

Lesen und Schreiben von anlagen Signalen in IL

Inhalt

1. ZUSAMMENFASSUNG	2
1.1 Funktionsbeschreibung	2
1.2 Möglicher Anwendungsbereich	2
1.3 Verwendete Hardware und Software	2
2. AUFBAU	3
2.1 Vorbereitung der PCD	3
2.1.1 Installation des Projekts	4
2.1.2 Anpassen der Hardware- und Software Settings im PG5	5
2.1.3 Anpassen des Codes für das vorliegende PCD System	6
2.1.4 Builden und Laden des Projektes in die PCD	6
2.2 Gelesene Werte Online beobachten	7
2.2.1 Das SAIA Watch Window	7
2.2.2 Der PG5 Online Debugger	8
3. FUNKTIONSBESCHREIBUNG UND EINSTELLUNGEN	9
3.1 Funktionsprinzip des Lesens I/O Modulen	9
3.1.1 Digitale I/O Module	9
3.1.2 Analoge I/O Module	10
3.1.3 Intelligente analoge I/O Module	11
3.2 Die Struktur der Beispielcodes	12
3.2.1 Symbolnamen Konvention	12
3.2.2 Einlesen von Werten einer Eingangskarte	13
3.2.3 Umrechnen des DV (Digital Value) in Benutzereinheit	13
3.2.4 4..20 mA Sensoren	13
3.2.5 Ausgeben von analogen Werten	14
4. FEHLERQUELLEN UND DEBUGGING	15
4.1 Häufige Fehler	15
4.2 Fehler suchen / Debugging Ansätze	15
4.3 Quellenverzeichnis	16

1. Zusammenfassung

1.1 Funktionsbeschreibung

Dieses Beispielprojekt soll zeigen, wie analoge Ein- und Ausgangssignale von verschiedenen Eingangskarten gelesen respektive geschrieben werden können. Für das Auslesen und Schreiben von analogen Werten mittels den Analogmodulen ist ein Grundlegendes Verständnis des Funktionsprinzips des I/O-Busses einer SAIA PCD Classic grundlegend. Dieses Thema wird in dem Kapitel 3.1 (Funktionsprinzip des Lesens von Analogen Modulen) behandelt.

Im Einzelnen werden folgende Module behandelt:

- PCD2/3.W1xx
- PCD2/3.W2xx
- PCD2/3.W3xx
- PCD2/3.W4xx
- PCD2/3.W5xx

In den meisten Beispielcodes wird jeweils ein Kanal des Moduls gelesen bzw. geschrieben (mit Ausnahme des Codes für die PCD2/3.W5xx Module). Der entsprechende Kanal ist in dem Register `Wx_Channel_to_read_1` spezifiziert. Um alle Kanäle zu lesen bzw. zu schreiben muss dieses Register in jedem Programmzyklus inkrementiert werden. Das ist in den vorliegenden Beispielen nicht ausprogrammiert, um die Modularität des Codes so hoch wie möglich zu lassen.

1.2 Möglicher Anwendungsbereich

Die in diesem Beispiel beschriebenen Anwendungen sind Basisanwendungen, welche in jedem Projekt mit analogen Ein- oder Ausgangssignalen verwendet werden können.

Die Beispiele sind jeweils direkt in einem COB (Cyclic Organisation Block) geschrieben. Um die Modularität des Programms zu erhöhen, können diese Codes natürlich auch in FB's integriert werden.

1.3 Verwendete Hardware und Software

Hardware:

- PCD3.M5540
- Je nach Aufgabe eines der oben aufgelisteten analogen I/O Module
- Programmierkabel PCD8.K111 oder USB Kabel (für PCD oder PCD2.M480)
- Bei Verwendung von Eingangsmodulen ist ein Signalgeber notwendig.

Für den Einsatz der Beispiele mit einer Steuerung PCD1.Mxxx, PCD2Mxxx oder PCD3.Mxxx sind die Hardware Settings des Projekts entsprechend anzupassen. Es ist möglich, die Konfiguration der bestehenden Steuerung direkt in das Projekt zu laden, indem die „Upload“ Funktion der Hardware Settings verwendet wird (Siehe Kapitel 2.1.2, Anpassen der Hardware- und Software Settings im PG5)

Minimale Softwareversion:

SAIA PG5 1.3.120

2. Aufbau

2.1 Vorbereitung der PCD

Das zu verwendende Modul kann auf einem beliebigen Steckplatz in die PCD3.M5540 eingefügt werden. Wenn ein PCD3.S100 (mit dem PCD3 Demo Modell) verwendet wird, so sind die Basisadressen des PCD3.W200 sowie PCD3.W400 durch das PCD3.S100 definiert.

