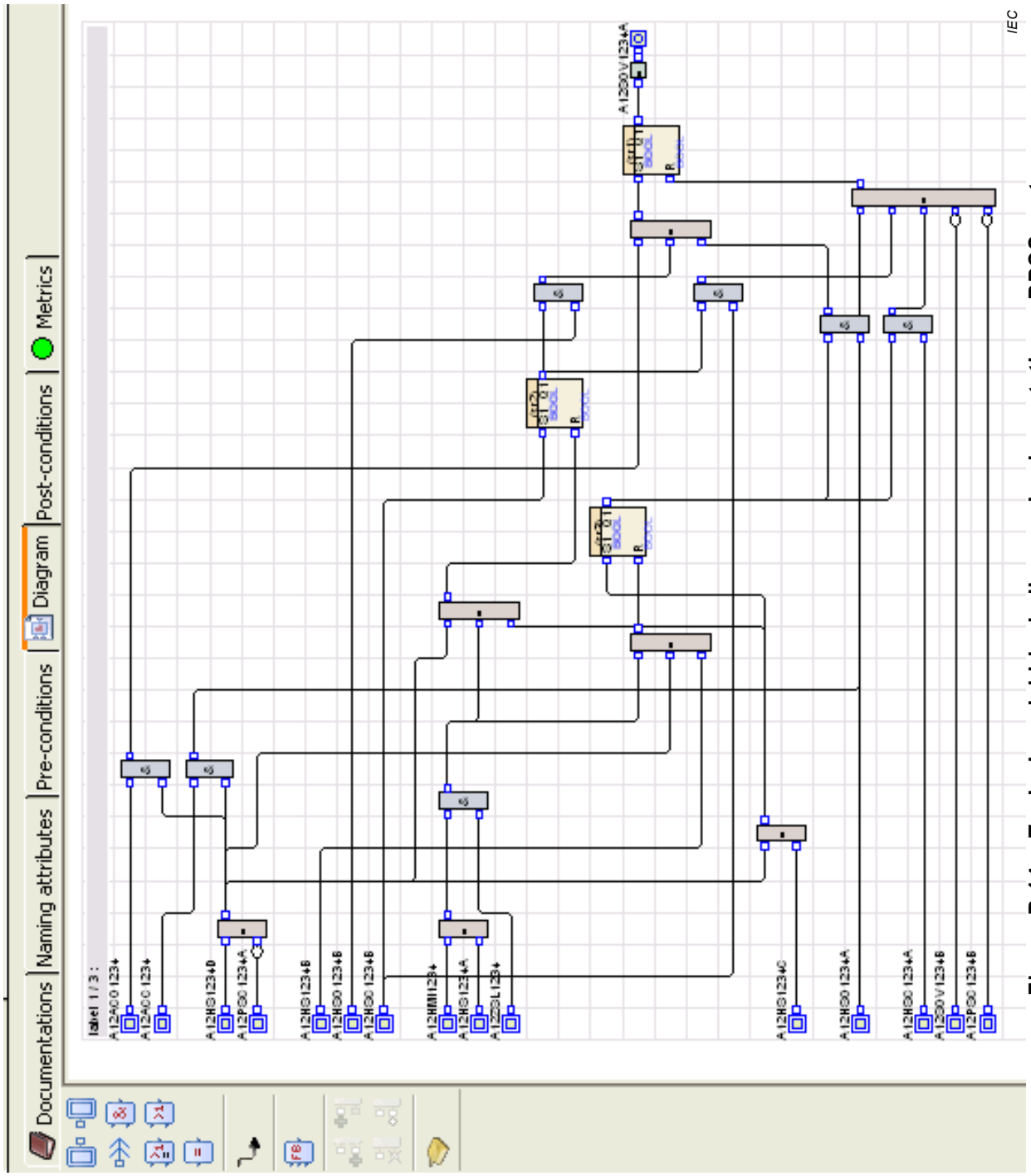

Figure B.14 – Typical model block diagram implementation – BPCS part

This is a preview. Click here to purchase the full publication.

