PETROBRAS	TECHNICAL SPECIFICATION	Nº: I-ET-3010.00-5520-800-P4X-002	
	CLIENT:		SHEET 1 of 27
	JOB:		
	AREA:		
DP&T-SRGE	TITLE: IMPLEMENTATION OF INTERLOCKING AND CONTROL LOGIC		NP-1
			ESUP

MICROSOFT® WORD / V. 365 / I-ET-3010.00-5520-800-P4X-002_B.DOCX

INDEX OF REVISIONS

REV.	DESCRIPTION AND/OR REVISED SHEETS
0	ORIGINAL ISSUE
A	WHERE INDICATED
B	REVISED WHERE INDICATED

	REV. 0	REV. A	REV. B	REV. C	REV. D	REV. E	REV. F	REV. G
DATE	AUG/25/20	APR/05/21	OCT/06/21					
DESIGN	ESUP	ESUP	ESUP					
EXECUTION	U44D	Q082	U44D					
CHECK	U5D6	U49R	U49R					
APPROVAL	U49R	U4JB	U4JB					

SUMMARY

1 INTRODUCTION

1.1 General

1.1.1 This specification describes the technical requirements that shall be complied to implement the Logic for CSS main systems of E&P's Production Units

1.1.2 This specification’s scope is to establish the basic structure to be followed when preparing CSS programs. Package Units shall also use this programming standard.

1.2 Definitions

1.2.1 Refer to I-ET-3010.00-1200-940-P4X-002 - GENERAL TECHNICAL TERMS.

1.3 Abbreviations, Acronyms And Initialisms

BDV – Blowdown Valve;

CDC – *Centro de Distribuição de Cargas* (Load Distribution System, in the portuguese initialism);

CSS – Control and Safety System: set of controllers responsible for interlocking and control functions of the unit;

D&ID – Duct and Instrumentation Diagram document;

Logic Diagrams – Documents in which functional relations between inputs and outputs of an interlocking and control system are represented, according to ISA 5.2 – Binary Logic Diagrams For Process Operations. For further details, refer to I-ET-3010.00-1200-800-P4X-012 - CRITERIA FOR DETAILING DESIGN CAUSE & EFFECT MATRIX;

MCC – Motor Control Center;

OM – Maintenance Inhibition (formerly called Maintenance Override);

OO – Operational Inhibition (formerly called Operational Override);

PLC – Programmable logic controller;

PMS – Power Management System;

P&ID – Process and Instrumentation Diagram document;

SDV – Shutdown Valve;

SOS – Supervision and Operation System, includes Supervision Screens, Data Servers and Historian;

XV – On-off Valve, fail last valve or maneuver valves;

2 REFERENCE DOCUMENTS, CODES AND STANDARDS

2.1 Internal References

I-ET-3010.00-1200-800-P4X-012	CRITERIA FOR DETAILING DESIGN CAUSE & EFFECT MATRIX;
I-ET-3010.00-5140-700-P4X-003	ELECTRICAL REQUIREMENTS FOR PACKAGES FOR OFFSHORE UNITS
DR-ENGP-M-I-1.3	SAFETY ENGINEERING GUIDELINE
I-ET-3010.00-5520-861-P4X-001	CONTROL AND SAFETY SYSTEM – CSS
I-ET-3010.00-5520-861-P4X-002	SUPERVISION AND OPERATION SYSTEM - SOS
I-ET-3010.00-5520-800-P4X-001	SUPERVISION AND OPERATION SYSTEM SCREENS

2.1.1 Names below and respective document codes may vary according to each project but, in general, the following documents shall be considered along with this technical specification.

- AUTOMATION AND CONTROL ARCHITECTURE
- EMERGENCY SHUTDOWN DIAGRAM
- AUTOMATION AND CONTROL FUNCTIONS DESCRIPTIVE MEMORANDUM

3 GENERAL DESCRIPTION FOR APPLICATION PROGRAM

3.1 General Requirements

3.1.1 Mapping the memory shall be performed in a way that all tables resulting in input or output for SOS have a contiguous addressing, which allows the communication through one transmission block, yet minimizing the occupancy time of the data network. Address allocation shall allow future expansion, of minimum 30%, without the need to readdress all the remaining variables.

3.1.2 This methodology is applied to discrete, physical, or virtual-type signals, where the treatment via data tables facilitates the programming/checking.

3.1.3 All functional blocks or customized functions and its programming for the project shall have their content available for PETROBRAS.

3.1.4 Programs and sub-routines shall be developed with identification of each variable starters, intermediary and final elements, being mandatory the use of TAGs similar to the physical element in case of input and output cases.

3.1.5 Each program line, function block or customized function for the project shall have comments on its features.

3.1.6 When developing the application, the programming languages to be used shall be those from IEC 61131-3 - Programmable controllers – Part 3: Programming languages.

3.1.7 All control logics shall be made considering P&IDs, D&IDs and logic diagrams.

3.1.8 All shutdown and emergency logics shall be made considering EMERGENCY SHUTDOWN DIAGRAM, P&IDs, D&IDs, Cause and Effect Matrices and DR-ENGP-M-I-1.3 – SAFETY ENGINEERING GUIDELINE.

3.1.9 All elements within a control loop logic or an interlocking logic (initiators, controllers, final elements etc.) shall be contained in the same CSS subsystem. No logic shall be made in SOS.

3.1.10 In case of power supply failure, the following events shall occur:

- The application program shall be kept in PLC retentive memory;
- After power supply is re-established, the PLCs shall reinitialize automatically, running all self-diagnose routines;
- All “sequence of event” logics shall go to a pre-determined standby state;
- All control loops automatic/manual states shall go to a pre-determined condition (to be defined in Detail Engineering Design according to each loop);
- All control loops configuration parameters shall receive the currently read values before the failure.

3.1.11 In case of input or output failure, the following events shall occur:

- The controllers primary variables (process variables) shall generate an alarm and switch to manual mode;
- The controllers secondary variables (manipulated variables) shall generate an alarm and replace their values to a pre-determined value;
- Every variable which is a part of a “sequence of events” logic shall generate an alarm and stop the respective sequence in the current step (or restart the sequence, for some sequences that shall be defined during Detail Engineering Design).
- For each analog input and output, underrange and overrange values (less than 4 mA and/or greater than 20 mA) shall be identified in order to execute failure logic. This logic (for instance, lead to shutdown) shall be confirmed during Detail Engineering Design Phase in conjunction with PETROBRAS.
- For each input / output failure, an alarm shall be generated. In SOS screens, these alarms shall be considered as Alerts.

3.1.12 In case of CSS equipment (CPU, cards, power supply, networks) self-diagnostic failure, the following events shall occur:

- If both CPUs lose their power supply: generate an alarm and make all outputs go to the fail-safe condition;
- Loss of only one of redundant components (such as a PLC CPU) shall generate an alarm.
- Undefined logical values (such as division by zero, square roots of negative numbers) shall make the respective output go to the fail-safe condition;

- Input signals incompatible with 4-20 mA (for analog inputs) or 24 Vdc (for discrete inputs) shall generate an I/O channel failure alarm and the corresponding outputs of the same loop of this input shall go to the fail-safe condition or other pre-defined value to be agreed during Detail Engineering Design Phase.

3.1.13 The logic shall be developed taking into account that it is fail safe, which means that on power loss, all equipment shall be led to a safe state.

3.1.14 All variables shall have a pre-defined (initial) value, to be defined during Detail Engineering Design Phase.

3.1.15 By the time the program is developed, Company internal Inhibition Policy shall be consulted. Whenever the term “bypass” is used in this document, it means input inhibition.

3.2 Basic Structure

3.2.1 Program shall be composed by the following parts:

- Input Polarization Logic.

- Input Filtering Logic.

- Input Override (Bypass Logic).

- Alarm Sealing and Acknowledgment.

- Process Interlocking Logic.

- Safety Interlocking Logic.

