Introduction to Industrial Communication Networks, Integration Systems and Control (PCWorx Express) - Apostilled material of the practical class - 004: Automated Production Control...

Hermes Jose Loschi
University of Campinas
54 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Yuzo Iano
University of Campinas
123 PUBLICATIONS **204** CITATIONS
SEE PROFILE

Alarm management in automated process control system

1) Tasks and functions of alarm unit

Alarm unit is a part of any automated production control system (APCS). Speed of response and elimination of emergency depends on reliability of alarm unit.

Alarm unit can be created for analog or digital channels. Monitoring of digital channel supposes the informing an operator about the changes of digital signal state (for example: switching of relay or limit switch). Monitoring of analog signal supposes the informing an operator when the signal goes outside the normal range (for example: rising of temperature).

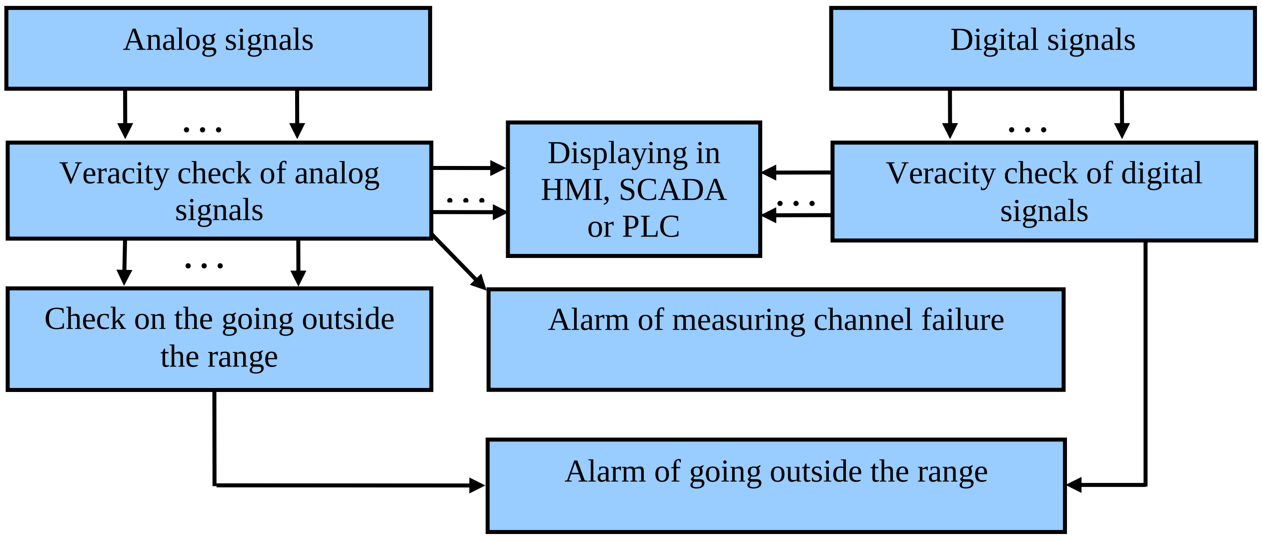

Analog signals
Digital signals
Veracity check of analog signals
Displaying in HMI, SCADA or PLC
Veracity check of digital signals
Check on the going outside the range
Alarm of measuring channel failure
Alarm of going outside the range

Figure 1 – Structure of alarm unit algorithm

The main tasks solving with alarm unit:

a) Monitoring of signals and displaying the current states of signals in PLC, SCADA or HMI.

b) Starting of sound and light signaling at the moment when a signal goes outside the normal range (analog channel) or when a signal changes its state (digital channel).

c) Protection of analog signal from spike, falling or connection line break.

d) Testing of good condition of light and sound alarm signals.

2) Block of veracity checking

According to Fig.1, the first block of alarm's algorithm is a block of checking signal on veracity. The structure of these blocks for analog and digital signals is shown in the Fig.2.

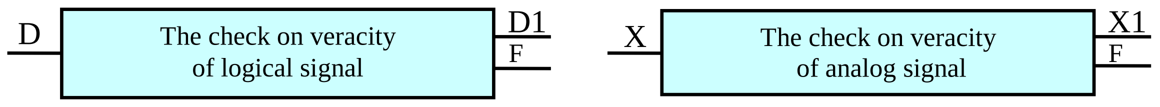

D
The check on veracity of logical signal
D1
F
X
The check on veracity of analog signal
X1
F

Figure 2 – Blocks of checking on veracity for analog and logical signals

In the figure: **D** – logical signal before veracity block, **D1** – logical signal after veracity block, **X** – analog signal before veracity block, **X1** – analog signal after veracity block, **F** – becomes true when a failure is detected.

