



# 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

## 1<sup>st</sup> Laboratory Assignment <sup>1</sup>

### Alarm System for Intrusion Detection

#### Part A - System Components

##### Report questions

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



**Q1. (Hardware identification)** Identify the inputs (switches) and outputs (LEDs and buzzer) of the intrusion detection alarm console and fill the next table.

Input (chose the name to use)	PLC Identifier (physical address)
Sensor (Switch de 2 pos)	%i0.2.2
Alarme (Switch de 3 pos) - DP	%i0.2.0
Alarme (Switch de 3 pos) - Activo	%i0.2.1

Output (chose the name to use)	PLC Identifier (physical address)
LED de Alarme (Vermelho)	%q0.4.1
LED Detector de Presença (Amarelo)	%q0.4.2
LED Activo (Verde)	%q0.4.3
Buzzer	%q0.4.0



**Q2. (Hardware diagram)** Draw a hardware diagram of the alarm console components, namely switches, LEDs and buzzer, connected to the PLC. Do not forget to include the power supply.

**Resposta:** Para tentar responder a esta questão fez-se o diagrama da *Figura 1*. Como entradas temos os switches que se encontram na placa. Relativamente ao MODO este ou está ligado a DP

<sup>1</sup> Original guide by Prof. Paulo J. Oliveira. 2019 revision by Prof. José Gaspar.

(Detecção de Presenças) ou é Active (Ligado) ou então está OFF (Desligado). Nas saídas temos os 3 tipos de LEDS que correspondem aos modos de Active, DP e Alarme (que inclui Buzzer) e o Buzzer. Este sistema encontra-se alimentado por uma fonte de 24V.

