



Sistema de monitorização e controlo de carga centralizado

SANDRO DANIEL MOREIRA DE CARVALHO

novembro de 2023

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistema de monitorização e controlo de carga centralizado

Sandro Daniel Moreira de Carvalho

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Automação e Sistemas



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Novembro, 2023

Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.

Candidato: Sandro Daniel Moreira de Carvalho, N.º 1151724,
1151724@isep.ipp.pt

Orientação Científica: Lino Figueiredo, lbf@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Novembro, 2023

Dedico este projeto à minha mãe Maria Rosária Devesa Moreira Pereira.

Agradecimentos

Quero agradecer ao Dr. Eng. Lino Figueiredo pelo apoio dado durante a realização deste projeto.

Resumo

Este projeto tem como objetivo fazer o estudo, desenvolvimento e demonstração de um sistema de monitorização e controlo de carga centralizado utilizando o conceito *Internet of Things* (IoT).

Para servir de exemplo a este projeto foram estudados vários produtos onde implementam o conceito IoT e/ou a tecnologia de monitorização de consumo de energia, como por exemplo a FIBARO, Eyedro, Lumin, Leviton e Trotec. Este estudo permitiu idealizar e conceber um sistema preparado para monitorizar os dados de energia elétrica e controlo de cargas e exibir esses dados numa interface gráfica.

O sistema de monitorização tem como objetivo realizar a medição da tensão eficaz, corrente eficaz, potência instantânea, fator de potência, energia instantânea e acumulada, sendo também possível verificar o custo monetário associado à energia utilizada. O sistema de monitorização envolve o estudo e implementação sobre a alimentação do sistema, condicionamento de sinal, *analog-to-digital converter* (ADC) externo, placa de desenvolvimento ESP32 DevKitC V4 e o display *Organic Light-Emitting Diode* (OLED). O sistema de controlo por sua vez tem como objetivo acionar um *light-emitting diode* (LED) e é realizado através da placa de desenvolvimento Mini WEMO D1 ESP32.

Para aceder à informação dos dados de energia elétrica e realizar o controlo do LED foi desenvolvida uma página de Internet baseada na plataforma Node-RED. Para desenvolver a plataforma utilizou-se um servidor que irá correr a plataforma Node-RED e disponibilizar o endereço *Internet Protocol* (IP) da página de monitorização e controlo. Os dados transmitidos entre os microcontroladores e servidor utilizam o protocolo de comunicação *Message Queuing Telemetry Transport* (MQTT).

Por fim, foram realizados testes ao sistema tendo como base de comparação os dados de energia resultantes do dispositivo de medição de energia Aubess. Os resultados obtidos do dispositivo Aubess e do nosso sistema são idênticos, concluindo assim que o sistema de monitorização desenvolvido funciona de acordo com o esperado. A implementação da placa de desenvolvimento Mini WEMO D1 ESP32 também demonstrou estar corretamente implementada na rede local sendo possível

alterar o estado do LED entre ligado e desligado.

Palavras-Chave: Resistência *shunt*, MQTT, WIFI, ESP32, Raspberry Pi 3, display OLED, NODE-Red.

Abstract

This project aims to study, develop and demonstrate a centralized load monitoring and control system using the IoT concept.

To use as an example several products were studied where they implement the IoT concept and/or energy consumption monitoring technology such as FIBARO, Eyedro, Lumin, Leviton and Trotec. This study allowed to idealize and design a system prepared to monitor the data of electric energy and load control and display these data in a graphical interface.

The monitoring system aims to measure the power factor, instantaneous and accumulated energy, instantaneous power, *Root Mean Square* (RMS) voltage and RMS current, also being possible to check the cost associated to the energy used. The monitoring system embraces the study and implementation of the system power, signal conditioning, external ADC, ESP32 DevKitC V4 development board and the OLED display. The control system, in turn, aims to activate a LED and it is carried out through the Mini WEMO D1 ESP32 development board.

To access the electrical energy data information and carry out the LED control an Internet page based on the Node-RED platform was developed. To develop the platform a server was used that will run the Node-RED platform and provide the IP address of the monitoring and control page. The data transmitted between microcontrollers and server uses the MQTT communication protocol.

Finally, system tests were carried out based on the energy data resulting from the Aubess energy measurement device. The results obtained from the Aubess device and from our developed device are identical, thereby concluding that the system works as expected. The implementation of the Mini WEMOD1 ESP32 development kit also proved to be correctly implemented in the local network, being possible to change the state of the LED between on and off.

Keywords: *Shunt* resistor, MQTT, WIFI, ESP32, Raspberry Pi 3, OLED display, NODE-Red.

Índice

Lista de Figuras	ix
Lista de Tabelas	xiii
Listagens	xv
Lista de Acrónimos	xvii
1 Introdução	1
1.1 Contextualização	2
1.2 Definição do Problema	3
1.2.1 Objetivos	3
1.2.2 Resultados esperados	3
1.3 Plano de Trabalho	4
1.4 Organização da Dissertação	5
2 Estado de arte	7
2.1 Dispositivos de medição de energia elétrica para uso doméstico	8
2.1.1 FIBARO	9
2.1.2 Eyedro	11
2.1.3 Lumin	12
2.1.4 Leviton	14
2.1.5 Trotec	16
2.2 Sensores de corrente	17
2.2.1 <i>Shunt</i>	17
2.2.2 Sensor de efeito Hall	19
2.2.3 Transformadores de corrente	21
2.3 Estudo e comportamento da energia em diferentes cargas	22
3 Protocolos de comunicação IoT	27
3.1 Internet Of Things	27
3.2 Protocolos da camada física	29
3.2.1 LoRaWAN	29
3.2.2 <i>Bluetooth</i>	30

3.2.3	WiFi	31
3.2.4	ZigBee	32
3.3	Protocolos da camada de aplicação	33
3.3.1	Message Queuing Telemetry Transport - MQTT	33
3.3.2	Constrained Application Protocol	35
4	Arquitetura do sistema	37
4.1	Arquitetura do sistema	38
4.2	Alimentação do sistema	38
4.3	Sensor e condicionamento de sinal	39
4.4	Conversor analógico digital	42
4.4.1	Captura do sinal analógico	43
4.4.2	Conversor analógico digital - ESP32 vs MCP3002	44
4.5	Microcontrolador	46
4.5.1	ESP32 DevKitC V4	47
4.5.2	Mini WEMO D1 ESP32	48
4.6	Display	48
4.7	Protocolo de Comunicação de dados - I2C, SPI e WIFI	49
4.7.1	Protocolo de Comunicação SPI - Comunicação entre ADC e microcontrolador	49
4.7.2	Protocolo de Comunicação I2C - Comunicação entre microcontrolador e display	53
4.7.3	Protocolo de Comunicação WIFI e MQTT - Comunicação entre microcontrolador e servidor	55
4.8	Servidor	58
4.9	Node-RED	59
4.10	Cuidados a ter no desenvolvimento da PCB	61
5	Implementação dos sistemas de Hardware e Software	65
5.1	Implementação do <i>Hardware</i>	66
5.1.1	Implementação da Alimentação do Sistema	68
5.1.2	Implementação do Condicionamento de sinal e sensores	69
5.1.3	Implementação do ADC e microcontrolador	70
5.1.4	Implementação do display e servidor	71
5.1.5	Implementação da PCB	72
5.2	Implementação do <i>Software</i>	73
5.2.1	Implementação da comunicação com o ADC utilizando o protocolo de comunicação SPI	73
5.2.2	Processamento dos dados capturados	78
5.2.3	Implementação da comunicação com o display utilizando o protocolo de comunicação I2C	84

5.2.4	Implementação da comunicação com a rede de internet utilizando o módulo de comunicação WIFI	87
5.2.5	Implementação da página web utilizando o Node-RED	89
6	Testes aos sistemas de Hardware e Software	93
6.1	Testes ao sistema de <i>Hardware</i> e <i>Software</i>	93
6.1.1	Testes ao <i>Hardware</i>	95
6.1.2	Testes ao <i>Software</i>	99
6.1.3	Testes ao Sistema de monitorização e controlo de carga centralizado	102
7	Conclusões	107
7.1	Trabalho Futuro	108
	Referências	109
	Anexo A Estimativa do custo do projeto	117

Lista de Figuras

1.1	Principais indicadores para alcançar as metas do PNEC para 2030 [3].	2
1.2	Calendarização do projeto	4
2.1	Rede comunicação de tecnologia <i>Z-Wave</i> [9]	9
2.2	Tomada de parede com monitorização do consumo de energia [10] . .	10
2.3	Monitorização do consumo de energia apresentado na aplicação móvel Fibaro [10]	10
2.4	Rede de comunicações do sistema eyedro [12]	11
2.5	Painel com dados do consumo energético acessível através da aplica- ção <i>web</i> MyEyedro [13]	12
2.6	Rede de comunicações do sistema eyedro [12]	12
2.7	Sistema de monitorização Lumin [14]	13
2.8	Representação dos dois módulos essenciais no sistema Lumin [15]. . .	13
2.9	Sistema de monitorização Lumin [16]	14
2.10	Interface gráfica utilizada pelo sistema Lumin [16].	14
2.11	Composição do sistema de disjuntor inteligente [17]	15
2.12	Interface gráfica Leviton [17]	16
2.13	Quadro elétrico inteligente Leviton [17]	16
2.14	Aparelho de medição de energia elétrico simples [18]	17
2.15	Aplicação de <i>shunt</i> no circuito elétrico [20].	18
2.16	Resistência <i>shunt</i> ,para medição de correntes até 100A [22].	18
2.17	Resistência <i>shunt</i> ,para medição de correntes até 80A [23].	19
2.18	Resistência <i>shunt</i> ,do tipo <i>surface-mount devide</i> (SMD) [24].	19
2.19	Princípio de funcionamento do sensores de efeito Hall [25].	20
2.20	Sensores de efeito Hall com diferentes encapsulamentos [27].	20
2.21	Sensor de ACS711EX, [28].	21
2.22	Princípio de funcionamento de um transformador de corrente [30]. .	21
2.23	Transformador de corrente [31].	22
2.24	Forma de onda sinusoidal.	23
2.25	Potência consumida por uma carga resistiva [34].	24
2.26	Representação da potência em cargas parcialmente reativas [34]. . .	24
2.27	Triângulo das potências [35].	25

3.1	Estimativa do valor económico em mil milhões de dólares para o ano 2030 na adoção da <i>Internet of Things</i> (IoT) em nove áreas distintas [39].	28
3.2	Arquitetura de rede LoRaWAN [40].	30
3.3	Versões Wi-Fi [47].	32
3.4	Clientes subscritores e publicadores [51].	34
3.5	Arquitetura do protocolo MQTT [52].	34
3.6	Princípio de funcionamento do protocolo <i>Message Queuing Telemetry Transport</i> (MQTT) [53].	35
3.7	Princípio de funcionamento do protocolo <i>Constrained Application Protocol</i> (CoAP) [55]	36
4.1	Arquitetura do sistema.	38
4.2	Implementação da resistência <i>shunt</i> no circuito de carga e respetiva relação tensão/corrente.	40
4.3	Isolamento do circuito de potência com circuito de carga e respetivo condicionamento de sinal	41
4.4	Condicionamento de sinal e seu resultado para tensão da rede elétrica	42
4.5	Tensão à saída do componente ACPL-785J [58].	42
4.6	Comparação entre o ADC do ESP32 e o ADC MCP3002.	44
4.7	Definição dos pinos do <i>analog-to-digital converter</i> (ADC) MCP3002 [61].	45
4.8	Características da taxa de conversão do MCP3002 [61].	46
4.9	Placa de desenvolvimento ESP32-DevKitC V4.	47
4.10	Placa de desenvolvimento mini WEMO D1 ESP32 [63].	48
4.11	OLED display 0.91"[64].	49
4.12	Protocolo de comunicação <i>Serial Peripheral Interface</i> (SPI) com comunicação <i>full-duplex/half-duplex</i> [65].	50
4.13	Configuração de bits do ADC MCP3002 [61].	51
4.14	Comunicação SPI com o ADC MCP3002 [61].	52
4.15	Considerações de tempo no protocolo SPI [66].	52
4.16	Formato de dados do protocolo <i>Inter-Integrated Circuit</i> (I2C) [67].	54
4.17	Gráfico da transmissão completa utilizando o protocolo de comunicação I2C [68].	54
4.18	Definição da condição de confirmação [67].	55
4.19	Microcontrolador com módulo <i>Wireless Fidelity</i> (WIFI) definido no modo estação [69].	56
4.20	Arquitetura da comunicação.	57
4.21	Raspberry Pi 3 modelo B+.	59
4.22	Linguagem de programação Node-RED [72].	60

4.23	Nó de subscrição de tópico [72].	60
4.24	Parametrização do nó de subscrição de tópico [72].	61
4.25	Parametrização do nó de publicação de tópico [72].	61
4.26	Exemplo da aplicação de circuito analógico e o ruído associado [73].	62
4.27	Implementação da alimentação e terra numa <i>Printed Circuit Board</i> (PCB) [73].	63
4.28	Exemplo da correta implementação de condensadores de desacopla- mento [73].	64
4.29	Localização dos componentes na PCB [73].	64
5.1	Sistema de monitorização de energia desconstruído em blocos funcionais.	66
5.2	Esquema elétrico - Circuito analógico.	67
5.3	Esquema elétrico - Circuito digital.	67
5.4	Esquema elétrico - segunda placa de desenvolvimento.	68
5.5	Representação a traço grosso do circuito de alimentação do sistema a 230VAC.	68
5.6	Representação a traço grosso do circuito de alimentação do sistema a 5VDC e a 3.3VDC.	69
5.7	Representação a traço grosso do circuito elétrico do condicionamento de sinal e sensores.	70
5.8	Representação a traço grosso do circuito elétrico do ADC e microcon- trolador.	71
5.9	Representação a traço grosso do circuito elétrico do display.	71
5.10	Vista de cima da placa de circuito impresso.	72
5.11	Vista de baixo da placa de circuito impresso.	73
5.12	Parâmetros de inicialização do ADC e protocolo de comunicação SPI.	74
5.13	Comunicação SPI com MCP3002 utilizando segmentos de 8 bits, no modo <i>Single-Ended</i> [61]	75
5.14	Fluxograma da interrupção interna que ocorre numa frequência de 10KHz	78
5.15	Fluxograma do tratamento das amostras de dados	80
5.16	Parâmetros de inicialização do display e protocolo de comunicação I2C	84
5.17	Comunicação <i>publish/subscriber</i> entre microcontrolador e página <i>web</i> .	88
5.18	Fluxograma da implementação da página gráfica.	90
5.19	Configuração do nó mqtt in.	90
5.20	Configuração do nó chart.	91
6.1	Exemplo de testes realizado ao bloco funcional isolado e integrado. .	94
6.2	Sinal de tensão medido à entrada do sensor de tensão.	95
6.3	Sinal de tensão medido à entrada do sensor de corrente.	96
6.4	Sinal de tensão à saída do sensor de tensão.	97

6.5	Sinal de tensão à saída do sensor de corrente sem carga ativa.	97
6.6	Sinal de tensão à saída do sensor de corrente com carga ativa.	98
6.7	Sinal de tensão à saída com carga associada.	99
6.8	Verificação da frequência de interrupção.	99
6.9	Exibição dos dados obtidos através do ADC.	100
6.10	Dados de energia apresentados no display com carga ativa.	100
6.11	Testes à comunicação de dados entre o ESP32 e o Raspberry Pi . . .	101
6.12	Dados de energia apresentados na página WEB com carga ativa. . .	101
6.13	Página de controlo.	102
6.14	Dados de energia apresentados na aplicação Tuya.	103
6.15	Página de Monitorização.	104
6.16	Página de Dados de energia completo.	104
6.17	Dados de energia apresentados no display.	105
6.18	Informação e controlo do estado do LED na página de controlo. . . .	105
6.19	Apresentação do estado do LED na placa de desenvolvimento.	105

Lista de Tabelas

4.1	Percentagem de erro relativa a diferentes frequências de amostragem.	43
4.2	Taxa de conversão por canal analógico.	46
5.1	Legenda das siglas das equações	83

Listagens

4.1	Inicialização da configuração do módulo wifi	56
4.2	Definição do modo wifi	56
4.3	Inicialização do módulo wifi	56
4.4	Conexão ao ponto de acesso da rede de internet	57
4.5	Identificador de cliente MQTT	57
4.6	Inicia o cliente MQTT com o identificador de cliente	58
4.7	Função para subscrever dados do tópico	58
4.8	Função para publicar dados no tópico	58
5.1	Instanciação do MCP3002	74
5.2	Configuração e comunicação com o MCP3002	76
5.3	Instanciação e leitura dos canais do ADC	77
5.4	Calculo da tensão e corrente eficaz potência aparente e real e fator de potência	82
5.5	Construção do objeto display utilizando a classe Adafruit_SSD1306	85
5.6	Construtor do objeto display na biblioteca Adafruit_SSD1306	85
5.7	Inicialização do objeto display na biblioteca Adafruit_SSD1306 . . .	85
5.8	Função para apresentar dados no display utilizando a biblioteca Ada- fruit_SSD1306 e Adafruit_GFX	86
5.9	Configuração do módulo wifi	87
5.10	Verificação da conexão à rede de Internet	87
5.11	Define o endereço do servidor com o qual o microcontrolador irá rea- lizar a transferência de dados	88
5.12	Transmissão da mensagem do ESP32 para o servidor	88

Lista de Acrónimos

AC	Alternating Current
ADC	<i>analog-to-digital converter</i>
API	<i>Application Programming Interface</i>
BBR	<i>Bluetooth Basic Rate</i>
BEDR	<i>Bluetooth Enhanced Data Rate</i>
BLE	<i>Bluetooth Low Energy</i>
CoAP	<i>Constrained Application Protocol</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Chip Select</i>
DAC	<i>digital-to-analog converter</i>
DC	Direct Current
EMI	<i>Electromagnetic Interference</i>
FSK	<i>frequency shifting keying</i>
GDDRAM	<i>Graphic Display Data RAM</i>
GND	<i>Ground</i>
GPIO	<i>General Purpose Input/Output</i>
HDMI	<i>High-Definition Multimedia Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IMC	Índice de Massa Corporal
IoT	<i>Internet of Things</i>

IP	<i>Internet Protocol</i>
LED	<i>light-emitting diode</i>
LoRa	<i>Long Range</i>
LoRaWAN	<i>Long Range Wide Area Network</i>
LPWAN	<i>Low Power Wide Area Network</i>
LSB	<i>Least Significant Bit</i>
M2M	<i>Machine to Machine</i>
MISO	<i>Master Input, Slave Out</i>
MOSI	<i>Master Output, Slave Input</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
MSB	<i>Most Significant Bit</i>
MSBF	<i>Most Significant Bit First</i>
OLED	<i>Organic Light-Emitting Diode</i>
PCB	<i>Printed Circuit Board</i>
PNEC	Plano Nacional de Energia e Clima
PSK	<i>Pre-Shared Key</i>
PWM	<i>Pulse Width Modulation</i>
QoS	Quality of Service
RAM	<i>Random Access Memory</i>
REST	<i>Representational State Transfer</i>
RMS	<i>Root Mean Square</i>
ROM	<i>Read-Only Memory</i>
RTC	<i>Real Time Clock</i>
RX	<i>Receive</i>
SA0	<i>Slave address bit</i>
SCL	<i>Serial Clock</i>

SCLK	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SMD	<i>surface-mount device</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
SSID	<i>Service Set Identifier</i>
SSL	<i>Secure Sockets Layer</i>
TCP	<i>Transmission Control Protocol</i>
TLS	<i>Transport Layer Security</i>
TX	<i>Transmit</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UDP	<i>User Datagram Protocol</i>
UE	União Europeia
VCC	<i>Voltage Common Collector</i>
WIFI	<i>Wireless Fidelity</i>
WPA	<i>Wi-Fi Protected Access</i>

Capítulo 1

Introdução

Atualmente a produção de eletricidade é obtida a partir de fontes de energia renováveis e da queima de combustíveis fósseis contribuindo este último de forma significativa para a emissão de gases poluentes. Estes gases poluentes têm um impacto severo para o meio ambiente, contribuindo para o agravamento do efeito de estufa e para a ocorrência de catástrofes naturais. Erros no armazenamento ou na sua extração também pode ter um impacto negativo no meio ambiente. De salientar que o aumento da poluição está relacionada com o aumento de problemas de saúde das pessoas. Para além disso, a elevada dependência destes recursos naturais não renováveis coloca em risco a produção de energia. Isto porque eles são extraídos a uma velocidade superior à da sua produção. Ou seja, a sua disponibilidade pode estar ameaçada.

De forma mitigar estes problemas a União Europeia (UE) traçou um plano intitulado Quadro Climático e Energético 2030 com diversos objetivos. Assim, tendo como referência os valores de 1990, e até 2030, os 28 países devem alcançar uma redução de 40% da emissão de gases com efeitos de estufa (prevê-se que consigam atingir uma redução de 55%). Estas medidas prevêm ainda, que 32% da energia seja proveniente de fontes de energia renováveis e uma melhoria na eficiência energética de, pelo menos, 32,5% [1]. A UE tem ainda como objetivo alcançar a neutralidade carbónica até ao ano de 2050.

Não diferindo do roteiro traçado pela UE, Portugal aprovou o Plano Nacional de Energia e Clima (PNEC) para o horizonte temporal 2021-2030. Foram delineadas um conjunto de metas dando prioridade à eficiência energética e à redução do consumo de

energia. As metas incluem a redução de emissões de gases com efeito de estufa entre 45% e 55% face a 2005; a redução do consumo de energia primária em 35% e atingir a meta da eficiência energética estabelecida em 35%. Inclui ainda aumentar a produção de energia através de fontes renováveis para 47% [2]. Este último implica que 80% da produção de eletricidade; 49% da energia utilizada para aquecimento/arrefecimento e 20% da energia utilizada nos transportes sejam de origem renovável. A Figura 1.1 ilustra o caminho percorrido e as metas a alcançar pelo país no que toca às fontes de energia renováveis (FER) no sector do aquecimento e arrefecimento (FER-A&A), na incorporação renovável na produção de eletricidade (FER-E) e no peso das energias renováveis nos transportes (FER-T) [3].

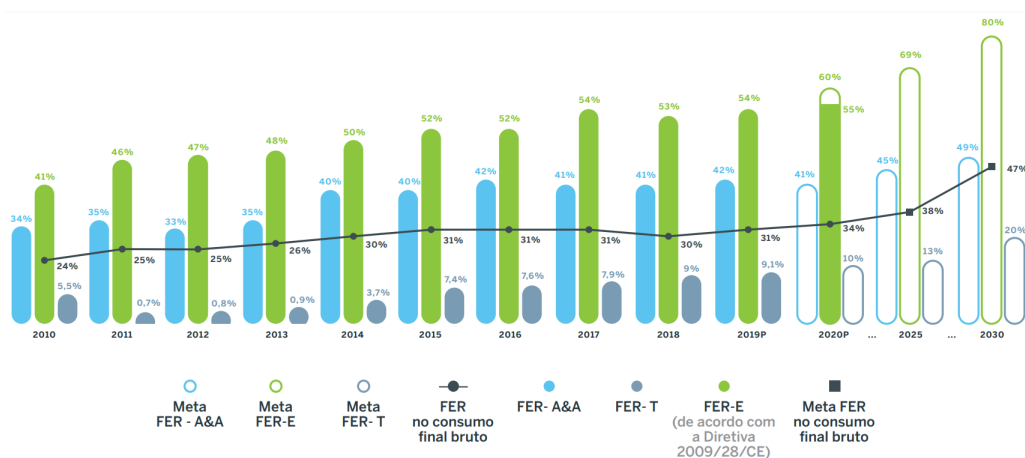


Figura 1.1: Principais indicadores para alcançar as metas do PNEC para 2030 [3].

Verifica-se que Portugal tem de continuar a trabalhar para atingir as metas estabelecidas no PNEC. A produção de eletricidade obtida, essencialmente, através de fontes renováveis será um passo preponderante em direção ao objetivo traçado e com impacto positivo na sociedade. De realçar que a sensibilização dos consumidores é crucial. Pois a adoção de medidas que permitam consumir de forma mais eficiente os recursos energéticos é essencial neste caminho.

Tendo em consideração a importância da eficiência energética propõe-se, com este trabalho, desenvolver um sistema que permita ao utilizador ter conhecimento dos seus consumos energéticos. E consciencializá-lo para o impacto energético de alguns dos seus hábitos.

1.1 Contextualização

Este projeto surge da vontade pessoal de desenvolver um sistema que permita monitorizar em tempo real os consumos energéticos de cargas presentes numa habitação. Com estes dados o utilizador pode consciencializar-se sobre o consumo energético

da sua habitação e adotar medidas que possibilitem reduzi-lo. A redução do consumo energético terá impacto direto na poupança mensal da família e na redução da pegada ecológica de cada indivíduo.

1.2 Definição do Problema

O desafio deste trabalho pode ser representado em 3 fases. Numa primeira fase é necessário desenvolver o *hardware* que irá realizar a recolha e processamento de dados. De seguida é necessário disponibilizar estes dados numa plataforma gráfica, que deve ser intuitiva e de fácil leitura. Por último, na terceira fase, pretende-se guardar os dados numa base de dados com o objetivo de os analisar à *posteriori*. A arquitetura do sistema será preponderante para o sucesso deste projeto pelo que será necessário estudar e aplicar as tecnologias que melhor se enquadram no mesmo.

1.2.1 Objetivos

A finalidade deste projeto é o desenvolvimento de um sistema de monitorização das cargas elétricas, permitindo ao utilizador aceder e visualizar os seus hábitos de consumo. Foram estabelecidos como objetivos:

- estudar conceitos fundamentais relativos à medição de energia elétrica;
- estudar os componentes que melhor se adequam ao projeto;
- desenvolver o sistema de monitorização;
- desenvolver *software* específico para o processamento dos dados;
- implementar uma tecnologia de comunicação que melhor se adequa ao projeto;
- desenvolver uma plataforma gráfica para que o utilizador tenha acesso aos seus dados de consumo;
- desenvolver uma plataforma para guardar os dados de consumo.

1.2.2 Resultados esperados

Deste trabalho é esperado obter os seguintes resultados:

- dispositivo de fácil implementação;
- medições de energia consumida com alto grau de precisão;
- dados do consumo energético visíveis ao utilizador através de uma interface gráfica;
- possibilidade de comparar consumo energético por carga;

- apresentação do custo energético atual e total da carga;
- implementação de uma rede de comunicação adequada ao projeto.

1.3 Plano de Trabalho

Na tabela 1 é visível a estruturação do projeto ao longo das 18 semanas de trabalho. O projeto está dividido em 11 atividades permitindo assim uma melhor gestão do trabalho ao longo do tempo.

A primeira atividade inicia com a formulação do problema e análise dos requisitos, com a duração de três semanas. De seguida é necessário realizar um estudo sobre as tecnologias existentes de forma a adquirir o material que melhor se adequa ao projeto. O desenvolvimento do *hardware* irá requerer cerca de cinco semanas de trabalho: três semanas dedicadas ao desenvolvimento de *hardware*; uma semana dedicada aos testes e validação; e outra semana inclui o desenvolvimento e os testes de *hardware*. Terminada esta parte iremos evoluir para o desenvolvimento de *software* de processamento dos dados. Estima-se uma duração de três semanas de trabalho. Por fim, as três semanas que se seguem, irão refletir o trabalho realizado no desenvolvimento da interface gráfica. É de realçar que ao longo do desenvolvimento do projeto o relatório será redigido consoante os avanços alcançados. As duas últimas semanas serão utilizadas na conclusão do relatório e na elaboração da apresentação do projeto.

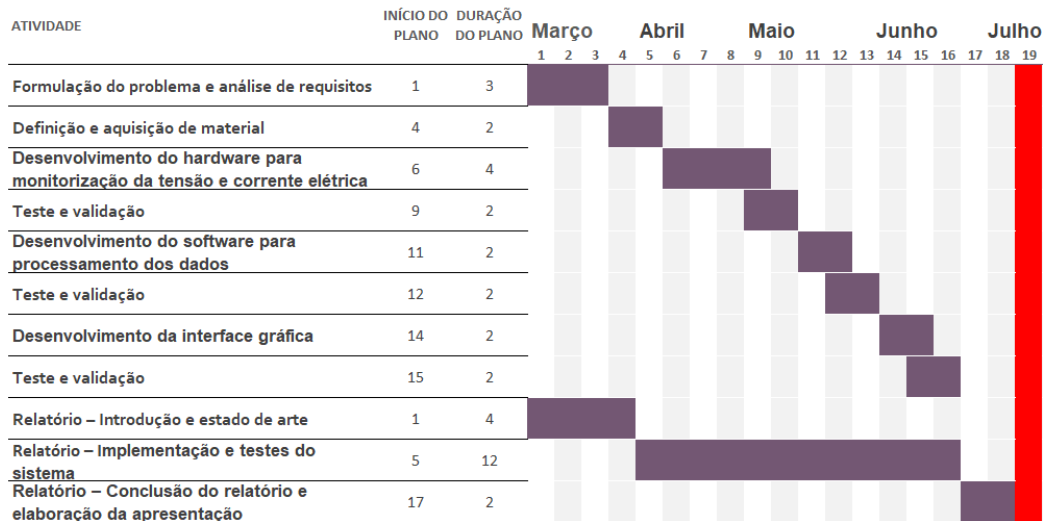


Figura 1.2: Calendarização do projeto

1.4 Organização da Dissertação

Este relatório está dividido em sete secções. A primeira secção apresenta a Introdução, dividida em quatro subsecções, Contextualização, Definição do Problema, Plano de Trabalho e a Organização da Dissertação. A segunda secção apresenta o estado de arte e as tecnologias utilizadas. A terceira secção apresenta um estudo sobre protocolos de comunicação IoT. A quarta secção aborda a arquitetura do projeto, onde se inclui o estudo dos componentes da alimentação do sistema, sensores e condicionamento de sinal, comparação entre ADC interno e externo, também são apresentadas as placas de desenvolvimento utilizadas, o display, o estudo dos protocolos de comunicação utilizados, também é referido o dispositivo que é utilizado como servidor, a introdução ao Node-RED e os cuidados a ter no desenvolvimento de uma PCB. A quinta secção aborda a implementação do sistema de *hardware* e do sistema de *software*. A sexta secção aborda os testes realizados e os resultados atingidos. São incluídos os testes realizados ao *hardware* e ao *software* de processamento dos dados elétricos e à interface gráfica. A sétima e última secção apresenta a conclusão do projeto, onde são apresentados de forma resumida os resultados alcançados e são referidas limitações deste projeto.

Capítulo 2

Estado de arte

A energia elétrica é um dos recursos mais importantes nos dias de hoje, sendo atualmente impensável levar uma vida normal sem usufruir das vantagens proporcionadas por esta energia. Para que a energia seja disponibilizada em nossa casa foi necessário desenvolver um sistema de produção, transporte e distribuição. Este sistema começou a ser pensado na época pelo inventor Thomas Edison (1847 – 1931), quando este ao inventar a lâmpada incandescente necessitou de implementar um sistema elétrico que permitisse alimentar as lâmpadas vendidas aos seus clientes. No ano de 1880 nasce a empresa *Edison Electric Illuminating Company*, que é responsável por desenvolver e instalar uma estação de produção de eletricidade [4]. De forma a rentabilizar o seu projeto, Edison teve de desenvolver aparelhos de medição de energia elétrica para o sistema iluminação de corrente contínua [5].

Naquela época a precisão e fiabilidade dos aparelhos era de pouca qualidade mas a evolução tecnológica possibilitou o aparecimento de aparelhos de medição de elevada qualidade. Atualmente os contadores de energia das casas portuguesas podem ser de quatro tipos: estáticos, eletromecânicos, híbridos ou inteligentes [6]. No futuro praticamente todas as residências terão contadores inteligentes, pois estes proporcionam uma gestão mensal eficaz, uma redução de custos para a empresa distribuidora de energia e uma maior comodidade aos seus clientes. Com estes contadores de energia é possível medir a energia total consumida pelas residências. No entanto é possível ir mais além e medir a energia elétrica consumida por carga, possibilitando uma noção sobre o consumo das cargas individuais.

Neste capítulo será analisado o atual estado de arte das tecnologias utilizadas

nos sistemas de medição e monitorização dos consumos energéticos por carga. Mencionando alguns aparelhos e descrevendo as suas características mais importantes. Algumas das características a ter em conta resumem-se aos seguintes pontos:

- **facilidade de uso** – é necessário que o aparelho seja de fácil utilização, quer seja quando se implementa o dispositivo pela primeira vez, quer seja quando se o utiliza para analisar os dados. A facilidade de implementação e utilização são características chave para despertar a atenção de possíveis clientes.
- **Visualização dos dados** – os dados tanto podem ser visualizados no *display* do dispositivo ou simplesmente através de uma interface gráfica utilizada noutros dispositivos, como por exemplo uma aplicação móvel ou uma aplicação *web*.
- **Conectividade** – Os aparelhos de medição de energia mais acessíveis são normalmente aparelhos que não têm ligação a outros dispositivos. No entanto, para o uso doméstico, dá-se preferência a tecnologias recentes que simplifiquem a utilização do mesmo e que possibilitem centralizar toda a informação num único local. Para isso é importante que o aparelho se conecte com outros dispositivos. A comunicação com outros dispositivos poderá ser realizada através das seguintes redes: *WiFi*, *Bluetooth*, *Z-Wave*, *ZigBee* [7].
- **medição do consumo de energia** – a característica mais importante do dispositivo é sem dúvida a possibilidade de o mesmo realizar medições do consumo energético com elevado grau de precisão. O consumo energético deverá ser principalmente apresentado em consumo energético (kWh) ou em valores monetários [8].

2.1 Dispositivos de medição de energia elétrica para uso doméstico

Esta secção apresenta dispositivos de uso doméstico com capacidade de realizar monitorização do consumo energético que estão atualmente no mercado. De forma a analisar e compreender as características de cada dispositivo em particular. Atualmente verifica-se uma tendência crescente na evolução tecnológica destes dispositivos elétricos para residências, possuindo características como monitorização do consumo energético, possibilidade de controlar cargas elétricas através de comandos de voz ou até mesmo a automatização dos dispositivos. De forma a tentar adquirir o máximo conhecimento possível sobre a tecnologia de monitorização de cargas elétricas a pesquisa irá incidir sobre cinco sistemas desenvolvidos por diferentes empresas. Estas empresas têm o objetivo comum de desenvolver sistemas que possibilitem melhorar o dia a dia dos seus utilizadores no entanto, utilizam meios diferentes para o alcançar.

2.1.1 FIBARO

Fibaro é uma empresa sediada na Polónia e que se dedica ao planeamento, desenvolvimento, produção e comercialização de sistemas de automação residencial. Estes sistemas integram dispositivos que utilizam a função *IoT*. Os dispositivos são desenvolvidos com a finalidade de tornar as residências com estruturas elétricas simples em residências inteligentes, sem necessidade de executar grandes intervenções na estrutura elétrica da mesma.

