

ASi-Beispielprogramm

Inhaltsverzeichnis

| | |
|--|---|
| Programmfunktion | 2 |
| Eingesetzte Hardware..... | 2 |
| Eingesetzte Software | 2 |
| Mitgeltende Unterlagen | 2 |
| Hardwarekonfiguration..... | 2 |
| AS-i/Profibus-Gateway..... | 3 |
| SIMATIC Step7 Bausteine | 3 |
| Programmablauf | 4 |
| Upload der ASi-Konfigurationsdaten..... | 4 |
| Download der ASi-Konfigurationsdaten | 6 |
| Durchführung der ASi-Mailboxbefehle..... | 7 |

Programmfunktion

In diesem Beispielprogramm werden die Ist-Konfigurationsdaten eines ASi-Stranges mit bis zu 62 Slaves ausgelesen und in einem Datenbaustein gespeichert. Die eingelesenen Ist-Konfigurationsdaten, bzw. per Hand eingetragene Soll-Konfigurationsdaten, können von dem Datenbaustein in das ASi-Gateway downgeloadet werden.

Eingesetzte Hardware

| | |
|---|--|
| SIMATIC S7 Netzteil | PS 407 4A |
| SIMATIC S7 CPU mit Profibus DP | CPU 412-2 PD Best.Nr.:6ES7 412-2XG00-0AB0 Firmware Version 3.0 |
| Pepperl+Fuchs AS-i/Profibus-Gateway | VBG-PB-K20-D |
| Pepperl+Fuchs AS-i Power Extender | |
| Pepperl+Fuchs AS-i 4E/4A-Modul | |
| Pepperl+Fuchs AS-i Leuchttastermodul | |
| Pepperl+Fuchs AS-i Drehgeber | |
| Pepperl+Fuchs AS-i 2E-Sicherheitsmodul | |
| Pepperl+Fuchs AS-i 2E/2A-Sicherheitsmodul | |
| Pepperl+Fuchs AS-i Sicherheitsmonitor | |
| Not-Aus-Schalter | |
| Netzteil | |

Eingesetzte Software

| | | |
|--|----------------------------|------------------------|
| Pepperl+Fuchs GSD-File für das AS-i/Profibus Gateway | | |
| SIMATIC Step7 | Version 5.1 Service Pack 3 | Ausgabestand: K5.1.3.0 |
| Programmbeispieldatei | | PF_ASi15.zip |

Mitgeltende Unterlagen

Pepperl+Fuchs AS-Interface/Profibus Gateway Bedienungsanleitung
SIEMENS S7-400 Dokumentation

Hardwarekonfiguration

Aus dem Hardwarekatalog werden die unter SIMATIC 400 befindlichen Baugruppen

1. Profilschiene
2. Netzteil PS 407 4A
3. CPU CPU 412-2 DP

dem Projekt hinzugefügt. Bei der Auswahl der CPU Baugruppe ist auf die richtige Hardwareversion und die Firmwareversion zu achten.

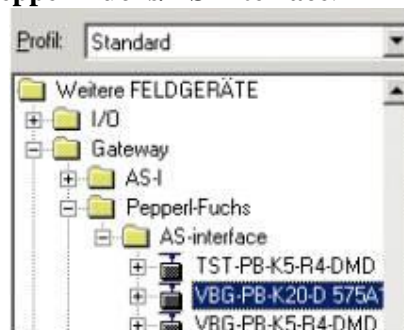
| Steckplatz | Baugruppe | Bestellnummer | Firmware | MPI-Adres... | E-Adresse | A-Adresse | Kommentar |
|------------|--------------|---------------------|----------|--------------|-----------|-----------|-----------|
| 1 | PS 407 4A | 6ES7 407-0DA01-0AA0 | | | | | |
| 2 | CPU 412-2 DP | 6ES7 412-2XG00-0AB0 | V3.1 | 2 | | | |
| X2 | DP | | | | 4095* | | |
| X1 | MPI/DP | | | 2 | 4094* | | |

Hardwarekonfiguration Netzteil und CPU-Baugruppe

AS-i/Profibus-Gateway

Nachdem die SIMATIC Hardware der Hardwarekonfiguration hinzugefügt und der Profibus konfiguriert wurde, kann das Pepperl+Fuchs AS-i/Profibus-Gateway dem Projekt hinzugefügt werden.