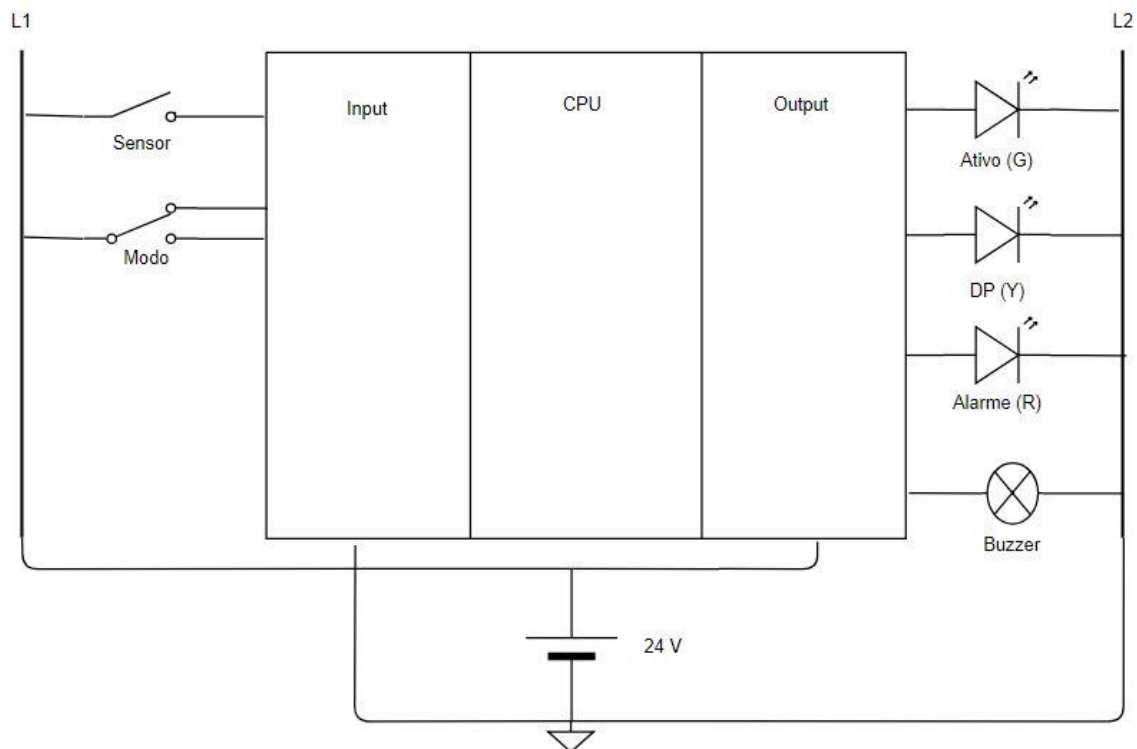

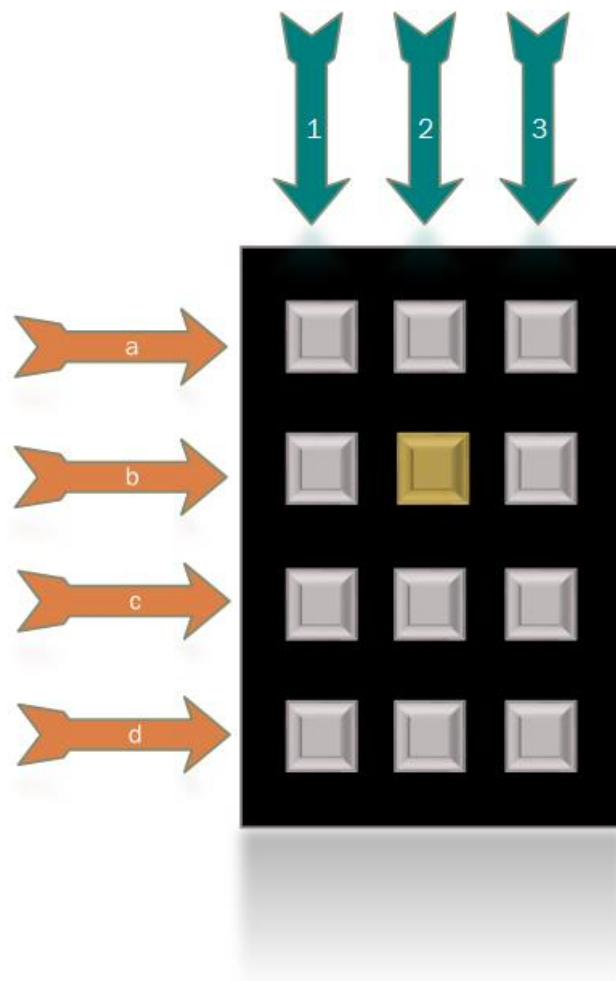


Figura 1 – Diagrama do hardware envolvido no alarme

 **Q3. (Keyboard interface)** Describe briefly how the PLC interfaces electrically to the keyboard in the alarm console (see some information in annex B). Draw a diagram to illustrate your description.

**Resposta:** Como é referido no anexo B, os teclados que estão disponíveis no laboratório têm 12 teclas disponíveis, orientados numa matriz 4x3. A maneira como estes funcionam é diferente de outros produtos fabricados e envolve alimentar uma coluna de cada vez, dado que existem menos colunas do que linhas. Uma vez ativada a coluna que se quer utilizar, no caso deste laboratório seria apenas a coluna 1, podemos analisar através da PLC a que input está associada cada tecla.

Se se quisesse resolver o problema de clicar numa tecla qualquer tendo 3 colunas ativadas, teríamos que ter por base o exemplo que se segue na *Figura 2*.