Dentro dos produtos Fibaro podemos encontrar diversos dispositivos inteligentes, tais como: interruptores, reguladores de luzes, tomadas, sensores de movimento, sensores de janela/porta e sensores de inundação. No centro de todos estes dispositivos temos o *Home Center 3*. Este dispositivo é o centro de toda a rede, ou seja, possibilita a integração de todos os dispositivos inteligentes num único sistema. Além disso permite automatizar ações para determinadas horas do dia de acordo com os hábitos pessoais, recolhe e analisa continuamente os dados de todos os dispositivos que integram a rede inteligente e realiza a gestão energética da residência sem necessidade de intervenção humana. De notar que todos os dispositivos podem ser geridos e/ou controlados individualmente sem a necessidade de instalar o *Home Center 3*, pois estes podem ser adquiridos no modelo *WiFi* ou no modelo *Z-Wave*.

A comunicação entre os dispositivos é realizada ou através da rede de comunicação *WiFi* ou através da rede de comunicação *Z-Wave*. Os canais oficiais da Fibaro garantem que a rede de comunicação *Z-Wave* é bastante estável e mais segura comparativamente com outras redes de comunicação *WiFi*. Este tipo de rede de comunicação, *Z-Wave*, permite ter todos os dispositivos a trabalhar e a comunicar entre si, mesmo não existindo uma conexão à rede de *internet*. Outra particularidade desta rede é o facto de cada dispositivo agir como um repetidor de sinal *Z-Wave*. Permitindo desta forma estender a área de comunicação tão longe quanto possível. A figura 2.1 ilustra o funcionamento da rede de comunicação *Z-Wave*.

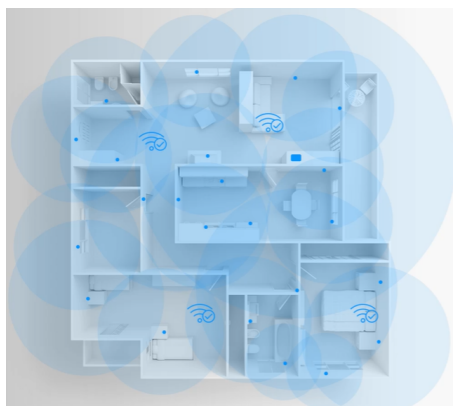


Figura 2.1: Rede comunicação de tecnologia *Z-Wave*[9]

A tecnologia de monitorização do consumo energético está presente nas tomadas

desenvolvidas pela Fibaro. Estas tomadas inteligentes permitem o controlo *On/Off* das cargas, monitorizar a potência e a energia consumida e, além disso, integra um *LED* em anel que permite mudar de cor consoante o consumo energético da carga. A figura 2.2 apresenta a tomada inteligente *Wall Plug - type B* [10]. Como referido anteriormente a instalação desta tomada não necessita do dispositivo central para se conectar à rede de Internet, pois a tomada pode ser adquirida no modelo *WiFi*.



Figura 2.2: Tomada de parede com monitorização do consumo de energia [10]

A interface gráfica da Fibaro apresenta bastantes soluções ao utilizador mas importa realçar a parte de monitorização de energia. Na figura 2.3 é apresentado um exemplo da página de monitorização de energia presente na aplicação móvel Fibaro. Esta página permite visualizar os consumos energéticos diário e mensais, dando destaque ao consumo diário de todas as cargas conectadas através do sistema Fibaro. O controlo e verificação de estado dos dispositivos pode ser feita em qualquer parte do mundo desde que exista uma conexão à Internet, permitindo desta forma integrar o conceito IoT.



Figura 2.3: Monitorização do consumo de energia apresentado na aplicação móvel Fibaro [10]

O modelo de tomada FIBARO FGWPF-102-5 com tecnologia *Z-Wave* pode ser adquirida no site da *amazon* por cerca de 66€ [11].

2.1.2 Eyedro

Eyedro Green Solutions é uma empresa especializada no desenvolvimento de *software* e dispositivos eletrônicos na área de monitorização de eletricidade em tempo real. Têm a sede localizada no Canadá e vende as suas soluções a clientes espalhados por mais de 150 países. Estes clientes abrangem o setor doméstico e empresarial.

A sua tecnologia engloba áreas tais como: monitorização de energia solar; sistema de gestão de energia de veículos elétricos; medição de energia residencial; monitorização de consumos de maquinaria e equipamento laboratorial. Relativamente à monitorização de energia residencial são apresentados quatro sistemas que diferem, essencialmente, no tipo de conexão ao dispositivo, quantidade de sensores incluídos (dois ou quatro sensores) e na aplicabilidade da monitorização energética solar. Entre as principais características encontra-se: a capacidade de monitorizar em tempo real a utilização e custo energético; guardar os dados na *cloud*; analisar o histórico de dados; comparar faturas e realizar estimativas mensais; analisar os dados em diferentes unidades, tais como, potência, corrente e valor monetário; instalação não invasiva; acessibilidade ao servidor *web* onde é possível visualizar graficamente os dados relativos aos consumos energéticos; e ser um sistema de fácil implementação e escalável. A solução apresentada pela Eyedro apresenta um valor de custo que varia aproximadamente entre os 116€ e 343€, dependendo do modelo pretendido.

A solução de monitorização do consumo energético disponibilizada pela empresa Eyedro é apresentada na figura 2.4. Esta solução destaca-se pela instalação do tipo não invasiva. Ou seja, não existe necessidade de abrir o circuito elétrico para instalar este sistema. A caixa negra apresentada à esquerda das pinças é o módulo central que processa os dados capturados da rede elétrica e os transmite para a aplicação *web*.



Figura 2.4: Rede de comunicações do sistema eyedro [12]

A análise dos dados relativos ao consumo energético é realizada através da aplicação *web* MyEyedro, apresentada na figura 2.5. Esta aplicação é baseada na *cloud* permitindo assim guardar e aceder aos dados sempre que existir uma ligação à *internet*.



Figura 2.5: Painel com dados do consumo energético acessível através da aplicação *web* MyEyedro [13]

A rede de comunicação utilizada por estes dispositivos são: *WiFi* 2.4 GHz, *ethernet* e *wireless mesh*. A figura 2.6 ilustra as três redes de comunicação possíveis de implementar neste sistema.

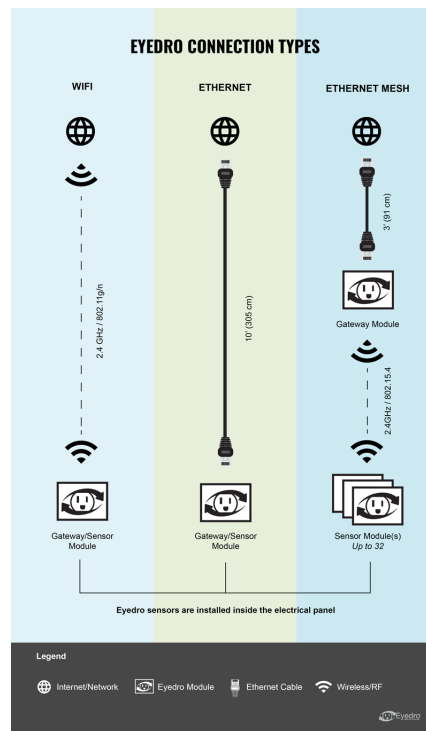


Figura 2.6: Rede de comunicações do sistema eyedro [12]

2.1.3 Lumin

Lumin é uma empresa com sede em Charlottesville, Virginia. Esta empresa criada em 2016 desenvolve sistemas de controlo inteligente para os quadros elétricos residenciais. A sua tecnologia permite que os circuitos elétricos comuns sejam convertidos em circuitos elétricos inteligentes. Os seus sistemas de *software* e *hardware* permitem controlar circuitos elétricos, monitorizar e apresentar o consumo de energia por circuito e guardar dados para futuras análises. Além disso, Lumin é um sistema orientado para habitações que possuem capacidade de produzir e armazenar

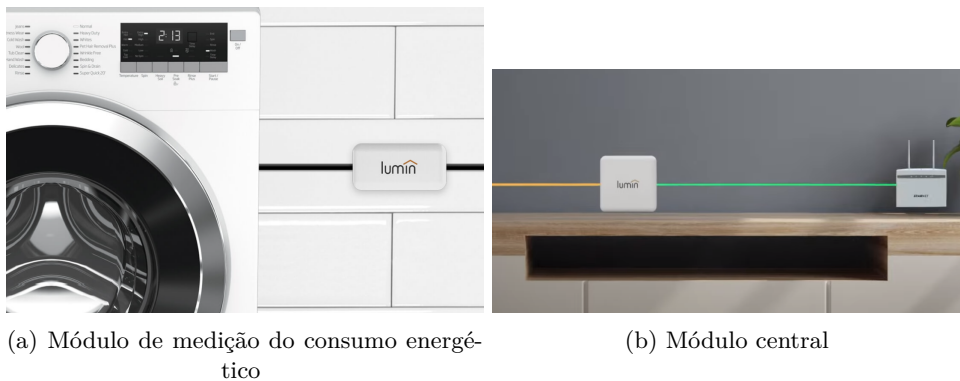
a sua própria energia, através de sistemas solares apoiados a baterias. Realizando assim uma gestão eficiente entre a energia produzida e energia consumida [14].

A figura 2.7 apresenta a arquitetura do sistema Lumin implementado numa residência ligado à rede elétrica e ao sistema solar de produção de energia. Pela figura é possível notar que o *hardware* é instalado entre o quadro elétrico e as cargas.



Figura 2.7: Sistema de monitorização Lumin [14]

Este sistema, representado na figura 2.8, é baseado num módulo central e em um ou mais módulos instalados em cada carga/circuito que realizam a monitorização do consumo energético. O módulo central, representado na figura 2.8b, realiza a transferência de dados entre a rede de *internet* e o módulo de monitorização de energia. Na figura 2.8a é apresentado o módulo de monitorização do consumo elétrico. Este módulo tem a particularidade de ser instalado de forma invasiva no circuito elétrico.



(a) Módulo de medição do consumo energético

(b) Módulo central

Figura 2.8: Representação dos dois módulos essenciais no sistema Lumin [15].

O sistema Lumin integra uma aplicação móvel intuitiva, permitindo ao utilizador aceder aos dados a partir de qualquer lugar com conexão à *internet*. Uma das páginas da aplicação é representada na figura 2.9. Nesta interface o utilizador terá acesso aos dados monitorizados em tempo real. Os dados referentes a cada carga ou circuito são apresentados por caixas, por exemplo o *Master Bath* mostra no centro da caixa o consumo total daquele circuito em específico. A carga *Air Conditioning* mostra o consumo total da carga e, por baixo, é possível visualizar um gráfico simples de energia/tempo com os consumos realizados numa base temporal.

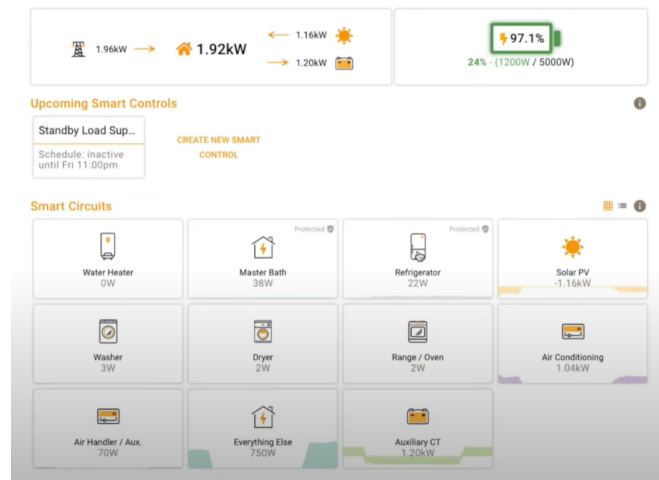
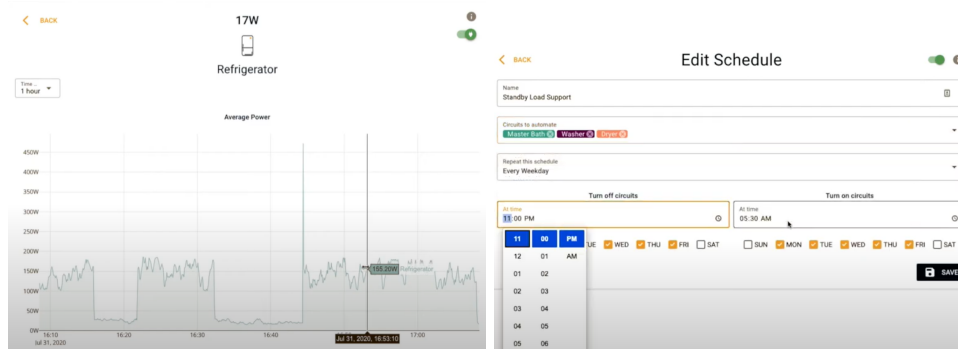


Figura 2.9: Sistema de monitorização Lumin [16]

Ao clicar na carga *Refrigerator* é possível ter acesso detalhado sobre o consumo do equipamento num determinado espaço temporal. A figura 2.10 apresenta duas sub-figuras 2.10a e 2.10b. Na sub-figura 2.10a é possível visualizar no topo do gráfico o consumo energético da carga médio no período de uma hora e também todo o histórico de consumo desse aparelho. Na sub-figura 2.10b é possível ver a página que permite agendar o controlo das cargas de acordo com os dias e horas desejados.



(a) Consumo energético detalhado

(b) Agendamento do controlo de cargas

Figura 2.10: Interface gráfica utilizada pelo sistema Lumin [16].

2.1.4 Leviton

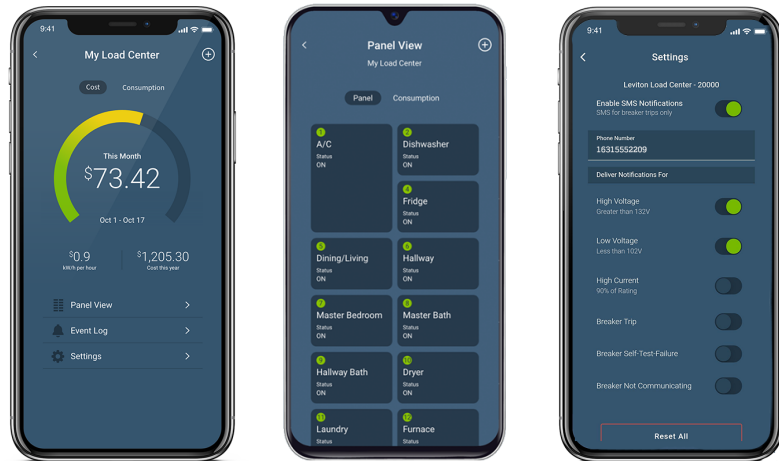
Leviton é uma empresa americana especializada no desenvolvimento de sistemas elétricos inteligentes para habitações familiares. Os seus produtos integram uma vasta panóplia de dispositivos inteligentes, incluindo disjuntores elétricos com capacidade de monitorizar, controlar e automatizar circuitos elétricos, interruptores, reguladores de luzes, tomadas, carregadores para veículos elétricos, entre outras soluções apresentadas no site da empresa. Relativamente aos disjuntores inteligentes, estes

mantêm o principal objetivo de proteger os circuitos elétricos adjacentes; monitorizar o consumo energético em tempo real por circuito; e controlar remotamente circuitos. Integram ainda um sistema de alertas que avisam o utilizador de situações anómalas presentes no quadro elétrico. Estes disjuntores necessitam de um dispositivo central para se conectar à rede de *internet* possibilitando assim a monitorização, controlo e automatização do sistema através da aplicação móvel *Leviton app*. Na figura 2.11 são apresentados os três principais componentes deste sistema. À esquerda é possível visualizar o dispositivo que permite conectar os disjuntores do quadro elétrico à rede de *internet*. Ao centro é apresentado o disjuntor inteligente. À direita vemos a vista da página principal da aplicação *Leviton app* [17].



Figura 2.11: Composição do sistema de disjuntor inteligente [17]

A figura 2.12 apresenta três sub-figuras. A sub-figura 2.12a mostra o consumo energético mensal, anual e por hora em valor monetário ou energético. A sub-figura 2.12b permite ao utilizador ver o estado ou desligar os disjuntores remotamente. A sub-figura 2.12c ilustra a página de alertas. Pois este sistema notifica o utilizador sobre o momento e o motivo do disparo do disjuntor de um circuito elétrico. Informa ainda se determinada carga está ativa ou inativa e permite atualizar o *firmware* com as últimas versões disponíveis pela Leviton.



(a) Detalhes do consumo energético (b) controlo remoto (c) Alertas e atualizações

Figura 2.12: Interface gráfica Leviton [17]

A figura 2.13 apresenta o quadro elétrico desenvolvido pela Leviton.

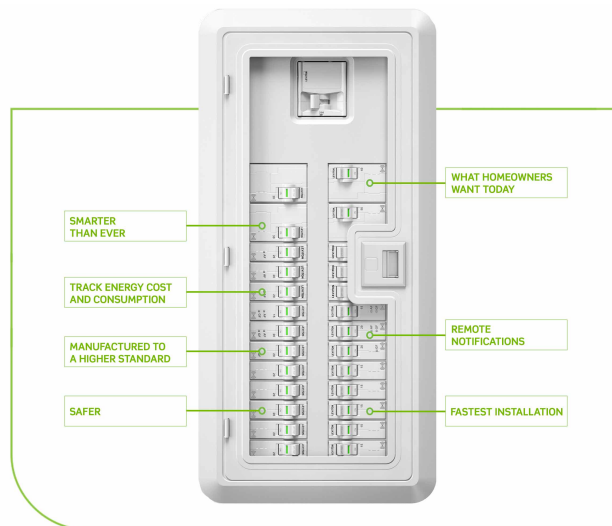


Figura 2.13: Quadro elétrico inteligente Leviton [17]

2.1.5 Trotec

Diferenciando-se dos aparelhos anteriormente abordados, este aparelho de medição de energia vendido pela empresa alemã Trotec representa uma gama de aparelhos de baixo custo. A energia consumida é visualizada somente no display, pois o mesmo não tem capacidade de se conectar a outros dispositivos, sendo assim um dispositivo que não integra o conceito IoT. Como características apresenta consumos a partir de 0,1W; possibilidade de visualizar a energia consumida em consumo energético (watts, kWh) ou em valor monetário e possibilidade de guardar registo dos dados em memória. A alimentação do aparelho é realizada pela tensão elétrica de 230V.

Comparando este dispositivo com os outros abordados anteriormente verifica-se uma grande diferença nas características oferecidas pelo mesmo. Por essa razão o preço de custo deste aparelho é bastante inferior, rondando os 15€ por dispositivo. Um exemplo deste dispositivo é visualizado na figura 2.14.



Figura 2.14: Aparelho de medição de energia elétrica simples [18]

2.2 Sensores de corrente

Neste capítulo é realizado o estudo sobre os sensores de corrente utilizados para medição de cargas elétricas para uso doméstico. Os sensores de corrente serão analisados em cinco características diferentes: Dimensão, tolerância de erro, consumo, preço e capacidade para monitorizar Alternating Current (AC). De acordo com os sensores disponíveis no mercado serão analisados três tipos de sensores: *shunt*, sensor de efeito Hall e transformadores de corrente.

2.2.1 *Shunt*

A resistência *shunt* é um condutor elétrico calibrado e de baixa resistência que permite obter valores de AC ou Direct Current (DC) com alto grau de precisão. A tolerância de erro varia entre $\pm 0,5\%$ e $\pm 5\%$ mas é muito comum encontrar estas resistências *shunt* com tolerâncias de erro de $\pm 1\%$ [19]. Como este condutor é aplicado em série com o circuito elétrico, é importante que a sua resistência seja baixa de forma a minimizar a potência elétrica consumida pelo mesmo. É também importante que a resistência seja calibrada, ou seja, que tenha um valor de resistência fixo dentro das gamas de corrente especificadas pelo fabricante, sendo assim possível obter uma correspondência linear entre a relação corrente/tensão.

A medição da corrente elétrica é obtida através da medição da tensão nos terminais da resistência *shunt*. Sabendo o valor da resistência e aplicando a lei de Ohm (ver equação 2.1) é possível obter uma correspondência linear entre a corrente que atravessa a resistência e a queda de tensão aos seus terminais.

$$V = RI. \quad (2.1)$$

Na figura 2.15 é possível verificar a aplicação da resistência *shunt* num circuito elétrico.

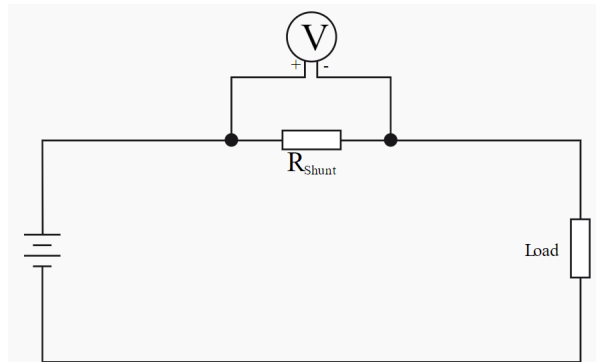


Figura 2.15: Aplicação de *shunt* no circuito elétrico [20].

A figura 2.16 representa uma resistência *shunt* utilizada para medir correntes até máximo de 100A. No entanto, é possível encontrar resistências em que a corrente máxima admitida varie entre os 1A e 3000A [21]. Num exemplo em que resistência é especificada para trabalhar com correntes até 100A e com uma queda de tensão aos seus terminais de $75mV$, segundo a lei de ohm a resistividade desta resistência é igual a $0,75m\Omega$. Ao aplicar esta resistência *shunt* no circuito teremos consequentemente um acréscimo na potência consumida na ordem dos $7,5W$. A potência consumida pela aplicação da resistência no circuito é pouco significativa, mas o grande desafio para aplicar a resistência *shunt* num circuito de monitorização de energia elétrica encontra-se na dimensão do *shunt* e na sua capacidade de medir correntes mais baixas. Utilizando esta resistência como exemplo, se no circuito de carga atravessar uma corrente na ordem dos $500mA$ a queda de tensão aos seus terminais será na ordem dos $0,375mV$. Sendo uma tensão muito baixa implica aplicar circuitos adicionais para realizar um apropriado condicionamento de sinal.



Figura 2.16: Resistência *shunt*, para medição de correntes até 100A [22].

No entanto, é possível encontrar resistências de menor dimensão tal como a apresentada na figura 2.17. Esta apresenta uma resistência calibrada e de dimensão reduzida. Dependendo do modelo da resistência é possível encontrar tolerâncias de erro que variam entre $\pm 1\%$, $\pm 5\%$ e $\pm 10\%$. A resistividade apresentada pela resistência situa-se entre $0,001\Omega$ a $0,5\Omega$ suportando correntes entre $6,5A$ a $80A$. A dimensão da resistência situa-se entre $9,89\text{ mm}$ a $25,40\text{ mm}$ e $5,46\text{ mm}$ a $72,77\text{ mm}$,

referências para valores de altura e comprimento respectivamente [23]. Ao contrário da resistência referida anteriormente esta apresenta um custo de aquisição mais alto.



Figura 2.17: Resistência *shunt*, para medição de correntes até 80A [23].

Se analisarmos resistências de fabrico SMD é possível obter uma redução significativa do tamanho e também uma redução no custo de aquisição. Em relação às características como tolerância de erro e potência máxima de trabalho estas mantêm-se idênticas às características das resistências referidas anteriormente [24]. Na figura 2.18 é apresentada uma resistência do tipo SMD.

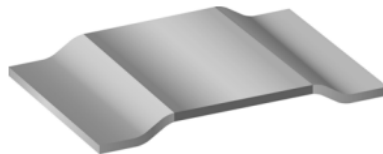


Figura 2.18: Resistência *shunt*, do tipo SMD [24].

2.2.2 Sensor de efeito Hall

Este tipo de sensor mede a corrente elétrica através da variação do campo magnético produzido pela passagem de corrente no circuito elétrico. Ou seja, a corrente elétrica ao atravessar o condutor elétrico cria um campo magnético. Este campo magnético é sentido pelo efeito Hall e produz uma tensão de saída proporcional à corrente de entrada [25]. Um exemplo do funcionamento de sensor de efeito Hall é representado na figura 2.19.

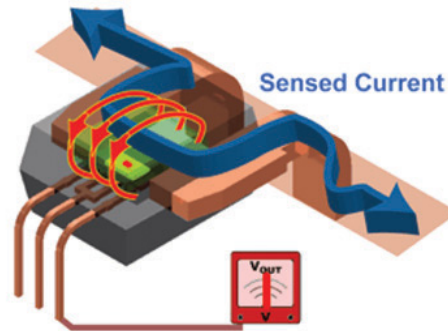


Figura 2.19: Princípio de funcionamento do sensores de efeito Hall [25].

Na figura 2.20 verifica-se três tipos de circuitos integrados com diferentes encapsulamentos desenvolvidos pela Allegro microsystems [26]. Estes sensores permitem medir correntes de 5A a 200A e foram desenvolvidos para aplicações da indústria automóvel. De notar que estes sensores são mais pequenos e oferecem uma ampla gama de medições.

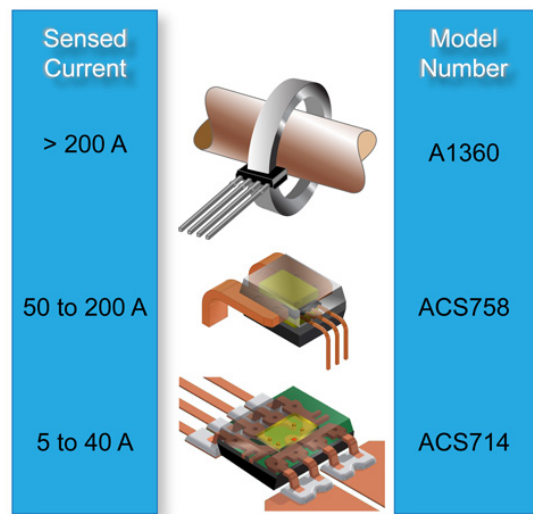


Figura 2.20: Sensores de efeito Hall com diferentes encapsulamentos [27].

O tipo de sensor ACS711EX, vendido pela PTROBOTICS [28], representado na figura 2.21 permite medir AC no máximo entre -31A e +31A. A sensibilidade do sensor é representada por 45mV/A ou 68mV/A se o sensor for alimentado por 3.3V DC ou 5V DC respetivamente. A percentagem de erro no valor de saída em tensão é na ordem do $\pm 5\%$ [29]. Um valor alto comparado com a tolerância de erro da resistência *shunt*. No entanto, este sensor ganha pelo pequeno tamanho quando comparado com a resistência de *shunt* e o transformador de corrente.

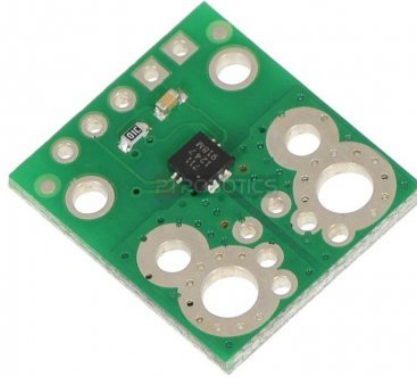


Figura 2.21: Sensor de ACS711EX, [28].

2.2.3 Transformadores de corrente

O sensor do tipo transformador de corrente é do tipo não invasivo. Este sensor é utilizado em circuitos de AC e é constituído por um enrolamento primário, um enrolamento secundário e um núcleo magnético, como representado na figura 2.22. O seu princípio de funcionamento consiste em converter as correntes altas do enrolamento primário em correntes baixas do enrolamento secundário. O condutor do circuito elétrico da carga é colocado no meio do anel de ferrite, sendo assim definido como enrolamento primário. O enrolamento primário produz um campo magnético e este induz uma corrente no enrolamento secundário. Como a corrente no secundário é inferior à corrente do primário é possível realizar medições .

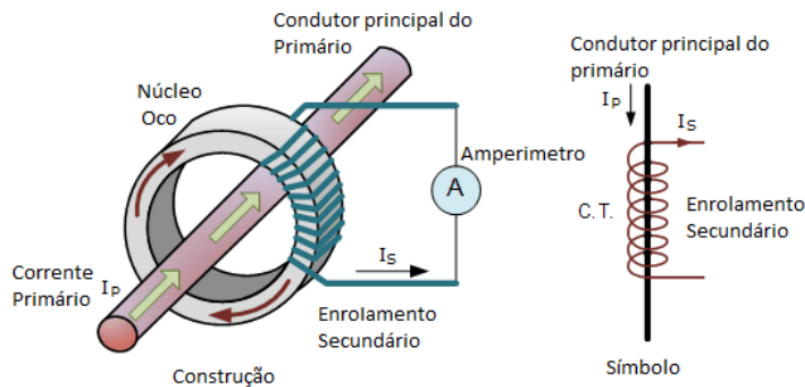


Figura 2.22: Princípio de funcionamento de um transformador de corrente [30].

Existe uma relação direta com o número de voltas do enrolamento primário com o número de voltas no enrolamento secundário e as suas correntes. A equação 2.2 representa essa relação entre o número de voltas do enrolamento com a corrente que

a atravessa. N_p representa o número de voltas do circuito primário e N_s do circuito secundário. I_p representa a corrente que atravessa o circuito primário e o I_s a do circuito secundário.

$$\frac{N_p}{N_s} = \frac{I_s}{I_p}. \quad (2.2)$$

A figura 2.23 apresenta um sensor do tipo transformador de corrente. Este tipo de sensor possibilita medições de 0 a 30A [31] com uma sensibilidade de 1V por 30A. Estes tipos de sensores podem medir correntes na ordem das centenas de ampere. Apresenta um erro de saída em torno de $\pm 3\%$ [32]. Estes sensores caracterizam-se por ser do tipo não invasivo, são sensores grandes comparativamente com os sensores de efeito Hall e custam aproximadamente 14€ [31].



Figura 2.23: Transformador de corrente [31].

2.3 Estudo e comportamento da energia em diferentes cargas

O projeto tem como objetivo monitorizar o consumo energético de cargas e para isso é essencial analisar o tipo de energia elétrica disponível nas habitações. A corrente elétrica pode ser definida em dois tipos, DC ou AC. A do tipo DC não varia ao longo do tempo, mantendo sempre a mesma direção e a mesma magnitude no circuito elétrico. Esta é muito utilizada nas baterias para alimentar pequenos dispositivos, tais como, relógios, telemóveis, lanternas, etc. A energia elétrica do tipo AC, é a energia disponibilizada pelos distribuidores de energia elétrica e é muito importante para alimentar todos os aparelhos que se conectam à tomada elétrica, tais como, frigoríficos, ar condicionados, fogões elétricos, máquinas de lavar, etc. Esta energia elétrica tem a forma de uma onda sinusoidal, como é representado na figura 2.24. Isto significa que a energia varia periodicamente ao longo do tempo, mudando de direção e mudando de magnitude.

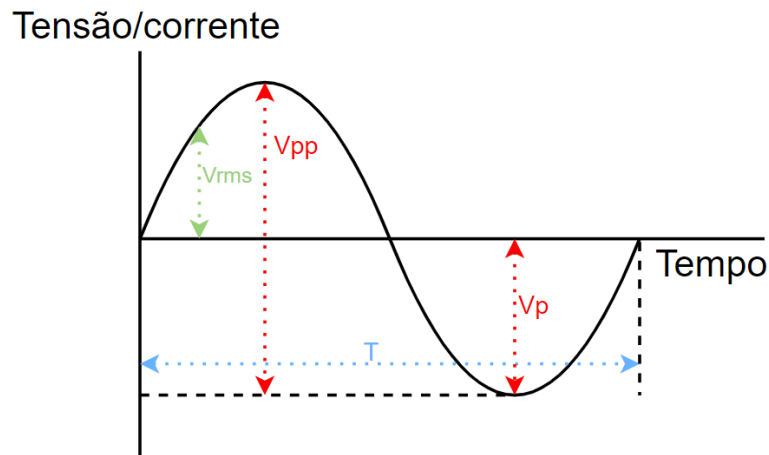


Figura 2.24: Forma de onda sinusoidal.

Existem alguns elementos importantes da onda sinusoidal a ter em consideração, são eles: valor máximo ou valor de pico (V_p); valor de pico a pico (V_{pp}); Valor eficaz ou *Root Mean Square (RMS)*; período (T); e frequência da onda (f). A frequência determina o número de ciclos que ocorre num segundo e a sua unidade é o Hz. O período, designado pela letra T , é o valor do intervalo de tempo entre repetições e a sua unidade é o segundo (s). A equação 2.3 representa a relação entre a frequência e o período da onda sinusoidal.

$$f = \frac{1}{T}. \quad (2.3)$$

O VP representa o valor instantâneo mais elevado da onda sinusoidal. O VPP representa a amplitude máxima de uma onda sinusoidal, indo desde o ponto mais negativo da arcada inferior até ao ponto mais positivo da arcada superior. O elemento mais utilizado para analisar a componente elétrica é o valor eficaz. O valor eficaz permite perceber qual a corrente, tensão e potência eficaz num determinado instante. O resultado do valor eficaz é obtido através de uma das duas seguintes equações: 2.4 ou 2.5 [33].

$$V_{rms} = \frac{1}{\sqrt{2}} V_p. \quad (2.4)$$

$$V_{rms} = \frac{1}{2\sqrt{2}} V_{pp}. \quad (2.5)$$

De modo a conhecer a potência que as cargas consomem, primeiro é necessário perceber que as cargas podem ser de três tipos: resistivas, indutivas ou capacitivas. Nas habitações residenciais é muito comum ter cargas resistivas e cargas parcialmente reativas. As cargas resistivas são compostas por resistências puras. A corrente elétrica e a tensão estão em fase, significando que a energia consumida é

totalmente utilizada na carga. São exemplos de cargas resistivas o: ferro de engomar, aquecedor elétrico, lâmpadas incandescentes, cilindro para aquecer água, etc. A figura 2.25, apresenta a potência consumida por uma carga resistiva. De notar que nestes casos a potência tem sempre valor positivo, significando que toda a potência é consumida pela carga.

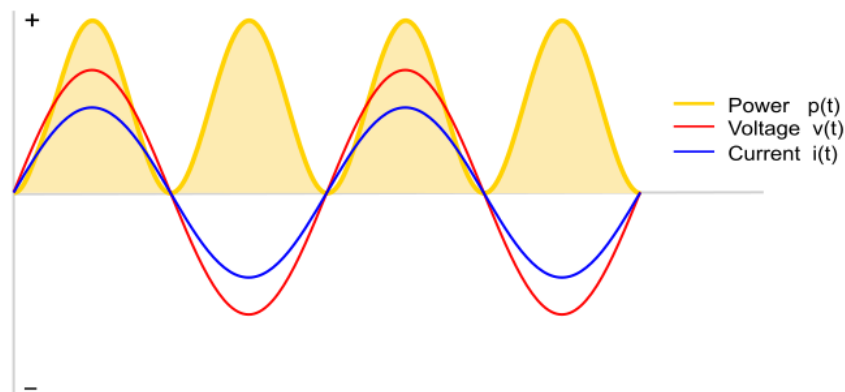


Figura 2.25: Potência consumida por uma carga resistiva [34].

As cargas parcialmente reativas incluem não só elementos resistivos como também elementos indutivos e capacitivos. A introdução destes elementos indutivos e/ou capacitivos num circuito elétrico tem a particularidade de desfasar a corrente e a tensão. No caso das cargas capacitivas a tensão está atrasada em relação à corrente. No caso de cargas indutivas a corrente está atrasada em relação à tensão. De notar que quanto maior for o desfasamento entre a tensão e a corrente maior será o trabalho realizado para entregar potência real às cargas. Nestas situações a potência terá valores negativos em determinados instantes, significando que o circuito gera potência ao invés de a consumir. A figura 2.26 representa a potência dissipada numa carga parcialmente reativa.

