Panasonic

Safety Multi Application Controller
SUNX-SFL
Safety Function Blocks

Description SFL | Version 1.5-04 en

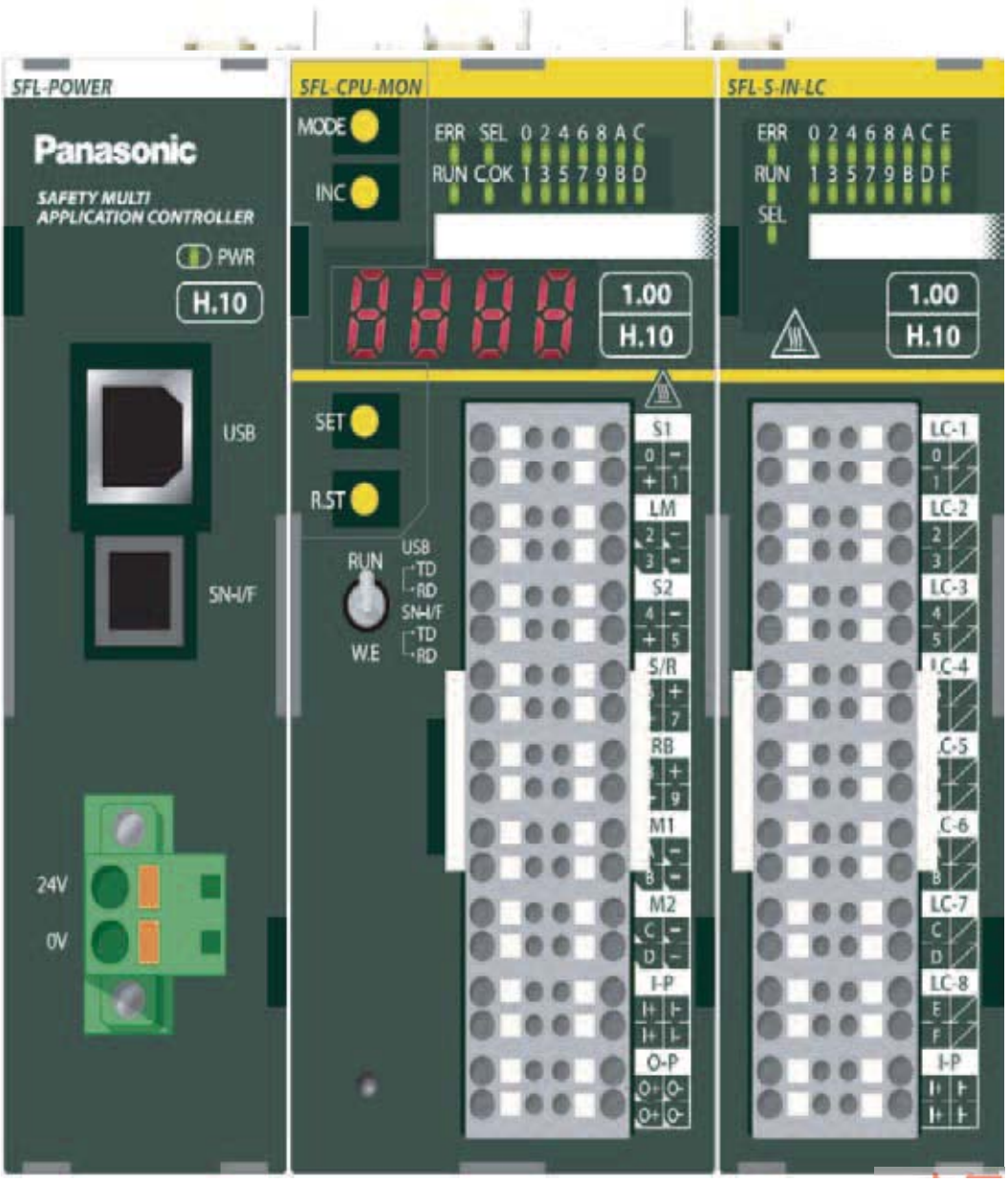


Table of contents

3 Protected functional modules within the **SFL-SOFT** library function
(continued)

1 General

The functional blocks described in this document (also referred to below as FB or FBs) are deployed in the library function of the **SFL-SOFT** programming software used to program the **SUNX-SFL** safety control system in operating mode 3.

A precise description of the **SUNX-SFL** safety control system, the **SFL-SOFT** programming software and incorporation of the library function into this software can be found in the **SUNX-SFL** instruction manual.

