

PM

**Recolha de Dados
numa Rede de Sensores
em Ambiente Urbano**

PROJETO DE MESTRADO

João Nuno Freitas Pereira

MESTRADO EM ENGENHARIA ELETROTÉCNICA - TELECOMUNICAÇÕES



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

fevereiro | 2020

Recolha de Dados numa Rede de Sensores em Ambiente Urbano

PROJETO DE MESTRADO

João Nuno Freitas Pereira

MESTRADO EM ENGENHARIA ELETROTÉCNICA - TELECOMUNICAÇÕES

ORIENTAÇÃO

Joaquim Amândio Rodrigues Azevedo



Faculdade de Ciências Exatas e da Engenharia

Recolha de dados numa rede de sensores em ambiente urbano

Dissertação submetida à Universidade da Madeira para obtenção do Grau de Mestre em Engenharia Eletrotécnica – Telecomunicações

Orientador: Professor Doutor Joaquim Amândio Rodrigues Azevedo

João Nuno Freitas Pereira

2020

“A persistência é o caminho do êxito.”

Charles Chaplin, ator-compositor. (1889-1977)

Resumo

Neste projeto é desenvolvida uma rede de monitorização urbana de baixo custo para a qualidade do ar. Deste modo, é necessário o estudo de diferentes métodos de recolha dos dados para uma rede de sensores sem fios (RSSF). Esta rede de monitorização é desenvolvida com o intuito de avaliar a qualidade do ar na cidade do Funchal, nomeadamente a afluência do tráfego urbano. Inicialmente é realizada uma breve pesquisa incidindo sobre o tema da qualidade do ar. Estudam-se diferentes tecnologias rádio no âmbito da transmissão e recolha de dados. Por fim, apresentam-se alguns exemplos de projetos de monitorização urbana ambiental.

Para o desenvolvimento da rede de monitorização urbana são construídos alguns protótipos. Estes são um nó de calibração, um nó sensor local, um gateway de aquisição de dados e um sistema para monitorização remota, os quais destinam-se à monitorização de monóxido de carbono (CO). As tecnologias de transmissão de dados utilizadas são Zigbee e LoRa (Long Range).

O nó de calibração serve-se de um sensor cujas características de resolução e gama dinâmica são apropriadas na monitorização de concentrações poluentes mais baixas. Os restantes protótipos utilizam um sensor de baixo custo. O nó local e o sistema de monitorização remoto são estações instaladas em pontos fixos. O gateway de aquisição de dados destina-se à recolha e tratamento de dados monitorizados por nós móveis. O sistema de monitorização remota serve-se das tecnologias LoRa e Zigbee. A rede de monitorização faz uso da rede de sensores existente na Universidade, a qual maioritariamente utiliza a tecnologia Zigbee, uma aplicação Java que recolhe e realiza o tratamento de dados e uma base de dados MySQL. Os utilizadores podem consultar a informação de interesse numa página Web.

Palavras-chave: Recolha dos dados, rede de sensores sem fios (RSSF), rede de monitorização urbana de baixo custo, qualidade do ar

Abstract

In this project, a low-cost urban monitoring network for air quality is developed. So, it is necessary to study different methods of data collection for a wireless sensor network (WSN). Such monitoring network is developed in order to assess the air quality in the city of Funchal, namely the influx of urban traffic. Initially a brief survey is carried out focusing on the topic of air quality. Different radio technologies are studied regarding data transmission and collection. Then some examples of urban monitoring projects are presented.

For the development of the urban monitoring network, some prototypes are built. These are a calibration node, a local sensor node, a data acquisition gateway and a system for remote monitoring. These are used to monitor carbon monoxide (CO). The data transmission technologies used are Zigbee and LoRa (Long Range).

The calibration node has a sensor which resolution and dynamic range features are appropriate for monitoring lower pollutant concentrations. The remaining nodes use a low-cost sensor. The local node and the remote monitoring system are stations installed at fixed points. The data acquisition gateway is for the collection and processing of data monitored by mobile nodes. The remote monitoring system uses LoRa and Zigbee technologies. The monitoring network makes use of the existing sensor network at the University. Which mostly uses Zigbee technology, a Java application that collects and performs data processing and a MySQL database. Users can consult the information of interest on a web page.

Keywords: Data collection, Wireless Sensor Network (WSN), Low-cost urban monitoring network, air quality

Agradecimentos

Durante a realização do projeto de mestrado como seria de esperar conta-se com o apoio e incentivo de diversas pessoas, as quais contribuíram para levar a cabo a sua realização, deste modo gostaria de expressar um sincero agradecimento a todas elas.

Em primeiro lugar ao Professor Doutor Joaquim Amândio Rodrigues Azevedo venho a expressar os meus agradecimentos pelos conhecimentos transmitidos, o interesse, a disponibilidade e a compreensão durante as minhas fases menos boas assim como sugestões construtivas no sentido de melhorar o trabalho desenvolvido.

Ao meu pai, minha mãe e minha irmã pelo apoio, o incentivo e paciência necessários durante o desenrolar da minha formação académica e em particular o projeto de mestrado. Naturalmente à minha restante família um grande obrigado pelo apoio, incentivo e interesse demonstrados.

Ao Engenheiro Filipe Santos um especial obrigado pelo auxílio, disponibilidade e incentivo, os quais foram cruciais para conseguir concluir este trabalho. À Universidade da Madeira e seus docentes que tiveram a sua contribuição ao longo do meu percurso académico. Um profundo agradecimento para todos os meus amigos e colegas, estes proporcionaram apoio, incentivo e ideias úteis durante os meus momentos de maior dificuldade. De um modo geral e realço só tenho a agradecer a todas as pessoas que fizeram parte desta fase decisiva para a minha vida, a todos eles um profundo obrigado!

Índice

Conteúdo

Resumo	IV
Abstract	V
Agradecimentos	VI
Índice	VII
Índice de Figuras	XII
Índice de Tabelas	XVI
Listagem de Acrónimos	XVII
1. Introdução	1
1.1. Motivação	1
1.2. Objetivos	1
1.3. Estruturação da Dissertação	2
2. Estado da Arte	3
2.1. Plataformas de qualidade do ar de baixo custo	3
2.1.1. Sensores de gás e sensores de partículas poluentes	3
2.1.2. Variedades de sensores de gás	4
2.1.3. Índices de qualidade do ar e métodos de medição	5
2.2. Estações de monitorização	6
2.3. Redes oportunistas	7
2.4. Redes de Sensores sem Fios	8
2.5. Zigbee	9
2.5.1. Camada Física	9
2.5.2. Camada MAC	10
2.5.3. Consumos energéticos	10
2.5.4. Formato tramas API	10
2.6. Wi-Fi	11
2.7. Bluetooth	13
2.8. LoRa (<i>Long Range</i>)	13
2.8.1. Parâmetros radiofrequência	14
2.8.2. Camada MAC	15
2.8.3. Formato da trama LoRa	16
2.8.4. Alcance de transmissão	17
2.8.5. Consumos de energia e regulamentação <i>duty cycle</i>	17
2.9. Exemplos de projetos de monitorização	18
2.9.1. Projeto EuNet Air Quality	18

INDICES

2.9.2.	Projeto Urbisnet.....	20
2.9.3.	Projeto uSense	23
2.9.4.	Rede de monitorização urbana LoRaWAN.....	25
2.9.5.	Síntese – Projetos de Monitorização	27
3.	Desenvolvimento dos sistemas de recolha de dados para a rede de sensores em ambiente urbano	28
3.1.	Sistemas implementados	28
3.2.	Arquitetura de rede de monitorização urbana	29
3.3.	Módulo de calibração	32
3.3.1.	Diagrama funcional.....	32
3.3.2.	Microcontrolador	33
3.3.3.	Sensor CO Alphasense B4.....	33
3.3.4.	Ecrã LCD TFT Arduíno	35
3.3.5.	Consumos de energia.....	36
3.3.6.	Arquitetura	36
3.3.7.	Controlo embebido	37
3.4.	Estação de monitorização local.....	38
3.4.1.	Diagrama funcional.....	38
3.4.2.	Alimentação	38
3.4.3.	Sensor CO TGS 5042.....	39
3.4.4.	Circuito de condicionamento de sinal TGS 5042	40
3.4.5.	Transferência de dados	41
3.4.6.	Inserção dos dados na base de dados	43
3.4.7.	Arquitetura	45
3.4.8.	Controlo embebido	45
3.5.	Gateway de aquisição de dados	46
3.5.1.	Diagrama funcional.....	46
3.5.2.	Arquitetura	47
3.5.3.	Controlo embebido	48
3.5.4.	Inserção dos dados na base.....	51
3.6.	Estação de monitorização remota.....	52
3.6.1.	Diagrama funcional.....	52
3.6.2.	Alimentação	52
3.6.3.	Módulo RF LoRa.....	53
3.6.4.	Configurações RFM9x LoRa emissor e mensagem de dados.....	55
3.6.5.	Consumos energéticos	57
3.6.6.	Arquitetura	57

INDICES

3.6.7.	Controlo embebido	58
3.7.	Módulo gateway LoRa.....	59
3.7.1.	Diagrama funcional.....	59
3.7.2.	Consumos energéticos	60
3.7.3.	Configurações módulo RFM9x LoRa recetor.....	60
3.7.4.	Configurações módulo RF XBee Series 2	61
3.7.5.	Inserção dos dados de saída na base	62
3.7.6.	Arquitetura	63
3.7.7.	Controlo embebido	64
4.	Resultados	66
4.1.	Sensor Alphasense B4 – testes nó de calibração	66
4.2.	Testes nó sensor local	67
4.2.1.	Estudo do nível de sinal.....	67
4.2.2.	Estudo da interferência causada pela radiação solar	69
4.2.3.	Estudo da influência da temperatura nas medições de CO	70
4.2.4.	Medições dos parâmetros na Web	72
4.3.	Gateway de aquisição de dados para estações móveis	75
4.4.	Estudo e testes – sistema remoto LoRa-Zigbee	77
4.4.1.	Estudo da eficiência de transmissão	77
4.4.2.	Estudo do alcance de transmissão RFM9x LoRa	78
4.4.3.	Dados recolhidos na página Web	82
4.5.	Síntese de resultados.....	85
5.	Conclusões e trabalhos futuros.....	87
5.1.	Conclusões	87
5.2.	Trabalhos futuros	89
6.	Referências Bibliográficas.....	90
7.	Anexos	95
	Anexo A - Wi-Fi – estudo como tecnologia de recolha de dados em redes de sensores	95
	Modelo de operação – nó com tecnologia de comunicação sem fios Wi-Fi	95
	Consumos de corrente e taxa de transmissão	96
	Tipos de pacotes de dados	96
	Formato das tramas	96
	Anexo B - Bluetooth – estudo como tecnologia de recolha de dados em redes de sensores	98
	Estrutura de rede - Bluetooth clássico	98
	Piconets em redes de sensores	98

INDICES

Estados de operação	99
Consulta	99
Paginação	100
Bluetooth clássico – formato da trama	100
<i>Bluetooth Low Energy</i> (BLE)	101
Formato trama – Bluetooth Low Energy.....	103
Bluetooth 5.0	103
Anexo C - Formato mensagem no protocolo LoRaWAN	105
Anexo D - Hardware nó sensor Urbisnet	106
Anexo E - Hardware nós de monitorização – projeto uSense.....	107
Anexo F - Microcontrolador Arduino Pro-Mini - pinout.....	108
Anexo G - XBee Series 2.....	109
Anexo H - Módulo de calibração	111
Esquemático	111
Hardware desenvolvido.....	111
Código - controlo embebido	112
Fotos – módulo de calibração	116
Anexo I – Nó sensor local	118
Circuito de condicionamento de sinal – sensor de CO Figaro TGS 5042 – design layout PCB e fotos	118
Estrutura anti-rradiação sensor de CO Figaro TGS 5042	119
Esquemático	119
Hardware nó local	120
Código - controlo embebido	121
Teste estudo da influência da temperatura nas medições de CO.....	126
Variabilidade dados de CO	126
Anexo J – Gateway de aquisição de dados	128
Esquemático	128
Código - controlo embebido	128
Dados temporais e geográficos para os pontos geográficos apresentados no mapa da seção 4.3.....	135
Gateway de aquisição de dados – Fotos	135
Anexo K – Terminal remoto	136
Esquemático e Hardware	136
Código - controlo embebido	138
Anexo L – Gateway LoRa/Zigbee	144
Esquemático e hardware.....	144
Código - controlo embebido	146

INDICES

Anexo M – Dados extras – sistema de monitorização remoto	152
--	-----

Índice de Figuras

Figura 2.1 Sensores de poluição: a) Sensor de gás Alphasense [2] b) Sensor de partículas poluentes PM10 ou PM2.5 [3].	3
Figura 2.2 Diagramas esquemáticos sensores estado sólido [5]: a) Sensor bead-type. b) Sensor chip type.	4
Figura 2.3 a) Arquitetura de um nó sensor [14]. b) Modelo genérico de uma rede de sensores [12].	8
Figura 2.4 Topologia da rede Zigbee [15].	9
Figura 2.5 Sequência de ações protocolo CSMA/CA [17].	10
Figura 2.6 Estrutura genérica para uma trama Zigbee API [19].	11
Figura 2.7 Modelo de uma rede Wi-Fi [20].	12
Figura 2.8 a) Arquitetura de rede LoRa [28]. b) Sequência de operações para as classes LoRaWAN [29].	15
Figura 2.9 formato da trama LoRa ao nível da camada física [30].	16
Figura 2.10 Ciclo de atividade de um nó emissor [29].	18
Figura 2.11 Laboratório móvel da qualidade do ar, realização da experiência na cidade de Aveiro [33].	19
Figura 2.12 Diagrama de blocos da estação móvel Urbisnet [34].	21
Figura 2.13 Diagrama da rede Urbisnet [35].	21
Figura 2.14 Exemplo de comunicações Wi-Fi entre a estação de monitorização Urbisnet e o computador. Software Urbisnet no computador [34].	22
Figura 2.15 Arquitetura do sistema uSense [36].	23
Figura 2.16 Visualização na página Web uSense através do Google Maps da localização geográfica de três nós numa região em Itália e respetivos índices de qualidade do ar [36].	24
Figura 2.17 App android que utiliza uSense [36]: a) Página Inicial. b) Localização geográfica dos sensores mais próximos.	25
Figura 2.18 Modelo da rede de sensores urbana desenvolvida em Delft [37].	26
Figura 3.1 Arquitetura da rede de monitorização urbana.	29
Figura 3.2 Arquitetura de rede do sistema da rede de sensores [38].	31
Figura 3.3 Diagrama funcional nó de calibração.	32
Figura 3.4 a) Microcontrolador Arduíno Pro-Mini [41]. b) Utilização do programador conversor série-USB (FTDI) [41].	33
Figura 3.5 a) Sensor CO Alphasense B4 [2]. b) Respetivo diagrama esquemático [2].	34
Figura 3.6 a) Circuito de condicionamento Individual Sensor Board (ISB) [42]. b) Pinagem de conexão à placa [42].	35
Figura 3.7 Ecrã LCD TFT Arduíno [44].	36
Figura 3.8 Arquitetura interna nó de calibração.	37
Figura 3.9 Fluxograma que caracteriza controlo embebido do nó de calibração.	37
Figura 3.10 Diagrama funcional para o nó local.	38
Figura 3.11 a) Sensor CO TGS 5042 [51]. b) Impressão sensibilidade e número de referência [51].	39
Figura 3.12 Circuito de condicionamento para o sensor CO TGS 5042 [52].	40
Figura 3.13 Opções de antenas utilizadas nos módulos RF XBee [58].	41

Figura 3.14 Comunicação UART entre módulo XBee e microcontrolador. Dispositivos com interface UART integrada. À esquerda caso emissor, à direita caso recetor [57].	42
Figura 3.15 Script do código fonte “msgCare_base.java” utilizado para o nó local.	44
Figura 3.16 Segmento de código que ilustra o processo de interpretação da mensagem Zigbee no código fonte “Msg_parser.java”	44
Figura 3.17 Arquitetura interna que caracteriza o nó local.	45
Figura 3.18 Fluxograma que caracteriza o controlo embebido do nó sensor local.	46
Figura 3.19 Diagrama funcional módulo gateway de aquisição de dados.	47
Figura 3.20 Módulo leitor de cartão MicroSD LC Technology [62].	47
Figura 3.21 Arquitetura interna do nó gateway de aquisição de dados. ...	48
Figura 3.22 Fluxograma código principal.	49
Figura 3.23 Algoritmo de decisão de tratamento de dados com ou sem coordenadas geográficas.	49
Figura 3.24 Algoritmos para o tratamento de dados considerando a tomada de decisão.	50
Figura 3.25 Algoritmo para o tratamento de dados de CO.	50
Figura 3.26 Segmento de código que ilustra o processo de interpretação da mensagem Zigbee no código fonte “Msg_parser.java”	51
Figura 3.27 Diagrama funcional nó remoto de monitorização.	52
Figura 3.28 Painel fotovoltaico 2 W e 6 V [63].	52
Figura 3.29 Controlador de carga MCP73871 Adafruit [64].	53
Figura 3.30 Módulo Rádio RFM9x LoRa [65].	53
Figura 3.31 Módulos RF LoRa [65]: a) RFM69HCW. b) RFM9x LoRa [65].	54
Figura 3.32 Configuração da potência de transmissão (dBm) e frequência (MHz) para o RFM9x LoRa do nó remoto [67].	56
Figura 3.33 Script a ilustrar a configuração do endereçamento RHReliableDatagram no RFM9x [67].	56
Figura 3.34 Configuração do modo de transmissão de longo alcance do RFM9x LoRa [67].	56
Figura 3.35 Constituição do payload LoRa – mensagem de dados do terminal remoto.	57
Figura 3.36 Arquitetura interna do nó remoto.	57
Figura 3.37 Fluxograma descrevendo o controlo embebido realizado para o nó remoto.	58
Figura 3.38 Diagrama funcional – nó gateway LoRa/Zigbee.	59
Figura 3.39 configuração do endereçamento cliente-servidor no programa [67].	61
Figura 3.40 Script do código fonte “msgCare_base.java”	63
Figura 3.41 Arquitetura interna nó gateway.	64
Figura 3.42 Fluxograma que descreve o controlo embebido do nó gateway.	64
Figura 4.1 Comparação sensor Alphasense com os sensores Fígaro TGS 5042.	66
Figura 4.2 Teste urbano com o sensor de CO Alphasense B4.	67
Figura 4.3 Planta do piso -2 da Universidade.	68
Figura 4.4 Estudo do nível médio do sinal (dBm) em função da distância.	68

INDICES

Figura 4.5 Estudo da influência da radiação solar – variação da tensão de saída do circuito de condicionamento de sinal do sensor Figaro TGS 5042.	69
Figura 4.6 Estudo da influência da radiação solar – variação dos níveis de monóxido de carbono medidos.	69
Figura 4.7 Circuito de condicionamento de sinal para ler o sinal de entrada do sensor de temperatura NTC.	70
Figura 4.8 Medições das concentrações de CO (ppm).	71
Figura 4.9 Temperatura no interior da plataforma medida pelo sensor NTC.	72
Figura 4.10 Medições de dados de monóxido de carbono considerando a influência da radiação solar.	72
Figura 4.11 Dados de índice de radiação solar.	73
Figura 4.12 Estrutura antirradiação para o sensor Figaro.	73
Figura 4.13 Dados monitorizados para o nível da bateria.	74
Figura 4.14 Medições de CO entre as 08:00 e as 10:30 alguns dias entre janeiro e maio.	74
Figura 4.15 Medições de CO entre as 16:30 e as 19:00 alguns dias entre janeiro e maio.	75
Figura 4.16 Dados de concentrações de monóxido de carbono recolhidas pelo gateway de nós móveis a 13-06-2018.	75
Figura 4.17 Representação das amostras recolhidas pelo gateway de aquisição referentes ao gráfico da figura 4.16.	76
Figura 4.18 Sistema de comunicação utilizado no estudo da eficiência de transmissão em função do período de timeout.	77
Figura 4.19 Testes RFM9x LoRa – Configuração médio alcance 128 chips/s.	77
Figura 4.20 Testes RFM9x LoRa – Configuração longo alcance 4096 chips/s.	78
Figura 4.21 a) Raspberry Pi [71]. b) Aplicação Android ViewRanger [72].	78
Figura 4.22 Sistema de comunicação utilizado.	79
Figura 4.23 Mapa no Google Earth Pro – locais de teste de alcance com o recetor RFM9x LoRa.	80
Figura 4.24 Monitorização diária dos níveis de monóxido de carbono.	83
Figura 4.25 Monitorização das baterias.	84
Figura 4.26 Monitorização dos níveis de monóxido de carbono entre as 08:00 e as 12:00.	84
Figura 4.27 Monitorização dos níveis de monóxido de carbono entre as 16:00 e as 20:00.	85
Figura 7.1 Operação típica para um sensor com tecnologia Wi-Fi [20].	95
Figura 7.2 Formato standard para uma trama MAC no Wi-Fi [73].	96
Figura 7.3 Apresentação de uma piconet (esquerda) e scatternet (direita) [74].	98
Figura 7.4 Formato de uma trama Bluetooth [77].	100
Figura 7.5 formato do cabeçalho do pacote de dados Bluetooth clássico [77].	101
Figura 7.6 Arquitetura de rede numa aplicação de sensores baseada em BLE [23].	102
Figura 7.7 formato de um pacote Bluetooth Low Energy [78].	103

Figura 7.8 Formatos do campo PDU: pacote publicitário (em cima), pacote de dados (em baixo) [78].	103
Figura 7.9 Topologia estrela vs topologia malha [79].	104
Figura 7.10 Bluetooth 5.0 topologia em malha com cenário many-to-many [79].	105
Figura 7.11 formato da mensagem MAC LoRaWAN [27].	105
Figura 7.12 a) Módulo Wi-Fi Ethernet Bridge [34]. b) Respetivo hardware do módulo [34].	106
Figura 7.13 a) Estação Urbisnet [34]. b) Respetivo hardware [34].	106
Figura 7.14 a) Placa de circuito dos sensores de gás Libellium [36]. b) Hardware do nó Libellium Waspnote [36].	107
Figura 7.15 Microcontrolador Arduíno Pro-Mini - pinout.	108
Figura 7.16 Representação do respetivo pinout.	109
Figura 7.17 Estrutura de tramas API do XBee.	110
Figura 7.18 Esquemático módulo de calibração.	111
Figura 7.19 Módulo de calibração – hardware design layout PCB.	111
Figura 7.20 Placa-mãe módulo de calibração.	112
Figura 7.21 a) Módulo de calibração. b) Respetivo hardware.	117
Figura 7.22 Ilustração do teste laboratorial realizado com os sensores de CO.	117
Figura 7.23 Design layout PCB – Circuito de condicionamento de sinal – sensor de CO Figaro TGS 5042.	118
Figura 7.24 Placa de circuito impressa – circuito de condicionamento sensor de CO.	118
Figura 7.25 Estrutura anti-rradiação para o sensor de CO Figaro TGS 5042.	119
Figura 7.26 Esquemático nó sensor local.	119
Figura 7.27 Design layout PCB placa mãe nó sensor local.	120
Figura 7.28 Ilustração de algumas fases de montagem do nó local.	120
Figura 7.29 Nó sensor local completo.	121
Figura 7.30 Ilustração dos testes do estudo da influência da temperatura nas medições de CO.	126
Figura 7.31 Medições de CO entre as 08:00 e as 10:30 nos dias 29-01-2019 e 02-04-2019.	127
Figura 7.32 Medições de CO entre as 16:30 e as 19:00 nos dias 29-01-2019 e 02-04-2019.	127
Figura 7.33 Esquemático gateway de aquisição de dados.	128
Figura 7.34 Gateway de aquisição de dados.	135
Figura 7.35 Esquemático terminal remoto.	136
Figura 7.36 Terminal remoto.	136
Figura 7.37 Terminal remoto.	137
Figura 7.38 Esquemático gateway LoRa/Zigbee.	144
Figura 7.39 Gateway LoRa/Zigbee.	145
Figura 7.40 Gateway LoRa/Zigbee.	145
Figura 7.41 Dados de índice de radiação solar nos dias 20-09-2019 e 22-09-2019.	152

Índice de Tabelas

Tabela 2.1 Quadro contendo a classificação dos índices de qualidade do ar propostos para o ano 2019 propostos pela Agência Portuguesa do Ambiente (APA) [8].	6
Tabela 2.2 Bandas de frequências utilizadas no protocolo IEEE 802.15.4: nº de canais, técnicas de modulação e taxa máxima de transmissão [16].	9
Tabela 2.3 Síntese dos projetos de monitorização.	27
Tabela 3.1 Descrição dos protótipos a desenvolver.	30
Tabela 3.2 Acesso ao servidor de base de dados.	32
Tabela 3.3 Especificações Arduino Pro-Mini [41].	33
Tabela 3.4 Características de desempenho do sensor Alphasense [2].	34
Tabela 3.5 Sensibilidade do sensor a vários gases interferentes [2].	34
Tabela 3.6 Parâmetros desvio offset DC de cada elétrodo e sensibilidade fornecidos pelo fabricante [2].	35
Tabela 3.7 Consumos energéticos nó de calibração.	36
Tabela 3.8 Características elétricas sensor CO TGS 5042 [51].	39
Tabela 3.9 Comparação de características elétricas para os XBee série 1 e 2 [58].	42
Tabela 3.10 Configurações realizadas no módulo de comunicação XBee Series 2.	43
Tabela 3.11 Constituição da trama de emissão Zig Bee.	43
Tabela 3.12 Pinout módulo leitor de cartão MicroSD [62].	47
Tabela 3.13 Constituição da trama de receção Zigbee.	51
Tabela 3.14 Características elétricas dos módulos RF LoRa [66].	54
Tabela 3.15 Descrição dos pinos dos módulos RF LoRa [66].	55
Tabela 3.16 Configurações rádio RFM9x LoRa [67].	55
Tabela 3.17 Consumos energéticos terminal remoto.	57
Tabela 3.18 Consumos energéticos módulos de comunicação.	60
Tabela 3.19 Consumos de energia nó gateway.	60
Tabela 3.20 Configurações módulo de comunicação LoRa [67].	60
Tabela 3.21 Configurações Router XBee Series 2.	62
Tabela 3.22 Formato dos dados de saída a transmitir no formato de protocolo Zig Bee.	62
Tabela 3.23 Parâmetros da classe que implementa o acesso à base de dados em “MyQSL.java”.	62
Tabela 4.1 Parâmetros de temperatura e resistência elétrica do NTC correspondente [69].	71
Tabela 4.2 Pontos geográficos dos testes de longo alcance com o RFM9x LoRa.	81
Tabela 4.3 Estudo da geografia topográfica do terreno considerando o perfil de terreno e obstáculo com maior elevação.	82
Tabela 7.1 Descrição dos pinos dos módulos XBee/XBee-Pro™.	109
Tabela 7.2 Dados temporais e de coordenadas geográficas dos pontos ilustrados no mapa da figura 4.17.	135
Tabela 7.3 Dados dos pontos de medição com o recetor RFM9x LoRa.	153
Tabela 7.4 Dados dos pontos de medição com o recetor RFM9x LoRa.	154

Listagem de Acrónimos

ACK - Acknowledge
ACL - Asynchronous connectionless
ADC - Analog Digital Converter
AGC - Automatic Gain Control
APA - Agência Portuguesa do Ambiente
API - Application Programming Interface
AQD - Air Quality Directive
AT - Application Transparent
BLE - Bluetooth Low Energy
BPSK - Binary Phase Shift Keying
CCDR - Comissões de Coordenação e Desenvolvimento Regional
CO - Monóxido de Carbono
CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance
CSS - Chirp Spread Spectrum
CTS - Clear to Send
DCF - Distributed Coordination Function
DDD - graus decimais
DIFS - Distributed Inter Frame Space
DMM - Graus e Minutos
DMS - Graus, Minutos e Segundos
DRA - Direções Regionais do Ambiente
DSI - Display Serial Interface
DSSS - Direct Sequence Spread Spectrum
DTN - Delay Tolerants Networks
FC - Frame Control
FCS - Frame Check Sequence
FHDR - Frame Header
FHSS - Frequency Hopping Spread Spectrum
FTDI - Future Technology Devices International
GFSK - Gaussian Frequency Shift Keying
GPIO - General-Purpose Input/Output
GPRS - General Packet Radio Service
GPS - Global Positioning System
GSM - Global System for Mobile Communications
H₂S - Sulfato de hidrogênio
HTTP - Hypertext Transfer Protocol
HTTPS - Hypertext Transfer Protocol Secure
I²C Inter-Integrated Circuit

LISTAGEM DE ACRÓNIMOS

IFS - Inter Frame Space
IoT - Internet of Things
IP - Internet Protocol
IQAr - Índice de Qualidade do Ar
ISI - intersymbol interference
ISM - Industrial Scientific Medical
IST - Instituto Superior Técnico
MAC - Media Access Control Layer
MHDR - MAC Header
MSB/LSB - most/least significant byte
NDIR - Non-Dispersive Infra-Red
NO, NO_x - Nitrogénio de Óxido
NO₂ - Dióxido de Nitrogénio/Azoto
NTC - Negative Temperature Coeficient
O₃ - Ozônio
O-QPSK - Offset Quadrature Phase Shift Keying
P2P - Peer to Peer
PAN ID - PAN Identifier
PDU - Protocol Data Unit
PER - Packet Error Rate
PHY - Physical Layer
PID - Photoionization Detector
PM10/PM2.5 - partículas poluentes com diâmetros inferiores a 10/2.5 µm
ppb - parts-per billion
PPDU - PHY Protocol Data Unit
ppm - parts-per million
PWM - Pulse-Width Modulation
RF - Rádiofrequência
RSSF - Redes de Sensores sem Fios
RSSI - Received Signal Strength Indicator
RTC - Real Time Clock
RTS - Request to Send
SC - Sequence Control
SCO - Synchronous connection-oriented
SMD - Surface-Mount Device
SNR - Signal Noise Ratio
SO₂ - Dióxido de enxofre
SPI - Serial Peripheral Interface
TCP/IP - Transmission Control Protocol/Internet Protocol
TDMA - Time Division Multiple Access

LISTAGEM DE ACRÓNIMOS

THT - Through-hole Technology

UART - universal asynchronous receiver-transmitter

UDP/IP - User Datagram Protocol/Internet Protocol

UTM WGS84 - Universal Transverse Mercator

VOC - Volatile Organic Compounds

WSN - Wireless Sensor Network

1. Introdução

Neste capítulo apresenta-se a motivação e o contexto do problema em estudo para esta dissertação, os objetivos que se pretendem atingir e a organização do documento.

1.1. Motivação

Nas grandes cidades, ao longo das últimas décadas, a poluição atmosférica tem crescido. Tal deve-se ao aumento da atividade industrial e do número de veículos de combustão interna. Estes fatores levam ao aumento da queima de combustíveis fósseis e à consequente libertação de gases e partículas poluentes, o que naturalmente leva à degradação da qualidade do ar.

A poluição atmosférica tem vários impactos negativos. É o caso das elevadas concentrações de dióxido de azoto na atmosfera que tem favorecido o aparecimento de graves doenças respiratórias. A queima e a libertação de gases poluentes têm agravado o aquecimento global, o que favoreceu o aparecimento de chuvas ácidas com um grau de afetação capaz de cobrir regiões exteriores às cidades a distâncias de centenas ou milhares de km. Tem havido influências na diminuição da camada do ozono.

Todos estes fatores têm comprometido a qualidade de vida para as populações, os animais e a vegetação. Isto motivou a uma maior preocupação com a qualidade do ar. Como tal, tem havido maior necessidade de estudos científicos e uma legislação mais rigorosa no âmbito da avaliação da qualidade do ar. Com essa finalidade desenvolveram-se sistemas de monitorização da qualidade do ar.

As áreas com maior concentração de níveis de poluição são monitorizadas por determinadas entidades. Estas entidades são autoridades governamentais, institutos científicos e especialistas na área de saúde, os quais servem-se de estações de monitorização. Estas encontram-se equipadas com instrumentos de referência certificados para a medição de gases e partículas poluentes.

1.2. Objetivos

O objetivo deste projeto consiste no desenvolvimento de uma rede de monitorização urbana de baixo custo e no estudo de métodos de transmissão e recolha dos dados de interesse. A rede de monitorização é desenvolvida com o intuito de avaliar a qualidade do ar na cidade do Funchal. Inicialmente serão estudados tópicos importantes no âmbito da qualidade do ar. São estudadas diferentes tecnologias rádio no âmbito da transmissão e a recolha de dados para as redes de sensores. Posteriormente, serão desenvolvidos os nós sensores para a rede de monitorização urbana.

1.3. Estruturação da Dissertação

Este documento está estruturado em cinco capítulos. No primeiro capítulo é realizada uma introdução ao tema e faz-se uma abordagem às motivações que levaram ao trabalho desenvolvido.

No segundo capítulo são estudados sistemas de monitorização de gases poluentes. Também se estudam possíveis soluções de transmissão e recolha de dados de monitorização nas redes de sensores sem Fios (RSSF). São descritos alguns exemplos de projetos de monitorização da qualidade do ar em ambientes urbanos.

No terceiro capítulo é descrito o trabalho de projeto realizado, no qual são apresentados os módulos desenvolvidos. Estes são um nó de calibração, um nó local, um *gateway* de aquisição de dados e um sistema remoto de monitorização. Para cada nó é descrito o hardware utilizado, a arquitetura geral de cada sistema, o controlo embebido implementado assim como eventuais configurações rádio de maior relevância no trabalho desenvolvido.

O quarto capítulo destina-se a apresentar os resultados alcançados, no qual se apresentam dados referentes a estudos realizados durante a fase de desenvolvimento de cada nó. Ainda são apresentados dados de monitorização recolhidos após a conclusão do desenvolvimento de cada nó.

A finalidade do capítulo cinco resume-se a retirar conclusões acerca do trabalho desenvolvido em geral, assim como sugestões de eventuais futuros projetos.

No Anexo começa-se por apresentar aspetos adicionais do estado da arte, nomeadamente o estudo de tecnologias de transmissão e recolha de dados. Aqui também se apresentam os nós desenvolvidos neste projeto e as diversas vertentes do trabalho realizado. Tal inclui esquemáticos de cada nó, assim como o respetivo *hardware* desenvolvido, a programação *scripting* realizada. Ainda é apresentada a construção dos nós, assim como alguns aspetos adicionais da análise de resultados.

2. Estado da Arte

Neste capítulo é realizada uma breve introdução no âmbito das redes de sensores sem fios. São apresentadas possíveis soluções de transmissão e recolha de dados para algumas tecnologias de comunicações sem fios. Numa fase posterior são abordados alguns exemplos de projetos. O objeto deste estudo são os sistemas de monitorização da qualidade do ar com aplicação em cenários urbanos. Os tópicos de estudo referem-se ao *hardware* utilizado e às tecnologias de sensores. No entanto, é dada maior ênfase no âmbito das tecnologias de transmissão e recolha de dados utilizadas.

2.1. Plataformas de qualidade do ar de baixo custo

Os estados membros da União Europeia devem cumprir as regulamentações referentes à avaliação e à gestão da qualidade do ar. As normas são definidas pela *Air Quality Directive* (AQD). É esta entidade que define os métodos de referência a aplicar em sistemas de monitorização da qualidade do ar. Em regra geral, estes métodos são utilizados nas estações de monitorização fixas [1]. Os sistemas de monitorização da poluição atmosférica são adequados aos locais com elevada densidade populacional. Estes são formados por um conjunto de nós sensores. Estas são plataformas de baixo custo e dimensões reduzidas [1]. As suas características permitem aos utilizadores monitorizar as áreas na sua proximidade.

2.1.1. Sensores de gás e sensores de partículas poluentes

Nas redes de sensores é necessário considerar alguns factores. Estes são o desempenho da tecnologia de sensores utilizada, a transmissão, a recolha e a partilha dos dados monitorizados. Em sistemas de monitorização desta natureza é necessário observar a variabilidade da medição dos parâmetros e avaliar o seu grau de incerteza, uma vez que as tecnologias de sensores sofrem afetação de factores externos. Isto geralmente influencia as medições. Geralmente essas fontes de perturbação são a interferência química ou a temperatura ambiente do local de medição [1].

A AQD regulamenta que os fabricantes das tecnologias de sensores e as redes de monitorização devem cumprir os requisitos propostos pela União Europeia. No entanto, no caso de aplicações acedidas e consultadas por cidadãos há liberdade para utilizar critérios de aceitação de resultados alternativos [1]. Pode-se categorizar os sensores de poluição atmosférica em dois grupos. Estes são os sensores de gás e os sensores de partículas poluentes. A figura 2.1 apresenta alguns exemplos de sensores de poluição atmosférica.

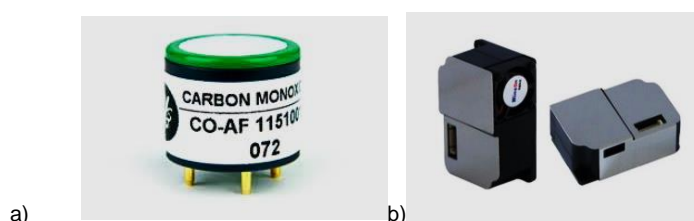


Figura 2.1 Sensores de poluição: a) Sensor de gás Alphasense [2] b) Sensor de partículas poluentes PM10 ou PM2.5 [3].

Os sensores de gás comerciais operam através de medições com base no fenómeno de interação química entre o material de deteção do mesmo e o gás poluente. Estes sensores podem ser eletroquímicos ou de óxido metal. Há também uma variante que pode atuar com base no efeito da absorção da luz a um determinado alcance. Os sensores de partículas poluentes as suas medições baseiam-se no efeito da absorção ou da dispersão da luz. Estes operam com algoritmos que comparam o sinal atenuado ao tamanho e composição da partícula [1].

Os sensores de poluição atmosférica constituem uma componente importante na plataforma de um nó sensor. Este possui a eletrónica necessária para o funcionamento eficiente do sensor. Além do sensor, o *hardware* utilizado constitui o seu circuito de condicionamento de sinal, a fonte de alimentação, os módulos de comunicação e o microcontrolador. Este último é responsável por realizar amostragens por meio de ADC (*Analog Digital Converter*). Este também realiza processamento e armazenamento dos dados monitorizados. Adicionalmente um nó pode conter ou não um recetor GPS (*Global Positioning System*) [1], este último caso depende se for um nó móvel ou fixo.

2.1.2. Variedades de sensores de gás

As variedades de sensores de gás mais usadas incluem sensores de óxido metal, sensores de estado sólido e os sensores eletroquímicos. Começando por referir os sensores em óxido metal, estes são elementos semicondutores cuja superfície é constituída por óxido metal. Em termos de funcionamento quando um determinado gás de interesse entra em contacto com a superfície do sensor vai ocorrer uma reação química. Esta reação gasosa também pode ocorrer se o gás entrar em contacto com oxigénio pré-absorvido pela superfície do sensor. Esta reação gasosa induz uma variação da resistência interna do sensor [4]. Este efeito de transdução caracteriza um sinal medido pelo sensor. A intensidade do sinal medida pelo sensor é correlacionada com as concentrações gasosas.

Para os sensores de estado sólido a sua constituição interna pode envolver um ou mais compostos em óxido metal. Há dois tipos de sensores de estado sólido. No primeiro tipo os compostos são preparados de modo a formar uma superfície de sensor esférica. Este é designado de sensor *bead-type* e encontra-se ilustrado na figura 2.2 a). Alternativamente, há os sensores em chip (*chip-type sensor*) como ilustrado na figura 2.2 b). Neste caso os compostos em óxido metal são depositados num *chip* de sílica.

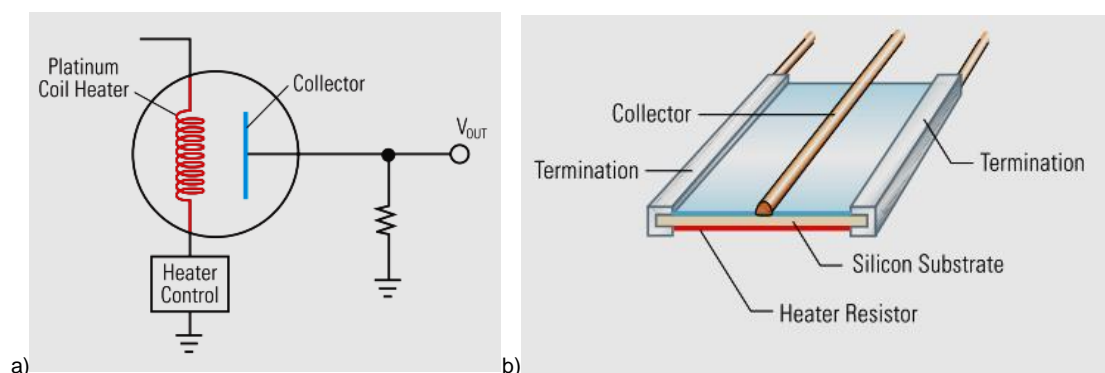


Figura 2.2 Diagramas esquemáticos sensores estado sólido [5]: a) Sensor *bead-type*. b) Sensor *chip type*.

Este tipo de sensores contém um elemento aquecedor que realiza a regulação da temperatura. Pois, os sensores de gás exibem diferentes respostas características a diferentes gamas de temperatura. A sua constituição interna pode variar. O elemento aquecedor pode ser formado por liga ou fio de platina. Opcionalmente pode ser formado por óxido metal com características resistivas ou por um depósito com uma fina camada de platina. O sensor é processado a uma elevada temperatura específica. Isto vai determinar as características do sensor.

Na presença de gás em contacto com a superfície do sensor vai ocorrer uma reação gasosa que origina transferência de eletrões. O elemento aquecedor do sensor pode ser regulado e controlado por um circuito eletrónico específico. É integrado um par de elétrodos à superfície do sensor [5]. A ideia é medir variações da condutividade induzidas pela presença de gás. Estas variações de condutividade traduzem-se num sinal medido pelo sensor. Um sinal mais intenso produzido pelo sensor traduz-se em elevadas concentrações gasosas.

Por fim, um sensor químico fornece uma resposta diretamente proporcional às concentrações de um determinado composto químico. Este tipo de sensores são elementos transdutores que fazem com que uma reação química induza um sinal. Este fenómeno de transdução ocorre através de métodos de instrumentação com circuitos eletrónicos específicos. Estes tipos de sensores podem ser elétricos, óticos ou térmicos. Estes são concebidos de modo a responder à presença de um determinado composto químico [6]. Há três tipos de sensores eletroquímicos. Estes são o potenciométrico, amperimétrico e condutométrico.

Para os sensores potenciométricos estabelece-se um equilíbrio local na superfície do sensor. O potencial dos seus elétrodos é medido. Através da medição de diferença de potencial entre os elétrodos consegue-se obter informação sobre a concentração de um determinado composto. Já com os sensores amperimétricos é utilizado um eléctrodo de trabalho (*working electrode*) e um ponto de referência. É aplicado um potencial elétrico que induz uma corrente que é medida. Esta corrente causa uma reação de um determinado composto químico. Deste modo é possível estimar as concentrações desse composto. Nos sensores condutométricos a obtenção das concentrações de um determinado composto são conseguidas medindo a condutividade para um conjunto de frequências diferentes [6].

2.1.3. Índices de qualidade do ar e métodos de medição

O índice de qualidade do ar (IQAr) é determinado diariamente. Os seus dados são disponibilizados pela Agência Portuguesa do Ambiente (APA). Estes são monitorizados em estações de qualidade do ar geridas pelas Comissões de Coordenação e Desenvolvimento Regional (CCDR) no continente. No caso das Regiões Autónomas da Madeira e Açores as entidades responsáveis são as Direções Regionais do Ambiente (DRA) [7].

Os índices de qualidade do ar são calculados através de médias aritméticas dos parâmetros poluentes. Essas médias resultam de várias horas de amostras medidas. Os parâmetros poluentes monitorizados são dióxido de azoto (NO₂), ozono (O₃), monóxido de carbono (CO), dióxido de enxofre (SO₂) e partículas poluentes PM10. Há diversas estações de qualidade do ar instaladas ao longo de Portugal Continental e das Regiões Autónomas.

Na tabela 2.1 está ilustrado o quadro contendo a classificação dos índices de qualidade do ar do ano 2019 [8].

Tabela 2.1 Quadro contendo a classificação dos índices de qualidade do ar propostos para o ano 2019 propostos pela Agência Portuguesa do Ambiente (APA) [8].

Poluente em causa / Classificação	CO (mg/m ³)		NO ₂ (µg/m ³)		O ₃ (µg/m ³)		PM ₁₀ (µg/m ³)		SO ₂ (µg/m ³)	
	Min	Máx	Min	Máx	Min	Máx	Min	Máx	Min	Máx
Mau	10,0	-----	400	-----	240	-----	100	-----	500	-----
Fraco	8,0	9,999	200	399	180	239	50	99	350	499
Médio	7,0	7,999	100	199	100	179	35	49	200	349
Bom	5,0	6,999	40	99	80	99	20	34	100	199
Muito Bom	0	4,999	0	39	0	79	0	19	0	99

São utilizadas diversas técnicas de calibração para avaliar a qualidade do ar em território nacional. Estas são regulamentadas pelo artigo 6.º do Decreto-Lei n.º 102/2010, de 23 de setembro [7]. Os métodos de calibração podem ser caracterizados por medições fixas, medições indicativas, modelação ou estimativas objetivas.

As medições fixas fazem uso de métodos de referência ou o seu equivalente. São efetuadas num local estacionário. As medições podem ser realizadas continuamente ou periodicamente. As medições indicativas são processos de medição de dados com menor rigor que os objetivos definidos para as medições fixas. A modelação é um método de simulação dos fenómenos que ocorrem na natureza. Este método permite estimar a concentração dos poluentes baseando-se num conjunto de variáveis que influenciam. Por último, as estimativas objetivas são métodos de avaliação que permitem estimar as concentrações de poluentes atendendo a objetivos de qualidade do ar menos rigorosos que o método da modelação [9].

De acordo com o Decreto-Lei n.º 102/2010, os regimes de avaliação da qualidade do ar para os parâmetros poluentes são estabelecidos com base numa comparação dos níveis de qualidade do ar nas zonas e aglomerações ao longo dos últimos 5 anos. Para tal são usados limiares inferiores/superiores. Estes limiares podem ser definidos para cada parâmetro poluente. É considerado que uma determinada região aumentou as suas concentrações de poluentes caso um determinado limiar superior em algum dos locais monitorizados tenha sido excedido num período de 3 ou mais dos últimos 5 anos [9].

2.2. Estações de monitorização

As localidades urbanas possuem os maiores índices de poluição. A monitorização é realizada por entidades como autoridades governamentais, institutos científicos ou especialistas na área de saúde. Para tal, faz-se uso de estações de monitorização. As estações estão equipadas com instrumentos de referência devidamente certificados para a medição de parâmetros poluentes. O equipamento é caro e necessita de rotinas de manutenção e de calibração periódicas. Isto é necessário para assegurar uma melhor qualidade das medições e haver compatibilidade de dados entre várias estações [1]. As estações fixas servem-se dos métodos de referência.

Nos países mais desenvolvidos é possível utilizar redes de monitorização de qualidade do ar em tempo real. No entanto, os custos de instalação e manutenção são muito elevados. A capacidade de monitorização é limitada. Pois, estes sistemas apenas conseguem fornecer dados de maior precisão referentes às áreas geográficas onde se encontram instalados [1].

Há a alternativa de utilizar estações de monitorização móveis, geralmente são instaladas em veículos. Podem utilizar os mesmos métodos de instrumentação periódica que as estações fixas. As estações móveis são muito úteis nas campanhas de medição urbanas. Estas campanhas são medições estacionárias que ocorrem por um período limitado nas localidades onde não há cobertura das estações fixas. No entanto, as características migratórias das estações móveis não permitem dar cobertura continuamente numa determinada área [1].

2.3. Redes oportunistas

As redes oportunistas (*oppnets*) são sistemas que combinam os conceitos das redes *Ad Hoc*, ponto a ponto (*Peer to Peer P2P*) e as redes de sensores sem fios (RSSF). O conceito das redes oportunistas oferece as condições necessárias para a evolução ou expansão de redes de sensores já existentes. Estas ainda permitem desenvolver novos tipos de aplicações de sensores no âmbito das redes híbridas [10]. As redes oportunistas também podem ser designadas de *Delay Tolerant Networks* (DTN).

As redes oportunistas herdaram características das redes *Ad Hoc*. Os nós são capazes de comunicar entre si, mesmo sem haver uma rota para os conectar diretamente. A particularidade destes sistemas é a arquitetura de rede poder ser formada por regiões separadas. Cada região caracteriza uma rede oportunista diferente. Deste modo, pode-se afirmar que as redes oportunistas conseguem interconectar-se.

O princípio de encaminhamento está condicionado às necessidades da aplicação. A estratégia de encaminhamento é *store-and-forward* [11]. As redes oportunistas podem tirar partido de diferentes tecnologias de comunicações sem fios. Tais opções podem incluir Zigbee, Wi-Fi, Bluetooth ou LoRa (*Long Range*). A finalidade é encaminhar diversos tipos de dados e aceder à Internet. Outra vantagem das redes oportunistas é que estas são capazes de armazenar temporariamente os dados para uma futura transmissão. Esta é uma característica indispensável quando a conexão à Internet não está disponível.

As redes oportunistas têm aplicação nos sistemas de monitorização ambiental urbana. É bastante útil nas estações de monitorização móveis. Geralmente estas recorrem a veículos como por exemplo autocarros para atuarem como estações móveis. Estas plataformas podem integrar dispositivos que monitorizam parâmetros poluentes ou a velocidade do tráfego.

As redes oportunistas têm também aplicabilidade nas estações fixas. Deste modo é possível recolher dados de várias estações instaladas ao longo da cidade. Em regra geral, os dados são recolhidos e transmitidos de um modo oportunista tirando partido de um *gateway*. Este serve de ponto de acesso para a Internet. Por fim os dados são processados num servidor de base de dados para a tomada de decisões, no qual os utilizadores podem consultar online os dados de uma ou mais estações.

No caso das estações móveis, estas plataformas comunicam com o *backbone* da rede central. Estes sistemas geralmente são redes híbridas. As tecnologias alternativas de comunicação podem ser Wi-Fi ou redes celulares. Nas comunicações celulares o alcance é relativamente elevado, mas há problemas associados à cobertura de rede local. Tal deve-se a áreas congestionadas ou que não possuem a cobertura de rede necessária. As taxas de transmissão são relativamente baixas. No Wi-Fi ao transmitir dados a partir das estações de monitorização consegue-se débitos mais elevados. Em contrapartida, o alcance de transmissão é menor [11].

2.4. Redes de Sensores sem Fios

As redes de sensores sem fios são sistemas de comunicação de dados constituídos por vários nós sensores autónomos. Estes encontram-se dispersos ao longo de uma determinada área geográfica [12]. A finalidade é monitorizar determinados parâmetros ou eventos do ambiente onde estão localizados. Existem diversos tipos de aplicações para sistemas de redes de sensores. Um nó sensor é uma plataforma que pode integrar um ou mais sensores. Deste modo, pode monitorizar um ou mais parâmetros físicos do ambiente onde se encontra. As amostras medidas pelos sensores são processadas e transmitidas recorrendo a comunicações sem fios [13]. Na figura 2.3 a) está ilustrada a arquitetura interna de um nó sensor. Esta é composta por quatro subsistemas: unidade sensora, unidade de processamento e armazenamento, unidade de comunicação e sistema de fornecimento de energia.

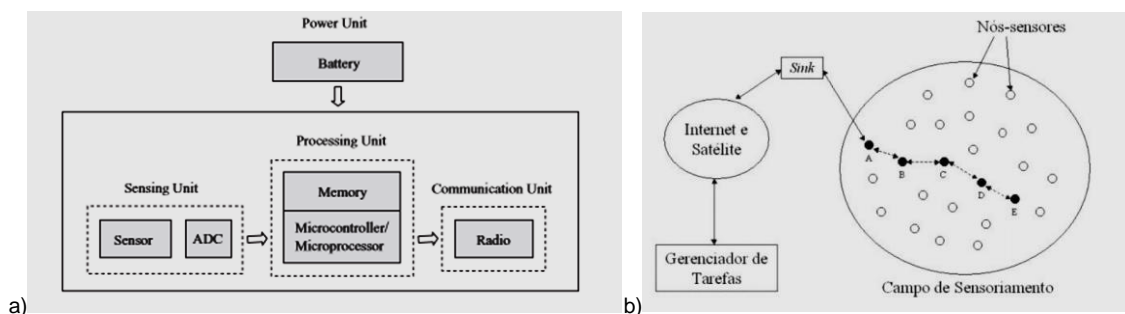


Figura 2.3 a) Arquitetura de um nó sensor [14]. b) Modelo genérico de uma rede de sensores [12].

A unidade sensora pode utilizar um sensor e um ADC ou “n” sensores e “n” ADC correspondentemente. A sua função é ler valores que caracterizam um determinado fenómeno físico. Estes valores são convertidos em sinais digitais. Posteriormente, é realizado o processamento dos dados. A unidade de processamento é caracterizada por um microcontrolador. Este obtém os dados amostrados, armazena-os na memória e fornece-os aos módulos de comunicação. A unidade de comunicação contém o módulo rádio e uma antena transcetora. Deste modo, a unidade de comunicação pode transmitir/receber sinais radiofrequência (RF). O módulo de comunicação internamente contém um oscilador. O sistema de fornecimento de energia tem a finalidade de assegurar a autonomia energética da estação. Tal implica a utilização de hardware que permita eficiência energética. Respeitando este princípio, consegue-se aumentar consideravelmente a autonomia energética nas redes de sensores cujos nós são alimentados por baterias. Para aplicações de longa duração cujos nós estejam instalados em locais de difícil acesso é necessário usar fontes de energia renovável.

Nas redes de sensores a finalidade é recolher dados de uma forma colaborativa entre os vários nós. Os dados podem ser dispersados pelos nós intermediários, os *routers*, que realizam encaminhamento multi-salto (*multi-hop*). Os dados são encaminhados para a estação base que é o nó *gateway*. Este serve como ponto de acesso para um servidor de base de dados. A figura 2.3 b) ilustra o modelo genérico de uma rede de sensores. Como os percursos de encaminhamento convergem para o *gateway*, tal obriga os nós vizinhos do *gateway* a terem de permanecer no estado ativo por maiores períodos, de modo a evitar perdas de pacotes provenientes de outros pontos da rede. Esta situação causa maior gasto energético para os nós vizinhos do *gateway*.

2.5. Zigbee

O Zigbee é uma tecnologia de comunicações sem fios adequada às redes de sensores de baixas taxas de transmissão [15]. Este faz uso da norma IEEE 802.15.4. A figura 2.4 apresenta uma topologia de rede Zigbee.

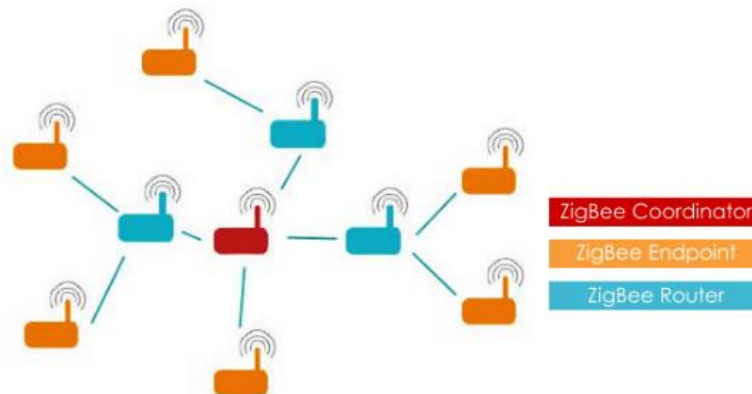


Figura 2.4 Topologia da rede Zigbee [15].

2.5.1. Camada Física

A finalidade da camada física (*Physical Layer PHY*) é transmitir/receber tramas PPDU (*PHY Protocol Data Unit*) do canal e prepará-las para enviar à camada MAC. São utilizados esquemas de modulação digital para transmitir a informação. Os mais utilizados são DSSS (*Direct Sequence Spread Spectrum*), O-QPSK (*Offset-Quadrature Phase Shift Keying*), BPSK (*Binary Phase Shift Keying*). O Zigbee opera na banda ISM (*Industrial Scientific Medical*). A região espectral mais utilizada é nos 2,4-2,4835 GHz, pois faz uso de um maior número de canais e as taxas de transmissão são mais elevadas. Na tabela 2.2 está ilustrado o IEEE 802.15.4, no que se refere a bandas espectrais, número de canais, esquemas de modulação e taxas de transmissão.

Tabela 2.2 Bandas de frequências utilizadas no protocolo IEEE 802.15.4: nº de canais, técnicas de modulação e taxa máxima de transmissão [16].

Frequency	Region	Modulation scheme	Bit rate (kbps)	Channels	Channel spacing
868 MHz	Europe	BPSK	20	1	N/A
915 MHz	America and Asia	BPSK	40	10	2 MHz
2.4 GHz	Global	O-QPSK	250	16	5 MHz

2.5.2. Camada MAC

A camada MAC (*Medium Access Control*) controla o acesso ao meio. Também pode interagir com camadas superiores através do uso e sincronização de *beacons*. Estes últimos são sinais RF que caracterizam pacotes de sinalização de mensagens. O coordenador Zigbee pode utilizar *beacons* de modo a obter sincronismo com os restantes dispositivos numa rede de sensores. O acesso ao canal é controlado por mecanismos de prevenção de colisões entre pacotes como o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Este é um protocolo de comunicação baseado num princípio de recepção recíproca entre emissor e recetor como ilustrado na figura 2.5 [17].

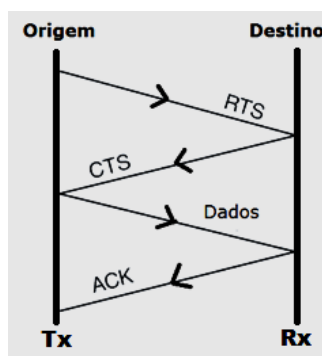


Figura 2.5 Sequência de ações protocolo CSMA/CA [17].

O emissor escuta o canal. Se estiver congestionado, a tentativa de transmissão é abortada. Em tal caso, esta será realizada posteriormente. Se o canal estiver livre durante uma determinada janela temporal (*Distributed Inter Frame Space DIFS*) a comunicação pode ser realizada [18]. Nestas condições o emissor transmite o pacote RTS (*Request to Send*). Este contém informação referente à mensagem de dados. O recetor responde com CTS (*Clear to Send*) assinalando ao emissor que pode transmitir a sua informação. Assim que o recetor recebe a mensagem de dados envia o pacote ACK (*Acknowledge*). Deste modo, é confirmado que os dados foram recebidos. Entretanto, as estações vizinhas aguardam durante um determinado período temporal. O objetivo é permitir a comunicação entre emissor e recetor sem haver risco de colisão.

2.5.3. Consumos energéticos

Os rádios Zigbee podem ser alimentados por baterias. Estas podem ser pilhas ou baterias de lítio. Sua vida útil é condicionada por factores como a capacidade da bateria (mAh) e o tipo de aplicação para o qual é usada. É possível otimizar a eficiência energética configurando o módulo de comunicação para utilizar o estado adormecido (*sleep mode*). Deste modo, o módulo de comunicação apenas utiliza o estado ativo para transmissão de dados periódica. Durante o restante tempo, encontra-se no estado adormecido. Neste período os consumos de corrente são da ordem dos μA .

2.5.4. Formato tramas API

Os pacotes de dados no formato API (*Application Programming Interface*) são designados de tramas. As tramas são responsáveis por encapsular os dados a transmitir. Elas podem ser transmitidas ou recebidas através da interface série do dispositivo.

Numa trama é encapsulado o conteúdo da mensagem e informação adicional como a fonte ou destino dos dados ou a qualidade de sinal. O conteúdo de dados interno pode variar, uma vez que está associado a diferentes ações ou eventos do dispositivo Zig Bee. Na figura 2.6 encontra-se ilustrado de um modo genérico a estrutura de uma trama Zigbee.

Start delimiter	Length		Frame data								Checksum
			Frame type	Data							
1	2	3	4	5	6	7	8	9	...	n	n+1
0x7E	MSB	LSB	API frame type	Frame-type-specific data							Single byte

Figura 2.6 Estrutura genérica para uma trama Zigbee API [19].

0x7E é o byte de início da trama (*start delimiter*). Os campos MSB e LSB (*most/least significant byte*) caracterizam o comprimento em *bytes* da mensagem. A contagem do número de *bytes* da mensagem é realizada entre os campos *length* e *checksum*. O campo referente à estrutura da mensagem (*frame data*) é composto pelo *byte* que caracteriza o identificador de API e os dados específicos da trama. O *byte* referente ao identificador de API (*API frame type*) permite determinar o tipo de trama e este indica como a informação é estruturada no campo de dados da mensagem. O campo *data* é o conteúdo da mensagem de dados. A estruturação da informação da mensagem neste campo é condicionada pelo tipo de trama que tiver sido definida pelo identificador de API.

Por fim, o último *byte* da trama API caracteriza o *checksum*. Este *byte* permite testar a integridade dos dados da mensagem. O seu cálculo pode ser realizado através da soma *hash* de todos os *bytes* anteriores ao *byte* de *checksum*. São excluídos os três primeiros *bytes* da trama API que se referem ao *byte* de início e o comprimento da mensagem.

2.6. Wi-Fi

O IEEE 802.11, mais conhecido como Wi-Fi, é uma das tecnologias de comunicações sem fios mais populares. Embora os consumos do Wi-Fi sejam superiores aos do Zigbee, é possível utilizar o Wi-Fi eficientemente em aplicações de baixa potência. Tal pode ser realizado tirando proveito da estrutura da rede Wi-Fi existente e dos dispositivos Wi-Fi que são compatíveis com a rede IP. No Anexo A é descrito um estudo mais pormenorizado referente ao Wi-Fi nas redes de sensores. A figura 2.7 ilustra o modelo de uma rede Wi-Fi. Esta é capaz de integrar diversos dispositivos. Tais como *smartphones*, computadores ou qualquer outro tipo de dispositivo eletrónico capaz de interagir com a rede IP. É necessário referir nós sensores ou atuadores de baixo consumo. Estes dispositivos são associados a um ponto de acesso (*Access Point AP*) para a Internet. As operações mais relevantes numa rede Wi-Fi são inicialização/associação, mensagens *keep-alive*, transmissões periódicas de dados, transmissões de dados acionadas por eventos e mensagens de comando.

Nas operações de inicialização/associação, primeiro, o sensor é alimentado. Em seguida, este associa-se a um determinado ponto de acesso nas proximidades e adquire um endereço IP. A duração da inicialização é condicionada por factores como o esquema de segurança utilizado pela rede Wi-Fi a conectar-se e o método de atribuição do endereço IP. O que têm consequências nos consumos de energia.

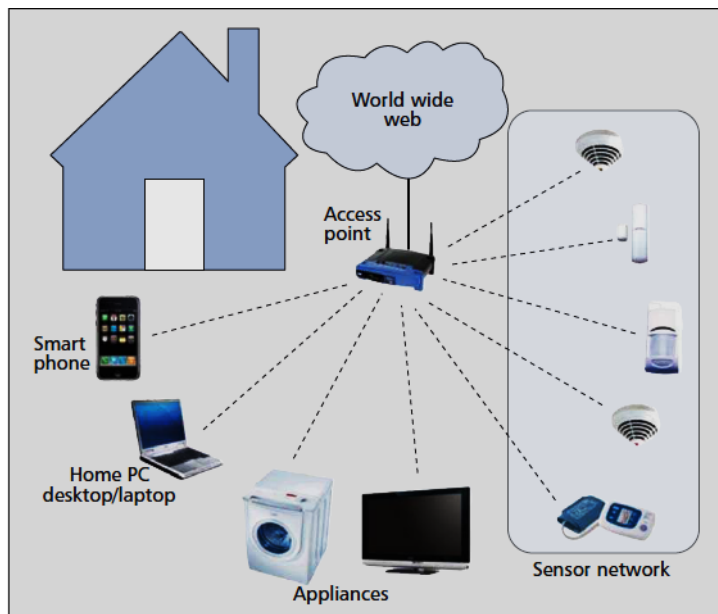


Figura 2.7 Modelo de uma rede Wi-Fi [20].

