

**UNIVERSIDADE DE PASSO FUNDO**  
**INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM**  
**COMPUTAÇÃO APLICADA**

**UMA PLATAFORMA DE COMUNICAÇÃO**  
**UNIVERSAL UTILIZANDO PROTOCOLO**  
**ZIGBEE**

**Renato Weiller Dallagasperina**

Passo Fundo

2017

**UNIVERSIDADE DE PASSO FUNDO**  
**INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

**UMA PLATAFORMA DE  
COMUNICAÇÃO UNIVERSAL  
UTILIZANDO PROTOCOLO ZIGBEE**

**Renato Weiller Dallagasperina**

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre em Computação  
Aplicada na Universidade de Passo Fundo.

**Orientador: Prof. Dr. Willingthon Pavan**  
**Coorientador: Prof. PhD. José Maurício C. Fernandes**

Passo Fundo

2017

CIP – Catalogação na Publicação

---

- D144p Dallagasparina, Renato Weiller  
Uma plataforma de comunicação universal utilizando  
protocolo Zigbee armazenamento / Renato Weiller  
Dallagasparina. – 2017.  
104 f. : il. color. ; 30 cm.
- Orientador: Prof. Dr. Willingthon Pavan.  
Coorientador: Prof. Phd. José Maurício C. Fernandes.  
Dissertação (Mestrado em Computação Aplicada) –  
Universidade de Passo Fundo, 2017.
1. Protocolo Zigbee. 2. Programas de computador.  
3. Redes de sensores. 4. Dispositivos eletrônicos. I. Pavan,  
Willingthon, orientador. II. Fernandes, José Maurício C.,  
coorientador. III. Título.

CDU: 004.41

---

Catalogação: Bibliotecária Marciéli de Oliveira - CRB 10/2113

**ATA DE DEFESA DO  
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**

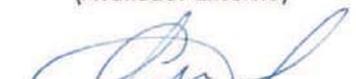
**Renato Weiller Dallagasperina**

Aos 14 dias do mês de **Fevereiro** do ano dois mil e **dezessete**, às **11:00** horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso "**UMA PLATAFORMA DE COMUNICAÇÃO UNIVERSAL UTILIZANDO PROTOCOLO ZIGBEE**", de autoria de **Renato Weiller Dallagasperina**, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores **Willingthon Pavan (UPF) – orientador e presidente**, **Lincoln Zotarelli (University of Florida)**, **Carlos Amaral Holbig (UPF)** e **José Maurício Cunha Fernandes (UPF) - coorientador**. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.

  
**Prof. Dr. Willingthon Pavan**

Presidente da Banca Examinadora (Orientador)

  
**Prof. Dr. Lincoln Zotarelli**  
(Avaliador Externo)

  
**Prof. Dr. Carlos Amaral Holbig**  
(Avaliador)

  
**Prof. Dr. José Maurício Cunha Fernandes**  
(Coorientador)

  
**Prof. Dr. Carlos Amaral Holbig**  
Coordenador do PPGCA



# UMA PLATAFORMA DE COMUNICAÇÃO UNIVERSAL UTILIZANDO PROTOCOLO ZIGBEE

## RESUMO

Com o rápido avanço tecnológico, a computação pervasiva tem ganho cada vez mais destaque. Em nosso dia a dia, facilmente encontramos dispositivos eletrônicos embarcados em objetos comuns, trazendo novas funcionalidades e aplicações a esses objetos. Essa geração está sendo chamada de *Internet of Things* (IoT), porém, para dar vazão às informações coletadas ou mesmo permitir o processamento remoto destas, os dispositivos ficam dependentes de um sistema de comunicação, sendo um ponto crítico em aplicações que devem apresentar um alto grau de disponibilidade e confiabilidade. A baixa disponibilidade de módulos de comunicação que possibilitem a implementação transparente, confiável e amigável muitas vezes impede o rápido desenvolvimento de aplicações nessa área. Visando suprir essa lacuna, desenvolveu-se uma plataforma de hardware e software para a comunicação sem fio entre dispositivos, utilizando o protocolo Zigbee. Essa plataforma é composta por dois módulos, um para curtas distâncias, de até 100 metros, e outro para aplicações em que a distância supere este valor. A plataforma visa facilitar o desenvolvimento de soluções para as mais diversas áreas, como redes de sensores sem fio (RSSF), coletores de dados e atuadores. Para a validação da plataforma, foram feitos ensaios com os módulos construídos para a verificação do consumo de energia e do comportamento do sistema de alimentação, além da verificação da distância de transmissão alcançada, da capacidade de formação de rede e da transmissão de dados. Após a validação, pôde-se concluir que a plataforma apresenta características de baixo consumo de energia, baixo custo e alta confiabilidade na transmissão de dados e formação de rede. A plataforma pode ser utilizada como uma plataforma universal de apoio ao desenvolvimento de aplicações utilizando o protocolo Zigbee.

Palavras-Chave: Comunicação, Zigbee, baixo consumo de energia, baixo custo.



# A UNIVERSAL COMMUNICATION PLATFORM USING ZIGBEE PROTOCOL

## ABSTRACT

With the rapid technological advancement, pervasive computing has gained more and more prominence. Nowadays, we easily find electronic devices embedded in common objects, bringing new functionalities and applications to these objects. This is being called Internet of Things (IoT), but in order to let the information collected or even remotely process it, the devices are dependent on a communication system, being a critical point in applications that must be reliable and highly available as well. The low offer of communication modules that enable transparent, reliable and user-friendly implementation often prevents the rapid development of applications in this area. Aiming to fill this gap, a hardware and software platform for wireless communication between devices was being developed using the Zigbee protocol. This platform is composed of two modules, one for short distances, up to 100 meters, and another for applications where the distance exceeds this value. The platform aims to facilitate the development of solutions for the most diverse areas, such as wireless sensor networks (WSNs), data collectors and actuators. For the validation of the platform, tests were carried out to check the energy consumption and behavior of the power system, as well as the verification of the transmission distance achieved, the capacity to form the network and the data transmission. After the validation, it was concluded that the platform presents characteristics of low power consumption, low cost and reliability in data transmission and network formation. The platform can be used as a universal platform to support the development of applications using the Zigbee protocol.

Keywords: Communication, Zigbee, low power consumption, low cost.



## LISTA DE FIGURAS

Figura 1.	Exemplo de um RSSF <i>ad hoc</i> [24]. . . . .	26
Figura 2.	Grandes grupos de redes sem fio com as suas respectivas características de alcance e taxa de transmissão [31]. . . . .	28
Figura 3.	Comparativo entre tecnologias inseridas nos grupos WPAN e WLAN, comparando a taxa de transferência, a complexidade de implementação e o consumo de energia [5]. . . . .	29
Figura 4.	Complementação entre os padrões IEEE 802.15.4 e Zigbee, quando comparados com o modelo ISO/OSI [6]. . . . .	30
Figura 5.	Canais e frequências utilizadas pelo padrão IEEE 802.15.4 [5]. . . . .	31
Figura 6.	Arquitetura completa do padrão Zigbee, incluindo a interligação entre as camadas [29]. . . . .	32
Figura 7.	Tipos de dispositivos e funções exercidas nos padrões Zigbee e IEEE 802.15.4 [36]. . . . .	33
Figura 8.	Topologias de rede previstas pelo padrão IEEE 802.15.4, sendo (a) topologia de rede <i>star</i> , (b) topologia de rede <i>mesh</i> e (c) topologia de rede <i>cluster-tree</i> [9] . . . .	34
Figura 9.	Primeiro protótipo da plataforma Arduino [46]. . . . .	38
Figura 10.	Módulos da plataforma <i>OpenMote</i> [47]. . . . .	39
Figura 11.	Camadas de rede implementadas pelo padrão LoraWAN [52]. . . . .	40
Figura 12.	Um exemplo de implementação de rede utilizando o padrão LoraWAN [52]. . . . .	41
Figura 13.	Comparação entre classes de dispositivos utilizando padrão LoraWAN considerando o consumo de energia e a latência na recepção de dados [52]. . . . .	42
Figura 14.	Diagrama em blocos, apresentando os dispositivos de entrada e saída, o sistema de alimentação e a etapa de RF do sistema proposto. . . . .	43
Figura 15.	SoC CC2538 comparado, em tamanho, a uma moeda. O tamanho reduzido do SoC CC2538 possibilita o desenvolvimento de produtos finais extremamente compactos. . . . .	46
Figura 16.	Diagrama elétrico simplificado do <i>range extender</i> CC2592 [62]. . . . .	47
Figura 17.	Uma aproximação entre as distâncias de transmissão alcançadas entre os módulos <i>low power</i> e <i>high power</i> [5]. . . . .	48
Figura 18.	Modelo utilizado para o desenho da <i>Inverted-F Antenna</i> [65]. . . . .	49
Figura 19.	Pinagem do conector para <i>debug</i> e gravação do SoC CC2538 utilizando o padrão ARM JTAG de 20 pinos [74]. . . . .	51
Figura 20.	Módulo <i>low power</i> desenvolvido utilizando o SoC CC2538 da <i>Texas Instruments</i> (do autor). . . . .	54

Figura 21.	Detalhes do sistema de alimentação, mostrando os conectores CN1, P7 e P1, e do sistema de RF, mostrando o conector AN1 e os indutores L2 e L4 (do autor).	54
Figura 22.	Módulo <i>high power</i> projetado, utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2592, ambos fabricados pela <i>Texas Instruments</i> (do autor).	55
Figura 23.	Detalhes do sistema de alimentação, mostrando os conectores CN1, P14, P15 e P1, e do sistema de RF, mostrando o conector AN1 e os capacitores C25 e C26 (do autor).	56
Figura 24.	<i>Sniffer</i> projetado, utilizando o SoC CC2531 fabricado pela <i>Texas Instruments</i> (do autor).	56
Figura 25.	Circuito implementado para medição do consumo de energia do módulo com o SoC CC2538 (do autor).	57
Figura 26.	Forma de onda da diferença de potencial medida sobre o resistor de 10 ohms. Potência de saída do dispositivo programada em +7dBm. Osciloscópio com resolução de 100 mV no eixo y e 20 ms no eixo x (do autor).	58
Figura 27.	Função <i>main</i> do SoC CC2538 modificada para utilização nas baterias de testes de consumo de energia (do autor).	59
Figura 28.	Função <i>main</i> do <i>firmware</i> utilizado nas baterias de testes para os ensaios de distância de transmissão do módulo <i>low power</i> desenvolvido (do autor).	60
Figura 29.	Visualização de informações recebidas dos <i>end devices</i> pelo nodo coordenador da rede (do autor).	68
Figura 30.	Trecho de código do arquivo "GenericApp.c", onde é declarado o vetor utilizado para a transmissão dos valores de temperatura lidos pelo conversor A/D (do autor).	68
Figura 31.	Trecho de código do arquivo "GenericApp.c", da função de inicialização de uma tarefa, onde é inicializado o conversor A/D do SoC CC2538 (do autor).	68
Figura 32.	Trecho de código do arquivo "GenericApp.C", da função de processamento da tarefa, onde é efetuada a leitura do conversor A/D (do autor).	69
Figura 33.	Trecho de código do arquivo "GenericApp.C", da função de transmissão de dados, onde o vetor com as informações lidas do conversor A/D é transmitido para o coordenador (do autor).	69
Figura 34.	Função do <i>firmware</i> utilizado no coordenador da rede Zigbee, para exibir a atualização dos valores de temperatura enviados pelos <i>end-devices</i> (do autor).	70
Figura 35.	Configuração de rede, utilizando os módulos desenvolvidos como <i>end device</i> e os módulos SMARTRF06EBK e CC2538EMK como coordenador (do autor).	70
Figura 36.	Configuração de rede, utilizando os módulos desenvolvidos como <i>end device</i> , um módulo CC2538EMK como roteador e os módulos SMARTRF06EBK e CC2538EMK como coordenador (do autor).	71

Figura 37.	Pacote de dados capturado utilizando o <i>sniffer</i> e a IDE <i>Packet Sniffer</i> (do autor).	71
Figura 38.	Diagrama elétrico parcial (1/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito do microcontrolador (do autor).	89
Figura 39.	Diagrama elétrico parcial (2/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito do conector USB (do autor).	90
Figura 40.	Diagrama elétrico parcial (3/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito de alimentação externa, seleção da alimentação e capacitores de desacoplamento (do autor).	91
Figura 41.	Diagrama elétrico parcial (4/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito dos conectores (do autor).	92
Figura 42.	Diagrama elétrico parcial (5/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito de RF (do autor).	93
Figura 43.	<i>Layer top</i> da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538. (do autor).	94
Figura 44.	<i>Layer bottom</i> da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538. (do autor).	94
Figura 45.	Diagrama elétrico parcial (1/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. Referente ao circuito do microcontrolador (do autor).	95
Figura 46.	Diagrama elétrico parcial (2/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. Referente ao circuito da conexão USB (do autor).	96
Figura 47.	Diagrama elétrico parcial (3/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. Referente ao circuito de alimentação externa, seleção da alimentação e capacitores de desacoplamento (do autor).	97
Figura 48.	Diagrama elétrico parcial (4/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. Referente ao circuito dos conectores (do autor).	98
Figura 49.	Diagrama elétrico parcial (5/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. Referente ao circuito de RF (do autor).	99
Figura 50.	<i>Layer top</i> da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. (do autor).	100
Figura 51.	<i>Layer bottom</i> da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2892. (do autor).	100

Figura 52.	Diagrama elétrico parcial (1/4) do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. Referente ao circuito do microcontrolador (do autor). . . . .	101
Figura 53.	Diagrama elétrico parcial (2/4) do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. Referente ao circuito do regulador de tensão e capacitores desacoplamento (do autor).102	102
Figura 54.	Diagrama elétrico parcial (3/4) do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. Referente ao circuito dos conectores, leds e botões (do autor). . . . .	103
Figura 55.	Diagrama elétrico parcial (4/4) do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. Referente ao circuito de RF (do autor). . . . .	104
Figura 56.	<i>Layer top</i> da placa de circuito impresso do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. (do autor). . . . .	104
Figura 57.	<i>Layer bottom</i> da placa de circuito impresso do <i>sniffer</i> desenvolvido utilizando o SoC CC2531. (do autor). . . . .	104

## LISTA DE TABELAS

Tabela 1.	Comparativo de propriedades entre tecnologias wireless [5]. . . . .	28
Tabela 2.	Taxa de transferência de acordo com a frequência de transmissão [6]. . . . .	31
Tabela 3.	Comparativo entre Características de módulos e Circuitos Integrados. . . . .	36
Tabela 4.	Valoração das características de cada módulo ou sistema estudado. . . . .	45
Tabela 5.	Medidas utilizadas na construção da <i>Inverted-F Antenna</i> [65]. . . . .	48
Tabela 6.	Custo para o desenvolvimento do módulo <i>low power</i> utilizando o SoC CC2538.	53
Tabela 7.	Custo para o desenvolvimento do módulo <i>high power</i> utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2592. . . . .	53
Tabela 8.	Valores de configuração da potência de saída, do rádio transmissor do SoC CC2538, em dBm e o valor equivalente em mW [60]. . . . .	57
Tabela 9.	Valores de diferença de potencial e corrente, com alimentação do circuito de 3,5VCC (do autor). . . . .	58
Tabela 10.	Valores de diferença de potencial e corrente lidos, com alimentação do circuito de 2,2VCC (do autor). . . . .	58
Tabela 11.	Levantamento das distâncias máximas alcançadas e quantidade de pacotes perdidos, na transmissão de dados entre dois dispositivos. Valores obtidos em campo aberto, com visada direta entre os dispositivos (do autor). . . . .	61
Tabela 12.	Detalhamento do material necessário adquirido para o projeto do módulo <i>sniffer</i> , utilizando o SoC CC2531 (do autor). . . . .	85
Tabela 13.	Detalhamento do material necessário adquirido para o projeto do módulo Zigbee, utilizando o SoC CC2538 (do autor). . . . .	86
Tabela 14.	Detalhamento do material necessário adquirido para o projeto do módulo Zigbee,, utilizando o SoC CC2538 em conjunto com o <i>range extender</i> CC2592 (do autor). . . . .	87
Tabela 15.	Módulos adquiridos para suporte no desenvolvimento do projeto (do autor). .	87



## LISTA DE SIGLAS

A/D – Analógico/Digital  
AES – Advanced Encryption Standard  
API – Application Programming Interface  
APS – Application Support  
BALUN – Balanced to Unbalanced Transformer  
CI – Circuito Integrado  
CJTAG – compact Joint Test Action Group  
CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance  
CSS – Chirp Spread Spectrum  
DBM – Decibel Milliwatt  
DMA – Direct Memory Access  
EEPROM – Electrically-Erasable Programmable Read-Only  
FFD – Full Function Device  
FSK – Frequency-Shift Keying  
GPIO – General-Purpose Inputs/Outputs  
GPRS – General Packet Radio Services  
HAL – Hardware Abstraction Layer  
I2C – Inter-Integrated Circuit  
IDE – Integrated Development Environment  
IDII – Interaction Design Institute Ivrea  
IEEE – Institute of Electrical and Electronics Engineers  
IFA – Inverted-F Antenna  
I/O – Input/Output  
IOT – Internet of Things  
IRDA – Infra Red Data Association  
ISM – Industrial, Scientific and Medical  
ISO/OSI – International Organization for Standardization / Open Systems Interconnection  
JTAG – Joint Test Action Group  
KBPS – Kilo bits per second  
LNA – Low Noise Amplifier  
LORA – Long Range  
LORAWAN – Long Range Wide Area Network

LPRF – Low Power Radio Frequency

LPWAN – Low Power Wide Area Network

LR-WPAN – Low Rate - Wireless Personal Area Network

M2M – Machine to Machine

MA – Miliampere

MAC – Media Access Control

MBPS – Mega bits per second

MCU – Micro Controller Unit

MIFA – Meandered Inverted-F Antenna

MIPS/W – Milhões de Instruções por Segundo por Watt

MIT – Massachusetts Institute of Technology

NWK – Network

OSAL – Operational System Abstraction Layer

OTA – Over The Air

PA – Power Amplifier

PAN – Personal Area Network

PHY – Physical Layer

RAM – Random Access Memory

RC – Resistive and Capacitive

RF – Rádio Frequência

RFD – Reduced Function Device

RSSF – Rede de Sensores sem Fio

RTC – Real Time Clock

SMA – SubMiniature version A

SMD – Surface Mount Device

SOC – System on Chip

SPI – Serial Peripheral Interface

SSI – Synchronous Serial Interface

TI – Texas Instruments

TIMAC – Texas Instruments Medium Access control

TX – Transmissão

UA – Microampere

UART – Universal Asynchronous Receiver/Transmitter

USB – Universal Serial Bus

UWB – Ultra Wide Band

VCC – Volts Corrente Contínua

WLAN – Wireless Local Area Network

WMAN – Wireless Metropolitan Area Network

WPAN – Wireless Personal Area Network

WSN – Wireless Sensor Network

WWAN – Wireless Wide Area Network

ZCP – ZigBee Compliant Platform

ZDO – Zigbee Device Objects



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>23</b>
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>25</b>
2.1	MÓDULOS DE COLETA E COMUNICAÇÃO	25
2.1.1	<b>Redes de Sensores Sem Fio (RSSF)</b>	25
2.1.2	<b>Tecnologias de Transmissão de Dados sem Fio</b>	27
2.1.3	<b>Wireless Personal Area Network</b>	29
2.1.3.1	O padrão IEEE 802.15.4	30
2.1.3.2	O Padrão Zigbee	31
2.1.3.3	Tipos de Topologias de Rede dentro do padrão Zigbee	33
2.1.4	<b>Sobre o consumo de energia em Redes de Sensores Sem Fio</b>	34
2.1.5	<b>Modo Sleep e Idle</b>	35
2.1.6	<b>Módulos e Circuitos Integrados</b>	35
2.2	A PLATAFORMA ARDUINO	36
2.3	TRABALHOS RELACIONADOS	38
2.3.1	<b>OpenMote</b>	39
2.3.2	<b>LoraWAN</b>	39
2.4	CONCLUSÕES DO CAPÍTULO	42
<b>3</b>	<b>SOBRE O HARDWARE</b>	<b>43</b>
3.1	ESCOLHA DO MICROCONTROLADOR	43
3.1.1	<b>System on Chip CC2538</b>	45
3.2	RANGE EXTENDER CC2592	46
3.3	MÓDULOS LOW POWER E HIGH POWER	47
3.4	ANTENA ON-BOARD DE 2.4GHZ	48
3.5	CIRCUITO DE RF	49
3.6	DISPOSITIVOS DE I/O	50
3.6.1	<b>Entradas e saídas de uso geral</b>	50
3.6.2	<b>Módulo de controle de entradas e saídas</b>	50
3.7	DEMAIS PERIFÉRICOS	51
3.7.1	<b>Conector JTAG</b>	51
3.7.2	<b>System clock e real time clock</b>	51

3.7.3	<b>BootLoader</b> .....	52
3.7.4	<b>Monitor de bateria e sensor de temperatura</b> .....	52
3.8	TESTES E RESULTADOS DE HARDWARE .....	52
3.8.1	<b>Circuito de alimentação</b> .....	52
3.8.2	<b>Custos de desenvolvimento dos módulos</b> .....	53
3.8.3	<b>Módulo low power</b> .....	54
3.8.4	<b>Módulo high power</b> .....	55
3.8.5	<b>Sniffer CC2531</b> .....	55
3.8.6	<b>Ensaio de consumo de energia</b> .....	56
3.8.7	<b>Ensaio de distância de transmissão</b> .....	58
3.9	CONCLUSÕES DO CAPÍTULO .....	60
<b>4</b>	<b>SOBRE O SOFTWARE</b> .....	<b>63</b>
4.1	FOUNDATION FIRMWARE .....	63
4.2	TIMAC .....	63
4.3	PILHA ZIGBEE .....	64
4.4	Z-STACK - MESH .....	64
4.5	SISTEMA OPERACIONAL .....	65
4.5.1	<b>OSAL</b> .....	65
4.5.2	<b>HAL</b> .....	66
4.5.3	<b>Power management API</b> .....	67
4.6	APLICAÇÃO DESENVOLVIDA PARA O ENSAIO DE FORMAÇÃO DE REDE .....	67
4.7	CONCLUSÕES DO CAPÍTULO .....	70
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>73</b>
<b>6</b>	<b>TRABALHOS FUTUROS</b> .....	<b>75</b>
	<b>REFERÊNCIAS</b> .....	<b>77</b>
	<b>APÊNDICE A</b> – Listas Detalhadas dos componentes e módulos adquiridos para o desenvolvimento do trabalho .....	<b>85</b>
	<b>APÊNDICE B</b> – Detalhes do Projeto do Módulo Utilizando o SoC CC2538 .....	<b>89</b>
	<b>APÊNDICE C</b> – Detalhes do Projeto do Módulo Utilizando o SoC CC2538 em Conjunto com o Range Extender CC2592 .....	<b>95</b>

<b>APÊNDICE D – Detalhes do Projeto do Sniffer Utilizando o SoC CC2531 . . . . .</b>	<b>101</b>
--	------------



## 1. INTRODUÇÃO

A comunicação, ou transmissão de informações, foi sempre um grande desafio para a humanidade. Séculos atrás, o homem não possuía nenhum tipo de tecnologia que auxiliasse nessa tarefa, o que fazia com que fosse muito difícil enviar informações a longas distâncias.

O advento da *World Wide Web* mudou totalmente a vida das pessoas. Mudou o modo como vivemos. Democratizou a informação. Não é mais necessário esperar dias até determinada informação chegar pelo correio da Alemanha, por exemplo. Essa mesma informação pode chegar em alguns instantes se enviada por e-mail. Tudo ficou muito mais ágil e acessível, podendo-se ter acesso a praticamente qualquer informação, em qualquer local e em qualquer horário, bastando, para isso, um dispositivo com conexão à internet [1].

Entretanto, os meios de comunicação estão em constante evolução. Em suas primeiras versões, redes de computadores eram necessariamente cabeadas, prendendo o usuário a um ponto fixo. Com o avanço tecnológico, novos meios de comunicação foram desenvolvidos, como as redes sem fio, cujo desenvolvimento traz ainda mais agilidade para o cotidiano das pessoas, pois, além do acesso à informação, ganha-se mobilidade [2].

Uma área que sofre muita influência do avanço tecnológico é a eletrônica. Muitos estudos são desenvolvidos nessa área para a criação ou o melhoramento de componentes eletrônicos, principalmente microprocessadores e microcontroladores. Nos últimos anos, esses componentes têm cada vez mais aumentado sua capacidade de processamento e memória, reduzido o consumo de energia e passado por um processo de miniaturização muito grande. Com isso, uma área que tem se destacado cada vez mais é a computação pervasiva. Dispositivos móveis já fazem parte do cotidiano das pessoas há diversos anos, porém, uma nova geração de dispositivos vem sendo desenvolvida, para que sejam embarcados em objetos comumente usados em nosso dia a dia. Essa nova geração está sendo chamada de *Internet of Things* (IoT), ou Internet das Coisas, que interliga estes objetos à rede, criando novas funções e características que auxiliam e automatizam tarefas cotidianas, como por exemplo uma geladeira que seja capaz de identificar quando determinado produto em seu interior está acabando e automaticamente faz um pedido de reposição. Isto representa uma nova revolução na vida das pessoas, pois além de todos estarem "conectados", tudo estará "conectado" [3, 4].

Atualmente, dispositivos móveis que fazem uso de tecnologias de redes sem fio, com grandes taxas de transferência de dados, são amplamente difundidos e utilizados. Podem ser citados como exemplo dessas redes sem fio o padrão *Wireless Local Area Network* (WLAN), IEEE 802.11 (a,b,g e n) e as redes de transmissão de dados móveis 3G e 4G, as quais têm como principal diferença entre si a área de cobertura. O padrão IEEE 802.11 normalmente possui uma área de cobertura menor (em torno de 100m), enquanto as tecnologias 3G e 4G cobrem áreas muito maiores. Essas duas tecnologias, apesar de serem imprescindíveis nos dias atuais, não podem ser utilizadas em qualquer tipo de aplicação. O consumo de energia e sua complexidade de implementação são dois

fatores impeditivos do seu uso em aplicações nas quais a fonte de energia e o poder computacional disponíveis são limitados [5, 4].

Assim, para essas aplicações, foram desenvolvidos os padrões IEEE 802.15.4 e Zigbee, os quais são utilizados especificamente em aplicações onde os dispositivos dispõem de baixo poder computacional e fonte de energia limitada, geralmente fazendo uso de baterias. Isso faz com que os dispositivos que utilizam esses padrões possuam baixa taxa de transferência de dados (até 250 Kbps) [5, 6].

Com o advento da Internet das Coisas, o desenvolvimento de novos dispositivos ou a implementação de novas funcionalidades a objetos comuns utilizados no dia a dia tem sido foco de muitos estudos. Para auxiliar na etapa de desenvolvimento, algumas ferramentas estão disponíveis no mercado, como por exemplo a plataforma Arduino [7] ou a plataforma para comunicação sem fio XBEE [8]. Essas são ferramentas excepcionais e apresentam algumas características únicas, porém, podem não contemplar a melhor solução possível para o desenvolvimento de dispositivos de comunicação sem fio. A plataforma Arduino apesar de apresentar um ótimo ambiente de desenvolvimento, não conta com um módulo integrado de comunicação sem fio, e, para algumas aplicações, sua capacidade computacional pode não ser suficiente. Por outro lado, a plataforma XBEE é específica para transmissão de dados sem fio, porém, o seu ambiente de desenvolvimento não é tão atrativo quanto o do Arduino por exemplo, e sua capacidade computacional também pode ser limitada para determinadas aplicações.

Dessa forma, este trabalho tem como objetivo o projeto e o desenvolvimento de uma plataforma de comunicação universal, para transmissão de dados sem fio utilizando o protocolo Zigbee. Para que esse objetivo seja alcançado, são desenvolvidos dois módulos. Um para aplicações em curtas distâncias, de até 100 metros, e um segundo módulo, para aplicações em que a distância supere este índice. Os módulos também apresentam outras características importantes, que são: baixo consumo de energia, permitindo o uso de fontes de energia limitadas, tais como pilhas e baterias; baixo custo, viabilizando o desenvolvimento de aplicações onde o uso de vários módulos se faça necessário; confiabilidade na transmissão dos dados, diminuindo a manutenção necessária; e a possibilidade de personalização do *firmware* embarcado, permitindo a adaptação e o uso dos módulos desenvolvidos em diferentes aplicações. A plataforma desenvolvida apresenta grande importância, principalmente por viabilizar o desenvolvimento de novas aplicações, nas quais as características apresentadas pelos módulos construídos são fundamentais.

## 2. REVISÃO DE LITERATURA

Neste capítulo, é feita uma breve revisão sobre os conceitos e as tecnologias utilizados neste trabalho. São apresentadas revisões sobre: módulos de coleta e comunicação; redes de sensores sem fio (RSSF); o protocolo IEEE 802.15.4; e o protocolo Zigbee. Também são levantadas algumas questões sobre consumo de energia, "*Open Source Hardware*" e alguns trabalhos relacionados. Ao final do capítulo, são apresentadas algumas considerações acerca dos conceitos e tecnologias apresentados.

### 2.1 MÓDULOS DE COLETA E COMUNICAÇÃO

Nos últimos anos, com os avanços nas tecnologias de sistemas microeletromecânicos, comunicações sem fio e sistemas eletrônicos e digitais, o desenvolvimento de sistemas multifuncionais de baixo custo, baixo consumo, de tamanho reduzido e com alguma capacidade de comunicação sem fio têm se tornado cada vez mais viáveis. O aumento na capacidade de memória, processamento de dados e comunicação destes sistemas tem permitido o desenvolvimento de RSSF cada vez maiores e mais complexas [9, 4, 10].

Uma nova geração de dispositivos vem sendo desenvolvida para que eles sejam embarcados em objetos comumente usados em nosso dia a dia. Essa nova geração está sendo chamada de *Internet of Things* (IoT). A Internet das Coisas interliga esses objetos à rede, criando novas funções e características que auxiliam e automatizam tarefas cotidianas. Muitos esforços estão sendo aplicados nesse sentido [11].

Considerando essa tendência, diversos fabricantes têm desenvolvido sistemas que oferecem microcontroladores e transceivers<sup>1</sup> de comunicação sem fio integrados em um único módulo [12, 13], ou em um Circuito Integrado (CI) [14, 15, 16, 17, 18, 19, 20]. Neste trabalho, a designação CI pode ser substituída por *System On Chip* (SoC). Os microcontroladores ainda podem oferecer diversos outros meios de comunicação (USB, UART, SPI, I2C), utilizados para a conexão com os mais variados tipos de sensores disponíveis no mercado. Com isso, o desenvolvimento de soluções [21, 22] que implementam redes de sensoriamento sem fio mais robustas, complexas e viáveis tende a ficar cada vez mais acessível.