SOV SIS control logic model implementation

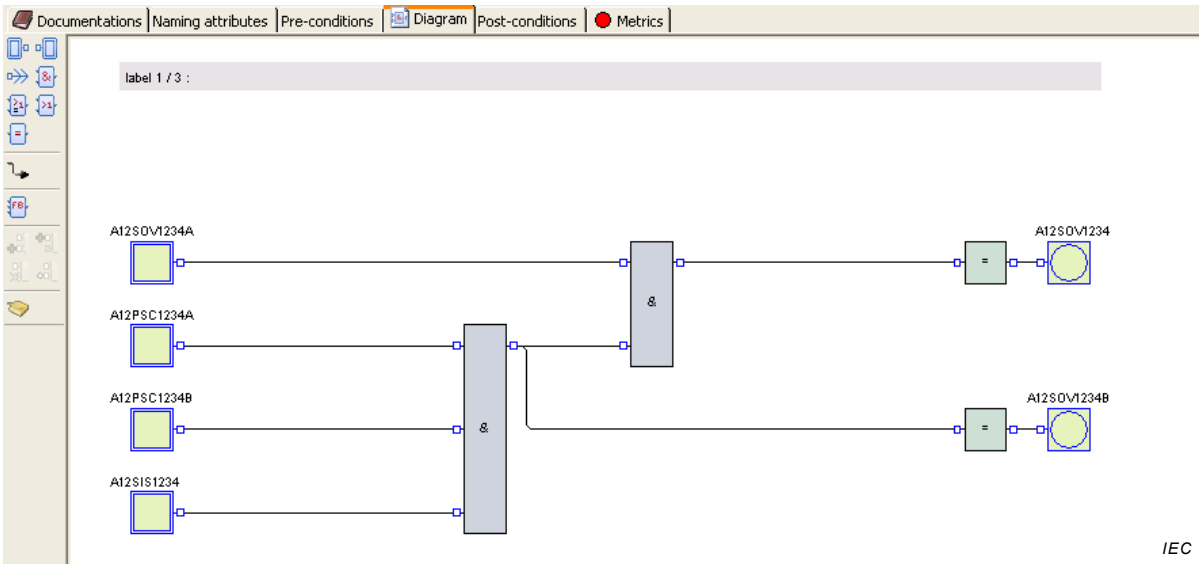

Figure B.15 – SOV application program typical model implementation – SIS part

Modelling continuation:

The same approach is followed to model:

- The safety properties of this SOV;
- The other necessary typicals that would be in this example:
  - 2oo3 voting;
  - 2oo3 safety properties;
  - network communication handling;
  - network communication safety properties;
  - sensor handling;
  - sensor safety properties.

B.4.3.4.2.2 Safety instrumented function level

The same approach applies at SIF level to model:

- SIF 02.01 interlocks model;
- SIF 02.01 safety properties model;
- SIF 06.02 interlocks model;
- SIF 06.02 safety properties model.

B.4.3.4.2.3 Plant level

The same approach applies at SIF level to model:

- plant interlocks model;
- plant safety properties model;
- plant model.

This is a preview. Click here to purchase the full publication.

B.4.3.4.3 Models testing results and defect correction

The models are independently run by the model checking tool in order to detect safe behavior violations. If errors are found by the model checker the concerned models are corrected and run again until they are free from these systematic design faults.

B.4.3.5 Application program integration modelling and testing

The previous steps have enabled the AP engineer to:

- describe an AP architecture;
- describe the functional AP modules content;
- demonstrate that the architecture and the modules specifications were:
  - compliant with AP functional specifications;
  - compliant with AP safety integrity specifications.

This is not sufficient to demonstrate that the AP concept is viable when integrated on the targeted physical hardware architecture because the functional behavior of the AP will depend also on the characteristics of the physical hardware architecture.

This design phase consists in adding models describing the impact of the distribution of the AP in the targeted hardware physical architecture in order to check that the SRS are still met. If it is the not case, the AP modules and/or the AP architecture and/or even the physical architecture would be modified. The step of Figure B.10 can skipped if the physical hardware architecture is known early during AP development life-cycle.

The following models are developed and added to the checking model within the following structure as described in Figure B.16.

This is a preview. Click here to purchase the full publication.

