|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Industrial Processes Automation***  *MSc in Electrical and Computer Engineering*  *Scientific Area of Systems, Decision, and Control*  *Winter Semester 2020/2021* |  | *Group: \_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_* |

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

# *Part C - Integration*

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 points towards the main entrance of the space to be protected. A switch is also installed on a window of the 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. *Important: while in effect the current suspension of presence classes, set to protect public health, this laboratory assignment will be based on* ***simulated-hardware****, where* ***digital inputs and outputs are PLC memory locations****, which are read and written by the fieldbus Modbus.*

|  |  |
| --- | --- |
| (a) Virtual terminal that communicates with the PLC. Keys are available provided keyboard columns are energized. The PLC string buffer, a null terminated string starting at %MW190, is displayed when the terminal is "Running". | (b) Options for communication, simulation times, strings logging and terminal refresh. The commands "show" report mostly on the Matlab command line. Command "Logged strings show" reports in a new window. |

Fig.1: Virtual terminal running in Matlab connects to the PLC simulator using Modbus (a). The virtual terminal allows logging strings after activation of option "Log final strings" (b). Logged strings display is available by clicking "Logged strings show".

In this part of the work the 12-keys keyboard available in the console is combined with the switches to build the primary interaction device with the alarm system under design. Sequences of keys must be validated resorting to the subroutines implemented in the previous part of this work. The integration of the complete system is suggested to be done in ***GRAFCET / Sequential Function Chart (SFC)***.

**Implementation guidelines**

The basic presence detection and active alarm system already developed is supposed to be running continuously. Note however that the function of selecting the operating mode (presence detection / active alarm) cannot be based just on switches, must be moved to an upper integration level (SFC).

Keyboard reading is also a function assumed to be running continuously. The upper integration level can validate codes saved in a buffer. Keyboard input is cleared if a timeout occurs or any other condition recommends flushing keys already read.

It is recommended to allow testing the hardware before entering into the active alarm mode. This option can be made available before entering the activation code by pressing the # key of the keyboard. After the hardware test, and in case the active alarm mode switch is ON, one can enter the activation code and therefore enter the active alarm mode.

In order to facilitate the interaction, any return to the OFF mode must be marked with 3-short-beeps done with the buzzer. A suggested timing is 0.02sec for each beep and a pause of 0.1sec in between.

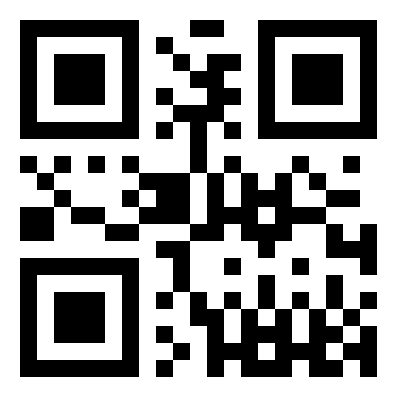
**Summary of the assignment**

Please develop, implement, and test one or more sections in *GRAFCET / SFC* that, according with the interaction with the user, activates a presence detection mode, validates sequences of keys to activate the alarm, to change the code, restart the code to factory settings, as well as other specifications in the annex. Please write the receptivity functions in *Ladder*, *Instruction List*, or *Structured Text*.

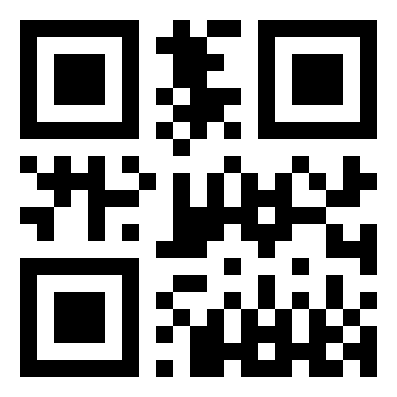
**Report questions**

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

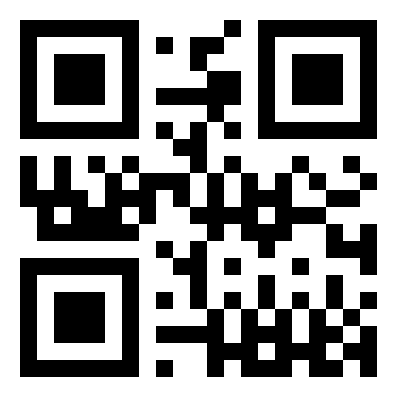
**C1.** Represent asa finite state machine the upper integration level of the complete system. Clearly identify the OFF mode, the presence detection mode, the active alarm mode, and the other states necessary for the complete system.



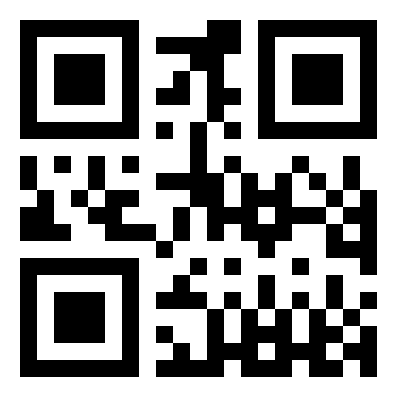
**C2.** Design a **GRAFCET / SFC** program representing the upper integration level. Please use macro steps to help making simple the upper integration level. Describe the implemented receptivity sections. Please write comments within the code to improve readability.



