



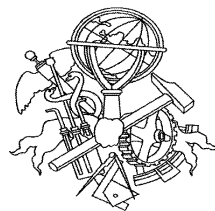
SISTEMA DE MEDIÇÃO DE TEMPERATURA CORPORAL BASEADO EM REDE DE SENSORES SEM FIOS ANT

JORGE FILIPE CASTRO DE SOUSA

Novembro de 2012

SISTEMA DE MEDIÇÃO DE TEMPERATURA CORPORAL BASEADO EM REDE DE SENSORES SEM FIOS ANT

Jorge Filipe Castro de Sousa



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2012

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores
- Automação e Sistemas

Candidato: Jorge Filipe Castro de Sousa, Nº 1070252, 1070252@isep.ipp.pt

Orientação científica: Lino Manuel Baptista Figueiredo, lbf@isep.ipp.pt

Coorientação Científica: António Meireles, atmeireles@googleemail.com



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

7 de Novembro de 2012

À minha namorada e família.

Agradecimentos

Foram várias pessoas, que em maior ou menor medida, contribuíram de alguma forma na elaboração deste projeto, e que eu apenas posso expressar os meus mais sinceros agradecimentos.

Em primeiro lugar gostaria de expressar um agradecimento muito especial à minha querida Lílana, por todo o apoio que sempre me deu, nos bons e maus momentos esteve sempre ao meu lado, por todo o tempo que lhe roubei na elaboração deste projeto, por ter demonstrado uma fé sem limites e por ter sempre acreditado em mim e no meu sucesso.

Quero expressar o meu mais sincero e profundo agradecimento aos meus pais, Maria Isabel e Manuel Sousa, e à minha irmã, Joana Sousa, por todo o apoio, preocupação e sacrifício que sempre demonstraram para comigo.

Expressos também os meus agradecimentos aos meus orientadores, Lino Figueiredo e António Meireles, pela oportunidade de participar neste projeto e pela dedicação e contributo para que este seja um êxito.

E por último, mas não menos importantes, a todos os meus amigos, que também estiveram sempre a meu lado com uma palavra amiga, sugestões e incentivos.

A todos eles, muito obrigado.

Resumo

O presente projeto enquadra-se na área das redes de sensores sem fios, onde o seu crescente desenvolvimento permite aplicar esta tecnologia em diversas áreas, como por exemplo, na monitorização ambiental, utilização militar, domótica, saúde, entre outras. Uma rede de sensores sem fios consiste em diversos nós dispersos num campo de aplicação onde procedem à recolha de dados do ambiente em que estão inseridos, como o valor de uma temperatura, humidade ou outra grandeza física, e os transmitem para uma estação base onde podem ser monitorizados.

Tendo a área da saúde uma importância significativa, este projeto focalizou-se na mesma. O projeto apresentado nesta dissertação teve como principal objetivo o desenvolvimento de uma rede de sensores sem fios, em que dois nós procedam à aquisição da temperatura corporal de uma pessoa em dois locais distintos, para posterior envio da mesma para os restantes nós da rede, onde será apresentada em estações de monitorização.

Este projeto foi desenvolvido baseado num recente protocolo de redes sem fios, nomeadamente o protocolo ANTTM. Assim sendo, em primeiro lugar, serão abordados neste relatório os objetivos e a contextualização deste projeto.

Em seguida, será apresentada uma comparação sobre alguns aspetos de algumas tecnologias de comunicações sem fios, nomeadamente o *ZigBee*, *Bluetooth* e ANT. Devido ao fato da tecnologia ANT ser a escolhida para o desenvolvimento deste projeto, será também apresentado um estudo mais detalhado sobre o mesmo.

Depois, será apresentado o desenvolvimento e implementações efetuadas, e por último serão apresentadas as conclusões técnicas e pessoais que este projeto permitiu obter.

Palavras-Chave

Redes de sensores sem fios, sistema de aquisição de temperatura corporal, novo protocolo ANTTM

Abstract

This project is related with wireless sensor network. This technology can be applied in many areas, like for example, in environmental monitoring, home automation, military and many others.

A wireless sensor network consists on several spread nodes. This nodes acquire data, like the value of a temperature or humidity, from the environment where they are inserted and then they transmit these data to others nodes of the network where they can be observed in monitoring stations.

This project is focused in healthcare area and its purpose is to develop a body temperature monitoring system based in a new wireless communication, namely ANTTM protocol.

In this network two nodes will acquire the body temperature in two different places and then send the data to others two nodes where the monitoring station are.

In this report it will be addressed a comparison between the ANTTM protocol, ZigBee and Bluetooth to better understanding the deferences between them.

After, a more detailed study about ANTTM will be presented to better introduce this new technology, and then it will be presented the development and implementations that was accomplished.

Finally the technical and personal conclusion will be presented.

.

Keywords

Wireless Sensor Network, System acquisition of body temperature, New ANTTM protocol.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. INTRODUÇÃO	1
1.2. CONTEXTUALIZAÇÃO	3
1.3. OBJETIVOS	4
1.4. ORGANIZAÇÃO DO RELATÓRIO	4
2. ESTADO DA ARTE	7
2.1. TECNOLOGIAS DE COMUNICAÇÕES SEM FIOS: <i>ZIGBEE</i> , <i>BLUETOOTH</i> E <i>ANT</i>	7
2.1.1. <i>Tecnologia ZigBee</i>	8
2.1.2. <i>Tecnologia Bluetooth</i>	9
2.1.3. <i>Tecnologia ANT</i>	10
2.2. CONSUMOS ENERGÉTICOS	11
2.2.1. <i>Consumos Energético do Protocolo ZigBee</i>	12
2.2.2. <i>Consumo Energético do Protocolo Bluetooth</i>	13
2.2.3. <i>Consumo Energético do Protocolo ANT</i>	14
2.3. TOPOLOGIAS DE REDE	17
2.3.1. <i>Topologias de Rede Suportadas pelo ZigBee</i>	18
2.3.2. <i>Topologias de Rede Suportadas pelo BlueTooth</i>	19
2.3.3. <i>Topologias de Rede Suportadas pelo ANT</i>	20
2.4. INTERFERÊNCIAS E COEXISTÊNCIA	21
2.4.1. <i>Mecanismos de Coexistência do Protocolo ZigBee</i>	21
2.4.2. <i>Mecanismos de Coexistência do Protocolo Bluetooth</i>	23
2.4.3. <i>Mecanismos de Coexistência do Protocolo ANT</i>	25
3. ESTUDO DA TECNOLOGIA ANT	27
3.1.1. <i>Redes Implementadas pelo Protocolo ANT</i>	28
3.1.2. <i>Comunicação Entre Nós de Rede</i>	29
3.1.4. <i>Emparelhamento de canais</i>	31

3.1.5.	<i>Tipos de Transmissão de Mensagens Transmitidas pela Rede</i>	34
3.2.	INTERAÇÃO ENTRE MICROCONTROLADOR E TRANSCEIVER ANT.....	38
3.2.1.	<i>Estrutura das mensagens suportadas pelo Transceiver ANT</i>	41
4.	DESENVOLVIMENTO DO PROJETO.....	43
4.1.	ARQUITETURA DA REDE E FUNCIONALIDADE DO SISTEMA	43
4.2.	<i>HARDWARE</i> UTILIZADO PARA O DESENVOLVIMENTO DO SISTEMA	44
4.4.	NÓ 1 E NÓ 2 - DEFINIDOS COMO MESTRES NA REDE	48
4.4.4.	<i>Aquisição da temperatura</i>	57
4.4.6.	<i>Nó 3 e Nó 4 - definidos como Escravos Na rede desenvolvida</i>	61
4.5.	COMUNICAÇÃO ENTRE OS NÓS CONSTITUINTES DA REDE	68
5.	CONCLUSÕES	71
	REFERÊNCIAS DOCUMENTAIS.....	74

Índice de Figuras

Figura 1	Pilha protocolar do protocolo ZigBee [19].....	8
Figura 2	Camadas do modelo OSI implementadas pelo protocolo ANT[11].....	11
Figura 3	Comparação entre os consumos energéticos de algumas tecnologias sem fios, realizada pelo ANT [26]	15
Figura 4	Consumo energético da tecnologia <i>Bluetooth Low Energy</i> , segundo ANT [26].....	16
Figura 5	- Comparação da duração de uma bateria CR2032 utilizando as diferentes tecnologias [32]	17
Figura 6	- Topologias de rede suportadas pelo ZigBee [17].....	19
Figura 7	- Exemplo de uma <i>scatternet</i> composta por duas <i>piconets</i> [37]	20
Figura 8	Topologias suportadas pelo ANT [48]	21
Figura 9	- Colisão entre um sinal “narrow band” e um sinal difuso [15]	22
Figura 10	- Divisão de canais da banda 2,4 GHz utilizado pelo ZigBee [15].....	23
Figura 11	- Mecanismo <i>frequency hopping</i> utilizado pelo Bluetooth.....	24
Figura 12	- Mecanismo AFH utilizado pelo Bluetooth[38]	25
Figura 13	-“Adaptative Isochronous Scheme” utilizado pelo ANT [16].....	26
Figura 14	- Comunicação entre nós numa rede ANT	29
Figura 15	- Estabelecimento de um canal	32
Figura 16	- Conhecimento Total do Identificador de Canal	33
Figura 17	- Conhecimento de dois campos do Identificador de Canal	34
Figura 18	- Conhecimento de um campo de Identificador de Canal.....	34
Figura 19	- Desconhecimento do campo de Identificador de Canal	34
Figura 20	- Exemplo de uma rede ANT.....	36
Figura 21	- Comunicação entre Microcontrolador e Transceiver ANT.....	39
Figura 22	- Comunicação Síncrona entre microcontrolador e o transceiver ANT	40
Figura 23	- Comunicação Assíncrona entre microcontrolador e módulo ANT	41
Figura 24	- Estrutura da mensagem ANT	41
Figura 25	- Estrutura do projeto.....	44
Figura 26	- Esquemático da placa de circuito impresso para o microcontrolador pic 18f4680....	45
Figura 27	- <i>kit de desenvolvimento</i> utilizado para o desenvolvimento do projeto	46
Figura 28	- <i>Transceiver</i> ANT utilizado para interligar formar os nós de rede	47
Figura 29	- <i>Plataforma de desenvolvimento</i> MPLAB™ para desenvolvimento da programação do microcontrolador	47
Figura 30	- Plataforma de desenvolvimento SharpDevelop utilizado para o desenvolvimento das interfaces gráficas.....	48
Figura 31	-Nós definidos como mestre na rede	49

Figura 32	- Fluxograma do funcionamento do microcontrolador.....	50
Figura 33	- Ligações físicas entre microcontrolador e transceiver ANT.....	51
Figura 34	- Mensagem de ativação de um canal num Transceiver ANT.....	54
Figura 35	- Mensagem de configuração do identificador de um canal num Transceiver ANT....	56
Figura 36	- Mensagem de ativação de um canal num Transceiver ANT.....	56
Figura 37	- Circuito aquisição de temperatura.....	57
Figura 38	- Fluxograma do ciclo para aquisição da temperatura.....	59
Figura 39	- Mensagem para envio de dados através do modo broadcast.....	60
Figura 40	- Fluxograma para o processamento de dados.....	61
Figura 41	- Placa de interface do <i>transceiver</i> ANT para <i>USB</i>	62
Figura 42	- Fluxograma da configuração do transceiver escravo	64
Figura 43	- Fluxograma da da função que processa os acontecimentos no canal.....	65
Figura 44	- Interface gráfica da estação de monitorização supervisora.....	66
Figura 45	- Ficheiro de registo da temperatura.....	67
Figura 46	- Interface gráfica da estação de monitorização local	68
Figura 47	- Comunicação entre os nós constituintes da rede.....	69

Índice de Tabelas

Tabela 1	Comparação entre Bluetooth Classico e Bluetooth Low Energy[13].....	14
Tabela 2	– Tipos de canais implementados pelo protocolo ANT	30
Tabela 3	Configurações do Canal entre o nó A e o nó B	36
Tabela 4	- Configuração do Canal entre o nó A e o nó C	37
Tabela 5	Configuração do Canal entre o nó A e o nó D	38
Tabela 6	– Estrutura de uma mensagem ANT	42
Tabela 7	– Tabela da relação dos pinos BR3, BR2 e BR1 com a seleção do Baud Rate.....	52
Tabela 8	– Configuração dos Nós Mestres.....	69
Tabela 9	– Configuração dos Nós Escravos.....	70

Acrónimos

ANT	–	Advanced and Adaptative Network Technology
AFH	–	Adaptative Frequency Hopping
CAN	–	Controller Area Network
CSMA	–	Carrier Sense Multiple Access
CAN	–	Controller Area Network
CSMA	–	Carrier Sense Multiple Access
CTS	–	Clear To Send
DLL	–	Dinamic-Link Library
DSSS	–	Direct Sequence Spread Spectrum
EUSART	–	Enhanced Universal Asynchronous Receiver Transmitter
FDMA	–	Frequency Division Multiple Access
IEEE	–	Institute Electrical and Electronics Engineers
ISM	–	Industrial, Scientific and Medical
MCU	–	MicroControllet Unit
NTC	–	Negative Thermistor Coefficient
OSI	–	Open Systems Interconnection
RISC	–	Reduced Instruction Set Computer
SPI	–	Serial Peripheral Interface

- ULP – Ultra Low-Power
- USB – Universal Serial Bus
- WPAN – Wireless Personal Area Network
- WPAN-LP – Wireless Personal Area Network – Low Power
- WSN – Wireless Sensor Network

1. INTRODUÇÃO

No presente relatório, descreve-se o trabalho desenvolvido associado à unidade curricular “Tese/Curricular”, passo final para a obtenção do grau Mestre em Engenharia Eletrotécnica e de Computadores, especialização em Automação e Sistemas.

1.1. INTRODUÇÃO

A crescente necessidade de mobilidade, flexibilidade e interação no dia-a-dia das pessoas aliciaram o desenvolvimento de novas tecnologias de comunicação e redes sem fios. O surgimento do telemóvel, que veio suprimir a necessidade de cabos para efetuar comunicações telefónicas entre pessoas, é talvez, o mais importante desenvolvimento efetuado nesta área, devido à importância que tem na vida das pessoas e nos dias de hoje, poucas são as que não possuem um equipamento destes. No entanto, as potencialidades deste género de tecnologia são evidentes e por esse motivo alastraram-se para outras áreas. Por exemplo, na indústria dos computadores, os fabricantes procuraram também incorporar nos seus produtos formas de estes comunicarem entre si sem a necessidade da utilização de cabos. Desde então foram surgindo vários equipamentos à escala mundial, como por exemplo computadores pessoais, sistemas de posicionamento global, e mais recentemente *smartphones*, que estão num conjunto de equipamentos que hoje utilizam tecnologias de comunicação sem fios. Os recentes desenvolvimentos nas tecnologias de comunicações

sem fios, aliada ao crescente ao avanço tecnológico ocorrido na área dos microprocessadores, na área dos sensores e na área de armazenamento de energia com a criação de novos e mais pequenos dispositivos, permitiram a migração para uma nova área de desenvolvimento, as redes de sensores sem fios.

Uma rede de sensores sem fios (*Wireless Sensor Network*) pode ser descrita como um conjunto de nós, leves, distribuídos espacialmente pelo local em que se pretende recolher informações e organizados de maneira a formarem uma rede cooperativa, utilizados para monitorizar condições físicas ou ambientais. Cada nó de rede consegue recolher informação sobre o ambiente que está inserido através dos seus sensores e envia-la através da rede para estações de monitorização. Geralmente, um nó é formado por um ou mais sensores, como por exemplo sensores térmicos ou biológicos, que procedem a aquisição das grandezas físicas em que estão inseridos, um microprocessador, memória, um circuito comunicação por radiofrequência e uma fonte de alimentação.

O microprocessador é normalmente responsável por gerir os protocolos de comunicação, processamento dos dados recebidos pelos sensores e efetuar a gestão da energia. O circuito de radiofrequência consiste num circuito que combina as funcionalidades de transmissor como recetor. No que diz respeito à memória, normalmente o microcontrolador já possui uma interna utilizada para armazenar o *software* e os dados necessários. Finalmente, a bateria. Esta é normalmente a maior contribuinte para as dimensões físicas e peso do nó e por essas razões são utilizadas as mais pequenas e leves possíveis. No entanto, sem energia, um nó não tem utilidade e não pode contribuir para a rede como um todo, ou seja, o tempo de vida de uma rede de sensores sem fios é determinada pela vida útil das baterias que constituem os seus nós. Assim sendo, uma das principais preocupações numa rede de sensores sem fios consiste no consumo de energia dos nós que a constituem. Sendo a comunicação de dados pela rede a funcionalidade que mais energia consome, estas são reduzidas ao máximo. Para tal, os nós de rede encontram-se num estado de suspensão e apenas saem desse estado periodicamente ou quando um evento no ambiente em que está inserido é despoletado.

Tipicamente, as redes de sensores sem fios não possuem ou têm pouca, infraestrutura, como por exemplo *routers*, e são constituídas por um número elevado de sensores implantados num local de interesse para monitorizar fenómenos físicos como uma temperatura, humidade, movimento, entre outros, e enviam os seus dados para estações de

centrais, normalmente interligadas a um computador. Funcionam portanto de uma forma *ad hoc*, ou seja, não utilizam um dispositivo central para encaminhamento dos seus dados, por outro lado, cada nó transmite os dados dos nós vizinhos de forma comunitária até a estação de destino.

As redes de sensores sem fios operam na frequência de 2,4 GHz, a banda *ISM*. Nessa banda de frequências operam várias tecnologias de redes sem fios desenvolvidas, nomeadamente o *ZigBee*, *Bluetooth* e *ANT*, esta última abordada no desenvolvimento deste projeto.

Finalizando esta introdução, as redes de sensores sem fios estão a ser cada vez mais utilizadas em diversas áreas de aplicação, como por exemplo para uso militar, monitorização ambiental, domótica, cuidados médicos, automação industrial, entre outras. Neste sentido, as redes de sensores sem fios prometem ser partes constituintes do dia-a-dia das pessoas de forma a facilitar os seus modos de vida.

1.2. CONTEXTUALIZAÇÃO

Uma das aplicações em que uma rede de sensores sem fios pode ser utilizada é na área da medicina onde pode proceder à monitorização dos sinais vitais de uma pessoa. O fato de cada nó da rede ser de pequenas dimensões, possuir uma fonte de energia única, aliado à sua capacidade de ser implementado causando pouco incómodo físico à pessoa, reforça a sua aplicabilidade na área dos cuidados de saúde. Por exemplo, uma unidade de cuidados intensivos de saúde, onde existem diversos pacientes com diferentes diagnósticos, uma rede de sensores sem fios poderia ser aplicada, onde cada nó adquiria a temperatura, ritmo cardíaco entre outros parâmetros relevantes de cada paciente e transmitia essa informação para uma estação central, onde um profissional de saúde poderia visualizar todos esses aspetos.

É neste sentido que o trabalho apresentado neste relatório se focaliza, nomeadamente na aquisição da temperatura de bebés para monitorização da mesma, implementado uma solução deste tipo. Esta solução foi desenvolvida com recurso a um protocolo recente desenvolvido para atuar em redes de sensores sem fios, denominado *ANT*. Esta tecnologia ainda pouco divulgada e utilizada promete ser uma solução viável para desenvolvimentos utilizando redes de sensores sem fios.

1.3. OBJETIVOS

O objetivo principal deste projeto consiste no desenvolvimento de um sistema que permita a aquisição e o envio da temperatura de uma pessoa através de uma rede de comunicação sem fios. Dada a complexidade inerente a este objetivo, sentiu-se a necessidade de o subdividir em múltiplas tarefas de realização mais simples, tais como:

- Estudo do protocolo ANT, que pode ser dividido nas seguintes partes:
 - Estudo teórico sobre o protocolo.
 - Estudo sobre como os módulos ANT comunicam com os controladores, desde ligações físicas a métodos de transmissão de dados.
 - Estudo tecnológico sobre o protocolo para conhecer as suas potencialidades, configurações e formulação de pacotes para serem corretamente enviados.
- Desenvolvimento completo de uma rede de sensores sem fios utilizando o protocolo ANT.
- Desenvolvimento de um circuito de aquisição de temperatura.
- Desenvolvimento de dois nós, com um microcontrolador e *transceiver* ANT interligados, desde as ligações físicas à programação desenvolvida para o efeito.
- Desenvolvimento de dois nós interligando um computador pessoal à rede ANT, através de um módulo USB.
- Realização de testes para analisar o comportamento e resultados do sistema.

1.4. ORGANIZAÇÃO DO RELATÓRIO

Este relatório encontra-se dividido em 5 capítulos. Nomeadamente, o capítulo 1 denominado “Introdução” encontra-se o enquadramento teórico do projeto realizado e os objetivos propostos para a realização deste projeto e pretende situar o leitor acerca da área em que este trabalho está inserido e as tecnologias que aborda.

O capítulo 2, denominado “Estado da Arte”, pretende apresentar três tecnologias diferentes, nomeadamente o *ZigBee*, *Bluetooth* e *ANT*, utilizadas nas redes de sensores sem

rios. Neste capítulo serão abordados alguns aspectos que diferenciam estas tecnologias, como consumos energéticos, topologias e mecanismos de coexistência.

O capítulo 3, denominado “Estudo da Tecnologia ANT” pretende apresentar o protocolo ANT e os conceitos que este utiliza para o seu funcionamento. Numa primeira fase é realizado um enquadramento mais teórico para melhor compreensão do protocolo ANT e de que forma este funciona, abordando assuntos como o mecanismo do tipo mestre-escravo e a utilização de canais que este protocolo utiliza para comunicar entre nós da rede. São também apresentados alguns exemplos de como uma rede pode ser configurada e implementada, assim como o tipo de mensagens que este protocolo utiliza. Depois é apresentada uma descrição de como este protocolo interage com os microcontroladores, nomeadamente os tipos de comunicação entre ambos que podem ser implementados.

O capítulo 4, denominado “Desenvolvimento do Projeto”, pretende demonstrar todos os desenvolvimentos e opções tomadas ao longo do projeto, assim como a validação dos resultados obtidos. Serão apresentados os desenvolvimentos referentes à constituição dos nós, tanto nos nós “mestres” como nos “escravos”, as configurações efetuadas no microcontrolador e no *transceiver* ANT, o circuito de aquisição da temperatura, entre outros aspectos que foram implementados. Serão também apresentados alguns testes que permitiram verificar o correto funcionamento de todo o sistema.