In der dem Modul entsprechenden AWL-Datei bzw. dem zugehörigen Symbol-Editor muss die Basisadresse des Moduls eingetragen werden (standardmässig eingetragen ist die Adresse 16 → zweiter Steckplatz).

Die Module werden gemäss dem Manual „PCD1-PCD2“, 26/737 oder „PCD3“, 26/789 verdrahtet.

Die Beispiele in diesem Projekt sind für die Module PCD2/3.Wx00 ausgelegt. Diese Module lesen einen Spannungseingang von 0..10V (oder geben eine Spannung aus). Der gelesene bez. geschriebene Wert wird in der Benutzereinheit (User unit) von mV angezeigt.

Wenn ein anderes Signal gelesen wird, so muss lediglich das entsprechende Symbol (Wx_Measurement_Range_1), welches den maximalen Wert der Benutzereinheit definiert, im Symboleditor entsprechend angepasst werden. Mehr dazu im Kapitel 3.2.3 (Umrechnen des DV in die reale Grösse).



Besonders die analogen Eingangsmodule sind auf Massenschlaufen empfindlich. Bitte beachten Sie deshalb unbedingt die entsprechenden Verdrahtungsvorschriften!

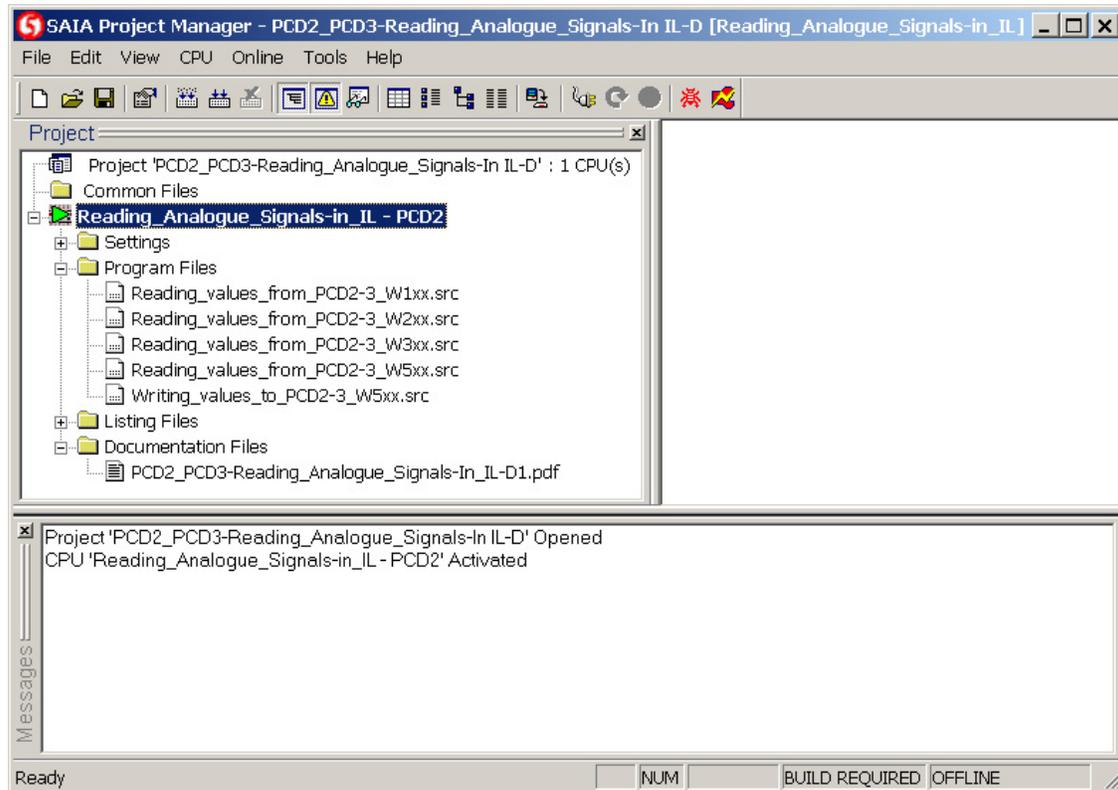


Der I/O Bus von SAIA PCD Classic Steuerungen ist nicht für „Hot-Plugging“ konzipiert. Für das Einschieben oder Entfernen von I/O Modulen muss die Steuerung spannungsfrei gemacht werden!

