|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Modeling and Automation of Industrial Processes***  *MSc in Electrical and Computer Engineering*  *Scientific Area of Systems, Decision, and Control*  *2021 / 2022* |  | *Group: \_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_* |

***1st Lab. - Alarm System for Intrusion Detection [[1]](#footnote-1)***

***Part A - System Components***

This work aims the implementation of an intrusion detection alarm system, in a restricted space as a single room retail store. The intrusion will be detected resorting to an infrared sensor, installed in such a way that it points towards the main entrance of the space to be protected. An open/close switch is also installed on a window of the aforementioned space.

The automation system that constitutes the alarm is to be implemented in the Schneider **PLCs available on the laboratory**, model Premium TSX P57 1634M or TSX P57 2634M, see Fig1(a) or (b). In addition, there is available **simulated-hardware**, wheredigital inputs and outputs are PLC memory locations, which are read and written by the fieldbus Modbus, see Fig1(c).

|  |  |
| --- | --- |
| (a) Setup with one input and one output modules  (b) Setup based on a single input and output module | (c) Simulated-hardware setup. The input and output are mapped on the PLC memory. |

Fig.1: PLC and terminal setups, real (a, b) and simulated (c). The virtual terminal, shown in (c), formed by switches and LEDs, is simulated in Matlab, and connects to the PLC simulator based on Modbus.

This document is composed by two sections: the first describes the guidelines for the first session on the lab and the second (see annex A) describes the functional specifications of all the work that will be developed in the next weeks.

**General characteristics of the alarm:**

The Alarm has three main modes of operation, namely *OFF*, *Presence Detector* and *Active*. The three modes are selected by a three positions switch. The general characteristics of the three modes of operation are detailed next:

**(Mode 1) OFF** – this mode deactivates the alarm completely.

**(Mode 2) PRESENCE DETECTOR** – the infrared sensor is used to detect movement in the room, and signalize it using both a lamp (LED) and the buzzer on the panel. The lamp should be on for 5 seconds, upon the detection of each person, and an acoustic signal with the duration of 1 second should be emitted.

**(Mode 3) ACTIVE** – in this mode the alarm is to be used.

Detailed specifications for mode 3, **ACTIVE**, are the following:

a) When requested for activation, a period of 30 seconds of inactivity is set to allow the user to abandon the space, and afterwards remains permanently activated.

b) Upon intrusion detection, by the infrared sensor or the window switch, the alarm evolves to the warning phase.

c) The alarm lights a warning on the panel and after 5 seconds the buzzer must be activated. The warning must be a periodic signal with 1 second ON and 2 seconds OFF.

d) The alarm can be deactivated pressing the # key on the command panel.

**Implementation details**

In order to implement the general characteristics of the alarm in a clear manner, it is important to separate into one sub-routine the mode 2, Presence Detector, and to another sub-routine the mode 3, Active. More sub-routines can be added in case they help making the code clearer.

In order to help the readability of the proposed solution, the identifiers used in the PLC program must have readable names. For example, instead of using "%I0.2.0" it is more readable to use "I\_switch1\_pos1". The following naming convention is therefore suggested: I\_xyz for inputs, O\_xyz for outputs, M\_xyz for memory variables, TM\_xyz for timers, SR\_xyz for sub-routines, etc, where xyz denotes meaningful names.

Note1: in this part of the work **the infrared sensor is replaced by the two positions switch**.

Note2: The **simulated-hardware** setup uses memories %M. More in detail, memories %M0 till %M19 (type EBOOL) and %MW180 till %MW199 (type INT) are reserved for simulated IO and communications. When the **memory variables represent inputs or outputs**, it is recommended to still use names as I\_xyz or O\_xyz. Names M\_xyz are used just in case of PLC memories are not representing digital inputs or outputs.

