|  |  |  |  |
| --- | --- | --- | --- |
| IST_logo | ***Industrial Processes Automation***  *MSc in Electrical and Computer Engineering*  *Scientific Area of Systems, Decision, and Control*  *Winter Semester 2016/2017* |  | *Group: 2*  *­­*75167*-Francisco Oliveira*  75322*-Catarina Gaspar*  75637*-Inês Lourenço*  84993*-Hugo Serieiro* |

***1st Laboratory Assignment [[1]](#footnote-1)***

***Alarm System for Intrusion Detection***

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

# Part C - Integration

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. In particular, 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) has to be moved to an upper integration level.

Keyboard reading is also a function assumed to be running continuously. The upper integration level has the ability to validate codes saved in a buffer and/or reset keyboard input if a timeout occurs or any other condition recommends flushing keys already read.

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

**Q1.** *Represent as**a 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.*

O sistema descrito pode ser representado como na seguinte máquina de estados:

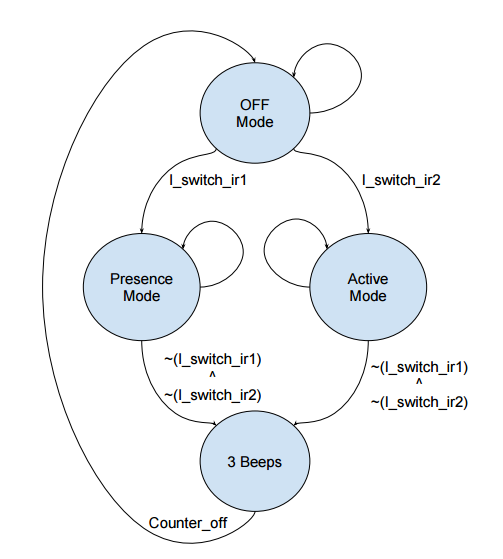
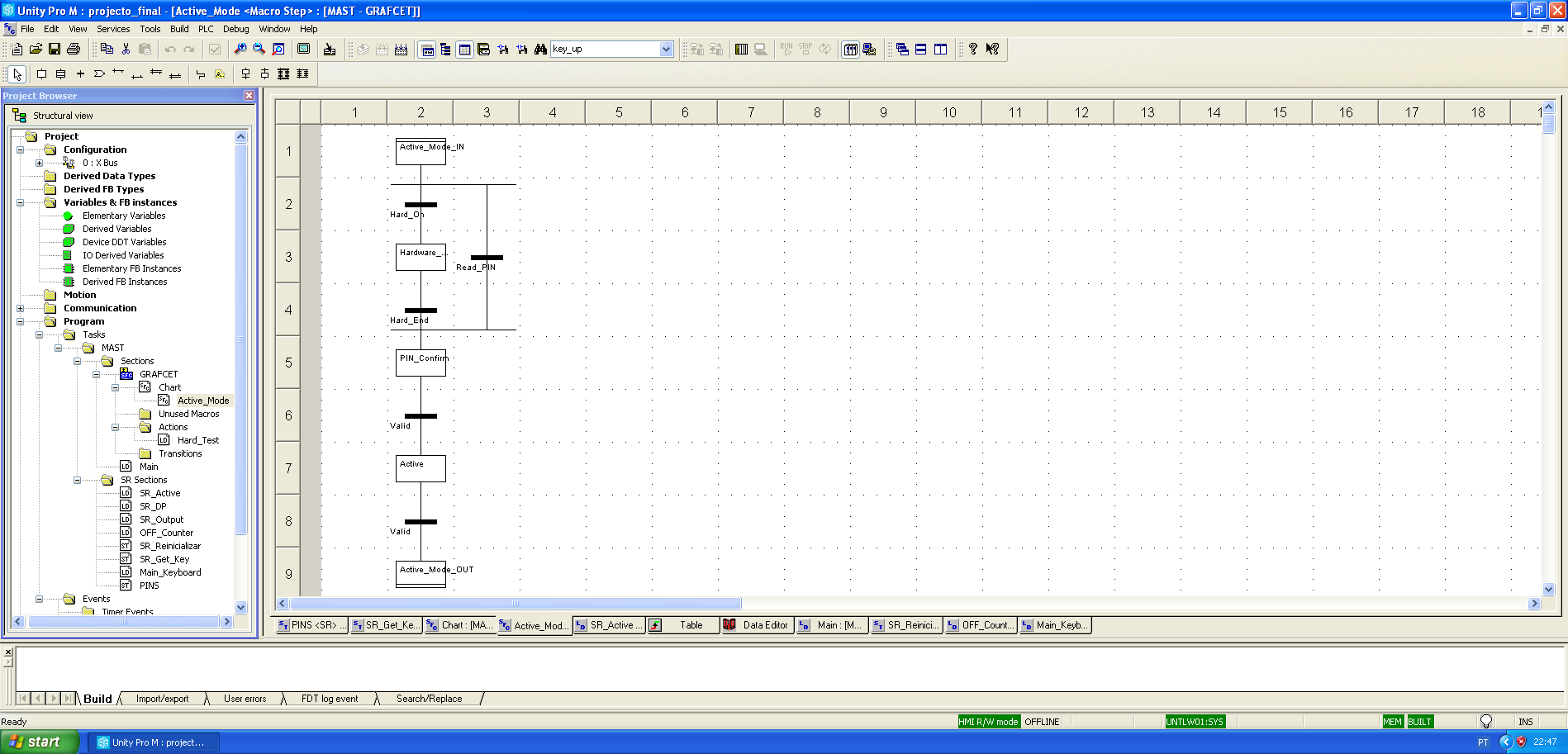


