

Dispositivo Eletrónico para Aquisição e Processamento de Múltiplos Sinais Fisiológicos e Ambientais

Relatório de projeto apresentado para a obtenção do grau de Mestre em
Instrumentação Biomédica

Autor

Ivo Micael Couceiro Brandão

Orientadores

Prof. Doutor José Pedro M. N. Amaro

(Instituto Superior de Engenharia de Coimbra
Departamento de Engenharia Eletrotécnica)

Prof. Doutor Fernando J. T. E. Ferreira

(Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Departamento de Engenharia Eletrotécnica e de Computadores)

Instituto Politécnico de Coimbra
Instituto Superior de Engenharia de Coimbra

Coimbra, dezembro, 2015

Agradecimentos

É com muita satisfação que expresso aqui o mais profundo agradecimento a todos aqueles que tornaram a realização deste trabalho possível. Gostaria de agradecer aos professores orientadores desta tese, José Amaro e Fernando J. T. E. Ferreira, pelo apoio, incentivo e disponibilidade demonstrada em todas as fases que levaram à concretização deste trabalho.

I. Resumo

Os sistemas de saúde atuais estão estruturados e otimizados para reagir a crises e para a gestão de doenças, no entanto enfrentam atualmente um novo desafio, a crescente população idosa (Carrilho, 2015). Este facto sugere que a prestação de cuidados de saúde requer uma mudança significativa na direção de soluções mais flexíveis e adaptáveis às necessidades emergentes. A reestruturação dos sistemas de cuidados de saúde poderá permitir, uma mudança de foco para uma gestão mais pró-ativa do bem-estar, em vez da gestão da doença. Deste modo, poderá ser possível a intervenção de profissionais na prevenção e deteção precoce das patologias. Os sistemas portáteis para a monitorização contínua de saúde poderão ser, portanto, uma tecnologia chave para ajudar na transição para cuidados de saúde mais preventivos, mais acessíveis por contraponto com os cuidados de saúde curativos e tendencialmente mais caros.

Este projeto discute a implementação e descrição da arquitetura de um protótipo de uma rede de sensores para monitorização de sinais fisiológicos e ambientais com base em componentes comerciais. Os nós da rede são compatíveis com redes de sensores sem fio *Bluetooth 4.0 Low Energy*. Este trabalho, apresenta ainda a arquitetura do sistema em termos de *hardware* bem como a arquitetura de *firmware* e *software* implementada. É apresentada também uma possível solução de *software* para visualização dos dados adquiridos. Foi igualmente realizada uma introdução ao processo de certificação e qualificação, de um dispositivo com a tecnologia *Bluetooth*. Este processo de certificação é um conjunto de procedimentos necessário para permitir colocar no mercado qualquer dispositivo baseado neste protocolo.

Palavras Chave: ARM, Bluetooth Low Energy 4.0, STM32F4, STM32, AFE, Aquisição de sinal

II. Abstract

The current health systems are structured and optimized to respond to crises and for managing diseases, but now face a new challenge, the growing elderly population (Carrilho, 2015). This suggests that the provision of health care requires a significant shift towards more flexible and adaptable solutions to emerging needs. The restructuring of health care systems may allow a change of focus to a more proactive management of well-being, rather than the management of the disease. Thus, it may be possible for professionals to have intervention of in the prevention and early detection of diseases. Portable systems for continuous health monitoring may therefore be a key technology to help in the transition to more preventive health care more accessible by contrast with curative care that tend to be more expensive.

This project discusses the implementation and description of the architecture of a prototype of a sensor network for monitoring physiological and environmental signals based on commercial components. The network nodes are compatible with wireless sensor networks based in the Bluetooth 4.0 Low Energy protocol. This work also shows the system architecture in terms of hardware and firmware architecture and implemented software. It is also presented a possible software solution for visualization of the acquired data. An introduction to the certification process and qualification of a device with Bluetooth technology has also been done. This certification process is a set of procedures necessary to allow placing on the market any device based on this protocol.