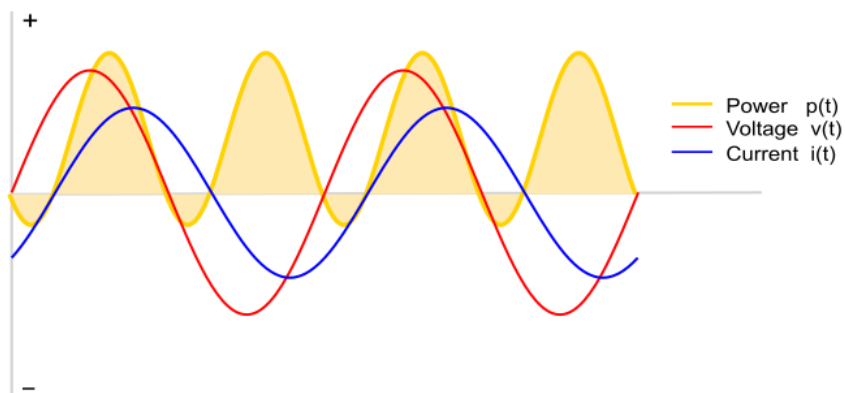


Figura 2.26: Representação da potência em cargas parcialmente reativas [34].

É devido aos diferentes tipos de carga presentes num circuito elétrico que a potência elétrica é composta por três tipos de potência: potência ativa ou real; potência reativa; e potência aparente. A potência ativa, medida em W, representa o trabalho real produzido na carga. A potência é representada pela equação 2.6, sendo que ϕ representa o ângulo entre a tensão e a corrente.

$$P = VI \cos(\phi). \quad (2.6)$$

A potência reativa é uma potência que resulta de cargas indutivas ou capacitivas. Esta potência não é consumida pela carga, sendo enviada novamente para a rede de distribuição elétrica. Esta potência é designada pela letra Q e a sua unidade é VAR.

$$Q = VI \sin(\phi). \quad (2.7)$$

A potência aparente é a combinação da potência real com a potência reativa. Esta é designada pela letra S e é medida em VA. A figura 2.27 apresenta a combinação entre as diferentes potências.

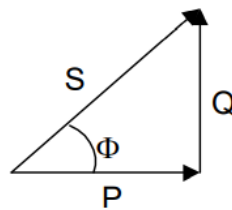


Figura 2.27: Triângulo das potências [35].

A equação 2.8 representa a fórmula da potência aparente.

$$S = \sqrt{P^2 + Q^2}. \quad (2.8)$$

Outro elemento importante é o fator de potência, permitindo-nos analisar a eficiência da rede elétrica através da relação entre a potência ativa e a potência aparente. O fator de potência pode ter valores entre 0 e 1. O fator de potência é 0 quando toda a potência é reativa sem potência real (geralmente carga indutiva). O fator de potência é 1 quando toda a potência é potência real sem potência reativa (carga resistiva). Esta relação é apresentada na equação 2.9.

$$\cos\phi = \frac{P}{S} = \frac{VI \cos\phi}{VI}. \quad (2.9)$$

Capítulo 3

Protocolos de comunicação IoT

Este capítulo tem como objetivo abordar o conceito IoT e estudar os diferentes tipos de protocolos de comunicação passíveis de serem implementados num sistema IoT. Os protocolos que iremos abordar estão categorizados em duas camadas diferentes: camada física e camada de aplicação. A camada física é a camada mais baixa, constituindo a interface com o meio físico de transmissão. Esta estabelece o modo como a representação lógica dos dados são transformados em dados físicos e permite a conexão entre os vários dispositivos de uma rede IoT. A camada de aplicação é a camada mais elevada e é responsável por fornecer os serviços da rede às aplicações do utilizador [36], isto é, permite ao utilizador da aplicação comunicar com os vários dispositivos da rede IoT. Posto isto, são estudados os seguintes protocolos da camada física: *Long Range Wide Area Network* (LoRaWAN); *Bluetooth*; WiFi; e ZigBee;. Relativamente à camada de aplicação são estudados os protocolos: MQTT; e CoAP.

3.1 Internet Of Things

Atualmente encontramos vários dispositivos que permitem capturar e analisar dados e enviá-los para uma plataforma onde são apresentados e guardados. Por exemplo os relógios inteligentes permitem monitorizar o estado físico, como cronometrar atividades físicas, analisar o ritmo do batimento cardíaco, nível de oxigénio no sangue, calorias queimadas, entre outras possibilidades. Outro exemplo de dispositivo são as balanças inteligentes que permitem recolher dados físicos sobre o utilizador, como o seu peso, Índice de Massa Corporal (IMC), percentagem de gordura corporal,

massa muscular, massa óssea, metabolismo basal, idade física, entre outros dados. Estes dois dispositivos além de mostrarem estas informações através dos seus visores também permitem enviar estes dados estatísticos e mostra-los por exemplo numa aplicação de telemóvel. Estes, bem como outros dispositivos, comunicam direta ou indiretamente com a rede de Internet para transmitir os seus dados. E em cada caso, o dispositivo que estava conectado à Internet não era um computador, *tablet* ou telemóvel, mas sim um objeto, uma Coisa [37].

Estas coisas ou dispositivos são compostos por sensores e pequenos microcontroladores capazes de processar os dados recolhidos pelos sensores e enviar estes dados para outro dispositivo. De realçar que nem todos os dispositivos têm de estar conectados à Internet, basta terem um canal de comunicação aberto com outro dispositivo que tem a comunicação à Internet e o conceito aplica-se na mesma.

De acordo com a McKinsey & Company a Internet das Coisas é definida como sendo objetos físicos embebidos com sensores e atuadores que comunicam com sistemas computadorizados através de redes de comunicação física ou *wireless*, permitindo ao mundo físico ser digitalmente monitorizado ou mesmo controlado [38].

Uma curiosidade em relação ao crescimento económico do IoT nos próximos anos é apresentado na figura 3.1. Nesta figura são apresentadas nove áreas diferentes onde é possível visualizar o potencial económico para o ano de 2030 de cada área, bem como na sua totalidade, sendo o total estimado estar compreendido na faixa dos 5500 e 12600 mil milhões de dólares.

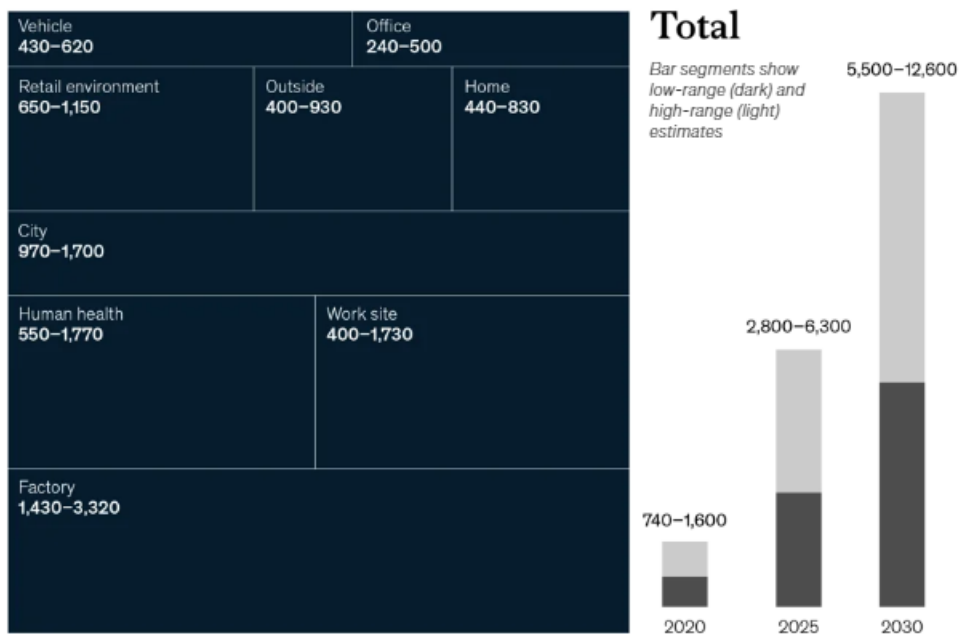


Figura 3.1: Estimativa do valor económico em mil milhões de dólares para o ano 2030 na adoção da IoT em nove áreas distintas [39].

3.2 Protocolos da camada física

Os protocolos da camada física permitem realizar as transmissões de dados através de meios de comunicação físicos, como cabos de cobre, fibra ótica e ondas de rádio. A camada física é a camada mais baixa do protocolo de comunicações IoT [7] e lida com a transmissão de dados de um dispositivo para outro. Neste capítulo iremos abordar os seguintes protocolos: LoRaWAN, Bluetooth, WiFi e ZigBee.

3.2.1 LoRaWAN

LoRaWAN é um protocolo de comunicação *wireless* do tipo *Low Power Wide Area Network* (LPWAN). Estas redes são desenvolvidas essencialmente para sensores e aplicações que funcionam a baterias e que precisam de enviar pequenas quantidades de dados através de longas distâncias. Enquanto que o LoRaWAN é responsável por definir o protocolo de comunicação e a arquitetura do sistema da rede, o *Long Range* (LoRa) representa a camada física e é responsável por transmitir dados através de longas distâncias utilizando ondas de rádio de baixa frequência [40]. Os dispositivos que utilizam o LoRa como módulo de comunicação permitem maximizar a duração das baterias e alcançar distâncias de até 15 km em ambiente suburbano e entre 2 a 5 km em ambiente urbano. No continente europeu o protocolo LoRaWAN trabalha na gama de frequências de 868 MHz e define dez canais de comunicação, oito dos quais permitem transferir dados a uma taxa de 0.250 Kbps até 5.5 Kbps, um canal LoRa para transmitir a taxas de 11 Kbps e outro canal com modulação *frequency shifting keying* (FSK) para transmitir até 50 Kbps [40].

O LoRaWAN é implementado com base em três classes de dispositivos diferentes: Classe A; Classe B; e Classe C. A Classe A é a classe com menor consumo energético mas em contrapartida coloca maior latência na comunicação. Esta classe realiza transferências de dados utilizando *slots* de transmissão programadas pelo dispositivo final, ou seja, a comunicação é sempre iniciada pelo dispositivo final. Depois de a transmissão *uplink* ser realizada seguem-se duas janelas curtas de *downlink*, dando a oportunidade de comunicação bidirecional. A Classe B é uma classe intermédia entre a Classe A e Classe C. A transmissão de dados ocorre de acordo com as características da Classe A e além disso são abertas *slots* de transmissão de *downlink* periódicas, possibilitando ao dispositivo final receber comandos do servidor. A Classe C é a classe com maior consumo energético mas com uma menor latência associada à transmissão de dados. Esta classe além de ter as características de transmissão da Classe A, mantém também as janelas de transmissão *downlink* sempre abertas, reduzindo a latência mas aumentando o consumo energético (50 mW) [41].

Numa rede LoRaWAN os dispositivos de captura e/ou processamento de dados são designados de nós finais. Estes nós não transmitem os dados entre si, em vez disso eles enviam os dados para uma ou várias *gateways*. Cada *gateway* tem o objetivo de

encaminhar os dados para o servidor de rede central. Este por sua vez gere a rede e filtra os pacotes redundantes recebidos, realiza verificações de segurança, etc. Por fim, as aplicações tratam, guardam ou apresentam os dados aos utilizadores. Na figura 3.2 é possível analisar a arquitetura de uma rede LoRaWAN.

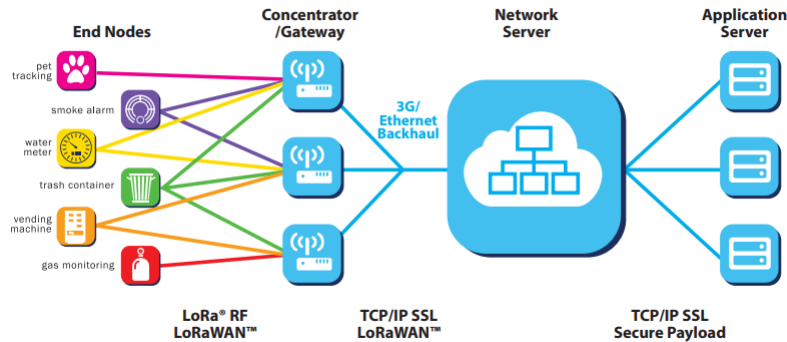


Figura 3.2: Arquitetura de rede LoRaWAN [40].

Esta tecnologia pode ser implementada utilizando o módulo RFM95. Este módulo permite uma taxa de transferência de até 300 Kbps, é especificado para uma transmissão contínua de dados uma potência do sinal de saída de 20 dbm, representando aproximadamente 100 mW de potência. Para uma transmissão de dados mais eficiente o consumo energético poderá reduzir até aos 25 mW. O módulo custa aproximadamente 9€ [42].

3.2.2 Bluetooth

O *Bluetooth* é uma tecnologia de comunicação de dados sem fios amplamente utilizada em dispositivos móveis. Atualmente distingue-se a tecnologia *Bluetooth* em dois tipos, o *Bluetooth Classic* e o *Bluetooth Low Energy* (BLE). Estes dispositivos são capazes de comunicar entre si, pois o BLE é retro-compatível, a comunicação do BLE com o *Bluetooth Classic* é realizada através do protocolo do *Bluetooth Classic*. Estes dois tipos de tecnologia divergem no *hardware*, na arquitetura e no propósito.

O *Bluetooth Classic*, também referido como *Bluetooth Basic Rate* (BBR) ou *Bluetooth Enhanced Data Rate* (BEDR), é um dispositivo de comunicação sem fios que utiliza ondas rádio de baixa potência e de pouco alcance. Os dados são enviados através de 79 canais numa gama de frequência 2.4GHz. Suporta somente comunicações entre dispositivos ponto-a-ponto. Esta tecnologia é maioritariamente aplicada na transmissão de dados áudio, sendo também utilizado para transferências de ficheiros.

O BLE foi concebido com o objetivo de transmitir dados utilizando baixos consumos de energia e operando em baixas larguras de banda. Isto permite operar o dispositivo por longos períodos de tempo através de pequenas baterias. Os dados

são transmitidos através de 40 canais na gama de frequência 2.4GHz. Enquanto que o *Bluetooth Classic* permite conexões ponto-a-ponto, no caso do BLE é possível conectar os dispositivos através de três tipologias diferentes: ponto-a-ponto; estrela; *broadcast*; e malha. Sendo a tipologia malha muito útil em aplicações IoT [43]. As áreas de aplicação desta tecnologia são: transferências de dados, serviços de localização e ligações de dispositivos em rede. Entre as características desta tecnologia importa realçar o alcance e a taxa de transmissão de dados. O alcance do dispositivo varia entre 50 a 150 metros e a taxa de transmissão de dados varia entre os 2 Mbps e 125 Kbps.

Esta tecnologia pode ser aplicada utilizando os módulos HM-10, ESP32 ou ESP8266. Os ESPs além de incluírem uma unidade de processamento que possibilita o desenvolvimento de código também integram tecnologia WIFI e *Bluetooth*. O módulo HM-10 é utilizado somente para realizar a transmissão de dados. Este módulo permite uma taxa de transmissão de dados até 24 Mbps com consumo de energia reduzido e um alcance de até 100 metros em espaço aberto. O preço deste módulo é de aproximadamente 6€ [44]

3.2.3 WiFi

Em relação ao Wifi existem várias versões, tais como: Wi-Fi 1 (802.11b), Wi-Fi 2 (802.11a), Wi-Fi 3 (802.11g), Wi-Fi 4 (802.11n), Wi-Fi 5 (802.11ac), Wi-Fi 6 (802.11ax), e Wi-Fi HaLow (802.11ah). Cada versão tem a sua própria especificidade, ou seja, as versões variam entre si de acordo com a velocidade de transmissão de dados, capacidade de reduzir a interferência de sinais de outras fontes, alcance e custo de *hardware* [45].

Em relação a outras tecnologias *wireless* o WiFi apresenta algumas diferenças, entre as quais se destaca a frequência de transmissão, podendo operar nas gamas de frequência abaixo de 1GHz (Wi-Fi HaLow), 2.4 GHz e 5 GHz (Wi-Fi 4 e Wi-Fi 6), 6GHz (Wi-fi 6E) e por fim 60GHz (WiGig). Na figura 3.3 são apresentadas as várias versões WiFi e as respetivas frequências de operação. De notar que as frequências mais altas permitem uma maior transferência de dados, sendo bastante úteis para aplicações de realidade virtual, transmissões multimédia, jogos, etc. Um inconveniente da utilização frequências altas é o maior consumo energético e o curto alcance. Em relação ao Wi-fi 6, a versão utilizada em *routers* mais recentes, destaca-se a taxa de transmissão de dados entre 600 Mbps a 9608 Mbps. Utiliza a topologia estrela e tem um alcance aproximado de 10 metros.

O WiFi HaLow foi desenvolvido especificamente para aplicações IoT. Este utiliza ondas de rádio abaixo de 1 GHz, operando na gama de frequências entre 750 MHz e 928 MHz. A operação nestas gamas de frequência permite alcançar distâncias de até 1 Km. A sua taxa de transmissão situa-se entre os 150 Kbps e os 4 Mbps. Utiliza modos de poupança de energia de forma a minimizar o seu consumo de energia. Por

exemplo, os dispositivos finais são colocados em repouso por determinados períodos e acordam quando atingem o tempo máximo de repouso, ficando à escuta de novos dados até voltarem ao estado de repouso [46].

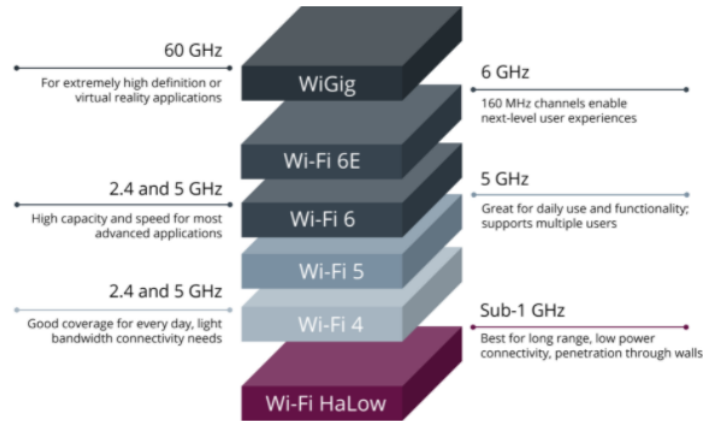


Figura 3.3: Versões Wi-Fi [47].

Esta tecnologia pode ser aplicada utilizando os módulos ESP32 ou ESP8266. O módulo ESP32 devKitC tem como principais características: microprocessador com frequência máxima de operação 240 MHz, 18 entradas ADC de 12 bits e gama de tensão entre os 0V e 3.3V, implementa protocolo de comunicação WiFi 802.11 b/g/n, versão 4.2 do *Bluetooth classic* e BLE. A utilização do protocolo WiFi permite obter uma taxa de transmissão de 150 Mbps. O preço do dispositivo é de aproximadamente 13€ [48].

3.2.4 ZigBee

O ZigBee foi desenvolvido no início do ano 2000 pela fundação *Connectivity Standard Alliance* com o objetivo de oferecer uma tecnologia de transmissão de dados sem fios com características diferentes das tecnologias existentes daquela época (WiFi e *Bluetooth*). Ou seja, uma tecnologia que permitisse a transmissão de dados utilizando um protocolo comum, com um reduzido consumo de energia, baixa latência e elevada fiabilidade.

ZigBee opera sobre o IEEE 802.15.4, que é um padrão de rádio da camada física que realiza o controlo de acesso às comunicações sem fio bastante fiável. Opera na banda de frequência 2.4 GHz a nível mundial, 915Mhz na América e 868 Mhz na Europa. A taxa de transmissão atinge os 250 Kbps utilizando a frequência 2.4 GHz de 16 canais, 500 Kbps na frequência de 915 a 921 Mhz de 27 canais e 100 Kbps na frequência de 868 Mhz de 63 canais. O alcance das transmissões variam entre 10 a 100 metros, dependendo da potência do sinal de saída bem como do meio ambiente onde o sinal é transmitido.

Este protocolo de comunicação permite implementar topologias em estrela, malha e árvore. Utilizando a topologia em malha é possível obter uma maior fiabilidade na transmissão de dados, como por exemplo estender o alcance total da rede através de saltos entre dispositivos, ou até, na reorganização da rede quando um dos dispositivos pára de comunicar.

De notar que uma rede de dispositivos que utilize ZigBee terá de ser constituída por 3 classes de dispositivos: ZigBee coordenador, ZigBee distribuidor e ZigBee dispositivo final.

Esta tecnologia pode ser aplicada utilizando os módulos XBee. Para o módulo XBee com uma taxa de transmissão de dados máxima de 250 Kbps, um alcance de 120 metros, com potência de saída de 2mW, 6 entradas ADC de 10 bits e 8 entradas/saídas digitais o custo aproximado é de 37€ [49].

3.3 Protocolos da camada de aplicação

Os protocolos de aplicação são uma parte importante da estrutura de protocolos de rede e estão localizados na camada mais alta do protocolo de comunicações IoT [7]. Estes protocolos são desenvolvidos para permitir que aplicativos e serviços em diferentes dispositivos de rede comuniquem e partilhem dados entre si. Neste capítulo serão abordados os seguintes protocolos de aplicação: MQTT e CoAP.

3.3.1 Message Queuing Telemetry Transport - MQTT

O protocolo MQTT foi desenvolvido pela IBM nos 1990 com o objetivo de ser utilizado em sistemas de supervisão e aquisição de dados em ambientes industriais. É um protocolo de comunicação concebido para trocar dados entre dispositivos, utilizando o padrão *publish/subscribe*. Sendo bastante utilizado na comunicação *Machine to Machine* (M2M) e em aplicações IoT. Utiliza como camada de transporte o protocolo *Transmission Control Protocol* (TCP). Destaca-se por ser um protocolo leve, por utilizar pouca largura de banda, simples de implementar e com elevada qualidade na transmissão de informação [50].

O protocolo é composto por dois elementos chave: cliente MQTT e *broker* MQTT. Os clientes MQTT podem ser clientes subscritores e/ou clientes publicadores. Os clientes nunca entram em contacto direto com outros clientes, pois o *broker* possibilita a intermediação da comunicação entre os clientes. Os clientes comunicam entre si utilizando mensagens e tópicos. Os clientes enviam/recebem mensagens escritas em tópicos específicos, permitindo assim saber quais os clientes que irão receber determinada mensagem.

As mensagens trocadas entre dispositivos são leves e rápidas permitindo um menor consumo de energia. A mensagem é constituída por quatro parâmetros fundamentais: informação; Quality of Service (QoS); propriedade de coleções; e nome do tópico [50].

O cliente é um programa ou um dispositivo que utiliza o MQTT. O cliente inicia a rede de transporte de mensagens com o *broker*; publica mensagens que outros clientes podem estar interessados; subscreve mensagens que tem interesse em receber; remove a subscrição de mensagens que não tem interesse em receber; termina a rede de transporte de mensagens com o *broker*. O cliente pode ser ao mesmo tempo um cliente subscritor e cliente publicador [50]. Exemplo de clientes subscritores e publicadores referido na figura 3.4.

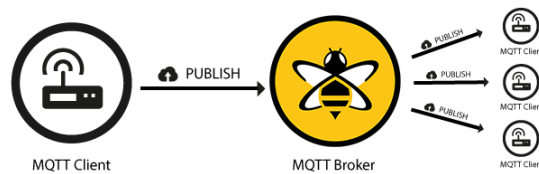


Figura 3.4: Clientes subscritores e publicadores [51].

O tópico permite identificar as mensagens que os clientes subscritores pretendem receber e é constituído por dois elementos: nome do tópico e o filtro do tópico. O nome do tópico é um identificador incluído no corpo da mensagem e o filtro do tópico é uma expressão contida no cliente subscritor que indica o interesse num ou mais tópicos [50]. Para o exemplo que se segue consideremos simplesmente o tópico como um só elemento. Por exemplo, se o cliente publicador enviar a mensagem 24°C através do tópico *temperature*, os clientes subscritores do tópico *temperature* recebem a mensagem 24°C. É uma excelente forma de colocar um dispositivo a receber somente as mensagens que lhes interessam. A figura 3.5 representa a arquitetura do protocolo MQTT, onde é possível visualizar os clientes, o *broker*, a mensagem e o conceito de publicação e subscrição do tópico *temperature*.

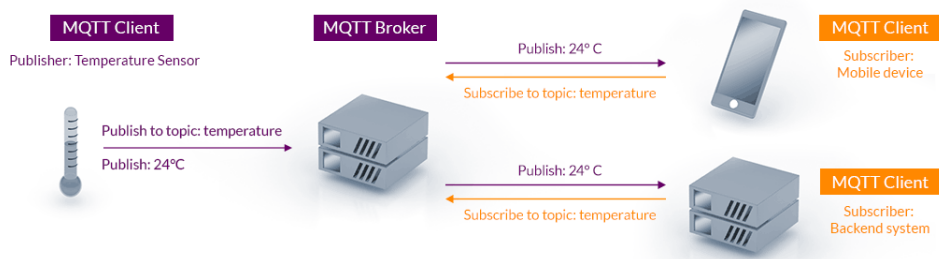


Figura 3.5: Arquitetura do protocolo MQTT [52].

O *broker* é o nó central no MQTT, sendo responsável por receber todas as mensagens enviadas pelos clientes publicadores, filtrá-las por tópicos e enviá-las aos clientes subscritores. Desta forma é possível endereçar e enviar as mensagens corretamente até ao seu destino. A figura 3.6 representa

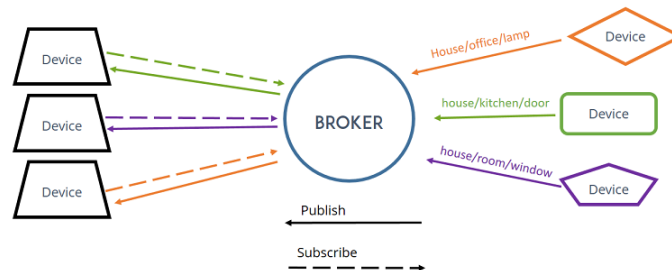


Figura 3.6: Princípio de funcionamento do protocolo MQTT [53].

Para implementar o protocolo MQTT é necessário implementar um broker. Existem alguns brokers que permitem a utilização do protocolo MQTT, entre eles: Mosquitto; EMQ X; Cassandra; Ejjaberd; e HiveMQ.

3.3.2 Constrained Application Protocol

O CoAP é um protocolo da camada de aplicação, especializado na transferência de dados entre a web e dispositivos. O protocolo é geralmente utilizado em aplicações M2M e foi desenvolvido para permitir que dispositivos e redes com características restritivas possam ser integrados num sistema IoT. Este protocolo permite a sua implementação em microcontroladores com capacidades bastante reduzidas como por exemplo de 10 KiB de *Random Access Memory* (RAM) e 100 KiB de espaço para código [54]. O CoAP tanto utiliza o padrão *request-response* como o padrão *publish/subscribe*. Utiliza o *User Datagram Protocol* (UDP) para transportar a informação entre dispositivos. Comparando com o MQTT, o CoAP tem um menor consumo de energia, mas em contrapartida tem menor fiabilidade no envio de dados.

O protocolo é similar ao modelo cliente servidor do *Hypertext Transfer Protocol* (HTTP), e tal como o HTTP é baseado no modelo *Representational State Transfer* (REST). A troca de informação entre os elementos deste protocolo é realizada através de pedidos e respostas. Sendo que os servidores recebem pedidos e respondem. Os clientes realizam os pedidos e recebem a resposta enviada pelo servidor. Os servidores disponibilizam os recursos através de um URL e os clientes acedem aos recursos utilizando métodos como GET, PUT, POST, e DELETE. Neste sistema os proxys podem agir como dispositivos finais numa rede CoAP de forma a realizarem a ligação entre uma rede HTTP e uma rede CoAP. O proxy, além de encaminhar os pedidos e retransmitir respostas, poderá também guardar pedidos e respostas

em memória e traduzir identificadores ou protocolos. Na figura 3.7 visualiza-se a arquitetura de comunicação do protocolo CoAP.

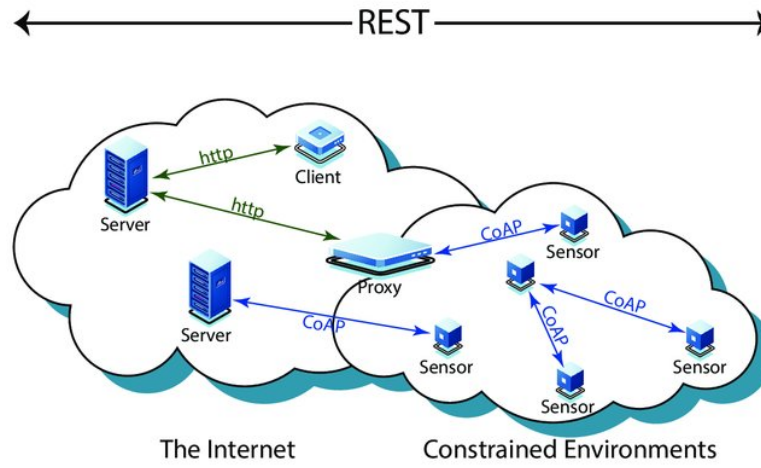


Figura 3.7: Princípio de funcionamento do protocolo CoAP [55]

Capítulo 4

Arquitetura do sistema

Neste capítulo é abordado o princípio de funcionamento do projeto, tendo como principal finalidade demonstrar as principais características de todas as partes constituintes do projeto bem como a sua aplicabilidade.

Na secção 4.1 é dado a conhecer a arquitetura do sistema e o seu diagrama funcional, abordando a componente IoT neste sistema. Na secção 4.2 é discutida a questão da alimentação do projeto. O sensor e o condicionamento de sinal são apresentados no capítulo 4.3. Neste capítulo são analisados os sensores e o circuito elétrico que integra o desenvolvimento do condicionamento de sinal de tensão e de corrente.

O capítulo 4.4 corresponde ao estudo realizado ao conversor analógico digital e está dividido em dois subcapítulos. O subcapítulo 4.4.1 compreende o estudo realizado à captura do sinal analógico. O subcapítulo 4.4.2 aborda o estudo sobre dois ADCs, o ADC integrado na placa de desenvolvimento DevKitC e o ADC externo MCP3002. As placas de desenvolvimento utilizados são analisados no capítulo 4.5. O display utilizado neste projeto é abordado no capítulo 4.6, sendo feita uma descrição do módulo display.

Este projeto integra vários protocolos de comunicação, sendo apresentados no capítulo 4.7. O servidor utilizado neste projeto é apresentado no capítulo 4.8. No capítulo 4.9 é apresentada a plataforma de desenvolvimento Node-RED. Por fim, no capítulo 4.10 é realizado um estudo sobre os cuidados a ter na conceção e desenvolvimento da PCB.

4.1 Arquitetura do sistema

Pretende-se com este projeto realizar a medição do consumo de energia numa carga, tendo em consideração os seguintes requisitos de sistema: taxa de amostragem no valor de 10KHz, medição em tempo real, apresentação dos dados de energia e capacidade de armazenar dados. O sistema é escalável, permitindo adicionar novos dispositivos de monitorização ou outros dispositivos atuadores ao sistema com o mínimo impacto possível. Como exemplo temos um segundo microcontrolador que tem como função controlar o estado de um *light-emitting diode* (LED).

A arquitetura do sistema de monitorização de energia IoT é apresentada na figura 4.1. Este sistema utiliza dois sensores para medir a tensão e a corrente elétrica instantânea e implementa um condicionamento de sinal de forma a adequar a amplitude de onda ao ADC. O microcontrolador captura amostras do sinal de onda a uma taxa de frequência de 10Khz e realiza os cálculos para obter a tensão, corrente e potência de uma determinada carga. Os dados são exibidos no display e transmitidos via WIFI para o servidor. Este servidor tem a responsabilidade de recolher os dados de todos os sensores da habitação, expor na interface gráfica e registá-los num ficheiro de texto. O segundo microcontrolador tem como objetivo controlar o estado de um LED, este controlo é realizado através da plataforma gráfica. A plataforma gráfica está dividida em dois grupos, estando um grupo destinado à monitorização dos dados de energia e o outro grupo destinado ao controlo do estado do LED.

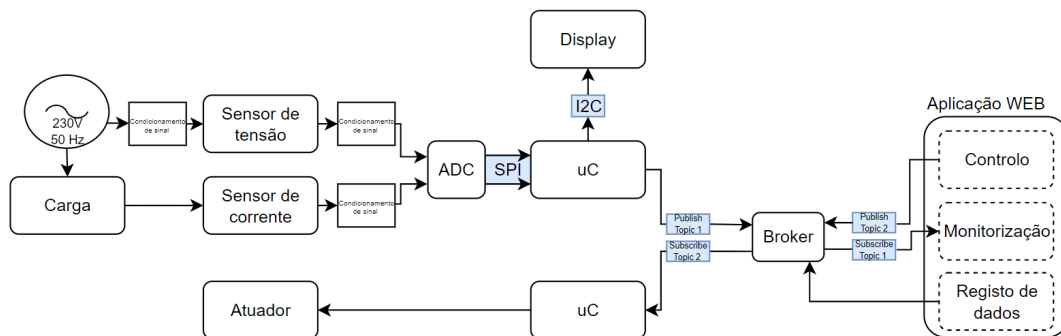


Figura 4.1: Arquitetura do sistema.

4.2 Alimentação do sistema

A tensão do sistema elétrico utilizado nas habitações em Portugal enquadra-se na gama de baixa tensão com dois tipos de níveis de tensão eficaz possíveis, 230V ou 400V. Este sistema foi estudado e desenvolvido para operar nos 230V tensão eficaz. O primeiro requisito técnico do sistema é efetuar leituras de cargas com tensão de alimentação de 230V eficaz. De acordo com a norma NP EN 50160 é possível ter uma variação de $\pm 10\%$ do seu valor de tensão eficaz. Sabendo que a tensão de alimentação é representada por uma onda sinusoidal teremos de ter em atenção ao

seu valor de pico. O valor de pico já com a tolerância de $\pm 10\%$ representa uma tensão máxima de pico no valor de 360V.

Estando o valor de tensão máximo definido a 360V é portanto, possível iniciar o desenvolvimento do circuito elétrico. Para fonte de alimentação utilizou-se o transformador de tensão HLK-PM01 da Hi-Link [56]. De acordo com a ficha técnica do fabricante a tensão de entrada varia entre 90 a 264 VAC, a corrente máxima de entrada é de 200 mA, à sua saída disponibiliza 5VDC e uma corrente máxima de 600 mA. Este transformador permite assim converter a tensão eficaz de entrada 230V AC em 5V DC com uma potência de 3W, disponibilizando 600mA de corrente máxima ao circuito elétrico. Com este transformador é possível fornecer potência suficiente para todo o sistema. No entanto, o sistema elétrico tem de ser construído por dois circuitos elétricos distintos, um circuito elétrico de potência e outro circuito elétrico de monitorização, sendo necessário isolar ambos os circuitos.

O isolamento do circuito de potência do circuito de monitorização permitirá alimentar e proteger o microcontrolador, o ADC e também permitirá alimentar de forma isolada parte do amplificador ACPL-785J. Para isolar ambos os circuitos foi utilizado o TMU 3-0511 [57]. Este componente isola os 5V de tensão de saída dos 5V de tensão de entrada, fornecendo no máximo 200mA à sua saída.