- Output Override Logic

- Output Logic

3.3 Block Diagram

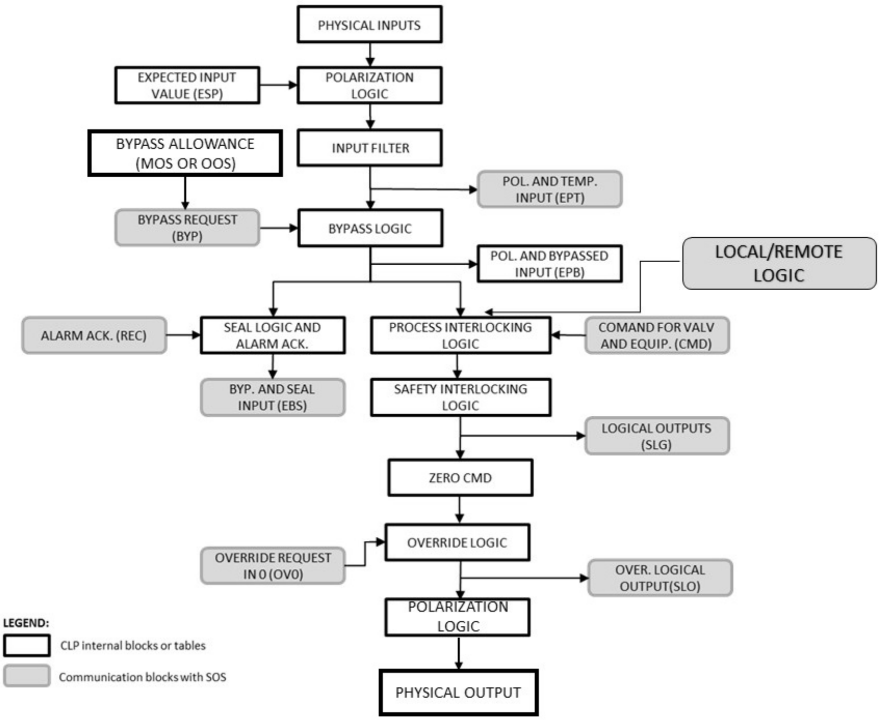

PHYSICAL INPUTS
EXPECTED INPUT VALUE (ESP)
POLARIZATION LOGIC
BYPASS ALLOWANCE (MOS OR OOS)
INPUT FILTER
POL. AND TEMP. INPUT (EPT)
BYPASS REQUEST (BYP)
BYPASS LOGIC
POL. AND BYPASSED INPUT (EPB)
LOCAL/REMOTE LOGIC
ALARM ACK. (REC)
SEAL LOGIC AND ALARM ACK.
PROCESS INTERLOCKING LOGIC
COMAND FOR VALV AND EQUIP. (CMD)
BYP. AND SEAL INPUT (EBS)
SAFETY INTERLOCKING LOGIC
LOGICAL OUTPUTS (SLG)
ZERO CMD
OVERRIDE REQUEST IN 0 (OV0)
OVERRIDE LOGIC
OVER. LOGICAL OUTPUT(SLO)
LEGEND:
CLP internal blocks or tables
Communication blocks with SOS
POLARIZATION LOGIC
PHYSICAL OUTPUT

Figure 1 – General block diagram for the programming logic

Note 1: The virtual inputs shall be created already containing the proper polarization, with no need to perform the operation again in the beginning of a logical process.

3.4 Maintenance Inhibition (OM) logic

3.4.1 Company internal Inhibition Policy/Standard shall be consulted.

3.4.2 OM is a maintenance inhibition command that may be applied to input instruments, including fire and gas detectors. During OM application, output logic and alarms shall be inhibited.

3.4.3 OM commands are manual and individually issued and confirmed through SOS screens.

3.4.4 When OM command is issued, all the logic switches associated to the instrument L, LL, LLL, H, HH, HHH) shall also be inhibited, but their real time values (input value and EPT table) shall remain readable in Supervisory System.

3.4.5 If the instrument is in an abnormal condition (at least one of the logical switches is still actuated), the removal of its inhibition shall not be allowed. When the normal condition is restored, the operator may remove its inhibition.

3.4.6 The removal of the maintenance inhibition shall also be individual for each instrument.

3.5 Operational Inhibition (OO) logic

3.5.1 Company internal Inhibition Policy/Standard shall be consulted.

3.5.2 OO is an Operational Inhibition command that shall be available for each logic signal (e.g., L, LL, LLL, H, HH, HHH) and shall be issued and confirmed through SOS.

3.5.3 OO duration time may be configured using a variable, configurable for each instrument/logic switch, or pre-determined according to Company´s internal inhibition policy. After this time, OO shall not be automatically removed, since the operation may still not be finished yet. Just before OO time is elapsed, an alarm shall be generated (typically 2 to 5 minutes before). Duration time and alarms are individual for each logic switch.

3.5.4 During OO time, interlocking logic and alarms are inhibited, but the real time values (variable and logic switches – EPT table) shall remain readable by the Supervisory System.

3.6 Startup-bypass logic

3.6.1 A group of inhibition commands may be automatically issued to signals during an equipment startup (for instance, pumps and their corresponding PSLL, PSHH). This automatic inhibition logic shall be generated in CSS. After the equipment is on, these inhibitions may also be automatically reset.

3.6.2 The equipment that will be subject to startup-bypass commands shall be confirmed during Detail Engineering Design Phase in conjunction with PETROBRAS.

NOTE: All inhibition commands shall be auditable and trackeable.

3.7 PLC Memory Map

3.7.1 Taking into account each PLC addressing characteristics as well as requirements of item 3.1, the PLC memory mapping shall foresee the following tables, not necessarily with each item in the same order as shown in the table:

PLC MEMORY		
MNEM	**TABLE**	**ACTON**
ENT	Physical Inputs	Intern. To PLC
ENV	Virtual Inputs	Intern. To PLC
EPB	Polarized and Bypassed Inputs	Intern. To PLC
SAI	Physical Outputs	Intern. To PLC
SLG	Logical Outputs	Supervisory reads
SLO	Overridden Logical Outputs	Supervisory reads
EPT	Polarized and Temporized Inputs	Supervisory reads
EBS	Polarized, Bypassed and Sealed Inputs	Supervisory reads
REC	Alarm Acknowledgment	Supervisory writes
BYP	Bypass Request	Supervisory writes
OV0	Override Request “0”	Supervisory writes
CMD	Equipment and Valve Command	Supervisory writes
ESP	Expected Input State	Def. in PLC

4 FUNCTIONAL BLOCKS

4.1 Polarization and Input Filter

4.1.1 Considering that inputs may assume different state values at NORMAL and ABNORMAL conditions, it is necessary to establish internal standard values to represent them, in order to simplify the programming. Internally to the PLC, the binary polarized input signals shall always assume the following values downstream the polarization logic:

- Variable in Normal Condition = 0

- Variable in Abnormal Condition = 1

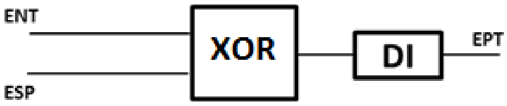
ENT (Physical Inputs):

- Open Contact = 0

- Closed Contact = 1

4.1.2 The normalization of the Input states is to be performed by means of “XOR” (exclusive “OR”) function applied to the “Expected Input State” table and “Physical Input”.

4.1.3 The logic representation for polarization and filtering of input is as follows:



ENT – Physical Inputs

ESP – Expected Input States

EPT – Polarized and Temporized Inputs

Figure 2 – Polarization and filtering logic

4.1.4 All Variables input shall be temporized in order to filter the alarm in case of signal bouncing or variable fluctuations. Typical values for typical variables are shown below.

- Pressure –1seconds
- Flow – 0.5 seconds
- Temperature – 2 seconds
- Level – 3 seconds
- FGS Detectors 1 seconds
- Physical/Limit switches – 0.3 seconds

4.1.5 Process timers or sequential timers are not considered on these timers. These Process timers are explicit on PI&D and/or Cause effect Matrix (interlock Matrix). Ex.: Turn on spare Fan in case of 30 seconds of FSL activated.

4.1.6 Debouncing timer shall be configurable individually

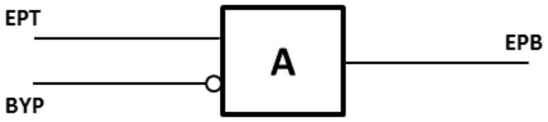
4.2 Input Inhibition (Bypass)

4.2.1 The input inhibition (also called bypass logic) allows the activation and deactivation of inhibition function at any input point on the process to allow equipment or instrument maintenance, process start-up or process restart after a shutdown due to a failure, without causing loss to the process or equipment.

4.2.2 It shall be possible to set or cancel any inhibition (OO, OM and start-up bypass) with commands executed through SOS or by writing the inhibition command directly on the “By-pass Request Table” (BYP) of the PLC.

4.2.3 On “By-pass Request Table”, the process input points requested to be inhibited will be indicated with logical state “1”. When a logic input is inhibited, its corresponding value in table EPB goes to false (zero).

4.2.4 The logic representation for bypassing the inputs is as follows:



EPT – Polarized and Temporized Inputs

BYP – Bypass Request

EPB – Polarized and Bypassed Input

Figure 3 – Logic representation of input bypass

4.2.5 For further details, refer to items 3.4 and 3.5.

4.3 Alarm and Seal Logic

4.3.1 The alarm and seal logic has the function to retain the alarm signal until it is acknowledged by operator, even if the correspondent input signal has already been normalized. The PLC will generate alarm points (EBS) that will be used by SOS for alarm annunciation. The Acknowledgment points table (REC) shall be set by SOS at the alarm acknowledgement to unlatch the seal, releasing the alarm.

