



**Universidade de  
Aveiro  
Ano 2015**

Departamento de Mecânica

**PAULO JORGE DA  
COSTA MARTINS**

**UMA SOLUÇÃO DE DOMÓTICA PARA O  
CONTROLO DA TEMPERATURA**



Universidade de Departamento de Mecânica  
Aveiro  
Ano 2015

# **PAULO JORGE DA COSTA UMA SOLUÇÃO DE DOMÓTICA PARA O MARTINS CONTROLO DA TEMPERATURA**

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob a orientação científica do Doutor José Santos, Professor Auxiliar do Departamento de Mecânica da Universidade de Aveiro



## **O Júri / The Jury**

Presidente / President

**Prof. Doutor Jorge Augusto Fernandes Ferreira**  
Professor Auxiliar da Universidade de Aveiro

Vogais / Committee

**Prof. Doutor José Paulo Oliveira Santos**  
Professor Auxiliar da Universidade de Aveiro (Orientador)

**Prof. Doutor José Luís Costa Pinto de Azevedo**  
Professor Auxiliar da Universidade de Aveiro (Arguente)



## **Agradecimentos**

Agradeço ao orientador José Paulo Santos pela sua disponibilidade e ensino ao longo de todos estes meses de trabalho.

Nesta fase final de mais uma etapa na minha vida queria agradecer o apoio incondicional da minha família, por me ajudar a ultrapassar as fases mais difíceis de toda a minha vida académica.

Agradecer também aos meus amigos por me dar forças para lutar cada vez mais e pelo conhecimento transmitido pelos mesmos.

**Palavras-chave**

Domótica; Arduino; Controlo de temperatura; Algoritmo de controlo; *Smartphone*;

**Resumo**

A domótica ajuda a melhorar a qualidade de vida dos utilizadores, melhorando o conforto, segurança e mais recentemente eficiência energética. Este trabalho tem como objetivo desenvolver uma solução de otimização do controlo da temperatura.

Numa primeira fase será desenvolvido um circuito de leitura da temperatura, para aquisição da temperatura real, e um circuito controlo de potência para efetuar o controlo da temperatura do reservatório de água. Para conseguir controlo eficiente recorreu-se a algoritmos de controlo: ON/OFF, PID e lógica difusa utilizando uma atuação com PWM com objetivo de manter uma resposta estável no *setpoint* predefinido. Este controlo será monitorizado através do *smartphone* e do computador que se encontram conectados com o sistema através de Bluetooth, sendo possível observar a temperatura e definir o *setpoint*.

**Keywords**

Domotic;Arduino; Temperature control; Control algorithm; Smartphone;

**Abstract**

Home automation helps improve the quality of the users, improving comfort, security and energetic efficiency. The main goal of this project is to develop an optimization solution of temperature control. Initially will be develop a temperature reading circuit for acquiring the real temperature and a power control circuit to perform the control of the temperature on the water tank. To achieve an efficient control was used control algorithms: ON/OFF, PID and fuzzy logic operating in the output with a PWM in order to maintain a stable response in a default setpoint. This control will be monitored vi smartphone and computer connected to the system via Bluetooth that records the temperature and temperature setpoint are set.



# Índice

1. Introdução.....	1
1.1. Contexto.....	1
1.2. Problema.....	2
1.3. Objetivos .....	3
1.4. Organização do documento .....	3
2. Estado da arte .....	5
2.1. Soluções comerciais .....	5
2.1.1. LK15 - Pack Iniciação X10.....	5
2.1.2. Tecnologia KNX.....	6
2.1.3. Clipsal C-Bus .....	8
2.2. Soluções académicas .....	10
2.2.1. Domótica Inteligente – Um Contributo Prático [1].....	10
2.2.2. Edifícios Inteligentes – Soluções para gestão de climatização em instalação de Domótica KNX [15].....	10
2.2.3. Desenvolvimento de Infraestrutura de Comando Multifunções EIB- KNX para <i>Smartphone</i> [16] .....	11
3. Tecnologias de suporte.....	13
3.1. Sensores de temperatura.....	13
3.1.1. RTD .....	14
3.1.2. Circuitos integrados .....	14
3.1.3. Termopares .....	15
3.2. Comunicação de dados.....	15
3.2.1. Radiofrequência.....	16
3.2.2. Modbus.....	16
3.3. Controlo da potência .....	19

3.3.1.	Dispositivos de controlo de potência .....	19
3.3.2.	PWM.....	22
3.4.	Algoritmos de controlo.....	22
3.4.1.	Controlo ON / OFF.....	23
3.4.2.	PID.....	23
3.4.3.	Lógica difusa.....	26
4.	Arquitetura da solução proposta .....	29
5.	Implementação da solução proposta .....	31
5.1.	<i>Hardware</i> desenvolvido.....	32
5.1.1.	RTD e acondicionamento .....	32
5.1.2.	Módulo de potência .....	34
5.1.3.	Bluetooth ZS-040.....	35
5.1.4.	LCD .....	37
5.2.	<i>Software</i> desenvolvido .....	37
5.2.1.	MOVICON .....	37
5.3.	Análise de resultados.....	38
5.3.1.	Controlador ON/OFF .....	38
5.3.2.	Controlador PID.....	40
5.3.3.	Controlador lógica difusa .....	41
5.4.	Comparação dos algoritmos .....	44
6.	Conclusão.....	47
7.	Trabalho Futuro .....	48
	Referências .....	49
	Anexo 1 - Circuito Elétrico .....	54
	Anexo 2 - LCD.....	56
	Anexo 3 - Bluetooth.....	58



# Lista de Figuras

Figura 1.1 Origem da palavra Domótica[1].....	2
Figura 2.1 Pack de iniciação X10.....	6
Figura 2.2 Topologia da rede KNX[8].....	7
Figura 2.3 Dispositivo de climatização KNX/ABB[11].....	8
Figura 2.4 Exemplo de configuração em estrela[13].....	9
Figura 2.5 Termostato C-BUS (adaptado de [14]).....	9
Figura 2.6 Tela inicial da Fuzzy Logical Toolbox[1].....	10
Figura 2.7 Arquitetura da Solução[15].....	11
Figura 2.8 Representação das componentes da infraestrutura dos comandos[16] .....	12
Figura 3.1 Comparação de Sensores de Temperatura[17].....	13
Figura 3.2 Sensor de Temperatura PT1000[19].....	14
Figura 3.3 Sensor Integrado LM35[21].....	15
Figura 3.4 Estrutura de Camadas do Protocolo Modbus[24].....	17
Figura 3.5 Estrutura da mensagem Modbus[24].....	17
Figura 3.6 Códigos do campo de função[24].....	18
Figura 3.7 Símbolo de um TRIAC[28].....	19
Figura 3.8 Controlo usando o TRIAC.....	20
Figura 3.9 Controlo usando IGBT.....	21
Figura 3.10 Controlo usando um MOSFET.....	21
Figura 3.11 Resposta PWM controlada pelo Duty Cycle adaptado de [30].....	22
Figura 3.12 Exemplo de uma saída de controlo On/Off[31].....	23
Figura 3.13 Métodos de defuzzificação[38].....	28
Figura 4.1 Diagrama de blocos da solução proposta.....	30
Figura 5.1 Diagrama de blocos especificado.....	32
Figura 5.2 Fonte de Corrente.....	33
Figura 5.3 Circuito de Potência.....	35
Figura 5.4 Aplicação Android Blueterm.....	36
Figura 5.5 Interface Movicon.....	38

Figura 5.6 Fluxograma do controlo ON/OFF .....	39
Figura 5.7 Controlador On / Off .....	40
Figura 5.8 Processo PID .....	40
Figura 5.9 Controlador PID.....	41
Figura 5.10 Diagrama de blocos da Lógica Difusa.....	42
Figura 5.11 Graus de associação.....	42
Figura 5.12 Grau de Defuzzificação .....	43
Figura 5.13 Controlador de Lógica Difusa.....	44
Figura 5.14 Gráfico de Comparação dos Algoritmos.....	45
Figura 38 Circuito de Leitura .....	55
Figura 39 Ligações LCD.....	57
Figura 40 Ligações Bluetooth.....	59



# Lista de Tabelas

Tabela 3.1 Comparação das Tecnologias de Comunicação Editado[24] .....	16
Tabela 3.2 Efeitos nas alterações de cada ganho[35].....	25
Tabela 3.3 Cálculos Para K, Ti, Td[36].....	26
Tabela 3.4 Operadores Difusos[27].....	27
Tabela 3.5 Métodos de Defuzzificação[27] .....	28



# Acrónimos

EIB	<i>European Installation Bus</i>
EHS	<i>European Home System</i>
UTP	<i>Unshielded Twisted Pair</i>
KNX	Konnex
PLC	<i>Power Line Communication</i>
RTD	<i>Resistance Temperature Detectors</i>
Pt1000	<i>Platinum resistance 1000Ω</i>
PWM	<i>Pulse Wide Modulation</i>
PID	<i>Proportional Integral Derivative</i>
K	Ganho Proporcional
$T_i$	Tempo Integrativo
$T_d$	Tempo Derivativo
CI	Circuito Integrado
ADC	<i>Analog Digital Converter</i>
RF	Radio Frequência
OSI	<i>Open Systems Interconnection</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
CRC	<i>Cyclic Redundancy Check</i>
LRC	<i>Longitudinal Redundancy Check</i>
RTU	<i>Remote Terminal Unit</i>
CR	<i>Carriage Return</i>
LF	<i>Line Feed</i>
$K_p$	Ganho Proporcional
$K_i$	Ganho Integral
$K_d$	Ganho Derivativo
$K_u$	<i>Ultimate Gain</i>

$P_u$	<i>Ultimate Period</i>
TRIAC	<i>Triode for Alternating Current</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i>
SCR	<i>Silicon Controlled Rectifier</i>



# Capítulo 1

## 1. Introdução

### 1.1. Contexto

A domótica provém da junção de duas palavras, do latim *Domus* (casa) e de robótica Figura 1.1. A domótica surgiu com o objetivo de melhorar a vida diária das pessoas:

- Conforto
- Segurança
- Conservação de Energia
- Eficiência Energética
- Gestão Centralizada ou Parcial

Contudo não tem sido muito implementada devido à complexidade destas tecnologias, a dificuldade de integração na habitação e a compreensão do utilizador sobre o produto. Embora apresentando grandes mais-valias em termos de controlo de temperatura e iluminação, segurança e poupança de energia estas tecnologias são dispendiosas.

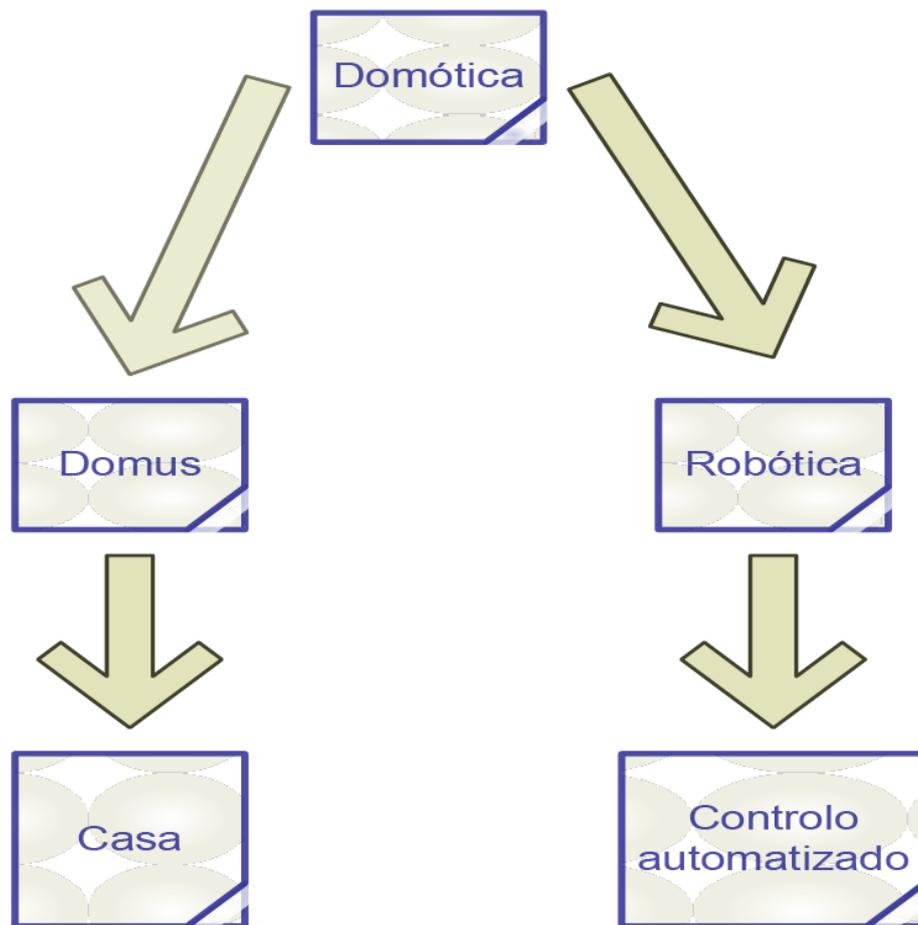


Figura 1.1 Origem da palavra Domótica[1]

## 1.2. Problema

O ser-humano desde sempre procurou uma melhor qualidade de vida, nomeadamente nas habitações. Mais recentemente tem procurado aumentar a eficiência energética das mesmas recorrendo a novas tecnologias. Especialmente devido ao aumento dos custos da energia. Mesmo recorrendo a novas tecnologias a falta de algoritmos inteligentes que permitam uma maior eficiência energética dificulta a rentabilização dos investimentos necessários. No controlo de temperatura verifica-se a falta de otimização, nomeadamente nos dispositivos de aquecimento.

### 1.3. Objetivos

Com a solução proposta neste trabalho pretende-se uma melhor gestão dos consumos energéticos melhorando o conforto e a eficiência energética recorrendo ao controlo da temperatura da habitação e respetivos dispositivos elétricos, propondo algoritmos que permitam otimizar essa gestão.

Na prática o objetivo é criar um sistema capaz de controlar os equipamentos elétricos, passando pela monitorização da temperatura real e a escolha do *setpoint* e recorrendo a algoritmos de controlo, nomeadamente On/Off, PID e logica difusa, de forma a obter um controlo mais eficiente da temperatura, sem *overshoot* e com uma resposta estável no *setpoint*, ou seja sem oscilações. Pois o aquecimento de uma habitação e águas representa, respetivamente, 40% e 26% do consumo elétrico conseguindo a sua redução através de um controlo rigoroso, sendo efetuada a supervisão através de *smartphones* e computadores[2].

### 1.4. Organização do documento

A dissertação encontra-se dividida em seis capítulos e 3 anexos, de maneira a apresentar ao leitor numa primeira fase as tecnologias existentes assim como alguns trabalhos já desenvolvidos. Numa segunda fase apresentam-se algumas tecnologias disponíveis para a elaboração da dissertação. Depois de uma melhor compreensão das tecnologias apresenta-se o hardware e software implementado na solução, nomeadamente o circuito usado para leitura da temperatura e os algoritmos de controlo utilizados. Numa penúltima fase são analisados os dados adquiridos e comparados entre si. Por fim, apresentam-se as conclusões retiradas da elaboração do trabalho e as melhorias a realizar futuramente.

- Capítulo 1 – Introdução
- Capítulo 2 – Estado Da Arte
- Capítulo 3 – Tecnologias de suporte
- Capítulo 4 – Arquitetura da Solução Proposta
- Capítulo 5 – Implementação da Solução Proposta
- Capítulo 6 – Conclusão e Trabalho Futuro



# Capítulo 2

## 2. Estado da arte

Com o objetivo que controlar os equipamentos na habitação e disfrutar de uma melhor qualidade de vida estão presentes no mercado algumas tecnologias, cada uma com as suas vantagens e desvantagens, que serão abordadas neste capítulo.

