



Implementação de uma Rede de Sensores
Sem Fios Baseada em SoC Arm e ZigBee

Eládio Júnior

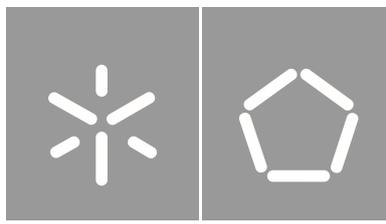
UMinho | 2019



Universidade do Minho
Escola de Engenharia

Eládio João Munar Júnior
**Implementação de uma Rede de Sensores
Sem Fios Baseada em SoC Arm e ZigBee**

dezembro de 2019



Universidade do Minho

Escola de Engenharia

Eládio João Munar Júnior

Implementação de uma Rede de Sensores Sem Fios Baseada em SoC Arm e ZigBee

Dissertação de Mestrado

Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação do(a)

Professor Doutor Jorge Cabral

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



**Atribuição-NãoComercial-Compartilhalgual
CC BY-NC-SA**

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Agradecimentos

As primeiras palavras de agradecimento vão sem dúvida alguma para os meus pais, Mateus Júnior e Elizeth Munar, por todo o apoio e suporte que sempre me deram ao longo do meu percurso académico e pessoal. Agradeço aos meus queridos irmãos Mavilde, Osvaldo, Hélio, Delcia, Juelson, Elídio, André, Ruth, Bárbara e Camilo que sem eles não conseguiria concluir esta etapa da minha vida.

Ao Professor Doutor Jorge Cabral pela orientação nesta dissertação e pela confiança depositada para a realização deste projeto.

Ao *Embedded Systems Research Group* do Departamento de Eletrónica Industrial da Universidade do Minho, que me acolheu e proporcionou todas as condições necessárias para a elaboração desta dissertação.

Ao pessoal do ESRG Lab -IB-S, Novais, RF, Mário, AC, Paulo, Eduardo, Cuca, Rui e Lobo, pela disponibilidade incansável, pela boa disposição e bom ambiente criado no laboratório.

Quero fazer um agradecimento especial à família Rosende Salgado, pelo apoio e suporte que me deram ao longo da minha estadia em Portugal.

Aos meus companheiros e amigos das residências universitárias de Azurém e dos Combatentes que desde cedo me acompanharam no meu percurso académico, com um agradecimento especial ao companheiro e amigo Andersen Bond.

Não podia deixar de agradecer ao pessoal do Judo do Vitória, em especial ao Mestre Emídio, pelo apoio e suporte, cada um à sua maneira.

Por último, mas não menos importante, à minha companheira Dilkarina, pelo carinho e atenção, por todo apoio e incentivo nos momentos menos bons.

A todos, muito obrigado!

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

As vias de comunicação terrestres são elementos fulcrais para o desenvolvimento das sociedades. A sinistralidade rodoviária é uma das principais ameaças à integridade dos elementos que constituem o sistema rodoviário. No sentido de minimizar este impacto negativo, surge esta dissertação cujo objetivo é apresentar um sistema de monitorização que informa com precisão a ocorrência de colisões de veículos com os guardas de segurança das autoestradas e alertar as entidades responsáveis pela gestão do sistema em tempo real sobre o sucedido.

A presente dissertação está orientada em melhorar o projeto, *Sustainable Infrastructure Management System* (SustIMS), desenvolvido pelo grupo de investigação *Embedded System Research Group* (ESRG) do Centro ALGORITMI da Universidade do Minho em co-promoção com a empresa Ascendi. Através da implementação de uma rede de sensores sem fios baseada em ZigBee, recorrendo a um *System-on-Chip* (SoC) com *transceiver Radio Frequency* (RF) da família Arm Cortex-M3, equipado com sensor acelerómetro *Microelectromechanical Systems* (MEMS).

O ZigBee, é um protocolo que assenta sobre as camadas definidas no *standard* IEEE 802.15.4, rede de baixo débito, mas com longa autonomia e baixa complexidade. De forma a cumprir com esses requisitos foi utilizado um processador Arm Cortex-M3 e um acelerómetro MEMS, os quais são caracterizados pelo seu baixo consumo energético.

Os resultados desta dissertação são: (1) melhorias na solução atual com o principal foco de reduzir o consumo de energia do sistema; (2) o aumento do alcance da comunicação dos dispositivos; (3) o *porting* da aplicação para nova versão da Z-Stack; e (4) compatibilização do sistema com os microserviços *web*, desenvolvidos pelo ESRG.

Palavras-chave: *Wireless Sensor Network*, Deteção de colisões, Consumo energético, Z-Stack, *Internet of Things*.

Abstract

Road communication is an essential component when it comes to society development. Road accidents are one of the main threats to the integrity of the elements that compose the road system. In order to minimize this negative impact, this dissertation aims to present a monitoring system that informs, with precision, the occurrence of vehicle collision against highway security rails and alert the entities responsible for the real-time management of the system about the event.

The present dissertation aims to improve the project, Sustainable Infrastructure Management System, developed by the University of Minho's Embedded System Research Group (ESRG) in co-promotion with the company Ascendi. By implementing a ZigBee-based wireless sensor network (WSN) using an Arm Cortex-M3 family Radio Frequency (RF) transceiver System-on-Chip (SoC) equipped with an Microelectromechanical Systems (MEMS) accelerometer sensor.

ZigBee is a protocol based on the layers defined in the IEEE 802.15.4 standard, low speed network, but with long autonomy and low complexity. In order to meet this requirements an Arm Cortex-M processor and MEMS accelerometer were used, which are characterized by their low power consumption.

The results of this dissertation are: (1) improvements on the current solution with the main focus on reducing the system energy consumption; (2) increasing the communication range of the devices; (3) the application porting for the new version of Z-Stack; (4) System compatibility with the microservices developed by the ESRG.

Keywords: Wireless Sensor Network, Collision detection, Energy consumption, Z-Stack, Internet of Things.

Conteúdo

Agradecimentos	iii
Resumo	v
Abstract	vi
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	2
1.3 Objetivos	3
1.4 Estrutura da Dissertação	3
2 Estado da Arte	5
2.1 Sistema de Monitorização	5
2.1.1 <i>Dynamic Monitoring of Guardrails</i>	5
2.1.2 <i>IoT System for Remote Monitoring of Bridges</i>	8
2.1.3 <i>Sustainable Infrastructure Management System</i>	11
2.1.4 Comparação entre os vários Sistemas de Monitorização	14
2.2 <i>Wireless Sensor Network (WSN)</i>	15
2.2.1 Aplicações e Características das WSN	15
2.2.2 Arquitetura de um Nó Sensorial	16
2.3 <i>Standard IEEE 802.15.4</i>	19
2.3.1 Componentes e Topologias de Rede	21
2.3.2 Camada Física PHY	23
2.3.3 Camada MAC	25
2.4 ZigBee	29
2.4.1 Aplicações e Características da Rede ZigBee	29

2.4.2	Arquitetura Protocolar	32
2.4.3	ZigBee 3.0	35
2.5	ZigBee <i>System-on-Chip</i>	37
2.5.1	<i>Microchip System-on-Chip</i> ATSAMR21G18A	38
2.5.2	<i>NXP System-on-Chip</i> JN5169	39
2.5.3	<i>Texas Instruments</i> SoC CC1352P	42
2.6	<i>Stacks da Texas Instruments</i>	44
2.6.1	SimpliciTI	44
2.6.2	Z-Stack	46
2.7	Resumo	48
3	Especificação do Sistema	49
3.1	Visão Geral do Sistema	49
3.2	Requisitos do Sistema	49
3.3	Arquitetura do Sistema	51
3.4	Casos de Uso do Sistema	52
3.5	Especificação do <i>Hardware</i>	53
3.5.1	<i>System-on-Chip</i> (SoC)	53
3.5.2	<i>Gateway</i>	55
3.5.3	Acelerómetro	56
3.5.4	Bateria	57
3.6	Especificação do <i>Software</i>	57
3.6.1	Z-Stack 3.0.2	58
3.6.2	Formato de Dados JSON	59
3.6.3	Base de Dados MongoDB	59
3.6.4	Microserviços ArchNET	60
3.6.5	Aplicação do Nó Sensor	61
3.6.6	Aplicação do Nó Coordenador	63
3.6.7	Aplicação do <i>Gateway</i>	64
3.7	Resumo	64

4	Desenvolvimento do Sistema	65
4.1	<i>Hardware</i> Desenvolvido	65
4.1.1	Nó Sensor	65
4.1.2	Nó Coordenador	68
4.2	<i>Software</i> Desenvolvido	69
4.2.1	<i>Access Point Task</i>	70
4.2.2	<i>End Device Task</i>	72
4.2.3	<i>Software</i> Desenvolvido para o <i>Gateway</i>	73
4.3	Resumo	75
5	Testes e Resultados	76
5.1	Cenários de Testes	76
5.2	Testes Desenvolvidos	77
5.2.1	Teste ao Protocolo Porta Série	77
5.2.2	Obtenção de Indicadores de QoS (RSSI e PER)	78
5.2.3	Teste ao <i>Gateway</i>	79
5.2.4	Teste de Integração do Sistema	80
5.3	Medição de Consumo	83
5.3.1	Consumo Médio de Corrente no Estado <i>Deep Sleep</i>	83
5.3.2	Consumo Médio de Corrente Durante o Envio do <i>Keep Alive</i>	84
5.3.3	Consumo Médio de Corrente do Acelerómetro em Diferentes Modos	86
5.3.4	Estimativa de Duração da Bateria	87
5.3.5	Resumo	87
6	Conclusões e Trabalho Futuro	89
6.1	Conclusões	89
6.2	Trabalho Futuro	90
	Bibliografia	92
	Anexo A Diferentes Modos de Comissionamento	99
	Anexo B Esquemáticos do PCB do Nó Coordenador	101

Lista de Figuras

1.1	Gráfico estatístico de acidentes de viação com vítimas em autoestradas.	2
2.1	Guarda de segurança utilizada nos testes do sistema.	6
2.2	Arquitetura do nó sensor utilizado.	7
2.3	Protótipo do nó sensor.	7
2.4	Testes ao subsistema.	8
2.5	Arquitetura geral do sistema.	9
2.6	Arquitetura proposta para a monitorização do comportamento estrutural de pontes. . . .	10
2.7	Arquitetura do nó sensor para deteção de colisões de veículo com as guardas de segurança.	11
2.8	Arquitetura geral do sistema.	12
2.9	Nó ED instalado na caixa de proteção.	13
2.10	Interface via navegador web de uma WSN instalada.	14
2.11	Arquitetura <i>multihop</i> , típica de uma <i>Wireless Sensor Network</i> (WSN).	15
2.12	Análise SWOT da aplicação em tempo real da WSN.	17
2.13	Arquitetura típica de <i>hardware</i> de um nó sensorial.	18
2.14	Arquitetura típica de <i>software</i> de um nó sensorial.	18
2.15	Opções de <i>design</i> para arquiteturas heterogéneas.	19
2.16	Comparação de <i>standards</i> de redes WPAN e WLAN relativamente a taxa de transferência, complexidade de implementação e consumo de energia.	20
2.17	Topologias de rede.	22
2.18	Canais de frequência do IEEE 802.15.4.	24
2.19	Banda de frequências e débitos do IEEE 802.15.4.	25
2.20	Modos de operação da camada MAC do IEEE 802.15.4.	26
2.21	Comunicação de um dispositivo para o coordenador.	27
2.22	Estrutura da supertrama de modo <i>beacon-enabled</i>	27
2.23	Formato da trama de dados.	28

2.24	Rede ZigBee.	31
2.25	Aplicações da rede ZigBee.	31
2.26	Arquitetura do protocolo de comunicação IEEE 802.15.4/ZigBee.	32
2.27	Formatos das mensagens da rede ZigBee.	36
2.28	Procedimento de atualização da chave de ligação do TC.	37
2.29	Diagrama de blocos típico de um RF-SoC.	38
2.30	<i>System-on-chip</i> ATSAMR21G18A da <i>Microchip</i> no <i>package</i> QFN48.	38
2.31	Kits de desenvolvimento SAM R21 Xplained Pro e ATSAMR21ZLL-EK da <i>Atmel</i>	40
2.32	<i>System-on-chip</i> JN5169 da <i>NXP Semiconductors</i> no <i>package</i> HVQFN40.	41
2.33	Kits de desenvolvimento JN5169XK010 e OM15020 <i>Dongle</i> da <i>NXP</i>	42
2.34	<i>System-on-chip</i> CC1352P da <i>Texas Instruments</i> no <i>package</i> VQFN48.	42
2.35	<i>Kits</i> de desenvolvimento LAUNCHXL-CC1352P e LAUNCHXL-CC1352P-2 da TI.	44
2.36	Comparação entre várias soluções sem fios da TI.	45
2.37	Arquitetura geral da <i>Stack SimpliciTI</i>	45
2.38	formato da trama SimpliciTI.	46
2.39	Interação da Z-Stack e camada de aplicação com as camadas OSAL e HAL.	47
2.40	Diagrama de blocos do processo de de comunicação da Z-Stack.	48
3.1	Visão geral do sistema.	50
3.2	Diagrama de blocos do sistema.	52
3.3	Diagrama de casos de uso do sistema	52
3.4	SoC CC2538 da Texas Instruments no package QFN56.	53
3.5	Módulo com SoC CC2538 e o <i>Range Extender</i> CC2592.	55
3.6	Placa de desenvolvimento RaspBerry Pi 3 <i>model/B</i>	56
3.7	Acelerómetro MEMS LIS331x.	57
3.8	Bateria SAFT LS17500.	57
3.9	Exemplo de um documento em formato JSON.	59
3.10	Estrutura básica da base de dados MongoDB.	60
3.11	Arquitetura do sistema ArchNET.	61
4.1	Esquemático do sistema de gestão de energia.	66
4.2	Esquemático do módulo CC2538 / CC2592.	67

4.3	Visualizações em 3D do PCB.	68
4.4	Visualização da caixa Hammond com IP66 e do PCB assembled.	68
4.5	Visualização em 3D do PCB do primeiro protótipo.	69
4.6	Visualizações em 3D do PCB.	70
4.7	Visualização da caixa e do PCB assembled.	70
4.8	Diagrama de funcionamento do nó coordenador.	71
4.9	Diagrama de funcionamento do nó sensor.	73
4.10	Diagrama de funcionamento do módulo acelerómetro.	74
4.11	Diagrama de funcionamento do <i>software</i> do <i>gateway</i>	75
5.1	Configuração do <i>software</i> CoolTerm	78
5.2	Mensagem de colisão e de KA reencaminhadas pelo coordenador via porta série.	78
5.3	Distância entre os dispositivos para as medições do alcance.	79
5.4	Configuração do <i>software</i> PuTTY.	80
5.5	Inicialização automática da <i>script</i> que implementa a interface <i>gateway</i>	80
5.6	Estado da base de dados <i>MongoDB</i>	80
5.7	mensagens enviadas pelo <i>gateway</i> para os microserviços.	81
5.8	Arquitetura geral, teste de integração.	81
5.9	Valores obtidos pelo microserviços ArchNET.	82
5.10	Esquema de ligação para medição de consumo em <i>deep sleep</i>	83
5.11	Esquema de ligação para medição de consumo em KA.	84
5.12	Gráfico de corrente elétrica calculada no estado de KA.	85
5.13	Esquema de ligação para medição de diferentes modos de consumo <i>low power</i> do acelerómetro.	86
A.1	Diferentes modos de comissionamento.	100

Lista de Tabelas

2.1	Comparação entre os quatro sistemas de monitorização.	14
2.2	Comparação entre uma LR-WPAN e o protocolo Bluetooth e Wi-Fi.	20
5.1	Especificação dos dispositivos utilizados no teste de integração	82
5.2	Consumo de corrente elétrica no estado de <i>deep sleep</i> (PM3) com acelerómetro em <i>power down</i>	84
5.3	Consumo de corrente elétrica do nó sensor em <i>deep sleep</i> (PM3).	84
5.4	Consumo médio de corrente elétrica em modo KA.	85
5.5	Consumo de energia durante a transmissão da mensagem.	86
5.6	Consumo médio de corrente do nó em diferentes modos de operação do acelerómetro.	86
5.7	Consumo médio do nó em diferentes modos de operação do acelerómetro.	88

Lista de Listagem

- 4.1 Informações relativas à rede. 72
- 4.2 Informações relativas à aplicação. 72

Lista de Equações

3.1	Equação de cálculo do valor da aceleração em g.	63
5.1	Equação de cálculo da quantidade média de corrente elétrica consumida durante 1 hora.	87
5.2	Equação de cálculo da estimativa do tempo de vida do nó.	87

Lista de Siglas e Acrónimos

6LoWPAN *IPv6 over Low-Power Wireless Personal Area Networks.*

ADC *Analog-to-Digital Converter.*

AES *Advanced Encrypt Security.*

AF *Application Framework.*

AP *Access Point.*

APDU *Application Support Sub-Layer Protocol Data Unit.*

API *Application Programming Interface.*

APL *Application.*

APO *Application Objects.*

Arm *Advanced risc machine.*

BaSta *Barriera Stradale Attiva.*

BDB *Base Device Behavior.*

BI *Beacon Interval.*

BLE *Bluetooth Low Energy.*

BO *Beacon Order.*

BPSK *Binary Phase-Shift Keying.*

BSN *Body Sensor Network.*

CAP *Contention Access Period.*

CCA *Clear Channel Assessment.*

CRC *Cyclic Redundancy Check.*

CFP *Contention Free Period.*

CSMA-CA *Carrier Sense Multiple Access with Collision Avoidance.*

ED *End Device.*

ESRG *Embedded System Research Group.*

FCS *Frame Check Sequence.*

FFD *Full-Function Device.*

FIFO *First In First Out.*

FTDI *Future Technology Devices International.*

GFSK *Gaussian Frequency-Shift Keying.*

GP *Green Power.*

GPIO *General Purpose Input / Output Pins.*

GTS *Guaranteed Time Slots.*

HAL *Hardware Abstraction Layer..*

HTTP *Hypertext Transfer Protocol.*

I2C *Inter-Integrated Circuit.*

IDE *Integrated Development Environment.*

IEEE *Institute of Electrical and Electronics Engineers.*

IoT *Internet of Things.*

IP *Ingress Protection.*

IRF *Internacional Road Federation.*

JSON *JavaScript Object Notation.*

JTAG *Joint Test Action Group.*

KA *Keep Alive.*

LAN *Local Area Network.*

LNA *Low Noise Amplifier.*

LR-WPAN *Low-Rate Wireless Personal Area Networks.*

LQI *Link Quality Indicator.*

MAC *Medium Access Control.*

MEMS *Microelectromechanical Systems.*

MFR *MAC footer.*

MHR *MAC header.*

NLDE *Network Layer Data Entity.*

NLME *Network Layer Management Entity.*

NPDU *Network Protocol Data Unit.*

NWK *Network.*

O-QPSK *Offset Quadrature Phase-Shift Keying.*

OSAL *Operating System Abstraction Layer.*

OSI *Open System Interconnect.*

OTA *Over the Air.*

PA *Power Amplifier.*

PAN *Personal Area Network.*

PCB *Printed Circuit Board.*

PDU *Protocol Data Unit.*

PER *Packet Error Rate.*

PHY *Physical.*

PWM *Pulse Width Modulation.*

QoS *Quality of Service.*

QREN *Quadro de Referência Estratégica Nacional.*

RAM *Random Access Memory.*

REST *Representational State Transfer.*

RF *Radio Frequency.*

RF-SoC *Radio Frequency System-on-Chip.*

RFD *Reduced-Function Device.*

RSSI *Received Signal Strength Indication.*

SAP *Service Access Point.*

SD *Superframe Duration.*

SO *Superframe Order.*

SoC *System-on-Chip.*

SPI *Serial Peripheral Interface.*

SustIMS *Sustainable Infrastructure Management System.*

ST *STMicroelectronics.*

SWOT *Strengths Weaknesses Opportunities Threats.*

TC *Trust Center.*

TCLK *Trust Center Link Key Updates.*

TCP *Transmission Control Protocol.*

TI *Texas Instruments.*

UART *Universal Asynchronous Receiver / Transmitter.*

USB *Universal Serial Bus.*

WLAN *Wireless Local Area Network Wireless.*

WPAN *Wireless Personal Area Network.*

WSN *Wireless Sensor Network.*

ZC *ZigBee Coordinator.*

ZCL *ZigBee Cluster Library.*

ZDO *ZigBeeDevice Object.*

ZED *ZigBee End Device.*

ZLO *ZigBee Lighting and Occupancy.*

ZR *ZigBee Router.*

Capítulo 1

Introdução

Neste capítulo é apresentado o enquadramento e as motivações da dissertação, bem como os objetivos propostos para a elaboração deste trabalho. Para além disso, também é feita a descrição desta dissertação.

1.1 Enquadramento

As vias de comunicação terrestres são elementos fulcrais para o desenvolvimento das sociedades, é responsabilidade de todos contribuir para sua preservação. A sinistralidade rodoviária é uma das principais ameaças à integridade dos elementos que constituem o sistema rodoviário. Em 2017, a sinistralidade rodoviária em Portugal teve um impacto económico e social equivalente a 2,3 mil milhões de euros [1]. No mesmo período, foram registados 51 vítimas mortais na sequência de desastres nas estradas ocorridos em autoestradas [2]. Em 2018, Portugal teve uma média de 1,8 mortes diárias causadas pela sinistralidade rodoviária. A Autoridade Nacional de Segurança Rodoviária revelou que, as colisões foram as situações que mais vítimas provocaram, seguido dos despistes e atropelamentos [3]. Na Figura 1.1 é possível observar o gráfico estatístico de acidentes de viação com feridos ou mortos em autoestradas [4].

No sentido de minimizar os problemas acima descritos, surge esta dissertação, cujo objetivo é apresentar um sistema de monitorização, que pode informar com precisão a ocorrência de colisões de veículos com os guardas de segurança das autoestradas e alertar as entidades responsáveis pela gestão do sistema em tempo real sobre o sucedido. A presente dissertação está orientada em melhorar o projeto, SustIMS [5], que será abordada em capítulos futuros, desenvolvida pelo grupo de investigação ESRG do Centro ALGORITMI da Universidade do Minho [6] em co-promoção com a empresa Ascendi [7].

Sistemas de monitorização baseados em videovigilância podem ser altamente dispendiosos em relação aos sistemas baseados em *Wireless Sensor Network* (WSN). Assim surgiu a necessidade da criação

de uma WSN que, através de nós sensoriais fixados às guardas de segurança da autoestrada, equipados com sensores de aceleração, responsáveis por medir a aceleração dinâmica resultante de movimento ou choque. Quando detetar uma aceleração superior aos limites previamente definidos, o nó sensor envia uma mensagem de colisão para estação base. Esta, por sua vez é responsável por reencaminhar a mensagem para os microserviços. A partir do momento que as informações estejam acessíveis na base de dados, a entidade responsável pela gestão do sistema, tem acesso em tempo-real à informação processada, através de uma aplicação.

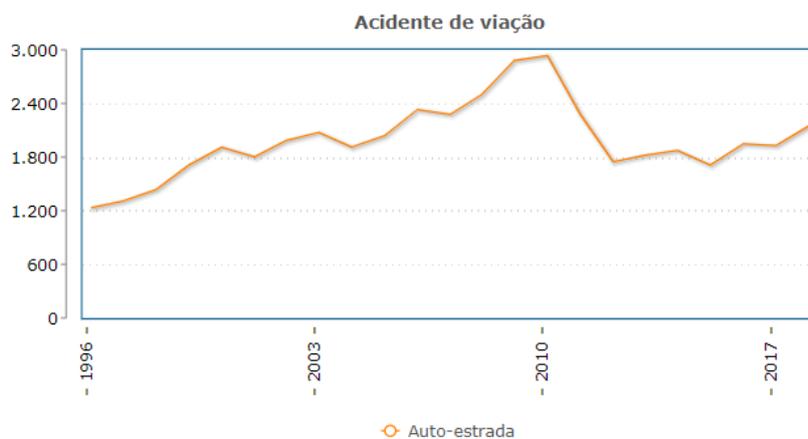


Figura 1.1: Gráfico estatístico de acidentes de viação com vítimas em autoestradas.

1.2 Motivação

A realização deste projeto permitiu adquirir competências técnicas e científicas, sobre o desenvolvimento de sistemas de monitorização baseados em redes de sensores sem fios. Para além disso, originou propostas de melhorias ao sistema SustlMS, que é um produto de grande envergadura à escala mundial, com prémios, distinções e *papers*. Em 2014 foi um dos 20 finalistas no concurso europeu "Texas Instruments: Innovation Challenge Europe Analog Design Contest 2014", com o projeto "HighWayMon" [8], no mesmo ano submeteu no IEEE o artigo "A Wireless Sensor Network for Collision Detection on Guardrails" [9]. Em 2016 submeteu um outro artigo no IEEE "An IoT-Based System for Collision Detection on Guardrails" [10]. Em 2017 foi distinguido pela *Internacional Road Federation* (IRF) como o melhor projeto mundial na categoria de "Preservação de Ativos & Gestão da Manutenção" [11].

O uso da tecnologia para o desenvolvimento de soluções para o bem-estar das sociedades, é sempre uma motivação extra para os projetos de Engenharia.

1.3 Objetivos

Neste trabalho pretende-se sugerir melhorias na solução atual com o principal objetivo de reduzir o consumo de energia do sistema, aumentar a distância máxima entre o nó sensor e o coordenador da rede de sensores sem fios, fazer o *porting* da aplicação para nova versão da Z-Stack da *Texas Instruments* (TI) e compatibilizar o sistema com os microserviços *web* desenvolvidos pelo ESRG.

Os objetivos acima referidos podem ser detalhados como:

- Fazer o *porting* da aplicação para a Z-Stack 3.0.2 da TI;
- Reformular o *hardware* / *software* do nó sensor e do coordenador;
- Reduzir o consumo energético do nó sensor;
- Criar uma interface entre a WSN e os microserviços *web*;

1.4 Estrutura da Dissertação

Neste capítulo é feita uma pequena introdução onde são apresentadas as motivações e objetivos para elaboração de um sistema de deteção de colisões com as guardas de segurança. Para além disso, também é apresentado o enquadramento e a estrutura desta dissertação.

No próximo capítulo, o capítulo 2, será apresentado o estado da arte relativamente algumas abordagens existentes que visam a detetar e reportar colisões de veículos com as guardas de segurança. Para além disso, são também apresentados o estado da arte relativo às principais tecnologias utilizadas neste projeto.

De seguida, no capítulo 3 é apresentado as especificações do *hardware* e do *software*, sendo feito um levantamento dos requisitos do sistema. É feita também uma breve apresentação do funcionamento do sistema.

No capítulo 4 é apresentado o modo como os principais componentes do sistema foram implementados. É demonstrado o *hardware* desenvolvido nas suas várias versões, bem como as decisões tomadas relativamente ao consumo energético. Por fim, as principais funcionalidades implementadas por *software*.

No capítulo 5 são apresentados testes de validação a que o sistema foi submetido. Foram realizados testes a cada componente do sistema e por último foi realizado um teste de integração de todo o sistema.

Por último, no capítulo 6 são apresentadas as principais conclusões e algumas sugestões de trabalho futuro.

Capítulo 2

Estado da Arte

Neste capítulo será apresentada uma visão geral da caracterização de uma WSN e do IEEE 802.15.4. De seguida serão analisadas as principais redes e protocolos, assentam sobre as camadas definidas no IEEE 802.15.4. Pretende-se assim analisar algumas soluções que atualmente estão a ser desenvolvidas e/ou em fase de estudo, e o projeto SustIMS, que originou essa dissertação.

2.1 Sistema de Monitorização

Nesta secção serão apresentadas algumas soluções propostas desenvolvidas e/ou em fase de estudo sobre a monitorização de colisões com as guardas de segurança.

2.1.1 *Dynamic Monitoring of Guardrails*

No âmbito do projeto *Barriera Stradale Attiva* (BaSta), este sistema foi proposto pela Universidade de Sannio em Benevento, Itália, com o suporte técnico da empresa italiana IMEVA S.p.A. [12]. Este sistema tem como objetivo criar uma WSN, capaz de monitorizar e analisar o comportamento das estruturas devido a dano ou defeito local. O sistema também pode ser útil para detetar em tempo real colisões de veículos em guardas de segurança, a cedência da guarda de segurança foi cortesia da IMEVA S.p.A. para realizar os testes do subsistema, como pode ser visualizado na Figura 2.1 [12].

Para detetar as colisões através dos nós fixados às guardas de segurança, são utilizados acelerómetros ADXL375 da *Analog Devices*. Este acelerómetro MEMS é apropriado para medir a aceleração estática da gravidade em aplicações de sensibilidade de inclinação, bem como aceleração dinâmica resultante de movimento ou choque. É caracterizado pelo seu baixo consumo energético (35 μA em modo de medição e 0.1 μA em *standby*), com alta resolução de 16 *bits* e que pode medir até ± 200 g. A saída digital pode ser



Figura 2.1: Guarda de segurança utilizada nos testes do sistema.

acessada pelas interfaces SPI ou I2C. Para minimizar as intervenções do processador utiliza um *buffer* FIFO.

A plataforma utilizada para unidade controladora é o Teensy 3.1, baseado no MK20DX256 traz um microprocessador Arm Cortex-M4 de 32 *bits* que permite um alto nível de poder de processamento. Esta placa de desenvolvimento traz também algumas outras características como 34 pinos de entrada e saída digital, 21 entradas analógicas de 12 *bits* de resolução, 12 saídas PWM, com as interfaces *standard* (3 UARTs, 2 I2C, 1 SPI) para comunicar com outros microprocessadores ou sensores. Todas estas funcionalidades encaixadas numa pequena placa de (3,5 x 1,7 cm). Pode ser programado com qualquer editor usando linguagem de programação C ou pode se instalar a biblioteca *Teensyduino* no ambiente de desenvolvimento integrado (IDE) do Arduino e escrever *sketches* para o Teensy. A arquitetura dos nós para medir a aceleração estática ou dinâmica é apresentada na Figura 2.2, onde não é só é possível visualizar os sensores, como também os restantes componentes requeridos para o armazenamento de dados, para as comunicações e para alimentação do nó [12].

A arquitetura apresentada na Figura 2.2, inclui um *modem* ZigBee permite efetuar a comunicação entre os nós, possibilitando a transmissão de dados das acelerações pelo meio sem fios. O subsistema *energy harvesting system* tem a função de recolher energia do ambiente para alimentar o nó, não foi implementado, desse modo ficou para trabalhos futuros [12]. O *microSD* permite o armazenamento local de dados. Na Figura 2.3 é possível observar nó protótipo colocado numa caixa plástica de dimensão de 100 x 100 x 50 mm³, no canto superior direito está ilustrado o sensor utilizado e no canto inferior direito está ilustrado o nó em cima do equipamento de teste de vibração [12].

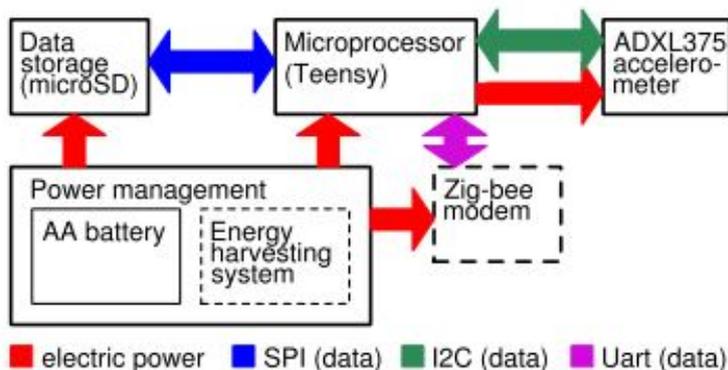
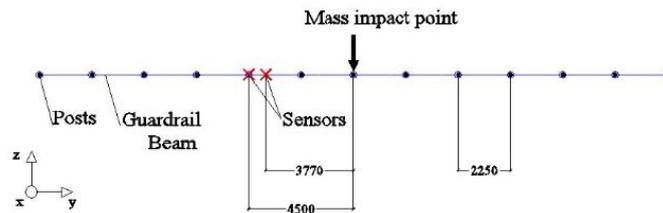


Figura 2.2: Arquitetura do nó sensor utilizado.