Por último, o capítulo 5 denominado “Conclusões” pretende apresentar as conclusões pessoais e técnicas retiradas do trabalho que foi realizado assim como outros aspectos que se podem incluir em futuras aplicações ou desenvolvimento.

2. ESTADO DA ARTE

As redes de sensores sem fios são uma área de desenvolvimento que deriva das tecnologias de comunicação sem fios. Nos dias de hoje, existem diversas tecnologias que procuram ser uma solução viável para aplicações nesta área. Todas as tecnologias desenvolvidas nas comunicações sem fios possuem características comuns independentemente de quem as desenvolve, no entanto, são as características intrínsecas que lhes permitem diferenciar-se das concorrentes e as tornam mais interessantes.

Neste sentido, foram abordadas as três tecnologias mais utilizadas nas redes de sensores sem fios, o *ZigBee*, *Bluetooth* e ANT. Todas elas podem operar em redes de sensores sem fios e oferecem diferentes características. Este capítulo pretende introduzir e apresentar as características e diferenças entre as tecnologias *ZigBee*, *Bluetooth* e ANT.

2.1. TECNOLOGIAS DE COMUNICAÇÕES SEM FIOS: ZIGBEE, BLUETOOTH E ANT

As especificações determinam o funcionamento e as características das tecnologias nelas abordadas. Esta secção pretende realizar uma apresentação sumária das especificações das tecnologias mencionadas.

2.1.1. TECNOLOGIA ZIGBEE

O desenvolvimento do protocolo *ZigBee* teve como base a especificação *IEEE 802.15.4*, que consiste na especificação para as redes de área pessoal sem fios (*WPAN*).

O seu desenvolvimento culmina com a interligação de dois grupos diferentes. Por um lado, o grupo designado no Instituto de Engenheiros Elétricos e Eletrónicos (*IEEE*) procurava criar uma especificação para um protocolo de redes sem fios com uma taxa de transmissão de dados baixa. Por outro lado, o grupo *ZigBee* (*ZigBee Alliance*), procurava criar um protocolo com um baixo consumo energético, com uma taxa de transmissão de dados baixa, com um custo reduzido e que garantisse a interoperabilidade de diversos dispositivos de diferentes fabricantes. Possuindo objetivos idênticos, estes dois grupos juntaram-se e deram origem ao protocolo *ZigBee*. [17]

O protocolo *ZigBee* teve como base do seu desenvolvimento a norma *IEEE 802.15.4*, que constitui um conjunto de especificações sobre a camada física (*PHY*) e a camada de controlo de acessos (*MAC*) para redes de área pessoal de baixa taxa de transmissão (*LR-WPAN*). O protocolo *ZigBee* desenvolveu as camadas superiores e funciona como intermediário entre as camadas mais baixas (*PHY* e *MAC*) da norma *IEEE 802.15.4* e do *software* que faz uso das comunicações sem fios. Define as especificações da camada de rede e fornece funcionalidades para o desenvolvimento de programas. Na Figura 1 encontra-se ilustrada a pilha protocolar implementada pelo *ZigBee*. [14][18]

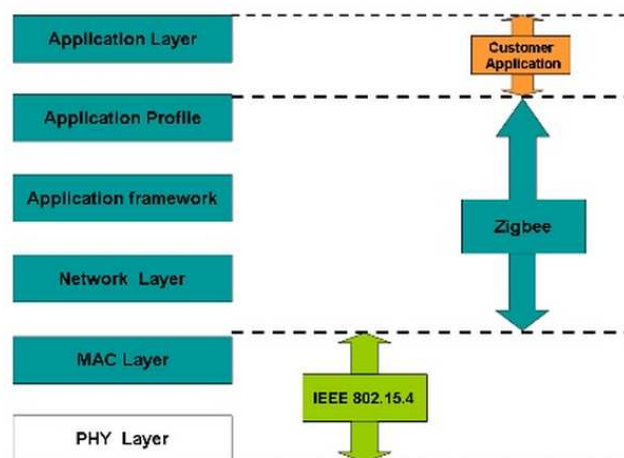


Figura 1 -Pilha protocolar do protocolo ZigBee [19]

Como pode ser visualizado na Figura 1, as duas camadas inferiores são implementadas pelo IEEE 802.15.4. A camada física (PHY) é responsável pela implementação física dos mecanismos de radiofrequência. A camada de controlo de acessos (MAC) é responsável pela interação entre as camadas superior e a camada física.

A camada de rede é responsável, entre outras coisas, pelo início e fim de ligações de novos dispositivos, ao armazenamento de informações relativa aos mesmos e à descoberta de novos dispositivos na vizinhança.

As camadas de aplicação permitem, entre outras funcionalidades, implementar mecanismos de segurança e especificar perfis, para que dispositivos de diferentes fabricantes possam comunicar entre si de forma transparente. [19]

O protocolo ZigBee suporta várias bandas de frequências de redes sem fios. Estas são as bandas de 2,4 GHz que possui 16 canais e pode ser utilizada em todo o mundo, 915 MHz que possui 10 canais e que pode ser utilizada na América do Norte e Austrália, e por último a banda 868 MHz com 1 canal que pode ser utilizada nos países da União Europeia.

O ZigBee é ainda capaz de atingir velocidades de 20 kbps na banda de 868 MHz, 40 kbps na banda de 915 MHz, 250 kbps na banda de 2,4 GHz e possui um alcance máximo entre dispositivos de aproximadamente 100 metros.[18][19][21]

2.1.2. TECNOLOGIA BLUETOOTH

Hoje em dia, dificilmente se encontra no mundo um telefone móvel que não possua a tecnologia *Bluetooth* para interligar o telefone com os seus acessórios. Computadores pessoais possuem também esta tecnologia para o mesmo propósito, assim como as novas gerações de automóveis que possuem dispositivos com esta tecnologia que permite ao condutor falar enquanto conduz. Esta tecnologia é, mais do que qualquer outra, a mais adotada em todo mundo, no que diz respeito a tecnologias de redes sem fios. A tecnologia *Bluetooth* foi inicialmente desenvolvida pela empresa Ericsson como uma alternativa à utilização de cabos RS232 para interligar os seus produtos [24]. Devido a esse fator, os consumos energéticos não foram tidos em conta. O *Bluetooth* é um protocolo orientado à conexão desenvolvido para permitir a transmissão contínua de dados a uma taxa relativamente elevada, exemplo disso é a sua aplicabilidade nas comunicações de voz. Ao longo do seu desenvolvimento, apesar de existir a preocupação relativa ao consumo

energético, o principal foco de desenvolvimento concentrou-se na capacidade de transferência de dados cada vez mais elevada. Exemplo disso foram as diversas versões que foram sendo lançadas ao longo dos anos. O Bluetooth 1.0 permite conexões assíncronas e síncronas entre dispositivos a velocidades de transferência de 720 kbps. A versão 2.0, lançada em 2004, pode atingir velocidade de transmissão de 2,1 Mbps. Em 2009, com o lançamento do Bluetooth 3.0, conseguiram aumentar essa velocidade a 24 Mbps. Essa focalização nas taxas de transferência permitiram atingir velocidades excelentes para um protocolo de redes sem fios, no entanto toda essa velocidade provoca um consumo energético elevado, levando a impraticabilidade desta tecnologia em dispositivos de baixo consumo. O elevado consumo energético levou a empresa Nokia a desenvolver um protocolo baseado no Bluetooth, nomeadamente o protocolo *Wibree*, que possui as características do protocolo Bluetooth mas ao contrario deste, os consumos energéticos são bastante reduzidos. Em 2007, o protocolo *Wibree* passou a ser parte integrante da especificação *Bluetooth*, originando desta forma o protocolo *Bluetooth Low Energy*, ou seja, este protocolo passou a fazer parte da especificação do *Bluetooth 4.0*. [13][22][29][37]

O *Bluetooth 4.0*, em contraste com as versões anteriores, foi desenvolvido para dispositivos sem fios que podem funcionar durante meses ou até mesmo anos com uma bateria do tipo CR2032. Os *transceivers* com este protocolo possibilitam comunicações assíncronas com outros dispositivos de baixo consumo energético, tais como sensores sem fios para enviar pequenas quantidades de dados. As conexões começam rapidamente e terminam assim que a transferência de dados é efetuada, permitindo assim uma elevada poupança energética. [13][22][29]

2.1.3. TECNOLOGIA ANT

O protocolo ANT não é baseado em nenhuma normalização e é um protocolo proprietário que opera na banda de frequência 2,4 GHz. Tipicamente, os *transceivers* ANT são tratados como “caixa negra”, devido ao fato de não existir o conhecimento de como ele trabalha por parte do público, não devendo apresentar muito esforço para implementar uma rede. Inicialmente foram desenvolvidos para permitir a comunicação entre sensores aplicados ao desporto e *displays*. Normalmente estes *transceivers* operam com baterias CR2032, e tal como o *Bluetooth Low Energy*, estas podem ter uma vida útil de anos.[11][25]

A sua pilha protocolar providencia uma comunicação de dados segura, operações de rede adaptativas e com imunidade de interferência. Esta é compacta e requer recursos mínimos do microcontrolador. [11]

O protocolo ANT providencia também flexibilidade em termos de definições implementadas pelo utilizador, como por exemplo, implementações ao nível de segurança. No entanto, o encargo computacional do microcontrolador é reduzido abstraindo as camadas inferiores do modelo *OSI* da aplicação do utilizador. As comunicações entre microcontrolador e os *transceivers* ANT foram desenvolvidas de forma a serem o mais simples possível para que deste modo seja realizado um desenvolvimento rápido. Para além do fato de ser fácil de implementar, o protocolo é também bastante leve, possui um *overhead* de negociação para início de transmissões pequeno e possui um consumo de energia muito reduzido. Tem a capacidade de enviar dados ate 1 mbps em operações com um baixo tempo de ciclo, tendo também um alcance máximo de 30 metros. Na Figura 2 encontra-se ilustrada as camadas implementadas pelo protocolo ANT. [11][26][16]

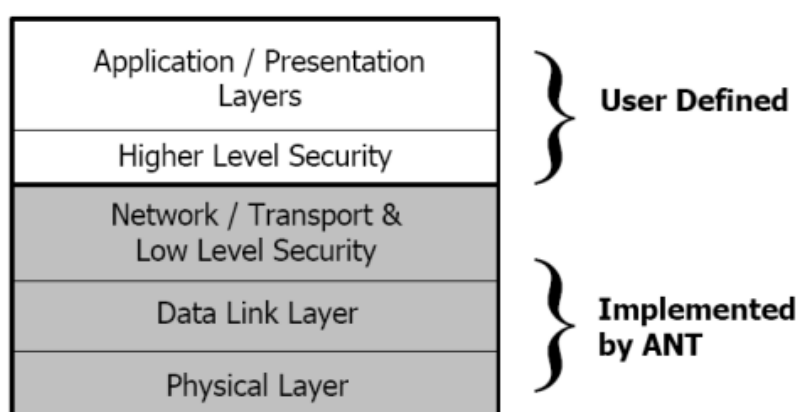


Figura 2 Camadas do modelo OSI implementadas pelo protocolo ANT[11]

2.2. CONSUMOS ENERGÉTICOS

Seja qual for o protocolo de redes sem fios, o consumo energético é uma das questões mais importantes que devem ser contempladas. Isto porque a tempo de vida útil das baterias dos diferentes dispositivos está diretamente relacionada com a energia que este consome na transmissão de dados, ou seja, quanto menor for o consumo de energia, maior será o tempo de vida útil das baterias. Este aspeto tem bastante importância pois está também

relacionado com a necessidade de carregar as baterias ou até mesmo na substituição dos dispositivos.

2.2.1. CONSUMOS ENERGÉTICO DO PROTOCOLO ZIGBEE

Uma das razões que esteve na origem do protocolo *ZigBee* consiste exatamente no desejo de criar um protocolo de comunicações sem fios que necessite de um baixo consumo energético para funcionar. Este protocolo implementa vários mecanismos de modo a minimizar os consumos energéticos.

O consumo energético do *ZigBee* está relacionado com a capacidade de cada dispositivo entrar num modo de suspensão sempre que não necessite enviar dados. Os dispositivos podem operar na maior parte do seu tempo neste modo obtendo assim, reduzidos tempos de ciclo (*duty cycle*). [28]

O consumo de energia mais significativo ocorre quando é efetuada a transmissão de dados, nomeadamente a energia relacionada com a potência da radiofrequência. No entanto, este aspeto adquire pouca importância pois a capacidade de o dispositivo entrar em modo de suspensão permite prolongar a vida útil das baterias. [14][28]

De modo a diminuir o tempo de transmissão, os dados são enviados a uma taxa elevada mas durante curtos períodos de tempo, o que permite poupar a carga das baterias. [14][28]

Devido a estes fatores, a *ZigBee Alliance* declara que o tempo de vida útil das baterias utilizando o seu protocolo pode variar entre alguns meses até alguns anos, e utiliza essa afirmação quando realiza o processo de certificação de um dispositivo (para obter a certificação da *ZigBee Alliance*, um dispositivo deve possuir um tempo de vida útil da sua bateria de pelo menos dois anos).[21][27]

No entanto, segundo um estudo realizado por *Chaitanya S. Misal* para sua dissertação de mestrado, o consumo energético de um dispositivo utilizando o protocolo *ZigBee* está relacionado com diversos fatores que dependem das configurações da rede. Entre os diversos fatores, está o fato do nó operar como transmissor ou recetor de dados (as baterias dos nós recetores têm 63% mais vida útil que um nós transmissor), se a rede permite a retransmissão de dados quando estes não chegam ao destino (as retransmissões de dados podem reduzir em 45% o tempo de vida útil da bateria) ou do tamanho dos pacotes transmitidos. [7]

2.2.2. CONSUMO ENERGÉTICO DO PROTOCOLO BLUETOOTH

Uma das características implementadas na versão Bluetooth 4.0 (ou *Bluetooth Low Energy*) que permitem uma redução dos consumos energéticos consiste em ativar as transmissões apenas por períodos reduzidos de tempo. Enquanto no Bluetooth Clássico (antes da implementação do protocolo *Wibree*), os dispositivos estavam constantemente a monitorizar 32 canais, requerendo 22,3 ms de cada vez que efetua essa monitorização, o Bluetooth 4.0 apenas necessita de monitorizar três canais, requerendo entre 0,6 a 1,2 ms para o fazer. Este fato por si só permite consumos energéticos entre 10 a 20 vezes mais reduzidos nos dispositivos com o *Bluetooth 4.0* em comparação com o *Bluetooth Clássico*. [8][13][22]

Os dispositivos que possuem o protocolo *Bluetooth 4.0* conseguem procurar novos dispositivos, estabelecer as conexões, enviar os dados, confirmar a receção dos mesmos e terminar a conexão em 3 ms, ao contrário do Bluetooth clássico que necessita de centenas de milissegundos para realizar as mesmas tarefas. Esta implementação permite desta forma uma elevada poupança de energia. [13][37]

Com uma largura de 1 Mbps, um transceiver Bluetooth 4.0 estará ligado $\frac{1}{8}$ do tempo que um transceiver *ZigBee* (operando na mesma banda 2,4 GHz) necessita para enviar os mesmos dados, resultando assim numa poupança energética oito vezes maior que o *ZigBee*, isto somente utilizando uma maior largura de banda. [13][22][29]

No entanto, o *Bluetooth Low Energy*, apesar de possuir um consumo energético bastante menor possui também uma taxa de transmissão do sinal bastante menor do que a versão *Bluetooth* anterior. Na tabela 1, encontra-se ilustrada uma comparação entre o *Bluetooth Clássico* e o *Bluetooth 4.0*. [13][22][37]

Na Tabela 1, pode-se visualizar, entre vários aspetos, a maior robustez do protocolo Bluetooth 4.0 face à versão anterior (*Bluetooth Clássico*), pode-se verificar uma diminuição acentuada da latência e do tempo necessário para enviar os dados pretendidos, uma maior segurança do protocolo *Bluetooth 4.0*, um consumo energético e uma corrente de pico muito menor que na versão clássica, características intrínsecas a dispositivos de baixo consumo energético. No entanto, pode-se também visualizar que o alcance máximo é menor na versão 4.0, assim como as taxas de transmissão de dados. Apesar de terem

aplicações idênticas, também possuem aplicações que as diferenciam. Talvez a principal deve-se ao facto da versão *Bluetooth Low Energy* não ser capaz de transmitir dados de voz, pois como o *Bluetooth 4.0* consiste na junção do *Bluetooth 3.0* com o *Bluetooth Low Energy*, e por esse motivo possui as características de ambos, no entanto analisando o *Bluetooth 4.0* como um todo este é capaz de transmitir dados de voz, mas analisando as diferenças apenas do *Bluetooth Low Energy* e da versão 3.0 este não é capaz de os transmitir dados de voz.

Tabela 1 Comparação entre Bluetooth Classico e Bluetooth Low Energy[13]

Technical Specification	Classic Bluetooth	Bluetooth low energy
Distance/Range	100 m (330 ft)	50 m (160 ft)
Over the air data rate	1-3 Mb/s	1 Mb/s
Application throughput	0.7-2.1 Mb/s	0.26 Mb/s
Active slaves	7	Not defined; implementation dependent
Security	64/128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check
Latency (from a non-connected state)	Typically 100 ms	6 ms
Total time to send data (det. battery life)	100 ms	6 ms
Voice capable	Yes	No
Network topology	Scatternet	Star-bus
Power consumption	1 as the reference	0.01 to 0.5 (depending on use case)
Peak current consumption	<30 mA	<20 mA (max 15 mA to run on coin cell battery)
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, gaming, headsets, stereo audio streaming, automotive, PCs, security, proximity, healthcare, sports & fitness, etc.	Mobile phones, gaming, PCs, watches, sports and fitness, healthcare, security & proximity, automotive, home electronics, automation, Industrial, etc.

2.2.3. CONSUMO ENERGÉTICO DO PROTOCOLO ANT

Em termos de consumo energético, o protocolo ANT evidencia-se dos concorrentes por ser uma solução sem fios “*Ultra Low-Power*”, sendo a tecnologia de redes sem fios mais

energeticamente eficiente do mercado, capaz de funcionar durante anos com uma bateria CR2032.[31][37]

A empresa responsável pelo seu desenvolvimento, a *Dynastream Innovations Inc.*, realizou uma análise de mercado sobre o consumo energético, alcance e taxa de transmissão de dados de diversas tecnologias.

Esta análise de mercado está ilustrada sob a forma de gráfico na Figura 3.[26]

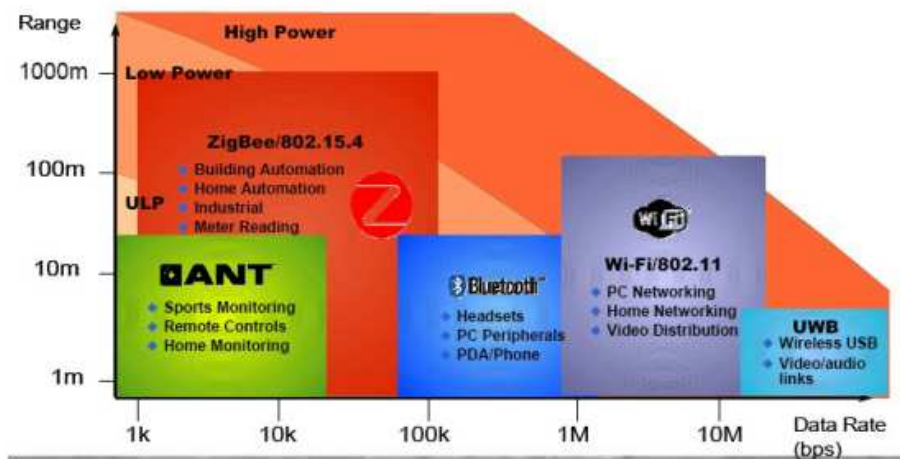


Figura 3 Comparação entre os consumos energéticos de algumas tecnologias sem fios, realizada pelo ANT [26]

Com o surgimento da nova especificação do *Bluetooth*, o *Bluetooth Low Power*, a mesma análise foi realizada mas nesta, foram consideradas as novas características deste protocolo. Nesta análise, o protocolo Bluetooth 4.0 sobrepõe-se ao protocolo ANT, podendo também apresentar baixos consumos energéticos, maior alcance e maiores taxas de transferência, fazendo dele o principal concorrente do protocolo ANT, devido às suas características mas também pela diversidade das suas aplicações. Na figura 4 encontra-se ilustrada o consumo do protocolo Bluetooth 4.0, segundo as expectativas do ANT. [26]

Na Figura 4 é possível visualizar a comparação efetuada pelo ANT.

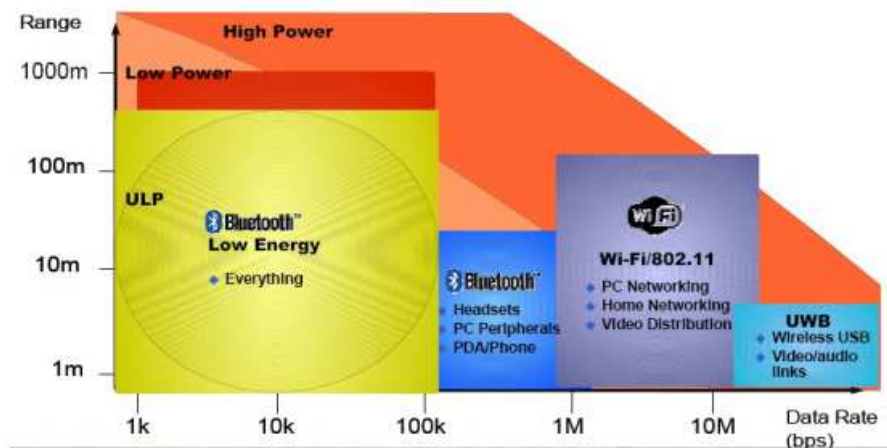


Figura 4 Consumo energético da tecnologia *Bluetooth Low Energy*, segundo ANT [26]

A *Dynastream Innovation Inc*, defende que esta característica, o consumo “*Ultra Low-Power*”, que diferencia este protocolo, pode ser alcançada através da eficiência, simplicidade e escalabilidade. [26]

Eles referem a eficiência pelo fato das comunicações serem efetuadas durante reduzidos espaços de tempo. No entanto, esta é uma característica presente também nas outras tecnologias e que por si só, não diferencia o protocolo ANT dos outros. Assim, a empresa procurou desenvolver um protocolo bastante leve e com um *overhead* tão reduzido quanto possível. De acordo com a empresa, o protocolo *ZigBee*, *Bluetooth Low Energy* e *Bluetooth*, possuem pelo menos o dobro de *overhead* do que o protocolo ANT, e este por sua vez, possui uma pilha protocolar muito menor que os concorrentes. [26][31][37]

No que diz respeito à simplicidade, esta consiste no desenvolvimento de um protocolo de troca de mensagens claro, que requer o mínimo de negociação possível ou *handshaking*.

