FAQ PLC Tutorial

Siemens PLC Programming Tutorial with a KEB S6 Drive

Part	Version	Revision	Date	Status
en	1.0	000	2024-11-20	NOT Released

Contents

1. Introduction **2**

2. Setup Profinet **3**

3. KEB Drive Control Data Block **4**

5. State Machine **6**

5.1 Startup OB100 7
5.2 PLC_PRG OB123 7
5.3 State 0 8
5.4 State 1 10
5.5 State 2 11
5.6 State 3 12
5.7 State 4 13
5.8 State 5 14
5.9 State 6 15
5.10 State 99 16
5.11 State 100 17

Disclaimer **18**

FAQ PLC Tutorial

1. Introduction

This document explains how to begin to PLC program within Siemens's TIA Portal software and communicate with a KEB S6/F6 Drive. This is a continuation from the communicaiton setup tutorial previously done by KEB America.

Below is the equipment that was used in this FAQ:

Siemens S7-1200 PLC
TIA Portal V15
KEB Combivert 10S6A32-1100 Drive
Combivis 6 Version 6.7.1.416
KEB USB Serial Adapter 0058060-0040

2. Setup Profinet

This sample project was set up using the standard process data setup with the addition of ru18 in the Transmit PDO's. Disable Transmit PDO's, drag ru18 from the available parameters, and re-enable process data. For further setup, refer to the Profinet Communication Setup FAQ created by KEB America.

