SPSSim Benutzerhandbuch

Martin Kögler e9925248@stud4.tuwien.ac.at

2. März 2005

Inhaltsverzeichnis

1 Einleitung

SPSSim ist im Rahmen zweier Praktika am Institut für Rechnergestützte Automation der TU-Wien entstanden.
Das Ziel war es, einen Simulator zu schaffen, mit dem man einen so großen Grundstock an Operationen einer Siemens S7-214 SPS ausführen kann, daß die Übungsbeispiele ablauffähig sind.
Das Ziel des ersten Praktikums war es, die ersten Übungsbeispiele ablauffähig zu machen. Der Simulator mußte dazu nur die SPS selbst und vom Bedienpanel Swt0 bis Swt7 und Led0 bis Led7 unterstützen.
Das Ziel des zweiten Praktikum war es, auch den zweiten Teil der Übungsbeispiele ablauffähig zu machen. Dazu mußte das Targetsystem samt ASI-Modulen simuliert werden.
Der Simulator ist primär für Microsoft Windows geschrieben. Durch die geringe Systemabhängigkeit ist er auch mittels Wine auf Linux-Systemen problemlos lauffähig.

2 Funktionsumfang

Der Simulator implementiert einen Teil des Befehlssatzes der S7-214 SPS, ein Subset des CP242-8, ein Subset diverser ASI-Module und des Targetsystems. Unterstützt werden (die Kapitel beziehen sich auf das S7-200 Handbuch):

- Timer: Es sind 128 Timer mit den entsprechenden Auflösungen vorhanden.
- Zähler: Es sind 256 Zähler verfügbar.
- Sondermerker: Es ist SM0 implementiert.
- Speicherbereiche: Es sind M, SM, V, S, E, AC und A implementiert, jeweils mit Größenbeschränkung.
- Befehle: Es sind alle Kontakte und Spulen aus Kapitel 10.2 bis 10.4 vorhanden, wobei Set und Reset nur eingeschränkt auf Zähler und Timer anwendbar sind.
- Zähler / Timer: Es sind Boxen lt. Kapitel 10.5 ohne High-Speed Zähler, Impulsausgabe und Echtzeituhr realisiert.
- Aritmetische Operationen: Aus Kapitel 10.6 fehlen PID-Regler. MUL und DIV sind zwar vorhanden, aber nicht wirklich getestet worden und haben in der KOP-Darstellung Probleme mit den Wertebereichen.

- Inc/Dec Operation: Alle Operationen aus Kapitel 10.7 sind vorhanden.
- Übertragungs / Tabellenoperationen: Alle Operationen aus Kapitel 10.8 sind vorhanden.
- Schiebeoperationen: Aus Kapitel 10.9 fehlt SHBR.
- Programmsteuerung: Aus Kapitel 10.10 sind nur CALL/RET und SCRs implementiert.
- Stackoperationen: Aus Kapitel 10.11 sind alle vorhanden.
- Verknüpfungsoperationen: Aus Kapitel 10.12 sind alle vorhanden.
- Umwandlungsoperationen: Aus Kapitel 10.13 sind nur Real<->DI Konversionen implementiert.
- ASI-Unterstütztung: Es ist Bank 0 des CP242-8 vorhanden und von den ASI-Slaves nur der normale Datenaustausch.
- Targetsystem: Es sind alle Komponenten lt. Spezifikation vorhanden. Die Werte für die Kalibrierung wurden auf Target 2 gemessen.
- Profibus DP Interface: Es wird der DP-Slave des CP242-8 emuliert. Es gibt als weiteres Packet, das grundlegende Interfaces des CP5613 emuliert.

3 Eingabedaten

Der Simulator liest AWL-Dateien ein und führt sie dann aus. Diese AWL-Dateien können mit Datei/Exportieren in der 7 Step 32 MicroWin Entwicklungsumgebung erstellt werden.
An Einschränkungen gibt es:

- Unterprogrammnamen müssen entweder mit einen Kleinbuchstaben oder mit den Prefix SBR beginnen. Die Defaultnamen können daher ohne Probleme verwendet werden.
- Man darf in den Projekt keine Bibliothek in der IDE verwenden, da diese unbrauchbar exportiert werden.

4 Benutzeroberfläche Teil 1

Die Benutzeroberfläche besteht aus einem Menü, über das alle Funktionen aufgerufen werden können. Darunter befinden sich die Schalter und die Leds des simulierten Bedienpanels.
Es besteht aus insgesamt 8 Schaltern und 9 Leds. Die Schalter sind an EB0 angeschlossen, die 8 Leds darüber an AB0 und die neunte Led (der Buzzer) an A1.0.

