PLCopen® - Technical Committee 5
–
Safety Functionality

Technical Specification

Part 3: Extensions to the Function Blocks

Version 1.0 – Official Release

Date: December 16, 2013.

Extensions to Concepts and Function Blocks for Safety Functions

The following paper is a document created within the PLCopen Technical Committee 5 – Safety Software.
It summarizes the results of the PLCopen Technical Committee meetings, containing contributions of its members:

Jochen Ost	Bosch Rexroth, Germany
Olaf Ruth	Phoenix Contact, Germany
Harry Koop	KW Software, Germany
Martin Gottwald	Siemens, Germany
Franz Kaufleitner	B&R, Austria
Thomas Baier	Logicals, Austria
Frank Bauder	Omron, Germany
Joachim Greis	Beckhoff, Germany
Erich Janoschek	TÜV Rheinland, Germany
Michael Huelke	BGIA, Germany
Eelco van der Wal	PLCopen, The Netherlands

Change Status List:

Version number	*Date*	*Change comment*
V 0.1	Jan. 20, 2009	Document created as decided on meeting January 13, 2009
V 0.2	July 14, 2009	Inclusion of four MC related FBs as provided by B&R
V 0.3	July 21, 2009	As result of the meeting in Cologne
V 0.4	Oct. 16, 2009	As result of the meeting at B&R. First document as Part 3 (vs. Part4)
V 0.5	Dec. 18, 2009	Prepared - As a result of the meeting at KW Software
V 0.6	March 23, 2010	As a result of the meeting at Beckhoff
V 0.7	July 1, 2010	As a result of the meeting in Frankfurt a.M.
V 0.8	Aug. 25, 2010	As a result of the meeting in Lemgo
V 0.9	Dec. 08, 2010	As last edited version before release for comments version
V 0.91	Dec. 16, 2010	As a result of the webmeeting on that day
V 0.92	Jan. 21, 2011	As result of the face2face meeting in January
V 0.93	Oct. 5, 2011	As result of the face2face meeting in October
V 0.94	Feb. 8, 2012	As result of meeting Jan 31, 2012 in Bad Pyrmont
V 0.95	Feb. 29 2012	As result of the meeting in February in Augsburg
V 0.96	April 4, 2012	Final version before release for comments. For internal feedback only
V 0.99	April 21, 2012	Published as ‘ Release for Comments’ for feedback till June 22, 2012
V 0.99a	July 12, 2012	As a result of the meeting in the vicinity of Amsterdam as well as webmeeting Sept. 2012
V 0.99b	Nov. 22, 2012	As result of the webmeeting. Changes in 1.2 Harmonization of diagnostic codes for new function blocks. And change in state diagram SF_Override.
V 1.0	Dec. 16, 2013	Official release in conjunction with Part 4

Contents

1 Introduction

In February 2006, the PLCopen Technical Committee 5 published their Safety Specification Part 1 - Concepts and Function Blocks for Safety Functions. It became obvious that additional functionalities were needed. The additions are partly dealt with in this document.

1.1. Extensions to General Output Parameters of Part 1

Function Block- Specific rules – General output parameters (extension to Part 1 Section 5.1.2)

Output Parameter		
Name	**Type**	**Description**
SafetyDemand	BOOL	Signal indicating that the FB is active and the primary safety function is demanded (e.g. related to the safety functionality). Other safety related input parameters are not considered (e.g. SafetyActive and EDM). The safety loop is not closed and the safe state is demanded for the related safety output. There is no error. TRUE: Safety demand FALSE: No Safety demand
ResetRequest	BOOL	Signal which can be used to signal the operator to press the reset functionality to continue. TRUE: Reset requested FALSE: Reset not requested.

Both SafetyDemand and ResetRequest set to TRUE does not provide unique information for the operator, and for this reason only one is SET at the same time.
By providing these outputs directly in the FB, it is easy to connect these to an operator interface and in this way help to identify the applicable actions to be done.

1.2. Harmonization of diagnostic codes for new function blocks

It was decided that for new function blocks the following DIAG codes will be used in order to make the evaluation in software easier and more straightforward coupled to the new outputs SafetyDemand and ResetRequest:

Name	DIAG	$DiagCode_{bin}$									Error	Safety Demand	Reset Request	Reset Error	Safety Outputs
		Nibble1			Nibble2			Nibble3	Nibble4						
		1	E	00	S	R	xx	xxxx	xxx	RE					
Error	Cyn0	1	1	00	0	0	xx	xxxx	000	0	1	0	0	0	0
Reset Error	Cyn1	1	1	00	0	0	xx	xxxx	000	1	1	0	0	1	0
Error AND ResetRequest	Cwn0	1	1	00	0	1	xx	xxxx	000	0	1	0	1	0	0
Error AND SafetyDemand	Not applicable (Error)														
		Nibble1			Nibble2			Nibble3	Nibble4						
		1	E	00	S	R	xx	xxxx	xxxx						
SafetyActive AND SafetyOutput	8yn0	1	0	00	0	0	xx	xxxx	0000		0	0	0	0	1
SafetyActive	8ynz	1	0	00	0	0	xx	xxxx	xxx0		0	0	0	0	0
Init AND ResetRequest	84n1	1	0	00	0	1	00	xxxx	0001		0	0	1	0	0
Init AND SafetyDemand	88n1	1	0	00	1	0	00	xxxx	0001		0	1	0	0	0
ResetRequest	84nz	1	0	00	0	1	00	xxxx	xxx0		0	0	1	0	0
SafetyDemand	88nz	1	0	00	1	0	00	xxxx	xxx0		0	1	0	0	0
															0
Idle	0000	0	0	00	0	0	00	0000	0000		0	0	0	0	0

Notes:

- S = 0 when only a reset is required. =1 when the safety link is not yet closed and needs operator attention. Equals the negation of the Safety Inputs.
- R = 0 when no reset is required. =1 when only a reset is required.
- RE = Reset Error
- x [0,1]
- n [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F] (In the combination ‘yn’, ‘n’ is leading over ‘y’, meaning that first ‘n’ is increased by one and after reaching ‘ F’, ‘y’ is increased by one. Similar for the other combinations)
- y [0, 1, 2, 3]
- z [2, 4, 6, 8, A, C, E]
- w [4, 5, 6, 7]

For clarification, hereunder the Diagnostic Code Definition as of Part1 are copied, with added codes for SafetyDemand and SafetyRequest values, as well as the new diagnostics codes:

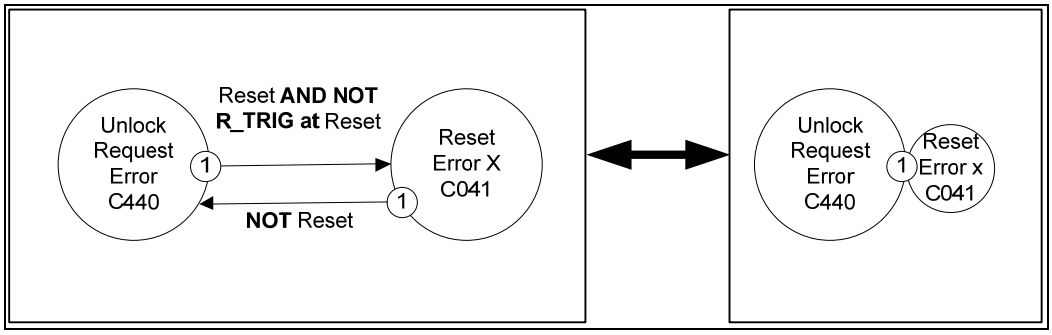
Generic Diagnostic Codes	
DiagCode	**Description**
$0000\_0000\_0000\_0000_{bin}$ **0000_{hex}**	The FB is not activated. This code represents the Idle state. For a generic example, the I/O setting for could be: Activate = FALSE S_In = FALSE or TRUE Ready = FALSE Error = FALSE S_Out = FALSE SafetyDemand = FALSE ResetRequest = FALSE

Generic Diagnostic Codes	
DiagCode	**Description**
1000_0000_0000_0000$_{bin}$ **8000$_{hex}$**	The FB is activated without an error or any other condition that sets the safety output to FALSE. This is the default operational state where the S_Out safety output = TRUE in normal operation. For a generic example, the I/O setting for could be: Activate = TRUE S_In = TRUE Ready = TRUE Error = FALSE S_Out = TRUE SafetyDemand = FALSE ResetRequest = FALSE
1000_0100_0000_0001$_{bin}$ **8401$_{hex}$**	An activation has been detected by the FB and the FB is now activated, but the S_Out safety output is set to FALSE. This code represents the Init state of the operational mode. For a generic example, the I/O setting for could be: Activate = TRUE S_In = TRUE Ready = TRUE Error = FALSE S_Out = FALSE SafetyDemand = FALSE ResetRequest = TRUE
1000_0100_0000_0001$_{bin}$ **8801$_{hex}$**	An activation has been detected by the FB and the FB is now activated, but the S_Out safety output is set to FALSE. This code represents the Init state of the operational mode. For a generic example, the I/O setting for could be: Activate = TRUE S_In = FALSE Ready = TRUE Error = FALSE S_Out = FALSE SafetyDemand = TRUE ResetRequest = FALSE
1000_1000_0000_0010$_{bin}$ **8802$_{hex}$**	The activated FB detects a safety demand ("Sicherheitsanforderung" in German), e.g., S_In = FALSE. The safety output is disabled. This is an operational state where the S_Out safety output = FALSE. For a generic example, the I/O setting for could be: Activate = TRUE S_In = FALSE Ready = TRUE Error = FALSE S_Out = FALSE SafetyDemand = TRUE ResetRequest = FALSE <Note: the detected safety demand refers to the states that are not IDLE or SAFESTATE>
1000_0100_0000_0011$_{bin}$ **8403$_{hex}$**	The safety output of the activated FB has been disabled by a safety demand. The safety demand is now withdrawn, but the safety output remains FALSE until a reset condition is detected. This is an operational state where the S_Out safety output = FALSE. For a generic example, the I/O setting for could be: Activate = TRUE S_In = FALSE => TRUE (continuing with static TRUE) Ready = TRUE Error = FALSE S_Out = FALSE SafetyDemand = TRUE ==> FALSE ResetRequest = R

Additional information on this matter can be found in section 2.4 Diagnostic FB.

1.3. Changes to the State Diagram

The additional outputs and the diagnostic codes reflect on the state diagram. In order to provide a clear overview, the following states are graphically merged in each state diagram: Reset Error.



The transition conditions are not shown, but always equal to the above. The priority is shown, and the DiagCode of the Reset Error. There is a relationship between the source of the transition (in this case Unlock Request Error) and the corresponding DiagCode: the second nibble is reused, e.g. C440 to C041.

2 Safety Function Blocks

2.1. Safety Guard Interlocking with Locking (Version 2)

2.1.1. Applicable Safety Standards

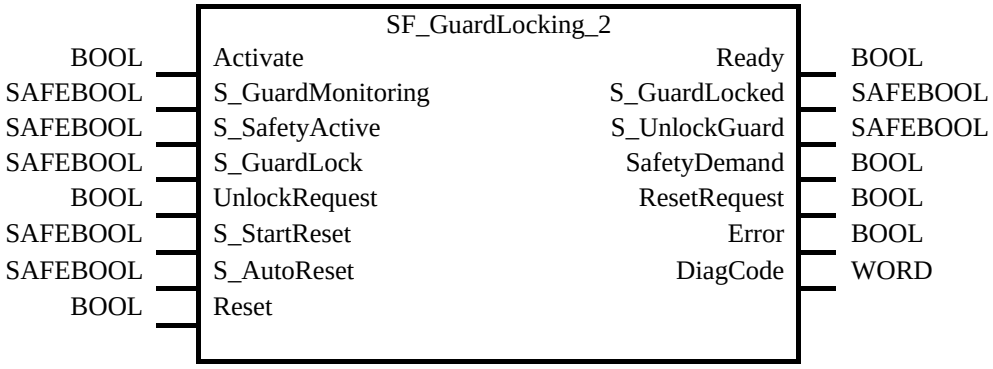
Standards	Requirements
EN 953: 1997 +A1:2009	3.3.3 Control Guard – The hazardous machine functions "covered" by the guard cannot operate until the guard is closed; – Closing the guard initiates operation of the hazardous machine function(s). A1:2009 3.3.3 control guard special form of an interlocking guard which, once it has reached its closed position, gives a command to initiate the hazardous machine function(s) without the use of a separate start control
EN 1088: 1995 +A2:2008	3.3 Definition: Interlocking Guard With Guard Locking – The hazardous machine functions "covered" by the guard cannot operate until the guard is closed and locked; – The guard remains closed and locked until the risk of injury from the hazardous machine functions has passed; – When the guard is closed and locked, the hazardous machine functions "covered" by the guard can operate, but the closure and locking of the guard do not by themselves initiate their operation. 4.2.2 – Interlocking Device With Guard Locking Conditional unlocking ("four-state interlocking"), see Fig. 3 b2)
EN 954-1: 1996 ISO 13849-1:2008	5.4 Manual reset <Note: a positive edge evaluation has the same quality as a negative edge evaluation>
ISO 12100: 2010	6.2.11.4 Restart after power interruption If a hazard could be generated, the spontaneous restart of a machine when it is re-energized after power interruption shall be prevented (for example, by use of a self-maintained relay, contactor or valve).

2.1.2. Interface Description

FB Name	**SF_GuardLocking_2**
This FB controls an entrance to a hazardous area via an interlocking guard with guard locking (“four state interlocking”). This FB has extended diagnostic features compared to the SF_GuardLocking in Part 1.	
VAR_INPUT	

Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
S_GuardMonitoring	SAFEBOOL	FALSE	Variable. Monitors the guard interlocking. FALSE: Guard open. TRUE: Guard closed.
S_SafetyActive	SAFEBOOL	FALSE	Variable. Status of the hazardous area (EDM), e.g., based on speed monitoring or safe time off delay. FALSE: Machine in "non-safe" state. TRUE: Machine in safe state.
S_GuardLock	SAFEBOOL	FALSE	Variable. Status of the mechanical guard locking. FALSE: Guard is not locked. TRUE: Guard is locked.

	UnlockRequest	BOOL	FALSE	Variable. Operator intervention – request to unlock the guard. FALSE: No request. TRUE: Request made.
	S_StartReset	SAFEBOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
	S_AutoReset	SAFEBOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
	Reset	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters. Also used to request the guard to be locked again. The quality of the signal must conform to a manual reset device.
VAR_OUTPUT				
	Ready	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
	S_GuardLocked	SAFEBOOL	FALSE	Interface to hazardous area which must be stopped. FALSE: No safe state. TRUE: Safe state.
	S_UnlockGuard	SAFEBOOL	FALSE	Signal to unlock the guard. FALSE: Close guard. TRUE: Unlock guard.
	SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
	ResetRequest	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
	Error	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
	DiagCode	WORD	16#0000	See Part 1 Section 5.1.2 General Output Parameters
Notes: --				



2.1.3. Functional Description

This function controls the guard lock and monitors the position of the guard and the lock. This function block can be used with a mechanical locked switch.
The operator requests to get access to the hazardous area. The guard can only be unlocked when the hazardous area is in a safe state. The guard can be locked if the guard is closed. The machine can be started when the guard is closed and the guard is locked. An open guard or unlocked guard will be detected in the event of a safety-critical situation.