Note3: A PLC program/project can be done **simultaneously for the real and the simulated setups**, by adding in the beginning of the scan cycle code as

%m20:= %i0.2.0; %i0.2.0:= write\_input\_ebool( %i0.2.0 OR %m0 );

and at the end of the scan cycle code as

%m10 := %q0.4.0; %i0.2.0:= write\_input\_ebool( %m20 );

The indicated code is to adapt and repeat for as many inputs and outputs as needed in your setup.

**Report questions**

*Note: this guide is distributed in an editable form so that it can be filled and then submitted online.*

**A1.** *(Real and simulated hardware diagram)* Draw a hardware diagram corresponding to the PLC and terminal hardware (Fig. 1a) combined with the simulated alarm. Use the elements listed in Fig.2. Indicate the physical addresses for the real inputs and outputs.



Fig.2: Components to create a real and simulated hardware diagram. Connect these components to make a complete diagram (double click the figure to complete the connections).

Detail: Fig.2 shows hardware components, **switches**, **LEDs** and **buzzer**, that are to connect to the **PLC**. Do not forget to include the **power suppliers**. In addition, there is one virtual terminal, ***myterminal5***, that allows writing to and reading from the **memory** of the PLC.

**A2.** *(Real and Simulated-hardware identification)* Identify the inputs (switches) and outputs (LEDs and buzzer) of the simulated intrusion-detection alarm-console and fill the next table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input (chose the name to use)** | **PLC Identifier (input address %I or memory address %M)** |  | **Output (chose the name to use)** | **PLC Identifier (output address %Q or memory address %M)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*(Insert as many lines as needed to the tables)*

**A3.** *(IO memory)* The distributed sample code uses PLC memories (%M0 till %M19) to represent binary inputs / outputs. Compare the operations "set value" and "force value", done with Unity Pro, on a memory representing the binary input.Suggestion: propose a small **Ladder** program where a memory bit copies its state from another memory bit, and try set or force actuations on the two memory bits.

**A4.** *(Hardware test program)* Design and test a **Ladder** program that verifies the proper functioning of the output devices available in the alarm console. More precisely, the program must beep the buzzer 0.5 seconds and then turn ON the three lamps (LEDs), one lamp after the other, 0.5 seconds each. Create **one project with one MAST section** to test this sub-routine.

**A5.** *(Multiple accesses to the buzzer)* Create a **Structured Text** sub-routine where the buzzer can be turned ON by one of two alternative memory variables (EBOOL), and can be unconditionally turned OFF by another memory variable (EBOOL). Create a fourth way of actuating the buzzer, also commanded by a memory variable (EBOOL), where the buzzer sounds with a periodic signal, 1 second ON and 2 seconds OFF. If the buzzer is **commanded to be simultaneously ON and alternating ON/OFF** then **select alternating ON/OFF**. Demonstrate this sub-routine by using switches or pushbuttons on the virtual terminal.

**A6.** *(Timers)* Consider the alarm application in this and the next questions. List and describe the timers that will be used, their function, and the delay times to be selected for each one.

|  |  |  |  |
| --- | --- | --- | --- |
| **Timer name and/or physical address** | **Operation mode** | **Time Delay** | **Short description of the  usage of the timer** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*(Insert as many lines as needed in the tables)*

**A7.** *(Sub-routines)* List the sub-routines to implement in the alarm application. Indicate for each sub-routine its name and list the variables that are shared with other routines or sub-routines. Classify the listed variables as "input", "output" or "input and output".

|  |  |  |  |
| --- | --- | --- | --- |
| **Routine name** | **Variable name** | **Input, Output or Input & Output** | **Short description of the  variable** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*(Insert as many lines as needed in the tables)*

**A8.** *(Demonstration on the hardware)* Design one or more **Ladder** or **Structured Text** sections to solve the aforementioned automation problem. Create one *main* section in **Structured Text** where the memory variable M\_MODE is used to indicate the current working mode of the complete system and actions are taken according to it.