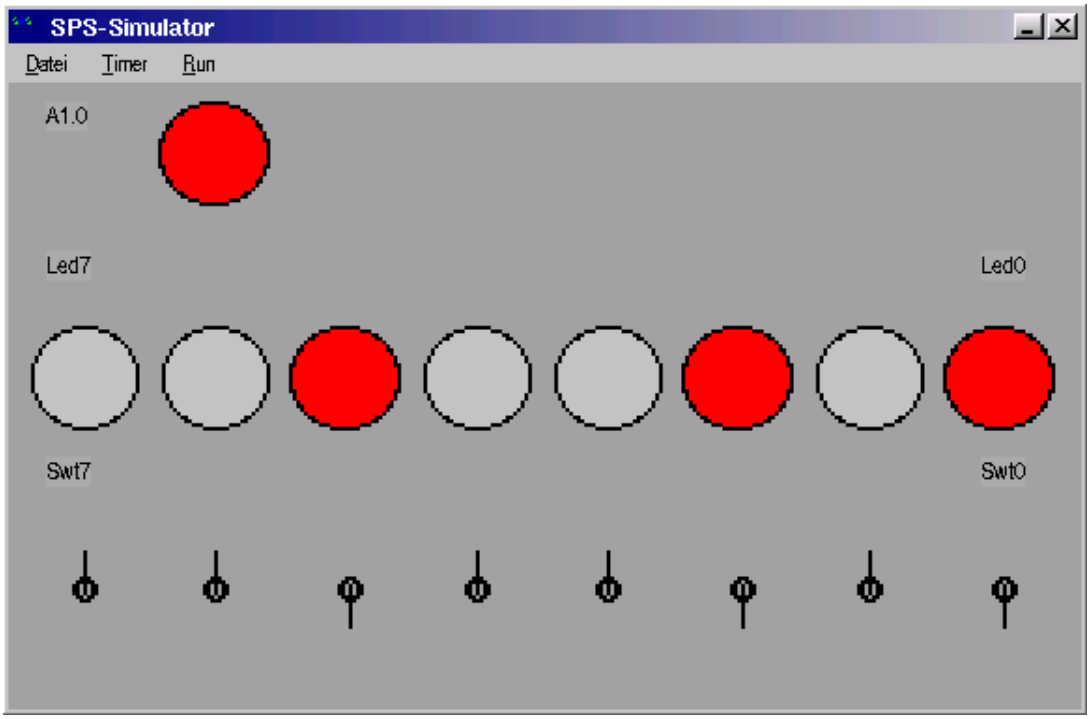

Abb 4.0.1: Hauptfenster

5 Bedienung Teil 1

Zuerst muß man per Datei/Laden (bzw. F3) eine AWL-Datei in den Simulator laden. Wenn dabei keine Fehlermeldung ausgegeben wird, kann man das Programm ablaufen lassen. Nach den Start des Simulators wird das Programm mit Echtzeittimern ausgeführt.
Die Bedienung erfolgt entweder mit der Maus oder mit der Tastatur.
Ein Schalter kann durch einfaches Anklicken umgeschaltet werden. Durch die

Eingabe von 0 - 7 wird der Schalter für E0.X umgeschalten, wobei X für die eingetippte Ziffer steht.

Abb 5.0.1: Dateimenü, Teil 1

6 Benutzeroberfläche Teil 2

Die Funktionsweise ist im Prinzip gleich wie bei der Version für Teil 1, nur wird das Targetsystem samt einen Teil der CP242-8 simuliert.

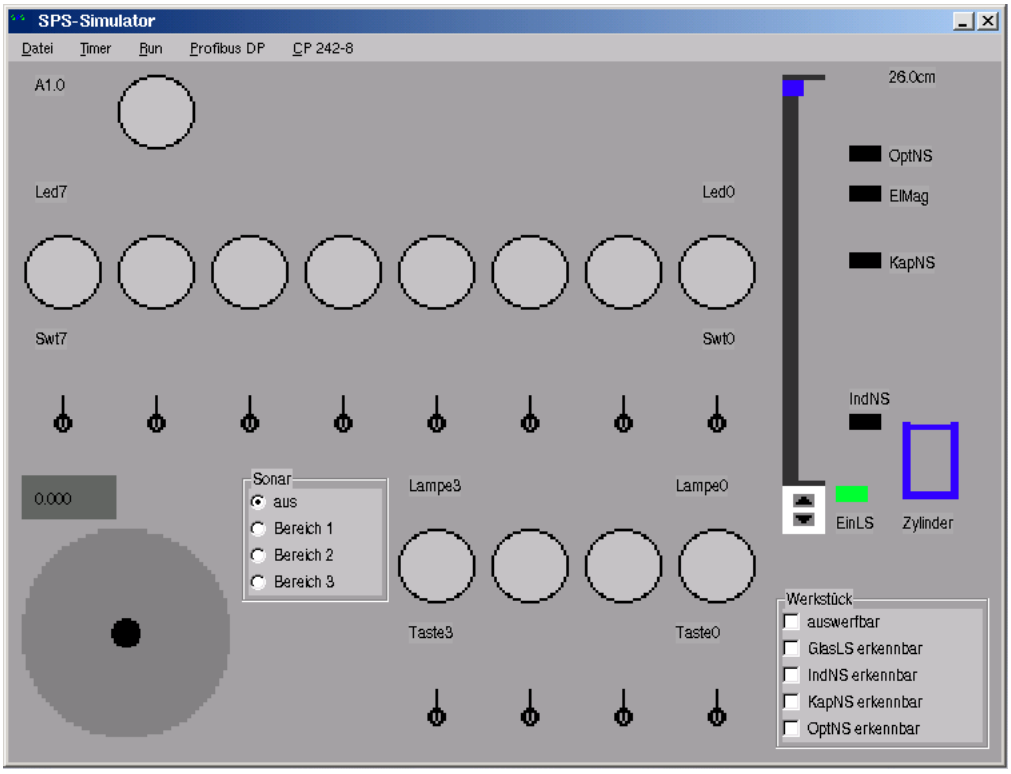

Abb 6.0.2: Hauptfenster der ASI-fähigen Version

Abb 6.0.3: Dateimenü, Teil 2

6.1 ASI-Master

Es wird ein Subset des ASI-Masters des CP242-8 zur Verfügung gestellt. Man muß unbedingt E2.1 (Data Ready) beachten. Weiters ist nur die Bank 0 verfügbar. Der Komunikationsprozessor läuft asynchron zur SPS.