Keywords: ARM, Bluetooth Low Energy 4.0, STM32F4, STM32, AFE, signal Acquisition

Índice

1. INTRODUÇÃO	3
1.1. Enquadramento do Tema.....	3
1.2. Objetivos	4
1.3. Estrutura do Relatório de Projeto	4
2. ESTADO DA ARTE	5
2.1. Tecnologia Bluetooth Low Energy 4.0	6
2.2. Tecnologia ZigBee	6
2.3. Tecnologia 6LoWPAN.....	8
2.4. Ciclo de Desenvolvimento dos Processadores ARM.....	9
2.5. Comparação com Dispositivos Similares.....	11
2.5.1. Plataforma Bitalino.....	11
3. ELEMENTOS DAS ARQUITETURA DO SISTEMA	13
3.1. Tecnologia Bluetooth Low Energy 4.0	13
3.1.1. Topologia de Redes Bluetooth	15
3.1.2. Camadas de Arquitetura de Sistema.....	16
3.2. Camada de Abstração de Hardware STM32Cube	21
3.3. Meios de Comunicação em Sistemas Embebidos	25
3.3.1. Universal Synchronous Asynchronous Receiver Transmitter	25
3.3.2. Inter-Integrated Circuit	27
3.3.3. Serial Peripheral Interface	29
3.4. Sinais e Grandezas.....	32
3.4.1. Sinais Bio Potenciais	32
3.4.2. Sinais e Grandezas ambientais	38
4. DESENVOLVIMENTO DE HARDWARE	39
4.1. Sensores.....	40
4.1.1. Sensor de Temperatura	40
4.1.2. Sensor de Pressão Atmosférica	42
4.1.3. Sensor de Humidade Relativa.....	44
4.1.3. Oxímetro de Pulso	46
4.1.4. Aquisição de Sinais Bio Potenciais	48
4.2. Microcontrolador.....	55
4.3. Módulos de comunicação Bluetooth 4.0	59
5. DESENVOLVIMENTO DE FIRMWARE	61
5.1. Inicialização do Projeto e Periféricos	61
5.2. Arquitetura.....	64
6. DESENVOLVIMENTO DO SOFTWARE DE APOIO.....	75
6.1. Arquitetura.....	75
6.2. Funcionalidades.....	77
7. RESULTADOS.....	79
7.1. Aquisições	79

8. CERTIFICAÇÃO E QUALIFICAÇÃO DE DISPOSITIVOS DE RÁDIO

FREQUÊNCIA.....	82
8.1. Processo de Qualificação de um dispositivo Bluetooth	82
8.2. Processo de Certificação de FCC	83
8.3. Processo de Certificação CE.....	85
8.4. Processo de Certificação IC.....	85

