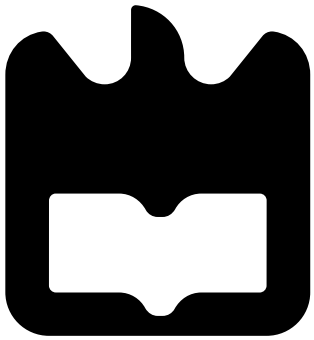
Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2017

João Eduardo Terlica Lagoa

Sistemas embutidos de muito baixo consumo para redes LPWAN



Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2017

João Lagoa

Sistemas embutidos de muito baixo consumo para redes LPWAN

"If knowledge can create problems, it is not through ignorance that we can solve them."

— Isaac Asimov

João Eduardo Terlica Lagoa

Sistemas embutidos de muito baixo consumo para redes LPWAN

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Pedro Fonseca, Professor Auxiliar do Departamento Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor Paulo Bacelar Reis Pedreiras
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Professor Doutor Jorge Augusto Fernandes Ferreira
Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro (Arguente Principal)

Professor Doutor Pedro Nicolau Faria da Fonseca
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

Agradeço ao meu orientador, Professor Doutor Pedro Fonseca, pela ajuda na escrita do relatório, pela paciência, disponibilidade e todas as sugestões que me deu.

Agradeço ao Deniz Martins, ao Marco Camisão e também a todos os funcionários da Exatronic pelos bons momentos passados e pelo conhecimento que me deram.

Por fim, um grande obrigada à minha família e a todos os que me acompanharam nestes anos.

Resumo

Low Power Wide Area Networks (LPWANs) é a designação de um conjunto de soluções que procuram ligar milhares de sistemas de baixo consumo energético, de baixo custo e de baixas necessidades de transmissão. Embora sejam colocadas no mesmo grupo, nenhuma LPWAN é igual e, como tal, é necessário determinar as características de cada uma para que seja possível identificar em que situações quais são as preferíveis.

Nesta dissertação propõe-se o desenvolvimento de um sistema de baixo consumo, com uma previsão de vida de 10 anos, que irá fazer contagem de água de até 4 habitações diferentes. Este sistema faz uso das soluções LPWAN para enviar e receber mensagens.

O consumo medido do sistema é dependente da sua configuração, sendo possível atingir objetivo de vida parametrizando adequadamente o *hardware*, nomeadamente no que respeita à velocidade de relógio usado para o processamento e comunicação.

Abstract

Low Power Wide Area Networks (LPWANs) is the common designation of a set of solutions that seek to connect thousands of low energy, low cost and low bandwidth transmission systems. Despite belonging to the same group, LPWAN technologies are quite diverse and, as such, it is necessary to determine their characteristics to be able to identify which one is the best suited for each particular case.

In this dissertation, the development of a low consumption metering system was proposed, with a life expectancy of 10 years. The system shall measure the water consumed by up to 4 different households and will use a LPWAN solution exchange messages with a central supervisory system.

The consumption of the system depends on several configuration parameters. It is possible to reach the life expectancy through an appropriate customization of the hardware, namely by using adequate clock sources for processing and communication.

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas

16-QAM 16-Quadrature Amplitude Modulation.

3GPP 3rd Generation Partnership Project.

AL Application Layer.

API Application Programming Interface.

ARIB Association of Radio Industries and Businesses.

ATIS Alliance for Telecommunications Industry Solutions.

BPSK Binary Phase-Shift Keying.

CCSA China Communications Standards Association.

CDMA Code Division Multiple Acess.

DBPSK Differential Binary Phase-Shift Keyin.

ETSI European Telecommunications Standards Institute.

FDMA Frequency Division Multiple Access.

FIFO First In First Out.

GPRS General Packet Radio Service.

GSM Global System for Mobile Communications.

HL Hidden Layer.

HSI High Speed Internal.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

ISM Industrial, Scientific and Medical.

ITU-R International Telecommunication Union Radio Sector.

LANs Local Area Networks.

LL Logic Layer.

LoRa Long Range.

LPWAN Low-Power Wide-Area Networks.

LTE Long-Term Evolution.

MIMO Multiple-Input Multiple-Output.

MSI Multi-Speed Internal.

MTC Machine-Type Communication.

NB-IoT Narrow Band-Internet of Things.

PAC Porting Authorization Code.

PAN Personal Area Network.

PCB Printed Circuit Board.

PSM Power Saving Mode.

R-FDMA Random Frequency Division Multiple Access.

RPMA Random Phase Multiple Access.

RTC Real Time Clock.

SINR Signal to Interference Ratio.

TDMA Time Division Multiple Access.

TSDSI Telecommunications Standards Development Society.

TTA Telecommunications Technology Association.

TTC Telecommunication Technology Committee.

UNB Ultra Narrow Band.

USRP Universal Software Radio Peripheral.

Capítulo 1

Introdução

1.1 Motivação

O desenvolvimento da tecnologia de sensores e da sua aplicabilidade tem facilitado o estudo de particularidades de áreas como a agricultura e a medicina, por exemplo. Este desenvolvimento caracteriza-se pelo aumento da precisão das medidas, pela diminuição do consumo de energia por medição e na transmissão de dados, e pelo número elevado de sensores presentes por área. Para sustentar as últimas duas características foram desenvolvidas soluções de conectividade conhecidas por *Low-Power Wide-Area Networking* (LPWAN). Estas soluções não procuram exclusivamente a conexão de sensores, mas a integração deles numa Internet of Things (IoT).

As redes LPWAN apresentadas neste trabalho podem ser divididas em dois tipos: as que usam bandas ISM (ver secção 2.1), disponíveis para qualquer uso, desde que seja cumprido o regulamento de utilização; e as que usam redes celulares já existentes. Também existem soluções que não são consideradas LPWANs, quer pela falta de opções de controlo de consumo de energia, quer pela limitada distância de comunicação, mas que também podem ser empregues na construção deste tipo de redes.

Para determinar qual ou quais as mais apropriadas em certas situações é necessário ter uma noção das características das principais redes existentes no mercado, as suas vantagens e os seus problemas.

Nesta dissertação vai ser feito o estudo das tecnologias descritas e a sua aplicação num sistema que irá ler informação de até quatro contadores de água iguais (ver secção 4.1.3), controlar as suas válvulas (ver secção 4.1.4) e ter uma sobrevivência mínima de 10 anos com apenas uma bateria.

1.2 Objetivos

Nesta dissertação pretende-se estudar algumas tecnologias LPWAN. A rede integrada no sistema final deve obedecer aos seguintes requisitos:

- permitir a comunicação entre dispositivos terminal e central (a conexão entre eles só precisa de ser ativada quando o dispositivo terminal está a enviar informação),
- transmitir informação do dispositivo terminal para o central, sendo que o inverso não é necessário. Como não há muita informação a enviar, a taxa de transmissão não tem que ser elevada, e
- operar e ter cobertura na Europa.

O dispositivo desenvolvido neste estudo faz a contagem e a regulação de até 4 sistemas de distribuição de água. Este deve manter-se em funcionamento normal durante um tempo mínimo de 10 anos, com apenas uma bateria, e através das tecnologias LPWAN enviar os dados adquiridos.

1.3 A Companhia

O trabalho apresentado nesta dissertação foi feito em colaboração com a empresa *Exatronic - Innovation Insight*. A *Exatronic* é especializada no desenvolvimento e produção de soluções eletrónicas a nível mundial. Os projetos levados a efeito pela companhia abordam diferentes ramos de automação como, por exemplo, domótica e controlo de sistemas médicos.

1.4 Organização do Documento

No capítulo 2 apresenta-se o estado da arte de várias redes LPWANs, procurando-se identificar as mais adequadas para diferentes tarefas e expõem-se, também, algumas características de sistemas de baixo consumo.

No capítulo 3 faz-se a descrição geral de todo o sistema.

No capítulo 4 descreve-se o hardware utilizado na implementação do sistema, apresenta-se o estudo dos consumos correspondentes aos seus diversos modos de funcionamento do seu funcionamento, os problemas detetados no desenvolvimento do mesmo e os resultados medidos.

Por último, no capítulo 5 apresentam-se as conclusões da dissertação em relação aos tópicos abordados e os problemas que foram deixados em aberto.

Capítulo 2

Estado da arte

Neste capítulo introduzem-se várias soluções de comunicação para sistemas de baixo consumo que procuram enviar informação para redes de grandes dimensões e apresentam-se algumas das características principais de sistemas de baixo consumo.

2.1 Introdução às Bandas ISM

As bandas ISM *(Industrial, Scientific and Medical)* são bandas de rádio que pertencem à parte não licenciada do espectro, ou seja, bandas que podem ser usadas por qualquer pessoa. Os standards de bandas ISM foram estabelecidos pelo *International Telecommunication Union Radio Sector* (ITU-R) e adotados pela *US Federal Communications Commission* em Maio de 1985, abrindo as portas às bandas que são usadas atualmente pelas *Local Area Networks* (LANs) [1]. As bandas ISM foram usadas, principalmente, para fins científicos, médicos e industriais.

Uma vez que as bandas ISM pertencem à parte não licenciada do espectro, elas podem ser empregues para a realização de transmissões sem que se tenha de pagar taxas ou obter uma licença de uso. Como tal, começaram a ser utilizadas para fins comercias por empresas como a Sigfox (ver secção 2.2.1), a Weightless (ver secção 2.2.3), entre outras.

Muitos sistemas de comunicação modernos usam bandas ISM. Estes dispositivos incluem LANs nas bandas 915 MHz, 2.45 GHz e 5.8 GHz; sensores nas bandas 915 MHz e 2.45 GHz; equipamento médico; etc.

Frequência	**Largura de banda**	**Região**
433.05 - 434.79 MHz	1.74 MHz	Europa
868 - 870 MHz	2 MHz	Europa
902 - 928 MHz	26 MHz	América do Norte e América do Sul
2.4 - 2.5 GHz	100 MHz	Internacional
5.725 - 5.875 GHz	150 MHz	Internacional
24 - 24025 GHz	250 MHz	Estados Unidos da América e Europa
57 - 64 GHz	7GHz	Estados Unidos da América e Europa
59 - 66 GHz	7GHz	Estados Unidos da América e Europa

Tabela 2.1: Bandas ISM existentes.

A tabela 2.1 apresenta as bandas ISM existentes, o tamanho do canal disponível para transmitir e a região em que essas bandas estão disponíveis.

2.1.1 Limitações das bandas ISM

As bandas ISM disponíveis no mundo são limitadas por um conjunto de restrições impostas pelas autoridades competentes de cada área geográfica. As transmissões estão limitadas, sobretudo, na quantidade de energia do sinal a enviar e pelo *duty cycle* do sinal. O *duty cycle* é definido como sendo a fração de um período de tempo em que um sistema ou dispositivo está ativo. Em algumas regiões, o *duty cycle* de um dispositivo é muito pequeno, o que afeta a quantidade de informação e a frequência de cada transmissão.

Estas restrições foram estabelecidas para evitar interferência excessiva no canal de transmissão e entre dispositivos. Todas as tecnologias que tencionem usar estas bandas devem obedecer às restrições impostas, caso contrário, não é possível garantir a transmissão e integridade da informação dentro da banda [2].

2.2 Soluções LPWAN ISM

As *Low-Power Wide-Area Networking* são tecnologias desenvolvidas com o intento de ligar sistemas de baixo consumo energético, de baixo custo e de baixas necessidades na transmissão de informação. Outra característica importante de LPWANs é a intenção de ligar milhares de dispositivos na mesma rede, possibilitando a comunicação entre eles a longas distâncias. Nesta secção abordam-se algumas dessas tecnologias, as quais usam bandas ISM como meio de comunicação. É feita uma descrição do modelo que é usado na transmissão de dados de cada solução e as suas principais características.

2.2.1 Sigfox

Sigfox é uma tecnologia LPWAN criada e desenvolvida em França pela empresa do mesmo nome. Esta tecnologia está disponível na maior parte da Europa Ocidental e em São Francisco, com operações de teste na América do Sul e Ásia. A tecnologia em questão foca-se em redes similares às usadas em sistemas de telemóvel, mas em vez de fornecer serviços de comunicação utilizador-utilizador, procura conectar equipamento de baixo consumo.

A rede criada pelo protocolo não é capaz de enviar grandes quantidades de dados em pouco tempo, como, por exemplo, emails, devido ao formato de pacotes utilizado e ao estilo de transmissão, sendo usado, preferencialmente, para transmitir informação obtida por sensores. Esta tecnologia permite comunicação bidirecional sendo começada, sempre, pelo dispositivo terminal. A rede é construída numa topologia do tipo estrela, isto é, em que todos os nós se conectam a um dispositivo central, como um *hub*, *switch* ou computador. O dispositivo central age como um servidor e os dispositivos periféricos como clientes, estando os clientes única e exclusivamente a comunicar com o servidor e não entre eles.

A Sigfox usa modulação *Ultra Narrow Band* (UNB), caracterizada pelo uso de uma banda muito estreita de 100 Hz e modulação *Binary Phase-Shift Keying* (BPSK). Devido a isto, o ruído encontrado no canal é muito baixo, por volta de 150 dBm a uma temperatura de 290 K, o que permite a demodulação de sinais de baixa potência, até -142 dBm [3].

A rede criada trabalha sobretudo nas frequências sub-GHz, 868 MHz na Europa e 900 MHz nos Estados Unidos da América. Na banda de 868 MHz o espectro é dividido em 401 canais de 100 Hz, começando em 868.180 MHz para o canal 0 e acabando em 868.220 MHz para o canal 400, como se pode ver na Figura 2.1. Os canais desde 181 a 219 não são usados. Numa transmissão é possível indicar qual dos canais a usar mas, caso não seja especificado, a frequência da onda portadora pode tomar qualquer um dos valores indicados anteriormente. A estação base é configurada para procurar em todo o espectro e utiliza técnicas de processamento de sinal para obter a mensagem, não tendo conhecimento prévio de onde esta virá. Esta característica

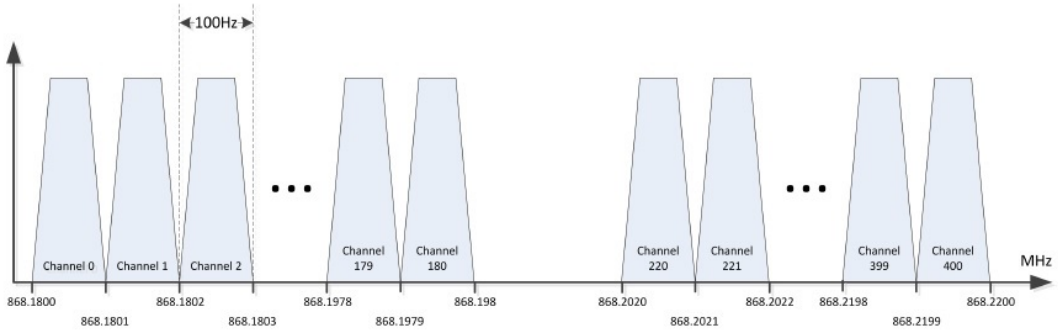


Figura 2.1: Canais usados por Sigfox [3].

é importante uma vez que os nós terminais vão sofrer de envelhecimento dos componentes elétricos, de *jitter* de oscilação, de variação de temperatura, entre outros, o que irá alterar a frequência de envio das mensagens. Este modelo de transmissão denomina-se *Random Frequency Division Multiple Access* (R-FDMA).

Cada mensagem transmitida é enviada 3 vezes, por defeito, o que pode ser modificado. Isto deve-se ao facto de não existir uma mensagem de receção dentro da rede. Logo, dependendo da importância da mensagem, esta pode ser enviada mais ou menos vezes. Cada mensagem retransmitida é enviada num canal diferente, diminuindo assim o efeito de *fading*.

As mensagens transmitidas são protegidas por um algoritmo *hash*, definido como um sistema lógico que comprime uma *string* de informação numa *string* numérica de tamanho fixo. A *string* gerada tende a ser mais pequena que a original. Os algoritmos *hash* são usados, frequentemente, em criptologia e como tal há uma grande variedade deles. A mensagem é decifrada com uma chave privada que é única em cada estação recetora.

Todas as mensagens transmitidas têm um *payload* até 12 bytes. A indicação do tempo em que foi enviada (4 bytes), o número identificador do emissor (2 bytes) e o *hash* (tamanho variável) são enviados separadamente. O código *hash* é usado por defeito para autenticar a *payload* no recetor, em vez de encriptar a mensagem, mas pode também ser usado para a encriptar. Devido às regulamentações europeias, qualquer dispositivo que esteja a usar uma banda não licenciada só pode transmitir através dela até 1% de uma hora. Por outras palavras, uma hora tem 3600 segundos, 1% desse período é 36 segundos. Isto quer dizer um total de 864 segundos por dia, por cada dispositivo. Transmitir uma mensagem pode demorar entre 4 a 6 segundos, máximo de 6 mensagens por hora, o que resulta num total de 140 a 144 mensagens por dia, com uma taxa de transmissão de 100 bits/s. Por outro lado, uma mensagem enviada da estação ocorre sempre após uma transmissão *uplink*, tem apenas 8 bytes e transmitir no máximo até 4 vezes por dia, de forma a diminuir o tempo de escuta dos nós terminais.

Não existe documentação pública sobre as camadas físicas deste protocolo [4]. Em 2015, G. Margelis, R. Piechocki , D. Kaleshi e P. Thomas analisaram a estrutura das mensagens através de USRPs (*Universal Software Radio Peripheral*) [3]. Como referido anteriormente, Sigfox não encripta a *payload*. Para o recetor decifrar a mensagem, o nó terminal envia primeiro um conjunto de bits de sincronização (*sync-word*), que são comparados com os existentes numa tabela interna no emissor e no recetor, seguindo depois a *payload*.

A tecnologia Sigfox enfrenta vários desafios no mercado Norte-Americano. Isto deve-se às regulamentações impostas, que não permitem manter transmissões constantes por mais de 0.4 segundos. Como as transmissões de sistemas que usam esta tecnologia duram mais que isso, é necessário o desenvolvimento de uma nova arquitetura. Além disto, a banda de frequência disponível na América está sujeita a níveis muito mais elevados de interferência do que a banda

utilizada na Europa [5].

Alcance (km)	30-50 áreas rurais; 3-10 áreas urbanas
Bandas (MHz)	868 (Europa); 902 (América) (ISM)
Tamanho do Canal	Ultra Narrow Band
Tamanho do Pacote	12 Bytes
Taxa transferência de dados Uplink	100 bps até 140-144 mensagens/dia
Taxa transferência de dados Downlink	Máximo de 4 mensagens de 8 bytes/dia
Topologia	Estrela
Movimentação/Remoção de nós	Sim

Tabela 2.2: Características do protocolo Sigfox [6].

A Tabela 2.2 resume as características da tecnologia Sigfox.

Requisitos de utilização de Sigfox

Para ter um dispositivo na rede Sigfox é necessário:

- ser compatível com o equipamento Sigfox,
- ter pelo menos um *token* (um *token* é igual a um ano de assinatura para um dispositivo),
- estar localizado numa área coberta pela rede, e
- associar o seu ID, que é um número identificador do dispositivo, e um número PAC (*Porting Authorization Code*), que é uma chave secreta gerada segundo o ID e não transferível caso haja mudança de proprietário ou remoção da rede, na conta, o que pode ser feito através de interface online ou via API (*Application Programming Interface*) [7, 8].

2.2.2 Lora

LoRa

LoRa (*Long Range*) é uma tecnologia desenvolvida pelo fabricante de chips Semtech. Trata-se de um sistema de comunicação sem fios que assegura a ligação entre sensores e uma estação base. Na camada física desta tecnologia é usada a modulação *chirp* (Chirp Spread Spectrum), que é definida pelo uso de uma forma de onda sinusoidal cuja frequência instantânea aumenta (*Upchirp*) ou diminui (*Downchirp*) linearmente ao longo do tempo [9], sem alterar a fase entre símbolos adjacentes. O sinal resultante tem características semelhantes ao ruído, o que dificulta a interceção por entidades exteriores e aumenta a resistência a interferências. Se a variação da frequência for pequena o suficiente, de forma a que haja uma grande quantidade de energia por símbolo, recetores relativamente distantes do local de emissão são capazes de descodificar o sinal, mesmo que este esteja atenuado alguns dBs abaixo do nível de ruído no canal [10].

Esta tecnologia opera em bandas de frequência não licenciadas: 868 MHz na Europa, 915 MHz na América do Norte e 433 MHz na Ásia. A utilização de frequências inferiores a 2,4 GHz permite uma melhor cobertura, especialmente em espaços com muitos obstáculos e dentro de edifícios. Devido à baixa taxa de transmissão (entre 0.3 Kbps e 50 Kbps) e à baixa quantidade de dados a enviar, a transmissão é feita em *narrow band*.

LoRa tem as seguintes larguras de banda de rede disponíveis: 7,8 kHz; 10,4 kHz; 15,6 kHz; 20,8 kH; 31,2 kHz; 41,7 kHz; 62,5 kHz; 125 kHz; 250 kHz e 500 kHz [11]. A largura de canal para a banda 868 MHZ é de 125 kHz e consoante a quantidade de dados necessários a transmitir

e o espaço em que é transmitida, isto é, se há obstáculos, se for dentro de um edifício, etc., usa um ou mais canais. A *payload* pode variar em tamanho entre 2 e 255 bytes.

LoRaWAN

LoRaWAN é o protocolo da camada MAC capaz de conectar uma grande quantidade de dispositivos a longa distância e com um baixo consumo de energia. Este protocolo é usado principalmente para suportar dispositivos LoRa, para melhorar o tempo de vida da bateria e a qualidade de serviço prestado. O protocolo LoRaWAN garante a transmissão em ambas as direções (dispositivo terminal para o recetor e vice-versa), permitindo a entrega de dados e a confirmação de transmissão/receção. Também tem a capacidade de encriptar dados [12].

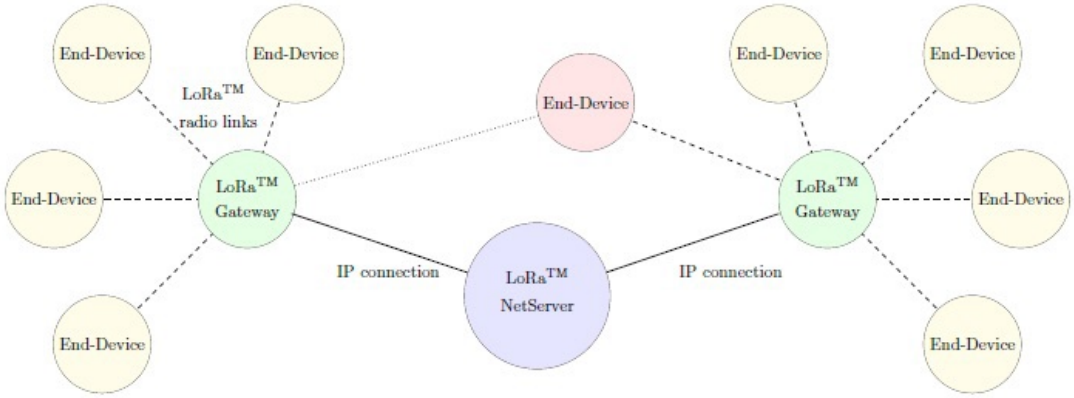


Figura 2.2: Arquitetura da rede LoRaWAN [4].

As redes LoRaWAN são normalmente dispostas numa topologia estrela, como se pode observar na Figura 2.2, em que o dispositivo terminal envia a informação para um recetor e este encaminha-a para um servidor. Os recetores são conectados aos servidores através de uma rede IP, enquanto os dispositivos terminais geralmente usam uma conexão *single-hop* diretamente para os recetores, sem a necessidade de outros equipamentos auxiliares. A comunicação entre os dispositivos terminais e os recetores é dividida em canais de várias frequências com taxas de bits variáveis. A fim de reduzir o consumo de energia e aumentar a capacidade da rede, cada dispositivo LoRa pode ser personalizado individualmente, conforme a necessidade. Os dispositivos terminais podem transmitir a qualquer momento, em qualquer um dos canais disponíveis, com qualquer taxa de bits, em conformidade com os termos e condições a seguir indicados. Cada dispositivo terminal deve:

- respeitar o ciclo de transmissão máximo (*duty cycle*) no canal em uso e estar em conformidade com as regulamentações locais, e
- deve respeitar o tempo máximo permitido de transmissão no canal em uso e estar em conformidade com a regulamentação local.

Classes de Redes de LoRaWAN

A rede LoRaWAN é dividida em três classes: A, B e C. Cada dispositivo deve ser pelo menos compatível com a classe A. As classes B e C oferecem outras opções/complementos. As três classes permitem comunicação bidirecional.

Em relação aos dispositivos da classe A, quando os dados são enviados de um dispositivo terminal para um recetor é aberto um canal que é usado para enviar mensagens curtas entre eles.

A frequência de cada mensagem depende das necessidades de *broadcast* do dispositivo terminal. Esta classe tem o menor consumo de energia para equipamentos terminais e é adequada para sistemas onde os dispositivos terminais são capazes de receber uma mensagem do servidor após o envio de dados. Qualquer outra comunicação do servidor para o dispositivo deve aguardar outro envio de dados.

Um dispositivo de classe B abre um canal de comunicação com o servidor em intervalos regulares. Este canal é utilizado pelo servidor para enviar mensagens para o dispositivo terminal. Desta forma, os dispositivos terminais sabem quando esperar mensagens do servidor e são capazes de manter sincronização com eles. Este método é totalmente integrado no método da classe A.

Comparado com as duas classes anteriores, a classe C fornece uma prontidão constante para receber mensagens do servidor. A única vez que os dispositivos finais não podem receber mensagens do servidor é quando estão a enviar dados. Os dispositivos que usam esta classe requerem maior consumo de energia em comparação com as classes anteriores, mas oferecem resposta mais rápida quando ocorre uma solicitação do servidor para comunicação.

Em suma, o consumo de energia é proporcional à quantidade de tempo que os dispositivos estão à espera de mensagens das estações base [3].

Alcance (km)	15 áreas rurais; 2-5 áreas urbanas
Bandas (MHz)	433 (Ásia); 868 (Europa); 915 (América) (ISM)
Tamanho do Canal	EU: 8x125 kHz; US: 64x125 kHz/8x125 kHz
Tamanho do Pacote	Definido pelo utilizador (2 a 255 bytes)
Taxa transferência de dados Uplink	EU: 300 bps to 50 kbps; US: 900 to 100 kbps
Taxa transferência de dados Downlink	EU: 300 bps to 50 kbps; US: 900 to 100 kbps
Topologia	Estrela
Movimentação/Remoção de nós	Sim

Tabela 2.3: Características dos protocolos LoRa e LoRaWAN [6].

As principais de LoRa caraterísticas encontram-se resumidas na Tabela 2.3.

Requisitos de utilização da tecnologia LoRa

Para um dispositivo ser capaz de utilizar uma rede LoRa precisa de ter componentes feitos pela Semtech para esse efeito. Numa rede privada de pequena dimensão não é necessário o uso de um *router* que suporte LoRa *gateways* para poder conectar os diferentes nós. A rede pode ser gerida por um *MultiConnect Conduit*. Para ligar-se a uma rede já existente de grandes dimensões é necessário um *router* que suporte LoRa *gateways* [13].

Há duas maneiras para um dispositivo conectar-se à rede: ligação *Over-The-Air* e por ativação personalizada. Em ligação *Over-The-Air* os dispositivos terminais enviam um pedido de entrada na rede que inclui o endereço do proprietário, o endereço que os identifica e um *nonce*. O *nonce* é um número aleatório usado com o objetivo de aumentar a segurança de transmissão, para gerar a *Network Session Key* e a *Application Session Key*. A estação base responde com uma mensagem que inclui o endereço do dispositivo terminal, um *nonce*, um identificador de rede e os canais que o dispositivo pode usar. Em ativação personalizada o dispositivo já contém as informações necessárias para entrar na rede LoRa e não executa o pedido de entrada [3].

Exemplos de uso

Em Outubro de 2016 M. Centenaro, L. Vangelista, A. Zanella e M. Zorzi testaram as capacidades da rede LoRa [4]. A primeira experiência focou-se num edifício com 19 andares, em que se pretendia monitorizar a temperatura e a humidade das diferentes divisões. Isto foi realizado com o objetivo de estudar os custos relacionados com o aquecimento e a ventilação. Para tal, foi instalada uma única *gateway* no nono andar e 32 nós por todo o edifício, existindo pelo menos um por andar. O sistema passou todos os testes de conectividade.

A segunda experiência focou-se no estudo da cobertura da rede LoRa na cidade de Pádua em Itália. Pádua tem aproximadamente 100 quilómetros quadrados e cerca de 200.000 habitantes. Concluiu-se que para haver completa cobertura seriam necessários pelo menos 30 *gateways*, o que é metade do número de "antenas" estabelecidas por uma das maiores operadoras telefónicas de Itália. Ou seja, cada *gateway* serviria 7.000 habitantes. Cada *gateway* suporta, pelo menos, 15.000 nós, o que quer dizer que cada pessoa poderia ter até 2 dispositivos na rede. Para aumentar o número de dispositivos na rede bastaria introduzir outra *gateway* [4].

2.2.3 Weightless

A tecnologia Weightless, desenvolvida pela empresa Weightless Special Interest Group (SIG), é composta por três protocolos: Weightless-W, Weightless-N e Weightless-P. Cada protocolo foi projetado para usos diferenciados [5].

O Weightless-W usa a configuração de rede estrela e opera nas bandas de *TV White Space*. Weightless-W é capaz de *frequency hopping* para evitar problemas de congestionamento e de interferência nas bandas de *TV White Space*. Esta tecnologia suporta vários esquemas de modulação de sinal, como o *16-Quadrature Amplitude Modulation* (16-QAM) e o *Differential-BPSK* (DBPSK). Dependendo das necessidades, os pacotes podem tomar até 10 bytes em tamanho e podem ser enviados entre 1 kbps e 10 Mbps [10].

O Weightless-W só é aplicável em lugares onde o *White Space* esteja disponível e quando é necessário uma grande quantidade de funcionalidades. Esta tecnologia é capaz de agendar/pré-planear, após cada transmissão, a próxima transferência de dados [14].

O Weightless-N é uma tecnologia LPWAN, configurada no modelo estrela, que opera nas bandas ISM sub-GHz através de tecnologia UNB. O Weightless-N foi criado à volta da modulação DBPSK, é capaz de encriptar as mensagens usando um método similar ao que a Sigfox usa (ver secção 2.2.1) e é ideal para redes de sensores, como de temperatura, pressão, etc. A empresa Nwave (ver secção 2.2.4) é a principal responsável pela criação e desenvolvimento desta tecnologia [15].