2 Safety information

Use of the FBs does not release the user from the need to perform a careful safety check of the overall program. All sub-functions realised using the FBs should in particular be inspected for achieving the desired PL/SIL in the overall system.

2.1 Hardware

The safety of the entire application is dependent on all components used to achieve the safety function (e.g. emergency stop control devices, guard monitoring, contactors etc.).

With two-channelled input/output wiring, the **SUNX-SFL** safety control system by itself (without the upstream sensor technology and downstream actuating elements) satisfies all requirements of Performance Level (PL) e in accordance with EN ISO 13849-1 or SIL 3 in accordance with EN IEC 62061. More details about this and about other conditions for switching the inputs/outputs (e.g. minimal pulse duration on the input > 15 ms) can be found in the **SUNX-SFL** instruction manual.

Caution: If a safety-critical setting has been performed on an FB, for example “no monitoring of the feedback loop” (deselect External Device Monitoring [EDM], auto reset or start reset, a procedure must be provided at a different point in the overall programme in order to prevent a faulty or unexpected restart or to achieve the required PL or SIL.

The Discrepancy Time between guard contact 1 (G_SW_1) and contact 2 (G_SW_2) is permanently set to 5 seconds. The FB must not be used if the Discrepancy Time for Guard Monitoring FB does not comply with the application.

2.2 Software

All FBs described below support the programmer of the **SFL-SOFT** programming software to prepare programs for the **SUNX-SFL** safety control system that satisfies requirements up to Performance Level e in accordance with EN ISO 13849-1 or SIL 3 in accordance with EN IEC 62061. The resultant PL/SIL of the overall system must be determined as part of the safety evaluation.

NB: All FBs serve to support the creation of a program. The functions programmed in the FBs are based on the ladder diagram (LD) graphic programming language in accordance with EN IEC 61131-3 and can also be individually replicated in **SFL-SOFT** using discrete LD programming. The FBs represent just one possibility for creating the function described.

The programmer himself must take additional constraints within the overall program into consideration. An example here is the incorporation of the feedback loop as an additional RESET condition (or for the auto reset condition).

All functional modules from the “**SUNX-SFL** Library” are protected, i.e. they cannot be changed by the programmer. Protected FBs are labelled “geschützt” (in a German programming environment) or “protected” (in an English programming environment) and have date and seal identifier (checksum). The programmer has the possibility starting from software version **SFL-SOFT** V1.2R01 to protect the FBs he has created himself using a password. This is done by actuating the “Settings for FB protection” button and then assigning a name and password in the “Settings for FB protection” screen after the FB has been saved in the “Function block properties” screen (see Figure 2-1).

2.3 Features of the SFL-SOFT

Due to the special programming structure of **SFL-SOFT**, the inputs, outputs or flags used are evaluated with respect to safety and labelled with an “S” (“assessed as safe”) or “U” (“assessed as unsafe”). This labelling is primarily based on the assessment of the input signals which are only assessed to be “safe” if they represent a “safe 1” or “safe 0”.

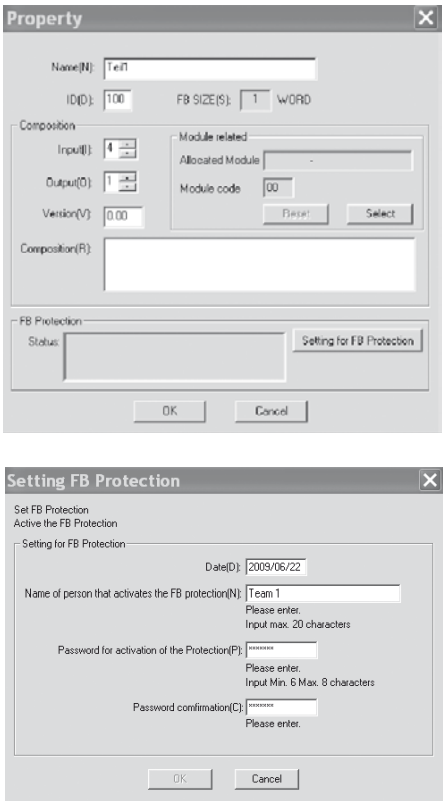


Fig. 2-1: Screens to protect own created FBs with a password.

The significance of a safe or an unsafe contact is set out in Chapter 7.10.4 of the **SUNX-SFL** instruction manual.

NB: The “True_False” FB can be used to generate a “safe 1” = TRUE or “safe 0” = FALSE. This generates a “safe 1” = TRUE and a “safe 0” = FALSE on its outputs if the input is wired with a safe contact. The “safe 1” and “safe 0” output signals generated can be used as flags in the rest of the program. See chapter 3.0.1 and also the example programs provided in the following chapters in this respect.

