DC302

smar
First in Fieldbus

FIELDBUS
REMOTE I/O

smar - DC302

FEV / 18

DC302

VERSION 3

FOUNDATION

DC302ME

smar

www.smar.com

Specifications and information are subject to change without notice.

Up-to-date address information is available on our website.

web: www.smar.com/contactus.asp

INTRODUCTION

Until all types of devices are available with FOUNDATION™ Fieldbus systems will have to be of a hybrid nature accepting both Fieldbus and conventional signals. A mixed traditional and fieldbus environment is inevitable during the transition to a Fieldbus technology. DC302 makes integration of Fieldbus and conventional I/O easy. Discrete devices such as pressure switches, push buttons, on/off valves, pumps and conveyors are integrated to the system over the FOUNDATION™ H1 field-level network using DC302. The DC302 remote I/O units can be distributed into the field where they are mounted close to the conventional devices without the need to run the conventional wiring to the control room. The DC302 is an integral part of SYSTEM302 but also it integrates into other systems supporting FOUNDATION™ Fieldbus.

The DC302 makes conventional analog and discrete inputs and outputs available using standard FOUNDATION™ Function blocks making the system homogenous and control strategy configuration easy as conventional I/O appears as if they were regular Fieldbus devices. Control loops are implemented consistently regardless of I/O being conventional or Fieldbus based. Only a single programming language has to be used.

The DC302 is a simple low-cost DIN-rail mounted unit. The DC302 is a single integrated easy to use piece of equipment including power, control, networking and I/O under one compact device requiring less panel space than other solutions.

An extensive function block library enables the DC302 to perform logic and regulatory control functions in the field integrating the control strategy with other H1 Fieldbus devices on the same network. Instantiable function blocks provide great flexibility in control strategy. Conventional discrete I/O now works together with pure Fieldbus devices on the same network and in the same loop. The DC302 is fully configured from the Syscon software in SYSTEM302 or any other FOUNDATION™ Fieldbus configuration tool. Function blocks provide logic such as AND, OR, NAND etc. as well as latches etc. Link master capability allows the DC302 to work as a backup LAS for greater availability of network communications.

The DC302 may be installed close to the sensors and actuators, thereby eliminating long wire runs and associated marshalling panels and cable trays for the conventional I/O, with subsequent savings further reducing overall system cost. Use DC302 to make it possible to distribute I/O at various locations in the field and connect them via H1 Fieldbus. DC302 is ideal to connect motor control centers, variable speed drives, and electrical actuators and motor operated valves to H1 Fieldbus.

Get the best result of the DC302 by carefully reading these instructions.

WARNING
This manual applies to equipment with a serial number from 2000 onwards.

NOTE
For previous DC302 versions (with plastic cowling), pick up equivalent manual.

Waiver of responsibility

The contents of this manual abides by the hardware and software used on the current equipment version. Eventually there may occur divergencies between this manual and the equipment. The information from this document are periodically reviewed and the necessary or identified corrections will be included in the following editions. Suggestions for their improvement are welcome.

Warning

For more objectivity and clarity, this manual does not contain all the detailed information on the product and, in addition, it does not cover every possible mounting, operation or maintenance cases.

Before installing and utilizing the equipment, check if the model of the acquired equipment complies with the technical requirements for the application. This checking is the user's responsibility.

If the user needs more information, or on the event of specific problems not specified or treated in this manual, the information should be sought from Smar. Furthermore, the user recognizes that the contents of this manual by no means modify past or present agreements, confirmation or judicial relationship, in whole or in part.

All of Smar's obligation result from the purchasing agreement signed between the parties, which includes the complete and sole valid warranty term. Contractual clauses related to the warranty are not limited nor extended by virtue of the technical information contained in this manual.

Only qualified personnel are allowed to participate in the activities of mounting, electrical connection, startup and maintenance of the equipment. Qualified personnel are understood to be the persons familiar with the mounting, electrical connection, startup and operation of the equipment or other similar apparatus that are technically fit for their work. Smar provides specific training to instruct and qualify such professionals. However, each country must comply with the local safety procedures, legal provisions and regulations for the mounting and operation of electrical installations, as well as with the laws and regulations on classified areas, such as intrinsic safety, explosion proof, increased safety and instrumented safety systems, among others.

The user is responsible for the incorrect or inadequate handling of equipments run with pneumatic or hydraulic pressure or, still, subject to corrosive, aggressive or combustible products, since their utilization may cause severe bodily harm and/or material damages.

The field equipment referred to in this manual, when acquired for classified or hazardous areas, has its certification void when having its parts replaced or interchanged without functional and approval tests by Smar or any of Smar authorized dealers, which are the competent companies for certifying that the equipment in its entirety meets the applicable standards and regulations. The same is true when converting the equipment of a communication protocol to another. In this case, it is necessary sending the equipment to Smar or any of its authorized dealer. Moreover, the certificates are different and the user is responsible for their correct use.

Always respect the instructions provided in the Manual. Smar is not responsible for any losses and/or damages resulting from the inadequate use of its equipments. It is the user's responsibility to know and apply the safety practices in his country.
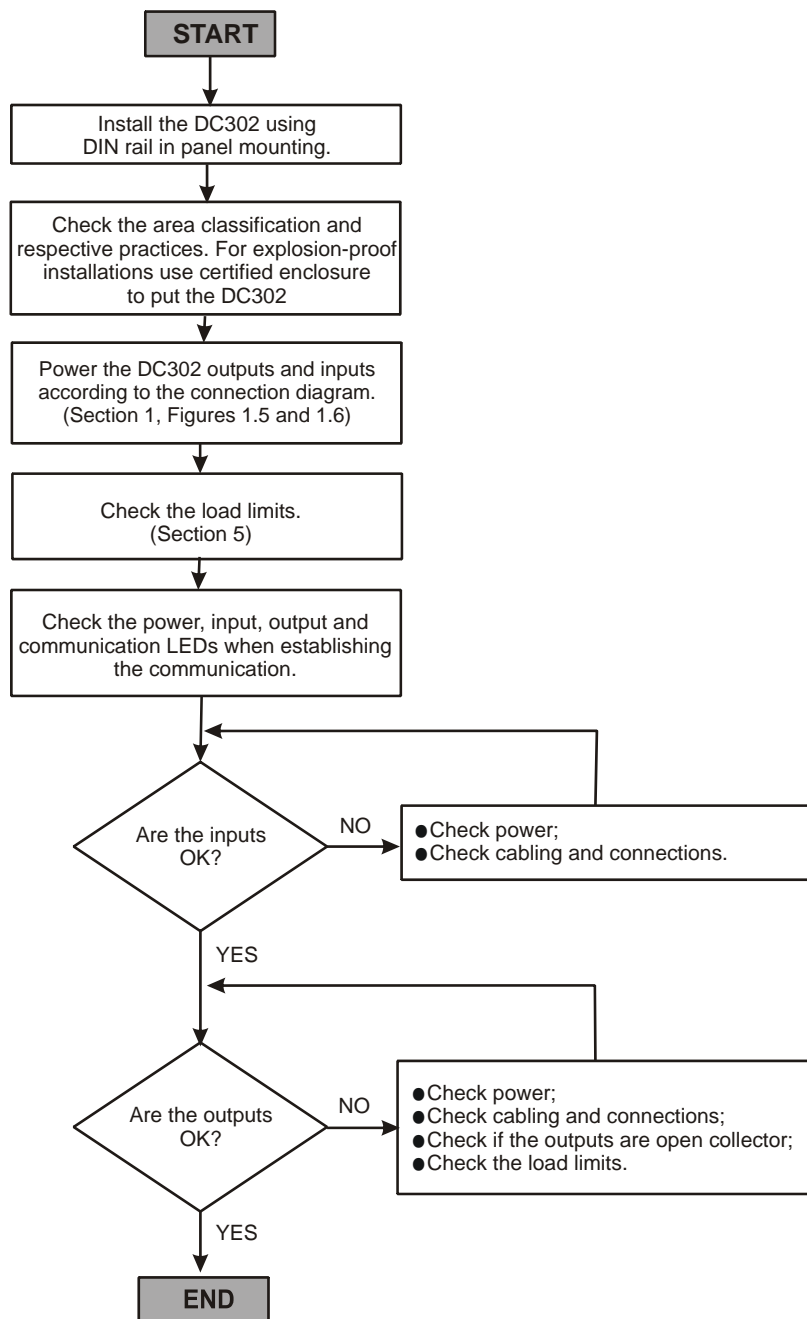
TABLE OF CONTENTS

Installation Flowchart

ATTENTION
Get the best results of the DC302 by carefully reading all instructions manual.

```
                    ┌──────────┐
                    │  START   │
                    └────┬─────┘
                         ▼
        ┌──────────────────────────────┐
        │   Install the DC302 using     │
        │   DIN rail in panel mounting. │
        └───────────────┬───────────────┘
                        ▼
        ┌──────────────────────────────┐
        │ Check the area classification │
        │ and respective practices. For │
        │ explosion-proof installations │
        │ use certified enclosure       │
        │ to put the DC302              │
        └───────────────┬───────────────┘
                        ▼
        ┌──────────────────────────────┐
        │ Power the DC302 outputs and   │
        │ inputs according to the       │
        │ connection diagram.           │
        │ (Section 1, Figures 1.5 and   │
        │ 1.6)                          │
        └───────────────┬───────────────┘
                        ▼
        ┌──────────────────────────────┐
        │   Check the load limits.      │
        │       (Section 5)             │
        └───────────────┬───────────────┘
                        ▼
        ┌──────────────────────────────┐
        │ Check the power, input,       │
        │ output and communication LEDs │
        │ when establishing             │
        │ the communication.            │
        └───────────────┬───────────────┘
```

Are the inputs OK? — NO → ● Check power; ● Check cabling and connections.

YES

Are the outputs OK? — NO → ● Check power; ● Check cabling and connections; ● Check if the outputs are open collector; ● Check the load limits.

YES

END

* More information in Section 1 from DC302 Operation, Maintenance and Instructions Manual.

Section 1

INSTALLATION

General

The overall accuracy of measurement and control depends on several variables. Although the Fieldbus Remote I/O has an outstanding performance, proper installation is essential, in order to maximize its performance.

Among all factors, which may affect the accuracy, environmental conditions are the most difficult to control. There are, however, ways of reducing the effects of temperature, humidity and vibration.
Locating the Fieldbus Remote I/O in areas protected from extreme environmental changes can improve its performance.
In warm environments, the Fieldbus Remote I/O should be installed to avoid, as much as possible, direct exposure to the sun. Installation close to lines and vessels subjected to high temperatures should also be avoided.

Use of sunshades or heat shields to protect the Fieldbus Remote I/O from external heat sources should be considered, if necessary.

Humidity is fatal to electronic circuits. In areas subjected to high relative humidity, the protection cover must be provided.
For details of mounting, please, refer to Figure 1.1 and Figure 1.2

Mounting

Use DIN rail (TS35-DIN EN 50022 or TS32-DIN EN50035 or TS15 DIN EN50045), as shown in Figure 1.1 – Mechanical Mounting. The DC302 can optionally be supplied preinstalled in an enclosure ready for field mounting.
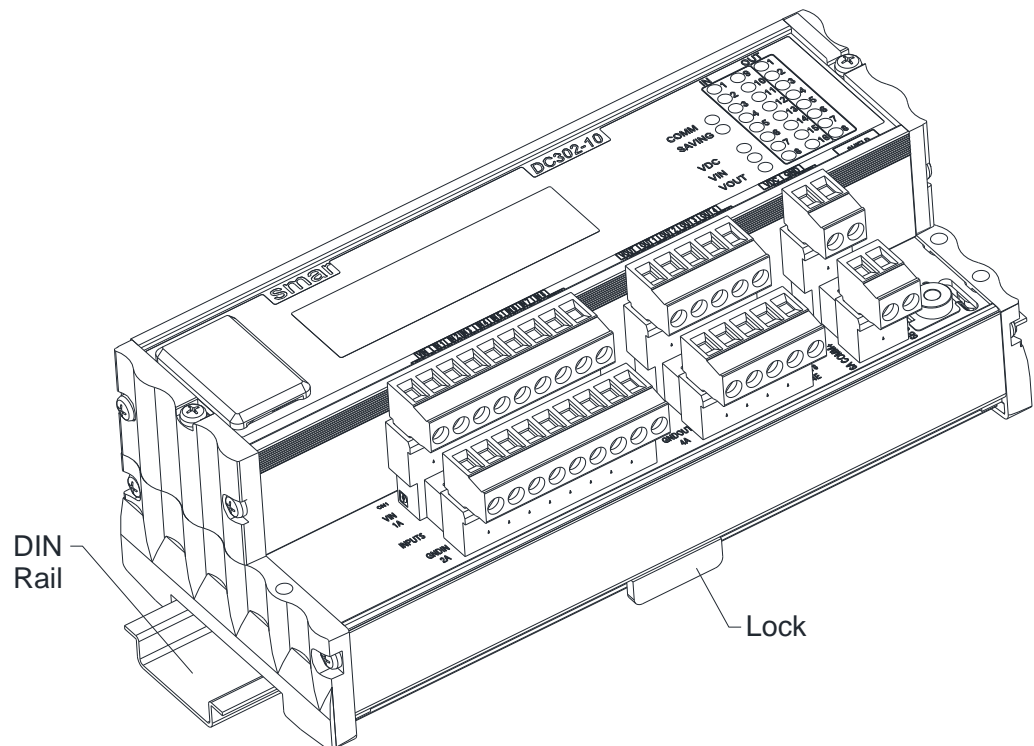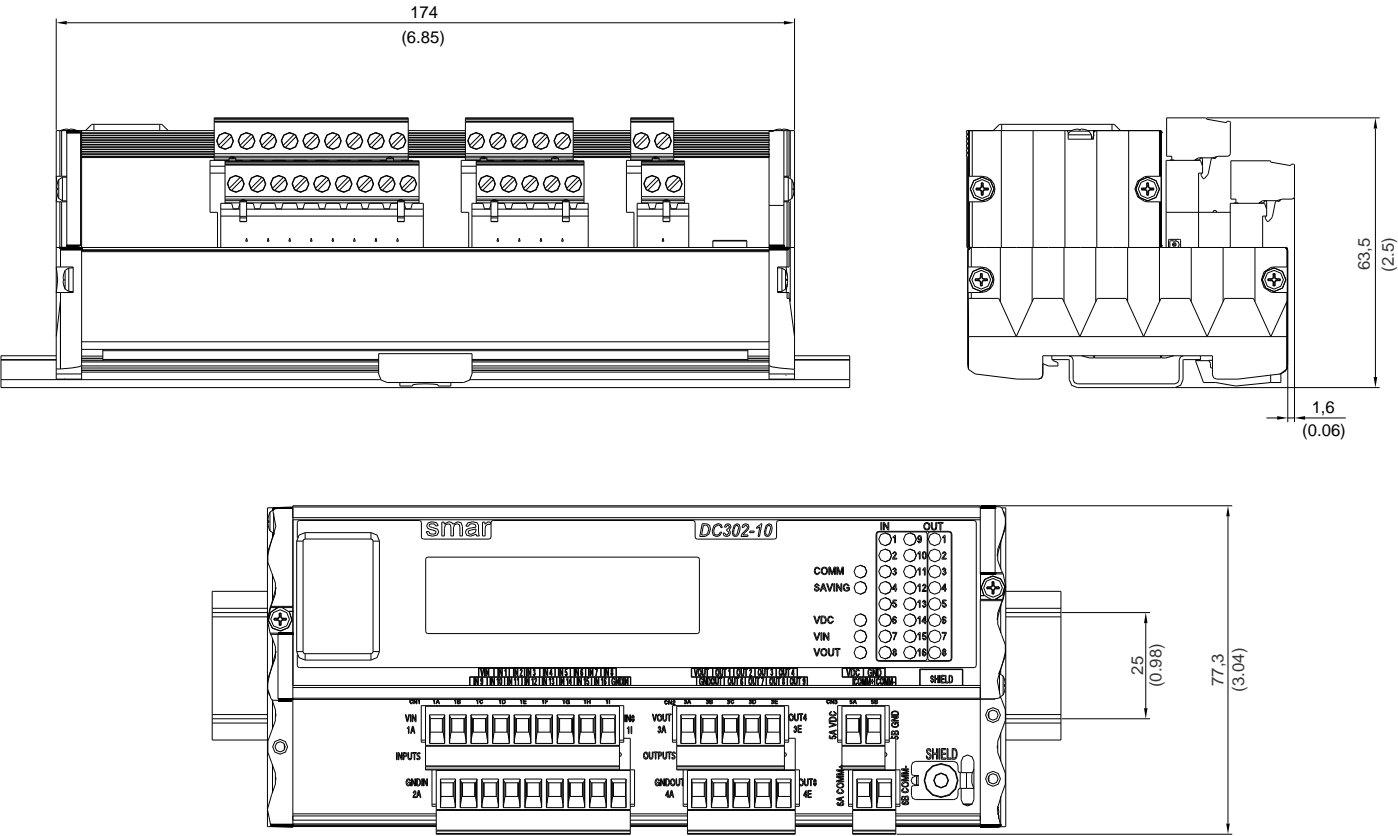

Figure 1.1- Mechanical Mounting

NOTE
The measurements are in mm.