A escalabilidade permite a inserção ou remoção de nós na rede sem a necessidade de dispositivos coordenadores, funcionando portanto de uma forma *ad hoc*.

O ex-diretor da empresa *This is ANT*, Brian Macdonald, apresenta, num artigo escrito para a *Nikkei Electronics Asia* em Dezembro de 2007, um exemplo que consiste num sensor a transmitir oito bytes de dados, 24 horas por dias, com um período de mensagem de 0,5 Hz. Para este exemplo, uma tecnologia que suporte um consumo de energia extremamente baixo (*Ultra Low-Power*), como é o caso da tecnologia ANT, demonstra um tempo de vida útil da bateria de aproximadamente 17 meses, o que de acordo com Brian, esta é uma

solução bastante melhor que a tecnologia ZigBee e Bluetooth que demonstraram um tempo de vida útil da bateria entre as 8 e 10 semanas.

Na Figura 5 encontra-se ilustrado o tempo de vida útil da bateria, segundo o estudo realizado

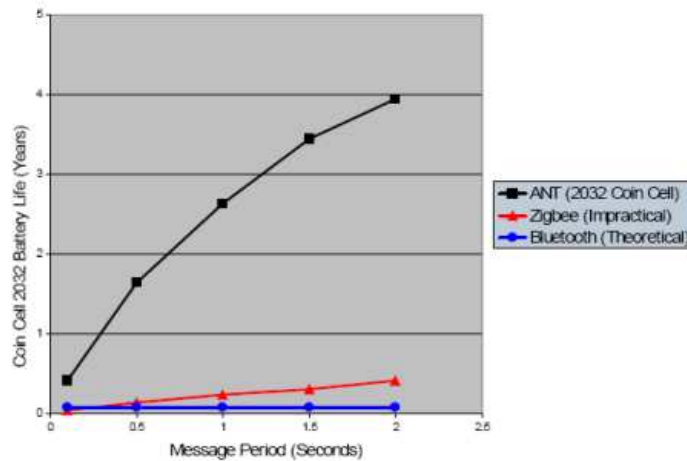


Figura 5 - Comparação da duração de uma bateria CR2032 utilizando as diferentes tecnologias [32]

No entanto, esta comparação não é realista, na medida em que a corrente de pico do ZigBee e do Bluetooth ultrapassa a capacidade da bateria CR2032 e por esse motivo o funcionamento dessas tecnologias com uma bateria deste tipo não é possível. Assim, os dados apresentados na Figura 5 são meramente teóricos. [25][32]

2.3. TOPOLOGIAS DE REDE

Existem diversas topologias de rede que satisfazem diversas necessidades. Este é um aspecto importante de qualquer protocolo de redes sem fios, pois quantas mais topologias um protocolo consegue implementar, mais flexível este é. Uma das mais importantes e complexas topologias é a do tipo malha (*mesh*). O conceito deste tipo de rede é importante consiste na capacidade de um dispositivo em comunicar com qualquer outro dentro do ambiente em que estão inseridos. Esta topologia tem também a capacidade de encaminhar mensagens utilizando diferentes caminhos para chegar ao mesmo destino, portanto é redundante.

Nesta secção são abordadas as topologias de rede suportadas pelo *ZigBee*, *Bluetooth* e *ANT*.

2.3.1. TOPOLOGIAS DE REDE SUPORTADAS PELO ZIGBEE

O protocolo *ZigBee* suporta diversos tipos de topologias de rede, a topologia em malha incluída, com um número máximo de 2^{64} nós por rede. [18]

Numa rede utilizando o protocolo *ZigBee*, existem dois tipos de dispositivos [17][18]:

- *Full Function Devices (FFD)* – Este tipo de dispositivo apresentam todas as funcionalidades e realiza as tarefas mais complexas da rede. Estes são capazes de efetuar comunicações e retransmitir o tráfego de dados para outros dispositivos.
- *Reduced Function Devices (RFD)* – Este é o tipo mais simples de dispositivo e que apenas é implementado como “*end-point*” na configuração da rede. Estes possuem funcionalidade apenas para retransmitir o tráfego para um único *FFD* e não são capazes de retransmitir os tráfegos proveniente de outros dispositivos.

Estes dois tipos de dispositivos podem no entanto assumir três funções diferentes numa rede:

- *ZigBee PAN Coordinator* – O dispositivo que toma este papel assume as funcionalidades principais e a gestão de toda a rede. Eles são os dispositivos mais funcionais e são utilizados para iniciar a rede e providenciam a interligação com outras redes. Esta função é sempre realizada por um dispositivo *FFD*.
- *ZigBee Router* – A função deste dispositivo é encaminhar o tráfego entre os diferentes nós da rede. Esta função é sempre realizada por um dispositivo *FFD*.
- *ZigBee End-Device* - Estes dispositivos são os nós terminais (*end-node*) numa rede. Normalmente, este tipo de nó possuem um sensor associado que procede à recolha de informação do ambiente em que está inserido ou um dispositivo que realiza funções e recebe comandos. Tipicamente, estes nós são constituídos utilizando um dispositivo do tipo *RFD*, podendo no entanto serem implementados por um dispositivo *FFD*. Na figura 6 encontram-se ilustradas algumas das topologias suportadas pelo *ZigBee*.

Na Figura 6 é possível visualizar algumas topologias suportadas pelo *ZigBee*.

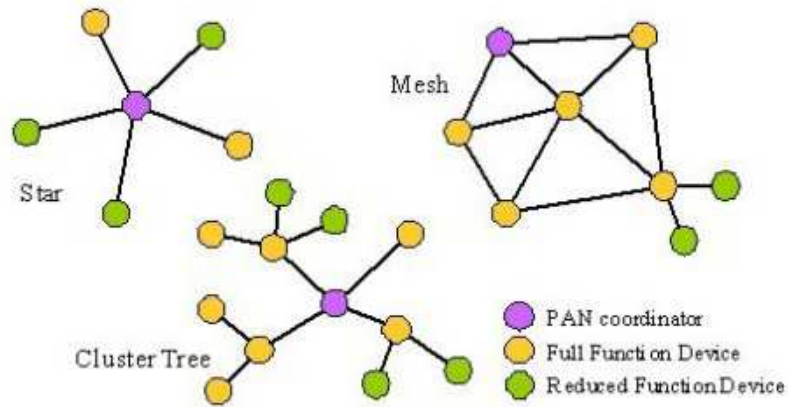


Figura 6 - Topologias de rede suportadas pelo ZigBee [17]

2.3.2. TOPOLOGIAS DE REDE SUPORTADAS PELO *BLUETOOTH*

Quando dois dispositivos *Bluetooth* estabelecem uma conexão, essa conexão dá origem a uma rede denominada *piconet*. Cada *piconet* possui um nó mestre e pelo menos um escravo, podendo possuir no máximo oito dispositivos ativos. Os nós escravos não comunicam diretamente entre si, assim estes dependem do nó mestre como um nó transitório entre os nós escravos. No entanto, estes papéis de dispositivo mestre e dispositivo escravo, tem apenas significado numa dada *piconet*, isto porque um dispositivo pode ter ambos papéis se pertencer a diferentes *piconets*. Um dispositivo pode assumir o papel de mestre apenas numa *piconet* mas pode por sua vez ter o papel de escravo noutras *piconets*. Quando tal acontece, a rede formada é uma *scatternet*. Uma *scatternet* não é nada mais do que um conjunto de duas ou mais *piconets* interligadas. Na Figura 7 encontra-se ilustrada uma *scatternet*. [36][37]

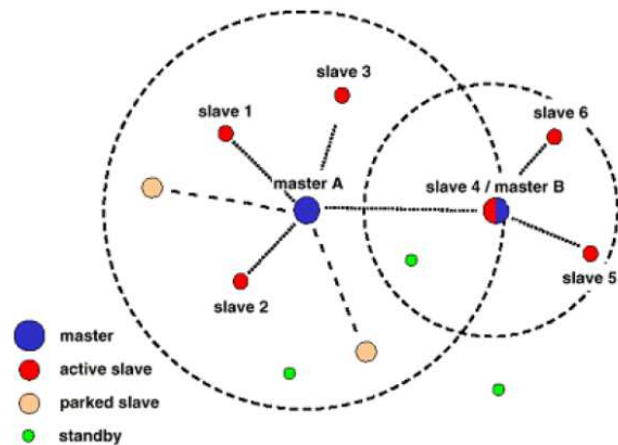


Figura 7 - Exemplo de uma *scatternet* composta por duas *piconets* [37]

2.3.3. TOPOLOGIAS DE REDE SUPORTADAS PELO ANT

O protocolo ANT suporta todas as topologias de rede que o protocolo ZigBee, no entanto utiliza um mecanismo diferente e pode possuir no máximo 2^{38} nós [25]. Numa rede ANT, os dispositivos não possuem diferentes funções, como acontece com os dispositivos (*Coordinator, Router e End-Point*) ZigBee. No caso do ANT, todos os nós são idênticos e possuem a capacidade de funcionarem como mestres e escravos, trocando de papéis sempre que necessário. Todos os nós podem atuar como transmissores, recetores ou transceiver para encaminhar o tráfego de dados, podendo entrar ou sair da rede, ou seja, a rede ANT possui as características de uma rede ad hoc.[16][25][33]

Quando se pretende implementar uma rede ZigBee, uma das limitações é a dificuldade de aumentar a rede e acrescentar nós de uma forma *ad hoc*. Isto deve-se ao fato de ser difícil para um nó entrar ou sair da rede a não ser que sejam do tipo certo, como por exemplo, um dispositivo “*End-Point*” só pode ser conectado a um FFD. [16][25][33]

Outra característica interessante do ANT deve-se ao fato de cada nó ser capaz de determinar qual é a melhor altura para transmitir os seus dados baseando-se na atividade dos nós vizinhos, e assim, não existe a necessidade de nós coordenadores. O ANT afirma que esta tecnologia é capaz de suportar este tipo de rede com centenas de nós, mantendo um *overhead* de sistema reduzido. Na Figura 8 encontram-se ilustradas as topologias que o protocolo ANT suporta.[16][33]

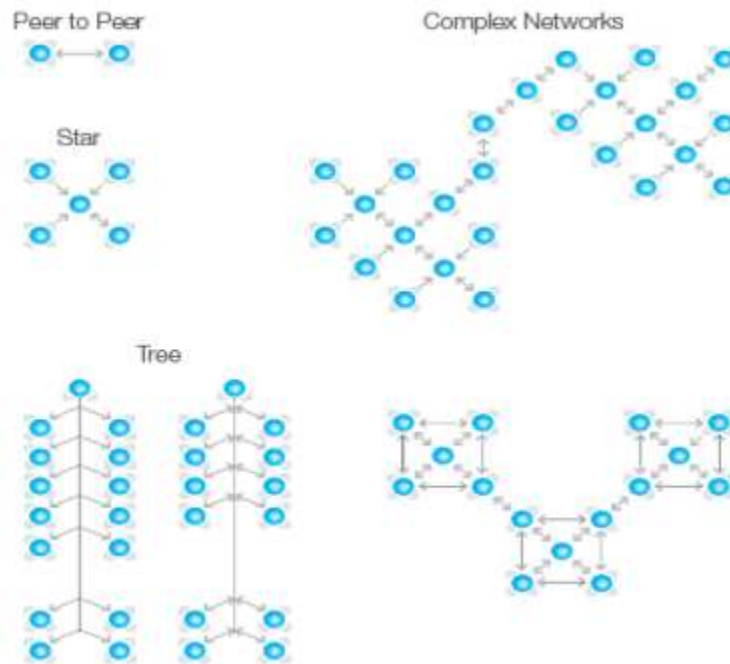


Figura 8 Topologias suportadas pelo ANT [48]

Como no caso do *ZigBee*, o ANT suporta topologias de rede em estrela, árvore e malha. No entanto, a utilização desta última insere complexidade e origina um maior consumo energético. Na maior parte dos casos, as implementações podem ser realizadas utilizando topologias de ponto-a-ponto, estrela ou árvore.[16][11]

2.4. INTERFERÊNCIAS E COEXISTÊNCIA

Estas tecnologias operam na frequência 2,4 GHz, denominada banda ISM. Sendo esta uma banda de frequências livre em todo o mundo, qualquer que seja a tecnologia de comunicações sem fios desenvolvida, esta deve garantir o seu correto funcionamento, seja em que ambiente esta esteja inserida. Para tal, os protocolos *ZigBee*, *Bluetooth* e ANT apresentam diferentes mecanismos que os permitem coexistir com outras tecnologias.

2.4.1. MECANISMOS DE COEXISTÊNCIA DO PROTOCOLO ZIGBEE

O protocolo *ZigBee* utiliza vários mecanismos para evitar a interferência com outras tecnologias. Estes mecanismos são os seguintes: *Direct Sequence Spread Spectrum (DSSS)*, múltiplos canais, elevada taxa de transferência de dados, *Carrier Sense Multiple Access (CSMA)* e retransmissão de dados. [15][34] [37]

O mecanismo DSSS consiste na difusão do sinal sem fios por uma largura de banda mais ampla do que o sinal de informação realmente necessita. O mecanismo DSSS, em contraste com sinais “*narrow-band*”, que quando uma colisão ocorre esta provoca uma sobreposição elevada dos sinais o que origina muita perda de informação. Os sinais que utilizam uma largura de banda ampla podem coexistir com sinais “*narrowband*” pois aquando uma colisão, o recetor apenas a interpreta como uma ligeira redução na relação entre sinal e ruído. [15][34] [37]

A técnica de difusão do sinal utiliza um código sequencial pseudoaleatório denominado “*chipping sequence*” que é utilizado para modular a portadora e codificar os dados a serem transmitidos.

Este código deve estar presente em todos os dispositivos da rede, pois apenas os dispositivos que possuem este código são capazes de reconstruir o sinal para o seu formato original. Na Figura 9, encontra-se ilustrada a colisão entre um sinal “*narrow-band*” e um sinal difuso, utilizado pelo *ZigBee*. [15][34]

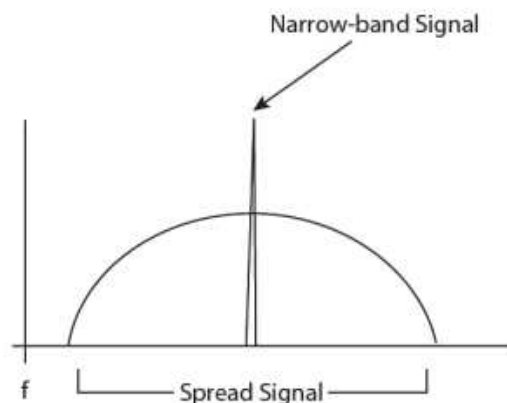


Figura 9 - Colisão entre um sinal “*narrow band*” e um sinal difuso [15]

O protocolo ZigBee também evita problemas relacionados com interferências recorrendo à utilização de múltiplos canais. Este mecanismo utiliza uma técnica denominada Múltiplo Acesso por Divisão de Frequência (FDMA) que consiste na divisão da banda ISM em 16 canais espaçados 5 MHz entre si. Quando num canal é detetada uma interferência o ZigBee este passa a funcionar noutra frequência. Na Figura 10 encontra-se ilustrada a divisão dos canais da banda ISM, utilizado pelo *ZigBee*. [15][34]

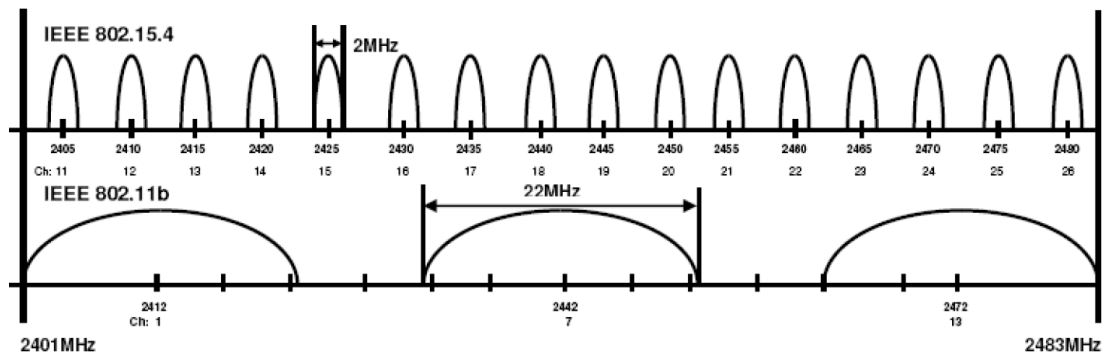


Figura 10 - Divisão de canais da banda 2,4 GHz utilizado pelo ZigBee [15]

No entanto, apesar das tentativas do protocolo *ZigBee* no sentido de minimizar as interferências utilizando canais múltiplos, estas ainda podem acontecer. Se um dispositivo transmitir a qualquer altura, esse fato vai originar colisões e perdas de dados. Para evitar esta situação é utilizada a técnica *CSMA*, que consiste em “escutar o meio” antes de proceder à transmissão de dados, ou seja, antes de transmitir os dados o dispositivo analisa o canal e no caso de não existir outro dispositivo a transmitir este inicia a sua transmissão, caso contrario não transmite. [15][34][37]

Outro mecanismo para evitar transferências consiste na utilização de uma relativamente elevada taxa de transmissão de dados, nomeadamente 250 kbps. O fato de a transmissão de dados ser elevada, permite uma menor ocupação dos canais e desta forma melhora a coexistência do protocolo *ZigBee* com outras tecnologias. [15][34]

Finalmente, o protocolo *ZigBee* recorre também ao mecanismo de retransmissão de dados quando uma mensagem enviada não é recebida pelo recetor. Quando um dispositivo recebe uma mensagem, este tem um curto período de tempo para enviar uma confirmação para o transmissor. Por sua vez, o transmissor se não receber essa mensagem no período de tempo definido, procederá à retransmissão da mensagem. Atingindo as cinco retransmissões, ele abandona essa mensagem e reporta um aviso. [15][34]

2.4.2. MECANISMOS DE COEXISTÊNCIA DO PROTOCOLO *BLUETOOTH*

O mecanismo que o Bluetooth utiliza para evitar interferências com outras tecnologias consiste numa técnica denominada por “salto de frequência” (*frequency hopping*). Esta técnica consiste na utilização de 79 canais dos 83,5 disponíveis na banda de 2,4 GHz, sendo que os saltos entre canais são efetuados aleatoriamente e com uma frequência de 1600

vezes por segundo. No entanto, quando o Bluetooth encontra-se num ambiente em que existe outra tecnologia de comunicação sem fios, algumas colisões ocasionais ocorrem. Por exemplo, num ambiente partilhado entre o *Bluetooth* e a tecnologia *WiFi*, algumas colisões irão ocorrer, utilizando este mecanismo. Na Figura 11 encontra-se ilustrado o mecanismo de *frequency hopping* utilizado pelo Bluetooth.

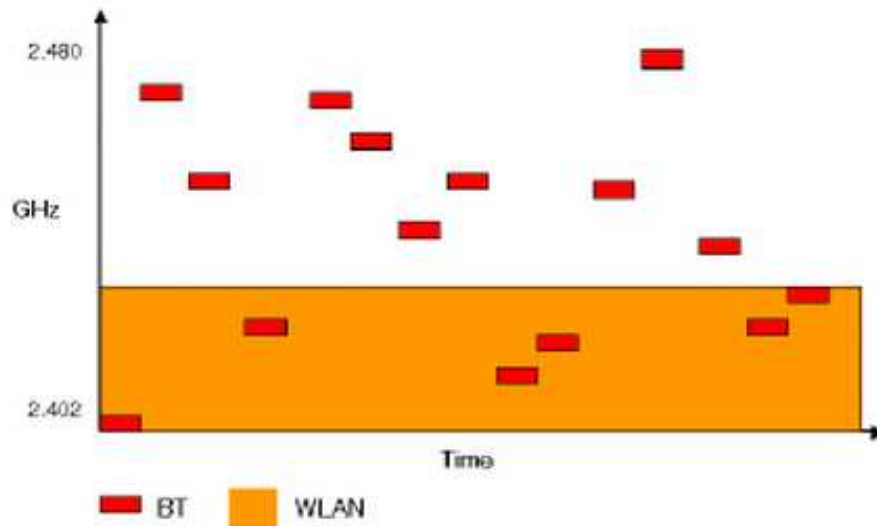


Figura 11 - Mecanismo *frequency hopping* utilizado pelo Bluetooth

De modo a evitar estas interferências e aumentar a segurança, o protocolo *Bluetooth4.0* implementa o mecanismo *Adaptive Frequency Hopping (AFH)*. O mecanismo *AFH* consiste na divisão dos dados em pacotes e na sua transmissão por vários canais utilizando “saltos de frequência”, com 1 MHz de diferença entre eles, num padrão aleatório. Até aqui, o mecanismo é exatamente igual. O que diferencia o mecanismo *AFH* do mecanismo *frequency hopping*, é que o *AFH*, apesar de também utilizar “saltos de frequência”, permite uma adaptação ao ambiente em que está inserido. Essa adaptação consiste na identificação da fonte de interferências fixas, excluindo-as da lista dos canais disponíveis. Este processo de identificação também envolve uma redução dos canais utilizados pelo Bluetooth, onde é possível utilizar um mínimo de 20 canais.[38]

Na Figura 12, encontra-se ilustrado o mecanismo *AFH*. Nesta figura pode-se visualizar que as colisões entre a tecnologia Bluetooth e a tecnologia *Wifi* foram evitadas. Utilizando este mecanismo o *Bluetooth* conseguiu identificar as fontes de interferência fixas utilizadas pelo *Wifi* e adaptou a utilização dos seus canais de forma a estes não coincidirem com os canais utilizados pela tecnologia *Wifi*.