Das AS-i/Profibus-Gateway befindet sich im Hardwarekatalog unter **Profibus/Weitere FELDGERÄTE/Gateway/Pepper-Fuchs/AS-interface**.



AS-i/Profibus-Gateway im Hardwarekatalog

Das AS-i/Profibus Gateway besitzt im Hardwarekatalog die Bezeichnung **VBG-PB-K20-D 575A1745** und kann nun per Drag and Drop dem Profibusstrang hinzugefügt werden.

Öffnet man das Gerät „VBG-PB-K20D 575A1745“ durch Betätigen des Pluszeichens im Hardwarekatalog erscheint eine Liste der möglichen Profibus Kommunikationsmodule.

Zu diesem Zeitpunkt sollte das gewünschte Profibus Kommunikationsmodul parametrierbar werden. Dies erfolgt in folgenden Schritten:

1. Markieren des AS-i/Profibus-Gateway durch einen Mausklick auf das Slave Symbol. Danach wird am unteren Bildrand eine Tabelle angezeigt, welche Zeilen mit Steckplatz 0 beginnend enthält.
2. Auswahl des benötigten Kommunikationsmodul „Command Interface“ aus dem Hardwarekatalog. Hier „12 Byte Command Interface“.
3. Ziehen des ausgewählten Kommunikationsmoduls auf die Tabellenzeile Steckplatz 0.

| Steckplatz | DP-Kennung | Bestellnummer / Bezeichnung | E-Adresse | A-Adresse | Kommentar |
|------------|------------|-----------------------------|-----------|-----------|-----------|
| 0 | 193 | 12 Byte Command Interface | 512...523 | 512...523 | |

Kommunikationsmodule des AS-i/Profibus-Gateways

SIMATIC Step7 Bausteine

- OB1 Zyklischer Programmbaustein
- OB82 Profibus Diagnosealarm. Dieser OB wird aufgerufen sobald ein Profibusteilnehmer in der Telegrammantwort das ExtDiagFlag gesetzt hat. Dieses ExtDiagFlag gibt einem Profibusteilnehmer

| | |
|---------------|--|
| | die Möglichkeit einem Profibusmaster einen Fehlerzustand zu melden. Ist der OB82 in der CPU nicht vorhanden, so geht die CPU bei einem gesetzten ExtDiagFlag eines Profibusteilnehmers in den STOP-Zustand |
| OB 86 | Profibus Peripheriefehler. Dieser OB wird aufgerufen, wenn der Profibusmaster den Ausfall eines Profibusteilnehmers erkennt. |
| OB100 | Anlauf-OB. Dieser OB wird beim Anlauf der CPU einmalig ausgeführt. |
| OB 101 | Anlauf-OB. Dieser OB wird beim Anlauf der CPU einmalig ausgeführt. |
| FB1 | FB zum Up- und Downloaden der ASi-Konfigurationsdaten |
| FB2 | FB zum Ausführen von Kommandos über die Mailbox |
| DB1 | Instanzen-DB FB1 |
| DB101 | DB zur Speicherung der ASi-Konfigurationsdaten |
| VAT_Steuerung | Variablen-Tabelle zur Steuerung des Up- und Downloaden sowie Anzeige eines Fehlers bei der Mailboxdiagnose |
| SFC14 | Konsistente Daten eines DP-Normslaves lesen |
| SFC15 | Daten konsistent auf einen DP-Normslave schreiben |
| SFC20 | Kopieren des Inhaltes eines Speicherbereiches in einen anderen Speicherbereich |



Step7 Programmbausteine

Programmablauf

Zwecks Übertragung der ASi-Konfigurationsdaten ruft der OB1 den FB1 auf. Das Up- und Downloaden wird mit Hilfe der Merker Upload und Download in der Variablen-Tabelle gestartet. Der FB1 ist so programmiert, dass sein Ablauf völlig unabhängig von der Hardwarekonfiguration programmiert ist und somit keine Änderungen zur Anpassung in seinem Ablauf vorgenommen werden müssen. Lediglich die Werte an den Eingängen müssen angepasst werden.

