Training Document for Comprehensive Automation Solutions
Totally Integrated Automation (T I A)

MODULE D14
PROFIsafe with
Master CPU 315F-2 PN/DP and DP Slave ET 200S
This document has been written by Siemens AG for training purposes for the project entitled “Siemens Automation Cooperates with Education (SCE)”. Siemens AG accepts no responsibility for the correctness of the contents.

Transmission, use or reproduction of this document is only permitted within public training and educational facilities. Exceptions require the prior written approval by Siemens AG (Michael Knust michael.knust@siemens.com).

Offenders will be liable for damages. All rights, including the right to translate the document, are reserved, particularly if a patent is granted or utility model is registered.

We would like to thank the following: Michael Dziallas Engineering, the teachers at vocational schools, and all others who helped to prepare this document.
# PREFACE

## Notes on the Use of the CPU 315F-2 PN/DP

## Notes on the Use of the ET 200S with IM 151-1 HF

## Starting Up PROFIsafe with CPU 315F-2 PN/DP and ET 200S

## Program Sample

The following symbols are used as a guide through Module D14:

- **Information**
- **Programming**
- **Sample Exercise**
- **Notes**
1. PREFACE

In terms of its contents, Module D14 is part of the teaching unit entitled 'Industrial Fieldbus Systems'.

Learning Objective:

In Module D14, the reader learns how to start up a safety-oriented application on the PROFIBUS (PROFIsafe). In our case, the CPU 315F-2 PN/DP is used on the PROFIBUS as the master with an ET 200S as slave to monitor the guard door at a press. Emergency Stop is also implemented with the ET200S. Module D14 shows the method in principle, using a brief example.

Prerequisites:

To successfully work through Module D14, the following knowledge is assumed:
- Knowledge in handling Windows
- Fundamentals of PLC programming with STEP 7 (for example, Module A3 - 'Startup' PLC Programming with STEP 7)
- Fundamentals of PROFIBUS DP (for example, Appendix IV - Basics of Fieldbus Systems with SIMATIC S7-300)
Hardware and software required

1. PC, operating system Windows 2000 Professional starting with SP4/XP Professional starting with SP1/Server 2003 with 600MHz and 512RAM, free hard disk storage 650 to 900 MB, MS Internet Explorer 6.0
2. Software STEP7 V 5.4
3. Software S7 Distributed Safety V5.4
4. MPI/PROFIBUS interface for the PC (for example, adapter USB)
5. PLC SIMATIC S7-300 with CPU 315F-2 PN/DP
   Sample configuration:
   - Power supply: PS 307 2A
   - CPU: CPU 315F-2 PN/DP
6. Distributed periphery ET 200S for PROFIsafe with digital inputs and outputs
   Sample Configuration:
   - Interface module IM151-1 HF
   - Power module PM-E DC 24V
   - Digital input module 2DI DC 24V for connecting a button and a switch
   - Digital input module 2DI DC 24V for connecting the feedback circuit of a consumer
   - Digital output module 4DO DC 24V/0.5A for connecting two lamps
   - Power module PM-E DC 24V...48V/AC24V...230V
   - Fault-tolerant digital input module 4/8 F-DI DC 24V for connecting a two-channel Emergency Off and two guard door contacts
     - Fault-tolerant digital output module 4 F-DO DC 24V/2A for connecting a consumer to two contactors K1 and K2 that can be switched off fail-safe separately.
     Here, the consumer is the supply voltage for a press.
7. PROFIBUS connection between IE/PB link and ET200S
8. Emergency Off button with 2-channel wiring at the F-DI module of the ET200S
9. Press with guard door; scan of guard door by means of two contacts wired to the F-DI module
10. Connection lines to the model press and to the emergency off button
Wiring Diagram

- Switch 2: System On
- Button 1: Acknowledge
- Emergency Off
- Protection for contacts
- Feedback Circuit
- Load/Press

Preface | Notes | StartUp | Programming
--- | --- | --- | ---
TIA Training Document | PROFIsafe with Master CPU 315F-2 PN/DP and DP Slave ET 200S
D14 | Issued: 02/2008 | Module |
2. NOTES ON USING THE CPU 315F-2 PN/DP

The CPU 315F-2 PN/DP is a CPU that is shipped with 2 integrated interfaces.

- The first interface is a combined MPI/PROFIBUS-DP interface that can be used on the PROFIBUS DP as master or slave for connecting distributed periphery/field devices with very fast response timing.
  Moreover, the CPU can be programmed here by means of MPI or PROFIBUS DP.
- The second interface is an integrated PROFINET interface.
  It allows for using the CPU as a PROFINET IO controller to run distributed periphery on the PROFINET. The CPU can be programmed by means of this interface also!
- On both interfaces, fault-tolerant IO devices with PROFI safe profile can be used.
- 1-bus concept, transmission with F-signals and standard signals by means of a bus medium (PROFIBUS DP or PROFINET)
- Fault-tolerant IO modules of the ET 200M/S/eco can be connected in the distributed mode.
- Mixed configuration of F-modules and standard modules in one station
- Field devices made by other manufacturers can be connected.
- Standard modules for non-safety related applications can be operated centrally as well as in the distributed mode
- Meets safety requirements up to SIL 3 according to IEC 61508, AK 6 according to DIN V 19250 and Kat. 4 according to EN 954-1
- Standard as well as safety-relevant tasks can be solved with only one CPU

The CPU 315F is based on a standard CPU (F means fault-tolerant here) whose operating system was augmented with different protection mechanisms to make processing safety-oriented user programs possible.

