José Domingos Reis Pinho Correia dos Santos

Controlo de acessos com comunicação LoRa

José Domingos
Reis Pinho
Correia dos Santos

Controlo de acessos com comunicação LoRa

"*The greatest challenge to any thinker is stating the problem in a way that will allow a solution*"

— Bertrand Russell

Universidade de Aveiro
2021

José Domingos Reis Pinho Correia dos Santos

Controlo de acessos com comunicação LoRa

Relatório de Estágio apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor André Zúquete, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha namorada e aos meus pais por estarem presentes em todos os momentos

o júri / the jury

presidente / president

Prof. Doutor Pedro Nicolau Faria da Fonseca
Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Nuno Miguel Abreu Luís
Professor Adjunto da Instituto Superior de Engenharia de Lisboa

Prof. Doutor André Ventura da Cruz Marnôto Zúquete
Professor Auxiliar da Universidade de Aveiro

agradecimentos / **acknowledgements**

Agradeço toda a ajuda a todos os meus colegas e companheiros.

Palavras Chave

Controlo de acessos, Abertura Remota, LoRa, Arduino, ATmega328p, PCB, Python

Resumo

O presente documento relata o trabalho desenvolvido, no âmbito do estágio curricular que frequentei durante o presente ano letivo. Estando a entidade acolhedora inserida no ramo do Controlo de Acessos, o desafio que me foi lançado, consistiu no projeto de um circuito eletrónico, com capacidade de abrir remotamente os equipamentos da empresa, através da tecnologia de comunicação LoRa.

Para além do circuito, foi também definido como requisito, a criação de uma interface gráfica, onde o utilizador possa tirar partido da solução desenvolvida, em termos de hardware.

Sendo assim, este relatório de estágio, descreve as várias fases do trabalho desenvolvido, nomeadamente, a definição dos requisitos do sistema, tanto a nível do hardware como do software/firmware, a apresentação do estudo realizado sobre a comunicação LoRa, incluindo o seu principio de funcionamento, as suas características, as suas mais-valias e limitações, bem como as principais áreas de aplicação e finalmente, a descrição mais detalhada da solução desenvolvida.

Desta forma, pretende-se descrever, com o detalhe possível, todo o trabalho realizado ao longo de seis meses de estágio.

Keywords

Access Controls, Remote Opening, LoRa, Arduino, ATmega328p, PCB, Python

Abstract

This document reports the work developed within the scope of the curricular internship which I attended during this school year. As the company is part of the Access Control business, the challenge that was set for me was the design of an electronic circuit, with the ability to remotely open the company's equipment, through the LoRa communication protocol.
In addition to the circuit, the creation of a graphical interface was also defined as a requirement, where the user can take advantage of the developed solution, in terms of hardware.
Therefore, this internship report describes the various phases of the developed work, namely, the definition of the system requirements, both at the hardware and software/firmware level, the presentation of the study carried out on the LoRa communication, including its working principle, its characteristics, its strengths and limitations, as well as the main areas of application and finally, a more detailed description of the developed solution.
Thus, it is intended to describe, with as much detail as possible, all the work carried out over the six months of the internship.

Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

ABP Activation By Personalization

ADR Adaptive Data Rate. Técnica utilizada em comunicação LoRa por forma a maximizar a autonomia dos dispositivos

AES Advanced Encryption Standard

APL Alowable Path Loss. Em telecomunicações, limite máximo de perdas no caminho para que uma mensagem seja transferida com sucesso

ASCII American Standard Code for Information Interchange

BJT Bipolar Junction Transistor. Tipo de Transistor

Blackout Period Tempo em que uma sub-banda / canal está indisponível para comunicar

chirp Sinal em que a frequência varia ao longo de tempo

CSS Chirp Spread Spectrum

downlink Envio de uma mensagem a partir de um gateway para um dispositivo

Duty-Cycle Ciclo de Trabalho. Razão entre o tempo que o sinal está HIGH e o tempo em que está LOW

EAGLE Easily Applicable Graphical Layout Editor

efeito de Doppler Fenómeno físico que descreve o desvio na frequência resultante do afastamento/aproximação entre o emissor e observador de uma onda/sinal

End-Device Designação atríbuida a um sensor/atuador inserido numa rede LoRa

FEC Foward Error Correction

FSCM Frequency Shift Chirp Modulation. Técnica de modulação em frequência utilizada em LoRa

FSK Frequency Shift Keying. Formato de modulação em frequência

FTDI Future Technology Devices International. Designação normalmente atribuída conversores TTL-Série

Gateway Sistema/Equipamento capaz de estabelecer uma ligação entre duas redes

HTTP Hypertext Transfer Protocol

IoT Internet of Things. Designação normalmente atribuída a um sistema sensorial/atuador onde são retirados dados de um ambiente real e são monitorizadas variáveis

ITS Intelligent Transport Systems. Sistemas Inteligentes de suporte à monitorização de tráfego rodoviário

LoRa Long Range. Tecnologia de Comunicação em Radiofrequência

LPWAN Low Power Wide Area Network

OTA Over the Air Activation

PCB Printed Circuit Board

python Linguagem de Programação

QoS Quality of Service. Parâmetro de avaliação de uma rede de comunicações

MAC Medium Access Control. Protocolo de gestão de acessos frequentemente utilizado em redes distribuídas

Mifare Tecnologia de Identificação por Radiofrequência

MISO Master Input Slave Output. Comunicação SPI

MOSI Master Output Slave Input. Comunicação SPI

RFID Identificação por Radiofrequência

Round Trip Time Em LoRa, tempo entre o envio de una mensagem, a sua receção e o envio de uma resposta de volta ao emissor

RS232 Protocolo de Comunicação Série

SPI Serial Peripheral Interface. Protocolo de Comunicação Série

Spreading Factor Parâmetro utilizado em comunicações LoRa, determina o quão espalhados estão os sinais na frequência

Time On Air Tempo de transmissão de uma mensagem

TTN The Things Network. Plataforma de gestão de mensagens LoRa

UART Universal Asynchronous Receiver/Transmitter

uplink Envio de uma mensagem de um dispositivo LoRa para o Gateway

CAPÍTULO 1

Apresentação da Entidade Acolhedora

Neste capítulo serão apresentados a empresa onde decorreu o estágio – iTEC e os objetivos do mesmo.

1.1 Descrição Geral da Atividade da Empresa

A empresa iTEC foi fundada em 2008 e desenvolve soluções inovadoras na área do Controlo de Acessos. Destacando-se pela inovação no plano tecnológico a empresa procura, de uma forma constante, a perfeição e a total simbiose entre os produtos que comercializa e as necessidades dos clientes, desígnio que está presente no lema da empresa – *One Step Forward*.

Os sistemas de controlo de acessos iTEC destinam-se predominantemente a empresas/complexos industriais, condomínios, casinos, hospitais mas também a qualquer ambiente/local que necessite de segurança. Como referido, a empresa pretende sempre, para além da vasta gama de produtos que já desenvolveu, criar soluções específicas para diferentes necessidades que possam surgir, tendo total abertura para novos desafios, estando sempre em busca da constante inovação tecnológica.

Neste sentido, a instituição marca presença em várias feiras internacionais na área do controlo de acessos, as quais, *ESSEN Security 2018*, *Expo Protection 2018*, *Decor Hotel 2018*, *ISC West 2019*, entre outras.

Em termos de produtos iTEC, destacam-se as fechaduras eletrónicas, sendo que estas podem ser utilizadas com cartão Identificação por Radiofrequência (RFID) (Tecnologia de Identificação por Radiofrequência (Mifare) – 13.56 MHz), com impressão digital, com código ou com reconhecimento facial. O utilizador pode optar pela versão *standalone* dos produtos ou pode fazer a gestão das suas fechaduras através de uma aplicação.

A empresa possui um vasta carteira de clientes em vários países, nomeadamente, Portugal, França, Canadá, Marrocos, Espanha, Inglaterra, Turquia, Croácia, Itália, Áustria, Polónia, Grécia, Vietname, China, entre outros.

Em Portugal, destaca-se o facto das soluções da iTEC estarem presentes em 7 dos 11 casinos existentes, com a implementação de sistemas de controlo de acesso por cartão, impressão digital e reconhecimento facial. Sendo que, a empresa foi responsável pelo desenvolvimento da solução que deu resposta à norma do governo português para implementação de medidas de segurança mais apertadas nos casinos, nomeadamente com identificação por impressão digital ou reconhecimento facial, à entrada de cada cliente.

1.2 Localização

A empresa situa-se no Feirapark, freguesia de São João de Vêr, concelho de Santa Maria da Feira.

1.3 Presença Online

Sendo a iTEC uma organização de índole tecnológica e inovadora, está presente nas principais redes sociais: Facebook[1], Youtube [2], Instagram [3] e LinkedIN [4]. Para além das plataformas referidas, detém também um site institucional [5].

1.4 Caderno de Encargos

Neste estágio curricular, o objetivo principal do trabalho proposto pela empresa foi o projeto e desenvolvimento de uma placa de circuito impresso capaz de abrir remotamente as fechaduras que a empresa já possui. Sendo que um dos desafios lançados seria a utilização da tecnologia de comunicação Long Range. Tecnologia de Comunicação em Radiofrequência (LoRa), a utilização deste tipo de protocolos de comunicação na área do controlo de acessos não é muito comum, assim seria um desenvolvimento inovador neste contexto.

O desafio foi então, desenhar um Printed Circuit Board (PCB) que possa ser incluído nas fechaduras que a empresa comercializa, aportando a estas a capacidade de serem acionadas remotamente. Sendo que, as fechaduras que a iTEC possui, dependendo do modelo, podem ser abertas através de cartão, código e impressão digital.

A empresa já comercializa uma solução para abertura remota através de uma aplicação móvel, sendo que a tecnologia utilizada para a comunicação entre as fechaduras e o dispositivo móvel é o Bluetooth. Sendo assim, o utilizador pode abrir a porta remotamente, contudo as limitações em termos de distância máxima de comunicação, que são tipicamente, 5 a 10 metros para o Bluetooth acabam por ser um pouco limitadoras para esta solução.

De realçar ainda que, a empresa disponibiliza um Software de *Desktop* que também permite abrir as fechaduras remotamente via Wi-Fi. O que, da mesma forma, limita a solução em termos de cobertura de rede e de gastos na instalação de pontos de acesso, sobretudo se estes sistemas forem instalados em edifícios de área considerável.

Ora estas questões levaram a que a empresa pensasse na implementação da comunicação LoRa, que possui uma capacidade de comunicação a longa distância considerável (da ordem das dezenas de quilómetro), o que, no contexto do controlo de acessos é muito significativo.

Para que o sucesso fosse conseguido neste estágio foi necessário desenvolver uma PCB capaz de, quando inserido nas fechaduras existentes, desbloquear a fechadura e permitir a sua abertura. Para além da PCB também teve de ser desenvolvido o *firmware* e uma interface com o utilizador.

Posto isto, espera-se então que, nesta fase, ou seja, no final do estágio, se encontre terminado o desenvolvimento de um sistema autónomo, capaz de realizar as funções pretendidas e que aporte um elemento diferenciador, aos produtos existentes na empresa.

1.5 Carga Horária

O referido estágio teve inicio no dia 26 de Outubro de 2020 e término no dia 6 de Maio de 2021, contabilizando-se um total de 944 horas. De referir ainda que, de 26 de Outubro a 19 de Janeiro o horário foi das 9 às 18 horas todos os dias da semana à exceção da terça feira em que o horário era apenas das 9 horas às 12 horas, devido à necessidade de frequentar uma unidade curricular na Universidade de Aveiro. No restante período o horário foi o normal das 9 horas às 18 horas, todos os dias da semana.

[1] `https://www.facebook.com/itecpt`
[2] `https://www.youtube.com/user/01iTec`
[3] `https://www.instagram.com/itecpt`
[4] `https://www.linkedin.com/company/itecpt`
[5] `https://itec.com.pt`)

1.6 Resultado Final do Projeto realizado no âmbito do estágio

No que diz respeito a resultados concretos, obtidos no final do estágio, foram entregues à empresa os seguintes componentes de hardware, um protótipo funcional em placa branca, onde está implementado todo a parte de hardware do sistema, para além deste, também ficaram na posse da empresa os protótipos desenvolvidos em PCB, isto é, o circuito que implementa a comunicação LoRa e o correspondente à antena.

Em relação à componente de software, a interface desenvolvida foi também deixada no servidor da empresa, esta poderá servir para que se desenvolva uma aplicação mais completa, e sobretudo que possa ser comercializada, em conjunto com o hardware desenvolvido.

Para além disto, foram entregues à empresa todo os programas onde se implementou o firmware necessário, para que o sistema detenha as funcionalidades exigidas.

De realçar ainda que as PCBs consituem o principal resultado do projeto, realizado no âmbito deste estágio, sendo que estas cumprem os requisitos definidos, apesar de terem sido identificados alguns erros, detalhados mais à frente neste documento, na fase de testes ao protótipo. Para que, a empresa conseguisse, efetivamente, vender a solução desenvolvida em maior escala, foram corrigidas estas situações e construída uma nova versão da placa principal, no programa Easily Applicable Graphical Layout Editor (EAGLE). É esperado ainda, e conforme acordado, entre ambas as partes, que essa placa entre em produção nos próximos meses e que o estudante possa acompanhar a sua introdução no mercado.

No que concerne ao acrescento de valor que este estágio aportou à empresa, este insere-se essencialmente numa perspetiva de contínua exploração desta tecnologia de comunicação, procurando diferenciar os produtos já existentes, conseguindo assim mais inovação para o portfolio da empresa. Obviamente, que nesta fase os lucros efetivos do estágio são diminutos, visto que o resultado final apenas são os referidos protótipos. Reforçando, que se espera que no futuro estes possam ser mais palpáveis com a produção e lançamento para o mercado das placas na sua versão "definitiva".

Em suma, foi entregue à empresa um sistema demonstrativo da utilidade da tecnologia LoRa no contexto do controlo de acessos. Sistema esse, que, é composto pelos circuitos, pelo *firmware* desenvolvido e, finalmente, pela interface do utilizador.

Feito este enquadramento, importa perceber a forma como o presente relatório está estruturado, então neste primeiro Capítulo é apresentada a entidade de acolhimento, no Capítulo 2 é feita uma descrição mais detalhada acerca do trabalho desenvolvido, sendo que são enumerados os requisitos do sistema, do ponto de vista do hardware e do software. O Capítulo 3 é constituído por um estudo detalhado da tecnologia LoRa, englobando os seus princípios de funcionamento, a sua origem, as suas vantagens e limitações e as suas aplicações típicas. Seguidamente, o Capítulo 4 e o Capítulo 5, contêm a apresentação da solução desenvolvida, do ponto de vista do hardware e do software. No último Capítulo são abordados alguns pontos que constituem uma análise crítica acerca do trabalho desenvolvido, sendo evidenciados alguns pontos onde o mesmo pode ser melhorado.

CAPÍTULO 2

Descrição do Projeto Realizado no Âmbito do Estágio Curricular

Neste capítulo será apresentado o Projeto realizado no contexto do estágio curricular.

2.1 Contexto

O objetivo, como já referido, foi desenvolver uma placa de circuito integrado com capacidade de comunicação baseada em LoRa.

A tecnologia de comunicação LoRa é baseada em radio frequência e permite a comunicação a longas distâncias – pode atingir o alcance de 10 Km em zonas rurais e cerca de 4 km em zonas urbana, com baixo consumo de energia. Quantificando, por exemplo, o módulo LoRa utilizado neste trabalho empírico, apresenta como corrente máxima, consumida durante as janelas de envio de dados, 120 mA. Sendo que, este valor é apresentado como o limite superior do consumo, ora, isto representa, considerando a tensão de alimentação nos 3.3 V, cerca de 400 mW.

A sua utilização incide principalmente no sistemas Internet of Things. Designação normalmente atribuída a um sistema sensorial/atuador onde são retirados dados de um ambiente real e são monitorizadas variáveis (IoT), nomeadamente, quando as mensagens trocadas são de curta duração, estando os mesmos colocados em locais de difícil acesso e com más condições de acesso à rede. No caso concreto do controlo de acessos, estas condições estão reunidas, pelo que fará todo o sentido o desenvolvimento de um circuito integrado, com a referida tecnologia de comunicação.

A ideia foi, então, fazer com que todas as fechaduras e/ou mecanismos de acessos, de um determinado edifício, comuniquem com a *Cloud* e que por sua vez o utilizador consiga, através de uma aplicação/interface, desbloquear/bloquear todos os acessos que pretender.

Isto é, as fechaduras têm de ser capazes de receber e transmitir informação, para o sistema principal.

A utilização do protocolo de comunicação LoRa, permite facilitar o controlo de acessos em grandes edifícios tais como complexos industriais, principalmente quando comparado com outras tecnologias de comunicação, nomeadamente, Wi-Fi e Bluetooth. Esta tecnologia de comunicação apresenta 3 vantagens significativas em relação ao Wi-Fi e Bluetooth, sendo elas o maior alcance, o menor consumo de energia e o facto de não ser necessária a instalação de vários pontos de acesso, dentro de um determinado edifício.

Contudo, esta tecnologia possui normas de utilização que podem ser limitadoras, nomeadamente, a imposição de um Ciclo de Trabalho. Razão entre o tempo que o sinal está HIGH e o tempo em que está LOW (Duty-Cycle) máximo de 1%. Isto é, na prática, um Sistema/Equipamento capaz de estabelecer uma ligação entre duas redes (Gateway), após enviar um Envio de uma mensagem a partir de um gateway para

um dispositivo (downlink) para abertura de uma fechadura, só estará disponível para comunicar passado x segundos. Em que x se obtém pela multiplicação do tempo de transmissão - *Over-The-Air Time* - por 99. Esta questão e as suas implicações práticas serão detalhadas no Capítulo 3.

Posto isto, com este protocolo de comunicação, apenas é necessário instalar um Gateway, num determinado local, idealmente sem necessidade de serem instalados replicadores de sinal, sendo que cada uma das fechaduras presentes no local tem de estar equipada com uma PCB capaz de efetuar a comunicação LoRa — Escrita e Leitura — com a antena.

Desta forma, pretendeu-se, com a elaboração da referida placa de circuito impresso, diferenciar o produto que a empresa já possui, tornando-o mais apelativo para o cliente final.

Com esta tecnologia, foi implementado o protocolo de comunicação LoRaWAN, que se baseia num sistema de comunicação ponto para multi ponto.

Com a solução apresentada pretende-se um sistema robusto, fiável e seguro que facilite a instalação e minimize os custos inerentes. Isto é, o sistema deve assegurar que qualquer individuo, sem permissões de acesso, não consiga desbloquear uma fechadura. Por outro lado, deve ser salvaguardada a possibilidade de um utilizador credenciado, desbloquear as suas fechaduras, sempre que assim pretender.

A placa de circuito impresso desenvolvida permite abrir remotamente as fechaduras iTEC, com base no protocolo de comunicação LoRaWAN. Esta é uma tecnologia de baixo consumo e longo alcance, sendo caracterizada também pela baixa largura de banda, e consequentemente a sua utilização é adequada, em aplicações onde o fluxo de informação se dá na forma de mensagens curtas, e com alguma periodicidade temporal, o que torna o Controlo de Acessos uma área a explorar, sendo este o principal objetivo do estágio curricular. [1]

Assim, o propósito consistiu em dotar as fechaduras existentes no portfolio da empresa, da capacidade de serem abertas remotamente, utilizando para tal as mais valias da comunicação LoRa, nomeadamente o longo alcance e o baixo consumo de potência, quando comparado a outros protocolos de comunicação.

2.2 Requisitos do sistema: Hardware

Em termos de requisitos para a componente de hardware do sistema, registam-se:

- a já referida capacidade de comunicação LoRa;
- o controlo da atuação de um motor DC de 6 V, responsável pelo desbloquear/bloquear das fechaduras;
- a PCB desenvolvida deverá possuir dimensão máxima de: 40×40 mm.

De realçar que, nesta lista não se encontra o Gateway LoRa visto que, a empresa adquiriu previamente um *kit* de Desenvolvimento LoRa que já possui esse mesmo componente. Sendo assim, esse Gateway foi o utilizado ao longo de todas as fases do trabalho.

Para a implementação do primeiro requisito definiu-se a utilização do módulo LoRa RFM95. Este módulo possui uma interface de comunicação Serial Peripheral Interface. Protocolo de Comunicação Série (SPI) a qual será utilizada para ligação ao microcontrolador usado.

No que concerne ao microcontrolador, foi utilizado o Atmega328p, sendo que a escolha recaiu sobre este devido à familiarização com a gama de microcontroladores da *Atmel*, nomeadamente com a programação de Arduinos.

Posto isto, de referir que estes dois componentes implementam o protocolo de comunicação LoRaWAN através da inserção do respectivo firmware, no microcontrolador. Como já referenciado, a comunicação dá-se através de SPI, sendo que este é um protocolo de comunicação série, bidirecional (Mestre $\leftrightarrow$ Escravo) e que se caracteriza por possuir taxas de débito de dados consideravelmente elevadas, quando comparados com outras (Protocolo de Comunicação Série (RS232), Universal Asynchronous Receiver/Transmitter (UART), etc). De qualquer modo, no caso do protocolo de comunicação implementado, este teria de ser mesmo o SPI, pois é o único disponível no módulo LoRa. Então, concluindo acerca desta questão, o microcontrolador desempenha o papel de Mestre controlando os *timings* e a periodicidade, em que se dão as transferências de dados, e o módulo LoRa será o Escravo.

Em relação à alimentação, esta é feita com um conjunto de 4 pilhas de 1.5 V, ou seja 6 V no total. Esta questão implica, logo à partida, o projeto de um circuito regulador de tensão, visto que a tensão máxima de alimentação do módulo LoRa utilizado é de 3.7 V. Este circuito, responsável pela conversão do nível de tensão, será baseado no integrado LM317.

E no que concerne à comunicação LoRa, estão apresentados os requisitos do sistema. Agora, importa perceber, como será feita a atuação, ou seja, o acondicionamento de sinal para conexão do microcontrolador ao motor das fechaduras.

Para isto são precisos dois circuitos:

- Circuito de amplificação dos sinais lógicos: foram utilizados dois pinos do microcontrolador, contudo os seus níveis de tensão são no máximo de 3.3 V, visto que o microcontrolador é alimentado a essa tensão. Sendo assim para acionar um motor de 6 V, precisa-se de um nível de tensão equivalente a esse. Ora, exatamente para suprir esta necessidade, foi desenvolvido um circuito de amplificação, baseado no Amplificador Operacional TLV226. A lógica de atuação do motor é: um pino de saída em modo *HIGH* e outro em modo *LOW*, levam ao rodar do motor num determinado sentido, sendo que o estado dual desses pinos, leva a que o motor rode no sentido inverso, consequentemente a fechadura bloqueia ou desbloqueia o trinco;
- Circuito responsável por fazer o *Drive* do motor, ou seja, colocar aos seus terminais, 6 V ou −6 V. Este circuito consiste numa Ponte-H, composta por transístores Bipolar Junction Transistor. Tipo de Transistors (BJTs) do tipo P e do tipo N.

Com isto, todos os circuitos do sistema estão apresentados, sendo que numa primeira fase estes foram montados em placa branca, testados, quando garantido o seu correto funcionamento, passou-se ao desenho da respectiva placa de circuito impresso no programa EAGLE.

2.3 Requisitos do Sistema: firmware e Software

Em termos de firmware e software registam-se uma série de requisitos, os quais, também se podem denominar de requisitos funcionais do sistema, ou seja, que tipo de funções/ações serão possíveis de implementar nas fechaduras, com o hardware desenvolvido.

