Industrial Processes Automation

MSc in Electrical and Computer Engineering
Scientific Area of Systems, Decision, and Control

Winter Semester 2018/2019

Group: C2
***70547** – João Ferreira*
***75268** – Rúben Tadeia*
***75987** – João Ribafeita*
***80978** – Gonçalo Pedro*

1st Laboratory Assignment [1]

Alarm System for Intrusion Detection

Part C - Integration

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.

Resposta: Para responder a esta questão fez-se o diagrama da *Figura 1*. Para esclarecer alguma ambiguidade: entenda-se por sensor, o switch (com o mesmo nome) que está na placa ficar ligado; a condição Switch ON/OFF refere-se ao Switch de três posições estar no modo respetivo (ON) ou não (OFF), por exemplo, para transitar para o estado Ativo o interruptor estar para o lado do DP é o mesmo que estar desligado.

Teste HW caracteriza o estado em que se realiza o teste ao hardware, acendendo sequencialmente os LED's e tocando o buzzer. Por fim, destaca-se a diferença entre os estados Activo e Alarme: no estado Activo o LED Activo (verde) estará ligado e o Alarme está armado; no estado Alarme o sensor já detectou um intruso e o alarme disparou, ativando o buzzer e o LED Alarme (vermelho).

[1] Original guide by Prof. Paulo J. Oliveira. 2019 revision by Prof. José Gaspar.

