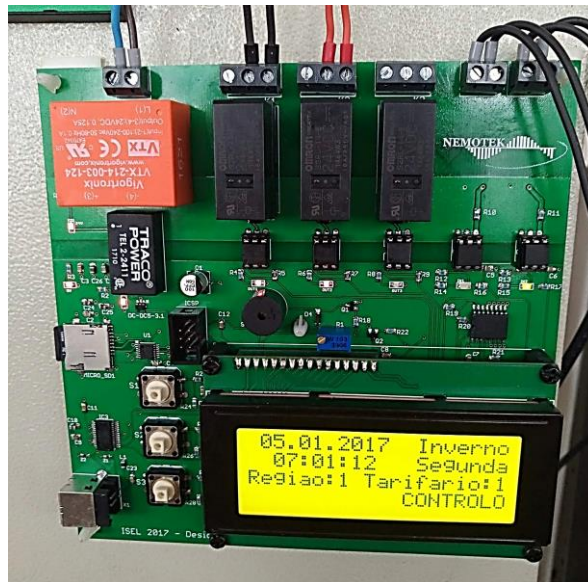


**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**Área Departamental de Engenharia Eletrotécnica Energia e Automação**



## **Desenvolvimento de Unidade Digital de Controlo Auxiliar para Baterias de Condensadores**

**PATRÍCIA CABAÇO NUNES**  
(Licenciada em Engenharia Eletrotécnica)

Trabalho Final de Mestrado para a Obtenção de Grau de Mestre em Engenharia Eletrotécnica  
– Ramo de Automação e Eletrónica Industrial

**Orientador(es):**

Professora Doutora Mafalda Maria Morais Seixas  
Professor Doutor Mário Rui Melício da Conceição

**Júri:**

Presidente: Professor Doutor Luís Manuel dos Santos Redondo

**Vogais:**

Professor Doutor Armando José Leitão Cordeiro  
Professor Doutor Mário Rui Melício da Conceição

**Dezembro 2017**



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**Área Departamental de Engenharia Eletrotécnica Energia e Automação**



**Desenvolvimento de Unidade Digital de Controlo Auxiliar  
para Baterias de Condensadores**

**PATRÍCIA CABAÇO NUNES**  
(Licenciada em Engenharia Eletrotécnica)

Trabalho Final de Mestrado para a Obtenção de Grau de Mestre em Engenharia Eletrotécnica  
– Ramo de Automação e Eletrónica Industrial

Orientador(es):

Professora Doutora Mafalda Maria Morais Seixas  
Professor Doutor Mário Rui Melício da Conceição

Júri:

Presidente: Professor Doutor Luís Manuel dos Santos Redondo

Vogais:

Professor Doutor Armando José Leitão Cordeiro  
Professor Doutor Mário Rui Melício da Conceição

**Dezembro 2017**

**Dissertação realizada sob orientação de:**

**Doutora Mafalda Maria Morais Seixas**  
Professora Adjunta da Área Departamental  
de Engenharia de Eletrotécnica de Energia e Automação

**Doutor Mário Rui Melício da Conceição**  
Professor Auxiliar com Agregação,  
Departamento de Física,  
Escola de Ciências e Tecnologia,  
Universidade de Évora

*If something is important enough, even if the odds are against you, you should still do it.”*

**Elon Reeve Musk** *(Businessman Who participates in companies such as SpaceX, PayPal, Tesla Motors).*



# Resumo

A dissertação é uma contribuição no âmbito do estudo, da implementação e do desenvolvimento de um protótipo funcional, destinado ao controlo auxiliar de uma bateria de condensadores. O protótipo é uma solução alternativa à atualmente utilizada pela empresa envolvida no projeto, a NEMOTEK, tendo como função principal implementar na bateria de condensadores os horários provenientes pela Entidade Reguladora dos Serviços Energéticos (ERSE). Não é função do equipamento o cálculo do consumo ou fator de potência.

A solução implementada recorre a duas entradas e três saídas, permitindo que as saídas comandem o ligar ou desligar total da bateria de condensadores, sendo esta formada por vários escalões, ou o ligar ou desligar de escalões individuais da bateria. O escalão pode ser de carácter capacitivo ou indutivo. Cada saída pode ser controlada exclusivamente pelo horário estabelecido, ou por uma das entradas, ou até por ambos, horário e entrada. As entradas podem ser configuradas para uma de duas funções, a de controlo ou de alarme, sendo possível escolher ambas as entradas com a mesma função, ou diferenciá-las. Para ser possível o controlo de uma saída a partir de uma entrada, esta entrada terá que ter a função de controlo, uma vez que a função de alarme está destinada apenas a desligar todas as saídas quando esta é acionada, emitindo um sinal sonoro, sobrepondo-se às eventuais entradas configuradas como controlo.

A solução é apresentada num protótipo funcional, que utiliza um microcontrolador do tipo Arduino Mega. A complexidade na construção do algoritmo a usar no protótipo, consiste no gerir automaticamente a data e hora; a de atrasar ou adiantar a hora, segundo a hora legal de verão e inverno; a de gerir os feriados, independentemente do país onde for implementado, ou qual o dia da semana, entre outras tarefas ligadas a esta regulação temporal.

Para a concretização do protótipo foi necessário planear e conceber a placa eletrónica, desde o *Printed Circuit Board* (PCB) até à soldagem dos componentes. Para isso foi utilizado o *Eagle CAD*. Além disso foram feitos testes laboratoriais para testar o comportamento de acordo com os requisitos físicos e de controlo estabelecidos neste projeto.

**Palavras-chave:** Microcontrolador, PCB, *Eagle*, Bateria de Condensadores, Horários da ERSE.



# Abstract

The purpose of this dissertation is a contribution to the study, implementation and development of a functional prototype, for the auxiliary control of a capacitor bank. This is an alternative solution, to the one that is currently used by the company involved in this project, the NEMOTEK. The main function of the prototype is to implement the schedules provide by the Regulator of Energy Services (ERSE). Although the calculation of consumption power or power factor is not a function of this equipment.

The implemented solution uses two inputs and three outputs, allowing the outputs to switch on or off the capacitor bank, being this formed by various steps, or to switch on or off only one step. These steps may have a capacitive or inductive character. In addition, the outputs can be controlled only by the schedule, or only by the input, or also by both, schedule and input. The inputs can be configured for one of two functions, control or alarm, and it is possible to choose both inputs with the same function or to differentiate them. To control an output by an input, the input must have the control function, because the alarm function is meant to turn off all the outputs, when the input with this function is activated, triggering a beep, overlapping the eventual inputs configured as control.

The solution is established for a functional prototype, which uses the Arduino Mega microcontroller. This automatically manages the date and time. Is capable of checking by itself, when there is the need to delay or advance the hour, according to the summer and winter time. This system can also manage national holidays, regardless the country, or any weekday, among other tasks related to time regulation.

For the implementation was necessary to design the all electronic board, from the Printed Circuit Board (PCB) to the soldering of the components. For this was used the Eagle CAD software. To verified the correct operations was performed tests in laboratory.

**Keywords:** Microcontroller, PCB, Eagle, Capacitor Bank, ERSE schedules.



## **Agradecimentos**

Gostaria de agradecer à minha família, especialmente aos meus pais, pelos valores que me inculcaram, pela educação e sobretudo pelas oportunidades que me proporcionaram, o que traduz a pessoa que sou hoje. Aos meus avós, pelos mimos e ao meu irmão pela paciência e motivação que me deu, nesta etapa da minha vida e todas as outras.

Quero agradecer aos meus orientadores por todo o apoio demonstrado ao longo desta dissertação, assim como por todo o material fornecido para a realização da mesma. Um especial agradecimento ao colega e amigo Paulo Almeida pela preciosa orientação, acompanhamento, dedicação e inestimáveis contributos, desde de conhecimentos técnicos, como científicos, como pela amizade demonstrada ao longo de todo este processo.

Os meus sinceros agradecimentos ao Instituto Superior de Engenharia de Lisboa, pelos conhecimentos instituídos e pela metodologia de trabalho que adquiri no período que frequentei esta instituição. Ao Departamento de Máquinas Elétricas pela disponibilização do espaço e equipamento para a realização da solução desenvolvida e testes experimentais.

Quero ainda expressar um sincero e grande agradecimento aos amigos, que me prestaram a sua amizade, ajuda e palavras amigas de incentivo em momentos menos bons durante o processo de desenvolvimento desta dissertação.

Por fim, mas não menos importante, um agradecimento final à empresa envolvida no projeto, pela disponibilidade, paciência e mais importante pela confiança que depositaram em mim, para a consecução deste protótipo funcional.



# Índice

|   |      |
|---|------|
| Resumo .....  | v    |
| Abstract.....   | vii  |
| Agradecimentos .....  | ix   |
| Índice .....  | xi   |
| Índice de Tabelas .....   | xv   |
| Índice de Figuras .....   | xvii |
| Acrónimos/Simbologia.....   | xxi  |
| Capítulo 1 Introdução.....  | 25   |
| 1.1. Introdução.....  | 27   |
| 1.2. Enquadramento e Motivação.....                                 | 28   |
| 1.3. Objeto de Estudo .....   | 29   |
| 1.4. Objetivos .....  | 29   |
| 1.5. Considerações Iniciais.....                                    | 30   |
| 1.6. Estrutura da dissertação.....                                  | 30   |
| 1.7. Notação.....   | 31   |
| Capítulo 2 Estado da Arte .....                                     | 33   |
| 2.1. Potência Ativa, Reativa e Aparente.....                        | 35   |
| 2.2. Regulamentação, ERSE .....                                     | 36   |
| 2.3. Equipamentos para a Compensação da Energia Reativa .....       | 37   |
| 2.3.1. Baterias de Condensadores .....                              | 37   |
| 2.3.2. Compensador Síncrono.....                                    | 39   |
| 2.3.3. Conversor eletrónico (Compensador Síncrono Estático).....    | 40   |
| 2.3.4. Conversores Eletrónicos (CA/CC-CC/CA) .....                  | 41   |
| 2.3.5. Conversor eletrónico (Filtro Ativo Paralelo Monofásico)..... | 43   |
| 2.3.6. Inversor Eletrónico (CC-CA) + STATCOM .....                  | 44   |
| 2.3.7. Resumo .....   | 44   |

|   |     |
|---|-----|
| 2.4. Microcontroladores.....                              | 46  |
| 2.4.1. Arduino .....                                      | 47  |
| 2.4.2. Controllino .....                                  | 48  |
| 2.4.3. Autómatos Programáveis (PLC) .....                 | 49  |
| 2.4.4. Comparação entre Arduino e Autómatos .....         | 52  |
| Capítulo 3 Trabalho Desenvolvido .....                    | 53  |
| 3.1. Introdução .....                                     | 55  |
| 3.2. Requisitos .....                                     | 60  |
| 3.3. Solução Desenvolvida, Hardware.....                  | 61  |
| 3.3.1. Arquitetura do equipamento a desenvolver.....      | 61  |
| 3.3.2. Implementação da Arquitetura .....                 | 63  |
| 3.3.3. Conceção do PCB.....                               | 68  |
| 3.4. Solução Desenvolvida, Software .....                 | 81  |
| 3.4.1. Interface entre o utilizador e o equipamento ..... | 82  |
| 3.4.1.1. Exemplo.....                                     | 84  |
| 3.4.2. Algoritmo .....                                    | 98  |
| 3.5. Protótipo Funcional .....                            | 115 |
| 3.6. Análise de custos .....                              | 118 |
| Capítulo 4 Resultados Experimentais .....                 | 119 |
| 4.1. Resultados Experimentais.....                        | 121 |
| 4.1.1. Código a funcionar .....                           | 126 |
| Capítulo 5 Conclusões e Desenvolvimentos Futuros .....    | 129 |
| 5.1. Conclusões.....                                      | 131 |
| 5.2. Desenvolvimentos Futuros .....                       | 132 |
| Bibliografia .....  | 135 |
| Apêndice I Dossier de Fabricação do PCB.....              | 141 |
| Apêndice II Conceitos para a Blindagem.....               | 169 |

|           |   |
|-----------|---|
| Anexo I   | Documento com os Requisitos da Empresa NEMOTEK              |
| Anexo II  | Catálogo do PLC Crouzet XD10                                |
| Anexo III | Catálogo do <i>Optocoupler</i>                              |
| Anexo IV  | Tabela Resumo da Corrente Máxima Permitida em cada Circuito |



## Índice de Tabelas

|  |     |
|--|-----|
| Tabela 2.1 - Resumo dos equipamentos anteriormente descritos, utilizados para compensar o FP.....              | 44  |
| Tabela 2.2 – Modelos do Controllino [26]. .....  | 49  |
| Tabela 2.3 - Resumo dos microcontroladores descritos, enumerando as vantagens e desvantagens. ....             | 52  |
| Tabela 2.4 – Diferenças entre os dois microcontroladores.....  | 52  |
| Tabela 3.1 – Período Horário em Portugal Continental, Ciclo Semanal, todos os fornecimentos [28].....          | 55  |
| Tabela 3.2 – Período Horário em Portugal Continental, Ciclo Semanal Opcional (MAT, AT e MT) [28]. .....        | 56  |
| Tabela 3.3 – Período Horário em Portugal Continental, Ciclo Diário (BTE e BTN) [28].....                       | 57  |
| Tabela 3.4 – Período Horário na Região Autónoma dos Açores, Ciclo Diário para todos os fornecimentos [28]..... | 57  |
| Tabela 3.5 – Período Horário na Região Autónoma dos Açores, Ciclo Diário Opcional (MT e BTE) [28].....         | 58  |
| Tabela 3.6 – Período Horário na Região Autónoma da Madeira, Ciclo Diário para todos os fornecimentos [28]..... | 58  |
| Tabela 3.7 – Período Horário na Região Autónoma da Madeira, Ciclo Diário Opcional (AT, MT e BTE) [28].....     | 59  |
| Tabela 3.8 – Dispositivos necessários para o desenvolvimento do hardware. ....                                 | 64  |
| Tabela 3.9 – Sequências possíveis para as entradas .....   | 95  |
| Tabela 3.10 – Bibliotecas utilizadas. ....   | 100 |
| Tabela 3.11 – Tipos de dados utilizados na declaração das variáveis. ....                                      | 102 |
| Tabela 3.12 – Identificação das rotinas utilizadas no setup. ....  | 106 |
| Tabela 3.13 – Distribuição do domingo na última semana do mês de outubro ou março.....                         | 107 |
| Tabela 3.14 – Cálculo para saber se já foi o último Domingo do mês de outubro ou março. ....                   | 107 |

Tabela 3.15 – Exemplo do cálculo para de terminar a hora do dia ..... 111

## Índice de Figuras

|   |    |
|---|----|
| Figura 2.1 - Triângulo de potências, ilustração gráfica da relação entre P, Q e S [1].                                    | 35 |
| Figura 2.2 - Exemplos de armários contendo BC [24].   | 37 |
| Figura 2.3 - Exemplo de uma rede com Compensação Individual [11].   | 38 |
| Figura 2.4 - Exemplo de uma rede com Compensação Parcial por Grupo de Cargas [11].  | 38 |
| Figura 2.5 - Exemplo de uma rede com Compensação Centralizada [11].   | 39 |
| Figura 2.6 - Condensador Síncrono, modelo Series SC [14].   | 40 |
| Figura 2.7 - Conexão do STATCOM a um sistema elétrico [19].   | 41 |
| Figura 2.8 - Conversor trifásico CA/CC-CC/CA [21].  | 42 |
| Figura 2.9 - Diagrama de blocos do sistema de alimentação de uma carga através de um inversor com transformador [22].     | 42 |
| Figura 2.10 - Diagrama de blocos do sistema de alimentação de uma carga através de um inversor sem transformador [22].    | 43 |
| Figura 2.11 - Diagrama de blocos da interface entre a eólica e a rede elétrica, utilizando um filtro ativo paralelo [22]. | 43 |
| Figura 2.12 - Placas de constituição simples, nível principiante.   | 48 |
| Figura 2.13 - Placa de constituição mais complexa, nível avançado.  | 48 |
| Figura 2.14 - Características de um Controllino MAXI [26].  | 49 |
| Figura 2.15 - Arquitetura de um autômato programável [27].  | 50 |
| Figura 2.16 - Autômato programável Crouzet XD10.  | 51 |
| Figura 3.1 – Diagrama da arquitetura dos componentes para a placa eletrônica.   | 62 |
| Figura 3.2 - Placa Arduino UNO [4].   | 64 |
| Figura 3.3 - Placa Arduino Mega 2560 [4].   | 64 |
| Figura 3.4 – Esquema de uma saída, no software Eagle [35].  | 66 |
| Figura 3.5 – Protótipo do sistema para desenvolvimento do hardware.   | 67 |
| Figura 3.6 – a) Catálogo do conversor DC-DC, b) Esquema do conversor DC-DC no Eagle [35].                                 | 69 |

|   |    |
|---|----|
| Figura 3.7 – a) Catálogo do microcontrolador, b) Esquema do microcontrolador no Eagle [35].   | 71 |
| Figura 3.8 – Os condensadores de bypass no esquema do microcontrolador no Eagle [35].   | 72 |
| Figura 3.9 – Localização da resistência de pull-up no esquema do microcontrolador no Eagle [35].                                    | 73 |
| Figura 3.10 – Localização do ICSP no esquema do microcontrolador no Eagle [35].   | 73 |
| Figura 3.11 – a) Catálogo do RTC, b) Esquema do RTC no Eagle [35].  | 74 |
| Figura 3.12 – Esquema do botão S1, no software Eagle [35].  | 75 |
| Figura 3.13 – Localização dos diferentes níveis de tensão no design do PCB [35].  | 77 |
| Figura 3.14 – Localização do microcontrolador e do RTC com a bateria no design do PCB [35].   | 77 |
| Figura 3.15 – Ilustração da ligação entre os pinos, onde vermelho representa a layer de cima e a azul a layer de baixo do PCB [35]. | 79 |
| Figura 3.16 – Folha do Excel para a inserção dos Feridos.   | 82 |
| Figura 3.17 – Botão que gera os ficheiros txt.  | 83 |
| Figura 3.18 – Folha do Excel que ilustra os horários da ERSE.   | 84 |
| Figura 3.19 – Diagrama de transição de estados, onde n representa o número de saídas existentes.                                    | 86 |
| Figura 3.20 – Folha do Excel que representa a configuração, Seleção das opções Região e Tarifário.                                  | 87 |
| Figura 3.21 – Folha do Excel que representa a configuração, Seleção das opções entrada 1, entrada 2 e tempo de atraso da saída 1.   | 87 |
| Figura 3.22 – Folha do Excel que representa a configuração, Seleção das opções para a saída 1.                                      | 88 |
| Figura 3.23 – Folha do Excel que representa a configuração, Seleção das opções finais para a saída 1.                               | 89 |
| Figura 3.24 – Rotina para a implementação das decisões do utilizador, leitura do documento .txt.                                    | 90 |
| Figura 3.25 – Rotina para a implementação das decisões do utilizador, seleção da Região.  | 91 |

|  |     |
|--|-----|
| Figura 3.26 – Rotina para a implementação das decisões do utilizador, seleção do Tarifário.                    | 92  |
| Figura 3.27 – Rotina para a implementação das decisões do utilizador, seleção da Entrada 1 e 2.                | 93  |
| Figura 3.28 – Rotina para a implementação das decisões do utilizador, seleção do tempo de atraso.              | 94  |
| Figura 3.29 – Rotina para a implementação das decisões do utilizador, seleção da função para a saída1.         | 97  |
| Figura 3.30 – Rotina para a implementação das decisões do utilizador, seleção da entrada para ativar a saída1. | 98  |
| Figura 3.31 – Fração do código respetivo às bibliotecas utilizadas.  | 99  |
| Figura 3.32 – Fração do código respetivo à declaração das variáveis.   | 102 |
| Figura 3.33 – Fração do código respetivo ao setup.   | 104 |
| Figura 3.34 – Os pinos dos interrupts associados às placas de Arduino (traduzido para português) [4].          | 104 |
| Figura 3.35 – Fração do código respetivo da rotina “VerificarVerao ()”.  | 109 |
| Figura 3.36 – Fração do código respetivo da rotina “VerificarHora ()”.   | 110 |
| Figura 3.37 – Fração do código respetivo da rotina “CiclosTarifario ()”.                                       | 112 |
| Figura 3.38 – Fração do código respetivo ao loop.  | 113 |
| Figura 3.39 – Fração do código respetivo da rotina “Manutencao ()”.  | 114 |
| Figura 3.40 – Fração do código respetivo da rotina “Botao1e2Reset ()”.   | 114 |
| Figura 3.41 – Aspeto da layer de cima do PCB pelo simulador 3D [42].   | 115 |
| Figura 3.42 – Aspeto da layer de baixo do PCB pelo simulador 3D [42].  | 115 |
| Figura 3.43 – Aspeto da layer de cima do PCB real.   | 116 |
| Figura 3.44 – Aspeto da layer de baixo do PCB real.  | 116 |
| Figura 3.45 – Aspeto do PCB com os componentes inseridos, sem o LCD colocado (placa eletrónica).               | 117 |
| Figura 3.46 – Aspeto do PCB com os componentes inseridos, com o LCD colocado (placa eletrónica).               | 117 |
| Figura 4.1 – Interior da BC.   | 121 |

|   |     |
|---|-----|
| Figura 4.2 – Colocação do equipamento na BC.....  | 122 |
| Figura 4.3 – Contactos do Relé [43]. .....  | 123 |
| Figura 4.4 – Circuito do optocoupler de uma das saídas [35].....                            | 124 |
| Figura 4.5 – Rotina para a implementação das decisões do utilizador, seleção da Região. ... | 126 |
| Figura 4.6 – Visualização da seleção da Região no LCD. ....                                 | 126 |
| Figura 4.7 – Visualização da ativação da entrada 2 (controlo). ....                         | 127 |
| Figura 4.8 – Visualização da ativação da entrada 1 (alarme). ....                           | 128 |

## Acrónimos/Simbologia

|           |   |
|-----------|---|
| ADC       | <i>Analog -to-Digital Converter</i> (Conversor Analógico/Digital)     |
| AT        | Alta Tensão   |
| BC        | Bateria de Condensadores  |
| BOM       | <i>Build of Material</i> (Lista de Material)                          |
| BT        | Baixa Tensão  |
| BTE       | Baixa Tensão Especial – potência contratada > a 41,4 kVA              |
| BTN       | Baixa Tensão Normal – potência contratada ≤ a 41,4 kVA                |
| CA        | Corrente Alternada, em A (Ampere)                                     |
| CC        | Corrente Contínua, em A (Ampere)                                      |
| CPU       | <i>Central Processing Unit</i>  |
| CS        | <i>Chip Select</i>  |
| DAC       | <i>Digital-to-Analog Converter</i> (Conversor Digital/Analógico)      |
| DRC       | <i>Design Rules Check</i> (Verificação das regras de desenho técnico) |
| DTE       | Diagrama de Transição de Estados                                      |
| D-STATCOM | <i>Distribution Static Synchronous Compensator</i>                    |
| EDP       | Energias de Portugal  |
| EMI       | <i>Electromagnetic Interference</i> (Interferências Eletromagnéticas) |
| ERSE      | Entidade Reguladora dos Serviços Energéticos                          |
| ESR       | <i>Equivalent Series Resistance</i>                                   |
| FACTS     | <i>Flexible AC Transmission Systems</i>                               |
| FC        | Frequência de Corte, em Hz (Hertz)                                    |
| FP        | Fator de Potência   |
| FTDI      | <i>Future Technology Devices Internacional</i>                        |
| GBL       | <i>Gerber Bottom Copper</i>   |
| GBO       | <i>Gerber Bottom Silk-Screen</i>                                      |
| GBS       | <i>Gerber Bottom Solder Mask</i>                                      |
| GTL       | <i>Gerber Top Copper</i>  |
| GTO       | <i>Gerber Top Silk-Screen</i>   |
| GTS       | <i>Gerber Top Solder Mask</i>   |
| HFV       | Horas Fora de Vazio   |
| HV        | Horas de Vazio  |

|                  |   |
|------------------|---|
| ICSP             | <i>In-Circuit Serial Programming</i>                                  |
| I <sup>2</sup> C | <i>Inter-Integrated Circuit</i>                                       |
| IEEE             | <i>Institute of Eletrical and Electronics Engineers</i>               |
| I/O              | <i>Input/Output</i> (Entradas e Saídas)                               |
| ISEL             | Instituto Superior de Engenharia de Lisboa                            |
| IVREA            | <i>Interaction Design Institute</i>                                   |
| LCD              | <i>Liquid Crystal Display</i>   |
| LED              | <i>Light Emitting Diode</i>   |
| MAT              | Muito Alta Tensão   |
| mil              | Milésimas de Polegada   |
| MISO             | <i>Master IN Slave OUT</i>  |
| MOSI             | <i>Master OUT Slave IN</i>  |
| MT               | Média Tensão  |
| MTBF             | <i>Mean Tome Between Failures</i>                                     |
| NA               | Normalmente Aberto  |
| NF               | Normalmente Fechado   |
| P                | Potência Ativa em kW ou MW (QuiloWatt ou MegaWatt)                    |
| PCB              | <i>Printed Circuit Board</i> (Placa de Circuito Impresso)             |
| PLC              | <i>Programmable Logic Controllers</i>                                 |
| PWM              | <i>Pulse-Width Modulation</i>   |
| Q                | Potência Reativa em kVAr (QuiloVolt-Ampere Reativo)                   |
| RAM              | <i>Random Access Memory</i>   |
| ROM              | <i>Read Only Memory</i>   |
| RRD              | Regulamento da Rede de Distribuição                                   |
| RTC              | <i>Real Time Clock</i>  |
| RV               | Relé Varimétrico  |
| Rx               | <i>Receiver</i> (Receção)   |
| S                | Potência Aparente em kVA (QuiloVolt-Ampere)                           |
| SCL              | <i>Serial Clock</i>   |
| SCK              | <i>Serial Clock</i>   |
| SDA              | <i>Serial Data</i>  |
| SPI              | <i>Serial Peripheral Interface</i>                                    |
| STATCOM          | <i>Static Synchronous Compensator</i> (Compensador Síncrono Estático) |

|                      |  |
|----------------------|--|
| TCXO                 | <i>Temperature Compensated Crystal Oscillator</i>                            |
| $\text{tg } \varphi$ | Relação entre a Potência Reativa e Ativa                                     |
| Tx                   | <i>Transmitter</i> (Transmissão)   |
| UART                 | <i>Universal Asynchronous Receiver/Transmitter</i>                           |
| USB                  | <i>Universal Serial Bus</i>  |
| VRRM                 | <i>Maximum Repetitive Reverse Voltage</i> (Tensão Inversa Máxima Repetitiva) |



# Capítulo 1

## Introdução

**Resumo:** *Neste Capítulo é apresentado uma abordagem sobre a Unidade Digital de Controlo Auxiliar para Baterias de Condensadores. Apresenta-se o enquadramento da problemática em estudo, enuncia-se os principais pontos que motivaram a análise da mesma e refere-se qual o objeto de estudo implícito. Especifica-se a organização, bem como a notação utilizada na elaboração deste documento.*



## ***1.1. Introdução***

Nos dias de hoje a tecnologia, enquanto conjunto de instrumentos, métodos e técnicas que permitem o aproveitamento prático do conhecimento científico, tem uma importância crucial no progresso das sociedades modernas. O desenvolvimento industrial, económico, social e cultural é favorecido pela contínua criação de novos dispositivos tecnológicos. A procura continuada da eficiência dos processos de manufatura, causada pela necessidade incessante da redução de custos de produção, cria as condições ideais para que a preocupação com a eficiência energética também se torne fulcral. Esta preocupação é sobretudo importante na indústria uma vez que, na manufatura de bens são utilizadas máquinas, para processamento de produtos, que apresentam muitas vezes um consumo de energia eléctrica mais elevado que em outros setores.

De um modo geral as máquinas utilizadas na indústria têm como componente impulsionador um, ou vários motores eléctricos, transformadores entre outros equipamentos. Na grande maioria estes equipamentos funcionam pela interação entre campos eletromagnéticos que são criados com o auxílio de bobines, alimentadas em Corrente Alternada (CA) pela rede eléctrica.

Consequentemente, do ponto de vista da rede eléctrica, os motores eléctricos são encarados como cargas com um consumo de energia eléctrica de carácter indutivo. A operação das cargas indutivas, que necessitam de campos eletromagnéticos para o seu funcionamento, requer dois tipos de potência: ativa, reativa.

A Potência Ativa ( $P$ ) é a necessária para produzir trabalho, sendo esta consumida na execução do mesmo. A Potência Reativa ( $Q$ ) não produz trabalho, no entanto, é necessária para criar e manter campos eletromagnéticos em cargas indutivas, como por exemplo em transformadores ou motores. Portanto na operação de motores eléctricos existe energia reativa gerada em excesso que circula entre a carga e a fonte de alimentação, ocupando “espaço” no sistema eléctrico. Da junção das duas potências anteriores, em módulo e argumento surge a potência aparente ( $S$ ), ou seja a potência total gerada e transmitida à carga [1].

A eficiência de um sistema eléctrico depende de vários fatores, sendo um deles traduzido pelo Fator de Potência (FP). Um FP com valor elevado, isto é, perto de 1,0 indica uma compensação adequada da energia eléctrica, enquanto que valores baixos traduzem um mau aproveitamento e uma sobrecarga em todo o sistema eléctrico [2].

## ***1.2. Enquadramento e Motivação***

Um FP baixo indicia a presença de um excesso de energia reativa, o que levanta problemas ao sistema elétrico tais como a diminuição da capacidade de transporte de energia das linhas; o aumento de perdas como consequência do sobreaquecimento dos condutores, devido ao aumento da corrente elétrica associado à energia reativa; o aumento dos custos com a manutenção e operação, assim como a redução do tempo de vida útil dos equipamentos em resultado do aumento da corrente elétrica associada à energia reativa.

De modo a mitigar alguns destes problemas existem entidades que regulam estes serviços energéticos. Em Portugal a entidade é a Entidade Reguladora dos Serviços Energéticos (ERSE), que instaura tarifas que incluem o pagamento dos custos das diversas atividades do setor elétrico, de acordo com a energia consumida pelos clientes ligados à rede de distribuição, limitando a energia reativa. Esta entidade comporta várias opções de tarifas, mediante escalões energéticos e horários de utilização dos serviços que se adequam aos consumidores consoante as suas necessidades [3].

Por modo a manter a energia reativa dentro dos limites estipulados são utilizadas variadas soluções, sendo a predominante as Baterias de Condensadores (BC). No entanto, a imposição dos horários de utilização dos serviços energéticos é colocada em prática através de equipamentos que contêm relógios internos, que permitem executar ações em tempos predeterminados, como por exemplo autómatos programáveis.

“O controlo mais apurado do uso de energia reativa é mais uma medida adotada pelo Regulador, visando estimular o consumidor, através da redução de perdas e melhor desempenho de suas instalações, como também para o setor elétrico nacional, pela melhoria das condições operacionais e a liberação do sistema para atendimento a novas cargas com investimentos menores” [1].

Por conclusão e mediante o anteriormente descrito, o intuito desta dissertação compromete-se com o desenvolvimento de uma solução que permita ao consumidor, de uma forma eficaz, impor os horários predeterminados pelas tarifas da ERSE, utilizando recursos de forma eficiente quer económica quer tecnicamente.

### **1.3. Objeto de Estudo**

Numa primeira fase serão analisadas as características dos horários respetivos às tarifas da ERSE e as potencialidades e limitações do microcontrolador Arduino. O microcontrolador contém um *software* e *hardware* específico cujas potencialidades ainda não foram completamente testadas, no domínio da indústria.

O conceito de tecnologia aberta a toda a comunidade científica, com grande flexibilidade e custos reduzidos, tem na plataforma Arduino um forte representante, que tira partido da difusão das linguagens de programação como um fenómeno global e inclusivo. Foi por reconhecer a importância deste conceito, aliado às vantagens económicas, que se escolheu o microcontrolador Arduino como objeto de estudo para a execução de uma solução que permita o controlo horário de tarefas significativas na regulação do FP [4].

Outra vantagem da placa eletrónica Arduino é a possibilidade de utilização de extensões apropriadas denominadas por *shields*. Estas são placas de circuito com características e funções específicas que podem ser conectadas ao Arduino, encaixando-se perfeitamente por cima dele, ou numa *breadboard* conectada ao Arduino por fios condutores [4].

### **1.4. Objetivos**

O principal objetivo é verificar a possibilidade de desenvolver um protótipo funcional, como solução alternativa à atualmente utilizada pela empresa NEMOTEK, para o controlo auxiliar de BC, tendo em vista a inovação da solução por estes utilizada.

A solução já existente recorre ao uso de autómatos industriais, deixando a empresa dependente de terceiros. Assim, o cerne do protótipo é a substituição do autómato por um microcontrolador, no sentido de o tornar uma solução proprietária da empresa NEMOTEK, tendo em conta os aspetos económico e funcional.

O protótipo apresentado será concebido inicialmente utilizando um Arduino e posteriormente convertida numa placa eletrónica criada a partir do *Eagle CAD*. Esta placa tem o intuito de ser colocada nos armários de BC, na proximidade do Relé Varimétrico (RV),

complementando-o quanto a algumas funções de horário, relacionadas com tarifas da ERSE, não pretendendo substituir as tarefas realizadas pelo RV.

Como o objetivo é o desenvolvimento de um protótipo, não foram otimizados os custos ou tamanho do mesmo. No entanto, foi desenhado para poder evoluir para um produto final futuramente.

### ***1.5. Considerações Iniciais***

Importa referir que o propósito deste documento é tentar desenvolver uma solução proprietária da empresa NEMOTEK, preferencialmente mais económica do que a atualmente utilizada, pelo que a empresa se dispôs a auxiliar no necessário ao desenvolvimento deste projeto.

A empresa NEMOTEK cobre todas as gamas de soluções no âmbito da eficiência energética, sejam soluções para Baixa Tensão (BT) como de Média Tensão (MT) [5]. No entanto, na presente situação pretende-se realizar um estudo para a possibilidade de se otimizar, personalizar e reduzir os custos de uma solução de controlo de BC, tendo em conta o custo de fabricação baixo, ser segura e funcional e mais importante responder aos requisitos básicos impostos pela empresa, descritos nesta dissertação.

A solução a criar pretende somente aplicar os horários das tarifas impostas pela ERSE, sendo que a forma como estas são desenvolvidas não é relevante nesta dissertação. É igualmente importante referir que o equipamento a desenvolver não terá que calcular a energia reativa que se encontra a ser consumida nem o FP.

### ***1.6. Estrutura da dissertação***

A presente dissertação encontra-se estruturada em cinco Capítulos principais, tendo cada um subtópicos metodicamente organizados, desenvolvidos com base no objetivo principal já mencionado.

No primeiro Capítulo é realizada uma introdução à temática inserida nesta dissertação. Esta introdução está organizada num breve enquadramento e motivação alusiva ao objeto de

estudo, a placa eletrônica com o microcontrolador Arduino, tal como a sua funcionalidade. É também exposto a estrutura do documento e notação utilizada.

No segundo Capítulo é elaborado o Estado da Arte, onde através de documentos, artigos, catálogos e sites fidedignos é exposto a atualidade da tecnologia envolvente na temática, sendo apresentado outras soluções existentes.

No terceiro Capítulo é apresentado o trabalho desenvolvido para a criação da placa eletrônica (*hardware*). É também apresentado o algoritmo formulado, tendo em conta as funções pedidas pela empresa envolvida. No entanto, devido a questões de confidencialidade perante a empresa NEMOTEK e a não divulgação das suas soluções de negócio não será possível apresentar todo o código de programação efetuado para o microcontrolador, sendo apresentado apenas as partes consideradas relevantes do mesmo.

Por fim, é indicado as conclusões da análise de custos, relativamente à solução existente e a criada, não sendo possível discriminar detalhadamente a análise efetuada, por questões de confidencialidade.

No quarto Capítulo são expostos os resultados experimentais obtidos no processo de testes do equipamento, onde serão apresentados os problemas que surgiram e soluções adotadas para os mesmos.

No quinto Capítulo são exibidas as conclusões provenientes do estudo elaborado, para a construção do protótipo funcional, e apresentados possíveis desenvolvimentos futuros.

Por fim são apresentadas as referências bibliográficas utilizadas de modo a complementar o estudo efetuado, bem como outras informações suplementares colocadas em anexos ou em apêndices.

## ***1.7. Notação***

O documento desenvolvido apresenta a numeração das figuras e tabelas com dois algarismos, o primeiro diz respeito ao Capítulo onde estão inseridas e o segundo algarismo refere-se à numeração sequencial por cada Capítulo.

As referências bibliográficas estão expostas segundo o acordo com a norma do *Institute of Electrical and Electronics Engineers* (IEEE). Estas referências são numeradas

sequencialmente ao longo do documento, sendo a sua identificação apresentada entre parêntesis retos [].

A identificação de expressões ou informação complementar, sobretudo respeitante a abreviaturas são apresentadas entre parêntesis curvos ( ).

Frases ou textos retirados na íntegra de documentos já existentes, encontram-se entre aspas “ ”.

Expressões em língua estrangeira são apresentadas em formato itálico. E a simbologia utilizada no decorrer do texto segue o previamente apresentado na lista de acrónimos/simbologia.

Por fim, é de referir que todo o documento se encontra escrito segundo o acordo ortográfico atual, com a exceção das citações transcritas de outros documentos.

# Capítulo 2

## Estado da Arte

**Resumo:** *Neste Capítulo é apresentado de forma resumida a atualidade do tema envolvente no objeto de estudo. Iniciando com uma pequena abordagem sobre alguns aspectos relevantes da eletricidade até aos equipamentos que permitem a compensação do fator de potência. Por último são identificadas algumas soluções existentes ao objeto de estudo deste documento.*



## 2.1. Potência Ativa, Reativa e Aparente

Em sistemas elétricos, a corrente que flui através de um dispositivo ao longo de um determinado intervalo de tempo realiza um trabalho que pode ser definido como potência. Num sistema elétrico a CA a potência dissipada em cada instante, potência instantânea, é igual ao produto da corrente pela diferença de potencial ao longo do tempo. Há instantes em que a potência se anula, significando que a resistência não absorve potência e outros instantes em que a potência atinge o máximo. Na prática, apenas nos interessa o valor médio dessa potência. Esta potência média é a  $P$ . De uma forma geral diz-se que  $P$ , medida em kW, é o que realiza trabalho, gerando calor. A  $Q$ , medida em kVAr, é necessária para criar e manter campos eletromagnéticos das cargas indutivas, no entanto não produz trabalho. A  $S$  surge da junção da potência ativa e reativa, medida em kVA, ou seja, a potência total gerada e transmitida à carga, como ilustra a Figura 2.1 [1], [6].

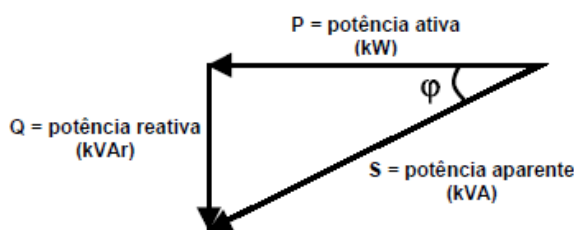


Figura 2.1 - Triângulo de potências, ilustração gráfica da relação entre  $P$ ,  $Q$  e  $S$  [1].

O FP, dado por  $\cos(\varphi)$ , é como a Figura 2.1 ilustra, a razão entre a potência ativa e a potência aparente. Portanto o FP indica qual a percentagem de potência total fornecida ( $S$ ) é efetivamente utilizada como  $P$ , traduzindo-se no grau de eficiência de sistemas elétricos. Os valores do FP estão compreendidos entre 0 e 1, em que valores perto de 1 representa um uso eficiente e valores perto de 0, um mau aproveitamento ou até mesmo uma sobrecarga para o sistema elétrico [1].

A existência de valores de FP baixos, representa quantidades elevadas de energia reativa. Nestas circunstâncias verifica-se um aumento da corrente total que circula nas redes de distribuição de energia elétrica e das unidades consumidoras, colocando sobrecargas nas subestações, nas linhas de transmissão e distribuição, prejudicando desta forma a estabilidade e as condições de aproveitamento dos sistemas elétricos. Além disso, prejudica também o consumidor, pois representa um acréscimo na sua fatura energética [1], [6].

Tudo o referido anteriormente, é aceite não considerando a existência da potência deformante, uma vez que esta não é relevante para o conteúdo deste documento.

## **2.2. Regulamentação, ERSE**

Em Portugal, os serviços energéticos são uma atividade regulada pela ERSE. Esta entidade é responsável pela regulação dos setores da eletricidade e do gás natural, e tem como objetivo zelar pelos interesses dos consumidores e incentivar uma utilização mais eficiente da energia.

Por forma a existir uma melhoria ao nível da eficiência energética, foram aprovadas novas regras em 2010, que visam controlar a circulação de energia reativa na rede elétrica nacional, impondo penalizações, de forma a atingir esse propósito [7].

Segundo o Regulamento da Rede de Distribuição (RRD) e Despacho n.º 7253/2010 da ERSE, existem novas regras para a faturação da energia reativa, introduzindo alterações nos limites do consumo da mesma, fracionado por diferentes escalões e penalizações. Estas normas conduzem a regras para a injeção de energia reativa por parte dos produtores e regras para os consumidores. “As principais alterações promovidas pela ERSE no regime de faturação da energia reativa consistem na fixação de um valor mais exigente para o limiar de faturação da energia reativa indutiva no período fora de vazio, na introdução de 3 escalões de preço em função da  $\text{tg } \varphi$  (0,3; 0,4 e 0,5), que se define pelo quociente entre a energia reativa e energia ativa, e na fixação do período de integração diário para determinação das quantidades de energia reativa a faturar para instalações em Muita Alta Tensão (MAT), Alta Tensão (AT) e Média Tensão (MT) em Portugal Continental, que disponham de telecontagem” [7] e [8].

Anteriormente à introdução de alterações ao RRD o período de faturação era mensal, não exigindo um controlo do trânsito de energia reativa muito rígido. Como tal, era permitido em alguns dias do mês, a ultrapassagem desses limites, podendo ser compensados nos dias seguintes. Com a entrada em vigor do novo regulamento, o período de integração passa a ser diário. Com esta alteração mais a dos 3 escalões, espera-se uma utilização mais eficiente nas redes elétricas, exigindo uma compensação mais próxima das cargas [7] - [10].

As instalações em Baixa Tensão (BT), estão divididas por Baixa Tensão Especial (BTE) ou Baixa Tensão Normal (BTN). Os clientes a BTE, têm o mesmo regime de faturação das redes anteriores, enquanto que para clientes de BTN existe algumas isenções [8]-[10].

O RRD determina que os produtores em regime especial, nas horas de Vazio Normal e Super Vazio, se estiverem a fornecer energia ativa, não podem injetar energia reativa. Nas horas de Cheia e Ponta apenas os produtores ligados à rede MT com potências de ligação igual ou inferior a 6 MW terão de acompanhar o fornecimento de energia ativa com a reativa no valor de 30 % da energia ativa injetada. Os restantes produtores, ou seja, ligados à rede AT e MT, terão de garantir uma  $\text{tg } \varphi = 0$ , quando estiverem a injetar energia ativa. Na situação em que os produtores se encontram a consumir energia ativa, não serão obrigados a cumprir os limites mencionados anteriormente, sendo caracterizados como consumidores [7].

Existem vários horários de entrega de energia elétrica. Estes dependem se o fornecimento é em Portugal Continental ou se o fornecimento é nos Arquipélagos da Madeira ou dos Açores [8], [9].

## ***2.3. Equipamentos para a Compensação da Energia Reativa***

### ***2.3.1. Baterias de Condensadores***

Em grande parte dos sistemas elétricos, o equipamento utilizado para a compensação da energia reativa são as BC, que são colocadas junto às cargas indutivas. Para este equipamento, pode-se utilizar as seguintes configurações: a compensação individual, a compensação parcial, a compensação centralizada e a compensação combinada [6].

Os armários com BC, apresentam o aspeto da Figura 2.2.



Figura 2.2 - Exemplos de armários contendo BC [24].

A compensação individual, tem como princípio a ligação dos condensadores aos terminais do equipamento cujo FP queremos corrigir, ou seja, a compensação é realizada localmente, apenas no equipamento a ser compensado, como demonstra a Figura 2.3. Desta forma é possível reduzir as perdas energéticas na instalação e melhorar os níveis de tensão. No entanto, é um processo bastante dispendioso e é difícil ajustar a potência a compensar com os valores disponíveis no mercado para os equipamentos [11].

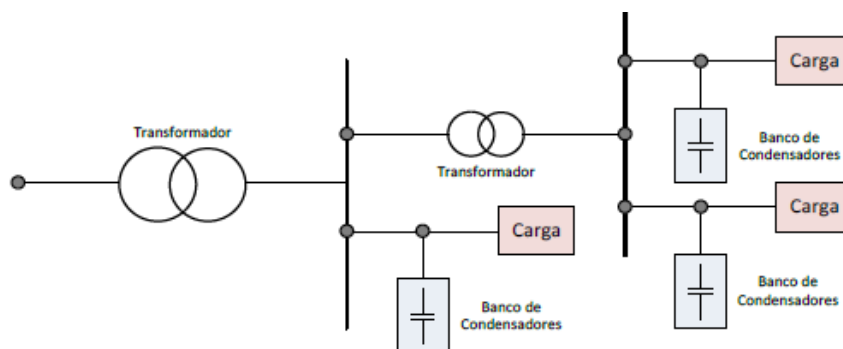


Figura 2.3 - Exemplo de uma rede com Compensação Individual [11].

A compensação parcial por grupos de cargas, consiste na instalação da BC de forma a compensar um setor, ou um conjunto de máquinas, como ilustra a Figura 2.4. A BC é colocada junto ao quadro parcial que alimenta esse sector [11].

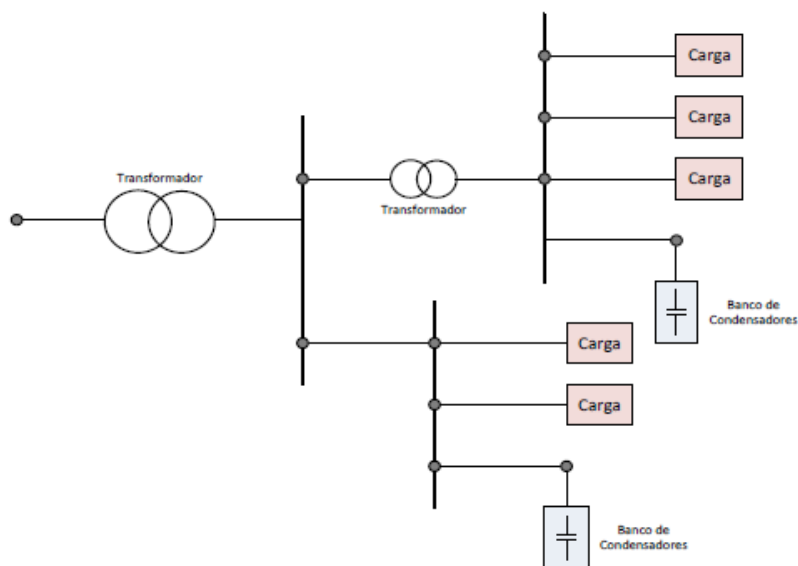


Figura 2.4 - Exemplo de uma rede com Compensação Parcial por Grupo de Cargas [11].

A compensação centralizada, consiste na instalação da BC à saída do transformador, caso a instalação seja em MT ou no Quadro Geral, se a instalação for em BT, como ilustra a Figura 2.5. Esta topologia é bastante vantajosa, caso a instalação seja constituída por diversas

cargas com diferentes potências e períodos de operação, para além de ser mais acessível o controlo das BC. Uma grande desvantagem é a probabilidade de avaria de equipamentos sensíveis (PCs, autómatos, etc.), devido a cavas de tensão e sobretensões provocadas pela comutação (ligar e desligar) de escalões [11], [12].

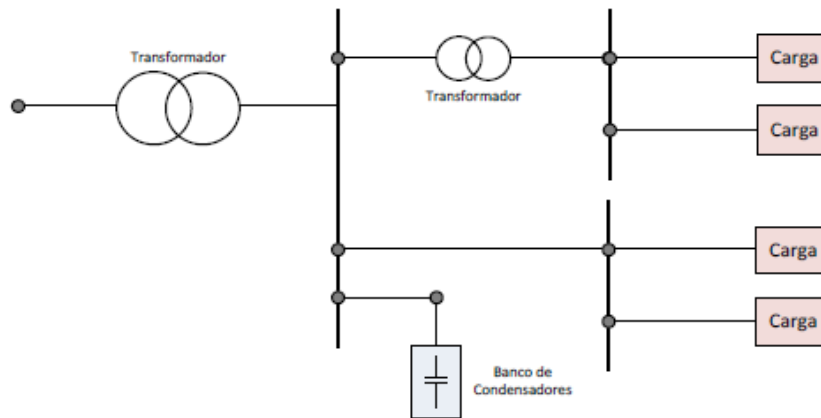


Figura 2.5 - Exemplo de uma rede com Compensação Centralizada [11].

A compensação combinada, como o nome sugere, consiste na combinação de dois dos três tipos de compensação anteriormente referidos. Por fim existe ainda a compensação com regulação automática que é uma solução utilizada juntamente com uma das soluções anteriores. Esta solução coloca os condensadores agrupados por escalões controláveis individualmente, utilizando um RV sensível às variações de energia reativa, este por sua vez comanda automaticamente a ligação dos condensadores necessários à obtenção do FP desejado [11], [13].

### 2.3.2. Compensador Síncrono

Os motores síncronos podem ser utilizados para compensação do FP por gerarem energia reativa, denominados de compensadores síncronos. Resumidamente um compensador síncrono é um modo de funcionamento do motor síncrono, em que a máquina não se encontra nem a gerar, nem a absorver P. A Q capacitiva fornecida por esta máquina à instalação, é em função da corrente de excitação e da carga no seu eixo [15].

Manusear um motor síncrono para correção do FP oferece muitas vantagens em relação às BC. É possível com um compensador síncrono corrigir o FP de forma mais suave (sem picos de tensão). Ou seja, a Q de um compensador síncrono pode ser continuamente ajustada, aumentando a corrente perante a diminuição da tensão da rede. Nas BC, verifica-se que a Q diminui quando a tensão da rede desce.

O compensador síncrono é pouco usual na correção do FP, uma vez que é bastante dispendioso, comparando com a solução BC e apresenta mais perdas. O motor síncrono torna-se interessante como solução, nas situações para potências superiores a 150 kW e funcionamentos por períodos longos, pois uma das grandes vantagens deste equipamento é conter uma vida útil longa e pouca manutenção, caso não possua escovas [13], [14], [16], [17].

Segundo a *Power Systems & Controls* contém um modelo *Series SC*, com o aspeto apresentado na Figura 2.6. Este modelo é uma máquina síncrona que fornece a Q necessária dinamicamente. O módulo é conectado à linha de alimentação e é executado, alterando a excitação dependendo da quantidade de correção do FP desejado e a quantidade de FP detetado pelos controlos do condensador. O condensador ajusta o nível de excitação automaticamente de forma a manter o FP na configuração correta. O *Series SC* monitoriza o FP e corrige-o reduzindo os transitórios de tensão e problemas associados à distorção harmónica encontradas no processo. Além disso está equipado para reiniciar automaticamente em caso de falha de energia [14].



Figura 2.6 - Condensador Síncrono, modelo *Series SC* [14].