The checking on veracity for digital channel means the checking on bounce and short disappearance of signal. In the Fig. 3-a a case of bounce (at the moment t1) and a short disappearance of signal **D** (at the moment t6) are shown.

These events must not lead to wrong activity of operator. Therefore such situations are analyzed by checking block and wrong commands do not arrive to automated control system. Changing of output signal **D1** is also displayed in Fig. 3-a.

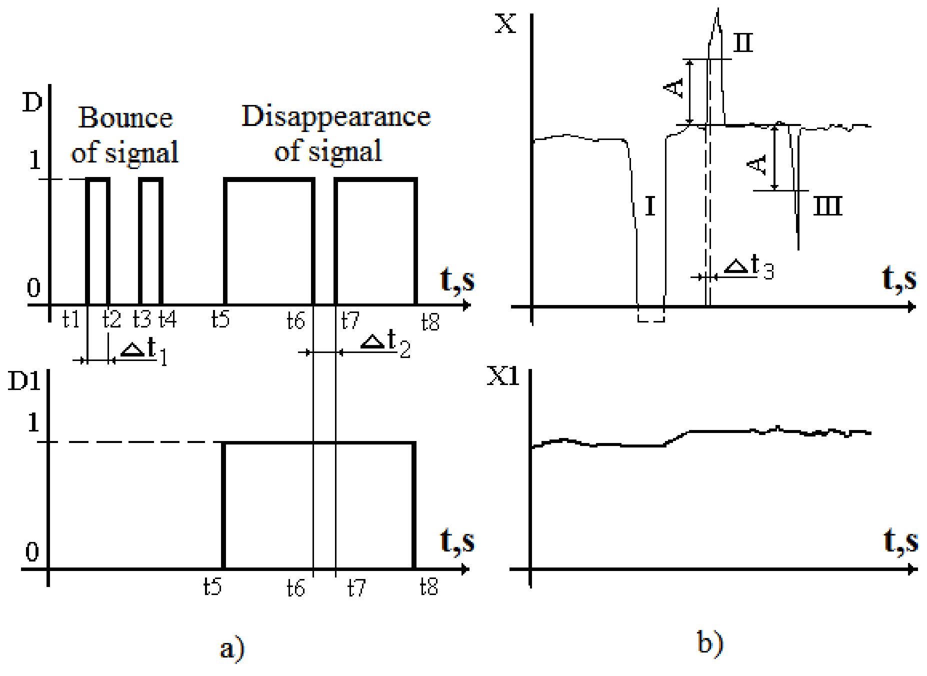


Figure 3 – Graphical illustration of the working of veracity checking blocks

In Fig. 3-b a typical situation for analog channel is shown. The first situation (I) corresponds to break of measuring line. The second situation (II) is an accidently spike of signal. The situation III is an accidently falling of analog signal. The value *A* is an allowable changing of analog signal during Δt_3 .

2.1) Veracity checking block for digital channel

The general structure of function block:

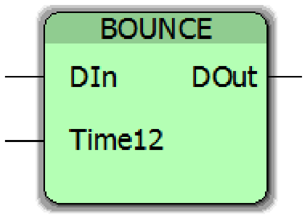


Figure 4 - The general structure of the veracity checking FB for digital channel

Inputs: **Din** – input of signal (BOOL type), **Time12** – time of checking on bounce (TIME type). Output: **DOut** – output of signal (BOOL type).

The **Time12** input must be more than maximum possible time of contact bounce. During the modeling of algorithm this value might be set to one second.