2.1.1 Installation des Projekts

Zur Installation des Projektes in Ihrem PG5 Project Verzeichnis ist die Verwendung der "Restore..." Funktion aus dem Menü „File“ des PG5 1.3 Project Managers vorgesehen. Diese Funktion wird das Projekt in Ihr Projektverzeichnis kopieren.

Bei Verwendung des PG5 1.1.x muss das Projekt manuell entzippt werden und in das PG5 Projekt Verzeichnis kopiert werden.



Dieses Dokument ist in dem „Documents“ Ordner im Project Tree des PG5 Project Managers zu finden und kann von dort aus direkt durch Doppelklicken geöffnet werden.

2.1.2 Anpassen der Hardware- und Software Settings im PG5

Als erster Schritt ist der PG5 Project Manager (PG5 SPM) zu starten.

Anschliessend wird folgendes Vorgehen ausgeführt:

- PCD mit PGU Programmierkabel (PCD8.K111) oder USB Kabel mit dem PC verbinden und einschalten.
- „Hardware Settings“ Fenster öffnen (im Ordner Settings im „Project Tree“ des Project Managers).
- Button „Upload“ betätigen um die Hardware Konfiguration der PCD auf den PC zu laden. Anschliessend mit „Ok“ speichern.
- „Software Settings“ Fenster öffnen (im Ordner Settings im Project Tree des Project Managers).
- Den dynamischen Bereich der Ressourcen durch ein Klicken auf den Button „Set Default“ anpassen und mit „Ok“ bestätigen.
- Linken der Fupla Datei, die verwendet werden soll (z.B. Reading_values_from_PCD2-3_W2xx.src für das Lesen von Werten eines PCD2/3.W2xx Eingansmoduls).

Mit der rechten Maustaste auf das gewünschte Fupla Programm im „Project Tree“ klicken und „Linked“ anwählen.

2.1.3 Anpassen des Codes für das vorliegende PCD System

Damit das richtige Modul ausgelesen wird und die Benutzereinheiten korrekt eingestellt sind, sind folgende Einstellungen vorzunehmen:

- Die Basisadresse des Steckplatzes, auf dem das Modul eingesetzt ist, muss definiert werden (W3_BaseAddress_1). Die Basisadresse ist ein Vielfaches von 16 und kann auf einer PCD2 direkt von dem Hauptprint abgelesen werden.
Bei einer PCD3 sind nur die Steckplätze nummeriert (beginnend mit 0). Die Basisadresse des I/O Busses kann hier durch die Multiplikation mit 16 (Modul Basis Adresse = $16 * (\text{Steckplatzadresse} + 1)$) errechnet werden.
Beachten Sie, dass die Steckplatzadresse auf einem Erweiterungsgehäuse die Fortsetzung der Adressen auf dem vorherigen Gehäuse darstellt.
- Die Benutzereinheit kann optional angepasst werden. Bei den Beispielen werden jeweils Eingangssignale von 0..10V gelesen. Die gelesenen Werte werden standardmässig in mV ausgedrückt.

2.1.4 Builden und Laden des Projektes in die PCD

Nach den Anpassungen des vorherigen Kapitels kann das Projekt nach einem „Rebuild All“ (Menü „CPU“, Option „Rebuild All...“ oder Alt+F2) in die Steuerung geladen werden.

Sollte sich die Steuerung bereits im Zustand „Run“ befunden haben, so wird nachgefragt, ob die Steuerung gestoppt werden darf. Dies ist während dem Testaufbau der Fall. Die Meldung wird aus Sicherheitsgründen angezeigt, da auf einer bestehenden Anlage im Einsatz die Steuerung unter Umständen nicht gestoppt werden darf.

Ein Vorteil der AWL-Programmierung liegt darin, dass Programme auch während der Laufzeit der PCD geladen werden können (Download in RUN). Voraussetzung dafür ist, dass alle Programmteile in AWL geschrieben sind.