Ora, estes são:

- Desbloqueio da fechadura no seu modo normal, ou seja, a fechadura é desbloqueada durante 10 segundos e volta a bloquear, sendo que esta janela temporal é suficiente, para que o utilizador possa abrir a porta;
- Desbloqueio da fechadura no seu modo sempre aberto, isto é, o trinco fica desbloqueado, até que seja dada outra indicação à mesma;
- Desbloqueio da fechadura, definindo para tempo de abertura, um dos seguintes valores: 1, 2, 3, 4, 5 6, 7, 8 ou 9 segundos;
- Realizar o *reset* do firmware, sendo que, esta é apenas uma forma de corrigir eventuais problemas de conexão ao *Gateway* LoRa.

Ou seja, estas possibilidades de ações, devem estar acauteladas tanto no firmware, como na interface do utilizador desenvolvida.

Para programação do microcontrolador foi utilizada a linguagem de programação C e a plataforma Arduino-IDE. No que concerne à injeção deste *firmware*, no microcontrolador, esta foi feita através da sua ligação a um PC, utilizando um cabo FTDI.

Como referido acima, a empresa adquiriu um kit de desenvolvimento LoRa, no qual se incluíam dois Arduinos UNO e duas *shields* LoRa. Sendo assim, foi implementado com o sucesso, o protocolo LoRaWAN, com esse hardware.

Ora, o Arduino UNO é baseado no microcontrolador ATmega328p, pelo que, se poderia utilizar o mesmo firmware, no protótipo em placa branca, daí a escolha do mesmo. Deste modo, procurou-se evitar problemas de adaptabilidade, entre o código já desenvolvido e outras famílias de microcontroladores.

Portanto, os requisitos foram cumpridos, através do firmware e de uma interface gráfica, onde é possível enviar os comandos desejados, para as fechaduras. Esta, serve como prova de conceito, de forma a mostrar que é possível interagir com o sistema, e implementar as funções que foram apresentadas acima. Esta interface foi desenvolvida em Python, com a biblioteca `WxPython`. Sendo assim, o utilizador terá um conjunto de botões ao seu dispor, os quais possibilitarão o envio de mensagens/comandos para o *Gateway*, que por sua vez as encaminha para o módulo LoRa. A informação, quando recebida pelo microcontrolador, é descodificada e inicia-se, a ação correspondente.

Todas os dados enviados/recebidos pelo Gateway, são armazenados na plataforma The Things Network. Plataforma de gestão de mensagens LoRa (TTN), onde é realizada uma integração Hypertext Transfer Protocol (HTTP) para que seja possivel agendar downlinks, isto é, o envio mensagens do Gateway para o módulo LoRa. Este processo, será detalhado no capitulo 4, deste relatório.

Esta interface serve de prova de conceito, ou seja, pretende-se que seja uma base funcional, para eventualmente, a empresa desenvolver uma aplicação mais robusta, e que possa ser comercializada. Desta forma, fica demonstrado, que é possível enviar comandos para o hardware, e que através do firmware, estes comandos sejam descodificados, para que as ações correspondentes, possam ser levadas a cabo.

2.4 Descrição Faseada do Trabalho Desenvolvido

Em termos de organização de trabalho, é de realçar que, numa primeira fase estudou-se, ao detalhe, a tecnologia LoRa e o respetivo estado da arte. Para além deste estudo, foi também analisado o hardware presente nas fechaduras, as quais seriam completadas, com a solução desenvolvida.

Desta forma, reuniram-se uma série de requisitos para o sistema. Nomeadamente, em termos de alimentação, de acondicionamento de sinal, necessário para ligação do microcontrolador ao motor, perceber quais as melhores opções em termos de módulos LoRa, e microcontroladores existentes no mercado, etc. De referir, ainda que, esta fase durou cerca de 2 meses, ou seja, desde o inicio do estágio curricular em 26 de outubro de 2021, até final do mês de dezembro do mesmo ano.

É importante sublinhar que, durante este período, a empresa adquiriu um *kit* de desenvolvimento LoRa, para que o estudo da tecnologia e modo de funcionamento não fosse apenas teórico, mas que o estudante também pudesse, perceber na prática, como é que estes sistemas poderiam ser implementados.

Uma vez reunidas todas estas informações e ainda na primeira fase, começou-se a projetar o esquema do circuito em si, sendo que uma vez terminado este processo, avançou-se para a segunda fase.

Nesta segunda fase, avançou-se para a encomenda de material, e consequente montagem do circuito em placa branca, de referir que, para que fosse mais prático, antes da montagem desenhou-se o esquema do circuito no EAGLE, até porque numa fase mais avançada, esse esquema iria ser necessário. Aquando da chegada do material, por volta do inicio do mês de janeiro de 2021, iniciou-se, então, a dita montagem do protótipo. Sendo que esta fase incluiu a montagem e testes ao protótipo. Esta revelou-se fundamental, visto que, algumas alterações ao circuito que tinha sido inicialmente projetado, foram realizadas tendo em vista a melhoria do seu desempenho, nomeadamente em termos de consumos, visto que, sendo alimentado por um conjunto de pilhas, essa questão revela-se fundamental.

Com o protótipo pronto, em meados de fevereiro de 2021, isto é, devidamente testado e a funcionar conforme o desejado, avançou-se para a terceira fase do projeto realizado em contexto de estágio: o desenho, no programa EAGLE, da PCB. Paralelamente com essa tarefa, também nesta fase começou a ser desenvolvida a interface com o utilizador, já referida.

O desenvolvimento do referido programa, foi deixado para esta fase pois apenas terminada a montagem do protótipo, se poderia avançar para a integração com a interface.

Esta terceira fase, durou cerca de 1 mês, tendo sido dada como terminada com a impressão das PCB, em meados de março de 2021.

Por último, a fase final teve duas tarefas principais: testes às PCB, procura de melhoramento da solução obtida e finalização do desenvolvimento da interface gráfica. De realçar que, esta fase durou desde meados de março até ao final do estágio, ou seja, até dia 6 de maio de 2021.

2.5 Funcionalidades implementadas

Com o término do referido estágio, regista-se a implementação das funcionalidades descritas acima, sendo que a principal, é a capacidade de abertura remota da fechadura, através da comunicação LoRa. Para isto, desenhou-se uma placa PCB, composta pelos circuitos referenciados acima, sendo que esta placa se coloca dentro das fechaduras existentes na empresa, aportando às mesmas a capacidade de serem controladas remotamente.

De referir que, apesar das dimensões da placa permitirem que esta seja instalada na maioria dos equipamentos da empresa, foi pensado, como trabalho futuro, em jeito de melhoramento da solução, a diminuição do tamanho da mesma, para que possa colocada em todos. Sendo que as dimensões da primeira placa produzida, são de $40 \times 40\,\text{mm}$. Para além da referida placa, foi desenvolvida e devidamente testada, a uma placa PCB que serve de antena LoRa. Os testes a estas placa, nomeadamente, em termos de alcance e integração com o TTN, são apresentados mais à frente, neste relatório.

De referir, ainda que, foram detectados alguns erros, após a impressão dos protótipos em PCB. Nomeadamente, o facto de, por lapso, se ter colocado na lista de material para a placa, um oscilador em vez de um cristal. Esta questão, inviabilizando o correto funcionamento da placa, teve de revista, sendo que se optou por dessoldar o referido oscilador e soldar um cristal de maior porte, dos que normalmente se usa em placa branca. Desta forma, este é um dos pontos a desenvolver, numa proposta, por parte da empresa, para continuação do acompanhamento da solução desenvolvida, nomeadamente, no que diz respeito à produção das placas em maior escala e à sua comercialização. Visto que, neste final do estágio curricular, apenas foram produzidos e testados 5 protótipos: conjunto placa principal + antena LoRa.

Para além do cristal, registou-se outro erro, que tem a ver com a ligação da resistência de *pull-up* do pino de *reset* à massa, e não à alimentação como deveria ser, aliás como está especificado nos esquema desenvolvidos. Para resolver esta questão seguiu-se o mesmo método, ou seja, dessoldou-se a referida resistência e colocou-se um circuito auxiliar para implementar a ligação referida.

Estas questões, sendo detectadas, nesta fase de produção e testes aos protótipos, permitem que, no futuro, seja desenvolvida uma placa já sem estes erros.

Para além das placas foi necessário, programar o microcontrolador, por forma, a que este implemente o protocolo de comunicação LoRaWAN e tome as ações necessárias em cada momento.

Este firmware foi desenvolvido, sendo que foi testado, ainda com o protótipo em placa branca. Sendo assim, o referido firmware foi entregue à empresa, sendo que foi confirmado o seu correto funcionamento.

Em relação à interface com o utilizador, esta foi desenvolvida em Linguagem de Programação (python), nas últimas semanas do estágio, sendo que esta serve de prova de conceito, demonstrando que é possível, a um qualquer utilizador interagir com o sistema, enviando para a fechadura os comandos que pretender.

Sendo assim, com a solução desenvolvida, o utilizador consegue, abrir as fechaduras remotamente, podendo especificar o tempo de abertura, ou se pelo contrário assim o pretender, pode enviar um comando para que a fechadura fique sempre desbloqueada.

Estas foram as funcionalidades que foram deixadas na empresa, sendo que existe a possibilidade, e foi pensada com os responsáveis de que eu continue a acompanhar o sistema desenvolvido, tendo em vista, o melhoramento da solução e até o acrescento de novas funcionalidades para o sistema.

CAPÍTULO 3

LoRa

Neste capítulo, será detalhada a tecnologia de comunicação LoRa, utilizada no desenvolvimento da solução.

3.1 Motivações e Origem do LoRa

Com a crescente disseminação dos sistemas baseados em IoT, foi necessário desenvolver uma técnica de comunicação em rede, que minimize o consumo de potência de cada elemento da rede (Nó), por forma a aumentar o seu tempo de vida. O objetivo era criar uma tecnologia que, ao invés de fazer com que cada nó transmita na máxima potência, redefina a potência de cada nó, de forma colaborativa [2].

Sendo uma tecnologia de comunicação sem fios, caracterizada pelo baixo consumo de potência e longo alcance, o termo LoRa deriva precisamente desta última característica — *Long Range*. A suas raízes remontam a uma técnica de comunicação, muito usada no âmbito militar na década de 40 do Séc. XX, devido a conseguir comunicação a longas distâncias e ser fiável e robusta, no que concerne a interferências, denominada Chirp Spread Spectrum (CSS) [3].

O LoRa consiste então, na primeira implementação de uma técnica de comunicação de baixa potência, para fins comerciais. Por forma a ultrapassar os desafios enumerados acima, os procedimentos de comunicação LoRa identificam numa determinada rede, os dispositivos que estão mais limitados do ponto de vista energético, transmitindo apenas alguns bytes, de cada vez que são acionados. A comunicação LoRa prevê que, tanto um sensor Designação atríbuida a um sensor/atuador inserido numa rede LoRa (End-Device) como uma entidade externa, possam iniciar uma comunicação. Sendo que, um End-Device, pode ser um sensor ou um atuador. Sendo assim, LoRa apresenta-se como uma excelente opção para aplicações "inteligentes", tipicamente, designa-se por aplicação inteligente ou aparelho inteligente, qualquer dispositivo que possa ser controlado remotamente, ou, a partir do qual, se possa, retirar informação de um ambiente real – sendo vastamente utilizada em *Smart Healtcare*, *Smart Cities*,*Environmental Monitoring*, *Industry*, etc [2].

Importa identificar duas camadas: uma camada física, correspondente à modulação dos sinais que é, como já referido, baseada na técnica CSS e possui características idênticas ao formato de modulação Frequency Shift Keying. Formato de modulação em frequência (FSK), no que ao alcance da comunicação diz respeito; um protocolo Medium Access Control. Protocolo de gestão de acessos frequentemente utilizado em redes distribuídas (MAC), que define a arquitetura da rede e toda a sua dinâmica. Especificando, o protocolo LoRaWAN impõe a autonomia de cada nó, a capacidade da rede, a qualidade do serviço Quality of Service. Parâmetro de avaliação de uma rede de comunicações (QoS), a segurança e fiabilidade dos sistemas em rede. Assim, como referido acima, o LoRaWAN consegue responder às necessidades e desafios das aplicações IoT.

O referido protocolo, diferencia-se por conseguir aliar duas características fundamentais: baixo consumo de energia e grande alcance comunicacional. Concretizando e comparando com as tecnologias de comunicação sem fios mais proeminentes — Wi-Fi, *ZigBee* e *Bluetooth* — consegue um alcance de até 15 km, em linha de

visão. Sendo que, as outras tecnologias vão até aos 100 metros no máximo. De realçar que, existem outras Low Power Wide Area Networks (LPWANs), que na sua especificação, possuem alcances similares — *Sigfox*, *Ingenu* e *DASH7*, contudo, existe uma característica que diferencia o LoRaWAN, que é possibilidade de se utilizarem diferentes Parâmetro utilizado em comunicações LoRa, determina o quão espalhados estão os sinais na frequência (Spreading Factor) — o quão espalhados os sinais ficam na frequência, para valores superiores, a largura de banda será maior – para os diferentes nós [4].

3.2 Topologia de Rede LoRaWAN

A Figura 3.1 ilustra a topologia de uma rede LoRaWAN, que se define como uma topologia estrela de estrelas/estrela em estrela.

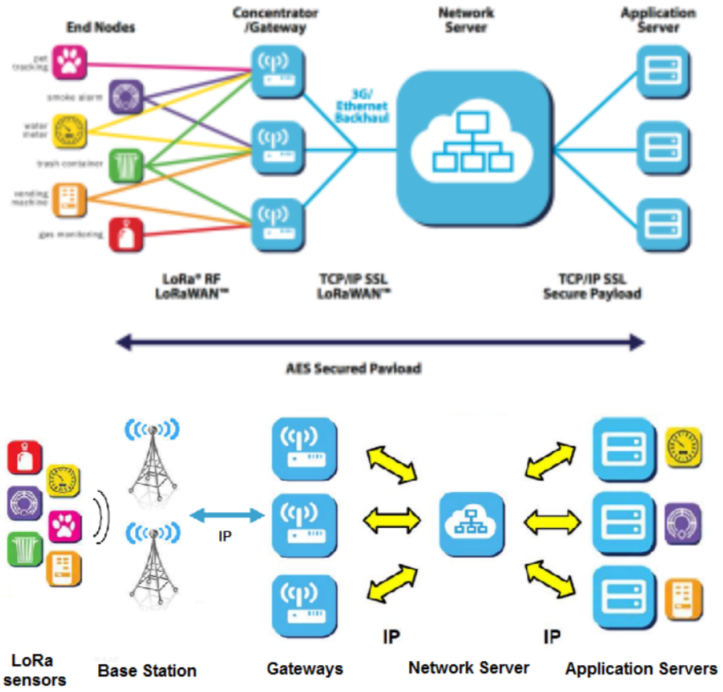


Figura 3.1: Protocolo e Arquitetura LoRaWAN[4]e[2]

A topologia em estrela, representa uma rede em que os dispositivos estão ligados a um ponto central, ora no caso do LoRaWAN temos:

- Os Sensores/Atuadores, normalmente na literatura por End-Devices, estão ligados a um ou mais Gateways, por via de comunicação LoRa. O que por si só, representa uma topologia de ligação em Estrela.
- Por sua vez os Gateways ligam-se por WiFi, Ethernet, Satélite ou 3G/4G a um servidor que, está conectado a uma ou mais aplicações, que serão as interfaces para o utilizador. Ora aqui, identificam-se, mais duas topologias de rede em Estrela.
- Para finalizar, se olharmos para a rede como um todo, dum lado temos os referidos End-Devices, do outro lado as interfaces do utilizador. Obviamente, aqui temos outra topologia em estrela, visto que, vários sensores/atuadores podem estar conectados às aplicações — sendo que, a informação percorre, todos os outros dispositivos intermédios. Daí, a topologia de ligação em rede LoRaWAN, ser tipicamente denominada de Estrela de Estrelas.

Para finalizar, a questão da topologia de ligação, de notar que, quando nos referimos a uma rede como Estrela, estamos a especificar sempre ligações bidirecionais, ou seja, a informação flui, do dispositivo para o ponto central, e do ponto central para o dispositivo. Do ponto de vista funcional, é importante perceber que papel têm estes dispositivos, na implementação de um sistema de comunicação LoRa.

Sendo assim e começando pelos End-Devices, estes tipicamente realizam duas funções: ler dados de um dado ambiente (sensor) / impor alguma ação no sistema (atuador) e comunicar com os Gateways, utilizando para o efeito comunicação LoRa. Os Gateway encaminham a informação, tipicamente denominada por LoRaWAN *frames / Payload*, para o servidor usando Ethernet, 3G/4G, satélite ou WiFi. O servidor descodifica os pacotes enviados pelos dispositivos, realizando ajustes em termos de débito de dados, e análises do ponto de vista da segurança, culminando na geração dos pacotes que vão ser reencaminhados de novo para os End-Devices. Por último cada aplicação do sistema, recebe os dados do servidor, descodificando os pacotes e tomando uma decisão, em relação à necessidade de impor a realização de alguma ação, no sistema [5].

3.3 Modulação em LoRa

A técnica de modulação usada em LoRa, patenteada pela *Semtech Corporation*, consiste no uso de um sinal Sinal em que a frequência varia ao longo de tempo (chirp) — sinal em que a frequência vai aumentando e diminuindo ao longo do tempo — tipicamente, a frequência deste sinal é superior à frequência dos dados em transmissão. Então, o sinal correspondente aos dados é codificado no sinal de chirp, o que resulta num sinal modulado, espalhado numa largura de banda superior à largura de banda do sinal original. O princípio subjacente a esta técnica é o de que, para transmitir um sinal sem erro, num canal com uma relação Sinal-Ruído fixa, basta aumentar a largura de banda do sinal a transmitir. Esta técnica é denominada de Frequency Shift Chirp Modulation. Técnica de modulação em frequência utilizada em LoRa (FSCM)

A implementação deste formato de modulação, permite a transmissão de múltiplos sinais, simultaneamente e no mesmo canal, garantindo uma mínima degradação no recetor. Assim o protocolo LoRaWAN, garante um Alowable Path Loss. Em telecomunicações, limite máximo de perdas no caminho para que uma mensagem seja transferida com sucesso (APL) de 157 dB e uma sensibilidade no recetor de até -137 dBm. Tipicamente, a camada física LoRa opera numa destas bandas de frequência: 433, 868 ou 915 MHz. Sendo que, na Europa, apenas a banda de 433 e 868 MHz, podem ser usadas. Por exemplo, na banda de 868 MHz existem 3 canais, a 125 KHz, que têm de ser implementados em todos os dispositivos, segundo a regulamentação existente [6].

Em termos matemáticos, importa especificar algumas relações, nomeadamente entre o débito de símbolos e *bits*, a largura de banda e o Spreading Factor.

$$\mathrm{R_s} = \frac{\mathrm{BW}}{2^{\mathrm{SF}}}$$
$$\mathrm{R_b} = \mathrm{SF} \times \frac{4}{4+\mathrm{CR}} \times \frac{\mathrm{BW}}{2^{\mathrm{SF}}}$$

$\mathrm{R_s}$	Débito de Símbolos (Símbolos/s)
BW	Largura de Banda de Modulação (Hz)
SF	*Spreading Factor* (7..12)
R_b	Débito de *bits* (*bits*/s)
$\frac{4}{4+CR}$	Débito de Código Foward Error Correction (FEC)

Pela análise das equações, consegue-se concluir alguns aspectos relevantes, na interpretação das grandezas associadas à comunicação LoRa, sendo eles:

- O débito de símbolos (R_s) depende da largura de banda (BW) e do fator de espalhamento (SF). Sendo que, o fator de espalhamento entra numa exponencial, portanto, aumentando este valor, diminui o débito de símbolos, ainda que, o aumento do fator de espalhamento também leve ao aumento da largura de banda.
- Logicamente, se o débito de símbolos diminui, o débito de *bits* correspondente também diminui, pelo que, conclui-se que o aumento do fator de espalhamento leva a uma diminuição do débito de *bits*.

3.4 *Adaptive Data Rate*

Como já referido, a tecnologia LoRa procura maximizar a autonomia dos dispositivos, utilizando um método de adaptação da taxa de transferência de dados — Adaptive Data Rate. Técnica utilizada em comunicação LoRa por forma a maximizar a autonomia dos dispositivos (ADR) — técnica de poupança de potência, onde o débito de dados e a potência de saída de Rádio Frequência, é adaptada conforme a distancia a que o nó está do Gateway. Concretizando, um dispositivo que esteja mais próximo, vai comunicar mais rapidamente (maior débito de dados) e com uma menor potência, em termos de Rádio Frequência. Desta forma, consegue-se que os dados provenientes desse nó, tenham um menor tempo em transmissão – *Time-on-Air*. Isto é conseguido, manipulando o parâmetro Spreading Factor, já que, utilizando um valor baixo para este parâmetro, tem-se uma taxa de transferência de dados elevada. Assim, este procedimento, permite acomodar alterações na estrutura da rede e variações nas perdas no caminho. Os débitos de dados estão entre 0.3 kbps e 50 kbps. Em contrapartida, o facto de se aumentar o tamanho das mensagens a transmitir, torna a comunicação mais suscetível a interferências e, consequentemente, os dados poderem não ser corretamente recebidos [7].

Para além disto, em LoRa os End-Devices são assíncronos, sendo a comunicação na rede iniciada, por eventos ou previamente agendada. Este facto, comparativamente com uma rede de comunicação síncrona, diminui a potência gasta, contribuindo para aumentar, mais uma vez, a autonomia dos dispositivos. A Figura 3.2, contém uma análise comparativa de protocolos de comunicação LPWAN, no que diz respeito ao consumo de potência e alcance de comunicação.

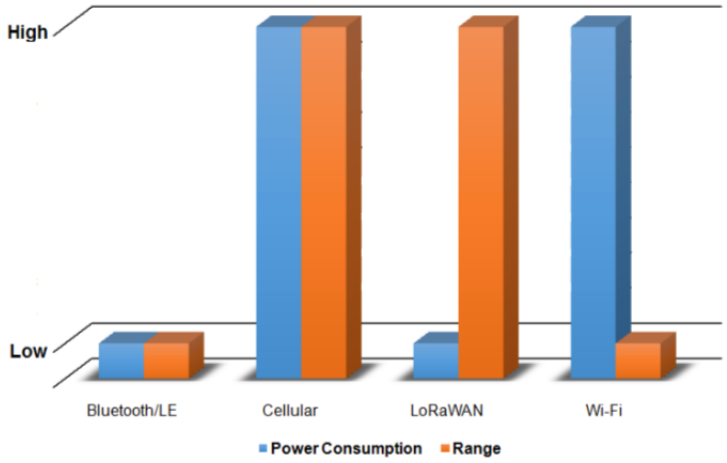


Figura 3.2: Análise Comparativa do Desempenho de Tecnologia LPWAN[2]

3.5 LoRaWAN MAC Layer

O protocolo MAC LoRaWAN, implementa o mecanismo que começa a comunicação, entre múltiplos dispositivos e Gateways. Como referido acima, esta camada define também a topologia de rede do LoRaWAN. Para além disto, este standard define três classes para os End-Devices: classe A, B e C.

Estas classes diferenciam-se pela periodicidade em que cada dispositivo está disponível para receber informação (downlink) e, consequentemente, isto tem impacto no consumo de energia dos dispositivos. Então, nos dispositivos da classe A, o processo de comunicação dá-se da seguinte forma: cada transmissão de informação, do dispositivo para o Gateway (Envio de uma mensagem de um dispositivo LoRa para o Gateway (uplink)), é seguido por dois pequenos períodos temporais, em que esse nó fica a aguardar por alguma informação, no sentido contrário downlink, posteriormente, acontece esse downlink por parte do nó. As janelas duram, respetivamente, 1 e 2 s, após o uplink. De referir que, os dispositivos de classe A são os que apresentam os menores consumos de energia, das três classes [6].