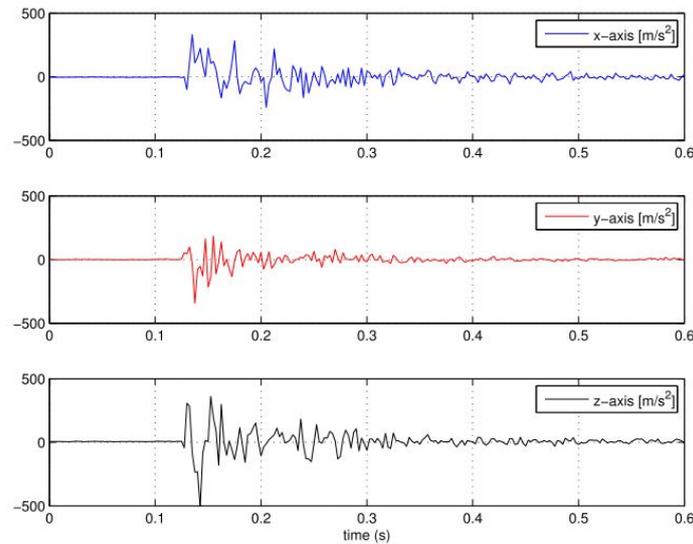


Figura 2.3: Protótipo do nó sensor.

O cenário escolhido para realizar os testes de impacto foi em laboratório com uma guarda de segurança, que pode ser observada na Figura 2.1 [12]. Em seguida, dois nós protótipos foram fixados nas guardas de segurança, de acordo com o *layout* da Figura 2.4 [12]. Uma massa de 1000 kg de cor amarela, como representado na Figura 2.1, foi afastada para 2 m e deixada cair no centro da guarda de segurança [12]. O sensor do nó foi ajustado com uma frequência de amostragem de 400 Hz. O gráfico temporal da aceleração medida em três eixos, do sensor colocado a 3,770 m, é ilustrado na Figura 2.4a [12]. A aceleração máxima foi cerca de 50 g, ao longo do eixo z (a orientação do eixo é representada na Figura 2.4b) [12]. Os dois nós protótipos não foram sincronizados devido à sua curta distância entre ambos (cerca de 730 mm).



(a) Layout da guarda de segurança com os nós protótipos e a massa. As distâncias são representadas em mm.



(b) Medição de aceleração de 3 eixos sobre a guarda de segurança.

Figura 2.4: Testes ao subsistema.

2.1.2 IoT System for Remote Monitoring of Bridges

O artigo [13] apresenta uma solução para monitorizar o comportamento estrutural de pontes, baseado em *Internet of Things* (IoT), o sistema proposto visa registrar: (1) a colisão de veículos com guardas de segurança, (2) vários parâmetros ambientais, (3) a expansão de juntas estruturais e (4) a ação do tráfego. Todos esses dados são recolhidos por um *gateway* e enviados para um servidor ligado à internet para processamento em tempo real das informações recebidas. Na Figura 2.5 é ilustrada a arquitetura geral do sistema proposto [13].

O nó sensor para a deteção de colisão de veículo em guarda de segurança visa classificar o impacto em três níveis baixo, médio e alto, de acordo com a magnitude da aceleração devido ao impacto. Além disso, quando um impacto aparece, ele envia ao *gateway* um alerta, via Wi-Fi, com as informações relacionadas ao nível de resistência ao impacto. Esses nós são fixados nas guardas de segurança da ponte.

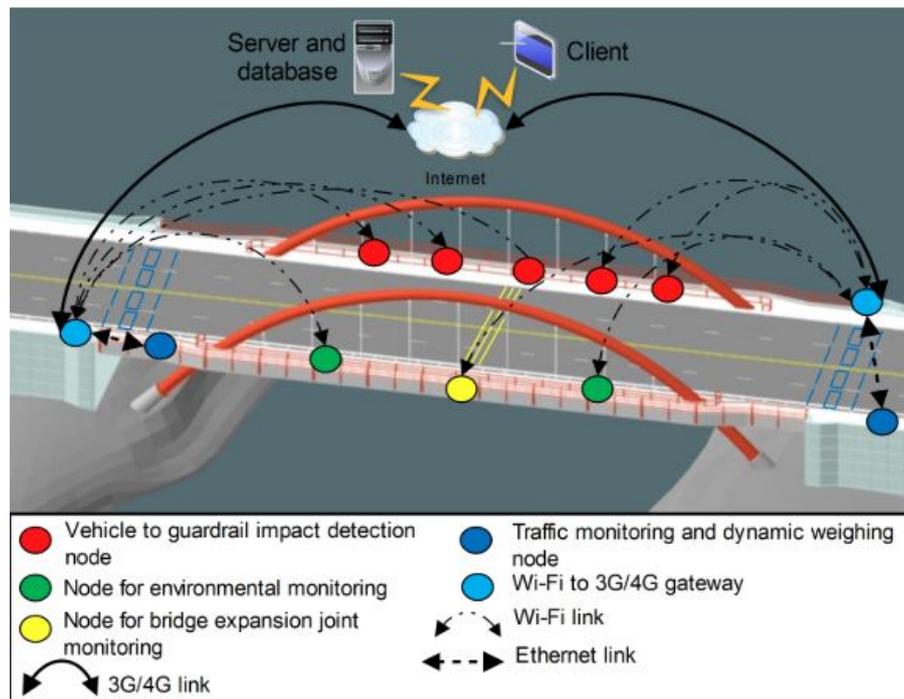


Figura 2.5: Arquitetura geral do sistema.

O nó sensor para a monitorização ambiental visa medir: (1) a temperatura ambiente, (2) a humidade relativa do ambiente, (3) a pressão atmosférica e (4) a velocidade e a direção do vento. Em particular, este nó envia os valores associados às quantidades medidas, periodicamente, para o *gateway* via WI-FI. O período de amostragem pode ser definido pelo consultor do sistema. Este nó é colocado em um suporte mecânico na berma da estrada.

O nó sensor para a monitorização da junta de expansão da ponte tem como objetivo medir: (1) as variações de deslocamento da junta de expansão, (2) as variações de temperatura e humidade relativa dentro da junta e (3) o nível de vibração que atua na junta. As medições de temperatura e humidade relativa são enviadas periodicamente, de acordo com a frequência de amostragem definida pelo consultor do sistema. Por outro lado, um alerta assíncrono é enviado se as variações de deslocamento da junta de expansão forem maiores que um valor limite de segurança. Este nó é colocado na área da junta de expansão da obra de arte.

O nó sensor para a monitorização de tráfego e pesagem dinâmica tem como objetivos: (1) medir a velocidade dos veículos que passam a ponte, (2) contar o número de veículos que passam pela ponte, (3) classificar os veículos que passam pela ponte de acordo com o número de eixos dos veículos, e (4) medir o peso de um veículo que atravessa a ponte. A parte sensível do nó está na superfície da estrada, o sistema de aquisição de dados é colocado em um suporte na berma da estrada. Para cada ponte,

são utilizados dois nós, um colocado na entrada e outro na saída da ponte. Em particular, se o peso do veículo exceder o limite de peso para a ponte específica, um alerta será enviado ao consultor do sistema. Além disso, todas as medições são enviadas para o *gateway* periodicamente.

O *gateway* recebe os dados fornecidos pelos nós e envia-os para o servidor por meio de uma interface de comunicação sem fios (*modem* 3G / 4G). O *gateway* é colocado em um suporte na berma da estrada, de acordo com área de cobertura da rede Wi-Fi.

O servidor é responsável por armazenar os dados adquiridos numa base de dados. Um cliente ou o consultor do sistema faz interface com o servidor via Internet para obter as informações armazenadas.

Todos os nós são alimentados por painéis fotovoltaicos. Na Figura 2.6 é possível observar a proposta desta solução para a monitorização do comportamento estrutural de pontes [13].

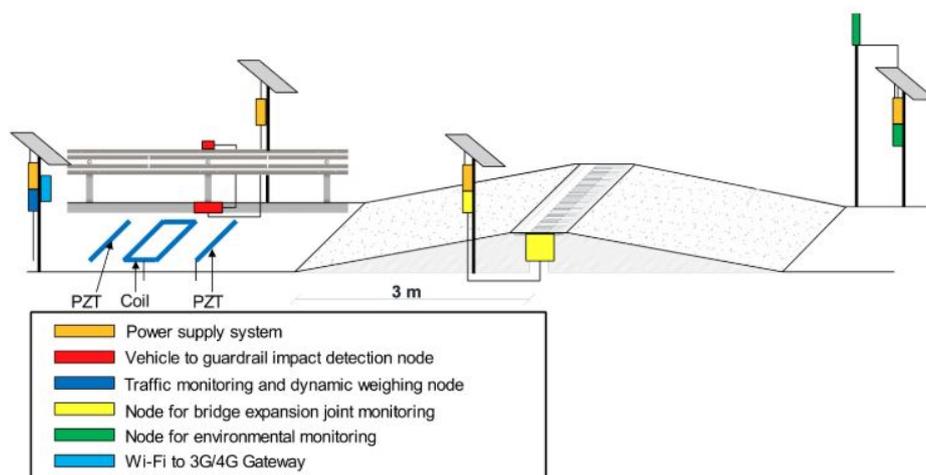


Figura 2.6: Arquitetura proposta para a monitorização do comportamento estrutural de pontes.

A arquitetura do nó sensor para a deteção de colisões em guardas de segurança é representada na Figura 2.7 [13]. Consiste em: (1) a plataforma Wasp mote, (2) o conversor de corrente para tensão, (3) um acelerómetro de três eixos e (4) o módulo WI-FI Pro.

O acelerómetro de três eixos utilizado é o IAC-HiRes-I-03-400g da *Micromega Dynamic*, que fornece três correntes de saída que variam de 4 mA a 20 mA de acordo com a aceleração medida em cada eixo. A faixa de medição de 4 kHz e sua sensibilidade é de 20 μ A/g e pode medir até ± 400 g com uma frequência de corte de -3 dB.

Os valores da corrente são convertidos em tensão por meio de um conversor. O conversor de corrente tem a função de fornecer alimentação ao sensor e converte a corrente numa queda de tensão através de uma resistência. Os três sinais de tensão fornecidos pelo conversor de corrente são adquiridos em

três canais de entrada do ADC incorporado no microcontrolador. De acordo com os valores adquiridos pelo microcontrolador, que se referem às acelerações ao longo dos três eixos, são avaliadas a magnitude da aceleração e sua direção. A magnitude da aceleração é comparada com dois valores limite. Se a aceleração medida for menor que o primeiro valor limite, nenhuma mensagem de alerta será enviada ao *gateway*. Se a aceleração estiver dentro do intervalo definido pelos dois valores limite, uma mensagem de alerta é enviada ao *gateway*, contendo o valor da aceleração, a direção da aceleração e um rótulo que contém o tipo de impacto, nesse caso, baixo impacto. Por outro lado, se o valor medido da aceleração for maior que o segundo valor limite, uma mensagem de alerta rotulada como alto impacto será enviada ao *gateway*.

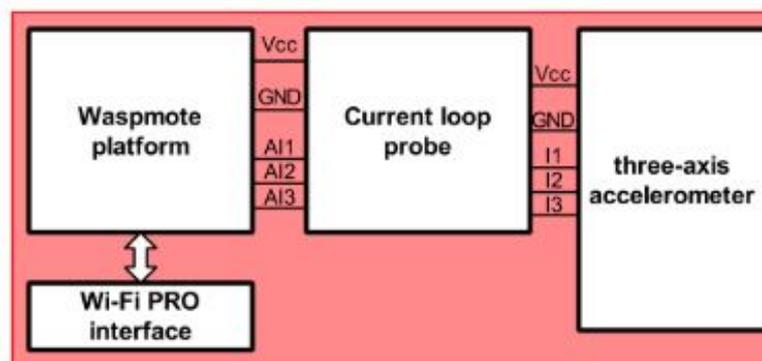


Figura 2.7: Arquitetura do nó sensor para detecção de colisões de veículo com as guardas de segurança.

2.1.3 Sustainable Infrastructure Management System

No âmbito do projeto do sistema de incentivo à investigação e desenvolvimento tecnológico do Quadro de Referência Estratégica Nacional (QREN), foi proposto o sistema de gestão de infraestruturas rodoviárias SustIMS, desenvolvido de 2012 a 2015 pela parceria entre a Universidade do Minho e a Universidade Nova de Lisboa, em co-promoção com a empresa Ascendi. O sistema tem como objetivo criar uma WSN, capaz de detetar colisões de veículos com os guardas de segurança das autoestradas em tempo real. O sistema também pode ser útil para detetar em tempo real os movimentos súbitos em muros de retenção/taludes das autoestradas [14].

O artigo [9] apresenta novas funcionalidades do sistema: monitorização da distância dos veículos com os guardas de segurança; monitorização da iluminação; da humidade; da temperatura e da qualidade do

ar (dióxido de enxofre, dióxido de carbono, dióxido de nitrogénio e partículas) em autoestradas e entradas de túneis. Na Figura 2.8 é possível observar a arquitetura geral do sistema [9].

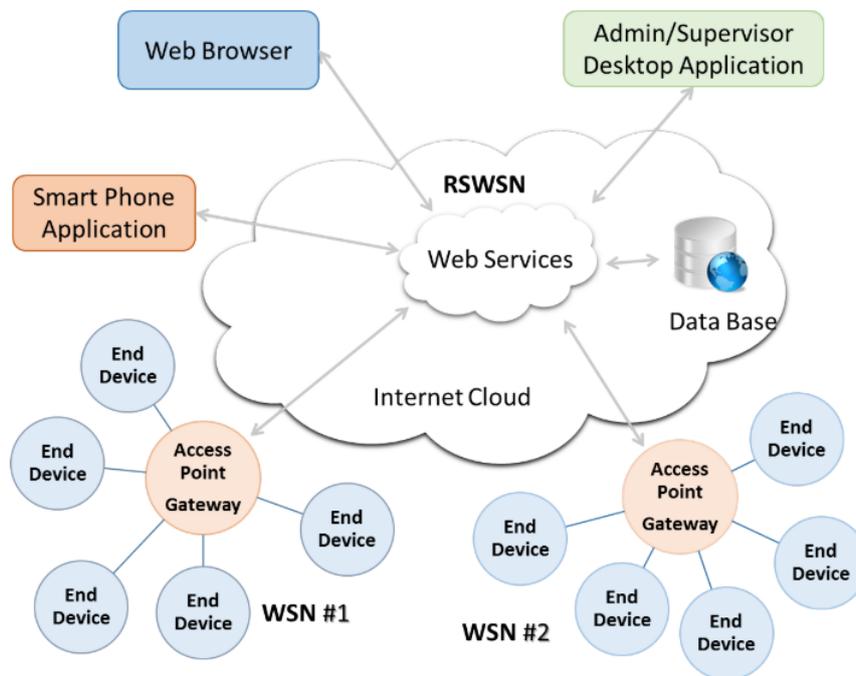


Figura 2.8: Arquitetura geral do sistema.

O sistema geral pode ser dividido em dois subsistemas [9]:

- Serviço de registo de rede de sensores fios (WSNRS - *Wireless Sensor Network Registration Service*), é baseado na arquitetura orientada a objetos (SOA - *Service-Oriented Architecture*) e contém serviços que fazem interface com a WSN e os responsáveis do sistema, através de aplicações móveis, web e *desktop*. Este sistema é responsável por registar e armazenar eventos das WSNs, como a ocorrência de uma colisão.
- Rede de sensores sem fios para deteção de colisão (CDWSN - *Collision Detection Wireless Sensor Network*). é baseado na topologia de rede em estrela, possui dois tipos de nós: *Access Point* (AP) e *End Device* (ED). Os nós ED são fixados às guardas de seguranças, servem para detetar a ocorrência de uma colisão. Os nós AP servem como *gateway* da WSN para reencaminhar dados recolhidos dos nós ED para o subsistema WSNRS.

O *hardware* dos nós é baseado no SoC CC2530EM (baseado no microcontrolador 8051) da TI, este módulo é composto por um *transceiver*RF compatível com o IEEE 802.15.4 e opera na gama de frequências de 2,4 GHz. O nó ED utiliza o acelerómetro LIS331DLH da ST, com resolução de 16 bits e pode

medir acelerações de ± 2 g, ± 6 g e ± 8 g em 3 eixos. A Figura 2.9 mostra o nó ED assembled composto pelo sensor e pelo módulo de comunicação [9].



Figura 2.9: Nó ED instalado na caixa de proteção.

No outro artigo [10] apresenta a atualização do *hardware* dos nós ED e AP, com a combinação entre o SoC CC2530 da TI e o SoC CC2591 TI. O SoC CC2591 é um *range extender* de RF, que permite alcances relativamente longos das comunicações. No mesmo artigo, apresenta também a adição de dois componentes ao sistema [10]:

- **Gateway.** O *gateway* desenvolvido é responsável pela interface da WSN e a *cloud*. Foi implementado usando uma placa Raspberry Pi e pode atuar como *router* ou *firewall*, protegendo a WSN contra acessos não autorizados ou indesejados. As informações recebidas/enviadas para os serviços da *cloud* usam o protocolo HTTP RESTful, permitindo a conectividade e disponibilidade de todo o sistema.
- **Cloud Services.** Os serviços da *cloud* implementados permitem que a WSN envie dados para a internet. Essas informações podem ser usadas para gerir ou monitorizar remotamente a WSN. Quando um evento é acionado, os serviços da *web* são responsáveis e acionar os mecanismos desejados e configurados. Todos os dados recebidos são armazenados numa base de dados. As informações armazenadas podem ser acessadas remotamente por utilizadores autenticados numa aplicação móvel ou via navegador da *web*.

A interface via navegador da *web* de uma WSN instalada, com todas as informações, como endereços dos dispositivos, estado de integridade da rede, localização geográfica, entre outras, que é possível observar na Figura 2.10 [10].

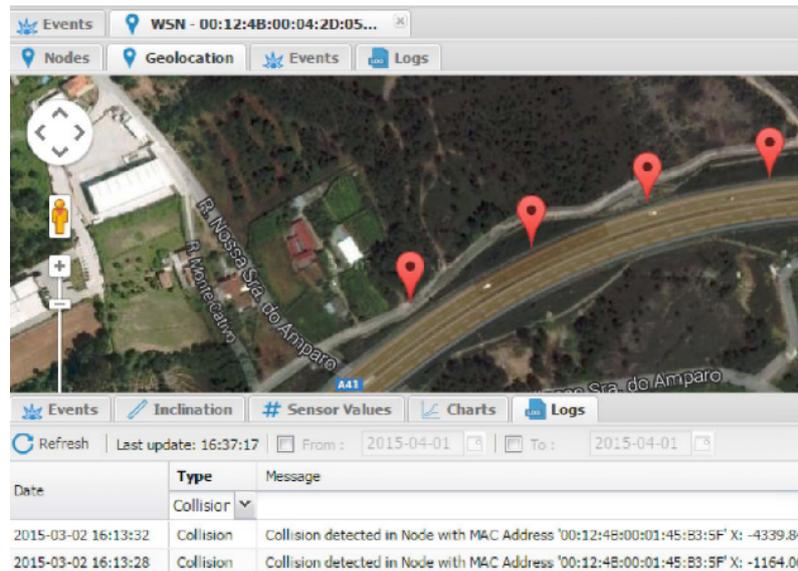


Figura 2.10: Interface via navegador web de uma WSN instalada.

2.1.4 Comparação entre os vários Sistemas de Monitorização

Após de ter sido descrito quatro sistemas de monitorização de colisões com as guardas de segurança. A Tabela 2.1 apresenta uma comparação entre sistemas de monitorização acima mencionados, comparando a plataforma, elemento sensor, protocolo de comunicação, consumo energético e um resumo das vantagens e desvantagens.

Tabela 2.1: Comparação entre os quatros sistemas de monitorização.

Parâmetro	<i>Dynamic Monitoring of Guardrails</i> [12]	<i>IoT System for Remote Monitoring of Bridges</i> [13]	SustIMS versão 2014 [9]	SustIMS versão 2016 [10]
Plataforma	Teensy 3.1	Waspote	CC2530	CC2530 / CC2591
Processador	Arm Cortex-M4	ATmega1281	8051	8051
Protocolo	ZigBee	Wi-Fi	SimpliciTI	ZigBee
Modo Tx	-	350 mA	29 mA	166 mA
Modo Rx	-	130 mA	24 mA	27 mA
Acelerómetro	ADXL375	IAC-HiRes-I-03-400g	LIS331DLH	LIS331DLH
Consumo acelerómetro	145 µA	20 mA	250 µA	250 µA
Vantagens	Maior poder de processamento.	Facilidade de conexão à internet.	<i>Transceiver</i> RF integrado; Baixo consumo energético.	<i>Transceiver</i> RF integrado; Maior alcance.
Desvantagens	<i>Transceiver</i> RF não integrado; <i>Gateway</i> complexo.	<i>Transceiver</i> RF não integrado; Elevado consumo energético.	<i>Gateway</i> complexo; Curto alcance.	<i>Gateway</i> complexo.

2.2 Wireless Sensor Network (WSN)

As WSNs revolucionaram o *design* de sistemas embecidos emergentes [15], com o avanço no campo do sensor como um sistema embecido, comunicação sem fio e processamento distribuído, originando um novo conjunto de aplicações em tempo real [16]. Diferente das redes convencionais, as WSNs são redes caracterizadas pela interação com o ambiente físico, para monitorização e/ou controlo de variáveis ambientais, através de sensores ou atuadores.

2.2.1 Aplicações e Características das WSN

As WSNs são constituídas por um número finito de nós sensoriais, colocados no ambiente físico de forma aleatória ou estrategicamente, dependendo das necessidades da aplicação que se deseja implementar. Estes nós transmitem a informação para uma estação base, que por sua vez, através de um *gateway*, guarda a informação num servidor. Os componentes tipicamente presentes numa WSN são apresentados na Figura 2.11 [16].

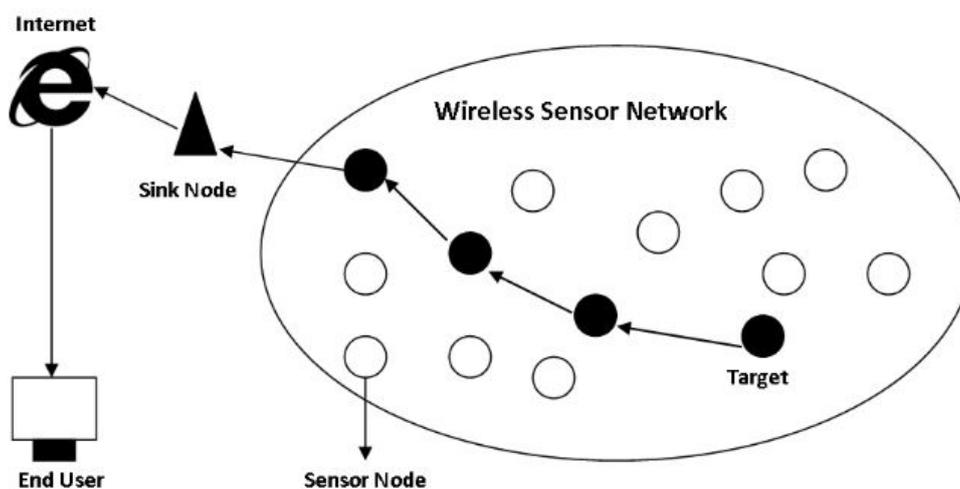


Figura 2.11: Arquitetura *multihop*, típica de uma *Wireless Sensor Network* (WSN).

Inicialmente, as WSNs eram utilizadas em aplicações sofisticadas, como sistemas de deteção de radiação e ameaças nucleares, aplicações biomédicas, monitorização sísmica, entre muitas outras aplicações [17]. Mais recentemente, podem ser aplicadas em inúmeros cenários, devido à flexibilidade na resolução de problemas em diferentes áreas de aplicação, como assistência médica, edifícios inteligentes, monitorização do meio-ambiente, monitorização ou controlo em ambientais industriais, monitorização de

estruturas de engenharia civil, etc [17, 18]. Para além disso, as WSNs têm sido introduzidas em aplicações de *Internet of Things* (IoT) [19].

Para se projetar uma WSN, é necessário conhecer-se com exatidão, seus principais parâmetros, ou seja, a eficiência energética, escalabilidade, confiabilidade e robustez [20]. A WSN pode ser classificada quanto ao tipo de sensores utilizados, como WSN terrestres, WSN subterrâneas, WSN subaquáticas, WSN de multimédias e WSN móveis [16].

Como está descrito em [17] as WSNs estão divididas em duas categorias:

- Categoria 1 WSNs (C1WSNs): são redes baseadas em topologias em malha (*mesh*) com comunicações de múltiplos saltos *multihop* entre os dispositivos de origem e destino, utilizando encaminhamento dinâmico. Normalmente os sistemas do teatro de operações pertencem a essa categoria.
- Categoria 2 WSNs (C2WSNs): são redes baseadas em topologias estrela e ponto a ponto (*peer-to-peer*) com comunicações geralmente de salto único entre os dispositivos de origem e destino, utilizando normalmente o encaminhamento estático. Geralmente os sistemas de automação residencial pertencem a essa categoria.

Em [16], é brevemente apresentada uma análise *Strengths Weaknesses Opportunities Threats* (SWOT) para as WSNs. A análise SWOT é utilizada para diagnosticar as forças/pontos fortes (*Strengths*), fraquezas/pontos fracos (*Weakness*), oportunidades (*Opportunities*) e ameaças (*Threats*) da aplicação em tempo real da WSN, tal como é possível visualizar na Figura 2.12 (adaptado de [16]).

2.2.2 Arquitetura de um Nó Sensorial

Para existir uma WSN, os nós sensoriais devem suportar as seguintes funcionalidades: processamento de sinal (incluindo processamento digital de sinal), compressão, deteção e correção de erros, criptografia, controlo e atuação, *clustering* e computação em rede, auto-montagem, comunicação, *routing* e encaminhamento [17]. A arquitetura de um nó sensorial pode ser dividida em duas áreas de estudo: *hardware* e *software* [17, 21, 22].

Uma arquitetura de *hardware* de um nó sensorial (Figura 2.13 adaptado de [17]) deve ser constituída por quatro principais componentes de modo a operar nas condições e requisitos anteriormente descritos [17, 21, 22]:



Figura 2.12: Análise SWOT da aplicação em tempo real da WSN.

- Sistema de alimentação;
- Sistema de processamento/armazenamento;
- Sistema sensorial;
- Sistema de comunicação

Cada um destes sistemas é fundamental para o bom funcionamento do nó sensorial, sendo que o sistema de alimentação é um dos mais importante, a operação dos restantes sistemas dependem da fonte de energia [17, 23].

Quanto à arquitetura de *software* de um nó sensorial (Figura 2.14) pode ser constituído por cinco sistemas básicos [17, 21, 22]:

- Sistema operativo;
- *Drivers* dos sensores;
- *Drivers* de comunicação;
- Gestão da comunicação da rede;

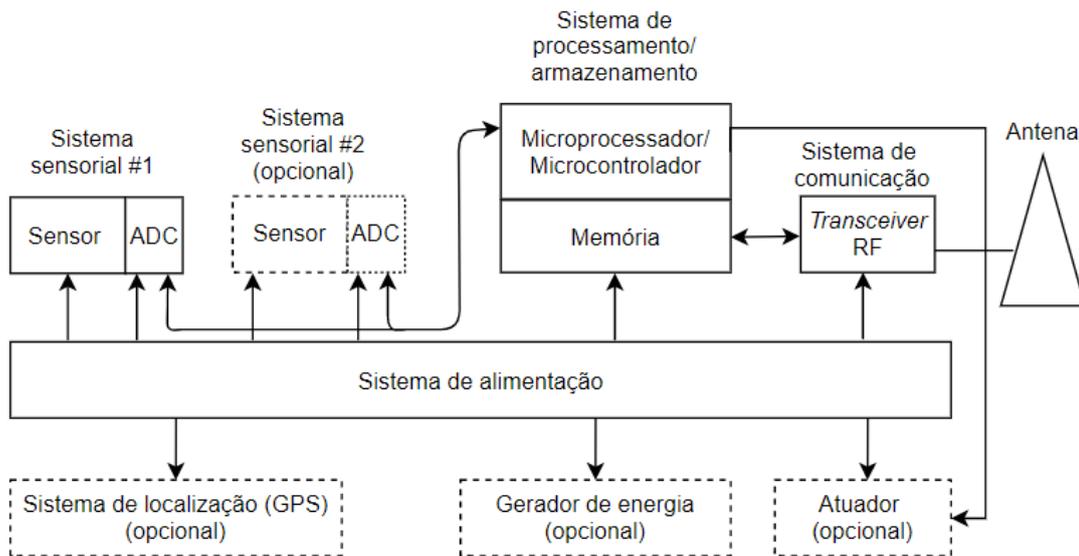


Figura 2.13: Arquitetura típica de *hardware* de um nó sensorial.

- Miniaplicações

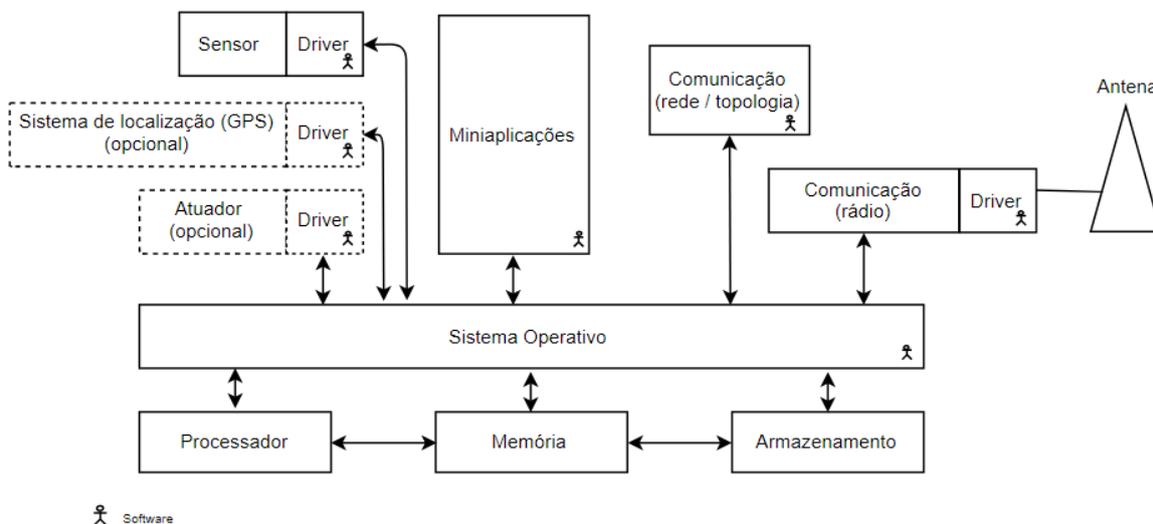


Figura 2.14: Arquitetura típica de *software* de um nó sensorial.

O sistema operativo é um dos blocos mais importantes, sendo que é a camada de abstração entre o *hardware* e as aplicações de *software* que constituem o nó sensorial, são normalmente *open-source* com rápida implementação e tamanho de código reduzido [21, 22].

Atualmente existe um grande interesse na investigação e desenvolvimento de novas soluções ao nível arquitetural do ED. Isto impõe alguns desafios adicionais relacionados à segurança e privacidade, desempenho e eficiência energética [24, 25]. O artigo [25], apresenta uma arquitetura heterogénea para os *End Devices*, que resulta da combinação de um microcontrolador, *hardware* reconfigurável e um *transceiver* RF compatível com o IEEE 802.15.4. A Figura 2.15 apresenta as possíveis opções de *design* de arquiteturas heterogéneas [25].

