

Zählermodul PCD2/3.H100

Inhalt

1. EINFÜHRUNG	2
1.1 Funktionsbeschreibung	2
1.2 Mögliche Anwendungen	2
1.3 Verwendete Hard- und Software	3
 2. GETTING STARTED MIT DEM H100.....	 4
2.1 Die PCD vorbereiten.....	4
2.1.1 Installation des Projekts	5
2.1.2 Anpassen von Hardware- und Software Einstellungen im PG5 ..	6
2.1.3 Das Programm and die vorhandene PCD anpassen	6
2.1.4 Building und Laden des Projekts in die PCD.....	6
2.2 Werte online beobachten.....	7
2.2.1 Das SAIA Watch Window.....	7
2.3 H100 Beispiele	8
2.3.1 Beispiel 1: H100_SingleCount-Down.sfc	8
2.3.2 Beispiel 2: H100_SingleCount-Down_extended.sfc.....	11
 3. KOMMANDOS FÜR DIE PROGRAMMIERUNG DES H100	 14
3.1.1 Schreiben auf das Modul	14
3.1.2 Lesen vom Modul.....	15

1. Einführung

Dieses Dokument hat zwei Funktionen. Einerseits dient es zur Dokumentation des Beispielprojekts PCD2-3_H100_Counting. Dieses Projekt soll aufzeigen, wie man mit H100 Modulen z.B. Impulse zählen oder Umdrehungen eines Inkrementalencoders messen kann. Folgende Beispiele sind in diesem Dokument enthalten:

- Von einem vordefinierten Wert (0..65535) abwärts zählen und bei Erreichen des Werts 0 den Ausgang CCO (Counter Controlled Output) setzen.
- Von einem vordefinierten Wert (>65535) abwärts zählen und bei Erreichen des Werts 0 den Ausgang CCO setzen.

Zusätzlich enthält dieses Dokument die Beschreibung der Kommandos für die Kontrolle des PCD2/3.H110 (Kapitel 3).

1.1 Funktionsbeschreibung

Das Zählermodul H100 für die PCD1, PCD2 und PCD3 kann eigenständig (asynchron zur PCD CPU) Zählaufgaben erledigen (Inkrementieren und Dekrementieren). Die Input Filter des H100 ist für Frequenzen bis zu 20 kHz ausgelegt, was wesentlich schneller ist als die Filter von digitalen Eingansmodulen wie dem E110.

Die Funktionen des H100 können mit einfachen PCD Instruktionen oder Kombinationen davon kontrolliert werden. Zudem kann der Zählmode mittels einem Jumper auf dem Modul eingestellt werden.

1.2 Mögliche Anwendungen

- Zählen von Umdrehungen oder Distanzen (Impulse)
- Von einem voreingestellten Wert abwärts zählen und den Ausgang CCO bei Erreichen von dem Wert 0 Ausschalten.
- Zählen während einer zusätzlichen Bedingung, z.B. einer aktiven fotoelektronischen Barriere.
- Zählen mit Erkennung der Zählrichtung eines Inkrementalencoders. Somit kann eine simple Motion Control Aufgabe erledigt werden.

1.3 Verwendete Hard- und Software

Hardware:

- PCD2.M480
- PCD2.E110 (für das Startsignal)
- PCD2.H100
- PCD8.K111 Programmierkabel oder USB Kabel)
- Ein Impulsgenerator. Das kann auch ein manuell bedienter Schalter/Taster sein.

In dem Beispielprojekt befindet sich auch ein Programm mit dem Namen Blinker.src. Dieses Programm ändert den Zustand des Ausgangs 16 auf dem optionalen Ausgangsmodule PCD2.A400 in Slot 1 in jedem Zyklus. Dieser Ausgang kann auch mit dem Eingang A auf dem H100 verbunden werden.

Um dieses Beispielprojekt mit einer andern Steuerung als einer PCD2.M480 zu betreiben müssen die entsprechenden Hardware Einstellungen im Projekt angepasst werden.

Mit der Funktion „Upload...“ des Hardware Konfigurators können diese Einstellungen direkt in eingelesen werden (siehe Kapitel 2.1.2, Hardware- und Software settings verändern).