This is needed for configuring a fault-tolerant automation system in plants with heightened safety requirements. The area of application is primarily production engineering. The distributed IO devices ET 200S PROFIsafe with fault-tolerant IO modules can be connected to integrated PROFIBUS DP/PROFINET interfaces as well as to external PROFIBUS/PROFINET CPs. Safety oriented communication takes place by means of PROFIBUS DP /PROFINET with a PROFI safe profile.
Safety Concept

The safety functions of the CPU 315F are included in the F-program and in the fault-tolerant signal modules. The fault-tolerant modules can be used in the distributed IO systems ET200M and ET200S.

The fault-tolerant signal modules monitor output and input signals by means of discrepancy analyses and test signal injections.
The CPU checks the correct operation of the controller through periodic self tests, command tests as well as logical and time-related program run checks. In addition, the IO is checked by requesting life signs.
If an error in the system is diagnosed, the system is brought to a safe state. No F-runtime license is required to operate the CPU 315F.

In addition to the fault-tolerant modules, standard modules can be used.
This makes it possible to set up a fully integrated control system for a plant where, in addition to safety-oriented applications, standard applications also exist.
The entire system is configured and programmed with the same standard tools.

Programming

The CPU 315F is programmed like other SIMATIC S7 systems. The user program for non-fault tolerant parts of the plant is created with the proven STEP 7 programming tools.
To program the safety-oriented programs, the software package “S7 Distributed Safety V5.4” is essential. It includes all elements that you need for the engineering process.

The CPU 315F is programmed with the STEP 7 languages F-LAD and F-FBD. Here, the following safety functions can be implemented, for example:
- User programmable safe combination of sensors and actuators
- Selective safe disconnection of actuators

The functional scope regarding operations and data types is restricted. Through a special input during compilation, a safety-oriented, password protected program is generated. In addition to the fault-tolerant program, a standard program can run on the CPU at the same time (coexistence) that is not subject to restrictions.

The F-library with prepared programming examples -approved by the German Technical Inspectorate- for safety-oriented functions is also part of this software package. The user can change these program examples, but these changes have to be recertified.
Option Package S7 Distributed Safety

The package includes all required functions and blocks for generating the F-program. For “S7 Distributed Safety V5.4“ to run, STEP 7 starting with V5.3+SP3 has to be on the PG/PC.

The F-program with the safety functions is wired in F-FBD or F-LAD, or with special function blocks from the F-library. Using F-FBD or F-LAD simplifies configuring and programming the plant and, because of the system-overreaching uniform representation, it also simplifies certification. The programmer can concentrate completely on configuring the safety-oriented application without having to use additional tools.

Notes:
- In Module D14, the CPU 315F-2 PN/DP is used on the PROFIBUS as master.
- F-modules as well as standard modules are integrated.
- A micro-memory card is needed for running this CPU!
- The addresses for the input and output modules can be programmed at this CPU.
3. NOTES ON USING THE ET200S WITH IM 151-1 HF

The SIMATIC ET 200S is a decentral IO device configured in a highly modular mode. It can be operated with different interface modules:

**IM 151-1 BASIC, IM 151-1 STANDARD** and **IM 151-1 FO STANDARD** for connecting a maximum of 63 IO modules (all types except PROFIsafe) to the PROFIBUS DP; alternatively, bus connection with RS 485 Sub-D connector or by means of integrated fiber-optic connection

**IM 151-1 HIGH FEATURE** for connecting a maximum of 63 IO modules (all types, including the clocked mode for PROFIsafe) to PROFIBUS-DP; bus connection with RS485 Sub-D connector

**IM 151-3 PN** for connecting a maximum of 63 IO modules (all types, including the clocked mode for PROFIsafe) to PROFINET IO controllers; bus connection by means of RJ45 connector

**IM 151-7/F-CPU, IM 151-7/CPU** or **IM 151-7/CPU FO** for connecting a maximum of 63 IO modules (all types; PROFIsafe only with IM151-7/F CPU) to PROFIBUS DP; alternatively bus connection with RS 485 Sub-D connector or by means of an integrated fiber-optic connection; with integrated CPU 314 of the SIMATIC S7-300, for preprocessing process data.

The following IO modules can be used:

- **Power modules** for individual grouping of load and encoder supply voltages and their monitoring
- **Digital electronic modules** for connecting digital sensors and actuators
- **Analog electronic modules** for connecting analog sensors and actuators
- **Sensor module** for connecting IQ sense sensors
- **Technology modules** Electronic modules with integrated technological functions, such as counting, positioning, data exchange, etc.
- **Frequency converters and motor starter modules**

For training purposes, an integrated system is available with which many technologies can be taught.