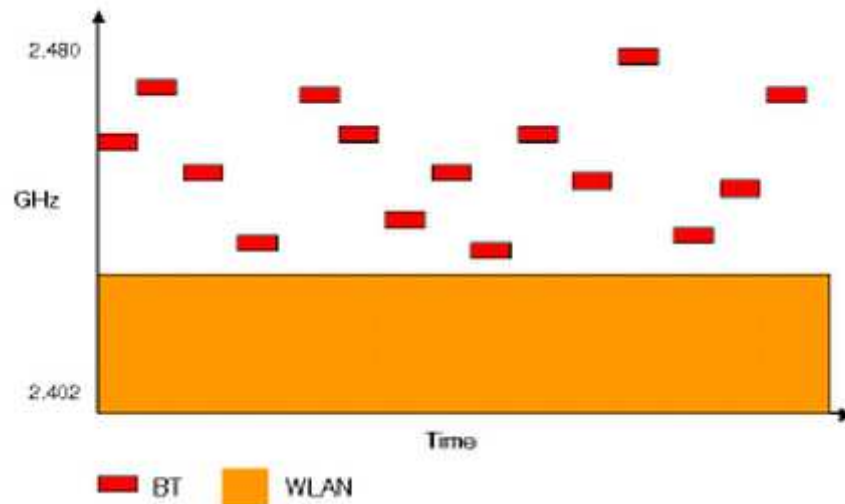


Figura 12 – Mecanismo AFH utilizado pelo Bluetooth[38]

2.4.3. MECANISMOS DE COEXISTÊNCIA DO PROTOCOLO ANT

O mecanismo que o ANT utiliza para evitar interferências e garantir a coexistência com outros protocolos denomina-se *Adaptive Isochronous Scheme* (AIS) que elimina eficazmente a maior parte dos problemas de interferência. [16][11]

O ANT divide a banda 2,4 GHz em vários canais de 1 MHz que por sua vez são divididos em vários *timeslots*. O mecanismo AIS consiste na transmissão de cada nó ser efetuado num *timeslot* livre, dentro do canal de 1 MHz. Cada canal por sua vez, apenas transmite por pequenos períodos de tempo (menos de 150 μ S) permitindo desta forma que um único canal de frequência consiga acomodar vários *timeslots*. Apesar do sistema utilizar este mecanismo, não é necessária a sincronização através de um tempo de relógio mestre. Cada nó transmite em espaços de tempo regulares, no entanto, este é capaz de alterar o seu espaço de transmissão se interferências de outros canais forem detetadas. Para além do fato de existirem *timeslots* específicos para cada nó, o protocolo ANT ainda possui outro mecanismo de coexistência. Se numa frequência de operação estiver a ser detetada muita interferência devido à existência de outra tecnologia que opera nesta gama de frequências (como o *Bluetooth* e *ZigBee*), o ANT possui a capacidade de “saltar para outro canal de 1 MHz”, evitando assim a frequência afetada. Na Figura 13 está ilustrado o mecanismo “*Adaptive Isochronous Scheme*” utilizado pelo ANT. [16][11]

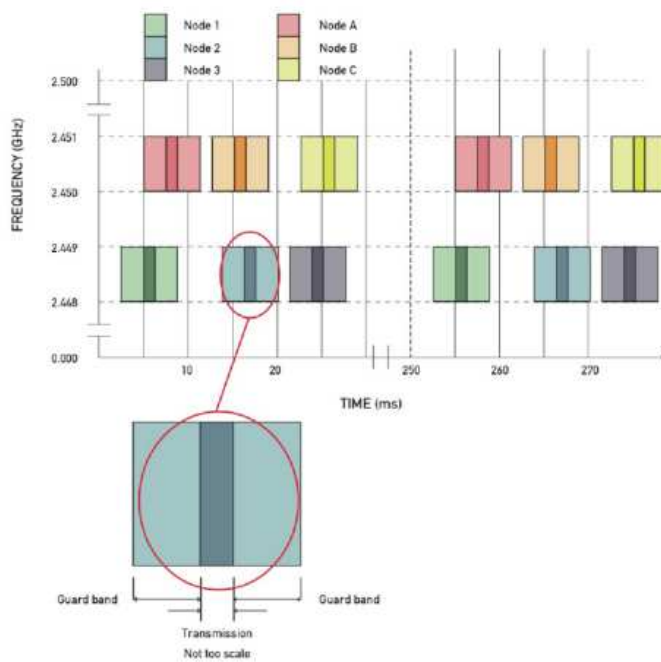


Figura 13 -“Adaptative Isochronous Scheme” utilizado pelo ANT [16]

Como pode ser visualizado na Figura 13, a banda de frequência 2,4 GHz é dividida em vários canais de 1 MHz, que por sua vez são devidos em vários *timeslots*. O número de *timeslots* presentes num canal depende do período do mesmo. Um *timeslot* consiste numa banda de guarda, num período pequeno de transmissão e noutra banda de guarda. Neste caso, os nós 1,2 e 3 partilham o mesmo canal utilizando diferentes *timeslots*. Por sua vez, os nós A,B e C utilizam *timeslots* que se sobrepõem aos dos nós 1,2 e 3, no entanto “saltaram” para um canal diferente, evitando assim interferências. [16]

Finalizando está pequena apresentação sobre os diferentes protocolos, pequena porque cada um deles poderia dar origem a uma tese só sobre eles, o próximo capítulo introduzirá mais detalhadamente o protocolo ANT, o protocolo sobre o qual o desenvolvimento deste projeto se baseou.

3. ESTUDO DA TECNOLOGIA ANT

Neste capítulo será apresentado, numa primeira etapa, os conceitos gerais e enquadramento ao protocolo ANT e em seguida demonstra-se de que forma os *transceivers* ANT comunicam com os microcontroladores.

3.1. ENQUADRAMENTO E APRESENTAÇÃO DO PROTOCOLO ANT

Em primeiro lugar, ANT é um acrónimo para “*Advanced and Adaptive Network Technology*”. É um protocolo proprietário desenvolvido pela empresa *Dynastream Innovations Inc.*, que foi desenvolvido para atuar em redes de sensores sem fios (*Wireless Sensor Network*), com o intuito de recolher e transmitir informações de sensores por meio de uma comunicação sem fios, como por exemplo, recolher informações sobre o ritmo cardíaco e velocidade de um ciclista e envia-la para um computador pessoal. Este protocolo opera na mesma frequência que o *WiFi*, *Bluetooth* e *ZigBee*, nomeadamente a 2.4GHz (*banda ISM*), mas com taxas de transferência de dados bastante mais lenta do que as tecnologias mencionadas, no entanto não necessita de transmitir tantos dados como as outras tecnologias, devido à natureza do seu desenvolvimento. Foi especificamente

desenvolvido para operar em dispositivos alimentados a baterias que requerem anos de vida útil das mesmas sem sacrificar as suas funcionalidades.

A rede de sensores sem fios ANT é constituída por nós. Um nó ANT tipicamente consiste num *transceiver* ANT, que processa o encapsulamento dos dados para serem transmitidos através da rede, um microcontrolador de baixo consumo energético onde o projetista desenvolve a sua aplicação e uma bateria para alimentação do nó.

Abordando este tema de como o protocolo ANT implementa uma rede, procede-se à demonstração do tipo de redes que este permite formar.

3.1.1. REDES IMPLEMENTADAS PELO PROTOCOLO ANT

Existem dois tipos de redes em que os *transceivers* ANT operam. A rede denominada como pública, onde não é necessário uma chave de rede e onde todos os dispositivos podem participar e trocar dados entre eles. Este é o tipo de rede definido por defeito num *transceiver* ANT. Por outro lado, pode ser também definida uma rede privada, onde apenas dispositivos específicos podem participar e em que todos devem possuir a mesma chave de rede. Quando se estiver a proceder à configuração da rede num *transceiver* ANT, é possível definir este dispositivo como participante numa rede pública colocando na configuração de rede o valor zero (*wildcard*). Qualquer outro valor define uma rede privada onde apenas os dispositivos com o mesmo número de rede podem participar. Neste caso, é necessário também configurar a chave de rede em todos esses dispositivos, pois sem este parâmetro os dispositivos não conseguem comunicar entre eles.

Outra característica da rede ANT é que esta consiste no princípio *ad-hoc* permitindo criar ou destruir conexões consoante a necessidade, sem utilizar uma infraestrutura fixa com dispositivos de controlo, como por exemplo encaminhadores de tráfego (*routers*).

No que diz respeito a topologias de rede, o protocolo ANT abrange uma elevada variedade de topologias, desde as mais simples como *peer-to-peer* até redes mais complexas como *mesh*, pelo que é capaz de formar qualquer tipo de topologia.

Em termos de coexistência com outras tecnologias, que é sempre um aspeto importante em qualquer tecnologia de rede sem fios, o protocolo ANT possui um mecanismo que permite que este esteja inserido no mesmo ambiente que outras tecnologias que utilizam a mesma frequência de funcionamento.

3.1.2. COMUNICAÇÃO ENTRE NÓS DE REDE

Cada nó comunica com outro recorrendo à implementação de canais. Este funcionamento é incorporado utilizando um mecanismo de comunicação do tipo mestre-escravo. Quando um canal é definido num *transceiver* ANT é necessário indicar se esse canal vai operar como mestre ou como escravo, fazendo com que desta forma os nós estabeleçam canais entre si, mediante a configuração neles efetuada, em que o mestre controla a comunicação. A tecnologia ANT possibilita também que o mesmo transceiver ANT funcione ao mesmo tempo como mestre e como escravo, ou seja, o canal 0 pode funcionar como mestre e o canal 1 como escravo. Na Figura 14 encontra-se ilustrada a comunicação entre nós de rede e onde se pode visualizar a comunicação do tipo mestre-escravo que os transceivers ANT utilizam para comunicar entre si. Cada transceiver, pode individualmente, possui canais mestres e escravos, permitindo desta forma funcionar como escravo num seguimento de rede (onde recebe dados dos outros nós mestres), e funcionar como mestre noutra seguimento de rede (enviando os dados para outro nó).

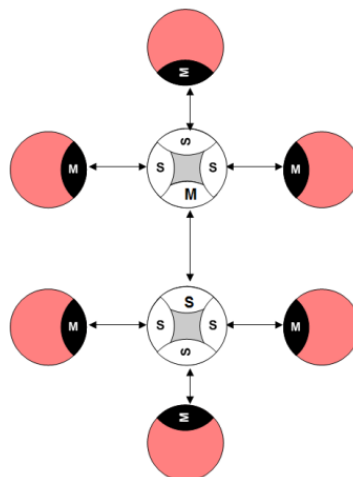


Figura 14 - Comunicação entre nós numa rede ANT

3.1.3. CANAIS UTILIZADOS PELO PROTOCOLO ANT

Existem diversos tipos de canais utilizados pelo protocolo ANT e estes especificam o tipo de comunicação que irá ocorrer no canal. Estes são os canais bidirecionais, os canais unidirecionais e os canais partilhados.

Cada nó é completamente independente, o que significa que não necessita de um coordenador central para encaminhar tráfego ou outro tipo de controlo, portanto cada canal gere-se de forma autónoma relativamente a outros canais. Cada *transceiver* ANT

possibilita a existência de múltiplos canais sendo todos eles completamente independentes entre si, permitindo ainda combinações de canais mestres e escravos no mesmo *transceiver* ANT. Na Tabela 2 estão representados os tipos de canais que o protocolo ANT suporta.

Tabela 2 – Tipos de canais implementados pelo protocolo ANT

Tipos de Canais	Descrição
Canais Bidirecionais	Canal Recetor Canal Transmissor
Canais partilhados	Canal Partilhado Recetor Canal Partilhado Transmissor
Canais unidirecionais	Canal Somente Recetor Canal Somente Transmissor

Com o intuito de melhor detalhar os canais incorporados no protocolo ANT, procede-se a explicação dos mesmos.

3.1.3.1. CANAIS BIDIRECIONAIS

Os canais bidirecionais suportam comunicações nos dois sentidos entre dispositivo mestre e dispositivo escravo. Estes são configurados como recetores nos dispositivos escravos e transmissores nos dispositivos mestres. Esta configuração define o modo de operação normal do canal, ou seja, estabelece qual é o *transceiver* mestre e qual é o *transceiver* escravo no normal funcionamento do canal, no entanto habilita a capacidade do canal em realizar a operação oposta ao seu modo normal de funcionamento.

3.1.3.2. CANAIS PARTILHADOS

Os canais partilhados são um tipo especial de canais bidirecionais em que a diferença para estes consiste na possibilidade de um *transceiver* mestre poder endereçar até 65535 *transceivers* escravos. Estes canais são utilizados para comunicações bidirecionais entre um *transceiver* mestre e vários escravos utilizando apenas um canal. Esta configuração é especialmente utilizada em situações em que um nó de rede central recolhe dados de um elevado número de sensores. Para tal, o *transceiver* mestre tem de possuir o conhecimento do endereço de cada *transceiver* escravo para poder efetuar os pedidos de envio de dados, através do método *pooling*.

3.1.3.3. CANAIS UNIDIRECIONAIS

Os canais unidirecionais permitem comunicação apenas num sentido. Este canal desabilita a operação oposta à sua configuração, por exemplo, se um canal for configurado como somente transmissor num *transceiver* mestre este apenas enviará dados e nunca receberá mensagens de outros nós.

3.1.4. EMPARELHAMENTO DE CANAIS

O pré-requisito para estabelecer um canal corretamente consiste no conhecimento comum, que tanto o *transceiver* mestre como o escravo, devem ter sobre o mesmo. Deve também respeitar alguns requisitos de sequência de configurações para que todas fiquem corretamente efetuadas, como por exemplo, definir em primeiro lugar a rede antes de abrir o canal. Para se configurar corretamente um *transceiver* ANT, em primeiro lugar é necessário configurar o tipo de rede em que este vai operar. Se for desejado uma rede privada em que apenas nós específicos tenham acesso aos dados que nela fluem é necessário configurar, em todos os dispositivos constituintes dessa rede, o número da mesma e a chave de rede. Por outro lado, se não se sentir esta necessidade esta configuração pode não ser realizada na medida em que a rede pública é a definida por defeito nos *transceivers*. Em seguida é necessário especificar como os canais vão operar, ou seja, configurar os canais desejados como mestres ou escravos. Depois, para que os nós de rede estabeleçam as conexões desejadas é necessário configurar o identificador de canal, em todos os canais. Finalmente, é necessário proceder à abertura dos canais.

Nesta altura o comportamento de um *transceiver* mestre e escravo é diferente. Quando um canal é aberto num *transceiver* escravo este procede à procura automática de um *transceiver* mestre que possua um canal com o mesmo identificador que o seu. Quando este é encontrado o *transceiver* escravo começa a receber as mensagens enviadas pelo mestre. Por outro lado, o *transceiver* mestre começa imediatamente a transmitir dados.

Na Figura 15 está representado o processo básico de configuração dos dispositivos para os mesmos consigam comunicar entre si.

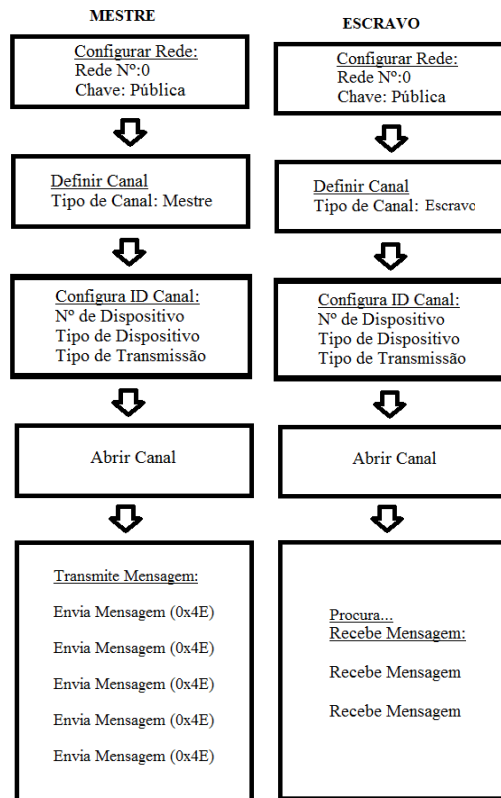


Figura 15 - Estabelecimento de um canal

O conhecimento comum que tanto os *transceivers* mestres como os escravos devem possuir sobre o canal consiste no identificador do canal. O identificador de canal cria uma identidade para que os dispositivos da rede consigam efetuar as conexões desejadas. Esta identidade é definida na altura da configuração do canal, que pode ser visualizada na terceira etapa da Figura 15. O identificador de canal consiste num conjunto de campos com diferentes finalidades, mas com um propósito comum, o de criar uma identidade com a finalidade de possibilitar a conexão dos *transceivers* escravos. Estes campos são definidos pelo utilizador podendo tomar os valores que este entender.

Assim sendo, o identificador de canal consiste em três campos distintos:

- Tipo de Transmissão – Consiste num campo de 1 byte. É utilizado para definir que tipos de dados irão ser transmitidos pelo canal. Por exemplo, se os dados transmitidos forem referentes a uma temperatura terá o valor 1, se forem referentes a uma velocidade terá o valor 2, etc., sendo possível desta forma identificar o tipo de dados que estão a ser transmitidos.

- Tipo de Dispositivo – Consiste num campo de 1 byte e é utilizado para definir o tipo de dispositivo que irá comunicar por esse canal. Por exemplo, um sensor de temperatura terá o valor 1, um sensor de humidade o valor 2, etc., e desta forma o dispositivo ANT sabe que tipo de dispositivo está a comunicar por esse canal.
- Número de Dispositivo – Consiste num campo de 2 bytes e pretende ser um valor único para um tipo de dispositivo, à semelhança de um número de série que identifica unicamente esse dispositivo.

Posto isto, existem vários métodos diferentes que um dispositivo escravo utiliza para se emparelhar com um dispositivo mestre. Quando um canal escravo é definido este inicia uma procura por um dispositivo mestre para efetuar o emparelhamento, de acordo com o método adotado.

Estes podem ser total ou parcial conhecimento do identificador de canal. No caso de ter conhecimento total do identificador de canal, o dispositivo escravo tem definido no seu identificador de canal, as mesmas configurações que o dispositivo mestre. Se possuir pelo menos um dos campos igual ao do dispositivo mestre tem um conhecimento parcial. Os campos que este não conhece são definidos com uma *wild-card* (preenchido com zeros). Na Figura 16 está representado um exemplo em que o dispositivo escravo conhece a totalidade do identificador de canal que irá procurar na rede.

Neste caso, o dispositivo escravo só se irá emparelhar com um dispositivo com campos exatamente iguais aos seus.

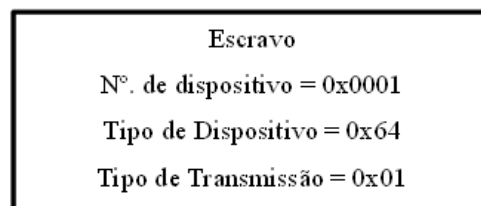


Figura 16 - Conhecimento Total do Identificador de Canal

Na Figura 17, está representado um exemplo em que o dispositivo escravo não conhece um campo do identificador de canal. Neste caso, procurará um dispositivo em que os outros dois campos coincidam.

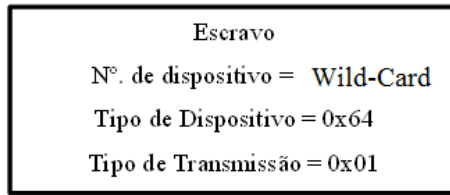


Figura 17 - Conhecimento de dois campos do Identificador de Canal

Na Figura 18 está representado um exemplo em que o dispositivo escravo apenas conhece um campo do identificador de canal. Neste caso, irá proceder da mesma forma e emparelhar-se com um dispositivo que tenha o campo igual ao seu.

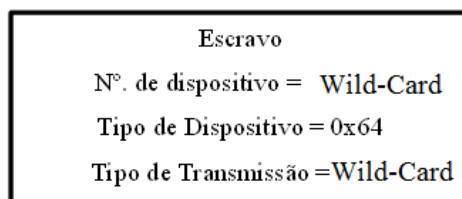


Figura 18 - Conhecimento de um campo de Identificador de Canal

Na Figura 19 está representado um exemplo em que o dispositivo escravo desconhece totalmente o identificador de canal. Neste caso o dispositivo escravo emparelha-se com o primeiro dispositivo mestre que encontra.

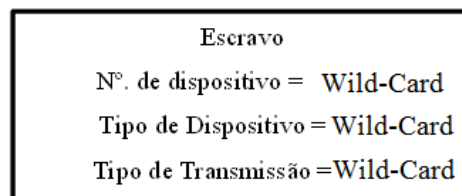


Figura 19 - Desconhecimento do campo de Identificador de Canal

Demonstrado o modo como os nós de rede estabelecem conexões entre si e como estas são efetuadas, serão apresentados em seguida o tipo de transmissões de mensagens que são transmitidas entre os mesmos.

3.1.5. TIPOS DE TRANSMISSÃO DE MENSAGENS TRANSMITIDAS PELA REDE

O protocolo ANT suporta diferentes tipos de transmissão de mensagens, nomeadamente três. A escolha do tipo de mensagens depende do tipo de aplicação está a ser desenvolvida e da necessidade ou não da confirmação da entrega de mensagem. Assim, os tipos de transmissão de mensagens suportadas pelo protocolo ANT são:

- *Broadcast* – As mensagens em *broadcast* são mensagens em que não é esperada uma resposta de confirmação de quem recebe. Isto significa que o transmissor da mensagem, quer seja um dispositivo mestre ou escravo, não tem conhecimento se a mensagem foi recebida com sucesso pelo recetor a que estava destinado. Dos três tipos de mensagens este é o que oferece menor consumo de energia e é o tipo de transmissão por defeito do protocolo ANT. Este tipo de mensagens são ideais para aplicações onde a transmissão é feita de “um-para-muitos”, ou seja de um nó mestre para vários escravos. Por exemplo, um sensor com aplicação num desporto que envie os seus dados simultaneamente para um relógio e um computador pessoal.
- *Acknowledgement* – Este tipo de mensagens originam e aguardam uma resposta do dispositivo recetor. Esta resposta é automaticamente tratada pelo protocolo ANT e por essa razão é transparente para o controlador. Ao enviar uma mensagem deste tipo o controlador será notificado, quer a mensagem seja recebida ou não. No caso de a mensagem não ser recebida com sucesso, a retransmissão da mesma deve ser feita pelo controlador pois o protocolo ANT não retransmite automaticamente este tipo de mensagens. Estas mensagens são preferenciais para aplicações onde o transmissor tem de ter conhecimento se a mensagem foi recebida com sucesso ou não. Por exemplo, um controlo remoto de qualquer tipo ou um atuador que confirme a receção correta da mensagem.
- *Burst* – As mensagens em rajada permitem a transferência, o mais rápido possível, de um elevado número de dados. O sucesso, ou não, de uma transmissão deste tipo é comunicado ao controlador, ou seja, se a transmissão dos dados não ocorreu de forma correta o protocolo ANT procede ao reenvio automatico dos mesmos, até um máximo de cinco vezes. Este tipo de mensagens é ideal para aplicações em que seja necessário enviar grandes quantidades de dados. Por exemplo, um relógio utilizando esta tecnologia pode enviar dados para um computador depois de uma sessão de treinos de um atleta.