#### 2.1.1 Redes de Sensores Sem Fio (RSSF)

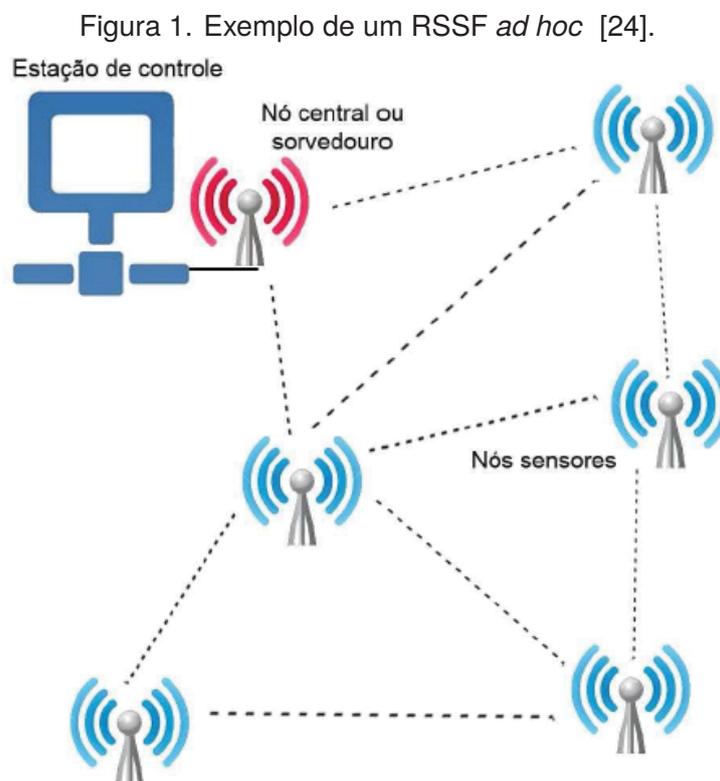
Uma RSSF, ou *Wireless Sensor Network* (WSN), é um tipo especial de rede sem fio que pode se utilizar de vários nodos equipados com diferentes tipos de sensores para monitorar variáveis ambientais. Cada nodo pode ser constituído de um ou mais sensores, um microprocessador e um

---

<sup>1</sup>É um módulo de hardware que possui integrado transmissor e receptor, utilizando componentes comuns para ambas as funções.

transceiver de comunicação sem fio. Esses nodos podem ser organizados com diferentes topologias de rede. A utilização das RSSF se torna um atrativo em lugares remotos onde o uso de uma rede cabeada é inviável [9].

As RSSF se diferenciam de redes de computadores tradicionais em diversos pontos. Elas possuem sistemas de autoconfiguração e adaptação na ocorrência de falhas de comunicação ou perda de algum nodo da rede. Devido a essas características, uma RSSF tende a ser autônoma [4, 9]. O envio e a recepção dos dados são descentralizados, sendo que qualquer nodo da rede pode assumir o papel de roteador, por isso, as RSSF podem ser consideradas um tipo especial de redes *ad hoc* [4, 23]. Em redes tradicionais, a comunicação entre os elementos de uma rede é feita por meio de estações rádio base, ou um *switch*. Já em redes *ad hoc*, qualquer nodo presente pode transportar dados enviados por outros nodos, exercendo a função de um *switch*, como mostra a Figura 1 [4].



RSSF podem ainda possuir, dependendo da aplicação, uma série de limitações, tais como baixa capacidade de processamento, memória e restrições no consumo de energia [4, 25]. A limitação mais crítica é a restrição no consumo de energia. Normalmente, os nodos são estacionários e alimentados por baterias, que possuem capacidade limitada. Assim, embora as posições dos nodos não mudem, a topologia de rede pode mudar dinamicamente devido ao gerenciamento de energia dos nodos sensores. Para reduzir o consumo de energia, os nodos podem alternar agressivamente o consumo de energia de seus transceivers, até desativá-los. Nesse ambiente dinâmico, é um grande desafio fornecer conectividade de rede, minimizando o consumo de energia. Porém, com um gerenciamento eficiente de energia, uma RSSF pode aumentar significativamente sua vida útil [26, 9].

O desenvolvimento de RSSF facilita muitas aplicações existentes e pode impulsionar a criação de aplicações totalmente novas. Essa premissa depende de muitos fatores, mas alguns podem ser destacados. Além da necessidade de se construir nodos de baixo custo, simples de usar e com longa vida útil, um ponto crucial para o desenvolvimento de aplicações reais é a capacidade de interfacimento com diferentes tipos de sensores. Para muitos parâmetros físicos, existe uma tecnologia de sensoriamento apropriada, que pode ser integrada a um nodo de uma RSSF. Alguns dos mais populares são sensores de temperatura, umidade, luz visível e infravermelha, acústica, vibração, pressão e de tensão mecânica, além de sensores químicos e magnéticos, entre outros [27]. As possíveis aplicações de uma RSSF são muitas. Pelas características apresentadas, as RSSF são muito utilizadas em ambientes onde se deseja o monitoramento constante de variáveis físicas de forma autônoma e com pouca ou nenhuma interferência no meio monitorado [28]. Do mesmo modo, as RSSF podem ser usadas também para o controle de atuadores [4, 27]. Assim o uso de RSSF não se limita a uma área de aplicação específica, podendo tais redes serem usadas em áreas que vão desde a medicina até a automação industrial [28, 27].

### 2.1.2 Tecnologias de Transmissão de Dados sem Fio

Com o rápido avanço da tecnologia, surgiu também a necessidade de desenvolvimento de meios de comunicação sem fio, provendo mobilidade a usuários e aplicações. Inicialmente, esses meios de comunicação foram desenvolvidos para a transmissão de voz e dados, com altas taxas de transferências, alto custo de implementação, alta complexidade e consumo de energia elevado. Com isso, algumas aplicações que não contavam com esses recursos ficavam inviabilizadas. Mas isso vem mudando. Novos padrões de comunicação sem fio vêm sendo desenvolvidos, cada um focado em uma necessidade ou característica específica, tais como Wifi, Bluetooth, WiMAX, Zigbee, entre outros [29].

O *Institute of Electrical and Electronics Engineers* (IEEE) considera a existência de quatro grandes grupos quando se trata de redes sem fio [29, 30]:

- *Wireless Personal Area Network* (WPAN) – aqui enquadram-se tecnologias wireless de curto alcance (de 10 a 100 metros), podendo ser bem definida como uma rede composta por dispositivos pessoais. Nesse grupo estão inseridas tecnologias como o Bluetooth, o *Infra Red Data Association* (IrDA) e o *Low Rate WPAN* (LR-WPAN) [29, 30];
- *Wireless Local Area Network* (WLAN) – enquadram-se as tecnologias que são destinadas a redes wireless com alcance entre 100 e 300 metros. Possui taxa de transferência elevada e tem o objetivo de ser uma opção ou mesmo substituir as redes cabeadas convencionais [29, 30];
- *Wireless Wide Area Network* (WWAN) – é orientado para serviços de telecomunicações (voz e dados) para longas distâncias de transmissão. A tecnologia de GPRS está inserida nesse grupo. As velocidades de transmissão de dados podem variar de poucos Kbps até aproximadamente 15 Mbps [29, 30]; e

- *Wireless Metropolitan Area Network (WMAN)* – destinado a tecnologias de longo alcance, podendo atingir quilômetros em sua área de cobertura. Tem como objetivo fornecer acesso à banda larga em âmbitos metropolitanos [29, 30].

Um comparativo entre os grandes grupos de redes sem fio existentes é apresentado na Figura 2. Já na Tabela 1 apresenta-se um comparativo de propriedades entre alguns protocolos de redes sem fio existentes.

Figura 2. Grandes grupos de redes sem fio com as suas respectivas características de alcance e taxa de transmissão [31].

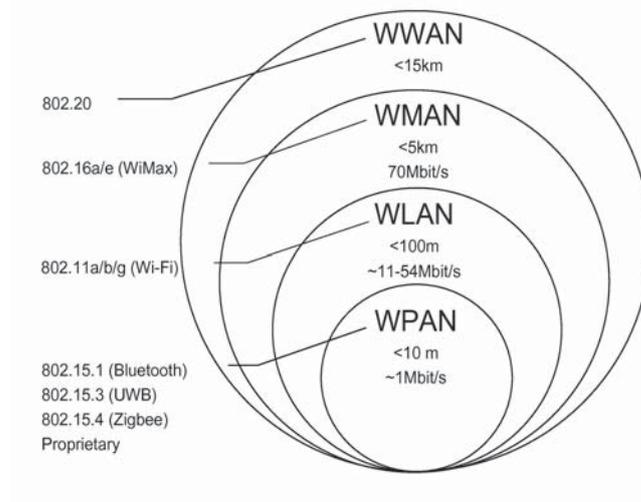
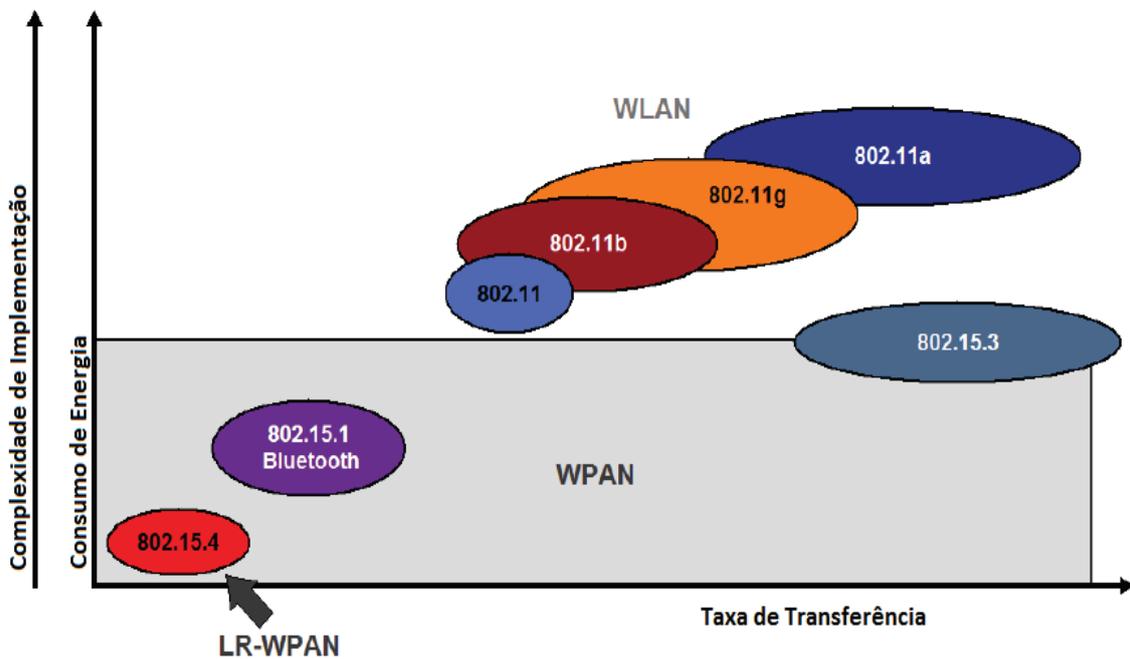


Tabela 1. Comparativo de propriedades entre tecnologias wireless [5].

Parâmetro	802.11g WLAN	Bluetooth WPAN	Zigbee (1mW) LR-WPAN
<b>Banda de Frequência</b>	2.4 GHz	2.4 GHz	868 / 915 MHz e 2.4 GHz
<b>Alcance</b>	10 a 100 metros	10 metros	10 a 100 metros
<b>Taxa de Transferência</b>	2 a 54 Mb/s	1 Mb/s	<= 250 Kb/s
<b>Consumo de Corrente</b>	400 mA (TX) 20 mA (Standby)	60 mA (TX)	25 a 35 mA (TX) 3 uA (Standby)
<b>Tempo Típico de Acesso a Rede</b>	1 s	>3 s	30 ms
<b>Nós por Rede (Máximo)</b>	32 por Acess Point	7	64 K
<b>Número de Canais</b>	13	19	16
<b>Custo / Complexidade</b>	Alto	Médio	Baixo

Na Figura 3, é apresentado um comparativo entre tecnologias inseridas nos grupos WPAN e WLAN, levando em consideração a taxa de transferência de dados, a complexidade de implementação e o consumo de energia.

Figura 3. Comparativo entre tecnologias inseridas nos grupos WPAN e WLAN, comparando a taxa de transferência, a complexidade de implementação e o consumo de energia [5].



### 2.1.3 Wireless Personal Area Network

Com a constante evolução da tecnologia e a capacidade de se desenvolver circuitos eletrônicos cada vez menores, a importância das redes pessoais sem fio tem aumentado significativamente nos últimos anos. As WPAN têm sido muito utilizadas em diversas áreas, tais como automação residencial, industrial, agrícola e comercial, embarcadas em veículos, e para o monitoramento de variáveis ambientais de forma autônoma. Esse tipo de rede apresenta características pervasivas, e está sendo considerada um dos pilares do advento da chamada *Internet of Things* (IoT) [6].

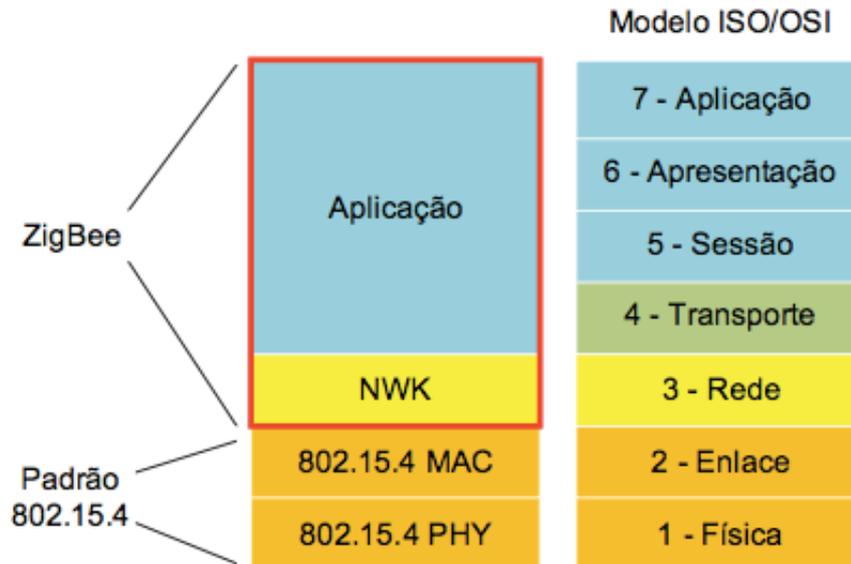
Uma WPAN é descrita como uma rede de curto e médio alcance, de 10 a 100 metros. Em alguns casos, com o uso de amplificadores, pode-se alcançar distâncias maiores, chegando a alguns quilômetros. Todas as tecnologias WPAN são descritas pelo padrão IEEE 802.15 [32]. As duas principais tecnologias contidas nesse padrão são o *Bluetooth* (IEEE 802.15.1) [33] e o IEEE 802.15.4 [34], que é a base do protocolo Zigbee [30, 6].

O protocolo de comunicação Zigbee foi desenvolvido para suprir as necessidades existentes, de um padrão de redes WPAN, que servissem a dispositivos de baixo custo, com consumo de energia reduzido e de baixa complexidade de implementação [5, 6]. Para que esses objetivos fossem alcançados, a taxa de transferência e o alcance dos transceptores foram limitados [5].

O Zigbee é um protocolo de comunicação de redes sem fio constituído por dois padrões distintos desenvolvidos para serem utilizados em conjunto. Os padrões que o compõem são o IEEE 802.15.4, desenvolvido pelo IEEE, e o Zigbee, desenvolvido pela *Zigbee Alliance* [6, 5, 35]. Quando comparados ao modelo ISO/OSI, o padrão IEEE 802.15.4 implementa as duas camadas mais baixas

da rede (MAC e PHY) e o padrão Zigbee é responsável por implementar as camadas superiores (NWK e Aplicação), como mostrado na Figura 4 [36].

Figura 4. Complementação entre os padrões IEEE 802.15.4 e Zigbee, quando comparados com o modelo ISO/OSI [6].



### 2.1.3.1 O padrão IEEE 802.15.4

O padrão IEEE 802.15.4 implementa a camada física (PHY) e a camada de controle e acesso ao meio (MAC). A camada física é responsável pela transmissão e recepção dos dados, ou seja, é a camada que faz a conversão dos pacotes de dados a serem transmitidos em sinais eletromagnéticos, assim como a recepção dos sinais eletromagnéticos e sua conversão em pacotes de dados. É também responsável por prover as funcionalidades de detecção de canais disponíveis, seleção de canais e estimativa de qualidade de conexão do canal [6]. A camada MAC é a responsável pelo gerenciamento da transmissão de dados entre nodos vizinhos, incluindo serviços de reenvio e de técnicas de prevenção de colisão (*Carrier Sense Multiple Access with Collision Avoidance* - CSMA/CA). A camada MAC é responsável ainda por prover os serviços de associação e desassociação de nodos a rede, roteamento, estabelecimento inicial da rede, pela definição da topologia de rede utilizada e pela atribuição de endereço aos nodos presentes na rede [26, 37].

Como pode ser visualizado na Figura 5, o padrão IEEE 802.15.4 determina as faixas de frequências utilizadas, que são 868 MHz (Europa – 1 canal), 915 MHz (Américas – 10 canais) e 2,4 GHz (Global – 16 canais) [29, 5]. A taxa de transferência máxima muda de acordo com a frequência utilizada para a transmissão, conforme apresenta-se na Tabela 2 [5, 6].

O padrão ainda define dois tipos de nodos ou dispositivos físicos que podem existir na rede. O *Full Function Device* (FFD), que pode fazer as funções lógicas de Coordenador PAN, roteador ou Dispositivo Final (*End Device*). E o *Reduced Function Device* (RFD), que somente pode executar a função de Dispositivo Final [5].

Figura 5. Canais e frequências utilizadas pelo padrão IEEE 802.15.4 [5].

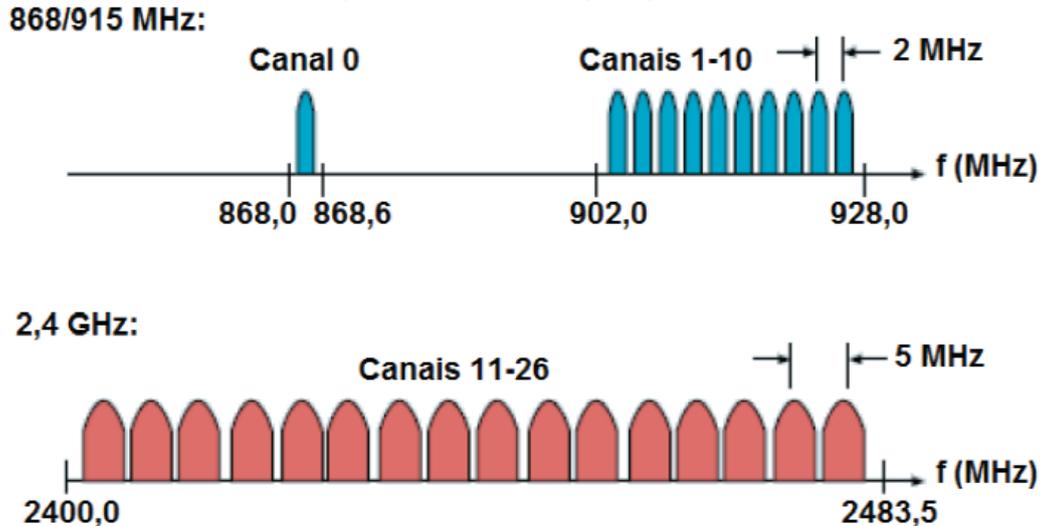


Tabela 2. Taxa de transferência de acordo com a frequência de transmissão [6].

PHY	Banda de Frequência (MHz)	Taxa de Bits (Kb/s)	Taxa de Símbolo	Símbolos
868	868 - 868,6	20	20	Binário
915	902 - 928	40	40	Binário
2450	2.400 - 2.483,5	250	62.5	16 símbolos

Os dispositivos lógicos definidos são três, o Coordenador PAN, o Roteador e o Dispositivo Final. O Coordenador PAN é responsável pela formação da rede, pela definição do canal a ser usado, pela identificação da rede, pela associação de dispositivos Roteadores e Dispositivos Finais, e só pode existir um Coordenador PAN em toda a rede. Os dispositivos Roteadores fornecem caminhos para os dados circularem na rede, podendo definir rotas alternativas quando necessário. Os Dispositivos Finais só podem se comunicar com Coordenadores e Roteadores, podendo funcionar como fonte ou destino de dados. E conforme o padrão IEEE 802.15.4, são os únicos dispositivos que podem entrar em modo de dormência (*sleep*), reduzindo drasticamente o seu consumo de energia [38, 39]. Existem estudos como em [40] que propõem que nodos roteadores também entrem em modo *sleep*, aumentando sua vida útil quando alimentados por baterias.

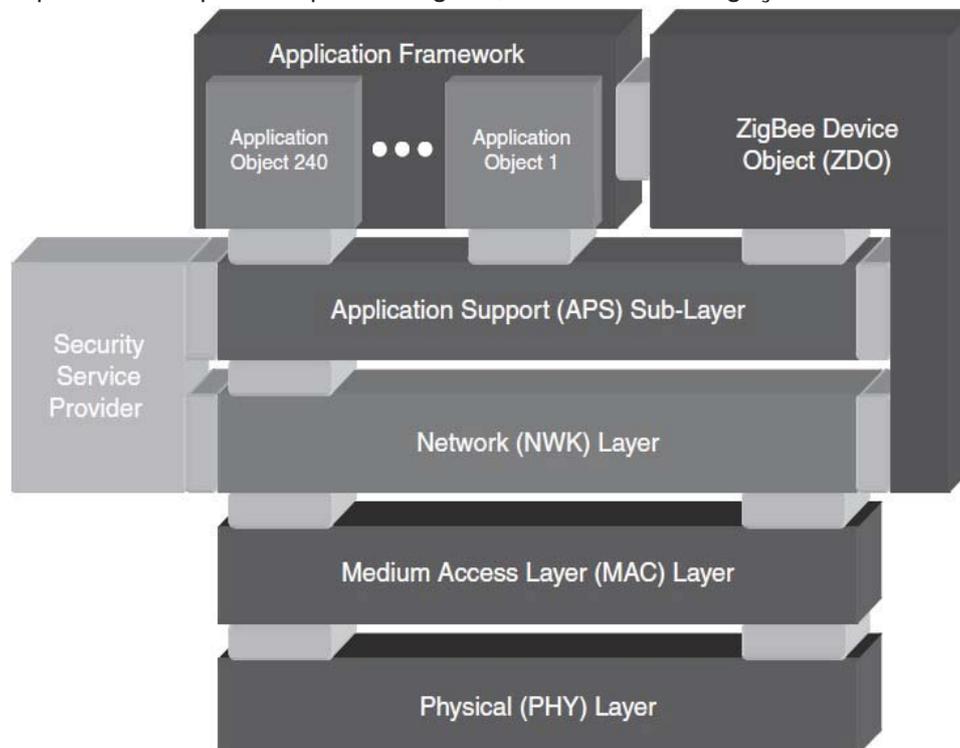
### 2.1.3.2 O Padrão Zigbee

O padrão Zigbee foi desenvolvido pela *Zigbee Alliance* [35], organização sem fins lucrativos aberta a todos que queiram colaborar. É formada por milhares de membros que incluem desde fabricantes de componentes eletrônicos a desenvolvedores de software [36].

Muitas vezes, os padrões Zigbee e IEEE 802.15.4 são descritos como se fossem o mesmo, porém, isso não é correto. Como ilustrado anteriormente na Figura 4, o padrão Zigbee utiliza o padrão IEEE 802.15.4 e implementa as camadas superiores de rede. O padrão IEEE 802.15.4 implementa

somente as camadas inferiores da rede e pode ser utilizado sozinho, sem as camadas superiores [36]. Na Figura 6, é apresentada a arquitetura do padrão Zigbee.

Figura 6. Arquitetura completa do padrão Zigbee, incluindo a interligação entre as camadas [29].



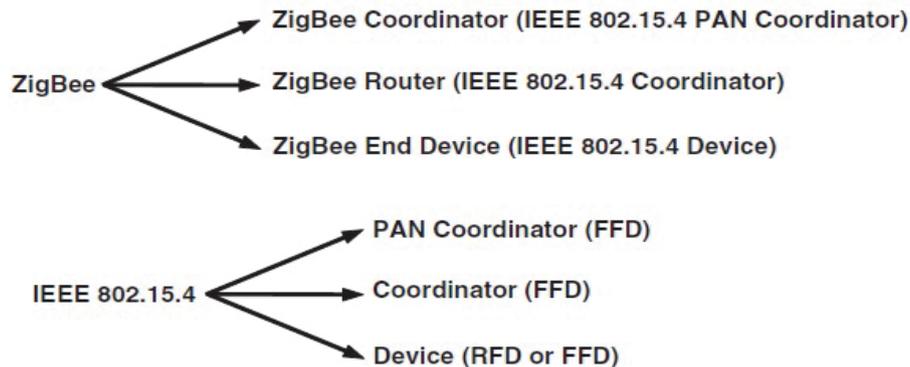
A camada de rede NWK (*Network Layer*) é responsável por garantir a transmissão de dados em múltiplos saltos, pela sincronização dentro de uma rede, pela determinação de rotas, pela verificação da entrega de pacotes, oferece serviços de segurança e criptografia de dados, sendo responsável pela atribuição de endereços. A camada de rede permite implementações da pilha de protocolos que visam balancear os custos de energia das unidades em aplicações específicas, buscando produzir soluções com perfis específicos [5, 29].

A camada de aplicação é dividida em três subcamadas: a subcamada de suporte à aplicação (APS – *Application Support*); a subcamada *Zigbee Device Objects* (ZDO); e a subcamada de Framework de aplicação. A subcamada APS é responsável por fazer a interface entre a camada de aplicação e a camada de rede (NWK) por meio de diversos serviços utilizados por aplicações desenvolvidas pelos fabricantes dos dispositivos e pela subcamada *Zigbee Device Objects*. A subcamada *Zigbee Device Objects* é responsável pela inicialização da subcamada APS e da camada NWK, pela definição do modo de operação do dispositivo (coordenador, roteador ou dispositivo final), pela *device discovery* (determinação de quais serviços de aplicação o dispositivo fornece) e pelo gerenciamento da segurança, e ainda é capaz de descobrir quais serviços de aplicação estão disponíveis em cada nodo [5]. A subcamada Framework de aplicação é responsável por armazenar os *applications objects*, que são responsáveis por controlar a subcamada APS [5, 29].

### 2.1.3.3 Tipos de Topologias de Rede dentro do padrão Zigbee

O padrão Zigbee, assim como o padrão IEEE 802.15.4, pode possuir dois tipos diferentes de dispositivos em sua rede, os dispositivos FFD e os dispositivos RFD. O padrão Zigbee, no entanto, utiliza denominações diferentes quanto à função exercida dentro da rede. Na Figura 7, são apresentadas essas diferenças [36].

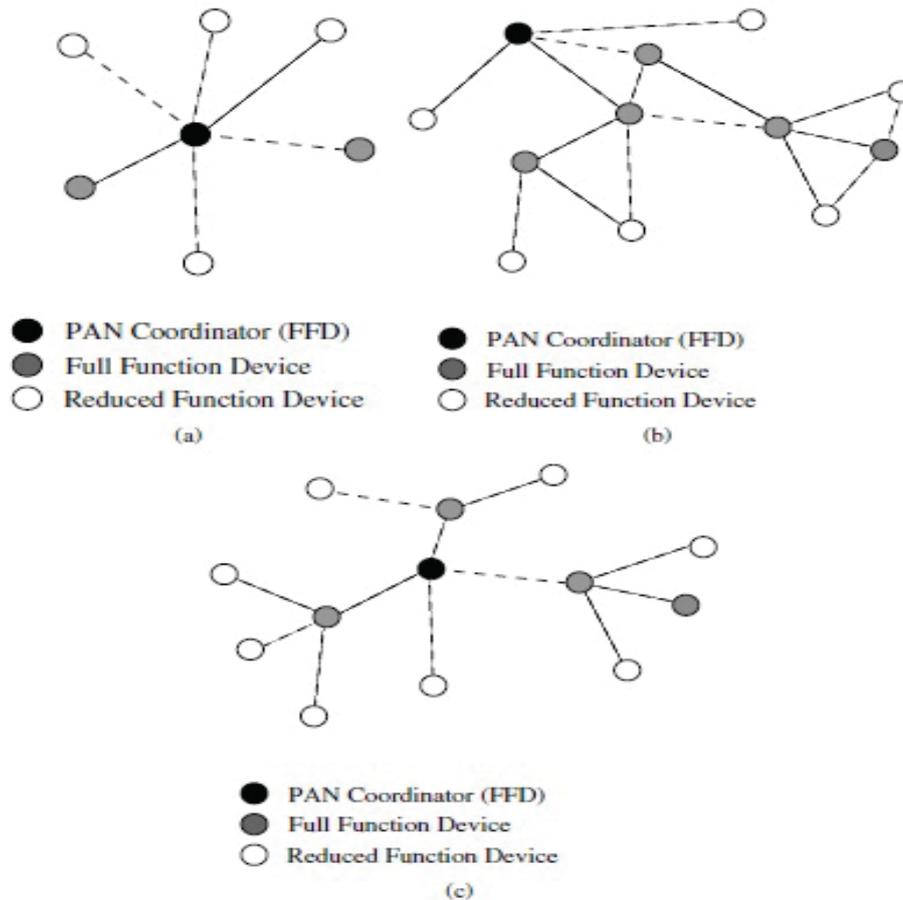
Figura 7. Tipos de dispositivos e funções exercidas nos padrões Zigbee e IEEE 802.15.4 [36].



O padrão IEEE 802.15.4 possibilita a implementação de três diferentes tipos de topologia de rede. Cada topologia possui características distintas, e a escolha da implementação de cada uma delas está diretamente ligada à sua aplicação final. As topologias podem ser [9, 5, 41]:

- *Star* - Na topologia *Star* o coordenador da rede se comunica com um ou mais *End-Devices*. Para um *End-Device* enviar uma mensagem para outro *End-Device*, essa mensagem obrigatoriamente deve ser enviada ao coordenador da rede. Este, por sua vez, a encaminha para o destino final, ou seja, os *End-Devices* não se comunicam entre si. Uma vantagem que essa topologia de rede apresenta é a baixa complexidade de implementação, porém, o alcance da rede fica limitado. Na Figura 8 (a), é apresentada uma rede com topologia *Star*.
- *Cluster-tree* - Na topologia *cluster-tree*, existe um coordenador central se comunicando com roteadores e/ou *End-Devices*. Na Figura 8 (c), é apresentada uma rede com topologia *cluster-tree*. Como a rede nessa topologia pode possuir roteadores, o seu alcance aumenta, assim como a complexidade de implementação. As mensagens dentro de uma rede *cluster-tree* são transmitidas de um nodo a outro, obedecendo a uma hierarquia entre "pai-filho".
- *Mesh* - Na topologia *mesh*, existe um coordenador geral conectado a um ou mais roteadores e/ou *End-Devices*. A principal diferença nessa topologia é que o roteamento das mensagens não é fixo, podendo ser alterado conforme a necessidade pelo algoritmo implementado. Na Figura 8 (b), é demonstrada a implementação de uma rede com topologia *mesh*. Uma das principais vantagens apresentada por essa topologia é a sua capacidade de autorregeneração em caso de falha de algum nodo roteador.