4.3.2 The set (logical state “1”) of the Alarm Acknowledgment Table (REC) is done by the Supervisory, at the moment the operator acknowledges the alarm. This “set” value is kept during a short interval of time so as to make it compatible with the supervisory software polling time. The time representation for those actions can be seen below:

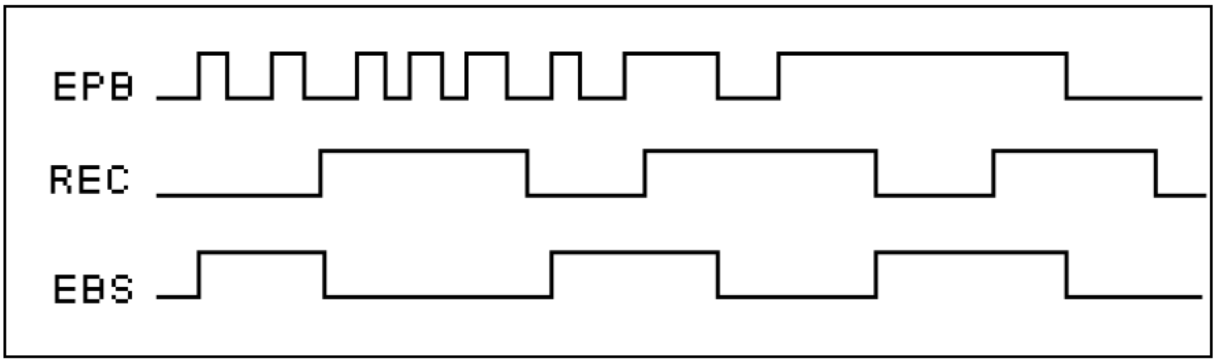


Figure 4 – Alarm acknowledgment in time

4.3.3 The logic representation for alarm and seal the input is as follows:

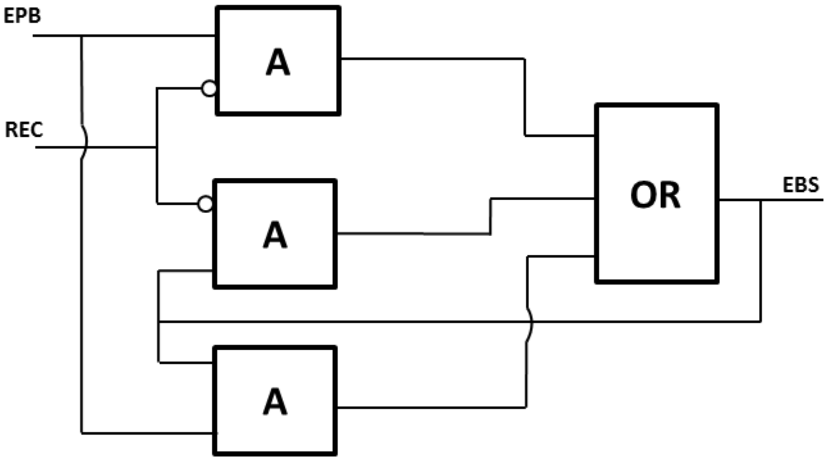


Figure 5 – Alarm Logic

NOTE: Alarm suppression logics shall be implemented in CSS, according to definitions found in other project´s documents and shall write in the final value of EBS.

4.4 Output Override Logic

4.4.1 The Output Override logic function allows the activation and deactivation of any process output signal or signals override, for equipment or instrument maintenance without causing any shutdown to the process. Actions initiated by Fire and Gas Detectors shall cancel or surpass override commands. Example: Operator can override a FSL, or other instrument to open a damper, but gas confirmed on damper intake shall prevail and close damper anyway. In this case, the output value, although overridden, shall still be “0”.

4.4.2 Considering that discrete inputs may assume different state values at NORMAL and ABNORMAL conditions, it is necessary to establish internal standard values to represent them, in order to simplify the programming. Internally to CSS controllers, the binary polarized input signals shall always assume the following values downstream the polarization logic:

TYPE OF DEVICE	LOGICAL “0” STATE	LOGICAL “1” STATE
On-Off Valves/Dampers	Closed	Open
Motors	Off	On
Alarm signals	Normal	Actuated
Shutdown signals	Normal	Actuated
Field States (ZSH/ZSL, YSHL, XS)	Normal	Actuated

4.4.3 The final control elements shall be energized under the following conditions:

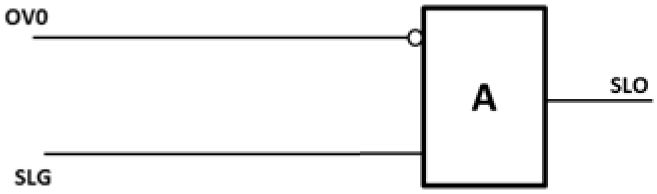
- SDVs and XVs fail close - Energize to open.

- BDVs and XVs fail open - Energize to close. In order to do so, since “1” means “valve opened”, as above, the output of the BDV and XV fail open blocks shall be connected to a logical inverter (“0” to “1” and “1” to “0”).

- CO2 (triggering valve) - Energize to open.

4.4.4 PLC will activate or cancel the override function through commands executed from the SOS or by directly writing on the PLC Tables. Actions initiated by Fire and Gas Detectors shall cancel or surpass override commands, as stated in item 4.4.1.

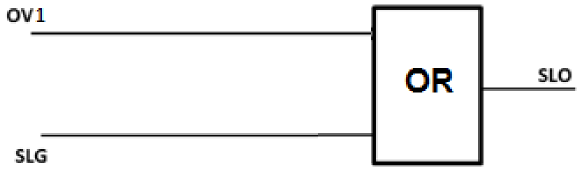
4.4.5 The logic representation for override the output is as follows:



OV0 – Request Override on “0”

SLG – Output without Override

SLO – Output with Override



OV1 – Request Override on "1"

SLG – Output without Override

SLO – Output with Override

Figure 6 – Output Override Logic

4.5 Equipment and Valve Command Logic

4.5.1 All command points shall be present in CMD table which shall be set through the supervisory. Table reset is done through PLC at the end of each cycle.

4.5.2 The outputs corresponding to the Shutdown Valves (SDVs) and Blowdown Valves (BDVs) shall only be energized after the individual reset signal is sent, originating from the Supervisory through CMD table (as stated in item 3.6.1) itself.

4.6 Analog variables logic

4.6.1 The values of analog variables sent to the supervisory shall be in engineering units.

4.6.2 The features described in the table below, represent the requirements related to logic performance that shall be in the implementation/settings of the functional block. The number of inputs and outputs described below may vary, depending on the number of alarms levels and other project/application specificities.

Inputs	Description	Note
IN	Analogic variable input of the sensor	Floating point-type variable
SPHHH	Alarm setpoint very very high	Floating point-type variable
SPHH	Alarm setpoint very high	Floating point-type variable
SPH	Alarm setpoint high	Floating point-type variable
SPL	Alarm setpoint low	Floating point-type variable
SPLL	Alarm setpoint very low	Floating point-type variable
SPLLL	Alarm setpoint very very low	Floating point-type variable
HYS_U	Common hysteresis considered for the up transition	Floating point-type variable
HYS_D	Common hysteresis considered for the down transition	Floating point-type variable
FLH	Sensor defective	Boolean-type variable
Outputs	**Description**	**Note**
HHH	IN variable equal to or higher than SPHHH	Boolean-type variable
HH	IN variable equal to or higher than SPHH	Boolean-type variable
H	IN variable equal to or higher than SPH	Boolean-type variable
L	IN variable equal to or lower than SPL	Boolean-type variable
LL	IN variable equal to or lower than SPLL	Boolean-type variable
LLL	IN variable equal to or lower than SPLLL	Boolean-type variable

Feature description
The functional block receives the analogic value converted to engineering units arising from the instrument, and compares it to the setpoints for generating outputs considering the hysteresis. The hysteresis works as the examples as follows: For SPH, SPHH and SPHHH setpoints: In case the signal exceeds the setpoint value considered, it will occur the output generation, and this output shall be disabled from the block when the signal is below the setpoint minus the HYS_D value. For SPL, SPLL and SPLLL setpoints: In case the signal is below the setpoint value considered, it will occur the output generation, and this output shall be cancelled from the block when the signal is above the setpoint plus the HYS_U value. When the input activation related to sensor defective (FLH) occurs the outputs shall be disabled. The FLH output shall be activated in case the input is below 4 mA or above 20 mA.

4.6.3 The logic shall follow the schematic below (the same applies to SPHH/SPLL/HH/LL and SPHHH/SPLLL/HHH/LLL):

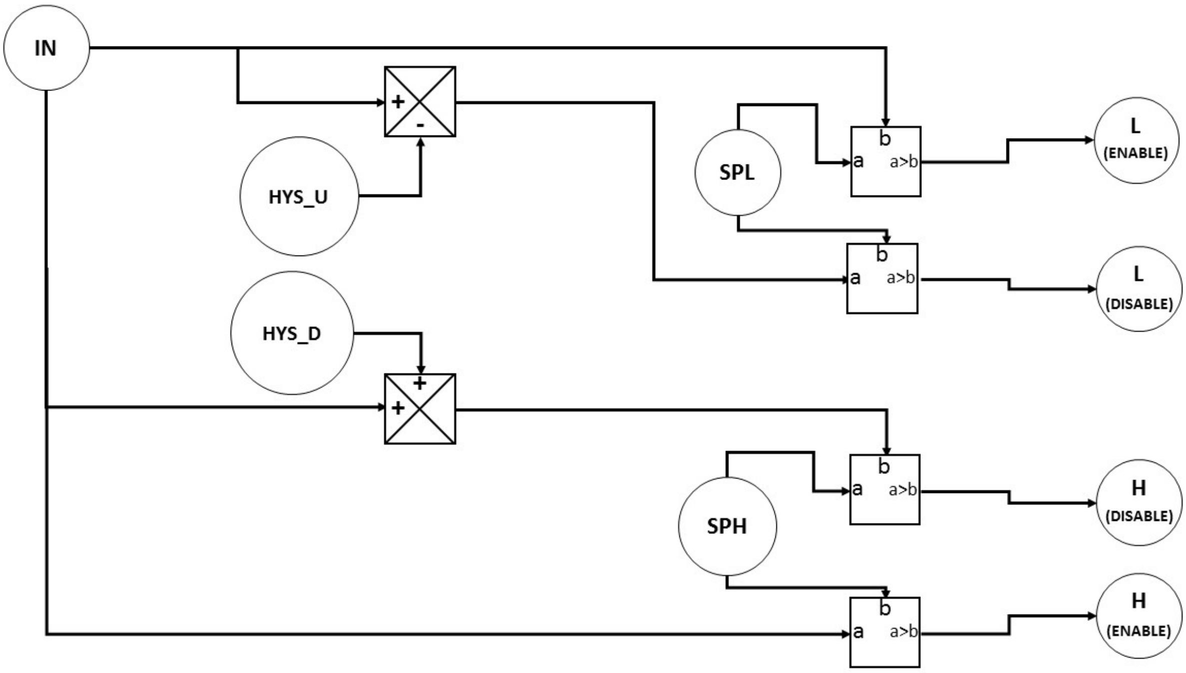


Figure 7 – Analog variables logic

4.7 Delay Initiation

4.7.1 The features described in the table below represent the minimum requirements related to logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application. All alarms described in item 4.6 shall have this functionality. The times for each alarm shall be defined during Detail Engineering Design

Inputs	**Description**	**NOTE**
IN	Variable Input	Boolean-type variable
DELAY	Delay time for output generation	Time in seconds
Outputs	**Description**	**NOTE**
OUT	Alarm output delayed	Boolean-type variable
Feature description		
The functional block receives the variable value, and after the predefined time occurs (DELAY), with an input value continuously enabled, the output is enabled. When the input value is disabled, the output is disabled at the same time.		