**Notes:**

- In Module D16, the interface module IM151-1 HF (HIGH FEATURE) is used as PROFIBUS DP slave.
- F-modules as well as standard modules are inserted.
- The PROFIBUS address is set, binary-coded, at 8 switches at interface module IM151-HF. The lowest switch has to be on OFF. A number is assigned to each of the other switches. These numbers add up to the PROFIBUS station address. If the PROFIBUS address is changed, the new setting will be accepted after voltage returns. Therefore, the interface module IM151-1 HF has to be switched off, and then switched on again.
4. STARTING UP PROFISAFE WITH CPU 315F-2 PN/DP AND ET 200S

Below, the startup the PROFIsafe application is shown. The CPU 315F-2 PN/DP is used on the PROFIBUS as master, with an ET 200S as slave, to monitor the guard door at a press. Emergency Stop is also implemented with the ET200S.

1. The central tool in STEP7 is the 'SIMATIC Manager'. It is called here with a double click. (→ SIMATIC Manager)
2. STEP7 programs are managed in projects. We are now setting up such a project (→ File → New)

3. The project is now assigned the name ‘Name’ 'CPU315F_PROFIsafe'. (→ CPU315F_PROFIsafe → OK)
4. Highlight your project and insert a 'PROFIBUS Subnet'. (→ CPU315F_PROFlsafe → Insert → Subnet → PROFIBUS)

5. After you have highlighted your station again, a 'SIMATIC 300 Station' is inserted. (→ CPU315F_PROFlsafe → Insert → Station → SIMATIC 300 Station)
6. With a double click, open the configuration tool for the 'Hardware'. (→ Hardware)
7. Clicking on the symbol opens the hardware catalog. (→)

There, arranged in the following directories:

- PROFIBUS-DP
- PROFIBUS-PA
- PROFINET IO
- SIMATIC 300
- SIMATIC 400
- SIMATIC PC Based Control
- SIMATIC PC Station

all racks, modules and interface modules are provided for configuring your hardware. With a double click, insert 'Rail'. (→ SIMATIC 300 → RACK-300 → Rail)

Then, the configuration table for configuring Rack 0 is displayed automatically.
8. From the hardware catalog, you can now select all modules that are also in your real rack, and insert them in the configuration table.

To this end, click on the name of the respective module, hold the mouse key and drag the module to a line in the configuration table.

We are starting with the power unit 'PS 307 2A'. (→ SIMATIC 300 → PS-300 → PS 307 2A)

**Note:** If your hardware differs from the one displayed here, simply select the corresponding modules from the catalog and insert them in your rack. The order numbers of the individual modules - that are also indicated on the components - are displayed in the footer of the catalog.
9. Next, we drag the 'CPU 315F-2 PN/DP' to the second slot. The order number and the version of the CPU can be read off the front of the CPU. (SIMATIC 300 → CPU-300 → CPU 315F-2 PN/DP → 6ES7 315-2FH10-0AB0 → V2.3)
10. When entering the CPU, the following window appears. By clicking on 'New', you can set up an 'Ethernet' network.

For correct networking, you have to assign an 'IP Address' to the CPU 315F-2 PN/DP and specify the 'Subnet Screen Form'. As an option, a 'Router Address' can be selected for network-overreaching communication. Confirm your inputs with 'OK' (→ New → IP Address: 192.168.1.10 → Subnet Screen Form: 255.255.255.0 → Ethernet(1) → Use Router → Address: 192.168.1.1 → OK)

![Image of network settings window]

**Note:** To process this module, parameter assignment on the Ethernet is not necessary since subsequently, the CPU is accessed by means of a MPI or PROFIBUS.
11. Now, the 'MPI/DP' interface has to be switched to PROFIBUS DP. To this end, double click on the line 'MPI/DP' in the hardware configuration. → MPI/DP

12. Now, select 'PROFIBUS' as 'Type' of interface, and then click on 'Properties' to set the parameters. (→ PROFIBUS → Properties)
13. The window below appears. In the window, assign a ‘PROFIBUS Address’ to the CPU 315F-2 PN/DP and select the ‘PROFIBUS’ net that has already been generated. If you want to change the parameters of the PROFIBUS network, highlight it and then click on ‘Properties’. Confirm your inputs twice with ‘OK’. (→ PROFIBUS Address: 2 → PROFIBUS(1) → Properties → OK → OK)

14. Double clicking on ‘CPU 315F-2 PN/DP’ opens its property catalog. (→ CPU 315F-2 PN/DP)
15. Under the tab 'Protection', do the following for setting the 'Protection Level':

- Below the option button '1: Access protection for F-CPU', select the option 'Can be canceled with password'
- Activate the option button '2: Write protection'
- Below the option button '3: Write/read protection', enter a password consisting of 8 characters maximum for the F-CPU; for example, "pw_fcpu". Repeat your input in the field 'Repeat input'.
- Activate the option box 'CPU contains Safety Program'.