Electric Wiring

Access the wiring block by the front View with label for inputs, outputs power supply and bus connection. The connections are made using the screws.
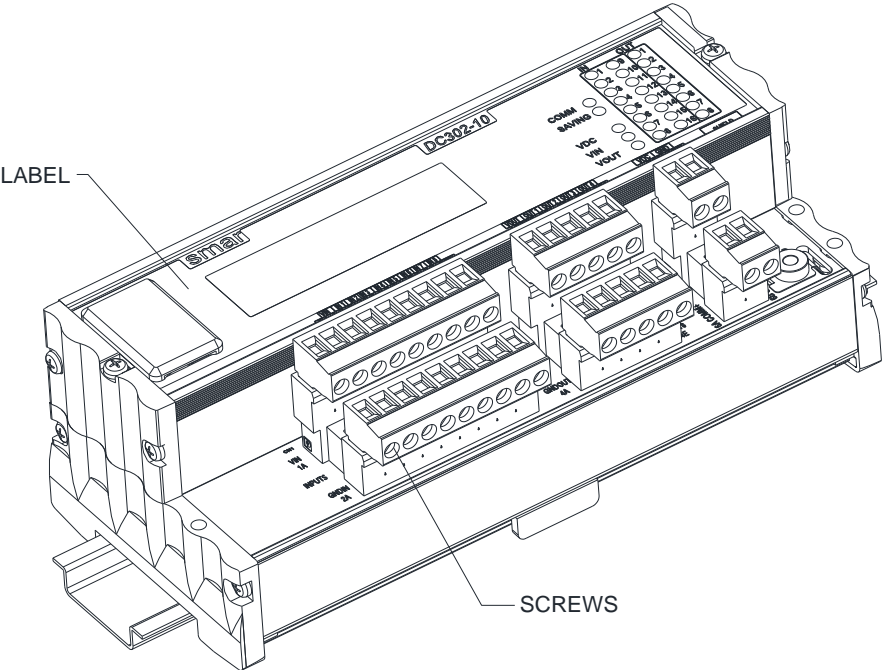

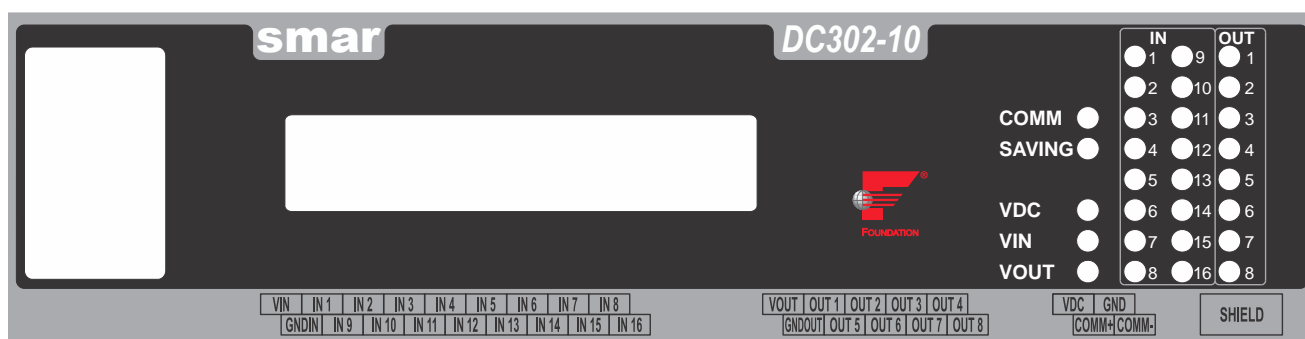*Figure 1.3 - Terminal Block Connections*

Figure 1.4 – Label DC302

The following table describes the terminal numbers for DC302.

Upper (U)	Lower (U)	Remark
VIN (1A)	GNDIN (2A)	Auxiliary power for inputs
IN1 (1B)	IN9 (2B)	
IN2 (1C)	IN10 (2C)	
IN3 (1D)	IN11 (2D)	
IN4 (1E)	IN12 (2E)	
IN5 (1F)	IN13 (2F)	**Digital Inputs**
IN6 (1G)	IN14 (2G)	
IN7 (1H)	IN15 (2H)	
IN8 (1I)	IN16 (2I)	
VOUT (3A)	GNDOUT (4A)	Auxiliary power to drive outputs
OUT1 (3B)	OUT5 (4B)	
OUT2 (3C)	OUT6 (4C)	**Digital Outputs**
OUT3 (3D)	OUT7 (4D)	
OUT4 (3E)	OUT8 (4E)	
VDC (5A)		Main power
GND (5B)		
	COMM+ (6A)	FOUNDATION fieldbus™ communication signal.
	COMM- (6B)	

Table 1.1 - Terminal Block Connections

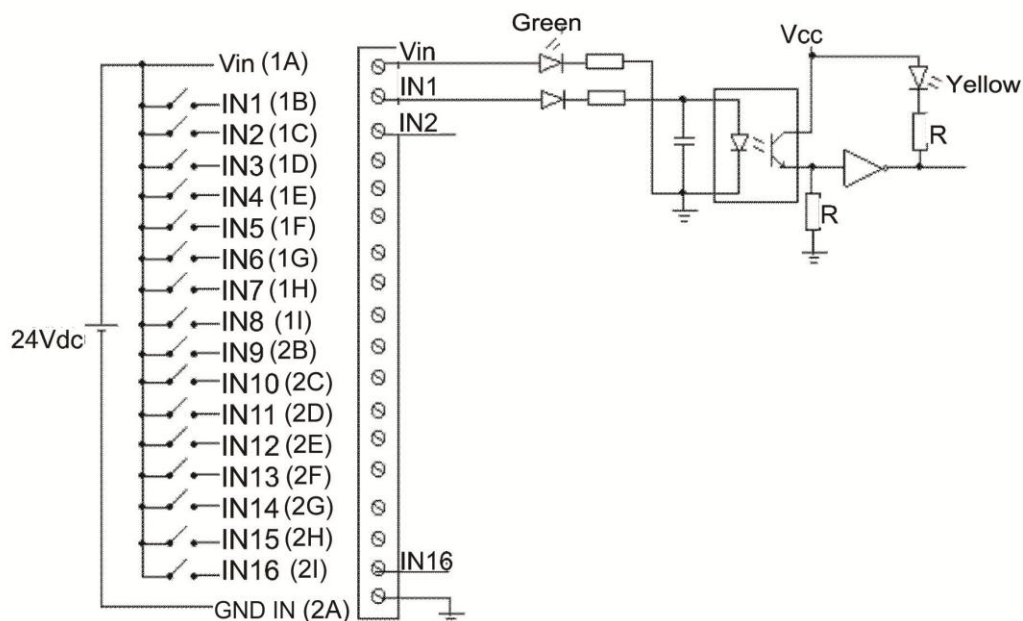The used connections should be plugged accordingly. For examples, please see the Figure 1.5 and Figure 1.6.

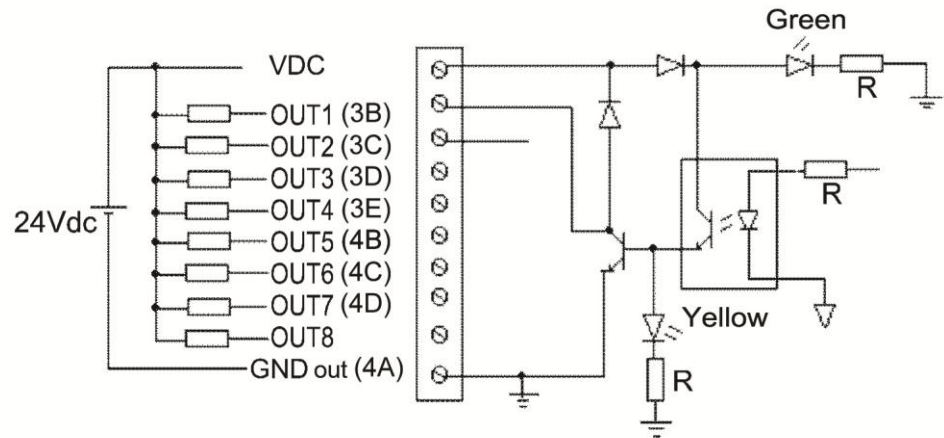*Figure 1.5 – Example of Input Connections*

Figure 1.6 – Example of Output Connections

The DC302 is a non bus-powered device. The DC302 uses a 31,25 Kbit/s voltage mode option for the physical signaling. Various types of Fieldbus devices may be connected on the same bus, being bus-powered or non-bus-powered. When bus-powered, the devices must use the same signaling. Up to 16 devices can be connected in parallel along the same fair of wires.

In hazardous area, the number of devices may be limited by intrinsically safe restrictions.

The **DC302** is protected against reverse polarity, and can withstand ±35 VDC without damage.

NOTE
Please refer to the General Installation, Operation and Maintenance Manual for more details.

WARNING

HAZARDOUS AREAS

In hazardous zones with intrinsically safe or non-incendive requirements, the circuit entity parameters and applicable installation procedures must be observed.

Cable access to wiring connections is obtained by the two conduit outlets. Conduit threads should be sealed by means of code-approved sealing methods.

Topology and Network Configuration

Bus topology (See Figure 1.7 - Bus Topology) and tree topology (See Figure 1.8 - Tree Topology) are supported. Both types have a trunk cable with two terminations. The devices are connected to the trunk via spurs. The spurs may be integrated in the device giving zero spur length. A spur may connect more than one device, depending on the length. Active couplers may be used to extend spur length.

Active repeaters may be used to extend the trunk length.

The total cable length, including spurs, between any two devices in the Fieldbus should not exceed 1900m.

The connection of couplers should be kept less than 15 per 250m.

⚠️ WARNING

Power Supplies

If there are requirements for power supply isolation between inputs and outputs, it is recommended to use at least two power supplies, one for inputs and another one for outputs and VDC.

If the application does not require isolation between inputs and outputs, only one power supply could be used for inputs, outputs and VDC.

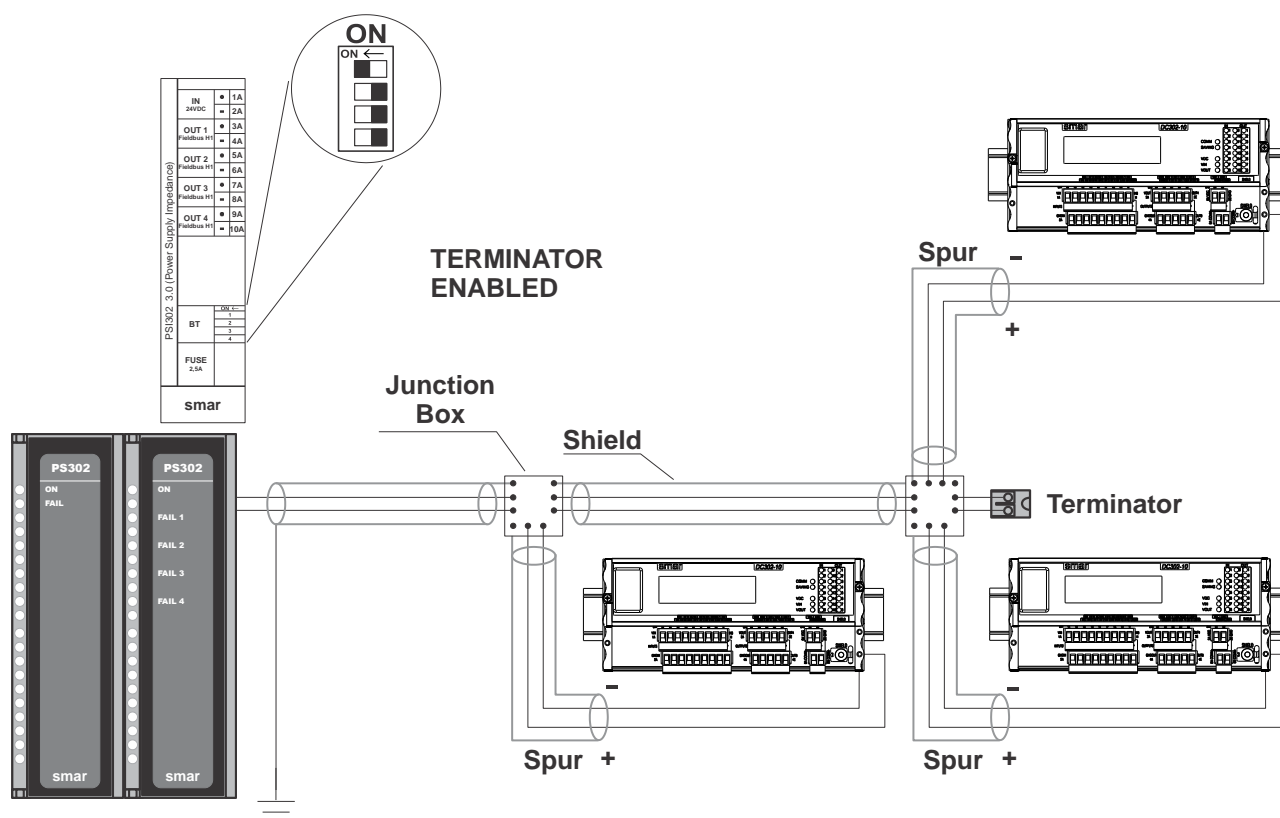Inputs and outputs are optically isolated from each other.