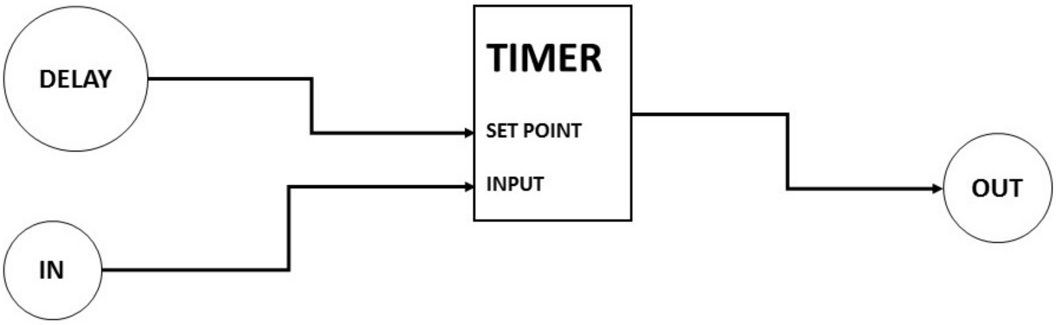


Figure 8 – Delay logic

4.8 ON/OFF Control

4.8.1 The features described in the table below represent the minimum requirements related to logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application.

Inputs	Description	NOTE
IN	Variable Input	Floating point-type variable
SP_SUP	Setpoint superior control	Floating point-type variable
SP_INF	Setpoint inferior control	Floating point-type variable
D_R	Direct/reverse control	Boolean-type variable
Outputs	**Description**	**NOTE**
OUT	Activation/deactivation	Boolean-type variable
Feature description		
When the D_R variable is disabled (0), the block control is direct, which means that when the IN variable is above SP_SUP variable, the output (OUT) is triggered and remains at this state until the input variable (IN) is lower than SP_INF. When the D_R variable is enabled (1), the block control is reverse, which means that when the IN variable is below SP_INF variable, the output (OUT) is enabled and remains at this state until the input variable (IN) is higher than SP_SUP. The block output shall match the equipment that needs different outputs to be triggered and disabled. D_R variable shall be visible to operator on SOS		

4.8.2 The logic shall follow the schematics below:

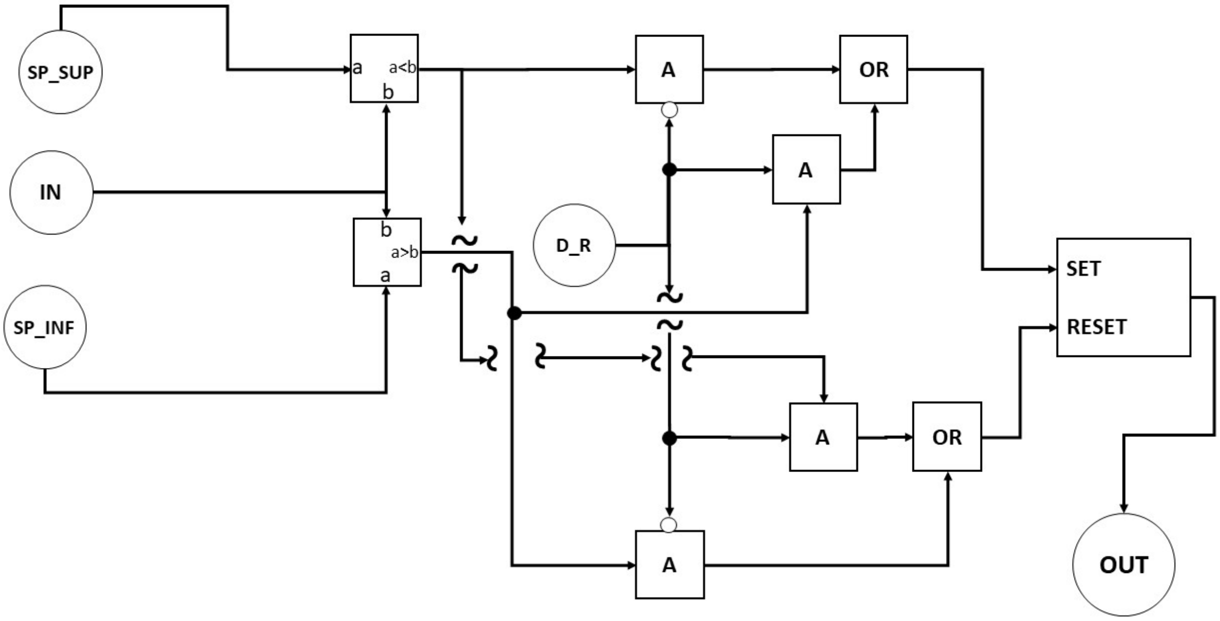


Figure 9 – On-off logic

4.9 ON/OFF Valve

4.9.1 The features described in the table below represent the minimum requirements related to ON/OFF valve logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application.

Inputs	Description	NOTE
HSH	Open valve command	Boolean-type variable
HSL	Close valve command	Boolean-type variable
ILO	Interlocking logic from process to open the valve	Boolean-type variable
ILC	Interlocking logic from process to close the valve	Boolean-type variable
ZSH	Open valve indication	Boolean-type variable
ZSL	Closed valve indication	Boolean-type variable
DLY_O	Delay time for failure detection in opening transition	Time in seconds
DLY_C	Delay time for failure detection in closing transition	Time in seconds
Outputs	**Description**	**NOTE**
OUT	Solenoid energization/de-energization	Boolean-type variable
OUT_O	Activation of solenoid to open XV (applicable only to XVs fail latch)	Boolean-type variable
OUT_C	Activation of solenoid to close XV (applicable only to XVs fail latch)	Boolean-type variable
FLH	Failure on valve command	Boolean-type variable

Feature description

For SDVs, the HSH shall energize the solenoid to open the valve. The HSL or ILC shall de-energize the solenoid and close the valve.

For BDVs, the HSL shall energize the solenoid to close the valve. The HSH or ILO shall de-energize the solenoid and open the valve.

For XVs, the energization or de-energization of the solenoid are given by its failure mode.

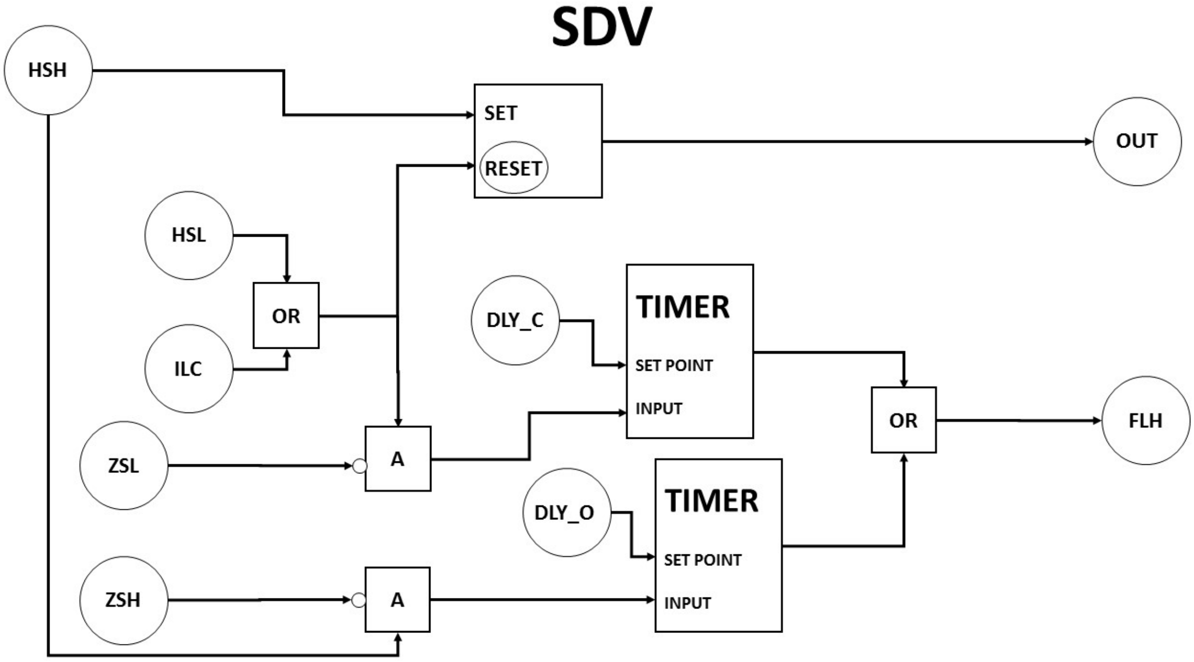
- If the XV is fail-open, the HSL or ILC shall energize the solenoid to close the XV and the HSH or ILO shall de-energize the solenoid to open the XV.
- If the XV is fail-close, the HSH or ILO shall energize the solenoid to open the XV and the HSL or ILC shall de-energize the solenoid to close the XV.
- If the XV is fail-last, there are two solenoids, one to open (OUT_O) and other to close (OUT_C) the XV. HSL/ILC energize OUT_C and HSH/ILO energize OUT_O. The absence of energy keeps the XV in its last position.