As mensagens *keep-alive* são pacotes que um dispositivo Wi-Fi transmite periodicamente ao ponto de acesso. A finalidade é manter o dispositivo Wi-Fi conectado à rede. Sem a transmissão destas mensagens o dispositivo Wi-Fi não seria detetado pelo ponto de acesso. Após um determinado período o ponto de acesso iria remover este dispositivo da sua lista de associações. Consequentemente, o dispositivo Wi-Fi perderia a conexão à rede.

Nas operações de transmissão periódica de dados é executada uma sequência de ações periódica. O nó começa por transitar do modo adormecido para o estado ativo. Em seguida, realiza amostragens e processamento de dados. Então transmite os seus pacotes de dados e retorna ao estado adormecido. Numa transmissão de dados acionada por eventos utilizam-se mensagens de alerta de eventos. Estas podem ser transmitidas a qualquer instante, o que implica um tempo de transmissão rápido e fiável. Neste tipo de operações o sensor monitoriza continuamente o ambiente. Se detetar um determinado evento é transmitida uma mensagem a reportar esse evento.

As mensagens de comando também podem ser transmitidas a qualquer instante. Os nós necessitam de receber estas mensagens em tempo real. Em contrapartida, os consumos de energia são maiores, pois o tempo de resposta do sensor é um factor crucial. Este necessita de uma janela temporal na qual retorna ao estado ativo e verifica a existência de mensagens de configuração. Este necessita de uma janela temporal na qual retorna ao estado ativo e deve verificar se há mensagens de configuração. Estas mensagens podem ser transmitidas ao sensor, como por exemplo uma mensagem de consulta de dados, ou a um dispositivo atuador, como por exemplo mensagens de configuração.

2.7. Bluetooth

O Bluetooth é outra tecnologia de comunicações sem fios. Este atua numa configuração de rede *ad hoc*. Opera na banda ISM nos 2,4-2,484 GHz. No entanto, o tráfego de dados nesta banda é elevado havendo por isso um elevado risco de interferências entre canais diferentes. Por esta razão, é utilizada a modulação em espalhamento espectral de salto em frequência (*Frequency Hopping Spread Spectrum FHSS*).

Idealmente, a velocidade de transmissão é da ordem de 1 Mbps, mas os problemas associados à latência limitam as taxas de transmissão até cerca de 722 kbps. O alcance de transmissão é da ordem dos 10 metros. É possível aumentá-lo até aos 100 metros, mas tal acarreta maiores consumos energéticos [21]. Um dos inconvenientes do Bluetooth é o consumo energético. Isto deve-se a diversos factores como pesquisa frequente por outras unidades, estabelecimento de conexões e as características da topologia de uma rede Bluetooth. No entanto, consegue-se reduzir os consumos tirando partido dos modos de baixa energia.

Numa aplicação de sensores baseada em Bluetooth os nós podem ser controlados por um ponto de acesso. Este encontra-se localizado geograficamente num ponto de interesse apropriado. O ponto de acesso pode ser um *smartphone*. Este comporta-se como uma unidade mestre. O ponto de acesso é responsável por decidir em que instantes um determinado nó deverá iniciar o seu ciclo ativo. Isto pode ser efetuado através de configurações. De tal modo que quando o ponto de acesso se conecta a esse nó, o mesmo inicie o seu ciclo ativo. Durante o ciclo ativo são realizadas amostragens e são transmitidos dados. Após encerrar o ciclo ativo a conexão é interrompida e o nó retorna ao estado adormecido.

As aplicações de sensores baseadas em Bluetooth implicam emparelhamento dos nós terminais de monitorização ao ponto de acesso. Também são necessárias tecnologias alternativas de comunicações sem fios. Respeitando este princípio, os nós terminais conectam-se ao mestre. O mestre é o ponto de acesso para o *backbone* da rede alternativa, ou seja, este assume a função de um *gateway*. Este é o princípio das comunicações híbridas no Bluetooth (*Bluetooth Bridging*) [22]. Isto permite utilizar vários nós terminais e permite interagir com mais dispositivos capazes de utilizar tecnologias alternativas de comunicações sem fios. Há uma tecnologia do Bluetooth adequada às aplicações de sensores no âmbito do *Internet of Things* (IoT). Este é o *Bluetooth Low Energy* (BLE) [23], o qual visa o baixo consumo energético. A versão base é o Bluetooth 4.0. No Anexo B é descrito um estudo em maior detalhe do Bluetooth clássico e *Bluetooth Low Energy*.

2.8. LoRa (Long Range)

A tecnologia LoRa (*Long Range*) é uma solução de comunicações sem fios para longos alcances de transmissão. Esta foi implementada pela LoRa Alliance. É adequada para sistemas autónomos nos quais a longa autonomia e os baixos consumos são parâmetros importantes.

2.8.1. Parâmetros radiofrequência

Na Europa, o LoRa está regulamentado para operar com uma potência de transmissão no máximo a 14 dBm e na banda de frequências dos 868 MHz. O LoRa serve-se de um esquema de modulação digital por espalhamento espectral. Este é o *Chirp Spread Spectrum* (CSS). Este pode ser caracterizado por [24],

$$s(t) = \sqrt{\frac{2E_s}{T_s}} \cos \left[2\pi f_c t \pm \pi \left(u \left(\frac{t}{T_s} \right) - w \left(\frac{t}{T_s} \right)^2 \right) \right] \quad (2.1)$$

no qual E_s é a energia de um símbolo e T_s a sua duração, f_c é a frequência da portadora de sinal. Os parâmetros u e w são o desvio em relação à frequência central e a largura de varrimento (*sweep width*) respetivamente. O *Chirp Spread Spectrum* utiliza modulação de pulsos de banda larga. A sua frequência varia dependendo da quantidade de símbolos transmitidos [25]. As características desta modulação permitem que a portadora RF tenha um determinado grau de imunidade contra os sinais interferidores ou ruído. Adicionalmente, a utilização de impulsos *Chirp* com uma largura de banda considerável é útil para minimizar os efeitos do desvanecimento. Deste modo, é possível aumentar o alcance, o que possibilita comunicações a maiores distâncias. Ainda é possível reduzir a potência de transmissão, deste modo, minimizam-se os consumos energéticos por parte dos nós terminais.

O *Chirp Spread Spectrum* integra um esquema de codificação *Forward Error Correction* (FEC). A finalidade é a correção de erros através da introdução de alguma redundância na trama. O fator de espalhamento espectral (*spreading factor*) ainda dá a possibilidade de transmitir pacotes a diferentes taxas de dados. Isto permite transmitir em simultâneo múltiplos sinais *Spread Spectrum* no mesmo canal sem haver degradação da relação sinal-ruído (*SNR - Signal Noise Ratio*). Os parâmetros frequência da portadora, fator de espalhamento espectral, largura de banda, taxa de codificação têm influência nos consumos energéticos, alcance e imunidade ao ruído [25]. O fator de espalhamento espectral é a razão entre a taxa de símbolos e a taxa de *chips*. Fatores de espalhamento espectral mais elevados possibilitam uma SNR mais elevada, melhor sensibilidade do recetor e maiores alcances de transmissão. A codificação FEC melhora a proteção da trama contra rajadas de bits errados. Em contrapartida, o maior fator de espalhamento espectral e de taxa de codificação aumentam a duração da trama. Isto resulta em maior duração do ciclo ativo e maiores consumos. O bit rate R_b no período ativo pode ser dado por [26],

$$R_b = \frac{S_F \cdot C_R}{\left(\frac{2^{S_F}}{BW} \right)} \quad (2.2)$$

no qual S_F é o fator de espalhamento espectral, C_R é a taxa de codificação FEC e BW a largura de banda. Ainda é possível obter esta expressão em função da taxa de símbolos R_s [27],

$$R_s = \frac{BW}{2^{S_F}} \quad (2.3)$$

em que a duração de um símbolo é definida por,

$$T_s = \frac{1}{R_s} \quad (2.4)$$

2.8.2. Camada MAC

O LoRa faz uso do protocolo proprietário LoRaWAN [28]. Este é otimizado especificamente para aplicações cujos nós terminais são autônomos energeticamente e baseia-se numa topologia em estrela. Os nós terminais comunicam com o servidor de base de dados através de *gateways*. Cada nó terminal pode transmitir a qualquer instante os seus dados servindo-se da taxa de transmissão disponível.

Se necessário, o *gateway* pode lhe instruir para realizar as transmissões numa taxa de dados específica. A versão 1.0 do protocolo LoRaWAN define 3 classes de nós. Estas são categorizadas em classes A, B e C. A implementação da classe A é obrigatória. As restantes classes são opcionais. Na figura 2.8 a) encontra-se ilustrada a arquitetura de rede LoRa. A figura 2.8 b) ilustra a sequência de operações para as classes LoRaWAN.

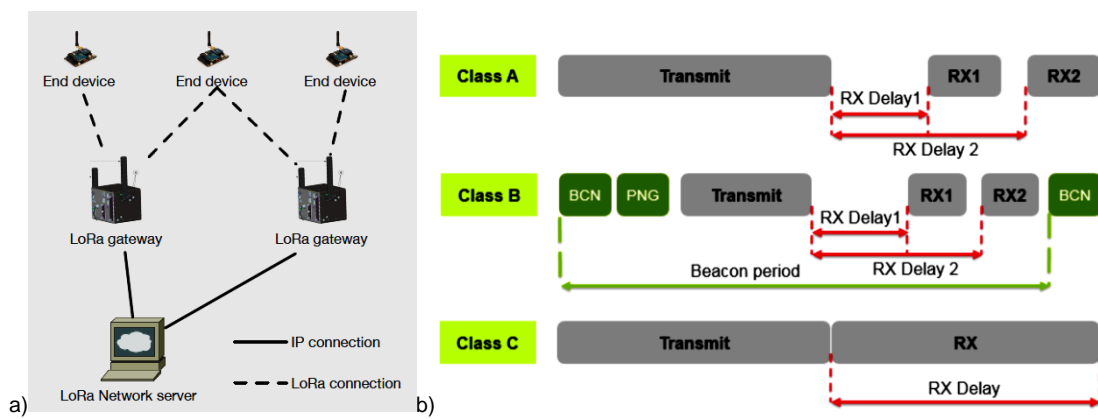


Figura 2.8 a) Arquitetura de rede LoRa [28]. b) Sequência de operações para as classes LoRaWAN [29].

Nos sistemas de classe A realizam-se comunicações em *uplink*, ou seja, do nó terminal de monitorização para o *gateway*. Há possibilidade de efetuar comunicações bidirecionais. Em tal caso, primeiro realiza-se a transmissão *uplink*. Após isso, o nó terminal abre duas janelas temporais (RX1, RX2) onde este pode receber os dados do *gateway*. Estas janelas temporais podem ocorrer no mesmo canal desde que os fatores de espalhamento espectral defiram da transmissão *uplink*. Se o nó terminal receber resposta de RX1, RX2 é omitido e vice-versa.

Para o caso da classe B, os nós assumem o mesmo comportamento descrito para a classe A. No entanto, para a receção dos dados utilizam-se *time slots* agendados. A finalidade é manter sincronização temporal entre os nós. Para tal, pode-se usar portadoras *beacon* periódicas. Já os nós classe C possuem um elevado número de *time slots*. O objetivo é permitir transmissões *downlink* a maior parte do tempo. Isto obriga o nó remoto a manter-se no estado ativo e escutar o canal continuamente. Naturalmente, isto tem maiores exigências energéticas comparativamente às outras classes, sendo uma solução inadequada em sistemas autônomos energeticamente.

2.8.3. Formato da trama LoRa

O formato completo de uma trama LoRa na camada física inclui preâmbulo, cabeçalho da mensagem, *payload* e o campo CRC do *payload*. O cabeçalho da mensagem e o *payload* CRC são opcionais. O pacote LoRa inicia-se com o preâmbulo. Este é composto por uma sequência de símbolos cuja frequência é variável ao longo do tempo. Estes impulsos são designados de chirps e ocupam toda a largura de banda disponível [25].

Os últimos 4,25 *chirps* caracterizam a palavra de sincronização (*sync word*) e o *start frame delimiter* (SFD) [25]. A palavra de sincronização é codificada pelos primeiros dois *chirps*. Este parâmetro é um *byte* que permite diferenciar redes LoRa que operem nas mesmas bandas espectrais. Por exemplo, um dispositivo configurado com uma determinada palavra de sincronização não irá escutar uma eventual transmissão caso a palavra de sincronização pertencente a esse pacote seja diferente. Os últimos 2,25 *chirps* são responsáveis por codificar a informação do *start frame delimiter*. A dimensão do preâmbulo pode variar desde os 10,25 até aos 65539,25 símbolos, sendo a duração do preâmbulo definida por [30],

$$T_{pre} = (n_{pre} + 4,25). T_{chirp} \quad (2.5)$$

no qual n_{pre} caracteriza o número “n” da sequência de símbolos do preâmbulo. O termo 4,25 caracteriza os *chirps* da palavra de sincronização e do *start frame delimiter*. T_{chirp} caracteriza a duração de um impulso *chirp* por símbolo. A figura 2.9 ilustra o formato de uma trama LoRa.

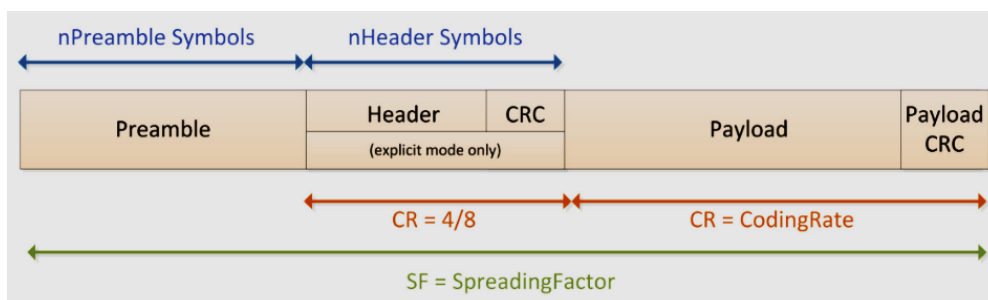


Figura 2.9 formato da trama LoRa ao nível da camada física [30].

O cabeçalho da mensagem quando utilizado é transmitido com uma taxa de codificação 4/8. Aqui é indicada a dimensão em *bytes* do *payload*, a taxa de codificação utilizada para o fim da transmissão e a eventual existência de um campo de codificação CRC para aplicar ao *payload*. Existe um campo de codificação CRC para aplicar ao cabeçalho de modo a assegurar a sua integridade. O recetor irá descartar pacotes cujos cabeçalhos sejam inválidos. O cabeçalho pode ser desativado em situações nas quais o recetor já conhece a dimensão do pacote, taxa de codificação e a presença de campo CRC. Após o cabeçalho transmite-se o *payload* e o campo de codificação CRC opcional para o *payload*. No Anexo C é brevemente estudado o formato da mensagem MAC LoRaWAN.

O *payload* é programável e o seu comprimento em bytes é variável, a sua duração é influenciada pelo método de codificação dos dados, os quais podem ser codificados de duas formas distintas. A codificação pode ser definida conforme especificado no cabeçalho do pacote LoRa como ilustrado na figura 2.9, este é o modo explícito (*explicit mode*).

Alternativamente, a codificação pode ser efetuada servindo-se dos registos de memória do módulo de comunicação, este é o modo implícito (*implicit mode*). A duração do *payload* é dada por [30],

$$T_{payload} = \begin{cases} T_{chirp} \left[8 + \left(\frac{8*PL-4*SF+24}{4*SF} \right) \right] (CR + 4) & \text{Modo implícito} \\ T_{chirp} \left[8 + \left(\frac{8*PL-4*SF+44}{4*SF} \right) \right] (CR + 4) & \text{Modo explícito} \end{cases} \quad (2.6)$$

no qual PL caracteriza o número de bytes do *payload*, cujo máximo é 255, SF o fator de espalhamento espectral e CR a taxa de codificação. O tempo total de transmissão do pacote LoRa é [30],

$$T_{pkt} = T_{pre} + T_{payload} \quad (2.7)$$

2.8.4. Alcance de transmissão

As redes LoRa conseguem abranger áreas vastas sem necessitarem de encaminhamento multi-salto, pois estes módulos têm alcances da ordem de km nos ambientes exteriores e várias dezenas de metros em ambientes interiores. Nas áreas urbanas o alcance é possível até aos 5 km e nas áreas rurais até aos 15 km. Os recetores tem sensibilidades na ordem dos -140 dBm. Como tal, o *link budget* total consegue atingir os 155 dB [31]. A potência do sinal recebido P_{RX} está condicionada pela potência transmitida P_{TX} e os restantes ganhos e perdas ocorrentes numa transmissão [24],

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{PL} - L_M + G_{RX} - L_{RX} \quad (2.8)$$

no qual P_{RX} e P_{TX} são as potências recebida/transmitida expressas em dBm, G_{TX} e G_{RX} são os ganhos da antena emissora/recetora expressas em dBi. L_{TX} caracteriza as perdas do emissor RF, como, por exemplo, o modulador ou os conectores de adaptação de impedância, L_{PL} são as perdas de percurso (*pathloss*), L_M são as perdas resultantes do desvanecimento e outras perdas adicionais e L_{RX} são as perdas no recetor RF, como, por exemplo, os conectores ou o desmodulador. A sensibilidade S no recetor é dada por [24],

$$S = -174 + 10 \log_{10}(BW) + N_F + SNR \quad (2.9)$$

no qual o primeiro termo da expressão descreve o ruído térmico a 1 Hz de largura de banda, sendo influenciado pela variação da temperatura do recetor, BW é a largura de banda, N_F é a figura de ruído e SNR é a relação sinal-ruído expressa em dB. A SNR depende do factor de espalhamento espectral S_F . Com um fator de espalhamento espectral mais elevado, a SNR aumenta.

2.8.5. Consumos de energia e regulamentação *duty cycle*

A figura 2.10 apresenta o ciclo de atividade para um emissor LoRa, a maioria do tempo, o nó encontra-se no estado adormecido. Considerando o maior *duty cycle* regulamentado para sistemas baseados em LoRa (1%) o emissor encontra-se 99% do tempo no estado adormecido [32]. O período ativo T_{cycle} correspondente caracteriza-se por ser a restante janela temporal desse período [29],

$$T_{cycle} = 100 * \frac{T_{pkt}}{Duty_Cycle} \quad (2.10)$$

em que T_{pkt} caracteriza o tempo necessário para transmitir um pacote, determinado pela expressão (2.7). Isto permite otimizar a eficiência energética do nó terminal, pois no estado ativo o microcontrolador tem maior consumo de corrente.

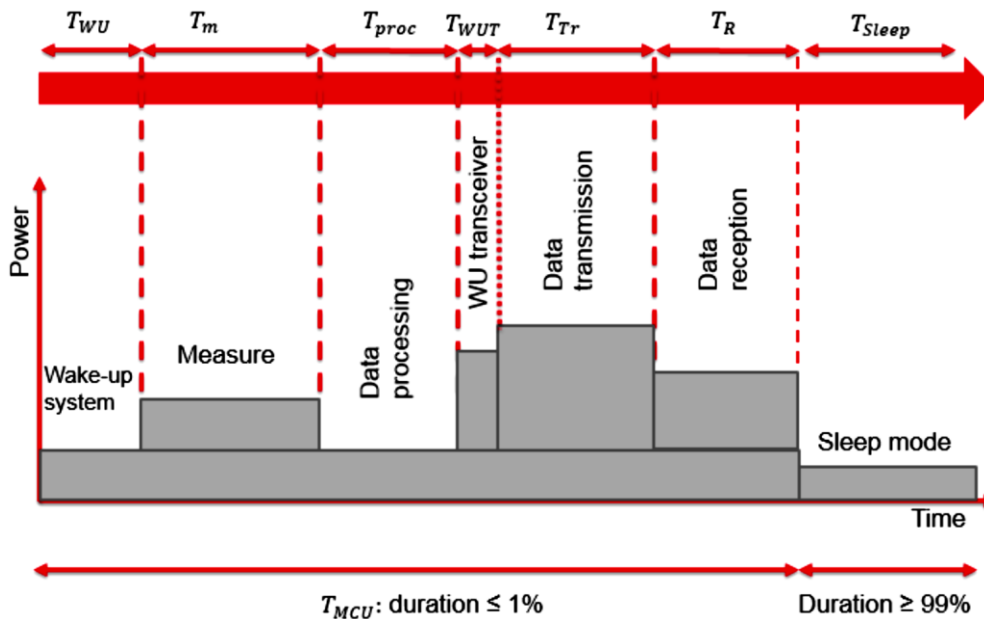


Figura 2.10 Ciclo de atividade de um nó emissor [29].

No modo adormecido o microcontrolador suspende suas operações, inativa seus ADC e o módulo de comunicação. Apenas se mantém o ativo o relógio interno do processador. A finalidade deste último é atuar como temporizador até ao início do ciclo de atividade seguinte. Os consumos de corrente no modo adormecido são da ordem de vários μA . Isto é condicionado por diversos factores como as características dos fabricantes e as configurações utilizadas na aplicação desenvolvida. O ciclo ativo de um sistema baseado em LoRa pode ser expresso [29],

$$T_{MCU}(f_{MCU}) = T_{WU} + T_m + T_{proc}(f_{MCU}) + T_{WUT} + T_{Tr} + T_R \quad (2.11)$$

em que $T_{MCU}(f_{MCU})$ é a duração do período de atividade do microcontrolador, T_{WU} caracteriza o tempo de retorno do microcontrolador ao ciclo ativo, T_m é o tempo necessário para amostragens, $T_{proc}(f_{MCU})$ é o período de processamento dos dados, sendo a duração desta janela temporal condicionada pela frequência interna do processador e número de instruções realizadas, T_{WUT} é o período de ativação do módulo rádio e T_{Tr} e T_R caracterizam os tempos necessários para transmitir a trama de dados e receber o pacote ACK respetivamente.

2.9. Exemplos de projetos de monitorização

Esta seção apresenta alguns exemplos de projetos de monitorização.

2.9.1. Projeto EuNet Air Quality

Em outubro de 2014, realizou-se em Aveiro uma campanha de medição com o objectivo de estudar a qualidade do ar num cenário urbano [33]. Nesta campanha participaram vários grupos pertencentes a institutos de pesquisas, universidades e empresas.

ESTADO DA ARTE

Foi utilizado um veículo como estação móvel de monitorização, ilustrado na figura 2.11. Este veículo integra vários módulos de monitorização ambiental. O carro deslocava-se pela cidade realizando medições em vários pontos da mesma. Os gases poluentes medidos foram CO, NO_x, O₃, SO₂. Também foram monitorizadas partículas em suspensão PM10, PM2.5. No veículo estava instalada uma torre meteorológica Vaisala WTX520. Os restantes parâmetros medidos são meteorológicos. Estes são temperatura, humidade, velocidade e direção do vento, radiação solar e níveis de precipitação. Os sensores estão instalados na torre meteorológica. Os dados monitorizados eram armazenados num servidor de base de dados a cada 15 minutos.



Figura 2.11 Laboratório móvel da qualidade do ar, realização da experiência na cidade de Aveiro [33].

O primeiro módulo descrito é o AUTH-ISAQ. Este foi baseado num nó comercial, que é o Waspmote™ desenvolvido pela Libellium. Este módulo utiliza sensores comerciais. O sensor de temperatura é o MCP9700/9700A *Linear Active Thermistor* da *Microchip*, com uma gama de medição dos -40 aos 125°C. O MPX4115A, da Motorola, mede a pressão atmosférica na gama de 15 a 115 KPa. O 808H5V5 da SENS, mede a humidade relativa. Este sensor serve-se do efeito capacitivo para medir a humidade. Estes dispositivos descritos são sensores de estado sólido. Foram utilizados os sensores MiCS 2710/2610 para medir os gases poluentes NO₂ e O₃. As gamas de medição são 0,05 a 5 ppm e 10 a 1000 ppb, respetivamente. Estes são sensores de óxido metal. O AUTH-ISAQ pode ser alimentado de forma autónoma por baterias, ou alternativamente servindo-se da rede elétrica de energia. Os dados monitorizados são armazenados num cartão SD a cada 5 minutos. Posteriormente, o ajuste destes dados é realizado servindo-se de curvas de calibração fornecidas pelos fabricantes. São também calculadas médias de resultados para os dados monitorizados a cada hora de amostras medidas.

Outro módulo instalado no veículo é o ECN Airbox. Este foi concebido para suportar condições climatéricas mais adversas. Aqui é realizada medição de partículas em suspensão PM10 e PM2.5 pelo sensor PPD42. Este é um sensor ótico cujo fabricante é a Shinyei. Este tem uma gama de medição até 200 µg/m³. Os níveis de NO₂ são medidos pelo 3E50, que é um sensor eletroquímico cujo fabricante é a Citytech. Este tem uma gama de medição até 1000 ppb. Este módulo também pode ser alimentado por baterias ou servindo-se da rede elétrica. Os dados são armazenados temporariamente num cartão SD. A cada 10 minutos estes são transmitidos para um servidor de rede através de GSM GPRS.

Outro módulo referido é o AQMesh. O nó foi utilizado em parceria pelo instituto IDAEA-CSIC e a empresa AQMesh. O módulo contém sensores do fabricante Alphasense. Estes monitorizam os gases poluentes NO, NO₂, O₃ e CO, com gamas de medição até 4000 ppb (NO/NO₂), 1800 ppb e 6000 ppb, respetivamente. Estes são sensores eletroquímicos. A temperatura e a humidade relativa são medidas pelo SHT21, da Sensirion. Este é um sensor de estado sólido que comunica por I2C. A pressão atmosférica é medida pelo MPL115A1, da Freescale, sendo também um sensor de estado sólido. A sua gama de medição é dos 500 aos 1500 mb. O AQMesh é alimentado por uma bateria de lítio. Os dados monitorizados são transmitidos por GSM GPRS para um servidor de base de dados. No veículo estão instalados 4 destes módulos. A finalidade é comparar resultados de dados.

O módulo EveryAware SensorBox foi desenvolvido para monitorizar poluição causada pelo tráfego urbano. São utilizados três sensores diferentes para medições CO. O primeiro é um sensor Alphasense CO-BF. Os restantes dois são sensores MiCS 5521/5525. O Alphasense CO-BF é um sensor eletroquímico que mede até 5000 p-p-m. Os sensores MiCS 5521/5525 são sensores óxido metal na gama de 1 até 1000 ppm. Para monitorizar temperatura/humidade relativa é utilizado o SHT21. Os restantes sensores que integram este sistema são sensores de óxido metal. As concentrações de NO₂ são medidas pelo MiCS-2710 que mede na gama de 0,05 a 1 ppm. Utilizam-se sensores do fabricante Figaro para monitorizar concentrações de gasolina/diesel. Estes são os Figaro 2201 com gamas de medição de 10 a 1000 ppm e 0,1 a 10 ppm, respetivamente. Nas medições de O₃ utilizou-se o MiCS-2610 com a gama de medição de 10 a 1000 ppb. O EveryAware SensorBox ainda está equipado com um sistema GPS, o que permite determinar a cada segundo o local das medições. É utilizado um módulo rádio Bluetooth que lhe possibilita transmitir em tempo real os dados monitorizados através de um smartphone.

2.9.2. Projeto Urbisnet

No Instituto Superior Técnico (IST) foi realizado o Urbisnet, um projeto em parceria com vários institutos científicos nacionais [34]. A finalidade consistia em utilizar estações móveis nos tejadilhos dos veículos para monitorizar parâmetros poluentes em vários pontos geográficos num cenário urbano. O tempo e a posição geográfica foram obtidos por GPS. Os dados monitorizados podem ser transmitidos de uma estação móvel para uma estação central servindo-se da rede GSM. Existe a alternativa de transmitir os mesmos via Wi-Fi e recolhê-los por *download* num computador.

A figura 2.12 ilustra a arquitetura interna da estação móvel Urbisnet. Segue no Anexo D uma breve descrição do seu *hardware*. Na estação Urbisnet são monitorizados gases poluentes CO, NO₂, CO₂, O₃ e SO₂. Ainda é monitorizada a temperatura e a humidade. O ADC utilizado para ler amostras foi o ADS7828, 12 bits, que utiliza a interface de comunicação I2C. Para monitorizar os gases poluentes CO/NO₂ é usado o sensor MiCS-4614. Já o Alphasense NDIR IRC-A1 monitoriza concentrações de CO₂. As concentrações de O₃ são monitorizadas pelo MiCS-2611. Os níveis de SO₂ são medidos pelo Alphasense SO2-BF. O Sensirion SHT25 monitoriza temperatura/humidade. O restante hardware da estação móvel inclui um microprocessador ARM, cartão SD/MMC e cartão SIM, módulos GSM/Wi-Fi e um recetor GPS.

ESTADO DA ARTE

Os sensores MiCS-4614/2611 são sensores de óxido metal. O seu método de deteção de gases é semelhante. O Alphasense IRC-A1 é um sensor ótico. Este serve-se do efeito da radiação infravermelha não dispersiva (*NDIR Non Dispersive Infra-Red*) para detetar concentrações de CO₂. O Alphasense SO2-BF é um sensor eletroquímico. Já o Sensirion SHT25 é um sensor de estado sólido. Este internamente possui sensores de humidade e temperatura e utiliza o protocolo I2C para comunicar.

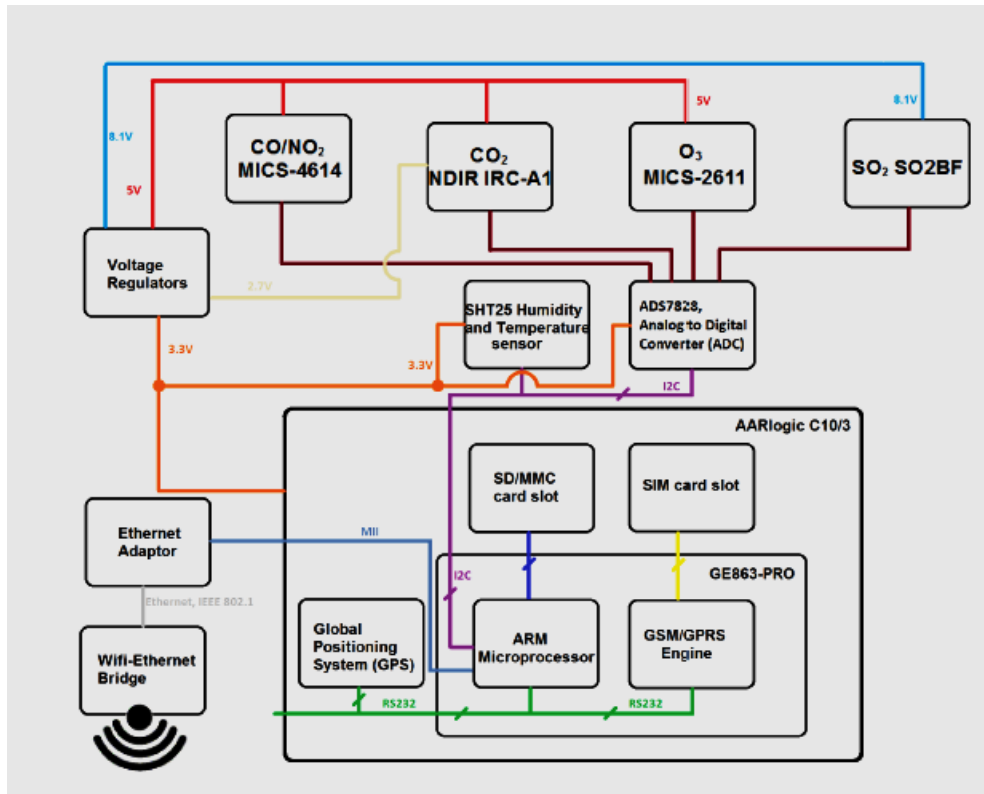


Figura 2.12 Diagrama de blocos da estação móvel Urbisnet [34].

A estação Urbisnet pode transmitir dados via Wi-Fi ou SMS por GSM. Tal permite que a estação interaja com um computador ou um servidor que recebe dados de mais estações móveis. Pode ainda interagir com uma rede composta de várias estações de monitorização e routers que encaminham os dados a um servidor de base de dados central [34]. A figura 2.13 ilustra o diagrama da rede Urbisnet [35].

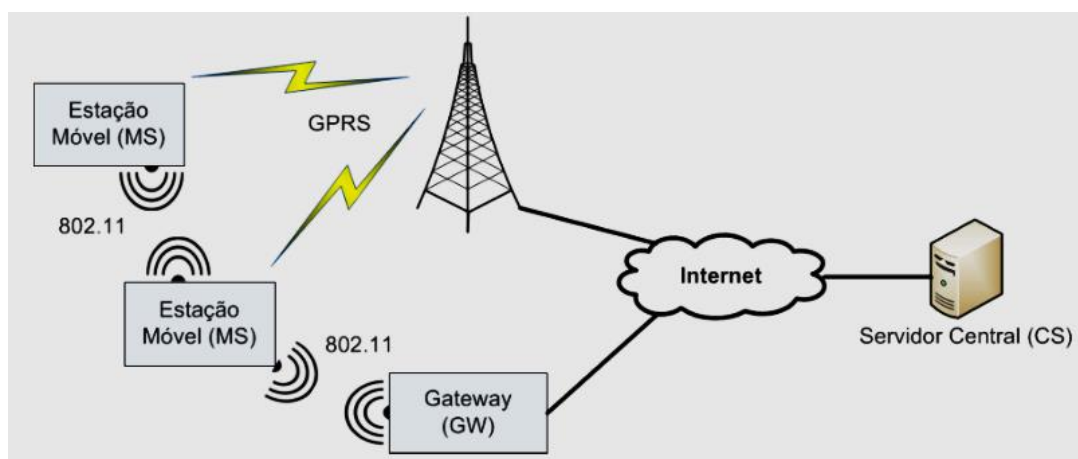


Figura 2.13 Diagrama da rede Urbisnet [35].

ESTADO DA ARTE

No caso de comunicações Wi-Fi, o computador deverá ter instalada uma interface de rede Wi-Fi e conectar-se à rede Wi-Fi da estação Urbisnet. Após isso, o utilizador efetua uma conexão Telnet. Para tal, escolhe a porta de comunicação de interesse e acede à estação Urbisnet servindo-se de uma consola Linux. No programa Urbisnet o utilizador especifica o endereço IP da estação de monitorização e a porta de comunicação para carregar as configurações da estação. Deste modo pode efetuar o *download* dos dados. O programa Urbisnet é executado paralelamente com o *software* utilizado pela estação de monitorização. O último é responsável por detetar as configurações enviadas pelo computador. A sua resposta depende das configurações definidas.

No caso de comunicações GSM, o computador deverá possuir porta série ou um adaptador conversor série-USB. Independentemente da opção este deve ser conectado num módulo GSM. A interface do módulo GSM faz uso de comandos AT para transmitir as configurações à estação Urbisnet. Após isto são recebidos os dados da estação de monitorização. O *software* executado na estação Urbisnet como suporta comunicações GSM há a liberdade de realizar ações remotamente. A estação Urbisnet pode ser controlada remotamente por um firmware Linux.

A figura 2.14 ilustra a janela principal do *software* Urbisnet num computador. Este trata-se de um exemplo de comunicações Wi-Fi entre a estação de monitorização e o computador. O *software* Urbisnet envia dados de configuração para a estação de monitorização. O programa da estação guarda estes parâmetros de configuração num ficheiro txt. Sempre que o programa na estação de monitorização é inicializado este ficheiro de configuração é lido. A estação irá atuar de acordo com as configurações definidas.

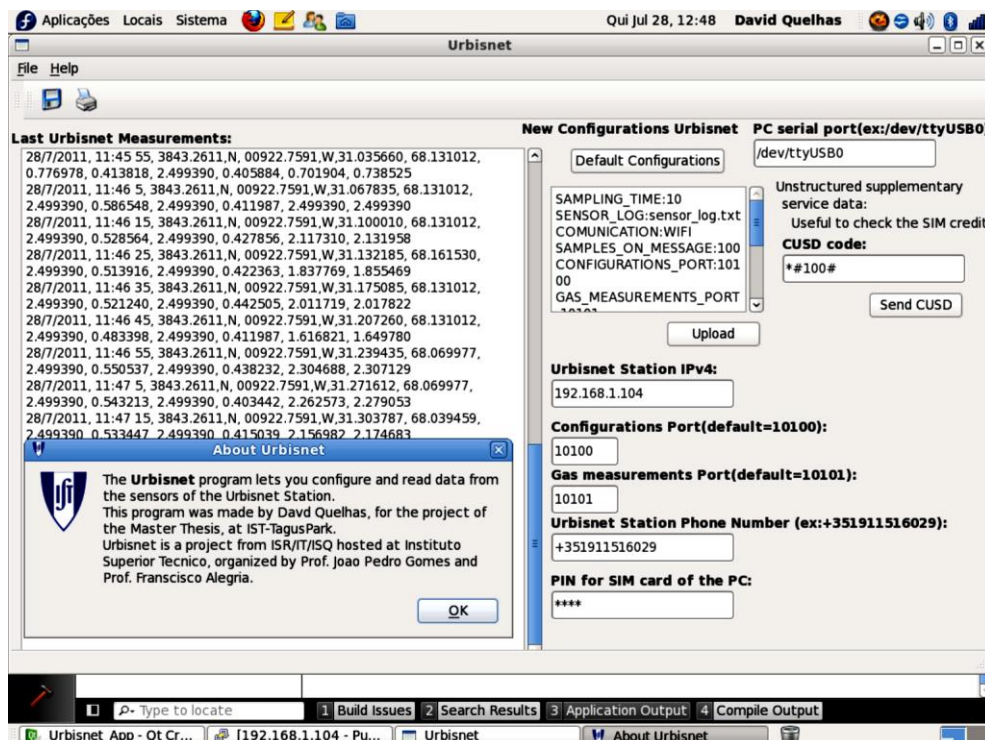


Figura 2.14 Exemplo de comunicações Wi-Fi entre a estação de monitorização Urbisnet e o computador. Software Urbisnet no computador [34].

2.9.3. Projeto uSense

O uSense é um sistema de monitorização da qualidade do ar em tempo real para cenários urbanos [36]. O sistema foi desenvolvido no Departamento de Engenharia de Informação da Universidade de Pisa em Itália. Cada utilizador pode servir-se de um nó que realiza medições periódicas para monitorizar concentrações de gases poluentes. Os dados monitorizados são enviados a um servidor Web. Deste modo os vários utilizadores podem partilhar os seus dados e ter cobertura sobre vários pontos da cidade. A figura 2.15 apresenta a arquitetura de rede do uSense [36]. No Anexo E segue uma breve descrição do *hardware* para um dos nós de monitorização utilizados.

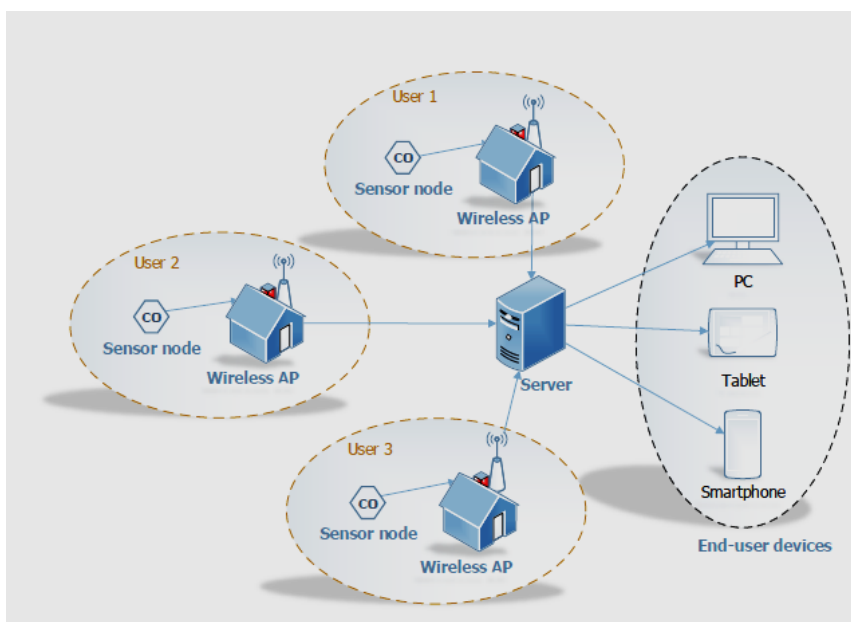


Figura 2.15 Arquitetura do sistema uSense [36].

São lidas amostras dos gases poluentes CO, NO₂, O₃. É tida em consideração uma melhor eficiência energética. Deste modo, minimizam-se os consumos de corrente e maximiza-se a autonomia das baterias. Para tal, faz-se os sensores comutar entre modo ativo/adormecido periodicamente. Isto é conseguido controlando o *duty cycle*. De forma sequencial, cada sensor é ativado e realiza amostragens. Cada sensor de gás necessita do seu período temporal de estabilização após a ativação. Apenas após o seu período de estabilização, o sensor é capaz de realizar amostragens com resultados coerentes.

Os dados monitorizados são encapsulados num pacote. Este contém identificador de nó e o timestamp. Em seguida, o módulo rádio é inicializado. Se a conexão à Internet está disponível, a estação entra em contacto com o servidor uSense. Os dados são transmitidos por HTTP (*Hypertext Transfer Protocol*). Caso não haja conexão à Internet os dados são armazenados temporariamente num cartão SD disponível na estação de monitorização. Assim que houver conexão os dados são transmitidos ao servidor. A transmissão pode ocorrer por meio de Routers Wi-Fi ou Access Points. Estes fornecem cobertura de rede local aos nós. Entretanto a estação de monitorização retorna ao modo adormecido. O único hardware que se mantém ativo é o RTC (*Real Time Clock*). Este realiza operações de temporização e acorda o microcontrolador para o próximo ciclo ativo.

ESTADO DA ARTE

O servidor uSense recebe dados de cada estação de monitorização. Aqui são determinados os índices de qualidade do ar (IQA). Todos estes dados são armazenados numa base de dados. A informação aí armazenada consiste no identificador do nó, *timestamp*, dados de coordenadas geográficas, dados de IQAr e concentrações gasosas. O servidor uSense também contém uma página Web, na qual, são apresentados os dados de cada estação de monitorização. Aqui também é calculado o IQAr. O servidor Web utilizado é o Apache. A página Web está programada em PHP. O servidor Web faz uso do Google Maps para visualizar a localização de cada estação e respetivos IQAr determinados na área.

O servidor Web ainda possui um Website uSense. Aqui os utilizadores podem consultar o IQAr. Esta ferramenta ainda dá liberdade para mais ações como criar uma conta, associar os nós a essa conta, ajustar coordenadas geográficas dos nós, selecionar nós aleatoriamente no mapa. A última opção pode ser usada para consultar o histórico de medições realizadas por um determinado nó. Também é possível a consulta de um mapa de poluição ilustrado a cores que qualificam o IQAr. Há também um servidor de aplicação que permite o acesso aos dados por parte de apps smartphones, isto facilita o acesso aos utilizadores. Pois uma app smartphone é capaz de oferecer os mesmos recursos que o servidor Web. Adicionalmente, pode tirar partido do recetor GPS do smartphone. Isto permite a visualização num mapa dos nós e analisar o IQAr.

A figura 2.16 ilustra a página Web uSense [36]. Esta apresenta um mapa contendo a localização geográfica de alguns nós na cidade. É sinalizado por cores os níveis de IQAr nos locais onde estão instaladas as estações de monitorização. A figura 2.17 apresenta uma app android que monitoriza os dados do sistema uSense. A figura 2.17 a) apresenta a página inicial e a figura 2.17 b) apresenta o mapa contendo a localização geográfica de alguns nós [36].

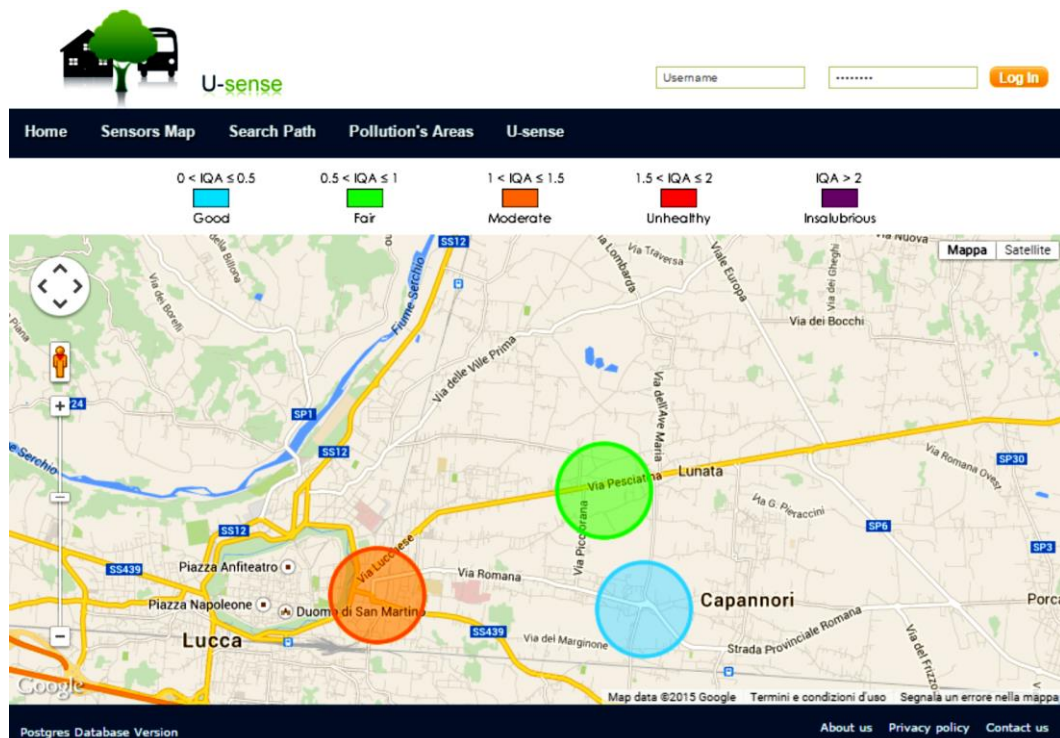


Figura 2.16 Visualização na página Web uSense através do Google Maps da localização geográfica de três nós numa região em Itália e respetivos índices de qualidade do ar [36].

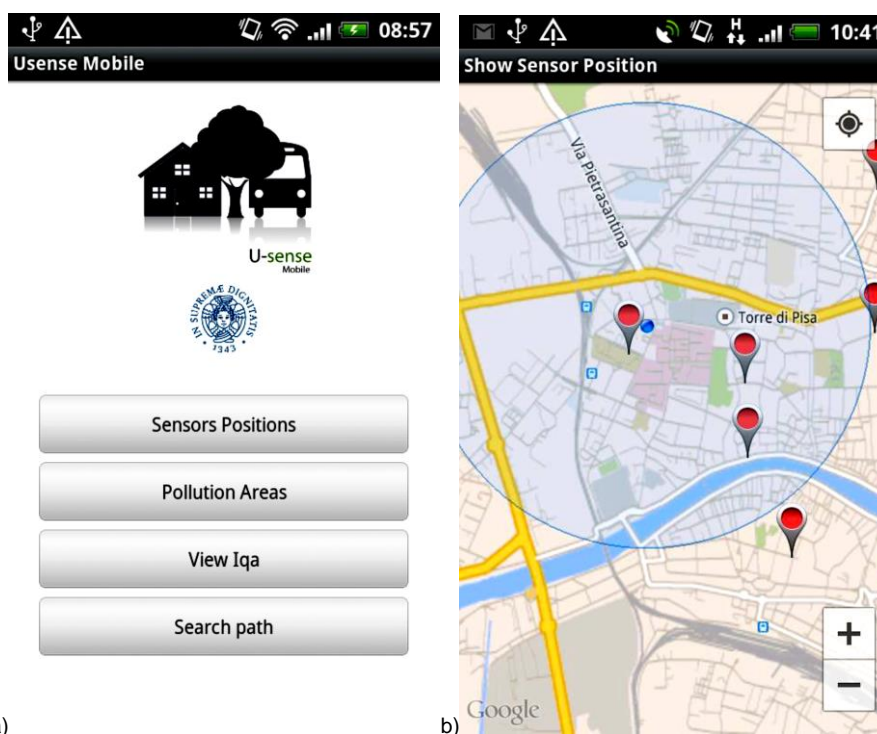


Figura 2.17 App android que utiliza uSense [36]: a) Página Inicial. b) Localização geográfica dos sensores mais próximos.

2.9.4. Rede de monitorização urbana LoRaWAN

Na primavera de 2017, realizou-se um estudo da qualidade ambiental urbana em Delft, nos Países Baixos [37]. O objetivo era estudar os problemas ambientais existentes no centro histórico da cidade e consciencializar a população. Este projeto foi realizado por estudantes da Universidade Técnica de Delft. Este tratava-se de um projeto no âmbito do tema das cidades inteligentes (*Smart Cities*), o que motivou o interesse do município local e outras entidades como a empresa de telecomunicações KPN.

Instalaram-se diversas estações de monitorização fixas. Estas foram instaladas em pontos da cidade com diferentes perfis de terreno. A finalidade era comparar os efeitos do tráfego urbano. Mediram-se dados de qualidade do ar, poluição sonora, temperatura e humidade. Cada estação de monitorização processava os seus dados e transmitia a um servidor de base de dados para uma posterior análise. Os cidadãos podiam aceder ao servidor e consultar os dados monitorizados. Neste projeto foi utilizada a tecnologia rádio de longo alcance LoRa.

As estações de monitorização foram implementadas de modo a atender às características dos módulos RF LoRa. Como tal eram transmitidos pequenos pacotes de dados a distâncias de alguns km. Houve cuidados quanto à redução dos consumos energéticos. A empresa de telecomunicações KPN forneceu cobertura de rede nos locais de instalação dos nós. Para tal, utilizaram-se gateways com tecnologia LoRa integrada. O microcontrolador utilizado foi o Pycom Lopy, a sua utilização foi motivada devido à sua eficiência energética. Esta placa de desenvolvimento ainda integra uma interface LoRaWAN. Cada estação de monitorização faz uso destes microcontroladores.

A nível de *hardware* cada estação ainda integra uma antena LoRa externa e conjuntos de sensores integrados numa placa de circuito. Para a monitorização da qualidade do ar utilizou-se o sensor PM5003, da PlanTower. Este serve-se do protocolo UART. No estudo da poluição sonora usou-se o MAX9814, da Maxim Integrated. O sensor contém um microfone acoplado e serve-se do protocolo OneWire. No que se refere à humidade e temperatura foi usado o AM302, da Kuongshun. Cada estação é alimentada por uma bateria instalada na placa de desenvolvimento. É utilizado um painel fotovoltaico com a finalidade de otimizar a autonomia da estação. A figura 2.18 apresenta o modelo da rede de sensores urbana [37].

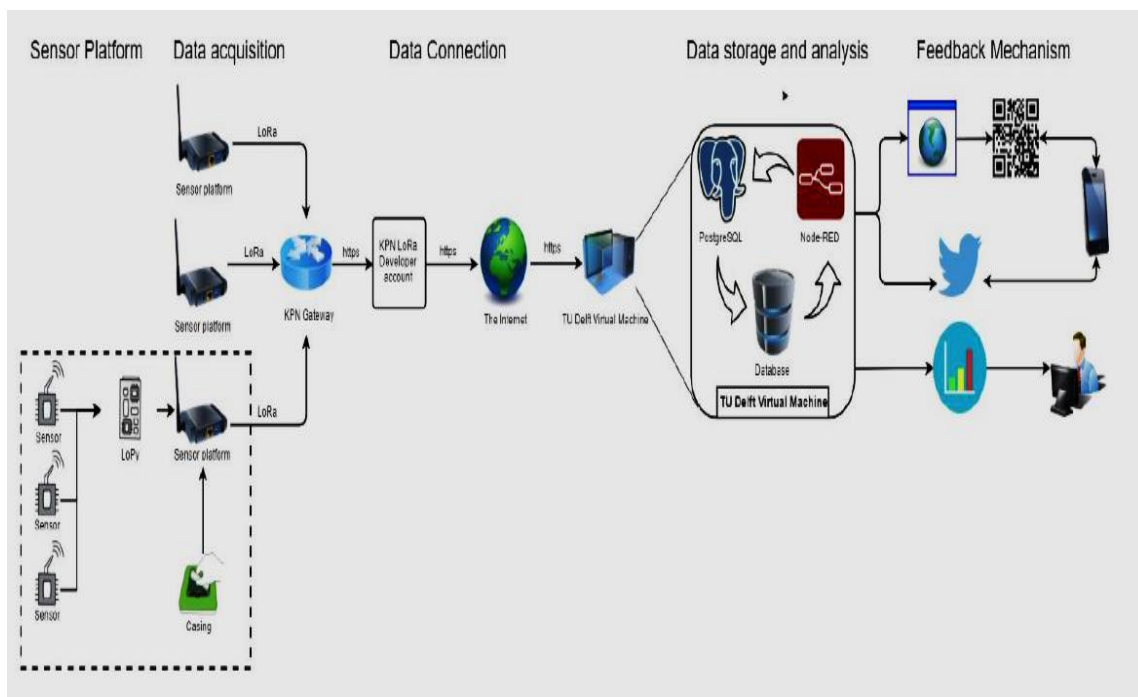


Figura 2.18 Modelo da rede de sensores urbana desenvolvida em Delft [37].

O programa executado no microcontrolador foi desenvolvido na linguagem de scripting MicroPython. São realizadas leituras de valores dos sensores, processamento e transmissão de dados. Os dados monitorizados são a temperatura, a humidade, o ruído e a qualidade do ar. Estes dados são processados e encapsulados num *payload* LoRa. Estes são transmitidos à gateway LoRaWAN. Em seguida o microcontrolador entra no estado adormecido de modo a reduzir os consumos de energia.

A gateway LoRaWAN envia os dados a uma aplicação de *software*. Esta executa duas interfaces: Node-RED e PostgreSQL. O Node-RED é uma interface *open-source* que permite conectar dispositivos com aplicações para IoT. Já o PostgreSQL é também uma interface *open-source*, esta é utilizada como servidor de base de dados. O Node-RED é responsável por gerir o fluxo dos dados. Este recupera os dados provenientes da gateway LoRaWAN. Em seguida descripta cada mensagem LoRaWAN, analisa os dados recebidos e armazena-os na base de dados PostgreSQL. Ao armazenar os dados é registado na base o *timestamp*. Esta não é uma aplicação em tempo real. Visto que após a leitura de amostras e o período temporal necessário para transmitir o *payload* ainda são necessários 30 segundos para armazenar os dados no PostgreSQL.

2.9.5. Síntese – Projetos de Monitorização

Esta seção apresenta uma síntese resumida quanto aos projetos de monitorização investigados na seção 2.9, os quais se apresentam na tabela 2.3.

Os projetos EuNetAir Quality e Urbisnet são realizados em território nacional servindo-se de nós móveis. O primeiro trata-se de uma campanha de medição realizada em Aveiro, na qual participaram institutos científicos, universidades e empresas de vários países. Neste projeto foi realizada a monitorização da qualidade do ar assim como parâmetros meteorológicos. As tecnologias de comunicações sem fios utilizadas foram GSM GPRS e Bluetooth. O segundo realizou-se no Instituto Superior Técnico de Lisboa, o qual caracterizava o desenvolvimento de nós móveis para uma rede de monitorização conhecida como Urbisnet. As suas tecnologias de transmissão e recolha de dados foram GSM GPRS e Wi-Fi.

Os projetos uSense e Rede de monitorização LoRaWAN realizaram-se na Itália e nos Países Baixos respetivamente, ambos se serviam de redes de monitorização com nós fixos. No projeto uSense realizava-se a monitorização de gases poluentes e parâmetros meteorológicos. A tecnologia de comunicação sem fios seria Wi-Fi. No caso da rede LoRaWAN adicionalmente seria realizada a monitorização quanto à poluição sonora. Os nós de monitorização assim como as *gateways* iriam-se servir da tecnologia LoRa. A cobertura de rede para os locais de instalação dos vários nós seria realizada pela empresa de telecomunicações KPN.

Tabela 2.3 Síntese dos projetos de monitorização.

Projetos de Monitorização	Sensores	Comunicação
Campanha EuNet Air Quality Aveiro	Nós móveis: <ul style="list-style-type: none"> ✓ Gás ✓ Partículas poluentes ✓ Parâmetros meteorológicos 	<ul style="list-style-type: none"> ✓ GSM GPRS ✓ GPS ✓ Bluetooth
Rede Urbisnet Lisboa	Nó móvel: <ul style="list-style-type: none"> ✓ Gás ✓ Parâmetros meteorológicos 	<ul style="list-style-type: none"> ✓ Wi-Fi ✓ GSM GPRS ✓ GPS
Rede uSense Itália	Nó fixo: <ul style="list-style-type: none"> ✓ Gás ✓ Parâmetros meteorológicos 	Wi-Fi
Rede de monitorização LoRaWAN Países Baixos	Nós fixos: <ul style="list-style-type: none"> ✓ Gás ✓ Poluição sonora ✓ Parâmetro meteorológicos 	<ul style="list-style-type: none"> ✓ Cobertura de rede nos locais de instalação dos nós fornecida pela empresa de telecomunicações KPN. ✓ Nós de monitorização e gateways utilizam tecnologia LoRa integrada.

3. Desenvolvimento dos sistemas de recolha de dados para a rede de sensores em ambiente urbano

O capítulo 3 apresenta o trabalho a desenvolver no projeto. O tema essencial centra-se em soluções de recolha de dados aplicadas no desenvolvimento de uma rede de monitorização urbana. Deste modo, é apresentada a arquitetura da rede de monitorização urbana e os seus requisitos de implementação. São apresentados os nós sensores desenvolvidos no projeto. Estes são um nó de calibração, um nó de monitorização local, um nó *gateway* de aquisição de dados e um sistema de monitorização remota. As tecnologias de comunicações sem fios são o Zigbee e o LoRa.

3.1. Sistemas implementados

Esta seção apresenta os sistemas implementados para o desenvolvimento da rede de monitorização urbana:

- Os protótipos a serem desenvolvidos são: um nó de calibração, um nó fixo local, um *gateway* de aquisição de dados e um sistema para monitorização remota;
- A monitorização tem em consideração a autonomia energética do nó e o gás poluente de interesse. Deste modo, é monitorizado o nível de tensão das baterias, o sinal elétrico de tensão lido do sensor de interesse e o respetivo gás poluente. O parâmetro poluente estudado é o monóxido de carbono (CO);
- São utilizados dois sensores para a monitorização de CO: Alphasense B4, Figaro TGS 5042. O primeiro tem custos relativamente elevados, melhor resolução, gama dinâmica limitada. O fabricante realiza a pré-calibração do sensor e fornece o respetivo circuito de condicionamento de sinal. Tais características determinam a sua função como sensor de calibração, nomeadamente as baixas concentrações de gás. O Figaro TGS 5042 é o sensor de escolha para os restantes protótipos a desenvolver, dado este ter menores custos;
- O nó sensor local realiza monitorizações ambientais periódicas. Dado o facto deste ser fixo é necessário um painel solar como fonte de captação de energia. Este utiliza a tecnologia rádio Zigbee;
- O *gateway* de aquisição de dados é uma solução de recolha de dados para os nós móveis. Os dados de interesse são encapsulados numa trama de receção do protocolo Zigbee. A *gateway* pode ser conectada a um computador que realiza o armazenamento dos dados;
- O sistema para monitorização remota necessita de dois nós: terminal remoto e *gateway* LoRa. Ambos se tratam de nós fixos. O terminal remoto destina-se à monitorização ambiental, este tem características semelhantes ao nó local, exceção feita à utilização da tecnologia rádio LoRa. O *gateway* LoRa permite integrar o terminal remoto na rede de sensores da Universidade. Este serve-se das tecnologias Zigbee e LoRa;
- A tecnologia base de comunicações sem fios é Zigbee. Deste modo, é necessária a configuração dos módulos RF como Router Zigbee. Estes comunicam com o Coordenador Zigbee, o qual está instalado no laboratório de Telecomunicações conectado a um computador;

DESENVOLVIMENTO

- A recolha dos dados a partir do Coordenador pode ser realizada por uma aplicação Java, a qual é executada no *software* NetBeans. Essencialmente, os dados são lidos por comunicação série e inseridos numa base de dados. A aplicação Java está estruturada para ler mensagens no formato do protocolo Zigbee;
- O servidor de base de dados é um servidor MySQL. O acesso a este é possível utilizando a ferramenta Xampp, nomeadamente a ativação das portas de comunicação correspondentes ao MySQL e Apache. Este servidor permite armazenar os dados provenientes dos vários nós da rede. Há a liberdade para armazenar estes em diferentes bases de dados. A sua consulta pode ser realizada numa página Web;
- É importante que a rede de monitorização a desenvolver seja escalável de modo a permitir a sua expansão. Esta pode tirar partido da rede de sensores já existente na Universidade. Tal permite aos utilizadores a consulta online dos dados de interesse monitorizados;
- O utilizador tem a liberdade de consultar o histórico de informação referente ao nó de interesse. Esta informação inclui dados referentes ao nível da bateria e as concentrações poluentes medidas.

3.2. Arquitetura de rede de monitorização urbana

A figura 3.1 apresenta a rede de monitorização urbana a desenvolver. A tabela 3.1 apresenta os respectivos nós a desenvolver. Os protótipos da tabela 3.1 são descritos em várias vertentes como o fornecimento de energia, microcontrolador e sensor utilizados, parâmetros de monitorização ambiental, comunicações rádio e método de armazenamento de dados.

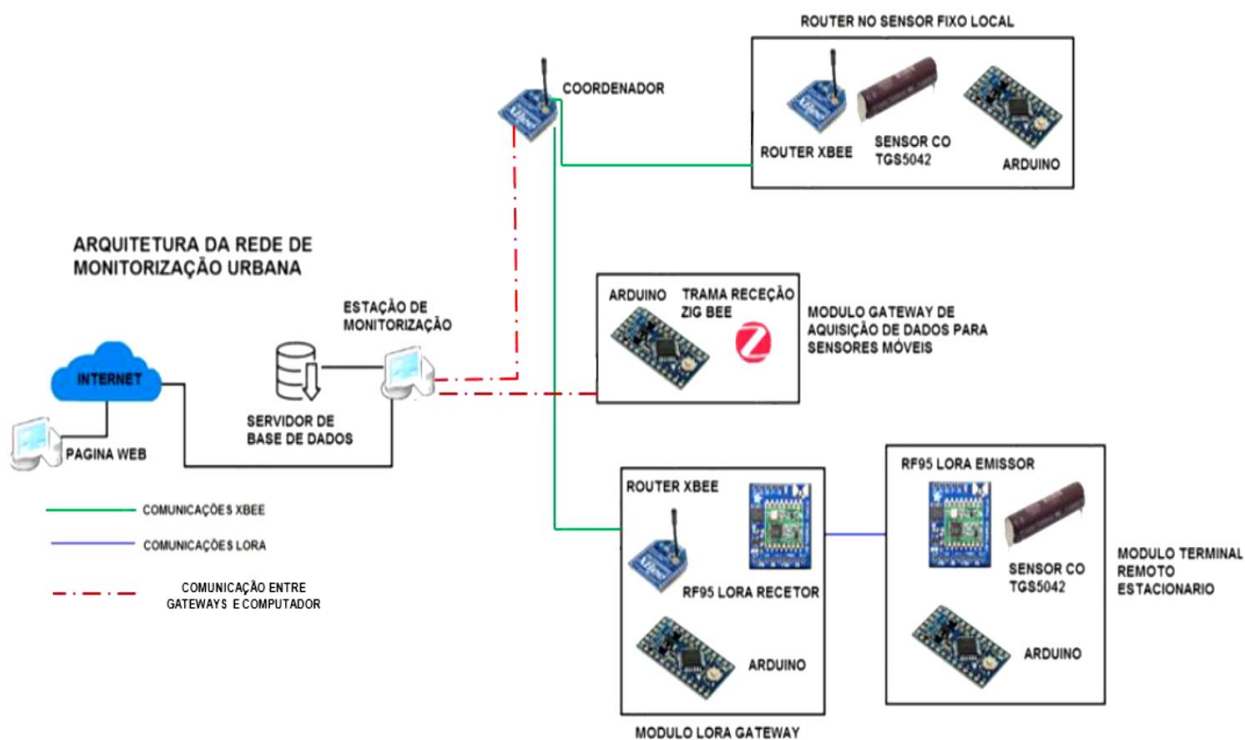


Figura 3.1 Arquitetura da rede de monitorização urbana.