Já a classe B, caracteriza-se pela capacidade que os dispositivos têm para abrir janelas temporais, em que estão disponíveis para receber dados, em períodos pré-agendados. Desta forma os Gateways, transmitem beacon no processo de downlink, por forma a que os nós fiquem sincronizados e o servidor consiga determinar, quais os que estão disponíveis para receber dados. Consequentemente, os dispositivos desta classe, consomem mais energia comparativamente à classe A, por abrirem mais janelas temporais. Por último, os nós de classe C abrem, de forma quase contínua, estas janelas temporais, pelo que acabam por estar quase sempre disponíveis para troca de informações [6].

No que concerne à latência, de realçar que a classe A apresenta uma latência elevada, a classe B uma latência baixa e a classe C a mais baixa de todas. A Figura 3.3, apresenta os diagramas temporais das fases associadas a cada classe.

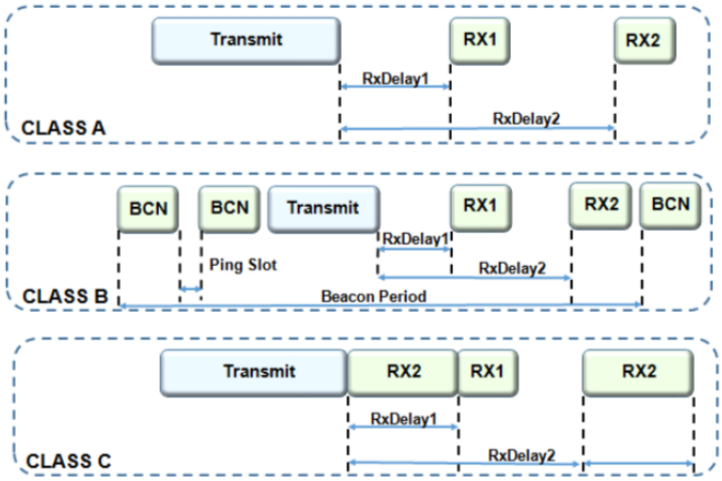


Figura 3.3: Classes para os Nós[2]

Tipicamente, as mensagens de um sistema de transmissão LoRa são compostas por: *preamble*, *physical header* (PHDR), *physical header Cyclic Redundancy Check* (PHDR_CRC), *physical payload* (PHY) e *error detection tail* (CRC). Sendo que, o tamanho do campo *preamble* é configurável em número de símbolos. O *Physical Payload* corresponde à informação que se pretende transmitir, normalmente organizada em *bytes*. O campo correspondente ao CRC só se aplica ao *uplink*, visto que o *downlink* é configurado, para o menor tempo de transmissão possível.

Preâmbulo [n símbolos]	PHDR [2 Bytes]	PHDR_CRC [4 Bits]	PHY Payload [Variável]	CRC [2 Bytes]

Tabela 3.1: Tramas em LoRa

3.6 Segurança

A segurança é uma característica fundamental, de todos os sistemas tecnológicos, nomeadamente, no que a sistemas de comunicação diz respeito. Neste aspeto, identificam-se, duas camadas de mecanismos que asseguram a segurança dos dados transmitidos, via LoRa. A Figura 3.4, representa estes mecanismos.

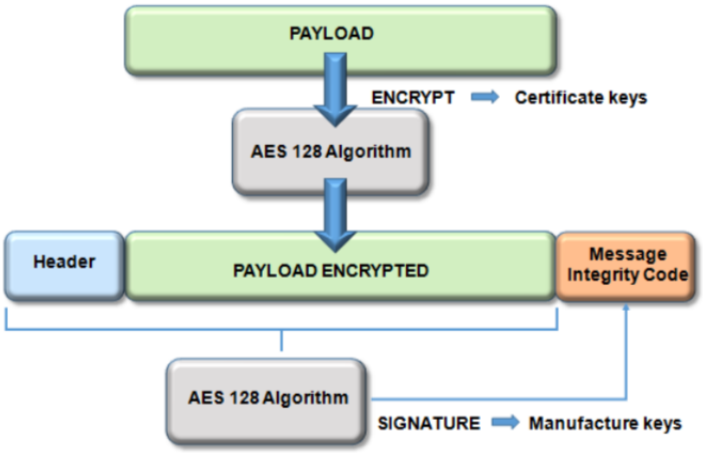


Figura 3.4: Segurança em LoRa[2]

A solução LoRaWAN, possui um mecanismo de autenticação baseada no Advanced Encryption Standard (AES) este algoritmo cifra as tramas para efeitos de confidencialidade, e gera mensagens de integridade. Tipicamente, cada dispositivo possui uma chave, que pode ser definida pelo seu fabricante ou por quem está a implementar o mesmo. De realçar que, no LoRaWAN o processo de autenticação e cifra é independente, sendo assim possível autenticar os pacotes e providenciar proteção, em termos de integração (com aplicações por exemplo) [8].

3.7 Outros Aspectos/Características importantes do LoRa

Esta tecnologia faz uso de uma parte do espectro eletromagnético, que não está licenciada, pelo que se pode transferir informação sem custos adicionais, para além disto, o protocolo de comunicação é assíncrono. Desta forma, em termos do parâmetro de qualidade de serviço da rede (QoS), não se encontra no leque de tecnologias de comunicação que providenciam melhores desempenhos. Embora o LoRa seja robusto, do ponto de vista da imunidade às interferências e à própria atenuação, não consegue providenciar a mesma qualidade de serviço, em relação a outras tecnologias de comunicação, como por exemplo, o *NB-IoT*. Este protocolo de comunicação, utiliza uma porção do espectro licenciada e o protocolo de comunicação é síncrono, o que o torna mais eficiente do ponto de vista do QoS [3].

Ainda no que diz respeito à autonomia, é importante referir que os dispositivos na rede LoRaWAN seguem o protocolo ALOHA.

Em sistemas de comunicação sem fios, importa relevar um fenómeno — Efeito de *Doppler* — e os seus impactos no desempenho dos sistemas. Este resume-se a uma diferença na frequência, entre o sinal radiado pelo transmissor e o sinal recebido, em termos práticos, este desvio, pode impossibilitar a correta receção do sinal transmitido no recetor. Como já referido, o protocolo de comunicação LoRa utiliza uma modulação CSS, ora se o *Chirp Rate* for grande, este desvio na frequência que, obviamente, leva a um desvio temporal, pode ser desprezado [1].

Como o LoRa tem um débito de dados que é influenciado, pelo facto de ser uma comunicação de longo alcance, ou seja, a diminuição do débito de dados é uma consequência da capacidade de transmitir a longas distâncias, esta característica tem impacto no *Chirp Rate*, fazendo com que este, também diminua, isto pode ser contraproducente, em relação à correta receção dos pacotes. Para ultrapassar estas dificuldades, são implementados os já referidos Spreading Factor, onde é possível acomodar maiores velocidades de transmissão, diminuindo estes valores. Concretizando, importa descrever, analiticamente, os impactos do Fenómeno físico que descreve o desvio na frequência resultante do afastamento/aproximação entre o emissor e observador de

uma onda/sinal (efeito de Doppler) nos sistemas. Assim, tipicamente, compara-se o *Coherence Time* (T_c) e *Symbol Time* (T_s).

$$\mathrm{T_c} = \frac{2\pi}{\mathrm{W_D}}, \mathrm{T_s} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}}$$

$\mathrm{T_c}$	Tempo de Coerência / Desvio Temporal associado ao efeito de *Doppler* (s)
$\mathrm{W_D}$	Desvio de Frequência associado ao efeito de *Doppler* (Hz
$\mathrm{T_s}$	Tempo de Símbolo (s)
BW	Largura de Banda de Modulação (Hz)
SF	*Spreading Factor* (7..12)

Realizando, então, a análise comparativa, se $T_s > T_c$, ou seja, se o tempo de símbolo for maior que o desvio temporal associado ao efeito de Doppler, este mesmo efeito torna-se dominante, fazendo com que haja distorção do sinal.

A Figura 3.5, contém um gráfico que ilustra uma análise comparativa, do tempo de coerência e do tempo de símbolo, em função da velocidade de transmissão, para diferentes fatores de espalhamento.

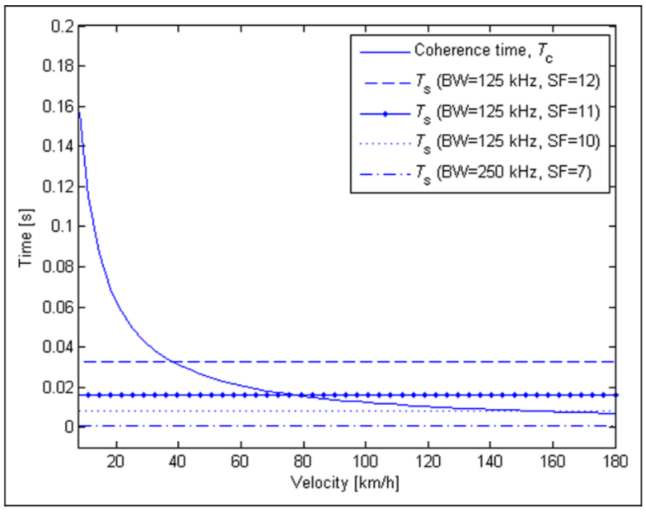


Figura 3.5: Comparação dos tempos de coerência e de símbolo para sinais LoRa com diferentes Fatores de Espalhamento[1]

De notar que, por exemplo, para um fator de espalhamento de 12, a curva de T_c cruza-se com a reta correspondente ao T_s, para uma velocidade de transmissão de cerca de 40 km/h. Ou seja, acima desta velocidade o impacto do Tempo em que uma sub-banda / canal está indisponível para comunicar (Blackout Period) será significativo, causando distorção do sinal. Por outro lado, diminuindo o Fator de Espalhamento, consegue-se velocidades de transmissão de dados sem chegar ao ponto crítico, no caso de utilizar um Spreading Factor de 11, a velocidade máxima será, aproximadamente 80 km/h. Conclui-se que, como já referido, a utilização de Fatores de Espalhamento mais diminutos, leva a que se consigam ter velocidades de transmissão maiores, garantindo que não há distorção do sinal.

Ainda, em relação ao débito de dados, o tempo de transmissão de uma trama é dado por:

$$\begin{aligned} \mathrm{T_{LoRa}} &= \mathrm{T_{preamble}} + \mathrm{T_{packet}} \\ &= \frac{1}{\mathrm{R_s}} \left(\mathrm{n_{preamble}} + \left(\mathrm{SW} + \max \left(\left\lceil \frac{8\mathrm{PL} - 4\mathrm{SF} + 28 + 16\mathrm{CRC} - 20\mathrm{IH}}{4\,(\mathrm{SF} - 2\mathrm{DE})} \,(\mathrm{CR}+4) \right\rceil, 0 \right) \right) \right) \end{aligned}$$

R_s	Taxa de Símbolos (Símbolos/s)
$n_{preamble}$	Número de Símbolos de Preâmbulo
SW	Comprimento da Palavra de Sincronização - 1 B
PL	Comprimento do *Payload (bytes)*
SF	Spreading Factor (7..12)
CRC	*Cyclic Redundancy Check*
IH	Modo de Operação
DE	Otimização de Débito de Dados

Relembrando a estrutura típica de uma trama em LoRa, representada na Tabela 3.1, o comprimento do *payload* em *bytes* é dado por:

$$\begin{aligned} PL &= MHDR + MAC_{payload} + MIC \\ &= MHDR + FHDR_{addr} + FHDR_{fctrl} + FHDR_{cnt} + FHDR_{opts} + F_{port} + FRM_{payload} + MIC \\ &= 12 + FHDR_{opts} + F_{port} + FRM_{payload} + MIC \end{aligned}$$

MHDR	Comprimento do *Mac Header*
$FHDR_{addr}$	Comprimento do *frame header address field*
$FHDR_{ctrl}$	Comprimento do *frame header control field*
$FHDR_{cnt}$	Comprimento do *frame header counter field*
$FHDR_{opts}$	Comprimento do *optional frame header*
F_{port}	Identificador do Porto
MIC	*Message Integrity Code*

3.8 Escalabilidade e Capacidade de Rede

Como já escalpelizado anteriormente, a técnica de modulação LoRa aporta capacidade de comunicação a longa distância, contudo existe uma questão importante, que é quantos dispositivos se podem ligar em rede apenas com um Gateway LoRaWAN.

De acordo com testes realizados e com a literatura disponível, este número depende da periodicidade da troca de informações entre os dispositivos e o Gateway. Concretizando, se essa mesma periodicidade for de 1 vez por dia e as mensagens trocadas forem de pequena dimensão, podem-se ligar milhares de dispositivos. Se pelo contrário, as mensagens forem trocadas com periodicidade de minuto a minuto, podem-se conectar, numa só rede, entre 100 a 1000 dispositivos.[9]

No que concerne à distância a que os Gateways têm de ser colocados, esta também depende da periodicidade de transmissão de dados, sendo que para os dispositivos que comunicam mais frequentemente, devem-se colocar Gateways de quilómetro em quilómetro, se pelo contrário, a comunicação for esporádica, a distância já poderá ser da ordem das dezenas de quilómetros. [6]

3.9 Aplicações LoRa

Para além de todas as questões técnicas já detalhadas, importa referir as aplicações onde se utiliza o LoRaWAN, sendo elas:

- Monitorização de Saúde e Bem-Estar [10];
- Monitorização Agrícola [9];
- Redes Sem Fios de Sensores [8];
- Monitorização de Tráfego [11];
- Aplicações de Localização [12];

- Cidades Inteligentes [13].

Sendo usada, maioritariamente, em aplicações que não requerem bons desempenhos no que concerne à latência, por outro lado, não é adequado para aplicações em que seja necessário transferir grandes quantidades de dados, por exemplo, em aplicações de Vídeo vigilância.

Em termos de desempenho, foram realizados testes, em ambiente citadino, que demonstram que para uma distância de comunicação de mais de 2 quilómetros, 95,5% dos pacotes foram recebidos com sucesso. Em termos de mobilidade, testes realizados comprovam que o desempenho piora, visto que, nas mesmas condições, para nós estáticos, se tem menos de 2% de perdas e para nós em movimento as perdas são de mais de 20%. De referir que, estes testes foram realizados nas seguintes condições: no interior de uma casa e distante do Gateway [8].

Detalhando um pouco os pontos realçados acima, começando pelas aplicações na área da monitorização da saúde e bem-estar, é de realçar um sistema de diagnóstico de fluidos corporais (sangue/urina) capaz de detetar níveis de leucócitos e nitratos, em que os resultados são enviados via LoRa*(Lorank Gateway)* para a *Cloud*. O sistema completo possui ainda um módulo *Bluetooth* para conexão a um dispositivo móvel através de uma aplicação. Desta forma, este é um dos exemplos em que se verifica a implementação de soluções de *remote healthcare*, sendo que as informações enviadas através da comunicação LoRa são armazenadas em servidores, onde mais tarde, poderá se aceder a um histórico das análises que o utente realizou [10].

Nesta aplicação, identificam-se os componentes principais de uma rede IoT: o sensor - sistema de diagnóstico das substâncias presentes nos fluidos corporais, a rede LoRa – envio das leituras para um servidor e uma interface com o utilizador – *Android App* onde este pode visualizar os resultados em tempo real (*Bluetooth*). De referir, ainda que o *hardware* responsável pela comunicação LoRa é baseado no módulo RN2483. [10]

Foram realizados uma série de testes em que se enviaram amostras cujo resultado já era previamente conhecido, tendo uma acusado positivo para um determinada substância e outra acusado negativo, o objetivo foi perceber se a os dados transmitidos para cada um dos testes eram recebidos de forma correta , sendo que, os testes se levaram a cabo em 3 zonas diferentes:

- numa zona rural, tendo-se obtido resultados que indicam uma correta transferência da informação, ou seja, para a amostra negativa foi recebida essa informação tal como para a amostra positiva, de realçar que os testes foram realizados colocando os sistemas de diagnóstico em 3 posições diferentes em relação à estação base (Gateway): respectivamente a 6, 5.5 e 4.3 quilómetros; [10]
- numa zona suburbana, onde de igual modo se identificaram resultados satisfatórios sendo que a distância a que os dispositivos se encontravam do Gateway diminui um pouco: de 1.7 a 3.8 quilómetros; [10]
- por último numa zona urbana, onde desta feita, 3 sistemas de diagnóstico se colocaram a 1.8, 1.1 e 3.4 quilómetros. [10]

Como já referido neste capitulo é de prever que devido à morfologia dos terrenos/zonas onde os sistemas LoRa são instalados, ou seja, mediante a existência ou não de edifícios, outras estações de comunicações e variadas infraestruturas, a distância a que se consegue com efetividade comunicar por LoRa vai variando. E isso está presente nesta descrição dos testes efetuados, onde para a zona rural conseguem-se as maiores distâncias de propagação, ressalvando que em todos os casos a comunicação se estabeleceu com êxito.

No que concerne às aplicações da tecnologia LoRa no âmbito da monitorização agrícola, destaca-se o trabalho desenvolvido em alguns países por parte da empresa *Libelium* com sede em Saragoça, Espanha.

Esta empresa desenvolveu alguns trabalhos que se destacam, sendo eles: sistema de rega inteligente para plantação de kiwis em Itália, onde com a instalação de um conjunto de sensores nas plantações se conseguiram identificar as necessidades das plantas em determinados momentos, e satisfazê-las da melhor forma, sendo que a comunicação escolhida foi o LoRa foi escolhida devido a dois fatores essenciais: o tamanho da plantação (da ordem das dezenas de quilómetros) e a falta de condições de acesso à rede no local. Desta forma foi conseguido um aumento de produção nos kiwis significativo. Sendo que apenas foi necessário instalar um Gateway numa zona com acesso à rede. [9]

Este é apenas um exemplo, contudo registam-se mais em que a aplicação da tecnologia de comunicação LoRa se mostrou adequada para monitorização de plantações agrícolas. É o caso, ainda dentro do portefólio da empresa *Libelium*, de: monitorização de áreas verdes usando um sistema de jardins inteligentes, melhoria de aspectos relacionados com a fertilização em campos de milho (Itália), desenvolvimento de um sistema de rega inteligente (Barcelona), controlo das condições climatéricas de uma plantação de tabaco (Itália), entre outros [14].

Em suma, o LoRa é aproveitado nesta área da monitorização de plantações agrícolas por ser facilmente integrado numa rede de sensores, mas sobretudo pela natureza e condições dos terrenos onde normalmente não é fácil instalar outro tipo de redes. Por exemplo, considerando uma plantação de tamanho da ordem das dezenas de quilómetros, seria dispendioso garantir cobertura de rede Wi-Fi ou mesmo por cabo. Assim, com apenas um Gateway e com a instalação do hardware necessário junto dos sensores, consegue-se cobrir áreas de terreno consideráveis, salientado que em ambiente rural conseguem-se distância de propagação em LoRa muito significativas.

Assim, consegue-se implementar redes de sensores em que se determinam as necessidades de um determinado cultivo e se implementam as ações necessárias para aumentar a produtividade de cada plantação.

De seguida, temos a apresentação de uma solução de utilização do protocolo LoRaWAN no contexto de vigilância/monitorização de veículos de transporte – Intelligent Transport Systems. Sistemas Inteligentes de suporte à monitorização de tráfego rodoviário (ITS). Contextualizando, estes sistemas englobam a implementação de estratégias de gestão de tráfico, navegação e localização. Estas estratégias tomam partido de vários processos, nomeadamente, de visão, de processamento de imagem e vídeo, monitorização de áudio e vídeo e a implementação de sensores no contexto dos veículos [11].

Ora, como referido, o que se irá descrever é a introdução da comunicação LoRa para integração destes dados em servidores e para que os utilizadores possam ter acesso a eles em aplicações móveis.

Para isso e neste exemplo de aplicação foram definidas algumas alterações em relação à estrutura típica de uma rede LoRa e do protocolo LoRaWAN.

Sendo que a principal diferença é a mobilidade dos Gateways, sendo que estes serão implementados em *drones*, contudo estes terão a mesma função que têm nas redes "tradicionais", ou seja, providenciam uma ponte entre a rede, os sensores e a interface dos utilizadores (aplicações móveis, tipicamente).

Desta forma, existe um conjunto de sensores e localizadores que são implementados nos veículos - que constituem os habituais sistemas ITS – onde estes comunicam via LoRa com os *drones* que vão sobrevoando uma determinada área, com a preocupação de cobrir a maior área possível em termos de distância de propagação, onde as informações sensoriais são enviadas para a *cloud* e consequentemente, disponibilizadas em aplicações móveis. [11]

De realçar, que todos estes exemplos não são mais que a implementação de uma rede sensores, destinados a retirar informações de algum processo físico, e cujos dados são enviados através da implementação do protocolo LoRaWAN onde estes podem ou não determinar ações a implementar no sistema (atuadores) e em que o utilizador consegue visualizar estes dados em algum tipo de interface gráfica. Ou seja, isto vai ao encontro da definição apresentada para uma rede LoRa.

Na área das Cidades Inteligentes destaca-se o projeto RIGERS que foi pensado com o objetivo de monitorizar as condições em residências e edifícios em duas zonas de Bolonha, Saragoça e Navile, em que foram colocados sensores de humidade, temperatura, luminosidade e dióxido de carbono. Estes sensores medem as respetivas grandezas de 5 em 5 minutos e a cada hora enviam a média dessas medições, via LoRa para um Gateway num determinado local estratégico da cidade [13].

É apenas mais um exemplo da topologia de rede apresentada nos outros exemplos de aplicação, este mais direcionado para a monitorização das condições de vida da população nomeadamente em aspectos como a qualidade do ar e mesmo as condições no interior das habitações.

Em suma, os sistemas de comunicação LoRa têm vindo a ser cada vez mais utilizados, em várias áreas, sendo que possuem vantagens claras no desempenho em relação a outras tecnologias de comunicação.

Contudo, todas as tecnologias possuem limitações e é precisamente esse aspecto que será detalhado na próxima secção deste capítulo.

3.10 Limitações da Implementação do Protocolo LoRaWAN e da tecnologia LoRa

Nesta secção serão abordadas as limitações da implementação da tecnologia LoRa bem como do protocolo LoRaWAN.

Como já foi referido ao longo deste capítulo, uma das principais limitações práticas da tecnologia LoRa é a imposição da norma relativa ao tempo de activação (*duty-cycle*). Sendo que, por exemplo na banda Europeia – EU 868 ISM – o máximo *duty-cycle* permitido é 1 % para cada uma das sub bandas de comunicação de cada um dos *End Devices*. Isto, na prática, impõe um máximo tempo de transmissão (para cada sub-banda) de 36 segundos por hora. Para além disto, tipicamente, existe outro fator que expõe ainda mais esta lacuna que é a preferência do uso de Fatores de Espalhamento mais elevados, isto para se atingirem distâncias de propagação mais substanciais. Contudo, o aumento do Fator de Espalhamento é diretamente proporcional ao aumento de outra grandeza – o *Time-On-Air* – ou seja, o tempo que medeia o inicio de transmissão de um pacote até este ser recebido.

Será então importante, exemplificar o impacto desta restrição numa transmissão baseada em LoRa, por exemplo, se um nó utiliza uma sub-banda durante 1 segundo para transmissão, este canal ficará indisponível durante os próximos 99 segundos. Tipicamente define-se este período de indisponibilidade como o *Blackout Period* da sub-banda/canal [15].