9. CONCLUSÃO E TRABALHO FUTURO 87

BIBLIOGRAFIA.....	89
--------------------------	-----------

ANEXO I – ESQUEMA DO MODULO DE AQUISIÇÃO DE SINAIS BIO

POTENCIAIS (AFE)	94
-------------------------------	-----------

ANEXO II – BILL OF MATERIALS AFE.....	96
--	-----------

ANEXO III – PCB MODULO DE AQUISIÇÃO DE SINAIS BIO – POTENCIAIS (TOP LAYER)	97
---	-----------

ANEXO IV – PCB MODULO DE AQUISIÇÃO DE SINAIS BIO POTENCIAIS (BOTTOM LAYER).....	98
--	-----------

ANEXO V – FUNÇÃO DE CÁLCULO CRC 8 BITS.....	99
--	-----------

ANEXO VI – FUNÇÃO DE CÁLCULO FRAME CHECK SEQUENCE.....	100
---	------------

III. Índice de Figuras

Figura 2.1 - Arquitetura da <i>Stack</i> ZigBee (ZigBee Alliance Organization, n.d.).....	7
Figura 2.2 - Exemplo da estrutura interna de um dispositivo <i>ZigBee</i>	8
Figura 2.3 – Ilustração do modelo OSI das camadas protocolares do 6LoWPAN (Olsson, n.d.)	9
Figura 2.4 – Ilustração da plataforma Bitalino Board kit (Bitalino, 2013).....	11
Figura 3.1 – Esquema geral da arquitetura do protótipo	13
Figura 3.2 – Evolução da especificação do Bluetooth(S.L, 2014)	14
Figura 3.3 - Arquiteturas de <i>firmware</i> dos diferentes tipos de dispositivos <i>Bluetooth</i> (Sig, 2010).	15
Figura 3.4 - Exemplo de uma <i>Scatternet Bluetooth</i> (Har-Shai et al., 2002).....	15
Figura 3.5 - Comunicação entre <i>Host</i> e <i>Controller</i> (Sig, 2010).	16
Figura 3.6 – Arquitetura de <i>firmware</i> de um dispositivo <i>Bluetooth Low Energy</i> (Nordic Semiconductor, 2014).....	17
Figura 3.7 - Bluetooth Low Energy Channels (Argenox, 2015)	18
Figura 3.8 - Estrutura utilizada pelo GATT	21
Figura 3.9 – Arquitetura <i>Cortex Microcontroller Standard Interface</i> (ARM, 2013).....	22
Figura 3.10 - Arquitetura STM32Cube (ST Microelectronics, 2015a)	23
Figura 3.11 – Interligação entre os vários níveis do <i>STM32Cube</i> (ST Microelectronics, 2015a)	24
Figura 3.12 – Interface serie entre dois dispositivos	27
Figura 3.13 – Exemplo da interface com barramento I2C	28
Figura 3.14 - Sequência completa de transferência I2C(NXP Semiconductors, 2004)	28
Figura 3.15 – Exemplo de uma interface SPI.....	30
Figura 3.16 – Interface barramento SPI múltiplos escravos.....	31
Figura 3.17 – Interface barramento SPI com múltiplos escravos, Daisy Chain.....	32
Figura 3.18 - Amostras dos sinais, (a) ECG; (b) EEG (c) EMG(Luca & G.De, 2000).....	33
Figura 3.19 - ECG de 12 derivações é adquirido usando os elétrodos braço direito (RA), braço esquerdo (LA), perna esquerda (LL), referência perna direita (RL), e seis no peito (C).(Luca & G.De, 2000)	34

Figura 3.20 - Colocação dos elétrodos sobre os bíceps e tríceps com uma referência independente para aquisição de um EMG(Luca & G.De, 2000)	35
Figura 3.21 – Colocação dos elétrodos do sistema de 10-20 derivações utilizando as orelhas como referência(Luca & G.De, 2000)	35
Figura 3.22 – Absorção da luz infravermelha e Vermelha no sangue oxigenado versus sangue desoxigenado. (Texas Instruments, 2014b).....	37
Figura 4.1 – Arquitetura da plataforma de hardware, dispositivo periférico apenas com sensores	39
Figura 4.2 - Arquitetura da plataforma de hardware, para aquisição de sinais Bio potenciais	40
Figura 4.3 - Placa de desenvolvimento TMP102.....	40
Figura 4.4 - Sequência de transições I2C, TMP102	41
Figura 4.5 - Placa de desenvolvimento BMP180.....	42
Figura 4.6 – Sequência de leitura dos parâmetros de calibração do sensor BMP180.....	42
Figura 4.7 – Leitura da temperatura não regulada do sensor BMP180.....	43
Figura 4.8 – Leitura da pressão atmosférica não regulada.....	44
Figura 4.9 - Placa de desenvolvimento HTU21D	44
Figura 4.10 – Sequência de transmissões I2C, leitura da temperatura HTU21D	45
Figura 4.11 – Sequência de transmissões I2C, leitura da humidade relativa HTU21D.....	46
Figura 4.12 - Oxímetro de pulso NellCor DS-100A.....	46
Figura 4.13 - Sinais de controlo do Oxímetro de pulso	47
Figura 4.14 - Esquema de ligação oxímetro de pulso	47
Figura 4.15 - Diagrama de Blocos internos ADS1299 (Texas Instruments, 2013).....	48
Figura 4.16 - Modos de operação das entradas analógicas ADS1299(Texas Instruments, 2013)	50
Figura 4.17 - Modos de operação das entradas analógicas (Texas Instruments, 2013).....	50
Figura 4.18 - Esquema de alimentação, para a operação com alimentação unipolar (Texas Instruments, 2013)	51
Figura 4.19 - Esquema de alimentação, para a operação com alimentação bipolar (Texas Instruments, 2013)	51
Figura 4.20 - Circuito de seleção de alimentação Unipolar ou Bipolar.....	52
Figura 4.21 - Esquema do princípio de operação do dispositivo TPS60403 (Texas Instruments, 2015)	53