Figura 8. Topologias de rede previstas pelo padrão IEEE 802.15.4, sendo (a) topologia de rede *star*, (b) topologia de rede *mesh* e (c) topologia de rede *cluster-tree* [9]



#### 2.1.4 Sobre o consumo de energia em Redes de Sensores Sem Fio

No projeto de RSSF, o consumo de energia é um dos pontos mais importantes a serem levados em consideração [42]. Alguns projetos recentes [40] têm estudado e proposto meios para a redução do consumo de energia em RSSF. Em diversas aplicações, as RSSF estarão dispostas em ambientes de difícil acesso, em áreas remotas, sendo a utilização de baterias a única fonte de energia possível. Infelizmente, a tecnologia empregada na fabricação de baterias não está progredindo no mesmo ritmo da tecnologia empregada à eletrônica. Assim, muitas vezes o tempo de vida útil de uma RSSF está diretamente ligado ao tempo de vida útil das baterias utilizadas em seus nodos [4, 42].

O consumo de energia de um nodo da rede pode ser dividido em sensoriamento, comunicação e processamento de dados. A energia consumida pela comunicação é muito maior que a energia gasta pelos outros processos. O custo para se transmitir um bit a uma distância de cem metros gasta a mesma quantidade de energia que o processamento de 3000 instruções em um processador com capacidade de processamento de 100 MIPS/W [4, 42]. Assim, o processamento local dos dados deve ser utilizado sempre que possível para diminuir ao máximo o consumo de energia do nodo [42].

O mapeamento sobre a quantidade de energia disponível em cada parte da rede é possível, sendo que essa informação pode auxiliar a prolongar o tempo de vida da rede por meio do uso de protocolos de roteamento específicos [4].

Muitos esforços têm sido empregados em estudos que envolvem desde a arquitetura até o hardware, dos protocolos aos algoritmos, para encontrar soluções que diminuam o consumo de energia em RSSF [42].

Assim, definitivamente, protocolos, algoritmos e aplicações para RSSF devem ser escolhidos levando em consideração o consumo de energia necessário [4].

### 2.1.5 Modo Sleep e Idle

Atualmente, a grande maioria dos microcontroladores disponíveis no mercado apresenta modos de operação diferenciados que visam à redução do consumo de energia. Esses modos são chamados de modo *idle* e modo *sleep*. O modo *idle* é definido como o estado em que o dispositivo fica enquanto não está recebendo, nem transmitindo informações. Nesse modo, todas as funcionalidades do dispositivo estão disponíveis, porém, não estão sendo utilizadas, fazendo com que o consumo seja menor. No modo *sleep*, o dispositivo é forçado a um estado de baixo consumo de energia. Nesse estado, o consumo de energia é muito baixo (geralmente próximo de zero), porém, a grande maioria das funções oferecidas pelo dispositivo não está disponível, sendo necessária uma interrupção externa para que o dispositivo volte a ficar ativo [18, 43].

Um exemplo da capacidade de redução de energia que o modo *sleep* tem pode ser retirado da folha de dados do módulo TELOSB [22], fabricado pela MEMSIC. Esse módulo apresenta um consumo de energia de 24,8  $\mu\text{A}$  em funcionamento normal (com o rádio ligado). Em modo *idle*, este valor é de 21  $\mu\text{A}$ . Já em modo *sleep*, esse valor cai para 6,1  $\mu\text{A}$ , representando uma redução extremamente significativa no consumo de energia.

Em RSSF que monitoram variáveis que não têm os seus valores alterados rapidamente (como temperatura, umidade, entre outras), o modo *sleep* pode ser implementado durante o período de tempo em que não existe a necessidade de leitura dessas variáveis, diminuindo o consumo de energia e aumentando a vida útil do nodo da rede.

### 2.1.6 Módulos e Circuitos Integrados

Com a disseminação das tecnologias de redes sem fio, vários fabricantes têm apresentado soluções que implementam rádios de comunicação sem fio, microcontroladores e conectividade com sensores e atuadores em um único módulo ou circuito integrado. Os módulos e circuitos integrados que fizeram parte da pesquisa deste trabalho foram:

- Módulo TELOSB da MEMSIC;
- Módulo MICAZ da MEMSIC;

- Módulo ZE51/ ZE61 - 2.4 da Telit;
- CI MKW2xDxxx da Freescale;
- CI MC13237 da Freescale;
- CI MC13224V da Freescale;
- CI JN5168-001 da NXP;
- CI AT86RF230 da ATMEL;
- CI CC2531 da Texas Instruments;
- CI CC2538 da Texas Instruments; e
- CI CC2630 da Texas Instruments.

Apresenta-se, na Tabela 3, um comparativo entre os módulos e os circuitos integrados, levando em consideração o consumo de energia para transmissão e recepção de dados, o consumo de energia em modo *sleep*, a potência de saída e a distância de transmissão alcançada estimada pelo fabricante.

Tabela 3. Comparativo entre Características de módulos e Circuitos Integrados.

<b>Módulo / CI</b>	<b>Consumo Transmissão</b>	<b>Consumo Recepção</b>	<b>Consumo Sleep</b>	<b>Potência Máxima de Saída</b>	<b>Distância Alcançada</b>
<b>TELOSB</b>	-	23 mA	6,1 uA	0 dBm	100 m
<b>MICAZ</b>	17,4 mA	19,7 mA	16 uA	0 dBm	100 m
<b>ZE51-2.4</b>	35 mA	26 mA	1 uA	4 dBm	1 Km
<b>ZE61-2.4</b>	150 mA	31 mA	1,5 uA	20 dBm	4 Km
<b>MKW2xDxxx</b>	19.5 mA	-	-	8 dBm	-
<b>MC13237</b>	28,2 mA	35 mA	-	2 dBm	-
<b>MC13224V</b>	29 mA	21 mA	-	4 dBm	-
<b>JN5168-001</b>	175 mA	22 mA	-	22 dBm	6 Km
<b>AT86RF230</b>	16,5 mA	15,5 mA	20 nA	3 dBm	-
<b>CC2531</b>	29 mA	24 mA	1 uA	3 dBm	-
<b>CC2538</b>	24 mA	20 mA	1,3 uA	7 dBm	-
<b>CC2630</b>	9,1 mA	5,9 mA	100 nA	5 dBm	-

## 2.2 A PLATAFORMA ARDUINO

Arduino é uma plataforma de prototipação *Open-Source*, baseada principalmente na facilidade de uso, tanto de software quanto de hardware. As placas Arduino apresentam entradas e saídas, por meio das quais podem ser conectados diversos tipos de sensores (analógicos e/ou digitais), sendo os valores lidos desses sensores processados, gerando saídas. Uma saída pode ser a ativação de

uma lâmpada, o controle de um motor, ou até mesmo o envio de informações para um servidor. A plataforma é totalmente programável, para isso, é utilizada a linguagem de programação Arduino, que é baseada em Wiring [44], e IDE Arduino, que é baseada em Processing [45, 7].

A plataforma Arduino surgiu de um grande desafio. Em 2002, o professor Massimo Banzi do *Interaction Design Institute Ivrea* (IDII) tinha como desafio ensinar os seus alunos a criarem sistemas eletrônicos, mas isso tinha que ser feito com tempo limitado de aula e orçamento reduzido [46].

Como muitos naquela época, Banzi utilizava a plataforma *BASIC Stamp*, desenvolvida pela *California Company Parallax*. Essa plataforma era composta por componentes básicos, fonte de alimentação, microcontrolador, memória e portas de entrada e saída para a conexão com hardware externo, sendo utilizada a linguagem *BASIC*. Era amplamente utilizada por engenheiros já há quase uma década. Mas o *BASIC Stamp* apresentava algumas limitações, como não possuir poder computacional suficiente para alguns projetos que precisavam ser desenvolvidos. O custo da plataforma não era acessível, podendo custar US\$ 100,00, e Banzi precisava de uma plataforma que rodasse em computadores Macintosh, que eram onipresentes entre os designers da IDII [46].

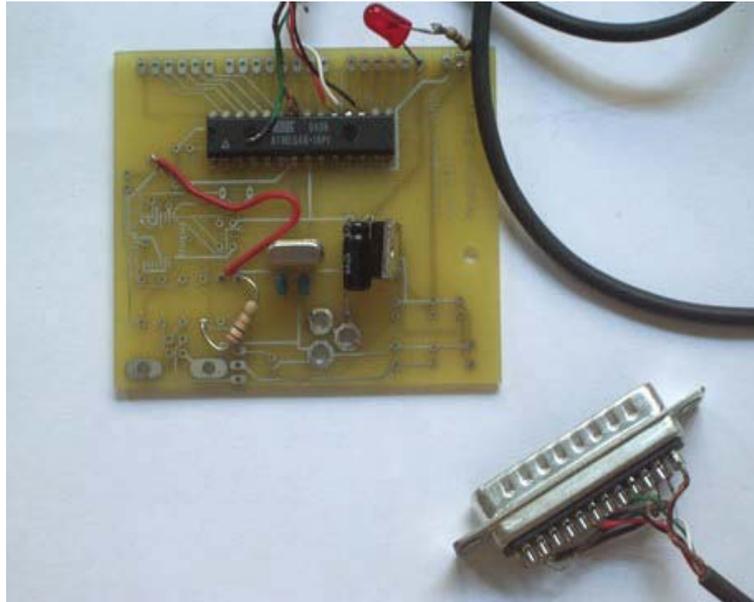
No *Massachusetts Institute of Technology* (MIT), foi desenvolvida uma linguagem de programação amigável chamada *Processing*. Essa linguagem ganhou popularidade rapidamente por permitir que até programadores inexperientes criassem complexas visualizações gráficas de dados. Uma das razões do seu sucesso foi o ambiente de desenvolvimento integrado (IDE) extremamente fácil de usar. Tendo como base esse trabalho, o professor Massimo Banzi tinha o desafio de criar uma ferramenta de software semelhante, mas aplicada à programação de microcontroladores [46].

Um estudante já havia dado alguns passos nessa direção, Hernando Barrágan, desenvolvendo uma plataforma de prototipação chamada *Wiring*, que possuía tanto uma IDE amigável quanto uma placa de circuito impresso *ready-to-use*. Esse projeto existe até hoje, mas a ideia do professor Massimo Banzi era maior, ele queria uma plataforma ainda mais simples, mais barata e mais fácil de usar [46].

O primeiro protótipo, apresentado na Figura 9, foi construído em 2005, sendo um projeto simples e ainda sem nome. O Professor Massimo Banzi e seus colaboradores, sempre acreditaram nos projetos de software *open-source*. Como o objetivo do projeto em desenvolvimento era criar uma plataforma acessível, eles acreditavam que seria melhor abrir o projeto para o maior número possível de pessoas, ao invés de mantê-lo fechado. Um segundo fator que impulsionou ainda mais a decisão de criar o projeto *open-hardware* foi que o *Interaction Design Institute Ivrea* estava passando por uma crise, correndo o risco de fechar suas portas. Temendo que isso ocorresse, a equipe resolveu tornar o projeto *open-source* [46].

O modelo *open-source* já tinha sido amplamente utilizado para alimentar o desenvolvimento e a inovação em software, mas nunca tinha sido utilizado em hardware. Foi necessário encontrar uma solução de licenciamento que pudesse ser aplicada ao hardware desenvolvido. Após uma pesquisa, acabaram utilizando uma licença da *Creative Commons*, um grupo sem fins lucrativos cujas licenças são normalmente utilizados em obras culturais, como música, por exemplo. O pensamento voltava-se a enxergar o hardware como parte de cultura que se desejava compartilhar com outras pessoas [46].

Figura 9. Primeiro protótipo da plataforma Arduino [46].



O produto criado (Arduino) consiste de componentes baratos e que podem facilmente ser encontrados no mercado, sendo um atrativo muito grande para os usuários que pretendem desenvolver o seu próprio hardware baseando-se no Arduino. Mas, uma das características mais importantes apresentadas por essa plataforma é a de que ela é *plug-and-play*, algo que o usuário pode tirar de uma caixa, ligar a um computador e utilizá-lo imediatamente, independente do sistema operacional utilizado pelo usuário [46].

A filosofia do Arduino é a de que se o usuário quiser aprender eletrônica, ele deve ser capaz de fazer isso na prática desde o primeiro dia de estudos, ao invés de aprender toda a parte teórica primeiro [46].

Muito do sucesso da plataforma Arduino se deu devido a existência prévia da IDE *Processing* e da plataforma *Wiring*. Esses projetos deram ao Arduino um de seus pontos fortes, um ambiente de programação de fácil utilização. Antes disso, o aprendizado necessário para a programação de microcontroladores era difícil. Com o Arduino, mesmo pessoas sem experiência anterior com eletrônica ganharam acesso a um mundo de hardware anteriormente impenetrável. Para o professor Massimo Banzì, este é, talvez, o impacto mais importante do Arduino: a democratização da engenharia [46].

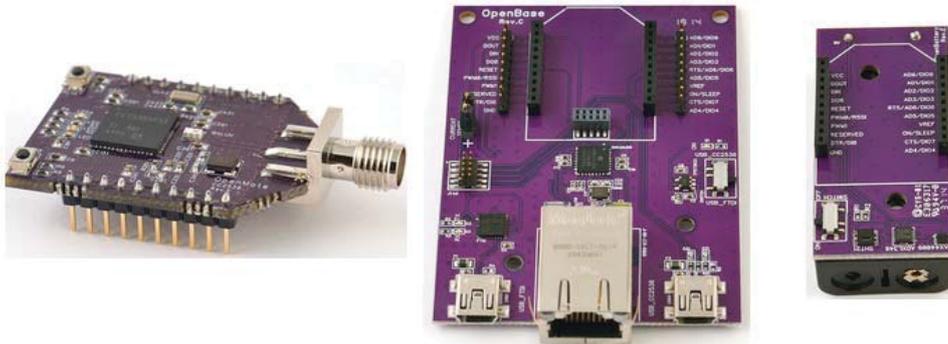
### 2.3 TRABALHOS RELACIONADOS

Nesta subseção, são apresentados trabalhos relacionados a esta pesquisa. A plataforma *OpenMote* utiliza o SoC CC2538 como base. Já o padrão de comunicação sem fio *LoraWAN* se apresenta como uma alternativa ao padrão Zigbee.

### 2.3.1 OpenMote

A plataforma *OpenMote* é uma proposta de hardware open-source com foco na Internet das coisas. A plataforma foi desenvolvida por Pere Tuset-Peiró e Xavier Vilajosana, sendo composta por três módulos: *OpenMote-CC2538*; *OpenBase*; e *OpenBattery*. O módulo *OpenMote-CC2538* é o coração da plataforma, ele utiliza o *System on Chip CC2538* da *Texas Instruments* que possui um microcontrolador ARM Cortex M3 de 32 bits e um transceiver de rádio. O rádio opera na banda de 2,4 GHz e é totalmente compatível com o padrão IEEE 802.11.4-2006. O módulo *OpenBase* possui três funções: a primeira é permitir a programação e a depuração do código, com suporte a *breakpoints*, sendo feita por meio de um conector padrão JTAG (*Joint Test Action Group*) de 10 pinos; a segunda consiste em permitir a comunicação com um computador por meio da porta serial ou USB; e a terceira volta-se a permitir a comunicação com uma porta ethernet 10/100 Mbps, possibilitando ao *OpenMote* conectar-se à internet sem a necessidade de conexão com um computador. O módulo *OpenBattery* basicamente possibilita que o módulo seja alimentado por meio de baterias, trazendo autonomia ao sistema. Na Figura 10, estão ilustrados os módulos *OpenMote-CC2538*, *OpenBase* e *OpenBattery* respectivamente [47].

Figura 10. Módulos da plataforma *OpenMote* [47].



Essa plataforma também suporta alguns sistemas operacionais open-source especialmente desenvolvidos para a Internet das Coisas, tais como o *Contiki* [48] o *OpenWSN* [49]. Outros sistemas operacionais suportados pela plataforma são o *FreeRTOS* [50] e o *RiOT* [51], os quais são sistemas operacionais de tempo real especialmente desenvolvidos para sistemas embarcados e a internet das coisas, respectivamente [47].

### 2.3.2 LoraWAN

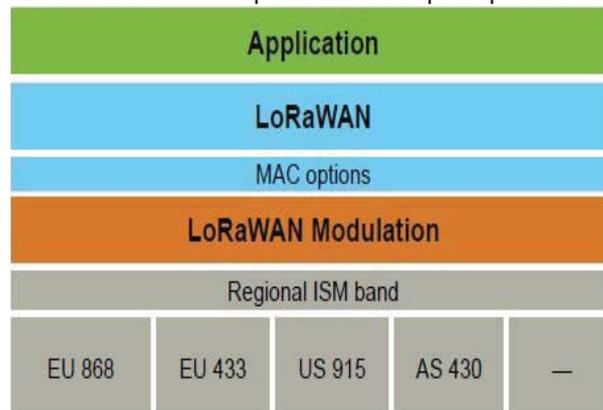
Muitos sistemas de comunicação sem fio utilizam codificação *frequency shifting keying* (FSK), ou modulação por chaveamento de frequência em sua camada física. Esse tipo de modulação é muito eficiente para sistemas que objetivam o baixo consumo de energia. *Long Range* (Lora) é a camada física responsável pela modulação sem fio utilizada pelo protocolo LoraWAN. O Lora baseia-se na modulação *chirp spread spectrum* (CSS), que mantém as mesmas características de baixa potência que

a modulação FSK, mas aumenta significativamente o alcance. A modulação CSS tem sido utilizada há muitos anos em comunicações espaciais e militares, devido ao seu longo alcance de comunicação e à robustez contra interferências, entretanto Lora é a primeira implementação de baixo custo para uso comercial [52, 53].

O padrão LoraWAN define o protocolo de comunicação e a arquitetura do sistema, enquanto a camada física Lora habilita o link de comunicação de longo alcance. O protocolo e a arquitetura de rede possuem maior influência na determinação da duração da bateria de um nodo, da capacidade da rede, da qualidade do serviço, da segurança e da variedade de aplicativos atendidos pela rede [52, 53].

Na Figura 11, está ilustrada a arquitetura de rede implementada pelo padrão LoraWAN, que é composta basicamente por quatro camadas: *Application*; *LoRaWAN*; *MAC*; e *LoRa* ou *LoRaWAN Modulation* [52, 53].

Figura 11. Camadas de rede implementadas pelo padrão LoraWAN [52].

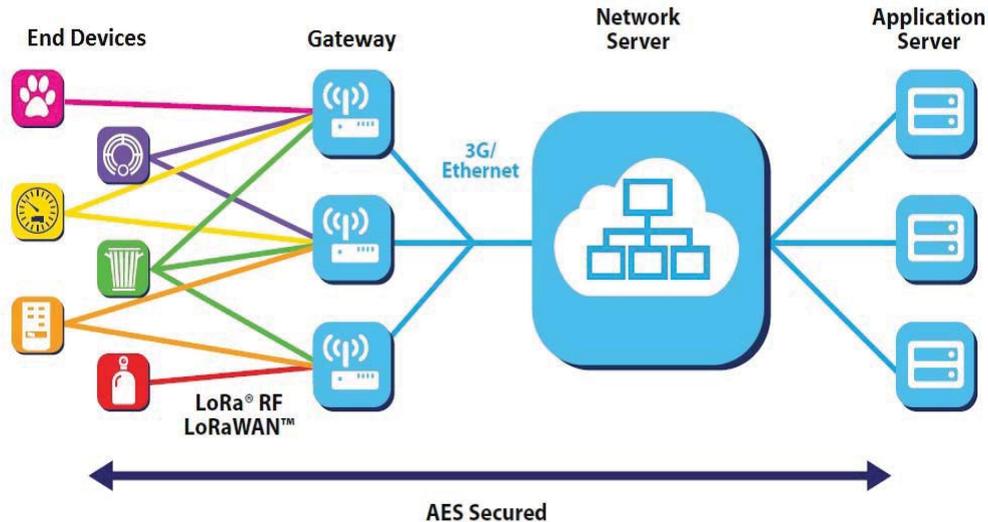


Com as tecnologias atuais, muitas redes são implementadas com a topologia *mesh*. Em uma rede *mesh*, os nodos da rede encaminham informações entre eles para aumentar o alcance da rede. Embora isso aumente muito o alcance da rede, também aumenta a complexidade de implementação da rede e reduz o tempo de vida útil dos nodos quando estes são alimentados por baterias, pois eles têm que retransmitir informações de outros nodos até o seu destino final. Em termos de conservação de energia em nodos alimentados por baterias, a utilização da topologia de rede *star* é mais indicada quando longas distâncias podem ser alcançadas pelos dispositivos utilizados [52, 53].

Para que uma rede de longo alcance, utilizando a topologia *star*, se torne viável, o *gateway* deve possuir a capacidade de receber e tratar os dados de muitos nodos da rede. As redes que utilizam o padrão LoraWAN possuem essa capacidade. Isso é alcançado por meio de uma taxa de dados adaptável e utilizando um transceiver multicanal no *gateway*, de modo que os dados simultâneos e em múltiplos canais possam ser recebidos [52, 53].

Na Figura 12, é ilustrado um exemplo de uma rede utilizando o padrão LoraWAN. No padrão LoraWAN, um *end-device* não se comunica somente com um *gateway* dentro da rede, vários *gateways* podem receber e tratar os dados enviados por um *end-device*. Isso garante uma maior segurança na transmissão dos dados [52, 53].

Figura 12. Um exemplo de implementação de rede utilizando o padrão LoraWAN [52].



Em uma rede LoraWAN, os nodos são assíncronos, ou seja, eles se comunicam com os *gateways* somente quando possuem dados prontos para serem transmitidos. Esse tipo de protocolo é normalmente referido como método *Aloha*. Em redes *mesh* ou em uma rede síncrona, os nodos frequentemente precisam se sincronizar com a rede para verificar e enviar dados. Essa sincronização consome uma energia significativa e, dependendo da aplicação, pode ser responsável pela maior parte do consumo de energia do nodo [52, 53].

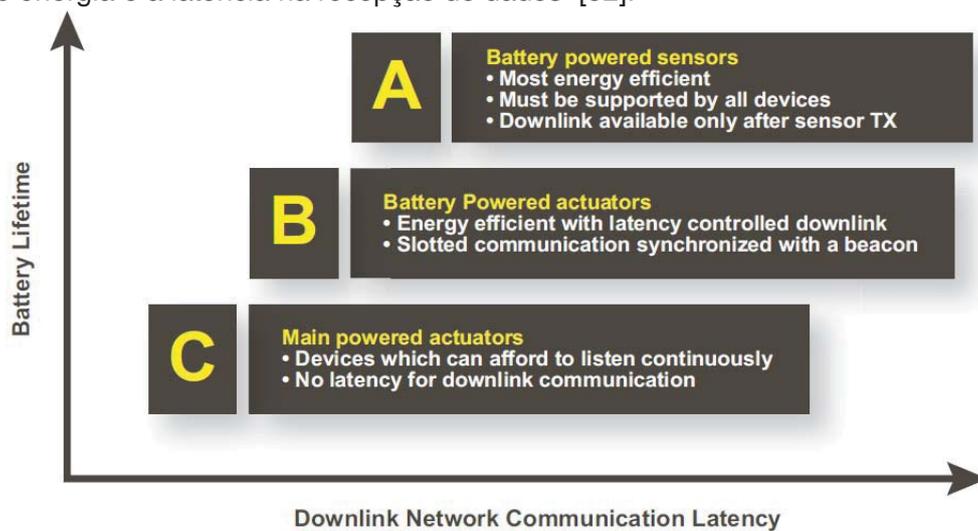
O padrão LoraWAN prevê três classes diferentes de dispositivos [52, 53]:

- Classe A: são dispositivos finais bidirecionais. Só podem receber dados após a transmissão de alguma informação para o *gateway*. Essa classe representa os dispositivos de mais baixo consumo de energia, os quais são normalmente utilizados em aplicações que não precisam receber informações da rede;
- Classe B: são dispositivos finais bidirecionais com janelas de recepção de dados programadas. Utiliza *beacons* para sincronização e sinalização da rede, por meio desses, o *gateway* reconhece a janela de comunicação aberta com o dispositivo final. Devido ao envio de *beacons*, o consumo de energia dessa classe de dispositivos é maior; e
- Classe C: são dispositivos finais bidirecionais, com janela de recepção de dados aberta quase que continuamente, fechadas apenas durante a transmissão de dados pelo nodo. Essa classe de dispositivos apresenta um consumo de energia superior, destinada a dispositivos finais que possuam uma fonte de alimentação não limitada.

Na Figura 13, é apresentada uma comparação entre as classes de dispositivos existentes no padrão LoraWAN, levando em consideração o consumo de energia e a latência na recepção de dados apresentada pelos dispositivos.

O padrão LoraWAN é mantido pela associação aberta e sem fins lucrativos *Lora Alliance*, que foi iniciada por líderes da indústria mundial. Tem a missão de padronizar a *Low Power Wide Area*

Figura 13. Comparação entre classes de dispositivos utilizando padrão LoraWAN considerando o consumo de energia e a latência na recepção de dados [52].



Network (LPWAN). Visa permitir o desenvolvimento de aplicações para IoT, M2M, *Smart Cities* e aplicações industriais. É um padrão extremamente novo, sendo que a versão 1.0 de suas especificações foi liberada em 2015, estando atualmente na versão 1.0.2 [52]. Mas, já podem ser encontrados no mercado módulos e sistemas de desenvolvimento que utilizam esse padrão, como pode ser visto em [54, 55, 56].

## 2.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo, foram revisados conceitos sobre redes de sensores sem fio, incluindo tecnologias e padrões de transmissão de dados utilizados atualmente. Outro ponto abordado foi o consumo de energia e modos de operação de dispositivos alimentados por baterias. Além disso, foram apresentados alguns trabalhos e produtos que de alguma forma estão relacionados a esta pesquisa.

Um padrão que, apesar de ser novo, surge forte no mercado, é o LoraWAN, o qual promete prover comunicação entre dispositivos e, em casos específicos, alcançar quilômetros de distância. Possui um consumo tão baixo a ponto de estimar a vida útil de baterias por até 20 anos [52].

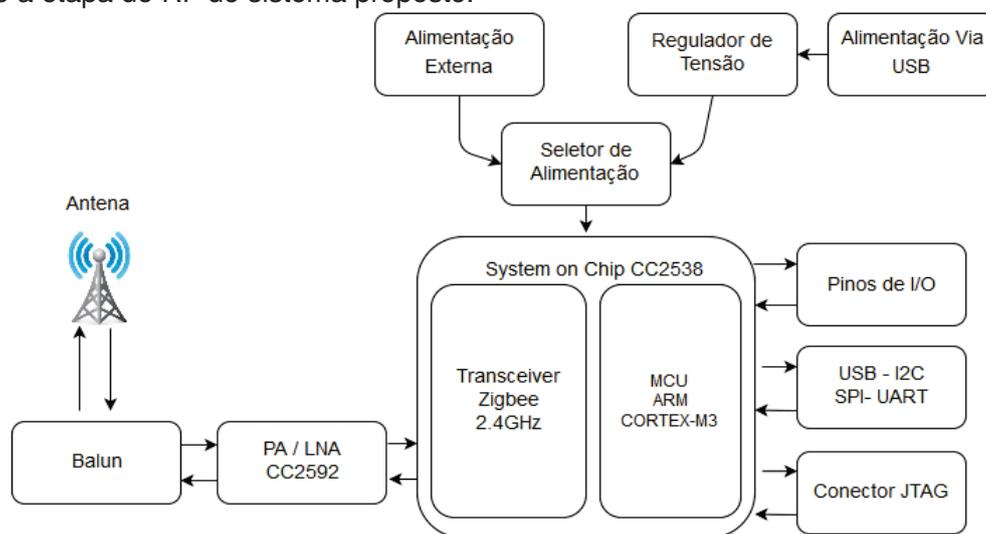
Porém, o padrão Zigbee é o mais utilizado para o desenvolvimento de redes sem fio que não necessitem de uma taxa de transferência de dados muito alta. Algumas características são desejadas quando se trata de módulos para o desenvolvimento de novas aplicações: baixo consumo de energia; confiabilidade do sistema e baixo custo.

A maior dificuldade enfrentada é a falta de uma plataforma de apoio ao estudo e desenvolvimento de dispositivos que agreguem as características desejadas. As plataformas disponíveis no mercado não satisfazem em partes ou totalmente essas necessidades. Vivemos o início do advento da Internet das Coisas, dessa forma, a existência de uma plataforma de apoio que permita a qualquer perfil de usuário desenvolver de uma forma fácil e rápida suas ideias torna-se um produto com grande potencial.

### 3. SOBRE O HARDWARE

Proceder ao desenvolvimento de um hardware personalizado foi uma decisão tomada durante a etapa de pesquisa do trabalho. Dentro de um conjunto de características desejadas, as mais relevantes para a tomada dessa decisão foram a redução do consumo de energia, a possibilidade de personalização do *firmware* embarcado e a redução no custo final do hardware desenvolvido. Na Figura 14, é apresentado um diagrama de blocos, que mostra as principais características do hardware do sistema proposto.

Figura 14. Diagrama em blocos, apresentando os dispositivos de entrada e saída, o sistema de alimentação e a etapa de RF do sistema proposto.



Neste capítulo, são apresentadas informações sobre todo o conjunto de hardware escolhido para o desenvolvimento deste projeto.

