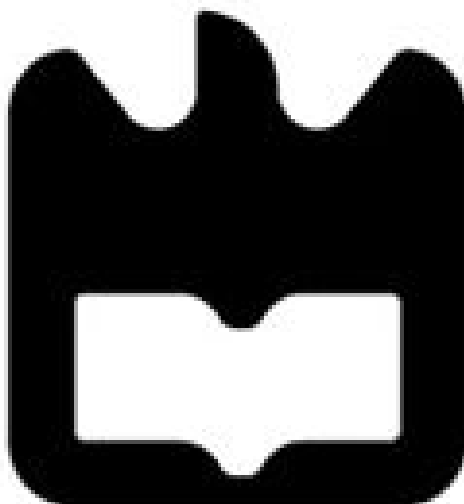




**Nuno Miguel Falcão
de Moura**

**Controlador e Device Drivers para um
Recuperador de Calor a Pellets**

(Controller and Device Drivers for a Pellet Stove)



**Nuno Miguel Falcão
de Moura**

**Controlador e Device Drivers para um
Recuperador de Calor a Pellets**

(Controller and Device Drivers for a Pellet Stove)

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Paulo Pedreiras, Professor Auxiliar do Departamento de Eletrónica e Telecomunicações da Universidade de Aveiro e do Doutor Joaquim Ferreira, Professor adjunto da Escola Superior de Tecnologia e Gestão de Águeda.

o júri / the jury

presidente

Prof. Doutor José Luís Costa Pinto de Azevedo

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof. Doutor Mário Jorge Rodrigues de Sousa

Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Paulo Bacelar Reis Pedreiras

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Agradecimentos

Agradeço à minha família, especialmente aos meus pais, por toda a confiança demonstrada e apoio que me deram nesta etapa da minha vida, bem como de todas as outras. Sem eles isto não seria possível.

Quero também agradecer aos amigos que sempre me apoiaram e incentivaram ao longo de todos estes anos, bem como à Telma, por toda a ajuda e incentivo.

Agradeço também ao Professor Valter Silva e ao Engenheiro Augusto Tavares, da Ecoama, que me ajudaram a ultrapassar algumas das dificuldades que surgiram.

Aos meus orientadores, Professor Paulo Pedreiras e Professor Joaquim Ferreira, que me apoiaram quando muitas vezes não sabia o que fazer, e asseguraram que o projecto corria sempre da melhor forma, um grande “Obrigado”.

palavras-chave

Sistemas Embutidos, Controlador, Caldeira, Recuperador de Calor, Device Drivers.

resumo

Hoje em dia é dada uma grande importância à sustentabilidade energética, procurando-se cada vez mais optar por fontes de energia renováveis.

Os *pellets* são uma das fontes desta energia, sendo um combustível utilizado maioritariamente em sistemas de aquecimento, como os fabricados pela Ecoama.

Existem vários controladores genéricos para este tipo de sistemas, apresentando no entanto várias contrariedades, como o custo elevado e a impossibilidade de adição de funcionalidades.

Nesta dissertação desenvolveu-se um protótipo de um controlador para um recuperador de calor. Este tem uma elevada possibilidade de customização, de forma a adaptar-se a outros aparelhos da Ecoama.

Ao projetar um controlador específico foi possível assegurar que este se adequa aos equipamentos, cumprindo as especificações e tendo um baixo custo.

O sistema desenvolvido foi testado no recuperador, tendo-se verificado a sua funcionalidade, bem como a da aplicação criada para computador, que permite a configuração do controlador. Construiu-se uma base sólida para a continuação do projeto, tendo como objetivo final a criação de um produto comercial.

keywords

Sistemas Embutidos, Controlador, Caldeira, Recuperador de Calor, Device Drivers.

abstract

Nowadays, society is very focused on energetic sustainability, with renewable energy sources becoming more and more prominent.

Pellets are one of these sources, being a fuel mainly used in heating systems, such as those manufactured by Ecoama.

There is a wide range of generic controllers for this kind of systems. However, they present a considerable amount of drawbacks, such as their high price and the impossibility of adding new functionalities.

In this dissertation a prototype for a pellet stove controller was developed. This controller has high customization capability, in order to be able to adapt to other devices of Ecoama.

By designing a specific controller it was possible to ensure that it is adequate to the systems, fulfilling all specifications and having a low cost.

The developed system was tested in the stove, where its functionality was verified. The computer program created in order to permit the configuration of the controller was also tested. A solid base was created for the project follow up, with the final goal being the creation of a commercial product.

Conteúdo

Conteúdo	XII
Lista de Figuras	XIV
Lista de Tabelas	XVI
Lista de Acrónimos	XVII
I. Introdução	1
1.1 – Enquadramento	2
1.2 – Motivação	3
1.3 – Objectivos	4
1.4 – Organização.....	5
II. Conceitos e Estado da Arte	7
2.1 – Pellets	8
2.1.1 – Em Portugal.....	10
2.2 – Aparelhos a Pellets	11
2.2.1 – Recuperadores de Calor	11
2.2.2 – Caldeiras	13
2.2.3 – Queimadores.....	16
2.3 – Controladores Genéricos	17
2.3.1 – Fumis ALPHA.....	19
2.4 – Hefesto e Funcionalidades Desejadas.....	22
III. Hardware	25
3.1 – Descrição dos Actuadores e Sensores	26
3.2 – Circuitos Projetados	28
3.2.1 – Módulos de Potência	31
3.2.2 – Alimentação e Detector de Passagem por Zero	33
3.2.3 – Sensores de Temperatura.....	34
3.3 – Placas Utilizadas.....	37
3.3.1 – PIC32-MAXI-WEB.....	37
3.3.2 – PCI	39
IV. Firmware e Software	41
4.1 – Firmware	42

4.1.1 – Device Drivers	42
4.1.1.1 – Sensores	42
4.1.1.2 – Actuadores.....	45
4.1.2 – Processo de Controlo.....	50
4.1.3 – Interface com o Touchscreen.....	54
4.2 – Software.....	60
4.2.1 – Aplicação para o Fabricante.....	60
4.2.2 – Rotina para Comunicação do Microcontrolador	64
V. Testes de Validação.....	69
5.1 – Validação de Circuitos e Device Drivers	70
5.1.1 – Sensores	70
5.1.2 – Actuadores	75
5.2 – Teste Ao Controlador.....	78
5.2.1 – Arranque Manual do Hefesto	78
5.2.2 – Controlo Automático do Sistema.....	78
VI. Conclusões.....	81
6.1 – Trabalho Futuro	82
Bibliografia.....	85

Lista de Figuras

Figura 1.1 – Pellets. _____	3
Figura 2.1 – Processo de fabrico de pellets. _____	9
Figura 2.2 – Recuperador de calor independente. _____	11
Figura 2.3 – Recuperador de calor embutido. _____	11
Figura 2.4 – Caldeira. _____	14
Figura 2.5 – Queimador. _____	17
Figura 2.6 – Controlador Fumis ALPHA e interface para o utilizador. _____	19
Figura 2.7 – Diagrama de Blocos do aparelho Hefesto. _____	24
Figura 3.1 – Termopar do tipo K. _____	28
Figura 3.2 – Circuito para as interfaces de potência. _____	31
Figura 3.3 – Formas de onda num Triac. _____	31
Figura 3.4 – Circuito de deteção de passagem por zero. _____	34
Figura 3.5 – Circuito para acondicionamento do sensor LM335. _____	35
Figura 3.6 – Circuito de acondicionamento para a sonda NTC. _____	36
Figura 3.7 – PIC32-MAXI-WEB. _____	38
Figura 3.8 – PCI. _____	39
Figura 4.1 – Esquema simplificado da função readADC(int channel). _____	43
Figura 4.2 – Potência entregue à VA relativamente ao instante de trigger. _____	49
Figura 4.3 - Potência entregue à VE relativamente ao instante de trigger. _____	49
Figura 4.4 – Diagrama de estados do recuperador de calor Hefesto. _____	50
Figura 4.5 – Fluxograma da função main. _____	52
Figura 4.6 – Ecrã de seleção de temperatura desejada. _____	56
Figura 4.7 – Ecrã de seleção da potência da VA. _____	56
Figura 4.8 – Interface para estabelecimento de conexão com o controlador. _____	61
Figura 4.9 – Ligação com o controlador estabelecida. _____	61
Figura 4.10 – Ligação estabelecida com o controlador após tentativa falhada. _____	62
Figura 4.11 – Tab de parâmetros do MA. _____	63
Figura 4.12 - Tab de parâmetros da VE. _____	63
Figura 4.13 - Tab de parâmetros da VA. _____	64
Figura 4.14 – Fluxograma da função UARTCheck(). _____	66
Figura 4.15 – Troca de mensagens entre o computador e o controlador. _____	67
Figura 5.1 – Temperatura medida pela sonda NTC e Termopar Fluke. _____	71
Figura 5.2 – Diferença de temperaturas medidas pela NTC e Termopar Fluke. _____	71
Figura 5.3 - Temperatura medida pelo sensor LM335 e Termopar Fluke. _____	72
Figura 5.4 - Diferença de temperaturas medidas pelo LM335 e Termopar Fluke. _____	72

Figura 5.5 - Temperatura medida pelo Termopar e Termómetro Fluke. _____	74
Figura 5.6 - Diferença de temperaturas entre Termopar e Termómetro Fluke. _	74
Figura 5.7 – Gráfico de Voltagem/Velocidade do Teste 1. _____	77
Figura 5.8 – Gráfico de Voltagem/Velocidade do Teste 2. _____	77
Figura 5.9 – Recuperador de calor em funcionamento. _____	80

Lista de Tabelas

Tabela 3.1 – Atuadores do aparelho Hefesto. _____	26
Tabela 3.2 – Funcionalidades desejadas para o controlador. _____	49
Tabela 5.1 – Temperaturas medidas. _____	70
Tabela 5.2 – Temperaturas medidas. _____	73
Tabela 5.3 – Valores de tensão medidos durante os testes. _____	76

Lista de Acrónimos

PCI Placa de circuito impresso

VA Ventoinha ambiente

VE Ventoinha de extracção

MA Motor de Alimentação

I/O *Inputs/Outputs*

PWM *Pulse Width Modulation*

RPM Rotações por minuto

ADC *Analog to Digital Converter*

RSI Rotina de Serviço à Interrupção

UART *Universal Asynchronous Receiver/Transmitter*

GOL *Graphics Object Layer*

MAL *Microchip Application Library*

I. Introdução



1.1 – Enquadramento

Um controlador é um dispositivo que recebe um sinal de entrada de uma variável medida, compara este valor com um valor pré-definido, e procede à determinação do valor requerido de um sinal de saída de modo a aplicar uma ação corretiva ao sistema. Nos dias de hoje, a grande maioria dos controladores electrónicos usam processadores digitais e sinais eléctricos de modo a realizarem as suas funções [1].

Os sistemas e variáveis a serem controlados podem ser variadíssimos, por exemplo a temperatura de uma sala, o percurso de um avião, a pressão numa caldeira, entre outros.

A empresa Ecoama é responsável pelo fabrico e venda de equipamentos de aquecimento, nomeadamente recuperadores de calor, queimadores e caldeiras. Os primeiros são máquinas que visam o aquecimento da temperatura ambiente, ao passo que as caldeiras pretendem aquecer água, podendo esta ser então utilizada para aquecimento central, águas sanitárias, ou ambos. Por último um queimador é utilizado para o aquecimento prévio de algo, por exemplo, um forno.

Qualquer um dos três aparelhos referidos funciona tendo por combustível *pellets* (Figura 1). Os *pellets* são um combustível sólido produzido a partir de biomassa, maioritariamente de resíduos de madeira, como serradura. Têm como principal vantagem, comparativamente à biomassa não processada, a sua elevada densidade e conteúdo energético, levando a um custo reduzido para armazenamento e transporte [2]. Tendo as suas propriedades reguladas por regulação específica, como será discutido posteriormente, a diferença entre *pellets* de diferentes fabricantes é reduzida, sendo também esta uma vantagem importante.



Figura 1.1 – Pellets [4].

Em Agosto de 2008, os *pellets* para a produção de uma determinada quantidade de energia eram 60% mais baratos que o gásóleo necessário para o mesmo [2]. No entanto, o investimento inicial para um sistema de aquecimento a pellets é consideravelmente superior ao de um a gásóleo[4]. Ainda assim, esta é uma tecnologia em franca expansão, tendo por enquanto uma quota de mercado reduzida, mas sendo estimado que, em 2020, 20% da energia utilizada na União Europeia proveniente de biomassa venha de *pellets* [2].

Para a correta operação de qualquer aparelho que utilize *pellets*, o controlador deste é de extrema importância, sendo responsável pelo comportamento de todo o sistema, assegurando um bom funcionamento e elevada eficiência energética.

1.2 – Motivação

Existem vários controladores genéricos (isto é, podem ser adaptados a diferentes aparelhos de diferentes fabricantes) para os aparelhos descritos. No entanto, todos estes apresentam vários problemas ou contrariedades. Por serem genéricos, acabam muitas vezes por terem bastantes funcionalidades disponíveis que não são utilizadas pelo fabricante, por estas não se adequarem ao equipamento produzido. Todas estas funcionalidades adicionais representam um custo acrescido



desnecessário, levando o controlador a atingir um preço bastante alto. Por outro lado, não têm a capacidade de implementar funcionalidades específicas a um dado produto, limitando assim os serviços que podem ser oferecidos, ou até os próprios aparelhos.

Ao projetar um controlador específico, é possível assegurar que este se vai adequar perfeitamente aos equipamentos, estando optimizado para tal. Deste modo, obtém-se um controlador que apresenta as funções requeridas e que não tem um custo desnecessariamente elevado.

É também desejável que o controlador seja uma *whitebox*, isto é, que os circuitos internos sejam conhecidos, de modo a permitir um *debug* mais fácil em caso de avaria bem como possibilitar efetuar alguma modificação aos mesmos, algo que não é possível nos controladores genéricos.

Os motivos acima referidos levam a que seja de grande interesse para uma empresa ter um controlador construído especificamente para os seus sistemas.

1.3 – Objectivos

O principal objectivo desta dissertação é o desenvolvimento de um sistema de controlo para um recuperador de calor, com elevada possibilidade de customização, de forma a adaptar-se a sistemas diferentes, e ainda a controlar cada um destes da melhor forma. Assim, é necessário que seja possível proceder ao ajustamento fino de vários parâmetros. Para tal, foram definidos vários sub-objectivos que visam o desenvolvimento metódico do projeto, nomeadamente:

- i. Investigação sobre os vários aparelhos vendidos pela Ecoama, de forma a determinar as interfaces necessárias ao controlador e definir os elementos necessários para agir sobre cada componente do aparelho.

- ii. Estudo dos controladores disponíveis no mercado de modo a detectar lacunas que estes possam ter, assim como funções presentes nos mesmos que sejam desejadas para o controlador que será desenvolvido.
- iii. Projeto dos circuitos de acondicionamento para cada interface presente no controlador e teste dos mesmos.
- iv. Desenho e construção da placa de circuito impresso (PCI).
- v. Criação de uma interface entre o utilizador e o sistema para a realização de testes sobre cada elemento do sistema e que permita efetuar a operação manual do recuperador de calor.
- vi. Desenvolvimento de *firmware*, possibilitando a um indivíduo operar o controlador de calor, mantendo-se abstraído dos estados por que este passa.
- vii. Realização de uma aplicação para computador que permita a configuração do controlador pelo responsável na empresa.

1.4 – Organização

Nesta secção sumariza-se a estrutura da presente dissertação que se encontra organizada da seguinte forma:

- O capítulo dois é encabeçado por uma breve apresentação do combustível utilizado nos aparelhos, o *pellet*. Seguidamente, são introduzidos os três tipos de aparelho para os quais o controlador foi projetado, detalhando primeiro os principais componentes que constituem cada um destes, assim como a sua função e apresentando o papel do controlador no correto funcionamento dos mesmos. O capítulo é terminado com uma reflexão sobre o sistema específico que se pretende controlar.



- O terceiro capítulo incide no *hardware* utilizado no sistema, sendo iniciado por uma descrição de cada sensor e actuador presentes, seguida pelos circuitos de acondicionamento utilizados para cada um destes. É então apresentada a PCI projetada, assim como a placa de desenvolvimento utilizada.
- O quarto capítulo consiste numa descrição do *firmware* e *software* construídos. Primeiramente, são descritas as rotinas criadas para o microcontrolador e a forma como estas interagem entre si e com os elementos do sistema. É então apresentada a aplicação para computador com o propósito de efetuar a configuração do controlador.
- No capítulo cinco encontra-se a exposição dos testes realizados com vista à validação do sistema, começando pelos realizados aos sensores e actuadores, individualmente. De seguida, ao controlo do sistema de forma manual e, por último, automaticamente.
- Finalmente, no sexto capítulo, são apresentadas as conclusões retiradas do trabalho efectuado e deixadas considerações sobre possível trabalho futuro.

II. Conceitos e Estado da Arte



2.1 – Pellets

Nos dias de hoje é dada uma grande importância ao tema da sustentabilidade energética, tentando-se cada vez mais optar por fontes de energia renováveis, ao invés dos combustíveis fósseis. Estima-se que a produção destas tenha duplicado entre 1973 e 2008 [4], e a União Europeia comprometeu-se a ter 20% da sua energia proveniente de fontes renováveis até 2020 [5]. Esta energia pode provir de várias fontes, como do vento (eólica), sol (solar), biomassa, entre outras.

Os *pellets* são uma das possíveis fontes para esta energia limpa, bastante utilizados na Europa Central e Escandinávia [6], sendo um combustível que pode ser utilizado em sistemas de aquecimento com uma eficiência bastante elevada e com emissões de carbono neutras [7,8]. Têm uma forma cilíndrica, com um máximo de 38 milímetros de comprimento e entre 6 e 8 mm de diâmetro [3]. O seu tamanho diminuto faz com que seja bastante simples utilizá-los em sistemas de alimentação automática, tornando também o seu armazenamento e transporte mais fáceis. Podem ser constituídos por diferentes elementos, sendo os mais comuns feitos a partir de serradura, devido à sua fácil aquisição. Existem também vários feitos a partir de desperdícios agrícolas e florestais, ou obtidos a partir de uma mistura [3,8,9].

A grande eficiência energética dos *pellets*, mesmo em relação a outros combustíveis de proveniência semelhante, como por exemplo a lenha, vem da sua elevada densidade, entre 4 a 10 vezes superior à biomassa utilizada para os produzir [10] e baixo teor de humidade, obtidos através da maneira como estes são fabricados. Este processo (Figura 2), desenvolve-se em 5 fases:

- i. Secagem – Nesta fase, a matéria-prima é aquecida para secar, de modo a que o *pellet* tenha uma quantidade final de água reduzida, preferencialmente inferior a 8%. Esta fase do processo é de extrema importância, uma vez que a humidade do *pellet* vai desempenhar um grande papel na sua energia calorífica.

- ii. Moagem – Já secos, os resíduos são triturados, até que o seu tamanho esteja reduzido a cerca de 1 mm^3 . Isto vai levar a que o *pellet* seja bastante denso. Esta fase e a de secagem são permutáveis.
- iii. Criação do *Pellet* – Após a moagem, é utilizada uma prensa para comprimir o material contra uma matriz aquecida, de modo a que este a atravesse, sendo cortado por lâminas à medida que tal acontece, criando os *pellets* na sua forma final.
- iv. Arrefecimento – Ao saírem da matriz, os *pellets* encontram-se a uma temperatura bastante elevada (cerca de 100°C , o que reduz uma vez mais a sua humidade). Assim, é necessário arrefecê-los, para que não percam a sua forma. Por norma são utilizados jactos de ar para este efeito.
- v. Verificação – A fase de verificação tem o propósito de separar resíduos que tenham passado pela matriz sem formarem um *pellet*, reintroduzindo-os no processo [3,6,9].

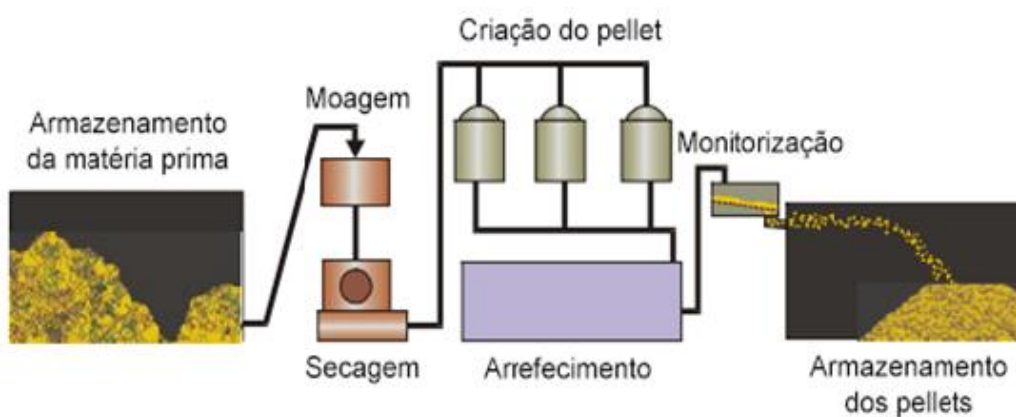


Figura 2.1 – Processo de fabrico de pellets (adaptado de [9]).

É importante referir que os *pellets* têm que cumprir um vasto conjunto de regras estritas, definidas pela norma europeia EN 14961-2. Esta norma separa-os em grupos consoante as suas dimensões, estabelece um limite para a humidade que



estes podem conter (sendo 15% o valor máximo) e um limite para as cinzas que os mesmos libertam após a combustão (variando entre 0,5 e 10%). Efetua ainda uma divisão em categorias dependendo da densidade do *pellet*, entre várias outras propriedades, como a sua durabilidade e a presença ou não de aditivos [11]. Para além desta norma, existem ainda outras certificações que podem ser atribuídas, dependendo da qualidade dos *pellets*, como o ENPlus [12].

2.1.1 – Em Portugal

Há em Portugal uma grande capacidade para a produção de energia a partir de biomassa, existindo cerca de 1,8 milhões de toneladas de resíduos adequados para combustão a serem criados por ano [13]. Em 2008 eram produzidas cerca de 400 mil toneladas de *pellets* por ano [10], com este valor a aumentar para cerca de 500 mil em 2010 [14]. No entanto, o consumo interno de *pellets* é bastante reduzido, sendo quase insignificante. Apenas cerca de 5 a 10 mil toneladas foram consumidas internamente, representando 1-2% da produção, sendo o resto exportado [10,14].

Um estudo indica que ao utilizar uma caldeira a *pellets* com o poder nominal de 50 kW num hotel, substituindo a caldeira existente no mesmo que opera a gásóleo, a poupança na compra de combustível daria retorno ao investimento ao fim de 3 anos e 5 meses, poupando-se 2545 euros ao fim de 5 anos [15]. Os sistemas de aquecimento a pellets têm no entanto uma contrariedade relativamente a sistemas que utilizem combustíveis diferentes, que se prende com o seu elevado custo inicial.