O Weightless-P é uma tecnologia LPWAN que transmite através de um canal com tamanho de 12.5 kHz e oferece transferência de dados do dispositivo terminal para o servidor e vice-versa. A empresa M2COMM (ver secção 2.2.4) é a principal responsável pela criação e desenvolvimento desta tecnologia [16].

Tal como a LoRaWAN (ver secção 2.2.2), todas as tecnologias Weightless usam algoritmos *hash* para assegurar a proteção das mensagens [10].

Após junho de 2017, as soluções Weightless-W e Weightless-N deixaram de ser mencionadas no site oficial. Em setembro de 2017 a Weightless-P foi oficialmente estabelecida com o nome de *Weightless Technology*[17].

Requisitos de utilização de Weightless

Em 14 de Março de 2017 a Weightless lançou o *Weightless Ignition Pack* que permite testar as capacidades do protocolo Weightless-P [18, 19].

	-W	-N	-P
Alcance (km)	5 (urbano)	3 (urbano)	2 (urbano)
Bandas (MHz)	TV white space (400-800 MHz)	Sub-GHz ISM	Sub-GHz ISM
Tamanho do Canal	5 MHZ	Ultra narrow band (200 Hz)	12.5 kHz
Tamanho do Pacote	10 byte min	Up to 20 bytes	10 byte min
Taxa transferência de dados Uplink	1 kbps to 10 Mbps	100 bps	200 bps to 100 kbps
Taxa transferência de dados Downlink	1 kbps to 10 Mbps	Sem downlink	200 bps to 100 kbps
Topologia	Estrela	Estrela	Estrela
Movimentação/ Remoção de nós	Sim	Sim	Sim

Tabela 2.4: Características do protocolo Weightless.

2.2.4 Outras tecnologias

Nesta secção são introduzidas algumas tecnologias LPWAN que, devido a falta de informação, não foram consideradas para o projeto final, mas que merecem ser mencionadas.

M2M Spectrum Networks

M2M Spectrum Networks (M2M SN) é uma empresa norte-americana de comunicação de redes sem fios, máquina-máquina (M2M). Esta tem por principal objetivo conectar dispositivos que por qualquer motivo não podem empregar soluções de conectividade de campo próximo, como Wi-Fi, Bluetooth e RFID, nem precisam de banda larga para transmissão, como os protocolos 4G e LTE [6].

Alcance (km)	Sem informação
Bandas (MHz)	800 MHz; 900 MHz; (Não ISM)
Tamanho do Canal	Sem informação
Tamanho do Pacote	Sem informação
Taxa transferência de dados Uplink	Sem informação
Taxa transferência de dados Downlink	Não tem
Topologia	Sem informação
Movimentação/Remoção de nós	Sem informação

Tabela 2.5: Características do protocolo M2M Spectrum Networks [6].

As suas principais caraterísticas encontram-se resumidas na Tabela 2.5.

Platanus

Platanus é um protocolo concebido com a intenção de manipular altas densidades de nós (até 50.000 clientes sem fios) em curtas distâncias. Este protocolo foi inspirado numa arquitetura de comunicação celular, mas usa bandas ISM Sub-GHz. É propriedade da M2COMM tendo sido doado como base para o protocolo Weightless-P (ver secção 2.2.3) [6].

A Tabela 2.6 resume as principais características deste protocolo.

Alcance (km)	Várias centenas de metros
Bandas (MHz)	Sub-GHz ISM
Tamanho do Canal	Sem informação
Tamanho do Pacote	Sem informação
Taxa transferência de dados Uplink	500 kbps
Taxa transferência de dados Downlink	Não tem
Topologia	Sem informação
Movimentação/Remoção de nós	Sim

Tabela 2.6: Características do protocolo Platanus [6].

NWave

O protocolo Nwave comunica em *Ultra Narrowband* que opera em bandas ISM Sub-GHz. O alcance do sinal varia de 20 a 30 km em áreas rurais e até ao máximo de 10 km em áreas urbanas. Opera numa topologia em estrela.

O Nwave usa uma técnica desenvolvida internamente, à qual chamaram *Advanced Demodulation*, que se destina a permitir a coexistência da sua rede com outras tecnologias de rádio, sem introdução de ruído adicional.

Não se sabe muito sobre esta tecnologia [20].

Alcance (km)	20 até 30 em áreas rurais; 10 em áreas urbanas
Bandas (MHz)	Sub-GHz ISM
Tamanho do Canal	Ultra Narrow Band
Tamanho do Pacote	Cabeçalho de 12 bytes, carga útil de 2 a 20 bytes
Taxa transferência de dados Uplink	100 bps
Taxa transferência de dados Downlink	Sem informação
Topologia	Estrela
Movimentação/Remoção de nós	Sim

Tabela 2.7: Características do protocolo NWave [6].

A Tabela 2.7 resume as principais características deste protocolo.

Ingenu (previamente conhecida por On-Ramp)

A Ingenu é propriedade da On-Ramp Wireless, empresa com sede em San Diego (EUA). A rede criada pela Ingenu adotou uma topologia estrela. O sistema favorece transmissões *uplink* mas também permite transmissões *downlink*.

A Ingenu usa uma tecnologia chamada *Random Phase Multiple Access* (RPMA), patenteada e criada a partir do zero pela On-Ramp Wireless [21]. A RPMA é usada apenas para transmissões *uplink* e o *Code Division Multiple Acess (CDMA)* é usado para transmissões *downlink*.

A RPMA é uma variação do CDMA concebido para ter um melhor *Signal to Interference plus Noise Ratio* (SINR), ou seja, para que o sinal enviado seja mais resistente a interferências externas. A RPMA permite que várias transmissões ocorram ao mesmo tempo. Há duas diferenças principais entre RPMA e CDMA: em CDMA todos os nós utilizam o mesmo código Gold para distribuir os bits na mensagem antes desta ser enviada; além disso todas as transmissões são não sincronizadas e começam com um atraso aleatório. Os recetores conseguem descodificar várias mensagens que chegam num mesmo intervalo de tempo [3, 10].

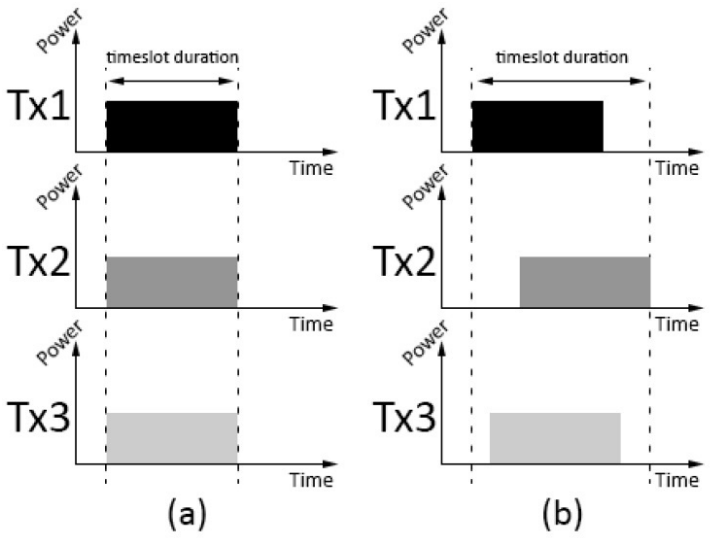


Figura 2.3: Comparação entre os métodos de transmissão de CDMA e RPMA [3].

Ao contrário de outras soluções LPWAN, esta tecnologia funciona na banda de 2,4 GHz mas pode ainda operar através de conexões sem fios a longas distâncias [6].

A Ingenu fornece dois pacotes para criar e testar a rede, o *Development Kit* e o *Exploration Kit*. O *Development Kit* permite a ligação à rede de uma área coberta [22]. O *Exploration Kit* é necessário quando não existe cobertura da rede [23]. Este kit inclui dois *Development Kits* e um ponto de acesso que permite simular a rede.

Alcance (km)	500
Bandas (MHz)	2.4 GHz ISM
Tamanho do Canal	1 MHz (40 canais disponíveis)
Tamanho do Pacote	Apr. 100 a apr. 1000 bytes (normalmente)
Taxa transferência de dados Uplink	156 kbps por setor (Assume-se o ponto de acesso de 8 canais)
Taxa transferência de dados Downlink	156 kbps por setor (Assume-se o ponto de acesso de 8 canais)
Topologia	Tipicamente Estrela
Movimentação/Remoção de nós	Sim

Tabela 2.8: Características do protocolo Ingenu [6].

As suas principais caraterísticas encontram-se resumidas na Tabela 2.8.

2.3 Soluções LPWAN Celulares

Nesta secção são abordadas algumas tecnologias de comunicação LPWAN Celulares. Estas tecnologias diferem das tecnologias que usam bandas ISM pelo facto de usarem sistemas de redes de comunicação telefónica já existentes que permitem transferências entres os diferentes nós integrados na rede. É feita uma descrição do modelo que é usado na transmissão de dados de cada solução e as suas principais características.

2.3.1 GSM/GPRS

Esta tecnologia foi desenvolvida pelo ETSI (*European Telecommunications Standards Institute*) em 1992. O GSM (*Global System for Mobile Communications*, conhecido previamente por *Groupe Spécial Mobile*) é considerado o standard da tecnologia 2G, permanecendo em uso nos dias de hoje. Uma das principais razões do uso contínuo do GSM foi o desenvolvimento de tecnologias como o GPRS (*General Packet Radio Service*), que se integrou no modelo GSM, melhorando a velocidade de transferência de 9.6 kbps para 171 kbps [24], entre outras novas funcionalidades [25]. Outra razão para o sucesso do GPRS foi a fácil integração em sistemas GSM físicos já existentes.

O aumento da velocidade de transferência deve-se ao facto de ter passado a utilizar a modulação FDMA (*Frequency Division Multiple Access*), em que os canais são separados em frequências diferentes, sendo cada canal ocupado por uma única transmissão, para modulação TDMA (*Time Division Multiple Access*) em que os sinais em vez de serem enviados em formato continuo são multiplexados temporalmente num único canal [26].

Alcance (km)	Sem Informação
Bandas (MHz)	890 a 915 MHz, 935 a 960 MHz
Tamanho do Canal	200 kHz
Tamanho do Pacote	Sem Informação
Taxa transferência de dados Uplink	171.2 kbps
Taxa transferência de dados Downlink	171.2 kbps
Topologia	Celular
Movimentação/Remoção de nós	Sim

Tabela 2.9: Características do protocolo GPRS [27].

A Tabela 2.9 resume as principais características deste protocolo.

2.3.2 LTE-M

O Long-Term Evolution (LTE) é uma solução desenvolvida pelo 3GPP (*3rd Generation Partnership Project*). O 3GPP une sete organizações dedicadas ao desenvolvimento de serviços de telecomunicações (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA e TTC). Também são conhecidos por *Organizational Partners*. O LTE é caracterizado pelo uso de sistemas de telefone já existentes para a criação de redes. Com o lançamento do *Release 13*, surgiram duas novas versões do standard, a NB-IoT (ver secção 2.3.3) e a LTE-M, que possuem uma largura de banda de 200 kHz e 1.4 MHz, respetivamente.

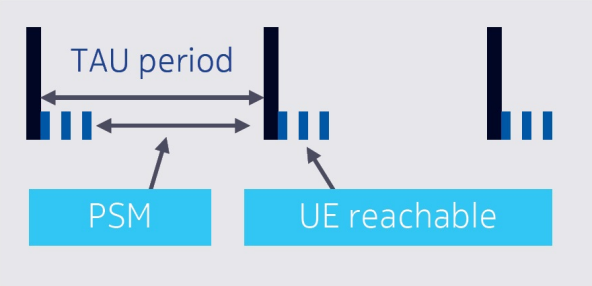


Figura 2.4: Ciclo do modo PSM [28].

O protocolo LTE-M é capaz de melhorar o tempo de vida de dispositivos, colocando-os em

PSM (*Power Saving Mode*). Um dispositivo que suporte esta funcionalidade irá pedir à sua rede um *active timer*, criando um período em que o dispositivo é alcançável por outras partes da rede e que quando acaba, coloca-o em modo PSM. O dispositivo mantém-se em modo PSM até que uma transmissão *uplink* ocorra [28].

No *Release 11* foi introduzido a funcionalidade *device triggering*. Esta funcionalidade permite que os servidores MTC (*Machine-Type Communication*) comuniquem com dispositivos inativos através de SMS, sem a necessidade de alocação de IP [29].

Também no *Release 11* foi adotado o uso de *Low Access Priority Indicator*. Este indicador permite, em caso de congestionamento, identificar qual é o tráfego de maior importância na rede [29].

LTE-M é retrocompatível com outros sistemas LTE. Como tal, consegue beneficiar de qualquer rede telefónica que use LTE para criar redes LTE-M ou enviar dados [30].

Alcance (km)	2.5 até 5
Bandas (MHz)	700 a 900 MHz
Tamanho do Canal	1.4 MHz
Tamanho do Pacote	100 até 1000 Bytes (tipicamente)
Taxa transferência de dados Uplink	1 Mbps
Taxa transferência de dados Downlink	1 Mbps
Topologia	Celular
Movimentação/Remoção de nós	Sim

Tabela 2.10: Características do protocolo LTE-M.

As suas principais caraterísticas encontram-se resumidas na Tabela 2.10.

2.3.3 NB-IoT

O *Narrow Band-Internet of Things* (NB-IoT) é um standard desenvolvido pelo 3GPP, tal como o LTE (ver secção 2.3.2). O NB-IoT foi desenvolvido com o objetivo de minimizar o custo de produção e de consumo de enrgia. Para garantir estes objetivos procurou-se manter o modelo de comunicação o mais simples possível, removendo a maioria das funcionalidades de controlo e comunicação do LTE [31].

O NB-IoT é capaz de 3 modos de implementação diferentes, consoante a forma de comunicação desejada entre os nós.

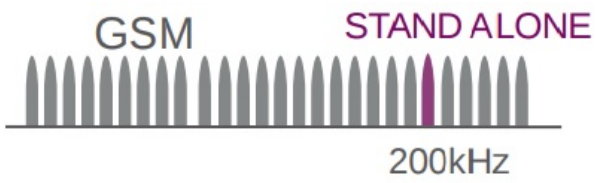


Figura 2.5: Operação *Stand-alone* [32].

O modo *Stand-alone* foi desenhado para sistemas já existentes em que o principal método de transmissão é o GSM (ver secção 2.3.1). Este modo procura utilizar algumas das ondas transportadoras do GSM para transmitir o tráfego de NB-IoT, como se pode ver na Figura 2.5 [33] .

Para operadores que possuem sistemas LTE, os modos *guard band* e *inband* estão disponíveis, sendo o modo *inband* o mais eficiente dos dois. A principal diferença entre os dois modos é a colocação da onda portadora no pacote enviado, como se pode ver na Figura 2.6.

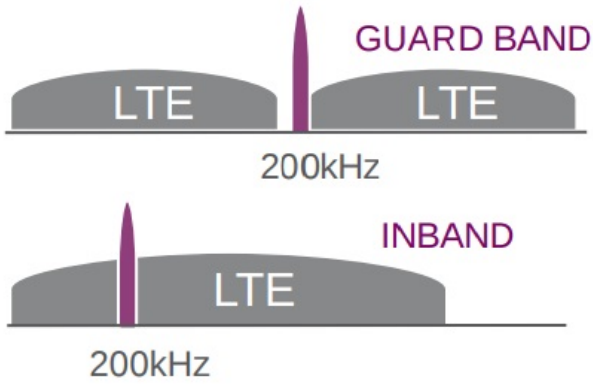


Figura 2.6: Operação em LTE *guard band* e *inband*, respetivamente [32].

No caso do modo *guard band*, o sinal é colocado na zona de banda de guarda sendo esta zona definida como o espaço entre dois canais de transmissão, para diminuir a interferência entre canais.

No caso do modo *inband*, o sinal NB-IoT é completamente integrado no sinal LTE funcionando como qualquer outro bloco LTE. A estação é responsável pela multiplexagem do sinal.

Deste modo, o canal *uplink* pode funcionar em *single-tone* (15 kHz ou 3.75 kHz), o que quer dizer que o sinal enviado pode tomar dois estados lógicos, como um sinal binário ou em *multi-tone* (n*15 kHz, em que n pode ir até ao valor 12), em que o sinal pode tomar n+1 níveis. Os pacotes transferidos neste modo têm um tamanho máximo de 680 bits em transferências *downlink* e de 1000 bits em transferências *uplink* [32] .

Alcance (km)	Menos de 35
Bandas (MHz)	700 a 900 MHz(Licenciada)
Tamanho do Canal	200 kHz
Tamanho do Pacote	Váriavel
Taxa transferência de dados Uplink	204.8 kbps
Taxa transferência de dados Downlink	234.7 kbps
Topologia	Celular
Movimentação/Remoção de nós	Sim

Tabela 2.11: Características do protocolo NB-IoT [31].

A Tabela 2.11 resume as características da tecnologia NB-IoT.

2.4 Outras Soluções

Estas tecnologias permitem a criação de redes para dispositivos de baixo consumo, no entanto sofrem de problemas de alcance, o que os distingue das soluções LPWAN.

2.4.1 ZigBee

O ZigBee é um protocolo desenvolvido pela ZigBee Alliance que o usa como base o standard 802.15.4 criado pelo IEEE (*Institute of Electrical and Electronics Engineers*), mas que possui funcionalidades de *routing* e *networking*. A ZigBee Alliance é um conjunto de empresas que cooperam no desenvolvimento de protocolos de *networking*, com baixas taxas de transmissão e que podem ser usados para fins comerciais e industriais.

O 802.15.4 é um standard em que a comunicação é feita através de um canal de 5 MHz, entre as frequências 2.405 e 2.480 GHz. Na frequência de 2.4 GHz é especificado que as transmissões são limitadas a uma taxa de 250 kbps. Embora seja discriminado que um canal deva ter 5 MHz, apenas 2 MHz são ocupados. O standard define que a comunicação deve ser feita nas seguintes bandas ISM: de 868 a 868.8 MHz, de 902 a 928 MHz ou de 2.400 a 2.4835 GHz.

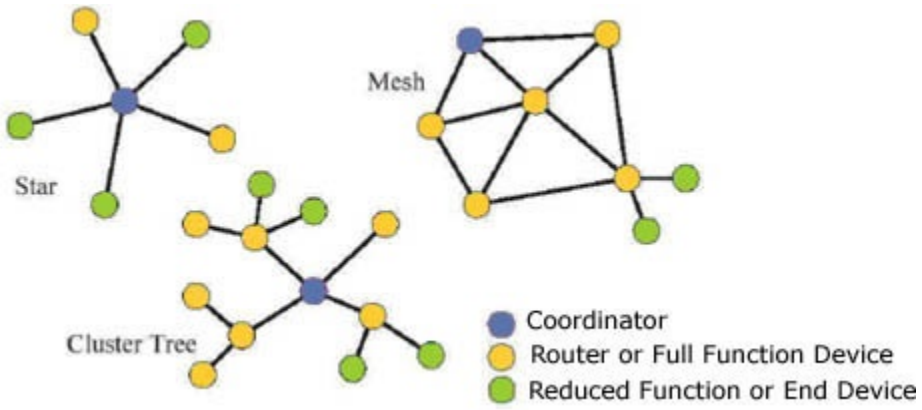


Figura 2.7: Possíveis topologias que a rede Zigbee pode tomar [34].

O ZigBee pode tomar as configurações de estrela, malha e árvore de *clusters*. No entanto é usado principalmente na configuração de malha. É uma configuração em que, caso seja impossível transmitir de um nó para outro nó por comunicação direta, a informação é passada para o nó seguinte até chegar ao nó terminal. A rede é capaz de ser criada de forma automática, sem a necessidade de intervenção de um utilizador. O protocolo faz todos os processos de *routing* de forma automática e, caso um nó seja removido, um novo conjunto de caminhos é criado. Além disso, todos os dispositivos terminais podem entrar em modo sleep, poupando energia [35]. A principal vantagem desta configuração é que caso haja perda de conectividade entre dois nós, o sistema consegue criar um caminho de comunicação por outro nó. No entanto, com o aumento do número de nós haverá um aumento da complexidade da rede gerada e cada nó terá de processar mais informação [9].

A rede é constituída por 3 tipos de nós. Como se pode ver na Figura 2.8, a rede é constituída por um nó coordenador (C), nós *router* (R) e por dispositivos terminais (E) [36]:

- O nó coordenador é responsável por criar a PAN (Personal Area Network), rede a que todos os outros dispositivos se vão ligar. Além disso, gere todos os dispositivos (R e E) que desejem ligar-se à rede. Não pode entrar em modo sleep.
- O nó *router* precisa de estabelecer-se na rede para depois poder adicionar até 10 nós R e E à infraestrutura. Não pode entrar em modo sleep.
- Os dispositivos terminais não permitem que outros dispositivos se conectem à PAN por eles, nem ajudam no routing de informação na rede. No entanto podem adormecer de forma a reduzir o gasto de energia.

As suas principais caraterísticas encontram-se resumidas na Tabela 2.12.

Requisitos de utilização de ZigBee

Cada nó precisa de um módulo ZigBee configurado conforme o tipo de nó. Um dos módulos mais usados é o XBee. O XBee é caracterizado pela sua fácil integração em redes Zigbee. Num

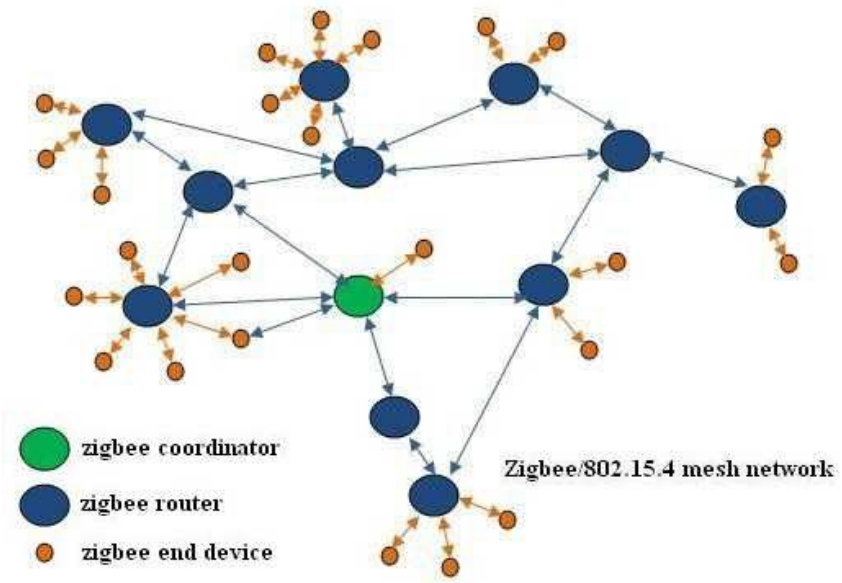


Figura 2.8: Exemplo de uma rede ZigBee [36].

Alcance (km)	10-100 m (em linha reta)
Bandas (MHz)	De 868 a 868.8 MHz; De 902 a 928 MHz; De 2.400 a 2.4835 GHz (ISM)
Tamanho do Canal	5 MHz
Tamanho do Pacote	Variável [37]
Taxa transferência de dados Uplink	250 kbps
Taxa transferência de dados Downlink	250 kbps
Topologia	Malha, mas pode tomar Estrela e Árvore de Clusters
Movimentação/Remoção de nós	Sim

Tabela 2.12: Características do protocolo ZigBee [6].

caso em que haja dois módulos ao alcance um do outro, eles automaticamente sincronizam sem necessidade de configuração à priori. A comunicação entre sensor-módulo e micro-controlador-módulo é feito através das portas series [39].

2.4.2 802.11ac

Em 1999 foi formada a Wi-Fi Alliance. Esta associação comercial é constituída por várias empresas, como a Apple, Sony, Microsoft e Texas Instruments [40] e é responsável pela *trademark* de Wi-Fi. A última versão de Wi-Fi chama-se 802.11ac. Esta versão é retrocompatível com todas as versões anteriores, inclusive a primeira. A 802.11ac foi desenvolvida com a intenção de ter taxas de transmissão wireless na ordem dos Gbps e ser sobretudo usada em aplicações WLAN (*Wireless Local Area Network*). A WLAN é uma rede local capaz de criar uma ligação à Internet através de ondas de rádio [41].

As versões anteriores do standard foram desenhadas para realizarem operações de um único utilizador. Contudo, na versão 802.11ac foram introduzidas capacidades MIMO (*Multiple Input Multiple Output*) e também a tecnologia *beamforming*, em que a informação é enviada apenas numa direção predefinida, em vez de *broadcast*. Ao contrário da 802.11n, a versão anterior à

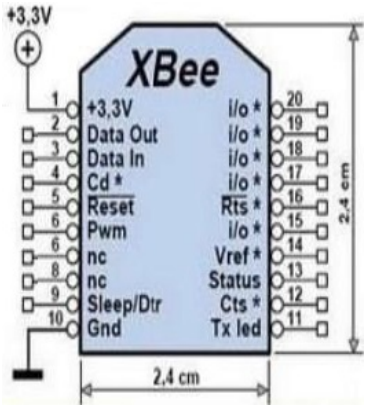


Figura 2.9: Esquemáticos de um módulo recetor XBee [38].

atual, que podia funcionar na banda 2.4 GHz, todos os produtos que usam 802.11ac só funcionam na banda de 5 GHz.

Alcance (km)	80 m com 3 antenas
Bandas (MHz)	5 GHz
Tamanho do Canal	20MHz,40MHz,80MHz,160MHz
Tamanho do Pacote	Variável [42]
Taxa transferência de dados Uplink	Dependente do número de antenas e da banda em uso
Taxa transferência de dados Downlink	Dependente do número de antenas e da banda em uso
Topologia	WLAN
Movimentação/Remoção de nós	Sim

Tabela 2.13: Características do protocolo 802.11ac.

As principais de 802.11ac caraterísticas encontram-se resumidas na Tabela 2.13.

Requisitos de utilização de 802.11ac

Uma rede 802.11ac necessita de um *router* que suporte esta tecnologia para servir como ponto de acesso, numa certa área. Para um dispositivo ligar-se à rede precisa de uma antena Wi-Fi ou de uma entrada de ethernet e as devidas permissões de acesso.

2.5 Sistemas de Baixo Consumo

Diogo Curto na sua dissertação de mestrado, realizada em 2015 [43], concluiu que para minimizar o consumo de energia em micro controladores é necessário estabelecer um equilíbrio entre:

- a velocidade de relógio, isto é, funcionar a uma frequência baixa limita o número de instruções que se podem realizar num certo período de tempo e, consequentemente, pode haver um gasto mais elevado de energia na realização do código do que se o micro controlador estivesse a uma frequência mais alta,

- a alimentação que irá limitar a velocidade de funcionamento e os periféricos disponíveis em diferentes modos de funcionamento,

- o tempo que demora a sair dos modos de baixa energia para níveis mais consumidores, também está dependente da velocidade de relógio,

- a gestão das memórias internas do micro controlador, uma vez que existe perda de dados da SRAM quando o micro controlador entra num modo de consumo mais baixo, e

- os periféricos de comunicação que, consoante a quantidade de dados a enviar e o tempo de envio, vão consumir energia de forma diferente.

A organização conhecida por *Embedded Microprocessor Benchmark Consortium* (EEMBC), criada em 1997, estabeleceu um standard de consumo de energia conhecido por ULPMark (*Ultra Low Power Mark*) que permite testar as várias características previamente descritas e estabelecer um valor indicativo das características de baixo consumo do micro controlador em relação as características de outros [44, 45].

No site da EEMBC [46] é feita a exploração do standard criado pela mesma para vários micro controladores.

2.6 Análise Comparativa das Soluções Apresentadas

O Sigfox é um protocolo de comunicação que usa modulação UNB e transmite nas bandas ISM Sub-GHz. A topologia de rede que este protocolo toma é a estrela. Este protocolo permite enviar informação *uplink*, limitado a 144 mensagens por dia, podendo também fazer transmissões *downlink*, limitada a 4 por dia. Devido ao favorecimento por transmissões *uplink*, os dispositivos terminais só terão de estar ativos no envio de mensagens para a estação base e não perderão energia na espera de mensagens da estação base. Para adicionar o dispositivo à rede é necessário uma subscrição online do serviço Sigfox, na qual é adicionado o número ID e PAC do dispositivo. Caso seja necessário a remoção do dispositivo e a eventual reinstalação na rede é necessário gerar um novo número PAC. O Sigfox é uma tecnologia que produz melhores resultados em termos de energia-alcance, relativamente às outras tecnologias, como se pode ver na Tabela 2.14. No entanto é limitada na quantidade de informação que consegue enviar.

A class A de LoRaWAN é um protocolo de transmissão bidirecional para sistemas que procuram minimizar os gastos de energia de dispositivos terminais. Esta tecnologia transmite nas bandas ISM Sub-GHz. A topologia de rede que este protocolo toma é estrela. Todas as comunicações são iniciadas pelo dispositivo terminal através do envio de uma mensagem, sendo aberto um canal onde o dispositivo e a estação trocam mensagens curtas. O gasto de energia do dispositivo terminal vai depender da quantidade de tempo que o canal vai estar aberto, o que é influenciado pela quantidade de informação a enviar. Para um dispositivo ser integrado numa rede LoRa é necessário que este seja compatível com módulos produzidos pela Semtech destinados a esse fim. A ligação é feita automaticamente caso o dispositivo seja aceite na rede.

O Weightless-N é o melhor em termos energéticos dentro das propostas da tecnologia Weightless, uma vez que não há transferências *uplink* e a taxa de transmissão não é elevada. O Weightless-P também é uma boa opção embora o limite inferior da taxa de transmissão seja o dobro do Weightless-N. O Weightless-W funciona nas bandas *TV White Space* e por isso não é considerado na dissertação. Todas os protocolos Weightless usam a topologia estrela e permitem a introdução e a remoção de nós. O Weightless-N e o Weightless-P transmitem nas bandas ISM Sub-GHz.

O GPRS é uma solução que usa os sistemas GSM existentes para comunicar. Tirando isso não apresenta nenhuma vantagem em relação às outras soluções.