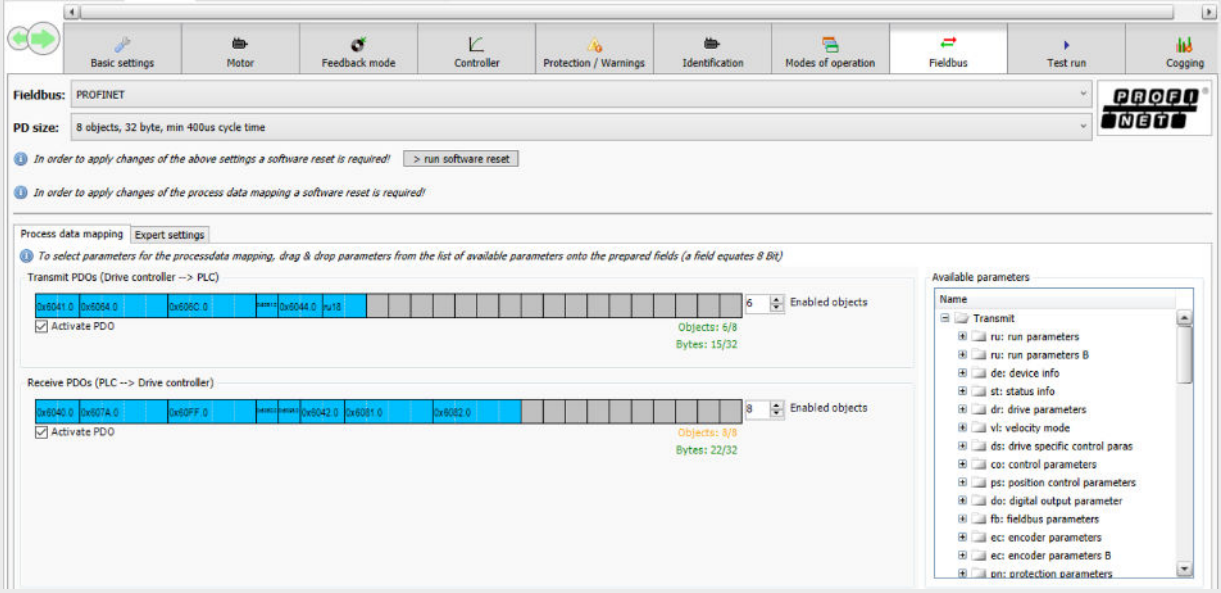


Fig. 1: Fieldbus Setup

3. KEB Drive Control Data Block

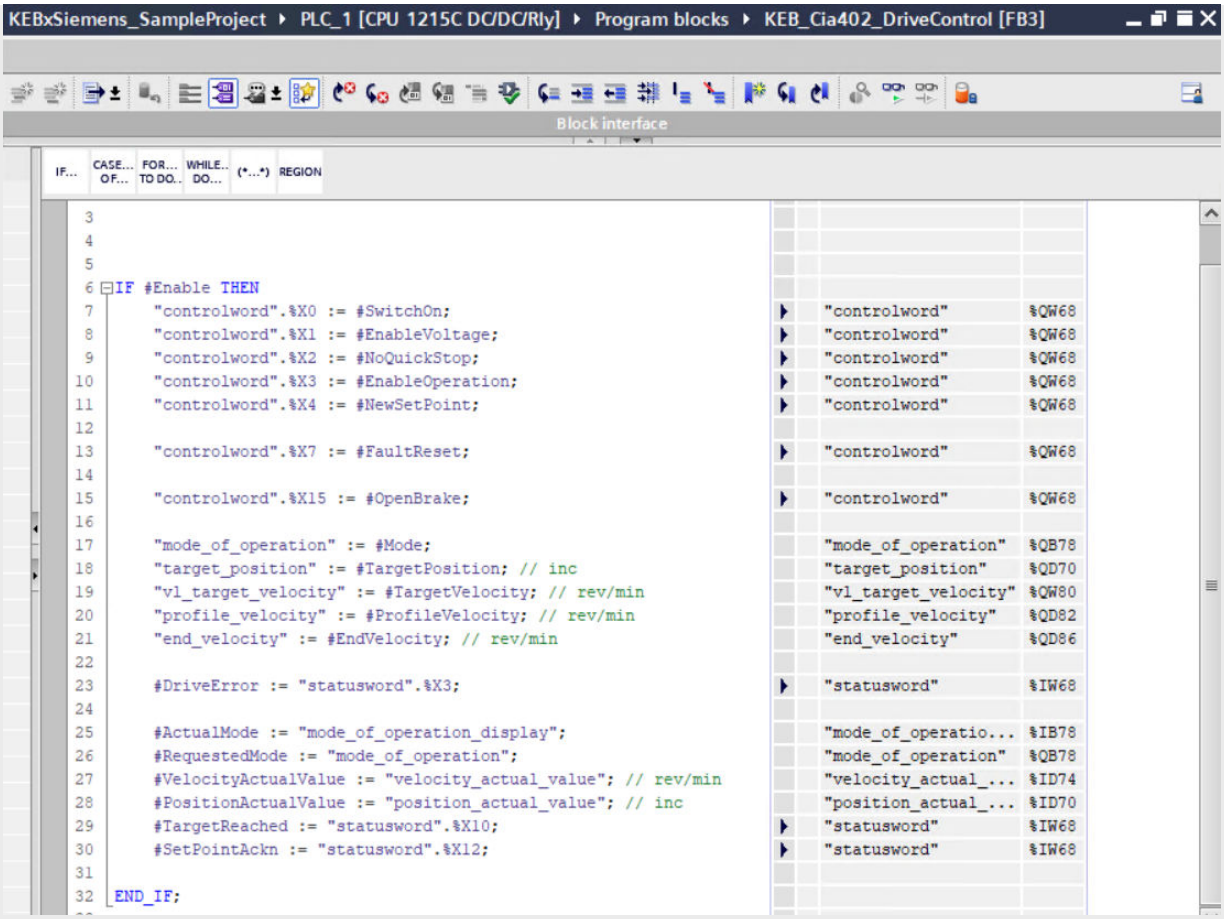


Fig. 2: Drive Control Program Block

A function block has been created that aligns with the CiA402 standard that KEB drives utilize. The most common controlword and statusword bits are implemented in the block. After defining a data block instance of KEB_Cia402_DriveControl, bits of the data type can be written-to and read.

Definition ex.

