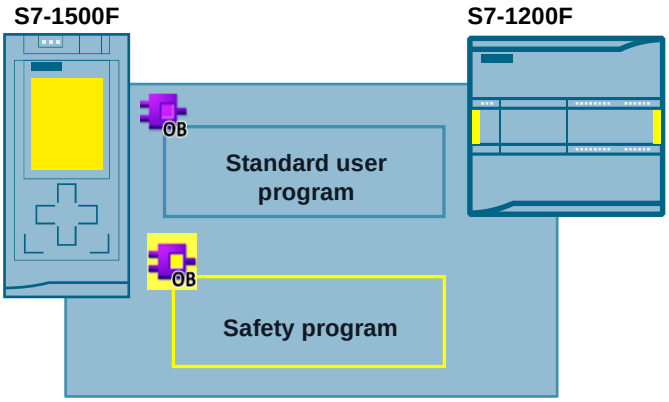
5 STEP 7 Safety in the TIA Portal

5.1 Introduction

TIA Portal V13 SP1 or higher are supported by fail-safe S7-1200F and S7-1500F CPUs. In these controllers, standard as well as fail-safe programming is possible in one device. For programming the fail-safe user programs, the SIMATIC STEP 7 Safety (TIA Portal) option package is used.

Figure 5-1: Standard and safety program



Advantages

- Uniform programming in standard and fail-safe program with an engineering tool: TIA Portal
- Familiar programming in LAD and FBD
- Uniform diagnostic and online functions

Note

Fail-safe does not mean that the program contains no errors. The programmer is responsible for the correct programming logic.

Fail-safe means that the correct processing of the fail-safe user program in the controller is ensured.

Note

Further information on the topic of safety, such as safety requirements or the principles of safety programs can be found in:

TIA Portal - An Overview of the Most Important Documents and Links - Safety
https://support.industry.siemens.com/cs/ww/en/view/90939626

Applications & Tools – Safety Integrated
https://support.industry.siemens.com/cs/ww/en/ps/14675/ae

STEP 7 Safety (TIA Portal) - Manuals
https://support.industry.siemens.com/cs/ww/en/ps/14675/man

5.2 Terms

This document consistently uses the terms with the following meaning.

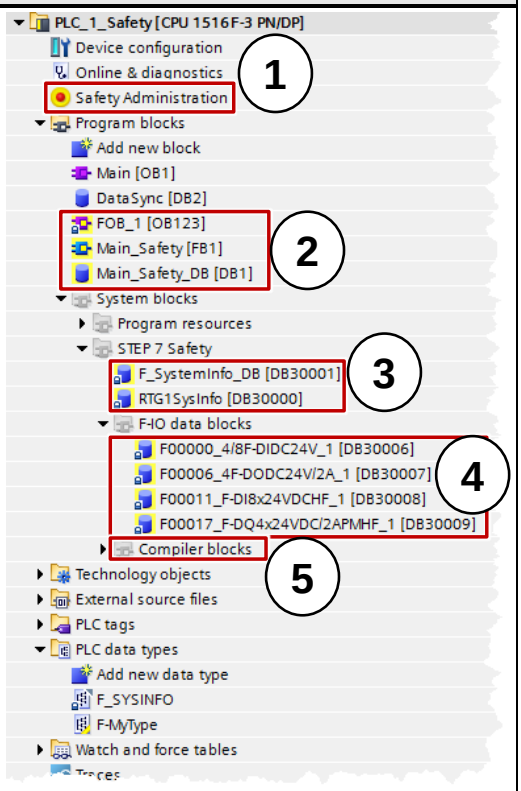
Table 5-1: Safety terms

Term	Description
Standard user program	The standard user program is the program part, which is not connected with F programming.
Safety program (F program, failsafe user program)	The fail-safe user program is the program part which is processed fail-safe independently of the controller. All fail-safe blocks and instructions are shaded yellow at the software user interface (e. g. in the project navigation) in order to distinguish blocks and instructions of the standard user program. The fail-safe parameters of F-CPUs and F-I/O are shaded yellow in the hardware configuration.