O LTE-M trabalha nas bandas de frequência usadas pelos telemóveis. Este protocolo pretende criar redes a partir de torres telefónicas já existentes. A comunicação é iniciada pelos dispositivos terminais através do envio de informação. O dispositivo terminal só é alcançável no período de tempo subsequente à primeira transmissão. Qualquer dispositivo consegue aderir à rede LTE-M desde que possua um módulo para o efeito e a área tenha cobertura.

O NB-IoT possui os mesmo defeitos que o LTE-M uma vez que usa a mesma infraestrutura. No entanto é uma solução melhor em termos de consumo que o LTE-M, porque procurou limitar as funcionalidades apenas ao essencial.

O Zigbee é um protocolo de transmissão usado principalmente em topologia de malha. Esta tecnologia transmite nas bandas ISM Sub-GHz e em 2.4 GHz. A comunicação é começada pelos dispositivos terminais e, caso a transmissão tenha sido concluída, este pode entrar num modo sleep, destinado a diminuir o gasto de energia. Para adicionar um dispositivo na rede é necessário um módulo ZigBee. Caso dois sistemas Zigbee estejam em proximidade um do outro, a comunicação entre eles começa de forma automática.

O 802.11ac é um protocolo de transmissão criado para aplicações WLAN que procura ter taxas de transmissão na ordem dos Gbps. Esta taxa de transmissão é elevada para a quantidade de informação que se pretende enviar e, consequentemente, irá causar um gasto de energia maior do lado do dispositivo terminal, em comparação com a que ele gastaria se enviasse a frequências mais baixas. Qualquer dispositivo que possua uma antena Wi-Fi e as devidas permissões de acesso pode ser introduzido na rede.

	Alcance	Bandas	Taxa Trans. Uplink	Topologia	Adicionar /Tirar nós	Tipo de Rede	Dependência da estrutura celular
Sigfox	30-50 km (rural); 2-5 (urbano)	Sub-GHz ISM	100 bps	Estrela	Sim	Licenciado	—
LoRa	15 km (rural); 2-5 (urbano)	Sub-GHz ISM	1300 bps - 50 kbps	Estrela	Sim	Licenciado	—
-N	3 km (urbano)	Sub-GHz ISM	100 bps	Estrela	Sim	Licenciado	—
-P	2 km (urbano)	Sub-GHz ISM	200 bps - 100 kbps	Estrela	Sim	Licenciado	—
GPRS	Sem Informação	890 a 915 MHz 935 a 960 MHz	171.2 kbps	Celular	Sim	Licenciado	Dependente
LTE-M	2.5-5 km	Sub-GHz Licenciadas	1 Mbps	Celular	Sim	Licenciado	Dependente
NB-IoT	Menos de 35 km	700 a 900 MHz	204.8 kbps	Celular	Sim	Licenciado	Dependente
ZigBee	10-100 m	Sub-GHz ISM e 2.4 GHz	250 kbps	Malha	Sim	Ad-hoc	—
802.11ac	80 m (3 Antenas)	5 GHz	Variável	WLAN	Sim	Licenciado	—

Tabela 2.14: Características resumidas dos principais protocolos [6].

Da tabela 2.14 observa-se que o Sigfox tende a ser a melhor tecnologia em termos de alcance em relação às outras. Isto deve-se sobretudo ao tipo de modulação que é usado nesta tecnologia [47].

O Sigfox e o Weightless-N têm a menor taxa de transferência de dados *uplink* de entre todas as tecnologias (100 bps). Contudo, o Sigfox está limitado a um máximo de 140 mensagens por dia. O Weightless-P e o LoRa partilham de um intervalo de taxas de transferência muito próximos um do outro. No entanto a taxa mínima de transferência da LoRa (300 bps) é metade da Weightless-P (200 bps). ZigBee, LTE-M, GPRS, NB-IoT e 802.11ac transmitem com taxas na ordem dos kbps até Mbps.

Todos os protocolos permitem adicionar ou remover nós e, tirando o 802.11ac e as redes celulares, podem tomar a topologia de estrela.

2.7 Conclusões do Capítulo

Neste capítulo estudaram-se as principais características de algumas tecnologias LPWAN com o objetivo de escolher-se de entre elas as mais apropriadas a utilizar numa rede de dispositivos de baixo consumo e de grande dimensão. Verificou-se que há uma grade variedade nas formas em que é feita a transmissão de dados, o que, consequentemente, faz com que as soluções tenham vantagens e desvantagens, umas em relação às outras. Como exemplo, o Sigfox é ideal para sistemas que não tenham de enviar muita informação e que estão espalhados em áreas extensas, enquanto que o LTE-M é melhor para sistemas que precisam de enviar informação mais consistentemente.

Conclui-se que na escolha de uma destas redes para a execução de um projeto é necessário determinar-se, em primeiro lugar, quais os parâmetros que o sistema terá e, só depois, será possível escolher-se a rede mais apropriada.

Capítulo 3

Descrição Geral do Sistema

Neste capítulo é feita uma descrição geral do sistema a nível de funcionamento dos vários componentes e da rotina que irá realizar, sendo que esta consiste num sistema de comunicação usado para a monitorização de contadores de água e para o controlo de válvulas.

3.1 Introdução

O sistema desenvolvido nesta dissertação é idealizado para um local em que existem vários contadores de água, como prédios, sendo capaz de receber e enviar informação adquirida e regular a passagem de água, através de 4 válvulas.

Pretende-se que o sistema tenha encargos de manutenção reduzidos, devendo conseguir funcionar num período mínimo de 10 anos com apenas uma bateria.

Para permitir versatilidade de opções em termos de comunicação, o sistema foi desenhado para ser compatível com 4 módulos de comunicação diferentes, sendo montado no sistema um de cada vez. Para garantir a integridade da informação enviada, cada mensagem é encriptada com uma chave conhecida pelo sistema e pelo servidor.

Para permitir o *update* do *firmware* existe uma memória externa ao micro controlador, na qual há uma versão estável do código e a última, que pode ser reescrita pelo micro controlador quando uma versão nova é introduzida. Estas duas versões são necessárias para que, caso haja problemas na última versão, o micro controlador tenha uma versão segura em que trabalhar.

3.2 Modelo de trabalho do Sistema

O sistema é constituído por um micro controlador que observa e controla até quatro conjuntos de contadores de água (Cyble Sensor V2) e de válvulas (CWX-15q cr05).

O contador pode tomar as duas configurações observadas na Figura 3.2. Neste projeto é usado a configuração com 5 fios. O contador indica a quantidade de água a ser entregue (sinal HF), a sua direção (sinal DIR) e se houve corte de cabo (sinal *Cable Cut Detection*), sendo representado pelos 3 sinais conectados ao contador (Figura 3.1).

Cada contador tem associado duas variáveis. Uma variável que conta sempre para cima e outra que conta para cima e para baixo, consoante o valor de DIR, para permitir determinar se houve alguém a tentar sabotar a distribuição da água e a quantidade desviada.

Pretende-se manter a informação do consumo de água por hora de cada contador até ao máximo de 3 dias. Para tal, usa-se um *buffer* circular no formato de *First In First Out* (FIFO), como se pode ver na Figura 3.3, em que cada parcela é a contagem de água entre duas horas

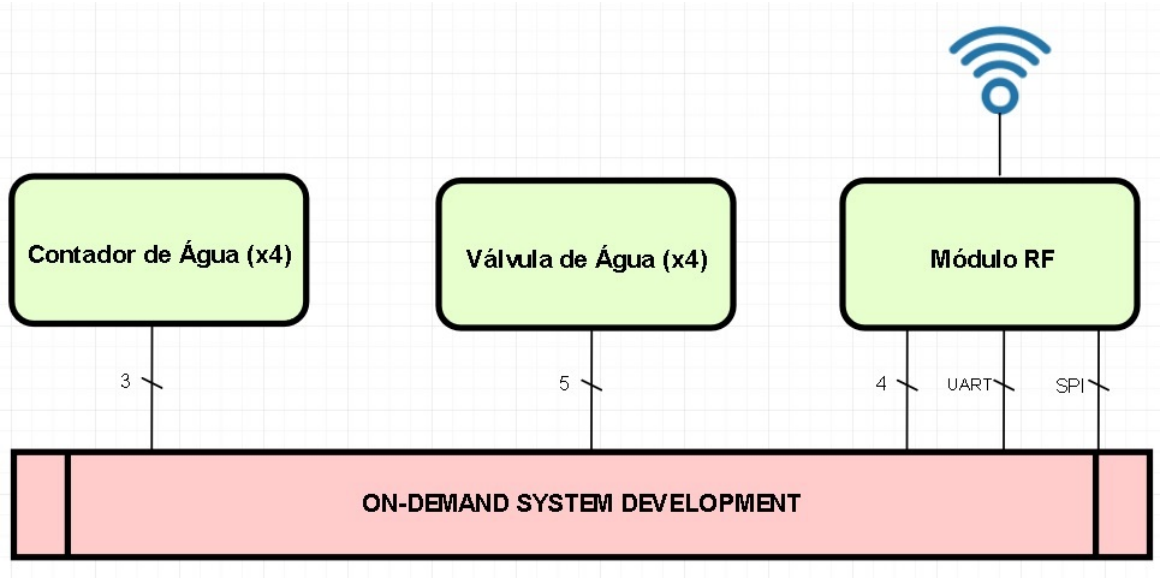


Figura 3.1: Modelo de conexões do sistema.

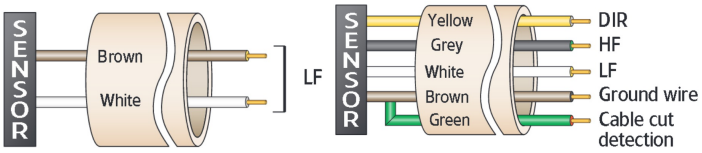


Figura 3.2: Configuração com 2 e 5 fios do sensor de contagem de água [48, 49].

imediatas. Após cada transmissão o *buffer* é completamente "limpo", sendo apenas guardado mais que um dia de informação quando houver complicações na comunicação.

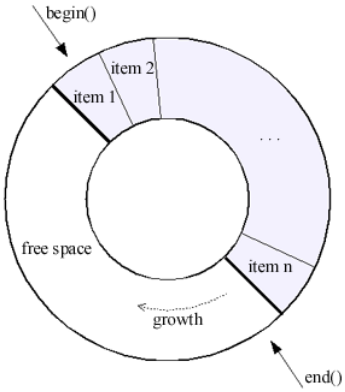


Figura 3.3: Buffer Circular FIFO.

A válvula controla o fluxo de água. Esta atuação é representada na Figura 3.4 através dos 5 contactos, dos quais 3 correspondem a 2 contactos secos com ponto comum dos fins-de-curso do movimento da válvula ("Fully open" e "Fully closed") e 2 correspondem às ligações de controlo do motor de abertura e fecho da válvula, como se pode ver na Figura 3.4. Prevê-se que o motor entre em função duas vezes por mês.

Além disto, o micro controlador transmite periodicamente a informação obtida pelos sensores para uma rede LPWAN, através de um módulo de comunicação. Pretende-se a capacidade de

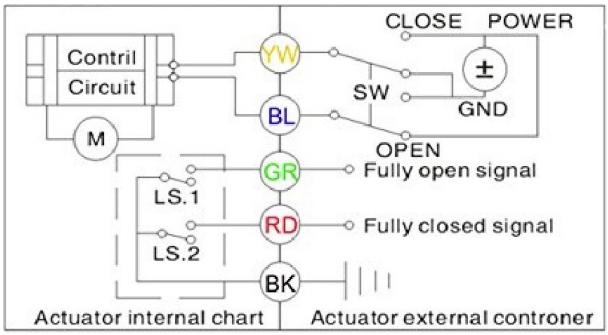


Figura 3.4: Esquema da válvula [50].

integração em *firmware* de até 4 módulos de comunicação, estando apenas um montado no sistema. Cada módulo acede a um tipo de rede diferente: LoRa, NB-IoT, GSM/GPRS ou Wi-Fi. Caso não haja nenhum módulo RF colocado no sistema, o micro controlador continua periodicamente a tentar comunicar como se existisse um módulo colocado. Na Figura 3.1 dois dos sinais são para identificação do módulo, um serve para ligar e desligar o módulo colocado no sistema e, outro ainda, serve como terra para o módulo.

A periodicidade de cada transmissão é gerida por um *Real Time Clock* (RTC) interno ao micro controlador. Para garantir a correta hora do relógio interno, o servidor, após cada transmissão, envia a hora local correspondente. O ciclo de funcionamento é caracterizado por 3 valores: o tempo de indicação de começo do ciclo, que serve como início e fim do ciclo, o tempo em que o micro controlador está acordado a enviar e o tempo em que está a dormir. É possível programar mais que uma transmissão por dia, consoante os valores que são atribuídos aos tempos de acordar e dormir, diminuindo o tempo de vida do sistema. O ciclo recomeça no tempo definido independentemente do tempo de dormir programado que falta para acordar, como se pode ver na Figura 3.5

Figura 3.5: Ciclo de funcionamento do RTC.

Para garantir a integridade da informação enviada, cada mensagem é codifica através do bloco de cifra Advanced Encryption Standard (AES) existente no micro controlador, em que a informação é encriptada e desencriptada com uma chave gerada a partir do *Identifier Number* (ID) de fabrico do micro controlador. Para o servidor conseguir descodificar as mensagens enviadas, uma de duas possibilidades deve ser escolhida: o servidor recebe a informação do ID internamente, ou seja, alguém que tenha conhecimento do ID introduz-lo no servidor ou, na primeira transmissão do micro controlador, a informação enviada não é encriptada e um dos valores entregues ao servidor é o ID.

Para haver a possibilidade de *update* e de manutenção do *firmware*, pretende-se que seja

possível programar o micro controlador através dos módulos RF e através dos pinos de *debugging*. Os dados recebidos são guardados numa memória externa (AT25SF041). Esta memória vai manter uma versão estável do código e a versão mais recente, que pode ser reescrita pelo micro controlador com a introdução de *updates*, como se vê na Figura 3.6. Prevê-se que o código receba *updates* de 6 em 6 meses.

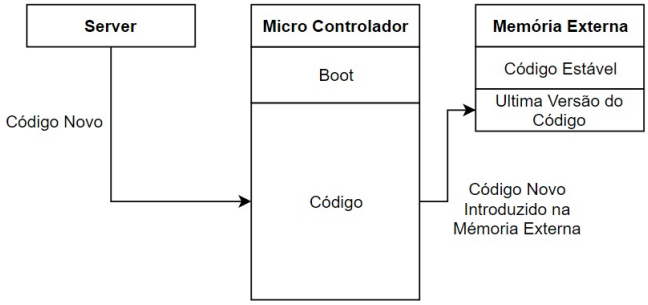


Figura 3.6: Diagrama de update do código.

O *update* da última versão não é feito numa única transmissão, ou seja, o código é enviado em partes nos momentos pré-programados de transmissão para não ser necessário estender o tempo de transmissão previsto do sistema. A última versão do código só entra em funcionamento quando recebe ordem do servidor, como se pode ver na Figura 3.7.

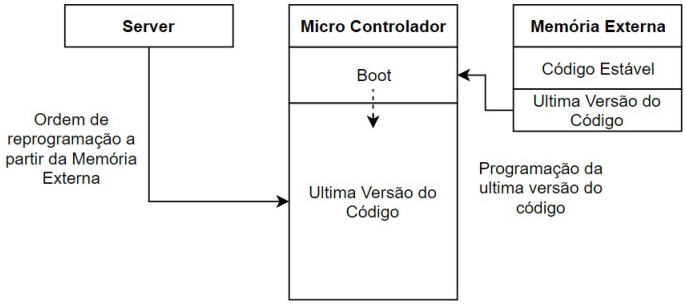


Figura 3.7: Diagrama de boot da última versão do código.

Caso o algoritmo de *Cyclic Redundancy Check* (CRC), um algoritmo de deteção de erros no código, indique a inexistência de problemas na última versão do código, este é instalado, como se vê na Figura 3.8

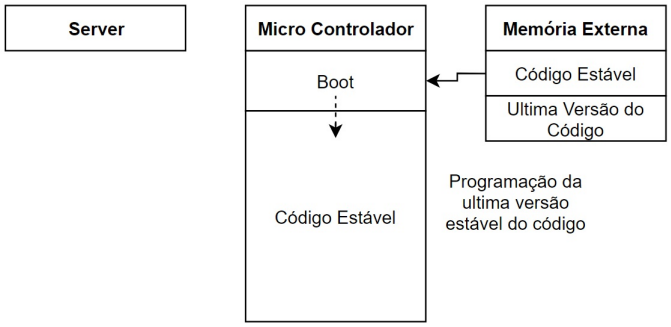


Figura 3.8: Diagrama de boot da versão estável.

O sistema deve ser capaz de funcionar durante um período de tempo longo, 10 anos, tendo

como única fonte de energia uma bateria (ER34615M). Para tal, ele é capaz de se colocar em modo de funcionamento de baixo consumo quando é necessário executar instruções. Estes momentos incluem a contagem de impulsos e corte de cabo. Quando está em *standby*, ele coloca-se em modo *sleep* de baixo consumo até ser necessário executar algum processo.

Por fim, para garantir a integridade do sistema, o micro controlador vai observar o estado de abertura da caixa através de um sensor de abertura, que é constituído por uma alavanca que quando perde pressão, isto é, quando a caixa é aberta, muda o estado lógico do seu pino de saída, enviando uma notificação através do módulo RF. Além disso, também são usadas pequenas resistências (33 *ohm*) nas entradas dos fios dos dispositivos de contagem para haver uma regulação da corrente, de forma a proteger o micro controlador de ligações inesperadas.

3.3 Diagrama de Entradas e Saídas do Sistema

No diagrama seguinte apresenta-se o diagrama de blocos do controlador, discriminando os componentes mais relevantes para o desenvolvimento de *hardware*.

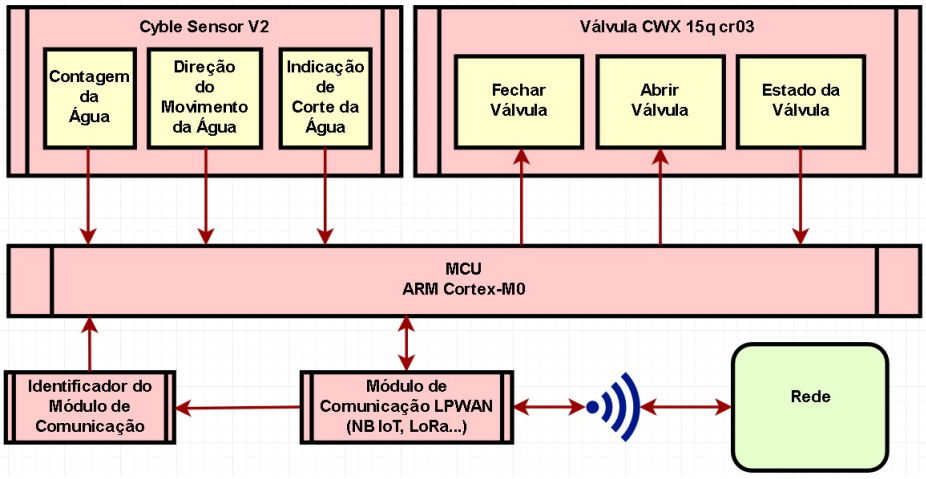


Figura 3.9: Diagrama de Entradas e Saídas do Micro Controlador.

O controlador deverá ler e atuar no conjunto de entradas e saídas da tabela 3.1.

3.4 Arquitetura Lógica do Sistema

Na Figura 3.10 é apresentado o diagrama lógico do sistema. No Apêndice B encontram-se todas as funções descritas e um breve resumo de cada uma.

3.4.1 Modbus

O Modbus é um protocolo de comunicação de alto nível que recorre a outros protocolos (e.g. rs-485, ethernet) a nível de camada física e de ligação. Ele permite a troca de informação entre dispositivos alocados a diferentes redes e barramentos, devido ao formato fixo de códigos e pacotes trocados entre eles. Este protocolo é implementado principalmente em comunicação TCP/IP através de cabos de *ethernet* e em dispositivos que usam comunicação série dessincronizada, mas pode ser implementado em qualquer outro meio de comunicação. Os códigos Modbus pré-estabelecidos encontram-se na tabela 3.2.

O cliente é definido como sendo o dispositivo que começa a comunicação Modbus e é responsável pela criação do pacote enviado ao servidor. O pacote possui os seguintes campos:

Tipo	Designação	Descrição
Entrada	Contador de Água Direção (Fio Amarelo; Ver Fig. 3.2))	Entrada digital para determinar a direção da água no cano (0 - Para a frente; 1 - Para trás).
	Contador de Água HF (Fio Cinzento; Ver Fig. 3.2))	Entrada digital para determinar a quantidade de água a ser transmitida. Um impulso para a válvula considerada representa x litros de água.
	Contador de Água Deteção de corte (Fio Verde; Ver Fig. 3.2))	Entrada digital para determinar se houve corte de cabo (0 - Esta a funcionar corretamente; 1 - Houve um corte).
	Válvula de Água Sinal para verificar válvula aberta (Fio Verde; Ver fig. 3.4))	Entrada digital para identificação de completamente aberto.
	Válvula de Água Sinal para verificar válvula fechada (Fio Vermelho; Ver fig. 3.4))	Entrada digital para identificação de completamente fechado.
	Módulo de Comunicação Sinal para indicar o módulo de comunicação conectado	2 entradas digitais para a identificação do módulo.
	Switch de abertura de caixa Sinal para indicar a caixa foi aberta	1 entrada digital.
Saída	Válvula de Água Sinal para Abrir (Fio Amarelo; Ver fig. 3.4)))	Saída digital para controlo da válvula (0 - Inativo; 1 - Abre a válvula).
	Válvula de Água Sinal para Fechar (Fio Azul; Ver fig. 3.4)))	Saída digital para controlo da válvula (0 - Inativo; 1 - Fecha a válvula).
	Módulo de Comunicação Sinal para ligar/desligar o módulo de comunicação	Saída digital para controlo do módulo selecionado (ON/OFF, SEL1, SEL2).
Comunicação	Módulo de Comunicação SPI	3 pinos; 2 para comunicação e 1 para o relógio (MISO, MOSI, SCK).
	Módulo de Comunicação Uart	2 pinos para comunicação UART. (Tx, Rx)

Tabela 3.1: Entradas e Saídas do Micro Controlador.

- *Additional address* - identifica o endereço do servidor que vai atender o pacote. Este campo pode tomar os valores entre 1 a 247;
- *Function code* - é o código hexadecimal que indica ao servidor a tarefa a realizar. Os códigos disponíveis podem ser observados na Figura 3.12. O *Function code* "0" não é válido. Quando há um erro o servidor retorna o código da função mais 0x80;
- *Data* - é dependente do código introduzido. Este parâmetro pode tomar o tamanho de 0.
- *Error Check* - identifica a existência de um ou mais erros na informação enviada pelo cliente. Isto é feito através de códigos de erro predefinidos (os *Exception Code*), como se pode ver na Tabela 3.2.

O último aspeto importante de Modbus é o modo como a informação é estruturada para ser enviada. Consoante o pedido que o código faz, o dispositivo pode devolver a informação existente de uma de quatro tabelas. Essas tabelas estão divididas em *Discretes Input*, *Coils*, *Input Registers* e *Holding Registers*. Todas as tabelas podem ser lidas mas só a tabela *Coils* e a *Holding Registers* podem ser escritas pelo servidor.

As tabelas são usadas para permitir ao cliente aceder a informação interna do sistema, como dados recolhidos, ou alterar/reprogramar o *firmware* interno do sistema.

Funcionalidades da Tabela Modbus

Na secção H encontra-se a tabela modbus, na qual vai atuar o micro controlador e o servidor através dos comandos descritos na secção 3.4.1.

As funcionalidades desenvolvidas incluem:

- a leitura da identificação do dispositivo e do tipo,
- a leitura da versão do firmware,

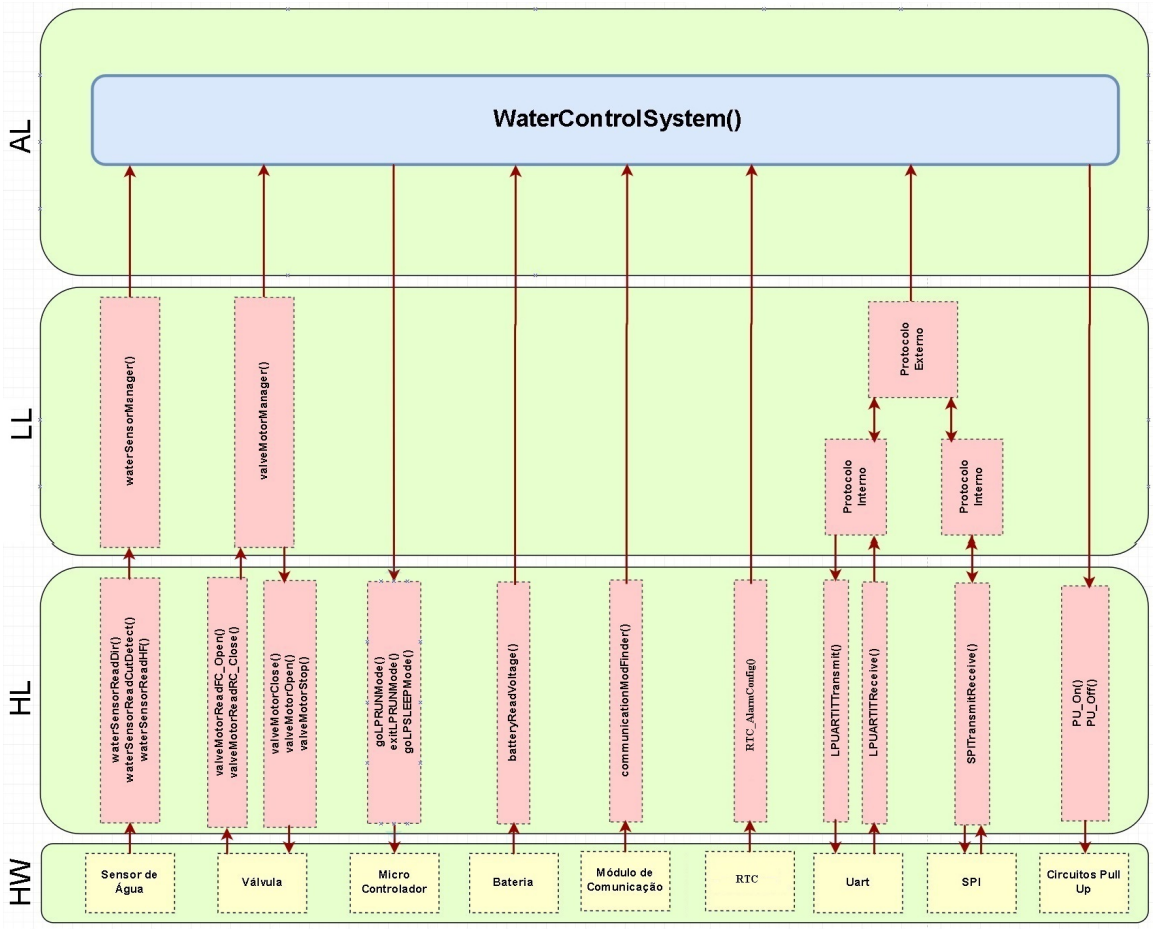


Figura 3.10: Diagrama do Sistema Lógico que inclui o HW (Hardware), o HL (Hidden Layer), o LL (Logic Layer) e o AL (Application Layer).

- a identificação do módulo de comunicação instalado,
- a leitura do tempo em que esteve colocado em trabalho,
- a programação do ciclo de emissão de dados,
- a identificação dos contadores e dos motores,
- a leitura da contagem dos impulsos lidos,
- a leitura dos estados de direção e do corte de cabo dos contadores,
- a leitura dos estados dos sensores de fim de curso das válvulas,
- o controlo dos motores,
- a informação da quantidade de água consumida por hora, mantendo em memória até um máximo de 3 dias,
- a leitura de um vetor de erros,
- programação de uma versão nova do firmware,

				Function Codes		
				code	*Sub code*	*(hex)*
Data Access	**Bit access**	Physical Discrete Inputs	Read Discrete Inputs	**02**		**02**
		Internal Bits Or Physical coils	Read Coils	**01**		**01**
			Write Single Coil	**05**		**05**
			Write Multiple Coils	**15**		**0F**
	16 bits access	Physical Input Registers	Read Input Register	**04**		**04**
		Internal Registers Or Physical Output Registers	Read Holding Registers	**03**		**03**
			Write Single Register	**06**		**06**
			Write Multiple Registers	**16**		**10**
			Read/Write Multiple Registers	**23**		**17**
			Mask Write Register	**22**		**16**
			Read FIFO queue	**24**		**18**
	File record access		Read File record	**20**		**14**
			Write File record	**21**		**15**
Diagnostics			Read Exception status	**07**		**07**
			Diagnostic	**08**	**00-18,20**	**08**
			Get Com event counter	**11**		**OB**
			Get Com Event Log	**12**		**0C**
			Report Slave ID	**17**		**11**
			Read device Identification	**43**	**14**	**2B**
Other			Encapsulated Interface Transport	**43**	**13,14**	**2B**

Tabela 3.2: Códigos Modbus [51].

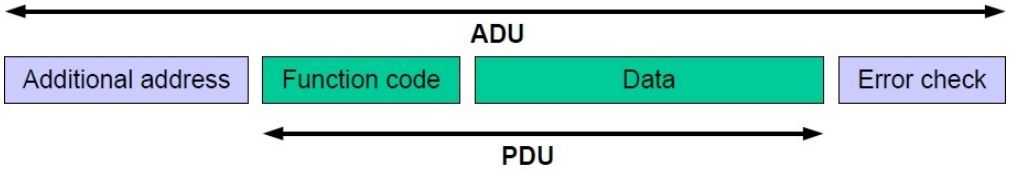


Figura 3.11: Formato de pacotes de protocolo Modbus [51].

- a leitura e escrita dos diferentes dados necessários à comunicação.

Através destas funcionalidades é possível identificar se houve algum problema no hardware. Por exemplo, no caso em que é necessário mandar executar a rotação da válvula e se não houver respostas dos sensores de fim de curso após várias tentativas, é possível deduzir que houve um problema com os motores.

3.5 Conclusões do Capítulo

Este capítulo procurou estabelecer o modelo de funcionamento do sistema, quais os sinais que são trocados entre o micro controlador e o componentes do sistemas e, por fim, as funções usadas no algoritmo final do sistema.