### 2.1. Soluções comerciais

#### 2.1.1. LK15 - Pack Iniciação X10

Este é um sistema *Plug & Play* que utiliza o protocolo X10, utilizando um módulo X10 para controlo remoto de iluminação, sendo a sua instalação fácil e possibilitando posteriormente adicionar mais módulos X10 [3].

O X10 é um protocolo de domótica que foi criado nos anos 70, usa a rede elétrica do edifício para transmitir os dados através da frequência injetada na rede elétrica, sendo este aspeto a sua maior vantagem face a outros protocolos pois não necessita de cablagem adicional nem de um elemento central para o seu funcionamento. O protocolo foi desenvolvido nos EUA e encontra-se bem presente no mercado americano estando também disponível na Europa mas é menos utilizado. Este protocolo é descentralizado ou seja cada dispositivo é controlado através de outro previamente instalado. Para o controlo dos dispositivos o protocolo X10 prevê a utilização de 16 códigos (A - P) para cada espaço da casa e outros 16 (1 - 16) códigos de dispositivos, apresentado na Figura 2.1, podendo assim endereçar um máximo de 256 dispositivos. Todos os aparelhos recebem a mesma mensagem mas apenas o aparelho endereçado será ativado [4].



*Figura 2.1 Pack de iniciação X10*

A comunicação do X-10 baseia-se na injeção de sinais de alta-frequência (120kHz) sobre a rede 220Vac, representando sinais binários de 1 e 0. O sinal é inserido logo a seguir à passagem por zero volts da onda sinusoidal de 50Hz, com um atraso de 200 microsegundos. Esta particularidade é usada pelos recetores para saberem quando devem escutar a linha [5].

### **2.1.2. Tecnologia KNX**

É uma tecnologia baseada nos protocolos EIB, EHS e BatiBus, que vem reduzir os problemas apresentado por estes. O objetivo desta tecnologia é proporcionar um maior conforto e segurança, através de um controlo e monitorização de todos os dispositivos, nomeadamente no controlo da temperatura, luz, segurança, controlo de água. O KNX possibilita a construção de um sistema modular reconfigurável, que permite depois da instalação acrescentar mais dispositivos ao sistema, com uma grande flexibilidade de ser possível a instalação de dispositivos de diferentes marcas desde que certificadas pela KNX. O sistema é descentralizado, ou seja, os dispositivos comunicam entre si sem necessidade de um nó central que faça a gestão da rede.

Os meios de comunicação suportados são o cabo entrelaçado, rede elétrica, radio frequência [6]. Uma das grandes vantagens deste protocolo é que a tecnologia KNX permite monitorizar vários tipos de dispositivos mesmo com diferentes tipos de comunicação devidos aos dispositivos (adaptadores) disponíveis no mercado para efetuar a conversão para o protocolo KNX. Uma das desvantagens do sistema é o seu elevado preço e apresentar uma instalação

complexa necessitando de especialista em ETS, *software* de configuração dos dispositivos KNX, para realizar as configurações dos dispositivos da rede[7].

A rede distribuída do KNX pode hospedar 65 536 dispositivos, em que cada endereço individual é constituído por 16 bits. A topologia permite 256 dispositivos numa linha, como representado na Figura 2.2 as linhas podem ser agrupadas com uma linha principal, denominada área. A rede pode ser formada por um máximo de 15 áreas, necessitando cada área de uma alimentação, com uma *backbone line* (linha de área) [8]. A sua topologia, embora grande parte das vezes seja hierárquica, Figura 2.2, pode também ser do tipo estrela, mista.

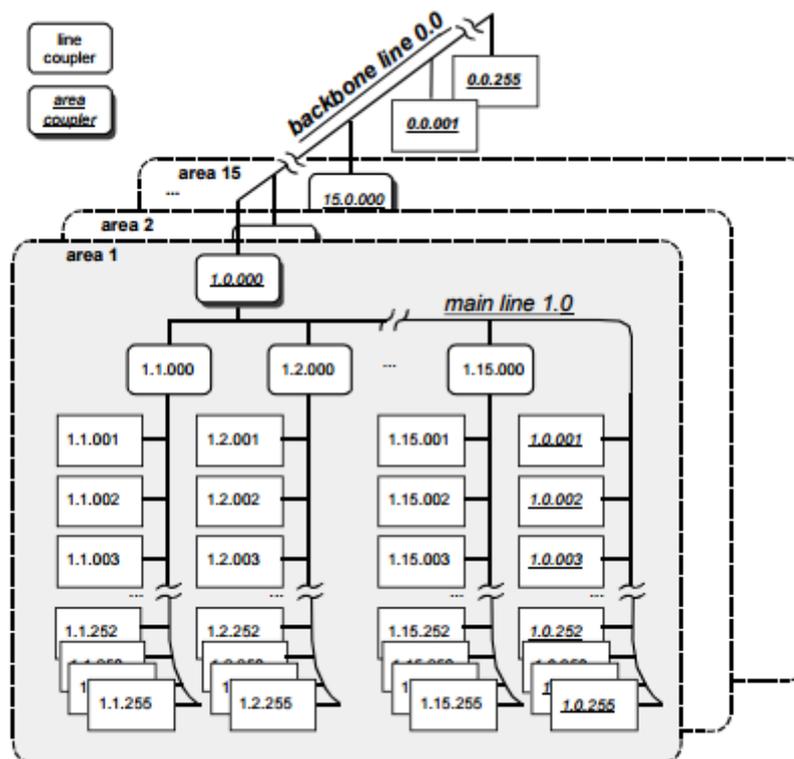


Figura 2.2 Topologia da rede KNX[8]

Muitas empresas trabalham na criação de dispositivos de controlo de temperatura para a rede KNX, havendo assim um grande leque de dispositivos disponíveis para o utilizador, embora tendo uma contra partida de os dispositivos apresentarem um elevado preço. Na Figura 2.3 é apresentado um dispositivo segundo as normas KNX que possui um controlo de temperatura ON/OFF com atuação em PWM [10].



Figura 2.3 Dispositivo de climatização KNX/ABB[11]

### 2.1.3. Clipsal C-Bus

O protocolo C-bus é usado para controlo de edifícios, nomeadamente da luz, da temperatura e de motores. Para as operações serem mais rápidas cada dispositivo possui o seu microprocessador, possibilitando a sua programação individual. Estas unidades não necessitam de um centro de controlo, pois cada um transmite os seus dados, sincronizado com o relógio do sistema, permitindo assim a troca de grandes quantidades de informação em pouco tempo. O C-bus para efetuar as comunicações usa cabos *unshielded twisted pair* (UTP) ou *wireless* bidirecional. Uma das desvantagens deste sistema é a instalação complexa, necessitando de um especialista para a efetuar, a vantagem é ser bastante rápido e podendo transmitir grandes quantidades de dados. É um sistema bastante flexível pois os módulos podem ser removidos, adicionados, movidos, reprogramados em qualquer posição da rede [12]. A rede C-bus permite a conexão de 255 redes usando diferentes topologias, como configuração em estrela, Figura 2.4, *daisy chain configuration* e uma combinação entre estas duas anteriores [13].

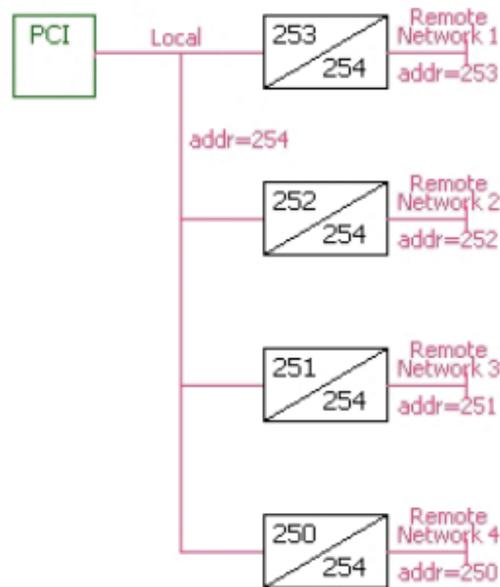


Figura 2.4 Exemplo de configuração em estrela[13]

A C-BUS oferece algumas opções no controlo de temperatura, como o termostato presente na Figura 2.5, o qual possui um controlo PID para controlar os relés ao qual está ligado[14].



Figura 2.5 Termostato C-BUS (adaptado de [14])

## 2.2. Soluções académicas

### 2.2.1. Domótica Inteligente – Um Contributo Prático [1]

Nesta dissertação o autor cria um módulo para controlar dispositivos numa habitação. O autor usa um *PLC* virtual, para a implementação do simulador/emulador de *Ladder* e o programa *Matlab* para a utilização de funções gráficas para a manipulação de dados e usando a *Fuzzy Logical Toolbox*, apresentado na Figura 2.6, para implementação de um controlo de lógica difusa. Efetuaram-se testes com autómato programável obtendo uma compatibilidade total do programa virtual. Uma das grandes vantagens desta aplicação foi o uso das grandes potencialidades do *Matlab*, mas não foram efetuados os testes práticos suficientes para a sua aplicação nomeadamente, testes em habitações e elaborar uma forma de comunicação entre o sistema de controlo e os módulos [1].

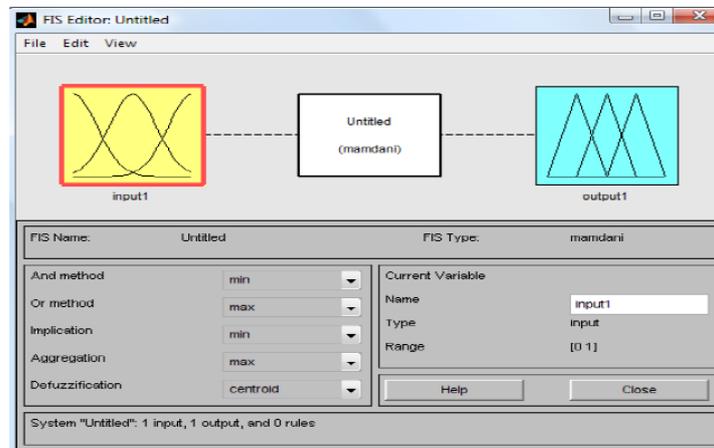


Figura 2.6 Tela inicial da Fuzzy Logical Toolbox[1]

### 2.2.2. Edifícios Inteligentes – Soluções para gestão de climatização em instalação de Domótica KNX [15]

O objetivo desta dissertação é efetuar o controlo do aquecimento num edifício residencial usando a tecnologia KNX e uma interface gráfica em *Labview*. Em cada

quarto seria instalado um sensor de temperatura, controlando as válvulas de aquecimento recorrendo um algoritmo PID comunicando depois com dispositivos KNX. Era controlada a temperatura do quarto tendo em consideração se a janelas estavam abertas pois caso estivessem se desligariam as válvulas de aquecimento [15].

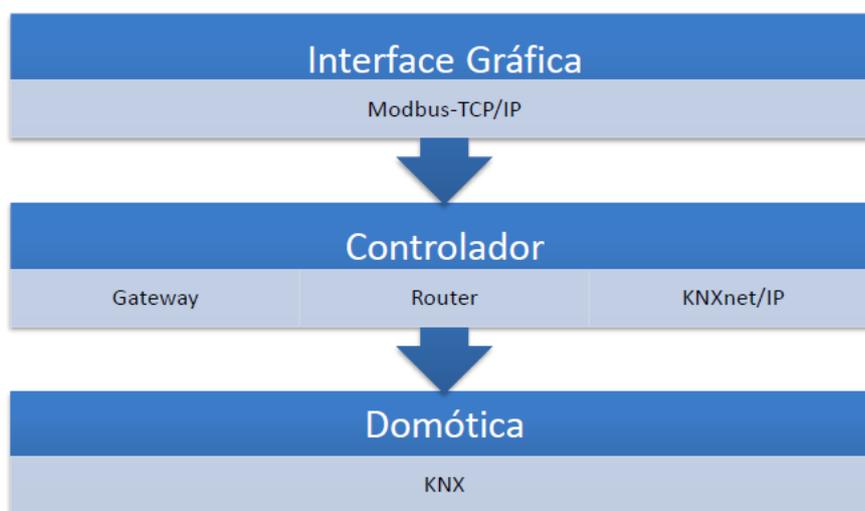


Figura 2.7 Arquitetura da Solução[15]

### 2.2.3. Desenvolvimento de Infraestrutura de Comando Multifunções EIB-KNX para *Smartphone* [16]

No desenvolvimento desta dissertação também foi aplicado um controlo de climatização, iluminação e de estores usando a tecnologia KNX e comunicando através de uma gateway (KNX-IP) para uma plataforma móvel capaz de monitorizar os dispositivos na habitação, trazendo assim a inovação dos *smartphones* [16].

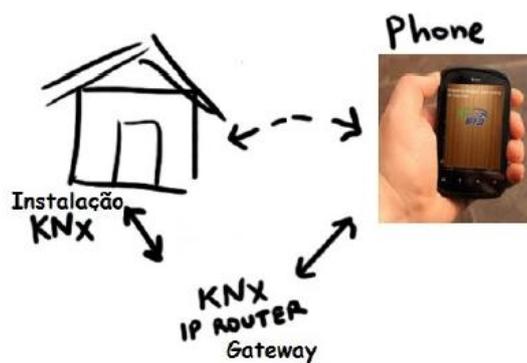


Figura 2.8 Representação das componentes da infraestrutura dos comandos[16]

# Capítulo 3

## 3. Tecnologias de suporte

Neste capítulo serão apresentadas as diferentes tecnologias e algoritmos que posteriormente serão aplicados no trabalho em desenvolvido.

### 3.1. Sensores de temperatura

Os sensores de temperatura são a parte essencial para a medição da temperatura num processo térmico. Para a medição de temperaturas serão abordados os Sensores de resistência (RTD), Circuitos Integrados (CI) e os termopares presentes na Figura 3.1, apresentando as suas propriedades.

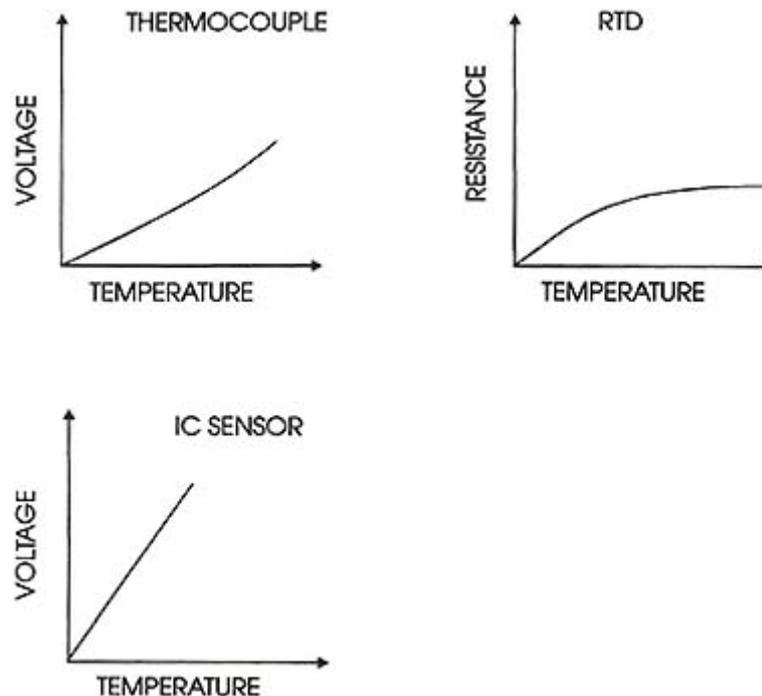


Figura 3.1 Comparação de Sensores de Temperatura[17]

### 3.1.1. RTD

Os sensores resistivos alteram a sua resistência em função da temperatura a que estão sujeitos. Consistem num metal ou liga metálica, por exemplo de platina, devido a sua estabilidade e linearidade. Para obter o valor da resistência é necessário aplicar ao sensor uma pequena corrente de forma a medir a diferença de potencial aos seus terminais. Contudo, essa corrente pode provocar auto aquecimento afetando a leitura. Para obter a resistência a uma determinada temperatura recorre-se à da equação de *Callendar-Van Dussen* ou a tabelas Resistência/Temperatura do sensor, possuindo normalmente escalas entre  $-200^{\circ}\text{C}$  a  $800^{\circ}\text{C}$  [18].