5.3 Components of the safety program

Das safety program always consists of user-generated, system-generated F blocks and the "Safety administration" editor.

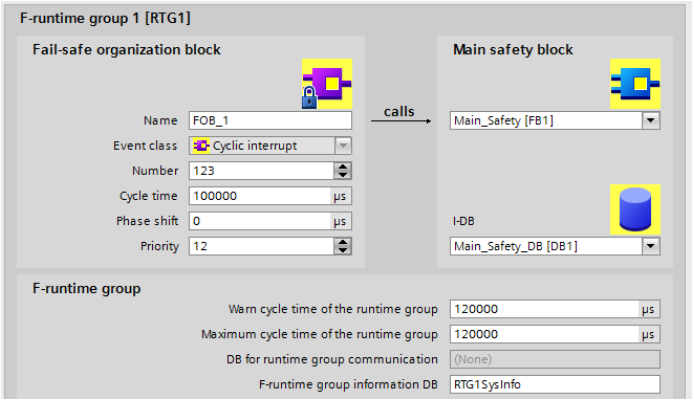
Table 5-2 Components of the safety program

Description	Screen
1. "Safety administration" editor - Status of the safety program - F collective signature - Status of the safety operation - Creating/organizing F runtime groups - Information on the F blocks - Information on F-conform PLC data types - Defining/changing the access protection 2. User-created F blocks 3. System-generated F runtime blocks - Blocks contain status information on the F runtime group. 4. System-generated F-I/O data blocks - Blocks contain tags for evaluating the F modules. 5. "Compiler blocks" System-generated verification blocks - These run in the background of the controller and provide for fail-safe processing of the safety program. - These blocks cannot be processed by the user.	

5.4 F-runtime group

A safety program is always processed in an F runtime group with defined cycle. An F runtime group consists of a "Fail-safe organization block" which calls a "Main safety block". All user-generated safety functions are called from the "Main safety block".

Figure 5-2 F runtime group in the "Safety administration" editor



Advantages

- Runtime groups can simply be created and configured in the "Safety Administrator".
- F-blocks in the runtime group are automatically created.

Properties

- A maximum of two F runtime groups can be created.

5.5 F signature

Each F component (station, I/O, blocks) has a unique F signature. Using the F signature it can be quickly detected whether an F device configuration, F blocks or a complete station still corresponds to the original configuration or programming.