Minimake Software Version:

SAIA PG5 1.4.110

2. Getting started mit dem H100

2.1 Die PCD vorbereiten

Prinzipiell können die verwendeten Module auf einem beliebigen Steckplatz (Slot) der PCD2.M480 eingesetzt werden. Damit keine Programmmodifikationen notwendig werden, sollten die Module in folgender Reihenfolge eingesetzt werden:

Slot 0 (Basisadresse 0):	PCD2.E110
Slot 1 (Basisadresse 16):	PCD2.A400 (optional)
Slot 2 (Basisadresse 32):	PCD2.H100

Sollte eine andere Konfiguration verwendet werden, so müssen die Basisadressen der Module in dem Symboleditor angepasst werden.

Die Verdrahtung der Module erfolgt gemäss dem Handbuch 26/737 „PCD1-PCD2_D14“.

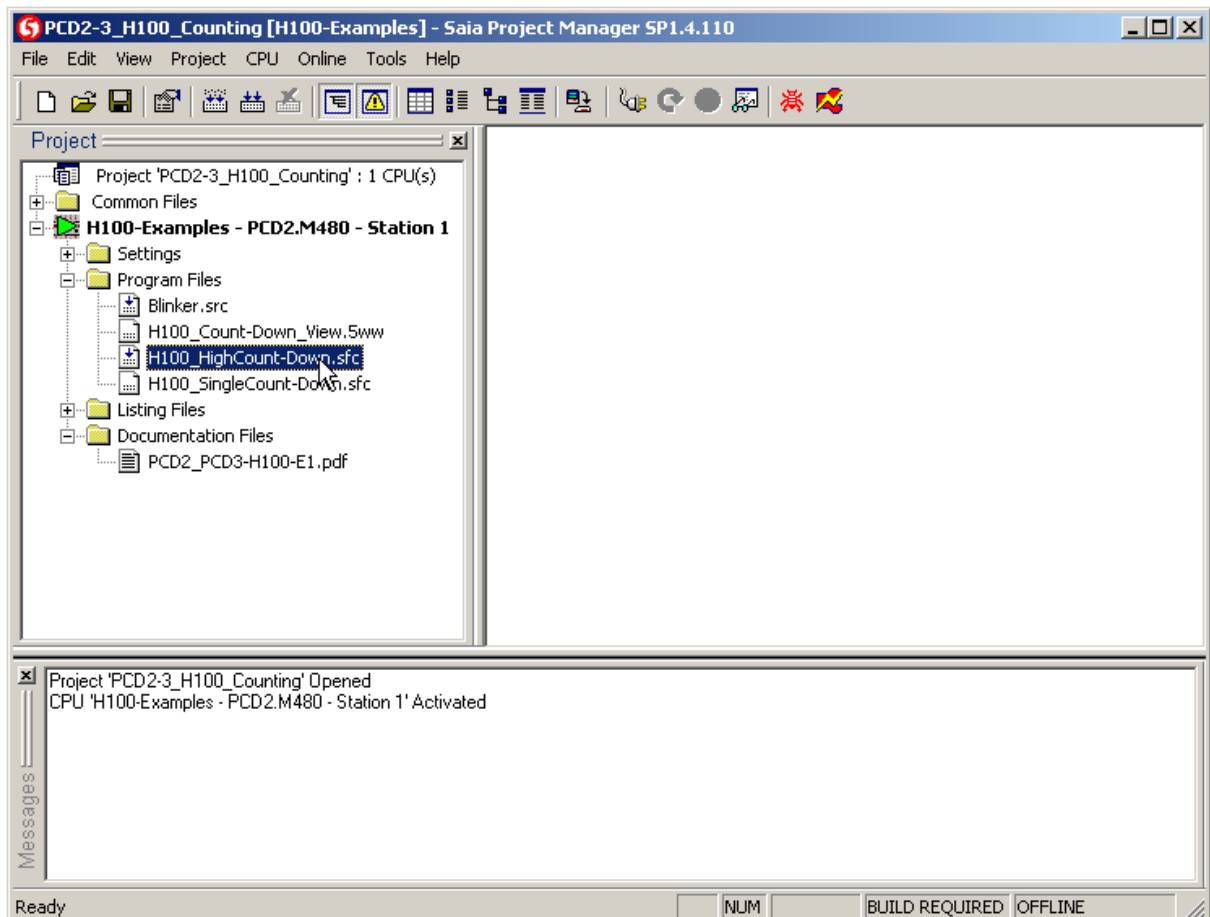
Wenn Impulse durch das optionale PCD2.A400 auf Steckplatz 1 generiert werden, so muss der Ausgang 16 mit dem Eingang „A“ auf dem PCD2.H100 verbunden werden.