### ***2.3.3. Conversor eletrónico (Compensador Síncrono Estático)***

Os Sistemas de transporte de CA que integrem dispositivos baseados em eletrónica de potência, e cujo objetivo seja aumentar o controlo e capacidade de transmissão de potência são denominados de *Flexible AC Transmission Systems (FACTS)*. Este tipo de dispositivos, utiliza elementos passivos como condensadores, bobines e resistências, em que a sua inserção é realizada pela comutação de semicondutores. O tiristor é o semicondutor mais utilizado

neste género de equipamentos, por conter a capacidade de suportar potências superiores em relação aos outros semicondutores e por ser usado no mercado há já vários anos [18].

O Compensador Síncrono Estático (STATCOM), é baseado nos equipamentos FACTS, portanto o STATCOM baseia-se em eletrónica de potência, utilizando dispositivos semicondutores de potência e tecnologias de processamento de sinais de última geração. Como ilustra a Figura 2.7, o STATCOM é colocado em paralelo com o sistema elétrico de modo a compensar a Q, sendo a sua função providenciar suporte de tensão, suprimindo a flutuação de tensão, ou regular o FP em áreas críticas do sistema [19], [20].

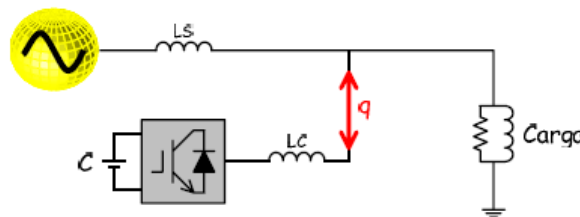


Figura 2.7 - Conexão do STATCOM a um sistema elétrico [19].

Inicialmente os STATCOM eram utilizados apenas em sistemas de transporte. No entanto, posteriormente foi possível instalá-los com o auxílio de dispositivos semicondutores e técnicas de comutação mais modernas, em sistemas de *Distribution Static Synchronous Compensator* (D-STATCOM) [19].

#### 2.3.4. Conversores Eletrónicos (CA/CC-CC/CA)

Conversores CA/CC comandados, utilizando dispositivos semicondutores à condução e ao corte, permitem quando comutados a alta frequência, o trânsito bidirecional de energia e a regulação do FP na ligação à rede elétrica.

Em muitos sistemas elétricos é usual utilizar-se conversores trifásicos CA/CC-CC/CA, como o ilustrado na Figura 2.8, para por exemplo: aplicações de acionamento de máquinas elétricas de CA (variadores de velocidade), carregamento de baterias em Corrente Contínua (CC) e aquecimento por indução e sistemas de aproveitamento eólico [21].

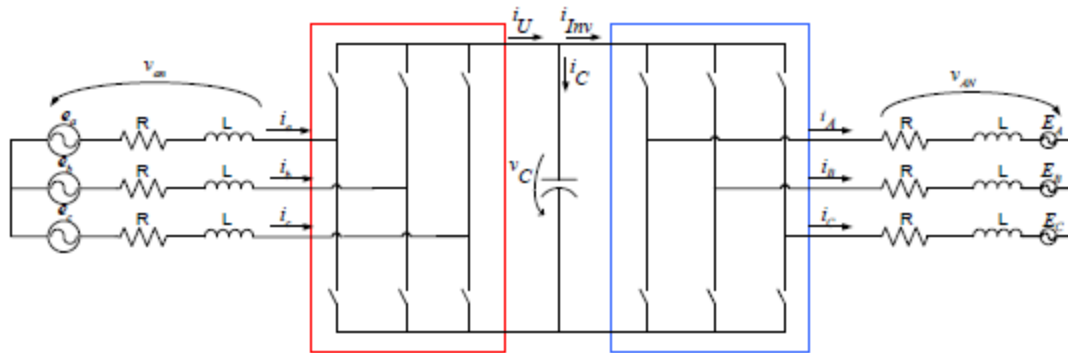


Figura 2.8 - Conversor trifásico CA/CC-CC/CA [21].

A interface da rede elétrica com pequenos geradores eólicos é um bom exemplo para aplicação de conversores (como ilustra as Figuras 2.9 a 2.11). A energia produzida por estes pequenos geradores eólicos é geralmente utilizada para consumo de cargas isoladas da rede elétrica. Os aerogeradores produzem energia na forma de CA, mas devido à inconstância da velocidade e direção do vento a energia produzida não tem uma frequência constante, como a utilizada nos sistemas elétricos, portanto não pode ser diretamente aproveitada. É usual estes aerogeradores estarem equipados com um retificador (conversor CA/CC) e disponibilizarem CC à saída (regulada para a ligação direta a baterias), e por um inversor (conversor CC/CA), para gerar uma forma de onda equivalente à rede elétrica [22].

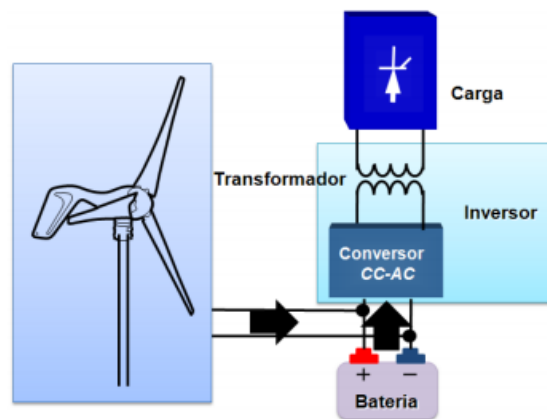


Figura 2.9 - Diagrama de blocos do sistema de alimentação de uma carga através de um inversor com transformador [22].

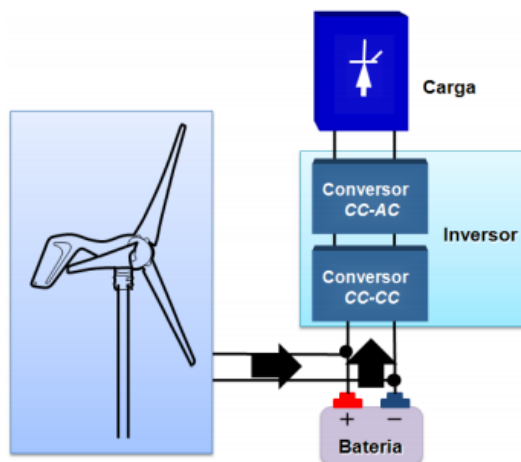


Figura 2.10 - Diagrama de blocos do sistema de alimentação de uma carga através de um inversor sem transformador [22].

### 2.3.5. Conversor eletrônico (Filtro Ativo Paralelo Monofásico)

Como ilustrado na Figura 2.11, os filtros ativos paralelos podem ser uma solução integrante entre fontes de energia renovável, como eólica e solar com a rede elétrica. Esta solução permite efetuar processos de fornecimento de energia elétrica à carga com ou sem auxílio da mesma, pois a energia pode ser proveniente de fontes renováveis, permitindo também corrigir problemas como o FP, harmônicas e compensação de desequilíbrios de corrente [22].

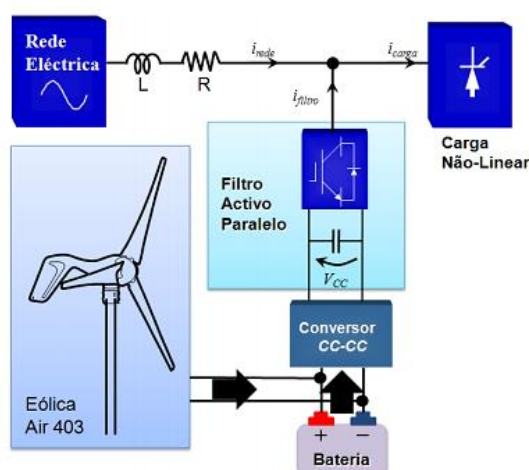


Figura 2.11 - Diagrama de blocos da interface entre a eólica e a rede elétrica, utilizando um filtro ativo paralelo [22].

### 2.3.6. Inversor Eletrónico (CC-CA) + STATCOM

Os inversores eletrónicos surgem do desenvolvimento das tecnologias da energia eólica. Existem países como os Estados Unidos, que alimentam propriedades com linhas monofásicas. Estes clientes têm muitas vezes, potencial para produzirem a sua energia utilizando turbinas eólicas de pequenas ou médias dimensões. O aumento destas turbinas poderá provocar problemas para utilitários locais, ao nível das harmónicas ou problemas com o FP [23].

Tradicionalmente os consumidores utilizam as BC para compensação do FP, o que aumenta o custo total do sistema. Uma alternativa moderna para compensar o FP é a utilização de STATCOM, em aplicações monofásicas e de pequenas ou médias turbinas, mas aumenta também o custo do sistema significativamente. Por conseguinte surge a solução com os inversores eletrónicos [23].

Acrescentando um inversor eletrónico com por exemplo um D-STATCOM, sem qualquer custo adicional, permite-se desta forma, não só converter o sinal elétrico de CC para CA, mas também fixar o FP adequado, de modo a injetar a Q suficiente à rede. A estratégia de combinar o inversor eletrónico e o D-STATCOM, possibilitou a criação de um novo inversor, em que este funciona como inversor com a opção de D-STATCOM, quando existe vento suficiente para produzir P e agir como D-STATCOM na inexistência de vento [23] e [19].

### 2.3.7. Resumo

O resumo dos equipamentos anteriormente descritos, utilizados para compensar o FP são apresentados na Tabela 2.1, com as vantagens e desvantagens de cada equipamento. É importante referir que existem outros equipamentos que permitem compensar o FP, no entanto foram apenas selecionados os mais diferentes e conhecidos.

Tabela 2.1 - Resumo dos equipamentos anteriormente descritos, utilizados para compensar o FP.

| Equipamentos             | Vantagens   | Desvantagens   |
|--------------------------|---|--|
| Bateria de Condensadores | - Método simples, económico, eficaz e de elevado rendimento [6];<br>- A instalação de BC pode | - Na introdução de um novo escalão é provocada uma distorção transitória na onda de tensão, sendo possível surgir um |

|                                       |  |   |
|---------------------------------------|--|---|
|                                       | reduzir perdas nas linhas, aumentar a capacidade no fornecimento de energia ativa, reduzir a fatura energética e melhorar os contornos da tensão na rede [6].                              | circuito ressonante, devido às capacidades das BC e indutâncias da rede, podendo levar à deterioração da onda de tensão [6];<br>- Funcionamento por escalões e como tal a compensação não é completa, ao contrário das restantes soluções aqui apresentadas [6].                  |
| Compensadores Síncronos               | - A correção do FP é feita de forma suave (sem picos de tensão) e não afeta sistemas carregados de harmónicas de corrente (sem ressonância) [14].  | - São necessárias manobras para o arranque, em que as soluções utilizadas são: um motor de indução auxiliar, ou arranque assíncrono produzindo um binário de escorregamento à custa do enrolamento amortecedor e provavelmente do enrolamento de excitação curto-circuitado [16]. |
| Conversor eletrónico (STATCOM)        | - Podem ser utilizados tanto em sistemas de transmissão como de distribuição, pois suportam potências elevadas [19];<br>- São facilmente implementados na rede de distribuição de BT [19]. | - Em MT não pode ser implementado de forma direta, pois a alta frequência de comutação necessária para gerar uma onda com baixo conteúdo harmónico contrasta com os altos níveis de potência do sistema [19].   |
| Conversores Eletrónicos (CA/CC-CC/CA) | - Não existem perdas significativas na comutação dos dispositivos semicondutores [21];<br>- Colocando o conversor em paralelo entre o transformador de distribuição e o local de           | - Tem de existir cuidados com os dispositivos semicondutores, como por exemplo garantir que apenas um e só um dispositivo está à condução em cada braço ou que os limites de corrente não são ultrapassados [21].   |

|   |  |  |
|---|--|--|
|   | armazenamento de energia, torna o sistema mais atrativo economicamente, pois não necessita da utilização de baterias de grandes capacidades ou super-condensadores [21].   |  |
| Conversor eletrônico (Filtro Ativo Paralelo Monofásico) | <ul style="list-style-type: none"> <li>- Permite integrar diferentes fontes de energia alternativa (partilhando o mesmo barramento de CC);</li> <li>- Soluciona alguns problemas de qualidade de energia elétrica, como a eliminação de harmónicas, a correção do FP, o equilíbrio das correntes e tensões e atenuar problemas de cavas de tensão e sobretensões [6], [22].</li> </ul> | <ul style="list-style-type: none"> <li>- Tem limitações técnicas, como por exemplo ao nível da potência (na ordem dos kVA) e a nível económico (dispendiosos) [6].</li> <li>- É um equipamento que deverá ser colocado perto da carga, de modo a restringir os problemas, pois a jusante do filtro o problema continuará a existir [6].</li> </ul> |

## 2.4. Microcontroladores

O microcontrolador é um sistema computacional inserido num único circuito integrado, que pode ser programado para funções específicas de modo a controlar um circuito ou um processo. A estrutura interna de um microcontrolador apresenta uma unidade central de processamento, ou *Central Processing Unit* (CPU), bem como circuitos de memória voláteis de leitura e escrita para armazenamento de dados do tipo *Random Access Memory* (RAM), ou circuitos de memórias não voláteis somente de leitura para armazenamento permanente de dados do tipo *Read Only Memory* (ROM), dispositivos periféricos como Conversores Analógicos/Digitais (ADC), Conversores Digitais/Analógicos (DAC), portas de Entradas e Saídas (I/O) digitais para propósito geral, um sistema *clock* (relógio) para permitir sequência de atividades do CPU [24].

Estes equipamentos têm inúmeras vantagens, como consumir pouca energia (em torno dos *miliWatts*), permanecer em modo “hibernação” enquanto aguardam por um acontecimento ou evento que o coloque a funcionar novamente, e são dispositivos de baixo custo e compactos.

Para o funcionamento do microcontrolador é necessário: desenvolver o programa que controla o processo (*software*) e a interface entre o mundo externo e o microcontrolador (*hardware*) [24].

### 2.4.1. Arduino

O Arduino nasce em Itália no ano de 2005, no *Interaction Design Institute* (IVREA), através do professor *Massimo Banzì*, que pretendia ensinar aos seus alunos conceitos de programação e eletrónica, contudo deparou-se com impasses, que se pretendiam essencialmente com o custo de uma placa eletrónica. Assim sendo, o Arduino foi criado com o intuito de qualquer indivíduo conseguir usufruir das capacidades de um equipamento eletrónico, desenvolvido por ele mesmo. Ou seja, desde programadores, empreendedores, especialistas em eletrónica ou até mesmo para os curiosos sem qualquer formação. Por esse motivo o Arduino traz inúmeras vantagens, o facto de ser bastante barato, de fácil uso e ser flexível o suficiente para usos avançados.

O Arduino é uma plataforma eletrónica de código aberto (*open source*), que fornece um conjunto de ferramentas para protótipos eletrónicos que visa tornar mais fácil a criação de aparelhos eletrónicos, baseando-se em *hardware* e *software* simples de usar. Portanto, o Arduino oferece a placa eletrónica (*hardware*) e proporciona que haja um ambiente de desenvolvimento e partilha de dados (bibliotecas) entre todos os que usufruem desta plataforma, utilizando a linguagem C/C++ (*software*). Além do já referido, este ainda pode ser equipado com *shields*, que são placas de *hardware* anexadas à placa Arduino, contendo circuitos eletrónicos que permitem desempenhar diversas funções adicionais [4], [25], [53], [54].

Em suma, segundo a OP SERVICES, o Arduino “é uma plataforma extremamente poderosa e deu início a um movimento que está democratizando o desenvolvimento de dispositivos eletrónicos em todo o mundo” [25].

Existem vários modelos de placas de Arduino, desde as mais simples às mais complexas, como ilustra as Figuras 2.12 e 2.13.



Figura 2.12 - Placas de constituição simples, nível principiante.







Figura 2.13 - Placa de constituição mais complexa, nível avançado.

### 2.4.2. Controllino

Uma das grandes desvantagens na utilização das placas de Arduino em relação aos *Programmable Logic Controllers* (PLCs) é a robustez do equipamento. No seguimento deste aspeto surge o *controllino*, que foi o primeiro equipamento PLC pronto para o uso industrial, que se baseia na utilização da placa Arduino, com o seu *software* compatível em *open-source*. Este equipamento serve praticamente para qualquer controlo de automação, desde fins educacionais, como para fins industriais.

O *controllino* tornou-se uma solução tão apetecida por tantos fabricantes e profissionais, que o mesmo, contém atualmente 3 tamanhos, o *MINI*, o *MAXI* e o *MEGA*, (como descreve a Tabela 2.2). Além disso oferece também, uma versão livremente personalizável, chamada de *CONTROLLINO myPLC*. Esta versão permite criar o seu próprio *controllino* num configurador *online*, com as I/O e componentes necessários, combatendo um dos grandes problemas dos PLCs, que é a compra de módulos extras com mais I/O [26].

Tabela 2.2 – Modelos do *Controllino* [26].

|  |  |  |  |
|--|--|--|--|
|  <p style="text-align: center;"><b>MINI</b></p> |  <p style="text-align: center;"><b>MAXI</b></p> |  <p style="text-align: center;"><b>MAXI Automation</b></p> |  <p style="text-align: center;"><b>MEGA</b></p> |
| <p>Pequeno, mas poderoso.</p>  | <p>A escolha perfeita para projetos grandes (<i>best-seller</i>)</p>   | <p>Especificamente adaptado às necessidades dos especialistas em automação.</p>  | <p>Com muitas entradas e saídas.</p>   |

Como referido na Tabela 2.2, o *controllino MAXI*, é o mais vendido, sendo as suas características as ilustradas na Figura 2.14.

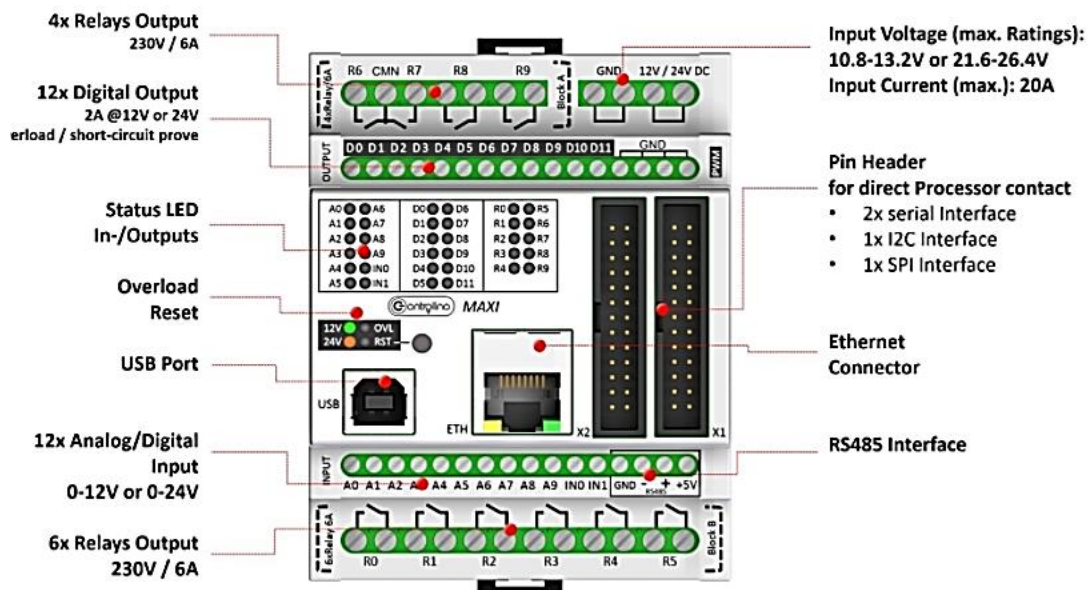


Figura 2.14 - Características de um *Controllino MAXI* [26].

### 2.4.3. Autómatu Programável (PLC)

Os autómatos programáveis, ou também vulgarmente conhecidos de PLCs, são unidades de *hardware*, constituídas por um CPU e memória, geralmente utilizados em ambiente industrial, no controlo de máquinas e processos.

Independentemente do tamanho, da complexidade ou arquitetura, regra geral todos os autômatos programáveis contêm os mesmos componentes básicos (Figura 2.15) e as mesmas características funcionais [27].

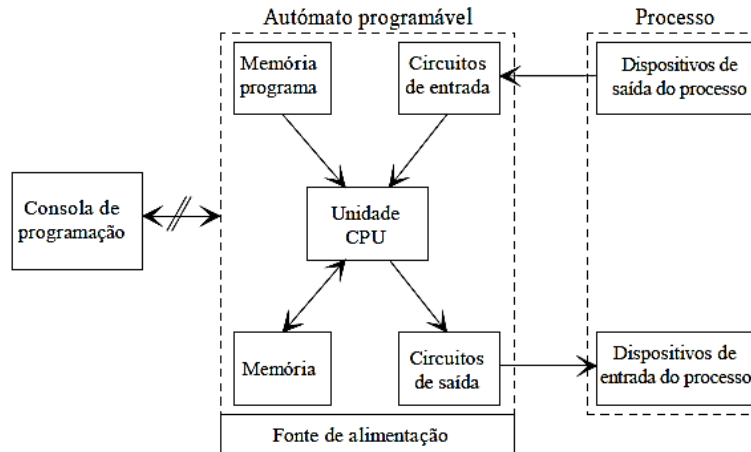


Figura 2.15 - Arquitetura de um autômato programável [27].

Um PLC é composto por: um CPU, que executa o programa armazenado em memória de programa; por um sistema de I/O que fornece a ligação física entre o CPU e o processo a controlar; memórias dos tipos RAM e ROM entre outras, de modo a armazenar o programa de controlo (memória de programa) e possibilitar o armazenamento e transferência de dados; várias linguagens de programação [27].

Os PLCs estão preparados para executarem ciclicamente uma sequência de tarefas, ou seja, funcionam por fases que se repetem continuamente. No início de cada ciclo o estado das entradas físicas, que depende do processo a controlar, é lido e armazenado nas respetivas variáveis de entradas, na memória de dados. A informação presente nas variáveis de entradas é processada pelo CPU, que de acordo com o programa existente na memória, toma decisões, executa operações, e atualiza o estado das variáveis de saídas, na memória de dados. Uma vez completa a execução do programa o PLC executa uma série de tarefas internas de diagnósticos e comunicações. Caso não existam erros os valores das variáveis de saída são transferidos para as saídas físicas. Estas são atualizadas e o autômato passa ao ciclo seguinte, recomeçando a sequência de tarefas [27].

Existe uma separação da memória de armazenamento de programa, com a memória de leitura e escrita de dados, o que permite o armazenamento de programas no PLC, mesmo com a alimentação desligada. Além disso, basta inserir o programa

uma vez, passando o PLC a ser autossuficiente quando em execução, ou enquanto não é carregado outro programa [27].

Usualmente todos os autômatos permitem programar em variadas linguagens, sendo estas a linguagem *ladder* que utiliza uma notação simbólica que, quando interligadas entre si, numa determinada forma, permitem executar etapas, com uma estrutura cronológica; a linguagem *Grafcet*, que utiliza uma metodologia aplicada em representação gráfica de algoritmos utilizados em sistemas de controlo, a representação gráfica permite distinguir as etapas e condições facilmente; e a linguagem de texto estruturado (*structured text*), que como o nome indica é uma linguagem textual, que possibilita solucionar problemas complexos com comandos básicos de programação (*if, else, while...*) [27].

A empresa envolvente no projeto utiliza como solução um autômato programável, sendo este o apresentado na Figura 2.16. Este é constituído por um microcontrolador ATmega128 16AU, com 24 pinos de I/O digitais, dos quais 4 podem ser utilizados como saídas *Pulse-Width Modulation* (PWM); 8 entradas analógicas; alimentação a 24 VCC; interface Bluetooth; 3 m de cabo de ligação em *Universal Serial Bus* (USB); bibliotecas específicas para funções (CD-ROM) e múltiplas linguagens de programação (Anexo II).



Figura 2.16 - Autômato programável Crouzet XD10.

### 2.4.4. Comparação entre Arduino e Autômato

As vantagens e desvantagens dos microcontroladores, referidos no subcapítulo 2.4, são apresentadas resumidamente na Tabela 2.3 enquanto que na Tabela 2.4 são apresentadas as diferenças entre os dois microcontroladores.

Tabela 2.3 - Resumo dos microcontroladores descritos, enumerando as vantagens e desvantagens.

| Equipamentos               | Vantagens  | Desvantagens  |
|----------------------------|--|---|
| Arduino                    | <ul style="list-style-type: none"> <li>- Baixo custo;</li> <li>- Acessível a qualquer pessoa;</li> <li>- Muita informação disponível;</li> <li>- Código aberto (<i>open-source</i>) [4] e [25].</li> </ul> | <ul style="list-style-type: none"> <li>- Pouco robusto;</li> <li>- Trabalha com sinais a 5 V [4].</li> </ul>                  |
| Autômato Programável (PLC) | <ul style="list-style-type: none"> <li>- Bastante Robusto;</li> <li>- Muito fiável;</li> <li>- Possibilidade de expansão por módulos.</li> </ul>   | <ul style="list-style-type: none"> <li>- Custo superior;</li> <li>- Mais difícil de programar por qualquer pessoa.</li> </ul> |

Tabela 2.4 – Diferenças entre os dois microcontroladores.

| Arduino  | PLC   |
|--|---|
| Programação em C/C++   | Programação em <i>Ladder</i> , texto estruturado e <i>Grafcet</i> . |
| <i>Hardware</i> frágil e de código aberto.                           | Traz certificações industriais e tem características de segurança.  |
| Disponível a qualquer um.  | Adequado para aplicações industriais.                               |
| Grande quantidade de informação e <i>software</i> entre os usuários. | Menor quantidade de informação de exemplos do seu <i>software</i> . |

# Capítulo 3

## Trabalho Desenvolvido

**Resumo:** *Neste Capítulo é exposta a conceção do protótipo, a interface entre o utilizador e o equipamento criado, e o código desenvolvido. É apresentado também o aspeto final do protótipo funcional.*

*É realizada também uma comparação qualitativa, a nível monetário, entre a solução criada e a já existente.*



### 3.1. Introdução

O algoritmo desenvolvido tem o intuito de realizar um controlo auxiliar de uma BC, ou seja, de impor os horários concebidos pela ERSE. Portanto, a solução aplicada neste documento pretende ser um protótipo funcional, no qual não está direcionada para realizar o controlo da energia reativa produzida ou consumida por uma instalação.

Segundo a ERSE, o sistema tarifário e a metodologia de cálculo das tarifas, definidas no regulamento do mesmo, devem promover de forma transparente: a eficiência dos recursos, a igualdade e justiça entre todos os consumidores que utilizam infraestruturas de rede. Resumidamente, segundo esta entidade, o sistema tarifário pretende manter o equilíbrio económico e financeiro das empresas reguladas, a qualidade do fornecimento de energia elétrica e a estabilidade da evolução tarifária [28].

Os tarifários estão divididos em períodos trimestrais e períodos horários. Os períodos trimestrais não fazem sentido serem explicados neste documento, uma vez que não são utilizados. Os períodos horários encontram-se organizados ainda segundo ciclo diário ou ciclo semanal. Ciclo diário são os períodos horários que são iguais em todos os dias do ano, enquanto que ciclo semanal são os períodos horários que diferem entre dias úteis, fim de semana e feriados. Além disso, estes ciclos são diferentes dependendo da Região e do tipo de energia fornecido [28].

Nos horários da ERSE, existem 3 Regiões, Portugal Continental, Região Autónoma dos Açores e Região Autónoma da Madeira. Em relação ao tipo de fornecimento existe para MAT, AT, MT e BT. Em que a BT está subdividida em Baixa Tensão Especial (BTE), para potências contratadas superiores a 41,4 kW e Baixa Tensão Normal (BTN), para potências inferiores ou iguais a 41,4 kW [28].

O cliente opta pelo período horário conveniente, consoante as características ditas anteriormente e outras diferenças, que se encontram expostas nas Tabelas 3.1 a 3.7.

Tabela 3.1 – Período Horário em Portugal Continental, Ciclo Semanal, todos os fornecimentos [28].

| <b>Ciclo Semanal para todos os fornecimentos em Portugal Continental</b> |   |                                    |                                       |                 |
|--|---|------------------------------------|---------------------------------------|-----------------|
|  | <b>Período de hora legal de Inverno</b> |                                    | <b>Período de hora legal de Verão</b> |                 |
| <b>De Segunda-</b>   | Ponta:                                  | 09:30 – 12:00 h<br>18:30 – 21:00 h | Ponta:                                | 09:15 – 12:15 h |

|                            |               |  |               |  |
|----------------------------|---------------|--|---------------|--|
| <b>feira a Sexta-feira</b> | Cheias:       | 07:00 – 09:30 h<br>12:00 – 18:30 h<br>21:00 – 24:00 h                    | Cheias:       | 07:00 – 09:15 h<br>12:15 – 24:00 h                                       |
|                            | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 07:00 h                                       | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 07:00 h                                       |
|                            | Super Vazio:  | 02:00 – 06:00 h  | Super Vazio:  | 02:00 – 06:00 h  |
| <b>Sábado</b>              | Cheias:       | 09:30 – 13:00 h<br>18:30 – 22:00 h                                       | Cheias:       | 09:00 – 14:00 h<br>20:00 – 22:00 h                                       |
|                            | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 09:30 h<br>13:00 – 18:30 h<br>22:00 – 24:00 h | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 09:00 h<br>14:00 – 20:00 h<br>22:00 – 24:00 h |
|                            | Super Vazio:  | 02:00 – 06:00 h  | Super Vazio:  | 02:00 – 06:00 h  |
|                            |               |  |               |  |
| <b>Domingo e Feriados</b>  | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 24:00 h                                       | Vazio Normal: | 00:00 – 02:00 h<br>06:00 – 24:00 h                                       |
|                            | Super Vazio:  | 02:00 – 06:00 h  | Super Vazio:  | 02:00 – 06:00 h  |

Tabela 3.2 – Período Horário em Portugal Continental, Ciclo Semanal Opcional (MAT, AT e MT) [28].

| <b>Ciclo Semanal Opcional para MAT, AT e MT em Portugal Continental</b> |   |  |                                       |  |
|---|---|--|---------------------------------------|--|
|   | <b>Período de hora legal de Inverno</b> |  | <b>Período de hora legal de Verão</b> |  |
| <b>De Segunda-feira a Sexta-feira</b>                                   | Ponta:                                  | 17:00 – 22:00 h  | Ponta:                                | 14:00 – 17:00 h  |
|   | Cheias:                                 | 00:00 – 00:30 h<br>07:30 – 17:00 h<br>22:00 – 24:00 h                    | Cheias:                               | 00:00 – 00:30 h<br>07:30 – 14:00 h<br>17:00 – 24:00 h                    |
|   | Vazio Normal:                           | 00:30 – 02:00 h<br>06:00 – 07:30 h                                       | Vazio Normal:                         | 00:30 – 02:00 h<br>06:00 – 07:30 h                                       |
|   | Super Vazio:                            | 02:00 – 06:00 h  | Super Vazio:                          | 02:00 – 06:00 h  |
| <b>Sábado</b>   | Cheias:                                 | 10:30 – 12:30 h<br>17:30 – 22:30 h                                       | Cheias:                               | 10:00 – 13:30 h<br>19:30 – 23:00 h                                       |
|   | Vazio Normal:                           | 00:00 – 03:00 h<br>07:00 – 10:30 h<br>12:30 – 17:30 h<br>22:30 – 24:00 h | Vazio Normal:                         | 00:00 – 03:30 h<br>07:30 – 10:00 h<br>13:30 – 19:30 h<br>23:00 – 24:00 h |

|                           |               |                                    |               |                                    |
|---------------------------|---------------|------------------------------------|---------------|------------------------------------|
|                           | Super Vazio:  | 03:00 – 07:00 h                    | Super Vazio:  | 03:30 – 07:30 h                    |
| <b>Domingo e Feriados</b> | Vazio Normal: | 00:00 – 04:00 h<br>08:00 – 24:00 h | Vazio Normal: | 00:00 – 04:00 h<br>08:00 – 24:00 h |
|                           | Super Vazio:  | 04:00 – 08:00 h                    | Super Vazio:  | 04:00 – 08:00 h                    |

Tabela 3.3 – Período Horário em Portugal Continental, Ciclo Diário (BTE e BTN) [28].

| <b>Ciclo Diário para fornecimentos em BTE e BTN em Portugal Continental</b> |   |                                       |   |
|---|---|---------------------------------------|---|
| <b>Período de hora legal de Inverno</b>                                     |   | <b>Período de hora legal de Verão</b> |   |
| Ponta:  | 09:00 – 10:30 h<br>18:00 – 20:30 h                    | Ponta:                                | 10:30 – 13:00 h<br>19:30 – 21:00 h                    |
| Cheias:   | 08:00 – 09:00 h<br>10:30 – 18:00 h<br>20:30 – 22:00 h | Cheias:                               | 08:00 – 10:30 h<br>13:00 – 19:30 h<br>21:00 – 22:00 h |
| Vazio Normal:   | 06:00 – 08:00 h<br>22:00 – 02:00 h                    | Vazio Normal:                         | 06:00 – 08:00 h<br>22:00 – 02:00 h                    |
| Super Vazio:  | 02:00 – 06:00 h                                       | Super Vazio:                          | 02:00 – 06:00 h                                       |

Tabela 3.4 – Período Horário na Região Autónoma dos Açores, Ciclo Diário para todos os fornecimentos [28].

| <b>Ciclo Diário para fornecimentos na Região Autónoma dos Açores</b> |   |                                       |   |
|--|---|---------------------------------------|---|
| <b>Período de hora legal de Inverno</b>                              |   | <b>Período de hora legal de Verão</b> |   |
| Ponta:   | 09:30 – 11:00 h<br>17:30 – 20:00 h                    | Ponta:                                | 09:00 – 11:30 h<br>19:30 – 21:00 h                    |
| Cheias:  | 08:00 – 09:30 h<br>11:00 – 17:30 h<br>20:00 – 22:00 h | Cheias:                               | 08:00 – 09:00 h<br>11:30 – 19:30 h<br>21:00 – 22:00 h |
| Vazio Normal:  | 05:30 – 08:00 h<br>22:00 – 01:30 h                    | Vazio Normal:                         | 05:30 – 08:00 h<br>22:00 – 01:30 h                    |
| Super Vazio:   | 01:30 – 05:30 h                                       | Super Vazio:                          | 01:30 – 05:30 h                                       |

Tabela 3.5 – Período Horário na Região Autónoma dos Açores, Ciclo Diário Opcional (MT e BTE) [28].

| <b>Ciclo Diário Opcional (MT e BTE) na Região Autónoma dos Açores</b> |                                    |                                       |   |
|---|------------------------------------|---------------------------------------|---|
| <b>Período de hora legal de Inverno</b>                               |                                    | <b>Período de hora legal de Verão</b> |   |
| Ponta:  | 17:00 – 21:00 h                    | Ponta:                                | 09:00 – 11:30 h<br>19:30 – 21:00 h                    |
| Cheias:   | 08:00 – 17:00 h<br>21:00 – 22:00 h | Cheias:                               | 08:00 – 09:00 h<br>11:30 – 19:30 h<br>21:00 – 22:00 h |
| Vazio Normal:   | 05:30 – 08:00 h<br>22:00 – 01:30 h | Vazio Normal:                         | 05:30 – 08:00 h<br>22:00 – 01:30 h                    |
| Super Vazio:  | 01:30 – 05:30 h                    | Super Vazio:                          | 01:30 – 05:30 h                                       |

Tabela 3.6 – Período Horário na Região Autónoma da Madeira, Ciclo Diário para todos os fornecimentos [28].

| <b>Ciclo Diário para todos os fornecimentos na Região Autónoma da Madeira</b> |   |                                       |   |
|---|---|---------------------------------------|---|
| <b>Período de hora legal de Inverno</b>                                       |   | <b>Período de hora legal de Verão</b> |   |
| Ponta:  | 10:30 – 12:00 h<br>18:30 – 21:00 h                    | Ponta:                                | 10:30 – 13:00 h<br>20:30 – 22:00 h                    |
| Cheias:   | 09:00 – 10:30 h<br>12:00 – 18:30 h<br>21:00 – 23:00 h | Cheias:                               | 09:00 – 10:30 h<br>13:00 – 20:30 h<br>22:00 – 23:00 h |
| Vazio Normal:   | 06:00 – 09:00 h<br>23:00 – 02:00 h                    | Vazio Normal:                         | 06:00 – 09:00 h<br>23:00 – 02:00 h                    |
| Super Vazio:  | 02:00 – 06:00 h                                       | Super Vazio:                          | 02:00 – 06:00 h                                       |

Tabela 3.7 – Período Horário na Região Autónoma da Madeira, Ciclo Diário Opcional (AT, MT e BTE) [28].

| <b>Ciclo Diário Opcional (AT, MT e BTE) na Região Autónoma da Madeira</b> |                                    |                                       |   |
|---|------------------------------------|---------------------------------------|---|
| <b>Período de hora legal de Inverno</b>                                   |                                    | <b>Período de hora legal de Verão</b> |   |
| Ponta:  | 18:00 – 22:00 h                    | Ponta:                                | 10:30 – 13:00 h<br>20:30 – 22:00 h                    |
| Cheias:   | 09:00 – 18:00 h<br>22:00 – 23:00 h | Cheias:                               | 09:00 – 10:30 h<br>13:00 – 20:30 h<br>22:00 – 23:00 h |
| Vazio Normal:   | 06:00 – 09:00 h<br>23:00 – 02:00 h | Vazio Normal:                         | 06:00 – 09:00 h<br>23:00 – 02:00 h                    |
| Super Vazio:  | 02:00 – 06:00 h                    | Super Vazio:                          | 02:00 – 06:00 h                                       |

Segundo o Decreto-lei 17/96, de 8 de março, o ano divide-se em período de verão e inverno, em que o período de verão inicia no último domingo de março e o período de inverno inicia no último domingo de outubro [29].

É possível verificar, pelos horários evidenciados nas Tabelas 3.1 a 3.7, que os dias são divididos em horas de Ponta, Cheias, Vazio Normal e Super Vazio. As horas de Ponta são as horas onde o consumo de eletricidade é mais caro, enquanto que as Horas de Vazio (HV) são as horas onde o consumo de eletricidade é mais barato. Por sua vez, nas horas de Cheia o custo do consumo de eletricidade é mediano. Nas HV encontram-se englobadas as horas de Vazio Normal e as horas de Super Vazio, tal como o nome indica, as horas de Super Vazio são as mais económicas [2], [28].

Segundo a empresa do setor energético Energias de Portugal (EDP), é durante o dia que ocorrem as Horas Fora de Vazio (HFV), onde o FP dos consumos é normalmente indutivo, pelo que se deve compensar localmente a energia reativa indutiva. Durante a noite, nas HV, devido à redução das cargas elétricas, o carácter capacitivo das linhas e cabos prevalece, sendo sugerindo por isso desligar as BC ou outros aparelhos que façam a compensação da energia reativa, uma vez, que a injeção de energia reativa capacitiva na rede, também está sujeita a faturação [30]. Portanto, quando existe consumo de energia reativa, caracterizando uma situação de compensação insuficiente, o FP é dito indutivo. Na situação

de haver fornecimento de energia reativa à rede, a compensação é caracterizada como excessiva, logo o FP é dito capacitivo [1].

Como foi referido no Capítulo 2, as BC funcionam por escalões e como tal, a compensação da energia reativa indutiva geralmente não é exata, podendo ser por defeito ou por excesso. Assim sendo, é muitas vezes introduzido um escalão de carácter indutivo, de forma a contrabalançar [6]. Perante o já referido anteriormente, são também introduzidos estes escalões de carácter indutivo nas situações de HV.

### **3.2. Requisitos**

Segundo o documento fornecido pela Empresa (Anexo I), a solução desenvolvida tem que responder a determinados requisitos físicos e de controlo.

#### Requisitos Físicos:

- A placa de controlo, será colocada em armários de BC, na proximidade do RV, estando por isso sujeito a interferências eletromagnéticas, perturbações harmónicas e variações térmicas;
- Robustez eléctrica para suportar tensões de 350VAC a 450 VAC;
- Deve conter 3 saídas a relé, 5 A e 400 VAC;
- As 3 saídas podem ser reconfiguradas para o corte da bateria (da alimentação do RV) ou para forçar a entrada em serviço de um escalão;
- Deve conter no mínimo 2 entradas;
- Seleccionador/ visualizador de horário, permitindo ao utilizador seleccionar o horário e visualizar o estado das saídas, alimentação e entradas;
- Porta de comunicação para ligação ao computador.

#### Requisitos de Controlo:

- Capacidade para suportar em memória os vários horários;
- Capacidade de armazenar vários feriados de pelo menos 5 anos;
- O sistema deve alterar automaticamente a hora legal (horário de verão e horário de inverno);
- Cada saída deve permitir ligar ou desligar a BC ou atuar apenas num condensador (escalão);

- Cada entrada pode funcionar como alarme ou controlo, a entrada com a função de alarme sobrepõe-se à de controlo;
- Cada saída pode ser acionada segundo a entrada, ou segundo o horário e a entrada, ou apenas pelo horário, dependendo das funções atribuídas às entradas;
- A entrada que funcionar como alarme deve conter uma indicação sonora;
- Deve haver indicação visual: caso uma entrada tenha sido ativada; quando as datas de feriados incluídas estejam a 3 meses de ser desatualizadas e quando a data atual é a da manutenção;
- Deve ser possível ativar uma data de manutenção pelo técnico;
- Deve ser criado um processo de transferência de dados simples e de interface com o cliente. Deve ser possível os serviços técnicos enviarem ao cliente de forma digital os horários e as programações de controlo e este transferi-las para o equipamento. Assim como o processo inverso para diagnóstico dos serviços técnicos;
- Deve ser criado um registo de acontecimentos (mínimo 100 registos), com a indicação da data-hora de cada acontecimento, para efeito de identificação de erros de funcionamento. Os acontecimentos mínimos a registar são a alteração das entradas e a alteração das saídas.

### ***3.3. Solução Desenvolvida, Hardware***

#### ***3.3.1. Arquitetura do equipamento a desenvolver***

Antes da conceção de uma placa eletrónica e por sua vez do equipamento, é necessário estabelecer a arquitetura da mesma, ou seja, identificar os dispositivos e os protocolos que utilizam, como ilustra a Figura 3.1.

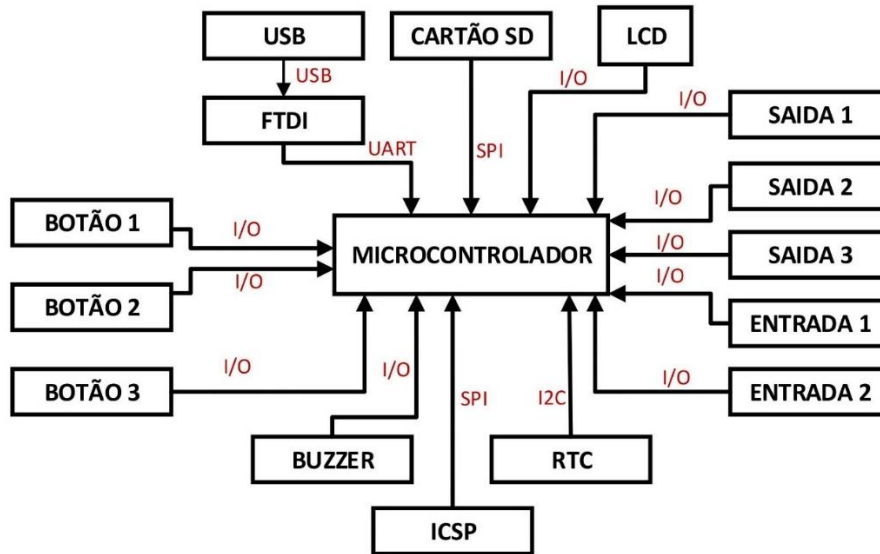


Figura 3.1 – Diagrama da arquitetura dos componentes para a placa eletrônica.

Para este projeto, os protocolos de comunicação necessários são o *Serial Peripheral Interface* (SPI), o *Inter-Integrated Circuit* (I<sup>2</sup>C), o USB e o *Universal Asynchronous Receiver/Transmitter* (UART).

O protocolo SPI é o utilizado por microcontroladores para comunicar com um ou mais periféricos rapidamente e em curtas distâncias. Este protocolo também pode ser utilizado entre dois microcontroladores. O funcionamento do protocolo consiste num *master* (geralmente o microcontrolador) que controla os dispositivos periféricos, os *slave*. Para a comunicação coexistir é necessário pelo menos em todos os dispositivos que se encontram a comunicar, as 3 linhas de sinal:

- *Master In Slave Out* (MISO) – Dados do *Master* para o *Slave*;
- *Master Out Slave In* (MOSI) – Dados do *Slave* para o *Master*;
- *Serial Clock* (SCK) – Relógio de sincronização para transmissão de dados entre o *Master* e o *Slave*;

Ainda existe a linha de sinal *Chip Select* (CS), que permite selecionar o *chip* [4].

O protocolo I<sup>2</sup>C, trabalha também com o modelo *master-slave*, podendo ter mais do que um *master*. Também este protocolo contém pinos próprios, o *Serial Data* (SDA) e o *Serial Clock* (SCL). O SDA é o pino que transfere os dados e os recebe (bi-direcional), enquanto que SCL serve para temporização entre os dispositivos, de forma que a comunicação realizada pelo pino SDA seja confiável [4].

O protocolo USB, é um protocolo simples que permite a transferência de dados entre computadores e dispositivos periféricos. Nos computadores é instalado o USB *host controller* enquanto que nos periféricos é instalado o USB *device controller*. A ligação é feita por cabo com conetores diferentes para evitar ligações indesejadas. A comunicação é realizada em modo série, ou seja, envia e recebe *bytes* de informação, 1 *bit* de cada vez. A interface série utiliza 4 linhas: a alimentação, a massa e um par de condutores torçados com tensão diferencial [31].

O protocolo UART, por ser uma comunicação assíncrona, funciona com dois pinos o de Transmissão (Tx) e o de Recepção (Rx). O Tx envia os *bits* para o pino Rx do outro dispositivo, que vai interpretar a comunicação *bit a bit*. Cada segmento de *bits* enviado contém 1 *bit* que indica o início da mensagem, 1 ou 2 *stop bits* para indicar o final da mensagem, 5 a 9 *bits* de informação e 1 *bit* de paridade para evitar erros [32].

Os restantes circuitos apresentados na Figura 3.1 funcionam como I/O.

Foi necessário utilizar um *Future Technology Devices International* (FTDI) para converter o protocolo USB em UART, para ser possível ao microcontrolador comunicar com este periférico.

### **3.3.2. Implementação da Arquitetura**

A arquitetura exposta em 3.3.1, foi selecionada segundo os requisitos anteriormente descritos em 3.2. A forma mais simples de verificar se a arquitetura selecionada é a adequada a este projeto, é desenvolver o código (*software*) a partir da implementação do *hardware*, utilizando uma placa de Arduino UNO e vários *shields*.

O Arduino genuíno UNO é um microcontrolador baseado no ATmega328P. Este é constituído por: 14 pinos de I/O digitais, dos quais 6 podem ser utilizados como saídas PWM; 6 entradas analógicas; um cristal de quartzo de 16 MHz; uma conexão USB; um conector de alimentação; um conector de pinos *In-Circuit Serial Programming* (ICSP) e um botão de reinicialização. Para a utilização deste microcontrolador, é necessário apenas conectá-lo a um computador com um cabo USB ou alimentá-lo com um adaptador CA-CC, ou ainda com uma bateria. O Arduino UNO, apresentado na Figura 3.2, foi o primeiro a existir e por isso o seu nome, pois UNO é a palavra italiana para 1 (primeiro) [4].



Figura 3.2 - Placa Arduino UNO [4].


Com o desenvolvimento do código, verificou-se a necessidade de mais memória, tendo por isso abandonado o Arduino UNO e optado pelo Arduino genuíno Mega 2560, apresentado na Figura 3.3. Este é um microcontrolador baseado no ATmega2560, sendo constituído por: 54 pinos de I/O digitais, dos quais 15 podem ser utilizados como saídas de PWM; 16 entradas analógicas; 4 UARTs (portas série de *hardware*); um oscilador de cristal de 16 MHz; uma conexão USB; um conector de alimentação; um conector de pinos ICSP e um botão de reinicialização. A sua alimentação é idêntica à do Arduino UNO e a maioria dos *shields* utilizados no UNO são compatíveis com o Mega 2560 [4].





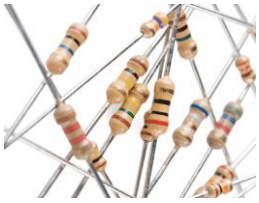




Figura 3.3 - Placa Arduino Mega 2560 [4].

Os *shields* necessários para a elaboração do *hardware* encontram-se apresentados na Tabela 3.8.

Tabela 3.8 – Dispositivos necessários para o desenvolvimento do *hardware*.

| Dispositivos   | Aplicação  |
|--|--|
|  <p>LCD</p> | <p>Por forma a ser possível visualizar pelo utilizador os estados das saídas, entradas, entre outras tarefas, é utilizado um <i>Liquid Crystal Display</i> (LCD)</p> |

|  |   |
|--|---|
| <p style="text-align: center;"><i>Can-Bus Shield</i></p>  | <p>Este dispositivo é muito útil para veículos, pois permite obter informações de temperatura de refrigeração, posição do acelerador, velocidade do veículo e do motor. Este <i>shield</i> permite armazenar estes dados e exibi-los com o auxílio de um LCD (contendo um conector para o LCD).</p> <p>No entanto, para este projeto este foi utilizado unicamente como suporte para o cartão SD, uma vez que a leitura dos horários entre outras configurações será por meio deste [33].</p> |
| <p style="text-align: center;">RTC</p>                    | <p>Para impor os horários da ERSE é necessário saber a hora, a data e o dia de semana, como tal é utilizado um <i>Real Time Clock</i> (RTC). Portanto, um RTC é um relógio que acompanha a hora atual e que pode ser programado para realizar ações num determinado momento [4].</p>  |
| <p style="text-align: center;">Botões</p>                | <p>São utilizados botões para permitir ao utilizador selecionar o horário que pretende, ou outras decisões necessárias. São também utilizados em representação de entradas.</p>   |
| <p style="text-align: center;">LEDs</p>                 | <p>Os <i>Light Emitting Diodes</i> (LEDs) representam as saídas, ou seja, o estado dos LEDs equivale ao ligar ou ao desligar da BC ou de escalões.</p>  |
| <p style="text-align: center;">Resistências</p>         | <p>São colocadas resistências nos botões e LEDs, de valores desde os 200 <math>\Omega</math> até a 1 k<math>\Omega</math>. A colocação destas resistências externas permite que os LEDs ou os botões funcionem sem se danificar ao final de algum tempo.</p> <p>Após a Tabela 3.8 encontra-se um exemplo para o cálculo de um valor de uma das resistências.</p>  |
| <p style="text-align: center;"><i>Buzzer</i></p>        | <p>Um dos requisitos referidos anteriormente é a indicação sonora da entrada que funciona como alarme, sendo para tal utilizado um <i>buzzer</i>. Este pode ser programado, selecionando a frequência e o tempo que este se encontra ativo.</p> <p>Um <i>buzzer</i> é um transdutor piezoelétrico, sendo que piezo é uma palavra de origem grega que significa pressionar, portanto efeito “piezoelétrico é a capacidade de alguns cristais gerarem</p>                                       |

|  |   |
|--|---|
|  | tensão elétrica por resposta a uma pressão mecânica”, no entanto o contrário também se verifica, ou seja, aplicar uma tensão no cristal, transformando energia elétrica em mecânica, produzindo som [34]. |
| <p style="text-align: center;">Cabos</p>  | <p style="text-align: center;">Para interligar os circuitos, são utilizados cabos selecionados propositalmente para eletrônica.</p>   |

Por forma a explicar a origem dos valores referidos na Tabela 3.8 para as resistências, será ilustrado um exemplo de cálculo. O dispositivo selecionado foi uma saída, que como ilustra a Figura 3.4 contém na entrada do *optocoupler* uma resistência de 1 k $\Omega$ .