O protocolo Modbus foi também introduzido como sendo o formato lógico dos comandos e dados trocados entre o sistema e o cliente.

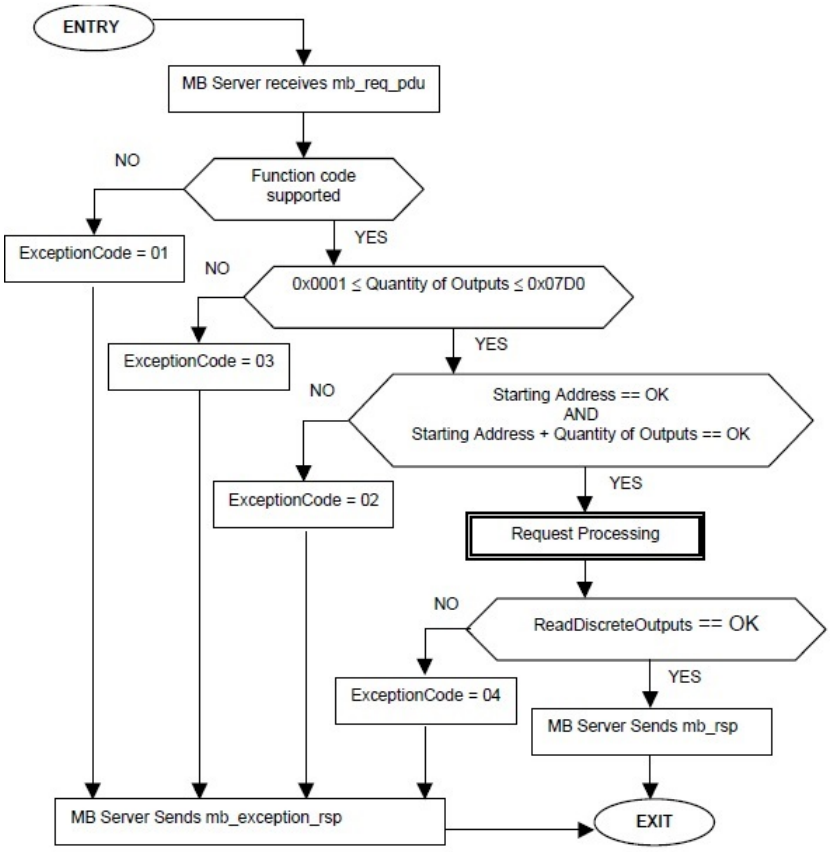


Figura 3.12: Diagrama de estados da função Modbus Read Coils [51].

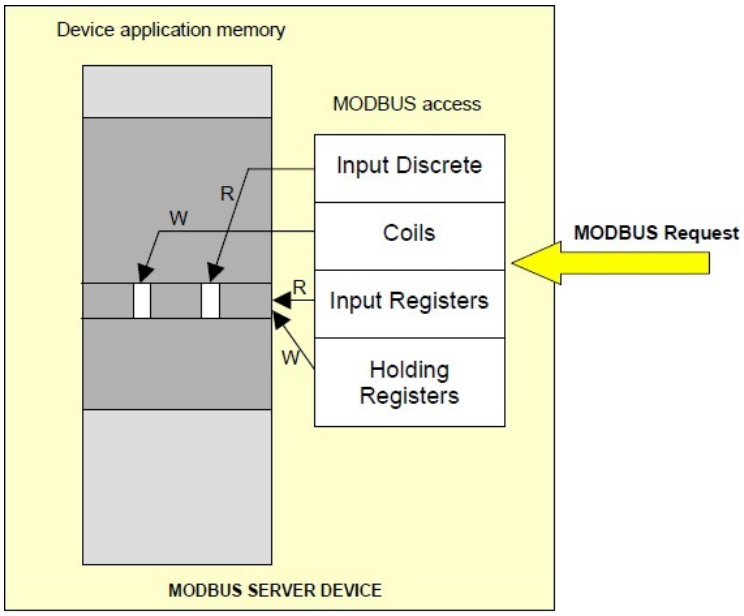


Figura 3.13: Modelo de dados Modbus para um bloco único de memória [51].

Capítulo 4

Implementação do projeto

Nesta secção é feita una descrição de todo os componentes, circuitos e PCBs usados para a criação do sistema descrito no capítulo anterior. Também são apresentados os resultados e os problemas encontrados no desenvolvimento do projeto.

4.1 Descrição do Hardware do Sistema

4.1.1 Características do Micro Controlador STM32L053R8

Características de Baixo Consumo do micro controlador [52]:

- alimentação entre 1.65 V e 3.6 V,
- suporta temperaturas entre -40 e 125 ° C,
- 0.27 μA Standby mode (2 pinos para acordar),
- 0.4 μA Stop mode (16 pinos para acordar),
- 0.8 μA Stop mode + RTC + 8 KB RAM retenção,
- 88 μA/MHz em Run mode,
- 3.5 μs tempo para acordar (da RAM),
- 5 μs tempo para acordar (da Flash memory).

Características do ARM 32-bit Cortex-M0+ com MPU:

- Desde 32 kHz até 32 MHz de frequência máxima, e
- 0.95 DMIPS/MHz.

O micro controlador possui os periféricos POR(*Power On Reset*) e BOR (*Brownout Reset*) com 5 níveis de tensão programáveis para garantir que não há falhas na execução de cada instrução.

Da Tabela 4.1, o modo *Run* é definido como tendo o relógio a *High-speed Clock* (HCLK), podendo usar qualquer periférico e executar código. Em modo *Sleep*, o núcleo CPU é desligado, mantém os estados dos pinos I/O, os periféricos mantêm o seu funcionamento e a frequência de relógio pode ser regulada. Em relação aos modos *Low Power Run* e *Low Power Sleep*, a única diferença é que o CPU corre num e está desligado no outro, respetivamente. Quando há transição de modo, os estados dos pinos I/O mantêm-se e todos os periféricos estão disponíveis.

Modos de baixo consumo	Tempo de acordar	Consumo de corrente
Run	-	555 μA (fHCLK = 4 MHz e código a correr da FLASH)
Sleep	0.36 μs	Entre 57.5 μA e 130 μA (para fHCLK = 1 MHz e código a correr da FLASH)
Low Power Run	3 μs	Entre 22 μA e 28 μA (para MSI clock = 65 kHz, fHCLK = 32 kHz e código a correr da FLASH)
Low Power Sleep	32 μs	Entre 17 μA e 23 μA (para MSI clock = 65 kHz, fHCLK = 32 kHz e código a correr da FLASH)

Tabela 4.1: Tempo que demora para mover de um estado para o modo Run e o consumo de corrente em cada modo.

4.1.2 Características dos Módulos RF

Nesta secção é abordado um conjunto de módulos de comunicação que foi previsto serem usados na comunicação do sistema. Os módulos, por ordem de aparição, correspondem ao seguinte conjunto de redes: LoRa, GPRS, NB-IoT e 802.11ac.

CMWX1ZZABZ-078

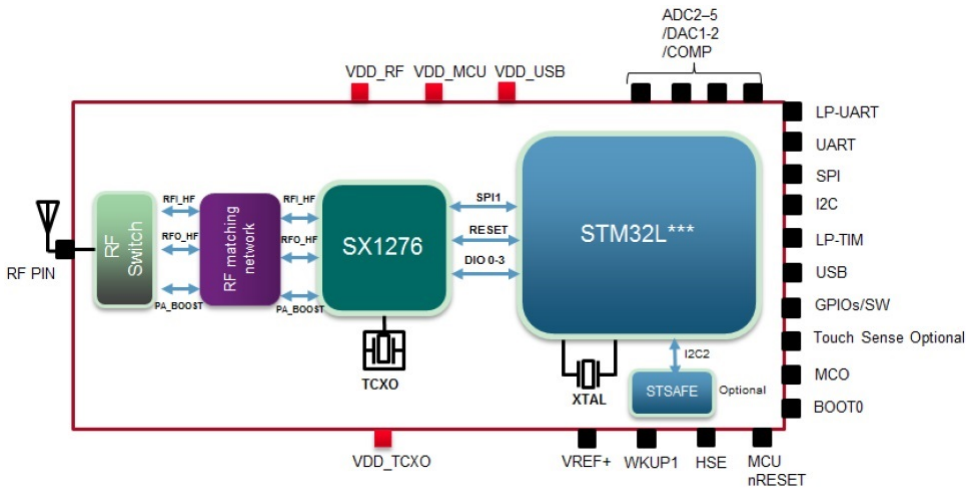


Figura 4.1: Modelo do chip CMWX1ZZABZ-078 [53].

CMWX1ZZABZ-078 é um módulo RF que incorpora o chip micro controlador STM32L082 da STMicroelectronics e o chip de comunicação LoRa SX1276, que trabalha nas frequências 868MHz, na Europa, e 915MHz, nos Estados Unidos. Este dispositivo tem a operação normal a tensões entre 3.0V e 3.6V com tensão máxima absoluta 3.9V e a temperaturas entre -40°C e +85°C. Além disso, o dispositivo tém uma potência nominal de +14 dBm, com a possibilidade

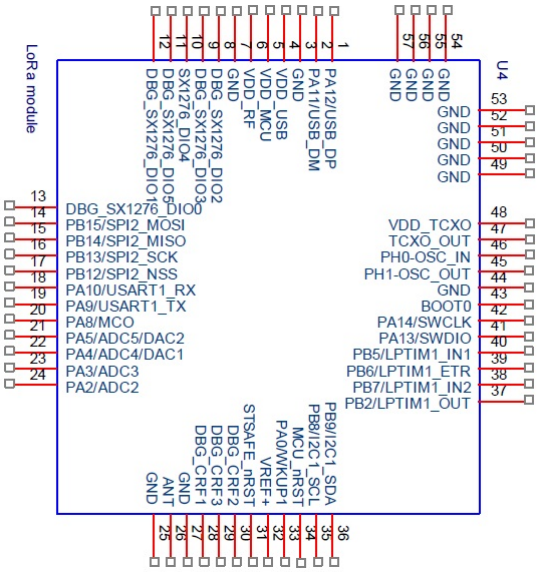


Figura 4.2: Chip ST interno do chip CMWX1ZZABZ-078.

---

de +20 dBm através de um amplificador de potência. Este dispositivo consome 128mA na transmissão e 21.5mA na receção [53].

GL865 QUAD

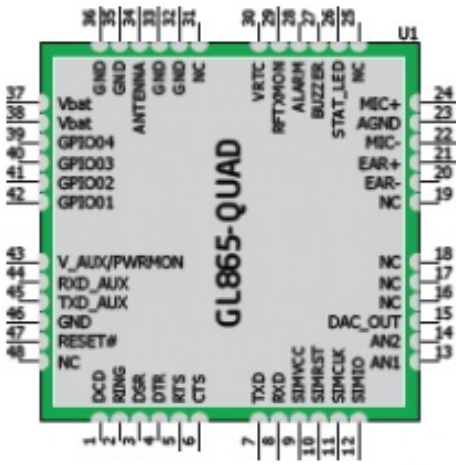


Figura 4.3: Chip GL865 QUAD [54].

GL865 QUAD é um módulo RF que usa o protocolo de rádio GPRS. É capaz de comunicar nas bandas ISM 850MHZ, 900MHZ, 1800MHZ e 1900MHZ para permitir as velocidades de comunicação de 2G. Este dispositivo tem *upload* de 40 kb/s e de *download* de 80 kb/s. Além disso, a operação normal ocorre entre as tensões de 3.4V e 4.2V, entre as temperaturas -40°C e +85°C.

A comunicação UART possui uma tensão interna de 2.8V, com uma operação máxima absoluta de 3.1V e uma *baud rate* variável entre 300 b/s e 115200 b/s [55].

Este dispositivo consome 200 mA quando está a comunicar na banda GSM900 PL5 e 140 mA quando está a comunicar na banda DCS1800 PL0 [56].

SARA-N2

Model	Region	Bands			Positioning			Interfaces					Features								Grade		
		3GPP Release Baseline	3GPP Category	NB-IoT Bands	GNSS via Modem	AssistNow Software	CellLocate®	UART	SPI	USB 2.0	GPIO	DDC for GNSS / I²C slave	Antenna Supervisor	Power saving mode	eDRX	Dee p sleep mode	Embedded UDP stack	FW update over AT (FOAT)	FW update over the air (FOTA)		Standard	Professional	Automotive
SARA-N200	Europe / APAC	13	NB1	8	□			2			□	□	□	•	•	•	•	•	•				
SARA-N201	APAC	13	NB1	5	□			2			□	□	□	•	•	•	•	•	•				
SARA-N210	Europe	13	NB1	20	□			2			□	□	□	•	•	•	•	•	•				
SARA-N211	Europe	13	NB1	8, 20	□			2			□	□	□	•	•	•	•	•	•				
SARA-N280	S. America / APAC	13	NB1	28	□			2			□	□	□	•	•	•	•	•	•				

□ = Available in future FW version

Tabela 4.2: Dispositivos Sara-N2 [57].

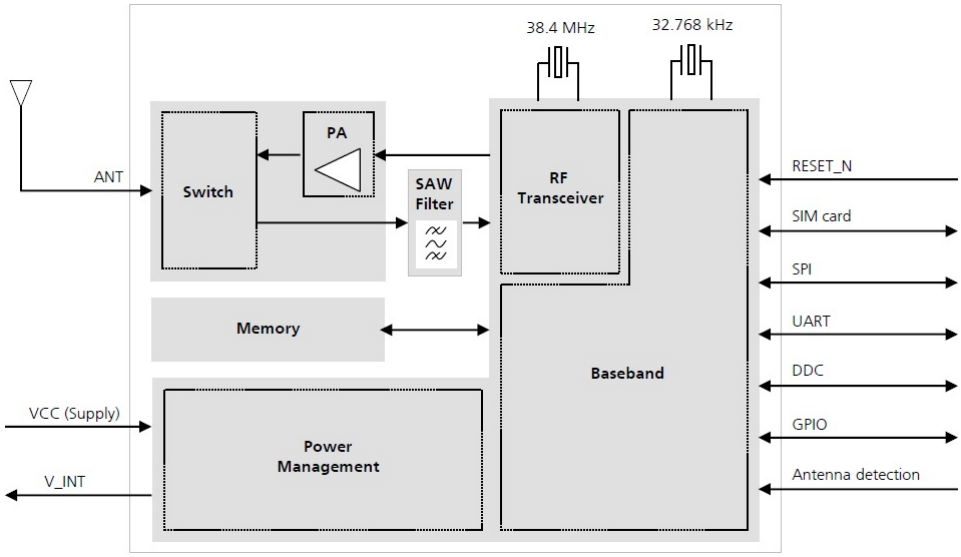


Figura 4.4: Diagrama de Blocos da série de dispositivos Sara-N2.

SARA-N2 é o módulo RF que utiliza o protocolo NB-IoT para comunicação. Como se pode observar na tabela 3, a série SARA-N2 possuem cobertura na Europa, América do Sul, Ásia e no Pacífico. Todos os dispositivos na série possuem a mesma estrutura de blocos e as mesmas características de funcionamento, sendo a única diferença entre eles a banda LTE em que a transmissão é realizada. Estes dispositivos têm uma potência nominal de +23 dBm, uma taxa de upload de 27.2 kb/s e uma taxa de download de 62.5 kB/s. Estes módulos podem ser alimentado entre 3.1V e 4.0V (tipicamente a 3.6V) e funcionam entre -40°C e +85°C.

Os módulos SARA-N2 possuem um modo *Deep-sleep* que os coloca a trabalhar com o relógio interno de frequência a 32 kHz, o que limita o consumo de corrente.

A comunicação serie possui duas UARTs. Uma trabalha entre 0V e VCC, com uma *baud rate* fixa de 9600 b/s, em que as mensagens têm o formato de 8N1 (8 bits, sem paridade, com 1 bit de stop). A segunda UART trabalha entre 0V e a tensão interna de 1.8V. Tem uma *baud rate* fixa a 921600 b/s com um formato de mensagem de 8N1.

Modo	Condição	Potência de Tx	Corrente
Deep-Sleep	Corrente média durante um período de 10s		5μA
Inativo	Corrente média durante um período de 10s		6mA
Rx	Corrente média durante um período de 10s		20mA
Tx	Corrente média durante um período de 2s	23dBm 13dBm 7dBm -3dBm	220mA 100mA 78mA 75mA

Tabela 4.3: Corrente consumida consoante os modos de funcionamento.

SPWF01SA

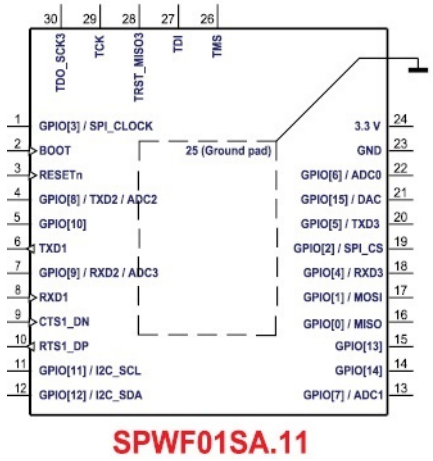


Figura 4.5: Chip SPWF01SA.

SPWF01SA é um módulo RF que incorpora o chip micro controlador STM32 ARM Cortex-M3 da STMicroelectronics e um transmissor-recetor 2.4 GHz IEEE 802.11 b/g/n. Este dispositivo pode ser alimentado no máximo entre -0.3V e 4.0V, trabalhando tipicamente a 3.3V e entre as temperaturas de -40°C e +85°C [58].

Os pinos usados na comunicação UART têm uma tensão máxima de 5V.

Para além do modo de funcionamento normal, este módulo tem as seguintes opções de funcionamento:

- *Run*, em que o micro controlador interno e o sistema de rádio estão prontos para transmitir e receber. Este modo consome tipicamente 70 μA [59].
- *Standby*, em que o micro controlador interno e o sistema de rádio são colocados num estado de *Standby*. Este modo consome tipicamente 43 μA.
- *Sleep*, em que o micro controlador interno é colocado em modo stop e o sistema de rádio é colocado em modo *Sleep*. Este modo consome tipicamente 15 μA.
- *Low power state*, em que, o micro controlador interno está em modo de funcionamento normal e o sistema de radio é colocado em modo *Sleep*. Este modo consome tipicamente 26 μA.

Este módulo consome 105 mA na receção de informação e 243 mA na transmissão de potência a 10 dBm.

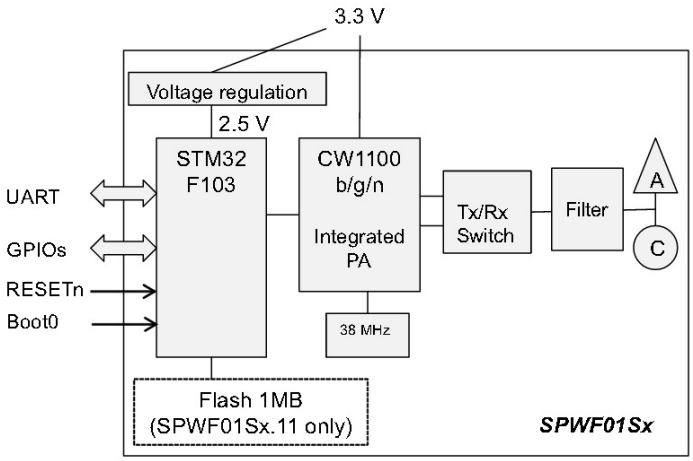


Figura 4.6: Modelo do sistema SPWF01SA.

4.1.3 Descrição dos Contadores de Água (Cyble Sensor V2)

O contador Cyble Sensor V2 é constituído por 5 fios, 4 de sinal e um ligado à massa. O fio amarelo permite determinar qual é a direção de fluxo de água (0V - vai para a frente; 3.6V - vai para a trás).

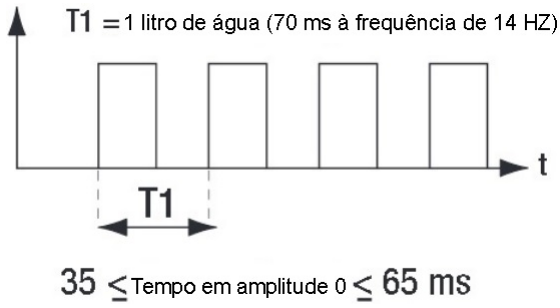


Figura 4.7: Sinal HF.

Os sinais HF e LF (LF = HF*k, com k = 1;2.5;10;25;100;1000) permitem a contagem de água através de impulsos. Estes dois sinais diferem em que HF conta a água indiferentemente do sentido enquanto que LF, quando a água aflui para trás, deixa de enviar impulsos. No sensor usado, k = 1, logo LF = HF e não é necessário ler o sinal LF. Esta ligação e a de deteção de corte precisam de um *pull up* permanente a 3.6V, uma vez que se pretende que qualquer sinal vindo de HF e de Corte de Cabo acorde o micro controlador para executar protocolo. O sinal de direção precisa de *pull up* permanente uma vez que se pretende acordar o micro controlador quando houver uma mudança do estado lógico. A quantidade de água a ser contada também é dependente do modelo do contador instalado. Isto quer dizer que um impulso de HF tanto pode representar 1 litro como pode representar 100 litros. É necessário ter em conta qual o modelo existente na localização que se quer instalar [48].

O contador não necessita de alimentação da bateria.

4.1.4 Descrição das Válvulas de Água (CWX-15q cr05)

Na válvula, os condutores amarelo e azul são a alimentação do motor, permitindo, consoante a polarização, movimentar a válvula para abrir/fechar. Para o bom funcionamento deste

movimento é necessário uma ponte H alimentada entre 3V a 6V. O processo de abrir ou fechar de um lado ao outro demora entre 3 a 5 segundos. Os fios verde e vermelho são sensores de fim de curso usados para identificar se a válvula foi aberta ou completamente fechada, respetivamente. Os fios de sensores de fim de curso necessitam de um circuito *pull up* para cada um. No entanto, os circuitos só precisam de estar ativos quando é necessário haver movimento. Um dos sinais observáveis na Figura 3.1 é usado para ligar/desligar a alimentação do circuito *pull up*. As válvulas dissipam até uma potência máxima de 2W. A 3.6V o motor consome aproximadamente 75 mA de corrente, dissipando até 0.27 W.

4.1.5 Circuito *Pull Up*

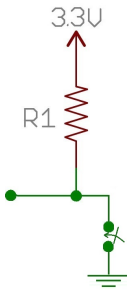


Figura 4.8: Circuito Pull Up.

O circuito *pull up* é usado para garantir que um sinal de um dispositivo externo esteja dentro de um conjunto de tensões definidas como níveis lógicos. Para garantir o menor gasto possível de energia da bateria alguns circuitos *pull up* do sistema podem ser ligados e desligados, sendo estes controlados pelo micro controlador. Alguns circuitos estão ligados permanentemente, uma vez que são usados para acordar o micro controlador quando necessário. Os circuitos que usam *pull ups* e o estado de permanência encoontram-se resumidos na Tabela 4.4.

Pull Up	**Estado**
Sinal de direção	Permanente
Sinal de corte de fio	Permanente
Sinal de HF	Permanente
Sensores de fim de curso	Controlado pelo micro controlador
Identificador de módulo de comunicação	Controlado pelo micro controlador (Interno ao Micro Controlador)

Tabela 4.4: Sinais pull up permanentes e temporários.

4.1.6 Circuito Ponte H

A ponte H permite que uma tensão seja aplicada a uma carga em direções diferentes, como se pode ver na Figura 4.9. Este circuito é normalmente usado para controlar um motor DC para andar para a frente e para trás. O chip da ponte H escolhido é o DRV8837CD da Texas Instruments. Este chip foi escolhido devido à quantidade de corrente que consegue transmitir aos motores (até 1 A) e devido às tensões lógicas que podem ser aplicadas (entre 1.8V até 5V). Para além disso também possui um pino para ser colocado em *sleep*. No entanto este é colocado mantido sempre no estado lógico '1' uma vez que, quando a ponte H não está em uso, esta não é alimentada.

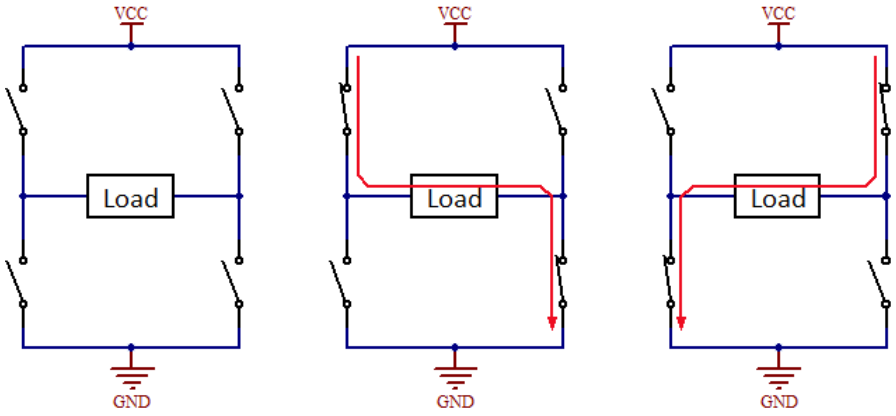


Figura 4.9: Circuito Ponte H.

4.1.7 Características da Bateria

A bateria usada tem as seguintes características:

EVE ER34615M	
Tipo de bateria	Não recarregável
Química	Lithium Thionyl Chloride
Gama de Tensões	3.6V
Capacidade Nominal	13.0Ah @ 15.0mA a 2.0V @ 20°C
Corrente Máxima Continua	2000mA
Taxa de auto-descarga	Menos de 3% por ano

Tabela 4.5: Características da bateria ER34615M [60].

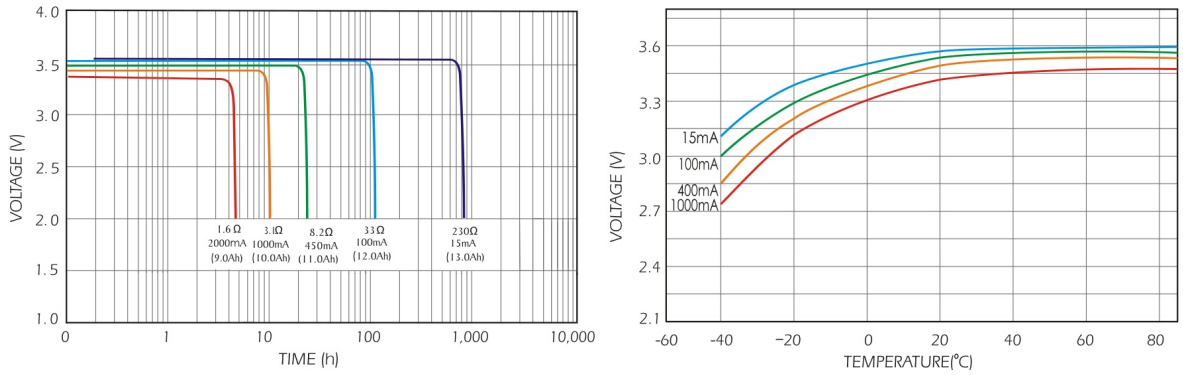


Figura 4.10: Variação da tensão da bateria consoante o tempo, a temperatura e a carga afetada.

Na figura 4.10, observa-se que a tensão é constante ao longo da descarga de energia para cargas e temperaturas constantes, isto quer dizer que um sistema alimentado por esta bateria não sofrerá de variações de tensão durante a vida de funcionamento. Como a queda de tensão é constante durante a vida da bateria, é difícil determinar o seu estado de carga de uma forma simples. Alguns métodos usados na determinação de carga de baterias podem ser encontrados em [61]. Alternativamente, pode usar-se o X-NUCLEO-LPM01A [62] da STM32, expressamente desenvolvido para o efeito de medida de consumo de energia de PCBs da mesma empresa, com o objetivo de compreender o que diferentes ações representam em energia consumida. Este

módulo possui uma resolução de medida entre 1 nA e 200 mA.

4.1.8 Estado de Abertura da Caixa

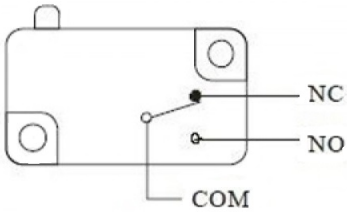


Figura 4.11: *Normally-Closed Switch*.

Para se certificar de que não houve adulteração do *hardware* do sistema é colocado um *Normally-Closed Switch* ligado a uma porta de interrupção externa do micro controlador. O *Switch* é constituído por 3 pinos que consoante o estado do botão, o sinal de saída vai alternar para uma das duas entradas. Neste sistema, este dispositivo vai criar uma mudança de sinal (de 0 para 1) quando a caixa é aberta. O dispositivo que se pretende usar é o SS-01GLPD que tem a particularidade de ter uma alavanca que vai manter o botão do *switch* premido até a caixa ser aberta [63].

4.1.9 Entradas e Saídas do micro controlador STM32L053R8

Na Figura 4.12 apresenta-se o micro controlador, todas as entradas a serem lidas e todas as saídas a serem controladas:

Na Figura 4.12 é possível identificar várias cores para diferentes pinos. Os pinos verdes escuros são os definidos pelo utilizador. Estes pinos são usados no sistema para receber/transmitir informação ou são necessários no uso de certos periféricos. Os alfinetes nos pinos verde escuro indicam que eles foram escolhidos manualmente, enquanto que aqueles que não têm foram escolhidos pelo software STM32CubeMX, software esse que permite a geração de código para dispositivos ST, dentro dos disponíveis para cada periférico. Os pinos laranja podem ser reconfigurados pelo utilizador. Estes pinos variam de placa para placa e são usados para verificar o bom funcionamento da mesma antes desta ser distribuída para venda. Os pinos amarelos não podem ser alterados.

Os pinos com a terminação EXTI são usados para acordar o micro controlador quando há uma mudança de estado lógico externo. No total, há 16 linhas de interrupção externas. Cada número nos pinos, de 0-15, representa uma dessas linhas. No entanto só um dos grupos de portas (A, B, C ou D) com um determinado número pode estar definido como tal. Por exemplo, se PA11 está definido como entrada EXTI, então PB11, PC11 e PD11 não podem ser colocadas para gerar interrupções.

Os pinos RCC estão conectados a cristais osciladores. Estes cristais são usados para calibrar o oscilador interno, usado como relógio para *timers*, entre outras funcionalidades.

Os pinos TMS, TCK e NRST são usados para programar o micro controlador através do programa ST-LINK/V2.