DESENVOLVIMENTO

Tabela 3.1 Descrição dos protótipos a desenvolver.

Protótipos	Fornecimento de energia	Microcontrolador	Sensor	Parâmetros monitorizados	Armazenamento	Comunicação
Nó de calibração	Baterias de lítio; Circuito de carregamento de baterias de lítio; Interruptores;	Arduino Pro-Mini ATMega 328 P	Alphasense B4	Nível das baterias; Sinais de tensão dos eléctrodos sensor CO; Níveis de CO;	Cartão microSD	-
Nó sensor local	Painel fotovoltaico; Díodo zener de potência; Bateria de lítio; Interruptor;		Figaro TGS 5042	Nível da bateria; Sinal de tensão sensor CO; Níveis de CO;	Servidor de base de dados	Zigbee
Gateway de aquisição de dados	FTDI conectado ao computador por USB		Dados temporais; Coordenadas geográficas; Sinal de tensão sensor CO; Nível da bateria; Níveis de CO;	Servidor de base de dados	UART	
Terminal remoto	Painel fotovoltaico; Circuito de controlo de carga; Bateria de lítio; Interruptor;		Nível da bateria; Sinal de tensão sensor CO;	-	LoRa	
Gateway LoRa	Painéis fotovoltaicos conectados em paralelo; Circuito de controlo de carga; Baterias de lítio;		Dados de entrada – terminal remoto; Nível das baterias – gateway; Níveis de CO;	Servidor de base de dados	LoRa Zigbee	

O nó de calibração é um dispositivo de pequenas dimensões portátil, o qual é alimentado por baterias de lítio e serve-se de um circuito de carregamento para as mesmas. Este serve-se de um sensor de CO Alphasense B4 cujas características de resolução e gama dinâmica tornam-lhe adequado para medir concentrações de CO mais baixas. Os seus dados são armazenados num cartão microSD. Os protótipos a desenvolver posteriormente servem-se do sensor Figaro TGS 5042, dado que este é um sensor de menor custo comparativamente ao Alphasense B4. O controlo embebido dos nós desenvolvidos é realizado através do microcontrolador Arduino Pro-Mini ATMega 328P.

O nó de monitorização local é fixo, como tal necessita de um painel fotovoltaico como fonte de captação de energia. O nó local realiza monitorização ambiental periódica. Este serve-se da tecnologia rádio Zigbee, o nó contém um módulo RF XBee configurado como Router Zigbee. Este comunica com um Coordenador, o qual atua como um *gateway* para a recolha de dados de todo o sistema.

DESENVOLVIMENTO

O *gateway* de aquisição de dados destina-se a recolher informação previamente monitorizada por nós móveis. Os dados de entrada são lidos de um cartão microSD e são submetidos a diversas operações de tratamento de dados. Os dados de saída são encapsulados numa trama do protocolo Zigbee, estes são transmitidos por comunicação UART. O dispositivo necessita de ser conectado a um computador.

O sistema de monitorização remota faz uso da tecnologia rádio de longo alcance LoRa (*Long Range*). É necessário desenvolver dois nós os quais são um terminal remoto e um *gateway* LoRa. Há que referir que ambos são nós fixos, como tal, são necessários painéis fotovoltaicos como fontes de captação de energia. O terminal remoto à semelhança do nó local realiza monitorização ambiental periódica, este transmite os dados servindo-se da tecnologia rádio LoRa. A função do *gateway* LoRa destina-se a integrar o terminal remoto na rede de sensores da Universidade. Para tal, este realiza o tratamento dos dados monitorizados pelo terminal remoto e serve-se das tecnologias rádio LoRa e Zigbee.

A principal tecnologia rádio a utilizar na rede de monitorização urbana é Zigbee. Os nós desenvolvidos passam a integrar a rede de sensores da Universidade, a qual baseia-se numa topologia em malha. Cada estação de monitorização da rede serve-se de um Router Zigbee. As mensagens devem ser transmitidas no formato de emissão do protocolo Zigbee. O Coordenador comanda a hierarquia, este coordena a rede de sensores e atua como *gateway* permitindo deste modo a interligação entre a rede de sensores e a Internet. Este encontra-se conectado por USB à porta de comunicação do computador do laboratório. A figura 3.2 apresenta a arquitetura para a rede de sensores da Universidade.

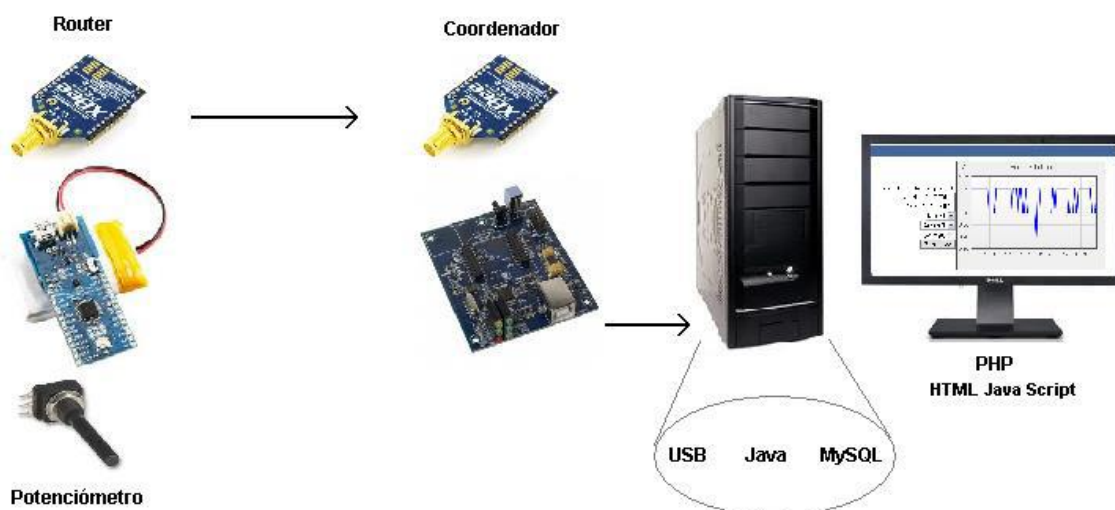


Figura 3.2 Arquitetura de rede do sistema da rede de sensores [38].

Os dados recebidos pelo Coordenador são recolhidos por uma aplicação Java, esta mesma aplicação é também necessária para o gateway de aquisição de dados. Esta é executada no *software* NetBeans [39], a qual interage com o Coordenador através de comunicação série. Deste modo, os dados são lidos da porta série à qual o Coordenador está conectado. A aplicação Java está estruturada para a leitura de mensagens no formato do protocolo Zigbee.

DESENVOLVIMENTO

Após a recolha dos dados, esta aplicação realiza o seu tratamento e insere os mesmos numa base de dados MySQL. Os dados são disponibilizados *online* na plataforma *web* da rede de sensores da Universidade [40]. Há que referir a importância da ferramenta Xampp, a qual é responsável por habilitar o acesso ao servidor MySQL através das portas de comunicação para o Apache e MySQL. Outro parâmetro importante é a configuração de rede do PAN ID, este é o identificador utilizado pelos Routers e o Coordenador Zigbee da rede de sensores, o qual é apresentado na tabela 3.2. Esta tabela ainda apresenta os ficheiros de código Java de maior importância para o armazenamento de dados no servidor MySQL, assim como, a ferramenta Xampp.

Tabela 3.2 Acesso ao servidor de base de dados.

Acesso ao servidor de base de dados	
MySQL.java	Parâmetros de acesso ao servidor: url, user, pass, dbName
Xampp	Ativação portas comunicação: Apache, MySQL
PAN ID Rede Zigbee	99
msgCare_base.java	Realiza separação de dados – nome, valor Filtra pacotes Zigbee de receção (0x90) no formato #valor1valor2
Msg_collector.java	Captura mensagens provenientes da porta série e armazena temporariamente num buffer.
Msg_parser.java	Lê cada mensagem do buffer e realiza o seu processamento. Tal é feito identificando o primeiro carácter do payload. O interpretador da aplicação Java deve reconhecer o carácter de interesse. Em tal caso, a mensagem é válida sendo inserida na base de dados.

3.3. Módulo de calibração

O primeiro dispositivo desenvolvido foi um nó de calibração. Este realiza monitorizações de monóxido de carbono. O dispositivo tem pequenas dimensões, é leve, compacto e portátil.

3.3.1. Diagrama funcional

A figura 3.3 ilustra o respetivo diagrama funcional.

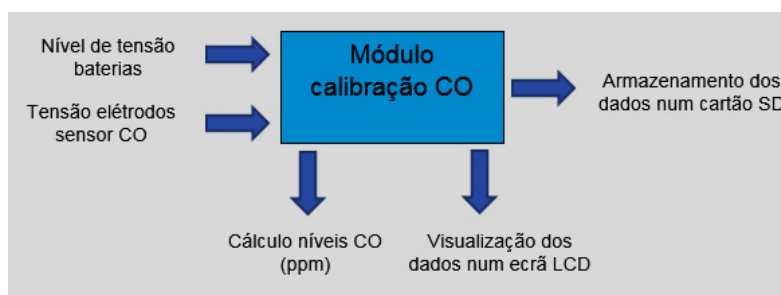


Figura 3.3 Diagrama funcional nó de calibração.

O nó está equipado com um sensor de CO de maior precisão e com baixa afetação a gases interferentes, tendo um custo mais elevado. O fabricante fornece o respetivo circuito de condicionamento de sinal e o sensor vem calibrado de fábrica. Tal lhe permite efetuar medições com maior grau de precisão. Estes factores foram determinantes na sua escolha para utilizar no módulo de calibração. Os dados de entrada são os níveis de tensão das baterias utilizadas e os sinais de tensão lidos dos elétrodos do sensor de CO. As operações de processamento incluem determinar as concentrações de CO em ppm e apresentar os dados de saída num ecrã LCD. Os dados são armazenados num cartão microSD.

3.3.2. Microcontrolador

Para o controlo embebido utilizou-se um Arduino Pro-Mini [41], o qual está ilustrado na figura 3.4 a). A sua placa de desenvolvimento tem dimensões compactas quando comparado com outras versões da família Arduino. É uma característica vantajosa quando utilizado para plataformas com pequenas dimensões.

Este é um microcontrolador de baixo custo. A sua linguagem de programação é relativamente simples. A programação é realizada no IDE Arduino utilizando linguagem da família C. A sua capacidade de processamento é adequada para aplicações de sensores de baixo débito. Este microcontrolador é eficiente em termos energéticos, o que permite a sua utilização em aplicações de baixo consumo energético. Este opera na tensão nominal de 3,3 V e numa frequência de 8 MHz.

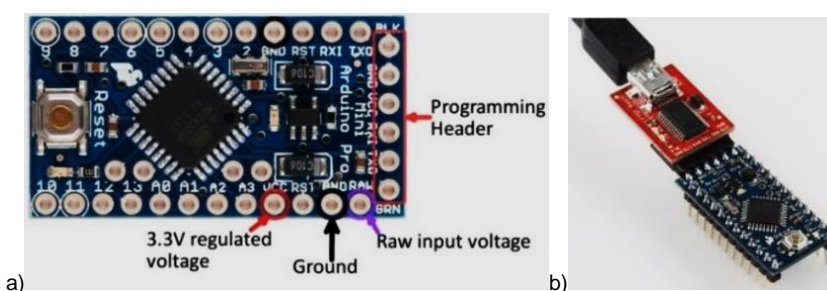


Figura 3.4 a) Microcontrolador Arduino Pro-Mini [41]. b) Utilização do programador conversor série-USB (FTDI) [41].

Quando utilizado *hardware* cuja tensão de operação seja superior a 3,3 V são necessários circuitos adicionais para regulação da tensão. O carregamento de um determinado programa para a placa do microcontrolador implica o uso do programador, o qual é ilustrado na figura 3.4 b). Este é um conversor série-USB, também denominado FTDI (*Future Technology Devices International*). A alimentação é realizada no pino RAW. A gama de tensão de entrada aceite é desde os 3,4 até aos 12 V. A placa de desenvolvimento contém um circuito de regulação de tensão interno, o qual realiza a conversão para a tensão nominal. O pino de tensão nominal é o VCC. O microcontrolador têm 8 ADC (A0...A7) e 13 pinos de entradas/saídas digitais (D0...D13). Entre estes últimos é de realçar os pinos do protocolo SPI (*Serial Peripheral Interface*), os pinos D10 a D13. O pinout do microcontrolador é referenciado no Anexo F. A tabela 3.3 ilustra as suas especificações técnicas.

Tabela 3.3 Especificações Arduino Pro-Mini [41].

Especificações Arduino Pro-Mini						
Microcontrolador	Tensão nominal	Corrente máxima por pino	Memória Flash	EEPROM	RAM	Frequência
ATMega328p	3,3 V	40 mA	32 kB	1 kB	2 kB	8 MHz

3.3.3. Sensor CO Alphasense B4

O Alphasense B4 é um sensor eletroquímico, com características amperimétricas. Este mede concentrações de monóxido de carbono. O sensor funciona do seguinte modo: na presença do gás de interesse na superfície do sensor é aplicado um potencial elétrico, o que gera uma corrente que induz uma reação química. Servindo-se desta reação eletroquímica o sensor mede as concentrações de CO.

DESENVOLVIMENTO

É necessário realçar que a intensidade do sinal de saída é proporcional às concentrações gasosas presentes na superfície do sensor. Como ilustrado na figura 3.5, este possui dois elérodos. Estes são o elédrodo de trabalho (*working electrode*) e o elédrodo auxiliar (*auxiliar electrode*).

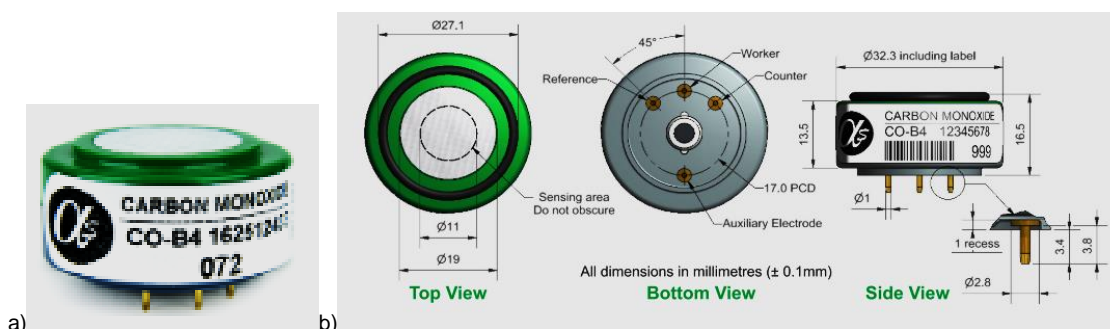


Figura 3.5 a) Sensor CO Alphasense B4 [2]. b) Respetivo diagrama esquemático [2].

A tabela 3.4 apresenta as características de desempenho do sensor de CO. A tabela 3.5 caracteriza a sua sensibilidade a outros gases interferentes. Estes dados foram consultados da respetiva folha de características [2].

Tabela 3.4 Características de desempenho do sensor Alphasense [2].

Sensibilidade nA/ppm (a 2 ppm de CO)	Tempo de resposta t90 (de 0 a 10 ppm)	Gama de medições (ppm)	Gama de temperatura (°C)	Peso (g)
420 a 650	<25	1000	-30 a 50	<13

Tabela 3.5 Sensibilidade do sensor a vários gases interferentes [2].

Gás interferente	Concentrações (ppm)	Sensibilidade nA/ppm
H ₂ S	5	<1
NO ₂	5	<1
Cl ₂	5	<1
NO	5	<-3
SO ₂	5	<0,1
H ₂	100	<10
C ₂ H ₄	100	<1
NH ₃	20	<0,1

O sensor serve-se de um filtro eletroquímico, cuja função é eliminar interferências causadas por outros gases, o que minimiza as perturbações das medições. É necessário referir que a exposição prolongada a tais gases tem como consequência a redução da eficiência do filtro ao longo do tempo. É documentado pelo fabricante que a capacidade do filtro é 250000 ppm horas [2]. O fabricante refere que é possível realizar medições em ambientes com ruído com uma resolução até 4 ppb. Tal se deve ao circuito de condicionamento de sinal utilizado. Este é disponibilizado pela Alphasense, o qual é denominado ISB (*Individual Sensor Board*) [42].

DESENVOLVIMENTO

A ISB contém a eletrônica adequada para aplicações de baixo ruído. A sua função consiste em extrair o respetivo sinal elétrico para ser lido pelo microcontrolador. Adicionalmente permite medir o efeito da oxidação que resulta da presença do gás de interesse e reduzir a interferência dos outros gases. A figura 3.6 ilustra a ISB e o respetivo *pinout* [42].

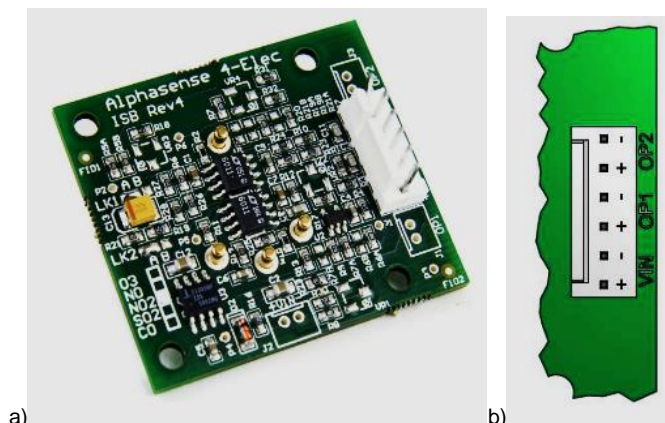


Figura 3.6 a) Circuito de condicionamento Individual Sensor Board (ISB) [42]. b) Pinagem de conexão à placa [42].

O pinout considera a alimentação e os elérodos. O sensor pode ser alimentado com uma tensão DC entre os 3,5 e os 6,4 V. Optou-se por alimentar a 6 V. Dado o facto de o microcontrolador operar a 3,3 V, isto implicou utilizar um regulador de tensão [43], cuja tensão de entrada seria fornecida diretamente da bateria principal. Ambos os elérodos do sensor fornecem em mV os respetivos sinais de saída. O microcontrolador serve-se de dois ADC de modo a ler os sinais. A eletrônica utilizada na extração do sinal de tensão de cada elétrodo é semelhante. Esta consiste em dois andares de amplificação e um buffer. Os parâmetros DC *offset* de tensão e o ganho são valores fixos. Os valores DC *offset* caracterizam a tensão de *offset* para cada elétrodo. Estes dados estão impressos no invólucro do sensor juntamente com a ISB correspondente. Os dados estão descritos na tabela 3.6 para o sensor utilizado [2].

Tabela 3.6 Parâmetros desvio offset DC de cada elétrodo e sensibilidade fornecidos pelo fabricante [2].

Offset DC elétrodo trabalho WE_{V_o} (mV)	Offset DC elétrodo auxiliar AUX_{V_o} (mV)	Sensibilidade (mV/ppm)
492	334	405

As concentrações de CO em p-p-m são determinadas por [2],

$$CO(ppm) = \frac{(WE_{OP1} - WE_{V_o}) - (AUX_{OP2} - AUX_{V_o})}{S} \quad (3.1)$$

em que WE_{OP1} e AUX_{OP2} caracterizam as tensões dos elérodos de trabalho e auxiliar em mV, respetivamente. Os parâmetros WE_{V_o} e AUX_{V_o} são as tensões de *offset* DC para cada elétrodo. Por fim S caracteriza a sensibilidade do sensor.

3.3.4. Ecrã LCD TFT Arduino

Os dados de saída são apresentados num ecrã. Este é um LCD TFT cujo fabricante é o Arduino. Este tem uma resolução de 160x128 pixéis e uma dimensão de 1,77'' [44], o qual é ilustrado na figura 3.7. O LCD TFT opera a 5 V DC e o microcontrolador a 3,3 V, sendo necessário um regulador de tensão [45].

DESENVOLVIMENTO

A tensão de entrada é fornecida pela bateria secundária, a qual é dedicada para alimentar o ecrã LCD. Este ecrã LCD interage com o microcontrolador através do protocolo SPI. Para tal, serve-se de uma biblioteca de Arduino desenvolvida para os LCD da família TFT. O LCD TFT tem acoplado um slot de cartão microSD. Este pode ser utilizado recorrendo à biblioteca SD.

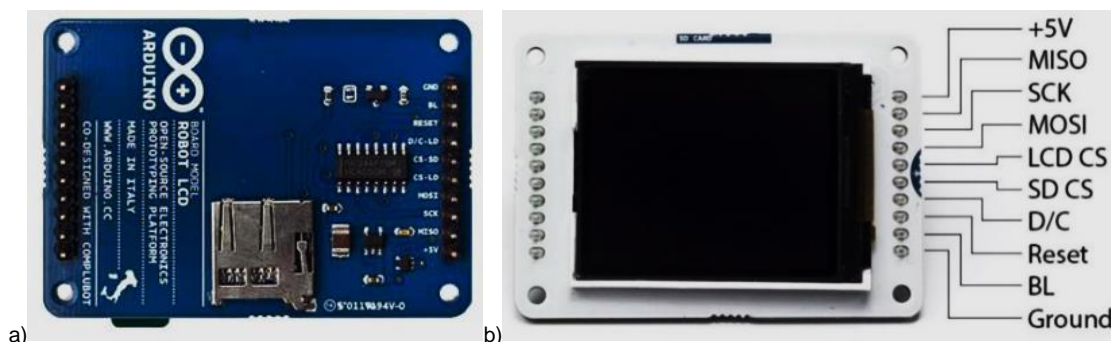


Figura 3.7 Ecrã LCD TFT Arduino [44].

3.3.5. Consumos de energia

O nó de calibração serve-se de um circuito de carregamento de baterias de lítio. O módulo inclui um circuito de controlo de carga (TP4056) e circuitos de proteção contra sobrecorrente [46]. É possível controlar a corrente de carregamento. Tal é feito através do ajuste de uma resistência externa na placa (R_{PROG}). A corrente máxima que é possível fornecer é 1 A. O nó de calibração é alimentado por duas baterias de lítio recarregáveis [47]. A sua tensão nominal é 3,7 V. A tensão máxima de carga é cerca de 4.2 V. Uma bateria alimenta o microcontrolador e o sensor de CO e tem uma capacidade de 2500 mAh. A outra alimenta o ecrã LCD e a sua capacidade é 2200 mAh. A tabela 3.7 apresenta os consumos energéticos do nó de calibração. De acordo com os resultados da tabela foi possível estimar que ambas as baterias teriam uma autonomia de aproximadamente 6 dias.

Tabela 3.7 Consumos energéticos nó de calibração.

Baterias	Capacidade das baterias (mAh)	Consumo médio de corrente (mA)	Autonomia (dias)
BAT1	2500	17,23	6
BAT2	2200	15,36	6

3.3.6. Arquitetura

A figura 3.8 apresenta a arquitetura interna do nó de calibração. O seletor de carga permite escolher a bateria a carregar. Os dois interruptores permitem ligar o circuito de medição e o ecrã. Os divisores resistivos possibilitam adequar os valores de tensão de modo a não ultrapassar a tensão máxima dos ADC do microcontrolador. Foi necessário realizar o *design* do *layout* PCB referente ao *hardware*, tendo-se utilizado o *software* da National Instruments Ultiboard [48], e desenvolvida uma placa de circuito impresso com o sistema. O circuito eletrónico do nó foi inserido numa caixa elétrica com as características de proteção IP69, de modo a resguardar a eletrónica do contacto exterior. O esquemático do circuito, o *design* do *layout* PCB, a placa de circuito impresso e o respetivo módulo resultante são apresentados no Anexo H.

DESENVOLVIMENTO

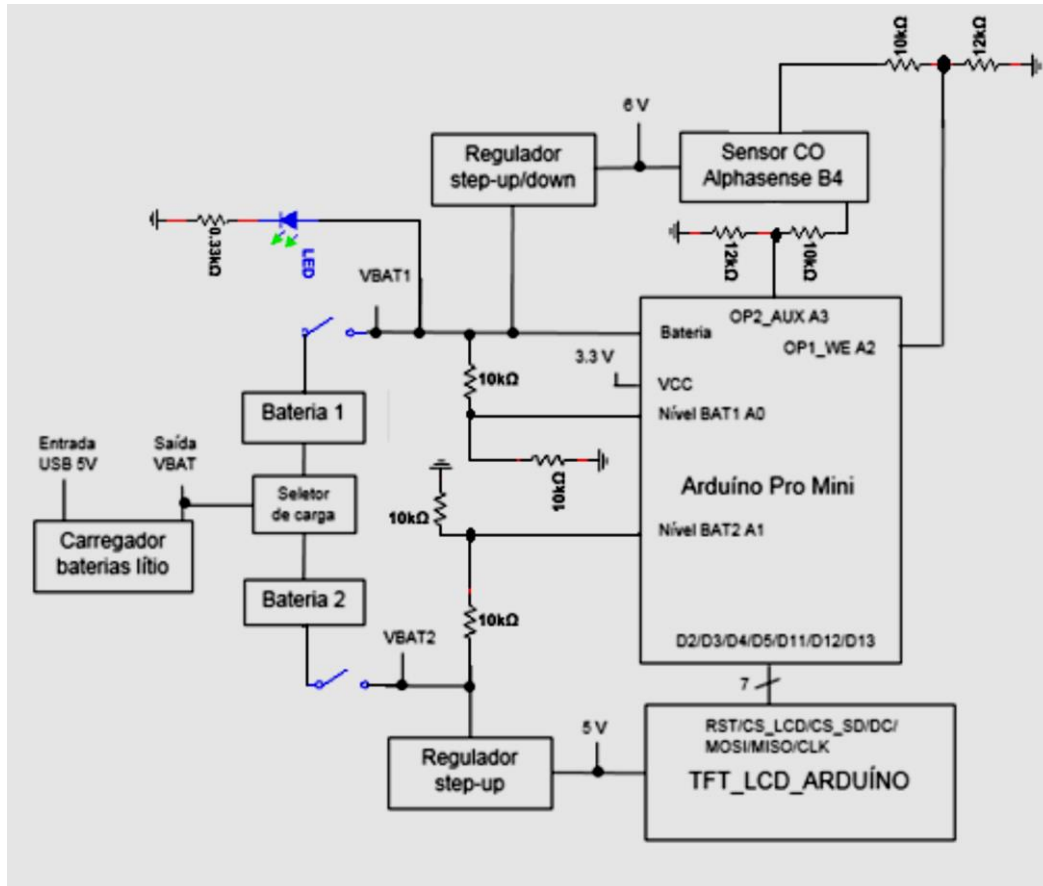


Figura 3.8 Arquitetura interna nó de calibração.

3.3.7. Controlo embebido

A figura 3.9 ilustra o fluxograma que caracteriza o controlo embebido implementado.

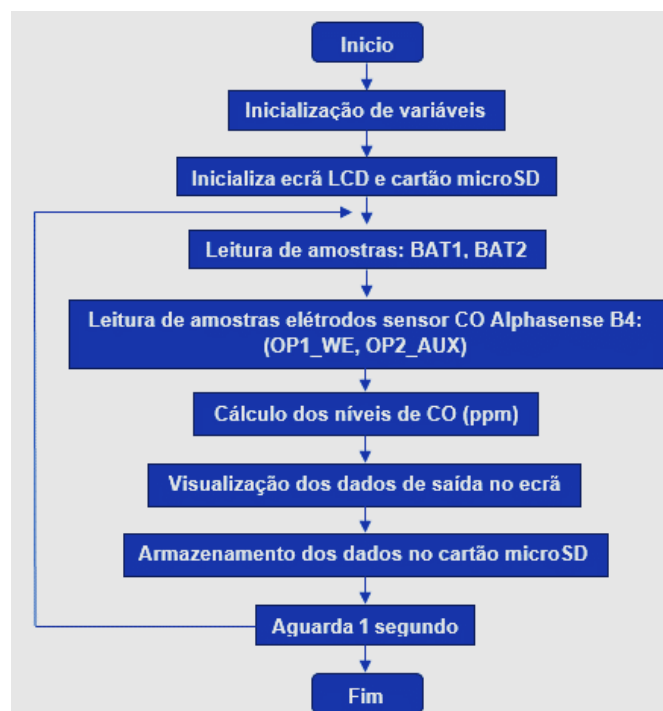


Figura 3.9 Fluxograma que caracteriza controlo embebido do nó de calibração.

As primeiras ações envolvem a inicialização de variáveis, a inicialização do ecrã LCD e do cartão microSD. Após isto, o sistema passa a efetuar uma sequência de ações iterativas. Esta sequência de ações é realizada uma vez por segundo. Primeiro, leem-se as amostras correspondentes aos níveis de tensão das baterias. Após isso, são lidas as amostras dos sinais de tensão dos elétrodos do sensor de CO. Tendo o conhecimento dos sinais de tensão dos elétrodos, é possível determinar as concentrações de CO em ppm. De seguida, os dados de saída são carregados para o ecrã LCD de modo a visualizar os mesmos. Após isto, estes dados são armazenados no cartão microSD. O código desenvolvido para o controlo embebido é apresentado no Anexo H.

3.4. Estação de monitorização local

Parte do trabalho desenvolvido no projeto envolveu construir estações de monitorização fixas. Esta seção descreve o nó sensor local, o qual foi instalado num ponto local na Universidade. Utilizou-se a tecnologia rádio Zigbee dada a compatibilidade com a rede de sensores existente. O nó local de monitorização é fixo. Pretende-se que este seja autónomo energeticamente, o que requer políticas de eficiência energética. Para tal definiu-se que este nó realizasse leitura de amostras, processamento e transmissão dos dados de saída de uma forma periódica. Durante a maior parte do tempo o nó encontra-se no estado adormecido. No ciclo adormecido o microcontrolador e o módulo de comunicação estão inativos. Foi utilizado um painel fotovoltaico para captação de energia.

3.4.1. Diagrama funcional

Na figura 3.10 está ilustrado o seu diagrama funcional. As amostras a ler pelo nó local são os níveis da bateria e o sinal de tensão do sensor de CO. São determinadas as concentrações de monóxido de carbono, os dados são transmitidos servindo-se de um módulo rádio XBee.

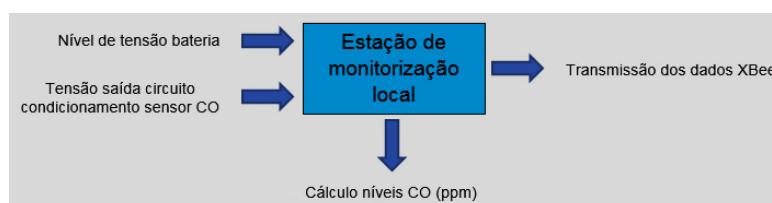


Figura 3.10 Diagrama funcional para o nó local.

3.4.2. Alimentação

De modo a assegurar autonomia energética utilizou-se um painel fotovoltaico de 500 mW, com uma tensão em circuito aberto de 4,6 V e uma corrente de curto-circuito de 160 mA [49]. Utilizou-se uma bateria de lítio recarregável de 2500 mAh. Houve necessidade de utilizar um díodo zener de potência à saída do painel fotovoltaico. Este semiconductor encontra-se em série à saída do painel [50]. Na folha de características é referido que este semiconductor pode suportar até 5 W. A tensão de zener pode variar entre os 3,3 e os 200 V. A finalidade era regular a tensão de saída do painel e minimizar eventuais correntes de retorno. Tal ocorre quando o painel fornece potência a uma carga. É uma solução que reduz perdas reativas causadas pelas correntes reversas, o que possibilita uma transferência de potência mais eficiente no carregamento da bateria e preserva o painel fotovoltaico.

3.4.3. Sensor CO TGS 5042

O sensor utilizado nas medições de monóxido de carbono é o TGS 5042, da Figaro [51], este é ilustrado na figura 3.11, o qual é um sensor eletroquímico. Há uma relação diretamente proporcional entre as concentrações de CO presentes na superfície do sensor e a intensidade do sinal medido. Existem diversos tipos de aplicações para este sensor, entre as quais detetores comerciais ou residenciais, monitorização de CO para aplicações industriais, controlo de ventilação para garagens de estacionamento, deteção de incêndios. A tabela 3.8 ilustra algumas características elétricas [51].

Tabela 3.8 Características elétricas sensor CO TGS 5042 [51].

Gama de deteção (ppm)	Sensibilidade (nA/ppm)	Gama de temperatura (°C)	Gama de humidade relativa (%)	Tempo de resposta (T90) s	Peso (g)
0-10000	1,2 até 2,4	-40 a 70	5 a 99	≈ 60	12

A sensibilidade do sensor tem um determinado grau de influência devido à temperatura, o que é descrito na folha de características [51]. O parâmetro sensibilidade é fornecido pelo fabricante, estando impresso no invólucro do sensor como ilustrado na figura 3.11, o que simplifica para os utilizadores o processo de calibração. O parâmetro de sensibilidade é caracterizado por um código “xxxx”, sendo x.xxx nA/ppm. O fabricante determina a sensibilidade. Este insere o sensor num ambiente com concentrações a aproximadamente 300 ppm de CO. O seu sinal de saída é medido nestas condições. O número de referência do sensor é impresso à esquerda dos dados de sensibilidade no formato YYMMDD. O Figaro TGS 5042 possui dois elétrodos, sendo o de trabalho (*working electrode*) e o auxiliar (*counter electrode*). Como o sinal de saída é um sinal de corrente na ordem dos nA, é necessário um circuito de condicionamento de sinal [52].

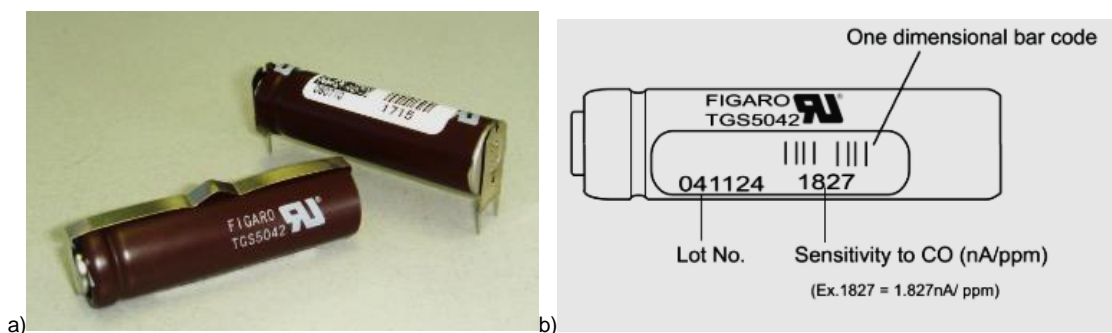


Figura 3.11 a) Sensor CO TGS 5042 [51]. b) Impressão sensibilidade e número de referência [51].

O fabricante recomenda cuidados na realização das conexões, tal deve-se à polaridade do sensor, a qual é oposta à de uma bateria. A corrente flui orientada do elétrodo de trabalho para o auxiliar. Outro cuidado a considerar é a tensão aplicada ao terminal de saída do sensor. Esta deve ser limitada até aos 10 mV de modo a evitar danificar este dispositivo. Outra particularidade, é quando este sensor é armazenado sem haver conexão entre os elétrodos vai ocorrer efeito de polarização. Em tal caso, quando conectado novamente a um circuito elétrico é necessário um período considerável para o sinal de saída estabilizar. O efeito de polarização pode ser evitado curto-circuitando os elétrodos. Alternativamente, pode ser conectada uma resistência $\leq 1 \text{ k}\Omega$ entre os elétrodos.

3.4.4. Circuito de condicionamento de sinal TGS 5042

A figura 3.12 ilustra o circuito de condicionamento de sinal necessário para o sensor de CO TGS 5042, o qual é alimentado a 3,3 V [52]. Este recebe o sinal do sensor e vai convertê-lo num sinal de tensão. Este é amplificado para um nível de tensão que o ADC do microcontrolador seja capaz de ler.

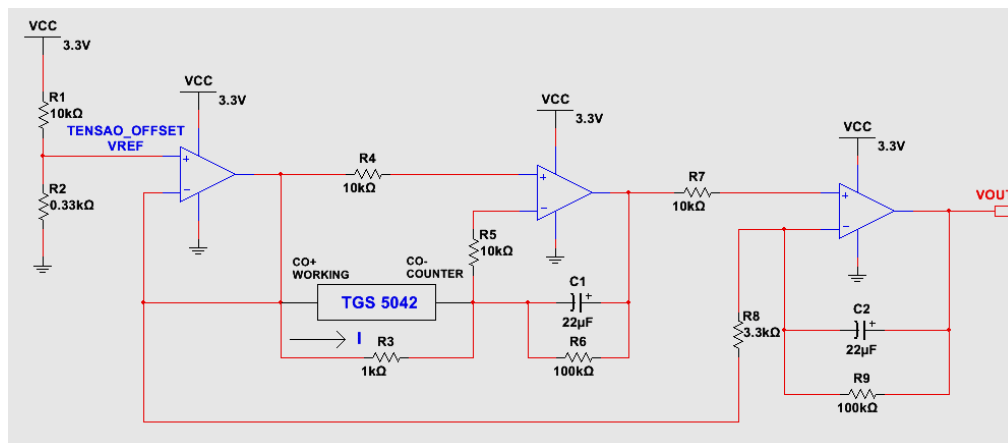


Figura 3.12 Circuito de condicionamento para o sensor CO TGS 5042 [52].

O circuito inicia-se com um divisor resistivo, a partir do qual é obtido o sinal de tensão *offset* (TENSÃO_OFFSET), o qual encontra-se em torno dos 95 mV. Este caracteriza um sinal de referência para um ambiente livre de concentrações de CO. O primeiro andar de amplificação é um *buffer*, os sinais de entrada são o sinal *offset* (TENSÃO_OFFSET) e o sinal do elétrodo de trabalho (CO+ WORKING).

O segundo e o terceiro andar de amplificação caracterizam um amplificador de transimpedância. Este circuito de amplificação destina-se a converter o sinal de corrente do sensor para um sinal de tensão. Deste modo, o sensor de CO deve ser conectada entre o *buffer* e o amplificador de transimpedância. Também é conectada uma resistência de 1 kΩ em paralelo aos elétrodos do sensor de CO, cuja função é prevenir o efeito de polarização do sensor e manter a tensão de saída do circuito estável. À saída do último andar de amplificação é extraído o sinal de tensão de saída (VOUT), o qual é lido pelo ADC do microcontrolador.

O Arduíno possui ADC com 10 bits de resolução. Em condições gerais de uso, a tensão de referência está configurada para 3,3 V, o que permite uma resolução de 3,23 mV. No entanto, pretendia-se otimizar a capacidade de leitura do microcontrolador para as menores concentrações de CO, o que poderia ser realizado tirando o melhor proveito dos recursos de resolução dos ADC. Assim sendo, optou-se por ajustar a tensão de referência dos ADC para 1,1V. O sinal da tensão de referência é gerado internamente no microcontrolador. Esta configuração permitiu otimizar a resolução dos ADC para os 1,08 mV, a qual realizou-se utilizando a função `analogReference(INTERNAL)` [53]. Deste modo, a concentração mais baixa lida pelo microcontrolador foi cerca de 0,21 ppm, sendo este o valor numérico em ppm mais baixo que o ADC é capaz de ler. Utilizou-se um sensor de CO com sensibilidade de 1,527 nA/ppm. A expressão para determinar as concentrações de monóxido de carbono (CO) em ppm é dada por [52],

$$CO(ppm) = ((VOUT - TENSÃO_OFFSET)/A)/at \quad (3.2)$$

com $TENSAO_OFFSET$ a tensão de saída do sensor de CO em condições ambientais específicas. Estas são um ambiente livre de CO, uma temperatura ambiente de $20^{\circ}C \pm 2^{\circ}C$ e uma humidade relativa de $40 \pm 10\%$. O parâmetro A caracteriza o ganho. Este parâmetro é obtido através de medições experimentais. O valor de ganho considerado foi 3,13. O valor foi baseado num projeto anterior de mestrado [54]. O parâmetro αt caracteriza a sensibilidade impressa no invólucro do sensor em nA/ppm. As concentrações de monóxido de carbono (CO) podem ser expressas nas unidades mg/m³ dadas por [55],

$$\text{Concentrações CO (mg/m}^3\text{)} = 0,0409 * CO(\text{ppm}) * M \quad (3.3)$$

no qual o parâmetro M caracteriza a massa molar para o monóxido de carbono. Esta é 28,01 g/mol. A montagem do circuito da figura 3.12 impunha algumas exigências, devido a problemas associados a ruído térmico, os recursos de sensibilidade dos sensores Figaro e o sinal de saída dos elétrodos ser da ordem dos nA, o que motivou a utilização de *hardware* da tecnologia SMD (*Surface-Mount Device*). Ainda se utilizou um amplificador de baixo ruído. Este é o LTC 6079, da Linear Technology [56]. O fabricante refere que o dispositivo é capaz de operar com correntes até 1 pA a uma temperatura ambiente de 25°C. O *design PCB layout* é apresentado no Anexo I. Este circuito foi impresso e montado numa placa de circuito impresso.

3.4.5. Transferência de dados

O equipamento rádio utilizado nas comunicações foi o módulo RF XBee, da Digi [57]. Estes rádios podem utilizar os protocolos de comunicação sem fios Zigbee ou IEEE 802.15.4 e operam nos 2,4 GHz da banda ISM. A figura 3.13 ilustra módulos RF XBee com diferentes tipos de conectores e antenas [58]. Há quatro tipos de dispositivos rádio com diferentes opções de antenas. Estas são as configurações antena Wire, conector U.FL, conector RPSMA e antena chip. Os XBee são módulos rádio de baixo custo, pequenas dimensões e simples utilização. Estes são categorizados em série 1 e 2. Cada série possui XBee regulares e XBee Pro. Os XBee Pro destacam-se por possuírem maior alcance e potência de saída, tal proporciona comunicações a maiores distâncias.

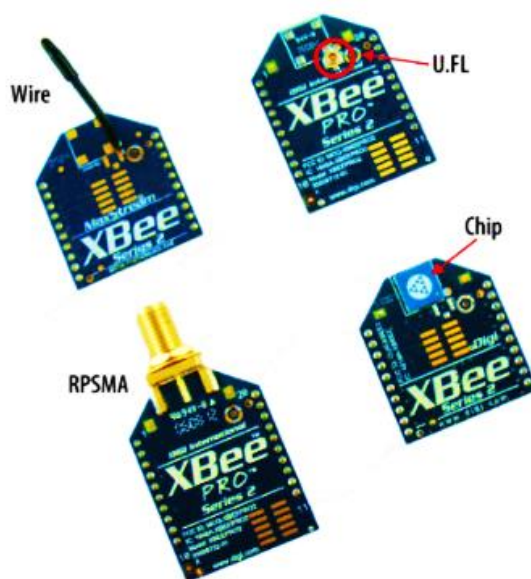


Figura 3.13 Opções de antenas utilizadas nos módulos RF XBee [58].

DESENVOLVIMENTO

A tabela 3.9 ilustra características elétricas e de desempenho [58].

Tabela 3.9 Comparação de características elétricas para os XBee série 1 e 2 [58].

Caraterísticas elétricas XBee	Serie 1 XBee / XBee Pro	Serie 2 XBee
Tensão alimentação	2,8 até 3,4 VDC	2,1 até 3,6 VDC
Corrente de transmissão	45 mA / 215 mA	35 mA (40 mA com boost mode)
Corrente na receção	50 mA/ 55 mA	38 mA (40 mA com boost mode)
Corrente modo adormecido	< 10 μ A	< 1 μ A (a 25°C)
Alcance ambiente interior/urbano	30 m / 90 m	40 m
Alcance ambiente exterior em linha de vista	90 m / 1.6 km	120 m
Potência de saída transmitida	1 mW (0 dBm) / 60 mW (+18 dBm)	1,25 mW (+1 dBm)/ 2 mW (+3 dBm) com boost mode
Sensibilidade do recetor a 1% PER (Packet Error Rate)	- 92 dBm / -100 dBm	- 97 dBm / -98 dBm (com boost mode)
Taxa de dados por comunicação série	1200 bps até 250 kbps	1200 bps até 1 Mbps
ADC	6 ADC 10 bits resolução	4 ADC 10 bits resolução
Número de pinos digitais (I/O)	8	10
Firmware utilizado no XCTU	802.15.4	Zig Bee

Os XBee possuem um *pinout* comum entre as várias versões, no entanto, as configurações internas ao nível do *firmware* diferem. O *firmware* permite realizar configurações de rede ao nível da camada MAC. As configurações são realizadas no *software* XCTU [59]. O módulo RF e o microcontrolador comunicam através de uma interface série. É opcional recorrer a dispositivos que realizam conversões lógicas para dispositivos série. É o caso do RS-232 ou interfaces USB. No caso de dispositivos com interface UART, as conexões físicas são realizadas como ilustrado na figura 3.14.

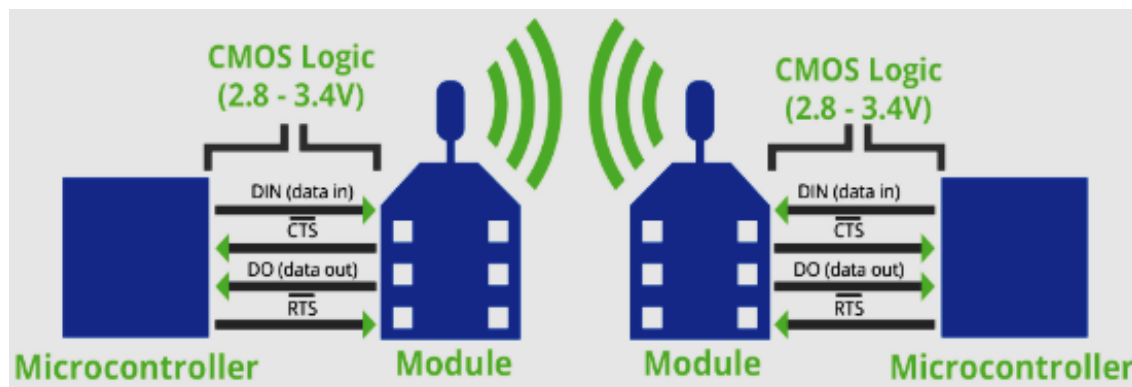


Figura 3.14 Comunicação UART entre módulo XBee e microcontrolador. Dispositivos com interface UART integrada. À esquerda caso emissor, à direita caso recetor [57].

Há dois modos de operação para o XBee. Estes são os modos AT (*Transparent mode*) e API (*Application Programming Interface*). No modo AT as transmissões ocorrem de forma transparente, ou seja, os dados são transmitidos numa sequência de bytes pré-definida. Já o modo API utiliza tramas para encapsular os dados a transmitir. No que se refere à imunidade a interferências espectrais são utilizados canais com modulação de espalhamento espectral DSSS. No Anexo G é apresentado o *pinout* para um XBee assim como a descrição dos pinos, é ainda apresentado o formato estrutural das tramas API [60]. A tabela 3.10 ilustra as configurações realizadas no módulo rádio. Utilizou-se um módulo RF XBee Series 2, o qual se configurou como Router Zigbee. As configurações realizaram-se no *software* Digi XCTU.

DESENVOLVIMENTO

Tabela 3.10 Configurações realizadas no módulo de comunicação XBee Series 2.

Configurações Router XBee Series 2	
Versão firmware	ZIGBEE ROUTER API
Endereço Router XBee	0013A200 4094451F
PAN ID	99
Scan Channels	0xFFFF
Endereço Destinatário 64 bits	0x000000000000FFFF
Endereço Destinatário 16 bits	0xFFFE
Nome	No Router Fixo
Número máximo saltos (NH)	1E
Configuração modo adormecido (SM)	Pin Hibernate (Pino DTR)
Duração do ciclo adormecido/ Cyclic sleep period x 10 ms (SP)	3E8 (10 segundos)

A tabela 3.11 ilustra o formato da trama do protocolo Zigbee utilizada na transmissão dos dados do nó local. Daqui se realçam três fatores. O primeiro é o campo referente ao identificador de API que identifica a mensagem como sendo uma trama de emissão (0x10). O segundo é o campo referente ao *checksum*. A sua função é verificar a integridade dos dados. O terceiro é o *payload* da mensagem, o qual engloba os dados de entrada lidos dos ADC e os dados de CO.

Tabela 3.11 Constituição da trama de emissão Zig Bee.

Constituição da trama Zig Bee	
Byte de início	0x7E
Comprimento da mensagem	50 bytes: 0x00 0x32
Identificador de API	Formato trama emissão: 0x10
Numerador da mensagem	0x00
Endereço geral Coordenador 64 bits	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Endereço Coordenador 16 bits	0xFF 0xFE
Campo número de saltos	0x00
Tipo de mensagem	Mensagem broadcast: 0x00
Conteúdo da mensagem	?VBATV=valor&COTGSppm=valor&VOMV=valor
Byte de checksum	0x00

3.4.6. Inserção dos dados na base de dados

Esta seção descreve o processo de inserção das mensagens na base de dados. Foi utilizada a ferramenta Xampp [61] e a aplicação Java executada no NetBeans [39]. Do lado da aplicação Java há 4 códigos fonte particularmente importantes a referir. Estes são o “MyQSL.java”, o “msgCare_base.java”, o “Msg_collector.java” e “Msg_parser.java”.

O código “MyQSL.java” contém uma classe que implementa o acesso à base de dados, a qual serve-se dos parâmetros url, user, pass, dbName. O parâmetro url fornece o IP do DNS “apoio.cee.uma.pt”. Este é executado no computador do laboratório de telecomunicações que recebe os dados do Coordenador e realiza a execução da aplicação Java para inserir os dados no servidor. O dbName identifica a base de dados que guarda os dados monitorizados pelo nó local. O nome da base de dados atribuído foi “CO Sensor local”. Este nome é identificado no servidor Web para fins de consulta dos dados de interesse.

DESENVOLVIMENTO

O código “msgCare_base.java” executa uma função responsável por separar os sensores e os respetivos valores de dados, esta utiliza o formato #valor1valor2. São atribuídos nomes aos sensores de forma a identificá-los, as mensagens de dados tratadas são do tipo 0x90, o que caracteriza pacotes de receção Zigbee. Para o nó local foram considerados três sensores, os respetivos campos de dados destinavam-se aos dados do nível da bateria (VBATV), o sinal de tensão lido do sensor de CO (VomV) e os dados das concentrações de CO (COTGSppm). Inicialmente, os dados de entrada são guardados em variáveis temporárias com o formato de dados apropriado. Posteriormente, os mesmos são armazenados numa matriz de dados com duas colunas, a qual é designada “dataMatrix”. Esta tem um formato de dados *string* e vai armazenar o nome de cada sensor e respetivos dados de interesse. Numa fase intermédia, o formato dos dados de entrada deve ser convertido em *string*. A figura 3.15 apresenta o script do “msgCare_base.java”.

```
int numberOfSensors = 3; // número de sensores
String [][] dataMatrix = new String[numberOfSensors][2];
// Matriz de dados com duas colunas: Nome do sensor, dados de interesse
// dados de entrada são guardados em variáveis temporárias
float VBATV=(((data[1]&0xFF)<<8)+(data[2]&0xFF)); // VBATV
int VomV=(((data[3]&0xFF)<<8)+(data[4]&0xFF)); // VomV
float COTGSppm=(((data[5]&0xFF)<<8)+(data[6]&0xFF)); // COTGSppm
// Armazenamento dos dados na matriz
// os dados com os valores de interesse são convertidos para o formato string
dataMatrix[0][0] = " VBATV "; dataMatrix[0][1] = Float.toString(VBATV); // VBATV
dataMatrix[1][0] = "VomV"; dataMatrix[1][1] = Integer.toString(VomV); // VomV
dataMatrix[2][0] = "COTGSppm"; dataMatrix[2][1] = Float.toString(COTGSppm); // COTGSppm
return dataMatrix;
```

Figura 3.15 Script do código fonte “msgCare_base.java” utilizado para o nó local.

O código “Msg_collector.java” é responsável por capturar as mensagens da porta série e depositá-las num *buffer* de dados. As mensagens provêm do Coordenador Zigbee da rede local, o qual está conectado ao computador do laboratório de Telecomunicações que executa a aplicação Java. O código “Msg_parser.java” lê cada mensagem do *buffer* e realiza o seu processamento. Uma particularidade a considerar é o formato da mensagem recebida, neste caso, o primeiro carater do payload é “?”. Este carater identificador permite ao interpretador do código “Msg_parser.java” reconhecer a mensagem Zigbee como sendo válida, o que permite filtrar e inserir a mensagem na base de dados. A figura 3.16 ilustra este processo.

```
private String[][] msgInterpreter(msgXBee m){
    switch(m.get_apild()){ // Escolhe o interpretador consoante o XBee API ID
        case 0x90: // Mensagem de dados
            switch(m.get_msgType()){ // Escolhe o interpretador consoante o carater que identifica o formato dos dados
                case '?': // Mensagem do tipo: "?Sensor1=valor1&Sensor2=valor2" ...
                    return msgCare_data.msgInterpreter(m.get_data());
            }
    }
}
```

Figura 3.16 Segmento de código que ilustra o processo de interpretação da mensagem Zigbee no código fonte “Msg_parser.java”.

DESENVOLVIMENTO

O Xampp executa as portas de comunicação do Apache e MySQL, o que ativa o acesso ao servidor MySQL. No endereço “C:\xampp\htdocs\wsn_web” está localizado um ficheiro com o nome “dbinfo.PHP”. Este ficheiro também armazena os parâmetros referenciados para o “MySQL.java”. Teve-se em consideração na atribuição do nome da base de dados que o parâmetro dbName necessitaria de ter um nome igual atribuído. Deste modo, atribuiu-se o nome “CO Sensor local” no referido campo de ambos os ficheiros “MySQL.java” e “dbinfo.PHP”, deste modo, os dados seriam descarregados na base de dados.

3.4.7. Arquitetura

A figura 3.17 apresenta a arquitetura do nó sensor local. Realizou-se o *design* do *layout* PCB do *hardware* referente a este sistema e procedeu-se à impressão da placa de circuito impresso, respetiva montagem e testes. O *hardware* foi resguardado numa caixa elétrica com as características IP69. Houve necessidade de construir uma estrutura anti-rradiação para o sensor de CO, o qual se destinava a proteger o sensor da radiação solar e outras interferências externas que pudessem comprometer as leituras das medições. A caixa elétrica e a estrutura anti-radiação são anexadas a uma estrutura principal em acrílico. No topo da estrutura instalou-se o painel fotovoltaico referido na seção 3.4.2. O esquemático deste sistema, o *layout* PCB, a placa de circuito impresso e respetiva estrutura física do nó são apresentados no Anexo I.

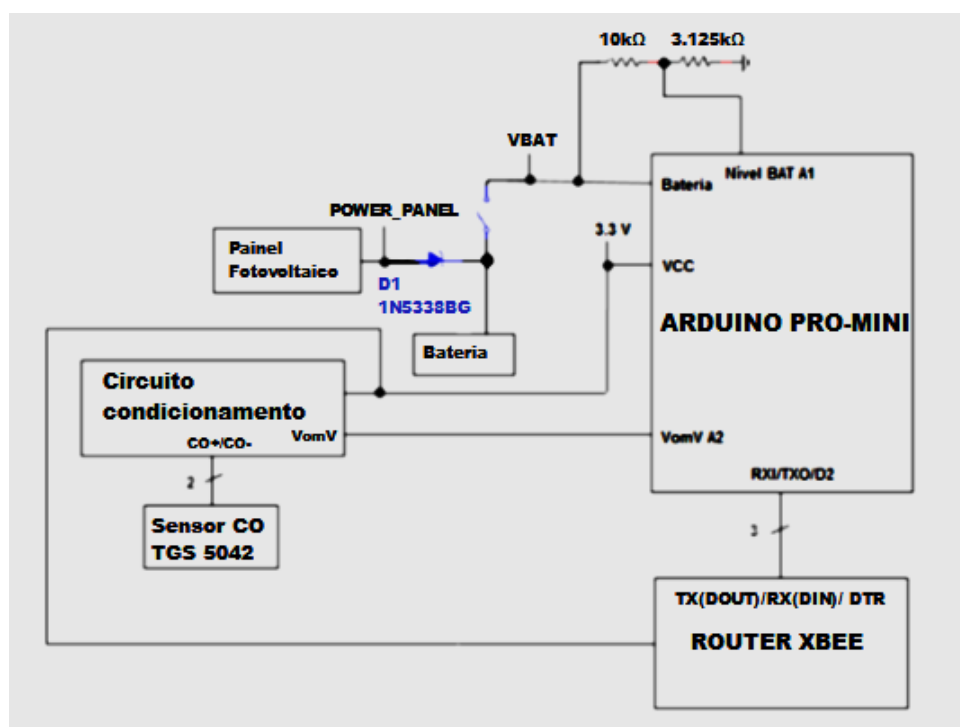


Figura 3.17 Arquitetura interna que caracteriza o nó local.

3.4.8. Controlo embebido

A figura 3.18 ilustra o fluxograma que caracteriza o controlo embebido do nó sensor local. O código desenvolvido para o controlo embebido é apresentado no Anexo I. Após a inicialização de variáveis é realizado o ajuste da tensão de referência dos ADC para os 1,1 V, e que foi aplicado o processo descrito na seção 3.4.4.

DESENVOLVIMENTO

O controlo embebido da estação de monitorização possui ciclos ativos/adormecidos, uma vez que o objetivo era o nó local ser autónomo energeticamente. Numa fase inicial são configurados os pinos I/O utilizados. Foi necessário configurar um *timer* para atuar como contador. Os dados de entrada são o nível de tensão da bateria (BAT A1) e o sinal de tensão do sensor de CO (VomV A2). São obtidas as concentrações de CO, de seguida, os dados são inseridos na trama do protocolo Zigbee para transmissão.

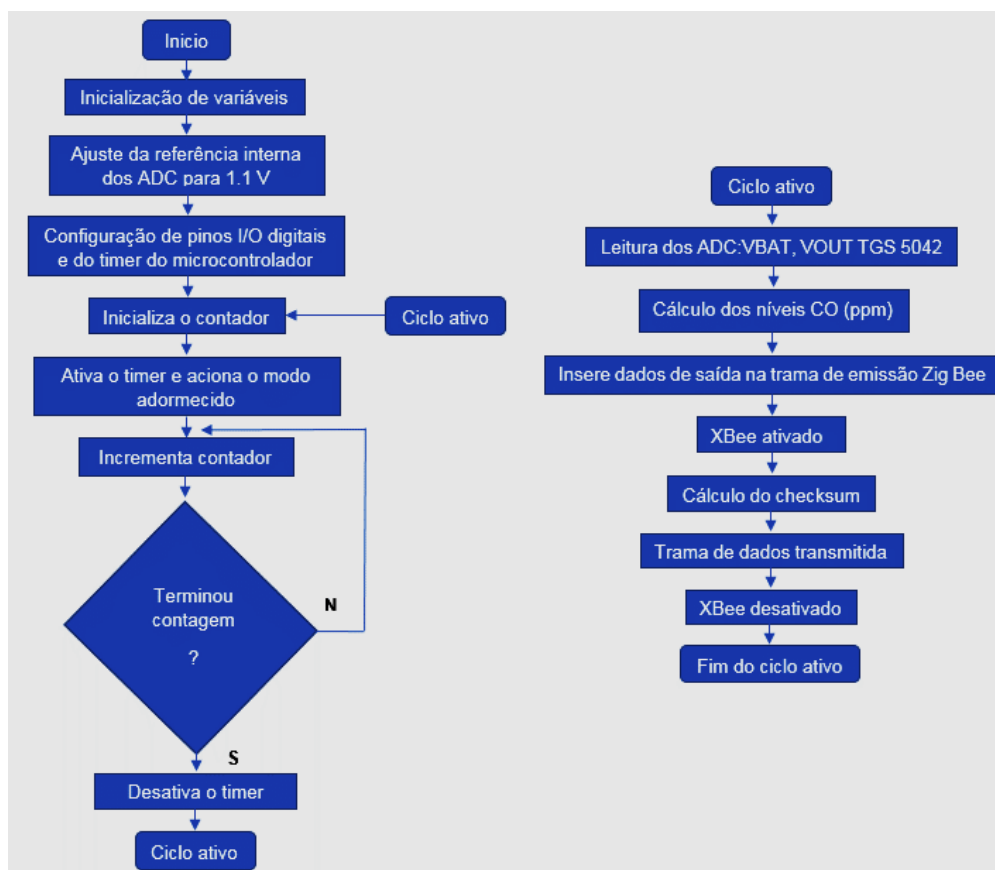


Figura 3.18 Fluxograma que caracteriza o controlo embebido do nó sensor local.

3.5. Gateway de aquisição de dados

Esta seção descreve o nó *gateway* para aquisição de dados das estações móveis. O nó móvel possui um recetor GPS e um sensor de monóxido de carbono. O sensor de CO é um Figaro TGS 5042.

3.5.1. Diagrama funcional

A figura 3.19 ilustra o diagrama funcional do nó *gateway* de aquisição de dados. A mensagem recolhida de um cartão SD no formato \$DDMMYY,HHMMSS,LAT,LONG,VomV,BAT. A mensagem recolhida inicia-se com os dados temporais. Os quais se encontram no formato DDMMYYHHMMSS. Seguem-se os campos referentes às coordenadas geográficas recebidas pelo recetor GPS, as quais são latitude e longitude. Os restantes dados tratam-se do sinal de tensão do sensor de CO e o nível da bateria que alimenta o nó móvel.

DESENVOLVIMENTO

Os dados da mensagem são guardados num *buffer* temporariamente. Esta pode ou não conter dados de GPS. O que depende do recetor GPS do nó móvel. Tal afeta a dimensão da mensagem. O microcontrolador realiza o tratamento de dados necessário atendendo a estes fatores. Os dados de saída são inseridos na trama de receção do protocolo Zigbee. Os quais são transmitidos por comunicação série. O payload da trama é transmitido no seguinte formato:

***ts=DDMMYYHHMMSS&LAT=DADOS&LONG=DADOS&CO=DADOS**



Figura 3.19 Diagrama funcional módulo gateway de aquisição de dados.

O módulo leitor de cartões MicroSD está ilustrado na figura 3.20. A tabela 3.12 ilustra o *pinout*. O fabricante é a LC Technology [62]. O dispositivo opera numa tensão nominal de 3,3 V. O módulo comunica com o microcontrolador através do protocolo SPI. Este pode servir-se da biblioteca open source do Arduino SD.h.

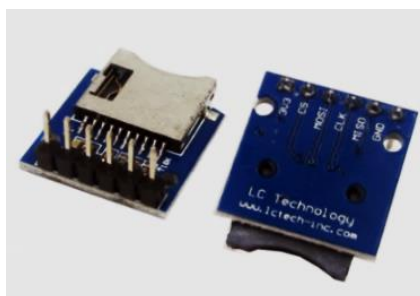


Figura 3.20 Módulo leitor de cartão MicroSD LC Technology [62].

Tabela 3.12 *Pinout* módulo leitor de cartão MicroSD [62].

Pino	Função
VCC	Alimentação 3,3 V
GND	GND Potência/Lógico
CS	Chip Select SPI
MOSI	Master Out Slave In SPI
MISO	Master In Slave Out SPI
CLK	Clock SPI

3.5.2. Arquitetura

A figura 3.21 ilustra a arquitetura do *gateway* de aquisição de dados. O *hardware* deste sistema engloba o microcontrolador, o programador FTDI e o leitor de cartão microSD. O programador FTDI utilizado nesta aplicação têm três funções. A primeira é o carregamento do código para o microcontrolador. A segunda é a comunicação série dos dados de saída da *gateway*. A última é o fornecimento de energia a este módulo, pois optou-se por não utilizar uma bateria dedicada, uma vez que o dispositivo apenas tem utilidade prática quando conectado a um computador.