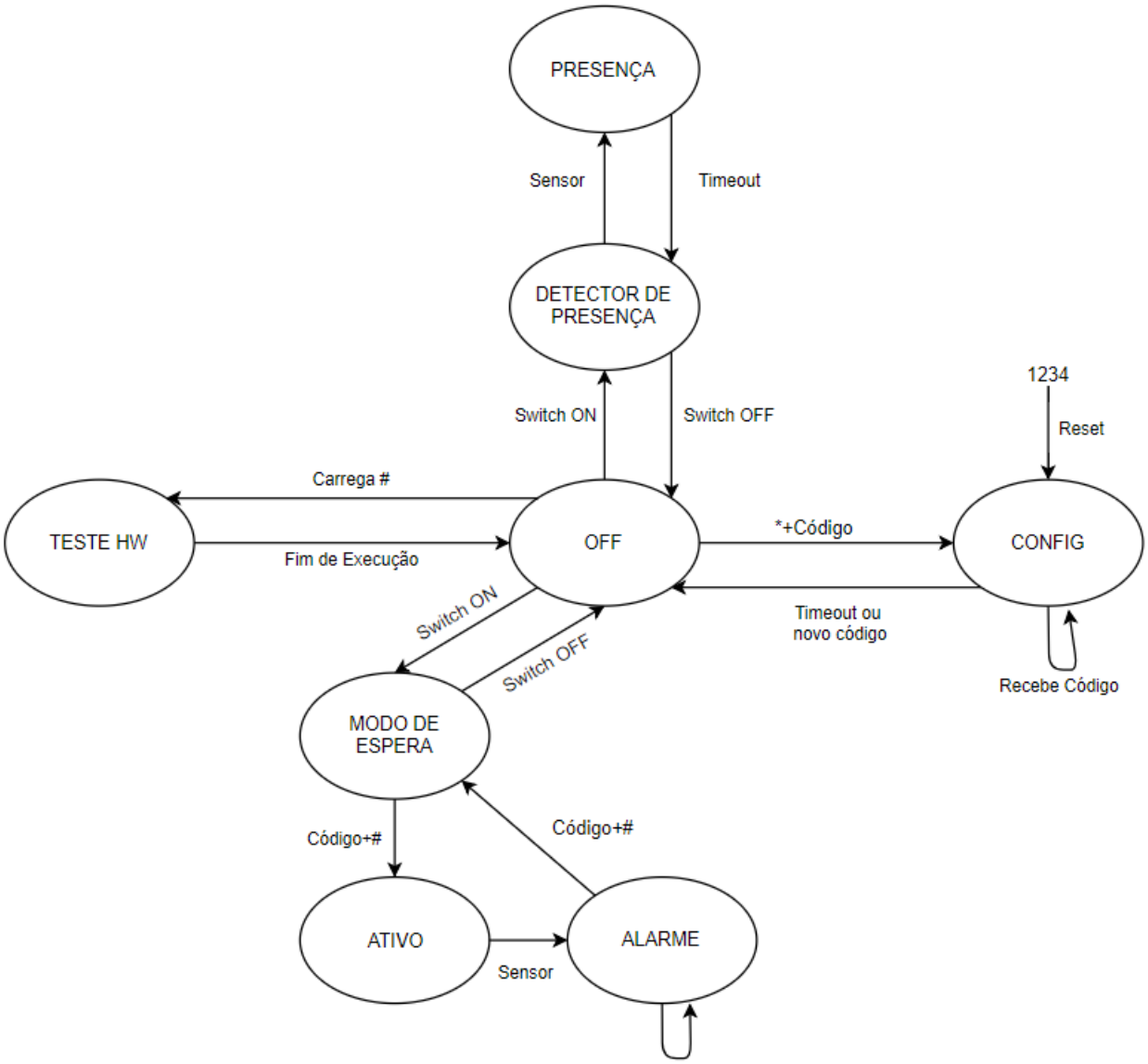


Figura 1 - Esquema da máquina de estados para o funcionamento global do alarme

Q2. Design a **GRAFCET / SFC** program representing the upper integration level. Please use macro steps to help to make simple the upper integration level. Describe the implemented receptivity sections. Please write comments within the code to improve readability.

Resposta: Para representar o nível superior de integração do alarme foi criado um programa em GRAFCET, tal pode ser observado na *Figura 2*. De modo a facilitar não só a visualização como também a implementação, foram utilizadas transições apenas com variáveis. Em termos de estados temos, como já foi observado na *Figura 1*, 5 estados principais, modo OFF, o teste de hardware, configuração de novo código, modo Ativo e modo de Deteção de presença. Temos ainda, 2 estados que correspondem à deteção de pessoas, o Alarme e o Presença. As variáveis consideradas no Grafcet que têm como prefixo "M_" correspondem a variáveis de estado, guardadas em memória, que permitem facilmente mudar de estado consoante o seu valor está a *True* ou a *False*.