Programming:

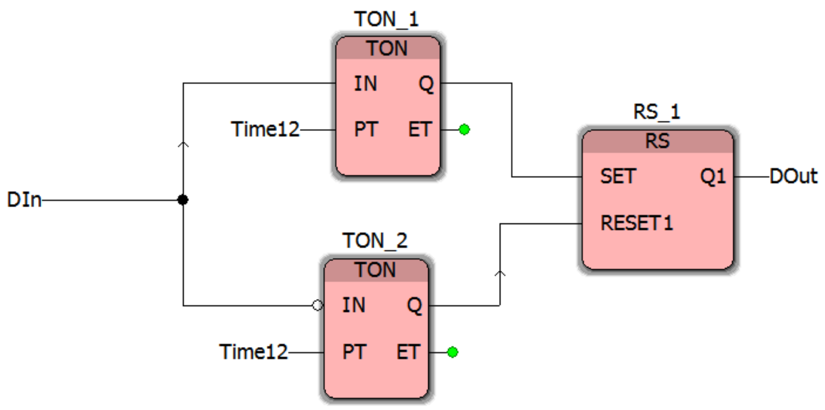


Figure 5 – The implementation of VC function block for digital channel

Description: two on-delay timers (**TON_1**, **TON_2**) and RS-trigger are used in the block. The algorithm actually implements a delay in transport path of signal from input (**DIn**) to output (**DOut**). **TON_1** eliminates the bounce of logical signal, and the **TON_2** prevents a short disappearance of signal (see Fig. 3-a).

2.2) Veracity checking block for analog channel

The general structure of function block:

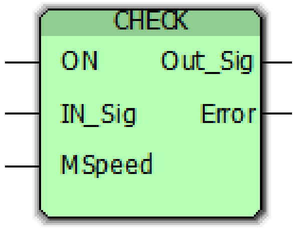


Figure 6 - The general structure of veracity checking FB for analog channel

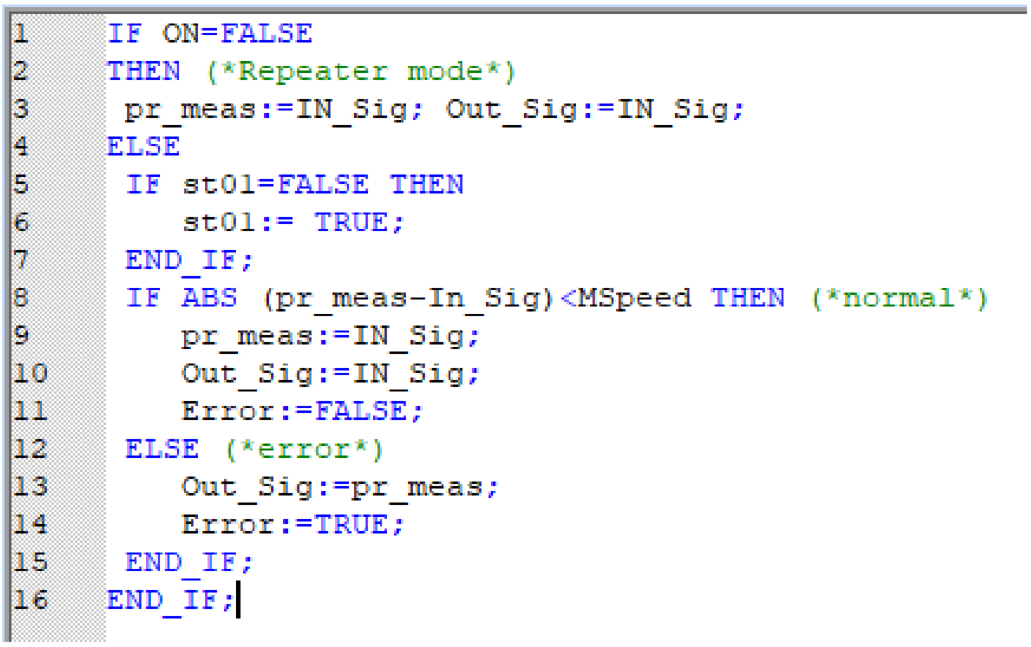
Inputs: **ON** – turn on/off the veracity checking function (BOOL type), **IN_Sig** (REAL type) – input analog signal, **MSpeed** (REAL type) – maximum allowable speed of signal change for signal falling or spike detecting. Outputs: **Out_Sig** (REAL type) – output signal, **Error** (BOOL type) – error flag (signal spike, falling or line breakage was detected).

The FB is developed in ST language (Fig.7) and its principle of operation is based on the periodically measuring of the speed of signal change during a short period of time. It is assumed that a measured signal has an inertia and it can not physically changes much in a short time.

While speed does not exceed the allowable value (**MSpeed**) the signal is transferred to the output (**Out_Sig**) without changing.

Otherwise, the output stores a previous value of signal and the **Error** output becomes TRUE. Once the signal is restored (i.e. the speed of signal changing becomes allowable) the **Error** is reset and the input signal is transmitted to output again.