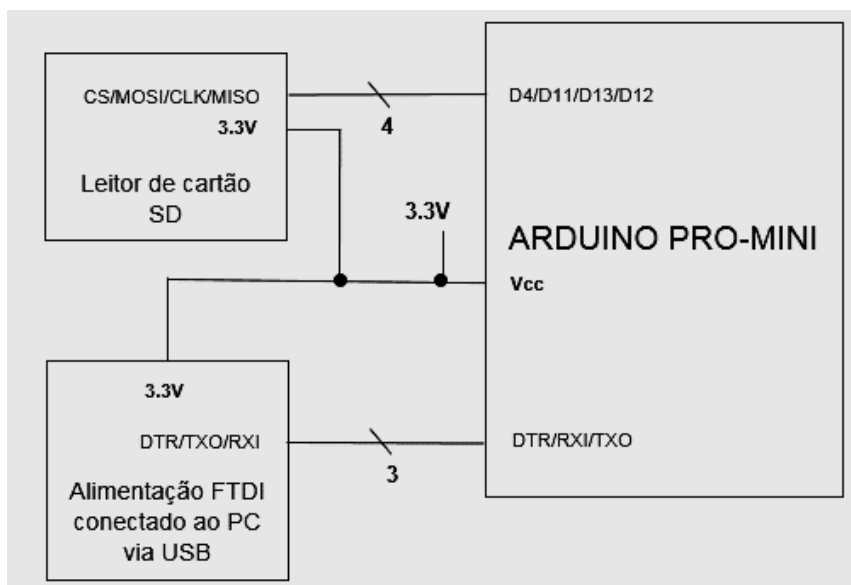


Figura 3.21 Arquitetura interna do nó gateway de aquisição de dados.

3.5.3. Controlo embebido

A figura 3.22 apresenta o fluxograma que caracteriza a execução do código principal. O controlo embebido é apresentado no Anexo J. As primeiras ações no programa envolvem a inicialização de variáveis e do cartão MicroSD. Após a fase de inicializações, o programa executado pelo microcontrolador procura um ficheiro com o nome FICHHD.txt. Se não for encontrado o programa reporta que o ficheiro não existe e termina, caso contrário o ficheiro é aberto e lido. A leitura é ao nível do *byte*. À medida que os dados são lidos estes são armazenados num buffer temporariamente. Quando a leitura for concluída o programa encerra o ficheiro e termina. Caso contrário prossegue a sequência de ações a descrever nesta seção.

Os dados lidos do ficheiro estão estruturados em linhas no formato \$DDMMYY,HHMMSS,LAT,LONG,VomV,BAT, como ilustrado na figura 3.19. Cada mensagem é finalizada pelo carácter terminador e então inicia-se uma nova mensagem. Dado o facto de a leitura ser ao nível do *byte* é necessário um método de distinção para as linhas. Para este fim definiu-se uma condição adicional no programa a qual consistia na deteção do carácter terminador.

Assim que o *byte* terminador é detetado tal significa que se leu uma mensagem completa. Aí, o programa executa um algoritmo de decisão ilustrado na figura 3.23, o qual é importante para as tarefas de tratamento de dados posteriores. Após a execução dessas tarefas, o programa retorna ao código principal. Então, o índice do *buffer* é reiniciado ficando disponível para a leitura de uma nova mensagem. Os dados de saída são transmitidos no formato de protocolo Zigbee, servindo-se da comunicação UART. Segue-se um período de espera ativa de 100 ms e então é repetido o processo de leitura de uma nova mensagem.

O algoritmo de decisão destina-se a efetuar a distinção de mensagens que podem eventualmente conter coordenadas GPS. Isto tem afetação no comprimento em *bytes* da mensagem, podendo assumir duas dimensões à partida conhecidas. Assim, a decisão a tomar depende do comprimento do *buffer*, a sua dimensão é conhecida através da leitura do índice.

DESENVOLVIMENTO

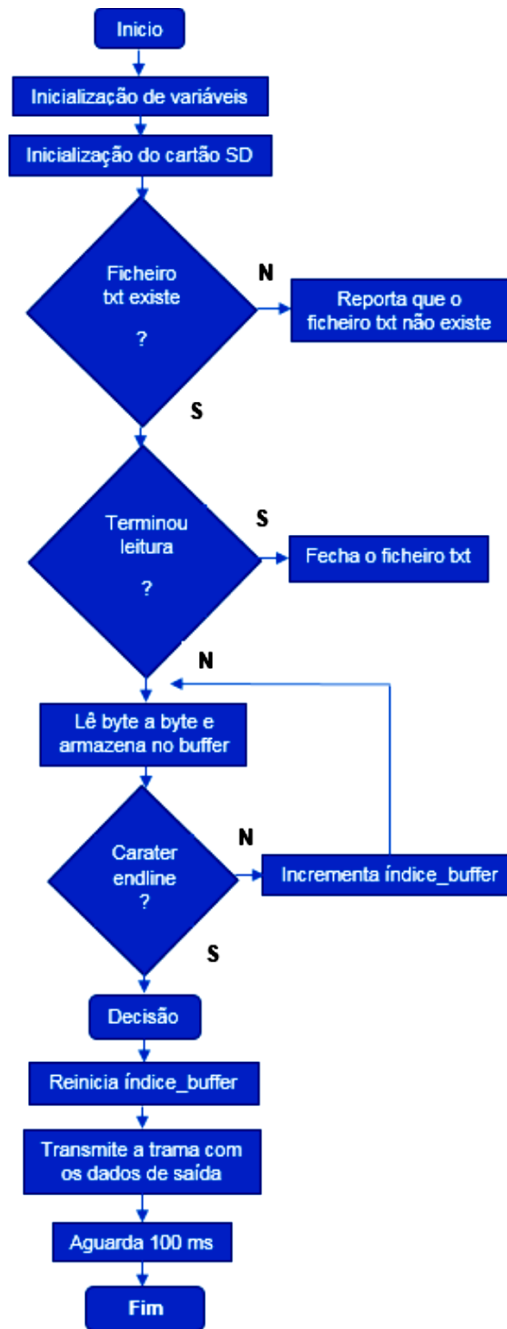


Figura 3.22 Fluxograma código principal.

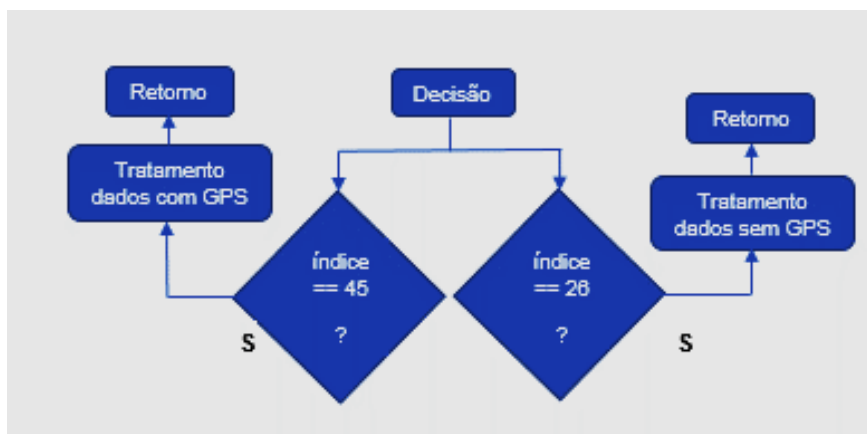


Figura 3.23 Algoritmo de decisão de tratamento de dados com ou sem coordenadas geográficas.

DESENVOLVIMENTO

A figura 3.24 apresenta os algoritmos de tratamento de dados. À esquerda, o caso da mensagem com dados de GPS. À direita, a situação sem coordenadas geográficas. O processo inicial é idêntico, este trata-se da leitura dos dados temporais, que são lidos do *buffer* no formato DDMMYYHHMMSS e inseridos na trama Zigbee.

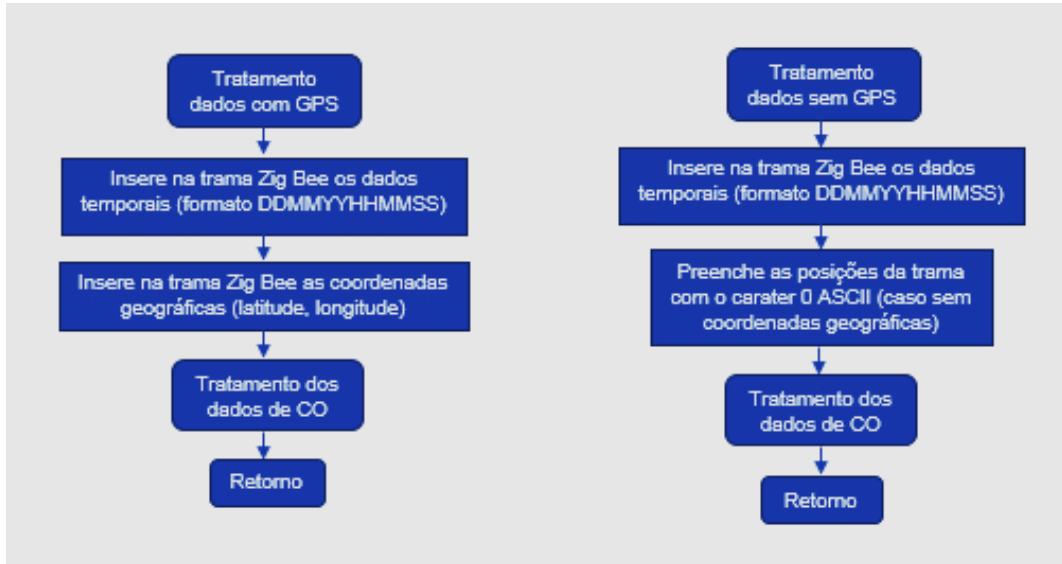


Figura 3.24 Algoritmos para o tratamento de dados considerando a tomada de decisão.

O que difere é a eventual existência de coordenadas GPS, por sua vez, influencia a posição dos dados de interesse para o cálculo das concentrações de CO. Se houverem coordenadas geográficas, estas são lidas das posições do *buffer* e inseridas na trama Zigbee. Caso contrário, os campos da trama Zigbee destinados aos dados de GPS, são preenchidos com o carácter 0 em ASCII. Por fim, é realizado o tratamento de dados de CO, o qual se apresenta na figura 3.25.

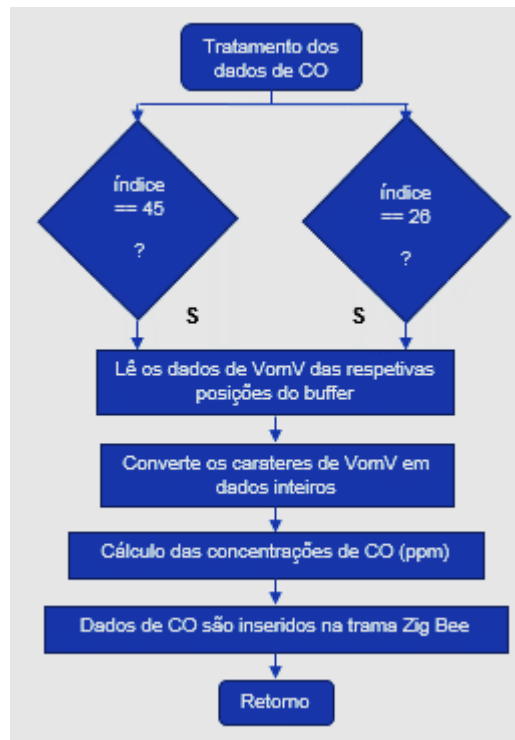


Figura 3.25 Algoritmo para o tratamento de dados de CO.

DESENVOLVIMENTO

O algoritmo de tratamento dos dados de CO considera ambos os casos acima descritos. Novamente, a dimensão do *buffer* é crucial para determinar as posições que guardam os dados de interesse, estes são, os dados do sinal de tensão lido do sensor de CO. Foi necessário converter estes dados para o formato inteiro, posteriormente, são determinadas as concentrações de CO. Então, estes dados são inseridos na trama Zigbee, em seguida, o programa retorna ao código principal. A tabela 3.13 ilustra o formato da trama do protocolo Zigbee. Três particularidades de relevância. A primeira o identificador de API, o qual foi configurado no formato 0x90 que caracteriza a trama Zigbee no formato de recepção. Segundo, o identificador de 64 bits do nó. Terceiro, o primeiro caráter do payload, este permite reconhecer o formato da mensagem.

Tabela 3.13 Constituição da trama de recepção Zigbee.

Constituição da trama Zigbee	
Byte de início	0x7E
Comprimento da mensagem	67 bytes: 0x00 0x43
Identificador de API	Formato trama recepção: 0x90
Identificador de 64 bits da gateway	0x3C,0x2B,0x4A,0x45,0x08,0x0A,0x05,0x01
Endereço Coordenador 16 bits	0xFF 0xFE
Campo número de saltos	0x00
Tipo de mensagem	Mensagem broadcast: 0x00
Conteúdo da mensagem	*ts=DDMMYYHHMMSS&LAT=DADOS&LONG=DADOS&CO=DADOS
Byte de checksum	0x00

3.5.4. Inserção dos dados na base

Tirou-se proveito da base de dados criada anteriormente para o nó local. Deste modo, os dados são apresentados na página Web da rede de sensores na base “CO Sensor local”. A figura 3.26 apresenta parte do script do código “Msg_parser.java”.

```
private String[][] msgInterpreter(msgXBee m){
    switch(m.get_apild()){ // Escolhe o interpretador consoante o XBee API ID
        case 0x90: // Mensagem de dados
            switch(m.get_msgType()){ //Escolhe o interpretador consoante o caracter que identifica o
                formato dos dados
                // Mensagem do tipo: "*ts=170203145001&Sensor1=valor1&Sensor2=valor2" ...
            case '*':
                return msgCare_data.msgInterpreter(m.get_data());
            }
    }
```

Figura 3.26 Segmento de código que ilustra o processo de interpretação da mensagem Zigbee no código fonte “Msg_parser.java”.

É necessário conectar o nó *gateway* a um computador, para tal, utiliza-se o programador FTDI. Através do código “Msg_collector.java”, os dados são recolhidos e armazenados temporariamente num *buffer*. No código “MyQSL.java”, para o campo url atribuiu-se o endereço IPv4 do computador ao qual o nó *gateway* foi conectado, isto permitiria o acesso ao servidor através do referido computador. O código “Msg_parser.java” é responsável por inserir os dados na base, tal é feito através da identificação do formato da mensagem. Neste caso, o primeiro caráter do *payload* a identificar pelo interpretador Java deverá ser “*”. No caso afirmativo, a mensagem Zigbee é reconhecida como válida e inserida no servidor de base de dados.

3.6. Estação de monitorização remota

Esta seção descreve o sistema de monitorização remoto. Tal como o nó local, este sistema monitoriza concentrações de CO. No entanto, a monitorização realiza-se em pontos mais afastados geograficamente, tal implica utilizar a tecnologia rádio LoRa. Estes rádios conseguem comunicar a distâncias da ordem de alguns km. A arquitetura de rede para uma aplicação de sensores baseada em LoRa necessita de dois nós, os quais são um terminal remoto e um *gateway*. O terminal remoto realiza monitorização periódica no ambiente de medição. A função do *gateway* é interligar o nó remoto e o servidor de base de dados, para tal este nó serve-se das tecnologias rádio LoRa e Zigbee.

3.6.1. Diagrama funcional

A figura 3.27 apresenta o diagrama funcional do terminal remoto, este é um nó fixo, tal implica políticas de baixo consumo. É de realçar que foi necessário atender a regulamentações de *duty cycle* para a banda ISM dos 870 MHz. O ciclo ativo destina-se à leitura de amostras, processamento e transmissão de dados. No restante período, o microcontrolador e o rádio estão inativos. Os dados de entrada a monitorizar são o nível da bateria e o sinal elétrico extraído do circuito para o sensor de CO. Os dados de saída são transmitidos pelo rádio LoRa.

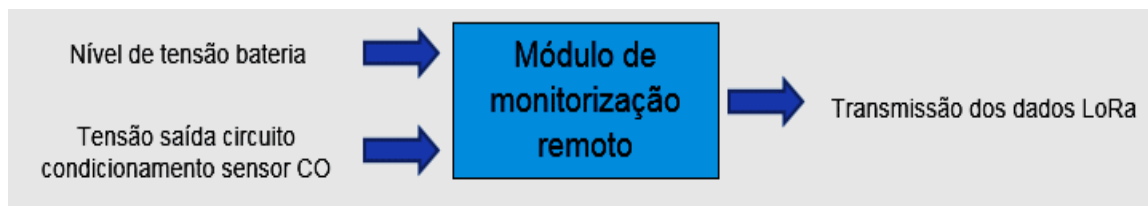


Figura 3.27 Diagrama funcional nó remoto de monitorização.

3.6.2. Alimentação

O sistema de alimentação é composto por um painel fotovoltaico, um controlador de carga e uma bateria de lítio. A função do painel fotovoltaico é manter a bateria carregada, a qual tem uma capacidade de 2500 mAh. Fez-se uso de um painel de 2 W, com tensão em circuito aberto de 6 V e corrente de curto-circuito de 300 mA [63]. As suas dimensões físicas são 120 x 120 x 2.8 mm, o qual é ilustrado na figura 3.28.

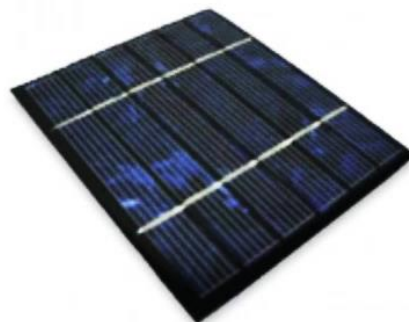


Figura 3.28 Painel fotovoltaico 2 W e 6 V [63].

Para o carregamento da bateria de lítio, utilizou-se o controlador de carga MCP73871 da Adafruit [64], ilustrado na figura 3.29, este aceita tensões de entrada DC desde 5 a 6 V.

DESENVOLVIMENTO

O controlador de carga possui um mecanismo que automaticamente tira proveito da potência de entrada disponível, de modo, a evitar que a bateria entre num ciclo contínuo de carga/descarga. O mecanismo atua quando os níveis de tensão da bateria estão aproximadamente em 4,2 V. O fabricante originalmente define que o controlador de carga está configurado para fornecer correntes até 500 mA. No entanto, existe a opção de ajustar a corrente de carregamento ($I_{CHGRATE}$) desde os 50 mA até 1 A. O ajuste pode ser efetuado soldando uma resistência externa na placa, a qual é soldada aos terminais da resistência R_{PROG} , modificando assim o seu valor resistivo e a corrente de carregamento. Esta resistência é um componente SMD existente na placa, cujo valor de origem é definido pelo fabricante. R_{PROG} permite o controlo da corrente de carregamento dada pelo controlador de carga de acordo com a seguinte relação:

$$I_{CHGRATE} = \frac{1000}{R_{PROG}} \quad (3.4)$$

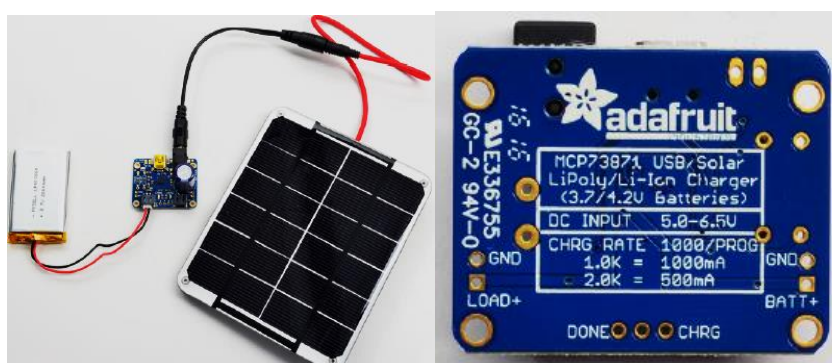


Figura 3.29 Controlador de carga MCP73871 Adafruit [64].

3.6.3. Módulo RF LoRa

Fez-se uso da tecnologia LoRa nas comunicações rádio do nó remoto para garantir longo alcance [65]. Estes operam nas bandas ISM dos 433 e os 870 MHz. Os módulos utilizados operam nos 870 MHz. Os fabricantes fornecem estes rádios contendo um conjunto de configurações pré-definidas, tais como o formato dos pacotes LoRaWAN e mecanismos de retransmissão. Isto simplifica o processo de transmissão, pois basta preocupar-se com o *payload*. Após a transmissão do pacote LoRaWAN, os módulos em escuta do canal irão receber a mensagem. Tal implica, que esses rádios recetores estejam configurados na mesma frequência que o emissor e que possuam chaves de encriptação idênticas. Há três opções de conexão de antenas, duas destas são utilizando conectores SMA ou uFL, a terceira opção consiste em utilizar um fio. A figura 3.30 ilustra um rádio LoRa com conector SMA.

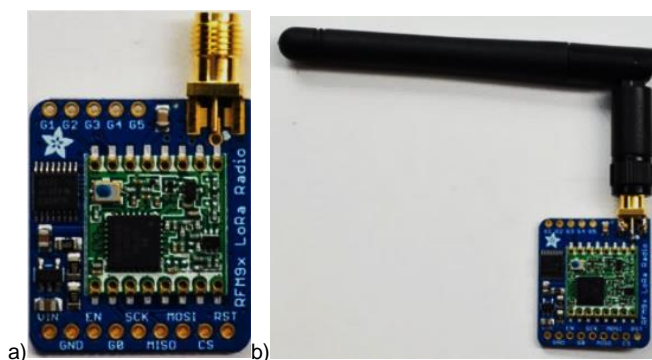


Figura 3.30 Módulo Rádio RFM9x LoRa [65].

DESENVOLVIMENTO

Há duas variantes destes rádios. Estas são o RFM69HCW e o RFM9x LoRa [65]. Estas variantes dos rádios LoRa são ilustradas na figura 3.31.



Figura 3.31 Módulos RF LoRa [65]: a) RFM69HCW. b) RFM9x LoRa [65].

A tabela 3.14 apresenta características elétricas e de desempenho dos rádios descritos [66], estes módulos RF são compatíveis com Arduino. Para tal, utilizam-se bibliotecas *open source* disponibilizadas *online*, estas simplificam a interação entre o microcontrolador e os rádios LoRa, estes comunicam através do protocolo SPI. Como apresentado na tabela 3.14, o alcance é condicionado por vários factores. Dois destes são a potência de transmissão (dBm) e a frequência (MHz), estes parâmetros são configuráveis no controlo embebido. Outros factores, são as características das antenas utilizadas, como por exemplo o ganho. Por fim, o ambiente de propagação no qual a eventual comunicação vá ocorrer é outra condicionante, geralmente, são comunicações sem linha de vista. Como apresentado na tabela 3.14, o módulo RFM9x LoRa possui maior alcance.

Tabela 3.14 Características elétricas dos módulos RF LoRa [66].

Caraterísticas elétricas	RFM69HCW	RFM9x LoRa
Bandas de frequências	433 MHz ou 868/915 MHz	433 MHz ou 868/915 MHz
Processador chip	SX1231	SX1276
Gama de potência de transmissão	+13 dBm até +20 dBm 100 mW	+5 dBm até +20 dBm 100 mW
Corrente consumida nas transmissões	50 mA (+13 dBm) 150 mA (+20 dBm)	100 mA (+20 dBm)
Corrente consumida em modo escuta contínuo	30 mA	30 mA
Alcance (linha de vista) com antenas unidireccionais	500 m	2 km
Alcance (linha de vista) antenas fios	200 a 500 m	2 km
Alcance (linha de vista) antenas diretivas e ajustes de configurações dos módulos rádio	Até 5 km	Até 20 km

O *hardware* de cada rádio LoRa inclui um regulador de tensão para 3,3 V. Ainda integra um circuito lógico que permite transições de níveis de tensão lógicos, tal torna o rádio LoRa compatível com microcontroladores que operem a 3.3 ou 5 V, sem necessitar de circuitos de regulação de tensão adicionais.

DESENVOLVIMENTO

Os rádios LoRa possuem um *pinout* idêntico, descrito na tabela 3.15, também utilizam 5 pinos GPIO (*General-Purpose Input/Output*) cujo nível lógico de tensão é 3.3 V.

Tabela 3.15 Descrição dos pinos dos módulos RF LoRa [66].

Pino	Funcionalidade
Vin	Pino de alimentação. Suporta desde 3.3 a 6 V DC
GND	Pino GND. Em termos de lógica e potência
EN	Pino Enable. É conectado ao regulador de tensão interno. Normalmente definido no estado HIGH para Vin
SCK	Pino SPI Clock. É um pino de entrada para o chip
MISO	Pino SPI MISO (Master In Slave Out). Os dados são enviados do módulo RF para o microcontrolador
MOSI	Pino SPI MOSI (Master Out Slave In). Os dados são enviados do microcontrolador para o módulo RF
CS	Pino SPI CS (Chip Select). É um pino de entrada para o chip. Utilizado para iniciar uma comunicação SPI
RST	Pino SPI Reset. Geralmente encontra-se no estado HIGH. Para a inicialização do módulo é colocado a LOW para acionar o reset ao módulo RF
GO	Pino SPI GO (IRQ GO Interrupt Request). Este pino permite acionar um pedido de interrupção de hardware. Este sinal é enviado do módulo rádio para o microcontrolador

3.6.4. Configurações RFM9x LoRa emissor e mensagem de dados

A tabela 3.16 ilustra as configurações dos parâmetros rádio e MAC realizados para o módulo de comunicação [67], as configurações descritas foram realizadas no controlo embebido do terminal remoto. Utilizou-se o módulo RFM9x LoRa pelas suas potencialidades de maior alcance. Na figura 3.32 segue o *script* do programa a ilustrar as configurações de potência de transmissão e frequência.

Tabela 3.16 Configurações rádio RFM9x LoRa [67].

Configurações rádio RFM9x LoRa	
Versão módulo rádio	RFM9x LoRa
Bibliotecas utilizadas	SPI, RH_RF95, RHGenericDriver, RHReliableDatagram
Endereçamento RHReliableDatagram	Endereço Nó Terminal 203 Endereço Nó Gateway 109
Frequência (MHz)	868
Potência de transmissão (dBm)	14
Configuração modo longo alcance	BW=125 kHz CRC=4/8 SF= 4096 chips/símbolo

```
#include <SPI.h>
#include <RH_RF95.h>
RH_RF95 rf95(RFM95_CS, RFM95_INT);
rf95.setTxPower(14); // configuração da potência de transmissão para os 14 dBm
```

DESENVOLVIMENTO

```
#define RF95_FREQ 868.0
if (!rf95.setFrequency(RF95_FREQ))
{
    Serial.println(" fail ");
    while (1);
}
rf95.setFrequency(RF95_FREQ); // ajuste da frequência para os 868 MHz
```

Figura 3.32 Configuração da potência de transmissão (dBm) e frequência (MHz) para o RFM9x LoRa do nó remoto [67].

Outra definição de importância foi a utilização do endereçamento cliente-servidor (*RHReliableDatagram*) [67], tal foi motivado pelo facto de se pretender assegurar uma comunicação dedicada entre o terminal remoto e o *gateway*. O endereçamento destina-se a proteger o pacote LoRaWAN da escuta por outros rádios LoRa. Quando aplicado o endereçamento, os rádios LoRa que estejam configurados à mesma frequência e dentro do alcance de transmissão, não conseguem detetar o pacote LoRaWAN encriptado. Na figura 3.33 segue o *script* a ilustrar esta configuração. Começa-se, por atribuir endereços em hexadecimal a ambos os nós. É utilizada uma classe *RHReliableDatagram manager*, que é responsável pelo endereçamento do pacote. Esta guarda os parâmetros que referenciam a classe *RH_RF95 rf95* e o endereço atribuído ao terminal remoto.

É utilizada uma função pertencente à classe *RHReliableDatagram manager* para a transmissão da mensagem endereçada, a qual serve-se de três parâmetros. O primeiro é a mensagem a transmitir, o segundo é a dimensão em *bytes* da mensagem e por fim o endereço atribuído ao *gateway*. O último é de grande importância para a identificação do nó destinatário.

```
#include <RHReliableDatagram.h>
// Atribuição endereços hexadecimal: cliente (terminal remoto 203), servidor (gateway 109)
#define TERMINAL_NODE 203
#define GATEWAY_NODE 109
/* instância da biblioteca RadioHead RH_RF95 para o RFM9x */
RH_RF95 rf95(RFM95_CS, RFM95_INT);
/* classe responsável pelo endereçamento das mensagens */
RHReliableDatagram manager(rf95, TERMINAL_NODE);
/* Função que transmite a mensagem endereçada ao gateway
Parâmetros: dados endereçados, dimensão da mensagem, endereço gateway */
manager.sendtoWait((byte *)data,sizeof(data),GATEWAY_NODE);
```

Figura 3.33 Script a ilustrar a configuração do endereçamento *RHReliableDatagram* no RFM9x [67].

Por último, configurou-se o RFM9x LoRa nas definições de longo alcance [67], isto de modo a tirar proveito das suas potencialidades a nível de alcance. A figura 3.34 ilustra a referida configuração realizada através da função *setModemConfig*.

```
#include <RHGenericDriver.h>
/* configuração modo longo alcance: BW=125 kHz CRC=4/8 Sf= 4096 chips/símbolo */
rf95.setModemConfig(RH_RF95::Bw125Cr48Sf4096);
```

Figura 3.34 Configuração do modo de transmissão de longo alcance do RFM9x LoRa [67].

DESENVOLVIMENTO

A sua utilização implica invocar a biblioteca RHGenericDriver. Definiram-se os parâmetros do rádio descritos na tabela 3.16, como a largura de banda, a codificação CRC e a taxa de transmissão máxima para a modulação em espalhamento espectral.

A figura 3.35 ilustra o *payload* a transmitir, o qual armazena os dados lidos dos ADC. Estes são os níveis de tensão da bateria e o sinal elétrico extraído do circuito de condicionamento para o sensor de CO. O *payload* tem uma dimensão limitada, uma vez que, pela regulamentação europeia, o *duty-cycle* para a banda ISM dos 760 MHz é de 1% [29]. Foi utilizado um pacote de 10 *bytes* para o envio dos dados.

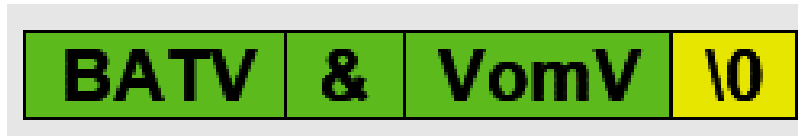


Figura 3.35 Constituição do payload LoRa – mensagem de dados do terminal remoto.

3.6.5. Consumos energéticos

A tabela 3.17 apresenta os dados referentes aos consumos energéticos do terminal remoto. De acordo com estes dados, conseguiu-se estimar uma autonomia energética de aproximadamente 19 dias.

Tabela 3.17 Consumos energéticos terminal remoto.

Bateria (mAh)	Corrente modo adormecido (μ A)	Corrente durante a transmissão (mA)	Tempo de amostragem dos dados (minutos)	Autonomia (dias)
2500	250	78	2	19

3.6.6. Arquitetura

A figura 3.36 ilustra a arquitetura interna do nó remoto.

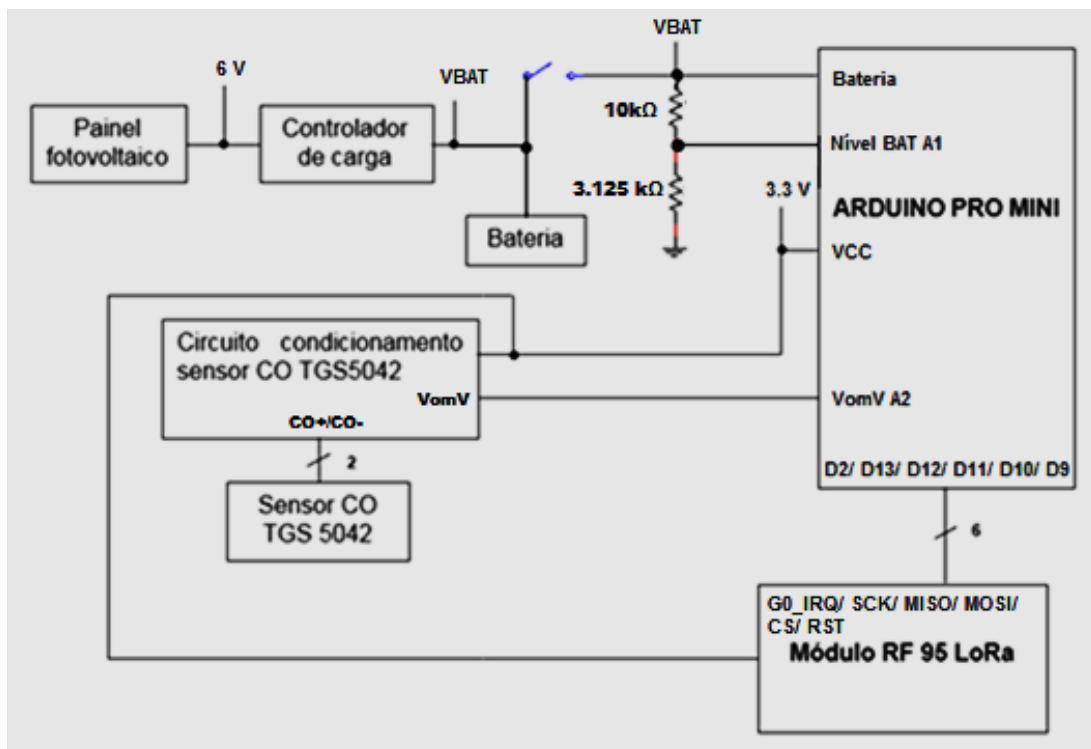


Figura 3.36 Arquitetura interna do nó remoto.

No desenvolvimento do *hardware* recorreu-se a placas de circuito pré-perfuradas. O *hardware* foi resguardado numa caixa elétrica com as características IP69. Utilizou-se uma estrutura em madeira rígida para afixar a caixa elétrica e o sistema anti-rradiação. No seu topo foi instalado o painel fotovoltaico referenciado na seção 3.6.2. O esquemático do circuito, a eletrónica na placa de circuito pré-perfurada e a estrutura do nó terminal remoto são disponibilizados no Anexo K.

3.6.7. Controlo embebido

A figura 3.37 ilustra o fluxograma que caracteriza o controlo embebido do nó terminal.

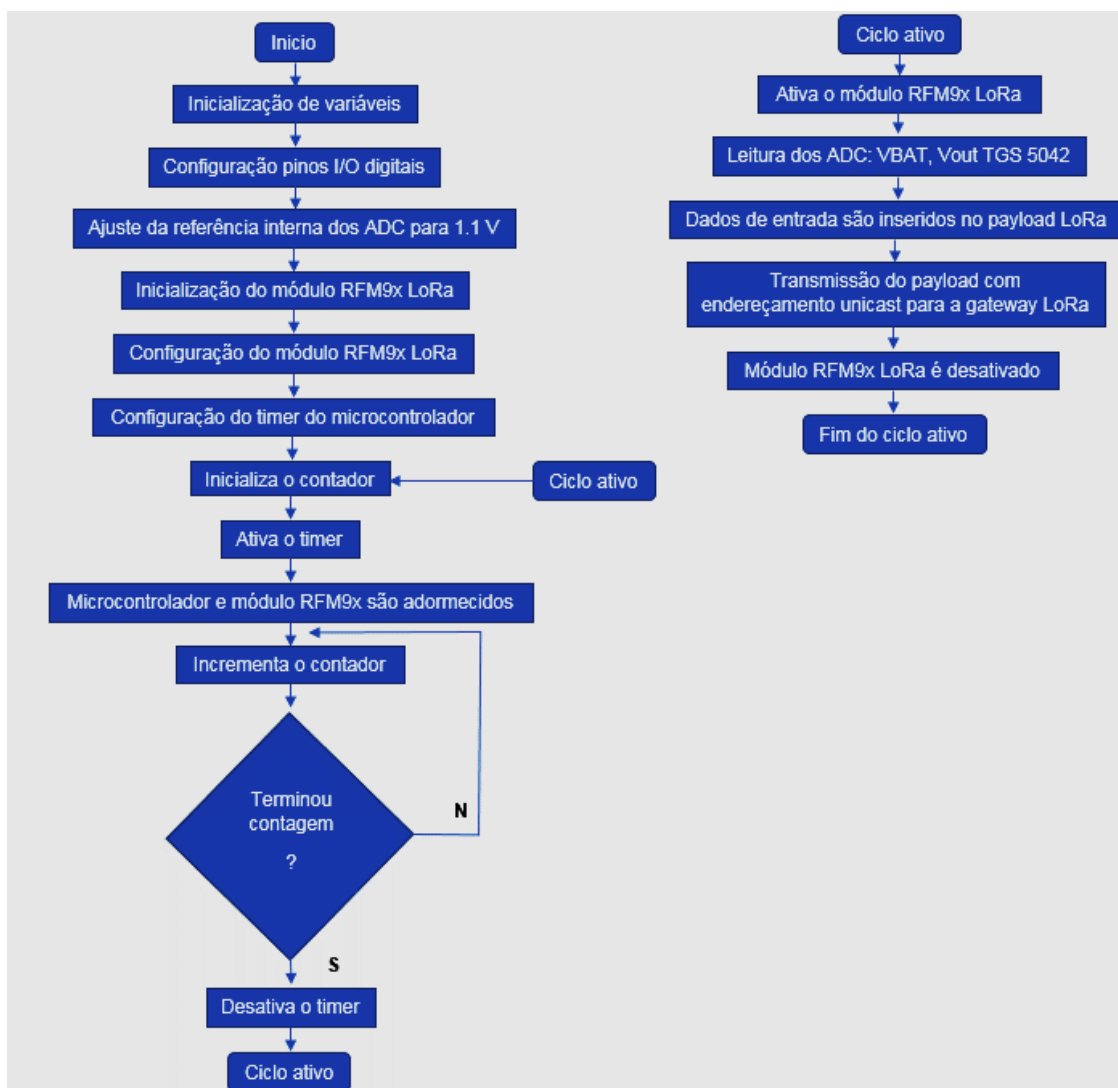


Figura 3.37 Fluxograma descrevendo o controlo embebido realizado para o nó remoto.

São inicializadas variáveis e configurados os pinos I/O digitais. É realizado o ajuste da tensão de referência para os 1,1 V, pois pretendia-se tirar partido da melhor resolução nas leituras de amostras. Em seguida, o módulo de comunicação é inicializado, realizaram-se as configurações do RFM9x LoRa descritas na seção 3.6.4.

DESENVOLVIMENTO

O *timer* do microcontrolador foi configurado como contador de forma semelhante ao realizado com o nó local. A sequência de instruções inclui o adormecimento do microcontrolador e do RFM9x LoRa. O adormecimento do rádio LoRa implicou o auxílio de recursos da biblioteca de Arduino RadioHead. [67]. Teve-se em consideração, uma regulamentação do *duty cycle* estabelecida na Europa para os rádios LoRa de 1%. É possível definir um *duty cycle* desta ordem percentual através do estabelecimento de uma relação temporal adequada entre o ciclo ativo e o adormecido, o que também otimiza a eficiência energética do terminal remoto.

O período ativo resume-se a leitura de amostras, inserção dos dados de entrada no *payload* e a transmissão. Os dados foram inseridos no *payload* do modo apresentado na seção 3.6.4. A duração do ciclo adormecido é estabelecida com base em dois fatores. O primeiro é a duração do ciclo ativo, o segundo é o *duty cycle* de interesse. Deste modo, a sua duração temporal foi atribuída considerando a equação (2.10). No final do ciclo ativo após se transmitir o *payload*, o rádio retorna ao estado inativo. Já o microcontrolador é adormecido assim que o *timer* é ativado. O código desenvolvido para o controlo embebido é apresentado no Anexo K.

3.7. Módulo gateway LoRa

Nesta seção é descrito o nó *gateway* LoRa/Zigbee. Este nó é importante na recolha de dados do terminal remoto. Este *gateway* é responsável por enviar os dados de saída para a base de dados.

3.7.1. Diagrama funcional

A figura 3.38 ilustra o respetivo diagrama funcional. O *gateway* recebe os dados do terminal remoto, os quais estão encapsulados no formato do protocolo LoRaWAN. Os dados de entrada são os descritos na seção 3.6.4. Estes são inseridos numa trama do protocolo Zigbee. O nível da bateria na *gateway* também é monitorizado.

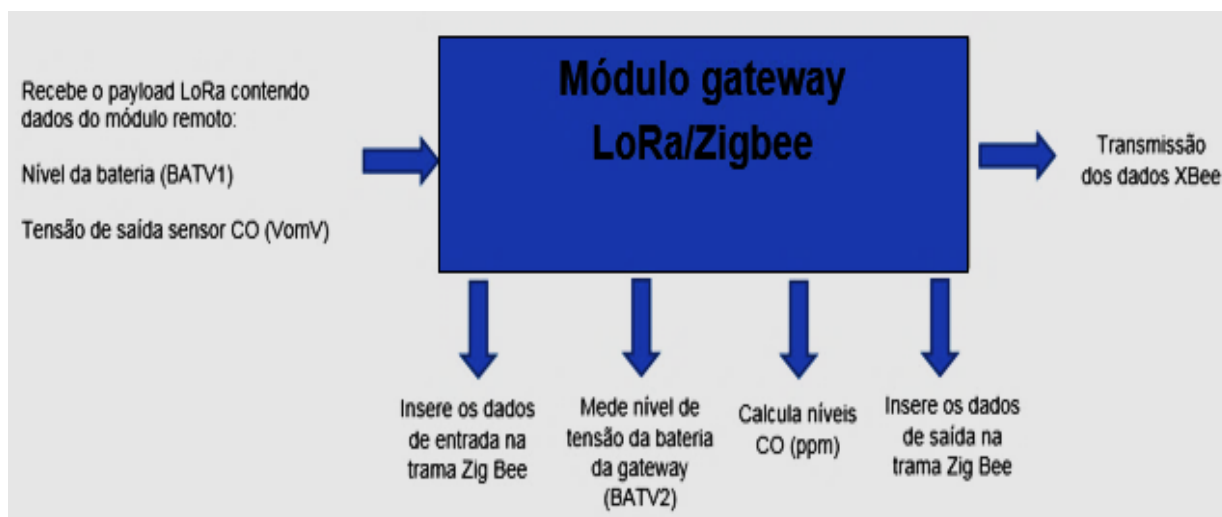


Figura 3.38 Diagrama funcional – nó gateway LoRa/Zigbee.

3.7.2. Consumos energéticos

A tabela 3.18 apresenta os consumos energéticos dos módulos de comunicação utilizados. A tabela 3.19 mostra os consumos de energia da *gateway*, daqui se conseguiu estimar uma autonomia de aproximadamente 5 dias.

Tabela 3.18 Consumos energéticos módulos de comunicação.

Consumos energéticos módulos de comunicação			
Recetor RFM9x LoRa		Router XBee Series 2	
Modo <i>stand by</i> (mA)	Modo recepção (mA)	Modo ativo (mA)	Modo adormecido (mA)
10	74	30	0,02

Tabela 3.19 Consumos de energia nó *gateway*.

Consumos energia nó <i>gateway</i>					
Bateria (mAh)	Corrente <i>stand by</i> (mA)	Período <i>stand by</i> (s)	Corrente período ativo (mA)	Período ativo (s)	Dias
5000	45	133	105	0,098	5

Como apresentado nas tabelas 3.18 e 3.19, o *gateway* tem consumos energéticos consideráveis. Um motivo é atribuído à dependência dos ciclos ativos do nó remoto, uma vez que, a *gateway* fica aguardando por mensagens. Durante este período, o microcontrolador está em espera ativa, enquanto o RFM9x LoRa escuta o canal. Desta forma, não se estabeleceram ciclos adormecidos. É de realçar que com a configuração do endereçamento cliente-servidor, a *gateway* apenas receberá mensagens do terminal remoto.

Outro fator são os consumos de corrente no ciclo ativo, os quais aumentam consideravelmente, como demonstrado nas tabelas 3.18 e 3.19. Um dos motivos é os consumos do rádio LoRa para receber os dados do nó remoto. Outro motivo é o processamento realizado pelo microcontrolador, ainda o XBee é ativado para a transmissão dos dados de saída. Para melhorar a autonomia energética foram necessários alguns cuidados. Para tal, utilizaram-se dois painéis fotovoltaicos de 2 W conetados em paralelo e um controlador de carga. Este *hardware* reunia características elétricas semelhantes ao descrito na seção 3.6.2. Ainda se utilizaram duas baterias de lítio recarregáveis de 2500 mAh.

3.7.3. Configurações módulo RFM9x LoRa recetor

A tabela 3.20 apresenta as configurações rádio do RFM9x LoRa recetor [67], estas são realizadas no controlo embebido do nó *gateway*. As configurações são semelhantes às realizadas no rádio LoRa emissor para o nó remoto, com a exceção do longo alcance. É necessário realçar a importância do endereçamento cliente-servidor no rádio recetor, pois possibilita um canal de comunicação exclusivamente para o terminal remoto e o *gateway*.

Tabela 3.20 Configurações módulo de comunicação LoRa [67].

Configurações módulo de comunicação LoRa		
Versão módulo rádio	RFM9x LoRa	
Bibliotecas utilizadas	SPI, RH_RF95, RHReliableDatagram	
Frequência (MHz)	868	
Potência (dBm)	14	
Endereçamento RHReliableDatagram	Endereço Nó Terminal 203	Endereço Nó Gateway 109

O *script* apresentado na figura 3.39, caracteriza a configuração do endereçamento cliente-servidor no programa [67]. Tal como no terminal remoto, atribui-se os respetivos endereços em hexadecimal que identificam ambos os nós. Há que realçar, que deverão ser os mesmos endereços previamente atribuídos no programa do terminal remoto. A classe RHReliableDatagram manager é responsável por estabelecer o endereçamento do rádio LoRa. Esta guarda dois parâmetros, o primeiro refere-se à classe RH_RF95 rf95, que é responsável pelas configurações do rádio LoRa. O segundo parâmetro trata-se do endereço que identifica o *gateway*.

A função `recvfromAck` é responsável pela receção da mensagem do terminal remoto, que pertence à classe RHReliableDatagram manager. Esta recebe como parâmetros, o conteúdo da mensagem (`buf`), a dimensão em *bytes* (`&len`) e o endereço em hexadecimal (`&from`). Para a mensagem ser filtrada pelo recetor LoRa, o campo `&from` deverá receber o endereço em hexadecimal que identifica o terminal remoto. Deste modo, é identificado que o pacote LoRaWAN recebido de facto provém do terminal remoto. Em tal eventualidade, o conteúdo da mensagem, dimensão e endereço do terminal remoto são armazenados no *buffer*, caso contrário, o pacote é simplesmente descartado.

```
#include <RHReliableDatagram.h>
// Atribuição endereços: cliente (terminal remoto 203), servidor (gateway 109)
#define TERMINAL_NODE 203
#define GATEWAY_NODE 109
/* instância da biblioteca RadioHead para o modulo RFM9x */
RH_RF95 rf95(RFM95_CS, RFM95_INT);
/* classe responsável pelo endereçamento das mensagens */
RHReliableDatagram manager(rf95, GATEWAY_NODE);
// se houverem mensagens
if (manager.available()) {
    /* recebe a mensagem com os parâmetros:
    conteúdo da mensagem, dimensão em bytes, endereço terminal remoto*/
    if (manager.recvfromAck(buf, &len, &from)) {
        // receção dos dados
        Serial.print(" got request from TERMINAL NODE : 0x"); // notifica
        Serial.print(from, HEX); // Endereço terminal remoto
        Serial.print(": ");
        Serial.println((char*)buf); // imprime a mensagem
    }
}
```

Figura 3.39 configuração do endereçamento cliente-servidor no programa [67].

3.7.4. Configurações módulo RF XBee Series 2

A tabela 3.21 mostra as configurações rádio realizadas para o XBee Series 2, há que referir duas destas em particular. A sua importância, deve-se às características do sistema remoto quanto à monitorização e transmissão de dados, concretamente, o terminal remoto realiza leitura de amostras periódica a cada 120 segundos. Deste modo, é importante manter o módulo RF activo durante um determinado intervalo de tempo que considere o ciclo de espera da gateway e restantes atrasos decorrentes nas comunicações.

DESENVOLVIMENTO

A primeira configuração é o *Cyclic sleep period*, este parâmetro caracteriza o tempo que um rádio XBee é capaz de se manter acordado e conectado a uma determinada rede enquanto não recebe pacotes. Como tal, introduziu-se a sequência de bytes assinalada na penúltima linha da tabela 3.21, para o referido campo. A segunda configuração é o multiplicador temporal para o *Cyclic sleep period*. Tal, permite multiplicar por um fator “n” o período ativo do rádio XBee configurado, como tal, atribuiu-se um valor apropriado a este multiplicador. O valor de saída, constitui o período ativo total atribuído ao rádio XBee, sendo este assinalado a negrito na última linha da tabela 3.21.

Tabela 3.21 Configurações Router XBee Series 2.

Configurações Router XBee Series 2	
Versão firmware	ZIGBEE ROUTER API
Endereço Router XBee	0013A200 40F74F1C
PAN ID	99
Scan Channels	0xFFFF
Endereço Destinatário 64 bits	0x000000000000FFFF
Endereço Destinatário 16 bits	0xFFFE
Nome	NO GATEWAY LORA
Número máximo saltos (NH)	1E
Duração do ciclo adormecido/ Cyclic sleep period x 10 ms (SP)	AF0 (2,8 segundos) *10 =28 segundos
Configuração sleep mode cíclico (SM)	SM=5 28*5=140 segundos

A tabela 3.22 apresenta o formato da mensagem de saída no protocolo Zigbee.

Tabela 3.22 Formato dos dados de saída a transmitir no formato de protocolo Zig Bee.

Constituição da trama Zig Bee	
Byte de início	0x7E
Comprimento da mensagem	61 bytes: 0x00 0x3D
Identificador de API	Formato trama emissão: 0x10
Numerador da mensagem	0x00
Endereço geral Coordenador 64 bits	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Endereço Coordenador 16 bits	0xFF 0xFE
Campo número de saltos	0x00
Tipo de mensagem	Mensagem broadcast: 0x00
Conteúdo da mensagem	?BATV1=valor1&BATV2=valor2&VomV=valor3&COTGSppm=valor4
Byte de checksum	0x00

3.7.5. Inserção dos dados de saída na base

A tabela 3.23 mostra os parâmetros de interesse da classe MySQL, que é implementada no ficheiro “MySQL.java”, o qual é responsável pelo acesso à base de dados MySQL. O endereço IP que o campo url guarda é o mesmo referenciado para o nó sensor local. Porém, foi atribuído um diferente nome ao campo dbName. Este identifica a base de dados para a estação remota, estando assinalado na última linha da tabela 3.21. O nome da dbName também foi atribuído no ficheiro “dbinfo.PHP”.

Tabela 3.23 Parâmetros da classe que implementa o acesso à base de dados em “MySQL.java”.

url	10.1.222.250
user	*
pass	*
dbName	CO remoto LORA

DESENVOLVIMENTO

No código fonte “msgCare_base.java” consideraram-se quatro sensores. Os primeiros dois campos de dados referem-se aos níveis das baterias para o terminal remoto e o *gateway*, BATV1 e BATV2, respetivamente. Os restantes campos dizem respeito ao sinal elétrico do sensor de CO (VomV) e às respetivas concentrações de monóxido de carbono (COTGSppm).

Inicialmente, os dados de entrada são guardados em variáveis temporárias com o formato de dados apropriado. Posteriormente, os mesmos são armazenados na matriz “dataMatrix”. A matriz de dados armazena o nome de cada sensor e respetivos dados de interesse. É necessário converter para o formato *string* os valores dos sensores. A figura 3.40 apresenta o *script* do código “msgCare_base.java”.

```
// Número de sensores que vão ser lidos : BATV1, BATV2, VomV, COTGSppm
int numberOfSensors = 4;
// Matriz de dados com duas colunas: Nome do sensor, dados de interesse
String [][] dataMatrix = new String[numberOfSensors][2];
// dados de entrada são guardados em variáveis temporárias
float BATV1=(((data[1]&0xFF)<<8)+(data[2]&0xFF)); // BATV1
float BATV2=(((data[3]&0xFF)<<8)+(data[4]&0xFF)); // BATV2
int VomV=(((data[5]&0xFF)<<8)+(data[6]&0xFF)); // VomV
float COTGSppm=(((data[7]&0xFF)<<8)+(data[8]&0xFF)); // COTGSppm
// Armazenamento dos dados na matriz
// os dados com os valores de interesse são convertidos para o formato string
dataMatrix[0][0] = "BATV1";    dataMatrix[0][1] = Float.toString(BATV1); //BATV1
dataMatrix[1][0] = "BATV2";    dataMatrix[1][1] = Float.toString(BATV2); //BATV2
dataMatrix[2][0] = "VomV";     dataMatrix[2][1] = Integer.toString(VomV); //VomV
dataMatrix[3][0] = "COTGSppm"; dataMatrix[3][1] = Float.toString(COTGSppm); //COTGSppm
return dataMatrix;
```

Figura 3.40 Script do código fonte “msgCare_base.java”.

O código “Msg_collector.java” é responsável por capturar as mensagens da porta série, estas são depositadas no *buffer* de dados. Como mencionado anteriormente, as mensagens provêm do Coordenador Zigbee. O código “Msg_parser.java” lê cada mensagem do *buffer* e realiza o seu processamento. Como se referiu anteriormente, o interpretador da aplicação Java realiza a distinção e a validação das mensagens recebidas através da leitura do primeiro carater do payload, o qual deverá ser “?”. Em caso afirmativo, a mensagem é inserida na base de dados.

3.7.6. Arquitetura

A figura 3.41 apresenta a arquitetura interna do nó *gateway*. No fornecimento de energia utilizaram-se dois painéis solares de 2 W conetados em paralelo, um controlador de carga e duas baterias recarregáveis de 2500 mAh. Os módulos de comunicação foram o RFM9x LoRa e o XBee Series 2. Fez-se uso de um divisor resistivo para a monitorização do nível das baterias. O microcontrolador é o Arduino Pro-Mini. O esquemático do circuito, a eletrónica na placa de circuito pré-perfurada e a estrutura do nó terminal remoto são disponibilizados no Anexo L.

DESENVOLVIMENTO

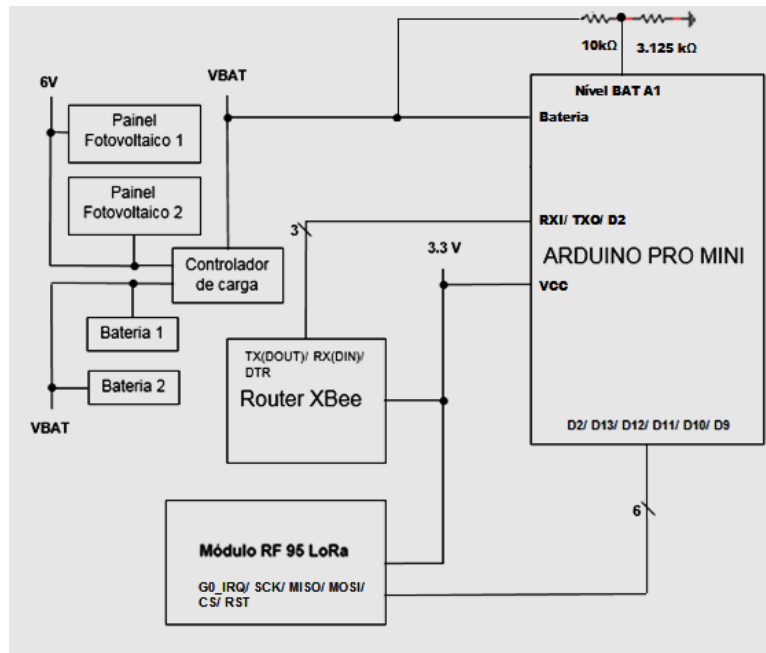


Figura 3.41 Arquitetura interna nó gateway.

3.7.7. Controlo embebido

A figura 3.42 apresenta o fluxograma que caracteriza o controlo embebido do gateway.

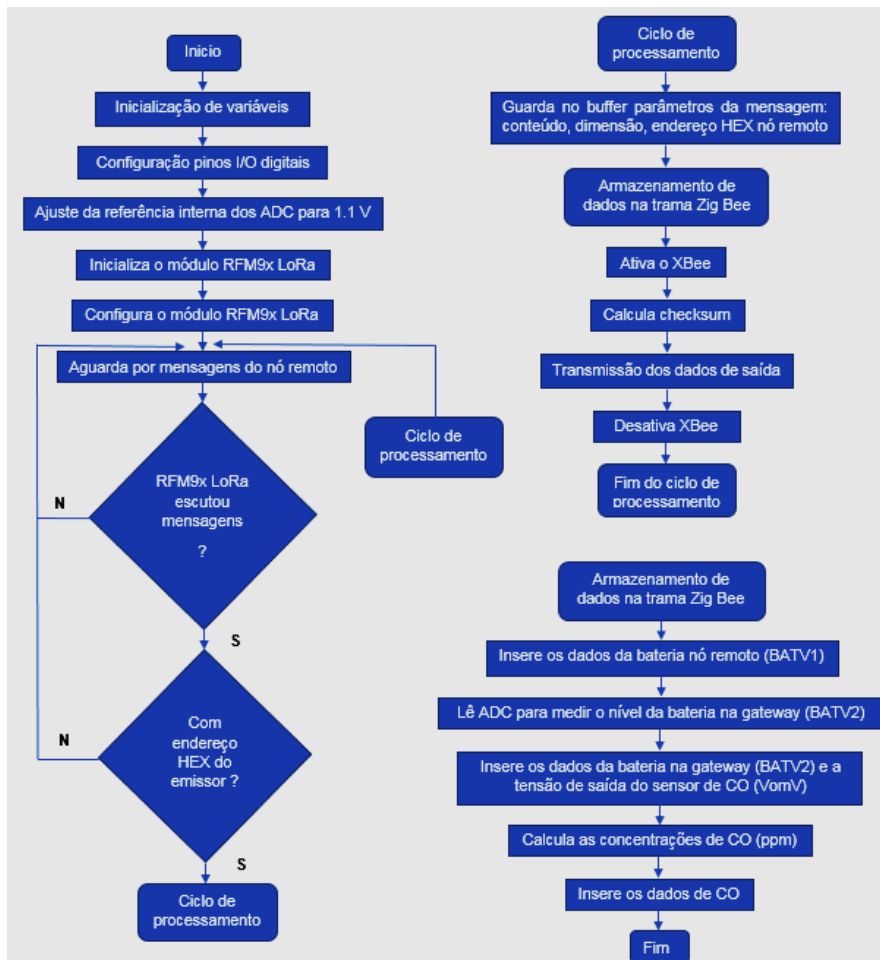


Figura 3.42 Fluxograma que descreve o controlo embebido do nó gateway.

DESENVOLVIMENTO

Após a inicialização de variáveis, são configurados os pinos I/O digitais utilizados. Dado o facto de monitorizar o nível da bateria do *gateway* por ADC, ajustou-se a tensão de referência dos ADC para os 1,1 V. O módulo RFM9x LoRa é inicializado e realizam-se as configurações descritas na seção 3.7.3.

Como referido, o controlo embestado da *gateway* é caracterizado por ciclos de espera e ciclos ativos, dos quais, a maioria do tempo a *gateway* encontra-se em ciclo de espera. Para o ciclo ativo iniciar-se, é necessário a receção do pacote LoRaWAN especificamente da estação remota.

Ao iniciar-se o período de processamento, o *payload* da mensagem de interesse é armazenado no *buffer*. Ainda é armazenada a dimensão em *bytes* da mensagem e o endereço hexadecimal que identifica a mensagem como sendo do nó remoto. Os dados do *payload* recebido são os apresentados na seção 3.6.4, estes são inseridos na trama Zigbee.

São realizadas outras operações de tratamento de dados, entre as quais monitorizar o nível da bateria local na *gateway*. Também são determinadas as concentrações de monóxido de carbono. Os dados de saída são encapsulados no formato do protocolo Zigbee. O módulo RF XBee é ativado, este irá transmitir os dados de saída. Após isso, o módulo RF XBee retorna ao estado inativo e inicia-se um novo ciclo de espera. O código desenvolvido para o controlo embestado é apresentado no Anexo L.

4. Resultados

Este capítulo apresenta a análise de dados para os resultados obtidos com os módulos de monitorização desenvolvidos neste projeto.

4.1. Sensor Alphasense B4 – testes nó de calibração

A figura 4.1 apresenta medições de monóxido de carbono realizadas com o sensor Alphasense B4 e sensores Figaro TGS 5042. O objetivo era estudar a resposta e desempenho dos sensores na presença de CO. Os testes realizaram-se no laboratório de telecomunicações da Universidade. Utilizou-se um isqueiro, papel e um frasco de modo a produzir-se um ambiente com CO.

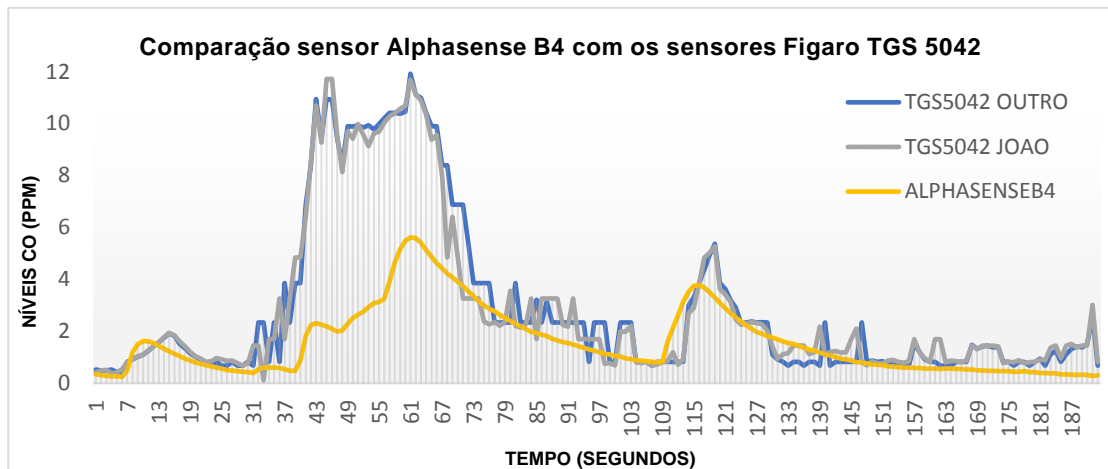


Figura 4.1 Comparação sensor Alphasense com os sensores Figaro TGS 5042.

É possível notar que os sensores Figaro TGS 5042 apresentam variabilidades de CO aproximadas. Por outro lado, o sensor Alphasense B4 apresenta discrepâncias em comparação aos restantes sensores. As suas concentrações de CO apresentam menor variabilidade assumindo curvas de dados mais suaves e menores níveis de CO.

As folhas de características de ambos os fabricantes [51] [2] comprovam que o Alphasense B4 possui uma melhor resolução e menor erro de medição. Deste modo, o Alphasense B4 possui uma gama de medição relativamente limitada comparado ao TGS 5042, tal traduz-se em menores variabilidades das concentrações de CO. O Alphasense B4 quando exposto a maiores concentrações de CO permitiu registar níveis próximos dos 13 ppm. De acordo com a expressão (3.3), este valor corresponde a cerca de 14,89 mg/m³. Pela tabela 2.1, o seu índice de qualidade do ar (IQA) é classificado como Mau. O valor registado foi atribuído como sendo o máximo da gama de medição, pois a partir deste nível o sensor de CO começava a entrar em saturação. Este nível em ppm era facilmente excedido pelo TGS 5042.

Os sensores Figaro possuem uma gama de medição até aos 10000 ppm. Como tal, menor precisão e maior erro. Isto conduz a maior variabilidade dos dados de CO registados. Tal, implica uma calibração mais rigorosa quando se pretende medir concentrações de CO mais baixas como explicado na seção 3.4.4. Da experiência realizada, aferiu-se a necessidade destes ajustes de calibração para os nós local e remoto posteriormente.

RESULTADOS

Entre os dados registados no gráfico da figura 4.1, os máximos registados pelo Alphasense B4 aproximavam-se dos 6 ppm. Nestes instantes, os sensores Figaro registaram concentrações próximas dos 10 ppm. Outro possível factor ao qual se poderiam atribuir as discrepâncias dos dados era o posicionamento e o distanciamento dos sensores na bancada do laboratório em relação a um frasco, o qual utilizou-se para a queima de papel. Dependendo do posicionamento e distância dos sensores, eventualmente, as suas superfícies iriam absorver diferentes concentrações de CO. Segue no Anexo H a figura 7.22, esta ilustra a montagem do circuito de teste dos vários sensores de gás.