→ Can be canceled with password → Write protection → pw_fcpu → pw_fcpu → CPU contains safety program

![Diagram showing settings for protection levels](image-url)
16. Under the tab 'Cycle/Clock flag', perform the following settings:
   - At "OB85-call if there is an IO access error", select "Only for first up and cleared errors".
   - Activate the option box 'Clock Flag' and enter the '0' as flag byte.

   \[\rightarrow \text{Cycle/Clock Byte} \rightarrow \text{Only for first up and cleared errors} \rightarrow 0\]

17. Change to the tab 'Time Interrupts' and set the call time for the time interrupt OB35. (In the time interrupt OBs, the safety program is called at fixed time intervals.)

   \[\rightarrow \text{Time Interrupt} \rightarrow \text{OB35} \rightarrow 50\]
18. Go to the tab 'F-Parameters' and set the following parameters:

- The basis for the PROFIsafe addresses
- A number band for F-data blocks
- A number band for F-function blocks
- The local data amount used by the F-system

Confirm with 'OK'. (→ F-Parameters → OK)

Note: When the safety program is generated, F-blocks are added automatically, in order to generate a runnable safety program. For these automatically added F-blocks, a number band has to be reserved here.

19. Now, close the message window for the required 'New generation of the safety program'. (→ Close → OK)
20. After you have accepted the network settings and the parameters for the 'CPU 315F-2 PN/DP', to the right of the CPU315F-2 a bar appears - the 'DP Master System' - where you can arrange the PROFINET slaves. To do this, click on the desired module (here the 'ET 200S' with 'IM151-1 HF') in the hardware catalog in the path 'PROFIBUS-DP' and drag the module to the 'DP Master System'.

(→ PROFIBUS-DP → ET 200S → IM151-1 HF)

21. When entering the slave, the window below appears. In it, you have to assign a PROFIBUS address to the slave. It has to be identical with the one that you set at the switches of the ET 200S. (→ 3 → OK)
22. By double clicking on 'IM151-1 HF', its property catalog is opened. (→ IM151-1 HF)

23. Under the tab 'Operating parameters', keep the standard settings as shown below. Clock synchronization is not used. (→ Operating parameters → OK)
24. Now, the submodules inserted in the ET200S have to be moved with Drag&Drop to the configuration table. We are starting with the power module 'PM-E DC24V' which is dragged to Slot 1. By double clicking on the 'PM-E DC24V', you are opening its properties. (→ PROFIBUS-DP → ET 200S → IM151-1 HF → PM → PM-E DC24V → PM-E DC24V)

25. Under parameters, activate 'Diagnosis: Load voltage L+ missing' with a ☑. (→ Diagnosis: Load voltage L+ missing → ☑ → OK)
26. Then, we drag the digital input module '2DI DC24V ST' to the 2nd and 3rd slot. The order number and the version can be read off the module. (→PROFIBUS-DP → ET 200S → IM151-1 HF → DI → 2DI DC 24V ST → 2DI DC 24V ST)

27. Then, we drag the digital output module '4 DO DC24V/0.5A ST' to the 4th slot. The order number and the version can be read off the module. (→PROFIBUS-DP → ET 200S → IM151-1 HF → DO → 4 DO DC24V/0.5A ST)
28. Now, another power module 'PM-E DC24...48V/AC24...230V' is moved to the 5th slot. The order number and the version can be read off the module. By double clicking on 'PM-E DC24...48V/AC24...230V', you open its properties. (→ PROFIBUS-DP → ET 200S → IM151-1 HF → PM → PM-E DC24...48V/AC24...230V → PM-E DC24...48V/AC24...230V)

29. Under Parameters, activate 'Diagnosis: Load voltage L+ missing' with a ✔. (→ Diagnosis: Load voltage L+ missing → ✔ → OK)
30. Next, we are dragging the fault-tolerant digital input module '4/8 F-DI DC24V' to the 6th slot. The order number and the version can be read off the module. By double clicking on '4/8 F-DI DC24V', you open its properties. (PROFIBUS-DP → ET 200S → IM151-1 HF → DI → 4/8 F-DI DC24V → 4/8 F-DI DC24V)

31. Under the tab 'Addresses', you can change the module’s address areas. However, it has to be ensured that identical values are assigned to the start addresses of the output and input areas. (Addresses → 200 → 200)
32. Under the tab 'Parameters', you can change the following parameters:

- F-Parameters for PROFIsafe
- Module parameters
- Channel-specific parameters

Here, a two channel emergency stop switch is to be connected to channels 0 and 4, and the position switches for monitoring a 2-channel guard door to channels 1 and 5. Perform the following settings, and then accept them with 'OK'. (→ Parameter → OK)
Note regarding "F-Parameters"

The PROFIsafe addresses have to be unique network-wide as well as station-wide. To prevent wrong parameter assignments, the addresses are assigned automatically. The PROFIsafe 'F_Destination_Address' has to be set at the F-module with a DIL switch. The F-CPU provides the PROFIsafe 'F_Source_Address' (F-Parameter 'Basis for PROFIsafe addresses'). During the F-monitoring time, a valid current safety message has to be received from the F-CPU. Otherwise, the F-module enters the safe mode.