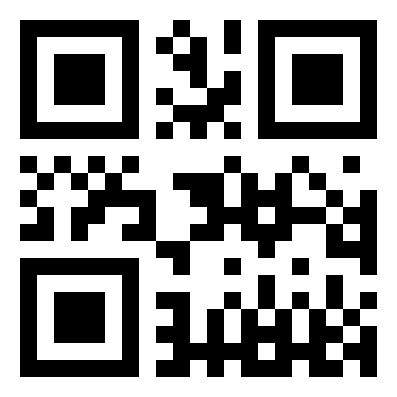
**C3.** Describe the **methodology used to validate sequences of keys** of a pre-specified length. Identify the auxiliary variables (memory) where the code sequence information is stored and the timers used for implementing the sequence validation procedure. Discuss how to proceed if the input fails, e.g. someone enters a wrong code.



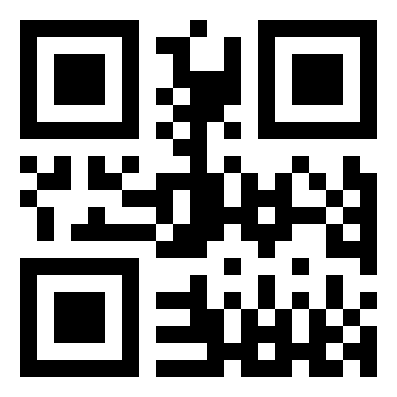
**C4.** Upload the program to the PLC, execute it and comment the results. Use logging-methodologies (see part A question 9) and **create** **plots that illustrate the functioning** of the system.



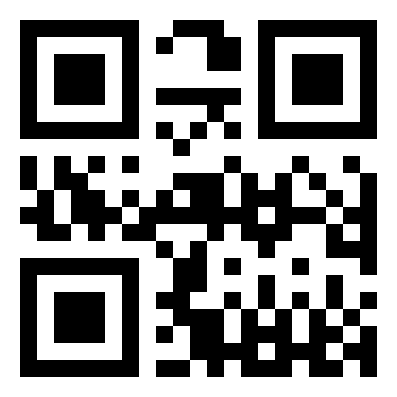
**C5.** Use the virtual terminal and its **string logging options** (see figure 1) to report the following experiment: (1) the alarm is activated using a four keys code, (2) the alarm is deactivated using the same keys code. Please show a **print screen of the window** opened when terminal option "Logged strings show" is selected. Write in the report a subroutine that copies the buffer of keys to the string buffer (starting at %MW190). The copy must convert key codes to ASCII characters so that the string is readable in the virtual terminal.



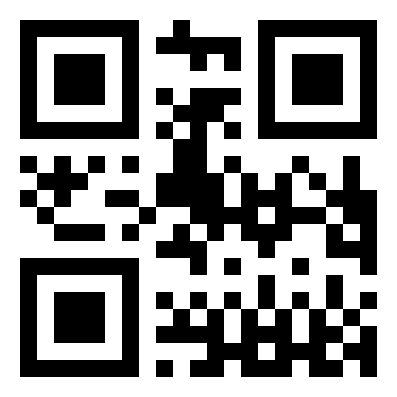
**C6.** (**Facultative question)** Use the keyboard to replace the switches for changing the mode of the system. More in detail, let #1\* set the off mode, #2\* set the presence detector and #3\* set the active mode. Show these keyboard commands are operational by using the string logging option (as used in C5). Make a graphical representation, e.g. syntax chart[[2]](#footnote-2), of sequences of keys leading to the various working modes. Do not represent eventual mistakes a person makes while pressing keys but describe, in a sentence or paragraph, what happens in the implemented PLC program.



**C7.** Indicate the **minimum time pressing keys** till successfully deactivating the active alarm mode. Indicate also the maximum time, if applicable.

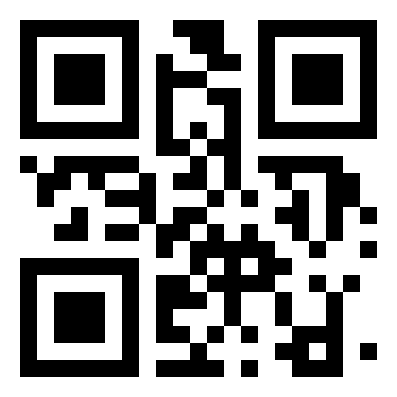


**C8.** Draw a hardware diagram that allows **testing automatically** your solution by using hardware as   
http://users.isr.ist.utl.pt/~jag/course\_utils/plc\_interf/plc\_flat\_cable\_interf.html



Comment whether running code similar to the one proposed in part B, question B7, can be of help for automating tests of your solution.

**C9.** Discuss possible future improvements to the alarm system. In particular try to identify aspects the **user of the system would like to see ameliorated**.



1. Original guide by Prof. Paulo J. Oliveira. 2021 revision by Prof. José Gaspar. [↑](#footnote-ref-1)
2. See section 1.7 of "Chapter 1, EBNF: A Notation toDescribe Syntax", by R. Pattis, https://www.ics.uci.edu/~pattis/misc/ebnf2.pdf [↑](#footnote-ref-2)