Figura 4.22 - TPS60403 Circuito implementado	53
Figura 4.23 - Referência Interna, ADS1299 (Texas Instrumments, 2013).....	53
Figura 4.24 – Representação 3D da PCB de aquisição de bio potenciais	55
Figura 4.25 - Família de microcontroladores STM32 (ST Microelectronics, n.d.-b)	56
Figura 4.26 - Placa de desenvolvimento STM32F401RE Núcleo.....	57
Figura 4.27 – Características STM32F401RE.....	57
Figura 4.28 - Configuração de pinos utilizada	58
Figura 4.29 - Troca de modos de consumo de energia no microprocessador.	59
Figura 4.30 - Modulo HM-10 Bluetooth	60
Figura 4.31 - Modulo Bluetooth para o dispositivo central.....	60
Figura 5.1 - Interface Gráfica STM32F4CubeMX.....	61
Figura 5.2 - Configuração do <i>oscilador</i> para o processador STM32F401RE.....	62
Figura 5.3 - Ambiente de Desenvolvimento Integrado Eclipse.....	63
Figura 5.4 - Configuração da FPU	64
Figura 5.5 - Arquitetura de <i>firmware</i> para o STM32F401RE.....	65
Figura 5.6 - Estrutura de identificação dos sensores e dos seus canais.....	66
Figura 5.7 - Pacote serial Geral	69
Figura 5.8 - Estrutura <i>MT Command</i>	69
Figura 5.9 - Estrutura da mensagem <i>Configure Notification</i>	70
Figura 5.10 - Estrutura da mensagem <i>Sensor Data Notification</i>	70
Figura 5.11 - Estrutura da mensagem <i>Discover</i>	71
Figura 5.12 - Diagrama de sequência de mensagens entre o PC e o protótipo	71
Figura 5.13 - Estrutura da identificação dos <i>MT Commands</i>	72
Figura 5.14 - Estrutura geral de uma fila FIFO	72
Figura 5.15 – Fila Circular	73
Figura 5.16 - Estrutura da fila para guardar amostras dos sensores	74
Figura 6.1 - Software desenvolvido.....	75
Figura 6.2 - Camadas de Software implementadas	75
Figura 6.3 - Barra de estado	76
Figura 6.4 – Esquema simplificado produtor múltiplos consumidores.....	76

Figura 6.5 – Estrutura final da camada HCI/BLE para envio/recepção de comandos Bluetooth	77
Figura 6.6 - UI de configuração das aquisições dos sensores	78
Figura 7.1 - Aquisição da temperatura ambiente (Período de 24horas).....	79
Figura 7.2 - Aquisição da Humidade Relativa (Período de 24horas)	80
Figura 7.3 - Aquisições da Pressão Atmosférica (Período de 24 horas).....	80
Figura 7.4 - Sinal obtido do Oxímetro de pulso.....	81
Figura 8.1 - Processo de colocação de um novo produto Bluetooth no mercado.....	83
Figura 8.2 - Etiqueta/marcação FCC-ID	84

IV. Índice de Quadros

Tabela 2.1 – Características da plataforma Bitalino Vs protótipo desenvolvido	11
Tabela 3.1 – Estrutura de dados utilizado na transmissão serie	26
Tabela 3.2 – Modos de Operação I2C (NXP Semiconductors, 2007).....	27
Tabela 3.3 – Modos de operação SPI (Motorola Inc, 2003)	31
Tabela 3.4 – Propriedades dos sinais bio potenciais (Luca & G.De, 2000)	33
Tabela 4.1 - Endereços sensor de temperatura TMP102 (Texas Instruments, n.d.-b)	41
Tabela 4.2 - Configuração da referência do <i>clock</i> ADS1299	54
Tabela 5.1 - Identificação do tipo de canal e das suas unidades	68
Tabela 5.2 - Identificação das mensagens MT	69

V. Simbologia e Abreviaturas

API - Application Programing Interface
AFE - Analog Front End
ARM – Advanced RISC Machine
BLE - Bluetooth Low Energy
BSIG - Bluetooth Special Interest Group
BSP - Board Suport Package
BQE - Bluetooth Qualification Experts
BOM - Bill of Materials
BQID - Qualified Design Identification
CSS - Core Specification supplement
CMSIS - Cortex Microcontroller Software Interface Standard
DSP - Digital Signal Processing
DoC - Declaração de conformidade
FPU – Floating Point Unit
GPIO - General Purpose Input Output
GPO – General Purpose Poutput
HCI - Host Controller Interface
HSI - High Speed Internal Oscilator
HSE - High Speed External Oscilator
HAL - Hardware Abstraction Layer
ISM - Industrial, Sientific and Medical
I2C – Inter-integrated Circuit
LSB - Least Significant Bit
MSB - Most Significant Bit
MEMS - Micro-Electro-Mechanical Systems
RTOS - Real Time Operating Systems
SPI - Serial Peripheral Interface
STM – ST Microelectronics
TDMA - Time Division Multiple Access
TCB - Telecommunications Certification Body
USART - Universal Synchronous Asynchronous Receiver Transmitter
WPAN - Wireless Personal Area Network
WSN - Wireless Sensor Network