Programming:



```
1   IF ON=FALSE
2   THEN (*Repeater mode*)
3    pr_meas:=IN_Sig; Out_Sig:=IN_Sig;
4   ELSE
5    IF st01=FALSE THEN
6       st01:= TRUE;
7    END_IF;
8    IF ABS (pr_meas-In_Sig)<MSpeed THEN (*normal*)
9       pr_meas:=IN_Sig;
10      Out_Sig:=IN_Sig;
11      Error:=FALSE;
12   ELSE (*error*)
13      Out_Sig:=pr_meas;
14      Error:=TRUE;
15   END_IF;
16  END_IF;
```

Figure 7 – The implementation of veracity checking block in ST

Description: if the input **ON** is FALSE the program performs code in third line (repeater mode). Line 45 checks the condition of normal speed of signal change and if the signal satisfies it the lines 5-7 are executed. Otherwise, the branch 9-10 is executed (storing the last value).

This algorithm might be created as a cyclic program with a determined time of cycle. In this case the cyclic part can be removed. Also a moving average calculation can be added to the algorithm for more reliability.

3) Multivibrator function block

Multivibrator is a unit that generates a sequence of rectangular pulses. This unit is required for implementation of blinking light signal at a determined frequency.

The general structure of function block:

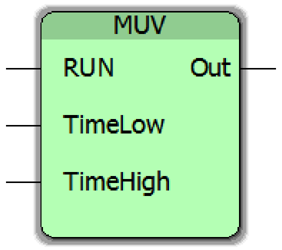


Figure 8 – The general structure of multivibrator FB

Inputs: **RUN** (BOOL type) – turn on/off the multivibrator, **TimeLow** (TIME type) – high-level pulse duration, **TimeHigh** – low-level pulse duration.

Outputs: **Out** (BOOL type) – a sequence of rectangular pulses.

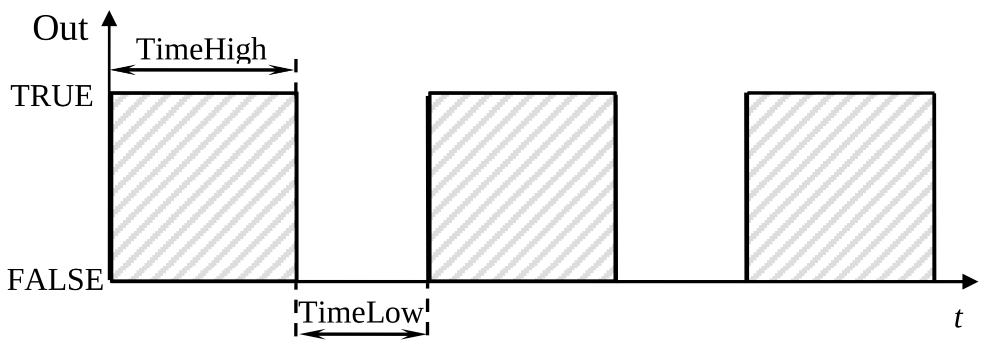


Figure 9 – Illustration of the multivibrator block working

Programming:

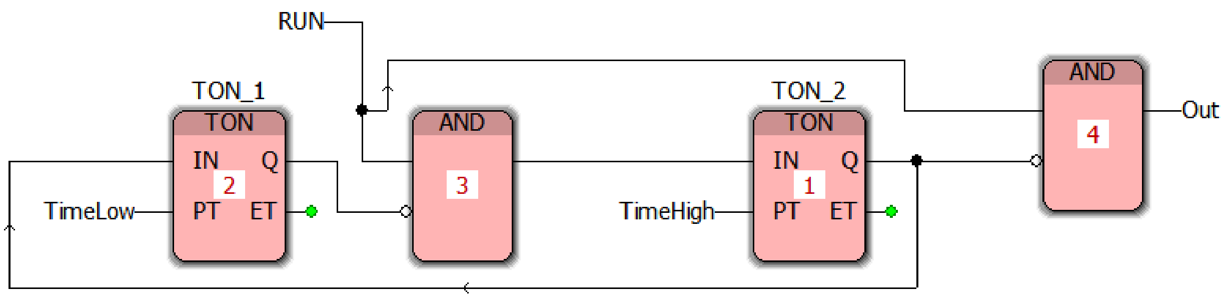


Figure 10 - The implementation of multivibrator function block in FBD

Description: two on-delay timers (**TON_1, TON_2**) are used in the diagram. **TON_1** timer is responsible for high-level pulse duration, and the timer **TON_2** - for the duration of a low level. The variable **RUN** forbids or permits the running of multivibrator by blocks **AND** (3), **AND** (4). Block **AND** (4) turn off the output pulse immediately when the **RUN** becomes FALSE.