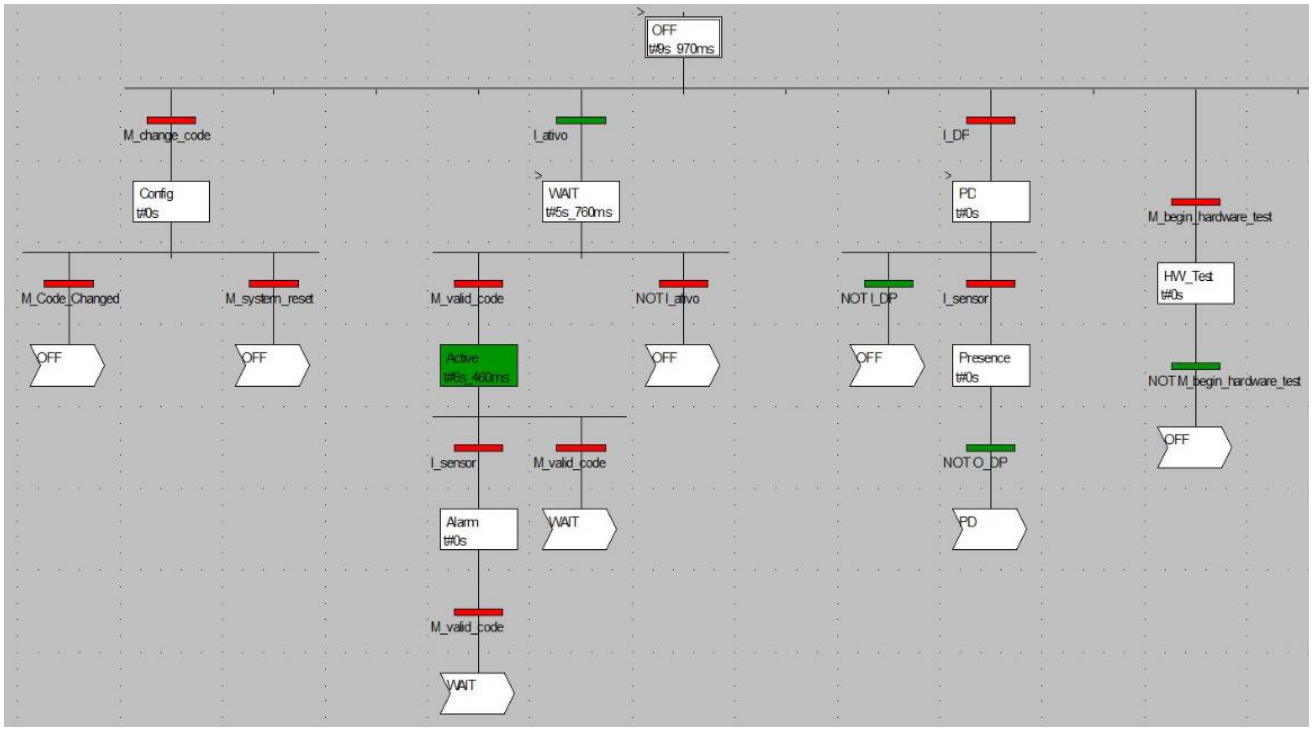


Figura 2 – Progama GRAFCET que contém o nível superior de integração Superior

Na Figura 2, conseguimos observar em Grafcet a localização do estado no modo activo. Uma das coisas que se poderia ter alterado, e que foi implementado em versões anteriores deste laboratório (partes A e B), seria a transição do modo Active para o modo Alarme, através de uma variável de memória que iria a True consoante o valor Rising Edge do I_sensor. Com esta funcionalidade, poderíamos ter um ladrão que entra em casa e fecha a porta (desliga o sensor), no entanto, não impede a ativação do alarme.

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.

Resposta: Nesta parte do laboratório, o teclado passou a ser verificado continuamente. Devido a isto, foram implementadas várias verificações, de modo a saber que acção deve ser desencadeada com cada pressionar de teclas (dado que as sequências a introduzir podem ter tamanhos fixos, mas diferentes). Todas as teclas detectadas são armazenadas no buffer.

Devido ao exposto anteriormente, diferentes procedimentos foram adoptados:

1. Quando o alarme está desligado, o utilizador pode carregar no # (desencadeando o teste ao hardware), ou carregar no *, indicando uma intenção de mudar o código.

2. No caso de o utilizador ter premido um *, o cursor é movido para a posição 0 e é activada uma flag, M_change_code, que indica uma intenção de mudar o código do sistema. Após esta sinalização, o sistema verifica as 4 teclas introduzidas pelo utilizador, para as comparar com o código do sistema e, caso sejam iguais, é activada a flag M_valid_code, que permite continuar a introduzir o novo código.

   Quando o novo código, seguido de *, foi introduzido, o sistema lê as últimas 4 posições do buffer antes do * e armazena o código novo na variável M_CODE, activando a flag M_code_changed.

   Se, durante a tentativa de mudança de código o utilizador se enganar no código, a mesma fica sem efeito.