2.2 – Aparelhos a Pellets

2.2.1 – Recuperadores de Calor

Estes aparelhos têm como objectivo o aquecimento da temperatura ambiente de uma ou mais divisões de um imóvel. Podem ser elementos independentes, como o da Figura 2.2, ou encontrarem-se embutidos numa lareira, ou parede, como o presente na Figura 2.3.



Figura 2.2 – Recuperador de calor independente [16].



Figura 2.3 – Recuperador de calor embutido [16].

O funcionamento destes aparelhos é idêntico. Através da queima de combustível (gasóleo, lenha, *pellets*), ou através do aquecimento de resistências eléctricas, é produzido calor que é então libertado para o meio ambiente através de uma ventoinha ambiente (VA). Para além desta, é necessária ainda uma outra responsável por efetuar a extração dos gases resultantes da combustão, a ventoinha de extração (VE). Nos aparelhos independentes, a VE é imprescindível, sendo usada nos aparelhos embutidos sempre que os mesmos não se encontrem



embutidos numa chaminé, podendo ainda assim ser usada neste caso, de modo a permitir uma melhor circulação do ar.

Podem ainda existir outros componentes nestes aparelhos, dependendo do fabricante e do tipo dos mesmos. Nos recuperadores a *pellets* é comum encontrar-se um motor que permite a alimentação automática, não sendo necessário ao utilizador colocar repetidamente *pellets* na câmara de combustão. Para além do motor de alimentação (MA), é também comum existir um dispositivo que permite ao aparelho iniciar a combustão automaticamente. Este é por norma uma resistência de aquecimento, que incinera os *pellets*, dando início à queima, sendo posteriormente desligada.

Estes são os quatro actuadores principais num recuperador de calor a *pellets*, estando presentes virtualmente em todos os aparelhos vendidos nos dias de hoje [16–19]. No entanto, é frequente encontrarem-se outras peças, quer para introduzir novas funcionalidades, quer para oferecer um melhor serviço, reduzindo a manutenção necessária. É possível por exemplo adicionar mais uma VA ao sistema, permitindo assim o aquecimento de uma outra divisão. Este caso torna necessária a presença de tubagens, podendo estas estar colocadas dentro ou fora de uma parede, de modo a levar o ar quente para um local diferente daquele em que se encontra o recuperador. Para dar mais conforto ao utilizador, é possível ainda instalar um sistema automático para a limpeza do recuperador, nomeadamente para impedir que as cinzas se acumulem.

Para o melhor funcionamento do aparelho, são necessários também alguns sensores. O sensor mais importante é o que permite a leitura da temperatura dos gases de escape. A sua extrema importância prende-se com o facto de um valor muito alto assinalar o mau funcionamento do sistema, podendo indicar um aumento de pressão na câmara de combustão o que, em casos extremos, pode levar a uma explosão.

Além do acima referido, é do interesse do utilizador que exista um sensor para medir a temperatura ambiente, de modo a que o recuperador ajuste

automaticamente o seu funcionamento para manter a temperatura desejada. Por vezes, existe ainda um sensor de luminosidade, responsável por detectar a combustão dos *pellets*, dando assim mais uma garantia de segurança (de modo a que o aparelho detecte a falha se por algum motivo deixar de haver queima, desligando o MA), e também levando a uma transição entre os estados de funcionamento do sistema [20]. Alguns sistemas têm também um sensor que mede o nível de *pellets* no reservatório, notificando o utilizador quando este deve efetuar o reabastecimento [21]. Finalmente, é possível instalar um sensor que analise o ar que circula no escape, e comunique com o controlador de modo a ajustar a potência fornecida para manter a pressão na câmara de combustão constante [22].

Existem alguns sensores de segurança que desligam o MA em caso de mau funcionamento. Os mais comuns são um interruptor térmico, colocado no reservatório dos *pellets*, de modo a prevenir que a temperatura destes suba até um ponto em que os mesmos entrem em combustão, e um medidor da pressão na câmara de combustão, que previne que esta suba a um nível que possa levar à explosão da mesma. Pode também ser ligado um termóstato para agir sobre o controlador do aparelho.

Para além dos sensores e atuadores descritos acima, existem ainda outros componentes do aparelho. É indispensável um depósito para a combustão dos *pellets*, e por norma encontra-se também um reservatório para os mesmos, embora por vezes este possa estar separado do aparelho. Existe ainda um depósito para onde as cinzas caem, colocado por baixo do depósito de combustão, de modo a que seja fácil proceder à limpeza do sistema (manual ou automática).

2.2.2 – Caldeiras

A base do funcionamento de uma caldeira é bastante semelhante ao de um recuperador de calor. É aquecida água (ou outro líquido, no caso de um propósito diferente), através da queima de combustível ou do aquecimento de resistências eléctricas, que é então utilizada para diferentes fins, consoante o aparelho.



Dependendo do intento do sistema, a água pode então ser armazenada num reservatório ou tanque (podendo este ser interno ou externo (Figura 2.4) ao aparelho) [23]. Ambos os sistemas têm as suas desvantagens.



Figura 2.4 – Caldeira.

No caso do sistema com reservatório, tem a seu favor o facto de existir água quente disponível de imediato (assumindo que esta não foi gasta previamente), não sendo necessário esperar que o aparelho entre em funcionamento para aquecer a água (o que pode levar algum tempo, especialmente no caso do uso de *pellets*), e o seu consumo energético momentâneo nunca ser tão alto como o sem tanque, uma vez que a água pode ser aquecida gradualmente [24]. É também mais facilmente adaptável a outros sistemas, como o de pré-aquecimento da água usando energia solar [25,26].

O sistema sem reservatório tem como vantagem a redução nos gastos energéticos, uma vez que ao não armazenar a água, esta não vai arrefecer, sendo assim necessário voltar a aquecê-la. A ausência de tanque leva também a um sistema mais compacto, sendo só necessária a unidade para aquecimento da água. Tem, no entanto, a desvantagem que reside no facto de, uma vez que a água é aquecida

on-demand, pode ser desejada mais água quente que a que o aparelho consegue produzir. Neste caso, o utilizador receberá água menos quente que o desejado [24].

Pelos pontos descritos acima, conclui-se que o sistema sem reservatório, numa caldeira a *pellets*, não é adequado para o aquecimento de águas sanitárias, graças ao seu lento tempo de arranque, embora possa ser utilizado para aquecimento central. No entanto, estes aparelhos são raros. O sistema com reservatório pode ser utilizado para ambos os fins. É ainda possível criar um sistema híbrido, que tenha um elevado poder de aquecimento instantâneo, mas que utilize um reservatório para as águas sanitárias.

Quanto aos componentes do aparelho, são bastante semelhantes aos presentes num recuperador de calor. A resistência de ignição, MA e VE, assim como o sensor para leitura de gases de escape, são indispensáveis. Para além destes, é necessário o uso de uma bomba, responsável por levar a água da caldeira até ao permutador de calor, uma serpentina que vai aquecer o reservatório, ou então para o circuito de aquecimento central. É também necessário utilizar um sensor de temperatura para a medição da temperatura da água, podendo esta ser efectuada no reservatório ou à saída do aparelho. Uma vez que as interfaces destes dois componentes podem ser iguais às dos que deixam de ser necessários (uma interface de potência e um sensor de temperatura), é possível criar um controlador facilmente adaptável a estes dois aparelhos.

Existem ainda outros componentes que podem ser utilizados para melhorar o sistema. O sistema de limpeza automático continua a ser relevante, e vários outros têm interesse para aplicações específicas, como a existência de uma válvula de 3 vias ou uma segunda bomba de água, para dar dois destinos diferentes à água que é aquecida. Esta segunda bomba pode também ser usada para controlar a água que volta à caldeira após ter circulado, de modo a evitar que ocorra condensação. Podem também ser utilizados sensores de temperatura extra, possibilitando a medição da temperatura da água no reservatório, à saída da caldeira e à entrada da mesma. De forma semelhante aos recuperadores de calor, as caldeiras por



norma estão preparadas para funcionar com um termóstato, se o utilizador assim o desejar.

Por último, é importante referir que existem sistemas que incorporam um recuperador de calor e uma caldeira, aquecendo tanto água como ar [20,27].

2.2.3 – Queimadores

Um queimador (Figura 2.6) distingue-se dos dois aparelhos discutidos acima pelo facto de poder ser utilizado para aplicações diferentes. O seu propósito é o de criar uma chama forte (os queimadores têm por norma potências bastante mais elevadas que os recuperadores de calor e caldeiras), que pode depois ser aproveitada para diversos fins. O mais comum é utilizar o queimador para aquecer um grande depósito de água, acabando assim por ser um componente de uma caldeira, mas este também pode ser utilizado para pré-aquecer um forno, por exemplo. No entanto, os queimadores raramente são utilizados domesticamente, sendo mais apropriados para hotéis, ginásios, edifícios públicos [28] e outras superfícies com uma necessidade de água quente considerável, devido à sua elevada potência (por vezes superior a 1 MW) [29].

O queimador é o aparelho, de entre os 3 apresentados, com o menor número de componentes. Uma vez que a única temperatura que é necessário monitorizar é a dos gases de escape, um queimador simples pode apenas possuir um sensor de temperatura para este efeito. Juntamente com este, encontra-se um sensor de chama, obrigatório nestes sistemas, perfazendo assim o conjunto de sensores. Ao nível dos actuadores, existe um MA, uma resistência de ignição e uma VE, também responsável por efectuar a limpeza automática do sistema [20,30].



Figura 2.5 – Queimador.

Os sistemas mais complexos podem vir já preparados para funcionar como caldeira, sendo possível que estes tenham um sensor que controle a temperatura da água do reservatório a que o queimador é acoplado, e até uma bomba de água para a fazer circular quando esta atingir a temperatura desejada. Também os queimadores podem ser associados a um termóstato que será responsável por auxiliar o seu controlo.

2.3 – Controladores Genéricos

Todos estes aparelhos necessitam de um controlador, responsável por assegurar o bom funcionamento dos mesmos, assim como por maximizar o seu rendimento. Existem alguns fabricantes de aparelhos a *pellets* que produzem os seus controladores, ou que os têm produzidos especificamente para os seus sistemas, o que lhes permite introduzir inovações nos controladores, destacando-se da



concorrência, e tendo o controlador com as características exatas desejadas [31,32]. No entanto, muitas empresas não têm esta possibilidade, utilizando assim um dos vários controladores genéricos presentes no mercado [20,33–35]. Existem ainda empresas que fabricam controladores para os seus sistemas e que os vendem a outras, com a possibilidade de adaptação aos aparelhos destas [36].

Embora a utilização de um controlador genérico evite que a empresa necessite de efetuar um elevado investimento inicial para o desenvolvimento do controlador, existem várias contrariedades inerentes, como a dependência de terceiros, a falta de customização de modo a incorporar interesses específicos que possam surgir, um custo elevado devido a funcionalidades que podem não ser aproveitadas, entre outras.

De um modo geral, estes controladores são bastante semelhantes entre si. Baseiam-se num microcontrolador, assente numa PCI preparada para a ligação a vários sensores e actuadores, contendo também um transformador e reguladores para a alimentação da placa, feita sempre através da rede eléctrica. É também importante que tenham forma de comunicar para um computador, de modo a que o fabricante dos aparelhos possa configurar o controlador de acordo com as suas especificações. O programa para a configuração é, por norma, fornecido num CD aquando da compra do controlador. Este permite efetuar uma configuração minuciosa, estabelecendo parâmetros desde a velocidade das ventoinhas para cada nível de potência até valores de temperaturas para que sejam reconhecidas as condições de paragem do aparelho [20,34].

Dentro dos controladores analisados, estes apresentam entre 4 a 7 interfaces de potência. Como foi visto anteriormente, 4 actuadores são o mínimo exigido para um funcionamento correto dos aparelhos sem que seja exigido ao utilizador interagir com o processo de queima em si. Estas interfaces são saídas com TRIAC, com exceção da interface para a resistência de ignição, onde alguns controladores apresentam um relé, devido à maior potência deste componente [20,35].

Os sensores para que os controladores se encontram preparados também não diferem muito. Existem interfaces para 3 a 5 sensores de temperatura, um sensor de luminosidade e interfaces para sensores que permitam obter *feedback* sobre o funcionamento dos actuadores, por exemplo sensores de Hall.

Para além das interfaces para sensores e actuadores encontra-se ainda sempre presente uma pilha para a alimentação do relógio, um transformador e reguladores apropriados, e alguma forma de comunicação com um computador. Finalmente, existe ainda uma ligação para a interface de utilizador, que é por norma constituída por um *display* LCD com botões [20,33–36].

É ainda importante referir que alguns controladores optam por uma placa de menores dimensões, com um número reduzido de funcionalidades em que estão presentes apenas as mais importantes, havendo a opção de serem adquiridas placas de expansão que integrem interfaces extra [33].

2.3.1 – Fumis ALPHA

O controlador utilizado pela Ecoama nos seus aparelhos é o Fumis ALPHA (Figura 2.7), produzido pela Fumis, pertencente à ATech Electronics.



Figura 2.6 – Controlador Fumis ALPHA e interface para o utilizador [37].



Este controlador está preparado para vários tipos de aparelho a *pellets*, e pode ainda operar sistemas que usem lenha, ou lenha e pellets. O controlador Fumis ALPHA vem preparado com 12 pré-configurações, 3 para recuperadores de calor, 4 para caldeiras, 3 para queimadores e 3 configurações direcionadas para caldeiras com a utilização de lenha e/ou *pellets*. Para além destas pré-configurações, o fabricante dos aparelhos pode ainda definir as suas configurações, desde que respeite o modelo para as mesmas e as limitações do controlador.

É um controlador com um elevado número de características, apresentando:

- Parâmetros completamente configuráveis;
- Bateria interna para o relógio e microprocessador;
- Possibilidade de configuração de programas para executar em dias diferentes (máximo de 6 programas com 3 diferentes a serem executados num dia);
- Regulação do fluxo de ar com um sistema de *feedback* do mesmo;
- Controlo da velocidade da rotação das ventoinhas através de um sensor para a leitura das mesmas;
- Regulação constante de baixa pressão na câmara de combustão quando são utilizadas 2 ventoinhas;
- Regulação do poder de queima quando utilizada uma sonda de temperatura ou um termóstato;
- Registo de erros com capacidade de registar até 15 alarmes;
- Registo de eventos e contador de operações [20].

Ao nível de *inputs* e *outputs* (I/O), o controlador Fumis ALPHA também está equipado com bastantes interfaces. Relativamente a *inputs* tolera:

- 3 sensores de temperatura *Negative Temperature Coefficient* (NTC);
- 1 sensor de temperatura termopar do tipo-K capaz de efetuar leituras até 400°C;
- 1 input analógico entre 0-5V;

- 2 inputs digitais de contacto aberto;
- 2 inputs de alta voltagem para segurança.

No que se refere a outputs encontra-se:

- 1 relé;
- 5 TRIACS;
- 1 output de 0-10V ou Pulse Width Modulation (PWM) de 10 KHz [20].

Independentemente do aparelho utilizado com o controlador, existem sempre 4 estados principais distintos inerentes ao funcionamento dos mesmos. No primeiro estado, “*Fire Up*”, o sistema entra em operação e dá-se a ignição do combustível na câmara de combustão. Na segunda fase, apelidada de “*Burning*”, é ajustado o poder de queima para se obter a temperatura desejada, e o sistema entra na sua fase de trabalho. Após o comando para desligar, ou no caso da temperatura ultrapassar o valor desejado, o controlador passa o aparelho para a fase de “*Fire Down*”, onde a chama é extinta, deixa de ser fornecido combustível à câmara de combustão, e esta é arrefecida pelas ventoinhas. Após isto, o sistema entra na sua fase final, “OFF”. Um diagrama com a máquina de estados imposta por este controlador é apresentado mais à frente, no capítulo IV, Figura 4.1.

O controlador está preparado para operar com vários sensores que permitem obter uma combustão nivelada e constante, como sensores de Hall para manter a velocidade das ventoinhas constante e o sensor FlowTronic [22], que mede o fluxo de ar de gases de escape, podendo assim a velocidade da ventoinha de escape ser ajustada de acordo com o desejado. No entanto, no caso da Ecoama, os dois componentes referidos acima são um exemplo de funcionalidades que não são utilizadas, representando apenas um custo adicional desnecessário.



2.4 – Hefesto e Funcionalidades Desejadas

O âmbito desta dissertação prende-se com a construção de um controlador capaz de efetuar o controlo de um dos 3 tipos de aparelho discutidos, sendo no entanto facilmente adaptável para os outros 2.

Devido ao facto de ser mais fácil obter *feedback* visual de um recuperador de calor do que de uma caldeira, assim como uma resposta mais rápida, e tendo em conta o menor número de interfaces, foi desenvolvido o controlador com base no primeiro, neste caso, o aparelho Hefesto (Figura 2.2) [17].

A potência do aparelho, quando em funcionamento, varia entre 2,4 e 5 KW, dependendo do nível de queima selecionado pelo utilizador. Tem uma autonomia de 20 horas a trabalhar na potência máxima, podendo queimar até 1 Kg de *pellets* por hora. O seu reservatório leva 20 Kg de *pellets*, o suficiente para manter o aparelho em funcionamento até 41 horas, no caso de se selecionar a potência mínima. É adequado para o aquecimento de divisões com até 120 m³ [16].

Como requisitos para este aparelho funcionar, são necessárias interfaces para os seguintes sensores e actuadores:

- Sensor de temperatura para temperatura ambiente (NTC);
- Sensor de temperatura para temperatura dos gases de escape (termopar do tipo K);
- Resistência de ignição;
- MA de *pellets* (sem-fim);
- VE;
- VA.

Na figura 2.8 encontra-se o diagrama de blocos deste sistema, com alguns dados das interações do controlador com os vários sensores e actuadores. Ambas as ventoinhas exigem um controlo que permita variar a sua velocidade, ao passo que

a resistência de ignição trabalha num regime ON/OFF. O MA trabalha com um *duty-cycle* variável, dependendo do poder de queima que é desejado pelo utilizador.

Uma vez que a temperatura ambiente se mantém num intervalo de valores reduzido, e visto ser pouco provável que o utilizador deseje que a divisão atinja temperatura superiores a 30°C, é possível utilizar um sensor para a medição da mesma com uma gama bastante reduzida. Quanto à temperatura dos gases de escape, esta deve situar-se entre os 60 e os 90°C durante a fase de trabalho do equipamento. No caso de ultrapassar os 120°C, um alarme tem de ser ativado para notificar o utilizador. O sistema cessa o seu funcionamento normal e resolve o problema, sendo necessária efetuar a manutenção do aparelho para avaliar a causa da avaria.

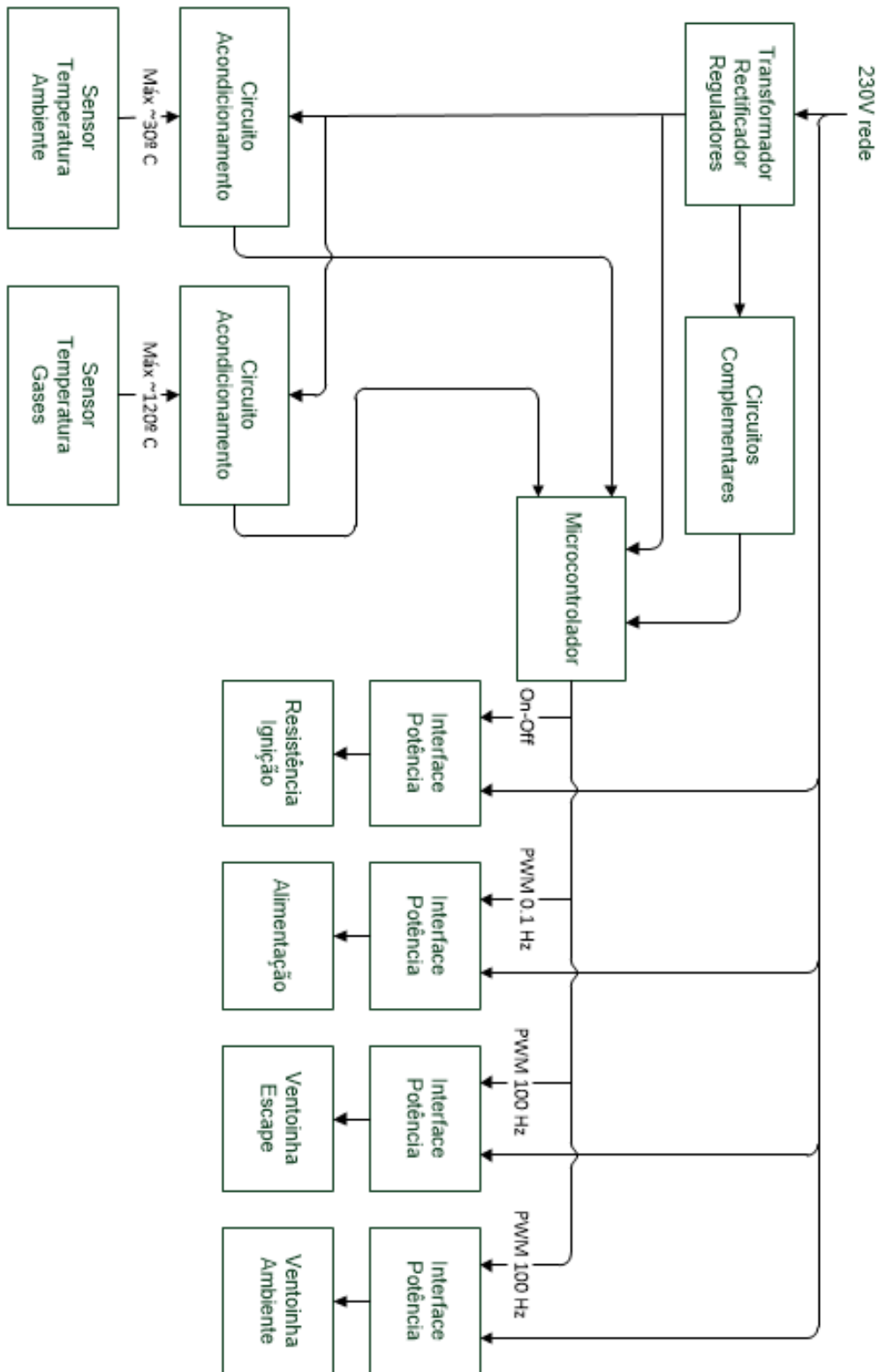


Figura 2.7 – Diagrama de Blocos do aparelho Hefesto.

III. Hardware



Neste capítulo é descrito o Hardware que foi utilizado na dissertação, começando por uma apresentação dos vários actuadores e sensores do sistema. Seguidamente, são enumerados os circuitos de acomodamento desenhados. Por fim, é apresentada a PCI projetada e a placa de desenvolvimento que foi utilizada.

3.1 – Descrição dos Actuadores e Sensores

O sistema é composto a nível de actuadores pelos 4 referidos anteriormente. O MA e as duas ventoinhas são motores com componente indutiva, ao passo que a resistência de ignição tem apenas componente resistiva. Na tabela 3.1 encontra-se a lista de actuadores do sistema, bem como a potência, intensidade da corrente, rotações por minuto (RPM) e fluxo de ar máximo por hora, no caso das ventoinhas [38–42].