6.2 DP-Slave

Es wird ein Subset des Profibus DP-Slave des CP242-8. Man muß E2.1 (Data Ready) beachten. Der Komunikationsprozessor läuft asynchron zur SPS.

7 Bedienung Teil 2

Der Simulator stellt die ASI-Sensoren lt. Targetsystembeschreibung zur Verfügung. Die einzelen Sensoren können wie folgt gesteuert werden:

7.1 Taster

Die Taster sind als die 4 Schalter mit den Lampen in der unteren Reihe ausgeführt. Sie werden als Schalter repräsentiert, da man so andere Funktionen aufrufen kann, ohne daß man den Taster dabei loslassen muß.

7.2 DVM

Das DVM ist das graue Feld links unter dem Schalter Swt7.

Abb 7.2.1: DVM

7.3 Joystick

Der große Kreis symbolisiert den Joystick. Die aktuelle Position wird durch den schwarzen Kreis angezeigt. Durch einen Klick auf eine Stelle wird der Steuerknüppel dort hinverschoben. Weiters kann der Joystick auch mit den Cursortasten beeinflußt werden.

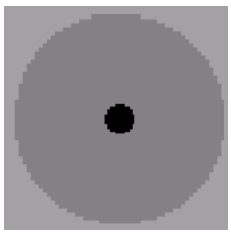

Abb 7.3.2: Joystick

7.4 Sonar

Der Wert des Sonars wird über die 4 Radiobuttons gesteuert. Bei *aus* wird kein Gegenstand zurückgemeldet, bei *Bereich 1* wird D0, bei *Bereich 2* wird D1 und bei *Bereich 3* wird D2 aktiviert.

Abb 7.4.3: Sonar

7.5 Schlitten

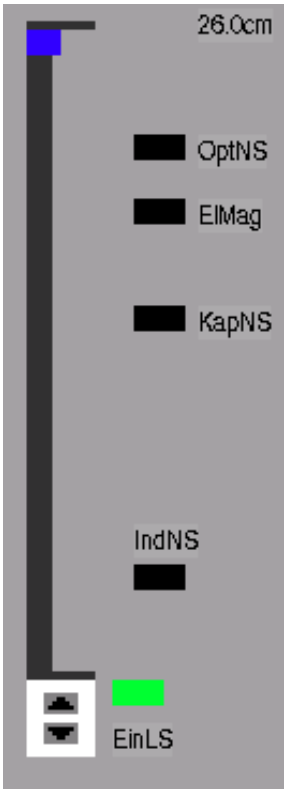

Abb 7.5.4: Werkbank

Der Schlitten kann durch das Anklicken der 2 Pfeile im weißen Kasten (bzw. PageUp / PageDown) manuell gesteuert werden. Das blaue Rechteck gibt die aktuelle Schlittenposition an. Rechts daneben sind noch zur Orientierung die Position der Sensoren der Sensorbank eingezeichnet.
Das Ansprechen der einzelnen Sensoren wird durch eine Änderung der Farbe dargestellt. Bei IndNS werden 3 Farben verwende: neben schwarz für den nicht angesprochenen Zustand, gibt es ein helles Grün, was anzeigt, das ein Gegenstand sicher erkannt wurde und einen dunkleren Farbton, der anzeigt, das möglicherweise ein Gegenstand erkannt wurde. Das entspricht den Bit 1 vom Sensorresultat.
Das Ansprechen von PosElm wird durch die dunklere Farbe dargestellt, ein ausgefahrener Magnetarm durch die hellere.

7.6 EinLS

Das rote/gelbe Rechteck symbolisert den Zustand von EinLS. Durch Anklicken kann er geändert werden. Wenn es rot ist, liefert EinLS den Wert 1 und der Schlitten kann nicht bewegt werden (auch nicht manuell).

7.7 Zylinder

Die aktuelle Stellung des Zylinders wird durch das blaue Gebilde am rechten Rand dargestellt. Der obere Strich gibt die Postion des Kolben an.

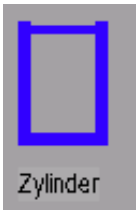

Abb 7.7.5: Zylinder

7.8 Werkstück

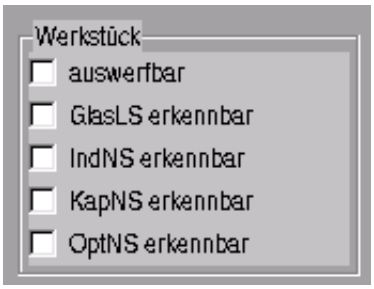

Abb 7.8.6: Werkstück

Im Simulator können virtuelle Werkstücke verwendet werden, deren Eigenschaften über die unter Werkstück zusammengefaßten Optionen steuerbar ist.

- *auswerbar*: Das Werkstück wird von PosElm erkannt, wenn der Schlitten dort ist. Wenn die Auswerffunktion erfolgreich durchgeführt wird, wird diese Option wieder deaktiviert.

- *GlasLS erkennbar*: Das Werkstück wird von GlasLS erkannt, wenn der Kolben oben ist.
- *IndNS erkennbar*: Das Werkstück wird von IndNS erkannt, wenn der Schlitten dort ist.
- *KapNS erkennbar*: Das Werkstück wird von KapNS erkannt, wenn der Schlitten dort ist.
- *OptNS erkennbar*: Das Werkstück wird von OptNS erkannt, wenn der Schlitten dort ist.