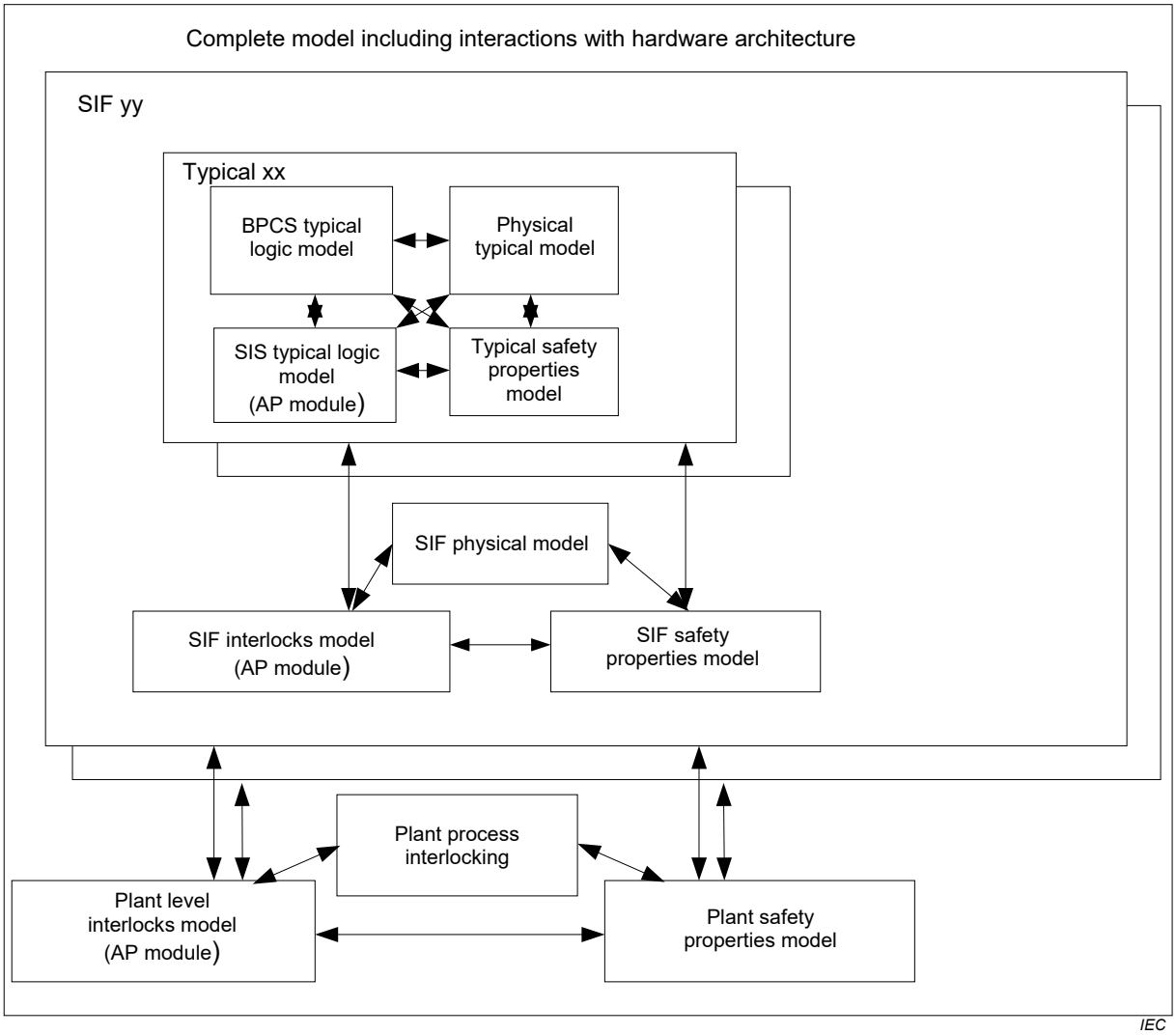

Figure B.16 – Complete model for final implementation model checking

B.4.4 Application program production

Some AP workbenches allow direct downloadable code generation from the models previously checked. Some are, in addition, IEC 61508 compliant.

B.4.5 Application program verification and testing

When it is not possible to automatically generate a downloadable code from an IEC 61508 compliant or prior use tool, the AP is manually generated and tested.

B.4.6 Validation

For an AP generated automatically from the models with a tool compliant with IEC 61511-1:201612.6, the validation consists in verifying through an analysis of the model checker documentation that all functional and safety integrity properties described in the AP SRS have actually been demonstrated.

For a manually generated AP, the validation consists in verifying that the implementation strictly follows the assumption of the models and that all functional and safety integrity properties described in the AP SRS have actually been demonstrated by the tests.

This is a preview. Click here to purchase the full publication.