Figura 1 – Máquina de Estados

**Q2.** *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.*

O GRAFCET implementado para o nível mais alto de integração do sistema tem a estrutura da máquina de estados representada em Q1, com transições entre eles. De modo a simplificar este programa principal, no estado “Active Mode” foi utilizado um *macro step* com a sequência de acções (steps), necessárias dentro deste estado para cumprir os requisitos avançados especificados, da maneira representada na Figura 2:



No modo OFF, são accionadas as variáveis “Set\_CODE”, valid e Flush\_Flag. A variável “Set\_CODE” está activa apenas enquanto se está neste modo e esta possibilita a função de mudaça de código do alarme, a ‘valid’ é reiniciada para se ter que voltar a confirmar o código inserido e a ‘Flush\_Flag’ limpa o buffer para que tudo volte ao normal ao entrar no Active\_Mode outra vez.

No modo PRESENÇA (modo “DP”, detect presence) está activa a variável “Modo\_DP”, estando desactiva caso não se esteja neste modo. Esta variável vai, através da secção MAIN, chamar a subroutina SB\_DP. Nesta subroutina está implementado funcionamento do modo presença: aquando da activação do *switch* que representa o infravermelho (“I\_switch\_2”), são activas duas *flags*, uma durante 5seg que irá controlar o *led* (“led\_DP”), e outra durante 1seg que controlará o buzzer (“Buzzer\_DP”). Está também ativa a variável ‘Cardinal\_Fla’g, que permite desligar o alarme quando este estiver a tocar neste modo.

Ao passar do modo OFF para o macro step “Modo\_Active”, espera-se que uma tecla seja pressionada. Se for a tecla #, entra-se no ‘Hardware\_test’, que liga as 3 luzes e no fim o buzzer, sequencialmente, e posteriormente fica à espera de receber outra tecla, que vai já testar se pertence ao PIN correto ou não. Caso contrário, ignora o ‘Hardware\_Test’ e testa logo o PIN. A validade do pin é testada na sub-rotina PINS e apenas quando este for válido (‘valid’ = true), se entra no ‘SR\_Active’, que funciona tal como definido na Parte A do laboratório. Apenas o PIN ter então sido bem inserido, é possível sair deste modo. Quer seja ou não detetada a presença de uma pessoa.

Figura 2 – Active\_Mode

Sempre que se sai de um dos 2 principais estados (‘Presence\_Mode’ e ‘Active\_Mode’) é ativa a variável Flag\_3Beeps e entra-se no modo ‘Buzzer\_OFF’, que faz tocar três vezes o buzzer e de imediato passa para o ‘OFF\_Mode’.