2.2 Gelesene Werte Online beobachten

Sobald das Programm in die Steuerung geladen ist, kann man eine Onlineverbindung mit der Steuerung aufnehmen, um die Werte online zu beobachten: Durch ein Klicken auf den Online Button (mit dem Stecker) verbindet sich der PC mit der PCD. Sollte sich die PCD noch nicht in Run befinden, so kann sie mit dem grünen runden Pfeil in dem Toolbar gestartet werden.

Mittels dem Watch Window (siehe folgendes Kapitel) können nun die gelesenen Werte online beobachtet werden.

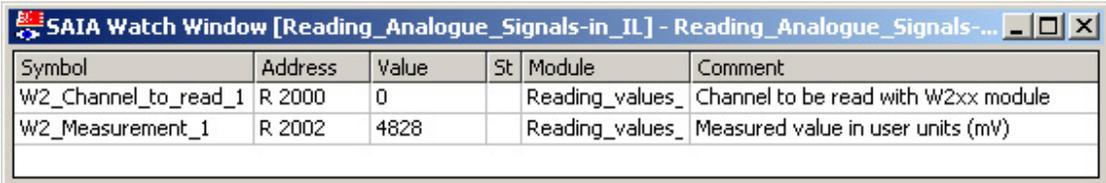


Theoretisch wäre es auch möglich, den Code in dem IL Editor online zu beobachten. Da die analogen I/O Module aber auf den Zugriff auf den I/O Bus reagieren, muss darauf verzichtet werden!

2.2.1 Das SAIA Watch Window

Um die Werte von Medias in einem Fenster anzuzeigen und zu verändern, kann das Watch Window verwendet werden. Zu öffnen ist das Watch Window unter dem Menu „View“ des PG5 Project Manager. Die anzuzeigenden Symbole können durch Drag and Drop in das Fenster gezogen werden.

Sobald das PG5 online ist, werden die entsprechenden Werte angezeigt werden.



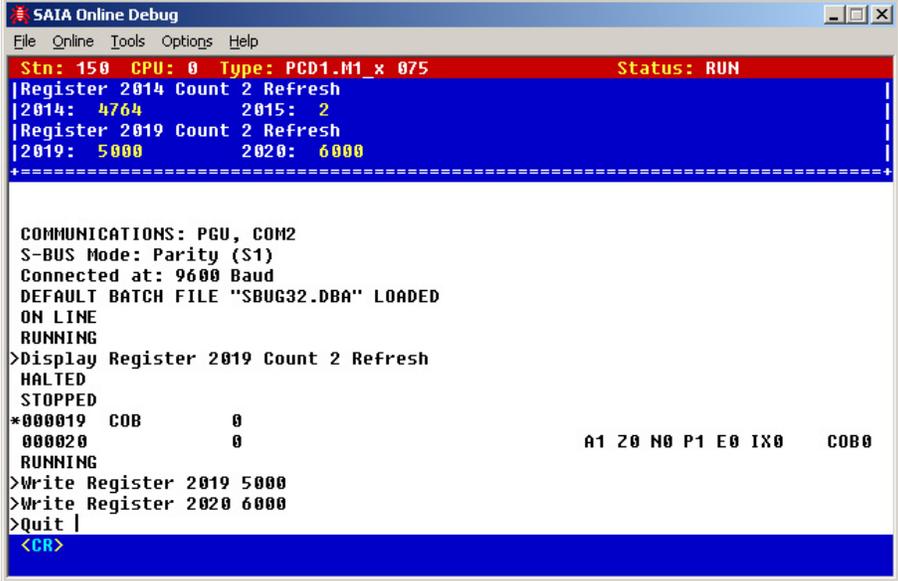
Symbol	Address	Value	St	Module	Comment
W2_Channel_to_read_1	R 2000	0		Reading_values_	Channel to be read with W2xx module
W2_Measurement_1	R 2002	4828		Reading_values_	Measured value in user units (mV)



Auch hier gilt, dass keine binären I/Os der Analogmodule auf dem I/O Bus visualisiert werden dürfen! Diese Visualisierung würde das Verhalten der analogen I/O Module beeinflussen.