Figure 1.7 - Bus Topology
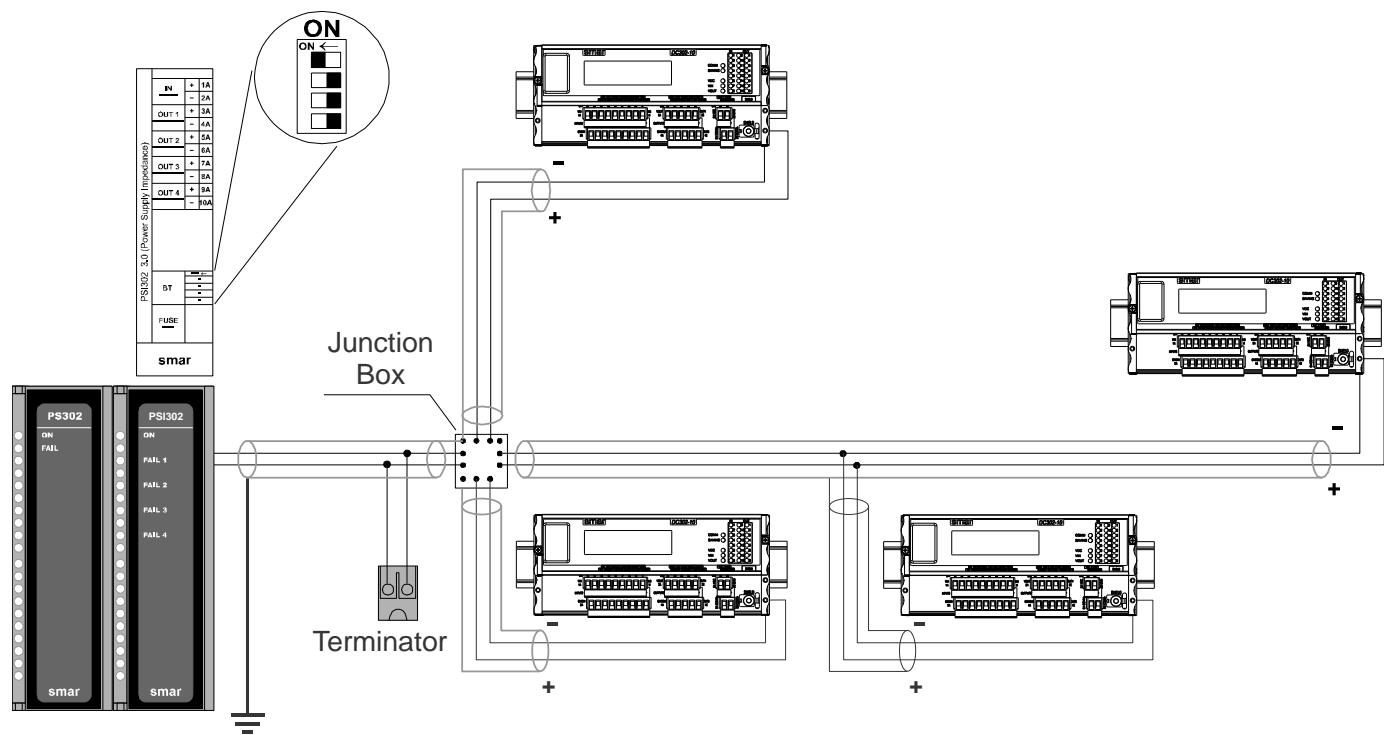

Figure 1.8 - Tree Topology

General System

According to the figure below, we can see a general network topology where the DC302 is integrated in a simple Fieldbus network.
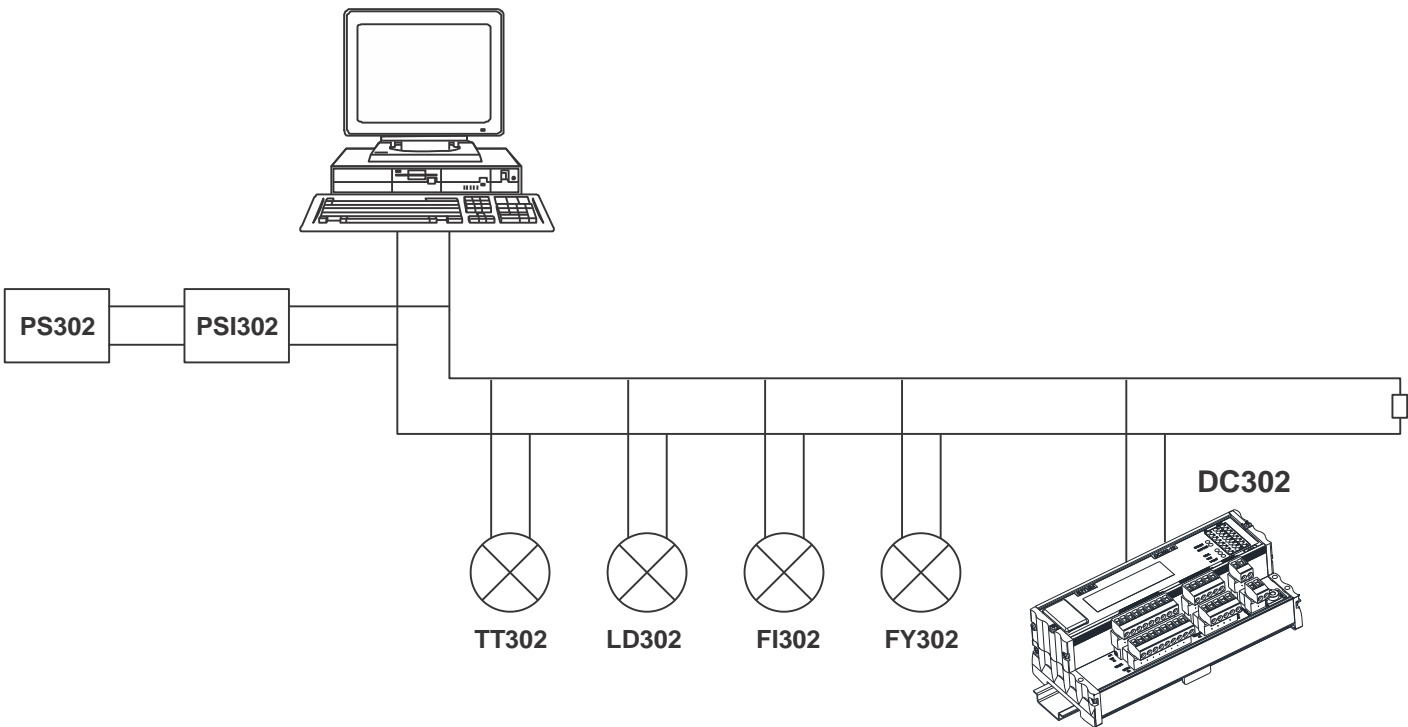

Figure 1.9 – DC302 and a general Fieldbus System

OPERATION

The **DC302** accepts up to 16 optically isolated inputs and up to 8 open collector outputs. It is therefore ideal for interfacing exisisting discrete points to a Fieldbus system.

An extensive Function Blocks library enables the DC302 to execute logic and functions of regulatory and descrete control integrated via H1 Bus. Function Blocks provide great flexibility to control strategies.

The conventional discrete I/Os work together with Fieldbus devices integrated in the same network and in the same control loop.
Output function blocks include standard Foundation safety mechanism in case of failures.

Inputs and Outputs are isolated from each other and are accessed via communication network through the function blocks channel. The leds are used to indicae the I/Os status. The use of Foundation™ Functional Blocks make the system homogeneous in a way that the device with convencional discrete and analog I/Os can be available in order to make the control strategies configuration easy.

Functional Description – Electronics

Refer to the block diagram (See Figure 2.1 – *DC302 Block Diagram*). The function of each block is described below.

(CPU) Central Processing Unit, FRAM
The CPU is the intelligent part of the **DC302**, being responsible for the management and operation of the execution block, self-diagnosis and communication. The program and the temporary data are stored in a FRAM memory. In the absence of energy, the data stored in the FRAM is not lost. The FRAM memory also stores the non-volatile data that will be used later. Examples of such data are: calibration, configuration and identification data.

Communication Controller
It monitors line activity, modulates and demodulates the signal from network line.

Power Supply
Takes power of the loop-line to power the Discrete Controller circuitry.

Factory Reset
The factory reset inscription can be found on the superior left side of the DC302 housing. In order to accomplish this operation, simply short-circuit the contacts on the circuit board, turn-on the DC302 in this short-circuit condition, and keep the key until the saving led goes to on state.

They are two mechanical contacts to perform the factory reset.

Input Latches
They are latches to hold the condition of inputs.

Otput Latches
They are latches to hold the condition of outputs.

Optical Isolation
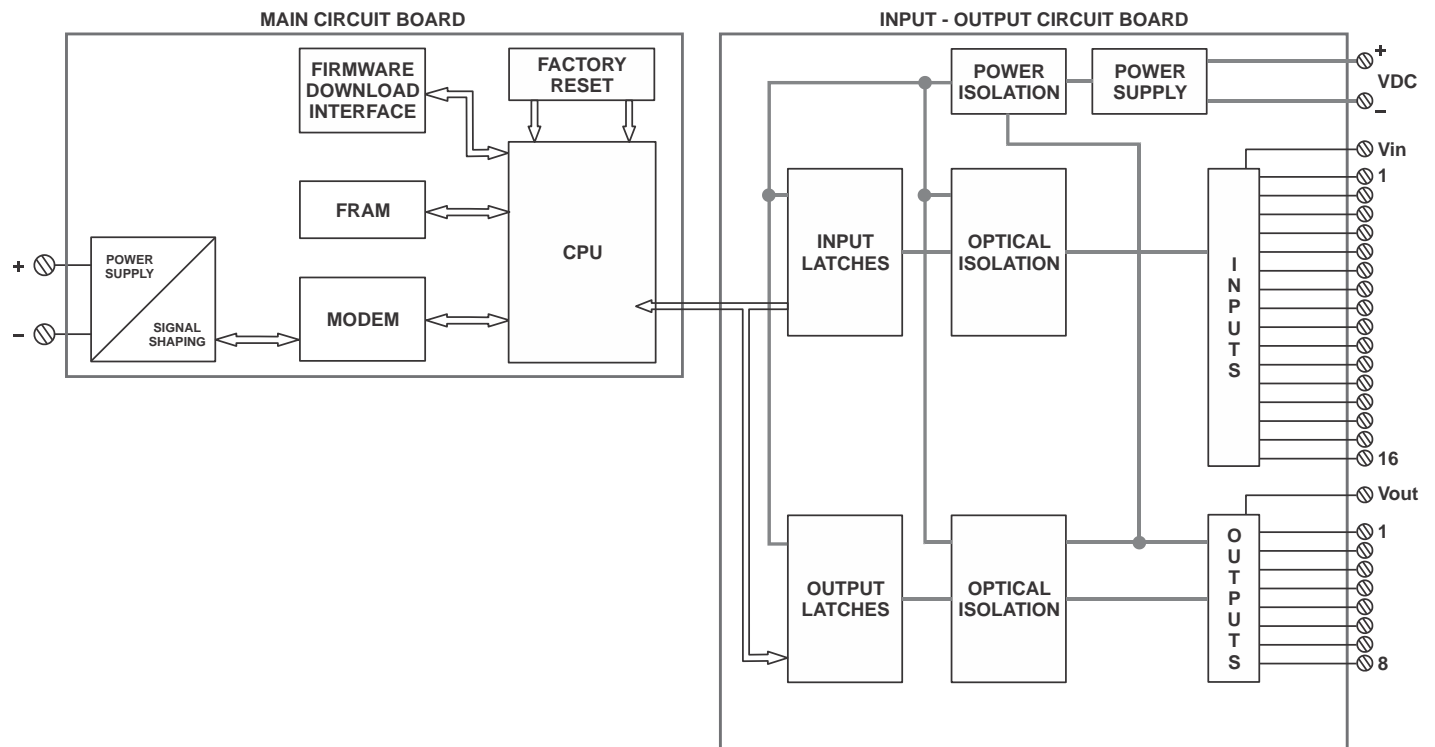Optical isolation for inputs and outputs.

Figura 2.1 – DC Block Diagram

CONFIGURATION

One of the many advantages of Fieldbus is that device configuration is independent of the configuration tool. The **DC302** may be configured by a third party terminal or operator console.

The **DC302** has several Function Blocks built in, such as Flip-Flop and Edge Trigger, Analog Alarm, Timer and Logic, Discrete Input, Discrete Output, Multiple Discrete Input, Multiple, Discrete Output, Arithmetic, Input Select, PID controller, PID Step and Flexible Function Block.

Function Blocks are not covered in this manual. For explanation and details of function blocks, see the *Function Blocks Manual.*

The DC302 can share its Function Blocks with other connected devices using the SYSCON.

For explanation and details for using SYSCON, please see the *SYSCON Manual.*

NOTE
(1) Do not use DIs together with MDIs and DOs together with MDO blocks.
(2) Do not instantiate FFB block together with MDI, this causes channel conflict.
(3) From the serial number 2000 the DC302 comes with 8 DI blocks and 8 DO blocks already instantiated.

The firmware version of the equipment must be used with the corresponding version of DD/CFF.

FIRMWARE VERSION	DD VERSION
3.60	0803

Connecting physical signals to Digital Input Block

The DI block takes the discrete input data, selected by channel number, and makes it available to other function blocks at its output.
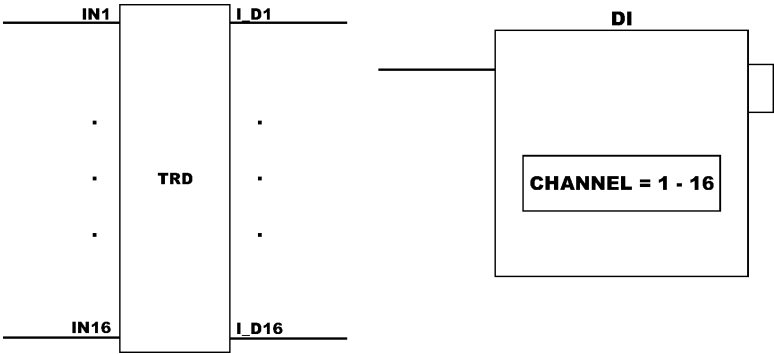For details, please see the Function Block Manual.

Figure 3.1 - DC302 and DI Block connections

Connecting physical signals to Digital Output Block

The DO block converts the value in SP_D to something useful for the hardware through the CHANNEL selection.
For details, please see the Function Blocks Manual.