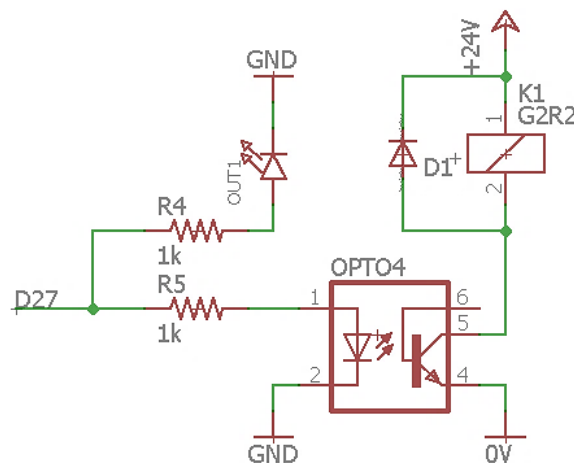


Figura 3.4 – Esquema de uma saída, no software Eagle [35].

O *pin* que representa a saída neste projeto será ligado a um relé, como tal é utilizado um *optocoupler*. Um *optocoupler* é um acoplador ótico que utiliza dispositivos semicondutores para transmissão ótica para transferir um sinal elétrico entre circuitos ou elementos de um circuito, mantendo-os isolados eletricamente um do outro. É muito utilizado na interface de componentes sensíveis (como o Arduino) e circuitos de maior tensão ou corrente (como os relés).

O *optocoupler* funciona a uma tensão de 5 V e 60 mA, como é possível verificar no catálogo que se encontra no Anexo III. Portanto, segundo a lei de ohm:

$$R = \frac{U}{I} = \frac{5}{60 * 10^{-3}} \cong 83,33 \Omega \quad (3.1.)$$

Para poupar a vida útil do equipamento (*Mean Time Between Failures*, MTBF), decidiu-se operar a 5 mA. Como tal o valor da resistência teórica é de 1k $\Omega$ .

Para as restantes resistências o cálculo é semelhante, no entanto é importante referir que existem valores típicos no mercado para as resistências e que foi optado por uniformizar o valor das mesmas, para ser possível comprar várias resistências com o mesmo valor, otimizando desta forma ligeiramente o custo.

Com já referido, foi essencial adquirir mais espaço de memória, por isso foi substituída a placa Arduino UNO pela placa Arduino Mega, pois esta placa é muito semelhante à anterior, sendo por isso fácil a sua substituição. A Figura 3.5 apresenta o sistema já com a substituição.

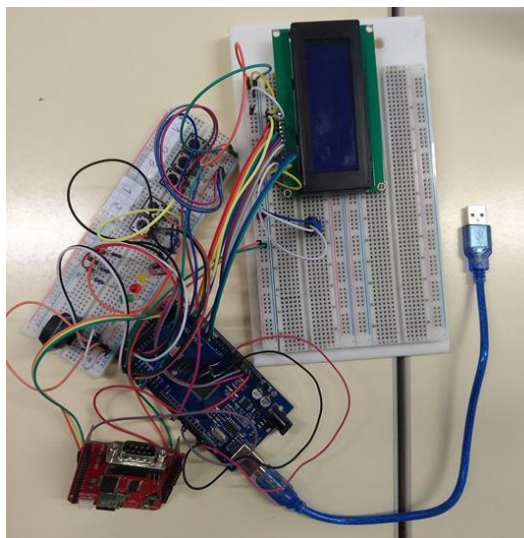


Figura 3.5 – Protótipo do sistema para desenvolvimento do *hardware*.

Como se pode verificar pela Figura 3.5, este sistema não é muito prático, como solução a adotar no interior do armário das BC. Além de que, facilmente existem erros de contacto e por esse motivo não foi verificado o seu funcionamento no interior do armário das BC. Em vez disso, foi idealizada e projetada uma placa eletrónica com os mesmos componentes anteriores, ou seja, um PCB.

### 3.3.3. Conceção do PCB

É importante esclarecer alguns conceitos na elaboração de um PCB, antes de explicar o método utilizado para o conceber. Os PCB são constituídos por camadas (*layers*), por exemplo um PCB que contenha 4 *layers*, significa que o PCB contém caminhos/trilhas gravados nas faces superior e inferior, e possui ainda mais duas camadas metálicas intermédias, onde existem igualmente caminhos gravados. Regra geral, as *layers* intermédias são usadas para alimentação elétrica dos componentes, noutros casos são utilizadas para serem ligadas à massa do circuito, para controlar as emissões de ruído eletromagnético.

Os caminhos existentes nas diferentes camadas são interligados através de orifícios com o interior em revestimento metálico, através de um processo eletroquímico. Estes orifícios metálicos são denominados de vias. As vias podem também servir para fixação e soldagem de terminais de componentes [36].

Para conceber um PCB é necessário escolher um *software* que permita esboçar o PCB e criar os ficheiros necessários para enviar a um fabricante para o criar. Assim foi optado por utilizar o *software Eagle CAD*. O PCB criado contém duas *layers* e foi realizado segundo 6 etapas:

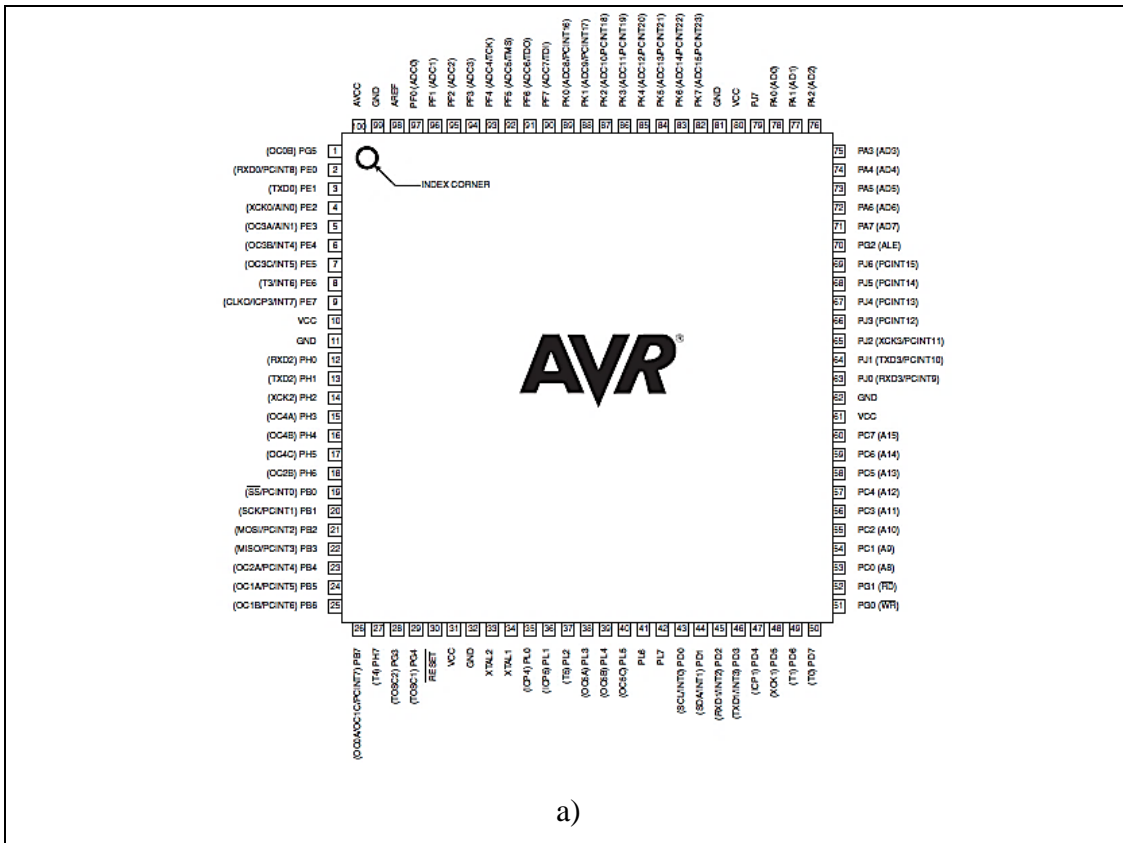
➤ Etapa 1: Esquemas

Esta etapa consiste em desenhar o circuito que vai integrar o PCB, considerando a arquitetura estabelecida anteriormente. Selecionando para isso os dispositivos necessários ao projeto e simultaneamente verificar se existem no mercado. A maioria dos dispositivos externos utilizados na plataforma Arduino, já contém o *footprint* do dispositivo disponível para o *Eagle*. Os dispositivos em que não são fornecidos os desenhos do equipamento para este *software*, terão que ser desenhados. Para isso é necessário visualizar o catálogo e utilizar as funcionalidades do *Eagle* para criar o esquema desse dispositivo.

De seguida, serão ilustrados três exemplos de dispositivos, com o intuito de demonstrar o processo no esboço de um esquema e ao mesmo tempo é aproveitado e explicado ao pormenor a importância destes dispositivos neste projeto. Assim sendo foi selecionado um conversor DC-DC como um exemplo, onde foi necessário criar o esquema; o microcontrolador utilizado pelo Arduino Mega e o RTC. Os esquemas destes dois últimos dispositivos foram retirados online.



O segundo exemplo selecionado é o microcontrolador do Arduino Mega, apresentado na Figura 3.7. Este foi selecionado por preceder de uma vantagem enorme para este projeto, para além do espaço de memória e a quantidade de *pins* disponíveis, o microcontrolador é bastante semelhante ao do Arduino mais utilizado por qualquer utilizador, o Arduino UNO. O Arduino UNO é o primeiro Arduino que se aconselha sempre para começar a trabalhar e é suficiente para a maioria dos projetos.



a)

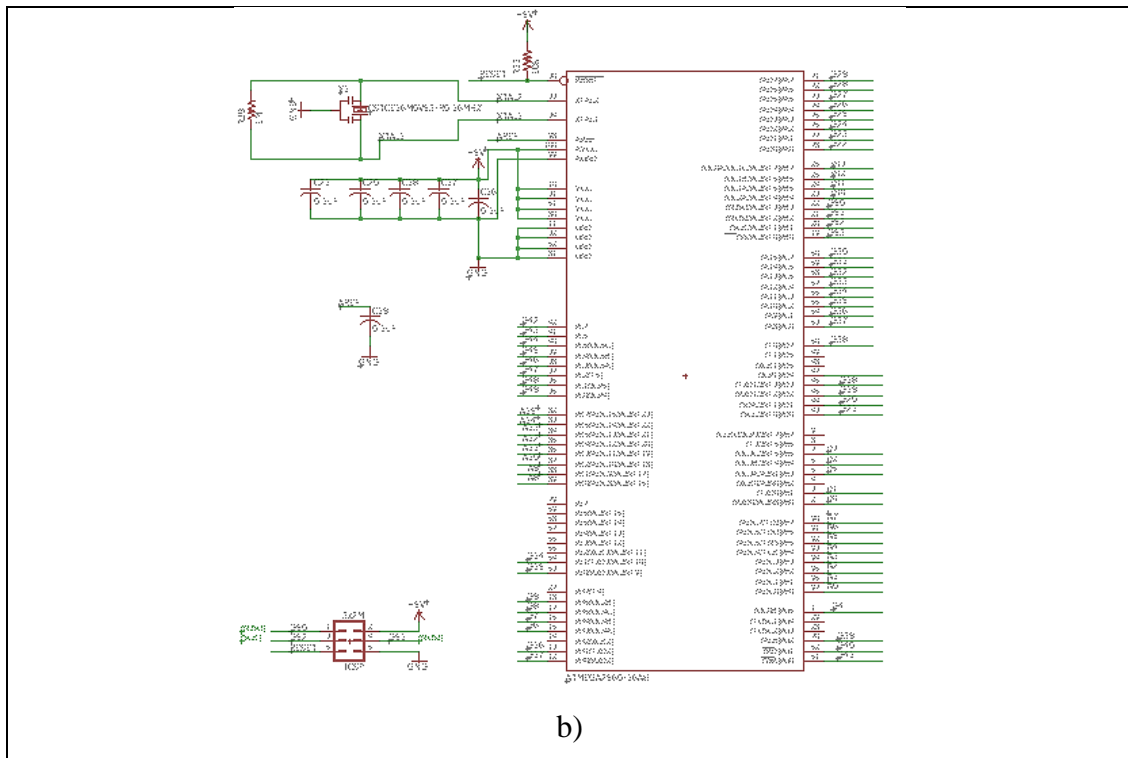


Figura 3.7 – a) Catálogo do microcontrolador, b) Esquema do microcontrolador no *Eagle* [35].

O esquema do microcontrolador foi retirado da plataforma Arduino [4], sendo importante realçar alguns aspectos, tais como:

- Em todos os *chips* é colocado um condensador junto de cada pino de alimentação, portanto no *chip* do microcontrolador não é exceção, assim como existem 5 pinos de alimentação, existem 5 condensadores, como ilustra a Figura 3.8. Estes condensadores têm o funcionamento de *bypass*. O que significa que servem para reduzir as frequências altas de corrente existentes num caminho com altas impedâncias, desviando-o para outro caminho utilizando um condensador. Ou seja, servem para ignorar o ruído da corrente existente muitas vezes nas alimentações em circuitos eletrônicos, impedindo distorções no barramento de tensão. Os condensadores de *bypass* são geralmente dimensionados por convenção de valores típicos, como  $1\mu\text{F}$  e  $0,1\mu\text{F}$  [37], [38];

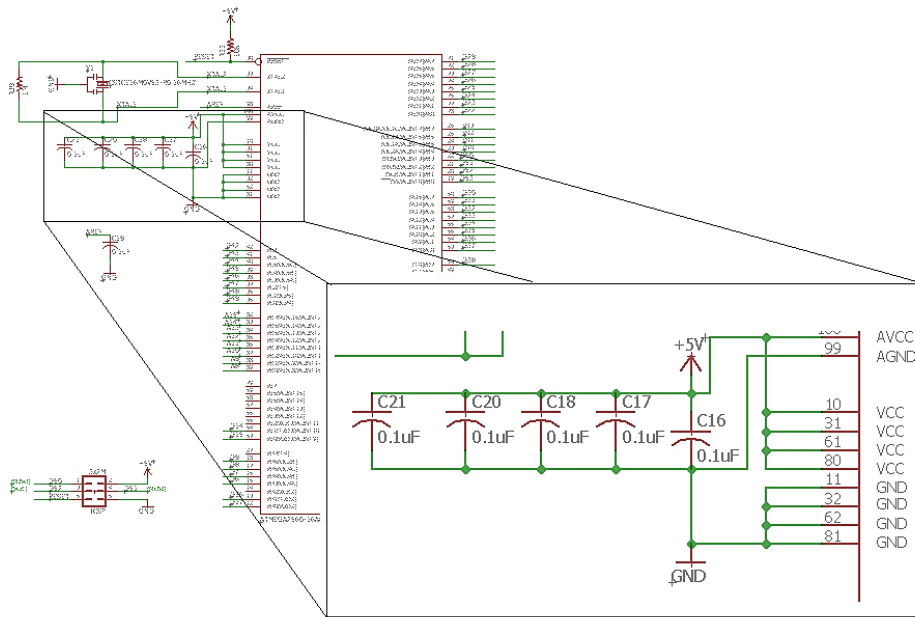


Figura 3.8 – Os condensadores de *bypass* no esquema do microcontrolador no *Eagle* [35].

- Em qualquer dispositivo de lógica digital (eletrônica) é costume colocar-se em todas as entradas (com a exceção daquelas que já o têm internamente), resistências *pull-up*. As resistências *pull-up*, são resistências que têm como função garantir que as entradas se encontram em níveis lógicos esperados, isto é, no nível lógico 1, portanto o conceito desta resistência é “puxar” (*pull*) a tensão do condutor para o valor de tensão a que ele está conectado. Logo, sem esta resistência, a leitura do estado lógico desta entrada seria flutuante, ou seja, ora o pino se encontrava num estado lógico alto ou num estado lógico baixo [33]. No caso do microcontrolador temos uma resistência de *pull-up* no pino correspondente ao *reset*, como ilustra a Figura 3.9;

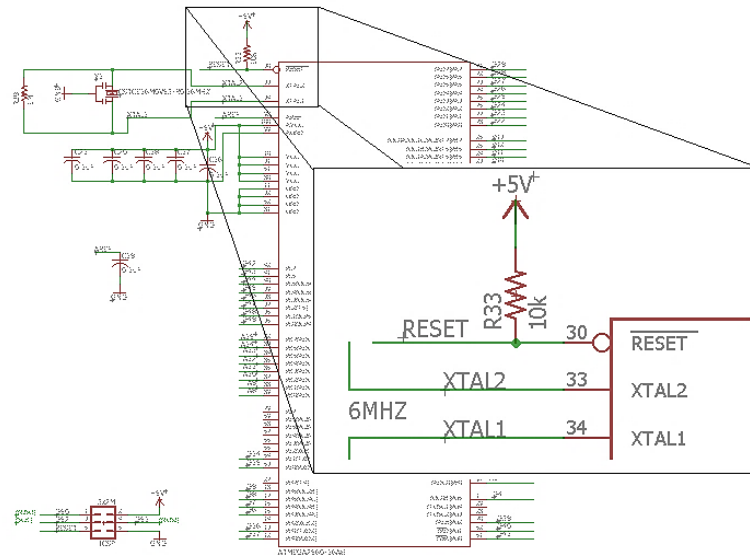


Figura 3.9 – Localização da resistência de *pull-up* no esquema do microcontrolador no *Eagle* [35].

- Pela Figura 3.10 é possível verificar que existe um componente no canto inferior esquerdo. Este componente é o ICSP, que é uma ficha de 6 pinos que permite o descarregamento do sistema operativo (*Bootloader*). Deste modo tornamos a placa eletrónica compatível com a plataforma Arduino utilizado um microcontrolador da Atmel [4], [33].

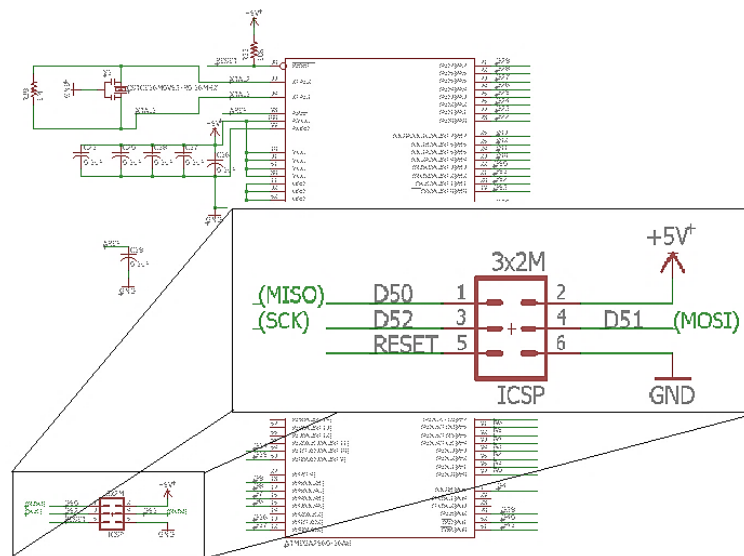


Figura 3.10 – Localização do ICSP no esquema do microcontrolador no *Eagle* [35].

O principal intuito da solução desenvolvida é incorporar os horários da ERSE, como tal, a base deste projeto é saber cada instante de tempo, à vista disso foi selecionado como

terceiro e último exemplo o RTC, apresentado na Figura 3.11. Como a fidelidade temporal é muito importante foi escolhido utilizar um DS3231 e não apenas o relógio interno existente no microcontrolador, pois o cristal existente no microcontrolador é menos preciso e seria necessário um circuito à parte com uma bateria externa para guardar instantes de tempo.

O DS3231 é um RTC extremamente preciso, que utiliza o protocolo I<sup>2</sup>C, além disso, vem com a compensação de temperatura integrada, com um oscilador de cristal (*Temperature Compensated Crystal Oscillator*, TCXO), uma entrada de bateria, o que permite guardar o tempo quando a energia principal é interrompida. Contém também um ressonador de cristal que permite aumentar a precisão do dispositivo a longo prazo. O RTC consegue guardar informação dos segundos, minutos, horas, dias, dia da semana, mês e ano. Para além disso, ajusta automaticamente os meses com menos de 31 dias, incluindo correções para anos bissextos.

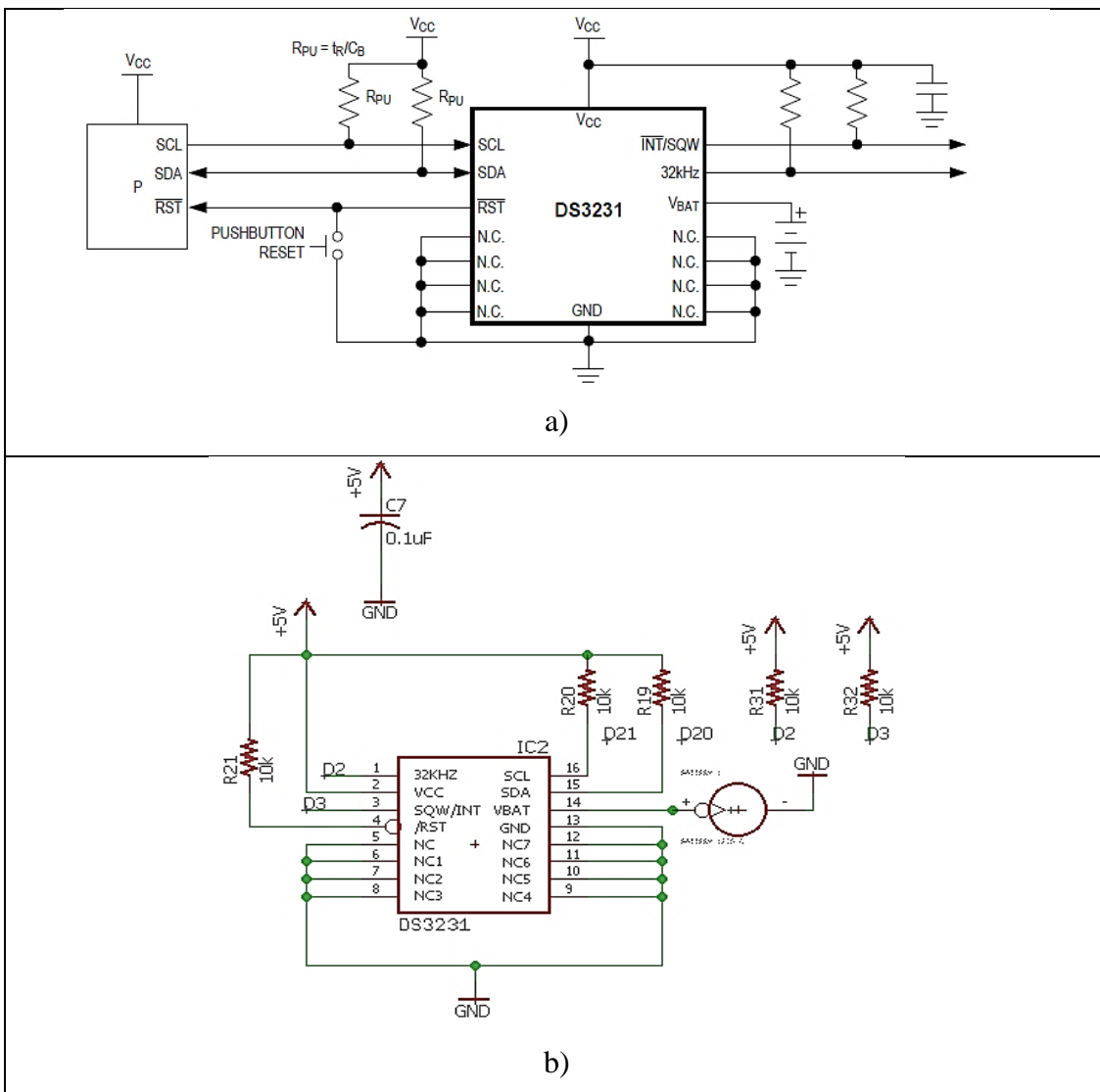


Figura 3.11 – a) Catálogo do RTC, b) Esquema do RTC no Eagle [35].

Todos os esquemas encontram-se no Apêndice I, sendo o princípio de desenvolvimento semelhante aos anteriores exemplos. Apenas é de realçar que são utilizados filtros passa-baixo na entrada dos botões, por forma a permitir a passagem apenas das frequências baixas e atenuar as frequências de amplitude superior à Frequência de Corte (FC). O valor da FC foi selecionado, de modo a que o seu valor fosse superior ao valor da frequência que se espera na entrada do botão, mas simultaneamente de valor inferior à menor frequência utilizada em todo o sistema. A frequência mais baixa presente no sistema é a do PWM, que tem o valor padrão de 490 Hz [4], [39].

A expressão utilizada para o cálculo desta frequência é a seguinte:

$$FC = \frac{1}{2\pi RC} \quad (3.2.)$$

Em que R, representa o valor da resistência e C o valor do condensador. Por exemplo, no botão S1, apresentado na Figura 3.12 e utilizando a equação (3.2.), obtém-se:

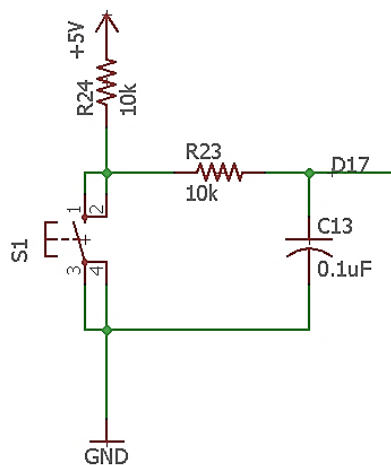


Figura 3.12 – Esquema do botão S1, no *software Eagle* [35].

$$FC = \frac{1}{2\pi R_{23} C_{13}} = \frac{1}{2\pi * 10 * 10^3 * 0,1 * 10^{-6}} \cong 159 \text{ Hz}$$

Assim, o valor utilizado para a FC foi de aproximadamente 159 Hz.

➤ Etapa 2: *Build of Material* (BOM)

A segunda etapa é a BOM, que consiste numa lista do material utilizado no esquema. Esta lista é gerada automaticamente pelo *software*, tendo sido acrescentada a esta lista o fabricante e a referência de cada produto.

Esta etapa é fundamental para saber se os dispositivos selecionados existem nos fornecedores de eletrónica comuns, pois apesar de um dispositivo ter disponível o esquema online, não significa que o mesmo exista no mercado. Selecionando os fornecedores, é possível estimar o custo do PCB com os componentes e por sua vez elaborar um orçamento.

A BOM, ou seja, a lista com a quantidade dos dispositivos, fornecedores e referências encontra-se no Apêndice I. É de realçar que nesta não consta os valores em euros por questões de sigilo com a empresa envolvente.

➤ Etapa 3: *Layout*

A Terceira etapa é uma das mais importantes, se não a mais importante, pois um bom *layout* é cerca de 90% responsável por uma boa concretização do PCB. O *layout* consiste na colocação dos dispositivos numa área restrita, que têm as dimensões desejadas para o PCB. No interior dos limites do PCB os dispositivos são colocados consoante o gosto do *designer*, sendo por isso possível vários *layouts* com os mesmos componentes. Certamente na existência de ligações entre os dispositivos, estes deverão ser dispostos na proximidade uns dos outros, no entanto mesmo assim é possível criar diferentes *layouts*.

O *layout* concebido para este projeto encontra-se no Apêndice I, no entanto será feita uma breve descrição da abordagem empregue para realizar o *layout*. Em primeiro lugar, havendo vários níveis de tensão, foi optado por dividir o PCB nesses mesmos níveis de tensão como ilustra a Figura 3.13.

Segundo a norma IPC-2221 [39], existe distâncias mínimas entre os níveis de tensão (Tabela 6.1 da norma). Portanto entre o nível de tensão 230/400 VAC e os 24 VDC deve existir uma distância de pelo menos 0,8 mm, entre o nível de tensão 24 VDC e 5 VDC deve existir pelo menos 0,05 mm, como ilustra a Figura 3.13.

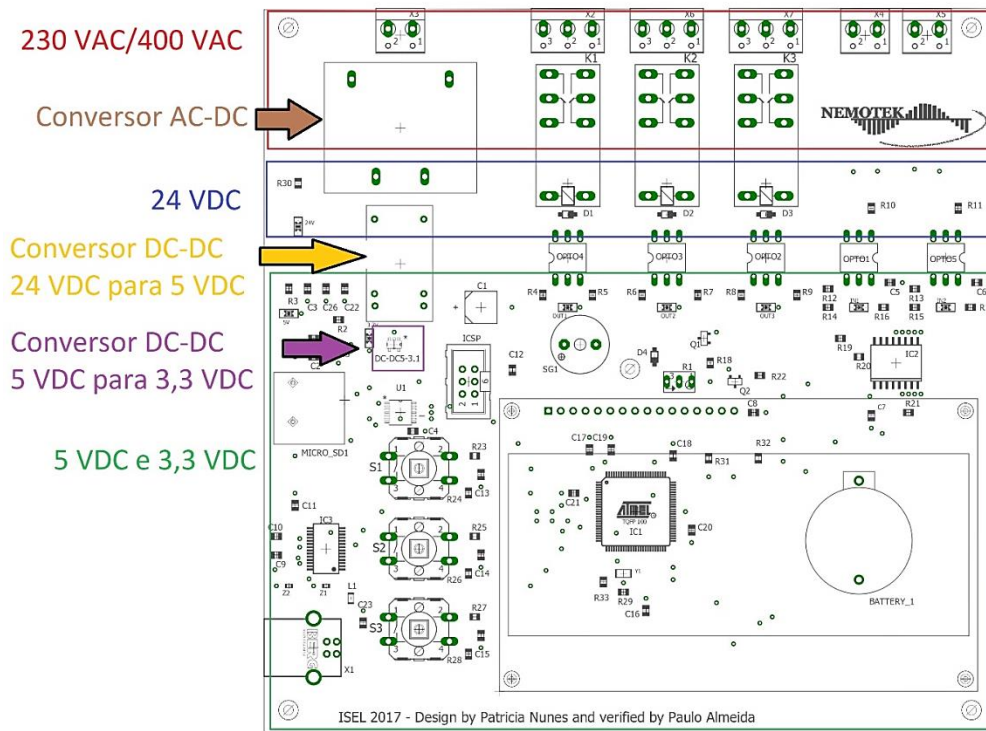


Figura 3.13 – Localização dos diferentes níveis de tensão no *design* do PCB [35].

Foi decidido colocar o LCD num canto, por forma a colocar os 3 botões perto deste. Os dispositivos restantes, foram colocados da forma mais prática possível. Na Figura 3.14 apresenta-se realçado dois dos três componentes que serviram de exemplo na etapa 1, o microcontrolador e o RTC, devido à importância dos mesmos.

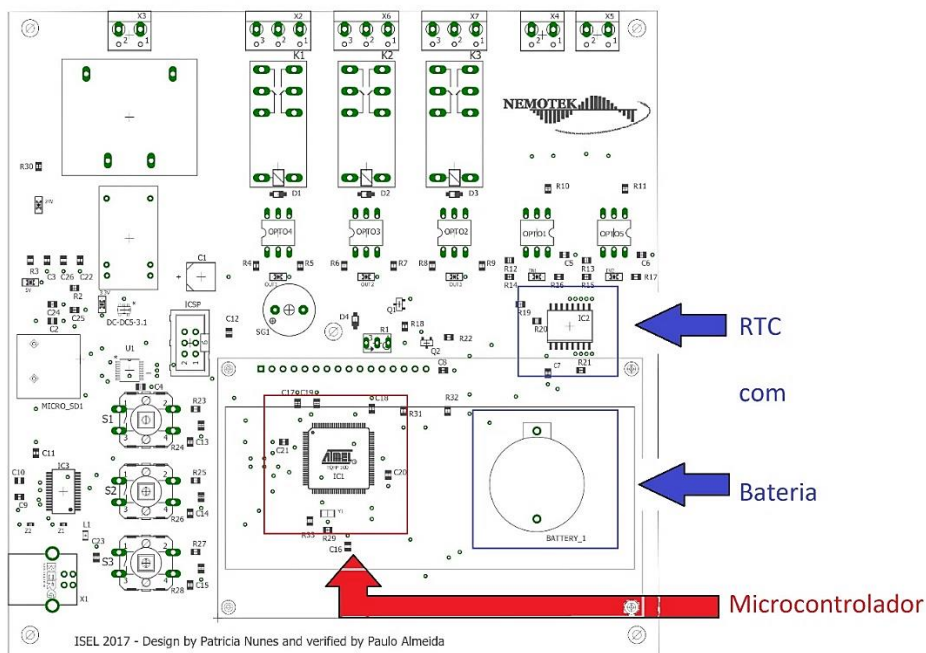


Figura 3.14 – Localização do microcontrolador e do RTC com a bateria no *design* do PCB [35].

Tal se pode verificar pela Figura 3.14, o microcontrolador foi colocado o mais central possível dentro da zona do nível de tensão de 5 VDC/3,3 VDC, pois este encontra-se ligado a vários dispositivos, tornando a tarefa da etapa seguinte mais fácil, como se irá verificar.

O RTC necessita de uma bateria externa e como se pode verificar a bateria contém dimensões consideráveis, sendo por isso optado colocá-la num espaço estratégico e isolado dos restantes dispositivos, uma vez que não é utilizada por estes. Por sua vez, o RTC é colocado perto da bateria.

O *layout* encontra-se mais perceptível no Apêndice I.

➤ Etapa 4: Routing

Uma vez definido o *layout* do PCB, é necessário realizar as ligações entre os dispositivos, concretizadas no esquema, construindo desta forma os circuitos, como ilustra a Figura 3.15. Em primeiro lugar deve ser realizado o perímetro da camada que representa as tensões positivas (230/400 VAC, 24 VDC, 5 VDC), sendo esta a camada superior e as respetivas massas de cada nível de tensão, representadas na camada inferior. Estas camadas encontram-se isoladas galvanicamente, prevenindo desta forma a circulação de correntes sobre impedâncias comuns nos vários circuitos, incluindo a massa. Se a massa fosse comum aos vários circuitos, apesar de teoricamente esta ser equipotencial, a sua impedância não é nula, entre dois quaisquer pontos de referência, podendo surgir diferenças de potencial resultantes da circulação de corrente [40].

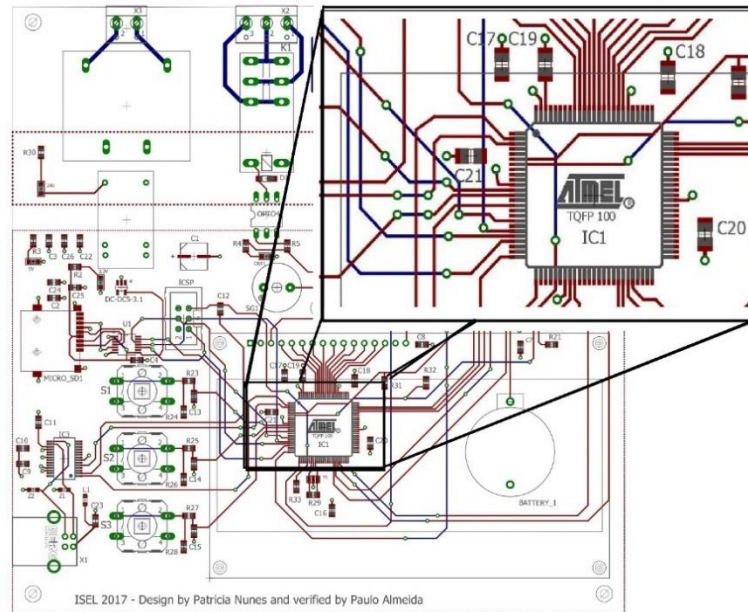


Figura 3.15 – Ilustração da ligação entre os pinos, onde vermelho representa a *layer* de cima e a azul a *layer* de baixo do PCB [35].

Como se verifica na Figura 3.15, a ligação entre os pinos inicia sempre com a *layer* vermelha se o componente estiver instalado nesta, e na necessidade de colocar o circuito na parte de trás do PCB para não sobrepor ligações, é alterado para a *layer* azul. Na interligação encontra-se um furo metalizado (via), que permite a condução entre o traçado de cima e o traçado de baixo, considerando apenas estas duas camadas. Não pode haver sobreposição de ligações, pois como estas são condutoras e fazem parte de circuitos diferentes, sobrepô-las ia alterar os circuitos definidos previamente.

Os caminhos existentes nas *layers* podem preceder de diferentes larguras. Neste projeto foram utilizadas apenas duas larguras, 40 mil (Milésimas de Polegada) no nível de 230/400 VAC e 12 mil para o nível de tensão 24 VDC e 5 VDC. A corrente máxima dos caminhos é estabelecida segundo a largura das próprias pistas, conforme indica a norma IPC-2221 [39]. O *software Eagle* também calcula a corrente máxima em cada circuito, estes valores encontram-se apresentados no Anexo IV.

Novamente o esquema do *Routing* encontra-se mais visível no Apêndice I.

#### ➤ Etapa 5: Design Rules Check (DRC)

Tendo o *design* e a ligação dos circuitos realizados, é possível transitar para a etapa seguinte, o DRC. O DRC consiste em verificar se o PCB é possível de ser concebido. Para

isso é necessário descarregar um ficheiro do fabricante selecionado, que indica os requisitos e limitações, pois cada fabricante contém os seus próprios requisitos e limitações diferentes na produção dos seus PCB. O fabricante selecionado para este projeto foi a ITEAD e como tal foi descarregado para o *software Eagle* os requisitos deste fabricante para testar se o PCB esboçado os cumpre. Uma grande vantagem do *software Eagle* é indicar o local exato onde não está a ser cumprido um requisito e dizer qual é, por forma a que possa ser alterado corretamente. Na inexistência de requisitos a serem comprometidos, é possível passar para a fase seguinte. Os requisitos do fabricante em questão encontram-se descritos no Apêndice I.

➤ Etapa 6: Gerbers

A última etapa consiste na criação dos *Gerbers*, ou seja, dos ficheiros que o fabricante pede para fabricar o PCB. Esses ficheiros são:

- *Gerber Bottom Copper* (GBL), que é o ficheiro da *layer* de baixo, que indica a superfície que está revestida de cobre;
- *Gerber Top Copper* (GTL), que é o ficheiro da *layer* de cima, que indica a superfície que está revestida de cobre;
- *Gerber Bottom Solder Mask* (GBS), que é o ficheiro da *layer* de baixo, com a máscara de solda (normalmente de cor verde) que geralmente cobre toda a superfície, com a exceção as áreas onde vai ser aplicado solda;
- *Gerber Top Solder Mask* (GTS), que é o ficheiro da *layer* de cima, com a máscara de solda (normalmente de cor verde) que geralmente cobre toda a superfície, com a exceção as áreas onde vai ser aplicado solda;
- *Gerber Bottom Silk-screen* (GBO), que é o ficheiro da *layer* de baixo, que contém os contornos dos componentes e sua designação (resistências (R1), condensadores (C1)), e texto livre;
- *Gerber Top Silk-screen* (GTO), que é o ficheiro da *layer* de cima, que contém os contornos dos componentes e sua designação (resistências (R1), condensadores (C1)), e texto livre;
- O documento com a extensão txt, indica as coordenadas onde existe furações e as suas dimensões.

É possível visualizar cada um destes ficheiros nos simuladores existentes online, desta forma consegue-se ter uma perceção de como vai ficar o PCB na realidade. As simulações dos ficheiros encontram-se no Apêndice I.

Como é possível constatar pela explicação sobre os passos necessários ao desenvolvimento do PCB, foi tido em conta as interferências internas existentes, sendo as soluções apresentadas como isolamento galvânico, filtros passa-baixo, resistências de *pull-up* e condensadores de *bypass*.

### **3.4. Solução Desenvolvida, Software**

Para desenvolver o algoritmo para o equipamento teve-se em conta os requisitos de controlo referidos anteriormente neste Capítulo. Como já foi referido, o objetivo do equipamento desenvolvido é controlar o ligar e o desligar de uma BC, (a partir da alimentação do RV) e/ou colocar em serviço um escalão (um condensador), de acordo com um conjunto de horários pré-estabelecidos (ERSE).

O algoritmo desenvolvido recorre a um microcontrolador Arduino (linguagem C/C++), e está organizado segundo várias rotinas, ou seja, por frações de código que realizam tarefas específicas. Deste modo é possível desenvolver o código de forma clara e objetiva. Portanto, o algoritmo opera segundo o formato estabelecido no Arduino, ou seja, a estrutura do código é desenvolvida por um programa principal, que tem a função de chamar as várias rotinas pela ordem correta, uma secção de declaração de variáveis e uma secção para as bibliotecas utilizadas.

Uma parte essencial para uma boa execução do algoritmo a utilizar neste projeto, é a estratégia a adotar como interface entre o utilizador e o equipamento, de modo a que este consiga seleccionar o horário que pretende entre outras características. Tendo em conta o funcionamento base do Arduino e as suas limitações, foi decidido utilizar um cartão SD como armazenamento dos dados provenientes das opções do utilizador. Portanto no interior do cartão SD existe um documento em *Excel* que permite ao utilizador escolher o que pretende para o equipamento, seguidamente este documento em *Excel* cria os documentos em formato .txt (texto) necessários para a configuração do equipamento. O utilizador tem também a opção de configurar o equipamento, pela manipulação dos botões disponíveis no equipamento, como será explicado posteriormente.

Uma vez concluído o algoritmo (*software*) e a estrutura física do equipamento (*hardware*), foram realizadas experiências no laboratório de Máquinas Elétricas do estabelecimento de ensino, Instituto Superior de Engenharia de Lisboa (ISEL), a partir de uma

BC fornecida pela NEMOTEK. Os resultados destas experiências encontram-se descritos posteriormente no Capítulo 4 , onde serão mencionados as dificuldades encontradas e o sucesso do equipamento para os objetivos pretendidos neste projeto.

### 3.4.1. Interface entre o utilizador e o equipamento

O equipamento desenvolvido precede de condições necessárias para o seu funcionamento, condições essas que são decisões tomadas pelo utilizador. Por forma a facilitar a interação do utilizador com o equipamento foi desenvolvido duas formas de comunicação com o dispositivo.

O utilizador pode estabelecer as suas decisões a partir da manipulação dos botões existentes no equipamento, ou a partir de um cartão SD, que contém armazenado os resultados, a partir de um documento *Excel*, que está programado para, mediante as decisões escolhidas, criar vários documentos em formato de texto para serem lidos pelo equipamento. Este documento em *Excel*, é constituído por 3 folhas, uma que diz respeito à digitação dos feriados existentes ao longo de vários anos do local onde será inserido o equipamento, outra sobre a configuração que o utilizador opta para as saídas, entradas e horário a estabelecer e por fim a outra folha que contém inserido os horários da ERSE.

A primeira folha tem o aspeto da Figura 3.16.

|    | A                   | B     | C           | D |
|----|---------------------|-------|-------------|---|
| 1  | 01/01/1970          | Dias  | Segundos    |   |
| 2  | 01/01/2017 00:00:00 | 17167 | ;1483228800 |   |
| 3  | 28/02/2017 00:00:00 | 17225 | ;1488240000 |   |
| 4  | 14/04/2017 00:00:00 | 17270 | ;1492128000 |   |
| 5  | 16/04/2017 00:00:00 | 17272 | ;1492300800 |   |
| 6  | 25/04/2017 00:00:00 | 17281 | ;1493078400 |   |
| 7  | 01/05/2017 00:00:00 | 17287 | ;1493596800 |   |
| 8  | 10/06/2017 00:00:00 | 17327 | ;1497052800 |   |
| 9  | 13/06/2017 00:00:00 | 17330 | ;1497312000 |   |
| 10 | 15/06/2017 00:00:00 | 17332 | ;1497484800 |   |
| 11 | 15/08/2017 00:00:00 | 17393 | ;1502755200 |   |
| 12 | 05/10/2017 00:00:00 | 17444 | ;1507161600 |   |

Figura 3.16 – Folha do *Excel* para a inserção dos Feridos.

O utilizador digita na coluna A, quantos feriados quiser, o limite encontra-se em 500, mas pode ser aumentado. Os feriados são declarados com a data e hora, a hora digitada é

sempre às 0 horas, 0 minutos e 0 segundos, para efeitos de programação. A coluna C converte automaticamente a data para segundos, valor este que vai ser lido pelo equipamento.

A segunda folha denominada de configuração representa as decisões que terão que ser tomadas pelo utilizador. É importante que o utilizador tenha conhecimento de que terá que seleccionar a região do país onde será inserida a BC, o tarifário com que pretende que o equipamento funcione, ter a consciência da existência de duas entradas e que estas podem funcionar em dois modos diferentes (sendo que cada entrada só pode funcionar num único modo), como alarme ou como controlo.

Se funcionar como alarme, significa que se a entrada não estiver ativa, a saída está a funcionar normalmente, com as funções seleccionadas pelo utilizador, no momento em que a entrada esteja ativa, todas as saídas são desligadas e é emitido um sinal sonoro. Se funcionar como controlo, nesse caso o utilizador pode optar que a saída funcione segundo essa entrada apenas, que funcione segundo o horário da ERSE e a entrada, ou apenas pelo horário.

O utilizador necessita também saber que o equipamento está projetado para o controlo, no máximo, de 3 saídas ativas.

O aspeto desta folha de *Excel* será apresentado no subcapítulo do Exemplo.

A terceira e última folha, como foi referido, ilustra os horários da ERSE. Esta folha contém um botão no canto superior esquerdo, que deve ser carregado, quando o utilizador terminar todas as alterações no documento em *Excel*. Este botão, que se encontra na Figura 3.17, vai permitir gerar os ficheiros de texto para o equipamento ler.

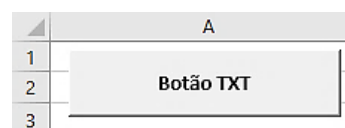


Figura 3.17 – Botão que gera os ficheiros txt.

Esta folha respeita os horários ilustrados no subcapítulo 3.1, como tal, encontra-se dividida em período de hora legal de inverno e período de hora legal de verão, e ainda subdividida em regiões e tarifários. Sempre que haja uma alteração nos horários pela entidade ERSE, é possível de uma forma simples alterar a folha de *Excel*, como representa a Figura 3.18.

|    | A                                | B                    | D            | F                  | H                  |
|----|----------------------------------|----------------------|--------------|--------------------|--------------------|
| 1  |                                  |                      |              |                    |                    |
| 2  | Botão TXT                        | Portugal Continental |              |                    |                    |
| 3  |                                  | Ciclo Semanal        |              |                    | Ciclo Semanal      |
| 4  | Período de hora legal de Inverno | De segunda a sexta   | Sábado       | Domingo e Feriados | De segunda a sexta |
| 5  | 00:00 h                          | Vazio Normal         | Vazio Normal | Vazio Normal       | Cheias             |
| 6  | 00:15 h                          | Vazio Normal         | Vazio Normal | Vazio Normal       | Cheias             |
| 7  | 00:30 h                          | Vazio Normal         | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 8  | 00:45 h                          | Vazio Normal         | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 9  | 01:00 h                          | Ponta                | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 10 | 01:15 h                          | Cheias               | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 11 | 01:30 h                          | Vazio Normal         | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 12 | 01:45 h                          | Super Vazio          | Vazio Normal | Vazio Normal       | Vazio Normal       |
| 13 | 02:00 h                          | Vazio Normal         | Super Vazio  | Super Vazio        | Super Vazio        |
| 14 | 02:15 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 15 | 02:30 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 16 | 02:45 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 17 | 03:00 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 18 | 03:15 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 19 | 03:30 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 20 | 03:45 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 21 | 04:00 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 22 | 04:15 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 23 | 04:30 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 24 | 04:45 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |
| 25 | 05:00 h                          | Super Vazio          | Super Vazio  | Super Vazio        | Super Vazio        |

Figura 3.18 – Folha do *Excel* que ilustra os horários da ERSE.

Como se pode verificar, carregando em qualquer uma das células, aparece uma janela que mostra as opções existentes e permite selecionar a opção correta para aquele local. Cada período está representado com uma cor, sendo desta forma mais visual e fácil de detetar algum erro cometido. Portanto o período Vazio Normal encontra-se a castanho, o período Ponta a amarelo, o período Cheias a bege e o período Super Vazio a amarelo torrado.

### 3.4.1.1. Exemplo

Será realizado um exemplo de como o utilizador deve proceder. Este pode realizar a tarefa de duas formas, manipulando os botões do equipamento ou através do *Excel*, por forma a tomar as decisões corretas. Para explicar o processo na tomada de decisão do utilizador, foi elaborado um Diagrama de Transição de Estados (DTE). Um DTE é uma representação do comportamento de um sistema, onde são retratadas as várias situações/estados no decorrer da execução de processos de um sistema, convencionalmente, em cada instante, um e um só estado está ativo, em que estado representa um conjunto de variáveis [41].

Existem 3 saídas, no entanto como estas não são distintas entre si, o DTE, apresenta a saída  $n$ , em que  $n$ , pode tomar o valor 1, 2 e 3.

O exemplo evidenciado é apenas para a saída 1, porém, o processo de seleção é semelhante para as outras duas saídas. O que se pode constatar pelo DTE apresentado na Figura 3.19, é que o que condiciona o funcionamento do sistema é a seleção do modo como as entradas se encontram a funcionar.

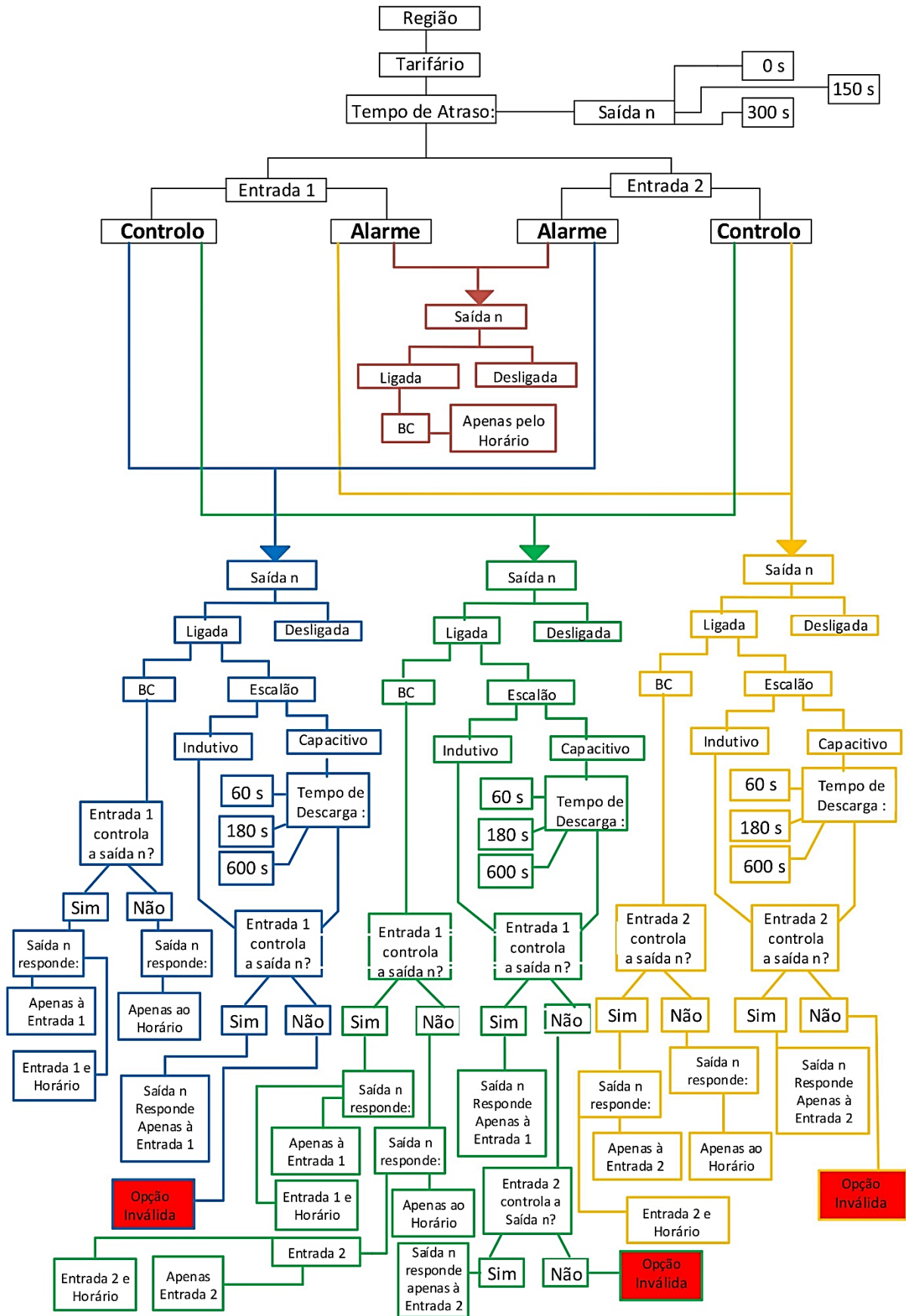


Figura 3.19 – Diagrama de transição de estados, onde n representa o número de saídas existentes.

