

UE 183.169 Dezentrale Automation

Dokumentation zur ASi Library

Michael Pimminger
Matr. Nr.: 9926986, Kenz.: 535

Wien, 4. April 2007

Inhaltsverzeichnis

| | | |
|----------|--|----------|
| 1 | Allgemeines | 2 |
| 2 | Einbinden der Library | 2 |
| 3 | Anwendung | 2 |
| 3.1 | Library-Bausteine | 2 |
| 3.2 | Ablauf | 3 |
| 4 | Speicherbereiche der Library | 4 |
| 4.1 | Übergabespeicherbereich | 4 |
| 4.2 | Library-Interner Speicherbereich | 6 |
| 5 | Funktion der Library | 8 |
| 5.1 | Initialisierung | 8 |
| 5.2 | Daten lesen | 8 |
| 5.3 | Daten schreiben | 8 |

1 Allgemeines

Dieses Dokument beschreibt die Library für den Zugriff auf ASi Slaves der (kleinen) Übungstargets im TILAB. Für diese Targetsysteme liegt eine detaillierte Beschreibung (siehe [4]) vor.

Als ASi -Master dient das CP 242-8 Modul, welches direkt an die SPS (S7-214) angeschlossen ist. Im Umgang mit den beiden genannten Komponenten sei auf deren Dokumentationen ([1] für SPS und [2] für das ASi-Master bzw. Profibus-Slave Modul) hingewiesen.

Die vorliegende Library soll das Initialisieren der CP 242-8 sowie die Kommunikation mit den ASi Slaves über das Modul erleichtern. Sie soll ein Werkzeug zum Handhaben der Kommunikation, insbesondere für das empfindliche Analogprotokoll sein. Eine gute Beschreibung dieses Protokolls findet sich im Handbuch *AS-Interface Analogmodule* [3] Seite 10-10 ff.

In den Kapitel 2 und 3 folgt eine Erklärung wie die Library in das SPS-Programm einzubinden ist. Kapitel 4 zeigt die Nutzung des Speichers. Die weiteren Kapitel dokumentieren die Funktionsweise und die Implementierung der Library.

2 Einbinden der Library

Die Library liegt in Form der Datei *tilab_asilib.mwl* vor. In der Programmierumgebung Step7-Micro/Win kann sie über das Kontextmenü eingebunden werden. Dazu einen rechten Mausklick auf *Bibliotheken* tätigen und *Bibliotheken ergänzen/entfernen...* auswählen. Mit einem Klick auf den Button *Ergänzen...* den Auswahldialog öffnen. Die Datei *tilab_asilib.mwl* im entsprechenden Pfad wählen und auf *Speichern* klicken. Den Vorgang mit *OK* beenden.

Vor dem ersten kompilieren des Programmes muss der Library ein Speicherbereich zugewiesen werden. Das geschieht durch den Menüpunkt *Datei* \mapsto *Speicher für Bibliothek...*. Achtung! Dieser Menüpunkt ist erst sichtbar wenn mindestens ein Bibliotheksbaustein auch im Programm verwendet wurde. Es muss eine Anfangsadresse (z.B. VB900) angegeben werden. Die folgenden 30 Byte werden dann der Bibliothek als interner Speicher zugewiesen.

3 Anwendung

3.1 Library-Bausteine

Die Library enthält 3 Bausteine:

- *ASi_init* initialisiert das CP 242-8 Modul als ASi Master für das Target.
- *ASi_read* dient zur Datenübermittlung der Slaves in Richtung Master.
- *ASi_write* dient zur Datenübermittlung des Masters in Richtung Slaves.