   Além disto, caso o utilizador deseje efectuar um reset ao sistema, basta carregar 4 vezes no *, após ter sinalizado que deseja mudar o código.
3. No caso de o alarme se encontrar no modo activo ou no modo de espera, o teclado é pesquisado, às espera de que seja introduzido um código seguido de #; o # indica a posição corrente do buffer, que é armazenada e, no caso de a mesma ser menor do que 4, é efectuada uma rotação para a direita do buffer, mantendo o ponteiro para o #; após esta rotação, o código presente nas 4 posições anteriores é armazenado numa variável, que é então comparada com o código do sistema e, em caso de ser igual, é activada a flag M_valid_code. No caso de o código não estar correcto, nada acontece, continuando o alarme activado.
4. Caso o utilizador intruduza o código errado, ou seja detectado o código correcto, ou seja mudado o código, é efectuado o *flush* do buffer.

Q4. Upload the program to the PLC, execute it and comment on the results. Use logging-methodologies (see part A question 9) and **create plots that illustrate the functioning** of the system.

Resposta: Para realizarmos estes dataloggers realizamos alterações nos temporizadores para permitir o registo entre scan cycles. Foi também alterado o código base do datalogger relativamente não só ao Unity como também o código Matlab, de modo a permitir mais transições de estado, esse código segue em anexo no ficheiro zip.

Foram realizados 3 testes com o datalogger. O primeiro para o modo Activo e modo de Deteção de Presença (Figura 3), o segundo para o modo Activo sem activação do alarme por um intruso (Figura 4) e por último reset do código para o valor de fábrica (Figura 5).