A USART1 é usada para *upgrade* do *firmware* do dispositivo.

O pino BootLoader_Mode é usado para forçar o dispositivo para modo *bootloader* após *reset* do sistema. Para tal, a entrada está ligada a sistema *pull up*. O *pull up* está conectado por um *jumper* a uma massa. Quando o *jumper* é retirado, a entrada do micro controlador deve estar à tensão de Vdd.

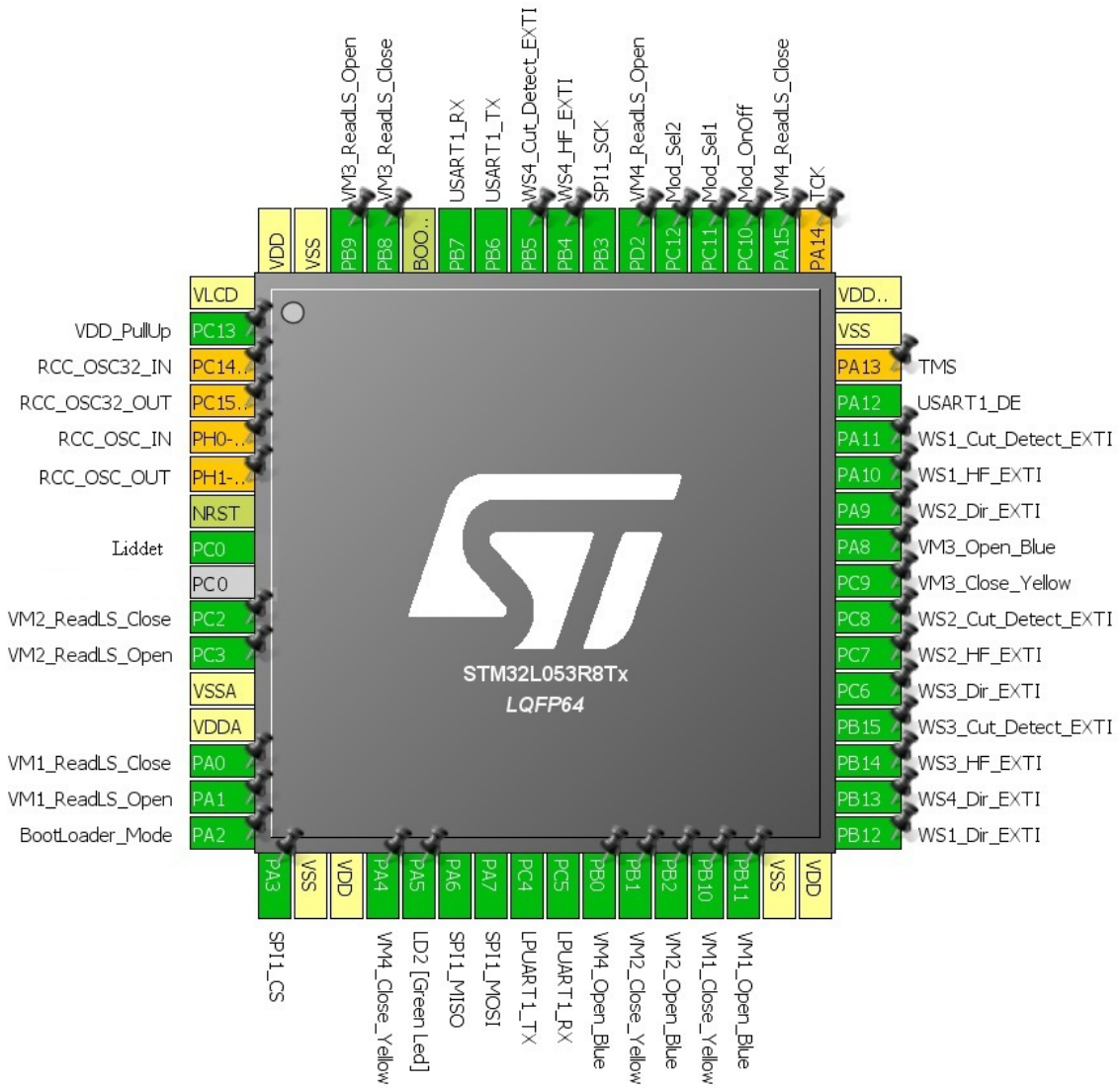


Figura 4.12: Micro controlador, Entradas e Saídas (WS - Water Sensor; VM - Valve Motor)

In Circuit Programming

In Circuit Programming é um método usado na programação de micro controladores soldados num PCB.

É possível programar a placa por meio dos pinos TMS, TCK e NRST com o programa *ST-LINK utility*. Para tal, é necessário um ST-LINK/V2 que conecta ao computador através de um cabo USB/Micro-USB e ao micro controlador através de um cabo com 20 pinos. A ligação do cabo com 20 pinos ao micro controlador é feita segundo as características seguintes:

(1) Fonte de energia ligado à placa usado para garantir compatibilidade de sinal entre a placa e o conector ST-LINK/V2.

(2) Ligar a GND para diminuir ruído no cabo conector.

(3) É necessário pelo menos estar ligado a GND para haver bom funcionamento (é recomendado a ligação de todos)

(4) Opcional: Usado para observar o SWV (*Serial Wire View*)

Para programar o dispositivo também é necessário retirar os *jumpers* como se pode ver na Figura 4.14. Caso a placa superior tenha sido retirada do sistema, basta ligar os cabos.

Figura 4.13: ST-LINK/V2 e um SWV flat ribbon com 20 pinos, respetivamente [64].

É possível programar outra placa STM32 através da placa da Figura 4.14, bastando para isso ligar os fios SWD (*Serial Wire Debug*) aos respetivos pinos da placa que se pretende programar e retirar os *jumpers* da placa removível.

4.1.10 Descrição do Sistema Identificador do Módulo de Comunicação

Dependendo da configuração das entradas Mod_Sel1 e Mod_Sel2 (ver Figura 4.12), o micro controlador identifica qual o módulo de comunicação montado no PCB. Os módulos são codificados com o '00' (Wi-Fi), '01' (LoRa), '10' (NB-IoT) e '11' (GSM/GPRS). Caso não haja módulo ligado ao sistema, a entrada é lida como '11'. Estas entradas usam circuitos *pull up* internos ao micro controlador para o seu funcionamento. A conexão Mod_OnOff liga e desliga o módulo.

4.1.11 Memórias

Flash

A memória Flash interna da classe 3 da série de micro controladores STM32L0 está dividida em dois bancos de dados (*Bank 1* e *Bank 2*). Cada banco é divido em 16 sectores de 32 páginas. Cada página é composta por 128 bytes [52].

A existência de dois bancos separados permite várias possíveis utilizações do espaço [66]:

- *Big Code*: Código que beneficia dos dois bancos, em que é necessário dois modos de funcionamento diferente (master/slave; transmitir/receber).
- *Data Storage*: A informação é guarda num banco e o código é executado no outro. Permite a rápida e fácil eliminação de toda informação do banco de dados. Este caso é possível devido recurso de *Read while Write*.
- *Code Backup*: Se houver alguma falha num dos bancos, o outro banco pode servir de reserva, podendo mesmo fazer uma copia dele para o banco em falha.
- *Dual bank field upgrade*: Permite o *upgrade* do *firmware*, evitando vários problemas possíveis para sistemas que só usam um único banco (o *loader* não conseguir completar algumas tarefas; não ser possível fazer *upgrade* ao *loader*; haver falha no *upgrade* e o sistema manter-se em modo *loader*).

Pin no.	ST-LINK/V2 connector (CN3)	Target connection (SWD)
1	VAPP	MCU VDD(1)
2		
3	TRST	GND(2)
4	GND	GND(3)
5	TDI	GND(2)
6	GND	GND(3)
7	TMS_SWDIO	SWDIO
8	GND	GND(3)
9	TCK_SWCLK	SWCLK
10	GND	GND(3)
11	NC	Not connected
12	GND	GND(3)
13	TDO_SWO	TRACESWO(4)
14	GND	GND(3)
15	NRST	NRST
16	GND	GND(3)
17	NC	Not connected
18	GND	GND(3)
19	VDD	Not connected
20	GND	GND(3)

Tabela 4.7: Ligações do ST-LINK/V2 com o micro controlador [65].

O bit BFB2 determina em qual dos bancos deve ser corrido o *bootloader*. Quando o bit está no estado lógico "0" corre do banco 1 e quando está no estado lógico "1" corre do banco 2.

Caso o módulo RF deixe de funcionar ou se houver necessidade de guardar informação extensa das leituras, o PCB possui uma memória externa EEPROM AT25SF041 [67]. Esta memória está ligada aos pinos do periférico SPI, inclusive o de seleção de chip, SPI1_CS.

EEPROM

O micro controlador possui uma memória programável EEPROM de 2 Kbytes.

Memória AT25SF041

Características da memória AT25SF041 [68]:

- 4 Mbits de memória,
- alimentação entre 2.5 V e 3.6 V,
- comunicação SPI,
- modos de funcionamento de baixo consumo (*Deep Power-Down Mode* - 2μA; *Standby Mode* - 10μA; *Active Read Mode* - 4 mA),

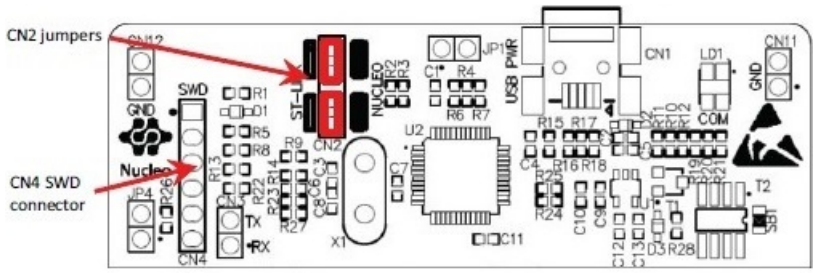


Figura 4.14: Ligações do ST-LINK/V2 com o micro controlador [65].

NVM	NVM addresses	Size (bytes)	Name	Description	
Flash program memory	0x0800 0000 - 0x0800 007F	128 bytes	Page 0	sector 0	Bank 1
	0x0800 0080 - 0x0800 00FF	128 bytes	Page 1		
	-	-	-		
	0x0800 0F80 - 0x0800 0FFF	128 bytes	Page 31		
	. . .	. . .	. . .	. . .	
	0x0800 7000 - 0x0800 707F	128 bytes	Page 224	sector 7	
	0x0800 7080 - 0x0800 70FF	128 bytes	Page 225		
	-	-	-		
	0x0800 7F80 - 0x0800 7FFF	128 bytes	Page 255		
	. . .	. . .	. . .	. . .	
	-	-	-		
	0x0801 7F80- 0x0801 7FFF	128 bytes	Page 767	sector 23	
	0x0801 8000 - 0x0801 807F	128 bytes	Page 768	sector 24	Bank 2
	. . .	. . .	. . .	. . .	
	0x0802 F000 - 0x0802 F07F	128 bytes	Page 1504	sector 47	
	0x0802 F080 - 0x0802 F0FF	128 bytes	Page 1505		
	-	-	-		
	0x0802 FF80 - 0x0802 FFFF	128 bytes	Page 1535		

Tabela 4.8: Memória Flash no micro controlador STM32L053R8 [52].

- comunicação SPI,
- capaz de ser reprogramada 100000 vezes,
- retenção de dados em memória até uma duração de 20 anos.

4.2 Estudo do funcionamento

Para determinar se é possível a sobrevivência do sistema para um período de 10 anos (3653 dias) é necessário identificar qual é a carga da bateria disponível:

$$Q_{BatNom} = 13.0 \text{ Ah} = 13 * 3600 = 46800 \text{ C} \tag{4.1}$$

Considerando uma descarga de 3 % ao fim de cada ano, temos então que a carga efetivamente disponível para um período de 10 anos é dada por:

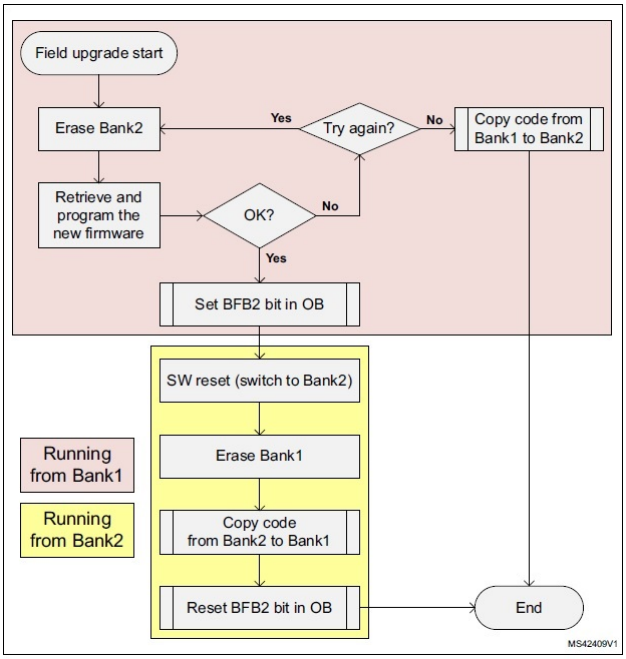


Figura 4.15: Rotina de *Dual bank field upgrade* [66].

NVM	NVM addresses	Size (bytes)	Name	Description
Data EEPROM	0x0808 0000 - 0x0808 07FF	2 Kbytes	-	Data EEPROM

Tabela 4.9: Memória EEPROM no micro controlador STM32L053R8.

$$\mathrm{Q}_{BatNomDis} = \mathrm{Q}_{BatNom} * (1 - \mathrm{AutoDesc})^{\mathrm{N}_{Anos}} = 46800 * (1 - 0.03)^{10} = 34511.45 \text{ C} \quad (4.2)$$

Para facilitar as contas é considerado o modelo de funcionamento de um dia. A carga inicial da bateria para um dia é de:

$$\mathrm{Q}_{DiaDisp} = \frac{\mathrm{Q}_{BatNomDis}}{\mathrm{N}_{Dias}} = \frac{34511.45}{3653} = 9.45 \text{ C/dia} \quad (4.3)$$

O sistema deve contar sempre que haja um impulso de um dos contadores de água. Daqui determina-se a existência de duas opções para o funcionamento do sistema em relação à contagem de água. A primeira opção resume-se ao micro controlador receber o impulso, acordar, fazer a contagem e voltar a dormir. A segunda opção procura encontrar um equilibro entre o tempo em que o micro controlador se mantém acordado, após cada leitura, e o número de impulsos que está a receber, ou seja, o consumo em funcionamento normal e o consumo que ocorre no acordar do micro. A segunda opção é relevante uma vez que é possível haver períodos em que há uma grande quantidade de impulsos a serem recebidos, o que quer dizer que haverá situações em que é preferível manter o micro controlador acordado em vez de acordar/adormecer a cada impulso. A decisão de funcionar numa das opções descritas é tomada consoante a quantidade de impulsos recebidos num certo período de tempo. Ou seja, se forem recebidos 15 ou mais impulsos, estando o micro controlador a acordar e a adormecer para cada um deles, caso o tempo de ocorrência do conjunto de impulsos seja menor relativamente aquele que seria

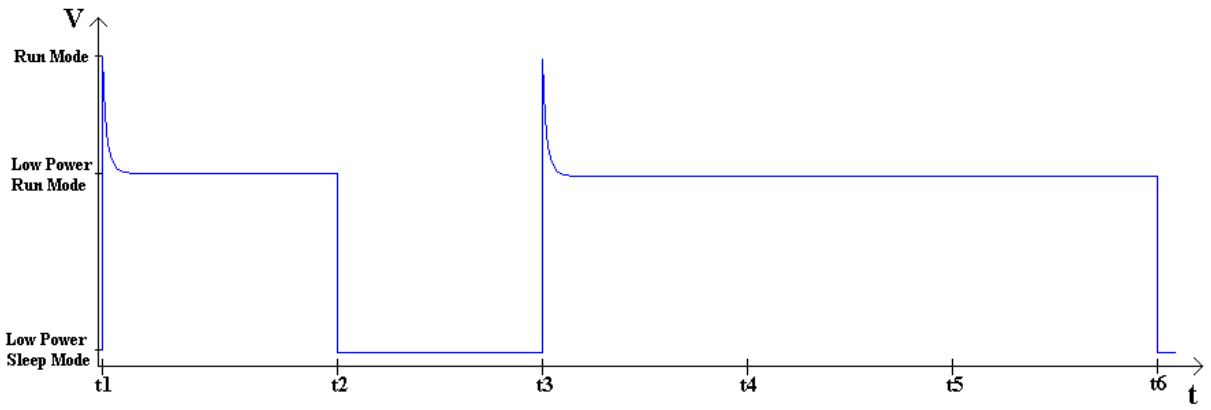


Figura 4.16: Exemplo de funcionamento da passagem de modos de trabalho do micro controlador.

---

necessário para que o consumo em funcionamento normal fosse igual ao consumo que ocorre no acordar do micro, este passa a esperar esse tempo antes de voltar a dormir.

O consumo médio de água por pessoa em 2011 foi de 220 litros por dia [69]. Isso quer dizer que para uma habitação de 3 pessoas o gasto em litros de água é de 660 litros. Na empresa em que foi realizado o projeto, determinou-se que é consumido em média 19 metros cúbicos de água por mês, ou seja, 633.3 litros de água por dia. Como os valores de água consumida são muito semelhantes e pretende-se projetar o sistema para o pior caso, decidiu-se usar o de 660 litros. Se o micro controlador gerir quatro habitações diferentes, ele vai receber em média 2640 impulsos por dia.

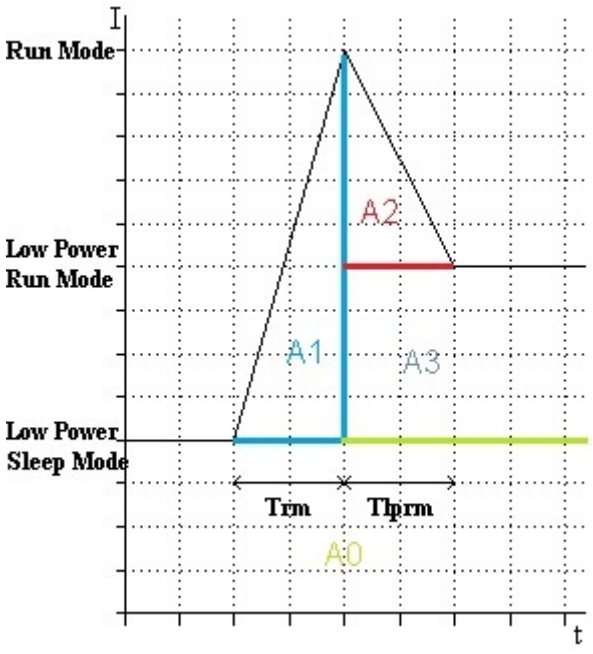


Figura 4.17: Exemplo de passagem de modo de Low Power Sleep para Low Power Run.

Os valores dos consumos dos modos podem ser encontrados na Tabela 4.1.

$$T_{rm} = 32\ \mu s \tag{4.4}$$

$$\mathrm{T}_{lprm} = 3\ \mu\mathrm{s} \tag{4.5}$$

$$\mathrm{Q}_{A1} = \frac{1}{2}(\mathrm{I}_{RunMode} - \mathrm{I}_{LPSleepM}) * \mathrm{T}_{rm} = \frac{1}{2}(555 * 10^{-6} - \frac{17+23}{2} * 10^{-6}) * 32 * 10^{-6} \tag{4.6}$$

$$\mathrm{Q}_{A1} = 8.56\ \mathrm{nC} \tag{4.7}$$

$$\mathrm{Q}_{A2} = \frac{1}{2}(\mathrm{I}_{RunMode} - \mathrm{I}_{LPRunM}) * \mathrm{T}_{lprm} = \frac{1}{2}(555 * 10^{-6} - \frac{22+28}{2} * 10^{-6}) * 3 * 10^{-6} \tag{4.8}$$

$$\mathrm{Q}_{A2} = 0.795\ \mathrm{nC} \tag{4.9}$$

$$\mathrm{I}_{A0} = \mathrm{I}_{LPSleepM} = \frac{17+23}{2} * 10^{-6} = 20 * 10^{-6}\ \mathrm{A} \tag{4.10}$$

$$\mathrm{I}_{A3} = (\mathrm{I}_{LPRunM} - \mathrm{I}_{LPSleepM}) = 5 * 10^{-6}\ \mathrm{A} \tag{4.11}$$

$$\mathrm{Q}_{Acordar} = \mathrm{I}_{A0} * (\mathrm{T}_{rm} + \mathrm{T}_{lprm}) + \mathrm{Q}_{A1} + \mathrm{Q}_{A2} + \mathrm{I}_{A3} * \mathrm{T}_{lprm} = 10.07\ \mathrm{nC} \tag{4.12}$$

$$\mathrm{T}_{Acordado} = \frac{\mathrm{Q}_{Acordar}}{\mathrm{I}_{LPRunM}} = \frac{10 * 10^{-}9}{25 * 10^{-6}} = 400\ \mu\mathrm{s} \tag{4.13}$$

Ou seja, para o gasto de estar em modo *Low Power Run Mode* ser maior em relação ao gasto da passagem de *Low Power Sleep* para *Low Power Run Mode*, o micro controlador precisa de estar em funcionamento mais 400 μs.

No caso em que todos os impulsos acordem o micro controlador do modo *Low Power Sleep* e que cada impulso corresponde a 1 litro de água, é consumido por dia:

$$\mathrm{Q}_{PiorCasoAcordar} = \mathrm{N}_{MaxImpDia} * \mathrm{Q}_{Acordar} = 2640 * 10 * 10^{-9} = 26.4\ \mu\mathrm{C} \tag{4.14}$$

Assumindo que são executado pelo menos 1000 instruções, para uma frequência de trabalho de 32 kHz, sempre que o micro controlador é acordado para contar, é consumido:

$$\mathrm{T}_{PorInst} = \frac{1}{32000} = 31.25\ \mu\mathrm{s} \tag{4.15}$$

$$\mathrm{T}_{1000Inst} = 1000 * \mathrm{T}_{PorInst} = 31.25\ \mathrm{ms} \tag{4.16}$$

$$\mathrm{Q}_{1000Inst} = \mathrm{I}_{LPRunM} * \mathrm{T}_{1000Inst} = 25 * 10^{-6} * 31.25 * 10^{-3} = 0.78\ \mu\mathrm{C} \tag{4.17}$$

O consumo total do micro controlador por dia, sem considerar o acordar do micro para comunicação, é:

$$\mathrm{Q}_{TotalMicro} = \mathrm{Q}_{PiorCasoAcordar} + \mathrm{Q}_{1000Inst} * \mathrm{N}_{MaxImpDia} + \mathrm{Q}_{ADormir} \tag{4.18}$$

$$\mathrm{Q}_{TotalMicro} = 26.4 * 10^{-6} + 0.78 * 10^{-6} * 2640 + 86400 * 20 * 10^{-6} = 1.73\ \mathrm{C} \tag{4.19}$$

Há 3 *pull ups* permanentes (ver secção 4.1.5) por conjunto válvula/motor.Os *pull ups* controlados pelo micro controladores são ignorados porque só consomem quando são puxados 0 e são desligados quando isto ocorre. Se for usado uma resistência de 100 kΩ para cada *pull up* , de forma a permitir corrente suficiente para poder haver leitura das mudanças de estados, temos:

$$\mathrm{I}_{ConsumoPullUp} = \frac{3.6}{100 * 10^3} = 36\ \mu\mathrm{A} \tag{4.20}$$

Este consumo só ocorre quando a entrada é puxada ao nível lógico "0". Quando se mantém no estado "1" o consumo é menosprezável.

O sensor envia cada impulso no máximo com uma frequência de 14 Hz. Assumindo que o tempo no estado "0" é igual ao tempo em "1", para cada *pull up* permanente é consumido por dia:

$$\mathrm{Q}_{PullUpPerma} = \mathrm{I}_{ConsumoPullUp} * \mathrm{T}_{Estado0} * \mathrm{N}_{Imp} = 36 * 10^{-6} * 0.065 * 2640 = 6.2\ \mathrm{mC} \tag{4.21}$$

Por medição em laboratório, determinou-se que a válvula consome 0.75 A para abrir e para fechar, demorando no máximo 5 segundos para executar o movimento. É consumido:

$$\mathrm{Q}_{MovValv} = \mathrm{I}_{MovValv} * \mathrm{T}_{MovValv} = 0.75 * 5 = 3.75\ \mathrm{C} \tag{4.22}$$

$$\mathrm{Q}_{LSPullUp} = \mathrm{I}_{ConsumoPullUp} * \mathrm{T}_{MovValv} = 62 * 10^{-6} * 5 = 310\ \mu\mathrm{C} \tag{4.23}$$

$$\mathrm{Q}_{ValvConsumoTotal} = \mathrm{Q}_{MovValv} + \mathrm{Q}_{LSPullUp} = 3.75\ \mathrm{C} \tag{4.24}$$

Todos os módulos de comunicação gastam sensivelmente o mesmo na transmissão (ver secção 4.1.2). Para o dispositivo de Wi-FI que consome 243 mA na transmissão de informação *uplink*, em que o tempo máximo de transmissão é de 5 segundos, e que é feito apenas uma transmissão por dia, temos:

$$\mathrm{Q}_{TransTotal} = \mathrm{I}_{Trans} * \mathrm{T}_{TransMaxima} + \mathrm{Q}_{Acordar} + (\mathrm{I}_{A0} + \mathrm{I}_{A3}) * \mathrm{T}_{TransMaxima} \tag{4.25}$$

$$\mathrm{Q}_{TransTotal} = 243 * 10^{-3} * 5 + 10.07 * 10^{-9} + 25 * 10^{-6} * 5 = 1.21\ \mathrm{C} \tag{4.26}$$

A memória externa consome 2 μA quando está em (Deep Power-Down Mode) e 4 mA quando está em (Active Read Mode). A memória só é escrita quando há uma transmissão de dados para ela. Pode ser necessário mais que uma transmissão para a transferência total do código.

A memória em (Deep Power-Down Mode) consome:

$$\mathrm{Q}_{MemDPDM} = \mathrm{I}_{MemDPDM} * \mathrm{T}_{Dia} = 2 * 10^{-6} * 24 * 3600 = 0.17\ \mathrm{C} \tag{4.27}$$

A memória em (Active Read Mode) consome:

$$\mathrm{Q}_{MemARDM} = \mathrm{I}_{MemARDM} * \mathrm{T}_{TransMaxima} + \mathrm{Q}_{TransTotal} = 4 * 10^{-3} * 5 + 1.21 = 1.23\ \mathrm{C} \tag{4.28}$$

O consumo total aproximado de um dia é dado então por:

$$\mathrm{Q}_{Total} = \mathrm{Q}_{Micro} + \mathrm{Q}_{PullUp} + \mathrm{Q}_{Valv} + \mathrm{Q}_{Modulo} + \mathrm{Q}_{Mem} \tag{4.29}$$

Em que:

$$\mathrm{Q}_{Micro} = \mathrm{Q}_{Acodar} * \mathrm{N}_{Impul} + \mathrm{Q}_{1000Inst} * \mathrm{N}_{Impul} + \mathrm{Q}_{LPSleep} \tag{4.30}$$

$$\mathrm{Q}_{PullUp} = \mathrm{I}_{ConsumoPullUp} * \mathrm{T}_{Estado0} * \mathrm{N}_{Imp} \tag{4.31}$$

$$\mathrm{Q}_{Valv} = \mathrm{Q}_{ValvConsumoTotal} * \mathrm{N}_{Movimetos} \tag{4.32}$$

$$\mathrm{Q}_{Modulo} = \mathrm{Q}_{TransTotal} * \mathrm{N}_{Trans} \tag{4.33}$$

$$\mathrm{Q}_{Mem} = \mathrm{I}_{MemDPDM} * \mathrm{T}_{MemDPDM} + \mathrm{I}_{MemARDM} * \mathrm{T}_{MemARDM} + \tag{4.34}$$

Tendo apenas um conjunto de contador/válvula, sem haver movimento de válvula, em que é feito apenas uma transmissão por dia e em que a memoria externa não é acedida, o consumo é de:

$$\mathrm{Q}_{GastoMinimo} = \mathrm{Q}_{Micro} + \mathrm{I}_{ConsumoPullUp} * \mathrm{T}_{Estado0} * \mathrm{N}_{Imp} + \mathrm{Q}_{TransTotal} * \mathrm{N}_{Trans} + \mathrm{Q}_{MemDPDM} \tag{4.35}$$

$$\mathrm{Q}_{GastoMinimo} = 1.73 + 6.2 * 10^{-3} + 1.21 * 1 + 0.17 = 3.11 \text{ C} \tag{4.36}$$

O tempo de vida corresponde a:

$$\mathrm{T}_{AnosTrabalho} = \frac{\mathrm{Q}_{BatNomDis}}{\mathrm{Q}_{GastoMinimo} * 365} = 30.3 \text{ Anos} \tag{4.37}$$

Tendo 4 conjuntos de contador/válvula, sem haver movimento de válvula, em que é apenas feito uma transmissão por dia e em que a memoria externa não é acedida, o consumo é de:

$$\mathrm{Q}_{GastoMinimo} = \mathrm{Q}_{Micro} + 4 * \mathrm{I}_{ConsumoPullUp} * \mathrm{T}_{Estado0} * \mathrm{N}_{Imp} + \mathrm{Q}_{TransTotal} * \mathrm{N}_{Trans} + \mathrm{Q}_{MemDPDM} \tag{4.38}$$

$$\mathrm{Q}_{GastoMinimo} = 1.73 + 4 * 6.2 * 10^{-3} + 1.21 * 1 + 0.17 = 3.13 \text{ C} \tag{4.39}$$

O tempo de vida corresponde a:

$$\mathrm{T}_{AnosTrabalho} = \frac{\mathrm{Q}_{BatNomDis}}{\mathrm{Q}_{GastoMinimo} * 365} = 30.2 \text{ Anos} \tag{4.40}$$

Em suma, para ambos os casos o sistema deve conseguir atingir e passar o tempo de vida esperado.

4.3 PCBs desenhados e funções

4.3.1 PCB Principal

Na Figura 4.18 encontra-se o principal *Printed Circuit Board* (PCB) desenhado no desenvolvimento do projeto. O respetivo modelo 3D gerado em Altium durante o desenho do PCB pode ser encontrado na secção A.