Advantages

- Simple and quick comparison of F blocks and F device configurations

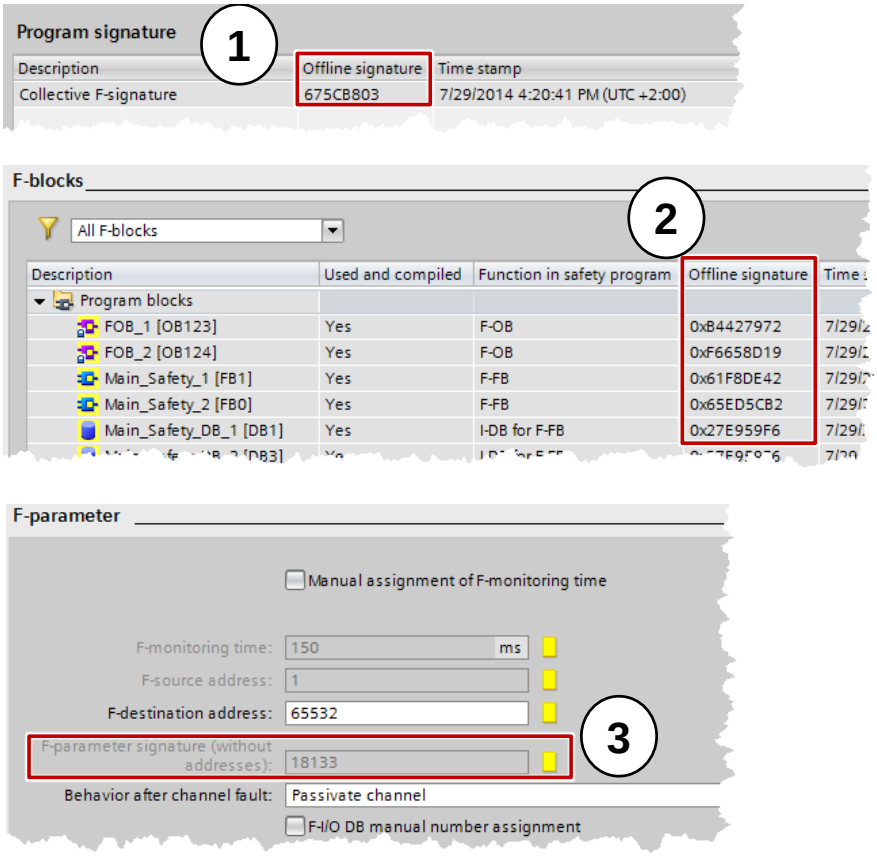
Properties

- F parameter signature (without address of F-I/O)…
  - only changed by adjusting the parameters.
  - remains unchanged when changing the PROFIsafe address. However, the F collective signature of the station changes.
- F block signature is only changed when the logic in the F block changes.
- F block signature remains unchanged by changing the
  - block number,
  - block interface,

- block version.

Example

Figure 5-3 Examples of F signatures



1. F collective signature of the station in the "Safety administration" editor
2. F block signatures in the "Safety Administration" editor (can also be read out from the properties of the block)
3. F parameter signature in the "Device view" at "Devices & Networks"

Note For S7-1500F controllers it is possible to read the F overall signature directly on the installed display or in the integrated web server.

5.6 Assigning the PROFIsafe address at the F-I/O

Each F-I/O device has a PROFIsafe address for identification and communication with F controllers. When assigning the PROFIsafe address, two different configurations are possible.

Table 5-3: Setting the F address

ET 200M / ET 200S (PROFIsafe address type 1)	ET 200MP / ET 200SP (PROFIsafe address type 2)
Assigning the PROFIsafe address directly at the modules via DIL switch In the device configuration of the TIA Portal and in the DIL switch position at the periphery, the PROFIsafe address must be the same.	Assigning the PROFIsafe address exclusively via TIA Portal The configured PROFIsafe address is loaded onto the intelligent coding module of the module.

Advantages

- Replacing an F module is possible without reassigning the PROFIsafe address at ET 200MP and ET 200SP. The intelligent coding module remains in the BaseUnit during module exchange.
- Simple configuration since TIA Portal indicates a faulty assignment of the PROFIsafe address warnings.
- The PROFIsafe addresses of all F modules can be assigned at the same time within an ET 200SP.

Note Further information on assigning the PROFIsafe address for the F-I/O is available at:

SIMATIC Industrial Software SIMATIC Safety – Configuring and Programming
https://support.industry.siemens.com/cs/ww/en/view/54110126

5.7 Evaluation of F-I/O

All of the current states of the respective F-I/O are saved in the F-I/O blocks. In the safety program the states can be evaluated and processed. The following differences exist between S7-1200F/1500F and S7-300F/400F.

Table 5-4: Tags in the F-I/O DB with S7-300F/400F and S7-1500F

Tag in F-I/O DB or value status in PAE	F-I/O with S7-300/400F	F-I/O with S7-1200F/1500F
ACK_NEC	yes	yes
QBAD	yes	yes
PASS_OUT	yes	yes
QBAD_I_xx *	yes	no
QBAD_O_xx *	yes	no
Value status	no	yes

* QBAD_I_xx and QBAD_O_xx show you the validity of the channel value and correspond to the **inverted** value status at S7-1200F/1500F (further information is available in the following chapter).

5.8 Value status (S7-1200F/1500F)

In addition to the diagnostic messages and the status and error display, the F module provides information on the validity of each input and output signal - the value status. The value status is stored in the same way as the input signal in the process image:

The value status informs about the validity of the respective channel value.

- 1: A valid process value is output for the channel.
- 0: a substitute value is output for the channel.