A Figura 3.2 apresenta uma RTD mais exatamente uma Pt1000 que tem uma resistência de 1000ohm a  $0^{\circ}\text{C}$ .

Na leitura devemos ter em conta o comprimento dos cabos do sensor pois os circuitos de acondicionamento do sinal encontra-se a alguns metros de distância e a resistência dos próprios cabos vão gerar erros nas leituras, sendo necessário a leitura a 4 e 3 fios quando a distância é elevada.



Figura 3.2 Sensor de Temperatura PT1000[19]

### 3.1.2. Circuitos integrados

Alguns sensores de circuitos integrados, Figura 3.3, permitem medir a temperatura possuindo uma saída proporcional à temperatura a que estão sujeitos, não necessitando de eletrônica adicional, o oposto dos sensores falados anteriormente, diminuindo assim o preço e dificuldade de calibração. Possuem um gama de temperatura limitada normalmente entre  $-55^{\circ}\text{C}$  a  $150^{\circ}\text{C}$  e com saídas em tensão ou



### 3.2.1. Radiofrequência

A radiofrequência efetua a transmissão de dados através de oscilações eletromagnéticas de um dispositivo para outro. A faixa de frequência encontra-se entre 3 kHz e 300 GHz. A rede sem fios utiliza a faixa 2.4 GHz, também conhecida como ISM, que não necessita licença, sendo assim fácil a sua aplicação. No entanto obriga a convivência entre dispositivos de outras fontes de RF na mesma faixa. As tecnologias que usam esta banda são o *Bluetooth* (IEEE 802.15.1), *ZigBee* (IEEE 802.15.4) e *Wi-Fi* (802.11b). Na Tabela 3.1 podemos verificar as vantagens e desvantagens de cada tecnologia [23]:

Norma IEEE (Nome de Mercado)	802.15.1 (Bluetooth)	802.11b (Wi-Fi)	802.15.4 (ZigBee)
Aplicação Principal	Eliminar a fiação atual.	Ethernet industrial	Controle e Monitoração
Frequência de operação	2,4 GHz	2,4 GHz	2,4 GHz
Taxa de comunicação (Kbps)	1000-3000	11000	20-250
Distâncias alcançadas com visada direta (m)	30 (Classe 2) 100+(Classe 1)	100 + (Antenas direcionais)	30-70, 100+ (Com amplificador externo)
Número de Dispositivos	7	32	2*64
Autonomia da bateria (dias)	1 a 7	0,5 a 5	100 - 1000+
Consumo na Transmissão	45mA (Classe 2) <150mA (Classe 1)	300mA	30mA
Conveniência para Controle e Supervisão, aplicações	Baixa (Boa média, mas conexão inicial lenta)	Baixa (Taxa alta, mas conexão inicial lenta)	Baixa (Bom compromisso entre taxa e custo de

Tabela 3.1 Comparação das Tecnologias de Comunicação Editado[23]

### 3.2.2. Modbus

O protocolo Modbus foi criado em 1979 pela *Modicon* com o objetivo de criar um protocolo de comunicação para os PLC's sendo mais tarde um protocolo público. O Modbus é um protocolo de mensagens da camada de aplicação, posicionado na

camada 7 do modelo OSI. O protocolo Modbus não define nenhum meio físico de transmissão, sendo usadas habitualmente RS232, RS485, TCP/IP para transmitir as mensagens Modbus, Figura 3.4.

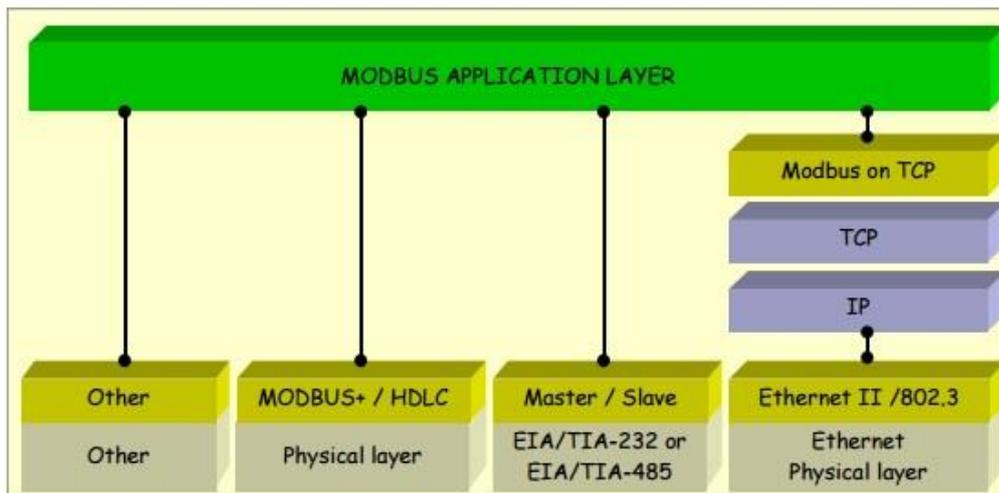


Figura 3.4 Estrutura de Camadas do Protocolo Modbus[24]

A comunicação Modbus é do tipo *master-slave* em que esta é iniciada pelo *master* (Cliente Modbus), responsável pelo meio de transmissão de forma a evitar colisões e enviar mensagens a todos os *slaves* (servidores Modbus).

As mensagens enviadas pelo *master* podem destinar-se a um único *slave* (*unicast*) ou a todos os *slaves* (*broadcast*). [25].

Como apresentado na Figura 3.5, a mensagem ModBus é composta, pelo endereço (1 byte) que identifica o destinatário, função (1byte), dados (0 a 252 bytes), CRC/LRC(2 bytes)

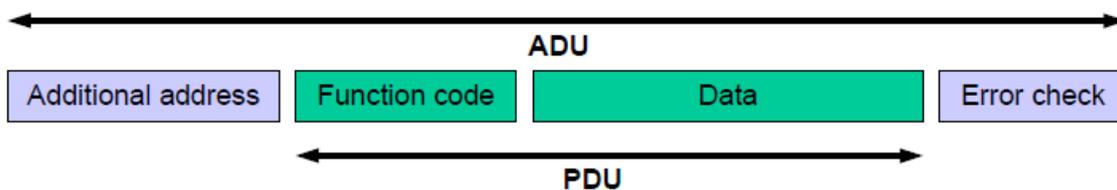


Figura 3.5 Estrutura da mensagem Modbus[24]

Se a mensagem tiver um endereço de destino 0 destina-se a todos os equipamentos, os *slaves* podem ter endereços de 1 a 247 e os endereços de 248 a 255 estão reservados. O campo de função define o pedido requerido pelo *master*, alguns deles pressupõem uma resposta do *slave*, outros não. Através destes

pedidos/mensagens o cliente pode ativar saídas digitais do servidor, escrever nas suas memórias internas ou pedir ao servidor que lhe envie uma mensagem com o estado das suas entradas e saídas e o valor das suas memórias. O código de função de cada mensagem é codificada em 8 bits, presente na Figura 3.6. O campo de dados contém informação que o *master* usa para efetuar uma ordem definida pelo código de função [24].

				Function Codes			
				code	Sub code	(hex)	Section
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
	Write Multiple Coils		15		0F	6.11	
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access	Read File record	20		14	6.14	
		Write File record	21		15	6.15	
	Diagnostics	Read Exception status	07		07	6.7	
		Diagnostic	08	00-18,20	08	6.8	
		Get Com event counter	11		0B	6.9	
		Get Com Event Log	12		0C	6.10	
		Report Slave ID	17		11	6.13	
		Read device Identification	43	14	2B	6.21	
	Other	Encapsulated Interface Transport	43	13,14	2B	6.19	

Figura 3.6 Códigos do campo de função[24]

Existem mensagens Modbus RTU e ASCII. Numa mensagem RTU cada byte da mensagem Modbus é enviado em binário através de palavras série RS232 ou RS485. Cada byte Modbus é enviada numa palavra série composta pelo start bit, seguido do bit menos significativo do byte Modbus, do bit de paridade e do stop bit seguido dos restantes bits da mensagem. Numa mensagem Modbus ASCII cada número é dividido em dois dígitos. Cada um deles é codificado em 7 bits de acordo com a tabela ASCII e cada um deles é enviado numa palavra série. Todas as mensagens Modbus ASCII começam pelo carácter “:” e terminam com os caracteres “CR” e “LF”.

Os dois tipos de mensagens utilizam algoritmos de detecção de erro diferentes. No modo RTU utiliza-se o CRC16 enquanto que no modo ASCII o LRC8. O CRC16 é gerado pelo equipamento emissor e calculado de novo pelo receptor se os dois CRC16 calculados forem iguais a mensagem foi bem transmitida. O algoritmo LRC8 calcula a soma de todos os bytes da mensagem, com exceção dos bytes “:”, “CR”, “LF”. O resultado da soma é negado e posteriormente incrementado em uma unidade. São gerados 8 bits, representando os 4 primeiros e os 4 últimos números de 0 a F. Cada um deles é considerado um caractere e codificado segundo a tabela ASCII [26].

### 3.3. Controle da potência

#### 3.3.1. Dispositivos de controle de potência

O TRIAC (*Triode for alternating current*) é um dispositivo semicondutor com uma estrutura interna semelhante à de dois tiristores em oposição e em paralelo, possuindo apenas uma *gate*. O TRIAC é constituído por dois terminais MT1 e MT2 e uma *gate* G. A corrente que flui através do TRIAC pode ser controlado nos dois sentidos. São muito utilizados no controle de aquecimento, no controle de iluminação, no controle máquinas elétricas e conversores CA, pois tem a vantagem de acionar grandes potências com circuitos em que as correntes de disparo são da ordem dos miliamperes [27].

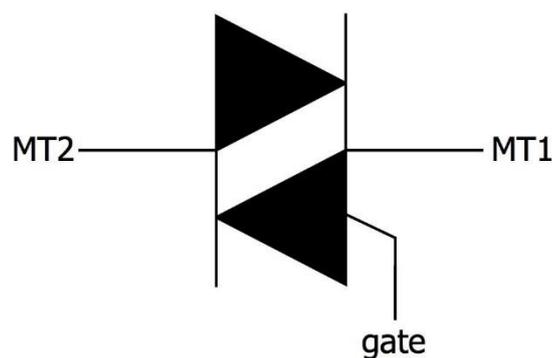


Figura 3.7 Símbolo de um TRIAC[28]

No funcionamento do TRIAC, este inicia a condução quando se aplica um mínimo de corrente e tensão necessária na Gate para a sua condução, entrando no corte apenas quando a corrente na gate e da carga seja zero [29].

Para o TRIAC passar ao corte não basta que a tensão na gate seja zero é necessário que a corrente principal seja nula, como se pode verificar na Figura 3.8, esta é a principal desvantagem do TRIAC.

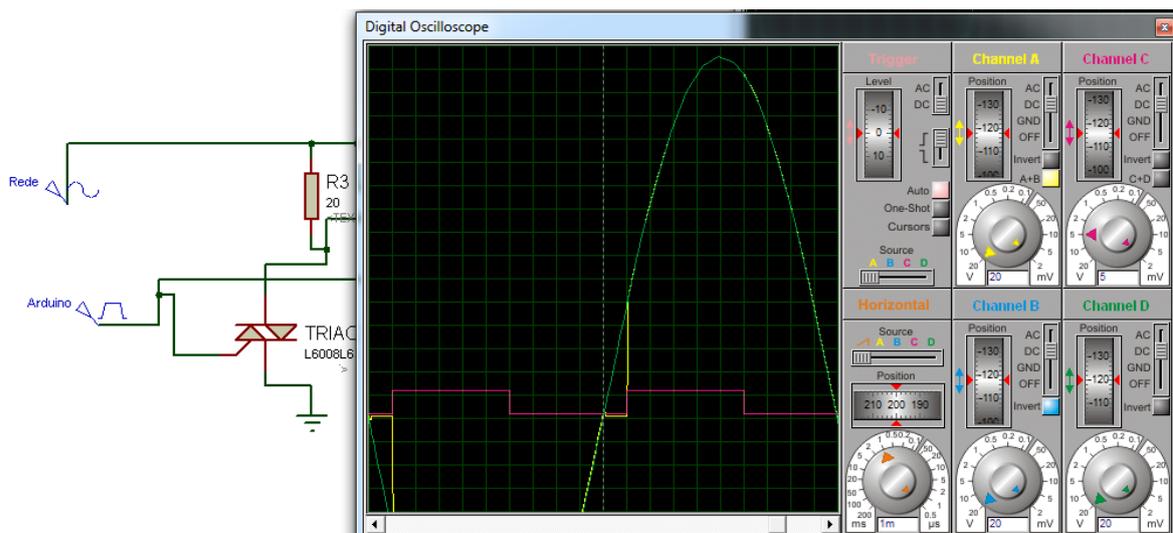


Figura 3.8 Controle usando o TRIAC

Também foram analisados outros dispositivos de potência, nomeadamente os IGBT, MOSFET e SCR. O IGBT (Insulated-Gate Bipolar Transistor) passa ao corte quando a tensão na gate é zero, mostrando ser uma vantagem em relação ao TRIAC, que apenas corta quando a corrente de carga está a zero e a da gate também. No entanto, este dispositivo apenas conduz num sentido, Figura 3.9

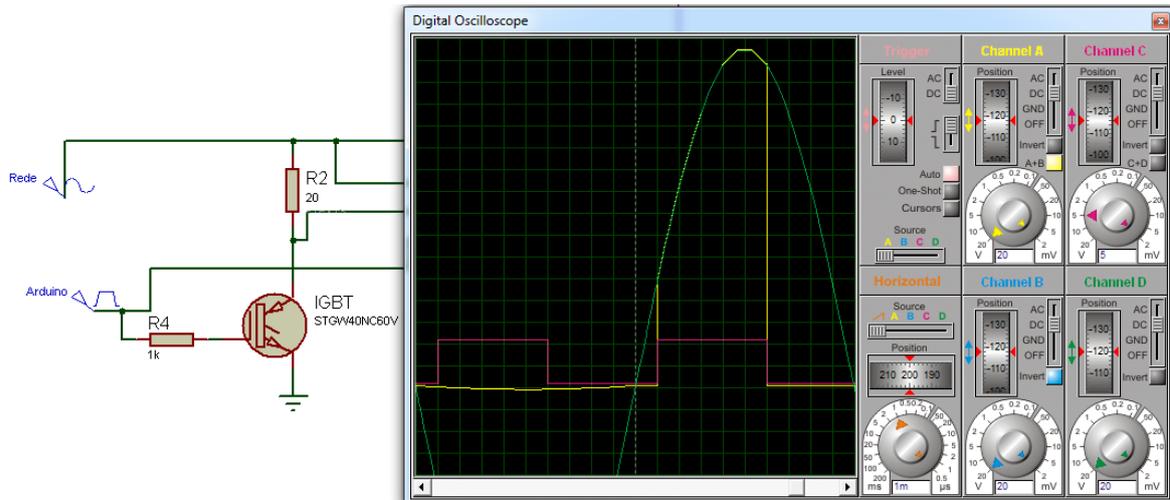


Figura 3.9 Controlo usando IGBT

O MOSFET é unidirecional tal como o IGBT mas permite cortar ou ligar de acordo com a tensão aplicada na gate, a qualquer altura.