O PCB principal é constituído por duas camadas de cobre que são usadas ambas como terra para todo o desenho. O desenho pode ser dividido em 3 áreas: a área da pilha, no lado superior esquerdo, com o *Top Overlay* que designada a posição da pilha; a área do micro controlador, no lado superior direito; e a área de ligações dos motores e dos contadores na parte de baixo.

A pilha vai ser mantida na mesma posição através de um atilho que passa pelo buraco existente no *Top Overlay* do lado esquerdo e pela borda da placa.

No lado direito é possível colocar um PCB *Piggyback* removível que irá alojar as placas dos módulos de comunicação, como se pode ver na Figura 4.19.

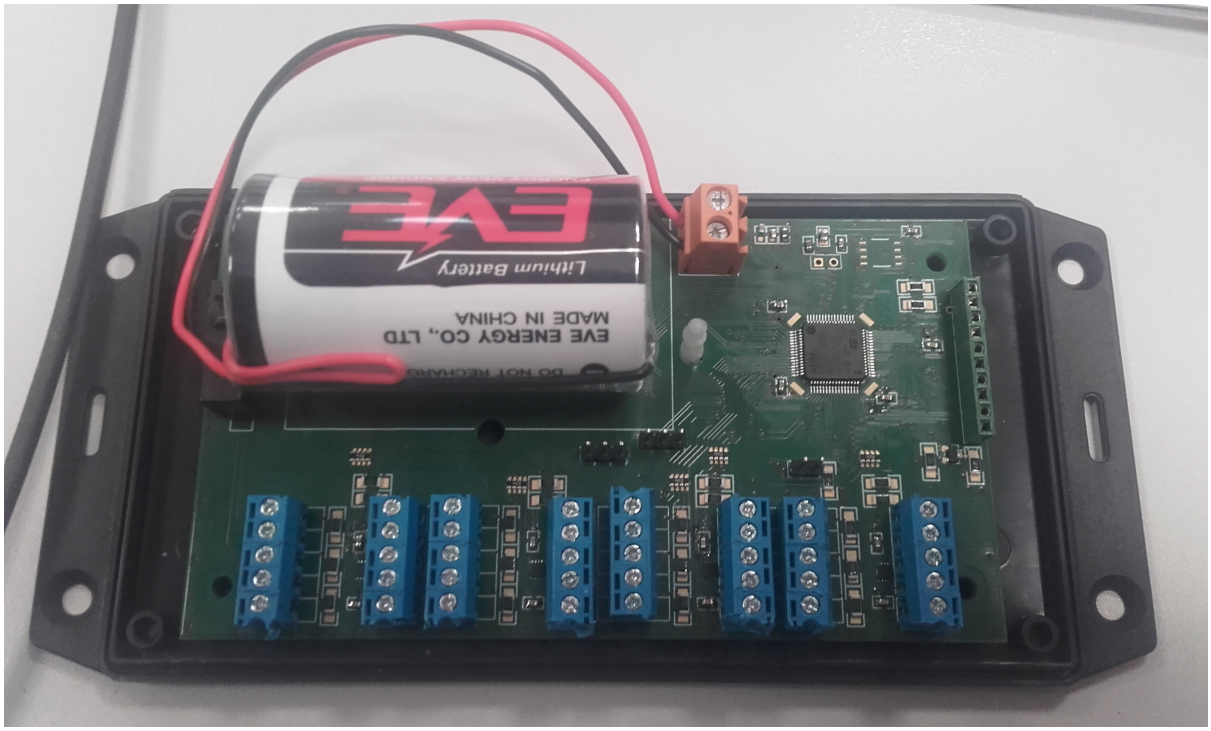


Figura 4.18: PCB principal.

Todas as pistas desenhadas no PCB possuem ângulos de 45° para minimizar a existência de problemas devido à acumulação de ácido nos cantos caso tivessem 90°. No forum [70] é argumentado que não há a necessidade de ter todas as pistas a 45°, que era uma precaução no método de fabrico, mas que, eventualmente, deixou de ser necessário de se ter e que eventualmente se transformou numa tradição de desenho, no entanto, para desenho de sistemas de alta tensão é uma característica relevante.

Tentou-se fazer o plano de massa de baixo o menos retalhado possível, mantendo a passagem das pistas de um plano para outro o mínimo possível, para tentar obter a menor resistência possível. Foram colocadas vias para uniformizar os dois planos de massa e para conectar várias pistas à massa.

Da esquerda para a direita do PCB na Figura 4.20, o *Top Overlay* com os 3 *pads* é onde se vai montar o *switch* de controlo de abertura de caixa. O *Top Overlay* com os 3 *pads* seguintes é usado para permitir *In Circuit Programing*. O *Top Overlay* com os 3 *pads* seguintes permite comunicação USART diretamente. Por baixo do micro controlador, o*Top Overlay* com os 2 *pads* permite acesso direto as funcionalidades de memória. Na área superior é colocado um LED para ser possível correr testes e haver controlo no desenvolvimento de código. Por erro não foi colocado espaço para uma resistência entre o LED e o micro controlador para controlar a corrente enviada. Por fim, na área direita da placa existe um *header* para conectar ao PCB *Piggyback*.

A placa foi projetada para ser colocada numa caixa 1591XXDSBK [71].

Para facilitar e melhorar o desenho de PCBs, ferramentas como a que se enconta em [72], que permitem escolher os melhores tamanhos para as vias e pistas, consoante as especificações do projeto. Estas ferramentas não foram usadas no desenvolvimento do projeto.

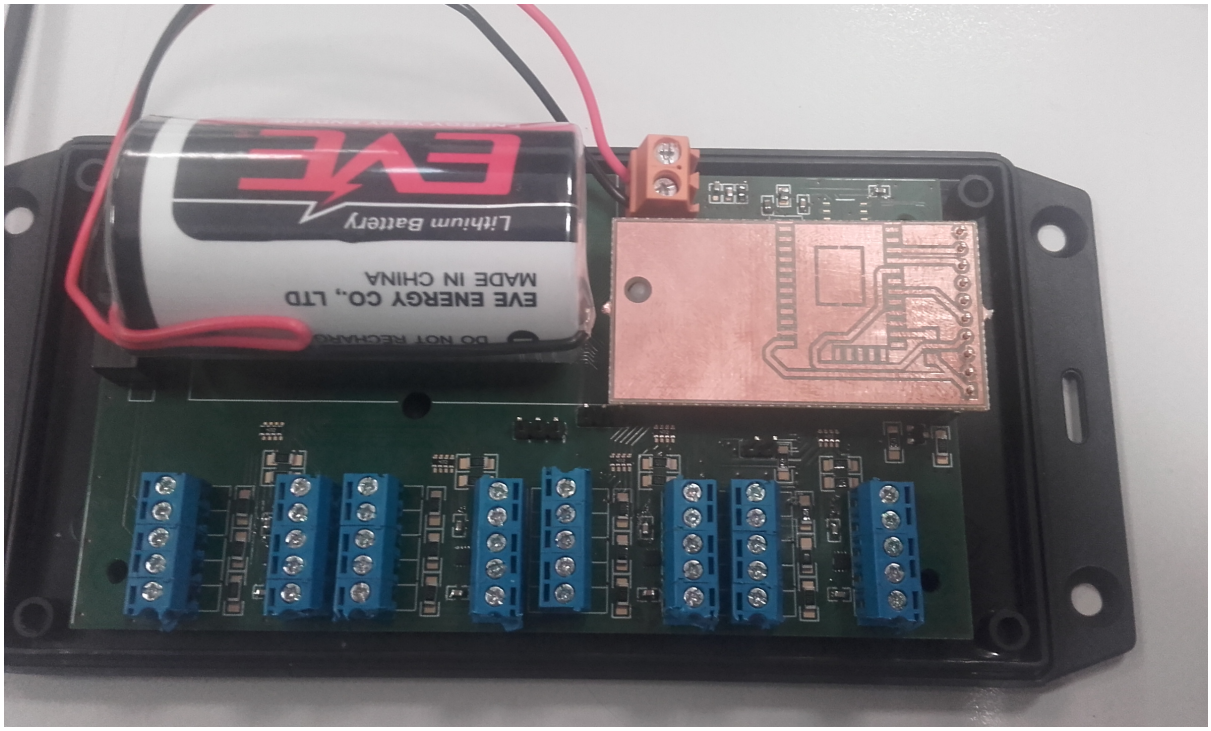


Figura 4.19: PCB principal com módulo SPWF015A.

4.3.2 PCB SPWF01SA

Como se pode ver na Figura 4.21, a placa consiste apenas de um furo, um *header* e o módulo de comunicação. O PCB foi desenhado com apenas uma camada. Este PCB não precisa de mais nada porque todos os componentes necessários para o funcionamento do módulo estão presentes no mesmo.

4.4 Resultados Experimentais

O projeto culminou no desenho do PCB (ver figura 4.18) que serve de suporte para todo o sistema, sendo este a base para todas as medições que foram feitas. Os resultados encontram-se resumidos na Tabela 4.10.

O código utilizado foi gerado a partir do programa STM32CubeMX [73], o qual permite configurar os periféricos e os relógios através de uma interface gráfica.

O código utilizado para as medidas de corrente encontra-se entre o Anexo C e o Anexo G. Para todos os casos, o código inicializa todos os pinos do micro controlador, os periféricos que seriam usados em funcionamento normal, as interrupções e os relógios que são utilizados pelo micro controlador. O micro controlador utiliza o relógio MSI (*Multi-Speed Internal*) configurado com a frequência mais baixa possível e os periféricos de comunicação são configurados com o relógio HSI (High Speed Internal) para poderem comunicar com o módulo de comunicação, uma vez que este tem uma *baudrate* fixa de 115200.

No Anexo C são apenas feitas as inicializações do sistema para se poder medir o consumo em modo normal de funcionamento. Mediu-se 341 μA.

No Anexo D são feitas as inicializações do sistema e o sistema é colocado em modo *Low Power Run*. Mediu-se 341 μA.

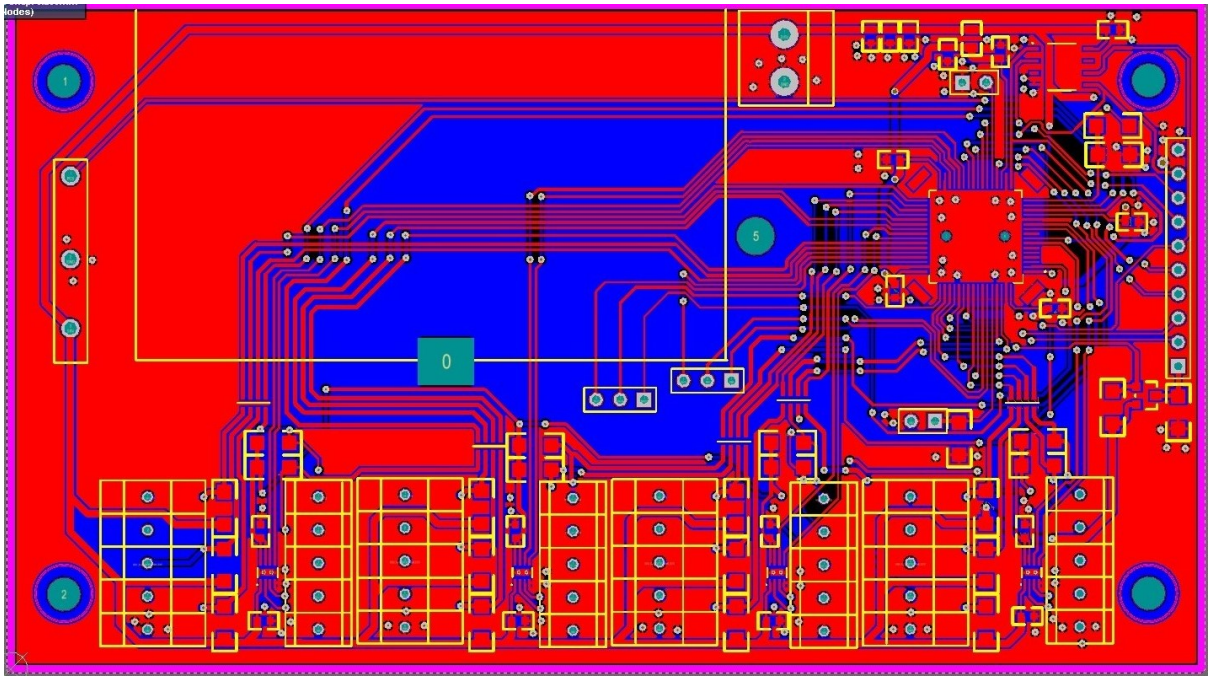


Figura 4.20: Desenho do PCB principal no Altium.

No Anexo E são feitas as inicializações do sistema e o sistema é colocado em modo *Low Power Sleep*. Mediu-se 341 μA.

No Anexo F são feitas as inicializações do sistema. Este entra em modo *Low Power Sleep*, espera por uma interrupção do RTC, o que ocorre de 10 em 10 segundos. Quando uma interrupção ocorre, o micro controlador entra no modo de *Low Power Run*, em que os *pull ups* controlados pelo micro são ativados e é dado a ordem para abrir ou fechar a válvula, até os sensores de fim de curso indicarem o fim do movimento. Quando o movimento termina, o micro controlador é colocado em modo *Low Power Sleep* até à próxima interrupção. Mediu-se 75 mA durante o movimento da válvula e 342 μA quando o sistema está a dormir.

No Anexo G são feitas as inicializações do sistema, o sistema é colocado em modo *Low Power Sleep*, entra em modo *Low Power Sleep*, espera por uma interrupção do RTC, o que ocorre de 35 em 35 segundos. Quando uma interrupção ocorre, o micro controlador entra no modo de *Low Power Run*, liga o módulo de comunicação e identifica-o como sendo de Wi-fi. O micro controlador espera que o módulo se ligue a uma rede pré-definida e que se conecte a um *socket* pré-determinado. O micro manda a mensagem IDCT01CT02CT03CT04E, em que ID são dois bytes usados para identificação do sistema, CT01 até CT04 são conjuntos de 4 Bytes com os valores lidos pelos os quatro contadores, respetivamente, e E um Byte com a informação de erros e problemas que tenham surgido no sistema, e pergunta se há algo novo para receber. Tendo recebido a resposta, o micro controlador desliga o módulo, sai de *Low Power Run*, entra em modo *Low Power Sleep* e fica à espera da próxima interrupção do RTC. Na localização em que foi realizado a medição não se conseguiu colocar o modem à rede, tendo sido medido o momento em que módulo está ligado sem fazer nada, 25 mA, e o momento em que o módulo está à procura de redes, 85 mA.

Num ensaio em que o micro controlador foi iniciado sem nenhum periférico ativo, em que todos os pinos estão no estado analógico e que o único relógio iniciado é o MSI configurado com a frequência mais baixa possível, em modo normal, o sistema consumia 43.4 μA, em LPSleep consumia 6.6 μA e em LPRun, consumia 25 μA.

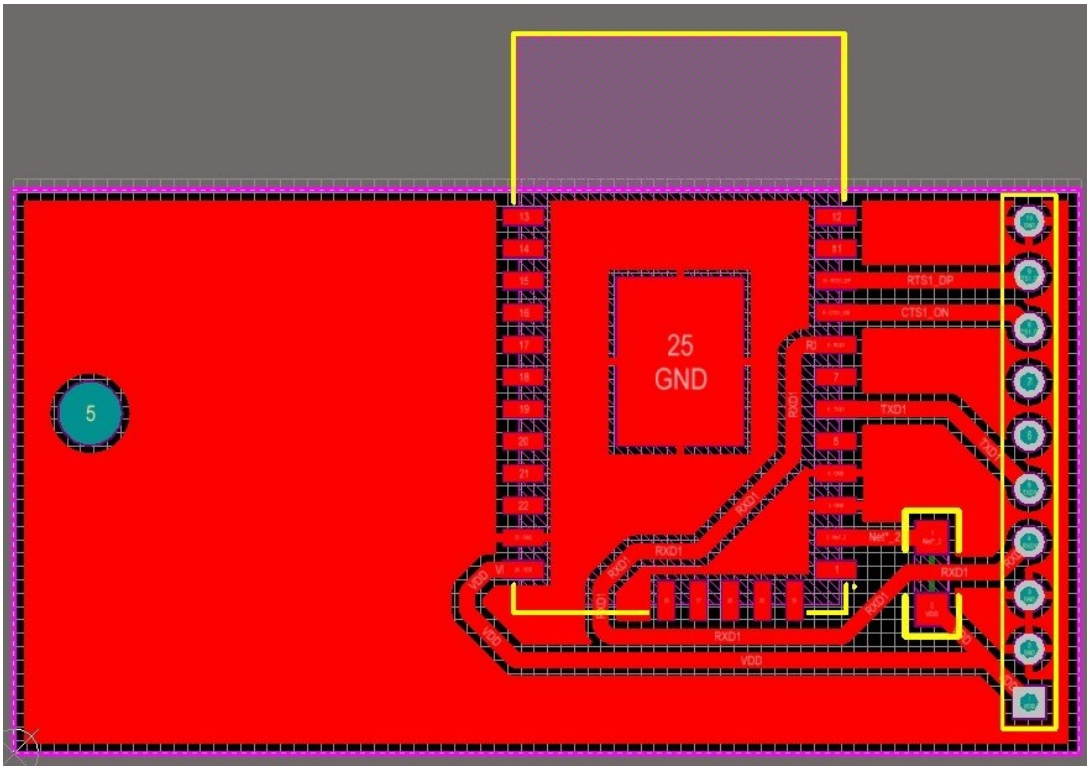


Figura 4.21: PCB do com módulo SPWF015A.

Usando os consumos encontrados na Tabela 4.10, em específico quando o micro controlador está a contar impulsos, como o modelo de trabalho mais comum que irá realizar, assumindo que o sistema em modo RUN possui a corrente máxima encontrada na Tabela 4.1, baseando-nos na equação 4.16, para um dia de funcionamento, tendo uma configuração de dois relógios (65 kHz para processamento e 16 MHz para periféricos), tem-se:

$$\mathrm{Q}_{TotalMicro} = 32 * 10^{-3} + 29.98 = 30 \text{ C} \tag{4.41}$$

$$\mathrm{T}_{AnosTrabalho} = \frac{\mathrm{Q}_{BatNomDis}}{\mathrm{Q}_{GastoMinimo} * 365} = 3 \text{ Anos} \tag{4.42}$$

Usando apenas um relógio (524 kHz para processamento e para periféricos), tem-se:

$$\mathrm{Q}_{TotalMicro} = 62 * 10^{-6} * 3600 * 24 = 5.4 \text{ C} \tag{4.43}$$

$$\mathrm{T}_{AnosTrabalho} = \frac{\mathrm{Q}_{BatNomDis}}{\mathrm{Q}_{GastoMinimo} * 365} = \frac{\mathrm{Q}_{BatNomDis}}{(6 + 3.13) * 365} = 11.1 \text{ Anos} \tag{4.44}$$

Na secção 4.2 previu-se o que sistema consumia por dia no máximo 3.13 C na situação em que o relógio usado para o processamento e para os periféricos fosse o MSI, à frequência de 65 kHz. Para a primeira configuração, o consumo é 10 vezes maior que aquele previsto. Isto deve-se ao relógio HSI, à frequência de 16 MHz, usado pelos os periféricos de transmissão de dados para garantir a comunicação com o módulo RF montado. A segunda configuração procurou aumentar o mínimo possível da frequência do relógio MSI para permitir também ser usado pelos periféricos de transmissão de dados. O MSI ficou configurado com a frequentaria de 524 kHz. O consumo para a segunda configuração está dentro daquele que é desejado para o sistema.

4.5 Conclusões do Capítulo

Este capítulo apresentou todos os componentes que foram usados no desenvolvimento do projeto e os PCBs desenhados para o sistema. Apresentou, também, o consumo esperado de todas as partes do sistema, como é esperado que este funcione e as medições que foram realizadas para confirmar estes pressupostos.

Concluiu-se que o consumo do sistema é acima do que era previsto para uma configuração de dois relógios, um para processamento e outro usado pelos periféricos de comunicação, não conseguindo cumprir o prazo de vida de 10 anos. No entanto, aumentando a frequência do relógio usado para o processamento, este pode ser usado também pelos periféricos, permitindo assim o tempo de vida de 10 anos.

Nome	Entrada - In Saída - Out Digital - Dig. Analógico - An.	Descrição	Gama de entrada	Gama de saída no firmware
WS1-4_Dir_EXTI	Dig. In	Entrada de interrupção externa assíncrona. Usado para acordar o micro controlador na mudança do estado da direção da água.	0 - Frente 1 - Trás	—
WS1-4_Cut_Detect_EXTI	Dig. In	Entrada de interrupção externa assíncrona. Usado para acordar o micro controlador na deteção de um corte de sensor.	0 - Inativo 1 - Corte	—
WS1-4_HF_EXTI	Dig. In	Entrada de interrupção externa assíncrona. Usado para acordar o micro controlador na deteção de um impulso. Cada impulso corresponde a uma certa quantidade de água.	0 - Inativo 1 - Passou x litros de água	—
VM1-4_Open_Blue	Dig. Out	Saída digital. Usado para indicar à válvula para abrir.	—	0V-3.3V
VM1-4_Close_Yellow	Dig. Out	Saída digital. Usado para indicar à válvula para fechar.	—	0V-3.3V
VM1-4_ReadFC_Open	Dig. In	Entrada digital. Usado para ler o sensor de fim de curso para identificar se a válvula está completamente aberta.	0 - Desconhecido 1- Aberto	—
VM1-4_ReadFC_Close	Dig. In	Entrada digital. Usado para ler o sensor de fim de curso para identificar se a válvula está completamente fechada.	0 - Desconhecido 1- Fechado	—
Mod_OnOff	Dig. Out	Saída digital. Usado para desligar ou ligar o módulo de comunicação selecionado.	—	0V-3.3V
Mod_Sel1-2	Dig. In	Entrada digital. Usado para identificar o módulo de comunicação desejado.	—	0V-3.3V
LPUART1_TX	Dig. Out	Saída série da UART. Usado para transmitir informação através do protocolo UART.	—	—
LPUART1_RX	Dig. In	Entrada série da UART. Usado para receber informação através do protocolo UART.	—	—
USART1_TX	Dig. Out	Saída série da USART. Usado para transmitir informação através do protocolo USART.	—	—
USART1_RX	Dig. In	Entrada série da USART. Usado para receber informação através do protocolo USART.	—	—
USART1_DE	Dig. Out	da UART. Pino de Hardware Flow Control RS-485 da USART.	—	—
SPI1_MOSI	Dig. Out	Saída série de SPI Master. Usado para transmitir informação através do protocolo SPI.	—	—
SPI1_MISO	Dig. In	Entrada série de SPI Master. Usado para receber informação através do protocolo SPI.	—	—
SPI1_SCLK	Dig. Out	Saída digital de SPI Master. Usado para sincronizar a transmissão entre Slave e Master.	—	—
SPI1_CS	Dig. Out	Saída digital de SPI Master. Usado para identificar a um certo Slave que está ativo.	—	—
LD2 [Green Led]	Dig. In	Entrada digital. Usado para acender o led embutido.	—	0V-3.3V
VDD_PullUp	Dig. Out	Saída digital. Usado para alimentar a maioria dos circuitos pull up.	—	0V-3.3V
BootLoader_Mode	Dig. In	Entrada digital. Identifica se a entra está a 0 ou 1. Quando há um reset do sistema, dependendo do estado do pino, o micro controlador não entra ou entra em modo bootloader, respetivamente	0 - Inativo 1 - Modo BootLoader.	—
BatteryDCState	An. In	Entrada analógica. Usado para verificar o nível de tensão da bateria.	0V-3.3V Resolução 8 bits	—
LidDetection	Dig. In	Entrada digital. Usado para verificar se houve da caixa do sistema.	0 - Caixa fechada 1 - Caixa foi aberta	—

Tabela 4.6: Descrição das Entradas e Saídas do micro Controlador).

Condições	**LPSLEEP**	**LPRUN**	**RUN**
Sem periféricos ativos Portas todas em estado analógico Relógio MSI a 65.536 kHz para processamento	6.6 μA	25 μA	43.4 μA
Sem periféricos ativos Portas configuradas para trabalho Relógio MSI a 65.536 kHz para processamento Relógio HSI a 16 MHz para periféricos de comunicação Sem código a correr	151 μA	171 μA	194 μA
Com periféricos ativos (LPUART1, SPI1, USART1, TIM6, CRC, RTC) Portas configuradas para trabalho Relógio MSI a 65.536 kHz para processamento Relógio HSI a 16 MHz para periféricos de comunicação Sem código a correr	341 μA	341 μA	341 μA
Com periféricos ativos (LPUART1, SPI1, USART1, CRC) Portas configuradas para trabalho Relógio MSI a 65.536 kHz para processamento Relógio HSI a 16 MHz para periféricos de comunicação Código a receber interrupções no pino WS1_HF_EXTI e a contá-los	347 μA	387 μA	-
Com periféricos ativos (LPUART1, SPI1, USART1, TIM6, CRC, RTC) Portas configuradas para trabalho Relógio MSI a 65.536 kHz para processamento Relógio HSI a 16 MHz para periféricos de comunicação Código a abrir e a fechar a válvula	342 μA	75 mA	-
Sem periféricos ativos Portas configuradas para trabalho Relógio MSI a 524.288 kHz para processamento e periféricos de comunicação Sem código a correr	37.5 μA	103.6 μA	126.6 μA
Com periféricos ativos (LPUART1, SPI1, USART1, TIM6, CRC, RTC) Portas configuradas para trabalho Relógio MSI a 524.288 kHz para processamento e periféricos de comunicação Sem código a correr	69.6 μA	135.4 μA	144.7 μA
Com periféricos ativos (LPUART1, SPI1, USART1, CRC) Portas configuradas para trabalho Relógio MSI a 524.288 kHz para processamento e periféricos de comunicação Código a receber interrupções no pino WS1_HF_EXTI e a contá-los	62 μA	98 μA	-
Com periféricos ativos (LPUART1, SPI1, USART1, TIM6, CRC, RTC) Portas configuradas para trabalho Relógio MSI a 524.288 kHz para processamento e periféricos de comunicação Código a abrir e a fechar a válvula	69.9 μA	75 mA	-

Tabela 4.10: Resumo das medições realizadas.

Capítulo 5

Conclusões

O principal objetivo desta dissertação foi a criação de um sistema de baixo consumo, que conseguisse sobreviver durante um período de 10 anos, com apenas uma bateria e que fosse capaz de comunicar através de redes LPWAN. O sistema destina-se à contagem de água, possui até 4 contadores e até 4 válvulas de controlo de corrente.

O primeiro passo para o desenvolvimento deste projeto foi o estudo das soluções LPWANs existentes no mercado, com vista a identificar, entre elas, as mais indicadas para as especificações acima descritas. Das soluções abordadas foram escolhidas a LoRa, o NB-IoT, o LTE-M e o Wi-Fi, tendo sido apenas implementado o módulo de Wi-Fi.

O passo seguinte consistiu na analise das principais características de funcionamento de micro controladores de baixo consumo, que permitissem gerir os gastos energéticos no sistema final.

Tendo-se concluído estes estudos, delinearam-se as características do sistema final, as operações que este iria realizar, a quantidade de vezes que elas seriam realizadas, a forma de poupar o máximo de energia possível e o formato dos dados que seriam enviados pelo sistema. Atendendo-se a que a longevidade do dispositivo está intrinsecamente relacionada com a forma como é montado e com o número de operações de transmissão que realiza, determinou-se quais dos periféricos existentes no micro controlador seriam os mais apropriados para cada tarefa, identificando-se a existência de periféricos similares, sendo uns mais limitados que outros mas que gastam menos energia na execução das suas tarefas.

Com o projeto especificado, delimitaram-se os vários circuitos necessários para a atuação do micro controlador no sistema e integraram-se no PCB que serviu de base para todo o projeto.

Conclui-se que o sistema consegue fazer as tarefas previstas, desde contar água a enviar a informação lida e viver o tempo desejado.

Bibliografia

[1] T. Perkins, "ISM radio bands: Will the concept work in the future?" Dec. 2011. [Online]. Available: http://www.highfrequencyelectronics.com/Dec11/1112_HFE_ism.pdf

[2] F. Gibbs, "ISM bands limit opportunity | EE times," Oct. 2015. [Online]. Available: http://www.eetimes.com/author.asp?section_id=36&doc_id=1328095

[3] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec. 2015, pp. 181–186.

[4] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," vol. 23, Oct. 2016.

[5] L. Labs, "What is SigFox?" Feb. 2015. [Online]. Available: https://www.link-labs.com/what-is-sigfox/

[6] B. Moyer, "Low power, wide area a survey of longer-range IoT wireless protocols." [Online]. Available: http://www.eejournal.com/archives/articles/20150907-lpwa

[7] "TD next • internet of things sigfox modules," 2017. [Online]. Available: http://rfmodules.td-next.com/modules/

[8] "Sigfox developer portal," 2017. [Online]. Available: http://makers.sigfox.com/getting-started/#send-sigfox

[9] S. Corporation, "AN1200.22 LoRa^{TM} modulation basics," May 2015. [Online]. Available: http://www.semtech.com/images/datasheet/an1200.22.pdf

[10] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," vol. PP, no. 99, pp. 1–1, Jan. 2017.