The designation TRUE/FALSE will be used in the following when options for an FB are activated or deactivated. “Safe 0” or “safe 1” will be used where signals of the application program are concerned (e.g. inputs, flags).

3 Protected functional modules within the SFL-SOFT library function

3.0.1 “True_False” FB (to generate a “safe 1” and a “safe 0”)

(Name of FB in the SUNX Library: “True_False” in the “Auxiliary” directory; seal ID: 3D89)

As described in Chapter 2.3, input or output signals from **SFL-SOFT** are evaluated with respect to their safety. All functional modules described in Chapter 3 are designed such that safe signals can be expected on their inputs to prevent outputs assessed as unsafe that are marked with a “U” (red, and thereafter in yellow).

According to the rules in Chapter 7.10.4 of the **SUNX-SFL** instruction manual, the “True_False” auxiliary function block is structured so that a “safe 1” (or safe high signal = TRUE) is generated on the OUT1 output or a “safe 0” (or safe low signal = FALSE) on the OUT2 output from a safe input signal (e.g. input of an emergency stop control device) on the “IN1” input. Usually the outputs are laid on flags in order to use them as safe input signals for the desired function on the FB.

The “True_False” FB precedes the respective safe FBs in the programming examples in Chapter 3.1 to 3.7.

3.0.2 Description of the “True_False” auxiliary function block

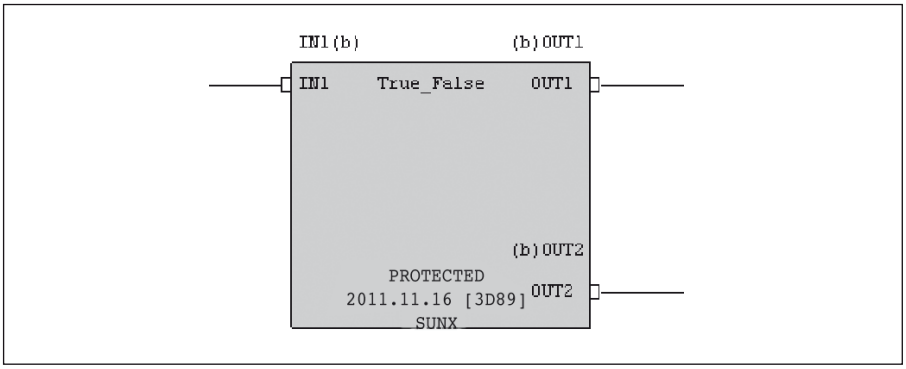


Fig. 3-1: “True_False” FB

Input/output	Name	Description
IN1		Safe input signal; e.g. safe input
OUT1		Output “safe 1” = TRUE: • On flag; for further use
OUT2		Output “safe 0” = FALSE: • On flag; for further use

Tab. 3-1: Description of I/O of “True_False” FB

3.1 “Emergency Stop” FB

Name of FB in the SUNX Library: “EmergencyStop”; seal ID: 57C7 (2011.11.16)

3.1.1 Application

The “EmergencyStop” FB is used to monitor emergency stop control devices.

Emergency stop control devices are generally incorporated in the safety circuit of a machine or system and have special significance as part of the machine and system control. The safe function and rapid accessibility of emergency stop control devices can have a crucial effect on the lives of people and the degree of personal injury in the event of a disturbance or accident or on the extent of damage to a machine in case of a fault.

New requirements apply to the operating mode of these control devices with the coming into force of the new harmonised European standard EN ISO 13850 on functional aspects and principles for design for emergency stop equipment.

The “Emer- gencyStop” FB described in this chapter helps to implement these requirements into the **SUNX-SFL** safety control system.

Caution: The possibility for programming the “EmergencyStop” FB in a function that contradicts the requirements of EN ISO 13850 (e.g. without feedback loop) may only be applied if the program in its entirety satisfies the requirements of the standard.

3.1.2 Features

The “EmergencyStop” FB has been created on the basis of programming guidelines from PLCopen, the organisation for the industrial control technology field.

