LOGIC CONTROLLER APPLICATIONS IN SECURITY SYSTEMS
DESIGN TO CONTROL AND MONITOR CIVIL FACILITIES

CRISTINA GABRIELA SĂRĂCIN, MARIN SĂRĂCIN

Key words: Programmable logic controllers, LADDER diagram.

The present article explains a reliable solution regarding the Programmable Logic Controller (PLC) implementation into a security system used to control and monitor a four floor building. The application is split in four different sections: secured access, motion detection, command and monitor elevator and alarm triggering in case of a fire. The PLC integration in management processes is often encountered due to its less surveillance, programs easy update and remote diagnosis accessibility.

1. INTRODUCTION

Having at our disposal the TWIDO, MICRO and PREMIUM PLC products, made by the Schneider Electric company, a PLC from MICRO category was used to implement the security application. The PLC can be configured for 24 digital inputs, 17 digital outputs and 3 analogical entries.

The base unit of the PLC Micro TSX 37-21 [1] is powered from a 220 V ac electric source and has built-in a processor with 20Kwords available memory able to store data, constants, program instructions and a nonvolatile memory. The PLC-PC dual communication is achieved via a serial interface (RS 485) and is based on Uni-Telway [2] and ModBus [3] protocols. The system digital data acquisition was performed by the TSX DEZ 32D2, an independent standard module with 32 inputs. Its functioning is based on positive logic and the sensors connection to the model is made by two wires. The generated signals are received, converted, electric isolated and protected against external influences. The system analogical data acquisition has been accomplished based on the half independent module of the slot, TSX AEZ 801. This module does not require a separate power source and it is capable to power itself from the base unit power source. The range domain of the inputs could be ±10 V or 0–10 V. For a better reliability, this module does not have to contain into the measurement link any electro-mechanical components such as multiplex relays, switches or potentiometers. According to the process data monitoring, it has been

University “Politehnica” of Bucharest, E-mail: csaracin@electro.masuri.pub.ro

necessary an independent standard module, TSX DSZ 32R5, with 32 outputs per relay. The module functions in positive logic, stores the instructions supplied by processor and after, controls thru processor the execution elements of the process. Also, a 24 V cc electric source was necessary to power the digital inputs and outputs of the modules.

2. THE APLICATION PRESENTATION

The application is written in PL7 and presents the functioning of a PLC as part of a security system design to monitor and control a four floor building. The application consists of four sections and fifteen subroutines. The first section is referring to the pedestrian access into a secured space, the second section detects the inside movement and turns on the light, the third section controls the elevator activity and the fourth section monitors the protection against fire.

The first section presents the pedestrian access into a secured space based on a magnetic card and a numeric code. The variables declared to accomplish this section are the following ones: ● three buttons for three digital inputs having the input addresses such as %I1.1, %I1.2, %I1.3 respectively Btn1, Btn2, Btn3 as symbols; ● one magnetic sensor magnetic for the digital input %I1.0 and M_sensor as sensor symbol; ● one button to reset the declared code thru digital variable %I1.4 with Reset_cod as symbol; ● two digital inputs having %Q3.0, %Q3.1 like variables and Door_unlook, Alarm_Set as symbols.

In Fig. 1, the input of the first digit can be observed and must be “3” using button three. The initialization of the Fig. 2 and Fig. 1 is performed in the same mode.

![Fig. 1 – The diagram used to initiate the access code input.](image-url)
In Fig. 2 it is presented the system diagram since the correct code is inserted and the door is unlocked and, until after 5 seconds the door is locked.

Fig. 2 – The diagram used to unlock/lock the door.

The application has been written in LADDER language and the explanation of the above diagram is the following one:

- to make the system to respond to the button pressing (the three figures) it is necessary that the %M0 variable to be 0 and %I1.0 to be 1; this means the system is not blocked and the magnetic card is present;
- four variables such as %M4, %M1, %M2 and %M3 exist and they correspond to the four status in which the system can be when it’s not blocked, when it was inserted any figure of the code and the inputs are not 1, 2, 3 i.e., the correct code figures;
- if it is inserted the correct figure the system passes in the next status from %M4 = 1 and rest 0 it is passed to %M1 = 1 and rest 0,..., respectively from %M3 = 1 and rest 0 is passed to %M4 = 1 and rest 0;
- when %M3 is active, the %Q3.0 output it’s activated, the door it’s unlocked and the system stays five seconds in this status and after it is translated in %M4 so the door is blocked again;
- when it is pressed a correct button belonging to the code, a timer delay of 1 second is inserted to avoid multiple insertions of the same button; in this case the status passing is done only when the button is kept pressed for 1 second minimum;
- when the system is blocked (%M0 = 1), the %Q3.1 output is set on 1 and the alarm is activated;
- when the reset input %I1.4 is activated, the Block is set on 0, %M4 on 1 and %M1, %M2, %M3 are set on 0, system is unblocked, the alarm is indirectly deactivated and the system goes on stand by status;
- if the magnetic card is pull out and the system is not blocked, the status will be the initial one and %M4 = 1 with rest 0.

The second program section implies the motion detection for each floor and turning the lights on for 1 minute. The inputs variables are 5 sensors, one for each floor: %I1.5, %I1.6, %I1.7, %I1.8, %I1.9, and Ground_floor_m_s, First_floor_m_s, Second_floor_m_s, Third_floor_m_s, Fourth_floor_m_s as symbols. The outputs variables are %Q3.5, %Q3.6, %Q3.7, %Q3.8, %Q3.9, Ground_floor_light, First_floor_light, Second_floor_light, Third_floor_light, Fourth_floor_light.
In Fig. 3, it is described the system diagram used to turn on the lights on the ground floor level.