Table 5-5: Differences Q_BAD (S7-300F/400F) and value status (S7-1200F/1500F)

Scenario	QBAD (S7-300F/400F)	Value status (S7-1200F/1500F)
Valid values at the F-I/O (no error)	FALSE	TRUE
Channel error occurs	TRUE	FALSE
Channel error going (ACK_REQ)	TRUE	FALSE
Acknowledgement of the failure (ACK_REI)	FALSE	TRUE

Properties

- The value status is entered into the process image of the inputs and outputs.
- Channel value and value status of an F-I/O must only be accessed from the same F run-time group.

Recommendation

- For improved readability assign the ending "`VS`", e.g. "`TagIn1VS`" as the symbolic name for the value status.

Example

Position of the value status bits in the process image using the example of an F-DI 8x24VDC HF module.

Table 5-6: Value status bits in the process image using the example of an F-DI 8x24VDC HF

Byte in the F-CPU	Assigned bits in the F-CPU							
	7	6	5	4	3	2	1	0
x + 0	DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0
x +1	Value status for DI_7	Value status for DI_6	Value status for DI_5	Value status for DI_4	Value status for DI_3	Value status for DI_2	Value status for DI_1	Value status for DI_0

x = module start address

Note More information about the value status of all ET 200SP modules is available at:

Failsafe CPUs - Manuals
https://support.industry.siemens.com/cs/ww/en/ps/13719/man

Failsafe I/O modules - Manuals
https://support.industry.siemens.com/cs/ww/en/ps/14059/man

5.9 Data types

5.9.1 Overview

There is an unrestricted scope of data types for the safety programs of the S7-1200/1500F.

Table 5-7 Integer data types

Type	Size	Value range
BOOL	1 bit	0 .. 1
INT	16 bit	-32.768 .. 32.767
WORD	16 bit	-32.768 .. 65.535
DINT	32 bit	-2.14 .. 2.14 Mio
TIME	32 bit	T#-24d20h31m23s648ms to T#+24d20h31m23s647ms

5.9.2 Implicit conversion

In safety-relevant applications is may be necessary to carry out mathematical functions with tags of different data types. The function blocks necessary for this, require a defined data format of the formal parameters. It the operand does not comply with the expected data type, a conversion has to be carried out first.

Under the following circumstances can the S7-1200/1500 also perform the data conversion implicitly:

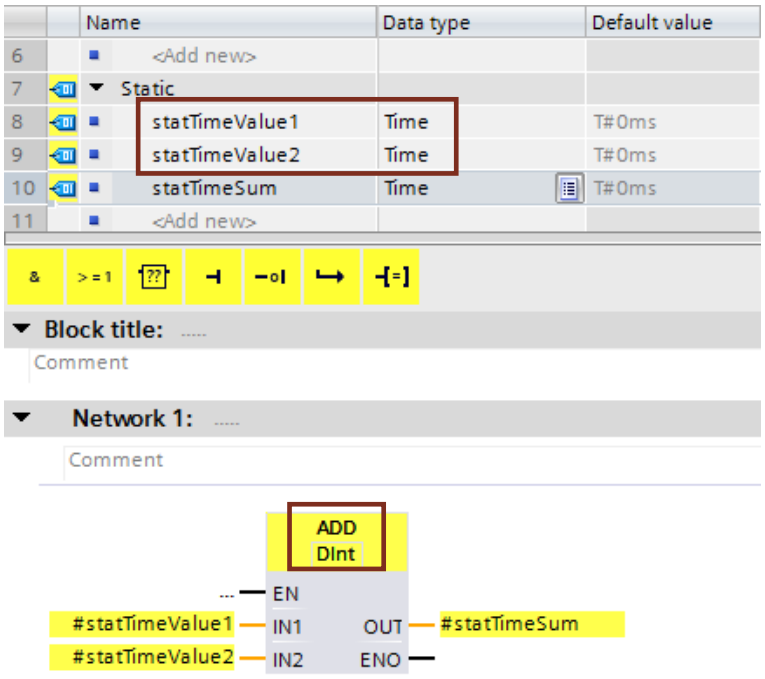
- IEC check is disabled.
- The data types have the same length.

