PLCopen - Technical Committee 5

–

Safety Software

Technical Specification

Part 4: Application Specific FBs for Presses

PLCopen Document, Version 1.0, Official Release

DISCLAIMER OF WARANTIES

Date: December 16, 2013.

Total number of pages: 136

Function blocks for Safety Functionality

The following paper is a document under construction within the PLCopen Task Force Safety.
It summarizes the results of the PLCopen Task Force Safety meetings, containing contributions of all its members.
The present specification was written thanks to the following members of the Task Force:

Thomas Schulz	BGHM as tester
Erich Janoschek	TUV
Stefan Haeb	TÜV Rheinland
Joachim Greis	Beckhoff
Andres Hager	B&R
Jochen Ost	Bosch Rexroth
Ralf Hänisch	Fraunhofer First
Wolfgang Zeller	Hochschule Augsburg
Harry Koop	KW Software
Ralf Braeuning	Omron
Frank Bauder	Omron
Carsten Gregorius	Phoenix Contact
Olaf Ruth	Phoenix Contact
Eelco van der Wal	PLCopen, Gorinchem, The Netherlands

Change Status List:

Version number	Date	Change comment
V 0.1	April 14,2009	As result of the meeting March 23 and 24, 2009, at Hima
V 0.2	May 20, 2009	As result of the meeting May 19 and 20, 2009 in Düsseldorf. Renamed to Part 5 versus Part 4.
V 0.3	January 8, 2010	As result of the meeting Oct. 15 & 16, 2009 renamed to Part 4.
V 0.4	March 24, 2010	As result of the meeting March 23 and 24 in Verl
V 0.5	July 1, 2010	As result of the meeting in Frankfurt a. M.
V 0.6	Aug 2010	As a result of the meeting in Lemgo
V 0.7	Jan. 21, 2011	As a result of the meeting in Nufringen
V 0.8	March 25, 2011	As a result of the webmeeting
V 0.9	April 2011	As result of the webmeeting
V 0.91	February 1, 2012	As result of the meeting in Lohr am Main and in Bad Pyrmont in January. Includes homework as decided during meeting Oct. 2011 in Lemgo
V 0.92	February 17, 2012	As result of the meeting in Augsburg
V 0.93	March 27	As result of the meeting in Berlin
V 0.94	May 9, 2012	All feedback merged in one document as preparation for webmeeting
V 0.95	May 14, 2012	As result of the webmeeting
V 0.96	May 25, 2012	As a result of the meeting in Bad Pyrmont
V 0.97	July 11, 2012	As a result of the meeting in the vicinity of Amsterdam
V 0.98	August 31, 2012	As a result of the homework supplied
V 0.98A	September 20, 2012	As a result of the webmeeting. Basis for identified homework.
V 0.98B	October 1, 2012	As a result of the homework supplied
V 0.98C	October 9, 2012	As a result of the webmeeting
V 0.98D	October 22, 2012	As a result of the webmeeting
V 0.98E	December 6, 2012	As a result of the webmeeting Nov. 22, minor editorial changes and incl. of translation of CycleControl
V 0.99	December 14, 2012	Published as Release for Comments after webmeeting Dec. 13
V 0.99A	October 24, 2013	First version with feedback included
V 0.99B	December 16, 2013	As result of the webmeeting Nov. 19 and Dec. 9. Version for approval BG. Last version before release as 1.0
V 1.0	December 16, 2013	Version 1.0 created. All changes accepted. Last links corrected.

Table of Contents

Table of Figures

Table of Tables

1. General

This document is an addition to the existing safety standards as provided by PLCopen, and should not be seen as a specification on itself but as a part of the set of specifications.
This document adds the safety requirements for presses.

1.1. Objectives

Allow the user to achieve the functional safety at the plant and machine level especially for presses.

1.1.1. Definition of the goals

- Standard FBs for standard Functionality
- Providing user guidelines / examples

1.1.2. Language context goals

The Safety Function Blocks shall be applicable in the IEC 61131-3 languages with following factors in consideration:

1.1.3. Definition of a set of Function Blocks

1.1.4. References to other Standards

It should be known that this specification is part of a suite of safe software specifications as developed within the Technical Committee 5 of PLCopen, starting with:

PLCopen Technical Committee 5 -Safety Software - Technical Specification: Part 1: Concepts and Function Blocks, Version 1.0 – Official Release of January 31, 2006

The other parts currently consist of:

- Part 2 – User Guidelines
- Part 3 – Extensions to Part 1

2. General Introduction for Machine Presses

2.1. General introduction

Wikipedia.org mentions the following about presses:
A press or a machine press is a tool used to work metal (typically steel) by changing its shape and internal structure.
A forge press reforms the work piece into a three dimensional object, not only changing its visible shape but also the internal structure of the material. A stronger part results from the process in comparison to a machined object.
Bending is a typical operation performed and occurs by a machine pressing, or applying direct pressure, to the material and forcing it to change shape. A press brake is a typical machine for this operation.
An easy to understand type of machine press is a set of rollers. Metal is fed into the rollers, which are turning to pull the material through. The space between the rollers is smaller than the unfinished metal, and thus the metal is made thinner and/or wider.
Another kind of press is a set of plates with a relief, or depth-based design, in them. The metal is placed between the plates, and the plates are pressed up against each other, deforming the metal in the desired fashion. This may be coining or embossing or forming. A punch press is used for forming holes. Capping Presses form caps from rolls of aluminium foil at up to 660 per minute.
Progressive stamping is a manufacturing method that can encompass punching, coining, bending and several ways of modifying the metal, combined with an automatic feeding system. The feeding system pushes a coil of metal through all of the stations of a progressive stamping die. Each station performs one or more operations until a finished part is made per the requirements on the print. The final operation is a cut-off operation, which separates the finished part from the carrying web. The carrying web, along with metal that is punched away in previous operations, is considered scrap metal.

Figure 1: Power press with a fixed barrier guard

A brake press is a special type of machine press that bends sheet metal into shape.

Figure 2: Brake Press

A good example of the type of work a brake press can do is the backplate of a computer case. Other examples include brackets, frame pieces and electronic enclosures just to name a few. Some press brakes have CNC controls and can form parts with accuracy to a fraction of a millimetre. These machines can be dangerous considering the knife-edge bending dies and forces in excess of 4,000 kilonewtons (900,000 lbf) However in the hands of a skilled operator the machine presents minimum hazard.
Machine presses are used extensively around the world for shaping all kinds of metals to a desired shape. A typical toaster (for bread) has a metal case that has been bent and pressed into shape by a machine press.
Machine presses can be hazardous, so safety measures must be taken. Injuries in a press may be permanent, because of the large forces used. Bimanual controls (controls the use of which requires both hands to be on the buttons to operate) are a very good way to prevent accidents, as are light sensors that keep the machine from working if the operator is in range of the die.

2.2. Application example of a mechanical press

To show the combination of the different functionalities, an example for a power press is described hereunder.
The press in the center is seen from the top.
The operator sides are on the top and bottom of the picture. They are protected by both two hand controls (S11-S18) and/or a light curtain (S19 and S20), one on the front side and back side. The two hand control devices are selectable.
Access from the left and the right side of the press are protected by interlocked guards (S21, S22)
On every corner of the press there is an emergency stop button installed.
The operator panel is located on a central position. It contains a mode-selector, and additional emergency stop functionality, the pre-selection for the 4 two-hand-control devices, and a switch for backward move. It also contains a reset button and two indicators (lamps) for status information.

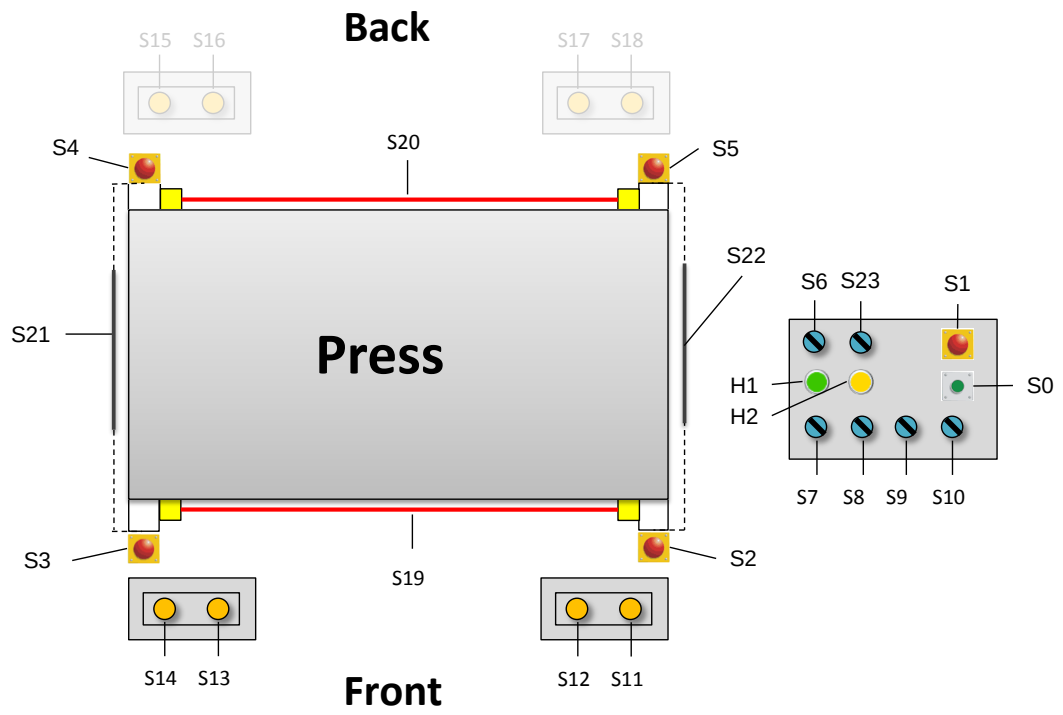


Figure 3: Basic safety outline of a press

The following inputs are identified:

S0	Reset Button	S13	TwoHandControl 2 Button 1
S1	Emergency Stop Button	S14	TwoHandControl 2 Button 2
S2	Emergency Stop Button	S15	TwoHandControl 3 Button 1 (optional)
S3	Emergency Stop Button	S16	TwoHandControl 3 Button 2 (optional)
S4	Emergency Stop Button	S17	TwoHandControl 4 Button 1 (optional)
S5	Emergency Stop Button	S18	TwoHandControl 4 Button 2 (optional)
S6	Mode Switch	S19	Light curtain (Front)

S7	Disable Switch operator console 1 (Front)	S20	Light curtain (Back)
S8	Disable Switch operator console 2 (Front)	S21	Doorinterlock
S9	Disable Switch operator console 3 (Back)	S22	Doorinterlock
S10	Disable Switch operator console 4 (Back)	S23	Backward Move Switch
S11	TwoHandControl 1 Button 1	H1	start necessary
S12	TwoHandControl 1 Button 2	H2	insert material

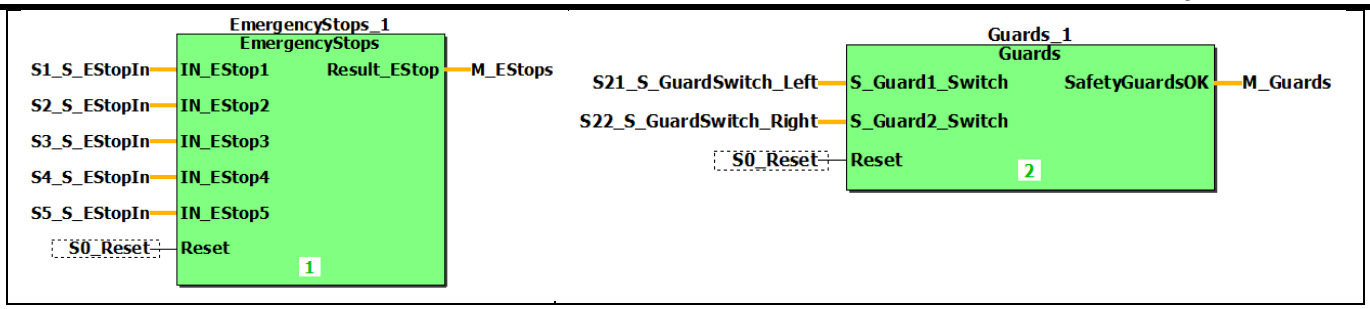
One can identify the following safety functions and reactions:

SF	Description	Priority	PLr	Modes
1…5	**Emergency push button S1…5 activated → switch off valves 1…4**	**1**	**e**	**all**
6…7	**Guard 1 or 2 opened → switch off valves 1…4**	**1**	**e**	**all**
8	**No valid two hand control configuration front → switch off valves 1…4**	**1**	**e**	**all**
9	**No valid two hand control configuration back→ switch off valves 1…4**	**1**	**e**	**all**
10	**If camshaft monitoring is not ok →switch off valves 1…4**	**1**	**e**	**all**
11	**If cam monitoring is not ok →switch off valves 1…4**	**1**	**e**	**all**
12	**If the lightcurtain backside is interrupted →switch off valves 1…4**	**2**	**e**	**setup, manual THC front, manual THC cyclic, automatic**
13	**If the lightcurtain frontside is interrupted →switch off valves 1…4**	**2**	**e**	**automatic**
14	**If one (two) front two hand control(s) is (are) activated → press cycle is enabled** **Simplified! Detailed description see TC5, part 4, pp. 29**	**3**	**e**	**setup, manual THC front, manual THC cyclic, automatic**
15	**If one (two) front two hand control(s) is (are) activated and one (two) back two hand controls is (are) activated → press cycle is enabled** **Simplified! Detailed description see TC5, part 4, pp. 29**	**3**	**e**	**manual THC front/back**

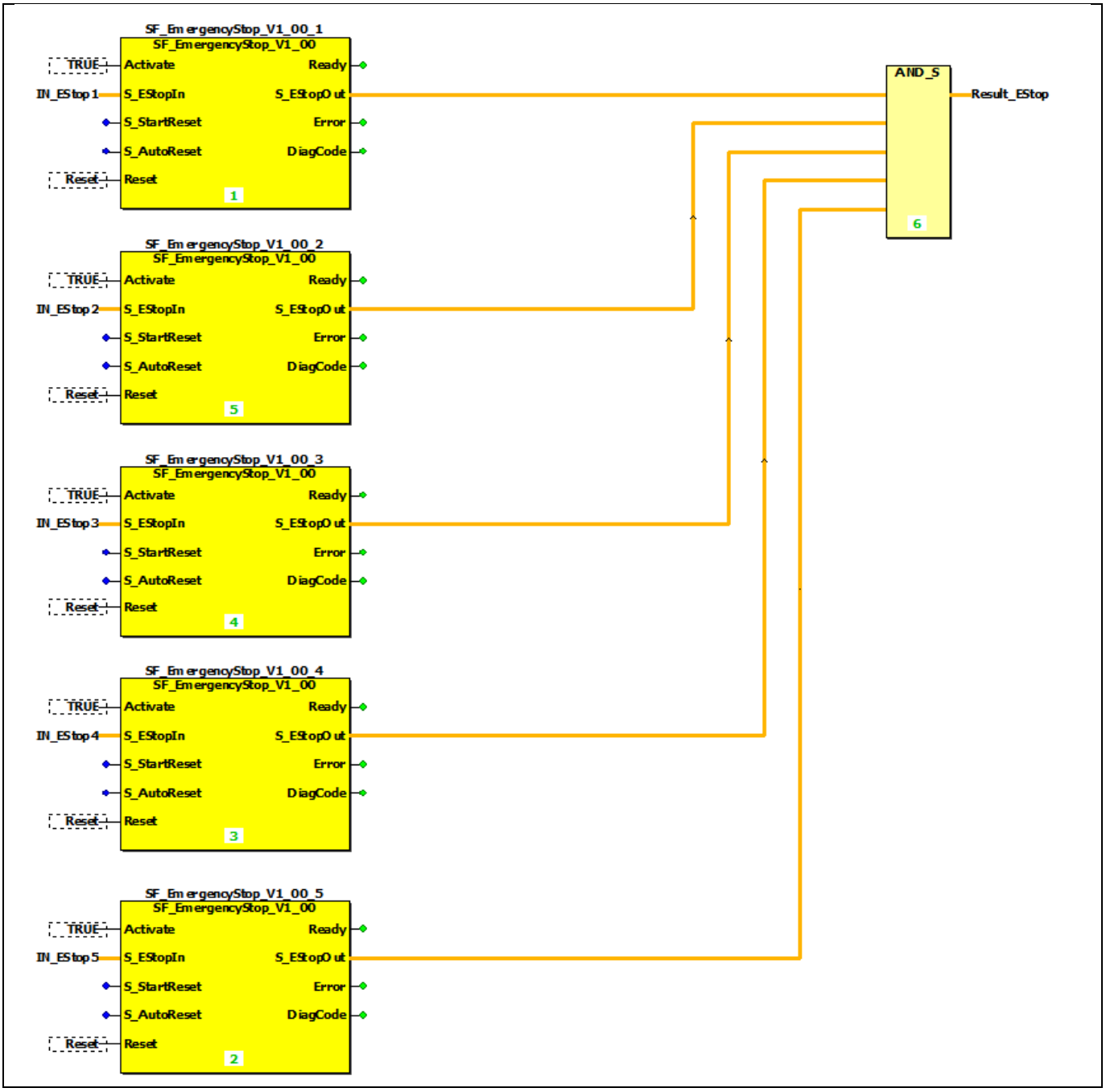
In principle the software application contains of 3 safety loops:

1. Basic safety loop coupled to the emergency stops, the two guards (door interlocks) on the left and right and the CAM-ShaftMonitoring
2. Safety loop at the front consisting of light curtain and/or two-hand control device
3. Safety loop at the back consisting of light curtain and/or two-hand control devices (optional)

Ad 1: Basic Safety Loop
This program part consists of 2 sections, one for the emergency stops and one for the guards.



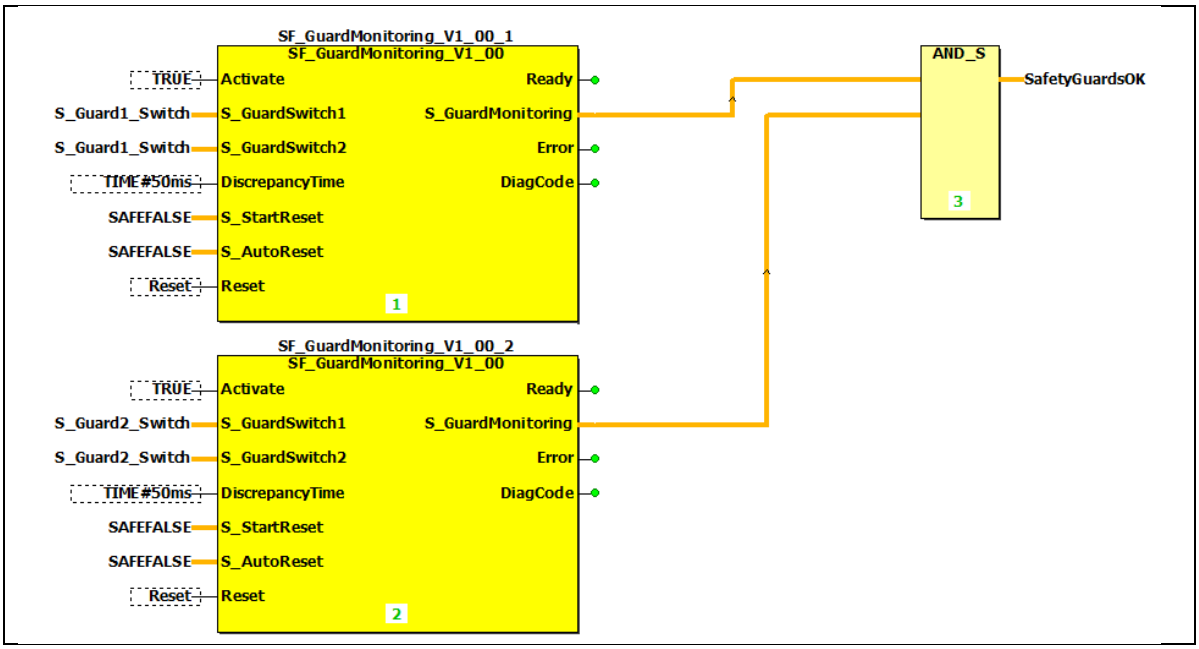
The user derived FB for the emergency combines all 5applicable switches in one output Result_Estop coupled to one variable (marker or connector) M_Estops:



Short description of the used function blocks:

FB Name	SF_EmergencyStop
This function block is a safety-related function block for monitoring an emergency stop button. This FB can be used for emergency switch off functionality (stop category 0), or - with additional peripheral support - as emergency stop (stop category 1 or 2)	

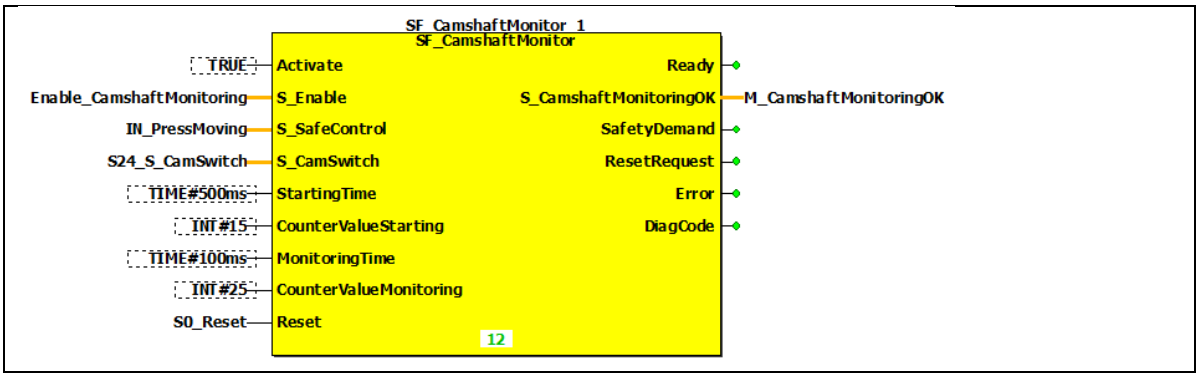
The user derived FB for the guards consist of the combination of two SF_GuardMonitoring FBs resulting in the output SafetyGuardsOK coupled to the variable M_Guards.



Short description of the used function block:

FB Name	SF_GuardMonitoring
This function block monitors the relevant safety guard. There are two independent input parameters for two switches at the safety guard coupled with a time difference (MonitoringTime) for closing the guard.	

The CamshaftMonitor detects if the Cam shaft is broken in case of an indirect driven Cam Switch Unit (“Nockenschaltwerk”).



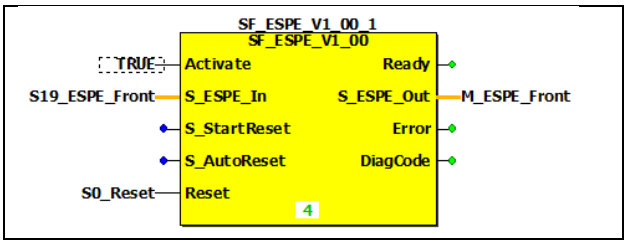
Short description of the used function block:

FB Name	SF_CamshaftMonitor
This function block provides the camshaft monitoring functionality. There must be a number of signal changes in a specified time periode.	

Ad 2: Safety loops at front

This consists of a light curtain and 2 two hand control devices.

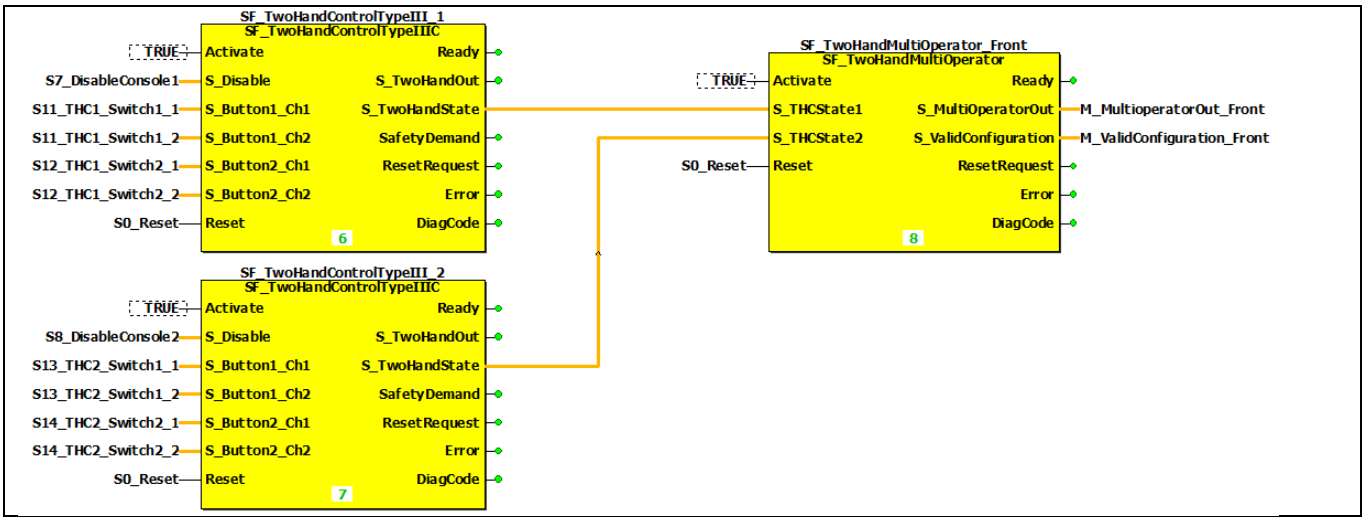
The light curtain results in the variables M_ESPE_Front.



Short description of the used function block:

FB Name	SF_ESPE
This function block is a safety-related function block for monitoring electro-sensitive protective equipment (ESPE).	

Two hand control for the front side, combining the two devices on the front into the variables M_MultiOperatorOut_Front and M_ValidConfiguration_Front showing the validation of the pluggable devices.



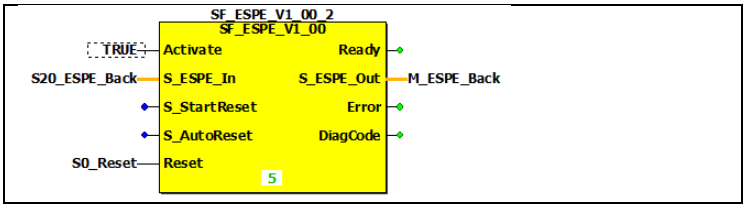
Short description of the used function blocks:

FB Name	SF_TwoHandControlTypeIIIC
This function block provides the pluggable two-hand control functionality (see EN 574, Section 4 Type III. Fixed specified time difference is 500 ms).	

FB Name	SF_TwoHandMultiOperator
This function block is used to control a press with two operator control stations. On each control stations is one two-hand control device.	

Ad 3: Protection for the optional backside

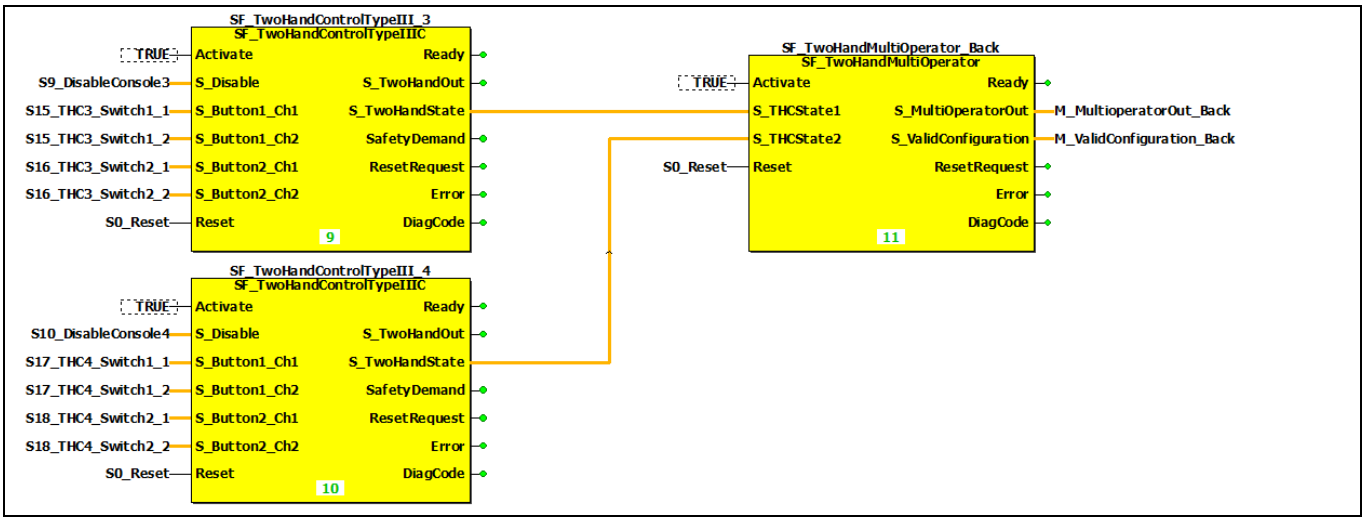
This optional section consists of a light curtain and 2 two hand control devices.
Lightcurtain backside, resulting in the variable M_ESPE_Back



Short description of the used function block:

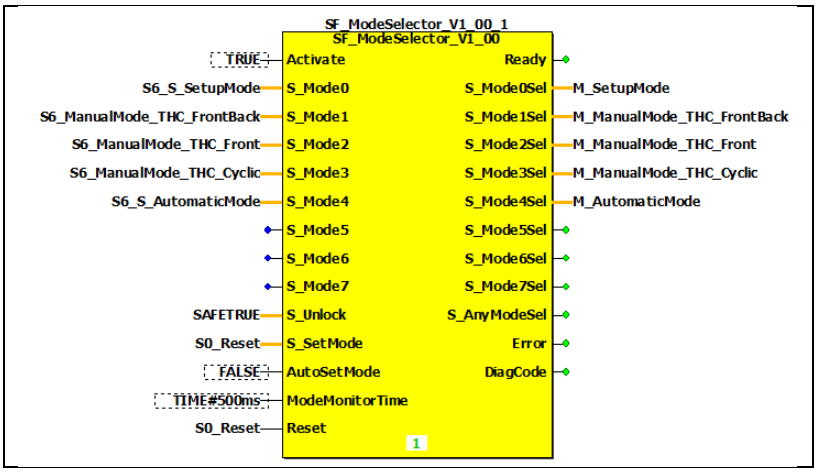
FB Name	SF_ESPE
This function block is a safety-related function block for monitoring electro-sensitive protective equipment (ESPE).	

Same for the back side, resulting in the variables M_MultiOperatorOut_Back and M_ValidConfiguration_Back.



The process side

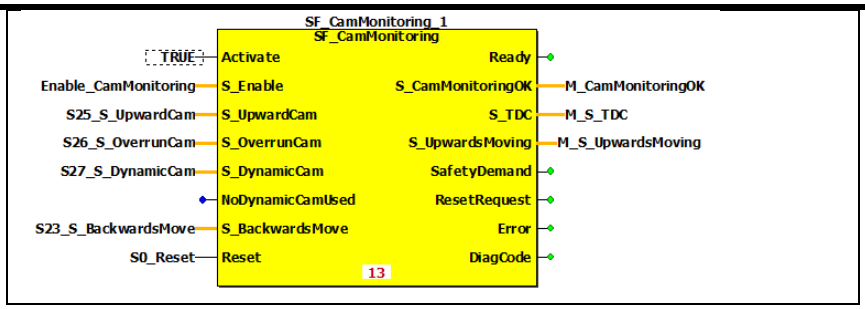
On the process side we start with the ModeSelector with 5 modes: M_SetupMode, M_Manual_THC_FrontBack for strict Two-Hand operation on front and backside, M_ManualMode_THC_Front for the combination of Two Hand Control and ESPE, M_ManualMode_THC_Cyclic for cyclic mode, and M_AutomaticMode:



Short description of the used function block:

FB Name	SF_ModeSelector
This function block selects the system operation mode, such as manual, automatic, semi-automatic, etc.	

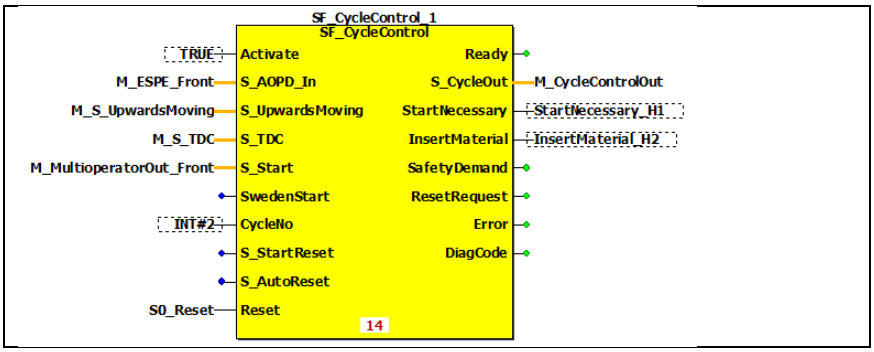
The next step deals with Cam Monitoring. This section does a plausability check between signals of the safety related Cams. In case of a plausability failure the output S_CamMonitoringOK will be FALSE. This FB controls the variables M_CamMonitoringOK, M_S_TDC and M_S_UpwardsMoving.



Short description of the used function block:

FB Name	**SF_CamMonitoring**
This function block provides the cam monitoring functionality. There must be a defined sequence of the cam signals.	