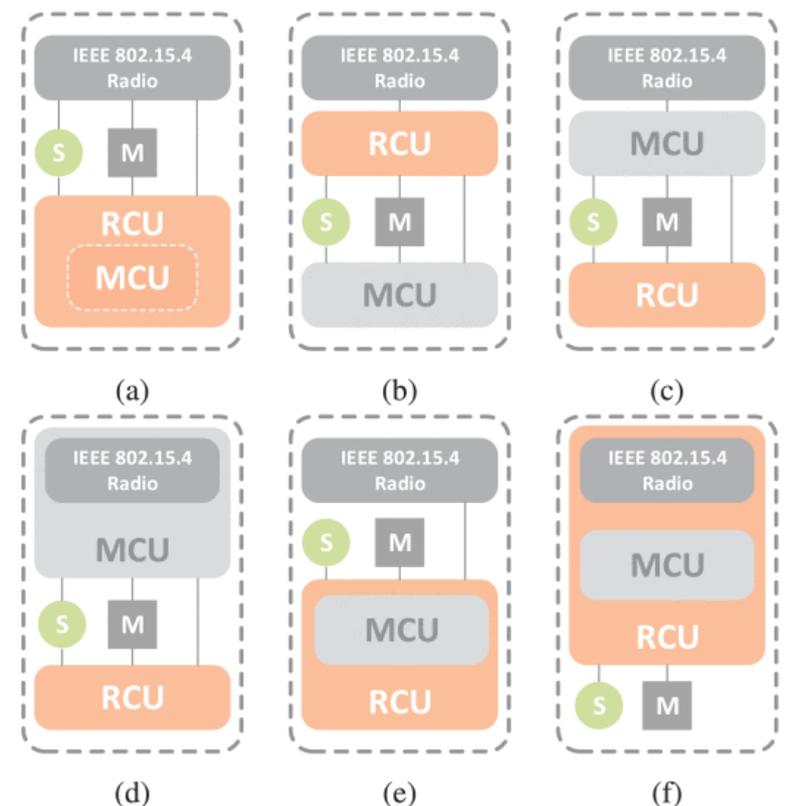


Figura 2.15: Opções de *design* para arquiteturas heterogéneas.

2.3 Standard IEEE 802.15.4

Desenvolvido pelo grupo de trabalho *Institute of Electrical and Electronics Engineers* (IEEE) 802.15, o IEEE 802.15.4 especifica a camada física (PHY) e a camada de acesso ao meio (MAC) para redes de área pessoal sem fios de baixo débito (LR-WPAN) [26, 27, 28]. Este *standard* consiste em desenvolver redes sem fios de baixo custo de implementação, dedicadas, com protocolos simples e baixo consumo de energia [29]. Serve de base de outras redes e protocolos, como é o caso do ZigBee, 6LoWPAN, IEEE 802.15.5, WirelessHART, WIA-PA e ISA100.11a [22, 30]. Os sistemas WSNs mais usados incorporam o

IEEE 802.15.4 nas suas redes e protocolos, abertos ou proprietários [20, 26]. A Figura 2.16 ilustra uma comparação entre tecnologias inseridas nos grupos de redes sem fios de área pessoal (WPAN) e redes de área local (WLAN), comparando a taxa de transferência, a complexidade de implementação e o consumo de energia [31].

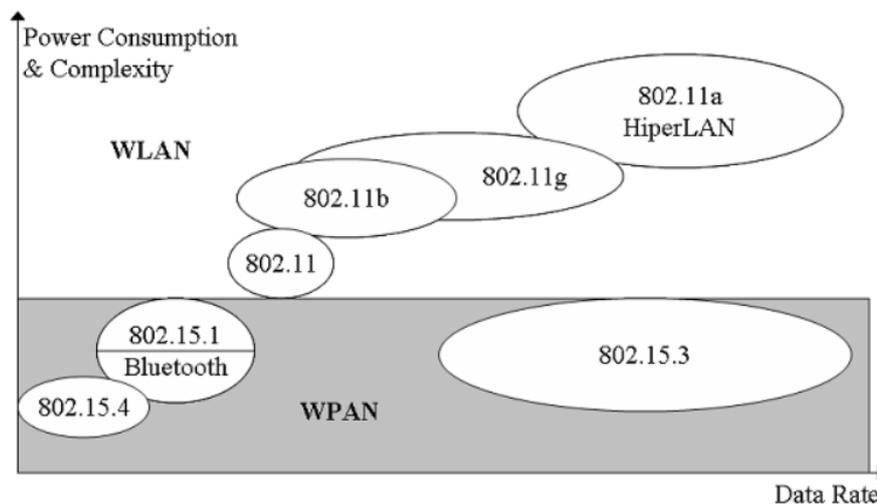


Figura 2.16: Comparação de *standards* de redes WPAN e WLAN relativamente a taxa de transferência, complexidade de implementação e consumo de energia.

A comparação entre uma LR-WPAN e o protocolo *Bluetooth* e *Wi-Fi* é apresentada na Tabela 2.2 (adaptado de [8]).

Tabela 2.2: Comparação entre uma LR-WPAN e o protocolo Bluetooth e Wi-Fi.

Tecnologias sem fios			
	LR-WPAN	Bluetooth	Wi-Fi
Débito	250 kbps	1 Mbps	300 Mbps
Consumo	Tx > 30 mA standby. < 10 µA	Tx > 400 mA standby. < 0,2 mA	Tx > 400 mA standby. < 20 mA
Pilha protocolar	≈ 32 kb	≈ 250 kb	≈ 1 Mb
Vantagens	Consumo, latência, número de nós e custo	Interoperabilidade e ausência de cabos	Elevada taxa de transferência
Principais aplicações	Monitorização e controlo sensores	Periféricos de PCs, telemóveis e PDAs	Internet, transferência de arquivos, vídeo/áudio
Alcance de transmissão (metros)	1-100+	1-10+	1-100+

2.3.1 Componentes e Topologias de Rede

2.3.1.1 Componentes de Rede

O IEEE 802.15.4 basicamente define três tipos de nós [15]:

- **Coordenador PAN:** é o controlador principal mestre da rede, que identifica seu *Personal Area Network* (PAN) e ao qual outros nós podem estar associados. Ele também fornece serviços de sincronização global para outros nós na rede através da transmissão da trama *beacon* contendo a identificação da PAN e outras informações relevantes.
- **Coordenador:** possui as mesmas funcionalidades do coordenador PAN, com exceção de que ele não cria o seu PAN. Um coordenador é associado a um coordenador da PAN e fornece serviços de sincronização local aos nós em seu alcance por meio de transmissão da trama de *beacon* que contém a identificação da PAN definida pelo coordenador da PAN ao qual está associado e outras informações relevantes.
- **Nó simples (escravo):** é um nó que não possui nenhuma funcionalidade de coordenação. É associado como escravo do coordenador da PAN (ou de um coordenador) por ser sincronizado com outros nós da rede.

2.3.1.2 Topologias de Rede

Relativamente às topologias de redes, foram definidas no IEEE 802.15.4 duas topologias de rede básicas que são topologia em estrela ou *peer-to-peer* [23, 15]. Um terceiro tipo de topologia em árvore, pode ser considerado como um caso específico da topologia *peer-to-peer* [15]. A Figura 2.17 [15] ilustra as topologias de rede.

Na topologia em estrela, um nó exclusivo opera como um coordenador da PAN. Esta topologia é baseada num sistema centralizado, isto é, cada nó que entra na rede e deseja se comunicar com os outros nós deve enviar seus dados ao coordenador da PAN, que enviará aos nós de destino. Devido às tarefas de consumo de energia de energia do coordenador da PAN na topologia em estrela, o IEEE 802.15.4 recomenda que o coordenador da PAN seja alimentado por energia, enquanto outros nós têm maior probabilidade de serem alimentados por bateria. A topologia em estrela pode ser promissora no caso de arquitetura baseada em *cluster*, a principal desvantagem é de não ser adequada para redes sensores sem fio tradicionais [15].

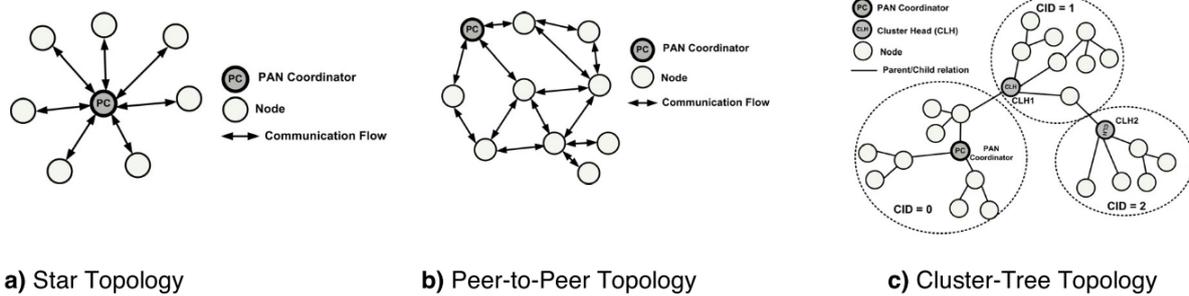


Figura 2.17: Topologias de rede.

Na topologia *peer-to-peer* também inclui um coordenador da PAN que identifica toda a rede. No entanto, o paradigma de comunicação nessa topologia é descentralizado, onde cada nó pode se comunicar diretamente com qualquer outro nó dentro de seu alcance de rádio [15]. No IEEE 802.15.4 não são consideradas funções de *routing* que permite criar redes com maior flexibilidade, do tipo da topologia em malha (*mesh*), mas induz complexidade adicional ao fornecer conectividade de ponta a ponta entre todos os nós da rede [15, 23].

Ao contrário da topologia em estrela, a topologia *peer-to-peer* pode ser mais eficiente em termos de energia e o uso de recursos da bateria é melhor dividido, devido a descentralização da rede, ou seja, o processo de comunicação não depende de um nó específico (o coordenador da PAN) [15].

Na topologia em árvore um coordenador é nomeado como coordenador da PAN, que identifica toda a rede. No entanto, qualquer nó pode atuar como um coordenador e fornecer serviços de sincronização para outros nós ou outros coordenadores. A nomeação de novos coordenadores é responsável pelo coordenador da PAN. Na verdade, o *standard* não define como construir uma rede em árvore de *cluster*. Apenas indica que isso é possível e pode ser iniciado por camadas superiores [15].

A formação do *cluster* é realizada da seguinte maneira, o coordenador da PAN forma o primeiro *cluster*, estabelecendo-se como *Cluster Head* (CLH) com um *Cluster Identifier* (CID) igual a zero. Em seguida, escolhe um *PAN Identifier* (PAN ID) não utilizado e transmite *beacons* para os nós vizinhos. Os nós que estão no intervalo deste CLH podem solicitar a associação à rede no CLH. Em caso de aceitação, o CLH adiciona o nó solicitante como um nó filho em sua lista de vizinhos, e o nó recém-adicionado adiciona o CLH como seu pai na lista de vizinhos e começa a transmitir sinais periódicos. Outros nós podem entrar na rede no último nó adicionado. Se, por algum motivo, o nó solicitante não puder entrar na rede no cabeçalho do *cluster*, ele procurará outro nó pai [15].

Para uma rede de sensores em larga escala, é possível formar uma malha a partir de vários *clusters*

vizinhos. Em tal situação o coordenador da PAN pode promover um nó para se tornar o CLH de um novo *cluster* adjacente ao primeiro. Outros nós gradualmente se conectam e formam uma estrutura de rede com vários *clusters* [15], tal como é ilustrado na Figura 2.17 [15].

2.3.2 Camada Física PHY

A camada física é a base do modelo OSI [32], é a camada mais próxima do *hardware* responsável pela interface entre o canal físico de rádio e a subcamada MAC [22], permite a transmissão e recepção de dados, com a técnica de modulação específica, bem como a potência do sinal emitido [23]. Esta camada também é responsável pelas seguintes operações [15]:

- **Ativação e desativação do *transceiver* RF.** O *transceiver* RF pode operar em três estados distintos: transmissão, recepção ou modo de poupança de energia (*sleep*). Mediante solicitação da subcamada MAC, o rádio é ligado ou desligado. A norma recomenda que o tempo de resposta da transmissão para o estado de recepção e vice-versa (*turnaround time*) não deve exceder a 12 períodos do símbolo (*symbol periods*).
- **Deteção de energia (*Energy Detection*).** É uma estimativa da intensidade de sinal recebida dentro da banda larga de um canal IEEE 802.15.4. Esta tarefa não envolve nenhuma identificação de sinal ou descodificação no canal. A especificação recomenda que a duração da deteção de energia deve ser igual a 8 períodos do símbolo. Essa medida geralmente é usada na camada de rede, como parte do algoritmo de seleção de canal, ou pela função CCA, na camada MAC.
- ***Link Quality Indicator (LQI)*.** A medição do LQI caracteriza a qualidade do sinal de um pacote recebido na ligação. O LQI pode ser implementado usando a técnica de deteção de energia medido pelo recetor, uma estimativa da relação sinal/ruído, ou mesmo uma combinação entre estes valores. O resultado do LQI pode ser usado pelas camadas superiores (camada de rede e aplicação), mas o *standard* não especifica esse procedimento.
- ***Clear Channel Assessment (CCA)*.** Esta operação é responsável por reportar o estado da atividade do meio: ocupado ou livre. O CCA pode ser realizado em três modos de operação:

Energy Detection mode : o CCA reporta que o meio está ocupado perante uma deteção de energia superior a energia detetada *threshold*.

Carrier Sense mode: o CCA reporta que o meio está ocupado apenas se detetar um sinal com as características de modulação e espalhamento espectral do IEEE 802.15.4, independentemente de esse sinal ser superior ou inferior a energia detetada *threshold*.

Carrier Sense with Energy Detection: combinação das técnicas descritas anteriormente. O CCA indica o meio como ocupado se, e somente se, detetar um sinal com as características de modulação e espalhamento espectral do IEEE 802.15.4 e com uma energia superior a energia detetada *threshold*.

- **Seleção do canal.** O *standard* IEEE 802.15.4 define 27 canais sem fios diferentes. Uma rede pode operar dentro de um conjunto de canais. A camada física deve ser capaz de sintonizar o transceptor no canal requerido por uma camada superior.

O IEEE 802.15.4 oferece três bandas de frequência de operação distintas: 2.4 GHz, 868 MHz e 915 MHz. Existe somente um canal entre 868 MHz e 868,6 MHz (canal 0), 10 canais (canais 1 a 10, não disponíveis na Europa) e, por fim, 16 canais (canais 11 a 26) [15], como se pode observar na Figura 2.18 [15].

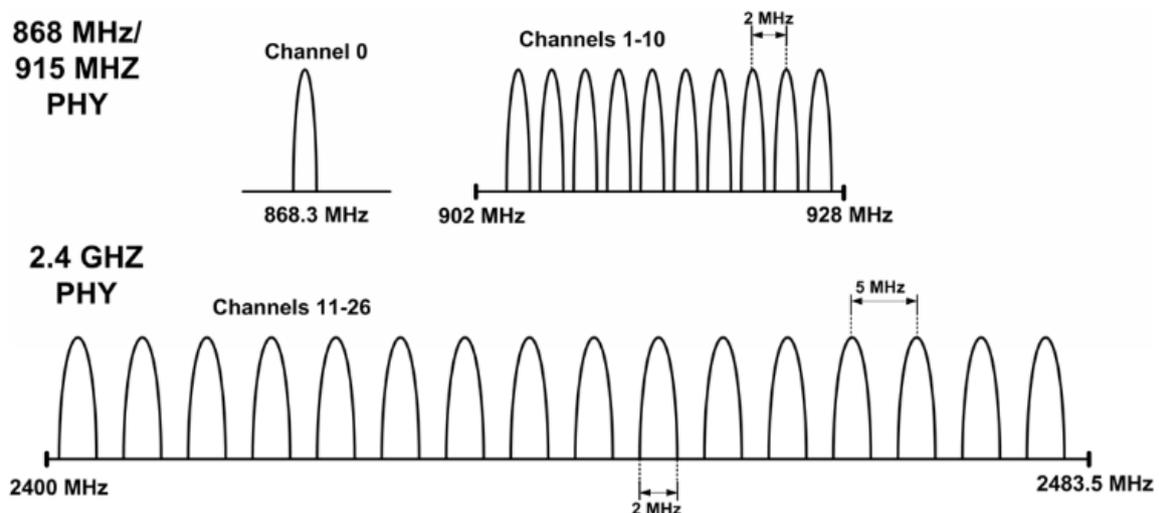


Figura 2.18: Canais de frequência do IEEE 802.15.4.

Quanto à taxa de transmissão, o IEEE 802.15.4 disponibiliza 250 kbps a 2,4 GHz, 40 kbps a 915 MHz e 20 kbps a 868 MHz. As gamas de frequências, são as mais adequadas para transmissões a longa distância, devido a menores perdas de propagação. No entanto, a vantagem da taxa de transmissão

mais alta possibilita maior débito, menor latência e menor *duty cycle* [15]. As características associadas a cada banda de frequências é apresentado na Figura 2.19 [15].

Frequency Band (MHz)	Spreading Parameters		Data Parameters		
	Chip rate (kchip/s)	Modulation	Bit rate (kbps)	Symbol rate (ksymbol/s)	Symbols
868	300	BPSK	20	20	Binary
915	600	BPSK	40	40	Binary
2400	2000	O-QPSK	250	250	16-ary

Figura 2.19: Banda de frequências e débitos do IEEE 802.15.4.

No que diz respeito a coexistência entre o IEEE 802.15.4 e o IEEE 802.11b, pois ambos operam na banda de frequência de 2,4 GHz, geralmente é possível com um desempenho aceitável, quando os dispositivos não estão muito próximos um do outro e os canais são selecionados adequadamente para evitar sobreposição [15].

2.3.3 Camada MAC

A camada de acesso ao meio (MAC) é responsável pela gestão do canal físico de rádio, encontra-se entre a camada física e a camada de rede. Esta camada é também responsável pelas seguintes operações [33, 32]:

- Gerar os *beacons*, caso o dispositivo seja um coordenador.
- Sincronização com os *beacons* da rede.
- Suporte à associação e desassociação de dispositivos na PAN.
- Suporte de segurança no dispositivo.
- Gestão do mecanismo (GTS).
- Implementação do mecanismo *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) para acesso ao canal.

A camada MAC do IEEE 802.15.4 suporta dois modos de operação, como mostra a Figura 2.20 [15]:

- **Modo *beacon-enabled*.** Neste modo, *beacons* são gerados e enviados periodicamente pelo coordenador da PAN para sincronizar os nós associados a ele e fornecer informação relacionada à PAN. O tempo é estruturado em supertramas, e pode haver períodos inativos para poupança de energia. O controlo de acesso ao meio é efetuado com base no mecanismo *slotted* CSMA-CA. No entanto, o modo ativado por *beacon* também permite alocação de alguns intervalos de tempo na supertrama, chamados de GTS.
- **Modo *non-beacon-enabled*.** Neste modo não existe uma estrutura em supertrama. O controlo de acesso ao meio é efetuado com base no mecanismo *unslotted* CSMA-CA.

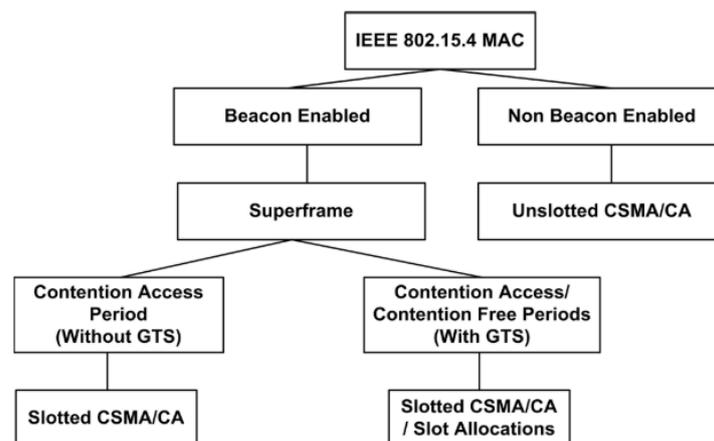


Figura 2.20: Modos de operação da camada MAC do IEEE 802.15.4.

A Figura 2.21 apresenta os procedimentos no caso da comunicação de um dispositivo para um coordenador [34].

Estrutura da supertrama

No modo *beacon-enabled* do IEEE 802.15.4, o tempo é dividido em supertramas com a estrutura apresentada na Figura 2.22 [15]. O coordenador da PAN envia periodicamente *beacon* para identificar a PAN e sincronizar os dispositivos com o coordenador. O intervalo entre os *beacons* (BI), que inclui a parte ativa e, opcionalmente, parte inativa. A parte ativa, é a supertrama que é composta por 16 *slots* de igual duração, que permite a transmissão de dados. Durante a parte inativa (se existir), todos os nós podem entrar em modo de poupança de energia [15].

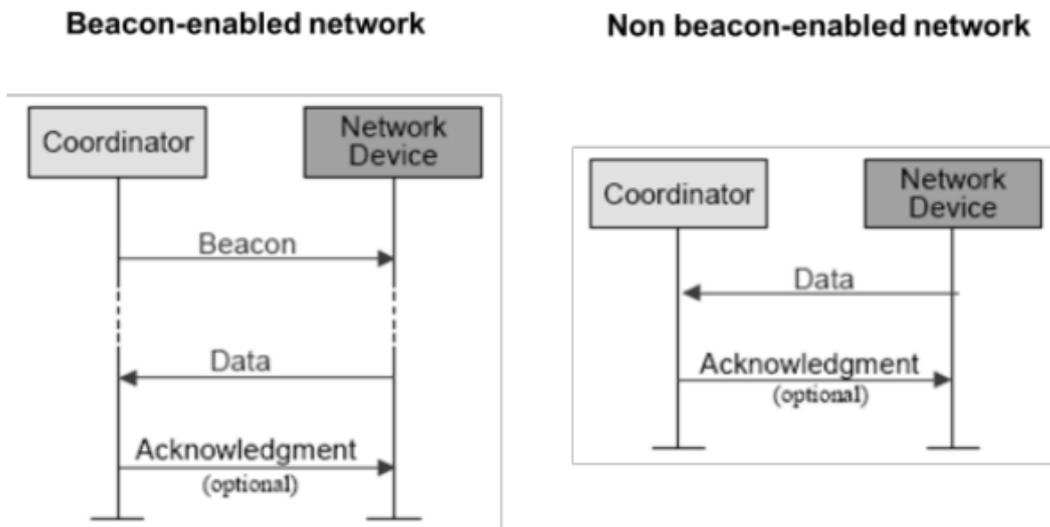


Figura 2.21: Comunicação de um dispositivo para o coordenador.

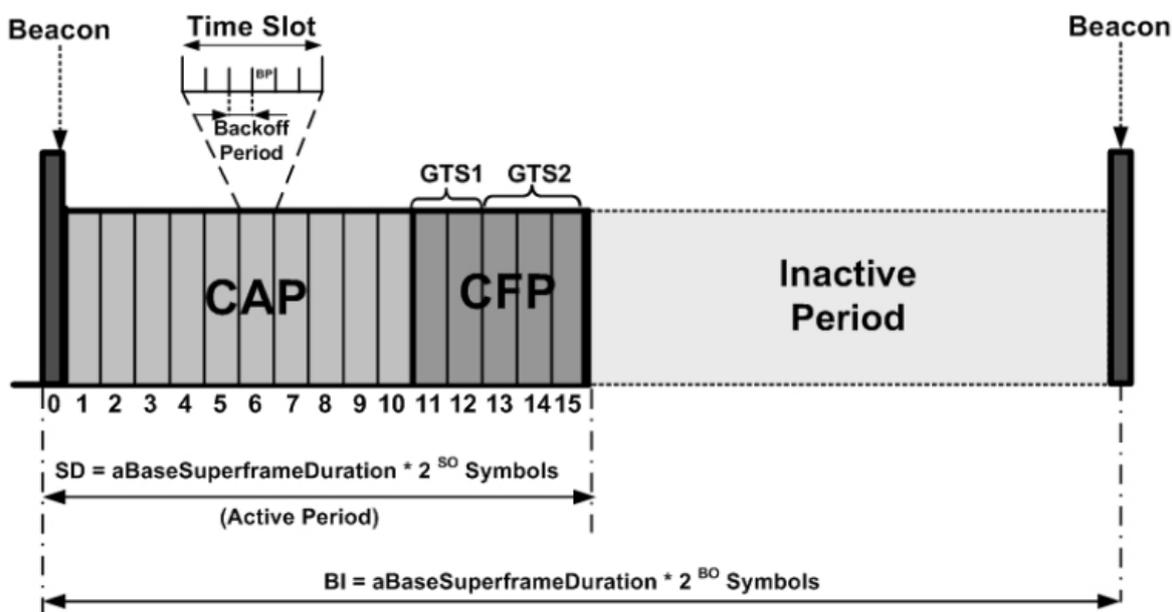


Figura 2.22: Estrutura da supertrama de modo *beacon-enabled*.

O intervalo entre os *beacons* e a duração da parte ativa da supertrama (*Superframe Duration*) são determinados por dois parâmetros, *Beacon Order* (BO) e *Superframe Order* (SO), respetivamente. O *Beacon Interval* (BI) é dado pela fórmula:

$$BI = aBaseSuperframeDuration \cdot 2^{BO}, 0 \leq BO \leq 14 \tag{2.1}$$

A duração da supertrama, que corresponde a parte ativa, é definida da seguinte forma:

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, 0 \leq SO \leq BO \leq 14 \quad (2.2)$$

No modo *non-beacon-enabled* não existe uma estrutura em supertrama, o acesso ao meio é feita de forma aleatória, através do mecanismo *unslotted* CSMA-CA. Neste modo é possível utilizar maiores períodos de poupança de energia [15, 32].

Tipos e formatos de tramas

O IEEE 802.15.4 quatro tipos de tramas [15]:

- **Trama *beacon***, utilizada por um coordenador para transmissão de *beacons*.
- **Trama de dados**, utilizada por todos tipos de dispositivos para as transferências de dados.
- **Trama de *acknowledgment***, utilizada para confirmar a receção bem-sucedida de uma trama.
- **Trama de comando MAC**, usada para sinalização de nível MAC.

O formato da trama de dados, representada na Figura 2.23 [32, 33]. O cabeçalho da trama MHR, os dois primeiros 2 *bytes* indicam o tipo de trama e o modo de endereçamento. Os seguintes *bytes* estão reservados para o número de sequência, para os endereços de destino e origem, seguido de um campo auxiliar de segurança [32, 33]. A cauda da trama MFR contém o campo *Frame Check Sequence* (FCS), que transporta um código CRC de 16 bits usado para deteção de erros na trama [15].

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
Addressing fields								
MHR							MAC Payload	MFR

Figura 2.23: Formato da trama de dados.

2.4 ZigBee

ZigBee é uma tecnologia desenvolvida pela *ZigBee Alliance* [35], que define um conjunto de protocolos para comunicações sem fio de baixo custo, baixo consumo de energia e baixo débito [36]. O IEEE 802.15.4 está muito associada ao ZigBee, que é um protocolo de rede que especifica as camadas acima do IEEE 802.15.4 para fornecer a estrutura protocolar completa para as redes *Low-Rate Wireless Personal Area Networks* (LR-WPAN) [15, 22]. Essa pilha protocolar resulta da colaboração entre a *ZigBee Alliance*, uma organização com mais de 200 empresas membros (*Ember, Freescale, Invensys, Mitsubishi, CompXs, AMI Semiconductors, ENQ Semiconductors*), de semicondutores e *software* que trabalham em conjunto com o IEEE 802.15.4 para obterem uma especificação completa do protocolo e também promoverem seu uso em todo mundo [15, 36]. As primeiras especificações do ZigBee foram aprovadas em 2004 e foram atualizadas sucessivamente até ao ano de 2015, cada uma das especificações tem a sua designação base da norma ZigBee, respetivamente, ZigBee 2004, ZigBee 2006 e ZigBee 2007 [22, 35, 37]. Atualmente as especificações ZigBee 2004 encontram-se obsoleta, as especificações 2006 não tiveram nenhuma alteração e têm compatibilidade com as especificações ZigBee 2007, que têm uma versão melhorada designada por ZigBee PRO [22, 35, 37]. Recentemente a partir do ZigBee PRO surgiu o ZigBee 3.0, que otimiza o IEEE 802.15.4, adicionando camada de rede em topologia malha e segurança, juntamente com uma estrutura de aplicação para se tornar numa solução ZigBee *full stack*, de baixo consumo e com certificação [35].

2.4.1 Aplicações e Características da Rede ZigBee

O dispositivo ZigBee é a combinação de aplicação (como por exemplo o sensor de luz, controle de iluminação, etc), ZigBee lógico (*ZigBee Coordinator, ZigBee Router e ZigBee End Device*) e tipos físicos de dispositivos (*Full-Function Device e Reduced-Function Device*) [38].

O IEEE 802.15.4 define os dois primeiros nós de *Full-Function Device* (FFD), o que significa que eles implementam todas as funcionalidades do IEEE 802.15.4 para garantir a sincronização e a gestão da rede. O terceiro tipo é conhecido como *Reduced-Function Device* (RFD), o que significa que o nó está operando com uma implementação mínima do protocolo IEEE 802.15.4, enquanto os nós simples podem ser FFDs ou RFDs [15].

Um FFD típico numa rede ZigBee será alimentado por energia elétrica fornecida pela rede, pois sempre deve estar ativo e ouvindo a rede. Os RFDs, por outro lado, implementam uma versão limitada do IEEE

802.15.4, estes dispositivos não encaminham pacotes e devem estar associadas a um FFD. Os RFDs são dispositivos finais, como atuadores e sensores, que executam apenas tarefas limitadas, como registrar dados de temperatura, monitorizar condições de iluminação ou controlar dispositivos externos. A especificação atual do ZigBee exige que os FFDs estejam sempre no modo ativo, o que na prática, significa que os FFDs devem ser constantemente alimentados. Os FFDs movidos a bateria têm uma vida útil na ordem de alguns dias [38].

Numa rede ZigBee pode haver três tipos nós: *ZigBee Coordinator* (ZC), *ZigBee Router* (ZR) ou *ZigBee End Device* (ZED), tal como é possível visualizar na Figura 2.24 [38]:

- **ZC**, forma a raiz da topologia de rede baseada em árvore e pode fazer a ponte para outras redes. Numa rede ZigBee só existe um único coordenador. Este dispositivo é responsável por iniciar a rede e selecionar os parâmetros da rede, como canal de radiofrequência, identificador exclusivo da rede (PAN ID) e definir outros parâmetros operacionais. Também pode armazenar as informações sobre rede, chaves de segurança, etc.
- **ZR**, atua como nós intermediários, retransmitindo dados de outros dispositivos. O *router* pode se conectar numa rede já existente, também capaz de aceitar conexões de outros dispositivos e ser algum tipo de retransmissor para a rede. A rede pode ser alargada através do uso de ZR.
- **ZEDs**, normalmente operam com alimentação de baterias. Estes dispositivos podem recolher várias informações de sensores. Não podem retransmitir dados de outros dispositivos, essa funcionalidade reduzida permite reduzir os seus custos. Esses dispositivos não precisam ficar acordados o tempo todo, enquanto os dispositivos pertencentes às outras duas categorias precisam. Cada dispositivo final pode ter 240 aplicações.

As redes ZigBee são redes baixo débito, permite atingir taxas de transmissão até 250 kbps, são vocacionadas para aplicações nas quais o baixo consumo de energia, baixo custo e pequeno tamanho dos nós são normalmente prioritários. O que torna interessante de usar em certas aplicações, tais como redes de sensores corporal (*Body Sensor Network*), controlo e automação residencial, leitura automática de medidores, sistemas de utilidade residencial e comercial, automação de edifícios, áreas dos cuidados de saúde, sistemas de segurança e vigilância, *ZigBee Smart Energy*, desporto e entretenimento [38], como se pode observar na Figura 2.25 [39].

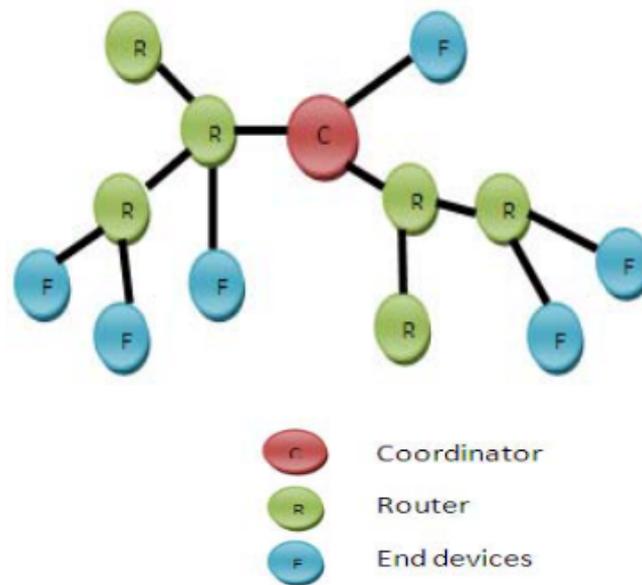


Figura 2.24: Rede ZigBee.



Figura 2.25: Aplicações da rede ZigBee.

Como ainda não existem padrões globais em redes de sensores sem fios, o ZigBee desempenha um papel vital na maioria das aplicações sem fios, na maioria das indústrias, observa-se uma demanda crescente de aplicações sem fio baseados em ZigBee [38].