Tabela 3.1 – Actuadores do aparelho Hefesto.

<i>Actuador</i>	<i>Referência</i>	<i>Potência (W)</i>	<i>Corrente (A)</i>	<i>Fluxo de Ar (m³/h)</i>	<i>RPM</i>
<i>Resistência de Ignição</i>	HTH Rohrheizer 79/11/200	175	0,79	-	-
<i>Motor de Alimentação</i>	BCZ 509 Stator 32	55	0,25	-	2
<i>Ventoinha Ambiente</i>	AACO VA2 C 50	55	0,3	190	-
<i>Ventoinha de Exaustão</i>	AACO ST25	54	0,44	128	2760

O motor de alimentação encontra-se acoplado a um trado o qual se encontra fechado num tubo ligado ao reservatório de *pellets*. Assim, à medida que o motor faz o trado rodar, os *pellets* vão descendo ao longo do tubo, até que chegam ao fim deste e caem na câmara de combustão. Na câmara de combustão encontra-se um

depósito, onde se dá a queima dos *pellets*. O depósito tem uma pequena abertura, onde está colocada a resistência de ignição. Esta é capaz de atingir temperaturas superiores a 500°C, dando início à combustão assim que um *pellet* lhe toque, ou, no caso de tal não acontecer, os *pellets* atinjam uma temperatura que os leve a entrarem em combustão espontânea. É no entanto necessário que a ventoinha de exaustão (também chamada ventoinha de extração ou de escape) garanta que existe oxigénio na câmara. A ventoinha de extração assegura uma contínua entrada de oxigénio na câmara de combustão, para além de expelir os fumos resultantes da queima. Por último, a ventoinha ambiente é responsável por fazer a transferência de calor para o meio ambiente, podendo também ser usada para arrefecer o recuperador, após este ter terminado o seu funcionamento.

O sistema tem 2 sensores, um termistor NTC e um termopar do tipo-K, ambos sensores de temperatura. A sonda NTC, para medição da temperatura ambiente, é um sensor resistivo, que varia o valor da sua resistência mediante uma variação na temperatura, existindo uma relação inversa entre temperatura e resistência (o aumento da temperatura leva a uma descida da resistência). Este sensor não tem uma gama de medição tão elevada como a de um termopar. No entanto, supera bastante os requisitos, permitindo medir temperaturas acima dos 120°C [44], que excedem a temperatura ambiente dos locais em que o aparelho se deverá encontrar. Este sensor é barato e apresenta uma resposta rápida, sendo bastante preciso, mesmo entre sensores, podendo serem substituídos sem ser necessário proceder à recalibração [43–46]. Para além disto, são os sensores mais resistentes, conseguindo suportar grandes choques, tanto mecânicos como térmicos [43]. As desvantagens deste sensor residem no facto de este não ser linear, no facto de ser necessário excitar o sensor utilizando uma fonte de corrente, e na sua reduzida gama de valores, sendo esta última irrelevante para esta aplicação. A NTC utilizada é fabricada pela Sacet-Probes, referência NS N E 1.5 5/6 E 20 N S E, que apresenta uma resistência de 10 K Ω à temperatura de 25°C [47].

Um termopar tem como princípio de funcionamento o efeito termoelétrico, isto é, o facto de uma junção de dois metais criar uma diferença de potencial proporcional à diferença de temperatura a que a junção se encontrar sujeita, como exemplificado



na Figura 3.2 [44]. Estes sensores permitem a medição numa elevada gama de valores, sendo o termopar utilizado, do tipo K, indicado para a leitura de temperaturas até 350°C [48]. As duas ligas metálicas que compõem estes termopares são o Alumel (95% Níquel e 5% Alumínio) e o Chromel (90% de Níquel e 10% de Crómio). Os termopares são sensores baratos e com uma boa linearidade, no caso dos tipo K. Têm também uma velocidade de resposta superior à dos termístores [43–46]. No entanto, degradam-se com mais facilidade, podendo aumentar o seu erro em alguns graus por ano [44], e são menos precisos. É ainda necessário efetuar uma compensação, uma vez que a temperatura lida por estes não é absoluta, mas apenas a diferença entre a do ponto de junção dos dois metais e a da junção de cada um dos metais ao circuito, neste caso à PCI. Assim, deve ser colocada uma das ligas do termopar a uma temperatura conhecida ou deverá ser utilizado um sensor auxiliar para medir a temperatura na ligação do termopar à PCI.

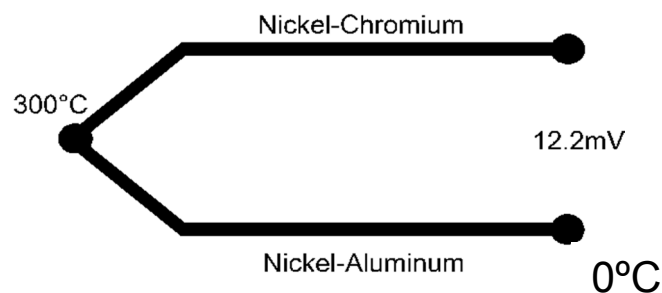


Figura 3.1 – Termopar do tipo K (adaptado de [49]).

3.2 – Circuitos Projetados

O microcontrolador selecionado foi um PIC32. Uma frequência de relógio suficiente (até um máximo de 80 MHz) e outras características importantes, como 6 UARTS, 5 *timers* de 16 bits, 5 PWMs, 5 pinos de interrupção externa, aliadas a diferentes opções de conectividade (CAN, USB, Ethernet, entre outras) [50], fazem deste um microcontrolador com uma excelente performance e com um preço relativamente baixo. Para além destas razões, o facto de existir experiência prévia com o mesmo,

assim como a disponibilidade de placas de desenvolvimento que o incorporam, levaram à escolha do mesmo.

É desejado um elevado número de componentes para este controlador, de modo a que seja possível utilizá-lo em todos os sistemas fabricados pela Ecoama, e de maneira a que possa proporcionar funcionalidades que não eram possíveis antes. Na tabela 3.2 encontra-se uma lista das mesmas, assim como os aparelhos em que estas seriam utilizadas e o seu grau de importância.

Tabela 3.2 – Funcionalidades desejadas para o controlador.

	Funcionalidade/ Componente	Importância	Recuperador de Calor	Caldeira	Queimador
Actuadores	Resistência de Ignição	Obrigatório	●	●	●
	Motor de Alimentação	Obrigatório	●	●	●
	Ventoinha Ambiente	Obrigatório	●	○	-
	Ventoinha de Exaustão	Obrigatório	●	●	●
	Bomba de Água	Obrigatório	○	●	○
	Ventoinha para Divisão Extra	Média	○	-	-
	Segunda Bomba de Água	Baixa	-	○	-
	Sistema de Limpeza Automática	Alta	○	○	-
Sensores	Sensor de Temperatura Ambiente / Água	Obrigatório	●	●	○
	Sensor de Temperatura de Gases de Escape	Obrigatório	●	●	●



	Sensor de Detecção de Chama	Obrigatório	○	○	●
	Segundo Sensor de Temperatura Ambiente / Água	Alta	○	○	-
	Sensor de Nível de Pellets	Baixa	○	○	○
	Sensor de Hall para Ventoinha de Exaustão	Baixa	○	○	○
	Sensor de Fluxo de Ar	Média	○	○	○
	Sensor de Temperatura da Placa	Obrigatório	●	●	●
Segurança	Sensor de Pressão na Câmara	Média	○	○	○
	Sensor de Temperatura dos Pellets	Média	○	○	○
	Sensor para Detectar Abertura da Porta	Baixa	○	○	○
Outras Funcionalidades	Interface para Utilizador	Obrigatório	●	●	●
	Interface para comunicação com PC	Obrigatório	●	●	●
	Interface para Comunicação sem Fios	Baixa	○	○	○
	Interface para Comunicação com outros aparelhos	Média	-	○	○
	Pilha	Média	○	○	○

- Não Aplicável; ○ Opcional; ● Obrigatório

3.2.1 – Módulos de Potência

O circuito utilizado em todas as interfaces de potência foi o mesmo, apresentado na Figura 3.2. Este é adaptado do modelo apresentado no *datasheet* do acoplador óptico MOC 3020 [51]. A este foram adicionados um *snubber*, composto pela resistência de 100Ω e o condensador de $0,1\mu\text{F}$ que se encontra em série com esta, e um varistor. O *snubber* foi feito de componentes discretos, e tem como principal objectivo a supressão da rápida subida de voltagem no Triac [52–55] (um BT139 [56]), o que pode danificá-lo. O varistor tem o mesmo propósito, conduzindo a corrente em demasia em caso de picos de voltagem, que podem ocorrer aquando do ligar ou desligar do circuito ou em caso de uma trovoada, entre outras [57–59].

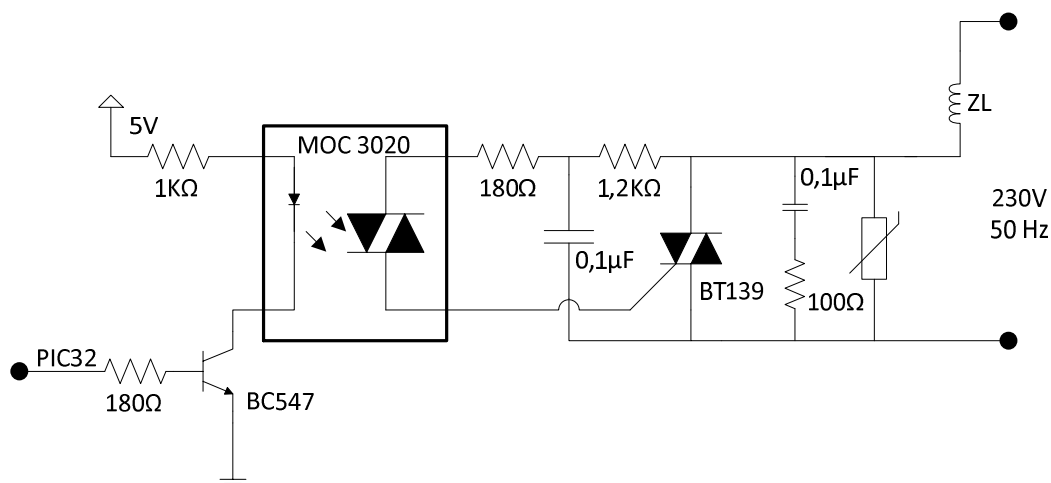


Figura 3.2 – Circuito para as interfaces de potência.

Ao acionar a saída ligada à resistência de 180Ω , o PIC32 coloca o transístor em condução, levando a que no MOC3020 o foto díodo ligue o Diac, que por sua vez vai ativar o BT139, transmitindo potência à carga ZL. Embora o circuito utilizado seja projetado tendo em vista cargas indutivas, como é o caso das ventoinhas, motor de alimentação, bomba de água, entre outros, este também pode ser utilizado para cargas resistivas, como a resistência de ignição. Optou-se por utilizar este circuito para ambos os casos de modo a dar mais flexibilidade ao controlador, uma vez que o custo e espaço ocupado pelos componentes é negligenciável.



Embora inicialmente se tenha tentado fazer o controlo das ventoinhas por contagem de ciclos, o instante inicial de cada pulso era perceptível sonoramente. Tal devia-se ao facto de que para conseguir ter um bom controlo sobre as ventoinhas, o período do PWM tinha que ser consideravelmente elevado. Uma vez que a frequência da rede é de apenas 50 Hz, para se conseguir obter uma precisão de 10%, o que já é bastante baixo, seria necessário utilizar uma frequência de PWM de 10 Hz, o que se traduz num período de 100 ms (10 meios ciclos na rede). Deste modo, optou-se por controlar as ventoinhas através de controlo de fase.

O controlo de fase é um método que usa o facto de os Triacs passarem ao estado OFF quando a tensão nos seus terminais é 0, o que acontece a cada meio ciclo. Após o Triac desligar, é necessário que haja corrente na sua *gate* para que este volte a passar ao estado ON. Assim, o controlo é feito disparando o Triac a cada meio ciclo, sendo o instante de disparo que controla a potência transmitida à carga (quanto mais tarde for o disparo menos a potência, e vice versa) [60–62]. Na Figura 3.3 encontram-se as formas de onda deste circuito com uma carga resistiva (com uma carga indutiva a corrente e a voltagem não se encontram em fase).

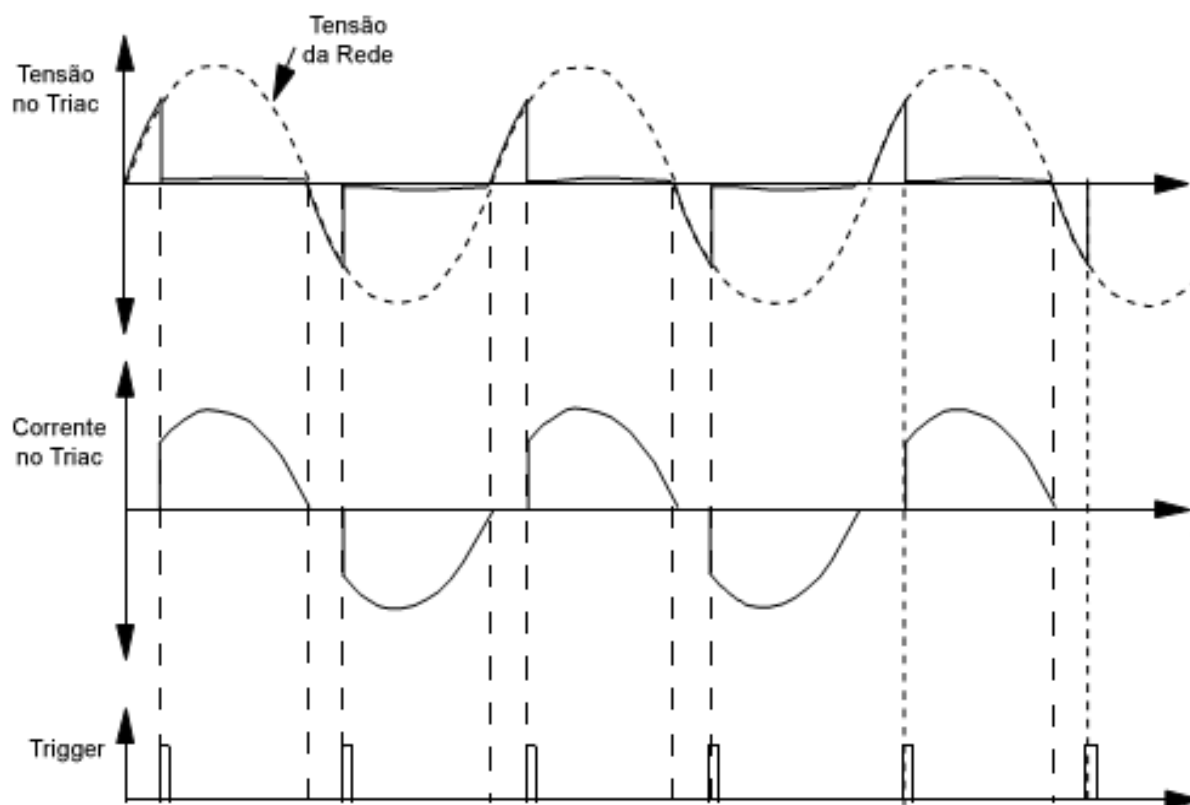


Figura 3.3 – Formas de onda num Triac (adaptado de [61]).

3.2.2 – Alimentação e Detector de Passagem por Zero

O uso deste método leva a que seja necessário detectar a passagem por zero da tensão da rede. Optou-se por fazer esta detecção após o transformador, de para evitar ligar componentes diretamente aos 220V. O circuito utilizado encontra-se na Figura 3.4. Durante a maior parte do tempo, o transístor Q1 conduz em modo de saturação, fazendo com que a tensão no *Trigger* seja aproximadamente 0. Quando a tensão na sua base desce o suficiente para levar o transístor a entrar em modo de corte, este deixa de conduzir, passando a tensão no *Trigger* a ser 3,3 V. Após a voltagem na base do transístor voltar a subir, o transístor volta ao modo de saturação, e a voltagem no *Trigger* volta a ser aproximadamente 0 V, levando assim à criação de pulsos.

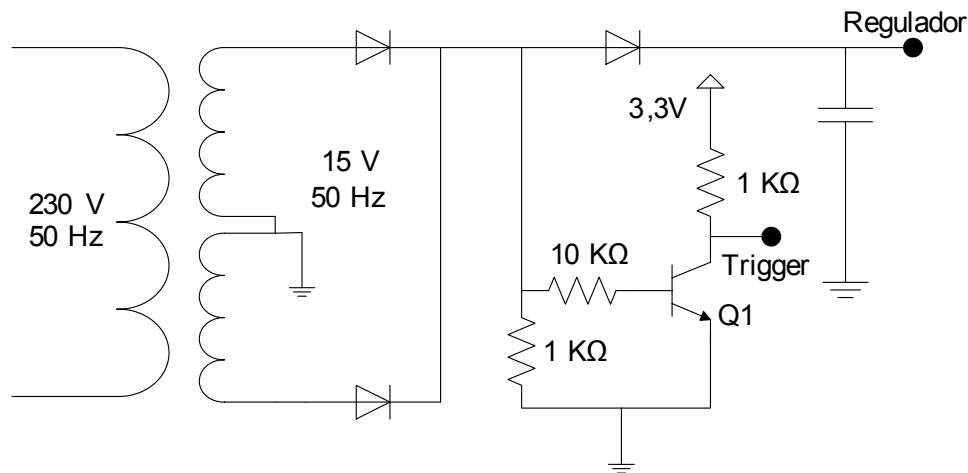


Figura 3.4 – Circuito de deteção de passagem por zero (adaptado de [63]).

Foram utilizados 3 reguladores no sistema, de 12, 5 e 3,3 V. O regulador de 12 V é necessário para a alimentação da placa de desenvolvimento que foi utilizada, que será discutida mais à frente. Os de 5 e 3,3 V são utilizados para os vários circuitos de acondicionamento.

3.2.3 – Sensores de Temperatura

O termopar do tipo K tem uma sensibilidade de aproximadamente $40 \mu\text{V}/^\circ\text{C}$ [64], tendo esta pequenas variações ao longo da sua escala. Como foi referido anteriormente, a temperatura dos gases de escape nunca deverá ultrapassar os 120°C . Assim, uma vez que a voltagem produzida pelo sensor a 150°C será de apenas 6,1 mV, é necessário amplificar este sinal para efetuar uma leitura com maior precisão. Para além deste andar de ganho, que foi construído com base num OPAMP de instrumentação *rail-to-rail*, tendo sido desenhado para um ganho de 180 (obtido através da utilização de uma resistência de 560Ω , segundo a fórmula presente em [66], $\text{Ganho} = \frac{100k}{R_g} + 1$) foram também implementados filtros RC, de modo a diminuir o ruído elevado a que o termopar é susceptível [65,66].

Para efetuar a leitura da temperatura da PCI, para efetuar a compensação do termopar, foi utilizado um sensor de temperatura LM335. Este sensor mede

temperaturas compreendidas entre os -40 e os 100 °C. O seu output a 25 °C é de $2,98$ V, variando este 10 mV/°C [67]. Para o acondicionamento deste sensor foi utilizado um OPAMP numa configuração diferencial, ilustrado na Figura 3.5, de modo a retirar o *offset* ao sinal e amplificando-o. Ao utilizar resistências de baixo para a excitação do sensor e do zéner (1K Ω e 220 Ω respectivamente), estas podem ser ignoradas no cálculo do ganho do amplificador.

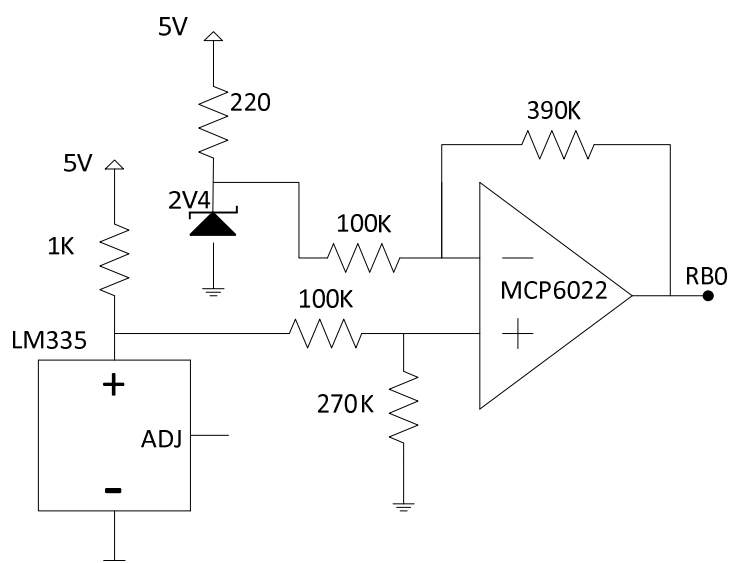


Figura 3.5 – Circuito para acondicionamento do sensor LM335.

Quanto à sonda de temperatura NTC, sendo esta um sensor resistivo, foi projetada uma fonte de corrente de cerca de 110 μ A para a excitar (Figura 3.6). É utilizado um OPAMP *rail-to-rail* em configuração de *buffer* para evitar que danos na sonda criem problemas no microcontrolador. O sensor tem uma resistência de 27285 Ω a 0 °C, descendo esta até 974 Ω quando este se encontra a 100 °C [68]. Esta variação na resistência do sensor associada à corrente de 110 μ A traduz-se numa variação de tensão de 3 V (a 0 °C) a 100 mV (a 100 °C). Deste modo, o circuito projetado para este sensor permite a que este seja utilizado tanto para a medição da temperatura ambiente como da temperatura da água, dependendo da aplicação desejada.

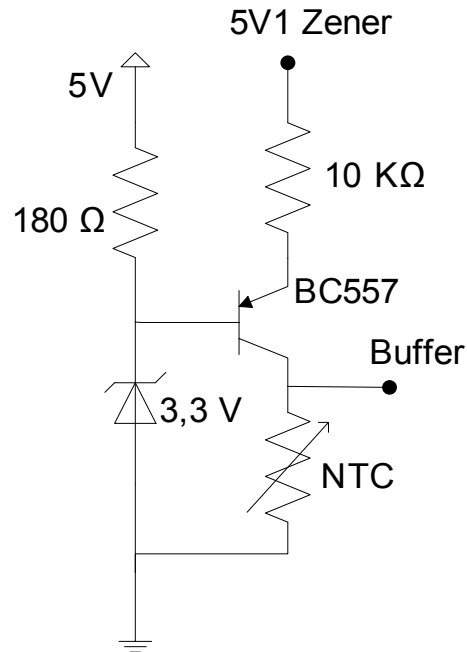


Figura 3.6 – Circuito de acondicionamento para a sonda NTC.