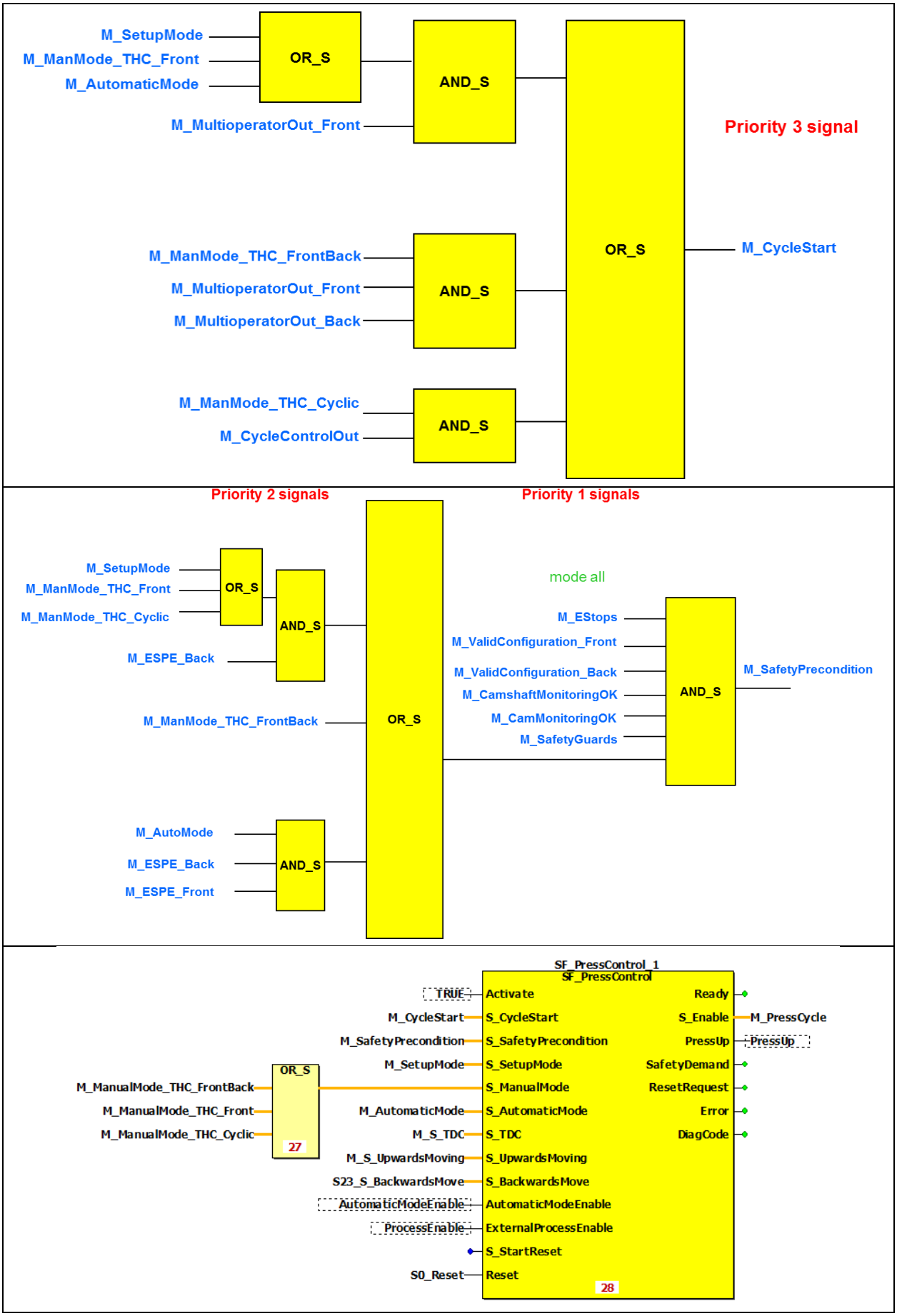
This section deals with the cycle control of the ESPE. It uses the variables from the ESPE and SF_CamMonitoring and controls the values of the variable M_CycleControlOut. With the input CycleNo one can select the number of applicable cycles (fixed value, not dynamically selectable). If the press provides two modes (e.g. single break and double break) the FB has to be called twice.



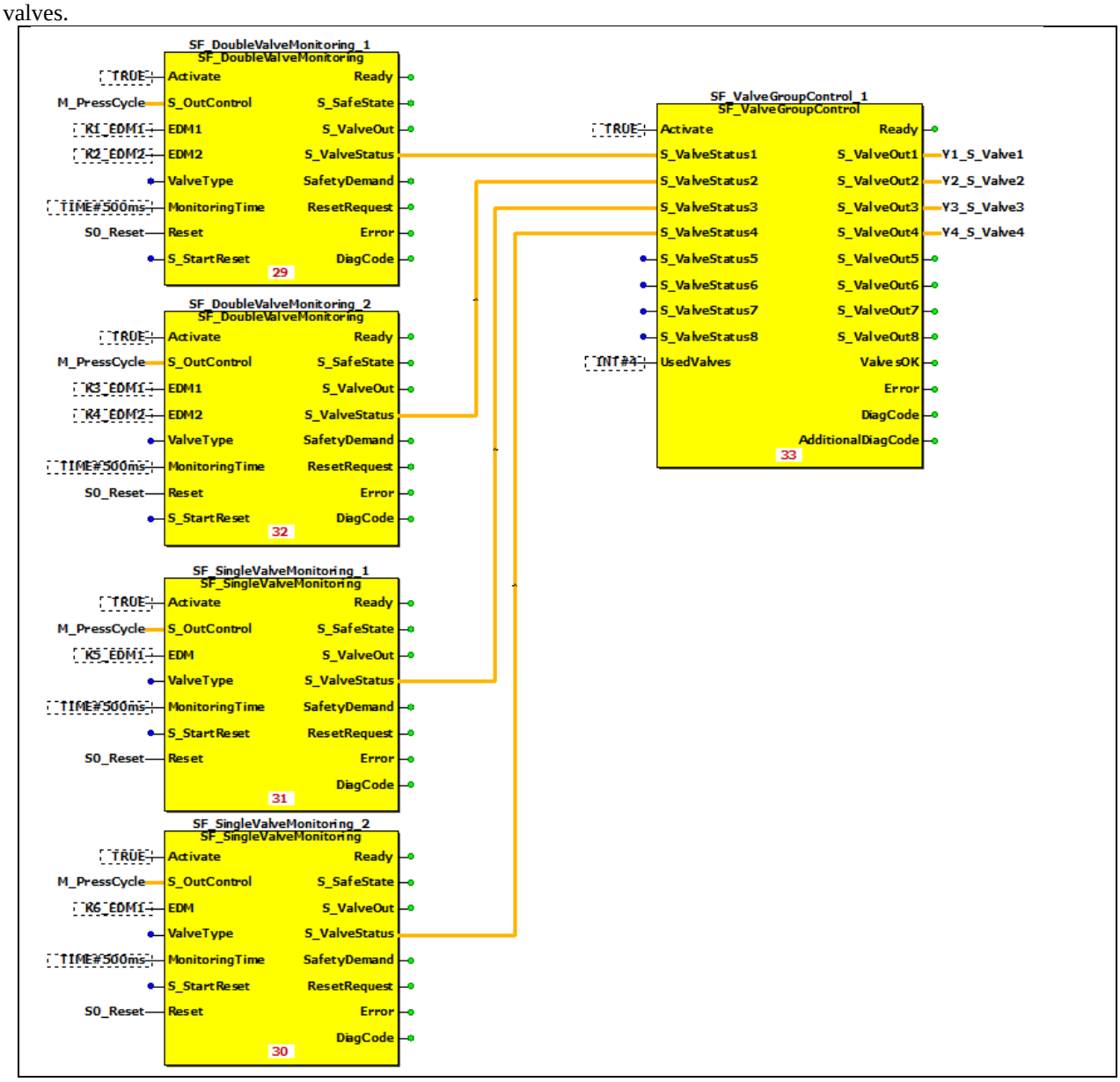
Short description of the used function block:

FB-Name	SF_CycleControl
The SF_CycleControl FB is required for controlling cycle mode (1 to n cycles) of an ESPE.	

This part of the application program controls the safety related process and enables the signal for the safety related valves, depending on the operation mode, the setup mode for the two hand controls and ESPE, combined with the 3 safety loops as defined before. The result is shown in the variable M_PressCycle.


M_SetupMode
M_ManMode_THC_Front
M_AutomaticMode
OR_S
AND_S
M_MultioperatorOut_Front
Priority 3 signal
M_ManMode_THC_FrontBack
M_MultioperatorOut_Front
M_MultioperatorOut_Back
AND_S
OR_S
M_CycleStart
M_ManMode_THC_Cyclic
M_CycleControlOut
AND_S
Priority 2 signals
Priority 1 signals
M_SetupMode
M_ManMode_THC_Front
M_ManMode_THC_Cyclic
OR_S
AND_S
M_ESPE_Back
M_ManMode_THC_FrontBack
OR_S
mode all
M_EStops
M_ValidConfiguration_Front
M_ValidConfiguration_Back
M_CamshaftMonitoringOK
M_CamMonitoringOK
M_SafetyGuards
AND_S
M_SafetyPrecondition
M_AutoMode
M_ESPE_Back
M_ESPE_Front
AND_S
SF_PressControl_1
SF_PressControl
TRUE
Activate
Ready
M_CycleStart
S_CycleStart
S_Enable
M_PressCycle
M_SafetyPrecondition
S_SafetyPrecondition
PressUp
PressUp
M_SetupMode
S_SetupMode
SafetyDemand
OR_S
M_ManualMode_THC_FrontBack
M_ManualMode_THC_Front
M_ManualMode_THC_Cyclic
27
S_ManualMode
ResetRequest
M_AutomaticMode
S_AutomaticMode
Error
M_S_TDC
S_TDC
DiagCode
M_S_UpwardsMoving
S_UpwardsMoving
S23_S_BackwardsMove
S_BackwardsMove
AutomaticModeEnable
AutomaticModeEnable
ProcessEnable
ExternalProcessEnable
S_StartReset
S0_Reset
Reset
28

This section deals with the control of the four valves: two valves for coupling and two valves for the breaking of the mechanical drive system of the excenter press. It uses the result of the SF_PressControl M_PressCycle to energize the safety related valves. In case of a valve failure as monitored by the Valve Monitoring FBs, the SF_ValveGroupControl FB will de-energize all safety related valves.



3. Introduction in the used model for presses

A press is used to convert material. For this it normally has a moving section which has a cyclic movement.
One can look to a cycle of a press as a 360 degrees circular movement during which certain actions will be applicable during a certain angle of the cycle. In the figure below the cycle is shown at the full 360 degrees, and several actions are shown on 3 different paths. For instance, on path 1 one sees in clockwise motion an activity over the first 85 degrees, as well as at 135 and 225 degrees. The converting of the material could be done in the angle between the 2 last points.

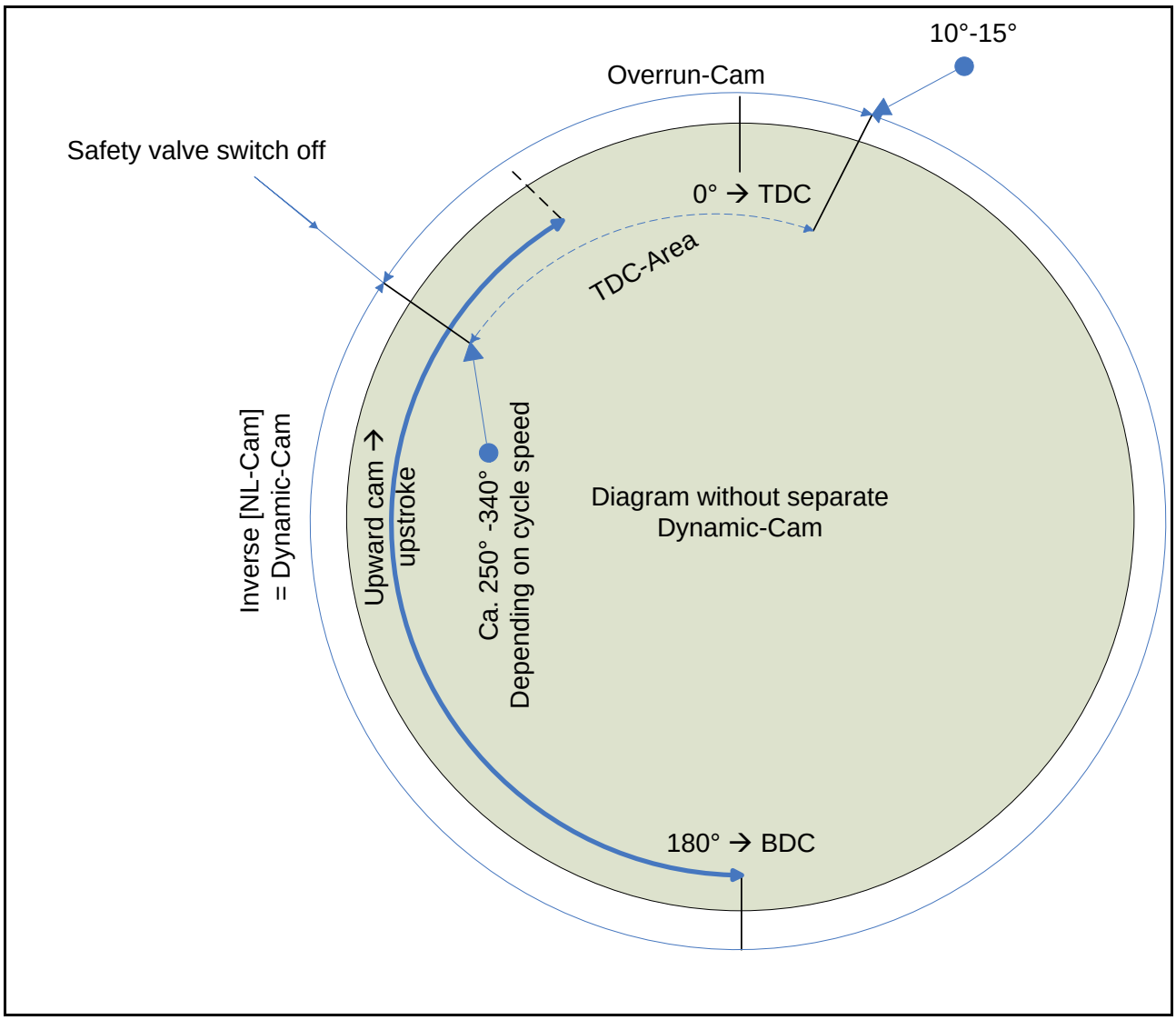


Figure 4: Basic press cycle and naming conventions

Although there are different types of presses, one can identify this cycle in all of them. The way this cycle is initiated consists basically of 3 different modes:

1. Single cycle: the cycle is only done once and stopped at the end. An initiation has to be done before a next cycle is started
2. 1-Cycle, 2-Cycles which are coupled for instance via a light curtain to the insertion and withtraction of the product
3. Continuous mode, where the work is on-going.

3.1. Overview of used Function Blocks

This overview lists most of the function blocks that can be used in relevant safety applications for presses:

Nr	PLCopen SF_FB	Press Type	Safety function	Notes
1	SF_EmergencyStop	Generic	Emergency stop with start up inhibit	Part 1 Ch. 6.4
2	SF_FootSwitch	Generic	Footswitch with enabling	Part 4 Ch. 4.1
3	SF_ESPE	Generic	Light curtain	Part 1 Ch. 6.5
4	SF_GuardLocking	Generic	Guard locking with interlocking	Part 1 Ch. 6.12
5	SF_GuardMonitoring	Generic	GuardMonitoring	Part 1 Ch. 6.2
6	SF_Mode_Selector	Generic	ModeSelect (1 of n) 1 of 8 applicable	Part 1 Ch. 6.3
7	SF_TwoHandControlTypeIII	Generic	Two hand control Type III	Part 1 Ch. 6.11
8	SF_TwoHandControlTypeIIIC	Generic	Pluggable two hand control	Part 4 Ch. 4.12
9	SF_Cycle Control	Mechanical	Cycle control with single or double break.	Part 4 Ch 4.10
10	SF_SingleValveMonitoring	Hydraulic/ Pneumatic	Monitoring of valveswith direct feedback	Part 4 Ch. 4.3
11	SF_SingleValveCycleMonitoring	Hydraulic	Monitoring of Cartridge Valves	Part 4 Ch. 4.4
12	SF_DoubleValveMonitoring	Hydraulic/ Pneumatic	Monitoring of double valves (Press Safety Valves)	Part 4 Ch. 4.5
13	SF_DirectionalValveMonitoring	Hydraulic/ Pneumatic	Control and monitoring of a direction valve	Part 4 Ch. 4.6
14	SF_ValveGroupControl	Generic	Summarizes all the connected valves to a group	Part 4 Ch. 4.7
15	SF_TwoHandMultiOperator	Generic	Enable for multi operator with two-hand control	Part 4 Ch. 4.8
16	SF_CamshaftMonitor	Mechanic & Servo	Camshaft monitoring (Wellenbruch-sicherung)	Part 4 Ch. 9
17	SF_CamMonitoring	Mechanic	Monitoring of a crankshaft (Nocken-schaltwerk)	Part 4 Ch. 11
18	SF_PressControl	Mechanic	Controls the safety related valves depending on mode and protection system	Part 4 Ch. 2

3.2. Diagnostics

For the diagnostics codes the extensions as defined in Part 3 are used. Also the extensions SafetyDemand and ResetRequest as defined in Part 3 are used here.

The Diagnostics Code 83FF is reserved for FBs which can be used in combination with the SF_ValveGroupControl.
The Diagnostics Code 83FE is reserved for FBs which can be used in combination with the SF_TwoHandMultiOperator.

3.3. FB interface

3.3.1. General rules

As referenced in Part 1, Ch. 5.1.1. General input parameters (Copied for convenience only. For references check Part 1)

Input Parameters		
Name	**Type**	**Description**
Activate	BOOL	Variable or constant. Activation of the FB. Initial value is FALSE. This parameter can be connected to the variable, which represents the status (Active or Not Active) of the relevant safety device. This ensures no irrelevant diagnostic information is generated if a device is disabled. If FALSE, all output variables are set to the initial values. If no device is connected, a static TRUE signal must be assigned.
S_StartReset	SAFEBOOL	Variable or constant. FALSE (= initial value): Manual reset when PES is started (warm or cold). TRUE: Automatic reset when PES is started (warm or cold). This function shall only be activated if it is ensured that no hazard can occur at the start of the PES. Therefore the use of the Automatic Circuit Reset feature of the function blocks requires implementation of other system or application measures to ensure that unexpected (or unintended) startup does not occur. It shall be noted in the FB manual that when using a SAFEBOOL variable additional validation of this application is necessary.
S_AutoReset	SAFEBOOL	Variable or constant. FALSE (= initial value): Manual reset when emergency stop button is released. TRUE: Automatic reset when emergency stop button is released. This function shall only be activated if it is ensured that no hazard can occur at the start of the PES. Therefore the use of the Automatic Circuit Reset feature of the function blocks requires implementation of other system or application measures to ensure that unexpected (or unintended) startup does not occur. It shall be noted in the FB manual that when using a SAFEBOOL variable additional validation of this application is necessary.
Reset	BOOL	Variable. Initial value is FALSE. Depending on the function, this input can be used for different purposes: • Reset of the state machine, and coupled error and status messages as indicated via DiagCode, when the error cause has been removed. This reset behavior is designed as an error reset. • Manual reset of a "restart interlock" ("Wiederanlaufsperre" in German) by the operator (see EN ISO 13849-1). This reset behavior is designed as a functional reset. • Additional FB-specific reset functions. This function is only active on a signal change from FALSE to TRUE. A static TRUE signal causes no further actions, but may be detected as an error in some FBs. The appropriate meaning must be described in every FB. It shall be noted in the FB manual that a SAFEBOOL must be connected instead of a BOOL depending on the safety requirements.

As referenced in Part 1, Ch. 5.1.2. General output parameters (Copied for convenience only. For references check Part 1. Also reference to Part 3 now included)

Output Parameter		
Name	**Type**	**Description**
Ready	BOOL	If TRUE, indicates that the FB is activated and the output results are valid (same as the "POWER" LED of a safety relay). If FALSE, the FB is not active and the program is not executed. Useful in debug mode or to activate/deactivate additional FBs, as well as for further processing in the functional program.
Error	BOOL	Error flag (same as "K1/K2" LED of a safety relay). When TRUE, indicates that an error has occurred, and the FB is in an error state. The relevant error state is mirrored at the DiagCode output. If FALSE, there is no error and the FB is in another state. This again is mirrored by DiagCode (this means that DiagCode must be set in the same cycle as the state change). Useful in debug mode as well as for further processing in the functional program.
DiagCode	WORD	Diagnostic register. All states of the FB (Active, Not Active, and Error) are represented by this register. This information is encoded in hexadecimal format in order to represent more then 16 codes. Only one consistent code is represented at the same time. In the event of multiple errors, the DiagCode output indicates the first detected error. For additional information, see chapter 5.2 Diagnostic Codes of Part 1, and chapter 1.2 “Harmonization of diagnostic codes for new function blocks” of Part 3. Useful in debug mode as well as for further processing in the functional program.

Copied from Part 3, Ch. 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.

Other rules:

Missing input parameters	According to IEC 61131-3, if any parameter of a function block input is missing (“open”) then the value from the previous invocation of this instance will be used. In the first invocation the initial value is applied.

Table 1: General Rules

4. Safety Function Blocks Overview

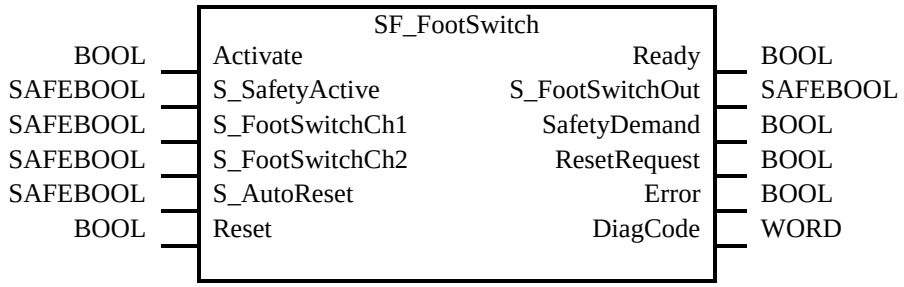
4.1. Foot Switch

4.1.1. Applicable Safety Standards

Standards	Requirements
IEC 60204-1 Ed. 5.1	9.2.6.3 Enabling control 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. 10.6 Start devices Actuators used to initiate a start function or the movement of machine elements (for example slides, spindles, carriers) shall be constructed and mounted so as to minimize inadvertent operation. However, mushroom-type actuators may be used for two-hand control (see also ISO 13851). 10.9 Enabling control device 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 and arranged so as to minimize the possibility of defeating. Enabling control devices shall be selected that have the following features: … – 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 692 2005+A1:2009	5.4.8 Control devices 5.4.8.1 Push button, foot switch and start control devices shall be shrouded to prevent accidental operation. Foot switches shall permit access from one direction only and by one foot only. Treadles shall not be used.
EN 693 2001+A2:2011	5.4.4.3 If a press is also intended to be used according to 5.3.2 a) or b) and at the same time operated, e.g. by foot switch, without any other safeguarding, this mode of production shall be chosen by an additional selector switch operated by a separate key or within a key locked enclosure. The selection of this mode shall automatically give a clear indication at the press that only closed tools or fixed enclosing guards shall be used. 5.4.6.1 Push button, foot switch and start control devices shall be adequately shrouded to prevent accidental operation. Foot switches shall permit access from one direction only and by one foot only. Treadles shall not be used.

4.1.2. Interface Description

FB-Name	SF_FootSwitch		
The FB SF_FootSwitch evaluates the signals of a foot switch with three positions.			
VAR_INPUT			
Input name	*Data type*	*Initial value*	*Description, Parameter values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
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_FootSwitchCh1	SAFEBOOL	FALSE	Variable. Signal of contacts E1 and E2 of the connected foot switch FALSE: Connected switches are open TRUE: Connected switches are closed
S_FootSwitchCh2	SAFEBOOL	FALSE	Variable. Signal of contacts E3 and E4 of the connected foot switch FALSE: Connected switches are open TRUE: Connected switches are closed
S_AutoReset	SAFEBOOL	FALSE	See Section 3.3.1General rules
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_FootSwitchOut	SAFEBOOL	FALSE	Safety related output: Indicates suspension of guard FALSE: Disable suspension of safeguarding TRUE: Enable suspension of safeguarding
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: A short circuit in the input signals is not detected by this FB.			



4.1.3. Functional Description

The SF_FootSwitch FB supports the suspension of safeguarding (DIN EN 60204 Section 9.2.4) using foot 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_FootSwitch FB.
The SF_FootSwitch FB evaluates the signals of a foot switch with three positions (DIN EN 60204 Section 9.2.5.8).
The S_FootSwitchCh1 and S_FootSwitchCh2 input parameters process the following signal levels of contacts E1 to E4:

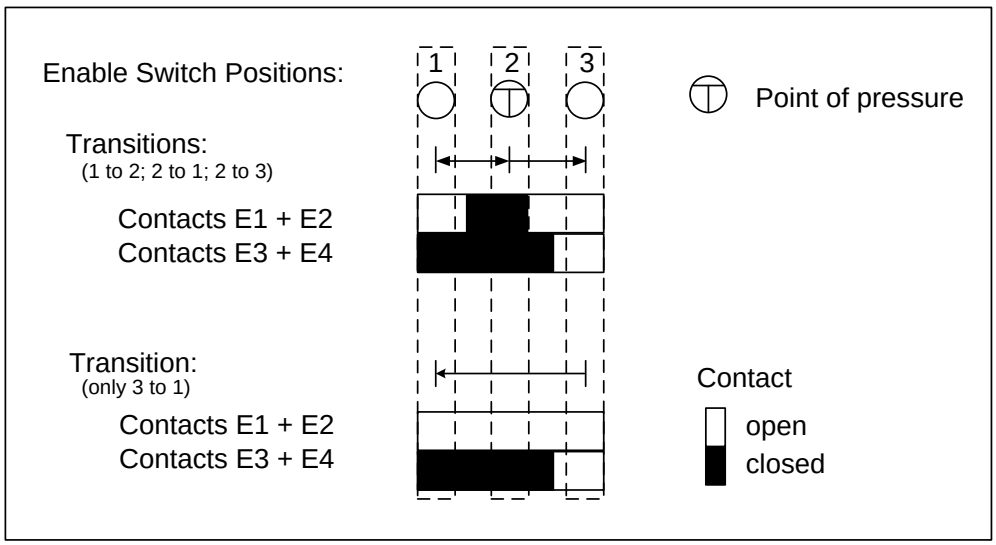


Figure 5: Switch positions

The signal from E1+E2 must be connected to the S_FootSwitchCh1 parameter. The signal from E3+E4 must be connected to the S_FootSwitchCh2 parameter. The position of the foot switch is detected in the FB using this signal sequence.
The transition from position 2 to 3 can be different from shown here.

The switching direction (position 1 => position 2/position 3 => position 2) can be detected in the FB using the defined signal sequence of the foot switch contacts. 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 DIN 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 (DIN 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_FootSwitch 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.

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

State diagram

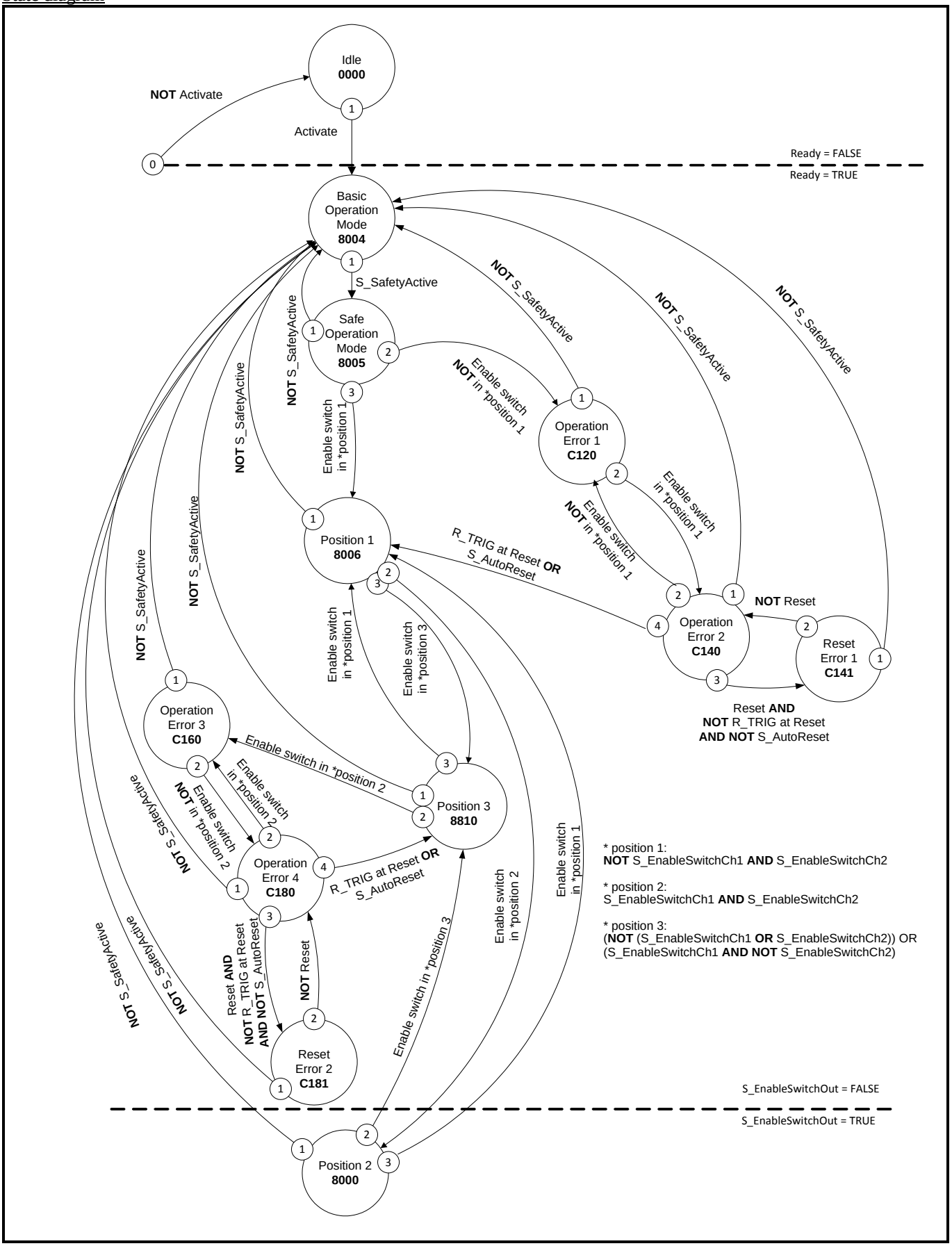
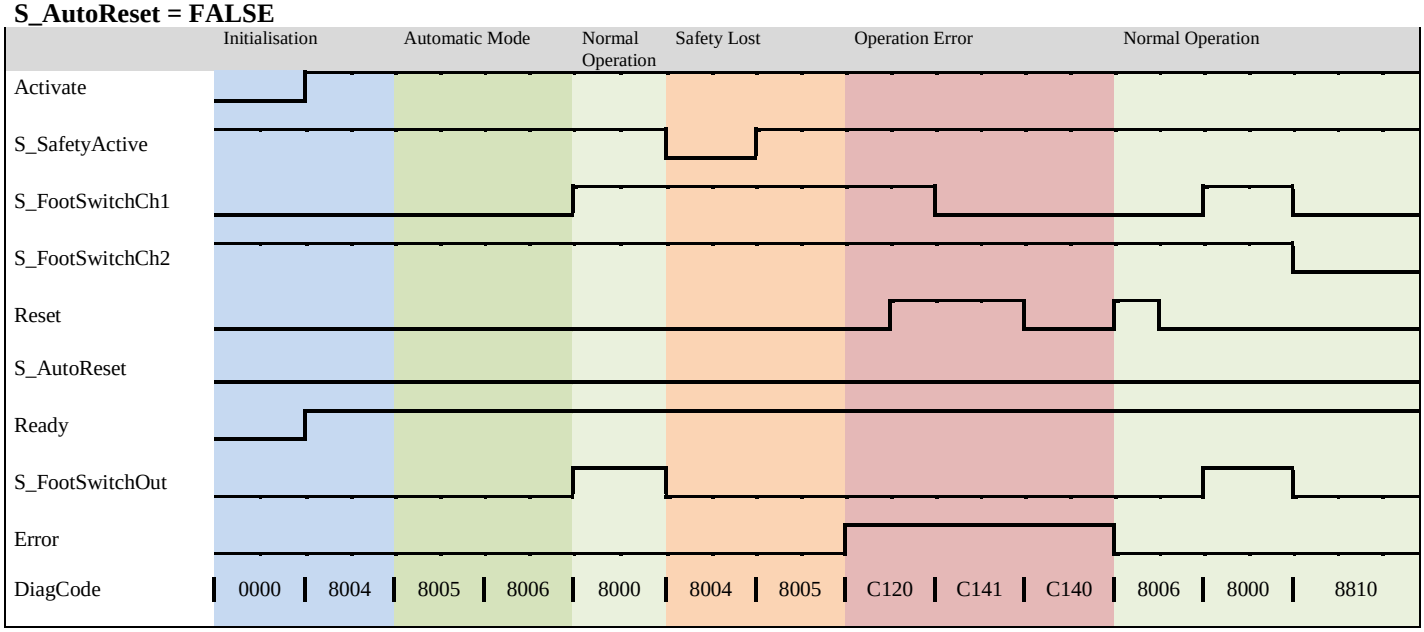


Figure 6: State diagram for SF_FootSwitch

Typcial timing diagrams

S_AutoReset = FALSE



S_AutoReset = TRUE

Figure 7: Timing diagram SF_FootSwitch

4.1.4. Error Detection

Following conditions force a transition from any operational state to ERROR state:

- Invalid static Reset signal in the process.
- Invalid switch positions.

4.1.5. Error Behavior

In the event of an error, the S_FootSwitchOut safe output is set to FALSE and remains in this Safe state.
Different from other FBs, a Reset Error state can be left by the condition Reset = FALSE
Once the error has been removed, the foot switch must be in the initial position specified in the process before the S_FootSwitchOut output can be set to TRUE using the foot switch. If S_AutoReset = FALSE, a rising trigger is required at Reset.