Figure 3.2 - DC302 and DO Block connections

Connecting physical signals to Multiple Digital Input Block

One MDI block makes available for the FF network eight discrete variables of the I/O subsystem through its eight output parameters OUT_D1 through OUT_D8. Status indication in the output parameters OUT_Dx depends on the I/O. For example, if there is individual detection of sensor failure, it will be indicated in the status of related OUT_Dx parameter. Problem in the interface to the I/O subsystem will be indicated in the status of all OUT_Dx as BAD – Device Failure.
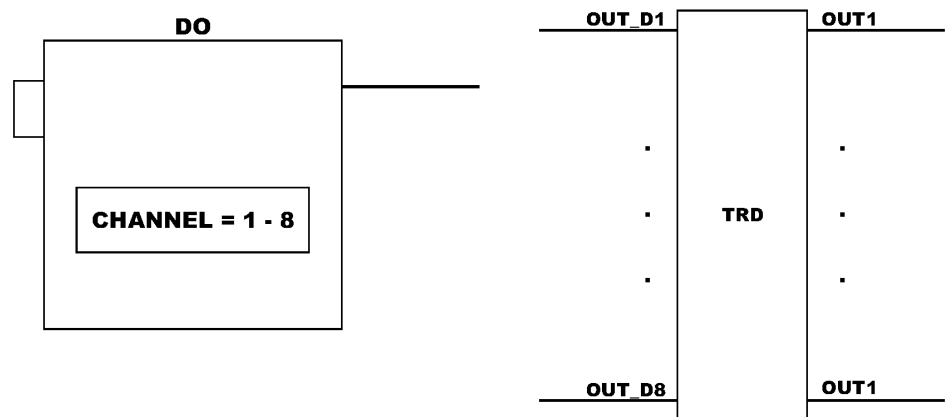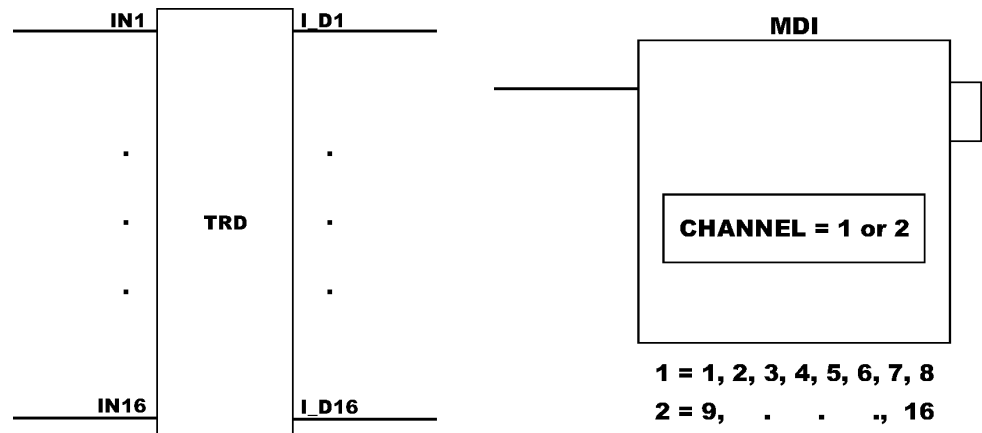For details, please see the Function Block Manual.

Figure 3.3 - DC302 and MDI Block connections

Connecting physical signals to Multiple Digital Output Block

The MDO block makes available to the I/O subsystem its eight input parameters IN_D1 through IN_D8.

This function block has the same fault state characteristics as the DO block. It includes option to hold the last value or go to a preset value when fault state activates individual preset values for each point, besides a delay time to go into the fault state. The actual mode will be LO only due to the resource block, otherwise bad status in input parameter and configuration of MO_STATUS_OPTS will not affect the mode calculation. However the functionality of fault state will be done only for that input parameter. The parameter FSTATE_STATE shows which points are in fault state active.

For details, please see the Function Block Manual.
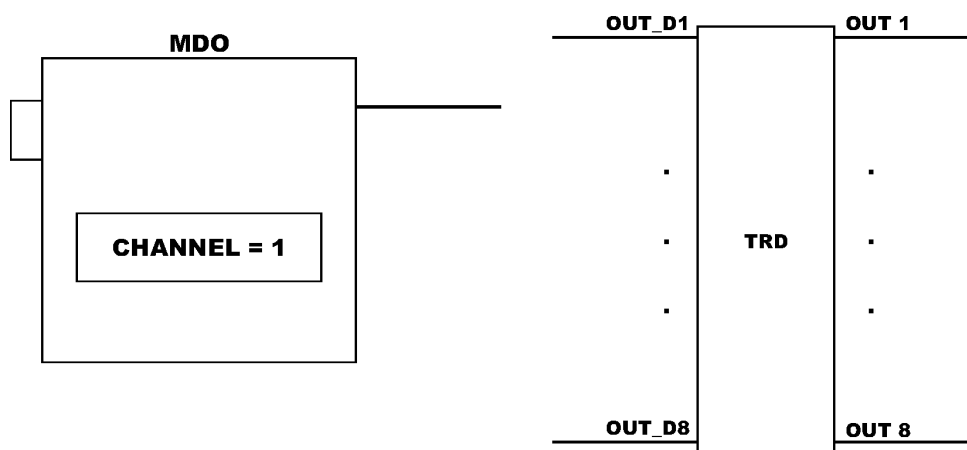
Figure 3.4 - DC302 and MDO Block connections

Connecting physical signals to PID Step

A Step Control Output block is used most commonly, when the final control element has an actuator driven by an electric motor. The final control element is positioned by rotating the motor clockwise or counterclockwise, which is accomplished by activating a discrete signal for each direction. A control valve, for example, needs a signal to open and another to close. If none of the signals is present, the valve stem would stay at the same position. Fieldbus actuators and switch gears are the transducer blocks of this block. For details, please see the Function Block Manual.
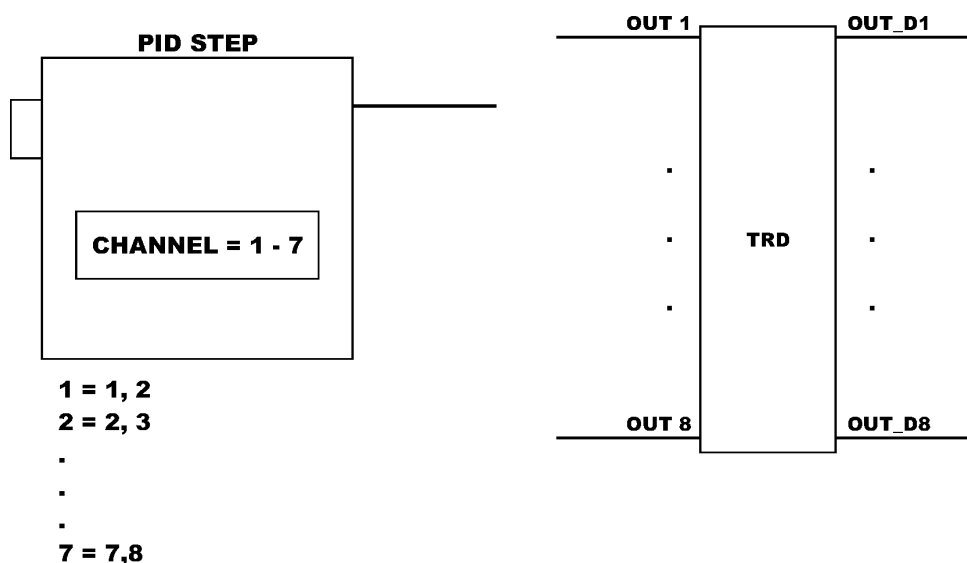
Figure 3.5 - DC302 and PID Step Block connections

Simulation Jumper

To perform the block simulation user must enable the *jumper simulation* first, which should be in the **ON** in DC302. See how to enable the hardware in the figure below. Then enable a DI or DO block on configuration software. Refer to Functional Blocks manuals: "Library A" and "Library B".

The SIMULATE parameter is used for the diagnostics and checking purposes. When active, transducer value and status will be replaced by the simulated value and status.
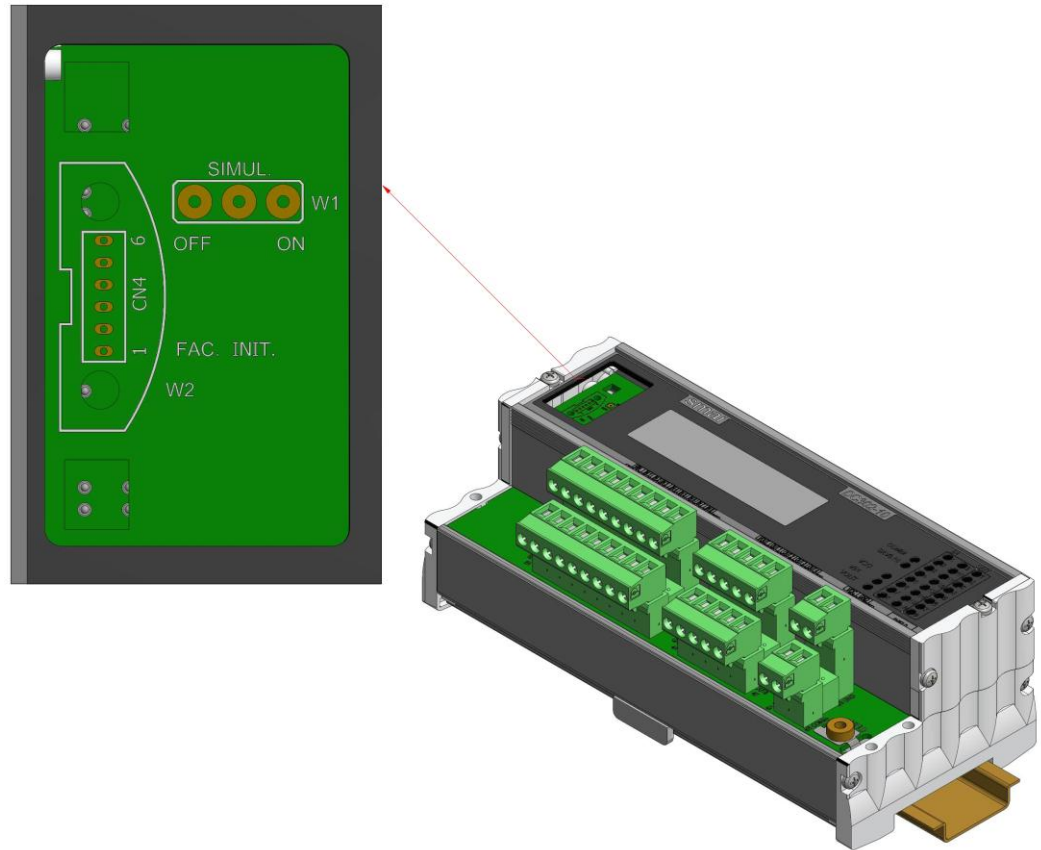
DC302 top view with detail of Simulation Jumper enabled in the "ON" position.

NOTE
The SIMULATE function can be disabled either by software in the SIMULATE parameter or hardware through the jumper in the **OFF** position. . Refer to Function Blocks manuals: "Library A" and "Library B for configuration details. See items: • Simulate • Input Function Block: DI - Discrete Input • Output Function Block: DO - Discrete Output.

Examples of Applications

Application 1: From the computer the input and output can be manipulated.
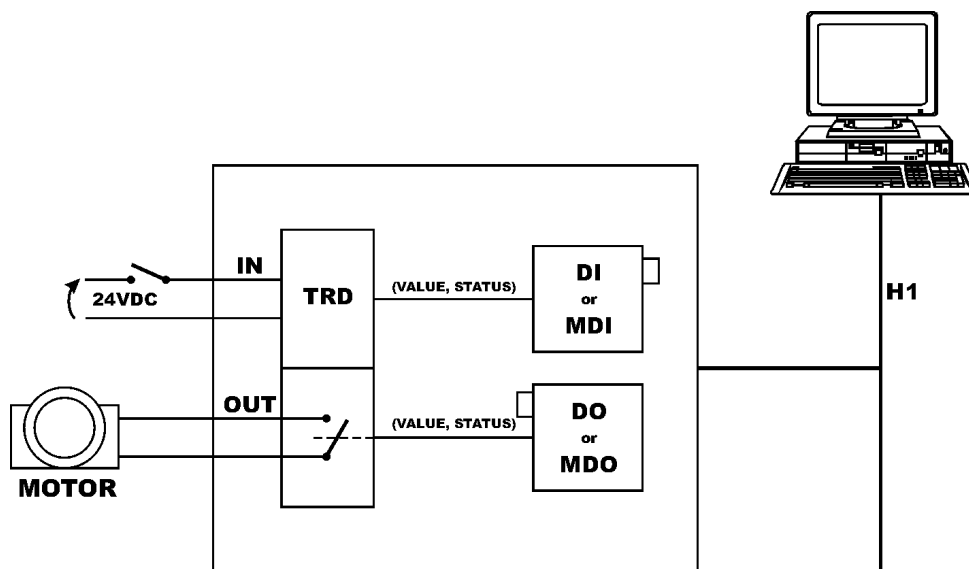
Figure 3.6 - DC302 – Appplication 1

Application 2: Distributed control (Level limit will start a motor, a pump, or open/close an on/off valve).
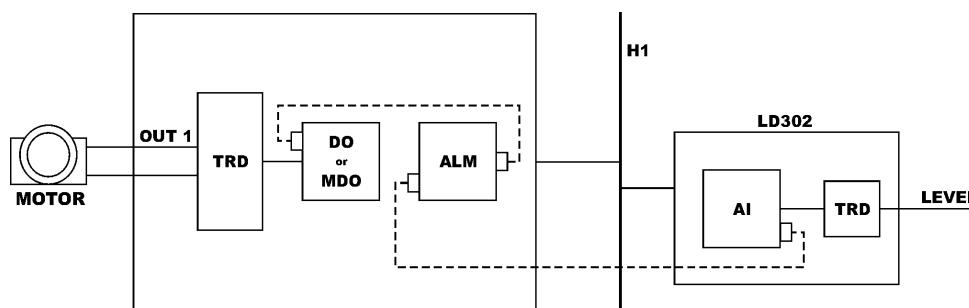
Figure 3.7 - DC302 – Appplication 2

Application 3: Distributed control (PID step).
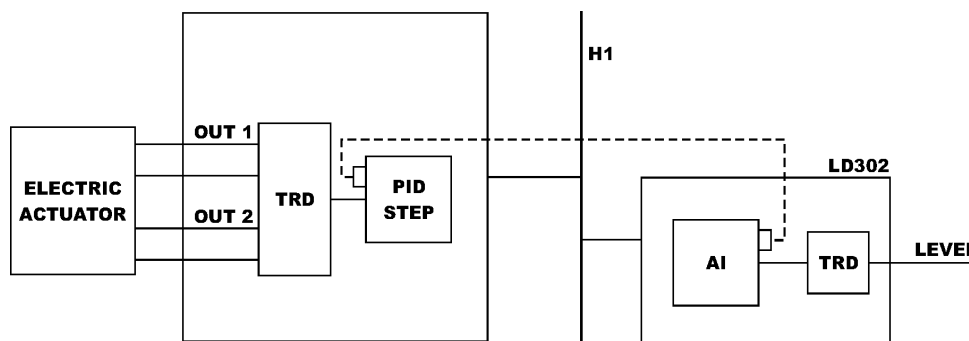
Figure 3.8 - DC302 – Appplication 3