7.8.1 Hinweise zum Auswerfen

Beim Auswerfen muß man beachten, das die erfolgreiche Durchführung dieser Aktion von den Sensor PosElm, der Stellung das Zylinders und den Aktivierungszeitpunkt des Elektromagneten abhängt.
Zuerst einmal wird *nicht* aus, wenn der Zylinder unten ist. Das entpricht der Tatsache, das ein eingespanntetes Werkstück vom Elektromagneten höchstens verschoben werden kann. Nach so einen erfolglosen Versuch muß man den Elektromagneten erst deaktivieren, bevor man einen neuen Auswurfversuch durchführen kann.
Weiters muß der Sensor PosElm ansprechen, wie beim richtigen Targetsystem. Es wurde auch die Schutzschaltung nachgebildet, die ein auswerfen verhindert, wenn der Elektromagnet vor den Sensor PosElm aktiviert wird. Man darf erst nach den Ansprechen von PosElm ElMag aktivieren, wobei der Zylinder teilweise geöffnet sein muß, um einen erfolgreichen Auswurf durchzuführen.

8 Debuggen

Der Simulator bietet umfangreiche Debugging-Möglichkeiten. Dabei sind die Funktionen in 6 Gruppen aufteilbar:

8.1 KOP-Ansicht

Die Funktion Datei/Debugansicht (bzw. F2) öffnet ein Fenster, in dem der KOP vom Programm dargestellt wird. Dort wird zu jedem Parameter auch der interne Zustand angezeigt, so daß man einen Überblick über den Zustand des Systems erhält.
Über Scrollbars, Pfeil-Tasten, PageUp/Down, Home und End kann man das Bild scrollen.

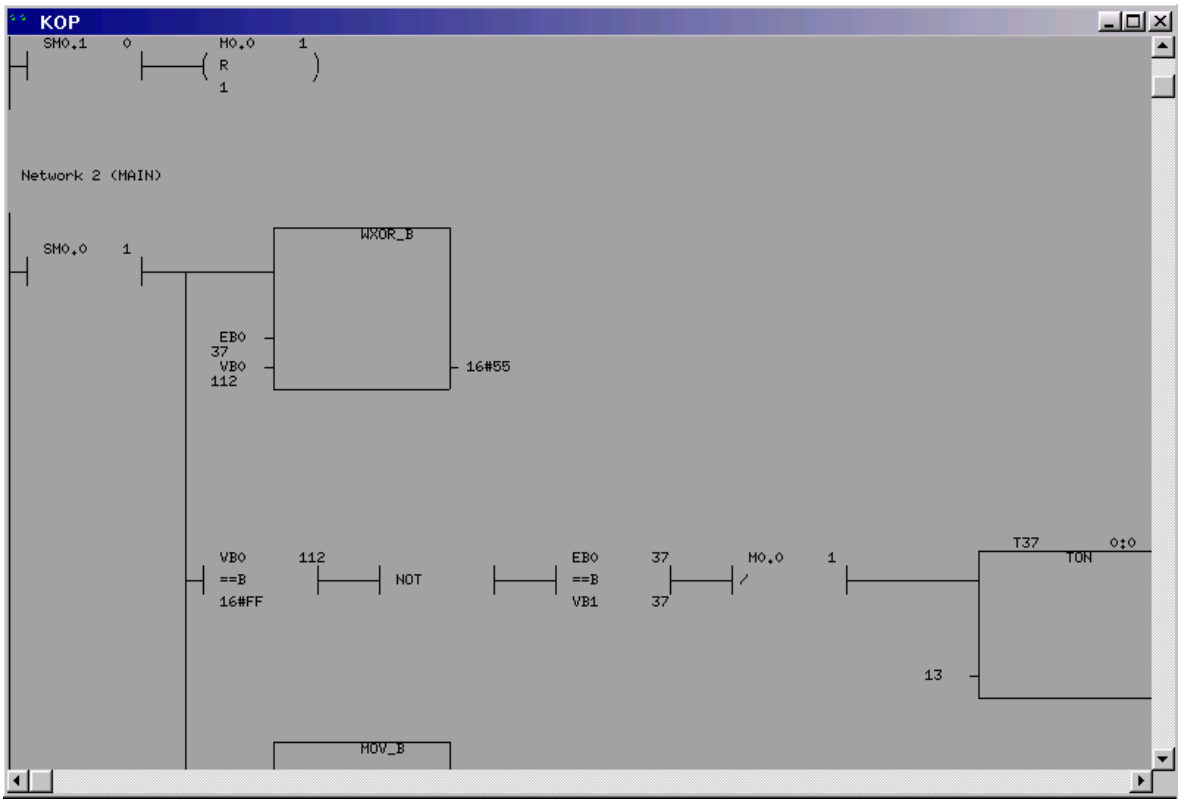

Abb 8.1.1: Debugansicht

Es wird ca. 2 mal pro Sekunde upgedatet.

8.2 CPU-Ansicht

Die Funktion Datei/CPU-Status (bzw. F4) öffnet ein Fenster, in dem einige Zustandswerte der CPU dargestellt werden. Die Bedinung entspricht der des KOP-Fensters.
Die Bedienung dieses Fensters funktioniert nach dem Prinzip das KOP-Fensters.