2.2.2 Der PG5 Online Debugger

Eine weitere Möglichkeit, Werte online anzuzeigen, ist der Online Debugger (PG5 Project Manager im Menü „Tools“, „Online Debug“ oder Taste F11).



```

SAIA Online Debug
File Online Tools Options Help
Stn: 150 CPU: 0 Type: PCD1.M1 x 075 Status: RUN
|Register 2014 Count 2 Refresh
|2014: 4764 2015: 2
|Register 2019 Count 2 Refresh
|2019: 5000 2020: 6000
=====
COMMUNICATIONS: PGU, COM2
S-BUS Mode: Parity (S1)
Connected at: 9600 Baud
DEFAULT BATCH FILE "SBUG32.DBA" LOADED
ON LINE
RUNNING
>Display Register 2019 Count 2 Refresh
HALTED
STOPPED
*000019 COB 0
000020 0 A1 Z0 N0 P1 E0 IX0 COB0
RUNNING
>Write Register 2019 5000
>Write Register 2020 6000
>Quit |
<CR>
  
```

Der Online Debugger ist ein sehr vielseitiges Tool, das unter anderem folgende Möglichkeiten bietet:

- Steuern der PCD (Stoppen, Starten etc.)
- Anzeigen von PCD Medias (Register, Flags, DB's etc. auch aufgefrischt)
- Schreiben von PCD Medias
- Modifizieren des Programms (wenn es in RAM gespeichert ist)
- Anzeigen des CPU Status (HW- und FW Version etc.)
- Auslesen der PCD History
- Anzeigen der aktuellen HW Konfiguration
- Programm „tracen“ (Step by Step abarbeiten lassen; nicht erlaubt beim Zugriff auf analoge E/As!)
- Abarbeiten lassen einzelner Instruktionen
- Suchen von Instruktionen
- Laufen lassen des Programms bis zu einem Event (z.B. der Veränderung eines bestimmten Medias oder dem gesetzten Error Status Flag)

Das Anzeigen der gelesenen Werte wird in dem Online Debugger durch folgende Eingabe erreicht (nur die fett gedruckten Zeichen eingeben):

> **Display Register** <Adresse des anzuzeigendes Register> Refresh <Enter>

Die Adresse des anzuzeigende Registers ist z.B. in dem „Data List View“ des PG5 Project Managers ersichtlich (Unter Menu „View“ „Data List“ wählen).

3. Funktionsbeschreibung und Einstellungen

3.1 Funktionsprinzip des Lesens I/O Modulen

Die SAIA PCD Classic Controller haben einen internen I/O Bus mit statischer Adressierung. Jedem Modulsteckplatz sind 16 binäre I/O Punkte zugeordnet, die in aufsteigender Reihenfolge adressiert sind. Daraus resultiert, dass dem auf dem ersten Steckplatz eingesteckten Modul die absoluten I/O Adressen 0 bis 15 zugeteilt sind. Jede I/O Bus Adresse kann gelesen und geschrieben werden.

Der Zugriff auf die I/O Punkte erfolgt **sofort** bei dem Lesen oder Schreiben durch das Benutzerprogramm.



Die Steuerungen der Familie SAIA PCD Classic arbeiten nicht mit einem Prozessabbild!

Dadurch kann ein I/O während einem Programmzyklus mehrfach gelesen und/oder geschrieben werden.

Wenn eine Adresse gelesen wird, jedoch kein oder ein defektes Eingangsmodul gesteckt ist, so wird der Wert 0 gelesen.

3.1.1 Digitale I/O Module

Die Datenpunkte von digitalen I/O Modulen sind direkt adressiert.

Wenn in dem Benutzerprogramm z.B. der Ausgang 0 gesetzt wird, so wird durch ein auf dem ersten Steckplatz eingeschobenen digitalen Ausgangsmodul der erste Ausgang eingeschaltet.

Wenn ein digitales Eingangsmodul gesteckt ist, so wird das beim Lesen der Zustand des physikalischen Eingangs zurückgegeben.

Bei digitalen I/O Modulen, welche weniger als 16 Ein- bzw. Ausgänge haben, bleiben die verbleibenden Adressen auf dem I/O Bus ungenutzt und können nicht von einem andern Modul verwendet werden.