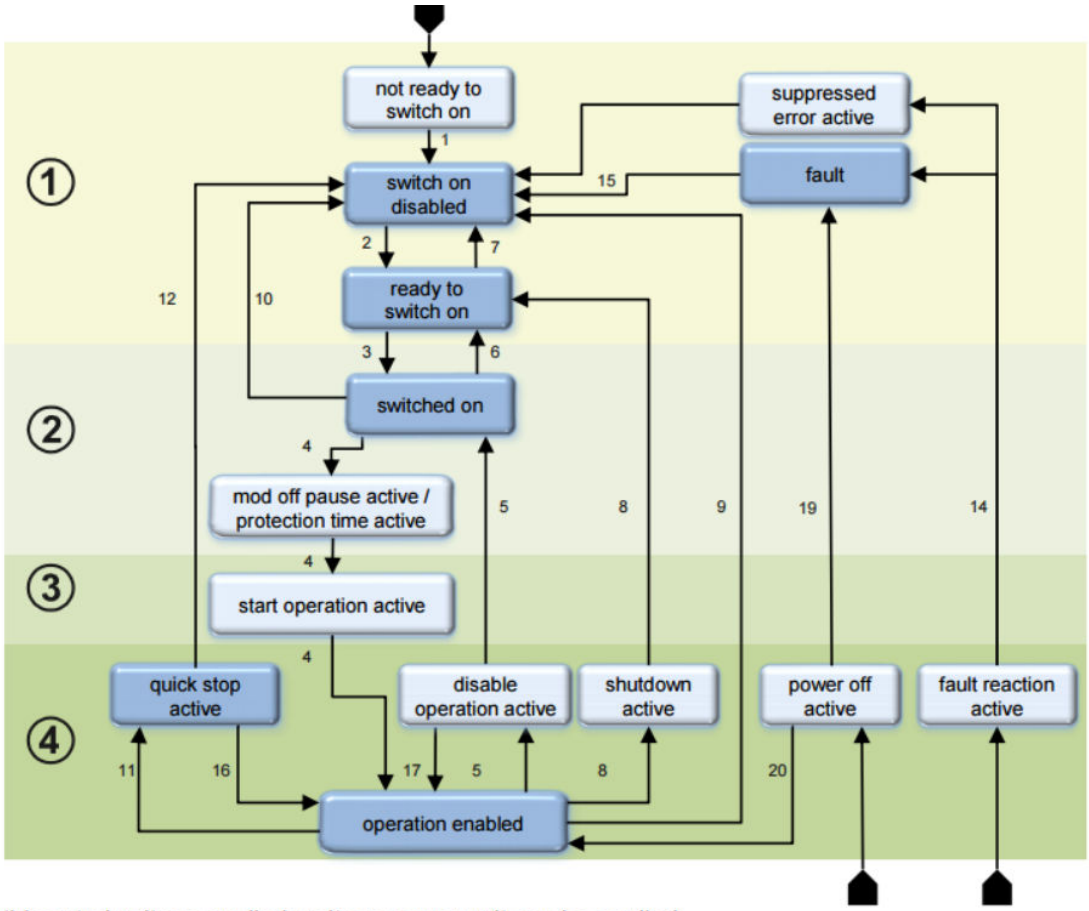
data block name: Drive1
type: KEB_Cia402_DriveControl

Implementation ex.

“Drive1”.EnableOperation := TRUE;
“Drive1”.TargetVelocity := 2000;

```
#CurrentPosition := Drive1.PositionActualValue;
```

5. State Machine



1 24V control voltage applied; voltage power unit can be applied.
2 Voltage power unit applied; no torque at the motor
3 Voltage power unit applied; torque can be applied (flux build-up, speed search etc.)
4 Torque at the motor

Fig. 3: CiA402 State Machine

The drive's internal state machine is controlled via the co00 (CiA 0x6040) controlword and internal events (from errors). The actual state is displayed via st00 (CiA 0x6041) statusword. The actual state can be determined additionally via st12 state machine display. The main path through the state machine that will be covered is 1 to 2, 2 to 3, 3 to 4, and 4 to 17.

5.1 Startup OB100

The startup organization block has a single function block, “IO2MOD”, that sets the HardwareID of the drive. The HardwareID is an important identifier within Siemens programming.