Outro ponto importante a ser considerado é a potência consumida pelo sistema. A fonte de alimentação tem como objetivo alimentar todo o sistema, e é pois essencial que esteja bem dimensionada. De forma a dimensionar a fonte de alimentação é necessário avaliar o consumo dos vários constituintes do sistema. Neste projeto foi medido o consumo do microcontrolador enquanto enviava dados através da comunicação WIFI e foi verificado um consumo máximo de 190mA. O consumo de potência do *display OLED* é de aproximadamente 20mA. Neste dois componentes temos um consumo de superior a 200mA. Para que a fonte de alimentação do sistema não esteja sub dimensionada é necessário escolher uma fonte de alimentação capaz de fornecer pelo menos 2W. A fonte de alimentação escolhida para o projeto foi a Hi-link 5V DC. A fonte é colocada no início do circuito ligada diretamente aos 230V AC do sistema elétrico.

4.3 Sensor e condicionamento de sinal

Foram escolhidos dois sensores, um para avaliar a corrente elétrica e outro para avaliar a tensão elétrica. O sensor escolhido para avaliar a corrente elétrica é uma resistência *shunt*. A escolha deste sensor deve-se, sobretudo, devido à baixa tolerância de erro, de aproximadamente 1%, e também devido ao seu baixo custo de aquisição. Na escolha do sensor *shunt* foi também tido em atenção o seu valor óhmico de aproximadamente 11 mOhms, pois quanto mais baixo for o seu valor óhmico mais baixa é a potência dissipada. O *shunt* escolhido admite uma corrente máxima

de 10 ampere e suporta tensões superiores à tensão da rede elétrica presente numa habitação, de aproximadamente 230V valor eficaz.

A figura 4.2 inclui três sub-figuras com o objetivo de apresentarem o comportamento da tensão/corrente num circuito com uma carga puramente resistiva. A figura 4.2a mostra a forma de onda da rede elétrica da habitação, na figura 4.2b é apresentada a implementação da resistência *shunt* em série com uma carga resistiva no circuito elétrico e na figura 4.2c é mostrada a relação direta entre a queda de tensão e a passagem de corrente elétrica na resistência *shunt*.

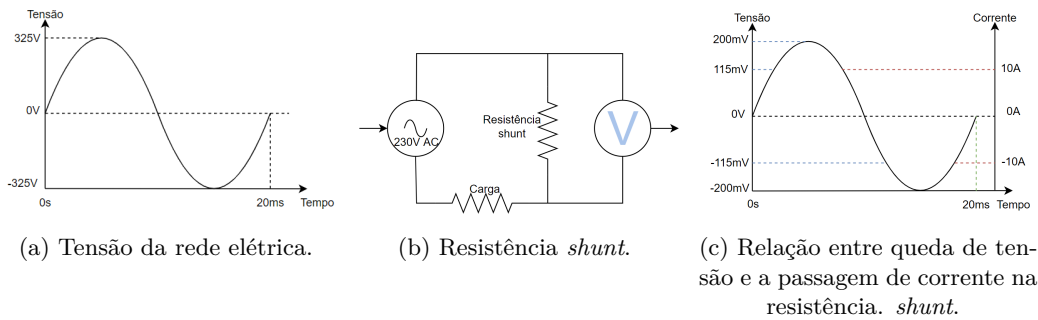
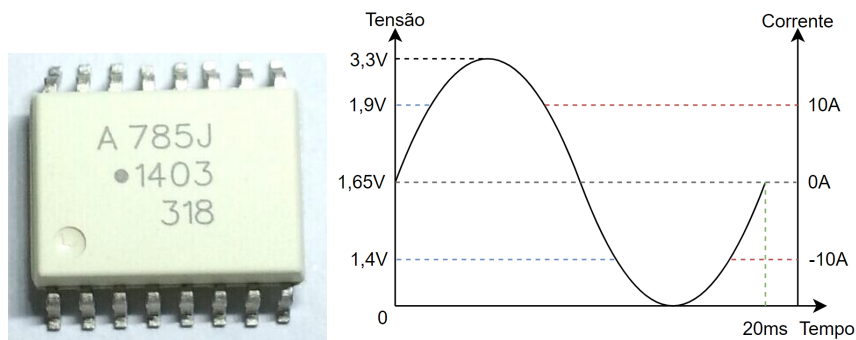


Figura 4.2: Implementação da resistência *shunt* no circuito de carga e respetiva relação tensão/corrente.

Para adequar a queda de tensão da resistência *shunt* aos valores de entrada do ADC é necessário realizar um condicionamento de sinal. Para realizar o condicionamento de sinal foi utilizado o componente ACPL-785J. O ACPL-785J é um amplificador isolado com deteção de curto circuito e sobrecarga, produzido pela Avago Technologies. Este permite isolar os circuitos de potência e controlo, realizar um ganho da tensão de entrada em 8 vezes e colocar a tensão média de saída 2,5 Volt acima da tensão média de entrada. De notar que a tensão de entrada aconselhada situa-se nos ± 200 mV, resultando numa tensão de saída com $\pm 2,4$ Volt com *offset* de 2,5 Volt.

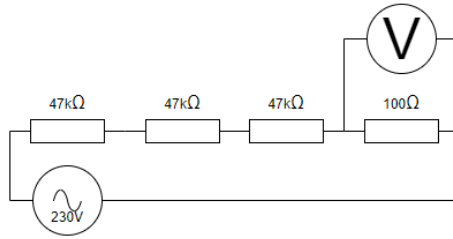
Na figura 4.3 verifica-se a relação entre tensão/corrente depois de aplicado o condicionamento de sinal. De salientar que quando a resistência é atravessada por 10A obtém-se aos seus terminais uma queda de tensão no valor de 110,6mV. Este valor está dentro do valor de operação recomendado do componente ACPL-785J, não sendo assim necessário realizar um tratamento de sinal adicional para o adequar à entrada do ACPL-785J. Depois do componente ACPL-785J é necessário aplicar um divisor de tensão de forma a reduzir os 5V para os 3,3V de tensão máxima. Os pinos do microcontrolador estão capacitados para operar com tensões máximas de 3,3V.



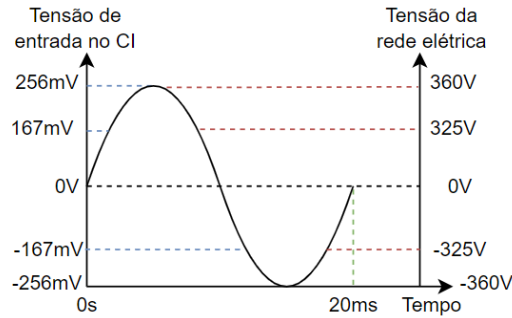
(a) Amplificador isolado. (b) Tensão de saída depois de aplicado o condicionamento de sinal.

Figura 4.3: Isolamento do circuito de potência com circuito de carga e respectivo condicionamento de sinal

Para avaliar a tensão na rede elétrica é necessário reduzir a tensão da rede para valores de trabalho adequados ao componente ACPL. A redução da tensão elétrica é definida por um conjunto de 4 resistências em série. Três das resistências são de $47k\Omega$, sendo a outra resistência de 100Ω . A figura 4.4 apresenta a implementação e o resultados da redução da tensão da rede elétrica para o valor de entrada adequado ao componente ACPL. Obteve-se o valor das 4 resistências de acordo com os seguintes definições: $360V$ é o valor máximo fornecido pela rede elétrica, $256mV$ é o valor máximo admitido pelo ACPL-785J e a potência consumida pelas resistências terá de ser mínima. Para isso, foram utilizadas resistências com uma potência máxima de $0,25W$. Sabendo que a lei de ohm é dada por $V = R * I$ e que o nosso V será de $256mV$, definimos na primeira fase como corrente $1mA$ e obtemos os 256Ω para a nossa resistência, sabendo que os valores das resistências são tabelados e que a potência máxima da resistência é $0.25W$ então poderemos reduzir o valor desta resistência para 100Ω . Sabendo o valor da resistência onde cairá os $256mV$ facilmente obtemos a corrente e descobrimos o valor das restantes resistências. Temos então os $2,56mA$ de corrente no ramo em série e uma tensão de $360V$ menos os $256mV$. Obtemos então o valor de $140,5k\Omega$.



(a) Implementação da redução da tensão da rede elétrica.



(b) Resultado da redução da tensão da rede elétrica.

Figura 4.4: Condicionamento de sinal e seu resultado para tensão da rede elétrica

Tal como na resistência *shunt* o componente ACPL-785J executará um ganho da tensão de entrada em 8 vezes e fará o *offset* de 2,5V. Elevando assim a tensão à saída do ACPL-785J para perto de 5V com a componente DC de 2,5V, como apresentado na figura 4.5.

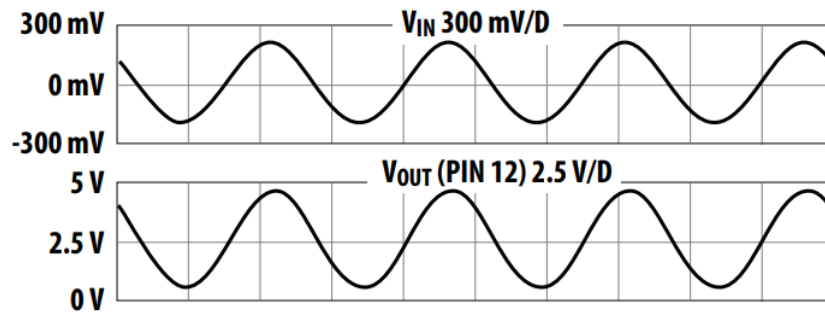


Figura 4.5: Tensão à saída do componente ACPL-785J [58].

4.4 Conversor analógico digital

Este capítulo tem como objetivo estudar e analisar a taxa de amostragem e os diferentes ADC possíveis de serem utilizados neste projeto. É abordado no subcapítulo

4.4.1 o estudo relativamente à taxa de amostragem versus erro associado e no sub-capítulo 4.4.2 é apresentada a análise efetuada a dois ADC, ao ADC integrado na placa de desenvolvimento e ao ADC MCP3002.

4.4.1 Captura do sinal analógico

O teorema de amostragem declara que para reconstituir o sinal original é necessário que a frequência de amostragem seja, no mínimo, igual ao dobro da frequência máxima do sinal original. No entanto, de forma a obter a menor taxa de erro entre o sinal original e o sinal reconstituído é necessário trabalhar com uma frequência de amostragem bastante superior à frequência do sinal original. A percentagem de erro associado à frequência de amostragem é obtida através da fórmula 4.1, onde $E(\%)$ representa a taxa de erro do sinal reconstituído, F_s , medido em Hertz, corresponde à frequência do sinal a amostrar e F_a , medido em Hertz, representa a frequência de amostragem.

$$E(\%) = \frac{F_s}{F_a} * 100. \quad (4.1)$$

Na tabela 4.1 está identificado a percentagem de erro associado ao número de amostras capturadas no período de 20ms. De notar, que com a frequência de amostragem na ordem dos 1000Hz são capturadas 20 amostras, tendo uma percentagem de erro associada na ordem dos 5%. Ao elevar a frequência de amostragem para os 10KHz, possibilita aumentar o número de amostras capturadas para 200 e reduzir o erro associado entre o sinal analógico e o sinal amostrado para 0,5%. Para que seja possível capturar as amostras com uma frequência de amostragem de 10KHz é necessário que o ADC capture e converta o sinal analógico em menos de 0,1ms, pois a cada 0,1ms é realizada uma nova captura do sinal analógico.

Tempo de amostragem (ms)	Freq. de amostragem (Hz)	Número de amostras num período de 20ms	Erro (%)
20	50	1	100
10	100	2	50
1	1000	20	5
0,1	10000	200	0,50
0,01	100000	2000	0,05

Tabela 4.1: Percentagem de erro relativa a diferentes frequências de amostragem.

De notar que nem todos os ADC estão capacitados para trabalhar com frequências de amostragem muito elevadas, existindo um compromisso em relação à frequência de captura do sinal e a sensibilidade da conversão do sinal. Temos como exemplo o ADC ADS1115 de quatro canais, 16 bits de resolução e taxa máxima de amostragem no valor de 860 *sample per second* (sps) [59].

4.4.2 Conversor analógico digital - ESP32 vs MCP3002

Este capítulo tem como objetivo analisar e caracterizar dois ADCs, o ADC do ESP32 e o MCP3002. O ADC do ESP32 tem vários pontos positivos, entre os quais se destacam: possibilidade de leitura através de dois canais em simultâneo; capacidade de capturar amostras à taxa de 200000 amostras por segundo; embutido no microcontrolador; e capacidade de resolução máxima de 12 bits. Como pontos negativos, a utilização do canal ADC2 é restrita pois partilha os recursos com a função WIFI, significando que não é possível utilizar o ADC2 se o WIFI estiver ligado [60]. Outro ponto negativo é a sua baixa taxa de amostragem quando ambos os canais são utilizados em simultâneo. Apesar do ADC ter alguns pontos positivos, noutros pontos ele falha e inviabiliza a ideia inicial do projeto.

Em relação ao MCP3002, este é um ADC com arquitetura de registo de aproximação sucessiva com 2 canais de 10 bits de resolução. A sua taxa de amostragem atinge os 200 ksps quando alimentado por 5V, no entanto, podemos obter uma taxa de amostragem na ordem dos 75 ksps se o alimentarmos com 2,7V. Utiliza a interface de comunicação SPI. Tem um consumo bastante baixo, consumindo 375 μ A no estado ativo [61].

A figura 4.6 compara de modo geral as características relevantes entre ambos os ADCs.

	ADC ESP32	MCP3002
Resolução máxima	12 bits	10 bits
Canal duplo	Sim	Sim
Interface	Interno	SPI
Alimentação	3,3V	2,7V-5,5V
Taxa de amostragem máxima com controlador RTC	200 ksps	200 ksps
Taxa de amostragem máxima com controlador DIG	2 Msps	-
Arquitetura	SAR	SAR
Consumo de corrente no estado ativo	---	375uA

Figura 4.6: Comparação entre o ADC do ESP32 e o ADC MCP3002.

A decisão de utilizar o MCP3002 deveu-se à impossibilidade de utilizar os dois canais do ADC do microcontrolador em simultâneo com o módulo WIFI. Sendo assim, o MCP3002 é estudado ao detalhe como se pode ver nos parágrafos seguintes.

De notar, que com o ADC MCP3002 perder-se-á alguma sensibilidade nas amostras capturadas, devido à resolução do ADC ser de 10 bits e devido aos pinos do ESP32 estarem limitadas a tensões máximas de 3,3V. No entanto, a redução da sensibilidade não terá um impacto significativo.

Para introduzir o MCP3002 será realizada uma descrição dos seus pinos. Como se verifica na figura 4.7 o MCP3002 apresenta 8 pinos. O pino 1 é designado de *Chip Select/Shutdown Input* e permite iniciar a comunicação quando é colocado a 0V, e quando colocado a 5V termina a conversão e coloca o dispositivo em modo de espera. O pino 2 e o pino 3 representam respetivamente a entrada analógica do canal 0 e do canal 1, estes canais podem ser utilizados no modo "*single-ended*" ou no modo "*single pseudo-differential*". Para este projeto será utilizado o modo "*single-ended*", pois pretende-se utilizar cada canal individualmente. O pino 4 representa o *ground*. O pino 5 é designado de entrada de dados série e é utilizado para transmitir os dados de configuração do canal de entrada. O pino 6 é designado de saída de dados série e tem como função transmitir os dados convertidos para o microcontrolador. O pino 7 designado de *serial clock* permite iniciar as conversões e para sincronizar cada bit à medida que a conversão ocorre. Por fim o pino 8 diz respeito à tensão de alimentação e tensão de referência.

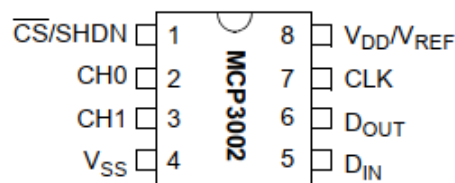


Figura 4.7: Definição dos pinos do ADC MCP3002 [61].

Através da ficha técnica do MCP3002 verificam-se os valores máximos possíveis para trabalhar com o ADC. Ora vejamos, a taxa de conversão anunciada pela ficha técnica diz-nos que o tempo de conversão necessita no máximo de 10 ciclos de relógio, presente na figura 4.8. Para obtermos este valor é necessário perceber qual será a taxa de amostragem utilizada e qual a frequência de operação do componente. A taxa de amostragem é obtida consoante a tensão de alimentação do ADC. Alimentando o ADC a 2.7V teremos uma taxa de amostragem de 75 ksps e se a alimentação for de 5V então teremos uma taxa de amostragem de 200 ksps. A ideia será tirar o máximo partido do ADC e por essa razão o mesmo será alimentado a 5V. Segundo a ficha técnica a frequência de relógio é obtida multiplicando a taxa de amostragem por 16. Desta forma obtemos assim uma frequência de relógio de 3.2 MHz. A frequência de relógio representa cada ciclo de relógio. Se o tempo de conversão necessita de 10 ciclos de relógio então podemos dizer que o tempo de conversão necessita de $3.125\mu\text{s}$. Outro tempo a ter em consideração é o tempo de amostragem, segundo a ficha técnica este tempo leva 1.5 vezes o ciclo de relógio, representando $0.468\mu\text{s}$. Como serão utilizados dois canais podemos concluir que o tempo teórico necessário para amostrar e converter as duas amostras é de aproximadamente $7.2\mu\text{s}$. De realçar que apesar de o MCP3002 ter dois canais de leitura a

amostragem e conversão são realizadas de forma sequencial.

All parameters apply at $V_{DD} = 5V$, $T_A = -40^\circ C$ to $+85^\circ C$, $f_{SAMPLE} = 200$ ksp/s and $f_{CLK} = 16 \cdot f_{SAMPLE}$, unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_A = +25^\circ C$, unless otherwise noted.						
PARAMETER	SYM	MIN	TYP	MAX	UNITS	CONDITIONS
Conversion Rate:						
Conversion Time	T_{CONV}	—	—	10	clock cycles	
Analog Input Sample Time	T_{SAMPLE}	1.5			clock cycles	
Throughput Rate	F_{SAMPLE}	—	—	200 75	ksp/s ksp/s	$V_{DD} = 5V$ $V_{DD} = 2.7V$

Figura 4.8: Características da taxa de conversão do MCP3002 [61].

É apresentado na tabela 4.2 o tempo necessário para converter os dados analógicos em dados digitais de cada entrada analógico. O tempo total necessário utilizando um canal é de $9.583\mu s$ e de dois canais é de $19.16\mu s$ para uma tensão de alimentação de $2.7V$. Para uma tensão de alimentação de $5V$ o tempo de conversão demora $3.593\mu s$ por canal, sendo que dois canais necessitam de $7.186\mu s$.

	Taxa de conversão $V_{DD} = 2.7V \rightarrow 75$ ksp/s	Taxa de conversão $V_{DD} = 5V \rightarrow 200$ ksp/s
Tempo de conversão	$8.333\mu s$	$3.125\mu s$
Tempo de amostragem da entrada analógica	$1.25\mu s$	$0.468\mu s$
Tempo total	$9.583\mu s$	$3.593\mu s$

Tabela 4.2: Taxa de conversão por canal analógico.

4.5 Microcontrolador

De modo a reproduzir um sinal analógico e reduzir o erro relativo à amostragem do sinal é necessário amostrar o sinal analógico com uma taxa de frequência de $10KHz$, sendo maior que a frequência do sinal a analisar que é de $50Hz$. Este requisito não só tem impacto na escolha do ADC, como também tem impacto na escolha do microcontrolador, pois a rápida aquisição e processamento do sinal analógico são essenciais. Outro ponto a ter em consideração na escolha do microcontrolador/placa de desenvolvimento deve-se aos requisitos iniciais deste projeto que contemplam a implementação da comunicação de dados sem fios. Por estes motivos decidiu-se utilizar o microcontrolador ESP32.

De realçar que neste projeto foram utilizados duas placas de desenvolvimento. Ambos as placas utilizam o ESP32, um microprocessador com dois *Central Processing Unit* (CPU) com arquitetura Harvard XTensa LX6, diferindo entre si somente

no tamanho das placas de desenvolvimento sendo a placa Mini WEMOS D1 de menor dimensão em comparação com a placa DevKitC V4.

4.5.1 ESP32 DevKitC V4

A placa de desenvolvimento ESP32 DevKitC V4 conta com um microcontrolador que inclui o microprocessador Xtensa® 32-bit LX6 de dois núcleos de baixa potência, com uma frequência de operação até 240 MHz. A sua memória interna contém 520KB de *Static Random Access Memory* (SRAM), 448KB de *Read-Only Memory* (ROM) e 16KB de *Real Time Clock* (RTC) SRAM, 8 KB para memória rápida e 8KB para memória lenta. Tem 34 *General Purpose Input/Output* (GPIO) programáveis entre os quais incluem-se funções como *Pulse Width Modulation* (PWM), ADC, *digital-to-analog converter* (DAC), I2C, SPI, *Universal Asynchronous Receiver/Transmitter* (UART). Este microcontrolador permite realizar comunicações *wireless* pois está equipado com um módulo *WiFi* e *Bluetooth*. O módulo *WiFi* suporta o padrão 802.11 b/g/n, suporta operações na faixa 2.4GHz e 5GHz, opera no modo Estação (cliente) ou no modo Ponto de Acesso (servidor), permite taxas de transmissão de dados até 150 Mbps e suporta como protocolos de segurança: *Wi-Fi Protected Access* (WPA)/WPA2-*Pre-Shared Key* (PSK), WPA/WPA2-*Enterprise*, e *Transport Layer Security* (TLS)/*Secure Sockets Layer* (SSL). Com o *Bluetooth* é possível realizar comunicações sem fios utilizando os padrões BBR, BEDR ou BLE. O ADC presente no microcontrolador tem 12 bits de resolução e uma taxa máxima de amostragem na ordem dos 200 ksp/s [62]. A placa de desenvolvimento utilizada é idêntica à apresentada na figura 4.9.

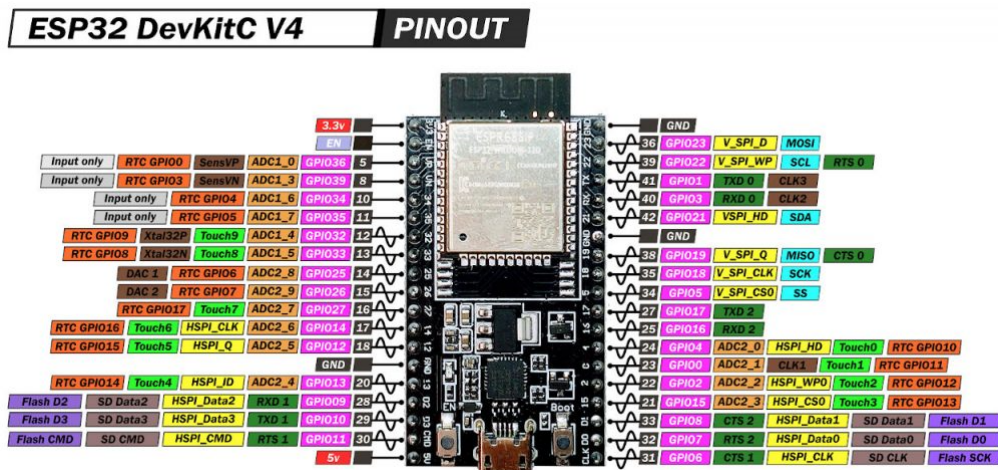


Figura 4.9: Placa de desenvolvimento ESP32-DevKitC V4.

Este display conta com quatro pinos. Dois pinos são utilizados para alimentação, VCC e *Ground* (GND). Os outros dois são utilizados para realizar a comunicação, sendo que o pino *Serial Clock* (SCL) determina a linha de relógio e o pino *Serial Data* (SDA) representa a linha utilizada para transmitir dados. Na figura 4.11 é possível ver o display utilizado no projeto.



Figura 4.11: OLED display 0.91"[64].

4.7 Protocolo de Comunicação de dados - I2C, SPI e WIFI

Neste capítulo serão abordados os protocolos de comunicação utilizados neste projeto. No primeiro subcapítulo é dada relevância ao protocolo de comunicação SPI, protocolo utilizado para a comunicação entre ADC e microcontrolador. No segundo subcapítulo é abordado o protocolo de comunicação I2C, utilizado para a comunicação entre o microcontrolador e o display. Por fim, no último capítulo é apresentado o protocolo de comunicação WIFI, utilizado para transmitir dados do microcontrolador para o servidor.

4.7.1 Protocolo de Comunicação SPI - Comunicação entre ADC e microcontrolador

De forma a estabelecer a comunicação com o ADC externo MCP3002 é necessário utilizar o protocolo de comunicação SPI [61].

A comunicação SPI é um protocolo de comunicação síncrono de dados série usado para realizar comunicações entre dispositivos designados de dispositivos mestre e escravo. O dispositivo mestre é normalmente um microcontrolador, enquanto os dispositivos escravos são designados de dispositivos periféricos, como por exemplo display, memórias externas, sensores e/ou cartões microSDs.

O microcontrolador ESP32 integra 4 controladores de SPI. O controlador SPI0 é utilizado como um *buffer* para aceder à memória externa. O controlador SPI1 pode ser utilizado como um mestre. Os controladores SPI2 e SPI3 podem ser configurados como mestre ou escravo. Os controladores SPI1, SPI2 e SPI3 compartilham

dois canais de DMA. Este protocolo conta com dois modos de comunicação, o modo *General Purpose SPI (GP-SPI)* e o modo *Parallel QSPI*. Para este projeto é utilizado o modo *General Purpose SPI (GP-SPI)* o que permite uma comunicação *full-duplex/half-duplex* de 4 linhas. A figura 4.12 apresenta a ligação elétrica entre dois dispositivos utilizando o protocolo de comunicação SPI com o modo *General Purpose SPI (GP-SPI)* numa comunicação *full-duplex/half-duplex*.

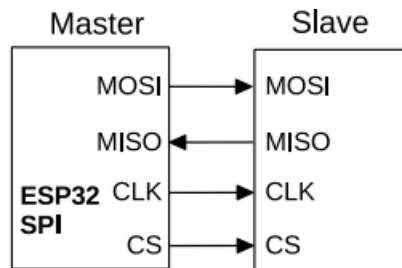


Figura 4.12: Protocolo de comunicação SPI com comunicação *full-duplex/half-duplex* [65].

A comunicação série utilizada pelo SPI utiliza 4 linhas e são definidas da seguinte forma:

- *Serial Clock (SCLK)*: É uma linha de relógio fornecida pelo mestre para sincronizar a transferência de dados.
- *Master Output, Slave Input (MOSI)*: É a linha pela qual o mestre envia os dados para o(s) escravo(s).
- *Master Input, Slave Out (MISO)*: É a linha pela qual o(s) escravo(s) envia(m) os dados para o mestre.
- *Chip Select (CS)*: É uma linha usada para selecionar o escravo com o qual o mestre deseja se comunicar.

O dispositivo escravo, neste caso o ADC (ver figura 4.7), utiliza o pino D_{in} para representar a linha MOSI e o pino D_{out} para representar a linha MISO. De forma a compreender como iniciar a comunicação com o ADC, primeiro é necessário perceber como configurar os bits de comunicação. Os bits de configuração são apresentados na figura 4.13.

	CONFIG BITS		CHANNEL SELECTION		GND
	SGL/ DIFF	ODD/ SIGN	0	1	
Single-Ended Mode	1	0	+		—
	1	1		+	—
Pseudo-Differential Mode	0	0	IN+	IN-	—
	0	1	IN-	IN+	—

Figura 4.13: Configuração de bits do ADC MCP3002 [61].

A comunicação de dados com o ADC MCP3002 no modo *Most Significant Bit* (MSB) pode ser compreendida através da figura 4.14. O início da comunicação é dado no flanco descendente do pino CS. O primeiro pulso do relógio recebido tendo o pino CS em baixo e o pino D_{in} em alto constituirá o bit de início. O bit SGL/DIFF e o bit ODD/SIGN seguem o bit de início e são utilizados para selecionar a configuração do canal de entrada.

O bit SGL/DIFF é utilizado para selecionar o modo *Single-Ended* ou *Pseudo-Differential*. O bit ODD/SIGN seleciona o canal a utilizar o modo *Single-Ended* e é utilizado para determinar a polaridade no modo *Pseudo-Differential*. Após o bit ODD/SIGN, o bit MSBF é transmitido e é utilizado para ativar o formato *Least Significant Bit* (LSB) primeiro para o dispositivo. Se o bit MSBF estiver em alto, então os dados serão recebidos do dispositivo no formato MSB primeiro e quaisquer outros pulsos de relógio com CS em baixo farão com que o dispositivo emita zeros. Se o bit MSBF estiver em baixo, então o dispositivo emitirá a palavra convertida LSB primeiro após a transmissão da palavra no formato MSB primeiro. O dispositivo começará a amostrar a entrada analógica no flanco ascendente do segundo pulso de relógio, após receber o bit de início. O período de amostragem terminará no flanco descendente do terceiro pulso de relógio após o bit de início.

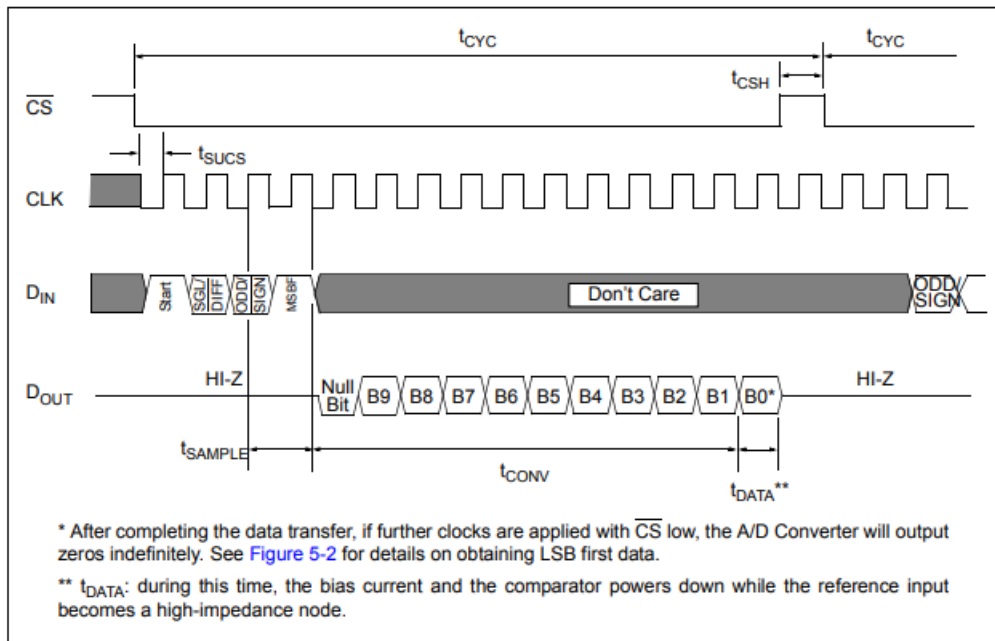


Figura 4.14: Comunicação SPI com o ADC MCP3002 [61].

Relativamente ao tempo de execução dos dados transferidos através do protocolo SPI do microcontrolador ESP32 podemos ter dois cenários distintos, um cenário onde a frequência máxima de transferência de dados é de 40 MHz e outro cenário onde a transferência atinge os 80 MHz. Estas diferentes velocidades de transmissão estão intrinsecamente relacionados com os GPIO utilizados. De acordo com a ficha técnica do intitulada *SPI Master Driver* [66], utilizando o *GPIO Matrix* a frequência do relógio de transmissão de dados fica limitada a 40 MHz, enquanto se utilizarmos os pinos *IO_MUX* então teremos a frequência de relógio limitada a 80 MHz. Como se pode ver na figura 4.15

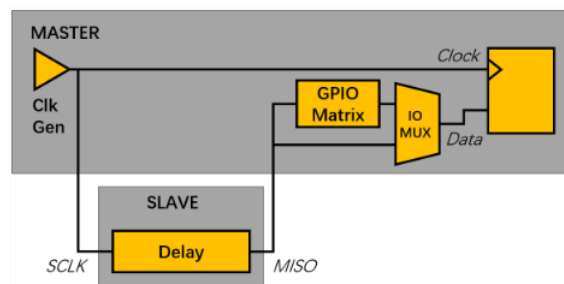


Figura 4.15: Considerações de tempo no protocolo SPI [66].

No entanto a velocidade de transmissão não será condicionada pelos limites máximos do ESP32, mas sim pelo limite máximo da frequência do relógio do ADC MCP3002. Segundo a ficha técnica do MCP3002, a frequência de relógio está limitada a 3.2 MHz. Este tempo de relógio de 3.2 MHz irá fazer com que os 10 bits transferidos do ADC para o microcontrolador demorem pelo menos 3,125µs.

4.7.2 Protocolo de Comunicação I2C - Comunicação entre micro-controlador e display

Concluído o processamento dos dados de energia capturados pelo ADC o micro-controlador transmite estes dados para o display e para um servidor onde serão apresentados ao utilizador. A transmissão dos dados do microcontrolador para o display é realizada utilizando o protocolo de comunicação I2C.

O protocolo de comunicação I2C tem dois modos distintos, o modo padrão e o modo rápido. O modo padrão permite atingir velocidades de comunicação de 100 Kbit/s, o modo rápido permite atingir velocidades de transmissão de 400 Kbit/s. O modo de endereçamento pode ser feito utilizando 7 bits ou 10 bits, sendo possível obter um modo de endereçamento duplo. Uma característica interessante desta placa de desenvolvimento é que o periférico I2C do ESP32 pode lidar com o protocolo I2C, libertando os núcleos do processador para outras tarefas.

Este protocolo define-se por utilizar um barramento de dois fios, composto por uma linha SDA e uma linha SCL. As linhas são compartilhadas por dois ou mais dispositivos: geralmente um ou mais mestres e um ou mais escravos. A linha SDA diz respeito à linha de dados, enquanto a linha SCL diz respeito à linha de relógio, permitindo sincronizar a transmissão de dados.

Abordadas as características anteriores, é importante compreender como é constituída e realizada a comunicação do protocolo I2C. Para isso é apresentada na figura 4.16 o formato de dados do protocolo I2C. A comunicação é iniciada com a condição de *start*. Esta condição é realizada colocando a linha SDA do estado alto para o estado baixo enquanto a linha SCL se mantém no estado alta. De seguida é transmitido um conjunto de 8 bits que dizem respeito ao endereço do dispositivo com o qual pretendemos comunicar. O endereço do dispositivo escravo é definido pelo byte 0111100x ou 0111101x, sendo que o segundo LSB diz respeito ao bit *Slave address bit* (SA0). O bit 0 diz respeito ao bit de escrita ou leitura, se for 0 é selecionado o modo de escrita, se for 1 é selecionado o modo de leitura. É recebido um sinal de confirmação depois de enviado um conjunto de 8 bits. Este sinal é enviado mesmo quando transmitimos o byte relativo ao endereço do dispositivo escravo. Depois de transmitido o endereço do dispositivo, é enviado um segundo byte que poderá ser um byte controlo ou um byte de dados. O byte de controlo consiste no bit Co (bit de continuação) e no bit D/C# (bit de seleção de dados/comando) seguido de 6 zeros. Quando o bit Co é colocado a 0 significa que é enviado um byte de dados. O bit D/C# quando colocado a 0 representa o envio de um byte de comando, se for colocado a 1 representa o envio de um byte de dados que serão guardados no *Graphic Display Data RAM* (GDDRAM), sendo que a GDDRAM representa um mapeamento de bits da RAM estática, guardando o padrão de bit a ser exibido no display. De seguida, é gerado um bit de confirmação após receber cada byte de controlo e dados. Por fim a comunicação termina com uma condição de *stop*, colocando

a linha de SDA do estado baixo para o estado alto enquanto a linha SCL permanece no estado alto [67].