#### 3.1 ESCOLHA DO MICROCONTROLADOR

Como mostrado na Tabela 3, na subseção 2.1.6, durante a etapa de pesquisa, foram estudadas diversas opções de módulos e circuitos integrados disponíveis no mercado, que poderiam ser utilizados no desenvolvimento desse projeto. As características desejadas foram elencadas para auxiliar na escolha do sistema a ser utilizado, tendo sido levadas em consideração as seguintes:

- A - consumo de energia: para o levantamento dessa característica, pode-se dividir o consumo de energia basicamente em três modos diferentes, o consumo de energia quando da transmissão de dados, o consumo de energia quando da recepção de dados e o consumo de energia em modo *sleep*;

- B - potência máxima de saída: essa característica está diretamente ligada à distância de transmissão que o módulo poderá alcançar, deve ser levada em consideração por representar uma limitação de uso dos módulos desenvolvidos;
- C - custo: é importante levar em consideração o custo de desenvolvimento para viabilizar aplicações onde vários nodos de rede necessitem ser utilizados;
- D - necessidade de uso de mais de um microcontrolador: diversos módulos disponíveis no mercado não permitem a personalização de seu *firmware* embarcado, sendo necessária a utilização de um microcontrolador extra para a configuração de operações pré-programadas;
- E - tipo de encapsulamento do microcontrolador: alguns encapsulamentos existentes dificultam muito a montagem de protótipos, como por exemplo o encapsulamento *Ball Grid Array* (BGA), que exige equipamentos específicos para a sua soldagem em placas de circuito impresso (PCI);
- F - suporte ao desenvolvimento de hardware pelo fabricante: alguns fabricantes de microcontroladores disponibilizam informações técnicas e circuitos modelos para o apoio ao desenvolvimento de novos hardwares personalizados, além da venda de módulos para testes;
- G - disponibilização da Pilha Zigbee: o fabricante do hardware utilizado deve fornecer a pilha de protocolos Zigbee utilizada pelo microcontrolador, viabilizando o desenvolvimento de *firmwares* personalizados;
- H - facilidade de aquisição: esse também é um ponto importante, pois frequentemente materiais difíceis de serem adquiridos são pouco utilizados, possuem poucas informações técnicas disponíveis, e têm uma vida útil de mercado mais curta;

Levando em consideração todos os pontos listados, foi possível montar a Tabela 4, na qual foram atribuídos valores a cada característica de cada sistema analisado. Ao final, foi somado o resultado total para cada sistema, e o resultado encontrado foi levado em consideração no momento da escolha do microcontrolador que seria utilizado no desenvolvimento do projeto. Todas as informações levadas em consideração nesse levantamento foram obtidas diretamente dos *DataSheets* fornecidos pelos fabricantes, dos sites oficiais dos fabricantes ou em fornecedores de componentes eletrônicos com elevado grau de confiabilidade, tais como [57, 58].

Os sistemas MC13224V e AT86RF230 ficaram sem os valores finais, em virtude de que, em uma pesquisa recente, constatou-se que eles não eram mais fabricados, não sendo recomendado, portanto o seu uso em novos projetos.

Analisando os resultados obtidos, optou-se utilizar o *System on Chip* CC2538 da *Texas Instruments*. Essa se escolha se justifica em razão de que, dentre os sistemas estudados, esse foi o que apresentou maior pontuação. Além disso, outra característica apresentada pelo SoC CC2538 é a possibilidade de utilização conjunta com o amplificador CC2592, também fabricado pela *Texas Instruments*. O CC2592 amplifica a potência de saída do sinal e, conseqüentemente, aumenta a distância de comunicação alcançada pelo sistema.

Tabela 4. Valoração das características de cada módulo ou sistema estudado.

<b>Característica / Sistema analisado</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>TOTAL</b>
<b>TELOSB</b>	8,5	5	3	7	10	7	5	5	50,5
<b>MICAZ</b>	9	5	3	7	10	7	5	4	50
<b>ZE51-2.4</b>	7	6	4	5	9	5	5	6	47
<b>ZE61-2.4</b>	3	9,5	5	5	9	5	5	6	47,5
<b>MKW2xDxxx</b>	7,5	8	9	10	8	7	9	10	68,5
<b>MC13237</b>	8	6	8	10	4	7	9	10	62
<b>MC13224V</b>	8	7	-	10	4	7	9	0	-
<b>JN5168-001</b>	3	10	6	5	9	7	5	10	55
<b>AT86RF230</b>	9	5,8	-	10	8	7	8	0	-
<b>CC2531</b>	8	5,8	9	10	8	8	8	10	66,8
<b>CC2538</b>	8	8	9	10	8	9	9	10	71
<b>CC2630</b>	10	7	10	10	8	5	9	10	69

### 3.1.1 System on Chip CC2538

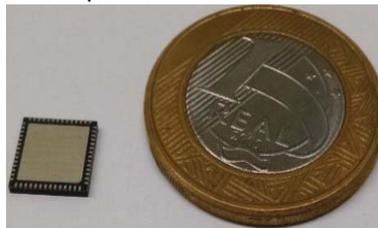
O SoC CC2538 possui um transceiver de rádio integrado de 2.4 GHz totalmente compatível com o padrão IEEE 802.15.4. Esse transceiver apresenta excelente sensibilidade na recepção de dados (-97dBm), permite a programação via software da potência de saída (até 7dBm), apresenta baixo consumo de energia, consumindo 20mA em modo de recepção de dados, 24mA em modo de transmissão de dados a 0dBm (zero) e 1,3uA em modo *sleep*. Também oferece suporte à segurança (AES-128/256). Pode oferecer 128KB, 256KB ou 512KB de memória *flash*, variando conforme o modelo, e possui 32KB de memória RAM. Além dessas, o SoC CC2538 ainda oferece as seguintes funcionalidades e características [18, 59, 60]:

- velocidade de clock de até 32MHz;
- *real time clock* integrado;
- suporta *On-Chip Over-the-Air upgrade*;
- suporte ao padrão Zigbee;
- quatro *timers* de 32 bits de uso geral;
- um *sleep timer* de 32 bits, 32KHz ;
- conversor A/D de 12 bits com 8 canais ;
- monitor de bateria e sensor de temperatura;
- USB 2.0 *Full-Speed*;
- 2 x SPI;

- 2 x UART;
- 1 x I2C;
- 32 pinos de I/O de uso geral (28 x 4mA e 4 x 20mA);
- *watchdog timer*; e
- alimentação de 2 a 3,6 Volts.

Outra característica interessante que o SoC CC2538 apresenta é o seu tamanho reduzido. Suas dimensões são: 8 mm x 8 mm. Isso possibilita o desenvolvimento de circuitos/produtos finais extremamente compactos. Na Figura 15, é apresentado o SoC CC2538 comparado, em tamanho, a uma moeda.

Figura 15. SoC CC2538 comparado, em tamanho, a uma moeda. O tamanho reduzido do SoC CC2538 possibilita o desenvolvimento de produtos finais extremamente compactos.

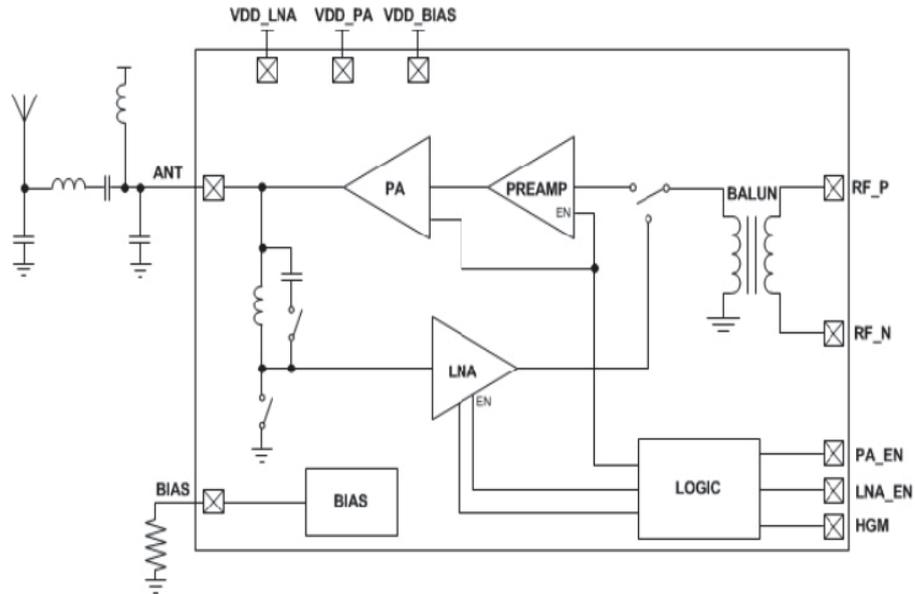


### 3.2 RANGE EXTENDER CC2592

Em determinadas aplicações, a distância entre dois dispositivos pode chegar a algumas centenas de metros. Para esse tipo de aplicação, o SoC CC2538 por si só não oferece suporte. Nessas situações, o uso de um amplificador se faz necessário, mas, como o rádio do SoC CC2538 opera em uma frequência de 2,4GHz, alguns cuidados devem ser tomados na escolha do amplificador utilizado. Dessa forma, o mais sensato foi utilizar o *Range Extender* CC2592, também fabricado pela *Texas Instruments*. O CC2592 é projetado especialmente para trabalhar em conjunto com a família de SoC CC25XX da *Texas Instruments* [61].

O *range extender* CC2592 é um *front-end* de alto desempenho para SoCs de baixo consumo de energia e baixa tensão. Para aumentar o alcance de transmissão e recepção de dados, o CC2592 provê um *power amplifier* (PA) e um *low noise amplifier* (LNA). Possui tamanho extremamente reduzido (4mm x 4mm) e componentes integrados, como *switches*, *RF-matching* e *balun*, para simplificar e facilitar a sua utilização. Apresenta consumo de energia de 4mA para recepção de dados, 155mA para transmissão de dados e 100nA em *power down*. Possui faixa de alimentação entre 2 e 3,6 Volts [62]. Na Figura 16, é mostrado o diagrama elétrico simplificado do *range extender* CC2592.

Figura 16. Diagrama elétrico simplificado do *range extender* CC2592 [62].



### 3.3 MÓDULOS LOW POWER E HIGH POWER

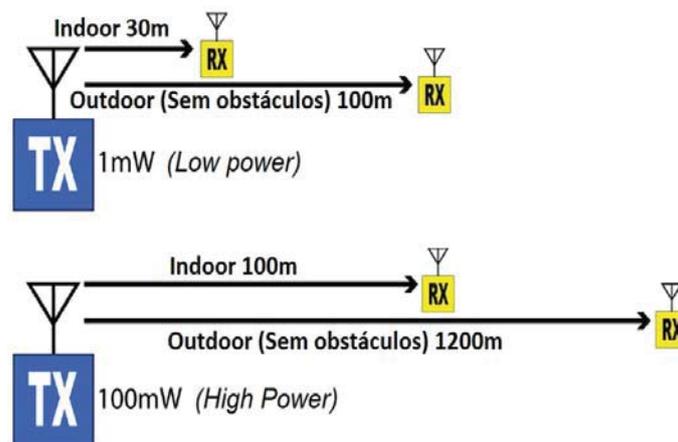
Os termos *low power* e *high power* fazem referência à potência de transmissão do módulo. Módulos *high power* podem apresentar consumo de energia muito superior ao dos módulos *low power*, superando a faixa dos 100mA, mas, ainda assim, podem se encaixar dentro dos padrões IEEE 802.15.4 e Zigbee como dispositivos de baixo consumo de potência [5, 34, 35].

Para os padrões IEEE 802.15.4 e Zigbee, dispositivos de baixo consumo de potência são dispositivos que possuem a capacidade de entrar em modo *sleep*, reduzindo, com isso, o seu consumo de energia. Em relação à potência de transmissão dos dispositivos, os padrões não estabelecem um valor máximo, somente o valor mínimo de -3dBm. Com isso, módulos *high power* podem ser considerados dispositivos com baixo consumo de potência [5, 34, 35].

Ao utilizarmos o SoC CC2538 em conjunto com o *range extender* CC2538, criamos um dispositivo *high power*, com capacidade de transmissão de dados de algumas centenas de metros, mas também com a possibilidade de redução de seu ciclo de trabalho por meio do modo *sleep*. Além disso, pode-se alterar a potência de saída do SoC CC2538 via software, reduzindo o seu consumo e transformando-o em um dispositivo *low power*.

Os dispositivos *high power* são de extrema importância na formação de redes Zigbee, principalmente em aplicações nas quais a distância entre os nodos é maior. Em vez de utilizar vários nodos para alcançar a mesma distância (em redes que possibilitam múltiplos saltos), utiliza-se apenas dois nodos com capacidade de transmissão igual à distância desejada. Na Figura 17, é ilustrada uma aproximação entre as distâncias de transmissão que podem ser alcançadas entre os módulos *low power* e *high power*.

Figura 17. Uma aproximação entre as distâncias de transmissão alcançadas entre os módulos *low power* e *high power* [5].



### 3.4 ANTENA ON-BOARD DE 2.4GHZ

Levando em consideração a facilidade de uso e a redução de custos, foi utilizada uma antena *on-board*, impressa no próprio circuito do módulo. Para a frequência de 2.4GHz, a *Texas Instruments* fornece sete diferentes modelos de referência, que podem ser encontrados em [63]. Foram selecionados inicialmente dois modelos de antena. A primeira antena estudada foi a *Meandered Inverted-F Antenna* (MIFA) [64], a qual representa a solução mais compacta. É utilizada principalmente em soluções com limitações de espaço na PCI. A segunda opção estudada foi a *Inverted-F Antenna* (IFA) [65], cuja implementação requer mais espaço da PCI, porém, ela é omnidirecional, possui melhor eficiência e maior largura de banda quando comparada à antena MIFA.

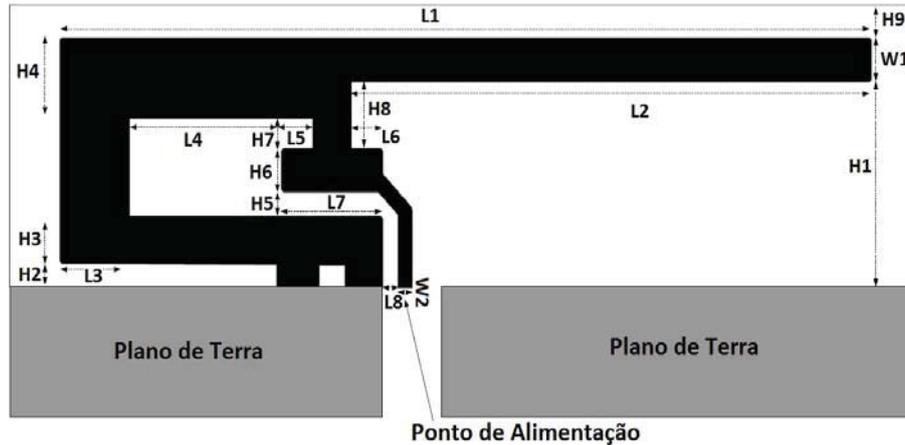
Outros modelos de referência de antenas podem ser encontrados em [66] (antena dipolo) e [67] (antena YAGI). As antenas dipolo de meia onda são as mais comuns e fáceis de implementar em RF, já as YAGI são antenas direcionais, com alto ganho [63, 66, 67].

Após análise, optou-se pela utilização da *Inverted-F Antenna*, por apresentar maior eficiência e ser omnidirecional. A *Inverted-F Antenna* pode ser visualizada na Figura 18 e as suas medidas, em milímetros, podem ser vistas na Tabela 5.

Tabela 5. Medidas utilizadas na construção da *Inverted-F Antenna* [65].

<b>H1</b>	5.70mm	<b>W2</b>	0.46mm
<b>H2</b>	0.74mm	<b>L1</b>	25.58mm
<b>H3</b>	1.29mm	<b>L2</b>	16.40mm
<b>H4</b>	2.21mm	<b>L3</b>	2.18mm
<b>H5</b>	0.66mm	<b>L4</b>	4.80mm
<b>H6</b>	1.21mm	<b>L5</b>	1.00mm
<b>H7</b>	0.80mm	<b>L6</b>	1.00mm
<b>H8</b>	1.8mm	<b>L7</b>	3.20mm
<b>H9</b>	0.61mm	<b>L8</b>	0.45mm
<b>W1</b>	1.21mm		

Figura 18. Modelo utilizado para o desenho da *Inverted-F Antenna* [65].



Para aplicações nas quais se faz necessária a utilização de uma antena com características diferentes da *Inverted-F Antenna*, foi inserido um conector SMA na PCI, onde podem ser conectados diferentes tipos de antenas, com maior ganho, direcionais, etc. Para que se possa fazer uso desse conector, é necessário mudar a posição de um resistor na PCI, desviando o sinal que iria para a antena *on-board* para o conector SMA. De qualquer forma, apenas antenas com impedância igual a 50 ohms podem ser utilizadas.

### 3.5 CIRCUITO DE RF

Qualquer sistema que se utilize de comunicação sem fio é dependente de algum tipo de antena. Uma antena pode ser definida como um dispositivo geralmente metálico para a radiação ou recepção de ondas de rádio [68]. Em outras palavras, é o último componente entre o hardware e o meio de transmissão (ar, água, etc), sendo normalmente alimentada por uma linha de transmissão [40, 68].

As linhas de transmissão são enlaces críticos em qualquer sistema de comunicação, mas, quanto mais alta a frequência, mais crítico é o projeto dessa linha. Podem ser caracterizadas como qualquer par de condutores destinados a conduzir ou a guiar uma onda eletromagnética, tendo a impedância característica como principal parâmetro. Para que uma linha de transmissão apresente máxima eficiência, toda a potência transmitida pelo emissor deve chegar à carga, mas isso só irá ocorrer se a carga e o emissor possuírem a mesma impedância. Nesse caso específico, diz-se que o sistema está casado. Em casos em que a carga e o emissor não possuem a mesma impedância, é possível e necessário fazer a inserção de circuitos casadores de impedância (balun) para aumentar a eficiência da transmissão de energia [40, 69, 70].

Trilhas de placas de circuito impresso também podem ser consideradas linhas de transmissão. São tratadas como um tipo especial de linhas de transmissão, chamadas de *microstrips* ou microlinhas de fita [40, 70]. Informações técnicas e estudos mais detalhados a respeito de linhas de transmissão e antenas podem ser encontrados em [40, 68, 69, 70, 71, 72].

A *Texas Instruments*, fabricante do SoC CC2538 e do *range extender* CC2592 fornece um *design* de referência para PCIs que utilizem seus produtos, porém, para a personalização do hardware, é necessário efetuar mudanças em seu *design*, para isso, como material de apoio, foi utilizado o *Application Note AN068* [73] que tem como título *Adapting TI LPRF Reference Designs for Layer Stacking*, e no qual podem ser encontradas, todas as informações a respeito de parâmetros que podem afetar a eficiência do circuito de RF.

## 3.6 DISPOSITIVOS DE I/O

### 3.6.1 Entradas e saídas de uso geral

O SoC CC2538 disponibiliza até 32 GPIOs (General-Purpose Inputs/Outputs), dependendo da configuração utilizada. Essas GPIOs estão divididas em quatro blocos individuais, chamados de portas, sendo as portas A, B, C e D. O dispositivo pode ser programado para ter acesso individual a cada uma das portas disponíveis. Além disso, cada porta permite acesso para leitura e gravação individual a cada GPIO [59, 60]. As GPIOs disponíveis possuem as seguintes características:

- *toggle*<sup>2</sup> rápido, com capacidade de uma mudança de estado a cada dois ciclos de *clock*;
- resistores de *pull-up*<sup>3</sup> e *pull-down*<sup>4</sup>;
- multiplexador de funções altamente flexível, que permite a utilização como GPIO, ou como algum periférico do sistema (UART, SPI, I2C, etc);
- controle programável para interrupções, com opção de escolha de interrupção por borda ou por nível.

### 3.6.2 Módulo de controle de entradas e saídas

O módulo de controle de I/O (Inputs/Outputs) está conectado a todos os GPIOs disponíveis, redirecionando os sinais por meio de uma matriz de multiplexadores. Dessa forma, os sinais podem ser recebidos e transmitidos por qualquer periférico em qualquer GPIO disponível no CC2538. Assim, por meio dos GPIOs, o SoC CC2538 oferece acesso aos seguintes periféricos [59, 60]:

- UART 0 e 1;
- SSI 0 e 1;
- I2C; e
- *timers* de uso geral, acesso aos *timers* 0, 1, 2 e 3.

---

<sup>2</sup>Inversão de estado lógico.

<sup>3</sup>Força nível lógico alto através do hardware.

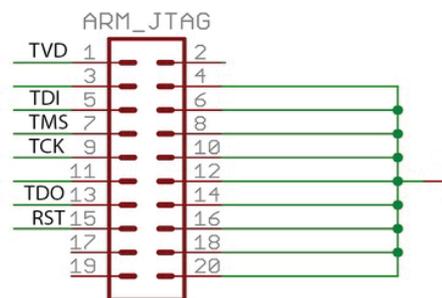
<sup>4</sup>Força nível lógico baixo através do hardware.

### 3.7 DEMAIS PERIFÉRICOS

#### 3.7.1 Conector JTAG

O SoC CC2538 oferece uma interface para *debug*, que suporta tanto o padrão JTAG de quatro pinos (IEEE 1149.1) quanto o padrão cJTAG de dois pinos (IEEE 1149.7). Após o reset do CC2538, o padrão configurado é o cJTAG. A configuração física do conector utilizado é o padrão ARM JTAG de 20 pinos. Na Figura 19, é mostrado o diagrama elétrico de ligações do conector utilizado [60].

Figura 19. Pinagem do conector para *debug* e gravação do SoC CC2538 utilizando o padrão ARM JTAG de 20 pinos [74].



#### 3.7.2 System clock e real time clock

Existem dois diferentes tipos de *clocks* no SoC CC2538, um chamado de *system clock* ou *main clock* e o *real time clock* (RTC). O *system clock* pode ser provido por duas fontes diferentes, de um oscilador RC interno com frequência de 16MHz, ou de um cristal oscilador externo com frequência de 32MHz. O oscilador de 16MHz consome menos energia do que o oscilador de 32MHz a cristal, porém, não apresenta a mesma precisão do oscilador de 32MHz a cristal, por isso, ele não pode ser utilizado pelo transceiver de RF, pelo conversor A/D ou pela porta USB. O tempo de inicialização do cristal de 32MHz pode ser muito longo para determinadas aplicações, sendo assim, pode-se iniciar o uso do sistema com o oscilador de 16MHz até que o oscilador de cristal esteja estabilizado. O CC2538 possui um módulo para a detecção da estabilidade do oscilador a cristal [60].

O *real time clock* também pode ser provido por um oscilador RC interno ou um cristal oscilador externo. O oscilador interno é utilizado para redução de custos e consumo de energia, porém, é menos preciso quando comparado ao oscilador a cristal. O oscilador interno é calibrado automaticamente quando o oscilador a cristal de 32MHz é ativado pelo sistema, após a calibração o valor do *clock* do RTC é 32.753 kHz. O oscilador externo provê uma frequência de *clock* de 32.768 KHz, sendo muito mais preciso, porém, aumenta o consumo de energia. As duas fontes de *clock* não podem ser utilizadas ao mesmo tempo, somente uma delas pode ser selecionada. Essa fonte de *clock* é responsável pelo *sleep timer* e pelo *watchdog timer* [60].

### 3.7.3 BootLoader

O CC2538 disponibiliza um sistema de *bootloader* que permite a gravação de uma aplicação na memória flash sem a necessidade de utilização de um gravador específico, utilizando, para isso, a comunicação UART ou SPI. O serial *bootloader* é implementado como um mecanismo cliente servidor que requer um dispositivo mestre para controlar o processo, isso é, um aplicativo hospedado em um PC com acesso à conexão serial do CC2538. Informações detalhadas e procedimentos sobre esse processo podem ser encontradas em [75, 76, 77].

### 3.7.4 Monitor de bateria e sensor de temperatura

O conversor A/D do SoC CC2538, além dos canais de conversão convencionais, apresenta duas opções extras em sua configuração, estes correspondem à leitura de um sensor de temperatura interno do conversor A/D e a um sistema de monitoramento da tensão de alimentação do CC2538. O sistema de monitoramento faz a leitura interna da alimentação do conversor A/D e divide esse valor por três, assim, em aplicações nas quais a alimentação do sistema é feita diretamente por baterias, sem nenhum regulador de tensão, consegue-se monitorar a descarga da bateria. Isso possibilita, por exemplo, a implementação de algoritmos que façam a leitura da quantidade de energia disponível na bateria e adaptem o funcionamento do hardware, reduzindo o consumo de energia e, conseqüentemente, aumentando a vida útil da bateria [60].

## 3.8 TESTES E RESULTADOS DE HARDWARE

### 3.8.1 Circuito de alimentação

O módulo foi construído de forma que a alimentação possa ser feita de duas maneiras, diretamente pela porta USB, ou por uma fonte de energia externa. A porta USB fornece uma tensão de 5 VCC. O sistema funciona com uma tensão máxima de alimentação de 3,6 VCC, com isso, se fez necessário utilizar um regulador de tensão com baixa queda de tensão entre sua entrada e saída, *Low-Dropout*. Dessa forma, foi selecionado o regulador da *Texas Instruments* TLV70233DBVR, o qual tem saída fixa de 3,3 VCC e 300mA de corrente máxima, além de circuitos internos para proteção em casos de sobrecorrente e sobreaquecimento [78].

Quando a alimentação do sistema não é feita pela porta USB, mas por outra fonte externa, o circuito não possui nenhum regulador ou limitador de tensão, sendo necessário utilizar uma fonte de energia com tensão máxima de 3,6 VCC. Essa forma de alimentação foi implementada para permitir o uso de baterias, diminuindo as perdas de energia e também tornando possível a utilização do sistema de monitoramento de baterias que o SoC CC2538 disponibiliza.

### 3.8.2 Custos de desenvolvimento dos módulos

Foram desenvolvidos dois módulos de hardware neste projeto, um módulo *low power*, utilizando somente o SoC CC2538, e um módulo *high power*, utilizando o SoC CC2538 em conjunto com o *range extender* CC2592. Todos os custos de componentes e PCI, com valores em dólar americano, estão descritos nas Tabelas 6 e 7.

Tabela 6. Custo para o desenvolvimento do módulo *low power* utilizando o SoC CC2538.

Descrição do Item	Custo (\$)
Capacitores	0,461
Resistores	0,168
Indutores	0,23
Filtro EMI	0,115
Regulador de tensão	0,45
Cristal 32MHz	0,80
Cristal 32.768KHz	1,55
CC2538	10,50
PCI	1,95
Total	\$16,22

Tabela 7. Custo para o desenvolvimento do módulo *high power* utilizando o SoC CC2538 em conjunto com o *range extender* CC2592.

Descrição do Item	Custo (\$)
Capacitores	0,515
Resistores	0,225
Indutores	0,29
Filtro EMI	0,23
Regulador de tensão	0,45
Cristal 32MHz	0,80
Cristal 32.768KHz	1,55
CC2538	10,50
CC2592	3,65
Conector coaxial	3,41
PCI	3,20
Total	\$24,82

Os custos apresentados são referentes ao desenvolvimento de uma unidade de cada módulo em pequena escala de produção. Para grandes escalas de produção, o valor final pode ser reduzido, por meio da compra de grandes quantidades de componentes. Outros materiais foram adquiridos para suporte ao desenvolvimento do projeto; a lista completa com os materiais adquiridos e utilizados encontra-se no Apêndice A.

### 3.8.3 Módulo low power

Uma das etapas deste trabalho compreendeu o projeto, o desenvolvimento e a construção de um módulo *low power*, utilizando o SoC CC2538. Esse módulo foi projetado e desenvolvido utilizando o software *Altium Designer* [79], que proporciona ótima precisão no desenho de PCBs. O projeto completo desse módulo, contendo todos os esquemáticos elétricos e os layouts da PCB, estão disponibilizados no Apêndice B. O módulo, após a montagem dos componentes, pode ser visto na Figura 20.

Figura 20. Módulo *low power* desenvolvido utilizando o SoC CC2538 da *Texas Instruments* (do autor).



O módulo desenvolvido possui conectores que disponibilizam o acesso a todas as GPIOs disponíveis no SoC CC2538. Além disso, neste projeto, os conectores foram posicionados estrategicamente, de maneira que possam ser utilizados *shields* de componentes desenvolvidos para a plataforma Arduino, facilitando, assim, em termos de hardware, o processo de desenvolvimento de novas aplicações. A gravação da memória flash e a depuração do CC2538 é feita por meio de um conector de 20 pinos, utilizando o padrão ARM-JTAG.

Os conectores P7 e CN1 (USB), Figura 21, são utilizados como meios de alimentação do sistema e não podem ser utilizados simultaneamente. Dessa forma, foi inserido um *jumper* (P1), com o propósito de possibilitar a seleção da fonte de alimentação do sistema.

Figura 21. Detalhes do sistema de alimentação, mostrando os conectores CN1, P7 e P1, e do sistema de RF, mostrando o conector AN1 e os indutores L2 e L4 (do autor).

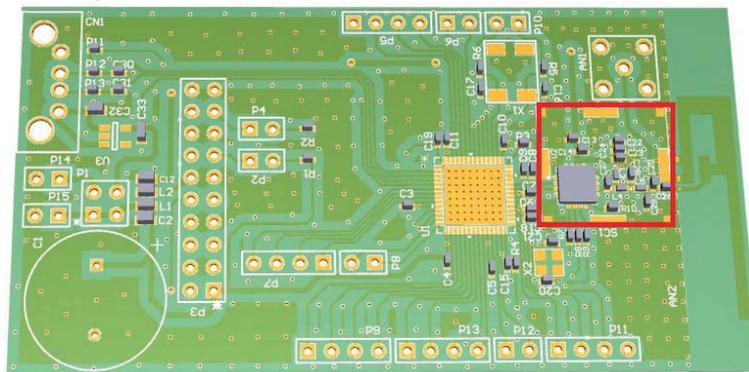


O conector de RF AN1 (Figura 21) é disponibilizado para a utilização de antenas externas. Para isso, é necessário fazer a soldagem do indutor L2 de 3,3nH e a remoção do indutor L4, de mesmo valor. Isso irá desviar o sinal da antena da PCI para o conector AN1.

### 3.8.4 Módulo high power

O módulo *high power* construído faz uso do SoC CC2538 em conjunto com o *range extender* CC2592. A PCI pode ser vista na Figura 22. O projeto completo desse módulo, contendo todos os esquemáticos elétricos e os layouts da PCI, está disponibilizado, no Apêndice C. O módulo *high power* disponibiliza maior potência de saída, aumentando a distância de transmissão alcançada.

Figura 22. Módulo *high power* projetado, utilizando o SoC CC2538 em conjunto com o *range extender* CC2592, ambos fabricados pela *Texas Instruments* (do autor).



Na Figura 22, a parte da PCI, destacada em vermelho, refere-se ao circuito do *range extender* CC2592. O módulo possui conectores que disponibilizam acesso a todas as GPIOs disponíveis no SoC CC2538. A gravação da memória flash e a depuração do CC2538 são feitas por meio de um conector ARM-JTAG de 20 pinos.

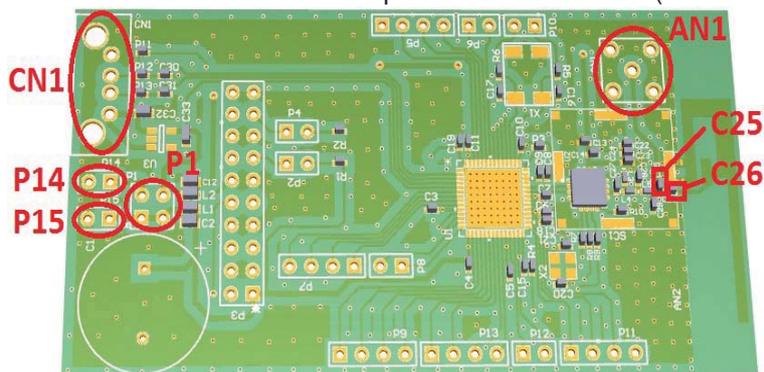
Os conectores P14 e CN1 (USB) (Figura 23) são utilizados como meios de alimentação do sistema, e não podem ser utilizados simultaneamente, dessa forma, foi inserido um *jumper* (P1) com o propósito de possibilitar a seleção da fonte de alimentação do sistema. O conector P15 oferece uma saída de tensão regulada de 3,3VCC quando a fonte de alimentação utilizada é a conexão USB.