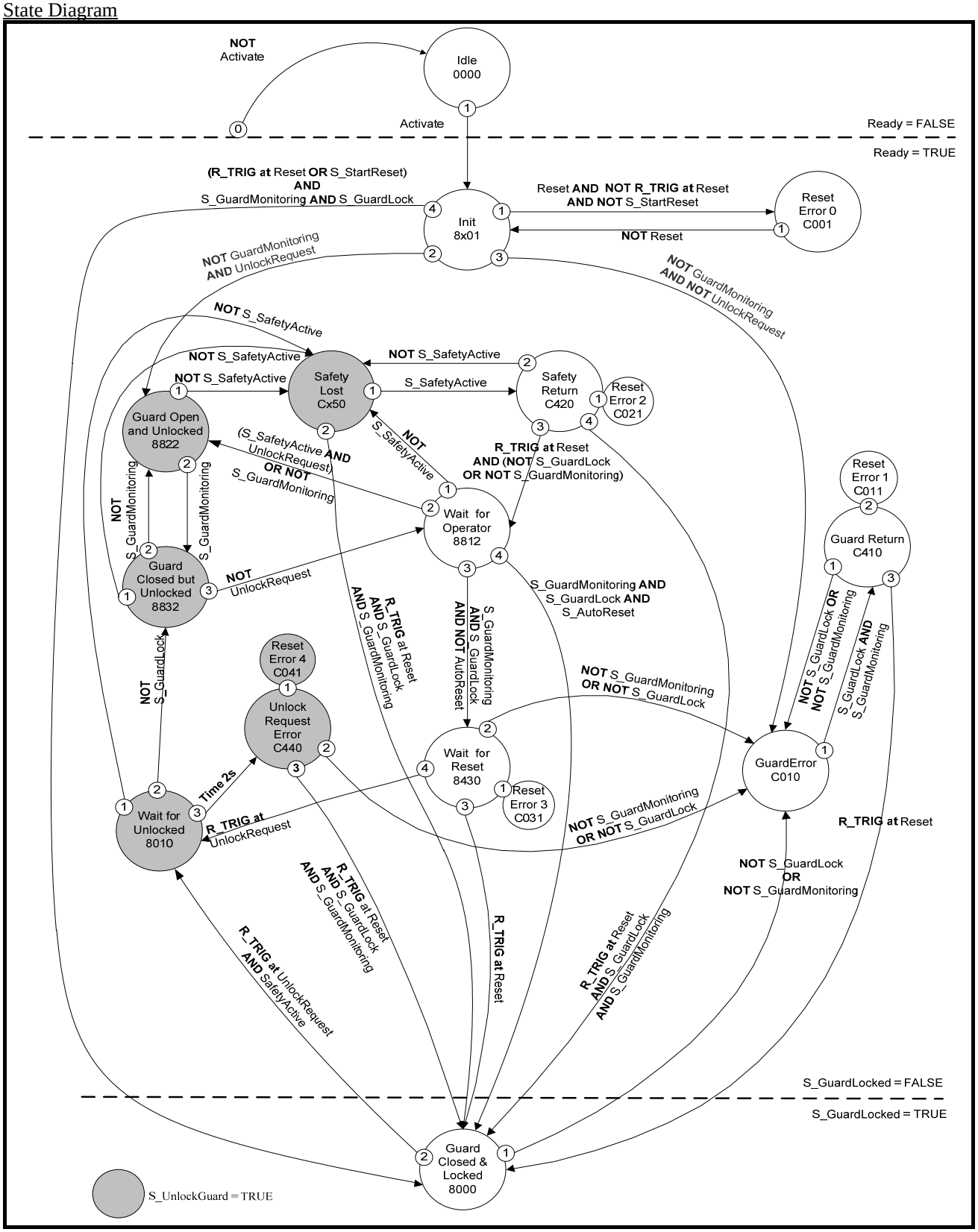
The S_StartReset and S_AutoReset inputs shall only be activated if it is ensured that no hazardous situation can occur when the PES is started.

Operation Sequence

1.	External	Request to get the hazardous area to a safe state - not part of this FB
2.	In	Feedback from applicable hazardous area that it is in a safe state (via S_SafetyActive)
3.	In	Operator request to unlock the guard (via UnlockRequest)
4.	Out	Enable guard to be opened (via S_UnlockGuard)
5.	In	Guard unlocked (via S_GuardLock). Guard can be opened now. (S_GuardLocked = FALSE)
		Operator opens the guard
6.	In	Monitoring of status guard via S_GuardMonitoring – signals when guard is closed again
7.	In	Feedback from operator to restart the hazardous area (Reset)

8.	Out	Lock guard guard (S_UnlockGuard)
9.	In	Check if guard is locked (S_GuardLock)
10.	Out	Hazardous area can operate again (S_GuardLocked = TRUE)
11.	Extern	Restart the operation in the hazardous area

State Diagram



Note: The transition from any state to the Idle state due to Activate = FALSE is not shown. However these transitions have the highest priority (0).

Figure 1: State diagram for SF_GuardLocking_2

Typical Timing Diagram:

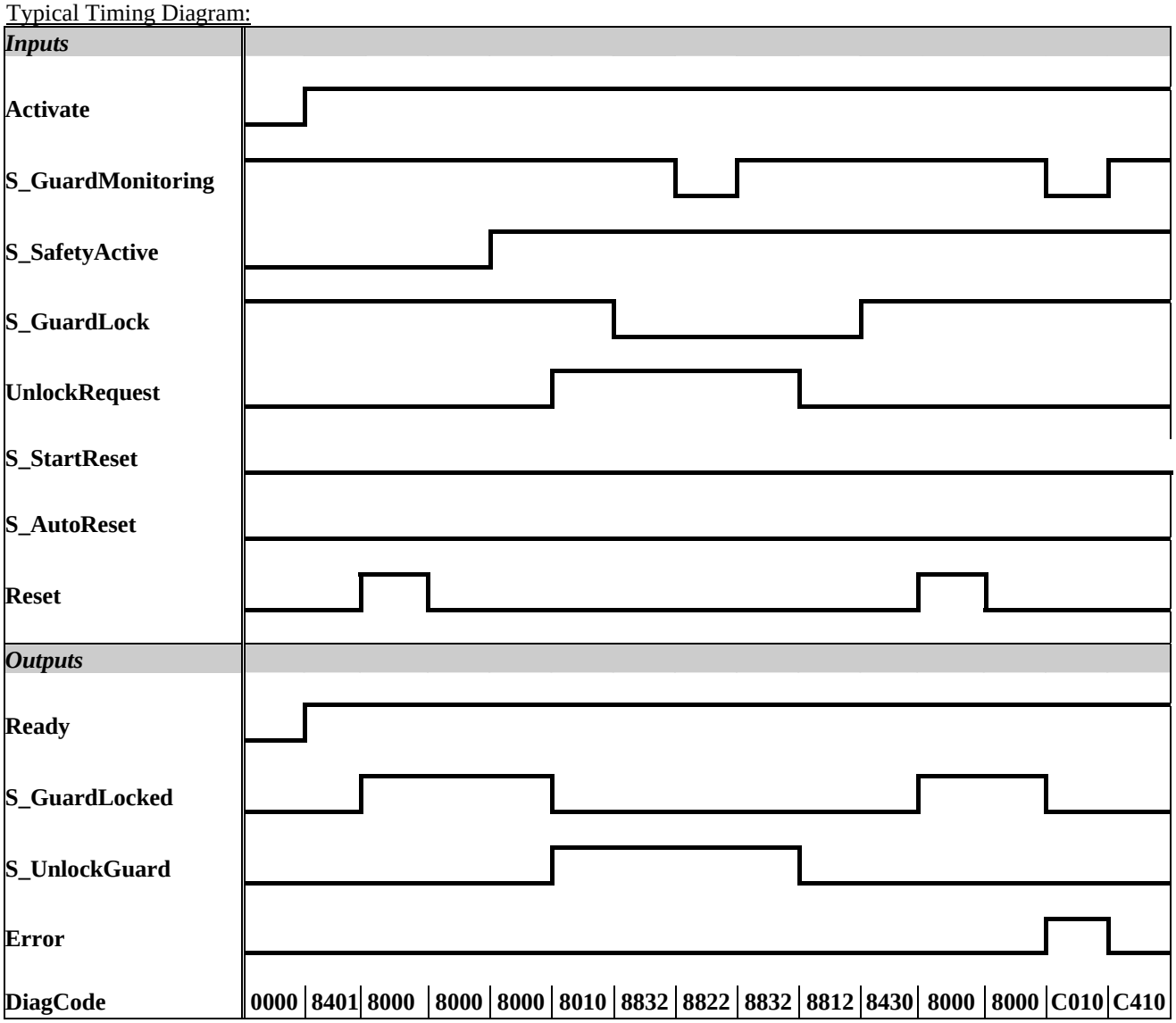


Figure 2: Timing diagram for SF_GuardLocking_2

2.1.4. Error Detection

Static signals are detected at Reset. Errors are detected at the Guard switches.

2.1.5. Error Behavior

In the event of an error the S_GuardLocked and S_UnlockGuard outputs are set to FALSE, the DiagCode output indicates the relevant error code, and the Error output is set to TRUE.
An error must be acknowledged by a rising trigger at the Reset input.

2.1.6. Function Block-Specific Error and Status Codes

FB-specific error codes:

DiagCode	State Name	State Description and Output Setting
C001	Reset Error 0	Static Reset detected in state 8x01. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C010	Guard Error	S_GuardLock and S_GuardMonitoring are not TRUE although the door was not requested to be opened. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in state C410. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C410	Guard Return	S_GuardLock and S_GuardMonitoring become TRUE again after being lost (C010) Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx50	Safety Lost	Lost safety acknowledge signal IF S_GuardMonitoring = TRUE AND S_GuardLock = TRUE THEN x = 4 ELSE x = 0 Output signals for x = 4 (C450): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE Output signals for x = 0 (C050): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

DiagCode	State Name	State Description and Output Setting
C021	Reset Error 2	Static Reset detected in state C420. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C420	Safety Return	Safety acknowledge signal becomes TRUE again after being lost (Cx50). Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C031	Reset Error 3	Static Reset detected in state 8433. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C440	Unlock Request Error	Waiting time to Unlock exceeded. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C041	Reset Error 4	Static Reset detected in state C440. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

DiagCode	State Name	State Description and Output Setting
0000	Idle	The function block is not active (initial state). Ready = FALSE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

DiagCode	State Name	State Description and Output Setting
8000	Guard Closed and Locked	Guard is closed and locked. Ready = TRUE S_GuardLocked = TRUE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8x01	Init	Function block was activated and initiated. IF S_GuardMonitoring = TRUE AND S_GuardLock = TRUE THEN x = 4 ELSE x = 8 Output signals for x = 4 (8401): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE Output signals for x = 8 (8801): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8430	Wait for Reset	Door is closed and locked, now waiting for operator reset Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8812	Wait for Operator	Waiting for operator to request to open the door (unlock request). Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8822	Guard Open and Unlocked	Lock is released and guard is open. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE

DiagCode	State Name	State Description and Output Setting
8832	Guard Closed but Unlocked	Lock is released but guard is closed. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8010	Wait for Unlocked	S_UnlockGuard is TRUE, however the acknowledge signal S_GuardLocked is still TRUE (so waiting for acknowledge <FALSE>) Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

2.2. Safety Guard Interlocking with Locking for switches with serial contacts

2.2.1. Applicable Safety Standards

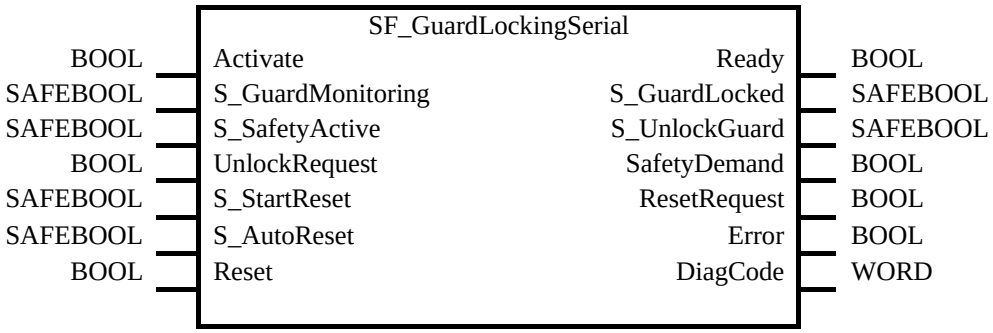
Standards	Requirements
EN 953: 1997 +A1:2009	3.3.3 Control Guard – The hazardous machine functions "covered" by the guard cannot operate until the guard is closed; – Closing the guard initiates operation of the hazardous machine function(s). A1:2009 3.3.3 control guard special form of an interlocking guard which, once it has reached its closed position, gives a command to initiate the hazardous machine function(s) without the use of a separate start control
EN 1088: 1995 +A2:2008	3.3 Definition: Interlocking Guard With Guard Locking – The hazardous machine functions "covered" by the guard cannot operate until the guard is closed and locked; – The guard remains closed and locked until the risk of injury from the hazardous machine functions has passed; – When the guard is closed and locked, the hazardous machine functions "covered" by the guard can operate, but the closure and locking of the guard do not by themselves initiate their operation. 4.2.2 – Interlocking Device With Guard Locking Conditional unlocking ("four-state interlocking"), see Fig. 3 b2)
EN 954-1: 1996 ISO 13849-1:2008	5.4 Manual reset <Note: a positive edge evaluation has the same quality as a negative edge evaluation>
ISO 12100-2: 2010	6.2.11.4 Restart after power interruption If a hazard could be generated, the spontaneous restart of a machine when it is re-energized after power interruption shall be prevented (for example, by use of a self-maintained relay, contactor or valve).

2.2.2. Interface Description

FB Name	**SF_GuardLockingSerial**
This FB controls an entrance to a hazardous area via an interlocking guard with guard locking (“four state interlocking”). The used switch does not distinguish between if the safety door is unlocked but not opened or unlocked and opened. Therefore we only have the S_GuardMonitoring input compared to SF_GuardLocking and SF_GuardLocking_2.	
VAR_INPUT	

Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
S_GuardMonitoring	SAFEBOOL	FALSE	Variable. Monitors the guard interlocking. FALSE: Guard open. TRUE: Guard closed.
S_SafetyActive	SAFEBOOL	FALSE	Variable. Status of the hazardous area (EDM), e.g., based on speed monitoring or safe time off delay. FALSE: Machine in "non-safe" state. TRUE: Machine in safe state.