3.1.6. EXEMPLOS DE CONFIGURAÇÃO DE UMA REDE ANT

Para melhor compreensão de como uma rede deve ser configurada, um exemplo simples é apresentado na Figura 20.

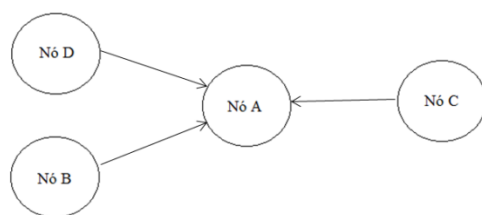


Figura 20 - Exemplo de uma rede ANT

Neste exemplo, pretende-se demonstrar uma situação em que diversos nós (B,C,D) recolhem dados de um sensor e enviam os dados para o nó A, mas que não comunicam entre si, ou seja, cada nó comunica unicamente com o nó A.

Na Tabela 3, está presente a configuração entre o nó A e o nó B. O nó B atuará como o dispositivo mestre na comunicação e o nó A como escravo. Como pode ser visualizado na mesma, ambos dispositivos têm configurações idênticas para se poderem emparelhar, excetuando o tipo de canal.

Tabela 3 Configurações do Canal entre o nó A e o nó B

Nó	Parâmetro	Valor	Descrição
B (Mestre)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	1	Número do Nó B
	Tipo de Transmissão	1	Tipo de Transmissão do Nó B
	Tipo de Dispositivo	1	Tipo de Dispositivo do Nó B
	Tipo de Canal	0x10	Canal Transmissor Bidirecional
	Tipo de Dados	0x4E	Dados: <i>Broadcast</i>
A (Escravo)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	1	Número do Nó A
	Tipo de Transmissão	1	Tipo de Transmissão do Nó A
	Tipo de Dispositivo	1	Tipo de Dispositivo do Nó A
	Tipo de Canal	0x00	Canal Recetor Bidirecional
	Tipo de Dados	0x4E	Dados: <i>Broadcast</i>

Na Tabela 4, está presente a configuração entre o nó A e o nó C. O nó C atuará como o dispositivo mestre na comunicação e o nó A como escravo. Como pode ser visualizado na mesma, ambos dispositivos têm também configurações idênticas para se poderem emparelhar, excetuando o tipo de canal e o tipo de dados que irão ser transmitidos pelo canal, que neste caso serão dados do tipo *Acknowledgement*.

Tabela 4 - Configuração do Canal entre o nó A e o nó C

Nó	Parâmetro	Valor	Descrição
C (Mestre)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	10	Número do Nó C
	Tipo de Transmissão	1	Tipo de Transmissão do Nó C
	Tipo de Dispositivo	2	Tipo de Dispositivo do Nó C
	Tipo de Canal	0x10	Canal Transmissor Bidirecional
	Tipo de Dados	0x4F	Dados: <i>Acknowledgement</i>
A (Escravo)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	10	Número do Nó A
	Tipo de Transmissão	1	Tipo de Transmissão do Nó A
	Tipo de Dispositivo	2	Tipo de Dispositivo do Nó A
	Tipo de Canal	0x00	Canal Recetor Bidirecional
	Tipo de Dados	0x4F	Dados: <i>Acknowledgement</i>

Por último, a configuração entre o nó A e o nó D está presente na Tabela 5. O nó D atuará como o dispositivo mestre na comunicação e o nó A como escravo. Como pode ser visualizado na mesma, ambos dispositivos têm configurações idênticas para se poderem emparelhar, excetuando o tipo de canal e o tipo de dados que irão ser transmitidos pelo canal, que neste caso serão dados do tipo *Broadcast*.

Tabela 5 Configuração do Canal entre o nó A e o nó D

Nó	Parâmetro	Valor	Descrição
D (Mestre)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	2	Número do Nó D
	Tipo de Transmissão	1	Tipo de Transmissão do Nó D
	Tipo de Dispositivo	1	Tipo de Dispositivo do Nó D
	Tipo de Canal	0x10	Canal Transmissor Bidirecional
	Tipo de Dados	0x4E	Dados: <i>Broadcast</i>
A (Escravo)	Número de Rede	0	Rede por Defeito
	Número de Dispositivo	2	Número do Nó A
	Tipo de Transmissão	1	Tipo de Transmissão do Nó A
	Tipo de Dispositivo	1	Tipo de Dispositivo do Nó A
	Tipo de Canal	0x00	Canal Recetor Bidirecional
	Tipo de Dados	0x4E	Dados: <i>Broadcast</i>

Como podemos visualizar na Tabela 5, o tipo de dispositivo é o mesmo. Este campo indica que se trata de dois dispositivos idênticos, como por exemplo dois sensores de temperatura a medir essa grandeza em dois locais diferentes. No entanto, por terem números de dispositivos diferentes, é possível diferencia-los na rede apesar de se tratarem do mesmo tipo de dispositivo.

Procede-se em seguida a demonstração de como um microcontrolador pode interagir com um transceiver ANT assim como as mensagens que ambos trocam entre si.

3.2. INTERAÇÃO ENTRE MICROCONTROLADOR E TRANSCEIVER ANT

Nesta secção serão apresentados os possíveis modos de comunicação que podem ser utilizados para efetuar uma comunicação entre um microcontrolador e um *transceiver* ANT. Na Figura 21 encontram-se ilustradas as duas formas possíveis de comunicar com um *transceiver* ANT.

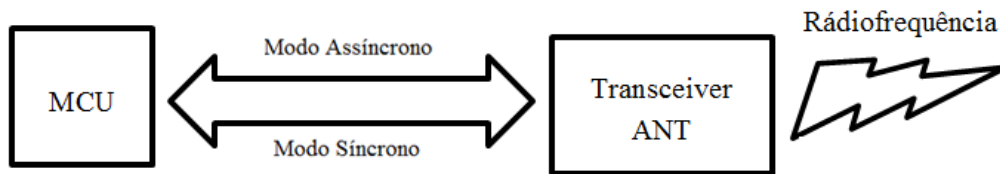


Figura 21 - Comunicação entre Microcontrolador e Transceiver ANT

Os modos de comunicação entre microcontrolador e transceiver ANT consistem em:

- Modo Síncrono – Neste modo, é necessário efetuar um procedimento específico para que a comunicação entre microcontrolador e transceiver ANT possa acontecer de maneira correta. A transmissão dos dados pode ser efetuada utilizando, por exemplo, um periférico *SPI*, que possui a capacidade de controlo de fluxo de dados. Neste modo o microcontrolador deve ser definido como “dispositivo escravo” pois o transceiver ANT é sempre um “dispositivo mestre”, que providencia o sinal de relógio para a troca de mensagens. Os procedimentos necessários para as comunicações são apresentados em seguida, onde também serão apresentados os pinos necessários para as ligações físicas entre ambos, que podem ser visualizados no *datasheet* do transceiver ANT. Assim o procedimento é o seguinte:
- Comunicação entre microcontrolador e transceiver ANT:
 1. O microcontrolador sinaliza o *transceiver* ANT que deseja transmitir dados, colocando a linha *SMSGRDY* a nível lógico baixo (0).
- Comunicação em ambos sentidos:
 2. O transceiver ANT coloca a linha *SEN* a nível lógico baixo (0), para indicar o início de uma transmissão de dados.
 3. Em seguida, o microcontrolador coloca a linha *SRDY* a nível lógico baixo para indicar se se encontra preparada para a transmissão.
 4. Depois de ambas as linhas estarem a nível lógico baixo, o transceiver ANT transmite sempre o primeiro byte, o de sincronismo. O bit menos significativo deste byte indica a direção dos restantes bytes da mensagem (0 – Receção de Mensagem, ANT-Microcontrolador; 1 – Envio de Mensagem, Microcontrolador-ANT)

5. Se o byte de sincronismo indicar a receção de uma mensagem, os restantes bytes da mensagem vão se enviados pelo transceiver.
6. Se o byte de sincronismo indicar o envio de uma mensagem, os restantes bytes devem ser enviados pelo microcontrolador.

O controlo de fluxo é realizado através da alternância da linha SRDY, ou seja, quando o microcontrolador deseja enviar um byte, coloca a linha a nível lógico baixo. Depois o transceiver ANT gera o sinal de relógio para o microcontrolador enviar o byte a essa taxa de relógio. Convém salientar que o controlo de fluxo tanto pode ser feito ao nível do byte como ao nível do bit. De qualquer uma das formas o controlo é efetuado da mesma forma. Na Figura 22 está representado o mecanismo de comunicação síncrona entre o microcontrolador e módulo ANT.

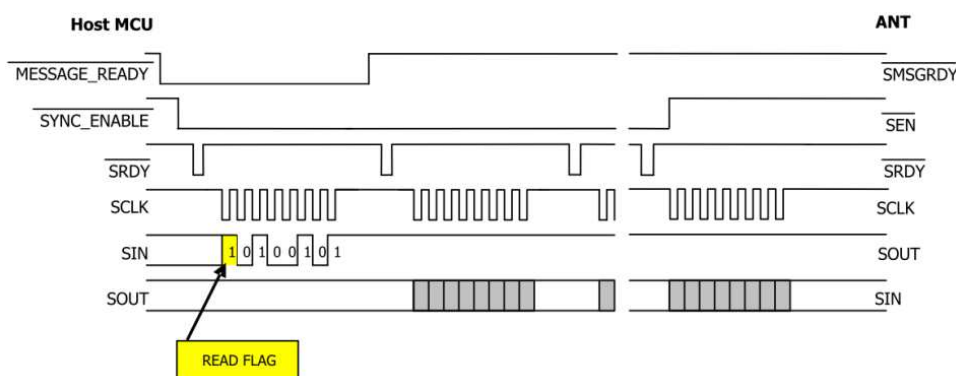


Figura 22 - Comunicação Síncrona entre microcontrolador e o transceiver ANT

- Modo Assíncrono – Neste modo, não existe um procedimento a efetuar para enviar os dados. Quando o microcontrolador desejar enviar dados, envia-os sem efetuar qualquer tipo de aviso, assim como o *transceiver* ANT. O controlo de fluxo é efetuado através da linha RTS (*Request To Send*) e esta é controlada pelo módulo ANT, que a cada mensagem corretamente recebida ou enviada coloca esta linha a nível lógico alto (1), aproximadamente 50 μ S, em que durante este tempo não recebe nem envia qualquer mensagem. Como não existe qualquer sinalização de transmissão por parte do transceiver, o microcontrolador deve estar sempre pronto para receber dados do transceiver ANT. A transmissão é feita a taxa definida pela seleção do *Baud Rate* que tem de ser igual em ambos dispositivos.

Não existe controlo de fluxo ao nível do bit nem do byte, apenas do pacote completo e caso as mensagens sejam incompletas ou incorretamente formatadas serão descartadas pelo módulo ANT. Na Figura 23 encontra-se representado o mecanismo de comunicação assíncrono entre microcontrolador e módulo ANT.

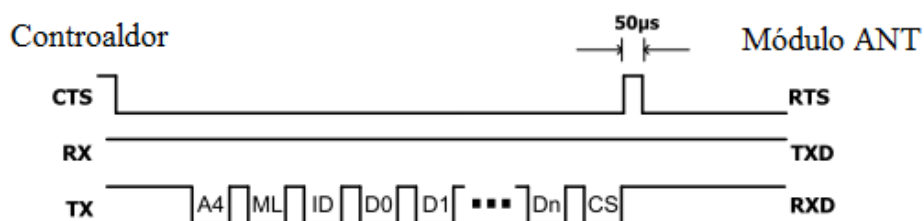


Figura 23 - Comunicação Assíncrona entre microcontrolador e módulo ANT

Apresentados os modos de comunicação entre um microcontrolador e um transceiver ANT procede-se à demonstração do formato que as mensagens entre ambos devem ter.

3.2.1. ESTRUTURA DAS MENSAGENS SUPORTADAS PELO *TRANSCEIVER ANT*

Nesta secção será apresentada a formatação dos pacotes e as estruturas que as mensagens devem ter para poderem ser corretamente transmitidas. Tipicamente, os pacotes têm a estrutura apresentada na Figura 24:

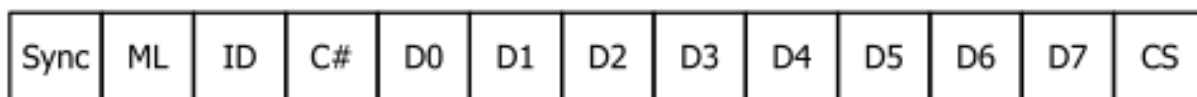


Figura 24 - Estrutura da mensagem ANT

Portanto, cada mensagem começa com o campo de sincronismo. Este campo difere consoante o tipo de comunicação entre o microcontrolador e o dispositivo ANT, ou seja, se a comunicação for assíncrona, este campo tem sempre o mesmo valor de 0xA4. Se a comunicação for síncrona este byte pode tomar dois valores, 0xA4 para transmissões no sentido do transceiver ANT para o microcontrolador, ou 0xA5 para comunicações no sentido inverso, pois neste modo é sinalizada a vontade de transmitir dados pelos dois dispositivos.

Segue-se o campo do tamanho da mensagem. O valor deste campo varia consoante o número de bytes que constituem o campo de dados a serem enviados, por exemplo, se forem enviados 8 bytes de dados, este campo possuirá o valor 8.

Em seguida vem o campo de identificação da mensagem. Este campo identifica o tipo de mensagem que o pacote contém, ou seja, cada mensagem suportada pelo protocolo *ANT* tem um identificador associado e quando o transceiver interpreta este valor sabe de que tipo de mensagem se trata e quais devem ser os campos seguintes. Por exemplo, se for uma mensagem do tipo *broadcast*, este campo possui um valor de identificação (0x4E) e desta forma o *transceiver* ANT tem conhecimento que os dados presentes no campo de dados devem ser enviados utilizando o modo *broadcast*, ou se for uma mensagem para configuração do canal, esta possui outro valor de identificação (0x42) e desta forma o transceiver ANT tem conhecimento de que os valores do campo de dados deverão ser utilizados na configuração do canal. Este campo consiste na forma que o *transceiver* ANT utiliza para diferenciar as mensagens que recebe.

Posteriormente surge o campo de dados. Neste campo são enviados os dados que realmente necessitam de ser transmitidos e possui um tamanho máximo de oito bytes.

Finalmente, surge o campo *checksum*. Este campo consiste na operação lógica XOR de todos os bytes anteriores do pacote. Este procedimento é realizado pelo transmissor e enviado no final da mensagem. O receptor realiza também este procedimento e compara o seu valor com o valor enviado no pacote e desta forma confirma, ou não, se a mensagem foi corretamente recebida.

Na Tabela 6, são apresentados sucintamente os campos constituintes de uma mensagem ANT e o tamanho de cada um deles.

Tabela 6 – Estrutura de uma mensagem ANT

Byte	Nome	Tamanho (bytes)	Descrição
0	SYNC	1	Byte de sincronismo que depende do tipo de comunicação utilizada
1	LENGTH	1	Número de bytes da mensagem
2	ID	1	Identificador do tipo de mensagem
3	DATA	N	Dados
N+1	CHECKSUM	1	XOR de todos os bytes anteriores

4. DESENVOLVIMENTO DO PROJETO

O principal objetivo deste projeto é o desenvolvimento de um sistema de aquisição da temperatura corporal em vários pontos, que o utilizador possa usar com o mínimo desconforto físico, para monitorização da mesma nas estações de monitorização. Este capítulo pretende demonstrar o desenvolvimento prático de todo o sistema e de que forma a rede de sensores sem fios foi implementada

4.1. ARQUITETURA DA REDE E FUNCIONALIDADE DO SISTEMA

A estrutura da rede desenvolvida consiste em quatro nós. O primeiro, configurado como mestre, realizará a aquisição da temperatura num ponto do corpo e procederá ao tratamento dos dados com o intuito de os enviar para os restantes nós. O segundo nó, possui um funcionamento exatamente igual ao primeiro, mas efetua a medição da temperatura nouro ponto do corpo. O terceiro nó, configurado como escravo, representa uma estação fixa de visualização dos dados, como um computador pessoal. Este nó simula uma estação de supervisora posicionada no local de trabalho de um profissional de saúde, em que este possa a qualquer momento visualizar o estado da temperatura do paciente e onde os dados são guardados num ficheiro de texto, com o propósito de criar um histórico da temperatura

corporal para uma possível futura análise. Por último, o quarto nó, configurado também como escravo, representará uma outra estação de visualização, operando num canal diferente da estação do nó B, e pretende representar uma estação de visualização local, como um computador portátil, onde o paciente pode também visualizar o valor da sua temperatura corporal. Assim, o primeiro passo é a definição da estrutura para atingir esse fim conforme ilustrado na Figura 25.

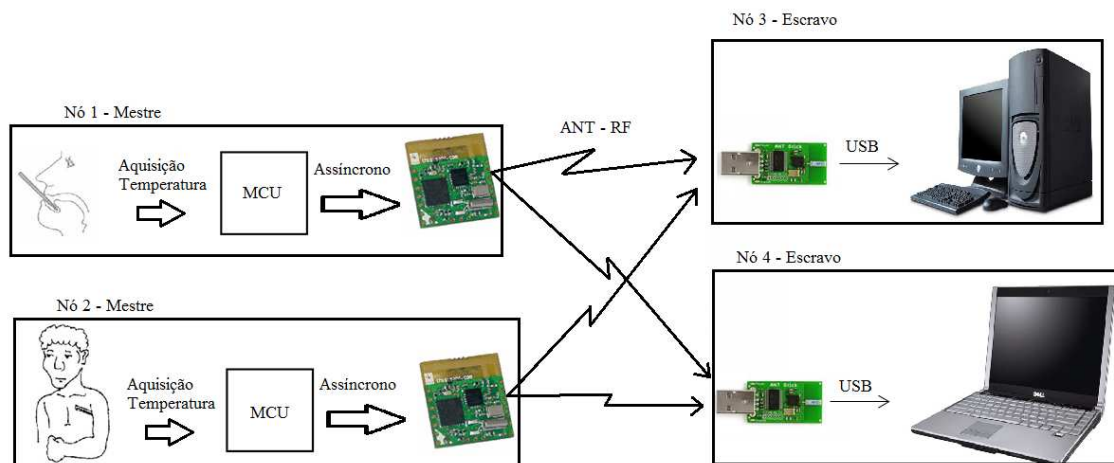


Figura 25 - Estrutura do projeto

4.2. HARDWARE UTILIZADO PARA O DESENVOLVIMENTO DO SISTEMA

A estrutura de rede desenvolvida consiste na utilização de transceivers ANT conectados a controladores com o intuito de os interligar numa rede sem fios. Normalmente estes controladores resumem-se a computadores pessoais e a microcontroladores de baixo consumo energético. No desenvolvimento deste projeto o microcontrolador utilizado foi o PIC 18f4680. Os microcontroladores PIC são bastante populares e versáteis, sendo amplamente utilizados no desenvolvimento de diversas aplicações e têm evoluído a um ritmo elevado, oferecendo hoje em dia muitas vantagens para quem sabe trabalhar com este tipo de produtos.

Este PIC tem uma arquitetura *RISC (Reduced Instruction Set Computer)* que é uma linha de arquitetura de computadores que favorece um conjunto simples e pequeno de instruções que levam aproximadamente a mesma quantidade de tempo para serem executadas. Possui uma memória de programa de 32 Kbytes, 2048 bytes de RAM, 1 temporizador de 8 bits e 3 de 16 bits, 13 canais de conversão Analógica/Digital, periféricos de comunicação digital como *USB, CAN, SPI e EUSART*, esta ultima utilizada para a comunicação com o

transceiver *ANT*. Para além de todas as características mencionadas, outro motivo bastante relevante para a escolha deste microcontrolador recai sobre o fato deste microcontrolador possuir a tecnologia *nanoWatt* que possibilita uma redução significativa do consumo energético, fundamental num projeto desta natureza.

Foi desenvolvida uma placa de circuito impresso que facilita a interação com o microcontrolador e possui todos os componentes necessários para que este funcione corretamente, desde cristais, conetores, interface RS-232 e regulador de tensão. Esta placa pode ser alimentada recorrendo a uma fonte de alimentação externa onde o regulador de tensão fornece a tensão desejada ou recorrendo a uma bateria de 3,3 V, que promove a independência e a mobilidade deste sistema. Esta ultima utilizada como alimentação tanto da placa como dos *transceiver*. O esquemático da placa de circuito impresso desenvolvida encontra-se ilustrado na Figura 26.