8.3 ASI-Ansicht

Die Funktion Datei/ASI-Status (bzw. F1) öffnet ein Fenster, in dem einige Zustandswerte der ASI-Komponenten dargestellt werden. Es werden immer die Werte angezeigt, die auf den Bus übertragen werden bzw. die von dort

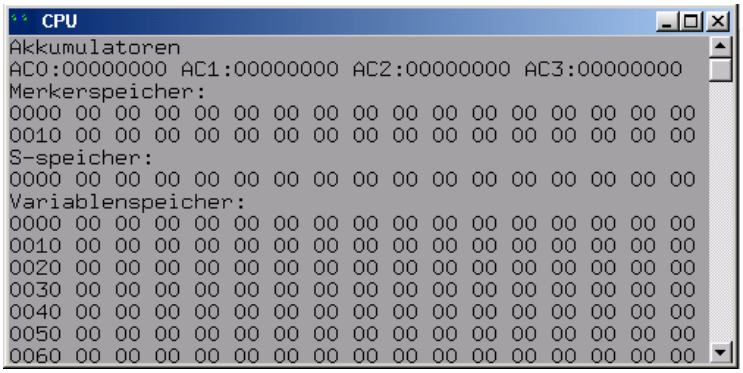

Abb 8.2.1: CPU-Info

gelesen werden.
Die Bedienung entspricht der des KOP-Fensters. Es steht nur in der ASI-fähigen Version zur Verfügung.
Die Bedinung diese Fenster funktioniert nach den Prinzip das KOP-Fensters.

8.4 Timer

Die Kontrolle über die Timer und SM0.5 kann über das Timermenü und die dort angegebenen Shortcuts ausgeübt werden. Es gibt 2 Timermodi:

- Automatisch: Die Timer werden automatisch weitergestellt. Über die Menüfunktionen Schneller (bzw +) und Langsamer (bzw -) kann die Timergeschwindigkeit beeinflußt werden.
- Manuell: In diesem Modus werden die Timer nicht manuell weitergestellt. Über die Funktionen 1 ms (bzw. F6), 10 ms (bzw. F7), 100 ms (bzw. F8) und 1000 ms (bzw. F9) kann ein Sprung in der Systemzeit der SPS um diesen Wert ausgelöst werden.

Die Modi werden über Timer/Automatisch und Timer/Manuell gewechselt. Es ist zu beachten, daß es durch den Moduswechsel zu Zeitsprüngen in der SPS kommen kann. Um ungewollte Nebeneffekte zu vermeiden sollte ein Reset danach ausgeführt werden.

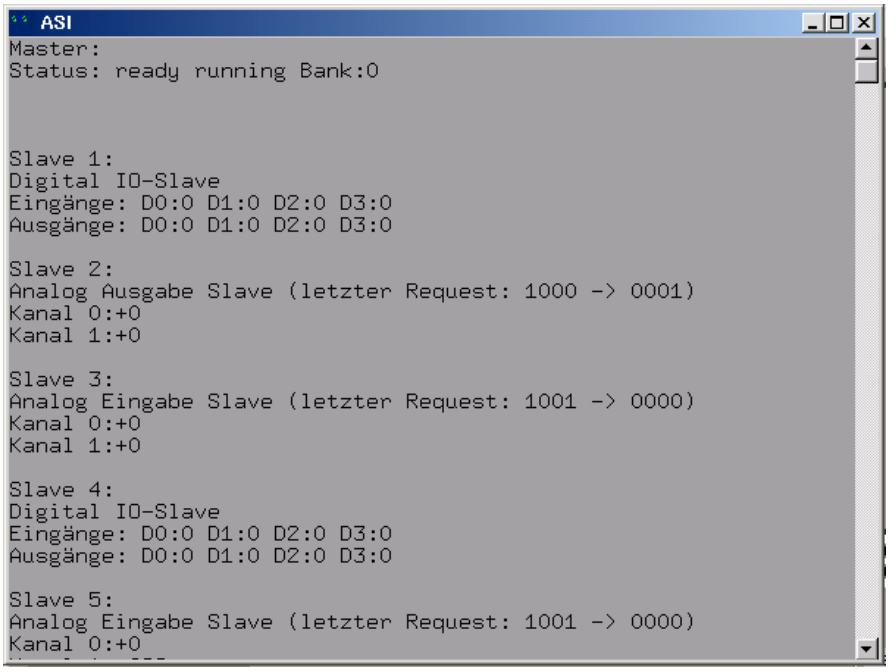

Abb 8.3.1: ASI-Info

Abb 8.4.1: Timermenü

Beim Debuggen mit den manuellen Timern in der ASI-fähigen Version gilt es zu beachten, daß sich der Schlitten und der Zylinder nur bewegt, wenn die Zeit manuell weitergestellt wird. Weiters werden vom Targetsystem nicht alle Eingaben verarbeitet, wenn die CPU gestoppt wird. Bei der Ausführung des nächsten Einzelschritts wird das aber nachgeholt.

8.5 CPU-Zustand

Der CPU-Zustand kann über das CPU-Menü beeinflußt werden. Die CPU kann über Run/Stop (bzw. F10) angehalten werden. Mit Run/Einzelschritt (bzw. SPACE) kann dann ein Zyklus simuliert werden. Mit Run/Continue (bzw. F5) wird die Anhaltung wieder aufgehoben.
Mit Run/Reset (bzw. F12) wird ein Reset durchgeführt. Dabei wird SM0.1 für einen Zyklus auf 1 gesetzt. Wenn bei den Reset auf der Speicher zurückgesetzt werden soll, verwendet man statt Run/Reset Run/Clear+Reset (bzw. F11).