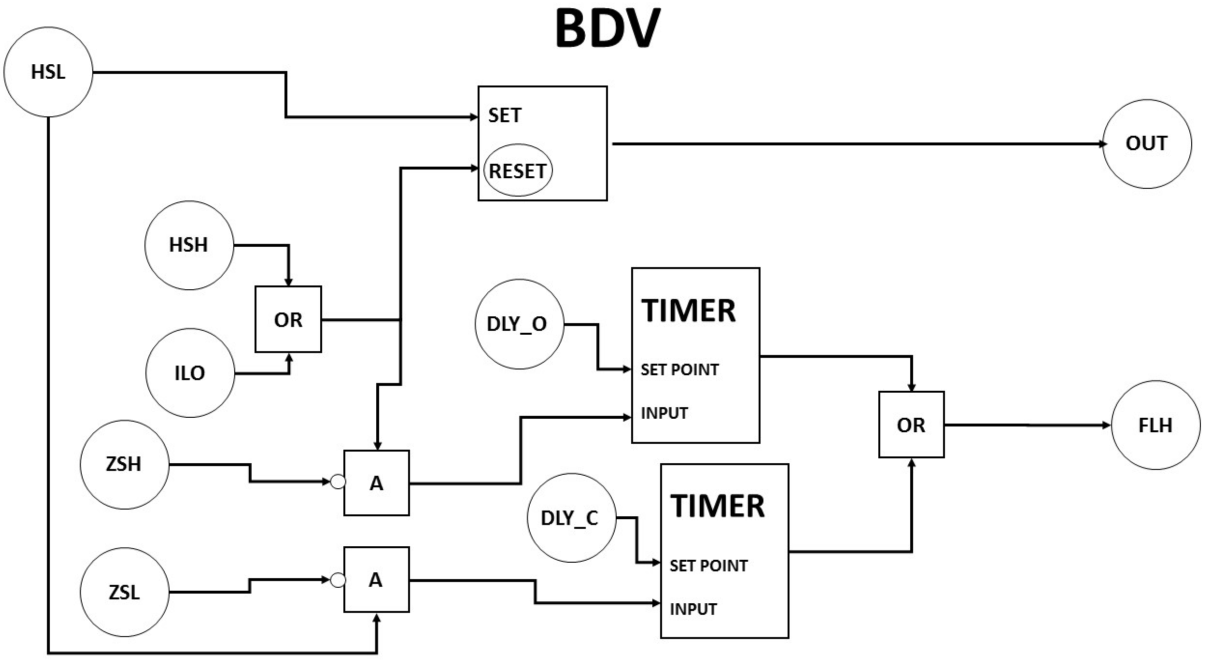
For all valves, if an open command is given and after DLY_O passes ZSH is not detected, or if a close command is given and after DLY_C passes ZSL is not detected, FLH is generated

NOTE: “TIMER” block works as follows:

- Set and begin counting time in a “0” to “1” transition
- Reset and stop counting time in a “1” to “0” transition

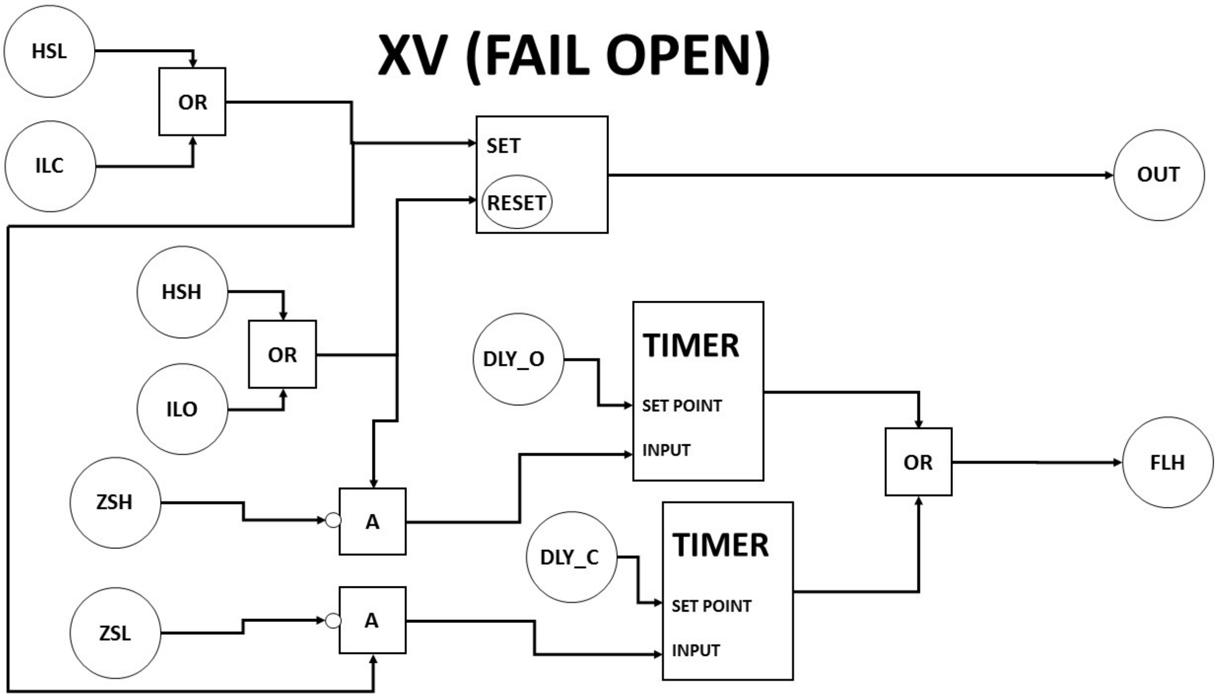
4.9.2 The logic shall follow the schematics below:





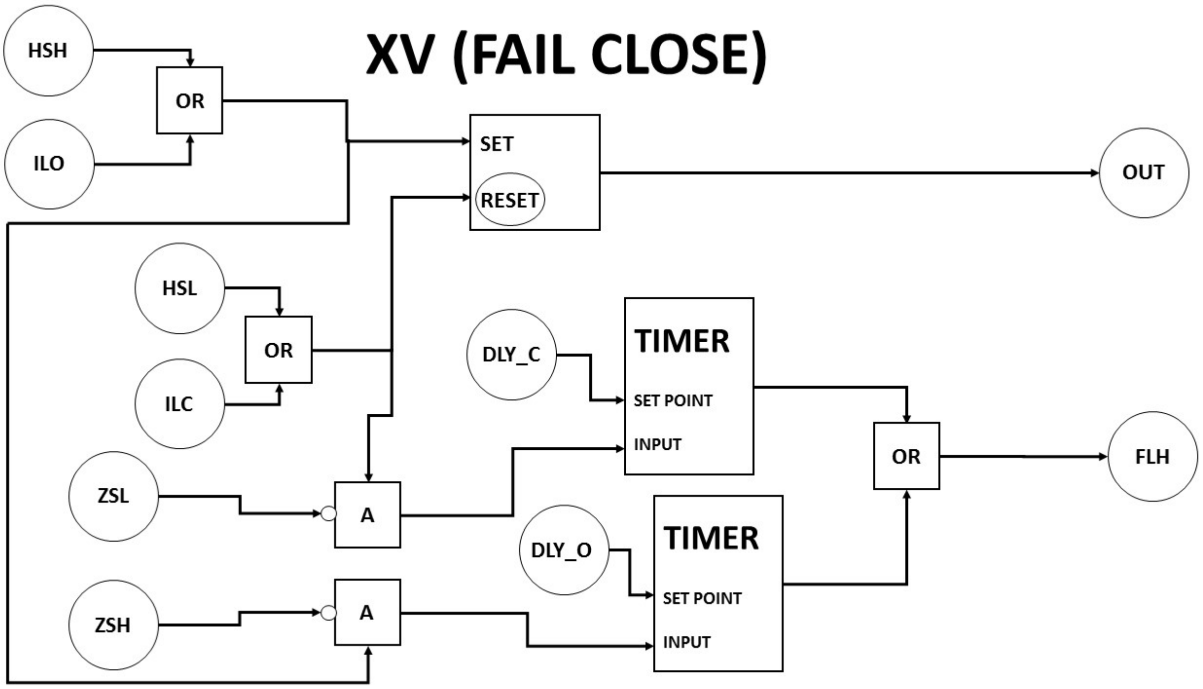
NOTE: It shall be connected an inverter to the output of each BDV block to achieve “0” as “valve closed” and “1” as “valve open”, as stated in item 4.1.1.