The third section refers at the command and control of an elevator located at a four floor building. The list with the variables names used is the following one:
- the digital input variables %I2.0, %I2.1, %I2.2, %I2.3, %I2.4, %I2.5 with Ground_floor_cmd, First_floor_cmd, Second_floor_cmd, Third_floor_cmd, Fourth_floor_cmd, respectively Rem_ctrl as symbols;
- the analogical input variable: %IW5.0 where Elevator_weight represents the elevator passengers weight;
- the digital output variables: %Q4.0, %Q4.1, %Q4.2, %Q4.3, %Q4.4, %Q4.5 with Ground_floor, First_floor, Second_floor, Third_floor, Fourth_floor as symbols and define where is the elevator, Alarm as the alarm signal in case of overloading, i.e. the Weight is greater than 320 kg.

In Fig. 4 is presented the main application program which controls the elevator movement based on passengers’ request.

The application will determine the maximum weight and remote control activation. This means that, if the passengers weight is over 320 kg (read on the
analogical input %IW5.0) and the remote is not activated, the elevator will not start and the warning light blinks (%Q4.5). In case the passengers’ weight is less than 50 kg, the elevator will not start. This will prevent the elevator using by unescorted children. The elevator can circulate to any floor, where is called due to the commands receive from the digital inputs (%I2.0, %I2.1, %I2.2, %I2.3, %I2.4). Due to the PLC digital outputs (%Q4.0, %Q4.1, %Q4.2, %Q4.3, %Q4.4), the floor number where the elevator is located will be displayed on the elevator control panel and on the PLC screen.

Within this program were used 15 subroutines, the first five Sr0, Sr1, Sr2, Sr3, Sr4 are used by the main program and the rest are operated by other subroutines. So, Sr5, Sr6, Sr7, Sr8, Sr9, Sr10, Sr11, Sr12 are used to command the elevator with one floor up or down. These subroutines have as inputs or outputs parameters the variables from Table 1.

| Table 1 |
| Subroutines variables presentation |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| Sr0 | Sr1 | Sr2 | Sr3 | Sr4 |
| Input variables | %Q4.4, %Q4.3 | %Q4.0, %Q4.4 | %Q4.0, %Q4.1 | %Q4.0, %Q4.1 |
| | %Q4.2, %Q4.1 | %Q4.3, %Q4.2 | %Q4.4, %Q4.3 | %Q4.2, %Q4.4 |
| Output variables | Call Sr12, Call Sr11 | Call Sr5, Call Sr12 | Call Sr5, Call Sr12 | Call Sr5, Call Sr6 |
| | Call Sr13, Call Sr7 | Call Sr11, Call Sr9 | Call Sr12, Call Sr11 | Call Sr8, Call Sr12 |

The subroutine 5 has like input variable, the initial status of the elevator (ground floor) what is next reset and like output variable, the first floor activated after a 5 seconds delay. The rest of the subroutines perform the same only the variables differ according to table 1. The subroutines Sr13 and Sr14 perform the elevator movement from the second floor to the ground floor and from the second floor to the fourth floor. The subroutines Sr0, Sr1, Sr2, Sr3, Sr4 perform the movement to the ground floor, first, second, third and fourth floor of the elevator, no matter of at what floor is the elevator. By example, the subroutine Sr0 performs the movement in the following mode: if the elevator is at the fourth floor, the Sr12 is called; if the elevator is at the third floor, the Sr11 is called; if the elevator is at the second floor, the Sr13 is called; if the elevator is at the first floor, the Sr7 is called. The rest of the subroutines perform the same only the variables differ according to Table 1.

Section 4 of the application refers to the building protection against fire. The list with the variables names used is the following one: • the digital input variables (%I2.6, %I2.7, %I2.8, %I2.9 where this are Close/Open_Alarm,
Smoke_s_detection, Flame_s_detection, Panic_alarm_b); • the analogical input variable (%IW5.1 which represents the temperature); • the digital output variables (%Q4.7, %Q4.8, %Q4.9, %Q4.10 i.e. Alarm_fire_set, Fire_warning, Fire, Siren).

This alarm system operated based on the diagram located in Fig. 5 and involves the following script: • alarm on (%I2.6) on output there is alarm on indicator (%Q4.7); • if it is detected the smoke signal (%I2.7) or the fire button is pressed (%I2.9) or the temperature becomes more than 50°C on output there is fire warning indicator (%Q4.8); • if it is detected one on the above situations and a signal is received from the flame sensors (%I2.8) on output there is a fire indicator (%Q4.9); • if it is detected the possible fire (%Q4.8) or fire indicator (%Q4.9) the fire siren it is started (%Q4.10).

![Fig. 5 – Alarm diagram in case of fire.](image)

3. CONCLUSION

The PLC applications perform data acquisition, data computing and data monitoring for a large scale of industrial processes. The major advantages in using the programmable logic controllers consists in their independence versus computers but, only after the application was configured and transferred in the PLC non-volatile memory. The implementation of one PLC in a complex process such as security, monitor and control, represents a major advantage. Also, if the PLC will have attached a communication module such as TSX ETY 410, the PLC will communicate thru the TCP/IP protocol, data from the system. The remote control capability (even thru a browser) represents a major benefit due to the operator absence in the area where the PLC operates.

*Received on September 30, 2008*
REFERENCES