5.2 PLC_PRG OB123

This program cycle organization block contains the state machine. The state machine is controlled by three digital inputs outlined below.

Digital Input I1: HIGH = single step through
Digital Input I2: HIGH = STOP
Digital Input I6: LOW = velocity mode active , HIGH = positioning mode active

5.3 State 0

```
28 CASE "GVL_APP".State OF
29     0:  //PRE-INIT
30
31         //write to parameter examples
32         "co21_Value" := 690;
33         "co21_BrakeControlMode"(enable := TRUE,
34                                 Adress := "co21",
35                                 Subindex := 0,
36                                 Device := "HardwareID",
37                                 Test => #co21_Test,
38                                 Status => #co21_Status,
39                                 Error => #co21_Error,
40                                 cmd := #writing,
41                                 value := "co21_Value");
42
43         "vl05_Value" := 2888;
44         "vl05_VlVelocityMaxAmountFOR"(enable := TRUE,
45                                       Adress := "vl05",
46                                       Subindex := 0,
47                                       Device := "HardwareID",
48                                       Test => #vl05_Test,
49                                       Status => #vl05_Status,
50                                       Error => #vl05_Error,
51                                       cmd := #writing,
52                                       value := "vl05_Value");
53
54         IF #SingleStep THEN
55             "GVL_APP".State := 1;
56             #SingleStep := FALSE;
57         END_IF;
58         IF "Drive1".DriveError THEN
59             "GVL_APP".State := 100;
60         END_IF; //check for fault
```

Fig. 4: State 0

State 0 is the pre-init state and has a few examples of acyclic communication writes. The first step is creating a variable in the “Read/Write Param Values” PLC tag group. Once a variable is created, an instance of Acyclic Communication can be created. Each input of the instance needs to be completed. The test, status, and error variables are created in the “temp” variables within PLC_PRG. The name of the parameter can be entered in the address, and in the “KEB Parameter to Address” PLC tag table, there is a string table with the associated address on the KEB drive. For the subindex, enter the subindex of the drive parameter. The device is using the constant “Hardware ID” found in the Startup organization block. The “cmd” input determines if its a write or a read. For a write, enter

"#writing", if a read, enter "#reading". Finally, enter the variable that the value will be written to/from.

To proceed to the next state, initiate a rising edge of I1. To proceed to the stop state, initiate a rising edge of I2.

5.4 State 1

```
62      1:  //INIT
63          "Drive1".SwitchOn := FALSE;
64          "Drive1".EnableVoltage := FALSE;
65          "Drive1".NoQuickStop := FALSE;
66          "Drive1".NewSetPoint := FALSE;
67          "Drive1".EnableOperation := FALSE;
68
69          "st14_ActiveControlword"(enable := TRUE,
70                                   Adress := "st14",
71                                   Subindex := 16#0,
72                                   Device := "HardwareID",
73                                   Test => #st14_Test,
74                                   Status => #st14_Status,
75                                   Error => #st14_Error,
76                                   cmd := #reading,
77                                   value := "st14_Value"); //read active controlword
78
79          IF "st14_Value" = 0 THEN
80              IF #SingleStep THEN
81                  "GVL_APP".State := "GVL_APP".State + 1;
82                  #SingleStep := FALSE;
83              END_IF;
84          END_IF;
85
86          "CheckForStop"(#Stop);
87          "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 5: State 1

State 1 is the init state, and sets the main controlword bits to false. St14, the active controlword, is being read to verify that there is no controlword bits active. If none are active, the user is allowed to initiate a state step.

In every state, there is a check for a fault or a stop and changes the state accordingly.

5.5 State 2