Upload der ASi-Konfigurationsdaten

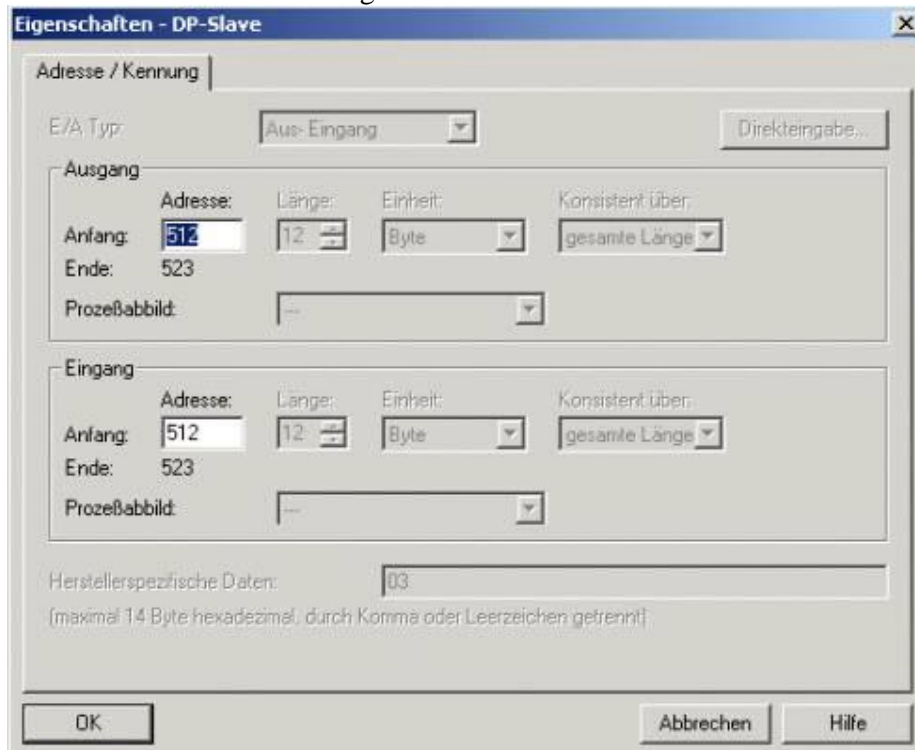
Die Befehle zum Uploaden der ASi-Konfigurationsdaten befinden sich in den Netzwerken 1, 2, 5 und 6 des FB1.

FB1 Netzwerk 1: Die Nummer des DBs zur Speicherung der ASi-Konfigurationsdaten wird in einer lokalen Variablen DBDest und die Nummer des InstanzenDBs wird in der lokalen Variable InstanzenDBDest gespeichert. Anschließend wird der Eingang Kreis des FBs ausgewertet um festzustellen, ob der Kreis 1 oder der Kreis 2 des ASi-Gateways betrachtet werden soll. Je nachdem um welchen Kreis es sich handelt, wird 0 oder 1 in der Variablen zur

Durchführung der ASi-Mailboxbefehle FB2_Kreis gespeichert. Ist der Eingang download bereits gesetzt, so wird upload zurückgesetzt und das Programm setzt im Netzwerk 3 mit dem downloaden der ASi-Konfigurationsdaten fort. Ist dies nicht der Fall, werden beim Anliegen einer positiven Flanke von upload die Anfangszustände eingestellt. Die Variable Zaehler wird mit 0 sowie die Variable P_Byte mit meinem Pointer auf das nullte Byte (P#DBx0.0) geladen. Ist FB2_Busy gesetzt, springt das Programm in das Netzwerk 5 zu dem Aufruf des FB2, FB zur Durchführung der ASi-Mailboxbefehle.

FB1 Netzwerk 2: Ansonsten wird der Wert der Variablen Zaehler mit 31 verglichen, stimmt der Vergleich überein, so wird Zaehler inkrementiert und gespeichert bevor das Programm an die Marke next gelangt. Stimmt der Vergleich nicht, so setzt das Programm direkt an der Marke next fort. Zaehler wird um 1 erhöht und gespeichert. FB2_Befehl wird mit 28_{HEX} und FB2_Parameter1 mit Zaehler geladen. FB2_Start wird gesetzt.