**Q3.** *Describe the* ***methodology used to validate sequences of keys*** *of a pre-specified length. Discuss how to proceed if the input fails. In particular, identify the auxiliary variables (memory) where the code sequence information is stored and the timers used for implementing the sequence validation procedure.*

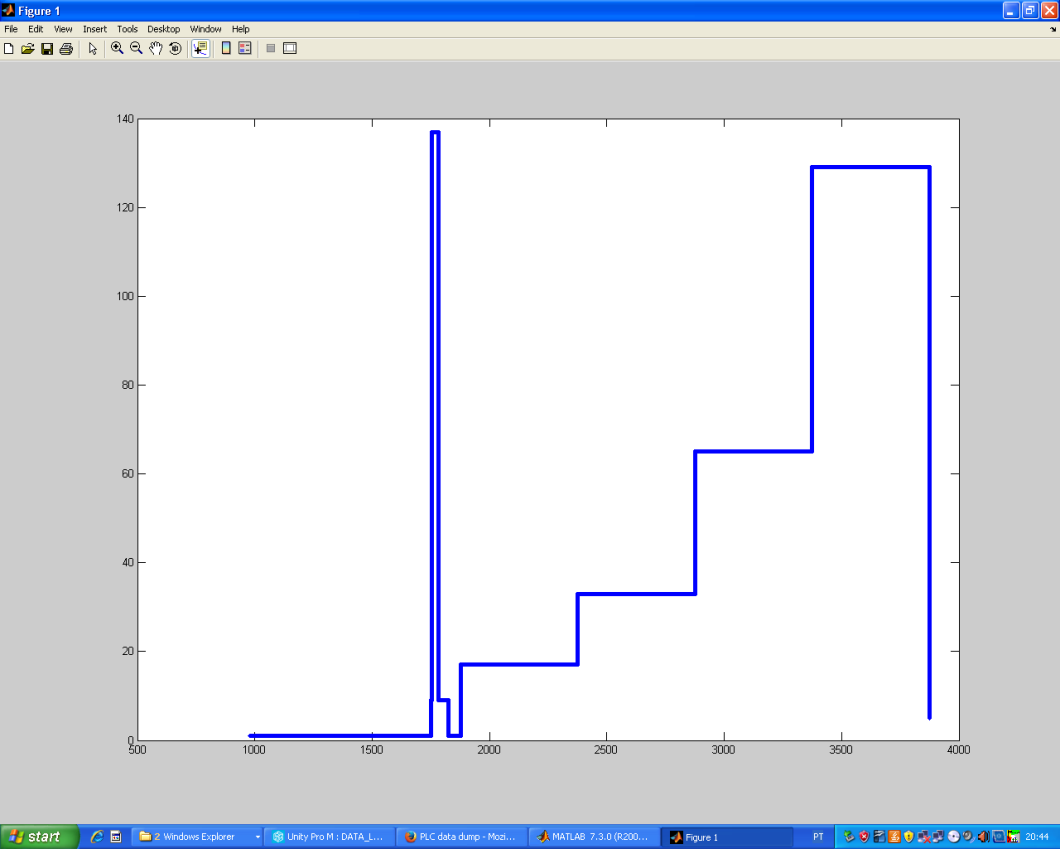
O programa ‘Main\_Keyboard’ está sempre a ser chamado, usando timers para controlar as colunas do teclado que são ligadas e desligadas e chama a sub-rotina ‘SR\_get\_key’, que foi implementada no laboratório anterior. A sub-rotina ‘PINS’ é a responsável por testar se o código inserido está de acordo com o último definido ou não, e reiniciá-lo quando desejado.

Assumiu-se que o código é uma sequência de 4 dígitos. Para a validação do código inserido, compara-se cada uma das primeiras 4 posições do buffer, onde são guardados os dígitos que foram inseridos (“PIN [0-9]”), com o vector de 4 posições que contém o pin correcto (“CODIGO [0-9]”). Estas comparações foram feitas com um ciclo *for*, onde em cada iteração do ciclo ela é feita. No caso de ser falsa, é feito *exit* do ciclo e nada acontece. Por outro lado, caso se chegue à quarta e última iteração e a comparação for verdadeira, então o código introduzido é o correto e portanto não se faz *exit* do ciclo, sendo então acionada uma *flag* (“valid”). No modo “OFF” o utilizador pode decidir definir um novo código, por questões de segurança. Para isso, deverá carregar no botão \*, seguido do código anteriormente definido, e logo depois uma qualquer nova sequência de 4 dígitos, seguidos de outro \* no fim. Isto faz com que as primeiras quatro posições do vetor CODIGO fiquem agora com a informação do novo código mudado, passando a comparação a ser feita com ele. Caso haja algum erro na leitura dos inputs do teclado, o utilizador deve carregar quatro vezes na tecla \*, o que faz com que a ‘Flush\_Flag’ seja ativa e a variável CODIGO volte a guardar o código inicialmente definido, 1234. O valor de 99 foi definido para a “limpeza do buffer” em vez do tradicional 0 para que fosse possivel o buffer reconhecer o valor 0 como um dígito novo válido para PIN.