**A9.** *(Experiment time plot)* Use data logging to document an experiment, i.e. make a time plot of binary variables, considering two usages of the alarm: (i) the alarm is activated, stays activated for a while, and then is deactivated without any intrusion, (ii) the alarm is activated, then, upon intrusion, activates the buzzer, and finally the alarm is disarmed. Start by downloading the demonstration file data\_log.zip and reading more instructions from the website:

http://users.isr.ist.utl.pt/~jag/course\_utils/plc\_log/plc\_data\_log.html

Notes: The referred zip file has one PLC project and Matlab scripts demonstrating PLC data logging. Considering the simulated hardware setup, please replace in the PLC demo code the physical addresses %i0.2.x and %q0.4.x by %i0.3.x and %q0.3.y, where y=x+16, in case you have a DMY28FK module. In case you want to do a simulated experiment the physical addresses have to be replaced by memory addresses.

**A10.** *(Comments on the system built)* Comment on how the complete program runs.

# Annex A - Functional Specifications of the Alarm

*(please do NOT include this annex in your report)*

The Alarm has three main modes of operation, OFF, Presence Detector and Active. The three modes are selected by a three positions switch. The three modes operate as detailed next:

**(Mode 1) OFF** – this mode deactivates the alarm completely.

**(Mode 2) PRESENCE DETECTOR** – the infrared sensor is used to detect the movement on the room/space, that be signalized resorting both to a lamp and to the buzzer on the panel. The lamp should be on for 5 seconds, upon the detection of each person, and an acoustic signal with the duration of 1 second should be emitted.

**(Mode 3) ACTIVE** – in this mode the alarm is to be used.

Detailed specifications for mode 3, **ACTIVE**, are the following:

a) When requested for activation, a period of 30 seconds of inactivity is set to allow the user to abandon the space, and afterwards remains permanently activated.

b) Upon intrusion detection, by the infrared sensor or the window switch, the alarm evolves to the warning phase.

c) The alarm lights a warning on the panel and after 5 seconds the buzzer must be activated. The warning must be a periodic signal with 1 second on and 2 seconds off.

d) The alarm can be deactivated pressing the # key on the command panel.

**Advanced Characteristics of the Alarm:**

An advanced alternative for the alarm activation/deactivation consists on the use of a code previously set by the human owner (e.g. 9887). To implement the activation function, the following procedure must be implemented:

a) switch the alarm mode to ACTIVE.

b) introduce the activation code (e.g. 9887).

c) press #, and wait for 30 seconds to allow the user to abandon the space.

d) start the intrusion detection function, i.e. the alarm is fully operational.

To deactivate the alarm, upon intrusion detection or to allow the use of the space, the following instructions must be accomplished:

a) Introduce the secret code (the same as the activation one, e.g. 9887).

b) Press #

c) Change the alarm mode to a mode other the ACTIVE.

**Special Characteristics of the Alarm:**

A safer mode of operation for the intrusion detection alarm is to allow the user to change the activation/deactivation code. The code 1234 is initially used, as a factory preset. To change the code, the following operations must be done:

a) Press \*, followed by the pre-programmed code.

b) Introduced the new code to be used, finished by \*

In the case where a mistake occurs, press the code \*\*\*\* to reset the code to the factory default.

**Available Material**

In the laboratory there are six different working places, all with similar PLCs but different consoles. All workplaces have a PLC Schneider model P57. All of them have a power supply with 24V and/or 12V and a desktop PC, with the Unity Pro v6 development software and the PLC manuals, in PDF format.

In each workplace there will be also an alarm console with the following components:

|  |  |
| --- | --- |
| 1 three positions switch  1 two positions switch  3 LEDs  1 keyboard (4x3 buttons)  1 buzzer (12V) |  |

The solution for this automation problem must be based on the languages described on the IEC-61131-3 standard, i.e. ladder diagrams, instruction list and structured text.

1. Original guide by Prof. Paulo J. Oliveira. 2022 revision by Prof. José Gaspar. [↑](#footnote-ref-1)