FB1 Netzwerk 5: Der FB2 zum Senden des ASi-Mailboxbefehls und zum Empfangen der Antwort wird aufgerufen. Beim Aufruf des FB2 ASi-Mailbox wird zum Einen die Startadresse des Managementkanals (Command Interface), welche mit dem Eingang StartadresseModul übergeben wird, und zum Anderen die befehlspezifischen Parameter übergeben. Die Startadresse des Managementkanals ist einstellbar.



Startadresse des Managementkanals in der Hardwarekonfiguration

Die Variable FB2_Start startet den FB2. FB2_Busy zeigt die momentane Bearbeitung an und FB2_Complete die fertige Bearbeitung. Ist ein Fehler aufgetreten, erkennt man dies anhand der Variablen FB2_Error und kann FB2_Status den Fehlercode entnehmen.

FB1 Netzwerk 6: Der Datenbaustein, welcher die ASi-Konfigurationsdaten beinhaltet, wird geöffnet. Wenn Zaehler kleiner gleich 63 und upload sowie FB2_Complete gesetzt und FB2_Error nicht gesetzt ist, springt das Programm nicht zur Sprungmarke err. Ansonsten wird die Variable Ziel des Parametertyps ANY zur Speicherung der Daten, welche mit der SFC20 kopiert werden sollen, mit den entsprechenden Werten geladen. Zuerst wird die

Anfangsadresse der ANY-Pointers in AR1 geladen. Die Syntax-ID sowie der Datentyp Byte werden geladen und in den ANY-Pointer transferiert. DA 2 Byte kopiert werden sollen, wird der Wert 2 in LW[AR1,P#2.0] geladen. Die Nummer des Zieldatenbausteins DBDest sowie die Byteadresse, gespeichert in P_Byte, werden geladen. Anschließend wird die SFC20 aufgerufen. FB2_Start wird zurückgesetzt. Das Adressregister 1 wird in SavePointer gesichert und zu P_Byte wird ein Offset von 2 addiert. SavePointer wird nun wieder in das Adressregister 1 geschrieben. Nun wird Zaehler mit 63 verglichen, ist der Vergleich wahr so setzt das Programm bei rset fort ansonsten bei end. An der Sprungmarke rset werden sämtliche Hilfsvariablen zurückgesetzt. Ist während der Bearbeitung des FB2 ein Fehler aufgetreten, wo wird zuvor noch der Ausgang Error gesetzt.

Download der ASi-Konfigurationsdaten

Die Befehle zum Downloaden der ASi-Konfigurationsdaten befinden sich in den Netzwerken 3, 4, 5 und 7 des FB1.

FB1 Netzwerk 3: Ist der Eingang upload gesetzt, wird download zurückgesetzt und der Baustein beendet. Ansonsten werden bei der positiven Flanke von download die Anfangszustände zum Downloaden hergestellt. Zaehler sowie P_Zaehler werden mit 0 geladen, P_LPS mit 1 und P_Bit mit P#DBX112.0. Ist download nicht gesetzt, fährt das Programm an der Sprungmarke end fort. Ansonsten wird der Zustand von FB2_Busy überprüft. Ist FB2_Busy gesetzt, springt das Programm in das Netzwerk 5 zu dem Aufruf des FB2, FB zur Durchführung der ASi-Mailboxbefehle.

FB1 Netzwerk 4: Der DB, in welchem die ASi-Konfigurationsdaten gespeichert werden, wird geöffnet. Ist prjMode gesetzt, springt das Programm zu down. Ist es nicht gesetzt, wird es gesetzt und der ASi-Mailboxbefehl C_{HEX} zum Setzen des Betriebsmodus wird nach FB2_Befehl und der Wert 1 zum Setzen des Projektierungsmodus nach FB2_Parameter1 gespeichert. FB2_Start wird gesetzt und das Programm fährt an der Sprungmarke cFB2 mit dem Aufruf des FB2 fort.

An der Marke down wird Zaehler mit 62 verglichen. Ist Zaehler kleiner gleich, so wird nicht zur Sprungmarke LPS gesprungen. Zaehler wird inkrementiert. 25_{HEX}, ASi-Mailboxbefehl zum projektieren der Konfigurationsdaten, wird nach FB2_Befehl, Zaehler nach FB2_Parameter1 gespeichert. P_Zaehler wird in das Pointerformat umgewandelt und in FB2_Parameter2 sowie mit einem Offset von 1.0 in FB2_Parameter3 gespeichert. Der Wert 2 wird zu P_Zaehler hinzuaddiert. FB2_Start wird gesetzt und es folgt ein Sprung zu cFB2.