	UnlockRequest	BOOL	FALSE	Variable. Operator intervention – request to unlock the guard. FALSE: No request. TRUE: Request made.
	S_StartReset	SAFEBOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
	S_AutoReset	SAFEBOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
	Reset	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters. Also used to request the guard to be locked again. The quality of the signal must conform to a manual reset device.
VAR_OUTPUT				
	Ready	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
	S_GuardLocked	SAFEBOOL	FALSE	Interface to hazardous area which must be stopped. FALSE: No safe state. TRUE: Safe state.
	S_UnlockGuard	SAFEBOOL	FALSE	Signal to unlock the guard. FALSE: Close guard. TRUE: Unlock guard.
	SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
	ResetRequest	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
	Error	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
	DiagCode	WORD	16#0000	See Part 1 Section 5.1.2 General Output Parameters
Notes: --				



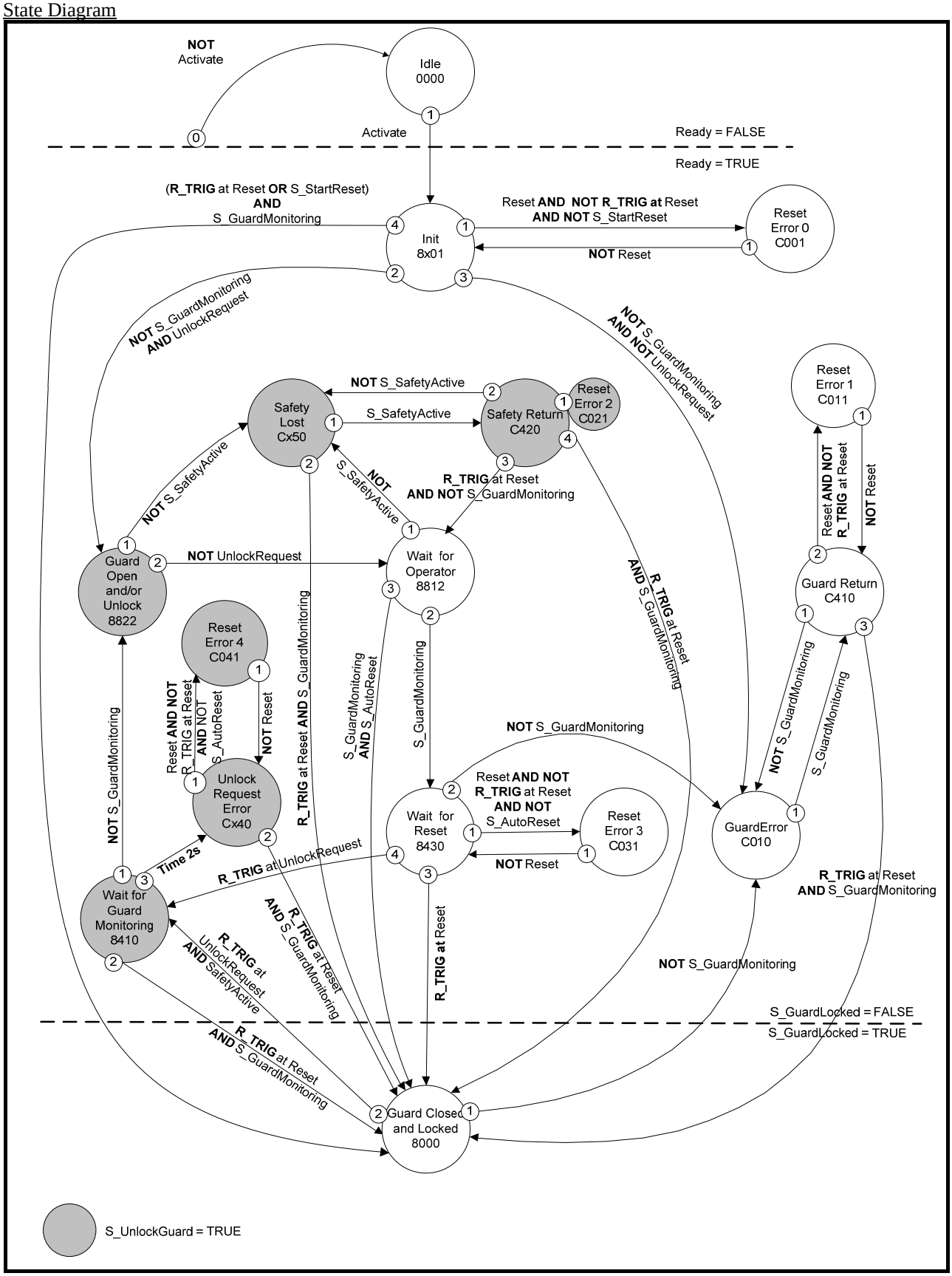
2.2.3. Functional Description

This function controls the guard lock and monitors the position of the combination of guard and lock. This function block can be used with a mechanical locked switch.
The operator requests to get access to the hazardous area. The guard can only be unlocked when the hazardous area is in a safe state. The guard can be locked if the guard is closed. The machine can be started when the guard is closed and the guard is locked. An unlocked guard will be detected to initiate a safety reaction.
The S_StartReset and S_AutoReset inputs shall only be activated if it is ensured that no hazardous situation can occur when the PES is started.

Operation Sequence

1.	External	Request to get the hazardous area to a safe state - not part of this FB
2.	In	Feedback from applicable hazardous area that it is in a safe state (via S_SafetyActive)
3.	In	Operator request to unlock the guard (via UnlockRequest)
4.	Out	Enable guard to be opened (via S_UnlockGuard)
5.	In	Guard unlocked (via S_Monitoring). Guard can be opened now. (S_GuardLocked = FALSE)
		Operator opens the guard
6.	In	Feedback from operator to restart the hazardous area (Reset)
7.	Out	Lock guard guard (S_UnlockGuard)
8.	In	Check if guard is locked (S_Monitoring)
9.	Out	Hazardous area can operate again (S_GuardLocked = TRUE)
10.	Extern	Restart the operation in the hazardous area

State Diagram



Note: The transition from any state to the Idle state due to Activate = FALSE is not shown. However these transitions have the highest priority (0).

Figure 3: State diagram for SF_GuardLockingSerial

Typical Timing Diagram

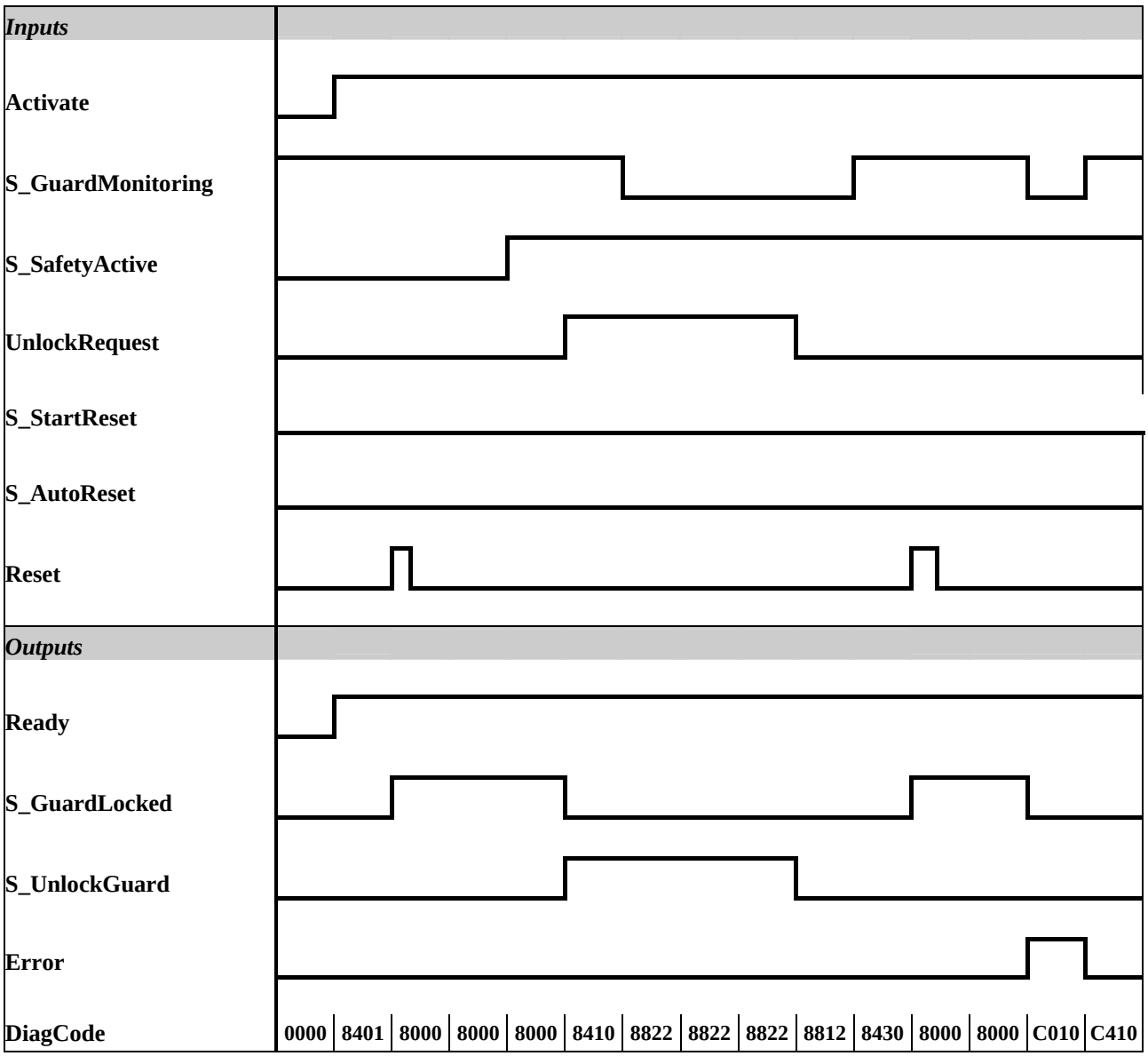


Figure 4: Timing diagram for SF_GuardLockingSerial

2.2.4. Error Detection

Static signals are detected at Reset. Errors are detected at the Guard switches.

2.2.5. Error Behavior

In the event of an error the S_GuardLocked and S_UnlockGuard outputs are set to FALSE, the DiagCode output indicates the relevant error code, and the Error output is set to TRUE.
An error must be acknowledged by a rising trigger at the Reset input.

2.2.6. Function Block-Specific Error and Status Codes

FB-specific error codes:

DiagCode	State Name	State Description and Output Setting
C001	Reset Error 0	Static Reset detected in state 8001. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C010	Guard Error	S_GuardMonitoring is not TRUE although the door was not requested to be opened. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in state C410. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C410	Guard Return	S_GuardMonitoring becomes TRUE again after being lost (C010). Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx50	Safety Lost	Lost safety acknowledge signal IF S_GuardMonitoring = TRUE THEN x = 4 ELSE x = 0 Output signals for x = 4 (C450): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE Output signals for x = 0 (C050): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

DiagCode	State Name	State Description and Output Setting
C021	Reset Error 2	Static Reset detected in state C420. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C420	Safety Return	Safety acknowledge signal becomes TRUE again after being lost (Cx50). Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C031	Reset Error 3	Static Reset detected in state 8430. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
Cx40	Unlock Request Error	Waiting time to Unlock exceeded. IF S_GuardMonitoring = TRUE THEN x = 4 ELSE x = 0 Output signals for x = 4 (C440): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE Output signals for x = 0 (C040): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C041	Reset Error 4	Static Reset detected in state Cx40. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

DiagCode	State Name	State Description and Output Setting
0000	Idle	The function block is not active (initial state). Ready = FALSE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8000	Guard Closed and Locked	Guard is closed and locked. Ready = TRUE S_GuardLocked = TRUE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8x01	Init	Function block was activated and initiated. IF S_GuardMonitoring = TRUE THEN x = 4 ELSE x = 8 Output signals for x = 4 (8401): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE Output signals for x = 8 (8801): Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8430	Wait for Reset	Door is closed and locked, now waiting for operator reset Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8812	Wait for Operator	Waiting for operator to request to open the door (unlock request). Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE

DiagCode	State Name	State Description and Output Setting
8822	Guard Open and/or Unlocked	Guard is unlocked. Door can be closed or open. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8410	Wait for Unlocked	S_UnlockGuard is TRUE, however the acknowledge signal S_GuardLocked is still TRUE (so waiting for acknowledge <FALSE>) Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE

2.3. Pressure Sensitive Equipment (PSE)

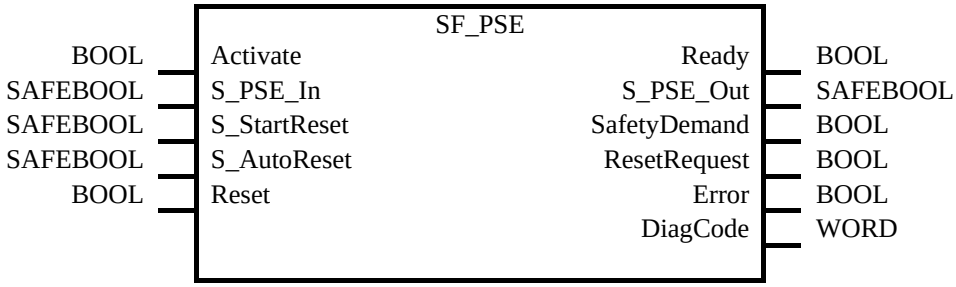
2.3.1. Applicable Safety Standards

Standards	Requirements
EN 1760-1	Pressure-sensitive protective devices Part 1: General principles for the design and testing of pressure-sensitive mats and pressure-sensitive floors 4.7 Response of output signal switching device(s) to the actuating force
EN 1760-2	Pressure-sensitive protective devices Part 2: General principles for the design and testing of pressure-sensitive edges and pressure-sensitive bars 4.11 Reset function
EN 1760-3	Pressure-sensitive protective devices Part 3: General principles for the design and testing of pressure-sensitive bumpers, plates, wires and similar devices 4.2.6.3 Reset function C.1.9 Result of sensor actuation
EN 954-1: 1996 ISO 13849-1:2008	5.4 Manual reset <Note: a positive edge evaluation has the same quality as a negative edge evaluation>
ISO 12100-2: 2010	6.2.11.4 Restart after power interruption If a hazard could be generated, the spontaneous restart of a machine when it is re-energized after power interruption shall be prevented (for example, by use of a self-maintained relay, contactor or valve).