1. Introdução

Este projeto discute a implementação e descrição da arquitetura de um protótipo de uma rede de sensores para monitorização de sinais fisiológicos e ambientais com base em componentes comerciais. Os nós da rede são compatíveis com redes de sensores sem fio *Bluetooth 4.0 Low Energy*. Este trabalho, apresenta ainda a arquitetura do sistema em termos de *hardware* bem como a arquitetura de *firmware* e *software* implementada. É apresentada também uma possível solução de *software* para visualização dos dados adquiridos. Além de todo o processo de desenvolvimento foi ainda realizado uma pequena investigação sobre o processo de certificação e qualificação, de um dispositivo com a tecnologia *Bluetooth*, de modo a que este possa ser colocado no mercado. Neste capítulo é feito o enquadramento do tema, são definidos os objetivos e descrita a estrutura do documento.

1.1. Enquadramento do Tema

Os sistemas de saúde atuais que estão estruturados e otimizados para reagir a crises e para a gestão de doenças, enfrentam atualmente um desafio com a crescente população de pessoas idosas (Carrilho, 2015). Este facto sugere que os cuidados de saúde requerem uma mudança no sentido de tornar a sua estrutura mais escalável, recorrendo a soluções mais acessíveis quer económica quer socialmente. Esta reestruturação dos sistemas de cuidados de saúde poderá permitir, uma mudança de foco para uma gestão mais pró-ativa do bem-estar, em vez da gestão da clássica gestão da doença. Esta mudança de foco poderá ajudar na prevenção e deteção precoce de doenças. Os sistemas portáteis para a monitorização contínua de saúde são uma tecnologia-chave para ajudar a auxiliar esta transição na metodologia de prestação de cuidados de saúde. Os sistemas de monitorização de parâmetros de saúde (tecnologias *wearable*) permitem a um indivíduo acompanhar de perto as mudanças nos seus sinais vitais e fisiológicos. É deste modo possível fornecer informações a um utilizador que permitam ajudar a manter o seu estado de saúde num ponto de equilíbrio considerado ideal. Através do uso de um sistema de telemedicina integrado, estes sistemas podem alertar o pessoal médico quando ocorrem mudanças com risco de vida. Além disso, os pacientes podem beneficiar de um acompanhamento a longo prazo, como parte de um procedimento de diagnóstico, pode igualmente ser possível atingir a manutenção ideal de uma condição crónica. Estes sistemas poderão também ser uteis, durante a recuperação de um evento, por exemplo, de um procedimento cirúrgico. A vigilância da saúde a longo prazo pode capturar as variações diurnas e circadianas em sinais fisiológicos. Estas variações são indicadores muito bons, por exemplo, na recuperação dos pacientes cardíacos após o infarto do miocárdio (Binkley, 2003). Além disso, o acompanhamento a longo prazo pode confirmar a adesão às diretrizes de tratamento (por exemplo, exercícios cardiovasculares regulares) ou ajudar a monitorização dos efeitos da terapia com recursos a medicamentos. Outros pacientes também podem se beneficiar desses

sistemas, por exemplo, podem ser utilizados durante a reabilitação física, acidentes vasculares cerebrais, ou reabilitação de outro trauma cerebral.

Outra área onde a monitorização constante dos parâmetros fisiológicos e ambientais pode beneficiar é a área desportiva. A atividade física tem um impacto positivo sobre o bem-estar das pessoas, e contribui para a diminuição da ocorrência de doenças crónicas (Ermes, Pärkkä, Mäntyjärvi, & Korhonen, 2008). O reconhecimento da atividade, com sensores portáteis pode fornecer feedback ao utilizador sobre o seu estilo de vida em relação à atividade física e aos desportos, e assim, promover um estilo de vida mais ativo. Até agora, o reconhecimento atividade físico tem sido quase sempre estudado em laboratórios específicos.