A figura 4.2 apresenta um teste urbano realizado no Funchal com o sensor Alphasense B4. Mais concretamente, na estrada Eng. Jaime Ornelas Camacho no túnel que faz a ligação entre a Rotunda dos Ilhéus e a Avenida Gulbenkian. O percurso realizou-se de forma pedestre transportando o módulo de calibração para este realizar a monitorização ambiental. O gráfico ilustra as amostras de CO medidas ao longo do percurso. O teste realizou-se entre as 15:30 e as 16:00 horas. As maiores concentrações poluentes registadas foram aproximadamente 3,5 ppm. De acordo com a expressão (3.3), isto corresponde a cerca de 4 mg/m³, pela tabela 2.1 este índice de qualidade do ar (IQAr) é classificado como Muito Bom.

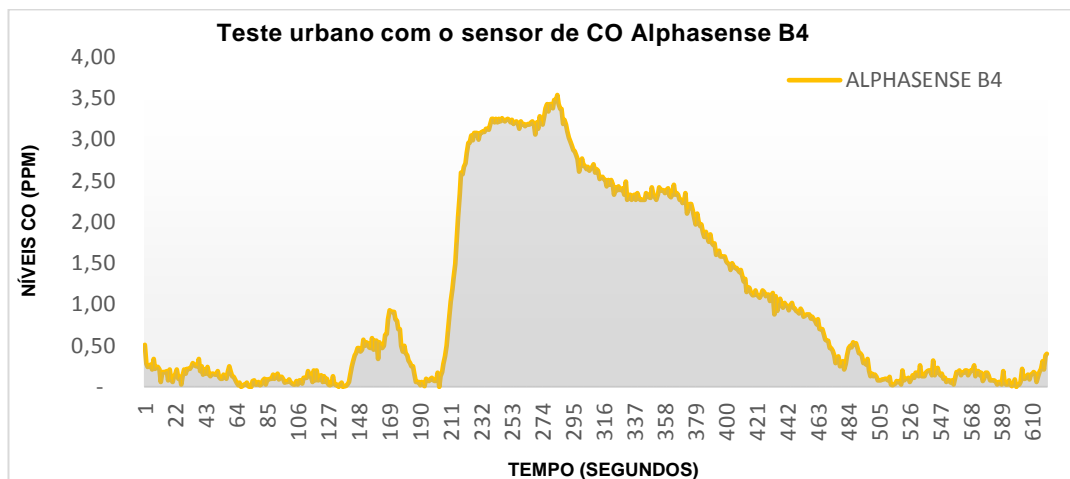


Figura 4.2 Teste urbano com o sensor de CO Alphasense B4.

4.2. Testes no sensor local

Nesta seção são apresentados os testes e os resultados obtidos com o nó sensor local.

4.2.1. Estudo do nível de sinal

O local a instalar o nó sensor foi junto à estrada próxima do laboratório de telecomunicações, de modo a medir a poluição resultante da passagem de automóveis. Antes de instalar o nó sensor foi necessário avaliar o desempenho de comunicação. Para isso, fizeram-se medições da intensidade do nível de sinal recebido (*Received Signal Strength Intensity* - RSSI) proveniente do nó local na zona circundante ao local desejado. Os referidos pontos de medição são apresentados na figura 4.3, esta caracteriza a planta da Universidade referente ao piso -2.

RESULTADOS

O ponto de medição do Coordenador é sinalizado a vermelho e os pontos de medição para o nó local são assinalados a verde. Para cada ponto se realizaram medições do RSSI durante aproximadamente 5 minutos, posteriormente, determinou-se o correspondente nível médio de sinal em dBm, os dados registaram-se no gráfico da figura 4.4.

O gráfico da figura 4.4 apresenta o estudo do nível médio de sinal em função da distância. O ponto A encontrava-se a cerca de 13 metros do Coordenador com -68 dBm de nível de sinal. O ponto B situa-se a cerca de 22 metros do Coordenador e apresenta um nível médio de sinal recebido de -83 dBm. Por fim, observou-se que o ponto D era o limiar do nível de sinal encontrado. Este encontrava-se próximo dos -90 dBm que o Coordenador Zig Bee seria capaz de detetar. A sua distância ao Coordenador era cerca de 40 metros. Determinou-se que o ponto B seria o escolhido para instalar o nó local.

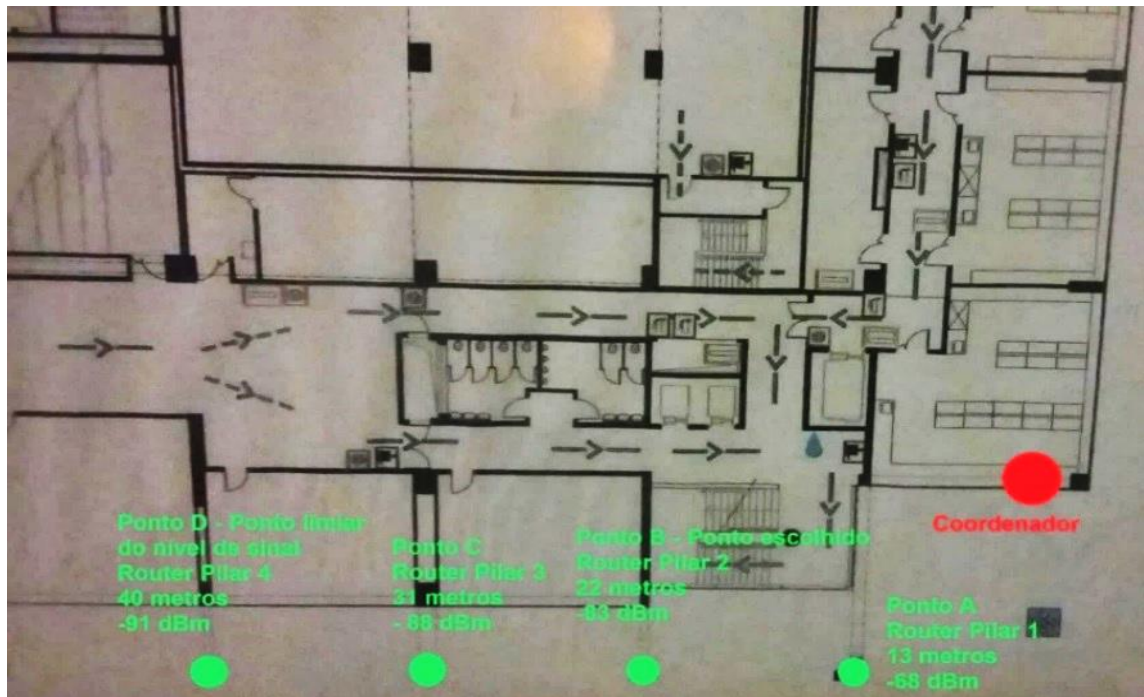


Figura 4.3 Planta do piso -2 da Universidade.

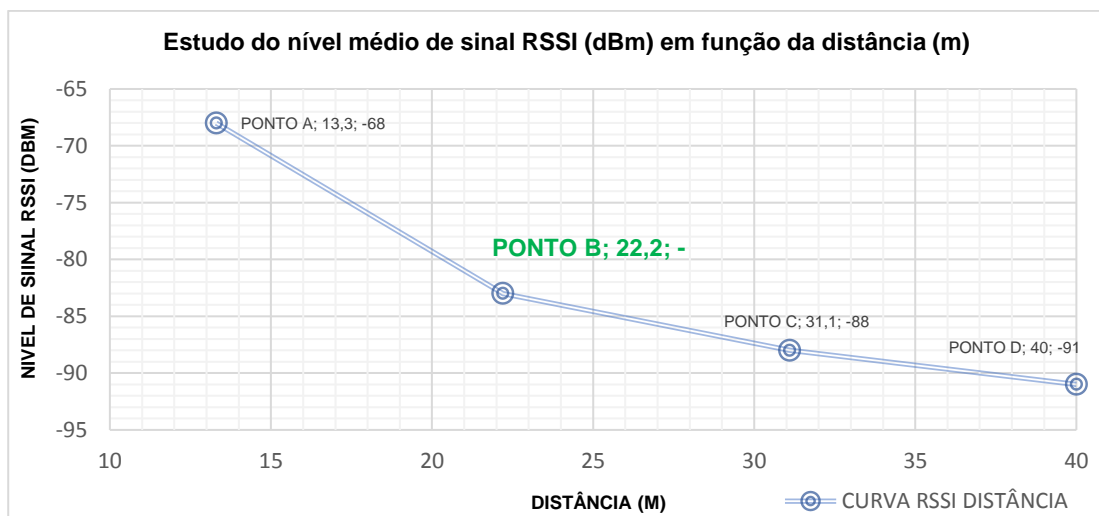


Figura 4.4 Estudo do nível médio do sinal (dBm) em função da distância.

RESULTADOS

4.2.2. Estudo da interferência causada pela radiação solar

Esta seção descreve o estudo do efeito da radiação solar no desempenho do sensor Figaro. A figura 4.5 apresenta a variação da tensão de saída do circuito de condicionamento de sinal. A figura 4.6 apresenta a variabilidade das concentrações de CO. Ambos os gráficos consideram o estudo de duas situações. A primeira caracteriza um ambiente interior ausente de radiação solar, sendo representado nos gráficos a azul. A segunda situação caracteriza um ambiente exterior no qual existe radiação solar, sendo representado pela cor laranja.

Os gráficos mostram que a radiação solar causa oscilações da tensão de saída (V_{OUT}). Conseqüentemente, resulta em oscilações dos níveis de CO. Deste modo, pode-se assumir que a radiação solar é um fator interferente nas medições de CO. Um aspecto particular a referir nestes testes é que o sensor se encontrava no exterior da caixa elétrica exposto ao ar.

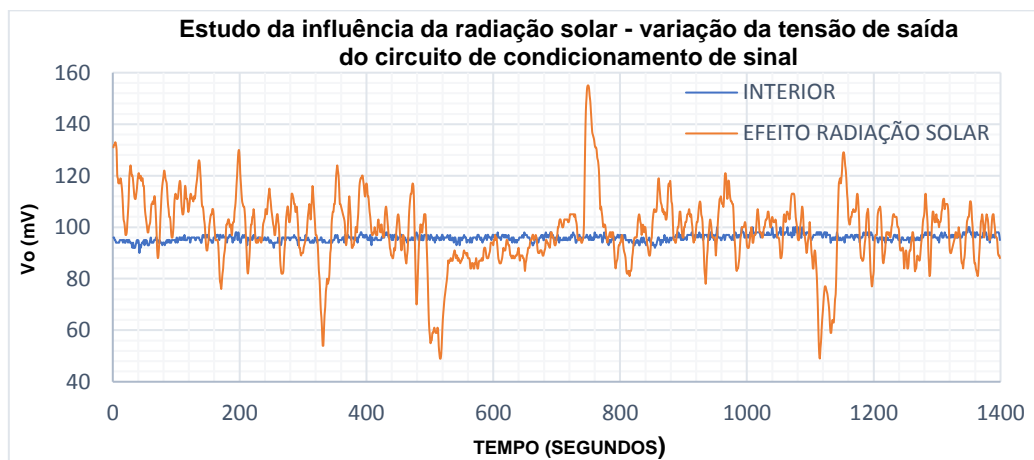


Figura 4.5 Estudo da influência da radiação solar – variação da tensão de saída do circuito de condicionamento de sinal do sensor Figaro TGS 5042.

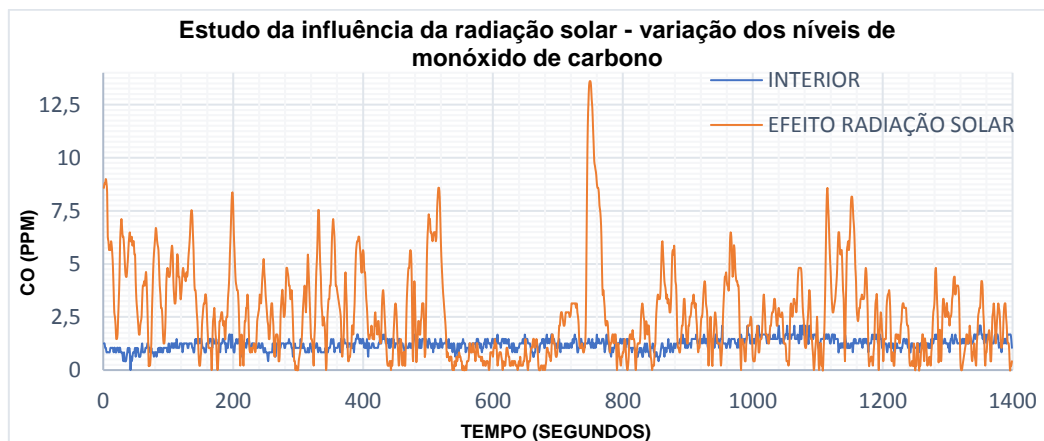


Figura 4.6 Estudo da influência da radiação solar – variação dos níveis de monóxido de carbono medidos.

4.2.3. Estudo da influência da temperatura nas medições de CO

Esta seção descreve o estudo da influência da temperatura nas medições de CO. A finalidade desta experiência consistia em estudar a eventual necessidade de compensar a temperatura interna de modo a evitar influências significativas nas medições de CO. Deste modo, utilizou-se um sensor de temperatura NTC (*Negative Temperature Coeficient*) de 10 kΩ [68], tratando-se de um termístor. Este é um elemento transdutor cuja resistência elétrica é influenciada pelas variações da temperatura ambiente. Concretamente, para o referido sensor, a sua resistência elétrica diminui quando a temperatura aumenta. A figura 4.7 apresenta o circuito de condicionamento de sinal utilizado. Este permite ao microcontrolador a leitura do sinal de entrada com recurso a ADC.

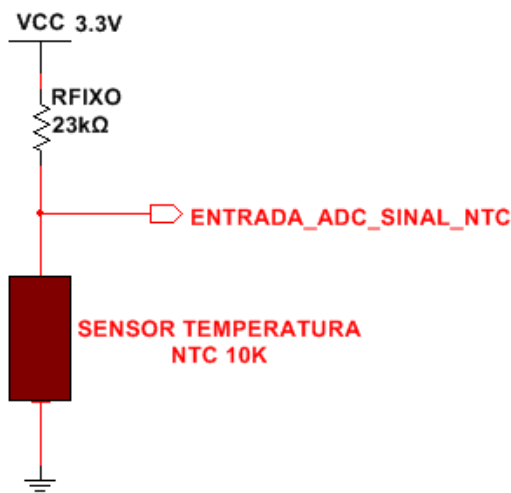


Figura 4.7 Circuito de condicionamento de sinal para ler o sinal de entrada do sensor de temperatura NTC.

O microcontrolador lê o sinal de tensão do termístor ($V_{NTC_{lido}}$) através da seguinte expressão:

$$V_{NTC_{lido}} = \frac{(1,1 * analogRead(A3))}{1023} \quad (4.1)$$

Tendo conhecimento do sinal de tensão do termístor e considerando a relação utilizada no divisor resistivo é possível determinar a resistência elétrica do termístor (R_{NTC}), dada por:

$$R_{NTC} = \frac{23k\Omega * V_{NTC_{lido}}}{3,3 - V_{NTC_{lido}}} \quad (4.2)$$

Um passo importante consistiu em estabelecer a função de calibração para o NTC. Esta permite relacionar a temperatura e a resistência elétrica para o termístor. Um coeficiente importante na calibração do sensor é o fator β , definido por [68]:

$$\beta = \frac{\ln\left(\frac{R_{T1}}{R_{T2}}\right)}{\frac{1}{T1} - \frac{1}{T2}} \quad (4.3)$$

em que R_{T1} , R_{T2} são os coeficientes de resistência elétrica do NTC às temperaturas $T1$, $T2$, respetivamente. Os coeficientes da expressão (4.3) foram consultados na folha de características do NTC [69]. Os seus valores foram registados na tabela 4.1. Deste modo, obteve-se aproximadamente $\beta = 97$.

RESULTADOS

Tabela 4.1 Parâmetros de temperatura e resistência elétrica do NTC correspondente [69].

Parâmetros de temperatura e resistência elétrica do NTC			
T1 (°C)	T2(°C)	RT1(Ω)	RT2(Ω)
25	100	549,4	10000

Outro coeficiente necessário na calibração é o R_{∞} [68]. R_0 caracteriza a resistência elétrica do NTC à temperatura ambiente (t_0). Deste modo, considerou-se $R_0 = 10\text{ k}\Omega$, $t_0 = 25^{\circ}\text{C}$. O valor determinado foi aproximadamente 209. Desta forma,

$$R_{\infty} = R_0 * e^{\left(-\frac{\beta}{t_0}\right)} \quad (4.4)$$

Conhecendo os valores dos coeficientes apresentados anteriormente e da resistência elétrica para o NTC é possível obter a função de calibração que permite calcular a temperatura (T) [68]:

$$T = \frac{\beta}{\ln\left(\frac{R_{NTC}}{R_{\infty}}\right)} \quad (4.5)$$

Deste modo, adicionou-se temporariamente à eletrônica do nó local o circuito da figura 4.7. No controlo embebido fizeram-se os ajustes necessários de modo a adicionar a função de calibração do termístor e efetuarem-se as medições da temperatura interior na caixa elétrica que resguarda o *hardware*. Ainda se adicionou ao *hardware* o módulo microSD referenciado na seção 3.5.1 para armazenamento de dados. Nesta experiência o sensor também estava diretamente exposto ao ar. No Anexo I estão ilustrados os testes para o estudo da influência da temperatura nas medições de CO. Os gráficos das figuras 4.8 e 4.9 apresentam dados de concentrações de CO e da temperatura interior respetivamente.

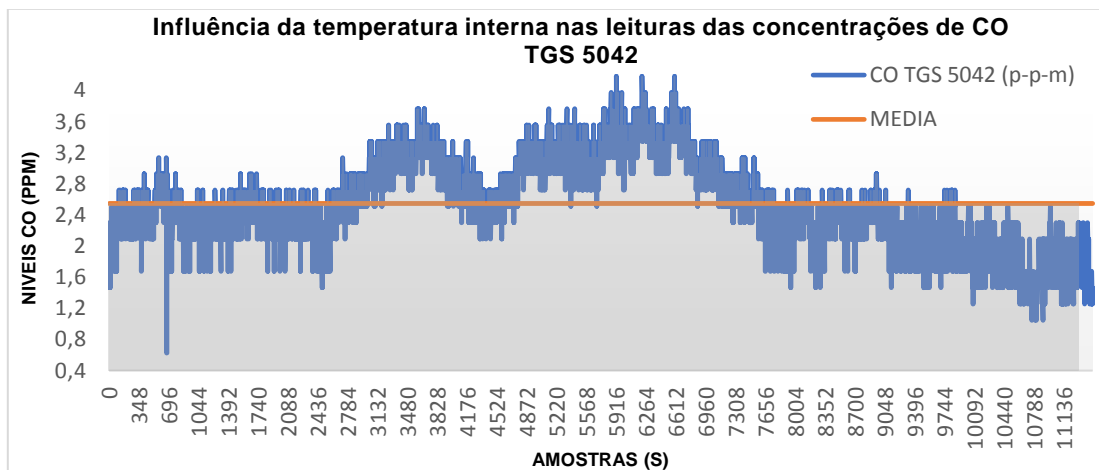


Figura 4.8 Medições das concentrações de CO (ppm).

Analisando as amostras apresentadas no gráfico da figura 4.8, estimou-se uma média de cerca de 2,5 ppm nas concentrações de CO. A temperatura máxima registada no interior da caixa elétrica, aproximou-se dos 55°C como demonstrado no gráfico da figura 4.9. Nesses instantes, registaram-se concentrações de CO próximas dos 4 ppm, verificando-se um desvio na ordem de 1,5 ppm quando a temperatura interior na caixa elétrica atingiu o valor máximo.

RESULTADOS

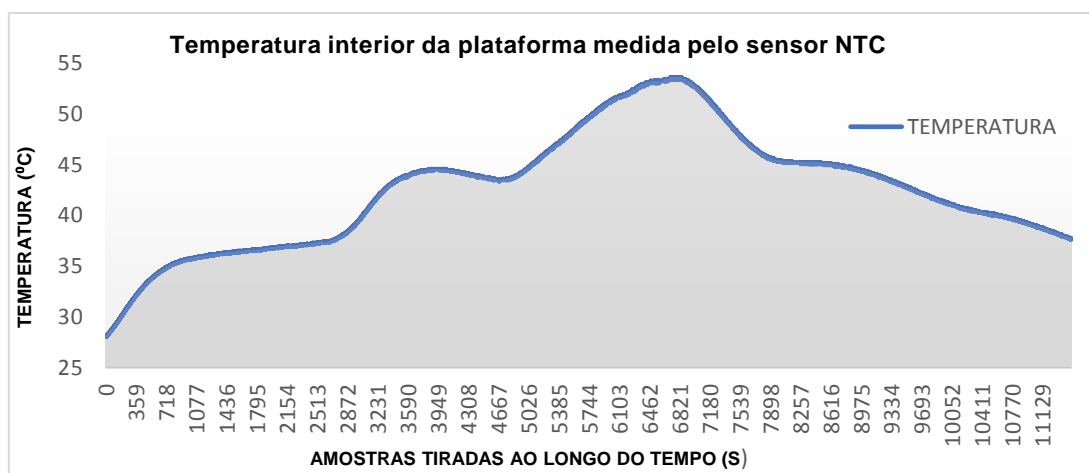


Figura 4.9 Temperatura no interior da plataforma medida pelo sensor NTC.

Ainda se consultou a folha de características do sensor Figaro [51], mais concretamente um gráfico o qual caracteriza a variação do coeficiente de dependência da temperatura do sensor (I/I_0). Os parâmetros I , I_0 caracterizam as correntes de saída do sensor a 400 ppm. O parâmetro I considera várias temperaturas, I_0 considera uma temperatura de 20°C e humidade relativa a 50%. Registou-se que a temperatura entre os 20 e os 60°C o coeficiente I/I_0 apresentava um factor de aproximadamente 1,2. Face aos resultados alcançados na experiência e aos dados consultados na folha de características do sensor Figaro foi decidido que não seria necessário realizar a compensação da temperatura interna da caixa elétrica.

4.2.4. Medições dos parâmetros na Web

Esta seção apresenta os dados monitorizados pelo nó local após a montagem e a instalação no local de monitorização. Estes dados são disponibilizados no servidor Web. O gráfico da figura 4.10 apresenta a variabilidade diária dos níveis de CO. O gráfico da figura 4.11 apresenta dados do índice de radiação solar que, como já se tinha verificado, afeta a medição do CO. Ambos os gráficos consideram dias de maior e menor radiação solar. Deste modo, conseguiu-se estimar uma média diária em torno dos 0,2 ppm de concentrações gasosas.

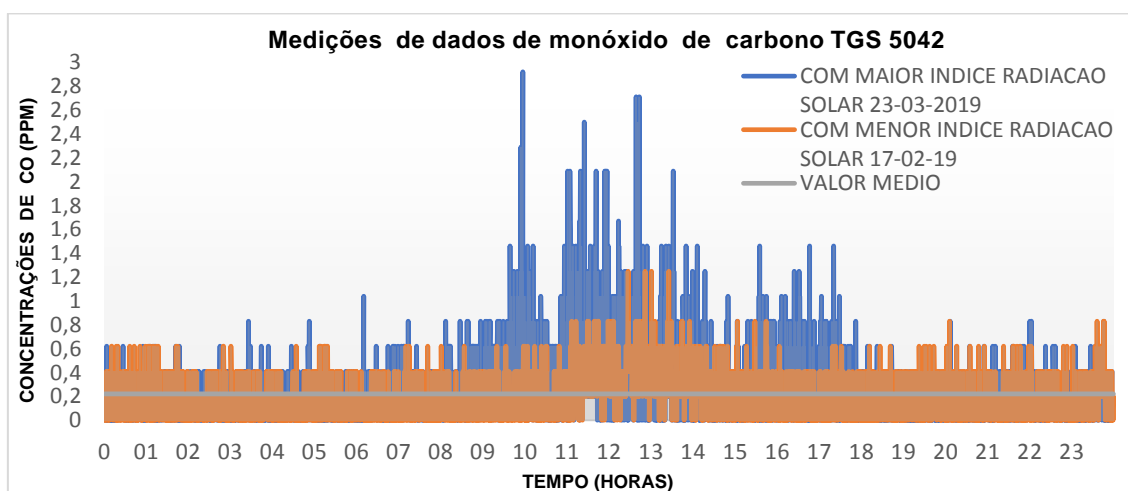


Figura 4.10 Medições de dados de monóxido de carbono considerando a influência da radiação solar.

RESULTADOS

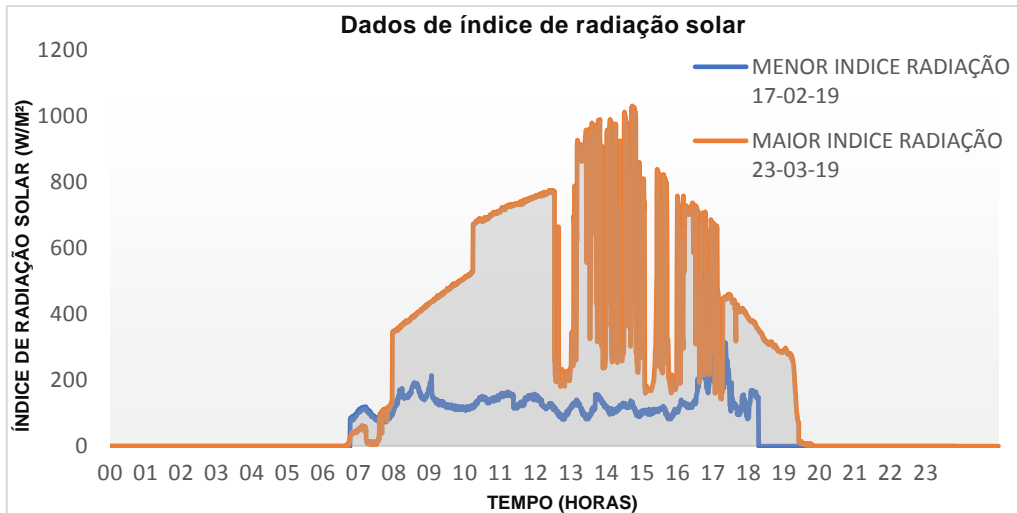


Figura 4.11 Dados de índice de radiação solar.

No dia 17-02-2019, o período de maior concentração gasosa foi entre as 12:00 e as 14:00, os níveis de CO atingiram os 1,2 ppm. Neste dia registaram-se menores índices de radiação solar, os máximos atingiram valores próximos dos 400 W/m². No dia 23-03-2019, os índices de radiação mais elevados foram da ordem de 1000 W/m². Neste dia, como apresentado no gráfico da figura 4.10, durante as horas de maior intensidade solar, os níveis de CO aumentaram. É de destacar o período entre as 10 e as 13 horas nos quais se registou níveis de radiação entre os 700 e os 900 W/m². Durante este período ocorreram picos nos níveis de CO de 2,8 a 2,6 ppm. No entanto, a variabilidade noturna de ambos os dias seguiu uma distribuição de dados semelhante.

De modo a minimizar os efeitos da radiação foi criada uma estrutura antirradiação. Ainda protege o sensor de outros fatores interferentes externos. A estrutura desenvolvida é apresentada na figura 4.12. O Anexo I apresenta a construção desta estrutura.



Figura 4.12 Estrutura antirradiação para o sensor Figaro.

O gráfico da figura 4.13 mostra os níveis de tensão da bateria nos referidos dias. No dia 23-03-2019 é de destacar o carregamento da bateria até aos 4 V durante as horas de maior intensidade da radiação solar, concretamente entre as 09:00 e as 14:00 o nível da bateria subiu de 3,91 até 4 V. Os índices de radiação durante este intervalo de tempo chegaram a atingir 1000 W/m².

RESULTADOS

Durante o dia 17-02-2019, a tensão baixou desde os 3,92 até aos 3,85 V, uma vez que os índices de radiação solar neste dia foram baixos. Na maioria das horas diurnas, encontravam-se próximos dos 200 W/m². No entanto, este índice de radiação foi suficiente para manter a tensão da bateria nos 3,88 V durante as horas diurnas. Durante as 24 horas do referido dia a tensão diminuiu cerca de 70 mV.

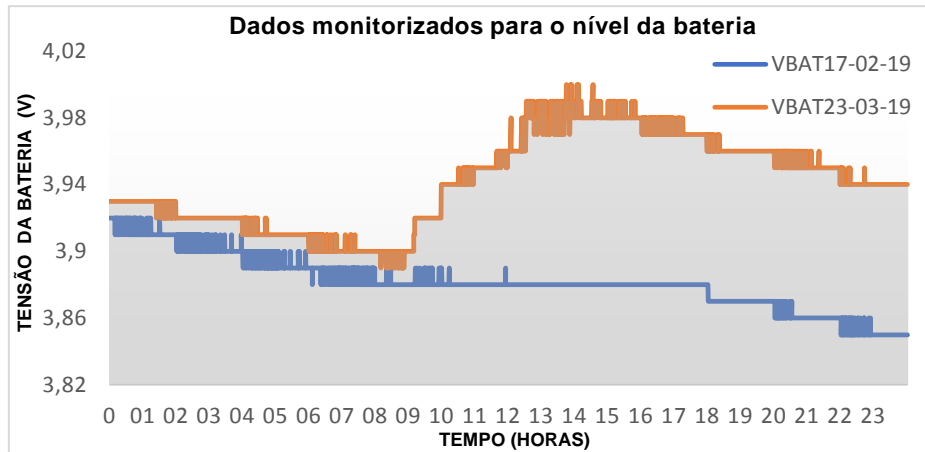


Figura 4.13 Dados monitorizados para o nível da bateria.

As figuras 4.14 e 4.15 apresentam gráficos da variabilidade das concentrações de CO registadas em horas com maior probabilidade de afluência de tráfego urbano no parque de estacionamento inferior da Universidade. Estes dados foram registados em alguns dias entre janeiro e maio de 2019. Os períodos do dia considerados foram entre as 08:00 e as 10:30 e as 16:30 e as 19:00. Para ambos os gráficos, os níveis de CO que se registaram mais frequentemente encontravam-se próximos de 1 ppm. Pela expressão (3.3), as concentrações aproximam-se de 1,15 mg/m³, pela tabela 2.1 o IQAr local é classificado como Muito Bom. O Anexo I apresenta gráficos adicionais da variabilidade de CO nas horas de maior afluência do tráfego urbano local.

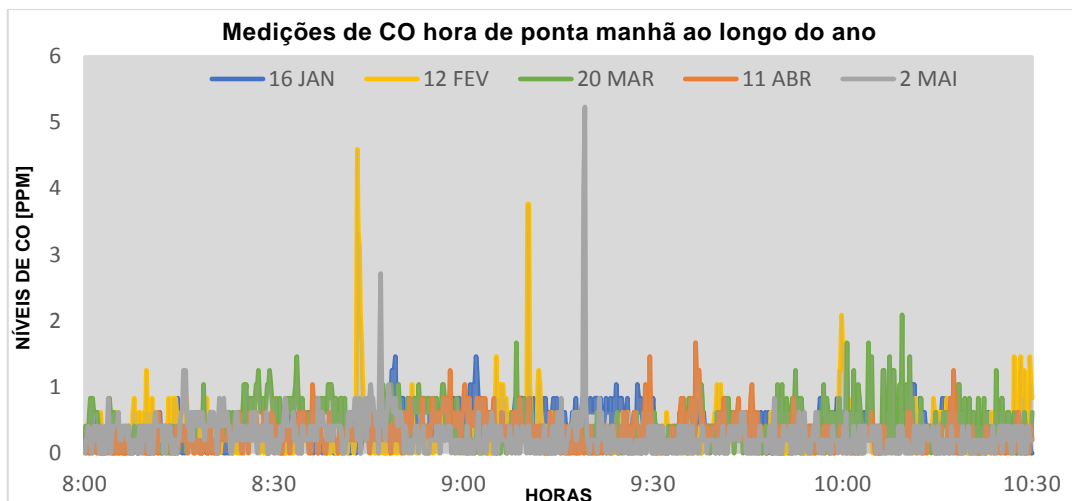


Figura 4.14 Medições de CO entre as 08:00 e as 10:30 alguns dias entre janeiro e maio.

Dos dados apresentados no gráfico da figura 4.14 se destacam as concentrações registadas a 12 de Fevereiro entre as 08:30 e as 09:00 e as 09:00 e 09:30, dos quais, se registaram níveis de 4,5 e 4 ppm. Pela expressão (3.3), isto traduz-se em níveis de 5,16 e 4,58 mg/m³.

RESULTADOS

É necessário referir o dia 2 de maio entre as 09:00 e as 09:30, onde se registou concentrações próximas dos 5 ppm, isto se traduz em cerca de 5,73 mg/m³. Quanto aos dados do gráfico da figura 4.15 foram registados diversos picos próximos de 1,5 ppm nos dias apresentados. É de destacar o pico registado a 20 de março próximo das 18:30, o qual, se encontrava os 3,5 e os 4 ppm.

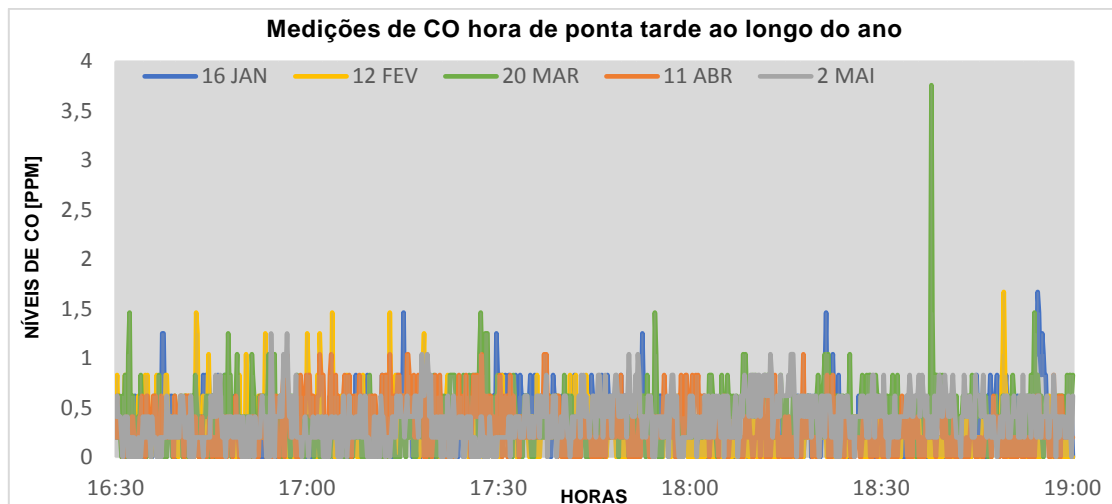


Figura 4.15 Medições de CO entre as 16:30 e as 19:00 alguns dias entre janeiro e maio.

4.3. Gateway de aquisição de dados para estações móveis

Esta seção apresenta dados recolhidos pelo *gateway* de aquisição de nós móveis. Os dados apresentados foram previamente monitorizados por um nó móvel com recetor GPS, o qual havia sido desenvolvido e utilizado num projeto de mestrado anterior [54]. A *gateway* recolheu esses dados e determinou as concentrações de CO, os referidos dados são apresentados no gráfico da figura 4.16. A figura 4.17 apresenta um mapa contendo alguns pontos geográficos referentes ao gráfico.

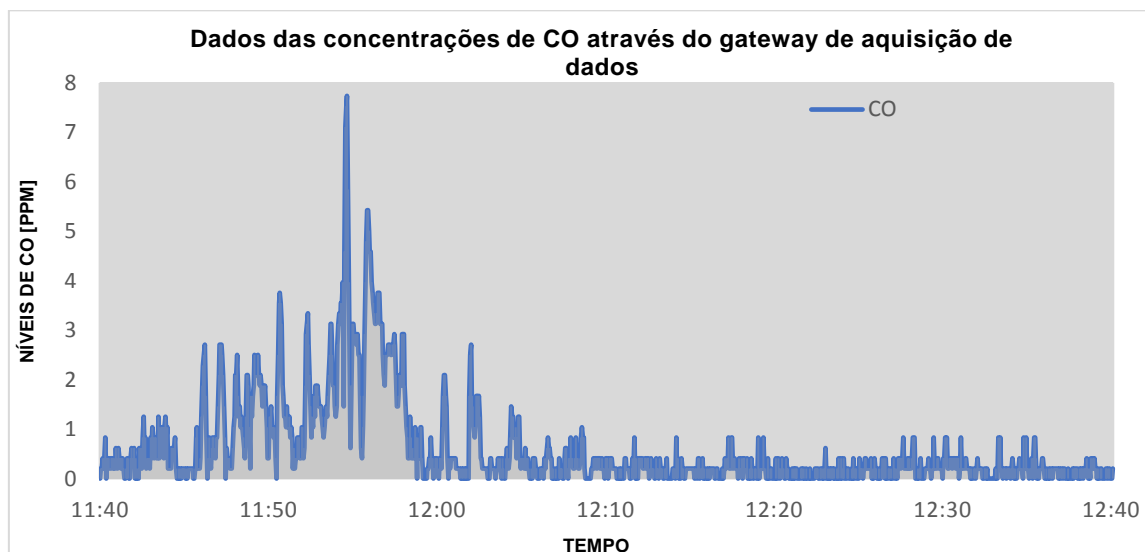


Figura 4.16 Dados de concentrações de monóxido de carbono recolhidas pelo gateway de nós móveis a 13-06-2018.

RESULTADOS

O gráfico da figura 4.16 apresenta dados de monóxido de carbono recolhidos a 13-06-2018 entre as 11:40 e as 12:40. O nível mais elevado registado aproximou-se dos 8 ppm, tal traduz-se em 9,16 mg/m³, em termos de IQAr é classificado como Fraco. No Anexo J é apresentada uma tabela que contém dados temporais e coordenadas geográficas. O seu intervalo de dados está estimado entre as 11:40 e as 11:45 do referido dia, sendo este o período que o recetor GPS havia conseguido receber com sucesso as coordenadas espaciais. Deste modo os pontos geográficos apresentados no mapa da figura 4.17 correspondem ao referido intervalo de tempo. Para se elaborar este mapa introduziram-se as coordenadas geográficas no Google Earth Pro [70]. Através de análise dos dados de CO, dados temporais e na existência de dados de GPS é possível localizar os pontos geográficos nos quais se obtiveram determinadas concentrações de CO. Servindo-se dos dados temporais e espaciais é possível determinar o percurso percorrido.

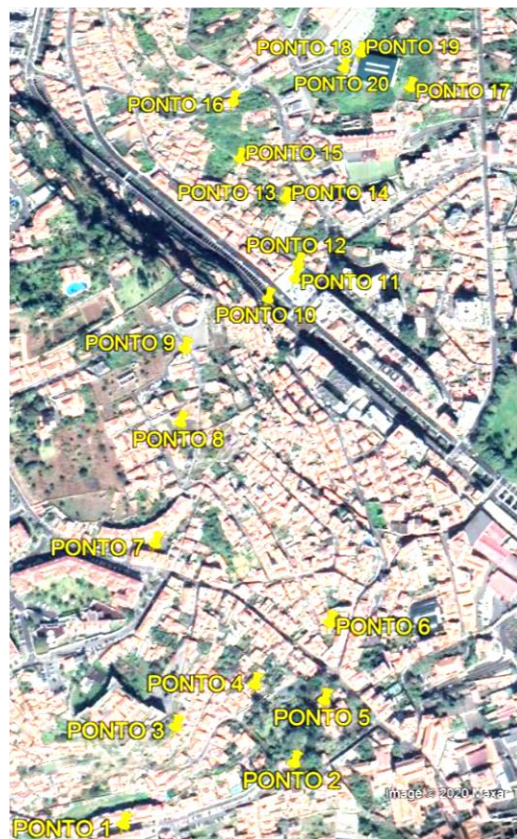


Figura 4.17 Representação das amostras recolhidas pelo gateway de aquisição referentes ao gráfico da figura 4.16.

4.4. Estudo e testes – sistema remoto LoRa-Zigbee

Esta seção apresenta os resultados de estudos e dados recolhidos com o sistema remoto de monitorização.

4.4.1. Estudo da eficiência de transmissão

Neste estudo analisou-se a eficiência de transmissão. Para tal, utilizou-se o sistema de comunicação ilustrado na figura 4.18.

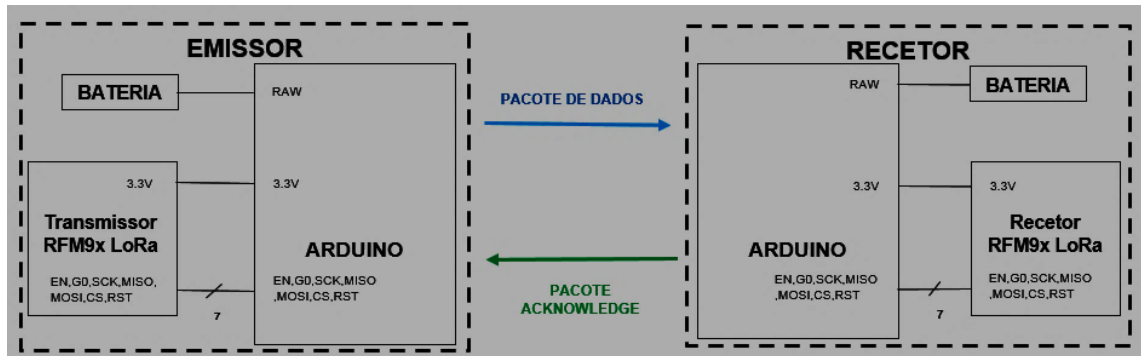


Figura 4.18 Sistema de comunicação utilizado no estudo da eficiência de transmissão em função do período de timeout.

Ambos os nós utilizam *hardware* semelhante. Isto inclui o microcontrolador, o módulo de comunicação e bateria. Utilizou-se o código de teste RadioHead [66], no qual o emissor transmite o pacote de dados ao recetor. O recetor confirma a entrega dos dados com o pacote Acknowledge. Este último foi crucial no estudo da eficiência de transmissão. Na monitorização das comunicações utilizou-se o monitor série do Arduino. A eficiência das comunicações foi testada nos modos rádio de médio e longo alcance.

O parâmetro de controlo é o ajuste de um determinado período temporal. Este caracteriza o tempo necessário para transmitir um pacote de dados e receber o pacote *Acknowledge*. Este é um período de espera ativa ajustável no controlo embebido. Se o pacote ACK não for recebido nesta janela temporal a mensagem de dados é assumida como perdida. Nos testes este período temporal seria reduzido até se encontrar um limite onde iriam ocorrer as perdas de pacotes. A partir daqui seria avaliada a taxa de sucesso percentual. As figuras 4.19 e 4.20 apresentam os dados registados para os respetivos modos de transmissão.

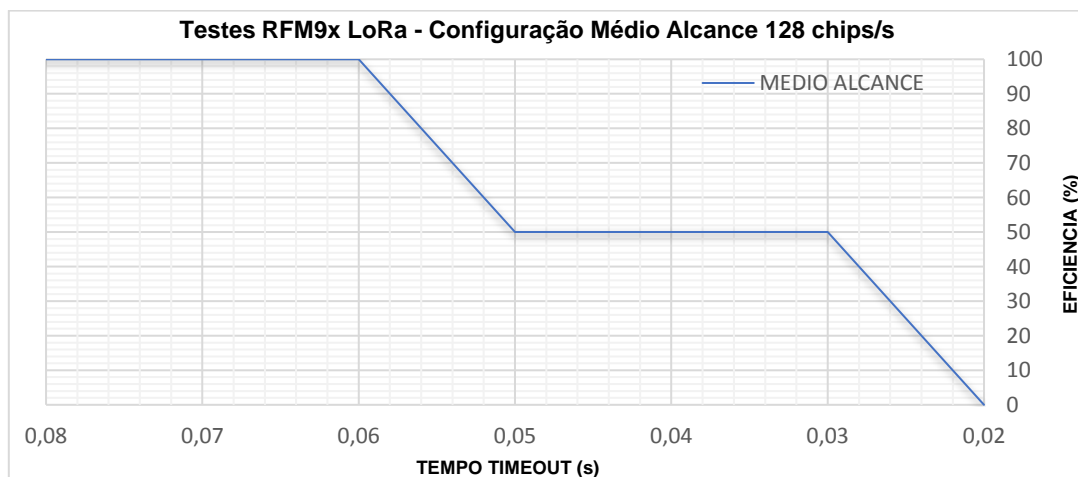


Figura 4.19 Testes RFM9x LoRa – Configuração médio alcance 128 chips/s.

RESULTADOS

Para o gráfico da figura 4.19, nas condições de médio alcance observou-se que abaixo dos 60 ms começou a haver perdas de pacotes. No longo alcance apresentado na figura 4.20 até aos 3 segundos a eficiência percentual manteve-se a 100%. Reduzido esse período temporal observaram-se perdas.

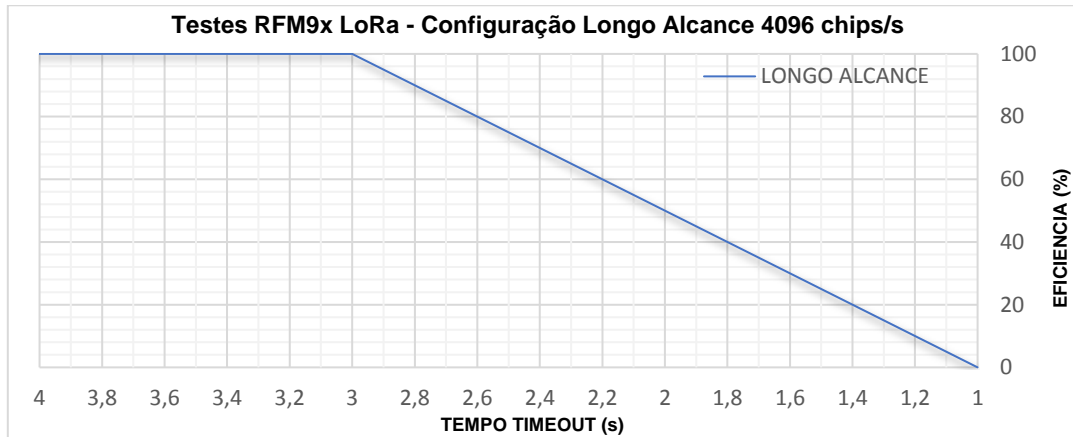


Figura 4.20 Testes RFM9x LoRa – Configuração longo alcance 4096 chips/s.

4.4.2. Estudo do alcance de transmissão RFM9x LoRa

Também se realizou um estudo ao nível do alcance de transmissão da tecnologia LoRa. Isto considerando as características geográficas da ilha. Utilizou-se a configuração rádio de longo alcance. Houve necessidade de utilizar um Raspberry Pi [71]. Esta placa de desenvolvimento é um computador de tamanho reduzido com um sistema operativo baseado em Linux, o qual se apresenta na figura 4.21 a). Também foi necessário utilizar uma *app* Android, esta é o ViewRanger [72]. Este trata-se de uma *app* gratuita que permite apresentar um mapa do terreno circundante. A sua funcionalidade de relevância neste estudo reside no facto de ser possível assinalar um determinado ponto geográfico e registar as coordenadas geográficas, as quais podem ser apresentadas em diversos formatos. Entre estes UTM WGS84 (Universal Transverse Mercator), DDD (graus decimais), DMM (Graus e Minutos) e DMS (Graus, Minutos e Segundos). A figura 4.21 b) ilustra uma posição no mapa na aplicação Android ViewRanger.



Figura 4.21 a) Raspberry Pi [71]. b) Aplicação Android ViewRanger [72].

RESULTADOS

A figura 4.22 apresenta o sistema de comunicação utilizado. O emissor é um nó com *hardware* semelhante ao apresentado na figura 4.18, este foi instalado no terraço da Universidade. Teve-se o cuidado de posicionar a antena do módulo de comunicação verticalmente, teve-se a mesma consideração no recetor de modo a respeitar as exigências de polarização das antenas e otimizar a qualidade de sinal. O Raspberry Pi foi utilizado para ver os dados, em cada instante, recebidos no recetor. Este permitia utilizar o IDE Arduino nativo e monitorizar a comunicação no monitor série. A visualização é realizada num ecrã de 7" ligado ao Raspberry Pi através da porta DSI (*display serial interface*). O fornecimento de energia foi através de uma bateria de 12 V e um regulador de tensão ajustável para os 5 V, o sistema ainda contém um interruptor geral. O nó recetor conecta-se ao Raspberry Pi através de uma porta USB. O sistema utiliza um circuito de monitorização auxiliar para o nível da bateria, o qual é alimentado por USB.

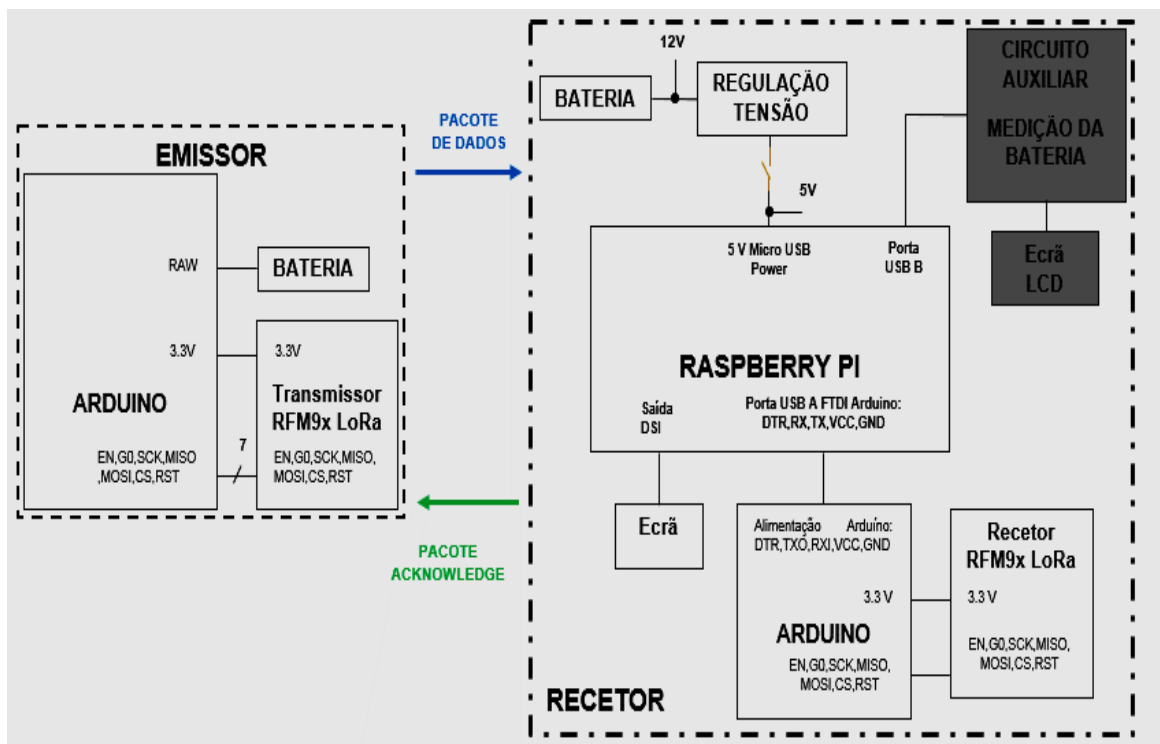


Figura 4.22 Sistema de comunicação utilizado.

A figura 4.23 ilustra os pontos geográficos nos quais se realizaram os testes de alcance. As suas coordenadas espaciais foram introduzidas no Google Earth Pro. Os pontos de medição estão sinalizados por cores de acordo com o nível médio de sinal (RSSI) obtido. Os locais com RSSI registado superior aos -120 dBm estão assinalados a verde, os locais com RSSI entre os -120 e os -130 dBm estão assinalados a amarelo. Por fim, os pontos geográficos com RSSI inferior a -130 dBm e/ou que tenham ocorrido perdas de pacotes estão assinalados a vermelho. O ponto emissor (A0) está assinalado a azul.

RESULTADOS

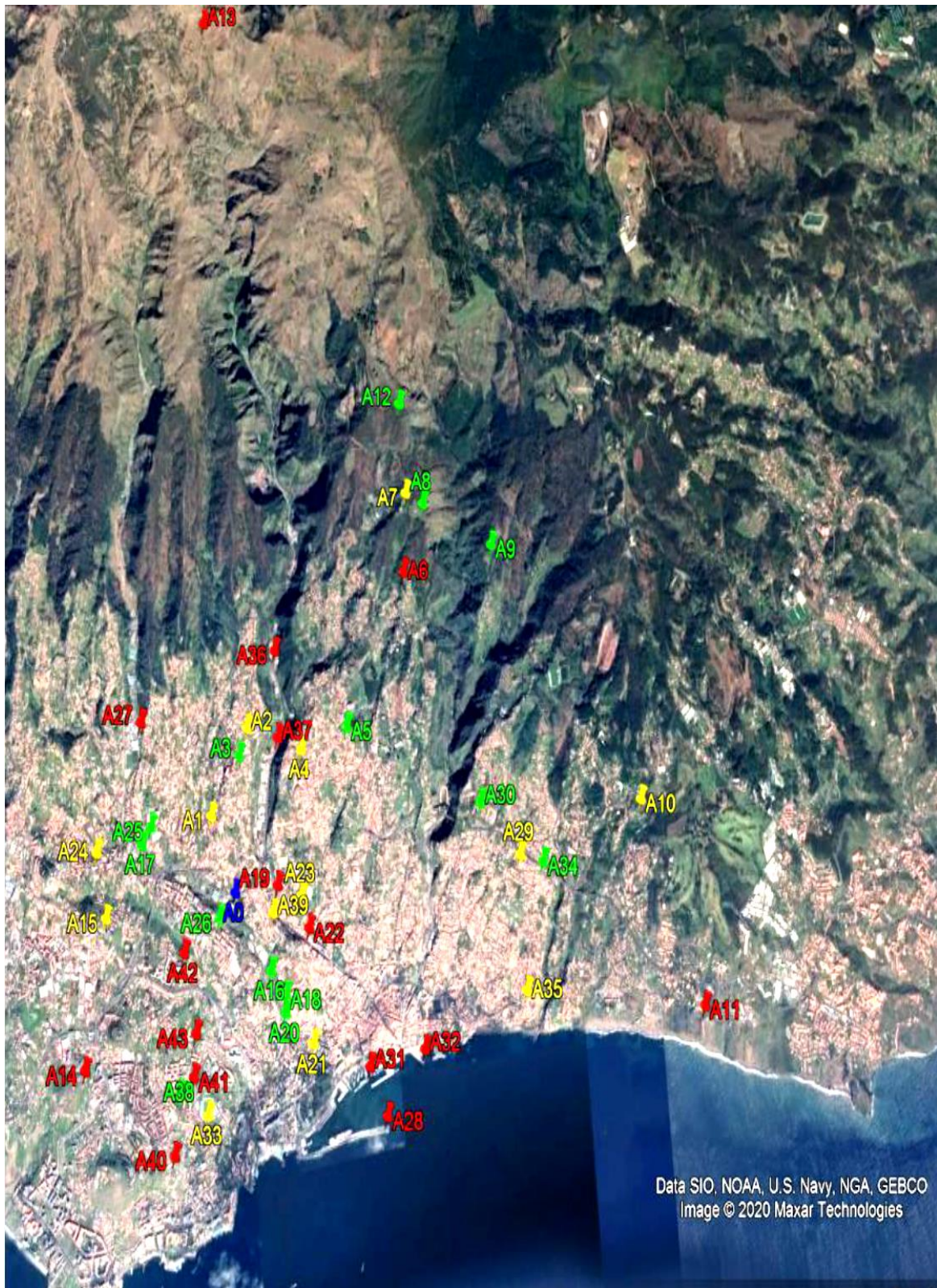


Figura 4.23 Mapa no Google Earth Pro – locais de teste de alcance com o recetor RFM9x LoRa.

RESULTADOS

Os pontos geográficos apresentados no mapa da figura 4.23, registaram-se na tabela 4.2 com a referida sinalização de cores.

Tabela 4.2 Pontos geográficos dos testes de longo alcance com o RFM9x LoRa.

Identificação ponto medição	Localização	Distância comunicação (km)		Identificação ponto medição	Localização	Distância comunicação (km)
A1	C.V.QUINTA	0,67		A23	APEL	0,76
A2	C.L.SEGUNDO	1,28		A24	S.ANTÓNIO IGREJA	1,64
A3	S.ROQUE IGREJA	1,07		A25	P.OLIMPICAS S.ANTÓNIO	1,12
A4	LIVRAMENTO ASSECOS	1,3		A26	AV.MADALENAS	0,26
A5	E.MARMELEIROS	1,77		A27	P.D.MARÍTIMO S.ANTÓNIO	1,78
A6	M.PALACE	3,1		A28	PONTINHA	2,48
A7	Q.T.LUTA	3,64		A29	B.NOVA IGREJA	3,26
A8	T.LUTA C.PRETOS	3,68		A30	B.SUCESSO J.BOTÂNICO	2,86
A9	C.P.ROMEIROS	3,94		A31	CAIS FUNCHAL	2,08
A10	P.GOLF MIRADOURO	4,67		A32	AV.MAR PROMENADE	2,52
A11	PINACULO	5,44		A33	EST.BARREIROS	1,68
A12	PICO ALTO	4,25		A34	B.NOVA V.R	3,52
A13	POÇO NEVE ARIEIRO	7,24		A35	LOUROS	3,44
A14	S.MARTINHO IGREJA	2,1		A36	ROT.FUNDOA	1,92
A15	P.BARCELOS	1,47		A37	E.FUNDOA	1,27
A16	M.CASH	0,75		A38	TUNEL NAZARÉ	1,6
A17	C.SAÚDE ST.ANTONIO	1,16		A39	MIRADOURO ACHADA	0,47
A18	R.DR.JOÃO S.V.CAROÇO	1		A40	R.EST.U.AMERICA	2,05
A19	CONTINENTE VIVEIROS	0,49		A41	R.VIRTUDES L.MARITIMO	1,44
A20	E.S.JOÃO GALP	1,11		A42	PILAR EDIFÍCIOS	0,72
A21	A21 LA VIE	1,5		A43	R.VIRTUDES	1,13
A22	R.31 JANEIRO	0,9		-	-	-

Entre as áreas assinaladas a verde há que referir os pontos A12 e A26. A primeira localiza-se no miradouro do Pico Alto a cerca de 4,25 km do ponto emissor, este é um ambiente que reúne condições de linha de vista. O segundo ponto localiza-se na Avenida das Madalenas a cerca de 0,26 km do ponto emissor. Este é um ambiente relativamente próximo do emissor com diversos obstáculos. Nos pontos A12 e A26 mediram-se, respetivamente, níveis médios de sinal a -104,5 e -96,2 dBm.

RESULTADOS

Entre as áreas com pior qualidade de sinal pode-se referir a título de exemplo os pontos A11, A13 e A28, os quais se localizam respetivamente no Pináculo (-133,5 dBm), Poço da Neve (-139 dBm) e a Pontinha (-137,2 dBm). Para os pontos assinalados a vermelho no mapa da figura 4.23, procedeu-se ao estudo do seu perfil topográfico do terreno, no qual se avaliou o relevo do terreno entre o ponto emissor e o recetor no Google Earth Pro. Foi registada a altura do obstáculo com maior elevação. Os resultados são os apresentados na tabela 4.3. Por exemplo, para os pontos A11, A13 e A28 os obstáculos com maior elevação possuíam alturas de 20 m, 150 m e 35 m, respetivamente. O Anexo M apresenta as respetivas tabelas de dados completas.

Tabela 4.3 Estudo da geografia topográfica do terreno considerando o perfil de terreno e obstáculo com maior elevação.

Identificação ponto medição	Localização/pontos de referência	Obstáculo maior elevação (m)	Distância entre emissor/recetor (km)	Valor médio RSSI (dBm)	Taxa de sucesso por 10 pacotes transmitidos (%)
A6	M.PALACE	7,5	3,1	--130,6	100
A11	PINACULO	20	5,44	-133,5	80
A13	POÇO NEVE ARIEIRO	150	7,24	-139	30
A14	S.MARTINHO IGREJA	57,5	2,1	-138,5	40
A19	CONTINENTE VIVEIROS	36	0,49	-131,4	100
A22	R.31 JANEIRO	35	0,90	-135	100
A27	P.D.MARÍTIMO S.ANTÓNIO	15	1,78	-131,9	100
A28	PONTINHA	35	2,48	-137,2	50
A31	CAIS FUNCHAL	35	2,08	-128,8	60
A32	AV.MAR PROMENADE	25	2,52	-136	20
A36	ROT.FUNDOA	28,75	1,92	-133,9	90
A37	E.FUNDOA	85	1,27	-127,9	90
A40	R.EST.U.AMERICA	17,2	2,05	-131,4	100
A41	R.VIRTUDES L.MARITIMO	22,4	1,44	-137	10
A42	PILAR EDIFÍCIOS	26,4	0,72	-131,8	100
A43	R.VIRTUDES	22,4	1,13	-132,6	100

4.4.3. Dados recolhidos na página Web

Esta seção apresenta o sistema de monitorização remota após a sua construção. O *gateway* LoRa/Zigbee foi instalado no terraço da Universidade escolhendo um ponto mais elevado para tal. Quanto ao nó remoto era pretendido a sua instalação num ponto mais afastado da Universidade, para tal, realizaram-se testes das comunicações na Avenida das Madalenas junto à estrada. A escolha deste local foi determinada pelos testes de alcance explicados na seção 4.4.2, uma vez que havia sido obtido um nível médio de sinal superior a -100 dBm.

No entanto, a instalação no local foi condicionada por alguns fatores. Primeiro, quando foram realizados os testes das comunicações na periferia e traseira dos edifícios surgiram perdas de pacotes. Pois, os edifícios constituem obstáculos na propagação do sinal RF do emissor causando reflexões e acentuando o desvanecimento consideravelmente.

RESULTADOS

Outro fator foi o método da montagem do hardware, o qual foi desenvolvido numa placa de circuito pré-perfurada. Assim sendo, o nó remoto era sensível à ação de força mecânica constituindo um eventual fator de interrupção das comunicações, em tal caso, era necessário reiniciar o programa do microcontrolador para retomar as comunicações. Uma solução de maior robustez seria o desenvolvimento de uma placa de circuito impresso.

Por fim, não haviam recursos que facultassem a instalação do nó remoto num ponto de maior elevação no referido local. Visto que a instalação do protótipo num ponto de maior elevação era essencial para proporcionar melhor qualidade de sinal, assim como, melhores condições de segurança para o protótipo. Pelas razões mencionadas, foi determinado que o nó remoto seria instalado temporariamente no ponto onde havia-se instalado o nó local. Deste modo, o nó remoto encontrava-se no ponto inferior do edifício e a *gateway* no terraço da Universidade. Assim, estimou-se uma distância entre ambos os nós da ordem dos 40 metros. A figura 4.24 apresenta um gráfico contendo a variabilidade diária de CO nos dias 20-09-2019 (a vermelho) e 22-09-2019 (a verde).

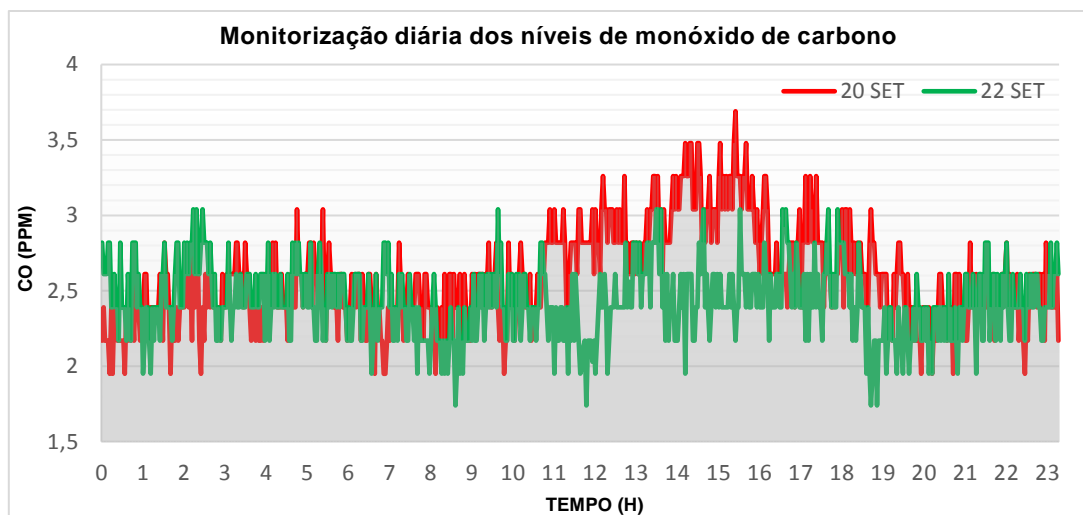


Figura 4.24 Monitorização diária dos níveis de monóxido de carbono.