2.3.2. Interface Description

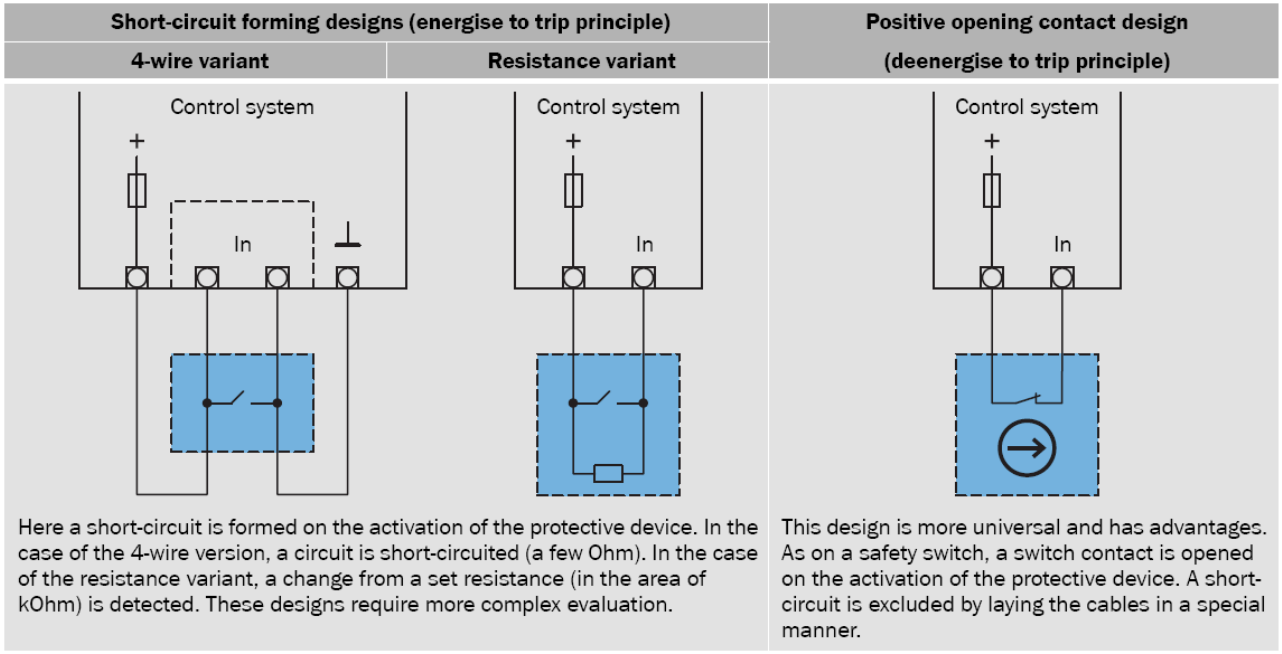
FB Name	SF_PSE		
This function block is a safety-related function block for monitoring Pressure-Sensitive-Equipment (PSE) like Safety Mats, Bumper etc.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 5.1.1 General Input Parameters
S_PSE_In	SAFEBOOL	FALSE	Safety demand input. Variable. FALSE: PSE actuated, demand for safety-related response. TRUE: PSE not actuated, no demand for safety-related response. Safety control system must be able to detect a very short interruption of the PSE (which is specified in EN 1760: minimum 200 ms), when the PSE is used in applications as a safety device.
S_StartReset	SAFEBOOL	FALSE	See Section 5.1.1 General Input Parameters
S_AutoReset	SAFEBOOL	FALSE	See Section 5.1.1 General Input Parameters
Reset	BOOL	FALSE	See Section 5.1.1 General Input Parameters
VAR_OUTPUT			
Ready	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
S_PSE_Out	SAFEBOOL	FALSE	Output for the safety-related response. FALSE: Safety output disabled. Demand for safety-related response (e.g., reset requested or internal errors active). TRUE: Safety output enabled. No demand for safety-related response.
SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
ResetRequest	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
Error	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters

DiagCode	WORD	16#0000	See Part 1 Section 5.1.2 General Output Parameters
Notes:			



2.3.3. Functional Description

This function block is a safety-related function block for monitoring Pressure-Sensitive-Equipment (PSE) like Safety Mats, Bumper etc.



Picture courtesy Sick AG

Figure 5: Overview of different configurations used in practice for PSE's

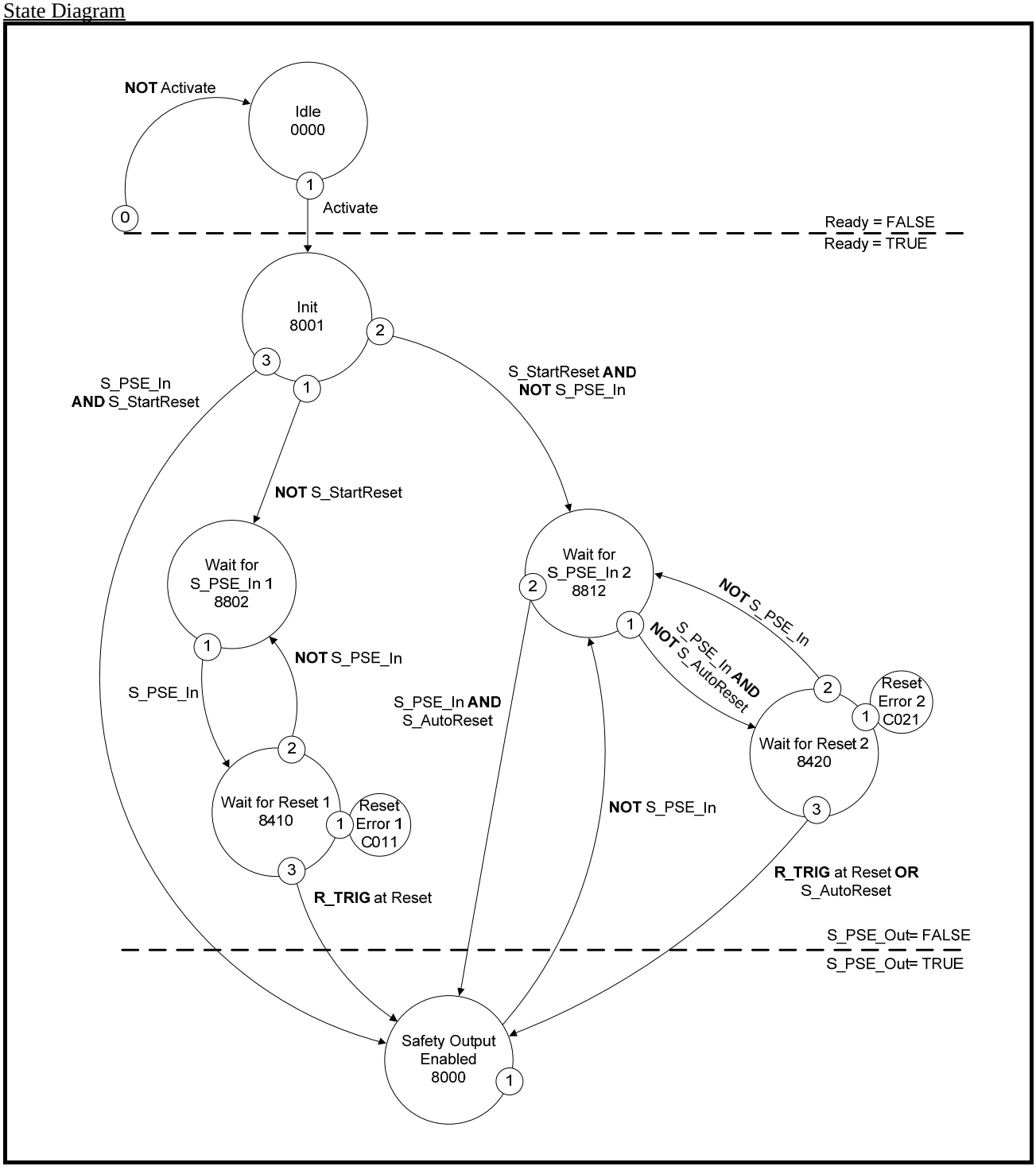
The Function Block requires a FALSE signal to activate the safety function. Therefore a PSE with positive opening contact design, as shown in the figure above on the right side, can be connected directly to a safety input device. However the other 2 principles as shown on the left require an evaluation unit to generate the applicable FALSE signal when the PSE is actuated.

The function is identical to SF_EmergencyStop (except for the 2 additional outputs SafetyDemand and ResetRequest). The S_PSE_Out output signal is set to FALSE as soon as the S_PSE_In input is set to FALSE. The S_PSE_Out output signal is set to TRUE only if the S_PSE_In input is set to TRUE and a reset occurs. The enable reset depends on the defined S_StartReset, S_AutoReset, and Reset inputs.
If S_AutoReset = TRUE, acknowledgment is automatic.
If S_AutoReset = FALSE, a rising trigger at the Reset input must be used to acknowledge the enable.
If S_StartReset = TRUE, acknowledgment is automatic the PES is started the first time.
If S_StartReset = FALSE, a rising trigger at the Reset input must be used to acknowledge the enable.

The S_StartReset and S_AutoReset inputs shall only be activated if it is ensured, that no hazardous situation can occur when the PES is started.

The SF_PSE must be selected in respect of the product standards EN 1760-1, -2 and -3 and the requested performance level according ISO 13849-1:2008.

State Diagram



Note: The transition from any state to the Idle state due to Activate = FALSE is not shown. However these transitions have the highest priority (0).

Figure 6: State diagram for SF_PSE

Typical Timing Diagrams

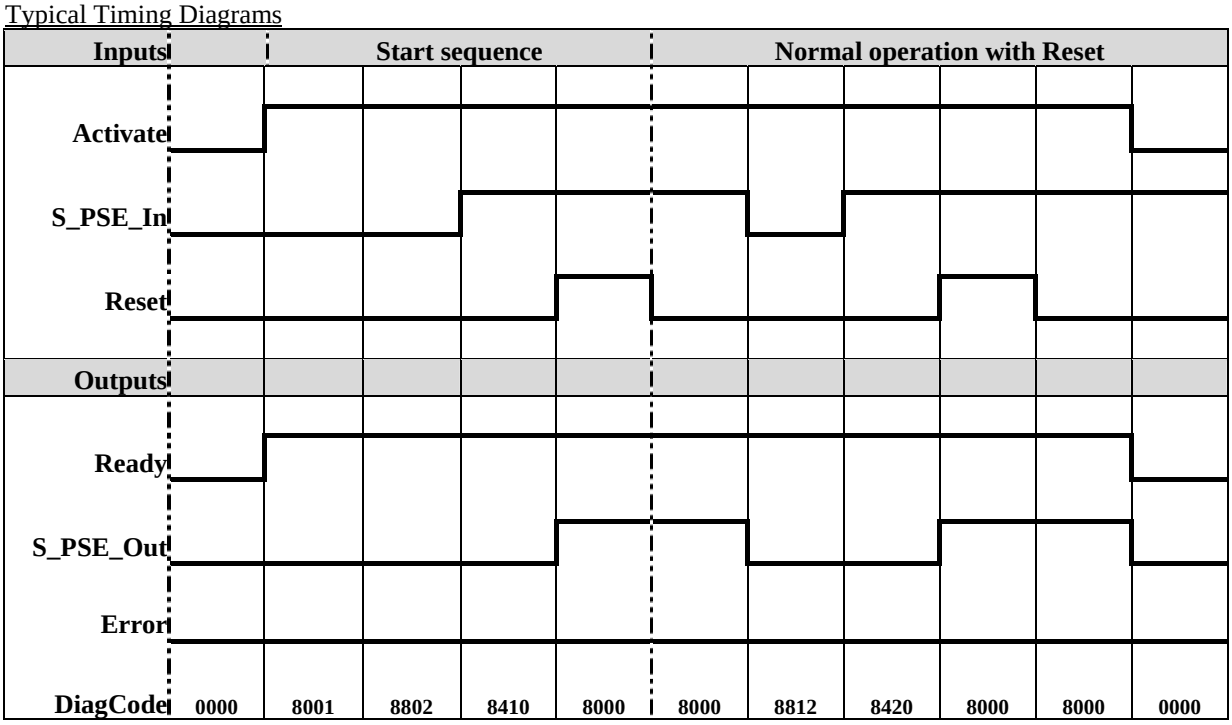


Figure 7: Timing diagram for SF_PSE: S_StartReset = FALSE; S_AutoReset = FALSE; Start, reset, normal operation, safety demand, restart

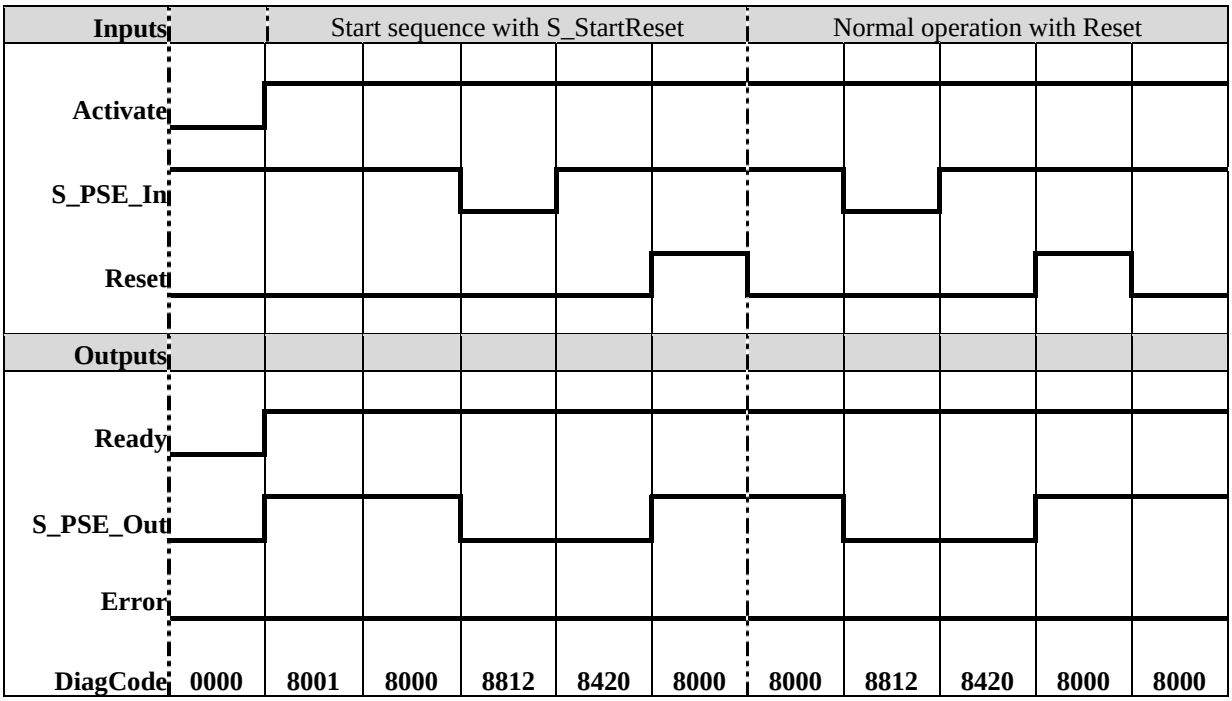


Figure 8: Timing diagram for SF_PSE: S_StartReset = TRUE, S_AutoReset = FALSE; Start, normal operation, safety demand, restart