```
89      2:  //SWITCH ON, VOLTAGE ENABLED, NO QUICKSTOP, PICK MODE
90          "Drive1".SwitchOn := TRUE;
91          "Drive1".EnableVoltage := TRUE;
92          "Drive1".NoQuickStop := TRUE;
93
94          "st14_ActiveControlword"(enable := TRUE, ...); //read active controlword
103
104         IF "st14_Value" = 7 THEN
105             "GVL_APP".StatusText := 'switched on';
106             IF #SingleStep THEN
107                 IF #PositioningModeActive = FALSE THEN
108                     "GVL_APP".State := 3;
109                     #SingleStep := FALSE;
110                 ELSE
111                     "GVL_APP".State := 4;
112                     #SingleStep := FALSE;
113                 END_IF;
114             END_IF;
115         ELSE
116             "GVL_APP".StatusText := 'waiting to switch on';
117         END_IF;
118
119         "CheckForStop"(#Stop);
120         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 6: State 2

State 2 is the 1 to 2 to 3 transition in the Cia402 state machine. By activating the switch on bit, enable voltage bit, and no quickstop bit, the Cia402 state machine is now at the “switched on” state. The drive is still waiting for the enable operation bit to be active for movement. *It is necessary for the enable voltage and switch on bit to be activated before the enable operation bit is activated.*

The mode (velocity or positioning) is determined based off of the value of I6 (set to #PositioningModeActive). A rising edge of I1 will proceed to the corresponding state.

5.6 State 3

```
122     3:  //SET VELOCITY MODE AND VELOCITY
123         "Drive1".Mode := #VelocityMode;
124         "Drive1".TargetVelocity := 100;
125
126         IF "Drive1".ActualMode = #VelocityMode THEN
127             "GVL_APP".StatusText := 'velocity mode enabled';
128             IF #SingleStep THEN
129                 "GVL_APP".State := 5;
130                 #SingleStep := FALSE;
131             END_IF;
132         ELSE
133             "GVL_APP".StatusText := 'waiting to enable velocity mode';
134         END_IF;
135
136         "CheckForStop"(#Stop);
137         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 7: State 3

State 3 is separate from the Cia402 state machine, and sets the “mode of operation” and the “target velocity”, both members of the “Drive1” instance of KEB_Cia402_DriveControl. Once the “actual mode” (drive response) equals the set “mode”, the program state machine waits for a rising edge of I1.

5.7 State 4

```
139     4:  //SET POSITIONING MODE AND PROFILE SPEED
140         "Drive1".Mode := #ProfilePositioningMode; //set profile positioning mode
141
142         "Drive1".ProfileVelocity := 300;
143         "Drive1".TargetPosition := 1000000;
144         "Drive1".EndVelocity := 0;
145
146         "Drive1".EnableOperation := TRUE; //enable operation
147         "Drive1".OpenBrake := TRUE; //open brake
148         "Drive1".NewSetPoint := TRUE; //set setpoint active
149
150         IF "Drive1".SetPointAckn THEN
151             "GVL_APP".StatusText := 'profile positioning mode enabled';
152             "Drive1".NewSetPoint := FALSE; //reset setpoint active
153             "GVL_APP".State := 6;
154         ELSE
155             "GVL_APP".StatusText := 'waiting to enable profile positioning mode';
156         END_IF;
157
158         "CheckForStop"(#Stop);
159         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 8: State 4

State 4 is is the positioning mode jump from 3 to 4 to 17 in the Cia402 state machine, and sets the “mode of operation,” “profile velocity” (speed when positioning), “end velocity” (speed once target has been reached), and “target position,” all members of the “Drive1” instance of KEB_Cia402_DriveControl. Once the “set point” has been acknowledged (drive response), the motor will start to spin. Different from velocity mode, the motor will only spin when the “new set point” bit is set, assuming the enable operation bit has already been set. Once the set point has been acknowledged within the drive, the “new set point” bit can be deactivated, and the program state machine will proceed to the positioning-active state.