4.1.6. Function Block Specific Error and Status codes

Diagcode	State name	Output setting
FB specific error codes:		
C141	Reset Error 1	Static Reset signal detected in state C140. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE
C181	Reset Error 2	Static Reset signal detected in state C180. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE
C120	Operation Error 1	Foot switch not in position 1 during activation. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE
C140	Operation Error 2	Foot switch in position 1 after C120. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE
C160	Operation Error 3	Foot switch in position 2 after position 3. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE
C180	Operation Error 4	Foot switch not in position 2 after C160. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = TRUE

FB specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = FALSE
8004	Basic Operation Mode	Safe operation mode is not active. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = FALSE
8005	Safe Operation Mode	Safe operation mode is active. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = FALSE Reset Request = FALSE Error = FALSE
8006	Position 1	Safe operation mode is active and the foot switch is in position 1. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = TRUE Reset Request = FALSE Error = FALSE
8810	Position 3	Safe operation mode is active and the foot switch is in position 3. Ready = TRUE S_FootSwitchOut = FALSE Safety Demand = TRUE Reset Request = FALSE Error = FALSE
8000	Position 2	Safe operation mode is active and the foot switch is in position 2. Ready = TRUE S_FootSwitchOut = TRUE Safety Demand = TRUE Reset Request = FALSE Error = FALSE

4.2. PressControl

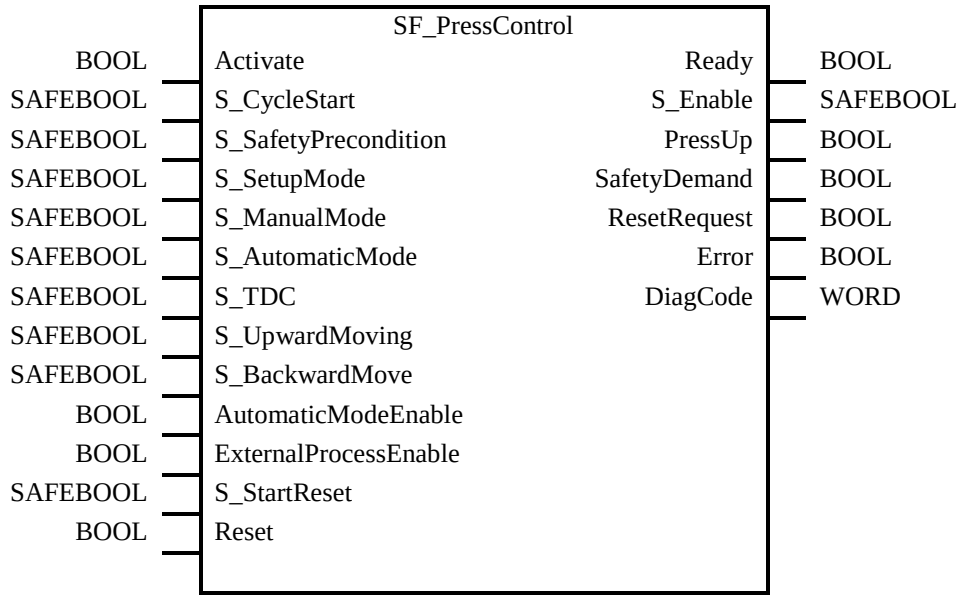
4.2.1. Applicable Safety Standards

Standards	Requirements
EN 692	3.1.6 cycle - automatic operating mode where the slide repeats continuously or intermittently, all functions achieved without manual intervention into the danger zone after initiation 3.1.8 cycle - single operating mode where each operating cycle of the slide has to be positively actuated by the operator 3.1.25 single stroke function feature used to limit the motion of the tool to one operating cycle at each engagement of the clutch even if the stroke initiating means (e.g. a pedal) is held in the operating position 5.3.16 Conditions of interlocking the motor and the clutch Reverse running shall only be possible in the setting mode. It shall not be possible to start the motor if the press clutch is engaged or to engage it or leave it engaged if the motor is stopped, except under various setting conditions. 5.3.17 Single stroke devices Where a press is used in single cycle mode, a single stroke device shall be fitted. This device shall prevent a subsequent stroke even if the control device is continuously actuated. A further stroke shall require the release of the control device and a new initiation. 5.4.1.2 Control systems shall include safety functions designed in such a way that controls have to be reactuated in order for the press to perform a stroke: 5.4.2.4 Where the provision is necessary for overrun monitoring, this shall conform to the following requirements: a) manually fed presses fitted with protective devices of the type listed in 5.4.1.4 shall include overrun monitoring devices to ensure that, if the crankshaft overruns its normal stopping position by an amount specified by the manufacturer, maximum 15° and preferably 10°, a stopping signal shall be immediately initiated and no new cycle initiation shall be possible; b) it shall only be possible to restore further operation of the press by a restricted means, e.g. by tool, key or electronic password;
EN 693	3.4 cycle – single operating mode where each operating cycle of the slide/ram has to be positively actuated by the operator. 3.19 single stroke function feature used to limit the motion of the tool to one operating cycle (single cycle) even if the stroke initiating means (e.g. a pedal) is held in the operating position.

4.2.2. Interface Description

FB Name	**SF_PressControl**		
This FB controls the safety related process and enables the signal for the safety related valves, depending on the operation mode.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_CycleStart	SAFEBOOL	FALSE	Variable This parameter starts the cycle mode depending on the operation mode. A transition from FALSE to TRUE starts the press cycle independent of the operation mode. *Setup mode*: a FALSE leads to a FALSE signal at S_Enable. *Manual mode*: a FALSE signal leads to a FALSE signal at S_Enable as long as the ram is not upward moving (Output S_UpwardMoving of CamMonitor is FALSE). *Automatic Mode:* A transition from FALSE to TRUE will start the cycle. A FALSE signal thereafter will not stop the process. Stopping the process is done via the input AutomaticModeEnable set to FALSE. FALSE: No request to start the Process TRUE: Request to start the process
S_SafetyPrecondition	SAFEBOOL	FALSE	Variable Signal of the preceding safety FBs (e.g. SF_EmergencyStop, SF_GuardMonitoring) FALSE: Safety related pre-conditions are not achieved TRUE: Safety related pre-conditions are achieved
S_SetupMode	SAFEBOOL	FALSE	Variable Selecting the setup mode on the SF_ModeSelector function block. FALSE: Setup mode not selected TRUE: Setup mode selected
S_ManualMode	SAFEBOOL	FALSE	Variable Selecting the manual mode on the SF_ModeSelector function block. FALSE: Manual mode not selected TRUE: Manual mode selected
S_AutomaticMode	SAFEBOOL	FALSE	Variable Selecting the automatic mode on the SF_ModeSelector function block. FALSE: Automatic mode not selected TRUE: Automatic mode selected
S_TDC	SAFEBOOL	FALSE	Variable Signal ofthe preceding safety FB „SF_CamMonitor“. FALSE: Press not in TDC area TRUE: Press in TDC area
S_UpwardMoving	SAFEBOOL	FALSE	Variable Signal ofthe preceding safety FB „SF_CamMonitor“. FALSE: Press not in the upward movement TRUE: Press in the upward movement

S_BackwardMove	SAFEBOOL	FALSE	Variable Allows backward moving of the ram. Input via a separate switch (see application example in Ch. 3) Applicable only in SetupMode. FALSE: No request for backward movement TRUE: Request for backward movement
AutomaticModeEnable	BOOL	FALSE	Variable Control signal from the functional application to enable and switch off the process in automatic mode at the next TDC. FALSE: Automatic mode not enabled or finish the cycle at TDC TRUE: Automatic mode is enabled
ExternalProcessEnable	BOOL	FALSE	Variable or Constant Control signal from the functional application as precondition to enable the process in all modes. Typical not safety critical. Restart is only possible via CycleStart. FALSE: Request to set S_Enable to FALSE TRUE: Request to set S_Enable to TRUE
S_StartReset	SAFEBOOL	FALSE	See Section 3.3.1General rules
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_Enable	SAFEBOOL	FALSE	Safety related output FALSE: Disable connected valves TRUE: Enable connected valves
PressUp	BOOL	FALSE	Output signal for diagnostic FALSE: Press not in the upward movement TRUE: Press in the upward movement
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes:			



4.2.3. Functional Description

This function block provides three operating modes for the press control. It is always used with the function blocks "SF_ModeSelector" and "SF_CamMonitor". The FB implemented in particular the single stroke function in the operating modes "Manual" and "Setup". On the FB "SF_ModeSelector" only one of the three modes must be selected. This is checked to detect systematic programming errors in the safety application program.
The following modes are possible with the module:

Manual Mode:

- In Manual Mode the single stroke function is active.
  - The stroke starts with a 0→1 transition at S_CycleStart when S_TDC = TRUE and S_SafetyPrecondition = TRUE and ExternalProcesEnable is TRUE
  - After the start S_CycleStart must be TRUE until the rising edge of UpwardMoving. Otherwise S_Enable is set to FALSE.
  - If the stroke is started and S_CycleStart is set to FALSE between the TDC area and the BDC it is possible to start again out of the TDC area. An overrun error prevents the start.
  - Upon reaching the TDC the output S_Enable is set to FALSE. The stroke has been fully completed and the process can be restarted.
  - The cyclic overrun control monitors that, after the rising edge of the TDC a renewed switching on is only allowed from the TDC.
  - An overrun error can be reset only in setup mode.

Setup Mode (clockwise):

- Starting the stroke is possible from every position.
- Single stroke function ("Einzelhubsicherung") is active.
  - See "Manual Mode"
  - An overrun error is detected only for diagnostic.
- Muting (BDC signal change from FALSE to TRUE. "Übernahme") is not active (Throughout the cycle S_CycleStart must be TRUE).

Setup Mode (counterclockwise):

- Starting the stroke is possible in every position.
- Single stroke function is not active.
- Automatic stop at rising edge on S_TDC
- Movement is only possible with two hand control device.

Automatic Mode:

- Starting the stroke is possible from every position.
- Single stroke function is not active.
- Muting (BDC signal change from FALSE to TRUE. "Ubernahme") not active.
- The first stroke starts with a rising edge at S_CycleStart (after the rising edge S_CycleStart could be FALSE) if S_SafetyPrecondition = TRUE and AutomaticModeEnable = TRUE.
- Stopped with S_SafetyPrecondition = FALSE or AutomaticModeEnable = FALSE.
- If the automatic mode is stopped with AutomaticModeEnable = FALSE the cycle will be executed until TDC is reached.
- If S_SafetyPrecondition = FALSE the movement is stopped immediately.

In all Modes:

- If S_SafetyPrecondition = FALSE the movement is stopped immediately.
- If S_SafetyPrecondition is set to TRUE after this situation, a new press cycle could be started without a rising edge on the Reset input.

State Diagram

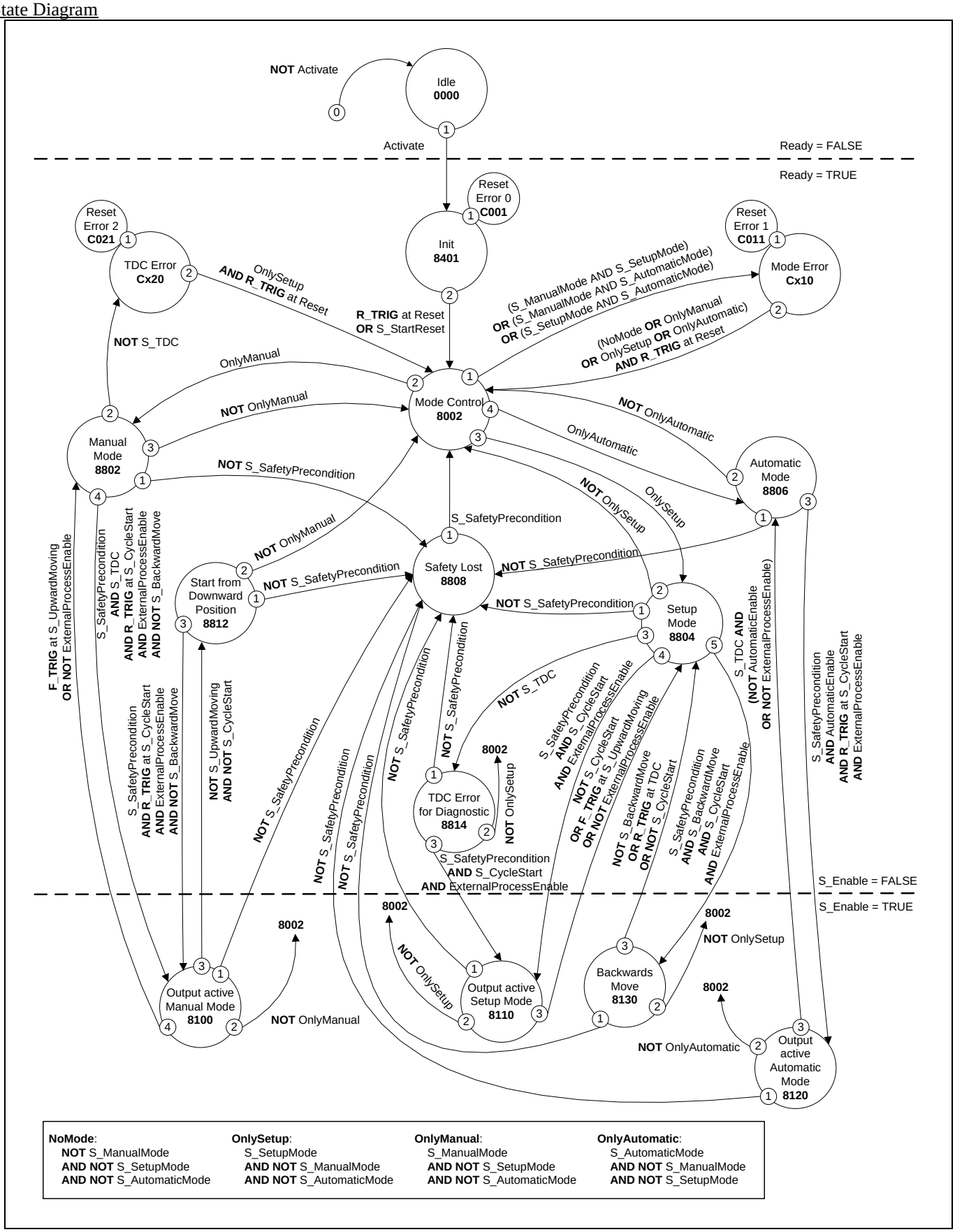


Figure 8: State diagram for SF_PressControl

Typical Timing Diagrams

Setup Mode

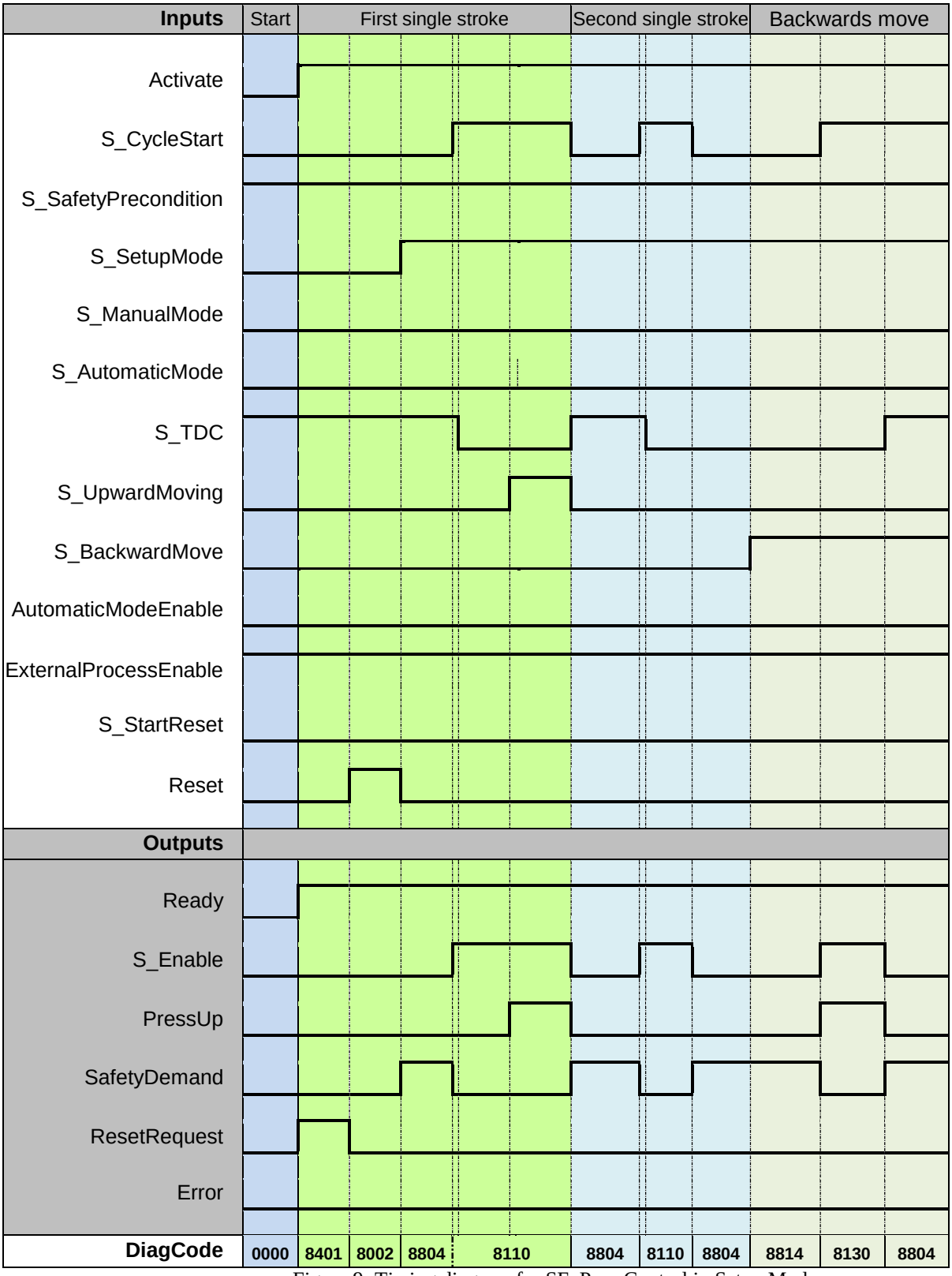


Figure 9: Timing diagram for SF_PressControl in Setup Mode

Manual Mode

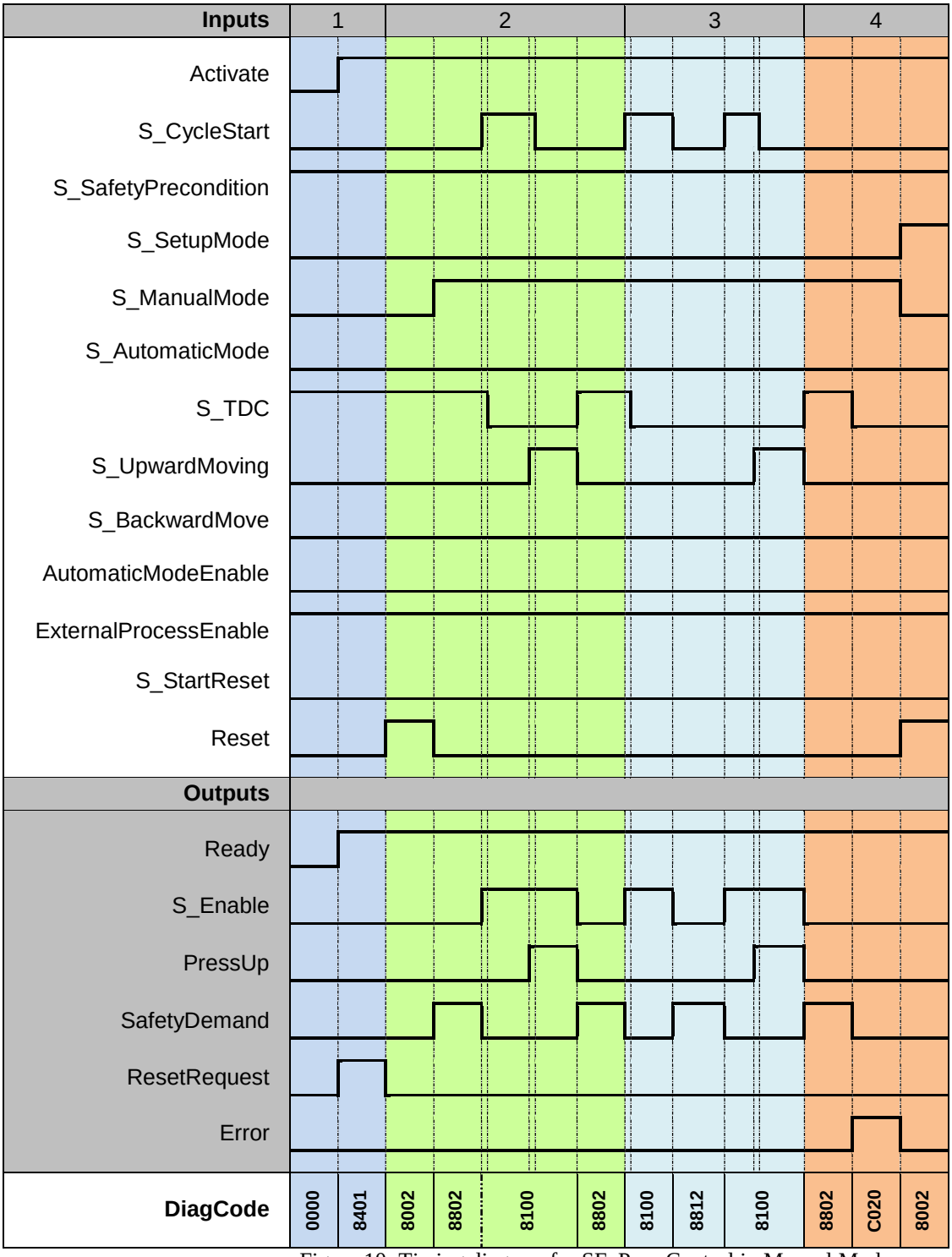


Figure 10: Timing diagram for SF_PressControl in Manual Mode

1: Initialisation and start
2: Single stroke in manual mode
3: Single stroke with stop between TDC and BDC and restart from downward position.
4: Overrun error after the last stroke with reset in setup mode.

Automatic Mode

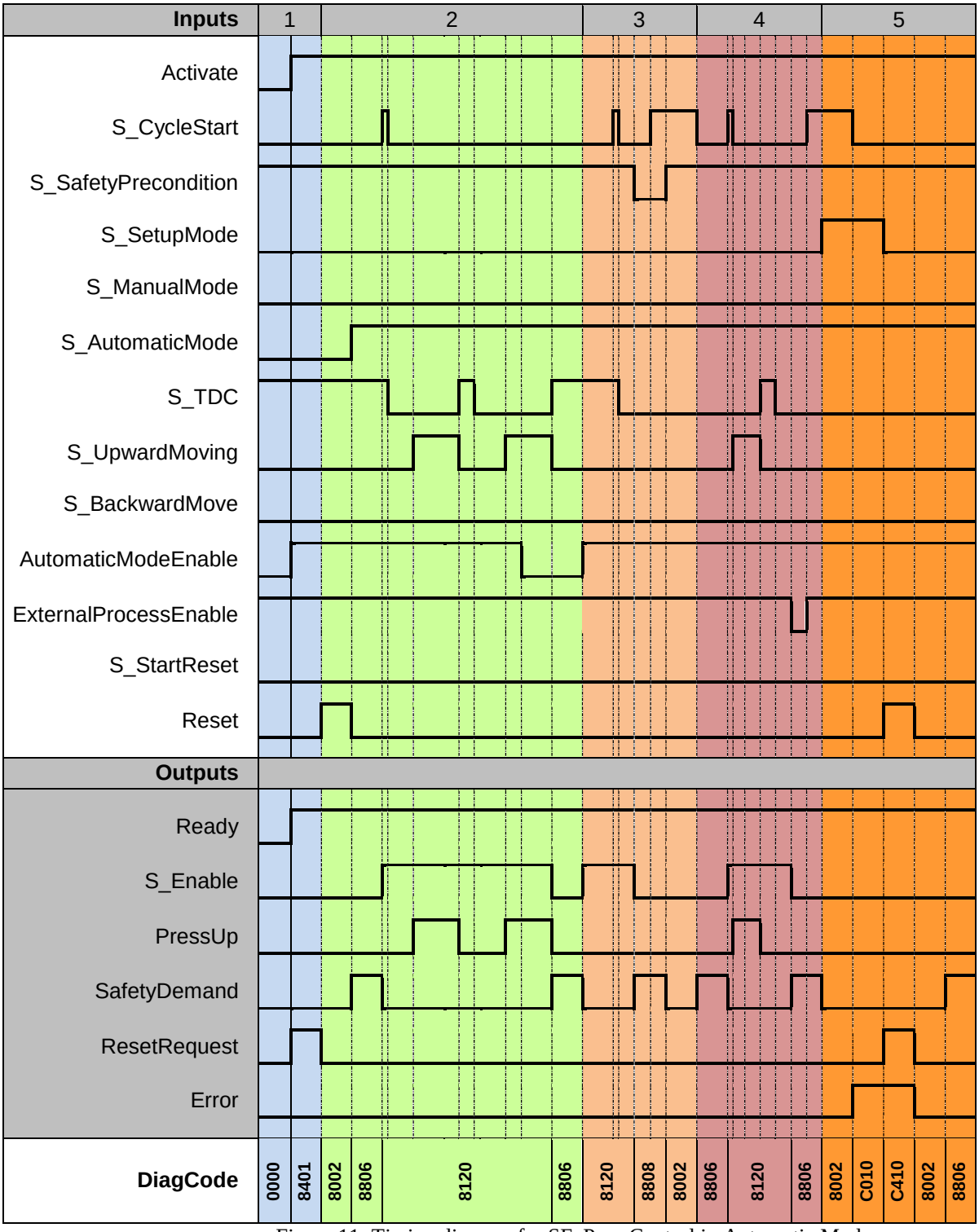


Figure 11: Timing diagram for SF_PressControl in Automatic Mode

1: Initialisation
2: Running an stop in automatic mode
3: During a stroke S_SafetyPrecondition is FALSE (for example an emergency stop button is pressed)
4: During a stroke the ExternalProcessEnable is FALSE
5: More than one mode is selected

4.2.4. Error Detection

The function block detects the following error conditions:

- A static TRUE signal at Reset input.
- After the Press cycle TDC is not active. (Overrun error)
- More than one mode is selected.

4.2.5. Error Behavior

In the event of an error, the S_Enable output is set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE.
A restart is inhibited until the error conditions are cleared and the safe state is acknowledged with Reset by the operator.

4.2.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting

FB-specific error codes:

C001	Reset Error 0	Static Reset detected in State 8401 Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
Cx10	Mode Error	More than one Mode is selected. IF (NoMode OR OnlyManual OR OnlySetup OR OnlyAutomatic) = TRUE THEN x = 4 ELSE x = 0 Output signals for x = 4 (C410): Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE Output signals for x = 0 (C010): Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in State Cx10 Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

Cx20	TDC Error	After the Press cycle TDC was not active. (Overrun) IF OnlySetup = TRUE THEN x = 4 ELSE x = 0 Output signals for x = 4 (C420): Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE Output signals for x = 0 (C020): Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Static Reset detected in State Cx20 Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8401	Init	Function block has activated and initiated. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8002	Mode Control	The selected Mode is analyzed. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE

8802	Manual Mode	Manual mode is selected and the FB is waiting to start the press cycle. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8804	Setup Mode	Setup mode is selected and the FB is waiting to start the press cycle. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8806	Automatic Mode	Automatic mode is selected and the FB is waiting to start the press cycle. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8808	Safety Lost	S_SafetyPreconditions is not TRUE. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8812	Start from Downward Position	The press has passed the TDC and the S_CycleStart signal is lost before the press is in upward moving. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8814	TDC Error for Diagnostic	After the Press cycle TDC was not active. (Overrun) In the setup mode this is no error but only for diagnostic. Ready = TRUE S_Enable = FALSE PressUp = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8100	Output active Manual Mode	The press cycle was started from manual mode. Ready = TRUE S_Enable = TRUE PressUp = FALSE → TRUE if R_TRIG at S_UpwardMoving SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

8110	Output active Setup Mode	The press cycle was started from setup mode. Ready = TRUE S_Enable = TRUE PressUp = FALSE → TRUE if R_TRIG at S_UpwardMoving SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8120	Output active Automatic Mode	The press cycle was started from automatic mode. Ready = TRUE S_Enable = TRUE PressUp = FALSE → TRUE if R_TRIG at S_UpwardMoving SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8130	Backwards Move	The press cycle was started from setup mode in backwards direction. Ready = TRUE S_Enable = TRUE PressUp = FALSE → TRUE if S_UpwardMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

4.3. Single Valve Monitoring

Owner: Jochen Ost

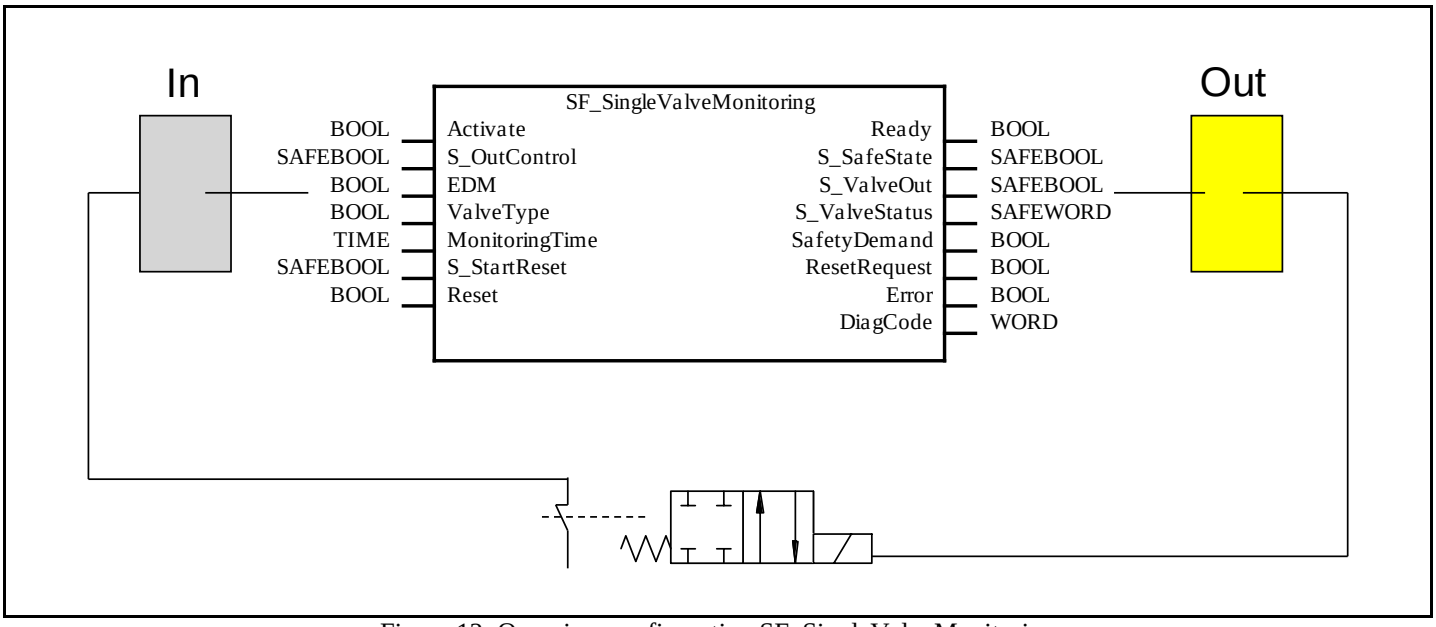


Figure 12: Overview configuration SF_SingleValveMonitoring

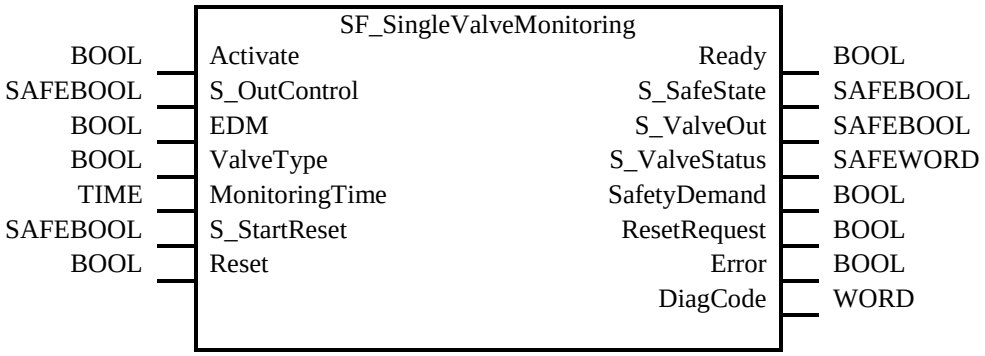
4.3.1. Applicable Safety Standards

Standards	Requirements
EN ISO 13849-1:2008	5.2.1: Stop function; stop initiated by safeguards shall put the machine in a safe state 6.2: Specification of categories: Fault detection (of the actuator, e.g. open circuits)
EN ISO 12100: 2010	6.2.11.4: Restart following power failure/spontaneous restart 6.2.11.6: Use of automatic monitoring
EN ISO 4413:2010	5.4.7.1 Unintended movement Control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators. This applies to all phases of the operation. 5.4.8.1 General requirements Provisions for diagnostic testing and condition monitoring should be made to facilitate preventive maintenance and trouble-shooting …..
EN ISO 4414:2010	5.4.6.1 Unintended movement In all phases of operation, control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators, particularly vertical and inclined motions.

Standards	Requirements
EN 692:2005+A1:2009 Machine tools – Mechanical presses – Safety (includes Amendment A1:2009) English version of DIN EN 692:2009-10	5.4.1.4…. If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases 1), at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated 5.4.1.6 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement irrespective of the condition of the other. Failure of either system shall be detected through monitoring and another closing stroke prevented. If failure of one system is self revealing i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required. 5.4.2.3 ... d) where for the valve monitoring function there is a need for sensors detecting the valve state, these sensors shall be an integral part of the valves. The valve may have an inherent monitoring system in which valve failure is self-revealing
EN 693:2001+A2:2011 Machine tools – Safety – Hydraulic presses (includes Amendment A2:2011)	5.4.1.3… If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases*, at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.4.1.5 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement, irrespective of the condition of the other. Failure of either system shall be detected through monitoring, and another closing stroke prevented. If failure of one system is self-revealing, i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required.
EN 12622:2009 Safety of machine tools – Hydraulic press brakes English translation of DIN EN 12622:2010-04	5.2.1 ... ... Failure of either system shall be detected through monitoring and another closing stroke prevented. 5.2.5 ... Table 2 Safety and reliability of control systems for single cycle manual load and/or unload or setting mode 5.2.5.2 Behaviour of the control system in case of failure ... If a fault occurs in the safety-related parts of these protective devices or control system then: a) an unexpected start up shall not be possible; d) the control system shall stop the press brake during the dangerous phase of the closing stroke within the response time specified by the manufacturer or, in other cases1) at the next demand upon the safety function or at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.2.6.1 Start ... Unexpected start up of hazardous movements, e.g. beam, back gauges, work-piece support, shall be prevented.

4.3.2. Interface Description

FB Name	**SF_SingleValveMonitoring**		
The FB monitors the switching behavior of fluidic valves, which confirm the safe state within a parameterizeable monitoring time (MonitoringTime) by a static feedback signal (spool position monitoring).			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_OutControl	SAFEBOOL	FALSE	Variable. FALSE: Safe state of the valve is requested TRUE: Safe state of the valve is not requested
EDM	BOOL	FALSE	Variable. Feedback signal of the valve
ValveType	BOOL	FALSE	Constant. This parameter allows configuring the polarity of the feedback signal indicating the safe position. FALSE: Safe state will be acknowledged by a LOW signal. TRUE: Safe state will be acknowledged by a HIGH signal.
MonitoringTime	TIME	T#0s	Constant. Within the monitoring time the feedback signal needs to acknowledge the requested switching of the valve.
S_StartReset	SAFEBOOL	FALSE	Variable or Constant. FALSE: No cold start. TRUE: cold start
Reset	BOOL	FALSE	Variable. Reset to acknowledge the restart inhibit or any error state.
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_SafeState	SAFEBOOL	FALSE	TRUE: Single valve is in the safe state, no error is present FALSE: Single valve is not in the safe state or an error occurred.
S_ValveOut	SAFEBOOL	FALSE	FALSE: De-energizes the solenoid of the valve. TRUE: Energizes the solenoid of the valve.
S_ValveStatus	SAFEWORD	16#0000	Output is used by the Function Block SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails. (e.g. the valve does not switch or the EDM signals are not valid). See part 4 section 4.6 ValveGroupControl for more details.
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: The closed position of the valve does not inevitably represent the safe state. In some applications the open position may be the safe state.			