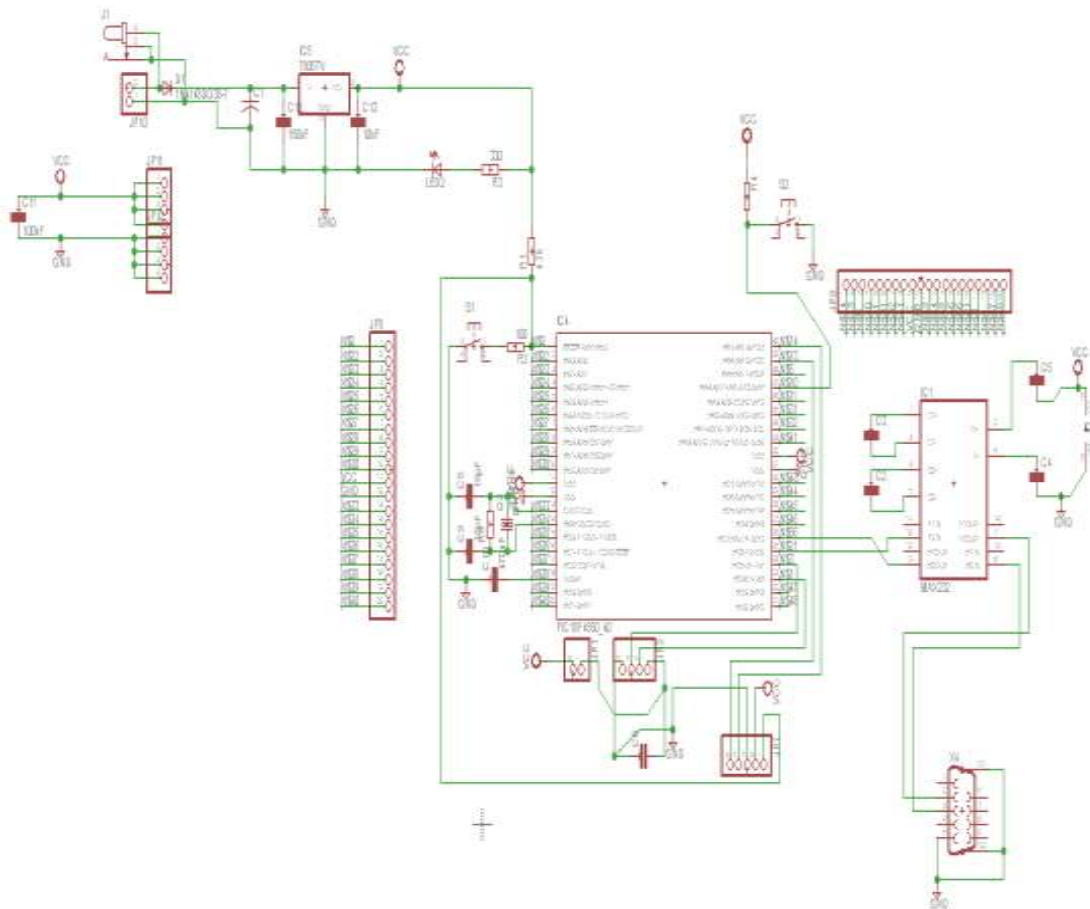


Figura 26 - Esquemático da placa de circuito impresso para o microcontrolador pic 18f4680

Os outros controladores utilizados foram dois computadores pessoais, que independentemente da marca ou modelo, dispensam qualquer tipo de demonstração.

Demonstrados os controladores utilizados serão em seguida apresentados os transceivers ANT utilizados no desenvolvimento do projeto. Estes *transceivers* ANT fazem parte integrante de um kit de desenvolvimento adquirido no fabricante dos mesmos, a *Dynastream Innovations INC.*, que os providencia a projetistas e empresas para desenvolverem produtos e soluções utilizando a sua tecnologia. Este kit de desenvolvimento possui quatro *transceivers* e por esse motivo redes de maior complexidade não podem ser efetuadas, no entanto permite uma introdução à tecnologia e efetuar pequenos projetos interessantes. Na Figura 27, encontra-se ilustrado o kit de desenvolvimento utilizado.



Figura 27 - kit de desenvolvimento utilizado para o desenvolvimento do projeto

No que diz respeito ao *transceiver* em si, este possui todas as funcionalidades incorporadas para o seu funcionamento, e apenas disponibiliza um interface sob a forma de conectores para interligação com os controladores.

Na Figura 28 encontra-se ilustrado o *transceiver* ANT utilizado no desenvolvimento deste projeto.

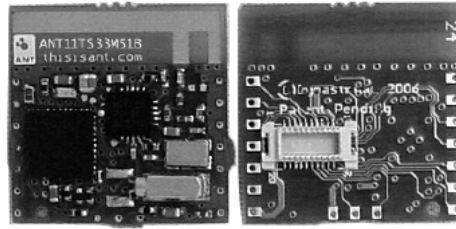


Figura 28 - Transceiver ANT utilizado para interligar formar os nós de rede

4.3. SOFTWARE UTILIZADO PARA O DESENVOLVIMENTO DO PROJETO

A plataforma de desenvolvimento utilizada para o desenvolvimento do projeto no que diz respeito à programação do microcontrolador foi o *MPLAB™* da *Microchip*, adquirido no site do fabricante e possui suporte técnico. Foi adicionado ao *software* um suplemento que permite o programador utilizar a linguagem de programação C para desenvolver o projeto. Este suplemento trata-se do compilador C18 e foi adquirido também no *site* do fabricante. **Na Erro! A origem da referência não foi encontrada.** é possível visualizar uma imagem do *software* utilizado.

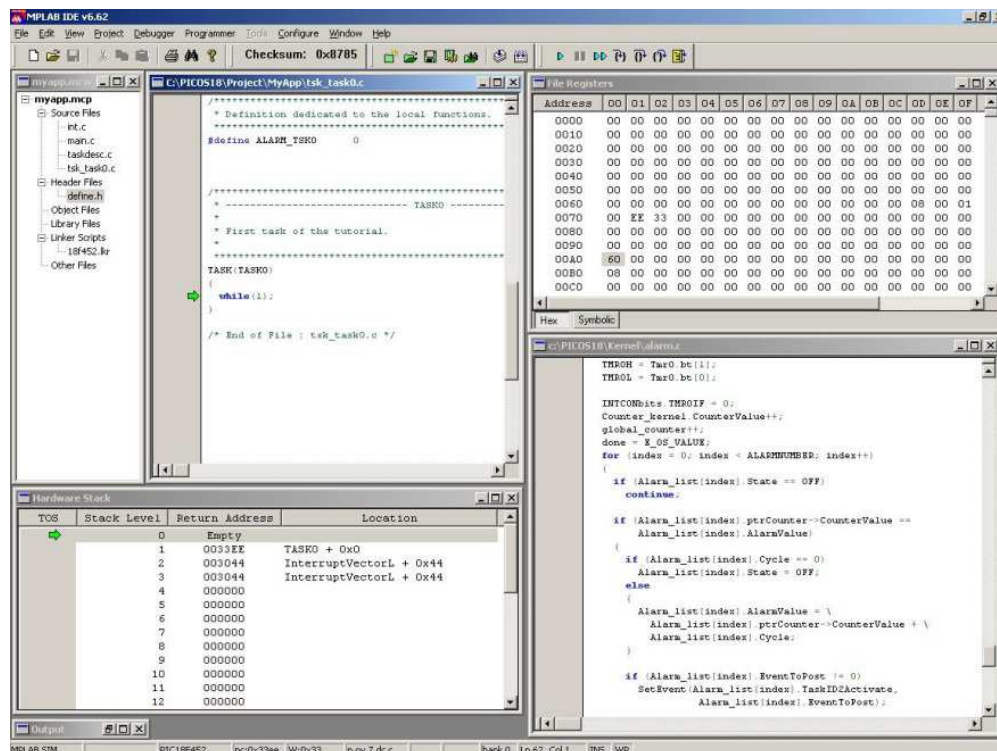


Figura 29 - Plataforma de desenvolvimento MPLAB™ para desenvolvimento da programação do microcontrolador

Outra plataforma de desenvolvimento necessária para a realização de todo o sistema foi o *SharpDevelop*. Este *software* permite o desenvolvimento de programas utilizando a linguagem *C Sharp*, que permitiu o desenvolvimento das interfaces gráficas para visualização dos dados. Na Figura 30 encontra-se representado o *software* em questão.

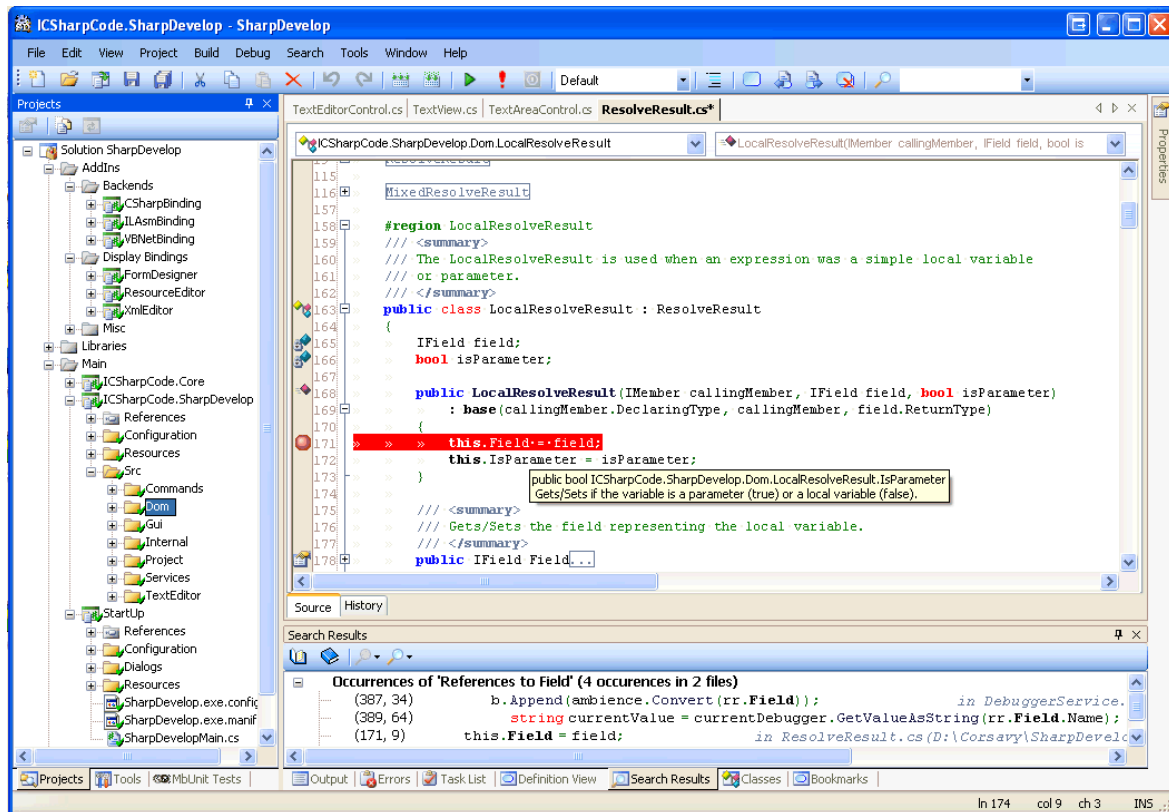


Figura 30 - Plataforma de desenvolvimento SharpDevelop utilizado para o desenvolvimento das interfaces gráficas

4.4. NÓ 1 E NÓ 2 - DEFINIDOS COMO MESTRES NA REDE

De forma a esclarecer quais os nós que estão a ser abordados, a Figura 31 pretende demonstrar os mesmos. Estes nós são definidos como mestres, pois são eles que vão proceder à aquisição dos dados. Quando os nós recolherem os dados tomarão a iniciativa e darão início a transmissão dos mesmos para os nós escravos da rede. Tipicamente, os nós definidos como mestres são responsáveis pela aquisição das grandezas físicas, enquanto que os nós escravos normalmente procedem à recolha dos dados de diversos nós mestres e estão normalmente associados a estações de monitorização.

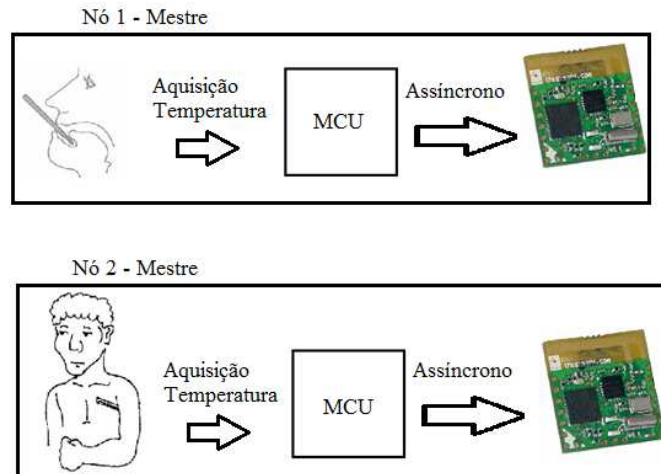


Figura 31 -Nós definidos como mestre na rede

O funcionamento dos nós configurados como mestres consiste numa primeira etapa, nas configurações necessárias, tanto no microcontrolador como no *transceiver ANT*. Este procedimento é apenas realizado uma vez em cada microcontrolador.

Depois, no normal funcionamento do sistema, é realizada periodicamente a aquisição da temperatura corporal. Em seguida é efetuado o processamento matemático dos dados e realizada a formulação das mensagens a serem trocadas entre microcontrolador e o *transceiver*. Quando esta etapa é finalizada, o microcontrolador coloca o sistema em suspensão, tanto *transceiver* como microcontrolador, de modo a pouparem a bateria que os alimenta e desta forma aumentar o tempo de vida útil das mesmas. Todo este procedimento será efetuado periodicamente mediante um evento temporal configurável pelo utilizador, que neste caso foi de dez em dez minutos.

Este evento temporal é realizado através de um *watchdog timer*, que periodicamente retira do modo de suspensão o microcontrolador. Em seguida, é utilizada uma variável como contador, que efetua as contagens de tempo necessárias para que a temporização de dez minutos seja conseguida. Foi equacionada a hipótese de efetuar a temporização de dez minutos com recurso a um temporizador de baixo consumo energético que o microcontrolador possui, no entanto, este requeria que o microcontrolador continuasse a funcionar normalmente, o que aumentaria o consumo de energia, e por esse motivo não foi utilizado.

Este procedimento é efetuado nos dois nós mestres responsáveis pela aquisição da temperatura. De modo a clarificar a sequência de operação deste nó, o fluxograma de funcionamento é demonstrado na Figura 32.

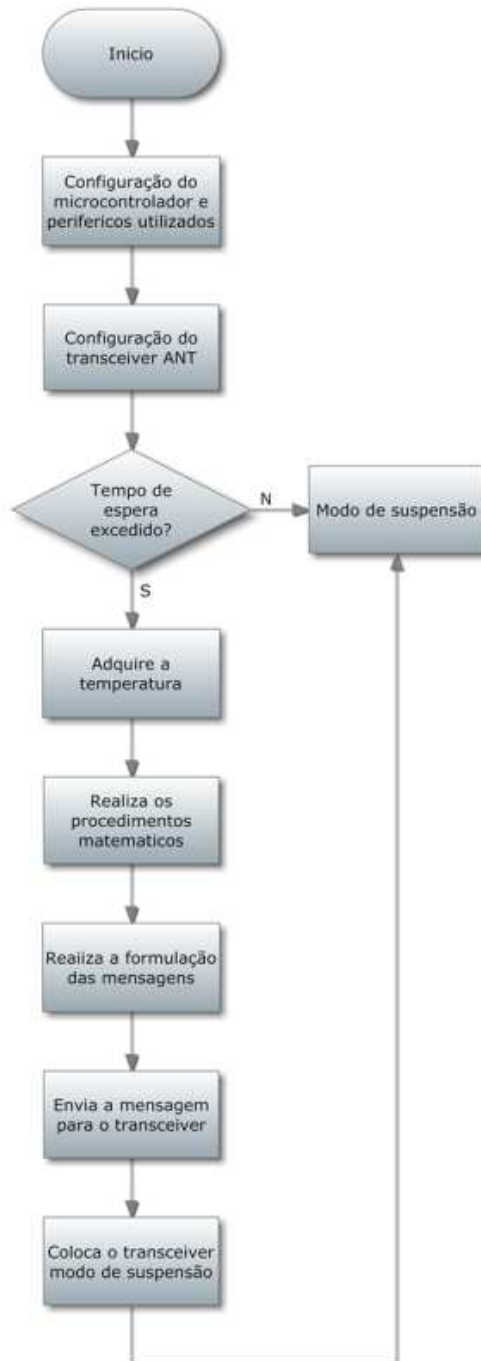


Figura 32 - Fluxograma do funcionamento do microcontrolador

Este nó pode se dividido em três partes funcionais distintas: A aquisição da temperatura, o processamento de dados e formulação de mensagens, e, por último, a comunicação entre microcontrolador e o *transceiver* ANT.

No entanto, antes de qualquer outro procedimento ser efetuado, é necessário configurar microcontrolador e o *transceiver* ANT de forma a atender às necessidades do desenvolvimento deste sistema, assim como efetuar as ligações físicas entre ambos.

4.4.1. COMUNICAÇÃO ENTRE MICROCONTROLADOR E *TRANSCEIVER* ANT

Para efetuar a comunicação entre o *transceiver* ANT e o microcontrolador foi necessário efetuar algumas ligações físicas entre ambos. O modo de comunicação utilizado na comunicação entre estes dois dispositivos foi o assíncrono, o qual pode ser visualizado na figura 10.

Estas ligações podem ser visualizadas na Figura 33, onde o sentido das setas indica o sentido de direção dos dados.

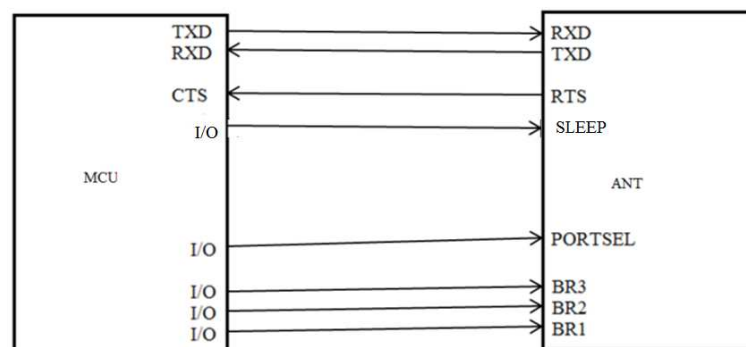


Figura 33 - Ligações físicas entre microcontrolador e *transceiver* ANT

A comunicação entre os dispositivos foi realizada através da utilização do periférico *EUSART* do microcontrolador. Este foi configurado para funcionar de forma assíncrona através da configuração dos registos *TXSTA* e *RCSTA*. No entanto, a seleção do modo de comunicação tem de ser efetuada nos dois dispositivos. Assim sendo, o *transceiver* ANT possui um pino (*PORTSEL*) que mediante o seu estado lógico, seleciona o tipo de comunicação a ser efetuada, assíncrona ou síncrona. Este pino é uma entrada do *transceiver* ANT, e é controlado pelo microcontrolador que o utiliza como um pino normal

de entrada/saída. Para ser selecionado o tipo de comunicação assíncrona este deve possuir o valor lógico zero, caso contrário seria selecionado a comunicação síncrona. O *baud rate* também necessita de ser selecionado nos dois dispositivos. No microcontrolador esta seleção é efetuada através da configuração do registo BAUDCON, enquanto no *transceiver ANT* a seleção é efetuada com a combinação dos estado lógico de três dos seus pinos (BR3, BR2, BR1) especialmente utilizados para o efeito, ou seja, o transceiver possui três pinos que consoante o seu estado lógico permitem selecionar diferente *baud rates*. Na Tabela 7 pode-se visualizar a relação entre o estado lógico dos pinos e a seleção do baud rate.

Tabela 7 – Tabela da relação dos pinos BR3, BR2 e BR1 com a seleção do Baud Rate

BR3	BR2	BR1	Baud Rate
0	0	0	4800
0	1	0	19200
0	0	1	38400
0	1	1	50000
1	0	0	1200
1	1	0	2400
1	0	1	9600
1	1	1	57600

Em seguida, foi efetuada a ligação do pino *RTS* do transceiver com um pino de entrada/saída do microcontrolador, denominado *CTS*. Sempre que o transceiver *ANT* recebe uma mensagem corretamente, este coloca a o pino *RTS* a nível lógico alto por aproximadamente 50 μ S. Durante este tempo o microcontrolador não deve enviar qualquer mensagem pois o *transceiver* necessita deste tempo para processar a mensagem que foi recebida. Para respeitar este tempo, foi realizada uma temporização de 50 μ S utilizando um dos temporizadores do microcontrolador, assegurando desta forma o tempo de espera indicado.

4.4.2. CONFIGURAÇÃO DO MICROCONTROLADOR DO NÓ MESTRE

Para configurar o microcontrolador é necessário configurar alguns registos para que este fique operacional de acordo com as necessidades impostas para a realização deste sistema. Em primeiro lugar, o microcontrolador foi configurado para operar com o oscilador

interno, ou seja a 8 MHz, através da configuração do registo *OSCCON* do mesmo. Em seguida foi configurado um conversor analógico/digital para proceder à aquisição dos dados provenientes dos sensores de temperatura. Depois, foram configurados dois temporizadores, em que um deles realiza a temporização de dez minutos para retirar o sistema do modo de suspensão e o outro realiza a temporização de 50 μ S necessária para a comunicação entre microcontrolador e o *transceiver ANT* ocorrer de forma correta.

Por último foi configurado o periférico *EUSART* de maneira a funcionar no modo assíncrono, com envio e receção de 8 bits, sem controlo de fluxo de dados e com um baud rate de 9600 *bps*. Todos estes registos podem ser facilmente consultados no *datasheet* do microcontrolador e por esse motivo não foram aqui expostos de forma mais detalhada mas sim de uma maneira geral.

4.4.3. CONFIGURAÇÃO DO *TRANSCEIVER ANT* DO NÓ DE AQUISIÇÃO DA TEMPERATURA

Depois de efetuar as configurações no microcontrolador é necessário proceder à configuração do *transceiver ANT*. Esta configuração é realizada através do envio, por parte do microcontrolador, de mensagens específicas para o *transceiver ANT*, que quando as recebe procede às configurações a elas associadas. No entanto, antes de enviar qualquer mensagem, é aconselhável proceder ao *reset* do dispositivo para ter conhecimento em que estado é que este se encontra. Para tal, basta colocar o pino de *reset* do *transceiver* a nível lógico alto. Posto isto, a primeira configuração a ser efetuada consiste na configuração de rede.

Como não foi evidente a necessidade de desenvolver uma rede privada para este projeto, esta foi configurada como pública, e sendo este o modo de configuração de rede por defeito do protocolo ANT, não foi também necessário proceder à sua configuração. No entanto, se fosse desejado, a configuração de uma rede poderia ser rapidamente efetuada através do envio de mensagens específicas para o efeito. Estas mensagens dizem respeito ao número atribuído à rede privada assim como a definição da chave de segurança da rede.

Assim sendo, a próxima configuração necessária diz respeito à inicialização dos canais. Esta configuração serve para indicar ao *transceiver ANT* qual é o canal que é necessário inicializar, como será o seu modo de funcionamento, se mestre ou escravo, e por último em que rede este canal vai operar.