XV (FAIL OPEN)



NOTE: It shall be connected an inverter to the output of each XV (fail open) block to achieve “0” as “valve closed” and “1” as “valve open”, as stated in item 4.1.1.

XV (FAIL CLOSE)



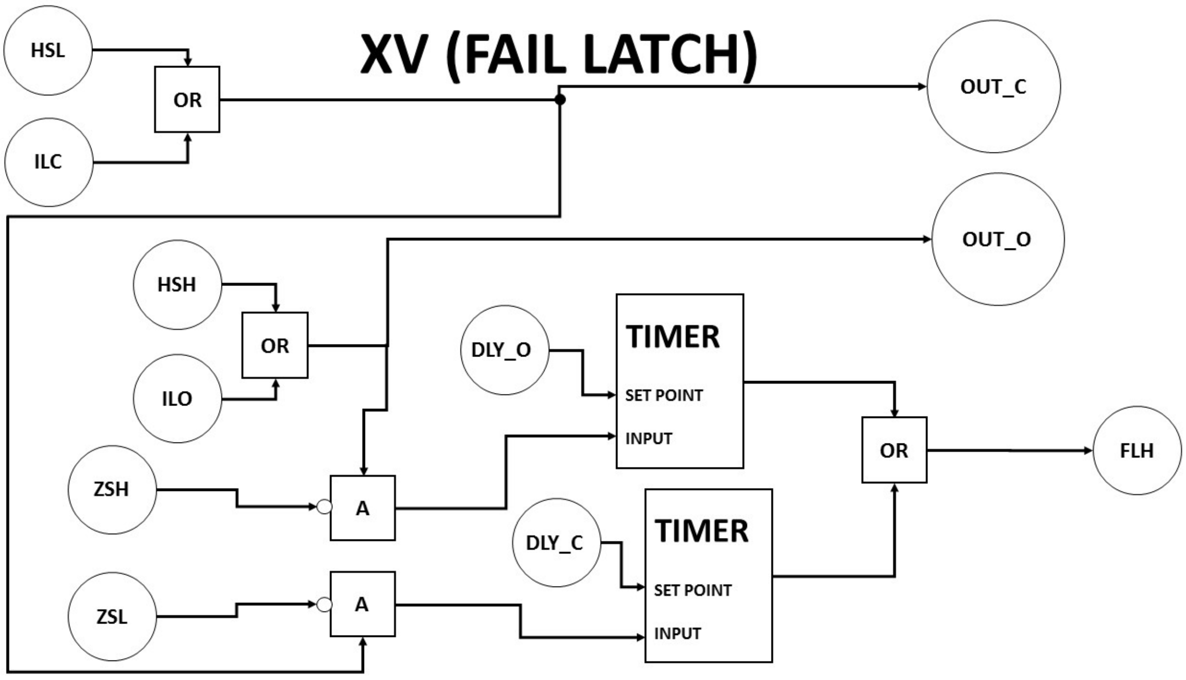


Figure 10 – Valves logic

4.10 Process sensors voting (KooN)

4.10.1 The features described in the table below represent the minimal requirements related to process sensors voting logic performance that shall be in the implementation/settings of the functional block.

4.10.2 The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application. This voting logic shall have an associated timer (as described in item 4.7) initially set in 0 seconds. The final value for this timer for each voting logic block will be defined in Detail Engineering Design.

4.10.3 This logic is valid for process sensors only. For Fire and Gas Detectors voting logic, see item 4.13.

<table>
<tr><th>Inputs</th><th>Description</th><th>NOTE</th></tr>
<tr><td>IN_I</td><td>Abnormal condition from sensor I (where i = 1, 2, …, N)</td><td>Boolean-type variable</td></tr>
<tr><td>BYP_I</td><td>Bypass from sensor I (where i = 1, 2, …, N)</td><td>Boolean-type variable</td></tr>
<tr><th>Outputs</th><th>Description</th><th>NOTE</th></tr>
<tr><td>OUT</td><td>Voting confirmed</td><td>Boolean-type variable</td></tr>
<tr><th colspan="3">Feature description</th></tr>
<tr><td colspan="3">A voting KooN (K out of N) works by the following premise: if any K sensors of the whole set of N sensors gives an abnormal condition, the output of the block is activated.
If any M of the sensors are bypassed (for example, to perform maintenance), the value of its BYP_I shall be changed to 1, and the voting changes to (K-M)oo(N-M). If K-M is less than or equal to 1, the voting shall stay as 1oo(N-M). The bypassed sensors are not considered in the voting.
For example, if the voting is 2oo3 and 1 sensor is bypassed for maintenance, the voting becomes 1oo2. If, then, another sensor is bypassed, the voting becomes 1oo1.
The logic of this block shall be done via iteration loops. An “if-then-else” logic is not allowed.
<b>NOTE:</b> K, N and M are positive integers, and K<N</td></tr>
</table>

4.10.4 Below, are given the schematic logic for voting

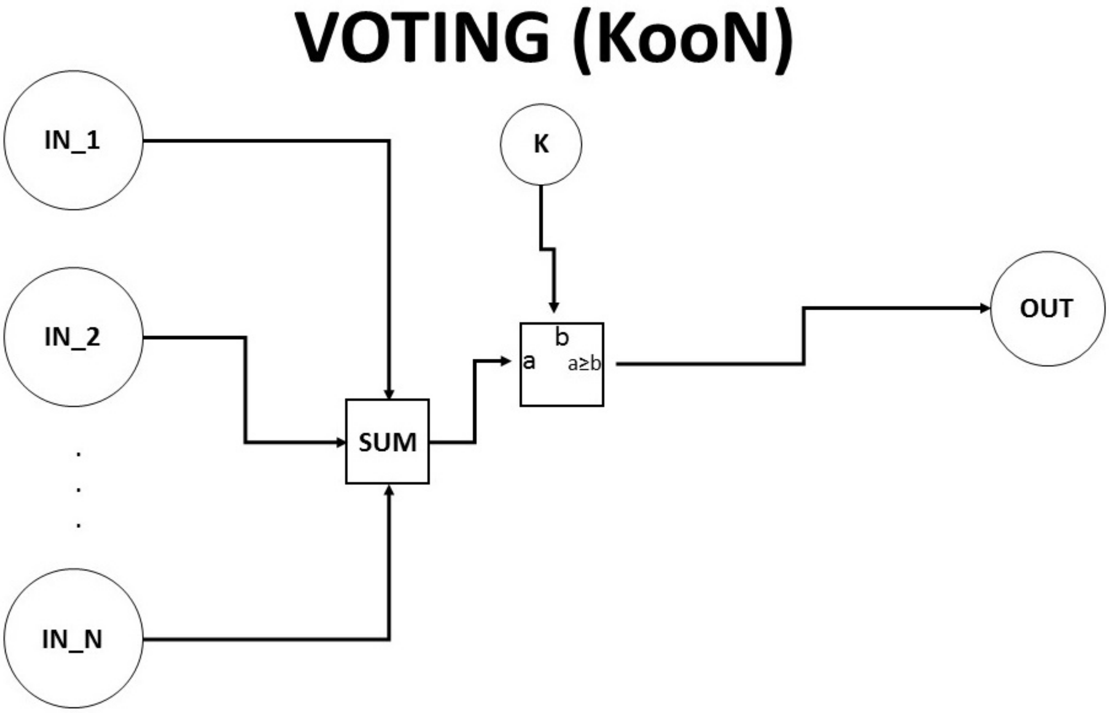


Figure 11 – Voting logic

4.11 Continuous Control

4.11.1 The features described in the table below represent the minimal requirements related to logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application. Split-range controls shall be programmed using the signal OUT_CD, splitting the value to the multiple final elements.

4.11.2 Manual commands to the valve are not presented in this logic. However, the implementation of manual commands shall be such that, despite the failure mode of the final element, operator shall indicate on SOS Screen the percentage of valve opening required, i.e. if 100% is informed, the valve shall be fully opened and if 0% is informed it shall be fully closed.