Note: The safety integrity (PL/SIL) of the output S_SafeState depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC). Some applications may request that the feedback sensors are directly connected to fail-safe inputs of a safety PLC.

4.3.3. Functional Description

General

The SF_SingleValveMonitoring monitors the switching behavior of a fluid valve which provides a static monitoring signal for the safe spool position of the valve. The function block monitors the initial state of the valve via the feedback signal (EDM). The EDM needs to signal the safe state before the valve can be enabled by the FB. The feedback signal gets monitored as well after the valve has been enabled by the FB. Should the EDM signal not correspond to the expected signal after the MonitoringTime, the output S_ValveOut is set to FALSE. This state is indicated by the ERROR output too. An appropriate diagnostic message is generated.

The FB signals also the safe state via the output S_SafeState (TRUE) if the safe position of the valve spool is detected and no error is present. This output may be used to implement a restart inhibit of the machine (e.g. start of a new cycle) if either the valve is not in its safe state or there was any error detected. (e.g. mismatch between detected and expected EDM signal). Please be aware that the safety integrity level of the S_SafeState output depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC).

The output S_ValveStatus can be used in combination with the SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails. (e.g. the valve does not switch or the EDM signal is not corresponding to the expected signal).

The polarity of the EDM signal is configurable via input ValveType, as the market offers valves where the acknowledge signal of the safe spool position can be either TRUE or FALSE. However it should be considered that regarding to the safety principle, a HIGH signal acknowledges the safe state, valves should be used which confirm the safe spool position with a HIGH signal.

An error can only be cleared by a rising trigger at the Reset input, when the EDM signal acknowledges the safe spool position adequately.

The EDM signal needs to be incorporated depending on the diagnostic coverage (DC) that is required for the safety function. Some applications may require connecting the EDM signal directly to a fail-safe input.

Optional startup inhibits:

Startup inhibit in the event of block activation.

The S_StartReset allows switching the valve when the PES is started without any additional Reset command. This input shall only be activated (TRUE) if it is ensured that no hazardous situation can occur.

State Diagram

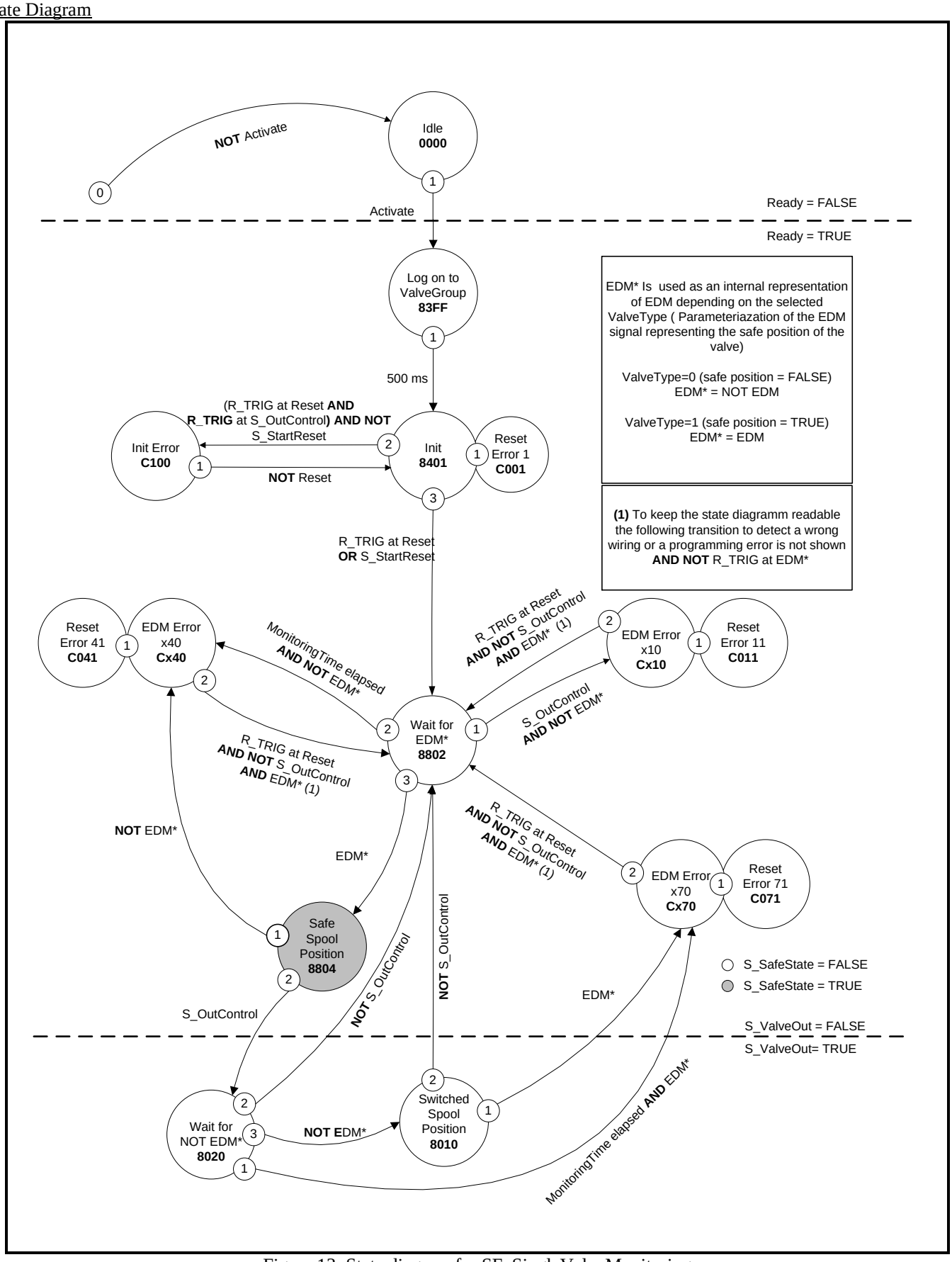


Figure 13: State diagram for SF_SingleValveMonitoring

Typical Timing Diagrams

ValveType = TRUE
S_StartReset = FALSE

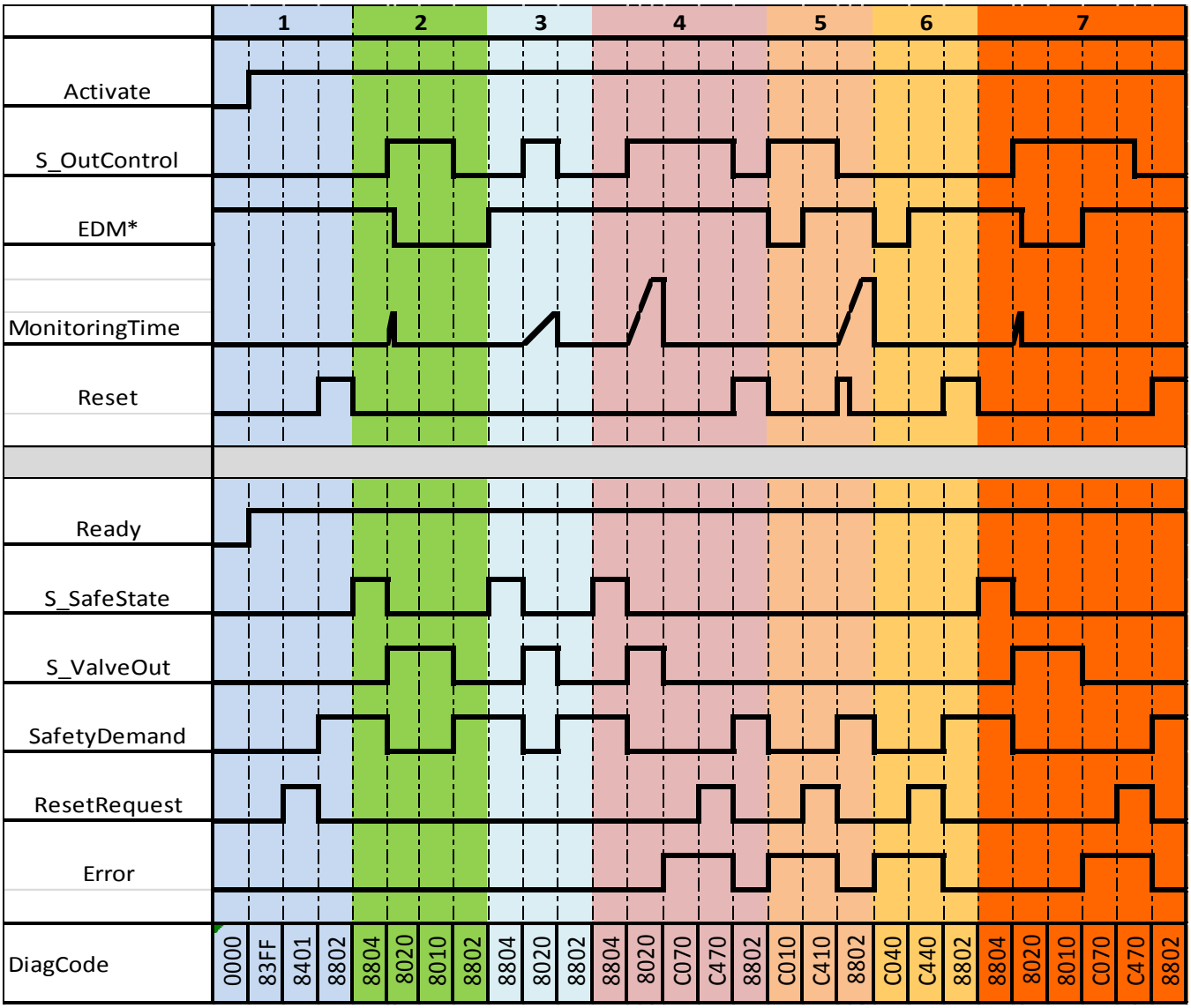


Figure 14: Timing diagrams for SF_SingleValveMonitoring

Normal operation
1: Initialisation
2: Normal switching on and off.
3: Normal switching on but S_OutControl switches off befor the EDM signal is FALSE.
Errors
4: Switching on but the EDM signal doesn't change befor the Monitoring time elapsed.
5: S_OutControl is set to TRUE but the EDM signal is FALSE.
6: The Monitoring time elapsed befor the EDM signal was set to TRUE.
7: Normal switching on the valve, but while the S_OutControl signal is still TRUE the EDM signal changes to TRUE.

4.3.4. Error Detection

The following conditions force a transition to the Error state:

- Invalid static Reset signal in the process.
- Invalid EDM signal in the process.
- S_OutControl and Reset are incorrectly interconnected due to programming error.
- S_OutControl changes the state before the EDM signal acknowledge the commanded position of the spool (process error)

4.3.5. Error Behavior

In error states, the outputs are as follows:

- In the event of an error, the S_ValveOut and S_SafeState are set to FALSE and remains FALSE until the error got fixed and the restart inhibit cleared by a rising edge at the Reset input.
- An EDM error must always be reset by a rising trigger at Reset input.
- A Reset error can be reset by setting Reset to FALSE.
- After block activation, the optional startup inhibit can be reset by a rising edge at the Reset input.

4.3.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	Output Setting

FB-specific error codes:

C100	Init Error	Similar signals at S_OutControl and Reset (R_TRIG at same cycle) detected (may be a programming error) Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C100 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C001	Reset Error 1	Static Reset signal in state 8401. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C001 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 11	Static Reset signal in state Cx10. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C011 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C041	Reset Error 41	Static Reset signal in state Cx40. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C041 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C071	Reset Error 71	Static Reset signal in state Cx70. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C071 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

Cx10	EDM Error x10	The signal at EDM* is not valid in the initial actuator state. In state 8802 the EDM* signal is FALSE when enabling S_OutControl Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR S_OutControl= TRUE S_ValveStatus = C010 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C410 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx40	EDM Error x40	The signal at EDM* is not valid either in state 8802 or 8804. In state 8802 the EDM* signal is FALSE and the monitoring time has elapsed. In state 8804 the EDM* signal becomes false for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR S_OutControl= TRUE S_ValveStatus = C040 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C440 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx70	EDM Error x70	The signal at EDM* is not valid in state 8010 or in state 8020. In state 8020 the EDM* signal is TRUE and the monitoring time has elapsed or the S_OutControl was set to FALSE before EDM* was confirming the switched spool position. In state 8010 the EDM* signal becomes TRUE for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR S_OutControl = TRUE S_ValveStatus = C070 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C470 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 0000 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
83FF	Log on to valve group	Wait state to ensure that the SF_ValveGroupControl can be used only in combination with valve monitoring FBs. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 83FF SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8401	Init	Block activation startup inhibit is active. Reset required. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8401 SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8802	Wait for EDM*	Wait for EDM signal to confirm the safe spool position of the valve. Timer starts when state is entered. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8802 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8804	Safe spool position	The EDM signal confirms that the valve are in the safe state (safe spool position) Ready = TRUE S_SafeState = TRUE S_ValveOut = FALSE S_ValveStatus = 8804 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8010	Switched Spool Position	Wait for EDM signal to confirm the switched spool position of the valve. Timer starts when state is entered. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8010 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8020	Wait for NOT EDM*	The EDM signal confirms that the valve is in the switched spool position. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8020 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

4.4. Single Valve Cycle Monitoring

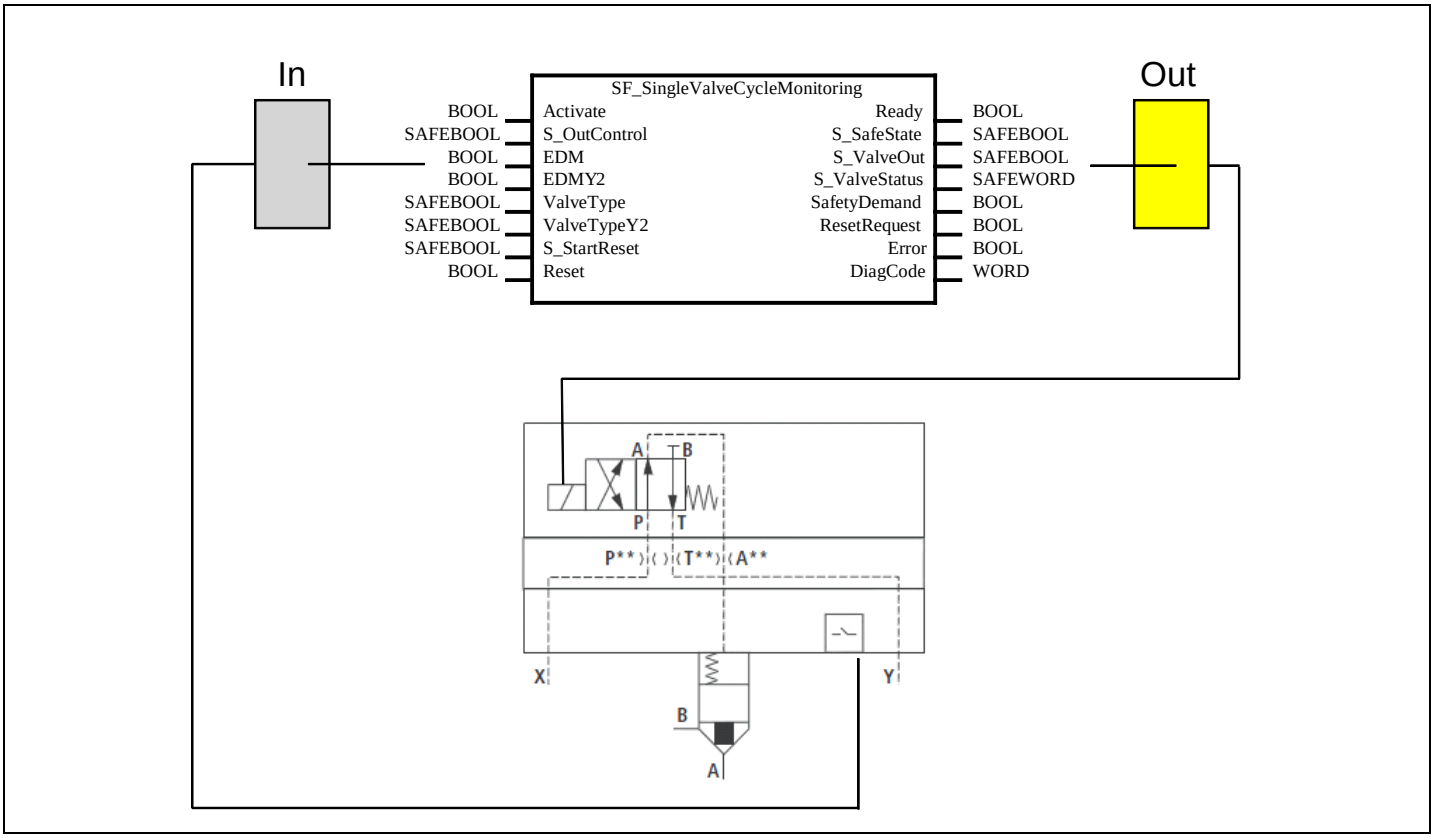


Figure 15: Basic configuration for SF_SingleValveCycleMonitoring

Cartridge valves operate in dependence upon pressure. Three pressurized areas are essential for the function: A1, A2, and A3. Thus there can be a fluid flow even if the pilot valve is closed. However if the EDM signal of the cartridge valve indicates the closed spool position there is no fluid flow.

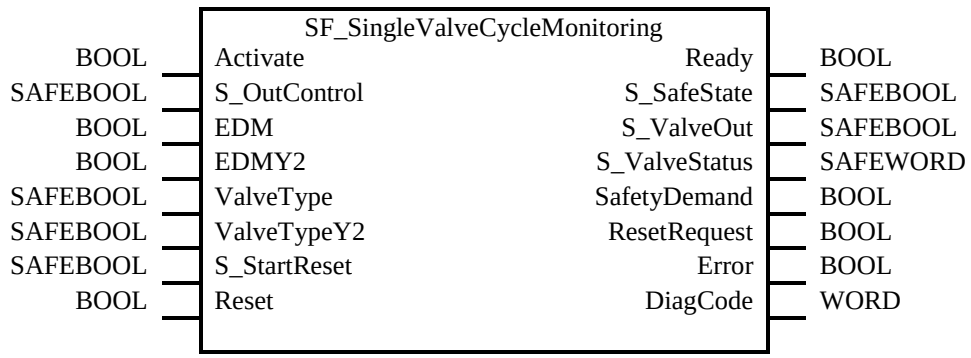
4.4.1. Applicable Safety Standards

Standards	Requirements
EN ISO 13849-1:2008	5.2.1: Stop function; stop initiated by safeguards shall put the machine in a safe state 6.2: Specification of categories: Fault detection (of the actuator, e.g. open circuits)
EN ISO 12100: 2010	6.2.11.4: Restart following power failure/spontaneous restart 6.2.11.6: Use of automatic monitoring
EN ISO 4413:2010	5.4.7.1 Unintended movement Control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators. This applies to all phases of the operation. 5.4.8.1 General requirements Provisions for diagnostic testing and condition monitoring should be made to facilitate preventive maintenance and trouble-shooting …..
EN ISO 4414:2010	5.4.6.1 Unintended movement In all phases of operation, control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators, particularly vertical and inclined motions.

Standards	Requirements
EN 692:2005+A1:2009 Machine tools – Mechanical presses – Safety (includes Amendment A1:2009) English version of DIN EN 692:2009-10	5.4.1.4…. If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases 1), at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated 5.4.1.6 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement irrespective of the condition of the other. Failure of either system shall be detected through monitoring and another closing stroke prevented. If failure of one system is self revealing i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required. 5.4.2.3 ... d) where for the valve monitoring function there is a need for sensors detecting the valve state, these sensors shall be an integral part of the valves. The valve may have an inherent monitoring system in which valve failure is self-revealing
EN 693:2001+A2:2011 Machine tools – Safety – Hydraulic presses (includes Amendment A2:2011)	5.4.1.3… If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases*, at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.4.1.5 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement, irrespective of the condition of the other. Failure of either system shall be detected through monitoring, and another closing stroke prevented. If failure of one system is self-revealing, i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required.
EN 12622:2009 Safety of machine tools – Hydraulic press brakes English translation of DIN EN 12622:2010-04	5.2.1 ... ... Failure of either system shall be detected through monitoring and another closing stroke prevented. 5.2.5 ... Table 2 Safety and reliability of control systems for single cycle manual load and/or unload or setting mode 5.2.5.2 Behaviour of the control system in case of failure ... If a fault occurs in the safety-related parts of these protective devices or control system then: a) an unexpected start up shall not be possible; d) the control system shall stop the press brake during the dangerous phase of the closing stroke within the response time specified by the manufacturer or, in other cases1) at the next demand upon the safety function or at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.2.6.1 Start ... Unexpected start up of hazardous movements, e.g. beam, back gauges, work-piece support, shall be prevented.

4.4.2. Interface Description

FB Name	**SF_SingleValveCycleMonitoring**		
The FB monitors the switching behavior of fluidic valve (cartridge valve), in that manner, that per machine cycle a signal change of the spool monitoring signal must be recognized. In case there is no signal change within a machine cycle the next cycle will be inhibited.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_OutControl	SAFEBOOL	FALSE	FALSE: Safe state of the cartridge valve is requested TRUE: Safe state of the cartridge valve is not requested
EDM	BOOL	FALSE	Feedback signal of the cartridge valve
EDMY2	BOOL	FALSE	Signal indicating until the signal change of EDM must have happened.
ValveType	BOOL	FALSE	Constant. This parameter allows configuring the polarity of the feedback signal indicating the safe position of the cartridge valve. FALSE: Safe state will be acknowledged by a LOW signal. TRUE: Safe state will be acknowledged by a HIGH signal.
ValveTypeY2	BOOL	FALSE	Constant. This parameter allows configuring the polarity of the feedback signal indicating the safe position of valve Y2. FALSE: Safe state will be acknowledged by a LOW signal. TRUE: Safe state will be acknowledged by a HIGH signal.
S_StartReset	SAFEBOOL	FALSE	FALSE: No cold start. TRUE: cold start
Reset	BOOL	FALSE	Reset to acknowledge the restart inhibit or any error state.
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_SafeState	SAFEBOOL	FALSE	TRUE: Cartridge valve is in the safe state, no error is present FALSE: Cartrigde valve is not in the safe state or an error occurred.
S_ValveOut	SAFEBOOL	FALSE	FALSE: De-energizes the solenoid of the pilot valve. TRUE: Energizes the solenoid of the pilot valve.
S_ValveStatus	SAFEWORD	16#0000	Output is used by the Function Block SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails. (e.g. the valve does not switch or the EDM signals are not valid). See part 4 section 4.6ValveGroupControlfor more details.
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: The closed position of the valve does not inevitably represent the safe state. In some applications the open position may be the safe state.			



Note: The safety integrity (PL/SIL) of the output S_SafeState depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC). Some applications may request that the feedback sensors are directly connected to fail-safe inputs of a safety PLC.

4.4.3. Functional Description

General

The SF_ SingleValveCycleMonitoring monitors the switching behavior of fluidic valve (cartridge valve), in that manner, that per machine cycle only a signal change (the EDM signal must not stay at LOW) of the spool monitoring signal must be detected. In case there was no signal change within a machine cycle the next cycle will be inhibited.
In presses the valve may be used to control the downstroke. Before the downstroke can be commanded the valve must indicate the safe position as well the upstroke valve must be closed (EDMY2 = TRUE). Before the upstroke is permitted there must be at least a signal change at EDM. The minimum time period for the signal to indicate a signal change can be parameterized as filter time in the input module. The FB also monitors the EDMY2 for dynamic changes. If for some reasons the EDMY2 stucks at HIGH this FB inhibits the activation of the cartridge valve too. Thus standard inputs for EDMY2 can be used.

The function block monitors the initial state of the valves via the feedback signal (EDM). The EDMs need to signal the safe state before the valve can be enabled by the FB. The FB acknowledges the safe state via the output S_SafeState (TRUE) if the safe position of the valve spool is detected and no error is present. This output may be used to implement a restart inhibit of the machine (e.g. start of a new cycle) if either the valve is not in its safe state or there was any error detected.
Please be aware that the safety integrity level of the S_SafeState output depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC).

The output S_ValveStatus can be used in combination with the SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails.

The polarity of the EDM signal is configurable via the input ValveType since the market offers valves where the acknowledge signal of the safe spool position can be either TRUE or FALSE. However it should be considered that regarding to the safety principle, a HIGH signal acknowledges the safe state, valves should be used which confirm the safe spool position with a HIGH signal.

An error can only be cleared by arising trigger at the Reset input, when the EDM signal acknowledge the safe spool position adequately.

The EDM signals needs to be incorporated depending on the diagnostic coverage (DC) that is required for the safety function. Some applications may require connecting the EDM signals directly to a fail-safe input.

Optional startup inhibits:
Startup inhibit in the event of block activation.

The S_StartReset allows switching the valve when the PES is started without any additional Reset command. This input shall only be activated (TRUE) if it is ensured that no hazardous situation can occur.

State Diagram

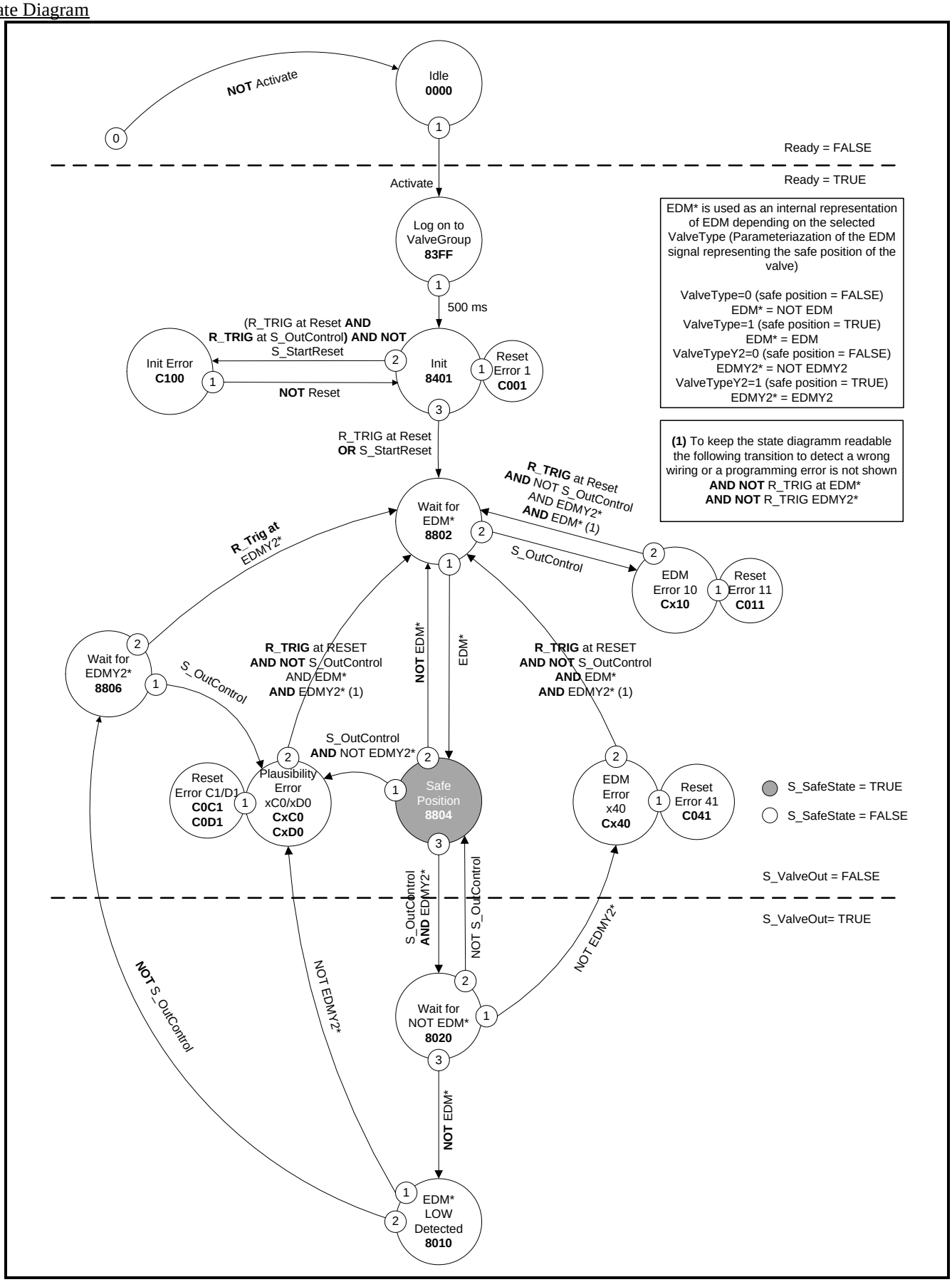


Figure 16: State diagram for SF_SingleValveCycleMonitoring

Typical Timing Diagrams

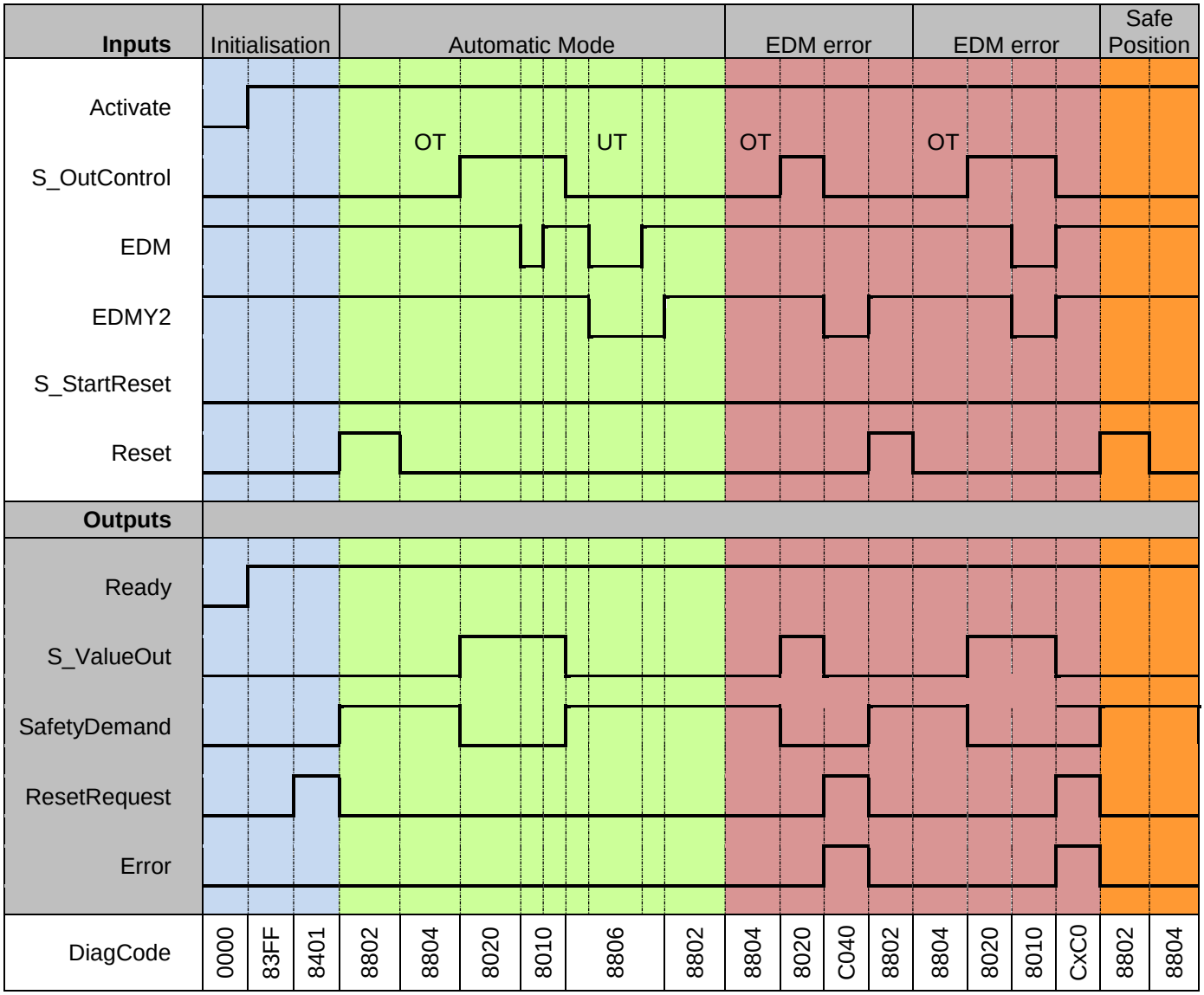


Figure 17: Timing diagram for SF_SingleValveCycleMonitoring

4.4.4. Error Detection

The following conditions force a transition to the Error state:

- Invalid static Reset signal in the process.
- EDM signal does not change the state within one machine cycle.
- EDMY2 signal does not change within one machine cycle.
- EDMY2 becomes/is FALSE when S_OutControl is TRUE
- S_OutControl and Reset are incorrectly interconnected due to programming error.
- S_OutControl becomes TRUE before the valve acknowledges its safe (initial) position

4.4.5. Error Behavior

In error states, the outputs are as follows:

- In the event of an error, the S_ValveOut and S_SafeState are set to FALSE and remains FALSE until the error got fixed and the restart inhibit cleared by a rising edge at the Reset input.
- An EDM error must always be reset by a rising trigger at Reset.
- A Reset error can be reset by setting Reset to FALSE.
- After block activation, the optional startup inhibit can be reset by a rising edge at the Reset input.