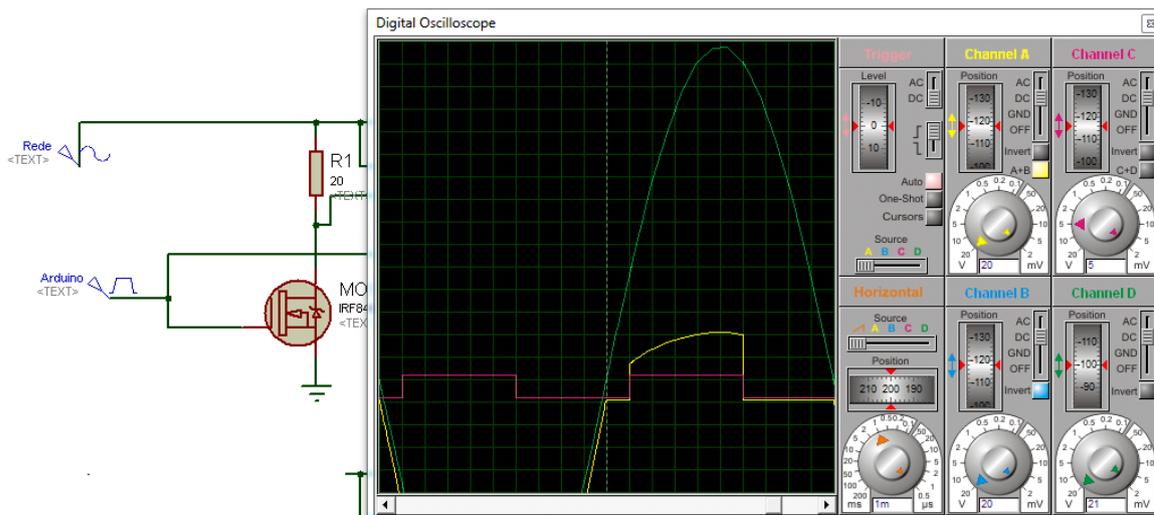


Figura 3.10 Controlo usando um MOSFET

Os SCR (Silicon Controlled Rectifier) é bidirecional e pode disparar tanto com sinais positivos ou negativos na gate, podendo ser controlado aplicando uma tensão na gate entrando na zona de condução e assim permanece mesmo que se retire o sinal da gate, conseguindo o corte apenas quando a corrente direta que o percorre baixa para além da corrente de manutenção,  $I_H$ , fornecida pelo fabricante.

### 3.3.2. PWM

O PWM (*Pulse Width Modulation*) é uma técnica usada para modular a largura de pulso de um sinal digital. Conforme a Figura 3.11 ilustra, o mesmo sinal com tensões de 0V e 5V está ativo uma percentagem de tempo (*duty cycle*) maior ou menos relativamente ao seu período.

$$Duty\ cycle = \frac{T_{on}}{(T_{on} + T_{off})} \quad (1)$$

A tensão média deste sinal varia conforme o seu *duty cycle*.

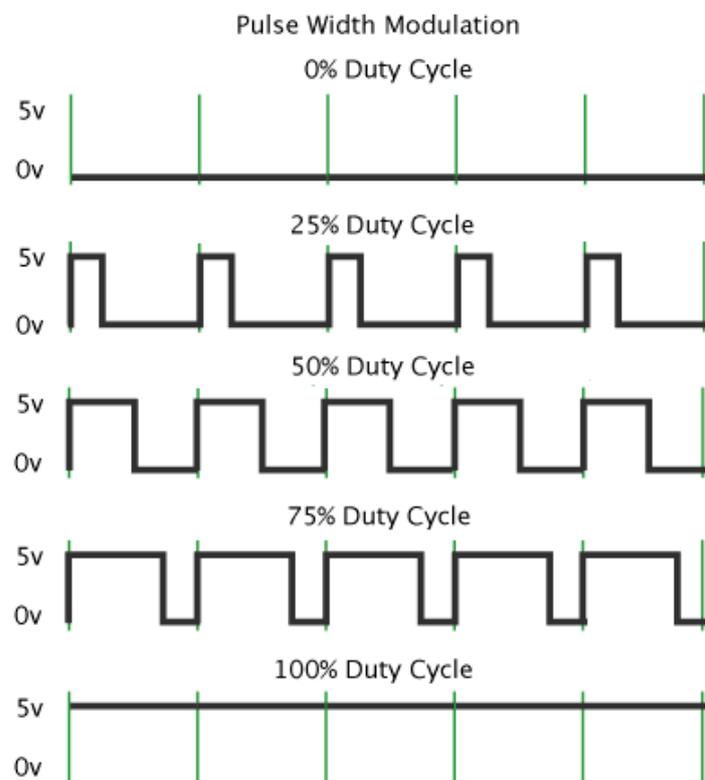


Figura 3.11 Resposta PWM controlada pelo Duty Cycle adaptado de [30]

### 3.4. Algoritmos de controlo

Para obter uma maior eficiência energética é necessário a aplicação de algoritmos de controlo de forma a criar um sistema estável de acordo com o *setpoint* definido. Pretende-se atingir o *setpoint* rapidamente e sem overshoot. Na prática foram testados os 3 algoritmos mais usados.

### 3.4.1. Controle ON / OFF

Algoritmo mais simples, que se baseia como o seu próprio nome indica em ligar e desligar o dispositivo a ser controlado, ou seja depois de ser definido o *setpoint* para se manter estável irá proceder a esse controle, Figura 3.12. Trate-se um controle grosseiro e não é recomendável para *setpoints* rigorosos. Na implementação foi útil para fornecer informações para os controles seguintes.

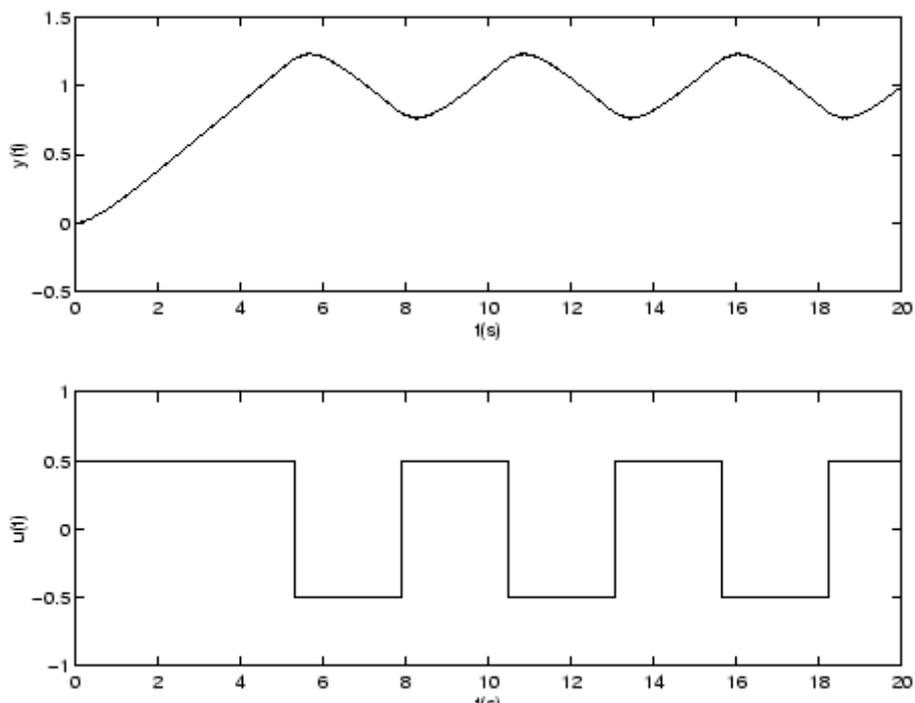


Figura 3.12 Exemplo de uma saída de controle On/Off[31]

### 3.4.2. PID

Usado em mais de metade da indústria este tipo de controle fornece um vasto leque de aplicações em diferentes dispositivos, com um controle robusto mas com a necessidade de um bom conhecimento do sistema para efetuar o seu *tuning* nas constantes do algoritmo. As letras sigla PID, representam P para a parte proporcional I integral e D para a parte derivativa, com ganhos  $K_p$ ,  $K_i$  e  $K_d$  respetivamente. Dependendo do sistema o algoritmo oferece diferentes formas de controle PID, PI, PD e P, sendo P semelhante ao controle ON/OFF sendo os mais usados PID e PI [32].

A expressão do algoritmo PID é dada pela seguinte equação [33]:

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2)$$

Em que  $u(t)$  é a saída do controlador,  $e(t)$  é a diferença entre o *setpoint* pretendido e a resposta do sistema. E  $K$ ,  $T_i$  e  $T_d$  respetivamente, o ganho proporcional, tempo integral e tempo derivativo. Na sua chamada forma paralela a equação apresenta esta configuração:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d(\tau) + K_d \frac{d}{dt} e(t) \quad (3)$$

Com a transformada de *Laplace*:

$$G(s) = K_p + \frac{K_i}{s} + sK_d \quad (4)$$

Podendo obter-se:

$$K_p = K \quad (5)$$

$$K_i = \frac{K}{T_i} \quad (6)$$

$$K_d = KT_d \quad (7)$$

Para implementação digital, é usado a transformada de *Z*:

$$U(z) = \left[ K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \right] E(z) \quad (8)$$

Rearranjando-se:

$$U(z) = \left[ \frac{(K_p + K_i + K_d) + (-K_p - 2K_d)z^{-1} + K_d z^{-2}}{1 - z^{-1}} \right] E(z) \quad (9)$$

Define

$$K_1 = K_p + K_i + K_d \quad (10)$$

$$K_2 = -K_p - 2K_d \quad (11)$$

$$K_3 = K_d \quad (12)$$

Reescrevendo (9)

$$U(z) - z^{-1}U(z) = [K_1 + K_2z^{-1} + K_3z^{-2}]E(z) \quad (13)$$

Pela equação de diferenças

$$u(k) = u[k - 1] + K_1e[k] + K_2e[k - 1] + K_3e[k - 2] \quad (14)$$

Cada constante de ganho tem um papel fundamental na resposta do sistema, controlando o overshoot, o tempo de subida, o *settling time* e *steady-state error* (S-S Error), representado na Tabela 3.2

*Tabela 3.2 Efeitos nas alterações de cada ganho[34]*

Ganho	Tempo de Subida	Overshoot	Settling Time	S-S Error
$K_p$	Diminui	Aumenta	Poucas Alterações	Diminui
$K_i$	Diminui	Aumenta	Aumenta	Elimina
$K_d$	Poucas Alterações	Diminui	Diminui	Sem Alterações

Para conseguir  $K_p$ ,  $K_i$ ,  $K_d$  ideais para o sistema pretendido podemos usar diferentes métodos de *tuning*, na qual consiste através de fórmulas e análise da resposta do sistema obter os valores corretos destas constantes. Alguns dos métodos mais conhecidos são:

- Método prático
- Método de *Ziegler-Nichols*

O método prático passa por ajustar o parâmetro  $K$  até existir apenas 10% de *overshoot*, usando apenas a parte proporcional do PID, afinar  $T_d$  para “remover” o *overshoot* e depois aumentar  $T_i$  até o S-S error ser eliminado. Este método é bastante simplesmente embora bastante trabalhoso na obtenção de dados para as variáveis.

O método de *Ziegler-Nichols* de malha fechada, permite calcular o ganho  $K_p$ , usando o ganho  $K_u$  e o período de oscilações de  $P_u$  recorrendo apenas a parte proporcional do PID. O ganho  $K_u$  é encontrado quando o sistema oscila indefinidamente no estado estacionário e o  $P_u$  é o tempo necessário para completar uma oscilação quando o sistema encontra-se no estado estacionário.

A aquisição das variáveis consiste em remover o tempo integral ( $T_i$ ) e o tempo derivativo ( $T_d$ ), de seguida ajustar a parte proporcional ( $K$ ), aumentando e diminuindo o ganho até as oscilações terem uma amplitude constante, gravar os valores  $K_u$  e  $P_u$  e de acordo com a Tabela 3.3 obter os valores das variáveis,  $K$ ,  $T_i$ ,  $T_d$ .

*Tabela 3.3 Cálculos Para  $K$ ,  $T_i$ ,  $T_d$  no método Ziegler-Nichols[35]*

	$K_c$	$T_i$	$T_d$
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

Um das grandes vantagens deste método é a sua simplicidade, pois apenas controla-se a parte proporcional e inclui a dinâmica de todo o processo fornecendo mais informação de como o processo se vai comportando. Duas das desvantagens deste método é a experiência pode ser demorada e podemos chegar a regiões instáveis a testar o controlador P, o que pode fazer com que o sistema fique fora de controlo[35].

### 3.4.3. Lógica difusa

A lógica difusa tenta implementar fórmulas matemáticas nos computadores para recriar a linguagem humana e processos cognitivos. É um algoritmo poderoso pois trabalha com respostas vagas, imprecisas e incompletas ajustando-se a quase qualquer tipo de sistema. Num sistema de aquecimento diz-se que abaixo de 40°C é frio e acima de 40°C é quente, ou seja com a temperatura a 39,5°C um sistema normal diria frio, na lógica difusa isto significaria que está 99,5% quente e 0,5% é fria [36].

O sistema difuso para controlar sistemas não lineares efetua quatro passos, definir as variáveis de entrada/saída, as regras difusas, mecanismo de inferência e a defuzzificação.

Nos primeiros passos trata-se da fuzzificação em que são definidas as variáveis de entrada/saída e tem de ser convertidas em graus qualitativos, denominados de termos difusos. Depois estes valores tem de passar por um conjunto de regras e funções difusas. Existem vários tipos de funções sendo as mais usadas as triangulares e trapezoidais apresentando uma maior eficiência. O eixo das abcissas apresenta os valores da entrada do sistema e o das ordenadas 0 e 1 representando o grau de veracidade que uma entrada tem para um determinado termo linguístico.

O comportamento do sistema é representado sob a forma de regras de inferência lógica *Se-Então*. Em que o *Se* contém uma ou mais condições denominadas de antecedentes e o *Então* contém uma ou mais ações denominadas conseqüências. Cada antecedente tem associado um grau de veracidade como resultado de fuzzificação. Na seguinte Tabela 3.4 mostra alguns operadores (E e OU) na combinação de antecedentes, como A e B sendo  $\mu_A(x)$  e  $\mu_B(x)$  respetivamente para originar apenas um grau de veracidade [37].

Tabela 3.4 Operadores Difusos[26]

OU		E	
Max	$\max\{\mu_A(x), \mu_B(x)\}$	Min	$\min\{\mu_A(x), \mu_B(x)\}$
Asum	$\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$	Prod	$\mu_A(x) \cdot \mu_B(x)$
Bsum	$\min\{1, \mu_A(x) + \mu_B(x)\}$	Bdif	$\max\{0, \mu_A(x) + \mu_B(x) - 1\}$

O último passo da lógica difusa é a defuzzificação para decifrar os valores produzidos pelos operadores e agregações da fuzzificação. Na Tabela 3.5 são apresentados alguns métodos de defuzzificação como o centro de gravidade, média dos máximos, máximo, altura, altura modificada, sendo o mais usado a técnica do centro de gravidade.

Tabela 3.5 Métodos de Defuzzificação[26]

Método	Fórmula
Centro de gravidade	$U = \frac{\int_{min}^{max} u\mu(u)du}{\int_{min}^{max} \mu(u)du}$
Centro de gravidade por singularidades	$U = \frac{\sum_{i=1}^p [u_i\mu_i]}{\sum_{i=1}^p [\mu_i]}$
Máximo mais à esquerda	$U = inf(u'), \mu(u') = sup(\mu(u))$
Máximo mais à direita	$U = sup(u'), \mu(u') = sup(\mu(u))$

A Figura 3.13 apresenta uma associação triangular, com saídas não nulas, utilizando diferentes métodos de defuzzificação.