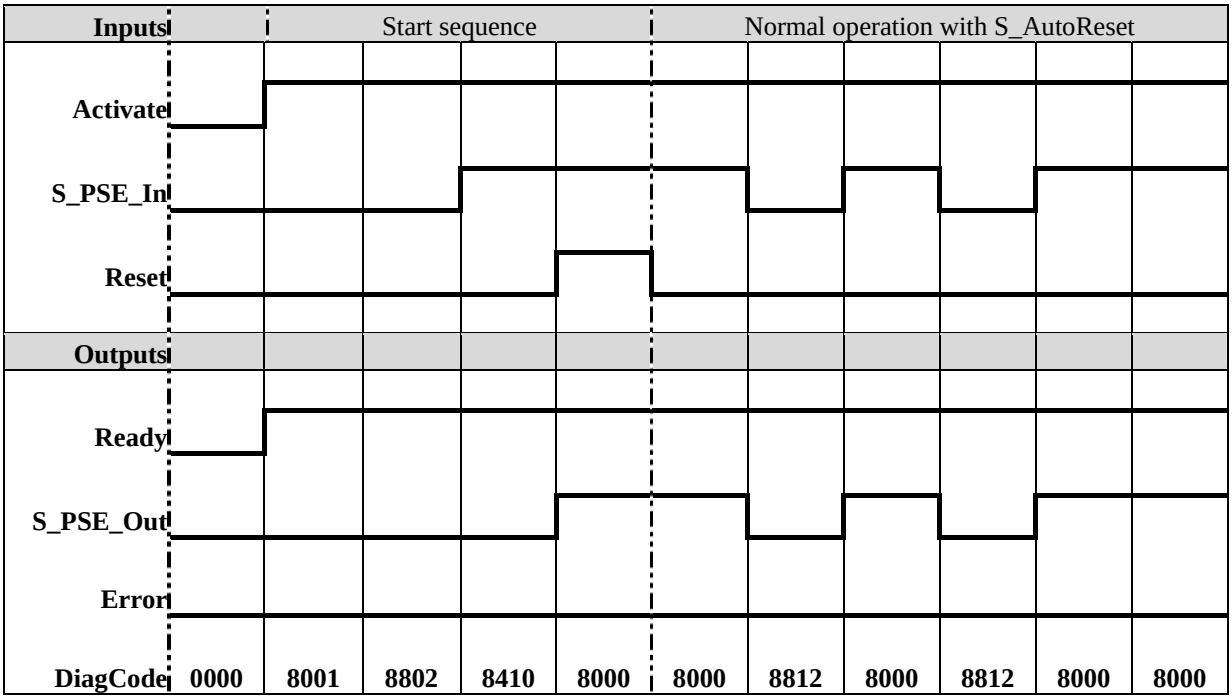


Figure 9: Timing diagram for SF_PSE: S_StartReset = FALSE, S_AutoReset = TRUE, Start, normal operation, safety demand, restart

2.3.4. Error Detection

The function block detects a static TRUE signal at Reset input.

2.3.5. Error Behavior

S_PSE_Out is set to FALSE. In case of a static TRUE signal at the Reset input, the DiagCode output indicates the relevant error code and the Error output is set to TRUE.
To leave the error states, the the Reset must be set to FALSE.

2.3.6. Function Block-Specific Error and Status Codes

FB-specific error codes:

DiagCode	State Name	State Description and Output Setting
C011	Reset Error 1	Reset is TRUE while waiting for S_PSE_In = TRUE. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Reset is TRUE while waiting for S_PSE_In = TRUE. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

DiagCode	State Name	State Description and Output Setting
0000	Idle	The function block is not active (initial state). Ready = FALSE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8001	Init	Activation is TRUE. The function block was enabled. Check if S_StartReset is requested. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8802	Wait for S_PSE_In 1	Activation is TRUE. Check if Reset is FALSE and wait for S_PSE_In = TRUE. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8410	Wait for Reset 1	Activation is TRUE. S_PSE_In = TRUE. Wait for rising trigger of Reset. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8812	Wait for S_PSE_In 2	Activation is TRUE. Safety demand detected. Check if Reset is FALSE and wait for S_PSE_In = TRUE. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8420	Wait for Reset 2	Activation is TRUE. S_PSE_In = TRUE. Check for S_AutoReset or wait for rising trigger of Reset. Ready = TRUE S_PSE_Out = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8000	Safety Output Enabled	Activation is TRUE. S_PSE_In = TRUE. Functional mode with S_PSE_Out = TRUE. Ready = TRUE S_PSE_Out = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

2.4. Diagnostic FB

The diagnostics concept is specified in Part 1 Section 5.2. The function blocks provide detailed diagnosis information regarding errors and states and contain information about transition conditions that needs to be fulfilled by the operator before a state can be left. To determine if a Reset is necessary and or applicable the diagcode WORD needs to be evaluated by the standard control. For simpler implementations it would be helpful to have the information when a Reset is necessary or a safety demand is required in general as binary information in the safety environment.

In order to provide this information the generic specification of Part 1 Section 5.1.2 General Output Parameters will be extended by the following parameters and should be considered with new implementations and further specifications.

Output Parameter		
Name	**Type**	**Description**
.....	....	…..
SafetyDemand	BOOL	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
ResetRequest	BOOL	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
Error	BOOL	see Part 1 Section 5.1.2
DiagCode	WORD	see Part 1 Section 5.1.2

Table 1: Output parameters

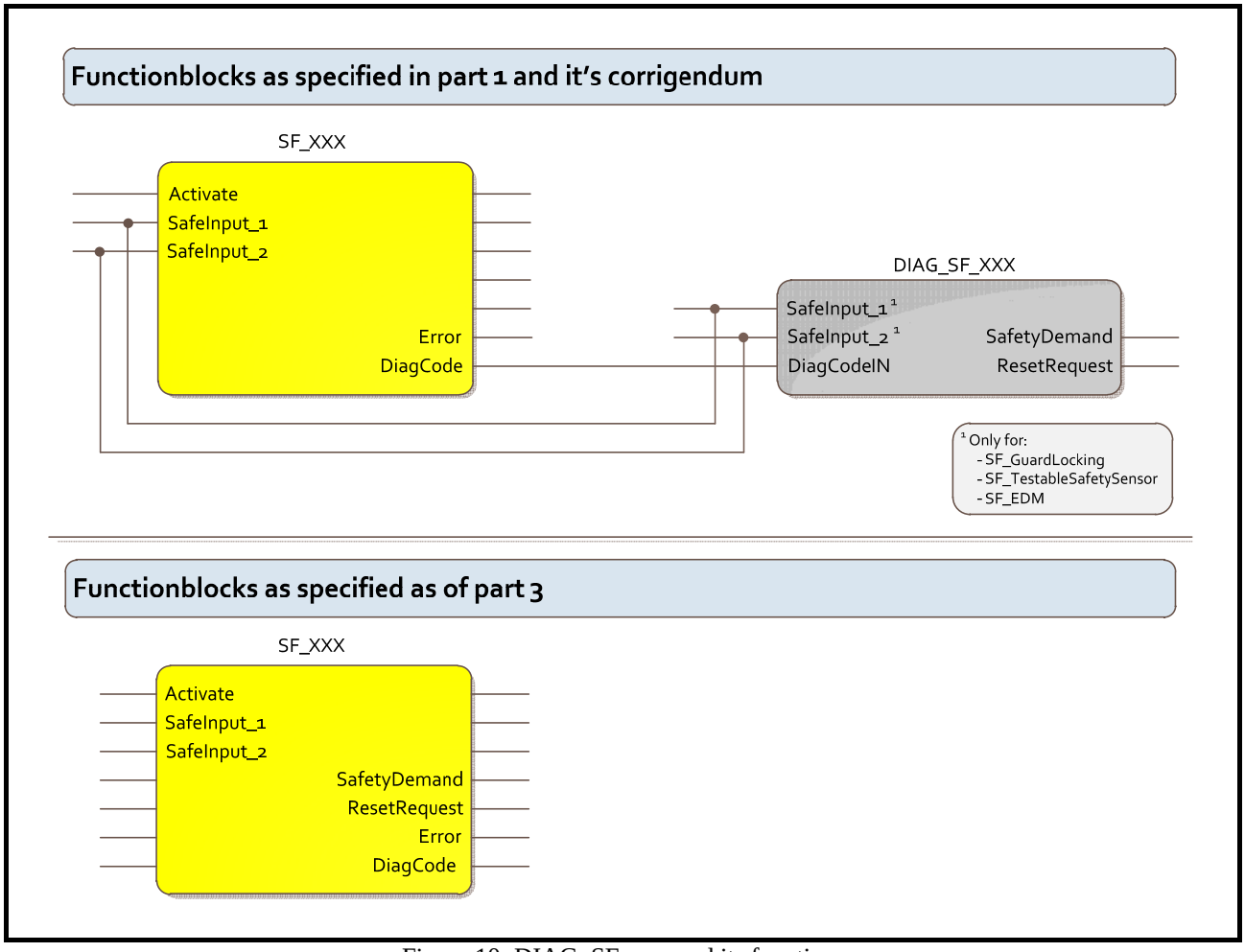


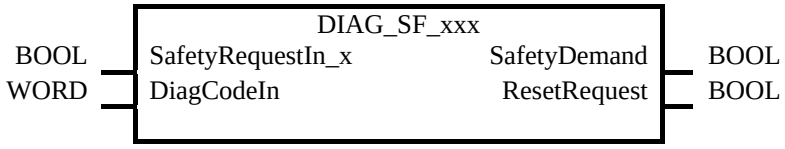
Figure 10: DIAG_SF_xxx and its function

2.4.1. Applicable Safety Standards

Not applicable since it provides operator information only

2.4.2. Interface Description

FB Name	**DIAG_SF_xxxxx**		
This function block converts the DiagCode information into a binary signal when a Reset is requested and applicable. A second output provides information if the safety chain is closed or not.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
SafetyRequestIn_x	BOOL	FALSE	If needed. See table below. Variable. Input for logical connection. x.. there might x Inputs.
DiagCodeIn	WORD	FALSE	Variable. Input for logical connection.
VAR_OUTPUT			
SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
ResetRequest	BOOL		See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
Notes: There can be more SafetyRequestIn inputs in the FB. See table below for examples.			



2.4.3. Functional Description

Function block	**Diag FB**	**Diag Code**	**ResetRequest**	**SafetyDemand**
SF_Equivalent	DIAG_SF_Equivalent	C001	FALSE	FALSE
		C002		
		C003		
		8001		TRUE
		8004		
		8005		
		8014		
		0000		FALSE
		8000		
SF_Antivalent	DIAG_SF_Antivalent	C001	FALSE	FALSE
		C002		
		C003		
		8001		TRUE
		8004		
		8005		
		8014		
		0000		FALSE
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_ModeSelector	DIAG_SF_ModeSelctor	C001	TRUE	FALSE
		C002		
		C003	FALSE	
		C004		
		8005		TRUE
		0000		FALSE
		8000		
		8004		
SF_EmergencyStop	DIAG_SF_EmergencyStop	C001	FALSE	FALSE
		C002		
		8001		
		8002		TRUE
		8003	TRUE	FALSE
		8004	FALSE	TRUE
		8005	TRUE	FALSE
		0000	FALSE	FALSE
		8000		
SF_ESPE	DIAG_SF_ESPE	C001	FALSE	FALSE
		C002		
		8001		
		8002		TRUE
		8003	TRUE	FALSE
		8004	FALSE	TRUE
		8005	TRUE	FALSE
		0000	FALSE	FALSE
		8000		
SF_SafeStop1	DIAG_SF_SafeStop1 1) defined in 'Appendix to Part 1'	C001 1)	FALSE	FALSE
		C002	TRUE	
		C003		
		C004	FALSE	
		C005		
		8001	TRUE	
		8002	FALSE	TRUE
		8003		FALSE
		8005		
		8012		TRUE
		0000		FALSE
		8000		
SF_SafeStop2	DIAG_SF_SafeStop2 1) defined in 'Appendix to part 1'	C001 1)	FALSE	FALSE
		C002	TRUE	
		C003		
		C004	FALSE	
		C005		
		8001	TRUE	
		8002	FALSE	TRUE
		8003		FALSE
		8005		
		8012		TRUE
		0000		FALSE
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_GuardMonitoring	DIAG_SF_GuardMonitoring	C001	FALSE	FALSE
		C011		
		C012		
		8001		
		8002		TRUE
		8003	TRUE	FALSE
		8004	FALSE	TRUE
		8005		FALSE
		8012		TRUE
		8014		
		0000		FALSE
		8000		
SF_SafelyLimited Speed	DIAG_SF_SafelyLimitedSpeed	C001	FALSE	FALSE
		C002	TRUE	
		C003		
		C004	FALSE	
		C005		
		8001	TRUE	
		8002	FALSE	TRUE
		8003		FALSE
		8005		
		8012		TRUE
		0000		FALSE
		8000		
		8004		
SF_TwoHand ControlTypeII	DIAG_ SF_TwoHandControlTypeII	C001	FALSE	FALSE
		C002		
		C003		
		8001		
		8004		TRUE
		8005		
		8006		
		8007		
		8008		
		8009		
		8019		
		0000		FALSE
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_TwoHand ControlTypeIII	DIAG_ SF_TwoHandControlTypeIII	C001	FALSE	FALSE
		C002		
		C003		
		C004		
		C005		
		C006		
		8001		
		8004		TRUE
		8005		
		8006		
		8007		
		8008		
		8009		
		8019		
		0000		FALSE
		8000		
SF_GuardLocking	DIAG_SF_GuardLocking This FB uses 2 inputs of SafetyRequestIn_x IF S_GuardMonitoring AND S_GuardLock = TRUE: R = TRUE ELSE R = FALSE [1)] See chapter 1.2 Harmonization of diagnostic codes for new function blocks	C001	FALSE	FALSE
		C002		
		C003		
		C004		
		8001	R [1)]	
		8003	TRUE	
		8011	FALSE	TRUE
		8012		
		8013		FALSE
		8014	TRUE	
		0000	FALSE	
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_TestableSafety Sensor	DIAG_SF_TestableSafetySensor This FB uses 2 inputs of SafetyRequestIn_x IF S_OSSD_IN = TRUE AND NoExternalTest = TRUE THEN R = TRUE ELSE R = FALSE [1)] See chapter 1.2 Harmonization of diagnostic codes for new function blocks	C000	FALSE	FALSE
		C001		
		C002		
		C003		
		C004		
		C005		
		C006		
		C007		
		C010	R [1)]	
		C020	R [1)]	
		8001	TRUE	
		8002	FALSE	TRUE
		8003	TRUE	FALSE
		8004	FALSE	
		8005		TRUE
		8006	TRUE	FALSE
		8012	FALSE	TRUE
		8013	TRUE	FALSE
		8010	FALSE	
		8020		
		8030		
		0000		
		8000		
SF_MutingSeq	DIAG_SF_MutingSeq	C001	FALSE	FALSE
		C002		
		C003		
		CYx4		
		C005		
		C006		
		8001	TRUE	
		8002	FALSE	TRUE
		8003	TRUE	FALSE
		8005	FALSE	
		8011		
		8012		
		8122		
		8112		
		0000		
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_MutingPar	DIAG_SF_MutingPar	C001	FALSE	FALSE
		C002		
		C003		
		CYx4		
		C005		
		C006		
		C007		
		C008		
		8001	TRUE	
		8002	FALSE	TRUE
		8003	TRUE	FALSE
		8005	FALSE	
		8011		
		8012		
		8014		
		8021		
		8122		
		8121		
		8114		
		8112		
		8311		
		8314		
		8414		
		8422		
		0000		
		8000		
SF_MutingPar_ 2Sensor	DIAG_SF_MutingPar_2Sensor	C001	FALSE	FALSE
		C002		
		C003		
		CYx4		
		C005		
		C006		
		C007		
		8001	TRUE	
		8002	FALSE	TRUE
		8003	TRUE	FALSE
		8005	FALSE	
		8011		
		8012		
		8311		
		0000		
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand
SF_EnableSwitch	DIAG_SF_EnableSwitch	C001	FALSE	FALSE
		C002		
		C010		
		C020	TRUE	
		C030	FALSE	
		C040	TRUE	
		8004	FALSE	
		8005		
		8006		TRUE
		8007		
		0000		FALSE
		8000		
SF_SafetyRequest	DIAG_SF_SafetyRequest [1] defined in 'Appendix to Part 1'	C001 [1]	FALSE	FALSE
		C002	TRUE	
		C003		
		C004	FALSE	
		C005		
		8001	TRUE	
		8002	FALSE	TRUE
		8003		FALSE
		8005		
		8012		TRUE
		0000		FALSE
		8000		
SF_OutControl	DIAG SF_OutControl	C001	FALSE	FALSE
		C002		
		C010		
		C111		
		C211		
		8001	TRUE	
		8002	FALSE	TRUE
		8003	TRUE	FALSE
		8010	FALSE	
		0000		
		8000		