4.4.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	Output Setting

FB-specific error codes:

C100	Init Error	Similar signals at S_OutControl and Reset (R_TRIG at same cycle) detected (may be a programming error) Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C100 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C001	Reset Error 1	Static Reset signal in state 8401. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C001 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 11	Static Reset signal in state Cx10. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C011 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C041	Reset Error 41	Static Reset signal in state Cx40. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C041 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C0C1	Reset Error C1	Static Reset signal in state CxC0. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C0C1 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C0D1	Reset Error D1	Static Reset signal in state CxD0. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C0D1 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

Cx10	EDM Error x10	The signal at EDM* is not valid in the initial actuator state. In state 8802 the EDM* signal is FALSE when enabling S_OutControl Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR EDMY2*=FALSE OR S_OutControl = TRUE S_ValveStatus = C010 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C410 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx40	EDM Error x40	The signal at EDM* is not valid in the state 8020. In state 8020 the EDM* signal is TRUE when EDMY2* becomes FALSE, which indicates a new cycle. IF EDM* = FALSE OR EDMY2*=FALSE OR S_OutControl = TRUE S_ValveStatus = C040 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C440 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
CxC0	Plausibility Error xC0	EDMY2* becomes FALSE in state 8010, which indicates a new cycle, before the valve is deactivated. (S_OutControl = FALSE) Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR EDMY2*=FALSE OR S_OutControl = TRUE S_ValveStatus = C0C0 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C4C0 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

CxD0	Plausibility Error xD0	S_OutControl becomes TRUE in state 8806. EDMY2* is still FALSE which indicates that the cycle of Y2 is not finished. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM* = FALSE OR EDMY2*=FALSE OR S_OutControl = TRUE S_ValveStatus = C0D0 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C4D0 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 0000 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
83FF	Log on to valve group	Wait state to ensure that the SF_ValveGroupControl can be used only in combination with valve monitoring FBs. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 83FF SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8401	Init	Block activation startup inhibit is active. Reset required. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8401 SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8802	Wait for EDM	The output S_ValveOut is set to FALSE. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8802 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8804	Safe spool position	The EDM signals confirm that the valves are in the safe state (safe spool position) Ready = TRUE S_SafeState = TRUE S_ValveOut = FALSE S_ValveStatus = 8804 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE

8806	Wait for EDM Y2	The EDM Y2 signal needs to change. Ready = TRUE S_SafeState = TRUE S_ValveOut = FALSE S_ValveStatus = 8806 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8010	EDM LOW Detected	The EDM signal change got detected. Now it is allowed to switch the valve again. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8010 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8020	Wait for NOT EDM	The output S_ValveOut is set to TRUE. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8020 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

4.5. Double Valve Monitoring

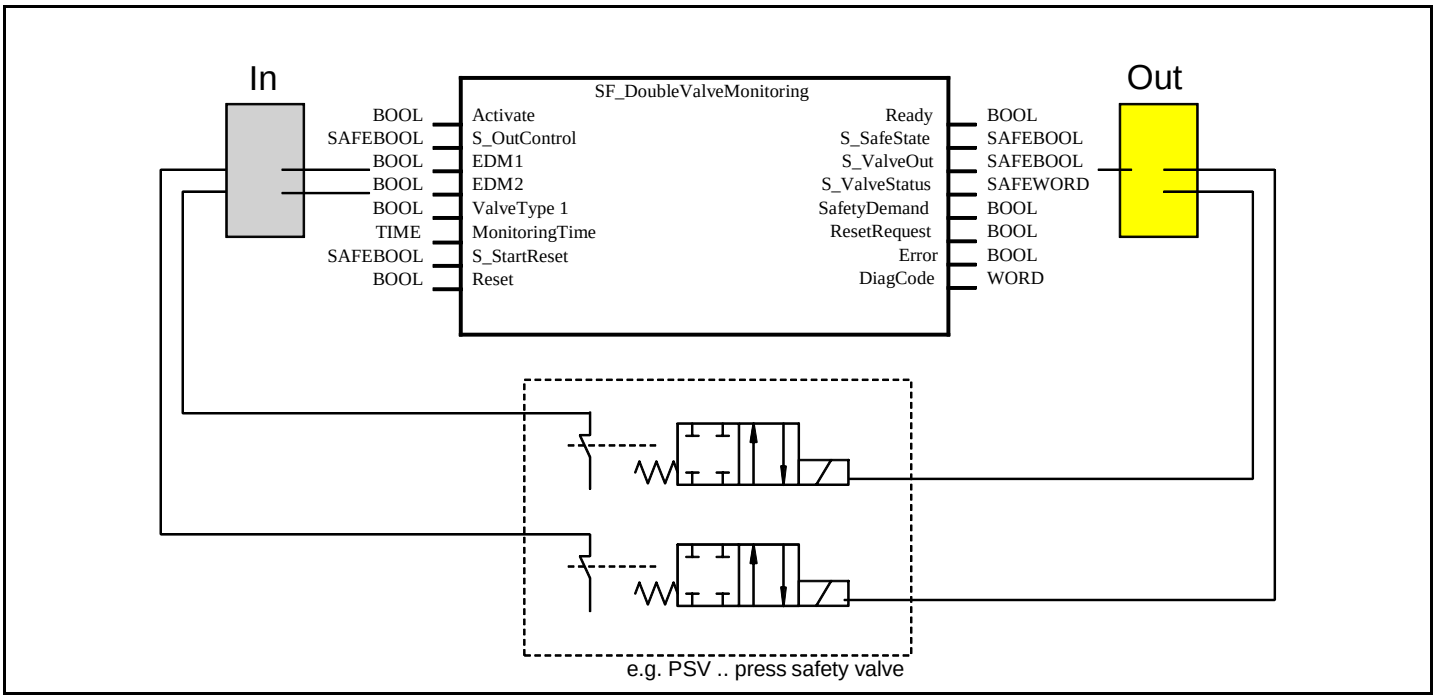


Figure 18: Basic configuration for SF_DoubleValveMonitoring

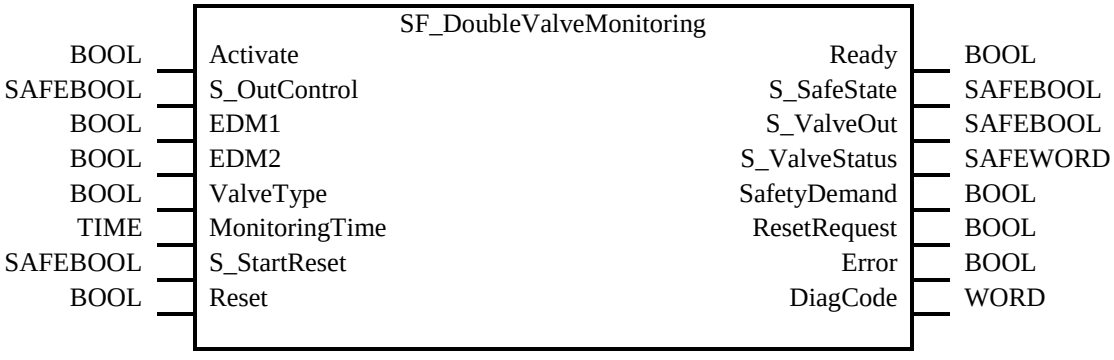
4.5.1. Applicable Safety Standards

Standards	Requirements
EN ISO 13849-1:2008	5.2.1: Stop function; stop initiated by safeguards shall put the machine in a safe state 6.2: Specification of categories: Fault detection (of the actuator, e.g. open circuits)
EN ISO 12100: 2010	6.2.11.4: Restart following power failure/spontaneous restart 6.2.11.6: Use of automatic monitoring
EN ISO 4413:2010	5.4.7.1 Unintended movement Control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators. This applies to all phases of the operation. 5.4.8.1 General requirements Provisions for diagnostic testing and condition monitoring should be made to facilitate preventive maintenance and trouble-shooting …..
EN ISO 4414:2010	5.4.6.1 Unintended movement In all phases of operation, control systems shall be designed to prevent unintended hazardous movement and improper sequencing of actuators, particularly vertical and inclined motions.

Standards	Requirements
EN 692:2005+A1:2009 Machine tools – Mechanical presses – Safety (includes Amendment A1:2009) English version of DIN EN 692:2009-10	5.4.1.4…. If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases 1), at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated 5.4.1.6 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement irrespective of the condition of the other. Failure of either system shall be detected through monitoring and another closing stroke prevented. If failure of one system is self revealing i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required. 5.4.2.3 ... d) where for the valve monitoring function there is a need for sensors detecting the valve state, these sensors shall be an integral part of the valves. The valve may have an inherent monitoring system in which valve failure is self-revealing
EN 693:2001+A2:2011 Machine tools – Safety – Hydraulic presses (includes Amendment A2:2011)	5.4.1.3… If a fault occurs in the safety related parts of these protective devices or control system then: a) an unintended start-up shall not be possible; b) the safe functioning of the protective device shall be maintained; c) it shall be possible to stop the machine during the dangerous movement; d) the control system shall stop the machine immediately during the dangerous phase of the closing stroke or, in other cases*, at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.4.1.5 ... The redundant and monitored press control system shall operate in two separate functioning systems. Either system shall be independently capable of stopping the hazardous movement, irrespective of the condition of the other. Failure of either system shall be detected through monitoring, and another closing stroke prevented. If failure of one system is self-revealing, i.e. the loss of the function itself prevents the next operating cycle, further monitoring of that system is not required.
EN 12622:2009 Safety of machine tools – Hydraulic press brakes English translation of DIN EN 12622:2010-04	5.2.1 ... ... Failure of either system shall be detected through monitoring and another closing stroke prevented. 5.2.5 ... Table 2 Safety and reliability of control systems for single cycle manual load and/or unload or setting mode 5.2.5.2 Behaviour of the control system in case of failure ... If a fault occurs in the safety-related parts of these protective devices or control system then: a) an unexpected start up shall not be possible; d) the control system shall stop the press brake during the dangerous phase of the closing stroke within the response time specified by the manufacturer or, in other cases1) at the next demand upon the safety function or at the latest at the end of the operating cycle; e) the control system shall prevent any initiation of the next operating production cycle until the fault is eliminated. 5.2.6.1 Start ... Unexpected start up of hazardous movements, e.g. beam, back gauges, work-piece support, shall be prevented.

4.5.2. Interface Description

FB Name	**SF_DoubleValveMonitoring**		
The FB monitors the switching behavior of fluidic double valves (press safety valves), which confirm the safe state within a parameterizable monitoring time (MonitoringTime) by a static feedback signal for each valve (spool position monitoring). Both valves used should be of the same type.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_OutControl	SAFEBOOL	FALSE	FALSE: Safe state of the double valve is requested TRUE: Safe state of the double valve is not requested
EDM1	BOOL	FALSE	Feedback signal of valve 1
EDM2	BOOL	FALSE	Feedback signal of valve 2
ValveType	BOOL	FALSE	Constant. This parameter allows configuring the polarity of the feedback signal indicating the safe position. FALSE: Safe state will be acknowledged by a LOW signal. TRUE: Safe state will be acknowledged by a HIGH signal
MonitoringTime	TIME	T#0s	Constant. Within the monitoring time the feedback signals needs to acknowledge the requested switching of the valve.
S_StartReset	SAFEBOOL	FALSE	FALSE: No cold start TRUE: cold start
Reset	BOOL	FALSE	Reset to acknowledge the restart inhibit or any error state.
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_SafeState	SAFEBOOL	FALSE	TRUE: Double valve is in the safe state, no error is present FALSE: Double valve is not in the safe state or an error occurred.
S_ValveOut	SAFEBOOL	FALSE	FALSE: De-energizes the solenoid of both valves TRUE: Energizes the solenoid of both valves.
S_ValveStatus	SAFEWORD	16#0000	Output is used by the Function Block SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails. (e.g. the valve does not switch or the EDM signals are not valid). See part 4 section 4.6ValveGroupControlfor more details.
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FLASE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: The closed position of the valve spool does not inevitably represent the safe state. In some applications the open position may be the safe state.			



Note: The safety integrity (PL/SIL) of the output S_SafeState depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC). Some applications may request that the feedback sensors are directly connected to fail-safe inputs of a safety PLC.

4.5.3. Functional Description

General

The SF_DoubleValveMonitoring monitors the switching behavior of a fluid double valve (e.g. press safety valve) which provides two static monitoring signals for the safe spool position of each single valve. The function block monitors the initial state of the valves via the feedback signals (EDM1 and EDM2). Both EDMs need to signal the safe state before the double valve can be enabled by the FB. The feedback signals of each valve get monitored as well after the double valve has been enabled by the FB. Should the EDM signals not correspond to the expected signals after the MonitoringTime, the output S_ValveOut is set to FALSE. This state is indicated by the ERROR output too. An appropriate diagnostic message is generated.
The FB signals also the safe state via the output S_SafeState (TRUE) if the safe position of the valve spools is detected and no error is present. This output may be used to implement a restart inhibit of the machine (e.g. start of a new cycle) if either the double valve is not in its safe state or there was any error detected. (e.g. mismatch between detected and expected EDM signals). Please be aware that the safety integrity level of the S_SafeState output depends on the characteristic data ($MTTF_d$) of the feedback sensors and their integration into the safety control system. (e.g. DC).
The output S_ValveStatus can be used in combination with the SF_ValveGroupControl to control multiple valves as a valve group, where other valves need to be switched into the safe state if any valve of the valve group fails. (e.g. the double valve does not switch or the EDM signals are not corresponding to the expected signals).

The polarity of the EDM signals is configurable via input ValveType, as the market offers valves where the acknowledge signal of the safe spool position can be either TRUE or FALSE. However it should be considered that regarding to the safety principle, a HIGH signal acknowledges the safe state, valves should be used which confirm the safe spool position with a HIGH signal.

An error can only be cleared by a rising trigger at the Reset input, when the EDM signals acknowledge the safe spool position adequately.

The EDM signals need to be incorporated depending on the diagnostic coverage (DC) that is required for the safety function. Some applications may require connecting the EDM signals directly to a fail-safe input.

Two single EDM signals must be used for an exact diagnosis of the connected double valve. A common feedback signal from the two connected valves must be used for a restricted yet simple diagnostic function of the connected actuators. When doing so, the user must connect this common signal to both parameter EDM1 and parameter EDM2. EDM1 and EDM2 are then controlled by the same signal.

Optional startup inhibits:

Startup inhibit in the event of block activation.

The S_StartReset allows switching the double when the PES is started without any additional Reset command. This input shall only be activated (TRUE) if it is ensured that no hazardous situation can occur.

State Diagram

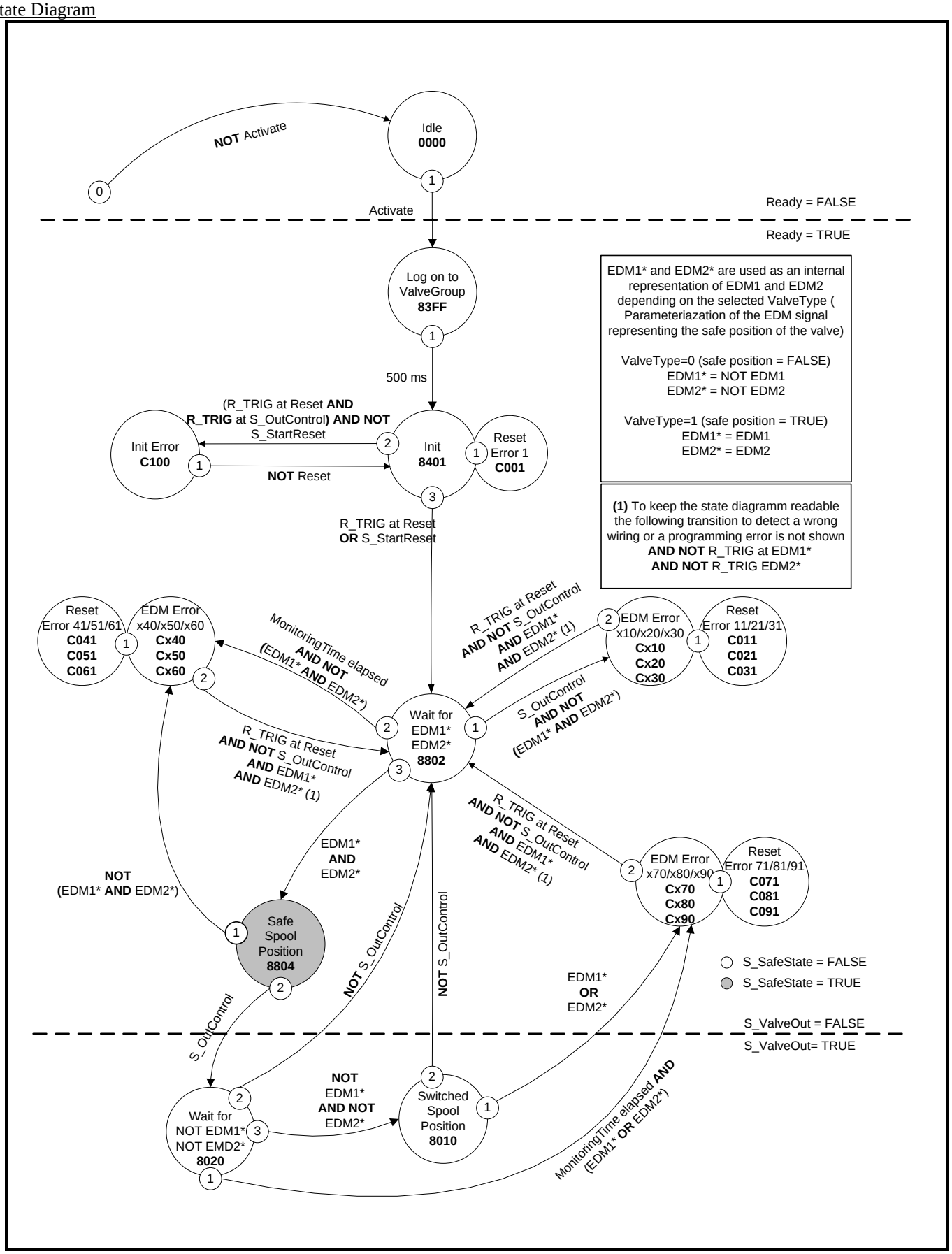


Figure 19: State diagram for SF_DoubleValveMonitoring

Typical Timing Diagrams

S_StartReset = FALSE

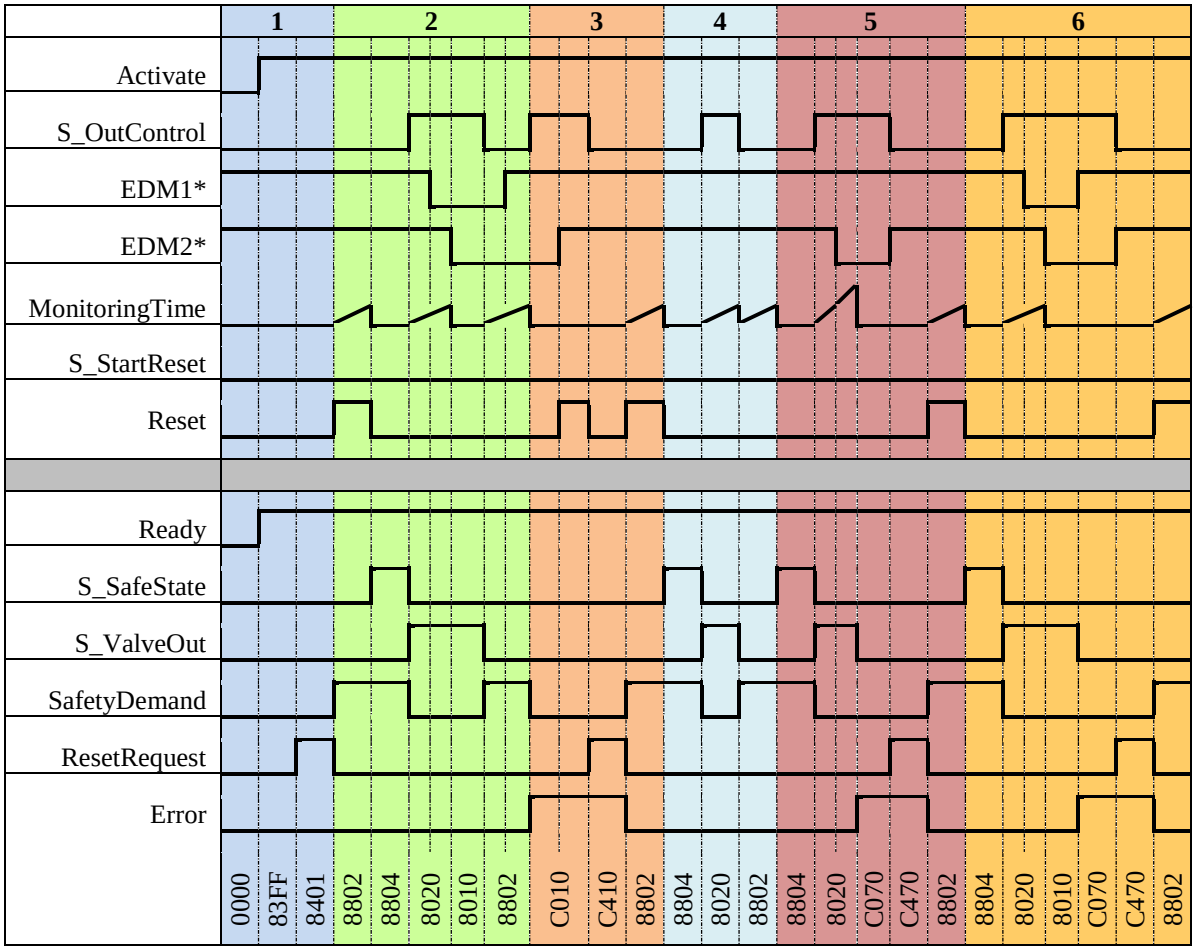


Figure 20: Timing diagrams for SF_DoubleValveMonitoring

1: Initialisation
2: Normal operation with switching on an off.
3: Error – S_OutControl is TRUE but EDM2 has not switched to TRUE after the last operation.
4: The S_OutControl is set to FALSE before the EDM Signals are switched to FALSE.
5: Error – The monitoring time elapsed and the EDM1 signal has not switched to FALSE.
6: Error – The EDM1 signal switched to TRUE but the S_OutControl signal is still TRUE.

4.5.4. Error Detection

The following conditions force a transition to the Error state:

- Invalid static Reset signal in the process.
- Invalid EDM signal(s) in the process.
- S_OutControl and Reset are incorrectly interconnected due to programming error.
- S_OutControl changes the state before the EDM signals acknowledge the commanded position of the spool (process error)

4.5.5. Error Behavior

In error states, the outputs are as follows:

- In the event of an error, the S_ValveOut and S_SafeState are set to FALSE and remain FALSE until the error got fixed and the restart inhibit cleared by a rising edge at the Reset input..
- An EDM error must always be reset by a rising trigger at Reset.
- A Reset error can be reset by setting Reset to FALSE.
- After block activation, the optional startup inhibit can be reset by a rising edge at the Reset input.

4.5.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	Output Setting

FB-specific error codes:

C100	Init Error	Similar signals at S_OutControl and Reset (R_TRIG at same cycle) detected (may be a programming error) Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C100 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C001	Reset Error 1	Static Reset signal in state 8401. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C001 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 11	Static Reset signal in state Cx10. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C011 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 21	Static Reset signal in state Cx20. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C021 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C031	Reset Error 31	Static Reset signal in state Cx30. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C031 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C041	Reset Error 41	Static Reset signal in state Cx40. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C041 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

C051	Reset Error 51	Static Reset signal in state Cx50. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C051 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C061	Reset Error 61	Static Reset signal in state Cx60. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C061 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C071	Reset Error 71	Static Reset signal in state Cx70. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C071 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C081	Reset Error 81	Static Reset signal in state Cx80. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C081 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C091	Reset Error 91	Static Reset signal in state Cx90. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = C091 SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
Cx10	EDM Error x10	The signal at EDM1* is not valid in the initial actuator state. In state 8802 the EDM1* signal is FALSE when enabling S_OutControl Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C010 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C410 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

Cx20	EDM Error x20	The signal at EDM2* is not valid in the initial actuator state. In state 8802 the EDM2* signal is FALSE when enabling S_OutControl Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C020 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C420 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx30	EDM Error x30	The signals at EDM1* and EDM2* are not valid in the initial actuator states. In state 8802 the EDM1* and EDM2* signals are FALSE when enabling S_OutControl. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C030 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C430 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx40	EDM Error x40	The signal at EDM1* is not valid either in state 8802 or 8804. In state 8802 the EDM1* signal is FALSE and the monitoring time has elapsed. In state 8804 the EDM1* signal becomes false for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C040 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C440 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

Cx50	EDM Error x50	The signal at EDM2* is not valid either in state 8802 or 8804. In state 8802 the EDM2* signal is FALSE and the monitoring time has elapsed. In state 8804 the EDM2* signal becomes false for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C050 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C450 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx60	EDM Error x60	The signals at EDM1* and EDM2* are not valid either in state 8802 or 8804. In state 8802 the EDM1* and EDM2* signals are FALSE and the monitoring time has elapsed. In state 8804 the EDM1* and EDM2* signal become false for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2*=FALSE OR S_OutControl= TRUE S_ValveStatus = C060 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C460 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx70	EDM Error x70	The signal at EDM1* is not valid in state 8010 or in state 8020. In state 8020 the EDM1* signal is TRUE and the monitoring time has elapsed or the S_OutControl was set to FALSE before EDM1* was confirming the switched spool position. In state 8010 the EDM1* signal becomes TRUE for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2* = FALSE OR S_OutControl = TRUE S_ValveStatus = C070 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C470 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

Cx80	EDM Error x80	The signal at EDM2* is not valid in state 8010 or in 8020. In state 8020 the EDM2* signal is TRUE and the monitoring time has elapsed or the S_OutControl was set to FALSE before EDM2* was confirming the switched spool position. In state 8010 the EDM2* signal becomes TRUE for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2* = FALSE OR S_OutControl = TRUE S_ValveStatus = C080 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C480 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
Cx90	EDM Error x90	The signal at EDM1* and EDM2* is not valid in state 8010 or in 8020. In state 8020 the EDM1* and EDM2* signal is TRUE and the monitoring time has elapsed or the S_OutControl was set to FALSE before EDM1* and EDM2* was confirming the switched spool position. In state 8010 the EDM1* and EDM2* signal becomes TRUE for any reason although the valve was not commanded to switch. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE IF EDM1* = FALSE OR EDM2* = FALSE OR S_OutControl = TRUE S_ValveStatus = C090 SafetyDemand = FALSE ResetRequest = FALSE ELSE S_ValveStatus = C490 SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 0000 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
83FF	Log on to valve group	Wait state to ensure that the SF_ValveGroupControl can be used only in combination with valve monitoring FBs. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 83FF SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

8401	Init	Block activation startup inhibit is active. Reset required. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8401 SafetyDemand = FALSE ResetRequest = TRUE Error = FALSE
8802	Wait for EDM1* / EDM2* (Output Disabled)	Wait for EDM signals to confirm the safe spool position of the valves. Timer starts when state is entered. Ready = TRUE S_SafeState = FALSE S_ValveOut = FALSE S_ValveStatus = 8802 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8804	Safe spool position	The EDM signals confirm that the valves are in the safe state (safe spool position) Ready = TRUE S_SafeState = TRUE S_ValveOut = FALSE S_ValveStatus = 8804 SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8010	Switched Spool Position*	The EDM signals confirm that the valves are in the switched spool position. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8010 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8020	Wait for NOT EDM1* / NOT EDM2*	Wait for EDM signals to confirm the switched spool position of the valves. Timer starts when state is entered. Ready = TRUE S_SafeState = FALSE S_ValveOut = TRUE S_ValveStatus = 8020 SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

4.6. ValveGroupControl

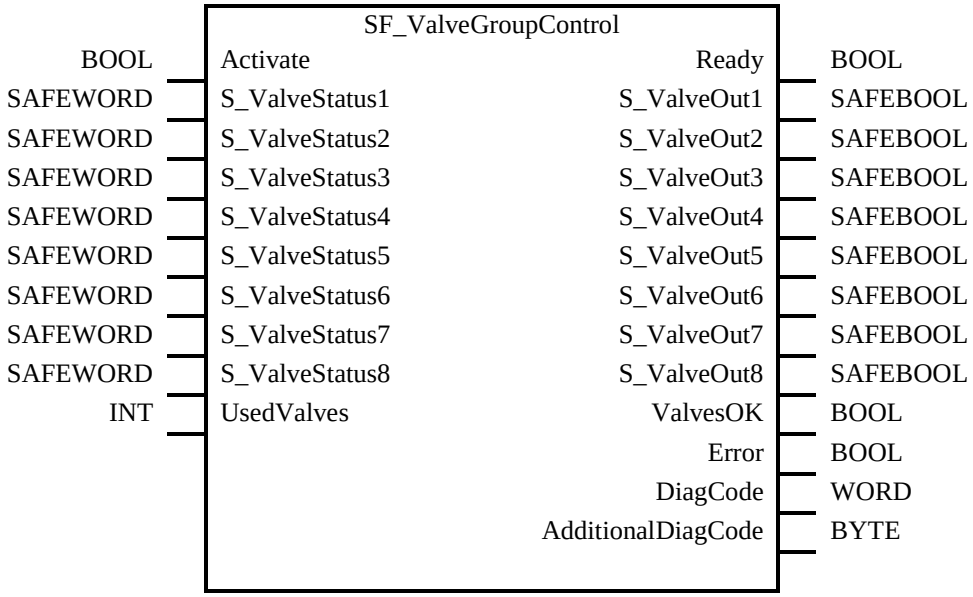
4.6.1. Applicable Safety Standards

Standards	Requirements
-	

4.6.2. Interface Description

FB Name	SF_ValveGroupControl		
This function block summarizes all the connected valves to a group. In case of an error of one valve all valves are switched off.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_ValveStatus1	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus2	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus3	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus4	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus5	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus6	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
S_ValveStatus7	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.

S_ValveStatus8	SAFEWORD	16#0000	Variable Signal of the preceding safety valve FB (e.g. SF_SingleValveMonitoring) which represents the state of the valve. Detailed information about the possible values see description at the valve FBs.
UsedValves	INT	0	Constant 0..8 Number of valves which are summarized to this group.
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_ValveOut1	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut2	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut3	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut4	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut5	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut6	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut7	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
S_ValveOut8	SAFEBOOL	FALSE	Safety related output to control the connected valve. FALSE: Disable connected valve TRUE: Enable connected valve
ValvesOK	BOOL	FALSE	Status of all summarized valves. This output can be used to inform the functional application about the status of all connected valves. FALSE: One or more valves are not in the safe state or an error occurred. TRUE: All valves are in the safe state, no error is present.
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
AdditionalDiagCode	BYTE	16#00	Additional diagnostic code for which valve has an error or is enabled. Every valve is represented by one bit.
Notes: -			



4.6.3. Functional Description

This function block is used for locking all valves which are connected to this FB in case of error. It is always used with the valve function blocks. The FB checks whether the connected valve function block has an error detected and accordingly switches **all** valves off. When an input is not connected, the corresponding output can not be set. Also he is not considered for the mutual locking.

An output will only be set if the function block is in state 0x8000 and the input from the corresponding valve function block is also in state 0x8000. The additional DiagCode shows which output is set.

The function block contains mechanisms to prevent systematic errors:

- In the valve function blocks a login algorithm is implemented, which is evaluated by this function block (83FF→8x01). This algorithm identifies the connected function block as a valve function block.
- The number of connected valves is compared with the predetermined number of valves (parameter “UsedValves”). The valves must be connected in ascending order without gaps.

If the valve function blocks are registered, the state is stored in the internal variable “InternalLoggedOn”. So the registration process is performed only in the event of a restart of the PLC.

State Diagram

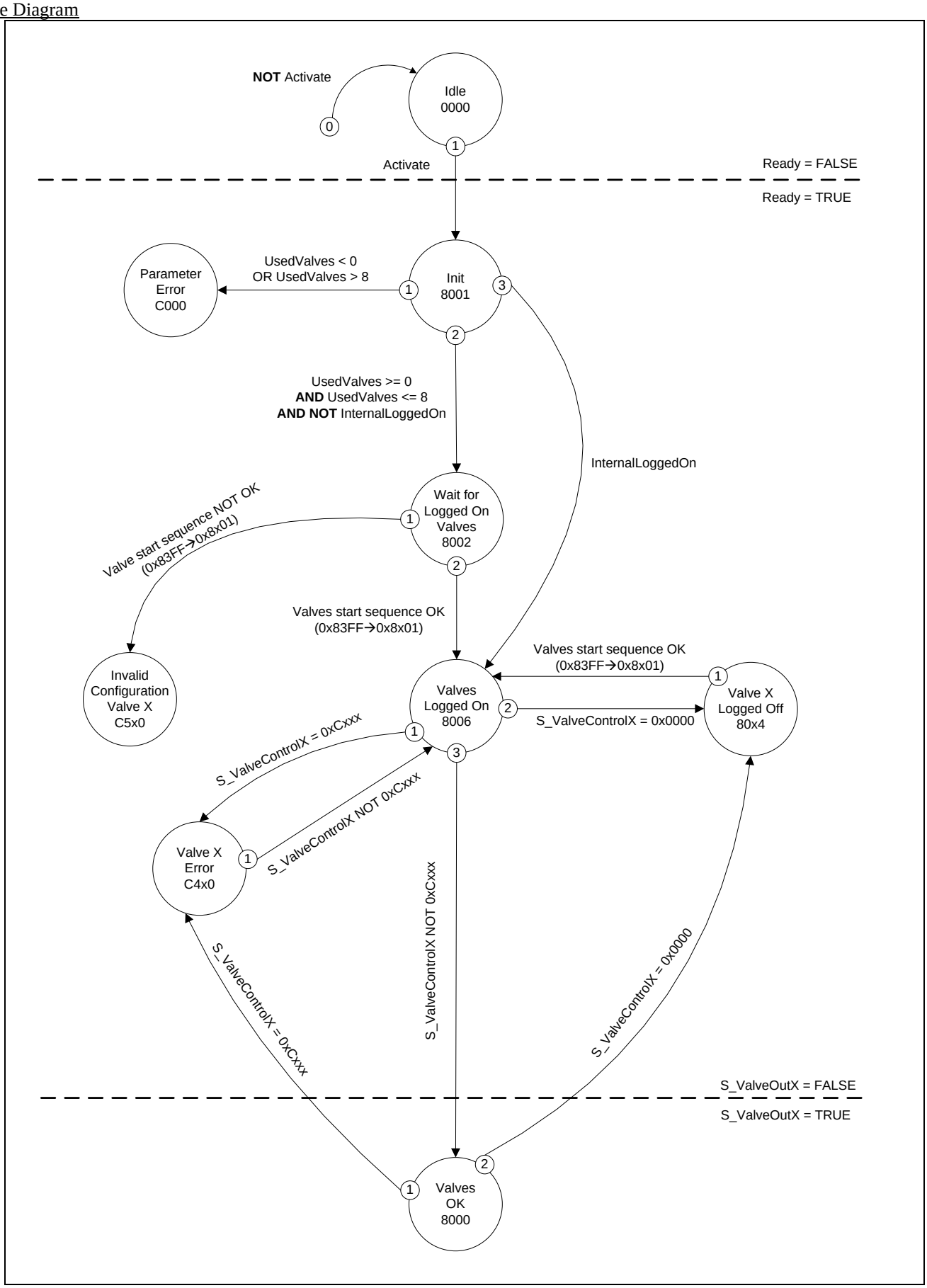


Figure 21: State diagram for SF_ValveGroupControl

Typical Timing Diagrams

Inputs	Start sequence				Normal operation with four valves							
Activate												
S_OpMode from the connected Valve FB												
S_ValveStatus1	0000	83FF	83FF	8401	8802	8804	8020	8010	8802	8804	8804	8804
S_OpMode from the connected Valve FB												
S_ValveStatus2	0000	83FF	83FF	8401	8802	8804	8020	8010	8802	8804	8804	8804
S_OpMode from the connected Valve FB												
S_ValveStatus3	0000	83FF	83FF	8401	8802	8804	8804	8804	8020	8010	8802	8804
S_OpMode from the connected Valve FB												
S_ValveStatus4	0000	83FF	83FF	8401	8802	8804	8804	8804	8020	8010	8802	8804
S_ValveStatus5	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus6	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus7	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus8	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
UsedValves	4	4	4	4	4	4	4	4	4	4	4	4
Outputs												
Ready												
S_ValveOut1												
S_ValveOut2												
S_ValveOut3												
S_ValveOut4												
S_ValveOut5												
S_ValveOut6												
S_ValveOut7												
S_ValveOut8												
ValvesOK												
Error												
AdditionalDiagCode	00	00	00	00	00	00	03	03	0C	0C	00	00
DiagCode	**0000**	**8001**	**8002**	**8002**	**8006**	**8000**	**8000**	**8000**	**8000**	**8000**	**8000**	**8000**

Figure 22: Timing diagram for SF_ValveGroupControl

Inputs	Start sequence				Errors from two connected valves							
Activate												
S_OpMode from the connected Valve FB												
S_ValveStatus1	0000	83FF	83FF	8401	8802	8804	8020	8010	8802	8804	8804	8804
S_OpMode from the connected Valve FB												
S_ValveStatus2	0000	83FF	83FF	8401	8802	8804	8020	8010	8802	8804	8804	8804
S_OpMode from the connected Valve FB												
S_ValveStatus3	0000	83FF	83FF	8401	8802	8804	8804	C040	C040	8802	8804	8804
S_OpMode from the connected Valve FB												
S_ValveStatus4	0000	83FF	83FF	8401	8802	8804	8804	8804	8020	C070	8802	8804
S_ValveStatus5	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus6	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus7	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
S_ValveStatus8	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
UsedValves	4	4	4	4	4	4	4	4	4	4	4	4
Outputs												
Ready												
S_ValveOut1												
S_ValveOut2												
S_ValveOut3												
S_ValveOut4												
S_ValveOut5												
S_ValveOut6												
S_ValveOut7												
S_ValveOut8												
ValvesOK												
Error												
AdditionalDiagCode	00	00	00	00	00	00	03	04	04	08	00	00
DiagCode	**0000**	**8001**	**8002**	**8002**	**8006**	**8000**	**8000**	**C430**	**C430**	**C440**	**8006**	**8000**

Figure 23: Second timing diagram for SF_ValveGroupControl

4.6.4. Error Detection

The function block detects an error on the parameter UsedValves if there is a constant connected that is not allowed.