For this reason, the following data types can be converted implicitly in the safety program:

- WORD ↔ INT
- DINT ↔ TIME

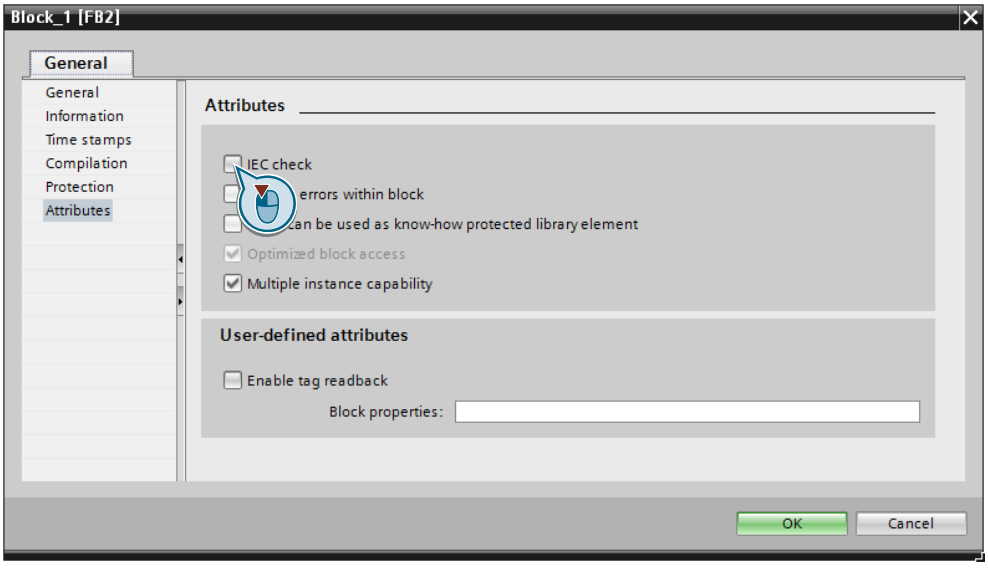
A practical application is the addition of two time values, although the function "Add" is required as "DInt" input. The result is then also output as "Time" tag.

Figure 5-4: Addition of two time values



Enable or disable the IEC check in the properties of the respective function block or function.

Figure 5-5: Disabling IEC check



5.10 F-conform PLC data type

For safety programs it is also possible to structure data optimal with PLC data types.

Advantages

- A change in a PLC data type is automatically updated in all usage locations in the user program.

Properties

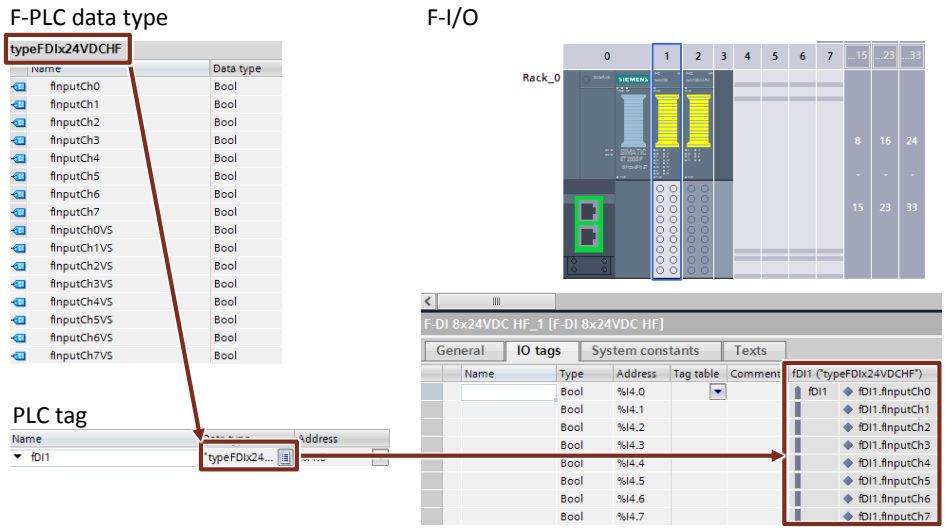
- F-PLC data types are declared and used in the same way as PLC data types.
- As F-PLC data types, all data types which are allowed in the safety program can be used.
- Nesting of F-PLC data types within other F-PLC data types is not supported.
- F-PLC data types can be used in the safety program as well as in the standard user program.