Annex C
(informative)

Considerations when converting from NP technologies to PE technologies

Many process sector facilities utilise electrical or solid state devices for their SIS design. These facilities may desire to capture the advantages offered with PE technology and as such plan on modification of, or addition to their SIS(s) with the use of PE devices. Some considerations that the facility may consider prior to and during this transition include:

a) PE devices for SISs should not be used unless the plant can successfully verify, validate, utilize, operate, and maintain PE devices in their BPCS;

b) PE devices for SISs should not be used unless the plant has the capability to utilize, design, modify, and maintain AP for PE devices;

c) A review should be carried out to determine what can be transferred from the existing plant application capability to the SIS AP, such as:

- skills and experience (e.g., programming experience, capability, availability); Portability of on-site AP skills to the SIS; management familiarity and involvement with PE support;
- security capability as it relates to allowing access (local and remote);
- application language use in existing facility;
- modules (e.g., existing control algorithms such as motor start/stop/try, comparison of actual position to position command status);
- real time response and delays moving from hardwired to PE systems;
- interfaces (e.g., existing, proven, reliable interfaces to DCS, HMI, networks, electrical interfaces, timing);
- documentation (e.g., capable of providing functionality to clearly describe the AP to maintenance personnel/operators);
- support to trouble-shoot the PE and AP 24/7;
- coverage (availability of support 24/7);
- response (e.g., time to problem resolution);
- simulator (e.g., for off-line AP analysis, modification, development, training);
- testing capability (e.g., plant testing approach and capability versus testing needs of the AP);
- operator experience (with PE and AP);
- plant bypass procedures (and their realization and control using application programming);
- conversion of the plant HMI solution from NP to PE technology;
- training (availability , coverage of application programming, HMI);
- tools (e.g., programming, development, testing); support for utility software;
- management support (existing, sufficient, awareness of need by all parties);
- management of change procedure, including control over modifications to the AP, secure storage and configuration management and provision for re-validation where changes have been made or can affect the AP;
- future enhancement/upgrade/obsolescence of the system(s);
- corporate/plant application programming standards and promulgation of industry codes of practice;

This is a preview. Click here to purchase the full publication.

- compatibility with “sister” PE solutions for interchangeability of experience, processes, equipment etc.

This is a preview. Click here to purchase the full publication.

Annex D
(informative)

Example of how to get from a piping and instrumentation diagram (P&ID) to application program

Annex D illustrates how an oil and gas separation process was evolved from a P&I diagram to an AP. Figure D.1 illustrates the P&I diagram. Figure D.2 illustrates the expression of the safety functions by means of a cause and effect diagram. Figure D.3 shows the conversion of the cause and effect diagram to the AP using function block programming.