Application 4: Distributed discrete control using TIME & FFET Function Blocks.
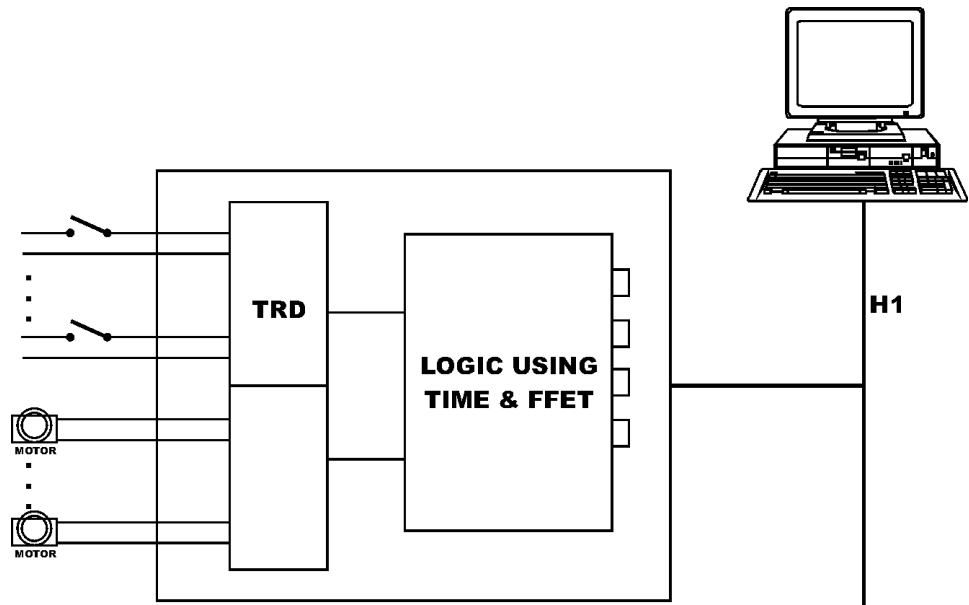
Figure 3.9 - DC302 – Appplication 4

Application 5: General application for DC302.
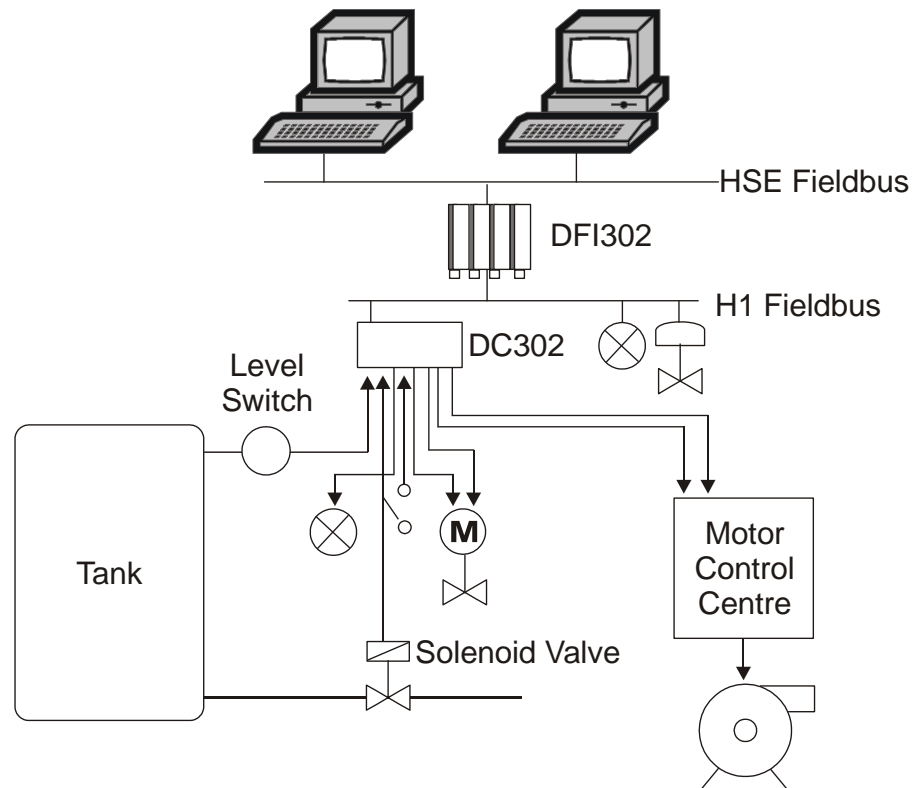
Figure 3.10 - DC302 – Appplication 5

Flexible Function Block

Description

The FFB block can receive up to 8 discrete input variables from the FF network through the IN_D1 to IN_D8 parameters and also make available to the FF network 8 discrete output variables through the OUT_D1 to OUT_D8 parameters. It can receive up to 16 discrete input variables from its hardware inputs (HW_IN) and also make available 8 discrete outputs through its hardware (HW_OUT).

Status indication for the inputs depends on the I/O subsystem. Status indication for the outputs depends on the block calculation

The FFB block provides logic such as AND, OR, XOR and NOT and functions such as Timer On-Delay, Timer Off-Delay, Timer Pulse, Pulse Counter Down (CTD), Pulse Counter Up (CTU), RS Flip-Flop and SR Flip-Flop. The logic is done using the 8 discrete variables available for the FF network (OUT_Dx), the 8 input parameters from the FF network (IN_Dx), the 16 input discrete variables from DC302 hardware(HIN), the 8 output discrete variables from DC302 hardware (HOUT), the failsafe (FSx) values and the auxiliary bit variables (AUX's).

BLOCK_ERR
The BLOCK_ERR of the FFB block will reflect the following causes:
- Block Configuration Error – the configuration error occurs when there is error in the logic line. It is indicated by the ERROLINE and the ERROCODE parameters.
- Input failure – When failure occurs in the input power supply.
- Output failure – When failure occurs in the output power supply.
- Out of Service – When the block is in O/S mode.

Status Handling
The status of OUT_Dx will be the following if the BLOCK_ERR indicates:
- Other – Bad : Configuration Error
- Input failure – Bad : Device Failure
- Power up – Bad : Device Failure

In the logic, a status that is greater or equal to 0x80 is considered to be true and a status that is less than 0x80 is considered to be false.

Supported Modes
O/S, MAN and AUTO.
The changes on the Logic Lines and its parameters depend on the selection of CHANGE_OPTION

Schematic

Parameters

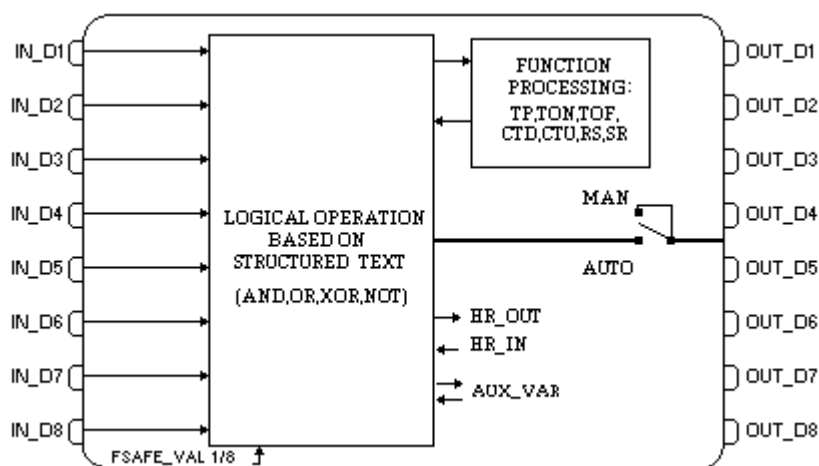| Idx	Parameter	Data Type/ (Length)	Valid Range/ Options	Default Value	Units	Store/ Mode	Description
1	ST_REV	Unsigned16		0	None	S/RO	The revision level of static data associated with the function block.
2	TAG_DESC	OctString(32)		Spaces	Na	S	The user description of intended application of the block.
3	STRATEGY	Unsigned16		0	None	S	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
4	ALERT_KEY	Unsigned8	1 to 255	0	None	S	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
5	**MODE_BLK**	**DS-69**		**O/S**	**Na**	**S**	**The actual, target, permitted and normal modes of the block.**
6	**BLOCK_ERR**	**Bitstring(2)**			**E**	**D / RO**	**This parameter reflects the error status associated with the hardware and software components associated with the block. It is a bit string, so multiple errors may be shown.**
7	PI_POINTER	Unsigned32		0	None	S	Index of the PI associated to the function block or resource. An index of 0 indicates that there is no PI associated to the function block or resource.
8	CONTENTS_REV	Unsigned32		0	None	S	This attribute indicates the revision level of the FFB algorithm. The low order 16 bits contain the minor revision level and the upper 16 bits contain the major revision level.
9	**IN_D1**	**DS-66**				**D**	**Discrete Input 1 for the calculation block.**
10	**IN_D2**	**DS-66**				**D**	**Discrete Input 2 for the calculation block.**
11	**IN_D3**	**DS-66**				**D**	**Discrete Input 3 for the calculation block.**
12	**IN_D4**	**DS-66**				**D**	**Discrete Input 4 for the calculation block.**
13	**IN_D5**	**DS-66**				**D**	**Discrete Input 5 for the calculation block.**
14	**IN_D6**	**DS-66**				**D**	**Discrete Input 6 for the calculation block.**
15	**IN_D7**	**DS-66**				**D**	**Discrete Input 7 for the calculation block.**
16	**IN_D8**	**DS-66**				**D**	**Discrete Input 8 for the calculation block.**
17	FSTATE_VAL_D1	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 1.
18	FSTATE_VAL_D2	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 2.
19	FSTATE_VAL_D3	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 3.
20	FSTATE_VAL_D4	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 4.
21	FSTATE_VAL_D5	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 5.
22	FSTATE_VAL_D6	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 6
23	FSTATE_VAL_D7	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 7.
24	FSTATE_VAL_D8	Unsigned8		0		S	The preset discrete value to use in failure for hardware output 8.
25	**OUT_D1**	**DS-66**				**D / Man**	**The calculated discrete output variable 1 of the block in AUTO mode or specified by the user when in MAN mode.**

Idx	Parameter	Data Type/ (Length)	Valid Range/ Options	Default Value	Units	Store/ Mode	Description
26	**OUT_D2**	**DS-66**				**D / Man**	**The calculated discrete output variable 2 of the block in AUTO mode or specified by the user when in MAN mode.**
27	**OUT_D3**	**DS-66**				**D / Man**	**The calculated discrete output variable 3 of the block in AUTO mode or specified by the user when in MAN mode.**
28	OUT_D4	DS-66				D / Man	The calculated discrete output variable 4 of the block in AUTO mode or specified by the user when in MAN mode.
29	OUT_D5	DS-66				D / Man	The calculated discrete output variable 5 of the block in AUTO mode or specified by the user when in MAN mode.
30	OUT_D6	DS-66				D / Man	The calculated discrete output variable 6 of the block in AUTO mode or specified by the user when in MAN mode.
31	OUT_D7	DS-66				D / Man	The calculated discrete output variable 7 of the block in AUTO mode or specified by the user when in MAN mode.
32	OUT_D8	DS-66				D / Man	The calculated discrete output variable 8 of the block in AUTO mode or specified by the user when in MAN mode.
33	**HW_IN**	**DS-160**				**D / Man**	**Data Structure: 16 unsigned8 values and 1 unsigned8 status for Hardware Discrete Inputs.**
34	**HW_OUT**	**DS-159**				**D / Man**	**Data Structure: 8 unsigned8 values and 1 unsigned8 status for Hardware Discrete Outputs.**
35	AUX_01_16	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 01_16.
36	AUX_17_32	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 17_32.
37	AUX_33_48	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 33_48.
38	AUX_49_64	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 49_64.
39	AUX_65_80	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 65_80.
40	AUX_81_96	Bitstring(2)				D/ OS	Auxiliary bit enumerated variable 81_96.
41	TON_PST	16 Floats	Positive	0	sec	S/ OS	Array of 16 float elements where the user can set the PST timer duration in seconds for each Timer ON Delay.
42	TON_CTA	16 Floats		0	sec	D	Array of 16 float elements where the user can read the lapsed time until the PST timer duration in seconds for each Timer ON Delay.
43	TON_OUT	Bitstring(2)				D	A bit enumerated that indicates the timer output states.
44	TOFF_PST	16 Floats	Positive	0	sec	S/ OS	Array of 16 float elements where the user can set the PST timer duration in seconds for each Timer OFF Delay.
45	TOFF_CTA	16 Floats		0	sec	D	Array of 16 float elements where the user can read the lapsed time until the PST timer duration in seconds for each Timer OFF Delay.
46	TOFF_OUT	Bitstring(2)				D	A bit enumerated that indicates the timer output states.
47	TP_PST	16 Floats	Positive	0	sec	S/ OS	Array of 16 float elements where the user can set the PST timer duration in seconds for each Timer Pulse.
48	TP_CTA	16 Floats		0	sec	D	Array of 16 float elements where the user can read the lapsed time until the PST timer duration in seconds for each Timer Pulse.
49	TP_OUT	Bitstring(2)				D	A bit enumerated that indicates the timer output states.

Idx	Parameter	Data Type/ (Length)	Valid Range/ Options	Default Value	Units	Store/ Mode	Description
50	CTU_PST	16 Unsigned32	Positive	0	None	S/ OS	Array of 16 unsigned integer32 elements where the user can set the PST value of each pulse counter. The counter will increment from zero to PST value.
51	CTU_CTA	16 Unsigned32		0	None	D	Array of 16 unsigned integer32 elements where the user can read the incremented value of each pulse counter.
52	CTU_OUT	Bitstring(2)				D	A bit enumerated that indicates the counter output states.
53	CTD_PST	16 Unsigned32	Positive	0	None	S/ OS	Array of 16 unsigned integer32 elements where the user can set the PST value of each pulse counter. PST is a preset value since the counter will decrement until zero.
54	CTD_CTA	16 Unsigned32		0	None	D	Array of 16 unsigned integer32 elements where the user can read the decremented value of each pulse counter.
55	CTD_OUT	Bitstring(2)				D	A bit enumerated that indicates the counter output states.
56	RS_OUT	Bitstring(2)				D	A bit enumerated that indicates the RS Flip-Flop output states.
57	SR_OUT	Bitstring(2)				D	A bit enumerated that indicates the SR Flip-Flop output states.
58	**LOGIC_01**	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 1.
59	**LOGIC_02**	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 2.
60	**LOGIC_03**	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 3.
61	LOGIC_04	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 4.
62	LOGIC_05	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 5.
63	LOGIC_06	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 6.
64	LOGIC_07	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 7.
65	LOGIC_08	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 8.
66	LOGIC_09	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 9.
67	LOGIC_10	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 10.
68	LOGIC_11	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 11.
69	LOGIC_12	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 12.
70	LOGIC_13	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 13.
71	LOGIC_14	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 14.
72	LOGIC_15	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 15.
73	LOGIC_16	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 16.
74	LOGIC_17	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 17.
75	LOGIC_18	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 18.
76	LOGIC_19	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 19.
77	LOGIC_20	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 20.
78	LOGIC_21	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 21.
79	LOGIC_22	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 22.
80	LOGIC_23	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 23.
81	LOGIC_24	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 24.
82	LOGIC_25	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 25.
83	LOGIC_26	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 26.
84	LOGIC_27	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 27.
85	LOGIC_28	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 28.
86	LOGIC_29	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 29.