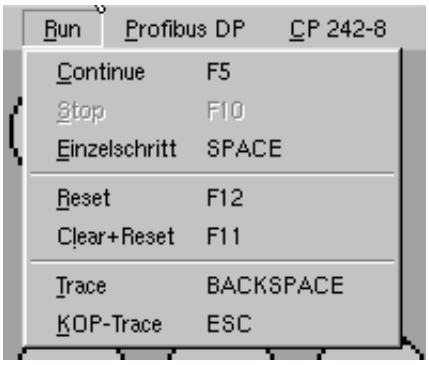

Abb 8.5.1: Befehle zur CPU-Steuerung

8.6 KOP-Trace

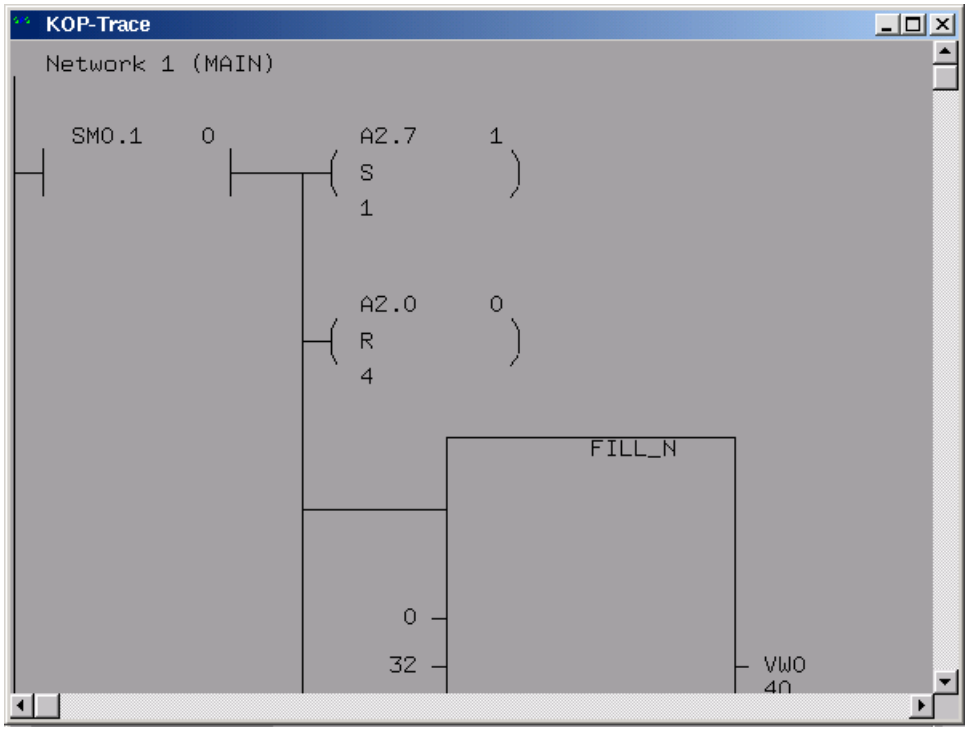

Abb 8.6.2: KOP-Trace

Mit den KOP-Trace kann man zumindestens teilweise die Änderung von Werten verfolgen, ohne auf die auf die AWL-Ebene gehen zu müssen.
Beim Aufruf dieser Funktion (Run/KOP Trace bzw ESC) wird ein Zyklus ausgefuehrt. Dabei wird immer, wenn ein Netzwerk komplett abgeschlossen ist, dessen momentaner Zustand als KOP an die Ausgabe angehängt.
Wenn man den KOP Trace von oben nach unten durchschaut, kann man einen Überblick über die Werteänderung im Zyklus bekommen. Der KOP Trace wird nur aktualisiert, wenn er aufgerufen wird.
Folgende Einschraenkungen gibt es:

- Netzwerke mit CALLS darinen werden immer nach den ausgeführten Unterprogramm ausgegeben.
- Bei der Verwendung von JMP koennen nicht ausgeführte Netzwerke angezeigt werden.
- Netzwerke, in denen ein RET ausgefuehrt wird, werden unterdrückt.