4.6.5. Error Behavior

In the event of an error, the S_ValveOutX outputs are set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE. A restart is inhibited until the error conditions are cleared.
This function block has an additional diag code to inform which valve was not logged on correctly, has an error or is set.

4.6.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting
FB-specific error codes:		
C000	Parameter Error 0	Parameter UsedValves out of range. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 00
C001	Reset Error 0	Static Reset detected in State 8001 Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = TRUE AdditionalDiagCode = 00
C010	Reset Error 0	Static Reset detected in State C000 Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = TRUE AdditionalDiagCode = 00

C020	Reset Error 2	Static Reset detected in State 80x4 Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = TRUE AdditionalDiagCode = 00
C410	Valve 1 Error	Valve 1 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 01
C420	Valve 2 Error	Valve 2 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 02
C430	Valve 3 Error	Valve 3 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 04

C440	Valve 4 Error	Valve 4 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 08
C450	Valve 5 Error	Valve 5 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 10
C460	Valve 6 Error	Valve 6 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 20
C470	Valve 7 Error	Valve 7 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 40

C480	Valve 8 Error	Valve 8 has an error detected. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 80
C510	Invalid Configuration Valve 1	Valve 1 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 01
C520	Invalid Configuration Valve 2	Valve 2 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 02
C530	Invalid Configuration Valve 3	Valve 3 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 04

C540	Invalid Configuration Valve 4	Valve 4 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 08
C550	Invalid Configuration Valve 5	Valve 5 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 10
C560	Invalid Configuration Valve 6	Valve 6 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 20
C570	Invalid Configuration Valve 7	Valve 7 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 40

C580	Invalid Configuration Valve 8	Valve 8 is set as used but has not logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = TRUE AdditionalDiagCode = 80

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = FALSE Error = FALSE AdditionalDiagCode = 00
8001	Init	Function block has activated and initiated. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 00

8002	Wait for Logged On Valves	Wait for specific Code from the Valves to log on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = XX (depending on which valve is marked for use but not logged on yet)
8006	Valves Logged On	All defined Valves has logged on. Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 00 InternalLoggedOn = TRUE
8014	Valve 1 Logged Off	Valve 1 has logged off (S_ValveStatus1 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 01

8024	Valve 2 Logged Off	Valve 2 has logged off (S_ValveStatus2 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 02
8034	Valve 3 Logged Off	Valve 3 has logged off (S_ValveStatus3 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 04
8044	Valve 4 Logged Off	Valve 4 has logged off (S_ValveStatus4 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 08
8054	Valve 5 Logged Off	Valve 5 has logged off (S_ValveStatus5 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 10

8064	Valve 6 Logged Off	Valve 6 has logged off (S_ValveStatus6 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 20
8074	Valve 7 Logged Off	Valve 7 has logged off (S_ValveStatus7 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 40
8084	Valve 8 Logged Off	Valve 8 has logged off (S_ValveStatus8 = 0x0000). Ready = TRUE S_ValveOut1 = FALSE S_ValveOut2 = FALSE S_ValveOut3 = FALSE S_ValveOut4 = FALSE S_ValveOut5 = FALSE S_ValveOut6 = FALSE S_ValveOut7 = FALSE S_ValveOut8 = FALSE ValvesOK = Depending on the states of all relevant valves Error = FALSE AdditionalDiagCode = 80

8000	Valves OK	Outputs S_ValveOut1 … S_ValveOut8 are active depending on the state of the S_ValveStatus inputs. Ready = TRUE S_ValveOut1 = TRUE or FALSE S_ValveOut2 = TRUE or FALSE S_ValveOut3 = TRUE or FALSE S_ValveOut4 = TRUE or FALSE S_ValveOut5 = TRUE or FALSE S_ValveOut6 = TRUE or FALSE S_ValveOut7 = TRUE or FALSE S_ValveOut8 = TRUE or FALSE ValvesOK = TRUE Error = FALSE AdditionalDiagCode = XX (depending on which valve is in State 0x8000)

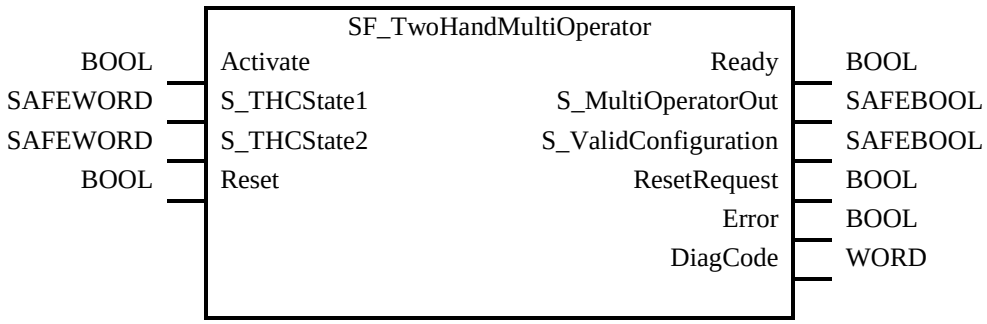
4.7. TwoHandMultiOperator

4.7.1. Applicable Safety Standards

Standards	Requirements
EN 692	5.3.14 Two-hand control devices shall comply with the following: b) the number of two-hand control devices in operation shall correspond to the number of operators indicated at the selection system; 5.4.5 Selector switches 5.4.5.4 If there is more than one operator at the press, the level of protection shall be the same for each operator. Where a number of two-hand control devices or de-connectable control stations are used, the press shall only be operable if the combination selected corresponds exactly to the combination physically connected to the press.

4.7.2. Interface Description

FB Name	**SF_TwoHandMultiOperator**		
This function block is used to control a press with two operator control stations. On each control stations is one two-hand control device.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_THCState1	SAFEWORD	16#0000	Variable Signal of the preceding safety FB "SF_TwoHandControlTypeIIIC" for the analysis of the first two-hand device.
S_THCState2	SAFEWORD	16#0000	Variable Signal of the preceding safety FB "SF_TwoHandControlTypeIIIC" for the analysis of the second two-hand device.
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_MultiOperatorOut	SAFEBOOL	FALSE	Safety related output. FALSE: no valid configuration or not all buttons pressed. TRUE: considering to a valid configuration, on one or both two-hand devices all buttons are pressed.
S_ValidConfiguration	SAFEBOOL	FALSE	Is provided as a static enable signal for further processing. FALSE: TRUE: The operator control stations are - deselected and unplugged or - selected and plugged
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: In case of more than two control stations to be controlled, more instances of this FB can be used			



4.7.3. Functional Description

This function block is used to control a press with two operator control stations. On each operator control stations is one two-hand control device. The FB is always used with the function block “SF_TwoHandCorntrolTypeIIIC”.

The operatorcontrol stations1 and 2 must always be placed on the same side of the press (front or back).In case of more than two control stations to be controlled, more instances of this FB have to be used.
This function block determines a valid configuration and the switching of the outputsfrom the two-hand function blocks from the parameters S_THCState1 and S_THCState2 of the “SF_TwoHandControlTypeIIIC” function blocks.

The output parameter “S_MultioperatorOut” is TRUE, if considering to a valid configuration, on one or both two-hand devices all buttons are pressed.
The output parameter „S_ValidConfiguration“ is TRUE if a valid confuguration isset.

Valid configurations are:

- Both operator control stations are selected and plugged
- One operator control station is selected an plugged and the other is deselected and unplugged
- Both operator control stations are deselected and unplugged. In this case the output “S_ValidConfiguration” is TRUE. The output “S_MultiOperatorOut” remains FALSE.

The function block contains a mechanism to prevent systematic errors:
In the TwoHandControlTypeIIICfunction block a login algorithm is implemented, which is evaluated by this function block (83FE→8001). This algorithm identifies the connected function block as a TwoHandControlTypeIIIC function block.
If the TwoHandControlTypeIIIC function blocks are registered, the state is stored in the internal variable “InternalLoggedOn”. So the registration process is performed only in the event of a restartof the PLC.

The function block has no time monitoring.

State Diagram

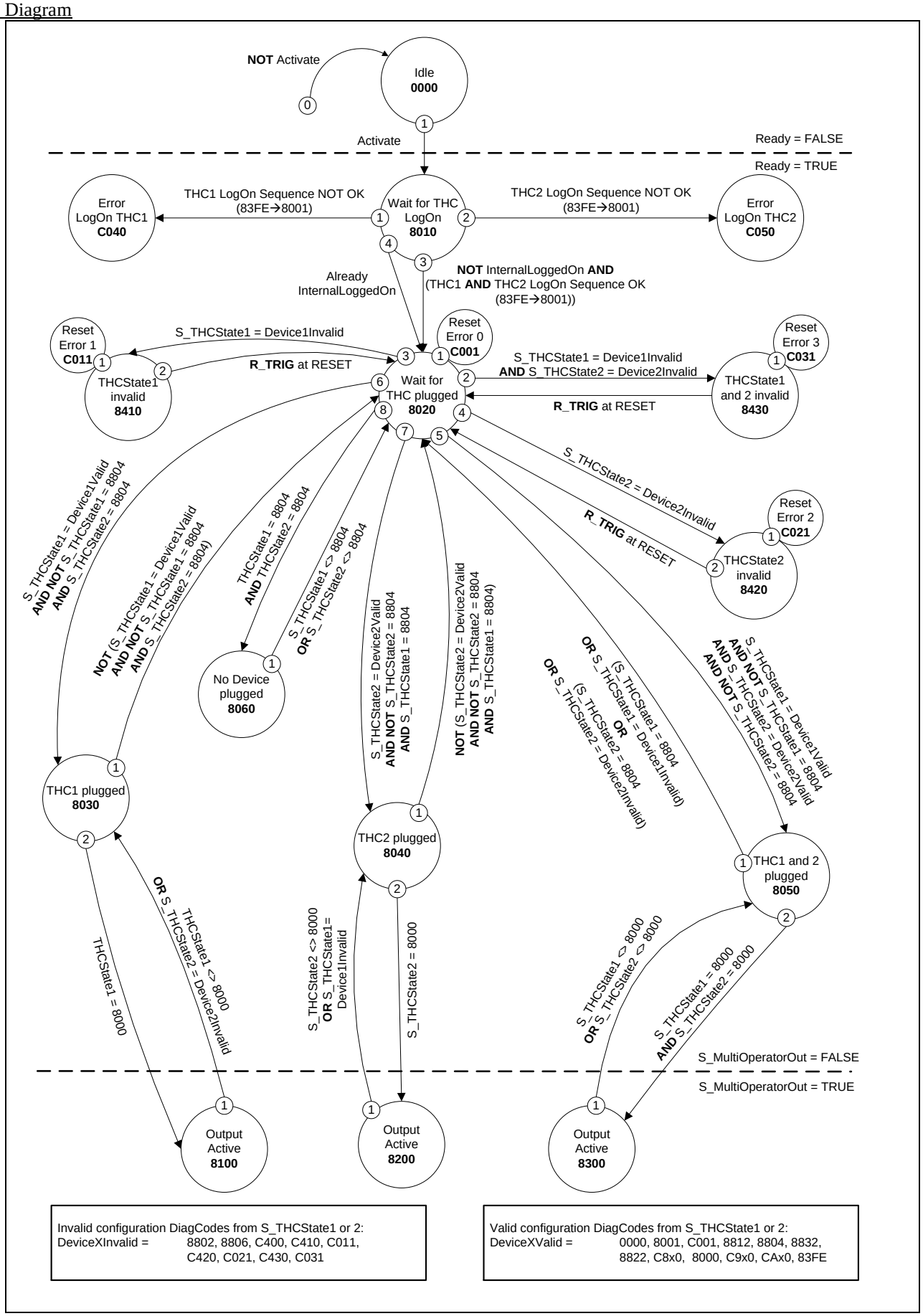


Figure 24: State diagram for SF_TwoHandMultiOperator

Typical Timing Diagrams

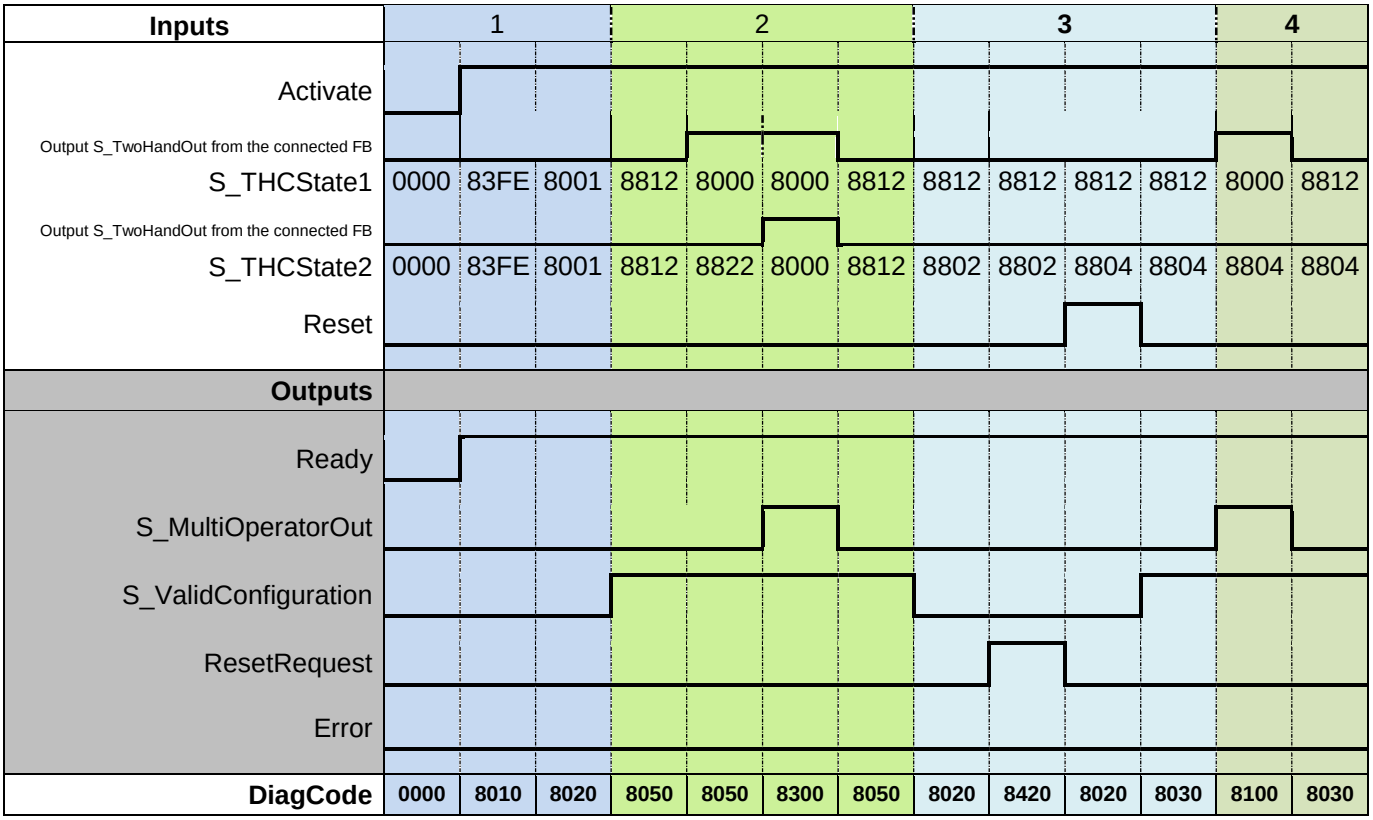


Figure 25: Timing diagram for SF_TwoHandMultiOperator

1: Start sequence
2: Normal operation with two control stations
3: Switching to one console
4: Normal operation with one console

4.7.4. Error Detection

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

4.7.5. Error Behavior

In the event of an error, the S_MultiOperatorOut output is set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE.
A restart is inhibited until the error conditions are cleared and the safe state is acknowledged with Reset by the operator.

4.7.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting

FB-specific error codes:

C001	Reset Error 0	Static Reset detected in State 8001 Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in State 8410 Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Static Reset detected in State 8420 Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE
C031	Reset Error 3	Static Reset detected in State 8430 Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE
C040	Error LogOn THC1	THC1 LogOn Sequence was not OK. (0000→83FE→8001) Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE
C050	Error LogOn THC2	THC2 LogOn Sequence was not OK. (0000→83FE→8001) Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = FALSE

8010	Wait for THC LogOn	Waiting for the LogOn sequence from THC1 and THC2. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = FALSE
8020	Wait for THC plugged	Function block has activated and initiated and is checking if a THC is plugged. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = FALSE Error = FALSE InternalLoggedOn = TRUE
8030	THC1 plugged	Only THC1 is valid and plugged and THC2 is valid and not plugged. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8040	THC2 plugged	Only THC2 is valid plugged and THC1 is valid and not plugged. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8050	THC1 and 2 plugged	THC1 and THC2 are valid and plugged. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8060	No Device plugged	No Device is plugged. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8410	THCState1 invalid	THCState1 has an invalid configuration. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = TRUE Error = FALSE
8420	THCState2 invalid	THCState2 has an invalid configuration. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = TRUE Error = FALSE

8430	THCState1 and 2 invalid	THCState1 and THCState2 have invalid configurations. Ready = TRUE S_MultiOperatorOut = FALSE S_ValidConfiguration = FALSE ResetRequest = TRUE Error = FALSE
8100	Output Active	Only THC1 is plugged and both buttons are pressed. THC2 is valid and unplugged. Ready = TRUE S_MultiOperatorOut = TRUE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8200	Output Active	Only THC2 is plugged and both buttons are pressed. THC1 is valid and unplugged. Ready = TRUE S_MultiOperatorOut = TRUE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE
8300	Output Active	THC1 and THC2 are plugged and all buttons are pressed. Ready = TRUE S_MultiOperatorOut = TRUE S_ValidConfiguration = TRUE ResetRequest = FALSE Error = FALSE

4.8. CamshaftMonitor

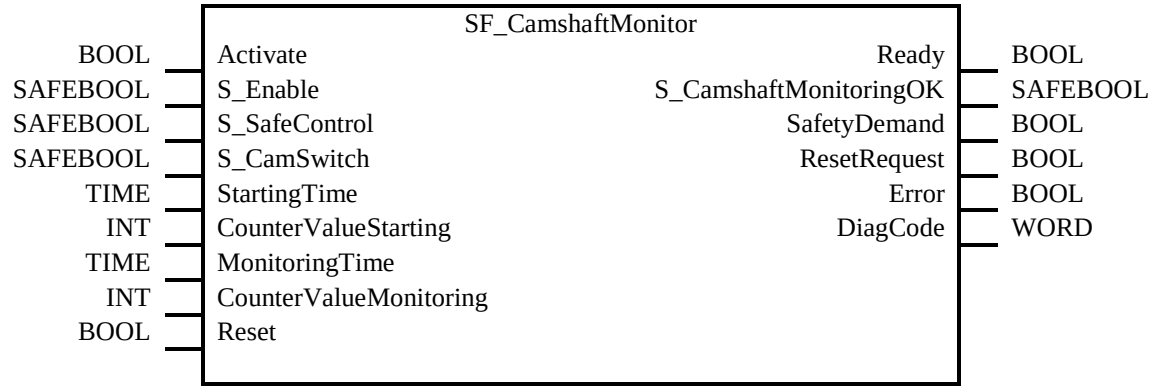
4.8.1. Applicable Safety Standards

Standards	Requirements
EN692:2005 - 5.4.2.4 c	5.4.2.4 Where the provision is necessary for overrun monitoring, this shall conform to the following requirements: a) manually fed presses fitted with protective devices of the type listed in 5.4.1.4 shall include overrun monitoring devices to ensure that, if the crankshaft overruns its normal stopping position by an amount specified by the manufacturer, maximum 15° and preferably 10°, a stopping signal shall be immediately initiated and no new cycle initiation shall be possible; b) it shall only be possible to restore further operation of the press by a restricted means, e.g. by tool, key or electronic password; c) where the cams to the overrun monitoring device are driven from a camshaft which is indirectly driven from a crankshaft, e.g. by a duplex chain drive between the camshaft and the crankshaft, indirect drive shall be monitored in such a way that, if it fails, a stopping signal shall be initiated and no new cycle initiation shall be possible until the fault is eliminated.

4.8.2. Interface Description

FB Name	**SF_CamshaftMonitor**		
This function block provides the camshaft monitoring functionality. There must be a number of signal changes in a specified time periode.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_Enable	SAFEBOOL	FALSE	Variable. Control signal to activate cam monitoring. FALSE: Camshaft monitoring not active. TRUE: Camshaft monitoring active.
S_SafeControl	SAFEBOOL	FALSE	Variable. Input to signal that the press is moving and the monitoring has to be performed. Usually connected to the control signal for the press safety value or the feedback signal from the press safety valve. FALSE: Press is not moving – monitoring not active. TRUE: Press is moving – monitoring active.
S_CamSwitch	SAFEBOOL	FALSE	Variable. Input of cam switch FALSE: Cam not active TRUE: Cam active
StartingTime	TIME	T#0ms	Constant. Range: 0 … x ms. Starting time of the machine. Within this time the value of CounterValueStarting must be reached.
CounterValueStarting	INT	INT#0	Constant. Range: 0 … 40. Number of signal changes that have to be reached during startup.
MonitoringTime	TIME	T#0ms	Constant. Range: 0 … x ms. MonitoringTime must be less than StartingTime. Monitoring time during cyclic operation. Within this time the value of CounterValueMonitoring must be reached.
CounterValueMonitoring	INT	INT#0	Constant. Range: 0 … 40. Number of signal changes that have to be reached during cyclic operation.
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_Camshaft MonitoringOK	SAFEBOOL	FALSE	Output indicationg the cam monitoring. FALSE: Camshaft monitoring not OK or signal error. TRUE: Camshaft monitoring sequence OK or not enabled.

SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: If multiple CAM switches are used multiple instances of this FB have to be used. The function block can handle up to 20 strokes (40 signal changes) – rising and falling edges of the S_CamSwitch are counted.			



4.8.3. Functional Description

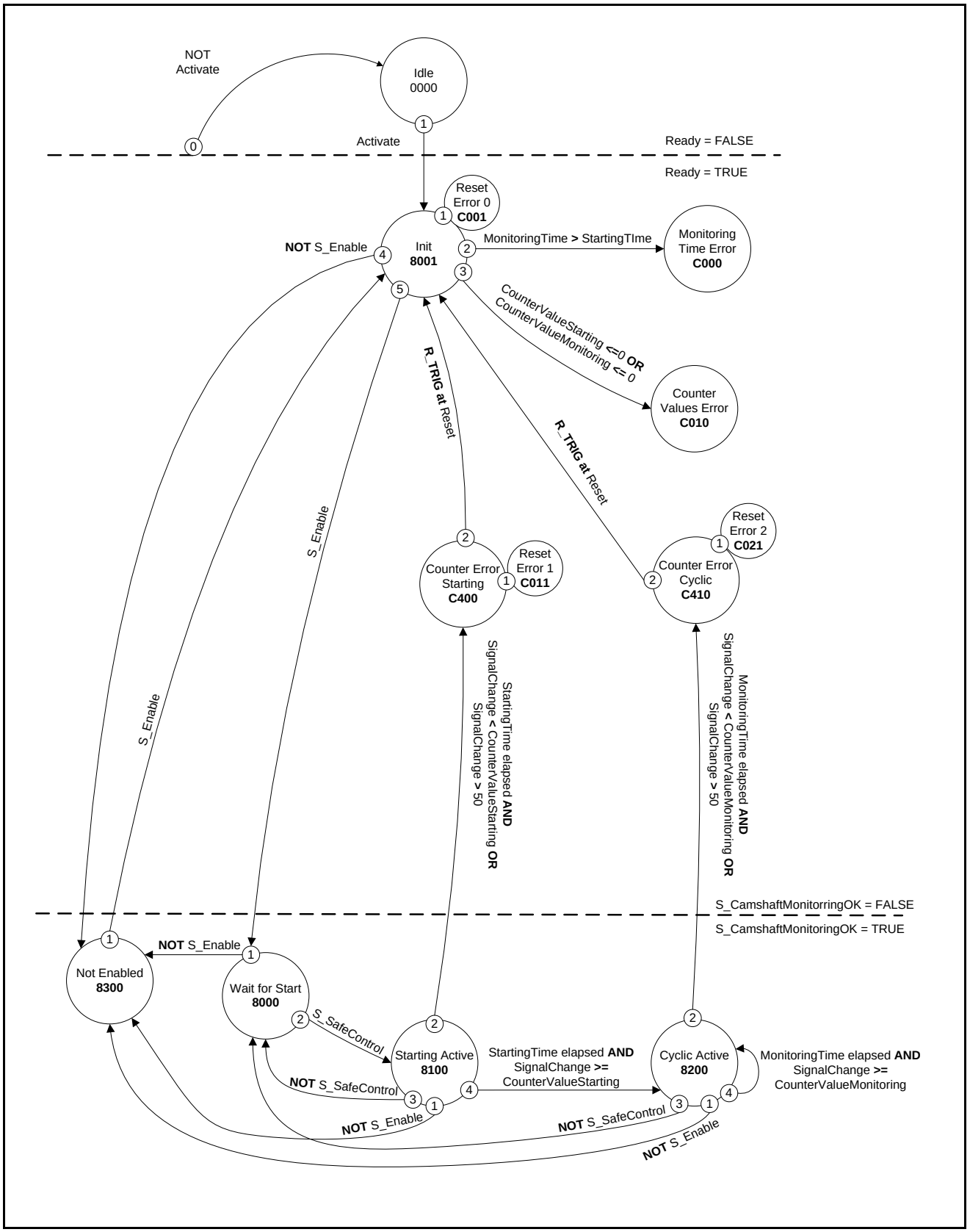
This function block provides the camshaft monitoring functionality. There must be a number of signal changes on the S_CamSwitch input in a specified time periode.

The monitoring function of the function block can be enabled or disabled via the S_Enable input.

Monitoring sequence:

1. Enable monitoring function
2. Press is moving – S_SafeControl input is TRUE
3. Timer with StartingTime is started – check number of signal changes
4. Timer with MonitoringTime is started – check number of signal changes
5. Repeat step 4

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 26: State diagram for SF_CamshaftMonitor

Typical Timing Diagrams

SafetyDemand and ResetRequest are not shown in the Timing Diagram
CounterValueStarting = 9
CounterValueMonitoring = 15

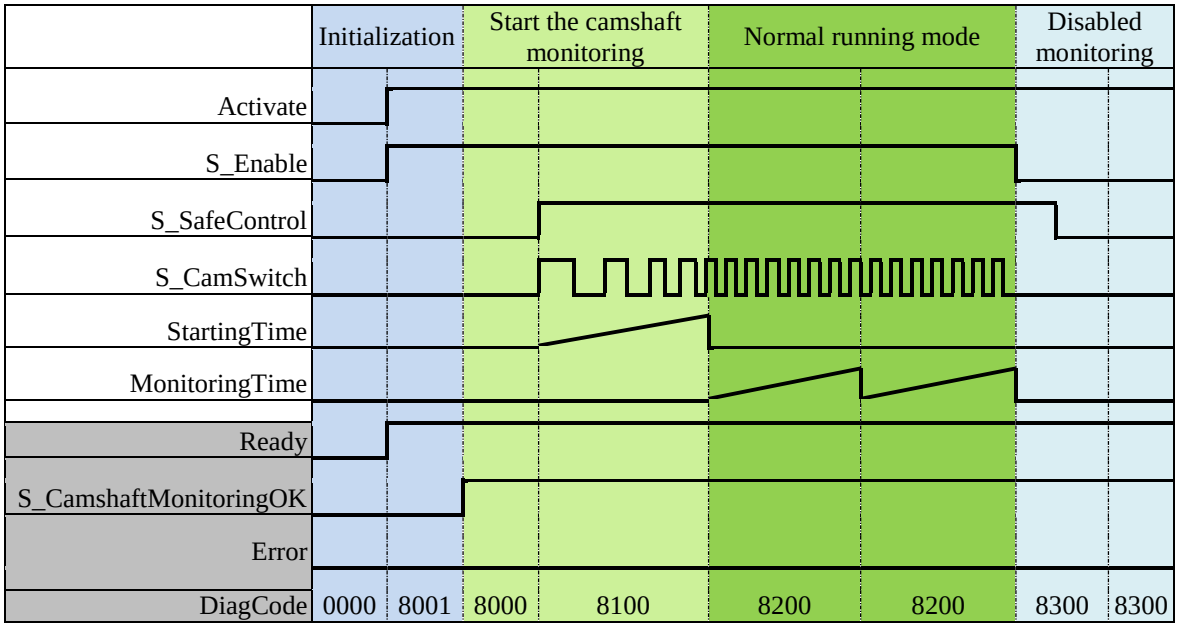


Figure 27: Timing diagram for SF_CamshaftMonitor

CounterValueStarting = 14
CounterValueMonitoring = 28

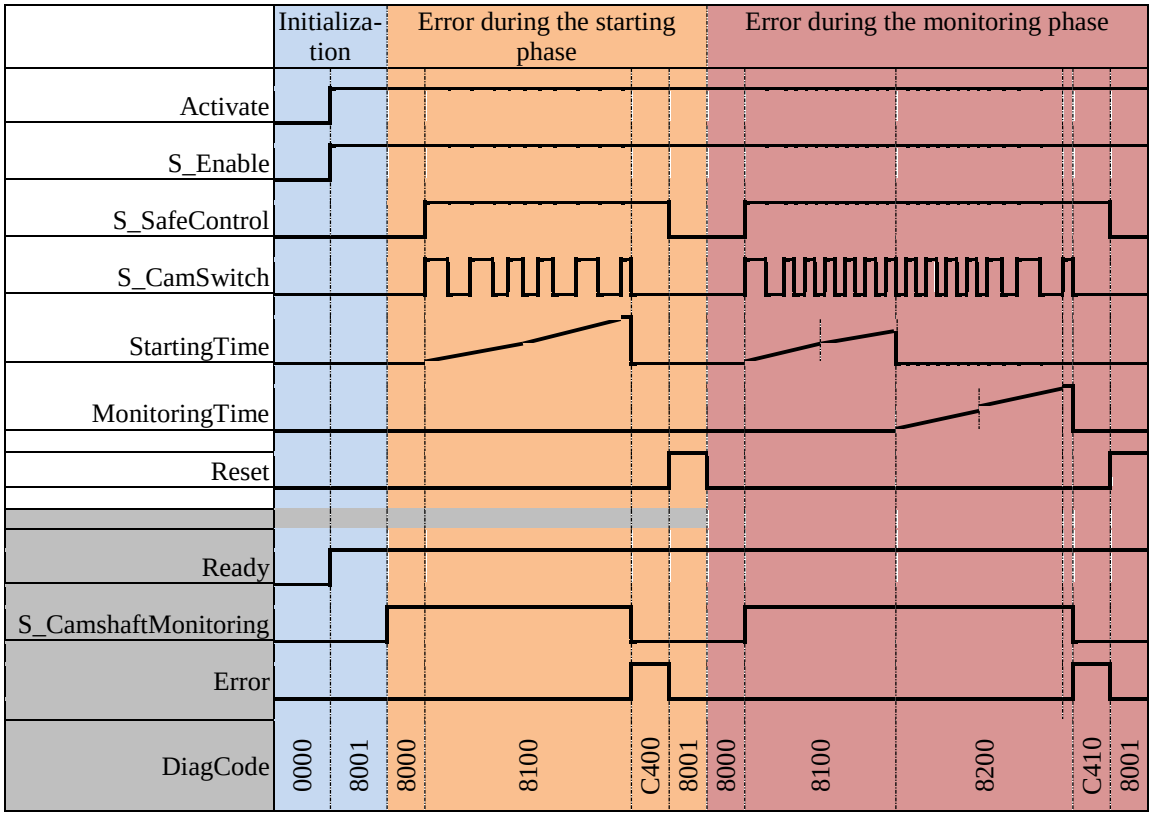


Figure 28: Second timing diagram for SF_CamshaftMonitor

4.8.4. Error Detection

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

4.8.5. Error Behavior

In the event of an error, the S_CamshaftMonitoringOK output is set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE.
A restart is inhibited until the error conditions are cleared and the safe state is acknowledged with Reset by the operator.