Der I/O Bus von Saia PCD Steuerungen ist nicht für “hot plugging” ausgelegt! Vor dem Einsetzen oder Entfernen von I/O Modulen muss die PCD immer stromlos gemacht werden!

2.1.1 Installation des Projekts

Für die Installation des projekts sollten Sie die Funktion „Restore...“ aus dem PG5 1.4 Project Manager Menü „Project“ verwenden. Diese Funktion wird das projekt (in *.zip Format) entpacken und und in Ihr Projektverzeichnis kopieren.



Dieses Dokument kann in dem Ordner „Documentation Files“ in dem PG5 Project Manager, im Projekt Fenster gefunden werden. Von hier aus kann es direkt mit einem Doppelklick geöffnet werden.

2.1.2 Anpassen von Hardware- und Software Einstellungen im PG5

Nachdem das PG5 geöffnet wurde und das Beispielprojekt installiert und geöffnet ist können folgende Schritte ausgeführt werden:

- Verbinden Sie die PCD mittels einem PGU- oder Programmierkabel (PCD8.K111) mit dem PC und schalten Sie die PCD an.
- Öffnen Sie die „Online settings“ (zu finden im Settings Ordner des Projektbaums im PG5 Project Manager). Passen Sie die Online Einstellung an (wählen Sie entweder PGU oder S-Bus USB mit aktivierter Checkbox „PGU“).
- Öffnen Sie das „Hardware settings“ Fenster (ebenfalls in dem Settings Ordner im Projekt-Baum des PG5 Project Managers).
- Wählen Sie „Upload“ um die Hardware Konfiguration der PCD in den PC zu laden. Nachdem das erfolgt ist, klicken Sie auf „OK“.
- Öffnen Sie das „Software settings“ Fenster (in dem Settings Ordner im Projekt-Baum des PG5 Project Managers).
- Wählen Sie „Set Default“ um den dynamisch Bereich der Ressourcen an die vorhandene PCD anzupassen und schliessen Sie das Fenster mit „OK“.
- „Linken“ sie die zu verwendenden Programmdateien (z.B. H100_SingleCount-Down.sfc und Blinker.src). Das wird bewerkstelligt im Kontext Menü der Datei (rechter Mausklick auf das gewünschte Programm und „Linked/Built“ wählen).

2.1.3 Das Programm and die vorhandene PCD anpassen

Um sicher zu stellen, dass das korrekte Module verwendet wird sollten folgende Einstellungen überprüft werden:

- Die Basisadresse des Moduls (z.B. H100.BaseAddress) muss mit der auf dem Basisprint der PCD erkennbaren Adresse übereinstimmen. Die Basisadresse ist immer ein Vielfaches von 16 und ist auf dem PCD Basisprint einer PCD1 oder PCD2 eingezeichnet.
Auf einer PCD3 sind nur die Steckplatznummern angegeben (mit 0 startend). Die Basisadresse kann errechnet werden, indem man die Steckplatzadresse mit 16 multipliziert.

Beachten Sie dass die Steckplatznummerierung auf einem Erweiterungsgehäuse fortgesetzt und nicht neu begonnen wird.



2.1.4 Building und Laden des Projekts in die PCD

Nachdem die oben beschriebenen Anpassungen durchgeführt wurden kann das Projekt kompiliert („Rebuild All“ im CPU Menü oder <Alt>+F2) und auf die PCD geladen werden („Download Program...“ im Menü „Online“).

Wenn die PCD bereits ein Programm abarbeitet wird das PG5 nachfragen, ob die PCD gestoppt werden kann. Das wird in diesem Fall zutreffen und kann bestätigt werden.

2.2 Werte online beobachten

Nachdem das Programm auf die PCD geladen wurde kann eine Onlineverbindung zu der PCD aufgebaut werden. Anschließend können alle Werte online beobachtet werden.

Die Verbindung kann durch ein Klicken auf den "Online" Button  aufgebaut werden. Sollte sich die PCD noch nicht im „Run“ status befinden kann die Abarbeitung des Programms mit dem „Run“ Button  gestartet werden.

Nun kann das Watch Window (siehe nächstes Kapitel) für die Anzeige von Oninewerten verwendet werden.