Assim sendo esta mensagem consiste nos seguintes campos:

- *SYNC* – Campo de sincronismo. Este campo possui um valor único para comunicações assíncronas. Este valor é 0xA4.
- *MSG_LENGTH* – Campo que indica o número de bytes presentes no campo de dados. Como nesta mensagem são enviados três bytes como dados, este campo possui o valor 0x03.
- *MSG_ID* – Este campo identifica o tipo de mensagem que vai ser enviada. Como está mensagem diz respeito à inicialização do canal, este campo possui o valor de 0x42.
- *CHANNEL_NUM* – Este campo indica qual é o canal que vai ser inicializado. Neste caso, como o canal a ser utilizado é o canal zero, este possui o valor 0x00.
- *CHANNEL_TYPE* – Este campo indica se este canal vai funcionar como mestre ou escravo. Como se trata de um canal mestre este campo possui o valor de 0x10.
- *NETWORK* – Este campo indica o número da rede em que este canal vai operar. Como a rede foi definida como pública, este campo contém o valor 0x00.
- *CHECKSUM* – Este campo corresponde à operação lógica XOR de todos os campos anteriores. Neste caso esse valor corresponde a 0xE6.

A mensagem formulada com os campos específicos para a realização desta configuração pode ser visualizada na Figura 34.

Designação	SYNC	MSG_LENGTH	MSG_ID	CHANNEL_NUM	CHANNEL_TYPE	NETWORK	CHECKSUM
Valor	0xA4	0x03	0x42	0x00	0x10	0x00	0xE6

Figura 34 -Mensagem de ativação de um canal num Transceiver ANT

Em seguida é necessário configurar o identificador do canal. Esta configuração permite indicar ao *transceiver* o número de série do dispositivo que irá enviar dados pelo canal, o tipo de dispositivo que este é, neste caso é um sensor de temperatura e permite atribuir um valor ao tipo de dados que este vai transmitir.

Assim sendo esta mensagem consiste nos seguintes campos:

- *SYNC* – Campo de sincronismo. Este campo possui um valor único para comunicações assíncronas. Este valor é 0xA4.
- *MSG_LENGTH* – Campo que indica o número de *bytes* presentes no campo de dados. Como nesta mensagem são enviados três bytes como dados, este campo possui o valor 0x05.
- *MSG_ID* – Este campo identifica o tipo de mensagem que vai ser enviada. Como está mensagem diz respeito à configuração do canal, este campo possui o valor de 0x51.
- *CHANNEL_NUM* – Este campo indica qual é o canal que vai ser inicializado. Neste caso, como o canal a ser utilizado é o canal zero, este possui o valor 0x00.
- *DEVICE_NUM_LSB* – Este campo indica o *byte* menos significativo do número do dispositivo. Neste caso, foi atribuído o número 54 a este dispositivo. Assim, este campo possui o valor 0x36.
- *DEVICE_NUM_MSB* – Este campo indica o *byte* mais significativo do número do dispositivo. Neste caso, como foi atribuído o número 54 a este dispositivo, este campo possui o valor 0x00.
- *DEVICE_TYPE* – Este campo indica que tipo de dispositivo vai comunicar por este canal. Neste caso, o sensor de temperatura foi definido como um dispositivo do tipo 1. Assim sendo, este campo possui o valor 0x01.
- *DATA_TYPE* – Este campo permite identificar o tipo de dados que o dispositivo vai enviar. Neste caso, foi definido o valor 5 para os dados que correspondem ao valor da temperatura. Assim, este campo possui o valor 0x05.
- *CHECKSUM* – Este campo corresponde à operação lógica XOR de todos os campos anteriores. Neste caso esse valor corresponde a 0xC2.

A mensagem formulada com os campos específicos para a realização desta configuração pode ser visualizada na Figura 35

Designação	SYNC	MSG_LENGTH	MSG_ID	CHANNEL_NUM	DEVICE_NUM_LSB	DEVICE_NUM_MSB	DEVICE_TYPE	DATA_TYPE	CHECKSUM
Valor	0xA4	0x05	0x51	0x00	0x36	0x00	0x01	0x05	0xC2

Figura 35 - Mensagem de configuração do identificador de um canal num Transceiver ANT

Finalmente, chegando a esta etapa é necessário enviar uma ordem para que o transceiver ative o canal correspondente. Esta mensagem consiste nos seguintes campos:

- *SYNC* – Campo de sincronismo. Este campo possui um valor único para comunicações assíncronas. Este valor é 0xA4.
- *MSG_LENGTH* – Campo que indica o número de *bytes* presentes no campo de dados. Como nesta mensagem é enviado 1 byte como dados, este campo possui o valor 0x01.
- *MSG_ID* – Este campo identifica o tipo de mensagem que vai ser enviada. Como está mensagem diz respeito à abertura do canal, este campo possui o valor de 0x4B.
- *CHANNEL_NUM* – Este campo indica qual é o canal que vai ser inicializado. Neste caso, como o canal a ser utilizado é o canal zero, este possui o valor 0x00.
- *CHECKSUM* – Este campo corresponde à operação lógica XOR de todos os campos anteriores. Neste caso esse valor corresponde a 0xEE.

A mensagem formulada com os campos específicos para a realização desta configuração pode ser visualizada na Figura 36.

Designação	SYNC	MSG_LENGTH	MSG_ID	CHANNEL_NUM	CHECKSUM
Valor	0xA4	0x01	0x4B	0x00	0xEE

Figura 36 - Mensagem de ativação de um canal num Transceiver ANT

Nesta altura, o *transceiver ANT* encontra-se preparado e a transmitir dados para a rede.

Este procedimento tem de ser realizado sempre que seja necessário atribuir novos canais ou inserir novos *transceivers* na rede.

Como foram utilizados dois canais em cada nó de rede, todo este procedimento foi novamente realizado mas para o canal 1. Isto implica, ligeiras mudanças nas mensagens enviadas. Estas mudanças consistem na indicação do número do canal 1, ao invés do canal 0, no campo do número de canal (*CHANNEL_NUM*) das diversas mensagens, a modificação do número de série de dispositivo presente nos campos *DEVICE_NUM_LSB* e *DEVICE_NUM_MSB* presentes na mensagem de configuração do identificar de um canal, pois os dispositivos devem possuir um número de série únicos na rede, e ao valor de *checksum* das diferentes mensagens que se modificam consoante os valores dos campos anteriores.

Executadas todas as configurações necessárias, procede-se a explicação de como a temperatura é medida.

4.4.4. AQUISIÇÃO DA TEMPERATURA

Para realizar a medição da temperatura foram utilizados dois sensores de temperatura do tipo *NTC*, que efetuam a medição da mesma em dois pontos distintos do corpo. Para o efeito, foi utilizado o circuito utilizado na Figura 37

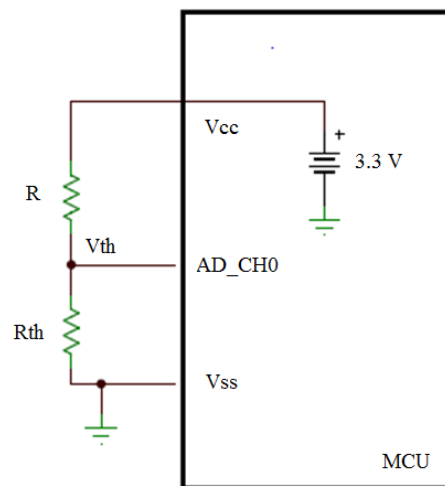


Figura 37 -Circuito aquisição de temperatura

Sendo a *NTC* um sensor de temperatura que varia a sua resistência consoante a temperatura em que está inserido, ou seja, quanto maior for a temperatura menor será a sua resistência, é possível obter o valor da sua resistência através de alguns procedimentos matemáticos.

Assim sendo, o circuito da Figura 37 consiste num divisor de tensão que permite relacionar a resistência do sensor de temperatura com a tensão aos seus terminais. Esta pode ser obtida através da equação 1.

$$V_{th} = \frac{R_{th}}{R_{th}+R} \times V_{CC} \quad (1)$$

Desta forma o valor V_{th} é proporcional ao valor da resistência do sensor de temperatura e será este valor que o conversor analógico/digital irá adquirir. Quando este realizar a leitura do valor V_{th} , gera um valor de acordo com a mesma. Este valor pode ser obtido utilizando a equação 2, onde o valor de n representa o valor de bits do conversor analógico/digital.

$$Valor_{adc} = \frac{V_{th}}{V_{CC}} \times 2^n \quad (2)$$

Nesta altura, é necessário relacionar o valor da resistência do sensor de temperatura com o valor no registo do conversor/analógico digital, ou seja, à medida que a temperatura muda o valor deste registo muda também logo é necessário traduzir computacionalmente o valor desse registo no valor da resistência do sensor de temperatura. Para o efeito, e como pode ser visualizado nas equações 1 e 2, estas têm uma grandeza em comum, a tensão V_{th} (*Tensão aos terminais do sensor de temperatura*) e assim, combinando as duas equações, obtemos a equação 3 que relaciona o valor do registo com a resistência do sensor.

$$R_{th} = R \times \frac{Valor_{adc}}{2^n - Valor_{adc}} \quad (3)$$

Por último, para obter o valor da temperatura é utilizada a equação 4 que relaciona o valor da resistência do sensor de temperatura com a temperatura que este está a medir. Nesta equação pode-se visualizar três parâmetros, fornecidos pelo fabricante do sensor, que demonstram o comportamento do sensor mediante a temperatura. Normalmente o fabricante fornece o valor da resistência do sensor de temperatura à temperatura de 25° Celsius. Neste caso, o sensor possui uma resistência de 10000 Ohm à temperatura de 25° Celsius e estão representados na equação como R_0 e T_0 , respetivamente. O valor β é uma indicação da inclinação da curva que representa a relação entre a resistência e a temperatura do sensor de temperatura medidos sobre condições de energia zero, ou seja, sem haver tensões envolvidas. Quanto maior for este valor maior será a variação da resistência por grau celsius. Neste caso particular o valor de β é 3977.

$$\frac{R_{th}}{R_0} = e^{\beta \left(\frac{1}{T+273,15} - \frac{1}{T_0+273,15} \right)} \quad (4)$$

Resolvendo a equação 4 em relação a T, é possível obter o valor da temperatura em que a NTC está inserida.

O fluxograma desenvolvido para a aquisição dos dados e processamento matemático encontra-se ilustrado na Figura 38. Nesta figura podemos visualizar que a temperatura é medida 256 vezes, para em seguida ser efetuada a média dos valores medidos. Depois é realizado o procedimento anteriormente mencionado tornando possível a obtenção do valor da temperatura. Este procedimento é referente à etapa de aquisição da temperatura que foi demonstrada na Figura 32.



Figura 38 - Fluxograma do ciclo para aquisição da temperatura

4.4.5. GERAÇÃO DAS MENSAGENS DE DADOS

Nesta fase, efetuada a aquisição da temperatura e possuindo já o valor da mesma é necessário enviar essa informação para o *transceiver* ANT para que este a encaminhe para

os outros nós da rede. Para o efeito é necessário gerar uma mensagem com a correta parametrização dos seus campos, que consistem:

- *SYNC* - Campo de sincronismo. Este campo possui um valor único para comunicações assíncronas. Este valor é 0xA4.
- *MSG_LENGTH* – A mensagem de envio de dados utilizando o tipo *broadcast* possui sempre oito *bytes* do campo de dados. Assim, este campo possui o valor 0x08.
- *MSG_ID* – Este campo identifica o tipo de mensagem que vai ser enviada. Como está mensagem diz respeito ao envio de uma mensagem utilizando o modo *broadcast*, este campo possui o valor de 0x4E.
- *DATA* – Este campo corresponde aos dados propriamente ditos. Nele vai contido o valor da temperatura medida.
- *CHECKSUM* – Este campo corresponde à operação lógica XOR de todos os campos anteriores. Neste caso, como o valor da temperatura pode variar, este campo não possui um valor fixo.

A mensagem formulada com os campos específicos para a realização desta configuração pode ser visualizada na Figura 39:

Designação	SYNC	MSG_LENGTH	MSG_ID	CHANNEL_NUM	DATA	CHECKSUM
Valor	0xA4	0x01	0x4B	0x00	TEMPERATURA	0xEE

Figura 39 - Mensagem para envio de dados através do modo *broadcast*

Sempre que seja necessário comunicar com o *transceiver* todas mensagens têm de ter este formato pois de outra forma o *transceiver* não reconhecerá as mensagens e descartará as mesmas.

Na Figura 40 pode-se visualizar o fluxograma responsável por este procedimento. Em primeiro lugar o valor da temperatura anteriormente adquirido é guardado num vetor. Em seguida, é realizada a formulação da mensagem, ou seja, são adicionados todos os campos necessários para que os dados sejam corretamente recebidos pelo transceiver ANT. Após

este procedimento todo o sistema é colocado em modo de suspensão, dando início ao processo demonstrado na Figura 32.



Figura 40 -Fluxograma para o processamento de dados

Demonstrado o funcionamento deste nó, segue-se a demonstração dos outros dois nós da rede.

4.5. NÓ 3 E NÓ 4 - DEFINIDOS COMO ESCRAVOS NA REDE

Após adquirir a temperatura, o nó mestre envia essa informação para a rede onde os nós escravos, com as conexões já efetuadas, aguardam a receção dessa informação. Esta informação só possui utilidade prática se chegar a alguém que a possa interpretar e analisar e tomar decisões mediante a informação que observou. Neste sentido, foram desenvolvidas duas interfaces gráficas para que essa informação possa ser visualizada por alguém que a possa interpretar e efetuar as ações necessárias.

Para desenvolver uma interface gráfica, em primeiro lugar, é necessário conectar o *transceiver* ANT à placa *USB* disponibilizada no kit de desenvolvimento que permite o

interface entre o *transceiver* ANT e o computador. Na Figura 41 encontra-se ilustrada a placa em questão.

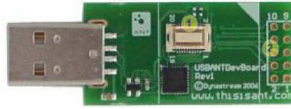


Figura 41 -Placa de interface do *transceiver* ANT para *USB*

Quando este procedimento é efetuado, o sistema operativo do computador instala os controladores necessários para o correto funcionamento deste dispositivo. Após este passo procedeu-se ao desenvolvimento das interfaces gráficas, mas previamente é necessário incluir no projeto as bibliotecas dinâmicas (*dll*) fornecidas pelo fabricante para que a interface desenvolvida consiga interagir com o módulo conectado ao computador. Estas bibliotecas são:

- *ANT_NET.dll*
- *ANT_WrappedLib.dll*
- *DSI_CP210xManufacturing_3_1.dll*
- *DSI_SiUSBXp_3_1.dll*

Em seguida, é necessário incluir também a biblioteca *ANT_Managed_Library* que possui todas as funções suportadas pelo protocolo ANT.

Efetuados estes procedimentos, procede-se ao desenvolvimento da interface propriamente dito.

Em primeiro lugar é concebido todo o aspeto visual da interface gráfica, desde caixas de texto para apresentação das diversas informações até termómetros desenvolvidos para uma rápida visualização e interpretação da temperatura.

Em seguida é necessário começar a interação com o dispositivo conectado ao computador e que possibilita a interligação deste à rede. Assim sendo, é concebido um descritor para o dispositivo *USB* conectado ao computador, que lhe atribui uma etiqueta e procede a deteção automática do *baud rate* utilizado por esse dispositivo. Se este descritor não for

criado porque nenhum dispositivo se encontra conectado ao computador, a interface gráfica apresenta uma mensagem de erro a indicar precisamente esse facto. É também criada uma função que processa os acontecimentos que ocorrem nos canais ativos no *transceiver*, desde receção de mensagens a erros ocorridos nas transmissões.

Em seguida realiza-se a configuração do *transceiver*, seguindo as mesmas diretrizes utilizadas quando foi utilizado o microcontrolador. Em primeiro lugar é também aconselhável proceder ao *reset* do *transceiver*. Após a realização deste, procede-se a configuração do mesmo. Neste caso, foi efetuada uma configuração mais extensiva do *transceiver*, no sentido em que foram configuradas características que não eram necessárias para o correto funcionamento do mesmo pois foram utilizados na mesma os valores por defeito do protocolo ANT, no entanto, a título demonstrativo de como tal poderia ser realizado foram efetuados. Estas características dizem respeito à configuração de rede e de frequência de funcionamento.

Assim, em primeiro lugar foi configurada a rede em que este *transceiver* está inserido, neste caso a rede é pública. Em seguida, o canal 0 foi inicializado como escravo e como pertencente à rede pública. Depois procedeu-se a configuração do identificador de canal. Sendo este um *transceiver* escravo, os valores constituintes deste campo devem ser iguais aos valores presentes no *transceiver* mestre, para que, quando o *transceiver* escravo inicie a procura de um *transceiver* mestre para estabelecer uma comunicação, encontre o dispositivo desejado e não um *transceiver* errado. Posteriormente procedeu-se, neste caso desnecessariamente, à configuração da frequência de funcionamento, que neste caso possui o valor por defeito de 2,4 Ghz. Finalmente, procedeu-se à abertura do canal.

Se todo este procedimento correu como desejado o *transceiver* ANT encontra-se neste momento a comunicar com a rede. Em caso contrario, a etapa que não foi corretamente configurada é apresentada numa janela de texto especificamente desenvolvida para o efeito, assim como uma mensagem de erro numa nova janela com a informação de que não foi possível abrir o canal em questão. Este procedimento foi realizado novamente mas para o canal 1, desta forma os canais 0 e 1 foram abertos

No fluxograma da Figura 42 encontra-se ilustrado o procedimento mencionado.



Figura 42 - Fluxograma da configuração do transceiver escravo

Após este procedimento este nó encontra-se neste momento ativo na rede e a receber mensagens dos nós mestres. No entanto, para ser possível visualiza-las é necessário criar uma função que processe os acontecimentos ocorridos no canal aberto. Para tal, foi criada a função *ChannelResponse*. Como mencionado, esta função processa todos os acontecimentos que ocorrem no canal, entre eles encontram-se os acontecimentos de transferências efetuadas sem sucesso, transferência efetuadas com sucesso, tempo de procura expirado, receção de dados falhada, receção de dados efetuada com sucesso, entre outros que podem ser visualizados no fluxograma da Figura 43. Assim sendo, quando é gerado um acontecimento no canal, é identificado qual o acontecimento que ocorreu. Quando esse acontecimento consiste na receção de uma mensagem do tipo *broadcast*, é

adquirido o seu campo de dados e dessa forma é possível ter acesso à informação que foi transmitida.

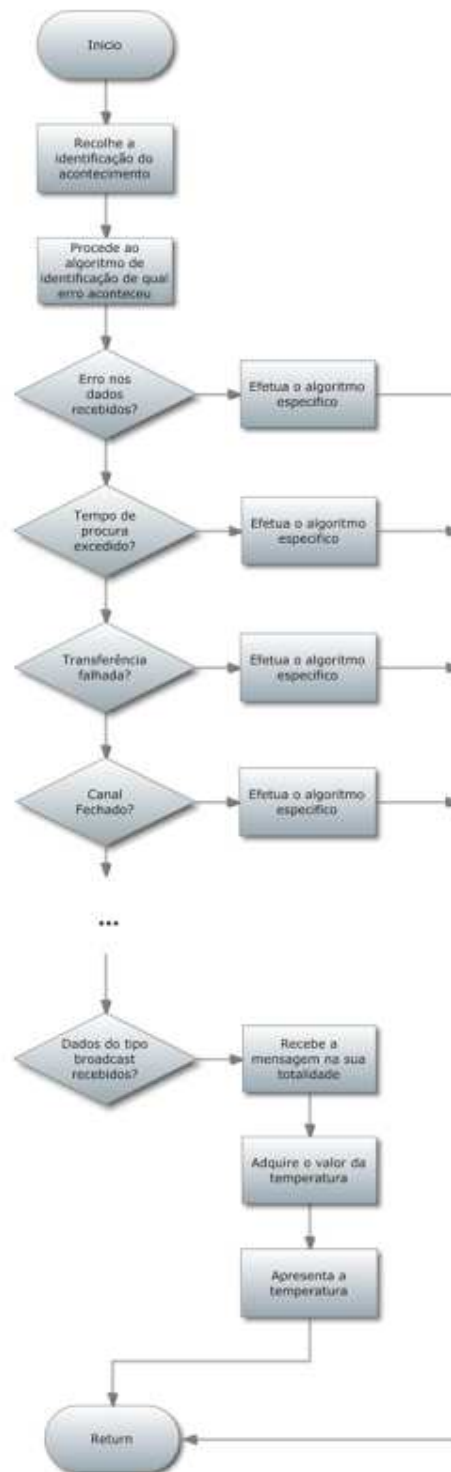


Figura 43 - Fluxograma da função que processa os acontecimentos no canal

Os dois interfaces gráficos desenvolvidos para visualização da informação nas estações de monitorização foram executados da mesma forma. No entanto existem algumas diferenças

entre ambos. Uma delas, à semelhança do que se passou com o nó mestre, consiste nas configurações dos canais que têm de possuir números de identificação de canal diferentes, pois trata-se de dois nós diferentes.

Outra diferença entre as interfaces gráficas consiste no registo da temperatura num ficheiro de texto para uma possível análise na estação de monitorização supervisora. Por último, na estação de monitorização é demonstrado um gráfico demonstrativo da temperatura medida a cada dez minutos. Assim sendo, a interface gráfica desenvolvida para a estação supervisora do nó 3 da rede, tem o aspeto ilustrado na Figura 44.