[11] I. Poole, "LoRa RF interface | physical layer | radio-electronics.com," 2016. [Online]. Available: http://www.radio-electronics.com/info/wireless/lora/rf-interface-physical-layer.php

[12] V. Prajzler, "LoRa gateways and concentrators | LORIOT.io," Aug. 2015. [Online]. Available: https://www.loriot.io/lorawan.html

[13] T. Spencer, "How to create a new LoRa network," 2016. [Online]. Available: http://www.thethingsnetwork.org/forum/t/how-to-create-a-new-lora-network/2784/9

[14] Weightless, "Weightless-w - weightless," 2016. [Online]. Available: http://www.weightless.org/about/weightlessw

[15] ——, "Weightless-n - weightless," 2016. [Online]. Available: http://www.weightless.org/about/weightlessn

[16] ——, "Weightless-p - weightless management ltd," 2016. [Online]. Available: http://www.weightless.org/about/weightlessp

[17] ——, "Weightless lpwan hardware officially shipping," 2017. [Online]. Available: http://www.weightless.org/news/LPWAN-Hardware-Officially-Shipping

[18] Weightless SIG, "Weightless Ignition Pack offer," 2017. [Online]. Available: http://try.weightless.org/WIP

[19] "Ubiik launches weightless-p kit - weightless management ltd," 2017. [Online]. Available: http://www.weightless.org/news/ubiik-launches-weightlessp-kit

[20] L. Labs, "A COMPREHENSIVE LOOK AT low power, wide area networks," 2016. [Online]. Available: http://cdn2.hubspot.net/hubfs/427771/LPWAN-Brochure-Interactive.pdf

[21] Myers, "United states patent n0.:7,773,664 b2," 2010. [Online]. Available: https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US7773664.pdf

[22] "RPMA development kit," 2017. [Online]. Available: http://www.ingenu.com/get-started/hardware/rpma-devkit/

[23] "RPMA exploration kit," 2017. [Online]. Available: https://www.ingenu.com/get-started/hardware/exploration-kit/

[24] tutorialspoint, "GPRS - architecture," 2017. [Online]. Available: https://www.tutorialspoint.com/gprs/gprs_architecture.htm

[25] Ian Poole, "GPRS general packet radio service tutorial," 2017. [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/gprs/gprs_tutorial.php

[26] taitradioacademy, "THE DIFFERENCE BETWEEN FDMA AND TDMA," 2017. [Online]. Available: https://www.taitradioacademy.com/topic/the-difference-between-fdma-and-tdma-1/

[27] Nuntius, "Wireless communications solutions general packet radio service (GPRS)," 2017. [Online]. Available: http://www.nuntius.com/solutions22.html

[28] Nokia Networks, "LTE-m – optimizing LTE for the internet of things," 2015. [Online]. Available: https://novotech.com/docs/default-source/default-document-library/lte-m-optimizing-lte-for-the-internet-of-things.pdf?sfvrsn=0

[29] A. Díaz-Zayas, C. A. García-Pérez, A. M. Recio-Pérez, and P. Merino, "3gpp standards to deliver LTE connectivity for IoT," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2016, pp. 283–288.

[30] S. Dawaliby, A. Bradai, and Y. Pousset, "In depth performance evaluation of LTE-m for m2m communications," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2016, pp. 1–8.

[31] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," vol. 3, no. 1, pp. 14–21, Mar. 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2405959517300061

[32] Antti Ratilainen, "NB-IOT," 2017. [Online]. Available: https://www.ietf.org/proceedings/96/slides/slides-96-lpwan-7.pdf

[33] SARA LANDSTRÖM, JOAKIM BERGSTRÖM, ERIK WESTERBERG,, "NB-IOT: A SUSTAINABLE TECHNOLOGY FOR CONNECTING BILLIONS OF DEVICES," vol. 93, 2016. [Online]. Available: https://www.ericsson.com/assets/local/publications/ericsson-technology-review/docs/2016/etr-narrowband-iot.pdf

[34] "Zigbee products | zigbee wireless interfaces and products," 2017. [Online]. Available: https://www.kanda.com/zigbee-wireless.html

[35] Digi, "Demystifying 80 2.15.4 and ZigBee WhitePaper," 2008. [Online]. Available: http://www.mouser.com/pdfdocs/digi-wp_zigbee.pdf

[36] R. W. World, "zigbee tutorial | zigbee protocol stack basics | tutorials," 2016. [Online]. Available: http://www.rfwireless-world.com/Tutorials/Zigbee_tutorial.html

[37] N. Shin, "What is the maximum ZigBee message payload length ... - silicon labs community," 2014. [Online]. Available: http://community.silabs.com/t5/Mesh-Knowledge-Base/What-is-the-maximum-ZigBee-message-payload-length-in-secure-and/ta-p/113417

[38] A. D. Deshmukh and U. B. Shinde, "A low cost environment monitoring system using raspberry pi and arduino with zigbee," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3, Aug. 2016, pp. 1–6.

[39] "XBee module - series 1 - 1mw with wire antenna [XB24-AWI-001] ID: 128 - $22.95 : Adafruit industries, unique & fun DIY electronics and kits," 2017. [Online]. Available: https://www.adafruit.com/product/128

[40] Wifi Alliance, "Wifi Alliance Member Companies," 2017. [Online]. Available: https://www.wi-fi.org/membership/member-companies

[41] aruba Networks, "802.11ac in-depth," 2014. [Online]. Available: http://www.arubanetworks.com/pdf/technology/whitepapers/WP_80211acInDepth.pdf

[42] R. W. World, "WLAN 802.11ac tutorial | 802.11ac basics | tutorials section," 2016. [Online]. Available: http://www.rfwireless-world.com/Tutorials/WLAN-802-11ac-tutorial.html

[43] D. Curto, "Sistemas de deteção por infravermelhos de muito Baixo consumo," 2015. [Online]. Available: http://ria.ua.pt/handle/10773/18663

[44] eembc, "ULPMark scores," 2017. [Online]. Available: http://www.eembc.org/ulpbench/

[45] ——, "Targeting the ultra-low power domain," 2017. [Online]. Available: http://www.eembc.org/ulpbench/about.php

[46] "ULPBench - beyond the data sheets of ultra-low-power MCU," 2015. [Online]. Available: http://www.eembc.org/ulpbench/elektroniknet_article_2015_07_21/

[47] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *2016 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–6.

[48] Itron Water Metering, "Cyble sensor," 2017. [Online]. Available: https://www1.itron.com/local/Sweden%20Product%20Portfolio/Cyble_Sensor_pb_EN_12-11%20(2).pdf

[49] Itron, "Cyble sensor," 2010. [Online]. Available: http://www.compteur-energie.com/media/Itron_CybleSensor_compteur-energie.com.PDF

[50] "Aliexpress.com : Buy 3/4" CWX 15q/N Mini Electric Ball Valve Stainless Steel Control Type CR01/CR02/CR05/12v from Reliable 3 wire motor control suppliers on YUEQING IDEA ELECTRICAL COMPANY," 2017. [Online]. Available: https://www.aliexpress.com/store/product/3-4-CWX-15Q-N-Mini-Electric-Ball-Valve-Stainless-Steel-CR01-CR02-or-CR05-Control/609688_599977704.html

[51] Modbus-IDA, "MODBUS APPLICATION ROTOCOL SPECIFICATION 1.1b," 2017. [Online]. Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

[52] "STM32l053r8 - ultra-low-power ARM cortex-m0+ MCU with 64 kbytes flash, 32 MHz CPU, USB, LCD - STMicroelectronics," 2017. [Online]. Available: http://www.st.com/en/microcontrollers/stm32l053r8.html

[53] "TYPE_abz | PDF | Murata Manufacturing," 2017. [Online]. Available: http://wireless.murata.com/datasheet?/RFM/data/type_abz.pdf

[54] "GSM Click Add on Board based on GL865-QUAD GSM/GPRS Module," 2017. [Online]. Available: https://www.element14.com/community/docs/DOC-70203/l/gsm-click-add-on-board-based-on-gl865-quad-gsmgprs-module

[55] Telit, "GL865 Hardware User Guide," 2017. [Online]. Available: http://www.telit.com/fileadmin/user_upload/products/Downloads/2G/CL865/1vv0300910_GL865_Hardware_User_Guide_r9.pdf

[56] ——, "GE866-QUAD Hardware User Guide," 2017. [Online]. Available: http://www.lte.com.tr/uploads/pdfc/65.pdf

[57] "SARA-N2 series," Jun. 2016. [Online]. Available: https://www.u-blox.com/en/product/sara-n2-series

[58] ST, "SPWF01sa, SPWF01sc," 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/datasheet/ba/8c/b2/64/02/bc/4e/05/DM00102124.pdf/files/DM00102124.pdf/jcr:content/translations/en.DM00102124.pdf

[59] ——, "SPWF01sx power management options," 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/application_note/group0/b5/ff/16/5f/23/f1/4b/2f/DM00184377/files/DM00184377.pdf/jcr:content/translations/en.DM00184377.pdf

[60] Eve Energy, "ER34615m Lithium-thionyl Chloride Spiral (Li-SOCl2) Battery," 2017. [Online]. Available: http://www.farnell.com/datasheets/1445897.pdf?_ga=2.147426815.958032001.1501661203-631435005.1496397080

[61] mpoweruk, "State of charge (soc) determination," 2017. [Online]. Available: http://www.mpoweruk.com/soc.htm

[62] ST, "X-nucleo-lpm01a," 2017. [Online]. Available: http://www.st.com/en/evaluation-tools/x-nucleo-lpm01a.html

[63] Omron, "SS-P Subminiature Basic Switch," 2017. [Online]. Available: http://www.mouser.com/ds/2/307/en-ss_p-230833.pdf

[64] ST, "ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32," 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/user_manual/65/e0/44/72/9e/34/41/8d/DM00026748.pdf/files/DM00026748.pdf/jcr:content/translations/en.DM00026748.pdf

[65] ——, "STM32 Nucleo-64 board," 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf

[66] ——, "Optimized usage of the dual bank structure of Flash memory in STM32 microcontrollers - Soft ware expansion for STM32cube," 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/application_note/group0/ab/6a/0f/b7/1a/84/40/c3/DM00230416/files/DM00230416.pdf/jcr:content/translations/en.DM00230416.pdf

[67] adesto Technologies, "SPI Serial Flash Memory with Dual-I/O and Quad-I/O Support," Jul. 2017. [Online]. Available: https://www.adestotech.com/wp-content/uploads/DS-AT25SF041_044.pdf

[68] mouser, "AT25sf041," 2017. [Online]. Available: http://www.mouser.com/ds/2/590/DS-AT25SF041_044-534001.pdf

[69] P. Rodrigues, "Portugal gasta 220 litros de água por pessoa por dia," 2013. [Online]. Available: https://www.publico.pt/2013/03/22/ecosfera/noticia/portugal-gasta-220-litros-de-agua-por-pessoa-por-dia-1588798

[70] (2016) PCB 90 degree angles [duplicate]. [Online]. Available: https://electronics.stackexchange.com/questions/226582/pcb-90-degree-angles

[71] Mouser, "1591xxds enclosure," 2017. [Online]. Available: http://www.mouser.com/ds/2/177/1591XXDS-747184.pdf

[72] Saturn, "Saturn PCB design toolkit," 2017. [Online]. Available: http://www.saturnpcb.com/pcb_toolkit.htm

[73] ST, "Stm32cubemx," 2017. [Online]. Available: http://www.st.com/en/development-tools/stm32cubemx.html

Apêndice A

Desenhos do Altium

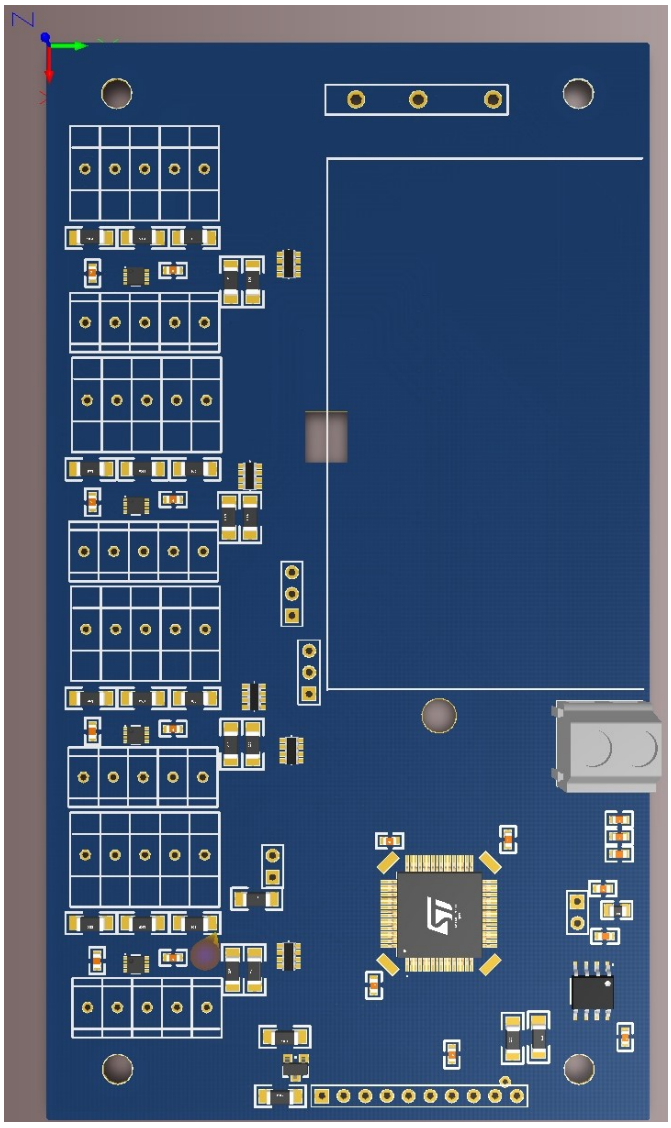


Figura A.1: Modelo 3D do PCB principal no Altium.

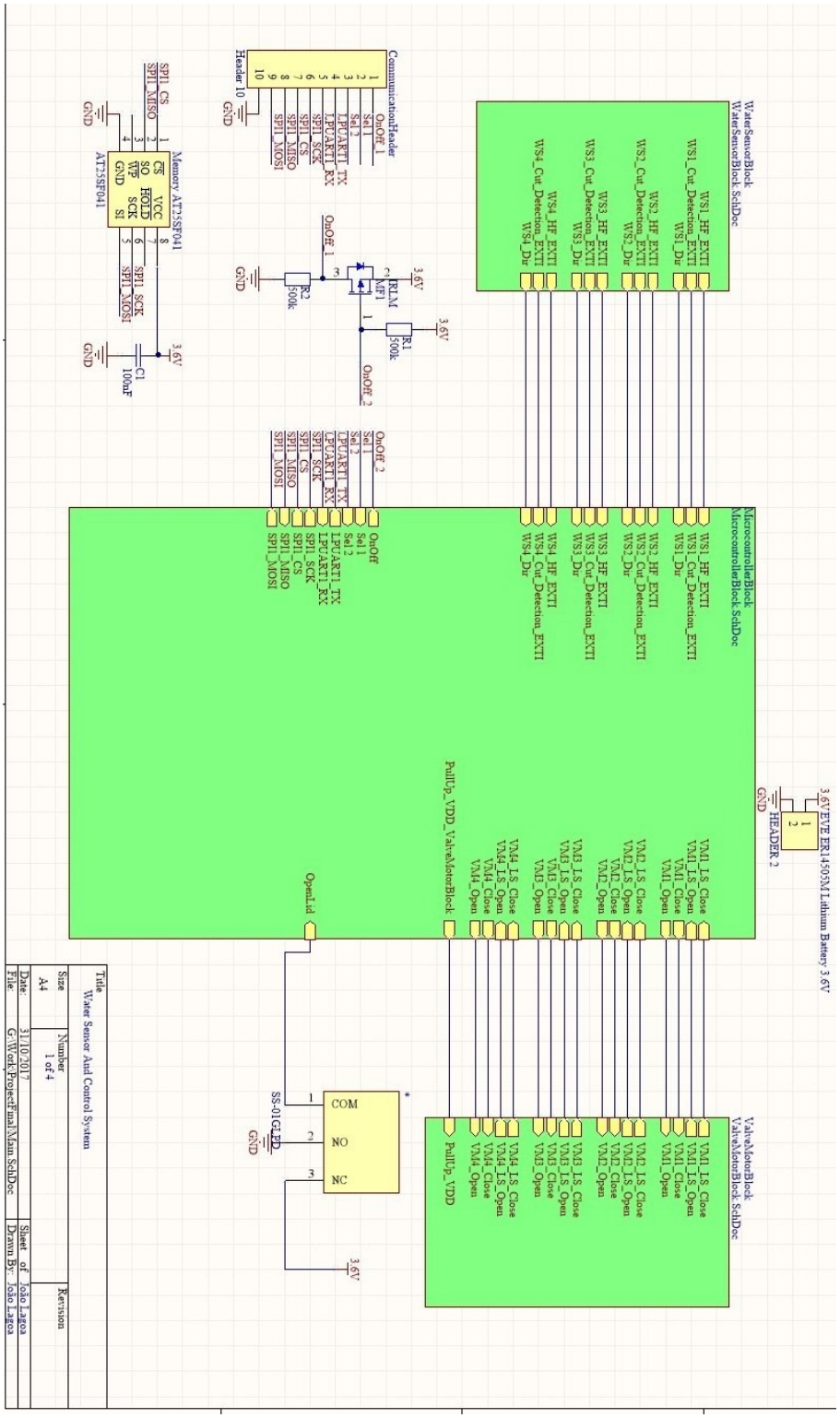


Figura A.2: Desenho de blocos principal do sistema no Altium.

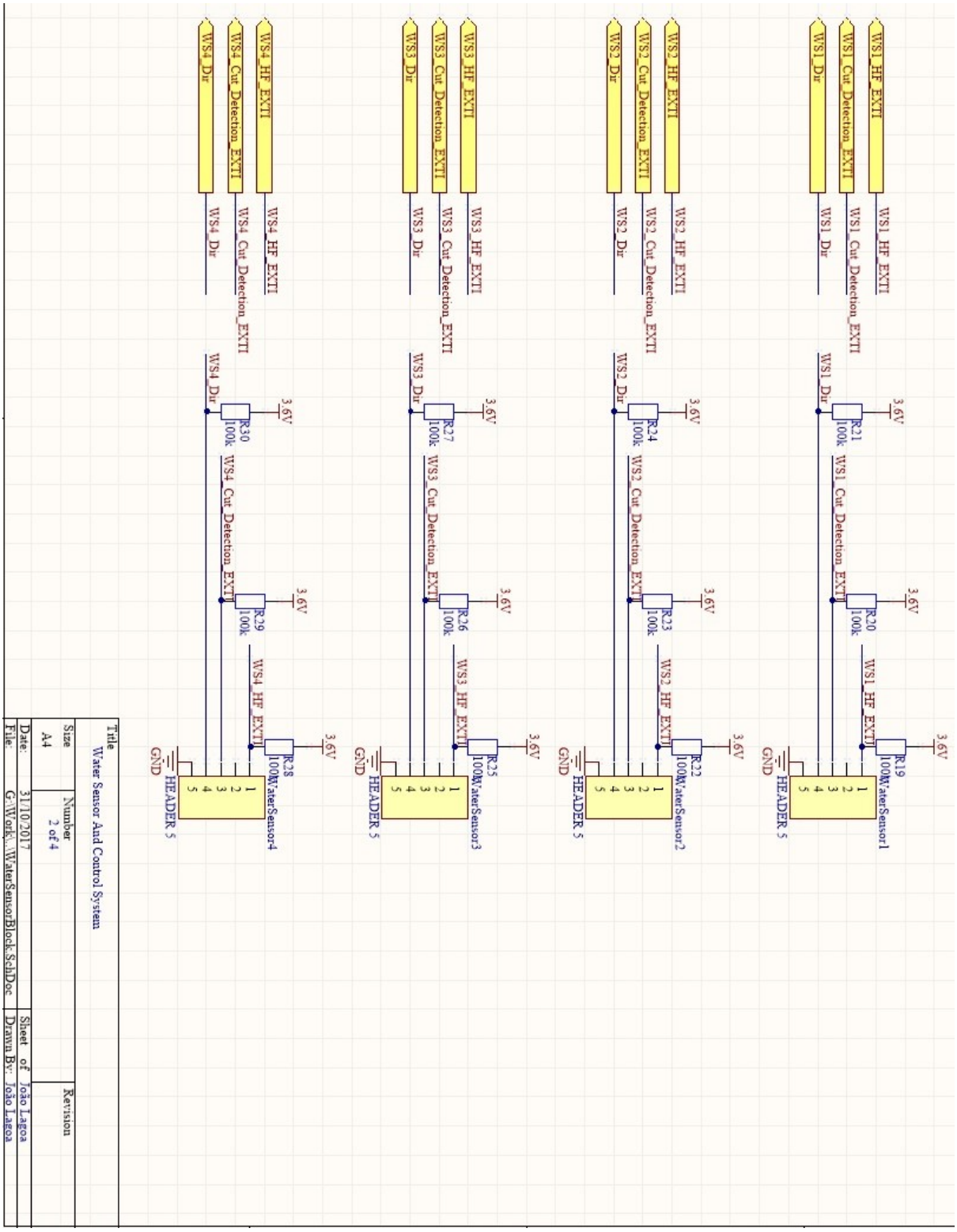


Figura A.3: Desenho de blocos do contador de água no Altium.

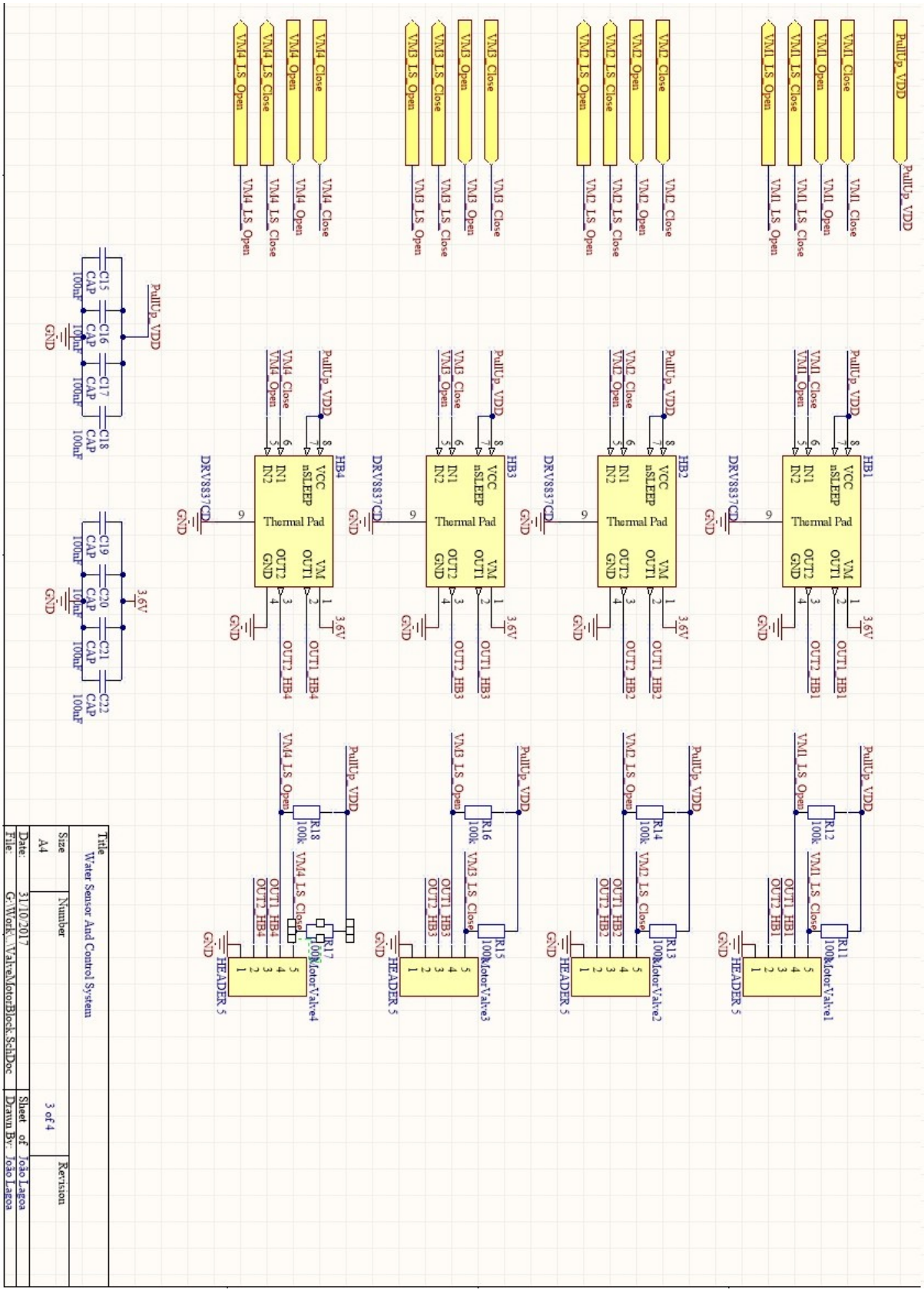


Figura A.4: Desenho de blocos do contador do motor no Altium.

Figura A.5: Desenho de blocos do contador do micro controlador no Altium.

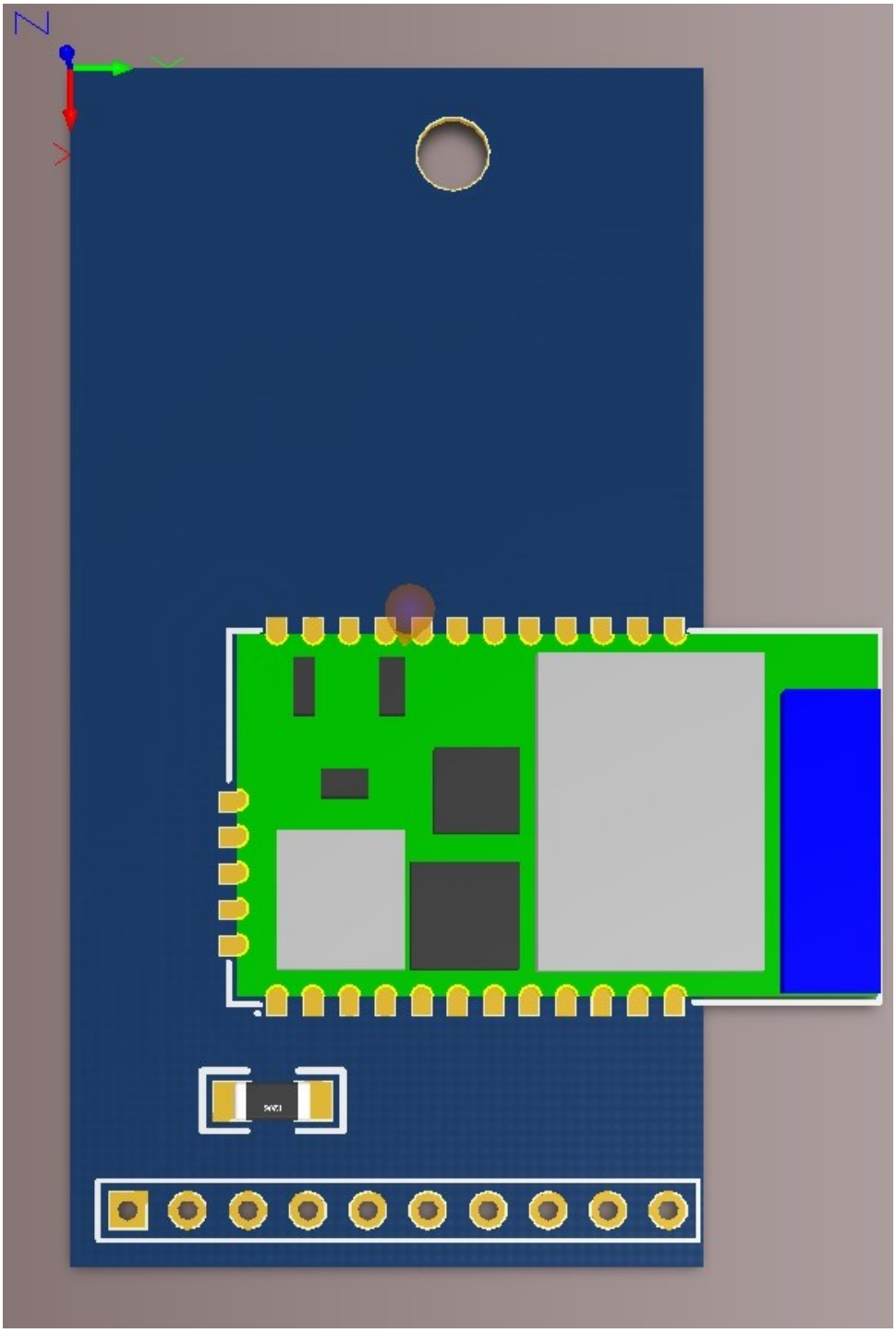


Figura A.6: PCB 3D do módulo SPWF015A.