On the one hand, the F-monitoring time should be so high that message delays are tolerated, on the other hand so low that the process can respond as fast as possible if there is an error, and is executed without detriment. Calculation tables that SIEMENS makes available on the Internet provide support for determining the timing. (http://www4.ad.siemens.de/ww/view/de/ under the Contribution ID 19138505)

Note regarding "Module Parameters"

For a cyclical short circuit test, the internal encoder supplies have to be used for all encoders connected to the F-module, and channels that are not used have to be deactivated. Otherwise, errors are detected on these channels. For our example, the settings of the module parameters should not be changed.

Note regarding "Channel x, y" parameters: 'Encoder Evaluation' and 'Type of Encoder Interconnection' have to parameterized according to the encoder wiring. The encoder wiring and the safety quality of the encoder are decisive for the attainable safety class. Deactivate the unused channels 2, 6 and 3, 7.

Note regarding 2of2 evaluation, Discrepancy performance and discrepancy time

If different levels (when checking for non-equivalence: same levels) are ascertained for two associated input signals ('2of2 evaluation' of the encoders), the 'Discrepancy time' that can be parameterized here starts. While the discrepancy time is running within the module, the affected input channel makes the 'last valid value' or '0' available to the F-CPU, depending on the parameter assignment of the discrepancy performance.

33. Now, close the message window for the necessary 'New generation of the safety program'.

(→ Close → OK)
34. Next, we are dragging the fault-tolerant output module ‘4 F-DO DC24V/2A’ to the 7th slot. The order number and the version can be read off the module. By double clicking on the ‘4 F-DO DC24V/2A’, its properties are opened. (→PROFIBUS-DP → ET 200S → IM151-1 HF → DO → 4 F-DO DC24V/2A → 4 F-DO DC24V/2A)
35. In addition to the 'Addresses' that are changed here to 210, the following parameter values can be changed under the tab 'Parameter':

- F-Parameter for PROFIsafe
- Module parameters/channel-specific parameters

Here, on Channel 0, the press -our consumer- is to be enabled indirectly by means of two contactors.

Perform the following settings, and then accept them with 'OK'. (→ Parameter → OK)

**Note regarding "DO Channel x" Parameters**

Each output channel has its own parameterizable readback time. This time specifies the maximum duration of the shut down test for the corresponding channel, and thus also the readback time for the shut down cycle of the channel. A wire break test is used for monitoring the connection from the output to the consumer. Deactivate channels that are not used.
36. Now close the message window for the required 'New generation of the safety program'. (→ Close → OK)

37. By clicking on , the configuration table is saved and compiled. Then, by clicking on , the configuration can be loaded to the PLC. The operating mode switch at the CPU should be on Stop! (→)

Note: Make sure that your programming device is connected to the CPU by means of the MPI.
38. The CPU 315F-2 PN/DP is confirmed as destination module for the loading process. (→ OK)

39. In the dialog window below, the connected devices in the network can be 'Display'ed. The CPU's MPI address in the MPI network is then selected. If you are connected to only one CPU, accept with 'OK'. (→ Display → OK)
5. PROGRAM EXAMPLE

In the safety program below, a press as Consumer 1 is to be shut down at a safety-related production area if:

- a guard door that is monitored by two contacts is opened, or
- an emergency stop that is connected to two channels is operated

After operating the emergency stop or after opening the guard door, a local user acknowledgement is required to restart production.

In our example, the following is to be programmed, and generated into a safety program: a fault-tolerant block with a guard door function, an emergency stop function (safety circuit for switch-off in the case of emergency stop and if the guard door is open), a feedback circuit (as reclosing protection if the consumer is faulty), and a user acknowledgement for reintegration.

The precondition for programming is a hardware configuration that is set up correctly, as described in Chapter 4.

F-Periphery Data Blocks
For each F-periphery, an "F-Periphery DB" is generated automatically in the hardware configuration during compilation, and a symbolic name is entered for it at the same time in the symbol table. You can view the F-periphery DBs generated for the sample periphery in the block container. These are the F-data blocks DB 601 and DB 602.

The symbolic name for the F-periphery DB is generated from the following: the fixed prefix "F", the start address of the F-periphery, and the name entered in the hardware configuration in the object properties for F-periphery (17 characters maximum).

F-Global DB
The F-global DB "DB 600" is a fault-tolerant data block that is inserted automatically. It contains all the global data of the safety program as well as additional information that the F-system needs.
Inputs and Outputs in the Safety Program

For programming the sample safety program, the following addresses and fault-tolerant periphery DBs are available to you, according to the hardware configuration described in Chapter 4.

You can access the variables of the F-periphery DB by means of a 'fully qualified DB access' (that is, by entering the symbolic name of the F-periphery DB and the name of the variable).