É necessário referir o efeito da radiação solar nas medições de CO para o dia 20-09-2019, no qual os níveis de CO excedem os 3,5 ppm em certos instantes entre as 14:00 e as 16:00. Já no dia 22-09-2019, os níveis de CO mantiveram-se em torno dos 2,5 ppm. Registaram-se níveis limite nos 1,7 e 3 ppm, aproximadamente. No Anexo M seguem os respetivos dados do índice de radiação solar para os referidos dias.

O gráfico da figura 4.25 mostra a monitorização do nível das baterias para o terminal remoto (a vermelho) e a *gateway* (a verde). Daqui se destacam os baixos consumos energéticos proporcionados pelo reduzido *duty cycle* no terminal remoto durante as horas noturnas. Durante as horas diurnas, registou-se um carregamento relativamente eficiente para ambos os nós, concretamente, no terminal remoto foram atingidos os 3,85 V e na *gateway* os 4,1 V aproximadamente.

RESULTADOS

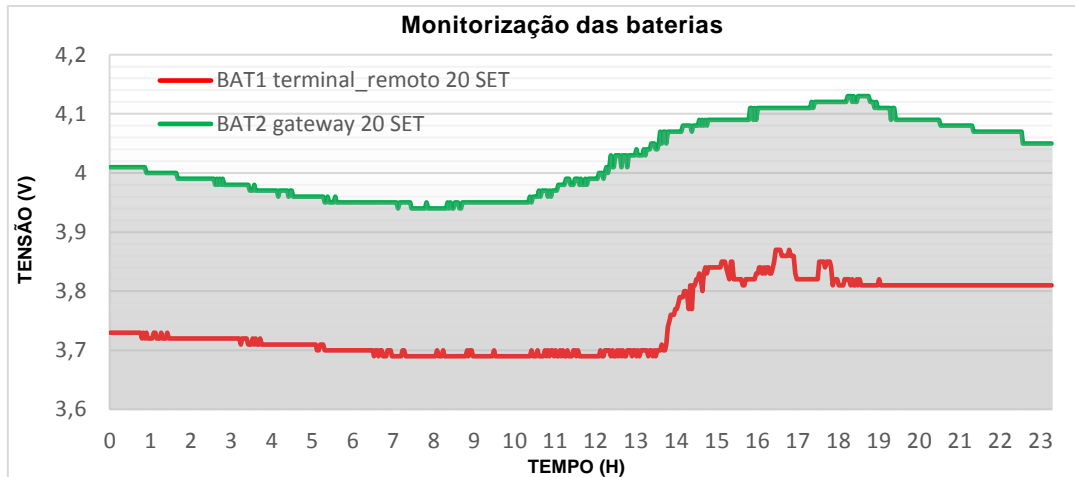


Figura 4.25 Monitorização das baterias.

A figura 4.26 apresenta dados de variabilidade de monóxido de carbono entre as 08:00 e as 12:00 para alguns dias. Estes são 20-09-2019 (a preto), 22-09-2019 (a verde), 24-09-2019 (a azul) e 26-09-2019 (a vermelho). A figura 4.27 apresenta os respetivos dados entre as 16:00 e as 20:00.

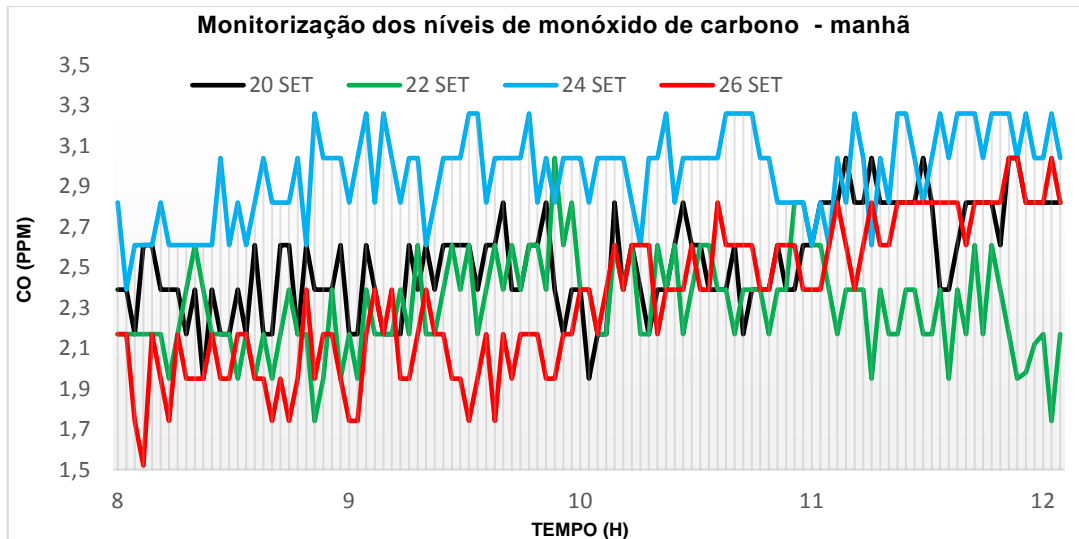


Figura 4.26 Monitorização dos níveis de monóxido de carbono entre as 08:00 e as 12:00.

Analisando o gráfico da figura 4.26, no dia 20-09-2019 é de destacar as 3 ppm atingidas próximo das 11:00, sendo as maiores concentrações registadas nesta manhã. No dia 22-09-2019, voltou-se a registar estes níveis próximo das 10:00. Em ambas as manhãs destes dias as concentrações de CO variaram a maior parte do tempo próximas dos 2,5 ppm. Já no dia 24-09-2019, registaram-se concentrações de CO mais elevadas em torno dos 3 ppm. Os valores de pico registados aproximavam-se dos 3,3 ppm em vários instantes durante esta manhã. O dia 26-09-2019 entre as 08:00 e as 10:00, os níveis de CO variavam principalmente em torno dos 2 ppm, em alguns instantes registaram-se valores de 1,7 e 2,3 ppm, estes ainda atingiram os 3 ppm próximo das 12:00.

RESULTADOS

Quanto ao gráfico da figura 4.27, nos dias 20-09-2019 e 24-09-2019 registaram-se 3,5 ppm próximo das 16:00. Como referido, no gráfico da figura 4.24 para o dia 22-09-2019 os níveis médios de CO variavam principalmente em torno de 2,5 ppm. No dia 26-09-2019, as concentrações de CO variaram a maior parte do intervalo de tempo em estudo entre os 2,7 e os 2,3 ppm. Dos dias apresentados, o nível mais significativo registado foi cerca de 3,5 ppm, através da expressão (3.3) são estimados aproximadamente 4 mg/m³. O limite inferior registado foi cerca de 1,5 ppm, o que corresponde a aproximadamente 1,72 mg/m³. Registaram-se principalmente variações das concentrações de CO entre os 2 e 3 ppm.

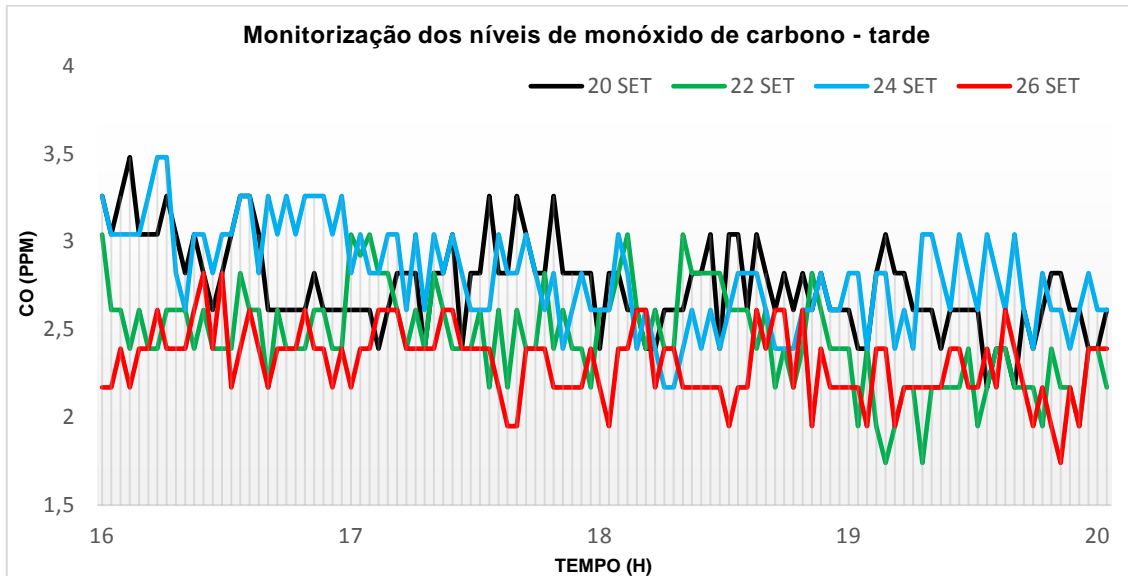


Figura 4.27 Monitorização dos níveis de monóxido de carbono entre as 16:00 e as 20:00.

4.5. Síntese de resultados

Nesta seção é apresentada uma breve síntese quanto aos dados analisados. Na seção 4.1, essencialmente, realizou-se um estudo comparativo dos sensores Alphasense B4 e Figaro TGS 5042. Durante o estudo laboratorial, verificou-se que o Alphasense B4 apresentava menor variabilidade de CO, quando comparado com o Figaro TGS 5042, assim como, valores mais baixos. As discrepâncias foram atribuídas a dois factores. O primeiro factor tratava-se das características de resolução e gama dinâmica dos sensores. Por exemplo, o máximo da gama dinâmica registado com o Alphasense B4 estava na ordem de 13 ppm. No caso do Figaro TGS 5042, verificou-se que este possuía uma menor resolução e maior gama dinâmica, a qual poderia chegar aos 10000 ppm, de acordo com o fabricante. Daqui se aferiu, a necessidade de calibrar este sensor para uma medição mais fiável das concentrações menores de gás. O segundo factor, atribuiu-se ao posicionamento dos sensores na bancada do laboratório em relação ao recipiente contendo o CO.

A seção 4.2 abordava o estudo do sensor Figaro face a fatores interferentes como a temperatura e a radiação solar. Ao longo deste estudo, verificou-se efetivamente que a radiação solar constituía a perturbação mais significativa para a monitorização de CO. De modo a minimizar a influência destas perturbações foi necessário construir uma estrutura anti-rradiação.

RESULTADOS

A seção 4.2 ainda abordou a monitorização pelo nó local. Do conjunto de dados analisados na seção 4.2.4, verificou-se efetivamente que em dias de maior intensidade de radiação solar, os dados medidos de CO poderiam oscilar desde os 2 até aos 3 ppm. A monitorização ambiental também envolveu o estudo da variabilidade diária e horas de maior afluência do tráfego urbano. Em dias de menor intensidade da radiação solar, estimou-se uma média diária em torno dos 0,2 ppm para o conjunto de dados de variabilidade diária. Nas horas de maior afluência do tráfego considerou-se os períodos entre as 08:00 e as 10:30 e as 16:30 e as 19:00, em alguns dias dos meses entre Janeiro e Maio de 2019. Daqui registaram-se em determinados instantes valores de pico entre os 4 e 5 ppm.

A seção 4.3 incidia na recolha de dados por um *gateway*. O conjunto de dados de estudo foram previamente monitorizados por um nó móvel com recetor GPS utilizado num projeto de mestrado anterior [54]. A informação de interesse a recolher continha dados temporais, coordenadas GPS, informações referentes à bateria do nó móvel e do sensor utilizado. A utilização dos dados temporais e coordenadas GPS permite recorrer à consulta numa ferramenta como o Google Earth. O que permite a consulta de locais de monitorização do nó móvel, assim como, o percurso percorrido pelo mesmo.

A seção 4.4 abordou o estudo da tecnologia de transmissão LoRa, na qual se realizaram estudos nomeadamente da eficiência de transmissão de pacotes e alcance de transmissão. O estudo da eficiência de transmissão de pacotes realizou-se com as configurações de médio e longo alcance respetivamente. Daqui, verificou-se a necessidade de um maior intervalo de tempo para o retorno do pacote ACK no modo de longo alcance. Já no estudo do alcance de transmissão, os casos mais otimistas a nível de RSSI destacam-se o miradouro do Pico Alto e a Avenida das Madalenas. O primeiro caso considera uma situação de linha de vista. O segundo caso, o recetor encontrava-se mais próximo do emissor instalado na Universidade, sendo este um cenário urbano. Os piores casos verificaram-se no Pináculo, Poço da Neve no Arieiro e a Pontinha no Funchal. A comunicação mais distante foi entre a Universidade e o Poço da Neve, numa distância de cerca de 7,24 km.

Por fim, abordou-se o sistema de monitorização remota, pelos motivos referidos na seção 4.4.3, o nó remoto foi instalado temporariamente no ponto onde havia-se instalado o nó local e a *gateway* no terraço da Universidade. A monitorização ambiental foi semelhante à realizada com o nó local. Deste modo, realizou-se monitorização do nível da bateria, variabilidade diária e estudo da afluência do tráfego urbano. À semelhança do nó local, este estudo considerou horas da manhã e da tarde. Os valores de pico registados encontravam-se na ordem de 3,5 ppm, tal aproxima-se dos 4 mg/m³. É necessário referir que devido a regulamentações de *duty cycle* impostas na utilização da tecnologia LoRa, este sistema operava com um período de amostragem consideravelmente mais elevado que o nó local.

5. Conclusões e trabalhos futuros

Este capítulo apresenta as conclusões reunidas no desenvolver do projeto, assim como, eventuais sugestões para trabalhos futuros.

5.1. Conclusões

Neste projeto desenvolveu-se uma rede de monitorização urbana de baixo custo, cuja finalidade, consistia em avaliar a qualidade do ar na cidade do Funchal. Esta permitiu a utilização das tecnologias rádio Zigbee e LoRa, na transmissão e recolha de dados. O utilizador pode realizar consulta *online* dos dados dos poluentes por local e data.

Começou-se por estudar brevemente, as características de funcionamento de diferentes tipos de sensores de gás. Foram abordadas regulamentações de relevância para o tema da qualidade do ar, assim como, métodos de medição. Realizou-se um breve estudo das estações de monitorização e do conceito das redes oportunistas. Em seguida, investigou-se com maior detalhe diferentes tecnologias de comunicações sem fios no âmbito das redes de sensores, as quais foram o Zigbee, o Wi-Fi, o Bluetooth e a LoRa. Por fim, foram apresentados alguns projetos de monitorização, cujo estudo focou-se nas tecnologias de sensores e de transmissão de dados e nos métodos de recolha e visualização dos mesmos.

Os protótipos desenvolvidos foram um nó de calibração, um nó local, um *gateway* de aquisição de dados e um sistema de monitorização remota. O nó de calibração fez uso de um sensor de gás pré-calibrado, o Alphasense B4. Este sensor é de elevada precisão e baixa afetação a outros gases. Este foi crucial no auxílio da calibração realizada para o sensor Figaro TGS 5042, o qual seria utilizado para os restantes nós. O protótipo foi testado no percurso pedestre no túnel que faz a ligação entre a Rotunda dos Ilhéus e a Avenida Gulbelkian.

Para o nó local, fez-se uso do sensor Figaro TGS 5042, o qual comparativamente ao Alphasense B4 é um sensor de menor custo e resolução. Tiveram-se alguns cuidados de modo a colmatar as limitações deste sensor. Um deles envolveu utilizar tecnologia SMD no desenvolvimento do circuito de condicionamento para minimizar problemas associados ao ruído térmico. Ainda foi realizada a calibração do sensor para a leitura das concentrações mais baixas de CO, servindo-se dos recursos dos ADC do Arduíno, no qual, limitou-se a tensão de referência interna nos 1.1 V. Por fim, avaliou-se o desempenho do sensor face a fontes de perturbação, como a temperatura e a radiação solar. Deste modo, constatou-se que a radiação solar causava as perturbações mais significativas na monitorização. Para minimizar estas perturbações foi construída uma estrutura anti-rradiação, na qual, o sensor foi encapsulado.

Devido ao nó local ser uma estação fixa, havia exigências de baixo consumo. Assim, implementou-se controlo embebido que atendia aos requisitos de monitorização ambiental periódica. Utilizou-se um painel solar para captação de energia. Os dados de interesse foram as tensões da bateria e do sinal elétrico para o sensor de CO, assim como, as concentrações de CO. A tecnologia rádio foi a Zigbee. O nó local foi instalado nas proximidades do laboratório de Telecomunicações junto à estrada.

CONCLUSÕES E TRABALHOS FUTUROS

A monitorização ambiental realizada envolveu o estudo da variabilidade diária de CO, análise da perturbação causada pela radiação solar, monitorização da tensão da bateria do nó e o estudo das horas de maior afluência do tráfego urbano. Neste último estudo, avaliaram-se alguns dados instantâneos de concentrações mais elevadas de CO (mg/m^3) e se avaliou o IQAr médio com base nas regulamentações da Agência Portuguesa do Ambiente.

O *gateway* de aquisição é uma solução de recolha de dados para os nós móveis. Os dados são extraídos de um cartão microSD, é realizado o seu tratamento que, eventualmente, pode conter coordenadas GPS. As restantes informações são dados temporais, dados referentes ao sensor de CO e ao nível da bateria. Para descarregar os dados na base de dados e apresentar os mesmos na Web, o módulo é conectado a um computador. A informação das concentrações poluentes, dos dados temporais e das coordenadas GPS é útil para criar um mapa através de uma ferramenta como o Google Earth. Com base nos dados referidos, é possível localizar pontos geográficos onde se registaram determinadas concentrações poluentes, analisar a qualidade do ar e estimar o percurso percorrido pelo nó móvel.

O sistema de monitorização remoto implicou desenvolver dois nós: um nó remoto de monitorização e um *gateway* LoRa. O nó remoto é uma estação fixa e possui um sensor Figaro, tal como, o nó local. Este também possui um painel solar e realiza monitorização periódica. No entanto, o controlo embebido desenvolveu-se com ênfase a conseguir o *duty cycle* a 1%. Os dados monitorizados foram o nível da bateria e o sinal elétrico do sensor Figaro. A comunicação rádio foi realizada pelo RFM9x LoRa. Uma das vertentes do controlo embebido deste nó foram as configurações rádio, nomeadamente, o modo de transmissão de longo alcance e o endereçamento cliente-servidor. A *gateway* LoRa foi desenvolvida especificamente para aguardar pela receção do pacote proveniente do terminal remoto, tal se conseguiu, com o endereçamento cliente-servidor. Uma vez recebido o pacote, seria realizado o ciclo de tratamento de dados e posterior inserção na rede de sensores da Universidade. Esta *gateway* servia-se das tecnologias rádio LoRa e Zigbee.

Foi realizado algum estudo quanto à utilização da tecnologia rádio LoRa, nomeadamente da eficiência de transmissão de pacotes e alcance de transmissão. O estudo da eficiência de transmissão de pacotes realizou-se com as configurações de médio e longo alcance respetivamente. Daqui, verificou-se a necessidade de um maior intervalo de tempo para o retorno do pacote ACK no modo de longo alcance. No estudo do alcance de transmissão, utilizou-se o modo de longo alcance. O nó emissor realizaria as transmissões a partir do terraço da Universidade e o recetor seria transportado para os locais enumerados no mapa do Google Earth na seção 4.4.2. Dos casos mais otimistas a nível de RSSI, destacaram-se o miradouro do Pico Alto e a Avenida das Madalenas. O primeiro caso considerava uma situação de linha de vista. O segundo caso, o recetor encontrava-se mais próximo do emissor, sendo este um cenário urbano. Os piores casos verificaram-se no Pináculo, Poço da Neve no Arieiro e a Pontinha no Funchal. A comunicação mais distante foi entre a Universidade e o Poço da Neve, numa distância de cerca de 7,24 km.

Devido aos motivos mencionados na seção 4.4.3, o nó remoto foi instalado temporariamente no ponto onde o nó local foi instalado anteriormente. A *gateway* foi instalada no terraço da Universidade. Os dois nós estavam distanciados a cerca de 40 metros entre si em condições de comunicação sem linha de vista.

A monitorização ambiental foi semelhante à realizada com o nó local. Deste modo, realizou-se monitorização do nível da bateria, variabilidade diária e estudo da afluência do tráfego urbano. Tal como o nó local para este estudo consideraram-se horas da manhã e da tarde. Os valores de pico registados encontravam-se na ordem de 3,5 ppm, tal aproxima-se dos 4 mg/m³. É necessário referir que devido a regulamentações de *duty cycle* impostas na utilização da tecnologia LoRa, este sistema operava com um período de amostragem consideravelmente mais elevado que o nó local. Como tal, o volume de dados era menor e a monitorização ambiental pelo terminal remoto era mais limitada, no entanto, os seus consumos de energia eram menores.

5.2. Trabalhos futuros

A tecnologia LoRa abre a possibilidade para uma monitorização remota mais alargada que considere uma maior variedade de parâmetros ambientais. Possíveis sugestões de projetos futuros a realizar seriam utilizar um maior número de RFM9x LoRa, maior variedade de sensores de gás (CO, CO₂, O₃, SO₂, NO₂) e de partículas poluentes (PM10, PM2.5), poluição sonora, temperatura e humidade. Isto implicaria um conjunto de estações remotas onde cada qual iria monitorizar o respetivo parâmetro ambiental de interesse.

Uma possível solução como *gateway* poderia incluir um Raspberry Pi e recetor LoRa. Um método de comunicação entre as estações remotas e a *gateway* LoRa poderia ser a utilização de vários canais de comunicação. Cada estação remota poderia se servir de um canal dedicado à comunicação com a *gateway* em *uplink*. Tal implicaria mecanismos de endereçamento para proteger o *payload* dos pacotes, de modo que apenas a *gateway* e a estação remota de interesse pudessem detetar as mensagens, no entanto, recomenda-se um estudo mais cuidado. Adicionalmente, poderia ser utilizada a comunicação *downlink* para confirmar a entrega dos dados através do pacote ACK.

Outra eventual sugestão de comunicação entre a *gateway* e as estações remotas poderia ser a utilização de mecanismos TDMA (*Time Division Multiple Access*). A monitorização periódica de cada estação remota seria controlada por um mecanismo que iria detetar *beacons* RF transmitidos a partir da *gateway*, o que iria acordar o microcontrolador e o módulo de comunicação para o ciclo ativo. Este mecanismo seria uma possibilidade para o sincronismo temporal necessário no TDMA. A *gateway* iria permanecer no estado ativo.

Do lado do servidor de base de dados, poderia se tirar partido da plataforma Web existente que recorre à utilização do protocolo Zigbee, ou, alternativamente, desenvolver uma nova plataforma Web capaz de receber diretamente dados no protocolo LoRaWAN ou outra tecnologia de comunicações sem fios. Adicionalmente, esta plataforma Web poderia permitir o acesso por aplicações android para monitorização ambiental.

6. Referências Bibliográficas

- [1] N. Castell *et al.*, «Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?», *Environment International*, vol. 99, pp. 293–302, Fev. 2017, doi: 10.1016/j.envint.2016.12.007.
- [2] H. Concentration e K. E. Y. Specifications, «CO-B4 Carbon Monoxide Sensor 4-Electrode», 2017.
- [3] World Air Quality Index Project, «The Shinyei experiment», 2011. [Em linha]. Disponível em: <http://aqicn.org/sensor/shinyei/es/>. [Acedido: 10-Fev-2020].
- [4] N. Barsan, U. Weimar, e T. Chemistry, «Fundamentals of Metal Oxide Gas Sensors», *MCS 2012 – The 14th International Meeting on Chemical Sensors*, pp. 618–621, 2012, doi: 10.5162/IMCS2012/7.3.3.
- [5] J. W. Gardner, «Electrical conduction in solid-state gas sensors», *Sensors and Actuators*, vol. 18, n. 3–4, pp. 373–387, 1989, doi: 10.1016/0250-6874(89)87043-X.
- [6] N. R. Stradiotto, H. Yamanaka, e M. V. B. Zanoni, «Electrochemical sensors: A powerful tool in analytical chemistry», *Journal of the Brazilian Chemical Society*, vol. 14, n. 2, pp. 159–173, 2003, doi: 10.1590/S0103-50532003000200003.
- [7] Agência Portuguesa do Ambiente, «Índice de Qualidade do Ar | Relatório do Estado do Ambiente», 2019. [Em linha]. Disponível em: <https://rea.apambiente.pt/content/índice-de-qualidade-do-ar>. [Acedido: 10-Fev-2020].
- [8] QualAR, «QualAR - Qualidade do AR», *a Rede De Medição*, 2019. [Em linha]. Disponível em: <https://qualar.apambiente.pt/node/rede-de-medicao-da-qualidade-do-ar>. [Acedido: 10-Fev-2020].
- [9] APA, «Avaliação da Qualidade do Ar na RLVT - Ambiente - CCDR LVT», 2016. [Em linha]. Disponível em: <http://www.ccdr-lvt.pt/pt/avaliacao-da-qualidade-do-ar-na-rlvt/8085.htm>. [Acedido: 10-Fev-2020].
- [10] M. Sofia e R. De Azevedo, «Monitorização da Qualidade do Ar utilizando Redes Sem Fios Oportunistas», Faculdade de Engenharia da Universidade do Porto, 2010.
- [11] D. Dias e L. Costa, «Análise da Capacidade de Dados de uma Rede de Ônibus Urbanos», *Anais de XXXIV Simpósio Brasileiro de Telecomunicações*, pp. 358–362, 2016, doi: 10.14209/sbtr.2016.46.
- [12] J. Alves, M. Petronillio, W. Machado, e M. de O. Barros, «Gerenciamento de Redes de Sensores Sem Fio», 2013.
- [13] V. Hejlová e V. Voženílek, «Wireless Sensor Network Components for Air Pollution Monitoring in the Urban Environment: Criteria and Analysis for Their Selection», *Wireless Sensor Network*, vol. 05, n. 12, pp. 229–240, 2013, doi: 10.4236/wsn.2013.512027.
- [14] E. J. K. Neto, M. F. L. Pereira, e R. B. de O. Pereira, «Utilizando o transceptor NRF24L01 em redes de sensores sem fio operando sobre TCP/IP», *Anais da Escola Regional de Informática da Sociedade Brasileira de Computação (SBC) – Regional de Mato Grosso*, vol. 6, n. 0, pp. 151–157, 2015, doi: 10.13140/RG.2.1.2477.3842.
- [15] E. E. J. M. Mário Saleiro, «ZigBee - uma abordagem prática».

REFERÊNCIAS BIBLIOGRÁFICAS

- [16] P. Yi, A. Iwayemi, e C. Zhou, «Building Automation Networks for Smart Grids», *International Journal of Digital Multimedia Broadcasting*, vol. 2011, pp. 1–12, 2011, doi: 10.1155/2011/926363.
- [17] M. Nascimento, «Acesso aos Meios em WLANs: CSMA/CA – CCNA», *DLTech do Brasil*, 2012. [Em linha]. Disponível em: <http://www.dltec.com.br/blog/cisco/acesso-aos-meios-em-wlans-csmaca-ccna/>. [Acedido: 10-Fev-2020].
- [18] «rssf/Protocolos». [Em linha]. Disponível em: https://www.gta.ufrj.br/grad/11_1/rssf/Protocolos.html. [Acedido: 10-Fev-2020].
- [19] X. Znet, X. Z. Oem, e R. F. Modules, «XBee-2.5-Manual».
- [20] S. Tozlu, M. Senel, Wei Mao, e A. Keshavarzian, «Wi-Fi enabled sensors for internet of things: A practical approach», *IEEE Communications Magazine*, vol. 50, n. 6, pp. 134–143, Jun. 2012, doi: 10.1109/MCOM.2012.6211498.
- [21] J. R. Kettimuthu e S. Muthuknshnan, «Is bluetooth suitable for large-scale sensor networks?», *Proceedings of the 2005 International Conference on Wireless Networks, ICWN'05*, pp. 448–454, 2005.
- [22] LinkLabs, «New Industrial Internet of Things Products Wireless Connectivity is Exploding», 2016.
- [23] C. Gomez, J. Oller, e J. Paradells, «Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology», *Sensors*, vol. 12, n. 9, pp. 11734–11753, Ago. 2012, doi: 10.3390/s120911734.
- [24] M. C. Bor, U. Roedig, T. Voigt, e J. M. Alonso, «Do LoRa Low-Power Wide-Area Networks Scale?», em *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16*, 2016, pp. 59–67, doi: 10.1145/2988287.2989163.
- [25] J. C. Liando, A. Gamage, A. W. Tengourtius, e M. Li, «Known and Unknown Facts of LoRa», *ACM Transactions on Sensor Networks*, vol. 15, n. 2, pp. 1–35, Abr. 2019, doi: 10.1145/3293534.
- [26] J. Petäjajarvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen, e J. Linatti, «Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring», *International Journal of Wireless Information Networks*, vol. 24, n. 2, pp. 153–165, Jun. 2017, doi: 10.1007/s10776-017-0341-8.
- [27] P. S. 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)*, 2017, vol. 2017-Decem, n. December, pp. 1–6, doi: 10.1109/SCVT.2017.8240313.
- [28] A. Augustin, J. Yi, T. Clausen, e W. Townsley, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things», *Sensors*, vol. 16, n. 9, p. 1466, Set. 2016, doi: 10.3390/s16091466.
- [29] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, e G. Andrieux, «Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN», *Sensors*, vol. 18, n. 7, p. 2104, Jun. 2018, doi: 10.3390/s18072104.
- [30] G. Description, K. E. Y. P. Features, e O. Information, «Sx1276/77/78/79», 2015.
- [31] B. C. Fargas e M. N. Petersen, «GPS-free geolocation using LoRa in low-power WANs», em *2017 Global Internet of Things Summit (GloTS)*, 2017, pp. 1–6, doi: 10.1109/GIOTS.2017.8016251.
- [32] L. Casals, B. Mir, R. Vidal, e C. Gomez, «Modeling the Energy Performance of LoRaWAN», *Sensors*, vol. 17, n. 10, p. 2364, Out. 2017, doi: 10.3390/s17102364.

REFERÊNCIAS BIBLIOGRÁFICAS

- [33] C. Borrego *et al.*, «Assessment of air quality microsensors versus reference methods: The EuNetAir joint exercise», *Atmospheric Environment*, vol. 147, n. 2, pp. 246–263, Dez. 2016, doi: 10.1016/j.atmosenv.2016.09.050.
- [34] D. N. Gomes e S. Sim, «Mobile Station for Air Quality Mapping Sensor Network», *Development*, n. November, p. 92, 2011.
- [35] Fábio Gameiro, «Outline of the URBISNET network architecture», 2011.
- [36] S. Brienza, A. Galli, G. Anastasi, e P. Bruschi, «A Low-Cost Sensing System for Cooperative Air Quality Monitoring in Urban Areas», *Sensors*, vol. 15, n. 6, pp. 12242–12259, Mai. 2015, doi: 10.3390/s150612242.
- [37] N. Bebelaar *et al.*, «Monitoring urban environmental phenomena through a wireless distributed sensor network», *Smart and Sustainable Built Environment*, vol. 7, n. 1, pp. 68–79, Abr. 2018, doi: 10.1108/SASBE-10-2017-0046.
- [38] Sistemas de Sensores e Captação de Energia, «Armazenamento e visualização dos dados da rede ZigBee».
- [39] «Bem-vindo ao NetBeans». [Em linha]. Disponível em: <https://netbeans.org/>. [Acedido: 10-Fev-2020].
- [40] F. S. J. Amândio Azevedo, «Redes de sensores sem fio - pela Faculdade de Ciências e Engenharia Exatas da Universidade da Madeira». [Em linha]. Disponível em: <http://www.cee.uma.pt/hardlab/wsngroup/wsn/>. [Acedido: 23-Abr-2020].
- [41] S. W. List, «Using the Arduino Pro Mini 3.3V». [Em linha]. Disponível em: <https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v/all>. [Acedido: 10-Out-2019].
- [42] H. Concentration, «Individual Sensor Board (ISB) Alphasense Alphasense B4 4-Electrode Gas Sensors», 2017.
- [43] P. Adjustable, S.-D. V. Regulator, P. Adjustable, e S.-D. V. Regulator, «POLOLU ADJUSTABLE 4-12V STEP-UP/STEPDOWN VOLTAGE REGULATOR S18V20ALV /POLOLU ADJUSTABLE 9-30V STEP-UP/STEPDOWN VOLTAGE REGULATOR S18V20AHV USER'S GUIDE».
- [44] A. Graphic *et al.*, «Arduino TFT LCD Screen». [Em linha]. Disponível em: <https://www.arduino.cc/en/Guide/TFT>. [Acedido: 10-Out-2019].
- [45] P. I. N. Connections e M. Diagram, «NCP1402 Micropower Switching Regulator MARKING DIAGRAM», 2014.
- [46] A. M. Ratings, «TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8 PACKAGE / ORDER INFORMATION Complete Charge Cycle (1000mAh Battery)».
- [47] S. J. Hu, «Lithium-ion Battery DATA SHEET», 2010.
- [48] National Instruments, «Ultiboard - National Instruments». [Em linha]. Disponível em: <https://www.ni.com/es-cr/shop/select/ultiboard>. [Acedido: 10-Fev-2020].
- [49] «1966535.pdf».
- [50] ON Semiconductor, «1N5333B Series 5 Watt Surtetic t 40 Zener Voltage Regulators AXIAL LEAD readily solderable MAXIMUM LEAD TEMPERATURE FOR SOLDERING PURPOSES »:, 2008.
- [51] figaro-sensor, «TGS 5042 -for the detection of Carbon Monoxide», 2014.
- [52] D. C. Design e M. Process, «Application notes for tgs5042».

REFERÊNCIAS BIBLIOGRÁFICAS

- [53] «Referência do Arduino». [Em linha]. Disponível em: <https://www.arduino.cc/reference/en/>. [Acedido: 11-Fev-2020].
- [54] D. De Mestrado, «Beto André Fernandes da Silva MESTRADO EM ENGENHARIA ELETROTÉCNICA-TELECOMUNICAÇÕES Sistema de Monitorização Ambiental Urbana com Redes de Sensores Sem Fios», Universidade da Madeira, 2019.
- [55] N. Million e V. One, «Understanding Units of Measurement», *October*, n. October, pp. 1–3, 2008.
- [56] L. T. Corporation, «Rail-to-Rail Input / Output Amplifiers».
- [57] DIGI, «ZigBee RF Modules», 2014.
- [58] Digi International Inc., «XBee OEM RF Modules - Datasheet», *Product Datasheet*, vol. Di, n. 866, pp. 1–2.
- [59] «XCTU - Plataforma de configuración Next Gen para soluciones XBee / RF - Digi International». [Em linha]. Disponível em: <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>. [Acedido: 10-Fev-2020].
- [60] Sistemas de Sensores e Captação de Energia, «Rede de sensores com módulos XBee».
- [61] Apache Friends, «XAMPP Installers and Downloads for Apache Friends», <https://www.apachefriends.org/index.html>, 2017. [Em linha]. Disponível em: <https://www.apachefriends.org/es/index.html>. [Acedido: 10-Fev-2020].
- [62] «Cartão Micro SD Mini Módulo de cartão SD TF LC Technology». [Em linha]. Disponível em: <https://dev.techmaze.ae/micro-sd-card-mini-sd-card-module-lc-technology->. [Acedido: 10-Fev-2020].
- [63] «Solar Panel 6V 2W | Solar». [Em linha]. Disponível em: <https://www.ptrobotics.com/solar/1628-solar-panel-6v-2w.html>. [Acedido: 10-Fev-2020].
- [64] P. M. Utc, «USB , DC & Solar Lipoly Charger Voltaic Panels with 1 . 3mm Connectors Other 6V Solar Panels If You Have a Pre-2013 Solar Charger Using the Charger Design Notes», 2018.
- [65] Adafruit Industries, «Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts», 2017.
- [66] «Overview | Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts | Adafruit Learning System». [Em linha]. Disponível em: <https://learn.adafruit.com/adafruit-rfm69hwc-and-rfm96-rfm95-rfm98-lora-packet-radio-breakouts/overview>. [Acedido: 10-Fev-2020].
- [67] «RadioHead: RH_RF95 Class Reference». [Em linha]. Disponível em: https://www.airspayce.com/mikem/arduino/RadioHead/classRH__RF95.html. [Acedido: 10-Fev-2020].
- [68] Autocore Robótica, «Sensor de Temperatura NTC 10k 3mm». [Em linha]. Disponível em: <https://www.autocorerobotica.com.br/sensor-de-temperatura-ntc-10k-3mm>. [Acedido: 10-Fev-2020].
- [69] E. Ag, «NTC thermistors for temperature measurement», 2013.
- [70] «Versões do Earth – Google Earth». [Em linha]. Disponível em: <https://www.google.com/earth/versions/>. [Acedido: 10-Fev-2020].
- [71] «NeoPixels: Adafruit Industries, Unique & fun DIY electronics and kits», 2017. [Em linha]. Disponível em: <https://www.adafruit.com/category/168>. [Acedido: 11-Fev-2020].

REFERÊNCIAS BIBLIOGRÁFICAS

- [72] «para ANDROID». [Em linha]. Disponível em: <https://www.viewranger.com/en-US/product-overview/product-features/for-android>. [Acedido: 10-Fev-2020].
- [73] B. P. Crow e T. M. Corporation, «IEEE 802.11 Wireless Local Area Networks», n. September, pp. 116–126, 1997.
- [74] «IEEE 802.15 Bluetooth - Camada Física e Camada de Acesso ao Meio». [Em linha]. Disponível em: https://www.gta.ufrj.br/grad/09_1/versao-final/bluetooth/Page323.htm. [Acedido: 28-Fev-2020].
- [75] O. Kasten e M. Langheinrich, «First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network», *Workshop on Ubiquitous Computing and Communications, PACT 2001, Barcelona, Spain, September 8-12, 2001*, n. 00, 2001.
- [76] P. Bonnet, A. Beaufour, M. B. Dydensborg, e M. Leopold, «Bluetooth-based sensor networks», *ACM SIGMOD Record*, vol. 32, n. 4, pp. 35–40, Dez. 2003, doi: 10.1145/959060.959067.
- [77] Y. Bo, «Bluetooth: Technology and Applications», 2017.
- [78] «BLE Advertising packet format | BLE Data packet format». [Em linha]. Disponível em: <https://www.rfwireless-world.com/Terminology/BLE-Advertising-and-Data-Packet-Format.html>. [Acedido: 28-Fev-2020].
- [79] J. Yin, Z. Yang, H. Cao, T. Liu, Z. Zhou, e C. Wu, «A Survey on Bluetooth 5.0 and Mesh», *ACM Transactions on Sensor Networks*, vol. 15, n. 3, pp. 1–29, Ago. 2019, doi: 10.1145/3317687.

7. Anexos

Anexo A - Wi-Fi – estudo como tecnologia de recolha de dados em redes de sensores

Modelo de operação – nó com tecnologia de comunicação sem fios Wi-Fi

A figura 7.1 apresenta o processo de operação realizado numa rede de sensores baseada em Wi-Fi.

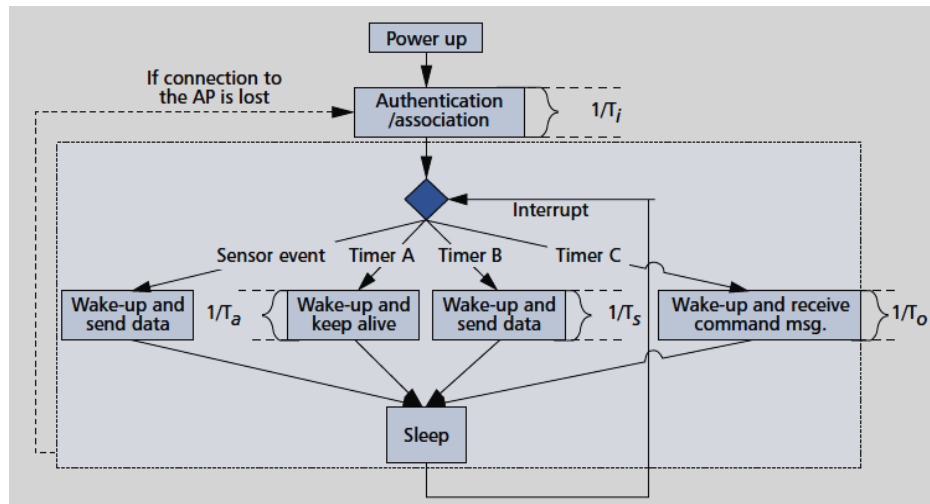


Figura 7.1 Operação típica para um sensor com tecnologia Wi-Fi [20].

O Wi-Fi também se serve dos modos de baixo consumo, a finalidade é reduzir os consumos de energia das estações de monitorização, deste modo, desativam-se os módulos rádio. Enquanto a estação de monitorização está no modo de baixo consumo, o *Access Point* armazena mensagens temporariamente. Posteriormente, essas mensagens são transmitidas à estação através de *beacons*. Estes são portadoras de sinalização, as quais, são transmitidas periodicamente com o objetivo de sinalizar a existência de mensagens.

O que implica a necessidade de haver uma janela temporal específica para o nó retornar ao estado ativo. O objetivo é escutar o canal e receber a mensagem sinalizada pelo *beacon*, no entanto, é necessário sincronismo temporal para evitar perdas de pacotes. Como tal, impõe-se que a janela temporal de escuta para o nó deva ter uma duração que seja um múltiplo inteiro do intervalo de transmissão do *beacon*. Se no intervalo de escuta não há mensagens em espera, o nó retorna ao estado adormecido. No ciclo de escuta seguinte a estação volta a verificar a eventual existência de mensagens.

Se houver mensagens em espera e forem mensagens *broadcast/multicast*, o *Access Point* irá transmitir a mensagem assim que o *beacon* sinalizar, o nó fica aguardando pela mensagem. Para uma mensagem *unicast*, o nó destinatário envia ao *Access Point* um pacote de sinalização, este conseqüentemente, irá transmitir a respetiva mensagem.

Consumos de corrente e taxa de transmissão

Nas aplicações de sensores baseadas no *duty cycle*, a corrente consumida é uma variável importante a considerar na autonomia das baterias. Durante a maioria do tempo, o nó encontra-se no estado adormecido. No instante que ocorra um determinado evento este transita para o estado ativo. Dois requisitos importantes em aplicações Wi-Fi focadas no baixo consumo são reduzir a corrente nos estados ativo e adormecido e reduzir o *duty cycle*.

Idealmente, a taxa de transmissão máxima do Zigbee é 250 kbps, no Wi-Fi esta pode variar de 1 até 54 Mbps. As taxas de dados mais elevadas do Wi-Fi têm como vantagem o período ativo ter menor duração para transmitir uma determinada mensagem, pois a energia consumida por bit é menor. Deste modo, os consumos energéticos no ciclo ativo são menores.

Tipos de pacotes de dados

No Wi-Fi, o método de acesso ao canal mais usado é o DCF (*Distributed Coordination Function*). Este serve-se do protocolo CSMA/CA para aceder ao canal de transmissão. As tramas MAC no Wi-Fi são caracterizadas em três tipos. O primeiro é as tramas de dados utilizadas para transmitir mensagens de dados. O segundo são as tramas de controlo, as quais, destinam-se a controlar o acesso ao canal de comunicação, por exemplo, os pacotes RTS, CTS, ACK. Por fim, as tramas de gestão, a sua finalidade é trocar informação de gestão.

Na transmissão de tramas de dados quando o recetor não é capaz de as detetar, há a possibilidade de as retransmitir “n” vezes. No entanto, se após esse número de tentativas de transmissão o recetor continuar a não detetar o pacote, este é descartado. As transmissões bem-sucedidas ou falhadas são reportadas aos protocolos das camadas superiores.

Formato das tramas

A figura 7.2 apresenta o formato de uma trama MAC Wi-Fi, a qual é composta por cabeçalho (*MAC Header*), conteúdo (*Frame Body*) e o campo *Frame Check Sequence* (FCS). O campo *Frame Control* está dividido em vários subcampos, este ocupa 2 bytes.

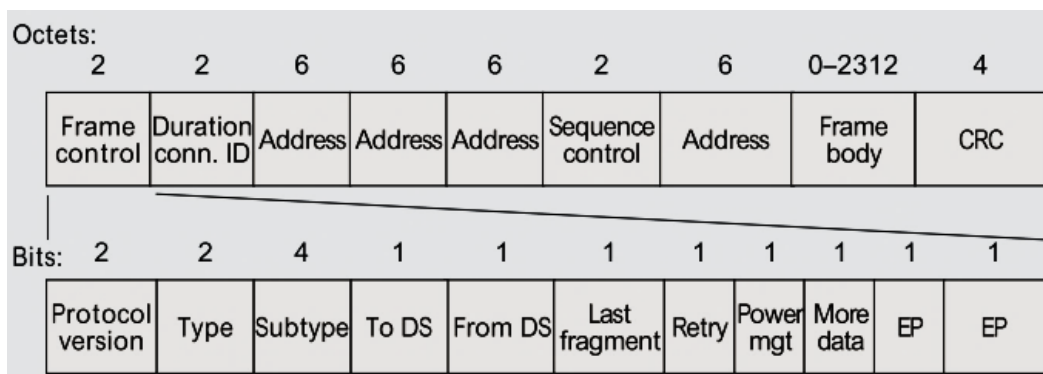


Figura 7.2 Formato standard para uma trama MAC no Wi-Fi [73].

O campo versão do protocolo (*Protocol Version*) ocupa 2 bits, este considera as evoluções do padrão IEEE 802.11. A sua função é assegurar compatibilidade entre as versões e os dispositivos Wi-Fi. Os campos tipo/subtipo (*type/subtype*) ocupam respetivamente 2 e 4 bits. A função do campo tipo é caracterizar o tipo de trama (gestão, controlo, dados).

ANEXOS

O campo subtipo é de codificação e este serve-se de uma sequência de 4 bits. Esta codificação permite associar diferentes operações na rede, por exemplo, pedidos de associação, envio de dados ou o pacote ACK. O campo To DS (*To Distribution System*), o bit sinaliza “1” quando a trama se destina ao sistema de distribuição, por exemplo, uma transmissão de uma estação a um ponto de acesso à rede Wi-Fi, nos restantes casos sinaliza “0”. O campo From DS (*From Distribution System*), o bit sinaliza “1” quando a trama provém do ponto de acesso. Nos restantes casos sinaliza “0”, por exemplo se ambos os campos To DS/From DS sinalizam “0”, isto se trata de uma comunicação direta entre duas estações.

O campo *Last Fragment*, o seu bit sinaliza o estado lógico “1”, quando há fragmentos a transmitir. O campo *Retry*, o respetivo bit sinaliza “1”, quando o fragmento a ser enviado é uma retransmissão. O campo *Power mgt* (*Power Management*), quando o seu bit se encontra no estado “1”, este sinaliza que a estação que transmitiu a trama está no modo de gestão de energia. O campo *More Data* é reservado para o modo de gestão energética, este pode ser usado pelo ponto de acesso para sinalizar a uma determinada estação que tramas adicionais estão em espera. Os últimos 2 bits EP pertencentes ao campo *Frame Control* são bits de sinalização. O primeiro destina-se ao algoritmo de codificação WEP para o corpo da trama. O segundo reporta a transmissão da trama utilizando a classe de serviço estritamente ordenada (*Strictly-Ordered service class*).

O campo “*Duration conn. ID*” ocupa 2 bytes, este permite indicar a duração do uso do canal de transmissão. O campo de endereços pode conter até 3 endereços, cada qual ocupa 6 bytes. O campo *Sequence Control* a sua finalidade é distinguir os vários fragmentos da mesma trama. Este é composto por dois subcampos estes são o número do fragmento e o número da sequência, os dois subcampos permitem a eventual reordenação dos fragmentos. Segue-se um campo de endereços com 6 bytes. O campo *Frame Body* destina-se a encapsular o *payload*, o qual pode conter até 2312 bytes, geralmente a mensagem é encriptada pelo algoritmo WEP (*Wired Equivalent Privacy*). Por fim, o campo CRC, o qual, ocupa 4 bytes, a sua finalidade é verificar a integridade da trama.

Anexo B - Bluetooth – estudo como tecnologia de recolha de dados em redes de sensores

Estrutura de rede - Bluetooth clássico

A rede Bluetooth é estruturada em *piconets*, cada qual pode servir até 8 dispositivos no estado ativo, sendo composta por um mestre e escravos. O mestre gere a *piconet*, este implementa controlo centralizado no acesso ao canal. A comunicação é estritamente entre mestre/escravo e é bidirecional. Nas áreas geográficas onde há várias *piconets* próximas entre si, as suas respetivas regiões de cobertura de rede vão se sobrepor espacialmente, estas são as *scatternets*. A figura 7.3 apresenta uma *piconet* e uma *scatternet*.

Um escravo pertencente a uma *piconet* “x” pode criar uma *piconet* “y” e assumir-se como mestre. No entanto, o mestre da *piconet* “y” permanece como escravo na *piconet* “x”. Deste modo, as unidades Bluetooth podem ser escravas em duas ou mais *piconets*, alternativamente podem ser mestres numa *piconet* e escravas nas restantes onde estejam conectadas. Uma unidade conectada a duas *piconets* pode servir de *gateway* entre as mesmas.

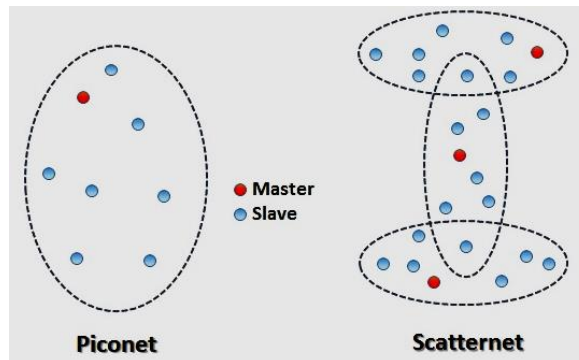


Figura 7.3 Apresentação de uma *piconet* (esquerda) e *scatternet* (direita) [74].

Piconets em redes de sensores

Em redes de sensores Bluetooth com elevada densidade de nós há uma forte possibilidade dos escravos necessitarem de comunicar entre si. No entanto, a topologia de rede do Bluetooth clássico obriga a envolver o mestre nas comunicações, o que aumenta o tráfego de rede e os consumos de energia. Algumas estratégias a adotar consistem num dos escravos trocar funções com o mestre ou configurar uma nova *piconet*. Estas soluções reduzem os consumos energéticos, mas continuam a aumentar o tráfego de rede [21].

Nas aplicações de sensores de maior densidade não é adequado utilizar conexões com 8 nós. Neste caso, a solução consiste em utilizar multiplexagem por divisão temporal, o que oferece a possibilidade de realizar transmissões *full-duplex*. É atribuído um conjunto de *time slots*, cada qual com duração de 625 μ s, os quais são encapsulados numa trama Bluetooth. As transmissões mestre-escravo começam num *time slot* par, já as transmissões escravo-mestre tem início num *time slot* ímpar.

A configuração de *piconets* adicionais para aplicações de sensores implica a necessidade de *gateways*, tal oferece a possibilidade às unidades de operar em diferentes *piconets*. Deste modo, uma unidade pode estar ativa numa *piconet* e inativa na outra num determinado instante.

Por outro lado, configurar *piconets* adicionais na mesma área geográfica pode causar interferências entre sinais RF. Uma vez que as *piconets* vizinhas não coordenam entre si os seus padrões de salto em frequência, o que aumenta a interferência intersimbólica (ISI *intersymbol interference*), isto traduz-se em perdas de pacotes de dados e obriga a haver retransmissões. Deste modo, o aumento do número de *piconets* na mesma área, aumenta a probabilidade de ocorrência de colisões e consequentes retransmissões, o que degrada as taxas de transmissão e aumenta a latência.

Estados de operação

O estado pré-definido numa unidade Bluetooth é o modo *standby*, este é um modo de baixo consumo. O seu *hardware* com exceção do relógio interno está inativo, neste modo não é possível conectar-se a outros dispositivos. As conexões apenas podem-se estabelecer no modo *connect*, este pode ser configurado nos estados: *active*, *sniff*, *hold* e *park* [75].

No estado ativo (*active*) a unidade participa ativamente no canal, as transmissões de dados são quase instantâneas, no entanto, os consumos de energia são relativamente elevados. Os restantes modos a referir são de baixo consumo. Os consumos energéticos são minimizados através da redução do *duty cycle* para as unidades que integram a *piconet*. No estado *sniff*, o escravo escuta o canal em intervalos de tempo específicos, os quais, são agendados pelo mestre.

Existe a opção de colocar as conexões da *piconet* no estado *hold*, neste modo cada participante do canal incluindo o mestre possui uma janela temporal específica para entrar no modo adormecido. Inicialmente, o mestre e os escravos negociam entre si um período do tempo no qual deverão retornar ao estado ativo. O estado *hold* é útil para realizar tarefas paralelas como atender às necessidades de unidades de *piconets* vizinhas ou escutar o canal para detetar outras unidades.

O estado *park* é adequado para os escravos que não necessitam de participar ativamente no canal, mas que, no entanto, necessitam de se manterem conectados. As unidades no estado *park* não são consideradas membros ativos da *piconet*, além dos até 8 participantes ativos ainda pode haver até 255 unidades configuradas no estado *park* dentro da mesma.

Há uma relação entre os consumos energéticos e o tempo de resposta da unidade. Reduzir o *duty cycle*, minimiza a corrente consumida, no entanto, aumenta a duração do tempo de resposta. Os modos de baixo consumo são eficientes com padrões de transmissão de dados periódicos, pois as unidades respeitam um determinado agendamento. Em contrapartida, para transmissões de dados instantâneas a solução é o estado ativo que apesar dos maiores consumos evita a perda de pacotes.

Consulta

A consulta (*inquiry*) do Bluetooth consiste na pesquisa de outras unidades nas vizinhanças. Dependendo da informação trocada podem se estabelecer conexões. Os elementos da consulta são o inquiridor e a unidade inquirida, os quais, estão configurados nos modos *inquiry* e *inquiry scan* respetivamente [76].

O inquiridor inicia a sequência de ações na consulta, este envia continuamente mensagens de interrogação *broadcast* para o canal. O modo *inquiry* atua durante um intervalo temporal pré-estabelecido até que se detetem “n” unidades ou essa janela temporal expire. Apenas as unidades inquiridas enviam mensagem de resposta ao inquiridor, enquanto as restantes limitam-se a escutar o canal. Para consultas periódicas, o intervalo entre duas janelas temporais *inquiry* é determinado aleatoriamente. Isto evita que duas unidades sincronizem as suas operações *inquiry* durante a fase de bloqueio da consulta.

O inquiridor transmite as mensagens de interrogação repetidamente em diferentes frequências. A unidade inquirida ao efetuar o *inquiry scan* têm de escutar uma frequência de cada vez durante um determinado intervalo de tempo, em seguida, muda-se para outra frequência e repete o processo de escuta sucessivamente. Na consulta o aumento dos consumos energéticos é inevitável, em particular para o inquiridor que acarreta as maiores exigências energéticas.

Paginação

No estabelecimento de conexões usa-se a paginação (*paging*), tal requer conhecer o endereço da unidade Bluetooth a conectar. Possuir informação referente ao relógio interno dessa unidade contribui para acelerar a conexão. A unidade que estabelece o emparelhamento assume-se como mestre na ligação. O mestre realiza um padrão de salto em frequência e a unidade escrava deverá replicar essa sequência.

O inconveniente da paginação é a sua latência, o que é limitativo para aplicações de sensores. Pois até ao momento da paginação cada unidade Bluetooth atua com o seu respetivo padrão de salto em frequência. A latência também pode ser causada por exigências de baixo consumo ou eventuais conexões antecedentes.

Bluetooth clássico – formato da trama

A figura 7.4 apresenta o formato de uma trama Bluetooth, esta é composta por campo de código de acesso (*access code*), um cabeçalho (*packet header*) e *payload*.

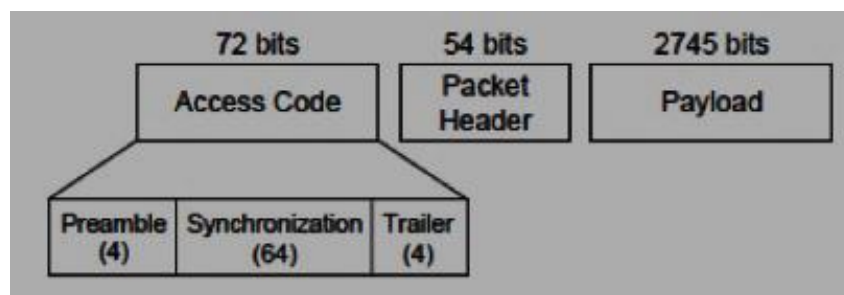


Figura 7.4 Formato de uma trama Bluetooth [77].

O campo do código de acesso ocupa 72 bits, está subdividido em preâmbulo (4 bits), palavra de sincronização (*sync word* - 64 bits) e *Trailer* (4 bits). O código de acesso serve para identificar o pacote, cada pacote de dados Bluetooth transmitido possui um código de acesso. O código de acesso é robusto e resistente a interferências, o qual é gerado pelo mestre. Os elementos da *piconet* comparam o seu código de acesso com o do pacote de entrada, se os códigos combinarem o pacote de dados é considerado válido, caso contrário é descartado.

A figura 7.5 apresenta o formato do cabeçalho (*packet header*). A função do cabeçalho é controlar o fluxo da ligação lógica, o qual está dividido em vários campos de controlo. O campo AM_ADDR guarda um endereço temporário atribuído aos membros ativos da piconet, o qual pode ser utilizado na comunicação bidireccionalmente. Este é composto por uma sequência de 3 bits, por exemplo numa sequência 000 é atribuída uma transmissão broadcast aos escravos.

O campo TYPE ocupa 4 bits, sua finalidade é identificar o tipo de pacote transmitido. Os campos FLOW, ARQN e SEQN são flags de controlo. FLOW assinala controlo de fluxo, ARQN assinala se uma mensagem necessita do pacote ACK ou não, SEQN assinala se existe número de sequência para ordenação de pacotes. O campo HEC é composto por 8 bits, sua função é verificar por erros de bits no cabeçalho para fins de integridade.

O *payload* guarda o conteúdo da mensagem, podendo guardar até 2745 bits. Há dois tipos de payload: voz, dados. Os pacotes com conexão síncrona (*Synchronous connection oriented SCO*) destinam-se a voz. Já os pacotes de conexão assíncrona (*Asynchronous connectionless ACL*) destinam-se a pacotes de dados.

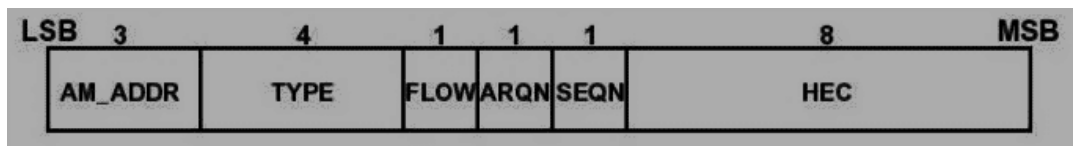


Figura 7.5 formato do cabeçalho do pacote de dados Bluetooth clássico [77].

Bluetooth Low Energy (BLE)

O *Bluetooth Low Energy* (BLE) opera na banda ISM, a sua versão base é o Bluetooth 4.0. Este ocupa 40 canais RF com separação espectral entre si de 2 MHz. O BLE utiliza canais publicitários e canais de dados. Os canais publicitários (*advertising*) são canais secundários, os quais se destinam à pesquisa de outras unidades, estabelecimento de conexões ou transmissões *broadcast* [23]. Os canais de dados são os canais principais, estes últimos destinam-se a comunicações bidirecionais entre unidades conectadas.

São reservados três canais para desempenhar funções como publicitários, são atribuídas frequências centrais que permitam evitar sobreposição espectral com as bandas IEEE 802.11. Os restantes canais são utilizados para transmissão de dados, para estes é utilizado um mecanismo adaptativo de salto em frequência. Este mecanismo é implementado selecionando num determinado instante um dos canais disponíveis para comunicação. O esquema de modulação utilizado é o GFSK (*Gaussian Frequency Shift Keying*). A taxa de transmissão de dados é de 1 Mbps.

No BLE quando uma unidade necessita de transmitir informação, primeiro enviam-se pacotes publicitários (*advertising packets*). As transmissões nos canais publicitários ocorrem em intervalos de tempo estabelecidos, estes são os eventos publicitários (*advertising events*). Durante um evento publicitário, os canais secundários são utilizados sequencialmente para transmitir o pacote. As unidades que recebem os pacotes publicitários são designadas de *scanners*.

Para haver condições para uma comunicação bidirecional entre duas unidades é necessário estas se conectarem. Para criar a conexão a unidade publicitária (*advertiser*) transmite um pacote a sinalizar que é um dispositivo conectável. O outro elemento é designado de iniciador. Primeiro, o iniciador limita-se a escutar atuando como *scanner*. Em seguida, o iniciador envia um pacote *Connect Request* ao publicitário, criando-se assim uma conexão ponto a ponto entre ambas as unidades. Agora já é possível comunicarem num canal de dados, o pacote *Connect Request* é identificado por uma chave aleatória de 32 bits. O mestre é o iniciador e o escravo é o publicitário [23].

A economia energética realiza-se ao ter as unidades escravas no modo baixo consumo. Estas acordam periodicamente para escutar os canais e procurar eventuais pacotes *Connect Request*. O mestre define em que instantes os escravos irão escutar o canal e coordena o acesso servindo-se de um esquema TDMA. O mestre ainda fornece ao escravo informação referente ao mecanismo de salto em frequência, esta informação é armazenada no pacote *Connect Request* e atualizada se necessário.

Após a criação do canal este é dividido em unidades temporais não sobrepostas, estes são os eventos de conexão. Num evento de conexão, os pacotes são transmitidos usando a frequência do canal de dados. O evento de conexão inicia-se com a transmissão de um pacote do mestre. Após a receção desse pacote, o escravo transmite os seus dados ao mestre. Há um intervalo de tempo entre cada evento de conexão, este é o *Inter Frame Space* (IFS), este deve ter uma duração de pelo menos 150 μ s [23].

Enquanto as unidades comunicarem entre si o canal é considerado aberto. Apenas quando a comunicação cessa, aí o evento de conexão é terminado. Não é requerido ao escravo que escute o canal até ao próximo evento de conexão. O evento de conexão pode ser terminado de forma forçada, essas situações incluem a receção de dois pacotes consecutivos com erros de bits ou se o campo de endereços de um dos dispositivos está corrompido [23]. O BLE é adequado para aplicações de IoT (*Internet of Things*), este pode ser utilizado por dispositivos móveis como smartphones. Esta solução poupa a necessidade de utilizar *gateways*, uma vez que a tecnologia já se serve oportunisticamente das soluções utilizadas por outras tecnologias de comunicações sem fios. A figura 7.6 apresenta a topologia de rede genérica para uma aplicação de sensores baseada em BLE.

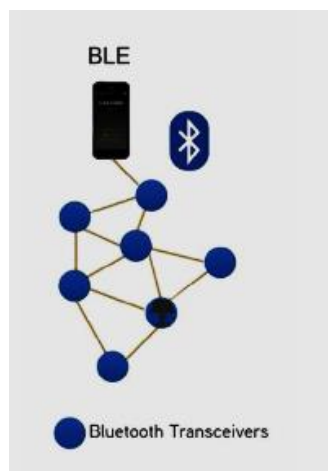


Figura 7.6 Arquitetura de rede numa aplicação de sensores baseada em BLE [23].

Formato trama – Bluetooth Low Energy

A figura 7.7 apresenta o formato de um pacote *Bluetooth Low Energy*, este é composto por preâmbulo (*Preamble*), endereço de acesso (*Access Address*), o campo *Protocol Data Unit* (PDU) e o campo CRC.

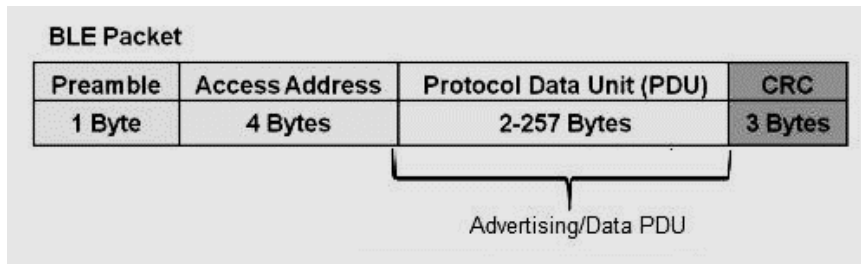


Figura 7.7 formato de um pacote *Bluetooth Low Energy* [78].