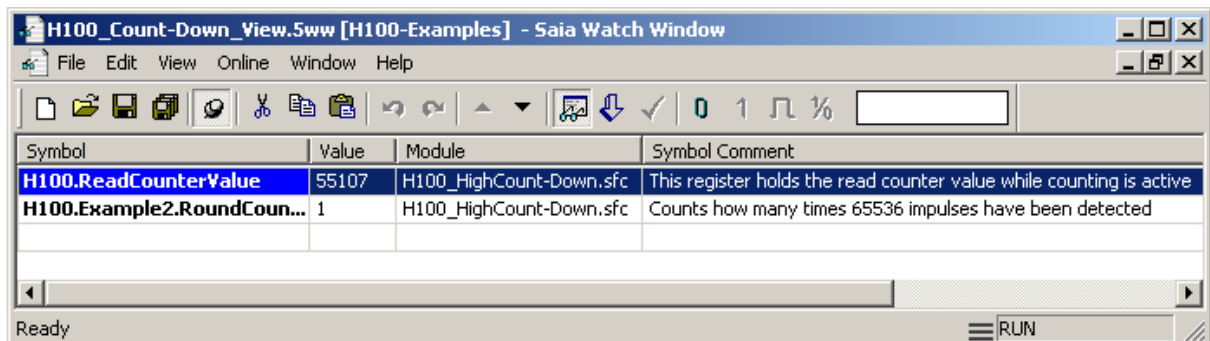


Theoretisch kann die Abarbeitung des Programms im AWL Editor online betrachtet werden. Da aber das H100 auf Zugriffe auf dem I/O Bus reagieren kann, wird das nicht empfohlen.

2.2.1 Das SAIA Watch Window

Um Werte auf der Steuerung zu betrachten und zu verändern wird das Watch Window eingesetzt. Das zu dem Beispiel gehörenden Watch Windw kann durch ein Doppelklick auf die Datei "H100_Count-Down_View.5ww" in dem PG5 Project Manager geöffnet werden.

Die anzuzeigenden Symbole können durch "drag and drop" in das Fenster gezogen werden. Eine Onlineverbindung wird aufgebaut duch das drücken der „F9“ Taste.



2.3 H100 Beispiele

Zählaufgaben werden praktisch immer in einer sequentiellen Programmstruktur realisiert:

1. Definition der Zählaufgabe
2. Start der Zählung
3. Warten auf das Ende der Zählung
4. Evaluieren des REsultats (z.B. vergangene Zeit seit Beginn der Zählung)

Als logische Schlussfolgerung wird das Programm für die Zählaufgabe in Graftec geschrieben.

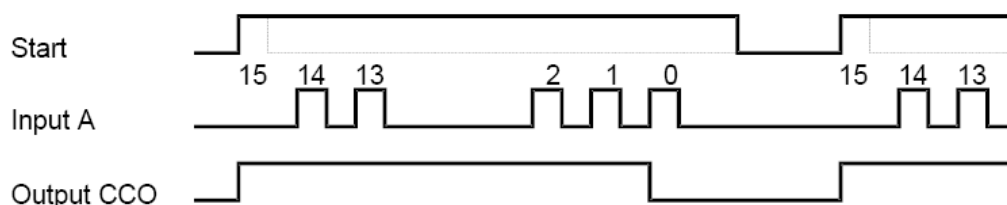
Da alle Programmbeispiele in dem Projekt PCD2-3_H100_Counting immer mit dem selben PCD2.H100 arbeiten und dieses nur einen Zählerkanal hat, kann nur ein Programm alleine ausgeführt werden. Deshalb darf nur entweder das Programm H100_SingleCount-Down.sfc oder das Programm H100_HighCount-Down.sfc kompiliert und auf die Steuerung geladen werden.

Das Programm Blinker.src kann immer mitkompiliert werden (selbst wenn sich kein PCD2.A400 auf der PCD befindet).