Exemplo para a saída 1, Excel:

Para configurar o equipamento, a partir do documento *Excel*, o utilizador recorre à página de *Excel* que tem o aspeto da Figura 3.20.

|            | Portugal                                | Açores                           | Madeira                              |
|------------|---|----------------------------------|--------------------------------------|
| Tarifários | Ciclo Semanal                           | Ciclo Diário                     | Ciclo Diário                         |
|            | Ciclo Semanal Opcional (em MAT,AT e MT) | Ciclo Diário Opcional (MT e BTE) | Ciclo Diário Opcional (AT, MT e BTE) |
|            | Ciclo Diário BTE e BTN                  |                                  |                                      |
|            |   | Selecionar:                      |                                      |
|            | Região:                                 | Portugal                         |                                      |
|            | Portugal                                |                                  |                                      |
|            | Açores                                  |                                  |                                      |
|            | Madeira                                 |                                  |                                      |
|            |   | Selecionar:                      |                                      |
|            | Tarifário:                              | Ciclo Semanal                    |                                      |
|            | ver parte superior da folha             |                                  |                                      |

Figura 3.20 – Folha do *Excel* que representa a configuração, Seleção das opções Região e Tarifário.

No início da folha é evidenciado os tarifários existentes para cada região e no restante da folha estão colocadas as opções a tomar, como ilustra a Figura 3.20. As células a bege, são as utilizadas para selecionar uma das opções evidenciadas na coluna à esquerda desta, portanto basta dar um clique e as opções possíveis aparecem, que correspondem às enumeradas na coluna ao lado. Vamos supor que o usuário seleciona as opções da Figura 3.21.

|                      |             |
|----------------------|-------------|
|                      | Selecionar: |
| Entrada 1:           | Controlo    |
| Alarme               | Alarme      |
| Controlo             | Controlo    |
|                      | Selecionar: |
| Entrada 2:           | Controlo    |
| Alarme               |             |
| Controlo             |             |
|                      | Selecionar: |
| Tempo para a Saída 1 | 0 s         |
| 0 s                  |             |
| 150 s                |             |
| 300 s                |             |

Figura 3.21 – Folha do *Excel* que representa a configuração, Seleção das opções entrada 1, entrada 2 e tempo de atraso da saída 1.

É necessário selecionar o funcionamento pretendido para a entrada 1 e para entrada 2, como é possível verificar, existem duas opções, como alarme ou como controlo. Imaginamos que é selecionado para a entrada 1, controlo e para a entrada 2, também controlo (percurso

verde do DTE, da Figura 3.19). O utilizador pode seleccionar um tempo de atraso para cada saída, para que no caso de haver mais do que uma saída em funcionamento estas não mudem de estado simultaneamente. Para este exemplo, foi seleccionado 0 segundos.

O utilizador escolhe se deseja que a saída 1 esteja ligada ou desligada e a folha de *Excel* automaticamente ajusta-se revelando as decisões que o usuário deve tomar seguidamente, em função das decisões anteriores, como ilustra a Figura 3.22.

|  |  |  |
|--|--|--|
|  | <b>Selecionar:</b>                           |  |
| <b>Saída 1 (led Encarnado):</b>              | Ligada                                       |  |
| Ligada                                       |  |  |
| Desligada                                    |  |  |
|  | <b>Selecionar:</b>                           |  |
| <b>Pergunta</b>                              | Ligar ou Desligar Escalão                    |  |
| Ligar ou Desligar a Bateria de Condensadores | Ligar ou Desligar a Bateria de Condensadores |  |
| Ligar ou Desligar Escalão                    | Ligar ou Desligar Escalão                    |  |
|  | <b>Selecionar:</b>                           |  |
| <b>Escalão</b>                               | Capacitivo                                   |  |
| Indutivo                                     |  |  |
| Capacitivo                                   |  |  |
|  | <b>Selecionar:</b>                           |  |
| <b>Tempo de descarga</b>                     | 600 s  |  |
| 60 s   |  |  |
| 180 s  |  |  |
| 600 s  |  |  |

Figura 3.22 – Folha do *Excel* que representa a configuração, Seleção das opções para a saída 1.

Supondo que é determinado que a saída 1 está ligada, então como ilustra a Figura 3.22, a decisão a tomar a seguir é se o utilizador deseja que funcione como ligar ou desligar a BC, ou se deseja que funcione como inserção ou não de um escalão. Caso se opte por ligar ou desligar um escalão, então a decisão seguinte para o utilizador é seleccionar se o escalão é de carácter indutivo ou capacitivo, uma vez que se for capacitivo, é necessário estabelecer quanto tempo é necessário para o condensador descarregar totalmente. Desta forma a saída só voltará a ser ligada após este tempo, independentemente do utilizador tentar ligar a saída antes deste tempo terminar.

Para finalizar, sendo a saída 1 direccionada para o ligar ou desligar um escalão, significa que este escalão só pode ser ativo a partir de uma entrada, como ambas as entradas funcionam como controlo, então tanto pode ser acionada pela entrada 1, como pela entrada 2, mas só a partir de uma delas. Na eventualidade de o utilizador se enganar e decidir que nem a entrada 1, nem a entrada 2, controlam o escalão, então aparece uma mensagem a vermelho,

como ilustra a Figura 3.23. Nesta situação o usuário deve rever as suas decisões e alterar, uma das opções anteriores.

|                                  |   |
|----------------------------------|---|
|                                  | <b>Selecionar:</b>                                    |
| <b>Pergunta</b>                  | Entrada 1 não controla a saída 1                      |
| Entrada 1 controla a saída 1     |   |
| Entrada 1 não controla a saída 1 |   |
|                                  |   |
|                                  | <b>Selecionar:</b>                                    |
| <b>Pergunta</b>                  | Entrada 2 não controla a saída 1                      |
| Entrada 2 controla a saída 1     | Entrada 2 controla a saída 1                          |
| Entrada 2 não controla a saída 1 | Entrada 2 não controla a saída 1                      |
|                                  |   |
| <b>Afirma</b>                    | OPÇÃO INVALIDA, por favor reveja as opções anteriores |

Figura 3.23 – Folha do *Excel* que representa a configuração, Seleção das opções finais para a saída 1.

#### Exemplo para a saída 1, Manipulação dos botões:

Utilizando as mesmas decisões tomadas no exemplo a partir do *Excel*, vai ser ilustrado na Figura 3.24, o código implementado no equipamento e o processo que o utilizador deve efetuar.

```
void EscolhaTarifario() {
    Alarm.delay(1000);
    lcd.clear();
    myFile = SD.open("CONF.txt");//Abertura do documento txt com o nome CONF

    if (myFile) {//Se o ficheiro existe
        if (myFile.available() > 0) {// lê o documento até não haver mais nada escrito nele

            Regiao = myFile.parseInt();
            Tarifario = myFile.parseInt();
            entrada1 = myFile.parseInt();
            entrada2 = myFile.parseInt();
            TempoSaidal = myFile.parseInt();
            TempoSaida2 = myFile.parseInt();
            TempoSaida3 = myFile.parseInt();

            LIGADO_S1 = myFile.parseInt();
            saidal = myFile.parseInt();
            indutivoS1 = myFile.parseInt();
            TempoBCS1 = myFile.parseInt();
            IN1OUT1 = myFile.parseInt();
            RespostaIN1OUT1 = myFile.parseInt();
            IN2OUT1 = myFile.parseInt();
            RespostaIN2OUT1 = myFile.parseInt();
        }
    }
}
```

```

LIGADO_S2 = myFile.parseInt();
saida2 = myFile.parseInt();
indutivoS2 = myFile.parseInt();
TempoBCS2 = myFile.parseInt();
IN1OUT2 = myFile.parseInt();
RespostaIN1OUT2 = myFile.parseInt();
IN2OUT2 = myFile.parseInt();
RespostaIN2OUT2 = myFile.parseInt();

LIGADO_S3 = myFile.parseInt();
saida3 = myFile.parseInt();
indutivoS3 = myFile.parseInt();
TempoBCS3 = myFile.parseInt();
IN1OUT3 = myFile.parseInt();
RespostaIN1OUT3 = myFile.parseInt();
IN2OUT3 = myFile.parseInt();
RespostaIN2OUT3 = myFile.parseInt();
}
myFile.close();// Fecha o ficheiro
}

```

Figura 3.24 – Rotina para a implementação das decisões do utilizador, leitura do documento .txt.

Como se pode verificar pela Figura 3.24, a rotina correspondente às decisões do utilizador é dominada de “EscolhaTarifario”. A declaração *Void*, representa a função utilizada para criar uma fração de código que não retorna nenhuma informação a partir da qual esta rotina foi chamada [4]. Primeiro é apagado tudo o que se encontra escrito no LCD, por forma a ser possível digitar as questões ao utilizador, em seguida é aberto o ficheiro em formato de texto do cartão SD, que armazena ou irá armazenar a informação das decisões do utilizador, este documento está denominado por CONF.

Se o documento de texto existir, o protótipo lê o valor das variáveis segundo a ordem descrita na Figura 3.24, ou seja, na situação de o utilizador ter declarado as funcionalidades do equipamento a partir do *Excel*, caso contrário aparece escrito no LCD que o ficheiro não existe e inicia as questões ao utilizador, como ilustra a Figura 3.25. Em qualquer altura o utilizador pode alterar as suas decisões, sem que para isso tenha que aceder ao cartão SD por via computador, basta pressionar o botão 1 e 3 em simultâneo e o utilizador toma as suas decisões manipulando os botões.

```

else {

    lcd.setCursor(0, 0);
    lcd.print("Conf.Nao encontrado");
    Alarm.delay(2000);// Tempo de espera de 2 segundo

    //****Seleção da Região*****
    lcd.setCursor(0, 0);
    lcd.print("Selecione a Regiao ");
    lcd.setCursor(0, 1);
    lcd.print("1-Portugal");
    lcd.setCursor(0, 2);
    lcd.print("2-Acores");
    lcd.setCursor(0, 3);
    lcd.print("3-Madeira");

    Regiao = ReadButtons(3);

    Alarm.delay(1000);
}

```

Figura 3.25 – Rotina para a implementação das decisões do utilizador, seleção da Região.

O utilizador, para efetuar a decisão, seleciona o botão que corresponde a opção 1, 2 ou 3, sendo que os botões se encontram dispostos, por ordem de cima para baixo. Como aparece na Figura 4.10, a rotina que efetua a leitura do botão pressionado é o “ReadButtons”, em que entre parêntesis aparece quantos botões estão disponíveis mediante as opções existentes em cada pergunta. Neste caso é guardado na variável denominada “Regiao” o valor do botão selecionado (1, 2 ou 3).

Imaginado que o utilizador seleciona o botão correspondente à opção 1, Portugal, então em seguida é realizada a pergunta da Figura 3.26.

```

//****Seleção do Tarifário*****
switch (Regiao) {
  case 1:
Regiao1:
  lcd.setCursor(0, 0);
  lcd.print("Selecione Tarifario ");
  lcd.setCursor(0, 1);
  lcd.print("1-Ciclo Semanal" );
  lcd.setCursor(0, 2);
  lcd.print("2-Sem.Opc (MAT,AT,MT) ");
  lcd.setCursor(0, 3);
  lcd.print("3-Diario (BTE,BTN)  ");
  lcd.home();

  Tarifario = ReadButtons(3);
  Alarm.delay(100);
  goto fim;
  break;
}

```

Figura 3.26 – Rotina para a implementação das decisões do utilizador, seleção do Tarifário.

Uma vez que a seleção do Tarifário depende da Região selecionada, foi optado por utilizar a função *switch case*. Esta função é semelhante à função *if (se)*, isto é, apenas uma parcela do código é efetuada, portanto quando a declaração do *case* contém o valor que corresponde ao da variável, o código na declaração desse caso é executado. No final de cada *case*, existe a função *break*, que é utilizada para declarar o final de cada caso [4].

Como ilustra a Figura 3.26, a variável “Regiao” tinha guardado o valor 1, como tal entra no caso 1 e executa o código, que consiste em perguntar ao utilizador que tarifário pretende, tendo em conta novamente as dimensões de visualização de leitura no LCD. Na variável “Tarifario” é guardado a opção tomada pelo utilizador, que dependo novamente do botão selecionado. Supondo que o usuário opta pela primeira opção, Ciclo Semanal.

Em seguida o utilizador deve selecionar a função que deseja para ambas as entradas respetivamente, como ilustra a Figura 3.27.

```

//****Seleção da função das Entradas*****
Alarm.delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Entrada 1:");
lcd.setCursor(0, 1);
lcd.print("1-Alarme");
lcd.setCursor(0, 2);
lcd.print("2-Controlo");

entrada1 = ReadButtons(2);

Alarm.delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Entrada 2:");
lcd.setCursor(0, 1);
lcd.print("1-Alarme");
lcd.setCursor(0, 2);
lcd.print("2-Controlo");

entrada2 = ReadButtons(2);

Alarm.delay(1000);
//*****

```

Figura 3.27 – Rotina para a implementação das decisões do utilizador, seleção da Entrada 1 e 2.

O processo de seleção é semelhante às situações anteriores, apenas é importante referir que foram colocados *delays* de 1 segundo, ou seja, um tempo de espera para assimilação de informação e para ser perceptível ao utilizador a alteração e o surgimento de novas questões. Supondo mais uma vez que o usuário seleciona ambas as entradas como controlo.

Como se encontra no *Excel* e no DTE, em seguida são seleccionados os tempos de atraso em cada uma das saídas, como o exemplo é apenas respetivo à saída 1, então o utilizador deve responder, como apresentado na Figura 3.28.

```

//*****Seleção da existência de atraso nas saídas*****
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Atraso Saida 1:");
lcd.setCursor(0, 1);
lcd.print("1-");
lcd.print(TempoSaida_a);
lcd.print(" s");
lcd.setCursor(0, 2);
lcd.print("2-");
lcd.print(TempoSaida_b);
lcd.print(" s");
lcd.setCursor(0, 3);
lcd.print("3-");
lcd.print(TempoSaida_c);
lcd.print(" s");

TempoSaida1 = ReadButtons(3);
if (TempoSaida1 == 1) {
    TempoSaida1 = TempoSaida_a;
}
else if (TempoSaida1 == 2) {
    TempoSaida1 = TempoSaida_b;
}
else if (TempoSaida1 == 3) {
    TempoSaida1 = TempoSaida_c;
}

```

Figura 3.28 – Rotina para a implementação das decisões do utilizador, seleção do tempo de atraso.

O processo de seleção é semelhante, a diferença de código consiste nas variáveis “TempoSaida\_”a,b e c serem declaradas com valores no início de todo o código, que são valores em segundos. Como no *Excel*, estes valores são 0, 150 e 300 segundos, a sua declaração é realizada no início para, caso se queira alterar estes valores, apenas será necessário fazê-lo num único sítio, ou seja, para facilitar o código. A variável *TempoSaida1*, guarda a opção selecionada pelo utilizador, isto é, botão 1, 2 ou 3.

A decisão a ser tomada seguidamente, depende das opções selecionadas para as entradas. Assim sendo, cada saída vai conter quatro parcelas de código que dependem das quatro sequências existentes e possíveis para as entradas, que se encontram apresentadas na Tabela 3.9.

Tabela 3.9 – Sequências possíveis para as entradas

| Situações | Entrada 1 | Entrada 2 |
|-----------|-----------|-----------|
| 1         | Alarme    | Alarme    |
| 2         | Controlo  | Controlo  |
| 3         | Alarme    | Controlo  |
| 4         | Controlo  | Alarme    |

O código é diferente para cada uma das situações, pois a função alarme difere bastante da de controlo. Se uma das entradas for de alarme, significa que sempre que a entrada com essa função for ativa, automaticamente todas as saídas são desligadas e emite um sinal sonoro. Na situação em que a entrada esteja como controlo e a saída esteja acionada por essa entrada, então sempre que é ativada a entrada, liga automaticamente a saída a que lhe corresponde.

Portanto, se ambas as entradas forem de alarme significa que as saídas são controladas unicamente pelo horário da ERSE, quando nenhuma das entradas está ativa. Se as duas entradas forem de controlo, então cada saída pode ser ativada por qualquer uma das entradas (mas não pelas duas), ou pelo horário e uma das entradas, ou apenas pelo horário. É de prever que se uma entrada for alarme e a outra de controlo, então as saídas só poderão ser controladas ou pela entrada de controlo, ou por essa entrada e horário, ou ainda só pelo horário. No entanto, se a entrada de alarme for ativada sobrepõe-se à entrada de controlo desligando todas as saídas.

Seguindo o mesmo raciocínio do exemplo dado para o *Excel*, as duas entradas foram selecionadas como controlo, então a parcela de código é a apresentada na Figura 3.29.

```

/*+++++/*S1 controlo controlo ++++++
else if (entrada1 == 2 && entrada2 == 2) {

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("OUT 1 ligada:");
    lcd.setCursor(0, 1);
    lcd.print("1-SIM");
    lcd.setCursor(0, 2);
    lcd.print("2-NAO");

    LIGADO_S1 = ReadButtons(2);

    Alarm.delay(1000);

    if (LIGADO_S1 == 1) {
        Alarm.delay(1000);
SaidaE1:
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Saida 1 :");
        lcd.setCursor(0, 1);
        lcd.print("1-ON/OFF BCondensado");
        lcd.setCursor(0, 2);
        lcd.print("2-ON/OFF Escalao");

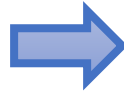
        saida1 = ReadButtons(2);

        Alarm.delay(100);

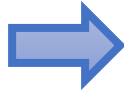
        if (saida1 == 1) {
            BC_S1 = true;
            Escalao_S1 = false;
        }
        if (saida1 == 2) {
            BC_S1 = false;
            Escalao_S1 = true;
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Tipo de Escalao ?");
            lcd.setCursor(0, 1);
            lcd.print("1-Indutivo");
            lcd.setCursor(0, 2);
            lcd.print("2-Capacitivo");
            Alarm.delay(1000);

            indutivoS1 = ReadButtons(2);
            Alarm.delay(2500);

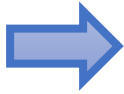
```



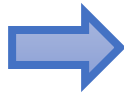
Parcela de código com a entrada 1 e 2 com a função de controlo.



Seleção da saída 1 como ligada ou desligada, ou seja, existente ou não.



Se for optado por a saída 1 estar ligada, então é questionado se é como ligar ou desligar a BC ou como ligar ou desligar um escalão. Supondo que o utilizador seleciona a opção 2, ligar ou desligar um escalão.



Sendo a opção ligar ou desligar um escalão, a decisão a tomar a seguir é se o escalão é de carácter indutivo ou de carácter capacitivo.

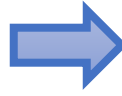
```

if (indutivoS1 == 2) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Tempo de descarga ?");
  lcd.setCursor(0, 1);
  lcd.print("1-");
  lcd.print(TempoBCS_a);
  lcd.print(" s");
  lcd.setCursor(0, 2);
  lcd.print("2-");
  lcd.print(TempoBCS_b);
  lcd.print(" s");
  lcd.setCursor(0, 3);
  lcd.print("3-");
  lcd.print(TempoBCS_c);
  lcd.print(" s");

  TempoBCS1 = ReadButtons(3);
  if (TempoBCS1 == 1) {
    TempoBCS1 = TempoBCS_a;
  }
  if (TempoBCS1 == 2) {
    TempoBCS1 = TempoBCS_b;
  }
  if (TempoBCS1 == 3) {
    TempoBCS1 = TempoBCS_c;
  }

  Alarm.delay(2000);
}
else {
  TempoBCS1 = 0;
}
}

```



Se a decisão for de carácter capacitivo (“indutivoS1 == 2”), é necessário ter em conta o tempo de descarga do condensador. Semelhante ao tempo de atraso, este tempo em segundos é definido no início do código, sendo que as opções de momento são 60, 180 e 600 segundos.

Significa que se for acionada esta saída a partir de uma entrada, e o condensador ainda estiver a descarregar, a saída não é novamente ligada até este tempo tiver terminado.

Figura 3.29 – Rotina para a implementação das decisões do utilizador, seleção da função para a saída1.

À semelhança do exemplo do *Excel* é necessário optar de que forma esta saída vai ser acionada. Como a função é respetiva a um escalão, então só pode ser acionada pela entrada 1 ou pela entrada 2. Caso o utilizador se engane e selecione a entrada 1 e a entrada 2 não acionam a saída 1, aparecerá a informação no LCD de opção inválida. Reiniciando as questões a partir da fração de código onde se encontra escrito “SaidaE1”, ilustrado na Figura 3.29, ordem que é dada na Figura 3.30, “goto SaidaE1”.

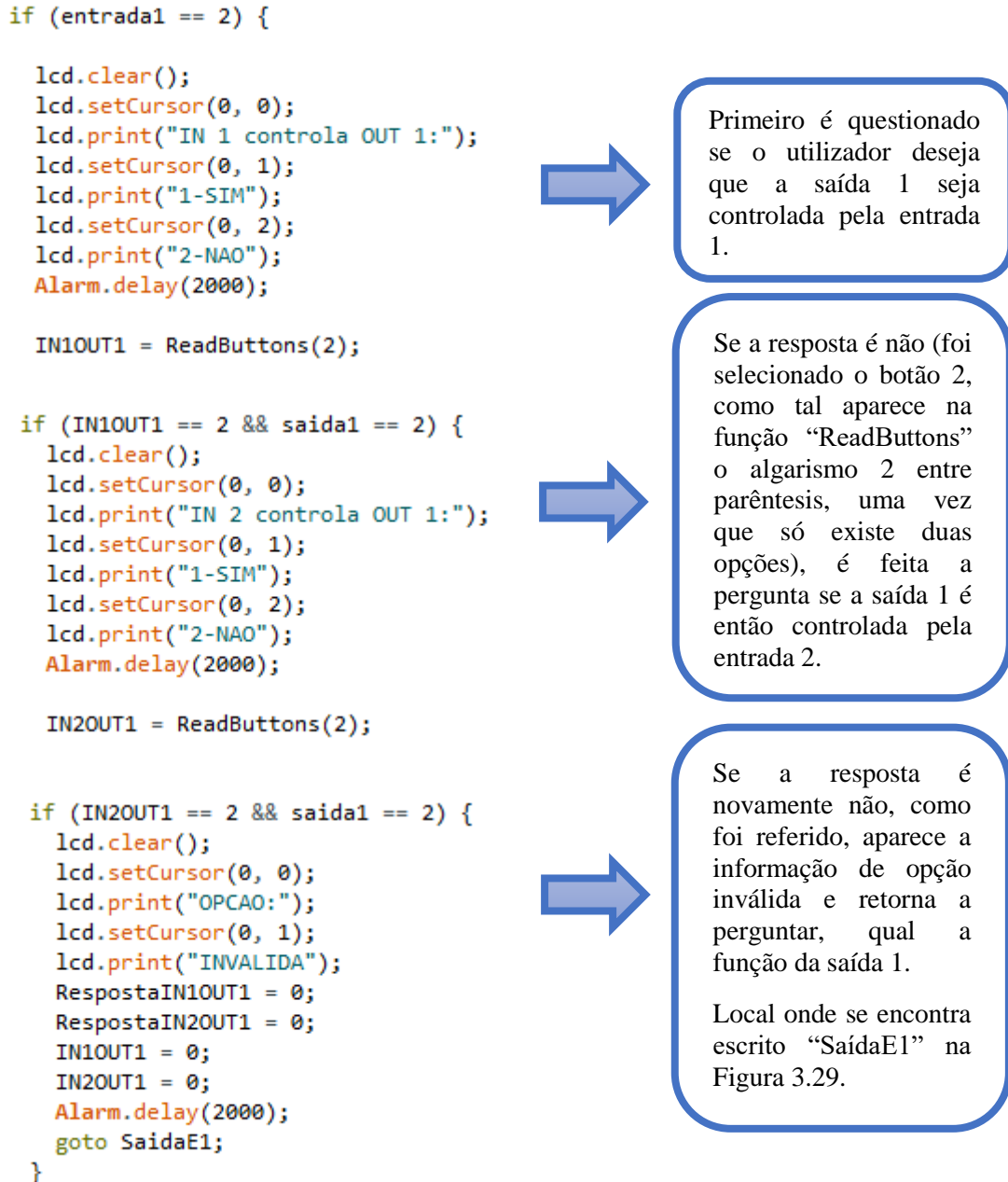


Figura 3.30 – Rotina para a implementação das decisões do utilizador, seleção da entrada para ativar a saída1.

Na Figura 3.30 é só demonstrado os conjuntos de *if* e *else*, mediante as opções que forma optadas como exemplo de respostas dadas pelo utilizador.

### 3.4.2. Algoritmo

Sempre que é criado um novo documento utilizando o programa Arduino, o local onde é digitado o algoritmo contém duas funções escritas, a de *setup* que é uma função que é

solicitada uma única vez, no momento em que o programa inicia ou quando é reiniciado e a função *loop*, que é uma função que como o nome indica é a fração de código que se repete continuamente, permitindo que o programa funcione dinamicamente [4].

Como foi referido na introdução, o código encontra-se distribuído segundo um programa principal, que engloba as duas funções descritas anteriormente o *setup* e o *loop*, onde são chamadas as várias rotinas pela ordem correta, uma secção de declaração de variáveis e uma secção para as bibliotecas utilizadas.

Serão seguidamente expostas as 3 frações que constituem o código, mas antes é importante referir que no programa principal, serão reveladas apenas algumas frações de código e não todo, por questões de sigilo com a empresa envolvida.

Fração do código com as bibliotecas utilizadas, Figura 3.31.

```

//*****Livrarias*****:
const int chipSelect = 9;
#include <LiquidCrystal.h>
#include <Wire.h>
#include <DS3231.h>
#include <SPI.h>
#include <SD.h>
#include <TimeAlarms.h>
#include <TimeLib.h>
//*****:
// Nome do ficheiro para leitura e escrita no cartão SD
File myFile;
// Pinos conetados ao LCD
LiquidCrystal lcd(A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15);
// Declaração dos pinos do RTC
DS3231 rtc(SDA, SCL);

```

Figura 3.31 – Fração do código respetivo às bibliotecas utilizadas.

A primeira linha que aparece na Figura 3.31 diz respeito não a uma biblioteca, mas à seleção do pino para o CS. Para a biblioteca do cartão SD funcionar é necessário selecionar o pino do CS, uma vez que é necessário ao protocolo SPI. As restantes bibliotecas são apresentadas na Tabela 3.10.

Tabela 3.10 – Bibliotecas utilizadas.

| Bibliotecas                | Função   |
|----------------------------|--|
| #include <LiquidCrystal.h> | Esta biblioteca permite que a placa Arduino controle o LCD [4].  |
| #include <Wire.h>          | Esta biblioteca permite a comunicação com dispositivos com o protocolo I <sup>2</sup> C [4].   |
| #include <DS3231.h>        | Esta biblioteca permite a utilização do RTC externo, de forma a obter a hora e data de forma simples lida e escrita [4].   |
| #include <SPI.h>           | Esta biblioteca permite a comunicação do microcontrolador com dispositivos periféricos a partir do protocolo SPI, como se encontra na Figura 3.1 [4].  |
| #include <SD.h>            | Esta biblioteca permite ler e escrever em cartões SD [4].  |
| #include <TimeAlarms.h>    | Esta biblioteca não requer de nenhum <i>hardware</i> especial, porque usa a hora e data fornecidas pela biblioteca Time. A função desta biblioteca é criar alarmes, que são usados para chamar uma função em uma hora específica do dia [4]. |
| #include <TimeLib.h>       | Esta biblioteca permite adicionar a funcionalidade de tempo para o Arduino com ou sem <i>hardware</i> externo, por forma a obter a hora e a data como: segundo, minuto, hora, dia, mês e ano [4].  |

Fração do código para declaração de variáveis, Figura 3.32.

```
// ***** Declaração das Variáveis*****

int Regiao, Tarifario, saidal, saida2, saida3, entradal, entrada2;
int LIGADO_S1, LIGADO_S2, LIGADO_S3;
int IN1OUT1, IN1OUT2, IN1OUT3, IN2OUT1, IN2OUT2, IN2OUT3;
int RespostaIN1OUT1, RespostaIN1OUT2, RespostaIN1OUT3;
int RespostaIN2OUT1, RespostaIN2OUT2, RespostaIN2OUT3;
int Button = 0;
int indutivoS1, indutivoS2, indutivoS3;
int Controlo_S1, Controlo_S2, Controlo_S3, TempoSaidal, TempoSaida2, TempoSaida3;
int TempoTotalS1, TempoTotalS2, TempoTotalS3, TempoBCS1, TempoBCS2, TempoBCS3;

//*****Tempo de Atraso para a Saída 1, 2 e 3*****
int TempoSaida_a = 0;
int TempoSaida_b = 150;
int TempoSaida_c = 300;
//*****

//*****Tempo de Decarga da Bateria de Condensadores na situação de Saída 1, 2 e 3***
int TempoBCS_a = 20;
int TempoBCS_b = 180;
int TempoBCS_c = 600;
//*****

int HoraDoDia;
Time t;
TimeElements tm;

String NomeFicheiro = "FFFFFF";
unsigned long Agora, Horas, Temporizador, botaol3, tempobotaol, tempobotao2;
unsigned long manutencao, botaol2, ultimaData;
boolean FeriadoDomingo, sabado, Seg_sexta, Verao;
boolean Escalao_S1, BC_S1, Escalao_S2, BC_S2, BC_S3, Escalao_S3;
boolean Registo_saidal, RegistoAnterior_saidal;
boolean Registo_saida2, RegistoAnterior_saida2;
boolean Registo_saida3, RegistoAnterior_saida3;
boolean Feriado3meses;
byte Comando;
unsigned long TempoOFF1, TempoOFF2, TempoOFF3;

int IDAlarmeS1Ligar, IDAlarmeS2Ligar, IDAlarmeS3Ligar;
int IDAlarmeS1Desligar, IDAlarmeS2Desligar, IDAlarmeS3Desligar;
```

```

boolean buzzer = true;
boolean Alarme = false;
boolean botao1FALLING = false;
boolean botao2FALLING = false;
boolean botao1RISING = false;
boolean botao2RISING = false;
boolean EstadoBotao1 = false;
boolean EstadoBotao2 = false;

int Button1Pin = 17;
int Button2Pin = 16;
int Button3Pin = 6;

```

Figura 3.32 – Fração do código respetivo à declaração das variáveis.

Como é possível verificar pela Figura 3.32, foram utilizados vários tipos de dados, na declaração das variáveis, sendo apresentados na Tabela 3.11.

Tabela 3.11 – Tipos de dados utilizados na declaração das variáveis.

| Tipo de dados        | Função  |
|----------------------|---|
| <i>Int</i>           | Representa os números inteiros, onde é possível armazenar valores de 16 bits (2 bytes), ou seja, um intervalo de -32 768 a 32 767 (no Arduino UNO e Mega) [4].        |
| <i>String</i>        | Permite a utilização e manipulação de sequências de texto [4].  |
| <i>Unsigned long</i> | Permite o armazenamento de números a 32 bites (4 bytes) e ao contrário do <i>int</i> , não armazena números negativos, obtendo um intervalo de 0 a 4 294 967 295 [4]. |
| <i>Boolean</i>       | Cada variável deste tipo ocupa um byte de memória, e apenas pode conter dois valores, verdadeiro ou falso (1 ou 0) [4].   |
| <i>Byte</i>          | Um byte armazena um número de 8 bites, de 0 a 255 [4].  |

Existem mais tipos de dados, no entanto como não surgiram no projeto, não serão explicados.

Fração do código do programa principal.○ Setup, Figura 3.33

```

void setup() { // Função executada somente uma vez,
//quando o Arduino é iniciado ou reiniciado

// Inicialização do RTC
rtc.begin();
// Inicialização do LCD(20 colunas por 4 linhas)
lcd.begin (20, 4);
// Inicialização da porta série do PC
Serial.begin(9600);

InicializarTempo();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Cartao SD");

if (!SD.begin(9)) {
  lcd.setCursor(0, 1);
  lcd.print("Inicializacao falhou!");
  Alarm.delay(3000);
  lcd.clear();
  return;
}

lcd.setCursor(0, 1);
lcd.print("Inicializacao ok.");
Alarm.delay(3000);
lcd.clear();

VerificarVerao();
VeraoInverno();
Alarmes();
VerificarHora();

BC_S1 = false;
BC_S2 = false;
BC_S3 = false;
Escalao_S1 = false;
Escalao_S2 = false;
Escalao_S3 = false;
//*****

pinMode (Button1Pin, INPUT_PULLUP);
pinMode (Button2Pin, INPUT_PULLUP);
pinMode (Button3Pin, INPUT_PULLUP);
//*****
pinMode (12, OUTPUT);
digitalWrite (12, HIGH);
//*****

```

Rotina para configurar a hora, a data e o dia da semana, a partir do RTC.

Verificação da inicialização da biblioteca e do cartão SD, a partir do CS (SD.begin(9)).

Se aparecer no LCD, "Inicialização falhou!", o utilizador deve tirar o cartão e voltar a coloca-lo.

São chamadas 4 rotinas, estas serão explicadas posteriormente.

Inicialização a falso (0), as variáveis que determinam se as saídas se aplicam a BC ou escalão.

Declaração das variáveis designadas aos botões.

Pino que designa a luminosidade do LCD.

```

//Entrada 1
pinMode(18, INPUT);

//Entrada 2
pinMode(19, INPUT);

Alarm.delay(50);

attachInterrupt(5, Bota01FALLING, FALLING);
attachInterrupt(4, Bota02FALLING, FALLING);

Alarm.delay(50);

EscolhaTarifario();

if (Regiao == 1)VerificarDia();

CiclosTarifario();

pinMode(27, OUTPUT); //saida 1
pinMode(28, OUTPUT); //saida 2
pinMode(29, OUTPUT); //saida 3

SaidasComHorario();
}

```

Pinos alusivos às entradas.

São usados *interrupts* nas entradas, para garantir que não existem flutuações que façam mudar o estado da entrada, sem que seja esse o propósito. Desta forma é desnecessário temporizações. Existem pinos específicos para o efeito.

4 Rotinas que serão explicadas a seguir. Declaração dos pinos respeitantes às saídas.

Figura 3.33 – Fração do código respetivo ao *setup*.

Como está referido na Figura 3.33 é utilizado a função “attachInterrupt ()” para as entradas. Para configurar esta função é colocado entre parêntesis em primeiro o número da interrupção, que difere entre os modelos de placa de Arduino, como é possível constatar na Figura 3.34.

| Placas                         | int.0                                | int.1 | int.2 | int.3 | int.4 | int.5 |
|--------------------------------|--------------------------------------|-------|-------|-------|-------|-------|
| Uno, Ethernet                  | 2                                    | 3     |       |       |       |       |
| Mega2560                       | 2                                    | 3     | 21    | 20    | 19    | 18    |
| 32u4 baseado (Leonardo, Micro) | 3                                    | 2     | 0     | 1     | 7     |       |
| Due, Zero, MKR1000, 101        | Número do interrupt = Número do pino |       |       |       |       |       |

Figura 3.34 – Os pinos dos *interrupts* associados às placas de Arduino (traduzido para português) [4].

O segundo campo é o nome da rotina que será associado à interrupção, e o terceiro campo é o modo de interrupção, para o Mega é possível em 4 modos [4]:

- *Low* (Baixo), dispara a interrupção sempre que o pino associado se encontra no estado 0;
- *Change* (Mudar), dispara a interrupção sempre que o pino associado muda de estado;
- *Rising* (Aumenta), dispara a interrupção sempre que o pino associado vai do estado 0 para 1;
- *Falling* (Cai), dispara a interrupção sempre que o pino associado vai do estado 1 para 0;

As interrupções são úteis quando se quer garantir que a mudança de estado é realizada somente segundo um dos modos anteriores, permitindo que na eventualidade de haver flutuações a entrada não mude de estado, se não foi atuado essa mudança de estado.

Várias rotinas são chamadas no corpo do código do *setup*, a “InicializarTempo ()”, a “VerificarVerao ()”, a “VeraoInverno ()”, a “Alarmes ()”, a “VerificarHora ()”, a “EscolhaTarifario ()”, o “VerificarDia ()”, a “CiclosTarifario ()” e a “SaidasComHorario ()”. Os nomes das rotinas mencionadas são escolhidos por forma a serem nomes sugestivos e evidentes no código expetáveis da rotina, apresentados na Tabela 3.12. Seguidamente será exposto e explicado o código de algumas das rotinas mencionadas.

Tabela 3.12 – Identificação das rotinas utilizadas no *setup*.

| Nome da Rotina        | Explicação  |
|-----------------------|---|
| “InicializarTempo ()” | Utiliza a configuração da biblioteca do RTC, para explicitar a hora, a data e o dia da semana.  |
| “VerificarVerao ()”   | Verifica se a presente data se encontra na hora legal de verão ou de inverno.   |
| “VeraoInverno ()”     | Perante a hora legal, coloca um alarme para atrasar ou adiantar a hora na altura correta.   |
| “Alarmes ()”          | Colocação de alarmes para a leitura do cartão SD, de 15 em 15 minutos.  |
| “VerificarHora ()”    | Em caso de falha de energia, quando reiniciado, vai verificar a hora atual e continua o processo a partir desta.  |
| “EscolhaTarifario ()” | Diz respeito à interface com o utilizador, para tomada de decisões, como ilustrado no tópico do exemplo.  |
| “VerificarDia ()”     | Segundo os horários da ERSE, quando a Região for igual a Portugal Continental (1), é necessário verificar o dia da semana, sendo esse o dever desta rotina. Para além disso também verifica se os feriados guardados no documento de texto respetivo, estão a 3 meses de expirar. |
| “CiclosTarifario ()”  | Seleção do documento de texto correto a ler, mediante as escolhas do utilizador.  |
| “SaidasComHorario ()” | Atualizar o estado das saídas quando o equipamento é iniciado ou reiniciado.  |

A rotina “VerificarVerao ()”, como o nome sugere, serve para averiguar se está presentemente na hora legal de verão ou na hora legal de inverno, uma vez que os horários fornecidos pela ERSE estão divididos segundo essas estações. A mudança da hora legal de verão para inverno e inverno para verão é segundo o último domingo do mês de outubro e o último domingo do mês de março, respetivamente. Para determinar o último domingo do mês, foi feito o seguinte raciocínio. Tanto o mês de outubro como de março, acabam no dia 31,

como tal, para descobrir o último domingo, existem as possibilidades apresentadas na Tabela 3.13.

Tabela 3.13 – Distribuição do domingo na última semana do mês de outubro ou março.

| D<br>I<br>A<br>S<br>D<br>O<br>M<br>I<br>N<br>G<br>O | Dias do Mês    |                |                |                |                |                |                |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|   | 25             | 26             | 27             | 28             | 29             | 30             | 31             |
|   | <b>Domingo</b> | Segunda        | Terça          | Quarta         | Quinta         | Sexta          | Sábado         |
|   | Sábado         | <b>Domingo</b> | Segunda        | Terça          | Quarta         | Quinta         | Sexta          |
|   | Sexta          | Sábado         | <b>Domingo</b> | Segunda        | Terça          | Quarta         | Quinta         |
|   | Quinta         | Sexta          | Sábado         | <b>Domingo</b> | Segunda        | Terça          | Quarta         |
|   | Quinta         | Quinta         | Sexta          | Sábado         | <b>Domingo</b> | Segunda        | Terça          |
|   | Terça          | Quinta         | Quinta         | Sexta          | Sábado         | <b>Domingo</b> | Segunda        |
|   | Segunda        | Terça          | Quinta         | Quinta         | Sexta          | Sábado         | <b>Domingo</b> |

O cálculo é simples, interessa somente saber se o dia atual está após o dia 25 ou inclusive, em seguida interessa saber se o dia presente é segunda, terça, quarta, quinta, sexta, sábado ou domingo, por forma a descobrir qual é o último domingo, como apresenta a Tabela 3.14.

Tabela 3.14 – Cálculo para saber se já foi o último Domingo do mês de outubro ou março.

| Dia da Semana | Cálculo                    |
|---------------|----------------------------|
| Domingo       | Dia atual $-0 + 7 \leq 31$ |
| Segunda       | Dia atual $-1 + 7 \leq 31$ |
| Terça         | Dia atual $-2 + 7 \leq 31$ |
| Quarta        | Dia atual $-3 + 7 \leq 31$ |
| Quinta        | Dia atual $-4 + 7 \leq 31$ |
| Sexta         | Dia atual $-5 + 7 \leq 31$ |
| Sábado        | Dia atual $-6 + 7 \leq 31$ |

Em síntese o código desta rotina é apresentado na Figura 3.35.

```

void VerificarVerao() {
    t = rtc.getTime();

    if ((rtc.getMonthStr() == "Abril" )|
        (rtc.getMonthStr() == "Maio" )|
        (rtc.getMonthStr() == "Junho")|
        (rtc.getMonthStr() == "Julho")|
        (rtc.getMonthStr() == "Agosto")|
        (rtc.getMonthStr() == "Setembro")) {
        Verao = true;// é verão
    }

    if ((rtc.getMonthStr() == "Novembro")|
        (rtc.getMonthStr() == "Dezembro")|
        (rtc.getMonthStr() == "Janeiro")|
        (rtc.getMonthStr() == "Fevereiro")) {
        Verao = false;// é inverno
    }

    if (rtc.getMonthStr() == "Marco") {
        if (t.date <= 24) {
            Verao = false;
        }

        else {
            if (rtc.getDOWStr() == "Segunda") {
                if (t.date - 1 + 7 <= 31) {
                    Verao = false;
                }
                else {
                    Verao = true;
                }
            }
            if (rtc.getDOWStr() == "Terca") {
                if (t.date - 2 + 7 <= 31) {
                    Verao = false;
                }
                else {
                    Verao = true;
                }
            }
            if (rtc.getDOWStr() == "Quarta") {
                if (t.date - 3 + 7 <= 31) {
                    Verao = false;
                }
                else {
                    Verao = true;
                }
            }
        }
    }
}

```



Guarda o tempo na variável t.



Se a data atual corresponder a algum daqueles meses, significa que estamos na hora legal de verão.



Se corresponder a estes meses, significa que estamos na hora legal de inverno.



Se não for nenhum dos meses anteriores e for março, então o dia é 24 ou menor? se sim é inverno, caso contrário:



Dependo do dia da semana, faz o cálculo explicitado da Tabela 3.14 e verifica se o último domingo já passou.

```

if (rtc.getDOWStr() == "Quinta") {
    if (t.date - 4 + 7 <= 31) {
        Verao = false;
    }
    else {
        Verao = true;
    }
}
if (rtc.getDOWStr() == "Sexta") {
    if (t.date - 5 + 7 <= 31) {
        Verao = false;
    }
    else {
        Verao = true;
    }
}
if (rtc.getDOWStr() == "Sabado") {
    if (t.date - 6 + 7 <= 31) {
        Verao = false;
    }
    else {
        Verao = true;
    }
}
if (rtc.getDOWStr() == "Domingo") {
    if (t.date - 0 + 7 <= 31) {
        Verao = false;
    }
    else {
        Verao = true;
    }
}
}
}

```

Para o mês de outubro o código é semelhante, não sendo por isso ilustrado.

Figura 3.35 – Fração do código respectivo da rotina “VerificarVerao ()”.

A rotina “VerificarHora ()”, ilustrada na Figura 3.36, é uma rotina chamada na inicialização do equipamento ou quando este reinicia, estando por isso a ser chamada no *setup*, com intuito de verificar a hora atual e mediante o horário estabelecido pela ERSE, colocar o equipamento a funcionar na situação correta do horário.

```

//*****:
void VerificarHora() {

    t = rtc.getTime();
    HoraDoDia = t.hour * 4 + t.min / 15;
    CiclosTarifario();
    AtualizarSaidas();
}
//*****:

```

Figura 3.36 – Fração do código respectivo da rotina “VerificarHora ()”.

Esta rotina guarda o tempo atual na variável *t*, em seguida na variável “HoraDoDia” é guardado um valor a partir de um cálculo. Este cálculo permite verificar em que situação nos encontramos no horário da ERSE, perante a hora atual, permitindo segui-lo corretamente.

O horário da ERSE, apresenta alterações em horas, com intervalos de 15 minutos, como tal e como já foi referido, foi optado por redigir o código verificando o horário a cada 15 minutos e efetuar alterações nas saídas se necessário. Na situação em que inicializamos o equipamento com uma certa hora, ou se este é reiniciado, é necessário compreender a partir de que situação do horário da ERSE o equipamento deve continuar a funcionar.

O raciocínio utilizado é que cada hora está dividida em 4 períodos de 15 minutos, portanto de uma hora para a outra, considerando a hora com 0 minutos são necessários 4 períodos. Dividindo os minutos por 15, permite-nos saber quantos mais períodos passaram para além dos já contabilizados nas horas, assim sendo:

$$\text{HoraDoDia} = \text{Hora} * 4 + \frac{\text{Minutos}}{15} \quad (3.3.)$$

Por exemplo:

Considerando que a hora atual é 02:30 h e utilizando a equação (3.3.), obtém-se:

$$\text{HoraDoDia} = 2 * 4 + \frac{30}{15} = 10$$

Significa que começando sempre nas 00:00 h, já passaram 10 períodos. Como apresentado na Tabela 3.15, desta forma é possível colocar o equipamento a funcionar a partir da situação correta do horário estabelecido.

Tabela 3.15 – Exemplo do cálculo para de terminar a hora do dia.

| Período | Horas que passaram até à atual |
|---------|--------------------------------|
| 1       | 00:15                          |
| 2       | 00:30                          |
| 3       | 00:45                          |
| 4       | 01:00                          |
| 5       | 01:15                          |
| 6       | 01:30                          |
| 7       | 01:45                          |
| 8       | 02:00                          |
| 9       | 02:15                          |
| 10      | 02:30                          |

Uma outra rotina que é fulcral para o correto funcionamento do equipamento é a “CiclosTarifario ()”. Esta rotina é a que permite seleccionar o documento de texto correto, consoante a região e tarifário estabelecido pelo utilizador. Em seguida a Figura 3.37 expõe o código e raciocínio utilizado.

```
void CiclosTarifario() {
    //Rotina para ler o ficheiro txt correto consoante as decisões do utilizador

    if (Verao == false) { // Se for inverno
        NomeFicheiro = "I"; //o nome do ficheiro inicia com a letra I
    } else {
        NomeFicheiro = "V"; // Caso contrário inicia com a letra V
    }
    if (Regiao == 1) { //Se a região for 1
        NomeFicheiro += "P"; // a segunda letra do nome do ficheiro é P(Portugal)
    } else if (Regiao == 2) { // caso a região seja 2
        NomeFicheiro += "A"; // a segunda letra do nome do ficheiro é A(Açores)
    } else { // Caso a região seja 3
        NomeFicheiro += "M"; // a segunda letra do nome do ficheiro é M(Madeira)
    }

    if (Regiao == 1 && Tarifario == 1) { // se a Região for 1 e o Tarifario for 1
        NomeFicheiro += "S"; // a terceira letra é S (Ciclo Semanal)
    } else if (Regiao == 1 && Tarifario == 2) { // se a Região for 1 e o Tarifario for 2
        NomeFicheiro += "SO"; // a terceira letra é SO (Ciclo Semanal Opcional)
    } else if (Regiao == 1 && Tarifario == 3) { // se a Região for 1 e o Tarifario for 3
        NomeFicheiro += "D"; // a terceira letra é D (Ciclo Diário)
    } else if (Tarifario == 1) { // se o Tarifario for 1 e a região diferente de 1
```

```

    NomeFicheiro += "D";// a terceira letra é D (Ciclo Diário)
} else if (Tarifario == 2) { // se o Tarifario for 2 e a região diferente de 1
    NomeFicheiro += "DO";// a terceira letra é DO (Ciclo Diário Opcional)
}
//Situação válida apenas para região = 1
if (Seg_sexta == true) { // Se for o dia da semana for de segunda a sexta
    NomeFicheiro += "SS";// ainda escreve SS
} else if (sabado == true) { // se for sabado
    NomeFicheiro += "SA";//ainda escreve SA
} else if (FeriadoDomingo == true) { // se for domingo ou feriado
    NomeFicheiro += "FD";//ainda escreve FD
}
myFile = SD.open(NomeFicheiro + ".txt");// Abre o nome do ficheiro correto
if (myFile) { // Se o ficheiro estiver aberto
    if (myFile.available() > 0) { //para ler o ficheiro até não haver mais nada escrito
        myFile = SD.open(NomeFicheiro + ".txt", FILE_READ);// Abre o documento correto e lê-o
        for (int i = 0; i <= HoraDoDia; i++) {
            // Descobre o tempo que seja igual ao tempo guardado na HoraDoDia
            Horas = myFile.parseInt();//Devolve o valor e guarda-o na variavel Horas
        }
    }
}

myFile.close();// Fecha o documento
}
if ((RespostaIN1OUT1 == 2) |
    (RespostaIN1OUT1 == 3) |
    (RespostaIN2OUT1 == 2) |
    (RespostaIN2OUT1 == 3) |
    (RespostaIN1OUT2 == 2) |
    (RespostaIN1OUT2 == 3) |
    (RespostaIN2OUT2 == 2) |
    (RespostaIN2OUT2 == 3) |
    (RespostaIN1OUT3 == 2) |
    (RespostaIN1OUT3 == 3) |
    (RespostaIN2OUT3 == 2) |
    (RespostaIN2OUT3 == 3)) {

    if (Horas == 0) { // Se as Horas for igual a 0
        lcd.setCursor(0, 3);
        lcd.print("Ponta      "); //Escreve Ponta no LCD
    } else if (Horas == 1) { // Se as Horas for igual a 1
        lcd.setCursor(0, 3);
        lcd.print("Cheias      "); //Escreve Cheias no LCD
    } else if (Horas == 2) { // Se as Horas for igual a 2
        lcd.setCursor(0, 3);
        lcd.print("VazioNormal");//Escreve Vazio Normal no LCD
    } else if (Horas == 3) { // Se as Horas for igual a 3
        lcd.setCursor(0, 3);
        lcd.print("Super Vazio");//Escreve Super Vazio no LCD
    }
}
}

```

Figura 3.37 – Fração do código respetivo da rotina “CiclosTarifario ()”.