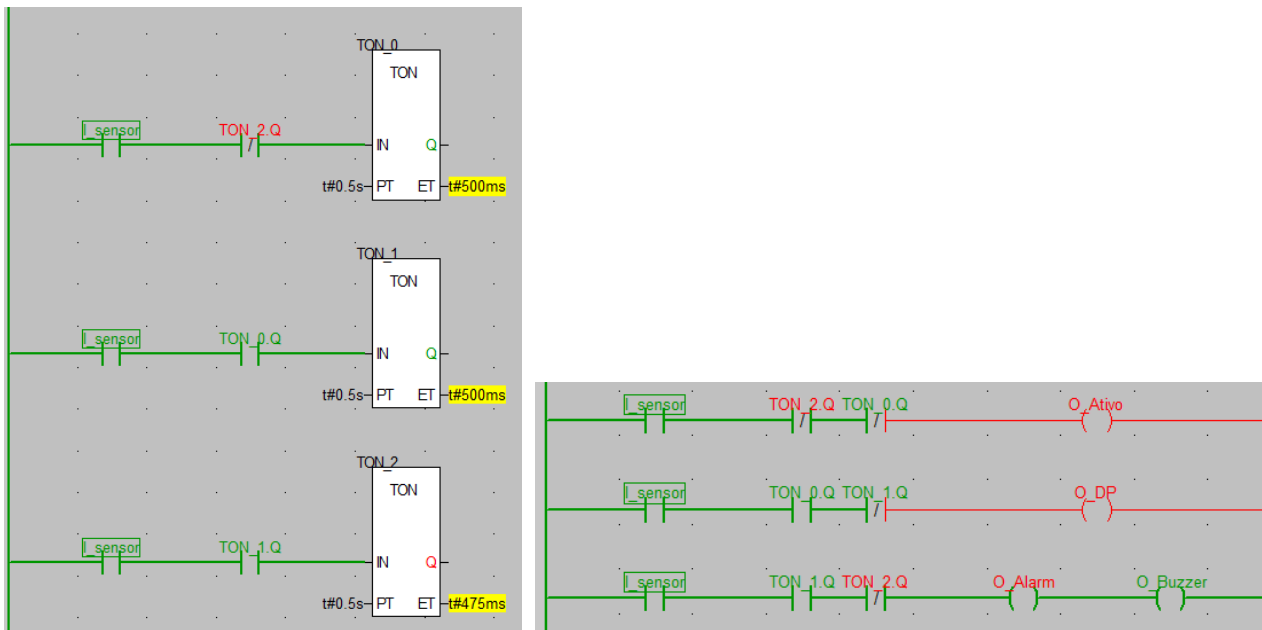
*Figura 2 - Esquema do teclado do laboratório*

Tendo por base a figura anterior, note-se que se utilizou o laranja para as linhas e azul para as colunas. Imagine-se que se pretende carregar na **tecla 5** do teclado, ao alimentar simultaneamente as três colunas, um curto circuito será detetado quando a tecla for pressionada. No entanto, a nossa PLC fica impossibilitada de identificar a tecla pressionada, uma vez que todas as colunas estão energizadas. O que se faz para contornar este problema é desligar as colunas até que o input desapareça, isto significa que o curto circuito era com a coluna que foi desligada. Por exemplo, desligando a coluna 3 o curto circuito mantém-se pelo que a tecla pressionada não foi a **6**. Desligamos a coluna seguinte (coluna 2) e a PLC deixa de reconhecer a tecla pressionada, logo sabemos que a tecla está a coluna 2, isto é, trata-se da **tecla 5**.



**Q4. (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 turn ON the three lamps (LEDs), one after the other, 0.5 seconds each. Create **one project with one MAST section** to test this sub-routine.

**Resposta:** É possível observar pela *Figura 3* como foi testado o programa. Na criação do programa para testar as variáveis de output foram utilizados 3 Timers On Delay (à esquerda) e lógica responsável por ligar os 3 LEDS sequencialmente e o Buzzer (à direita).



*Figura 3 – Diagrama de Ladder para teste dos outputs*



**Q5. (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 (BOOL), and can be unconditionally turned OFF by another memory variable (BOOL). Create a fourth way of actuating the buzzer, also commanded by a memory variable (BOOL), 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 push-buttons on the terminal.