Function block	Diag FB	Diag Code	ResetRequest	SafetyDemand

SF_EDM	DIAG_SF_EDM
	This FB uses 2 inputs of SafetyRequestIn_x
	
	IF EDM_1 = TRUE AND EDM_2 = TRUE THEN
	R = TRUE
	
	ELSE
	R = FALSE
	
	[1] See chapter 1.2 Harmonization of diagnostic codes for new function blocks
		C001	FALSE	FALSE

		C111		
		C010	R[1]	
		C020		
		C030		
		C011	FALSE	
		C021		
		C031		
		C040	R[1]	
		C050		
		C060		
		C041	FALSE	
		C051		
		C061		
		C070	TRUE	
		C080		
		C090		
		C071	FALSE	
		C081		
		C091		
		8001	TRUE	
		8010	FALSE	TRUE
		0000		FALSE
		8000		

2.5. SF_Override

2.5.1. Applicable Safety standards

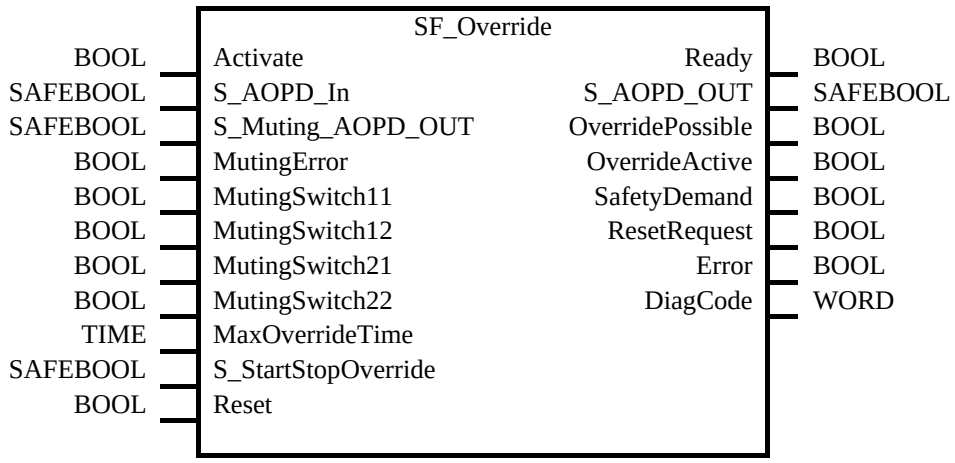
Standards	Requirements
EN IEC 62046:2008	5.5.4 Mute dependent override A manually operated, mute dependent override function can be necessary to allow blockages to be removed from the detection zone of the protective equipment. When a mute dependent override function is active, access to the hazardous zone can be possible without actuating the trip function. Mute dependent override shall permit operation of the hazardous elements only in reduced risk conditions. For details of reduced risk conditions see ISO 12100-2, 4.11.9. When a product or transport unit is stopped in the detection zone of the ESPE or of the muting sensors, the muting function shall be cancelled and all dangerous action once safe operation conditions have been re-established. The override function shall be enabled only when the output of the ESPE is in the OFF-state and/or at least one muting sensor is actuated. From a lockout condition (when a dangerous fault is detected) it shall not be possible to actuate the override function. The mute dependent override function shall: • be activated either: - by the use of a spring return hold-to-run device located so that is not possible to enter the hazardous zone whilst maintaining the action on the hold-to-run device, and so that the hazardous zone is visible while actuating the device; - or by the use of a key operated switch or equally secure momentary action pushbutton when: - the override function is automatically terminated after a correct muting signal sequence is identified, and - no access to the hazardous zone is possible during tho override sequence; - an emergency stop can be initiated from the same position. • only be activated when at least one of the muting sensors is actuated; • automatically terminate when all the muting sensors are de-actuated; • automatically terminate after a pre-determined time limit has expired; • only enable those movements that are necessary to permit blockages to be removed from the detection zone of the protective equipment. Measures shall be provided to prevent activation of the mute dependent override function due to a fault or inadvertent operation of the initiating device.
EN 954-1: 1996 ISO 13849-1:2008	5.4 Manual reset <Note: a positive edge evaluation has the same quality as a negative edge evaluation>

Standards	Requirements
EN ISO 12100-2010	6.2.11.9 Control mode for setting, teaching, process changeover, fault-finding, cleaning or maintenance Where, for setting, teaching, process changeover, fault-finding, cleaning or maintenance of machinery, a guard has to be displaced or removed and/or a protective device has to be disabled, and where it is necessary for the purpose of these operations for the machinery or part of the machinery to be put into operation, the safety of the operator shall be achieved using a specific control mode which simultaneously a) disables all other control modes, b) permits operation of the hazardous elements only by continuous actuation of an enabling device, a two-hand control device or a hold-to-run control device, c) permits operation of the hazardous elements only in reduced risk conditions (for example, reduced speed, reduced power/force, step-by-step, for example, with a limited movement control device), and d) prevents any operation of hazardous functions by voluntary or involuntary action on the machine's sensors. NOTE For some special machinery other protective measures can be appropriate. This control mode shall be associated with one or more of the following measures: - restriction of access to the danger zone as far as possible; - emergency stop control within immediate reach of the operator; - portable control unit (teach pendant) and/or local controls (allowing sight of the controlled elements). See IEC 60204-1.

2.5.2. Interface description

FB-Name	SF_Override		
This FB makes it possible to move a product in the production line even when the sequential muting was aborted due to an error. This FB is only applicable in combination with a muting FB.			
VAR_INPUT			
Name	*Data type*	*Initial value*	*Description, Parameter values*
Activate	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
S_AOPD_In	SAFEBOOL	FALSE	Variable. OSSD signal from AOPD. FALSE: Protection field interrupted. TRUE: Protection field not interrupted.
S_Muting_AOPD_OUT	SAFEBOOL	FALSE	Variable. S_AOPD_OUT signal from the previous muting function block. FALSE/ TURE: The Status of the Safety related output S_AOPD_OUT from the previous muting function block.
MutingError	BOOL	FALSE	Error output of the previous connected Muting-FB FALSE: No error TRUE: Error in Muting Process
Muting-Switch11	BOOL	FALSE	Variable. Status of the Muting sensor signal which is connected at the input MutingSwitch11 of the previous muting function block. FALSE: Muting sensor 11 not actuated. TRUE: Workpiece actuates muting sensor 11.
Muting-Switch12	BOOL	FALSE	Variable. Status of the Muting sensor signal which is connected at the input MutingSwitch12 of the previous muting function block. FALSE: Muting sensor 12 not actuated. TRUE: Workpiece actuates muting sensor 12.

Muting-Switch21	BOOL	FALSE	Variable. Status of the Muting sensor signal which is connected at the input MutingSwitch21 of the previous muting function block. FALSE: Muting sensor 21 not actuated. TRUE: Workpiece actuates muting sensor 21. It shall be noted that this parameter is not connected if the previous muting function is the SF_MutingPar_2Sensor.
Muting-Switch22	BOOL	FALSE	Variable. Status of the Muting sensor signal which is connected at the input MutingSwitch22 of the previous muting function block. FALSE: Muting sensor 22 not actuated. TRUE: Workpiece actuates muting sensor 22. It shall be noted that this parameter is not connected if the previous muting function is the SF_MutingPar_2Sensor.
MaxOverrideTime	Time	T#0s	Constant 0..10 min; Maximum time for the overall Override proces. The time is started when the start conditions for the override proces are available. The timer is stopped when all the muting sensors are not muted anymore.
S_StartStopOverride	SAFEBOOL	FALSE	Signal for the start and stop of override functionality. A rising edge is needed to start the override functionality. TRUE: If all override conditions are fulfilled, the override process starts. At this moment also the timer for the MaxOverrideTime starts. FALSE: The override process stops. The timer for the MaxOverrideTime continues till the muting process is finished (transition from 8832 to 8802).
Reset	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
VAR_OUTPUT			
Ready	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
S_AOPD_OUT	SAFEBOOL	FALSE	Safety related output, indicates status of the muted guard or override signal. FALSE: AOPD protection field interrupted and muting not active or override is not active. TRUE: AOPD protection field not interrupted or muting active or override is active.
OverridePossible	BOOL	FALSE	Status signaling that override is possible FALSE: Override not possible TRUE: Override possible
OverrideActive	BOOL	FALSE	Indicates the status of Override process. FALSE: Override not active. TRUE: Override active.
SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
ResetRequest	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1.
Error	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
DiagCode	WORD	16#0000	See Part 1 Section 5.1.2 General Output Parameters
Notes: -			



2.5.3. Functional Description

A manual operated override function can be required to remove blockades in the safety area which resulted during the muting process. If override is active a stop request of the safety equipment is not effective.

This FB SF_Override is only to be used in combination with a muting FB.
In the application program itself, first the muting FB must be processed and then the override FB.

Notice: The Outputs Error and DiagCode of the preconnected Muting are not transmitted to the Outputs Error and DiagCode of the FB SF_Override

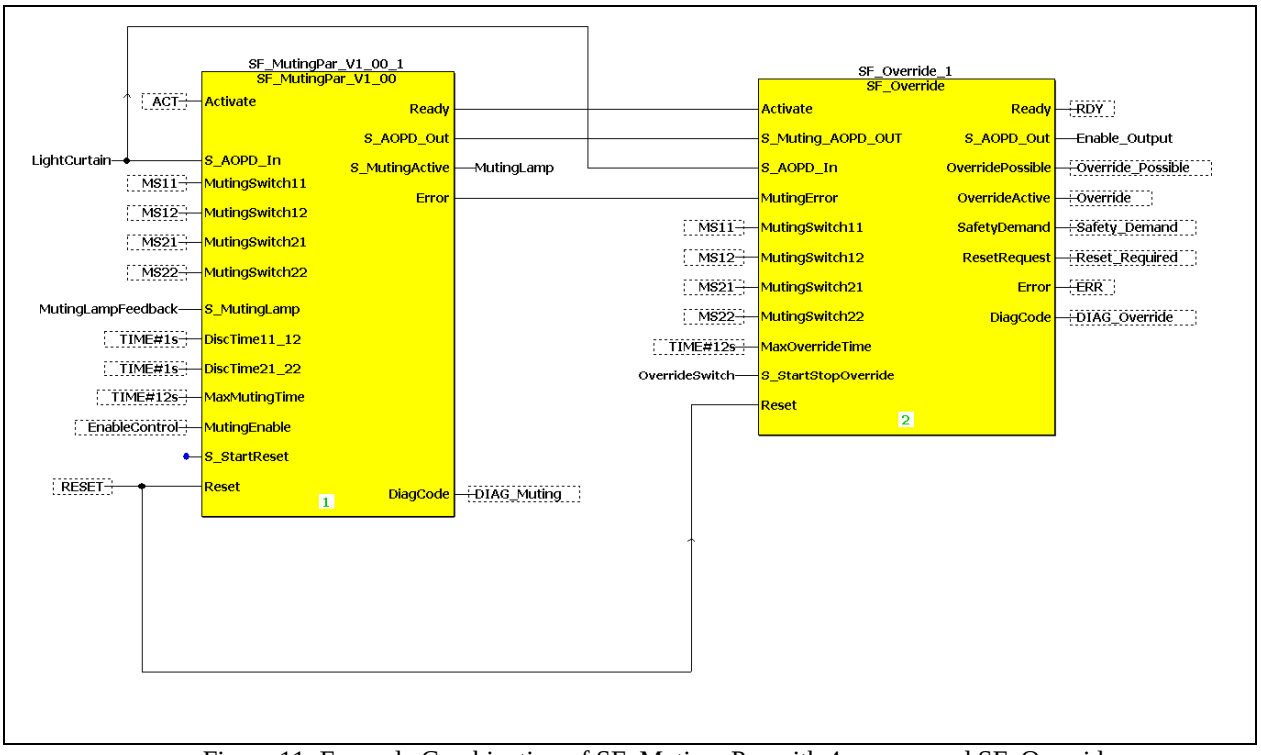


Figure 11: Example Combination of SF_Muting_Par with 4 sensors and SF_Override

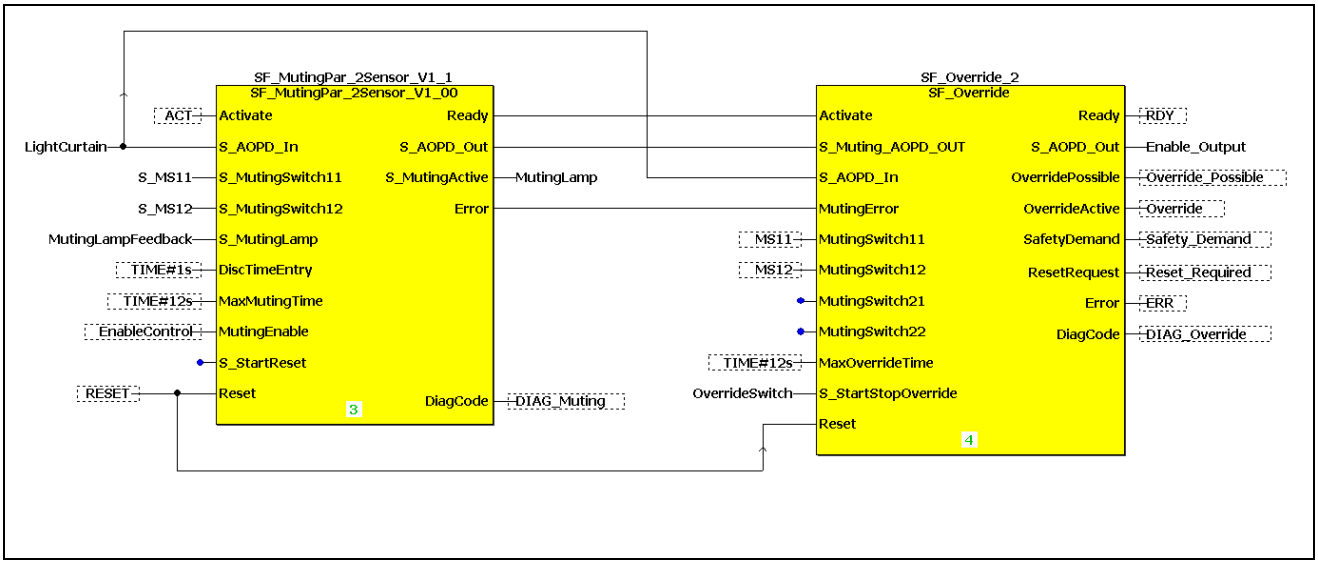


Figure 12: Example Combination of SF_Muting_Par_2Sensor and SF_Override

The override signal (S_AOPD_Out of the SF_Override FB) is set by the FB if:
- the pre-connected muting FB shows an error
- an applicable S_StartStopOverride signal has a rising edge and a static TRUE
- the safeguard (e.g. light curtain) is interrupted and/or
- at least one muting sensor is blocked

The override signal (S_AOPD_Out of the SF_Override FB) is reset by the FB if:
- all muting sensors are ‘clear’ and the safeguard (e.g. light curtain) is not interrupted
- the applicable maximum override time is expired
- the S_StartStopOverride signal is FALSE.