Como se verifica pela Figura 3.37 os nomes dos documentos de textos são criados segundo uma sequência de letras, sequência essa que é criada mediante as escolhas do utilizador, em relação à região e tarifário. Desta forma é lido somente o documento correto e o código praticado fica pouco extenso e simples.

- *Loop*, Figura 3.38

```
void loop() {
    interruptBotaol();
    interruptBotao2();
    Manutencao();
    TextoLCD();
    Botaole3Reset();
    Botaole2Reset();
    Alarm.delay(1000);
}
```



Fração de código que se repete continuamente. Portanto apenas estão as rotinas que convêm que estejam sempre a ser repetidas.

As rotinas para os *interrupts* das entradas, verificando sempre o seu estado; a manutenção, que corresponde a uma data estabelecida que emite um sinal sonoro aquando dessa data; todo o texto que é apresentado no LCD atualizado; a rotina para quando é pressionado o botão 1 e 3 em simultâneo, que corresponde a perguntar ao utilizador se pretende apagar o documento de texto que guarda as decisões tomadas por este, e desse modo alterá-las e por fim a rotina que permite desligar o sinal sonoro respeitante à data de manutenção, pressionando simultaneamente o botão 1 e 2.

Figura 3.38 – Fração do código respetivo ao *loop*.

A rotina com o nome “Manutencao ()”, que aparece na Figura 3.39, contém um código curto e simples. O objetivo desta rotina destina-se apenas ao técnico que instalará o equipamento.

```

//*****
void Manutencao() {
    t = rtc.getTime();
    setTime(t.hour, t.min, t.sec, t.date, t.mon, t.year);
    if (buzzer == true) {
        if (t.date == 06 && t.mon == 01 && t.year == 2017) {
            tone(13, 2400, 1000);
        } else {
            noTone(13);
        }
    }
}
//*****

```

Figura 3.39 – Fração do código respetivo da rotina “Manutencao ()”.

Como é possível constatar pela Figura 3.39, o técnico digita a data para a próxima revisão e quando o dia atual for essa data, é tocado um alarme interruptamente. Desta forma o utilizador sabe quando deve chamar o técnico, ou quando este vai aparecer.

Para desligar o aviso sonoro, basta pressionar o botão 1 e 2 em simultâneo. Para que seja possível esta interação, existe a rotina “Botaole2Reset ()”. Como é fácil de perceber a partir da Figura 3.40, estes botões podem ser pressionados sem nenhuma sequência previsível, o que significa que a sua rotina tem de estar sempre disponível, ou seja, na função *loop* (Figura 3.38).

```

//*****
void Botaole2Reset() {
    if (digitalRead(17) == LOW && digitalRead(16) == LOW) {
        buzzer = false;
        Manutencao();
    }
}
//*****

```

Figura 3.40 – Fração do código respetivo da rotina “Botaole2Reset ()”.

É possível verificar, pela Figura 3.40, a simplicidade do código empregue. É utilizado uma variável cujo nome é “*buzzer*”, para identificar quando o sinal sonoro deve estar ativo ou inativo, mediante seja atuado, pois encontra-se na data para tal ou quando é pressionados os botões para desligar o sinal sonoro.

### 3.5. Protótipo Funcional

Segundo um simulador 3D, o resultado final do PCB tem o aspeto apresentado na Figura 3.41 e na Figura 3.42.

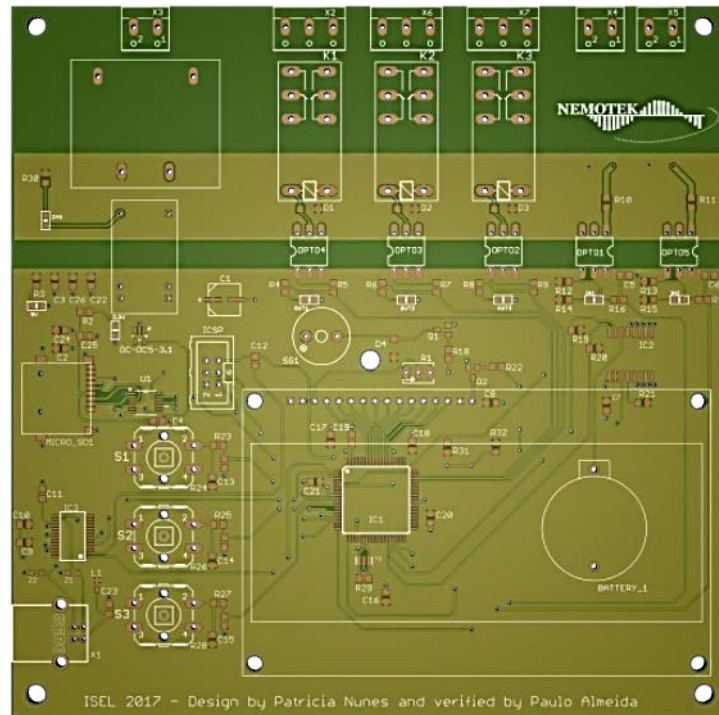


Figura 3.41 – Aspeto da *layer* de cima do PCB pelo simulador 3D [42].

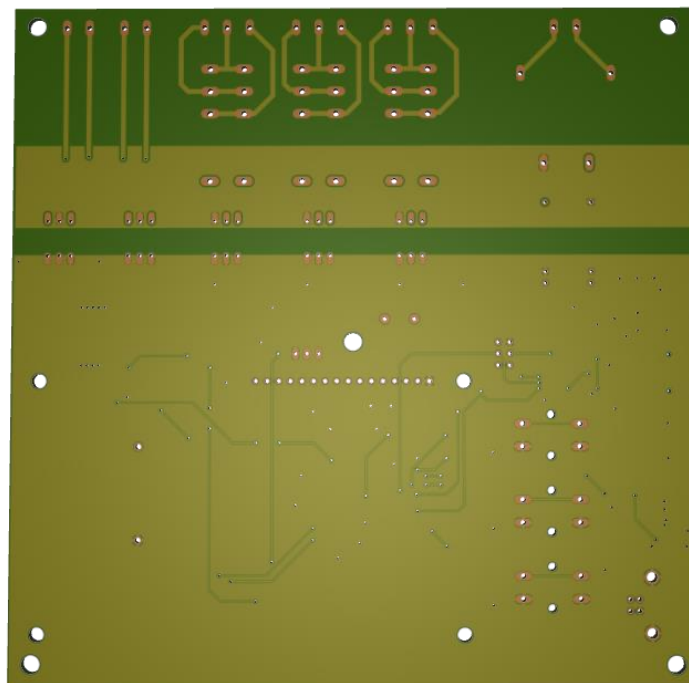


Figura 3.42 – Aspeto da *layer* de baixo do PCB pelo simulador 3D [42].

Como se pode verificar pela comparação das Figura 3.41 e na Figura 3.42 com as Figura 3.43 e Figura 3.44, o aspeto do PCB do simulador é bastante semelhante ao da realidade.

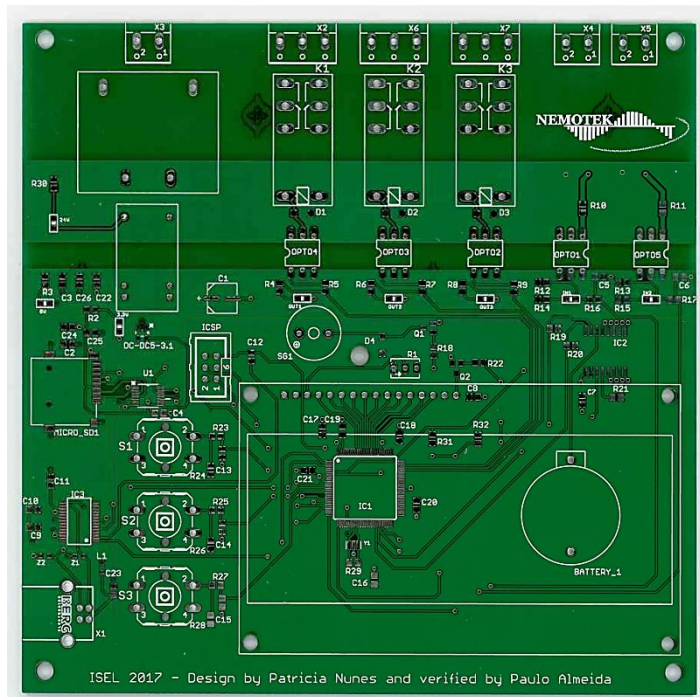


Figura 3.43 – Aspeto da *layer* de cima do PCB real.

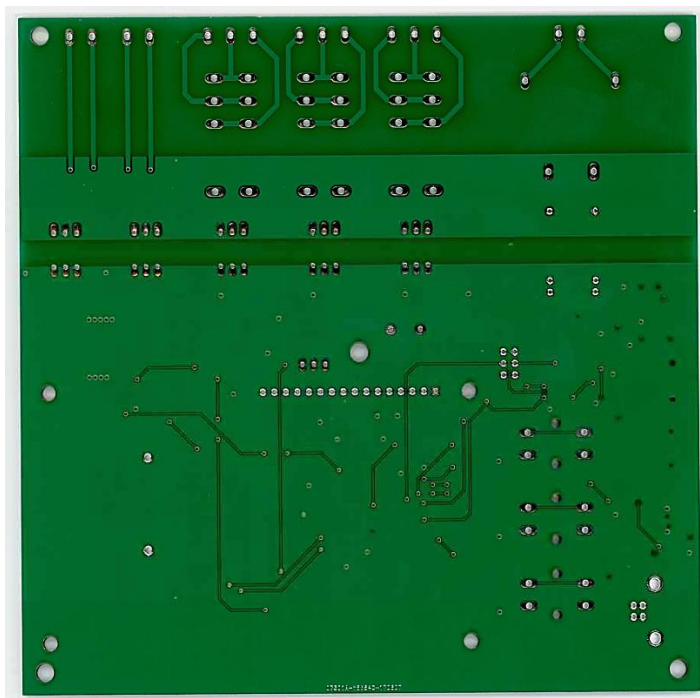


Figura 3.44 – Aspeto da *layer* de baixo do PCB real.

A concretização final do *hardware* necessita da soldadura dos componentes ao PCB, apresentado na Figuras 3.45 e na Figura 3.46.

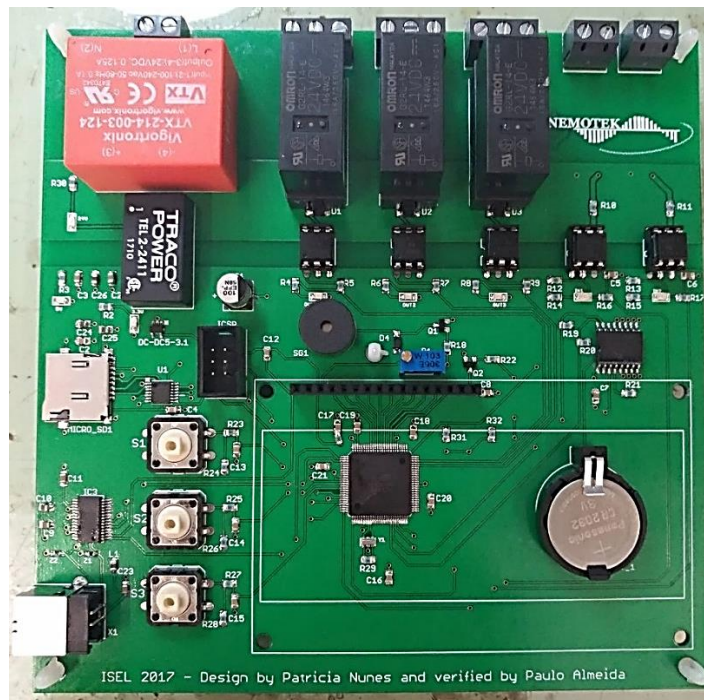


Figura 3.45 – Aspeto do PCB com os componentes inseridos, sem o LCD colocado (placa eletrónica).

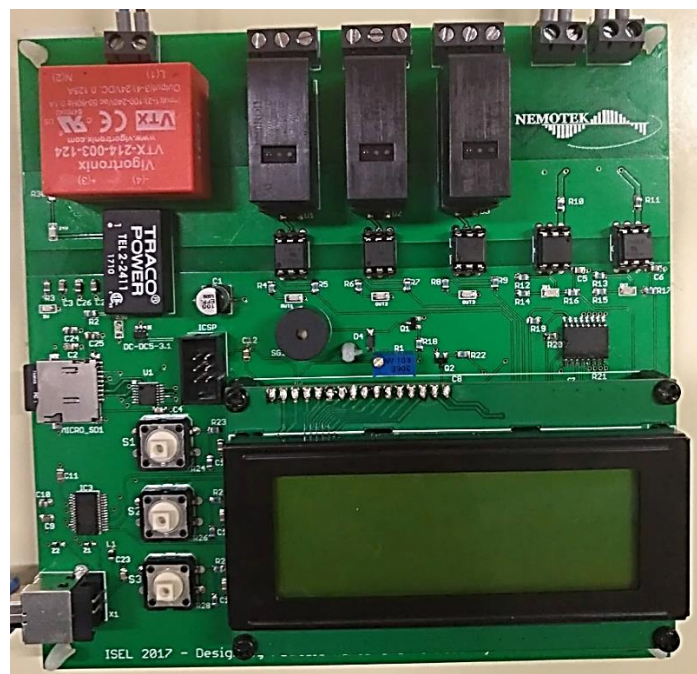


Figura 3.46 – Aspeto do PCB com os componentes inseridos, com o LCD colocado (placa eletrónica).

As Figuras 3.41 a 3.46, com maior pormenor podem ser encontradas no Apêndice I.

### ***3.6. Análise de custos***

Relativamente aos custos do equipamento desenvolvido, será meramente uma abordagem qualitativa, não especificando os valores em euros envolvidos, tanto na solução criada como na existente. Esta decisão foi tomada de forma a não comprometer a empresa envolvida no projeto.

Esta análise de custos tem o intuito de extrair informações que podem auxiliar numa tomada de decisão, pois um dos objetivos do presente projeto consiste em compreender se a solução criada é viável economicamente, em relação à solução já existente e utilizada pela NEMOTEK.

Desta forma é importante realçar que o principal objetivo do presente projeto é criar um protótipo funcional, como tal não se teve em consideração a otimização do custo nem do tamanho do equipamento. No entanto foi possível verificar que na produção em quantidades de 100 unidades, o preço de produção por unidade do sistema desenvolvido é competitivo face aos preços praticados pela empresa que fornece a solução utilizada atualmente pela Empresa.

É expetável, com base na análise efetuada, que a industrialização deste produto (otimização em termos de custo, dimensões e certificações), venha a resultar num equipamento que represente uma mais valia económica, face ao atualmente utilizado pela Empresa.

# Capítulo 4

## Resultados Experimentais

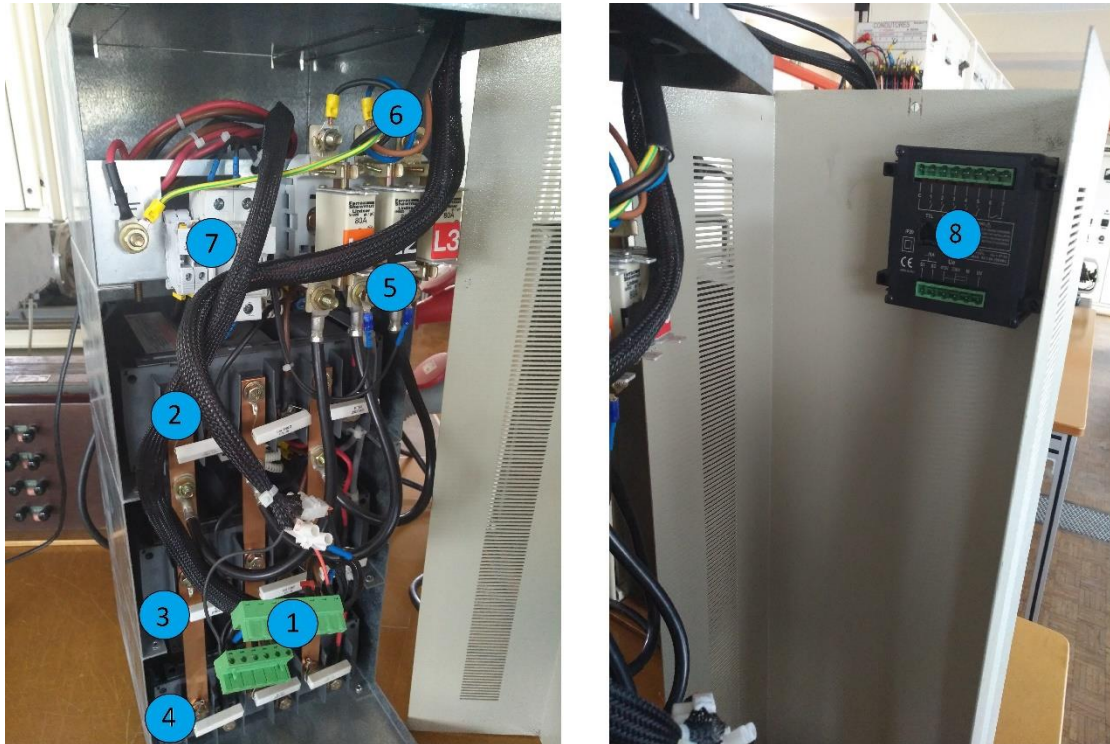
**Resumo:** *Neste Capítulo é especificado os resultados experimentais obtidos, onde são expostos os problemas que surgiram para o correto funcionamento do protótipo, as soluções desenvolvidas para estes problemas e demonstrações do correto funcionamento do código desenvolvido.*



## 4.1. Resultados Experimentais

No presente tópicos serão apresentados os resultados experimentais obtidos através do *hardware* e do *software* implementado na consecução do equipamento. Os ensaios realizaram-se no estabelecimento de ensino ISEL, numa BC fornecida pela Empresa envolvida no projeto.

A BC fornecida pela empresa é apresentada na Figura 4.1.



- 1 - Conector da alimentação dos escalões para o Relé Varimétrico
- 2,3,4 - Escalões com um contactor cada um
- 5 - Alimentação trifásica (3 fases)
- 6 - Conexão à rede elétrica a partir de uma tomada trifásica de 16 A
- 7 - Proteção por fusíveis
- 8 - Relé Varimétrico

Figura 4.1 – Interior da BC.

Para efeitos de testes, foi optado por funcionar com uma das entradas como alarme e a outra como controlo. Em relação às saídas, escolheu-se funcionar com apenas duas, uma com a função de ligar e desligar a BC a partir da entrada com a função de controlo e pelo horário, a outra saída para o acionamento de um escalão, obviamente a partir da entrada com a função

de controlo. O equipamento foi colocado na parte exterior da BC, como apresentado na Figura 4.2.

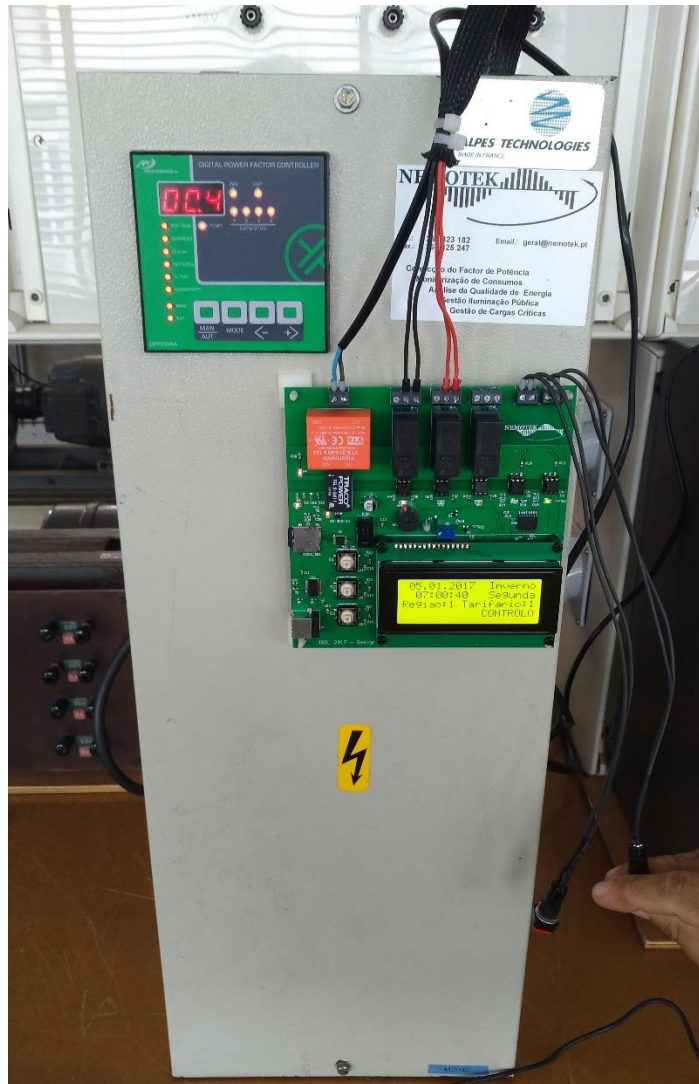


Figura 4.2 – Colocação do equipamento na BC.

Para verificar o correto funcionamento, a saída 1 foi ligada ao RV para o acionamento da BC, enquanto para a saída 2 foi necessário desconectar um dos escalões do RV e liga-lo à saída 2, para dessa forma se poder controlar um escalão diretamente a partir da entrada.

Verificou-se na fase de testes que não funcionava o contacto normalmente aberto (NA) dos relés associados às saídas. Os relés escolhidos contêm um contacto central que é o comum, como apresentado na Figura 4.3. Quando o relé está com a bobina desenergizada, o contacto comum C faz conexão com o contacto normalmente fechado (NF), mantendo o circuito fechado. Quando é energizada a bobina, o contacto C encosta-se ao contacto NA [43].

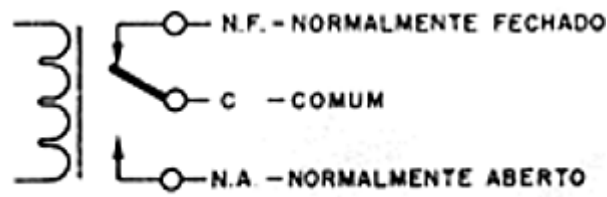


Figura 4.3 – Contactos do Relé [43].

Para solucionar o problema foi verificado inicialmente se o conector onde é colocado o relé, encontrava-se bem soldado ao PCB. Tendo sido apurado que o problema não surgia do conector, colocou-se em causa que o relé poderia estar danificado, como tal foi ligado a uma bateria externa e averiguado se os contactos funcionavam corretamente, mais uma vez, constatou-se que estavam em ótimas condições. Em seguida, ponderou-se que o diodo existente no circuito poderia ter sido soldado ao contrário, no entanto observou-se que este estava devidamente colocado. Optou-se por ir mais a fundo na questão, verificando através de um multímetro as tensões em todos os pontos e constatou-se que o problema provinha do diodo.

Analisou-se o catálogo do diodo selecionado e constatou-se que este suportava uma Tensão Inversa Máxima Repetitiva (VRRM) de 20 V, no entanto a tensão que está a ser utilizada no circuito é de 24 V. Significa que o funcionamento do diodo era semelhante a um diodo zener, ou seja, independentemente da corrente que circulava a tensão mantinha-se nos 20 V, o que não era suficiente para atracar o contacto NA. Assim sendo foi selecionado outro diodo com uma tensão VRRM superior.

Outro problema que se constatou foi o valor utilizado para a resistência associada ao circuito do *optocoupler*. O modelo básico de um *optocoupler* consiste num LED que produz luz infravermelha, e um dispositivo fotossensível semicondutor que é usado para detetar o feixe infravermelho emitido. A corrente que passa pelo LED de entrada que emite uma luz infravermelha contém uma intensidade proporcional ao sinal elétrico. Esta luz emitida cai sobre a base do foto-transistor, que vai conduzir de forma semelhante a um transístor bipolar normal [44].

O circuito do *optocoupler* de uma das saídas é apresentado na Figura 4.4, inicialmente foi dimensionado uma resistência de 1 k $\Omega$  (tópico 3.3.2.) na entrada do sinal de corrente colocado no LED infravermelho, no entanto, verificou-se que este valor de resistência estava

a limitar muito a corrente para o transístor, pois quanto mais alto o valor da resistência menor será a corrente e como tal será menor a luminosidade por parte do LED e consequentemente o transístor não estará a plena condução.

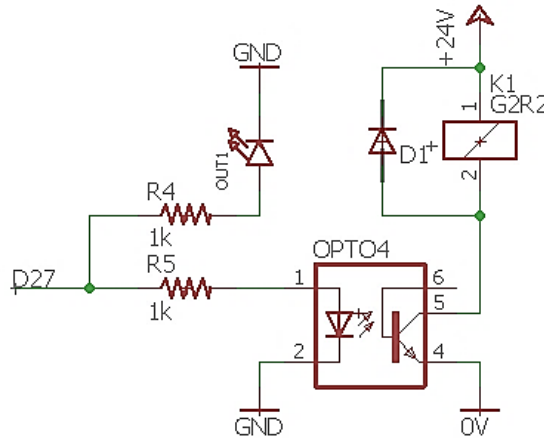


Figura 4.4 – Circuito do *optocoupler* de uma das saídas [35].

Foi necessário dimensionar novamente o valor da resistência para evitar o exposto anteriormente, como tal foi verificado os valores de corrente e tensão suportados pelo *optocoupler* selecionado (Anexo III). Utilizando a mesma equação, (3.1.), obtém-se:

$I_{OPTO}$  – Intensidade máxima de corrente à entrada do *optocoupler*, de 60 mA;

$U_{OPTO}$  – Tensão à entrada do *optocoupler*, de 5 V.

$$R = \frac{U_{OPTO}}{I_{OPTO}} = \frac{5}{0,06} \cong 83 \Omega \quad (4.1.)$$

O valor da resistência a colocar devia ser de 83  $\Omega$ , no entanto segundo as especificações do Arduino Mega, este suporta no máximo 20 mA e não 60 mA [4]. Para obter a corrente máxima que o microcontrolador consegue entregar ao *optocoupler*, é realizada a seguinte Lei de *Kirchhoff* das Correntes, tendo em vista o circuito da Figura 4.4.

$$I_{LED} = \frac{U_{OPTO} - U_{LED}(out1)}{R4} = \frac{5 - 2,1}{1000} = 2,9 \text{ mA} \quad (4.2.)$$

$$I''_{\text{OPTO}} = I'_{\text{OPTO}} - I_{\text{LED}} = 20 \text{ mA} - 2,9 \text{ mA} = 17,1 \text{ mA} \quad (4.3.)$$

Em que:

$U_{\text{LED}}(\text{out1})$  – Tensão do LED existente no circuito da saída 1 (Figura 4.4);

$R_4$  – Resistência associada ao LED existente no circuito da saída 1 (Figura 4.4);

$I_{\text{LED}}$  – Intensidade de corrente do LED existente no circuito da saída 1 (Figura 4.4);

$I'_{\text{OPTO}}$  – Intensidade de corrente máxima suportada pelo microcontrolador utilizado;

$I''_{\text{OPTO}}$  – Intensidade de corrente máxima suportada à entrada do *optocoupler*, tendo em conta o microcontrolador utilizado.

Como é possível verificar pelos cálculos efetuados o máximo que o circuito permite fornecer ao *optocoupler* é de 17,1 mA. Assim sendo a resistência a colocar à entrada do *optocoupler* é:

$$R_{\text{OPTO}} = \frac{U_{\text{OPTO}} - U_{\text{LED OPTO}}}{I''_{\text{OPTO}}} = \frac{5 - 1,3}{0,0171} \cong 22 \Omega \quad (4.4.)$$

Em que:

$U_{\text{LED OPTO}}$  – Tensão do LED infravermelho existente no *optocoupler*;

$R_{\text{OPTO}}$  – Valor de Resistência mínimo para a entrada do *optocoupler*.

Em suma, o valor mínimo que a resistência à entrada do *optocoupler*, de modo a não danificar o sistema é de 22  $\Omega$ . Tendo em conta este facto foi selecionado uma resistência de 22  $\Omega$ . Após se solucionar estes dois problemas, observou-se que o equipamento se encontrava a funcionar segundo o expetável, realizando todas as tarefas nos momentos certos, sem que para isso fosse necessário revestir o equipamento com alguma blindagem.

### 4.1.1. Código a funcionar

Na Figura 4.5 é revelado como se encontra estruturado o código referente a uma questão realizada ao utilizador e como esta aparece exposta no LCD (Figura 4.6).

```

//****Seleção da Região*****
lcd.setCursor(0, 0);
lcd.print("Selecione a Regiao ");
lcd.setCursor(0, 1);
lcd.print("1-Portugal");
lcd.setCursor(0, 2);
lcd.print("2-Acores");
lcd.setCursor(0, 3);
lcd.print("3-Madeira");

Regiao = ReadButtons(3);

Alarm.delay(1000);

```

Figura 4.5 – Rotina para a implementação das decisões do utilizador, seleção da Região.

O LCD é constituído por 4 linhas e 20 colunas, portanto, a função “lcd.setCursor (coluna, linha)”, tem o intuito de localizar o texto referido seguidamente na posição adequada. Portanto, no LCD aparece o texto descrito da Figura 4.5, iniciando sempre na primeira coluna, ou seja, sem espaçamento, e cada frase na respetiva linha. É necessário ter em conta que cada frase pode conter no máximo 20 caracteres, com espaços incluídos, caso contrário a formatação do texto dispõem-se incorretamente. A visualização da seleção da Região no LCD é apresentada na Figura 4.6.



Figura 4.6 – Visualização da seleção da Região no LCD.

Após se ter selecionado as funções destinadas às entradas e saídas, foi verificado se o protótipo se encontrava a funcionar corretamente. Como a Figura 4.7 e a Figura 4.8

apresentam foi optado que a entrada 1 funcionasse com a função alarme e a entrada 2 com a função controlo. Sendo que esta última ativaria duas saídas.

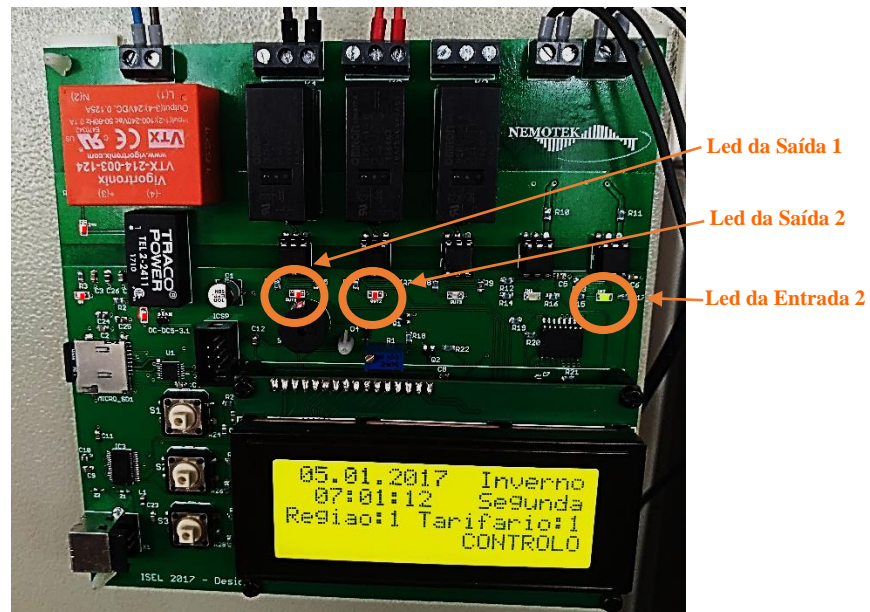


Figura 4.7 – Visualização da ativação da entrada 2 (controlo).

Como verificado pela Figura 4.7, no LCD encontra-se sempre exposto a data, a hora, o dia da semana, a seleção escolhida para a Região e o Tarifário, e o período de hora legal (Inverno ou Verão). A Figura 4.7, ilustra a situação em que a entrada 2 está ativada. Nessa situação é escrito a função correspondente à entrada no LCD, é acionado um sinal visual (LED) que refere que a entrada se encontra ativa e nesta situação que a saída 1 e 2 estão ativas, visível igualmente a partir de um LED. A visualização da ativação da entrada 1 (alarme) é apresentada na Figura 4.8.



Figura 4.8 – Visualização da ativação da entrada 1 (alarme).

Na situação em que é a entrada 1 que está ativa nenhuma saída pode estar ativa. Como a Figura 4.8 demonstra, efetivamente nenhuma saída se encontra ativa, é escrito no LCD a função correspondente a esta entrada, neste caso “ALARME” e é visível o sinal luminoso que confirma que a entrada 1 está ativa.

Existem mais 3 sinais luminosos, no canto superior esquerdo da Figura 4.8 que confirmam que existe tensão na secção dos 24 VDC, nos 5 VDC e nos 3,3 VDC.

# Capítulo 5

## Conclusões e Desenvolvimentos

### Futuros

**Resumo:** *Neste Capítulo são apresentadas as conclusões obtidas através do estudo e elaboração do protótipo funcional. São também destacados possíveis desenvolvimentos futuros no âmbito da otimização do produto tanto a nível de software como de hardware.*



## 5.1. Conclusões

Terminada a elaboração da presente dissertação, conclui-se que os objetivos principais propostos foram concretizados. Uma vez que após a realização de um estudo teórico sobre o microcontrolador Arduino, foi possível estruturar, desenvolver e implementar um equipamento que utiliza este microcontrolador e executa as tarefas desejadas e impostas, pela empresa envolvida no projeto. Este equipamento dispõe de duas soluções simples e fáceis de utilização como interface entre o equipamento e o utilizador. Uma das soluções utiliza um LCD e os botões incorporados no equipamento para obter as opções de configuração do utilizador, a outra solução utiliza um ficheiro em *Excel* programado para transformar as opções de configuração do utilizador, em documentos de texto que serão posteriormente lidos pelo equipamento, através de um cartão SD.

É de aferir que apenas dois requisitos propostos pela empresa não foram passíveis de serem concretizados, sendo um deles um requisito físico e outro um requisito de controlo. Um dos requisitos físicos impostos pela NEMOTEK era que a alimentação do equipamento fosse a 400 VAC, no entanto o elaborado neste documento tem uma alimentação a 230 VAC. Para executar este requisito seria necessário utilizar um transformador, equipamento este, que não ia no intuito das expectativas da empresa, ou criar um conversor para o efeito, sendo que o conversor expandiria demasiado a presente dissertação. Foi tomada a decisão de colocar este tema como tópico de um desenvolvimento futuro. O requisito de controlo, que não foi executável, está relacionado com as temporizações atribuídas às saídas. O intuito implícito no documento fornecido pela Empresa (Anexo I), seria concretizar a opção de uma temporização por antecipação ou após a referência horária, no entanto por questões de elaboração de código, não foi possível realizar a temporização por antecipação, uma vez que o algoritmo desenvolvido não permitia tempos negativos, tendo sido por isso, substituído para a opção de 3 temporizações diferentes após a referência horária. Alternativas estas que são colocadas ao dispor do utilizador.

Com base nos resultados experimentais, pode-se concluir que a arquitetura utilizada e a execução do equipamento, desde o PCB até à soldagem dos componentes satisfazem as funções e definições impostas pelo algoritmo (código) formulado. O equipamento desenvolvido foi projetado para a possibilidade de desenvolvimentos futuros, como os que

serão enumerados posteriormente neste Capítulo e mais importante cumprir como solução alternativa à atual utilizada pela empresa no controlo auxiliar de uma BC.

Havendo necessidade de realizar um estudo e testar a possibilidade de concretização do projeto a partir da plataforma Arduino, o desenvolvimento do equipamento direcionou-se para a criação de um protótipo funcional. Não tendo sido por isso otimizado tanto na redução de custos, como dimensões do equipamento, no entanto segundo um estudo teórico julga-se possível que a solução após a sua otimização, satisfaça financeiramente as expetativas da NEMOTEK.

Sabendo que a colocação do equipamento será em ambiente industrial, teve-se em conta que a seleção dos componentes englobados no sistema suportasse perturbações harmónicas e variações térmicas. Já a partir dos resultados experimentais constatou-se que colocando o equipamento na parte exterior da BC sem qualquer outra estrutura externa que efetuasse uma proteção do equipamento em relação às interferências eletromagnéticas, o equipamento não pareceu ser afetado por estas. No entanto, tendo em conta que posteriormente o equipamento deverá ser colocado no interior da BC, seria importante elaborar uma estrutura física que lhe dê robustez para o meio industrial.

## ***5.2. Desenvolvimentos Futuros***

De acordo com os resultados experimentais obtidos, é viável proferir que o protótipo funcional elaborado é uma solução viável para as tarefas a que foi destinado, sendo por isso fácil de concluir que será interessante dar continuidade a este estudo e desenvolvimento deste protótipo, tornando-o num produto final. Como foi evidenciado na conclusão um dos desenvolvimentos futuros seria a concretização de um conversor compacto e de pequenas dimensões que satisfaça o requisito físico que não fora cumprido e mais importante uma otimização das dimensões do produto e do custo.

Seria importante também executar um estudo mais pormenorizado que o realizado no Apêndice II, realizando os devidos cálculos teóricos e testes laboratoriais ao protótipo, por forma a concluir que blindagem se adequa ao mesmo, tornando-o robusto para o meio industrial.

O algoritmo desenvolvido no protótipo funcional, não foi também otimizado ou simplificado, como tal, numa investigação futura seria útil trabalhar no algoritmo e seria enriquecedor a inserção de outras funções e opções tais como: a inclusão de uma medição de corrente (0 a 5 A), de forma a criar cenários para as saídas em função de valores de consumo da instalação. Outra vertente a poder seguir seria a construção de uma interface gráfica dos consumos praticados durante um certo período de tempo, a definir, onde incluísse valores de FP, ou até mesmo a possibilidade de visualizar potências ativas, reativas e aparentes.

Uma melhoria que poderia ampliar a proteção do sistema, seria a medição da temperatura interna de atuação das saídas ou ainda a medição do valor de tensão composta por forma a gerir as saídas em função desses valores, criando cenários de alarme ou até de proteção, garantido que o sistema se encontrava a funcionar corretamente e em segurança.

Por fim, outra opção como incorporação ao equipamento, seria uma porta de comunicação RS485 como integração em sistemas de gestão técnica.



## Bibliografia

- [1] Comité de Distribuição de Energia Elétrica. Manual de Orientação aos consumidores- Energia Reativa Excedente.
- [2] E. D. P, “Perguntas frequentes - faturação de energia reactiva.” 2012.
- [3] Voltimum, “Eficiência energética na indústria | 2.ª PARTE - ANÁLISE DE FACTURAS DE ENERGIA ELÉCTRICA.pdf.” 2013.
- [4] Arduino. Disponível em <https://www.arduino.cc>. [Consultado em 22/05/2017].
- [5] NEMOTEK - Gestão Técnica Centralizada, Lda. Disponível em <http://www.nemotek.pt>. [Consultado em 20/04/2017].
- [6] Pimenta de Freitas, Tiago Joel (2014). Qualidade de serviço em redes elétricas inteligentes. ipb.
- [7] Leite, Carlos Miguel (2011). Compensação da Energia Reactiva na Rede de Distribuição. FEUP.
- [8] ERSE- Entidade Reguladora dos Serviços Energéticos, "Despacho n.º 7253/2010", Abril de 2010.
- [9] ERSE- Entidade Reguladora dos Serviços Energéticos, "Despacho n.º 12605/2010", Agosto de 2010.
- [10] ERSE - Entidade Reguladora dos Serviços Energéticos, “Nota informativa sobre as novas regras de faturação da energia reactiva”, Abril de 2010.
- [11] Marques, Gonçalo (2015). Compensação da Energia Reactiva com Minimização de Perturbações sobre a Rede Elétrica. ISEL.
- [12] Soleymani, Soodabeh et al. (2017). *Detection and rapid improvement of the transient regime switching in ASD equipment by wavelet transform*. IJCSNS.
- [13] Ribeiro de Sá, André Fernando (2016). Guia de Aplicações de Gestão de Energia e Eficiência Energética - 3ª edição. PUBLINDUSTRIA.
- [14] Power Systems & Controls. Disponível em <http://www.pscpower.com/psc-power-quality-testing>. [Consultado em 30/04/2017].

- [15] Palma, João (2008). Accionamentos Electromecânicos de Velocidade Variável - 2.<sup>a</sup> Edição. Fundação Calouste Gulbenkian. cap.8, p.377-378.
- [16] Bim, Edson (2009). *Máquinas Elétricas e Acionamento*. Elsevier. cap.9, p.328.
- [17] WEG - Compensadores síncronos rotativos. Disponível em <http://ecatalog.weg.net/files/wegnet/WEG-compensadores-sincronos-rotativos-50041259-catalogo-portugues-br.pdf>. [Consultado em 02/09/2017].
- [18] Ferreira Rodrigues, Justino Miguel (2010). Análise e Modelação de Dispositivos FACTS Aplicados nas Redes de Transporte e de Distribuição de Energia Eléctrica. FEUP.
- [19] Frizzera Encarnação, Lucas (2009). Compensador Síncrono Estático Multinível em Média Tensão para Sistemas de Distribuição. COPPE/UFRJ.
- [20] Prity Bisen e Amit Shrivastava (2013). *Comparison between SVC and STATCOM FACTS Devices for Power System Stability Enhancement*. International Journal on Emerging Technologies 4(2).
- [21] Vairinhos de Lima, Francisco Manuel (2012). Reguladores Electrónicos de Tensão para Transformadores de Baixa Tensão. IST.
- [22] Campos Teixeira, André. Interface de um Gerador Eólico com a Rede Eléctrica. Universidade do Minho.
- [23] Sotoodeh, Pedram and Douglas, Ruth (2012). *Design and Control of a New Single-phase Inverter For Grid-connected Renewable Energy Systems*. IEEE.
- [24] Melo, João e Leandro, Gideon (2011). Mini Curso Microcontrolador pic 18F452/PROTEUS. Universidade Federal do Paraná.
- [25] OP SERVICES. Disponível em <https://www.opservices.com.br/o-que-e-o-arduino>. [Consultado em 24/05/2017].
- [26] Controllino. Disponível em <https://controllino.biz> [Consultado em 25/05/2017].
- [27] Ferreira, Jorge (1994). Virtualização de autómatos programáveis. Universidade de Aveiro.
- [28] Portal ERSE. Disponível em <http://www.erse.pt/pt/electricidade/tarifaseprecos/periodoshorarios/Paginas/default.asp>

- x [Consultado em 13/06/2017].
- [29] Alteração do Regime da hora legal em Portugal Continental, "Decreto-lei 17/96", 8 de Março de 1996.
- [30] Rodrigues, Paulo (2011). Gestão de energia reactiva na rede de distribuição da EDP. FEUP.
- [31] Computer Solutions Ltd - USB protocol. Disponível em [http://www.computer-solutions.co.uk/info/Embedded\\_tutorials/usb\\_tutorial.htm](http://www.computer-solutions.co.uk/info/Embedded_tutorials/usb_tutorial.htm). [Consultado em 18/09/2017].
- [32] Tutoriais, Comparação Entre Protocolos de Comunicação Serial. Disponível em <https://www.robocore.net/tutoriais/comparacao-entre-protocolos-de-comunicacao-serial.html>. [Consultado em 20/07/2017].
- [33] Shields. Disponível em <https://www.sparkfun.com> [Consultado em 04/07/2017].
- [34] Vida de Silício, Robótica e Sistemas Digitais. Disponível em <http://blog.vidadesilicio.com.br/arduino/basico/usando-o-buzzer-com-arduino-transdutor-piezo-eletrico>. [Consultado em 20/07/2017].
- [35] *Eagle CAD Software*. Disponível em <https://www.autodesk.com/products/eagle/overview>. [Consultado em 02/09/2017].
- [36] Mehl, Ewaldo. Conceitos Fundamentais sobre Placas de Circuito Impresso.
- [37] Kundert, Ken (2004). *Power Supply Noise Reduction*. The Designer's Guide Community.
- [38] Schmitz, Tamara; Wong, Mike (2011). *Choosing and Using Bypass Capacitors*. Intersil.
- [39] IPC-2221 (1998). *Generic Standart on Printed Board Design*. From: Association Connecting Electronics Industries.
- [40] Soares, Vasco (2008). *APS-Aquisição e Processamento de Sinais*. Instituto Superior de Engenharia de Lisboa (ISEL).
- [41] Fernandes, João Miguel (2000). MIDAS: Metodologia Orientada ao Objecto para Desenvolvimento de Sistemas Embebidos. Universidade do Minho.

- [42] 3D Gerber Viewer. Disponível em <http://mayhewlabs.com/webGerber> [Consultado em 13/07/2017].
- [43] C.Braga, Newton (2012). Relés, Circuitos e Aplicações. Metaltext.
- [44] *Eletronics Tutorials*. Disponível em <http://www.electronicstutorials.ws/blog/optocoupler.html>. [Consultado em 21/08/2017].
- [45] Fabricante ITEAD. Disponível em <https://www.itead.cc> [Consultado em 12/07/2017].
- [46] Online Gerber Viewer. Disponível em <http://www.gerber-viewer.com>. [Consultado em 13/07/2017].
- [47] Requisitos no *design* do PCB. Disponível em <http://support.iteadstudio.com/support/solutions/articles/1000156316-requirements-on-the-pcb-design>. [Consultado em 20/07/2017].
- [48] Requisitos para os *Gerbers*. Disponível em <http://support.iteadstudio.com/support/solutions/articles/1000156317-gerber-files-requirements>. [Consultado em 20/07/2017].
- [49] Pinheiro, José (2004). Interferência Eletromagnética em Redes de Computadores.
- [50] VIVAGE Process Instruments Exceeding Automation Limits - EMI Interferência Eletromagnética em Instalações Industriais e Muito Mais. Disponível em <http://www.vivaceinstruments.com.br/pt/artigo/emi-interferencia-eletromagnetica-em-instalacoes-industriais-e-muito-mais>. [Consultado em 13/09/2017].
- [51] França, Gleisson (2013). Desenvolvimento de Compensador Série para Mitigação Harmônica Distribuída e Correção Dinâmica de Fator de Potência. Universidade Federal de Minas Gerais.
- [52] IEC 61000-3-2 (2014). *Electromagnetic compatibility: Limits for harmonic current emissions*.
- [53] N.C. Batista, R. Melício, J.C.O Matias, J.P.S. Catalão, "ZigBee standard in the creation of wireless networks for advanced metering infrastructures", in: Proceedings of the 16th IEEE Mediterranean Electrotechnical Conference (MELECON'2012), pp. 220–223, Medina Yasmine Hammamet, Tunisia, 25–28 March 2012.
- [54] N.C. Batista, R. Melício, J.C.O Matias, J.P.S. Catalão, "ZigBee wireless area network

for home automation and energy management: field trials and installation approaches",  
in: Proceedings of the 3rd IEEE PES Europe Conference on Innovative Smart Grid  
Technologies — ISGT 2012, pp. 1–5, Berlin, Germany, 14–17 October 2012.



# Apêndice I

## Dossier de Fabricação do PCB



Para a fabricação de uma placa de circuito impresso (PCB), são necessárias 6 etapas:

- 1) Esquemas;
- 2) BOM;
- 3) *Layout*;
- 4) *Routing*;
- 5) DRC;
- 6) *Gerbers*.

## **1. Esquemas**

Os esquemas encontram-se no final do Apêndice I, distribuídos pelos seguintes desenhos:

Desenho 1 - Circuitos de Alimentação;

Desenho 2 - Circuito do Micro-SD;

Desenho 3 - Circuito das Saídas;

Desenho 4 - Circuito das Entradas;

Desenho 5 - Circuito do *Buzzer*;

Desenho 6 - Circuito do RTC;

Desenho 7 - Circuito do LCD;

Desenho 8 - Circuito do USB;

Desenho 9 - Circuito dos Botões;

Desenho 10 - Circuito do Microcontrolador.

## **2. BOM**

Na tabela seguinte encontra-se a BOM, que consiste num orçamento do PCB. Por questões de sigilo com a empresa envolvente, não serão revelados os custos dos dispositivos.

Tabela Ap1.1 – BOM, Lista do Material com o fabricante selecionado.

| Quantidade | Dispositivo/Valor/Nome Eagle               | Designação   | Vendedor | Referência          |
|------------|--|--|----------|---------------------|
| 1          | Ficha IDC(3X2M)(PINHD-2X3)                 | ICSP   | Mauser   | 011-0800            |
| 5          | StandOff                                   |  | Mauser   | 096-0861            |
| 3          | Header (W237-102)                          | X3, X4, X5   | Mauser   | 011-0911            |
| 3          | Header (W237-103)                          | X2, X6, X7   | Mauser   | 011-0913            |
| 1          | Conetor LCD                                |  | Mauser   | 012-0083            |
| 3          | Botões(40-XX)                              | S1, S2, S3   | Farnell  | 176435              |
| 5          | Optocoupler (4N28)                         | OPTO1,<br>OPTO2,<br>OPTO3,<br>OPTO4,<br>OPTO5  | Farnell  | 1652509             |
| 8          | LED 0805                                   | 3.3V1, 5V1,<br>24V1, IN1,<br>IN2, OUT1,<br>OUT2,<br>OUT3   | Farnell  | 2112118             |
| 1          | Condensador (100uF)(EEE-FPE101XAP-CASE-D8) | C1   | Farnell  | 1539488             |
| 1          | Pilha (CR2032)                             |  | Farnell  | 2065171             |
| 2          | VARISTORCN0603                             | Z1, Z2   | Farnell  | 1838966             |
| 1          | BUZZER(F/QMBIII)                           | SG1  | Farnell  | 2215079             |
| 1          | Conetor USB(PN61729-S)                     | X1   | Farnell  | 1076665             |
| 4          | 12mm Spacer                                |  | Farnell  | 1336156             |
| 1          | Conversor(VTX-214-003)                     | AC-DC1   | Farnell  | 2401038             |
| 4          | Diodo(MBR0520LT)                           | D1, D2, D3,<br>D4  | Farnell  | 9556915             |
| 1          | Indutância (L-USL2012C)                    | L1   | Mouser   | 875-MI0805K400R-10  |
| 1          | Potenciometro(10k)(TRIM_EU-B64Y)           | R1   | Mouser   | 571-2-1761603-1     |
| 19         | Condensador (0.1uF)(C-USC0805)             | C2, C3, C4,<br>C5, C6, C7,<br>C8, C9, C11,<br>C12, C13,<br>C14, C15,<br>C16, C17,<br>C18, C19,<br>C20, C21 | Mouser   | 80-C0805C104J5RAUTO |
| 19         | Reistência(10k)(R-US_R0805)                | R10, R11,<br>R12, R13,<br>R16, R17,<br>R19, R20,<br>R21, R23,<br>R24, R25,<br>R26, R27,<br>R28, R31,       | Mouser   | 660-RK73H2ATTD1002F |

|    |                                       |  |          |                      |
|----|---------------------------------------|--|----------|----------------------|
|    |                                       | R32, R33   |          |                      |
| 1  | Condensador (10nF)(C-USC0805)         | C23  | Mouser   | 80-C0805C103K5RAUTO  |
| 1  | Condensador (10uF)(C-USC0805)         | C22  | Mouser   | 80-C0805C106K3P      |
| 1  | Reistência(1M)(R-US_R0805)            | R29  | Mouser   | 660-RK73H2ATTD1004F  |
| 11 | Reistência(1k)(R-US_R0805)            | R3, R4, R5,<br>R6, R7, R8,<br>R9, R14,<br>R15, R18,<br>R22 | Mouser   | 660-RK73H2ATTD1001F  |
| 3  | Condensador (1uF)(C-USC0805)          | C24, C25,<br>C26   | Mouser   | 80-C0805C105K3P      |
| 2  | Mosfet (2N7002)(BSS123)               | Q1, Q2   | Mouser   | 512-2N7002           |
| 1  | Condensador (4.7uF)(C-USC0805)        | C10  | Mouser   | 80-C0805C475K3P      |
| 1  | Reistência(420)(R-US_R0805)           | R2   | Mouser   | 660-RK73H2ATTD4220F  |
| 1  | Reistência(6.6k)(R-US_R0805)          | R30  | Mouser   | 660-RK73H2ATTD6651F  |
| 1  | Conversor de nível(ADG3304BRUZ)       | U1   | Mouser   | 584-ADG3304BRUZ      |
| 1  | Conversor (ADP162AUJZ-3.3)            | DC-DC5-3.1   | Mouser   | 584-ADP162AUJZ-3.3R7 |
| 1  | Microcontrolador(ATMEGA25602560-16AU) | IC1  | Mouser   | 556-ATMEGA2560-16AU  |
| 1  | BATTERY_1236_C                        | BATTERY_1  | Mouser   | 534-1066             |
| 1  | RESONATOR                             | Y1   | Mouser   | 81-CSTCE16M0V53-R0   |
| 1  | RTC(DS3231/SO)                        | IC2  | Mouser   | 700-DS3231SN#        |
| 1  | USB interface(FT232RL)                | IC3  | Mouser   | 895-FT232RL          |
| 3  | Header(G2R2)                          | K1, K2, K3   | Mouser   | 653-G2RL-14-E-DC24   |
| 3  | Socket                                |  | Mouser   | 655-RP78602          |
| 1  | LCD-HD447804-HOLES20X4-4HOLES         | U\$1   | Mouser   | 992-LCD-20X4Y        |
| 1  | Conversor(TEL_2-2422)                 | DC-DC24-1  | Mouser   | 495-TEL-2-2411       |
| 1  | MICROSD_PP                            | MICRO_SD1  | Dipmicro | DE1594               |
| 1  | PCB                                   |  | ITEAD    |                      |

### 3. Layout

O layout encontra-se na Figura Ap1.1.