2.4.2 Arquitetura Protocolar

Uma das características de uma rede de comunicação (com ou sem fio) está no uso do conceito de camadas de rede, onde cada camada é responsável por determinadas funções na rede. As camadas normalmente transmitem dados e comandos somente para as camadas diretamente acima e abaixo delas [36].

O ZigBee baseia-se no IEEE 802.15.4 (*hardware*), que define as camadas física e de controlo de acesso ao meio (MAC). A *ZigBee Alliance* (*software*) define as camadas de rede, segurança e aplicação [36]. A arquitetura do protocolo de comunicação IEEE 802.15.4 definida na norma ZigBee é apresentada na Figura 2.26 [36].

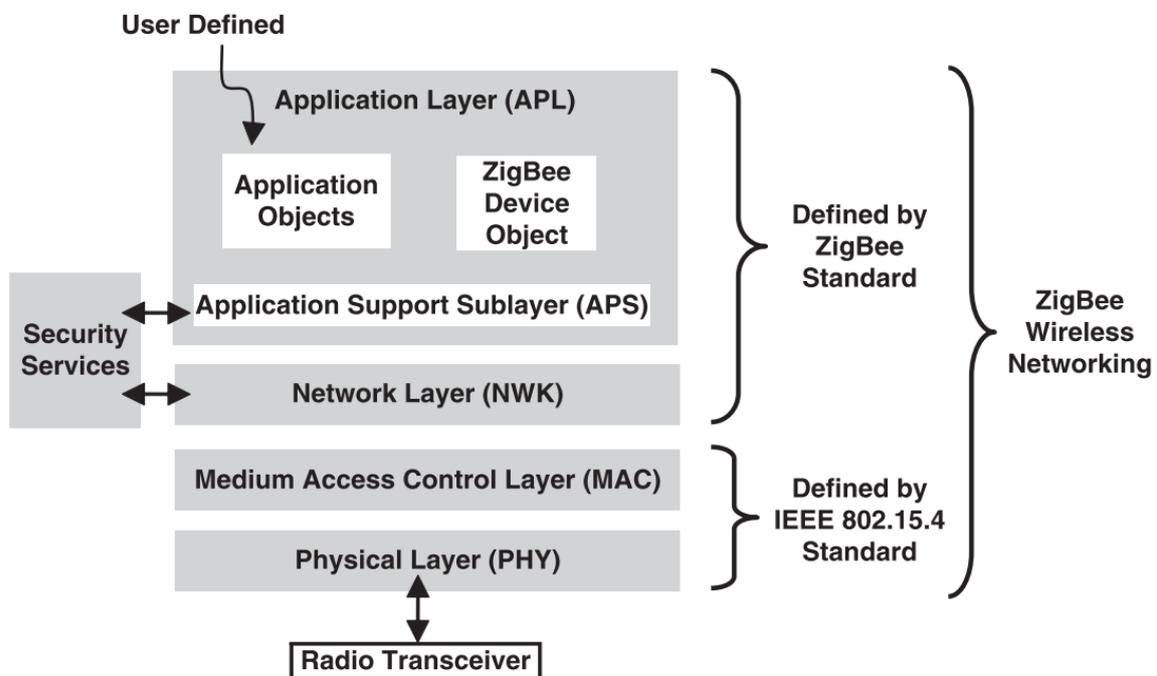


Figura 2.26: Arquitetura do protocolo de comunicação IEEE 802.15.4/ZigBee.

Camada de Rede (NWK)

A camada de rede é a interface entre a camada de aplicação e camada MAC [22, 35]. Essa camada é responsável pela formação, endereçamento de rede, encaminhamento, *broadcasting*. O *routing* é o processo de seleção de caminho para retransmitir as mensagens para o nó de destino. Essa camada fornece segurança em toda a rede e permite que dispositivos de baixa energia maximizem a vida útil da bateria [36, 38]. A camada de rede de um coordenador ZigBee é responsável por estabelecer uma nova

rede e selecionar a topologia da rede (árvore, estrela ou malha), armazenar informações relativas aos dispositivos, monitorização dos dispositivos da rede, também é responsável por atribuir os endereços de rede aos dispositivos em sua rede [36, 40].

Para um correto funcionamento da camada MAC e da camada de aplicação, esta camada fornece duas entidades responsáveis pelo serviço de dados e de serviço de gestão através do seu ponto de acesso ao serviço (SAP) [40]. A *Network Layer Management Entity* (NLME) é a entidade de gestão responsável por permitir a uma aplicação interagir com a pilha do protocolo ZigBee através do transporte de comandos de gestão entre a camada de aplicação (APL) e a entidade de gestão da camada NWK. As principais responsabilidades do NLME são [22, 35, 40]:

- Configurar a pilha do protocolo ZigBee de um novo dispositivo, em que dentro das opções de configuração se encontram início da operação como ZC ou junção a uma rede já existente;
- Estabelecer uma nova rede;
- Permite aos dispositivos efetuarem a associação (*joining*) ou reassociação (*rejoining*) à rede de maneira segura, permite também que um ZC ou um ZR solicite a um dispositivo o abandono da rede;
- Atribuir endereços de rede, através do ZC e do ZR, a dispositivos que se juntem à rede;
- Descobrir dispositivos na vizinhança e registar as respetivas informações;
- Descobrir rotas e registar os caminhos existentes na rede para tornar o encaminhamento de mensagens eficiente;
- Controlar o recetor do dispositivo no instante e período de atividade, para possibilitar a sincronização da camada MAC ou receção direta;
- Fornece diferentes mecanismos de roteamento como, por exemplo, *unicast*, *multicast*, *broadcast* ou *many to one* para troca de dados na rede.

A *Network Layer Data Entity* (NLDE) é a entidade de dados que fornece o serviço de transmissão de dados responsável por permitir a uma aplicação o transporte de unidades de dados do protocolo *Application Support Sub-Layer Protocol Data Unit* (APDU), e está capacitada a fornecer serviços como [22, 35, 40]:

- Criação de *Network Protocol Data Unit* (NPDU), a partir do PDU da subcamada de suporte à aplicação (*Application Support*) e da adição do respetivo cabeçalho;
- Topologia específica de roteamento;
- Garantir a segurança da transmissão, quer em autenticidade, quer confidencialidade;

Camada de Aplicação (APL)

A camada de aplicação é a camada de protocolo mais alta da rede sem fio ZigBee e hospeda os objetos de aplicação. Os fabricantes desenvolvem os objetos de aplicação para personalizar um dispositivo para várias aplicações. Os objetos de aplicação controlam e gerem as camadas de protocolo num dispositivo ZigBee. Pode existir até 240 objetos de aplicação num único dispositivo [36]. A especificação do ZigBee separa a camada APL em três subcamadas diferentes: a subcamada de suporte à aplicação (APS), os objetos de dispositivo ZigBee (ZDO) e o *framework* da aplicação (AF) definida como objetos da aplicação (APO) [38].

- **Objetos de aplicação (APO).** É um pedaço de *software* que controla o *hardware*. Cada objeto de aplicação atribui um número único de ponto final que outros APOs podem usar uma extensão para o endereço do dispositivo de rede para interagir com ele. Pode haver até 240 objetos de aplicação num único dispositivo ZigBee. Uma aplicação ZigBee deve estar em conformidade com um perfil existente que seja aceite pela *ZigBee Alliance*. Um perfil de aplicação define formatos e protocolos de mensagens para interações entre objetos de aplicação. A aplicação estrutura de perfil permite que diferentes fornecedores construam e vendam independentemente os dispositivos ZigBee que podem interoperar entre si num perfil de aplicação.
- **Objetos de dispositivo ZigBee (ZDO).** A principal definição do ZigBee é objeto de dispositivo ZigBee, que aborda três operações principais: descoberta, segurança e ligação de serviços. O papel da descoberta é encontrar nós e perguntar sobre o seu endereço MAC do ZC ou ZR usando mensagens *unicast*. A descoberta também está facilitando o processo de localizar alguns serviços por meio de seus indicadores de perfil. Então, o perfil desempenha um papel importante. Os serviços de segurança neste objeto de dispositivo ZigBee têm a função de autenticar e derivar as chaves necessárias para criptografia de dados. A gestão de rede é implementada no coordenador e seu papel é seleccionar um PAN existente para interligar. Também suporta a criação de novos

PANs. O papel da gestão de ligação é vincular nós aos recursos e aplicações e também vincular dispositivos aos canais.

- **Subcamada de suporte à aplicação (APS).** Fornece uma interface entre as camadas NWK e APL através de um conjunto geral de serviços fornecidos pelas entidades de gestão e dados do APS. Esta subcamada é responsável pelo processamento de tramas de saída e saída para transmissão e receção com segurança as tramas e estabelecer e gerir as chaves criptográficas. O *APS Layer Security* inclui os seguintes serviços: estabelecer chave, chave de transporte, atualizar dispositivo, remover dispositivo, chave de solicitação, autenticação de entidade e tabela de configuração de permissões.
- **Fornecedor de serviços de segurança (*Security service provider*).** O ZigBee fornece mecanismo de segurança para a camada de rede e as subcamadas de suporte à aplicação, cada uma das quais é responsável por proteger suas tramas. O serviços de segurança incluem métodos para estabelecimento de chaves, transporte de chaves, proteção de tramas e gestão de dispositivos.

Os formatos das mensagens criadas pelas diferentes camadas da pilha protocolar ZigBee é apresentada na Figura 2.27 [40, 41].

2.4.3 ZigBee 3.0

O ZigBee 3.0 surgiu a partir da especificação ZigBee PRO 2015 (R21), recentemente foi atualizado foi atualizado pela especificação ZigBee PRO 2017 (R22). A especificação mais antiga baseada na especificação R21 ainda é compatível com a nova especificação R22 [42].

A especificação ZigBee PRO adiciona a gestão de dispositivos filhos, recursos de segurança aprimorados e novas opções de topologia de rede às redes ZigBee. O comissionamento de dispositivos nas redes também foi aprimorado e tornado mais consistente por meio do *Base Device Behavior* (BDB), estão no anexo A. Além disso, o ZigBee 3.0 fornece a funcionalidade *Green Power Basic Proxy* v1.1.1 em todos os dispositivos para oferecer suporte adicional aos recursos do *Green Power* (GP) e compila todos os *Clusters* de perfis numa única especificação, *ZigBee Cluster Library* (ZCL) v7. Por fim, formaliza a nomenclatura comum da arquitetura das aplicações de dispositivos ZigBee, expande a especificação de dispositivos *ZigBee Lighting and Occupancy* (ZLO), e adiciona a certificação ZigBee 3.0 [42].

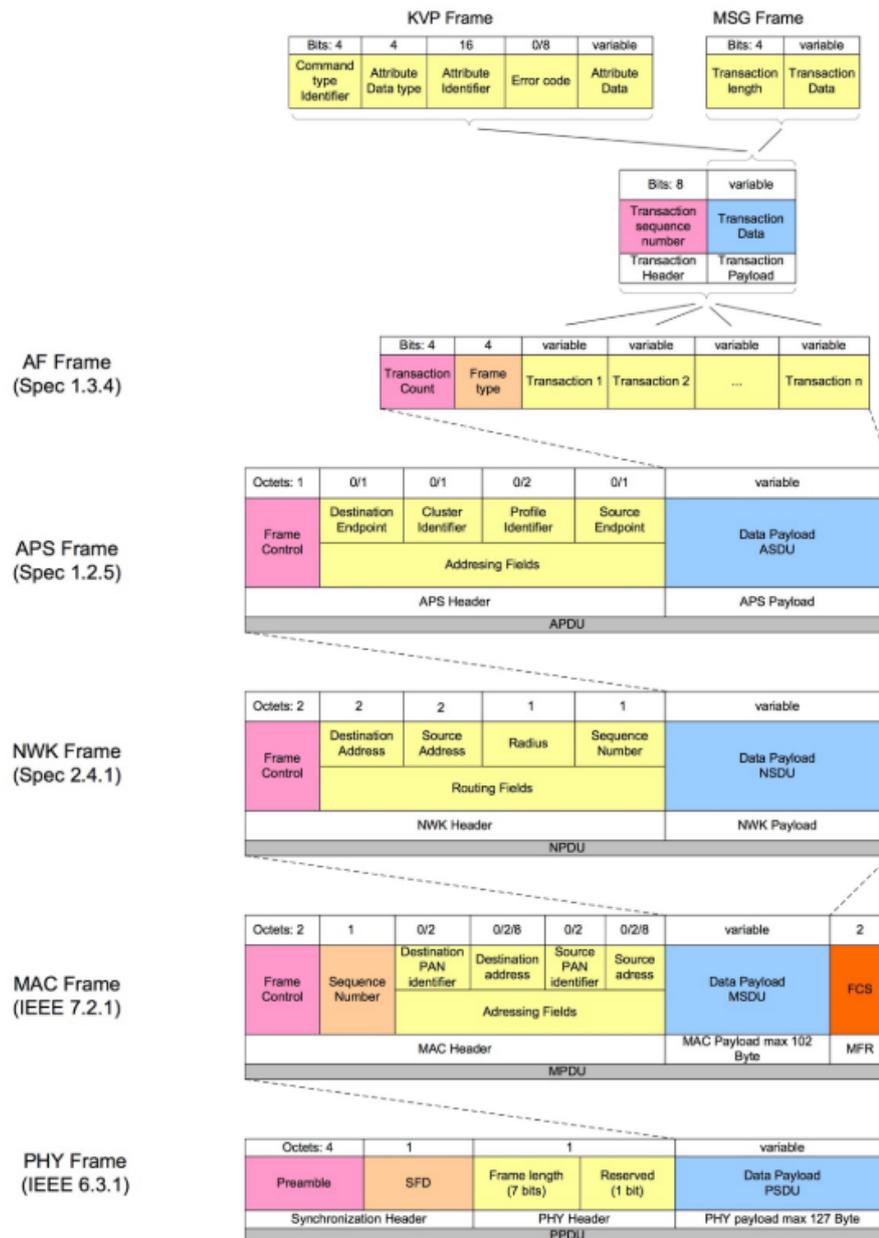


Figura 2.27: Formatos das mensagens da rede ZigBee.

Trust Center Link Key Updates (TCLK)

Os dispositivos R21 que ingressam numa rede centralizada ZigBee 3.0 devem iniciar um procedimento de atualização da chave de ligação do TC entrando na rede. Essa chave de ligação do TC exclusiva é usada para toda a comunicação da subcamada APS criptografada. A trama *Node Descriptor* é enviada durante o procedimento de associação de rede, indica a versão do dispositivo de ingresso. Os coordenadores R21 atuam como TC, podem ser configurados para aceitar ou rejeitar dispositivos herdados que não

iniciam o procedimento de atualização da chave de ligação TC [42]. Na Figura 2.28 é possível visualizar o procedimento de atualização da chave de ligação do TC [42].

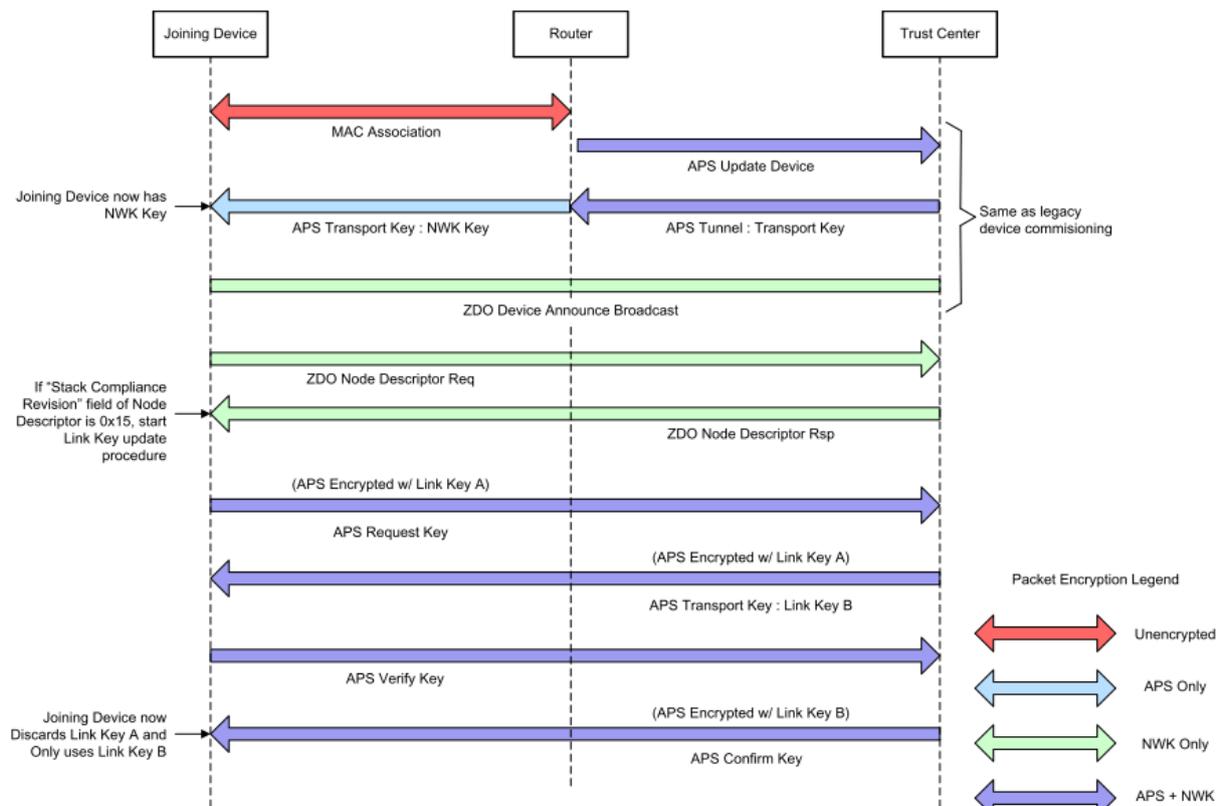


Figura 2.28: Procedimento de atualização da chave de ligação do TC.

2.5 ZigBee System-on-Chip

Radio Frequency System-on-Chip (RF-SoC), são circuitos de radiofrequência, sejam eles analógicos ou digitais, integrados com blocos de memória, microprocessadores e/ou DSP num chip formando um sistema de comunicação digital [43]. Diferente de se utilizar um microcontrolador ligado a um *transceiver* RF, com a utilização de um RF-SoC, o sistema será menos suscetível à captação de ruído externo, terá menores dimensões e possuirá uma maior eficiência energética [23, 43]. Estes dispositivos são normalmente utilizados em sistemas embebidos e *Internet of Things* (IoT). A Figura 2.29 mostra o diagrama de blocos típico da arquitetura de um RF-SoC [44].

A arquitetura de *hardware* dos nós para implementar a pilha protocolar ZigBee, é baseado no *System-on-Chip* (SoC) CCC2538 (baseado no microcontrolador Arm Cortex-M3 de 32 *bits*) [45] da TI [46], este

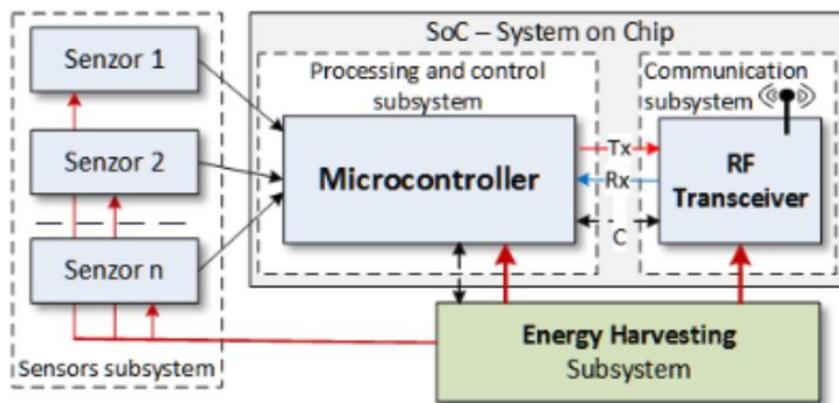


Figura 2.29: Diagrama de blocos típico de um RF-SoC.

SoC é composto por um *transceiver* RF compatível com o IEEE 802.15.4, integrados no mesmo *chip*, que será abordado na subsecção 3.5.1. Existem diferentes ZigBee SoC no mercado.

2.5.1 *Microchip System-on-Chip* ATSAMR21G18A

O *System-on-Chip* (SoC) ATSAMR21G18A [47] da *Microchip Technology Inc.* [48], o SoC combina num único *chip* um microcontrolador de 32 *bits* baseado em Arm Cortex-M0+ e um *transceiver* RF compatível com o IEEE 802.15.4. Este SoC ATSAMR21G18A foi concebido para aplicações *low power* na gama de frequências 2,4 GHz, e como tal, permite diversos modos de funcionamento. Na Figura 2.30 é possível visualizar a imagem deste SoC.

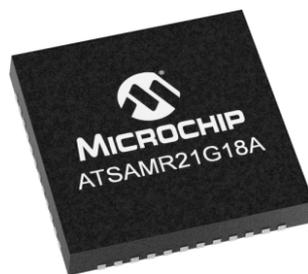


Figura 2.30: *System-on-chip* ATSAMR21G18A da *Microchip* no *package* QFN48.

As principais características do microcontrolador são [47]:

- Microcontrolador de 32 *bits* Arm Cortex-M0+;
- Frequência de relógio máxima de 48 MHz;
- Memória *Flash* de 256 kB;

- Memória *Static Random Access Memory* (SRAM) de 32 kB;
- *Debugging* por *Serial Wire Debug* (SWD);
- Interfaces: 5 SPI, 5 I2C, 4 UART, USART, USB;
- Interrupções externas: 15;
- 28 *General Purpose Input / Output Pins* (GPIO);
- ADC de 12 bits com 8 canais;
- 6 *Timers*
- Tamanho: (7 mm x 7 mm no *package* QFN48)
- *Transceiver* RF IEEE 802.15.4 de 2,4 GHz.

O ATSAMR21G18A é alimentado por uma tensão que pode variar de 1,8 V a 3,6 V. O SoC apresenta os seguintes modos de consumo de energia [47]:

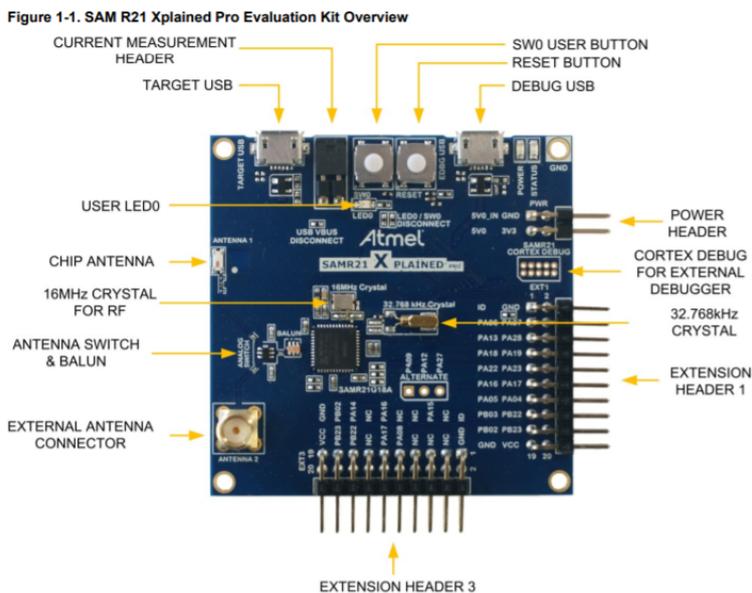
- Modo *deep sleep*: 20 nA;
- Modo Rx: 11,8 mA;
- Modo Tx (+4 dBm): 13,8 mA

O desenvolvimento de soluções neste SoC pode se tornar mais fácil, através de plataformas de desenvolvimentos disponíveis: *Atmel/SAM R21 Xplained Pro* [49] (Figura 2.31a) e *Atmel/ATSAMR21ZLL-EK* [50] (Figura 2.31b).

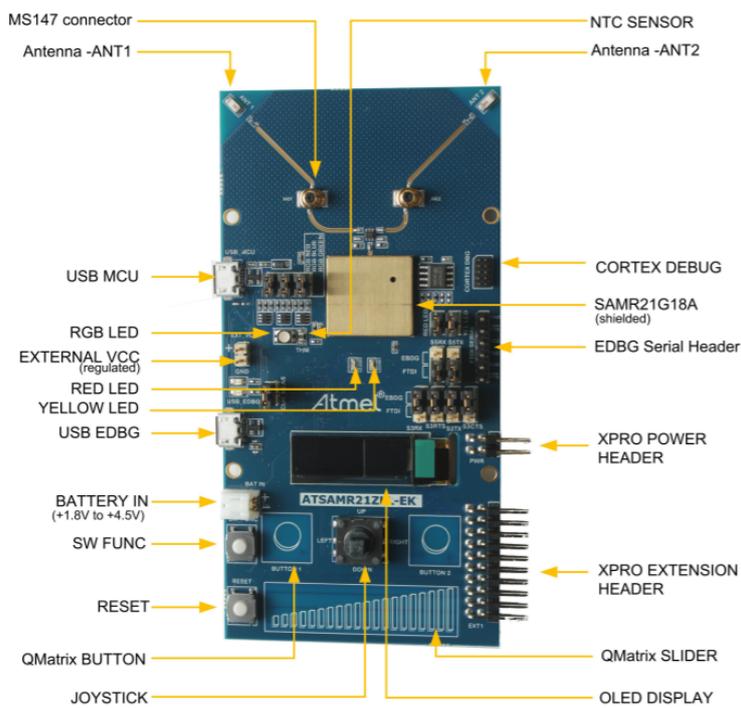
2.5.2 **NXP System-on-Chip JN5169**

O *System-on-Chip* (SoC) JN5169 [51] da *NXP Semiconductors* [52], é um SoC *ultra low power* e de alto desempenho, que suporta a pilha protocolar IEEE 802.15.4 / ZigBee para facilitar o desenvolvimento de aplicações *Smart Energy*, automação residencial, sistema de iluminação inteligente e sensores sem fios. Este SoC combina um processador RISC de 32 bits com um *transceiver* RF compatível com IEEE 802.15.4 de 2,4 GHz. Na Figura 2.32 é possível visualizar a imagem deste SoC.

As principais características do microcontrolador são [53]:



(a) SAM R21 Xplained Pro da Atmel.



(b) ATSAMR21ZLL-EK da Atmel.

Figura 2.31: Kits de desenvolvimento SAM R21 Xplained Pro e ATSAMR21ZLL-EK da Atmel.

- Processador de 32 *bits* RISC;
- Frequência de relógio máxima de 32 MHz;
- Memória *Flash* de 512 kB;



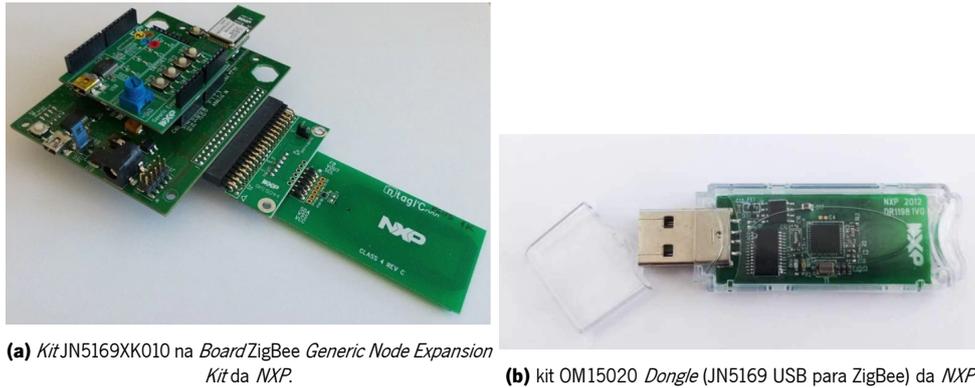
Figura 2.32: *System-on-chip* JN5169 da *NXP Semiconductors* no *package* HVQFN40.

- Memória *Random Access Memory* (RAM) de 32 kB;
- Suporta atualizações *Over the Air* (OTA) sem memória externa;
- *Debugging* por JTAG;
- Interfaces disponíveis: 2 UART, SPI, I2C;
- ADC de 10 *bits* com 6 canais;
- 5 PWM (4 *timers*, 1 *timer/counter*);
- 20 Digital I/O (DIO) *pins*;
- Tamanho : (6 mm x 6 mm x 0,85 mm no *package* HVQFN40)
- Sensor de bateria e de temperatura;
- *Transceiver* RF IEEE 802.15.4 de 2,4 GHz.

O JN5169 é alimentado por uma tensão que pode variar de 2,0 V a 3,6 V. O SoC apresenta os seguintes modos de consumo de energia [53]:

- Modo *deep sleep*: 50 nA;
- Modo Rx: 15 mA;
- Modo Tx (+10 dBm): 23,3 mA

No que diz respeito ao desenvolvimento, a *NXP Semiconductors* disponibiliza ferramentas que tornam mais eficiente o processo de desenvolvimento de soluções neste SoC. A Figura 2.33 apresenta o *kit* JN5169XK010 Figura 2.33a e o *kit* OM15020 *Dongle* (JN5169 USB for ZigBee) 2.33b da *NXP Semiconductors*.



(a) Kit JN5169XK010 na Board ZigBee Generic Node Expansion Kit da NXP.

(b) kit OM15020 Dongle (JN5169 USB para ZigBee) da NXP.

Figura 2.33: Kits de desenvolvimento JN5169XK010 e OM15020 Dongle da NXP.

2.5.3 Texas Instruments SoC CC1352P

O *System-on-Chip* (SoC) CC1352P [54] da *Texas Instruments*, é composto por um microcontrolador de 32 bits e por um *transceiver* RF multiprotocolo que funciona na gama de frequências de sub-1 GHz e de 2,4 GHz, compatível com o *Wireless* M-Bus, IEEE 802.15.4g, 6LoWPAN, Thread, ZigBee, KNX RF, BLE e protocolos proprietários. Este SoC contém integrado um amplificador de alta potência de +20 dBm com melhor eficiência para aplicações *low power* de longo alcance. Na Figura 2.34 é possível visualizar a imagem deste SoC.



Figura 2.34: *System-on-chip* CC1352P da *Texas Instruments* no package VQFN48.

As principais características do microcontrolador são [54]:

- Microcontrolador de 32 bits Arm Cortex-M4F;
- Frequência de relógio máxima de 48 MHz;
- Memória *Flash* de 352 kB;
- Memória *Read Only Memory* de 256 kB para bibliotecas e protocolos;
- *Debugging* por cJTAG e JTAG;

- Interfaces: 2x UART, 2x SSI (SPI, MICROWIRE, TI), I2C, I2S;
- Suporta atualizações *Over the Air* (OTA);
- 26 *General Purpose Input / Output Pins* (GPIO);
- ADC de 12 bits com 8 canais;
- Tamanho (7 mm x 7 mm no *package* VQFN48).

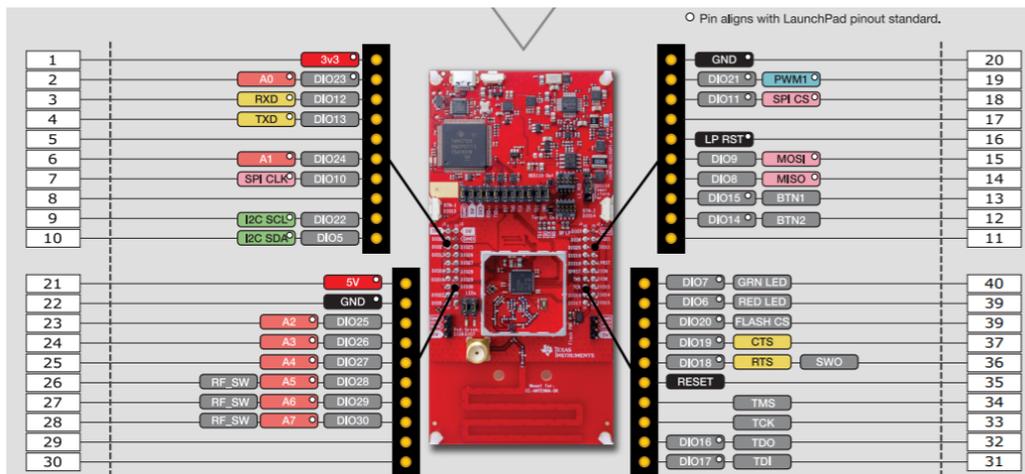
As principais características do *transceiver* RF são [54]:

- Dupla banda de frequência sub-1 GHz e 2,4 GHz, compatível com BLE e IEEE 802.15.4 (camada PHY e MAC);
- Transmissores, receptores e amplificador integrados no *transceiver* RF;
- Potência de sinal: +20 dBm;
- sensibilidade
 1. -121 dBm no modo *longe range*;
 2. -110 dBm com débito de 50 kbps;
 3. -105 dBm para Bluetooth com débito de 125 kbps.