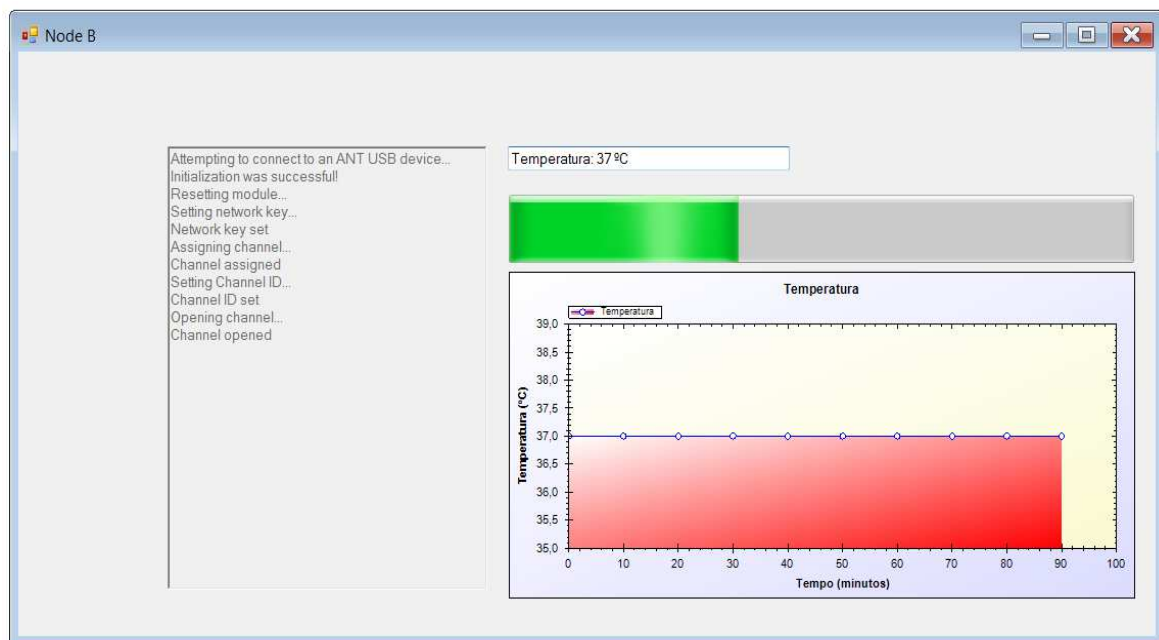


Figura 44 -Interface gráfica da estação de monitorização supervisora

Como podemos visualizar na Figura 44, na caixa de texto do lado esquerdo da interface gráfica é possível visualizar as mensagens de erro e as etapas de configuração do *transceiver*. Em seguida, do lado direito, pode-se visualizar um caixa de texto com a representação numérica da temperatura medida, seguida de uma representação da mesma sob a forma de barra oferecendo desta forma uma monitorização mais visual e por último, pode-se visualizar um gráfico com o comportamento da temperatura ao longo do tempo para efetuar uma possível análise. Esta estação pretende simular um local de trabalho de um profissional de saúde onde este pode a qualquer momento analisar a temperatura que

está a ser medida. Como mencionado anteriormente, esta estação também procede ao registo da temperatura num ficheiro de texto onde fica registado a data, a hora e o valor de temperatura, para arquivo da mesma ou qualquer outra utilidade que este possa ter. Este ficheiro encontra-se ilustrado na Figura 45.

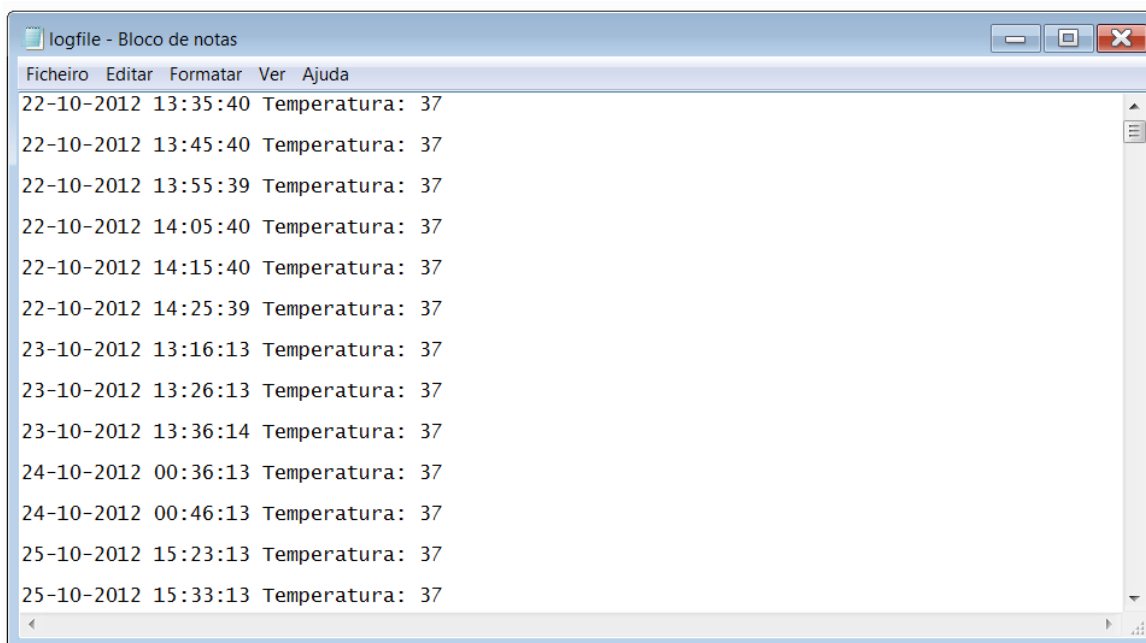


Figura 45 -Ficheiro de registo da temperatura

A interface gráfica da outra estação supervisora, a do nó 4 da rede, pretende ter um aspeto mais simples, e neste sentido apresenta a temperatura sob a forma de um termómetro. O desenvolvimento desta interface pretende simular uma estação de monitorização utilizada pelo paciente para que este possua também conhecimento do estado da sua temperatura corporal. Na Figura 46 encontra-se ilustrada a interface desenvolvida para o efeito.

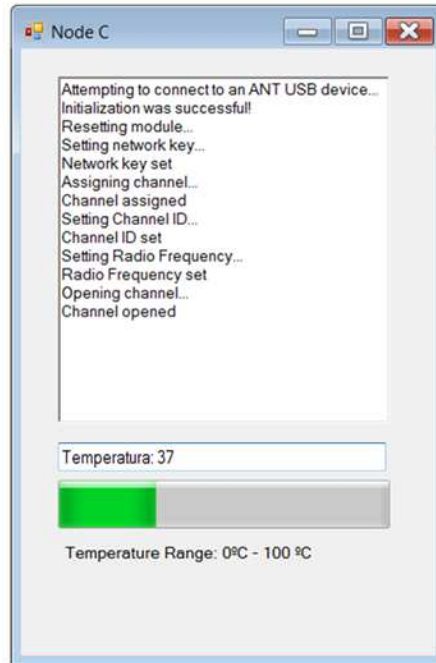


Figura 46 -Interface gráfica da estação de monitorização local

Como podemos visualizar na Figura 46, a interface gráfica é mais simples que a do nó 3. Nesta interface gráfica podemos visualizar as mensagens de erro e as etapas de configuração do *transceiver*. Apesar de estas mensagens não possuírem muita relevância para o utilizador do sistema, esta informação é também disponibilizada pois em caso de avaria ou erro de configurações é possível visualizar essa informação nesta caixa de texto.

Depois, é apresentada a temperatura numa forma mais visual sob a forma de termómetro mas também é apresentada sob a forma numérica.

4.6. COMUNICAÇÃO ENTRE OS NÓS CONSTITUINTES DA REDE

Nesta secção serão apresentadas de forma mais evidente as configurações efetuadas nos canais dos diversos *transceivers* de maneira a que estes fossem capazes de criar uma rede de sensores sem fios. Serão apresentados de forma clara os identificadores de canal configurados nos diversos canais de forma a torna possível a formação da rede.

Na Figura 47 encontram-se ilustradas quais as conexões que estão a ser abordadas nesta secção.

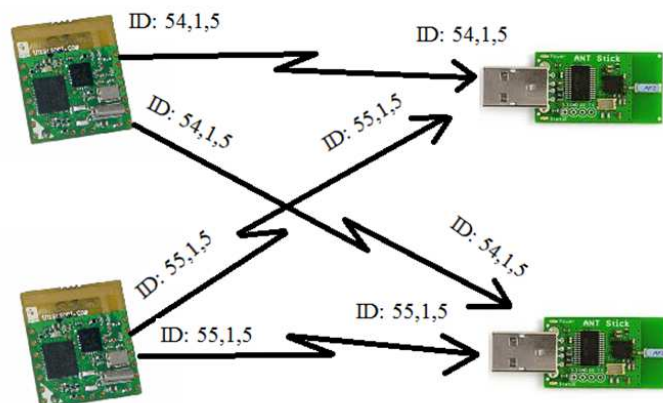


Figura 47 - Comunicação entre os nós constituintes da rede

Durante a descrição do desenvolvimento do projeto foram demonstrados os valores escolhidos para configurar os diferentes nós de rede, no entanto, sentiu-se a necessidade de sumarizar essa informação para melhor compreensão da mesma. Assim sendo, na Tabela 8 encontram-se as configurações efetuadas no *transceiver* mestre.

Tabela 8 – Configuração dos Nós Mestres

Canal 0	Canal 1
Rede Nº. 0 – pública	Rede Nº. 0 – pública
Canal 0 – Canal Transmissor Bidirecional	Canal 1 – Canal Transmissor Bidirecional
Número de Dispositivo - 54	Número de Dispositivo - 55
Tipo de Transmissão – 5	Tipo de Transmissão – 5
Tipo de Dispositivo - 1	Tipo de Dispositivo - 1
Tipo de dados - <i>Broadcast</i>	Tipo de dados - <i>Broadcast</i>

Como podemos observar, os dois canais possuem configurações idênticas, o que faz sentido pois ambos pertencem à mesma rede. A diferença entre as configurações reside na diferença do número de dispositivo que, como mencionado anteriormente, necessita de ser diferente para que seja possível a distinção dos diferentes *transceivers* que constituem a rede. Na Tabela 9 encontra-se demonstrada a configuração efetuada nos nós escravos da rede.

Tabela 9 – Configuração dos Nós Escravos

Canal 0	Canal 1
Rede Nº. 0 – pública	Rede Nº. 0 – pública
Canal 0 – Canal Recetor Bidirecional	Canal 1 – Canal Recetor Bidirecional
Número de Dispositivo - 54	Número de Dispositivo - 55
Tipo de Transmissão – 5	Tipo de Transmissão – 5
Tipo de Dispositivo - 1	Tipo de Dispositivo - 1
Tipo de dados - <i>Broadcast</i>	Tipo de dados - <i>Broadcast</i>

Como pode ser observado na tabela 9, a configuração é idêntica à efetuada no *transceiver* mestre, pois desta forma é garantida que este *transceiver* escravo apenas vai estabelecer uma conexão com um *transceiver* mestre que possua um identificador de canal igual ao seu e assegura que o *transceiver* mestre escolhido é o correto. A diferença entre as configurações reside apenas no fato deste canal ter sido configurado como canal recetor bidirecional e não como canal transmissor bidirecional.

Ao longo do desenvolvimento do projeto foram efetuados pequenos testes que permitiram compreender e observar o comportamento do sistema. Os testes consistiram na medição da temperatura corporal de uma pessoa, em dois pontos distintos do corpo. A temperatura obtida foi de 37 °C, o que permite concluir que o sistema de aquisição de temperatura e os procedimentos matemáticos para a obtenção da mesma estão corretos. Em seguida foi montada a rede descrita, esta consistia em dois computadores pessoais com as interfaces gráficas e com a placa *USB* e *transceiver* ANT conectadas aos mesmos, correspondendo aos dois nós escravos, e pelos nós mestres que efetuavam a medição da temperatura. Quando esta foi totalmente montada, foi possível observar o valor da temperatura corporal medida em ambas interfaces gráficas desenvolvidas e presentes nos computadores distintos, assim como a criação do ficheiro de registo no computador onde a interface gráfica supervisora funcionava, no qual foi possível observar o histórico da temperatura, o que permite concluir que todo o desenvolvimento efetuado traduziu-se num sucesso.

5. CONCLUSÕES

Este projeto abordou um tema bastante interessante, dado que a área das comunicações tem sido um dos sectores com maior desenvolvimento nos últimos anos, especialmente a área das comunicações sem fios, sendo que a maior foco de desenvolvimento foi na indústria das telecomunicações onde os fabricantes sentiram a necessidade de integrar nos seus produtos mecanismos de transferirem dados sem a necessidade de cabos. No entanto, estas tecnologias foram adotadas por outros setores e nos dias de hoje estão presentes em diversos produtos e soluções.

Posto isto, em primeiro lugar foi efetuado um estudo desta nova tecnologia, das suas especificações e como os *transceivers* interagem entre si, assim como com o controlador. Em termos funcionais, foi dada especial atenção no desenvolvimento e parametrização da rede, desenvolvimento de um sistema de aquisição e registo da temperatura, configuração e operação do *transceiver*, e no desenvolvimento de interfaces gráficas para poder visualizar os dados recebidos e validar os resultados e objetivos do projeto. Em seguida, foram desenvolvidas as comunicações entre os controladores e os *transceivers* ANT, quer no caso onde foi utilizado microcontrolador PIC quer no caso onde foi utilizado o computador pessoal e o módulo USB, o que permitiu uma correta comunicação entre os nós da rede, que funcionaram de forma transparente para o utilizador, o que era um dos objetivos deste trabalho. Por último, os testes realizados permitem concluir que todo o projeto foi bem-

sucedido na medida em que o valor da temperatura corporal foi de encontro ao esperado e transmitida pela rede, conforme pretendido.

Pela experiência que adquiri na realização deste projeto, esta não é uma tecnologia de fácil compreensão numa primeira fase e pode ser apelidada com o termo “caixa negra”, pois não fazia ideia do que se tratava nem sequer como funcionava. Aliado a esta situação, o fato de existir muito pouca informação disponível sobre este novo protocolo, talvez por ser um protocolo proprietário, e a informação existente ser bastante direta e condensada dificultou bastante a compreensão e a introdução neste protocolo. No entanto, numa fase final do projeto e após todo este trabalho, posso concluir que é uma tecnologia bastante interessante que torna bastante fácil a implementação de uma rede, ou seja, depois de saber configurar um nó de rede em todas as suas vertentes, quer na comunicação entre nós quer na comunicação entre microcontrolador e *transceiver*, todos os outros são implementados da mesma forma. Desta forma, é possível retirar e inserir nós da rede, necessitando apenas de efetuar algumas configurações para este entrar em funcionamento, pois este não é um protocolo que permita uma implementação do tipo “*plug and play*”. Assim sendo, a complexidade deste projeto está na compreensão deste protocolo proprietário e de que forma é possível desenvolver um sistema que comunique com estes *transceivers* e que estes por sua vez comuniquem entre si de forma correta para formarem uma rede de sensores sem fios com inúmeras aplicabilidades. Depois de ultrapassados estes desafios, a complexidade deste projeto está relacionada com a complexidade da rede a ser implementada e com o condicionamento de sinal efetuado.

Finalmente, a solução encontrada e o desenvolvimento efetuado foi de encontro aos objetivos pressupostos para este projeto e por esse motivo, este traduziu-se num sucesso.

Como não poderia ser de outra forma, existem várias abordagens diferentes que podem ser implementadas em desenvolvimentos futuros. Por exemplo, a utilização de um *smartphone* que já possua o protocolo ANT embebido, desenvolvendo uma aplicação móvel ou a utilização de mais nós e criar uma rede mais complexa com diversos sensores a transmitir dados para uma unidade central, dependendo claro da capacidade financeira para a aquisição de kits de desenvolvimento ou dos *transceivers* individuais.

Referências Documentais

- [1] Kohvakka, Mikko; Hannikainen, Marko; Hamalainen, Timo — *Ultra Low Energy Wireless Temperature Sensor Network Implementation*. IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications. 2005
- [2] Shan, Qingshan Brown, David, Liu Ying — *Wireless Temperature Sensor Network for Refrigerated Vehicles*. IEEE ISIE Dubrovnik Croatia. 2005
- [3] Narmada, Alaparathi; Sudhakara Rao, Parvataneni — *ZigBee Based WSN with IP Connectivity*. Fourth International Conference on Computational Intelligence, Modelling and Simulation. 2012
- [4] Veerasingam, Sridevi; Karodi, Saurabh; Shukla, Sapna; Yeleti, Mehar — *Design of Wireless Sensor Network node on ZigBee for Temperature Monitoring*. International Conference on Advances in Computing, Control, and Telecommunications Technologies. 2009
- [5] Shan, Qingshan; Brown, David; Liu, Ying — *Wireless Temperature Sensor Network for Refrigerated Vehicles*. IEEE members
- [6] Ebenezer, Jemimah, D. Baghyalakshmi, S.A.V. Satyamurty — *WSN based Temperature monitoring for High Performance Computing Cluster*. IEEE – international Conference on Recent Trends in Information Technology, ICRTIT. 2011
- [7] Misal, Chaitanya S. — *Analysis of power consumption of an end device in a ZigBee mesh network*. Thesis submitted to the faculty of the University of North Carolina at Charlotte. 2007
- [8] Honkanen, M; Lappetelainen; Kivekas, Kalle. — *Low end extension for Bluetooth*. Nokia Research Center, Tampere Finland. 2004
- [9] Kamath, Sandeep; Lindh, Joakim — *Measuring Bluetooth Low Energy Power Consumption*. TEXAS INSTRUMENTS. 2012
- [10] Callaway, Ed — *Low Power Consumption Features of the IEEE 802.15.4/ZigBee LR-WPAN Standard*. Florida Communication Research Lab, Motorola. 2003
- [11] This is ANT — *ANT Message Protocol and Usage*. 2012
- [12] This is ANT — *Interfacing with ANT General purpose Chipsets and Modules*. 2012
- [13] Donovan, John — *Bluetooth Goes Ultra Low-Power*. DIGIKEY
- [14] SCHNEIDER ELECTRIC — *ZigBee – Setting Standards for Energy-Efficient Control Network*. Schneider Electric
- [15] Thonet, Gilles; Allard-Jacquín, Patrick; Colle, Pierre — *ZigBee – Wifi Coexistence*. Schneider Electric Innovation Department White Paper and Test Report. 2008
- [16] This is ANT — *ANT Q&A's*. 2008

- [17] Ergen, Sinem— *ZigBee/IEEE 802.15.4 Summary*. Berkeley University of California.2004
- [18] Stevanovic, Dusan — *Zigbee / IEEE 802.15.4 Standard*. ZigBee Alliance. 2007
- [19] Sylla, Iboun — *To ZigBee or Not to ZigBee? Factors to consider when selecting ZigBee Technology*. TEXAS INSTRUMENTS.2009
- [20] Woodings, Ryan Winfield and Gerrior, Mark— *Avoiding Interference in the 2.4-GHz ISM Band*. CYPRESS Semiconductor Corp.
- [21] Kinney, Patrick — *ZigBee Technology: Wireless Control that Simply Works*.Chair of IEEE 802.15.4 Task Group. 2003
- [22] NOKIA Research Center — *Wibree forum merges with Bluetooth SIG*. Nokia.
- [23] Bluetooth SIG — *Bluetooth Low Energy Technology-Technical Info*. Bluetooth SIG.2012
- [24] Wikipedia, the free encyclopedia - *IEEE 802.15*
- [25] This is ANT - *How ANT Compares*. This is ANT.2
- [26] This is ANT – *ULP – Advantages ANT*.
- [27] Giovino, Bill — *New ATMEL Microcontrollers Target Low-Power*.ATMEL.2006
- [28] White Paper— *RF Power Options in ZigBee Solutions*. Cirronet Inc.2005
- [29] Reynolds, Melanie — *Wibree becomes ULP Bluetooth*. ElectronicsWeekly.2007
- [30] Bluetooth SIG — *Bluetooth Low Energy Technology FAQ*
- [31] Macdonald, Brian — *2008 ANT+ Symposium – The Opportunities*. This is ANT.2008
- [32] Nikkei Electronics Asia — *Building a Pratical Wireless Sensor Network*.Nikkei Electronics Asia.2007
- [33] Morris, Rod — *ANT vs ZigBee*
- [34] ZigBee Alliance — *ZigBee and Wireless Frequency Coexistences*. Zigbee Alliance White Paper.2007
- [35] Bluetooth SIG — *Communications Topologies*. Bluetooth SIG.2012
- [36] Nokia — *Bluetooth Protocol*. Nokia.
- [37] Smith, Phil — *Compariong Low-Power*. Wireless Technologies. DIGIKEY
- [38] Hodgdon, Charles — *Adaptative Frequency Hopping for Reduced interference between Bluetooth and Wireless LAN*. Ericsson Technology Licensing.2003

Outras Referências Documentais

- [1] Salatas, Vlasios— *Object Tracking Using Wireless Sensor Networks*. Naval Postgraduate School, Monterey California. 2005
- [2] Mottola, Luca — *Programming Wireless Sensor Networks: From Physical to Logical Neighborhoods*. Politecnico Di Milano.2008
- [3] This is ANT — *AT3 RF Transceiver Module Datasheet*. 2012
- [4] This is ANT — *ANTAP1MxIB RF Transceiver Module*. 2012
- [5] This is ANT — *AP2 RF Transceiver Module Datasheet*. 2012
- [6] This is ANT — *C7 RF Transceiver Module Datasheet*. 2012
- [7] MICROCHIP TECHNOLOGY INC.—*PIC18F2585/2680/4585/4680 Data sheet:28/40/44-Pin, High- Temperatura, High-Performance Microcontrollers with ECAN™, 10- Bit A/D ando nano Watt Technology*, 2009.
- [8] MICROCHIP TECHNOLOGY INC. — *Understanding A/D Converter Performance Specifications*. 2000
- [9] MICROCHIP TECHNOLOGY INC. — *Interfacing the TC72 SPI™ Digital Temperature Sensor to a PICmicro^R Microcontroller*, 2004
- [10] MICROCHIP TECHNOLOGY INC. — *Interfacing a MCP9700 Analog output Temperature Sensor to a PICmicro^R Microcontoller*,2005
- [11] MICROCHIP TECHNOLOGY INC. — *Temperature Sensor Design Guide – Temperature Measurement Solutions for Silicon IC Temperature Sensor, Thermocouple, RTD and Thermistor-Based Applications*,2010
- [12] MICROCHIP TECHNOLOGY INC. — *Temperature Sensing Technologies*, 1998
- [13] NORDIC SEMICONDUCTOR — *nRF24AP2 – Single-Chip ANT™ ultra-low power wireless network solutions, Product Specification v1.2*. 2012
- [14] TEXAS INSTRUMENTS — *1-and 8- Channel ANT™ RF Network Processors*. 2010
- [15] ZigBee Comparison — http://www.stg.com/wireless/ZigBee_comp.html. This is ANT.2009
- [16] ZigBee Alliance — *ZigBee and Wireless Frequency Coexistances - ZigBee White Paper*.
- [17] MICROCHIP TECHNOLOGY INC. - *Microchip MiWi P2P Wireless Protocol*.
- [18] James, Baldwin— *Know More Knowledge Of Bluetooth*
- [19] BBC NEWS — *Bluetooth rival unveiled by Nokia*
- [20] TGDAILY — *Bluetooth 4 specs reduces energy consumption by 90 %*
- [21] Wikipedia, the free encyclopedia — *Bluetooth Low Energy*

- [22] Guerin, R.;Rank, J.;Sarkar,S.;Vergetis.E — *Forming Connected Topologies in Bluetooth Ad-hoc Networks – An Algorithmic Perspective*. Department of Electrical and Systems Engineering, University of Pennsylvania.
- [23] Libelium— Smartphones Detection AFH. Libelium