Da mesma maneira que o módulo *low power* desenvolvido, o *high power* também oferece a possibilidade de uso de antenas externas, via conector AN1. Para isso, é necessário fazer a solda do capacitor C25 de 18pF e a remoção do capacitor C26, de mesmo valor (Figura 23). Com esse procedimento, o sinal de RF será recebido e transmitido via antena externa, desabilitando a antena na PCI.

### 3.8.5 Sniffer CC2531

Um *sniffer* é um analisador de redes, mais precisamente um analisador de pacotes. O *sniffer* tem a função de capturar pacotes que estão trafegando em uma rede, possibilitando a visualização

Figura 23. Detalhes do sistema de alimentação, mostrando os conectores CN1, P14, P15 e P1, e do sistema de RF, mostrando o conector AN1 e os capacitores C25 e C26 (do autor).

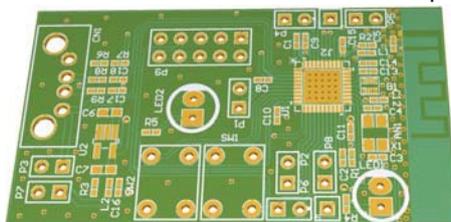


do conteúdo desses pacotes. É uma ferramenta importante no desenvolvimento de dispositivos e aplicações em redes.

A *Texas Instruments* fornece um *sniffer* para apoio ao desenvolvimento de redes que utilizem o padrão de comunicação Zigbee [80]. Fornece também um *design* de referência [81] e o firmware utilizado [82].

Como o *sniffer* é uma ferramenta importante no desenvolvimento de dispositivos, optou-se também por desenvolver um módulo que pudesse ser utilizado para essa função. Tomando como base o material fornecido pelo fabricante, desenvolveu-se o módulo apresentado na Figura 24, baseado no SoC de 8 bits CC2531 [17].

Figura 24. *Sniffer* projetado, utilizando o SoC CC2531 fabricado pela *Texas Instruments* (do autor).



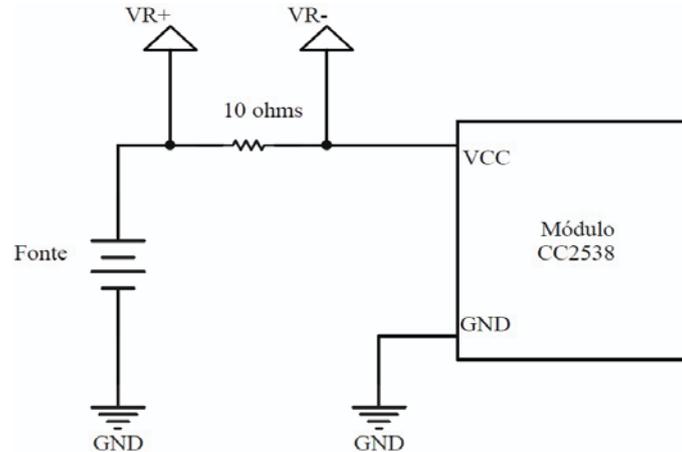
Em comparação com o módulo fornecido pela *Texas Instruments*, neste foram incluídos conectores que dão acesso às GPIOs e foi adequado o conector para gravação da memória flash e depuração do firmware. O projeto completo desse módulo, contendo todos os esquemáticos elétricos e os layouts da PCB, está disponibilizado no Apêndice D.

### 3.8.6 Ensaio de consumo de energia

O valor da potência de saída do rádio afeta diretamente o consumo de energia do dispositivo desenvolvido. Para que fosse possível medir esse consumo de energia, foi inserido um resistor de 10 ohms em série com a alimentação do dispositivo, como é mostrado na Figura 25. Com isso, a corrente de alimentação que passa pelo resistor gera uma diferença de potencial (VR) que pode ser medida nos pontos VR+ e VR-, com o auxílio de um osciloscópio. Para se chegar ao valor da corrente I, é

necessário aplicar a Equação 1. O SoC CC2538 oferece a possibilidade de configuração da potência de saída do rádio transmissor, via software. Na Tabela 8, são mostrados todos os valores possíveis.

Figura 25. Circuito implementado para medição do consumo de energia do módulo com o SoC CC2538 (do autor).



$$I = \frac{VR}{10ohms} \quad (1)$$

Tabela 8. Valores de configuração da potência de saída, do rádio transmissor do SoC CC2538, em dBm e o valor equivalente em mW [60].

Potência de saída (dBm)	Potência de saída (mW)
-15 dBm	0,0316 mW
-9 dBm	0,125 mW
-3 dBm	0,501 mW
0 dBm	1 mW
3 dBm	2,51 mW
7 dBm	5,01 mW

Em uma primeira bateria de testes, foi utilizada uma fonte de alimentação com saída fixada em 3,5VCC. Variando a potência de saída, via software, do módulo construído e com o auxílio de um osciloscópio, *Agilent InfiniVision* modelo DSO-X 2004A, foram efetuadas medições da diferença de potencial gerada sobre o resistor de 10 ohms, nos pontos VR+ e VR-. Assim, gerou-se os valores de consumo de corrente, os quais podem ser visualizados na Tabela 9.

Uma segunda bateria de testes foi efetuada. Para verificar a resposta do dispositivo à variação da tensão de alimentação, reduziu-se a tensão de alimentação do circuito para 2,3VCC, e novas medidas da diferença de potencial sobre o resistor de 10 ohms foram efetuadas. Os valores resultantes podem ser visualizados na Tabela 10.

Um exemplo da forma de onda capturada com o auxílio do osciloscópio pode ser visto na Figura 26, a qual se refere ao dispositivo programado com potência de saída de +7dBm, sendo a alimentação do circuito de 3,5VCC.

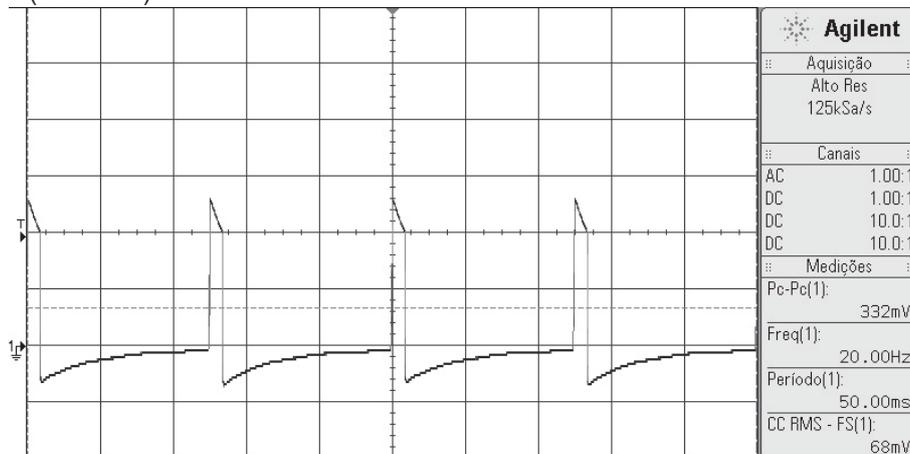
Tabela 9. Valores de diferença de potencial e corrente, com alimentação do circuito de 3,5VCC (do autor).

Potência de saída	Diferença de potencial medida	Corrente calculada
-15 dBm	179 mV	17,9 mA
-9 dBm	181 mV	18,1 mA
-3 dBm	193 mV	19,3 mA
0 dBm	206 mV	20,6 mA
3 dBm	235 mV	23,5 mA
7 dBm	332 mV	33,2 mA

Tabela 10. Valores de diferença de potencial e corrente lidos, com alimentação do circuito de 2,2VCC (do autor).

Potência de saída	Diferença de potencial medida	Corrente calculada
-15 dBm	173 mV	17,3 mA
-9 dBm	178 mV	17,8 mA
-3 dBm	190 mV	19 mA
0 dBm	202 mV	20,2 mA
3 dBm	232 mV	23,2 mA
7 dBm	320 mV	32 mA

Figura 26. Forma de onda da diferença de potencial medida sobre o resistor de 10 ohms. Potência de saída do dispositivo programada em +7dBm. Osciloscópio com resolução de 100 mV no eixo y e 20 ms no eixo x (do autor).



O *firmware* utilizado no SoC CC2538 nessas baterias de testes pode ser encontrado em [83]. Foram efetuadas alterações em sua função *main*, a qual, com as alterações, pode ser visualizada na Figura 27. A linha 8 da Figura 27 refere-se à programação da potência de saída do dispositivo. Nesse exemplo, está definido o valor de -15dBm (0x80|15).

### 3.8.7 Ensaio de distância de transmissão

O valor da potência de saída do SoC CC2538 afeta diretamente o alcance da transmissão de dados entre dois dispositivos. Levando isso em consideração, foram efetuados alguns testes para

Figura 27. Função *main* do SoC CC2538 modificada para utilização nas baterias de testes de consumo de energia (do autor).

---

```

1. int main(void)
2. {
3.     bspInit(BSP_SYS_CLK_SPD);
4.     bspSpiInit(BSP_SPI_CLK_SPD);
5.     perConfig.mode = PER_MODE_TX;
6.     perConfig.state = PER_IDLE;
7.     perConfig.channel = 11;
8.     perConfig.txPower = (0x80|15);
9.     perConfig.burstSize = 100000;
10.    perConfig.pktRate = 20;
11.    perConfig.gainMode = PER_GAIN_MODE_NONE;
12.    basicRfConfig.panId = PAN_ID;
13.    basicRfConfig.ackRequest = false;
14.    appTransmitter();
15. }

```

---

levantar as distâncias máximas alcançadas. Esses testes foram executados utilizando o módulo *low power* desenvolvido, como base transmissora dos dados. Como base receptora de dados, foram utilizados os módulos de desenvolvimento SMARTRF06EBK [84] e CC2538EMK [85], da *Texas Instruments*.

O módulo de desenvolvimento CC2538EMK é uma PCI contendo basicamente o SoC CC2538, uma antena *on board* e conectores padrões para a conexão com o módulo SMARTRF06EBK; não disponibiliza acesso direto a nenhuma GPIO. Já o módulo de desenvolvimento SMARTRF06EBK é uma PCI com periféricos, tais como botões e display. Por meio desse módulo, se tem acesso às GPIOs do SoC CC2538 contido no módulo CC2538EMK. Com o SMARTRF06EBK, também é possível gravar e depurar *firmwares* no CC2538.

Os testes efetuados consistiram na transmissão de um número fixo de pacotes de dados. Com isso, foi possível determinar com precisão o número de pacotes de dados transmitidos com sucesso e o número de pacotes perdidos na transmissão. Como módulo transmissor, foi utilizado o módulo *low power*, desenvolvido neste trabalho. Na recepção dos dados, foram utilizados os módulos SMARTRF06EBK e CC2538EMK. O *firmware* utilizado para a transmissão dos dados pode ser visto na Figura 28. A linha 9 é referente à quantidade de pacotes de dados a serem transmitidos, o valor utilizado foi 1000 (mil). Na recepção de dados, o *firmware* utilizado foi o disponibilizado pela *Texas Instruments* em [83] e nenhuma alteração foi efetuada para essa função.

Com a base responsável pela recepção de dados fixa em um ponto, variou-se a potência e a distância do módulo transmissor. Dessa forma, os dados com os valores das distâncias máximas e a quantidade de pacotes perdidos foram obtidos. Essas informações são mostradas na Tabela 11. Os testes foram efetuados em área aberta, sem obstáculos ou fontes de interferência visíveis, e com visada direta entre os dispositivos.

Figura 28. Função *main* do *firmware* utilizado nas baterias de testes para os ensaios de distância de transmissão do módulo *low power* desenvolvido (do autor).

---

```

1. int main(void)
2. {
3.     bspInit(BSP_SYS_CLK_SPD);
4.     bspSpiInit(BSP_SPI_CLK_SPD);
5.     perConfig.mode = PER_MODE_TX;
6.     perConfig.state = PER_IDLE;
7.     perConfig.channel = 11;
8.     perConfig.txPower = (0x80|15);
9.     perConfig.burstSize = 1000;
10.    perConfig.pktRate = 20;
11.    perConfig.gainMode = PER_GAIN_MODE_NONE;
12.    basicRfConfig.panId = PAN_ID;
13.    basicRfConfig.ackRequest = false;
14.    appTransmitter();
15. }

```

---

### 3.9 CONCLUSÕES DO CAPÍTULO

O desenvolvimento de dispositivos de hardware para apoio ao desenvolvimento de aplicações de redes sem fio que agreguem baixo consumo de energia, confiabilidade e baixo custo é atualmente, um grande desafio. Nesse sentido, durante a etapa de pesquisa e escolha dos materiais a serem utilizados, foram levadas em consideração características como o custo, o suporte fornecido pelo fabricante e o consumo de energia, fatores que amparam a decisão de tomar como base para o desenvolvimento deste projeto o SoC CC2538.

Tendo como base o levantamento dos custos de desenvolvimento dos módulos (Tabelas 6 e 7), pode-se afirmar que o desenvolvimento de um dispositivo de hardware próprio é uma opção mais barata, quando comparada a outros dispositivos com características semelhantes disponíveis no mercado. Um conjunto contendo cinco módulos da plataforma OpenMote [47] custa 500 euros [86]. Outro fator favorável é a possibilidade de personalização do dispositivo.

Com relação ao consumo de energia, os resultados apresentados foram satisfatórios (Tabela 9). Mesmo programado em sua potência máxima (+7dBm), o módulo *low power* consome apenas 33,2 mA, com tensão de alimentação de 3,5 VCC. O módulo também apresentou boa resposta quando a tensão de alimentação foi reduzida a 2,2 VCC, mantendo a corrente consumida praticamente estável (Tabela 10). Desse modo, foi possível alimentar o sistema utilizando até mesmo duas pilhas alcalinas comuns. Isto possibilita maior mobilidade e menor custo na utilização dos módulos desenvolvidos.

Com o intuito de validar a antena *on-board* utilizada e de fazer o levantamento das distâncias máximas alcançadas, foram efetuados testes de transmissão de dados. Os resultados obtidos nesses testes foram satisfatórios (Tabela 11). A distância máxima alcançada foi 160 metros, mas o número de pacotes perdidos, nesse caso, é muito elevado (980, de 1000). Recomenda-se, portanto, não

Tabela 11. Levantamento das distâncias máximas alcançadas e quantidade de pacotes perdidos, na transmissão de dados entre dois dispositivos. Valores obtidos em campo aberto, com visada direta entre os dispositivos (do autor).

<b>Distância (m) / Potência</b>	<b>7 dBm</b>	<b>3 dBm</b>	<b>0 dBm</b>	<b>-3 dBm</b>	<b>- 9 dBm</b>	<b>-15 dBm</b>
<b>1m</b>	0	0	0	0	0	15
<b>5m</b>	0	0	0	2	0	197
<b>10m</b>	0	0	0	2	0	481
<b>20m</b>	0	0	4	3	878	826
<b>30m</b>	2	0	3	0	-	-
<b>40m</b>	3	0	0	2	-	-
<b>50m</b>	0	3	2	4	-	-
<b>60m</b>	0	0	0	173	-	-
<b>70m</b>	2	2	2	422	-	-
<b>80m</b>	0	2	159	981	-	-
<b>90m</b>	3	89	564	-	-	-
<b>100m</b>	2	346	921	-	-	-
<b>110m</b>	27	578	-	-	-	-
<b>120m</b>	86	894	-	-	-	-
<b>130m</b>	127	-	-	-	-	-
<b>140m</b>	486	-	-	-	-	-
<b>150m</b>	814	-	-	-	-	-
<b>160m</b>	980	-	-	-	-	-

ultrapassar 100 metros de distância entre dois nodos, assim, a perda de pacotes é menor (2, de 1000), diminuindo o desperdício de energia na retransmissão de pacotes perdidos.

As PCIs foram projetadas levando em consideração fatores como a disposição dos conectores de acesso às GPIOs do SoC CC2538, permitindo o uso de *shields* desenvolvidos com o padrão de pinagem do arduino. O sistema de seleção da fonte de alimentação permite que os módulos sejam alimentados por diferentes fontes de energia, desde conversores de tensão até pilhas e baterias diversas. Outra característica apresentada é a possibilidade de utilização de antenas externas, de tal modo que antenas com propriedades específicas também podem ser utilizadas.

Após a aplicação dos testes, o módulo utilizando o SoC CC2538 apresentou baixo consumo de energia, que o alcance obtido na transmissão de dados foi satisfatório e que o custo final de desenvolvimento dos módulos, incluindo a fabricação da PCI, mostrou-se aceitável. Com isso, o conjunto de resultados obtidos pode ser considerado satisfatório.



## 4. SOBRE O SOFTWARE

Neste capítulo, são apresentadas ferramentas e recursos utilizados no desenvolvimento e na implementação do *firmware* para o SoC CC2538.

### 4.1 FOUNDATION FIRMWARE

*Foundation Firmware* é uma coletânea de bibliotecas de software disponibilizadas pela *Texas Instruments*, composta pelos seguintes componentes [87]:

- *Driverlib* - é uma biblioteca de drivers periféricos de baixo nível, presentes no CC2538. Define todos os registradores e máscaras de bits. Disponibiliza funções para configuração e controle de periféricos, tais como o conversor A/D, GPIOs, I2C, SSI, UART, DMA, *timers*, entre outros;
- *Usblib* - é um *framework* completo para o controlador USB disponível no CC2538. Implementa todo o protocolo de comunicação USB, além de disponibilizar exemplos de aplicações;
- *Bsp* - conjunto de bibliotecas que oferece suporte a periféricos disponíveis no módulo de avaliação *SmartRF06EB* [84], fabricado pela *Texas instrument*. Esse módulo oferece periféricos para testes do CC2538, tais como display LCD, botões, sensor de iluminação, entre outros. Disponibiliza ainda um programador e um depurador de memória flash.

O *Foundation Firmware* é escrito em linguagem C, tornando a implementação eficiente e fácil [87]. O uso de diretrizes intrínsecas específicas é reduzido, possibilitando o uso com diferentes compiladores. Dois compiladores suportados são: o *Code Composer Studio* [88], da *Texas Instruments*; e o *IAR Embedded Workbench for ARM* [89], da *IAR Systems*.

### 4.2 TIMAC

Uma alternativa ao protocolo Zigbee, também disponibilizada pela *Texas Instruments*, é o protocolo TIMAC [90]. Esse protocolo é mais simples, porém, possui algumas limitações. A principal é não suportar a topologia de rede *mesh*. As principais características apresentadas pelo protocolo TIMAC são [90]:

- suporte à topologia de rede *star*;
- suporte ao padrão IEEE 802.15.4 - 2006;
- funcionamento em modo síncrono e assíncrono; e
- recursos de segurança de rede.

O protocolo TIMAC é utilizado em aplicações nas quais se façam necessárias soluções de comunicação sem fio ponto-a-ponto ou ponto-multiponto. Representa uma alternativa com menor complexidade de implementação. Entretanto, seu uso fica inviabilizado em redes com maior complexidade.

#### 4.3 PILHA ZIGBEE

A *Texas Instruments* fornece a pilha de protocolos Zigbee ( 2.1.3.2), chamada *Z-Stack*, a qual encontra-se disponível em [91]. Essa solução é baseada no protocolo Zigbee PRO 2012, e possui certificação *ZigBee Compliant Platform* (ZCP) e certificação MAC 802.15.4. Essas certificações comprovam que a pilha de protocolos segue os padrões estabelecidos pela *Zigbee Alliance* e pela IEEE, respectivamente.

A *Z-Stack* é um conjunto de bibliotecas de software, as quais implementam configurações e funcionalidades necessárias para a construção de todas as camadas de rede do protocolo Zigbee. A *Texas Instruments* disponibiliza a *Z-Stack* em quatro diferentes formatos [35, 91]:

- *Z-Stack - Home* - desenvolvida para uso em automação residencial. Interliga os mais variados tipos de dispositivos, tais como sensores de alarme, controle de abertura de portas, controle de temperatura ambiente, entre outros;
- *Z-Stack - Lighting* - desenvolvida para uso em controles de sistemas de iluminação;
- *Z-Stack - Mesh* - desenvolvida para uso em qualquer tipo de aplicação. Oferece maior número de recursos, porém, possui maior complexidade de implementação;
- *Z-Stack - Linux-Gateway* - desenvolvida para oferecer um padrão para conectar vários tipos de redes Zigbee à internet.

O SoC CC2538 é compatível somente com as pilhas *Z-Stack-Home* e *Z-Stack-Mesh*. A pilha *Z-Stack-Lighting* é destinada ao uso com microcontroladores de oito bits, com menor capacidade de processamento e de memória. Já a pilha *Z-Stack-Linux-Gateway* é destinada ao uso com microprocessadores, necessitando de maior capacidade de processamento e memória [91].

#### 4.4 Z-STACK - MESH

A *Z-Stack-Mesh* implementa a arquitetura Zigbee, que pode ser vista na Figura 6, no subcapítulo 2.1.3.2, e é composta pelas camadas [92]:

- ZDO - oferece funcionalidades para gerenciar o dispositivo Zigbee. A interface para as aplicações, incluindo a criação, a descoberta e a associação a uma rede Zigbee;
- AF - disponibiliza as estruturas de dados e as funções auxiliares que o desenvolvedor necessita. É o multiplexador para as mensagens recebidas;

- APS - é um conjunto geral de serviços de suporte, que são utilizados pela camada ZDO e por aplicativos definidos pelo fabricante;
- NWK - provê serviços de gerenciamento de dados para componentes das camadas superiores;
- ZMAC - é uma interface entre as camadas MAC e PHY e a camada NWK.

Uma descrição completa e detalhada de todas as funções contidas na *Z-Stack-Mesh* e nas camadas inferiores (MAC e PHY) implementadas pode ser encontrada em [92] e em [93], respectivamente.

## 4.5 SISTEMA OPERACIONAL

A *Z-Stack-Mesh* implementa os protocolos de rede necessários do padrão Zigbee. Entretanto, ela não é um sistema operacional, e sim um conjunto de bibliotecas que devem ser utilizadas em conjunto com um sistema operacional. A *Texas Instruments* fornece um sistema operacional que é usado em conjunto com o SoC CC2538, acompanhando a *Z-Stack-Mesh* [91].

### 4.5.1 OSAL

O *Operational System Abstraction Layer* (OSAL) [94] é o sistema operacional disponibilizado pela *Texas Instruments* e utilizado na implementação da *Z-Stack-Mesh* no Soc CC2538. O OSAL funciona por meio do agendamento de tarefas, possui um lista de tarefas declarada previamente, e a executa em sequência. Todas as tarefas têm o mesmo tempo de execução. Após o término desse tempo, a tarefa é encerrada, podendo ser retomada ou reiniciada no próximo *loop* da lista de tarefas. Após a execução da última tarefa declarada na lista, a primeira tarefa é executada novamente, criando assim, um *loop* infinito [94].

Interrupções externas do SoC CC2538 podem executar tarefas que não estejam na lista definida inicialmente. Essas tarefas são disparadas somente com a ação de algum periférico, como por exemplo um botão pressionado em alguma GPIO. Dessa forma, as tarefas referentes às interrupções não são executadas a cada ciclo dentro do *loop* [94].

O OSAL é utilizado para abstrair os componentes da *Z-Stack-Mesh* do ambiente de desenvolvimento, oferecendo um conjunto de bibliotecas com as seguintes funcionalidades [94]:

- registro e inicialização de tarefas - é utilizado para adicionar e gerenciar tarefas no OSAL. Cada tarefa deve ser composta por uma função de inicialização e uma função de processamento de eventos. O OSAL utiliza uma tabela de tarefas, para chamar o processador de eventos;
- troca de mensagens entre as tarefas - permite a uma tarefa alocar e desalocar mensagens na memória, enviar mensagens de comando a outra tarefa e receber mensagens de respostas;

- sincronização das tarefas - permite que uma tarefa aguarde a ocorrência de eventos e retorne dados, pode ser usada para definir eventos e notificar uma tarefa assim que o evento for definido;
- manipulação de interrupções - habilita uma tarefa para interface com interrupções externas. Permite a uma tarefa associar uma rotina específica para cada interrupção. As interrupções podem ser habilitadas e desabilitadas dentro de qualquer tarefa;
- temporizadores (*timers*) - permitem a utilização de *timers* por tarefas internas (*Z-Stack-Mesh*), e tarefas externas a nível de aplicação. Fornecem funções para iniciar e parar um *timer*. Os *timers* podem ser ajustados em incrementos de 1 milissegundo;
- alocação de memória volátil- permite a alocação dinâmica de memória;
- API de memória não volátil - fornece uma maneira para que as aplicações armazenem informações na memória do dispositivo de forma persistente. Também é utilizado pela *Z-Stack-Mesh*. Pode ser implementada tanto para a memória flash quanto para a memória EEPROM, e existe um intervalo exclusivo e limitado de memória para aplicações de usuário;
- API de memória não volátil simples - assim como o sistema de alocação de memória volátil, fornece uma maneira para que as aplicações armazenem informações na memória do dispositivo de forma persistente. Porém, sua implementação é muito mais simples, diminuindo o tamanho do código gerado;
- *clock system* - oferece uma maneira de manter a data e a hora no dispositivo;
- gerador de números randômicos - oferece uma função para a geração de números randômicos de 16 bits.

O OSAL oferece, portanto, um conjunto de funcionalidades para auxiliar no desenvolvimento de aplicações que utilizem a *Z-Stack-Mesh*.

#### 4.5.2 HAL

O *Hardware Abstraction Layer* (HAL) fornece um conjunto de bibliotecas que permitem o acesso à camada de hardware do SoC CC2538. São *drivers* usados pelos aplicativos para acesso ao hardware. O HAL é disponibilizado pela *Texas Instruments* e oferece uma interface para acesso aos *timers*, GPIOs, UART, conversor A/D, SPI e I2C. É uma plataforma independente que fornece uma série de recursos para cada periférico de hardware. O HAL apresenta três tipos de funções [95]:

- funções de inicialização - usadas para inicializar um serviço e/ou configurar parâmetros opcionais para dados específicos. São normalmente chamadas na inicialização do dispositivo;
- funções de acesso a serviços de hardware - são usadas para acessar diretamente os registradores, para obter ou definir determinado valor, ou para controlar os componentes de hardware;

- funções de retorno - são funções que devem ser implementadas pelo aplicativo do usuário, sendo utilizadas para passar eventos gerados pelo hardware (interrupções, contadores, temporizadores) para camadas superiores.

A utilização do HAL permite acesso direto ao hardware, possibilitando o desenvolvimento de aplicações, sem a necessidade de desenvolvimento do *driver* específico. A diferença entre o HAL e o *Foundation Firmware* é que o HAL já é disponibilizado acoplado ao OSAL e pode ser usado em diferentes microcontroladores fabricados pela *Texas Instruments*. Já o *Foundation Firmware* é específico para uso com o SoC CC2538 e também pode ser utilizado com o OSAL. Porém, se necessário, suas bibliotecas devem ser manualmente incluídas.

### 4.5.3 Power management API

O OSAL possui um sistema de gerenciamento de energia. O sistema disponibiliza uma forma pela qual as aplicações podem notificar o OSAL quando é possível desligar o rádio e o hardware externo, ativando o modo *sleep*.

Existem dois dispositivos que controlam o sistema de gerenciamento de energia. No primeiro, é definido se o sistema pode, ou não, entrar em modo *sleep*. O segundo dispositivo está nas tarefas executadas pelo OSAL. Somente quando todas as tarefas permitirem, o gerenciador de energia poderá entrar em modo *sleep*. Além disso, por padrão, o dispositivo só poderá entrar em modo *sleep* quando este fizer parte de uma rede. Caso contrário, fica permanentemente com o rádio ligado, tentando se conectar a uma rede. Para modificar esse parâmetro é, necessário alterar a função de inicialização de tarefas do aplicativo, permitindo, assim, que o dispositivo entre em modo *sleep* mesmo não estando associado a uma rede [94].

## 4.6 APLICAÇÃO DESENVOLVIDA PARA O ENSAIO DE FORMAÇÃO DE REDE

Com o hardware desenvolvido neste trabalho e os módulos de desenvolvimento adquiridos, foram efetuados testes de formação de redes. Os módulos desenvolvidos neste projeto foram utilizados como *end devices*. Em cada *end device*, foi conectado um sensor de temperatura LM35 [96], assim, periodicamente, o *end device* lê o sensor de temperatura, trata essa informação e, em seguida a transmite para a rede. No nodo coordenador, foram utilizados os módulos SMARTRF06EBK e CC2538EMK. O nodo coordenador é responsável por receber as informações enviadas pelos *end devices* e disponibilizar no display LCD do módulo SMARTRF06EBK, como pode ser visto na Figura 29.

Para o desenvolvimento da aplicação descrita, foram utilizados o OSAL e a *Z-Stack-Mesh*. Inicialmente, para padronização das informações transmitidas, foi declarado um vetor de caracteres, como variável global, conforme pode ser visualizado na linha 4 da Figura 30.

Figura 29. Visualização de informações recebidas dos *end devices* pelo nodo coordenador da rede (do autor).



Figura 30. Trecho de código do arquivo "GenericApp.c", onde é declarado o vetor utilizado para a transmissão dos valores de temperatura lidos pelo conversor A/D (do autor).

---

```

1. //Time interval between sending messages
2. static uint32 txMsgDelay = GENERICAPP_SEND_MSG_TIMEOUT;
3. //vetor contendo os valores lidos pelo conversor A/D
4. char MessageDataADC[] = "T2 = XX,X C";

```

---

Para o correto funcionamento do conversor A/D, faz-se necessário a sua inicialização e configuração prévia. Essa etapa é executada na função de inicialização do sistema "*GenericApp Init*", que pode ser visualizada na Figura 31, onde a instrução da linha 6 inicializa o conversor A/D e a instrução da linha 7 configura a tensão de referência do A/D.

Figura 31. Trecho de código do arquivo "GenericApp.c", da função de inicialização de uma tarefa, onde é inicializado o conversor A/D do SoC CC2538 (do autor).

---

```

1. void GenericApp_Init( uint8 task_id )
2. {
3.   GenericApp_TaskID = task_id;
4.   GenericApp_NwkState = DEV_INIT;
5.   GenericApp_TransID = 0;
6.   HalAdcInit();
7.   HalAdcSetReference(HAL_ADC_REF_AVDD);
8.   GenericApp_DstAddr.addrMode = (afAddrMode_t)AddrNotPresent;

```

---

Depois de inicializado e configurado, o conversor A/D pode ser utilizado. Na Figura 32, pode ser visualizado um trecho de código do processamento da tarefa, contendo as etapas de leitura do conversor A/D e do processamento do valor lido. Nas linhas 5 e 6, são declaradas as variáveis auxiliares utilizadas. Na linha 11, é efetuada a leitura do conversor A/D, no canal 7 e com resolução de 12 bits. Na linha 12, é efetuada a conversão do valor lido, e nas linhas 13, 14 e 15, o valor de temperatura é dividido em centena, dezena e unidade, e é armazenado nas posições correspondentes do vetor para a transmissão dos dados.