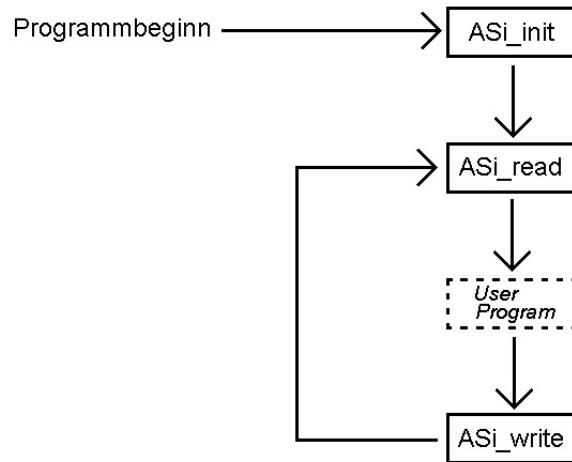


Abbildung 1: Ablaufdiagramm

Dabei ist zu beachten, dass die Bausteine *ASi_read* und *ASi_write* immer erst nach dem Aufruf von *ASi_init* im Programm vorkommen dürfen. Generell sollten die Aufrufe von *ASi_read* und *ASi_write* auch nur einmal pro Zyklus im SPS Programm vorkommen, da sonst Daten überschrieben werden, welche noch nicht an die ASi-Slaves übermittelt wurden. Dies kann zur Verletzung des überaus empfindlichen Analogprotokolls, und in Folge zu einem „hang up“ des Programms bzw. einem undefinierten Programmmzustand führen.

3.2 Ablauf

1. Initialisierung (nur im ersten Zyklus)
2. Daten von den Slaves lesen
3. Daten manipulieren (User Programm)
4. Daten an die Slaves ausgeben

Zu Beginn des SPS-Programms, spätestens vor dem ersten Aufruf der Bibliotheks-Bausteine *ASi_read* oder *ASi_write* ist das CP 242-8 Modul zu initialisieren. Diese Initialisierung darf nur im allerersten Zyklus des SPS-Programms (SM0.1 siehe [1] Seite 30) vorgenommen werden.

Danach kann mit *ASi_read* die Datenübertragung der Slaves in Richtung Master angestoßen werden. Die Daten aller digitalen Slaves werden hiermit in den dafür vorgesehenen Speicherbereich gestellt. Steht im aktuellen Zyklus auch ein neuer gültiger Wert eines oder mehrerer Analog-Slaves zur Verfügung, so wird dieser ebenfalls in den dafür vorgesehenen Speicherbereich geschrieben. Die Übermittlung eines Analog-Wertes mit Hilfe des ASi Analogprotokolls dauert mindestens 7 Zyklen. Im besten Fall werden also Digitale Input-Daten jeden Zyklus und Analoge Input-Daten jeden 7 Zyklus aktualisiert.

Nun können die Daten der Slaves aus den vorgesehenen Speicher gelesen, sowie Daten an die Slaves in den jeweilig dafür vorgesehenen Speicher geschrieben werden.

Am Ende des SPS-Programms, frühestens nach der letzten Manipulation im Speicherbereich der ASi Slaves, wird mit dem Bibliotheksbaustein *ASi_write* die Datenübertragung vom Master in Richtung Slaves angestoßen. Damit werden die Daten für Digitale ASi Slaves unmittelbar an diese gesandt. Für die Datenübertragung an analoge ASi Slaves ist wiederum das ASi-Analogprotokoll erforderlich. Die Übermittlung eines Analogwertes dauert ebenfalls mindestens 7 Zyklen. Es gilt hier also die gleiche Regel: Digitale Output-Daten werden im besten Fall jeden Zyklus, analoge Output-Daten werden im besten Fall jeden 7. Zyklus vollständig übertragen.

Durch die zyklische Abarbeitung eines SPS-Programms beginnt nun das Spiel mit dem Lesen der Slave-Daten von neuem.

4 Speicherbereiche der Library

Die Library benutzt im folgenden zwei Speicherbereiche des Variablenspeichers der SPS.

- Übergabespeicher, in dem die ASi-Daten dem Benutzer übergeben werden. Dieser ist statisch vergeben und belegt den Bereich von VB0 bis VB16.
- Library-Interner Speicherbereich. Der interne Speicherbereich braucht 30 Byte. Die Adressen werden durch die Funktion *Speicher für Bibliothek...* festgelegt.

Der Library-Interne Speicherbereich sollte vom Benutzer nicht manuell verändert werden, da es sonst zu Komplikationen im ASi Analogprotokoll und nachfolgend zu einem „hang up“ des Programms bzw. einem undefinierten Programmzustand kommen kann!