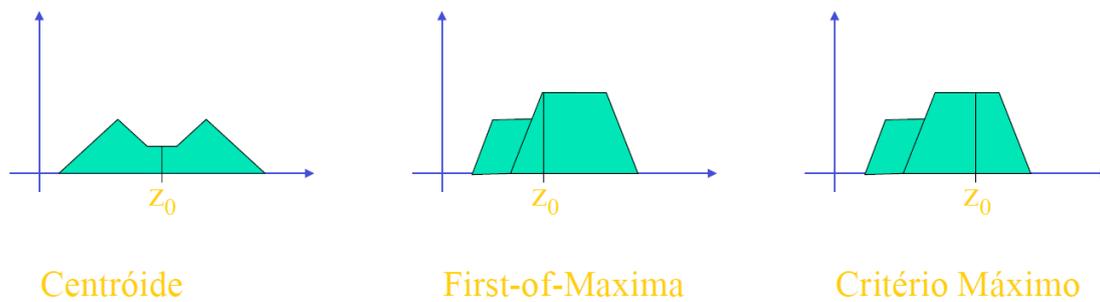


Figura 3.13 Métodos de defuzzificação[38]

# Capítulo 4

## 4. Arquitetura da solução proposta

Depois de analisadas as diferentes tecnologias e comerciais existentes proporcionou informações vitais para a arquitetura da solução proposta. Para implementação deste tipo de tecnologia temos de ter em conta os pontos fortes abordados anteriormente nomeadamente, redes sem-fios possibilitando assim a comunicação sem necessidade de extra cablagem, algoritmos de controlo de forma a haver uma eficiência energética mais positiva, os diferentes sensores de temperatura para obtenção da temperatura de um forma mais rigorosa, formas de atuação mais rápidas e eficazes nomeadamente transístores e triacs. O objetivo é cumprir os requisitos dos pontos fortes mencionados anteriormente, para oferecer uma proposta integradora e dinâmica dentro da domótica, capaz de agregar os requisitos mais importantes numa solução de domótica.

Nesta solução, representado na Figura 4.1, é efetuado o controlo através de um sistema central recolhendo informações sobre o estado de temperatura do sistema de aquecimento e enviando informações ao mesmo sistema com a temperatura pretendida, efetuando a comunicação através Modbus ou pela rede sem-fios RF. Para obter uma boa eficiência energética é necessário recorrer a algoritmos de controlo implementados no microcontrolador, fazendo uma gestão programatizada do controlo da temperatura.

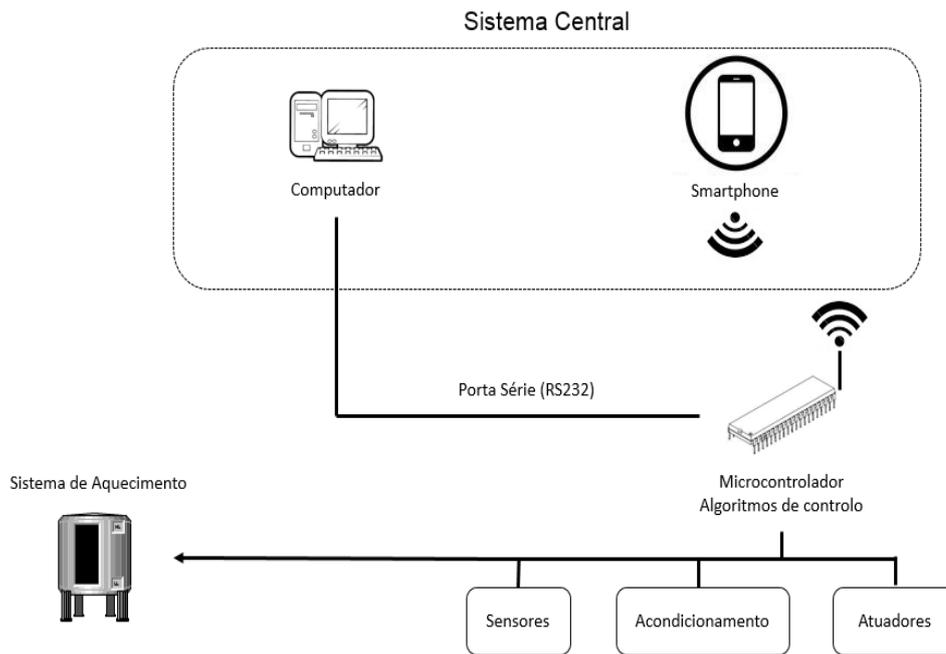


Figura 4.1 Diagrama de blocos da solução proposta

# Capítulo 5

## 5. Implementação da solução proposta

Neste capítulo, explica-se passo a passo o desenvolvimento da solução proposta seguindo a organização da figura anteriormente apresentada, mas especificando os componentes utilizados, representados na Figura 5.1. No sistema central será usada a plataforma Movicon e o bluetooth para comunicação sem fios com a unidade de controlo baseada em um ATMEGA (arduino). Com recurso ao *software* Movicon criou-se uma interface de fácil utilização ao utilizador, permitindo definir *setpoints* de temperatura assim como a visualização em tempo real da temperatura. Para que o módulo central possa interagir com a unidade de controlo foi utilizado a comunicação Bluetooth. O Bluetooth fornece uma comunicação sem fios facilitando a observação de temperatura num limite de, 10m-100m. Com recurso ao microcontrolador ATMEGA do Arduino implementa-se os códigos com diferentes funções, como a transmissão de dados via bluetooth e RS232, leitura da temperatura do sensor PT1000 e a atuação do TRIAC, através de PWM, e os algoritmos de controlo da temperatura entre 0°C – 100°C.

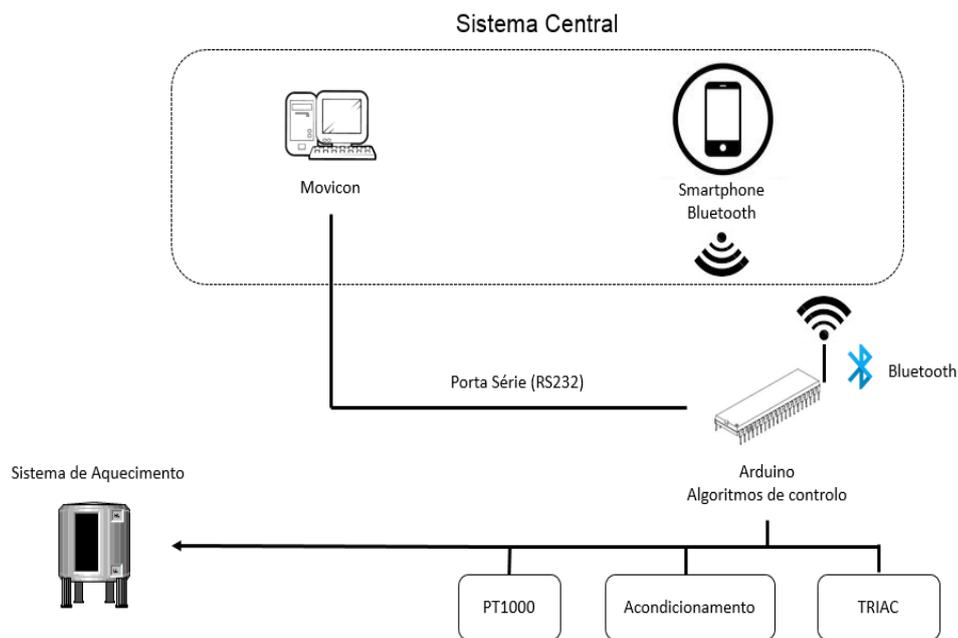


Figura 5.1 Diagrama de blocos especificado

## 5.1. Hardware desenvolvido

### 5.1.1. RTD e acondicionamento

O sensor de temperatura aplicado foi uma RTD mais exatamente uma PT1000. Para efetuar a sua leitura foi usada uma fonte de corrente, Figura 5.2, de 1mA para produzir uma diferença de potencial na ordem dos volts ( $V_{Sensor}$ ), ou seja a 0°C é obtido aproximadamente 1V e aproximadamente 1,4V a 100°C.

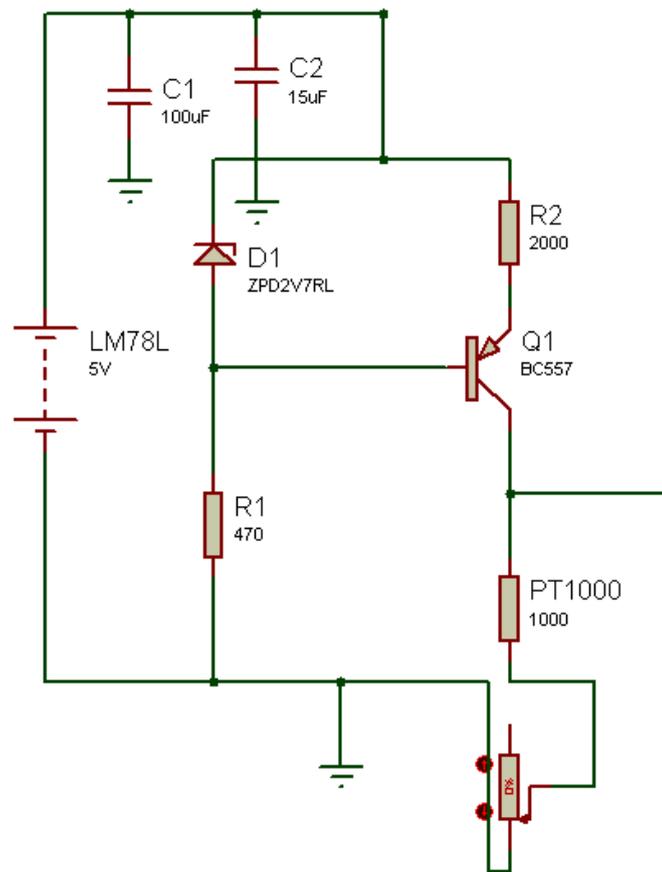


Figura 5.2 Fonte de Corrente

Depois de conhecida a tensão do sensor, esta foi amplificada e acondicionada, Anexo 1, por um amp-op *rail-to-rail*, MCP6022, para obter tensões na gama do microcontrolador, 0V- 5V. Para obter valores de acordo com a gama foi necessário uma tensão de referência com uma tensão semelhante à tensão da temperatura a 0°C, sendo usado uma referência de precisão TL431, fornecendo 2.5V constantes a saída e através de um divisor resistivo obter 1V como tensão de referência ( $V_{Ref}$ ). Recorrendo á formula (15), calculou-se o ganho que o amp-op necessitava para obter tensões na gama do microcontrolador.

$$Ganho = \frac{R_8}{R_7} \quad (15)$$

$$V_{U1:B} = \frac{R_8}{R_7} (V_{Sensor} - V_{Ref}) \quad (16)$$

A saída do amp-op foi ligado a uma entrada analógica do microcontrolador, que através da fórmula (17) obtém-se a temperatura que o sensor está a medir, tendo sido necessário fazer uma calibração no *software* devido a alguns ruídos na eletrónica, mesmo após de se instalar alguns condensadores para manter estável a alimentação e as tensões de leitura.

$$Temperatura = \frac{(A0 * 100)}{1023} \quad (17)$$

Sendo A0 a entrada analógica proveniente do circuito de leitura, 100 representando a gama dinâmica de temperatura (0°C-100°C), a dividir pelo número de bits do microcontrolador, 1024.

### 5.1.2. Módulo de potência

Para o controlo da temperatura e da resistência de aquecimento usa-se um optoacoplador MOC3021 e o TRIAC BT136 600, representado na Figura 5.3. Sendo R a resistência de aquecimento a controlar, neste protótipo um reservatório de água. O microcontrolador possui um algoritmo de controlo que envia a sua resposta em PWM através de uma porta digital do mesmo, aplicado à base do TRIAC, Figura 5.3, controlando o aquecimento da resistência.

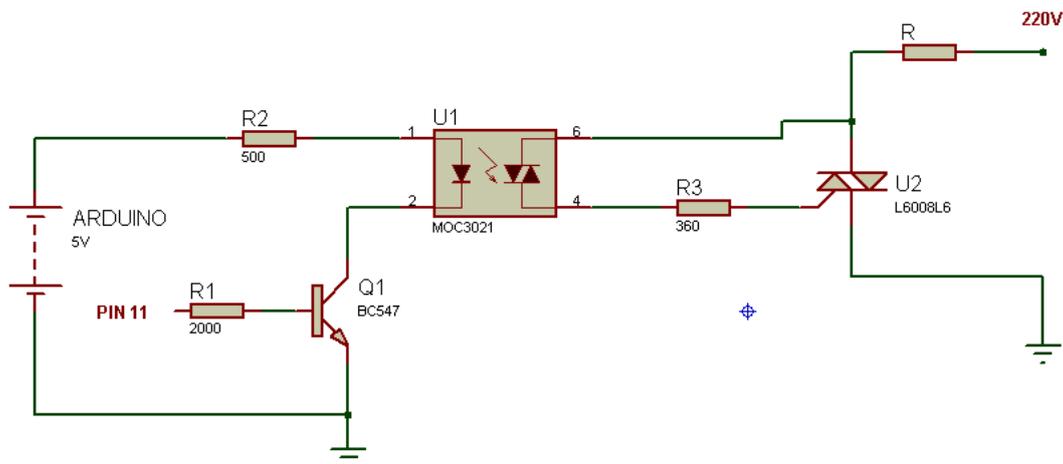


Figura 5.3 Circuito de Potência

Nomeadamente para efetuar o controle do TRIAC e da resistência de aquecimento, foi necessário a implementação de um *Timer* no Arduino para colocar a saída digital do Arduino à mesma frequência da rede, 50Hz, usando assim um *prescaler* de 8bits, para obter aproximadamente 50hz. A atuação do TRIAC foi efetuada através da contagem de ciclos durante 1,5s, durante este tempo a rede efetua 75 ciclos, pois o período da rede é 20 ms. Depois, de acordo com o *duty cycle* o microcontrolador ativa o TRIAC, por exemplo, durante 45 ciclos (0,9s) ativa o TRIAC e durante 30 ciclos (0,6s) poem o TRIAC ao corte, obtendo um *duty cycle* de 60% e controlando dessa o aquecimento da resistência de aquecimento.

### 5.1.3. Bluetooth ZS-040

O dispositivo de radio frequência usado foi o Bluetooth que usa a banda de 2.4GHz da ISM. Este dispositivo permite transmissão de dados de curta distância sem necessidade de uma infraestrutura de rede, podendo ter dois módulos como HC-05 e HC-06, configuráveis para *master/slave* ou *master* ou *slave*. Neste caso foi usado o módulo HC-05 com a opção de *slave* ativada na linha de comando AT (comandos de configuração do Bluetooth).

Este Bluetooth, usa uma alimentação de 3.6V - 6V sendo por isso possível a sua alimentação através do Arduino não sendo necessário uma alimentação externa. Para efetuar as comunicações foi necessário instalar um divisor resistivo entre o Tx

do microcontrolador e o Rx do Bluetooth para obter 3.3v, pois valores superiores a 3.3v destruiria a porta Rx do Bluetooth. Para realizar o carregamento do programa no microcontrolador (via USB) estas portas devem ser desligadas pois usam o mesmo Tx/Rx. Podemos efetuar algumas alterações no Bluetooth como nome, *password*, módulos (*master/slave* ou *master* ou *slave*), definir o *baudrate* entre outras aplicações, utilizando os comandos AT do Bluetooth.