Com os valores lidos e convertidos, a função de transmissão é chamada. O trecho principal dessa função pode ser visualizado na Figura 33. Após a transmissão dos dados, o OSAL executa a próxima tarefa em sua lista. E, periodicamente, voltará a executar a tarefa de leitura do conversor A/D e a transmissão dos dados.

Figura 32. Trecho de código do arquivo "GenericApp.C", da função de processamento da tarefa, onde é efetuada a leitura do conversor A/D (do autor).

---

```

1. uint16 GenericApp_ProcessEvent( uint8 task_id, uint16 events )
2. {
3.   afIncomingMSGPacket_t *MSGpkt;
4.   afDataConfirm_t *afDataConfirm;
5.   uint16 valor_adc;
6.   uint16 calc=0;
7.   byte sentEP;
8.   ZStatus_t sentStatus;
9.   byte sentTransID;
10.  (void)task_id;
11.  valor_adc=HalAdcRead(HAL_ADC_CHN_AIN7,HAL_ADC_RESOLUTION_12);
12.  calc=((valor_adc*3.3)/4096)*1000;
13.  MessageDataADC[5]=((calc%1000)/100)+0x30;
14.  MessageDataADC[6]=((calc%100)/10)+0x30;
15.  MessageDataADC[8]=((calc%10))+0x30;

```

---

Figura 33. Trecho de código do arquivo "GenericApp.C", da função de transmissão de dados, onde o vetor com as informações lidas do conversor A/D é transmitido para o coordenador (do autor).

---

```

1. static void GenericApp_SendTheMessage( void )
2. {
3.   if ( AF_DataRequest( &GenericApp_DstAddr, &GenericApp_epDesc,
4.     GENERICAPP_CLUSTERID,
5.     (byte)osal_strlen( MessageDataADC ) + 1,
6.     (byte *)&MessageDataADC,
7.     &GenericApp_TransID,
8.     AF_DISCV_ROUTE, AF_DEFAULT_RADIUS ) == afStatus_SUCCESS )
9.   }

```

---

No nodo coordenador, a tarefa de recepção dos dados é executada sempre que um novo pacote de dados é recebido. A função de tratamento dos dados recebidos pode ser vista na Figura 34. Nas linhas 11 e 14, são executados testes para verificar de qual *end device* foram enviadas as informações e, na sequência, essa informação é escrita no display LCD do módulo SMARTRF06EBK (linhas 13 e 16).

Com os módulos disponíveis, e utilizando a *Z-Stack-Mesh*, foi possível montar duas configurações de rede, mostradas nas Figuras 35 e 36. O processo de pareamento dos dispositivos foi feito por meio do mecanismo *ZigBee Binding* [97, 98], disponibilizado pela *Z-Stack-Mesh*, e acionado através de um botão conectado ao pino PC5 do SoC CC2538. Quando levado a nível lógico baixo, inicia o processo de pareamento. Se dois dispositivos iniciarem o processo de pareamento em um mesmo intervalo de tempo, eles irão formar a rede.

Utilizando o *sniffer* e a IDE *Packet Sniffer* disponibilizada em [99], foi possível capturar os pacotes de dados transmitidos na rede. Um pacote capturado é apresentado na Figura 37. Para que

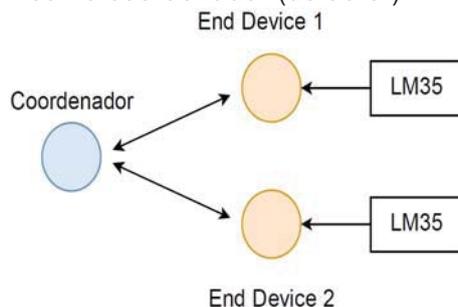
Figura 34. Função do *firmware* utilizado no coordenador da rede Zigbee, para exibir a atualização dos valores de temperatura enviados pelos *end-devices* (do autor).

```

1. static void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
2. {
3.   unsigned char *MessageIncoming;
4.   MessageIncoming=(pkt->cmd.Data);
5.   switch ( pkt->clusterId )
6.   {
7.   case GENERICAPP_CLUSTERID:
8.     rxMsgCount += 1;
9.     HalLedSet ( HAL_LED_4, HAL_LED_MODE_BLINK );
10.    #if defined( LCD_SUPPORTED )
11.    if(MessageIncoming[1]==0x31)
12.    {
13.    HalLcdWriteString( (char*)pkt->cmd.Data, HAL_LCD_LINE_1 );
14.    }else if(MessageIncoming[1]==0x32)
15.    {
16.    HalLcdWriteString( (char*)pkt->cmd.Data, HAL_LCD_LINE_2 );
17.    }
18.    HalLcdWriteStringValue( "Rcvd:", rxMsgCount, 10, HAL_LCD_LINE_3 );
19.    #elif defined( WIN32 )
20.    WPRINTSTR( pkt->cmd.Data );
21.    #endif
22.    break;
23.   }
24. }

```

Figura 35. Configuração de rede, utilizando os módulos desenvolvidos como *end device* e os módulos SMARTRF06EBK e CC2538EMK como coordenador (do autor).



possa ser feita a captura dos pacotes transmitidos, a segurança da rede deve obrigatoriamente estar desabilitada.

#### 4.7 CONCLUSÕES DO CAPÍTULO

O desenvolvimento de *firmwares* em dispositivos que utilizam sistemas de comunicações sem fio com padrões complexos é uma tarefa árdua. Neste sentido, a *Z-Stack-Mesh* e o OSAL são ferramentas de apoio ao desenvolvimento de grande valor. Com o uso dessas, o tempo de desen-

Figura 36. Configuração de rede, utilizando os módulos desenvolvidos como *end device*, um módulo CC2538EMK como roteador e os módulos SMARTRF06EBK e CC2538EMK como coordenador (do autor).

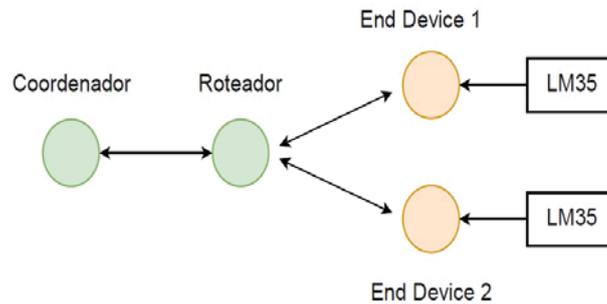


Figura 37. Pacote de dados capturado utilizando o *sniffer* e a IDE *Packet Sniffer* (do autor).

P.nbr.	Time (us)	Length	Frame control field					Sequence number	Dest. PAN	Dest. Address	Source Address	
RX	+657693		Type	Sec	Pnd	Ack.req	PAN	compr				
30	=5885527	57	DATA	0	0	1		1	0x75	0x26E0	0x0000	0x4AC7
MAC payload												
48 02 00 00 C7 4A 1E 22 28 9D 01 00 00 69 06 34 04 00 4B 12 00 00 31												
9B 67 70 A8 72 35 7E AC 08 9D 86 AF 18 36 06 D2 08 63 2D B5 0F 13 40												
NWK Frame control field					NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq.num				
Type	Version	DR	GA	Sec	Address	Address	Radius	Seq.num				
DATA	0x2	1	0	1	0	0	0x0000	0x4AC7	0x1E	0x22		
NWK payload												
28 9D 01 00 00 69 06 34 04 00 4B 12 00 00 31 9B 67 70 A8												
72 35 7E AC 08 9D 86 AF 18 36 06 D2 08 63 2D B5 0F 13 40												
APS Frame control field				APS Dest. Endpoint	APS Cluster Id	APS Profile Id	APS Src. Endpoint					
Type	Del.mode	Ind.am	Sec	Ack	Endpoint	Cluster Id	Profile Id	Endpoint				
Data	Broadcast	0	1	0	0x9D	0x01	0x0200	0x69				
APS Payload										LQI	FCS	
06 34 04 00 4B 12 00 00 31 9B 67 70 A8 72 35 7E										120	OK	
AC 08 9D 86 AF 18 36 06 D2 08 63 2D B5 0F 13 40												

volvimento é reduzido. Ainda, garante-se a conformidade do sistema de comunicação sem fio com padrões pré-estabelecidos pelas entidades competentes.

O OSAL permite a sua personalização, seja por meio da inclusão de novas tarefas, ou pela alteração de parâmetros de configurações, permitindo ao desenvolvedor utilizar-se dessa ferramenta como base para o desenvolvimento de novas aplicações, não deixando-o limitado a um número específico de rotinas de software pré-concebidas.

Para os testes de formação de rede, foi desenvolvida uma aplicação simples, para o monitoramento da temperatura ambiente, utilizando sensores de temperatura LM35. Com os módulos disponíveis, foi possível montar duas configurações de rede (Figuras 35 e 36). Para o desenvolvimento do *firmware*, foram utilizadas: a *Z-Stack-Mesh*; o *Operational System Abstraction Layer* (OSAL); e o *Hardware Abstraction Layer* (HAL), disponibilizados pela *Texas Instruments*. Essas ferramentas mostraram-se robustas e, apesar da complexidade, foi possível o desenvolvimento de uma aplicação personalizada.

Com os testes realizados, foi possível comprovar o funcionamento da *Z-Stack-Mesh*, com relação às tarefas de formação de rede e transmissão de dados. Os testes de formação de rede foram

executados utilizando poucos nodos, o que não os invalida, pois o algoritmo utilizado na formação da rede não sofre nenhuma alteração, independente do número de nodos presentes na rede.

## 5. CONSIDERAÇÕES FINAIS

Redes de comunicação sem fio que utilizam o protocolo Zigbee são as mais indicadas em aplicações que necessitam de dispositivos de baixo custo, com consumo de energia reduzido e confiabilidade na transmissão de dados. Neste projeto, foram desenvolvidos dois dispositivos que se encaixam nesse perfil. O primeiro dispositivo utiliza somente o SoC CC2538 e apresenta baixo consumo de energia, com bom alcance na transmissão de dados, superando os 100 metros. O segundo dispositivo, além do SoC CC2538, também faz uso do *range extender* CC2592, que amplifica a potência de saída do dispositivo, conseguindo maior alcance na transmissão de dados. Porém, esse segundo módulo apresenta um consumo superior ao anterior aproximadamente cinco vezes maior quando em sua potência máxima.

Os sistemas de alimentação de ambos os módulos foram projetados para permitir a utilização de diferentes fontes de energia. Isso foi possível com a utilização de um sistema de seleção. Em situações onde a fonte de energia utilizada possuir tensão superior a 3,6VCC e obrigatoriamente menor que 10VCC, faz-se o uso de um regulador de tensão, presente no próprio módulo, que reduz a tensão para um valor fixo de 3,3VCC. Quando a fonte de energia utilizada fornecer tensão menor do que 3,6VCC, o regulador de tensão não é utilizado. Assim, a fonte externa alimenta diretamente o circuito do módulo. A seleção da fonte de alimentação é feita por meio do jumper P1, que pode ser visualizado nas Figuras 21 e 23. Dessa forma, os módulos não ficam dependentes de uma fonte de alimentação específica, sendo possível utilizar desde pilhas e baterias comuns até sistemas de alimentação que utilizem células fotovoltaicas.

Outro ponto que cabe ser destacado é o baixo custo de desenvolvimento alcançado. Conforme os valores levantados, que podem ser visualizados nas Tabelas 6 e 7, pode-se afirmar que o desenvolvimento de um dispositivo de hardware próprio é uma opção com menor custo, quando comparada a outros dispositivos com características semelhantes disponíveis no mercado. Como exemplo, o conjunto de cinco módulos da marca *OpenMote*, que tem o custo de €500 (quinhentos euros) [86].

Utilizando a *Z-Stack-Mesh*, o *Operational System Abstraction Layer* (OSAL) e o *Hardware Abstraction Layer* (HAL), foi possível desenvolver uma aplicação simples para o monitoramento da temperatura ambiente. Utilizando os módulos desenvolvidos neste projeto, juntamente com os módulos CC2538EMK e SMARTRF06EBK, conseguiu-se implementar uma rede Zigbee com duas topologias diferentes, *star* e *mesh*. As bibliotecas disponibilizadas pela *Texas Instruments* mostraram-se robustas e, apesar de sua complexidade, foi possível o desenvolvimento de uma aplicação personalizada.

Neste trabalho, algumas etapas apresentaram maiores dificuldades. Os componentes eletrônicos utilizados neste projeto foram todos importados, pois, no Brasil, não encontrou-se nenhum fornecedor que disponibilizasse todos os componentes necessários. Devido a algumas especificidades presentes nos projetos das PCIs, também optou-se pela sua fabricação fora do Brasil. Isso acabou acarretando atrasos no desenvolvimento do projeto. Outra dificuldade encontrada foi no processo de

montagem dos componentes SMDs nas PCBs. Normalmente, esse tipo de montagem é feita de maneira automatizada, porém, neste projeto, essa etapa foi executada de forma manual, consumindo diversos dias de trabalho.

Com os resultados obtidos neste trabalho, pode-se afirmar que a plataforma construída possibilita o desenvolvimento de aplicações nas mais diversas áreas, tais como uma rede de sensores sem fio (RSSF), que pode ser utilizada como fonte de informações em sistemas de tomada de decisão para aplicação de defensivos agrícolas. A RSSF é composta por vários nodos interligados em malha e instalados no ambiente a ser monitorado. Por meio de sensores, cada nodo faz a coleta de dados, como temperatura ambiente e umidade, e envia essas informações periodicamente a um servidor, para que possam ser tratadas. Com a interpretação dos dados coletados na RSSF, pode-se decidir pela aplicação ou não de defensivos agrícolas, podendo, com isso, evitar o uso desnecessário desse tipo de material ou sinalizar a necessidade de uso, auxiliando no combate a doenças.

Assim, este trabalho apresenta como resultado uma plataforma de hardware que serve como base para o desenvolvimento de inúmeras aplicações, que necessitem de uma ou mais características apresentadas pelos módulos desenvolvidos. Também, pode-se afirmar que os dispositivos aqui desenvolvidos possuem grande potencial de industrialização e comercialização.

Por fim, conclui-se que o objetivo deste trabalho foi atingido, pois os dispositivos desenvolvidos apresentam baixo custo, baixo consumo de energia e confiabilidade na transmissão de dados sem fio, além da possibilidade de personalização do *firmware*, possibilitando sua utilização como uma plataforma universal de apoio ao desenvolvimento de novas aplicações.

## 6. TRABALHOS FUTUROS

Neste trabalho, foram alcançados alguns objetivos pré estabelecidos. O projeto voltava-se ao desenvolvimento de dispositivos de hardware personalizados, com capacidade de comunicação sem fio, baixo consumo de energia e baixo custo, tudo isso a partir do desenvolvimento de uma aplicação utilizando a *Z-Stack-Mesh*. Isso não encerra, contudo, o trabalho de pesquisa; ao contrário, aumenta a gama de trabalhos que podem ser desenvolvidos. Com isso, algumas sugestões de trabalhos futuros são apresentadas:

- Apesar de os módulos desenvolvidos possuírem tamanho relativamente reduzido, uma compactação ainda maior pode ser aplicada. Para que isso se torne possível, soluções como *balluns* integrados [100, 101] e antenas *on chip* [102] podem ser utilizadas. Outra possibilidade seria a retirada de alguns conectores da PCI, porém, isso poderia restringir o acesso às GPIOs do CC2538.
- Outra alternativa, com relação ao hardware, seria a inclusão de um circuito conversor serial (UART - USB) *on board*, possibilitando a utilização do sistema de *bootloader*. Com a inclusão desse circuito, a memória *flash* do CC2538 pode ser gravada apenas com o uso de um cabo USB e o software *SmartRF Flash Programmer* [103].
- Com relação ao *firmware*, as soluções *Z-Stack-Mesh*, *Operational System Abstraction Layer* e *Hardware Abstraction Layer* são totalmente funcionais. Porém, para a utilização dessas ferramentas, o desenvolvedor precisa ter conhecimento da linguagem de programação C. Talvez, uma forma de acelerar o desenvolvimento de aplicações utilizando o CC2538 seja disponibilizar essas bibliotecas oferecidas pela *Texas Instruments* na IDE Energia [104], que consiste em uma IDE *open source* e baseada em *Wiring* [44], que oferece uma estrutura robusta e um ambiente de codificação intuitivo. Sua interface é muito parecida com a disponibilizada pela plataforma Arduino. A IDE Energia já oferece suporte a vários microcontroladores fabricados pela *Texas Instruments*, porém, não ao CC2538.
- Outro ponto relevante em relação ao *firmware* é o desenvolvimento e a disponibilização de bibliotecas de *drivers* de sensores, as quais são essenciais no desenvolvimento de aplicações de coleta de dados. Isso visa acelerar o processo de desenvolvimento de novas aplicações e tornar atrativo o uso da plataforma desenvolvida a novos usuários.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FOROUZAN, S. C. F. B. A. *Comunicação de Dados e Redes de Computadores*. São Paulo: MC Graw Hill, 2008. 1115 p.
- [2] HAYKIN, M. M. S. *Sistemas Modernos de Comunicação Wirelles*. São Paulo: Bookman, 2008. 565 p.
- [3] A revolução da Internet das Coisas: oportunidade de negócio para startups. 2015. Disponível em: <<http://startupsebraeminas.com.br/revolucao-da-internet-das-coisas-oportunidade-de-negocio-para-startups/>>. Acesso em: Ago. 17, 2015.
- [4] LOUREIRO, A. A. F. et al. Redes de sensores sem fio. In: XXI SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES. Natal: UFMG, 2003. 48 p.
- [5] SALOMÃO, A. *Desenvolvimento de um módulo Zigbee para o monitoramento Remoto do Consumo de Água em Instalações Prediais Aeroportuárias*. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica–ITA, São José dos Campos, SP, Brasil, 2009.
- [6] SANTOS, E. L. F. dos. A ieee 802.15.4 como plataforma de comunicação de dados. *Revista Ilha Digital*, v. 4, p. 97–105, 2013.
- [7] WHAT is Arduino? 2015. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: Out. 02, 2015.
- [8] XBEE. 2015. Disponível em: <<http://www.digi.com/lp/xbee>>. Acesso em: Ago. 04, 2015.
- [9] AKYILDIZ, M. C. V. I. F. *Wireless Sensor Networks*. Torquay, UK: John Wiley & Sons Ltd, 2010.
- [10] ROCCIA, C. J. *Avaliação de Redes de Sensores Sem Fio Aplicadas a Cultivos de Milho, Eucalipto e Pinhão*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Limeira, SP, Brasil, 2011.
- [11] AKYILDIZ, I. et al. Wireless sensor networks: a survey. *Computer Networks*, v. 38, n. 4, p. 393–422, Jun. 2002.
- [12] NXP. *Data Sheet: JN5168-001-Myy JenNet-IP, ZigBee PRO and IEEE802.15.4 Module*. Sheffield, UK, 2013. Disponível em: <<http://www.nxp.com/documents/other/JN-DS-JN5168MO-1v2.pdf>>.
- [13] TELIT. *ZE51-2.4 | ZE61-2.4*. Morrisville, NC, USA, 2013. Disponível em: <<http://www.telit.com/products/product-service-selector/product-service-selector/show/product/ze51-24-ze61-24/>>.
- [14] ATMEL. *Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications, AT86RF230*. San Jose, CA, USA, 2009. Disponível em: <<http://www.atmel.com/images/doc5131.pdf>>.

- [15] SEMICONDUCTOR, F. *MC13234/MC13237 Low Cost SoC Remote Control Platform for the 2.4 GHz IEEE® 802.15.4 Standard*. Chandler, Arizona, USA, 2013. Disponível em: <<http://www.freescale.com>>.
- [16] SEMICONDUCTOR, F. *MKW2xDxxx Data Sheet*. Chandler, Arizona, USA, 2013. Disponível em: <<http://www.freescale.com>>.
- [17] INSTRUMENTS, T. *CC2531 - A USB-Enabled System-On-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee*. Dallas, Texas, USA, 2013. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2531.pdf>>.
- [18] INSTRUMENTS, T. *CC2538 System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee/ZigBee IP*. Dallas, Texas, USA, 2013. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2538.pdf>>.
- [19] SEMICONDUCTOR, F. *MC1322x Advanced ZigBee™ - Compliant SoC Platform for the 2.4 GHz IEEE® 802.15.4 Standard Reference Manual*. San Jose, CA, USA, 2012. Disponível em: <<http://www.nxp.com/files/rfif/doc/refmanual/MC1322xRM.pdf>>.
- [20] INSTRUMENTS, T. *CC2630 SimpleLink™ 6LoWPAN / ZigBee® Wireless MCU*. Dallas, Texas, USA, 2015. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2630.pdf>>.
- [21] INC., M. *MICAZ Wireless Measurement System*. San Jose, California, USA, 2013. Disponível em: <<http://www.memsic.com/userfiles/files/Datasheets/WSN/micazdatasheet-t.pdf>>.
- [22] INC., M. *TELOSB Mote Platform*. San Jose, California, USA, 2013. Disponível em: <[http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb\\_datasheet.pdf](http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf)>.
- [23] RABER, A. et al. Aplicação de redes de sensores sem fio (rssf) na agricultura. *Encontro Anual de Tecnologia da Informação – EATI*, v. 3, n. 1, p. 261–265, 2013.
- [24] CARVALHO, F. B. S. et al. Aplicações ambientais de redes de sensores sem fio. *Revista de Tecnologia da Informação e Comunicação*, v. 2, n. 1, p. 14–19, 2012.
- [25] BARROS, M. F. de. *Proposta de Interconexão do Padrão ISO 11783 com Redes de Sensores sem Fio Padrão Zigbee*. Dissertação (Mestrado) — Universidade de São Paulo, São Paulo, SP, Brasil, 2010.
- [26] LAZZAROTTO, P. *Algoritmos de Roteamento Hierárquicos em Redes de Sensores Sem Fio Utilizando Algoritmos Evolutivos para Determinação de Cluster-Head's*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, PR, Brasil, 2008.
- [27] KARL, A. W. H. *Protocols and Architectures for Wireless Sensor Networks*. West Sussex, UK: John Wiley & Sons Ltd, 2005.

- [28] SILVA, M. A. M. da. *Rede de Sensores para Aplicação em Agricultura: Um Estudo de Caso*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil, 2009.
- [29] GISLASON, D. *Zigbee Wireless Networking*. Boston, USA: Newnes, 2007.
- [30] TEIXEIRA, L. M. *Desenvolvimento de uma Aplicação com o Protocolo Zigbee Aplicado em Instrumentação de Ensaio em Vôo*. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica-ITA, São José dos Campos, SP, Brasil, 2006.
- [31] Roel Peeters Katelijin Vleugels. *Apparatus and method for integrating short-range wireless personal area networks for a wireless local area network infrastructure*. 2010.
- [32] IEEE 802.15™: Wireless Personal Area Networks (PANs). Disponível em: <<https://standards.ieee.org/about/get/802/802.15.html>>. Acesso em: Jun. 10, 2015.
- [33] ASSOCIATION, I. S. *Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*. New York, NY, EUA, 2005. Disponível em: <<https://standards.ieee.org/getieee802/download/802.15.1-2005.pdf>>.
- [34] ASSOCIATION, I. S. *IEEE Standard for Low-Rate Wireless Networks*. New York, NY, EUA, 2015. Disponível em: <<http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf>>.
- [35] ZIGBEE Alliance. 2016. Disponível em: <<http://www.zigbee.org/>>. Acesso em: Jun. 05, 2015.
- [36] FARAHANI, S. *Zigbee Wireless Networks and Transceivers*. Boston, EUA: Newnes, 2008.
- [37] RIVERO, I. A. S. *Rede de Sensores sem Fio para Monitoramento de Equipamentos Eletrônicos*. Dissertação (Mestrado) — Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, MG, Brasil, 2011.
- [38] TOSE, T. et al. Redes de sensores sem fio zigbee aplicada em uma estação de tratamento de esgoto. In: XIX CONGRESSO BRASILEIRO DE AUTOMÁTICA. Campina Grande: CBA, 2012. 8 p.
- [39] NETO, A. S. de A.; CASILLO, D. S. da S.; ARAÚJO, S. R. F. de. Proposta de um sistema de automação residencial baseado no padrão zigbee e computadores pessoais. In: XIX CONGRESSO BRASILEIRO DE AUTOMÁTICA. Campina Grande: CBA, 2012. 7 p.
- [40] GUEDES, A. L. *Uma Plataforma de Hardware e Software para desenvolvimento de Rede de Sensores sem Fio Aplicada à Agrometeorologia*. Dissertação (Mestrado) — Universidade Estadual de Londrina, Londrina, PR, Brasil, 2013.
- [41] FLORIDO, I. R. *Rede de Sensores Sem Fio em Ambientes Veiculares Baseada no Padrão Zigbee*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brasil, 2008.

- [42] TEIXEIRA, I. *Roteamento com Balanceamento de Consumo de Energia para Redes de Sensores sem Fio*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2005.
- [43] PINHO, L. B. de; AMORIM, C. L. de. Otimizando o consumo de energia de dispositivos móveis em sistemas de voo baseados em pontos de acesso co-localizados. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Curitiba, PR, Brasil: SBRC, 2006. v. 24.
- [44] WIRING. 2015. Disponível em: <<http://wiring.org.co/>>. Acesso em: Out. 05, 2015.
- [45] PROCESSING. 2015. Disponível em: <<https://processing.org/>>. Acesso em: Out. 05, 2015.
- [46] THE Making of Arduino. 2015. Disponível em: <<http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>>. Acesso em: Out. 05, 2015.
- [47] OPENMOTE. 2015. Disponível em: <<http://www.openmote.com/>>. Acesso em: Ago. 15, 2015.
- [48] CONTIKI. 2015. Disponível em: <<http://www.contiki-os.org/>>. Acesso em: Ago. 03, 2015.
- [49] OPENWSN. 2015. Disponível em: <<https://openwsn.atlassian.net/wiki/>>. Acesso em: Ago. 03, 2015.
- [50] FREERTOS. 2015. Disponível em: <<http://www.freertos.org/>>. Acesso em: Ago. 03, 2015.
- [51] RIOT. 2015. Disponível em: <<http://www.riot-os.org/>>. Acesso em: Ago. 03, 2015.
- [52] WORKGROUP, L. A. T. M. *A technical overview of LoRa and LoRaWAN*. San Ramon, CA, EUA, 2016. 20 p.
- [53] LORA Alliance Wide Area Networks for IoT. 2016. Disponível em: <<https://www.lora-alliance.org/>>. Acesso em: Mar. 20, 2016.
- [54] RN2486. 2015. Disponível em: <<http://www.microchip.com/wwwproducts/en/RN2483>>. Acesso em: Ago. 10, 2016.
- [55] RN2903. 2015. Disponível em: <<http://www.microchip.com/wwwproducts/en/RN2903>>. Acesso em: Ago. 10, 2016.
- [56] S.L., L. C. D. *Networking Guide*. Zaragoza, Espanha, 2016. 57 p.
- [57] MOUSER Electronics. 2016. Disponível em: <<http://br.mouser.com/>>. Acesso em: Nov. 05, 2016.
- [58] NEWARK Element 14. 2016. Disponível em: <<http://www.farnell.com/>>. Acesso em: Nov. 05, 2016.
- [59] CC2538 - A Powerful System-On-Chip for 2.4-GHz IEEE 802.15.4-2006 and ZigBee Applications. 2016. Disponível em: <<http://www.ti.com/product/CC2538?keyMatch=cc2538&tisearch=Search-EN-Everything>>. Acesso em: Dez. 04, 2016.

- [60] INSTRUMENTS, T. *CC2538 System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee®/ZigBee IP® Applications - User's Guide*. Dallas, Texas, USA, 2013. Disponível em: <<http://www.ti.com/lit/ug/swru319c/swru319c.pdf>>.
- [61] INSTRUMENTS, T. *CC2592 2.4-GHz Range Extender*. Dallas, Texas, USA, 2014. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2592.pdf>>.
- [62] CC2592 - 2.4 GHz Range Extender. 2016. Disponível em: <<http://www.ti.com/product/CC2592?keyMatch=cc2592&tisearch=Search-EN-Everything>>. Acesso em: Dez. 20, 2016.
- [63] WALLACE, R. *Antenna Selection Guide*. Dallas, Texas, USA, 2010. Disponível em: <<http://www.ti.com/lit/an/swra161b/swra161b.pdf>>.
- [64] ANDERSEN, A. *Small Size 2.4 GHz PCB antenna*. Dallas, Texas, USA, 2008. Disponível em: <<http://www.ti.com/lit/an/swra117d/swra117d.pdf>>.
- [65] ANDERSEN, A. *2.4 GHz Inverted F Antenna*. Dallas, Texas, USA, 2008. Disponível em: <<http://www.ti.com/lit/an/swru120b/swru120b.pdf>>.
- [66] WIUM, E. *Using CC253X or CC254X with Dipole PCB Antennas*. Dallas, Texas, USA, 2013. Disponível em: <<http://www.ti.com/lit/an/swra421/swra421.pdf>>.
- [67] WALLACE, R.; DUNBAR, S. *2.4 GHz YAGI PCB Antenna*. Dallas, Texas, USA, 2010. Disponível em: <<http://www.ti.com/lit/an/swra350/swra350.pdf>>.
- [68] BALANIS, C. A. *Teoria de Antenas, Terceira Edição, Volume I*. Rio de Janeiro, RJ, Brasil: LTC - Livros Técnicos e Científicos Editora S.A., 2009.
- [69] JUNIOR, L. E. F. *Fundamentos de Comunicação Eletrônica, Terceira Edição*. Porto Alegre, RS, Brasil: AMGH Editora Ltda., 2013.
- [70] RIBEIRO, J. A. J. *Engenharia de Micro-Ondas, Primeira Edição*. São Paulo, SP, Brasil: Editora Érica Ltda., 2012.
- [71] SCHMITT, R. *Electromagnetics explained: a handbook for wireless/RF, EMC, and high-speed electronics*. Woburn, MA, EUA: Newnes, 2002.
- [72] BALANIS, C. A. *Teoria de Antenas, Terceira Edição, Volume 2*. Rio de Janeiro, RJ, Brasil: LTC - Livros Técnicos e Científicos Editora S.A., 2009.
- [73] SCHMID, R. *Adapting TI LPRF Reference Designs for Layer Stacking*. Dallas, Texas, USA, 2009. Disponível em: <<http://www.ti.com/lit/an/swra236a/swra236a.pdf>>.
- [74] CC2650 Programming. 2015. Disponível em: <[https://e2e.ti.com/support/wireless\\_connectivity/bluetooth\\_low](https://e2e.ti.com/support/wireless_connectivity/bluetooth_low)>. Acesso em: Mai. 25, 2015.