Idx	Parameter	Data Type/ (Length)	Valid Range/ Options	Default Value	Units	Store/ Mode	Description
87	LOGIC_30	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 30.
88	LOGIC_31	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 31.
89	LOGIC_32	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 32.
90	LOGIC_33	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 33.
91	LOGIC_34	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 34.
92	LOGIC_35	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 35.
93	LOGIC_36	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 36.
94	LOGIC_37	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 37.
95	LOGIC_38	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 38.
96	LOGIC_39	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 39.
97	LOGIC_40	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 40.
98	LOGIC_41	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 41.
99	LOGIC_42	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 42.
100	LOGIC_43	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 43.
101	LOGIC_44	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 44.
102	LOGIC_45	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 45.
103	LOGIC_46	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 46.
104	LOGIC_47	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 47.
105	LOGIC_48	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 48.
106	LOGIC_49	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 49.
107	LOGIC_50	VisibleString(24)		Spaces	Na	S/ OS	Logical Line Command 50.
108	LOGIC_CHECk	Unsigned8	0 - Enable., 1 – Checked. 2- Changed but not checked yet.	1 - Checked.	Na	D/OS	Allows the check for logic line.
109	ERROR_LINE	Unsigned8	0-50	1	Na	S	Indicates the logic line where there is an error.

Idx	Parameter	Data Type/ (Length)	Valid Range/ Options	Default Value	Units	Store/ Mode	Description
110	ERROR_CODE	Unsigned8	0 - Logic Ok. 1 - Exceed String Length or string not valid. 2 - Non valid operand. 3 - No implemented logic or missing ';' 4 - Missing parentheses or argument not valid. 5 - Non valid resource. 6 - Argument not valid. 7 - Function not valid 8 - Non available resource. 9 - Non valid attribution. 10 - First Argument not valid. 11- Second Argument not valid.	3 - No implemented logic or missing ';'	Na	S	Indicated the code for the error in the logic line.
111	CHANGE_OPTION	Unsigned8	0 - Logic parameter changes are only allowed in Out of Service. 1 - Always accept Logic parameter changes.	0 - Logic parameter changes are only allowed in Out of Service.	Na	S	Enable logic parameter changes independent of Mode Block parameter
112	UPDATE_EVT	DS-73			Na	D	This alert is generated by any change to the static data.
113	BLOCK_ALM	DS-72			Na	D	The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.

Legend:
E – Enumerated parameter;
Na – Admensional parameter;
RO – Read only;
D – dynamic;
N – non-volatile;
S – Static
Gray Background Line: Default Parameters of Syscon

The following table describes the Logic Operation and Command line and the correspondent Symbols used in the logic line:

Logic Operation and Command line	Symbol - description
AND	&
OR	|
XOR	^
NOT	!
EQUAL	=
(arg1,arg2)	To define function arguments
;	End of logic line

The logic does NOT (!) work only with simple variables. Example:
OUT1=!IN1;
Note that it is not allowed to have, for example,
OUT1=!TP01(IN1);
To use it this way, we should have:
A01= TP01(IN1);. -> OUT1=!A01;

The logic is always executed line by line and from left to right in the logic line. Spaces are not allowed between the characters. **Empty lines are not allowed between logic lines and the implemantation of logic lines must be in sequence.**

After writing the logic into the LOGIC_XX (XX:01 -> XX:50) parameters, the user needs to select the option "Enable" in the parameter LOGIC_CHECK in order to verify the errors. **When the logic is configured using the downloading process, it is necessary to configure first the LOGIC_XX (XX:01 -> XX:50) parameters and then the LOGIC_CKECK parameter. This sequence is fundamental to performing the check.**

The following table shows the mnemonic for each block parameter used in the logic lines. The mnemonic must be in capital letters:

Parameter	Mnemonic
HW_IN.Value1	I01
HW_IN.Value2	I02
HW_IN.Value3	I03
HW_IN.Value4	I04
HW_IN.Value5	I05
HW_IN.Value6	I06
HW_IN.Value7	I07
HW_IN.Value8	I08
HW_IN.Value9	I09
HW_IN.Value10	I10
HW_IN.Value11	I11
HW_IN.Value12	I12
HW_IN.Value13	I13
HW_IN.Value14	I14
HW_IN.Value15	I15
HW_IN.Value16	I16
HW_IN.Status	SI
HW_OUT.Status	SO
HW_OUT.Value1	O1
HW_OUT.Value2	O2
HW_OUT.Value3	O3
HW_OUT.Value4	O4
HW_OUT.Value5	O5
HW_OUT.Value6	O6
HW_OUT.Value7	O7
HW_OUT.Value8	O8
IN_D1.Status	IN1S
IN_D2.Status	IN2S
IN_D3.Status	IN3S
IN_D4.Status	IN4S
IN_D5.Status	IN5S
IN_D6.Status	IN6S
IN_D7.Status	IN7S
IN_D8.Status	IN8S
IN_D1.Value	IN1
IN_D2.Value	IN2
IN_D3.Value	IN3
IN_D4.Value	IN4
IN_D5.Value	IN5
IN_D6.Value	IN6
IN_D7.Value	IN7
IN_D8.Value	IN8
OUT_D1.Status	SOUT1
OUT_D2.Status	SOUT2
OUT_D3.Status	SOUT3
OUT_D4.Status	SOUT4
OUT_D5.Status	SOUT5

Parameter	Mnemonic
OUT_D6.Status	SOUT6
OUT_D7.Status	SOUT7
OUT_D8.Status	SOUT8
OUT_D1.Value	OUT1
OUT_D2.Value	OUT2
OUT_D3.Value	OUT3
OUT_D4.Value	OUT4
OUT_D5.Value	OUT5
OUT_D6.Value	OUT6
OUT_D7.Value	OUT7
OUT_D8.Value	OUT8
FSTATE_VAL_D1	FS1
FSTATE_VAL_D2	FS2
FSTATE_VAL_D3	FS3
FSTATE_VAL_D4	FS4
FSTATE_VAL_D5	FS5
FSTATE_VAL_D6	FS6
FSTATE_VAL_D7	FS7
FSTATE_VAL_D8	FS8
AUX_01_16	A01-A16
AUX_17_32	A17-A32
AUX_33_48	A33-A48
AUX_49_64	A49-A64
AUX_65_80	A65-A80
AUX_81_96	A81-A96
TON	TON01-TON16
TOFF	TOF01-TOF16
TP	TP01-TP16
CTU	CTU01-CTU16
CTD	CTD01-CTD16
RS	RS01-RS16
SR	SR01-SR16

Functions

For each type of function there are 16 available resources and the user can use only once each resource. To use the function results, the user can make attribution for auxiliary bits.

TP TIMER PULSE

This function generates a fixed time pulse in the output timer for every rising (false to true transition) on the input timer. The pulse width is determined by TP_PST parameter in seconds. Transitions in the input timer will be ignored while the pulse is active. The current time is available in the TP_CTA parameter.
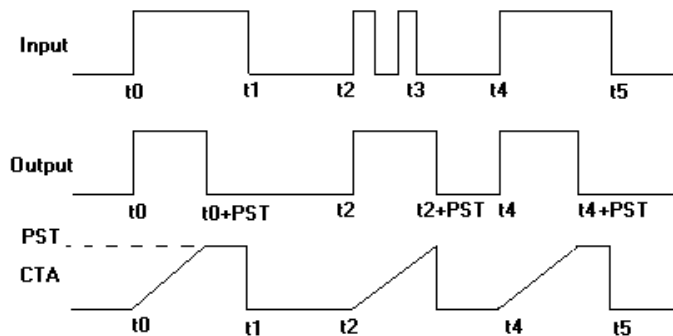
Timer Pulse Function – timing diagrams

The syntax for Timer Pulse is: TPxx(arg)

Where, xx is the used resource from 01 to 16 and arg is the function argument and it must be a simple variable. Examples:

O1=TP01(IN1);
OUT1= TP01(A05);
OUT3=TP08(FS1);

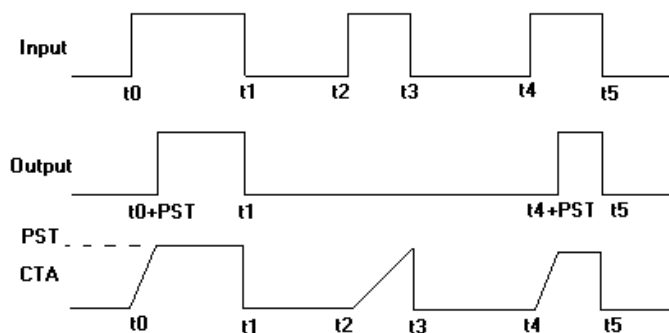For example, the following examples are not allowed in the logic line:
O1=TP01(IN1&IN2);: note that the argument is a result of an operation, it is not allowed.
O1=TP10(!IN1);: note that the argument is a result of NOT function, it is not allowed.
O1=TP10(CTD01(IN1,IN2));: note that the argument is a result of a function, it is not allowed.

TON TIMER ON-DELAY

This function delays the timer output of going to true for a period of time after the input has moved to true. This period is configured by TON_PST parameter in seconds. If the input goes to false before the PST time, the output timer will remain in false. The CTA parameter will show the remainder time until PST value.

Timer On-Delay Function – timing diagrams

The syntax for Timer On-Delay is: TONxx(arg)

Where, xx is the used resource from 01 to 16 and arg is the function argument and it must be a simple variable . Examples:

O1=TON01(IN1)&SI;
OUT1= TON01(A05);
OUT3=TON08(FS1);

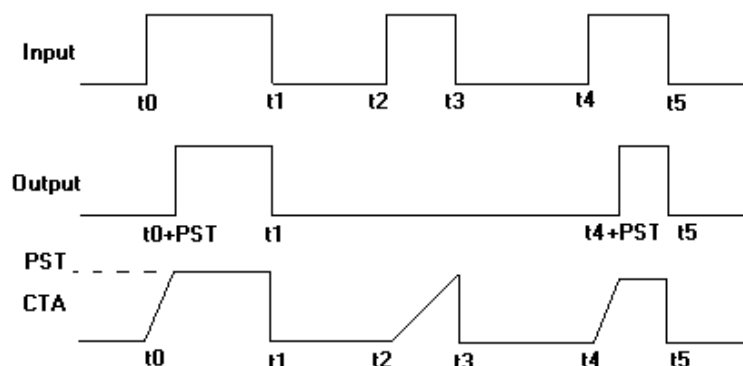For example, the following examples are not allowed in the logic line:

O1=TON01(IN1&IN2);: note that the argument is a result of an operation, it is not allowed.
O1=TON10(!IN1);: note that the argument is a result of NOT function, it is not allowed.
O1=TON10(CTD01(IN1,IN2));: note that the argument is a result of a function, it is not allowed.

TOF TIMER OFF-DELAY

This function extends the true state of timer input for a determined period of time for the output timer. This period is configured by TOF_PST parameter in seconds. If the input goes to true before the out goes to false, the out will stay on true and the time period will begin to count again at the moment when the input goes to false. The CTA parameter will show the remainder time until PST value.

Timer OFF-Delay Function – timing diagrams

The syntax for Timer Off-Delay is: TOFxx(arg)

Where, xx is the used resource from 01 to 16 and arg is the function argument and it must be a simple variable .Examples:

O1=TOF01(IN1)&SI;
OUT1= TOF01(A05);
OUT3=TOF08(FS1);

For example, the following examples are not allowed in the logic line:
O1=TOF01(IN1&IN2);: note that the argument is a result of an operation, it is not allowed.
O1=TOF10(!IN1);: note that the argument is a result of NOT function, it is not allowed.
O1=TOF10(CTD01(IN1,IN2));: note that the argument is a result of a function, it is not allowed.

CTD PULSE COUNTER DOWN

This function is used to count rising transitions (from false to true) in the counter input(arg1). Every time it is seeing a rising transition the internal counter accumulator (CTA) decrements of one. When the CTA reaches zero the counter output will go to true. The counter value will be preset for PST.A transition from false to true in the second argument(arg2) presets the counter.

The syntax for CTD is: CTDxx(arg1,arg2)
Where, xx is the used resource from 01 to 16 and arg1 and arg2 are the function arguments and they must be simple variables. Examples:

O3=CTD10(IN1,IN2);
OUT1=CTD03(A11,A14)&SI;

For example, the following examples are not allowed in the logic line:
O1=CTD01(IN1&IN2,IN3);: note that the argument is a result of an operation, it is not allowed.
O1=CTD10(!IN1,IN3);: note that the argument is a result of NOT function, it is not allowed.
O1=CTD10(TP01(IN1),IN2);: note that the argument is a result of a function, it is not allowed.

CTU PULSE COUNTER UP
This function is used to count rising transitions (from false to true) in the counter input(arg1). Every time it is seeing a rising transition the internal counter accumulator (CTA) increments of one. When the CTA reaches the preset value PST, the counter output will go to true. A transition from false to true in the second argument(arg2) resets the counter.

The syntax for CTU is: CTUxx(arg1,arg2)
Where, xx is the used resource from 01 to 16 and arg1 and arg2 are the function arguments and they must be simple variables. Examples:

O3=CTU10(IN1,IN2);
OUT1=CTU03(A11,A14)&SI;

For example, the following examples are not allowed in the logic line:
O1=CTU01(IN1&IN2,IN3);: note that the argument is a result of an operation, it is not allowed.
O1=CTU10(!IN1,IN3);: note that the argument is a result of NOT function, it is not allowed.
O1=CTU10(TP01(IN1),IN2);: note that the argument is a result of a function, it is not allowed.

RS FLIP-FLOP
This function has the following operation table:

R(arg1)	S(arg2)	OUT
0	0	Last state
0	1	1
1	0	0
1	1	0

The syntax for RS Flip-Flop is: RSxx(arg1,arg2)
Where, xx is the used resource from 01 to 16 and arg1 and arg2 are the function arguments and they must be simple variables. Examples:

O3=RS10(IN1,IN2);
OUT1=RS03(A11,A14)&SI;

For example, the following examples are not allowed in the logic line:
O1=RS01(IN1&IN2,IN3);: note that the argument is a result of an operation, it is not allowed.
O1=RS10(!IN1,IN3);: note that the argument is a result of NOT function, it is not allowed.
O1=RS10(TP01(IN1),IN2);: note that the argument is a result of a function, it is not allowed.