Os recentes avanços tecnológicos nas tecnologias MEMS (MEMS - *Micro-Electro-Mechanical Systems*) (mems-exchange, 2010). permitem a criação de novos sensores e circuitos integrados de baixo consumo energético. Estes avanços existem também nas redes de comunicação sem fio permitindo a miniaturização e o baixo consumo energético dos nós de sensores inteligentes que podem ser utilizados para monitorização fisiológica e ambiental (Milenković, Otto, & Jovanov, 2006). Estes nós são capazes de realizar medições, processamento e comunicação de um ou mais parâmetros fisiológicos ou ambientais e podem ser integrados em redes pessoais sem fio WPAN (*Wireless Personal Area Network*) para monitorização da saúde e do ambiente circundante. Através do uso destas redes de sensores será possível realizar uma monitorização ambulatória da saúde de forma pouco dispendiosa, não-invasiva e contínua.

1.2. Objetivos

Este projeto tem como objetivo o desenvolvimento de um sistema de aquisição de dados provenientes de sensores, capazes de medir grandezas ambientais e grandezas fisiológicas. Foi selecionado o *hardware* a utilizar, bem como realizado o desenvolvimento necessário sobre as plataformas de *hardware* para suportar a medição dos sinais pretendidos. As funcionalidades pretendidas são as seguintes:

- Eletrocardiograma (ECG);
- Eletromiograma (EMG);
- Oxímetro de pulso (SpO₂);
- Acelerómetro, Giroscópio, Magnetómetro;
- Temperatura ambiente, pressão atmosférica e humidade relativa;

Foi selecionada uma tecnologia WPAN para a realização das comunicações, e também realizada a estrutura do *firmware* de forma a garantir a correta aquisição dos sinais pretendidos.

1.3. Estrutura do Relatório de Projeto

Para além do capítulo de Introdução, esta dissertação contém mais 7 capítulos. No capítulo 2, é descrito o estado da arte mediante uma introdução às tecnologias Wireless PAN, e aos processadores ARM. No capítulo 3 são descritos os elementos de arquitetura de sistema, onde

são descritas as noções básicas, necessárias para a realização deste trabalho. São descritas as camadas de protocolo da *stack Bluetooth* 4.0 LE, da *stack STM32F4Cube* e da arquitetura *Cortex Microcontroller Software Interface Standard* (CMSIS). Neste capítulo são ainda descritos os protocolos e barramentos mais comuns em sistemas embebidos. São igualmente identificadas as grandezas a adquirir. No capítulo 4, é apresentado o desenvolvimento de hardware efetuado no decurso deste trabalho. É igualmente identificado e descrito o conjunto de plataformas de *hardware* utilizadas. No capítulo 5, é descrita a arquitetura de *firmware* desenvolvida, bem como os principais módulos dessa componente de *software*. O capítulo 6, apresenta a arquitetura do *software* desenvolvido para computador que permite implementar a comunicação, configuração e aquisição das grandezas físicas geradas pelos diversos sensores e enviadas pelo dispositivo. No capítulo 7 são apresentados os resultados obtidos dos testes realizados ao projeto. O capítulo 8 descreve os processos de certificação e qualificação de dispositivos de rádio frequência, bem como o processo de qualificação de um dispositivo *Bluetooth* 4.0 LE. No capítulo 9 é tecida um conjunto de considerações que consubstanciam as conclusões obtidas no trabalho apresentado. Neste capítulo é sugerido um conjunto de desenvolvimentos que constituem uma proposta de trabalho futuro.

2. Estado da Arte

Esta seção descreve o estado de arte dos vários elementos necessários para a arquitetura de sistema. Desta forma foi possível efetuar uma seleção dos vários tipos de tecnologias necessárias à realização deste projeto.

A utilização de redes de sensores sem fios é atualmente uma realidade em numerosos projetos industriais e/ou científicos. “Atualmente a utilização de redes sem fios encontra-se em constante expansão, inclusive as WPAN. O baixo custo e consumo energético destas, tornam-nas ideais para a sua utilização em redes de sensores e de pequenos dispositivos alimentados por baterias...” (Macedo, 2010). Estas tecnologias encontram aplicação em contexto industrial, bem como na automação de edifícios, na monitorização ambiental bem como na vigilância de parâmetros de saúde (*healthcare*). A generalização destas tecnologias torna interessante o seu estudo. Ao longo deste trabalho foi possível identificar um conjunto de tecnologias disponíveis para redes WPAN. Nesta secção é realizada uma introdução às tecnologias consideradas mais relevantes para a utilização do trabalho proposto. Foi efetuado também um resumo do ciclo de desenvolvimento dos processadores ARM, desde a sua criação até aos dias de hoje. A utilização destes processadores é corrente em muitas aplicações de sistemas embebidos. Foi selecionado um processador ARM da família *Cortex-M4* para o desenvolvimento deste projeto.