Para comunicar com o Bluetooth através do *smartphone* recorreu-se a uma aplicação gratuito, Blueterm, presente na Google Store, onde é possível enviar o *setpoint* pretendido assim como receber informação da temperatura real, Figura 5.4.



Figura 5.4 Aplicação Android Blueterm

#### 5.1.4. LCD

Para a visualização de uma forma mais intuitiva usou-se um LCD 16x2 (16 colunas por 2 linhas), para a visualização do *setpoint* definido e a visualização da temperatura em tempo real. O LCD era alimentado a 5V do Arduino ligado a um potenciómetro para o controlo da luminosidade, Anexo 2. O código de envio de informação para o LCD é uma biblioteca já existente no Arduino, definindo o número de linhas e colunas que possui o LCD e mudança de pinos se necessário[39].

### 5.2. Software desenvolvido

#### 5.2.1. MOVICON

Com recurso á ferramenta Movicon é possível criar interfaces gráficas para um grande leque de dispositivos, *smartphones*, paginas web e computadores.

No desenvolvimento deste trabalho elaborou-se uma interface gráfica para o computador para comunicar com o microcontrolador através do bluetooth (tipo de comunicação RS232). No microcontrolador foi desenvolvido um programa para a leitura de mensagens, pois as mensagens provenientes do Movicon vinham em hexadecimal e numa longa mensagem apresentada no protocolo Modbus.

A interface gráfica interage com o sistema remoto, nomeadamente definir o *setpoint* pretendido, um alarme caso a temperatura ultrapasse 5°C do valor do *setpoint*, pois o máximo *overshoot* observado foi de 3°C a 5°C e um gráfico intuitivo na qual regista o valor do *setpoint* assim como a subida da temperatura ao longo do tempo, presente na Figura 5.5.

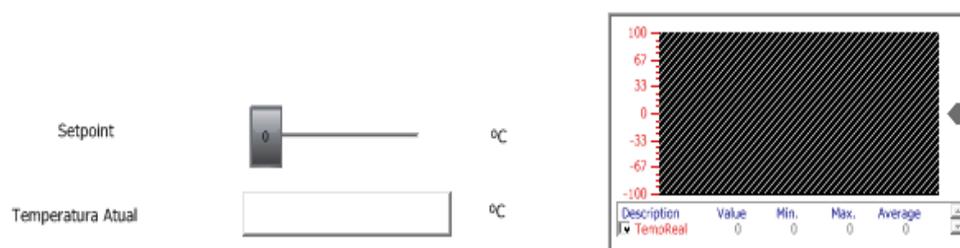


Figura 5.5 Interface Movicon

### 5.3. Análise de resultados

Neste ponto serão efetuados os testes com diferentes algoritmos num protótipo para a domótica apenas controlando a corrente elétrica fornecida à resistência elétrica de 750W de um reservatório de água.

Para efetuar o controlo da temperatura, amenizar o *overshoot* e o consumo de energia foram usados os algoritmos PID, lógica difusa e de controlo ON/OFF sendo os resultados comparados. A análise de controlo é feita de maneira a diminuir o overshoot e manter a temperatura no *setpoint* pretendido, definido anteriormente pelo utilizador. O sistema central apenas é responsável pela monitorização da temperatura e definição do *setpoint*. Com este protótipo podemos adquirir dados para uma melhor interpretação dos algoritmos implementados.

#### 5.3.1. Controlador ON/OFF

Conforme o fluxograma da Figura 5.6, depois de introduzido o *setpoint* no sistema central, superior à temperatura atual, o controlador responsabiliza-se por ligar a resistência do reservatório de água e proceder ao aquecimento do sistema até atingir o *setpoint*. Caso o *setpoint* seja inferior à temperatura atual o sistema mantém-se desligado, sem fornecer qualquer potência ao reservatório de água.

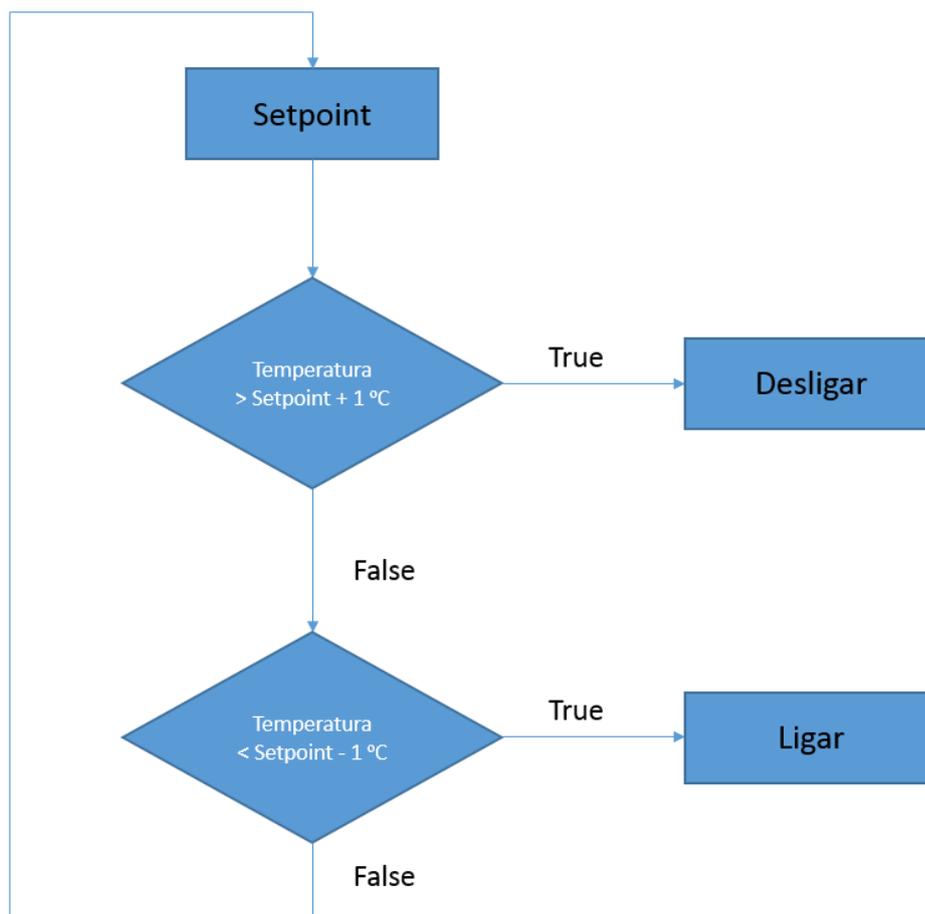


Figura 5.6 Fluxograma do controlo ON/OFF

De facto a temperatura excedeu o setpoint, existe um *overshoot* e a temperatura oscila entre 71°C e 69°C obtendo-se assim a curva apresentada na Figura 5.7. Caso a temperatura atual seja inferior ao *setpoint* o sistema mantém-se desligado. Neste tipo de controlo obteve-se um *overshoot* de 3,5°C e manteve-se instável ao longo do *setpoint*.

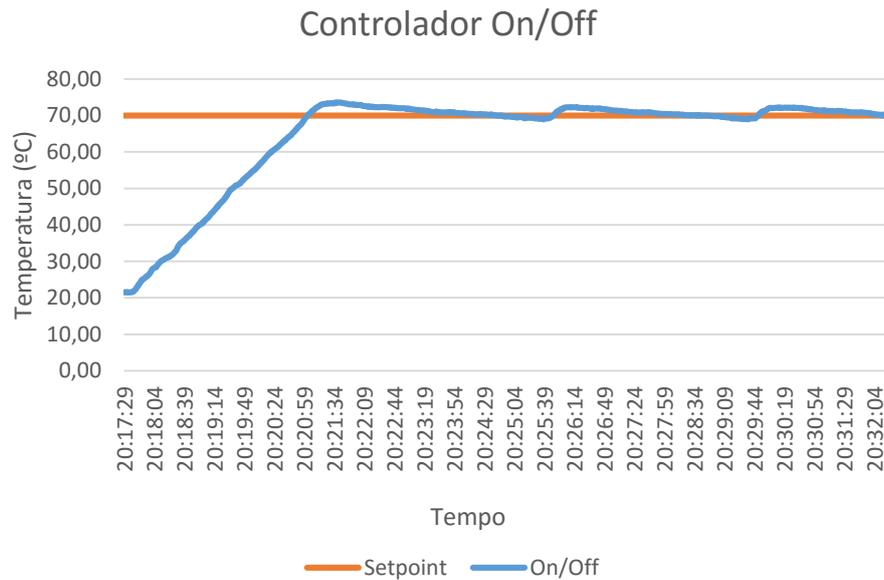


Figura 5.7 Controlador On / Off

### 5.3.2. Controlador PID

No controlador PID, é necessário ajustar os ganhos  $K_p, K_i, K_d$  nomeadamente o ganho proporcional, integrativo e derivativo respetivamente. O algoritmo PID usa a diferença entre o *setpoint* e a temperatura atual e esse erro é depois introduzido nos cálculos do sistema, representado na Figura 5.8.

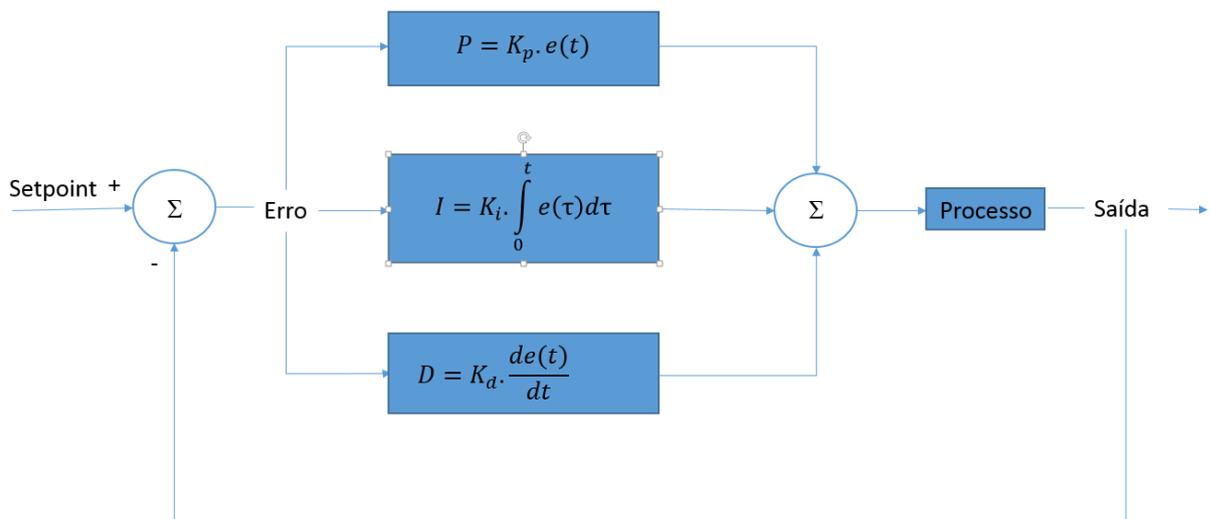


Figura 5.8 Processo PID

A parte proporcional (P) irá controlar a rapidez do sistema a atingir o *setpoint*, existindo um maior *overshoot* conforme o aumento do ganho proporcional. A parte integrativa (I) aumenta o tempo, de a resposta se manter estável no *setpoint* (*settling time*) mas elimina o *steady-state error*. A parte derivativa (D) vai diminuir o *overshoot* do sistema e vai manter-se constante no *setpoint* mais rápido, sendo necessário definir bem os parâmetros para obtenção de uma boa resposta sem muito *overshoot*, rápida e estável ao longo do *setpoint*, Figura 5.9. A saída deste controlador corresponde ao valor de *duty cycle* de uma atuação PWM.

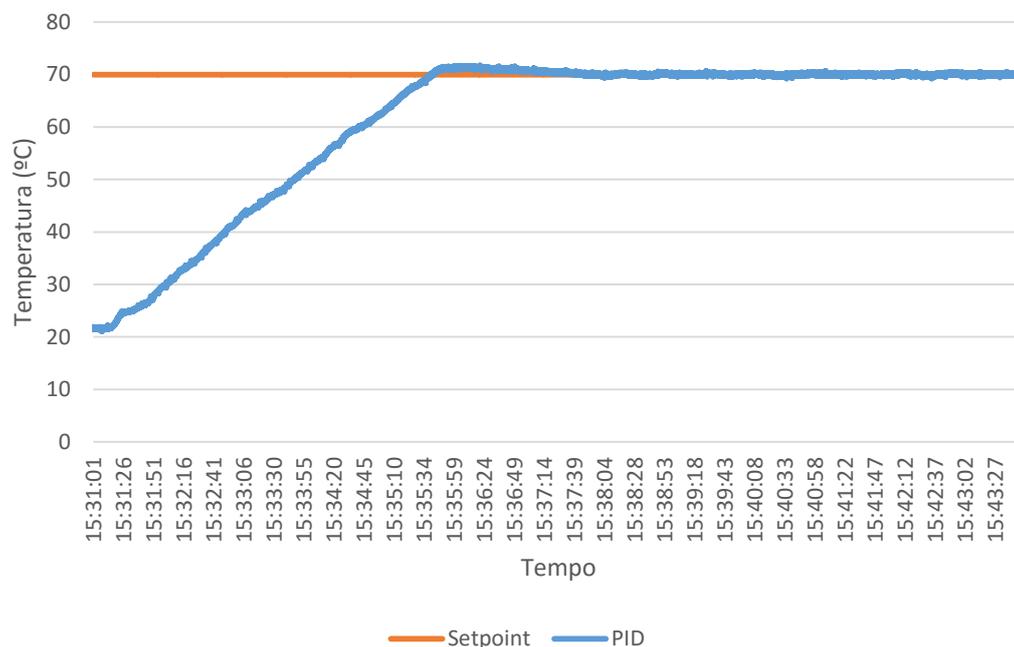


Figura 5.9 Controlador PID

Neste caso foi utilizado o método prático, que depois de alguns testes obteve-se um  $K_p=5$ ,  $K_i=4$  e um  $K_d=10$  apresentando esta resposta satisfatória presente na Figura 5.9, com um *overshoot* de apenas  $1^\circ\text{C}$  e mantendo-se constante no *setpoint* ao longo do tempo.

### 5.3.3. Controlador lógica difusa

Este tipo de controlador passa por 3 fases, a fuzzificação que define os parâmetros de entrada, os mecanismos de inferência onde são aplicadas as regras difusas e os seus graus de associação, Figura 5.3.3, e por último a defuzzificação para obter valores na saída correspondente ao *duty cycle*.

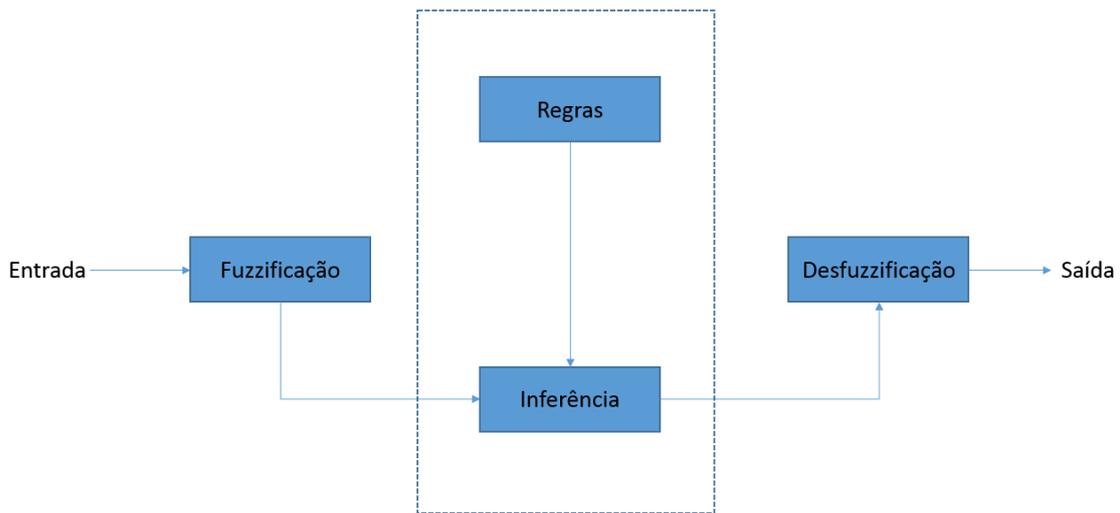


Figura 5.10 Diagrama de blocos da Lógica Difusa

## Fuzzificação

Nesta primeira fase define-se os quatro termos difusos (FRACO, MEIO, BOM e MAX), Figura 5.11, usados para aplicar na entrada que representa a diferença entre o *setpoint* e a temperatura do sistema. Foram usadas funções triangulares para o bom desempenho do controlador em que para erros superiores a 5°C o *duty cycle* seria de 100% comportando-se como um On/Off até atingir o *setpoint*, erros abaixo desse valor iriam sofrer *duty cycle* de diferentes proporções.