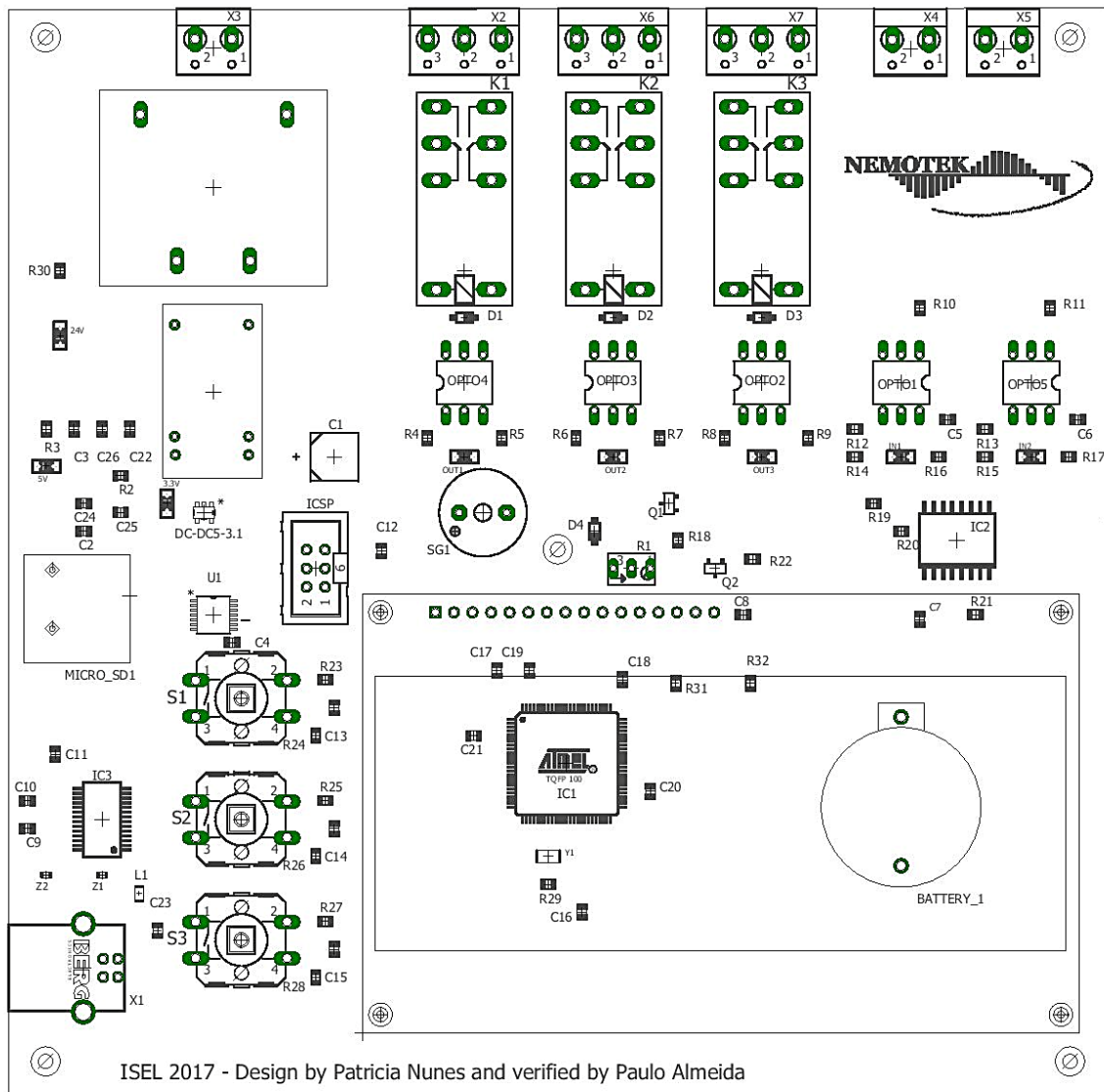


Figura Ap1.1 - Layout do PCB [35].

## 4. Routing

A Figura Ap1.2 representa o Routing.

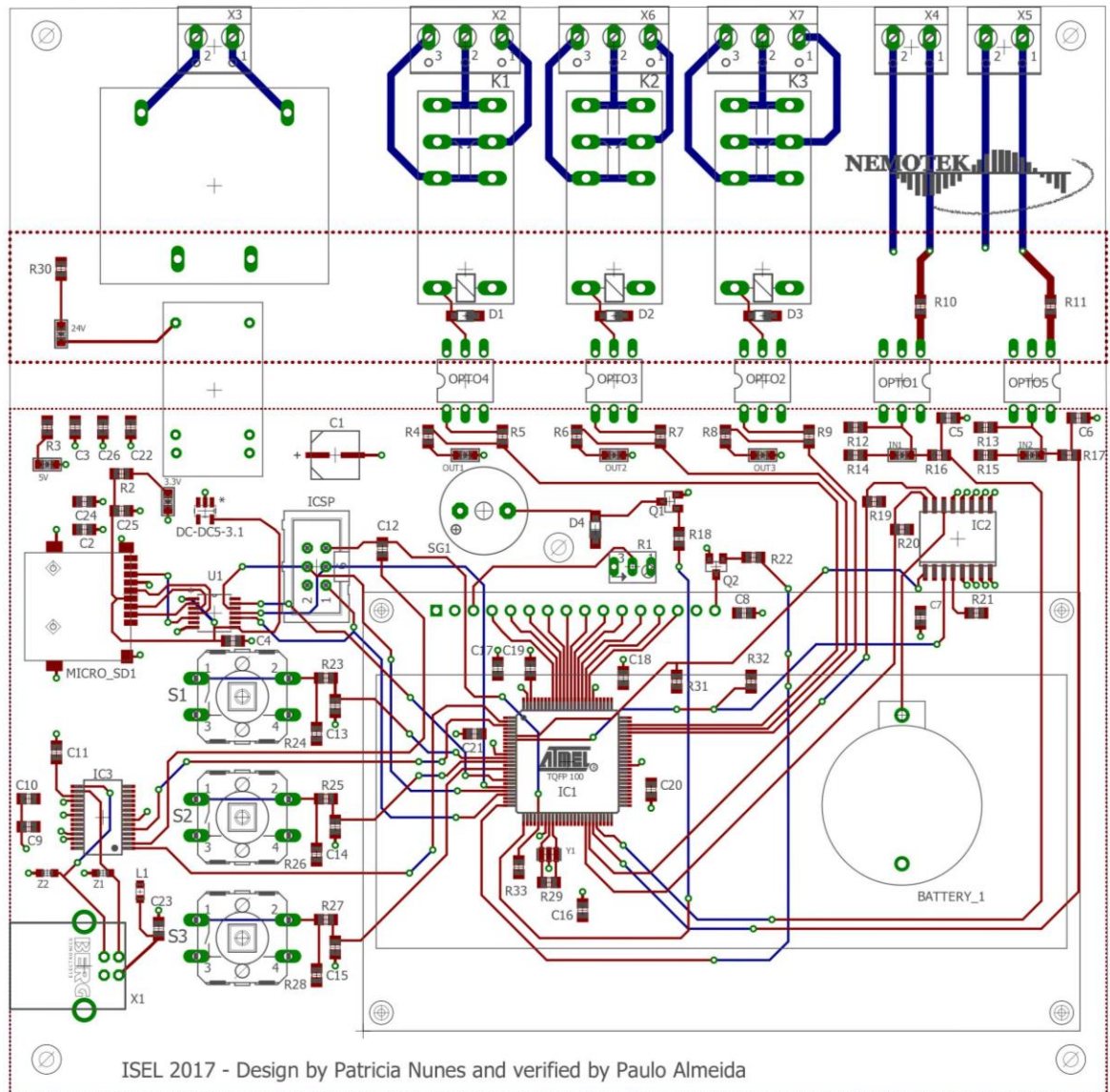


Figura Ap1.2 - Routing do PCB [35].

## 5. DRC

Segundo o fabricante escolhido, ITEAD, existem os seguintes requisitos a cumprir:

Requisitos no design do PCB [47]:

- Utilizar os comprimentos e larguras do PCB disponíveis pelo fabricante, independentemente do design praticado posteriormente;
- A largura mínima da linha para a ligação entre pinos (*silkscreen*) é de 6 mm;
- A altura mínima do texto existente no PCB é de 32 mm;

- As vias aceites são apenas aquelas que vão desde uma camada à outra, como demonstra a figura seguinte;

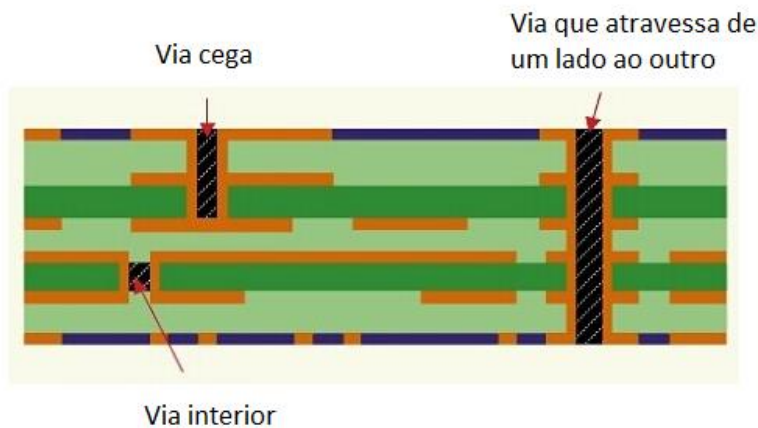


Figura Ap1.3 - Tipo de vias que existentes (traduzido para português) [45].

Em que via cega (*Blind Via*), é definida como aquela via que inicia numa camada externa, mas termina numa camada interna. A via interior (*Buried via*), é definida como a via que existe apenas entre duas camadas internas do PCB, como tal não começa ou termina em nenhuma externa. Por fim, a via que atravessa de um lado ou outro (*Through via*), que como o nome indica, é definida como a via que inicia numa camada externa e termina na outra externa.

O fabricante escolhido apenas permite vias que atravessam de um lado ao outro, sendo por isso apenas utilizadas estas no projeto.

#### Requisitos para os Gerbers[48]:

- O Gerbers necessários para arquivo de um PCB de 1 e 2 camadas:
  - GBL;
  - GTL;
  - GBS;
  - GTS;
  - GBO;
  - GTO;
  - O documento com a extensão .txt.

## 6. Gerbers

As Figuras Ap1.4 a Ap1.10, representam as *layers* dos ficheiros *Gerbers* provenientes de um simulador online.

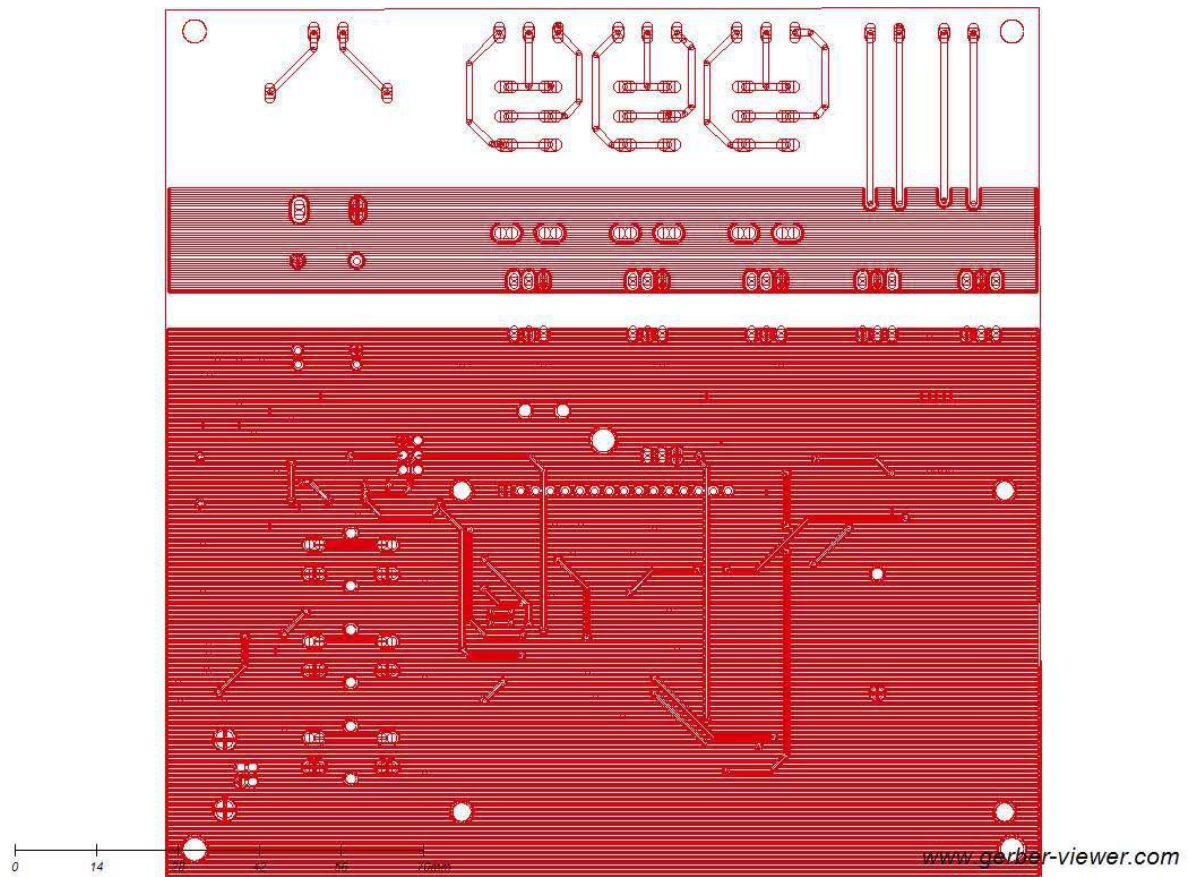


Figura Ap1.4 - Ficheiro GBL [46].

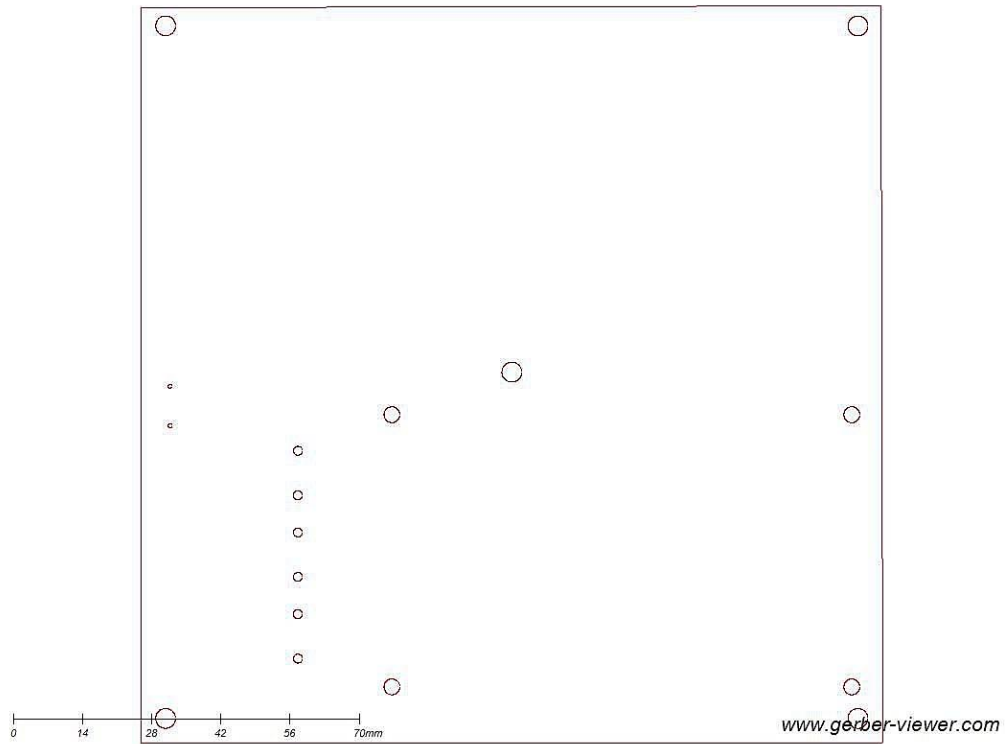


Figura Ap1.5 - Fichero GBO [46].

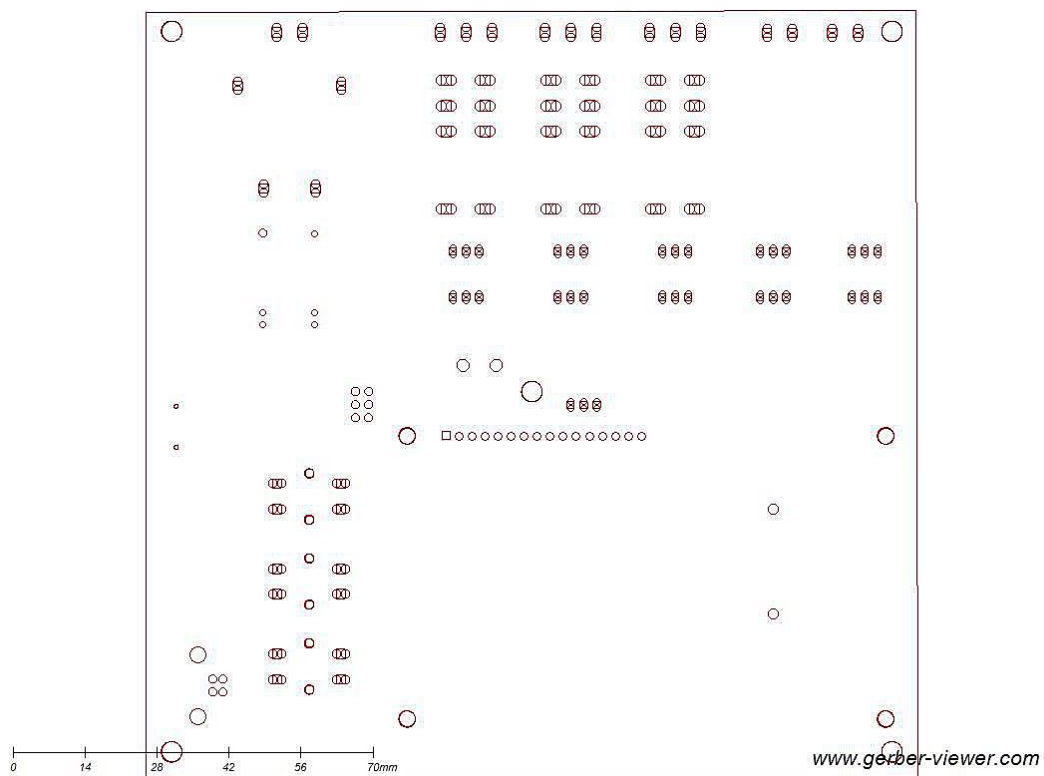


Figura Ap1.6 - Fichero GBS [46].

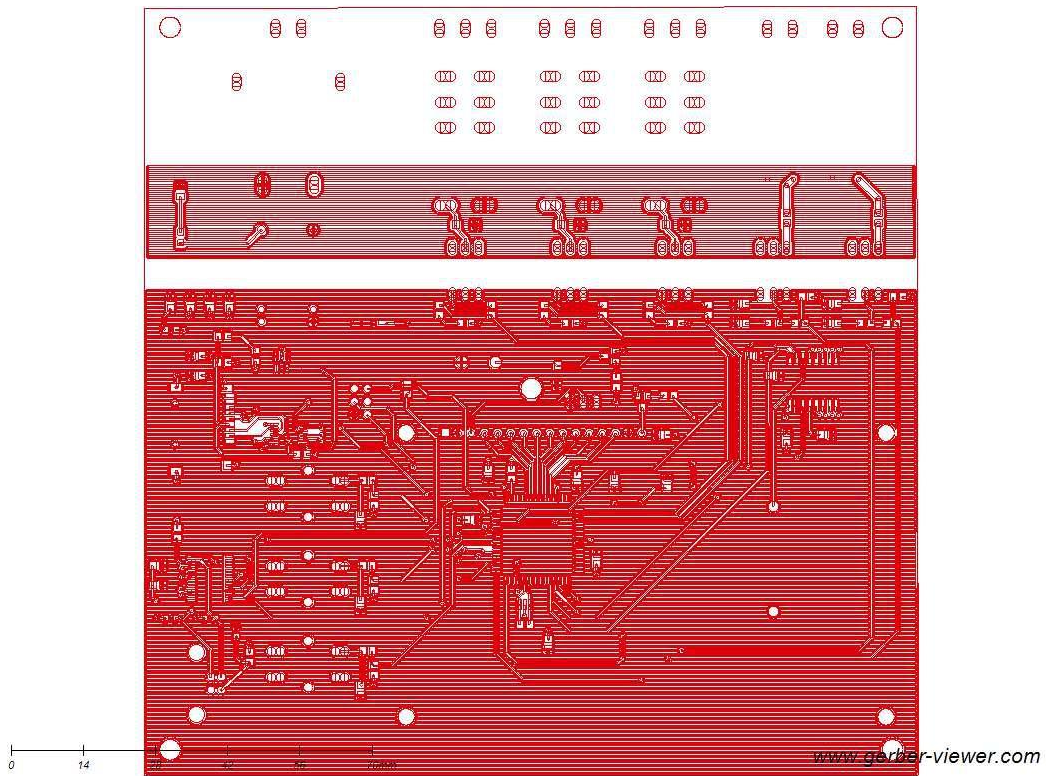


Figura Ap1.7 - Ficheiro GTL [46].

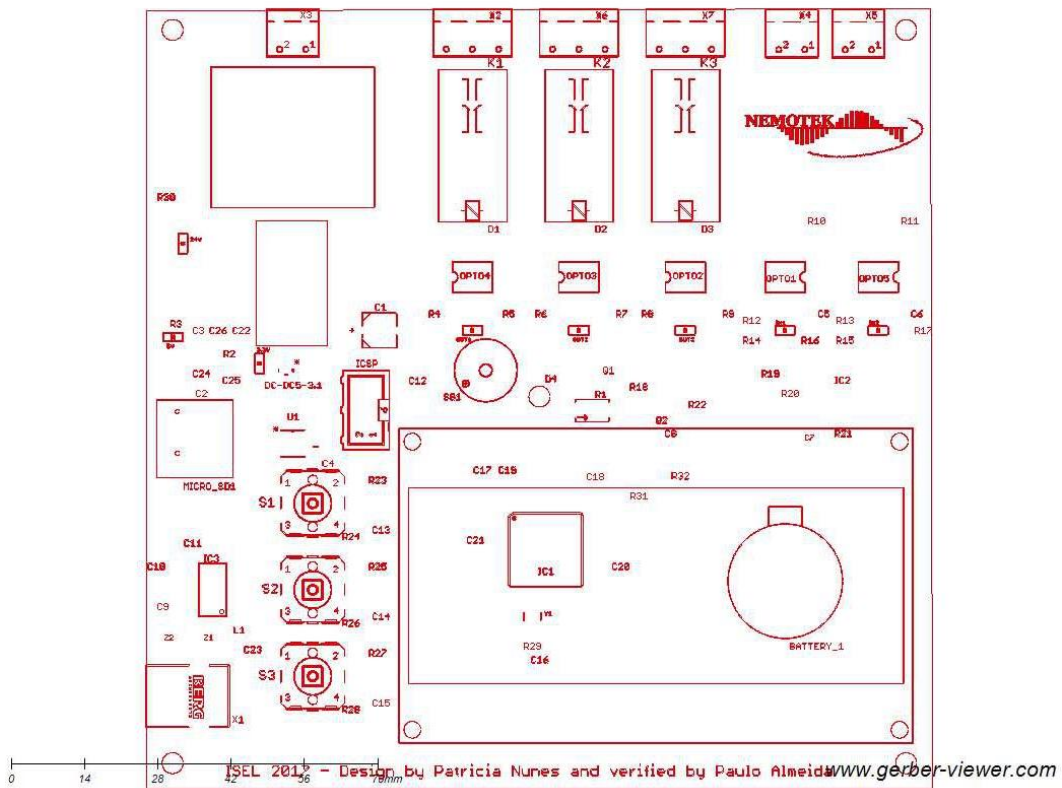


Figura Ap1.8 - Ficheiro GTO [46].

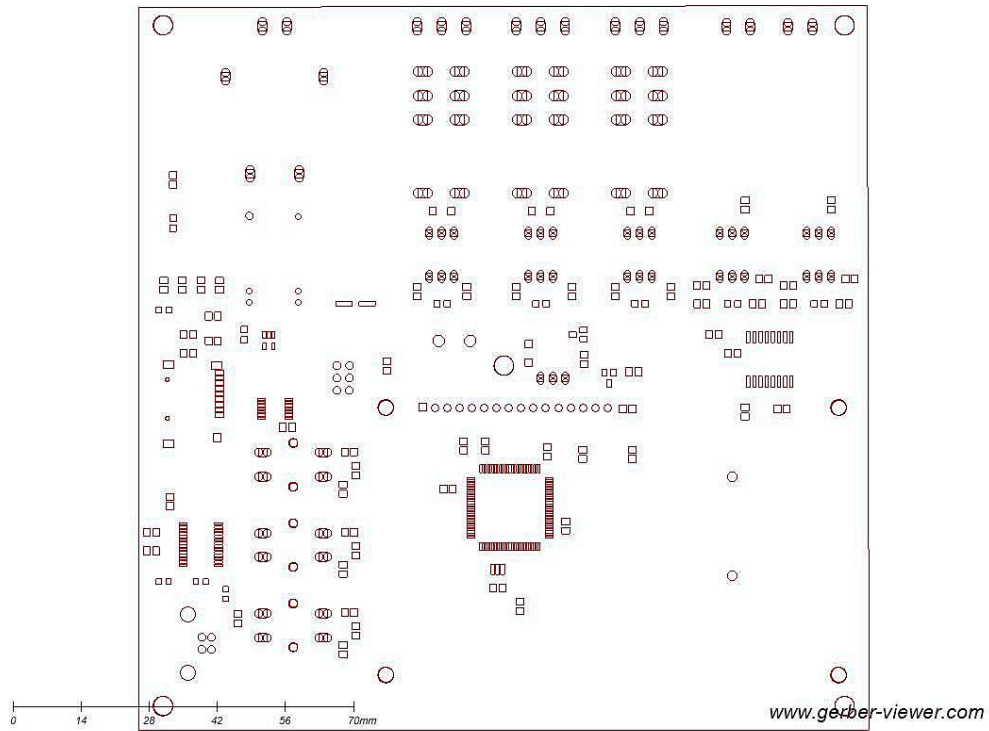


Figura Ap1.9 - Ficheiro GTS [46].

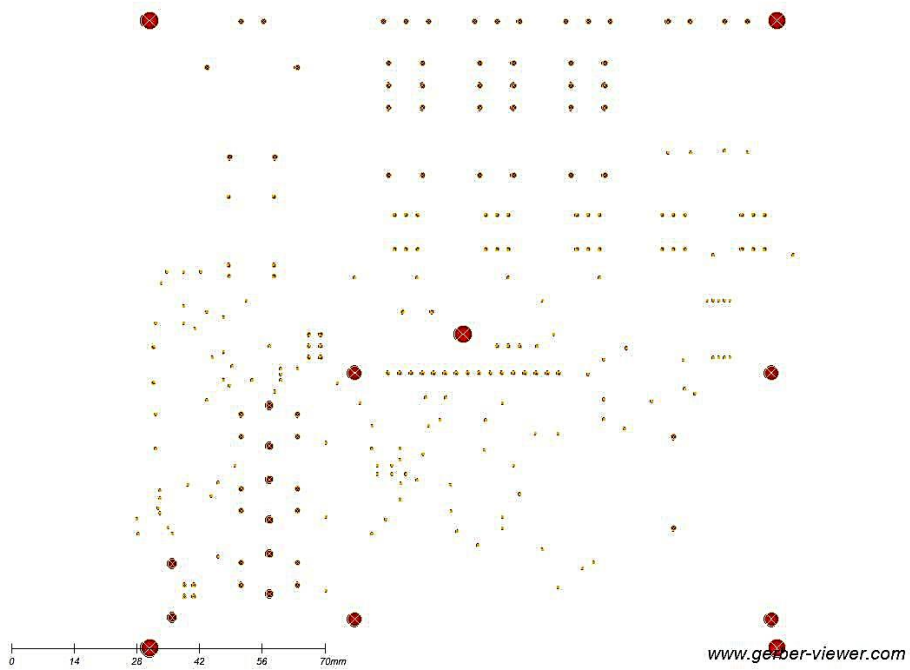


Figura Ap1.10 - Ficheiro com a extensão .txt [46].

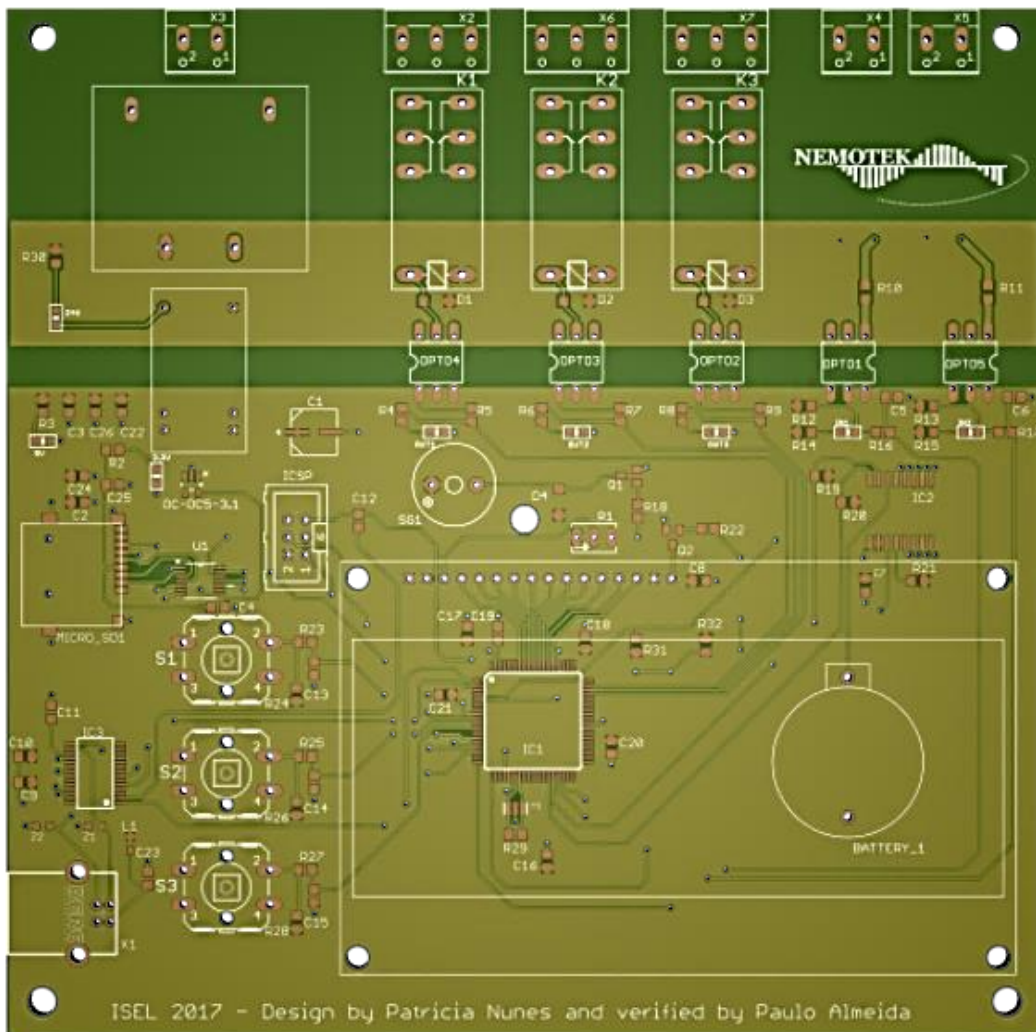


Figura Ap1.11 - Renderização dos Gerbers do PCB, layer de cima [42].

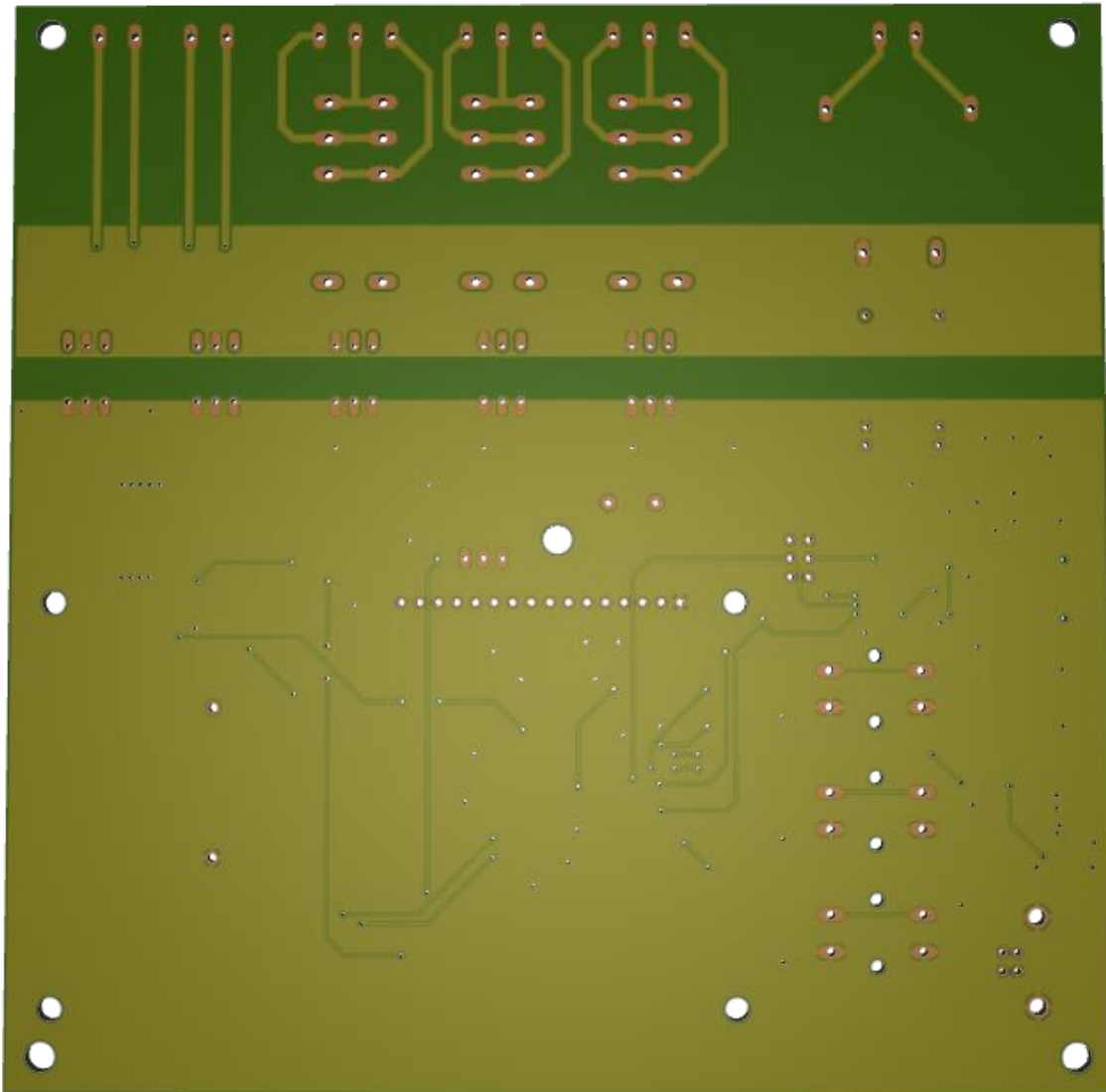


Figura Ap1.12 - Renderização dos *Gerbers* do PCB, *layer* de baixo [42].

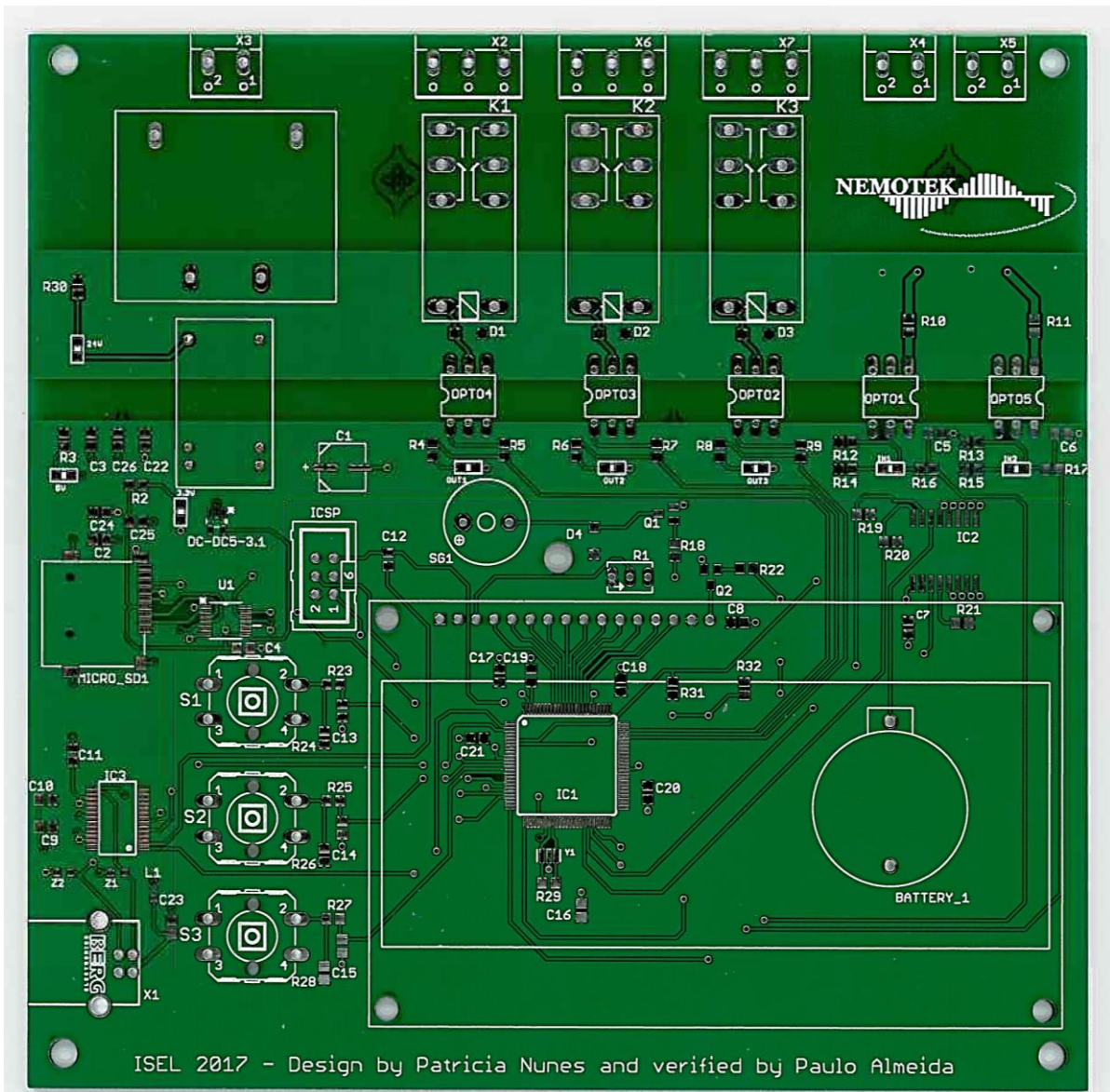


Figura Ap1.13 – Aspetto da *layer* de cima do PCB real.

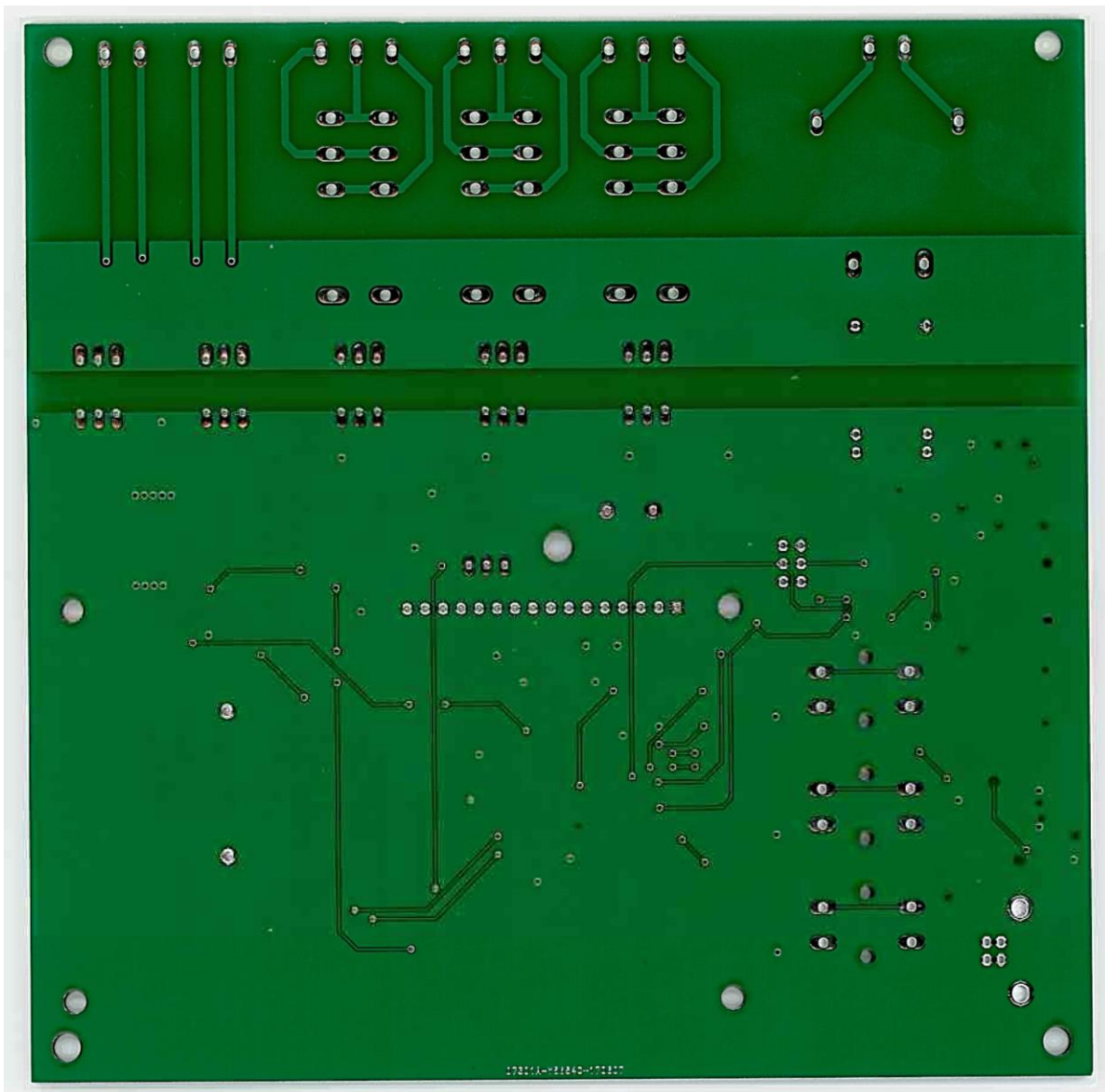


Figura Ap1.14 – Aspeto da *layer* de baixo do PCB real.

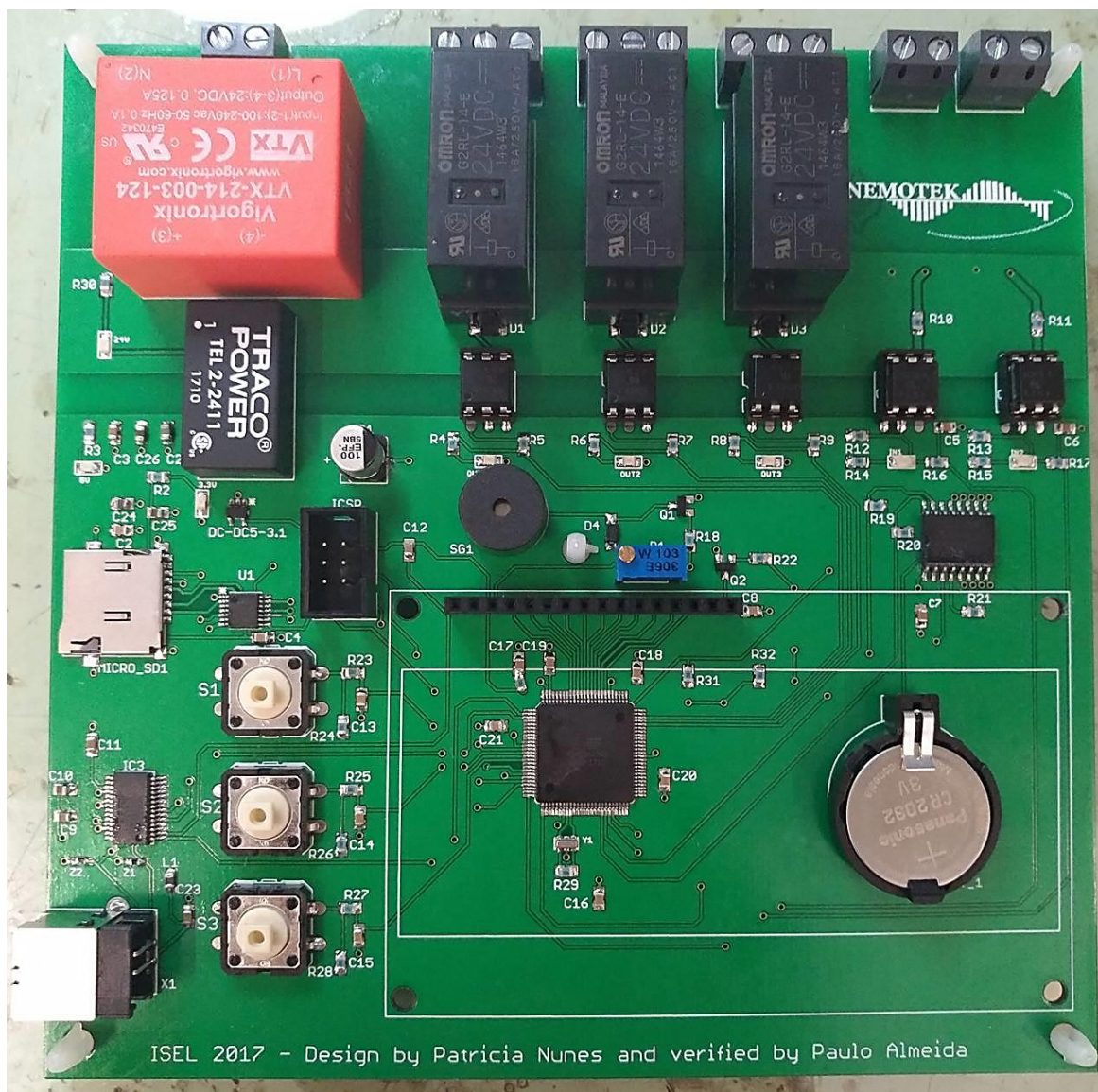


Figura Ap1.15 – Aspetto do PCB com os componentes inseridos, sem o LCD colocado (placa eletrônica).