8.7 Trace

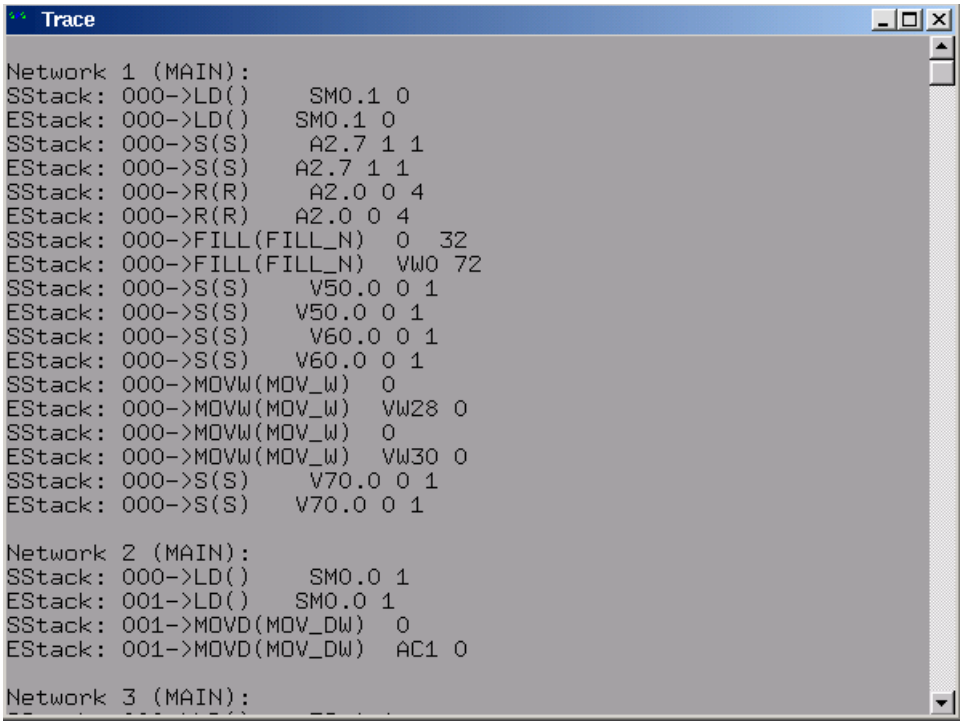

Abb 8.7.3: Trace

Als Lowlevel-Debughilfe gibt es die Trace-Funktion. Sie ist für Fortgeschrittene gedacht, vor allen für Probleme, die innerhalb eines Zykluses auftreten, wie z.B. das man bei einen SCR-Beispiele eine Schleife hat, so daß in jeden Zyklus alle zustände durchläuft, was dann aussieht, als wie wenn kein Zustandswechsel stattgefunden hätte.
Die Trace-Ansicht wird mit BACKSPACE oder Run/Trace aufgerufen. Die Anzeige kann wie bei die KOP-Ansicht gescrollt werden.
Der Trace dient zur Darstellung des Verhaltens der SPS während eines Zykluses. Durch den Aufruf der Funktion wird ein Schnappschuß von einem Zyklus erzeugt, der dann bis zum nächstes Aufruf der Funktion in den Fenster angezeigt wird. Die Funktion kann im Einzelschritt wie auch im normalen Ablaufmodus ausgeführt werden.
Da der Output nicht sehr verständlich ist, folgt hier eine genauer Erklärung: Wenn begonnen wird, ein Netzwerk auszuführen, wird dessen Kennung angezeigt.
In dieser Anzeige wird die Anweisungsliste und nicht KOP als Basis genommen. Es wird jeder ausgeführte Befehl mitprotokolliert, nicht angezeigte

Befehle werden nicht ausgeführt.
Jede Befehl wird durch 2 Zeilen dargestellt:

- In der ersten Zeile wird der Zustand vor der Ausführung des Befehls dargestellt.
- In der zweiten Zeile wird der Zustand nach der Ausführung des Befehls dargestellt.

Bei Unterprogramm-Aufrufen, wird das Unterprogramm zwischen die 2 Zeilen für den Call-Befehl eingebunden.
Der Aufbau einer Zeile ist folgende:

- SStack für den Befehlsanfang, EStack für das Befehlsende
- Die 3 obersten Stackwerte, wobei der oberste Wert rechts steht.
- Der Befehlsname: Es folgt der AWL-Befehlsname, auf den in Klammern der KOP-Name zur Information folgt. Man sollte sich an den AWL-Namen orientieren, da KOP-Programme in Befehle aufgelöst werden, die es nur in AWL gibt.
- Nun folgen die Parameter: Es wird immer der Name des Parameters wie in der KOP-Ansicht dargestellt, darauf folgt dann noch der Inhalt, wenn er auch in der KOP-Ansicht angezeigt werden würde. In der ersten eines Befehls werden die Parameter vor der Befehlsausführung, in der zweiten nach der Befehlsausführung angezeigt.

Folgendes gibt es zu beachten:

- Bei der Verwendung von JMP koennen Netzwerke angezeigt werden, die nicht ausgefuehrt werden. Aus den Befehlen sollte hervorgehen, das das nicht der Fall ist.
- Beim Verlassenen eines Unterprogrammes wird der Rest vom Netzwerk, das das Unterprogramm aufgerufen hat, direkt an das letzte ausgefuehrte Netzwerk vom Unterprogramm angehaengt, ohne das das der Netzwerkwechsel angezeigt wird.

Bei Fragen zu AWL wird die Lektüre des SPS-Manuals geraten.

8.8 Race Conditions

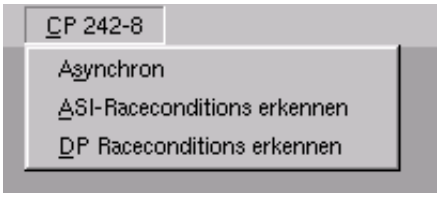

Abb 8.8.4: Race Condition Erkennung

Es gibt bei den späteren Teilen immmer, die auf den richtigen Targetsystem funktionieren, aber am Simulator nicht, oder umgekehrt. In den meisten Fällen kommt das dadurch Zustande, das entweder der Ready Marker des CP242-8 nicht beachtet wird, eine Register des Kommunikationskontrollers mit zwei verschieden Werten schnell hintereinander beschrieben wird oder das erwartet wird, das ein die Daten von CP242-8 zu bestimmten Zeiten geliefert oder gelesen werden.
Zum Erkennen solcher Situationen gibt es im Menü *CP 242-8* folgende Optionen:

Asynchron Diese Option kann man durch anwählen abwechselnd ein und ausschalten. Sie bewirkt, das der Zyklus des CP242-8 nicht mehr synchron zum Zyklus der SPS ausgeführt wird. Durch sie sollten Programme mit Raceconditions im ASI-Teil ausführbar werden.

Ein gutes Programm sollte mit deaktivierten Option ohne Fehler laufen.

ASI-Raceconditions erkennen Diese Option kann man durch anwählen abwechselnd ein und ausschalten. Wenn sie aktiv ist, kommt eine Fehlermeldung und der Simulator wird gestoppt, wenn ein Wert auf der ASI-Bank vom Programm mit einen anderen überschrieben wird, bevor er vom CP242-8 verarbeitet worden ist. Mit *Run/Continue* kann die Ausführung fortgesetzt werden.