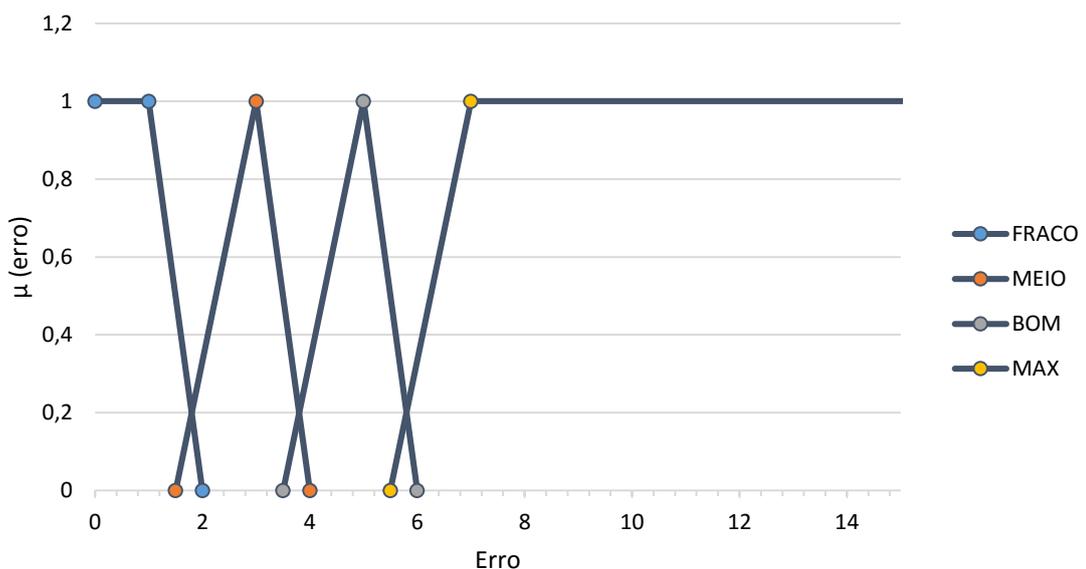


Figura 5.11 Graus de associação

## Mecanismos de inferência

Nesta fase são definidas as regras difusas que se baseiam apenas na regra Se-Então, conforme a saída de defuzzificação:

- **Se** Erro = FRACO **Então** *Duty Cycle* = baixoduty
- **Se** Erro = MEIO **Então** *Duty Cycle* = halfduty
- **Se** Erro = BOM **Então** *Duty Cycle* = duty
- **Se** Erro = MAX **Então** *Duty Cycle* = duty100

Nomeadamente o baixoduty, halfduty, duty e duty100 representam 25%, 50%, 75% e 100% *duty cycle* respetivamente. Quando o erro é inferior a zero o sistema mantém-se desligado.

## Defuzzificação

Por fim através da defuzzificação usando o método do centro de gravidade são obtidos valores de saída respetivos ao *duty cycle*. Obtendo assim o gráfico na Figura 5.12:

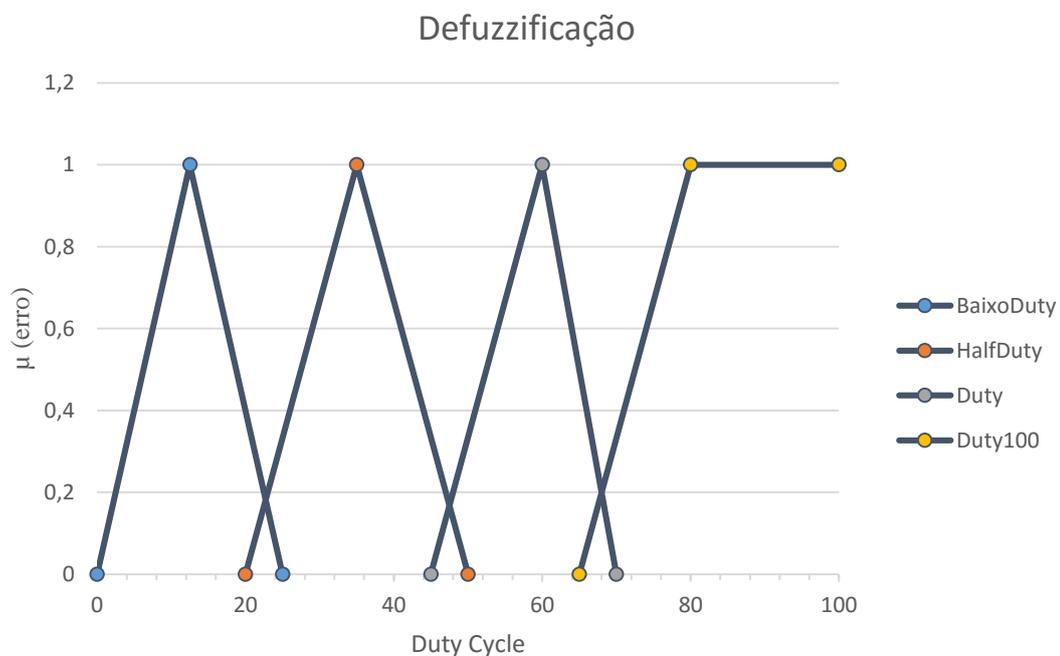


Figura 5.12 Grau de Defuzzificação

Com os 3 passos completos a saída será a apresentada na Figura 5.13

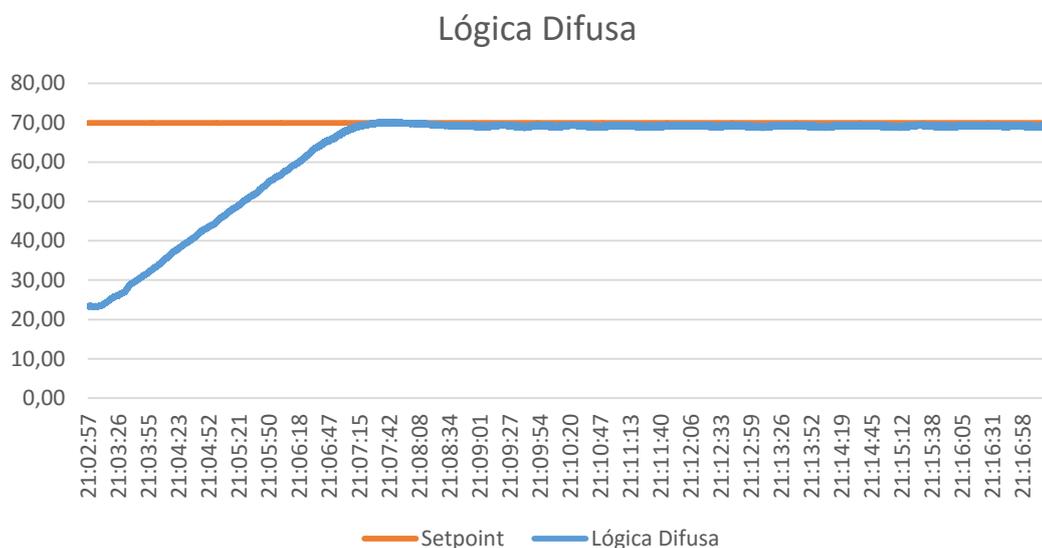


Figura 5.13 Controlador de Lógica Difusa

## 5.4. Comparação dos algoritmos

A comparação dos três tipos de algoritmos tem como objetivo minimizar o consumo de energia, assim como obter uma temperatura estável ao longo do tempo no *setpoint* definido. Como se pode verificar o controlo ON/OFF tem um controlo brusco, aquecendo mais rápido pois a resistência está com a sua potência máxima, atingindo assim o seu *setpoint* e havendo um *overshoot* de 3,5°C do *setpoint* pretendido, originando assim num pequeno desperdício de energia, e oscilando bastante no *setpoint* devido ao intervalo de 1°C definido no algoritmo. Para melhorar o *overshoot* e a estabilidade da temperatura recorreu-se ao PID e Logica Fuzzy.

No PID existe um *overshoot* de 1,5°C, podendo ser atenuado com alterações nas variáveis, e apresenta um tempo de subida mais lento que o controlo On/Off mas uma maior estabilidade a manter-se no *setpoint*.

O controlo por Lógica difusa apresentou melhores resultados, com um tempo de subida mais rápido que o PID e sem a presença de *overshoot* e uma estabilidade semelhante ao PID, apresentado na Figura 5.14.

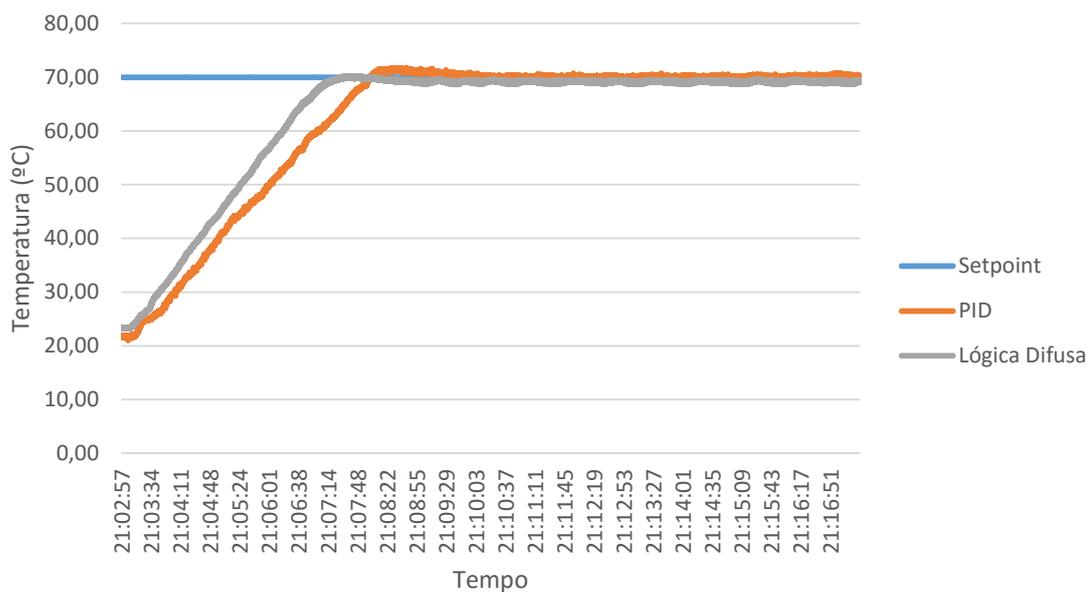


Figura 5.14 Gráfico de Comparação dos Algoritmos



# Capítulo 6

## 6. Conclusão

Na realização deste trabalho pode-se afirmar que os objetivos de criar um sistema de supervisão e o controlo otimizado dos dispositivos foram cumpridos.

Nomeadamente, para atingir estes objetivos foi criado um sistema centralizado, usando o *smartphone* e o Movicon, usando redes de comunicação RF, como o bluetooth, e efetuando o controlo rigoroso através de algoritmos de controlo presente no microcontrolador do sistema de controlo.

A utilização de um o *smartphone* com Bluetooth mostrou-se ser uma boa opção porque muitos utilizadores já possuem *smartphones* deste tipo facilitando assim o controlo do sistema de aquecimento e baixando os custos. No entanto com a ferramenta poderosa Movicon, muito utilizada na indústria, permitiu a facilidade da criação de uma interface intuitiva e com enormes possibilidades da criação de interfaces para outros dispositivos além do computador.

Na aquisição de temperatura o sensor mostrou-se linear mas com alguns contra no sentido de ser necessário uma tensão de entrada constante pois mudaria os valores de saída em tensão do sensor, assim como tensões parasitas na saída do amp-op.

O módulo Bluetooth é uma tecnologia viável para o projeto, permitindo boas taxas de comunicação, baixo custo embora só permita um número limitado de dispositivos ligado em simultâneo.

Constatou-se que os algoritmos de controlo permitiram estabilizar a temperatura no setpoint havendo assim uma maior eficiência energética. No entanto estes resultados foram apenas testados num reservatório de água, sendo necessário a sua implementação numa situação real, numa residência. Contudo os algoritmos de lógica difusa apresentou uma melhor resposta neste sistema de aquecimento em particular.

## 7. Trabalho Futuro

Com o desenvolvimento desta plataforma de domótica pretende-se criar um sistema capaz de efetuar o controlo do aquecimento numa habitação. O recurso a novos algoritmos de controlo, por exemplo de algoritmos genéticos e redes neuronais, permitiram obter um maior leque de resultados e comparar a diferença em diferentes ambientes de controlo. Também a criação de um algoritmo automático de *tuning*, no controlo PID seria uma boa opção já que se mostrou um pouco trabalhoso o *tuning* dos ganhos do PID.

O desenho de uma placa PCB seria um ponto forte a desenvolver futuramente para diminuir o tamanho da eletrónica e obter leituras mais rigorosas, devido a pouco ruído eletrónico e uma maior estabilidade nos componentes quando a movimentos da placa.

O desenvolvimento de uma aplicação android mais criativa e intuitiva através das potencialidades do Movicon, pois apenas foi utilizado, com sucesso, uma aplicação simples já desenvolvida e presente na *Google store*.

# Referências

- [1] D. L. Cardoso, “Domótica Inteligente - Um Contributo Prático,”, Dissertação de Mestrado, UTAD, Vila Real,PT, 2009.
- [2] Iberdrola, “Com Ihos de Poupança,” 2015. [Online]. Disponível em: <http://www.iberdrola.pt/02sicb/corporativa/iberdrola?IDPAG=PTWCOABOE NESAV&codCache=14374953417455092#5>. [Acedido em: 28-Jul-2015].
- [3] EuroX10, “LK15 - Pack Iniciação X10.” [Online]. Disponível em: <http://www.eurox10.com/Product/EasyX10/Packs/LK15.htm>. [Acedido em: 16-May-2015].
- [4] EuroX10, “O que é o X10?” [Online]. Disponível em: <http://www.eurox10.com/Content/X10Information.htm>. [Acedido em: 16-May-2015].
- [5] E. X10, “Teoria da transmissão de sinais X-10.” [Online]. Disponível em: <http://www.eurox10.com/Content/X10SignalTheory.htm>. [Acedido em: 10-Nov-2015].
- [6] KNX, “Main Advantages,” 2014. [Online]. Disponível em: <http://www.knx.org/knx-en/knx/benefits-advantages/main-advantages/index.php>. [Acedido: 05-Nov-2015].
- [7] KNX, “O que é o KNX?,” 2013. [Online]. Disponível em: <http://www.knx.org/pt/knx/o-que-e-knx/>. [Acedido: 17-May-2015].
- [8] KNX, “KNX System Specifications,” p. 26, 2013.
- [9] S. Buddle, “Best Practice: Educating M&E Consultants About KNX Integration,” 2015. [Online]. Disponível em: <http://knxtoday.com/2015/01/5297/best-practice-educating-me-consultants-about-knx-integration.html>.
- [10] Niessen, “Novo Catálogo Geral de Aparelhagem Terminal ABB Niessen para 2008,” 2008.