2.3.1 Beispiel 1: H100_SingleCount-Down.sfc

Dieses Programm erfüllt folgende Aufgabenstellung:

- Der Eingang 0 dient als Startsignal. Wenn eine ansteigende Flanke auf Eingang 0 erkannt wird, soll der Zähler des H100 mit dem Wert 15 geladen werden und der Ausgang CCO soll eingeschaltet werden.
- Nachdem 15 Impulse auf dem Zählereingang "A" erkannt wurden soll der Ausgang CCO ausgeschaltet werden.
- Wenn das Startsignal inaktiv wird und erneut ansteht (nachdem die Messung erfolgt ist) soll die Sequenz von neuem gestartet werden.
- Der aktuell anstehende Wert soll während der Zählung kontinuierlich betrachtet werden können (z.B. im Watch Window H100_Count-Down_View.5ww)



Vorgehen

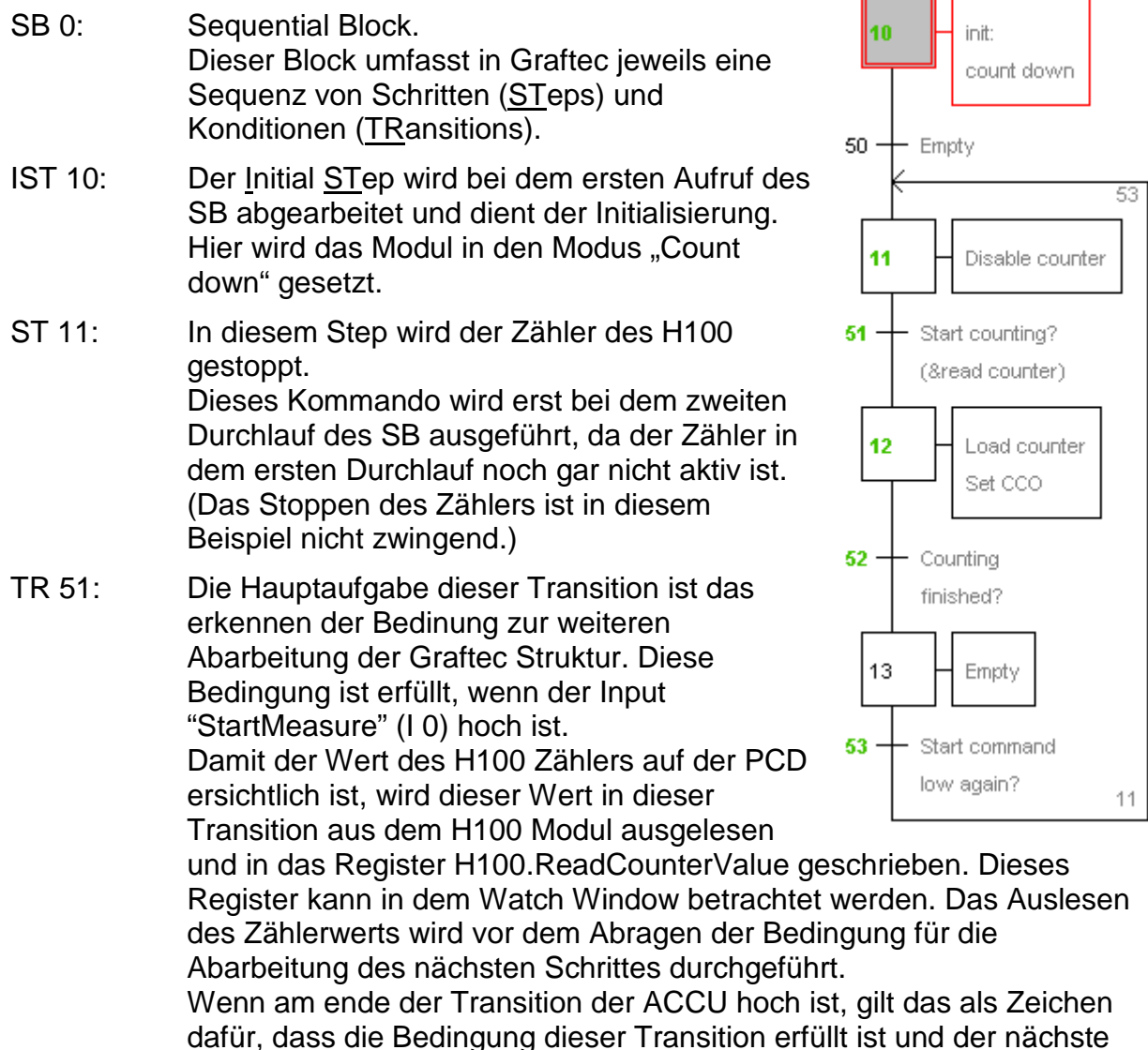
- Der Modus „Single count“ (SC) muss aktiv sein → Der Jumper auf dem H100 muss in Position "SC" sein.
- Der Zähler wird mit dem Wert 15 geladen.
- Abwärtszählen wird gestartet und funktioniert automatisch.