Recommendation

- You use F-PLC data types for accessing I/O areas (as in chapter 3.6.5 Access to I/O areas with PLC data types)
- The following rules must be observed here:
  - The structure of the tags of the F-conform PLC data type must match the channel structure of the F-I/O.
  - An F-conform PLC data type for an F-I/O with 8 channels is, for example:
    - 8 BOOL tags (channel value) or
    - 16 BOOL tags (channel value + value status)
  - Access to F-I/O is only permitted for activated channels. When configuring a 1oo2 (2v2) evaluation, the higher channel is always deactivated.

Example

Figure 5-6: Access to I/O areas with F-PLC data types



5.11 TRUE / FALSE

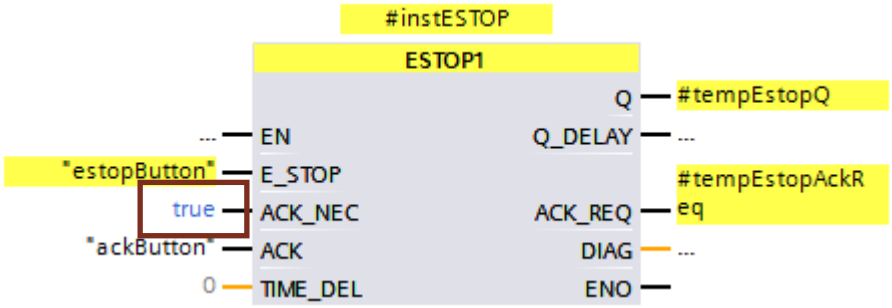
The use of "TRUE" and "FALSE" signals in the safety programs can be differentiated in two application cases:

- as actual parameter at blocks
- as assignment to operations

Actual parameter at blocks

For S7-1200F/1500F controllers you can use the Boolean constants "FALSE" for 0 and "TRUE" for 1 as actual parameter for supplying formal parameters during block calls in the safety program. Only the keyword "FALSE" or "TRUE" is written to the formal parameter.

Figure 5-7: "TRUE" and "FALSE" signals as actual parameter



Assignments to operations

In order to create "TRUE" or "FALSE" signals for operations, proceed as follows:

1. Create two static tags "statTrue" and "statFalse" of the type BOOL.
2. Assign the default value "false" to the statFalse tag.
3. Assign the default value "true" to the statTrue tag.

You can use the tags as "True" and "False" read signals in the complete function block.

Figure 5-8: "TRUE" and "FALSE" signals

Name	Data type	Default value	Retain
Static			
statTrue	Bool	true	Non-retain
statFalse	Bool	false	Non-retain