- [75] INSTRUMENTS, T. *CC2538 ROM User's Guide*. Dallas, Texas, EUA, 2014. Disponível em: <<http://www.ti.com/lit/ug/swru333a/swru333a.pdf>>.
- [76] INSTRUMENTS, T. *Serial Boot Loader For CC2538 SoC*. Dallas, Texas, EUA, 2013. Disponível em: <<http://www.ti.com/tool/z-stack>>.
- [77] BOE, A. *CC2538/CC26xx Serial Bootloader Interface*. Dallas, Texas, EUA, 2015. Disponível em: <<http://www.ti.com/lit/an/swra466a/swra466a.pdf>>.
- [78] INSTRUMENTS, T. *TLV702 300-mA, Low-IQ, Low-Dropout Regulator*. Dallas, Texas, EUA, 2015. Disponível em: <<http://www.ti.com/lit/ds/symlink/tlv702.pdf>>.
- [79] ALTIUM Designer. 2015. Disponível em: <<http://www.altium.com/>>. Acesso em: Out. 10, 2015.
- [80] CC2531 USB Evaluation Module Kit. 2016. Disponível em: <<http://www.ti.com/tool/cc2531emk>>. Acesso em: Ago. 28, 2016.
- [81] CC2531 USB Dongle Reference Design. 2016. Disponível em: <<http://www.ti.com/tool/cc2531usb-rd>>. Acesso em: Set. 01, 2016.
- [82] CC USB Firmware Library and Examples (Rev. C). 2016. Disponível em: <<http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=swrc088&fileType=zip>>. Acesso em: Ago. 28, 2016.
- [83] CC2538 PER Test (v1.1.0) (Rev. A). 2016. Disponível em: <<http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=swrc272&fileType=zip>>. Acesso em: Dez. 12, 2016.
- [84] SMARTRF06 Evaluation Board. 2015. Disponível em: <<http://www.ti.com/tool/smartrf06ebk>>. Acesso em: Dez. 02, 2015.
- [85] CC2538 Evaluation Module Kit. 2016. Disponível em: <<http://www.ti.com/tool/cc2538emk>>. Acesso em: Jan. 28, 2016.
- [86] OPENMOTE Silver kit. 2017. Disponível em: <<http://www.openmote.com/shop/openmote-silver-kit.html>>. Acesso em: Fev. 13, 2017.
- [87] CC2538 Foundation Firmware. 2015. Disponível em: <<http://www.ti.com/tool/cc2538-sw>>. Acesso em: Jul. 19, 2015.
- [88] CODE Composer Studio (CCS) Integrated Development Environment (IDE). 2015. Disponível em: <<http://www.ti.com/tool/ccstudio?keyMatch=code>>. Acesso em: Nov. 15, 2015.
- [89] ARM Cortex-M edition. 2015. Disponível em: <<https://www.iar.com/iar-embedded-workbench/tools-for-arm/arm-cortex-m-edition/>>. Acesso em: Out. 11, 2015.
- [90] IEEE802.15.4 Medium Access control (MAC) software stack. 2015. Disponível em: <<http://www.ti.com/tool/timac>>. Acesso em: Jan. 18, 2016.

- [91] ARCHIVE for Z-Stack Fully Compliant ZigBee 2012 Solution. 2015. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>. Acesso em: Mar. 20, 2015.
- [92] INSTRUMENTS, T. *Z-Stack Application Programming Interface*. San Diego, California, EUA, 2012. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [93] INSTRUMENTS, T. *802.15.4 MAC Application Programming Interface*. San Diego, California, EUA, 2013. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [94] INSTRUMENTS, T. *OS Abstraction Layer Application Programming Interface*. San Diego, California, EUA, 2011. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [95] INSTRUMENTS, T. *HAL Drivers Application Programming Interface*. San Diego, California, EUA, 2013. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [96] INSTRUMENTS, T. *LM35 Precision Centigrade Temperature Sensors*. Dallas, Texas, EUA, 1999. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm35.pdf>>.
- [97] INSTRUMENTS, T. *Z-Stack Developer's Guide*. San Diego, California, EUA, 2012. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [98] INSTRUMENTS, T. *Z-Stack Generic Application User's Guide*. San Diego, California, EUA, 2014. Disponível em: <<http://www.ti.com/tool/z-stack-archive>>.
- [99] SMARTRF Protocol Packet Sniffer. 2016. Disponível em: <<http://www.ti.com/tool/packet-sniffer>>. Acesso em: Jan. 02, 2017.
- [100] INSTRUMENTS, T. *Johanson Technology, Inc. Highly temperature-stable Impedance Matched RF Front End Differential Balun-Band Pass Filter Integrated Ceramic Component*. Dallas, Texas, EUA, 2012. Disponível em: <<http://www.ti.com/lit/an/swra297b/swra297b.pdf>>.
- [101] KERVEL, F. *Murata Balun for CC253x and CC254x LFB182G45BG2D280*. Dallas, Texas, EUA, 2011. Disponível em: <<http://www.ti.com/lit/an/swra380a/swra380a.pdf>>.
- [102] INSTRUMENTS, T. *Compact Reach Xtend TM Bluetooth®, 802.11b/g WLAN Chip Antenna*. Dallas, Texas, EUA, 2008. Disponível em: <<http://www.ti.com/lit/an/swra092b/swra092b.pdf>>.
- [103] SMARTRF Flash Programmer. 2016. Disponível em: <<http://www.ti.com/tool/flash-programmer>>. Acesso em: Jan. 07, 2017.
- [104] ENERGIA. 2016. Disponível em: <<http://www.energia.nu/>>. Acesso em: Jan. 07, 2017.



## APÊNDICE A – LISTAS DETALHADAS DOS COMPONENTES E MÓDULOS ADQUIRIDOS PARA O DESENVOLVIMENTO DO TRABALHO

Tabela 12. Detalhamento do material necessário adquirido para o projeto do módulo *sniffer*, utilizando o SoC CC2531 (do autor).

Quantidade	Descrição	Fabricante
1	Capacitor, 1uF, 0603, X5R, 15%, 6.3V	Murata
1	Capacitor, 4,7uF, 0603, X5R, 10%, 6.3V	Murata
1	Capacitor, 2,2uF, 0402, X5R, +/-20%, 4V	Murata
1	Capacitor, 0,5pF, 0402, NP0, +/-0,05pf, 50V	Murata
2	Capacitor, 47pF, 0402, NP0, 5%, 50V	Murata
1	Capacitor, 10pF, 0402, NP0, 5%, 50V	Murata
5	Capacitor, 100nF, 0402, X5R, 10%, 10V	Murata
1	Capacitor, 1nF, 0402, NP0, 1%, 50V	Murata
2	Capacitor, 27pF, 0402, NP0, 5%, 50V	Murata
1	Capacitor, 220pF, 0402, NP0, 5%, 50V	Murata
2	Capacitor, 1uF, 0402, X5R, 10%, 6.3V	Murata
1	EMI, filter bead, 0402, 1Kohms, Tape GHz Band	Murata
1	Inductor, 6,8nH, 0402, Monolithic type, ±5%	Murata
1	Resistor, 2 ohms, 0402, ±1%	Koa
3	Resistor, 0 ohm, 0402	Koa
2	Resistor, 270 ohms, 0402, ±1%	Koa
2	Resistor, 33 ohms, 0402, 2%	Koa
1	Resistor, 1k5 ohms, 0402, 2%	Koa
1	Resistor, 56k ohms, 0402, ±1%	Koa
1	Resistor, 2k2 ohms, 0402, 2%	Koa
1	TPS76933DBVR	Texas Instruments
1	Crystal, 32.000MHz, NX3225DA, 10/15ppm, 16 pF,	NDK
1	CC2531	Texas Instruments

Tabela 13. Detalhamento do material necessário adquirido para o projeto do módulo Zigbee, utilizando o SoC CC2538 (do autor).

Quantidade	Descrição	Fabricante
1	Capacitor, X5R, 2.2uF, 10V, +-20%, 0603	Murata
1	Capacitor, X5R, 1uF, 15%, 10V, 0603	Murata
2	Capacitor, C0G/NP0, 47pF, 50V, +-5%, 0402	Murata
1	Capacitor, X5R, 4.7uF, 6.3V, +-10%, 0603	Murata
7	Capacitor, X7R, 100nF, 6.3V, +-10%, 0402	Murata
1	Capacitor, X7R, 1nF, 50V, +-10%, 0402	Murata
3	Capacitor, X5R, 1uF, 10V, +-10%, 0402	Murata
2	Capacitor, C0G/NP0, 12pF, 50V, +-5%, 0402	Murata
2	Capacitor, C0G/NPO, 18pF, 50V, +-5%, 0402	Murata
2	Capacitor, C0G/NP0, 1pF, 50V, +-0.1pF, 0402	Murata
1	Capacitor, C0G/NP0, 1.2pF, 50V, +-0.25pF, 0402	Murata
1	Capacitor, C0G/NP0, 220pF, 50V, +-5%, 0402	Murata
2	Capacitor, C0G/NP0, 22pF, 50V, +-5%, 0402	Murata
1	EMI filter, 1500@100MHz, 0603	Murata
3	Indutor, 2nH, +-0.3nH, 0.3A, 0402	Murata
1	Indutor, 3.3nH, +-0.3nH, 0.3A, 0402	Murata
2	Resistor, 33 ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 1.5k ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 2 ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 2.2k ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 56k ohms, +-1%, 0.063W, 50V, 0402	Koa
2	Resistor, 0 ohms, 0402	Koa
1	TPS76933DBVT	Texas Instruments
1	Cristal, 32MHz, 10pF, +-10PPM	Epson, Toyocom
1	Cristal, 32.768kHz, +-20PPM, 12.5pF	Epson, Toyocom
1	CC2538	Texas Instruments

Tabela 14. Detalhamento do material necessário adquirido para o projeto do módulo Zigbee,, utilizando o SoC CC2538 em conjunto com o *range extender* CC2592 (do autor).

Quantidade	Descrição	Fabricante
2	Capacitor, X5R, 2.2uF, 10V, +-20%, 0603	Murata
2	Capacitor, X5R, 1uF, 10V, +-15%, 0603	Murata
2	Capacitor, C0G/NP0, 47pF, 50V, +-5%, 0402	Murata
7	Capacitor, X7R, 100nF, 6.3V, +-10%, 0402	Murata
1	Capacitor, X7R, 1nF, 50V, +-10%, 0402	Murata
4	Capacitor, X5R, 1uF, 10V, +-10%, 0402	Murata
4	Capacitor, C0G/NP0, 12pF, 50V, +-5%, 0402	Murata
1	Capacitor, C0G/NP0, 220pF, 50V, +-5%, 0402	Murata
2	Capacitor, C0G/NP0, 22pF, 50V, +-5%, 0402	Murata
2	Capacitor, C0G/NP0, 2.2pF, 50V, +-0.25pF, 0402	Murata
1	Capacitor, C0G/NP0, 0.2pF, 50V, +-5%, 0402	Murata
1	Capacitor, C0G/NP0, 1nF, 50V, +-5%, 0402	Murata
1	Capacitor, C0G/NP0, 18pF, 50V, +-5%, 0402	Murata
1	Capacitor, X7R, 1nF, 50V, +-5%, 0402	Murata
2	EMI filter, 1500@100MHz, 0603	Murata
1	Indutor, 4.7nH, +-0.3nH, 0.3A, 0402	Murata
1	Indutor, 1nH, +-0.3nH, 0.3A, 0402	Murata
1	Indutor, 2nH, +-0.3nH, 0.3A, 0402	Murata
2	Resistor, 33 ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 1.5k ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 2.2k ohms, +-1%, 0.063W, 50V, 0402	Koa
1	Resistor, 56k ohms, +-5%, 0.063W, 50V, 0402	VISHAY
2	Resistor, 0 ohms, 0402	Koa
3	Resistor, 10k ohms, +-5%, 0.063W, 50V, 0402	VISHAY
1	Resistor, 3.9k ohms, +-1%, 0.063W, 50V, 0402	VISHAY
1	Mechanical Frame SMD	LAIRD
1	Mechanical Cover	LAIRD
1	TLV70233DBVR	Texas
1	Cristal, 32MHz, 10pF, +-10PPM	Epson, Toyocom
1	Cristal, 32.768kHz, +-20PPM, 12.5pF	Epson, Toyocom
1	CC2538	Texas
1	CC2592	Texas Instruments

Tabela 15. Módulos adquiridos para suporte no desenvolvimento do projeto (do autor).

Quantidade	Descrição	Fabricante
1	Emulators,/ Simulators XDS100V3 DSP / ARM HIGH SPEED USB JTAG	Olimex Ltda.
1	Development,Boards & Kits - Wireless CC2531 Eval Mod	Texas Instruments
1	Zigbee/802.15.4 Development Tools CC2538 Eval Mod Kit	Texas Instruments
1	Zigbee/802.15.4 Development Tools CC2538-CC2592 Eval Mod Kit	Texas Instruments
1	RF,Development Tools SMARTRF06 EVAL BRD	Texas Instruments



## APÊNDICE B – DETALHES DO PROJETO DO MÓDULO UTILIZANDO O SOC CC2538

Figura 38. Diagrama elétrico parcial (1/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito do microcontrolador (do autor).

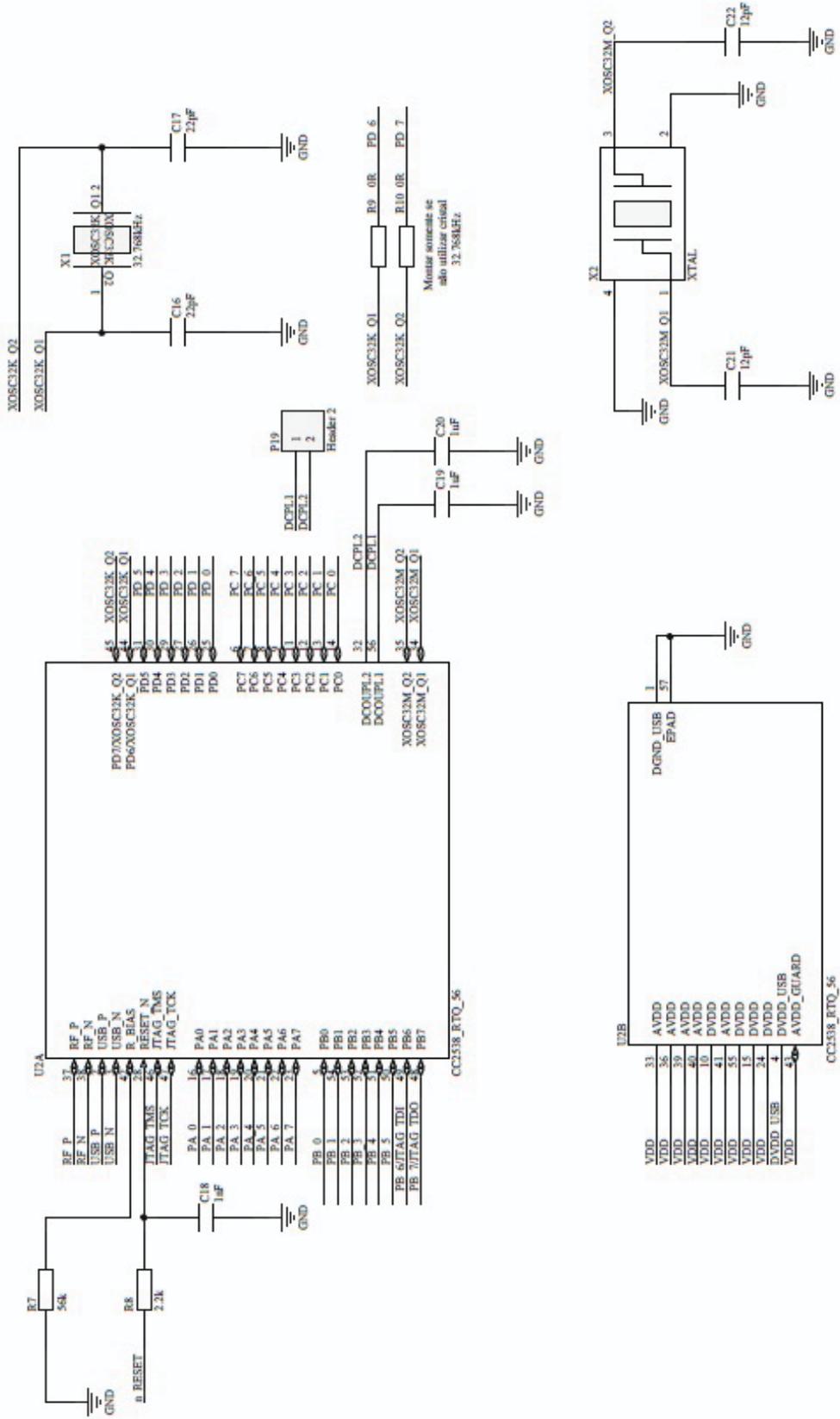


Figura 39. Diagrama elétrico parcial (2/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito do conector USB (do autor).

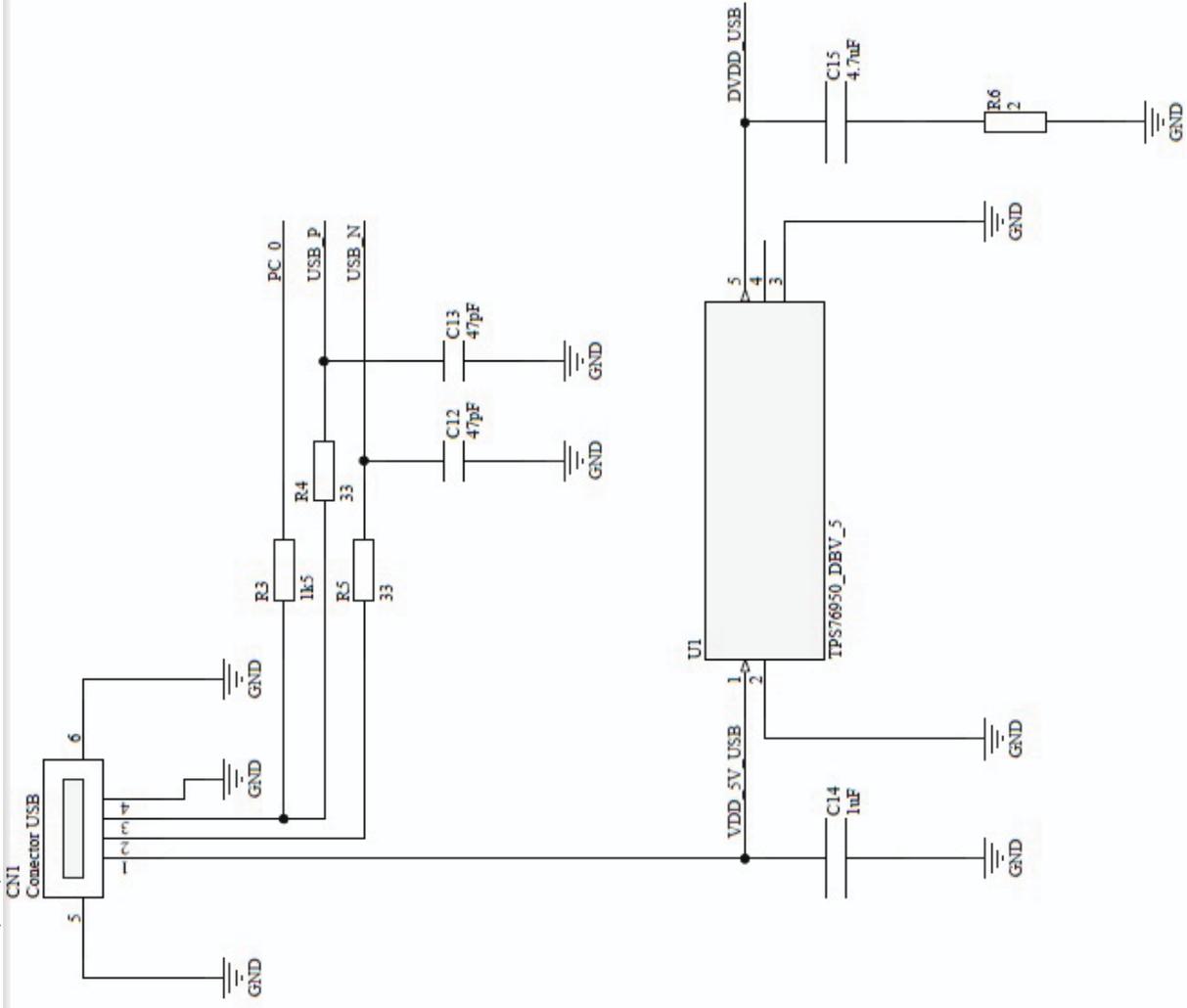


Figura 40. Diagrama elétrico parcial (3/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito de alimentação externa, seleção da alimentação e capacitores de desacoplamento (do autor).

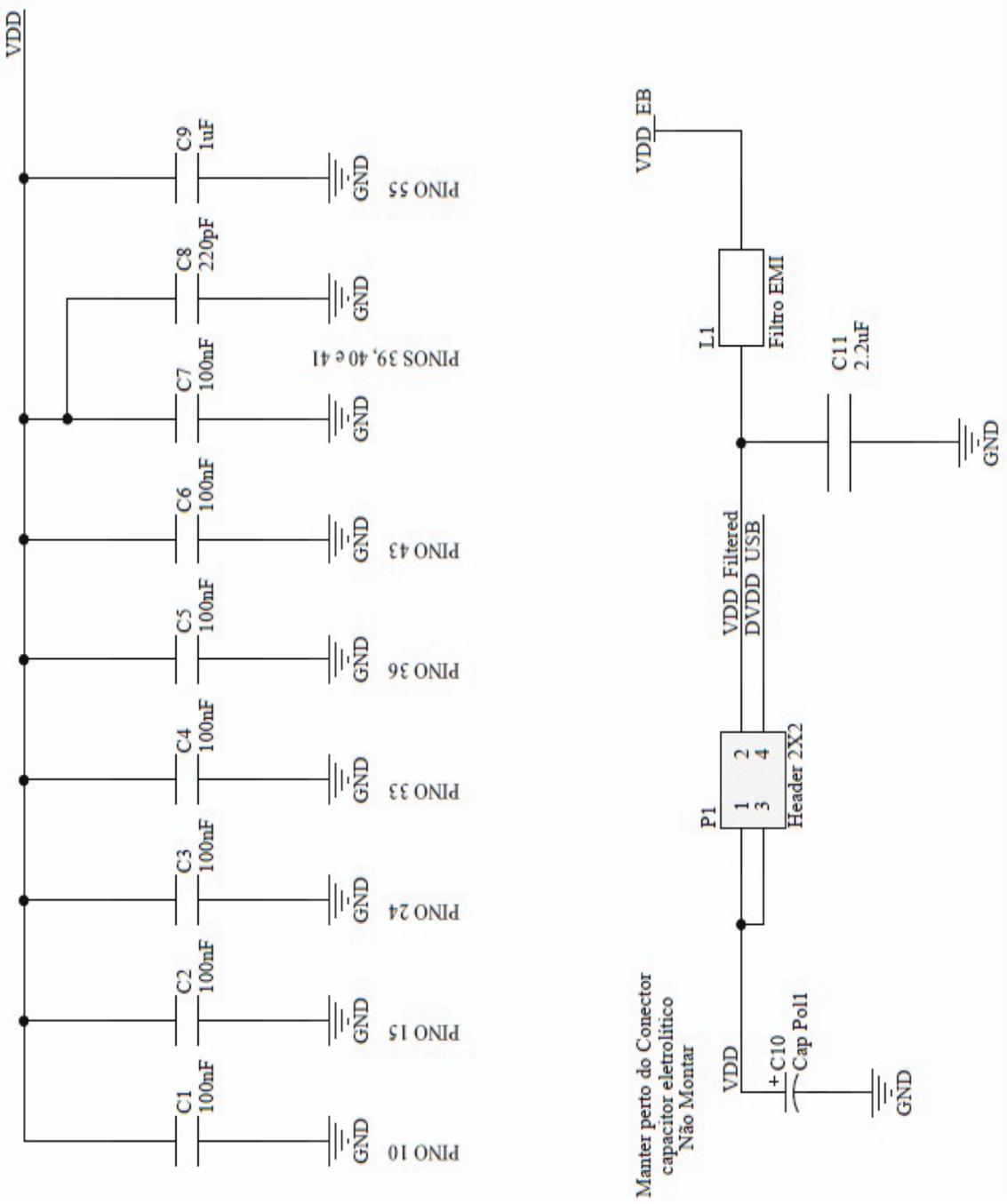


Figura 41. Diagrama elétrico parcial (4/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito dos conectores (do autor).

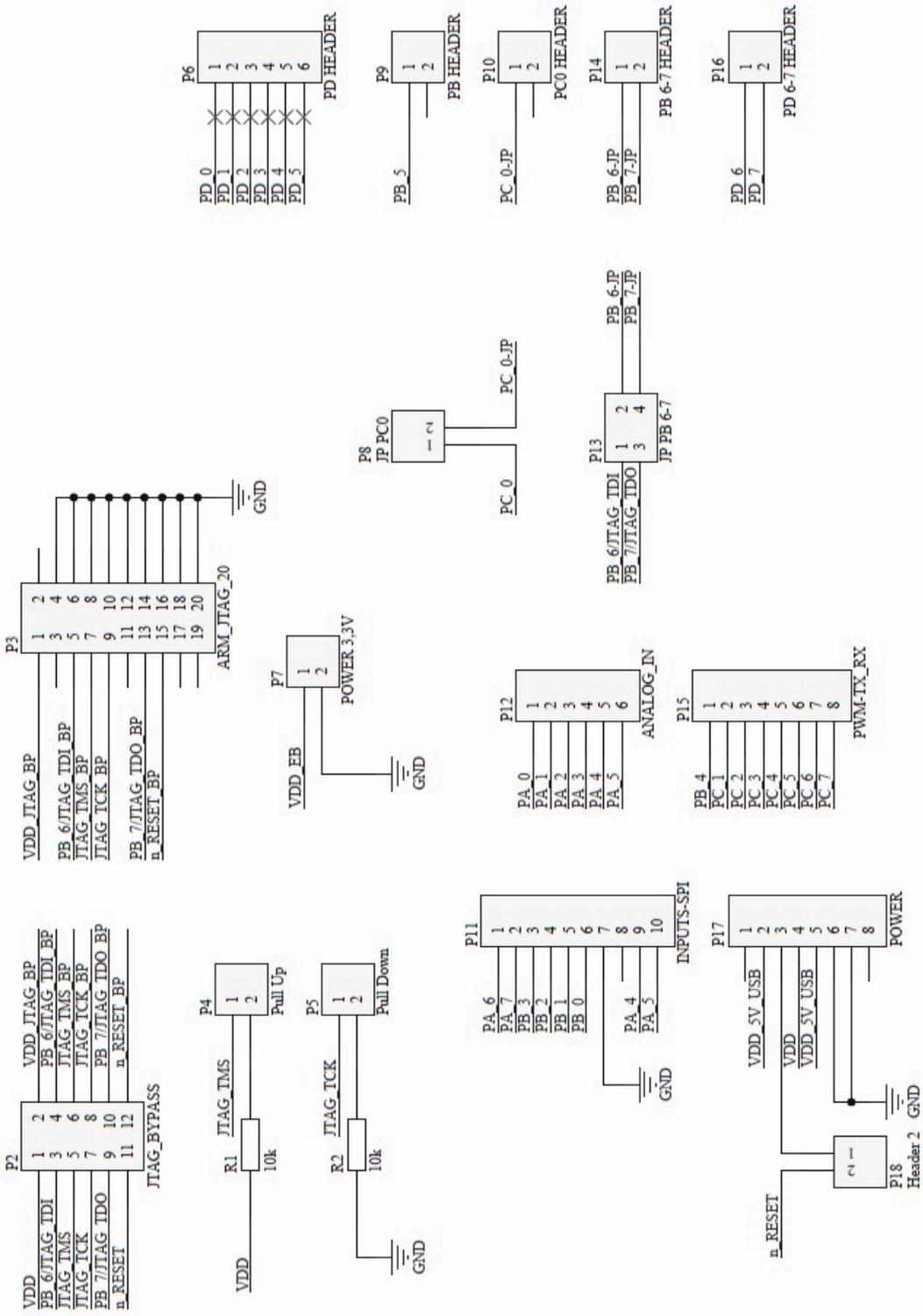


Figura 42. Diagrama elétrico parcial (5/5) do módulo desenvolvido utilizando o SoC CC2538. Referente ao circuito de RF (do autor).

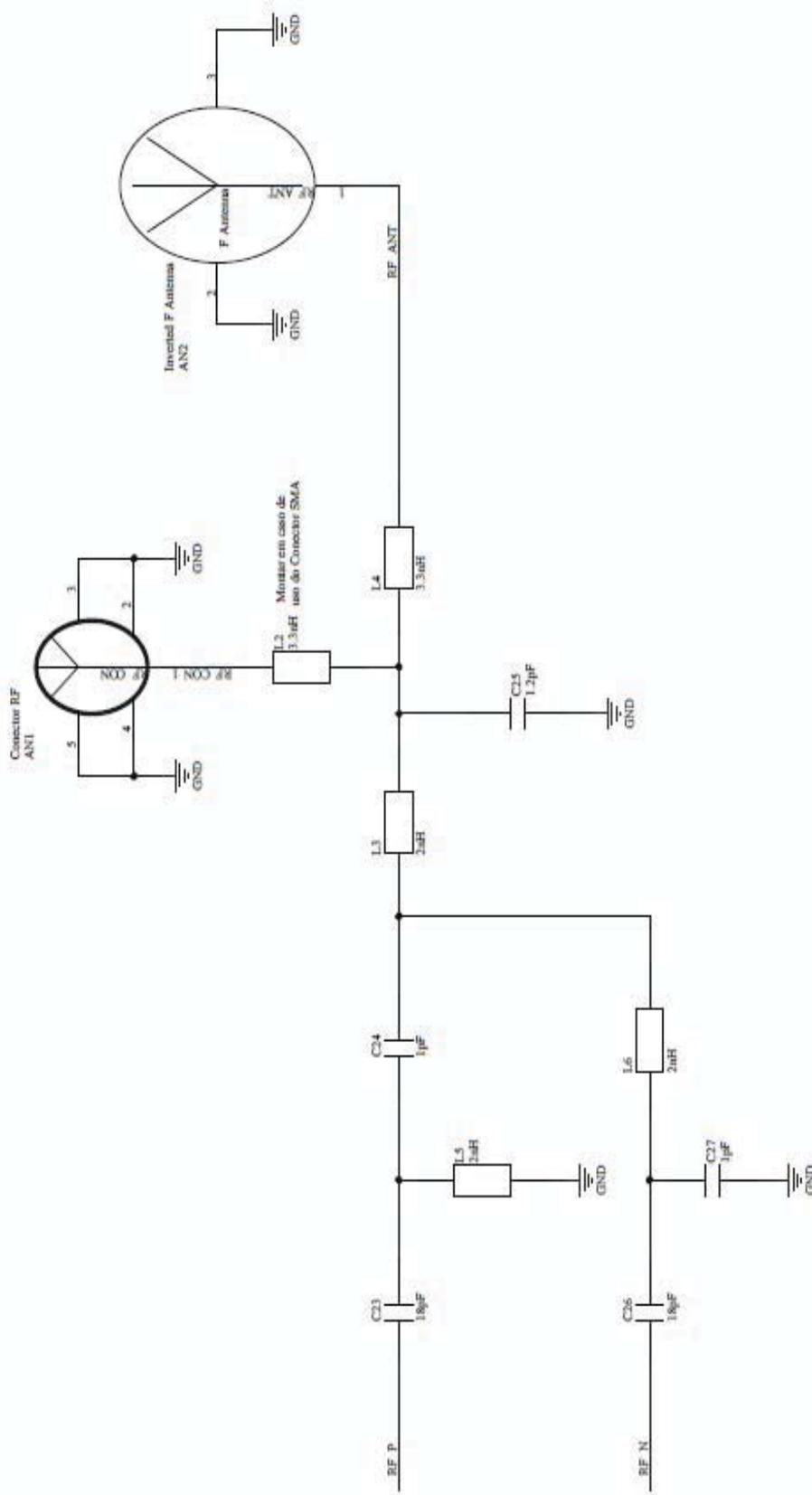


Figura 43. *Layer top* da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538. (do autor).