Der Eingang „A“ des Moduls PCD2.H100 muss mit der Impulsquelle (z.B. ein einfacher Schalter, ein Pulsgenerator oder der Ausgang 16 des optionalen A400 Moduls) verbunden sein

An Eingang "B" müssen 24 V anliegen, damit die Impulse Eingang „A“ erreichen (AND gate).

Beachten Sie dass die Eingangsfilter der Eingänge „A“ und „B“ für 20 kHz ausgelegt sind. Dadurch werden diese Eingänge bei Verwendung eines einfachen Schalters empfindlich auf Prellende Signale.

Beschreibung der einzelnen Schritte und Bedingungen (Beispiel 1)



Schritt wird abgearbeitet bevor wieder zurück in den COB (Cyclical Organisation Block) gesprungen wird.

Wenn am Ende der Transition der ACCU tief ist wird das Programm direkt wieder zurück in den COB springen. Bei dem nächsten Aufruf des SB wird dieselbe Transition wieder abgearbeitet und wird somit das Register H100.ReadCounterValue auffrischen.

ST 12: In diesem Schritt wird der Zähler mit dem Wert 15 geladen.
Anschliessend wird der Ausgang CCO (Counter Controlled Output) aktiviert und der Zähler des H100 wird gestartet.
Die letzte Anweisung

`RES H100.BaseAddress+14`

setzt das "CCO Reset Enable". Als Folge davon wird der Ausgang CCO ausgeschaltet, sobald der H100 Zähler Null erreicht.

TR 52: Nun arbeitet das H100 selbständig. Sobald der Zähler Null erreicht, wird der CCO durch das Modul ausgeschaltet.
Auch in diesem Schritt wird das Register H100.ReadCounterValue mit dem aktuellen Wert des H100 Zählers aufgefrischt.
Die Bedingung für das Abarbeiten des nächsten Schrittes wird das der Wert 0 des "Counter Flags" verwendet. Die Bedeutung dieses Flags ist, dass der Zähler des H100 den Wert 0 erreicht hat. Gelesen wird das Counter Flag mit der folgenden Anweisung

`STL H100.BaseAddress+0`

TR 53: Die Weiterschaltbedingung dieser Transition ist, dass der Input "StartMeasure" (I 0) gleich Null ist.

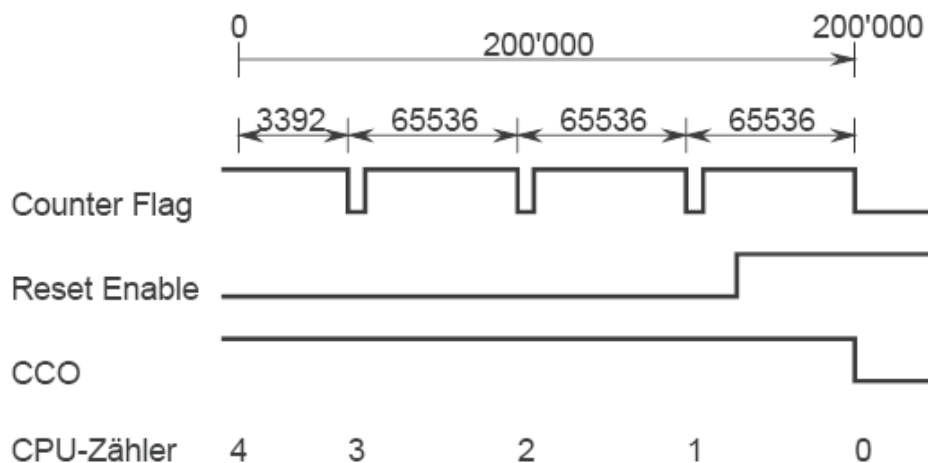
Anschliessend wird zu Schritt 11 gesprungen, in welchem der Zähler deaktiviert wird.

2.3.2 Beispiel 2: H100_SingleCount-Down_extended.sfc

Das zweite Beispiel erfüllt die Identische Aufgabe wie das erste Beispiel, aber mit dem Unterschied das ein Wert von mehr als 66535 (maximaler Wert des H100 Zählers) gezählt wird bevor der CCO (Counter Controlled Output) rückgesetzt wird.