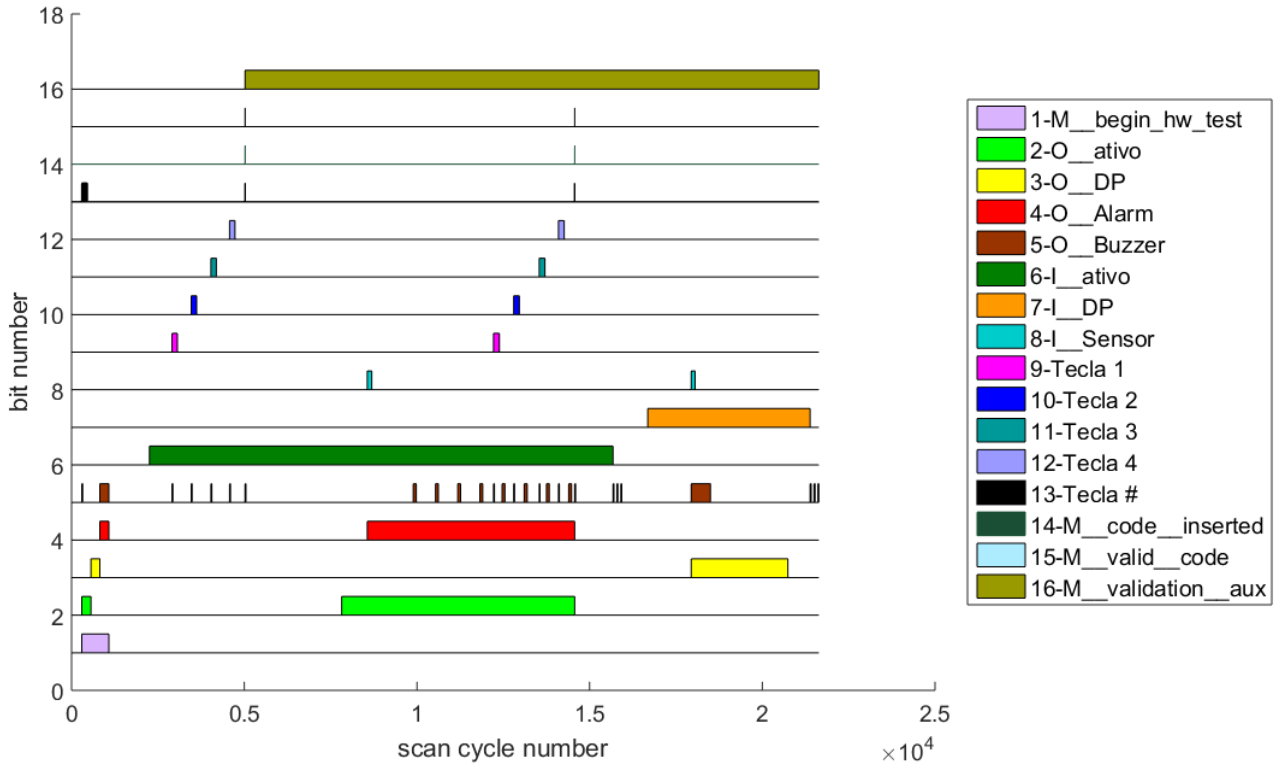


Figura 3 - Teste do modo Ativo e Detecção de Presença

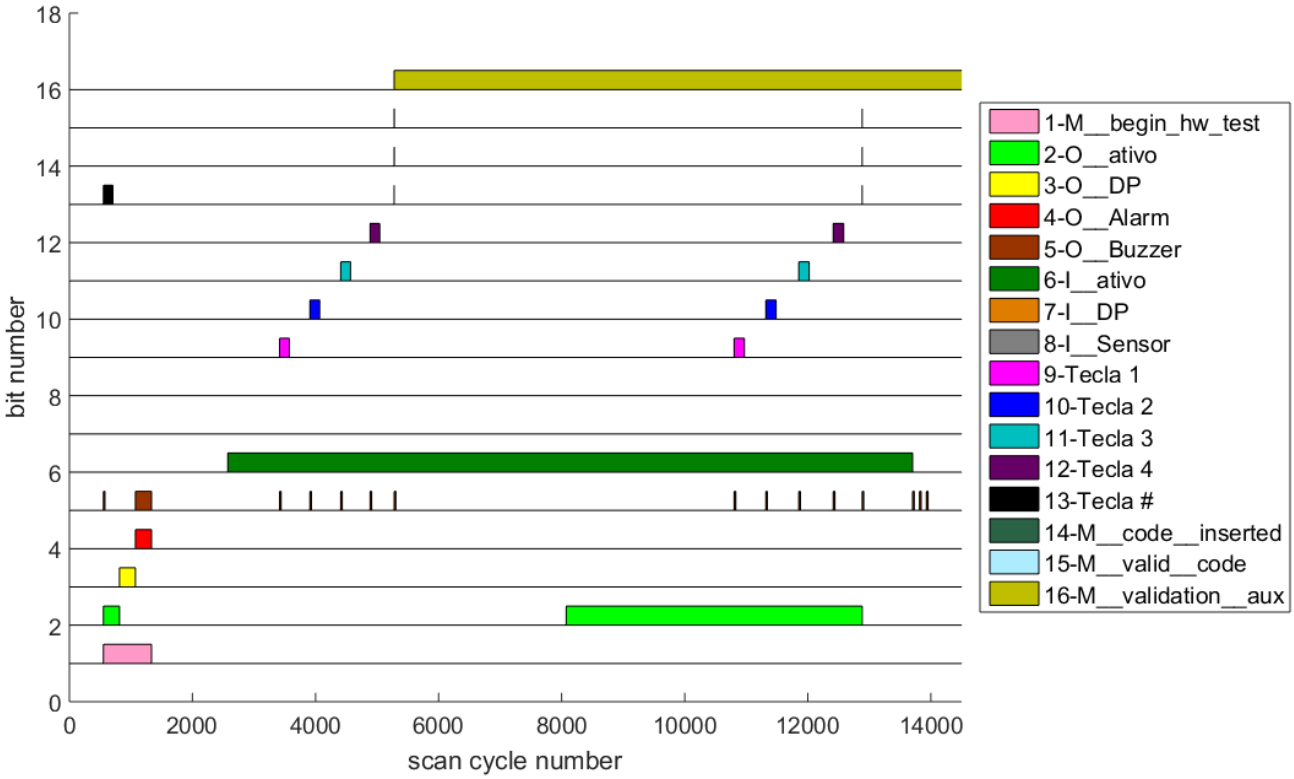


Figura 4 - Teste do modo Ativo sem ativação do alarme

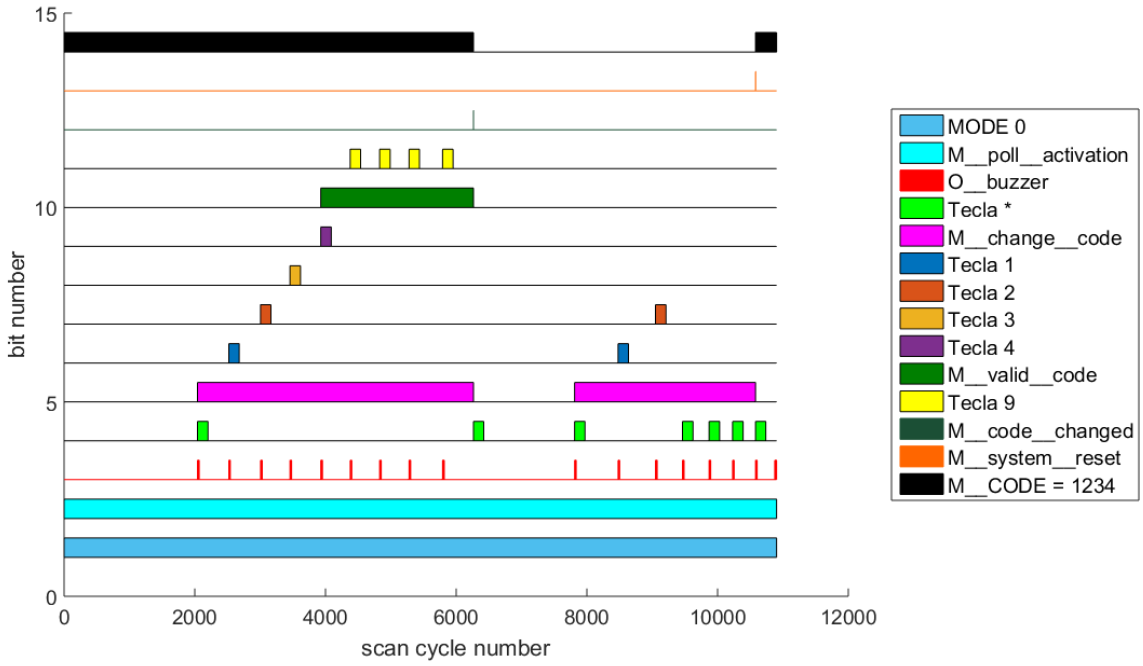


Figura 5 - Reset do código para o valor de fábrica

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

Resposta: Devido à implementação usada, o tempo mínimo corresponde a cerca de 4 intervalos entre teclas premidas (de 300ms cada um), acrescidos do tempo que demora a detectar cada uma das cinco teclas necessárias.

Dado que não foi implementado nenhum timeuot, não existe um tempo máximo para desactivar o alarme (o que pode levantar um potencial risco de segurança, devido a não proteger contra ataques *brute force*).