O preâmbulo é composto por um *byte*, a unidade recetora utiliza este campo para sincronismo temporal/frequência e controlo automático do ganho. O campo do endereço de acesso guarda endereços de 4 *bytes*, este é aplicável a pacotes de dados e publicitários. Para os pacotes publicitários, o endereço é fixo. Para os pacotes de dados, a unidade Bluetooth gera um endereço aleatório quando está inicializando, este endereço é usado nos pedidos de conexão.

O campo PDU pode ocupar até 257 *bytes*, os seus primeiros 2 *bytes* referem se ao cabeçalho do *payload*. A sua dimensão depende do tipo de pacote. A figura 7.8 apresenta o formato de um campo PDU considerando um pacote publicitário (em cima) e um pacote de dados (em baixo). Os últimos 3 *bytes* estão reservados ao campo CRC para codificação de erros de bits. Os pacotes publicitários têm duas funções, uma é transmissões *broadcast* para aplicações que não é necessário estabelecer conexões ao canal de dados, a outra é a pesquisa de unidades escravas e realizar o seu processo de conexão. Os pacotes de dados são apenas usados após se estabelecer conexão entre as unidades, estes são transmitidos no canal de dados servindo-se do agendamento de eventos de conexão.

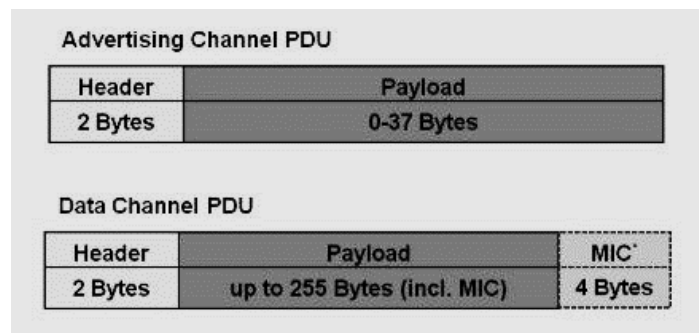


Figura 7.8 Formatos do campo PDU: pacote publicitário (em cima), pacote de dados (em baixo) [78].

Bluetooth 5.0

O Bluetooth 5.0 pode ser visto como uma evolução do BLE, este teve otimizações em termos de taxas de transmissão, alcance de propagação e capacidade de envio de pacotes publicitários. Esta última particularidade permite-lhe estabelecer conexões mais eficientemente. O esquema de modulação do Bluetooth 5.0 têm como base o esquema de modulação do BLE. Este possui três variantes de esquemas de modulação: 1 Mbps sem codificação, 1 Mbps com codificação e 2 Mbps sem codificação [79].

Idealmente, as taxas de transmissão podem chegar aos 2 Mbps, no entanto, estas são limitadas pela aplicação de esquemas de codificação de erros e os recursos de largura de banda disponíveis. Concretamente, a taxa de transmissão real está estimada nos 1,4 Mbps, isto considerando a utilização do esquema de modulação a 2 Mbps sem codificação. O melhor alcance de transmissão é conseguido utilizando o esquema de modulação a 1 Mbps com codificação. Além da codificação CRC para a deteção de bits errados, ainda se utilizam códigos de correção de erros. A sua função é corrigir erros de bits em rajada (*error burst*), o que permite aumentar o alcance e região de cobertura de sinal.

Como referido, a capacidade de envio de pacotes publicitários foi otimizada, isto através de um aumento considerável do número de canais secundários. Concretamente, utilizam-se 37 canais para a transmissão de pacotes publicitários. Os restantes 3 canais são os canais de dados para a comunicação entre unidades conectadas. A maior utilização de canais publicitários deve-se à introdução na trama do BLE de vários campos PDU adicionais [79]. A finalidade é visar um melhor desempenho nas transmissões broadcast.

O Bluetooth 5.0 introduziu a possibilidade de utilizar a topologia em malha. Esta particularidade abre a possibilidade das unidades Bluetooth poderem comunicar entre si diretamente, já que elimina a necessidade de um nó central para atuar como mestre. Também é possível a retransmissão de mensagens. A introdução da topologia em malha permite ao Bluetooth aumentar significativamente a sua cobertura de rede e o número de conexões. A topologia em malha do Bluetooth 5.0 quando comparada à topologia em estrela característica do Bluetooth clássico e versões que antecedem ao 5.0, em caso de falha de um nó, os danos nas comunicações são menos críticos, dada a maior possibilidade de conexões. A figura 7.9 exemplifica uma situação de falha num nó considerando cada uma das topologias referidas.

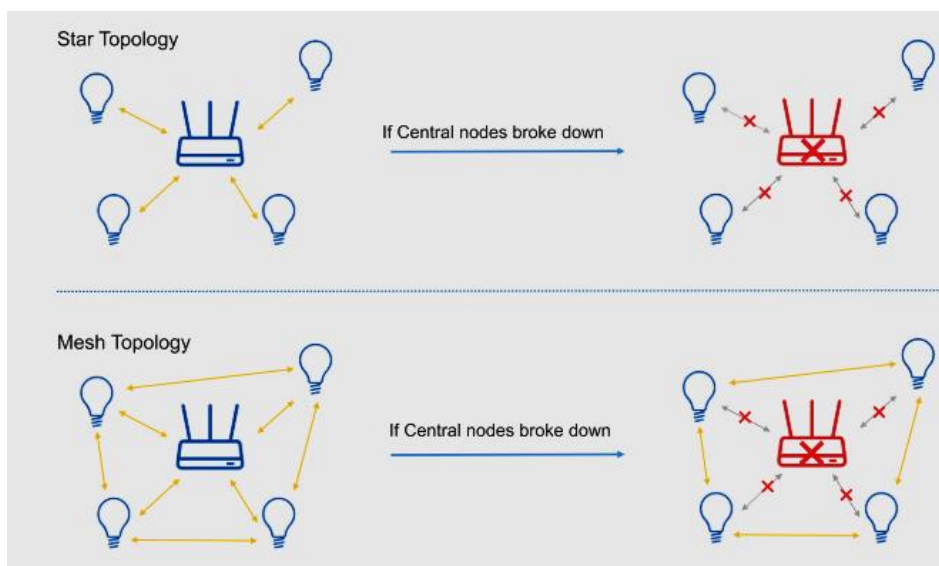


Figura 7.9 Topologia estrela vs topologia malha [79].

Na rede Bluetooth em malha, os nós comunicam entre si através da referida abordagem de pacotes publicitários. Deste modo, cada pacote contém o *payload* da mensagem, os endereços do nó emissor e recetor, entre outros campos de controlo. Atendendo às necessidades da aplicação há a liberdade de utilizar o pacote ACK.

ANEXOS

O Bluetooth 5.0 beneficia da vantagem da maioria dos dispositivos móveis, como é o caso dos smartphones já estarem equipados com Bluetooth. Não é necessário *hardware* adicional e há a liberdade de utilizar de forma oportunista meios de comunicação sem fios alternativos. Ainda a topologia em malha do Bluetooth 5.0 abre a possibilidade para diferentes cenários de aplicações de sensores, por exemplo, o cenário *many-to-many* apresentado na figura 7.10. Comparativamente ao Zigbee, o Bluetooth 5.0 é capaz de utilizar taxas de transmissão mais elevadas enquanto tendo consumos energéticos da mesma ordem.

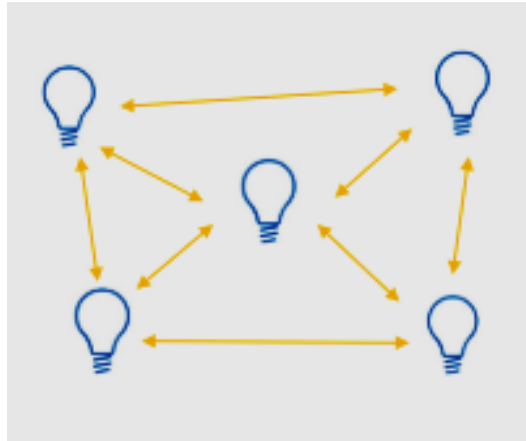


Figura 7.10 Bluetooth 5.0 topologia em malha com cenário *many-to-many* [79].

Anexo C - Formato mensagem no protocolo LoRaWAN

O protocolo LoRaWAN especifica um conjunto de mensagens MAC. A figura 7.11 apresenta o formato de uma mensagem MAC LoRaWAN, a qual é composta por cabeçalho (*MAC Header MHDR*), o payload (*MAC Payload*) e o campo de codificação da mensagem (*MIC Message Integrity Code*).

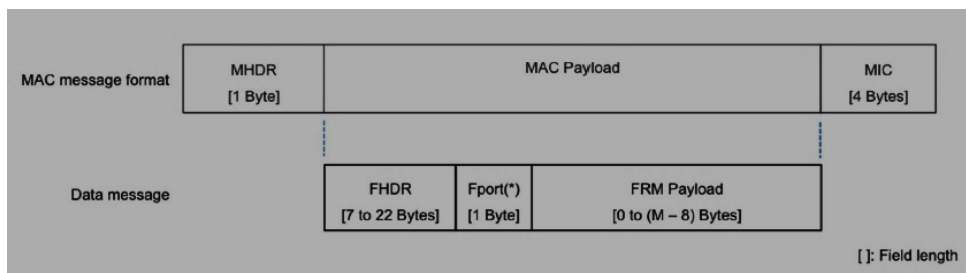


Figura 7.11 formato da mensagem MAC LoRaWAN [27].

O cabeçalho da mensagem MAC indica o tipo de mensagem. Há três tipos de mensagens: mensagens de conexão à rede (*Join Message*) e mensagens de dados. Estas últimas podem eventualmente ter confirmação (*Confirmed/Unconfirmed Data Message*). O *payload* pode conter mensagens de dados ou de conexão à rede. Caso, sejam mensagens de dados o *payload* divide-se em três campos: cabeçalho da trama (*Frame Header FHDR*), o campo *FPort* e o *payload* da trama (*FRM Payload*).

O cabeçalho da trama (FHDR) armazena mensagens MAC, o que depende do valor sinalizado pelo campo *FPort*. O *FPort* ainda possui um subcampo que guarda um bit, o qual faz o reconhecimento da última mensagem de dados entregue com confirmação.

O *FPort* ocupa um *byte*, o qual atua como uma *flag* sinalizando a existência de dados no *payload* da trama. Já o *payload* da trama a sua dimensão em *bytes* é variável. Este guarda comandos MAC, os quais caracterizam a configuração de parâmetros de radiofrequência e MAC. Por fim, o campo de codificação (MIC) permite ao recetor a avaliar a integridade da mensagem recebida.

Anexo D - Hardware nó sensor Urbisnet

A figura 7.12) a apresenta o módulo Wi-Fi Ethernet Bridge utilizado na estação Urbisnet da seção 2.9.2, este é o ASUS WL-330gE, o qual está configurado como *Ethernet Wi-Fi Bridge*. Opcionalmente, pode ser configurado como repetidor de sinal, *Access Point Wi-Fi* ou *Router Wi-Fi*, a figura 7.12) b apresenta o seu *hardware*. Este serve-se de duas antenas e a placa de circuito oferece a possibilidade de soldar duas antenas adicionais. O módulo é alimentado a 5 V, mas a estação móvel opera a 12 V, o que implica utilizar um conversor comutado DC/DC para realizar a regulação de tensão necessária.

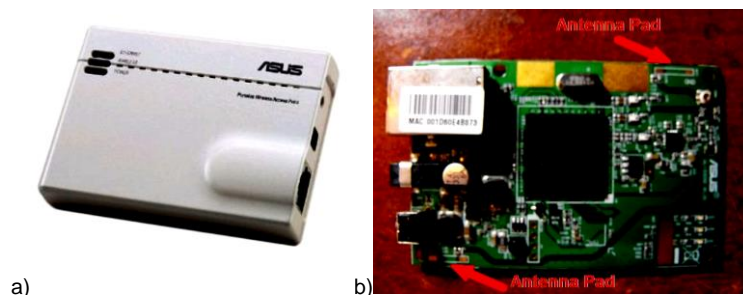


Figura 7.12 a) Módulo *Wi-Fi Ethernet Bridge* [34]. b) Respetivo *hardware* do módulo [34].

A figura 7.13) a apresenta a estação Urbisnet e a figura 7.13) b o seu *hardware*, do qual se utilizam duas placas de circuito. A placa mãe contém um *socket* para encapsular a placa C10/3 ilustrada na figura 2.11 da seção 2.9.2. A nível de *hardware* contém o sensor de temperatura/humidade Sensirion SHT25, circuitos de regulação de tensão e o ADC ADS7828. A placa mãe ainda possui um circuito de inicialização automática destinada ao módulo GSM quando a placa C10/3 é alimentada. Também possui uma interface *Ethernet* e uma porta série para aceder a uma consola *Linux*.



Figura 7.13 a) Estação Urbisnet [34]. b) Respetivo *hardware* [34].

A nível de *hardware* a placa C10/3 contém *slots* para cartão SD/MMC e cartão SIM, módulo GSM, microprocessador ARM e recetor GPS. A placa secundária integra os sensores de gás e respetivos circuitos de condicionamento referidos na seção 2.9.2. É utilizada cablagem para conetar a placa secundária e a placa mãe, é utilizada comunicação I2C.

Anexo E - Hardware nós de monitorização – projeto uSense

O sistema de monitorização uSense apresentado na seção 2.9.3, serve-se de nós Libellium Wasmote apresentados na figura 7.14) a. Estes nós estão equipados com um microcontrolador de 8 bits e um módulo Wi-Fi. O último permite conexões TCP/IP, UDP/IP e comunicações HTTP/HTTPS. Cada nó integra uma placa de circuito contendo os sensores de gás para CO, NO₂ e O₃ assim como os respetivos circuitos de condicionamento de sinal. Os sensores de gás são o TGS2442, MiCS-2714 e MiCS-2614 respetivamente. Também se utiliza sensores de temperatura e humidade. São lidas amostras a cada 30 minutos.

A eletrónica é instalada no interior de uma caixa PVC apresentada na figura 7.14) b. A caixa também contém uma antena, um interruptor, uma bateria recarregável com autonomia de 6600 mAh e uma ventoinha sem escovas (*brushless*). Esta última é alimentada por uma bateria adicional para fins de refrigeração do interior da plataforma.

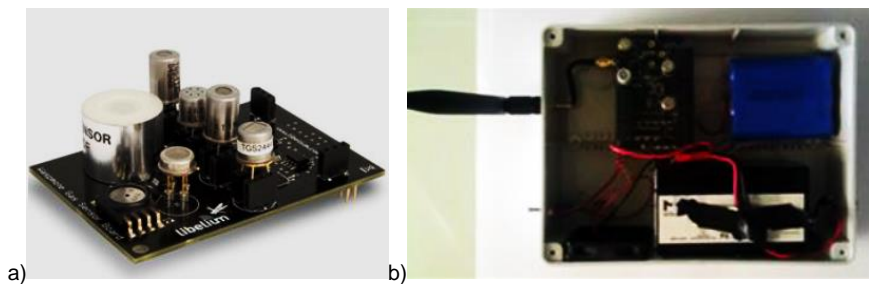


Figura 7.14 a) Placa de circuito dos sensores de gás Libellium [36]. b) Hardware do nó Libellium Wasmote [36].

Anexo G - XBee Series 2

Tabela 7.1 Descrição dos pinos dos módulos XBee/XBee-Pro™.

Pino #	Nome	Direção	Descrição
1	VCC	-	Alimentação, tipicamente 3,3 V
2	DOUT	Saída	Saída de dados da UART
3	DIN / CONFIG	Entrada	Entrada de dados da UART
4	DO8*	Saída	Saída digital 8
5	RESET	Entrada	Inicializa módulo (um pulso nível 0 de pelo menos 200ms)
6	PWM0 / RSSI	Saída	Saída do PWM 0 / Indicador de Força do sinal de RF (RX)
7	PWM1	Saída	Saída do PWM 1
8	(Reservado)	-	Ainda não tem uma função definida (futura implementação)
9	DTR / SLEEP_IRQ / DI8	Entrada	Linha de Controlo da Função Sleep ou Entrada digital 8
10	GND	-	Terra
11	AD4 / DIO4	Entrada/Saída	Só Entrada Analógica 4 ou Entrada/Saída Digital 4
12	CTS / DIO7	Entrada/Saída	Controle de Fluxo CTS ou Entrada/Saída Digital 7
13	ON / SLEEP	Saída	Indicador de Estado do Módulo
14	VREF	Entrada	Tensão de referência para as entradas A/D
15	Associação / AD5 / DIO5	Entrada/Saída	Indicador de associação, só Entrada Analógica 5 ou Entrada/Saída Digital 5
16	RTS / AD6 / DIO6	Entrada/Saída	Controlo de Fluxo RTS, só Entrada Analógica 6 ou Entrada/Saída Digital 6
17	AD3 / DIO3	Entrada/Saída	Só Entrada Analógica 3 ou Entrada/Saída Digital 3
18	AD2 / DIO2	Entrada/Saída	Só Entrada Analógica 2 ou Entrada/Saída Digital 2
19	AD1 / DIO1	Entrada/Saída	Só Entrada Analógica 1 ou Entrada/Saída Digital 1
20	AD0 / DIO0	Entrada/Saída	Só Entrada Analógica 0 ou Entrada/Saída Digital 0

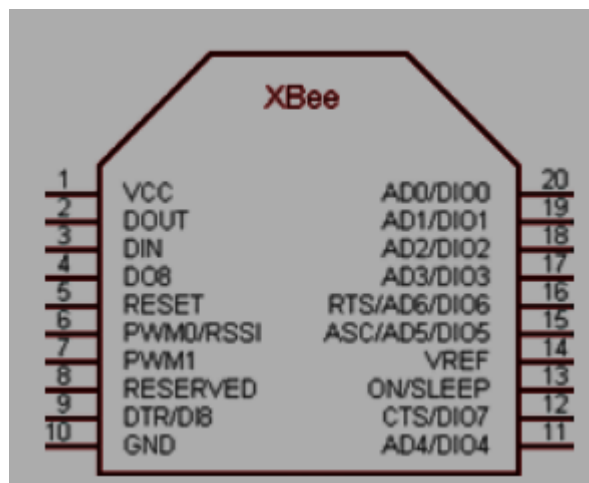


Figura 7.16 Representação do respetivo pinout.

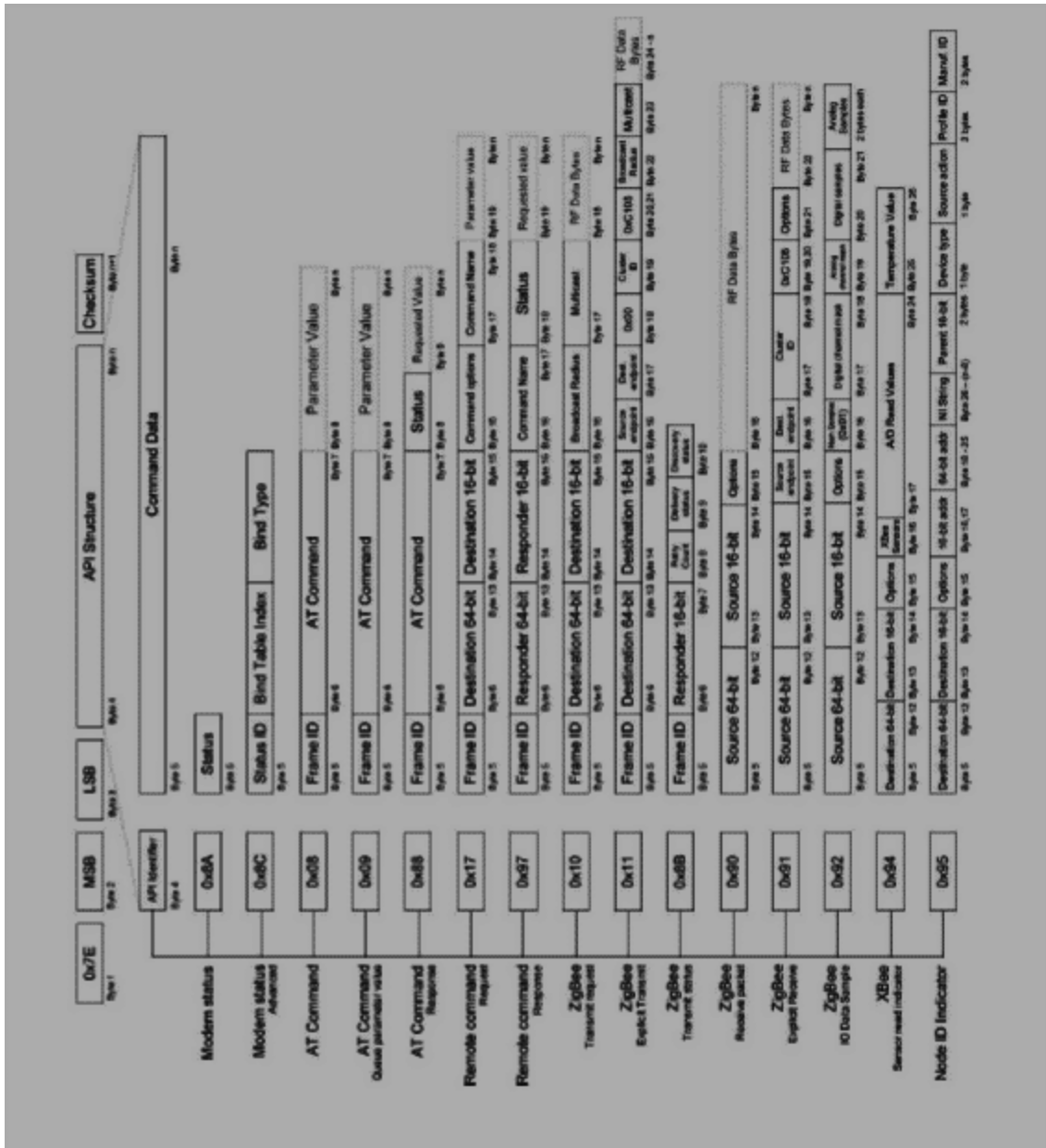


Figura 7.17 Estrutura de tramas API do XBee.

ANEXOS

A placa de circuito desenvolvida é de dupla face, nesta é acoplado o microcontrolador, circuitos de regulação de tensão e o circuito de carregamento para as baterias de lítio. Ainda se utilizaram resistências para o dimensionamento de divisores resistivos para a leitura de ADC, um LED sinalização e interruptores. Por fim utilizaram-se conectores *headers* de modo a facultar a conexão das baterias de lítio, o sensor de CO Alphasense B4 e o ecrã LCD TFT.

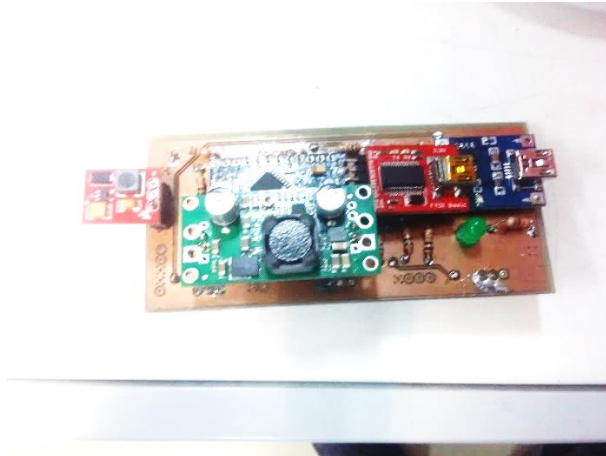


Figura 7.20 Placa-mãe módulo de calibração.

Código - controlo embestado

```
/* ALPHASENSE B4: dados mostrados no ecrã TFT LCD e guardados no cartão SD
o sensor CO funciona a 6V, o LCD TFT é alimentado a 5V, o processamento é realizado por um arduino PRO
MINI 3.3 V
PINAGEM USADA ARDUINO PRO MINI 3.3 V
Pinagem do SPI: MOSI (PINO D11), MISO (PINO D12), CLK (PINO D13)
PINAGEM CHIP SELECT: LCD (CS LCD PINO D3), CARTAO SD (CS SD PINO D4)
DATA CONVERTER: D/C PINO D5, RESET PINO D2
declaracao das bibliotecas */
#include <TFT.h>
#include <SPI.h>
#include <math.h>
#include <SD.h>
/* definicao dos pinos digitais:
* CS LCD PINO D3, CS SD CARD PINO D4
* DATA CONVERTER D/C PINO D5, RESET PINO D2*/
#define CSSD 4
#define CSLCD 3
#define DATACONVERTER 5
#define RESET 2
TFT TFTscreen = TFT(CSLCD, DATACONVERTER, RESET); // criacao de uma instancia na
biblioteca TFT
File FICHEIRO; // declaracao do nome da estrutura de dados para o cartão SD
char nomeVBAT1[10], valorVBAT1[10]; // criacao do array para mostrar os caracteres no ecrã: bateria
principal nome/valor
char nomeVBAT2[10], valorVBAT2[10]; // criacao do array para mostrar os caracteres no ecrã: bateria
externa nome/valor
```

ANEXOS

```
char nomeOP1WE[10], valorOP1WE[10], nomeOP2AUX[10], valorOP2AUX[10]; // criacao
do array para mostrar os caracteres no ecrã: eletrodos do Alphasense B4 nome/valor
char nomeniveisCO[10], valorniveisCO[10]; // criacao do array para mostrar os caracteres no ecrã:
niveis CO Alphasense B4 nome/valor
float batpr1, batex2; // niveis de tensao nas baterias principal e externa
int i=1; // variavel de estado

// a parte referente ao sensor CO Alphasense B4
#define OFFSET_OP1 492 // tensão de offset do elétrodo de trabalho offset_WE 492 mV
#define OFFSET_OP2 334 // tensão de offset do elétrodo auxiliar offset_AUX 334 mV
#define SENSIBILIDADE 405 // Sensibilidade 405 mV/ppm S
float OP1V, OP2V; // valores eletrodos convertidos em mV: eletrodo principal OP1_WE, eletrodo auxiliar
OP2_AUX
float CALCULO_OP1V, CALCULO_OP2V; // elétrodo OP1 calculo intermedio e elétrodo OP2 calculo
intermedio consiste em OP1-WE_VO e OP2-AUX_VO
float COB4; // variavel que guarda os valores das concentrações de monóxido de carbono CO do
AlphaSense B4 em p.p.m

void setup()
{
  // inicializacao do programa ECRA LCD TFT
  TFTscreen.begin(); // Inicializacao do ecrã LCD TFT
  TFTscreen.background(0, 0, 0); // procedimento que faz a limpeza dos caracteres no ecrã LCD
  TFTscreen.stroke(255, 255, 255); // ecrã esta pronto para mostrar os dados, e colocada a cor branca
  TFTscreen.setTextSize(1); // ajuste do tamanho do titulo
  TFTscreen.text("MEDICOES CO ALPHASENSE B4", 0, 0); // titulo
  TFTscreen.setTextSize(1); // tamanho para as strings seguintes
  // CARTAO SD
  Serial.begin(9600); // inicializacao do serial port
  pinMode(CSSD,OUTPUT); // o chip select do cartao SD e definido como output
  digitalWrite(CSSD,HIGH); // o chip select do cartao SD em seguida e colocado no estado HIGH

  // Inicializacao do cartao SD
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.print("Inicializar cartao SD...");
  if (!SD.begin(CSSD))
  {
    Serial.println("inicializacao falhou!");
    while (1);
  }
  Serial.println("inicializacao completa, pode utilizar o cartao SD.");
}
```

ANEXOS

```
void nivelbaterias()
```

```
{  
  /* divisores resistivos de 10 k para poder medir a tensao na bateria */  
  batpr1=2*3.3*analogRead(A0)/1024; // valor medido em Volts bateria principal ADC 0  
  batex2=2*3.3*analogRead(A1)/1024; // valor medido em Volts bateria externa ADC1  
}
```

```
void Alphasense()
```

```
{  
  /* funcao de calibracao do sensor Alphasense B4  
  * os valores lidos dos eletrodos sao convertidos em mV, sao calculados os niveis de CO  
  * o fator 1.83 e o fator do divisor resistivo responsavel pela regulacao da tensao dos 6 V para os 3.3 V */  
  // leitura e conversao em mV: OP1_WE, OP2_AUX  
  OP1V=1000*1.83*3.3*analogRead(A2)/1024; // eletrodo principal OP1 em mV OP1_WE ADC2  
  OP2V=1000*1.83*3.3*analogRead(A3)/1024; // eletrodo auxiliar OP2 em mV OP2_AUX ADC3  
  // calculo intermedio: OP1_WE-OFFSET OP1, OP2_AUX-OFFSET OP2  
  CALCULO_OP1V=OP1V-OFFSET_OP1; // OP1_WE-OFFSET OP1  
  CALCULO_OP2V=OP2V-OFFSET_OP2; // OP2_AUX-OFFSET OP2  
  // calculo dos niveis CO em p.p.m  
  COB4=abs((CALCULO_OP1V-CALCULO_OP2V)/SENSIBILIDADE);  
}
```

```
void TFTLCD()
```

```
{  
  /* funcao de escrita dos dados no ecrã TFT lcd  
  * dados escritos: nome/ valor tensao nas baterias, eletrodos Alphasense B4, niveis CO (p.p.m) */  
  /* conversao dos dados para enviar para o ecrã bateria principal */  
  String BATERIA1NOME = String("VBAT1(V)"); // criacao da string com o texto VBAT1(V)  
  BATERIA1NOME.toCharArray(nomeVBAT1,10); // os dados VBAT1(V) sao convertidos num  
  array de dados  
  String BATERIA1VALOR = dtostrf(batpr1, 4, 3, valorVBAT1); // 4 is minimum width, 3 is  
  precision; o valor float e convertido em char valorVBAT1  
  BATERIA1VALOR.toCharArray(valorVBAT1,10); // o valor é armazenado temporariamente em  
  valorVBAT1  
  // bateria externa  
  String BATERIA2NOME = String("VBAT2(V)"); // criacao da string com o texto VBAT2(V)  
  BATERIA2NOME.toCharArray(nomeVBAT2,10); // os dados VBAT2(V) sao convertidos num  
  array de dados  
  String BATERIA2VALOR = dtostrf(batex2, 4, 3, valorVBAT2); // 4 is minimum width, 3 is  
  precision; o valor float e convertido em char valorVBAT2  
  BATERIA2VALOR.toCharArray(valorVBAT2,10); // o valor é armazenado temporariamente em  
  valorVBAT2  
  // Alphasense B4 ELETRODO PRINCIPAL OP1_WE  
  String OP1mVNOME = String("OP1(mV)"); // criacao da string com o texto OP1(mV)  
  OP1mVNOME.toCharArray(nomeOP1WE,10); // os dados OP1(mV) sao convertidos num array  
  de dados  
  String OP1mVALOR = dtostrf(OP1V, 4, 3, valorOP1WE); // 4 is minimum width, 3 is precision;  
  o valor float e convertido em char valorOP1WE
```

ANEXOS

```
OP1mVALOR.toCharArray(valorOP1WE,10); // o valor é armazenado temporariamente em
valorOP1WE
// Alphasense B4 ELETRODO AUXILIAR OP2_AUX
String OP2mVNOME = String("OP2(mV)"); // criaçao da string com o texto OP2(mV)
OP2mVNOME.toCharArray(nomeOP2AUX,10); // os dados OP2(mV) sao convertidos num array
de dados

String OP2mVALOR = dtostrf(OP2V, 4, 3, valorOP2AUX); // 4 is minimum width, 3 is precision;
o valor float e convertido em char valorOP2AUX
OP2mVALOR.toCharArray(valorOP2AUX,10); // o valor é armazenado temporariamente em
valorOP2AUX
// Alphasense B4 NIVEIS CO
String COalphaNOME = String("ALPHAB4"); // criaçao da string com o texto niveis CO (p.p.m)
COalphaNOME.toCharArray(nomeniveisCO,10); // os dados niveis CO (p.p.m) sao convertidos
num array de dados
String COValphaALOR = dtostrf(COB4, 4, 3, valorniveisCO); // 4 is minimum width, 3 is
precision; o valor float e convertido em char niveis CO (p.p.m)
COValphaALOR.toCharArray(valorniveisCO,10); // o valor é armazenado temporariamente em
niveis CO (p.p.m)

// escrita dos dados no ecrã TFT LCD
TFTscreen.background(0, 0, 0); // procedimento que faz a limpeza dos caracteres no ecrã LCD
TFTscreen.stroke(255, 255, 255); // e colocada a cor branca
TFTscreen.text("CO ALPHASENSE B4",10, 0); // titulo centrado
TFTscreen.text(nomeVBAT1,10,20); // os dados sao mostrados no ecrã nome tensao da bateria 1
TFTscreen.text(valorVBAT1,80,20); // os dados sao mostrados no ecrã
TFTscreen.text(nomeVBAT2, 10,40); // os dados sao mostrados no ecrã nome tensao da bateria 2
TFTscreen.text(valorVBAT2,80,40); // os dados sao mostrados no ecrã
TFTscreen.text(nomeOP1WE, 10,60); // os dados sao mostrados no ecrã nome OP1 mV
TFTscreen.text(valorOP1WE,80,60); // os dados sao mostrados no ecrã
TFTscreen.text(nomeOP2AUX,10,80); // os dados sao mostrados no ecrã nome OP2 mV
TFTscreen.text(valorOP2AUX,80,80); // os dados sao mostrados no ecrã
TFTscreen.text(nomeniveisCO, 10,100); // dados sao mostrados no ecrã nome niveis CO (p.p.m)
TFTscreen.text(valorniveisCO,80,100); // os dados sao mostrados no ecrã
}

void escritadados()
{
  /* funcao para salvar os dados no cartao SD
  * os dados sao guardados no ficheiro txt ALPHAB4.TXT */
  FICHEIRO=SD.open("ALPHAB4.TXT", FILE_WRITE);
  // if the file opened okay, write to it:
  if(FICHEIRO)
  {
    Serial.print("Writing to ALPHAB4.TXT...");
    FICHEIRO.print(" OP1(A2) "); // texto ADC2 ELETRODO PRINCIPAL
    FICHEIRO.print(analogRead(A2)); // dados lidos do OP1(A2) ELETRODO PRINCIPAL OP1_WE
    FICHEIRO.print(" OP2(A3) "); // texto ADC3 ELETRODO AUXILIAR
```

ANEXOS

```
FICHEIRO.print(analogRead(A3)); // dados lidos do OP2(A3) ELETRODO AUXILIAR OP2_AUX
FICHEIRO.print(" OP1(mV) "); // texto OP1(mV) ELETRODO PRINCIPAL
FICHEIRO.print(OP1V); // dados lidos do OP1(A2) ELETRODO PRINCIPAL convertidos em mV
FICHEIRO.print(" OP2(mV) "); // texto OP2(mV) ELETRODO AUXILIAR
FICHEIRO.print(OP2V); // dados lidos do OP2(A3) ELETRODO AUXILIAR convertidos em mV
FICHEIRO.print(" ALPHAB4 "); // texto niveis CO (ppm) Alphasense B4
FICHEIRO.print(COB4); // dados lidos dos niveis CO (ppm) Alphasense B4
FICHEIRO.println(" "); // mudar de linha
FICHEIRO.close(); // close the file
Serial.println("done");
}

// esta condicao se verifica caso o ficheiro nao abra
else {
    // if the file didn't open, print an error:
    Serial.println("error opening ALPHAB4.TXT");
}

// funcao principal
void loop()
{
    while(i==1)
    {
        nivelbaterias(); // mede o nivel de tensao das baterias
        Alphasense(); // funcao de calibracao do Alphasense B4
        TFTLCD(); // mostra os dados no ecrã LCD TFT
        escritadados(); // guarda os dados no cartao SD no ficheiro ALPHAB4.TXT
        delay(1000); // tira as amostras a cada segundo
    }
}
```

Fotos – módulo de calibração

A figura 7.21) a apresenta o módulo de calibração e a figura 7.21) b o seu respetivo *hardware*, o qual foi resguardado numa caixa elétrica com as características de proteção IP69. Esta possui aberturas perfuradas destinadas aos interruptores, à entrada USB do circuito de carregamento de baterias de lítio, o sensor de CO Alphasense B4 e o ecrã LCD TFT. A placa de circuito foi posicionada no interior da caixa tendo em consideração o encaixe nas aberturas por parte dos interruptores e da entrada USB do circuito de carregamento para as baterias. O sensor de CO foi posicionado na abertura frontal de modo que a sua superfície absorvente ficasse em contacto direto com o ar. O ecrã LCD foi posicionado na abertura do topo da caixa elétrica.

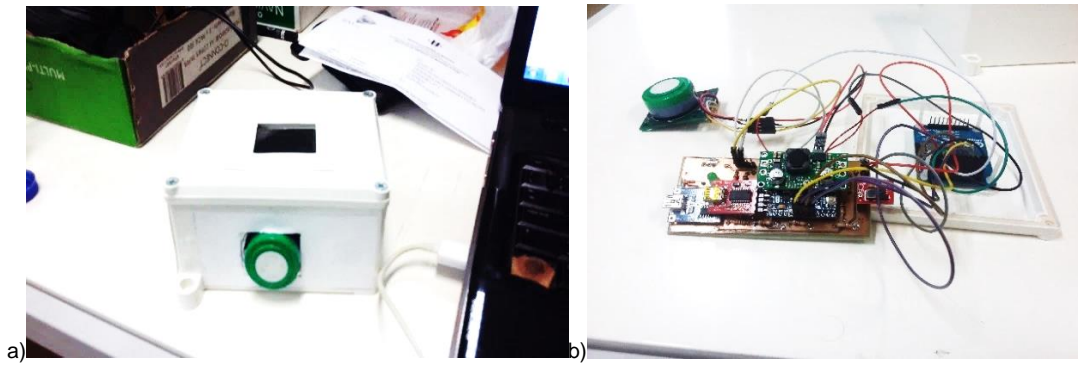


Figura 7.21 a) Módulo de calibração. b) Respetivo hardware.

A figura 7.22 ilustra o teste laboratorial realizado com os sensores de gás Alphasense B4 e Figaro TGS 5042 atendendo às condições descritas na seção 4.1, a partir deste teste se obtiveram os dados apresentados no gráfico da figura 4.1.



Figura 7.22 Ilustração do teste laboratorial realizado com os sensores de CO.

Anexo I – Nó sensor local

Circuito de condicionamento de sinal – sensor de CO Figaro TGS 5042 – design layout PCB e fotos

A figura 7.23 apresenta o *design PCB layout* realizado para o circuito de condicionamento de sinal do sensor Figaro TGS 5042, o qual é apresentado na seção 3.4.4.

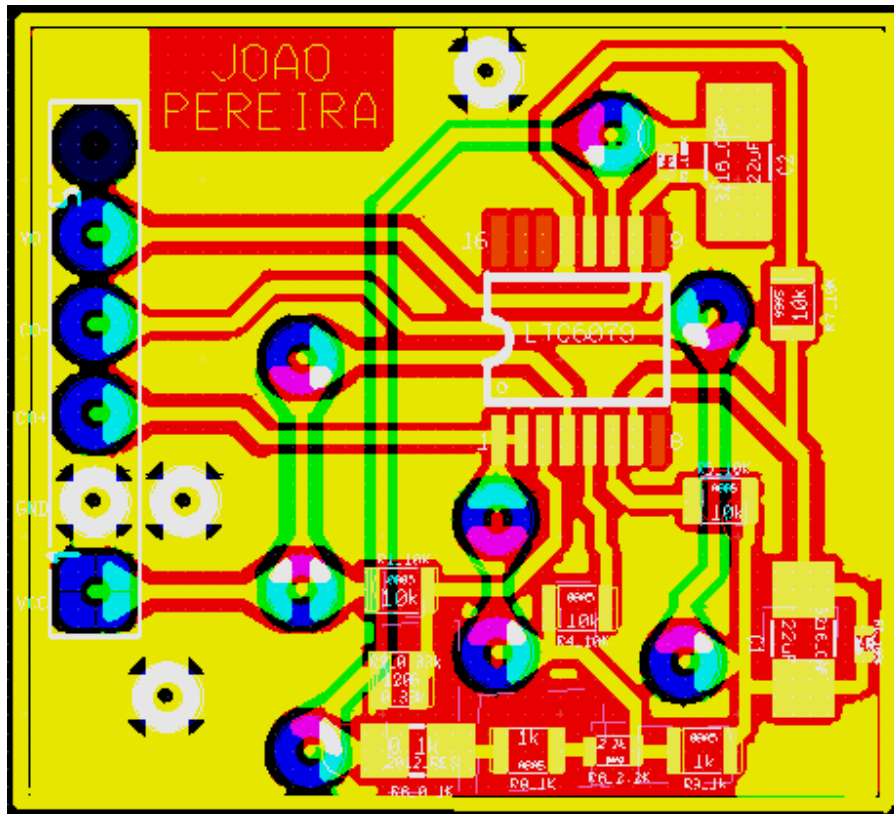


Figura 7.23 *Design layout PCB* – Circuito de condicionamento de sinal – sensor de CO Figaro TGS 5042.

Para o desenvolvimento deste circuito foi utilizada tecnologia SMD nomeadamente ao nível de resistências e condensadores. Utilizou-se também um amplificador de baixo ruído *Rail to Rail* que acopla 4 andares internos de amplificação, este é o LTC 6079 da *Linear Technology*. A placa de circuito impresso desenvolvida é de dupla camada, esta é apresentada na figura 7.24.



Figura 7.24 Placa de circuito impresso – circuito de condicionamento sensor de CO.

Estrutura anti-rradiação sensor de CO Figaro TGS 5042

A figura 7.25 apresenta a montagem da estrutura anti-rradiação para o sensor de CO Figaro TGS 5042. Esta foi construída servindo-se de um conjunto de pratos plásticos, os quais foram sobrepostos e afixados através de três varões de rosca. Para conferir melhor robustez à estrutura os varões de rosca também foram afixados. As extremidades superiores foram afixadas a uma chapa metálica e as extremidades inferiores foram afixadas a uma pequena base de madeira. Esta última ainda serviria de ponto de afixação à estrutura principal do nó local. A estrutura anti-rradiação foi pintada com tinta branca. O sensor de CO foi encapsulado e selado no interior visto que a finalidade consistia em proteger o sensor da radiação solar e de outros factores interferentes externos. Os fios de conexão dos eléctrodos são conectados ao circuito de condicionamento de sinal.



Figura 7.25 Estrutura anti-rradiação para o sensor de CO Figaro TGS 5042.

Esquemático

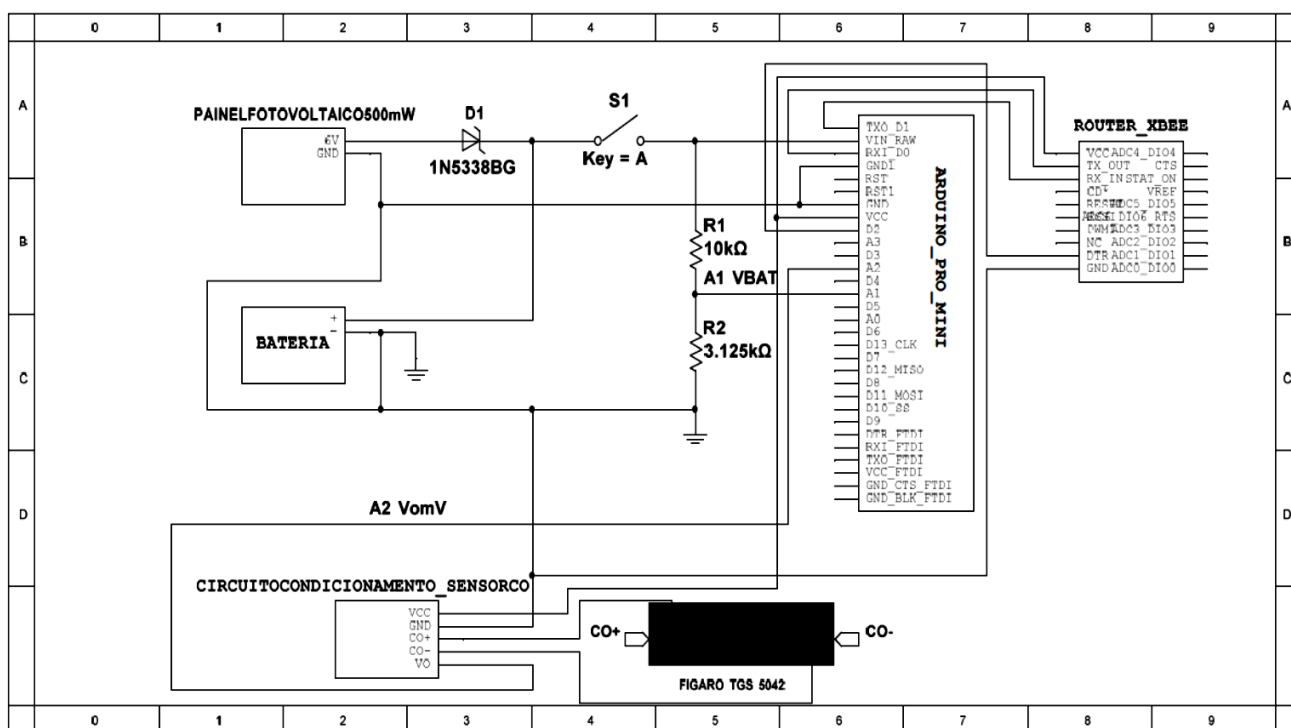


Figura 7.26 Esquemático nó sensor local.

Hardware nó local

A figura 7.27 apresenta o *design layout* do *hardware* para o nó local, o qual é referenciado na seção 3.4.7. A placa-mãe permite acoplar o microcontrolador, o circuito de condicionamento para o sensor Figaro e o módulo de comunicação XBee. Esta também possui resistências de modo a formar um divisor resistivo para leitura ADC e um interruptor. Foram utilizados conectores para o encaixe da bateria de lítio. Os restantes pontos de conexão externos tratam-se dos pontos de ligação ao painel fotovoltaico e ao sensor de CO respetivamente.

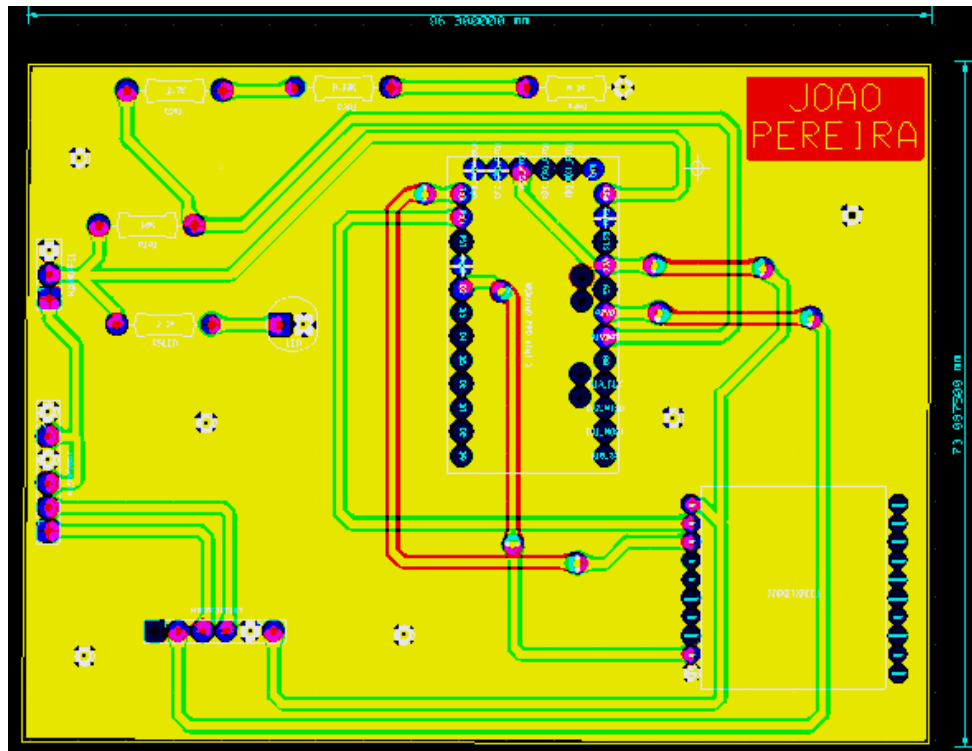


Figura 7.27 Design layout PCB placa mãe nó sensor local.

As figuras 7.28 e 7.29 apresentam fases de montagem do nó local assim como a estrutura completa desenvolvida.



Figura 7.28 Ilustração de algumas fases de montagem do nó local.

A eletrónica do nó local foi resguardada no interior de uma caixa elétrica com as características de proteção IP69, é de realçar que foi necessário vedar esta caixa elétrica de modo a evitar problemas associados ao ruído térmico e circulação de ar interior. Esta possui três aberturas perfuradas as quais se destinam ao interruptor geral, a cablagem exterior e a antena do XBee Series 2.

ANEXOS

A estrutura principal é constituída em acrílico com o objetivo de conferir robustez à mesma. A estrutura anti-rradiação foi afixada no ponto inferior da estrutura principal. No ponto superior foi instalado o painel fotovoltaico, para o qual se utilizou uma estrutura de suporte secundária de modo a se assegurar robustez na afixação. Ainda se teve o cuidado de posicionar o painel num ângulo apropriado a uma melhor captação da energia solar. À saída do painel fotovoltaico, conectou-se um diodo zener de potência para fins de regulação de tensão e minimizar perdas associadas a correntes reversas. A caixa elétrica foi instalada num ponto intermédio da estrutura principal.



Figura 7.29 Nó sensor local completo.

Código - controlo embebido

```
// declaracao das bibliotecas
#include <avr/sleep.h>
#include <math.h>

// inicializacao variaveis
#define OFFSETVOLTAGE_VREF 90 // tensao de referencia 90 mV TGS5042
float vbat; //tensao da bateria
int valorlidosensorCO; // valor lido sensor CO em tensao mV
float COTGS5042; // niveis CO em p.p.m
unsigned int tcnt2; // timer
int seconds = 10; // Período em segundos
int start = seconds*30; // contagem de restart do timer
int counter = 0; // contador
```


ANEXOS

```
TIMSK2 &= ~(1<<OCIE2A);
/* Now configure the prescaler to CPU clock divided by 1024
   configuracao do prescaler do CPU*/
TCCR2B |= (1<<CS22) | (1<<CS21) | (1<<CS20); // Set bits
// Save value globally for later reload in ISR */
tcnt2 = 0; // valor e guardado temporariamente
TCNT2 = tcnt2; // valor guardado no registo do timer
TIMSK2 |= (1<<TOIE2); // o timer e ativado
set_sleep_mode(SLEEP_MODE_PWR_SAVE);
sleep_enable();
}

ISR(TIMER2_OVF_vect)
{
    /* Rotina de Servico de interrupcao Reload the timer */
    TCNT2 = tcnt2; // carrega o timer
    counter++; // contador efetua as contagens
}

void leitura_valores()
{
    // usei esta funcao por causa de definir a referencia interna para 1.1 V
    for (int i = 0; i < 8; i++)
    {
        vbat=(4.12*1.1*analogRead(A1))/1023; // ADC1 tensao bateria VBAT (V)
        valorlidosensorCO=(1000*1.1*analogRead(A2))/1023; // ADC2 tensao sensor CO (mV)
    }
}

void leituraADCs_CO()
{
    /* funcao de tratamento dos dados: leitura dos ADCs: tensao bateria e do sensor CO
    * e envio dos dados
    * os ADCs sao lidos com o analog reference internal a 1.1 V para uma melhor resolucao dos resultados*/
    leitura_valores();
    delay(10); // atraso de 10 ms devido à leitura
    /* niveis CO equacao calibracao sensor CO:
    COTGS5042 = ( valorlidosensorCO-TENSAOFFSET)/A/at
    SENSIBILIDADE TGS5042 BOARD JOAO 1.527 nA/ppm GANHO A 3.13
    FATOR A/at calculado 4.77951 */
    COTGS5042=abs((valorlidosensorCO-OFFSETVOLTAGE_VREF)/4.77951);
    // processo necessario devido ao tipo de dados float truncagem: VBAT, Niveis CO
    AUX1BAT=vbat;
    AUX2BAT=vbat*100-AUX1BAT*100;
    AUX1CO=COTGS5042;
    AUX2CO=COTGS5042*100-AUX1CO*100;
```

ANEXOS

```
printf(acBuf,"%1d.%2d",AUX1BAT,AUX2BAT); // VBAT float
trama[24] = acBuf[0]; // parte inteira
trama[25] = acBuf[1]; // ponto
// agora vem a parte decimal
if(acBuf[2]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[2]=48; // 0 ASCII (48 decimal)
    trama[26] = acBuf[2]; // decima
}
trama[26] = acBuf[2]; // decima
trama[27] = acBuf[3]; // centesima

printf(acBuf,"%2d.%2d",AUX1CO,AUX2CO); // Niveis CO float
// para o primeiro byte de valor de tensao de saida
if(acBuf[0]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[0]=32; // space ASCII (32 decimal)
    trama[38] = acBuf[0]; // primeiro carater
}
trama[38] = acBuf[0]; // parte inteira
trama[39] = acBuf[1]; // parte inteira
trama[40] = acBuf[2]; // ponto
// agora vem a parte decimal
if(acBuf[3]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[3]=48; // 0 ASCII (48 decimal)
    trama[41] = acBuf[3]; // decima
}
trama[41] = acBuf[3]; // parte decimal
trama[42] = acBuf[4]; // byte menos significativo
printf(acBuf,"%4d",valorlidosensorCO); // valores de tensao de saida em mV inteiro

// para o primeiro byte de valor de tensao de saida
if(acBuf[0]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[0]=32; // space ASCII (32 decimal)
    trama[49] = acBuf[0]; // primeiro carater
}
// para o segundo byte de valor de tensao de saida
if(acBuf[1]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[1]=32; // space ASCII (32 decimal)
    trama[50] = acBuf[1]; // decima
}
```

ANEXOS

```
trama[49] = acBuf[0];
trama[50] = acBuf[1];
trama[51] = acBuf[2];
trama[52] = acBuf[3];

// Wake up XBee
digitalWrite(xbeeSleep, LOW); // Liga xbee
delay(15); // Espera que acorde
// Cálculo do checksum
checksum = 0;
for ( i = 3; i < sizeof(trama)-1; i++)
{
    checksum+= trama[i]; // Soma os valores que contam para checksum
}
trama[sizeof(trama)-1] = 0xFF - checksum; // Realiza o complemento para 2
Serial.write(trama, sizeof(trama)); // Envia a trama pela porta série
delay(100);
digitalWrite(xbeeSleep, HIGH); // Sleep XBee
}

void loop()
{
    if(counter == start)
    {
        TIMSK2 &= ~(1<<TOIE2); // desativa o timer
        leituraADCs_CO(); // executa a tarefa
        counter = 0; // contador faz o restart
        TCNT2 = tcnt2; // vai reiniciar o timer
        TIMSK2 |= (1<<TOIE2); // ativa o timer
    }
    // modos Sleep
    sleep_enable();
    sleep_mode();
    sleep_disable();
}
```

Teste estudo da influência da temperatura nas medições de CO

A figura 7.30 ilustra a realização dos testes do estudo da influência da temperatura nas medições de CO, para os quais se obtiveram os resultados apresentados na seção 4.2.3. No exemplo ilustrado, utilizou-se um multímetro e um termístor NTC adicionais, deste modo era possível monitorizar a temperatura interior da caixa elétrica.

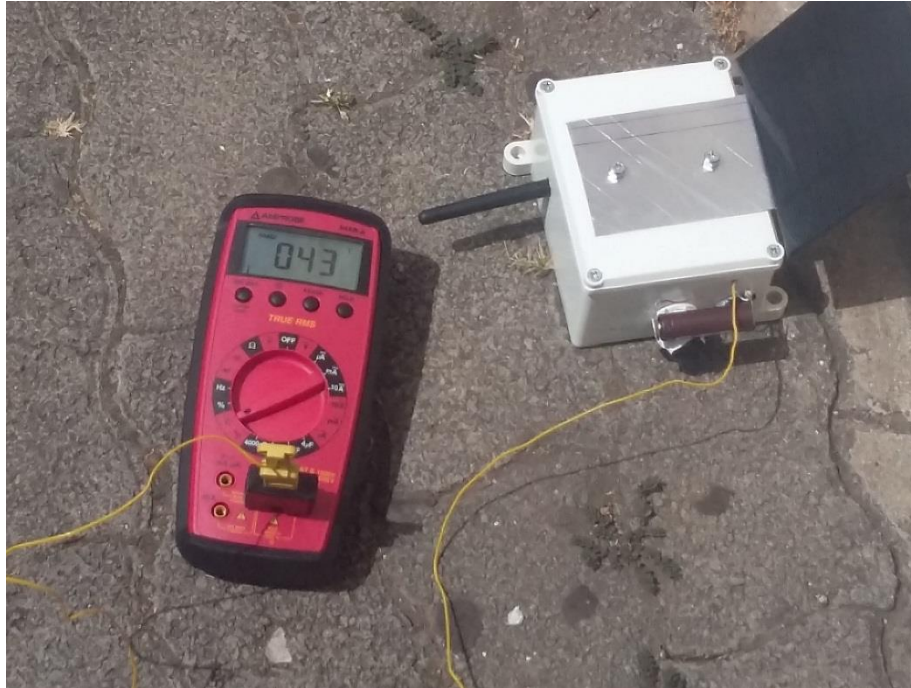


Figura 7.30 Ilustração dos testes do estudo da influência da temperatura nas medições de CO.

Variabilidade dados de CO

As figuras 7.31 e 7.32 caracterizam gráficos nas horas de maior afluência do tráfego urbano. À semelhança dos gráficos apresentados na seção 4.2.4 são consideradas as horas da manhã (08:00 às 10:30) e da tarde (16:30 às 19:00) nos dias 29 de Janeiro e 2 de Abril de 2019. Estes casos caracterizam situações com concentrações de monóxido de carbono mais elevadas.

No gráfico da figura 7.31 para o dia 29 de Janeiro, é de destacar os instantes próximos das 09:00 e entre as 10:00 e as 10:30 respetivamente. Nesses instantes, registaram-se concentrações na ordem de 4,5 , 3,5 , 2,5 e 2 ppm. De acordo com a expressão (3.3), isto traduz-se em níveis de 5,16 , 4 , 2,86 e 2,29 mg/m³. Pela tabela 2.1, o IQAr para o nível de CO mais elevado aqui registado é classificado como Bom.

Quanto ao gráfico da figura 7.32 para o dia 29 de Janeiro destacam-se os instantes entre as 18:00 e as 19:00 com níveis de CO próximos dos 6 e 4 ppm. O que se traduz em valores de 6,87 e 4,58 mg/m³. O caso de CO mais poluente é classificado no limiar de Bom aproximando-se de Médio. No dia 2 de Abril entre as 18:00 e as 18:30, o caso de CO mais poluente registado encontrava-se na ordem de 10 ppm. O que corresponde a 11,46 mg/m³. De acordo com a tabela 2.1 este nível de IQAr é classificado como Mau.

ANEXOS

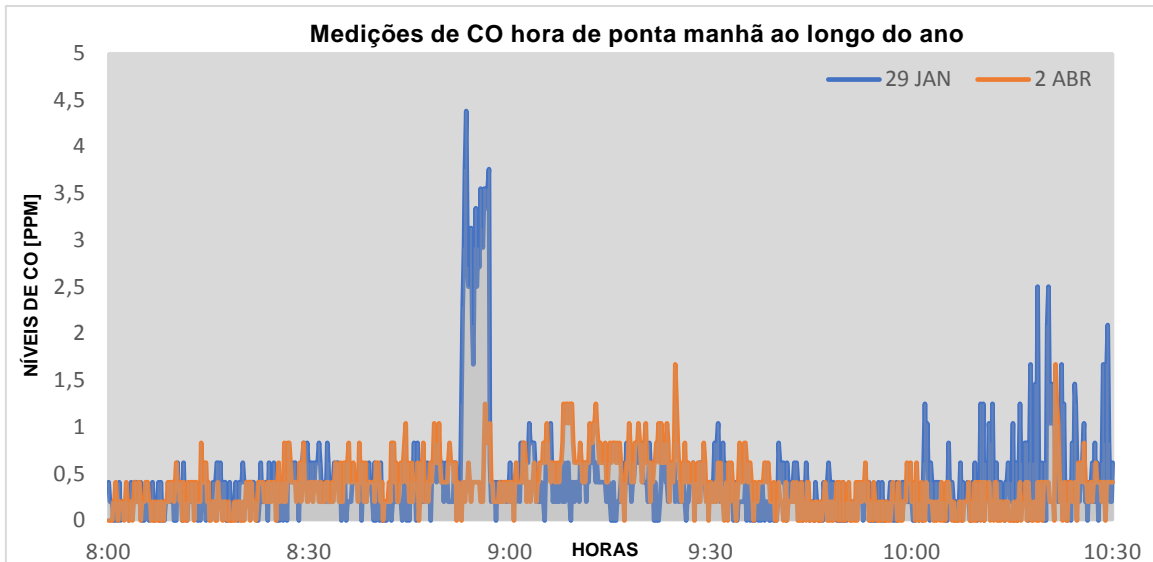


Figura 7.31 Medições de CO entre as 08:00 e as 10:30 nos dias 29-01-2019 e 02-04-2019.

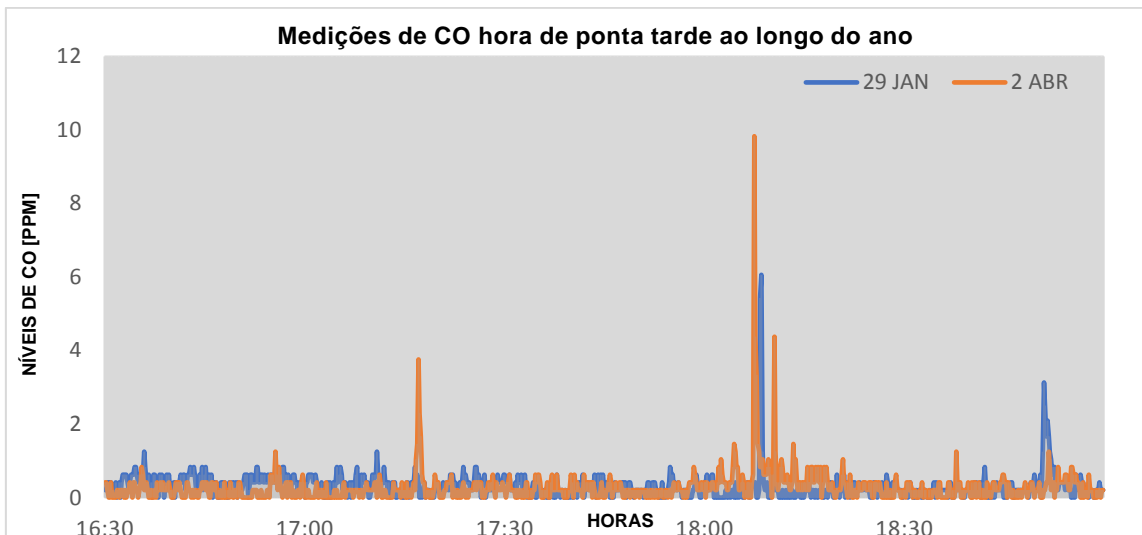


Figura 7.32 Medições de CO entre as 16:30 e as 19:00 nos dias 29-01-2019 e 02-04-2019.