5.12 Optimizing the compilation and program runtime

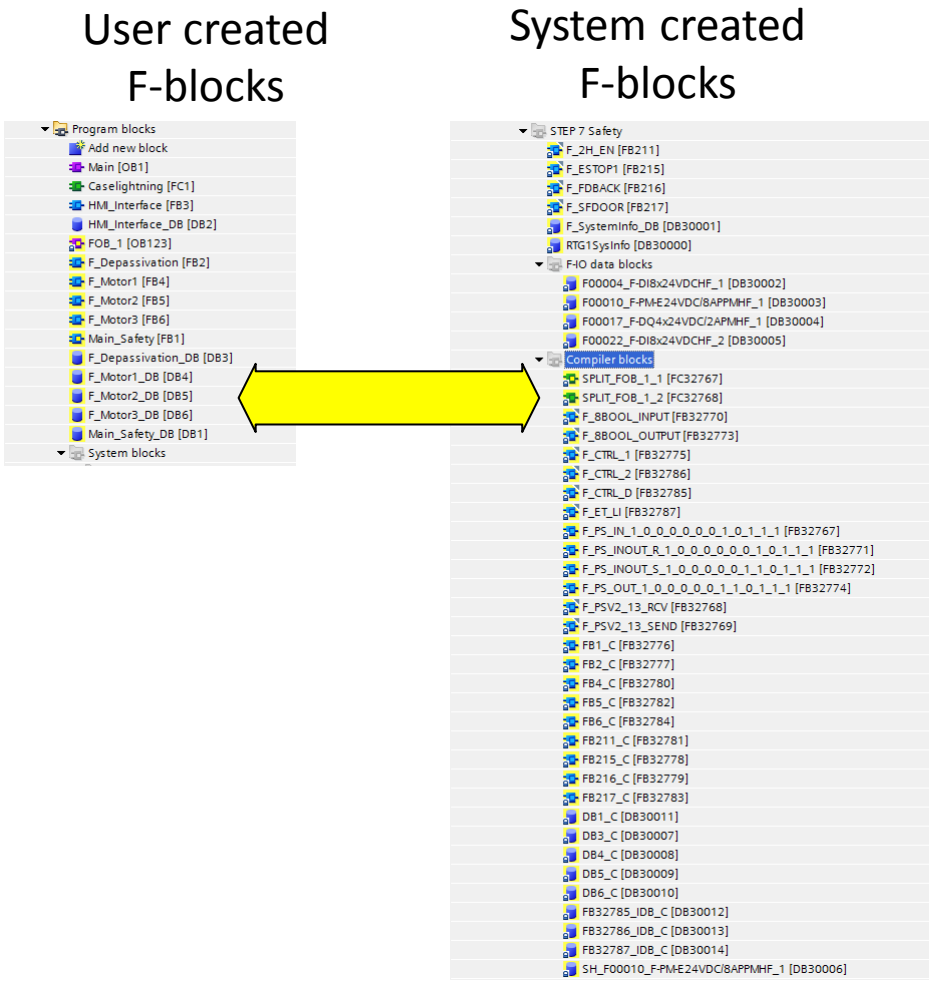
An important part of the safety program is the protection of the user programming by coded processing. The aim is to discover any kind of data corruption in the safety program and therefore to prevent unsafe conditions.

This protection program is created during the compilation and therefore prolongs the compilation time. The runtime of the F-CPU is also prolonged through the protection program, since the F-CPU processes it additionally and compares the results with the user program.

The protection program that is generated automatically by the system can be found in the system block folder of your F-CPU.

Example

Figure 5-9: User and system created F blocks



This chapter shows you the different options for shortening the compilation and program runtime.

Depending on the use it will not always be possible to use all suggestions. They nevertheless provide information why certain programming methods cause shorter compilation and program runtimes then a non-optimized program.

5.12.1 Avoiding of time-processing blocks: TP, TON, TOF

Every time-processing block (TP, TON, TOF) requires additional blocks and global data corrections in the protection code.

Recommendation

Use these blocks as little as possible.

5.12.2 Avoiding deep call hierarchies

Deep call hierarchies enlarge the code of the system-created F blocks, since a larger scope of protective functions and test is required. When the nesting depth of 8 is exceeded, the TIA Portal will emit a warning during the compilation.

Recommendation

Structure your program in a way as to avoid unnecessary deep call hierarchies.

5.12.3 Avoiding JMP/Label structures

If a block call is jumped via JMP/LABEL this leads to an additional protection in the F blocks on the system side. Here, a correction code has to be carried out for the skipped block call. This costs performance and time in the compilation

Recommendation

Avoid JMP/Label structures as far as possible to reduce F-blocks on the system side.