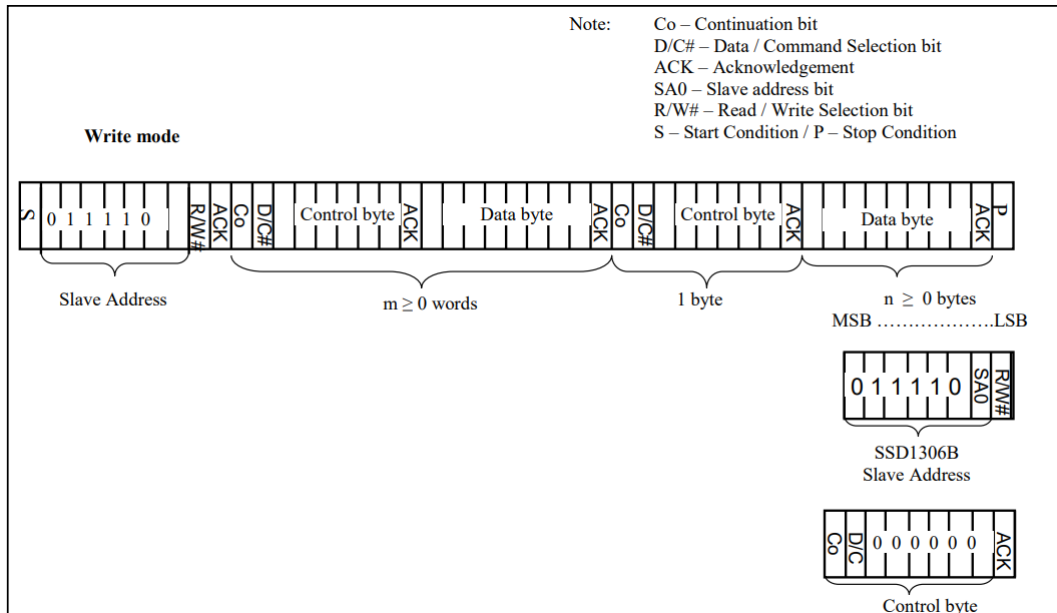


Figura 4.16: Formato de dados do protocolo I2C [67].

A transmissão de dados representada na figura 4.17 permite compreender o comportamento da transmissão de dados ao nível do bit.

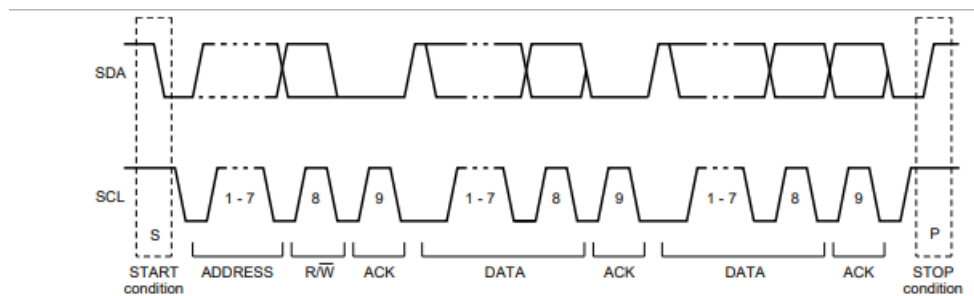


Figura 4.17: Gráfico da transmissão completa utilizando o protocolo de comunicação I2C [68].

Tal como referido no parágrafo anterior, a comunicação é iniciada quando um dispositivo mestre envia um sinal de *start*, colocando a linha SDA no estado baixo enquanto a linha SCL se mantém no estado alto. São enviados nove pulsos de relógio através da linha SCL. Os primeiros oito pulsos são usados para transmitir um byte composto por um endereço de 7 bits e um bit de leitura/escrita. Se um escravo com esse endereço estiver ativo no barramento, o escravo pode responder puxando a linha SDA para o estado baixo no nono pulso de relógio, como representado na figura 4.18. O mestre pode então enviar mais um conjunto de pulsos de relógio de 9 bits e, dependendo do bit de leitura/escrita enviado, o dispositivo mestre irá transmitir

dados na linha SDA, sendo que o outro lado confirma a transferência puxando a linha SDA para baixo no nono pulso de relógio. Durante a transferência de dados, a linha SDA só muda quando a linha SCL está baixa. Quando o mestre termina a comunicação, ele envia um sinal de *stop* para o barramento, levantando a linha SDA, enquanto a linha SCL se mantém no estado alto.

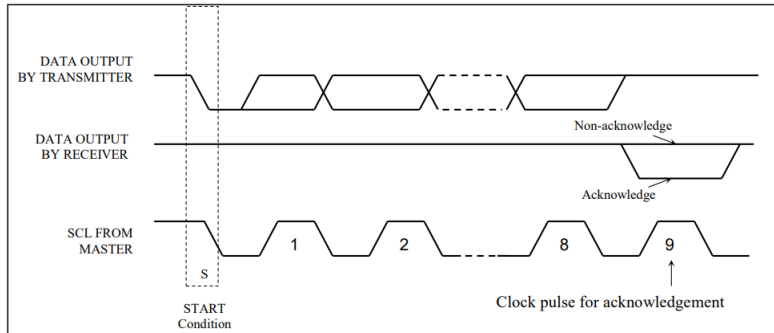


Figura 4.18: Definição da condição de confirmação [67].

4.7.3 Protocolo de Comunicação WIFI e MQTT - Comunicação entre microcontrolador e servidor

A transferência de dados do microcontrolador para o servidor será realizada através da tecnologia WIFI.

A WIFI tecnologia será utilizada de acordo com o módulo de comunicação presente no microcontrolador permitindo associar este dispositivo à rede doméstica. Por essa razão foram utilizados dois microcontroladores com módulo WIFI, um microcontrolador é utilizado para realizar a medição da rede elétrica numa carga e o outro microcontrolador é utilizado como atuador, acendendo um LED ao nosso comando. A inclusão dos dois microcontroladores na rede permite demonstrar a aplicabilidade do conceito IoT, sendo sempre possível adicionar mais sensores ou atuadores à rede.

O módulo WIFI do microcontrolador pode ser configurado em dois diferentes modos, modo Estação e modo Ponto de Acesso. O modo Ponto de Acesso permite providenciar uma rede de comunicação e o modo Estação o microcontrolador é utilizado como cliente da rede, conectando-se por exemplo à rede de internet e mostrando-se disponível para realizar comunicações de dados com outros dispositivos que se conectem à mesma rede. Na figura 4.19 é possível verificar os microcontroladores definidos no modo Estação, conectados a um *router*. O modo WIFI escolhido para este projeto é o modo Estação.

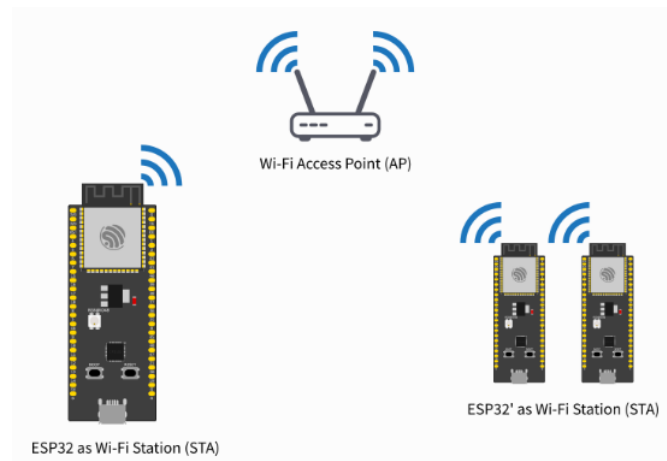


Figura 4.19: Microcontrolador com módulo WIFI definido no modo estação [69].

A configuração no Modo Estação é realizada de acordo com as informações técnicas disponibilizadas no site da Espressif Systems [70]. Algumas das configurações são demonstradas a seguir.

A função de configuração do wifi na listagem 4.1 é dada por:

```
1 esp_err_t esp_wifi_init(const wifi_init_config_t *config);
```

Listagem 4.1: Inicialização da configuração do módulo wifi

Esta função inicializa a alocação dos recursos para o *driver* do WIFI, como por exemplo a estrutura de controlo WIFI, *Transmit (TX)/Receive (RX) buffer*, estrutura NVS do WIFI. A macro `WIFI_INIT_CONFIG_DEFAULT` inicializa a configuração de valores no modo predefinido e o ponteiro `config` é utilizado para apontar a estrutura de configuração inicializada do WIFI.

A função listada na 4.2 permite definir o modo de operação do WIFI:

```
1 wl_status_t begin(const char* ssid, const char *passphrase = NULL,
    int32_t channel = 0, const uint8_t* bssid = NULL, bool connect
    = true);
```

Listagem 4.2: Definição do modo wifi

Para o modo de operação é possível escolher um de entre os seguintes modos disponíveis: Estação (`WIFI_MODE_STA`), Ponto de Acesso (`WIFI_MODE_AP`), Estação + Ponto de Acesso (`WIFI_MODE_APSTA`) ou NAN (`WIFI_MODE_NAN`). Sendo que o modo predefinido é o modo Estação.

A função listada 4.3 permite iniciar o módulo WIFI:

```
1 esp_err_t esp_wifi_start(void);
```

Listagem 4.3: Inicialização do módulo wifi

Inicia o WIFI de acordo com a configuração definida anteriormente. Sendo que esta função gera um bloco de controlo de estação e inicia a estação.

A listagem 4.4 apresenta a função que permite realizar a conexão ao ponto de acesso disponibilizado pelo *router*:

```
1 esp_err_t esp_wifi_connect(void);
```

Listagem 4.4: Conexão ao ponto de acesso da rede de internet

De notar que o protocolo de comunicação na camada física utilizado para estabelecer a comunicação entre os sensores/atuadores e o servidor é o protocolo WIFI, no entanto, também será utilizado como protocolo de camada de aplicação o MQTT, sendo que este foi escolhido por ser um protocolo leve e escalável, permitindo sempre que necessário adicionar novos dispositivos à rede. Na figura 4.20 é apresentada a arquitetura de comunicação entre os dispositivos.

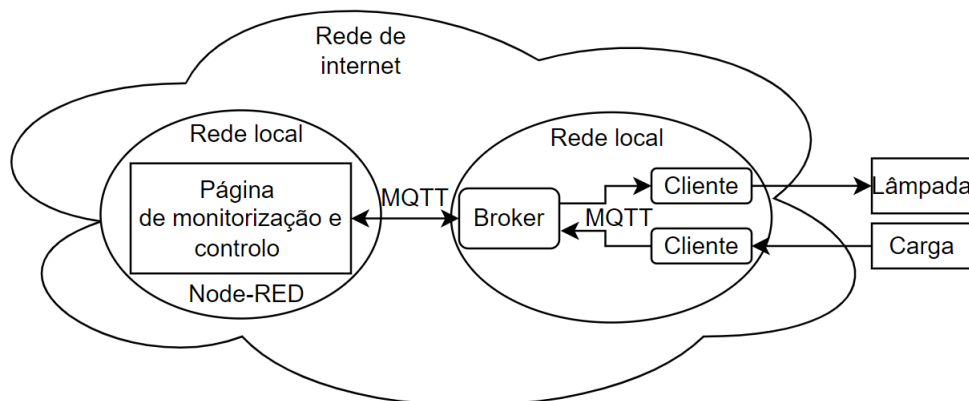


Figura 4.20: Arquitetura da comunicação.

No site da Espressif Systems é apresentada documentação relativa às *Application Programming Interface* (API) para a implementação do protocolo MQTT. As funções como configuração, inicialização, subscrição e publicação são definidas a seguir de acordo com a documentação das API exposta no site:

A função listada 4.5 cria um identificador de cliente MQTT com base na configuração predefinida, permitindo inicializar a estrutura referente à configuração do cliente MQTT.

```
1 esp_mqtt_client_handle_t esp_mqtt_client_init(const
    esp_mqtt_client_config_t *config)
```

 Listagem 4.5: Identificador de cliente MQTT

A função listada 4.6 inicia o cliente MQTT com o identificador de cliente já criado.

```
1 esp_err_t esp_mqtt_client_start(esp_mqtt_client_handle_t client)
```

Listagem 4.6: Inicia o cliente MQTT com o identificador de cliente

A função listada 4.7 permite ao cliente subscrever dados do tópico com QoS definido.

```
1 int esp_mqtt_client_subscribe(esp_mqtt_client_handle_t client,
    const char *topic, int qos)
```

Listagem 4.7: Função para subscrever dados do tópico

A função listada 4.8 permite ao cliente publicar dados num tópico.

```
1 int esp_mqtt_client_publish(esp_mqtt_client_handle_t client, const
    char *topic, const char *data, int len, int qos, int retain)
```

Listagem 4.8: Função para publicar dados no tópico

4.8 Servidor

O Raspberry Pi 3 Model B+ é um mini computador capaz de executar um sistema operativo. Permite a utilização de periféricos como rato, teclado e monitor e devido à sua versatilidade permite que este dispositivo de pequenas dimensões possa ser utilizado como um módulo central num projeto de IoT.

Neste caso o Raspberry Pi é utilizado como mediador na receção de dados enviados através do protocolo de comunicação MQTT pela placa de desenvolvimento.

O Raspberry terá como função receber os dados através dos diferentes tópicos e exibi-los numa plataforma gráfica intuitiva e de fácil utilização.

Relativamente às características técnicas do Raspberry, este possui um processador quad-core Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz, 1GB LPDDR2 SDRAM, 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet através da porta USB 2.0 (máximo de 300 Mbps), porta *High-Definition Multimedia Interface* (HDMI), 4 portas USB 2.0 e porta Ethernet. É alimentado

através de 5V/2.5A DC. A figura 4.21 representa o Raspberry Pi 3 modelo B+ utilizado no projeto.



Figura 4.21: Raspberry Pi 3 modelo B+.

4.9 Node-RED

Para que o Raspberry Pi tenha acesso aos dados de energia através do protocolo MQTT foi necessário instalar a ferramenta Mosquitto *Broker*. Esta ferramenta é *open-source* e está disponível no site da Eclipse Mosquitto [71]. O Mosquitto é uma aplicação leve e muito útil para sistemas IoT utilizando o modelo *publish/subscribe* para transmitir dados. Depois da instalação do *Broker* é necessário utilizar uma ferramenta que permita mostrar ao utilizador os dados de energia. A ferramenta escolhida para realizar a interface com o utilizador é o Node-Red.

O Node-RED é uma ferramenta de programação baseada em fluxos, desenvolvida pelo grupo *Emerging Technology Services* pertencente à empresa IBM. A programação baseada em fluxo permite separar o programador das linhas de código, tornando a programação, leitura e execução do código mais intuitiva, pois a programação em fluxo é composta por nós ligados entre si, com cada nó a ter uma função específica. Os dados são enviados para um nó, esse nó processa os dados de acordo com a sua função e envia os dados processados para o nó seguinte ou para outra aplicação ou dispositivo. Apesar de a programação em fluxo tentar separar o programador das linhas de código, isso nem sempre será possível, pois os blocos ou nós precisam de ser parametrizados e também programados quando necessitamos de desenvolver novas funções. A linguagem de programação utilizada no Node-Red é javascript.

Define-se o Node-RED como uma programação de baixo nível, permitindo conectar dispositivos de *hardware*, APIs e serviços *online* baseado em fluxo ou ligação de nós [72]. A figura 4.22 mostra um exemplo da linguagem de programação Node-RED.

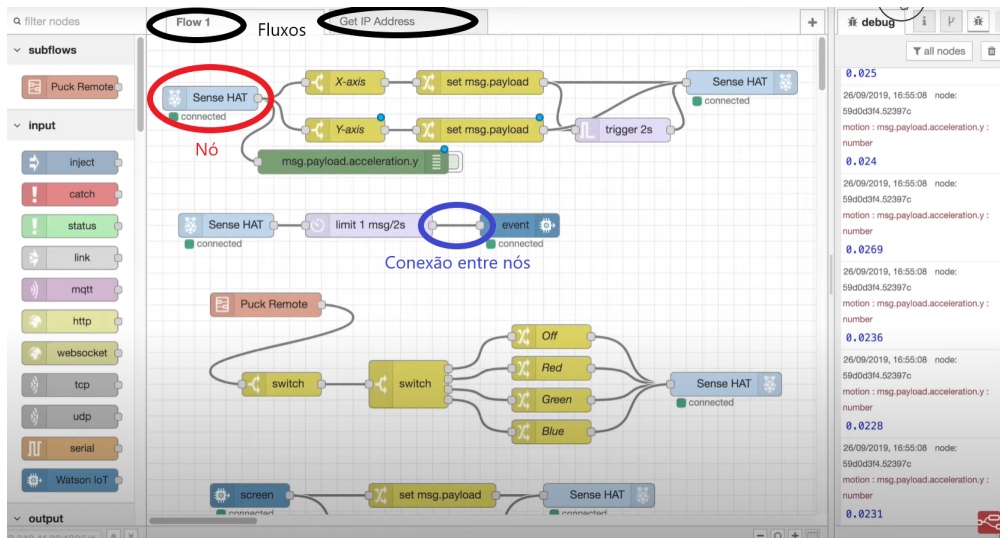


Figura 4.22: Linguagem de programação Node-RED [72].

A implementação do código em fluxo foi realizado tendo por base os tutoriais expostos no site do Node-RED. De seguida serão apresentadas algumas funções implementadas em fluxo necessárias à receção dos dados de energia provenientes do microcontrolador, bem como as funções que permitem tratar esses dados e disponibiliza-los tanto na interface gráfica como num ficheiro de texto.

Por exemplo, a figura 4.23 apresenta a utilização de um nó utilizado para subscrever um tópico e receber as mensagens desse tópico.

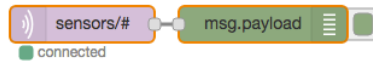


Figura 4.23: Nó de subscrição de tópico [72].

Depois de inserir o nó é necessário parametriza-lo, adicionando os dados do servidor e do tópico a subscrever, como apresentado na figura 4.24.

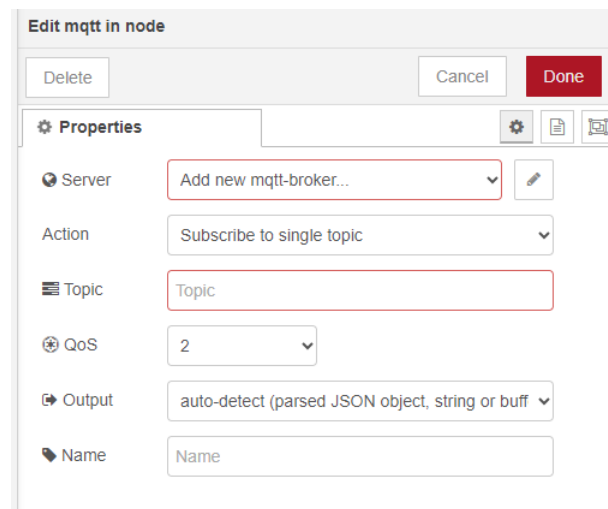


Figura 4.24: Parametrização do nó de subscrição de tópico [72].

Para publicar uma mensagem no tópico, é necessário adicionar o nó de publicação de tópico e parametrizar adicionando o tópico no qual pretendemos transmitir a mensagem. A figura 4.25 apresenta o nó de publicação e a sua parametrização.

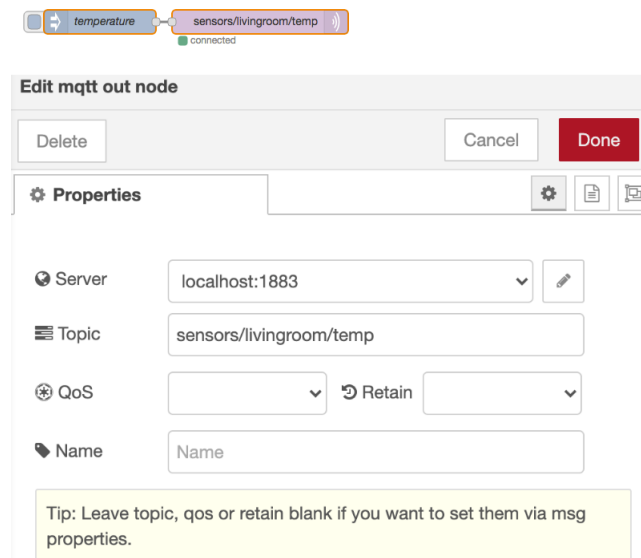


Figura 4.25: Parametrização do nó de publicação de tópico [72].

4.10 Cuidados a ter no desenvolvimento da PCB

Neste subcapítulo serão abordados os principais pontos relativamente ao desenvolvimento da PCB. Um dos aspetos mais importantes no desenvolvimento da PCB é a minimização do ruído. O ruído pode ser categorizado em três tipos diferentes, como o ruído dispositivo, o irradiado e o condutivo. O ruído do dispositivo pode ser gerado pelo aquecimento de uma resistência ou pela comutação de um transístor, uma fonte

de alimentação também é geradora deste tipo de ruído devido ao ruído associado às comutações rápidas internas. O ruído irradiado é originado por fontes *Electromagnetic Interference* (EMI) como por exemplo, motores ou interruptores. Estas fontes são responsáveis por adicionar ruído ao sinal analógico. O ruído condutivo está presente nos caminhos condutivos do circuito, como por exemplo os caminhos de sinais de uma PCB. Este tipo de ruído é obtido através de ruído do dispositivo ou de ruído irradiado [73].

O ruído originado pelos dispositivos, irradiados ou condutivos são visualizados na figura 4.26.

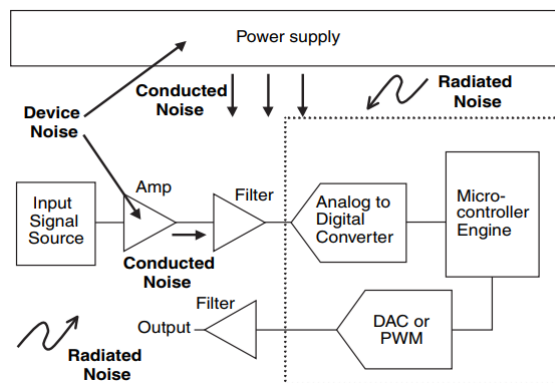
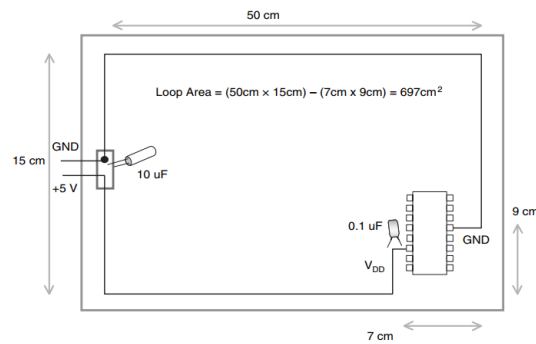


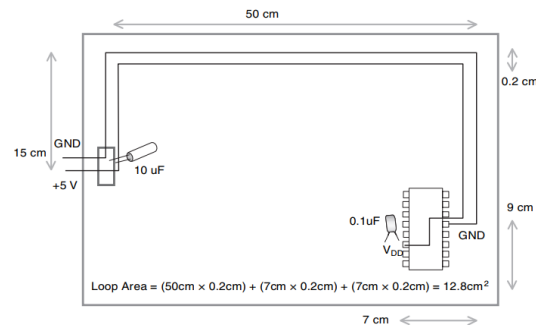
Figura 4.26: Exemplo da aplicação de circuito analógico e o ruído associado [73].

Estes ruídos devem ser tratados ou minimizados de forma a obter um sinal analógico o mais limpo possível de ruídos. Algumas técnicas devem ser utilizadas, como por exemplo a escolha de dispositivos com baixo ruído associado, ou por exemplo o simples facto de definir a localização dos dispositivos na PCB, separando-os por componentes digitais ou componentes analógicos.

Por exemplo a alimentação elétrica e o *ground* devem ser colocados próximos um do outro. Ou seja, devem seguir o mesmo caminho para minimizar as EMI e garantir o bom funcionamento do sistema. A figura 4.27a mostra uma implementação das linhas de energia suscetíveis de criar maior EMI, tendo neste caso uma área de *loop* de 697 cm². Este *loop* é uma antena perfeita para captar ruído EMI. Isso significa que a placa pode captar sinais de rádio ou outras fontes de ruído elétrico [73]. Para reduzirmos o ruído EMI é necessário reduzir a área de *loop*. Para isso podemos implementar os condutores de alimentação próximos um do outro, como apresentado na figura 4.27b. Outra solução para reduzir os ruídos EMI seria ter um plano de *ground*, que ficaria por baixo do caminho de alimentação de energia, ou então ter um plano de *ground* e um plano de alimentação de energia separados.



(a) Implementação da alimentação e terra numa PCB sujeita a interferências elétricas.



(b) Implementação da alimentação e terra numa PCB minimizando as interferências elétricas.

Figura 4.27: Implementação da alimentação e terra numa PCB [73].

Outro ponto relevante a ter em consideração está relacionado com o dimensionamento das trilhas de sinal. As trilhas de sinal na placa (tanto digitais como analógicas) devem ser o mais curtas possível. Esta regra básica minimizará as oportunidades para sinais estranhos se acoplarem no caminho do sinal. Os pinos de entrada dos dispositivos analógicos também devem ser tidos em conta, isto porque normalmente têm uma impedância mais alta do que os pinos de saída ou de alimentação. Como exemplo, o pino de entrada de referência de tensão para o ADC é mais sensível quando está a ocorrer uma conversão [73].

Outro ponto a ter em atenção e muito importante no desenvolvimento do circuito elétrico é a implementação de condensadores de desacoplamento na entrada da fonte de alimentação e o mais próximo possível das entradas dos dispositivos eletrónicos. Os terminais dos condensadores, também, deverão ser os mais curtos possíveis de modo a reduzir a indutância. Na figura 4.28 é mostrado um exemplo da correta implementação de condensadores de desacoplamento numa PCB [73].

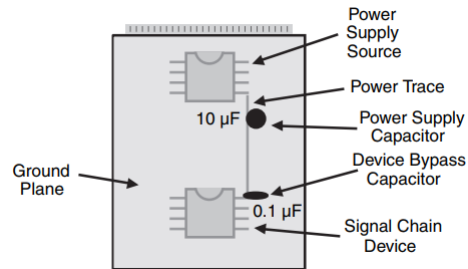
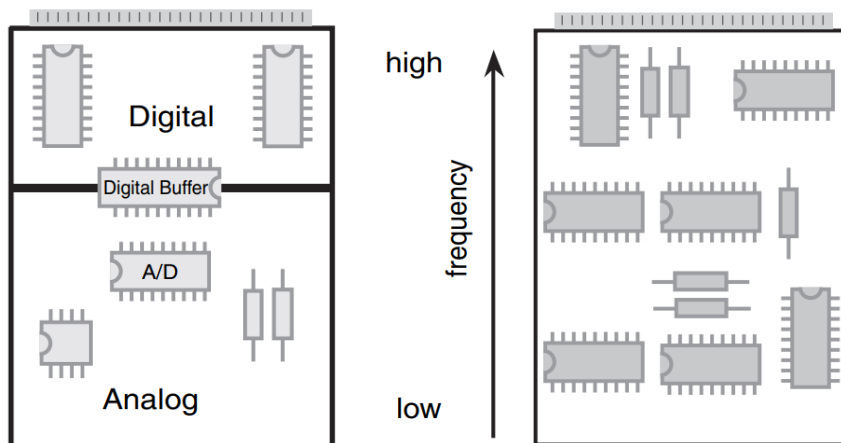


Figura 4.28: Exemplo da correta implementação de condensadores de desacoplamento [73].

Um outro tópico relativamente ao ruído prende-se com os dispositivos analógicos e digitais. Isto porque os dispositivos digitais são geradores de ruído, e ao mesmo tempo são menos sensíveis a estes ruídos devido às grandes margens de ruído de tensão. Os circuitos analógicos têm margens de ruído de tensão menores que os circuitos digitais. Assim sendo temos os circuitos analógicos mais sensíveis ao ruído causado pelas rápidas comutações presentes nos dispositivos digitais. Por esta razão devemos separar os dois domínios. A figura 4.29a apresenta a correta localização dos componentes eletrónicos analógicos e digitais na PCB. A figura 4.29b apresenta um *layout* em que os componentes eletrónicos de alta frequência devem estar o mais próximo possível dos conectores de entrada.



(a) Localização dos componentes de acordo com o tipo analógico ou digital. (b) Localização dos componentes de acordo com a sua frequência de trabalho.

Figura 4.29: Localização dos componentes na PCB [73].

Capítulo 5

Implementação dos sistemas de Hardware e Software

Este capítulo tem como objetivo abordar o desenvolvimento e implementação do projeto do sistema de monitorização de energia. É abordada a implementação dos componentes de *hardware*, como por exemplo a alimentação do sistema e o condicionamento de sinal, e também aos componentes de *software*, onde é explicado como se definiu e implementou o código fonte, como por exemplo o código que realiza o processamento dos dados de energia capturados.

Para ajudar a entender como foi desenvolvido o projeto é apresentada a figura 5.1. O projeto é decomposto em dois sistemas, um sistema é dedicado ao *hardware* e o outro dedicado ao *software*. Cada sistema está dividido em cinco blocos funcionais. Esta desconstrução do sistema em blocos funcionais permite não só, um maior controlo sobre o estudo e desenvolvimento como também, permite facilitar a exposição de todo o projeto.

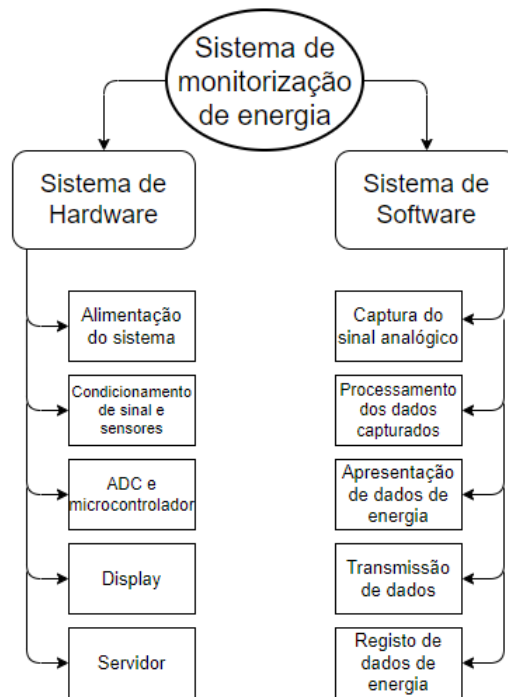


Figura 5.1: Sistema de monitorização de energia desmontado em blocos funcionais.

O desenvolvimento do projeto iniciou-se pelo sistema de *hardware*, sendo apresentado no capítulo 5.1. O sistema de *hardware* representa a implementação física do projeto, realizando a implementação do circuito elétrico de todo o sistema. O primeiro bloco a ser pensado e desenvolvido foi o bloco Alimentação do sistema. Depois desenvolveu-se o circuito elétrico para o condicionamento de sinal, sensores, ADC, microcontrolador e display.

No capítulo 5.2 é explicado o desenvolvimento do sistema de *software*. Onde as implementações foram realizadas por blocos funcionais. Sendo o primeiro bloco referente ao desenvolvimento da função de captura das amostras, o segundo bloco é referente à função de processamento dos dados capturados, o terceiro bloco é referente à exibição dos dados de energia consumidos em tempo real, no quarto bloco desenvolveu-se a função para transmitir os dados, e no quinto e último bloco, são apresentados e registados e os dados de energia de forma a ter um histórico de dados.

5.1 Implementação do *Hardware*

Neste capítulo iremos expor os cinco blocos funcionais referentes ao sistema de *hardware*. O primeiro bloco faz referência à alimentação do sistema. Seguido da apresentação ao circuito elétrico relativo ao condicionamento de sinal e dos sensores. O

terceiro bloco aborda a implementação do circuito elétrico do ADC e do microcontrolador. No quarto e quinto bloco, são apresentadas as implementações do circuito elétrico do display e do servidor, sendo que neste último caso a implementação do circuito é simples pois o servidor só estará alimentado a 5VDC. Por fim, é apresentado um capítulo referente ao desenvolvimento e implementação da PCB.

Antes de iniciarmos a apresentação da implementação da alimentação do sistema serão apresentadas duas figuras relativas ao circuito elétrico do sistema de monitorização de energia. Estando o circuito elétrico dividido em duas partes, a parte analógica e a parte digital e está representado, respetivamente, pelas figuras 5.2 e 5.3.

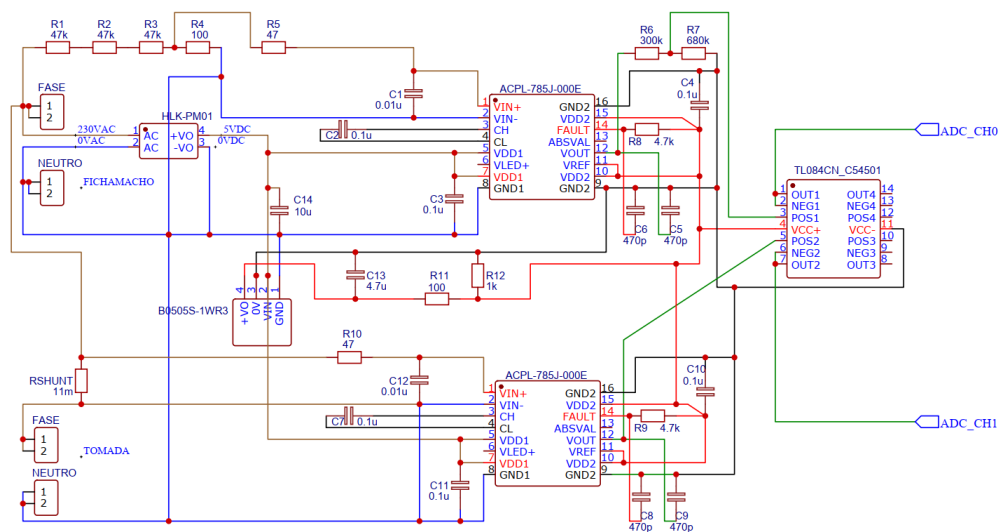


Figura 5.2: Esquema elétrico - Circuito analógico.