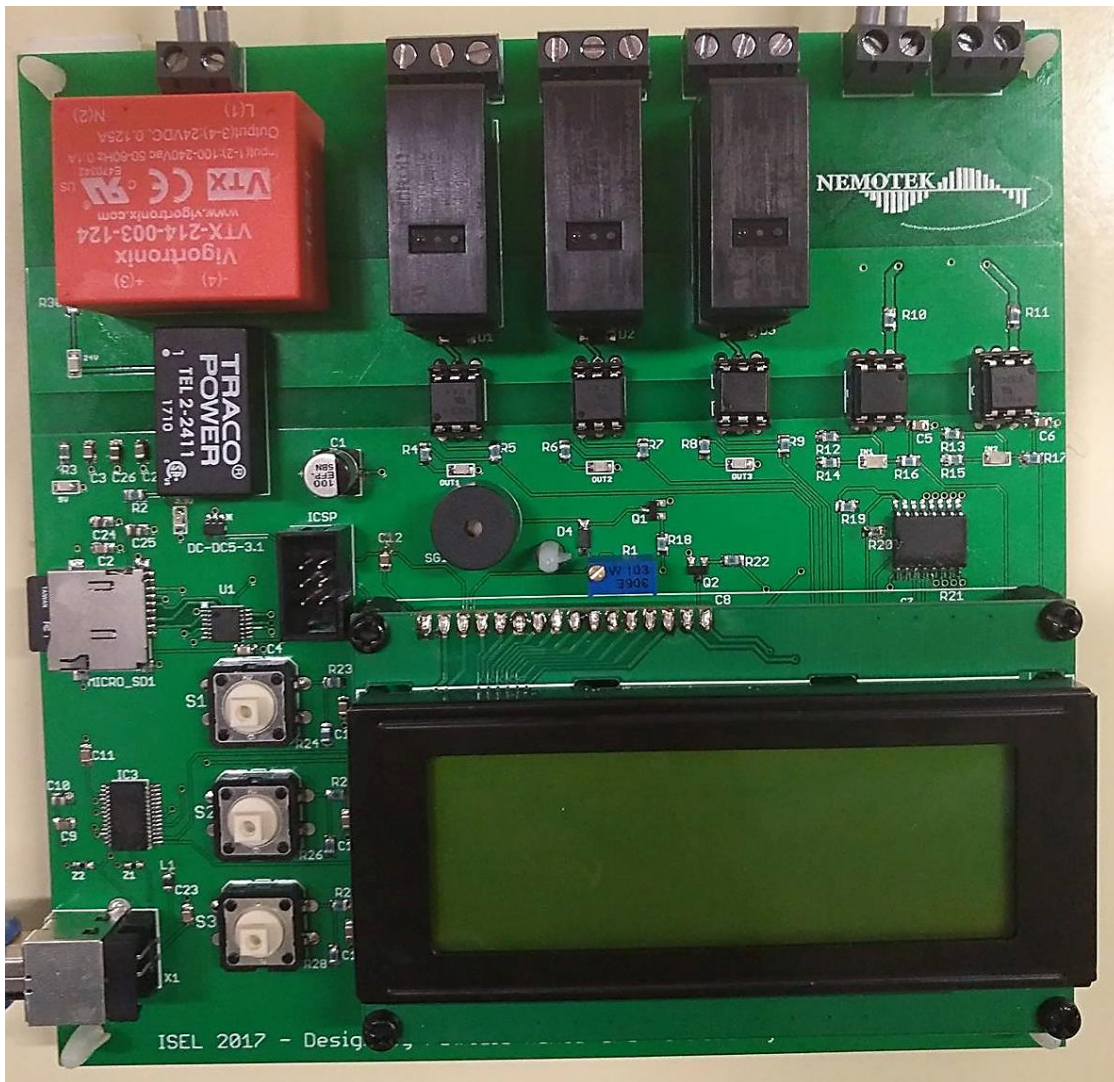
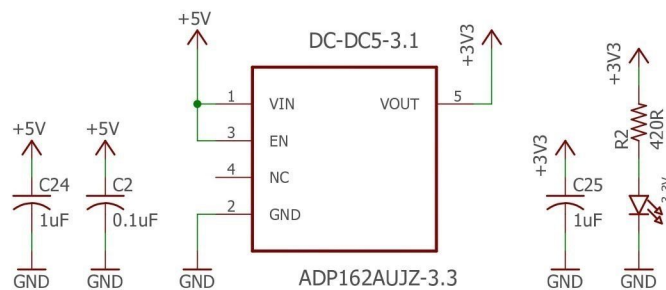
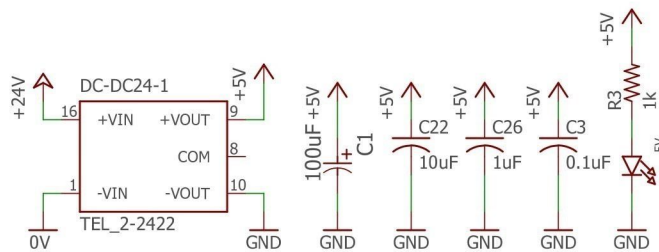
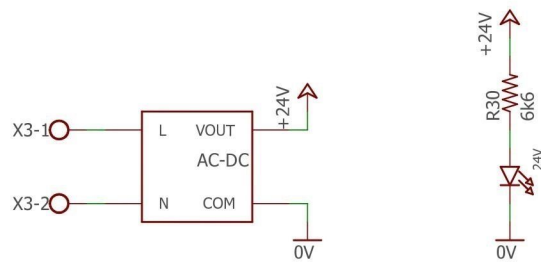


Figura Ap1.16 – Aspetto do PCB com os componentes inseridos, com o LCD colocado (placa eletrónica).



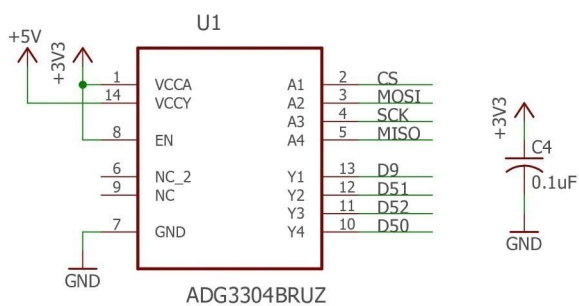
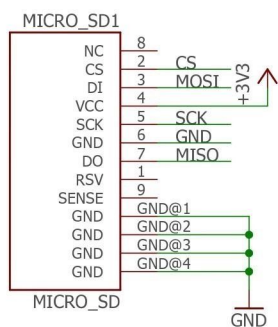
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



# CIRCUITOS DE ALIMENTAÇÃO

# DESENHO 1

O presente desenho é propriedade da Patrícia Cabago Nunes e não pode ser usado ou reproduzido no seu todo ou em parte sem autorização expressa da autora.



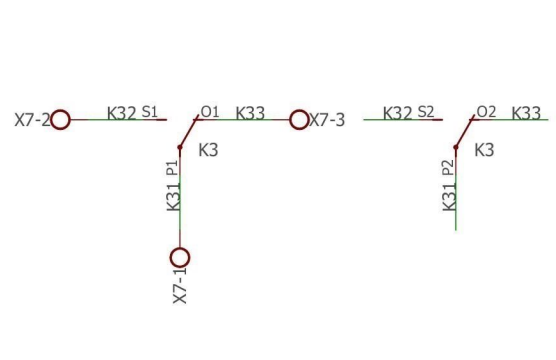
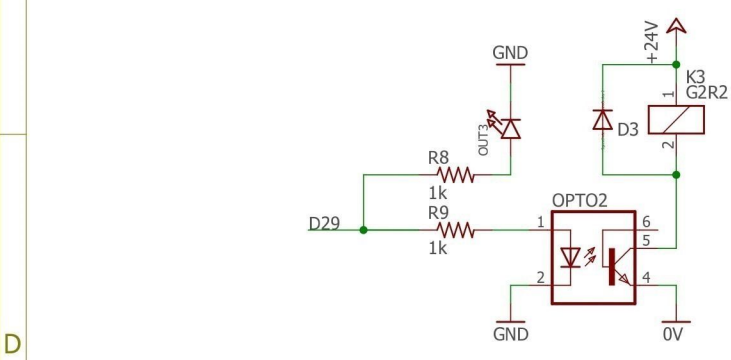
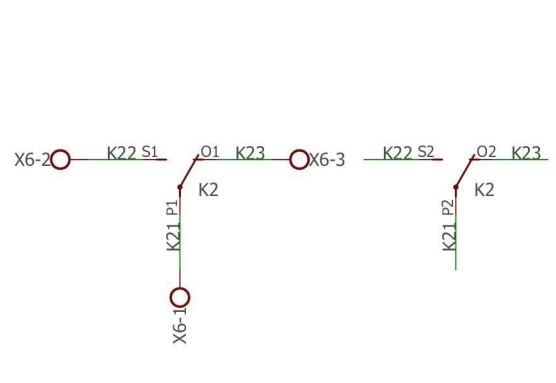
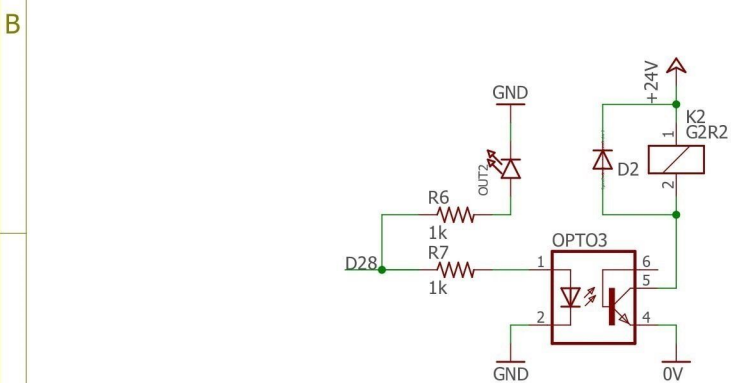
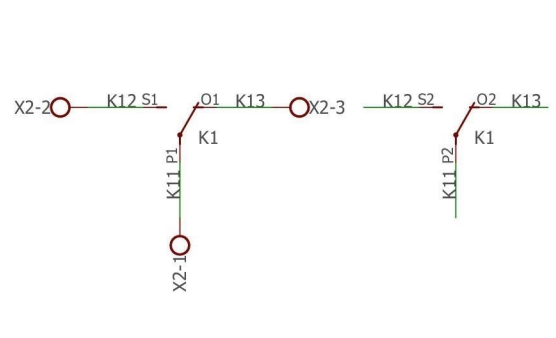
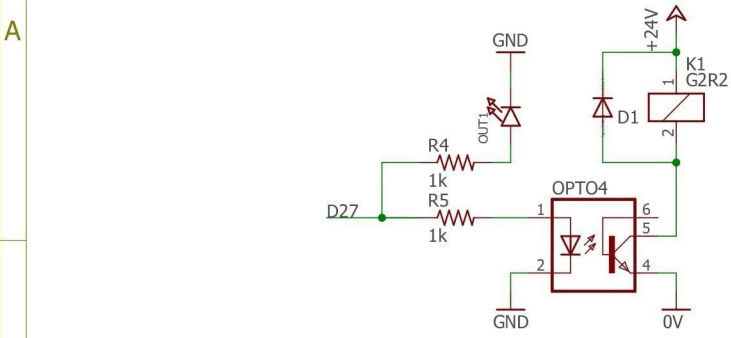
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DO MICRO-SD

DESENHO 2

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



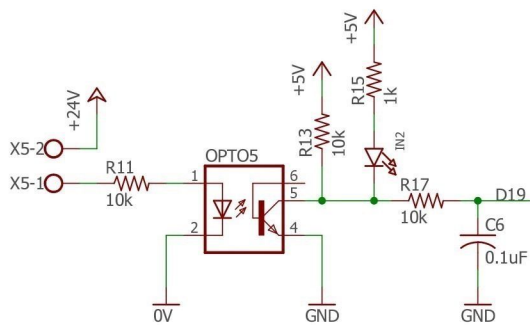
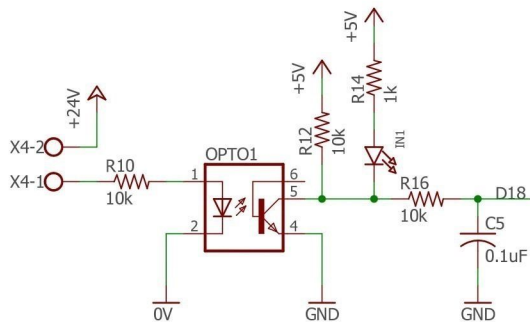
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



# CIRCUITO DAS SAIDAS

# DESENHO 3

O presente desenho é propriedade da Patrícia Cabago Nunes e não pode ser usado ou reproduzido no seu todo ou em parte sem autorização expressa da autora.



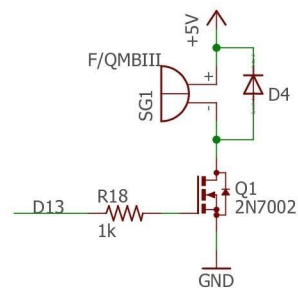
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DAS ENTRADAS

DESENHO 4

O presente desenho é propriedade da Patrícia Cabago Nunes e não pode ser usado ou reproduzido no seu todo ou em parte sem autorização expressa da autora.



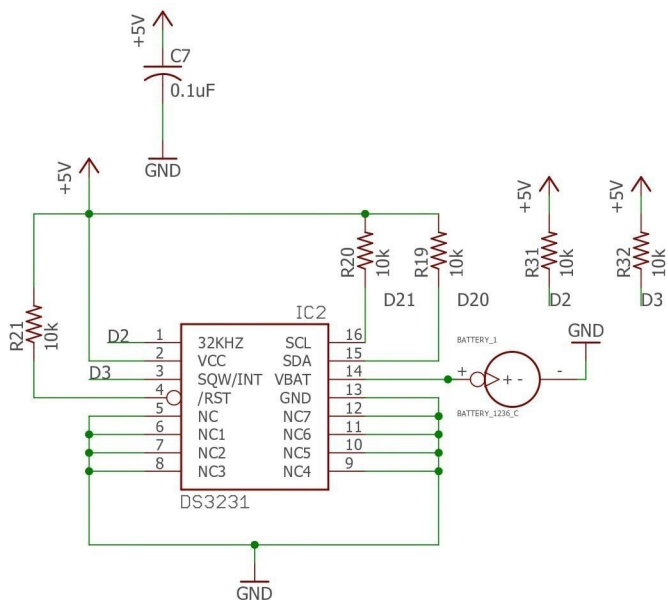
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DO BUZZER

DESENHO 5

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



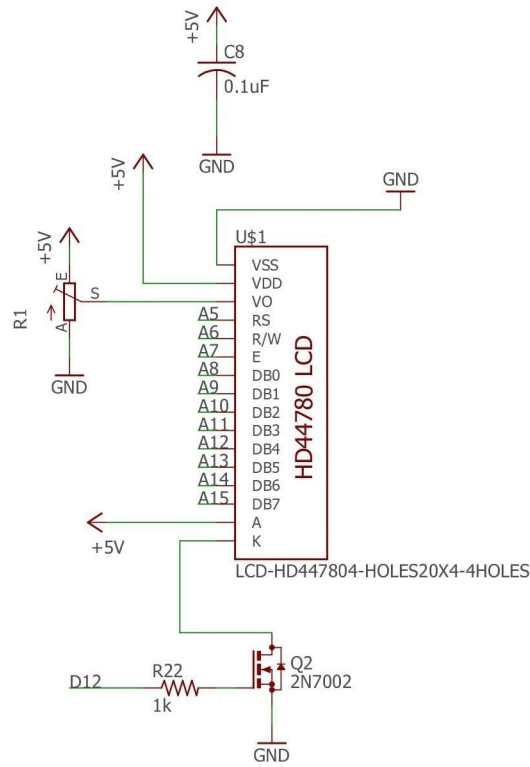
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DO RTC

DESENHO 6

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



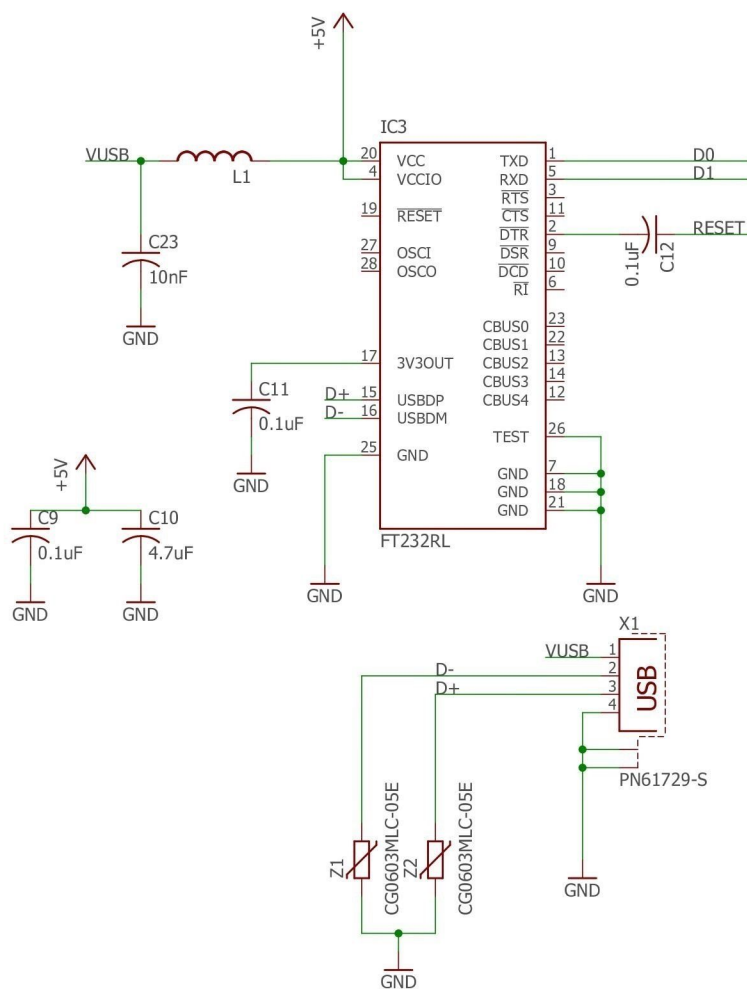
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DO LCD

DESENHO 7

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



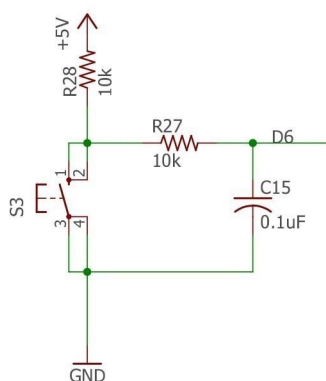
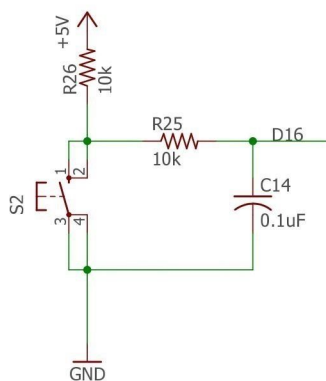
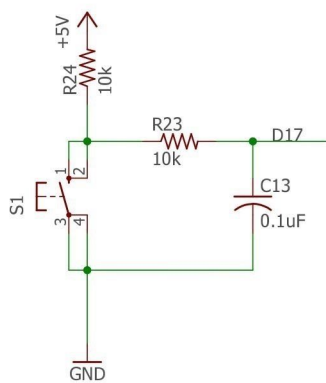
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



CIRCUITO  
 DO USB

DESENHO 8

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



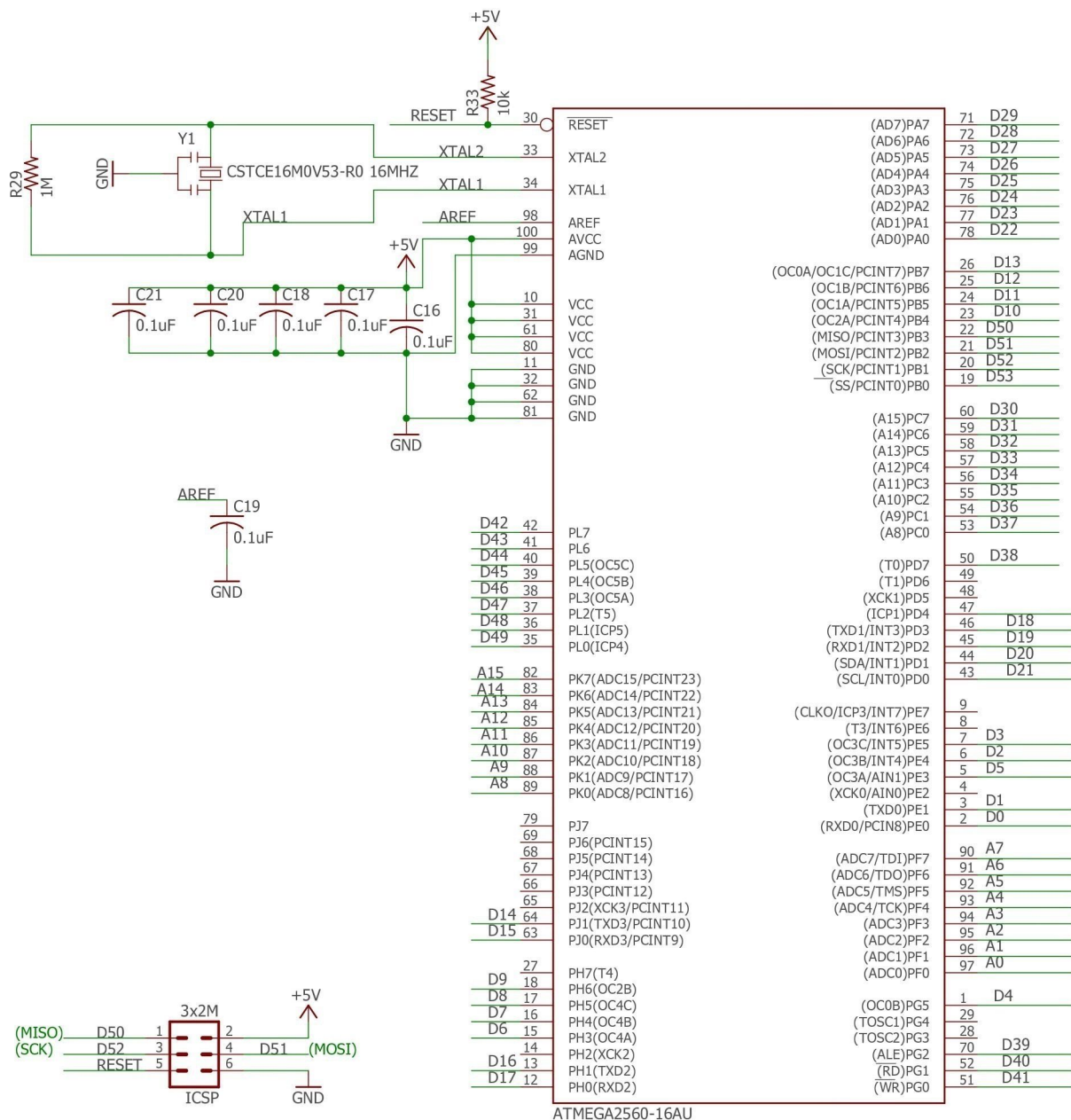
ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



# CIRCUITO DOS BOTOES

# DESENHO 9

O presente desenho é propriedade da  
 Patrícia Cabago Nunes e não pode ser usado  
 ou reproduzido no seu todo ou em parte  
 sem autorização expressa da autora.



ATMEGA2560-16AU

ISEL 2017 - Desenhado por  
 Patrícia Nunes  
 e Verificado por Paulo Almeida



# CIRCUITO DO DESENHO 10 MICROCONTROLADOR

O presente desenho é propriedade da Patrícia Cabago Nunes e não pode ser usado ou reproduzido no seu todo ou em parte sem autorização expressa da autora.

## Apêndice II

### Conceitos para a Blindagem



## **Blindagem**

A solução desenvolvida estará sujeita a interferências eletromagnéticas, perturbações harmônicas e variações térmicas. Para impedir que a solução esteja diretamente submetida a estes comportamentos será necessário proteger o equipamento (blindagem).

As interferências externas/perturbações são tensões ou correntes variáveis e indesejáveis que se sobrepõem ao sinal produzido ou tratado por um determinado dispositivo elétrico. A alteração da informação pode ter consequências na atuação correta do circuito elétrico [40].

A Interferência Eletromagnética (EMI) expressa o ruído existente na transmissão de informação ou no funcionamento de um circuito. As EMI “podem ser originadas internamente e/ou externamente ao sistema de comunicação”. É importante referir que perturbações eletromagnéticas são diferentes de interferências eletromagnéticas, pois a primeira designa a causa e a segunda o efeito observado. Como prevenção das EMI são utilizados acessórios com os seguintes materiais [49]:

- Plástico, que é um excelente isolante elétrico, mas não oferece proteção contra campos eletromagnéticos;
- Alumínio, que não oferece proteção elétrica (pois é um bom condutor de eletricidade), mas oferece boa blindagem eletromagnética;
- Aço (zincado ou pintado), que não é um bom condutor de eletricidade, não oferece proteção elétrica, porém proporciona boa blindagem eletromagnética.

O alumínio pode ser considerado um bom material para blindagem eletromagnética. Uma vez que, “segundo *Faraday*, um campo magnético variável pode criar uma corrente elétrica que por sua vez gera um campo eletromagnético contrário ao que lhe deu origem. Esse efeito é o responsável pela atenuação das interferências quando são utilizados canaletas ou dutos de alumínio. Além desse fato, as canaletas de alumínio são praticamente imunes às correntes de *Foucault* devido a sua condutibilidade elétrica” [49]. No entanto, blindar campos magnéticos de baixa frequência não é tão fácil como blindar campos elétricos. “A eficácia da blindagem magnética depende do tipo de material e sua permeabilidade, a sua espessura e as frequências envolvidas” [50]. Assim para baixas frequências (<100 kHz), devido à sua permeabilidade, o aço é mais eficiente do

que o alumínio e o cobre. Para altas frequências é mais comum o uso de cobre ou alumínio [50].

Numa eventual passagem deste protótipo para uma aplicação industrial terá de ser realizado ensaios no local de aplicação, de forma a verificar a imunidade do equipamento, ao ruído de origem eletromagnética. Dessa forma será possível selecionar o material a usar, juntamente com a espessura. Este trabalho não foi desenvolvido, devido a não existir forma de avaliar os campos eletromagnéticos, não possuindo o ISEL equipamento para tal.

Atualmente, qualquer equipamento gera harmónicas, uma vez que a sua maioria contém cargas não lineares [51]. O único equipamento neste projeto que se encontra ligado à rede elétrica é o conversor AC-DC de 230 VAC para 24 VDC. De acordo com o catálogo do conversor este encontra-se, em relação à distorção harmónica, protegido segundo a norma europeia EN 61000-3-2, que é a análoga da norma IEC 61000-3-2 [52]. Esta norma menciona as limitações das harmónicas de corrente injetadas na rede pública de alimentação.

Aplica-se a equipamentos elétricos e eletrónicos cuja corrente de entrada é de 16 A por fase, conectados a uma rede pública de baixa tensão alternada, de 50 ou 60 Hz, com uma tensão fase-neutro entre 220 e 240 V.

Os equipamentos encontram-se classificados consoante 4 classes:

- Classe A: Equipamentos com alimentação trifásica equilibrada e todos os que não são incluídos nas classes seguintes.
- Classe B: Ferramentas portáteis e equipamentos de soldagem não profissionais.
- Classe C: Dispositivos de iluminação com potência acima de 25 W.
  - Para potência igual ou inferior a 25 W e exclusivamente para lâmpadas de descarga, aplicam-se os limites da classe A.
  - Reguladores de intensidade para lâmpadas incandescentes, aplicam os limites da classe A.
- Classe D: Equipamentos de TV, computadores pessoais e monitores de vídeo. A potência ativa de entrada deve ser igual ou inferior a 600 W, obedecendo às condições de ensaio estabelecidas na norma (que variam de acordo com o tipo de equipamento) [52].

Tabela Ap2.1 – Limites das harmônicas de corrente para as diferentes classes [52].

|                           | <b>Ordem da Harmónica (n)</b> | <b>Classe A</b><br>Máximo de Corrente [A] | <b>Classe B</b><br>Máximo de Corrente [A] | <b>Classe C</b><br>(>25 W) % da fundamental | <b>Classe D</b><br>(≤ 600 W) [mA/W] | <b>Classe D</b><br>Máximo |
|---------------------------|-------------------------------|---|---|---|-------------------------------------|---------------------------|
| <b>Harmónicas Ímpares</b> | 3                             | 2,30                                      | 3,45                                      | 30.FP                                       | 3,4                                 | 2,30                      |
|                           | 5                             | 1,14                                      | 1,71                                      | 10  | 1,9                                 | 1,14                      |
|                           | 7                             | 0,77                                      | 1,155                                     | 7   | 1,0                                 | 0,77                      |
|                           | 9                             | 0,40                                      | 0,60                                      | 5   | 0,5                                 | 0,40                      |
|                           | 11                            | 0,33                                      | 0,495                                     | 3   | 0,35                                | 0,33                      |
|                           | 13                            | 0,21                                      | 0,315                                     | 3   | 0,296                               | 0,21                      |
|                           | 15 ≤ n ≤ 39                   | 2,25/ n                                   | 3,375/ n                                  | 3   | 3,85/ n                             | 2,25/ n                   |
| <b>Harmónicas Pares</b>   | 2                             | 1,08                                      | 1,62                                      | 2   |                                     |                           |
|                           | 4                             | 0,43                                      | 0,645                                     |   |                                     |                           |
|                           | 6                             | 0,3                                       | 0,45                                      |   |                                     |                           |
|                           | 8 ≤ n ≤ 40                    | 1,83/ n                                   | 2,76/ n                                   |   |                                     |                           |

O equipamento, como está desenvolvido, tolera variações térmicas de  $-20\text{ }^{\circ}\text{C}$  a  $+60\text{ }^{\circ}\text{C}$ . Estes valores correspondem ao intervalo de temperatura que todos os componentes inseridos na placa eletrónica suportam, como demonstra a Tabela Ap2.2.

Tabela Ap2.2– Tabela resumo das variações de temperatura toleradas pelos componentes.

| <b>Dispositivo</b>                          | <b>Temperatura mínima [°C]</b> | <b>Temperatura máxima [°C]</b> |
|---|--------------------------------|--------------------------------|
| W237-102 e 103 (Bloco de 2 e 3 terminais)   | -30                            | +120                           |
| Ficha IDC de 6 pinos                        | -40                            | +105                           |
| Conetor LCD (barra de 16 pinos macho-fêmea) | –                              | –                              |
| Espaçador de PCB                            | –                              | –                              |
| Botões                                      | -25                            | +70                            |
| 4N28(Optocoupler)                           | -55                            | +85                            |
| Led vermelho                                | -40                            | +80                            |
| Condensador eletrolítico 100uF              | -55                            | +105                           |
| Pilha da Bateria                            | -20                            | +70                            |
| VARISTOR                                    | -40                            | +85                            |
| F/QMBIII(Buzzer)                            | -30                            | +70                            |

|                               |     |      |
|-------------------------------|-----|------|
| Suportes do LCD               | –   | –    |
| USB                           | -55 | +85  |
| Conversor AC-DC 230-24V       | -25 | +70  |
| Díodos                        | -65 | +125 |
| Bobina circuito USB           | -40 | +125 |
| Potenciômetro                 | –   | –    |
| Condensador 0,1uF             | -55 | +125 |
| Resistência de 10k $\Omega$   | -55 | +125 |
| Condensador de 0,01uF         | -55 | +125 |
| Condensador de 10uF           | -55 | +85  |
| Resistência de 1M $\Omega$    | -55 | +125 |
| Resistência de 1k $\Omega$    | -55 | +125 |
| Condensador de 1uF            | -55 | +85  |
| Mosfet                        | -55 | +150 |
| Condensador de 4,7uF          | -55 | +85  |
| Resistência de 422 $\Omega$   | -55 | +125 |
| Resistência de 6,65k $\Omega$ | -55 | +125 |
| Transdutor de nível           | -40 | +85  |
| Conversor DC-DC 5-3,3 V       | -40 | +125 |
| Microcontrolador              | -40 | +85  |
| Bateria                       | –   | –    |
| Cristal                       | -40 | +125 |
| RTC                           | -40 | +85  |
| FTDI                          | -40 | +85  |
| Relés                         | -40 | +85  |
| Suporte dos Relés             | –   | –    |
| LCD-4 buracos                 | -20 | +60  |
| Conversor DC-DC 24-5V         | -40 | +75  |
| Suporte para o cartão SD      | –   | –    |
| PCB                           | –   | –    |

# Anexo I

Documento com os Requisitos

da Empresa NEMOTEK





## Controlo auxiliar de bateria de condensadores

*Controlo horário de uma bateria de condensadores – Julho de 2016,*

Projecto de desenvolvimento em parceria ISEL – NEMOTEK para uma solução alternativa à actualmente utilizada pela Nemotek para o controlo auxiliar de uma bateria de condensadores, tendo em vista a inovação da solução, a optimização e redução de custos.

## Conteúdo

|                         |   |
|-------------------------|---|
| Introdução:             | 2 |
| Solução a desenvolver:  | 2 |
| Requisitos Físicos:     | 3 |
| Requisitos de controlo: | 4 |
| Opções:                 | 5 |
| Expectativa Nemotek:    | 5 |

# Projecto, controlo horário da bateria.

---

## **Introdução:**

A Nemotek, é uma empresa Portuguesa, com 20 anos de existência, e actua essencialmente nos mercados em Portugal e Angola, encontrando-se em expansão das suas actividades para Espanha, França e países Africanos de expressão portuguesa.

Uma das principais actividades da Nemotek é o controlo da energia reactiva. Sendo representante de diversas marcas, nomeadamente Alpes Technologies (Grupo Legrand) e ZEZ.

A actividade da Nemotek cobre todas as gamas de soluções para o controlo de energia reactiva do mercado, sejam soluções para Baixa Tensão como de Média Tensão, com filtros ou simples, com contactores estáticos ou electromecânicos, ou ainda para soluções indutivas ou capacitivas.

De forma a efectuar um controlo preciso da instalação e adaptar os equipamentos de controlo standard às necessidades de cada cliente e da legislação de cada país, a Nemotek tem desenvolvido várias soluções técnicas próprias.

No presente caso, pretende otimizar, personalizar e reduzir os custos de uma solução de controlo de baterias de condensadores actual a qual actualmente é assegurada via a utilização de um relé lógico programável da Crouzet, modelo XD10.

A solução deverá possuir um custo de fabricação baixo, ser segura e funcional e corresponder aos requisitos básicos descritos neste documento.

Pretende igualmente a Nemotek iniciar uma relação de proximidade com o sector universitário permitindo a troca de conhecimentos prático-teóricos para um melhor desenvolvimento das duas instituições, proporcionar ganhos comuns na competitividade do mercado através da inovação, optimização e adaptação das soluções propostas.

## **Solução a desenvolver:**

A solução que se pretende desenvolver deverá substituir o actual equipamento XD10, assim como a necessidade de integrar externamente um transformador e protecções eléctricas para o equipamento.

Será instalado na proximidade do RV aproveitando a alimentação eléctrica do RV (400 V<sup>~</sup>), assim como a existência das saídas de controlo dos escalões.

Para melhor compreensão junta-se em anexo alguns manuais simples de uso comum, da instalação de uma bateria, da integração do XD10 e da utilização do mesmo.

## Requisitos Físicos:

A placa de controlo, deverá ser colocada no interior dos armários das baterias de condensadores, na proximidade do Relé Varimétrico (RV), complementando-o quanto a algumas funções de horário as quais não existem nas funções standard dos mesmos.

A localização do sistema implica que estará sujeito a interferências electromagnéticas e a perturbações harmónicas importantes, assim como variações térmicas, assim o sistema proposto deve possuir alguma blindagem para estes fenómenos e ser testada em ambiente industrial (com o apoio da Nemotek).

O sistema deve possuir as seguintes características básicas mínimas:

1. Alimentação: bifásica 400 V~ .
2. Robustez eléctrica: Suportar tensões de 350->450 V~;
3. Temperatura: -5°C a 50 °C.
4. Saídas a relé : 5 A, 400 V~ (cargas indutivas). Os relés podem ser extraíveis.
5. Saídas necessárias : No mínimo 3 saídas configuráveis, para :
  - a. Corte da bateria (da alimentação ao RV) no horário escolhido.
  - b. Forçar a entrada em serviço de um escalão no horário escolhido.
6. Entradas: No mínimo 2 entradas, as entradas tem de estar protegidas, o equipamento deve fornecer a referência de tensão a estas entradas:
  - a. Entrada Alarme,
  - b. Entrada de controlo,
7. Seleccionador / visualizador de horário: O cliente deve poder escolher de forma simples o horário de serviço (horários de referencia do distribuidor ou outros), quando em serviço deve-se poder identificar o horário escolhido.
8. Visualização de estado:
  - a. Alimentação,
  - b. As saídas,
  - c. As entradas,
9. Porta de comunicação para ligação ao PC.
10. Protecção eléctrica (fusível) do equipamento.
11. Comutador On/off.

## Requisitos de controlo:

O controlador deve ligar e desligar a bateria (desligar a alimentação do RV) assim como colocar em serviço um escalão (um condensador) de compensação, de acordo com um conjunto de horários pré-estabelecidos.

Para o efeito deverá possuir os seguintes requisitos básicos:

1. Capacidade para suportar em memória os vários horários semanais de funcionamento da bateria, mínimo 10 ( em Portugal os horários são definidos pela ERSE : <http://www.erse.pt/pt/electricidade/tarifaseprecos/periodoshorarios/paginas/default.aspx> ) Outros horários podem ser requeridos para as regiões autónomas ou para os mercados internacionais onde a Nemotek actua. No caso português existem quatro períodos horários: horas de ponta, horas cheias, horas de vazio e horas de super-vazio, a bateria deve estar ligada nas horas fora de vazio (ponta e cheia) e desligada em horas de vazio (vazio e super-vazio). O controlo deve ser flexível e definir para cada período se cada saída deve estar ligada ou desligada.
2. Os horários semanais podem incluir dias feriados, cujo comportamento é normalmente idêntico ao domingo, o sistema deve permitir incluir vários dias feriados (20) para 5 anos.
3. Em Portugal e na maioria dos países existe alteração da hora verão / Inverno, o sistema deve alterar automaticamente a hora legal.
4. Em Portugal a alteração da hora verão / inverno implica alterações aos períodos horários, deve o sistema possibilitar a alteração do horário em função da mudança de hora.
5. Cada saída pode funcionar para corte do RV (desligar a bateria) ou para actuar um condensador (contactador da bateria), deve existir a possibilidade de atribuir a cada saída o tipo de serviço.
6. Caso a uma saída seja para actuar um condensador, esta saída estará sujeita a uma temporização ao trabalho, evitando religar o mesmo antes de terminado o tempo de descarga do condensador. Esta temporização deverá ser programável de 1 a 1200 s (por norma nas baterias em baixa tensão possuem um tempo de descarga de 3 minutos e as de média tensão de 10 minutos).
7. A cada saída deve poder ser atribuída uma temporização por antecipação ou após a referência horária de -300 a 300 s, esta temporização permite que as actuações não sejam simultâneas e considerar o tempo de reinicialização do RV.
8. Cada entrada pode funcionar como alarme ou controlo, deve ser possível atribuir a cada entrada uma função distinta, a entrada de alarme sobrepõe-se à de controlo.
9. Caso uma a entrada seja de controlo ou alarme deve ser possível atribuir as seguintes funções:
  - a. Saídas que vão ser afectadas,
  - b. Ligar ou desligar,
  - c. Prioridade ou não sobre o horário,
  - d. Tempo para actuação (evitar ruído na entrada),
10. Deve estar previsto a indicação de alarme sonora ou visual distinta para os seguintes casos :
  - a. Entrada de alarme activa foi activada,
  - b. Datas de feriados fixos a 3 meses de estarem desactualizados,
  - c. Manutenção sistemática não realizada à data estipulada,
  - d. Outra a definir (temperatura, tensão ...),
11. Deve ser possível activar uma data de manutenção pelo técnico manutenção.
12. Deve ser criado um processo de transferência de dados simples e de interface com o cliente. Deve ser possível os serviços técnicos enviarem ao cliente de forma digital os

horários e as programações de controlo e este transferi-las para o equipamento. Assim como o processo inverso para diagnóstico dos serviços técnicos.

13. O cliente deve poder de forma simples, e visual identificar a data / hora no momento e corrigir se necessário.
14. Deve ser criado um registo de acontecimentos (mínimo 100 registos), com a indicação da data-hora de cada acontecimento, para efeito de identificação de erros de funcionamento. Os acontecimentos mínimos registados:
  - a. Alteração de entradas,
  - b. Alteração das saídas,

## Opções:

Em função de diversas opções a incluir (não obrigatórias para o projecto), serão desenvolvidos requisitos específicos.

1. Inclusão de uma medição de corrente (0 a 5 A), de forma a criar cenários para as saídas em função de valores de consumo da instalação.
2. Interface gráfica.
3. Identificação do estado das saídas do RV.
4. Medição da temperatura internamente e actuação das saídas de forma a proteger a instalação.
5. Porta de comunicação RS485 (para actuação nos RV ou integração em sistemas de gestão técnica).
6. Medição de valor da tensão composta (bi-fásica). Criar cenários para as saídas em função dos valores, por exemplo alarme e protecção.
7. Cálculo do Factor de potência, potencia activa e reactiva.

## Expectativa Nemotek:

A Nemotek tem a expectativa de ter um equipamento de controlo para substituir a solução actual num prazo razoável de um ano (a acordar), consciente dos riscos associados ao facto de ser um estudo académico.

Durante a realização do trabalho a Nemotek espera receber:

1. Proposta geral sobre o estudo, a incluir:
  - a. Descrição geral da proposta,
  - b. Plano de execução (estimativa),
  - c. Estimativa de materiais que a Nemotek poderá fornecer, exemplos: material de teste, pequena bateria, contactor, RV, etc.
  - d. Estimativa de custos diversos, materiais de laboratório, consumíveis, componentes, etc.
2. Relatórios do desenvolvimento do projecto e desvios dos objectivos (custos, ...), com periodicidade a definir (exemplo, trimestral).
3. Relatórios de ensaios finais,
4. Um protótipo,
5. Um dossier de fabricação completo.
6. Manual de instalação,
7. Manual de utilização,
8. Software desenvolvido,
9. Programas comentados,
10. Autorização de alteração, produção, instalação e venda da solução por parte da Nemotek.

## **Anexo II**

# **Catálogo do PLC Crouzet**

**XD10**



# Millenium 3

## → Expandable version

- "High-performance" expandable solution with display
- Extended memory: 120 lines in LADDER language and up to 700 "typical" blocks in FBD language
- LCD with 4 lines of 18 characters and configurable backlighting
- Selective parameter setting: You can choose the parameters that can be adjusted on the front panel
- Analogue inputs 0-10 V  $\ddot{=}$  or 0-20 mA/Pt 100 with converters see page 46
- Open to XN network communication extensions and digital I/O or analogue extensions



XD10



XD26

### Part numbers

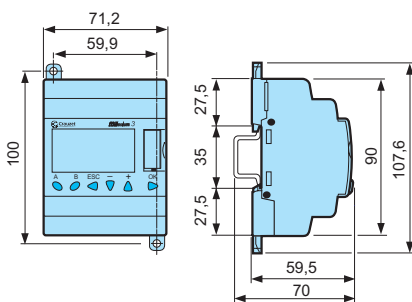
| Type | Input                              | Output                           | Supply             | Code     |
|------|------------------------------------|----------------------------------|--------------------|----------|
| XD10 | 6 digital of which 4 are analogue  | 4 relay                          | 24 V $\ddot{=}$    | 88970141 |
|      | 6 digital of which 4 are analogue  | 4 solid state of which 1 is PWM  | 24 V $\ddot{=}$    | 88970142 |
|      | 6 digital                          | 4 relay                          | 100 → 240 V $\sim$ | 88970143 |
|      | 6 digital                          | 4 relay                          | 24 V $\sim$        | 88970144 |
| XD26 | 16 digital of which 6 are analogue | 10 relay                         | 24 V $\ddot{=}$    | 88970161 |
|      | 16 digital of which 6 are analogue | 10 solid state of which 4 is PWM | 24 V $\ddot{=}$    | 88970162 |
|      | 16 digital                         | 10 relay                         | 100 → 240 V $\sim$ | 88970163 |
|      | 16 digital                         | 10 relay                         | 24 V $\sim$        | 88970164 |
|      | 16 digital of which 6 are analogue | 10 relay                         | 12 V $\ddot{=}$    | 88970165 |

### Accessories

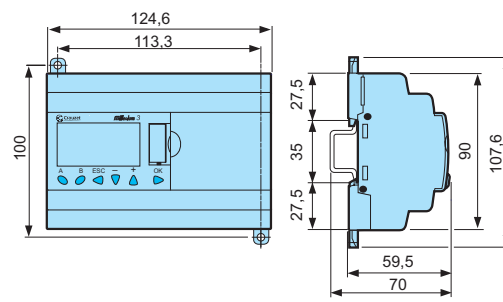
| Type                  | Description                                      | Code     |
|-----------------------|--|----------|
| M3 SOFT               | Multilingual programming software (CD-ROM)       | 88970100 |
| M3 SPECIFIC FUNCTIONS | Library of specific functions (CD-ROM)           | 88970103 |
| PA                    | EEPROM memory cartridge                          | 88970108 |
|                       | 3 m serial link cable: PC → Millenium 3          | 88970102 |
|                       | 3 m USB link cable: PC → Millenium 3             | 88970109 |
|                       | Millenium 3 → Bluetooth interface (class A 10 m) | 88970104 |

### Dimensions (mm)

XD10



XD26



For adapted products, see page 49

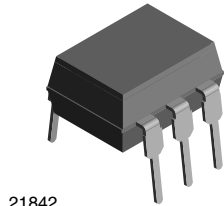


## Anexo III

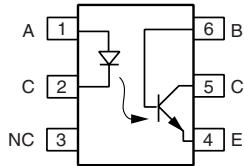
### Catálogo do *Optocoupler*



## Optocoupler, Phototransistor Output, with Base Connection



21842



1179004-5

### DESCRIPTION

The 4N25 family is an industry standard single channel phototransistor coupler. This family includes the 4N25, 4N26, 4N27, 4N28. Each optocoupler consists of gallium arsenide infrared LED and a silicon NPN phototransistor.

### FEATURES

- Isolation test voltage 5000 V<sub>RMS</sub>
- Interfaces with common logic families
- Input-output coupling capacitance < 0.5 pF
- Industry standard dual-in-line 6 pin package
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



RoHS COMPLIANT

### APPLICATIONS

- AC mains detection
- Reed relay driving
- Switch mode power supply feedback
- Telephone ring detection
- Logic ground isolation
- Logic coupling with high frequency noise rejection

### AGENCY APPROVALS

- UL1577, file no. E52744
- BSI: EN 60065:2002, EN 60950:2000
- FIMKO: EN 60950, EN 60065, EN 60335

| ORDER INFORMATION |                   |
|-------------------|-------------------|
| PART              | REMARKS           |
| 4N25              | CTR > 20 %, DIP-6 |
| 4N26              | CTR > 20 %, DIP-6 |
| 4N27              | CTR > 10 %, DIP-6 |
| 4N28              | CTR > 10 %, DIP-6 |

| ABSOLUTE MAXIMUM RATINGS (1)        |                |                   |       |      |
|-------------------------------------|----------------|-------------------|-------|------|
| PARAMETER                           | TEST CONDITION | SYMBOL            | VALUE | UNIT |
| <b>INPUT</b>                        |                |                   |       |      |
| Reverse voltage                     |                | V <sub>R</sub>    | 5     | V    |
| Forward current                     |                | I <sub>F</sub>    | 60    | mA   |
| Surge current                       | t ≤ 10 μs      | I <sub>FSM</sub>  | 3     | A    |
| Power dissipation                   |                | P <sub>diss</sub> | 100   | mW   |
| <b>OUTPUT</b>                       |                |                   |       |      |
| Collector emitter breakdown voltage |                | V <sub>CEO</sub>  | 70    | V    |
| Emitter base breakdown voltage      |                | V <sub>EBO</sub>  | 7     | V    |
| Collector current                   |                | I <sub>C</sub>    | 50    | mA   |
|                                     | t ≤ 1 ms       | I <sub>C</sub>    | 100   | mA   |
| Power dissipation                   |                | P <sub>diss</sub> | 150   | mW   |



## **Anexo IV**

### **Tabela Resumo da Corrente**

**Máxima Permitida em cada Circuito**



|                             |                   |           |                 |
|-----------------------------|-------------------|-----------|-----------------|
| fmax.[MHz]                  | Frequência máxima | R[mOhm]   | Resistência     |
| C[mm]                       | Comprimento       | L min[mm] | Largura mínima  |
| A[mm2]                      | Área              | L max[mm] | Largura máxima  |
| Espessura do cobre =0,035mm |                   | Imax [A]  | Corrente máxima |

| Sinal | fmax.[MHz] | C[mm]   | A[mm2] | R[mOhm] | L min [mm] | L max [mm] | Imax [A] |
|-------|------------|---------|--------|---------|------------|------------|----------|
| D18   | 2075.96    | 144.415 | 0.011  | 235.55  | 0.305      | 0.305      | 0.95     |
| D19   | 2125.58    | 141.044 | 0.011  | 230.05  | 0.305      | 0.305      | 0.95     |
| GND   | 2527.07    | 118.636 | 0.011  | 193.50  | 0.305      | 0.406      | 0.95     |
| D12   | 2594.97    | 115.531 | 0.011  | 188.44  | 0.305      | 0.305      | 0.95     |
| D27   | 2640.60    | 113.535 | 0.011  | 185.18  | 0.305      | 0.305      | 0.95     |
| D28   | 3124.15    | 95.962  | 0.011  | 156.52  | 0.305      | 0.305      | 0.95     |
| D13   | 3197.01    | 93.775  | 0.011  | 152.95  | 0.305      | 0.305      | 0.95     |
| D21   | 3322.04    | 90.246  | 0.011  | 147.20  | 0.305      | 0.305      | 0.95     |
| D20   | 3339.66    | 89.770  | 0.011  | 146.42  | 0.305      | 0.305      | 0.95     |
| +3V3  | 3522.43    | 85.112  | 0.011  | 138.82  | 0.305      | 0.406      | 0.95     |
| D3    | 3729.33    | 80.390  | 0.011  | 131.12  | 0.305      | 0.305      | 0.95     |
| N\$70 | 3750.95    | 79.926  | 0.011  | 130.36  | 0.305      | 0.406      | 0.95     |
| D29   | 3788.56    | 79.133  | 0.011  | 129.07  | 0.305      | 0.305      | 0.95     |
| D0    | 4036.40    | 74.274  | 0.011  | 121.14  | 0.305      | 0.305      | 0.95     |
| D50   | 4207.60    | 71.252  | 0.011  | 116.22  | 0.305      | 0.305      | 0.95     |
| D52   | 4224.20    | 70.972  | 0.011  | 115.76  | 0.305      | 0.305      | 0.95     |
| D2    | 4245.95    | 70.609  | 0.011  | 115.17  | 0.305      | 0.305      | 0.95     |
| D51   | 4299.13    | 69.735  | 0.011  | 113.74  | 0.305      | 0.305      | 0.95     |
| RESET | 4432.14    | 67.642  | 0.011  | 110.33  | 0.305      | 0.305      | 0.95     |
| D1    | 4743.41    | 63.203  | 0.011  | 103.09  | 0.305      | 0.305      | 0.95     |
| +24V  | 5158.96    | 58.112  | 0.036  | 28.44   | 1.016      | 1.016      | 3.01     |
| D9    | 6082.44    | 49.289  | 0.011  | 80.39   | 0.305      | 0.305      | 0.95     |
| D6    | 7078.19    | 42.355  | 0.011  | 69.08   | 0.305      | 0.305      | 0.95     |
| N\$55 | 7709.63    | 38.886  | 0.036  | 19.03   | 1.016      | 1.016      | 3.01     |
| N\$56 | 8016.36    | 37.399  | 0.036  | 18.30   | 1.016      | 1.016      | 3.01     |
| D+    | 8889.57    | 33.725  | 0.011  | 55.01   | 0.305      | 0.305      | 0.95     |
| K13   | 9108.81    | 32.913  | 0.036  | 16.10   | 1.016      | 1.016      | 3.01     |
| K33   | 9108.81    | 32.913  | 0.036  | 16.10   | 1.016      | 1.016      | 3.01     |
| K23   | 9409.60    | 31.861  | 0.036  | 15.59   | 1.016      | 1.016      | 3.01     |
| N\$64 | 9640.73    | 31.097  | 0.011  | 50.72   | 0.305      | 0.305      | 0.95     |
| K31   | 9666.85    | 31.013  | 0.036  | 15.18   | 1.016      | 1.016      | 3.01     |
| D17   | 9903.31    | 30.273  | 0.011  | 49.38   | 0.305      | 0.406      | 0.95     |
| D16   | 9982.88    | 30.031  | 0.011  | 48.98   | 0.305      | 0.305      | 0.95     |
| N\$66 | 10332.04   | 29.017  | 0.011  | 47.33   | 0.305      | 0.305      | 0.95     |
| K11   | 10678.38   | 28.075  | 0.036  | 13.74   | 1.016      | 1.016      | 3.01     |
| K21   | 11399.70   | 26.299  | 0.011  | 42.89   | 0.305      | 1.016      | 0.95     |
| D-    | 11483.36   | 26.107  | 0.011  | 42.58   | 0.305      | 0.305      | 0.95     |
| N\$73 | 12729.29   | 23.552  | 0.011  | 38.41   | 0.305      | 0.305      | 0.95     |
| N\$75 | 12729.29   | 23.552  | 0.011  | 38.41   | 0.305      | 0.305      | 0.95     |
| N\$71 | 12733.37   | 23.544  | 0.011  | 38.40   | 0.305      | 0.406      | 0.95     |
| N\$63 | 13562.52   | 22.105  | 0.011  | 36.05   | 0.305      | 0.305      | 0.95     |
| A15   | 15061.13   | 19.906  | 0.011  | 32.47   | 0.305      | 0.305      | 0.95     |
| A14   | 15728.82   | 19.061  | 0.011  | 31.09   | 0.305      | 0.305      | 0.95     |
| A5    | 17135.52   | 17.496  | 0.011  | 28.54   | 0.305      | 0.305      | 0.95     |

|       |          |        |       |       |       |       |      |
|-------|----------|--------|-------|-------|-------|-------|------|
| A13   | 17159.27 | 17.472 | 0.011 | 28.50 | 0.305 | 0.305 | 0.95 |
| OV    | 17605.05 | 17.029 | 0.011 | 27.78 | 0.305 | 0.406 | 0.95 |
| K12   | 17646.71 | 16.989 | 0.036 | 8.31  | 1.016 | 1.016 | 3.01 |
| K32   | 17646.71 | 16.989 | 0.036 | 8.31  | 1.016 | 1.016 | 3.01 |
| K22   | 17835.26 | 16.809 | 0.036 | 8.23  | 1.016 | 1.016 | 3.01 |
| SCK   | 17845.96 | 16.799 | 0.011 | 27.40 | 0.305 | 0.305 | 0.95 |
| A6    | 18847.19 | 15.907 | 0.011 | 25.94 | 0.305 | 0.305 | 0.95 |
| A12   | 18875.93 | 15.883 | 0.011 | 25.91 | 0.305 | 0.305 | 0.95 |
| VUSB  | 19307.13 | 15.528 | 0.011 | 25.33 | 0.305 | 0.406 | 0.95 |
| N\$58 | 19408.08 | 15.447 | 0.011 | 25.20 | 0.305 | 0.305 | 0.95 |
| N\$59 | 19571.36 | 15.318 | 0.011 | 24.98 | 0.305 | 0.305 | 0.95 |
| MISO  | 19594.66 | 15.300 | 0.011 | 24.96 | 0.305 | 0.305 | 0.95 |
| MOSI  | 20759.75 | 14.441 | 0.011 | 23.55 | 0.305 | 0.305 | 0.95 |
| A7    | 20781.54 | 14.426 | 0.011 | 23.53 | 0.305 | 0.305 | 0.95 |
| A11   | 20781.54 | 14.426 | 0.011 | 23.53 | 0.305 | 0.305 | 0.95 |
| A10   | 22074.52 | 13.581 | 0.011 | 22.15 | 0.305 | 0.305 | 0.95 |
| A8    | 22074.52 | 13.581 | 0.011 | 22.15 | 0.305 | 0.305 | 0.95 |
| N\$29 | 22328.81 | 13.427 | 0.036 | 6.57  | 1.016 | 1.016 | 3.01 |
| N\$41 | 22328.81 | 13.427 | 0.036 | 6.57  | 1.016 | 1.016 | 3.01 |
| A9    | 23539.07 | 12.736 | 0.011 | 20.77 | 0.305 | 0.305 | 0.95 |
| N\$53 | 25356.28 | 11.824 | 0.011 | 19.28 | 0.305 | 0.305 | 0.95 |
| N\$51 | 25356.28 | 11.824 | 0.011 | 19.28 | 0.305 | 0.305 | 0.95 |
| N\$45 | 27001.34 | 11.103 | 0.011 | 18.11 | 0.305 | 0.305 | 0.95 |
| N\$48 | 28407.63 | 10.554 | 0.011 | 17.21 | 0.305 | 0.305 | 0.95 |
| N\$47 | 28619.91 | 10.475 | 0.011 | 17.09 | 0.305 | 0.305 | 0.95 |
| N\$46 | 28639.24 | 10.468 | 0.011 | 17.07 | 0.305 | 0.305 | 0.95 |
| CS    | 32466.42 | 9.234  | 0.011 | 15.06 | 0.305 | 0.305 | 0.95 |
| XTAL2 | 33454.25 | 8.961  | 0.011 | 14.62 | 0.305 | 0.305 | 0.95 |
| XTAL1 | 34375.81 | 8.721  | 0.011 | 14.22 | 0.305 | 0.305 | 0.95 |
| N\$69 | 39066.33 | 7.674  | 0.011 | 12.52 | 0.305 | 0.305 | 0.95 |
| N\$65 | 40442.33 | 7.413  | 0.011 | 12.09 | 0.305 | 0.305 | 0.95 |
| N\$42 | 43512.34 | 6.890  | 0.011 | 11.24 | 0.305 | 0.305 | 0.95 |
| N\$44 | 45818.35 | 6.543  | 0.011 | 10.67 | 0.305 | 0.305 | 0.95 |
| N\$54 | 55518.52 | 5.400  | 0.036 | 2.64  | 1.016 | 1.016 | 3.01 |
| N\$57 | 55518.52 | 5.400  | 0.036 | 2.64  | 1.016 | 1.016 | 3.01 |
| N\$68 | 60367.19 | 4.966  | 0.011 | 8.10  | 0.305 | 0.305 | 0.95 |
| N\$50 | 61484.18 | 4.876  | 0.011 | 7.95  | 0.305 | 0.305 | 0.95 |
| N\$49 | 61484.18 | 4.876  | 0.011 | 7.95  | 0.305 | 0.305 | 0.95 |
| N\$52 | 61484.18 | 4.876  | 0.011 | 7.95  | 0.305 | 0.305 | 0.95 |
| N\$43 | 63153.31 | 4.747  | 0.011 | 7.74  | 0.305 | 0.305 | 0.95 |
| N\$60 | 68919.54 | 4.350  | 0.011 | 7.10  | 0.305 | 0.305 | 0.95 |
| N\$61 | 68919.54 | 4.350  | 0.011 | 7.10  | 0.305 | 0.305 | 0.95 |
| AREF  | 79246.40 | 3.783  | 0.011 | 6.17  | 0.305 | 0.305 | 0.95 |
| N\$62 | 92234.34 | 3.250  | 0.011 | 5.30  | 0.305 | 0.305 | 0.95 |
| N\$67 | 92234.34 | 3.250  | 0.011 | 5.30  | 0.305 | 0.305 | 0.95 |

# Auxiliary Digital Control Unit for Capacitor Banks

P. Nunes

ISEL – Instituto Superior de  
Engenharia de Lisboa  
Lisboa, Portugal  
patricia.cnunes.94@gmail.com

M. Seixas

ISEL – Instituto Superior de  
Engenharia de Lisboa  
Lisboa, Portugal  
mafalda.seixas@gmail.com

R. Melício Senior Member IEEE  
Departamento de Física, Escola de  
Ciência e Tecnologia  
Universidade de Évora, Portugal  
ruimelicio@gmail.com

**Abstract**—This paper presents the study, development and implementation of a solution to control a capacitor bank, using the schedules provide by the Regulator of Energy Services in Portugal. The solution is established for a functional prototype, that automatically manages the date and time for some certain functions. This equipment uses a microcontroller for is hardware and the arduino platform for is software, although the hardware was created using the tool Eagle. The Eagle is a software that allows to conceive our own printed circuit board (PCB). The equipment was tested and designed for an industrial use. It supports electromagnetic interferences and a certain limit of thermal variations. It is also protected according the current standards for harmonic disturbances.