Este valor é dado por:

$$B_P = \frac{T_A}{D} - T_A$$

B_P	Blackout Period
T_A	Tempo de transmissão de uma mensagem (Time On Air)
D	Duty-Cycle

Concluindo, a tendência de utilização de Spreading Factors mais elevados para obtenção de maiores distâncias de propagação levará a um aumento do Blackout Period de cada sub-banda após uma dada transmissão.

Ora outra questão importante, ainda dentro da problemática enunciada é o número de canais/sub-bandas disponíveis, sendo que este número pode variar de acordo com a rede em que estão ligados os nós. A especificação LoRaWAN determina um mínimo de 3 canais disponíveis para cada um dos End-Device, contudo, por exemplo a plataforma TTN coloca à disposição dos mesmos 8 sub-bandas. Sendo que, para cada uma destas o Duty-Cycle é 1% [15].

Logicamente, se cada nó tem à sua disposição N canais para transmissão de informação e a estes é imposto um máximo Duty-Cycle de 1%, é atribuído ao nó um Duty-Cycle de N%.

Isto, porque está a utilizar N canais com um Duty-Cycle de 1%. Considerando a equação acima, o aumento do Duty-Cycle vai levar a uma diminuição do Blackout Period.

Todavia para o cálculo do tempo de não activação ou indisponibilidade de um nó é preciso ter em conta outro aspeto, o chamado Em LoRa, tempo entre o envio de una mensagem, a sua receção e o envio de uma resposta de volta ao emissor (Round Trip Time) – tempo que medeia o envio de uma mensagem do nó para o *Gateway* e o envio da respectiva resposta para o nó. Este valor é dado por:

$$RTT = 2 \times T_A + R_D$$

RTT	Round Trip Time
T_A	Time On Air
R_D	*Receive Delay*

Com esta grandeza calculada, é possível determinar, então, o Blackout Period teórico para um nó que transmite através de N canais/sub-bandas. Ora a seguinte equação demonstra como:

$$B_{PN} = B_P - (N - 1) \times RTT$$

B_{PN}	*Blackout Period N*
B_P	Blackout Period
N	Número de sub-bandas/canais disponíveis
RTT	Round Trip Time

A Figura 3.6 ilustra a variação do Blackout Period conforme se alteram o Spreading Factor e o *Payload Size* em duas situações: 3 canais disponíveis para cada nó e 8 canais disponíveis para cada nó.

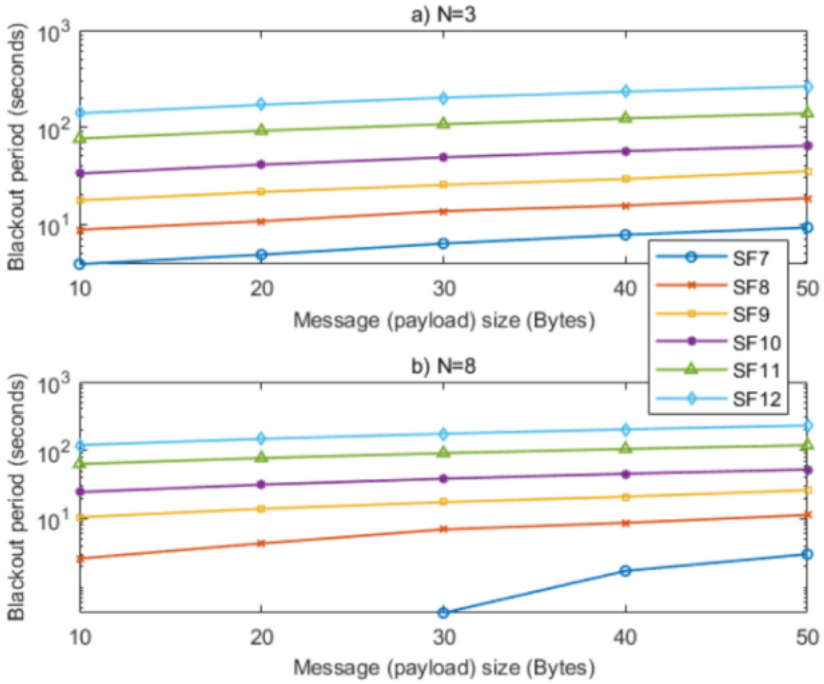


Figura 3.6: Evolução do *BlackoutPeriod* com o número de canais disponíveis[15]

Identifica-se uma melhoria significativa da performance no que diz respeito à disponibilidade de acesso aos canais no caso em que o nó utiliza 8 canais/sub-bandas, sendo que esta melhoria é essencialmente visível para Spreading Factors mais baixos, nomeadamente para valores de 7, 8 e 9. A partir do valor 10 regista-se poucas melhorias, sendo que isto permite reforçar a ideia deixada acima, de que o aumento do fator de espalhamento contribui para um aumento dos tempos de transmissão – Time On Air e consequentemente do Round Trip Time – o que leva a que o tempo de inativação não sofra a redução desejada, quando comparadas as duas situações (Número de canais = 3 e Número de Canais = 8). Sendo que, esta análise apenas se verifica quando as mensagens têm um tamanho igual ou inferior a 30 bytes.

Concluindo, a imposição de um limite em termos de Duty-Cycle pode ser contornada com a alocação de mais canais a cada nó de uma determinada rede, contudo limita um pouco a gama de aplicações onde o LoRa pode ser aplicado com sucesso. Por exemplo, em aplicações de vídeo vigilância identificam-se a priori dois problemas: a quantidade de dados que será necessário transmitir em cada *Payload* - tipicamente seriam vídeos em tempo real – o que de todo não se coaduna com as características do LoRa nomeadamente em termos de

largura de banda – para além disto, como envolveria uma exigência de transmissão de dados em "tempo-real" não seria exequível precisamente por questões de Duty-Cycle.

Contudo, na área dos sensores e da monitorização de dados – IoT – tipicamente, as mensagens são de tamanho reduzido e a exigência em termos de periodicidade não é elevada, pelo que se torna na área de excelência para implementação de redes de comunicação em LoRa.

Outro aspecto que importa detalhar é averiguação da existência de vulnerabilidades em termos de segurança na implementação do protocolo LoRaWAN.

Como já foi detalhado acima, o protocolo de comunicação LoRaWAN, implementa duas chaves com cifra AES com chaves de 128 bits): a *Network Session Key* e a *Application Session Key*. Estas são responsáveis por:

- Verificação da integridade e confidencialidade dos dados entre o dispositivo IoT e infraestrutura de rede (ex. TTN), bem como entre o dispositivo e uma aplicação/interface com o utilizador;
- Verificação da integridade e confidencialidade da ligação entre os dispositivos IoT e a aplicação.

A Figura 3.7 apresenta o referido.

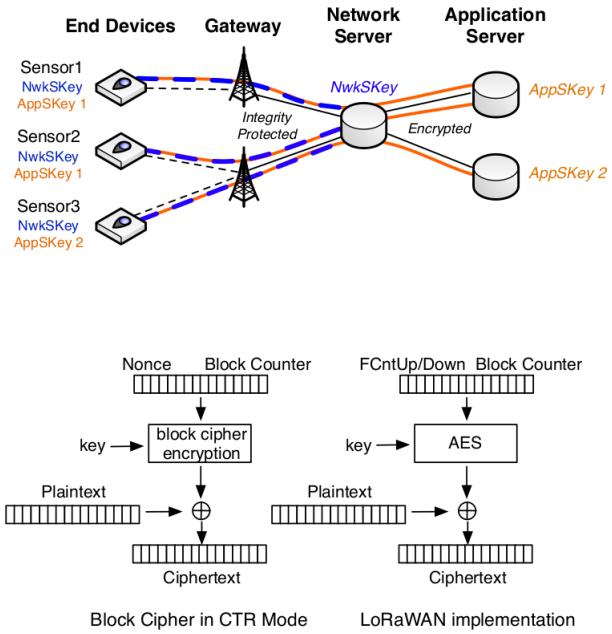


Figura 3.7: Segurança LoRaWAN com AES[16]

De uma forma sucinta, quando uma mensagem é enviada para o servidor, a trama (*payload*) é cifrada, primeiramente, pela *AppSKey*. A confidencialidade dos dados é assegurada por um bloco de cifra AES que opera em modo *Counter* (CTR). A cifra por blocos providencia uma permutação pseudoaleatória que posteriormente é usada para cifra do conteúdo original através do uso de um OU exclusivo (XOR). Sendo assim o protocolo LoRaWAN segue o modelo CTR. As mensagens FCntUp ou FCntDown como *nonce* que é incrementado a cada mensagem sendo que caso estas mensagens não se repitam, o modo CTR é idêntico ao seguido pelo protocolo LoRaWAN. Como tal, é necessário "introduzir" estas chaves no dispositivo IoT e isto é feito, tipicamente de uma de duas formas:

- *Over-the-Air Activation* Over the Air Activation (OTA)): O dispositivo/nó envia, primeiramente, um pedido de adesão (*Join Request*), que possui um DevNonce ou DevAddr de 3 bytes. Posteriormente, o servidor, recebendo o pedido, decide se o nó pode ser aceite ou não na rede. Caso não seja aceite,

não haverá resposta, caso seja aceite é enviada uma mensagem de aceitação – *Join Accept*. Sendo que, OTA deriva as chaves – NwkSKey e AppSKey cifrando os dados usando a AppKey que é definida pelo utilizador aos nós.

- *Activation by Personalization* (Activation By Personalization (ABP)): Não se implementa o processo de adesão à rede (*Join*) pelo que após a ativação os parâmetros DevAddr, NwkSKey e AppSKey são definidos no nó e armazenados no servidor.

Apresentada a forma como LoRaWAN implementa os mecanismos de segurança necessários, importa perceber ainda assim, que vulnerabilidades existem na implementação do protocolo.

A principal vulnerabilidade identificada é, aquando da utilização do modo de ativação ABP pelo facto de após reinicio do End-Device, este utilizar o *frame counter* a partir do 0 (zero) mas com as mesmas chaves. Para além do *Reset* do dispositivo, existe outra forma de fazer com que o *counter* volte a 0 (zero) que é a situação de *counter overflow*. Assim, um atacante – um dispositivo com capacidade de comunicação LoRa, cuja finalidade de utilização, seja aceder aos dados enviados/recebidos por outro sistema de comunicação – poderá aceder às ultimas mensagens trocadas e a partir das chaves, que serão as mesmas, iniciar uma nova sessão de comunicação. [16]

Portanto, para além deste mecanismo de ataque, este dispositivo, pode ainda utilizar outro método, vulgarmente designado por método de *replay*. Este consite, em traços gerais, na monitorização e armazenamento das mensagens enviadas para o Gateway e na verificação do momento em que um determinado dispositivo efetua o reset ao *counter* (`FCnt`). Nesta situação é possível enviar repetições de mensagens, sendo que apenas uma condição deve ser respeitada, para que estas sejam aceites pela rede.

$$\mathrm{FCnt_m} - \mathrm{FCnt_{curr}} \leq \mathrm{Gap}$$

$\mathrm{FCnt_m}$	Valor do *counter* para a mensagem maliciosa
$\mathrm{FCnt_{curr}}$	Valor atual do *counter* do *End-Device*
Gap	Máximo valor para a diferença entre *counters*

Para evitar que esta situação possa acontecer, ou pelo menos para impedir que seja tão simples da parte dos atacantes, deverá, sempre ser utilizado o método de activação OTA. Este método, como já escalpelizado, implica que todos os nós, tenham que enviar um pedido para se "juntarem" à rede. Este pedido será, posteriormente, aceitado ou recusado, mediante esta decisão, será enviada ou não uma mensagem para o nó de *Join* aceite.

De realçar que, a cada reinicio das sessões, ou seja, após cada processo de *Join*, as chaves são sempre alteradas, pelo que a situação anterior, será, neste contexto, impraticável. Daí o método OTA ser preferencial no desenvolvimento de aplicações e redes de sensores LoRa.

CAPÍTULO 4

Exploração da Solução: *Hardware*

Neste capítulo será detalhado o processo de projeto do hardware necessário para implementação da solução.

4.1 Diagrama de Blocos do Sistema

Do ponto de vista do hardware, o sistema é constituído por: fechadura (*iTEC Lock)*, onde está inserida a PCB desenvolvida e Gateway LoRa. No que concerne ao software identificam-se um LoRa *Server* e uma interface do utilizador. Sendo que a forma como a informação flui é ilustrada na Figura 4.1.

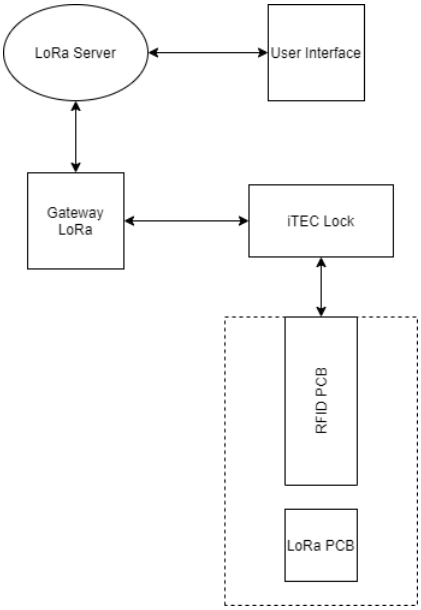


Figura 4.1: Diagrama de Blocos Sistema

Deste modo, identificam-se 4 blocos principais:

- iTEC *Lock*: fechaduras da empresa que possuem um PCB capaz de abrir/fechar através de cartão RFID e nas quais foi introduzido o PCB desenhado no decurso do estágio, o qual aporta a capacidade de abertura à distância;
- Gateway LoRa: providencia um meio de comunicação entre a fechadura e o servidor;

- LoRa *Server*: Local onde são armazenados as tramas enviadas/recebidas pelo Gateway, sendo que é a partir deste sitio que é possível realizar uma integração HTTP para que o utilizador consiga enviar comandos para o circuito;
- *User Interface*: interface onde o utilizador pode acionar as fechaduras remotamente.

Importa, então perceber qual a constituição e modo de funcionamento do circuito que foi projetado.
O circuito é composto por:

- Microcontrolador: Responsável por todo o processamento de sinal, nomeadamente a comunicação com o Módulo LoRa e a atuação. O microcontrolador usado foi o Atmega328p da Microchip [17];
- Módulo LoRa: recetor e transmissor de informação com base no protocolo de comunicação LoRaWAN. Foi utilizado o RFM95W da HopeRF [18];
- *Power Supply*: Conjunto de 4 pilhas *AA* de 1.5 V, alimentação de 6 V;
- *Motor*: Motor DC de 6 V;
- Antena LoRa;
- Drive do Motor: circuito baseado numa ponte-H de transístores *NPN* e *PNP* com a função de acionar o motor responsável pela abertura e fecho da fechadura;
- Amplificador Operacional *Rail-to-Rail*: A sua função é amplificar os sinais de controlo do motor provenientes do microcontrolador, visto que este é operado a 3.3 V e o motor é operado a 6 V. Para o efeito foi utilizado o integrado TLV226 [19];
- Regulador de Tensão *Low Drop Out*: Por forma a garantir portabilidade do sistema, este é operado por uma bateria de 6 V. Ora, a tensão de alimentação máxima do Módulo LoRa é 3.3 V, pelo que é necessário um dispositivo que regule a tensão de 6 para 3.3 V. Para o efeito foi utilizado o LM317 [20].

Na Figura 4.2 é apresentado um diagrama de blocos que ilustra a forma como estes componentes estão ligados.

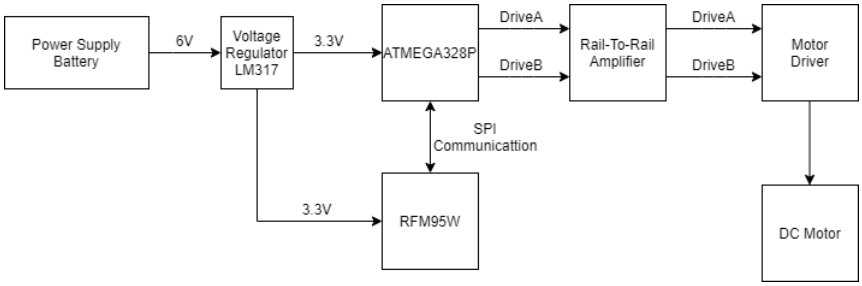


Figura 4.2: Diagrama de Blocos Hardware

De referir que, na Figura 4.2, estão representados, em termos de alimentação, apenas os módulos principais do sistema, ou seja, o microcontrolador ATMEGA328p e o módulo LoRa RFM95, sendo que, logicamente, todos os circuitos de acondicionamento de sinal são alimentados, a 5 Volts, como poderá ser constatado nas secções seguintes. Os sinais *DriveA* e *DriveB*, são os provenientes dos pinos do microcontrolador que fazem o controlo do motor DC. Detalhando, seguem-se, nas próximas secções, alguns esquemas dos circuitos para se perceber melhor como são efetuadas as ligações.

4.2 Ligação entre o Microcontrolador e o Módulo LoRa

Como já ilustrado na Figura 4.2, o microcontrolador e o Módulo LoRa comunicam através do protocolo de comunicação SPI, ou seja, através dos pinos Master Input Slave Output. Comunicação SPI (MISO), Master

Output Slave Input. Comunicação SPI (MOSI), CLK (*Clock Signal*) e SS (*Slave Select*). Para além destes, também é utilizado o pino `PB2` do microcontrolador para ligar ao *Reset* do módulo e três pinos (`PD2`, `PD5`, `PD6`) que estão conectados aos pinos `DIO0`, `DIO1` e `DIO2` que são utilizados para a implementação do protocolo de comunicação LoRaWAN. De realçar que, são suficientes os 3 pinos e, cada um tem uma finalidade específica, o `DIO0` tem o propósito de possibilitar a implementação do protocolo LoRaWAN com o ATMEGA328p, o `DIO1` para a utilização do modo de modulação dos dados LoRa e finalmente, o `DIO2` para modulação FSK, caso se pretendesse formatar os dados dessa forma.

De referir ainda que os pinos de alimentação – VDD e GND – estão ligados a 3.3 V e massa, respectivamente.

O protocolo de comunicação SPI é muito utilizado no contexto de ligação de dispositivos periféricos a microcontroladores. Neste caso, é importante salientar que esta é uma comunicação *Master-Slave* bidirecional em que o papel de *Slave* é desempenhado pelo módulo LoRa RFM95 e o papel de *Master* é, por sua vez, desempenhado pelo microcontrolador Atmega328p. Ainda no que concerne à comunicação série, de referir que, o microcontrolador possui apenas uma interface SPI, o que causa um inconveniente pois para injeção do *bootloader* no microcontrolador também é utilizada esta interface. Esta questão foi contornada por via de uma pequena modificação ao nível do hardware que será detalhada, neste relatório, aquando da apresentação dos esquemas de ligação.

O esquema da Figura 4.3 seguinte ilustra estas ligações:

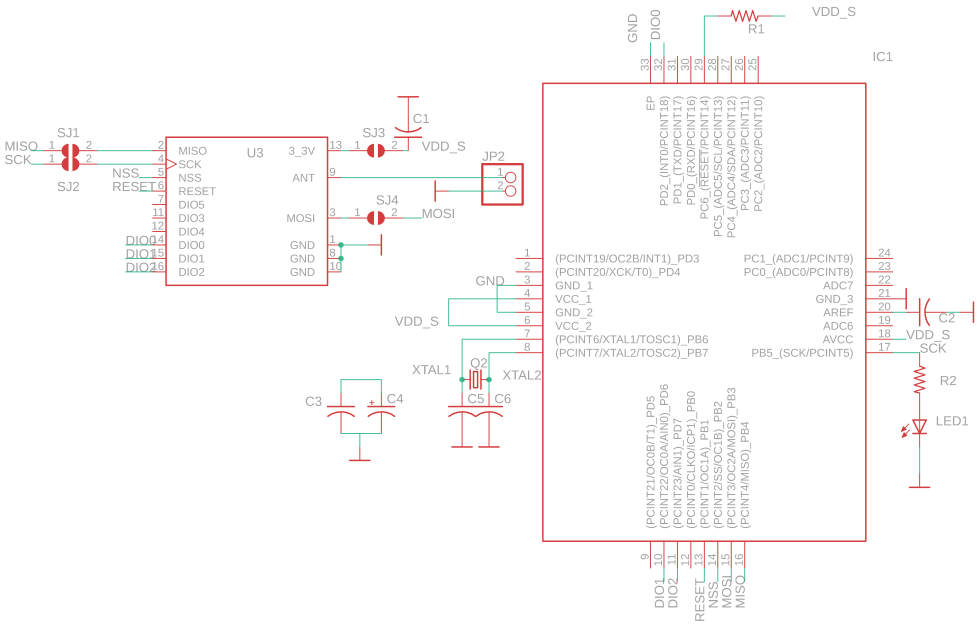


Figura 4.3: Esquema Ligação entre o Módulo LoRa e o Microcontrolador

Como se pode ver na figura acima, estão definidas as ligações que suportam o protocolo de comunicação SPI entre os dois dispositivos, sendo elas MOSI, MISO, SCK e NSS. Para além destas, chama-se a atenção para a alimentação – VDD_s e GND.

No que diz respeito a VDD_s, esta resulta da tensão à saída do circuito de proteção, o qual será detalhado mais à frente. Mas, como já referido anteriormente, este valor é de 3.3 V.

Em relação às resistências presentes no esquema, `R1` é um resistência de *pull-up*, com o valor de 10 kΩ, que garante que o pino de `Reset` do microcontrolador está sempre à tensão VDD_s. A não ligação deste pino à tensão de alimentação, levaria a que o microcontrolador tivesse um comportamento instável, visto que estaria suscetível a sofrer *resets* sucessivos.

Por sua vez, a resistência `R2` tem o valor de 150 Ω, sendo que a sua função é proteger o `LED1`, visto que se este estivesse ligado diretamente poderia queimar.

Os condensadores `C1`, `C2`, `C3` são cerâmicos e têm o valor de 100 nF. Estes condensadores servem, essencialmente, para filtrar ruído presente nas pistas de alimentação, sendo que para além destes é utilizado um condensador eletrolítico (`C4`), cujo valor é 10 µF.

Por último, referir que os componentes `SJ1`, `SJ2`, `SJ3` e `SJ4` correspondem aos *Solder Jumpers* cujo objetivo é permitir que o *bootloader* seja instalado no microcontrolador – o que não é possivel se o módulo LoRa estiver ligado às linhas de SPI. Isto será detalhado mais à frente neste capítulo.

4.3 Acondicionamento de Sinal

Em relação à tensão de alimentação há um requisito importante que se prende com a tensão máxima que o módulo RFM95 suporta que é 3.7 V. Como o circuito é alimentado a partir de 6 V é necessário introduzir um regulador de tensão.

O regulador de tensão escolhido para o efeito é o integrado LM317, que possui o seguinte esquema de ligação. Este possui 3 terminais, sendo eles a entrada, o `Adjust` e a saída. A Figura 4.4 representa a topologia do integrado bem como a sua aplicação típica [20].