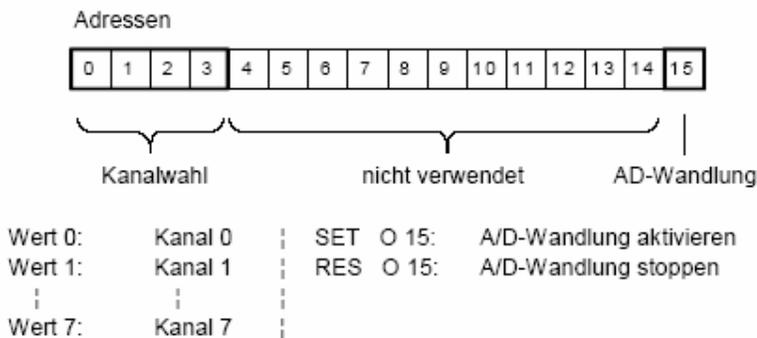
3.1.2 Analoge I/O Module

Da auch die Analogen I/O Module nur über die oben beschriebenen 16 I/O Adressen verfügen, müssen die zu lesenden/schreibenden Werte (8..12 Bit Werte) aller Kanäle „multiplext“ werden. Zudem ist eine Möglichkeit, z.B. eine Analog-digital Wandlung zu initiieren, erforderlich. Dazu sind je nach Modul spezifische digitale Adressen auf dem (dem Modul zugeteilten) I/O Bus reserviert.

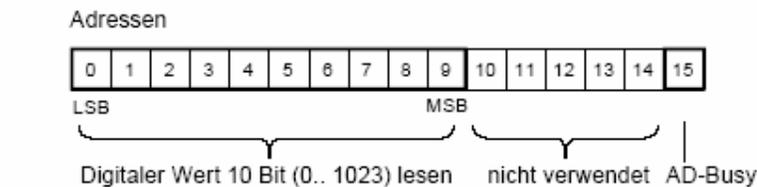
Folgend die Bedeutung der Adressen eines PCD2/3.W2xx Moduls. Das Schreiben bezieht sich auf das Wählen des zu lesenden Kanals. Anschliessend kann gemäss der Aufteilung für das Lesen der gelesene Wert vom I/O Bus gelesen werden.

Bedeutung der 16 Adressen

- **Schreiben:**



- **Lesen:**



AD-Busy = 1 : A/D-Wandlung läuft
 AD-Busy = 0 : A/D-Wandlung beendet

Die analogen Eingangsmodule der SAIA PCD Classic Controller werden prinzipiell in folgenden Schritten behandelt:

- Wählen des zu lesenden/schreibenden Kanals auf dem Modul
- Aktivierung der Analog-digital Wandlung
- (Abwarten bis Wandlung ausgeführt wurde)
- Lesen des Resultats der Analog-digital Wandlung über den I/O Bus
- Umrechnen des digitalen Wertes in benutzerfreundliche Einheiten

Unterschiede zwischen den verschiedenen analogen I/O Modulen

Die verschiedenen Modulfamilien (W1xx, W2xx etc.) weisen Unterschiede in der Handhabung auf. So sind z.B.

- die Auflösung der Module unterschiedlich (8, 10 oder 12 Bit)
- das Signal zur Aktivierung der Wandlung nicht immer auf der selben Adresse
- Die Wandlungszeit des Analog-digital Wandlers unterschiedlich.

Deshalb sind die Module den Anforderungen entsprechend anzusprechen.

Speziell zu erwähnen ist hierbei das Modul PCDx.W1xx. Der Analog-digital Wandler dieses Moduls benötigt wesentlich mehr Zeit zur Wandlung als die auf den restlichen Modulen eingesetzten Wandler. Da ein Warten auf die Wandlung die Abarbeitung des Programms wesentlich verzögern würde, wird in einem Programmzyklus die Wandlung ausgelöst und erst in einem späteren Zyklus das Resultat dieser Wandlung eingelesen.

3.1.3 Intelligente analoge I/O Module

Ein weiterer Fall sind die **intelligenten analogen Module (W3x5, W6x5, W7xx)**, welche bereits auf dem Modul mittels eines Prozessors den gelesenen digitalen Wert verarbeiten. Diese Module müssen entsprechend der Anwendung konfiguriert werden, was ebenfalls über den I/O Bus geschieht.