4.8.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting
FB-specific error codes, reset required		
C400	Counter Error Starting	Number of signal changes during StartingTime to low or signal changes greater than 50 Ready = TRUE S_CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C410	Counter Error Cyclic	Number of signal changes during MonitoringTime to low or signal changes greater than 50 Ready = TRUE S_CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
FB-specific error codes, no reset required		
C000	MonitoringTime Error	MonitoringTime greater than StartingTime Ready = TRUE S_CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C001	Reset Error 0	Static Reset detected in state 8001 Ready = TRUE S_ CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C010	Counter Values Error	Values for Counter Values not valid Ready = TRUE S_CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in state C400 Ready = TRUE S_ CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

C021	Reset Error 2	Static Reset detected in state C410 Ready = TRUE S_ CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_ CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8000	Wait for Start	Waiting for control signal input Ready = TRUE S_ CamshaftMonitoring = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8001	Init	The function block is activated and initiated Ready = TRUE S_ CamshaftMonitoring = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8100	Starting Active	Machine is in starting phase Ready = TRUE S_ CamshaftMonitoring = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8200	Cyclic Active	Machine is in cyclic operation Ready = TRUE S_ CamshaftMonitoring = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8300	Not Enabled	Function for Camshaft Monitoring is not enabled Ready = TRUE S_ CamshaftMonitoring = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

4.9. CycleControl

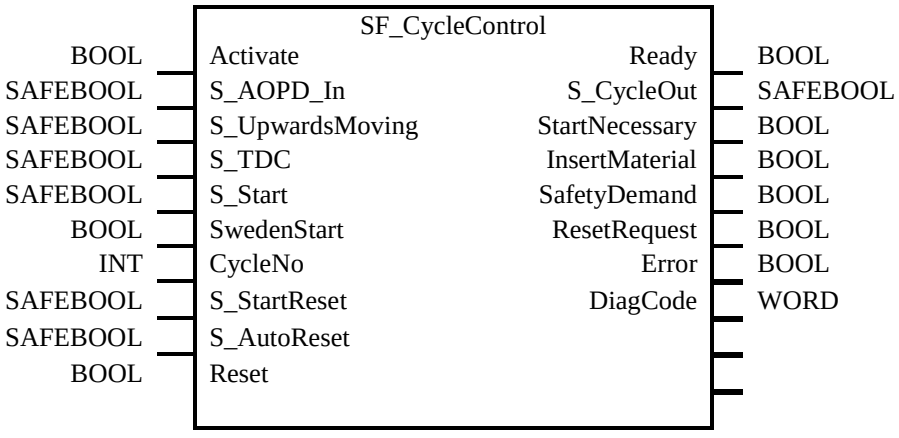
4.9.1. Applicable Safety standards

Standards	Requirements
DIN EN IEC 62046	Section 4.7.4 Reinitiation of cyclic operation by the protective equipment • One interruption (one cycle mode) if machine operation is reinitiated by activating and enabling the sensor part once • Two interruptions (two cycle mode) if machine operation is reinitiated by activating and releasing the sensor part twice in succession
DIN EN ISO 61496-1	Annex A8 ESPE for restarting a machine Annex A8.1 General • Activation and deactivation of the sensor part reinitiates a machine movement, referred to as one cycle mode • Double activation and deactivation of the sensor part reinitiates a machine movement, referred to as two cycle mode

4.9.2. Interface Description

FB name	SF_CycleControl		
The SF_CycleControl FB is required for controlling cycle mode (1 to n cycles) of an ESPE.			
VAR_INPUT			
Input name	*Data type*	*Initial value*	*Description, parameter values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_AOPD_In	SAFEBOOL	FALSE	Variable. OSSD signal from AOPD. FALSE: Protection field interrupted. TRUE: Protection field not interrupted.
S_UpwardsMoving	SAFEBOOL	FALSE	Variable or constant. Variable: Signal from the press which indicates the return stroke of the press (e.g., from the cam controller). This parameter must be connected to a relevant variable, if the press return stroke is not dangerous and light grid intervention during the press return stroke should not result in a shutdown. FALSE: press is not in return stroke mode TRUE: press is in return stroke mode Constant: If the press return stroke is dangerous and/or there must not be any intervention during the return stroke, the formal parameter must be connected to the FALSE literal.
S_TDC	SAFEBOOL	FALSE	Variable. Status signal of the machine (e.g., from the cam controller) for indicating normal position. FALSE: The machine is not in normal position. TRUE: The machine is in normal position.
S_Start	SAFEBOOL	FALSE	Variable. Edge-triggered signal to start the dangerous movement after irregular interruption of the ESPE and/or to initialize the process.

SwedenStart	BOOL	FALSE	Constant: Specification of the starting behavior (initialization of the function) FALSE: standard start TRUE: Sweden start
CycleNo	Int	INT#0	Constant: This value indicates the number/sum of cycles which causes the safe output to be set to TRUE for one operating cycle. A single cycle is defined by the signal transition from FALSE to TRUE at S_AOPD_In.
S_StartReset	SAFEBOOL	FALSE	See Section 3.3.1General rules
S_AutoReset	SAFEBOOL	FALSE	See Section 3.3.1General rules
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_CycleOut	SAFEBOOL	FALSE	Signal for controlling the cyclic movement.
StartNecessary	BOOL	FALSE	TRUE AND InsertMaterial TRUE: Diagnostic signal for indicating that the FB is in standard start initialization mode. For initializing cycle mode, input S_AOPD_In must be set according to the number at CycleNo (TRUE => FALSE => TRUE). TRUE AND InsertMaterial FALSE: Diagnostic signal for indicating that S_Start must be set from FALSE to TRUE in order to start cycle mode, if InsertMaterial is in the FALSE state.
InsertMaterial	BOOL	FALSE	Diagnostic signal for indicating that material needs to be inserted in order to start or maintain cycle mode. For initializing or maintaining cycle mode, input S_AOPD_In must be set according to the number at CycleNo (TRUE => FALSE => TRUE).
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	0	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: -			



4.9.3. Functional Description

SF_CycleControl is needed for applications which require cyclic process-related access to the danger zone and in which the safety device controls the movement.

Normal operation:
The first cycle/machine cycle must be started/initiated manually.
Two operating modes for starting the function (cycle function initialization) are available on the FB (Sweden start/normal start).

After the first cycle/machine cycle has been executed (TDC was left and reached again), the following machine cycles can be controlled by the safety device as long as MaxOperatorTime is not exceeded. If MaxOperatorTime is exceeded, cycle mode must be reinitiated.

In order for cycle mode to be started, the machine must be in normal position (TDC) (safe state).
Cycle mode initialization (normal start):
- The machine is in TDC (normal position).
- The safe output is FALSE.
- The ESPE must be interrupted while the machine is in the normal position (inserting workpieces).
  The ESPE is interrupted according to the specification at CycleNo (1 to n).
  The time between two interventions must not exceed 30 s.
- The safe output is still set to FALSE.
- The operator must set S_Start from FALSE to TRUE within 30 s.

Cycle mode initialization (Sweden start):
- The machine is in TDC (normal position).
- The safe output is FALSE.
- The operator sets S_Start from FALSE to TRUE (30 s time monitoring is started).
- The safe output is still set to FALSE.
- The ESPE is interrupted according to the specification at CycleNo (1 to n)
  (inserting workpieces).
  The time between two interventions must not exceed 30 s.
- When reaching the number of cycles specified at CycleNo, the machine cycle is started.

Action/reaction for cycle mode:
- The machine is in TDC (normal position) or in return stroke mode.
- The time that elapses after the last initialization must not exceed 30 s.
  - TDC:
    - The ESPE must be interrupted (inserting workpieces).
    - The ESPE is interrupted according to the specification at CycleNo (1 to n).
    - The time between two interventions must not exceed 30 s.
  - Return stroke:
    - The counter for the operating cycles counts the interruptions on S_AOPD_In (1 to n).
    - The ESPE is interrupted according to the specification at CycleNo (1 to n).
    - The time between two interventions must not exceed 30 s.
- When reaching the number of interruptions as specified at CycleNo, the machine cycle is started.
- The MaxOperatorTime for time monitoring of the material supply is stopped (T1).
- The safe output is set to TRUE. TDC is left.
- TDC is reached again (normal position). The safe output is set to FALSE again.

Irregular operation:
If the safe output is in the TRUE state and the ESPE is interrupted, the safe output is switched off and an error message is triggered under the following conditions:
- The plunger moves down.
- The return stroke is dangerous for the operator (S_UpwardsMoving).

To exit the error state a signal change from FALSE to TRUE is optionally required at Reset. The cycle must be reinitialized in any case.

State diagram

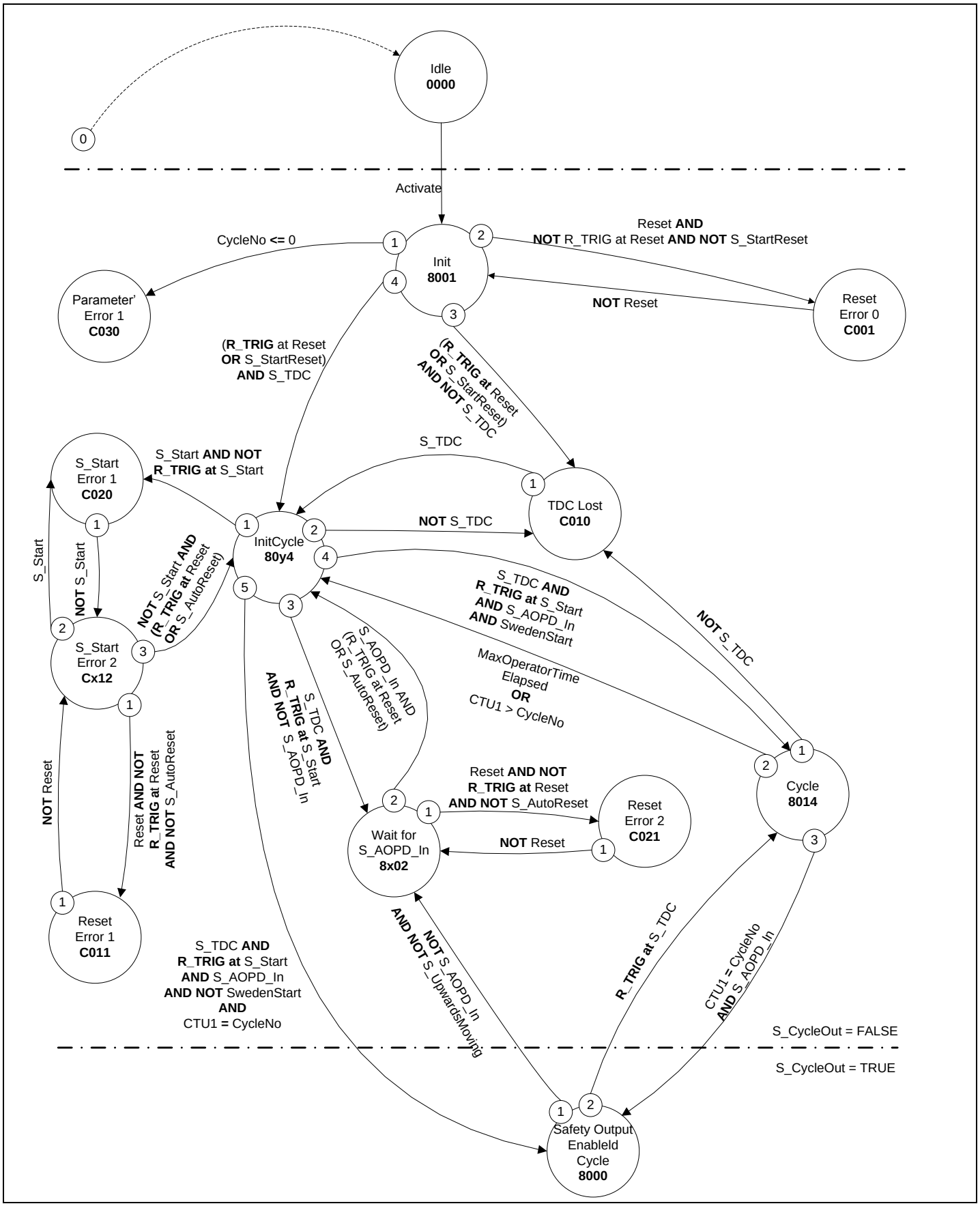


Figure 29: State diagram SF_CycleControl

Typcial tming diagrams
S_StartReset = FALSE; S_AutoReset = FALSE; SwedenStart = FALSE, CycleNo = INT#2.

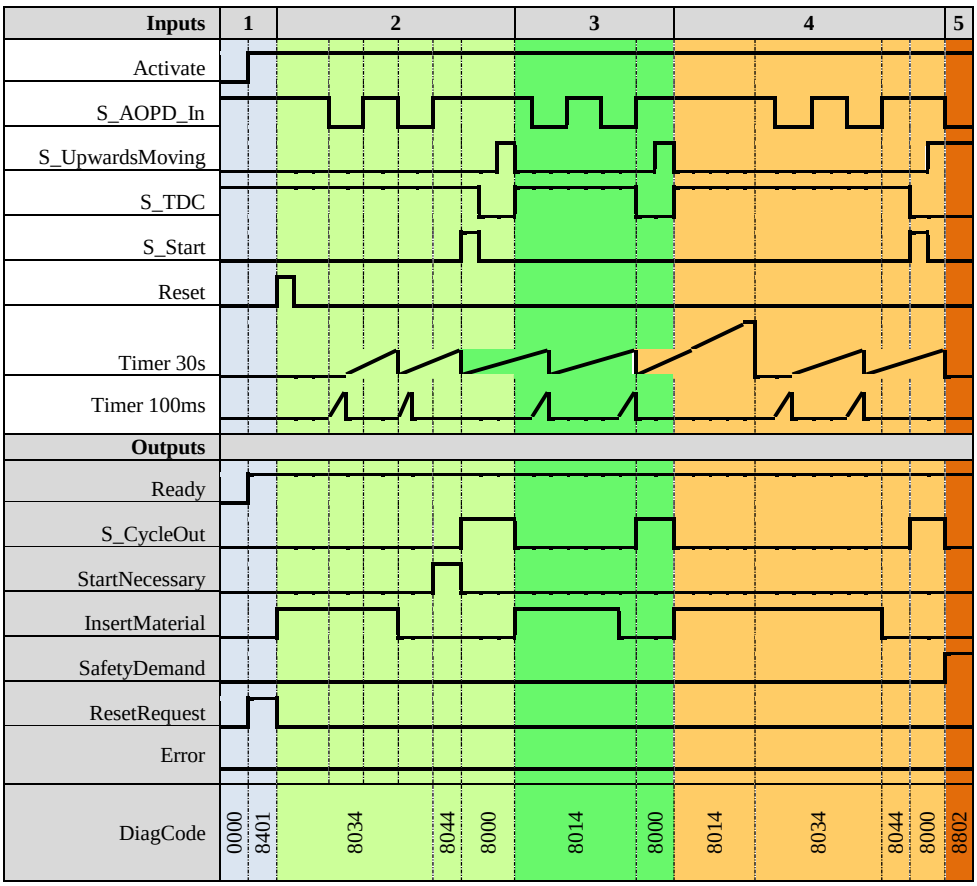


Figure 30: Timing diagram SF_CycleControl

1: Initialisation of the FB
2: Initialisation of the first cycle with standard start
3: Normal operation
4: MaxOperatorTime (30s) elapsed; a new initialization cycle is recommended
5: S_AOPD_In is lost between the press cycle

S_StartReset = FALSE; S_AutoReset = FALSE; SwedenStart = TRUE, CycleNo = INT#2:

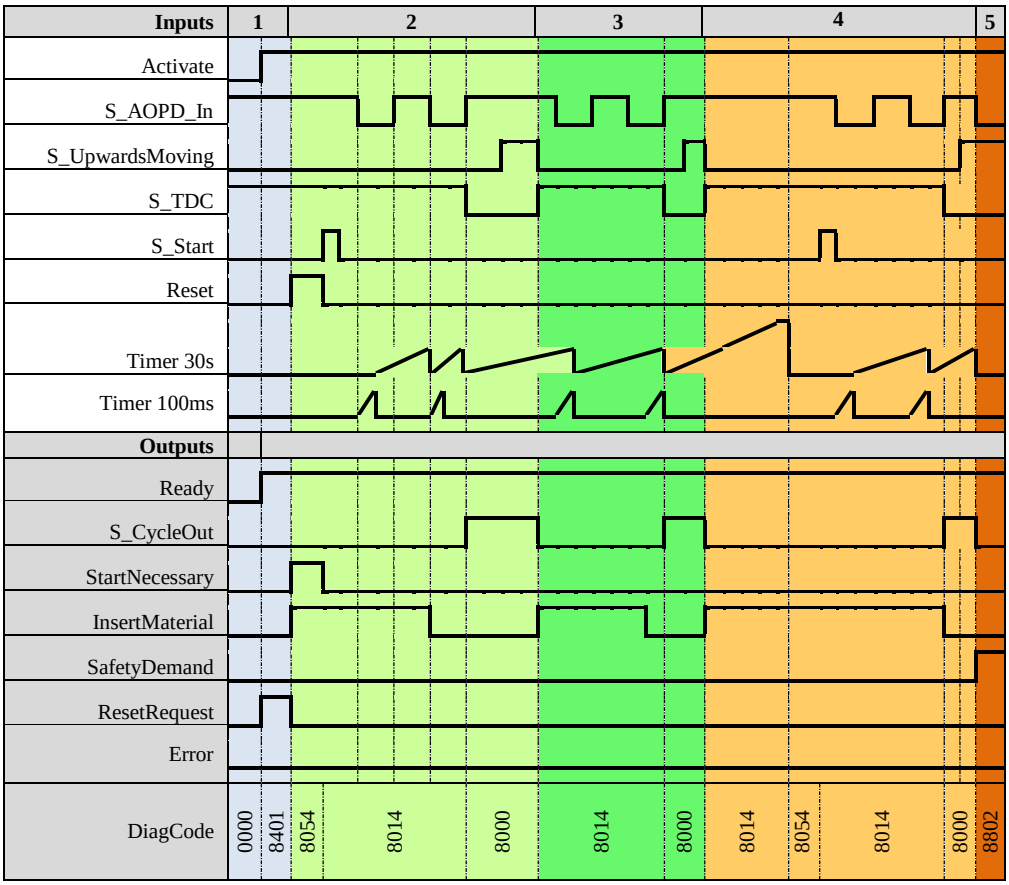


Figure 31: Timing diagram SF_CycleControl with Sweden start

1: Initialisation of the FB
2: Initialisation of the first cycle with Sweden start
3: Normal operation
4: MaxOperatorTime (30s) elapsed; a new initialization cycle is recommended
5: S_AOPD_In is lost between the press cycle

4.9.4. Function Block Specific Error and Status codes

Diagcode	State name	Output setting
FB specific error codes:		
C001	Reset Error 0	Static reset detected in state 8001 Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static reset detected in state C012 Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Static reset detected in state 8002 Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE
C010	TDC Lost	Normal position was left illegally Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE
C020	S_Start Error 1	Static S_Start detected in state 8004 S_Start is TRUE. S_Start must be set to FALSE. Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE

Cx12	S_Start Error 2	IF S_AutoReset = FALSE THEN x = 4 ELSE x= 0 Static S_Start detected in state 8004. S_Start ist FALSE. Reset must be set from FALSE to TRUE. Output signals if x = 0 (C012): Ready = TRUE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE Output signals if x = 4 (C412): Ready = TRUE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = TRUE Error = TRUE
C030	Parameter Error	The value at the CycleNo input parameter is not permissible (< = 0). The value of the constant must be > 0. Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state). Ready = FALSE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = FALSE

8000	Safety Output Enable Cycle	The function block accepts cycle control. Ready = TRUE S_CycleOut = TRUE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = FALSE **IF S_UpwardMoving = FALSE then:** The cycle counter (CTU1) will be reset. S_AOPD_In = FALSE is an invalid interruption. **IF S_UpwardMoving = TRUE then:** Every signal change from FALSE to TRUE at S_AOPD_In (FALSE) is counted as a cycle (CTU1). The LOW signal at S_AOPD_In for each interruption must last for 100ms, minimum. Shorter LOW signals at S_AOPD_In are not interruptions. The time between two interventions/interruptions or between cycle initialization and the subsequent intervention required must not exceed 30 s (T1). When exceeding 30 s between two interruptions, the counter (CTU1) is set to 0.
8x01	Init	IF S_AutoReset = FALSE THEN x = 4 ELSE x= 0 The function block is activated and initiated. R_TRIG at Reset is necessary. Counter (CTU1) and timer (T1) will be reset. Output signalsif x = 0 (8001): Ready = TRUE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = FALSE Output signalsif x = 4 (8401): Ready = TRUE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = TRUE Error = FALSE

8x02	Wait for S_AOPD_In	IF S_AutoReset = FALSE AND S_AOPD_In = TRUE THEN x = 4 ELSE x= 8 The protection field was or is interrupted irregularily. Counter (CTU1) and timer (T1) will be reset. Output signalsif x = 8 (8802): Ready = **TRUE** S_CycleOut = FALSE SafetyDemand = **TRUE** InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = FALSE Error = FALSE Output signalsif x=4 (8402): Ready = TRUE S_CycleOut = FALSE SafetyDemand = TRUE InsertMaterial = FALSE StartNecessary = FALSE ResetRequest = **TRUE** Error = FALSE

80y4	Init Cycle	Cycle control can be started. In the first control cycle, the cycle counter (CTU1) is reset when reaching state 8004. **IF SwedenStart = FALSE AND CTU < CycleNo then y = 3 (8034)**: Ready = **TRUE** S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = **TRUE** StartNecessary = FALSE ResetRequest = FALSE Error = FALSE Every signal change from FALSE to TRUE at S_AOPD_In (S_AOPD_In = FALSE) is counted as a cycle (CTU1). The LOW signal for each interruption must last for 100 ms, minimum. Shorter LOW signals are not interruptions. The time between two interventions/interruptions (S_AOPD_In = FALSE) must not exceed 30 s (T1). When exceeding 30 s between interruptions, the counter (CTU1) is set to 0. **IF SwedenStart = FALSE AND CTU = CycleNo then y = 4 (8044)**: Ready = **TRUE** S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = **TRUE** ResetRequest = FALSE Error = FALSE The time between the last interruption required for standard initialization at S_AOPD_In (CTU = CycleNo) and the signal change from FALSE to TRUE at S_Start must not exceed 30 s. When exceeding 30 s, the counter (CTU1) is set to 0 (DiagCode 8034). In this case, initialization must be performed again completely. Every signal change from FALSE to TRUE at S_AOPD_In (S_AOPD_In = FALSE) is counted as a cycle (CTU1). The LOW signal for each interruption must last for 100 ms, minimum. Shorter LOW signals are not interruptions. If CTU > CycleNo, then CTU is set to 0 and DiagCode is set to 8034. In this case, initialization must be performed again completely. **IF SwedenStart = TRUE then y = 5 (8054)**: Ready = **TRUE** S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = FALSE StartNecessary = **TRUE** ResetRequest = FALSE Error = FALSE The time between the signal change from FALSE to TRUE at S_Start and the first interruption at S_AOPD_In must not exceed 30 s.

8014	Cycle	Cycle control is active. The machine is in the starting position. Material must be inserted. Ready = TRUE S_CycleOut = FALSE SafetyDemand = FALSE InsertMaterial = TRUE StartNecessary = FALSE ResetRequest = FALSE Error = FALSE Every signal change from FALSE to TRUE at S_AOPD_In is counted as a cycle (CTU1). The LOW signal for each interruption must last for 100 ms, minimum. Shorter LOW signals are not interruptions. The time between two interventions/interruptions must not exceed 30 s (T1). When exceeding 30 s between interruptions cycle control must be reinitialized.

4.10. CAM Monitoring

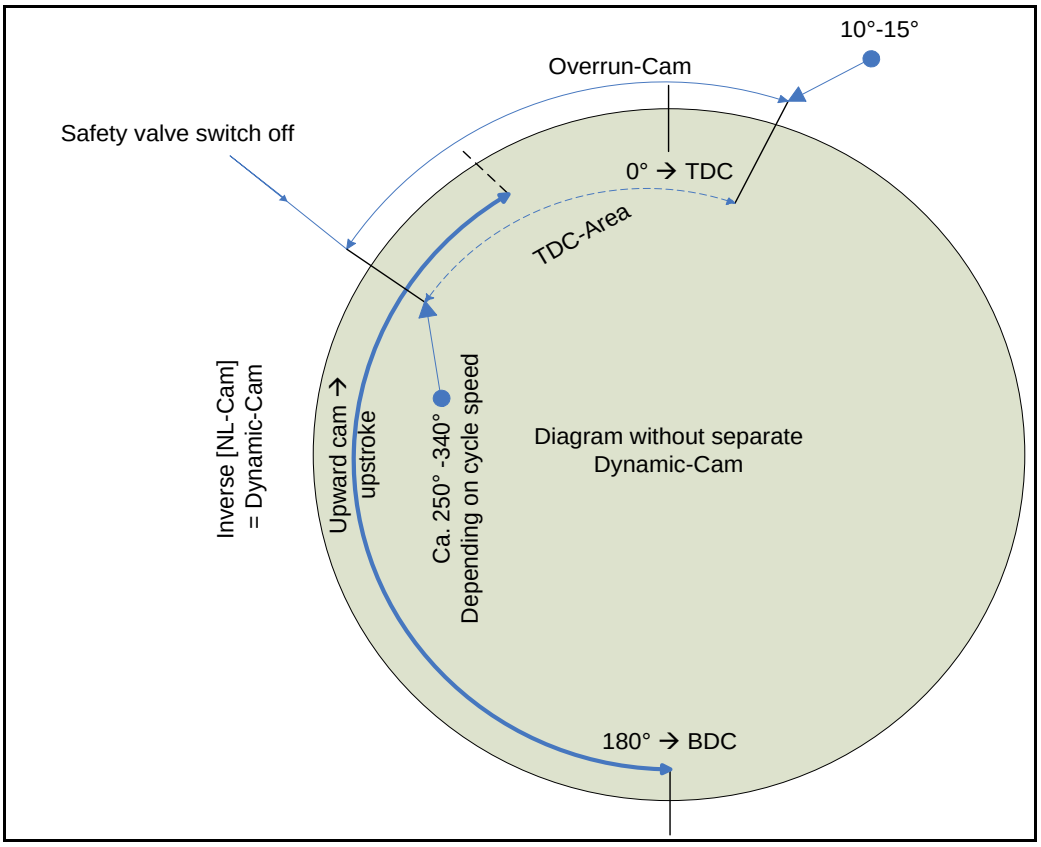


Figure 32: Example of SF_CamMonitoring

4.10.1. Applicable Safety Standards

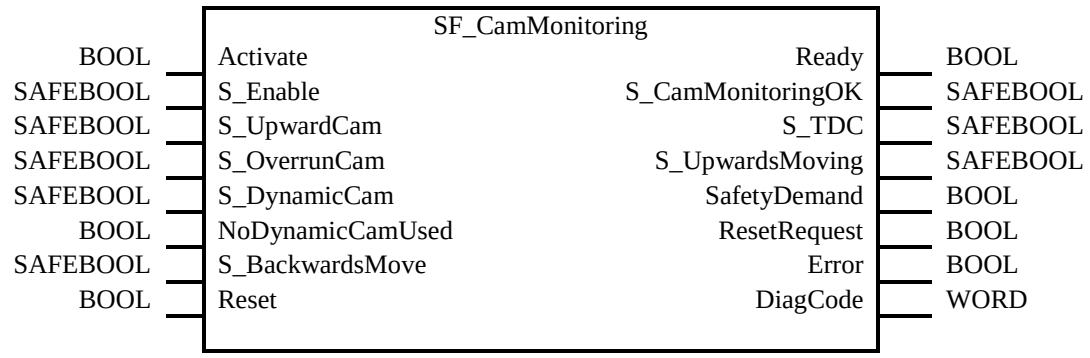
Standards	Requirements
EN692:2005 - 5.4.2.4 c	5.4.2.4 Where the provision is necessary for overrun monitoring, this shall conform to the following requirements: a) manually fed presses fitted with protective devices of the type listed in 5.4.1.4 shall include overrun monitoring devices to ensure that, if the crankshaft overruns its normal stopping position by an amount specified by the manufacturer, maximum 15° and preferably 10°, a stopping signal shall be immediately initiated and no new cycle initiation shall be possible; b) it shall only be possible to restore further operation of the press by a restricted means, e.g. by tool, key or electronic password; c) where the cams to the overrun monitoring device are driven from a camshaft which is indirectly driven from a crankshaft, e.g. by a duplex chain drive between the camshaft and the crankshaft, indirect drive shall be monitored in such a way that, if it fails, a stopping signal shall be initiated and no new cycle initiation shall be possible until the fault is eliminated.
EN692:2005 - 5.4.2.5	5.4.2.5 The cams and relevant switches for overrun monitoring, single cycle stop function and muting shall be linked one with another in a positive way so that the relative position between the cams, and the relative position between the switches cannot be altered. However, on presses fitted with a variable speed arrangement, the single cycle stop function switch may be capable of separate adjustment. All cams shall be positively secured to the shaft. All cams and relevant switches shall be in a locked enclosure
EN692:2005 - 5.4.2.6	5.4.2.6 In cases where the stroke length can be varied, adjustment of cams or cam operated switches used for cycle control shall be linked in a positive way so that their relative position cannot be altered, in order to minimize the probability of miss-setting by the user, e.g. to compensate for deterioration in braking performance which should be remedied by maintenance of the brake.

Standards	Requirements
EN692:2005 - 5.4.2.9 d	5.4.2.9 Cam discs shall be so applied that a wrong or unintended fitting cannot cause accidents, and: a) the limit switches and cam for control of the press shall be fixed to each other and adequately secured. Unsecured nuts or bolts are not permissible; b) the position of the limit switches and cam plates in relation to each other shall be marked; c) the possibility of re-adjustment of limit switches or cam shall be limited by fixed end stops, so that the muting time during the closing movement of the press cannot exceed the press over-all response time with any combination of speed and length of stroke; d) there shall be no possibility of damage to the impulse device when the press is reversed.
EN692:2005 - 5.4.7.2	5.4.7.2 The means of operation of the switch and the switch itself shall be designed to maintain their correct relationship to one another, the operating cam and particularly the stroke.
EN692:2005 - 5.4.7.3	5.4.7.3 The mechanism, e.g. cam and follower, shall be so designed that reverse rotation is possible without damage.
EN693:2001 - 5.4.5	5.4.5 Position switches (see 10.1.4 of EN 60204-1:1997) The means of operation of a position switch (see 3.16) and the switch itself shall be designed to maintain, after setting their position, their correct relationship to one another, the operating cam and particularly the stroke.

4.10.2. Interface Description

FB Name	**SF_CamMonitoring**		
This function block provides the cam monitoring functionality. There must be a defined sequence of the cam signals.			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_Enable	SAFEBOOL	FALSE	Variable. Control signal to activate cam monitoring. FALSE: Cam monitoring not active. TRUE: Cam monitoring active.
S_UpwardCam	SAFEBOOL	FALSE	4. Variable. Input for upward cam. FALSE: Cam active. TRUE: Cam not active.
S_OverrunCam	SAFEBOOL	FALSE	Variable. Input for overrun cam. FALSE: Cam active. TRUE: Cam not active.
S_DynamicCam	SAFEBOOL	FALSE	Variable. Input for dynamic cam. FALSE: Cam active. TRUE: Cam not active.
NoDynamicCamUsed	BOOL	FALSE	Constant. Input to disable dynamic cam usage. FALSE: Dynamic cam is used. TRUE: No dynamic cam is used.
S_BackwardsMove	SAFEBOOL	FALSE	Variable. Input for backward movement of cam. FALSE: Press is moving forwards. TRUE: Press is moving backwards.
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_CamMonitoringOK	SAFEBOOL	FALSE	Output indicationg the cam sequence. FALSE: Cam sequence not OK or signal error. TRUE: Cam sequence OK, no signal error or not enabled.
S_TDC	SAFEBOOL	FALSE	Output indicating the Top Dead Center point. FALSE: Press not in TDC. TRUE: Press is in TDC.

S_UpwardsMoving	SAFEBOOL	FALSE	Output indicating the upward movement of the press (from BDC to TDC). FALSE: Press is not moving upwards. TRUE: Press is moving upwards.
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules
Notes: If there is no dynamic cam used the input NoDynamicCamUsed must be set to TRUE.			



4.10.3. Functional Description

This function block provides the cam monitoring functionality. There must be a defined signal sequence on the inputs and the function block provides the information if the press is in the Top Dead Center (TDC) point – S_TDC - or is moving upwards – S_UpwardsMoving.
It is possible to work with 2 cams (Upward and Overrun) or 3 cams (Upward, Overrun and Dynamic).

The monitoring function of the function block can be enabled or disabled via the S_Enable input.