Por último, foi ainda criado um circuito de acondicionamento para o sensor de chama, de modo a que o *hardware* projetado seja compatível com qualquer um dos modelos mais simples dos 3 tipos de aparelho. Também este é um sensor resistivo. A sua resistência diminui conforme é exposto a uma maior intensidade de luz. Foi utilizado um OPAMP numa configuração de comparador com histerese para evitar que pequenas alterações na luminosidade não correspondentes à extinção ou ignição da chama fossem sinalizadas como tal, assim como para evitar que no processo de ignição algum ruído no sensor pudesse enviar um sinal errado ao controlador.

3.3 – Placas Utilizadas

3.3.1 – PIC32–MAXI–WEB

Devido à dificuldade e tempo necessário para projetar de raiz uma PCI com um microcontrolador incorporado, foi decidido usar uma placa de desenvolvimento. Inicialmente foi utilizada a placa DETPIC, mas devido ao facto de esta não possibilitar o uso da interface CAN, uma vez que estes pinos não se encontram acessíveis [69], foi necessário encontrar uma alternativa.

A escolha recaiu na PIC32-MAXI-WEB (Figura 3.7) [72], uma placa criada pela Olimex. Esta placa baseia-se num microprocessador PIC32MX795F512L, o mais desenvolvido da família PIC32. Para além de todas as características fornecidas pelo microcontrolador, a placa tem ainda vários componentes adicionais que fornecem outras funcionalidades, como a leitura de cartões de memória micro SD, 3 botões físicos para o utilizador, interfaces CAN, Ethernet e USB, RS-232, um ecrã *Touchscreen* LCD de 3,2 polegadas com retroiluminação LED, entre outras. De todas estas características, o *Touchscreen* é a mais útil, uma vez que permite a construção de uma interface gráfica, apelativa ao utilizador, sem as limitações inerentes a uma interface *standard*, com um display LCD e botões de pressão.



Figura 3.7 – PIC32-MAXI-WEB [70].

A placa está ainda equipada com outros elementos que poderiam ser utilizados no sistema considerado, como é o caso dos relés, podendo um ser usado para a resistência de ignição, ou os dois *inputs* digitais que poderiam ser usados para a detecção de passagem por zero ou para o sensor de detecção de chama. No entanto, optou-se por não se utilizar os mesmos, de modo a reduzir a dependência da placa, uma vez que o uso da mesma apenas se justifica para uma fase de prototipagem, devido ao seu elevado custo. O facto de a placa ser alimentada a 12 V justifica a presença de um regulador 7812 na PCI desenvolvida. A comunicação entre as placas é efectuada através de 2 conectores UEXT, encontrando-se ainda alguns pinos disponíveis nos mesmos. A comunicação com o PC é feita através de UART, usando a interface RS-232 presente na placa PIC32-MAXI-WEB.

3.3.2 – PCI

De modo a criar as interfaces necessárias para a comunicação da placa PIC32-MAXI-WEB e os sensores e actuadores utilizados no sistema, foi necessária a criação de uma PCI, apresentada aqui na Figura 3.8.

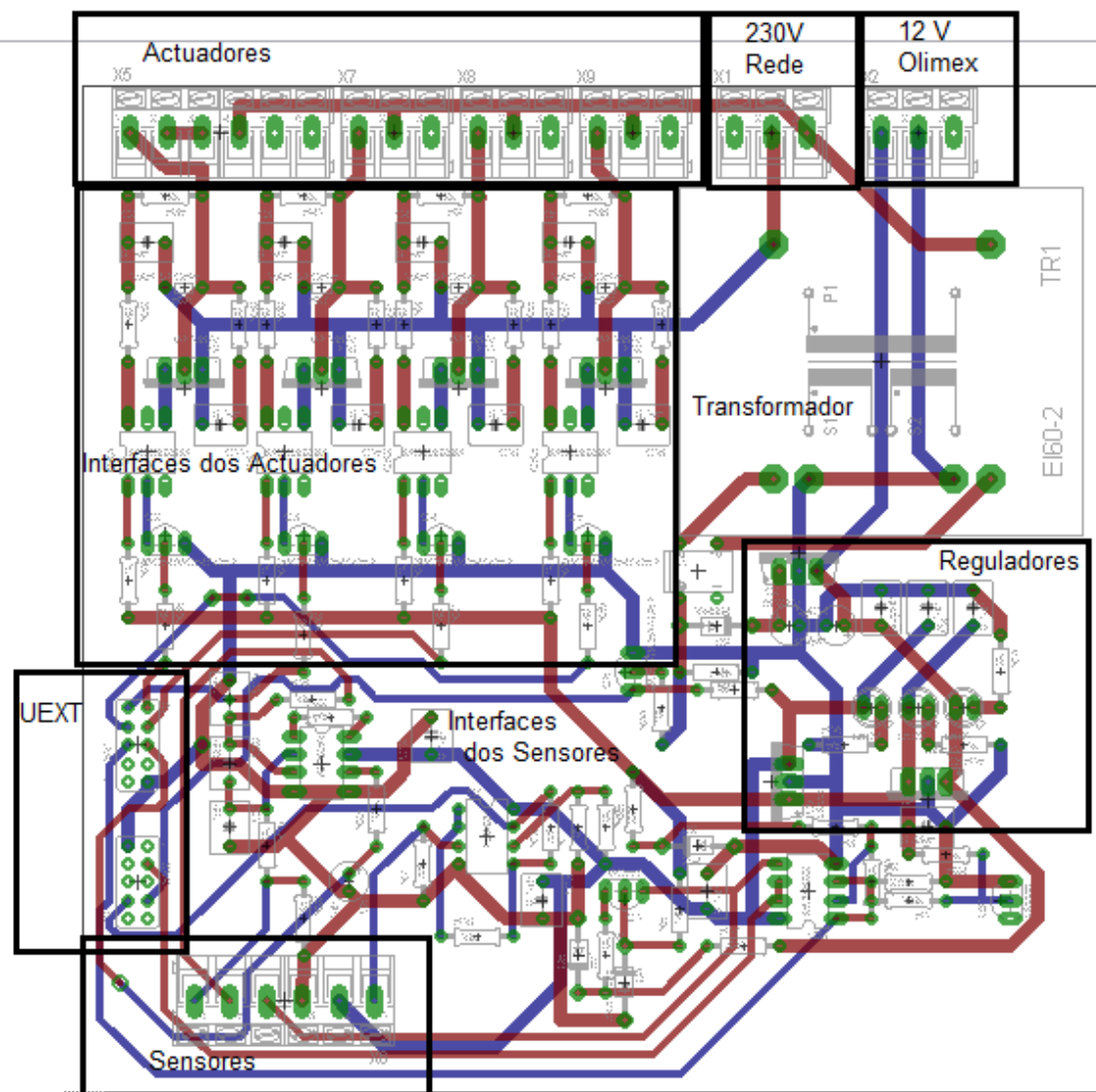


Figura 3.8 – PCI.

Como é possível observar na figura, tentou-se fazer uma separação entre os vários blocos do sistema, de modo a facilitar o *debug* da placa. Esta encontra-se equipada



para controlar qualquer um dos 3 aparelhos discutidos, desde que estes não apresentem mais que 4 actuadores e não necessitem de mais que uma sonda NTC.

Decidiu-se para este protótipo inicial manter todos os componentes do circuito na mesma face da PCI, de modo a tornar mais fácil a medição de voltagens nestes. Tal leva ao elevado tamanho da placa, que poderia ser reduzido substancialmente com algumas modificações. Outras limitações advêm do facto de a placa ter sido feita *in-house*, não sendo assim possível ter furos estanhados, assim como a largura mínima para as pistas ser bastante elevada. Idealmente, o transformador utilizado será substituído por um de menor tamanho, uma vez que o utilizado é excessivamente largo, e excede os requisitos necessários.

IV. Firmware e Software



Neste capítulo são apresentados o *firmware* e *software* desenvolvidos para o controlador. Primeiramente é apresentado o *firmware*, um programa pequeno, responsável por controlar o *hardware* do sistema. Após este, é apresentado o *software* projetado, sendo neste caso um simples programa que permite ao fabricante de um aparelho programar o controlador de acordo com as suas especificações.

4.1 – Firmware

O *firmware* é o código presente no processador do controlador. Este é por norma responsável pela realização de uma série de tarefas de nível bastante baixo sem as quais um aparelho não seria funcional [73]. Este não deve ser alterado pelo utilizador, nem pelo fabricante, sendo no entanto possível que este último configure os valores de algumas das variáveis com recurso ao *software* desenvolvido.

4.1.1 – Device Drivers

Antes de proceder à criação do algoritmo que vai permitir efetuar o controlo de cada sistema, é necessário primeiro conseguir efetuar os procedimentos básicos sobre cada um dos componentes deste. As rotinas que permitem a realização destas operações de I/O são chamadas de *device drivers*.

4.1.1.1 – Sensores

Existem vários modos de implementação dos *device drivers*. Para a leituras das temperaturas, por exemplo, poderia ser criada uma rotina que tinha como parâmetro de entrada apenas o sensor que se pretendia ler, sendo então a rotina responsável pela leitura do *analog to digital converter* (ADC), e conversão para a unidade pretendida. Por outro lado, é possível tornar o *device driver* apenas responsável pela leitura do valor que se encontra na ADC e devolução do mesmo “em bruto”.

A solução adoptada para os sensores foi um compromisso entre as duas descritas acima. Foi criada uma função (`readADC`), responsável por efetuar a leitura para um *array* do valor de um dos canais do ADC (sendo o número do canal passado à rotina quando esta é chamada), e posterior conversão para tensão, sendo este valor devolvido em milivolts. Para além disto, é ainda feita uma filtragem simples. Esta filtragem é feita através da leitura de 4 valores do ADC, eliminando posteriormente os extremos superior e inferior, e sendo o valor a ser devolvido obtido através do cálculo da média dos 2 valores restantes. Este processo permite eliminar o erro aleatório. O número de medidas realizadas é suficiente para que o erro seja eliminado, e, ao ser baixo, não leva a uma maior demora na realização da leitura. Um esquema simplificado desta função encontra-se abaixo, na Figura 4.1.

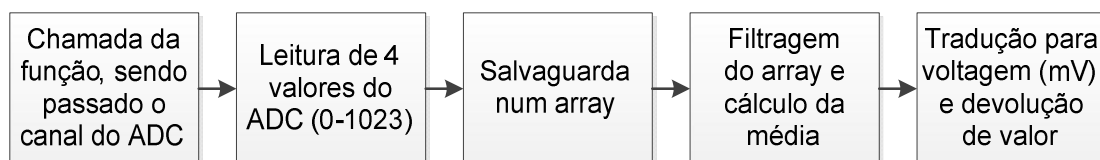


Figura 4.1 – Esquema simplificado da função `readADC(int channel)`.

Após obter os valores através da função descrita acima, é necessário processá-los de modo a poder atualizar as diversas variáveis com os novos valores de temperaturas.

Para a sonda da temperatura ambiente, é utilizada a rotina `Upd_Amb_Temp()`, que é responsável por invocar a função de leitura do ADC, passando o número do seu canal como argumento, e, após a devolução do seu valor, fazer a conversão para um valor de temperatura, segundo a fórmula abaixo, onde V representa a tensão lida no ADC, igual à resistência da sonda multiplicada pela corrente que a atravessa (0,11 mA). Esta é calculada através de uma regressão quadrática obtida através da tabela presente em [70].

$$T_{amb} = 7,7 * V^2 - 40,71V + 60,579$$



Após alguns testes, verificou-se que por vezes existia uma variação bastante elevada no valor lido, o que não poderia corresponder à realidade, uma vez que a temperatura de uma divisão não pode à partida ter uma oscilação rápida. Assim, foi adicionado um segundo nível de filtragem, para eliminar estes valores. É utilizado um outro *array*, com 10 posições, onde é colocado o novo valor lido do ADC e eliminado o mais antigo. Após isto, o *array* é ordenado por ordem crescente de valores de temperatura. Finalmente, é feita a média dos 2 valores intermédios (posições 4 e 5), sendo este o valor que é copiado para a variável correspondente à temperatura ambiente (Temp_Amb). Esta filtragem leva a uma resposta mais lenta do sensor, pois se o valor lido for um extremo este é ignorado durante algumas leituras, mas uma vez que a alteração da temperatura ambiente também é por si demorada, e a rotina é corrida com frequência, este aspecto acaba por não ter muito peso.

Para ser possível efetuar a compensação da junção fria do termopar é necessário ler a temperatura a que se encontra a placa. Uma vez que o sensor utilizado, um LM335 apresenta uma diferença de potencial de 2,98 V à temperatura de 25 °C, subindo este valor 0,01V/°C, foi necessário utilizar um amplificador operacional em configuração diferença. Para referência foi utilizado um díodo de zéner de 2,4 V. No entanto, verificou-se que o valor deste variava de forma significativa. Tal pode ser causado por ruído de alta frequência, o que poderia ser solucionado através de um filtro passa-baixo. No entanto, esta abordagem torna necessário proceder a alterações na PCI já criada, o que era indesejável. Uma vez que existiam bastantes pinos analógicos disponíveis, optou-se por também efetuar a leitura deste valor, de modo a conseguir obter uma maior precisão na leitura do valor da temperatura da placa. Assim, a rotina Upd_Brd_Temp() invoca duas vezes a rotina readADC, para dois canais distintos, e utiliza as fórmulas abaixo, onde V_1 representa a tensão do díodo de zéner, V_2 a voltagem do sensor LM335, $R_1 = R_2 = 100K\Omega$, $R_f = 390K\Omega$ e $R_g = 270K\Omega$ para calcular o valor da temperatura da placa, atualizando a variável correspondente, Brd_Temp.

$$T_{brd} = (V_{out} - 2,98) * 100 + 25$$

$$V_{out} = \frac{R_1 + R_f}{R_1} * \frac{R_g}{R_g + R_2} * V_2 - \frac{R_f}{R_1} * V_1$$

Já a leitura do termopar, efectuada pela função `Upd_Flue_Temp()`, verificou-se ser mais fácil, uma vez que este sensor apresenta uma precisão superior aos discutidos previamente. Para a medição da temperatura dos gases, a função invoca a rotina `readADC`, passando como argumento um inteiro que identifica o canal do ADC a ser lido, e após o retorno do mesmo, aplica a seguinte fórmula, obtida através da tabela presente em [65] e [74], para o cálculo da temperatura:

$$T_{flue} = T_{brd} + \frac{(v - v_0) * (p_1 + (v - v_0) * (p_2 + (v - v_0) * (p_3 + p_4 * (v - v_0))))}{1 + (v - v_0) * (q_1 + (v - v_0) * (q_2 + q_3 * (v - v_0)))}$$

No entanto, esta temperatura é calculada em relação ao ponto de referência para o termopar, que neste caso é o local em que este se liga à PCI. Assim, é necessário então adicionar à temperatura calculada através da fórmula acima a temperatura a que se encontra a placa (obtida através da função `Upd_Brd_Temp`). Finalmente, é atualizada a variável correspondente à temperatura dos gases de escape, `Temp_F_Gas`.

4.1.1.2 – Actuadores

Para a criação das rotinas para os actuadores foi feita uma abordagem diferente. Tendo cada um destes modos de operação distintos (com exceção das duas ventoinhas, que funcionam ambas por controlo de fase e com um período igual), como apresentado no diagrama de blocos apresentado na Figura 2.8, foram criadas rotinas bastante diferentes para cada um.

Sendo a resistência de ignição o actuador com o modo de controlo mais simples (apenas ON/OFF), também as funções criadas para esta são as menos complexas. Foram criadas para este propósito duas rotinas, `Igniter_ON()` e `Igniter_OFF()`. De acordo com o nome, a primeira liga a resistência de ignição, através da colocação



de um pino do microcontrolador (RD10) ao nível lógico 1, ao passo que a segunda coloca este mesmo pino a 0.

Para o motor de alimentação, também as rotinas responsáveis pela interação direta com este são bastante simples. Foi criada uma rotina com o propósito de colocar o motor em operação, `Feeder_ON()`, e uma para o desligar, `Feeder_OFF()`. De modo semelhante às funções criadas para a interação com a resistência de ignição, as funções alteram o valor de um pino (RD9), consoante seja pretendido que o motor esteja parado ou a funcionar. No entanto, como foi discutido anteriormente, no capítulo 3, pretende-se que o motor trabalhe um certo número de segundos, para depois parar durante o resto do seu período, isto é, que opere em PWM, com um *duty cycle* variável. Para o controlo destes períodos de tempo, foram criadas outras variáveis, como é o caso do contador `count`, do tipo `int`, que é incrementado a cada segundo, sendo feito um *reset* ao mesmo pelas duas rotinas descritas acima, de cada vez que são chamadas. A variável é também colocada a zero noutros processos, discutidos na segunda parte desta secção. Para o incremento desta variável, foi utilizado o *timer 5* do microcontrolador PIC, tendo sido criada uma rotina de serviço à interrupção (RSI), com um período de 125 ms. Uma vez que não é possível ter um *timer* de 16 bits com um período de 1 segundo, devido ao valor necessário para o comparador ser demasiado elevado, e estando os *timers* todos em utilização, optou-se por usar um valor mais baixo, que permite também obter uma precisão maior futuramente (tornando possível ajustar o *duty cycle* num intervalo inferior ao utilizado presentemente, de 1 segundo).

São ainda utilizadas outras três variáveis adicionais, `Feeder_T_Total`, `Feeder_On_Startup` e `Feeder_T_ON`. A primeira corresponde ao período do funcionamento do motor durante a fase normal de queima do recuperador de calor (tendo sido determinado um valor *default* de 10 segundos), ao passo que a segunda estabelece o tempo de operação do motor na fase de arranque do sistema, antes de se dar a ignição. A última variável é um *array* com 6 posições que determinam o tempo de trabalho do motor durante a fase de queima (determinando assim o seu *duty cycle*), sendo cada uma correspondente a um dos diversos graus de poder de

queima em que o sistema pode operar, estando a primeira posição fixa a 0, o que na prática indica que o motor se encontra parado. Qualquer uma destas variáveis pode ser alterada pelo fabricante dos aparelhos, de modo a proceder a um ajuste dependendo do poder de queima pretendido em cada nível.

O controlo das duas ventoinhas é feito de forma idêntica entre estas. Foram criadas 2 rotinas por ventoinha, `AmbFan_ON()` e `AmbFan_OFF()`, relativas à VA, e `FlueFan_ON()` e `FlueFan_OFF()`, relativas à VE. Estas duas funções alteram o valor de comparação para o *timer* respectivo à ventoinha que controlam (PR2 e PR3). Os *timers* utilizados, 2 e 3, vão sendo incrementados até atingirem os valores destas variáveis, gerando uma interrupção quando tal acontece. No caso das rotinas responsáveis por desligarem as ventoinhas, este valor é fixo, de 49999. Este valor foi seleccionado por ser o que impõe uma frequência de 100 Hz aos timers, igual à frequência dos meios-ciclos da rede eléctrica, permitindo assim o ajuste da potência por meio de controlo de fase. As rotinas `_ON()`, alteram este valor para que o *trigger* à ventoinha correspondente seja aplicado no instante determinado, de modo a criar o efeito demonstrado na Figura 3.3. As RSIs, para além de fazerem um *reset* ao timer correspondente e à sua *flag* de interrupção, verificam se a variável correspondente à ventoinha que controlam (`Amb_Fan` ou `Flue_Fan`) se encontram a 1. Nesse caso, fazem o *set* ao pino correspondente à sua ventoinha (RD0 para a VA ou RC4 para a VE). O ciclo *while* na função `main` verifica constantemente o estado destes dois pinos, e aplica-lhes um *reset* no caso de estes estarem com o valor lógico de “1”, concluindo o processo de criação do trigger.

De modo a ser possível existir sincronismo entre o controlo feito sobre as ventoinhas e a tensão na rede eléctrica, é utilizado o circuito detector de passagem por zero apresentado anteriormente. É gerada uma RSI cada vez que existe uma alteração no valor lógico do pino RF4, onde se encontra ligado o detector. Esta rotina vê o valor do pino, e se este se encontrar com o valor lógico “1”, indicando uma *rising edge*, verifica os valores das variáveis `Amb_Fan` e `Flue_Fan`. No caso de algum destes ser “1”, a rotina coloca o valor do *timer* correspondente a zero e liga-o, para que este inicie a sua contagem.



Inicialmente, a abordagem utilizada para o controlo das ventoinhas era bastante simples. No caso de se pretender uma ventoinha a funcionar a 80%, era dado o *trigger* após passarem 2 ms de uma passagem por zero, por exemplo. Tal levava a que fosse entregue potência à ventoinha durante 8 ms, uma vez que existe um novo meio ciclo na rede a cada 10 ms. No entanto, devido à não linearidade das ventoinhas e ao desfasamento existente entre a tensão e a corrente na rede, verificou-se que este método não atingia os resultados desejados, causando ao invés que a partir de um certo ponto a ventoinha rodasse muito lentamente, e que apenas começasse a rodar quando a potência aplicada era superior a um dado valor. Assim, optou-se por medir estes 2 limiares para ambas as ventoinhas, e reajustar os instantes de *trigger* para as desejadas potências.. Como efeito secundário, esta configuração permite também uma maior precisão, uma vez que uma mudança de 1% vai corresponder a uma alteração no instante de *trigger* menor que anteriormente. Verificou-se, através da medição exaustiva de um conjunto de valores, que os triggers deveriam variar entre os 4,5 (o valor máximo antes de a ventoinha começar a reduzir a velocidade) e os 6,7 (o mínimo para que a ventoinha tivesse potência suficiente para começar a rodar) ms, após a passagem por zero, no caso da VA, e entre os 4 a 6 ms para a VE. Os gráficos presentes nas Figuras 4.2 e 4.3 representam este cenário.

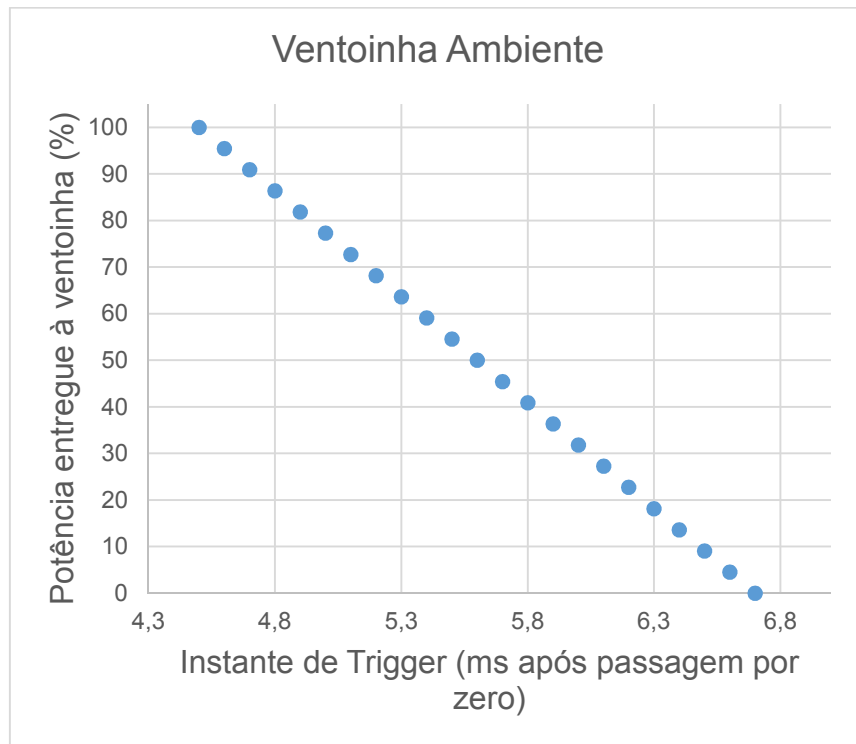


Figura 4.2 – Potência entregue à VA relativamente ao instante de trigger.