Da die Konfigurierung komplizierter ist als die Handhabung von nicht intelligenten Modulen, werden für diesen Fall FB's eingesetzt.

Die Programmierung der intelligenten analogen Module wird nicht in diesem Beispiel behandelt.

3.2 Die Struktur der Beispielcodes

Die in diesem Beispielprojekt vorhandenen Codes sind immer nach demselben Prinzip aufgebaut:

- **Dokumentkopf**
Enthält eine Kurzbeschreibung, Erstellungsdatum und Autor sowie die History des Dokuments.
- **Codesegment für das Auslesen des Analogmoduls**
Wird durch die Direktive \$COBSEG 0 in den COB 0 (erster „Cyclic Organisation Block“) eingefügt.
- **Umrechnung von dem digitalen Wert vom Analog-digital Wandler in Benutzerdefinierte Einheiten.** Dieser Teil wird ebenfalls in den COB 0 geschrieben.
- **Initialisierung von Medias**
Um zu verhindern, dass gleich nach dem Start der PCD Zufallswerte in wichtigen Registern enthalten sind.
Diese Initialisierung wird durch die Direktive \$INIT in den XOB 16 (Cold Start Block) geschrieben, welcher immer als erster Programmblock abgearbeitet wird (nach einem Kaltstart).

Die verschiedenen Dateien unterscheiden sich hauptsächlich durch unterschiedliche Routinen für das Auslesen der Module. Die Kommentare in diesen Dateien beschreiben die genauen Operationen, die ausgeführt werden.

Für Hilfe bezüglich der verwendeten Instruktionen kann die entsprechende Instruktion angewählt werden und mit F1 die Online Hilfe des IL-Editors aufgerufen werden.

3.2.1 Symbolnamen Konvention

Die in diesen Beispielen verwendeten Symbolnamen haben folgende Struktur:

W2_Measurement_Range_1

- Das „W2“ zu Beginn des Namens weist auf den auszulesenden Modultypen hin.
- Der Hauptteil des Namens entspricht der Bedeutung des Symbols
“Measurement_Range“ → Messbereich des Eingangswertes
- Das _1 am Ende des Namens ist der Index des Moduls (das zweite W2 Modul auf der PCD bekäme in diesem Sinne den Index _2 usw.)

3.2.2 Einlesen von Werten einer Eingangskarte

Wie in dem Kapitel 3.1.2 erklärt, wird jeweils als Erstes die Kanalwahl ausgeführt. Dazu werden die entsprechenden Bits auf den I/O Bus gesetzt. Meistens wird dazu die Instruktion BITO verwendet. BITO schreibt eine definierte Anzahl Bits eines Registers (beginnend mit dem LSB) auf digitale PCD Medias, in diesem Falle Ausgänge.

Anschliessend wird die der Analog-digital Wandler aktiviert. In den meisten Fällen ist das das letzte Bit des I/O Bereiches. Je nach Modul wird dieses Bit sofort oder erst nach dem Lesen des Digitalen Wertes (DV) wieder auf 0 gesetzt.

Nach beendeter Wandlung des Wertes wird der Gelesene Wert on dem I/O Bus gelesen. Das wird meistens mit der BITIR Instruktion realisiert. Diese Instruktion liest die binären Datenpunkte von dem Bus und schreibt sie in ein Register, welches anschliessend den DV (Digital Value) enthält.

Wichtig ist, dass das geschriebene Register in den von der Instruktion BITIR nicht beschriebenen Bits den Wert 0 enthält. Andernfalls würde anschliessend ein falscher (und viel zu hoher) Wert resultieren.

3.2.3 Umrechnen des DV (Digital Value) in Benutzereinheit

Da es sich bei dem Ausgabewert der Analogmodule jeweils um den digitalen Wert handelt, muss dieser anschliessend in die vom Programmierer benötigte Grösse umgerechnet werden. Dies geschieht nach der allgemeinen Formel:

$$\text{Messwert} = \frac{\text{DigitalerWert} * \text{MaximalerMesswert}}{\text{AuflösungdesModuls}}$$

Die Einheiten dieser Formel sind die folgenden:

Messwert in benutzerdefinierter Einheit (in den Beispielen: mV)

„AuflösungdesModuls“ und „DigitalerWert“ sind ohne Einheit

„MaximalerMesswert“ ist wie der Messwert selbst in mV.