**Keywords**—Microcontroller, Eagle, Capacitor Bank, ERSE schedules.

## I. INTRODUCTION

Nowadays, the technology has a fundamental role in the industrial, economic, social and cultural development, and is favored by the continuous creation of new technological devices. The continued demand for efficiency processes of manufacturing, causes an incessant need to reduce costs in production, creating the ideal conditions to energy efficiency became one of the main concerns. This concern is especially important in the industry, because normally the process of manufacturing products uses electrical machines that needs an electric energy consumption. For the most part, these machines work by the interaction between electromagnetic fields, that are created with the aid of coils, fed with an alternating current by the electric grid. Consequently, from the grid point of view, electric motors are regarded as an inductive power load consumption. Operating inductive loads, requires three types of power: active, reactive and deformed power.

For this paper its only important to know the meaning of active and reactive power. So, the active power is the necessary one to produce work, which is consumed in the devices operation. While the reactive power does not produce work, however, it is necessary to create and maintain electromagnetic fields in inductive loads, such as in transformers or motors. Normally in the operation of electric motors excess reactive energy is generated, between the load and the power supply [1]. Having an excess of reactive power, means lower power factor values. These circumstances, create an increase in the total current flowing in the electrical distribution grid, and the consumer units, placing overloads in the substations, transmission lines and distribution. These events impair the

stability and the conditions of an electrical system as also harms the consumers, since as it represents an increase in their energy bill [1], [2]. The power factor that was establish before is the efficiency level in electrical systems. A high value of power factor namely close to one, indicates an efficient use of electric energy, while low values means a poor utilization and overload in the entire electrical system [3].

In most electrical systems, the equipment used to compensate for reactive energy is the capacitor banks, which are normally placed next to the inductive loads [4]. In Portugal, the energy services are an activity regulated by ERSE. This entity is responsible for the regulation of electrical and natural gas sectors, and aims to take care of consumers interests and to encourage a more efficient use of energy. Accordingly, in order to improve energy efficiency, ERSE use regulations to control the circulation of reactive power in the national grid, imposing penalties to achieve this purpose [5].

To impose the schedules, the companies usually use commercial programmable controllers (PLCs). This paper proposes a fast and robust solution, created using the microcontroller arduino to replace the known PLC. The arduino was created to provide any individual the ability to enjoy the capabilities of an electronic equipment, created by themselves. That means from programmers, entrepreneurs or experts in electronics to the curious without any training. For this reason, arduino brings numerous advantages, the easy way of using, the inexpensive costs and the flexibility, even enough for the advanced uses. Briefly, arduino is an open source electronic platform, which provides an electronic prototyping tools, that aims to make it easier to create electronic devices, based on simple hardware and software. Therefore, arduino “offers” the electronic board, shields to attached to the arduino board, and provides an environment for developing and sharing data (libraries), among of all those who use this platform. The language used in their software is C/C++ [6], [7].

## II. HARDWARE

### A. Architecture of the electronic board

The hardware of the equipment consists in the design of an electronic board with the following architecture.

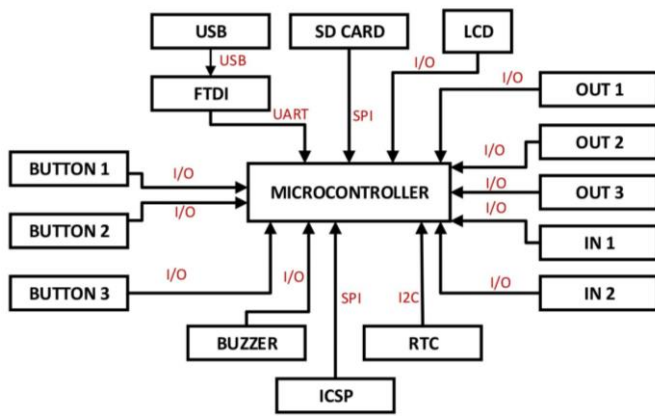


Fig. 1. – Architecture of the components to the electronic board.

The main purpose of the solution developed is to incorporate the ERSE schedules, so the basis of this project is to know every time instant. With this vision, it was selected a Real-Time Clock (RTC) and not only the internal clock of the microcontroller to know and manipulate the time. A RTC is more precise than the crystal on the microcontroller and doesn't require an external battery to save instants of time, and also, besides that it is provided, with an integrated temperature compensation. The RTC can store information of the seconds, minutes, hours, days, days of the week, months and years. It also automatically adjusts months with less than thirty-one days, including leap years corrections.

This solution need as well an SD card to save the user decisions, a LCD with buttons to visualize and interact with the user, a buzzer as a beep to warn, a USB connection to link the device to a computer, inputs and outputs. Finally, it requires the following communication protocols:

The Peripheral Interface (SPI) used by microcontrollers to communicate with one or more peripherals components, quickly and over short distances. This protocol can also be used between two microcontrollers. The operation of this protocol consists of a master (usually the microcontroller) that controls the peripheral devices, the slaves. For the communication to coexist, it is normally necessary in all the devices four signal lines: Master In Slave Out (MISO), Master Out Slave In (MOSI), Serial Clock (SCK) and Chip Select (CS) [6];

The Inter-Integrated Circuit (I<sup>2</sup>C), that works also with the master-slave model, which may have more than one master. This protocol also contains their own pins to establish communication, the Serial Data (SDA) and the Serial Clock (SCL). The SDA is the pin that transfers the data and receives it (bi-directional), while SCL controls the timing between the devices, to ensure a reliable communication from the SDA [6];

The Universal Serial Bus (USB) is a simple protocol that allows the transfer of data between computers and peripheral devices. On the computer is installed the host controller, while in the peripheral is installed the device controller. The host carries the polling to allow communication with all devices and authorizes it only in case, the bandwidth has an occupation less

than 90%. The communication is in serial mode and uses four lines, the power, the ground and a pair of twisted wires with differential voltage [8];

The Universal Asynchronous Receiver/Transmitter (UART) protocol, has an asynchronous communication, works with two pins, the transmission (Tx) and the receiving (Rx). The Tx sends the bits to the Rx pin of the other device, which will read them bit by bit. Each bit segment sent, contains one bit that indicates the beginning of the message, one or two stop bits to indicate the end of the message, five to nine information bits and one parity bit to avoid errors [9];

The remaining circuits shown in the figure, represent functions as inputs/outputs (I/O). It's important to mention, that it was necessary to use a Future Technology Devices International (FTDI) to convert the USB protocol into UART, otherwise the microcontroller couldn't communicate with the USB peripheral device.

### B. PCB Design and Conception

The PCB created in this propose has two layers, which means that it contains tracks recorded on the top and bottom layers. The paths in the different layers are linked through holes with a metallic coating inside, accomplished from an electrochemical process. These metal holes are called vias. The vias can also be used for fixing and soldering components terminals [10].

To design a PCB, its needed a software that allows, design and create the files needed to send to a manufacturer in order to create it. The chosen software was Eagle CAD [11], because it easily allows to design the PCB in six steps:

#### ➤ Step one: Schematic

This step consists in designing the circuit that will integrate the PCB, considering the previously established architecture and simultaneously selecting the necessary devices and checking if they exist in the market. Most external devices needed, already have their footprint available in the arduino platform to Eagle. The devices for which the design of the equipment for this software is not provided, must be drawn. To do this, the data sheet and the tools of the Eagle are used to create the schematic. In addition, to ensure the correct function of the system, it was necessary to estimate the value of components, as capacitors or resistances.

An example where it was necessary to add capacitors was in the DC-DC converter from 24 VDC to 5 VDC. This converter exists because the relays of the outputs are fed at 24 VDC.

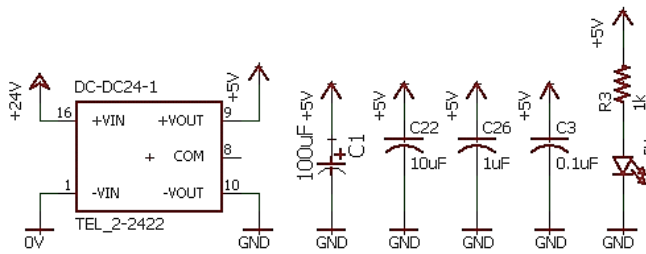


Fig. 2. – Eagle Schematic for the DC-DC converter.

The Fig.3 shows a lot of different values on to the capacitors (interval of one decade). The intention of these is to stabilize the voltage at different frequencies (switching). As can be seen, one of the capacitors is electrolytic and the remain are ceramic. This decision was intentional, because an electrolytic capacitor, usually has more energy, which better stabilizes the voltage, however the response is slower, because of the Equivalent Series Resistance (ESR). So, for high frequencies is better to use the ceramic capacitors.

Also, it is shown in the Fig.2 a led. This led indicates the existence of power in the 5 VDC section. That way it provides an easy visualization of the existence of feeding mistakes.

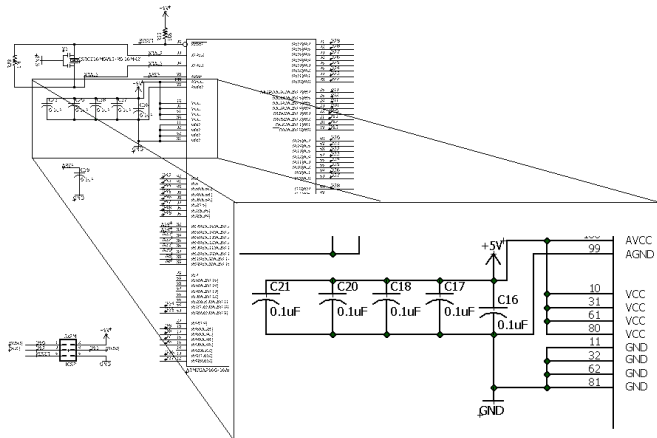


Fig. 3. – Eagle Schematic for the bypass capacitors in the microcontroller.

In all the chips, a capacitor is placed next to each power pin, as for example in the microcontroller chip, there are five power pins, so it is need five capacitors. These capacitors have bypass operation. This means that they are used to reduce high current frequencies in a high impedances path, diverging it to another path using a capacitor. In other words, they are used to ignore the noise of the current existent many times in the electronic circuits feeds, preventing distortions in the voltage bus. The bypass capacitors have typical values, such as 1 µF and 0.1 µF [12], [13].

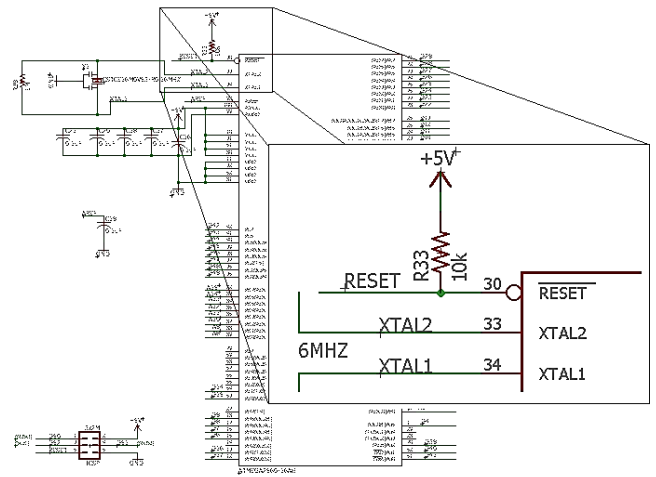


Fig. 4. – Eagle Schematic for the pull-up resistance in the microcontroller.

In any electronic digital logic device, it is common to provide to in all the entrances pull-up resistors (exception of those that already have it internally). The function of this resistors is to ensure that the inputs are in the expected logical levels, that is logic level one. So, the concept of this resistance is to “pull” the voltage to the value of who is connected. Without this resistance, the logic state of the input would be floating, namely in high logic state or low logic state [14]. As the Fig.5 shows, in the microcontroller there is a pull-up resistance in the reset pin.

An essential component to the operation of the system is the In-Circuit Serial Programming (ICSP), which has six pins, that allows to download the operating system (Bootloader), to make the electronic board compatible with the arduino platform, using an Atmel microcontroller [6], [14], as the Fig.5 shows.

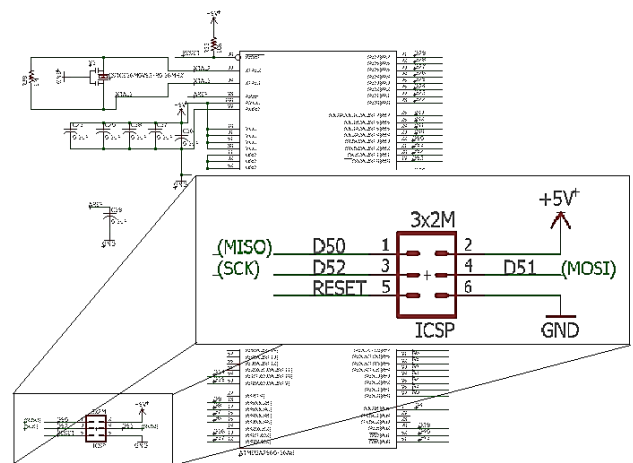


Fig. 5. – Eagle Schematic for the ICSP in the microcontroller.

Low-pass filters are used at the input buttons to allow the passage only of the low frequencies and to attenuate the frequencies with an amplitude bigger than the cut frequency. The value of the cut frequency ( $f_c$ ) was selected to be higher than the expected frequency at the input button, but simultaneously lower than the lower frequency used in all the

system. The lowest frequency present in the system is the PWM, which has a default value the 490 Hz [6], [15].

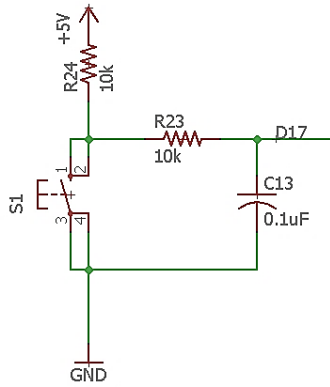


Fig. 6. – Eagle Schematic for one button.

$$f_c = \frac{1}{2\pi R_{23} C_{13}} = \frac{1}{2\pi * 10 * 10^3 * 0.1 * 10^{-6}} \cong 159 \text{ Hz} \quad (1)$$

➤ Step two: Build of Material (BOM)

The BOM is a list of all the devices used in the schematic. This is generated automatically by the software and with that list it is possible to find one or more manufacturers and withdraw the reference of each product.

This step is crucial to know if the selected devices exists in the common electronic suppliers, because even if a device has the online schematic available, does not mean that it exists in the market. Selecting the suppliers makes it possible to estimate the cost of the PCB and making a budget.

➤ Step three: Layout

The third step is one of the most important, if not the most important steps, because a good layout is about 90 % responsible for a good implementation of the PCB. The layout consists of placing the devices in a restricted area, which have the desired dimensions for the PCB. The devices are placed according to the linking of the designer, being possible several layouts with the same components. Of course, if there are links between devices, these should be arranged close to each other, however it is still possible to create different layouts.

The design of the PCB on this project is divide by the different voltage levels existing in the schematic, as the Fig.7 shows. According to the IPC-2221, there are minimum distances between voltage levels. So, between the voltage level 230/400 VAC and 24 VDC there must be a distance of at least 0.8 mm, and between the voltage level 24 VDC and 5 VDC there must be at least 0.05 mm [16].

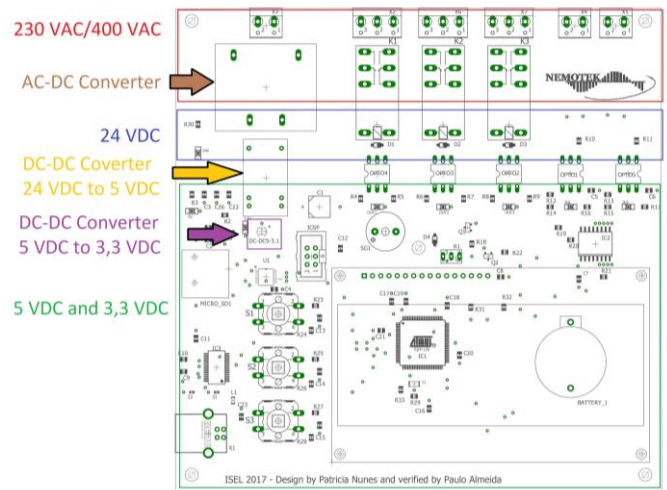


Fig. 7. – The voltage levels in the PCB design.

The chosen design was to place the microcontroller as central as possible in the 5 VDC/3.3 VDC level zone, because there are several devices connected to him. It was decided to place the LCD in a corner, in order to put the three buttons near to it.

The battery of the RTC was placed in a strategic position and isolated from the other devices, since it is not necessary for them to use it. Obviously the RTC was placed near the battery.

➤ Step four: Routing

Once the layout of the PCB is defined, it is necessary to make the connections between the devices, constructing the circuits of the schematic. Firstly, it is created the perimeter of the layer representing the positive voltages (230/400 VAC, 24 VDC, 5 VDC), laid in the top layer. Next is represented in the lower layer the respective ground of each voltage level. In this way, these layers are galvanically isolated, preventing currents circulation over common impedances in various circuits [15].

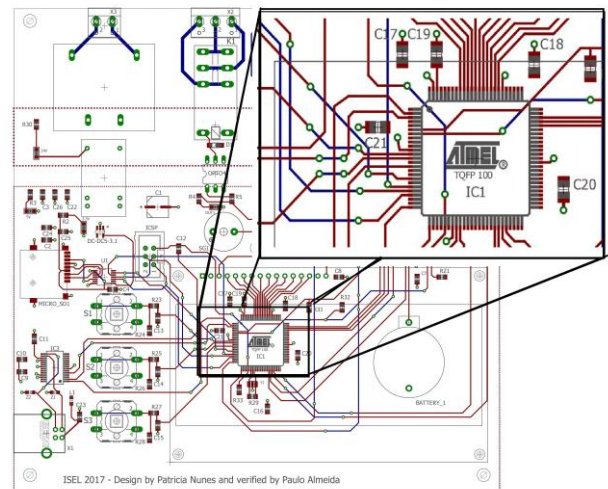


Fig. 8. – The circuits in the PCB, where red represents the top layer and the blue represents the bottom layer.

As the previous Fig.8, the connection between the pins always starts with the red layer (top). If the component is placed in the top layer, and to avoid overlapping connections, it is need to put the remaining circuit in the back of the PCB, changing to the blue layer. In the connection of the layers exchange, it exists a metalized hole (via), which allows the conduction between the top and the bottom track.

The paths in the layers can have different widths, depending of the manufacturer requirements. In this project, it was used only two widths, 40 mil (thousandth of an inch) to the 230/400 VAC level and 12 mil to the 24 VDC and 5 VDC level. The maximum current of the tracks is established by the width of the tracks themselves, according to the IPC-2221 standards [16].

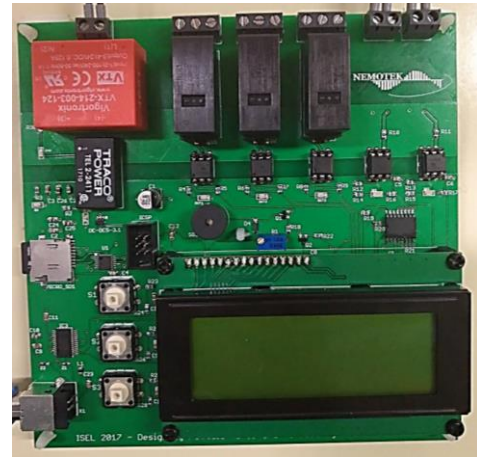
➤ Step five: Design Rules Check (DRC)

The DCR is a group of rules and parameters which determines whether the design of a PCB can be built. This rules must be download depending on the chosen manufacturer. The manufacturer selected for this project was ITEAD, therefore there were download the requirements of this manufacturer to the Eagle software. The great advantage of Eagle software is to indicate the exact location where a requirement is not being observed and which requirement. With this, it is possible to change the design correctly and faster.

➤ Step six: Gerber's

The last step is to create the Gerber's, that is, the files needed for the manufacturer to create the PCB. These files are: Gerber Bottom Copper (GBL), Gerber Top Copper (GTL), Gerber Bottom Solder Mask (GBS), Gerber Top Solder Mask (GTS), Gerber Top Solder Mask (GTS), Gerber Top Silk-screen (GTO) and a document with the extension .txt, indicating the coordinates where drills exist and their dimensions. It is possible to visualize each one of these files in the online simulators and verified if they are exactly the way we wanted, because they correspond how the PCB will be.

Finally, having the PCB done, it was needed to solder the components, creating a circuit with the following aspect.



b)

Fig. 9. – The PCB with the components a) without LCD b) with LCD.

### III. SOFTWARE AND FINAL RESULTS

The equipment developed is intend to control the on/off switching of a capacitor bank, (from the power supply of the varmeter controller) and/ or to control the switching of a step (capacitor) according to a set of predetermined schedules.

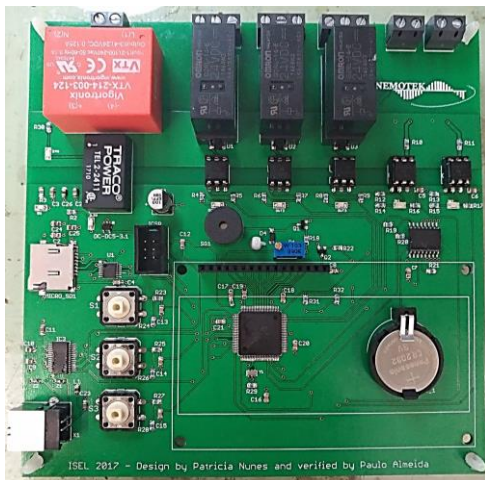
The algorithm uses the arduino platform (C/C++ language), and is organized according several routines, that is fractions of code to perform specific tasks. The algorithm operates according to the established format in arduino, that is, a main program, a section of variables declaration and a section for the libraries. A huge part to a good execution of this project, is the strategy used as an interface between the user and the equipment. For that, it was decided to use an SD card to storage the user decisions. So inside the SD card there is a document in Excel that allows the user to make those decisions. This Excel document, is the one who creates the several .txt documents which will be read by the equipment. Another alternative for the user is to manipulate the buttons available in the equipment to make the decisions. That information is, once more, storage in the SD card, to be read when needed.

#### A. Interface between the user and the equipment

As already has been said, it is necessary to previously define the operating conditions for the developed equipment precedes, which are decisions made by the user. The user can communicate with the equipment in two different ways, by an Excel document or by manipulation of the buttons available in the equipment.

➤ Excel Document

The Excel document has three sheets, one where are written the holidays over several years of the place where the equipment will be located, another for the configurations from the user decisions and finally other sheet that contains the ERSE schedules.



a)

|    | A                   | B     | C           | D |
|----|---------------------|-------|-------------|---|
| 1  | 01/01/1970          | Days  | Seconds     |   |
| 2  | 01/01/2017 00:00:00 | 17167 | ;1483228800 |   |
| 3  | 28/02/2017 00:00:00 | 17225 | ;1488240000 |   |
| 4  | 14/04/2017 00:00:00 | 17270 | ;1492128000 |   |
| 5  | 16/04/2017 00:00:00 | 17272 | ;1492300800 |   |
| 6  | 25/04/2017 00:00:00 | 17281 | ;1493078400 |   |
| 7  | 01/05/2017 00:00:00 | 17287 | ;1493596800 |   |
| 8  | 10/06/2017 00:00:00 | 17327 | ;1497052800 |   |
| 9  | 13/06/2017 00:00:00 | 17330 | ;1497312000 |   |
| 10 | 15/06/2017 00:00:00 | 17332 | ;1497484800 |   |
| 11 | 15/08/2017 00:00:00 | 17393 | ;1502755200 |   |
| 12 | 05/10/2017 00:00:00 | 17444 | ;1507161600 |   |
| 13 | 01/11/2017 00:00:00 | 17471 | ;1509494400 |   |

Fig. 10. – Excel sheet for the holidays insertion.

The user types in column A, as many holidays as necessary. The limit is 500, but it can be increased. As it can be seen in the Fig.10, the holidays are declared with data and time, where time is always zero hours, zero minutes and zero seconds, for programming specifications. The column C automatically converts the data to seconds, which is the value that will be read by the device.

The second sheet is the configuration, which represents the decisions that should be made by the user. It is important for the user to be aware that the region of the country where the capacitor bank will be located must be selected, as well as the tariff at which the device will operate.

The user also should be aware to the existence of two inputs, that can operate in two different modes (but each input can only work in one single mode), as an alarm or as control. Alarm means, when the input is not active, the output is functioning normally, when the input is active, all the outputs are turned off and a beep sounds. If it works as control, in this case the user can chose how the output works, only controlled by the input, by the input and the schedules or only by the schedules. The device is designed only to control a maximum of three outputs.

The user makes decisions by clicking in the Excel sheet. Depending on the previous decisions the options of the sheet fields change.

The final sheet is divide by legal winter time and the legal summer time and further subdivided into regions and tariffs. Whenever there is a change in the schedules by the ERSE entity, it is possible to simply change this sheet.

|    | A                                | B                  | D            |
|----|----------------------------------|--------------------|--------------|
| 1  |                                  |                    |              |
| 2  | Botão TXT                        |                    |              |
| 3  |                                  | Ciclo Semanal      |              |
| 4  | Período de hora legal de Inverno | De segunda a sexta | Sábado       |
| 5  | 00:00 h                          | Vazio Normal       | Vazio Normal |
| 6  | 00:15 h                          | Vazio Normal       | Vazio Normal |
| 7  | 00:30 h                          | Vazio Normal       | Vazio Normal |
| 8  | 00:45 h                          | Vazio Normal       | Vazio Normal |
| 9  | 01:00h                           | Ponta Cheias       | Vazio Normal |
| 10 | 01:15 h                          | Vazio Normal       | Vazio Normal |
| 11 | 01:30 h                          | Super Vazio        | Vazio Normal |
| 12 | 01:45 h                          | Vazio Normal       | Vazio Normal |
| 13 | 02:00 h                          | Super Vazio        | Super Vazio  |
| 14 | 02:15 h                          | Super Vazio        | Super Vazio  |
| 15 | 02:30 h                          | Super Vazio        | Super Vazio  |
| 16 | 02:45 h                          | Super Vazio        | Super Vazio  |
| 17 | 03:00 h                          | Super Vazio        | Super Vazio  |
| 18 | 03:15 h                          | Super Vazio        | Super Vazio  |
| 19 | 03:30 h                          | Super Vazio        | Super Vazio  |
| 20 | 03:45 h                          | Super Vazio        | Super Vazio  |
| 21 | 04:00 h                          | Super Vazio        | Super Vazio  |
| 22 | 04:15 h                          | Super Vazio        | Super Vazio  |
| 23 | 04:30 h                          | Super Vazio        | Super Vazio  |
| 24 | 04:45 h                          | Super Vazio        | Super Vazio  |
| 25 | 05:00 h                          | Super Vazio        | Super Vazio  |

Fig. 11. – Excel sheet for the ERSE schedules.

### ➤ Button manipulation

The routine where the respective code is inserted to accommodate the user decisions, starts with the verification of the existence or the absence of the configuration document. If it does not exist, questions are asked through an LCD, to which the user responds with buttons. The questions arise depending on the answers previously given.

The user makes the decision, selecting the button corresponding to option one, two or three. The buttons are arranged in order from top to bottom. So, the first button counting from top, selects the option 1 and so on.

```

//****Seleção da Região*****
lcd.setCursor(0, 0);
lcd.print("Selecione a Região ");
lcd.setCursor(0, 1);
lcd.print("1-Portugal");
lcd.setCursor(0, 2);
lcd.print("2-Acores");
lcd.setCursor(0, 3);
lcd.print("3-Madeira");

Regiao = ReadButtons(3);

Alarm.delay(1000);

```

Fig. 12. – First question in the code.

The function who process the choice appears in the Fig. 12 named ReadButtons. In parentheses, is shown how many buttons are available through each question. In this case, the value 1,2, or 3 is stored in the variable named “Regiao”, and it is this variable that will be read or write in the text document saved in the SD card.

### B. Algorithm, some code

When a new document is created using the arduino program, two functions, setup and loop are available. The setup function is a fraction of code that is only requested, when the program starts or when is restarted. The loop function is the fraction of code that repeats itself continuously, allowing the program to work dynamically [6].

As already mentioned, the code is distributed according to a main program, which includes the two above described functions, various routines; a section for variable declaration and a section for libraries.

```

//*****Livrarias*****
const int chipSelect = 9;
#include <LiquidCrystal.h>
#include <Wire.h>
#include <DS3231.h>
#include <SPI.h>
#include <SD.h>
#include <TimeAlarms.h>
#include <TimeLib.h>
//*****
// Nome do ficheiro para leitura e escrita no cartão SD
File myFile;
// Pinos conetados ao LCD
LiquidCrystal lcd(A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15);
// Declaração dos pinos do RTC
DS3231 rtc(SDA, SCL);

```

Fig. 13. – Section of code for the libraries.

The first library is to control the LCD; the second allows the communication with devices with I<sup>2</sup>C protocol; the third allows to get the time and date using the external RTC; the fourth library allows the communication of the microcontroller with peripheral devices using SPI protocol; the fifth allows to read and write on the SD card; the sixth provides the date and time creating alarms, which are used to call functions at specific times; and the final library allows to get the time and date with the format second, minute, hour, day, moth and year [6].

```

int HoraDoDia;
Time t;
TimeElements tm;

String NomeFicheiro = "FFFFFF";
unsigned long Agora, Horas, Temporizador, botaol3, tempobotaol, tempobotaol2;
unsigned long manutencao, botaol2, ultimaData;
boolean FeriadoDomingo, sabado, Seg_sexta, Verao;
boolean Escalao_S1, BC_S1, Escalao_S2, BC_S2, BC_S3, Escalao_S3;
boolean Registo_saidal, RegistoAnterior_saidal;
boolean Registo_saida2, RegistoAnterior_saida2;
boolean Registo_saida3, RegistoAnterior_saida3;
boolean Feriado3meses;
byte Comando;
unsigned long TempoOFF1, TempoOFF2, TempoOFF3;

int IDAlarmeS1Ligar, IDAlarmeS2Ligar, IDAlarmeS3Ligar;
int IDAlarmeS1Desligar, IDAlarmeS2Desligar, IDAlarmeS3Desligar;

```

Fig. 14. – Section of code for the variables declaration.

In the Fig.14, it is possible to see five types of data used:

- Int - Represents the integers and can store values with 16 bits (2 bytes), so the range is the -32 768 to 32 767 (in arduino UNO e MEGA);
- String - Allows the using and manipulation of text sequences;
- Unsigned long – Allows to storage numbers of 32 bits (4 bytes) and unlike Int, does not store negative numbers, obtaining a range of 0 to 4 294 967 269;
- Boolean – Each variable of this type occupies 1 byte of memory, and can only contain two states, true or false (1 or 0);

- Byte – It is a byte that store a number with 8 bits from 0 to 255 [6].

```

void setup() { // Função executada somente uma vez,
//quando o Arduino é iniciado ou reiniciado

// Inicialização do RTC
rtc.begin();
// Inicialização do LCD(20 colunas por 4 linhas)
lcd.begin (20, 4);
// Inicialização da porta série do PC
Serial.begin(9600);

InicializarTempo();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Cartao SD");

if (!SD.begin(9)) {
  lcd.setCursor(0, 1);
  lcd.print("Inicializacao falhou!");
  Alarm.delay(3000);
  lcd.clear();
  return;
}
lcd.setCursor(0, 1);
lcd.print("Inicializacao ok.");
Alarm.delay(3000);
lcd.clear();

VerificarVerao();
VeraoInverno();
Alarmes();
VerificarHora();

pinMode(Button1Pin, INPUT_PULLUP);
pinMode(Button2Pin, INPUT_PULLUP);
pinMode(Button3Pin, INPUT_PULLUP);
//*****
pinMode(12, OUTPUT);
digitalWrite(12, HIGH);
//*****
//Entrada 1
pinMode(18, INPUT);

//Entrada 2
pinMode(19, INPUT);

Alarm.delay(50);

attachInterrupt(5, BotaolFALLING, FALLING);
attachInterrupt(4, Botao2FALLING, FALLING);

Alarm.delay(50);

EscolhaTarifario();

if (Regiao == 1)VerificarDia();

CiclosTarifario();

pinMode(27, OUTPUT); //saida 1
pinMode(28, OUTPUT); //saida 2
pinMode(29, OUTPUT); //saida 3

SaidasComHorario();
}

```

Fig. 15. – Section of code for the setup.

The setup begins with the initialization of the RTC, the LCD and the serial port. Then the “IniciarTempo()” routine is called. This routine uses the RTC library to write easily and without any extra programming, the time, date and day of the week. The next code appearing, is to guaranteed the correct placing of the SD card, to be read or write by the device.

Next, four other routines are called, the “VerificarVerao( )”, to check if the current day belongs to the legal summer time, or to the legal winter time; the next routine called is the “VeraoInverno( )”, this one, is to advance or delay the hour, if the actual day is the last one belonging to the legal winter/summer time, this is done with an alarm with the specific hour to change; after it emerges the “Alarmes( )” routine, this one is to put a lot of alarms, specifically the 15 minutes intervals, because the ERSE schedules only have eventual alterations on that time interval, making only the device reading the SD card when needed; the last routine called is “VerificarHora( )”, which is important, only in case of power failure, because it process what is the current time and day, insuring that the device works in the right set of the process.

Then it is initialized the three buttons, the pin for the light adjustment for the LCD, the inputs using interrupts. There are called four more routines and the outputs are initialized. The “EscolhaTarifario( )” routine is the one which have the questions/decision for the user; the “VerificarDia( )” is only for a specific task, because in the schedules the day of the week only matters when the region is Portugal; the “CiclosTarifario( )”, is to select the text document to be read depending on the region and tariff chosen; the final routine, the “SaidasComHorario( )” is to verify and change the output condition.

To use the “attachInterrupt( )” function in the inputs, it is need the arduino board model, because the number of the interrupt and the pin is different , like the Fig.16 shows.

| Board                            | int.0                         | int.1 | int.2 | int.3 | int.4 | int.5 |
|----------------------------------|-------------------------------|-------|-------|-------|-------|-------|
| Uno, Ethernet                    | 2                             | 3     |       |       |       |       |
| Mega2560                         | 2                             | 3     | 21    | 20    | 19    | 18    |
| 32u4 based (e.g Leonardo, Micro) | 3                             | 2     | 0     | 1     | 7     |       |
| Due, Zero, MKR1000, 101          | interrupt number = pin number |       |       |       |       |       |

Fig. 16. – Interrupts associated to the arduino boards [6].

The arduino MEGA, used in this project can actuate by four modes: Low, Change, Rising and Falling. The interruptions are useful when it is necessary to ensure that the state changes only in one of the previous modes, allowing it in the event of fluctuations. So, the input only takes action if needed.

```
void loop() {
    interruptBotaol();
    interruptBotaol2();
    Manutencao();
    TextoLCD();
    Botaole3Reset();
    Botaole2Reset();
    Alarm.delay(1000);
}
```

Fig. 17. – Section of code for the loop.

In the loop, because it is the code that repeats continuously, was chosen to be put the routines of the interrupts, in order to always verify their state. The routine of maintenance, which corresponds to an established date chosen by the expert. When the current date is the same of the one in the maintenance, it

gives a sound signal. It is called the routine that writes all the information needed to the user to see in the LCD, date, time, when an input is active etc.

Finally, two similar routines are called, one is when the buttons 1 and 3 are pressed simultaneously. By doing that, it is asked to the user if the decisions are to be remade by deleting the previous ones. The other routine is to turn off the sound signal of the maintenance by pressing simultaneously the buttons 1 and 2.

### C. Final Results

Before the prototype was placed in the capacitor bank cabinet, the correct operating of the code was verified. For example if the text appears correctly on the LCD, as the Fig.18 shows.

```

/*****Seleção da Região*****
lcd.setCursor(0, 0);
lcd.print("Selecione a Região ");
lcd.setCursor(0, 1);
lcd.print("1-Portugal");
lcd.setCursor(0, 2);
lcd.print("2-Acores");
lcd.setCursor(0, 3);
lcd.print("3-Madeira");

Regiao = ReadButtons(3);

Alarm.delay(1000);
a)

```

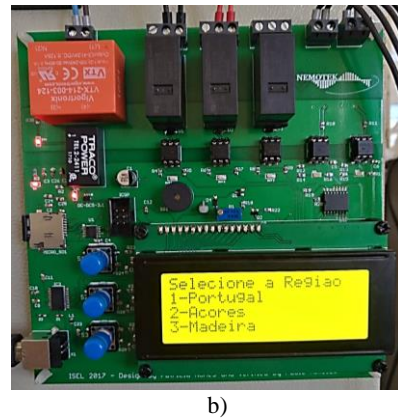


Fig. 18. – The question a) in code b) on the LCD.

The device was placed on the outside of the capacitor bank cabinet, as show in the Fig.19.



Fig. 19. – Device on the capacitor bank.

To verify the correct operation of the device, it was chosen one output connected to the varmeter controller and that output controlled by one of the inputs, to perform the switching of the capacitor bank. For the second output, it was necessary to disconnect one capacitor from the varmeter controller, and connect it to the other input.

It was found in the tests, that the normally open contact of the relays of the outputs did not work. The relays chosen contains a central contact, which is the common one, as the Fig.20 shows. When the relay has the coil deenergized, the common contact (C) makes the connection with the normally closed contact, keeping the circuit closed. When the coil is energized, the contact C closes by the normally open contact [17].

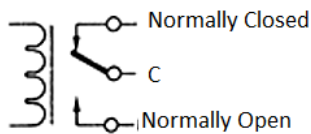


Fig. 20. – Relay Contacts.

To solve the problem, first was checked if the connector of the relay was placed correctly with welded to the PCB. Verifying that the problem did not arise from the connector, it was questioned whether the relay could be damaged. To verify it, the relay was connected to an external battery and confirmed if the contacts were working properly, again, that was not the problem. Then, it was considered that the diode in the circuit establish with the relay may by soldered incorrectly, nevertheless it was observed, that it was not the case.

To find the problem, it was decided to check with a multimeter all the voltages in the circuits. It was found that the problem was coming from the diode. The data sheet selected had a maximum repetitive reverse voltage (VRRM) of 20 V,

however the voltage used in the circuit is 24 V. It means that the diode operates similar to a Zener diode, so no matter how much current were flowing, the voltage was maintained at 20 V, which was not enough. The solution was simple, it was to replace the diode with one with an appropriate VRRM.

Another issue that was found, was the value used for the resistance on the optocoupler circuit. The basic model of an optocoupler is a led that produces infrared light and a photosensitive semiconductor device, used to detect the infrared light. The current that flows to the input of the led which emits the infrared light, contains an intensity of light proportional to the electric signal. Is this light that falls on the base of the photo-transistor, which will behave similar to a normal bipolar transistor [18].

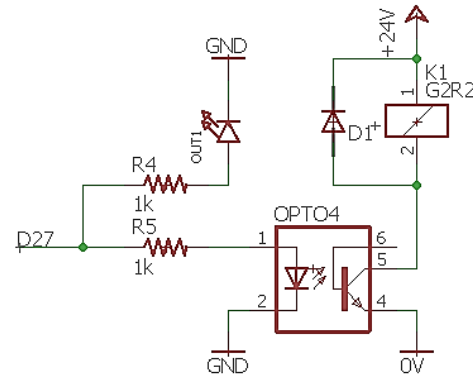


Fig. 21. – Optocoupler Circuit.

As it can be seen from the Fig.21, initially the value of the resistance (R5) of the current signal in the optocoupler input was 1 k $\Omega$ , however it was verified that this resistance value was limiting the current in the transistor. Because the higher is the value of the resistance, the lower will be the current, so the lightness will be lower as well, resulting to the not full condition of the transistor.

It was necessary to re-dimension the resistance value to avoid the previously mentioned. The current and voltage values supported by the select optocoupler is 60 mA and 5 V.

$$R = \frac{U_{OPTO}}{I_{OPTO}} = \frac{5}{0.06} \cong 83 \Omega \quad (2)$$

The value of the resistance to be placed should be 83  $\Omega$ , however according to the arduino Mega specifications, it supports a maximum of 20 mA and not 60 mA [6].

The maximum current that the microcontroller can give to the optocoupler is the following node analysis, considering the circuit of Fig. 20.

$$I_{Led} = \frac{U_{OPTO} - U_{Led}(out1)}{R4} = \frac{5 - 2.1}{1000} = 2.9 \text{ mA} \quad (3)$$

$$I''_{OPTO} = I'_{OPTO} - I_{Led} = 20 \text{ mA} - 2.9 \text{ mA} = 17.1 \text{ mA} \quad (4)$$

As it can be verified, the optocoupler can support an input current of 17.1 mA. Therefore, the resistance value is:

$$R_{\text{OPTO}} = \frac{U_{\text{OPTO}} - U_{\text{Led OPTO}}}{I''_{\text{OPTO}}} = \frac{5 - 1.3}{0.0171} \cong 22 \Omega \quad (5)$$

The minimum value supported in the input is a resistance of  $22 \Omega$ , in order to not damage the system.

After solved these two problems, it was verified the correct work of the device, performing all the tasks in the right moments, without any shielding covering the device.

#### IV. CONCLUSIONS

This paper proposes an alternative and developed device to control a capacitor bank, implementing the schedules provide by the Regulator of Energy Services in Portugal (ERSE).

The device uses the arduino platform and proceeds with two different solutions as an interface between the equipment and the user. One of the solutions uses an LCD and the buttons available, to obtain the user decisions. The other solution uses an Excel document to extract that decisions. Both solutions are storage to be read and write through an SD card.

The establish architecture was made for the device hardware considering all the procedures to build the PCB and protocols needed. To design the PCB was used the Eagle software, to be sent later to an acquired manufacturer. The experimental results showed, after been solved all the problems encountered, that the hardware and software implemented was suitable to the main objectives proposed.

This device is only a prototype, so it was designed and created to future innovations possibilities.

#### ACKNOWLEDGMENT

This work is funded and supported by the company NEMOTEK, Portugal.

#### REFERENCES

- [1] Comité de Distribuição de Energia Elétrica. Manual de Orientação aos consumidores- Energia Reativa Excedente.
- [2] Pimenta de Freitas, Tiago Joel (2014). *Qualidade de serviço em redes elétricas inteligentes*. ipb.
- [3] E. D. P, "Perguntas frequentes - faturação de energia reactiva." 2012.
- [4] Marques, Gonçalo (2015). *Compensação da Energia Reactiva com Minimização de Perturbações sobre a Rede Elétrica*. ISEL.
- [5] Leite, Carlos Miguel (2011). *Compensação da Energia Reactiva na Rede de Distribuição*. FEUP.
- [6] Arduino. Disponível em <https://www.arduino.cc>. [Consultado em 22/05/2017].
- [7] OP SERVICES. Disponível em <https://www.opservices.com.br/o-que-e-o-arduino>. [Consultado em 24/05/2017].
- [8] Computer Solutions Ltd - USB protocol. Disponível em [http://www.computer-solutions.co.uk/info/Embedded\\_tutorials/usb\\_tutorial.htm](http://www.computer-solutions.co.uk/info/Embedded_tutorials/usb_tutorial.htm). [Consultado em 18/09/2017].
- [9] Tutoriais, Comparação Entre Protocolos de Comunicação Serial. Disponível em <https://www.robocore.net/tutoriais/comparacao-entre-protocolos-de-comunicacao-serial.html>. [Consultado em 20/07/2017].
- [10] Mehl, Ewaldo. *Conceitos Fundamentais sobre Placas de Circuito Impresso*.
- [11] Eagle CAD Software. Disponível em <https://www.autodesk.com/products/eagle/overview>. [Consultado em 02/09/2017].
- [12] Kundert, Ken (2004). *Power Supply Noise Reduction*. The Designer's Guide Community.
- [13] Schmitz, Tamara; Wong, Mike (2011). *Choosing and Using Bypass Capacitors*. Intersil.
- [14] Shields. Disponível em <https://www.sparkfun.com> [Consultado em 04/07/2017].
- [15] Soares, Vasco (2008). *APS-Aquisição e Processamento de Sinais*. Instituto Superior de Engenharia de Lisboa (ISEL).
- [16] IPC-2221 (1998). *Generic Standart on Printed Board Design. From: Association Connecting Electronics Industries*.
- [17] C.Braga, Newton (2012). *Relés, Circuitos e Aplicações*. Metaltext.
- [18] *Eletronics Tutorials*. Disponível em <http://www.electronics-tutorials.ws/blog/optocoupler.html>. [Consultado em 21/08/2017].