State diagram

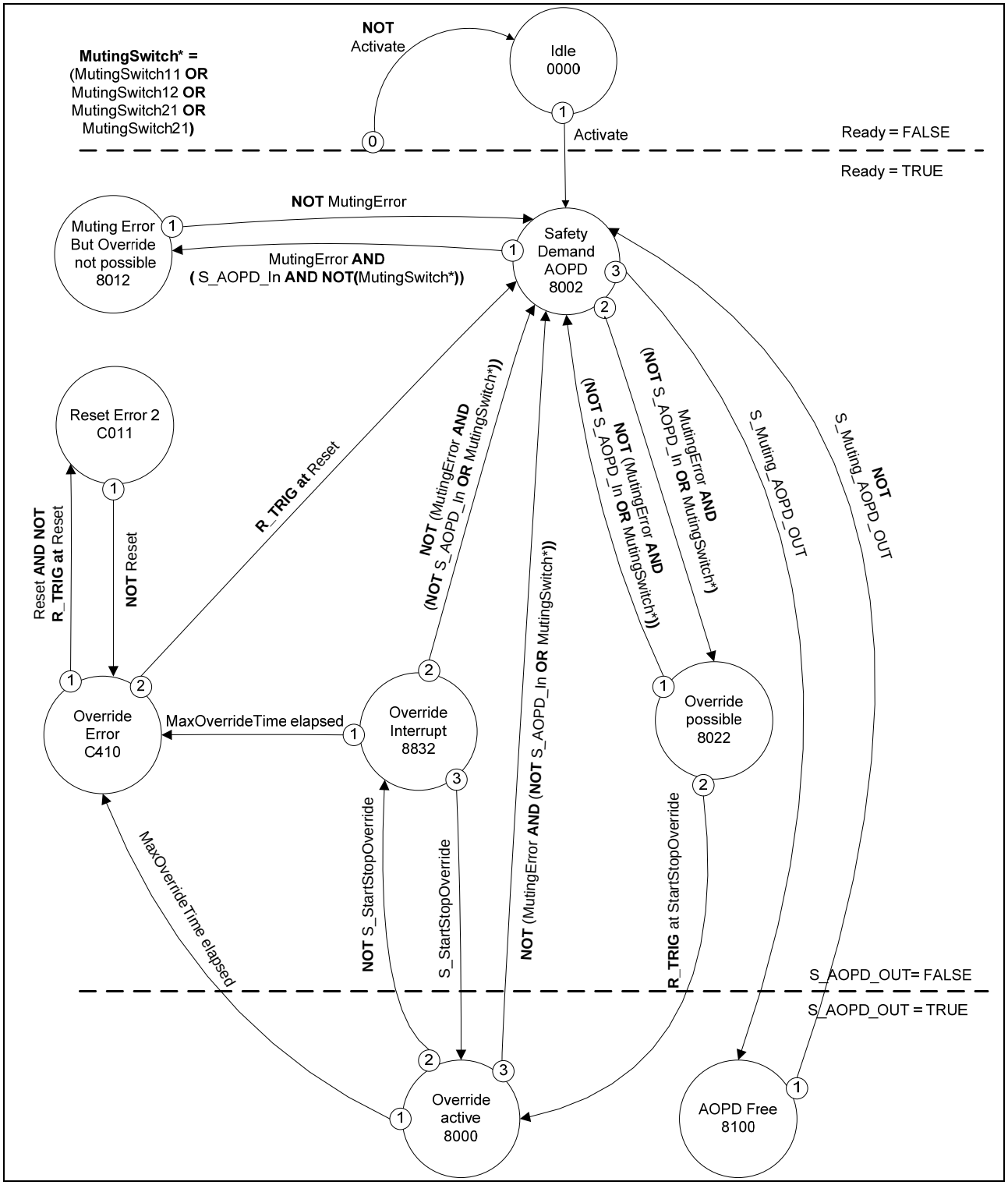


Figure 13: State diagram SF_Override

2.5.4. Function Block-Specific Error and Status Codes

FB-specific error codes:

DiagCode	State Name	State Description and Output Setting
C011	Reset Error 2	Static Reset condition detected after FB activation. Ready = TRUE S_AOPD_Out = FALSE OverridePossible = FALSE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C410	Override Error 2	Max Override time elapsed Ready = TRUE S_AOPD_Out = FALSE OverridePossible = FALSE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific status codes (no error):

DiagCode	State Name	State Description and Output Setting
8002	Safety Demand AOPD	Protection field interrupted and muting not active or override is not active and the timer for the MaxOverrideTime will be reset. Ready = TRUE S_AOPD_Out = FALSE OverridePossible = FALSE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8012	Muting Error but Override not possible	The pre-connected muting FB shows an error but the safeguard (e.g. light curtain) is not interrupted and no muting sensor is blocked. Ready = TRUE S_AOPD_Out = FALSE OverridePossible = FALSE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8022	Override Possible	The pre-connected muting FB shows an error and the safeguard (e.g. light curtain) is interrupted and/or at least one muting sensor is blocked Ready = TRUE S_AOPD_Out = FALSE OverridePossible = TRUE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

DiagCode	State Name	State Description and Output Setting
8832	Override Interrupt	The override start signal is set to FALSE during override process. The time for the MaxOverrideTime is still running. Ready = TRUE S_AOPD_Out = FALSE OverridePossible = TRUE OverrideActive = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8000	Override Active	Override is active and the timer for the MaxOverrideTime is starting to run. Ready = TRUE S_AOPD_Out = TRUE OverridePossible = TRUE OverrideActive = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8100	AOPD Free	S_AOPD_Out from the pre-connected function block is TRUE Ready = TRUE S_AOPD_Out = TRUE OverridePossible = FALSE OverrideActive = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

Typical Timing Diagram

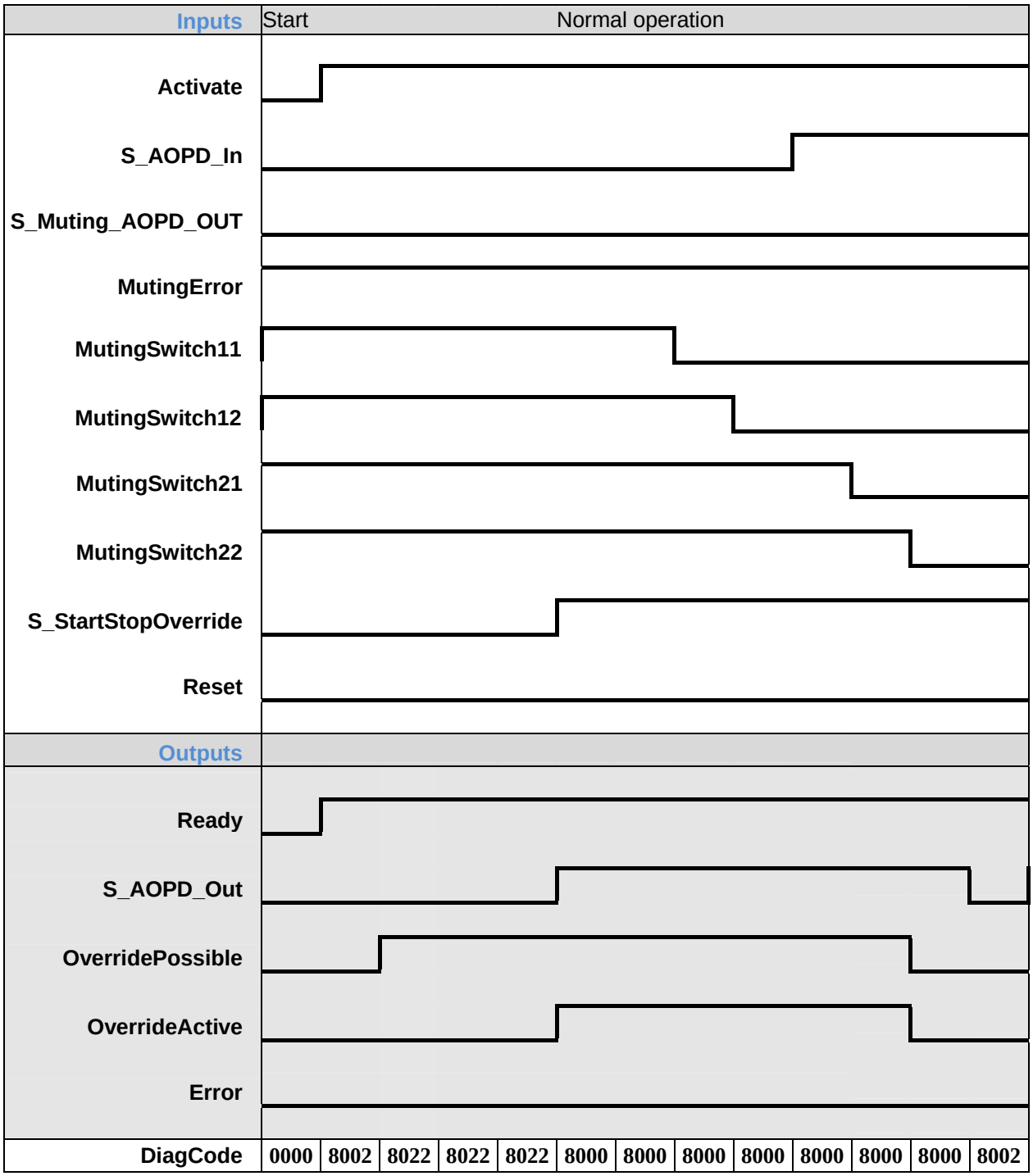


Figure 14: Timing Diagram of SF_Override with parallel muting (cf. Figure 11)

Note: SafetyDemand and ResetRequest are not shown in the Timing Diagram

This diagram shows the functionality of the Overwrite FB in combination with sequential muting. This is visible in the transition of the muting inputs while in state 8000. This is related to the moving of the object in the muted area.

2.6. SF_EnableSwitch 2 (without detection of panic position)

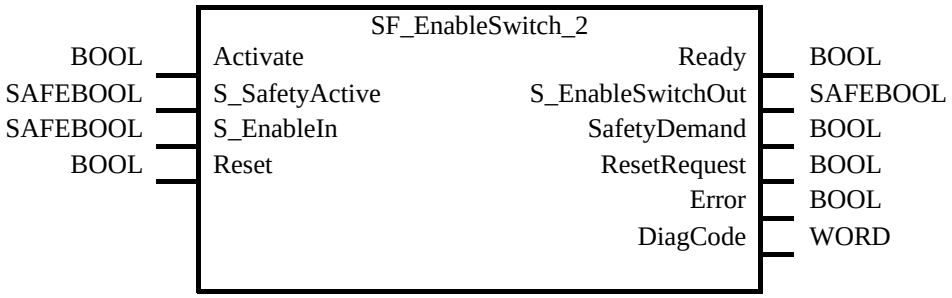
2.6.1. Applicable Safety Standards

Standards	Requirements
IEC 60204-1, Ed. 5.0: 2003	9.2.6.3: Enabling control (see also 10.9) is a manually activated control function interlock that: a) when activated allows a machine operation to be initiated by a separate start control, and b) when de-activated – initiates a stop function, and – prevents initiation of machine operation. Enabling control shall be so arranged as to minimize the possibility of defeating, for example by requiring the de-activation of the enabling control device before machine operation may be reinitiated. It should not be possible to defeat the enabling function by simple means. 10.9: When an enabling control device is provided as a part of a system, it shall signal the enabling control to allow operation when actuated in one position only. In any other position, operation shall be stopped or prevented. Enabling control devices shall be selected that have the following features: … – for a two-position type: - position 1: off-function of the switch (actuator is not operated); - position 2: enabling function (actuator is operated); – for a three-position type: - position 1: off-function of the switch (actuator is not operated); - position 2: enabling function (actuator is operated in its mid position); - position 3: off-function (actuator is operated past its mid position); - when returning from position 3 to position 2, the enabling function is not activated.
EN 954-1: 1996 ISO 13849-1:2008	5.4 Manual reset <Note: a positive edge evaluation has the same quality as a negative edge evaluation>
ISO 12100- 2010	6.2.11.4 Restart after power interruption If a hazard could be generated, the spontaneous restart of a machine when it is re-energized after power interruption shall be prevented (for example, by use of a self-maintained relay, contactor or valve).

Note: Many three position switches are wired internally and do not provide an external contact for evaluating the panic position (position 3). If a position switch is used that offers an external contact to evaluate externally the position 3, the SF_EnableSwitch shall be used.