**Resposta:** Através da *Figura 4* conseguimos observar a implementação da Sub-rotina SR\_Buzzer. Esta sub-rotina actua o buzzer de acordo com o modo desejado. Caso o modo desejado seja o modo cíclico, são utilizados 2 timers-on-delay, de modo a gerar uma onda quadrada (com *duty cycle* de 33%), responsável pela periodicidade do buzzer.

```

1  TON_5    (IN := NOT(%M21) (*BOOL*),
2           PT := t#2s(*TIME*),
3           Q => %M20 (*BOOL*));
4
5  TON_6    (IN := %M20 (*BOOL*),
6           PT := t#1s(*TIME*),
7           Q => %M21 (*BOOL*));
8
9  IF %M11 OR %M12
10     THEN %q0.4.0 := TRUE;
11 ELSIF %M14
12     THEN IF (%M20 AND Not %M21) THEN
13         %q0.4.0 := TRUE;
14     ELSE %q0.4.0 := FALSE;
15     END_IF;
16 (*buzzer off*)
17 ELSIF %M13
18 THEN
19     %q0.4.0 := FALSE;
20 ELSE
21     %q0.4.0 := FALSE;
22 END_IF;

```

Figura 4 – Código em Structured Text da sub-rotina SR\_Buzzer



**Q6. (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 [Segundos]	Short description of the usage of the timer
TOF_1	Off Delay	1	Modo DP: Duração do Buzzer
TOF_2	Off Delay	5	Modo DP: Duração do Led Amarelo
TON_5	On Delay	2	Modo Activo: Pausa do Buzzer
TON_6	On Delay	1	Modo Activo: Duração do Buzzer
TON_7	On Delay	30	Modo Activo: Tempo de espera para ligar
TON_8	On Delay	5	Modo Activo: Período antes de iniciar Buzzer após detectar ladrão

**Resposta:** Tendo em conta a aplicação do alarme (questão 8 deste relatório), foram utilizados apenas 4 timers como se consegue verificar pela *Figura 5*. Estes timers estão na base da implementação de todas as funcionalidades do alarme para os 3 Modos (Off, Detector de Presença e Activo).

```

1  TOF_1 (IN := I_sensor(*BOOL*), 10 (* Estes 2 timers servem para criar 23 (* Timer 7
2           PT := t#1s(*TIME*), 11 a onda quadrada a utilizar pelo 24 Used to wait 30 sec before start
3           Q => %M17(*BOOL*)); 12 buzzer *) 25 Detecting people *)
4
5  TOF_2 (IN := I_sensor(*BOOL*), 14 TON_5 (IN := NOT(%M22) (*BOOL*), 26 TON_7 (IN := I_ativo(*BOOL*),
6           PT := t#5s(*TIME*), 15 PT := t#2s(*TIME*), 27 PT := t#30s(*TIME*),
7           Q => %M18(*BOOL*)); 16 Q => %M23 (*BOOL*)); 28 Q => %M19(*BOOL*));
8
9                                     17 TON_6 (IN := %M23 (*BOOL*), 29
                                     18 PT := t#1s(*TIME*), 30 TON_8 (IN := %M19(*BOOL*),
                                     19 Q => %M22 (*BOOL*)); 31 PT := t#5s(*TIME*),
                                     20                                     32 Q => %M21(*BOOL*));
                                     33

```

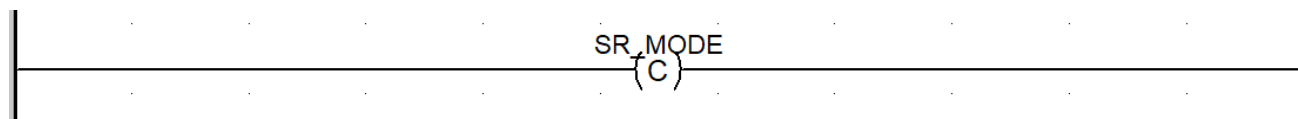
Figura 5 – Timers utilizados na implementação do alarme



**Q7. (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
SR_MODE	I_Ativo	Input	Switch do modo activo
	I_DP	Input	Switch que activa modo DP
	I_Sensor	Input	Switch que simula disparo do sensor
	I_alarm_shut	Input	Linha que corresponde ao #
	O_Ativo	Output	Led indicador de modo activo
	O_DP	Output	Led que indica modo DP
	O_Alarm	Output	Led que indica alarme
	O_Buzzer	Output	Buzzer
	O_KB_COL	Output	Coluna do teclado que corresponde a #

**Resposta:** Nesta questão apenas se considerou uma sub-rotina que realiza todas as funcionalidades relacionadas com o alarme, como se consegue ver pela *Figura 6*.



