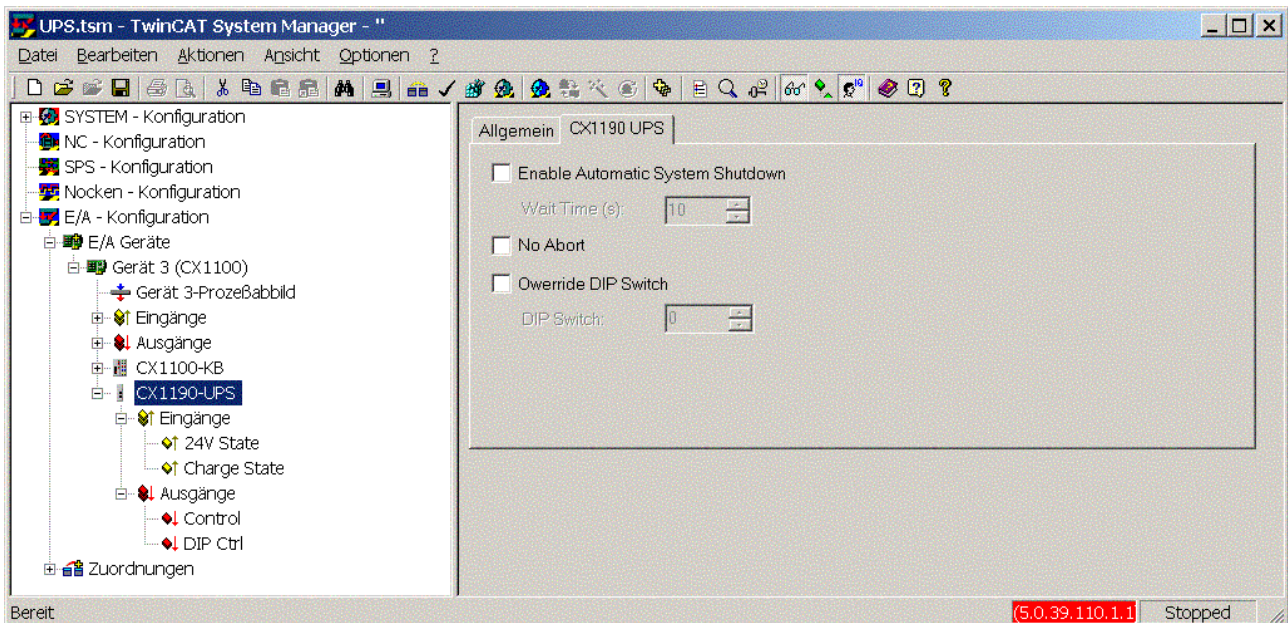
e_CX1100_WriteLine1: Schreiben der ersten Zeile.

e_CX1100_WriteLine2: Schreiben der zweiten Zeile.

4.4 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.



VAR_INPUT

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

bDIPDisable: Wenn TRUE, dann wird die Stellung des Ladezustandsschalters an der USV ignoriert und stattdessen iDischargeLevel verwendet.

iDischargeLevel: Ausschaltsschwelle für Entladung: 0 = 100% (Maximale Entladung), 9 = 90%, 8 = 80%, ..., 2 = 20%, 1 = 10% (Minimale Entladung).

tDelay: Haltezeit, bevor der Shutdown durchgeführt wird. Sie wird benutzt, um kurze Spannungsausfälle (bis zu 10s) zu überbrücken. Nach Überschreiten der 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.

VAR_OUTPUT

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

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

bPowerFailure: Wird TRUE, wenn ein Spannungsausfall der Versorgungsspannung erkannt wird, wird FALSE, wenn die Eingangsspannung wiederkehrt.

bShutdownActive: Wird TRUE, sobald der Stop bzw. Shutdown ausgeführt wird.

bUpsReady: Wird TRUE, sobald die USV die Ausgangsspannung bereitstellt.

b24VInOK: Wird TRUE, sobald die 24V-Eingangsspannung bereitsteht.

bHolding: Wird TRUE, wenn ein Spannungsausfall der Versorgungsspannung erkannt wird und die Haltezeit noch nicht abgelaufen ist.

tTimeUntilShutdown: Zeigt die verbleibende Haltezeit bis zum Shutdown an.

eUpsState: Zeigt den Status der USV an [UNDEF | CHARGING | CHARGED | DISCHARGE | DISCHARGE_RESTART | OUTPUT_OFF | OVERLOAD].

VAR_CONFIG

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

Ii24VState: Muss mit Eingang '24V State' gelinkt sein, siehe Bild oben.

IiChargeState: Muss mit Eingang 'Charge State' gelinkt sein, siehe Bild oben.

QiControl: Muss mit Ausgang 'Contol' gelinkt sein, siehe Bild oben.

QiDipControl: Muss mit Ausgang 'DIP Ctrl' gelinkt sein, siehe Bild oben.

5 Datentypen

5.1 ST_CX_DeviceIdentification

```

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'.

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 [μ s].

dwMinExecTime: Die minimale Ausführungszeit in [μ s].

dwMaxExecTime: Die maximale Ausführungszeit in [μ s].

dwAverageExecTime: Die mittlere Ausführungszeit der 100 letzten Messungen in [μ s].

dwMeasureCycle: Anzahl der bereits durchgeführten Messungen.

Mehr Informationen:
www.beckhoff.de/tx1200

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