4.1 Übergabespeicherbereich

Der Übergabebereich ist in folgender Symboltabelle beschrieben. Aus diesem Speicherbereich werden die Daten der ASi Slaves vom User entnommen bzw. Daten an die ASi Slaves vom User gespeichert.

ASi Digital Slaves Eingänge:

| Symbol | Adresse | Kommentar |
|-------------|---------|--|
| Tast_3 | V0.3 | Taster 3 rot |
| Tast_2 | V0.2 | Taster 2 weiß |
| Tast_1 | V0.1 | Taster 1 grün |
| Tast_0 | V0.0 | Taster 0 gelb |
| EinLS | V0.4 | Sicherheitslichtschranke (0=nicht unterbrochen) |
| KapNS | V0.5 | kapazitver Näherungsschalter (0=Werkstück weg) |
| OptNS | V0.6 | optischer Näherungsschalter (0=Werkstück weg) |
| GlasLS | V0.7 | Glasfaser-Lichtschranke (0=nicht unterbrochen) |
| IndNS | V1.0 | induktiver Näherungsschalter (0=Werkstück weg) |
| IndNS_valid | V1.1 | Valid (1=Gültig) |
| IndNS_fail | V1.2 | Fail (0=Fehlerfall) |
| ZEndDo | V1.3 | Zylinder Endschalter unten (0=nicht unten) |
| ZEndUp | V1.4 | Zylinder Endschalter oben (0=nicht oben) |
| MagE_D | V1.5 | Magnetischer Endschalter unten (0=Schlitten weg) |
| MagE_U | V1.6 | Magnetischer Endschalter oben (0=Schlitten weg) |
| Sonar0 | V2.0 | Sonar 55-125mm |
| Sonar1 | V2.1 | Sonar 125-220mm |
| Sonar2 | V2.2 | Sonar 220-300mm |

ASi Digital Slaves Ausgänge:

| Symbol | Adresse | Kommentar |
|---------|---------|--|
| Lampe_3 | V3.3 | Lampe 3 rot |
| Lampe_2 | V3.2 | Lampe 2 weiß |
| Lampe_1 | V3.1 | Lampe 1 grün |
| Lampe_0 | V3.0 | Lampe 0 gelb |
| McGo | V3.4 | Schlittenmotor (0=aus 1=ein) |
| McDir | V3.5 | Schlittenmotor Fahrtrichtung (0=up 1=down) |
| ZylDo | V3.6 | Zylinder absenken (0=nicht absenken) |
| ZylUp | V3.7 | Zylinder anhaben (0=nicht anhaben) |

ASi Analog Slaves Ein- und Ausgänge:

| Symbol | Adresse | Kommentar |
|----------|---------|---|
| Joy_LR | VW4 | Joystick Left/Right |
| Joy_LR_S | V2.7 | Joystick Left/Right Sign Bit (0=Rechts 1=Links) |
| Joy_UD | VW6 | Joystick Up/Down |
| Joy_UD_S | V6.7 | Joystick Up/Down Sign Bit (0=Up 1=Down) |
| Dist | VW8 | Distanzsensor Schlittenposition |
| Druck | VW10 | Drucksensor |
| DVM | VW12 | Anzeige Digitalvoltmeter |
| McSpeed | VW14 | Geschwindigkeit Schlittenmotor |

Diese Auflistung kann unverändert als Symboltabelle in die Step7 Programmierungsumgebung übernommen werden. Symbolnamen stimmen mit der Dokumentation des Targetsystems ([4]) überein. Die Symboltabelle liegt unter dem Namen *asilib-symbols_v2.ods* als Open-Office-Calc Datei vor. Sie kann via Copy and Paste direkt in die Step7-Micro/Win Symboltabelle eingefügt werden. Es kann auch die Symboltabelle des Demoprogramms kopiert werden.