- [11] KNX and ABB, “KNX Technical Reference Manual KNX Sensors KNX Technical Reference Manual,” 2011.
- [12] C. I. Systems, “What is Clipsal C-Bus?” [Online]. Disponível em: <http://www.cbus-enabled.com/what-cbus.htm>. [Acedido: 17-May-2015].
- [13] Clipsal, “C-Bus® Basic Training Manual,” vol. 2, pp. 1–64, 2007.
- [14] C-BUS, “C-Bus™ Thermostats.”, Training Guide, Schneider Electric, 2009.
- [15] T. Abreu, “Edifícios Inteligentes – Soluções para gestão de climatização em instalação de Domótica KNX Estudo de um Caso,” Dissertação de Mestrado, IPB, Bragança,PT, 2013.
- [16] A. M. A. Ferreira, “Desenvolvimento de infra-estrutura de comando multifunções EIB-KNX para smartphone,” Dissertação de Mestrado, IPB, Bragança,PT, 2012.
- [17] I. Electronics, “Comparison of Temperature.” [Online]. Disponível em: [http://www.industrial-electronics.com/DAQ/industrial\\_electronics/input\\_devices\\_sensors\\_transducers\\_transmitters\\_measurement/Comparison\\_Temperature\\_Sensors.html](http://www.industrial-electronics.com/DAQ/industrial_electronics/input_devices_sensors_transducers_transmitters_measurement/Comparison_Temperature_Sensors.html). [Acedido em: 30-Jul-2015].
- [18] D. P. Jones, *Biomedical Sensors*. New York: Momentum Press, LLC, 2010.
- [19] Heidolph, “Sensor de temperatura Pt 1000 – acero inoxidable.” .
- [20] T. Instruments, “LM35 Precision Centigrade Temperature Sensors,” 2015.
- [21] T. Instruments, “LM35,” 2015. [Online]. Disponível em: <http://www.ti.com/product/LM35>. [Acedido: 21-Oct-2015].
- [22] T. P. Wang, “Thermocouple Materials,” *ASM Handbook*, vol. 2, 1990.
- [23] A. Baratella and D. Sobrinho, “Tecnologias wireless para automação industrial: Wireless Hart, Bluetooth, Wisa, Wifi, zigbee e SP-100.”
- [24] ModBus, “MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b,”

- 2006.
- [25] L. Miúdo, “Automação distribuída com protocolos CAN e Modbus,” Dissertação de Mestrado, UNL, Lisboa, PT, 2014.
- [26] J. Riço, “Módulo sem fios para controlo de temperatura de uma habitação,” Dissertação de Mestrado, DEM, UA, Aveiro, PT, 2014.
- [27] F. de M. Maciel, “CIRCUITO PARA REGULAGEM DE POTÊNCIA UTILIZANDO TIRISTOR TRIAC E MICROCONTROLADOR 8051,” CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB, 2010.
- [28] R. Keim, “Ambient Light Monitor: Using a Triac to Adjust Lamp Brightness,” 2015. [Online]. Disponível em: <http://www.allaboutcircuits.com/projects/ambient-light-monitor-using-a-triac-to-adjust-lamp-brightness/>. [Acedido em: 06-Dec-2015].
- [29] C. Serôdio, P. Mestre, and J. B. Cunha, *Eletrónica de Potência Part2 - Comportamento térmico e dispositivos de rectificação controlada*. Vila Real: SDB, 2010.
- [30] T. Hirzel, “PWM.” [Online]. Disponível em: <https://www.arduino.cc/en/Tutorial/PWM>. [Acedido em: 28-Oct-2015].
- [31] J. Silva, “Ação Liga-Desliga ( On-off),” 2000. [Online]. Disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node20.html>. [Acedido em: 28-Oct-2015].
- [32] K. Ogata, *Modern Control Engineering*, Fourth Edi. Aeeizh, 2002.
- [33] K. J. Åström, “Control System Design,” 2002.
- [34] U. O. Michigan, “Introduction: PID Controller Design.” [Online]. Disponível em: <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID>. [Acedido em: 26-Oct-2015].
- [35] J. Bennett, A. Bhasin, J. Grant, and W. C. Lim, “PIDTuningClassical,” 2006. [Online]. Disponível em: <https://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical#Process>

- \_Reaction\_Curve. [Acedido em: 26-Oct-2015].
- [36] N. Schulze, “Fuzzy Logic : Introduction,” 2010. .
- [37] J. M. Mendel, “Fuzzy Logic Systems for Engineering : A tutorial,” *Ieee*, vol. 83, no. 3, pp. 345–377, 1995.
- [38] R. C. Prati, “Sistemas Fuzzy.”
- [39] Revolution, “ALPHANUMERIC LCD DISPLAY (16 X 2) datasheet.”

# Anexos

# **Anexo 1 - Circuito Eléctrico**

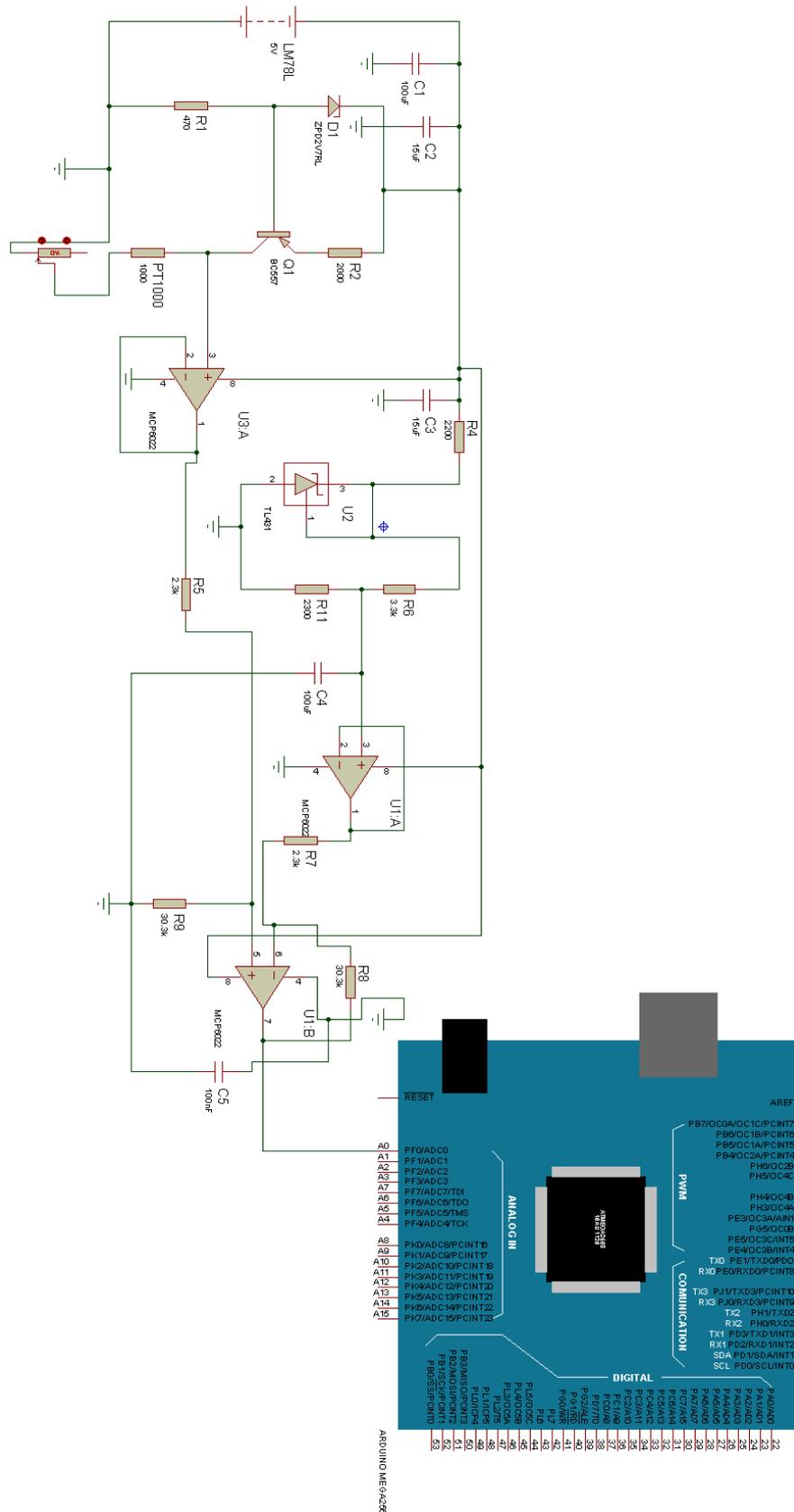


Figura 15 Circuito de Leitura

## **Anexo 2 - LCD**

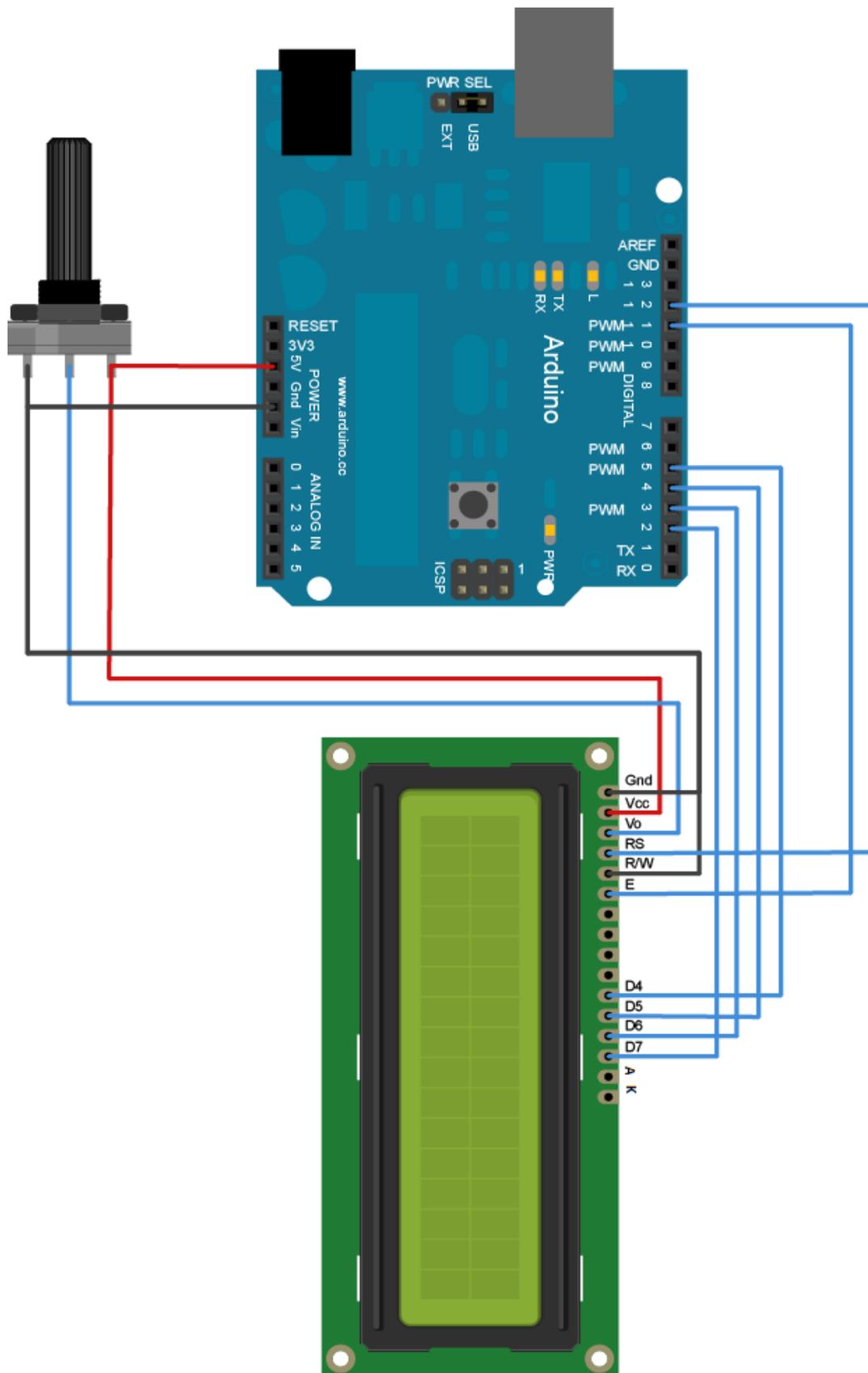


Figura 16 Ligações LCD

## **Anexo 3 - Bluetooth**

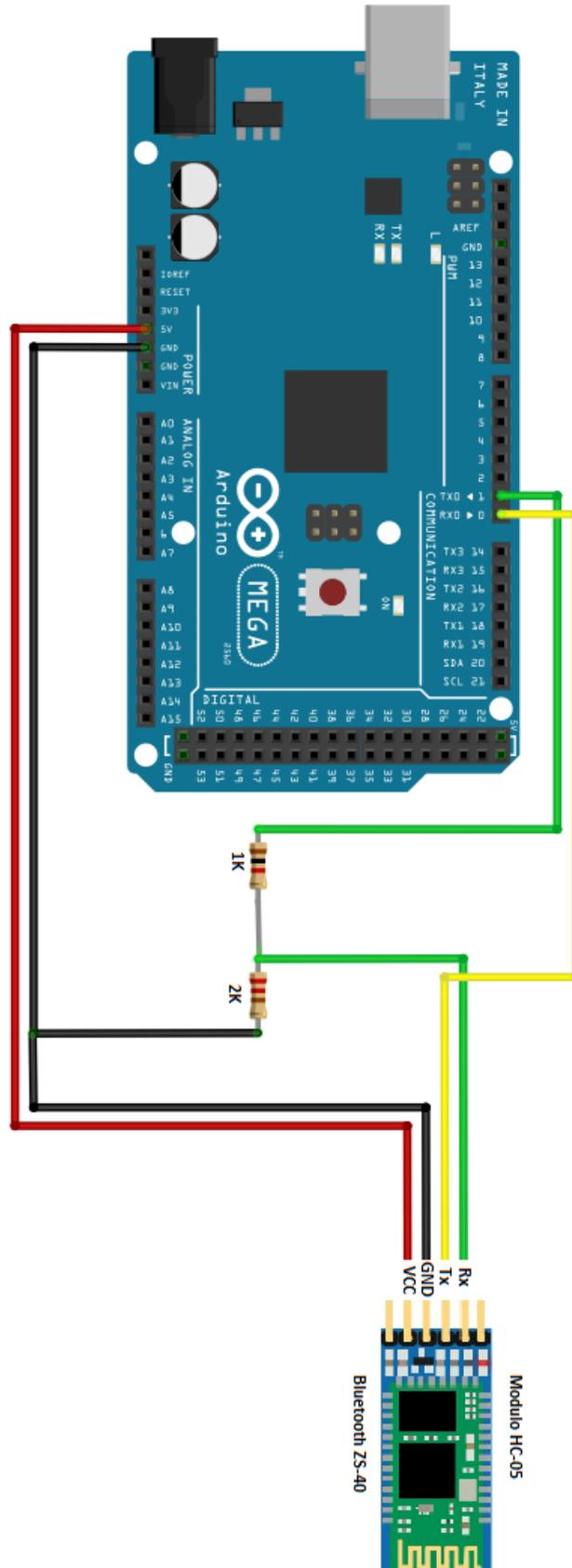


Figura 17 Ligações Bluetooth