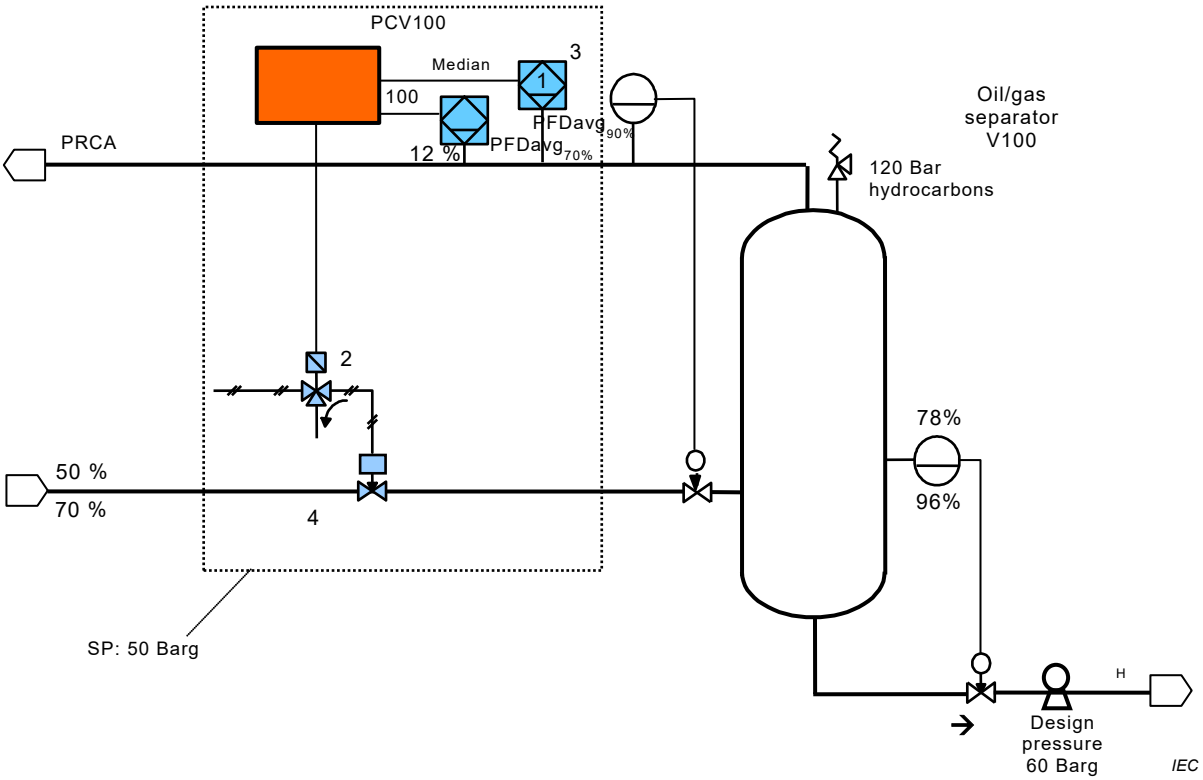

Figure D.1 – Example of P&ID for an oil and gas separator

This is a preview. Click here to purchase the full publication.

LEGEND

THE LINKING MATRIX HAS A NUMBER OF IDENTIFYING SYMBOLS

X - THE INITIATING DEVICE CAUSES THE CONTROL DEVICE TO PERFORM THE FUNCTIONS LISTED

R - THE INITIATING DEVICE CAUSES THE CONTROL DEVICE TO PERFORM THE REVERSE FUNCTION

V - VOTED: ONE OF A GROUP OF THE INITIATING DEVICES MUST TRIP TO PERFORM THE FUNCTION INDICATED

P - BEFORE 'X' OR 'R' MEANS THE INITIATING DEVICE PERMITS THE CONTROL FUNCTION TO BE INITIATED BY ANOTHER DEVICE

T - AFTER 'X' OR 'R' MEANS A TIME DELAY BEFORE THE ACTION TAKES PLACE

I - BEFORE 'X' OR 'R' MEANS THE INITIATING DEVICE INHIBITS THE CONTROL DEVICE FUNCTION

L - ACTION REQUIRED ONLY IF CERTAIN CONDITIONS EXIST

S - START-UP OV

M - MAINTENANCE OVERRIDE

INITIATING DEVICE					FUNCTION						
NOTES	OVERRIDE	SIL	DESCRIPTION	IDENTIFICATION		IDENTIFICATION	DESCRIPTION	ACTION	NOTES	Etc.	
						XY31495	HYDROCARBAN INLET SDV31495				
						1	2	3	4	5	Etc.
8	M	2	SEPARATOR - HIGH PRESSURE	PAHH31403A	1	V					
8	M	2	SEPARATOR - HIGH PRESSURE	PAHH31403B	2	V					
	M, S	1	SEPARATOR - LOW LEVEL	LALL31407	3	X					
					4						
					5						
					Etc.						

Safety Solenoid Valve XY31495

IEC

Figure D.2 – Example of (part of) an ESD cause & effect diagram (C&E)

This is a preview. Click here to purchase the full publication.

Below is an example of an AP for a SIF with two 1oo2 voted analogue signals from pressure sensors and a digital output to a single final element. The safety concept is DTS (de-energize to safe state). The program language is Function Blocks.