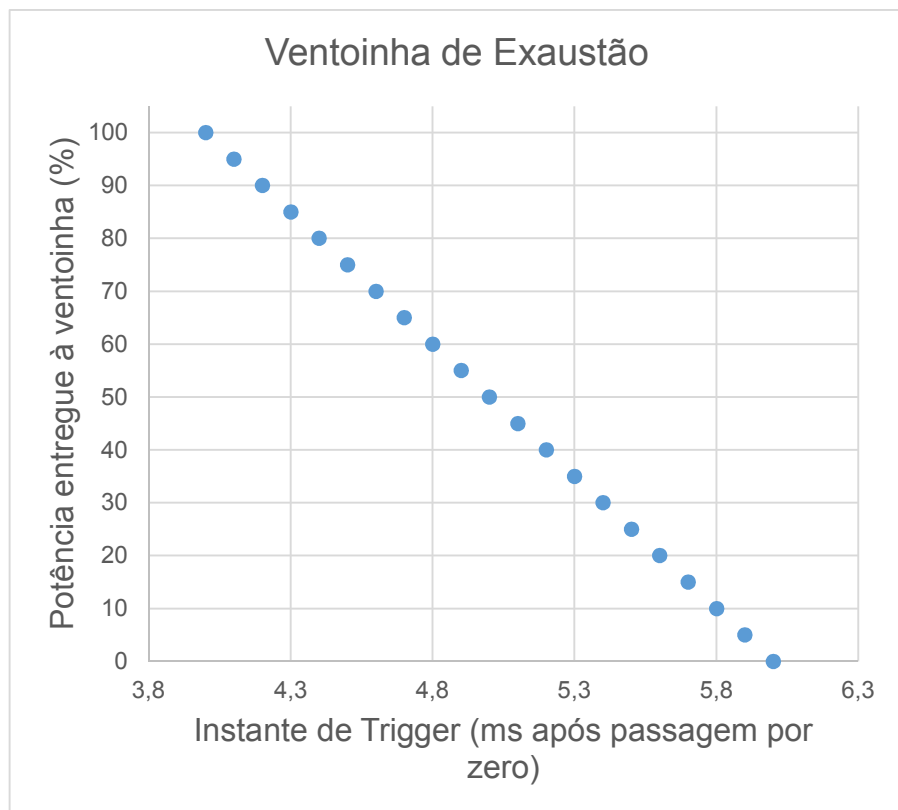


Figura 4.3 - Potência entregue à VE relativamente ao instante de trigger.

4.1.2 – Processo de Controlo

Após a verificação do correto funcionamento dos *device drivers*, procedeu-se à criação de rotinas para fazer a ligação entre estes de modo a que fosse possível realizar o controlo da caldeira. O processo que se pretende controlar é o identificado pela máquina de estados presente na Figura 4.4. Em qualquer um dos estados apresentados, exceptuando o “Idle”, existem várias variáveis a serem observadas, e é necessário utilizar vários actuadores para que o sistema tenha o comportamento desejado.

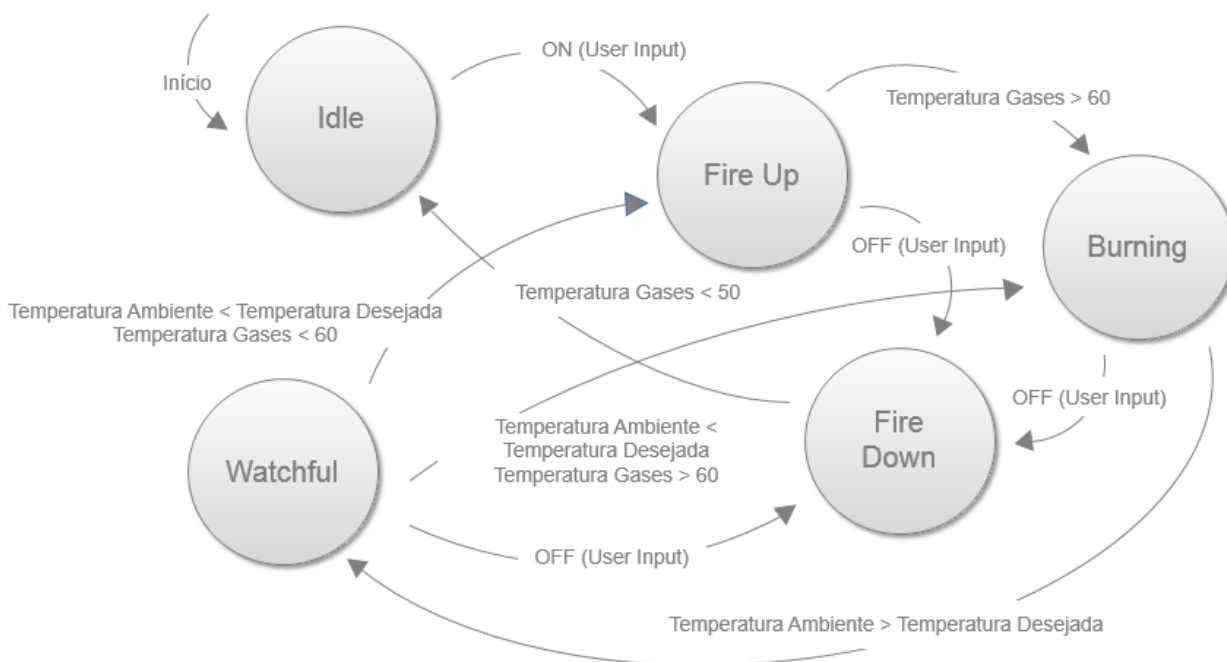


Figura 4.4 – Diagrama de estados do recuperador de calor Hefesto.

É utilizada uma função *main*, representada no fluxograma da Figura 4.5, com um ciclo *while* infinito para o controlo do recuperador de calor. Esta função é responsável por invocar a rotina `InitializeBoard()`, que vai proceder a toda a configuração inicial do controlador necessária, nomeadamente:

- Verificação da performance do sistema e verificação da frequência do relógio (80 MHz);

- Configuração da *Universal Asynchronous Receiver/Transmitter* (UART) e respectivas interrupções, para possibilitar a comunicação com um computador (para efeitos de *debugging*, manutenção e configuração pelo fabricante);
- Inicialização do *Graphics Object Layer* (GOL), que utiliza a *Microchip Application Library* (MAL) para permitir a interação entre o utilizador com o Touchscreen;
- Configuração dos pinos I/O, *timers* e ADC do microcontrolador PIC32;
- Calibração e arranque do *Touchscreen* (a calibração apenas é necessária uma vez, ou no caso do microcontrolador perder a sua memória *flash*).

Após isto, a função entra no ciclo *while*, onde realiza as seguintes tarefas:

- Leitura e processamento de mensagens enviadas pelo e para o *Touchscreen* (novos objetos a serem desenhados, objectos a serem alterados, e objetos que tenham sido seleccionados pelo utilizador);
- Verificação do estado dos pinos correspondentes às ventoinhas, colocando-os com o valor lógico “0”, finalizando assim o *trigger* (este processo poderia ser optimizado recorrendo aos *timers* já utilizados para colocar os pinos a “1”, devendo esta alteração ser efectuada futuramente);
- Leitura da *flag* da UART, e invocação da rotina UARTCheck() para lidar com mensagens recebidas por esta em caso de necessidade;
- A cada segundo:
 - Comparação entre o contador e o tempo a que o MA se encontra em funcionamento ou desligado, de modo a determinar se este deve ser ligado ou desligado;
 - Actualização dos valores de temperaturas, através das funções Upd_Amb_Temp e Upd_Flue_Temp (que por sua vez corre a Upd_Brd_Temp);
 - Invocação da rotina StageCheck(), responsável por verificar o estado em que o sistema se encontra, assim como os valores das várias



variáveis do mesmo, de modo a determinar se o estado deve ser alterado.

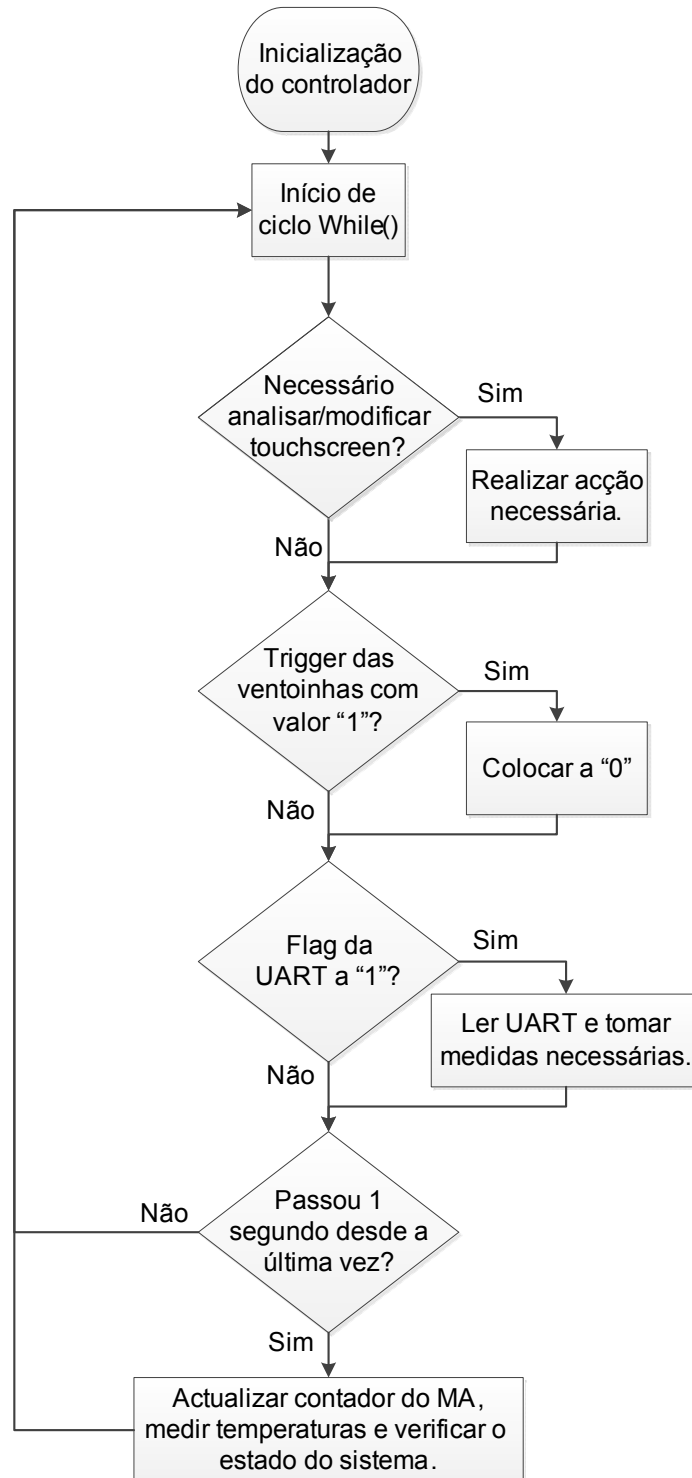


Figura 4.5 – Fluxograma da função main.

A função StageCheck é uma das funções mais importantes para o correto funcionamento de todo o sistema, devido a ser esta que efetua as transições entre estados, e é nela que se encontram implementados os vários mecanismos de segurança por *firmware* do sistema. São efectuados os seguintes processos:

- Verificação da temperatura dos gases de escape. No caso de esta ser superior a 130 °C (valor por defeito, podendo este ser alterado pelo fabricante), passagem automática para o estado de Fire Down, uma vez que uma temperatura tão alta indica uma falha no sistema;
- Leitura do estado do sistema (salvaguardado numa variável inteira, stage). Chamada da rotina Igniter_OFF(), no caso de este se encontrar ligado sem que o estado do sistema seja Fire Up;
- É por último feito um *switch* ao estado do sistema.

No caso de o estado ser Idle, é verificada uma variável, chamada “*maintaining*”, que indica se o estado é de facto o estado Idle ou se é Watchful. O estado Watchful ocorre no caso da temperatura ambiente ultrapassar a desejada pelo utilizador. Neste estado o controlador verifica o valor da temperatura ambiente e no caso de esta descer abaixo da temperatura desejada mais que 0,5 °C, o sistema altera o seu estado para Fire Up, de modo a proceder de novo ao aquecimento da divisão. Se a variável não estiver set, a rotina não faz mais nada.

No estado Fire Up, a rotina começa por comparar o valor do contador do MA com uma variável onde se encontra o número de segundos que o mesmo deve trabalhar neste estado. Se este valor ainda não tiver sido atingido, o controlador garante que a RI se encontra ligada, assim como a VE (estando esta a trabalhar com pouca potência, apenas para garantir a circulação de ar necessária para que se dê início à combustão), e mantém o MA ligado (ou liga-o, em caso de tal não ter acontecido previamente). No caso de o valor ter sido atingido, o MA é desligado. Após isto, é verificado o valor da temperatura dos gases de escape. Se esta ultrapassar o valor estabelecido de 60 °C, tal indica que já existe combustão, e a função desliga a RI,



e passa o MA e a VI para os seus modos de operação durante o estado de Burning. Para além disto, liga também a VA com o nível de potência definido pelo utilizador.

O terceiro estado, Burning, é o estado de trabalho do recuperador, isto é, o estado em que realmente este se encontra a produzir calor para a divisão. Neste estado, a rotina StageCheck apenas verifica os valores de temperatura ambiente e dos gases de escape. No caso dos gases de escape se encontrarem acima de 90 °C, a VE começa a funcionar a 100%, de modo a conseguir arrefecer o aparelho. Por outro lado, no caso de estes se encontrarem abaixo de 70 °C, irá trabalhar apenas a 40%. Relativamente à temperatura ambiente, esta é monitorizada para que em caso de se aproximar da temperatura desejada, a velocidade da VA seja reduzida até à velocidade mínima de operação, definida como 15%. No caso de esta ultrapassar num valor superior a 1 °C o valor desejado, o controlador vai passar o sistema ao estado Watchful, até que a temperatura desça de novo.

Por último, temos o estado Fire Down. Neste estado o controlador é responsável por desligar o sistema de uma forma planeada, de modo a garantir que todo o combustível é queimado e que a chama seja então extinta. Para este efeito, são ligadas as 2 ventoinhas ao máximo, e o MA é desligado. A temperatura dos gases de escape é utilizada como indicador do final da combustão. Quando esta desce abaixo dos 50 °C, o controlador desliga as 2 ventoinhas e passa o sistema ao modo Idle. É ainda garantido que o sistema nunca fica menos de 15 segundos neste modo, como medida de segurança.

A rotina UARTCheck será discutida na próxima secção, uma vez que o seu propósito se prende na comunicação com o *software* desenvolvido, de modo a proceder à configuração do controlador por parte do fabricante dos aparelhos.

4.1.3 – Interface com o Touchscreen

Por último, a nível de *firmware*, é importante discutir o conjunto de funções utilizadas para a criação e operação da interface gráfica presente no *Touchscreen*.

A comunicação entre as rotinas criadas e o ecrã é realizada com recurso ao GOL, parte pertencente da secção de *Graphics* da MAL. Este contém recursos para a criação de vários objetos pré-definidos, como botões, *sliders*, entradas para texto, entre outros.

Para a criação de interfaces para *Touchscreens*, foi utilizado o programa Visual Graphics Display Designer (VGDD), na sua versão free [75]. Este é baseado no programa da Microchip, Graphics Display Designer (GDD), também disponível como *plug-in* para o MPLABX, sendo no entanto mais desenvolvido, na medida em que permite realizar uma melhor organização dos elementos no ecrã, permitindo alinhar os objetos uns pelos outros. Tem ainda uma outra função não presente no GDD, que é a possibilidade da criação de MasterScreens, que permitem que um grupo de objetos seja reutilizado em diferentes ecrãs.

As Figuras 4.6 e 4.7 mostram dois os ecrãs que são apresentados no *Touchscreen*, nomeadamente o correspondente ao ecrã para seleção da temperatura pretendida, e o de seleção da potência da VA. Ao lado das Figuras pode ser vista a legenda correspondente.

Os 3 primeiros elementos da barra que está na parte superior do ecrã (*menubar*) representam os 3 alarmes que podem ser acionados pelo controlador. O primeiro indica uma temperatura dos gases de escape demasiado elevada. Neste caso o controlador deve tentar parar imediatamente a combustão, assegurando que o MA e a RI se encontram desligados, desligando-os se não for o caso, e ligar as 2 ventoinhas no máximo. O segundo elemento indica a necessidade de limpeza da câmara de combustão, no caso de um acumular de cinzas excessivo. Tal pode ser detectado se durante a fase de Fire Up, passado um determinado limite de tempo, não existir ignição, isto é, não for atingido o limite de 60 °C da temperatura dos gases.

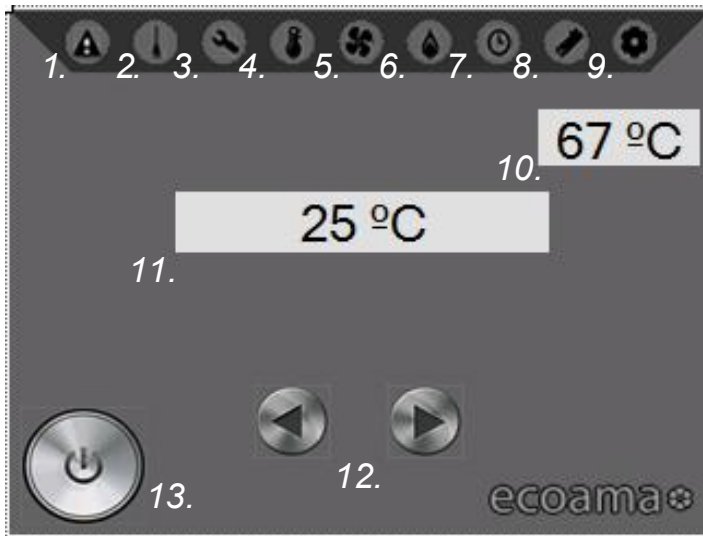


Figura 4.6 – Ecrã de seleção de temperatura desejada.

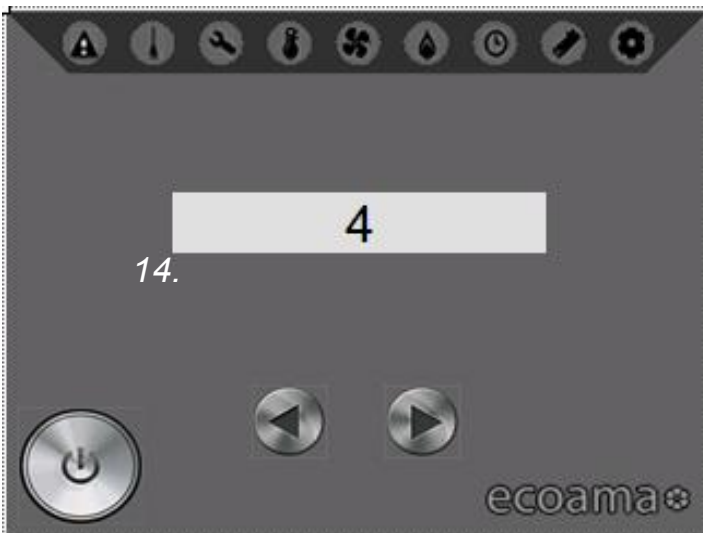


Figura 4.7 – Ecrã de seleção de potência da VA.

1. - Ícone de aviso de temperatura de gases elevada.
2. - Ícone de aviso de necessidade de limpeza.
3. - Ícone de aviso de necessidade de manutenção.
4. - Botão para selecção de menu de temperatura.
5. - Botão para selecção de menu da potência da VA.
6. - Botão para selecção de menu da potência de queima.
7. - Botão para selecção de menu de relógio, programador.
8. - Botão para selecção de menu do tipo de combustível.
9. - Botão para selecção de menu de definições.
10. - Mostrador da temperatura dos gases de escape
11. - Mostrador da temperatura desejada.
12. - Botões para alteração de parâmetros.
13. - Botão ON/OFF.
14. - Mostrador da potência da VA.

O último destes 3 ícones representa a necessidade de manutenção ao equipamento, e pode ser ativado no caso de um dos outros alarmes ocorrer repetidamente, ou for detectada outra anomalia, como por exemplo a não detecção de passagem por zero. Estes alarmes são sinalizados pelos seus ícones passarem de preto a vermelho. Neste momento apenas o alarme para o excesso de temperatura se encontra operacional.

O quarto ícone da *menubar*, primeiro botão desta, leva o utilizador para o ecrã da Figura 4.6, onde este pode seleccionar a temperatura desejada (apresentada no texto 11), através dos botões direccionais abaixo desta. Neste ecrã é ainda possível verificar a temperatura a que se encontram os gases de escape (texto 10), bem como ligar ou desligar o aparelho, recorrendo ao botão ON/OFF que se encontra no canto inferior esquerdo. O próximo botão desenha o ecrã para selecção da potência da VA, ilustrado na Figura 4.7. Como podemos observar, o texto em que no ecrã anterior se encontrava a temperatura desejada apresenta agora o nível de potência em que se encontra a VA, que pode variar de 1 a 5. Este texto indica ainda o poder de queima, no caso de estarmos no ecrã para a escolha deste, que pode ser seleccionado através do botão número 6.

O botão número 7 levaria o utilizador para um menu onde este poderia preparar o aparelho para entrar em funcionamento a uma determinada hora (ou várias horas), assim como uma hora para o mesmo se desligar. Esta função é especialmente útil para as caldeiras, onde é muito comum o utilizador desejar água quente de manhã, podendo esta estar disponível se a caldeira entrar em funcionamento mais cedo, sem que exista interação por parte do utilizador. Este menu não foi implementado devido ao facto do controlador não possuir um relógio de tempo real e esta função não ter sido considerada essencial para a fase de protótipo.

O botão seguinte permite ao utilizador ir até um ecrã onde pode seleccionar o tipo de combustível a utilizar, de entre uma lista designada pelo fabricante. Embora este menu não tenha sido implementado, uma vez que nenhum dos aparelhos analisados permite um combustível que não *pellets*, este foi pedido pela Ecoama, por existirem sistemas que permitem a utilização de lenha assim como a de *pellets*. Por fim, o último botão na *menubar* permite ao utilizador aceder a um menu onde poderá alterar algumas configurações do aparelho, como data e hora, poder da VA e de queima *default* após ligar o sistema, limite de tempo até que o *Touchscreen* entre em modo de poupança de energia, entre outras. Esta barra horizontal permanece no topo do ecrã independentemente do menu seleccionado, de modo a permitir uma navegação fácil e intuitiva pelos vários ecrãs do controlador.