4) Block of bounds check

The general structure of function block:

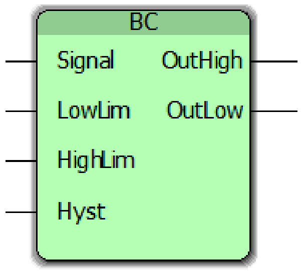


Figure 11 - The general structure of BC function block:

Inputs: **Signal** (REAL type) – input analog signal, **LowLimit** (REAL type) – low allowable limit, **HighLim** (REAL type) – high allowable limit, **HYST** (REAL type) - hysteresis of the high/low limit.

Outputs: **OutHigh** (type BOOL) - indicates when the signal goes outside the high limit with influence of hysteresis, **OutLow** (type BOOL) - indicates when the signal goes outside the low limit with influence of hysteresis.

Programming:

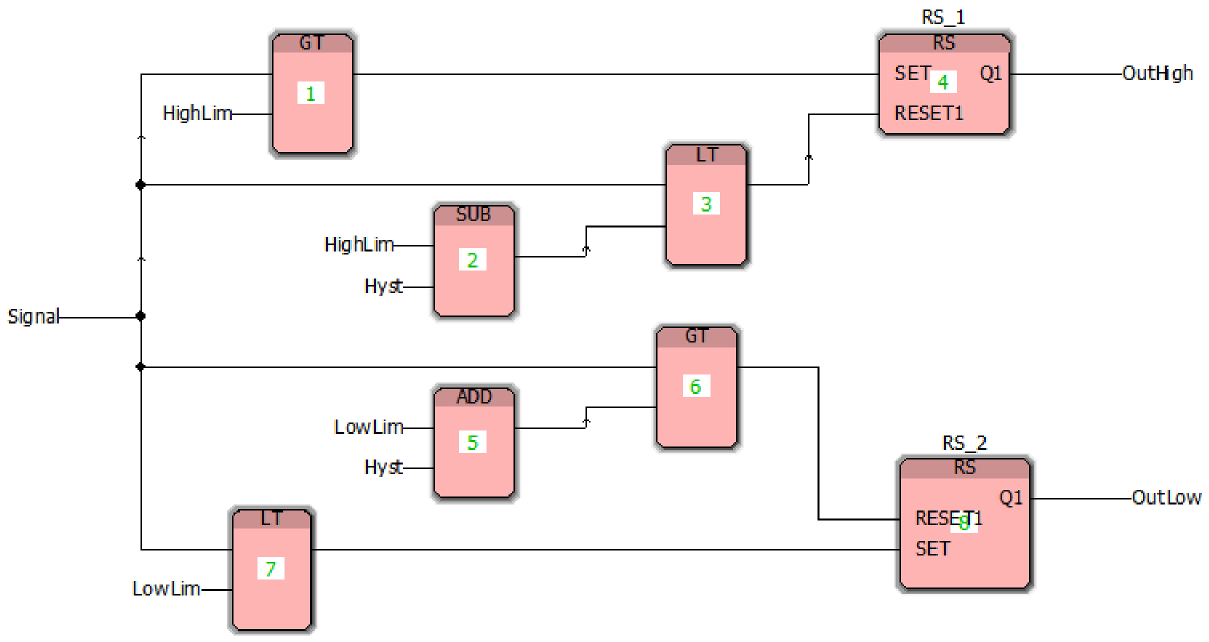


Figure 12 – The implementation of BS function block in FBD

Description: two RS-triggers **(RS_1, RS_2**) are the key elements of diagram. **RS_1** trigger is set when the signal goes out of higher bound (**HighLim**) because of the condition of **GT**(1) block (greater than). **RS_2** trigger is set when the signal goes out of lower bound because of the condition of **LT**(7) block (less than). The triggers are reset when the signal returns into the range and overcomes the hysteresis value (blocks: **SUB**(2), **ADD**(5) and the comparison blocks: **LT**(3), **GT**(6)).