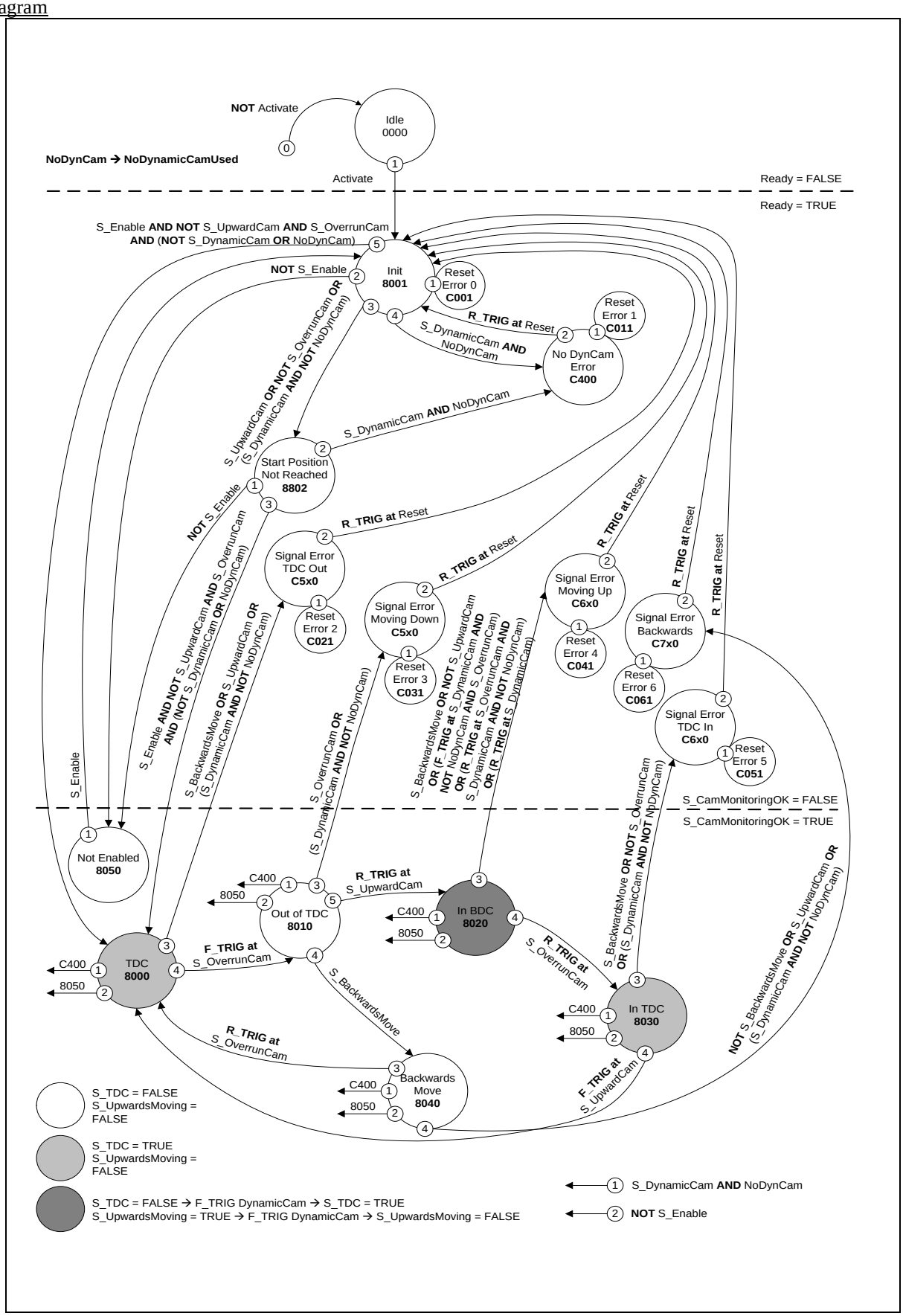
Press movement sequence

1. Enable monitoring function
2. Press is in TDC area → 8000
   a. S_UpwardCam = FALSE
   b. S_OverrunCam = TRUE
   c. S_DynamicCam = FALSE
   d. S_TDC = TRUE
   e. S_UpwardsMoving = FALSE
3. End of Overrun cam
   a. Press is moving out of TDC area
   b. F_TRIG @ S_OverrunCam → 8010
   c. S_UpwardCam = FALSE
   d. S_OverrunCam = FALSE
   e. S_DynamicCam = FALSE
   f. S_TDC = FALSE
   g. S_UpwardsMoving = FALSE
4. Start of Upward cam
   a. Press is in BDC area
   b. R_TRIG @ S_UpwardCam → 8020
   c. S_UpwardCam = TRUE
   d. S_OverrunCam = FALSE
   e. S_DynamicCam = TRUE
   f. S_TDC = FALSE
   g. S_UpwardsMoving = TRUE
5. Start of Overrun cam
   a. R_TRIG @ S_OverrunCam → 8030
   b. S_UpwardCam = TRUE
   c. S_OverrunCam = TRUE
   d. S_DynamicCam = FALSE

e. S_TDC = TRUE → F_TRIG @ S_DynamicCam
f. S_UpwardsMoving = FALSE → F_TRIG @ S_DynamicCam

6. End of Upward cam
   a. Press is in TDC area
   b. F_TRIG @ S_UpwardCam → 8000
   c. S_UpwardCam = FALSE
   d. S_OverrunCam = TRUE
   e. S_DynamicCam = FALSE
   f. S_TDC = TRUE
   g. S_UpwardsMoving = FALSE
7. Backwards Move
   a. Signal to move backwards
   b. S_BackwardsMove = TRUE → 8040
   c. S_Upwardcam = FALSE
   d. S_OverrunCam = FALSE
   e. S_DynamicCam = FALSE
   f. S_TDC = FALSE
   g. S_ UpwardsMoving = FALSE
   h. R_TRIG @ S_OverrunCam → 8000

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 33: State diagram for SF_CamMonitoring

Typical Timing Diagrams

S_DynamicCam is used

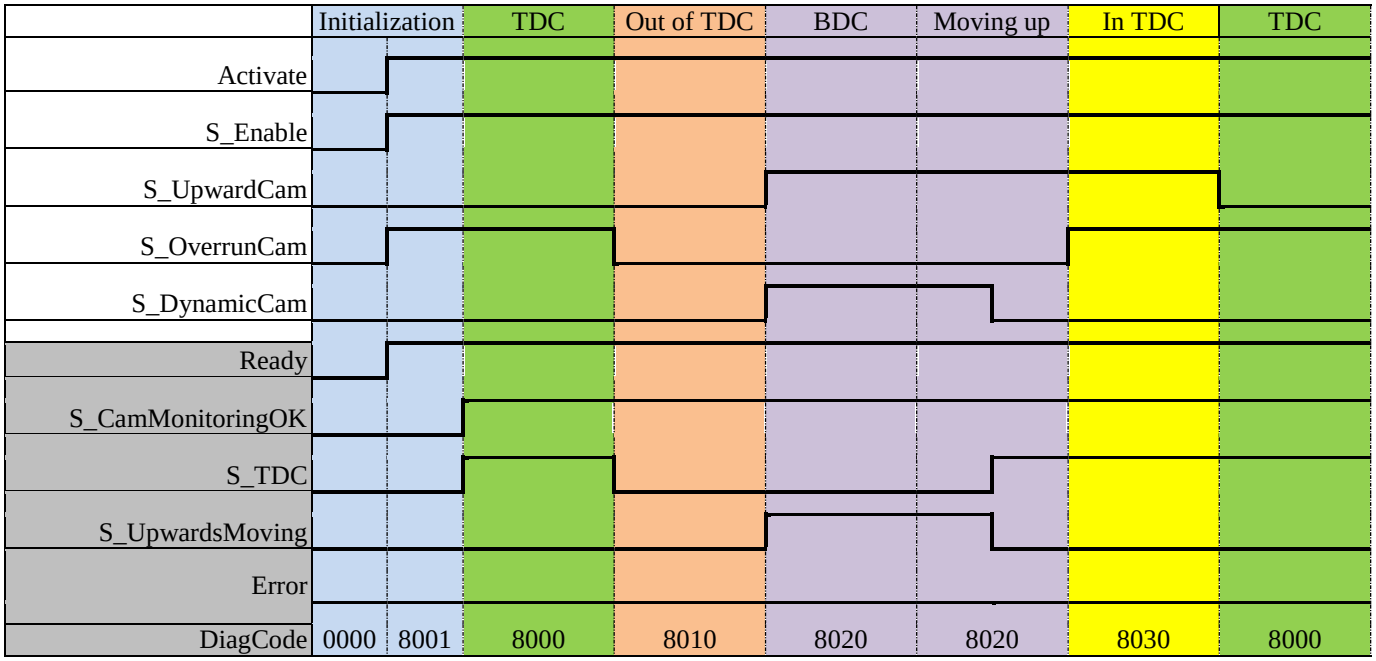


Figure 34: Timing diagram for SF_CamMonitoring

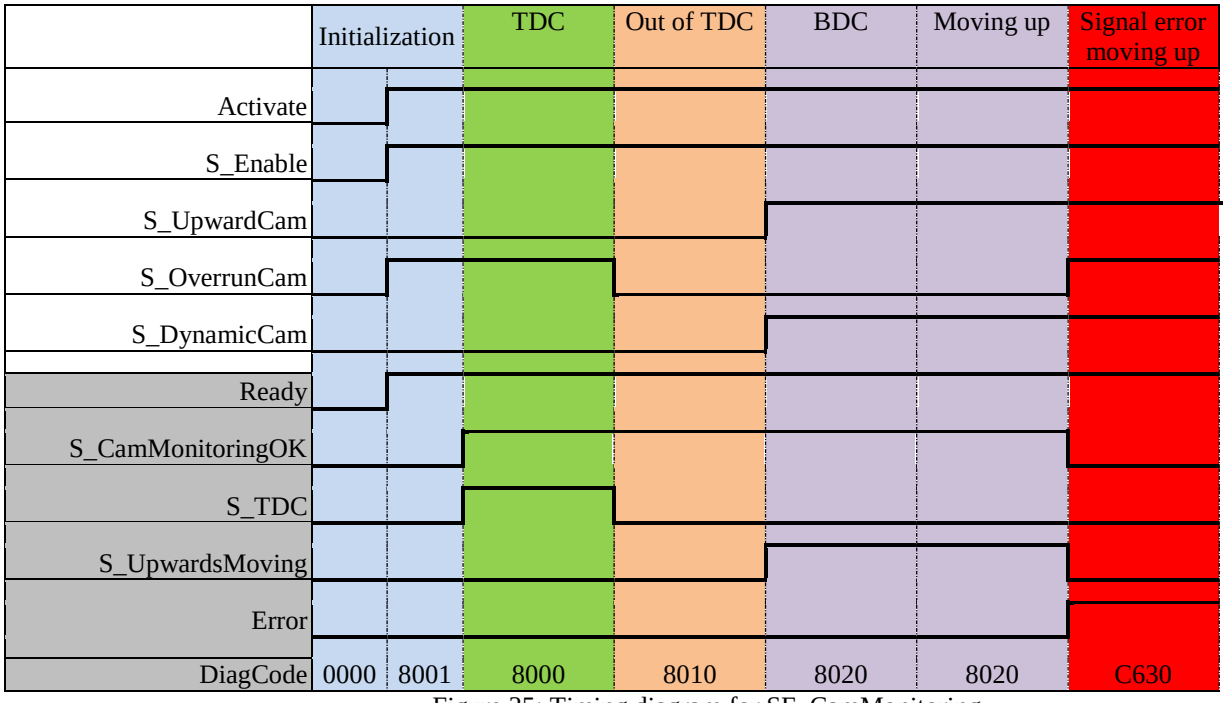


Figure 35: Timing diagram for SF_CamMonitoring

4.10.4. Error Detection

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

4.10.5. Error Behavior

In the event of an error, the S_CamMonitoringOK output is set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE.
A restart is inhibited until the error conditions are cleared and the safe state is acknowledged with Reset by the operator.

4.10.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting
FB-specific error codes, reset required		
C400	No DynCam Error	Configuration für Dynamic Cam is not correct Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C5x0	Signal Error TDC Out	Wrong Cam signal combination in TDC If S_BackwardsMove then x = 0 If S_UpwardCam then x = 1 If (S_DynamicCam AND NOT NoDynCam) then x = 2 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C5x0	Signal Error Moving Down	Wrong Cam signal combination when moving down If S_OverrunCam then x = 3 If (S_DynamicCam AND NOT NoDynCam) then x= 4 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

C6x0	Signal Erro Moving Up	Wrong Cam signal combination when moving up
		
		If S_BackwardsMove then x = 0
		If NOT S_UpwardCam then x = 1
		If (F_TRIG at S_DynamicCam AND NOT NoDynCam AND S_OverrunCam) then x = 2
		If (R_TRIG at S_OverrrunCam AND S_DynamicCam AND NOT NoDynCam) then x = 3
		
		Ready = TRUE
		S_CamMonitoringOK = FALSE
		S_TDC = FALSE
		S_UpwardsMoving = FALSE
		SafetyDemand = FALSE
		ResetRequest = TRUE
		Error = TRUE

C6x0	Signal Error TDC In	Wrong Cam signal combination when entering TDC If S_BackwardsMove then x = 4 If NOT S_OverrunCam then x = 5 If (S_DynamicCam AND NOT NoDynCam) then x = 6 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C7x0	Signal Error Backwards	Wrong Cam signal combination when moving backwards If NOT S_BackwardsMove then x = 0 If S_UpwardCam then x = 1 If (S_DynamicCam AND NOT NoDynCam) then x = 2 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific error codes, no reset required

C001	Reset Error 0	Static Reset detected in State 8001 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in State C400 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Static Reset detected in State C5x0 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

C031	Reset Error 3	Static Reset detected in State C5x0 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C041	Reset Error 4	Static Reset detected in State C6x0 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C051	Reset Error 5	Static Reset detected in State C6x0 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C061	Reset Error 6	Static Reset detected in State C70 Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial State). Ready = FALSE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8000	TDC	Press is in TDC area Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = TRUE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

8001	Init	The function block is activated and initiated Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8010	Out of TDC	Press is moving down Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8020	In BDC	Press is moving up Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = FALSE S_UpwardsMoving = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE S_TDC is FALSE and S_UpwareMoving is TRUE as long as no F_TRIG @ S_Dynamic is detected
8030	In TDC	Press is in TDC area Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = TRUE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8040	Backwards Move	Press is moving backwards Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8050	Not Enable	Function is not enables Ready = TRUE S_CamMonitoringOK = TRUE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

8802	Start Position Not Reached	Cam Signals for starting not OK Ready = TRUE S_CamMonitoringOK = FALSE S_TDC = FALSE S_UpwardsMoving = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE

4.11. TwoHandControlTypeIIIC

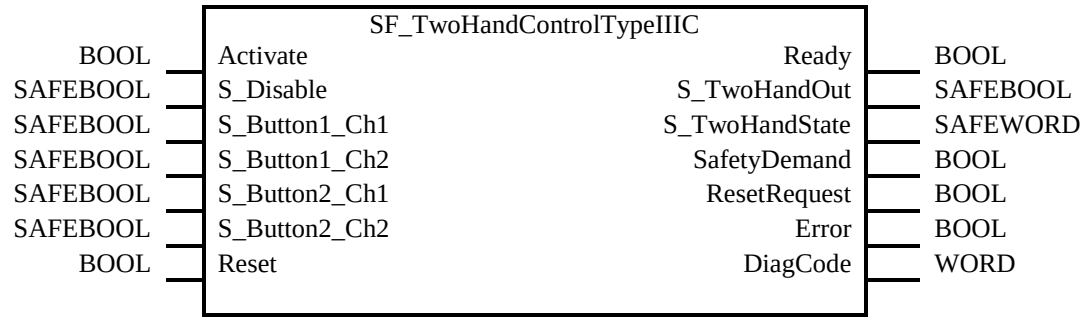
4.11.1. Applicable Safety Standards

Standards	Requirements
EN 574: 1996	Clause 4, Table 1, Type III C. 5.1 Use of both hands / simultaneous actuation. 5.2 Relationship between output signal and input signals. 5.3 Completion of the output signal. 5.6 Reinitiation of the output signal. 5.7 Synchronous actuation. 6.3 Use of DIN EN 954-1 category 3. (Can only be realized by NO and NC switches together with antivalent processing) 6.4 Use of DIN EN 954-1 category 4. (Can only be realized by NO and NC switches together with antivalent processing)
ISO 12100-2: 2003	4.11.4: Restart following power failure/spontaneous restart

4.11.2. Interface Description

FB Name	**SF_TwoHandControlTypeIIIC**		
This function block provides the pluggable two-hand control functionality (see EN 574, Section 4 Type III. Fixed specified time difference is 500 ms).			
VAR_INPUT			
Name	*Data Type*	*Initial Value*	*Description, Parameter Values*
Activate	BOOL	FALSE	See Section 3.3.1General rules
S_Disable	SAFEBOOL	FALSE	Variable. Signal input FALSE: Device is enabled TRUE: Device is disabled
S_Button1_Ch1	SAFEBOOL	FALSE	Variable. Input channel 1 (NC) of button 1 – Normally Closed. FALSE: NC contact open. TRUE: NC contact closed.
S_Button1_Ch2	SAFEBOOL	FALSE	Variable. Input channel 2 (NO) of button 1 – Normally Open. FALSE: NO contact open. TRUE: NO contact closed.
S_Button2_Ch1	SAFEBOOL	FALSE	Variable. Input channel 1 (NC) of button 2 – Normally Closed. FALSE: NC contact open. TRUE: NC contact closed.
S_Button2_Ch2	SAFEBOOL	FALSE	Variable. Input channel 2 (NO) of button 2 – Normally Open. FALSE: NO contact open. TRUE: NO contact closed.
Reset	BOOL	FALSE	See Section 3.3.1General rules
VAR_OUTPUT			
Ready	BOOL	FALSE	See Section 3.3.1General rules
S_TwoHandOut	SAFEBOOL	FALSE	Safety related output signal. FALSE: No correct two hand operation. TRUE: S_Button1 and S_Button2 inputs changed from FALSE to TRUE within 500 ms and no error occurred. The two hand operation has been performed correctly.
S_TwoHandState	SAFEWORD	16#0000	State of the actual device. Should be connected to the input of the SF_TwoHandMultiOperator function block. Represents the actual state of the device – plugged / pressed / unplugged / error.
SafetyDemand	BOOL	FALSE	See Section 3.3.1General rules
ResetRequest	BOOL	FALSE	See Section 3.3.1General rules
Error	BOOL	FALSE	See Section 3.3.1General rules
DiagCode	WORD	16#0000	See Section 3.3.1General rules

Notes: Simultaneous actuation (in 500ms both buttons) is checked inside the function block.



4.11.3. Functional Description

This function block provides the pluggable two-hand control functionality according to EN 574, Section 4 Type III C. If S_Button1 and S_Button2 are set to TRUE within 500 ms, then the S_TwoHandOut output is also set to TRUE. The FB also controls the release of both buttons before setting the output S_TwoHandOut again to TRUE.
With the input S_Disable it is possible to unplug (switch off) the device. If S_Disable is set to TRUE and the signals from all the buttons must be FALSE. In this case the output S_TwoHandOut is never set to TRUE.

To unplug the device without any error message or acknowledgement the following sequence must be maintained:

1. Device is plugged – buttons are not pressed
2. Set S_Disable to TRUE
3. Unplug device

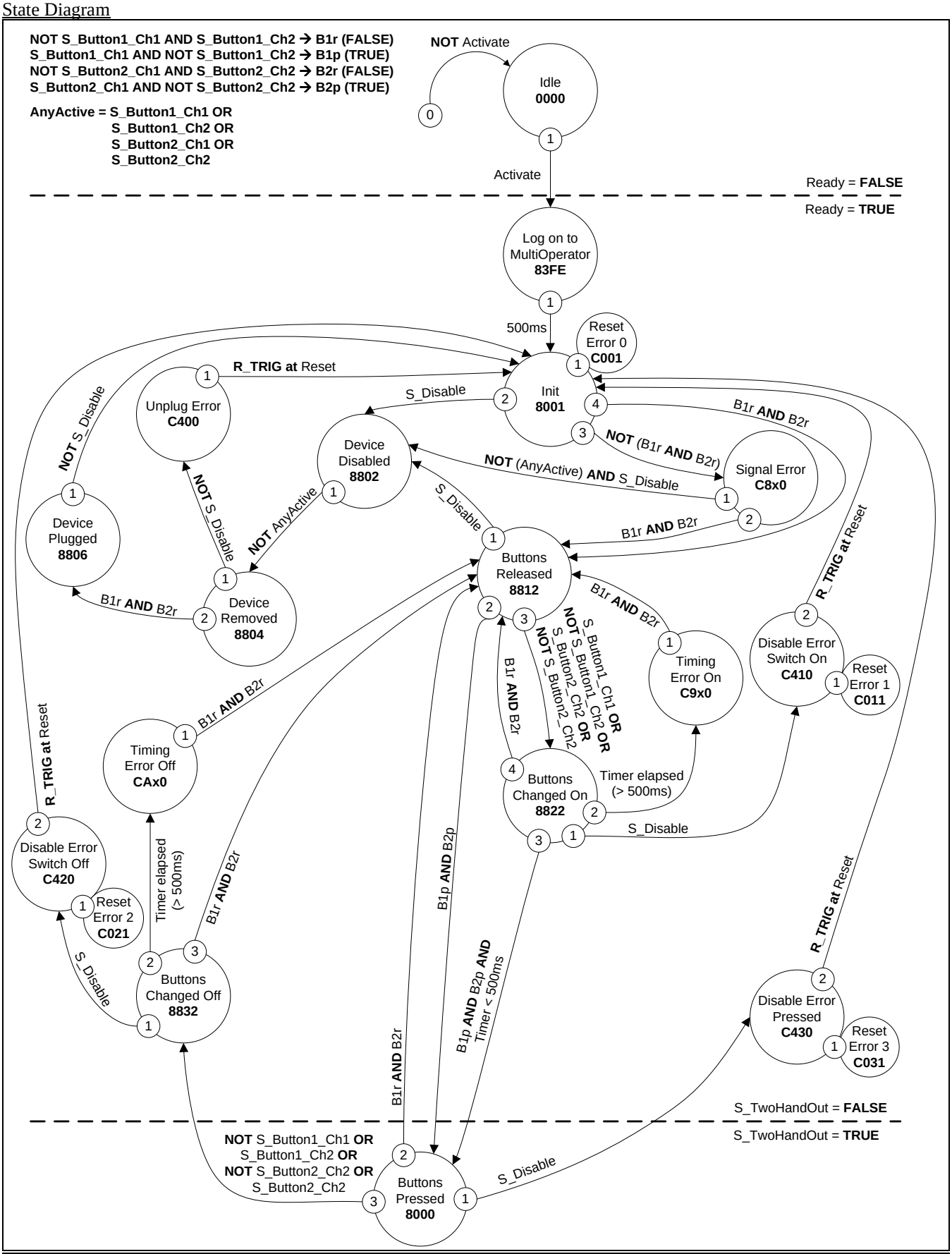
To plug the device without any error message or acknowledgement the following sequence must be maintained:

1. Device is unplugged – S_Disable is set to TRUE, signals from the buttons are FALSE
2. Plug device
3. Set S_Disable to FALSE
4. Device is plugged – buttons are not pressed

If the function block detects a signal change on one of the channels from the button inputs both buttons (all channels) must change the state within 500ms. A wrong signal combination leads to a timing error.

When a button is released – or at least one channel changes it state - the function block checks again the signal change within 500ms. A wrong signal combination leads to a timing error.

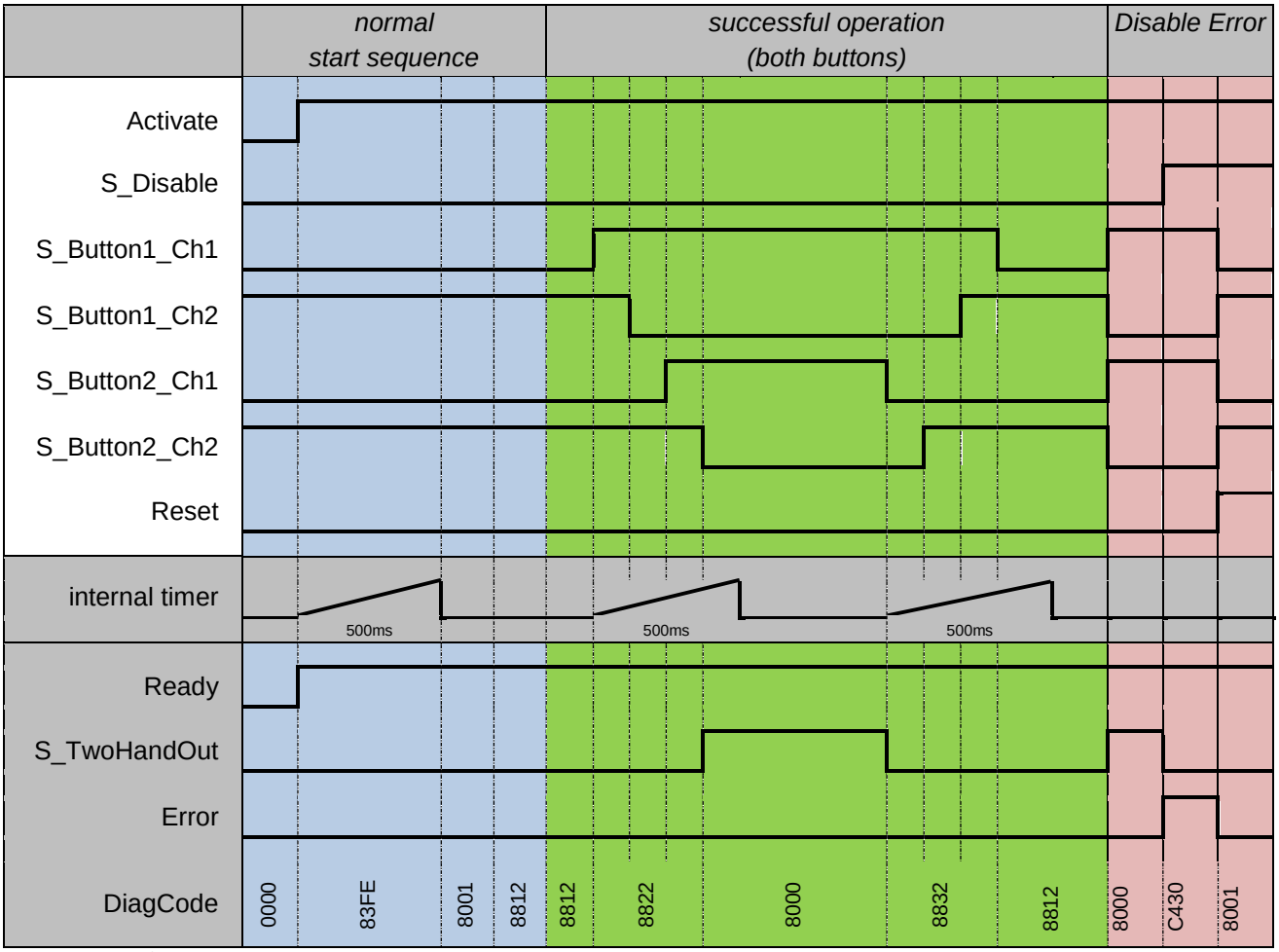
State Diagram



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).

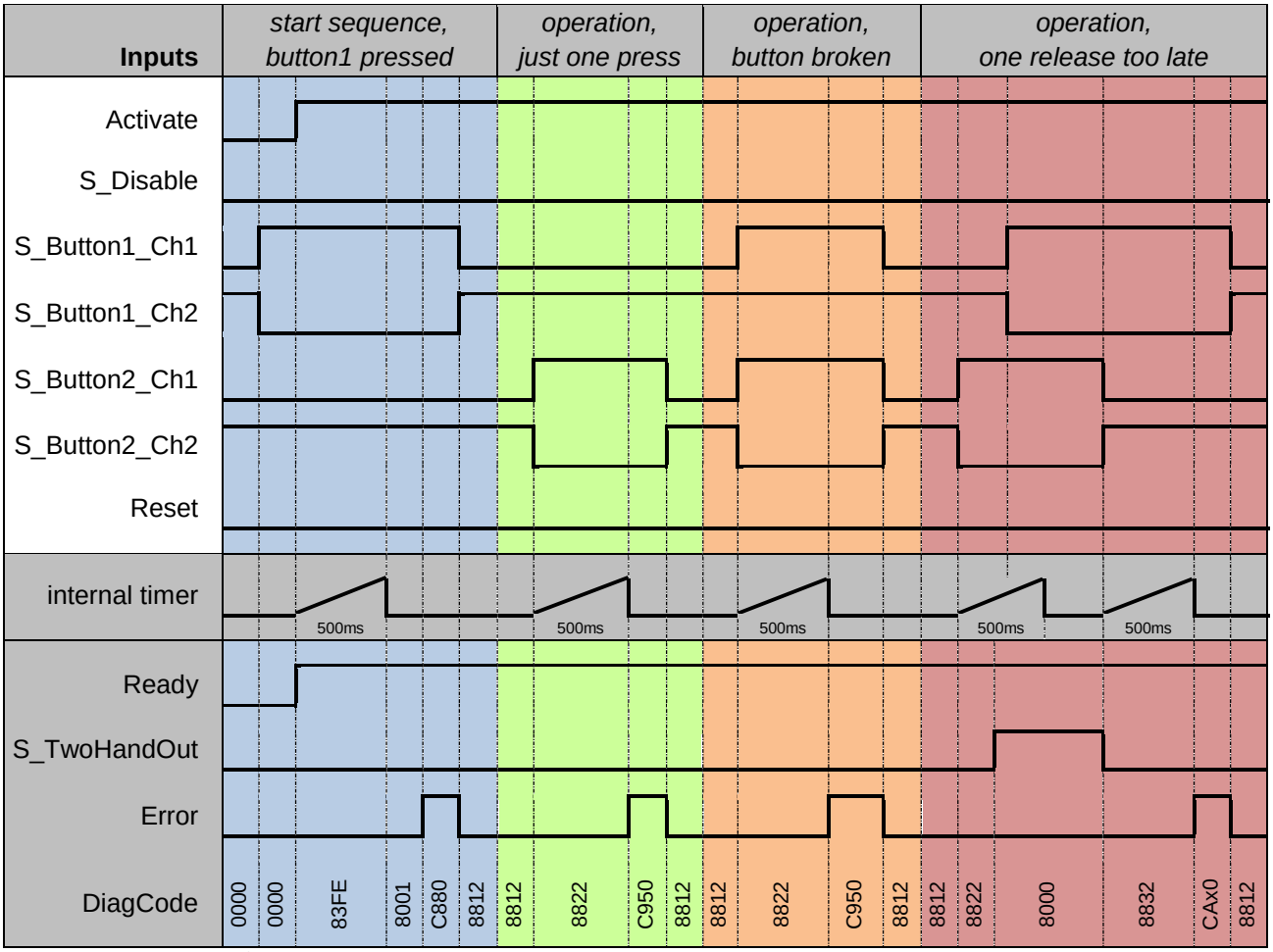
Figure 36: State diagram for SF_TwoHandControlTypeIIIC

Typical Timing Diagrams



Note: Outputs SafetyDemand, OperatorRequest, ResetRequest are not shown in the Timing Diagram

Figure 37: First Timing diagram for SF_TwoHandControlTypeIIIC



Note: Outputs SafetyDemand, OperatorRequest, ResetRequest are not shown in the Timing Diagram

Figure 38: Second Timing diagram for SF_TwoHandControlTypeIIIC

4.11.4. Error Detection

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

4.11.5. Error Behavior

In the event of an error, the S_TwoHandOut output is set to FALSE. The DiagCode output indicates the relevant error code and the Error output is set to TRUE.
A restart is inhibited until the error conditions are cleared and the safe state is acknowledged with Reset by the operator.

4.11.6. Function Block-Specific Error and Status Codes

DiagCode	State Name	State Description and Output Setting
FB-specific error codes, reset required		
C400	Unplug Error	Device is not plugged and not disabled Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

C410	Disable Error Switch On	Buttons are actuated – disable signal received Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C420	Disable Error Switch Off	Buttons are released – disable signal received Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE
C430	Disable Error Pressed	Output is On, buttons are pressed – disable signal received Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = TRUE Error = TRUE

FB-specific error codes, no reset required

C001	Reset Error 0	Static Reset detected in State 8001 Ready = TRUE S_ TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C011	Reset Error 1	Static Reset detected in State C410 Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C021	Reset Error 2	Static Reset detected in State C420 Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE
C031	Reset Error 3	Satic Reset deteced in State C430 Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = TRUE

<table>
<tr><td>C8x0</td><td>Signal Error</td><td>Wrong signal state in State 8001

<table>
<tr><th>B1_Ch1</th><th>B1_Ch2</th><th>B1 State</th><th>B2_Ch1</th><th>B2_Ch2</th><th>B2 State</th><th></th></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C800</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C810</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C820</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C830</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C840</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C850</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C860</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C870</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C880</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C890</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C8A0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C8B0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C8C0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C8D0</td></tr>
</table>

Ready = TRUE
S_TwoHandOut = FALSE
SafetyDemand = TRUE
ResetRequest = FALSE
Error = TRUE</td></tr>
<tr><td>C9x0</td><td>Timing Error On</td><td>Discrepancy Time error when switching on

<table>
<tr><th>B1_Ch1</th><th>B1_Ch2</th><th>B1 State</th><th>B2_Ch1</th><th>B2_Ch2</th><th>B2 State</th><th></th></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C900</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C910</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C920</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C930</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C940</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C950</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>TRUE</td><td>Pressed</td><td>C960</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C970</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C980</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C990</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>C9A0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>C9B0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>C9C0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>C9D0</td></tr>
</table>

Ready = TRUE
S_TwoHandOut = FALSE
SafetyDemand = TRUE
ResetRequest = FALSE
Error = TRUE</td></tr>
</table>

<table>
<tr><td>CAx0</td><td>Timing Error Off</td><td>Discrepancy Time error when switching on

<table>
<tr><th>B1_Ch1</th><th>B1_Ch2</th><th>B1 State</th><th>B2_Ch1</th><th>B2_Ch2</th><th>B2 State</th><th></th></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>CA00</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>CA10</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>CA20</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>CA30</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>CA40</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>CA50</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Released</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>CA60</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>CA70</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>CA80</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>CA90</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>FALSE</td><td>Failure</td><td>CAA0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>FALSE</td><td>TRUE</td><td>Released</td><td>CAB0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>FALSE</td><td>Pressed</td><td>CAC0</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Failure</td><td>TRUE</td><td>TRUE</td><td>Failure</td><td>CAD0</td></tr>
</table>

Ready = TRUE
S_TwoHandOut = FALSE
SafetyDemand = TRUE
ResetRequest = FALSE
Error = TRUE</td></tr>
</table>

FB-specific status codes (no error):

0000	Idle	The function block is not active (initial state) Ready = FALSE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8000	Output Active	Device is plugged and buttons are pressed Ready = TRUE S_TwoHandOut = TRUE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
8001	Init	The function block is activated and initiated Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE
83FE	Log on to Multi-Operator	Wait state to ensure that the SF_TwoHandMultiOperator can be used only in combination with the SF_TwoHandControlTypeIIIC FBs. Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = FALSE ResetRequest = FALSE Error = FALSE

8802	Device Disabled	Device is disabled and plugged Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8804	Device Removed	Device is removed Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8806	Device Plugged	Device is plugged and disabled Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8812	Buttons Released	Device is plugged and buttons are not pressed Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8822	Buttons Changed On	Button status changed – buttons pressed Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE ResetRequest = FALSE Error = FALSE
8832	Button Changed Off	Button state changed – buttons released Ready = TRUE S_TwoHandOut = FALSE SafetyDemand = TRUE 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: 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_FootSwitch		
SF_PressControl		
SF_SingleValveMonitoring		
SF_SingleValveCycleMonitoring		
SF_DoubleValveMonitoring		
SF_ValveGroupControl		
SF_TwoHandMultiOperator		
SF_CamshaftMonitor		
SF_CycleControl		
SF_CamMonitoring		
SF_TwoHandControlTypeIIIC		

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 39: 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.