3.2.4 4..20 mA Sensoren

Eine zusätzliche Umrechnung ist notwendig bei Signalgebern von 4..20mA. Ein entsprechendes Programm enthält die Datei „Current_4-20mA.src“. Basis für diese Umrechnung ist ein Messwertbereich von 0..10000 User Units (Standardresultat der restlichen Beispielcodes).

Das Resultat wird ausgedrückt in 0..1000 $\frac{1}{10}\%$.

3.2.5 Ausgeben von analogen Werten

Das Ausgeben von analogen Werten auf die I/O Module wird nach demselben Prinzip durchgeführt wie das Lesen. Der Unterschied besteht darin, dass die Werte in diesem Fall auf den Bus geschrieben und nicht gelesen werden.

Zudem muss in dem Programm die Umwandlung des Wertes in den Benutzereinheiten in den DV vor dem Schreiben erfolgen.

Diese Umrechnung wird nach der allgemeinen Formel:

$$\text{DigitalerWert} = \frac{\text{Ausgabewert} * \text{AuflösungdesModuls}}{\text{MaximalerAusgabewert}}$$

berechnet.

Die Einheiten dieser Formel sind die folgenden:

Ausgabewert in benutzerdefinierter Einheit (in den Beispielen: mV)

„AuflösungdesModuls“ und „DigitalerWert“ sind ohne Einheit

„MaximalerAusgabewert“ ist wie der Ausgabewert selbst in mV.

4. Fehlerquellen und Debugging

Um Fehler rasch einzukreisen und zu beheben sind in diesem Kapitel einige häufig auftretende Fehlerbilder beschrieben.

4.1 Häufige Fehler

Hier eine Auflistung von häufigen Ursachen für eine Fehlfunktion des beschriebenen Beispiels:

Fehlerbild	Ursache und Beheben des Fehlers
Der eingelesene Wert eines Eingangsmoduls entspricht nicht der gewünschten Einheit	Wahrscheinlich ist der maximalwert der Benutzerdefinierten Einheit nicht korrekt definiert. Überprüfen Sie den Wert der Konstante Wx_Measurement_Range_x
Der analoge Ausgangswert ist immer =0 oder maximaler Ausschlag	Ev. ist eine fehlerhafte Verdrahtung des Moduls die Ursache. Bitte überprüfen Sie die Verdrahtung gemäss dem Hardware Handbuch Ihrer PCD.
Der analoge Eingangswert ist immer =0	
Das Modul funktioniert nicht richtig und gibt keine oder falsche Werte aus.	Eine mögliche Ursache ist ein Zugriff auf I/O Adressen des Moduls durch den Online Debugger oder das Watch Window. Stellen Sie sicher, dass die Bus Adressen des Moduls nicht online angezeigt werden!
Der gelesene Wert einer oder mehrerer analoger Eingänge springt zeitweise und ist nicht konstant.	Die Ursache für dieses Phänomen könnte ein „Massenloop“ im System sein. Bitte überprüfen Sie das Massenkonzept ihres Systems. Die Masse am „-“ des Moduls muss <u>kurz und massiv</u> mit der „-“-Klemme der PCD verbunden sein (kein Weiterlaufen der Masse um die PCD!)

4.2 Fehler suchen / Debugging Ansätze

Bei der Fehlersuche ist es empfohlen, mit einer Grundlegenden Funktion zu beginnen und schrittweise weitere Funktionen zu testen. So ist es z.B. sinnvoll, als erstes ein kleines Testprogramm für das Lesen von Eingangswerten zu schreiben.

Um sicher zu sein, dass auch ein Signal am Eingang ansteht, sollte bei der Inbetriebnahme das Eingangssignal mit einem Multimeter verifiziert werden.

Die Werte in der PCD können mittels dem Watch Window oder mittels dem Online Debugger verifiziert werden.

4.3 Quellenverzeichnis

Die verschiedenen Prozeduren für das Handhaben der Module ist spezifisch auf die Hardware der entsprechenden Module angepasst.

Die Beschreibungen für die AWL Instruktionen können in der Online Help des IL-Editors gefunden werden.

Hardwarespezifische Daten wie Klemmenbelegungen und Verdrahtungsschemen sind im jeweiligen Hardware Handbuch der entsprechenden Steuerung zu finden.