SR FLIP-FLOP
This function has the following operation table:

S(arg1)	R(arg2)	OUT
0	0	Last state
0	1	0
1	0	1
1	1	1

The syntax for SR Flip-Flop is: SRxx(arg1,arg2)

Where, xx is the used resource from 01 to 16 and arg1 and arg2 are the function arguments and they must be simple variables. Examples:

O3=SR10(IN1,IN2);
OUT1=SR03(A11,A14)&SI;

For example, the following examples are not allowed in the logic line:
O1=SR01(IN1&IN2,IN3);: note that the argument is a result of an operation, it is not allowed.
O1=SR10(!IN1,IN3);: note that the argument is a result of NOT function, it is not allowed.
O1=SR10(TP01(IN1),IN2);: note that the argument is a result of a function, it is not allowed.

Error Code

Some examples of error conditions:

Error Code: "Exceed String Length or string not valid."
a) OUT1=IN1&IN2&IN2|IN4^IN5|IN6;

Note that they are 29 characters on the string and the maximum allowed is 24.

b) OUT1=IN1&in2;

Note that the logic is case sensitive. All characters must be in capital letters.

Error Code: "Non valid operand."
OUT1=IN1%IN2;

Note that the % is not allowed. See the table that describes the Logic Operation and Command line.

Error Code: "No implemented logic or missing ';' ."
OUT1=IN1

Note that the "; " is missing at the end of the logic line.

Error Code: "Missing parentheses or argument not valid."
OUT1=TP10(IN1;

Note that the parentheses is missing in the timer pulse function.

Error Code: "Non valid resource."
OUT1=TP18(IN1);

Note that there are 16 resources for each function

Error Code: "Argument not valid."
OUT1=TP10(IN10);

Note that there are only 8 inputs. IN10 is not a valid argument.

Error Code: "Function not valid."
OUT1=TR10(IN1);

Note that TR is not a valid function.

Error Code: "Non available resource."
OUT1=TP10(IN1);

A03=TP10(IN7);

Note that there are 16 resources for each function. The resource 10 for the timer has already been used and can not be used again. However, the result of the function can be assigned to an auxiliary variable and this auxiliary variable can be used several times.
A03=TP10(IN7);
A03=TP10(IN7);

Error Code: "Non valid attribution."
IN1=IN2^TP03(IN4);

Note that is not allowed attribution to inputs.

Error Code: "First Argument not valid."
OUT1=CTD01(!IN1,IN2);

Note that the arguments are necessarily simple variables and not functions.

OUT1=RS11(IN15,IN2);
Note that the first argument is not allowed.

**Error Code: "Second Argument not valid."**
a) OUT1=CTD01(IN1,!IN2);
Note that the arguments are necessarily simple variables and not functions.

b) OUT1=RS11(IN1,IN20);

Note that the second argument is not allowed.

Example of applications

1) According to the next figure, we have an industrial application where the aim is to fill up the bottles with a chemical fluid. The conveyor moves the bottles up to the filling direction and then the bottle is detected by a sensor .The conveyor must stop and open the valve of filling and the level is detected by another sensor. After detecting the level, the system must wait for 10 seconds and then move the conveyor again until the next bottle.
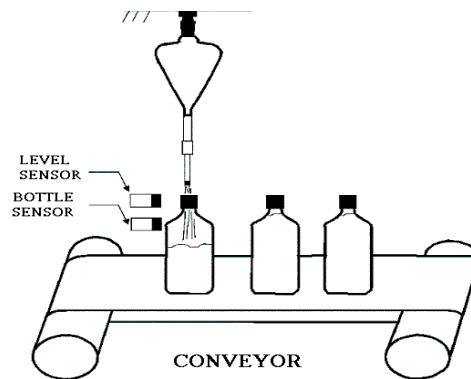
Using the Flexible Function Block we have the following definitions:
- ☑ The conveyor will be turned on using the hardware output 01 (O1);
- ☑ The fluid valve will be turned on using the hardware output 02 (O2);
- ☑ The bottle sensor will be connected to the hardware input 01 (I01);
- ☑ The level sensor will be connected to the hardware input 02 (I02);
- ☑ The power system will be connected to the hardware input 03 (I03);
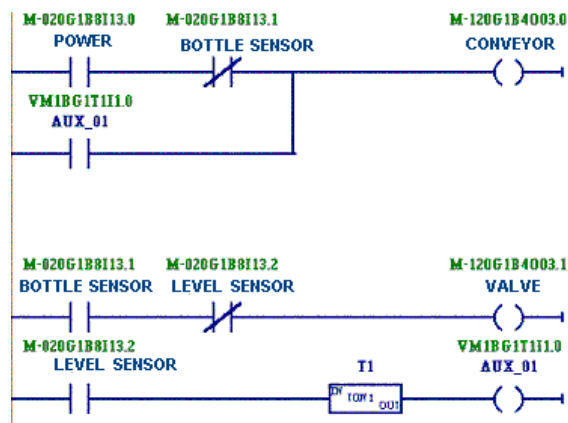
We have the following configuration:
```
TON_PST resource [01] = 10.0s.
LOGIC_01    A01=TON01(I02);
LOGIC_02    O1=I03&!I01|A01;
LOGIC_03    O2=I01&!I02;
```
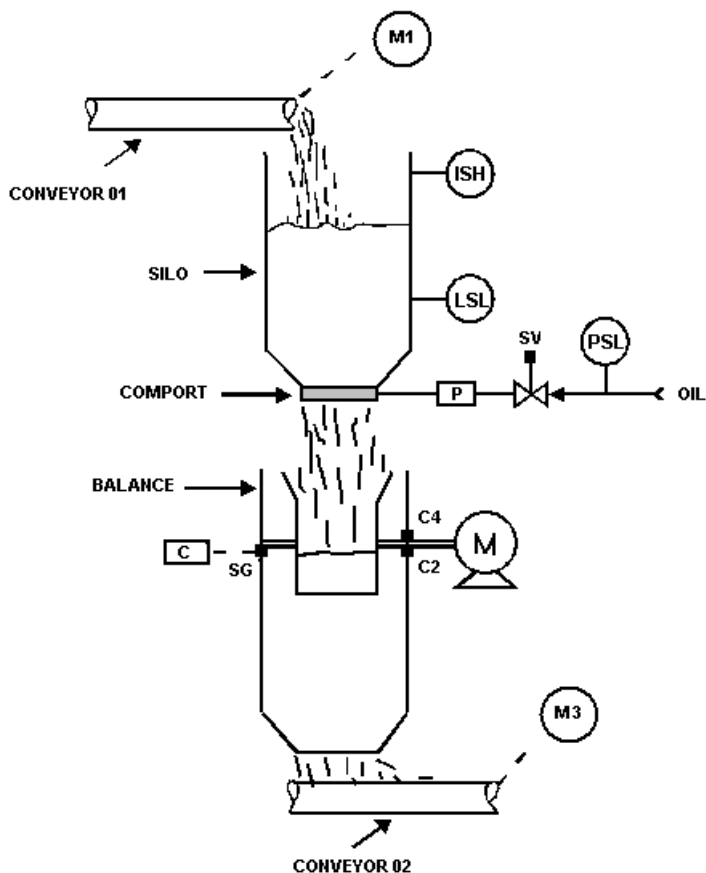
Making an analogy to ladder programming, we have:

2) In the following application we have the control of steps to operate an electro-mechanical balance, that weights phosphatic stone .

The weight process is done by boat-load, the system executes one full weight cycle each interval time of 20 seconds. See the following figure:

M1 e M3 - Motors for the conveyors
C2 e C4 - Limit Switches
LSH - High Level Sensor
LSL - Low Level Sensor
SG - Load Cell
SV - Solenoid Valve
M - Bucket Motor
P - Comport Piston
C - Weight Circuit

Process:

The system requires the following conditions to startup:
- ☑ The phosphatic stone level (LSL non activated);
- ☑ Oil Pressure (PSL on);
- ☑ Conveyor 02 active (M3 on);
- ☑ Bucket in initial position (C4 on);

After the initial conditions, we note:
- ☑ By activating the power switch, the comport opens, and the bucket starts to get loaded;
- ☑ After reaching the desired weight, the comport closes. After 5 seconds, the bucket rotates 180º and unload the product into the conveyor 02.

Note:

- This new position will be detected by C2 and after 5 seconds, the bucket will have to return to initial position, which will be detected by C4.
- After the bucket returns to the initial position, a new weight cycle begins.

Comment:

- The operation sequence must be stopped if any requirement is not satisfied.
- The silo comort is activated by a hydraulic piston.

Using the Flexible Function Block we have the following definitions:

- ☑ LSL will be connected to the hardware input 01 (I01);
- ☑ LSH will be connected to the hardware input 02 (I02);
- ☑ PSL will be connected to the hardware input 03 (I03);
- ☑ C2 will be connected to the hardware input 04 (I04);
- ☑ C4 will be connected to the hardware input 05 (I05);
- ☑ Power will be connected to the hardware input 06 (I06);
- ☑ M3 will be connected to the hardware input 07 (I07);
- ☑ M will be activated by hardware output 01 (O1);
- ☑ The Comport will be activated by hardware output 02 (O2);
- ☑ M1 will be activated by hardware output 03 (O3);

We have the following configuration:

TON_PST resource [01] = 5.0s.
LOGIC_01   A01=!I01&I03&I07&I05;
LOGIC_02   A02=I06&RS01(I02,I01);
LOGIC_03   O3=A02&I03;
LOGIC_04   A03=I03&I07;
LOGIC_05   O2=I06&A03&!I04;
LOGIC_06   O1=TON01(I04)&!I05&A03;

3) Using Fault-State values:

Lets suppose we have the following condition:

- A01 receives the logic between the status for discrete inputs as follows:
  A01=IN1S&IN2S;
  when the status is bad for one of these inputs, then, A01=false(0),
  otherwise, A01=true (1);
- FS1 is the fault-state value for O1;
- A02 is the bit containing the logic for O1;

We have the following table between the FS1, A01 and A02:

FS1	A01	A02	O1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Then,

A03=!FS1&A01&A02;
A04=FS1&!A01&!A02;
A05=FS1&!A01&A02;
A06=FS1&A01&A02;
O1=A03|A04|A05|A06;

MAINTENANCE PROCEDURES

General

SMAR **DC302** Fieldbus Remote I/O are extensively tested and inspected before delivery to the end user. Nevertheless, during their design and development, consideration was given to the possibility of repairs by the end user, if necessary.

In general, it is recommended that the end user do not try to repair printed circuit boards. Instead, he should have spare circuit boards, which may be ordered from SMAR whenever necessary.

Troubleshooting	
Symptom	**Probable Sources of Trouble**
No Quiescent Current	Fieldbus Remote I/O Connections: **Check wiring polarity and continuity.** Power Supply: **Check power supply output. The voltage at the DC302 Fieldbus terminals must be between 9 and 32 VDC.** Electronic Circuit Failure: **Check the boards for defect by replacing them with spare ones.**
No Communication	**Network Connections:** Check the network connections: devices, power supply, and terminators. **Network Impedance:** Check the network impedance (power supply impedance and terminators). **Controller Configuration:** Check configuration of communication parameters of controller. **Network Configuration:** Check communication configuration of the network. **Electronic Circuit Failure:** **Try to replace the controller circuit with spare parts.**
Incorrect Inputs	**Input Terminals Connection:** Check wiring polarity and continuity. Power supply for Inputs: **Check power supply. The voltage must be between 18 and 30 VDC and the typical consumption when all input is ON is 120mA.**
Incorrect Outputs	**Output Terminals Connection:** Check wiring polarity and continuity. **Power supply for Outputs:** **Check power supply. The voltage must be between 20 and 30 VDC and the maximum current per output is 0.5 A.**

Disassembly Procedure

Refer to the Figure 4.1 DC302 Exploded view. Make sure to disconnect power supply before disassembling the DC302.

WARNING
The boards have CMOS components, which may be damaged by electrostatic discharges. Observe correct procedures for handling CMOS components. It is also recommended to store the circuit boards in electrostatic-proof cases.

Loose the lateral locks that attach the housing cover and then the main lock. You will have the access to the main circuit board and the I/O electronic board. Gently pull out the main board. To remove the electronic boards, first unscrew the screws that anchors them to the housing, and gently pull out the boards.

Reassembly Procedure

- Put the boards into housing.
- Anchors the board with their screws.
- Make sure all inter connecting pins are connected.
- Observi the LEDs mounting positions, gently lock the housing cover in the lateral locks and the main lock.

Firmware Update Procedure

For firmware update of the DC302 equipment see FDI302Plus manual, visit website Smar: www.smar.com

NOTA
The FDI302Plus should be connected to the CN4 connector
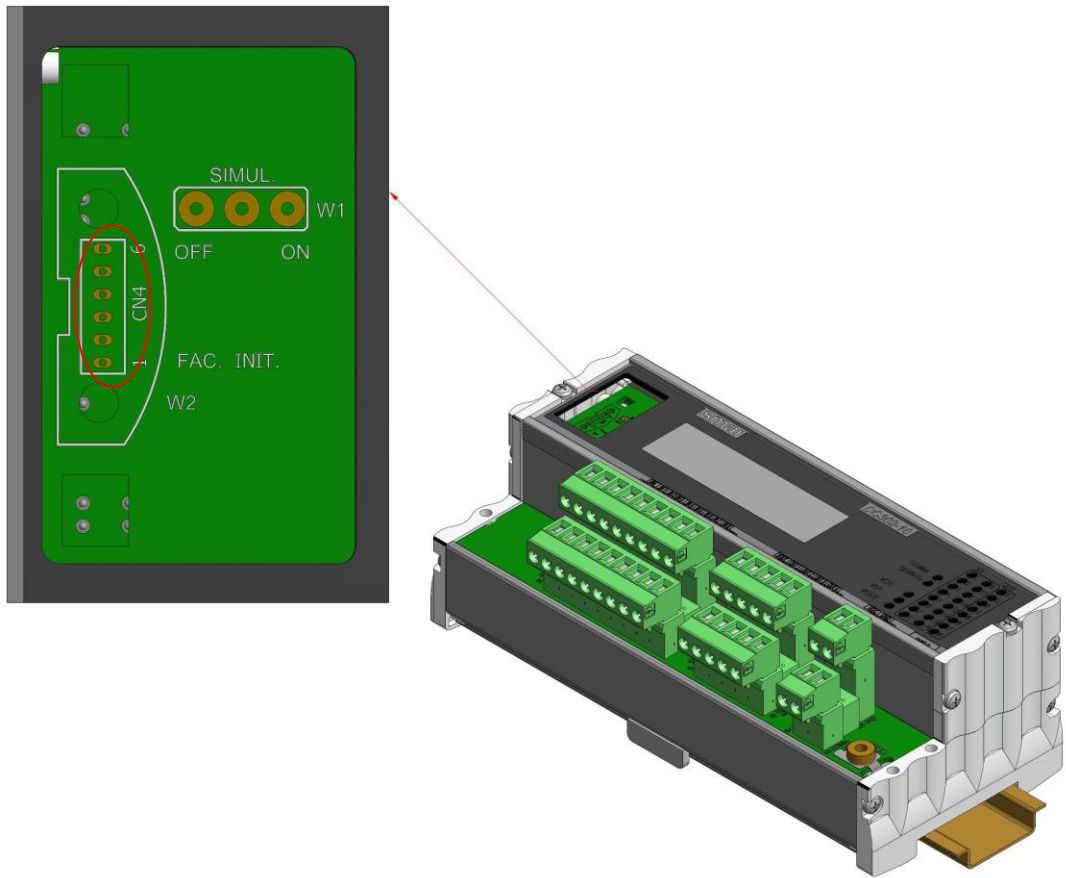
Boards Interchangeability

Main and I/O boards can be changed independently.