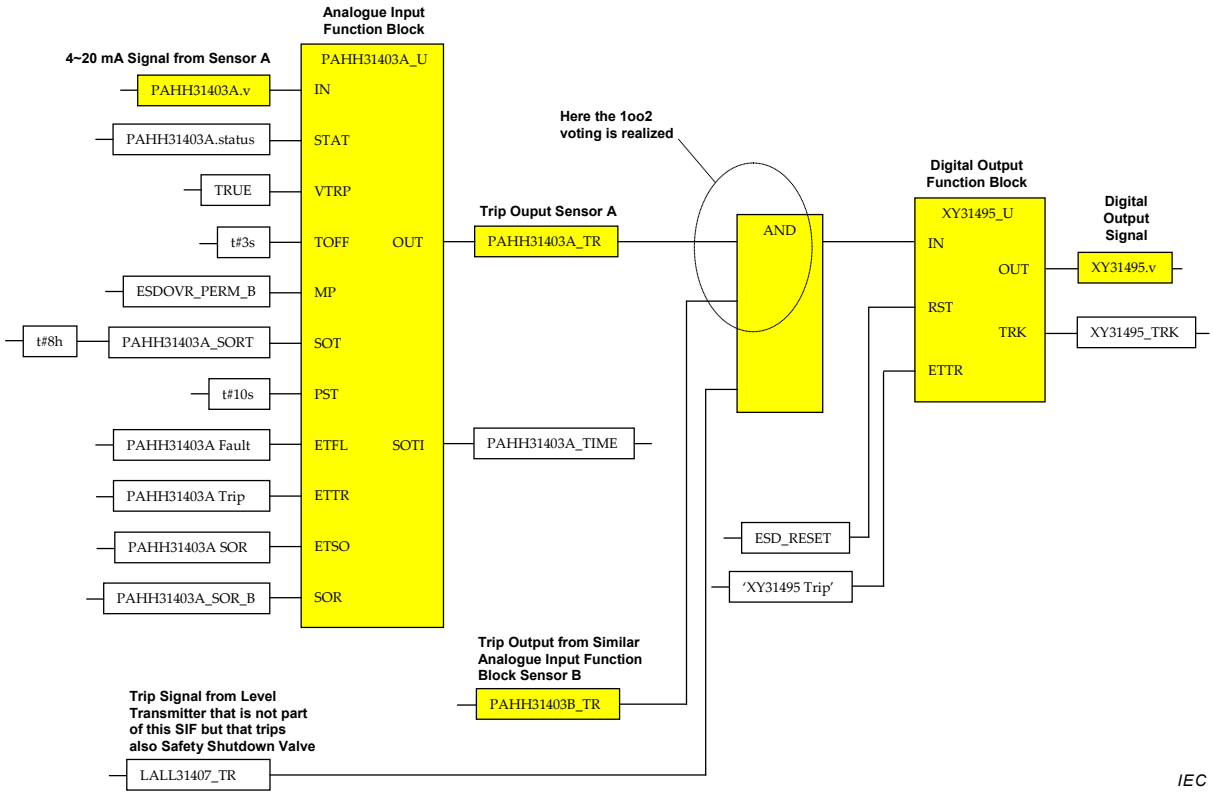

Figure D.3 – Example of (part of) an application program in a safety PLC function block programming

This is a preview. Click here to purchase the full publication.

Annex E
(informative)

Methods and tools for application programming

E.1 Typical toolset for application programming

Typically, the toolset supporting the PE programming will include the following capabilities:

a) Configuration editor. This editor is used to configure the I/O SIS subsystem, the I/O memory variables, and communication functions.

b) Language editors. These editors are used by the application programmer to develop the programs that perform all the functions needed by the system (safety and non-safety).

c) Libraries of previously assessed functions and function blocks. These functions and function blocks can be used in the APs.

d) Custom function and function block development capability. Some suppliers provide a development environment that allows the user to develop custom functions and function blocks that can be used by the supported application languages. These custom functions and function blocks should be thoroughly tested prior to use in the AP.

e) AP scheduling facility. These scheduling facilities support the setting of the order of desired execution sequence and their scan rates.

f) Downloading capability. This allows the developer to download the AP, function block libraries, variable data and other configuration information into the logic solver hardware for execution.

g) Emulation capability. Some suppliers provide a development environment with the capability to emulate all of the APs on the computer that supports the development environment. This allows thorough off-line testing of the APs before they are downloaded into the logic solver.

h) Program monitoring capability. The monitoring capability allows the user to view data from the executing program on user-defined screens or on the actual function block or ladder diagram program screens. The development environment may also provide the capability to monitor the execution of the emulator. In addition, the programs executing in the logic solver can be monitored.

i) Diagnostic displays of the logic solver. These displays show the status of the main processor modules, communication modules, and the I/O modules in the system. Typically, the pass, fail, active status of each module is shown; and in many cases, more detailed information about faults in the system is available.

The PE system will be supported by a programming environment that supports the coding of the AP, the configuration of application parameters and interfaces and the testing/ monitoring of the AP execution. Many PE systems intended for use in safety applications will be supported by a dedicated set of tools, together with a manual which will describe how to use the tools to ensure that the AP achieves the intended integrity. The design environment and application language should possess features that facilitate:

- abstraction, modularity and other features which control complexity; wherever possible, the AP should be based on well-proven modules that may include user library functions and well-defined rules for linking the modules;
- expression of:
  - functionality, ideally as a logical description or as algorithmic functions;
  - information flow between modular devices of the application functions;
  - sequencing requirements;
  - assurance that SIF always operate within the defined time constraints;

This is a preview. Click here to purchase the full publication.