Die von Druck- und Distanzsensor empfangenen Analogwerte werden ohne SVO-Trippel in den Übergabespeicher gemappt. Bei Analogwerten vom Joystick ist das Sign-Bit im MSB des Datenwortes zu finden (Bit 7). Das Valid- und das Overflow-Bit werden verworfen, da der ganze Wert nur dann im Übergabespeicher abgelegt wird, wenn er gültig ist. Ist ein Wert nach Durchlauf des Übertragungsprotokolls als ungültig gekennzeichnet, so wird er nicht im Benutzerspeicherbereich abgelegt. Der alte Wert bleibt erhalten bis ein neuer gültiger Wert vorhanden ist. Die Fehlererkennung wird somit von der Library-Routine durchgeführt.

Druck- und Distanzwert:

| | | | | | | | | | | | | | | | | |
|------|----------------|---|---|---|--------------|-----|-----|----|-----------------|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VWx: | | | | | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 |
| | unbenutzt | | | | 12 Bit Daten | | | | | | | | | | | |
| | VBx (Highbyte) | | | | | | | | VBx+1 (Lowbyte) | | | | | | | |

Joystick-Werte:

| | | | | | | | | | | | | | | | | |
|------|----------------|-----------|---|---|--------------|-----|-----|----|-----------------|----|----|----|----|----|----|----|
| VWx: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | S | | | | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 |
| | ± | unbenutzt | | | 12 Bit Daten | | | | | | | | | | | |
| | VBx (Highbyte) | | | | | | | | VBx+1 (Lowbyte) | | | | | | | |

4.2 Library-Interner Speicherbereich

Dem Library-Internen Speicher muss, wie schon erwähnt, ein Speicherbereich zugewiesen werden. Dazu siehe Kapitel 2. In den folgenden Tabellen wurde ein Speicherbereich ab dem Byte VB900 gewählt. Dieser Bereich ist nur beispielhaft und muss nicht immer bei der Adresse VB900 beginnen.

Im internen Speicherbereich wird ein vollständiges Abbild der Analogeingangs- und Analogausgangswerte des CP 242-8 erstellt. Von diesem Speicherabbild werden die ASi Digitaldaten gelesen bzw. geschrieben. Die Datenbereiche der Analogslaves werden, im jeweiligen Schritt des Analogprotokolls angepasst.

Außerdem befinden sich in diesem Speicherbereich Datenworte zur Zwischenspeicherung unvollständig gesendeter oder empfangener Analogwerte.

Zuletzt sind hier noch Speicherworte zur Ablaufkontrolle der Analogprotokolle (ein Kontrollwort pro Analogslave) abgelegt. Durch diese Kontroll-Datenworte wird der richtige Ablauf des Übertragungsprotokolls gesteuert.

Die folgende Tabelle gibt Aufschluss über den verwendeten Speicherbereich:

| Adresse | Kommentar |
|---------|--------------------------------------|
| VW900 | ASi Speicherabbild von AEW0 |
| VW902 | ASi Speicherabbild von AEW2 |
| VW904 | ASi Speicherabbild von AEW4 |
| VW906 | Joystc Ctl. Word |
| VW908 | Joystick Up/Down Zwischenspeicher |
| VW910 | Joystick Left/Right Zwischenspeicher |
| VW912 | Druck/Dist Ctl. Word |
| VW914 | Druck Zwischenspeicher |
| VW916 | Distanz Zwischenspeicher |
| VW918 | DVM/MCSpeed Ctl- Word |
| VW920 | DVM Zwischenspeicher |
| VW922 | McSpeed Zwischenspeicher |
| VW924 | ASi Speicherabbild von AAW0 |
| VW926 | ASi Speicherabbild von AAW2 |
| VW928 | ASi Speicherabbild von AAW4 |

Wie man sieht werden insgesamt für den Library-Internen Speicherbereich 30 Byte gebraucht.

Die Speicher für Analogdaten sind so organisiert, dass im hinteren Teil des Datenwortes der Wert und im vorderen Teil des Wortes das SVO-Trippl gespeichert ist. Zu beachten ist, dass vom Slave ein 12 Bit großer Wert gelesen werden kann, jedoch zum Slave lediglich 11 Bit geschrieben werden können.