Accessories

ACCESSORIES	
ORDERING CODE	**DESCRIPTION**
BC302	Fieldbus/RS232 Interface
SYSCON	System Configuration Tool
PS302	Power Supply
PSI302	Power Supply Impedance
BT302	Terminator
PCI	Process Control Interface
FDI-302-2	Field Device Interface - Foundation Fieldbus & PROFIBUS PA
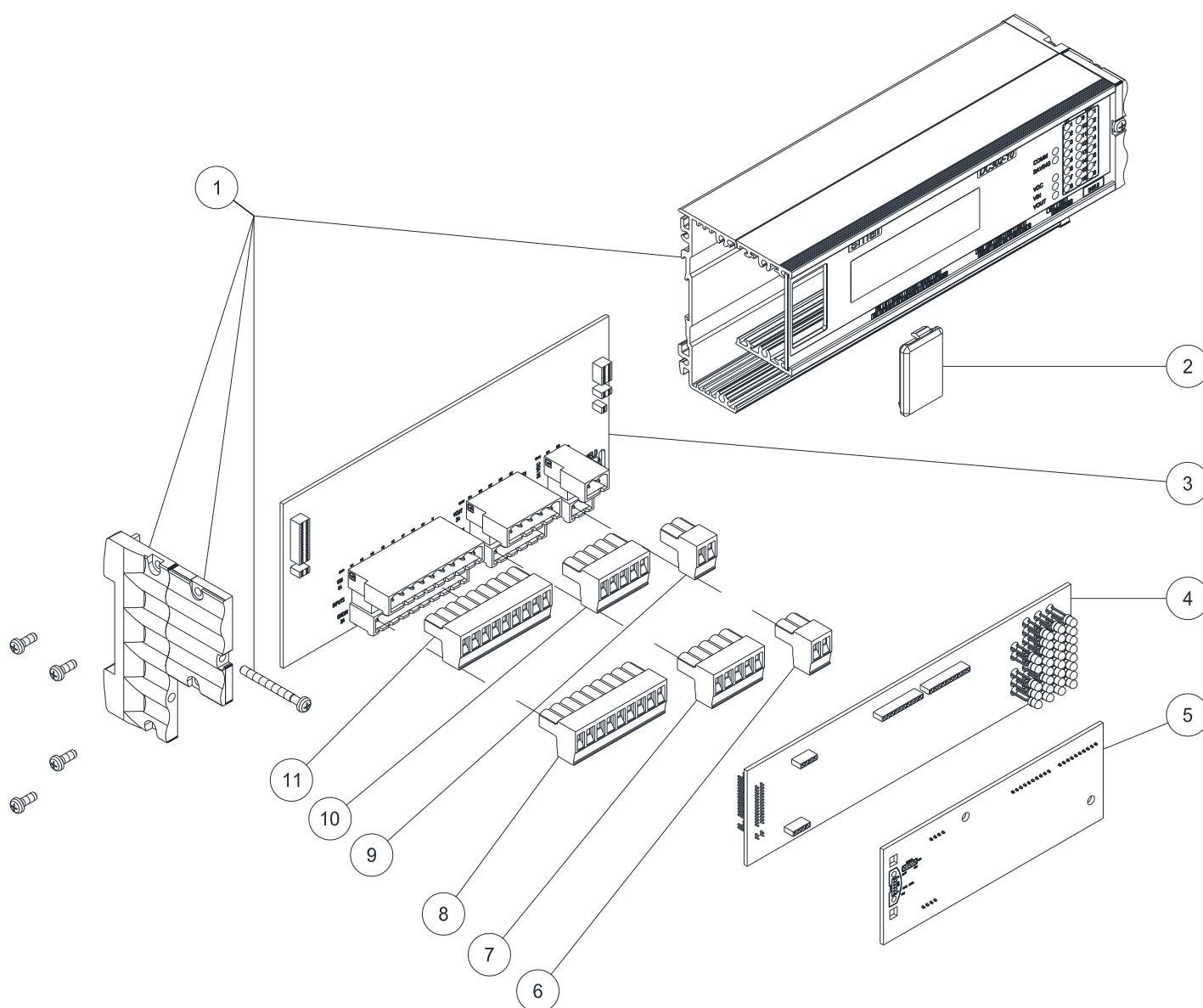
Figure 4.1 – DC Exploded view

Spare Parts

SPARE PARTS		
NAME	**POSITION**	**CODE**
Housing	1	400 - 1371
Housing Cover	2	400 - 1372
I/O Board (GLL 1469)	3	400 - 1373
Font Board (GLL 1468)	4	400 - 1374
Main board (GLL 1467)	5	400 - 1375
2-way terminal numbered 6A to 6B	6	400 - 1376
2-way terminal numbered 5A to 5B	9	400 - 1377
5-way terminal numbered 4A to 4E	7	400 - 1378
5-way terminal numbered 3A to 3E	10	400 - 1379
9-way terminal numbered 2A to 2I	8	400 - 1380
9-way terminal numbered 1A to 1I	11	400 - 1381

TECHNICAL SPECIFICATIONS

General

Signal (Communication)	Digital only. Fieldbus, 31.25 Kbits/s voltage mode
Power Supplies	If there are requirements for power supply isolation between inputs and outputs, it is recommended to use at least two power supplies, one for inputs and another one for outputs and VDC. If the application does not require isolation between inputs and outputs, only one power supply could be used for inputs, outputs and VDC. Inputs and outputs are optically isolated from each other.
Current consumption quiescent	150 mA from VDC power supply
Turn-on Current	400 mA during the first 20s after power
Turn-on Time	Approximately 10 seconds.
Update Time	Approximately 100 ms. The update time is related to the update of the inputs and outputs. It is independent of the system macrocycle.
Output impedance	Non-intrinsic safety from 7.8 kHz - 39 kHz should be greater or equal to 3 k. Intrinsic safety output impedance (assuming an IS barrier in the power supply) from 7.8 kHz - 39 kHz should be greater or equal to 400.
Vibration Effect	Meets SAMA PMC 31.1.
Temperature Limits	**Operation:** -40 to 85ºC (-40 to 185 ºF) **Storage:** -40 to 110ºC (-40 to 230 ºF)
Housing	**Protection:** it has IP20 rating (finger protected) and meets VBG4 and other European accident prevention requirements.
Mounting	Using DIN rail (TS35-DIN EN 50022 or TS32-DIN EN50035 or TS15-DIN EN50045.

DC302 Inputs

Description-Inputs

The input module senses the DC input voltage and converts it into a True (ON) or False (OFF) logic signal.   It has 1 optically isolated group of 16 inputs to detect 24Vdc.

In case of failuring of input power supply there will be an indication in the BLOCK_ERR parameter for input function blocks such as DI, MDI, FFB (See Function Blocks Manual).

Technical specifications

Architecture	Number of Inputs is 16
Isolation, groups are individually isolated	Optical Isolation up to 5000 Vac
External Power	Voltage Source for Inputs 18 - 30 Vdc
Typical Consumption per group (all inputs ON)	120 mA
Power Indicator	Green LED
Inputs	ON State Level (True Logic) 15 - 30 Vdc
	OFF State Level (False Logic) 0 - 5 Vdc
Typical Impedance	3k9 Ω
Status display	Red LED
Switching Information	Time from "0" to "1": 30 μs
	Time from "1" to "0": 50 μs
Wire	One wire 14 AWG (2 mm2)
	Two wires 20 AWG (0.5 mm2)

DC302 Open Collector Outputs

Description – Outputs

The outputs are designed with open collector NPN transistors that are able to drive relays, solenoids and other DC loads with up to 0.5 A per output. All channels within a group share the same ground whereas groups are isolated from each other and the Fieldbus network.

In case of failing of output power supply there will be an indication in the BLOCK_ERR parameter for output function blocks such as DO, MDO, FFB, STEP_PID (See Function Blocks Manual).

NOTE
The discrete outputs can be used by several blocks, such as DO, MDO, STEP_PID, FFB. However, an output or group that is being used with one block, can not be used by another block.

Technical specifications

Architecture	Number of Outputs 8
Isolation	Optical Isolation up to 5000 Vac
External Power	Voltage Source for Outputs 20 to 30 Vdc
Maximum Consumption	35 mA
Power Indicator	Green LED
Outputs	Maximum Switched Voltage 30 Vdc
	Maximum Saturation Voltage 0.55 V @ 0.5 A
	Maximum Current per Output 0.5 A
	Status Display Red LED
	Indicator Logic ON when the transistor is on
	Maximum Leakage Current 100 µA @ 35 Vdc
Output Status During: **Power-Up** **Firmware Download** **Configuration Download**	OFF
Independent Protection per Output	Thermal Shutdown 165 ºC
	Thermal Hysteresis 15 ºC
	Over-Current Protection 1.3 A @ 25 Vdc maximum
Clamp Diode, switching information	Time from 0 to 1: 250 µs
	Time from 1 to 0: 3 µs
Wire	One wire 14 AWG (2 mm^2)
	Two wires 20 AWG (0.5 mm^2)

Ordering Code

MODEL	DESCRIPTION
DC302-10 Fieldbus Remote I/O	- 1 Group of 16 24VDC optically isolated inputs.
	- 1 group of 8 optically isolated open collector outputs.

Appendix A

smar	**SRF** – Service Request Form
	Fieldbus Remote Input and Output

GENERAL DATA

Model: DC302

Serial Number: _____

TAG: _____

Channels being used in DC302:

INPUT   1-4 ( )   9-12 ( )
             5-8 ( )   13-16 ( )

OUTPUT   1-4 ( )   5-8   ( )

Configuration: PC ( )   Software: _____   Version: _____

INSTALLATION DATA

Type/Model/Manufacturer of device connected to DC302:   _____
_____

PROCESS DATA

Hazardous Area Classification: ( ) Yes, please specify: _____

                     ( ) No

                     More details:__ _____

Interference types present in the area: No interference ( )   Temperature ( )   Vibration ( )   Other: _____

Environment Temperature: From _____ºC up to _____ºC.

OCCURRENCE DESCRIPTION

_____
_____
_____
_____
_____

SERVICE SUGGESTION

Adjustment ( )   Cleaning ( )   Preventive Maintenance ( )   Update / Up-grade ( )

Other: _____

USER INFORMATION

Company: _____

Contact: _____

Title: _____

Section: _____

Phone: _____ _____   _____ _____   **Extension:** _____

E-mail: _____   **Date:** _____/ _____/ _____

For warranty or non-warranty repair, please contact your representative.
Further information about address and contacts can be found on www.smar.com/contactus.asp.

Returning Materials

If necessary to return the DC302 to SMAR, simply contact our office, informing the defective instrument serial number, and return it to our factory.

In order to speed up analysis and solution of the problem, the defective item should be returned with a description of the failure observed, with as much details as possible. Other information concerning the instrument operation, such as service and process conditions, is also helpful.

Instruments returned or to be revised outside the guarantee term should be accompanied by a purchase order or a quote request.

SMAR WARRANTY CERTIFICATE

1. SMAR guarantees its products for a period of 24 (twenty four) months, starting on the day of
2. issuance of the invoice. The guarantee is valid regardless of the day that the product was installed.

3. SMAR products are guaranteed against any defect originating from manufacturing, mounting, whether of a material or manpower nature, provided that the technical analysis reveals the existence of a quality failure liable to be classified under the meaning of the word, duly verified by the technical team within the warranty terms.

4. Exceptions are proven cases of inappropriate use, wrong handling or lack of basic maintenance compliant to the equipment manual provisions. SMAR does not guarantee any defect or damage caused by an uncontrolled situation, including but not limited to negligence, user imprudence or negligence, natural forces, wars or civil unrest, accidents, inadequate transportation or packaging due to the user's responsibility, defects caused by fire, theft or stray shipment, improper electric voltage or power source connection, electric surges, violations, modifications not described on the instructions manual, and/or if the serial number was altered or removed, substitution of parts, adjustments or repairs carried out by non-authorized personnel; inappropriate product use and/or application that cause corrosion, risks or deformation on the product, damages on parts or components, inadequate cleaning with incompatible chemical products, solvent and abrasive products incompatible with construction materials, chemical or electrolytic influences, parts and components susceptible to decay from regular use, use of equipment beyond operational limits (temperature, humidity, etc.) according to the instructions manual. In addition, this Warranty Certificate excludes expenses with transportation, freight, insurance, all of which are the customer's responsibility.

5. For warranty or non-warranty repair, please contact your representative.

    Further information about address and contacts can be found on
    www.smar.com/contactus.asp

6. In cases needing technical assistance at the customer's facilities during the warranty period, the hours effectively worked will not be billed, although SMAR shall be reimbursed from the service technician's transportation, meals and lodging expenses, as well dismounting/mounting costs, if any.

7. The repair and/or substitution of defective parts do not extend, under any circumstance, the original warranty term, unless this extension is granted and communicated in writing by SMAR.

8. No Collaborator, Representative or any third party has the right, on SMAR's behalf, to grant warranty or assume some responsibility for SMAR products. If any warranty would be granted or assumed without SMAR's written consent, it will be declared void beforehand.

9. Cases of Extended Warranty acquisition must be negotiated with and documented by SMAR.

10. If necessary to return the equipment or product for repair or analysis, contact us.
    See item 4.

11. In cases of repair or analysis, the customer must fill out the Revision Requisition Form (FSR) included in the instructions manual, which contains details on the failure observed on the field, the circumstances it occurred, in addition to information on the installation site and process conditions. Equipments and products excluded from the warranty clauses must be approved by the client prior to the service execution.

12. In cases of repairs, the client shall be responsible for the proper product packaging and SMAR will not cover any damage occurred in shipment.

13. In cases of repairs under warranty, recall or outside warranty, the client is responsible for the correct packaging and packing and SMAR shall not cover any damage caused during transportation. Service expenses or any costs related to installing and uninstalling the product are the client´s sole responsibility and SMAR does not assume any accountability before the buyer.

14. It is the customer's responsibility to clean and decontaminate products and accessories prior to shipping them for repair, and SMAR and its dealer reserve themselves the right to refuse the service in cases not compliant to those conditions. It is the customer's responsibility to tell SMAR and its dealer when the product was utilized in applications that contaminate the equipment with harmful products during its handling and repair. Any other damages, consequences, indemnity claims, expenses and other costs caused by the lack of decontamination will be attributed to the client. Kindly, fill out the Declaration of Decontamination prior to shipping products to SMAR or its dealers, which can be accessed at www.smar.com/doc/declarationofcontamination.pdf and include in the packaging.

15. This warranty certificate is valid only when accompanying the purchase invoice.