Dieses Ziel wird dadurch erreicht, dass das „Reset CCO Enable“ erst nach mehreren „Runden“ aktiviert wird. Eine „Runde“ bedeutet, dass der Zähler von 65535 auf 0 abwärts gezählt hat. Diese Runden werden auf der PCD mit einem Counter gezählt. Bei jedem „Überqueren von 0“ des H100 Zählers wird dieser PCD Counter (H100.Example2.RoundCounter) dekrementiert. Erkannt wird eine „Überquerung von 0“ des H100 Zählers durch das „Counter Flag“ wie in Beispiel 1.

Nachdem der H100 Zähler vier mal 0 erreicht hat, wird das „Reset CCO Enable“ aktiviert, was zur folge hat, dass der Ausgang CCO beim nächsten Erreichen von 0 ausgeschaltet wird:



Vorgehen

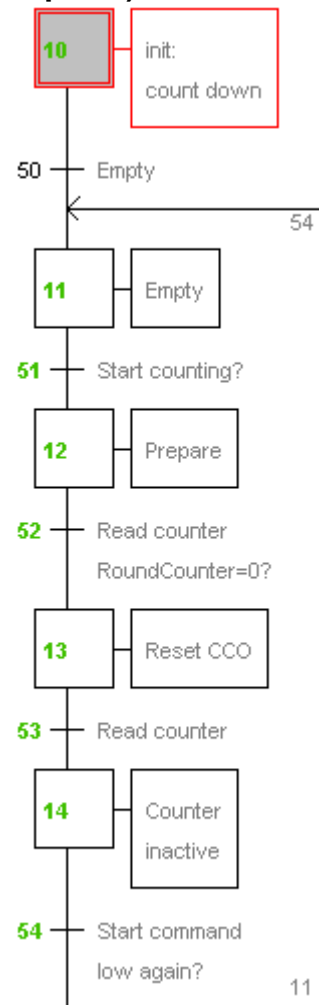
- Der Modus „Single count“ (SC) muss aktiv sein → Der Jumper auf dem H100 muss in Position "SC" sein.
- Der H100 Zähler wird mit dem Wert 3392 geladen und der CPU Counter H100.Example2.RoundCounter wird mit 4 geladen.
- Abwärtszählen wird gestartet und funktioniert automatisch. Immer wenn das H100 Counter Flag Null wird, wird es wieder auf 1 gesetzt und der CPU Counter H100.Example2.RoundCounter wird dekrementiert.
- Sobald das H100.Example2.RoundCounter = Null ist, wird das „Reset CCO Enable“ aktiviert. Somit wird der CCO rückgesetzt, sobald der H100 Zähler Null erreicht.

Der Eingang „A“ des Moduls PCD2.H100 muss mit der Impulsquelle (z.B. ein einfacher Schalter, ein Pulsgenerator oder der Ausgang 16 des optionalen A400 Moduls) verbunden sein

An Eingang "B" müssen 24 V anliegen, damit die Impulse Eingang „A“ erreichen (AND gate).

Beschreibung der einzelnen Schritte und Bedingungen (Beispiel 2)

- IST 10: Sequential Block.
Dieser Block umfasst in Graftec jeweils eine Sequenz von Schritten (Steps) und Konditionen (Transitions).
- ST 11: Leer, nur präsent für die Graftec Struktur.
- TR 51: Die Hauptaufgabe dieser Transition ist das erkennen der Bedingung zur weiteren Abarbeitung der Graftec Struktur. Diese Bedingung ist erfüllt, wenn der Input "StartMeasure" (I 0) hoch ist.
Wenn am ende der Transition der ACCU hoch ist, gilt das als Zeichen dafür, dass die Bedingung dieser Transition erfüllt ist und der nächste Schritt wird abgearbeitet bevor wieder zurück in den COB (Cyclical Organisation Block) gesprungen wird.
Wenn am Ende der Transition der ACCU tief ist wird das Programm direkt wieder zurück in den COB springen. Bei dem nächsten Aufruf des SB wird dieselbe Transition wieder abgearbeitet.
- ST 12: Der H100 Zähler wird mit dem Wert 3392 geladen (mit Hilfe des Registers H100.HelpRegister). Zudem wird der H100.Example2.RoundCounter mit dem Wert 4 geladen.
Der Ausgang CCO wird anschliessend eingeschaltet und der H100 Zähler wird gestartet.
- TR 52: Das H100 zählt nun automatisch abwärts. Sobald der Wert 0 erreicht ist, wird das Counter Flag vom Modul rückgesetzt. Das wird in der Graftec Struktur erkannt. In diesem Fall wird sofort das Counter Flag wieder gesetzt und der PCD Counter H100.Example2.RoundCounter wird dekrementiert.
Das Register H100.ReadCounterValueis wird mit dem aktuellen Wert des H100 Zählers aufgefrischt.
Die Bedingung für das Weiterfahren mit dem nächsten Schritt ist, dass der Wert des CPU Counters 0 erreicht hat.
- ST 13: Das "Enable CCO Reset" wird gesetzt. Aufgrund dessen wird der Ausgang CCO beim nächsten Errichen des Werts 0 vom H100 Zähler ausgeschaltet.
- TR 53: Das Register H100.ReadCounterValue wird mit dem aktuellen Wert des H100 Zählers aufgefrischt.
Die Bedingung für das Weiterspringen zum nächsten Schritt ist, dass das H100 Counter Flag den Wert 0 hat.