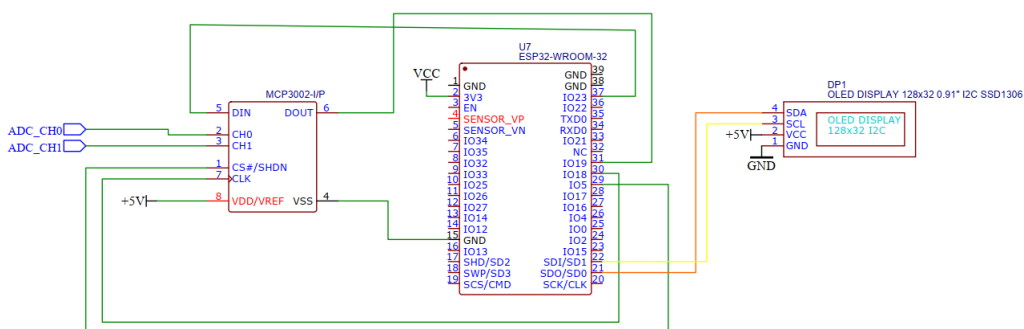


Figura 5.3: Esquema elétrico - Circuito digital.

Neste projeto foi adicionada uma segunda placa de desenvolvimento para demonstrar que é possível adicionar outros dispositivos à nossa rede sem impactar o sistema de monitorização previamente instalado. Esta placa conecta-se à rede local e permite transmitir dados através do protocolo de comunicação MQTT para o servidor, mostrando-os na página de Internet. Através da página de controlo é possível

alterar o estado do LED. O circuito elétrico da segunda placa de desenvolvimento utilizada neste projeto é apresentada na figura 5.4.

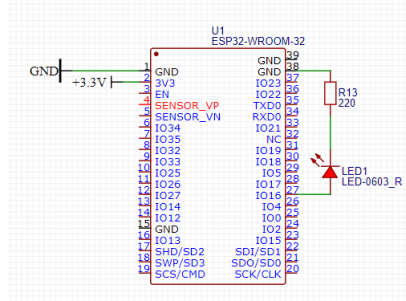


Figura 5.4: Esquema elétrico - segunda placa de desenvolvimento.

5.1.1 Implementação da Alimentação do Sistema

O circuito elétrico do sistema de *hardware* foi implementado em várias partes. A primeira parte é relativa à alimentação do sistema, onde na figura 5.5 é possível verificar, a traço grosso azul e castanho, a alimentação de 230VAC. Os 230VAC alimentam um circuito paralelo onde entram as resistências em série R1, R2, R3 e R4. Este circuito paralelo permite obter o sinal de tensão reduzido ao valor de trabalho adequado do componente ACPL-785J, que é de no máximo 220VDC. A resistência *shunt* é alimentada num circuito em série permitindo obter o sinal relativo ao valor corrente, sendo que o valor de corrente é proporcional ao valor de queda de tensão.

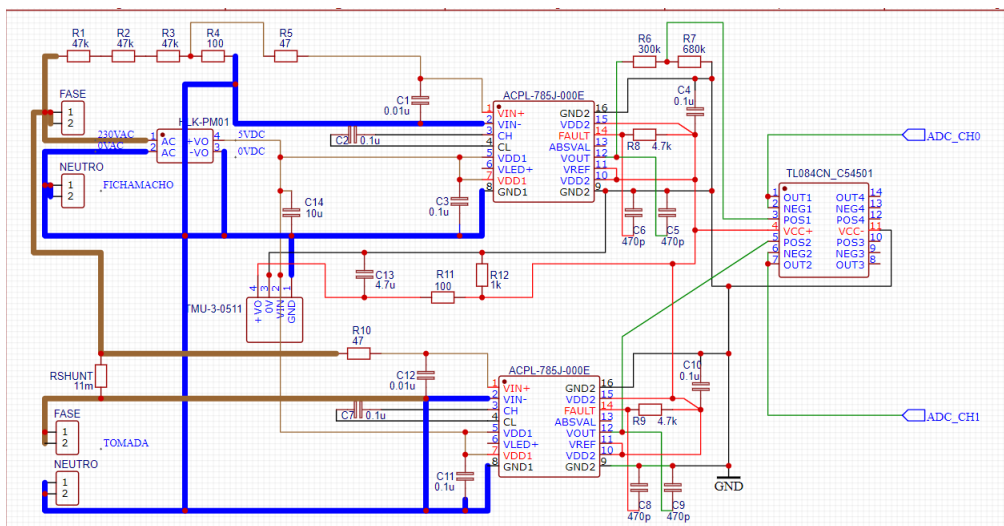


Figura 5.5: Representação a traço grosso do circuito de alimentação do sistema a 230VAC.

Os 230VAC são reduzidos a 5VDC no transformador HLK-PM01. Estes 5VDC são isolados no componente TMU-3-0511 sendo a parte não isolada utilizada para

alimentar a parte não isolada dos componentes ACPL-785J e a parte isolada para alimentar vários componentes, como a parte isolada do componente ACPL-785J, ADC, microcontrolador, display e o amplificador operacional utilizado como *buffer*. A figura 5.6 representa a traço grosso a alimentação do sistema de 5VDC.

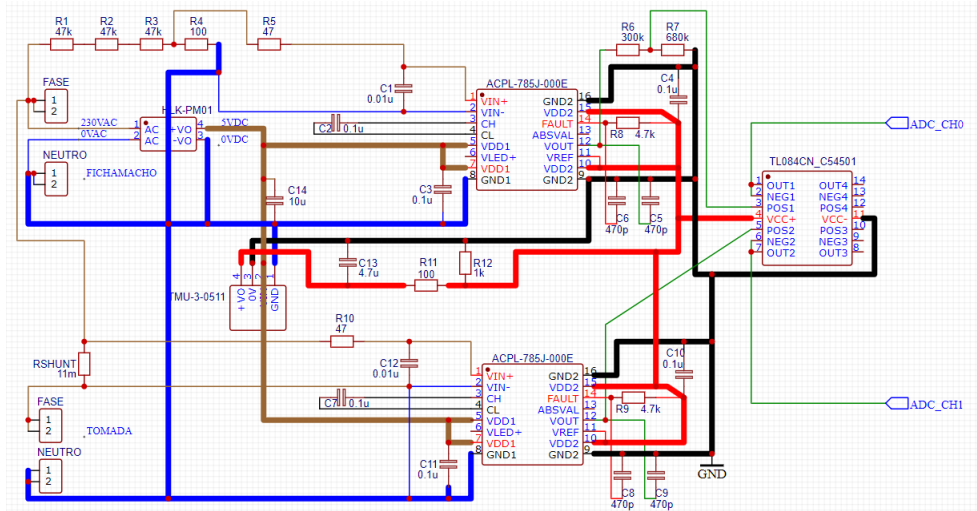


Figura 5.6: Representação a traço grosso do circuito de alimentação do sistema a 5VDC e a 3.3VDC.

5.1.2 Implementação do Condicionamento de sinal e sensores

Relativamente à implementação do sensores e condicionamento de sinal realizado, este é possível de analisar na figura 5.7. O ACPL-785J recebe o sinal de tensão e o sinal de corrente na sua entrada VIN+ e trata de ampliar até ao valor máximo de 5V e de fazer o *offset* em 2,5VDC, permitindo assim colocar a onda sinusoidal do sinal na parte positiva. As resistências R6 e R7 permitem reduzir a amplitude da tensão do sinal de saída do ACPL-785J, reduzindo o valor máximo de 5VAC para 3.3VDC. Os condensadores de desacoplamento foram adicionados de acordo com as recomendações da ficha técnica do ACPL-785J.

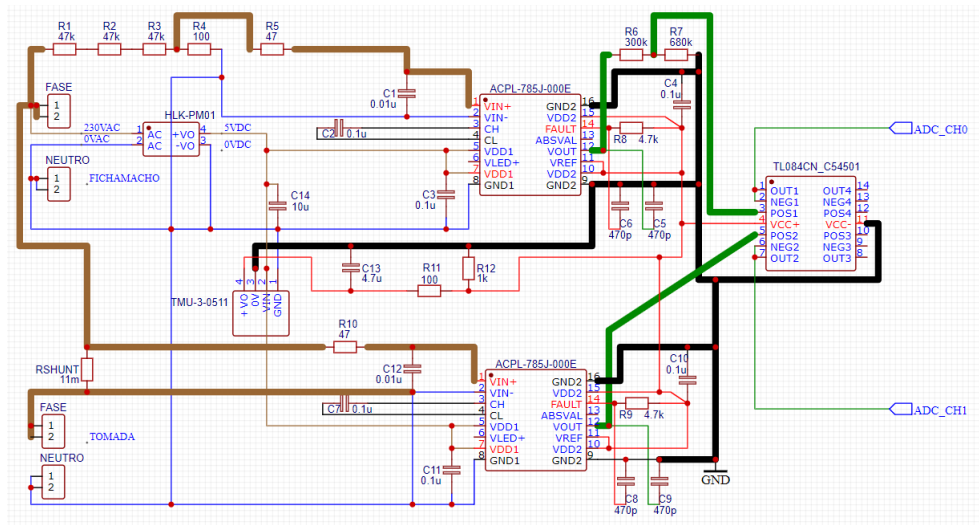


Figura 5.7: Representação a traço grosso do circuito elétrico do condicionamento de sinal e sensores.

5.1.3 Implementação do ADC e microcontrolador

A implementação do circuito elétrico do bloco ADC e microcontrolador é apresentada a traço grosso verde na figura 5.8. Depois do sinal passar pelo componente ACPL-785J e ser estabilizado pelo TL084 é conectado às entradas do canal 0 e canal 1 do ADC. O ADC captura as amostras analógicas e transforma-as em amostras digitais, disponibilizando-as nas entradas do microcontrolador, através do pino do ADC DOUT para o pino do microcontrolador IO19 (MISO). A comunicação do microcontrolador para o ADC é estabelecida com a ligação entre os pinos DIN e IO23 (MOSI). O pino CS do ADC conecta ao pino IO5 do microcontrolador, é através desta ligação que é permitido ao microcontrolador iniciar a comunicação com o ADC. O pino CLK liga ao pino IO18, permitindo ao microcontrolador iniciar a comunicação e recolher os dados amostrados pelo ADC. A alimentação do microcontrolador é realizada através dos 3.3VDC isolados.

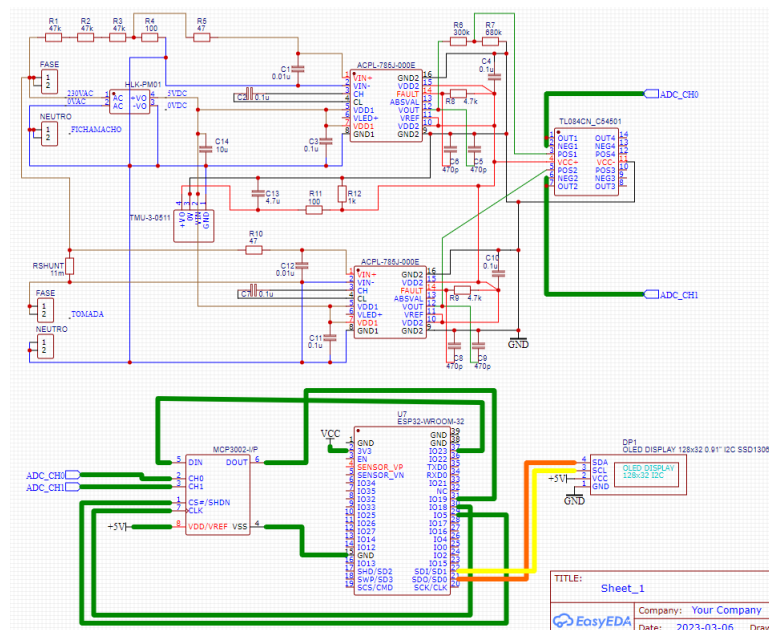


Figura 5.8: Representação a traço grosso do circuito elétrico do ADC e microcontrolador.

5.1.4 Implementação do display e servidor

A implementação do circuito elétrico do bloco display é apresentado a traço grosso na figura 5.9. A transmissão de dados ocorre na conexão entre as linhas SDI do microcontrolador e SCL do display, que representam o sinal do relógio, e também a ligação da linha SDO do microcontrolador e SDA do display, que representa a transmissão de dados entre os dispositivos. O display é alimentado a 5VDC isolados. O servidor foi alimentado ligando-o a um transformador que converte 230VAC para 5VDC e ligado à rede de Internet local.

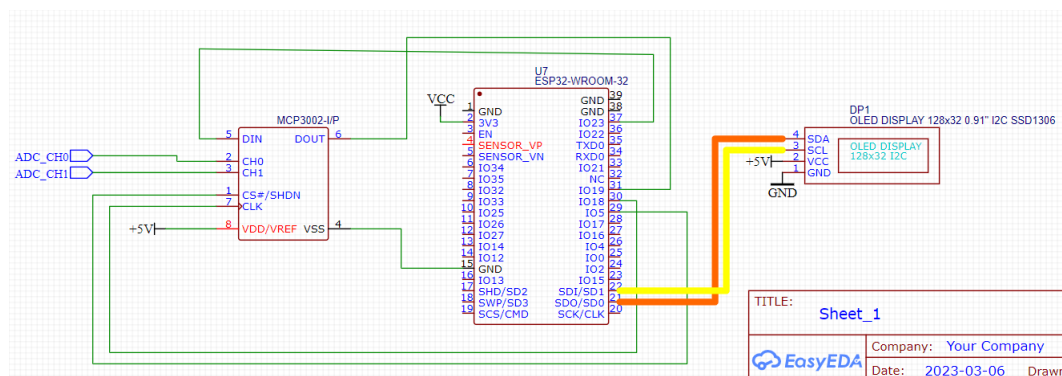


Figura 5.9: Representação a traço grosso do circuito elétrico do display.

5.1.5 Implementação da PCB

A PCB foi desenvolvida tendo por base as considerações relativamente à minimização de ruídos referidas no capítulo 4.10. As figuras 5.10 e 5.11 apresentam a vista de cima e a vista de baixo respetivamente. A placa foi desenhada para se utilizar ambos os seus planos, permitindo desta forma agrupar os componentes. Como se pode verificar na figura 5.10 o circuito analógico está separado do circuito digital, estando o circuito analógico implementado na metade esquerda da PCB e o circuito digital implementado na sua metade direita. Os condensadores também foram desenhados para serem implementados o mais perto possível dos pinos do componente ACPL-785J. As linhas de alimentação e o plano de *ground* foi mais difícil de implementar pois são utilizados cinco níveis de alimentação diferentes. Foram utilizadas linhas de alimentação de 230VAC, 5VDC não isolados, 5VDC isolados e 3.3VDC isolados.

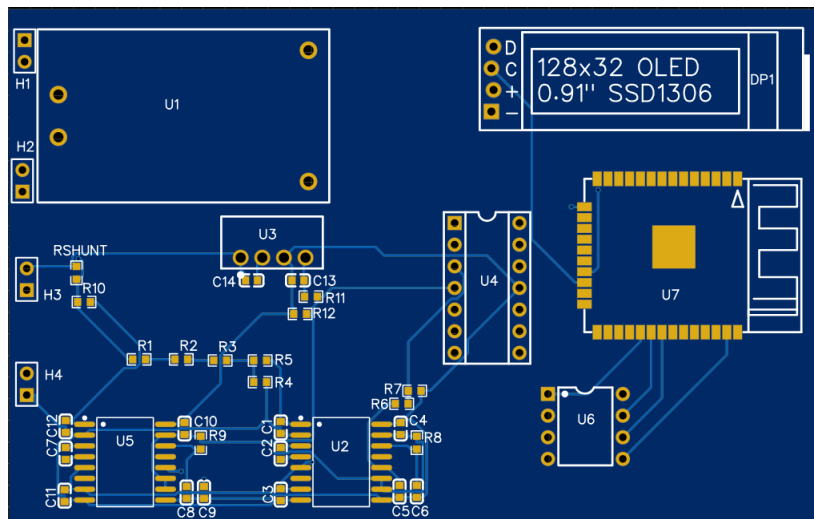


Figura 5.10: Vista de cima da placa de circuito impresso.

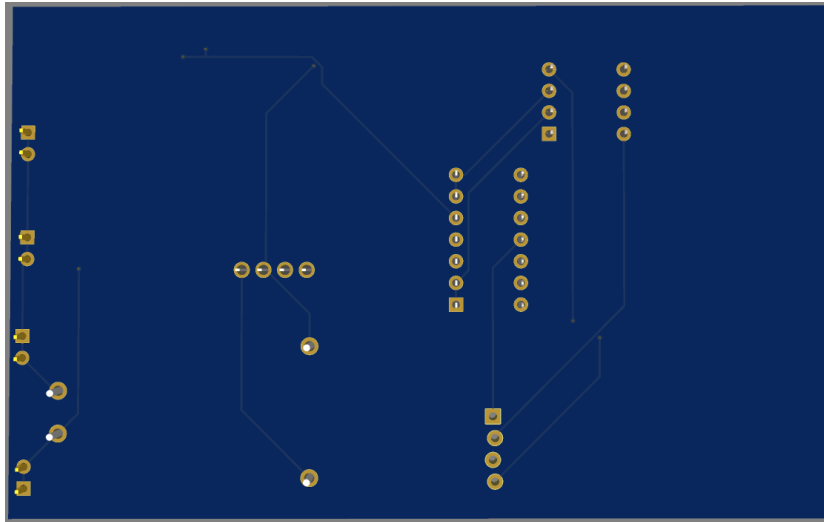


Figura 5.11: Vista de baixo da placa de circuito impresso.

5.2 Implementação do *Software*

Neste capítulo são abordados as implementações de *software* realizada no microcontrolador e no servidor. As implementações de *software* a abordar são: comunicação entre ADC e microcontrolador, processamento dos dados, comunicação entre microcontrolador e display, comunicação entre microcontrolador e servidor, e por fim a implementação realizada através da plataforma Node-RED.

Antes de abordar as implementações é importante referir que o código-fonte do programa foi desenvolvido em C++ seguindo as melhores práticas de programação. Esta linguagem permite um maior desempenho do sistema, maior flexibilidade, é abstrata, tem um grande suporte da comunidade e é orientada a objetos o que permite escrever o código de forma mais organizada e reutilizável. Sendo uma linguagem bastante útil na programação de sistema embebidos.

5.2.1 Implementação da comunicação com o ADC utilizando o protocolo de comunicação SPI

Este bloco tem como objetivo demonstrar a implementação do protocolo de comunicação SPI com o ADC. A implementação do protocolo realizou-se utilizando bibliotecas predefinidas. Para este caso foi necessário utilizar a biblioteca MCP3XXX.h e a biblioteca SPI.h. A biblioteca MCP3XXX permite realizar a comunicação com o ADC e a biblioteca SPI permite realizar a comunicação do protocolo SPI utilizando funções predefinidas.

O ADC e o protocolo de comunicação são parametrizados conforme representado na figura 5.12. São definidos os pinos físicos do microcontrolador, número de bits e

de canais a serem utilizados pelo ADC e a definição da taxa de comunicação utilizada pelo protocolo SPI.

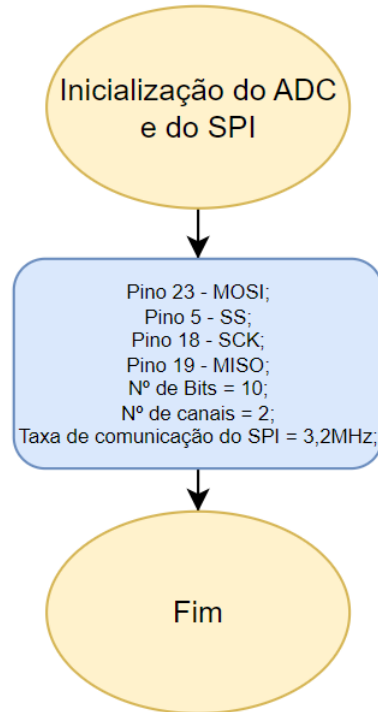


Figura 5.12: Parâmetros de inicialização do ADC e protocolo de comunicação SPI.

A utilização da biblioteca MCP3XXX.h tem como vantagem a possibilidade de trabalhar com as diferentes versões do ADC MCP, sendo que para este projeto o dispositivo utilizado foi o MCP3002. A classe MCP3XXX integra um *template* onde se inclui o número de bits, número de canais, velocidade máxima de comunicação do protocolo SPI em hertz e o número de bytes transferido. Esta classe é definida através do nome MCP3002 e preenche as variáveis com os dados obtidos através da ficha técnica do MCP3002. O número de bits, número de canais, velocidade máxima de comunicação do protocolo SPI e o número de bytes transferidos são, respetivamente, definidos de acordo com os seguintes valores: 10, 2, 3200000, 2 (linha 1). A definição do MCP3002 é exposta na listagem 5.1.

```

1 typedef MCP3XXX_<10, 2, 3200000, 2> MCP3002;
2
3 template<uint8_t NumBits,
4         uint8_t NumChannels,
5         uint32_t MaxSPIClockSpeed,
6         uint8_t SPITransferLength = 3>
  
```

Listagem 5.1: Instanciação do MCP3002

A comunicação entre o microcontrolador e o MCP3002 é realizada enviando grupos de 8 bits, sendo neste caso suficiente enviar somente 2 bytes. A transmissão é realizada quando colocamos o pino CS para baixo e enviamos o *start bit* a 1. De notar que as portas SPI do microcontrolador necessitam de ser configuradas de modo que o envio de dados ocorra no flanco descendente do pulso de relógio e a recepção de dados ocorra no flanco ascendente do pulso de relógio. O exemplo da comunicação entre o microcontrolador e o MCP é apresentado na figura 5.13.

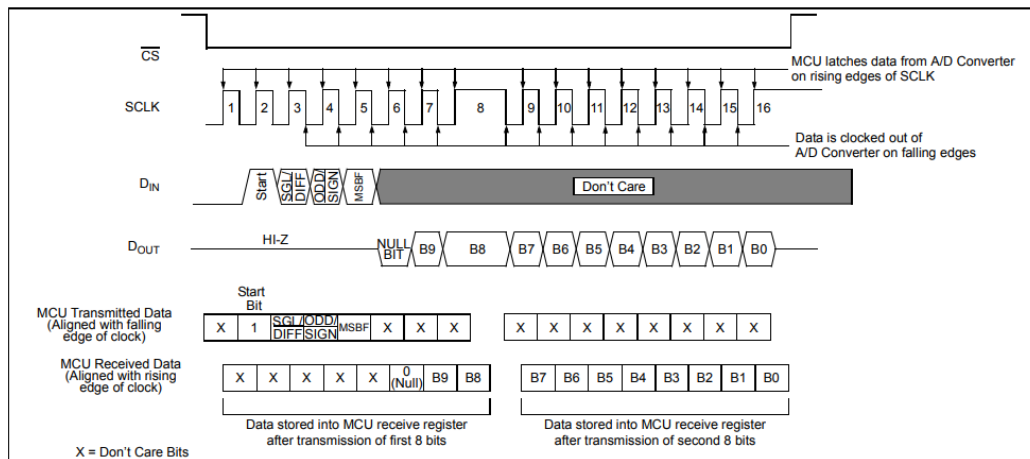


Figura 5.13: Comunicação SPI com MCP3002 utilizando segmentos de 8 bits, no modo *Single-Ended*[61]

A comunicação SPI entre o microcontrolador e o MCP3002 foi implementado como exposto na listagem 5.2. A função *analogRead* (linha 1) é utilizada para ler o canal 0 ou canal 1. Esta função verifica se o número de canais utilizado pelo MCP3002 está correto e chama a função *_read* (linha 8) que tem como objetivo iniciar a comunicação com o ADC. A função *_read* verifica se o número de canais é igual a 2 (linha 15) e de seguida verifica se o número de bits é igual a 10 (linha 17), o que é o caso do MCP3002. O primeiro byte é enviado definindo o *start bit*, *differential bit*, *channel bit* e por fim o *Most Significant Bit First* (MSBF) (linha 20, 22, 24 e 26). O bit de início (*start bit*) é definido como 0b01000000. Esse bit é o primeiro bit enviado para iniciar a comunicação. A definição do bit *differential* é baseado no parâmetro *differential*, definido com o valor zero. De acordo com a figura 4.13 presente no capítulo 4.7.1 este parâmetro tem de ser colocado a um, sendo o bit 5 colocado a um, 0b00100000. O bit de canal é definido com base no valor do parâmetro *channel*. Se o canal for 0, o bit será definido como 0, caso contrário, será definido como 0b00010000. O bit MSBF é definido a 1. Esse bit é usado para especificar a ordem de transmissão dos bits. O segundo byte não é importante na inicialização da comunicação, mas é importante na recepção dos dados amostrados pelo ADC.

```
1  uint32_t analogRead(uint8_t channel) const
2  {
3      if (channel < NUM_CHANNELS)
4          return _read(channel, false);
5      return ADC_ERROR_INVALID_CHANNEL;
6  }
7
8  uint32_t _read(uint8_t channel, bool differential) const
9  {
10     // Data transfers are done using "8-bit segments" approach
11     // in data sheet.
12     // The sent data alignment results in correctly aligned
13     // return bytes after the SPI transfer.
14     uint8_t data[SPI_TRANSFER_LENGTH];
15
16     // Check for MCP3002
17     if (NUM_CHANNELS == 2)
18     {
19         if (NUM_BITS == 10)
20         {
21             // Start bit.
22             data[0] = 0b01000000;
23             // Differential bit.
24             data[0] |= (differential ? 0b00000000 : 0b00100000);
25             // Channel bit.
26             data[0] |= (channel == 0 ? 0b00000000 : 0b00010000);
27             // MSBF bit is set to 1. See section 5.1 of the data
28             // sheet.
29             data[0] |= 0b00001000;
30             // It doesn't matter what data[1] is set to.
31         }
32         else
33         {
34             return ADC_UNSUPPORTED_CONFIGURATION;
35         }
36     }
37     else
38     {
39         {...}
40         if (_useHardwareSPI)
41         {
42             {...}
43             else
44             {
45                 digitalWrite(_csPin, LOW);
46                 for (size_t i = 0; i < SPI_TRANSFER_LENGTH; ++i)
47                 {
48                     for (size_t j = 8; j-- > 0;)
49                     {
50                         // Set MOSI data.
51                         digitalWrite(_mosiPin, bitRead(data[i], j));
```

```

47         // Set Clock HIGH.
48         digitalWrite(_sckPin, HIGH);
49         // Read MISO data.
50         bitWrite(data[i], j, digitalRead(_misoPin));
51         // Set Clock LOW.
52         digitalWrite(_sckPin, LOW);
53     }
54 }
55 digitalWrite(_csPin, HIGH);
56 }
57
58 // Here we take the second two bytes returned as our value.
59 // This value is already correctly aligned since we are
60 // using the 8-bit segments approach.
61 // The BIT_MASK is calculated based on the number out bits
62 // specified in the template parameters.
63 return ((data[SPI_TRANSFER_LENGTH - 2] << 8) | data[
64 SPI_TRANSFER_LENGTH - 1]) & BIT_MASK;
65 }

```

Listagem 5.2: Configuração e comunicação com o MCP3002

A listagem 5.3 apresenta a instanciação da variável *adc* na biblioteca MCP3002 (linha 3), a inicialização do ADC (linha 7) e do *Timer0* (linha 8) e também apresenta os vetores (linha 13, 14) que guardará as amostras do sinal de tensão e de corrente capturadas pela função *adc.analogRead(x)*. Sendo que o *x* representa qual o canal a ser lido. De notar que *Timer0* é inicializado com uma frequência de chamada de 10kHz.

```

1 #include <MCP3XXX.h>
2 // Create an instance of the MCP3002 ADC.
3 MCP3002 adc;
4 void setup()
5 {
6     Serial.begin(115200);
7     adc.begin();           //inicializacao do ADC
8     timer_init();        //inicializacao do timer
9     ...
10 }
11 bool IRAM_ATTR TimerHandler0(void * timerNo)
12 {
13     vetor_de_tensao[nr_amostra] = adc.analogRead(0);
14     // ler canal 0 para MCP3002 ADC
15     vetor_de_corrente[nr_amostra] = adc.analogRead(1);
16     // ler canal 1 para MCP3002 ADC
17     amostras_totais_tensao += vetor_de_tensao[nr_amostra];
18     // Soma das amostras

```

```
16 amostras_totais_corrente += vetor_de_corrente[nr_amostra];  
    // Soma das amostras  
17 nr_amostra++;  
    // Numero de amostras  
18 return true;  
19 }
```

Listagem 5.3: Instanciação e leitura dos canais do ADC

5.2.2 Processamento dos dados capturados

Este capítulo tem como objetivo apresentar como é realizado o tratamento das amostras de energia capturadas pelo ADC.

Na figura 5.14 é apresentado o fluxograma referente à interrupção interna do *Timer0*. Esta interrupção é chamada à frequência de 10KHz para que possa guardar as amostras capturadas do ADC num vetor com capacidade para guardar 200 amostras. Como é do nosso interesse calcular o valor eficaz de tensão e o valor eficaz de corrente, é necessário criar dois vetores, um vetor guardará a amostra referente ao valor de tensão e o outro vetor guardará a amostra referente ao valor de corrente.

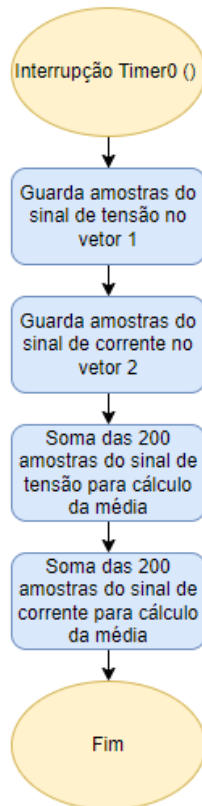


Figura 5.14: Fluxograma da interrupção interna que ocorre numa frequência de 10KHz

Para explicar como os dados são tratados é apresentada a figura 5.15. Esta figura apresenta o fluxograma correspondente à implementação de *software* cujo objetivo é calcular o valor eficaz de tensão, valor eficaz de corrente, potência aparente, potência real e fator de potência. O programa está sempre a correr a função *Main()*, verificando quando é que o vetor guarda as 200 amostras dos sinais analógicos. Como referido anteriormente, estas amostras são guardadas à frequência de 10kHz num vetor com capacidade para 200 posições, ou seja a cada 50Hz temos um ciclo do sinal de tensão e de corrente completos, bem como as 200 amostras de cada sinal. Tendo o vetor do sinal de tensão e do sinal de corrente preenchido com 200 amostras paramos a interrupção do *Timer0* e chamamos a função *Calc()*. A função *Calc()* calcula a média das 200 amostras do sinal de tensão e do sinal de corrente e calcula o valor instantâneo da tensão, corrente e potência.

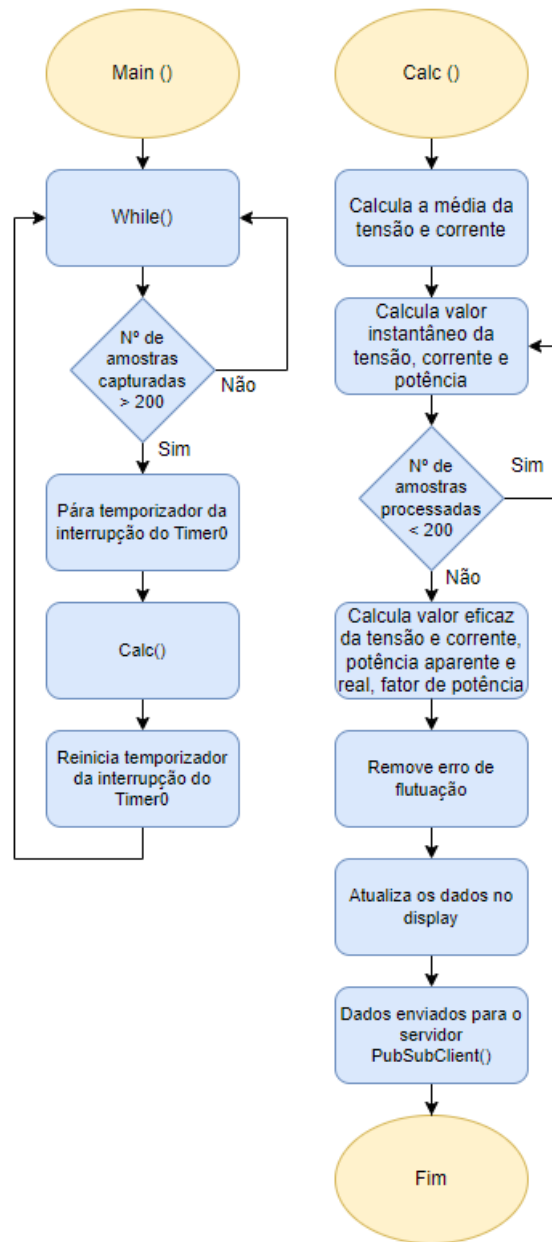


Figura 5.15: Fluxograma do tratamento das amostras de dados

O cálculo é realizado com base nas equações que serão mostradas no decorrer da explicação sobre o tratamento das amostras. As equações estão expressas no domínio do tempo discreto e representam a potência real, corrente eficaz, tensão eficaz, potência aparente e fator de potência. No fim do capítulo são apresentadas na tabela 5.1 as legendas das siglas usadas para apresentar as equações. A equação 5.1 é utilizada para calcular a tensão eficaz, onde cada amostra de tensão é elevada ao quadrado e multiplicada pelo somatório das amostras, este valor é dividido pelo número de amostras totais, por fim calcula-se a raiz quadrada deste valor e obtém-se a tensão eficaz. A corrente eficaz, equação 5.2 é obtida da mesma forma que a tensão

eficaz.

$$U_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} u^2(n)}{N}} \quad (5.1)$$

$$I_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} i^2(n)}{N}} \quad (5.2)$$

De notar que o valor digital obtido pelo microcontrolador tem uma componente média que é necessário remover antes de realizarmos qualquer calculo referente à tensão e corrente eficaz. Além da remoção da componente média também é necessário multiplicar o valor de tensão e corrente eficaz pela constante de ajuste. Esta constante permite apresentar o valor de tensão, corrente e potência de acordo com o valor real. O valor teórico da constante de ajuste é obtido de acordo com as seguintes assunções:

- O valor de tensão máximo na rede elétrica varia entre -325VAC a 325VAC.
- O ADC tem de resolução 10 bits e permite capturar amostras na faixa de 0V a 5V
- O microcontrolador não permite aos seus pinos tensões superiores a 3.3V.

Assim sendo, podemos garantir que à entrada do microcontrolador não teremos um valor superior a 3.3V. E sabendo que 5V é representado pelo valor digital 1024, então os 3.3V representam o valor digital de 675,84. Para realizar corretamente os cálculos é necessário retirar a componente média do sinal amostrado, sendo o valor teórico da componente média metade do valor máximo o que coloca o nosso sinal amostrado a variar entre -337.92 a 337.92. Desta forma concluímos que o valor de 337.92 representa o valor de 325VAC, concluindo assim que a constante de ajuste tem o valor teórico de 0.96. De realçar que o valor teórico da constante de ajuste pode divergir um pouco do valor real da constante de ajuste, que foi definido com o valor de 1.08.