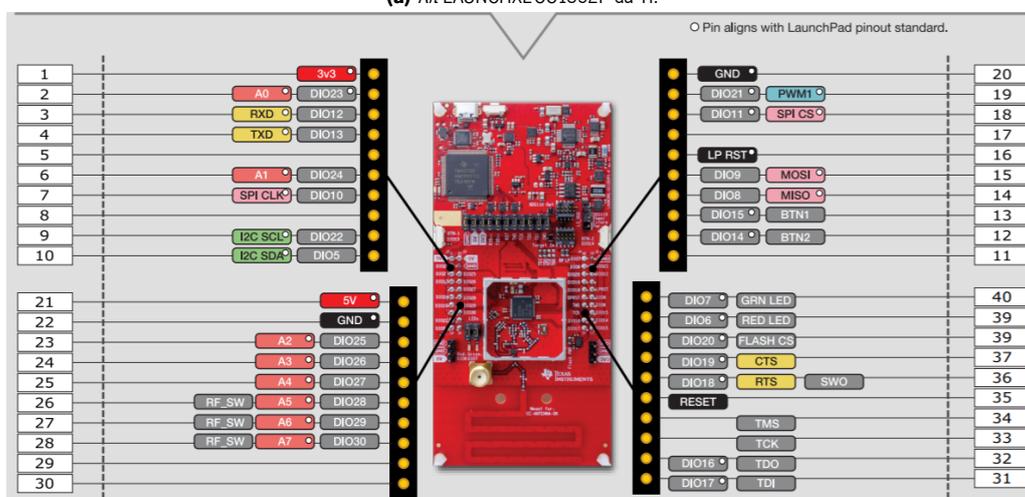
O CC1352P é alimentado por uma tensão que pode variar de 1,8 V a 3,8 V. Utiliza dois cristais externos de 32,768 kHz. O SoC apresenta os seguintes modos de consumo de energia [55]:

- Modo *deep sleep*: 150 nA (só acorda via interrupção externa);
- Modo ativo (48 MHz): 2,9 mA;
- Modo RX (3 V, 2,4 GHz): 6,9 mA;
- Modo TX em +20 dBm: 85 mA (3 V, 2,4 GHz).

No desenvolvimento de uma solução, o uso de plataformas de desenvolvimento, permite a aprendizagem de uma nova tecnologia, prova de conceito de determinadas soluções e o processo de prototipagem. A Figura 2.35 apresenta o *kit* LAUNCHXL-CC1352P Figura 2.35a e o *kit* LAUNCHXL-CC1352P-2 Figura 2.35b da TI.



(a) *Kit* LAUNCHXL-CC1352P da TI.



(b) *Kit* LAUNCHXL-CC1352P-2 da TI.

Figura 2.35: *Kits* de desenvolvimento LAUNCHXL-CC1352P e LAUNCHXL-CC1352P-2 da TI.

2.6 Stacks da Texas Instruments

A *Texas Instruments* (TI), é uma empresa norte-americana que desenvolve e fabrica semicondutores para dispositivos eletrônicos. Esta empresa é caracterizada também desenvolvimento e fornecimento de *softwares* que permitem a implementação da pilha protocolar para dispositivos sem fios.

A comparação entre várias soluções sem fios da TI é apresentada na Figura 2.36 [56].

2.6.1 SimpliciTI

A *stack SimpliciTI* [57] é um protocolo totalmente *Open Source* de baixa complexidade para as LR-WPAN, desenvolvido pela *Texas Instruments*. Esta *Stack* ocupa tamanho muito reduzido de memória, resulta

Sub 1-GHz	2.4-GHz			5-GHz	
SimpliciTI™ (open protocol) Proprietary	PurePath® Wireless Audio Proprietary	ZigBee® (PRO & IP) 6LoWPAN 802.15.4 RF4CE	Bluetooth® Bluetooth low energy ANT	Wi-Fi® 802.11 a/b/g/n + Bluetooth + FM	GPS + Bluetooth + FM
Applications					
Metering Home automation Electronic shelf labeling Alarm & security	Headphones Gaming headsets Speakers HID/ Gaming	Smart metering Sensor networks Remote control	Mobile accessories Gaming controllers Consumer medical, health & fitness	Portable media players Portable data terminals eBooks Camera Navigation Asset tracking Remote clocking	
Products					
CC1101/10 CC430 CC112x	CC8520 CC2500/10	CC2520 CC2530/1 CC2533	CC254x CC256x CC257x	WL127x	NL55xx
				WL128x	

Figura 2.36: Comparação entre várias soluções sem fios da TI.

numa boa solução para módulos compatíveis com o padrão IEEE 802.15.4, com pouco poder de processamento ou armazenamento [29]. Na Figura 2.37 é possível visualizar a arquitetura geral da *Stack SimpliciTI* [58].

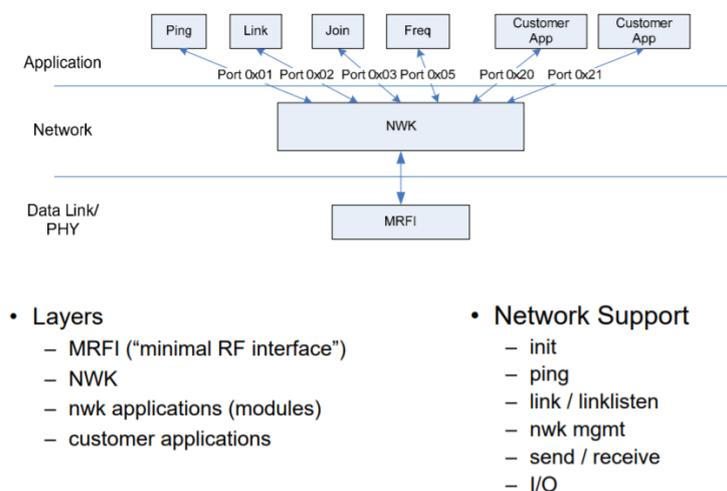


Figura 2.37: Arquitetura geral da Stack SimpliciTI.

As principais características do SimpliciTI são [58]:

- Protocolo de rede proprietário da TI para dispositivos de RF *low-power*;
- Utiliza dois tipos topologias de redes básicas, que são topologia em estrela ou *peer-to-peer*;
- Utiliza três tipos de nós: *End Device* (ED), *Repeater* e *Access Point* (AP);
- Utiliza pouca taxa de ocupação do processador;

- Suporta modo de poupança de energia.

O SimpliciTI está atualmente disponível para os dispositivos e/ou *transceiver* da TI, MSP430+CC110x / 2500, CC1110 / 2510, CC1111 / CC2511, CC2430 e CC2520 ,na gama de frequência 433 MHz, 868 MHz e 2.400 MHz, suportando modulação OOK, FSK, GFSK, MSK. normalmente, tem uma taxa de transmissão de dados OTA de 250 kbps, que pode ser alterada facilmente pelos utilizadores, assim como outras configurações de rádio. Diferente do IEEE 802.15.4, o SimpliciTI permite que os utilizadores escolham se gostariam de usar ou não o CCA [59]. Na Figura 2.38 é ilustrado o formato da trama SimpliciTI [59].



Figura 2.38: formato da trama SimpliciTI.

2.6.2 Z-Stack

A Z-Stack [60] é um sistema operativo baseado em *rotary query* [61], desenvolvido pela TI, que implementa a pilha protocolar IEEE 802.15.4 / ZigBee, com a exceção da camada física que está implementada ao nível do *hardware* [40]. Esta *Stack* também é responsável pela disponibilização de uma *Application Programming Interface* (API), escrita em linguagem C, para a camada *Hardware Abstraction Layer*. (HAL), camada *Operating System Abstraction Layer* (OSAL) e o *user application* [22, 60].

O seu princípio funcionamento é que, quando há um evento, a camada OSAL distribui esse evento para a tarefa que é capaz de lidar com esse evento com responsabilidade e, em seguida, essa tarefa determina o tipo de programa de processamento apropriado para lidar com o evento [61].

Em geral, a Z-Stack tem duas funções. A primeira é a inicialização do sistema e a outra está iniciando a entidade da Z-Stack. A entidade da Z-Stack possui apenas uma linha de código, a saber, "*osal_start_system()*", *osal_start_system()* chama (*taskArr[idx]*) (*idx, events*) para executar funções específicas de processamento. *taskArr[]* é um vector de apontadores de função e executa funções diferentes, dependendo de diferentes IDs variáveis. OSAL é o núcleo de processamento da *Stack*. Cada tarefa do OSAL é planeada para um subsistema da Z-Stack no processo de inicialização da camada OSAL. *sosal_Add*

Tasks() do *osal_inti_system* fornece e adiciona listas de tarefas que atendem às necessidades dos usuários de maneira limitada [61].

Geralmente, a inicialização do sistema inclui várias funções, descritos abaixo [61]:

- Inicialização da frequência de relógio;
- Detecção da tensão de alimentação do *chip*;
- Inicialização da *Stack* em cada módulo de *hardware*;
- Inicialização da memória *flash*;
- Formação do endereço MAC;
- Inicialização da variável *nonvolatile* e a camada MAC;
- Inicialização do sistema operativo;
- Inicialização da aplicação.

A interação da Z-Stack e camada de aplicação com as camadas OSAL e HAL é possível visualizar na Figura 2.39 [8].

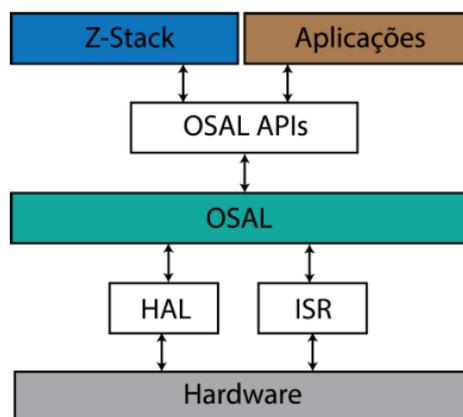


Figura 2.39: Interação da Z-Stack e camada de aplicação com as camadas OSAL e HAL.

A forma como a Z-Stack se comunica com outros dispositivos pode ser descrita na Figura 2.40 [62]. Os clientes iniciam um comando ou solicitam um pedido, e os servidores agem sob um comando ou entregam a resposta.

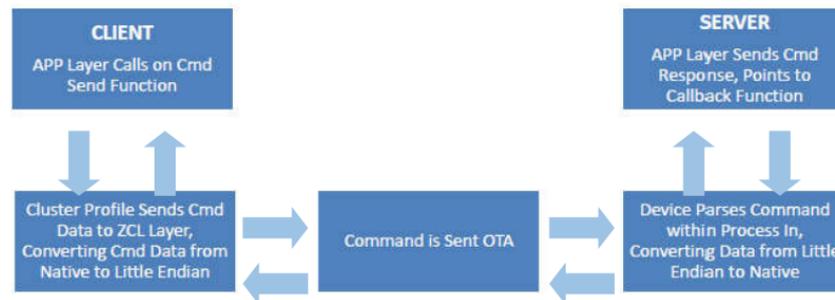


Figura 2.40: Digrama de blocos do processo de de comunicação da Z-Stack.

2.7 Resumo

Neste capítulo foi apresentado o estado da arte relativamente a sistema de monitorização de colisões com guardas de segurança. Foi apresentado também um estudo das WSNs e do padrão IEEE 802.15.4, uma vez que estão diretamente relacionados com o desenvolvimento deste projeto. É também apresentado um resumo sobre a rede ZigBee e os SoCs que suportam este protocolo. Por último, foi apresentado as *stacks* fornecidas pela TI.

Capítulo 3

Especificação do Sistema

Neste capítulo pretende-se especificar o sistema a desenvolver. Primeiramente serão apresentados os requisitos a que este projeto se submete. Posteriormente, é apresentada a visão geral do sistema, e especificado o *hardware* e *software* utilizado.

3.1 Visão Geral do Sistema

O sistema a desenvolver nesta dissertação tem como objetivo sugerir melhorias a WSN do projeto já existente. Baseados em redes de sensores sem fios, que tem como principal missão detetar a ocorrência de colisões de veículos em guardas de segurança da autoestrada e obter periodicamente o estado do nó sensor.

Na Figura 3.1 estão representados os principais componentes do sistema. O nó sensor encontra-se fixado à guarda de segurança, é o elemento responsável por detetar a ocorrência de colisões de veículos e informar ao nó coordenador o ocorrido, o nó coordenador é o elemento responsável por reencaminhar os dados recebidos do nó sensor para um *gateway*. Este, por sua vez é o elemento responsável por enviar, via Ethernet ou Wi-Fi, os dados recebidos pelo coordenador para os microserviços que carrega uma base de dados. A partir do momento que as informações estejam acessíveis na base de dados pode ser consultada, por um utilizador autenticado, através de uma aplicação remota ou local. A WSN é baseada numa topologia de rede em estrela.

3.2 Requisitos do Sistema

Os requisitos deste sistema podem ser funcionais e não funcionais. Os requisitos funcionais correspondem a um conjunto de funcionalidades do sistema, enquanto que os requisitos não funcionais descrevem

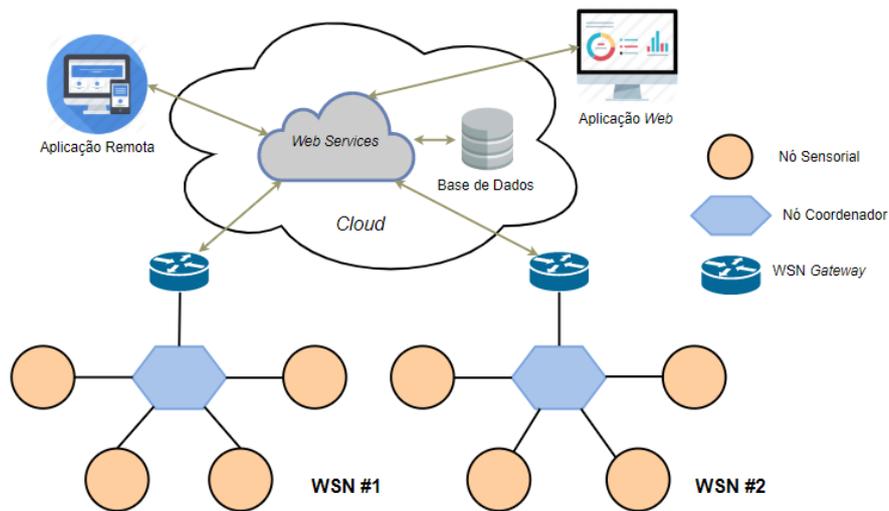


Figura 3.1: Visão geral do sistema.

um conjunto de restrições impostas ao sistema, como por exemplo, a segurança, desempenho, espaço ocupado, entre outros.

Requisitos Funcionais

O sistema é definido pelos seguintes requisitos funcionais:

- Garantir a fiabilidade e robustez na deteção de uma colisão de um veículo com os guardas de segurança, através da leitura de uma aceleração superior à predefinida;
- O nó sensor tem de reportar para o nó coordenador assim que detete uma colisão, quando os limites da aceleração esperada são ultrapassados;
- O nó sensor deve enviar periodicamente mensagens para o coordenador, designadas por *Keep Alive* (KA), que serve para identificar se o nó sensor ainda se encontra no ativo;
- O nó sensor tem de enviar uma mensagem em intervalos constante para o nó coordenador, designado por KA, que serve para indicar se o nó sensor se encontra na rede;
- O nó sensor tem de comutar para um modo de muito baixo consumo de energia (*deep sleep*) sempre que possível, reativando o nó somente quando há uma deteção de colisão ou KA por enviar;

- O nó coordenador tem de encaminhar as mensagens recebidas do nó sensor para um *gateway* no formato JSON;
- O *gateway* tem de enviar os dados recebidos dos nó coordenador para os microserviços. Caso os microserviços estejam indisponíveis, armazena os dados temporariamente numa base dados até que seja restabelecido o funcionamento desse serviço.

Requisitos Não Funcionais

Os requisitos não funcionais definidos para o sistema são:

- O nó sensor tem de ter longa autonomia (no mínimo dois anos, utilizando baterias) e com complexidade muito baixa;
- O nó coordenador tem de operar ligado à rede elétrica;
- O coordenador tem de estar conectado ao *gateway*, através da comunicação UART;
- O *gateway* tem de ter ligação à Internet, via Ethernet ou Wi-Fi;
- Utilizar o módulo CC2538 / CC2592 da TI;
- Utilizar acelerómetro MEMS LIS331DLH da ST;
- Utilizar Z-Stack 3.0.2 da TI;
- Utilizar microserviços desenvolvido pelo ESRG.

3.3 Arquitetura do Sistema

Este projeto é composto por uma WSN responsável por detetar a ocorrência de uma colisão, através de nós sensoriais fixados às guardas de segurança, equipados com sensores de aceleração. As informações adquiridas são enviadas para o nó coordenador contido a um *gateway* com ligação à internet, responsável por enviar estas informações para uma base de dados. Na Figura 3.2 estão representados os principais componentes do sistema.

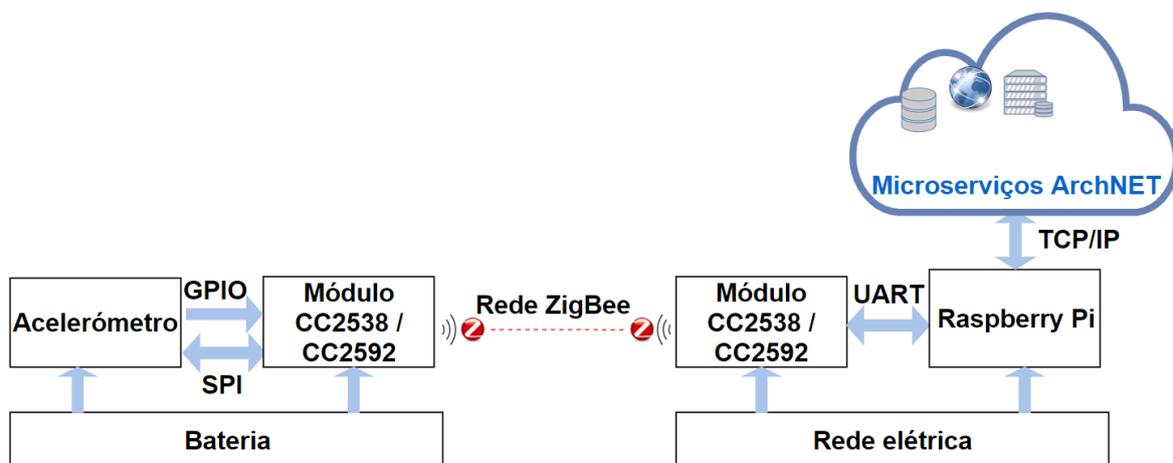


Figura 3.2: Diagrama de blocos do sistema.

3.4 Casos de Uso do Sistema

O sistema a desenvolver nesta dissertação será constituído por dois casos de usos: casos de uso para os veículos e o tempo, como é ilustrado na Figura 3.3.

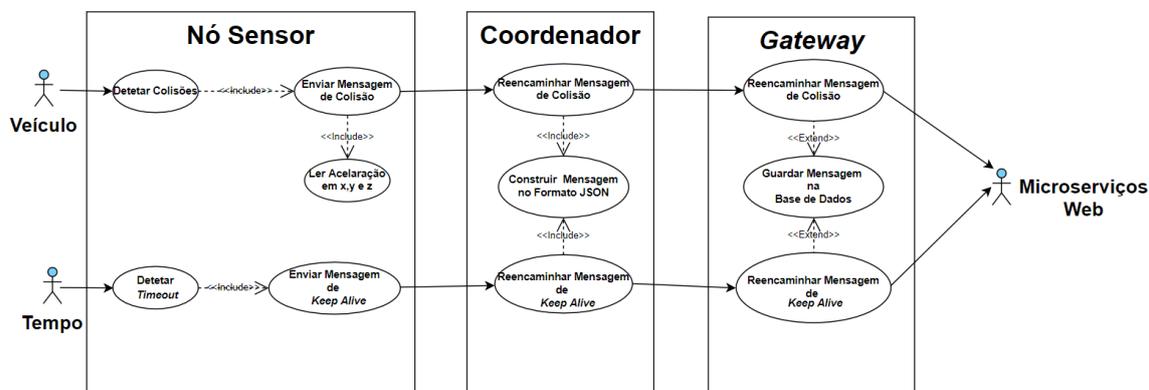


Figura 3.3: Diagrama de casos de uso do sistema

O sistema é composto por vários subsistemas e diferentes atores (veículos e o tempo), que podem originar diferentes tipos de mensagens do sistema. Quando um veículo causa uma colisão com os guardas de segurança, o nó sensor acorda e faz da aceleração nos eixos x, y e z através do acelerómetro incorporado nele. Uma vez que a leitura termine, o nó sensor através do *transceiver* RF envia uma mensagem de colisão para o nó coordenador. Por sua vez, o nó coordenador converte essa mensagem em formato JSON e reencaminha para o *gateway*. Quando o *gateway* recebe a mensagem e tiver ligação

à internet e acesso aos microserviços web, envia a mensagem para os microserviços *web*, caso contrário a mensagem será armazenada temporariamente numa base de dados local até que tenha acesso à internet e aos microserviços. Quando ocorre um *timeout*, o nó sensor acorda e envia uma mensagem de KA e segue as mesmas etapas do primeiro caso de uso. O envio periódico de KA permite reportar a operabilidade do nó sensor.

3.5 Especificação do *Hardware*

Nesta secção são analisadas as tecnologias e alguns componentes adequados para a implementação da solução, nomeadamente, o microcontrolador que vai implementar o nó sensor e o coordenador, o dispositivo que vai implementar o *gateway* e a bateria que vai alimentar os nós. Todos os componentes de *hardware* foram definidos no início deste trabalho, a escolha foi feita de acordo com o *feedback* positivo de outros trabalhos realizados pelo grupo de investigação *Embedded System Research Group* (ESRG) do Centro ALGORITMI da Universidade do Minho.

3.5.1 *System-on-Chip* (SoC)

Para a implementação dos nós da rede de sensores sem fios, na gama de frequências 2.4 GHz, optou-se por se utilizar um SoC da TI, o CC2538 [45], tal como ilustra a Figura 3.4 . Este SoC é considerado para aplicações ZigBee de alto desempenho. O dispositivo combina um microcontrolador baseado em Arm Cortex-M3 com um *transceiver* RF compatível com o padrão IEEE 802.15.4. Pode ainda alcançar um débito de 250 kbps.



Figura 3.4: SoC CC2538 da Texas Instruments no package QFN56.

O microcontrolador incorporado no SoC apresenta um tamanho reduzido (8mm x 8mm no *package* QFN56), facilitando a sua integração em sistema com limitações de dimensões. Este SoC devido a sua

natureza de baixo consumo energético, e permite diferentes modos de operação, cumprindo assim, com os requisitos da rede de sensores sem fios.

As principais características do microcontrolador são [45]:

- Microcontrolador de 32 *bits* Arm Cortex-M3;
- Frequência de relógio máxima de 32 MHz;
- Memória *Flash* de 128 kB, 256 kB e 512 kB;
- Memória RAM de 32 kB;
- *Debugging* por cJTAG e JTAG;
- Interfaces: 2 * SSI, 2 * SPI, 2 * UART, I2C, USB;
- Suporta atualizações OTA;
- 32 GPIO;
- ADC de 12 *bits* com 8 canais;

As principais características do *transceiver* RF são [45]:

- Banda de frequência 2,4 GHz, compatível com IEEE 802.15.4, ZigBee e 6LoWPAN;
- Sensibilidade máxima de -97 dBm;
- Potência de sinal máxima de +7 dBm;
- Transmissor e recetor integrados no *transceiver* RF.

O CC2538 é alimentado por uma tensão que pode variar de 2 V a 3,6 V. O SoC apresenta os seguintes modos de consumo de energia [45]:

- Modo PM3 (*deep sleep*): 0,4 μ A;
- Modo PM2: 1,3 μ A;
- Modo PM1: 0,6 mA;
- Modo Rx: 20 mA;

- Modo Tx (0 dBm): 24 mA.

O módulo selecionado para o desenvolvimento do projeto resulta da combinação entre o SoC CC2538 da TI e o SoC CC2592 da TI [63]. O *System-on-Chip* (SoC) CC2592 é um *range extender* de RF *low power* de alto desempenho e eficiência para dispositivos com *transceiver* RF que operam na gama de frequências 2.4 GHz, permite alcances relativamente longos. Este *range extender* de RF é constituído por um amplificador de potência (PA) para aumentar a potência de transmissão e por um *Low Noise Amplifier* (LNA) que permite melhorar a sensibilidade na receção de dados. Este SoC apresenta um tamanho muito reduzido (4 x 4 no *package* QFN16). Foram realizadas alterações no módulo, nomeadamente foram retirados e substituídos condensadores, de modo a minimizar o consumo de energia quando alimentado a bateria.

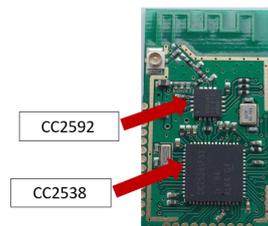


Figura 3.5: Módulo com SoC CC2538 e o *Range Extender* CC2592.

3.5.2 Gateway

O dispositivo selecionado para implementar a interface entre a WSN e os microserviços *web*, foi a Raspberry Pi 3 *model/B* (Figura 3.6) da *Raspberry Foundation* [64]. Esta placa foi desenvolvida com o intuito de facilitar a aprendizagem de *software* para crianças, com o andar do tempo tornou-se numa ótima opção para concepção de produtos, quando não precisam de operar em ambientes instáveis.

A placa de desenvolvimento é baseado no processador Broadcom BCM2837 e no processador *Quad-Core* Arm Cortex-A53 de 64 bits a 1,2 GHz. Este dispositivo incorpora conectividade *wireless* LAN 802.11b/g/n e *Bluetooth* clássico e BLE [65].

A Raspberry Pi 3 *model/B* para o seu correto funcionamento, deverá ter no mínimo uma corrente de 2,1 A. As principais características do dispositivo são [65]:

- Processador Broadcom BCM2837 e Arm Cortex-A53 de 64 bits;
- Memória *Synchronous Dynamic Random Access Memory* (SDRAM) de 1 GB;



Figura 3.6: Placa de desenvolvimento RaspBerry Pi 3 *model/B*.

- Armazenamento por *MicroSD*;
- Porta LAN de 1 x10 / 100 Mbps;
- Comunicação sem fios por Placa *onboard*/Wi-Fi BCM43438 + Bluetooth 4.1;
- 40 GPIO;
- Alimentação máxima de 2,5 A e 5 V.

3.5.3 Acelerómetro

Relativamente ao acelerómetro que efectua as leituras de aceleração provocada pela ocorrência de uma colisão, foi utilizado o LIS331DLH da ST. Este acelerómetro MEMS tem como característica, serem de baixo consumo energético com 5 modos de configuração (*low power*), interface SPI e I2C, resolução de 16 *bits* e permitem medir acelerações de ± 2 g, ± 6 g e ± 8 g em 3 eixos. O seu tamanho reduzido (3mm x 3mm no *package* LGA 16). Na Figura 3.7 é possível visualizar a imagem do acelerómetro.

O LIS331DLH pode ser alimentado por uma tensão que pode variar 2,16 V e 3,6 V. O acelerómetro apresenta os seguintes modos de consumo de energia [66]:

- Modo *power down*: 1 μ A;
- Modo *ultra low power*: 10 μ A;
- Modo normal: 250 μ A.



Figura 3.7: Acelerómetro MEMS LIS331x.

3.5.4 Bateria

A bateria selecionada para fornecer energia para o nó da rede de sensores fios é a LS17500 da SAFT *Batteries* [67]. Este tipo de bateria é composta por *Lithium Thionyl Chloride* (Li-SOCl_2), são baterias primárias (descartáveis) com um tamanho reduzido (aproximadamente 5 cm x 1,7 cm), sendo que possuem uma tensão nominal de 3,6 V e tem uma gama de de temperaturas de funcionamento elevado entre -60 a +85 °C. Na Figura 3.8 é possível visualizar a imagem da bateria LS17500.



Figura 3.8: Bateria SAFT LS17500.

3.6 Especificação do *Software*

Nesta secção serão analisados alguns detalhes sobre o funcionamento do sistema. Além disso, aborda-se a seleção da Z-Stack, formato da mensagem a enviar para o *gateway*.

3.6.1 Z-Stack 3.0.2

Na escolha do sistema operativo para o SoC foram excluídos os sistemas operativos genéricos, como é o caso do TinyOS, SenSpire OS ou Contiki, dado que a implementação destes sistemas revelam-se não ser uma solução ótima em termos de desempenho e consumo de energia [32, 68]. A Z-Stack 3.0.2 é a versão do *software* da TI que implementa a especificação ZigBee 3.0 para os SoCs CC253x. Este conjunto de APIs foi proposto para implementar a camada aplicacional da WSN do projeto, utilizando como base uma das aplicação exemplo que esta continha, denominada *GenericApp*.

A *GenericApp* é um *template* de aplicação que não inclui nenhum exemplo de funcionalidade, é um bom ponto de partida para desenvolver a sua própria aplicação sem a sobrecarga de uma *Sample App*. A aplicação regista apenas os *Clusters* básicos e *Identify* no *EndPoint 0x08*. Outras funcionalidades do *Cluster* devem ser adicionadas para atender ao comportamento específico do sistema [69].

Para além dessa *Sample App*, a *Stack* disponibiliza também outras aplicações: *SampleDoorLock*, *SampleDoorLockController*, *SampleLight*, *SampleDoorLockController*, *SampleLight*, *SampleSwitch*, *SampleTemperatureSensor* e *SampleThermostat*. Cada uma das soluções permite configurar os dispositivos como ZC, ZR ou ZED.

Se um *End Device* (ED) perder contacto com o dispositivo pai ou for redefinido enquanto estiver em rede, o módulo BDB notificará a aplicação sobre o estado *BDB_COMMISSIONING_PARENT_LOST*, após o qual o ED não poderá executar nenhum outro método de comissionamento. O dispositivo deve restaurar sua rede localizando outro dispositivo pai ou fazer o *reset* e depois ser comissionado novamente [62].

Para restaurar a rede, o dispositivo deve chamar a função *bdb_ZedAttemptRecoverNwk ()*, isso fará com que o dispositivo execute uma única procura ativa no mesmo canal em que fazia parte da rede, à procura de qualquer pai apropriado (mesmo *Extended PAN ID*). Isso significa que o dispositivo enviará apenas uma solicitação de *beacon* e, se nenhum dispositivo pai adequado for encontrado, outra notificação *BDB_COMMISSIONING_PARENT_LOST* será enviada para aplicação [62].

A aplicação é responsável por tentar restaurar a rede, mas é recomendável ter um período em que as tentativas tenham um intervalo curto e depois passem para um intervalo maior, para reduzir o consumo de energia [62].

3.6.2 Formato de Dados JSON

JavaScript Object Notation (JSON) é um popular formato leve de dados semiestruturado, baseado nos tipos de dados da linguagem *JavaScript* [70]. Essa popularidade, ocorreu devido à facilidade de escrita e/ou neste formato, quer seja por humanos ou máquinas [71]. Tornou-se o principal formato de troca de dados na internet nos últimos anos [70]. O JSON encontra-se em formato de texto, independente de linguagem de programação, mas usa convenções que são familiares às usadas em linguagens de programação como C, C++, *Java*, *Perl*, *Python*, entre outras [71]. Um exemplo de um documento em formato JSON 3.9.

```
{ "NTW" :  
  { "WPANID": "0x244",  
    "GROUPID": "0",  
    "CLUSTERID": "1",  
    "GATEWAYMAC": "00:12:4b:0005:b3:c2:a6",  
    "ENDPOINT": "10",  
    "BROADCAST": "0",  
    "LINKQUALITY": "78",  
    "CORRELATION": "99",  
    "RSSI": "-57",  
    "SECURITY": "0",  
    "SEQNUMBER": "36"},  
  "APP" :  
    { "DEVICEMAC": "00:04:00:0000:01:ff:ff",  
      "SAMPLING_RATE": "1",  
      "DEVICETYPE": "4"} }
```

Figura 3.9: Exemplo de um documento em formato JSON.