**Q4.** *Upload the program to the PLC, execute it and comment the results. Use logging-methodologies (see part B) and* ***plots to illustrate the functioning*** *of the system.*

Para mostrar algumas funcionalidades do sistema de alarme implementado com o método de logging foram escolhidas duas situações dentro do modo activo que está descrito na figura 2. Os dois caminhos possíveis são ou carregar no cardinal e efectuar um teste ao hardware, ou premir numa outra tecla qualquer passando assim logo para a confirmação do PIN.

Primeiro, foi feito o Hardware test, que apresentou os seguintes resultados:

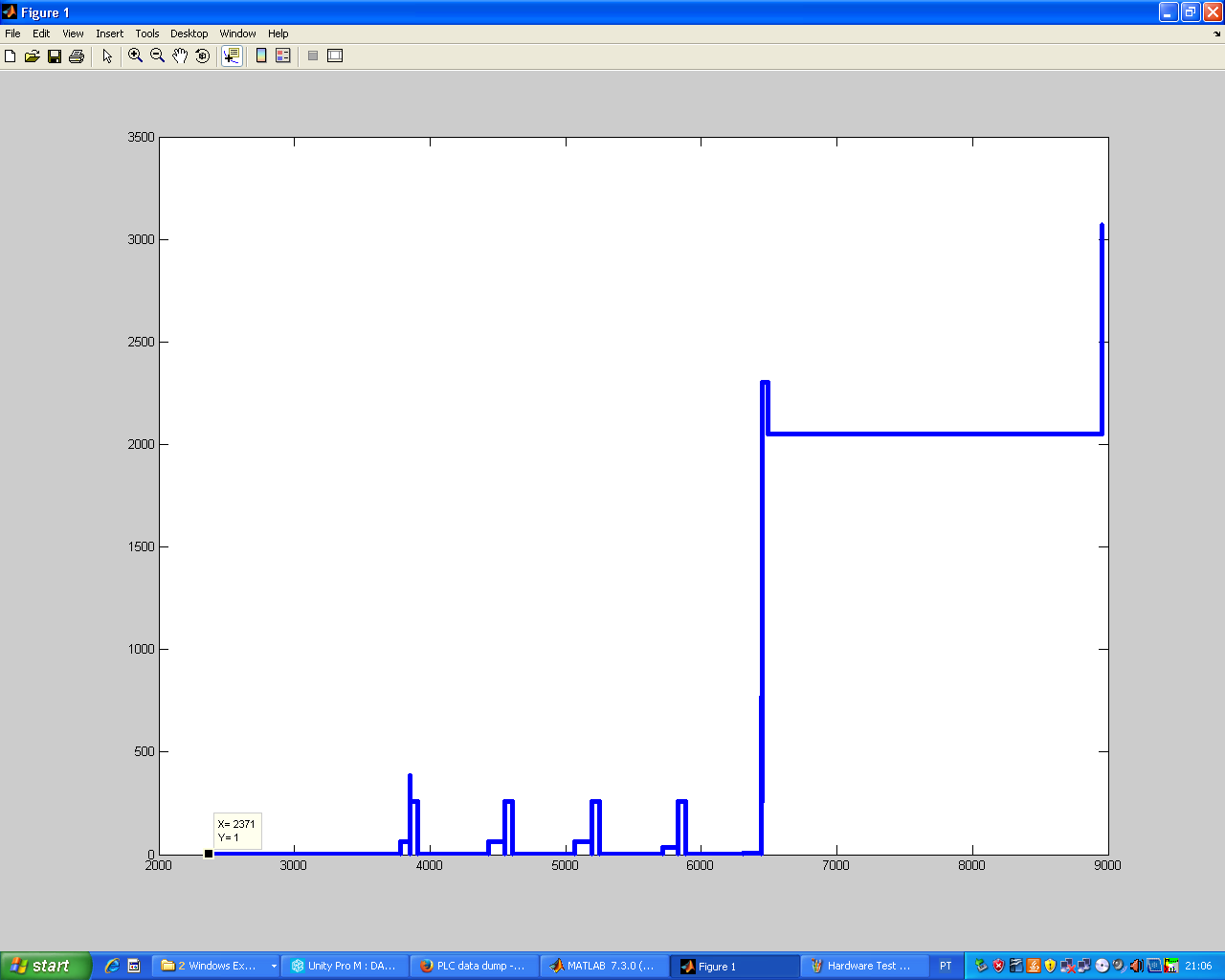


A palavra geradora de eventos teve a seguinte sequência (e composição):

*O\_Buzzer | O\_Alarme | O\_DP | O\_Activo | Linha\_4 | Hard\_End | Hard\_On | I\_Switch\_2*

Portanto, olhando para a figura e a sequência da palavra vê-se que primeiro o I\_Switch\_2 fica a 1 ou seja, entra no modo que dá início ao active mode. Depois disto, como é o cardinal que dá início ao teste de hardware, a Linha\_4 é activa seguido do buzzer que liga e desliga confirmando assim que o cardinal foi pressionado. Após largar o cardinal (5º passo) o flag Hard\_On é activo assinalando o início do teste de hardware, fazendo a transição para o modo Hardware Test, desligando-se imediatamente após isto. De seguida é feito o teste, ligando-se consecutivamente os Leds seguidos do buzzer. No ultimo passo é activa a flag que assinála o fim do teste (Hard\_End) dando por terminado esta funcionalidade.

A outra funcionalidade é a de entrar no modo activo através do PIN directamente sem ter que ser feito algum teste de Hardware. Este obteve os seguintes resultados:

****

A palavra geradora de eventos teve a seguinte sequência (e composição):

*Flag\_Active | O\_Activo | Valid | Key\_Up | Read\_PIN | Linha\_1 | Linha\_2 | Linha\_3 | Linha\_4 | Hard\_End | Hard\_On | I\_Switch\_2*

Nesta situação vê-se novamente que começa por activar o I\_Switch\_2 para entrar no modo que dá início ao active mode. Desta vez, no entanto, a Linha\_1 é reconhecida (por carregar no botão ‘1’) logo o programa percebe que o utilizador quer introduzir o PIN saltando o Hardware Test. Isto vê-se não só no facto do Hard\_On e o Hard\_End não activarem mas também quando no 4º passo liga a flag Read\_PIN. Aqui a flag Key\_Up também liga indicando que a tecla já foi largada. Os seguintes passos são a inserção do PIN pré definido ,1234, seguido do cardinal ‘#’.

- Passos 7 a 10 – Carregar no botão 2 – Linha\_1 + Key\_Up

- Passos 11 a 14 – Carregar no botão 3 – Linha\_1 + Key\_Up

- Passos 15 a 18 – Carregar no botão 4 – Linha\_2 + Key\_Up

- Passos 19 a 20 – Carregar no botão # – Linha\_4

Após largar o cardinal, para além de se ter o Key\_Up, o código já foi inserido correctamente, ligando assim a flag que verifica a validade do PIN inserido. Como está válido, é feita a transição para o modo activo, em que é desligada a flag da validade e é ligada a Flag\_Active. A última alteração ocorre após os 30 segundos de entrada no modo active (no caso deste teste foram utilizados 5 segundos de forma a poupar tempo), onde é ligada a led verde (O\_Activo) assinalando que o alarme irá disparar assim que alguém for detectado.