3.1.3 Description of “Emergency Stop” FB inputs/outputs

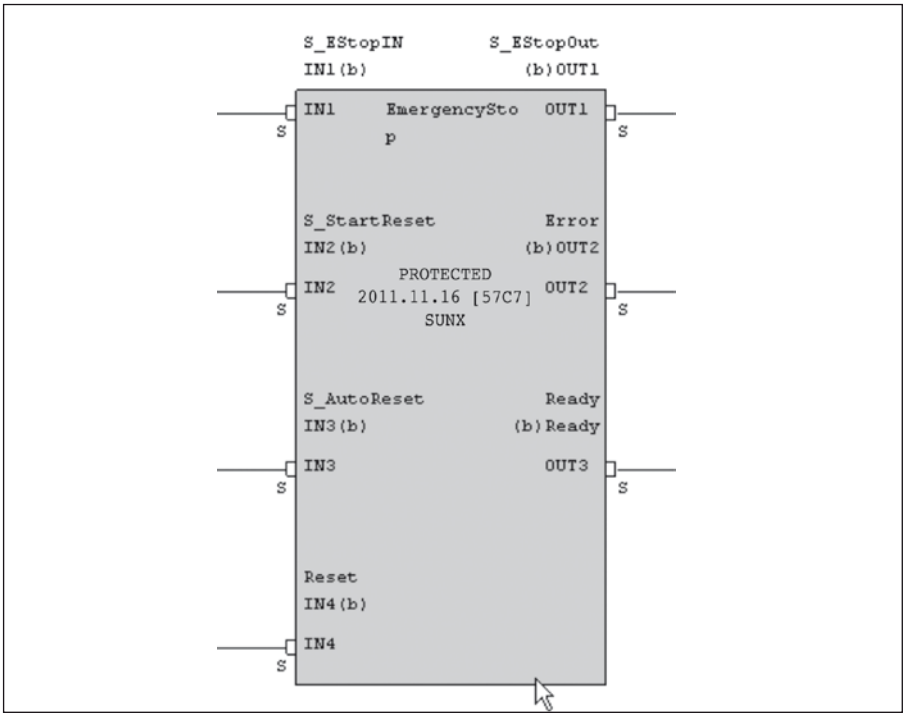


Fig. 3-2: “Emergency Stop” FB

Input/output	Name	Description
IN1	S_EStopIN	Safety request: • “safe 1”: emergency stop not actuated • “safe 0”: FALSE: emergency stop actuated
IN2	S_StartReset	Power on reset: • TRUE: automatic start takes place after Power-on Reset (IN4) not necessary here • FALSE: reset (IN4) must be actuated after Power-on
IN3	S_AutoReset	Automatic reset: • TRUE: automatic reset (input IN4 without function) • FALSE: reset (IN4) must be actuated after safety request
IN4	Reset	Input of the reset signal: reset on trailing edge (change “safe 1” → “safe 0”
OUT1	S_EStopOut	Safety enabling output: • “safe 1”: safety output cleared • “safe 0”: safety output blocked
OUT2	Error	Error message: • TRUE: error (error reset function) • FALSE: no error (reset function OK)
OUT3	Ready	Function block operation status: • TRUE: FB in use • FALSE: FB not in use

Tab. 3-2: Description of I/Os of “Emergency Stop” FB

3.1.4 Connection example

Connection example: Emergency stop with monitored reset (trailing edge)