<table>
<thead>
<tr>
<th>Configured Hardware</th>
<th>Start Address</th>
<th>Symbolic Name</th>
<th>F-Periphery DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Electronic Module 2DI DC24V ST (6ES7 131-4BB01-0AA0)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Electronic Module 2DI DC24V ST (6ES7 131-4BB01-0AA0)</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Electronic Module 4DO DC24V/0.5A ST (6ES7 132-4BD01-0AA0)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault-Tolerant Digital Input Module 4/8 F-DI DC24V (6ES7 138-4FA01-0AB0)</td>
<td>200</td>
<td>F00200_4_8_F_DI_DC24V</td>
<td>DB 601</td>
</tr>
<tr>
<td>Fault-Tolerant Digital Output Module F-DO DC24V/2A (6ES7 138-4FB01-0AB0)</td>
<td>210</td>
<td>F00210_4_F_DO_DC24V_2A</td>
<td>DB 602</td>
</tr>
</tbody>
</table>

1. Now, open the symbol table in your project. (→ Symbols)
2. In the symbol table, assign symbolic names for the fault-tolerant and the standard inputs and outputs as well as for the flags used. Save the symbol table and close it. (→ [Image] → [Image])

3. In the SIMATIC Manager, set up an F-FB. (→ Blocks → Insert New Object → Function Block)
4. Under the tab 'General – Part 1', enter the 'Name', and the 'Symbolic Name'. As programming language, select 'Programming Language F-FBD' and close the dialog field with 'OK'. (→ FB1 → FB_MAIN PROGRAM → F-FBD → OK)

5. Open 'FB1' by double clicking in the SIMATIC Manager. (→ FB1)
6. Enter the 8-character password twice, and accept with 'OK'. (\(\rightarrow\) pw_fprog \(\rightarrow\) pw_fprog)

7. Now, in the editor 'LAD/STL/FBD', set up a static variable with the name 'ENA_GUARDDOOR' (enable guard door). (\(\rightarrow\) STAT \(\rightarrow\) ENA_GUARDDOOR)

**Note:**
The programming languages F-FBD and F-LAD basically correspond to the standard FBD/LAD. For programming, the standard *FBD/LAD Editor in STEP7* is used.
F-FBD and F-LAD differ from the standard essentially through limitations in the operation set, and the useable data types and operand areas.
The following is listed in the program element catalog:
- The operations that are supported
- F-FBs and F-FCs from the block container of your S7 program
- F-blocks from F-libraries, such as F-application blocks of the F-library *Distributed Safety* (V1), for guard door surveillance etc.
- Multi-instances
8. Insert the fault-tolerant application block "FB 217 "F_SFDOOR" (guard door surveillance) from the block container 'F-Application Blocks', set up the instance DB and initialize the inputs and outputs, as shown in the figure below. (→ Libraries → F-Application Blocks → FB217 → DB217 → Yes)

The non-safety related "acknowledgement button" signal from the standard program is colored yellow.

**Notes:** If you need the Boolean constants "0" and "1" in your safety program to initialize parameters for block calls, access the variables "RLO0" and "RLO1" in the F-Global DB by means of a fully qualified DB access ("F_GLOBDB".RLO0 or "F_GLOBDB".RLO1). In our example, the F-Global DB has the number "DB 600" in the block container.

The enable input EN and the enable output ENO must not be wired, initialized with "0" or evaluated in the case of fault-tolerant programming!
### FB 217 Connections:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;HV01-S210&quot;</td>
<td>IN1</td>
<td>BOOL</td>
<td>Guard door contact 1</td>
<td>0</td>
</tr>
<tr>
<td>&quot;HV01-S210&quot;</td>
<td>IN2</td>
<td>BOOL</td>
<td>Guard door contact 1</td>
<td>0</td>
</tr>
<tr>
<td>&quot;F00200_4_8_F_DI_DC24V&quot;.QBAD</td>
<td>QBAD_IN1</td>
<td>BOOL</td>
<td>QBAD signal from F-Periphery DB of Input IN1*</td>
<td>0</td>
</tr>
<tr>
<td>&quot;F00200_4_8_F_DI_DC24V&quot;.QBAD</td>
<td>QBAD_IN2</td>
<td>BOOL</td>
<td>QBAD signal from F-Periphery DB of input IN2*</td>
<td>0</td>
</tr>
<tr>
<td>&quot;F_GLOBDB&quot;.RLO1</td>
<td>OPEN_NEC</td>
<td>BOOL</td>
<td>Fully qualified access to variable RLO1 from F-Global DB**</td>
<td>1</td>
</tr>
<tr>
<td>&quot;F_GLOBDB&quot;.RLO1</td>
<td>ACK_NEC</td>
<td>BOOL</td>
<td>Fully qualified access to variable RLO1 from F-Global DB**</td>
<td>1</td>
</tr>
<tr>
<td>&quot;HV01-S220&quot;</td>
<td>ACK</td>
<td>BOOL</td>
<td>User acknowledgement (with button)</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>#ENA_GUARDDOOR</td>
<td>Q</td>
<td>BOOL</td>
<td>Enable Guard door</td>
<td>0</td>
</tr>
<tr>
<td>ACK_REQ</td>
<td>BOOL</td>
<td>Acknowledgement request</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DIAG</td>
<td>BYTE</td>
<td>Service information</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

---

* = Both inputs QBAD_IN1 and QBAD_IN2 have to be wired; in our example both with the QBAD signal from the F-Periphery DB of the 4/8 F-DI to which the guard door position switches are connected. The block number of the F-periphery DB is provided by means of the symbolic name in the symbol table, or in the SIMATIC Manager.