Fig. 13a e 13b explains the influence of hysteresis on the BC block working. Let's assume that there is no hysteresis. Since the actual analog signal always "floats" within the measurement error (sensor and ADC error), the signal can pass the allowable bound and come back several times.

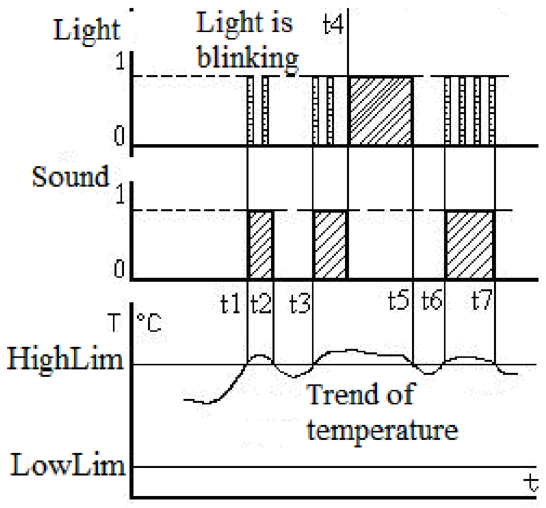


Figure 13a – Trends of light and sound signaling (without hysteresis)

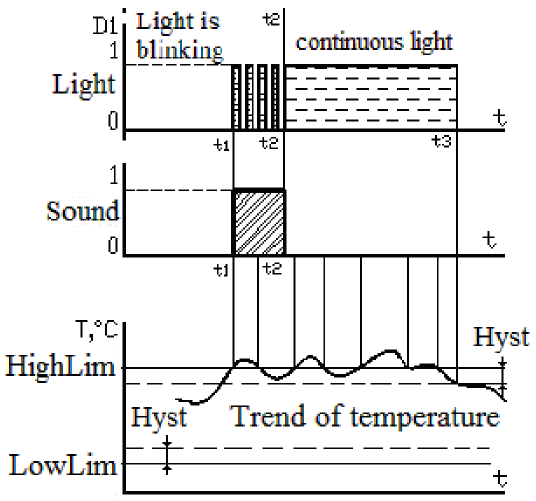


Figure 13b – Trends of light and sound signaling (with hysteresis)

5) Implementation of the alarm unit for one analog and one digital channel

FBD program of alarm unit for one analog and one digital channel is developed with using the function blocks described above (Fig. 16). The program was tested on STARTER kit with ILC 131 ETH controller. Global variables using in program are the onboard digital inputs and outputs.