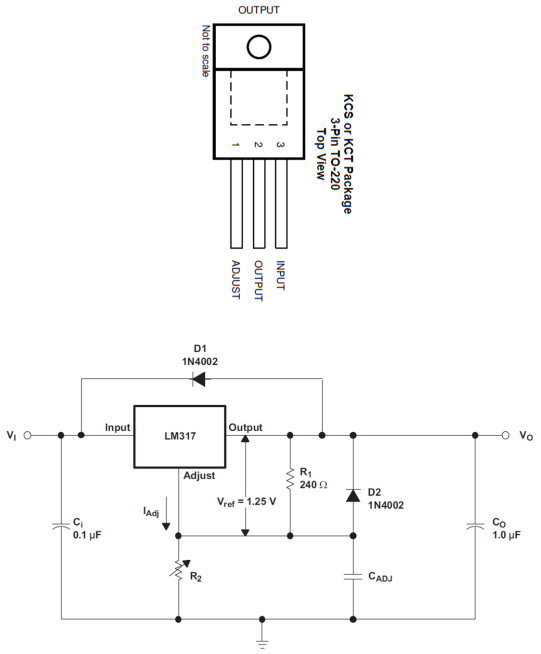


Figura 4.4: LM317: Pinout e Aplicação Típica

A tensão de saída pode variar entre os 1.25 V e os 37 V e é dada por:

$$V_o = V_{REF} \times \left(1 + \frac{R_2}{R_1}\right)$$

V_o	Tensão de Saída
V_{REF}	Tensão entre o Pino `Output` e o Pino `Adjust` 1.25 V
R_1	Resistência entre o Pino `Adjust` e o Pino `Output` 220 Ω
R_2	Resistência entre o Pino `Adjust` e a Massa

De referir também que, a tensão de saída não depende da tensão de alimentação/entrada, pelo que, apenas variando o valor de R_2 se consegue levar a tensão de saída para o valor desejado.

Ora, considerando que se quer uma tensão de saída de 3.3 V e, aplicando a equação anterior, tem-se:

$$\begin{aligned} V_o &= V_{REF} \times \left(1 + \frac{R_2}{R_1}\right) \\ V_o &= 3.3\,V \\ R_2 &= 360\,\Omega \\ R_1 &= 220\,\Omega \end{aligned}$$

Tal como anteriormente, importa detalhar o esquema de ligações implementado para este integrado, que está representado na Figura 4.5.

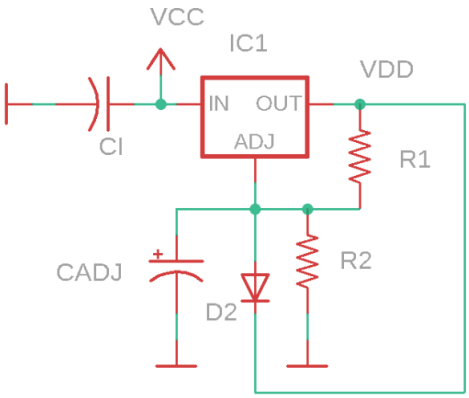


Figura 4.5: Esquema LM317

Sendo assim, importa referir que o díodo `D2` é um `1N4007` e tem a função de providenciar proteção à saída do regulador, em caso de descarga do Condensador `C_ADJ`. Este condensador, por sua vez, garante que o impacto do s*ripple* na tensão de saída é baixo, sendo que foi utilizado um condensador eletrolítico de 100 µF. De realçar que, o circuito representado na Figura 4.4 é uma aplicação típica do integrado `LM317`, sendo que, por exemplo, foi decidido não incluir, no circuito da Figura 4.5, o condensador `Co` e o díodo `D1`.

De notar também, que, na Figura 4.5 `VCC` corresponde à tensão de entrada do circuito – `Vi` na Figura 4.4 – e possui um valor de 6 V e `VDD` corresponde à tensão de saída – `Vo` na Figura 4.4 – e possui um valor de 3.3 V. Por fim, reforçar que as resistências `R1` e `R2`, têm o valor de 220 Ω e 360 Ω, respectivamente.

Não saindo do acondicionamento de sinal, importa perceber a forma como foi garantido outro requisito do sistema. Ora, as fechaduras existentes possuem um pequeno motor DC de 6 V que dependendo da tensão aos seus terminais faz com que a fechadura/porta seja aberta ou fechada.

Aqui o aspeto importante é que este motor requer 6 V para rodar num determinado sentido e −6 V para rodar no sentido inverso. Sendo assim, é necessário transformar o nível de tensão de operação do microcontrolador – que está definido como 3.3 V devido à tensão de alimentação máxima do módulo LoRa – em 6 V para que, através de dois pinos, se consiga acionar o motor num sentido (para desbloquear a fechadura) e no outro sentido (para voltar a bloquear a fechadura).

Como o motor tem de ser acionado com a voltagem máxima da alimentação, não se pode usar Amp-Ops normais, porque a tensão de saída destes tipicamente fica 2V abaixo da sua tensão de alimentação. Assim, usámos dois Amp-Ops rail-to-rail, onde as tensões de saída e entrada podem igualar a de alimentação.

A Figura 4.6 apresenta a topologia do integrado em causa:

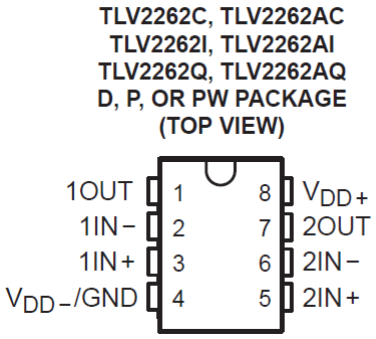


Figura 4.6: TLV226

Então, foram montados os dois Amp-Ops numa configuração não-inversora, isto porque se pretende um ganho positivo. Esta configuração tem um ganho dado por:

$$V_o = V_i \times \left(1 + \frac{R_2}{R_1}\right)$$

V_o	Tensão de Saída 6 V
V_i	Tensão de entrada 3.3 V
R_1	Resistência entre a entrada inversora e a massa
R_2	Resistência entre a entrada inversora e a saída

De notar que estes dois dispositivos são alimentados a 6 V. Pretende-se então, obter uma tensão à saída de ambos de 6 V, sendo que à entrada temos uma tensão que pode ser no máximo de 3.3 V – Pinos do Microcontrolador que vão acionar a subida/descida do motor.

Sendo assim, pela equação acima temos:

$$\begin{aligned} V_i &= 3.3\,V \\ V_o &= 6\,V \\ \left(1 + \frac{R_2}{R_1}\right) &= 1.82 \\ \left(\frac{R_2}{R_1}\right) &= 0.82 \\ R_2 &= 8.2\,k\Omega \\ R_1 &= 10\,k\Omega \end{aligned}$$

Com os valores para as resistências calculados, apresenta-se a Figura 4.7, com a ilustração das ligações efetuadas.

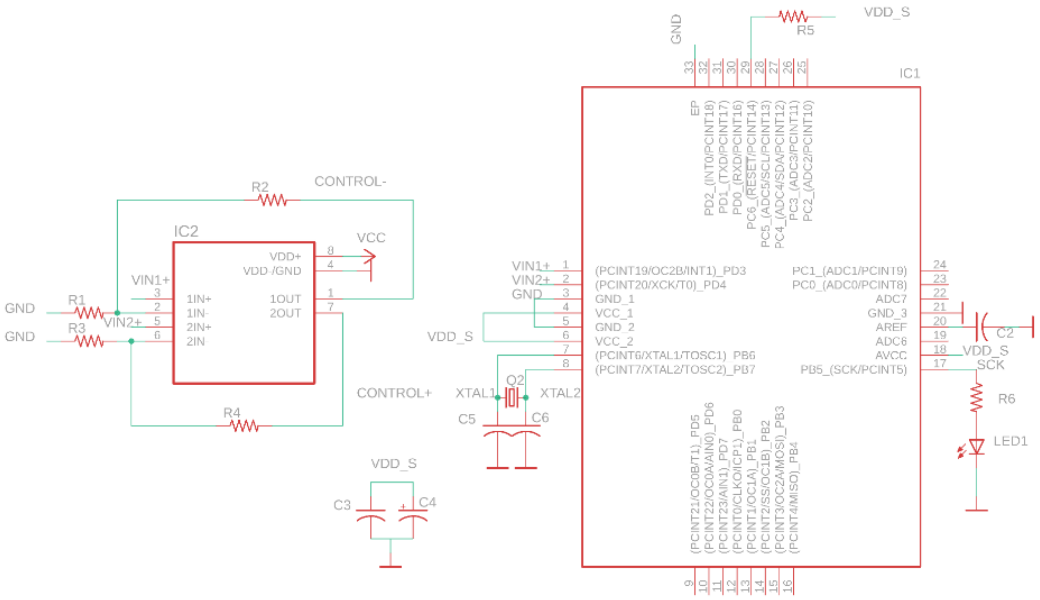


Figura 4.7: Esquema de Ligação entre o integrado TLV226 e o microcontrolador ATmega

De referir que este circuito possui duas entradas, `VIN1+` e `VIN2+`, que estão ligadas aos pinos de microcontrolador responsáveis pelo controlo do motor. Para além disto, registam-se duas saídas, `CONTROL-` e `CONTROL+`, que correspondem às entradas do próximo circuito a analisar, a Ponte-H.

Na Figura 4.7 está identificado o integrado TLV226 (`IC2`), este possui dois Amp-Ops que, como já referido estão montados numa configuração não-inversora. Sendo que as resistências `R1` e `R3` são de 10 kΩ e as resistências `R2` e `R4` são de 8.2 kΩ.

Desta forma, quando as linhas `VIN1+` e `VIN2+` estiverem com 3.3 V, teremos as linhas `CONTROL-` e `CONTROL+` com 6 V.

Explicado o circuito de amplificação dos sinais do microcontrolador, segue-se a apresentação do circuito que permite a atuação do motor DC.

Ora, a lógica subjacente ao funcionamento do motor DC utilizado é a seguinte:

- 6 V aos terminais do motor: este gira num determinado sentido, que faz com que a patilha desça, bloqueando a `latch` e consequentemente fechando a porta;
- −6 V aos terminais do motor: este gira no sentido contrário, fazendo com que a patilha suba, desbloqueando a `latch` e abrindo a porta.

Para que isto seja exequível é necessário um circuito que, a partir de uma tensão DC de 6 V, consiga colocar na saída essa tensão e o seu simétrico. Para isto recorreu-se a um circuito tipicamente utilizado para fazer acondicionamento do sinal em motores DC, que é a Ponte-H.

Concretizando, o esquema do circuito utilizado apresenta-se na Figura 4.8.

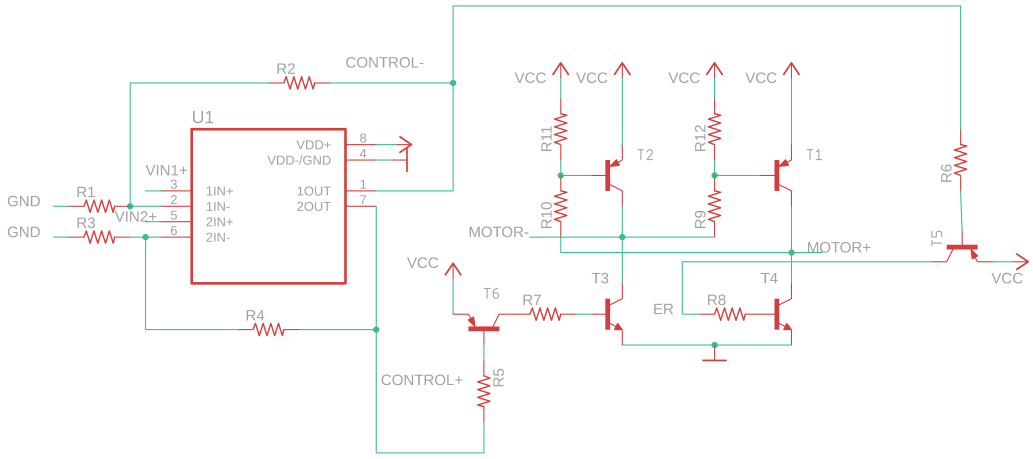


Figura 4.8: Esquema Ponte-H Motor

Ora na Figura 4.8 está representado o circuito que é responsável pelo acondicionamento do sinal para o motor DC utilizado.

Por um lado, temos TLV226 (`IC2`) que amplifica os sinais vindos do microcontrolador (`VIN1+` e `VIN2+`), dos 3.3 V para os 6 V. Para além deste, a referida Ponte-H que é responsável por, em função dos sinais `Control+` e `Control-`, colocar 6 V num dos terminais do motor (e massa no outro) e vice-versa. Assim conseguimos colocar 6 V ou -6 V, sendo que a tensão positiva faz com que o motor bloqueie a fechadura e a tensão negativa faz com que esta seja desbloqueada.

Como tal, é importante referir que os transístores apresentados são BJTs, do tipo pnp (`T1`,`T2`,`T5` e `T6`) e do tipo npn (`T3` e `T4`). Sendo que o principio base do funcionamento deste circuito baseia-se no corte ou ativação dos transístores.

Concretizando, se, por exemplo, na linha `Control+` tivermos um nível de tensão de 6 V e na linha `Control-` tivermos massa (`gnd`). Olhando para o comportamento dos transístores `T6` e `T5`, teremos o primeiro ao corte e o segundo a conduzir. Isto significa que, na base de `T3` teremos 0 V e na base de `T4` teremos 6 V. Relembrando que estes dois transístores são do tipo npn, isto resultará na condução de `T4` e no corte de `T3`. Sendo assim, como estão ligados, este estado leva a que `T2` esteja a conduzir e `T1` esteja ao corte.

No que concerne a resultados práticos destes estados dos transístores, temos que, na situação detalhada, a linha `Motor+` estará com 0 V e a linha `Motor-` estará com 6 V. Reforçando, que `Motor+` corresponde ao terminal positivo do motor e `Motor-` ao terminal negativo.

Sendo a diferença de potencial aos terminais do motor será de -6 V, pelo que o motor desbloqueará a fechadura.

Sendo que o estado dual das linhas `Control+` e `Control-` – 0 V para `Control+` e 6 V para `Control-` – levará a que o motor gire na direção contrária pelo que a fechadura será bloqueada.

Por fim, referir que os valores das resistências são: 5.6 kΩ para `R11` e `R12`, 150 Ω para `R10` e `R9` e 1 kΩ para as restantes resistências.

Por fim, realçar que os transístores são `BC847` (NPN) e `BC857` (PNP).

Posto isto, importa olhar para um circuito de proteção para o módulo LoRa. Este, como já referido, tem uma tensão de alimentação máxima de 3.7 V, então é conveniente que no sistema esteja implementado um circuito que corte a alimentação ao módulo sempre que, por algum motivo inesperado, a tensão de alimentação do módulo suba. Para isso utilizou-se um circuito baseado num díodo Zener de 3.3 V e dois transístores PNP, o esquema do circuito está representado na Figura 4.9.

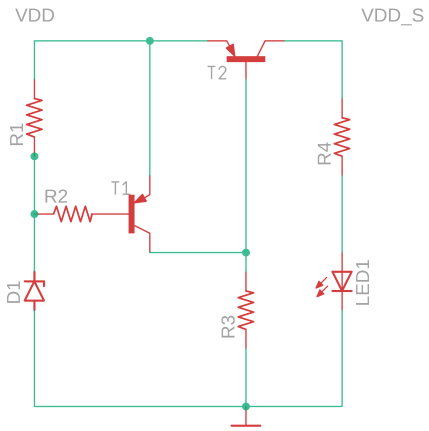


Figura 4.9: Esquema Proteção Alimentação

Assim, o zener força a tensão na base do transístor `T1` a 3.3 V. Sendo assim, se a tensão `VDD` for superior a 3.3 V, este transístor fica cortado, assim como o `T2`, desta forma, impede-se que a saída do circuito – `VDD_S` – ultrapasse esses valores de tensão. De realçar que, `VDD_S` é a tensão que vai alimentar o módulo LoRa, que tendo restrições no que concerne à tensão de alimentação máxima, fica salvaguardado. Posto, isto referir que os dois transístores (`T1` e `T2`) são `BC857`, e que os valores das resistências são: `R1` – 2 kΩ, `R2` – 1 kΩ, `R3` – 5.6 kΩ e `R4` – 100 Ω. O `LED1` apenas serve para indicar que está a ser fornecido o nível de tensão adequado ao circuito.

4.4 Projeto das placas PCB

De referir que, numa fase inicial, o protótipo foi montado e testado em placa branca e posteriormente foi desenhada a PCB no programa EAGLE. De referir que foi desenvolvida, uma PCB de duas camadas, sendo que elas estão identificadas na Figura 4.10, com a cor vermelha para camada superior e com a cor azul para a camada inferior.

De acordo com o proposto pela organização, a PCB deve ser incluída nas fechaduras existentes, sendo que para tal, a dimensão máxima desta deverá ser de aproximadamente 40 × 40 mm. Este objetivo foi conseguido, sendo que a Figura 4.10 ilustra o resultado final do desenho da placa no programa EAGLE.

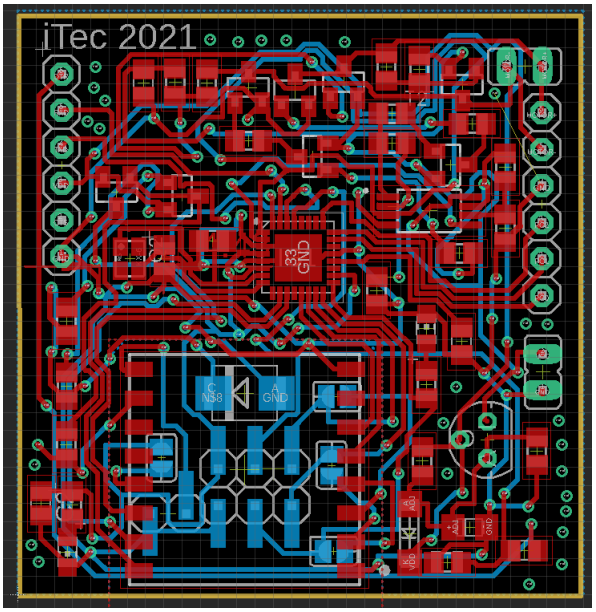


Figura 4.10: PCB layout EAGLE

Posto isto, segue-se a descrição do processo de desenho da antena para comunicações LoRa, tanto para o protótipo em placa branca, como para a PCB.

Sendo que, para o protótipo em placa branca, foi usado um fio com um diâmetro de 2.54 mm. A utilização desta solução para antena, prende-se essencialmente, com a facilidade de acesso a fios, conseguindo uma solução prática e simples de implementar. Sendo que, como já referido para o circuito em PCB, por força da necessidade de obter uma solução mais compacta, para que pudesse ser incluida nos equipamentos da empresa, optou-se pelo desenvolvimento de uma pequena antena em PCB.

Para o cálculo do comprimento da antena, seguiu-se o princípio base da propagação de ondas rádio, sendo que o comprimento de onda é tipicamente dado por:

$$\lambda = \frac{c}{f}$$

λ	comprimento de onda associado ao tipo de radio-frequência
c	velocidade de propagação no vácuo
f	frequência de propagação

Ora, tendo em conta a comunicação LoRa, podemos quantificar estas grandezas, da seguinte forma:

$$\begin{aligned} \lambda &= \frac{\lambda_0}{4} \\ c &= 3 \times 10^8 \,\mathrm{m/s} \\ f &= 868 \,\mathrm{MHz} \end{aligned}$$

De realçar que, λ_0 corresponde ao comprimento de onda no vacúo.

Sendo assim, para calcular o comprimento da antena, temos de obter o parâmetro λ:

$$\begin{aligned} \lambda_0 &= \frac{3 \times 10^8}{868 \times 10^6} = 0.346 \,\mathrm{m} \\ \lambda &= \frac{\lambda_0}{4} = 0.086 \,\mathrm{m} \end{aligned}$$

Deste modo, foi utilizado uma antena, na forma de fio, com o diâmetro de 2.54 mm e de comprimento 8.6 cm.

De seguida, descreve-se o projeto da antena em PCB.

Um dos aspetos mais importantes no que diz respeito ao projeto de antenas, especialmente em PCB, é garantir adaptação a 50 Ω.

De realçar, que tipicamente esta adaptação consegue-se por via da utilização de componentes passivos, cujo valor varia com a frequência.

Ainda que, se possa alcançar o mesmo por via de outra técnica como a utilização de *stubs*.

A topologia de circuito de adaptação está representada na Figura 4.11.

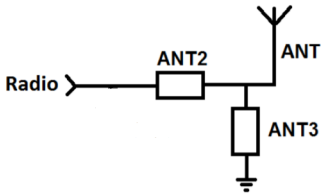


Figura 4.11: Circuito de adaptação da antena

Ora, o circuito de adaptação utilizado é composto por um condensador (ANT2) e uma bobina (ANT3), cujos valores são 1 pF e 12 nH, respectivamente.

Desta forma, é garantida adaptação à frequência de trabalho (868 MHz) a 50 Ω.

Na Figura 4.12 é possível verificar a placa desenhada no EAGLE para a antena.

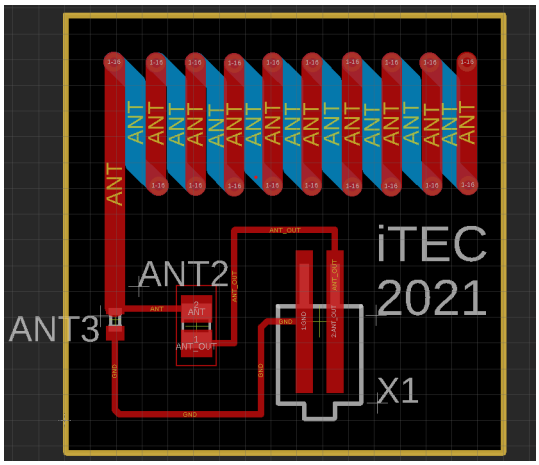


Figura 4.12: Desenho da Antena em PCB no EAGLE

De realçar que na Figura 4.12, a bobina está identificada com a *label* `ANT3`, sendo que como representado na Figura 4.11, os seus terminais estão ligados à massa e à pista com a label `ANT`, enquanto que o condensador está identificado com a *label* `ANT2` e os seus terminais estão ligados à pista `ANT` e ao conector `X1` – que vai ligar ao respetivo pino de antena do RFM95.

4.5 Firmware

Para que o sistema realize o propósito desejado é necessário injetar *firmware* no microcontrolador Atmega328p. Nesse sentido, o primeiro passo é carregar o *bootloader* para o `chip` ATMEGA328p. O *bootloader* é como que um sistema operativo que permitirá, quando instalado, que o microcontrolador seja programável. Neste programa, estão definidos todos os portos, módulos, protocolos de comunicação, etc.

Ora há várias formas para desencadear este processo, sendo que, no caso, foi utilizado um Arduino onde foi carregado o *bootloader* e ao qual se ligou, através do protocolo SPI, o microcontrolador Atmega328p.

A interface utilizada foi a plataforma de desenvolvimento Arduino IDE. Sendo assim, carregou-se no Arduino o programa referido e posteriormente selecionou-se as características/modo de funcionamento pretendidas para o microcontrolador, as quais: Modelo, Frequência e Tipo de Relógio, BOD (*Brown-Out Detector*), EEPROM (*retained/not retained*) e LTO.

Sendo que, destes parâmetros, dois são particularmente importantes para o sistema em causa, são eles a definição do Relógio e a desativação do BOD. A Figura 4.13 ilustra o processo referido acima.