O valor de potência aparente é obtido ao multiplicar os valores de tensão e corrente eficazes, como apresentado na equação 5.3. A potência real, apresentada na equação 5.4, é obtida através do somatório da multiplicação entre a amostra do valor de tensão e a amostra do valor de corrente, dividindo-se o valor resultante pelo número de amostras totais. O fator de potência, apresentado na equação 5.5 é conseguido através da divisão entre a potência real da potência aparente.

$$PA = U_{rms} * I_{rms} \quad (5.3)$$

$$PR = \frac{1}{N} \sum_{n=0}^{N-1} u(n) * i(n) \quad (5.4)$$

$$FP = \frac{PR}{PA} \quad (5.5)$$

A listagem 5.4 representa parte da implementação da função responsável pelo cálculo da energia.

```

1 void calculate_energy()
2 {
3 ...
4
5     tensao_media = amostras_totais_tensao/nr_amostra;
6     corrente_media = amostras_totais_corrente/nr_amostra;
7
8     for (int amostra_temp = 0; amostra_temp < nr_max_amostras;
9         amostra_temp++) {
10        tensao_sem_componente_media += (vetor_de_tensao[amostra_temp]-
11            tensao_media);
12        corrente_sem_componente_media += (vetor_de_corrente[
13            amostra_temp]-corrente_media);
14
15        amostra_potencia_inst = (vetor_de_tensao[amostra_temp]-
16            tensao_media) * (vetor_de_corrente[amostra_temp]-corrente_media
17            ); //potencia instantanea
18        somatorio_potencia_inst_aux += amostra_potencia_inst;
19        amostra_tensao_sem_componente_media = (vetor_de_tensao[
20            amostra_temp] - tensao_media);
21        //quadrado da tensao menos o valor medio
22        somatorio_tensao_quadrada +=
23            amostra_tensao_sem_componente_media *
24            amostra_tensao_sem_componente_media; //
25        Somatorio do quadrado da tensao
26        amostra_corrente_sem_componente_media = (vetor_de_corrente[
27            amostra_temp] - corrente_media);
28        somatorio_corrente_quadrada +=
29            amostra_corrente_sem_componente_media *
30            amostra_corrente_sem_componente_media;
31    }
32    if (somatorio_potencia_inst_aux < 0){
33        somatorio_potencia_inst = 0;
34    } else somatorio_potencia_inst = somatorio_potencia_inst_aux;
35
36    tensao_quadrada = somatorio_tensao_quadrada / nr_amostra;
37    corrente_quadrada = somatorio_corrente_quadrada / nr_amostra;
38
39    tensao_eficaz = sqrt(tensao_quadrada) * kAjusteTensao;
40        //tensao eficaz

```

```

28  corrente_eficaz = sqrt(corrente_quadrada) * kAjusteCorrente;
                                //corrente eficaz
29  potencia_aparente = tensao_eficaz * corrente_eficaz;
                                //potencia aparente
30  potencia_real = kAjusteCorrente * kAjusteTensao *
    somatorio_potencia_inst/(nr_amostra); //potencia real
    - W
31  fator_potencia = (potencia_real/potencia_aparente) * 100;
                                //Fator de potencia em
    percentagem
32
33  if (instante_atual - ultimo_instante >= intervalo) {
                                //soma energia a cada
    segundo
34  double potencia_real_convertida = static_cast<double>(
    potencia_real);
35  energia = potencia_real_convertida / 3600;
36  energia_wh += energia;
37  energia_kwh += energia / 1000;
38  ultimo_instante = instante_atual;
39
40  static unsigned long counter = 0;
41  counter++;
42  if (counter >= duracao) {
43      energia_diaria = energia /1000;
44      energia = 0.0;
45      counter = 0;
46  }
47  }
48  ...
49  }

```

Listagem 5.4: Cálculo da tensão e corrente eficaz potência aparente e real e fator de potência

Tabela 5.1: Legenda das siglas das equações

Legenda	
PR	Potência real
PA	Potência aparente
FP	Fator de potência
$u(n)$	instância amostrada do sinal analógico de tensão
$i(n)$	instância amostrada do sinal analógico de corrente
N	número de amostras
U_{rms}	Tensão eficaz
I_{rms}	Corrente eficaz

5.2.3 Implementação da comunicação com o display utilizando o protocolo de comunicação I2C

A implementação de *software* deste protocolo foi suportada por três bibliotecas. A biblioteca <Adafruit_SSD1306.h> e <Adafruit_GFX.h>, que permitem realizar a comunicação com o módulo SSD1306 e utilizar funções predefinidas de escrita. A biblioteca <Wire.h>, que permite realizar a transferência de dados do microcontrolador para o display utilizando o protocolo I2C.

A inicialização do display e do protocolo I2C está representado na 5.16. Os parâmetros representados dizem respeito à ligação elétrica dos pinos SDA e SCL, às características físicas do display, velocidade de transferência permitida pelo protocolo de comunicação I2C, inicialização da tensão de alimentação interna do display e também o endereço lógico do display para que o microcontrolador possa transmitir os dados de energia.

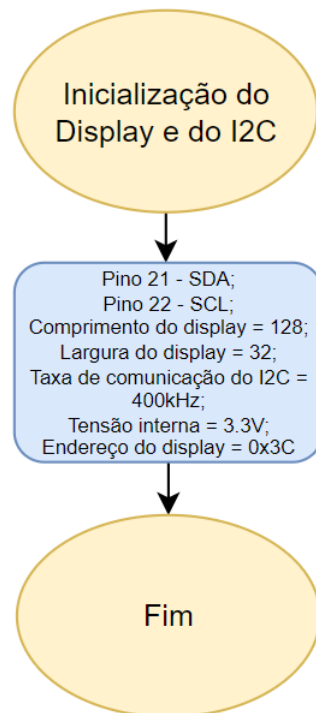


Figura 5.16: Parâmetros de inicialização do display e protocolo de comunicação I2C

Na listagem 5.5 é representada a construção do objeto display (linha 1) recebendo como argumentos o comprimento do display, altura do display, biblioteca *wire* e por fim o valor lógico de *reset*, representado pelo -1. Os argumentos relacionados com o comprimento e largura dizem respeito ao número de pixels totais do display, sendo que o display tem 128 pixels de largura e 32 pixels de altura, o -1 é utilizado como argumento *reset* devido à inexistência do pino *reset* na placa do *display*.

```

1 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
    OLED_RESET);
2 MCP3002 adc;
3 ESP32Timer ITimer0(0);
4 WiFiClient espClient;
5 PubSubClient client(espClient);

```

Listagem 5.5: Construção do objeto display utilizando a classe Adafruit_SSD1306

A construção do objeto display é representado na listagem 5.6 e tem como argumentos (linha 1) o comprimento e a largura do display, a biblioteca Wire, o *reset*, a velocidade de transmissão durante a transmissão de dados e a velocidade de transmissão depois de realizada a transmissão de dados. Durante a transmissão de dados a velocidade é colocada a 400kHz e depois da transmissão de dados a velocidade de transmissão é colocada a 100kHz, em concordância com os valores definidos na ficha técnica.

```

1 Adafruit_SSD1306::Adafruit_SSD1306(uint8_t w, uint8_t h, TwoWire *
    twi, int8_t rst_pin, uint32_t clkDuring, uint32_t clkAfter)
2     : Adafruit_GFX(w, h), spi(NULL), wire(twi ? twi : &Wire),
    buffer(NULL), mosiPin(-1), clkPin(-1), dcPin(-1), csPin(-1),
    rstPin(rst_pin)
3 #if ARDUINO >= 157
4     ,
5     wireClk(clkDuring), restoreClk(clkAfter)
6 #endif
7 {
8 }

```

Listagem 5.6: Construtor do objeto display na biblioteca Adafruit_SSD1306

Na listagem 5.7 é apresentada a função de inicialização do objeto display. Esta função tem como argumentos o valor 0x02, que indica a tensão de alimentação interna do display de 3.3V, e como segundo argumento é passado o valor 0x3C, indicando que o display a utilizar é de 128x32 pixels.

```

1 void display_init()
2 {
3     // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V
    internally
4     if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
5         Serial.println(F("SSD1306 allocation failed"));
6         for(;;); // Don't proceed, loop forever

```

```

7   }
8   ...
9   }

```

Listagem 5.7: Inicialização do objeto display na biblioteca Adafruit_SSD1306

Os dados de energia são apresentados no display depois de chamar a função *display_oled_results*. Esta função tem como objetivo mostrar os dados de energia no display. Conforme se pode verificar no excerto da função *display_oled_results* representada na listagem 5.8 os dados são apresentados chamando as funções já predefinidas das bibliotecas Adafruit_SSD1306 e Adafruit_GFX.

```

1 void display_oled_results(float inp1, float inp2, float inp3,
2   double inp4, double inp5)
3 {
4   display.clearDisplay();
5   display.setTextSize(1);           // Normal 1:1 pixel scale
6   display.setTextColor(SSD1306_WHITE); // Draw white text
7
8   display.setCursor(0,0);           // Start at top-left corner
9   display.print("V: ");             //Tensao
10  display.print(inp1);               //Valor de Tensao
11
12  display.setCursor(60, 0);
13  display.print("A: ");
14  display.print(inp2);               //Valor de corrente
15
16  display.setCursor(0, 8);
17  display.print("W: ");
18  display.print(inp3);               //Valor de Potencia
19
20  display.setCursor(0, 16);
21  display.print("whInst: ");
22  display.print(inp4);               //Valor de energia
23  display.print("instantaneo");
24
25  display.setCursor(0, 24);
26  display.print("whAcm: ");
27  display.print(inp5);               //Valor de energia acumulado
28
29  display.display();
30  delay(200);
31 }

```

Listagem 5.8: Função para apresentar dados no display utilizando a biblioteca Adafruit_SSD1306 e Adafruit_GFX

5.2.4 Implementação da comunicação com a rede de internet utilizando o módulo de comunicação WIFI

A implementação da comunicação WIFI foi baseada na API do site da Espressif Systems. A Espressif Systems disponibiliza a biblioteca wifi e detalha as funções necessárias para utilizar o módulo WIFI. No site são apresentadas as APIs que permitem configurar, inicializar e conectar à rede de internet local [69]. Exemplos das API utilizadas na implementação do bloco de comunicação WIFI são descritas e identificadas de seguida.

A função listada 5.9 tem como objetivo conectar o microcontrolador à rede de internet, utilizando para isso o nome da rede, designado através do *Service Set Identifier* (SSID), e a palavra passe da mesma.

```
1 WiFi.begin(ssid, password);
```

Listagem 5.9: Configuração do módulo wifi

A função listada 5.10 permite perceber se a conexão à rede de internet foi bem sucedida.

```
1 while (WiFi.status() != WL_CONNECTED) {  
2     delay(500);  
3     Serial.print(".");  
4 };
```

Listagem 5.10: Verificação da conexão à rede de Internet

Depois da comunicação WIFI estar definida e inicializada, avançamos com a implementação do protocolo de comunicação MQTT. Este protocolo vai permitir transmitir as mensagens entre microcontrolador e servidor.

A figura 5.17 ilustra a implementação da comunicação entre ambos os microcontroladores e o servidor. O microcontrolador responsável pela monitorização de dados de energia publica oito tópicos, sendo eles referentes à tensão eficaz, corrente eficaz, potência real, potência aparente, fator de potência, energia instantânea, energia em Wh e energia Wh/dia. O microcontrolador responsável pelo estado do LED publica no tópico MQTT_Publish_estadoLED o estado do LED e subscreve o tópico MQTT_Subscribe_controloLED do comando do LED.

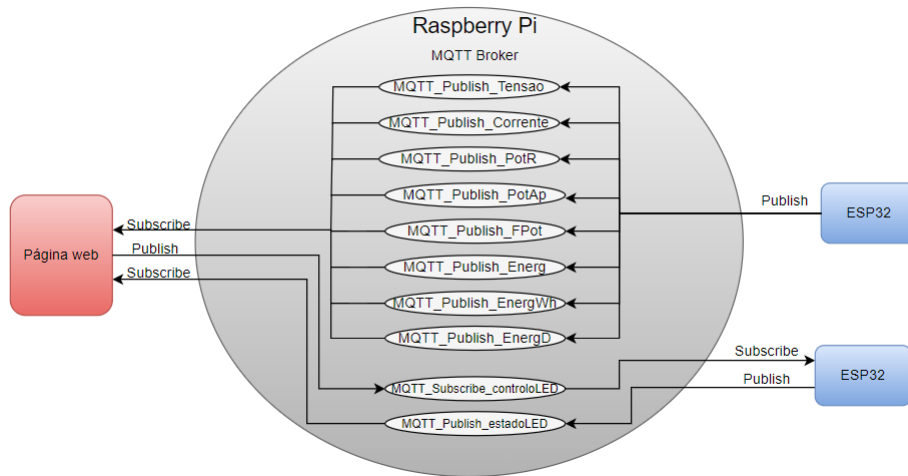


Figura 5.17: Comunicação *publish/subscriber* entre microcontrolador e página *web*.

Para implementar a comunicação através do protocolo MQTT é necessário conhecer o endereço do dispositivo com o qual pretendemos comunicar e especificar os tópicos onde os dados serão transmitidos. De salientar, que o cliente (microcontrolador) e o *broker* (Raspberry Pi) necessitam de estar ligados à mesma rede local de forma a garantir a transmissão de dados.

A função listada 5.11 permite definir o endereço do servidor com o qual o microcontrolador irá realizar a transferência de dados. O parâmetro `mqtt_server` diz respeito ao endereço *Internet Protocol* (IP) do servidor e o parâmetro `mqtt_port` diz respeito à porta utilizada para realizar a comunicação de dados. O número da porta é definido consoante o tipo de transporte de dados escolhido, neste caso é utilizado o MQTT sobre TCP, o que significa que o número da porta é o 1883 [74]. De notar que a utilização desta função necessita da biblioteca `pubSubClient`.

```
1 client.setServer(mqtt_server, mqtt_port);
```

Listagem 5.11: Define o endereço do servidor com o qual o microcontrolador irá realizar a transferência de dados

Na listagem 5.12 é apresentada a função que transmite a mensagem de dados para o respetivo tópico. A transmissão de dados ocorre de um em um segundo para não sobrecarregar o *buffer* do servidor.

```
1 //envia mensagens mqtt de 1 em 1 segundo
2 if (instante_atual - ultimo_instante_mqtt >= intervalo) {
3     ultimo_instante_mqtt = instante_atual;
4     display_web_results(tensao_eficaz, MQTT_Publish_Tensao);
5     display_web_results(corrente_eficaz, MQTT_Publish_Corrente);
```

```
6   display_web_results(potencia_real, MQTT_Publish_PotenciaReal);
7   display_web_results(potencia_aparente,
8   MQTT_Publish_PotenciaAparente);
9   display_web_results(fator_potencia, MQTT_Publish_FatorPotencia
10  );
11  display_web_results(energia, MQTT_Publish_Energia);
12  display_web_results(energia_wh, MQTT_Publish_EnergiaWh);
13  display_web_results(energia_diaria, MQTT_Publish_EnergiaDiaria
14  );
15  }
```

Listagem 5.12: Transmissão da mensagem do ESP32 para o servidor

5.2.5 Implementação da página web utilizando o Node-RED

Em relação ao Node-RED a implementação foi realizada de acordo com o fluxograma na figura 5.18. Adicionou-se um relógio de forma a guardar os dados de energia num ficheiro de texto de um em um segundo, e também para ter um registo da data e hora da apresentação dos dados. A inicialização da comunicação entre dispositivos é realizada com um nó designado de *mqtt in*. A apresentação da página com os dados de energia e página do controlo do LED são configuradas nos campos destinados à implementação de grupos e de abas dentro dos grupos.

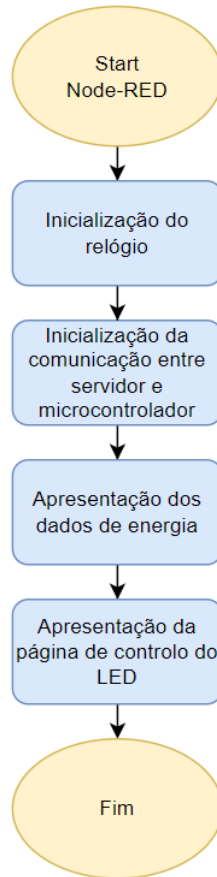


Figura 5.18: Fluxograma da implementação da página gráfica.

O nó *mqtt in* diz respeito à ligação e transmissão de dados do microcontrolador e ao servidor. Na figura 5.19 visualiza-se os campos necessários para realizar a configuração do nó *mqtt in*, designado por *esp32/Tensao*. No campo *server* defini-se o endereço IP do servidor, a porta e o protocolo de comunicação a utilizar. Insere-se o tópico com o qual pretendemos receber as mensagens, define-se o nível de QoS e o tipo de saída de dados pretendida.

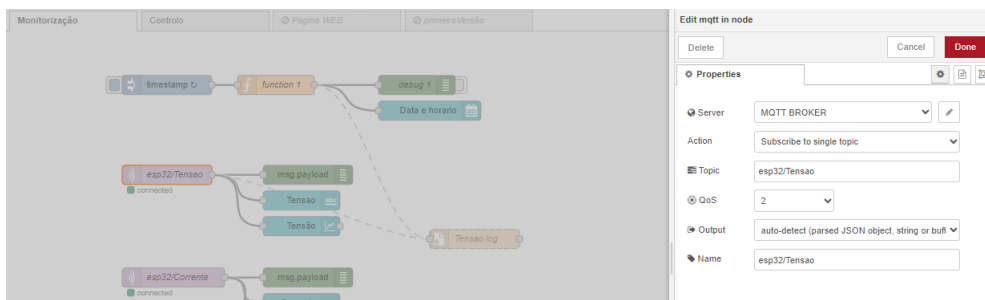


Figura 5.19: Configuração do nó mqtt in.

A apresentação dos dados de energia é realizada com o nó *text* e o nó *chart*. As

configurações destes nós são realizadas como apresentado na figura 5.20. Esta figura apresenta a configuração de um nó do tipo *chart*, sendo necessário definir o grupo e a aba onde é apresentado o gráfico, definir o tipo de gráfico a apresentar e configurar o eixo X e eixo Y.

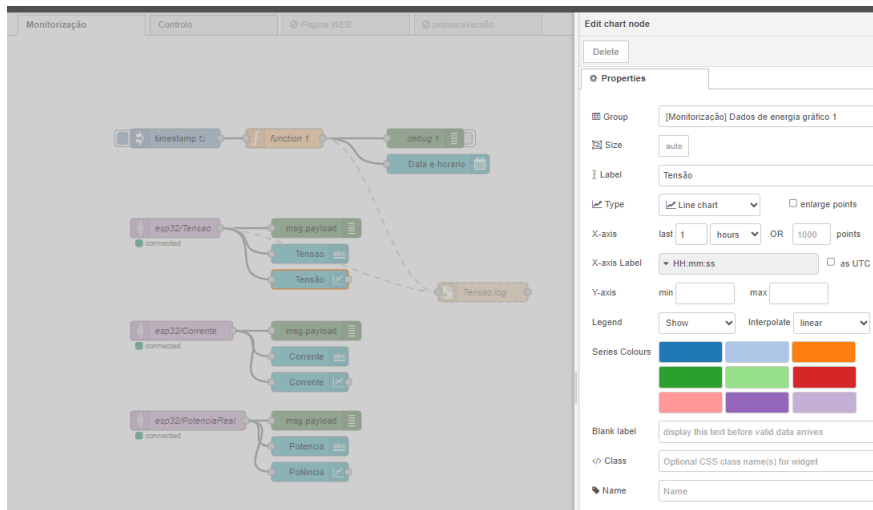


Figura 5.20: Configuração do nó chart.

Capítulo 6

Testes aos sistemas de Hardware e Software

No capítulo 6.1 são abordados os testes efetuados ao sistema. Os testes foram repartidos em duas partes, uma parte dedicada ao sistema de *hardware* e outra parte é dedicada ao sistema de *software*. Neste capítulo procura-se também realçar a relevância de um plano de testes realizado ao sistema com o exemplo prático.

6.1 Testes ao sistema de *Hardware e Software*

A realização de testes aos sistemas implementados garantem que os sistemas funcionem de acordo com as especificações iniciais.

Para realizar os testes é necessário elaborar um plano de testes. O plano de testes para este projeto foi elaborado tendo em conta os seguintes pontos:

- Definir o âmbito do teste e o sistema a ser testado;
- Definir o momento para realizar o teste;
- Identificar os objetivos dos testes e os critérios de aceitação;
- Selecionar as ferramentas e técnicas de teste adequadas.

O plano de testes, será utilizado para realizar os testes manuais ao sistema de *hardware* e ao sistema de *software*. No primeiro ponto do plano de testes, os blocos funcionais do sistema de *hardware* foram testados a nível individual e a nível de sistema. No que diz respeito ao sistema de *software* não foi possível testar todas as funções isoladamente, no entanto todas as funções foram testadas a nível de sistema.

Para o segundo ponto e relativamente ao sistema de *hardware*, foi definido como início de testes o momento após a montagem de cada bloco funcional do sistema de *hardware* na placa de ensaio. Para o sistema de *software* foi definido como momento inicial de testes o momento em que as diferentes fases de implementação mostravam maturidade suficiente, ou seja, depois de implementar a primeira versão da função.

No terceiro ponto é importante ter objetivos de testes claros e realistas. Os objetivos têm de ir ao encontro do correto funcionamento do sistema, não negligenciando os requisitos *hardware* e *software* previamente definidos nos capítulos anteriores.

Por fim, as ferramentas e as técnicas de teste são essenciais para realizar e validar os testes. Para o sistema de *hardware* foram utilizados como ferramentas de testes o multímetro e o osciloscópio. Para o sistema de *software* foi utilizada a consola *Integrated Development Environment* (IDE) para realizar os primeiros testes de *software*, sendo que numa fase posterior recorreu-se ao display e à página *web* para testar as funções de *software*.

Na figura 6.1 verifica-se um exemplo de testes efetuado ao bloco funcional e à integração dos blocos funcionais. No exemplo é realçado o teste ao bloco funcional relativo à fonte de alimentação e também é realçado um exemplo de teste executado à integração de dois blocos funcionais, captura do sinal analógico e processamento do dados capturados.

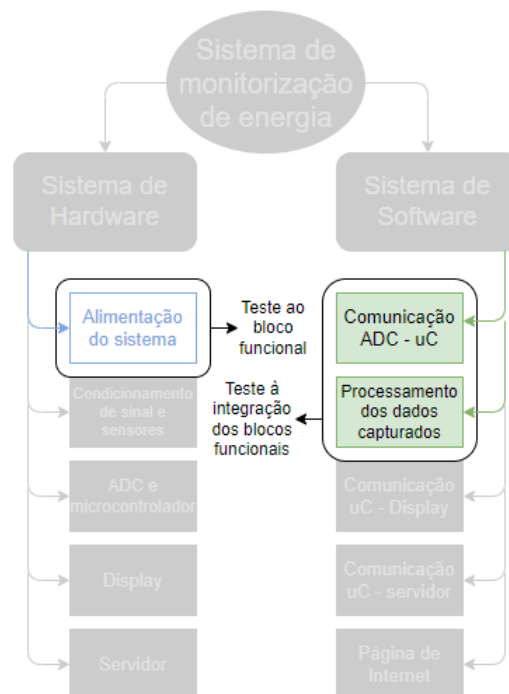


Figura 6.1: Exemplo de testes realizado ao bloco funcional isolado e integrado.

6.1.1 Testes ao Hardware

Nesta secção serão mostrados alguns exemplos dos testes realizados ao sistema de hardware.

Para o teste da figura 6.2 foi utilizado um multímetro de marca CIRCUITMATE DM25L. O multímetro apresenta o valor de 167.1 mV do sinal de tensão medido à entrada do sensor de tensão. Os 167,1 mV são o resultado da tensão eficaz da rede elétrica depois de trabalhados no divisor de tensão, sendo este divisor composto no ramo superior pelas 3 resistências em série de $47k\Omega$ cada uma e de 100Ω a do ramo inferior. Temos então à entrada do sensor uma onda sinusoidal que varia entre os ± 236 mV um pouco superior ao valor recomendado an ficha técnica do ACPL-785J. No entanto, o componente ACPL-785J só começa a cortar o sinal de saída quando o valor de entrada é superior ao valor de ± 256 mV. Para que o valor de tensão à entrada do ACPL-785J seja superior aos ± 256 mV o sinal de tensão da rede elétrica tem de ter o seu pico igual a 360VAC.



Figura 6.2: Sinal de tensão medido à entrada do sensor de tensão.

A figura 6.3 apresenta o teste realizado à resistência *shunt*, sendo possível verificar que o multímetro mostra uma queda de tensão aos terminais da resistência no valor de 98 mV quando temos uma carga de 8,9A ligada.

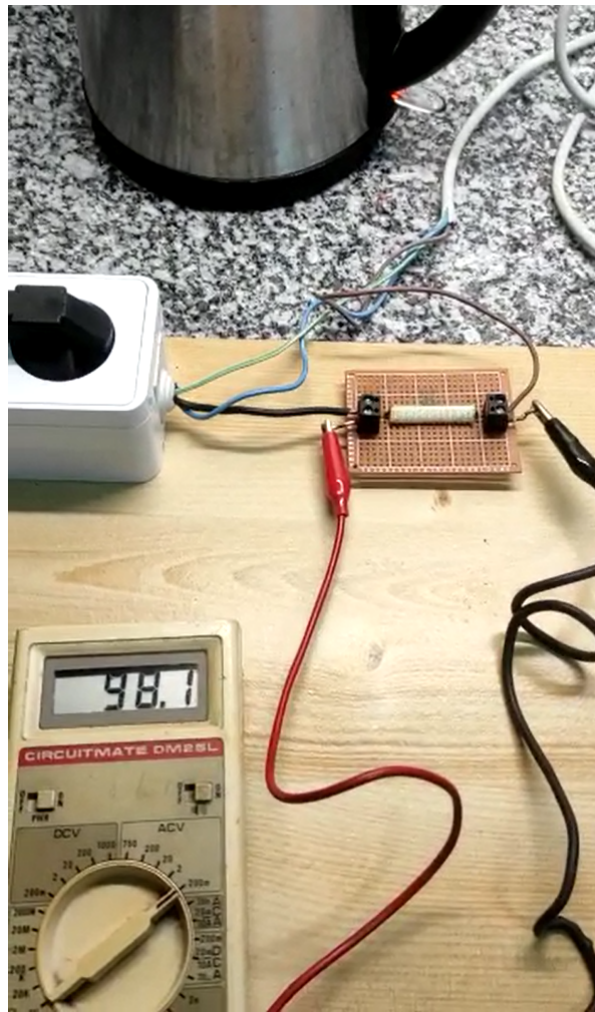


Figura 6.3: Sinal de tensão medido à entrada do sensor de corrente.

Para os seguintes testes foi utilizado o osciloscópio SDS 1202X-E da Siglent. Os seguintes testes tiveram como objetivo medir a onda sinusoidal processada pelo componente ACPL-785J de uma carga elétrica puramente resistiva.

A figura 6.4 mostra o resultado do sinal de tensão à saída do sensor de tensão.

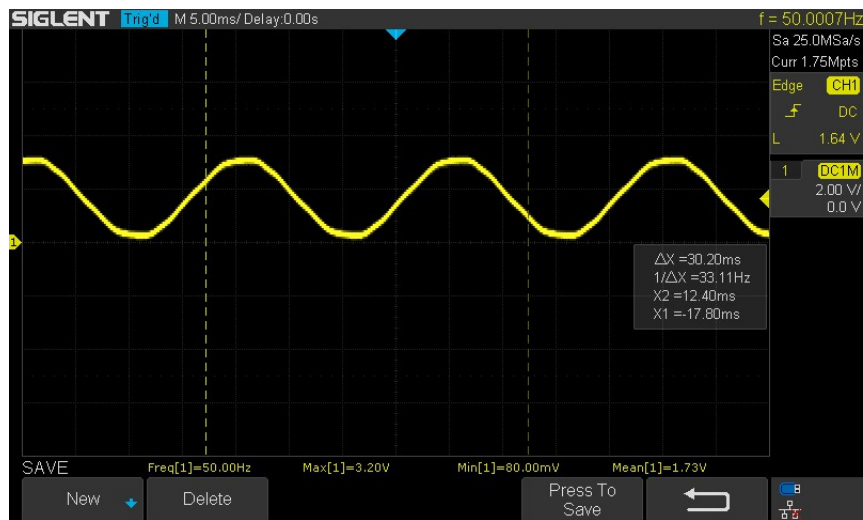


Figura 6.4: Sinal de tensão à saída do sensor de tensão.

Na figura 6.5 é possível visualizar o sinal à saída do sensor de corrente. Neste teste foi possível verificar a onda sinusoidal sem carga ativa.

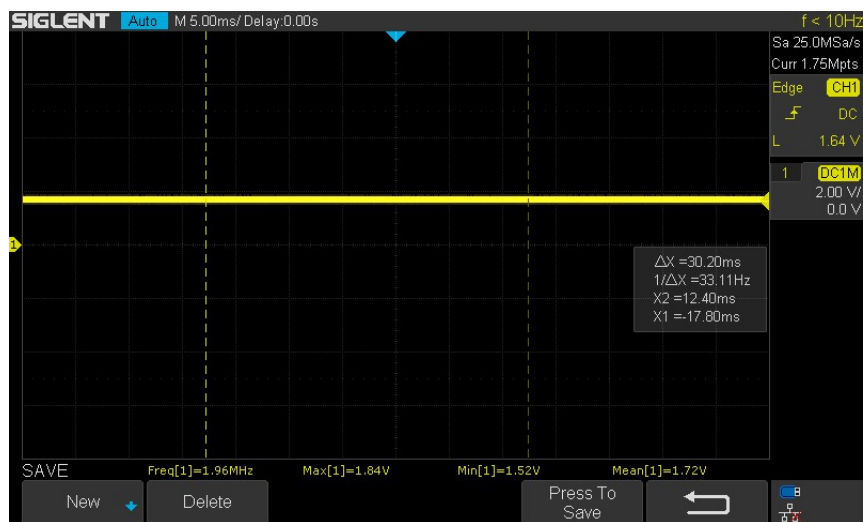


Figura 6.5: Sinal de tensão à saída do sensor de corrente sem carga ativa.

Na figura 6.6 é possível visualizar a onda sinusoidal à saída do sensor de corrente com a carga ativa. De notar que este sinal sinusoidal representa uma carga de 9A e 2000W de potência.

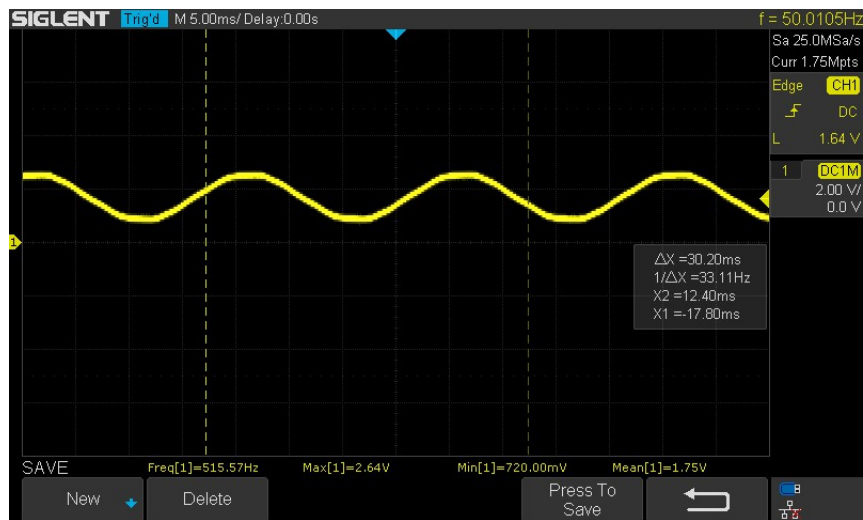


Figura 6.6: Sinal de tensão à saída do sensor de corrente com carga ativa.

Os seguintes testes tiverem como objetivo verificar e validar a frequência máxima de captura das amostras.

Na figura 6.7 é possível visualizar o tempo necessário para a função de interrupção. O teste foi realizado com o intuito de verificar o tempo de captura da amostra. Para isso foi dada a ordem para acender o LED no momento em que entramos na função de interrupção e no momento antes de sair da função de interrupção é dada a ordem para apagar o LED. Pode-se se ver que a captura da amostra necessita sensivelmente de 30 micro segundos. De acordo com os dados teóricos referenciados no capítulo 4.4.2 são necessários pelo menos $7.186\mu\text{s}$ para amostrar e converter o sinal, mais o tempo da transmissão de dados do ADC para o microcontrolador na ordem dos $3.125\mu\text{s}$, referenciado no capítulo 4.7.1, totalizando um tempo teórico mínimo de $10.311\mu\text{s}$. Temos então um valor prático três vezes superior ao valor teórico.

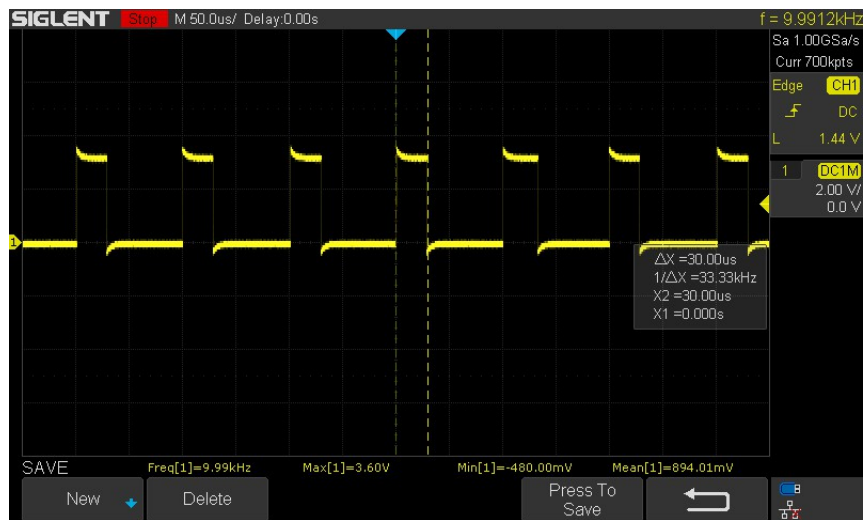


Figura 6.7: Sinal de tensão à saída com carga associada.

Na figura 6.8 é possível visualizar a frequência da função de interrupção. Pode-se ver que a interrupção é chamada a cada 10kHz. Utilizou-se o mesmo método que o teste anterior, acender e apagar o LED dentro da função de interrupção.

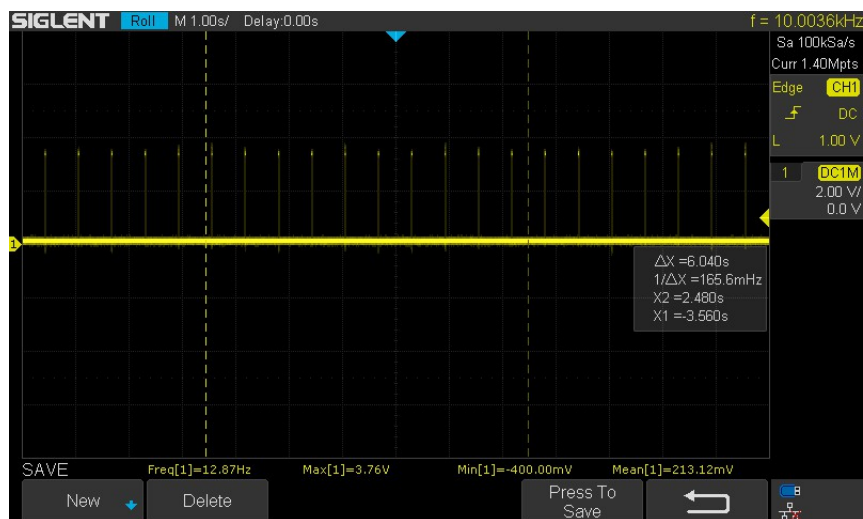


Figura 6.8: Verificação da frequência de interrupção.