2.6.2. Interface Description

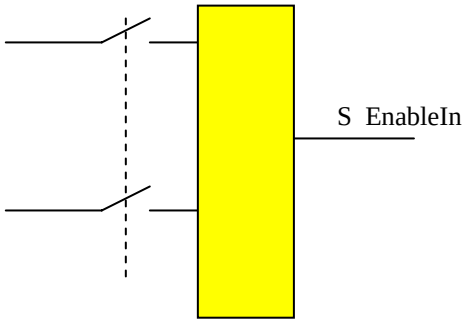
FB Name	**SF_EnableSwitch_2**		
The SF_EnableSwitch FB_2 evaluates the signals of an enable switch with two or three positions.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, parameter values*
Activate	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
S_SafetyActive	SAFEBOOL	FALSE	Variable or constant. Confirmation of the safe mode (limitation of the speed or the power of motion, limitation of the range of motion). FALSE: Safe mode is not active. TRUE: Safe mode is active.
S_EnableIn	SAFEBOOL	FALSE	Variable. Signal of connected enable switch. The evaluation of the signals (discrepancy) will be done within the IO unit or the FB_Equivalent FALSE: Not Enabled. TRUE: Enabled.
Reset	BOOL	FALSE	See Part 1 Section 5.1.1 General Input Parameters
VAR_OUTPUT			
Ready	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
S_EnableSwitchOut	SAFEBOOL	FALSE	Safety related output: Indicates suspension of guard. FALSE: Disable suspension of safeguarding. TRUE: Enable suspension of safeguarding.
SafetyDemand	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1
ResetRequest	BOOL	FALSE	See Part 3, section 1.1 Extensions to General Output Parameters of Part 1
Error	BOOL	FALSE	See Part 1 Section 5.1.2 General Output Parameters
DiagCode	WORD	16#0000	See Part 1 Section 5.1.2 General Output Parameters
Notes: -			



2.6.3. Functional Description

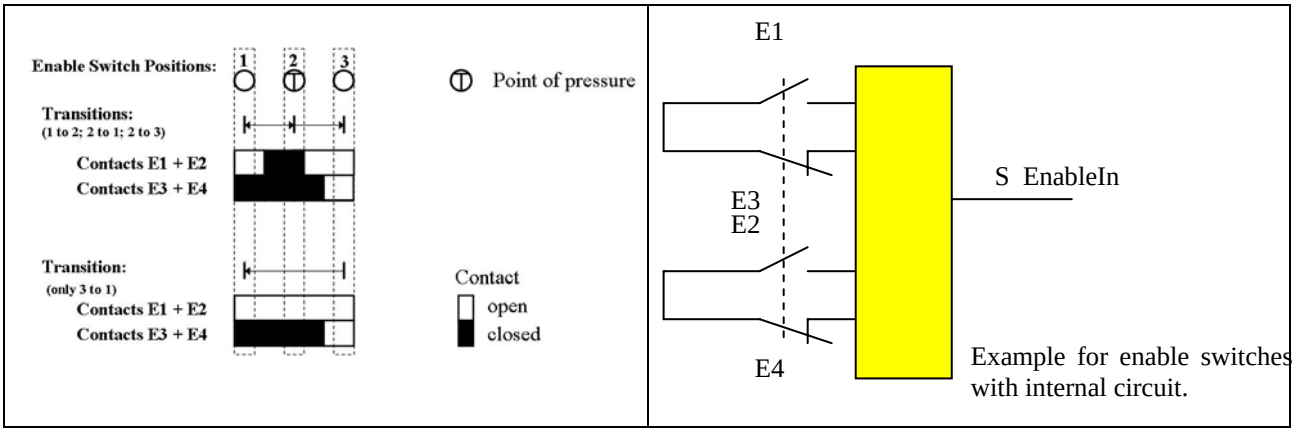
The SF_EnableSwitch_2 FB supports the suspension of safeguarding (DIN EN 60204 Section 9.2.4) using enable switches (DIN EN 60204 Section 9.2.5.8), if the relevant operating mode is selected and active. The relevant operating mode (limitation of the speed or the power of motion, limitation of the range of motion) must be selected outside the SF_EnableSwitch_2 FB.
The SF_EnableSwitch_2 FB evaluates the signals of an enable switch with two or three positions (DIN EN 60204 Section 9.2.5.8).

Two position switch



Three position switch

There is an internal circuit between the normally closed and normally open contacts as shown below. The output is either HIGH if the enable switch is in Pos 2 or LOW if either the enable switch is released (Pos1) or in the panic position (Pos3).



The suspension of safeguarding can only be enabled by the FB after a move from position 1 to position 2. Other switching directions or positions may not be used to enable the suspension of safeguarding. This measure meets the requirements of EN 60204 Section 9.2.5.8.
In order to meet the requirements of EN 60204 Section 9.2.4, the user shall use a suitable switching device. In addition, the user must ensure that the relevant operating mode (EN 60204 Section 9.2.3) is selected in the application (automatic operation must be disabled in this operating mode using appropriate measures).

The operating mode is usually specified using an operating mode selection switch in conjunction with the SF_ModeSelector FB and the SF_SafeRequest or SF_SafelyLimitedSpeed FB.

The SF_EnableSwitch FB processes the confirmation of the "safe mode" state via the "S_SafetyActive" parameter. On implementation in an application of the safe mode without confirmation, a static TRUE signal is connected to the "S_SafetyActive" parameter.

State Diagram

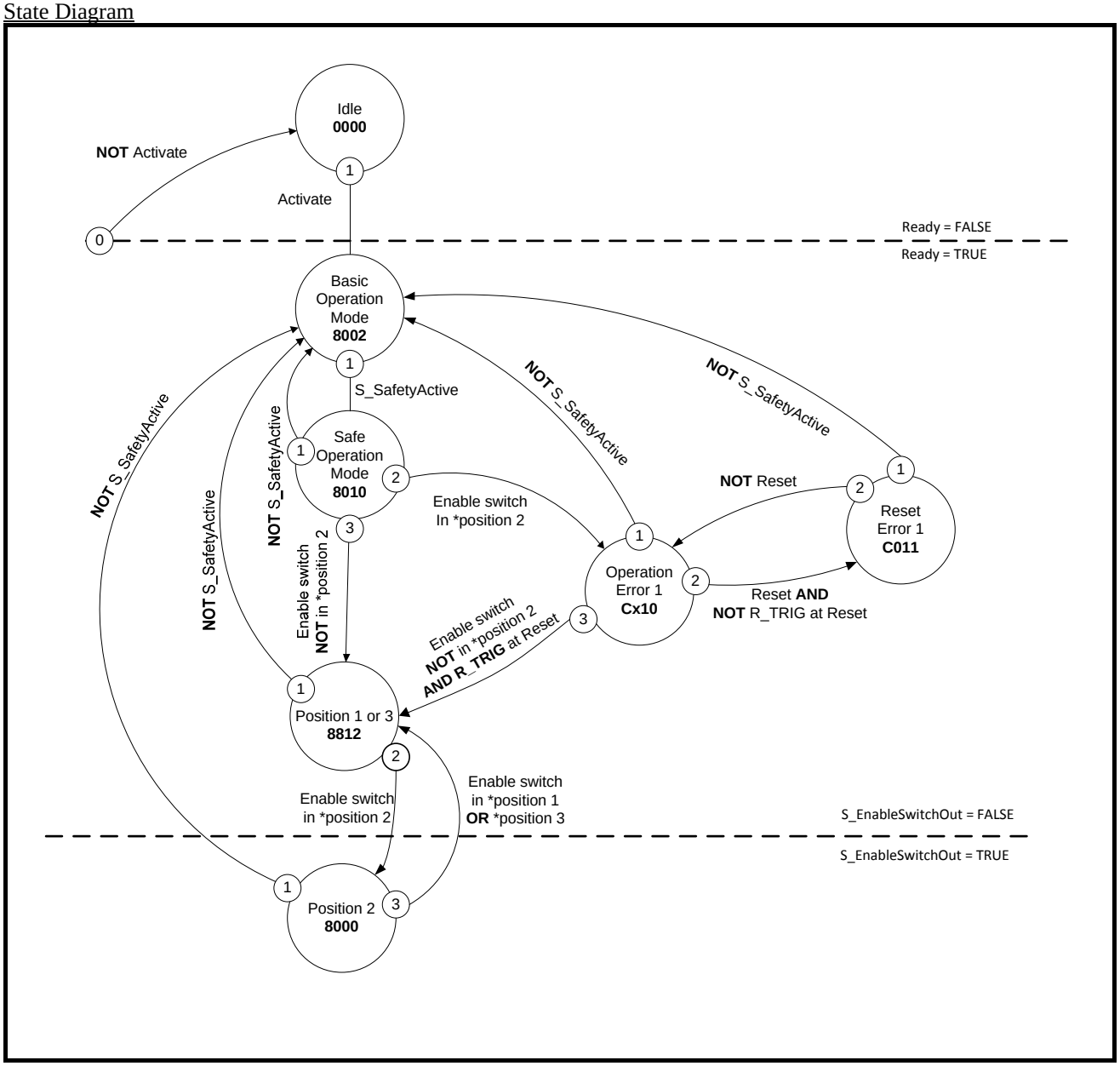


Figure 15: State diagram for SF_EnableSwitch_2

Note: The transition from any state to the Idle state due to Activate = FALSE is not shown. However these transitions have the highest priority (0).

Typical Timing Diagrams

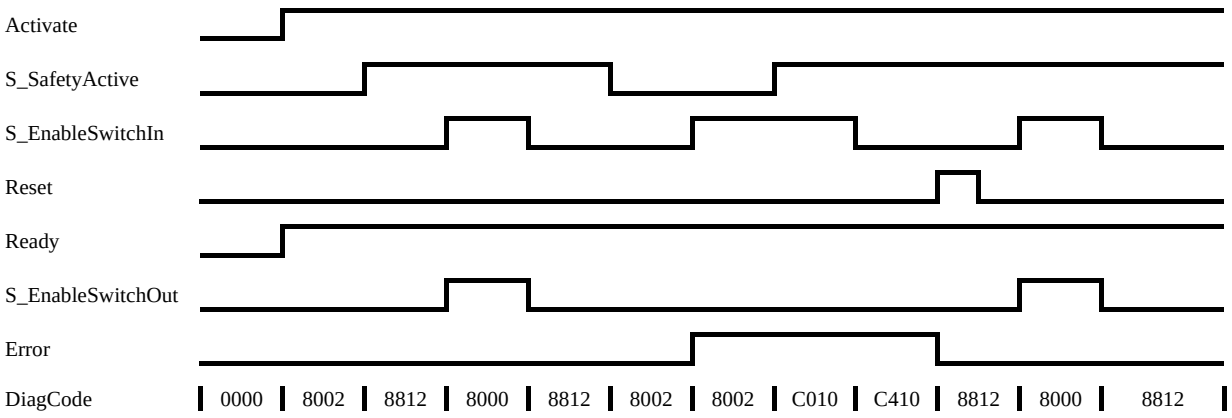


Figure 16: Timing diagram for SF_EnableSwitch_2

2.6.4. Error Detection

It will be detected if the enable Switch is already pressed when Safety becomes active. The machine must be put in a safe state first before the enable switch can be used.
In case Reset is requested, a permanent Reset signal TRUE will be detected (Reset error).

2.6.5. Error Behavior

In the event of an error, the S_EnableSwitchOut safe output is set to FALSE and remains in this Safe state. Once the S_EnableSwitchIn becomes FALSE, via releasing the enable switch by the operator, the error can be reset via the Reset input. If during the error condition TRUE, S_SafetyActive becomes FALSE, there is no need for a separate Reset. However, if the EnableSwitch is not released before S_SafetyActive becomes TRUE again, a transition to the error state C010 is made.

2.6.6. Function Block-Specific Error and Status Codes

FB-specific error codes:

DiagCode	State Name	State Description and Output Setting
C001	Reset Error 1	Static Reset signal detected in state Cx10. Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
Cx10	Operation Error 1	Enable switch not in position 1 during activation of S_SafetyActive. IF S_EnableIn = TRUE x = 0 ELSE x = 4 Output signals for x = 0 (C010) Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE Output signals for x = 4 (C410) Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific status codes (no error):

DiagCode	State Name	State Description and Output Setting
0000	Idle	The function block is not active (initial state). Ready = FALSE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8002	Basic Operation Mode	Safe operation mode is not active. Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8010	Safe Operation Mode	Safe operation mode is active. Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8812	Position 1 or 3	Safe operation mode is active and the enable switch is in position 1 or 3. Ready = TRUE S_EnableSwitchOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8000	Position 2	Safe operation mode is active and the enable switch is in position 2. Ready = TRUE S_EnableSwitchOut = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

Appendix 1. Compliance Procedure and Compliance List

Listed in this Appendix are the requirements for the compliance statement from the supplier of the safety specification. Be aware that this part cannot be seen as separate part for this part 3 the compliance statement of Part 1 should also be included. The compliance statement consists of two main groups:

1. Reduction in programming languages and functionality (see "Appendix 1.2 Reduction in the Development Environment").
2. The definition of a set of function blocks with safety-related functionality (see "Appendix 1.3 Overview of the Function Blocks").

The supplier must fill out the tables for their implementation, according to their product, committing their support to the specification itself.

By submitting these tables to PLCopen, and following approval by PLCopen, the list will be published on the PLCopen website (http://www.PLCopen.org) as specified in "Appendix 2 The PLCopen Safety Logo and Its Use" below.

In addition to this approval, the supplier is provided with access and usage rights for the PLCopen Safety logo, as described in Appendix 2 - The PLCopen Safety Logo and Its Use.

Appendix 1.1. Supplier Statement

Supplier name	
Supplier address	
City	
Country	
Phone	
Fax	
Website	
Product name	
Product version	
Release date	
Certified by	

I hereby state that the following tables as filled out and submitted correspond to our product and the accompanying user manual, as stated above.

Name of representative:

Date of signature (dd/mm/yyyy):

Signature:

Appendix 1.2. Overview of the supported Function Blocks

Function Blocks	Supported	Comments (<= 48 Characters)
SF_GuardLocking_2		
SF_GuardLockingSerial		
SF_PSE		
DIAG_SF_xxxxx		
SF_Override		
SF_EnableSwitch_2		

Table 2: Overview of the function blocks

Appendix 2. The PLCopen® Safety Logo and Its Use

For quick identification of compliant products, PLCopen has developed a logo for the Safety Specification:

Figure 17: The PLCopen® Safety logo

This logo is owned and trademarked by PLCopen®.

In order to use this logo free of charge, the relevant company must meet all of the following requirements:

1. The company must be a voting member of PLCopen;
2. The company must comply with the existing specification, as specified by the PLCopen Technical Committee 5 - Safety, and as published by PLCopen, and of which this statement is a part;
3. This compliance is submitted in writing by the company to PLCopen, clearly stating the applicable software package and the supporting elements of all the specified tables, as specified in this document;
4. The company is aware that this compliance is only a statement of the supporting elements as specified in this document. In particular, the company is aware that this statement does not have any relationship to the implementation itself, nor the fulfillment of any requirements as specified in any safety standard, safety procedure, or development procedure, and does not state anything with regard to the quality of the product itself, nor certification procedures performed by a third party;
5. In the event of non-fulfillment, which must be decided by PLCopen, the company will receive a written statement to this effect from PLCopen. The company will have a period of one month to either adapt their software package in such a way that it is compliant, i.e., by issuing a new compliance statement, or removal of all reference to the specification, including the use of the logo, from all their specifications, be they technical or promotional material;
6. The logo must be used as is - i.e., in its entirety. It may only be altered in size as long as the original scale and color settings are maintained;
7. The logo must be used in the context of PLCopen Safety.