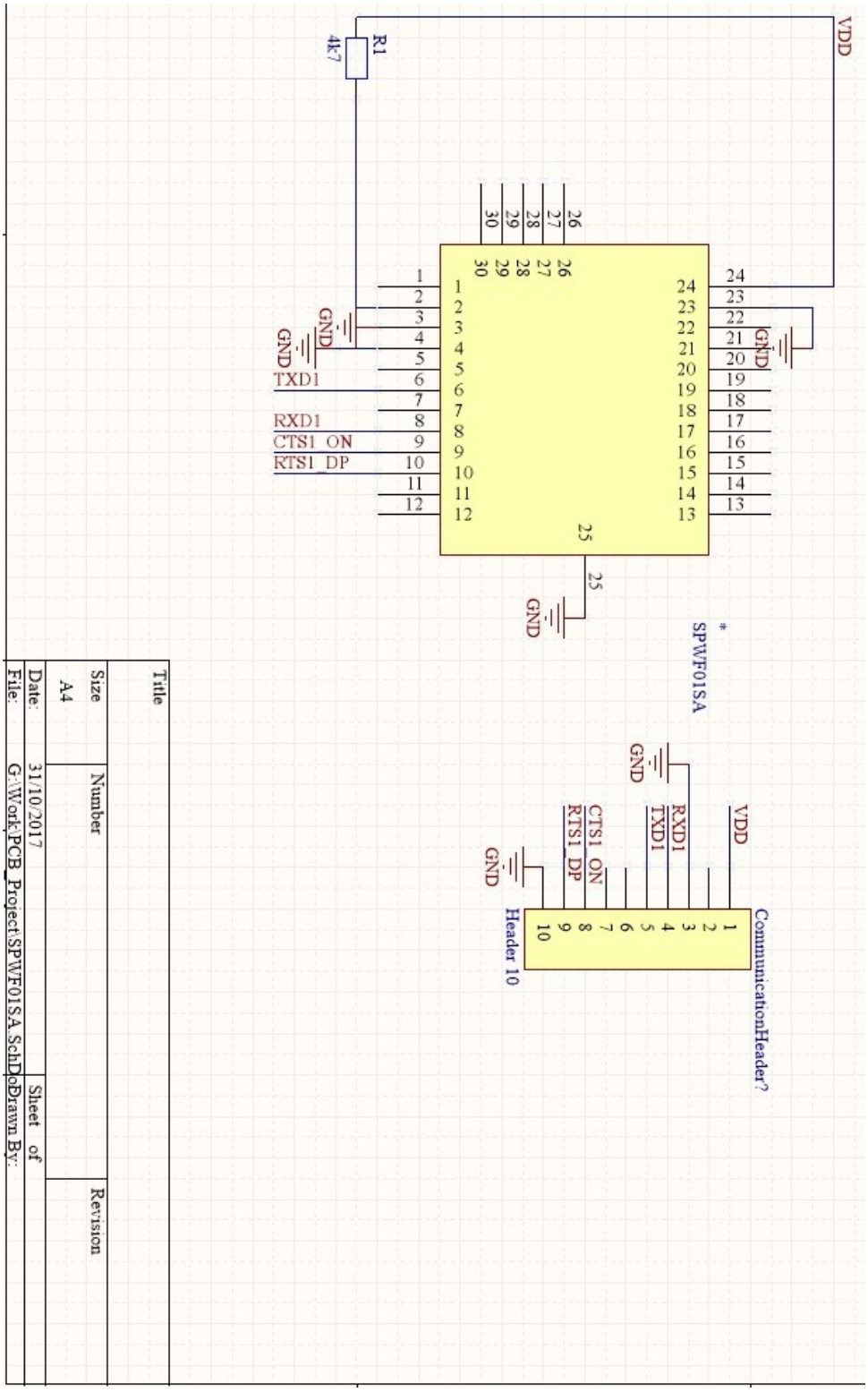


Figura A.7: Desenho de blocos do PCB SPWF015A

Apêndice B

Cabeçalho das Funções Implementas

```
//Contador de agua

extern uint8_t allDirState; //0 D3 0 D2 0 D1 0 D0, em que Dx
                            //corresponde ao estado de direcao de
                            //agua de cada valvula.

uint8_t allCutDetectState; // 0 CD3 0 CD2 0 CD1 0 CD0, em que CDx
                           // corresponde ao estado de corte de fio
                           // de cada valvula.

extern uint16_t counterHFUp; // Contagem ascendente do numero de
                             // impulsos de HF, independente da
                             // direcao.

extern uint16_t counterHFUpDown; // Contagem ascendente (quando dir
                                 // = 0) e descendente (quando dir
                                 // = 1) do numero de impulsos de
                                 // HF.

void waterSensorReadDir(void); // Le a direcao da agua de todos
                               // os sensores.

boolean resetCounter(uint8_t sensor, uint8_t counter);
                     // sensor recebe WS0-WS3; counter recebe 0-2,
                     // 0 - poem a zero o contado ascendente;
                     // 1 - poem a zero o contado que sobe e desce;
                     // 2 - poem a zero os dois contadores;
                     // retorna false se sensor ou counter nao
                     // pertencerem ao conjunto esperado.

void resetAllCounters(void); // Apaga todos os contadores.

// Valvula

extern uint8_t allLSState; // LSO0 LSO1 LSO2 LSO3 LSC0 LSC1 LSC2
                           // LSC3, em que LSOx e LSCx correspondem
```

```
                                // de aos estados dos sensores de fim de
                                // curso abertos e fechados,
                                // respetivamente.

extern uint8_t allLSState; // LSO0 LSO1 LSO2 LSO3 LSC0 LSC1 LSC2
                                // LSC3, em que LSOx e LSCx correspondem
                                // de aos estados dos sensores de fim de
                                // curso abertos e fechados,
                                // respetivamente.

void valveMotorReadLS(void); // Faz a update da informacao de estado
                                // dos sensores de fim de curso da
                                // valvula.

boolean valveMotorClose(uint8_t valve);
                                // Fecha a valvula; valve recebe VM0-
                                // VM3; retorna false se valve nao
                                // pertencer ao conjunto esperado.

boolean valveMotorOpen(uint8_t valve);
                                // Abre a valvula; valve recebe VM0-
                                // VM3; retorna false se valve nao
                                // pertencer ao conjunto esperado.

boolean valveMotorStop(uint8_t valve);
                                // Para o movimento da a valvula; valve
                                // recebe VM0-VM3; retorna false se
                                // valve nao pertencer ao conjunto
                                // esperado.

// Micro Controlador

void GPIOPinInit(void);         // Inicia os estados dos pinos do micro
                                // controlador.

void goLPRUNMode(void);         // O micro controlador entra modo
                                // low power run.

void exitLPRUNMode(void);       // O micro controlador sai do modo
                                // low power run.

void goLPSLEEPMode(void);       // O micro controlador entra modo
                                // low power sleep.

// Circuitos Pull Up

PU_On();                        // Alimenta os circuitos Pull Up.

PU_Off();                       // Desliga os circuitos Pull Up.
```

```
// LED

LED_On();                       // Liga o LED.

LED_Off();                      // Desliga o LED.


// Modulo de Comunicacao

extern uint8_t mod;             // Modulo alocado ao sistema.

void communicationModFinder(void)
                                // Determina qual o modulo de
                                // comunicacao ao sistema.

void turnOnMod(void);           // Liga o modulo de comunicacao.

void turnOffMod(void);          // Desliga o modulo de comunicacao.


//LPUART

void MX_LPUART1_UART_Init(void);
                                // Inicia as configuracoes da LPUART.

void MX_UART_MspInit(UART_HandleTypeDef* uartHandle);
                                // Inicia os pinos usados na
                                // comunicacao.

int8_t LPUARTITTransmit(UART_HandleTypeDef* uartHandle,
uint8_t *aTxBuffer, uint16_t TXBUFFERSIZE);
                                // Transmite dados atraves do protocolo
                                // UART. Retorna RESET ou SET consoante
                                // o estado de transmissao. Retorna -1
                                // se houve algum problema com
                                // informacao transmitida nos
                                // argumentos.

int8_t LPUARTITReceive(UART_HandleTypeDef* uartHandle,
uint8_t *aRxBuffer, uint16_t RXBUFFERSIZE);
                                // Recebe dados atraves do protocolo
                                // UART. Retorna RESET ou SET consoante
                                // o estado de transmissao. Retorna -1
                                // se houve algum problema com
                                // informacao transmitida nos
                                // argumentos.
```

```
void HAL_UART_TxCpltCallback(UART_HandleTypeDef* uartHandle);
                                // Funcao de callback chamada no fim
                                // de uma transmissao.

void HAL_UART_RxCpltCallback(UART_HandleTypeDef* uartHandle);
                                // Funcao de callback chamada no fim
                                // de uma rececao.

// USART2

void MX_USART2_Init(void);      // Inicia as configuracoes da USART2.

void MX_UART_MspInit(UART_HandleTypeDef* uartHandle);
                                // Inicia os pinos usados na
                                // comunicacao.

int8_t LPUARTITTransmit(UART_HandleTypeDef* uartHandle,
uint8_t *aTxBuffer, uint16_t TXBUFFERSIZE);
                                // Transmite dados atraves do protocolo
                                // UART. Retorna RESET ou SET consoante
                                // o estado de transmissao. Retorna -1
                                // se houve algum problema com
                                // informacao transmitida nos
                                // argumentos.

int8_t LPUARTITReceive(UART_HandleTypeDef* uartHandle,
uint8_t *aRxBuffer, uint16_t RXBUFFERSIZE));
                                // Recebe dados atraves do protocolo.
                                // UART Retorna RESET ou SET consoante
                                // o estado  de transmissao. Retorna -1
                                // se houve algum  problema com
                                // informacao transmitida nos
                                // argumentos.

void HAL_UART_TxCpltCallback(UART_HandleTypeDef* uartHandle);
                                // Funcao de callback chamada no fim
                                // de uma transmissao.

void HAL_UART_RxCpltCallback(UART_HandleTypeDef* uartHandle);
                                // Funcao de callback chamada no fim
                                // de uma rececao.


// SPI

void MX_SPI1_Init(void);        // Inicia a configuracao do
                                // periferico SPI.

void MX_SPI1_MspInit(void);     // Inicia os pinos usados pelo
```

```
                                // periferico SPI.

void SPITransmitReceive(SPI_HandleTypeDef* spiHandle, uint8_t
*aTxBuffer, uint8_t *aRxBuffer, uint16_t BUFFERSIZE,
uint32_t timeout);
                                // Transmite/Recebe dados atraves
                                // do protocolo SPI1.

void SPI1_On_CS(void);          // Liga o Chip Selector.

void SPI1_Off_CS(void);      // Desliga o Chip Selector.


// RTC

void RTC_AlarmConfig(void);   // Inicia as variaveis necessarios
                                // para o RTC.

void HAL_RTC_AlarmAEventCallback(RTC_HandleTypeDef *hrtc)
                                // Funcao de callback chamada
                                // quando ocorre um alarme.


// Water Sensor Manager

void waterSensorManager_Init(void);
                                // Inicia as variaveis de Dir e Cut
                                // Detection; Faz a verificacao delas
                                // para garantir que nao ha ja algum
                                // problema no sistema.


// Valve Motor Manager()

void valveMotorManager_Init(void);
                                // Inicia as variavel dos
                                // sensores de fim de curso.

void valveMotorManager_OpenCompletely(void);
                                // Abre a valvula completamente.

void valveMotorManager_CloseCompletely(void);
                                // Fecha a valvula completamente.
```

Apêndice C

Código Usado na Medição do Modo Run do Sistema

```
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32l0xx_hal.h"
#include "crc.h"
#include "usart.h"
#include "rtc.h"
#include "spi.h"
#include "tim.h"
#include "gpio.h"

/* USER CODE BEGIN Includes */
#include "stm32l0xx_hal_conf.h"
#include "stm32l0xx_it.h"
#include <stdio.h>
#include <string.h>

#include "moduleControl.h"
#include "PWR_library.h"
#include "LogicLevel.h"
/* USER CODE END Includes */

/* Private variables ---------------------------------------------------------*/

/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
UART_HandleTypeDef hlpuart1;
TIM_HandleTypeDef htim6;

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
```

```
static void MX_NVIC_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------------*/

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */


/* USER CODE END 0 */

int main(void)
{

  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  // Reset of all peripherals, Initializes the
  //Flash interface and the Systick.
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LPUART1_UART_Init();
  MX_SPI1_Init();
  MX_USART1_UART_Init();
  //MX_TIM6_Init();
  MX_CRC_Init();
  //MX_RTC_Init();

  /* Initialize interrupts */
  MX_NVIC_Init();

  /* USER CODE BEGIN 2 */
```

```
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */

}

/** System Clock Configuration
*/
void SystemClock_Config(void)
{

  RCC_OscInitTypeDef RCC_OscInitStruct;
  RCC_ClkInitTypeDef RCC_ClkInitStruct;
  RCC_PeriphCLKInitTypeDef PeriphClkInit;

    /**Configure the main internal regulator output voltage
    */
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI
                                   |RCC_OSCILLATORTYPE_LSI
                                   |RCC_OSCILLATORTYPE_MSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = 16;
  RCC_OscInitStruct.LSIState = RCC_LSI_ON;
  RCC_OscInitStruct.MSIState = RCC_MSI_ON;
  RCC_OscInitStruct.MSICalibrationValue = 0;
  RCC_OscInitStruct.MSIClockRange = RCC_MSIRANGE_6;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK
```

```
                                    |RCC_CLOCKTYPE_SYSCLK
                                    |RCC_CLOCKTYPE_PCLK1
                                    |RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_MSI;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0)
  != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

  PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1
                                    |RCC_PERIPHCLK_LPUART1
                                    |RCC_PERIPHCLK_RTC;
  PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_HSI;
  PeriphClkInit.Lpuart1ClockSelection = RCC_LPUART1CLKSOURCE_HSI;
  PeriphClkInit.RTCClockSelection = RCC_RTCCLKSOURCE_LSI;
  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /** Configure the Systick interrupt time
    */
  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /** Configure the Systick
    */
  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

  /* SysTick_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/** NVIC Configuration
*/
static void MX_NVIC_Init(void)
{
  /* PVD_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(PVD_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(PVD_IRQn);
  /* RTC_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(RTC_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(RTC_IRQn);
  /* FLASH_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(FLASH_IRQn, 0, 0);
```

```
  HAL_NVIC_EnableIRQ(FLASH_IRQn);
  /* RCC_CRS_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(RCC_CRS_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(RCC_CRS_IRQn);
  /* EXTI0_1_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(EXTI0_1_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI0_1_IRQn);
  /* EXTI4_15_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(EXTI4_15_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI4_15_IRQn);
  /* TIM6_DAC_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(TIM6_DAC_IRQn, 1, 0);
  HAL_NVIC_EnableIRQ(TIM6_DAC_IRQn);
  /* SPI1_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SPI1_IRQn, 1, 0);
  HAL_NVIC_EnableIRQ(SPI1_IRQn);
  /* USART1_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(USART1_IRQn, 1, 0);
  HAL_NVIC_EnableIRQ(USART1_IRQn);
  /* AES_RNG_LPUART1_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(AES_RNG_LPUART1_IRQn, 1, 0);
  HAL_NVIC_EnableIRQ(AES_RNG_LPUART1_IRQn);
}
```

Apêndice D

Código Usado na Medição do Modo de LPRUN

```
int main(void)
{

  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  // Reset of all peripherals,
  //Initializes the Flash interface and the Systick.
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LPUART1_UART_Init();
  MX_SPI1_Init();
  MX_USART1_UART_Init();
  //MX_TIM6_Init();
  MX_CRC_Init();
  //MX_RTC_Init();
```

```
  /* Initialize interrupts */
  MX_NVIC_Init();

  /* USER CODE BEGIN 2 */

   goLPRUNMode();
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */

}

 /**
* @brief  Turn up low power run mode
* @param  None
* @retval None
*/
 void goLPRUNMode(void)
{
        /* Enter LP RUN mode */
    HAL_PWREx_EnableLowPowerRunMode();

    // Wait until the system enters LP RUN
    //and the Regulator is in LP mode
    while(__HAL_PWR_GET_FLAG(PWR_FLAG_REGLP) == RESET)
    {
    }
}

 /**
* @brief  Exit low power run mode
* @param  None
* @retval None
*/
 void exitLPRUNMode(void)
{

    /* Exit LP RUN mode */
```

```
    HAL_PWREx_DisableLowPowerRunMode();

    // Wait until the system exits LP RUN
    //and the Regulator is in main mode
    while(__HAL_PWR_GET_FLAG(PWR_FLAG_REGLP) != RESET)
    {
    }
}
```

Apêndice E

Código Usado na Medição do Modo de LPSLEEP

```
int main(void)
{

  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  // Reset of all peripherals,
  //Initializes the Flash interface and the Systick.
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LPUART1_UART_Init();
  MX_SPI1_Init();
  MX_USART1_UART_Init();
  //MX_TIM6_Init();
  MX_CRC_Init();
  //MX_RTC_Init();
```

```
  /* Initialize interrupts */
  MX_NVIC_Init();

  /* USER CODE BEGIN 2 */

   goLPSLEEPMode();
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */

}

 /**
* @brief  Turn up low power sleep mode
                  If any interrupt occurs it leaves this mode
* @param  None
* @retval None
*/
void goLPSLEEPMode(void)
{
    /* Enable the power down mode during Sleep mode */
    __HAL_FLASH_SLEEP_POWERDOWN_ENABLE();

      //Suspend Tick increment to prevent
      //wakeup by Systick interrupt.
      //Otherwise the Systick interrupt will wake
      //up the device within 1ms (HAL time base)
    HAL_SuspendTick();

    // Enter Sleep Mode , wake up is
    //done once Key push button is pressed
    HAL_PWR_EnterSLEEPMode(PWR_LOWPOWERREGULATOR_ON,
    PWR_SLEEPENTRY_WFI);


    /* Resume Tick interrupt if disabled prior to sleep mode entry*/
    HAL_ResumeTick();
}
```

Apêndice F

Código Usado na Medição do Consumo da Válvula

```
int main(void)
{

  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */


  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LPUART1_UART_Init();
  MX_SPI1_Init();
  MX_USART1_UART_Init();
  //MX_TIM6_Init();
  MX_CRC_Init();
  MX_RTC_Init();

  /* Initialize interrupts */
  MX_NVIC_Init();
```

```
  /* USER CODE BEGIN 2 */

  uint8_t tmp= 1;
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */

  while (1)
  {
          goLPSLEEPMode();
              goLPRUNMode();

            if(flagRTC == 1)
            {
                    HAL_GPIO_WritePin(VDD_PullUp_GPIO_Port,
                    VDD_PullUp_Pin, SET);
                    wait(500);
                    if(tmp == 1)
                    {
                            openValve(VM1);
                            while(checkLSOpen(VM1) != 0);
                            stopValve(VM1);
                            tmp = 0;
                    }
                    else if(tmp == 0)
                    {
                            closeValve(VM1);
                            while(checkLSClose(VM1) != 0);
                            stopValve(VM1);
                            tmp = 1;
                    }
                    HAL_GPIO_WritePin(VDD_PullUp_GPIO_Port,
                    VDD_PullUp_Pin, RESET);

            flagRTC = 0;
            }
            exitLPRUNMode();

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */

}


/**
```

```
  * @brief Change valve to open
  * @param valve: VM1, VM2, VM3, VM4 to represent each valve
  * @retval state: state = 1, pin was written; state = 0, pin
  * was not written, wrong valve value
  */
uint8_t openValve(uint8_t valve)
{
        uint8_t state;
        switch(valve)
        {
                case VM1:
                        HAL_GPIO_WritePin(VM1_Open_Blue_GPIO_Port,
                        VM1_Open_Blue_Pin, GPIO_PIN_SET);
                        HAL_GPIO_WritePin(VM1_Close_Yellow_GPIO_Port,
                        VM1_Close_Yellow_Pin, GPIO_PIN_RESET);
                        state = 1;
                        break;
                default:
                        state = 0;
                        break;
        }
        return state;
}

/**
 * @brief Change valve to close
 * @param valve: VM1, VM2, VM3, VM4 to represent each valve
 * @retval state: state = 1, pin was written; state = 0,
 * pin was not written, wrong valve value
 */
uint8_t closeValve(uint8_t valve)
{
        uint8_t state;
        switch(valve)
        {
                case VM1:
                        HAL_GPIO_WritePin(VM1_Open_Blue_GPIO_Port,
                        VM1_Open_Blue_Pin, GPIO_PIN_RESET);
                        HAL_GPIO_WritePin(VM1_Close_Yellow_GPIO_Port,
                        VM1_Close_Yellow_Pin, GPIO_PIN_SET);
                        state = 1;
                        break;
                default:
                        state = 0;
                        break;
        }
        return state;
}
```

```
/**
 * @brief Stop valve movement
 * @param valve: VM1, VM2, VM3, VM4 to represent each valve
 * @retval state: state = 1, pin was written; state = 0,
 * pin was not written, wrong valve value
 */
uint8_t stopValve(uint8_t valve)
{
        uint8_t state;
        switch(valve)
        {
                case VM1:
                        HAL_GPIO_WritePin(VM1_Open_Blue_GPIO_Port,
                        VM1_Open_Blue_Pin, GPIO_PIN_RESET);
                        HAL_GPIO_WritePin(VM1_Close_Yellow_GPIO_Port,
                        VM1_Close_Yellow_Pin, GPIO_PIN_RESET);
                        state = 1;
                        break;
                default:
                        state = 0;
                        break;
        }
        return state;
}

/**
 * @brief Stop valve movement
 * @param valve: VM1, VM2, VM3, VM4 to represent each valve
 * @retval state: state of LS sensor, if state = 0,
 * valve is completely open; if state = 1, valve state is unknown;
 * if state = -1, valve select error
 */
GPIO_PinState checkLSOpen(uint8_t valve)
{
        GPIO_PinState state;
        switch(valve)
        {
         case VM1:
         state = HAL_GPIO_ReadPin(VM1_ReadLS_Open_GPIO_Port,
         VM1_ReadLS_Open_Pin);
            break;
                default:
                        state = -1;
                        break;
        }
        return state;
}
```

```
/**
 * @brief Stop valve movement
 * @param valve: VM1, VM2, VM3, VM4 to represent each valve
 * @retval state: state of LS sensor, if state = 0,
 * valve is completely close; if state = 1, valve state is unknown;
 * if state = -1, valve select error
 */
GPIO_PinState checkLSClose(uint8_t valve)
{
        GPIO_PinState state;
        switch(valve)
        {
        case VM1:
 state = HAL_GPIO_ReadPin(VM1_ReadLS_Close_GPIO_Port,
 VM1_ReadLS_Close_Pin);
 break;
                default:
                        state = -1;
                        break;
        }
        return state;
}
```

Apêndice G

Código Usado na Medição do Consumo do Módulo

```
void testcodeComm(void)
{

        turnUpPower();
        wait(500);
        if(moduleDetect()== 0)
        {

          timersInit();
          portSerialEnable( 1, 0 );
          portTimersEnable( );
          strcpy(internalBuffer3, "WiFi Up");
          while(flagEmpty!=1);
          wait(500);

          flagEmpty = 0;
          strcpy(internalBuffer3, "OK");
          uint8_t aTxBuffer1[] =
          "AT+S.SOCKON=cd.exatronic.io,6006 ,t\r";
          UARTITTransmit(&hlpuart1,aTxBuffer1,strlen(aTxBuffer1));
          while(UartReady == SET);
          timersInit();
          portSerialEnable( 1, 0 );
          portTimersEnable( );
          while(flagEmpty!=1);

          flagEmpty = 0;
          strcpy(internalBuffer3, "OK");
          uint8_t aTxBuffer2[]= "AT+S.SOCKW=00,19\r";
          uint8_t aTxBuffer3[]= "IDCT01CT02CT03CT04E\r";
          UARTITTransmit(&hlpuart1,aTxBuffer2,strlen(aTxBuffer2));
          while(UartReady == SET);
          UARTITTransmit(&hlpuart1,aTxBuffer3,strlen(aTxBuffer3));
```

```
            while(UartReady == SET);
            timersInit();
            portSerialEnable( 1, 0 );
            portTimersEnable( );
            while(flagEmpty!=1);

            flagEmpty = 0;
            strcpy(internalBuffer3, "OK");
            uint8_t aTxBuffer4[]= "AT+S.SOCKQ=00\r";
            UARTITTransmit(&hlpuart1,aTxBuffer4,strlen(aTxBuffer4));
            while(UartReady == SET);
            timersInit();
            portSerialEnable( 1, 0 );
            portTimersEnable( );
            while(flagEmpty!=1);
            flagEmpty = 0;
            cutPower();
        }
}
```

Apêndice H

Tabela Modbus

StartAddress	StartAdressHEX	# registers	Permition	Description	Notes	Default value	NVM
0	0	8	Read only	IMEI/Identificação		N/A	NO
8	8	1	Read only	Type Device	7 - SMART	N/A	YES
9	9	2	Read-Write	Firmware version	First word Version second Word Revision	N/A	YES
11	B	3	Read	Field Time	Tempo de existência em trabalho; (Primeira palavra, anos; Segunda palavra, meses; Primeira palavra, dias)		
14	E	1	Read-Write	TimeAwake	Tempo para o controlador ficar acordado em minutos	5	YES
15	F	1	Read-Write	TimeAsleep	Tempo para o controlador ficar a dormir em minutos	55	YES
16	10	6	Read-Write	RealTimeClock Fix	Update da informação RTC pelo server	0	
22	16	2	Read-Write	StartTime	Indicação do inicio do ciclo de contagem(Horas; Minutos)	0	
24	18	1	Read-Write	Water Sensor/Motor Valve Existence	WS1 WS2 WS3 WS4 MV1 MV2 MV3 MV4 (0-Não Montado; 1-Montado)	0	
25	19	2	Read	Up Counter 1	Conta os impulsos lidos do contador, independemente do estado de Dir 1	0	
27	1B	2	Read	Up Counter 2	Conta os impulsos lidos do contador, independemente do estado de Dir 2	0	
29	1D	2	Read	Up Counter 3	Conta os impulsos lidos do contador, independemente do estado de Dir 3	0	
31	1F	2	Read	Up Counter 4	Conta os impulsos lidos do contador, independemente do estado de Dir 4	0	
33	21	2	Read	UpDown Counter 1	Sobe ou desce consoante o estado de Dir 1	0	
35	23	2	Read	UpDown Counter 2	Sobe ou desce consoante o estado de Dir 2	0	
37	25	2	Read	UpDown Counter 3	Sobe ou desce consoante o estado de Dir 3	0	
39	27	2	Read	UpDown Counter 4	Sobe ou desce consoante o estado de Dir 4	0	
41	29	1	Read	Dir/Cut Detection	0D1 0D2 0D3 0D4 (0-Normal;1-Invertido) 0CD1 0CD2 0CD3 0CD4 (0-Normal;1-Corte)	0	
42	2A	1	Read	LS Sensor Open/Close	0LSO1 0LSO2 0LSO3 0LSO4 (0-Aberto;1-Desconhecido) 0LSC1 0LSC2 0LSC3 0LSC4 (0-Fechado;1-Desconhecido)	0	
43	2B	4	Read	0H-1H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
47	2F	4	Read	1H-2H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
51	33	4	Read	2H-3H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
55	37	4	Read	3H-4H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
59	3B	4	Read	4H-5H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
63	3F	4	Read	5H-6H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
67	43	4	Read	6H-7H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
71	47	4	Read	7H-8H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
75	4B	4	Read	8H-9H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
79	4F	4	Read	9H-10H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
83	53	4	Read	10H-11H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
87	57	4	Read	11H-12H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
91	5B	4	Read	12H-13H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
95	5F	4	Read	13H-14H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
99	63	4	Read	14H-15H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
103	67	4	Read	15H-16H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
107	6B	4	Read	16H-17H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
111	6F	4	Read	17H-18H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
115	73	4	Read	18H-19H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
119	77	4	Read	19H-20H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
123	7B	4	Read	20H-21H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
127	7F	4	Read	21H-22H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
131	83	4	Read	22H-23H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
135	87	4	Read	23H-0H Count1	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)	0	
139	8B	4	Read	0H-1H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
143	8F	4	Read	1H-2H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
147	93	4	Read	2H-3H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
151	97	4	Read	3H-4H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
155	9B	4	Read	4H-5H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
159	9F	4	Read	5H-6H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
163	A3	4	Read	6H-7H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
167	A7	4	Read	7H-8H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
171	AB	4	Read	8H-9H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
175	AF	4	Read	9H-10H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
179	B3	4	Read	10H-11H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
183	B7	4	Read	11H-12H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
187	BB	4	Read	12H-13H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
191	BF	4	Read	13H-14H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
195	C3	4	Read	14H-15H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
199	C7	4	Read	15H-16H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
203	CB	4	Read	16H-17H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
207	CF	4	Read	17H-18H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
211	D3	4	Read	18H-19H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
215	D7	4	Read	19H-20H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
219	DB	4	Read	20H-21H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
223	DF	4	Read	21H-22H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
227	E3	4	Read	22H-23H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
231	E7	4	Read	23H-0H Count2	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
235	EB	4	Read	0H-1H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
239	EF	4	Read	1H-2H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
243	F3	4	Read	2H-3H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
247	F7	4	Read	3H-4H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
251	FB	4	Read	4H-5H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
255	FF	4	Read	5H-6H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
259	103	4	Read	6H-7H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
263	107	4	Read	7H-8H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
267	10B	4	Read	8H-9H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
271	10F	4	Read	9H-10H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
275	113	4	Read	10H-11H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
279	117	4	Read	11H-12H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
283	11B	4	Read	12H-13H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
287	11F	4	Read	13H-14H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
291	123	4	Read	14H-15H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
295	127	4	Read	15H-16H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
299	12B	4	Read	16H-17H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
303	12F	4	Read	17H-18H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
307	133	4	Read	18H-19H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
311	137	4	Read	19H-20H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
315	13B	4	Read	20H-21H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
319	13F	4	Read	21H-22H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
323	143	4	Read	22H-23H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
327	147	4	Read	23H-0H Count3	Contagem de impulsos num certo tempo(Primeira palavra, contador1;Segunda palavra, contador2;...)		
331	14B	1	Read-Write	Valve Motor Control	00000000 0MV1 0MV2 0MV3 0MV4 (Abre e fecha a válvula, 0-Aberto;1-Fechado)	0	
332	14C	1	Read-Write	ErrorArray	0WSNP 0VNP 0OLWO RE PE (0-Ok;1-Error)(Water Sensor Not Present; Valve No Signal; OpenLid Switch Open; Reset; Prog Error)	0	
333	14D	1	Read	Comm Mod	Identificador do módulo de comunicação presente (00-Wifi;01-GSM/GRPS;10-LoRa;11-NB-IoT)	0	
334	14E	1	Read-Write	EndCommEarly	0-Nada;1-Acaba a comunicação entre o server cedo	0	
335	14F	1	Read-Write	Program Memory	0-Nada;1-A informação enviada vai ser guarda na memoria externa		
336	150	1	Read-Write	Reboot New Code	0-Nada; 1-Manda fazer reprogramação do micro controlador	9600	YES
337	151			GSM/GPRS			
337	151	25	Read-Write	Gateway IP/Hostname 1	50 caracteres		
362	16A	1	Read-Write	Gateway port 1	inferior 65535	api.ion.ssa-dynamics.com	YES
363	16B	25	Read-Write	Gateway IP/Hostname 2	50 caracteres	19002	YES
388	184	1	Read-Write	Gateway port 2	inferior 65535	83.240.189.154	YES
389	185	2	Read-Write	PIN	Validação das SMS recebidas	19002	YES
391	187					eX78	YES
391	187	1	Read-Write	Action code	Comando da SMS COMMAND		
392	188	8	Read only	Phone number 1	Gateway tem numero nacional?	N/A	NO
400	190	14	Read-Write	APN Server	Endereço do servidor APN	?	YES
414	19E	14	Read-Write	APN User	Nome de utilizador para a autenticação no servidor APN		
428	1AC	14	Read-Write	APN Password	Password utilizada para a autenticação no servidor APN		
442	1BA	1	Read-Write	APN Authentication	Tipo de autenticação usado no servidor APN (0 - Sem autenticação;1 - Autenticação PAP;2 - Autenticação CHAP)		
443	1BB						
443	1BB			WiFI			
443	1BB	25	Read-Write	Gateway IP/Hostname 1	50 caracteres	api.ion.ssa-dynamics.com	
468	1D4	1	Read-Write	Gateway port 1	inferior 65535	19002	
469	1D5	25	Read-Write	Gateway IP/Hostname 2	50 caracteres	83.240.189.154	
494	1EE	1	Read-Write	Gateway port 2	inferior 65535	19002	
495	1EF	2	Read-Write	Password	Validação das SMS recebidas	eX78	

Tabela H.1: Tabela Modbus.