ST 14: Hier wird der Counter deaktiviert.

TR 54: Warten bis das Startsignal ausgeschaltet wird.

Anschliessend wird in den Schritt 11 gesprungen.

.

3. Kommandos für die Programmierung des H100

Für die Programmierung des PCD2/3.H100 Moduls sind die folgenden Funktionen verfügbar.

Kommandos für das Schreiben auf das Modul

- Zählrichtung (aufwärts/abwärts, nur im Modus SC)
- Starten und Stoppend des Zählers
- Setzen des Ausgangs CCO
- Setzen des Counter Flags
- Reset enable des CCO Ausgangs
- Laden des Zählers mit einem Wert von 0 bis 65 535

Kommandos zum Lesen des Moduls

- Lesen des Counter Flags
- Lesen des Zählerwertes

Diese Funktionen werden durch einzelne PCD Anweisungen (z.B. SET oder RES) oder deren Kombination ausgeführt. Instruktionen, welche eine Hardware Adresse ansteuern müssen mit dem Offset der Modul Basisadresse addiert werden.

3.1.1 Schreiben auf das Modul

- **Zählrichtung (Jumper in Position "SC")**

```

SET      O BA+13    ; abwärts zählen
RES      O BA+13    ; aufwärts zählen

```
- **Starten und Stoppen des Zählers**

```

SET      O BA+10    ; start Zähler
RES      O BA+10    ; stop Zähler

```
- **Setzen des Ausgangs CCO**

```

SET      O BA+14    ; Ausgang CCO wird direkt gesetzt
RES      O BA+14    ; der Ausgang CCO wird beim nächsten
                    ; Mal, wenn das Counter Flag
                    ; zurückgesetzt wird, ebenfalls
                    ; zurückgesetzt (Reset CCO Enable)

```
- **Setzen des Counter Flags**

```

STL      O BA+0     ; Lesen des Counter Flags und
SET      O BA+11    ; erneutes Setzen des Counter Flags
RES      O BA+11    ; wenn der Zähler Null erreicht.
(DEC     C y)       ; z.B. Dekrementieren des PCD Counters

```

- **Laden des Zählers mit einem Wert zwischen 0 und 65 535**

```

LD      R x      ; Lade PCD Register R 0-4'095/16'383
        Value    ; 0 ... 65'535
BITO    8        ; Lade untere 8 Bit
        R x
        O BA+0
SET     O BA+8    ; Lade anliegenden Wert ins Modul
RES     O BA+8
ROTR    R x
        8
ACC     H
BITO    8        ; Lade obere 8 Bit
        R x
        O BA+0
SET     O BA+9    ; Lade anliegenden Wert ins Modul
RES     O BA+9
  
```

Das Counter Flag is automatisch gesetzt wenn diese Routine ausgeführt wird.

3.1.2 Lesen vom Modul

- **Lese Counter Flag**

```

STL     O BA+0    ; ACCU wird 1 (= H),
                  ; wenn das Counter Flag 0 (= L) ist
  
```

- **Lese aktuellen Wert des Zählers *)**

```

SET     O BA+12    ; Für einen aktualisieren Wert
BITI    16         ; des Zählers muss diese Routine
        O BA+0    ; kontinuierlich ausgeführt werden,
        R z       ; z.B. einmal jede Sekunde.
RES     O BA+12    ;
  
```

*) Das Lesen des Zählers soll nur verwendet werden für Visualisierungszwecke, nicht aber für Vergleichsfunktionen.