*Figura 6 – Chamada da Sub-rotina SR\_MODE*

Numa primeira abordagem, decidiu-se implementar as funcionalidades do alarme com apenas uma sub-rotina, *SR\_MODE*. Devido a esta implementação, as variáveis usadas são as indicadas acima.

Uma implementação mais correcta consistiria em compartimentar a actuação dos modos e do buzzer por sub-rotinas, o que iria acrescentar na tabela acima os bits de memória, usados como inputs e outputs.

Foram realizados vários testes, de modo a garantir que não existiam sobreposições dos valores de memória, contudo, considerando o contexto da *parte A* pedida para este laboratório decidiu-se manter esta abordagem na implementação. Relativamente às variáveis de memória que podem funcionar como input ou output, estas foram inicializadas e utilizadas apenas como auxiliar, não tendo sido atribuído um nome à variável. Nas partes seguintes, B e C, foram realizadas alterações a nível do código, que demonstra a evolução ao nível da aprendizagem.



**Q8.** (*System demonstration, with switches, without keyboard*) 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.

**Resposta:** Para se implementar estas funcionalidades criou-se apenas uma sub-rotina, a SR\_MODE em Structured Text. A variável M\_Mode foi declarada como um inteiro, que podia assumir os valores 0 para modo OFF, 1 para modo de Detecção de Presenças e 2 para modo Activo. Dado que só se utilizou uma sub-rotina, o código seria um pouco extenso (+ de 100 linhas) para se colocar uma imagem aqui, daí se recomendar a observação do código em caso de dúvida



**Q9.** (*PLC experiment time plot*) Use data logging to document an experiment, i.e. make a time plot of binary variables, where 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](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. In case your workbench has a 28FK input and output module, please replace in the PLC demo code the string `%i0.2` by `%i0.3` and the string `%q0.4.1` by the string `%q0.3.17`. In case the physical output address coincides with the buzzer, please change it to a LED.

**Resposta:** Nesta questão foi utilizado o código proporcionado pelo professor para demonstração do data logger. Foram realizados vários testes na tentativa de se obter não só os inputs e outputs desejados como também os resultados diferentes. Sabendo que o data logger apenas regista apenas as 10 primeiras mudanças de estado foi possível testar o modo activo. Dois resultados interessantes de observar são que o click do botão # no fim despoleta a passagem das variáveis para zero e consequente desactivação do alarme e que a tentativa de desligar o sensor após o alarme ter disparado não faz com que o mesmo seja desactivado. Estes resultados podem ser verificados na *Figura 7*.