Analogeingang:

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|---|---|---|--------------|-----|-----|----|-------------------|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VW _x : | S | V | O | | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 |
| | SVO-Trippl | | | | 12 Bit Daten | | | | | | | | | | | |
| | VB _x | | | | | | | | VB _{x+1} | | | | | | | |

Analogausgang:

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|---|---|---|---|--------------|-----|----|-------------------|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VW _x : | S | V | O | | | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 |
| | SVO-Trippl | | | | | 11 Bit Daten | | | | | | | | | | |
| | VB _x | | | | | | | | VB _{x+1} | | | | | | | |

5 Funktion der Library

5.1 Initialisierung

Im Initialisierungsblock wird der Speicherbereich von VW900 ff. rückgesetzt. Damit wird verhindert, dass unbeabsichtigt alte Dateninhalte übernommen werden.

Durch das Setzen des PLC_Run Bits wird dem CP 242-8 Modul der »RUN-Zustand« der CPU mitgeteilt. Siehe [2] Seite 2-13.

Ebenfalls im Steuerbyte des Moduls wird die Bank ausgewählt. In diesem Fall Bank 0 durch Rücksetzen aller vier Bankauswahl-Bits (0h). Siehe [2] Seite 2-17 ff.

Zum Anstoß der Analogwertübertragung Richtung Slaves werden die McSpeed_init, DVM_init und das Kanalwechsel-Bit im DVM/McSpeed Control-Word initialisiert.

5.2 Daten lesen

Der Datenempfangsbaustein beginnt mit der Abfrage ob das CP242-8 Modul kommunikationsbereit ist. Falls nicht, wird sofort zum Hauptprogramm zurückgekehrt.

Danach wird ein Abbild der drei benötigten Analog-Eingangsworte in den Library internen Speicherbereich gelegt. Von dort werden die Daten der Digital-Slaves sofort dem User übergeben.

Folgend wird jeweils eine Runde des Analogprotokolls (siehe [3] Seite 10-12) für den Joystick (Slave 3) und Druck- bzw. Distanzsensor (Slave 5) absolviert.

Zuletzt wird überprüft, ob einer oder mehrere der vier Analogwerte (Joystick L/R, Joystick U/D, Druck und Distanz) vollständig übertragen wurde und gültig ist. Ist dies der Fall so wird der Wert dem User im Übergabespeicherbereich zur Verfügung gestellt. Falls ein Wert als invalid oder overflow gekennzeichnet ist wird er verworfen, eine neue Übertragungsrunde wird gestartet. In diesem Fall bleibt der alte (zuletzt gültige) Wert im Übergabespeicher erhalten.

5.3 Daten schreiben

Der Datensendebaustein arbeitet in umgekehrter Reihenfolge. Zunächst wird jedoch, wie beim Daten Empfangen, die Kommunikationsbereitschaft des CP 242-8 Moduls geprüft.

Danach werden die zu sendenden Daten für Digital-Slaves vom Übergabebereich in das Speicherabbild der Analogausgangsworte kopiert.

Die Bearbeitung einer Runde des Analogprotokolls (siehe [3] Seite 10-13) folgt. Es kann jedoch immer nur ein Kanal des Analogslaves (Slave 2), also entweder McSpeed oder DVM bedient werden. An die beiden Aktoren werden die Daten abwechselnd gesendet. Jeweils ein ganzes Datenwort. Diese Abwechslung wird durch das Kanalwechsel-Bit erzielt. Wurde ein vollständiges Datenwort an den Slave gesandt, so wird ein neues Datenwort vom Übergabebereich in den Zwischenspeicher (im Library-Internen Speicherbereich) geladen.

Zuletzt wird das Speicherabbild der drei Analogausgangsworte auf diese hinaus geschrieben.

Literatur

- [1] Siemens AG. *SIMATIC S7-200 Automatisierungssystem Systemhandbuch*.
Dokumentation für SPS Programmierung mit S7-200 Komponenten.
- [2] Siemens AG. *SIMATIC NET CP 242-8 AS-Interface Master / PROFIBUS-DP Slave*
Dokumentation des CP 242-8 Moduls.
- [3] Siemens AG. *AS-Interface Analogmodule*.
Dokumentation über ASi Analogmodule.
- [4] Wolfgang Kastner, Ulrich Schmid, Friedrich Kastner-Masilko. *Laborübung Prozeßautomatisierung, Targetsysteme*.
Dokumentation der Targetsysteme.