A estrutura base de um formato de dados JSON, é constituído por [71]:

- Objeto, é um conjunto não ordenado de pares chave-valor. Um objeto inicia com “{” e termina com “}”, cada chave é seguida por ‘:’ e os pares chave-valor são separados por ‘,’;
- *Array*, é uma coleção ordenada de valores que inicia com ‘[’ e termina com ‘]’, sendo cada elemento do *array* separado por ‘,’;
- Valor, pode ser um tipo primitivo (*string*, número ou *boolean*), um valor nulo, um objeto ou *array*.

3.6.3 Base de Dados MongoDB

MongoDB [72] é uma base de dados *open source* orientada a documentos, desenvolvida em C++. Criada pela empresa de *software 10gen* para ser usada como serviço para outros *softwares*. Em 2009, passou

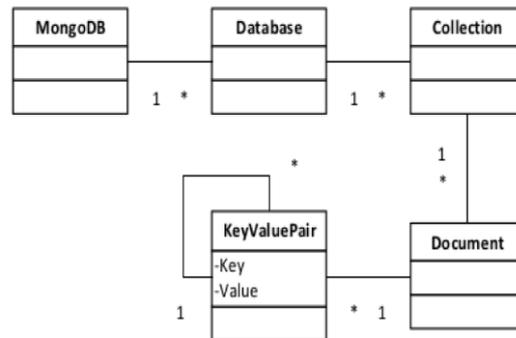


Figura 3.10: Estrutura básica da base de dados MongoDB.

para *open source*, tornando o MongoDB como *software* de *back-end* para muitos sites [73]. Na Figura 3.10 [73] é possível visualizar a estrutura base do MongoDB.

A estrutura básica da base de dados MongoDB é a seguinte [73]:

- *Collection*, é igual à estrutura da tabela. Ela armazena vários registros que são do mesmo tipo ou relacionadas, mas que servem ao mesmo propósito;
- *Documents*, todos os dados são armazenados no objeto chamado *Document*. Possui vários campos e valores relacionados a esse registro específico;
- *Key-Value*, armazena dados em formato JSON, que o MongoDB chama em formato *Binary JavaScript Object Notation* (BJSON). Possibilita o armazenamento de diferentes tipos de *Documents* na mesma *Collection*.

Este *software*, fornece alto desempenho usando o protocolo binário para interagir com o servidor. O MongoDB não usa solicitação HTTP ou REST. Possui otimizador de consultas que lembra como executar a consulta mais rapidamente. O seu mecanismo de armazenamento de arquivos é mapeado na memória, que deixa na responsabilidade da gestão de memória do sistema operativo [73].

3.6.4 Microserviços ArchNET

Criado pelo grupo ESRG da Universidade do Minho, ArchNet são microserviços responsáveis por implementar redes seguras de dispositivos. Essa *framework*, tem como objetivo principal a redução de *overhead* dos protocolos de comunicação [74]. Na figura 3.11 [74] é possível visualizar a arquitetura do sistema ArchNET.

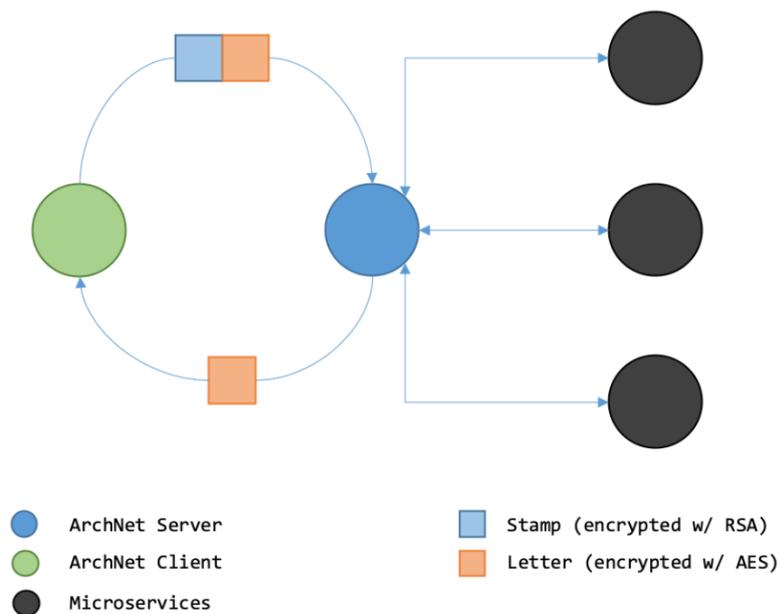


Figura 3.11: Arquitetura do sistema ArchNET.

A mensagem de solicitação enviada pelo cliente ao servidor possui dois elementos [74]:

- O *stamp*, tem um tamanho fixo de 128 *bytes* de dados enviados na mensagem. O *stamp* melhora a segurança e permite uma comunicação rápida e eficiente entre o cliente e o servidor;
- O *letter*, é a mensagem enviada pelo cliente ou pelo servidor.

Estabelece ligação com o cliente via TCP/IP, os dados são criptografados de diferente forma *end-to-end*, para garantir que a comunicação seja segura. O *stamp* é criptografado usando uma chave pública fornecida pelo servidor, e o *stamp* no *letter* usa a chave AES [74].

3.6.5 Aplicação do Nó Sensor

Usando a camada de aplicação da pilha protocolar Z-Stack 3.0.2 da TI será desenvolvido o *software* do nó sensor. Este *software* será desenvolvido e compilado no IDE *IAR Embedded Workbench for Arm*. Para o aumento da eficiência energética do sistema, o nó será colocado em *deep sleep* quando não tiver uma tarefa a realizar. As tarefas a realizar pelo nó serão: a detecção de uma colisão e o envio periódico de KA. Para a detecção de uma colisão será implementado uma interrupção externa no SoC, causada pelo acelerómetro, quando detetar uma aceleração superior aos limites esperados da aceleração. O envio periódico de KA será provocado pelo *sleep timer* do microcontrolador que está configurado para acordar

o nó, quando ocorre o *timeout* predefinido pelos responsáveis da gestão sistema, o processo de envio da mensagem seguem os mesmos procedimentos do primeiro caso de uso. Esse tipo de mensagem permite informar que o nó está em rede.

Os principais módulos da aplicação do nó sensor são:

- **Módulo Principal da Aplicação.** É responsável pela configuração e obtenção dos parâmetros necessários para o funcionamento da aplicação. Para além disso, é responsável também pela definição e execução das tarefas da aplicação;
- **Módulo da Estrutura Dados da Aplicação.** É responsável pela definição de dados principais no envio das mensagens (de colisão e de KA);
- **Módulo do Perfil da Aplicação.** É responsável pela definição do perfil privado da aplicação utilizando o registo do *EndPoint* da aplicação da camada AF do dispositivo;
- **Módulo do acelerómetro.** É responsável pela deteção de colisões, através da leitura de acelerações;
- **Módulo SPI.** É responsável pela transferência de dados entre o microcontrolador e o acelerómetro;
- **Módulo de Gestão de Comissionamento.** É responsável pela gestão do nó na rede;
- **Módulo de Inicialização e Processamento de eventos da OSAL.** É responsável por informar quais são as tarefas a serem executadas pela OSAL;
- **Módulos de Configuração do Nó Sensor.** É responsável pela configuração do módulo CC2538 / CC2592 e do acelerómetro.

3.6.5.1 Módulo do Acelerómetro

O módulo do acelerómetro, desenvolvido pelo grupo ESGR permite a deteção de colisões, cálculo de acelerações e inclinações. Este módulo é capaz de efetuar leitura de acelerações e cálculo dos ângulos.

O acelerómetro pode medir acelerações de ± 2 g, ± 6 g e ± 8 g em 3 eixos. Esses valores são utilizados no parâmetro da escala da equação 3.1. A leitura em 3 eixos permite obter uma leitura em 6 direções. Este dispositivo pode operar em 5 modos de configuração (*low power*), possui interface SPI e I2C para a transferência de dados, com resolução de 12 *bits*. O acelerómetro foi configurado para detetar

acelerações iguais ou superiores a 2 g (valor de *threshold*), no modo *low power* com a frequência de leitura de 20 Hz.

Relativamente ao cálculo de acelerações, a equação 3.1 é responsável essa operação. O valor da aceleração lido em cada eixo (xx, yy e zz) é multiplicado pelo dobro da escala e dividido pela resolução do acelerómetro.

$$Aceleração[g] = Acc[N] * \left(\frac{2 * Escala}{2^{12}} \right) \quad (3.1)$$

Equation 3.1: Equação de cálculo do valor da aceleração em g.

3.6.6 Aplicação do Nó Coordenador

Usando a camada de aplicação da pilha protocolar Z-Stack 3.0.2 da TI será desenvolvido o *software* do nó coordenador. As tarefas a realizar pelo nó coordenador serão: a criação e gestão da rede, conversão dos dados recebidos dos nós em formato JSON e o reencaminhamento dos dados convertidos para o *gateway*.

Os principais módulos da aplicação do nó coordenador são:

- **Módulo Principal da Aplicação.** É responsável pela inicialização e formação da rede, e função de gerir os eventos após a associação do nó sensor;
- **Módulo de Envio de Mensagem para o Gateway.** É responsável pela conversão das mensagens recebidas dos nós em formato JSON e no envio das mensagens convertidas para o *gateway*;
- **Módulo do Perfil da Aplicação.** É responsável por definir o perfil privado da aplicação através do registo do *EndPoint* da aplicação da camada AF do dispositivo;
- **Módulo UART.** É responsável pela inicialização da comunicação via porta série do dispositivo;
- **Módulo de Gestão de Comissionamento.** É responsável pela gestão da rede;
- **Módulo de Inicialização e Processamento de eventos da OSAL.** É responsável por informar quais são as tarefas a serem executadas pela OSAL;
- **Módulos de Configuração do Nó Coordenador.** É responsável pela configuração do módulo CC2538 / CC2592 como nó coordenador.

3.6.7 Aplicação do *Gateway*

Usando *Python 3.7* no sistema operativo *Raspbian Buster* baseado no *Debian 10*, será desenvolvido o *software* do *gateway*. As tarefas a realizar pelo *gateway* serão: reencaminhamento das mensagens recebidas em JSON do nó coordenador para os microserviços *web* desenvolvido pelo grupo ESRG e no armazenamento temporário das mensagens no caso da falta de ligação a internet ou indisponibilidade dos microserviços.

- **Módulo Principal da Aplicação.** É responsável pela gestão e envio das mensagens para os microserviços;
- **Módulo da Base de Dados (PyMongo).** É responsável pelo armazenamento temporário das mensagens;
- **Módulo da UART (PySerial).** É responsável pela inicialização da comunicação via porta série;
- **Módulo JSON.** É responsável pela verificação do formato da mensagem;
- **Módulo Microserviços.** É responsável pela inicialização e configuração dos microserviços;

3.7 Resumo

Neste capítulo foram apresentadas as características mais importantes do sistema a implementar. Posteriormente, foi especificado a arquitetura de *hardware* e *software* utilizado.

Capítulo 4

Desenvolvimento do Sistema

Este capítulo descreve o desenvolvimento do sistema especificado no capítulo anterior, começando com as alterações efetuadas ao *hardware* existente do nó sensor e do coordenador, seguindo-se a implementação do *software* utilizado no sistema.

4.1 *Hardware* Desenvolvido

4.1.1 Nó Sensor

O desenvolvimento do *hardware* do nó sensor foi dividido em duas partes. Primeiramente é feita uma abordagem sobre as alterações efetuadas no sistema de gestão de energia do nó. Posteriormente é explicado as alterações efetuadas nas ligações do módulo CC2538 / C2592 da TI. Todas as alterações efetuadas foram com o objetivo de tornar o nó mais eficiente do ponto de vista do consumo de energia.

4.1.1.1 Sistema de Gestão de Energia

Efetuaram-se alterações na versão anterior do projeto [32], na parte de potência existia um diodo em série com a bateria, dois condensadores em paralelo na entrada do *Low-Dropout Regulator* (LDO) TPS783 e três condensadores em paralelo na saída e um botão *on/off* em série com um *Light-Emitting Diode* (LED). Como o nó é alimentado por bateria, foi retirado o diodo que estava em série com a bateria. Relativamente à remoção dos condensadores na entrada do LDO TPS783, quando o microcontrolador acorda há transientes na carga e este regulador porta-se bem. Quanto à remoção dos condensadores na saída do regulador, como o módulo CC2538 / CC2592 da TI e o acelerómetro LIS331DLH da ST, podem operar a uma tensão de 2,6 V, baixou-se a tensão de saída do regulador de 3 V para 2,6 V, através da troca de reguladores da mesma família TPS783, obteve-se uma menor *Dropout Voltage* quando o

microcontrolador está em *sleep* de modo a perder o menos possível por efeito de Joule, a corrente quiescente baixa e ajuda a minimizar estas perdas.

O botão *on/off* e o LED que sinaliza o *on/off*, foram removidos, o nó estará sempre alimentado, o processo de poupança de bateria será feita pelos modos *low power* do microcontrolador. Para ligar ou desligar o nó, esse processo passa a ser efetuado através de um *jumper* em série com a bateria.

O esquemático do sistema de gestão de energia do nó é apresentada na Figura 4.1.

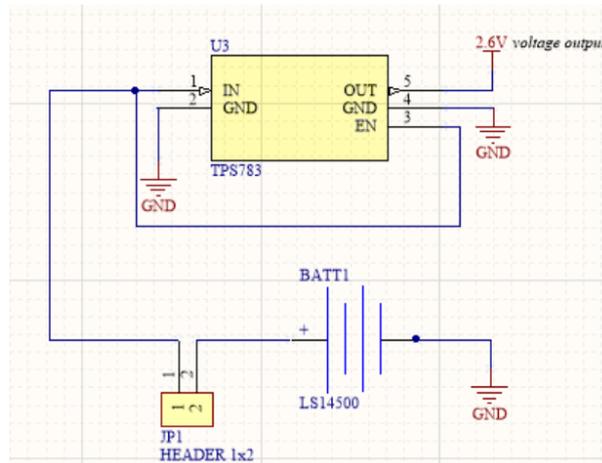


Figura 4.1: Esquemático do sistema de gestão de energia.

4.1.1.2 Módulo CC2538 / CC2592 da *Texas Instruments*

Existia a ligação do pino de *reset* a um interruptor de pressão que, quando pressionado, reiniciava o microcontrolador. Este interruptor foi eliminado, o microcontrolador passa a fazer o *reset* por *software* depois de um número programado de tentativas ao se ligar à rede.

Relativamente a ligação do microcontrolador com o acelerómetro, foi desabilitada fisicamente a interrupção externa 2 que estava conectada ao pino PD1, não era utilizada e adiciona consumo ao sistema, desse modo chegou-se aos valores esperados do consumo de energia do módulo quando se encontra no modo PM3 (*Power Down Mode 3*). Removeu-se também o condensador desacoplamento da alimentação do módulo CC2538 / CC2592, depois de alguns testes do subsistema. Para além disso, adicionou-se ao esquemático resistências de 10 kΩ NC (*Not Connected*) nos pinos que fazem a interface com o acelerómetro, de maneira que em trabalhos futuros se possa estudar a influência desse componente no consumo de energia quando o módulo estiver no modo *deep sleep*.

A Figura 4.2 apresenta as ligações do módulo CC2538 / CC2592 às interfaces externas.

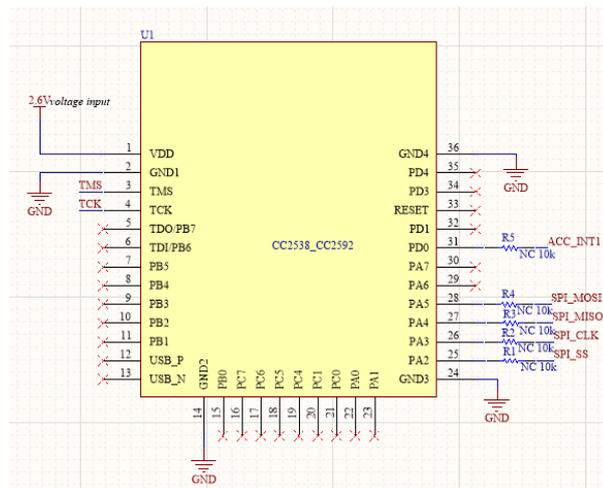


Figura 4.2: Esquemático do módulo CC2538 / CC2592.

4.1.1.3 PCB Desenvolvida

Para a implementação do *hardware* em questão foi utilizada a ferramenta CAD *Altium Designer* para desenvolver a placa de circuito impresso (PCB). No *design* do nó, foi levado em conta em primeiro lugar as dimensões disponíveis nas caixas de proteção. O esquemático desta PCB pode ser consultado no anexo B. Na Figura 4.3 pode ser visualizado em 3D a PCB do nó. O nó desenvolvido, tem as seguintes características:

- Bateria SAFT LS17500 - Para fornecer energia para nó;
- Regulador de tensão TPS783 - Para reduzir a tensão de 3.6 V para 2.6 V
- Conector JTAG - Para programação e *debug* do SoC;
- Acelerômetro MEMS LIS331DLH - Para medir a aceleração;
- Módulo CC2538 / CC2592 - Para o processamento, armazenamento, envio ou recepção de dados.

A PCB assembled do primeiro protótipo do nó com as respectivas alterações é apresentada na Figura 4.4.

Numa fase inicial, foi assembled um primeiro protótipo. Com o protótipo foi possível estudar e/ou validar estratégias para tornar o nó mais eficiente possível. Na Figura 4.5 é pode ser visualizado em 3D o primeiro protótipo do PCB.

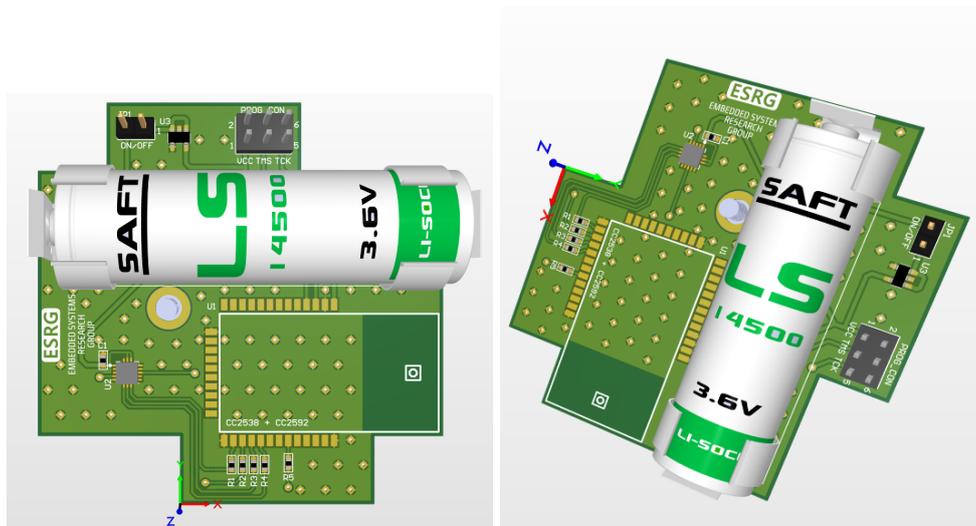


Figura 4.3: Visualizações em 3D do PCB.

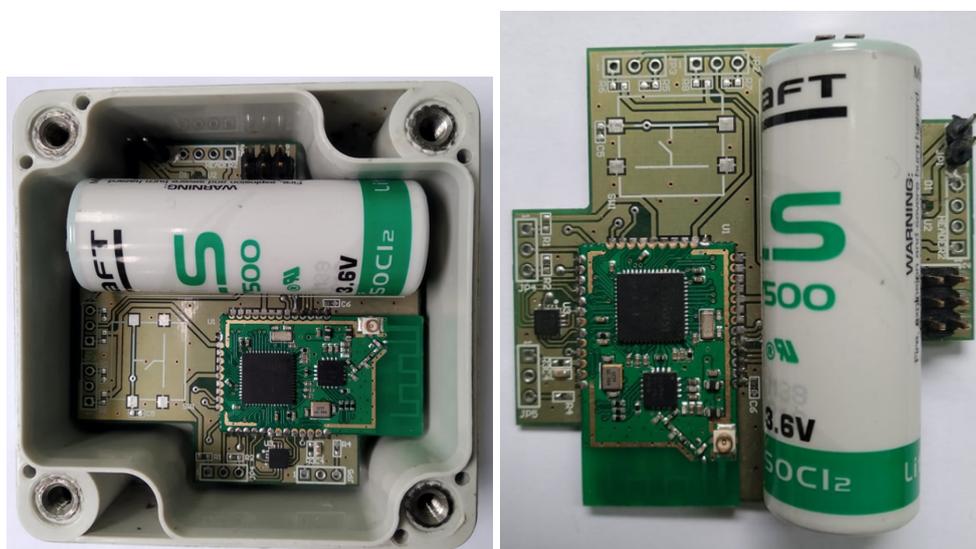


Figura 4.4: Visualização da caixa Hammond com IP66 e do PCB assembled.

4.1.2 Nó Coordenador

Na versão anterior do projeto [32], o *hardware* do nó coordenador consistia apenas no módulo CC2538 / CC2592 com as interfaces JTAG e UART, e a alimentação era feita através de uma fonte de tensão. Para se poderem usar esses módulos conectados a um *gateway*, foi reformulado o *hardware* do nó coordenador. Esta PCB, foi desenvolvida com novas regras e *template Altium* do grupo ESRG da Universidade do Minho, inspirado nas placas de desenvolvimento USB *dongle*. O esquemático desta PCB pode ser consultado no anexo B. Na Figura 4.6 pode ser visualizado em 3D a PCB do nó coordenador. O nó coordenador desenvolvido, tem as seguintes características:

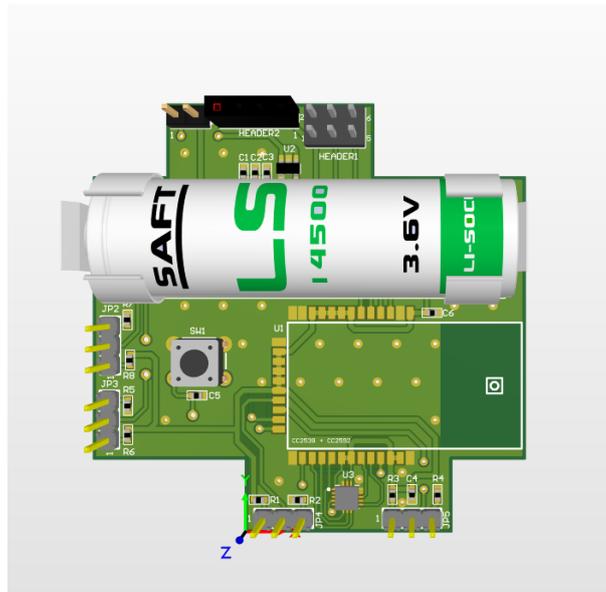


Figura 4.5: Visualização em 3D do PCB do primeiro protótipo.

- Conector USB - Para fornecer energia ao nó, aproveitando o facto de estar ligado a um *gateway* por *serial port*;
- Circuito integrado *Transient Voltage Suppressor* (TVS) SP0503BAHTG - Para fazer a protecção do módulo contra transientes e picos de corrente que podem chegar pela alimentação do circuito;
- Regulador de tensão MIC5504 - Para reduzir a tensão da porta USB de 5 V para 3.3 V;
- Circuito integrado FT230XS-R da *FTDI*- Para fazer o interface entre *serial*/UART e porta USB;
- Conector JTAG - Para programação e *debug* do módulo;
- Módulo CC2538 / CC2592 - Para o processamento, armazenamento, envio ou receção de dados.

Na Figura a seguinte (4.7), pode se observar a PCB assemblada do nó coordenador com todos os componentes.

4.2 Software Desenvolvido

Nesta secção será apresentado o *software* desenvolvido. Primeiramente serão apresentadas as principais componentes da WSN e os dispositivos que a compõe. Este *software* está dividido em duas tarefas: AP *Task* da rede e ED *Task*, desenvolvidos utilizando a *Z-Stack* 3.0.2. Posteriormente é feita a descrição da criação de uma interface entre o *gateway* e os *microserviços web*.

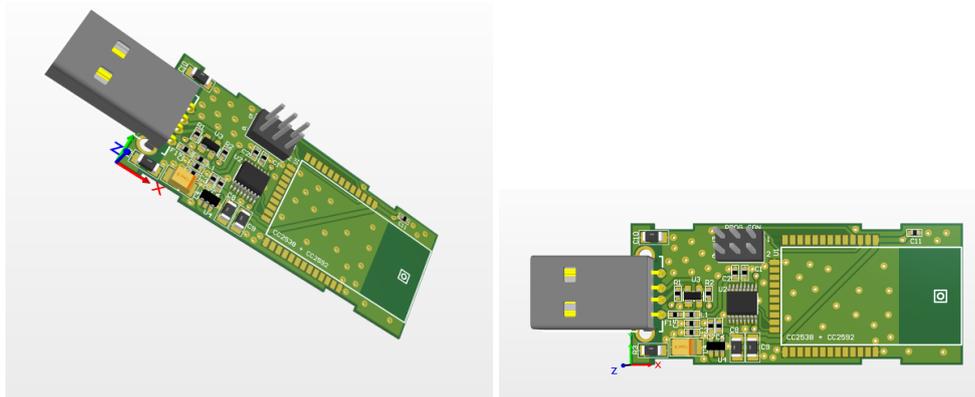


Figura 4.6: Visualizações em 3D do PCB.



Figura 4.7: Visualização da caixa e do PCB assembled.

4.2.1 Access Point Task

O *software* desenvolvido para o nó coordenador será responsável pela criação e gestão da WSN, e pelo envio dos dados recebidos dos nós para um *gateway*. Na Figura 4.8 é possível visualizar o algoritmo de funcionamento do coordenador. Após do coordenador ser ligado, são efetuadas todas as inicializações necessárias para o seu correto funcionamento, desde os *drivers* até as miniaaplicações da *Stack*. Depois deste processo o coordenador está habilitado para associar nós a sua rede.

Quando o processo de associação e configuração do nó a rede for efetuado com sucesso, o coordenador fica à espera de dois tipos de mensagens do nó de KA e/ou de ocorrência de uma colisão. De seguida, converte a mensagem em formato JSON e envia pela porta série a um *gateway*.

4.2.1.1 Módulos de Envio de Mensagem para o *Gateway*

Os dados recebidos pelo nó coordenador são processados pelos módulos que constituem o envio de mensagem em formato JSON para o *gateway* via porta série. A mensagem convertida é composta por

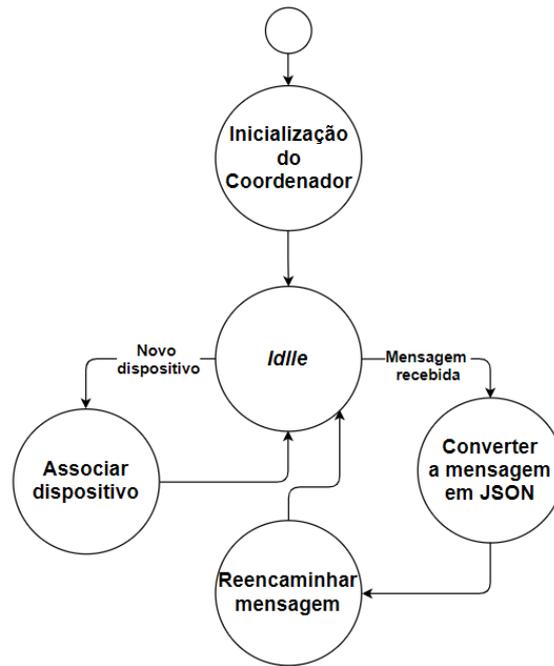


Figura 4.8: Diagrama de funcionamento do nó coordenador.

informações relativas à rede e por informações relativas à aplicação. No código apresentado na listagem 4.2.1.1 e no código apresentado na listagem 4.1 é possível visualizar respectivamente informações relativas à rede e informações relativas à aplicação.

```

// Start of the JSON Message
sprintf ( buff, "\"NTW\": {\"");
puts ( buff );
sprintf ( buff, "\"WPANID\": \"0x%X\",", ZDAPP_CONFIG_PAN_ID);
puts ( buff );
sprintf ( buff, "\"GROUPID\": \"%d\",", pkt->groupId);
puts ( buff );
sprintf ( buff, "\"CLUSTERID\": \"%d\",", pkt->clusterId );
puts ( buff );
sprintf ( buff, "\"GATEWAYMAC\": \"%s\",", ntw_MacAddress_toString (macaddr, mac_str));
puts ( buff );
sprintf ( buff, "\"ENDPOINT\": \"%d\",", pkt->endPoint);
puts ( buff );
sprintf ( buff, "\"BROADCAST\": \"%d\",", pkt->wasBroadcast);
puts ( buff );
sprintf ( buff, "\"LINKQUALITY\": \"%d\",", pkt->LinkQuality );
puts ( buff );
sprintf ( buff, "\"CORRELATION\": \"%d\",", pkt->correlation );
puts ( buff );
sprintf ( buff, "\"RSSI \": \"%d\",", pkt->rsi);

```

```

puts ( buff );
sprintf ( buff , "\SECURITY\":"%d\",", pkt->SecurityUse );
puts ( buff );
sprintf ( buff , "\SEQNUMBER\":"%d\"", pkt->nwkSeqNum);
puts ( buff );
//End the JSON Message
sprintf ( buff , "}" );
puts ( buff );

```

Listagem 4.1: Informações relativas à rede.

```

//0. Clean up the buff
puts ("\APP\ : {");
// Buff the MAC Address & Sampling Rate & NodeType
sprintf ( buff , "\DEVICEMAC\":"%s\",",
ntw_MacAddress_toString ( appPacket->macaddr, mac);
puts ( buff );
// Buff the Sampling Rate
sprintf ( buff , "\SAMPLING_RATE\":"%lu\",", appPacket->samplingRate_sec );
puts ( buff );
// Buff the NodeType
sprintf ( buff , "\DEVICETYPE\":"%d\"", appPacket->nodeType);
puts ( buff );
// Buff the acceleration
if ( appPacket->pck_id & ACCELERATION ){
sprintf ( buff , "\ACCELERATION\":{"X\":"%f \",\"Y\":"%f \",\"Z\":"%f \",
"\TOTAL\":"%u \",\"INDEX\":"%u \"}",
appPacket->acc.acc [0],
appPacket->acc.acc [1],
appPacket->acc.acc [2],
appPacket->acc.sampleCounter ,
appPacket->acc.sampleIndex
);
puts ( buff );
}
//End the JSON Message
puts ("}");

```

Listagem 4.2: Informações relativas à aplicação.

4.2.2 End Device Task

Tal como foi visto na subsecção 4.2.1, o nó sensor precisa também para o seu correto funcionamento, efetuar todas as inicializações necessárias, desde os *drivers* até as miniaaplicações da *Stack*. Depois estar

associado a uma rede, o nó entra no modo *sleep* (PM2), para cumprir com o requisito do baixo consumo de energia. Somente acorda para o envio periódico de KA ou quando deteta a ocorrência de uma colisão. Na figura 4.9 é possível visualizar o algoritmo de funcionamento do nó sensor.

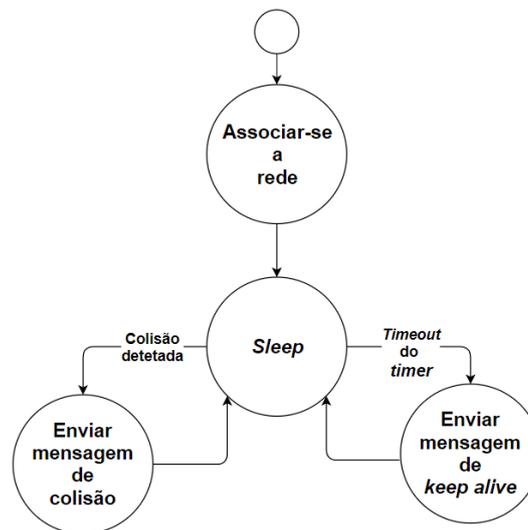


Figura 4.9: Diagrama de funcionamento do nó sensor.

4.2.2.1 Módulo do Acelerómetro

O módulo acelerómetro é essencial no sistema. Isto deve-se ao fato de ser o responsável pela deteção de colisão, o que significa se capturar uma aceleração superior ao valor limite estipulado (*threshold*), o acelerómetro acorda o SoC através de uma interrupção externa, configura-se no modo normal e faz a leitura dos valores das acelerações nos três eixos, e depois disso, através do módulo CC2538 / CC2592 do nó sensor são enviadas em g como mensagem de colisão para o nó coordenador. O acelerómetro volta ao modo *low power* e SoC no modo PM2. Na Figura 4.10 é possível visualizar o algoritmo de funcionamento do módulo acelerómetro.