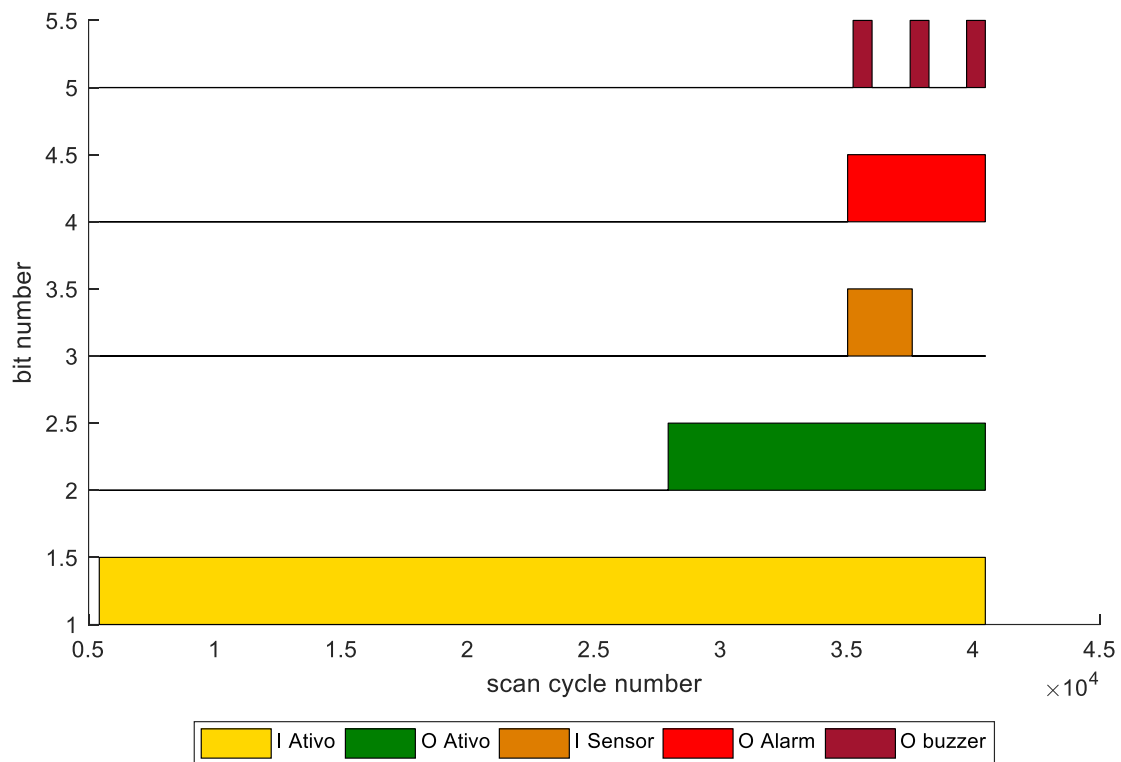


Figura 7 - Resultado obtido pelo data logger

O resultado experimental passou por activar o modo Active e esperar 30 segundos para que o utilizador possa abandonar o local. De seguida foi detectada uma presença em casa e após o ladrão ter entrado em casa pela janela, fechou-a na tentativa de calar o alarme. Mas já seria tarde, pois o sinal (Led Vermelho) de alarme já tinha sido activado e após 5 segundos começou a tocar o alarme (buzzer) em intermitente. E assim, num abrir e fechar de olhos chegou a polícia para prender o criminoso.



**Q10.** (*Comments on the system built*) Comment on how the complete program runs.

**Resposta:** Ao longo desta parte do primeiro laboratório foram desenvolvidas 3 sub-rotinas. Destas, duas foram realizadas em Structured Text, a *SR\_MODE* (implementa o alarme) e a *SR\_Buzzer* (que actua o buzzer) e uma em Ladder, a *SR\_self\_Test* (testa os leds e o buzzer em modo sequencial).

Algumas interpretações feitas resultaram numa implementação funcional, mas que pode facilmente não ser a melhor solução. Como já foi dito, é utilizada uma variável *M\_MODE* que foi declarada como um inteiro e varia entre 0 (OFF), 1 (Detector de Presenças) e 2 (Activo). Caso o Modo seja igual a 1 (DP) e o sensor detecte uma presença (o que é feito nesta implementação é ligar e desligar o sensor de presenças muito rapidamente, de modo a simular um impulso enviado por um sensor), o Led Amarelo é ligado durante 5 segundos e o Buzzer é activado durante 1 segundo; caso o sensor não



fosse desligado (simulando uma presença continuada no espaço), o alarme iria estar sempre a tocar e o Led Amarelo estaria ligado até que a pessoa saísse do local.

Quanto à implementação no modo Activo, esta é bastante directa e não gerou ambiguidades. Estes resultados podem ser comprovados através dos resultados do data logger.