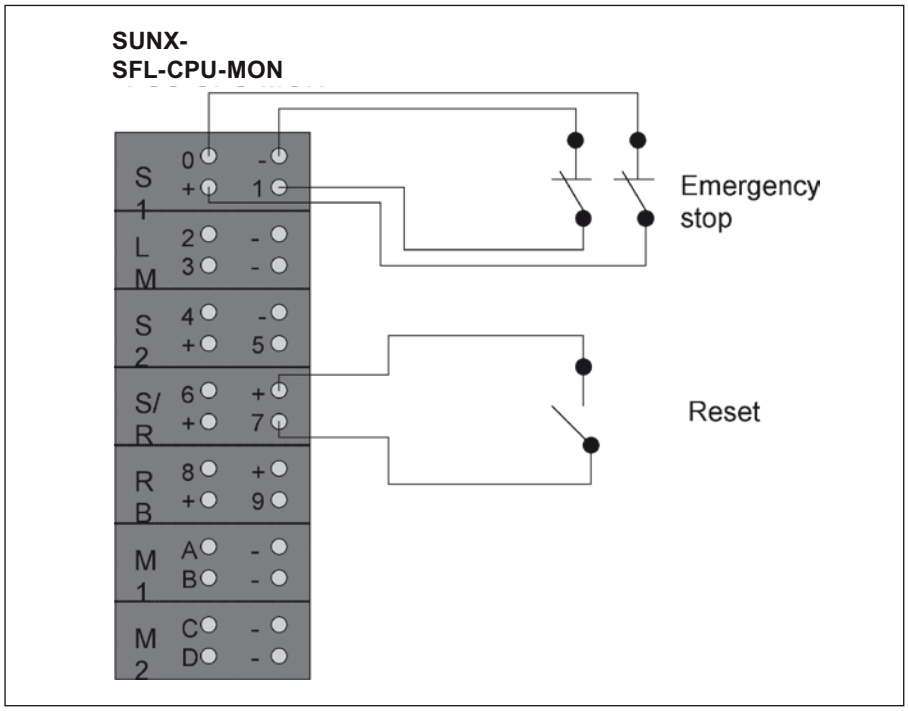


Fig. 3-3: Connection of **SFL-CPU-MON** for example programs 3.1.5

Description

An emergency stop control device and a reset button are connected to the **SUNX-SFL-CPU-MON** module as shown.

After a program start or after resetting the emergency stop control device, the condition (referred to in subsequent example programs as “M000” flag) required to enable the FBs (S_EStopOut, OUT1 = “safe 1” is provided by actuating or by releasing the reset button. The “M000” flag must therefore be integrated accordingly in the overall program.

3.1.5 Example program

a) Emergency stop with monitored reset through integration of the "EmergencyStop" FB in the ladder diagram of the SUNX-SFLsw

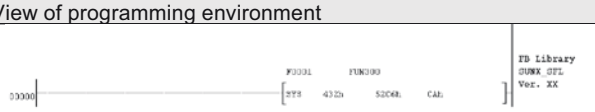
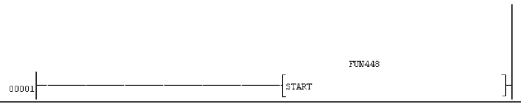
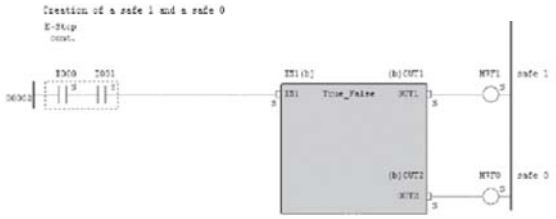
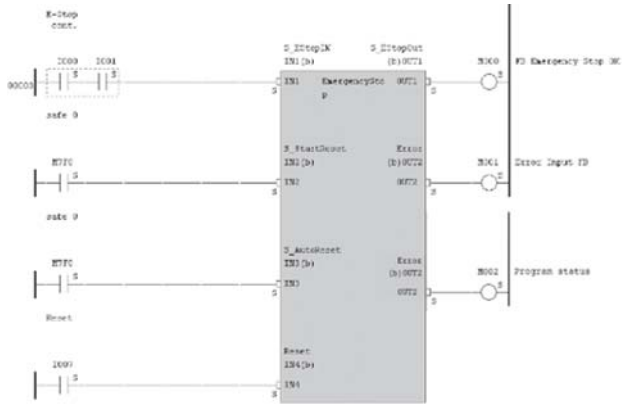
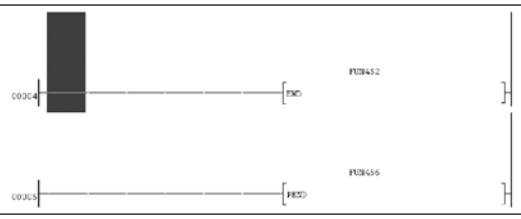
View of programming environment	Comment
	Link to library takes place
	Program start
	Wiring of the "True_False" functional module: • Generation of a "safe 1" (here: M7F1) to OUT1 and a "safe 0" (here: M7F0) to OUT2 with the help of a safe input contact (here: I000 and I001) to IN1. The "True_False" FB from the "Auxiliary" folder is used for this.
	Wiring of the "Emergency Stop" function block: • IN1: Connection for the contacts to be monitored; here: two-channelled monitoring of the emergency stop contacts with the addresses I000 and I001. • IN2: Power-On reset not desired (wired with FALSE) • IN3: Automatic reset not desired (wired with FALSE). • IN4: Reset with trailing edge desired (IN2 and IN3 = FALSE), IN4 is assigned the address of the reset switch (here: I007). • OUT1: Safe output • Addressed output or flag (here: M000) is "safe 1" where all selected conditions of the selected input functions (IN1 - IN4) have been satisfied. • OUT2: Error report • Addressed output or flag (here: M001) is TRUE where the selected input function (IN1 – IN4) is implausible. • OUT3: Program status • Addressed output or flag (here: M0002) is TRUE where the user program is started and in use.
	End of program

Fig. 3-4: **SFL-SOFT** programming environment – example program a: "True_False" and "EmergencyStop" FBs