4.2.3 Software Desenvolvido para o Gateway

Quando desenvolve-se uma WSN, uma necessidade presente na grande maioria das redes sensores sem fios é a implementação de uma interface entre a WSN e os serviços *web*, ou seja, uma maneira para que os responsáveis do sistema possam ter acesso remoto aos dados da WSN. Como referido na subsecção 3.6.7, o *software* do *gateway* foi desenvolvido usando o *Python 3.7* no sistema operativo *Raspbian*

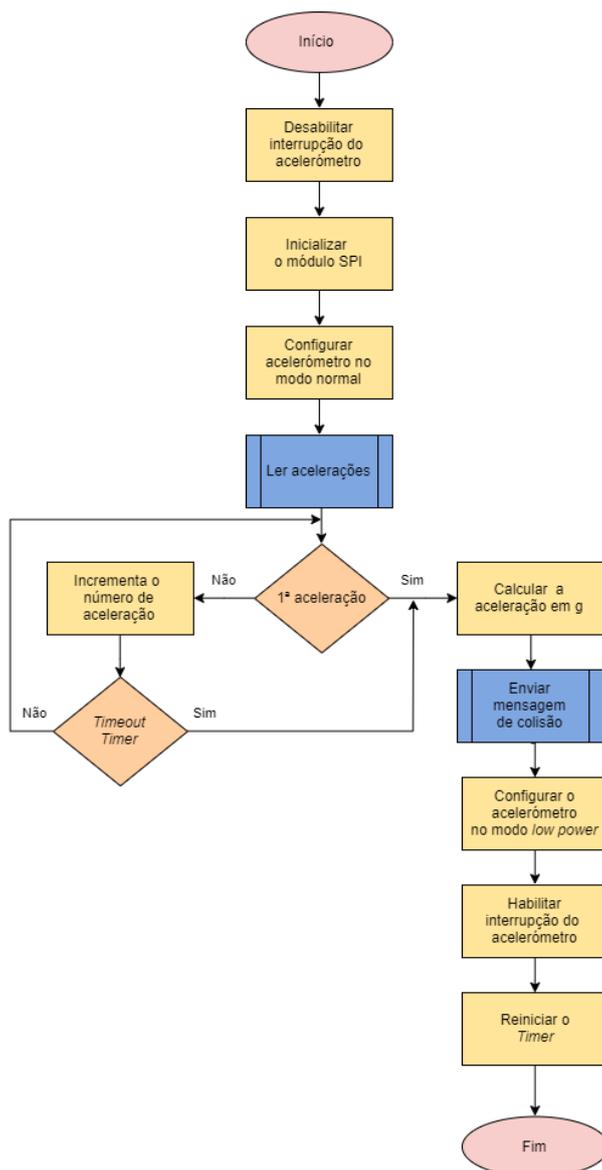


Figura 4.10: Diagrama de funcionamento do módulo acelerómetro.

Buster. O *software* oferece ao sistema a possibilidade de gerir e enviar as mensagens da WSN para os microserviços *web*. Nesta subsecção será descrita a implementação dos principais módulos do *software*. Quando liga-se o *gateway*, inicializa automaticamente o *software* e a base dados MongoDB com o sistema operativo. Inicializa-se a porta série, a interface *Python* (PyMongo) para manipulação da base dados e configura-se os microserviços. O *gateway* verifica se tem dados pendentes na base de dados e enviá-las para os microserviços através do protocolo TCP/IP. Por último, fica à espera de mensagens em formato JSON do nó coordenador. Na Figura 4.11 é possível visualizar o algoritmo do funcionamento do *software* do *gateway*.

As mensagens não enviadas para os microserviços, são armazenadas numa base dados. Quando o

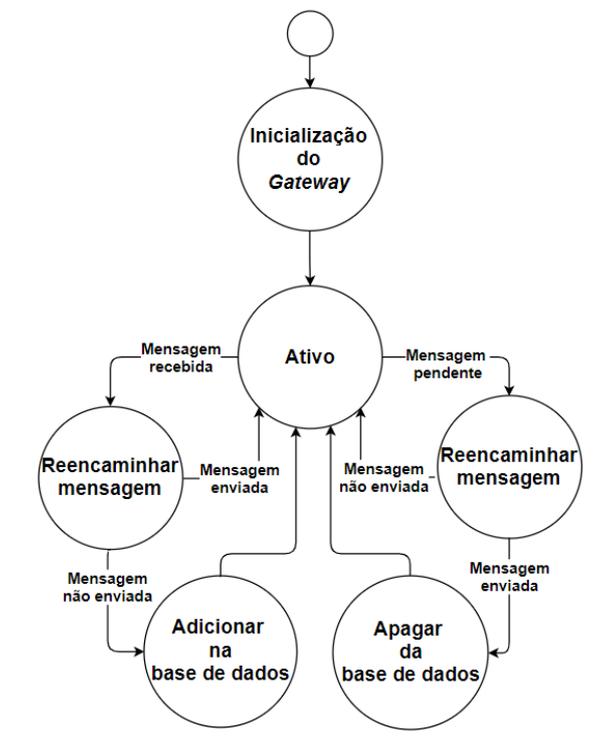


Figura 4.11: Diagrama de funcionamento do *software* do *gateway*.

gateway restabelece ligação com os *microserviços*, exporta-se as mensagens da base de dados em formato *Binary JavaScript Object Notation* (BJSON) e convertidas em JSON, afim de serem enviadas para os *microserviços*. Quando enviadas, são apagadas da base de dados.

4.3 Resumo

Este capítulo teve como objetivo, numa primeira fase, apresentar o *hardware* desenvolvido. Em seguida foram apresentadas as principais funcionalidades do sistema desenvolvidas em *software*.

Capítulo 5

Testes e Resultados

Este capítulo descreve os testes efetuados na validação do sistema desenvolvido. São apresentados os resultados de *performance* do sistema, relativamente ao consumo energético, fiabilidade e robustez dos subsistema, e os indicadores de Qualidade de Serviços (QoS). Para além disso, também é feita uma descrição do cenário de testes, especificando os parâmetros e as ferramentas utilizadas.

5.1 Cenários de Testes

Para validação do *hardware* foram utilizados os seguintes instrumentos:

- Fonte de alimentação da *Keysight – Dual Output DC Power Supply E348A* (0 – 8 V, 5 A / 0 – 20 V, 2,5 A);
- Osciloscópio da *Keysight – Digital Storage Oscilloscope InfiniiVision DSOX2002A* (70 Mhz, 2 GSa/s);
- o Multímetro da *Keysight- Digit Multimeter 34410A*;

Para validação do *firmware* foram utilizadas as seguintes ferramentas:

- Ambiente de desenvolvimento integrado (IDE - *Integreted Develpment Environment*) da *IAR Systems* - *IAR Embedded Workbench Arm*;
- Emulador JTAG / USB da TI - XDS100 V3.0
- *Serial port terminal*- CoolTerm 1.6.0.708
- Osciloscópio da *Keysight – Digital Storage Oscillospe InfiniiVision DSOX2002A* (70 Mhz, 2 GSa/s);

Para validação do *software* foram utilizadas as ferramentas:

- Sistema operativo (OS - *Operating System*) *Raspbian* da *RaspBerry Pi Foundation* - *Raspbian Buster Lite*;
- Sistema de Gestão de Base Dados (SGBD) MongoDB;
- Python 3;
- *Packages* PySerial e PyMongo;
- Microserviços IoTClient.

5.2 Testes Desenvolvidos

A metodologia de testes do sistema é dividida em duas partes:

- Testes isolados a cada componente do sistema, verificando o consumo de energia e os indicadores de QoS (RSSI e LQI). Além disso, analisar também a fiabilidade e robustez de cada componente.
- Testes do demonstrador, consiste na integração de todos componentes do sistema, com o objetivo de compatibilizar a WSN desenvolvida com os microserviços desenvolvidos pelo grupo ESRG. De maneira a validar o desempenho do sistema completo, verificando a fiabilidade e o tempo de resposta.

5.2.1 Teste ao Protocolo Porta Série

Para validação da comunicação via porta série utilizou-se o *software* Terminal *CoolTerm* v1.6.0. Foi utilizado o *firmware* que configura o nó sensor para o envio periódico de KA e mensagens de colisão. O coordenador (baseado em USB *dongle*) foi conectado ao PC com um *firmware* responsável por converter em formato JSON as mensagens recebidas do nó sensor e enviar via porta série. A comunicação série do nó coordenador configurou-se com o *baud rate* de 115200 bps, 8 *bit* de dados, sem paridade, com 1 *stop bit* e sem controlo de fluxo (Figura 5.1).

A Figura 5.2 é possível observar que o nó coordenador recebe as mensagens de KA e de colisão do nó sensor, converte-las em JSON e reencaminha pela porta série.

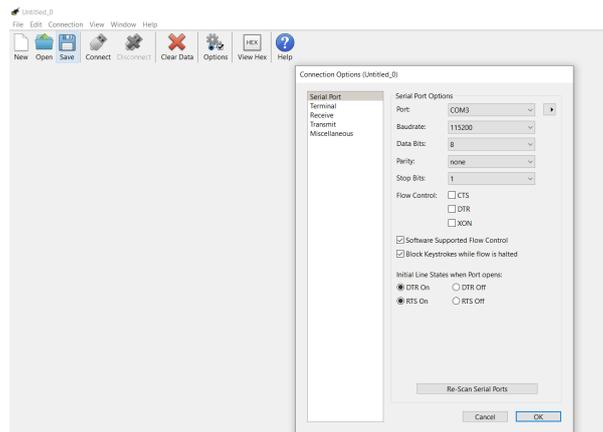


Figura 5.1: Configuração do *software* CoolTerm

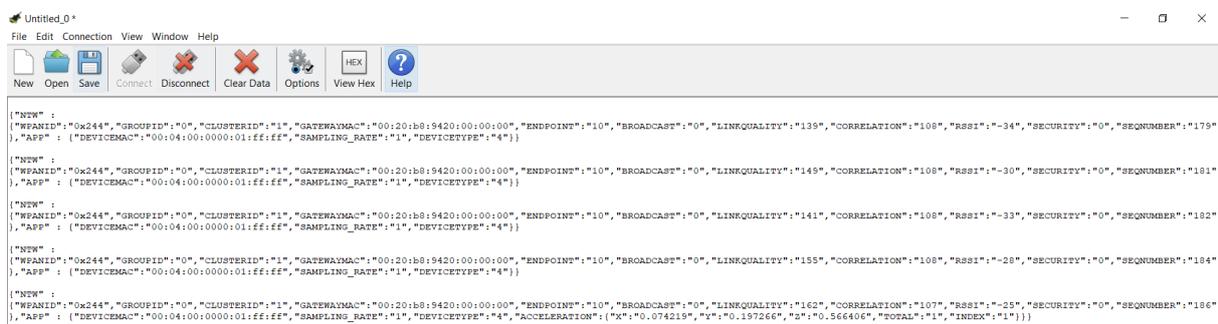


Figura 5.2: Mensagem de colisão e de KA reencaminhadas pelo coordenador via porta série.

5.2.2 Obtenção de Indicadores de QoS (RSSI e PER)

Para obtenção de indicadores de Qualidade de Serviços (QoS) foi utilizado o RSSI e o *Packet Error Rate* (PER) para as medições de alcance. Para a realização do teste, colocou-se o nó coordenador junto a uma das janelas do laboratório ESRG, enquanto que o nó sensor foi colocado no exterior, com barreiras físicas colocadas a 95,04 metros de distância entre si. Estas distâncias foram obtidas por intermédio da aplicação *Google Maps*. Os nós foram configurados com a potência de transmissão de 22 dBm e no canal 25. O nó coordenador foi ligado ao *gateway* e os dados foram recolhidos através de uma *script* em *Python 3* para leitura de dados via porta série. Na Figura 5.3 é possível visualizar a posição global dos dispositivos para a realização dos testes.

Foi realizado um único teste a uma distância de 95,04 metros entre os nós com barreira físicas. Assim sendo, o valor da média do RSSI é de -89,86 dBm e o PER é de 36,17 %. Não utilizou-se prato de alumínio na base da antena do nó coordenador.

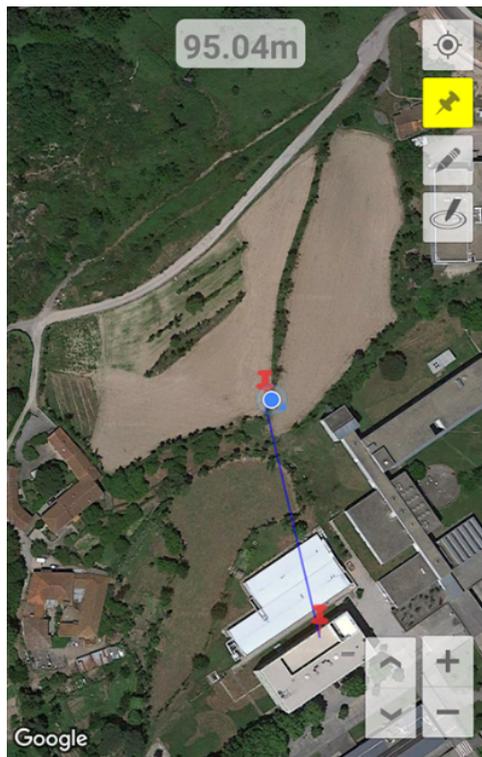


Figura 5.3: Distância entre os dispositivos para as medições do alcance.

5.2.3 Teste ao *Gateway*

As mensagens recebidas através da porta série são processadas pela interface que implementa o *gateway* do sistema. Foi realizado teste de validação do *gateway*, através do *software* de emulação PuTTY (Figura 5.4). As mensagens recebidas são resultantes dos reencaminhamento de mensagens no formato JSON pelo nó coordenador dos dados provenientes dos nós sensoriais.

De modo a satisfazer os requisitos tolerância a falhas, o *gateway* pode operar independentemente de ter acesso à internet ou aos microserviços *web*, armazena os dados temporariamente numa base de dados local baseada em *MongoDB* até que os serviços seja restabelecidos. Quanto à auto-reconfiguração, esta interface inicializa-se com o sistema operativo e continua a funcionar no caso de desconectar-se momentaneamente o nó coordenador. Na Figura 5.5 e na Figura 5.6 podem ser visualizados respetivamente o auto-arranque do *gateway* e a base dados *MongoDB* ativa no no sistema.

Quanto as mensagens recebidas pelo *gateway*, são enviadas para os microserviços ArchNET, após o envio o *gateway* recebe dos microserviços uma mensagem de confirmação "inserido". Para além disso, o sistema verifica se tem mensagens pendentes na base dados. Na Figura 5.7 é possível verificar as mensagens enviadas para os microserviços.

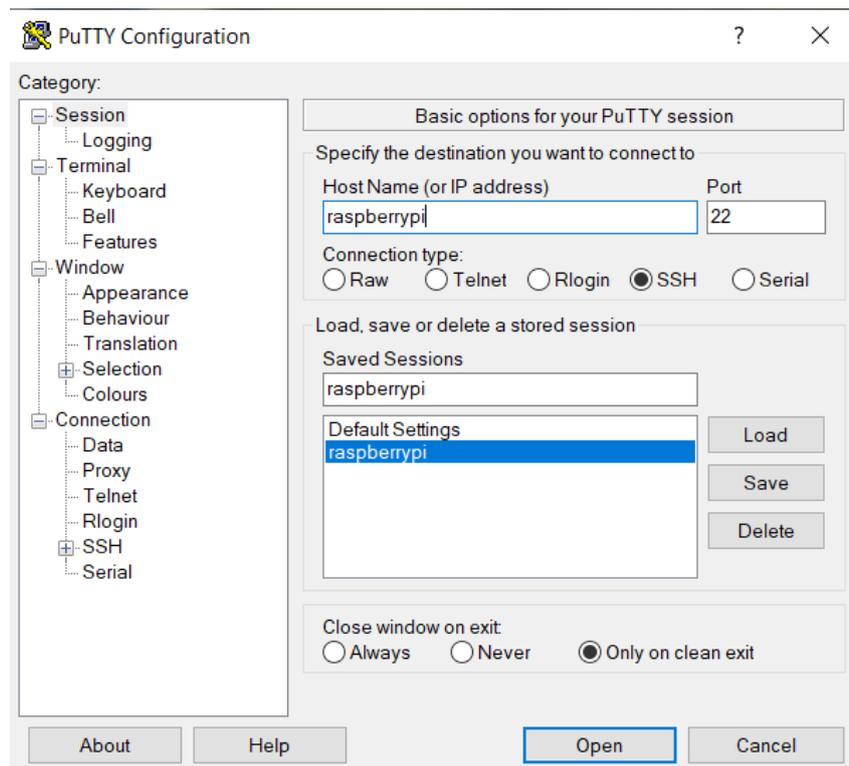


Figura 5.4: Configuração do *software* PuTTY.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ sudo ps -ax | grep SustIMS
406 ?        Rl        1:24 /usr/bin/python3 /home/pi/Documents/SustIMS/Port_COM.py
827 pts/0    S+       0:00 grep --color=auto SustIMS
pi@raspberrypi:~$

```

Figura 5.5: Inicialização automática da *script* que implementa a interface *gateway*.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ sudo service mongodb status
● mongodb.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-10-23 16:22:27 WEST; 40min ago
     Docs: man:mongod(1)
    Main PID: 310 (mongod)
      Tasks: 13 (limit: 2200)
     Memory: 77.3M
    CGroup: /system.slice/mongodb.service
            └─310 /usr/bin/mongod --config /etc/mongodb.conf

out 23 16:22:27 raspberrypi systemd[1]: Started An object/document-oriented database.
out 23 16:22:29 raspberrypi mongod[310]: all output going to: /var/log/mongodb/mongod.log
pi@raspberrypi:~$

```

Figura 5.6: Estado da base de dados *MongoDB*.

5.2.4 Teste de Integração do Sistema

De forma a validar os diferentes módulos e dispositivos que constituem o sistema com os *microserviços web* desenvolvidos pelo grupo de investigação ESRG, foi realizado um teste de integração durante um período de 24 horas no laboratório do ESRG. Na Figura 5.8 é ilustrado a arquitetura do cenário de testes utilizado.

```

pi@raspberrypi: ~/Documents/SustIMS
pi@raspberrypi:~/Documents/SustIMS $ python Port_COM.py
Local database size: 0
{'NTW': {'WPANID': '0x244', 'GROUPID': '0', 'CLUSTERID': '1', 'GATEWAYMAC': '00:20:b8:9420:00:00:00', 'ENDPOINT': '10', 'BROADCAST': '0', 'LINKQUALITY': '107', 'CORRELATION': '107', 'RSSI': '-46', 'SECURITY': '0', 'SEQNUMBER': '3'}, 'APP': {'DEVICEMAC': '00:04:00:0000:01:ff:ff', 'SAMPLING_RATE': '1', 'DEVICETYPE': '4'}}
0
Inserido
{'NTW': {'WPANID': '0x244', 'GROUPID': '0', 'CLUSTERID': '1', 'GATEWAYMAC': '00:20:b8:9420:00:00:00', 'ENDPOINT': '10', 'BROADCAST': '0', 'LINKQUALITY': '107', 'CORRELATION': '108', 'RSSI': '-46', 'SECURITY': '0', 'SEQNUMBER': '5'}, 'APP': {'DEVICEMAC': '00:04:00:0000:01:ff:ff', 'SAMPLING_RATE': '1', 'DEVICETYPE': '4'}}
0
Inserido
{'NTW': {'WPANID': '0x244', 'GROUPID': '0', 'CLUSTERID': '1', 'GATEWAYMAC': '00:20:b8:9420:00:00:00', 'ENDPOINT': '10', 'BROADCAST': '0', 'LINKQUALITY': '105', 'CORRELATION': '107', 'RSSI': '-47', 'SECURITY': '0', 'SEQNUMBER': '9'}, 'APP': {'DEVICEMAC': '00:04:00:0000:01:ff:ff', 'SAMPLING_RATE': '1', 'DEVICETYPE': '4'}}
0
Inserido
{'NTW': {'WPANID': '0x244', 'GROUPID': '0', 'CLUSTERID': '1', 'GATEWAYMAC': '00:20:

```

Figura 5.7: mensagens enviadas pelo gateway para os micros serviços.

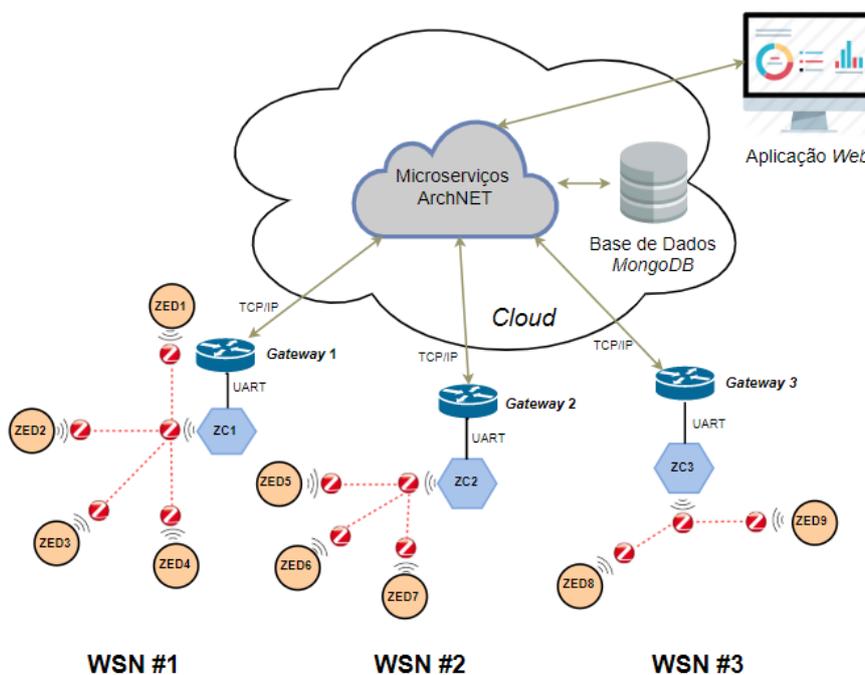


Figura 5.8: Arquitetura geral, teste de integração.

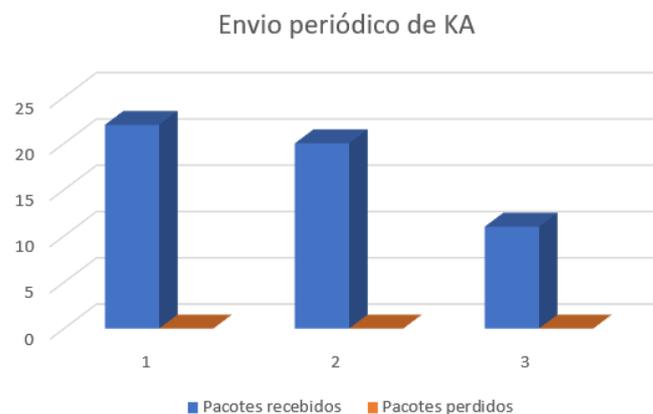
O cenário de testes é constituído por três WSN com diferentes PANIDs. Cada WSN é por um gateway, por um nó coordenador (ZC) e por dois ou mais nó sensoriais (ZEDs). Os nós sensoriais alguns forma ligados à rede elétrica e outros alimentados por bateria. Quanto ao envio periódico de KA foram definidas diferentes taxas de amostragem. Na Tabela 5.1 é possível visualizar com detalhes os módulos e dispositivos que constituem o cenário de testes.

Na Figura 5.9 são apresentados os valores dos obtidos pelos micros serviços ArchNET, através do envio

Tabela 5.1: Especificação dos dispositivos utilizados no teste de integração

WSN	Dispositivo	Envio periódico KA	Módulo	Detalhes
1	ZED1	4 h	CC2538 / CC2592	Ligado à rede elétrica
1	ZED2	4 h	CC2538 / CC2592	Bateria SAFT LS17500
1	ZED3	3 h	CC2538 / CC2592	Ligado à rede elétrica
1	ZED4	3 h	CC2538 / CC2592	Ligado à rede elétrica
1	ZC1	-	CC2538 / CC2592	Ligado à rede elétrica
1	Gateway 1	-	Raspberry PI 3 <i>model/B</i>	Ligado à rede elétrica
2	ZED5	2 h	CC2538 / CC2592	Ligado à rede elétrica
2	ZED6	3 h	CC2538 / CC2592	Ligado à rede elétrica
2	ZED7	4 h	CC2538 / CC2592	Ligado à rede elétrica
2	ZC2	-	CC2538 / CC2592	Ligado à rede elétrica
2	Gateway 2	-	Raspberry PI <i>model/B</i>	Ligado à rede elétrica
3	ZED8	4 h	CC2538 / CC2592	Bateria SAFT LS17500
3	ZED9	3 h	CC2538 / CC2592	Ligado à rede elétrica
3	ZC3	-	CC2538 / CC2592	Ligado à rede elétrica
3	Gateway 3	-	Raspberry PI <i>model/B</i>	Ligado à rede elétrica

periódico de *keep alive* dos nós sensoriais. Ao longo de 24 horas o sistema teve resultados esperados, com o PER de 0%. É de saliente que o teste foi realizado no interior do laboratório do ESRG, os nós sensoriais foram colocados a uma distância máxima de 30 cm, sem barreiras físicas, com a exceção dos nós que operam com baterias, que foram colocados a uma distância de aproximadamente 4 metros.

**Figura 5.9:** Valores obtidos pelo microserviços ArchNET.

5.3 Medição de Consumo

O baixo consumo energético do nó sensor, é um dos principais requisitos do sistema desenvolvido. Foram efetuadas medições dos nós no estado de *Deep Sleep*, no estado de KA. Para além disso, também efetuou-se a estimativa do tempo de vida do nó sensor.

5.3.1 Consumo Médio de Corrente no Estado *Deep Sleep*

As medições do consumo de energia do nó sensor foram realizadas no laboratório do ESRG. A distância entre o nó sensor e o coordenador foi 4,4 metros. Os instrumentos e o esquema de ligação das medições é possível observar na Figura 5.10.

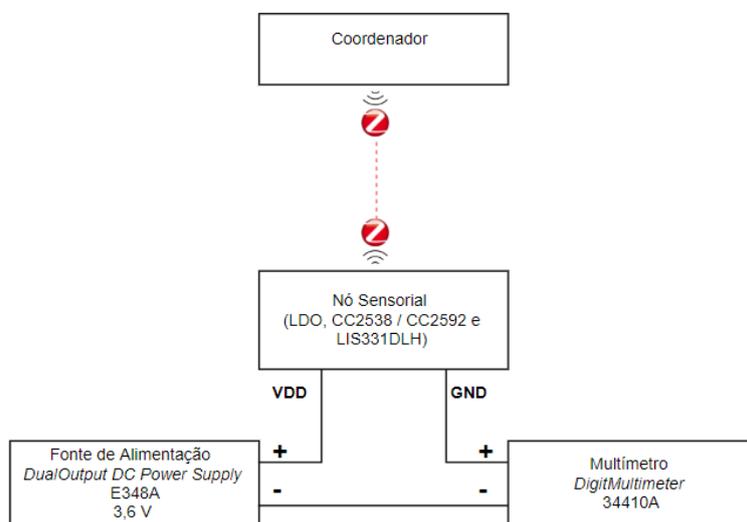


Figura 5.10: Esquema de ligação para medição de consumo em *deep sleep*.

Quando o nó sensor associou-se ao nó coordenador, iniciou-se a medição do consumo energético e termina depois de 1 hora. O nó sensor foi configurado no modo PM3 e o acelerómetro no modo *power down*. Utilizou-se o modo *power down* para evitar a influência do consumo de energia existente nos outros modos *low power* do acelerómetro. Os valores mínimo, médio e máximo da corrente elétrica consumida, e o número das amostras podem ser observados na Tabela 5.2.

Comparativamente às versões anteriores, o nó sensor apresenta um consumo inferior, como ilustra a Tabela 5.3. É de salientar que, não foi possível configurar o sistema da versão 2019 [32] no modo PM3.

Tabela 5.2: Consumo de corrente elétrica no estado de *deep sleep* (PM3) com acelerómetro em *power down*.

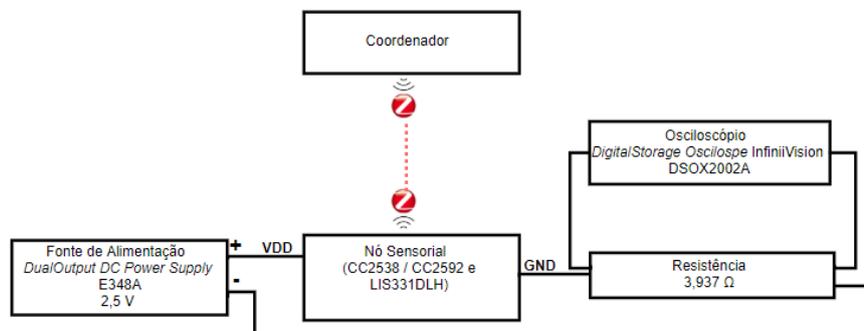
Consumo de corrente elétrica do nó em <i>deep sleep</i>	
Valor mínimo	0,359 μ A
Valor médio	1,406 μ A
Valor máximo	4,234 μ A
Número de amostras	38.297

Tabela 5.3: Consumo de corrente elétrica do nó sensor em *deep sleep* (PM3).

Valor máximo da corrente do nó em <i>deep sleep</i>	
Versão 2014 [8]	130 μ A
Versão 2018 [22]	30 μ A
Versão 2019 [32]	-
Versão atual	4,2 μ A

5.3.2 Consumo Médio de Corrente Durante o Envio do *Keep Alive*

Para informar o funcionamento do nó sensor, são enviadas mensagens periódicas de *keep alive* a cada *timeout* do *Timer*. Para a medição do consumo energético durante o envio do KA foram utilizados a fonte de alimentação, osciloscópio e uma resistência, ilustrada na figura 5.11, que permitiram aferir o consumo do estado KA.

**Figura 5.11:** Esquema de ligação para medição de consumo em KA.

Foi criado um *firmware* de testes para o nó sensor que envia mensagem de KA via OTA com intervalos de tempo entre as tramas de 1 segundo. O nó sensor foi configurado no modo PM2, a potência de transmissão utilizada foi de 22 dBm. Utilizou-se a potência de transmissão de 22 dBm por ser o maior valor disponível para esse módulo e por ser o pior caso em termos de consumo de energia. Para o teste em modo *keep alive* não utilizou-se o LDO no sensor, para evitar a influência do componente no consumo

de energia. O nó sensor foi alimentado com uma tensão 2,5 V, colocou-se uma resistência de 3,937 Ω em série com o o circuito e a queda de tensão aos terminais da mesma foi registada pelo osciloscópio Figura 5.11. Para obter o consumo do KA, foi o utilizado o ficheiro de saída do osciloscópio com os valores das tensões, com esses valores foram calculadas as correntes de cada estado do KA, o tempo de amostragem utilizado foi de 10 ms. O gráfico da corrente elétrica calculada no estado de KA pode ser observado na Figura 5.12.

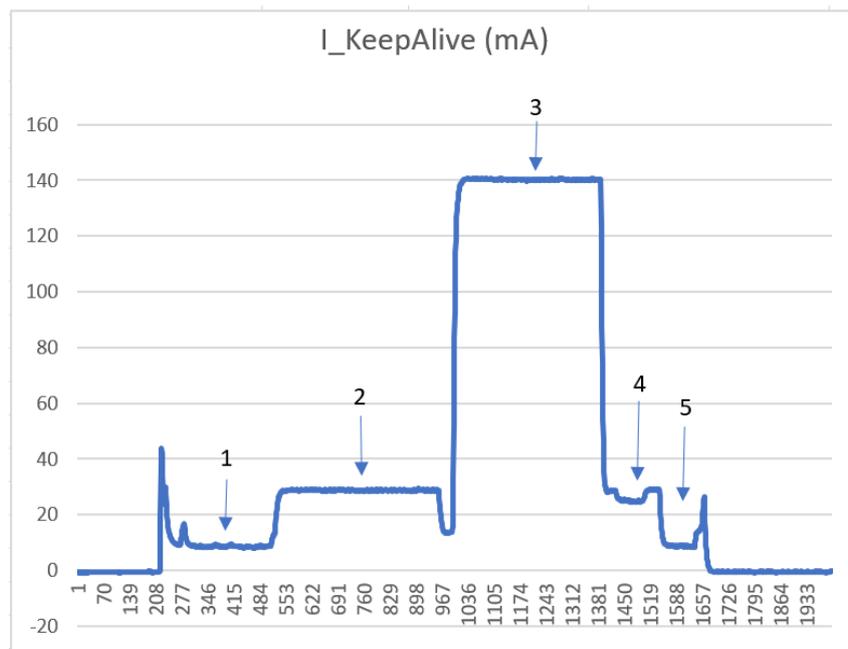


Figura 5.12: Gráfico de corrente elétrica calculada no estado de KA.