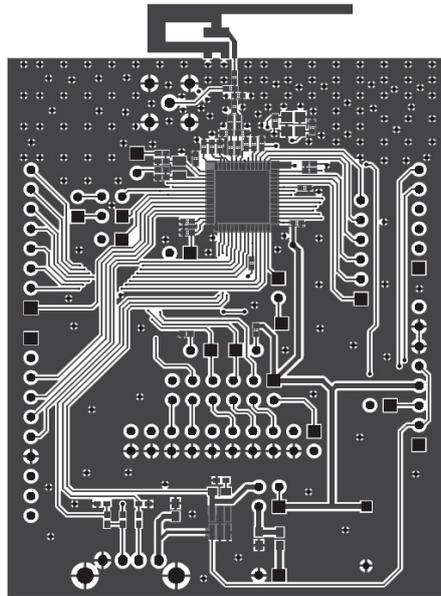


Figura 44. *Layer bottom* da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538. (do autor).

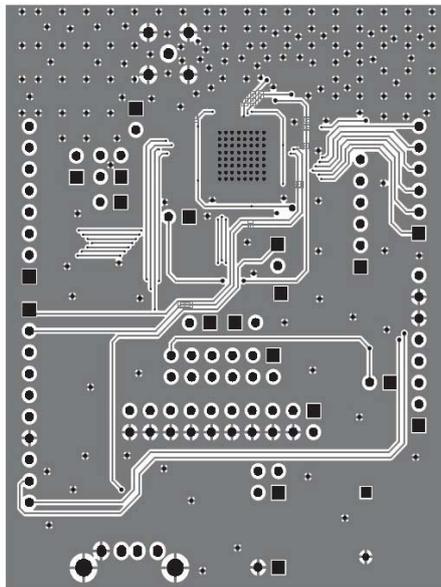




Figura 46. Diagrama elétrico parcial (2/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o *range extender* CC2892. Referente ao circuito da conexão USB (do autor).

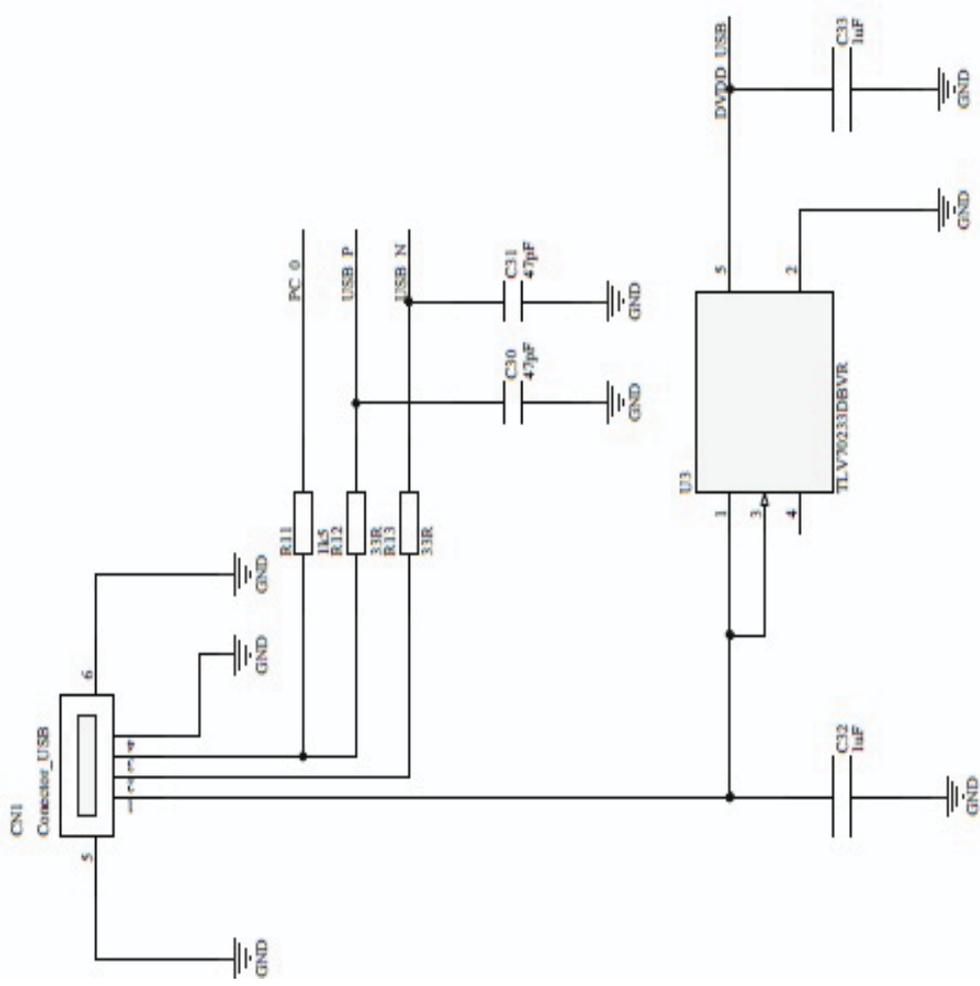


Figura 47. Diagrama elétrico parcial (3/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o *range extender* CC2892. Referente ao circuito de alimentação externa, seleção da alimentação e capacitores de desacoplamento (do autor).

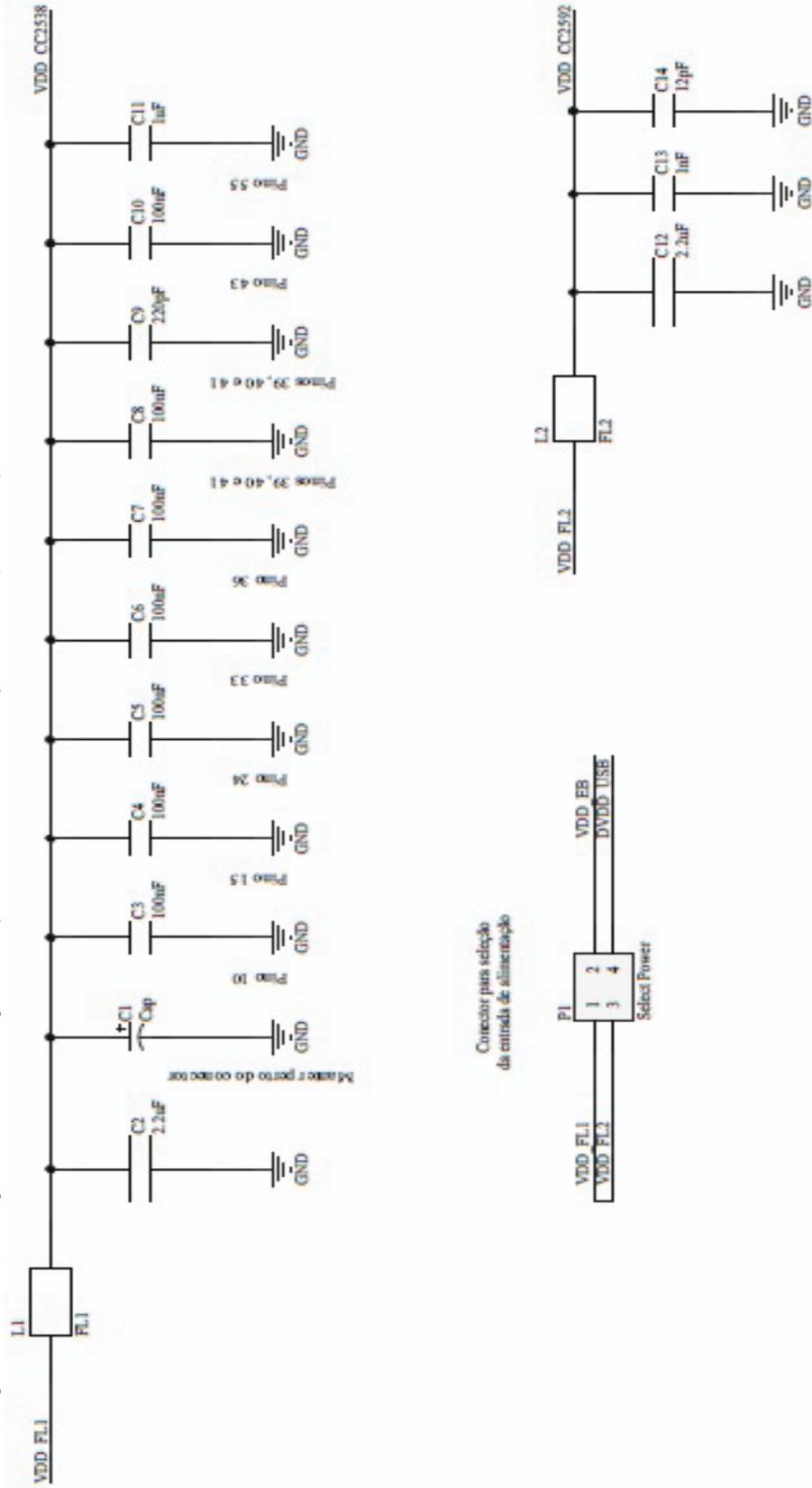


Figura 48. Diagrama elétrico parcial (4/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o range extender CC2892. Referente ao circuito dos conectores (do autor).

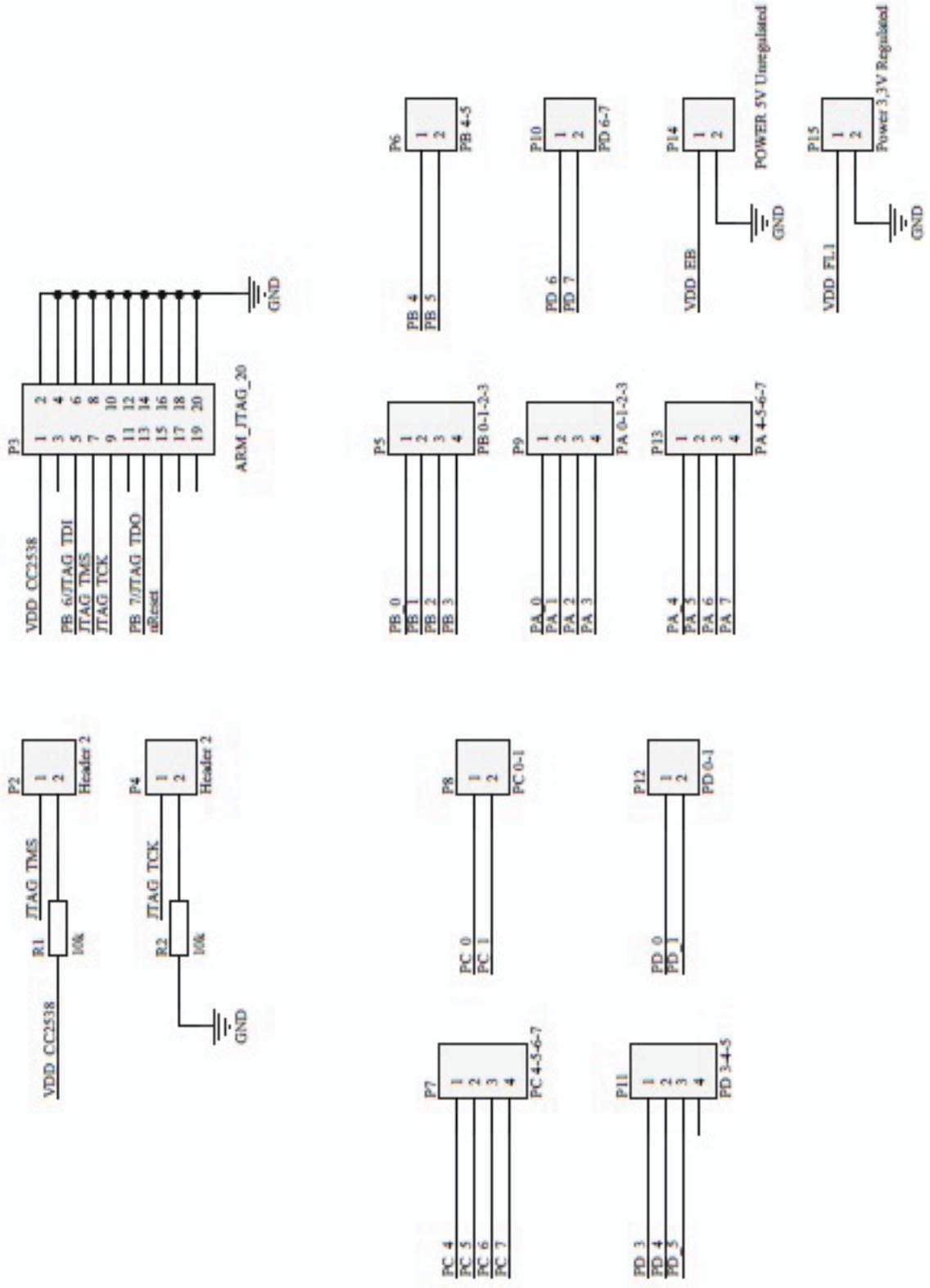


Figura 49. Diagrama elétrico parcial (5/5) do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o *range extender* CC2892. Referente ao circuito de RF (do autor).

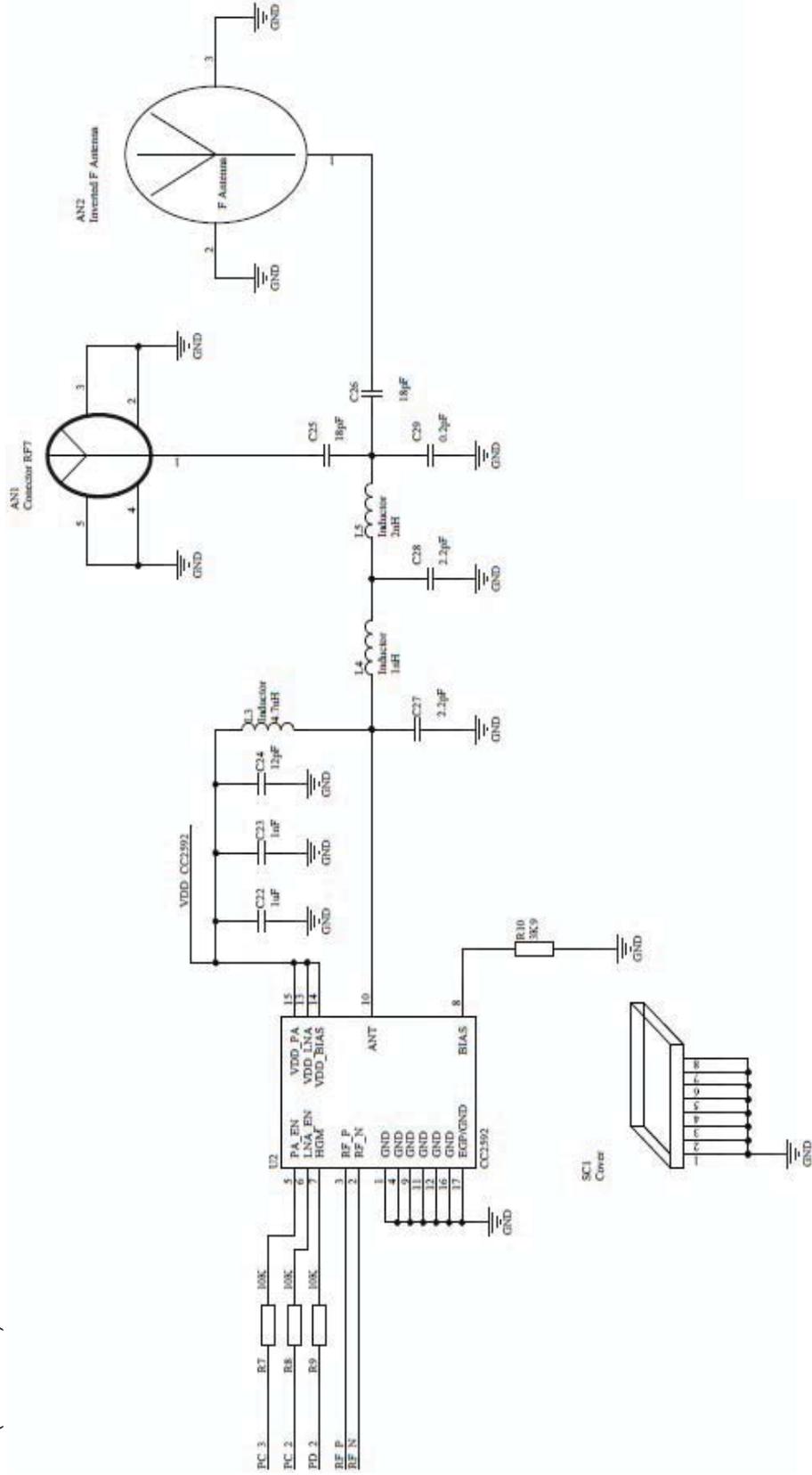


Figura 50. *Layer top* da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o *range extender* CC2892. (do autor).

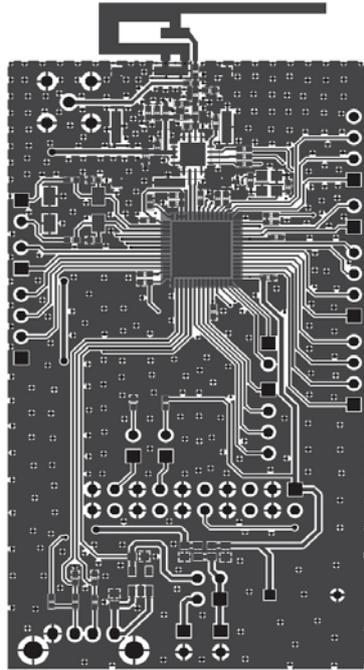
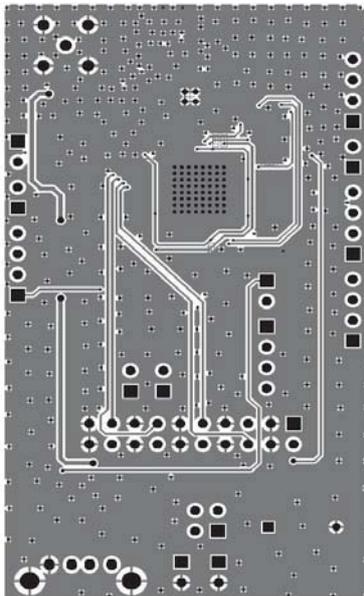


Figura 51. *Layer bottom* da placa de circuito impresso do módulo desenvolvido utilizando o SoC CC2538 em conjunto com o *range extender* CC2892. (do autor).



## APÊNDICE D – DETALHES DO PROJETO DO SNIFFER UTILIZANDO O SOC CC2531

Figura 52. Diagrama elétrico parcial (1/4) do sniffer desenvolvido utilizando o SoC CC2531. Referente ao circuito do microcontrolador (do autor).

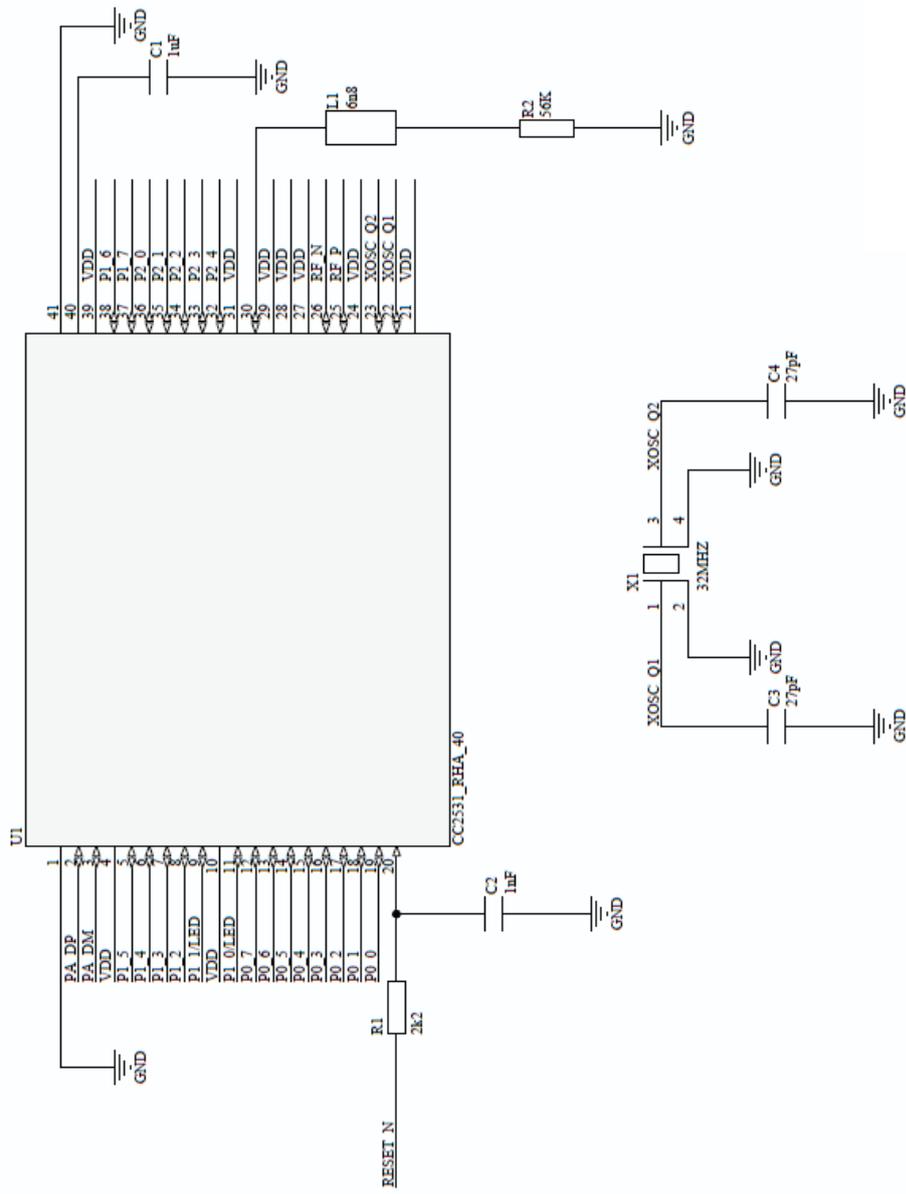


Figura 53. Diagrama elétrico parcial (2/4) do sniffer desenvolvido utilizando o SoC CC2531. Referente ao circuito do regulador de tensão e capacitores desacoplamento (do autor).

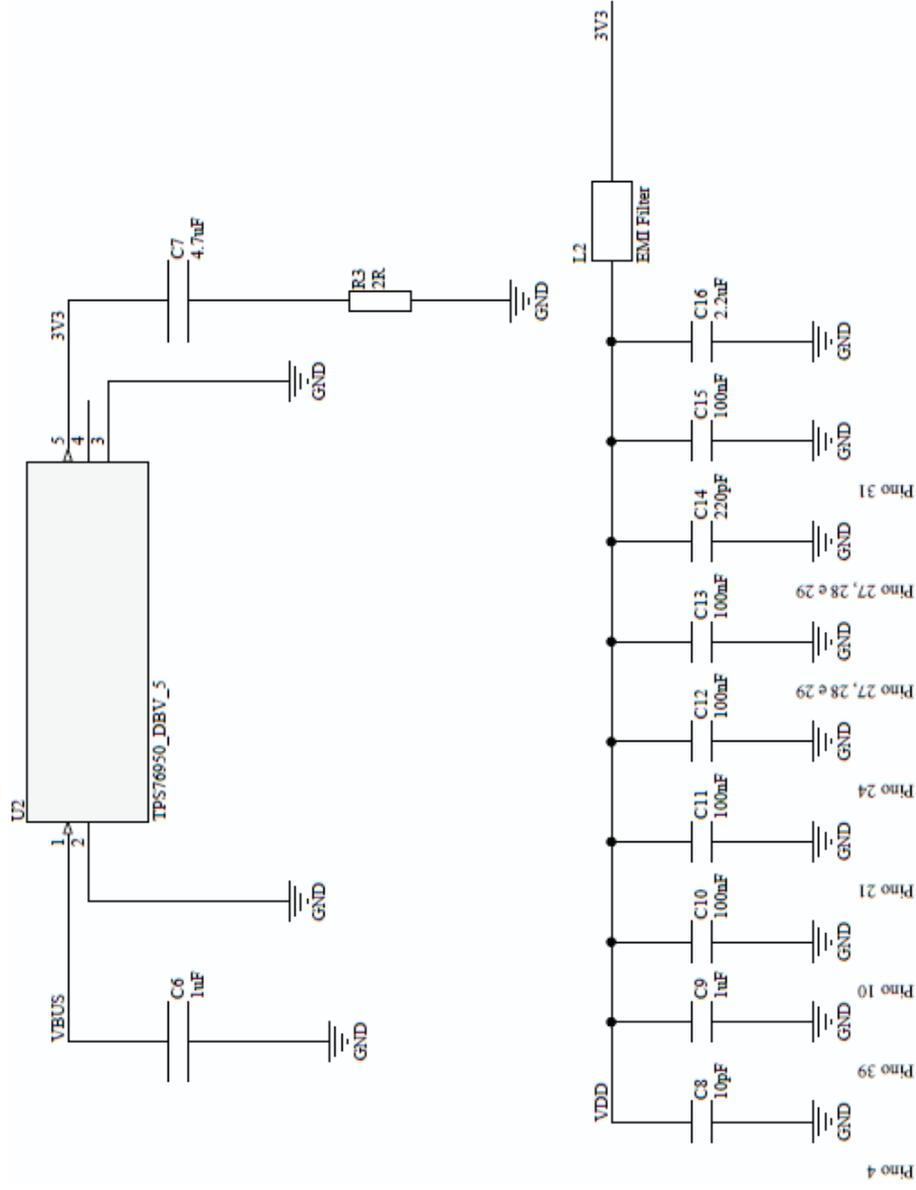


Figura 54. Diagrama elétrico parcial (3/4) do sniffer desenvolvido utilizando o SoC CC2531. Referente ao circuito dos conectores, leds e botões (do autor).

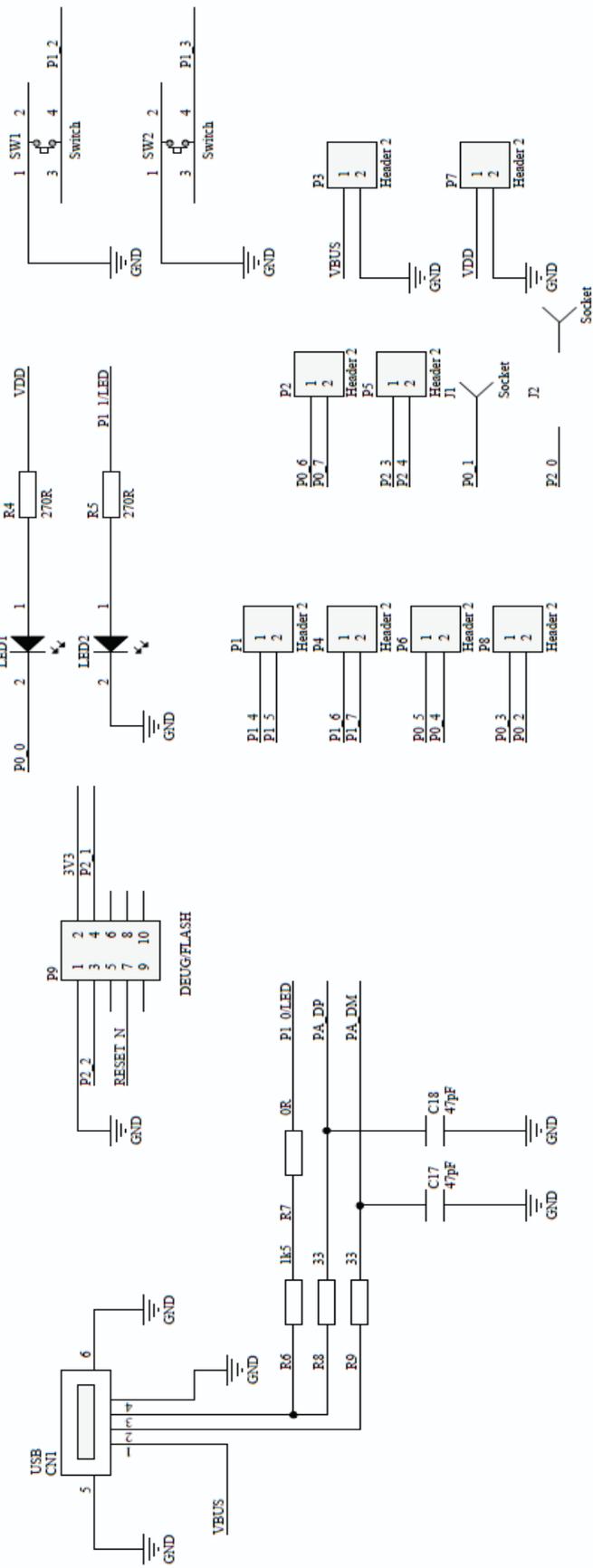


Figura 55. Diagrama elétrico parcial (4/4) do *sniffer* desenvolvido utilizando o SoC CC2531. Referente ao circuito de RF (do autor).

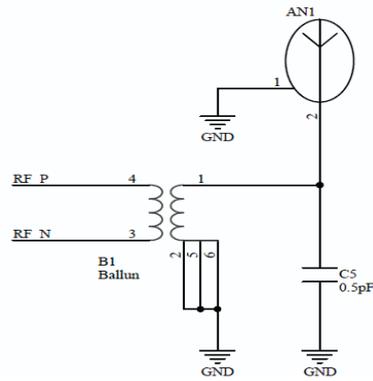


Figura 56. *Layer top* da placa de circuito impresso do *sniffer* desenvolvido utilizando o SoC CC2531. (do autor).

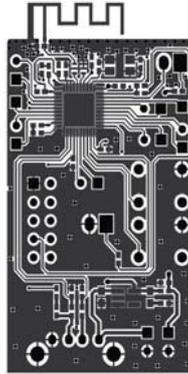


Figura 57. *Layer bottom* da placa de circuito impresso do *sniffer* desenvolvido utilizando o SoC CC2531. (do autor).