Inputs	Description	NOTE
IN	Input from the process variable	Floating point-type variable
SP	Set Point	Floating point-type variable
Outputs	**Description**	**NOTE**
OUT_MV	Output From the Controller, Manipulated Variable	Floating point-type variable
OUT_CD	Physical output to the I/O Card	Floating point-type variable
Parameters	**Description**	**NOTE**
D_R	Action(Direct = 0, Reverse = 1)	Boolean-type variable
F_FE	Failure Mode of the Final Element (Fail Close = 0, Fail Open = 1)	Boolean-type variable
K_P	Proportional Action	Floating point-type variable
K_I	Integral Action	Floating point-type variable
K_D	Derivative Action	Floating point-type variable
K_T	Anti-Reset Windup Action	Floating point-type variable

Feature description

IN is subtracted from SP, which produces a quantity called "error". This error passes through proportional, integral and derivative actions and is summed, and then, depending on the action, the value for OUT_MV can be as follows:

- If D_R=0 (action is direct), OUT_MV is equal to the output of the PID controller
- If D_R=1 (action is reverse) OUT_MV is equal to 100% minus the output of the PID controller

The value of OUT_MV is meant to be displayed in the SOS, but depending on the failure mode of the final element can be still modified:

- If F_FE=0 (final element is fail close), OUT_CD is equal to OUT_MV
- If F_FE=1 (final element is fail open), OUT_CD is equal to 100% minus OUT_MV

4.11.3 The logic shall follow the schematics below:

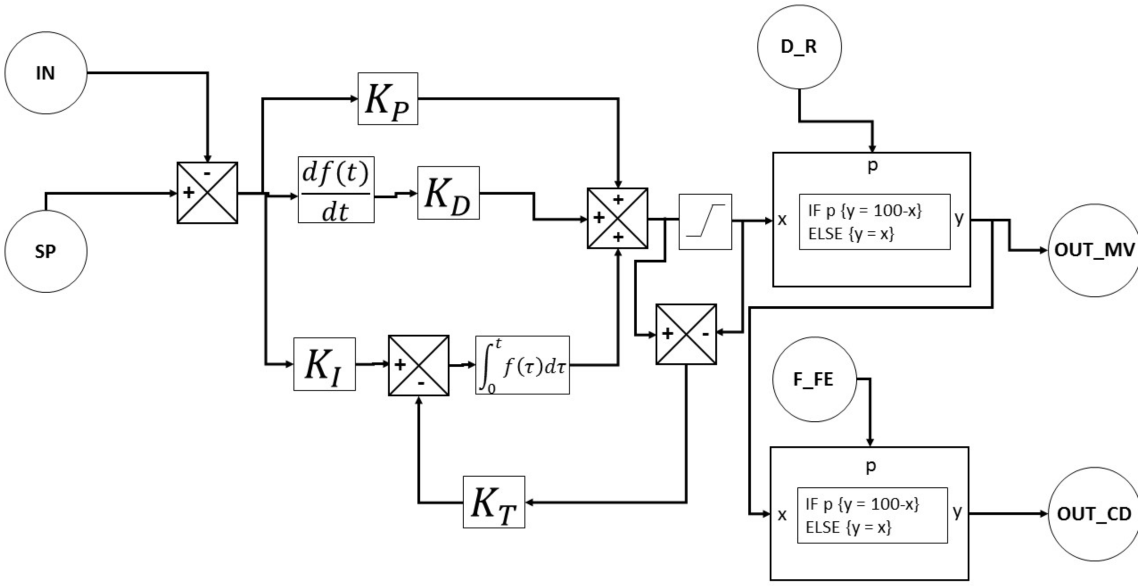


Figure 12 – PID control logic

4.11.4 The values of set-point, controller gains and other controller settings shall be kept in PLC retentive memory.

4.12 Electrical load control

4.12.1 The features described in the table below represent the minimal requirements related to logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application.

4.12.2 The control is dependent on electrical actuation type (EA) of each load. This document shows the typical controls for load types EA01 (electrical loads controlled and monitored by CSS/HCSS), EA02 (electrical loads monitored but not controlled by CSS/HCSS) and EA03 (electrical load not controlled nor monitored by CSS/HCSS). Electrical loads from type EA04 are not covered in this document, since all their control is done in the Package Control Panel.

4.12.3 In case of loss of power supply to a specific electrical load, this block shall receive a signal indicating the power supply unavailability. If this load is a motoric load, it shall go to “stop” state.

4.12.4 The classification for each load is shown in document I-ET-3010.00-5140-700-P4X-003 – ELECTRICAL REQUIREMENTS FOR PACKAGES FOR OFFSHORE UNITS

4.12.5 For EA01 Loads:

Inputs	Description	NOTE
HSH_R	Remote Start Command (from SOS)	Boolean-type variable
HSH_L	Local Start Command (From Field)	Boolean-type variable
ILH	Interlocking Logic Start Command (from CSS)	Boolean-type variable
HSL_R	Remote Stop Command (from SOS)	Boolean-type variable
HSL_L	Local/Emergency Stop Command (From Field)	Boolean-type variable
ILL	Interlocking Logic Stop Command (from CSS)	Boolean-type variable
PER_H	Permission to Start (from PMS)	Boolean-type variable
ESD_L	Emergency Shutdown Stop Command (from CSS)	Boolean-type variable
TMR	Time to start load after command, in milliseconds	Floating point-type variable
L_R	Local/Remote (from SOS, Remote = 0, Local = 1)	Boolean-type variable
Outputs	**Description**	**NOTE**
REQ_H	Request to Start (to PMS)	Boolean-type variable
OVR	Override Alarms	Boolean-type variable
OUT_H	Start Load (to Electrical System Controllers)	Boolean-type variable
OUT_LC	Stop Load (to Electrical System Controllers)	Boolean-type variable
OUT_LF	Stop Load (to MCC/CDC)	Boolean-type variable
Feature description		
A command from HSH_R or from ILH with L_R = 0 (remote) or a command from HSH_L with L_R = 1 (local) generate a REQ_H to PMS, requesting to start the load. After PMS gives its permission (PER_H), the start is authorized (for loads non-dependent on PMS, PER_H should stay equal to 1). Then, an override of interlocking (OVR, typically low pressure, that should not interlock during start-up of pump loads) is sent to Process Shutdown (PSD) or Hull Shutdown (HSD) PLCs. This override is configured in the PSD or HSD PLCs to last the amount of time needed to start the load. After TMR seconds, the OUT_H command is given to start the load. This time is to account for the time that PSD or HSD PLCs take to override the interlocking logic. To stop the load, a command from ESD_L or HSL_L stop the load directly in the MCC/CDC, cutting its power via output OUT_LF. HSL_R or ILL commands, with L_R = 0 (remote) stop the load through Electrical System Controllers, via output OUT_LC. These logic shall be made in the PSD/HSD controllers		

4.12.6 The logic for EA01 loads shall follow the schematics below:

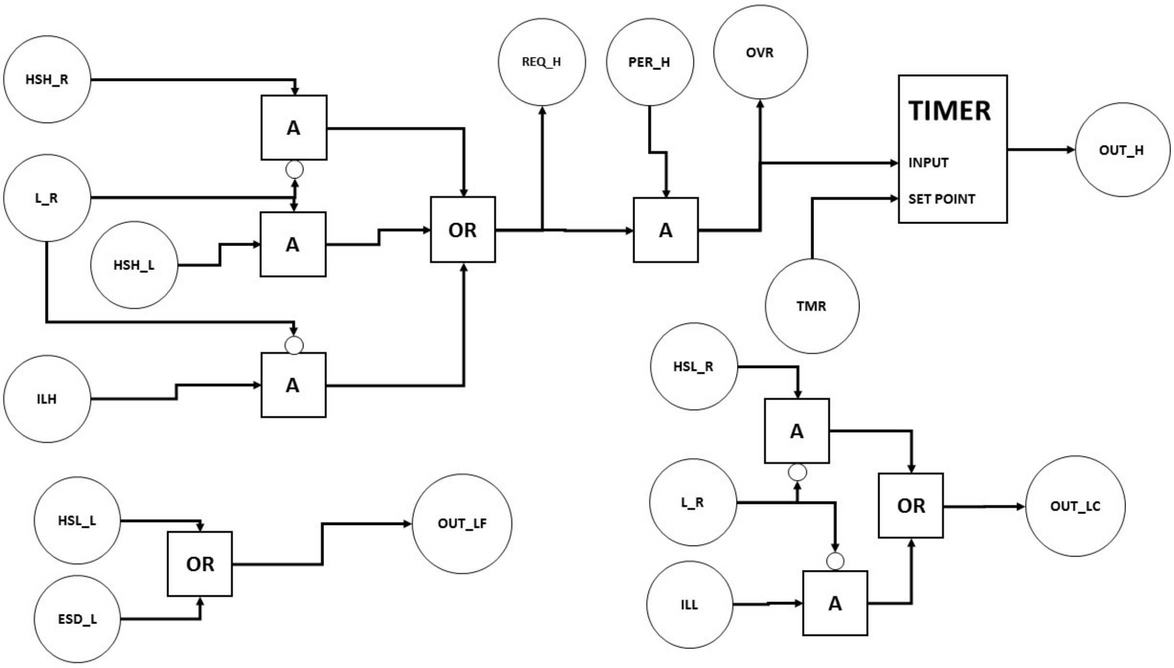


Figure 13 – EA01 loads logic

4.12.7 For EA02 and EA03 Loads:

Inputs	Description	NOTE
HSH_L	Local Start Command (From Field)	Boolean-type variable
HSL_L	Local/Emergency Stop Command (From Field)	Boolean-type variable
ESD_L	Emergency Shutdown Stop Command (from CSS)	Boolean-type variable
Outputs	**Description**	**NOTE**
OUT_H	Start Load (to Electrical System Controllers)	Boolean-type variable
OUT_LF	Stop Load (to MCC/CDC)	Boolean-type variable
Feature description		
A command from HSH_L starts the load via Electrical System Controllers (OUT_H) A command from HSL_L or ESD_L stop the load directly in the MCC/CDC, cutting its power via output OUT_LF.		