Ein gutes Programm sollte mit aktivierten Option ohne Fehler laufen.

DP-Raceconditions erkennen Diese Option kann man durch anwählen abwechselnd ein und ausschalten. Wenn sie aktiv ist, kommt eine Fehlermeldung und der Simulator wird gestoppt, wenn ein Wert auf einer DP-Bank vom Programm überschrieben wird, bevor er vom CP242-8 verarbeitet worden ist. Mit *Run/Continue* kann die Ausführung fortgesetzt werden.

Ein gutes Programm sollte mit aktivierten Option ohne Fehler laufen.

9 Profibus DP

Der Simulator für Teil 2 unterstützt auch den DP-Slave des CP242-8.

Abb 9.0.5: Profibus DP Menü

Zum Verbinden mit einen laufenden Masterprogramm, ruft man *Profibus DP/Verbinden* auf. Die Standartwerte für die Übung sollten genügen. Man kann aber auch die IP-Adresse eines anderen Rechners, wie auch ein andere TCP Port eintragen.
Weiters kann man die Slave Addressen ändern, wenn man mehrere SPSSIM Instanzen mit einem Masterprogramm verbinden will.

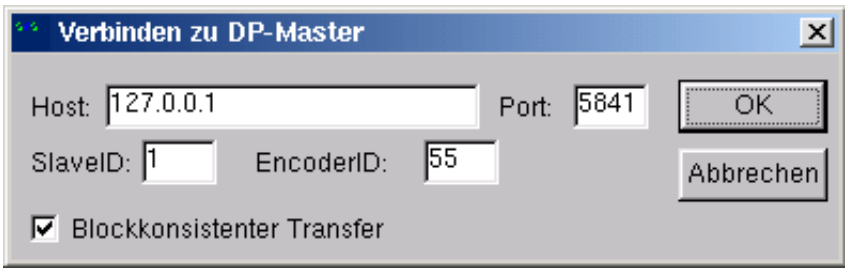

Abb 9.0.6: Profibus DP Verbindung

Beim Verbinden wird eine TCP-Verbindung aufgebaut. Wenn man eine Firewall verwendet, muss man SPSSIM den Zugriff gestatten.
Mit *Profibus DP/Trennen* kann man die Verbindung wieder beenden. Ob eine Verbindung besteht, kann man überprüfgen, indem man prüft, ob *Trennen* oder *Verbinden* im Menü *Profibus DP* anwählbar ist.
Für die Einbindung der SPSSIM-Schnittstelle in ein Masterprogramm gibt es ein eigenes Handbuch.

10 Einschränkungen

Um Windows nicht total auszulasten, wird nur ca. jede ms ein Zyklus ausgeführt, daher können nicht alle Seiteneffekte beobachtbar sein. Weiters kann

der Simulator KOP-Konstrukte anzeigen und ausführen, die in der 7-Step MicroWin Entwicklungsumgebung nicht erzeugbar sind.
Zudem erzeugt die 7-Step MircoWin aus Funktion wie z.B. SUB_B teilweise länger Befehlesfolgen, die der Simulator in mehreren Funktionsboxen in der KOP-Ansicht anzeigt.
ASI ist nur Teilweise implementiert, es ist nur Bank 0 des CP242-8 und von den Slaves nur der normale Datenaustausch vorhanden.
Von Profibus DP ist so viel implementiert, wie für die Übung notwendig ist.

A Allgemeine Shortcuts

Shortcut	Funktion
F2	KOP-Ansicht
F3	Laden
F4	CPU-Status
F5	Aufführung vorsetzen
F6	1 ms vergeht (im manuellen Modus)
F7	10 ms vergeht (im manuellen Modus)
F8	100 ms vergeht (im manuellen Modus)
F9	1000 ms vergeht (im manuellen Modus)
F10	Ausführung stoppen
F11	Clear+Reset
F12	Reset
+	Timer schneller (im automatischen Modus)
-	Timer langsamer (im automatischen Modus)
Leerzeichen	Einzelschritt ausführen
0 - 7	Schalter Swt0 - Swt7 umschalten
BACKSPACE	Trace erzeugen
ESC	KOP-Trace erzeugen

B ASI-spezifische Shortcuts

Shortcut	Funktion
F1	ASI-Status
Pfeiltasten	Joystick bewegen
PageUp	Schlitten fährt hinauf
PageDown	Schlitten fährt hinunter
E e	Wert von EinLS ändern
A a	Zustand von Taster 0 ändern
B b	Zustand von Taster 1 ändern
C c	Zustand von Taster 2 ändern
D d	Zustand von Taster 3 ändern
Tabulator	Joystick in 0-Stellung bringen

C Targetsystem-Werte für den Simulator

DIST-Wert	Sensor
700	Anschlag Oben
1158	OptNS
1354	ElMag
1838	KapNS
2827	IndNS
3130	Anschlag Unten

Der Schlitten hat eine Bewegungsstrecke vom 260 mm. Die Sensoren haben einen Ansprechbereich von ca. 1 cm.
Der Joystick liefert einen Wertebereich von ca. -350 bis +350.
10 V beim DVM entspricht den Wert von +3456.
Der Druck im System ist ca. 3.5 bar.

D Werkstücke

Werkstück	auswerfbar	GlasLS	IndNS	KapNS	OptNS
Alublock	x	x	x	x	x
schwarzer Metallblock	x	x	x	x	
weißer Metallblock	x	x		x	x
grauer Plastikblock	x	x		x	