Anexo J – Gateway de aquisição de dados

Esquemático

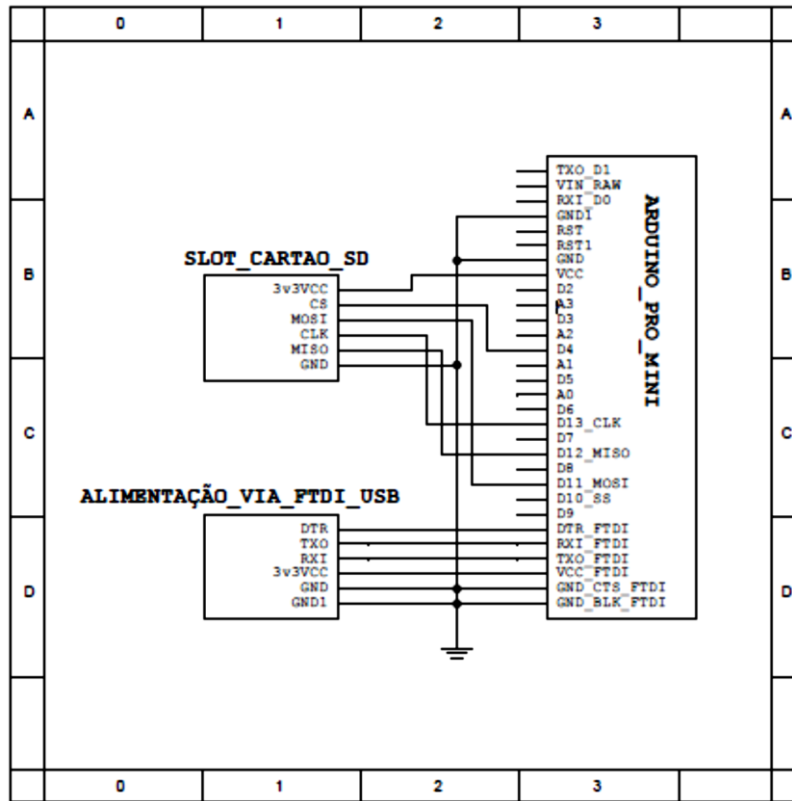


Figura 7.33 Esquemático gateway de aquisição de dados.

Código - controlo embestado

```

/** implementação de um modulo de aquisicao de dados **/
/* leitura de dados do ficheiro FICHD.TXT, este contém os dados de um nó sensor móvel
 * os dados sao guardados num buffer temporariamente, estes podem ter dados de GPS ou nao,
 * estes sao encapsulados na trama de rececao do Zig Bee e enviados para o serial monitor
 * TRAMA QUE PRETENDO ENVIAR: *ts=DDMMYYHHMMSS&LAT=DADOS&LONG=DADOS&CO=DADOS
 * pinagem do arduino fio para o leitor de cartao sd:
 * GND pino 1 leitor cartao - arduino 3V3 pino 6 leitor cartao - arduino
 * MISO leitor cartao pino 2 - pino digital 12 (D12) arduino
 * Clk leitor cartao pino 3 - pino digital 13 (D13) arduino
 * MOSI leitor cartao pino 4- pino digital 11 (D11) arduino
 * CS leitor cartao pino 5- pino digital 4 (D4) arduino */
// bibliotecas usadas
#include <math.h>
#include <stdlib.h>
#include <SPI.h>
#include <SD.h>
// inicializacoes variaveis
#define VREF 360 // tensao de referencia do sensor CO
#define CS 4 // chip select
File FICHEIRO; // estrutura de dados do cartao SD
char buff[50]; // buffer

```

ANEXOS

```
int indice=0; // indice do buffer
int VomV=0; // valores VomV
float CO=0; // valores CO (ppm)

/* TRAMA DE RECECAO ZIG BEE
 * informacao 0x7E – Byte de início
 * 0x001E – Comprimento da mensagem (em bytes): contagem dos bytes entre os campos lenght e checksum
 * decimal (67 bytes) HEXADECIMAL 0x00 0x43
 * 0x10 – Identificador de API para transmissão de dados
 * 0x90 – Identificador de API para rececao de dados
 * Numeracao da mensagem 0x00
 * Endereço do coordenador 64 bits - identificador gateway: 0x3C,0x2B,0x4A,0x45,0x08,0x0A,0x05,0x01
 * Endereço 16 bits destinatario 0xFF,0xFE
 * 0x00 – Número máximo de saltos de uma mensagem de broadcast (0 para número máximo de saltos);
 DADOS DA TRAMA QUE PRETENDO ENVIAR:
 * ts=DDMMYYHHMMSS&LAT=DADOS&LONG=DADOS&CO=DADOS
 *ts= 0x2A (*) 0x74 (t) 0x73 (s) 0x3D (=)
 DDMMYYHHMMSS 0x00 0x00 (DD) 0x00 0x00 (MM) 0x00 0x00 (YY) 0x00 0x00 (HH) 0x00 0x00 (MM) 0x00 0x00 (SS)
 &LAT= 0x26 (&) 0x4C (L) 0x41 (A) 0x54 (T) 0x3D (=)
 DADOS LATITUDE 3239.1997 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 &LONG= 0x26 (&) 0x4C (L) 0x4F (O) 0x4E (N) 0x47 (G) 0x3D (=)
 DADOS LONGITUDE 01655.1215 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 &CO= 0x26 (&) 0x43 (C) 0x4F (O) 0x3D (=)
 DADOS CO %2d.2d 0x00 0x00 0x00 0x00 0x00
 0x00 - byte de checksum */
byte trama[ ] ={0x7E,0x00,0x43,0x90,0x3C,0x2B,0x4A,0x45,0x08,0x0A,0x05,0x01,
0xFF,0xFE,0x00,0x2A,0x74,0x73,0x3D,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x26,0x4C,0x41,0x54,0x3D,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x26,0x4C,0x4F,0x4E,0x47,0x3D,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x26,0x43,0x4F,0x3D,0x00,0x00,0x00,0x00,0x00,0x00};

/***** SETUP *****/
void setup()
{
  /* inicializacao do serial port e ativacao dos pinos digitais */
  Serial.begin(9600); // inicializa o serial port
  pinMode(CS,OUTPUT); // chip select do cartao SD definido como output
  digitalWrite(CS,HIGH); // chip select do cartao SD colocado no estado HIGH

  /* inicializacao do cartao SD */
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  Serial.print(" Inicializando o cartao SD ...");
}
```

ANEXOS

```
if (!SD.begin(CS))
{
  Serial.println(" Erro inicializacao, tente outra vez! ");
  return;
}
Serial.println(" Inicializacao completa, pode usar o cartao! ");
recolhaenviodados(); // corro o codigo principal no setup
}

/***** LOOP *****/
void loop()
{
  // nao acontece nada
}

/***** recolhaenviodados() *****/
void recolhaenviodados()
{
  /* nesta funcao os dados do cartão SD no ficheiro FICHD.TXT, ficheiro do nó movel
  * sao guardados no buffer temporariamente, e chamada a funcao de decisao
  * que atua em funcao do tamanho do buffer e vai preencher a trama de rececao Zig Bee */
  /* leitura dos dados do ficheiro FICHD.TXT */
  FICHEIRO = SD.open("FICHD.txt");
  if (FICHEIRO)
  {
    // le o ficheiro FICHD.txt byte a byte ate ao fim
    while (FICHEIRO.available())
    {
      buff[indice]=FICHEIRO.read(); // le byte a byte e guarda no buffer de dados
      // se ainda nao terminou a leitura ate ao fim do ficheiro
      if(buff[indice]!=0x0A)
      {
        indice++; // continua a incrementar o indice
      }
    }
    // se ja tiver terminado a contagem dos bytes
    else
    {
      decisao(); // vai correr o algoritmo de decisao
      indice=0; // reinicia o indice
      Serial.write(trama,sizeof(trama)); // transmite a trama
      delay(100); // espera 100 ms
    }
  }
}
FICHEIRO.close(); // fecha o ficheiro
```

ANEXOS

```
}
else
{
    // no caso de nao abrir da mensagem de erro
    Serial.println(" erro ao abrir o ficheiro FICH.TXT ");
} }
/***** DECISAO *****/
void decisao()
{
    /* algoritmo que analisa se a mensagem tem dados de GPS ou nao
    * tem como base a comparacao do indice do buffer
    * a trama de rececao Zig Bee é preenchida em funcao dos casos*/
    if(indice == 26) // caso dos dados sem GPS
    {
        tratamento_dadoslidossemgps(); // chamo a funcao que trata dos dados sem GPS
    }
    if(indice == 45) // caso dos dados com GPS
    {
        tratamento_dadoslidosgps(); // chamo a funcao que trata dos dados com GPS
    }
}

/***** TRATAMENTO DADOS SEM GPS *****/
void tratamento_dadoslidossemgps()
{
    /* nesta funcao trato dos dados recebidos sem gps
    * e feita a insercao dos mesmos na trama Zig Bee
    * parametros temporais: DDMMYYHHMMSS
    *
    * TRAMA DE DADOS SEM GPS
    * $050180,235945,,464,3.87 (26 carateres)
    * $DDMMYY,HHMMSS,,VO(mV) */
    trama[19]=buff[1]; // DD
    trama[20]=buff[2];
    trama[21]=buff[3]; // MM
    trama[22]=buff[4];
    trama[23]=buff[5]; // YY
    trama[24]=buff[6];
    trama[25]=buff[8]; // HH
    trama[26]=buff[9];
    trama[27]=buff[10]; // MM
    trama[28]=buff[11];
    trama[29]=buff[12]; // SS
    trama[30]=buff[13];
```

ANEXOS

```
/* preencho a trama Zig Bee rececao nas posicoes dos dados GPS
com 0 ASCII (48 decimal) */
trama[36]=48; // latitude parte 1: %4d
trama[37]=48;
trama[38]=48;
trama[39]=48;
trama[40]=48; // onde ficaria o ponto: 0 ASCII (48 decimal)
trama[41]=48; // latitude parte 2: %4d
trama[42]=48;
trama[43]=48;
trama[44]=48;
trama[51]=48; // longitude parte 1: %5d
trama[52]=48;
trama[53]=48;
trama[54]=48;
trama[55]=48;
trama[56]=48; // onde ficaria o ponto: 0 ASCII (48 decimal)
trama[57]=48; // longitude parte 2: %4d
trama[58]=48;
trama[59]=48;
trama[60]=48;
/* chamo a funcao que le os dados de Vo, calcula niveis CO
* e coloca os resultados na trama rececao Zig Bee */
leituradadosCO();
}

/***** TRATAMENTO DE DADOS COM GPS *****/
void tratamento_dadoslidsgps()
{
/* nesta funcao trato dos dados com gps
* e feita a insercao dos mesmos na trama Zig Bee
* parametros temporais: DDMMYYHHMMSS
*
* TRAMA DE DADOS COM GPS
* $140618,111633,3239.4767,01655.2442,362,4.08 (45 caracteres)
* $DDMMYY,HHMMSS,LAT,LONG,VO(mV) */
trama[19]=buff[1]; // DD
trama[20]=buff[2];
trama[21]=buff[3]; // MM
trama[22]=buff[4];
trama[23]=buff[5]; // YY
trama[24]=buff[6];
trama[25]=buff[8]; // HH
trama[26]=buff[9];
trama[27]=buff[10]; // MM
```

ANEXOS

```
trama[28]=buff[11];
trama[29]=buff[12]; // SS
trama[30]=buff[13];
/* dados GPS: parametros espaciais: latitude, longitude*/
trama[36]=buff[15]; // latitude parte 1: %4d
trama[37]=buff[16];
trama[38]=buff[17];
trama[39]=buff[18];
trama[40]=buff[19]; // onde fica o ponto
trama[41]=buff[20]; // latitude parte 2: %4d
trama[42]=buff[21];
trama[43]=buff[22];
trama[44]=buff[23];
trama[51]=buff[25]; // longitude parte 1: %5d
trama[52]=buff[26];
trama[53]=buff[27];
trama[54]=buff[28];
trama[55]=buff[29];
trama[56]=buff[30]; // onde fica o ponto
trama[57]=buff[31]; // longitude parte 2: %4d
trama[58]=buff[32];
trama[59]=buff[33];
trama[60]=buff[34];
/* chamo a funcao que le os dados de Vo, calcula niveis CO
 * e coloca os resultados na trama rececao Zig Bee */
leituradadosCO();
}

/*****LEITURA DE DADOS CO *****/
void leituradadosCO()
{
/* crio uma string nova, preencho com os caracteres pretendidos do buffer
 * converto para inteiro com atoi, aplico a equacao de calculo de CO
 * coloco o valor de CO na trama Zig Bee */
int AUX1CO,AUX2CO; // variaveis auxiliares para CO
/* buffers temporarios:
 * primeiro guarda os caracteres do valor de Vo
 * segundo guarda os caracteres do valor de CO calculado */
char Vo[2],acBuf[5+1];

/* primeiro a seguinte verificacao: as posicoes do buffer com os dados de Vo
 * dependem se existem ou nao dados de GPS,
 * consoante o caso: preencho o vetor Vo[ ] com os caracteres pretendidos do buffer
 * em seguida, converto em int e obtenho o valor de Vo */
```

ANEXOS

```
if(indice == 26) // caso dos dados sem GPS
{
    Vo[0]=buff[17];
    Vo[1]=buff[18];
    Vo[2]=buff[19];
    VomV=atoi(Vo);
}
if(indice == 45) // caso dos dados com GPS
{
    Vo[0]=buff[36];
    Vo[1]=buff[37];
    Vo[2]=buff[38];
    VomV=atoi(Vo);
}

/* Tratamento dos valores de CO: equacao de calculo de CO
* CO(ppm)=(Vo(mV)-VoREF(mV))/FATOR_SIMPLIFICADO */
CO=abs((VomV-VREF)/4.77951); // equacao de calculo de CO
// processo necessario devido ao tipo de dados float truncagem: Niveis CO
AUX1CO=CO;
AUX2CO=CO*100-AUX1CO*100;
sprintf(acBuf,"%2d.%2d",AUX1CO,AUX2CO); // Niveis CO float

// para a parte decimal valor CO quarto byte
if(acBuf[3]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuf[3]=48; // 0 ASCII (48 decimal)
    trama[68] = acBuf[3]; // quarto byte
}

trama[65]=acBuf[0];
trama[66]=acBuf[1];
trama[67]=acBuf[2];
trama[68]=acBuf[3];
trama[69]=acBuf[4];
}
```

ANEXOS

Dados temporais e geográficos para os pontos geográficos apresentados no mapa da seção 4.3

Tabela 7.2 Dados temporais e de coordenadas geográficas dos pontos ilustrados no mapa da figura 4.17.

Pontos representados no mapa	Dados temporais (HHMMSS)	Latitude (DDD)	Longitude (DDD)
Ponto 1	11:40:00	3238,9898	1654,8822
Ponto 2	11:40:12	3239,0162	1654,8158
Ponto 3	11:40:24	3239,025	1654,7673
Ponto 4	11:40:36	3239,0424	1654,7367
Ponto 5	11:40:48	3239,0391	1654,7066
Ponto 6	11:41:00	3239,0668	1654,7085
Ponto 7	11:41:12	3239,0898	1654,6884
Ponto 8	11:41:24	3239,1367	1654,686
Ponto 9	11:41:37	3239,1662	1654,6895
Ponto 10	11:41:49	3239,1894	1654,6552
Ponto 11	11:42:01	3239,1988	1654,6442
Ponto 12	11:42:14	3239,2052	1654,6436
Ponto 13	11:42:28	3239,2326	1654,7539
Ponto 14	11:42:40	3239,2325	1654,7539
Ponto 15	11:42:53	3239,2483	1654,7784
Ponto 16	11:43:05	3239,2736	1654,7859
Ponto 17	11:43:16	3239,2863	1654,8016
Ponto 18	11:43:30	3239,3004	1654,8277
Ponto 19	11:43:45	3239,3008	1654,8282
Ponto 20	11:44:06	3239,2926	1654,8351

Gateway de aquisição de dados – Fotos

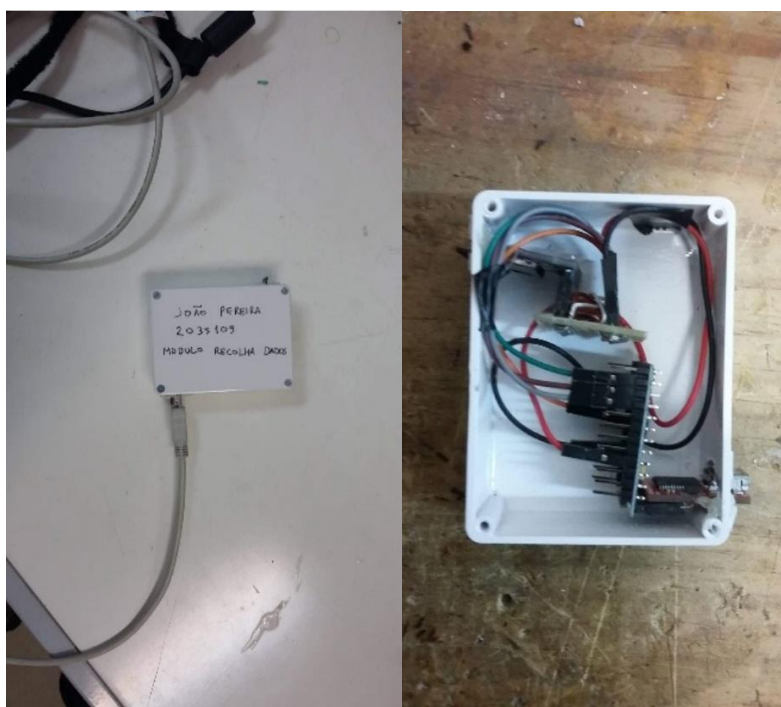


Figura 7.34 Gateway de aquisição de dados.

Anexo K – Terminal remoto

Esquemático e Hardware

A figura 7.35 apresenta o respetivo esquemático e as figuras 7.36 e 7.37 ilustram estação remota desenvolvida.

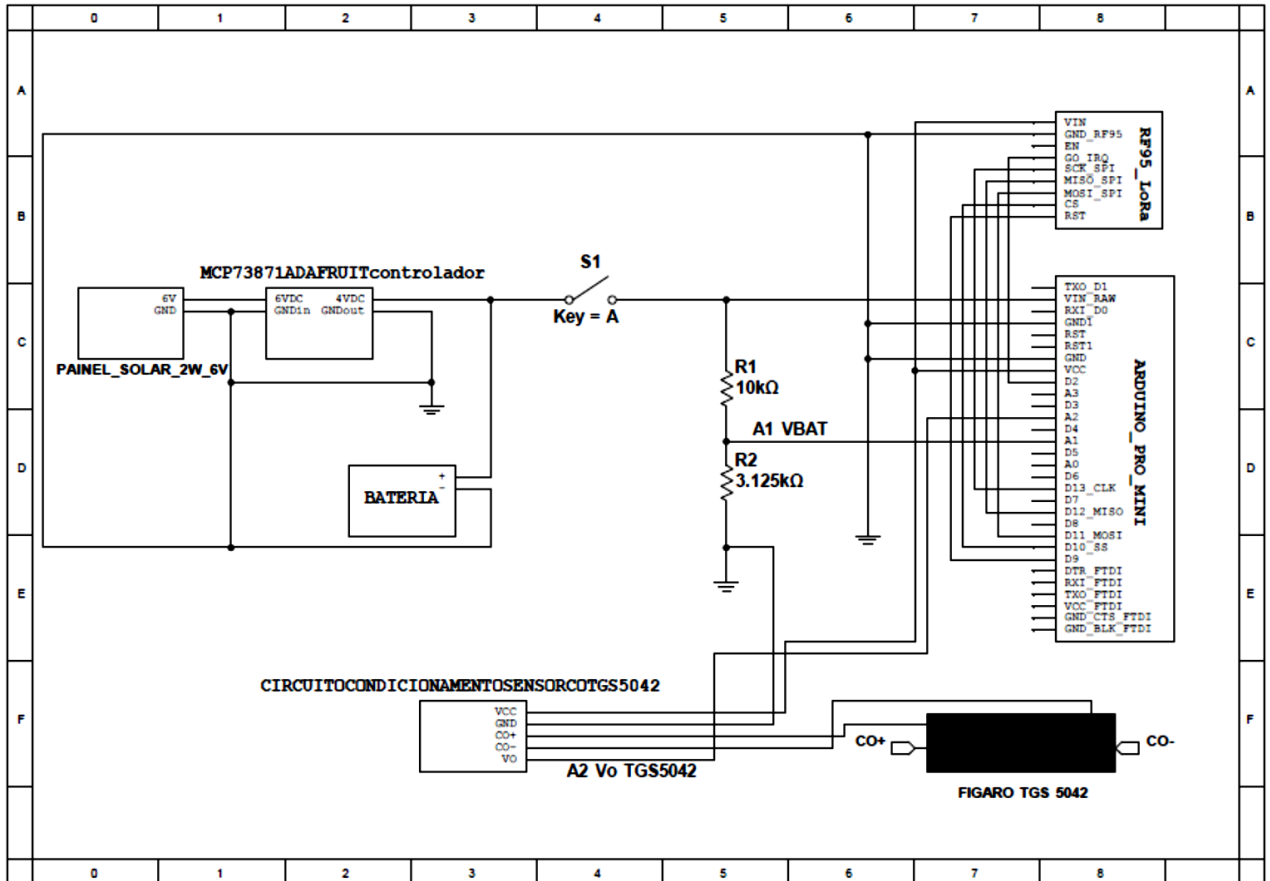


Figura 7.35 Esquemático terminal remoto.



Figura 7.36 Terminal remoto.

ANEXOS

No desenvolvimento do *hardware* utilizou-se uma placa de circuito pré-perfurada. O *hardware* é o apresentado na figura 7.35, o qual é similar ao utilizado para o nó local. As diferenças destacam-se ao nível do módulo de comunicação utilizado (RFM9x LoRa), o controlador de carga e as características elétricas do painel utilizado. Tal como o nó local, a eletrónica foi resguardada numa caixa elétrica com as características de proteção IP69. Esta também foi convenientemente vedada para evitar circulação de ar interior e problemas associados ao ruído térmico. Deste modo, esta contém apenas duas aberturas perfuradas, a primeira destina-se ao monopolo do RFM9x LoRa. A segunda destina-se à saída de cablagem, a qual possui as características H07-V dada a sua considerável robustez à ação de força mecânica. A cablagem exterior destinava-se ao sensor Figaro e ao painel fotovoltaico.

A estrutura principal da estação remota é constituída em madeira rígida, a caixa elétrica foi instalada no ponto intermédio. O painel solar foi instalado no topo da estrutura principal. Houve necessidade de construir uma estrutura de suporte adicional que permitisse afixar o painel num ângulo adequado a uma melhor captação da radiação solar. Esta também deveria possibilitar uma rotação manual do painel até 180° tendo em consideração a posição de observação do Sol ao longo do dia. À semelhança do nó local o sensor Figaro foi encapsulado e selado numa estrutura anti-rradiação, a qual foi afixada à extremidade inferior da estrutura principal. Esta também era possível uma rotação manual até 180°.

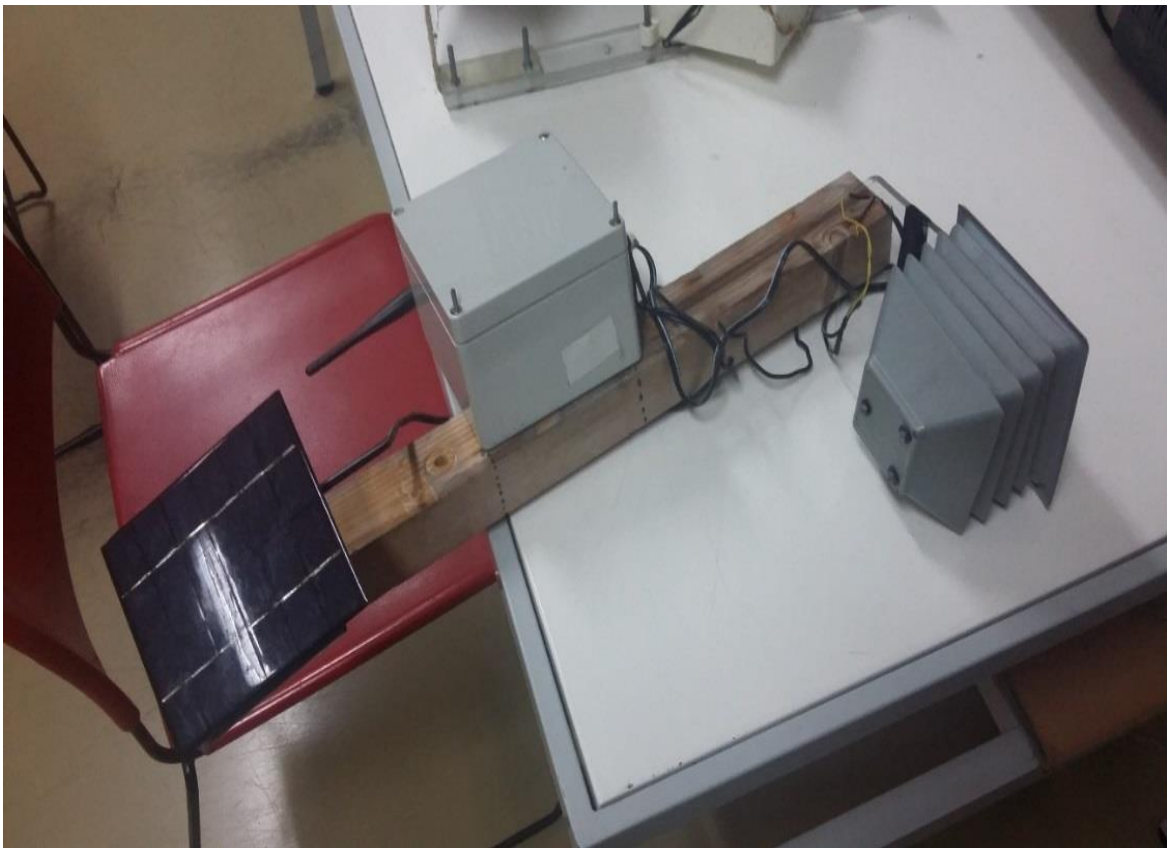


Figura 7.37 Terminal remoto.

Código - controlo embebido

```

/* CODIGO NO REMOTO TERMINAL:
 * LoRa low power modo adormecimento 120 segundos para ter 1% DC
 * SLEEP MODE LoRa Pinout RF9Mx : SX127x
 * Vin (posso usar 3.3 V ate 6 V), GND
 * SCK (CLK) D13 ARDUINO, MISO D12 ARDUINO, MOSI D11 ARDUINO
 * CS D10 ARDUINO, RST D9 ARDUINO
 * G0 IRQ (Interrupt Request pin) D2 ARDUINO (PINO INTERRUPTCAO INT0),
 * used for interrupt request notification from the radio to
 * the microcontroller, 3.3V logic level
 *
 * tambem englobado: modo long range 4096 chips/s, enderecamento unicast
 * enderecamento cliente-servidor: no terminal e configurado como cliente
 * Atribuicao enderecos: cliente (terminal remoto 203), servidor (gateway 109)
 *
 * ADCs: ADC1 (BATV), ADC2 (VomV sensor CO)
 * dados sao convertidos e encapsulados na estrutura da mensagem LoRa RFM9x RF95 */
/* bibliotecas SPI, RH_RF95, avr.sleep, RHGenericDriver, RHReliableDatagram
 * para rf95.setModemConfig(RH_RF95::Bw125Cr48Sf4096) implica o uso de RHGenericDriver */
#include <SPI.h>
#include <RH_RF95.h>
#include <RHGenericDriver.h>
#include <RHReliableDatagram.h>
#include <avr/sleep.h>

// Atribuicao enderecos: cliente (terminal remoto 203), servidor (gateway 109)
#define TERMINAL_NODE 203
#define GATEWAY_NODE 109

/* CS D10 ARDUINO, RESET D9 ARDUINO, INTERRUPT PIN D2 ARDUINO
 * INTERRUPT PIN D2 ARDUINO para acordar/ adormecer modulo RF LoRa */
#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2

/* Change to 434.0 or other frequency, must match RX's freq!
 * definido nos 868.0 MHz a frequencia de operacao na Europa */
#define RF95_FREQ 868.0
/* Singleton instance of the radio driver
 * instancia da biblioteca RadioHead para o modulo RFM9x
 * esta faz uso dos pinos CS e IRQ */
RH_RF95 rf95(RFM95_CS, RFM95_INT);
/* Class to manage message delivery and receipt, using the driver declared above
 * classe responsavel pelo enderecamento das mensagens */
RHReliableDatagram manager(rf95, TERMINAL_NODE);

```

ANEXOS

```
/* parametros da mensagem a transmitir (10 bytes/ c terminador)
 * tensao da bateria em Volts ADC1, valor lido sensor CO em tensao mV ADC2 */
byte data[8+1];
float BATV;
int VomV;
// usando o timer do avr sleep
unsigned int tcnt2; // timer
int seconds = 120; // Período em segundos: defini 120 segundos para 1% Duty cycle
int start = seconds*30; // contagem para o restart do timer
int counter = 0; // contador
unsigned long time; // contador temporal para o modo ativo

/*****SETUP*****/
void setup()
{
  /* configuracao dos pinos digitais usados:
   * pino D2 (G0) LoRa para o sleep mode
   * pino D9 RESET LoRa */
  pinMode(RFM95_INT, OUTPUT);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  // defino a referencia interna de leitura dos ADCs para os 1.1 V
  analogReference(INTERNAL);
  // inicializacao do serial monitor
  while (!Serial);
  Serial.begin(9600);
  delay(100);
  time=millis(); // ativacao do timer para o modo acordado

  // manual reset para o modulo RF95
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);

  // inicializacao do modulo RF95
  if (!manager.init())
  {
    Serial.println(" fail ");
  }
  Serial.println(" OK "); // inicializacao completa

  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
  // ajuste nos 868 MHz
  if (!rf95.setFrequency(RFM95_FREQ))
```

ANEXOS

```
{
  Serial.println(" fail ");
  while (1);
}
rf95.setFrequency(RF95_FREQ); // ajuste da frequencia para os 868 MHz
rf95.setModeTx(); // configuro o modulo LoRa como transmissor
rf95.setTxPower(14); // configuracao da potencia de transmissao para os 14 dBm
// configuracao no modo long range 4096 chips/symbol
rf95.setModemConfig(RH_RF95::Bw125Cr48Sf4096);
/* configuracoes do timer para o modo sleep */
/* First disable the timer overflow interrupt while we're configuring */
TIMSK2 &= ~(1<<TOIE2); // timer 2 e desativado

/* Configure timer2 in normal mode (pure counting, no PWM etc.)
   configuracao do timer 2 para atuar como contador */
TCCR2A &= ~((1<<WGM21) | (1<<WGM20));
TCCR2B &= ~(1<<WGM22);
/* Select clock source: internal I/O clock
   * frequencia do clk interno e definida */
ASSR &= ~(1<<AS2);
/* Disable Compare Match A interrupt enable (only want overflow) */
TIMSK2 &= ~(1<<OCIE2A);
/* Now configure the prescaler to CPU clock divided by 1024
   * configuracao do prescaler do CPU*/
TCCR2B |= (1<<CS22) | (1<<CS21) | (1<<CS20); // Set bits
/* Save value globally for later reload in ISR,
   * o timer e configurado para começar a contagem a partir de 0 */
tcnt2 = 0; // valor e guardado temporariamente
TCNT2 = tcnt2; // valor guardado no registo do timer
TIMSK2 |= (1<<TOIE2); // o timer e ativado
// ativacao dos adormecimentos
rf95.sleep(); // modulo RF95 LoRa e adormecido
set_sleep_mode(SLEEP_MODE_PWR_SAVE); // define o modo adormecido do microcontrolador
sleep_enable(); // ativacao do sleep mode
sleep_mode(); // entra no modo adormecido
}

/*****ISR*****/
ISR(TIMER2_OVF_vect)
{
  /* Rotina de Servico de interrupcao: o timer e reinicializado
   * e realiza a contagem aqui */
  TCNT2 = tcnt2; // carrega o timer
  counter++; // contador efetua as contagens
}
}
```

ANEXOS

```
/****** leitura_ADCs() *****/  
void leitura_ADCs()  
{  
    /* funcao de leitura dos ADCs: BATV (ADC1), VomV (ADC2)  
    * defini a referencia interna para 1.1 V */  
    for (int i = 0; i < 8; i++)  
    {  
        BATV=(4.11*1.1*analogRead(A1))/1023; // valor tensao V lido ADC1 tensao bateria BATV  
        VomV=(1000*1.1*analogRead(A2))/1023; // valor tensao mV lido ADC2 sensor CO VomV  
    }  
}
```

```
/******conversao_dados()*****/  
void conversao_dados()  
{  
    /* funcao de conversao dos dados lidos dos ADCs  
    * e feita a conversao dos dados de modo a estes serem encapsulados  
    * na estrutura da mensagem a enviar pelo modulo RFM9x RF95 */  
    int AUX1BAT,AUX2BAT; // variaveis auxiliares para converter BAT em inteiro  
    char Buf[3+1]; // buffer de dados auxiliar  
    // dados de BATV  
    AUX1BAT=BATV;  
    AUX2BAT=BATV*100-AUX1BAT*100;  
    sprintf(Buf,"%1d.%2d",AUX1BAT,AUX2BAT); // VBAT float  
    data[0]=Buf[0]; // parte inteira  
    data[1]=Buf[1]; // separador decimal  
    // parte controlo dos dados  
    // agora vem a parte decimal: byte mais significativo  
    if(Buf[2]<49) // se for inferior a 1 em ASCII (49 decimal)  
    {  
        Buf[2]=48; // preenche com 0 ASCII (48 decimal)  
        data[2]=Buf[2]; // guarda na mensagem  
    }  
  
    // agora vem a parte decimal: byte menos significativo  
    if(Buf[3]<49) // se for inferior a 1 em ASCII (49 decimal)  
    {  
        Buf[3]=48; // preenche com 0 ASCII (48 decimal)  
        data[3]=Buf[3]; // guarda na mensagem  
    }  
    data[2]=Buf[2]; // caso normal: byte mais significativo da parte decimal  
    data[3]=Buf[3]; // byte menos significativo da parte decimal  
    data[4]=38; // separador &
```

ANEXOS

```
// dados de VomV
sprintf(Buf,"%4d",VomV); // VomV int
// parte controlo dos dados
// para o primeiro byte de valor de tensao de saida
if(Buf[0]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    Buf[0]=48; // preenche com 0 ASCII (48 decimal por causa do NetBeans)
    data[5]=Buf[0]; // guarda na mensagem
}
// para o segundo byte de valor de tensao de saida
if(Buf[1]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    Buf[1]=48; // preenche com 0 ASCII (48 decimal por causa do NetBeans)
    data[6]=Buf[1]; // guarda na mensagem
}
// para byte menos significativo
if(Buf[3]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    Buf[3]=48; // preenche com 0 ASCII (48 decimal por causa do NetBeans)
    data[8]=Buf[1]; // guarda na mensagem
}
data[5]=Buf[0];
data[6]=Buf[1];
data[7]=Buf[2];
data[8]=Buf[3];
}

/***** LOOP *****/
void loop()
{
    if(counter == start)
    {
        TIMSK2 &= ~(1<<TOIE2); // desativa o timer
        digitalWrite(RFM95_INT, LOW); // desativa a interrupcao acorda o RFM9x RF95

        // tarefas a executar no sistema
        delay(20); // espera 20 ms para dar tempo ao sistema
        leitura_ADCs(); // leitura dos ADCs
        delay(10); // espera 10 ms devido ao processo de leitura dos ADCs com a referencia interna
        conversao_dados(); // os dados lidos dos ADCs sao convertidos e encapsulados na mensagem
        /* Send a message to manager_server: endereçamento unicast
        * terminal envia a mensagem para o gateway
        * parametros: mensagem, dimensao e endereço do gateway */
        // transmissao da mensagem
        manager.sendtoWait((byte *)data,sizeof(data),GATEWAY_NODE);
    }
}
```

ANEXOS

```
Serial.write((byte *)data,sizeof(data));
Serial.println(time); // imprime o tempo do timer no modo acordado
delay(1000);

/* processo para o sistema entrar no modo adormecido novamente */
digitalWrite(RFM95_INT, HIGH); // ativa a interrupcao, adormece o RFM9x RF95
counter = 0; // contador faz o restart
TCNT2 = tcnt2; // vai reiniciar o timer
TIMSK2 |= (1<<TOIE2); // ativa o timer
}
/* o microcontrolador e o modulo LoRa RF sao adormecidos */
rf95.sleep(); // modulo RF95 LoRa e adormecido
set_sleep_mode(SLEEP_MODE_PWR_SAVE); // define o modo adormecido do microcontrolador
sleep_enable(); // ativacao do sleep mode
sleep_mode(); // entra no modo adormecido
}
```

Anexo L – Gateway LoRa/Zigbee

Esquemático e hardware

A figura 7.38 apresenta o respetivo esquemático e as figuras 7.39 e 7.40 ilustram o gateway LoRa/Zigbee desenvolvido.

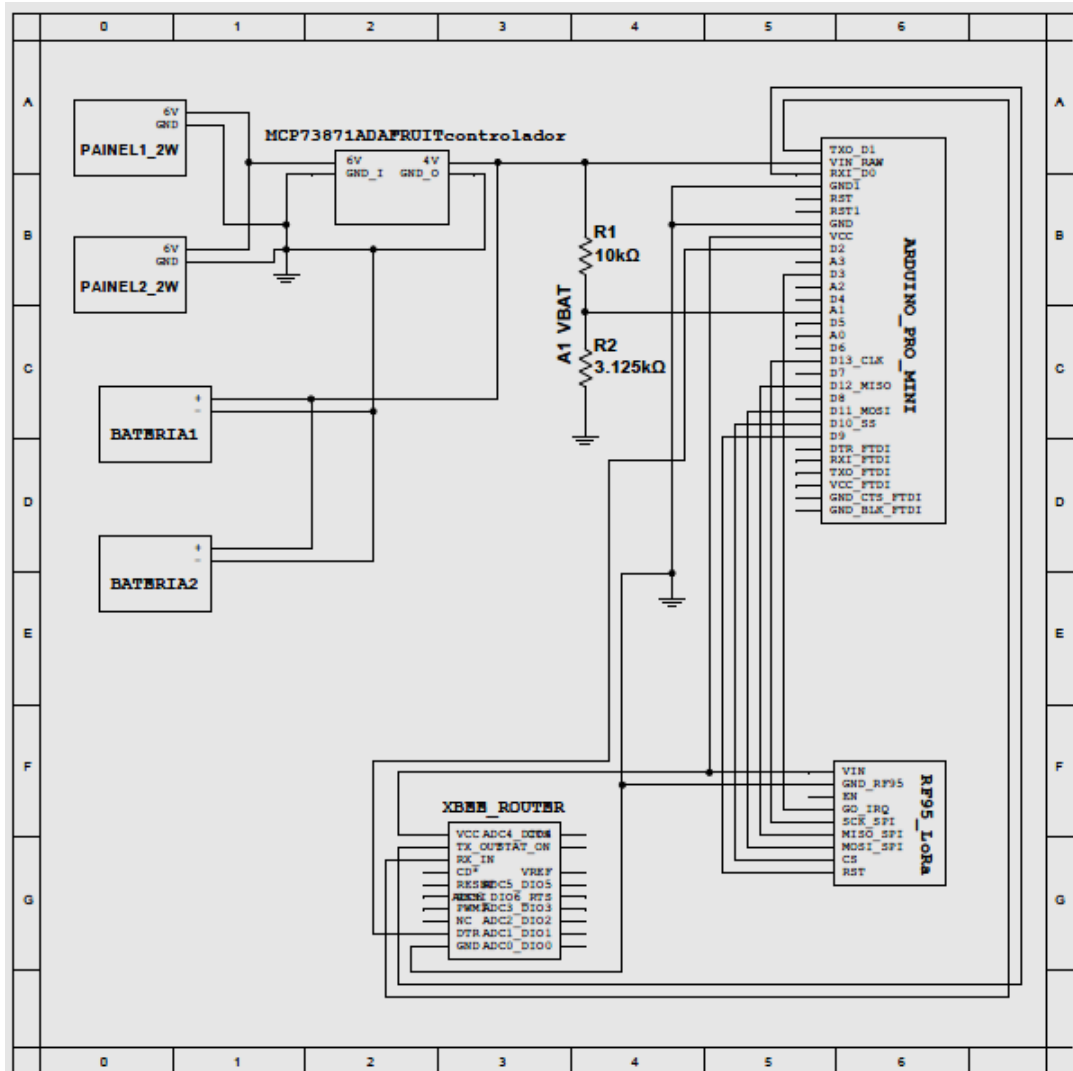


Figura 7.38 Esquemático gateway LoRa/Zigbee.

À semelhança do nó remoto, fez-se uso de uma placa de circuito pré-perfurada para o desenvolvimento do *hardware*. O *hardware* utilizado é o apresentado na figura 7.38, este consiste no microcontrolador, os rádios RFM9x LoRa e XBee Series 2 e um controlador de carga. Dimensionou-se um divisor resistivo para monitorizar o nível da bateria por leitura ADC. Utilizaram-se dois painéis solares de 2 W em paralelo como fonte de captação energética, estes destinavam-se a fornecer energia a duas baterias de lítio.

À semelhança dos módulos anteriores, resguardou-se a eletrónica numa caixa elétrica IP69, a qual possui três aberturas perfuradas. Estas se destinam às antenas dos módulos de comunicação e à saída de cablagem. A cablagem exterior, tal como para o nó remoto têm as características H07-V conferindo robustez nas conexões.

ANEXOS

Montou-se uma estrutura principal em madeira rígida na qual afixou-se a caixa elétrica na extremidade inferior e na extremidade superior instalaram-se os painéis fotovoltaicos. Houve o cuidado de distanciar a caixa elétrica e os painéis solares de modo que os últimos não constituíssem um obstáculo na recepção de sinal. Outro cuidado motivado pela qualidade de sinal consistiu em instalar a caixa elétrica de modo que as antenas ficassem posicionadas verticalmente devido a características de polarização de sinal. Ainda foi necessário construir uma estrutura de suporte de maior robustez para a instalação dos painéis e que proporcionasse um ângulo de captação da radiação adequado.

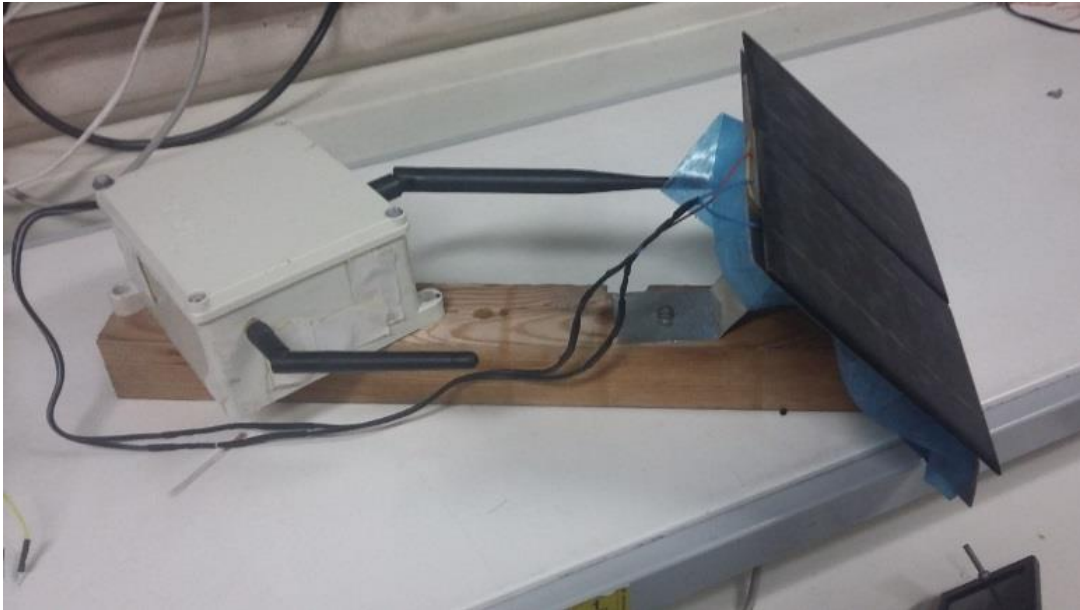


Figura 7.39 Gateway LoRa/Zigbee.



Figura 7.40 Gateway LoRa/Zigbee.

Código - controlo embebido

```

/* Codigo gateway:
* O gateway recebe as mensagens compactadas do no remoto terminal (RFM9x RF95 recetor)
* realiza o processamento e calculo dos niveis de CO (ppm):
* calculo dos niveis CO (ppm): CO(ppm)= (VomV-VREFTGS)/4.59484
* lido a tensao na bateria do gateway local BATV2
* os dados de BATV1,BATV2, VomV, CO (ppm) sao encapsulados na trama
* de emissao Zig Bee e entao sao transmitidos pelo modulo RF XBee router
* para o coordenador do laboratorio da universidade
* o microcontrolador arduino, o modulo RFM9x RF95 permanecem acordados,
* aciono o modo adormecido no modulo RF XBee
* enderecamento cliente-servidor no qual o gateway e configurado como servidor
* Atribuicao enderecos: cliente (terminal remoto 203), servidor (gateway 109) */
#include <SPI.h>
#include <RH_RF95.h>
#include <RHReliableDatagram.h>

// Atribuicao enderecos: cliente (terminal remoto 203), servidor (gateway 109)
#define TERMINAL_NODE 203
#define GATEWAY_NODE 109
/* CS D10 ARDUINO, RESET D9 ARDUINO, INTERRUPT PIN D3 ARDUINO */
#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 3
/* Change to 434.0 or other frequency, must match RX's freq!
ajuste para 868.0 MHz a frequencia de operacao na Europa */
#define RF95_FREQ 868.0
/* Singleton instance of the radio driver
instancia da biblioteca RadioHead para o modulo RFM9x
esta faz uso dos pinos CS e IRQ */
RH_RF95 rf95(RFM95_CS, RFM95_INT);
/* Class to manage message delivery and receipt, using the driver declared above
* classe responsavel pela rececao da mensagem e envio do pacote ACK */
RHReliableDatagram manager(rf95, GATEWAY_NODE);
/* parametros:buffer de dados, dimensao mensagem do terminal, endereco no terminal*/
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
uint8_t from;
unsigned long time; // contador temporal para o modo acordado

/* parametros sensor CO:
* sensor CO usado no fixo remoto: 1.468 nA/ppm ganho 3.13 fator 4.59484
* calculo dos niveis CO (ppm): CO(ppm)= abs((VomV-VREFTGS)/4.59484) */
#define VREFTGS 96 // tensao referencia sensor CO TGS 96 mV

```

ANEXOS

```
/* TRAMA A ENVIAR
 * informacao 0x7E – Byte de início da trama
 * 0x001E – Comprimento da mensagem (em bytes): contagem dos bytes entre os campos lenght e
checksum: adaptando a trama para 4 sensores 61 bytes 0x00 0x3D
 * 0x10 – Identificador de API para transmissão de dados, para o coordenador/router (PAN ID 99): trama
emissao Zig Bee
 * Numeracao da mensagem 0x00
 * Endereço geral do coordenador 64 bits 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
 * Endereço 16 bits destinatario 0xFF,0xFE
 * 0x00 – Número máximo de saltos de uma mensagem de broadcast (0 para número máximo de saltos);
 * 0x00 – mensagem broadcast: Opções de transmissão (0x08 – Multicast, caso contrário Unicast);
 * FORMATO DOS DADOS: ?Sensor1=1024&Sensor2=1024
 * MENSAGEM A TRANSMITIR: ?BATV1=valor1&BATV2=valor2&VomV=valor3&COTGSppm=valor4
 * dados 1: 0x3F (?) 0x42 (B) 0x41 (A) 0x54 (T) 0x56 (V) 0x31 (1) 0x3D (=) 0x00,0x00,0x00,0x00 (VALORES)
0x26 (&)
 * dados 2: 0x42 (B) 0x41 (A) 0x54 (T) 0x56 (V) 0x32 (2) 0x3D (=) 0x00,0x00,0x00,0x00 (VALORES) 0x26 (&)
 * dados 3: 0x56 (V) 0x6F (o) 0x6D (m) 0x56 (V) 0x3D (=) 0x00,0x00,0x00,0x00 (VALORES) 0x26 (&)
 * dados 4: 0x43 (C) 0x4F (O) 0x54 (T) 0x47 (G) 0x53 (S) 0x70 (p) 0x70 (p) 0x6D (m)
 * 0x3D (=) 0x00,0x00,0x00,0x00,0x00 (VALORES)
 * 0x00 - byte de checksum trama adaptada */
byte trama[] = {0x7E,0x00,0x3D,0x10,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0xFF,0xFE,0x00,0x00,0x3F,0x42,0x41,0x54,0x56,0x31,0x3D,0x00,0x00,0x00,0x00,0x26,
0x42,0x41,0x54,0x56,0x32,0x3D,0x00,0x00,0x00,0x00,0x26,0x56,0x6F,0x6D,0x56,0x3D,
0x00,0x00,0x00,0x00,0x26,0x43,0x4F,0x54,0x47,0x53,0x70,0x70,0x6D,0x3D,0x00,0x00,
0x00,0x00,0x00,0x00};

int xbeeSleep=2; // pino D2 DTR XBee
int i=0;
byte checksum = 0;
float BATV2=0; // tensao bateria gateway local

/***** SETUP *****/
void setup()
{
  /* configuracao dos pinos digitais: RESET LoRa, DTR XBee */
  pinMode(RFM95_RST,OUTPUT);
  pinMode(xbeeSleep,OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  digitalWrite(xbeeSleep, HIGH);
  // defino a referencia interna de leitura dos ADCs para os 1.1 V
  analogReference(INTERNAL);
  // inicializacao do serial monitor
  while (!Serial);
  Serial.begin(9600);
  delay(100);
  time=millis(); // ativacao do timer para o modo acordado
```

```

// manual reset
digitalWrite(RFM95_RST, LOW);
delay(10);
digitalWrite(RFM95_RST, HIGH);
delay(10);

// inicializacao modulo RF95
if (!manager.init())
{
  Serial.println(" fail ");
}
Serial.println(" OK "); // inicializacao completa

// Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
// ajuste para os 868 MHz
if (!rf95.setFrequency(RF95_FREQ))
{
  Serial.println(" fail ");
  while (1);
}
rf95.setFrequency(RF95_FREQ); // ajuste da frequencia para os 868 MHz

// Defaults after init are 434.0MHz, 13dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on
// The default transmitter power is 13dBm, using PA_BOOST.
// If you are using RFM95/96/97/98 modules which uses the PA_BOOST transmitter pin, then
// you can set transmitter powers from 5 to 23 dBm:
rf95.setTxPower(14); // configuracao da potencia de transmissao para 14 dBm
}

/***** GUARDA_DADOS *****/
void guarda_dados()
{
  /* nesta funcao os dados recebidos pelo recetor RFM9x RF95
  * sao processados e encapsulados na trama de emissao Zig Bee
  * e sao transmitidos na porta serie */
  // dados da tensao na bateria no remoto: BATV1
  trama[24]=buf[0];
  trama[25]=buf[1];
  trama[26]=buf[2];
  trama[27]=buf[3];
  // dados da tensao na bateria do gateway local: BATV2
  bateria_gateway();
  // dados da tensao de saida sensor CO: VomV
  trama[45]=buf[5];
}

```

ANEXOS

```
trama[46]=buf[6];
trama[47]=buf[7];
trama[48]=buf[8];
// agora a parte referentes aos dados dos niveis de CO para o sensor TGS 5042
leituradadosCO();

// Wake up XBee
digitalWrite(xbeeSleep, LOW);
delay(15);

// Cálculo do checksum
checksum = 0;
for ( i = 3; i < sizeof(trama)-1; i++)
{
    checksum+= trama[i]; // Soma os valores que contam para checksum
}
trama[sizeof(trama)-1] = 0xFF - checksum; // Realiza o complemento para 2
Serial.write(trama, sizeof(trama)); // Envia a trama pela porta série
delay(100);
digitalWrite(xbeeSleep, HIGH); // Sleep XBee
}

/***** bateria_gateway*****/
void bateria_gateway()
{
    /* nesta funcao e lido o nivel da bateria do gateway local
    * os dados sao convertidos e encapsulados na trama de emissao Zig Bee */
    int AUX1BATV2,AUX2BATV2; // variaveis auxiliares para BATV2
    char acBuffer[3+1]; // buffer de dados
    /* leitura ADC1: valor tensao V tensao bateria gateway local BATV2
    * e usada a referencia interna de leitura dos ADCs */
    for (int i = 0; i < 8; i++)
    {
        BATV2=(4.11*1.1*analogRead(A1))/1023;
    }

    // espera 10 ms devido ao processo de leitura dos ADCs com a referencia interna
    delay(10);
    // processo de conversao necessario devido ao tipo de dados
    AUX1BATV2=BATV2;
    AUX2BATV2=BATV2*100-AUX1BATV2*100;

    // os dados de BATV2 sao guardados no buffer auxiliar acBuffer
    sprintf(acBuffer,"%1d.%2d",AUX1BATV2,AUX2BATV2);
    trama[35]=acBuffer[0]; // parte inteira
```

ANEXOS

```
trama[36]=acBuffer[1]; // separador decimal

// agora vem a parte decimal: byte mais significativo
if(acBuffer[2]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuffer[2]=48; // preenche com 0 ASCII (48 decimal)
    trama[37]=acBuffer[2]; // guarda na mensagem
}
// agora vem a parte decimal: byte menos significativo
if(acBuffer[3]<49) // se for inferior a 1 em ASCII (49 decimal)
{
    acBuffer[3]=48; // preenche com 0 ASCII (48 decimal)
    trama[38]=acBuffer[3]; // guarda na mensagem
}

trama[37]=acBuffer[2]; // caso normal: byte mais significativo da parte decimal
trama[38]=acBuffer[3]; // byte menos significativo da parte decimal
}

/*****LEITURA DE DADOS CO *****/
void leituradadosCO()
{
    /* nesta funcao e feito o calculo dos niveis CO e o posterior preenchimento
    * na trama de emissao Zig Bee */
    int AUX1CO,AUX2CO; // variaveis auxiliares para CO
    int VomV=0; // variavel auxiliar para VomV
    float COTGS=0.0; // variavel auxiliar calculo niveis CO TGS
    /* criacao dos buffers temporarios:
    * primeiro guarda o valor de Vo, segundo guarda o valor de CO calculado */
    char Vo[3+1],acBuf[4+1];
    // o buffer auxiliar Vo e preenchido com os valores de VomV
    Vo[0]=buf[5];
    Vo[1]=buf[6];
    Vo[2]=buf[7];
    Vo[3]=buf[8];
    VomV=atoi(Vo); // os dados sao convertidos em valor int
    /* Tratamento dos valores de CO: equacao de calculo de COTGS:
    * CO(ppm)=(VomV-VREFTGS)/FATOR_SIMPLIFICADO 4.59484 */
    COTGS=(VomV-VREFTGS)/4.59484; // equacao de calculo de COTGS
    // processo de conversao necessario devido ao tipo de dados
    AUX1CO=COTGS;
    AUX2CO=COTGS*100-AUX1CO*100;
    // os dados de CO sao guardados no buffer auxiliar acBuf
    sprintf(acBuf,"%2d.%2d",AUX1CO,AUX2CO);
```

ANEXOS

```
// controlo dos dados
// byte mais significativo parte inteira
    if(acBuf[0]<49) // se for inferior a 1 em ASCII (49 decimal)
    {
        acBuf[0]=48; // preenche com 0 ASCII (48 decimal) por causa do NetBeans
        trama[59]=acBuf[0]; // guarda na trama emissao Zig Bee
    }
// byte menos significativo parte inteira
    if(acBuf[1]<49) // se for inferior a 1 em ASCII (49 decimal)
    {
        acBuf[1]=48; // preenche com 0 ASCII (48 decimal)
        trama[60]=acBuf[1]; // guarda na trama emissao Zig Bee
    }
// byte mais significativo parte decimal
    if(acBuf[3]<49) // se for inferior a 1 em ASCII (49 decimal)
    {
        acBuf[3]=48; // preenche com 0 ASCII (48 decimal)
        trama[62]=acBuf[3]; // guarda na trama emissao Zig Bee
    }
// byte menos significativo parte decimal
    if(acBuf[4]<49) // se for inferior a 1 em ASCII (49 decimal)
    {
        acBuf[4]=48; // preenche com 0 ASCII (48 decimal)
        trama[63]=acBuf[4]; // guarda na trama emissao Zig Bee
    }
// os dados de CO sao guardados na trama emissao Zig Bee
trama[59]=acBuf[0];
trama[60]=acBuf[1];
trama[61]=acBuf[2];
trama[62]=acBuf[3];
trama[63]=acBuf[4];
}
```

```
/******LOOP******/
void loop()
{
    // se houverem mensagens
    if (manager.available())
    {
        /* recebe a mensagem proveniente do no terminal parametros:
        * a mensagem de dados e guardada no buffer, dimensao do buffer
        * endereço do no terminal
        // Wait for a message addressed to us from the client*/
        if (manager.recvfromAck(buf, &len, &from))
        {
```

ANEXOS

```
// processo de rececao dos dados do no terminal remoto
Serial.print(" got request from TERMINAL NODE : 0x"); // notifica
Serial.print(from, HEX); // recebe o endereço terminal em hexadecimal
Serial.print(" : ");
Serial.println((char*)buf); // imprime a mensagem
Serial.print("RSSI: ");
Serial.println(rf95.lastRssi(), DEC); // valor RSSI
// funcao que guarda os dados na trama Zig Bee e os transmite via porta serie
guarda_dados();
Serial.println(); // muda de linha
Serial.print(time); // imprime o tempo do timer no modo acordado
delay(1000);
}
}
}
```

Anexo M – Dados extras – sistema de monitorização remoto

Esta seção apresenta dados extra referentes ao sistema de monitorização remoto. O gráfico da figura 7.41 apresenta dados de índice de radiação solar registados a 20 e 22 de Setembro de 2019, os quais foram registados no contexto da seção 4.4.3, aludindo à influência da radiação solar nas medições de monóxido de carbono pelo sensor Figaro TGS 5042. Quanto às tabelas 7.3 e 7.4, caracterizam os dados registados nos testes das comunicações a longo alcance com o RFM9x LoRa, os quais são descritos na seção 4.4.2. É de realçar, que os dados das tabelas estão assinalados a cores atendendo aos critérios descritos na referida seção. Os quais se tratam de níveis de RSSI e da taxa de sucesso de transmissão de pacotes.

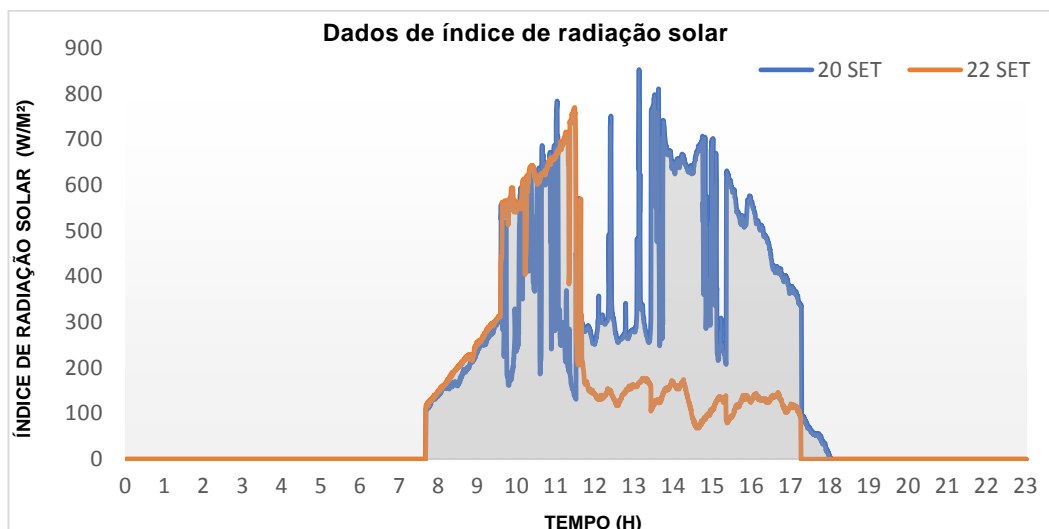


Figura 7.41 Dados de índice de radiação solar nos dias 20-09-2019 e 22-09-2019.

ANEXOS

Tabela 7.3 Dados dos pontos de medição com o recetor RFM9x LoRa.

Identificação ponto medição	Localização/pontos de referência	Linha de vista	Coordenadas espaciais (graus decimais)	Taxa de sucesso por 10 pacotes transmitidos (%)	Valor médio RSSI (dBm)	Distância entre emissor/recetor (km)
A0	UMA	-	32,65905 16,92426	-	-	-
A1	C.V.QUINTA	N	32,6647 16,92677	100	-126,3	0,67
A2	C.L.SEGUNDO	N	32,67043 16,92198	100	-128,1	1,28
A3	S.ROQUE IGREJA	N	32,66859 16,92308	100	-107,7	1,07
A4	LIVRAMENTO ASSECOS	N	32,66817 16,91547	100	-127,8	1,3
A5	E.MARMELEIROS	N	32,66917 16,90976	100	-106,5	1,77
A6	M.PALACE	N	32,67952 16,90172	100	-130,6	3,1
A7	Q.T.LUTA	N	32,68527 16,90088	100	-123,8	3,64
A8	T.LUTA C.PRETOS	N	32,68428 16,89993	100	-118	3,68
A9	C.P.ROMEIROS	S	32,68031 16,89065	100	-107,5	3,94
A10	P.GOLF MIRADOURO	N	32,66008 16,87445	100	-127,5	4,67
A11	PINACULO	N	32,64506 16,86868	80	-133,50	5,44
A12	PICO ALTO	S	32,69201 16,90105	100	-104,5	4,25
A13	POÇO NEVE	N	32,72437 16,9236	30	-139	7,24
A14	S.MARTINHO IGREJA	N	32,64887 16,94296	40	-138,5	2,1
A15	P.BARCELOS	N	32,659 16,93997	100	-120,4	1,47
A16	M.CASH	N	32,65321 16,92033	100	-112	0,75
A17	C.SAÚDE ST.ANTONIO	N	32,66365 16,93531	100	-109,2	1,16
A18	R.DR.JOÃO S.V.CAROÇO	N	32,65138 16,9187	100	-111,8	1
A19	CONTINENTE VIVEIROS	N	32,65898 16,91909	100	-131,4	0,49
A20	E.S.JOÃO GALP	N	32,65016 16,91881	100	-118,4	1,11
A21	A21 LA VIE	N	32,64768 16,91566	100	-129,3	1,50
A22	R.31 JANEIRO	N	32,65561 16,91552	100	-135	0,90
A23	APEL	N	32,65796 16,91626	100	-126,8	0,76
A24	S.ANTÓNIO IGREJA	N	32,66369 16,94083	100	-127,2	1,64
A25	P.OLIMPICAS S.ANTÓNIO	N	32,66474 16,93411	100	-106,4	1,12

ANEXOS

Tabela 7.4 Dados dos pontos de medição com o recetor RFM9x LoRa.

Identificação ponto medição	Localização/pontos de referência	Linha de vista	Coordenadas espaciais (graus decimais)	Taxa de sucesso por 10 pacotes transmitidos (%)	Valor médio RSSI (dBm)	Distância entre emissor/recetor (km)
A26	AV.MADALENAS	N	32,65748 16,92624	100	-96,2	0,26
A27	P.D.MARÍTIMO S.ANTÓNIO	N	32,67226 16,93494	100	-131,9	1,78
A28	PONTINHA	N	32,64187 16,90732	50	-137,2	2,48
A29	B.NOVA IGREJA	N	32,65786 16,88959	100	-128,4	3,26
A30	B.SUCESSO J.BOTÂNICO	N	32,66203 16,89396	100	-111,5	2,86
A31	CAIS FUNCHAL	N	32,64541 16,90908	60	-128,83	2,08
A32	AV.MAR PROMENADE	N	32,64586 16,90242	20	-136	2,52
A33	EST.BARREIROS	N	32,64431 16,92855	100	-125,50	1,68
A34	B.NOVA V.R	N	32,65708 16,88685	100	-115,9	3,52
A35	LOUROS	N	32,64845 16,88982	100	-129,5	3,44
A36	ROT.FUNDOA	N	32,66933 16,91826	90	-133,9	1,92
A37	E.FUNDOA	N	32,67558 16,91813	90	-127,9	1,27
A38	TUNEL NAZARÉ	N	32,64612 16,93172	100	-115,9	1,60
A39	MIRADOURO ACHADA	N	32,65717 16,91977	100	-128,1	0,47
A40	R.EST.U.AMERICA	N	32,64201 16,933	100	-131,4	2,05
A41	R.VIRTUDES L.MARITIMO	N	32,64701 16,93008	10	-137	1,44
A42	PILAR EDIFÍCIOS	N	32,65557 16,93067	100	-131,8	0,72
A43	R.VIRTUDES	N	32,64991 16,92962	100	-132,6	1,13