Os cálculos para os valores médios de corrente elétrica, para os diferentes modos de operação podem ser observados na Tabela 5.4.

Tabela 5.4: Consumo médio de corrente elétrica em modo KA.

Intervalo	Descrição	Corrente (mA)	Tempo (ms)
1	<i>Idle</i>	24,89	1,465
2	Pre	27,29	2,405
3	Tx	136,83	2,01
4	Pre	26,8	0,715
5	<i>Idle</i>	19,51	0,67

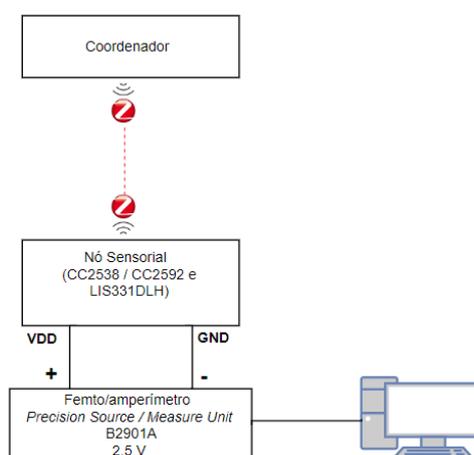
Comparativamente às versões anteriores, o nó sensor apresenta um consumo superior ao sistema da versão 2014 [8] durante a transmissão da mensagem e que apresenta um consumo inferior em relação as versões [22, 32] do sistema, como ilustra a Tabela 5.5.

Tabela 5.5: Consumo de energia durante a transmissão da mensagem.

Sistema	Energia consumida
Versão 2014 [8]	0,306 mJ
Versão 2018 [22]	1,440 mJ
Versão 2019 [32]	1,229 mJ
Versão atual	1,064 mJ

5.3.3 Consumo Médio de Corrente do Acelerómetro em Diferentes Modos

Para a realização das medições de consumo de corrente do acelerómetro em diferentes modos de operação *low power* (5 Hz, 10 Hz e 20 Hz) recorreu-se ao *femto*-amperímetro. Os instrumentos e esquemas de ligação podem ser observados na figura 5.13.

**Figura 5.13:** Esquema de ligação para medição de diferentes modos de consumo *low power* do acelerómetro.

De forma a obter-se o consumo médio de corrente do acelerómetro, configurou-se o nó sensor no modo PM3. Utilizou-se o modo PM3 para evitar a influência do consumo de energia existente nos outros modos do módulo. Os valores médios de corrente elétrica, obtidos pelo *femto*-amperímetro através de um PC, para os diferentes modos de operação podem ser observados na Tabela 5.6.

Tabela 5.6: Consumo médio de corrente do nó em diferentes modos de operação do acelerómetro.

Modo (Hz)	Corrente (μA)
5	270,94
10	272,08
20	270,24

5.3.4 Estimativa de Duração da Bateria

Para a realização do teste de estimativa de duração da bateria foram obtidos o consumo do nó sensor no modo *deep sleep* e durante o envio da mensagem de KA. Para além disso, foi realizado testes de consumo do acelerómetro em três modos de operação. Sabendo que a capacidade da bateria é de 3600 mAh e uma perda anual de 1% devido à sua auto-descarga. Porém, utilizou-se uma auto-descarga de 10 % para a estimativa do consumo, por ser um pior caso devido a diversos fatores ambientais, como por exemplo a oxidação dos terminais e a variação de temperatura.

A estimativa de duração de bateria, por amostragem de 1 hora, é praticamente determinado pelo tempo que o acelerómetro está ativo. Assim sendo, o nó sensor está ativo durante 1,453 ms e o acelerómetro (acc) está ativo durante 237600 ms (5 Hz), 396720 ms (10 Hz) e 949536 ms (20 Hz). Tendo por base a equação 5.1 foi possível determinar a quantidade média de corrente elétrica consumida durante 1 hora.

$$Carga_{média}[mAh] = \frac{I_{DeepSleep}[mA] * tempo[h] + I_{acelerómetroON}[mA] * tempo[h] + I_{KeepAlive}[mA] * tempo[h]}{3600} \quad (5.1)$$

Equation 5.1: Equação de cálculo da quantidade média de corrente elétrica consumida durante 1 hora.

Através da equação 5.2 foi possível obter a autonomia do sistema em horas

$$Tempo_{vida}[h] = \frac{(1 - \frac{self-discharge}{100}) * CapacidadeBateria[mAh]}{ConsumoSistema[mA]} \quad (5.2)$$

Equation 5.2: Equação de cálculo da estimativa do tempo de vida do nó.

Como calculado em 5.2, a autonomia do sistema com o acelerómetro no modo *low power* de 20 Hz tem um valor de 43484,42 horas, o que corresponde a 4,96 anos. Na Tabela 5.7 é possível observar autonomia do sistema com diferentes valores de perdas, modo de operação do acelerómetro e o envio periódico de KA.

5.3.5 Resumo

Ao longo do presente capítulo foram apresentados alguns testes realizados na validação do sistema desenvolvido. Através de testes unitários e de um teste de integração foi demonstrado que os sistema foi

Tabela 5.7: Consumo médio do nó em diferentes modos de operação do acelerómetro.

Envio periódico de KA (h)	Modo (Hz)	<i>Self-disc</i> (%)	Anos
0,25	5	40	11,277
0,50	5	40	11,262
1	5	40	11,233
0,25	10	30	8,516
0,50	10	30	8,509
1	10	30	8,494
0,25	20	10	4,969
0,50	20	10	4,967
1	20	10	4,963

implementado com sucesso, tendo incorporado todas as funcionalidades propostas. Quanto ao consumo, com o teste de autonomia foi possível simular a longevidade do sistema com diferentes parâmetros.

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo final serão apresentadas as principais conclusões do sistema desenvolvido, bem como algumas sugestões de trabalho futuro, de forma a dar continuidade ao trabalho realizado.

6.1 Conclusões

A presente dissertação de mestrado insere-se no conceito de rede de sensores sem fios, tendo como base uma solução já existente. Esta solução consiste no desenvolvimento de um sistema de baixo custo, capaz de detetar, em tempo real, colisões com guardas de segurança e alertar os responsáveis do sistema e/ou serviços de emergência. Esta colisão pode ser provocada por veículos ou por falha estrutural. Ao longo deste trabalho foram obtidos conhecimentos em: desenho *Altium*; metrologia (recorrendo a Matlab para fazer o controlo de instrumentos científicos); simulação de circuitos eletrónicos (consumo energético); *firmware* Z-Stack; base de dados MongoDB; linguagem de programação *Python*; linguagem de marcação de texto *Latex*; entre outros.

O principal objetivo desta dissertação visava em reduzir o consumo de energia do sensor. Nesse sentido, redesenhou-se o *hardware* e o *software* do nó sensor. O *hardware* do nó coordenador foi reformulado. Quanto à aplicação da WSN, transferiu-se para Z-Stack 3.0.2 da *Texas Instruments* que implementa a pilha protocolar ZigBee para os SoCs CC253x. Além disso, desenvolveu-se um *gateway* responsável pela interface entre a WSN e os micros serviços *web*.

Relativamente ao consumo energético, com as alterações efetuadas no sistema de gestão de energia implementado, obteve-se um consumo inferior em relação às abordagens anteriores [22, 32] no estado *deep sleep* de aproximadamente 4,2 μA , resultado muito próximo dos valores fornecidos dos fabricantes de componentes eletrónicos presentes no *hardware* do nó sensor. É importante salientar que o *hardware* é composto por um LDO, módulo CC2538 / CC2592 da TI e acelerómetro MEMS LIS331DLH.

Quanto à autonomia, o nó sensor apresenta uma autonomia de 4,62 anos para um pior cenário do sistema, com o acelerómetro no modo *low power* de 20 Hz e a bateria com uma auto-descarga de 10% anual. A autonomia do sistema é praticamente determinado pelo tempo que o acelerómetro está ativo. O sistema desenvolvido tem um consumo equivalente ao sistema anterior [32] durante a transmissão da mensagem de KA (1,229 mJ e 1,064 mJ).

Relativamente ao *hardware* reformulou-se o nó sensor e inclusive do nó coordenador com novas regras *Altium* do grupo ESRG.

O *porting* da aplicação do nó sensor e do nó coordenador para nova Z-Stack foi realizada com sucesso. A Z-Stack 3.0.2 tem a certificação do ZigBee 3.0, novas funcionalidades de suporte para *stack* e medidas de segurança. Além disso, libertou-se o *software* do nó sensor de efetuar o cálculo do módulo dos componentes das acelerações e estas passam a ser enviadas em g. Também aumentou-se a potência de transmissão para 22 dBm, valor máximo do módulo CC2538 / CC2592 da TI, o que permite aumentar o alcance das comunicações entre o nó e o coordenador para mais de 200 metros em campo aberto. Por último, com recurso às funcionalidades da nova Z-Stack, implementou-se a recuperação do nó sensor do estado de órfão, ou seja, quando perde a conexão com o nó coordenador.

Quanto ao *gateway* desenvolveu-se uma interface de baixa complexidade entre a WSN e os microserviços *web*. Esta interface é auto-reconfigurável.

Relativamente aos indicadores de qualidade de serviços (RSSI e PER), identifica-se que os obstáculos e a distância entre os dispositivos causam atenuação do sinal.

Os objetivos inicialmente definidos foram cumpridos de uma forma geral. Apesar de todos os objetivos serem cumpridos é possível melhorar alguns pontos do sistema atual, que serão apresentados na seção seguinte.

6.2 Trabalho Futuro

Apesar do sucesso do desenvolvimento do sistema de deteção de colisões, e de todas as vantagens que este sistema oferece não está imune à melhorias. Nesse sentido, como trabalho pode-se enumerar os seguintes pontos:

- Efetuar um estudo sobre estratégia de eficiência energética para o módulo CC2538 / CC2592 da TI;

-
- Customização da Z-Stack da TI de modo a remover funcionalidades não pretendidas e minimizar a taxa de ocupação do processador;
 - Efetuar testes para aferir sobre a distância máxima do módulo CC2538 / CC2592 do nó sensor e do coordenador em campo aberto;
 - Realizar um estudo sobre os tipos de ataques à segurança que o sistema está sujeito;
 - Efetuar mais testes de obtenção de indicadores de QoS;

Bibliografia

- [1] “Sinistralidade rodoviária tem impacto económico e social negativo de 1,2% do PIB | Segurança Rodoviária | PÚBLICO,” [Accessed: 2019-01-03]. [Online]. Available: <https://www.publico.pt/2018/11/18/sociedade/noticia/sinistralidade-rodoviaria-impacto-economico-social-negativo-12-pib-1851543>
- [2] “PORDATA - Vítimas mortais em acidentes de viação:auto-estradas e localidades,” [Accessed: 2019-09-21]. [Online]. Available: <https://www.pordata.pt/Europa/V{itimas+mortais+em+acidentes+de+via{a}}o+auto+estradas+e+localidades-3044>
- [3] “Sinistralidade rodoviária. Número de mortos aumentou 12% em 2018 - Renascença,” [Accessed: 2019-09-29]. [Online]. Available: <https://rr.sapo.pt/2019/08/27/pais/sinistralidade-rodoviaria-numero-de-mortos-aumentou-12-em-2018/noticia/162476/>
- [4] “PORDATA - Acidentes de viação com vítimas: total e por tipo de via - Continente,” [Accessed: 2019-09-21]. [Online]. Available: <https://www.pordata.pt/Portugal/Acidentes+de+via{a}}o+com+v{itimas+total+e+por+tipo+de+via+++Continente-3094-262257>
- [5] “Projeto SustlMS,” [Accessed: 2018-12-03]. [Online]. Available: <https://www.eng.uminho.pt/pt/investigareinnovar/projetoscomempresas/Paginas/projetosustims.aspx>
- [6] “Embedded Systems Research Group (ESRG) - Centro Algoritmi | Centro ALGORITMI,” [Accessed: 2019-05-25]. [Online]. Available: <http://algoritmi.uminho.pt/research-teams/esrg/>
- [7] “Ascendi |,” [Accessed: 2018-11-28]. [Online]. Available: <https://www.ascendi.pt/>
- [8] H. C. Abreu, “Monitorização das Condições Atmosféricas nas Autoestradas utilizando WSNs,” 2014.
- [9] J. Miranda, T. Gomes, R. Abrishambaf, F. Loureiro, J. Mendes, J. Cabral, and J. L. Monteiro, “A Wireless Sensor Network for Collision Detection on Guardrails,” *2014 IEEE 23rd Int. Symp. Ind. Electron.*, pp. 1430–1435, [Accessed: 2018-11-29]. [Online]. Available: <https://ieeexplore.ieee.org/document/6864824>

- [10] T. Gomes, D. Fernandes, M. Ekpanyapong, and J. Cabral, "An IoT-based system for collision detection on guardrails," *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2016-May, pp. 1926–1931, 2016, [Accessed: 2019-02-16]. [Online]. Available: <https://ieeexplore.ieee.org/document/7475061>
- [11] "Ascendi conquista prémio internacional com o projeto SustIMS | Ascendi," [Accessed: 2018-11-28]. [Online]. Available: <https://www.ascendi.pt/noticias/titulo-6-2-3/>
- [12] D. Davino, M. R. Pecce, C. Visone, C. S. Clemente, and A. Ielardi, "Dynamic monitoring of guardrails: Approach to a low-cost system," *2015 IEEE Work. Environ. Energy, Struct. Monit. Syst. EESMS 2015 - Proc.*, pp. 56–60, 2015, [Accessed: 2019-05-12]. [Online]. Available: <https://ieeexplore.ieee.org/document/7175852>
- [13] E. Balestrieri, L. D. Vito, S. Member, and F. Picariello, "IoT System for Remote Monitoring of Bridges : Measurements for Structural Health and Vehicular Traffic Load," pp. 279–284, 2019, [Accessed: 2019-06-16]. [Online]. Available: <https://ieeexplore.ieee.org/document/8792866>
- [14] C. Neves, L. Neves, and U. N. D. Lisboa, "INFRAESTRUTURAS RODOVIÁRIAS SUSTENTÁVEL," 2015, [Accessed: 2019-01-04]. [Online]. Available: http://www.crp.pt/docs/A48S186-8_CRP_T5_074.pdf
- [15] A. Koubaa, M. . Alves, and E. Tovar, "IEEE 802.15.4: a Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks," pp. 2–3, 2013, [Accessed: 2019-04-12]. [Online]. Available: <http://www.cister.isep.pt/docs/ieee802.15.4%3Aafederatingcommunicationprotocolfortime-sensitivewirelessensornetworks/802/view.pdf>
- [16] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A comprehensive survey on real-time applications of WSN," *Futur. Internet*, vol. 9, no. 4, 2017, [Accessed: 2019-03-25]. [Online]. Available: <https://www.mdpi.com/1999-5903/9/4/77/pdf>
- [17] K. Sohraby, D. Minoli, and T. Znati, *Basic Wireless Sensor Technology*, 2007.
- [18] T. Gomes, J. Brito, H. Abreu, H. Gomes, and J. Cabral, "Greenmon: An efficient wireless sensor network monitoring solution for greenhouses," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, March 2015, pp. 2192–2197.
- [19] S. Pinto, J. Cabral, and T. Gomes, "We-care: An iot-based health care system for elderly people," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, March 2017, pp. 1378–1383.

- [20] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018, [Accessed: 2019-07-15]. [Online]. Available: <http://dx.doi.org/10.1016/j.jksuci.2016.10.003>
- [21] H. Karl and A. Willig, *PROTOCOLS AND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS*, 2006.
- [22] A. F. d. S. Ribeiro, "Implementação de nó sensor baseado em SoC ARM e ZigBee," 2018.
- [23] T. M. R. Gomes, "WECO - Wireless Eco ponto," 2011, [Accessed: 2019-01-02]. [Online]. Available: <http://repositorium.sdum.uminho.pt/handle/1822/20190>
- [24] T. Gomes, S. Pinto, T. Gomes, A. Tavares, and J. Cabral, "Towards an fpga-based edge device for the internet of things," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, Sep. 2015, pp. 1–4.
- [25] T. Gomes, F. Salgado, A. Tavares, and J. Cabral, "Cute mote, a customizable and trustable end-device for the internet of things," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6816–6824, Oct 2017.
- [26] "IEEE 802.15 Working Group for Wireless Personal Area Networks (WPANs)," [Accessed: 2019-03-07]. [Online]. Available: <http://www.ieee802.org/15/>
- [27] T. Gomes, S. Pinto, F. Salgado, A. Tavares, and J. Cabral, "Building ieee 802.15.4 accelerators for heterogeneous wireless sensor nodes," *IEEE Sensors Letters*, vol. 1, no. 1, pp. 1–4, Feb 2017.
- [28] T. Gomes, "A Sensor Node SoC Architecture for Extremely Autonomous Wireless Sensor Networks," Ph.D. dissertation, Universidade do Minho, 2017.
- [29] A. Costa, "Detecção de Colisões com Guardas de Segurança," 2013.
- [30] "Internet of Things: Wireless Sensor Networks," Tech. Rep., 2014.
- [31] K. Klues, "Power Management in Wireless Networks." [Online]. Available: https://www.cse.wustl.edu/~jain/cse574-06/ftp/energy_mgmt/index.html
- [32] L. M. M. Pereira, "Sistema ciber-físico para de colisões," 2019.

- [33] M. Standards Committee of the IEEE Computer Society, "IEEE Std 802.15.4-2011, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs)," Tech. Rep., 2006.
- [34] IEEE Standard, "IEEE standard for local and metropolitan area networks. Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.15.4-2006*, pp. 1–26, 2006, [Accessed: 2019-03-11]. [Online]. Available: <https://ieeexplore.ieee.org/servlet/opac?punumber=4152702>
- [35] Z. Alliance, "ZigBee / IEEE 802.15.4," Tech. Rep., [Accessed: 2019-05-16]. [Online]. Available: <http://www.zigbee.org>
- [36] M. Windows, M. Corporation, K. Hori, and A. Sakajiri, *ZigBee Wireless Networks and Transceivers*.
- [37] D. Gislason, *Zigbee Wireless Networking*. Elsevier Ltd, 2008.
- [38] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on ZigBee technology," *ICECT 2011 - 2011 3rd Int. Conf. Electron. Comput. Technol.*, vol. 6, no. February 2019, pp. 297–301, 2011.
- [39] M. Saleiro and E. Ey, "ZigBee uma abordagem prática," 2013, [Accessed: 2019-04-19]. [Online]. Available: <https://pt.scribd.com/document/257004456/Introducao-Ao-Zigbee-Por-Msaleiro>
- [40] R. A. P. Azevedo, "Domótica sem fios baseada em redes ZigBee," 2016.
- [41] "ZigBee Specification," Tech. Rep., 2012.
- [42] "Zigbee PRO Specification 2.2," Tech. Rep., 2019. [Online]. Available: www.ti.com
- [43] D. Y. C. Lie, "'RF-SoC': Integration Trends of On-Chip CMOS Power Amplifier: Benefits of External PA versus Integrated PA for Portable Wireless Communications," *Int. J. Microw. Sci. Technol.*, vol. 2010, pp. 1–7, 2010.
- [44] A. Chirap, V. Popa, E. Coca, and D. A. Potorac, "A study on light energy harvesting from indoor environment: The autonomous sensor nodes," *2014 Int. Conf. Dev. Appl. Syst. DAS 2014 - Conf. Proc.*, pp. 127–131, 2014.
- [45] "CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee® Applications 1 Device Overview," Tech. Rep., 2012, [Accessed: 2018-11-27]. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc2538.pdf>

- [46] "Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com," [Accessed: 2018-12-03]. [Online]. Available: <http://www.ti.com/>
- [47] "ATSAMR21G18A - Wireless Modules," 2016, [Accessed: 2019-05-05]. [Online]. Available: <https://www.microchip.com/wwwproducts/en/ATSAMR21G18A>
- [48] "Home | Microchip Technology," [Accessed: 2019-07-05]. [Online]. Available: <https://www.microchip.com/>
- [49] "SMART ARM-based Microcontrollers," Tech. Rep., 2016, [Accessed: 2019-04-05]. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/Atmel-42243-SAMR21-Xplained-Pro_User-Guide.pdf
- [50] "Atmel ATSAMR21ZLL-EK [USER GUIDE]," Tech. Rep., 2016, [Accessed: 2019-05-04]. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/Atmel-42462-ATSAMR21ZLL-EK_UserGuide.pdf
- [51] "Zigbee and IEEE 802.15.4 wireless microcontroller with 512 kB Flash, 32 kB RAM | NXP," [Accessed: 2019-07-04]. [Online]. Available: <https://www.nxp.com/products/wireless/zigbee/zigbee-and-ieee802.15.4-wireless-microcontroller-with-512-kb-flash-32-kb-ram:JN5169>
- [52] "NXP Semiconductors | Automotive, Security, IoT," [Accessed: 2019-05-04]. [Online]. Available: <https://www.nxp.com/>
- [53] N. Semiconductors, ". General description," Tech. Rep., 2017, [Accessed: 2019-07-04]. [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/JN5169.pdf>
- [54] "CC1352P SimpleLink™ multiband wireless MCU with integrated power amplifier | TI.com," [Accessed: 2019-07-04]. [Online]. Available: <http://www.ti.com/product/CC1352P>
- [55] "Product Folder Order Now CC1352P CC1352P SimpleLink™ High-Performance Dual-Band Wireless MCU With Integrated Power Amplifier 1 Device Overview," Tech. Rep., 2018, [Accessed: 2019-07-04]. [Online]. Available: www.ti.com
- [56] Texas, "TI Wireless Connectivity Portfolio," Tech. Rep., 2011, [Accessed: 2019-01-16]. [Online]. Available: www.ti.com/medical

- [57] “SIMPLICITI SimpliciTI Compliant Protocol Stack | TI.com,” [Accessed: 2019-06-17]. [Online]. Available: <http://www.ti.com/tool/SIMPLICITI>
- [58] “Introduction to SimpliciTI Low-power RF protocol from Texas Instruments Free source code available,” Tech. Rep., [Accessed: 2019-06-17]. [Online]. Available: <http://www.ti.com/lit/ml/swru130b/swru130b.pdf>
- [59] K. Mikhaylov, N. Plevritakis, and J. Tervonen, “Performance analysis and comparison of bluetooth low energy with IEEE 802.15.4 and SimpliciTI,” *J. Sens. Actuator Networks*, vol. 2, no. 3, pp. 589–613, 2013.
- [60] “Z-STACK A fully compliant ZigBee 3.x solution: Z-Stack | TI.com,” [Accessed: 2018-12-07]. [Online]. Available: <http://www.ti.com/tool/Z-STACK>
- [61] B. Huang and Y. Yufeng, *Proceedings of the 5th International Conference on Electrical Engineering ... - Google Livros*, 2016, [Accessed: 2019-04-08]. [Online]. Available: https://books.google.pt/books?id=5GKvDAAAQBAJ&printsec=frontcover&hl=pt-PT&source=gbs_ge_summary_r&cad=0#v=onepage&q=Z-Stack&f=false
- [62] T. Instruments, “Z-Stack 3.0 Developer’s Guide,” Tech. Rep., 2015.
- [63] “CC2592 CC2592 2.4-GHz Range Extender,” Tech. Rep., 2014.
- [64] “Teach, Learn, and Make with Raspberry Pi – Raspberry Pi,” [Accessed: 2019-10-18]. [Online]. Available: <https://www.raspberrypi.org/>
- [65] “Raspberry Pi 3 Model B RASPBerryPI3-MODB-1GB RPI3-MODB-16GB-NOOBS,” Tech. Rep., 2015, [Accessed: 2019-10-07]. [Online]. Available: www.mathworks.com/trademarks
- [66] STMicroelectronics, “MEMS digital output motion sensor ultra low-power high performance 3-axes “nano” accelerometer,” Tech. Rep., 2009, [Accessed: 2018-12-03]. [Online]. Available: www.st.com
- [67] “Primary lithium battery LS 17500,” Tech. Rep. [Online]. Available: www.saftbatteries.com
- [68] T. Gomes, N. Brito, J. Mendes, J. Cabral, and A. Tavares, “Weco: A wireless platform for monitoring recycling point spots,” in *2012 16th IEEE Mediterranean Electrotechnical Conference*, March 2012, pp. 468–472.

- [69] T. Instruments, "Z-Stack 3.0 Sample Application User's Guide," 2016.
- [70] T. Lv, P. Yan, and W. He, "Survey on JSON Data Modelling," *J. Phys. Conf. Ser.*, vol. 1069, no. 1, 2018.
- [71] R. IMHOF, A. A. FROZZA, and R. d. S. MELLO, "Um Survey sobre Extração de Esquemas de Documentos JSON," *Esc. Reg. Banco Dados - ERBD 2017*, no. April, pp. 26–35, 2017.
- [72] "The most popular database for modern apps | MongoDB," [Accessed: 2019-09-18]. [Online]. Available: <https://www.mongodb.com/>
- [73] S. Agrawal, J. P. Verma, B. Mahidhariya, N. Patel, and A. Patel, "Survey on Mongodb : an Open-," vol. 6, no. 12, pp. 1–11, 2015, [Accessed: 2019-10-12].
- [74] S. Branco, "ArchNet - White Paper Encryp on Layer," 2019, [Accessed: 2019-10-05]. [Online]. Available: https://gitlab.com/asergio.branco/white-papers/blob/master/ArchNet_Server/white_paper.md

Anexo A

Diferentes Modos de Comissionamento

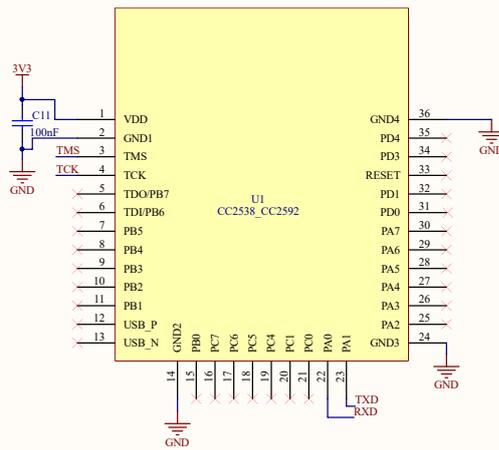
Commissioning mode (BDB_COMMISSIONING_mode)	Status reported (BDB_COMMISSIONING_status)	Description
INITIALIZATION	NETWORK_RESTORED	Only send if the device did restore its network parameters. On end devices, if no parent is found with the restored network parameters, a <i>Parent Lost</i> mode is with status <i>No Network</i> is notified.
NWK_STEERING (for Router and End Devices)	IN_PROGRESS	Notifies when network steering is started (only if the device is not in a network, otherwise reports <i>success</i>)
	NO_NETWORK	No suitable network was found in primary channel or secondary channel masks or the joining process did fail in the attempted networks.
	TCLK_EX_FAILURE	The device successfully joined the network, but could not perform the Trust Center Link Key exchange process. The device will reset to factory new after this notification is reported to the application.
	SUCCESS	The device is now on a network and broadcasted a Management Permit Joining ZDO frame.
NWK_STEERING (for Coordinators)	NO_NETWORK	The device is not on a network, so it cannot perform this action.
	SUCCESS	The device is in a network and has broadcasted a Management Permit Joining ZDO frame.
FORMATION	IN_PROGRESS	Notifies when formation process is started.
	SUCCESS	The network has been created successfully.
	FORMATION_FAILURE	The device could not create the network with the given parameters.
FINDING_BINDING	FB_TARGET_IN_PROGRESS	Indicates the start of the Finding and Binding as target. No notification is given by this callback when the process ends.
	FB_INITIATOR_IN_PROGRESS	Indicates the start of the Finding and Binding as Initiator.
	FB_NO_IDENTIFY_QUERY_RESPONSE	After complete the Finding and Binding process as initiator (single attempt of periodic attempt), no identify query responses were received.
	FB_BINDING_TABLE_FULL	During the Finding and Binding process the binding table got full, so the process stops and no additional binds can be added.
	FAILURE	No endpoint was found to perform Finding and Binding, or the endpoint did not have implemented the Identify cluster properly.
TOUHLINK	TL_TARGET_FAILURE	A node has not joined a network when requested during touchlink.
	TL_NOT_AA_CAPABLE	The initiator is not address assignment capable during touchlink
	TL_NO_SCAN_RESPONSE	No response to a Scan Request inter-PAN command has been received during touchlink
	TL_NOT_PERMITTED	A touchlink steal attempt was made when a node is already connected to a centralized security network.
PARENT_LOST (Only for End Devices)	NO_NETWORK	This is notified if the end device does lose contact with the parent device or if after initialization it cannot find a parent device in the commissioned network.
	NETWORK_RESTORED	Notification that a suitable parent device got found and the rejoin process was successful.

Figura A.1: Diferentes modos de comissionamento.

Anexo B

Esquemáticos do PCB do Nó Coordenador

CPU



Sheet Title **[3.1] - CPU.SchDoc**

Project Title **SustIMS**

Size: **A4** Number: Revision:

Date: **10/30/2019** Time: **5:38:39 PM** Sheet of

Client **UMinho**

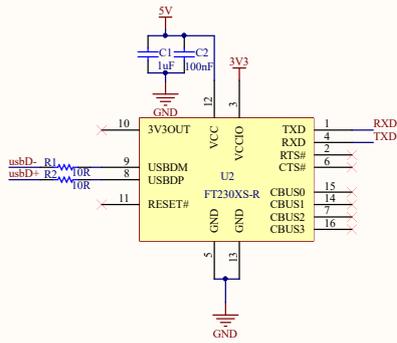
Designer **Elidio Munar Jr.**

Contact **eladiomy14@gmail.com**



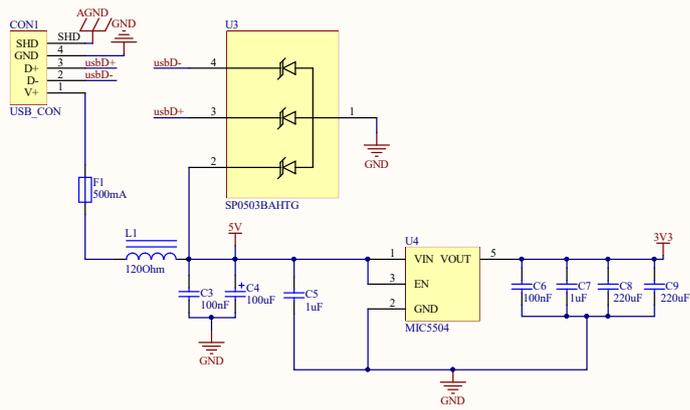
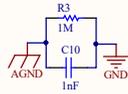
EMBEDED SYSTEMS
RESEARCH
GROUP

FTDI



Sheet Title [3.2] - FTDI.SchDoc			Client UMinho	ESRG EMBEDED SYSTEMS RESEARCH GROUP
Project Title SustIMS			Designer <i>Elidio Munar Jr.</i>	
Size: A4	Number:	Revision:	Contact <i>eladiomy14@gmail.com</i>	
Date: 10/30/2019	Time: 5:38:40 PM	Sheet of		

USB / POWER



Sheet Title **[3.3] - USB_Power.SchDoc**

Project Title **SustIMS**

Size: A4 Number: Revision: Date: 10/30/2019 Time: 5:38:41 PM Sheet of

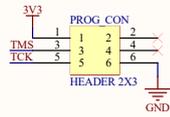
Client **UMinho**

Designer *Elidio Munar Jr.*
Contact *eladiomyl4@gmail.com*



EMBEDDED SYSTEMS
RESEARCH
GROUP

CONNECTORS



Sheet Title [3.4] - Connectors.SchDoc			Client UMinho	 EMBEDDED SYSTEMS RESEARCH GROUP
Project Title SustIMS			Designer <i>Elidio Munar Jr.</i>	
Size: A4	Number:	Revision:	Contact <i>eladiomy14@gmail.com</i>	
Date: 10/30/2019	Time: 5:38:41 PM	Sheet of		