4.12.8 The logic for EA02 and EA03 loads shall follow the schematics below:

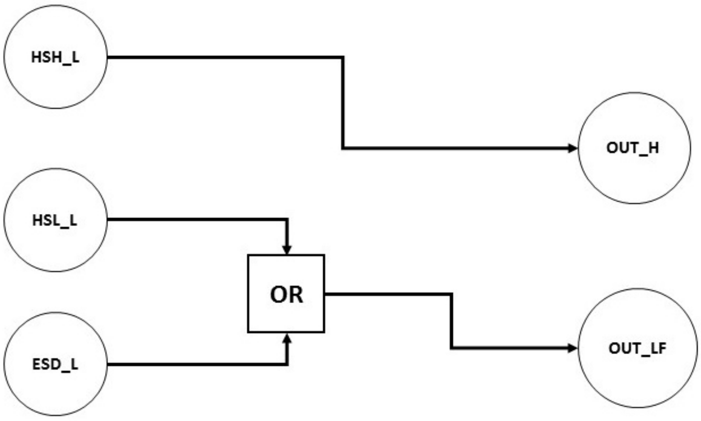


Figure 14 – EA-02 and EA-03 logic

4.13 Fire and Gas detection voting

4.13.1 Fire and Gas detection voting shall be made in accordance with DR-ENGP-M-I-1.3 – SAFETY ENGINEERING GUIDELINE.

4.14 First Event Logic

4.14.1 The features described in the table below represent the minimal requirements related to logic performance that shall be in the implementation/settings of the functional block. The minimum to be implemented is shown in the table below. Other variables can be required according to project and/or application.

Inputs	Description	NOTE
IN_I	Activated input from the ESD-initiating sensor I	Boolean-type variable
RES	ESD reset	Boolean-type variable
Outputs	**Description**	**NOTE**
OUT_I	Indication that sensor I is the first event	Boolean-type variable
Feature description		
Every ESD-initiating sensor shall be connected to this block. If a particular sensor is active and the respective ESD (or a higher ESD) is not active, a SR flip-flop is set. The output of this SR flip-flop is connected to the respective OUT_I. Since the “set” of the flip-flop is conditioned to the non-occurrence of the ESD, when a particular sensor sets its respective flip-flop, no other flip-flops will be set. Thus, only the value of the output corresponding to this sensor will be equal to “1”, being this sensor the first event. The RES input corresponds to a manual reset after the occurrence of an ESD level and it’s responsible for resetting all flip-flops.		

4.14.1 The logic for ESD-2, ESD-3P and ESD-3T first events are shown below, in figures 15, 16 and 17, respectively.

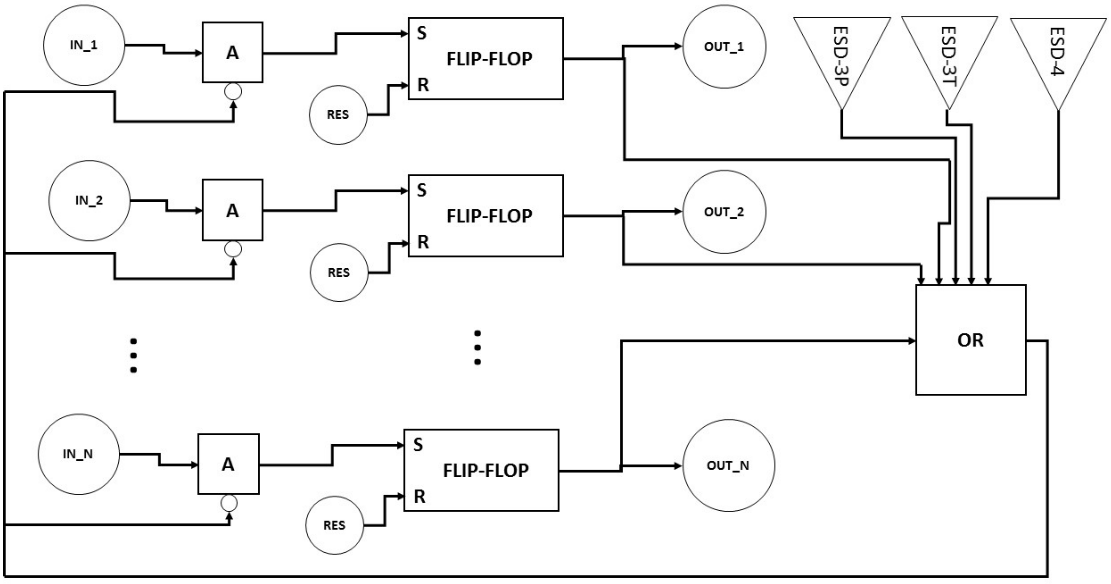


Figure 15 – First event logic for ESD-2

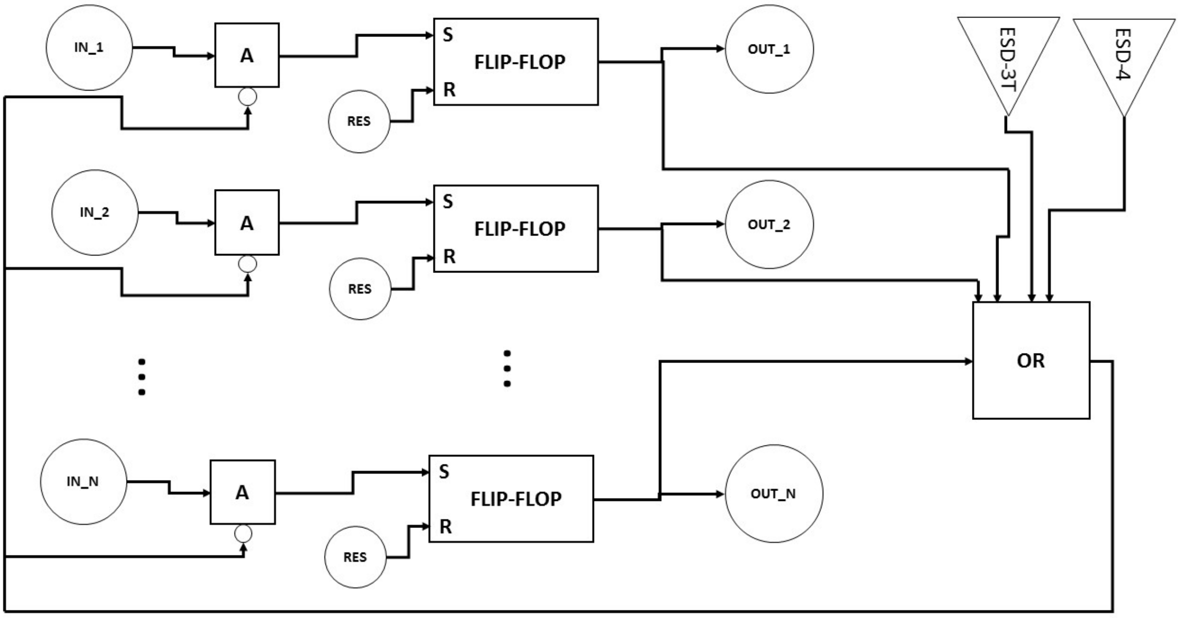


Figure 16 – First event logic for ESD-3P

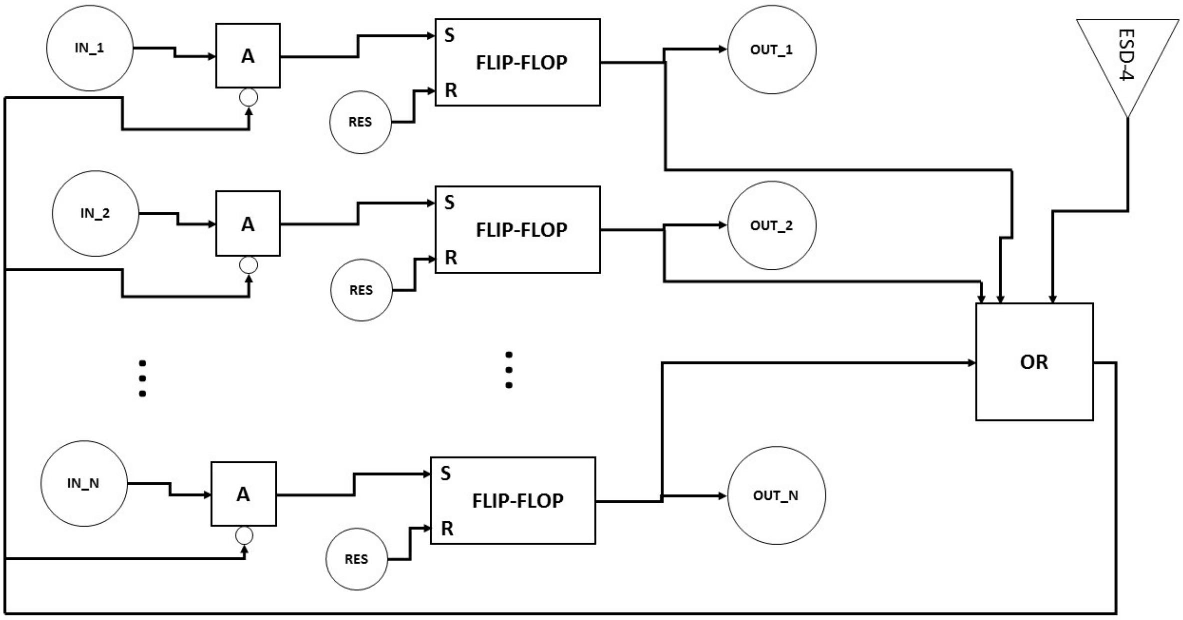


Figure 17 – First event logic for ESD-3T