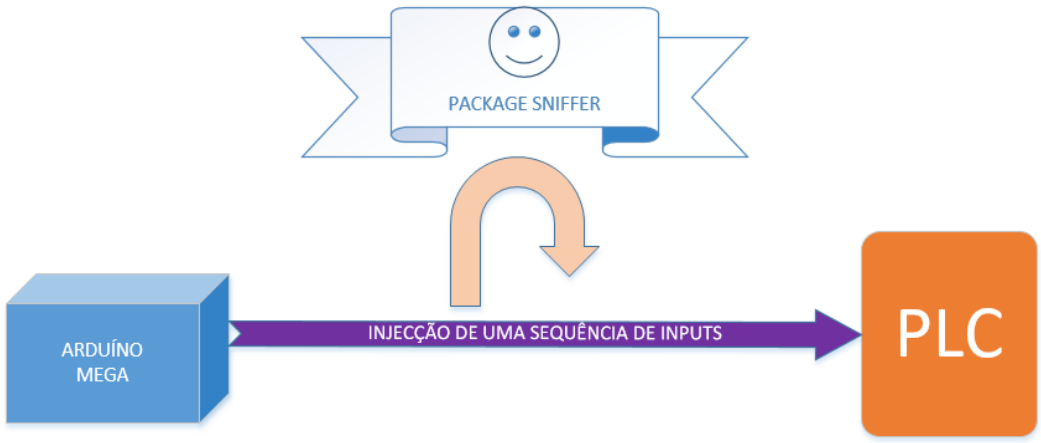
Q6. Draw a hardware diagram that allows **testing automatically** your solution by using hardware as:

http://users.isr.ist.utl.pt/~jag/course_utils/plc_interf/plc_flat_cable_interf.html

Resposta: Tendo por base a informação obtida no link acima, sabemos que podemos utilizar um gerador de impulsos, por exemplo um Arduíno Mega, para testar o nosso hardware automaticamente. O Arduíno, consegue gerar sinais que estimulam sinais que forçam o funcionamento da PLC a seguir uma sequência de comandos, ajudando no debug da aplicação. Durante a execução destas acções

poderíamos ter um package sniffer que analisasse os outputs, funcionando por exemplo como um datalogger que realizámos durante este laboratório (mais rudimentar).

Tal pode ser observado pela *Figura X*, existe um Arduíno Mega que estabelece uma ligação com uma PLC e realiza uma injecção de inputs e que será posteriormente analisado por um package sniffer.



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**.

Resposta: Dada por concluída a implementação do alarme para detecção de intrusos, entra-se agora numa fase mais criativa com ideias para novas implementações surgidas por brainstorm. Uma das ideias seria a implementação de uma interface gráfica, simplificando não só o teste das funcionalidades como também inseria o conceito de *user friendly* no alarme.

Se estivéssemos a lidar com alarme para ser implementado numa casa com um agregado familiar de por exemplo 3 elementos, podíamos ter um alarme por várias secções da residência. Secções como, a garagem, o quarto do bebé, a cave, etc. De notar que algumas destas secções podiam ser ativadas por pessoas diferentes e/ou em modos diferentes. Por exemplo, o alarme do quarto do bebé podia ser ativado pela mulher para detetar presença, ou seja, se o bebé acordou ou está alguém no quarto enquanto o alarme da garagem era ativado pelo homem para detetar intrusões, aumentando assim a necessidade de ter múltiplos utilizadores com códigos individuais. Outra funcionalidade interessante a ter, seria uma interface remota por IoT, por exemplo conseguir analisar o estado dos alarmes e/ou ativar os mesmos através de uma aplicação. Continuando a linha de pensamento por coisas na moda (trending), seria possível adicionar um conceito de domótica ou controlo automatizado, para

consoante as horas de entrada e saída na casa de uma pessoa da família, ativar/desativar o alarme. Em termos de acessibilidade seria útil ter uma referência em braille no teclado e uma referência sonora aquando o clique. Adicionalmente poderia existir um timeout na tentativa de preenchimento e leitura de um código, ou uma sequência de 3 tentativas para adição de um novo código. Uma confirmação visual e sonora após a mudança de código seria também útil para a confirmação do sucesso da operação.

Por fim, uma garantia de segurança seria avisar a polícia se o alarme não fosse desligado um determinado tempo após o disparo. Poderiam também ser ligadas das luzes do edifício, simulando alguém a verificar a ocorrência do alarme, estimulando a fuga do intruso ou simplesmente para obter melhores imagens no circuito de videovigilância.