5.8 State 5

```
161     5:  //VELOCITY MODE - ENABLE OPERATION AND OPEN BRAKE
162         "Drive1".EnableOperation := TRUE;
163         "GVL_APP".StatusText := 'velocity mode - operation enabled';
164         IF #SingleStep THEN
165             "GVL_APP".State := 99;
166             #SingleStep := FALSE;
167         END_IF;
168
169         "CheckForStop"(#Stop);
170         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 9: State 5

State 5 is the velocity mode jump from 3 to 4 to 17 in the Cia402 state machine. The “enable operation” bit is set, and as soon as that happens, the motor will spin at the given “target velocity.” The motor will spin until a rising edge of I1 or I2 (stop).

5.9 State 6

```
172     6: //POSITIONING MODE ACTIVE
173         "R_TRIG_TargetReached"(CLK := "Drive1".TargetReached,
174                                Q => "GVL_APP".TargetReached);
175
176         IF "GVL_APP".TargetReached THEN
177             "GVL_APP".State := 99;
178         END_IF; //target reached
179
180         "CheckForStop"(#Stop);
181         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 10: State 6

Its helpful to separate the positioning mode states into a set state and a positioning active state. State 6 is the positioning active state, and there is no change in the Cia402 state machine. The PLC waits for a rising edge of the “target reached” bit from the “Drive1” instance of KEB_Cia402_DriveControl. As soon as that happens, the program state machine will proceed to the stop state.

5.10 State 99

```
183     99: //STOP
184         "Drive1".EnableOperation := FALSE;
185         "Drive1".OpenBrake := FALSE;
186         #Stop := FALSE;
187         "GVL_APP".StatusText := 'operation stopped';
188         "GVL_APP".TargetReached := FALSE;
189
190         IF #SingleStep THEN "GVL_APP".State := 1; #SingleStep := FALSE; END_IF;
191         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 11: State 99

State 99 is the stop state. At any point in the program state machine, the stop state can be triggered by a rising edge of I2. At entry of the stop state, the “enable operation” bit and brake bit are deactivated. This is equivalent to the “operation enabled” to “disable operation active” to “switched on” move in the Cia402 state machine. Once the stop state is left and the init state is re-entered, “switch on”, “enable voltage”, and “no quickstop” are all deactivated. Once that happens, the Cia402 state machine will be at “switch on disabled.”

5.11 State 100

```
183     99: //STOP
184         "Drive1".EnableOperation := FALSE;
185         "Drive1".OpenBrake := FALSE;
186         #Stop := FALSE;
187         "GVL_APP".StatusText := 'operation stopped';
188         "GVL_APP".TargetReached := FALSE;
189
190         IF #SingleStep THEN "GVL_APP".State := 1; #SingleStep := FALSE; END_IF;
191         "CheckForDriveFault"("Drive1".DriveError);
```

Fig. 12: State 99

State 100 is functionally very similar to the stop state, except it requires that the drive be free from faults to return to the program init state. The “fault reset” bit is mapped to I5, and if the fault is resettable, it will reset. Once reset, a rising edge of I1 will return to the program init state. This state outlines path 14 and 15 in the Cia402 state machine, from “fault reaction active” to “fault” to “switch on disabled” on returning to the init state.

Disclaimer

KEB America, Inc. reserves the right to change/adapt specifications and technical data without prior notification. The safety and warning reference specified in this manual is not exhaustive. Although the manual and the information contained in it is made with care, KEB does not accept responsibility for misprint or other errors or resulting damages. The marks and product names are trademarks or registered trademarks of the respective title owners.

The information contained in the technical documentation, as well as any user-specific advice in verbal or in written form are made to the best of our knowledge and information about the application. However, they are considered for information only without responsibility. This also applies to any violation of industrial property rights of a third-party.

Inspection of our units in view of their suitability for the intended use must be done generally by the user. Inspections are particular necessary, if changes are executed, which serve for the further development or adaption of our products to the applications (hardware, software or download lists). Inspections must be repeated completely, even if only parts of hardware, software or download lists are modified.

Application and use of our units in the target products is outside of our control and therefore lies exclusively in the area of responsibility of the user.

Americas:

KEB America, Inc.
5100 Valley Industrial Blvd South
Shakopee, MN 55379, USA
(+1) 952-224-1400
info@kebamerica.com

Headquarters:

KEB Automation KG
Suedstrasse 38
D - 32683 Barntrup, Germany
(+49) 5263 401-0
info@keb.de