Ist LPS gesetzt wird an der Sprungmarke LPS direkt zu comp gesprungen. An der Marke Slav wird der DB, welcher die ASi-Konfigurationsdaten enthält geöffnet. P_LPS wird in das Pointerformat umgewandelt und in das Adressregister 1 kopiert. Das zu der Adresse zugehörige Byte wird mit FF_{HEX} verglichen. P_LPS wird um 2 erhöht. Aufgrund dieser Erhöhung ist zu erkennen, dass nur jedes zweite Byte verglichen wird. Die Extended ID-Codes werden nicht von jedem Slave unterstützt, ist kein Slave vorhanden oder einer, welcher diese Funktion nicht unterstützt, so beinhaltet diese Byte den Wert FF_{HEX}. Jedoch besitzt aber jeder ASi-Slave eine E/A-Konfiguration sowie einen ID-Code, so dass dieses Byte nur den wert FF_{HEX} beinhaltet, wenn der Slave nicht vorhanden ist. Aus diesem Gründen werden lediglich die Werte der E/A-Konfiguration und des ID-Codes überprüft. P-Bit wird in das Adressregister 1 geladen, ein Offset von einem Bit - P#0.1 - wird addiert und der Inhalt des Adressregisters 1 wird wieder in P_Bit kopiert. Der InstanzenDB wird geöffnet. War der Vergleich wahr, so wird das zugehörige Bit zurückgesetzt, ansonsten wird es gesetzt. Stimmt der Wert von P_LPS mit 63 überein, so springt das Programm zur Marke S132, an welcher zu P-Bit mit Hilfe des Adressregisters 1 ein Offset von einem Bit addiert wird, bevor das Programm an der Sprungmarke Slav fortfährt. Diese Addition ist notwendig, da der Platz des

Slaves 0B übersprungen werden muss. Beinhaltet P_LPS nicht den Wert 63, so springt es solange zu Slav bis P_LPS den Wert 125 beinhaltet. Dann springt es zu sLPS. Die Hilfsvariable LPS wird gesetzt. Der Wert 29_{HEX} – ASi-Mailboxbefehl zur nichtflüchtigen Speicherung der Liste der projektierten ASi-Slaves - wird nach FB2_Befehl und der Wert 0 nach FB2_Parameter1 kopiert. Die in dem InstanzenDB gespeicherte LPS wird in die FB2_Parameter2 bis FB2_Parameter9 kopiert. FB2_Start wird gesetzt und das Programm fährt an der Marke cFB2 mit dem Aufruf des FB2 fort.

An der Marke comp angelangt, springt das Programm zu cFB2 wenn die Hilfsvariable complete nicht gesetzt ist. Die Hilfsvariable complete wird gesetzt und der Wert C_{HEX} – ASi-Mailboxbefehl zum Setzen des Betriebsmodus- in FB2_Befehl kopiert. Die Variable FB2_Parameter1 muss nicht mit dem, zum Wechseln in den geschützten Betriebsmodus erforderlichen, Wert 0 geladen werden, da FB2_Parameter1 diesen Wert bereits von der hervorgehenden Zuweisung enthält. FB2_Start wird gesetzt.

FB1 Netzwerk 5: Siehe unter Upload der Konfigurationsdaten FB1 Netzwerk5

FB1 Netzwerk 7: Ist upload oder FB2_Complete gesetzt oder FB2_Error nicht gesetzt, so springt das Programm zur Marke err. Ansonsten wird Zaehler mit 0 verglichen, ist dies wahr, wird FB2_Start zurückgesetzt. Ansonsten folgt kein Sprung zu lete, wenn Zaehler kleiner 63 und complete nicht gesetzt ist. Beinhaltet Zaehler den Wert 31, so wird Zaehler vor dem Rücksetzen von FB2_Start inkrementiert, anderenfalls entfällt das Inkrementieren. Ist an der Sprungmarke lete complete gesetzt, so folgt ein Sprung zu rstD. Anderenfalls wird bei gesetztem FB2_Complete und FB2_Error der Ausgang Error gesetzt. Anschließend folgt das Zurücksetzen, an der Marke rset, sämtlicher während des Programmablaufs benötigter Hilfsvariablen, FB2_Start, upload, download, prjMode, LPS und complete.