5.13 Data exchange between standard program and F program

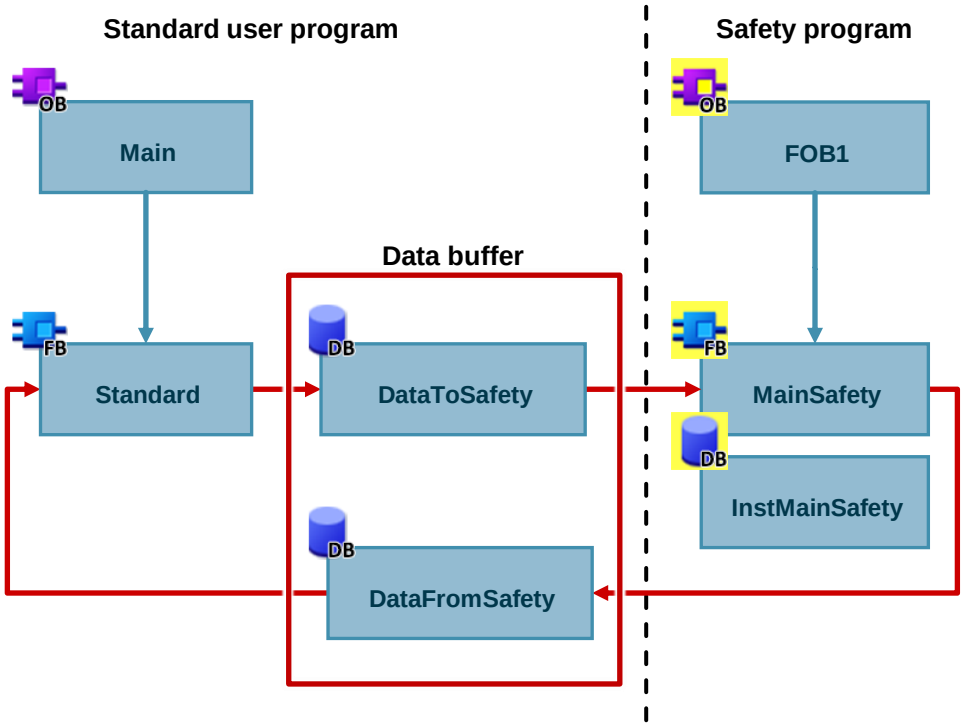
In some cases it is necessary to exchange data between the safety program and the standard user program. The following recommendations should urgently be noted in order to guarantee data consistency between standard and the safety program.

Recommendations

- No data exchange via bit memory (see chapter 4.2 No bit memory but global data blocks)
- Concentrate the access between safety program and the standard user program on two standard DBs.

  Changes in the standard program will therefore have no influence on the safety program. The controller also does not need to be in STOP mode to load the standard program.

Figure 5-10: Data exchange between standard and safety program



5.14 Testing the safety program

In addition to the always controllable data of a standard−user program you can change the following data of a safety program in the deactivated safety mode.

- Process image of F−I/O
- F−DBs (except DB for F−runtime group communication), instance−DBs of F-FBs
- F-I/O DBs

Properties

- Controlling F−I/O is only possible in F−CPU RUN mode.
- From a watch table you can control a maximum of 5 inputs/outputs in a safety program.
- You can use several watch tables.
- As trigger point you need to set "permanent” or "once” for "cycle start” or "cycle end”.
- Forcing is not possible for the F-I/O.
- If you still wish to use stop points for testing, you need to deactivate the safety mode beforehand. This leads to the following errors:
  - Error during communication with the F-I/O
  - Error at fail-safe CPU-CPU communication

5.15 STOP mode in the event of F errors

In the following cases, the STOP mode is triggered for F-CPUs:

- In the "System blocks" folder you must not add, change or delete any blocks.
- There must not be any access to instance DBs of F-FBs which are not called in the safety program.
- The "Maximum cycle time of the F-runtime group" must not be exceeded. Select the maximal permitted time for "Maximum cycle time der F run-time group" which can elapse between two calls of this F runtime group (maximum 20000 ms).
- If tags are read from a DB for F runtime group communication whose runtime group is not processed (main safety block of the F runtime group is not called).
- Editing the start values in instance DBs of F-FBs is not permitted online and offline and can lead to STOP of the F-CPU.
- The main safety block must not contain any parameters since they cannot be supplied.
- Outputs of F-FCs must always be initialized.

5.16 Migration of safety programs

Information on migrating safety programs is available at:

https://support.industry.siemens.com/cs/ww/en/view/109475826

5.17 General recommendations for safety

Generally, the following recommendations apply for handling STEP 7 Safety and F modules.

- Whenever possible, always use F controllers. Thus, a later expansion of safety functions can be realized very easily.
- Always use one password for the safety program to prevent unauthorized changes. The password is set in the "Safety administration" editor.