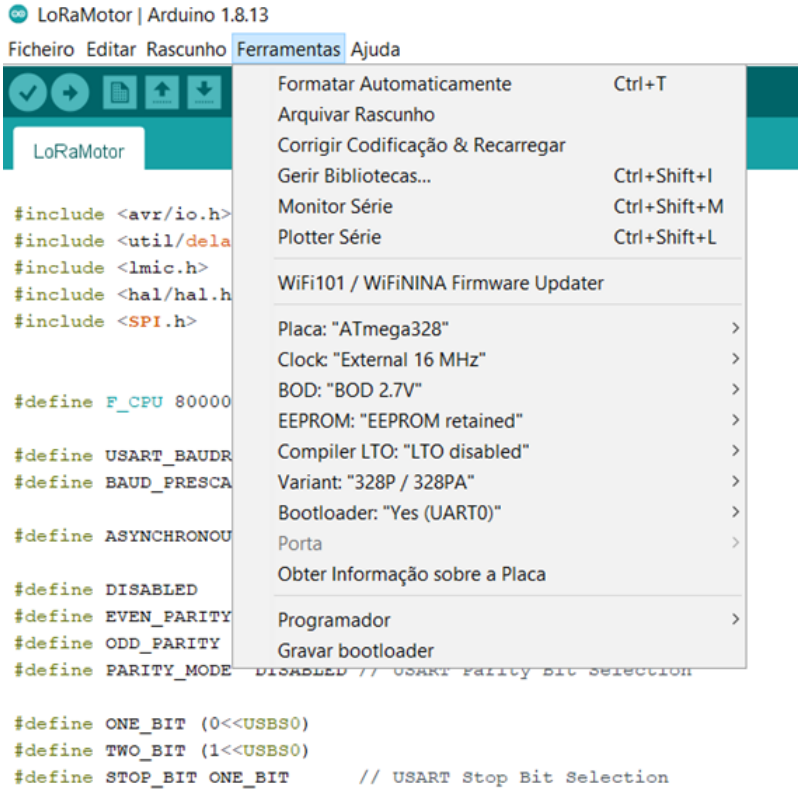


Figura 4.13: Configuração dos Parâmetros para o *burn* do *Bootloader*

Ora, o relógio será imposto externamente por via da inclusão de um cristal de 8 MHz. A escolha desta frequência prende-se com o facto de esta estar intimamente ligada com a tensão de alimentação do microcontrolador, isto é, ao utilizar frequências de relógio superiores a tensão de alimentação mínima do Atmega328p é maior. Por exemplo, para utilizar um relógio de 16 MHz teria de se alimentar o micro a pelo menos 4.5 V, o que, face ao exposto em relação ao módulo LoRa não é exequível.

O BOD é um mecanismo que pode estar ou não ativo, que faz o *reset* do microcontrolador sempre que a sua tensão de alimentação descer de um determinado nível de tensão, tipicamente os valores para os limiares são 2.7, 1.8 e 4.3 V.

Sendo que, no caso este parâmetro foi desativado, pois devido ao facto do circuito ser alimentado por um conjunto de pilhas, a tensão aos seus terminais varia com o aumento da corrente que o circuito necessita para operar corretamente, e como é lógico, não seria desejável ter-se sucessivos *resets* sempre que houvesse um pico de corrente – por exemplo, quando o motor é acionado há no circuito um pico de corrente, principalmente na sua fase ascendente. Voltando ao processo de instalação do *bootloader* no Atmega328p, foram implementadas as ligações presentes no esquema da Figura 4.14.

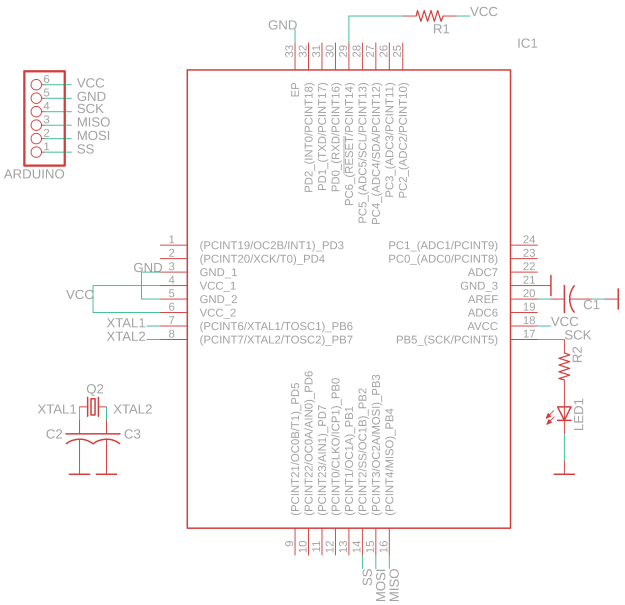


Figura 4.14: Topologia de Ligação entre o Arduino e o chip Atmega328p para instalação do Bootloader

Como ilustrado na Figura 4.13, para instalar o *bootloader* é necessário conectar os pinos SPI do arduino aos pinos SPI do ATMEGA328p, para além disto, os pino alimentação do Arduino também se ligam aos do microcontrolador. De forma a que o microcontrolador não esteja sempre a sofrer resets, tem que se colocar uma resitência ligada aos 5 V. Para garantir, que se tem uma fonte de relógio para o ATMEGA328p, conecta-se um cristal de 8 MHz aos pinos `OSC1` e `OSC2` do mesmo. De realçar que, na instalação do *bootloader*, se pode especificar que o microcontrolador use o seu oscilador interno, assim já se poderia remover o cristal.

Como referido na secção anterior, quando se abordou o protocolo de comunicação SPI implementado entre o Atmega328p e o módulo LoRa, para injeção do *bootloader* também é utilizada a dita interface SPI.

Pelo que, como está representado na Figura 4.10, foram incluídos *solder jumpers* na camada inferior da placa, estes garantem que, por defeito, a ligação via SPI entre o microcontrolador e o módulo LoRa não está implementada.

Assim, é permitida a injeção do *bootloader* no Atmega328p, por via do conector de 6 pinos que está localizado precisamente na camada inferior da placa. Desta forma, uma vez instalado o *bootloader* basta colocar um pouco de solda nesses *solder jumpers*, por forma a efetivar cada uma das ligações entre o módulo e o microcontrolador. Em alternativa, e pensando numa forma mais prática de se realizar o procedimento, poderia-se também colocar jumpers amovíveis.

Assim, o hardware fica pronto a ser programado, e uma vez concluído esse processo os dois dispositivos ficam aptos a comunicar.

A injeção de *firmware* no microcontrolador foi feita através de um cabo FTDI, que não é mais do que um conversor TTL-Série, que permite que o PC comunique por UART com o *chip*. Para tal, também foi utilizada a plataforma Arduino IDE. O referido conversor tem 6 pinos:

- VCC: tensão de alimentação, que pode ser 3.3 ou 5.5 V, mediante a disposição de um *jumper*;
- GND: massa;
- RXD: Pino de receção de informação Atmega328p → FTDI (PC), que deve ser conectado ao pino TXD (envio de informação) do microcontrolador – `PD1`;
- TXD: Pino de transferência de informação FTDI (PC) → Atmega328p, que deve ser conectado ao pino RXD (Receção de informação) do microcontrolador — `PD0`;

- DTR: Pino de *Reset*, liga-se ao pino de *reset* do microcontrolador — `PC6`;
- CTS: Pino de controlo do fluxo de informação, *Clear-To-Send*, sendo que no caso não é necessário conectar ao microcontrolador.

As ligações detalhadas anteriormente, podem ser confirmadas no esquema da Figura 4.15.

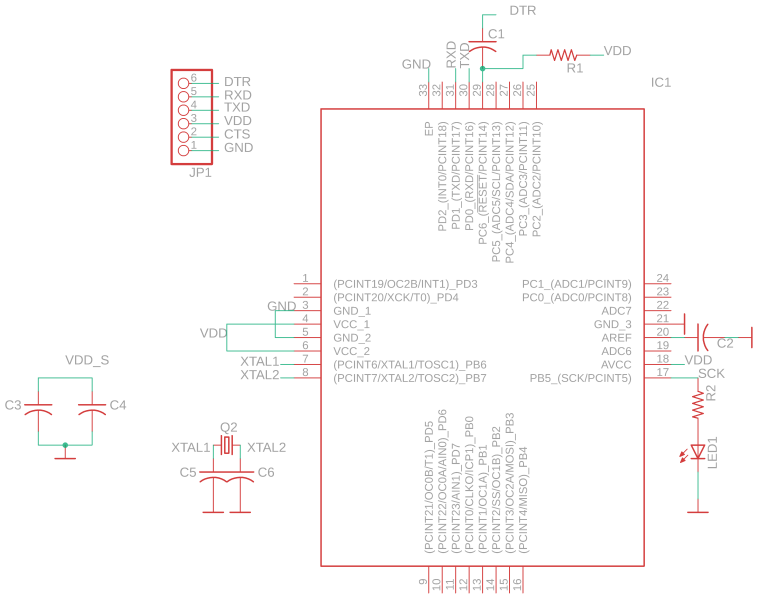


Figura 4.15: Topologia de ligação entre o conversor Série-TTL (cabo FTDI) e o microcontrolador

Desta forma, é possível a partir da plataforma Arduino IDE carregar o *firmware* para o microcontrolador. Para que o protótipo em placa branca fique completo resta acrescentar: a alimentação, ou seja, a bateria de 6 V que em conjunto com o circuito do integrado LM317 será responsável por providenciar a alimentação de 3.3 V. O circuito de amplificação do nível de tensão de 3.3 V para 6 V (circuito do integrado TLV226) e finalmente, o circuito da Ponte-H que servirá de acondicionamento de sinal para ligação do motor.

Esta disposição, será posteriormente, replicada, para que com a PCB, o método de injeção do firmware seja o mesmo.

4.6 Testes realizados para determinar distância máxima de comunicação

Por forma a testar a capacidade de comunicação do circuito, em termos de alcance, foi realizado o seguinte teste. O Gateway foi deixado no escritório da *iTEC*, sendo que o circuito foi levado pelas redondezas, até se encontrar um ponto/zona onde se deixava de receber mensagens na plataforma TTN.

A Figura 4.16 representa o resultado desses testes.

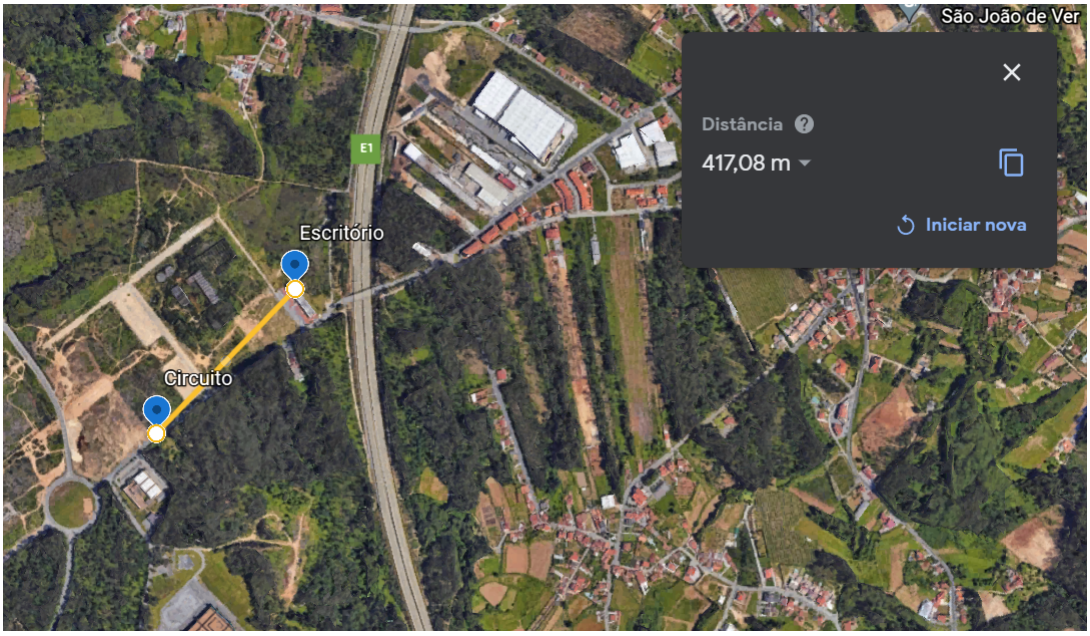


Figura 4.16: Resultado dos testes realizados acerca do alcance da comunicação LoRa

De referir que, em relação ao teste apresentado, o ponto marcado com a *label* `Escritório` indica a posição geográfica do Gateway, enquanto que a *label* `Circuito` referencia a posição (aproximada) do circuito LoRa. A distância apresentada é de aproximadamente 417 m.

Com esta questão abordada, finda-se a exploração da solução desenvolvida, do ponto de vista do hardware, segue-se, neste documento, a descrição dos procedimentos relativos à integração dos dados com a *cloud* e o desenvolvimento de uma interface gráfica do utilizador.

CAPÍTULO 5

Exploração da Solução: firmware e Integração com a *Cloud*

Neste capítulo será detalhado o processo de desenvolvimento do *firmware* e de integração com a *Cloud*.

5.1 Firmware

Para o desenvolvimento do *firmware* foi utilizada a linguagem de programação *C* e, mais uma vez, a plataforma Arduino IDE. O código desenvolvido tem duas funcionalidades principais:

- Implementação do protocolo LoRaWAN;
- Ativação e Controlo da Atuação do Motor.

De referir, que, para implementação do protocolo LoRaWAN foi incluída a integração com a plataforma TTN, onde é possível, registar aplicações LoRa e seus *Devices*, bem como visualizar o histórico de downlinks e uplinks que vão ocorrendo ao longo do tempo. A integração desta plataforma será documentada mais à frente neste capítulo.

Em relação, novamente, ao *firmware*, importa referir que a implementação do protocolo de comunicação LoRaWAN pressupõe, entre outras, as seguintes definições:

- atribuição dos pinos do microcontrolador e suas funções, nomeadamente, o pino de *Slave Select* (NSS), Reset (RST) e DIO;
- definição do período em que são realizadas as transferências de dados `TXINTERVAL`;
- definição das chaves de segurança associadas ao dispositivo, as quais, a *Network Session Key* (NwkSKey), *Application Session Key* (AppSKey) e o *End-Device Address* (DevAddr);
- definição da mensagem a ser enviada (`myData`);
- definição dos fatores de espalhamento (*Spreading Factor*).

O trecho de código onde são feitas as definições referidas acima é apresentado de seguida:

```
#include <avr/io.h>
#include <lmic.h>

// LoRaWAN NwkSKey, network session key
// This is the default Semtech key, which is used by the early prototype TTN
// network.
static const PROGMEM u1_t NWKSKEY[16] = { 0x2C, 0x3D, 0xD9, 0x0A, 0x42, 0xC5, 0x85, 0xF7
,0x0E, 0x4E,0xAF, 0x3F, 0x18, 0x75, 0xF5, 0x29 };


// LoRaWAN AppSKey, application session key
// This is the default Semtech key, which is used by the early prototype TTN
// network.
static const u1_t PROGMEM APPSKEY[16] = { 0x90, 0x3B, 0x3C, 0xA3, 0x06, 0x9B, 0x16, 0x67
,0xD5, 0x66 ,0xBE, 0x52,0x74, 0x8D, 0x59, 0x3E };

// LoRaWAN end-device address (DevAddr)
static const u4_t DEVADDR = 0x26011A9C;

static uint8_t mydata[1] = {0x01};
static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).

const unsigned TX_INTERVAL = 10;


// TTN uses SF12 for its RX2 window.
LMIC.dn2Dr = DR_SF12;

// Set data rate and transmit power for uplink (note: txpow seems to be ignored
// by the library)

LMIC_setDrTxpow(DR_SF7,14);

// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {5, 6, 7},
};
```

De notar, que aos pinos referenciados do microcontrolador acresce os pinos constituintes do protocolo SPI (MISO, MOSI e SCK).

Ainda em relação à implementação do protocolo LoRaWAN, foram desenvolvidas as seguintes funções:

- `void setup()`: definição do estado dos pinos I/O do microcontrolador, inicialização da usart (comunicação Série) e da comunicação LoRa;

- `void onEvent (ev_t ev)`: gestão dos eventos gerados pelo módulo LoRa, entre os quais: processos de junção nas aplicações OTAA – `EV_JOINED` e `EV_JOINING`, estado das transferências de informação em ambos os sentidos (*Downlink* e *Uplink*) – `EV_RXCOMPLETE` e `EV_TXCOMPLETE`, entre outras;
- `do_send(osjob_t* j)`: Verifica se, num determinado momento, se encontra alguma tarefa de envio/receção de informação e agenda uma nova transferência de dados;
- `void loop()`: Ciclo infinito onde o programa corre.

5.2 Ligação ao TTN *(The Things Network)*

Para implementar a comunicação LoRa, para além do módulo RFM95, é necessário um Gateway LoRa e um plataforma/servidor onde seja possível visualizar a informação transmitida/recebida ao longo do tempo. Ora foi utilizado o Gateway LoRa Dragino LG01-P, e após configuração e associação a uma rede *Wi-Fi*, é possível através da plataforma TTN registar aplicações e dispositivos na rede LoRa.

São definidos, tipicamente, dois modos de ativação para cada dispositivo:

- OTA: a forma mais segura de cada se dispositivo se ligar ao Gateway, é realizado um processo de junção (*join*), no qual é atribuído um endereço ao dispositivo – DevAddr – e as chaves da sessão são negociadas;
- ABP: nesta forma de ligação não se desencadeia o processo de junção bem como da negociação das chaves de sessão, o que torna o processo, inerentemente mais suscetível em termos de segurança. Esta suscetibilidade, prende-se com o já referido, anteriormente, no Capítulo 3, de a não existência de processo de *join* fazer com que, um atacante, se conseguir provocar o reinicio de um dispositivo, consiga ter acesso às chaves e entrar na rede. Devido ao facto, de após reset, na configuração ABP, se usar as mesmas chaves.

Desta forma, foram criados na referida plataforma dois dispositivos, sendo que. os dois modos de ativação foram comparados em termos de desempenho.

A Figura 5.1 apresenta uma comparação entre a forma como os dados são recebidos na plataforma TTN, no caso de se utilizar, o modo de ativação ABP ou o OTA.

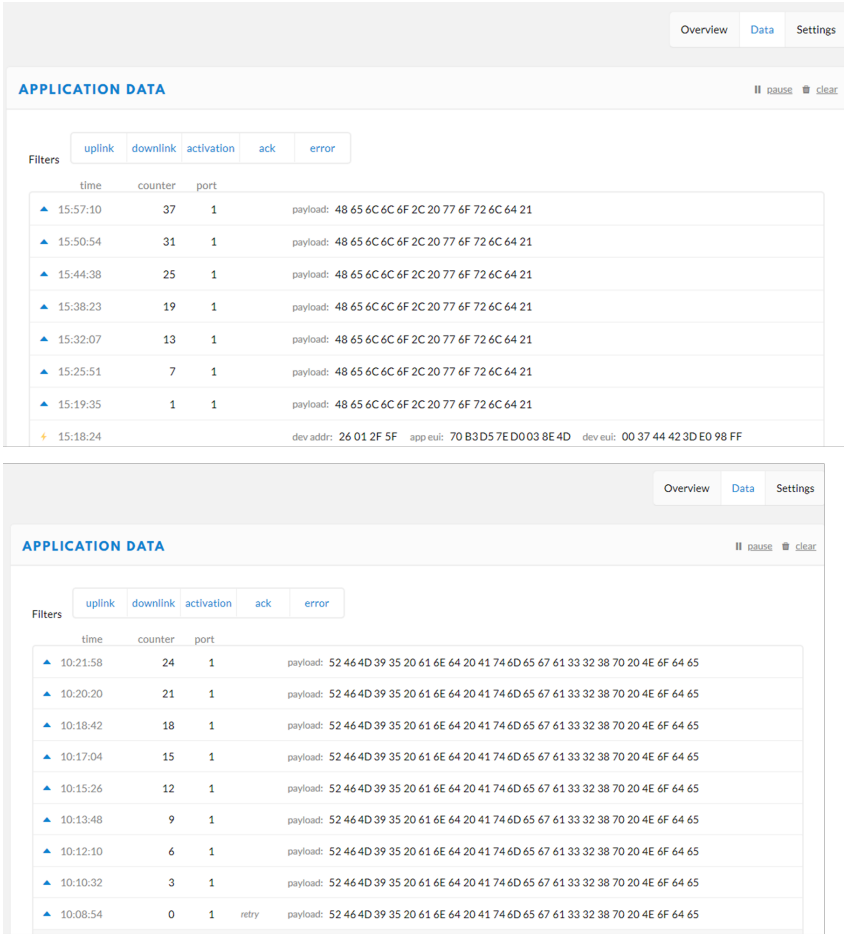


Figura 5.1: Comparação entre um dispositivo configurado para operar no modo ABP e OTAA

De realçar que na Figura 5.1, os dados que se encontram na parte inferior referem-se ao modo de ativação ABP e os da parte superior ao modo OTA.

A principal diferença, consiste no existência de um processo de *join*, sendo que como consequência disso, o modo de ativação OTA é mais lento a concluir a ligação ao Gateway, em comparação com o modo ABP.

No que concerne ao *Time-On-Air*, verifica-se que, no caso de se utilizar o método ABP, cada *frame* é recebido em cerca de 30 segundos. Já no OTA, verifica-se que este tempo é de cerca de 60 segundos para cada *frame*. Salvarguandando que, no *firmware* foi definido para o ABP um `TX_INTERVAL` de 30 segundos, para o OTA foi definido esse valor como 60 segundos. Assim, os valores "medidos" neste exemplo, estão condizentes com o que foi determinado no código, em cada caso.

Pelo que, se conclui que a utilização de um ou outro método de ativação não tem repercussões ao nível do desempenho esperado, em termos de comunicação, apenas, e como já referido, o método OTA se revela mais seguro em termos de implementação, por incluir a janela de *Join*, já referida.

De referir, que as mensagens enviadas/recebidas pelo Gateway são apresentadas na plataforma em formato American Standard Code for Information Interchange (ASCII) – *American Standard Code for Information Interchange*. Em termos do sistema implementado, importa perceber que o foco, no que toca ao fluxo de informação recairá essencialmente sob o downlink, ou seja o envio de mensagens do Gateway para o módulo LoRa.

Para tal, foi desenvolvida uma integração HTTP que, permitirá, enviar três tipos de mensagens/ordens para o módulo RFM95, as quais estão representadas Tabela 5.1.

Uma integração HTTP é criada quando se necessita de enviar mensagens para o Gateway que por sua vez as reencaminha para o dispositivo correspondente, sendo que a plataforma TTN disponibiliza várias

Mensagem	Ação
Unlock	Desbloqueia fechadura, passado x (configurável) segundos bloqueia
Unlock Always Open (*Unlock_AO*)	Desbloqueia fechadura permanecendo aberta
Reset Lock (Reset)	Realiza um Reset ao hardware em caso de problemas de conexão

Tabela 5.1: Mensagens enviadas e ações correspondentes

plataformas onde se podem fazer as referidas integrações pelo que a escolha recaiu na plataforma *pipedream*. Nesta plataforma é possível criar fontes de eventos às quais podem ser associados vários dispositivos ligados em rede LoRa ao TTN. Ora, para além de ser possível monitorizar todos os dados enviados (`POST`) e recebidos (`GET`) pelo Gateway, também está disponível no *pipedream* um *link* a partir do qual é possível realizar a dita integração HTTP.

Na Figura 5.2 apresenta-se o procedimento para realização de uma integração HTTP–TTN e Pipedream.