**Q5.** *Indicate the* ***minimum time pressing keys*** *till successfully deactivating the active alarm mode. Indicate also the maximum time, if applicable.*

Estando no modo activo, após ter sido detectada uma presença, é necessário introduzir um PIN com 4 caractéres seguido do cardinal ‘#’. Assumindo o PIN pré-definido ‘1234’ como exemplo, o tempo mínimo necessário para desactivar é o tempo que as colunas demoram a transitar para que todas as teclas, incluindo o cardinal sejam lidas. As colunas são ativas sequencialmente com timers TP com duração de 20ms, intervalando-se com um período de espera em que nenhuma coluna está activa durante outros 20ms. Desta forma, entre pressionar um botão na coluna 1 e 2 irião passar 40ms, e o mesmo sucede entre a coluna 2 e 3. Na sequência de botões necessária para desactivar o alarme “1234#” a sequencia de colunas necessárias é: 1 – 2 – 3 – 1 – 3 ou seja:

**Q6.** *Discuss* ***testing automatically*** *your solution by using hardware as*   
http://users.isr.ist.utl.pt/~jag/course\_utils/plc\_interf/plc\_flat\_cable\_interf.html

Antes de admitir que o equipamento está pronto para realizar as funções pretendidas, é fundamental realizar testes ao seu funcionamento, idealmente de uma maneira automática. Isto é possível ao inserir por exemplo um gerador de sinais e monitor entre os módulos do PLC e o terminal de hardware usado. Os sinais gerados adquirem uma grande relevância no teste do funcionamento do código do PLC e para efeito de debug. O monitor grava todas as mudanças nos sinais gerados, podendo mesmo gerar sinais que simulem uma específica sequência dos modos de funcionamento do PLC, o que consequentemente permite um debug de erros do código programado no PLC.

Segue-se a sequência possível de inputs automáticos que permitiriam verificar os vários funcionamentos deste sistema:

**DP**

Ligava-se o I\_Switch\_1 de forma a passar para modo DP ligando de seguida o I\_Switch\_IR para fazer soar o alarme e ligar a LED amarela (O\_DP). Após isto seria desligado ambos os switches de forma a verificar que o buzzer soa três vezes e que depois ambos os outputs desligam e voltam ao estado inicial.

**Mudança de PIN**

Já no modo OFF, poderia-se inserir o \* seguido do PIN, e depois PIN novo seguido de \* de forma a testar a alteração do PIN novo.

**Active**

Ligava-se o I\_Switch\_2 de forma a passar para o início do modo activo, seguido do cardinal para verificar o hardware test. Após isto seria inserido primeiro uma sequência de teclas que não o novo PIN definido no teste anterior para testar se não há validações erradas. Depois, seria inserida a sequência certa de teclas seguido do cardinal para verificar que a validação funciona correctamente, devendo-se ver o led verde (O\_Active). Após isto seria de novo ligado o I\_Switch\_IR para verificar o funcionamento do buzzer e led vermelho (O\_Alarme).

**Q7.** *Discuss possible future improvements to the alarm system. In particular try to identify aspects the* ***user of the system would like to see ameliorated****.*

O sistema de alarme tem como principal objetivo alertar alguém para uma situação fora do normal e muitas vezes indesejada, que aconteça num certo local. No caso do problema no nosso laboratório, e de modo a tornar o sistema um pouco mais real, uma melhoria seria usar um sensor de infravermelhos para detetar movimento em vez dessa deteção ser feita com um switch. Outra melhoria no sistema de alarme seria resolver a passagem do modo DP para o Active, sem ter que ir para estado OFF intermédio, o que não pode acontecer com a existência de um switch de 3 posições. Para o caso em que são inseridos alguns dígitos no keypad que não têm consequência porque o utilizador teve que interromper para fazer algo não completando assim com um ‘#’, seria interessante adicionar uma funcionalidade que limpasse o buffer passado um certo tempo para que teclas inseridas numas situação nao afectassem um futuro input do utilizador.

Para além das características avançadas implementadas no alarme, pensou-se noutras extra que poderiam melhorar o seu funcionamento, como por exemplo:

- Ligar o alarme a um circuito de imagem, em que uma câmara filmaria a o local assim que o alarme fosse ativado;

- Ter vários sensores de infravermelhos em diferentes compartimentos e consoante o que fosse ativado trancar as portas desse compartimento;

- Estar ligado ao telemóvel do utilizador e enviar uma mensagem de aviso tanto para ele como para os

# Annex - Functional Specifications of the Alarm

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. 9665). 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. 9665).

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. 9665).

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:

|  |  |
| --- | --- |
| 12 buttons keyboard  12V buzzer  1 three positions switch  1 two positions switch  3 LEDs |  |

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. 2010 original guide by Prof. Paulo J. Oliveira, 2016 rev. by Prof. José Gaspar [↑](#footnote-ref-1)