** = OPEN_NEC: 1 = OPEN is required at startup/ACK_NEC: 1 = acknowledgement required

9. Set up the static variables "ENA_SafetyCircuit" (enable safety circuit), "HM01" and "HM02" as auxiliary flags. Insert a 'New network' and generate a program for the safety circuit as shown below. (→ ENA_SafetyCircuit → HM01 → HM02 → New Network)
10. In the 'SIMATIC Manager', open the F-library Distributed Safety (V1) and copy the F-application block F_TOF (FB 186) from the block container 'F-Application Blocks' to the block container of your S7 program. It is needed by the following fault-tolerant application block. (→ FB186)

11. In your project, open the 'Object Properties' of FB 186. (→ FB186 → Object Properties)
12. Assign the symbolic name 'F_TOF' to the F-application block FB 186. (→ F_TOF → OK)
13. Set up a 'New Network' and insert the fault-tolerant FB 216 "F_FDBBACK" (feedback circuit monitoring) from the block container 'F-Application Blocks'. Set up the instance DB, and initialize the inputs and outputs as shown in the figure below. (→ New network → Libraries → F-Application Blocks → FB216 → DB216 → Yes)

Notes: If you need the Boolean constants "0" and "1" in your safety program to initialize parameters for block calls, access the variables "RLO0" and "RLO1" in the F-Global DB by means of a fully qualified DB access ("F_GLOBDB".RLO0 or "F_GLOBDB".RLO1). In our example, the F-Global DB has the number "DB 600" in the block container.

The enable input EN and the enable output ENO must not be wired, initialized with "0" or evaluated in the case of fault-tolerant programming!


### FB 216 Connections

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Switch_On&quot; and #ENA_SafetyCircuit</td>
<td>ON</td>
<td>BOOL</td>
<td>Enable conditions for Consumer 1/Press</td>
<td>0</td>
</tr>
<tr>
<td>&quot;HV01-S222&quot;</td>
<td>FEEDBACK</td>
<td>BOOL</td>
<td>Readback input</td>
<td>0</td>
</tr>
<tr>
<td>&quot;F00210_4_F_DO_DC24V_2A_.QBAD&quot;</td>
<td>QBAD_FIO</td>
<td>BOOL</td>
<td>QBAD signal from F-Periphery DB of output Q*</td>
<td>0</td>
</tr>
<tr>
<td>&quot;F_GLOBDB&quot;.RLO1</td>
<td>ACK_NEC</td>
<td>BOOL</td>
<td>Fully qualified access to variable RLO1 from the F-Global DB**</td>
<td>1</td>
</tr>
<tr>
<td>&quot;HV01-S220&quot;</td>
<td>ACK</td>
<td>BOOL</td>
<td>User acknowledgement (with button)</td>
<td>0</td>
</tr>
<tr>
<td>T#1s</td>
<td>FDB_TIME</td>
<td>TIME</td>
<td>Readback time</td>
<td>T#0ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;HV01-H100&quot;</td>
<td>Q</td>
<td>BOOL</td>
<td>Consumer 1 / Press</td>
<td>0</td>
</tr>
<tr>
<td>ERROR</td>
<td>BOOL</td>
<td>Readback error</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ACK_REQ</td>
<td>BOOL</td>
<td>Acknowledgement request</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DIAG</td>
<td>BYTE</td>
<td>Service information</td>
<td>B#16#0</td>
<td></td>
</tr>
</tbody>
</table>

* = in our example, this is the QBAD signal from the F-Periphery DB of the F-DO to which the consumer (the contactors) is connected. The block number of the F-periphery DB is provided by means of the symbolic name in the symbol table, or in the SIMATIC Manager.** = ACKN_NEC: 1 = acknowledgement required.
14. Set up the static variable "HM03" as auxiliary flag. Insert a 'New Network' and generate in your safety program for each F-periphery a user acknowledgement for re-integration by means of the variable ACK_REI of the respective F-Periphery DB, as shown in the figure below. Then, save FB1500. (→ HM03 → New Network →)

Note: For re-integrating the F-Periphery (that is, for switching from substitute values (0) to process values) after the errors are remedied, a user acknowledgement with a positive edge at the variable ACK_REI of the F-periphery DB is required:
- always after communication errors
- after F-periphery/channel errors only if parameter assignment ACK_NEC = 1.