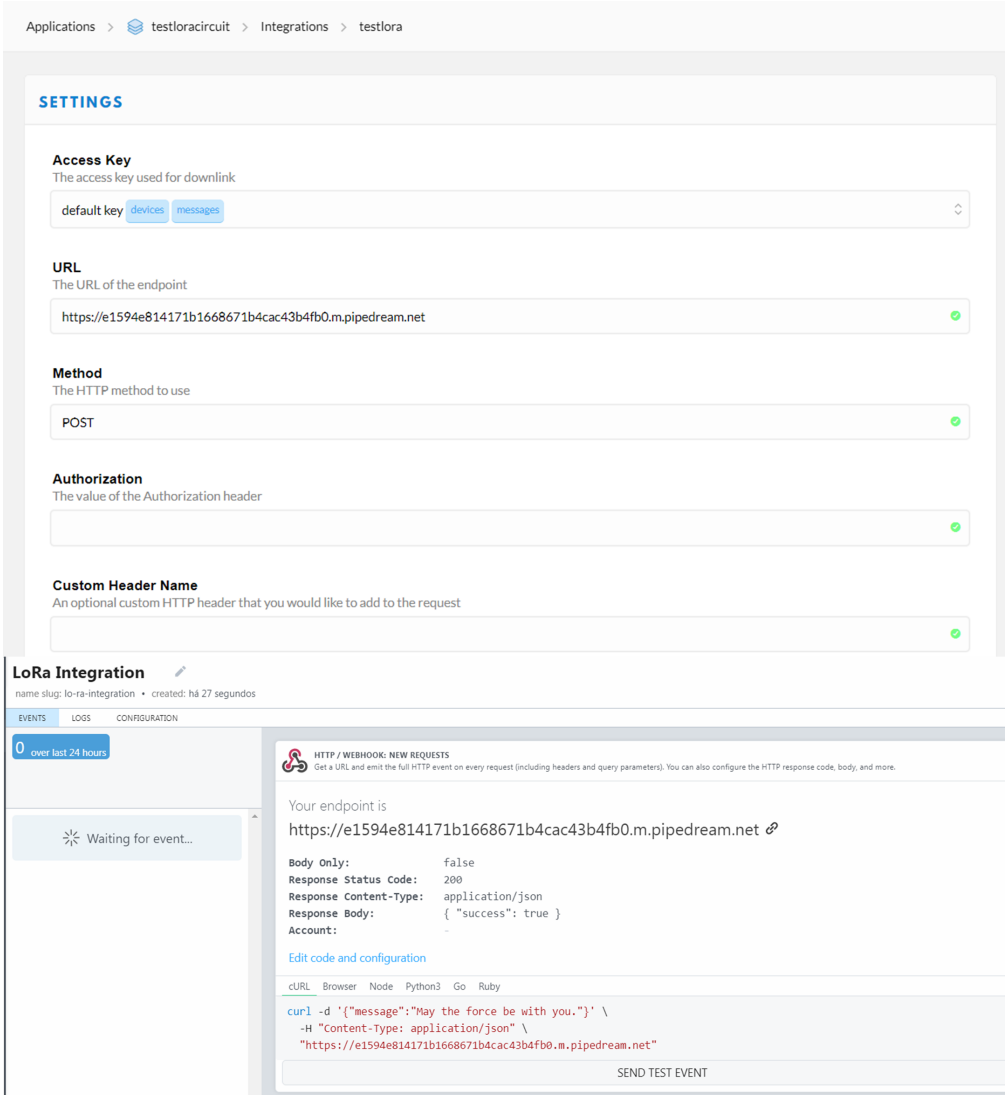


Figura 5.2: Definição dos parâmetros para integração HTTP na página TTN e Servidor *Pipedream*

Após a realização destas configurações, todos as transferências de informação realizadas no Gateway LoRa vão gerar eventos no servidor *pipedream*. Desta forma é possível não só monitorizar e visualizar os dados, como adquirir um *link* para fazer a integração HTTP, ou seja, um *link* – `downlink_URL` a partir do qual será possível enviar as mensagens para o módulo LoRa.

O objetivo é agora, criar uma interface onde o utilizador possa selecionar a ação que pretende despoletar no sistema. Esta interface será apresentada na próxima secção.

Desta forma o utilizador, poderá através de uma interface gráfica, selecionar qual a informação que quer enviar para a fechadura que despoletará a ação correspondente.

Como tal, o sistema está completo, sendo constituído pelo hardware detalhado no capítulo 4, onde corre o *firmware* que implementa o protocolo de comunicação LoRaWAN

5.3 Interface com o Utilizador

Para utilização como prova de conceito, foi desenvolvido um programa em python onde é possível actuar sobre o sistema de acordo com o que o utilizador pretender conforme apresentado na Tabela 5.1.

Este programa seguiu as funcionalidades/opções de utilização que o *software* da empresa possui, que são, essencialmente, as referidas acima. Desta forma, será possível provar que é exequível uma reprodução dos sistemas existentes na empresa através da implementação do protocolo de comunicação LoRaWAN. A referida interface do utilizador apresenta-se na Figura 5.3.

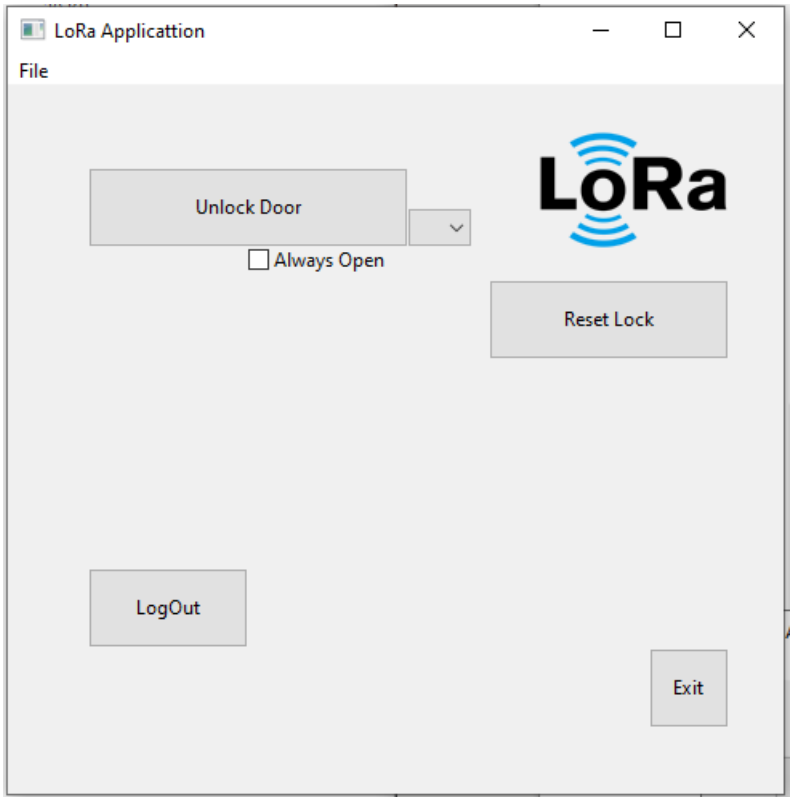


Figura 5.3: Interface do Utilizador

Portanto, o utilizador com a interface acima pode :

- Desbloquear a fechadura no modo normal e selecionar o tempo de abertura, premindo o botão *Unlock Door* e selecionar o tempo de abertura (*slide box*);
- Desbloquear a fechadura no modo sempre aberto, selecionando a caixa de verificação *Always Open*;

- Efetuar o *Reset* do hardware premindo o botão *Reset Lock*.

Esta interface foi desenvolvida em python e o envio das mensagens é realizado através das bibliotecas `urllib3`, `requests` e `json`. Sendo que as mensagens são codificadas em Base64 através do uso da biblioteca `base64`.

O seguinte trecho de código exemplifica este processo, no caso para envio da mensagem `"Unlock"`.

```
import requests
import json
import urllib3
import base64

message = "Unlock"
message = message.encode("ascii")
message_raw = base64.b64encode(message)

message_raw = message_raw.decode("ascii")
print(message_raw)

url = "https://integrations.thethingsnetwork.org/ttn-eu/api/v2/
down/testloracircuit/" +
      "testlora?key=ttn-account-v2.oE5uTqJw8Euy2OBbR67Db71J1NZ9r3uV82FkupNgbQY"
payload = {"dev_id" : "testlora","payload_raw":message_raw}
header = {"Content-Type":"application/json"}
response = requests.post(url,data = json.dumps(payload),
headers = header, verify=False)
```

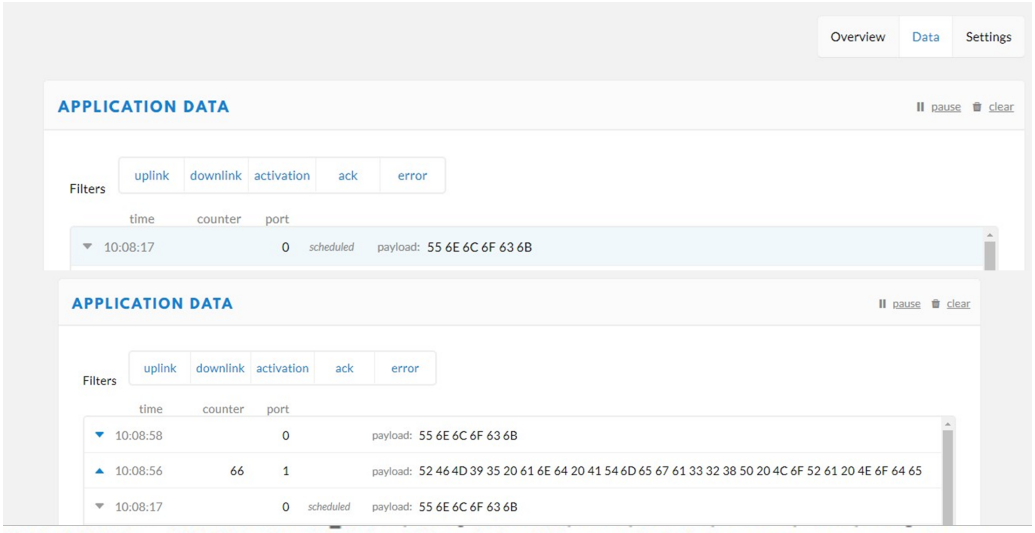
No que concerne ao trecho de código apresentado acima importa referir que:

- a instrução `message.encode("ascii")` converte a *string* `"Unlock"` para a correspondente em código ASCII;
- a instruções `base64.b64encode(message)` e `message_raw.decode("ascii")` são responsáveis pela transformação da *string* convertida anteriormente em base64. Esta é uma forma de codificação comum nas transferência de dados na *Internet*, sendo que é a utilizada no envio de mensagens para a plataforma TTN. Concretizando, por exemplo a *string* `"Unlock"` corresponde, em base64, a `"VW5sb2Nr"`;
- as restantes instruções consistem na preparação dos dados para envio da mensagem, sendo que é necessário especificar o `downlink_URL`, que foi obtido anteriormente na plataforma *pipedream*, bem como o *payload* e o *header* e finalmente utilizar o método `POST` para enviar o downlink. De referir ainda que, no *header* tem de ser especificado o identificador do dispositivo (`dev_id`) que foi criado no TTN bem como o conteúdo da mensagem (`payload_raw`).

De seguida, apresentam-se alguns testes realizados, estes consistem no premir dos botões da interface, confirmação da mensagem recebida na plataforma TTN e ação despoletada no monitor série do Arduino IDE.

Desta forma, podemos ver se realmente, as mensagens enviadas para o Gateway LoRa estão a ser recebidas pelo *hardware* e se a interpretação destas por parte do microcontrolador leva ou não à ação pretendida em cada caso.

Primeiramente, foi premido o botão *Unlock Door* e o resultado é apresentado na Figura 5.4.




```
10:08:58.709 -> 57119792: Received downlink, window=RX1, port=1, ack=0
10:08:58.758 -> 57119888: EV_TXCOMPLETE (includes waiting for RX windows)
10:08:58.803 -> Received 6 bytes of payload: Unlock
10:08:58.849 -> ACCESS GRANTED!!! Normal Operation Type
10:09:09.825 -> 57815582: engineUpdate, opmode=0x800
```

Figura 5.4: Premir do botão *Unlock Door* e consequentes eventos

A mensagem enviada é `"Unlock"` o que corresponde em código ASCII a `55 6E 6C 6F 63 6B` sendo essa a mensagem recebida no TTN. Esta mensagem é depois recebida pelo módulo LoRa e é feita, no *firmware*, a comparação do conteúdo com a *string* `"Unlock"`. O conteúdo da mensagem é então validada e a ação correspondente é despoletada – desbloqueio da porta no modo normal. Sendo que, como não foi selecionado nenhum tempo de abertura específico, esse tempo por definição é 2 segundos.

Sendo que a "saída" presente no monitor série do Arduino IDE corresponderá à ação despoletada pela mensagem recebido, no caso em que esta não corresponda a nenhuma das ações pré definidas, será apresentada a seguinte mensagem "*The received message has not lead to any action*".

No segundo teste realizado, foi selecionada a caixa de verificação *Always Open* e premido o botão *Unlock Door*. Nestas condições, a mensagem enviada é `"Unlock_AO"` pelo que a trama recebida na plataforma é `55 6E 6C 6F 63 6B 5F 41 4F`. Novamente é efetuada a validação desta mensagem e o resultado é o desbloqueio da fechadura no modo sempre aberta. A Figura 5.5 seguinte ilustra o referido.

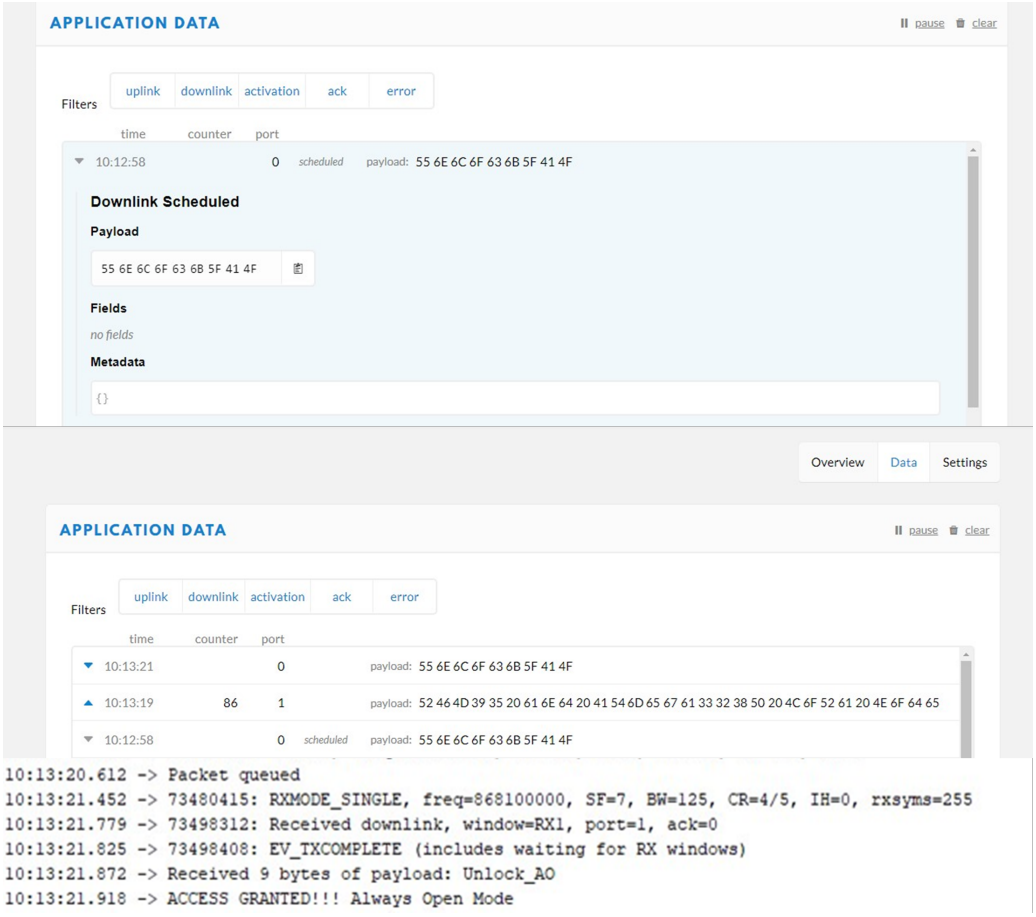


Figura 5.5: Premir do botão *Unlock Door* e da caixa de verificação *Always Open*

No próximo teste, foi definido o tempo de abertura da porta como 9 segundos, sendo que a mensagem enviada nesse caso é `Unlock_9` e os resultados são apresentados na Figura 5.6.

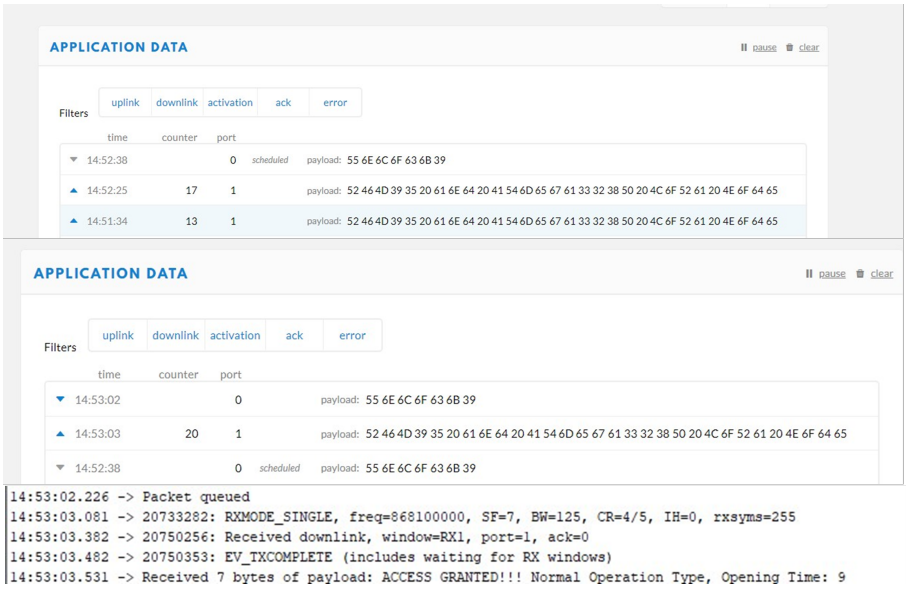


Figura 5.6: Premir do botão *Unlock Door* e seleção do tempo de abertura da porta como 9 segundos

Por último, foi efetuado um teste em que se premiu na interface o botão de *Reset Lock* e se verificou o resultado no monitor série do Arduino IDE, este procedimento está documentado na Figura 5.7.

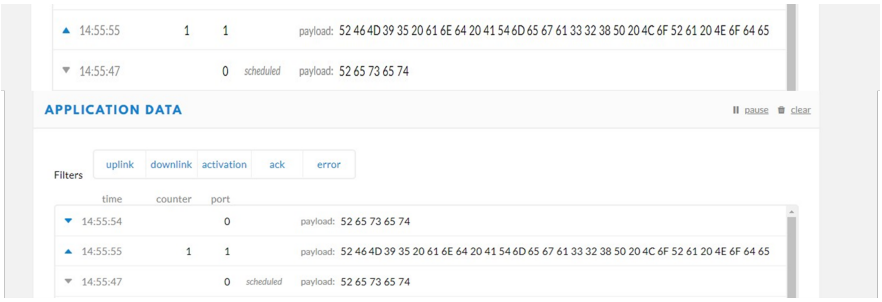


Figura 5.7: Premir do botão *Reset Lock* e verificação dos consequentes eventos

O objetivo da inclusão desta opção de *Reset* é corrigir situações em que a conexão do dispositivo LoRa ao Gateway, não se efetue em algum momento. Assim, o módulo é reiniciado em conjunto com o microcontrolador e o processo de junção do dispositivo à rede inicia-se uma vez mais.

Deste modo, evita-se que o sistema fique indeterminadamente a tentar estabelecer ligação em rede. Sendo que este cenário é obviamente pouco provável, ainda que fique salvaguardada essa possibilidade. Apesar desta ser uma solução, no futuro, poderia-se implementar um botão do reset na PCB, para facilitar o processo.

Posto isto, o sistema encontra-se completo, visto que para além do hardware que em conjunto com o *firmware* cumpre os requisitos do mesmo foi acrescentada uma interface que permite que qualquer utilizador que possua uma fechadura onde foi integrado o circuito produzido, consiga efetivamente proceder à abertura remota da mesma, tendo à sua disposição a possibilidade de execução de todas as funções referidas.

Passando então à descrição mais detalhada da interface desenvolvida, esta é composta por uma janela principal, onde o utilizador poderá atuar sobre o sistema da forma que pretender, dentro das funções desenvolvidas. A referida janela encontra-se representada na Figura 5.8.

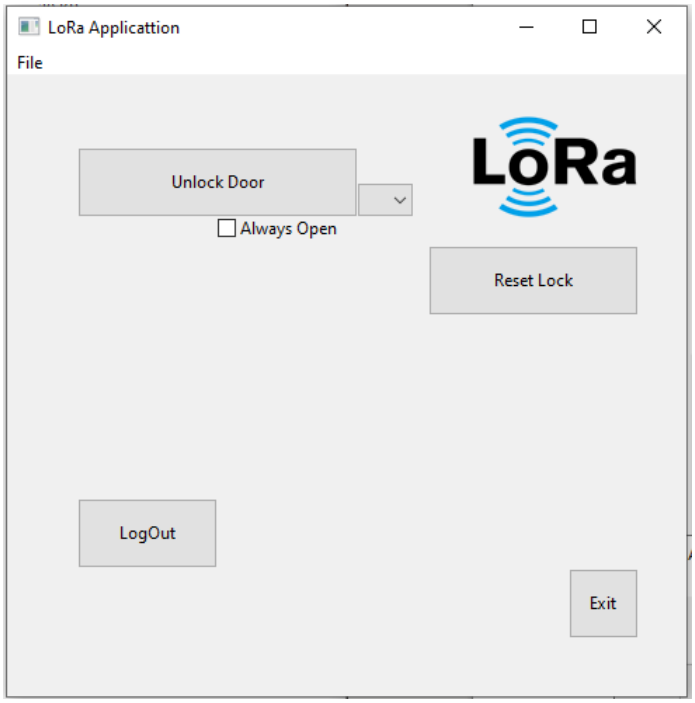


Figura 5.8: Janela Principal – *Main Frame* – da interface com o utilizador

Sendo que, para além desta o utilizador tem ao seu dispor mais duas janelas: a de *Login* e a de *Sign Up*. A primeira é aquela como a qual se deparará sempre que iniciar a aplicação, nesta fase, estão previstos dois cenários:

- o utilizador já possui uma conta com as suas credenciais e sendo assim, apenas tem de inserir as mesmas na janela de *login* para se autenticar e seguir para o *Main Frame*. De referir, ainda que o *login* é feito com *Email* ou *Username* e *Password*;
- por outro lado caso esta primeira situação não se verifique, é colocada ao dispor do usuário a possibilidade de criar uma conta, onde este terá de introduzir *email, Username e Password*.

As duas janelas descritas acima estão representadas na Figura 5.9.

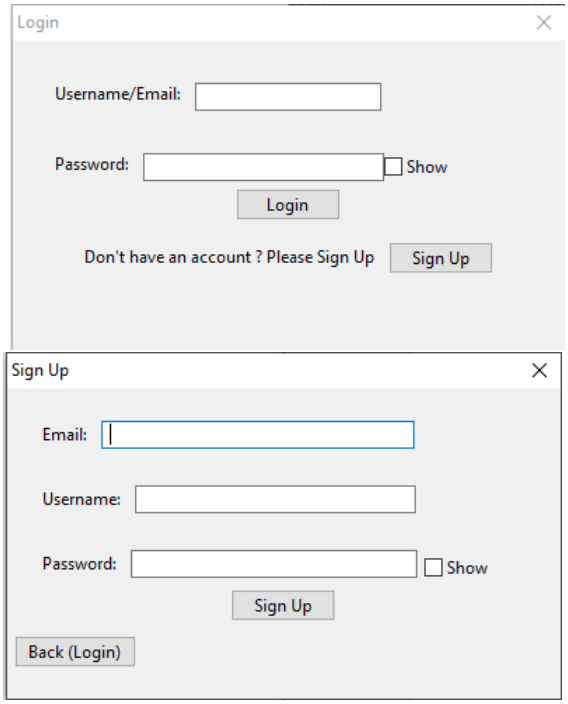


Figura 5.9: Janela de *Login* e de *SignUp* da interface com o utilizador

De notar que, no caso do utilizador introduzir credenciais inválidas é exibida uma janela com uma mensagem de erro, sendo que isto é válido na fase de *Login* e na fase de *SignUp*.