Para a comunicação entre o controlador e o *Touchscreen* é utilizado um conjunto de funções implementado com recurso à MAL, sendo todo o trabalho gráfico propriamente dito, nomeadamente de desenho dos vários objetos no ecrã, realizado por funções existentes nesta biblioteca, não sendo alvo de discussão na presente dissertação.

Foram criadas 2 funções principais para realizar esta comunicação. A primeira, *GOLDrawCallback(void)*, é responsável pela transição entre os vários ecrãs disponíveis ao utilizador. É composta apenas por um *switch* a uma variável apelidada de *screenState*, que pode variar entre 3 valores por cada ecrã definido.

Uma vez que apenas estão definidos 3 ecrãs de momento (temperatura, poder da VA e poder de queima), esta variável pode ter 1 de 9 valores. Para cada ecrã existe:

- *CREATE_SCREEN_<nome_do_ecrã>*
- *UPDATE_SCREEN_<nome_do_ecrã>*
- *DISPLAY_SCREEN_<nome_do_ecrã>*

O próprio GOL vai invocar esta rotina, que, dependendo do *screenState* definido, vai ter um comportamento diferente. No caso de este indicar um *CREATE_SCREEN*, irá ser invocada uma rotina *CreateScreen_<X>(void)* (onde X pode ser *Temperature*, *Fan* ou *Power*), e é dentro destas rotinas que os objetos, bem como os esquemas de cores dos mesmos, são criados, cabendo depois ao GOL o desenho dos mesmos no ecrã.

Dependendo do objecto, um conjunto diferente de parâmetros é passado no momento de criação do mesmo (o que é feito através de funções que pertencem à MAL, como *PictCreat*, *BtnCreat*, entre outras), sendo no entanto algumas comuns entre todos, como os limites do objecto em coordenadas em x e y. Estas rotinas encontram-se num ficheiro auxiliar chamado:

VGDD_OlimexMaxiWebDemo1_Screens.c.

Após invocar a rotina para a criação do ecrã, a rotina vai alterar o `screenState` para o `UPDATE` correspondente e terminar a sua execução. No caso de o `screenState` ser um `UPDATE`, este é alterado para o `DISPLAY` que lhe corresponde, e nada mais é feito. Finalmente, no caso de termos um `DISPLAY`, a função apenas efectua um *break*.

A segunda função, `GOLMsgCallback`, é invocada pelo `GOL` quando o utilizador interage com algum dos objetos presentes no ecrã, e tem como parâmetros de entrada um ponteiro para o objecto indicado e o tipo de mensagem, onde está indicado o tipo de interação, como por exemplo o libertar de um botão. As interações com os botões são ignoradas no caso de a mensagem passada não ser de *release* de modo a evitar que um toque no *touchscreen* seja interpretado mais que uma vez.

Dentro da rotina é feito um *switch* ao objeto com que foi feita a interação, sendo depois realizados testes a variáveis de modo a verificar o tipo de interação. No caso dos botões de seleção de menu, por exemplo, é verificada se a mensagem é `BTN_MSG_RELEASED`, sendo nesse caso modificado o `screenState` para o `CREATE_SCREEN` correspondente. A ação para os botões para alterações de parâmetros é mais complexa, uma vez que estes objetos são comuns a vários ecrãs. É primeiro verificado se a mensagem é de *release*, e depois é feito um outro *switch* por `screenState`, de modo a verificar qual a variável a ser alterada (temperatura desejada, potência da VA ou poder de queima). São ainda depois efetuadas algumas verificações, de modo a avaliar se a alteração pretendida é válida, como por exemplo se o poder de queima resultante está compreendido entre 1 e 5. No caso de as condições não serem verificadas, o controlador vai ignorar o pedido. Em caso contrário, vai ser alterada a variável correspondente e atualizado o valor apresentado no ecrã.

Já o botão `ON/OFF` vai efetuar uma ação diferente dependendo do estado em que o sistema se encontra, podendo mudar o estado do sistema para `Fire Down` ou `Fire Up`, dependendo do estado prévio, podendo ainda fazer outras alterações conforme seja necessário, como um *reset* ao contador utilizado, ou desligar a VA e ajustar a



VE no caso de se desejar colocar o sistema em funcionamento e este se encontrasse no estado de Fire Down.

4.2 – Software

Foi criada uma aplicação com recurso à ferramenta Visual Studio, com vista a que seja possível à Ecoama configurar cada controlador dependendo do tipo de aparelho a que se destina, bem como proceder ao *fine tuning* de alguns parâmetros, como o *duty cycle* do MA para cada poder de queima.

4.2.1 – Aplicação para o Fabricante

O programa criado é bastante simples e intuitivo, facilitando a utilização e adaptação deste pelo fabricante, sem que seja necessária formação ou despende de muitas horas para perceber o que faz cada campo. Para utilizar a aplicação é necessário um computador com porta série e um cabo série RS 232. No caso do computador a utilizar não ter uma porta série, pode ser utilizado um adaptador USB-RS232 em conjunto com o cabo referido. Este cabo é ligado ao controlador, que é detetado automaticamente pelo computador, independentemente da porta série a que esteja ligado, através de uma troca de mensagens. A interface da aplicação utilizada para a ligação ao controlador encontra-se representada na Figura 4.8. O processo de ligação ao PC pode ser visto nas Figuras 4.9, caso em que a ligação se dá à primeira tentativa, e 4.10, onde a conexão não é estabelecida inicialmente, uma vez que o controlador não se encontrava no seu estado inicial.

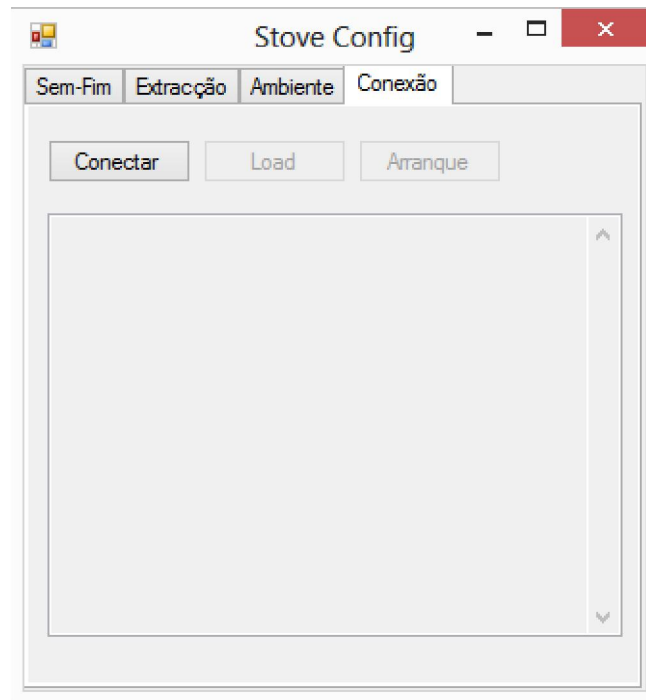


Figura 4.8 – Interface para estabelecimento de conexão com o controlador.

Como é possível observar nas Figuras seguintes, são fornecidas mensagens ao utilizador da aplicação, de modo a que este consiga fazer *troubleshoot* se tal for necessário.

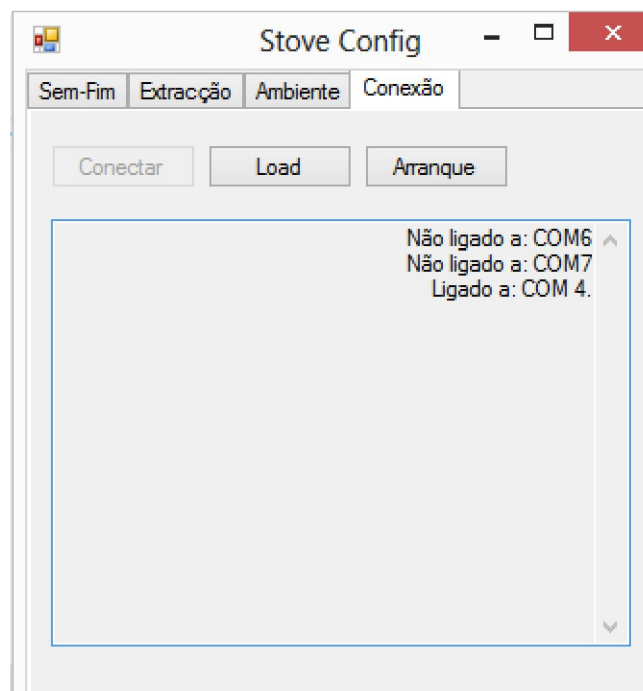


Figura 4.9 – Ligação com o controlador estabelecida.

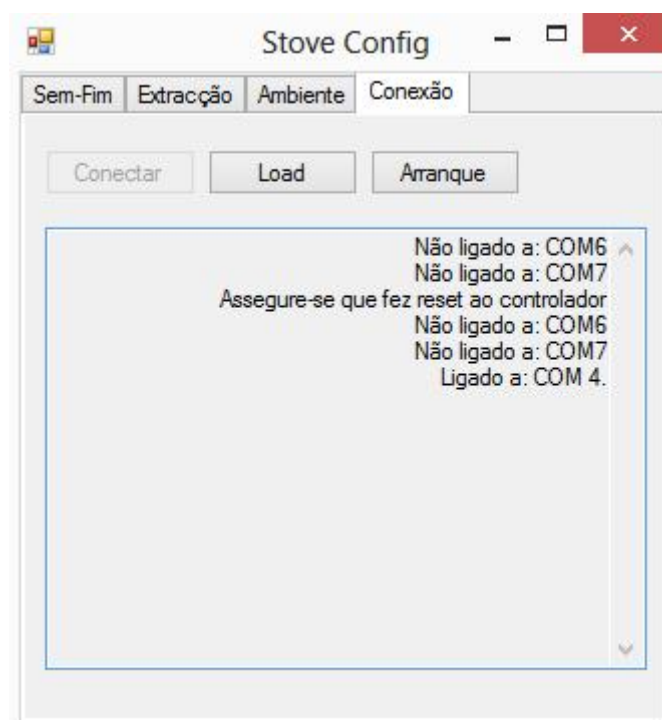


Figura 4.10 – Ligação estabelecida com o controlador após tentativa falhada.

A aplicação encontra-se preparada apenas para a configuração de controladores que tenham em vista recuperadores de calor, como o Hefesto. Assim, ao ser executada, inicia logo na *tab* “Conexão”. Idealmente, quando esta puder ser utilizada para configurar controladores para diferentes aparelhos, será perguntado ao utilizador qual o aparelho que deseja configurar, podendo ser utilizada uma outra *tab* para este efeito, ou existir um *pop-up* que peça ao utilizador para fazer a seleção antes de entrar no ecrã principal do programa. Neste momento, existem três *tabs* para além da já apresentada da conexão, uma para cada um dos atuadores do aparelho, à exceção da RI, uma vez que o controlo desta não tem opções. A primeira é a do MA, apresentada na Figura 4.11.

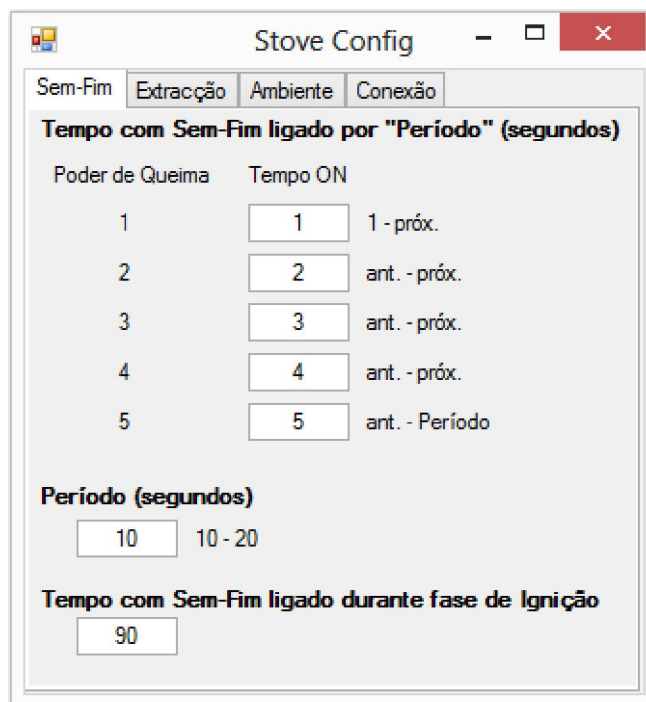


Figura 4.11 – Tab de parâmetros do MA.

Como podemos ver, a interface é bastante intuitiva, com textos elucidativos do propósito de cada campo. O mesmo se verifica para as duas ventoinhas, cujas *tabs* para configuração estão ilustradas nas Figuras 4.12 e 4.13.

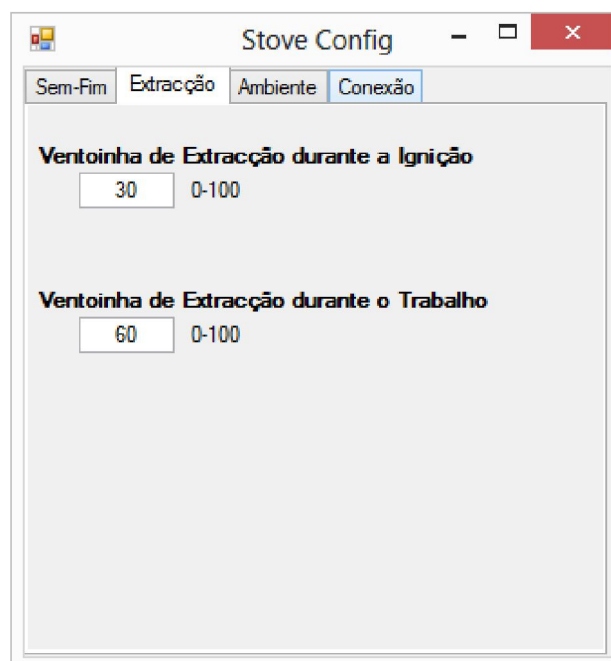


Figura 4.12 - Tab de parâmetros da VE.

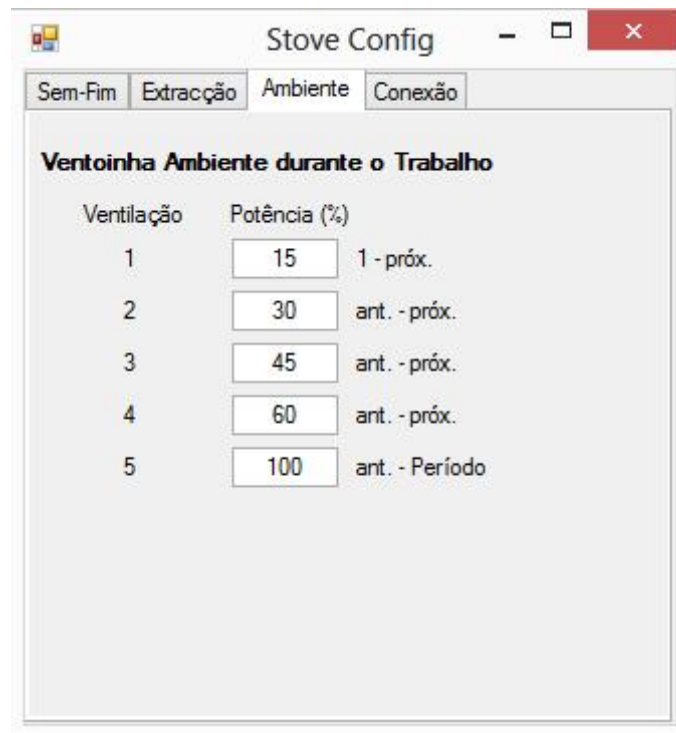


Figura 4.13 - Tab de parâmetros da VA.

Embora a aplicação criada seja apenas uma versão *alpha*, esta implementa já vários dos conceitos que foram discutidos com a Ecoama, como a interface “*clean*” e intuitiva, assim como a presença apenas dos parâmetros de maior importância. É no entanto possível permitir que outros sejam alterados através de um botão que lance uma janela com uma lista mais completa, não sendo necessário recorrer à mesma frequentemente.

4.2.2 – Rotina para Comunicação do Microcontrolador

Verificou-se ser necessário ajustar a *baudrate* utilizada na comunicação entre o microcontrolador e o computador, pois por vezes haviam mensagens corrompidas no caso de ser usado um valor *default* de 115200. Para este propósito, é utilizado um ciclo de mensagens por repetição para o ajuste, de modo a assegurar que não são perdidos dados. O computador vai enviar a *string* “*cbaud*” para o controlador, que verifica se recebeu de fato “*cbaud*”. Em caso afirmativo, vai aumentar uma

variável, *baudfound*. Após este passo, faz um *echo* para o computador da *string* que recebeu, que verifica se não existiram alterações à mesma. Se esta se mantiver, o processo de repetição continua, até que se atinjam 25 repetições. No caso do resultado de uma das transmissões não ser o esperado, o computador vai incrementar a *baudrate* que está a utilizar em 25 e reinicia o processo, ao passo que o PIC32 vai fazer um *reset* à variável *baudfound*. Assim, é necessário que se verifiquem 25 transmissões bidirecionais corretas para que a conexão seja reconhecida de ambos os lados.

Para efetuar a comunicação com o programa criado para propósitos de configuração do controlador é utilizada a rotina `UARTCheck()`, cujo fluxograma se encontra na Figura 4.14. A função *main* invoca esta rotina no caso de verificar que a *flag* utilizada pela RSI, `noRX`, se encontra a 0, sendo esta verificação feita a cada segundo. A rotina começa por verificar se existe uma ligação com o programa, através da leitura de uma variável, *Connected*, que é estabelecida quando o fabricante prime o botão “Conectar”. Tal é feito para permitir ao controlador ver se está ligado à aplicação, ou a um outro terminal diferente, podendo enviar dados diferentes neste caso (nomeadamente para efeitos de manutenção). Se a verificação for negativa, verifica se a mensagem recebida foi a *string* “Connected1”, respondendo com “Connected2” em caso afirmativo. Após receber esta mensagem, a aplicação de computador encontra-se preparada para enviar a configuração pretendida, e o controlador em estado de a receber.

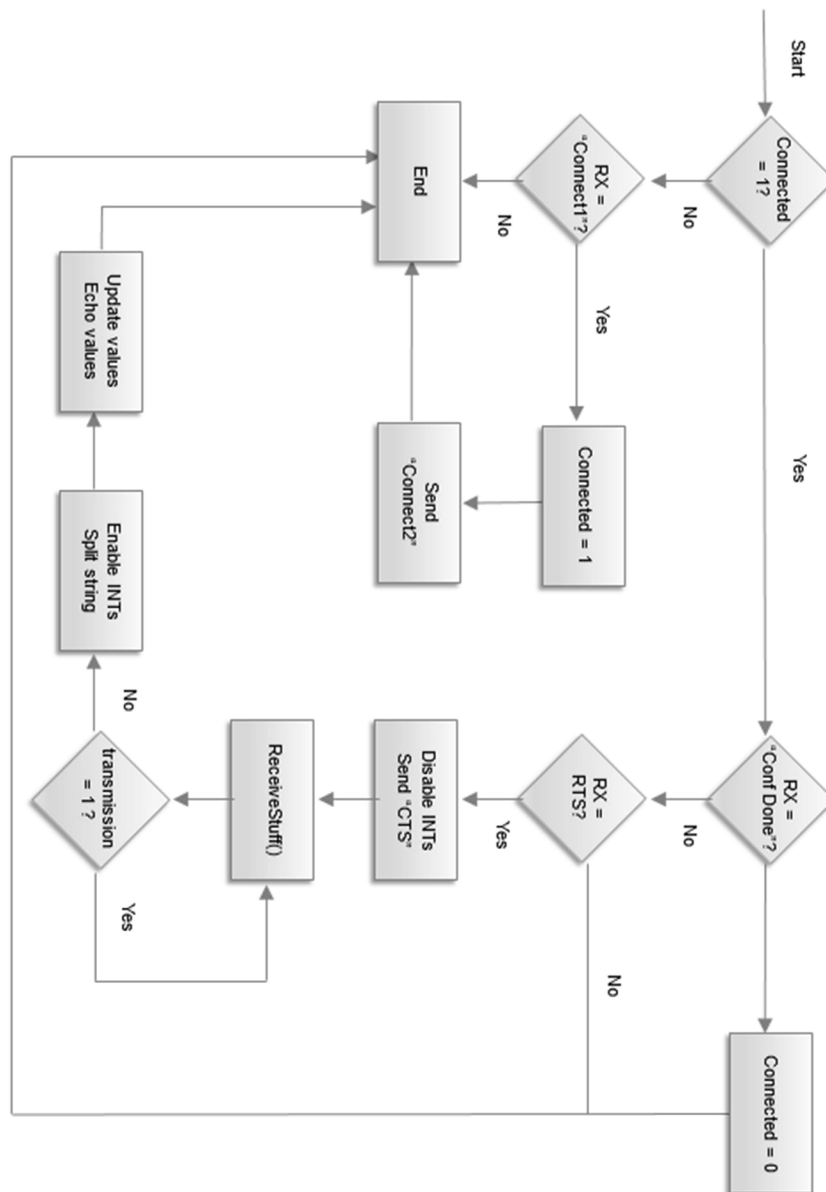


Figura 4.14 – Fluxograma da função UARTCheck().

No caso da conexão já se encontrar estabelecida, isto é, a variável *Connected* ter valor “1”, a próxima verificação é feita sobre a *string* recebida. Primeiro, se esta é “Conf Done”, o controlador interpreta esta mensagem como o sinal que o fabricante acabou de realizar a configuração, e coloca a *Connected* a “0”. Se a *string* for diferente é, de seguida, comparada com “RTS”. Em caso de igualdade, o controlador reconhece que o computador deseja carregar uma nova configuração

de parâmetros. Neste caso, o controlador começa por desligar as interrupções, enviando depois “CTS”, de modo a indicar que está pronto para começar a receber informação. É então invocada a rotina `ReceiveStuff()`, que é responsável por realizar a troca de informação com o computador, efectuada com um sistema de *handshake*, sendo cada parâmetro recebido confirmado através de uma mensagem “ACK#”, onde # é o índice da variável recebida. De seguida, as interrupções voltam a ser ligadas, o *buffer* em que os valores foram guardados é dividido (com recurso ao carácter ‘;’ como separador), os valores carregados para as respetivas variáveis e é feito um *echo* de tudo o que foi recebido. Por último, em caso da *string* recebida não corresponder a nenhuma das estabelecidas para a comunicação, esta é ignorada. Na Figura 4.15 encontra-se um diagrama ilustrativo da troca de mensagens entre o computador e o microcontrolador durante uma configuração executada com sucesso.