Durchführung der ASi-Mailboxbefehle

FB2: Die Fehlervariablen des FB2 werden zurückgesetzt. Durch betätigen von Reset wird der FB2 in seinen Startzustand zurückgesetzt. Die Modulstartadresse wird gespeichert und in das Pointerformat umgewandelt. Der Zeiger P_Out_Data wird initialisiert. Über die SFC14 werden die Daten eingelesen. Ist bei der Durchführung der SFC14 ein Fehler aufgetreten, so wird die Variable SFC14Error gesetzt und das Programm setzt mit der Fehlerauswertung, Sprungmarke Err im Netzwerk 9, fort. Ansonsten wird aus der Befehlsantwort das Togglebit bestimmt. Durch eine Word-Und-Verknüpfung von InData.Ergebnis und 80_{HEX} wird das Togglebit herausgefiltert. Mit Hilfe eines Vergleiches der Und-Verknüpfung mit 80_{HEX} wird die BOOL-Variable TooglebitEcho je nach Zustand des Togglebits gesetzt oder nicht. Ist der Ausgang Busy nicht gesetzt, so wird an der Sprungmarke End2, Ende Netzwerk 5, fortgefahren. TZeitueberwachung wird als Impuls gestartet. BefehlEcho wird mit BefehlSend und TooglebitEcho mit TooglebitSend verglichen. Stimmen die Werte überein, setzt das Programm an der Sprungmarke STA1 mit der Überprüfung des Statusbits fort. Ist dieses 0 wurde der Befehl ohne Fehler beendet, ansonsten wird StatusError gesetzt. Ist der Vergleich nicht wahr, so wird bei gesetzter Zeitueberwachung und nicht gesetzter Tzeitueberwachung TimeoutError gesetzt und mit der Fehlerauswertung im Netzwerk 9 fortgesetzt. Liefert die UND-Verknüpfung keine 1 als Ergebnis wird Zeitueberwachung gesetzt. Wurde der Befehl erfolgreich ausgeführt, StatusOK und Busy gesetzt, so werden die Variablen Zeitueberwachung, Error, Busy und Status zurückgesetzt, bzw. mit 0 geladen und die übermittelten Daten werden von InData in AsiMailboxData kopiert. Complete wird gesetzt. Ist Start gesetzt und Busy nicht, so werden die Parameter für die Profibusübertragung des Befehls zum ASi-Gateway gesetzt. Das Togglebit wird invertiert und in die Ausgangsdaten eingetragen. Start, Complete und Error werden zurückgesetzt und Status sowie AsiMailboxData werden gelöscht. Mit Hilfe der SFC15 werden die Daten von dem

Quellbereich OutData konsistent zu dem ASi-Gateway übertragen. Anhand eines Vergleiches von RET_VAL_SFC15 und 0 wird der Variable SFC15Error gesetzt oder nicht. Beinhaltet die Variable RET_VAL_SFC15 den Wert 0, so wurde die SFC15 fehlerfrei ausgeführt. Im Netzwerk 9 wird gegebenenfalls der aufgetretene Fehler ausgewertet. Start, Busy und Zeitueberwachung werden gesetzt und Complete und Error zurückgesetzt. Nacheinander werden nun StatusError, SFC14Error, SFC15Error und TimeoutError abgefragt und der entsprechende Wert in Status geladen.

Variablentabelle VAT_Steuerung

Mit dem Setzen der Merker Upload bzw. Download wird das Up- bzw. Downloaden der ASi-Konfigurationsdaten gestartet. Der Merker Error zeigt an, ob ein Fehler während dem Übertragen des ASi-Mailboxbefehles, der Durchführung des FB2, aufgetreten ist.

| | Operand | Symbol | Anzei | Statuswert | Steuerwert |
|---|---------|------------|-------|------------|------------|
| 1 | M 20.0 | "Upload" | BOOL | false | |
| 2 | M 20.1 | "Download" | BOOL | false | |
| 3 | M 21.0 | "Error" | BOOL | false | |

Variablentabelle VAT_Steuerung