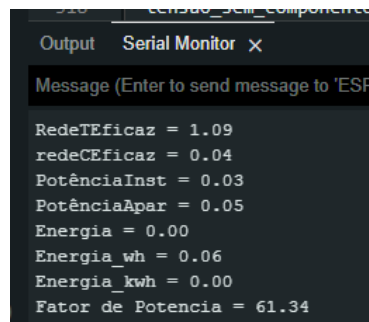
6.1.2 Testes ao *Software*

Os testes ao sistema de *software* foram desenvolvidos ao longo da implementação e foram divididos em duas fases: Na primeira fase testa-se o funcionamento da função isolada, e na segunda fase integra-se a função no sistema e volta-se a testar. O primeiro bloco funcional do sistema de *software*, Captura do sinal analógico, foi testado no teste 1, teste 2, teste 3 e teste 4. No teste 5 testou-se o bloco funcional Processamento dos dados capturados. O teste 6 permitiu testar o bloco funcional Apresentação de dados de energia. Para testar o bloco funcional Transmissão de

dados foram utilizados o teste número 7 e 8. No teste 9 testou-se o bloco funcional Registo de dados de energia.

- Teste 1: LED acende/apaga com a frequência de 1Hz através bloqueio de código;
- Teste 2: LED acende/apaga com a frequência de 1Hz através de interrupções temporizadas;
- Teste 3: Captura de dados através do ADC externo;
- Teste 4: Captura de dados através do ADC externo com a frequência de 10kHz através de interrupções temporizadas;
- Teste 5: Teste à função que permite calcular os dados capturados e convertê-los em tensão, corrente e potência
- Teste 6: Teste à exibição num *display OLED* dos dados processados;
- Teste 7: Comunicação através do serviço MQTT entre diferentes terminais no Raspberry Pi;
- Teste 8: Comunicação de dados entre o microcontrolador e o Raspberry Pi;
- Teste 9: Apresentação e registo de dados na página web.

No teste 4 foi possível testar e visualizar a função que captura os dados do ADC fazendo uso do terminal do IDE. A figura 6.9 apresenta os resultados obtidos sem carga associada.



```
Output Serial Monitor x
Message (Enter to send message to 'ESP32')
RedeTEficaz = 1.09
redeCEficaz = 0.04
PotênciaInst = 0.03
PotênciaApar = 0.05
Energia = 0.00
Energia_wh = 0.06
Energia_kwh = 0.00
Fator de Potencia = 61.34
```

Figura 6.9: Exibição dos dados obtidos através do ADC.

No teste 6 foi possível testar a função que calcula a energia dos dados recebidos através do ADC e mostrar os dados de energia no display OLED. Os dados são apresentados na figura 6.10, onde é possível visualizar a tensão, corrente, potência, energia instantânea e energia acumulada de uma carga ativa.

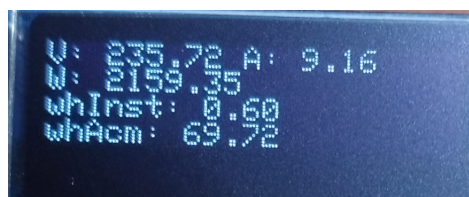


Figura 6.10: Dados de energia apresentados no display com carga ativa.

Depois de executar os testes relativos aos dados capturados e processados foi necessário avançar para a fase de implementação de comunicação sem fios e realizar os devidos testes de forma a validar a comunicação entre ambos os dispositivos. A verificação da comunicação entre o microcontrolador e o Raspberry pi é apresentada na figura 6.11.

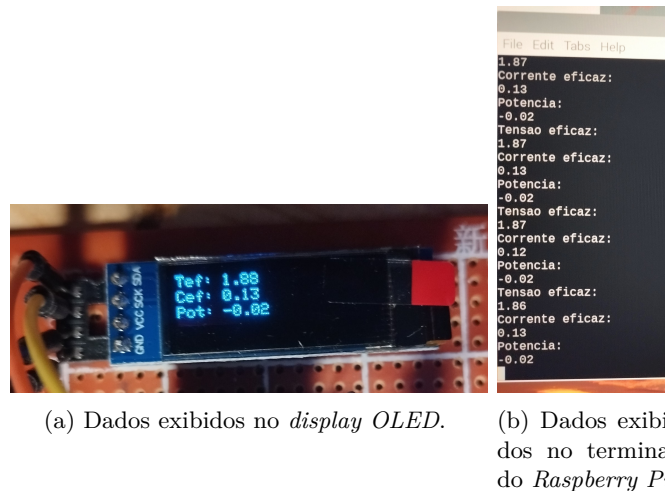


Figura 6.11: Testes à comunicação de dados entre o ESP32 e o Raspberry Pi

A figura 6.12 mostra o resultado do teste 9. Nesta imagem é possível visualizar os dados de energia em tensão, corrente, potência, bem como a energia instantânea e o custo associado à energia consumida até ao momento.

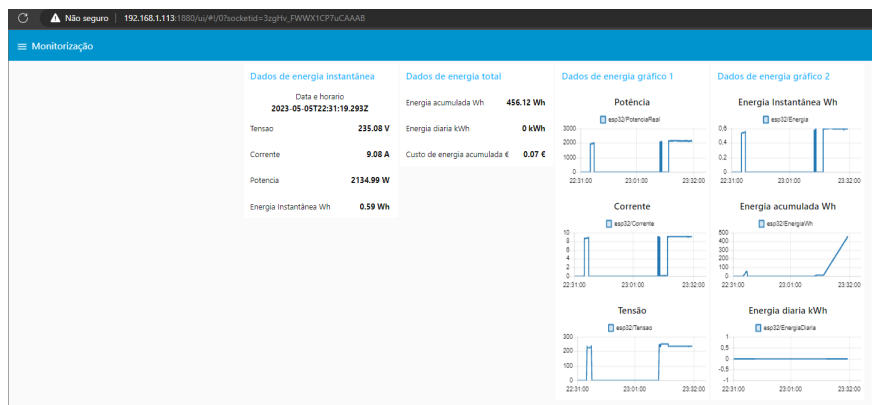


Figura 6.12: Dados de energia apresentados na página WEB com carga ativa.

Na figura 6.13 é apresentada a página web associada ao controlo do LED. Nesta página de controlo é também possível visualizar o estado do LED.

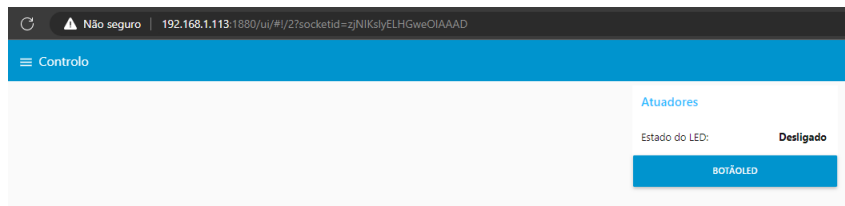


Figura 6.13: Página de controlo.

6.1.3 Testes ao Sistema de monitorização e controlo de carga centralizado

Neste capítulo foram realizados testes ao sistema como um todo. Serão apresentados os dados de energia na página de Internet e no display. Foram também realizado testes à segunda placa de desenvolvimento para que esta acenda/apague o LED através da página de controlo.

Para compararmos os resultados do sistema de monitorização desenvolvido neste projeto é utilizado o aparelho de medição Aubess Tuya Power Monitor Switch [75]. Este aparelho faz parte de um conjunto de dispositivos de controlo e monitorização de cargas lançado para o mercado pela empresa Aubess. Os dados de monitorização e controlo são acedidos através da aplicação desenvolvida pela empresa Tuya. Será através desta aplicação que serão mostrados os resultados que visa comparar os dados apresentados pelo sistema desenvolvido com os dados do dispositivo Aubess.

A figura 6.14 apresenta a aplicação Tuya com informações sobre a corrente instantânea em miliampere, a potência instantânea em Watt, a tensão em Volt e a potência total acumulada em kWh do respetivo dia e também a potência total acumulada em kWh ao longo do ano corrente.

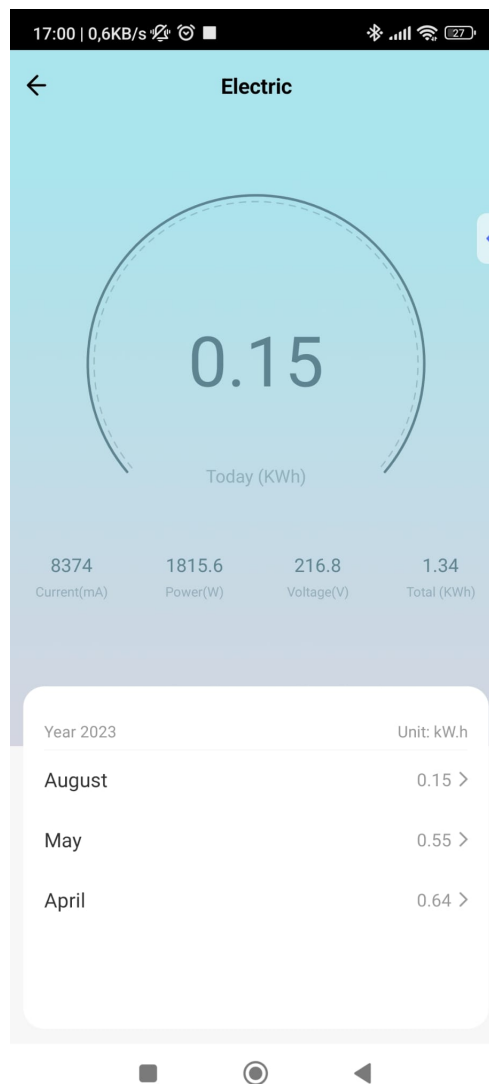


Figura 6.14: Dados de energia apresentados na aplicação Tuya.

A figura 6.15 apresenta a página de monitorização dividida em quatro grupos. O primeiro grupo diz respeito aos dados de energia instantâneos e apresentam os dados de tensão em Volt, a corrente em Ampere, potência em Watt e energia instantânea em Watt-hora. Já no grupo Dados de energia total são apresentados os dados sobre a energia acumulada em Watt-hora, a energia diária em quilowatt-hora e o custo da energia acumulada em euros tendo por base 0.15€/kWh. Os outros dois grupos apresentam os dados de energia mas em forma de gráfico.

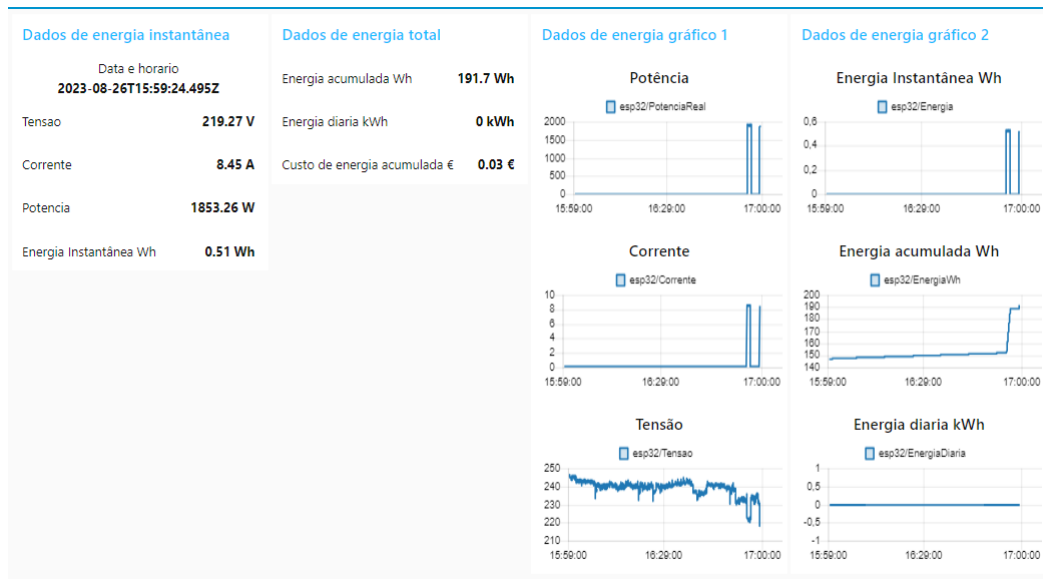


Figura 6.15: Página de Monitorização.

Na figura 6.16 são apresentados os dados sobre a potência aparente em Volt-Ampere, fator de potência, energia instantânea em Watt-hora, Energia instantânea acumulada em Watt-hora e por fim a energia diária em quilowatt-hora. Estes dados de energia são apresentados em complemento aos dados de energia apresentados na página de monitorização.



Figura 6.16: Página de Dados de energia completo.

Na figura 6.17 a tensão é apresentada em Volt na primeira linha do lado direito do V. A corrente é apresentada em Ampere na primeira linha do lado direito do A. A potência é apresentada em Watt na segunda linha do lado direito do W. A energia instantânea é apresentada em Watt-hora na terceira fila do lado direito do whInst. Por fim, a potência acumulada em Watt-hora é apresentada na quarta linha do lado direito do whAcm.

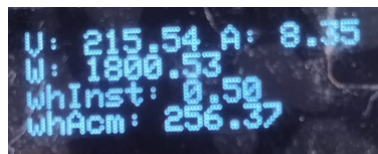


Figura 6.17: Dados de energia apresentados no display.

A figura 6.18 apresenta na página de controlo o botão que permite alterar o estado do LED e a informação sobre o estado do LED. O seu estado é alterado quando o botão LED deixa de ser pressionado.

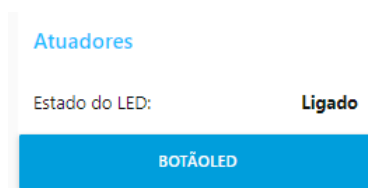


Figura 6.18: Informação e controlo do estado do LED na página de controlo.

A figura 6.19 apresenta o estado do LED de acordo com o estado apresentado na página de controlo.

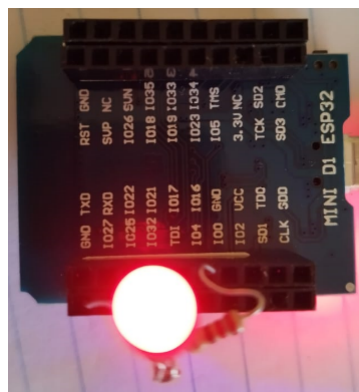


Figura 6.19: Apresentação do estado do LED na placa de desenvolvimento.

Capítulo 7

Conclusões

A aplicação dos conhecimentos adquiridos ao longo do curso foram fundamentais para desenvolver o sistema de monitorização e controlo de energia com interface gráfica, mostrando os consumos diários e instantâneos, bem como o controlo de interruptor externo.

Além do conhecimento técnico foi também fundamental estruturar o sistema de monitorização e controlo de carga centralizado. O sistema foi dividido em dois subsistemas, de *hardware* e de *software* e cada subsistema foi dividido em cinco blocos funcionais. Esta estrutura permitiu analisar, implementar e testar cada bloco funcional garantindo o funcionamento do projeto com a qualidade pretendida.

O desenvolvimento deste sistema teve alguns desafios, entre os quais o ruído presente na placa elétrica utilizada como protótipo, a realização da comunicação entre o microcontrolador e o ADC externo, o processamento dos dados do sinal analógico, a comunicação entre o microcontrolador e o servidor e o desenvolvimento da página de Internet. Um dos maiores desafios esteve relacionado com o desenvolvimento do *software* no microcontrolador, mais especificamente na conjugação da execução da interrupção temporizada com a execução da comunicação WIFI. O microcontrolador não executava simplesmente o código. Testou-se cada função individualmente e ambas funcionavam bem mas quando se integrava as funções o microcontrolador não inicializava.

Ultrapassados os desafios, pode-se dizer que este projeto foi concluído com sucesso. Este sistema possibilita a medição de tensões elétricas eficazes até 230V AC,

correntes eficazes até 10A AC, potência instantânea real e aparente, fator de potência e o custo monetário associado ao consumo da carga. No display é apresentado a tensão eficaz, corrente eficaz, potência real e a potência acumulada ao longo do dia. Na página gráfica é possível visualizar graficamente a tensão elétrica eficaz, corrente eficaz, potência real e aparente, fator de potência e o custo monetário instantâneo e diário. Este sistema foi desenvolvido com o intuito de ser escalável e portanto adicionou-se um microcontrolador que permite ser controlado através da página gráfica com o objetivo de acender/apagar um LED. Todo o sistema foi desenvolvido tendo um servidor principal que coleta toda a informação e envia comandos aos dispositivos ligados a si. Desta forma, é possível adicionar mais dispositivos, sem necessitar de retrabalhar a implementação dos dispositivos já adicionados.

7.1 Trabalho Futuro

Como melhorias de projeto sugere-se adicionar um display maior podendo assim ser exibida mais informação sobre a carga elétrica. Adicionando novos dispositivos de medição, como por exemplo um medidor do caudal de água ou um medidor de gás. Adicionando mais dispositivos que permitam realizar o controle de cargas. Outra ideia de melhoria interessante seria a possibilidade de abrir/fechar os circuitos elétricos do quadro elétrico, permitindo assim o corte da energia elétrica quando se detetasse um curto circuito ou descarga elétrica. O redimensionamento da resistência *shunt* para que permitisse a passagem de correntes elétricas até 30A. A possibilidade de monitorizar tensões elétricas até aos 400V AC também seria uma melhoria ao sistema.

Referências

- [1] C. Europeia, “2030 climate & energy framework.” Available at https://ec.europa.eu/clima/eu-action/climate-strategies-targets/2030-climate-energy-framework_pt/, Feb. 2022. (Last accessed in 28/02/2022). [Citado na página 1]
- [2] D. G. de Energia e Geologia, “Política energética.” Available at <https://www.dgeg.gov.pt/pt/areas-transversais/relacoes-internacionais/politica-energetica/>, Mar. 2022. (Last accessed in 06/03/2022). [Citado na página 2]
- [3] A. P. de Energias Renováveis, “Anuário apren 2021.” Available at <https://www.apren.pt/pt/publicacoes/apren/anuario-apren-2020/>, Mar. 2022. (Last accessed in 06/03/2022). [Citado nas páginas ix e 2]
- [4] Edison, “Electricity.” Available at <https://web.archive.org/web/20141229033421/http://www.coned.com/history/electricity.asp>, Mar. 2022. (Last accessed in 12/03/2022). [Citado na página 7]
- [5] S. Energy, “The history of the electricity meter.” Available at <https://www.smart-energy.com/features-analysis/the-history-of-the-electricity-meter/>, Mar. 2022. (Last accessed in 12/03/2022). [Citado na página 7]
- [6] EDP, “Contadores inteligentes.” Available at <https://www.edp.pt/particulares/apoio-cliente/contadores/>, Mar. 2022. (Last accessed in 12/03/2022). [Citado na página 7]
- [7] Azure, “Tecnologias e protocolos de iot.” Available at <https://azure.microsoft.com/pt-pt/overview/internet-of-things-iot/iot-technology-protocols/>, Mar. 2022. (Last accessed in 12/03/2022). [Citado nas páginas 8, 29 e 33]
- [8] gearhungry, “Bestmonitordevices.” Available at <https://www.gearhungry.com/best-electricity-usage-monitor/>, Mar. 2022. (Last accessed in 10/03/2022). [Citado na página 8]

- [9] fibaro, “zwave.” Available at <https://www.fibaro.com/en/products/home-center-2/>, Mar. 2022. (Last accessed in 09/03/2022). [Citado nas páginas ix e 9]
- [10] fibaro, “wall plug.” Available at <https://www.fibaro.com/en/products/wall-plug/>, Mar. 2022. (Last accessed in 09/03/2022). [Citado nas páginas ix e 10]
- [11] Amazon, “Fibaro fgwvf-102-5.” Available at <https://www.amazon.es/>, Mar. 2022. (Last accessed in 18/03/2022). [Citado na página 10]
- [12] eyedro, “connection types.” Available at <https://eyedro.com/home-electricity-monitors/>, Mar. 2022. (Last accessed in 10/03/2022). [Citado nas páginas ix, 11 e 12]
- [13] Eyedro, “Aplicação web.” Available at <https://eyedro.com/landlord-and-second-property-energy-metering/>, Mar. 2022. (Last accessed in 17/03/2022). [Citado nas páginas ix e 12]
- [14] Lumin, “Lumin.” Available at <https://www.luminsmart.com/>, Mar. 2022. (Last accessed in 16/03/2022). [Citado nas páginas ix e 13]
- [15] Lumin, “Lumin edge.” Available at <https://edge.luminsmart.com/>, Mar. 2022. (Last accessed in 19/03/2022). [Citado nas páginas ix e 13]
- [16] Lumin, “Lumin.” Available at <https://www.luminsmart.com/platform/smart-electrical-panel/>, Mar. 2022. (Last accessed in 16/03/2022). [Citado nas páginas ix e 14]
- [17] Leviton, “Smart breakers.” Available at <https://www.leviton.com/en/products/residential/load-centers/the-leviton-smart-load-center>, Mar. 2022. (Last accessed in 16/03/2022). [Citado nas páginas ix, 15 e 16]
- [18] Trotec, “Medidor de energia simples.” Available at <https://pt.trotec.com/shop/medidor-de-consumo-eletrico-bx09.html>, Mar. 2022. (Last accessed in 12/03/2022). [Citado nas páginas ix e 17]
- [19] Yageo, *Data sheet: Shunt resistor*, 2019. [Citado na página 17]
- [20] dewesoft, “Sensores de corrente.” Available at <https://dewesoft.com/br/aquisicao-de-dados/como-medir-corrente-usando-sensores-de-corrente>, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix e 18]
- [21] Rishabh Instruments Pvt. Ltd., *Data sheet: Shunts Class 0.5, 1.0*, 2021. [Citado na página 18]

- [22] mauser, “resistencia shunt.” Available at https://mauser.pt/catalog/product_info.php?cPath=324_808&products_id=096-8473, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix e 18]
- [23] Vishay Mills, *Data sheet: Wirewound Resistor, Open Style, Current Shunts, Very Low Value*, 2021. [Citado nas páginas ix e 19]
- [24] Vishay Dale, *Data sheet: Power Metal Strip® Resistors, Very High Power (to 10 W), Low Value (down to 0.0002), Surface Mount*, 2022. [Citado nas páginas ix e 19]
- [25] sparkfun, “Sensores de efeito hall.” Available at <https://learn.sparkfun.com/tutorials/current-sensor-breakout-ac723-hookup-guide>, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix, 19 e 20]
- [26] A. Systems, “Current sensors.” Available at <https://www.allegromicro.com/en/products/sense/current-sensor-ics>, Mar. 2022. (Last accessed in 16/03/2022). [Citado na página 20]
- [27] A. Systems, “Sensores de efeito hall.” Available at <https://www.allegromicro.com/en/insights-and-innovations/technical-documents/hall-effect-sensor-ic-publications/hall-effect-current-sensing-in-hybrid-electric-vehicle-hev-applications>, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix e 20]
- [28] pttrobotics, “Sensores de efeito hall.” Available at <https://www.pttrobotics.com/sensor-de-corrente/3085-ac711ex-current-sensor-carrier-31a-to-31a.html>, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix, 20 e 21]
- [29] Allegro MicroSystems, *Data sheet: ACS711 - Hall-Effect Linear Current Sensor with Overcurrent Fault Output for <100 V Isolation Applications*, 2022. [Citado na página 20]
- [30] T. L. Companhoni, “Dispositivo microcontrolado alimentado por transformador de corrente e supercapacitor por meio do princípio de colheita de energia,” Master’s thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. [Citado nas páginas ix e 21]
- [31] electrofun, “transformador de corrente.” Available at <https://www.electrofun.pt/sensores-arduino/sensor-de-corrente-nao-invasivo-30a>, Mar. 2022. (Last accessed in 13/03/2022). [Citado nas páginas ix e 22]
- [32] YHDC, *Data sheet: Split core current transformer*, -. [Citado na página 22]

- [33] D. Coleman, “Rms voltage calculator.” Available at <https://elekonika.com/rms-voltage-calculator/>, Mar. 2022. (Last accessed in 28/03/2022). [Citado na página 23]
- [34] OpenEnergyMonitor, “An introduction to ac power.” Available at <https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/introduction>, Mar. 2022. (Last accessed in 28/03/2022). [Citado nas páginas ix e 24]
- [35] M. F. Alves, “Abc dos circuitos elétricos em corrente alternada,” Feb. 1999. [Citado nas páginas ix e 25]
- [36] D. Reynders, S. Mackay, and E. Wright, *Practical industrial data communications: best practice techniques*. Newnes, 1 ed., 2005. [Citado na página 27]
- [37] A. M. . H. Cassimally, *Designing the Internet of Things*. Adrian McEwen, Hakim Cassimally, 1 ed., 2014. [Citado na página 28]
- [38] M. . Company, “What is the internet of things?.” Available at <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-the-internet-of-things#/>, Aug. 2023. (Last accessed in 16/08/2023). [Citado na página 28]
- [39] M. . Company, “Iot value set to accelerate through 2030: Where and how to capture it.” Available at <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/iot-value-set-to-accelerate-through-2030-where-and-how-to-capture-it>, Aug. 2023. (Last accessed in 16/08/2023). [Citado nas páginas x e 28]
- [40] LoRa® Alliance, *A technical overview of LoRa® and LoRa WAN™*, 2015. [Citado nas páginas x, 29 e 30]
- [41] L. Alliance, “What is lorawan® specification.” Available at <https://loro-alliance.org/about-lorawan/>, Apr. 2022. (Last accessed in 16/04/2022). [Citado na página 29]
- [42] Botnroll, “Rfm95.” Available at <https://www.botnroll.com/pt/rf-lora/2600-rfm95-m-dulo-lora-868mhz.html>, Apr. 2022. (Last accessed in 15/04/2022). [Citado na página 30]
- [43] Bluetooth, “Bluetooth® wireless technology.” Available at <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>, Apr. 2022. (Last accessed in 07/04/2022). [Citado na página 31]
- [44] Botnroll, “Módulo bluetooth 4.0 at-09 ble ti cc2541 (hm-10).” Available at <https://www.botnroll.com/pt/bluetooth/>

- 3034-m-dulo-bluetooth-4-0-at-09-ble-ti-cc2541-hm-10.html, Apr. 2022. (Last accessed in 15/04/2022). [Citado na página 31]
- [45] Leverage, “Introduction to iot - what is wifi.” Available at <https://www.leverage.com/iot-ebook/iot-wifi>, Apr. 2022. (Last accessed in 08/04/2022). [Citado na página 31]
- [46] W.-F. Alliance, “Wi-fi halow™—worth the wait.” Available at <https://www.wi-fi.org/beacon/kevin-walsh/wi-fi-halow-worth-the-wait>, Apr. 2022. (Last accessed in 09/04/2022). [Citado na página 32]
- [47] W.-F. Alliance, “Wi-fi certified halow.” Available at <https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-halow>, Apr. 2022. (Last accessed in 09/04/2022). [Citado nas páginas x e 32]
- [48] Botnroll, “Esp32-devkitc-32d - placa de desenvolvimento espressif.” Available at <https://www.botnroll.com/pt/ethernet-wi-fi/2452-placa-de-desenvolvimento-esp32-espressif.html>, Apr. 2022. (Last accessed in 15/04/2022). [Citado na página 32]
- [49] Botnroll, “Xbee 2mw com antena - serie 2 (zigbee mesh).” Available at <https://www.botnroll.com/pt/zigbee-xbee/654-xbee-2mw-com-antena-serie-2-zigbee-mesh.html>, Apr. 2022. (Last accessed in 15/04/2022). [Citado na página 33]
- [50] OASIS, “Mqtt version 5.0.” Available at https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#_Toc3901018, Mar. 2022. (Last accessed in 23/03/2022). [Citado nas páginas 33 e 34]
- [51] M. Essentials, “Mqtt publish, subscribe & unsubscribe - mqtt essentials: Part 4.” Available at <https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>, Mar. 2022. (Last accessed in 23/03/2022). [Citado nas páginas x e 34]
- [52] MQTT, “Mqtt: The standard for iot messaging.” Available at <https://mqtt.org/>, Mar. 2022. (Last accessed in 23/03/2022). [Citado nas páginas x e 34]
- [53] R. nerd tutorials, “What is mqtt and how it works.” Available at <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>, Mar. 2022. (Last accessed in 23/03/2022). [Citado nas páginas x e 35]
- [54] CoAP, “Rfc 7252 constrained application protocol.” Available at <https://coap.technology/>, Apr. 2022. (Last accessed in 11/04/2022). [Citado na página 35]

- [55] M. T. R. K. . Muhammad Ashar Tariq 1, Murad Khan 2 and D. Kim, “Enhancements and challenges in coap—a survey,” *Sensors*, vol. 20, p. 29, 2020. [Citado nas páginas x e 36]
- [56] Mauser, “Transformador de alimentação encapsulado para pcb 100..240vac.” Available at https://mauser.pt/catalog/product_info.php?cPath=23_3033&products_id=035-0714, Mar. 2023. (Last accessed in 19/03/2023). [Citado na página 39]
- [57] TRACO POWER, *Data sheet: TMU 3 Series, 3 Watt*, 2022. [Citado na página 39]
- [58] Avago Technologies, *Data sheet: ACPL-785J*, 2015. [Citado nas páginas x e 42]
- [59] Texas Instruments, *Data sheet: Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference*, 2009. [Citado na página 43]
- [60] Espressif, “Analog to digital converter.” Available at <https://docs.espressif.com/projects/esp-idf/en/v4.2/esp32/api-reference/peripherals/adc.html>, Mar. 2023. (Last accessed in 22/03/2023). [Citado na página 44]
- [61] Microchip, *Data sheet: MCP3002*, 2011. [Citado nas páginas x, xi, 44, 45, 46, 49, 51, 52 e 75]
- [62] Espressif Systems, *Data sheet: ESP32 Series*, 2009. [Citado na página 47]
- [63] electronicparts, “Mini d1 esp32.” Available at <https://www.eelectronicparts.com/products/0-66-inch-oled-wemos-mini-d1-esp8266-cp2104-esp32-32s-wifi-bluetooth-development-board>, May 2023. (Last accessed in 02/07/2023). [Citado nas páginas x e 48]
- [64] P. Supplies, “Oled 0.91 128×32 i2c white display.” Available at <https://protosupplies.com/product/oled-0-91-128x32-i2c-white-display/>, Mar. 2023. (Last accessed in 31/03/2023). [Citado nas páginas x e 49]
- [65] Espressif Systems, *ESP32 Technical Reference Manual*, 2023. [Citado nas páginas x e 50]
- [66] Expressif, “Spi master driver.” Available at https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/spi_master.html#timing-considerations, Aug. 2023. (Last accessed in 12/08/2023). [Citado nas páginas x e 52]
- [67] SOLOMON SYSTECH, *Data sheet: SSD1306*, 2010. [Citado nas páginas x, 54 e 55]

- [68] NXP Semiconductors, *I2C-bus specification and user manual*, 2021. [Citado nas páginas x e 54]
- [69] Expressif, “Wi-fi api.” Available at <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html>, Aug. 2023. (Last accessed in 03/08/2023). [Citado nas páginas x, 56 e 87]
- [70] Expressif, “Wi-fi esp32.” Available at https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html, Aug. 2023. (Last accessed in 06/08/2023). [Citado na página 56]
- [71] E. Mosquitto, “An open source mqtt broker.” Available at <https://mosquitto.org/>, Apr. 2023. (Last accessed in 07/04/2023). [Citado na página 59]
- [72] Node-RED, “Low-code programming for event-driven applications.” Available at <https://nodered.org/>, Mar. 2022. (Last accessed in 25/03/2022). [Citado nas páginas x, xi, 59, 60 e 61]
- [73] B. Baker, *A Baker’s dozen: real-world analog solutions for digital designers*. Elsevier Inc, 1 ed., 2005. [Citado nas páginas xi, 62, 63 e 64]
- [74] Expressif, “Esp-mqtt.” Available at <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/protocols/mqtt.html?highlight=mqtt>, Aug. 2023. (Last accessed in 06/08/2023). [Citado na página 88]
- [75] Aubess, “Power monitor switch.” Available at https://pt.aliexpress.com/item/1005004478082353.html?spm=a2g0o.store_pc_topSellerIng.8148356.10.296d462aSZQZfX&pdp_npi=3%40dis%21EUR%21%E2%82%AC%2011%2C86%21%E2%82%AC%200%2C59%21%21%21%21%4021038eda16930622056118213e133f%2112000034156884482%21sh%21PT%21163950685, Aug. 2023. (Last accessed in 26/08/2023). [Citado na página 102]
- [76] R. Mauser, “Transformador de alimentação encapsulado.” Available at https://mauser.pt/catalog/product_info.php?cPath=1667_2889_2619&products_id=096-8586, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [77] Farnel, “Tmu 3-0511.” Available at <https://pt.farnell.com/tracopower/tmu-3-0511/dc-dc-converter-5v-0-6a/dp/4058572>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [78] Farnel, “Rpi3-modbp.” Available at <https://pt.farnell.com/raspberry-pi/rpi3-modbp/sbc-board-raspberry-pi-3-model/dp/2842228>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]

- [79] R. Mauser, “Módulo wifi - espressif esp32.” Available at https://mauser.pt/catalog/product_info.php?cPath=1667_2889_2619&products_id=096-8586, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [80] UBUY, “Mini d1 esp32.” Available at <https://www.ubuy.com.pt/pt/product/B5KLZ1PWQ>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [81] Mouser, “Acpl-785j-060e.” Available at <https://pt.mouser.com/ProductDetail/Broadcom-Avago/ACPL-785J-060E?qs=4GD1ZkxJ45moWZx2%252BC9LkQ%3D%3D>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [82] Farnell, “Mcp3002.” Available at <https://pt.farnell.com/microchip/mcp3002-i-p/ic-adc-10bit-200ksps-pdip-8/dp/1852015?st=mcp3002>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [83] R. Mauser, “Display lcd oled 0,91” 128x32 (ssd1306) - azul.” Available at https://mauser.pt/catalog/product_info.php?cPath=1667_2604_2609&products_id=096-8754, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [84] Mouser, “Resistência *shunt*.” Available at <https://pt.mouser.com/ProductDetail/Vishay-Dale/WSBS8536L1250JK40?qs=DRkmTr78QAQkEXI49uI2ug%3D%3D>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]
- [85] L. Merlin, “Tomada 2p+t.” Available at <https://www.leroymerlin.pt/produtos/electricidade-e-smart-home/interruptores-e-tomadas/todos-os-interruptores-e-tomadas/tomada-2p-t-forix-branco-17377276.html?src=clk>, May 2023. (Last accessed in 02/05/2023). [Citado na página 117]

Anexo A

Estimativa do custo do projeto

Custo do material utilizado:

HLK-PM01: 4,2€ [76]

TMU 3-0511: 13,5€ [77]

Raspberry Pi 3 model B+: 32€ [78]

ESP32: 15€ [79]

Mini WEMO D1: 7€ [80]

ACPL-785J: 4,2€ [81]

MCP3002: 2,5€ [82]

OLED display: 5,4€ [83]

Resistência *shunt*: 20€ [84]

Tomada 2P+T: 3.5€ [85]

Diversos: 15€

Custo total aproximado: 120€.