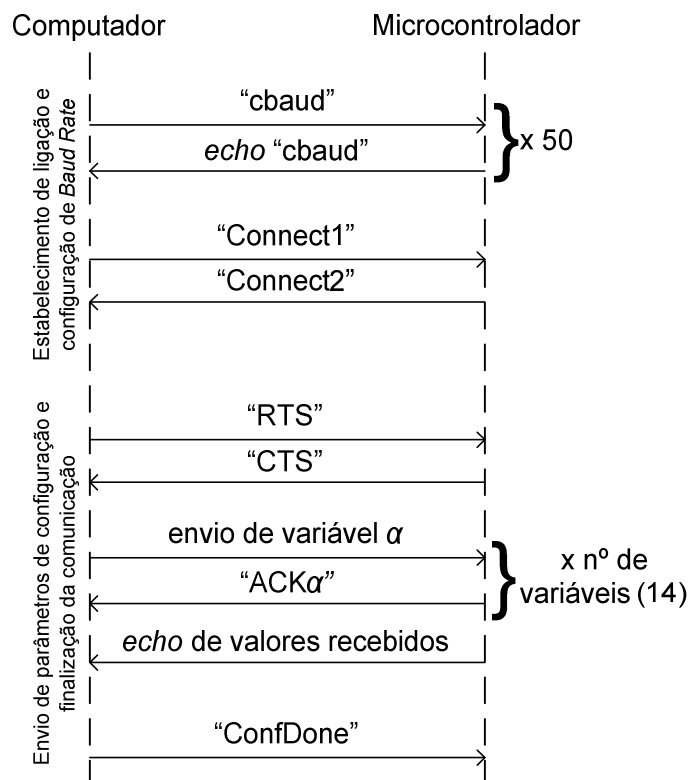


Figura 4.15 – Troca de mensagens entre o computador e o controlador.

V. Testes de Validação



De forma a validar o *hardware*, *firmware* e *software* projetados, foram realizados vários testes. Inicialmente, foi verificado o funcionamento de cada elemento do sistema, para depois ser analisado o comportamento deste como um todo.

5.1 – Validação de Circuitos e Device Drivers

Os primeiros testes realizados tinham como objetivo a validação dos circuitos projetados relativamente aos *device drivers*, tanto dos medidores de temperatura como para atuação sobre as ventoinhas, MA e RI.

5.1.1 – Sensores

Para a validação dos circuitos e *device drivers* criados para os três sensores de temperatura foram efetuadas várias medições ao longo de 10 dias, de modo a verificar a sua precisão durante um intervalo de tempo maior, ao invés de em apenas um dia. Na Tabela 5.1 estão presentes os valores registados para cada um dos sensores (LM335, sonda NTC e termopar do tipo K). Na mesma tabela, encontra-se o valor medido pelo sensor utilizado como referência, um termopar fabricado pela Fluke, modelo Fluke 51 II.

Tabela 5.1 – Temperaturas medidas (em °C).

Termopar Fluke	NTC	LM335	K-Type
22,1	22,3	23	23
22,3	22,2	23,2	23,2
24,3	24,6	24,9	24,9
24,5	24,6	25,6	25,6
24,5	24,8	25,3	25,3
24,8	24,8	25,4	25,4
25,4	25,4	26,1	26,1
26,7	26,6	27,7	27,7
27,8	27,6	28,3	28,3
28,9	28,7	29,4	29,4

De acordo com a Tabela 5.1 e os gráficos seguintes (Figuras 5.1 e 5.2), os valores registados pela sonda NTC têm uma boa precisão, sendo o erro máximo verificado

de 0,3 °C, o que pode ser considerado pouco significativo, uma vez que a temperatura desejada pelo utilizador tem uma variação mínima de 1 °C.

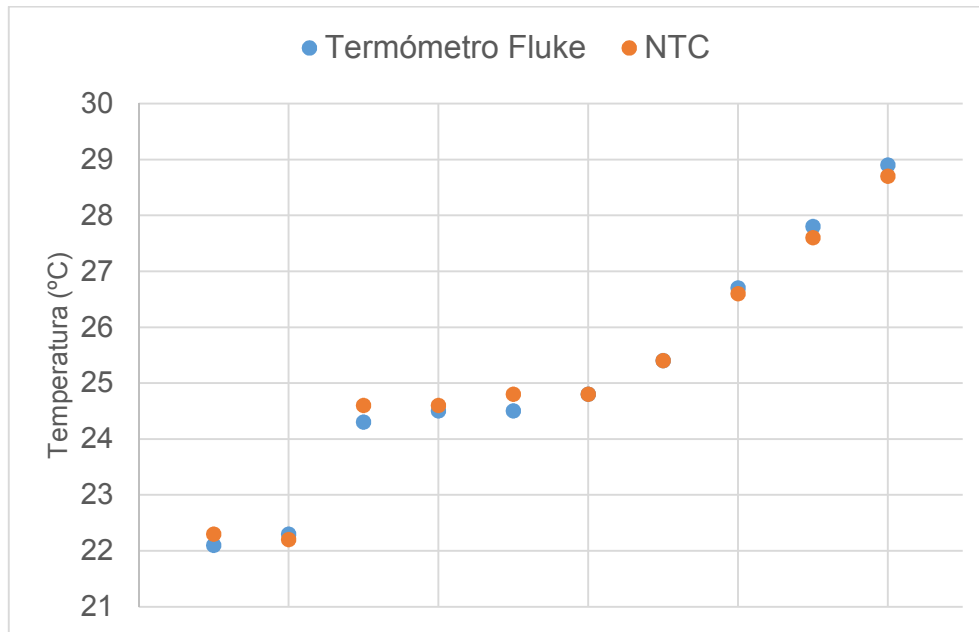


Figura 5.1 – Temperatura medida pela sonda NTC e Termopar Fluke.

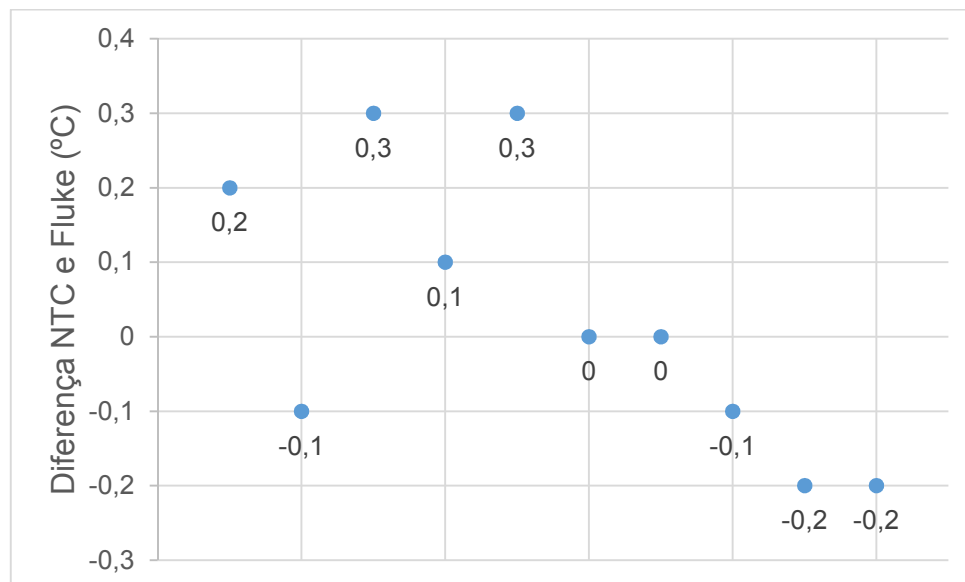


Figura 5.2 – Diferença de temperaturas medidas pela NTC e Termopar Fluke.

Nas Figuras abaixo (5.3 e 5.4), encontram-se as medições realizadas com o sensor LM335 por oposição ao termopar. Através do sensor LM335 obtiveram-se medidas superiores às do termopar Fluke. Tal deve-se ao facto de a temperatura da PCI aumentar em relação à temperatura ambiente, graças ao calor dissipado pelos



vários componentes. Embora os valores obtidos tenham sido medidos sem nenhum atuador em funcionamento, uma vez que tal levaria a algum aquecimento da placa, verificou-se que a PCI se encontrava com uma temperatura em média 0,76 °C acima da ambiente. A maior diferença entre valores verificada foi de 1,1 °C.

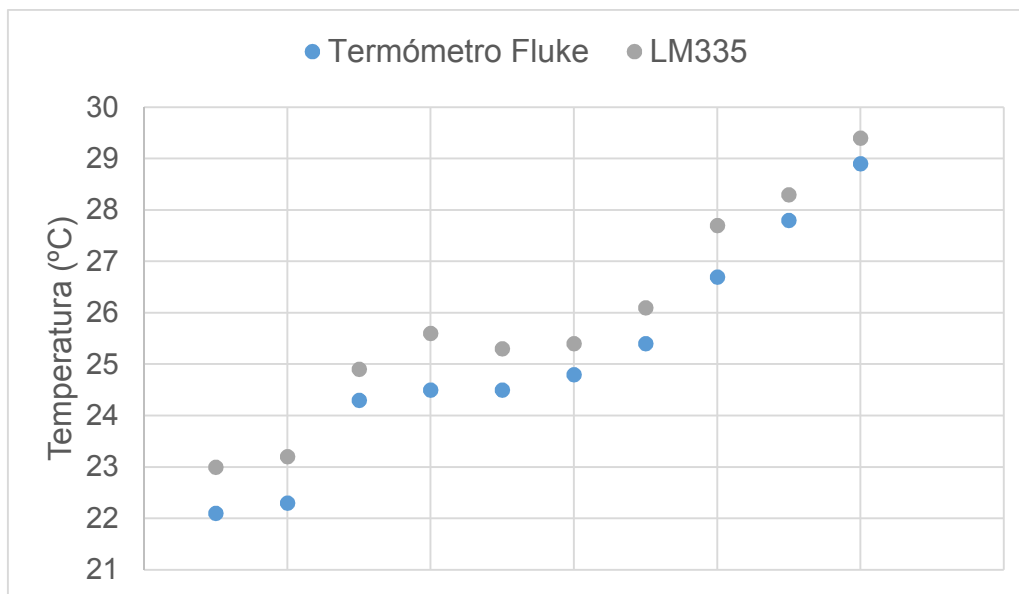


Figura 5.3 - Temperatura medida pelo sensor LM335 e Termopar Fluke.

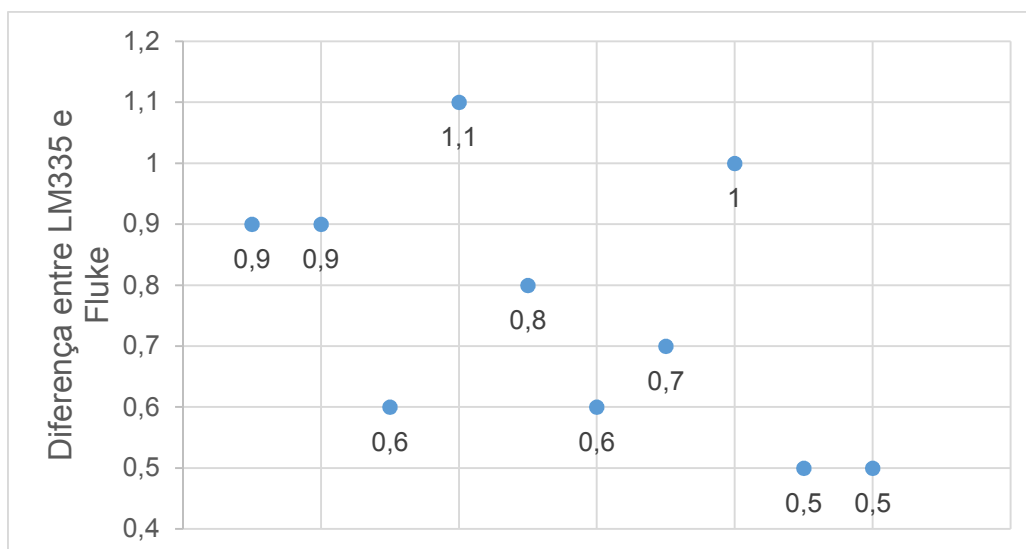


Figura 5.4 - Diferença de temperaturas medidas pelo LM335 e Termopar Fluke.

Como referido anteriormente, a temperatura da PCI é superior à ambiente. Deste modo, torna-se impossível realizar uma medição correta com o termopar do tipo K. Uma vez que a sua junção está a uma temperatura mais elevada que a ponta, a

diferença de potencial criada é negativa. No entanto, o sistema não se encontra preparado para lidar com valores de tensão negativos, o que leva a que o PIC32 entenda que a temperatura a que se encontra o termopar é sempre maior ou igual à temperatura da placa, assumindo um valor igual no caso de esta ser menor (como apresentado na Tabela 5.1). Deste modo, recorreu-se a uma torradeira para aquecer o termopar e o termómetro até uma temperatura mais elevada, para que fosse possível medir a precisão do primeiro. Estas medições encontram-se representadas na Tabela 5.2.

Tabela 5.2 – Temperaturas medidas (em °C).

Termopar Fluke	LM335	K-Type
151,3	24,6	149,8
154,2	26	152,1
156,8	26,5	154,3
157,1	25,6	155,9
158,6	24,7	156,2
159,7	24,1	157,7
160,2	25,3	157,6
161,3	27,2	157,3
163,5	25,4	159,8
166,4	25,8	162,5

Como se pode verificar pela análise da tabela, os valores obtidos com o termopar do tipo K são inferiores aos medidos com o termómetro Fluke, sendo a maior discrepância observada de 3,9 °C, e a média entre as 10 medidas realizadas de -2,59 °C. Embora este seja um erro considerável, de cerca de 1,6%, existe uma precisão satisfatória, uma vez que não é necessário que os valores sejam bastante precisos para efetuar o controlo do sistema. É ainda importante ressaltar que, devido à forma como o teste foi conduzido, a temperatura dentro da torradeira não era uniforme, gerando diferenças na temperatura a que se encontravam os sensores. No entanto, o problema foi simplificado assumindo que todo o interior da torradeira se encontrava à mesma temperatura aquando da realização do teste.



Nas Figuras 5.5 e 5.6 encontram-se os gráficos com as temperaturas medidas e a diferença entre valores dos dois sensores.

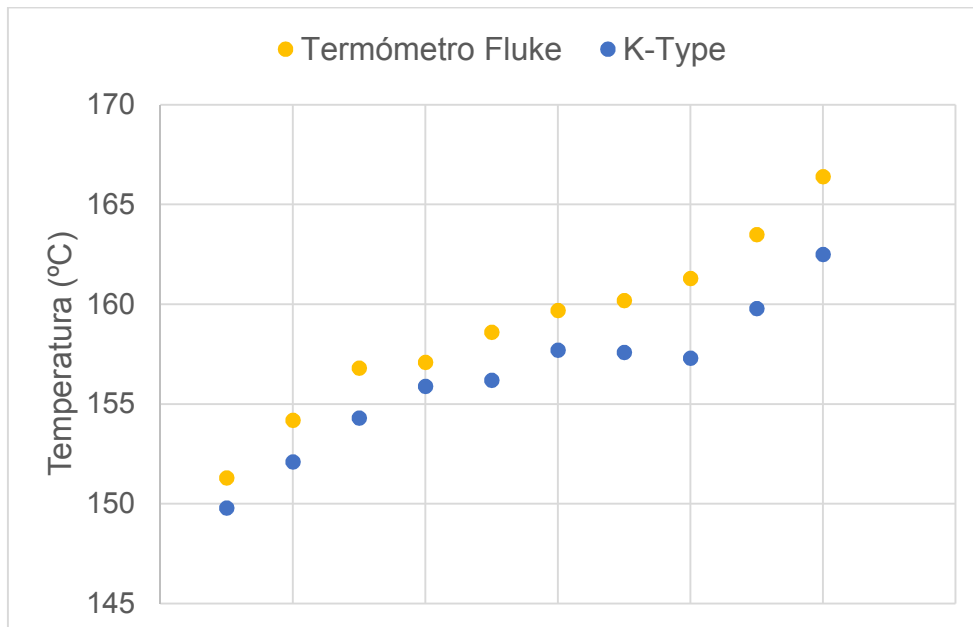


Figura 5.5 - Temperatura medida pelo Termopar e Termómetro Fluke.

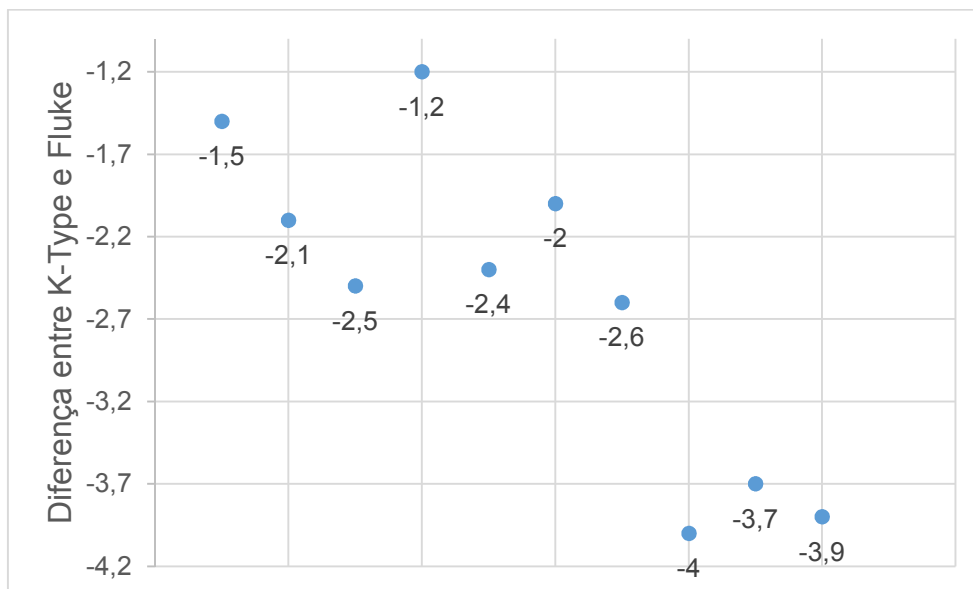


Figura 5.6 - Diferença de temperaturas entre Termopar e Termómetro Fluke.

5.1.2 – Atuadores

Os testes realizados para os atuadores procuraram validar os *device drivers* construídos para a interação com os mesmos. O primeiro atuador a ser testado foi a Resistência de Ignição, sendo este também o mais simples. O único teste realizado foi o ligar e desligar da resistência, verificando-se que o procedimento era realizado sem problemas, com aquecimento da RI.

O comportamento do Motor de Alimentação foi avaliado através de testes simples. Uma vez que este se encontrará a trabalhar com um *duty-cycle* de período considerável (igual ou superior a 10 segundos), optou-se por definir um período de 10 segundos, fazendo-se variar o *duty-cycle* de 0 a 100% em incrementos de 10%, correspondentes a 1 segundo. Assim, foram testados todos os níveis de poder de queima que podem ser definidos, uma vez que o T_{on} do motor tem uma variação mínima de 1 segundo.

Verificou-se com um cronómetro que o MA se comportava da forma esperada, respeitando os tempos definidos para ligar e desligar. Foi ainda verificado, com recurso a um osciloscópio, que os instantes em que o microcontrolador alterava o estado do pino correspondente ao motor eram os esperados (iguais aos instantes de ligar/desligar). Por último, foi também feito um teste de *stress* ao motor, colocando-o a trabalhar ininterruptamente durante um período mais longo, de 2 minutos, de modo a simular o seu funcionamento na fase *Fire Up*, uma vez que nesta este tem que assegurar que são entregues mais *pellets* à câmara de combustão. Também este teste foi aprovado com sucesso, sendo o comportamento do MA o esperado.

Para avaliar o funcionamento das ventoinhas foram realizados testes para os diferentes instantes de *trigger* destas, tendo já em conta os instantes ajustados de modo a permitir uma alteração da velocidade destas que não levasse ao bloqueamento das mesmas. Esta situação encontra-se descrita com maior detalhe no capítulo IV, secção 4.1.1.2.



As ventoinhas foram testadas primeiro separadamente e a potência entregue a estas variou entre 0 e 100%, com incrementos de 10%, tendo sido medidos os valores de tensão no ponto em que estas estão ligadas à PCI (Teste 1). De seguida, realizou-se o mesmo procedimento, mas com as ventoinhas a trabalhar em simultâneo (Teste 2). Em baixo, na Tabela 5.3, encontram-se os valores de tensão medidos durante os testes.

Tabela 5.3 – Valores de tensão medidos durante os testes.

Velocidade (%)	Teste 1		Teste 2	
	VA (Volt)	VE (Volt)	VA (Volt)	VE (Volt)
0	5,3	8,2	5,2	7,9
10	128,2	154,3	127,5	153,9
20	136,3	161,4	136,1	161,5
30	144,7	171,3	143,9	171,4
40	152,3	174,5	151,6	174,6
50	159,7	181,1	159,3	180,8
60	170,3	186,8	169,8	187,1
70	176,6	192,9	176,2	193,2
80	182,3	202,5	181,9	201,3
90	189,4	207,8	189,6	204,2
100	200,2	213,5	199,8	212,4

Como se pode observar na Tabela 5.3, a diferença de potencial diminuiu no segundo teste, o que pode ser justificado pelo fato de, ao se encontrarem ambas as ventoinhas em funcionamento, a potência entregue ser ligeiramente menor, uma vez que os circuitos de interface têm várias pistas em comum. No entanto, a diferença de valores entre os testes é muito pequena, podendo ser considerada irrelevante. Na página seguinte encontram-se dois gráficos (Figuras 5.7 e 5.8) representativos da tensão medida aos terminais das ventoinhas. Nestes gráficos não se encontram os resultados das medições efetuadas com a velocidade a 0%, de modo a proporcionar uma melhor visão da semi-linearidade que existe no caso das ventoinhas se encontrarem em funcionamento.

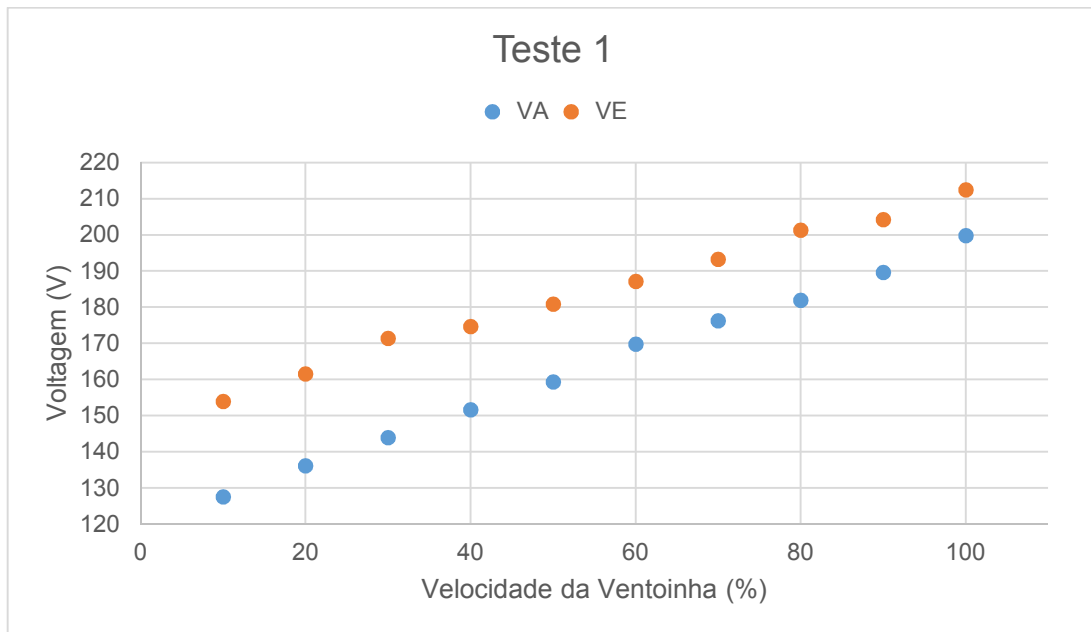


Figura 5.7 – Gráfico de Voltagem/Velocidade do Teste 1.