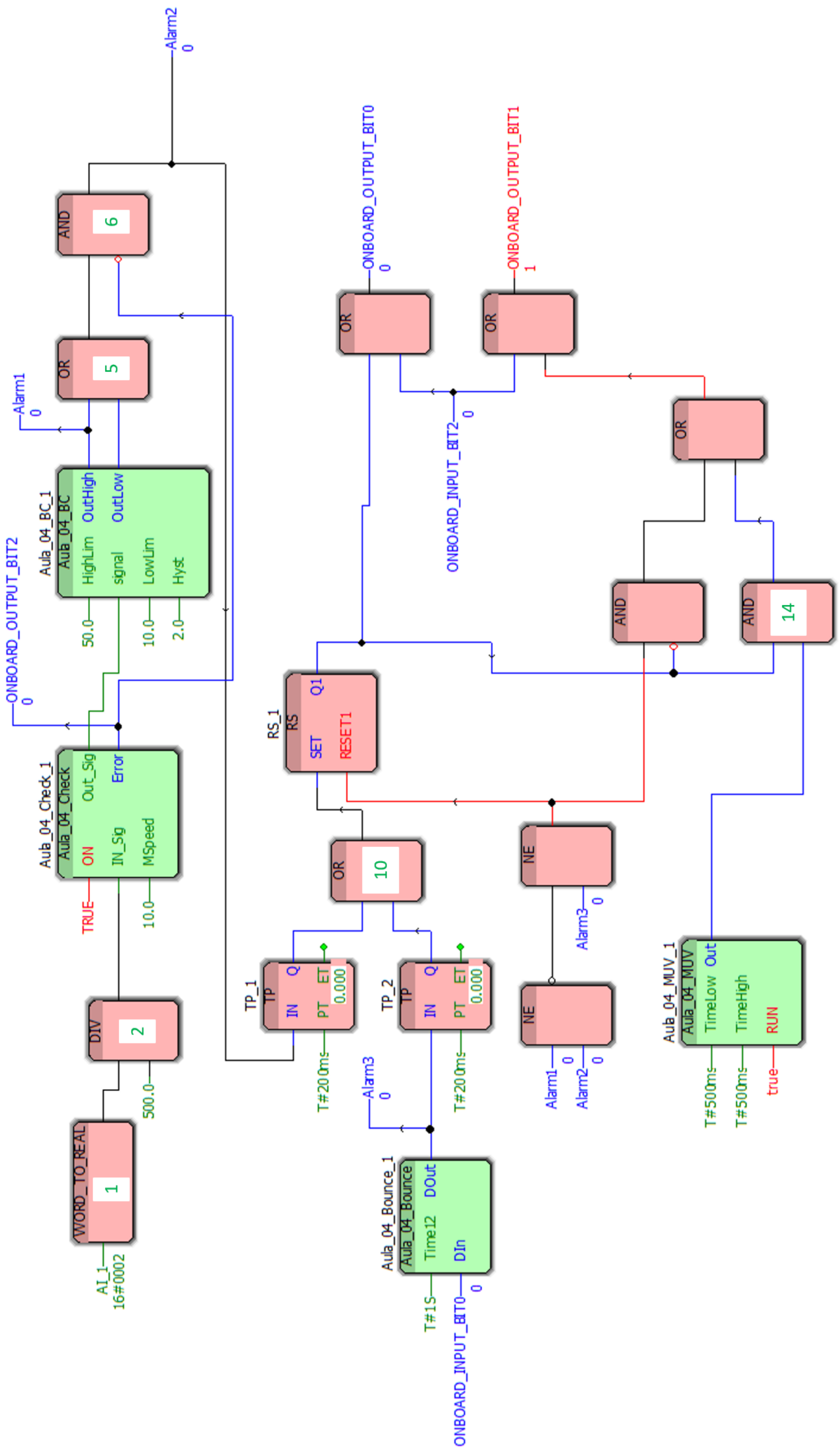
The source of analog signal is a variable **AI_1** (WORD type), coming from IB IL AI 2/SF-ME module. The source of digital signal is a variable **ONBORAD…** (BOOL type). It is assumed that the digital signal is a signal of accident (e.g. coming from alarm relay). So, if it enters the system, then it is necessary to start the sound and light signaling.

Programming in the next page, Figure 14 – FBD algorithm of alarm unit for one digital and one analog channel.

Description of diagram: the analog signal size of 15 bits (0-32767) is scaled to the range of about 0-60 via the blocks (1), (2). After that signal arrives to veracity checking block **CHECK_1** and then to bounds check block **BC_1**. **TP_1** (creates a pulse for the duration of PT time=200ms) through the blocks **OR**(5) and **AND**(6). Block **AND**(6) forbids an appearance of violation flag if a hardware fault is detected (**Error** output of **CHECK_1**, **ONBORAD..2** variable).

The digital signal is checked on veracity by **BOUNCE_1** block and after that it arrives to **TP_2** block.

All signals (of analog and digital channels) are collected in **OR** (10) block. The result flag of violation set the RS-trigger **RS_1**. Also the RS-trigger output multiplying with pulse signal from **MUV_1** in **AND**(14) starts the blinking light signaling (**ONBORAD..1**).


Alarm2
ONBOARD_OUTPUT_BIT0
ONBOARD_OUTPUT_BIT1
AND 6
OR 5
Alarm1
ONBOARD_OUTPUT_BIT2
Aula_04_BC_1
Aula_04_BC
HighLim
signal
LowLim
Hyst
OutHigh
OutLow
50.0
10.0
2.0
OR
ONBOARD_INPUT_BIT2
AND
AND 14
RS_1
RS
SET
RESET1
Q1
Aula_04_Check_1
Aula_04_Check
ON
IN_Sig
MSpeed
Out_Sig
Error
TRUE
10.0
OR 10
NE
Alarm3
TP_1
TP_2
TP
IN
PT
Q
ET
0.000
T#200ms
DIV 2
500.0
WORD_TO_REAL 1
AI_1
16#0002
Aula_04_Bounce_1
Aula_04_Bounce
Time12
DIn
DOut
T#1S
ONBOARD_INPUT_BIT0
Aula_04_MUV_1
Aula_04_MUV
TimeLow
TimeHigh
RUN
Out
T#500ms
true