Ora, ainda que isto aconteça nas duas fases, é importante perceber que, no caso do *Login* acontece devido ao facto das credenciais introduzidas num dado momento pelo utilizador não se encontrarem registadas na base de dados, no caso da fase de *SignUp* estes erros dão-se sempre que sejam introduzidos dados (*Username* ou *Email*) que já tenham sido registados na base de dados por outro utilizador ou ocorra a tentativa de criar uma conta com um *Email* inválido.

O aspeto das referidas mensagens de erro, apresenta-se na Figura 5.10.

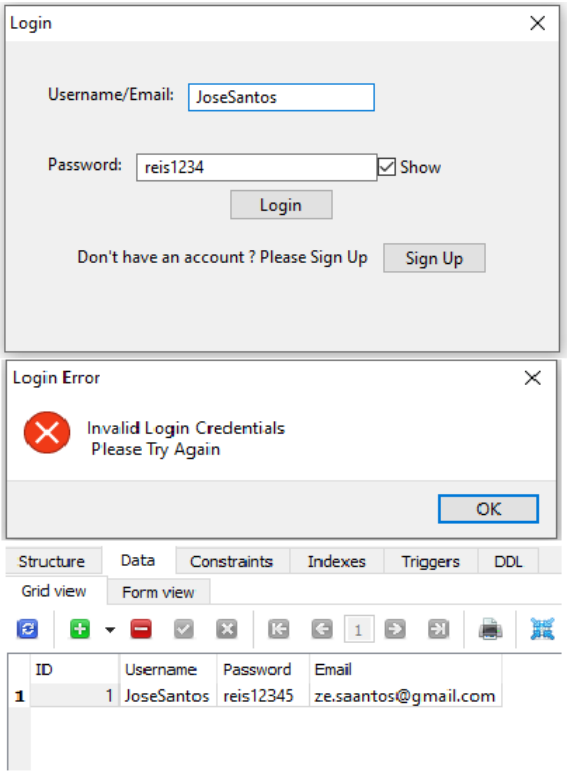


Figura 5.10: Situação em que o utilizador introduz credenciais inválidas na janela de *login*

Como se pode ver na figura acima, o utilizador insere um *Username* válido, contudo a *Password* introduzida não corresponde à que está associada ao *User JoseSantos*, pelo que, o sistema apresenta a respetiva mensagem de erro.De realçar que no momento a que os *PrintScreens* se reportam a base de dados apenas possuía um elemento.

A próxima situação configura um erro no processo de criação de uma conta – *Sign Up* – devido ao facto do utilizador inserir um *Username* que já existe na base de dados.Ora, o referido está ilustrado na Figura 5.11.

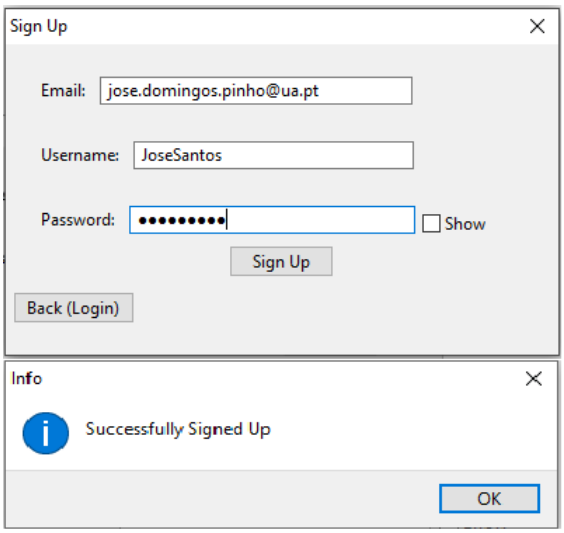


Figura 5.11: Situação em que o utilizador introduz um Username já existente na janela de *Sign Up*

Finalmente, segue-se a descrição de uma ocorrência em que um usuário no processo de registo de conta, tenta inserir um *Email* inválido, isto é, um conjunto de caracteres que não se coaduna com a estrutura típica de um endereço de correio eletrónico. Esta verificação é feita no código desenvolvido para a interface e sempre que se verificar é exibida a mensagem de erro, alertando o utilizador para o facto.Atentem, então na seguinte Figura 5.12.

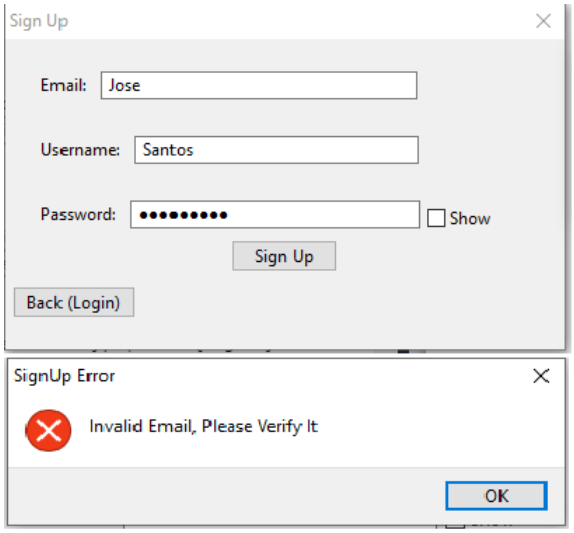


Figura 5.12: Situação em que o utilizador introduz um Email inválido na janela de *Sign Up*

Por outro lado, sempre que a tentativa de registo do utilizador for bem sucedida, ou seja, os dados sejam gravados na base dados sem ocorrência de erros, será também exibida uma mensagem em conformidade. A Figura 5.13, ilustra a exibição da referida mensagem.

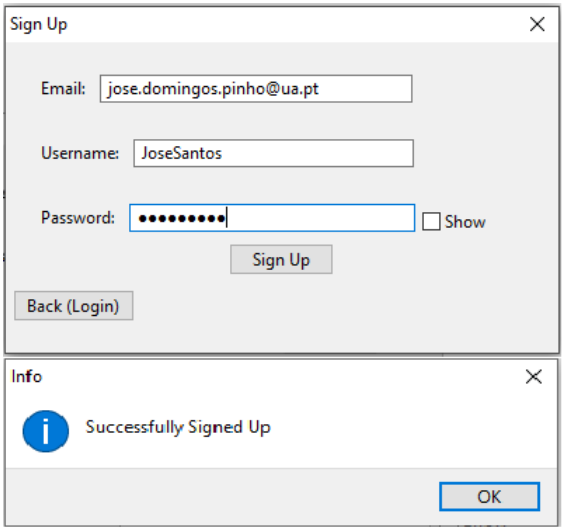


Figura 5.13: Situação em que o utilizador introduz credenciais válidas conseguindo criar a conta com sucesso

Desta forma, o utilizador faz o *Login* com as suas credenciais, sendo que caso, ainda não possua uma conta, pode criar a mesma na janela *Sign Up*, depois de criar a conta, pode premir o botão *Back (Login)* para voltar à janela de *Login* e ter a possibilidade de introduzir as suas credenciais para conseguir utilizar a interface.

Como já apresentado, no *Main Frame* estão disponíveis os botões *Unlock Door*, *Reset Lock*, *LogOut* e *Exit*, bem com a caixa de verificação *Always Open* e a caixa de seleção do tempo de abertura.

Está prevista ainda outra situação em que o programa exibe uma mensagem de erro, quando o utilizador seleciona simultaneamente a caixa de verificação *Always Open* e o tempo de abertura para fechadura. Estas duas ações são, inerentemente, incompativeis e por isso é exibida a referida mensagem e nenhum comando é enviado para a fechadura. O utilizador neste caso, terá de decidir qual das ações pretende.

A situação referida está representada na Figura 5.14.

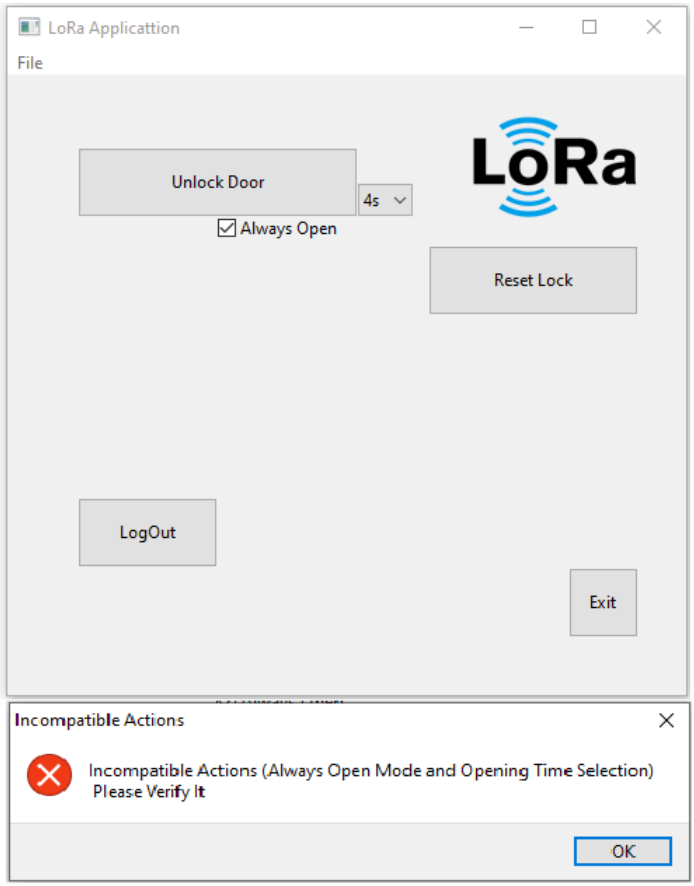


Figura 5.14: Alerta/Mensagem de erro: Ações Incompatíveis

O utilizador poderá ainda aceder a um ficheiro de texto onde estão descritas algumas instruções em relação à utilização da referida interface.

A interface que foi descrita até ao momento foi desenvolvida em python e é baseada na biblioteca `wxpython`. Para simular a implementação de uma base de dados para guardar os dados de *login* foi utilizado o programa `SQLiteStudio` – como já apresentado acima – o qual é ilustrado na Figura 5.15.

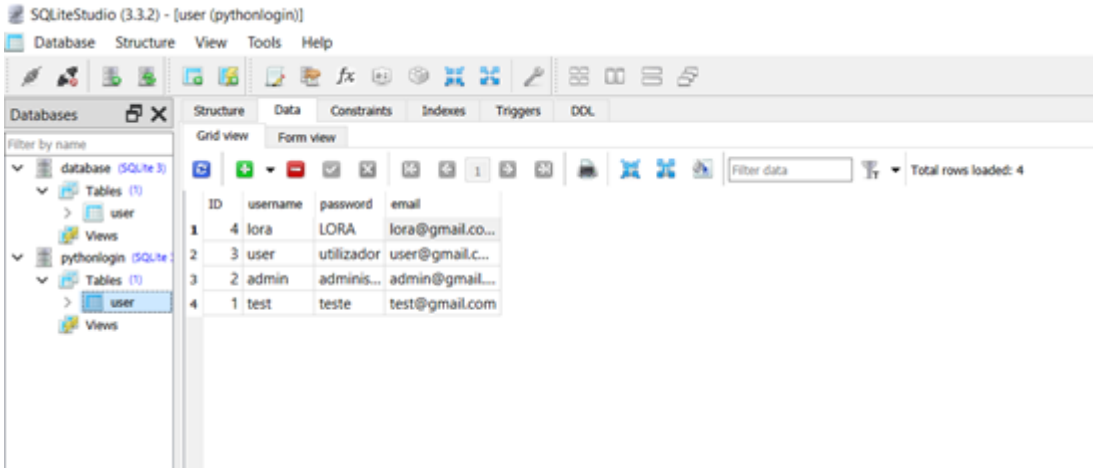


Figura 5.15: Programa para criação da base de dados de suporte ao *login*

A interface desenvolvida serve de prova de conceito, no sentido de facilitar ao utilizador a integração entre o hardware desenvolvido e o servidor LoRa alojado na plataforma TTN. Está, assim, provado que é possível enviar comandos/instruções para o Gateway LoRa através de integração HTTP, que por sua vez as envia para o módulo RFM95 e a partir daí interpretar as mensagens recebidas e agir da forma correspondente, através do firmware inserido no microcontrolador.

CAPÍTULO 6

Conclusão

Neste capítulo serão apresentadas as principais conclusões do trabalho desenvolvido, bem como algumas questões que poderiam ser trabalhadas, para melhoramento da solução obtida, numa perspectiva de futuro.

Como demonstrado ao longo deste documento, o principal objetivo do projeto realizado durante este estágio curricular, foi aplicar a tecnologia de comunicação LoRa à área de atividade da entidade acolhedora, ou seja, ao controlo de acessos.

Este objetivo foi cumprido, visto que foi desenvolvida uma placa de circuito impresso, capaz de alcançar este desiderato.

Relembrando o que foi referido, no Capítulo 3, uma das limitações da tecnologia LoRa prende-se com a sua norma de utilização, isto é, a imposição de um limite para o Duty-Cycle de 1%. Ora, também nesse capítulo é apresentada uma das formas de contornar essa limitação, que se relaciona com o número de canais que se coloca à disposição de cada End-Device. Esta questão, juntamente com o facto das mensagens trocadas serem curtas, isto é, possuírem menos de 30 *bytes*, faz com que a limitação referida, não tivesse repercussões evidentes na performance do sistema.

Como referido, no Capítulo 4, numa primeira fase foi desenvolvido um protótipo em placa branca, sendo que este foi devidamente testado, antes de se passar para o desenvolvimento da PCB. Este procedimento, revelou-se importante para evitar possíveis problemas, que poderiam surgir apenas, posteriormente, ou seja, na fase de testes à placa de circuito impresso, em si.

Estes problemas, surgiram, efetivamente, na fase de testes ao protótipo, principalmente devido a erros no dimensionamento do circuito de acionamento do motor, a `ponte-h`. Assim, devido ao incorreto dimensionamento das resistências, os transístores estavam a consumir demasiada corrente, e isso fazia com que a tensão de alimentação do circuito variasse de forma altamente indesejada.

Por outro lado, a escolha do microcontrolador ATmega328p acabou por não se revelar muito certeira, visto que este tem uma particularidade que, juntamente com a utilização do módulo RFM95, fazem com que a implementação do *firmware* na placa não seja um processo tão expedito quanto desejável. Principalmente, se pensarmos numa hipotética produção da mesma, em grandes quantidades.

Ora, concretizando, os chips ATmega328p não vem com *bootloader* instalado – detalhado no Capítulo 4 – este tem de ser instalado depois da produção da PCB, através da (única) interface SPI que o microcontrolador possui, sendo que esta também é utilizada pelo módulo RFM95. Que por ironia do destino, também é a única interface de comunicação que este possui. Resumindo, tem de ser incluídos na placa os *solder jumpers*, por forma a que a ligação entre o RFM95 e o ATmega328p não esteja implementada, por omissão.

Para contrariar este problema, poderia-se incluir no futuro, jumpers amovíveis, em vez de jumpers de solda como foram usados. Desta forma, sempre que fosse necessário instalar o *bootloader* apenas se alterava a posição dos jumpers para interromper a ligação entre o ATMEGA e o RFM95. Assim após a instalação do *bootloader* é que poderia-se efetivar esta ligação, de forma mais prática.

Ainda dentro das perspectivas de melhoria da solução obtida, regista-se a possibilidade de incluir um sistema capaz de detetar uma situação em que, a tensão da bateria, descesse de um determinado nível e enviasse essa informação através da rede LoRa. Desta forma o utilizador, poderia ter acesso a essa informação sem ter que se deslocar ao local onde o produto esteja instalado. De referir, ainda que isto pudesse, efetivamente, representar uma melhoria da solução atual, não foi considerado uma prioridade, visto que, os produtos aos quais esta placa será acrescentada, já possuem um mecanismo que deteta baterias fracas e sinaliza essa situação ao utilizador. Ainda, que, se volte a frisar que este mecanismo é apenas "presencial".

Outra situação que poderia ser acautelada futuramente, seria o envio de notificações para o utilizador de cada vez que uma fechadura fosse operada. Assim, pensando num cenário de utilização com aplicação móvel ou similar, o utilizador ao enviar o comando para abertura de uma porta, poderá, instantaneamente, saber se este processo se desenrolou com sucesso, ou se, por outro lado, aconteceu algum problema. Para além disso, com esta informação poderia ser possível aceder a um histórico de utilizações, e até o associar de cada acesso a um utilizador/pessoa, o que pode ser útil, dentro do contexto do controlo de acessos.

Estas questões, não tendo sido referidas, como requisito por parte da entidade acolhedora, acabariam por aportar ao sistema funcionalidades extra, que fariam com que a solução ficasse mais completa.

Por último, referir que, conforme apresentado no capitulo 4, a distância de comunicação máxima conseguida, foi de aproximadamente, 417 metros. Considerando toda a teoria abordada no Capítulo 3, esta distância deveria ser maior. Concretizando, a maior parte dos artigos estudados referem, muitas vezes com suporte em termos de testes realizados, que o alcance da comunicação LoRa, em zonas rurais, deve ser da ordem das (poucas) dezenas de quilómetros. Ora, como é óbvio, o alcance referido anteriormente encontra-se longe destes valores, pelo que, esta poderá ser também uma questão a estudar, possivelmente, no futuro, para se perceber a razão destes valores não serem mais próximos.

Sendo assim, em termos de mais-valias entregues à empresa, no final deste estágio, registam-se um protótipo funcional em placa branca, devidamente testado, quer em termos de implementação do protocolo LoRaWAN, como de controlo da atuação do motor DC, responsável pelo desbloqueio da fechadura. Para além disto, e ainda dentro da componente de hardware, foram entregues à empresa protótipos em PCB, quer do circuito LoRa, quer da antena. Estes foram sujeitos a uma série de testes que visavam perceber se estariam funcionais em termos do referido acima. Ora, apesar dos erros referidos acima e após correção dos mesmos, verificou-se que estes circuitos cumprem todos os requisitos do sistema, em termos de hardware.

No que diz respeito ao *firmware*, este foi devidamente testado, quer na implementação da comunicação LoRa, ou seja, o envio/receçáo de mensagens para o Gateway, quer, posteriormente, na integração com a interface do utilizador, onde foram desenvolvidas as funções que estão à disposição do utilizador. Concretizando, foi necessário garantir que o *firmware* estaria preparado para receber comandos vindos do Gateway e agir em conformidade. Ora, isso foi conseguido e constitui naturalmente, uma peça importante no resultado final do trabalho desenvolvido.

Finalmente, como já abordado ao longo deste relatório, para completar a solução desenvolvida, criou-se uma interface do utilizador, para demonstrar que é possível que um utilizador, controle as suas fechaduras através da comunicação LoRa, sendo que este apenas terá de instalar uma aplicação num computador. Mais uma vez, a ideia será, a partir desta demonstração, que a empresa desenvolva uma aplicação mais robusta e que possa ser comercializada, em conjunto com a solução desenvolvida.

De uma forma geral, este estágio curricular foi enriquecedor para a minha formação, visto que muitos dos conhecimentos adquiridos ao longo do curso puderam ser postos em prática, no sentido de criar um produto, que pudesse, de alguma forma, colmatar necessidades existentes no mercado, mais concretamente, na área do controlo de acessos.

Referências

[1] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen e J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage", *International Journal of Distributed Sensor Networks*, vol. 13, n.º 3, p. 1 550 147 717 699 412, 2017.

[2] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic e A. L. Aquino, "LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities", em *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, IEEE, 2017, pp. 1–6.

[3] R. S. Sinha, Y. Wei e S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT", *Ict Express*, vol. 3, n.º 1, pp. 14–21, 2017.

[4] P. San Cheong, J. Bergs, C. Hawinkel e J. Famaey, "Comparison of LoRaWAN classes and their power consumption", em *2017 IEEE symposium on communications and vehicular technology (SCVT)*, IEEE, 2017, pp. 1–6.

[5] Z. Ali, S. Henna, A. Akhunzada, M. Raza e S. W. Kim, "Performance evaluation of LoRaWAN for green internet of things", *IEEE Access*, vol. 7, pp. 164 102–164 112, 2019.

[6] J. Haxhibeqiri, E. De Poorter, I. Moerman e J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application", *Sensors*, vol. 18, n.º 11, p. 3995, 2018.

[7] Q. Zhou, K. Zheng, L. Hou, J. Xing e R. Xu, "Design and implementation of open LoRa for IoT", *IEEE Access*, vol. 7, pp. 100 649–100 657, 2019.

[8] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia e N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks", em *2016 IEEE SENSORS*, IEEE, 2016, pp. 1–3.

[9] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad e M. Ismail, "Energy-efficient wireless sensor networks for precision agriculture: A review", *Sensors*, vol. 17, n.º 8, p. 1781, 2017.

[10] P. A. Catherwood, D. Steele, M. Little, S. Mccomb e J. McLaughlin, "A community-based IoT personalized wireless healthcare solution trial", *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–13, 2018.

[11] V. Sharma, I. You, G. Pau, M. Collotta, J. D. Lim e J. N. Kim, "LoRaWAN-based energy-efficient surveillance by drones for intelligent transportation systems", *Energies*, vol. 11, n.º 3, p. 573, 2018.

[12] N. Podevijn, D. Plets, J. Trogh, L. Martens, P. Suanet, K. Hendrikse e W. Joseph, "TDoA-based outdoor positioning with tracking algorithm in a public LoRa network", *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[13] G. Pasolini, C. Buratti, L. Feltrin, F. Zabini, C. De Castro, R. Verdone e O. Andrisano, "Smart city pilot projects using LoRa and IEEE802. 15.4 technologies", *Sensors*, vol. 18, n.º 4, p. 1118, 2018.

[14] S. Gil-Lebrero, F. J. Quiles-Latorre, M. Ortiz-López, V. Sánchez-Ruiz, V. Gámiz-López e J. J. Luna-Rodríguez, "Honey bee colonies remote monitoring system", *Sensors*, vol. 17, n.º 1, p. 55, 2017.

[15] I. Tomić, L. Bhatia, M. J. Breza e J. A. McCann, "The limits of LoRaWAN in event-triggered wireless networked control systems", em *2018 UKACC 12th International Conference on Control (CONTROL)*, IEEE, 2018, pp. 101–106.

[16] X. Yang, E. Karampatzakis, C. Doerr e F. Kuipers, "Security vulnerabilities in LoRaWAN", em *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, IEEE, 2018, pp. 129–140.

[17] "8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash", Atmega328p, 7810D–AVR–01/15, Atmel, 2015.

[18] "RFM95/96/97/98(W) - Low Power Long Range Transceiver Module", RFM95, V1.0, HopeRF Eletronics, 2015.

[19] "TLV226x,TLV226xA,Advanced LinCMOS Rail-to-Rail Operational Amplifiers", TLV226, SLOS186C, Texas Instruments, 2006.

[20] "LM317 3-Terminal Adjustable Regulator", LM317, SLVS044Y, Texas Instruments, 2020.