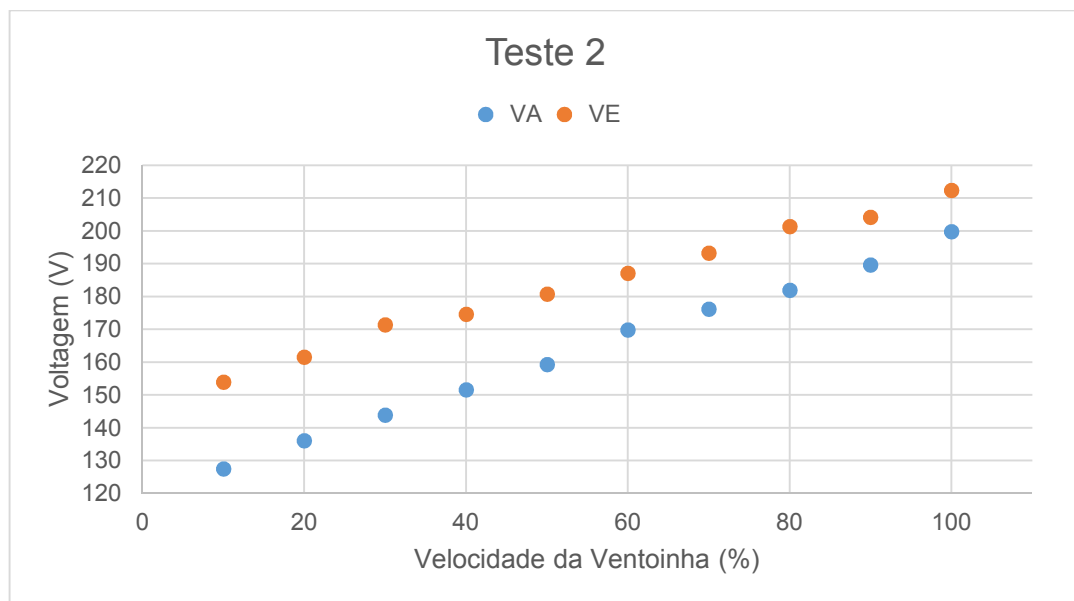


Figura 5.8 – Gráfico de Voltagem/Velocidade do Teste 2.

Por último, é importante referir que, por várias vezes, se colocaram todos os atuadores a operar na sua potência máxima, de modo a verificar se a PCI se encontra em condições de suportar esse tipo de uso (embora tal não se verifique durante funcionamento normal do aparelho), teste que também foi passado com sucesso.



5.2 – Teste ao Controlador

5.2.1 – Arranque Manual do Hefesto

Após o teste de cada elemento do sistema em contexto laboratorial, procedeu-se à preparação de um teste num cenário real. Ao invés da análise individual do funcionamento de cada peça, efectuou-se a análise do conjunto de todos os elementos, emulando as funções do controlador. Para este fim, optou-se por proceder ao arranque manual do sistema, isto é, controlar o sistema através do controlo de cada atuador e da monitorização das temperaturas registadas pelos vários sensores. Deste modo, o controlador é simulado, assumindo uma pessoa o papel que este teria no funcionamento do aparelho.

Este teste teve por finalidade a validação da máquina de estados projetada, bem como a verificação dos vários sensores e atuadores, quando ligados ao recuperador de calor Hefesto. Para a realização do teste, foi criado um programa que permitia fazer o controlo de cada atuador e visualizar as temperaturas dos sensores através do *Touchscreen* presente na placa PIC32-MAXI-WEB.

O teste foi realizado com sucesso nas instalações da Ecoama, onde se procedeu ao arranque do recuperador de calor, fazendo-o passar pelos 4 estados do sistema (Fire Up, Burning, Fire Down, Idle).

5.2.2 – Controlo Automático do Sistema

Após o arranque manual do sistema acima referido, foi necessário verificar o funcionamento do controlador de forma automática, sem que existisse necessidade de qualquer interação por parte do utilizador.

Foram realizados vários testes em laboratório de modo a avaliar a resposta do controlador a diversos estímulos, como por exemplo a deteção do limiar de temperatura dos gases de escape ao qual corresponde uma possível transição de

estado. Ainda no laboratório, foram feitas verificações de segurança, nomeadamente o aumento da temperatura dos gases de escape acima do limite estabelecido, verificando-se uma resposta correta do controlador conforme o previsto, ligando ambas as ventoinhas na máxima potência.

Findos os testes em laboratório, procedeu-se à verificação do funcionamento do sistema nas instalações da Ecoama, onde o controlador teve o desempenho esperado. O controlador colocou o recuperador de calor em trabalho, sendo que para tal apenas foi necessário dar ordem de início, premindo o botão ON. Durante o teste foi alterada a velocidade da VA pretendida, bem como o poder de queima desejado, alterações a que o controlador correspondeu conforme o esperado, aumentando ou diminuindo a velocidade da ventoinha ou o *duty cycle* do MA.

Foi ainda verificado o correto funcionamento do modo Maintaining. Para tal, foi definida uma temperatura pretendida menor que a temperatura ambiente a que se encontrava a divisão. Após a estabilização do sistema, com uma temperatura de gases de escape baixa, voltou a aumentar-se o valor da temperatura desejado, de modo a que o controlador levasse o aparelho para o estado de Burning de forma automática. Também este teste correu de acordo com o planeado, comprovando o adequado funcionamento do sistema. A imagem da página seguinte (Figura 5.9) foi tirada durante estes testes.



Figura 5.9 – Recuperador de calor em funcionamento.

VI. Conclusões



Foi construído um controlador capaz de controlar um recuperador de calor, tendo sido implementadas e testadas as funções necessárias para permitir o funcionamento do aparelho Hefesto. As interfaces para interação com os sensores e atuadores utilizados foram testadas e comprovou-se que se encontram a funcionar corretamente. Também a máquina de estados projetada se provou adequada, sendo esta fácil de adaptar a outros aparelhos.

Para além das funcionalidades já implementadas, o controlador encontra-se preparado para a adição de outras, já tendo por exemplo uma interface gráfica desenhada a pensar numa expansão das características apresentadas.

O programa criado para o computador, responsável pela configuração do controlador para os diferentes aparelhos, encontra-se preparado, necessitando contudo de ser alterado, de modo a ser compatível com as funcionalidades e aparelhos que ainda não foram implementados.

A Ecoama considerou que o trabalho realizado estava de acordo com pretendido, tendo demonstrado interesse na continuidade do projeto aquando da realização dos últimos testes.

6.1 – Trabalho Futuro

O trabalho realizado durante esta dissertação conduziu à criação de um controlador funcional, capaz de controlar todos os processos inerentes a um recuperador de calor, contudo existe ainda bastante trabalho a ser desenvolvido até se chegar a um produto final, pronto para produção.

Alguns dos pontos a serem considerados são:

- Preparação do controlador ao nível de *hardware* e *firmware*, assim como do programa criado, para operação com outros aparelhos (caldeiras e queimadores, recuperadores de calor com mais elementos);

- Criação de diferentes interfaces para o utilizador, como por exemplo uma aplicação para *smartphones* ou uma interface disponível na Internet;
- Adição de outras funcionalidades ao sistema, como programação de horário e escolha do tipo de combustível;
- Criação de uma nova PCI, que incorpore o que de momento se encontra dividido entre a PCI criada e a placa PIC32-MAXI-WEB, de modo a reduzir os custos do controlador, assim como compactar o espaço ocupado pelo mesmo;
- Realização de um conjunto extenso de testes, de modo a validar o funcionamento do controlador sob diferentes condições, assim como testar a reação do sistema a situações inesperadas (por exemplo ser detetada uma falha num atuador ou sensor), o que será fundamental para obter as certificações necessárias para que o projeto se possa transformar num produto.

O trabalho realizado no decorrer desta dissertação constituiu uma base sólida para o desenvolvimento de projetos futuros, fornecendo as ferramentas necessárias para a transição do protótipo criado para um produto comercial.



Bibliografia



1. Trerice. What is an Electronic Controller ? [Internet]. 2001. p. 43–4. Available from: http://www.cpinc.com/Trerice/Control/control_43_44.pdf
2. Europe Biomass Association. A Pellet Road Map for Europe [Internet]. 2008. Available from: http://www.aebiom.org/wp/wp-content/uploads/file/Publications/BrochurePRME_LR.pdf
3. Ciolkosz D, Wilson TH, Wilson TO. Manufacturing Fuel Pellets from Biomass [Internet]. 2009. p. 1–4. Available from: <http://pubs.cas.psu.edu/FreePubs/pdfs/uc203.pdf>
4. Wood Pellets 2U. Fuel Price Comparison [Internet]. Available from: <http://www.woodpellets2u.co.uk/fuel-price-comparison-0>
5. Trømborg E, Ranta T, Schweinle J, Solberg B, Skjevraak G, Tiffany DG. Economic sustainability for wood pellets production – A comparative study between Finland, Germany, Norway, Sweden and the US [Internet]. Biomass and Bioenergy. 2013 [cited 2013 May 28]. p. 1–10. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0961953413000457>
6. Mäkelä M, Lintunen J, Kangas H-L, Uusivuori J. Pellet promotion in the Finnish sawmilling industry: The cost-effectiveness of different policy instruments [Internet]. J. For. Econ. 2011 [cited 2013 Jul 16]. p. 185–96. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S1104689911000067>
7. Peksa-Blanchard M, Dolzan P, Grassi A, Heinimö J, Junginger M, Ranta T, et al. Global Wood Pellets Markets and Industry : Policy Drivers , Market Status and Raw Material Potential [Internet]. IEA Bioenergy. 2007. Available from: <http://www.bioenergytrade.org/downloads/ieatask40pelletandrawmaterialstudydynov2007final.pdf>
8. Fiedler F. The state of the art of small-scale pellet-based heating systems and relevant regulations in Sweden, Austria and Germany [Internet]. Renew. Sustain. Energy Rev. 2004 [cited 2012 Nov 5]. p. 201–21. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S1364032103001242>
9. Maxwell S. Turn Waste Wood Into Home Heat [Internet]. Mother Earth News. 2009. Available from: <http://connection.ebscohost.com/c/articles/36237130/turn-waste-wood-home-heat>
10. Paju P, Alakangas E. Wood pellets in Finland - technology , economy , and market [Internet]. Available from: http://www.ateap.cz/new/Pelety_Finsko.pdf

11. Monteiro E, Mantha V, Rouboa A. Portuguese pellets market: Analysis of the production and utilization constrains. *Energy Policy* [Internet]. Elsevier; 2012 Mar [cited 2013 Jun 6];42:129–35. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0301421511009505>
12. Alakangas E. New Europeans Pellet Standard - EN14961-1 [Internet]. *Conf. Eur. Pellet*. 2010. p. 3–4. Available from: http://www.biointernational.ru/sites/default/files/Alakangas_Pellet_standard_EN14961-1.pdf
13. European Pellet Council. Handbook for the Certification of Wood Pellets for Heating Purposes [Internet]. 2013. Available from: http://cogent.controlunion.com/cusi_production_files/SISI_files/FL_111711101308_ENplus-handbook-3_5_11.pdf
14. Dias J, Costa M, Azevedo JLT. Test of a small domestic boiler using different pellets [Internet]. *Biomass and Bioenergy*. 2004 [cited 2013 Jun 6]. p. 531–9. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0961953404000947>
15. Cocchi M, Nikolaisen L, Junginger M, Goh CS, Hess R, Jacobson J, et al. Global Wood Pellet Industry - Market and Trade Study [Internet]. Available from: http://www.bioenergytrade.org/downloads/t40-global-wood-pellet-market-study_final.pdf
16. Fernandes U, Costa M. Potential of biomass residues for energy production and utilization in a region of Portugal [Internet]. *Biomass and Bioenergy*. Elsevier Ltd; 2010 [cited 2013 Jun 16]. p. 661–6. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0961953410000103>
17. Ecoama. Catálogo Ecoama [Internet]. Available from: <http://www.ecoama.pt/index.html>
18. Quadrafire. Quadrafire Pellet Stove Brochure [Internet]. Available from: <http://www.quadrafire.com/Browse/Stoves/Pellet-Stoves.aspx#>
19. Regency. Regency Pellet Brochure [Internet]. Available from: <http://www.regency-fire.com/Files/Brochures/Regency-pellet-brochure-low.aspx>
20. Summer's Heat. Summer's Heat Webpage [Internet]. Available from: http://www.englishstoves.com/sh_pellet_stoves.html
21. ATech Electronics. Fumis ALPHA Wood biomass combustion controller User Guide. 2012.



22. Fumis. Fumis LevelTronic INFO Sheet [Internet]. Available from: http://www.fumis.si/catalog-documents/INFO_Sheet_Fumis_LevelTronic_eng_WEB.pdf
23. Fumis. Fumis FlowTronic INFO Sheet [Internet]. Available from: http://www.fumis.si/catalog-documents/INFO_Sheet_Fumis_LevelTronic_eng_WEB.pdf
24. Fletcher G. Introduction to biomass boilers , hydraulic design , & accumulators [Internet]. 2012. Available from: <http://www.cibseyorkshire.org/Files/YEN-Biomass-seminar-Feb-12.aspx>
25. Priyanonda J, Milward R. Electric Tankless Water Heating : Competitive Assessment [Internet]. Available from: <http://www.energystar.gov/products/specs/sites/products/files/ElectricTanklessCompetitiveAssessment.pdf>
26. Morgan J. Tank-type vs. Tankless Water Heaters [Internet]. 2009. Available from: <http://buildingengineer.wordpress.com/2009/10/08/tank-type-vs-tankless-water-heaters/>
27. Fiedler F, Bales C, Persson T. Design method for solar heating systems in combination with pellet boilers / stoves [Internet]. Glasgow: Eurosun Conference; 2006. Available from: <http://du.diva-portal.org/smash/get/diva2:521615/FULLTEXT01>
28. Comfort Solutions. Boiler Wood Pellet Stoves [Internet]. Available from: <http://www.comfortsolutions.ie/heating-systems/wood-pellet-stoves/back-boiler-stoves.html>
29. Kordas Heating. Industrial Pellet Burners [Internet]. Available from: <http://www.kordas-heating.com/industrial-pellet-burners/>
30. Termo Cabi. Wood pellet burners - SPL1000 [Internet]. p. 12–3. Available from: http://www.termocabi.it/wp-content/uploads/2012/09/schedatecnica_spl1000qc_ita-eng_ed03-12.pdf
31. Termomont. Pellet Burner TERMEC in Operation [Internet]. Available from: <http://www.termomont.co.rs/2011/wood-pellet-burner-termec-in-operation/?lang=en>
32. Kedel. Kedel Standart Boilers [Internet]. Available from: <http://kedelboilers.com/meet-kedel/kedel-pellet-boiler-standard-options/>
33. Quadrafire. Why Quadra-Fire: Key Technologies [Internet]. Available from: <http://www.quadrafire.com/Why-Quadra-Fire/Key-Technologies.aspx>

34. Zoppas Industries. Electronic Control For Pellet Stoves [Internet]. Available from:
http://www.ricaheatingelements.com/c/document_library/get_file?uuid=41491985-9033-42f2-bce0-51703183ccc6&groupId=13623
35. Naturela Electronics. Electronic Pellet Burner Controller NPBC-V3-1 [Internet]. p. 1–22. Available from: http://www.naturela-bg.com/files/NPBC-V3_rev1_2_EN.pdf
36. ATech Electronics. Fumis OXY 120 Information Sheet [Internet]. Available from: http://www.fumis.si/catalog-documents/INFO_Sheet_Fumis_OXY_120_eng_WEB.pdf
37. Kordas Heating. Puma PID Controller [Internet]. Available from:
<http://www.kordas-heating.com/controller-puma-pid/>
38. ATech Electronics. Fumis ALPHA Biomass Combustion Control [Internet]. Available from: <http://www.fumis.si/en/about-fumis/fumis-combustion-controllers/fumis-alpha>
39. AACO. Centrifugal Datasheet [Internet]. Available from:
http://www.aaco.it/j/materiale/prodotti/ventilatori/AACO_centrifugal.pdf
40. AACO. Fume Extractors for Pellet Stoves [Internet]. Available from:
http://aaco.altervista.org/files/aaco_aspirazione_aria_calda.pdf
41. Rauschert. HTH ZOnder Datasheet.
42. Rauschert. HTH Rohrheizer 79/11/200 Datasheet.
43. BCZ. BCZ Model 509 Datasheet [Internet]. Available from:
http://www.tornavento.it/bcz/_pdf_product/_4_.pdf
44. Sacet-Probes. NTC - NS N E 1.5 5/6 E 20 N S E Datasheet.
45. Svab A. Thermocouples, Thermistors & RTDs [Internet]. 2009. Available from:
http://www.sensorsci.com/index.php?option=com_content&view=article&id=24%3Athermocouplesthermistorsrtds&catid=13%3Apressreleases&showall=1
46. Enercorp. Which Sensor? Thermistor , RTD or Thermocouple [Internet]. p. 1–4. Available from:
http://www.enercorp.com/temp/Thermistors_comparision.html



47. National Instruments. RTD, Thermistor, Thermocouple Comparison Chart [Internet]. Available from: <http://digital.ni.com/public.nsf/allkb/C50FA55B3B2F85D9862572D00083350E>
48. Heraeus Materials Technology. Comp. NTC, KTY, PRTD [Internet]. 2013. Available from: http://heraeus-sensor-technology.com/en/technology/vergleichstabelle/vergleichstabelle_1.aspx
49. Sacet-Probes. Thermocouple - C K A 3 4 V 10 I N I S/M Datasheet.
50. Picotech. Thermocouple Application Note [Internet]. Available from: <http://www.picotech.com/applications/thermocouple.html>
51. Microchip. PIC32MX5XX/6XX/7XX Family Data Sheet [Internet]. 2010. Available from: PIC32MX5XX/6XX/7XX Family Data Sheet
52. Texas Instruments. Moc3020 thru moc3023 optocouplers/optoisolators [Internet]. 1998. p. 1–7. Available from: <http://www.ti.com/lit/ds/symlink/moc3020.pdf>
53. Severns R. Design of Snubbers for Power Circuits [Internet]. Available from: <http://www.cde.com/tech/design.pdf>
54. Cornell D. Designing RC Snubber Networks [Internet]. Available from: <http://www.cde.com/catalogs/igbtAPPguide.pdf>
55. Robbins WP. Snubber Circuits [Internet]. 2009. p. 1–25. Available from: <http://aboutme.samexent.com/classes/spring09/ee5741/Snubbers.pdf>
56. Chowdhury N. Snubber Circuits [Internet]. p. 1–8. Available from: http://www.engr.usask.ca/classes/EE/443/notes/Snubber_Circuits.pdf
57. Philips. BT139 series Datasheet [Internet]. 2004. p. 1–12. Available from: http://www.nxp.com/documents/data_sheet/BT139_SERIES.pdf
58. IBS Electronics. Metal Oxide Varistors [Internet]. p. 1–16. Available from: http://www.ibselectronics.com/ibs/cmpnts/rgaco/catalog/D/MetalOVD2_16-opt.pdf
59. Vishay Bcomponents. Varistors Introduction [Internet]. 2003. Available from: <http://www.vishay.com/docs/29079/varintro.pdf>
60. Murata Manufacturing. Circuit Protection from Transient Voltage by Varistor [Internet]. p. 31–2. Available from: <http://www.murata.com/products/emc/knowhow/pdf/31to32.pdf>

61. Littelfuse. AN 1003 - Phase Control Using Thyristors [Internet]. 2002. p. 1–10. Available from:
http://www.littelfuse.com/~/media/Electronics/Application_Notes/Switching_Thyristors/Littelfuse_Thyristor_Phase_Control_Using_Thyristors_Application_Note.pdf
62. Philips Semiconductors. Chapter 6 - Power Control with Thyristors and Triacs [Internet]. Available from:
http://www.nxp.com/documents/application_note/APPCHP6.pdf
63. Teccor Electronics. AN 1007 - Thyristors Used as AC Static Switches and Relays [Internet]. Available from:
http://www.littelfuse.com/data/en/Application_Notes/an1007.pdf
64. Elliot R. Zero Crossing Detector and Comparators [Internet]. Available from:
<http://sound.westhost.com/appnotes/an005.htm>
65. Pyromation. Type K Thermocouple - thermoelectric voltage as a function of temperature (°C) [Internet]. p. 30–3. Available from:
http://www.pyromation.com/downloads/data/emfk_c.pdf
66. Analog Devices. AD 623 Datasheet [Internet]. 2008. Available from:
http://www.analog.com/static/imported-files/data_sheets/AD623.pdf
67. Analog Devices. Precision Thermocouple Amplifiers with Cold Junction Compensation - AD8494-8497 Datasheet [Internet]. 2011. Available from:
http://www.analog.com/static/imported-files/data_sheets/AD8494_8495_8496_8497.pdf
68. Duff M, Towey J. Two Ways to Measure Temperature Using Thermocouples Feature Simplicity , Accuracy , and Flexibility [Internet]. 2010. p. 1–6. Available from: Two Ways to Measure Temperature Using Thermocouples Feature Simplicity , Accuracy , and Flexibility
69. Texas Instruments. Precision Temperature Sensors - LM135 / LM235 / LM335 , LM135A / LM235A / LM335A Datasheet [Internet]. 2008. Available from: <http://www.ti.com/lit/ds/symlink/lm135.pdf>
70. Redfish Sensors. Resistance-Temperature Characteristics [Internet]. Available from: <http://www.redfishsensors.com/PDFs/R805-103F-3435-C.pdf>
71. Azevedo JL. DETPIC32, V2. 2011.
72. Olimex. PIC32-MAXI-WEB development board Users Manual [Internet]. 2011. Available from:
<https://www.olimex.com/Products/PIC/Development/PIC32-MAXI-WEB/resources/PIC32-MAXI-WEB.pdf>



73. Anton O, Gelineau B, Sauget J. Firmware and bootloader [Internet]. 2012. Available from: <http://rose.eu.org/2012/wp-content/uploads/2012/03/Firmwares-and-bootloaders.pdf>
74. Industries M. Type K Thermocouple Calibration [Internet]. Available from: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/temperature-measurement/thermocouple/type-k-calibration-table>
75. VirtualFab. VGDD:Visual Graphics Display Designer [Internet]. Available from: http://virtualfab.it/mediawiki/index.php/VGDD:Visual_Graphics_Display_Designer