15. Acknowledge the following message with 'Yes' and then close FB1 and the FBD/LAD Editor. (→ Yes →)

Note: For re-integrating the F-Periphery (that is, for switching from substitute values (0) to process values) after the errors are remedied, a user acknowledgement with a positive edge at the variable ACK_REI of the F-periphery DB is required:
- always after communication errors
- after F-periphery/channel errors only if parameter assignment ACK_NEC = 1.
16. Now, open the safety program. (→ Blocks → Options → Edit Safety Program)

17. Then, click on 'F-Sequence Groups'. (→ F-Sequence Groups)
18. Click on 'New', specify a 'New F-Sequence Group' and perform the following settings:

- Specify "FC1" as F-call block F-CALL for the new sequence group. This FC is set up automatically as soon as you exit the dialog field "Edit F-sequence group" with "OK".
- Specify the F-program block for the F-sequence group by selecting from the drop down menu the previously programmed F-FB that you want to specify as F-program block for the F-sequence group; in our example, "FB1".
- Since in our example, the F-program block is a function block, assign an instance DB to it (for example, "DB 1"). This I-DB is set up automatically, as soon as you exit the dialog field "Edit F-sequence groups" with "OK".
- As the maximum cycle time for the F-sequence group, set "200ms". Accept with 'OK'. (→ New → FC1 → FB1 → DB1 → 200ms → OK)

Notes: The F-CALL is the F-block for calling the F-sequence group from the standard user program. The F-CALL contains the call for the F-program block and the calls for the automatically added F-blocks of the F-sequence group. Although you are setting up the F-CALL, you can not edit it. The F-program block is an F-FC or F-FB (with instance DB) that changes into an F-program block by being assigned to the F-CALL. You can do the following in the F-program block:

- Program the safety program with F-FBD or F-LAD
- Call additional F-FBs/F-FCs that were created to structure the safety program
- Insert F-blocks of the block container F-Application Blocks
- Insert F-blocks from "User-generated F-libraries"

Within the F-program blocks, you determine the call sequence of the F-blocks.
19. After you have confirmed the automatic generation of the missing blocks with 'Yes', return to the dialog field "Edit F-sequence groups", which now looks like the screenshot below. Close it with 'OK'.

(→ Yes → OK)

20. The safety program is prepared, but not generated yet. The total signature of all F-blocks with the F-attribute of the block container and the total signature of the safety program differ. Now, 'Generate' your safety program. (→ Generate)
21. When the safety program is generated, the consistency of the sequence-relevant F-blocks is checked; that is, the safety program is checked for errors. Possible error messages are read out in an error window. After a successful consistency check, the F-system blocks that are needed in addition are generated automatically, and added to the sequence group, in order to generate a runnable safety program. Now, 'Close' the message list. (→ Close)

22. After the generation was successful, a consistent safety program is always in the block container. This safety program consists of all F-blocks with F-attribute. Now, the 'Total signature of all F-blocks with F-attribute of the block container' and the 'Total signature of the safety program' are the same. We now have a consistent safety program and a safety program that is ready for approval. Confirm with 'Close'. (→ Close)
23. We enter the safety program by calling the FC1 "F-CALL" from a time interrupt OB. To this end, set up OB35 in the SIMATIC Manager. (→ Blocks → Insert New Object → Organization Block → OB35 → OK)

**Note:** Time interrupt OBs have the following advantage: they interrupt cyclical program processing in OB1 of the standard user program at fixed time intervals. That means, in a time interrupt OB, the safety program is called and run in fixed time intervals. After the safety program is processed, the standard user program continues to be processed.
24. Open 'OB35' by double clicking in the SIMATIC Manager. (→ OB35)

25. Call FC1 "F-CALL" in the time interrupt OB 35 as shown in the figure. Save and close OB35.

(→ Call FC 1 → ☒ → ☒)
26. Open 'OB1' with a double click in the SIMATIC Manager and select 'FBD' as 'Programming Language'. (OB1 → FBD → OK)
27. Program the activation of flag M10.0 for the operational enable of the press from the standard program. Save and close OB1. (→ → )
28. Switch the interface for loading the safety program to PROFIBUS. (→ Options → Set PG/PC interface → PC-Adapter (PROFIBUS) → Settings → OK → OK)
29. Open the safety program again. (→ Blocks → Options → Edit safety program)

![Image of SIMATIC Manager with menu options]

30. Click on the button 'Load'. (→ Load)

![Image of SIMATIC Manager with safety program loaded]

<table>
<thead>
<tr>
<th>Preface</th>
<th>Notes</th>
<th>StartUp</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIA Training Document</td>
<td>Page 58 of 60</td>
<td>Module</td>
<td>PROFIsafe with Master CPU 315F-2 PN/DP and DP Slave ET 200S</td>
</tr>
<tr>
<td>Issued: 02/2008</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
31. Confirm the query whether the F-CPU is to be switched to the operating mode STOP with 'Yes'. (→ Yes)

32. Confirm the query whether the standard blocks are also to be loaded with 'Yes'. (→ Yes)

33. Enter the password. (→ pw_fcpu → OK)

34. Confirm the query whether the CPU is to be started with 'Yes'. (→ Yes)

Notes: If you only load the F-blocks, the block in which the F-CALL block is called (in our example, time interrupt OB35) will not be loaded. You will have to load this OB separately, as in the standard program.
The complete safety program can only be loaded in the STOP mode.
35. In the dialog field 'Safety program', activate successively the optional buttons 'Offline' and 'Online' and check whether the total signatures of all F-blocks with F-attribute of the block container match online and offline. If they match, loading the safety program was successful. If not, repeat the load process. (→ Online → Offline → Close)

![Safety program dialog field](image)

36. To activate the safety mode, perform a STOP/RUN transition of the F-CPU.

**Note:** After a safety program is generated, you have to perform a complete function test according to your automation task.