Após o estudo das várias tecnologias WPAN foi selecionada a tecnologia *Bluetooth Low Energy 4.0* para implementar neste projeto, em detrimento das tecnologias *ZigBee* e *6LoWPAN* devido ao facto de que esta tecnologia já se encontra disponível na maior parte dos dispositivos, sendo por isso considerada uma tecnologia *legacy* de acordo com a terminologia anglo-saxónica. Esta escolha deve-se ainda ao facto de o *Bluetooth Low Energy* ser otimizador para redes de sensores de baixo consumo energético. As tecnologias *ZigBee* e *6LoWPAN*, ainda não são consideradas tecnologias *legacy* e, portanto, seria necessário a sua integração em todos os dispositivos.

2.1. Tecnologia Bluetooth Low Energy 4.0

O *Bluetooth Low Energy* (BLE) é uma tecnologia desenvolvida pelo *Bluetooth SIG* (*Bluetooth Special Interest Group, SIG*) para comunicações de curta distância. Em contraste com as versões de *Bluetooth* anteriores, o advento do BLE ocorreu em resposta a um conjunto de tecnologias que podem ser consideradas como similares ou com objetivos semelhantes. Ao longo deste trabalho foi possível identificar os protocolos *ZigBee* e 6LoWPAN, como sendo os principais concorrentes do BLE, apesar de serem especialmente apontados a aplicações que requerem múltiplos saltos de rede (*multihop*) (Gomez & Paradells, 2010)(Ludovici, Calveras, & Casademont, 2011). O BLE constitui uma solução especialmente vocacionada para aplicações em que os nós de rede de origem e destino de dados se encontram a apenas um salto de comunicação rádio (*singlehop*). O BLE é assim especialmente dedicado a aplicações da eletrónica de consumo, monitorização de aplicações médicas, soluções gestão de energia inteligente (*Smart Energy*) e segurança. O facto de a tecnologia *Bluetooth* ser praticamente omnipresente nos diversos dispositivos móveis e de, portanto ser considerada como uma tecnologia *legacy* poderá proporcionar a rápida adoção do BLE, uma vez que a implementação deste último pode aproveitar as semelhanças com o *Bluetooth* clássico. É esperado que o uso do BLE seja utilizado em milhões de dispositivos num futuro próximo (West, 2013).

Em 2013, o IETF 6LoWPAN WG (IETF 6LoWPAN WG - *Internet Engineering Task Force 6LoWPAN Working Group*), reconheceu a importância da tecnologia BLE para a internet das coisas (IOT - *Internet of things*), e começou a desenvolver uma especificação para a transmissão de pacotes de dados IPv6 através de BLE.(Nieminen, Savolainen, Isomaki, & Shelby, 2013). Atualmente a especificação *Bluetooth* encontra-se na versão 4.2, tendo sido esta especificação adotada a 2 de dezembro de 2014. A 14 de julho de 2014 foi lançado um suplemento CSS (CSS - *Core Specification supplement*) para esta versão, no entanto o *hardware* e as *stacks* de *firmware* para esta versão apenas deverão ser disponibilizados após de janeiro de 2016.

2.2. Tecnologia ZigBee

A tecnologia *ZigBee* foi criada pela *ZigBee Alliance* (*ZigBee Alliance Organization*, n.d.) e descreve um protocolo de comunicações sem fios de baixo consumo, baixo custo, bidirecional e de curto alcance. A Figura 2.1, ilustra a arquitetura da *stack ZigBee*, que inclui um número de componentes distribuídos sobre camadas de acordo com o modelo OSI de interligações para software aberto. As camadas inferiores deste protocolo são a camada de controlo de acesso ao meio, (MAC - *Medium Access Control*) e a camada física (PHY - *Physical Layer*). Estas camadas são definidas pela norma IEEE 802.15.4 – 2003. A especificação do *ZigBee* define apenas duas outras camadas, que consistem na camada de rede *ZigBee* (NWK - *ZigBee Network*) e na camada de aplicação (APL – *Application layer*).(*ZigBee Standard Organization*, 2012)

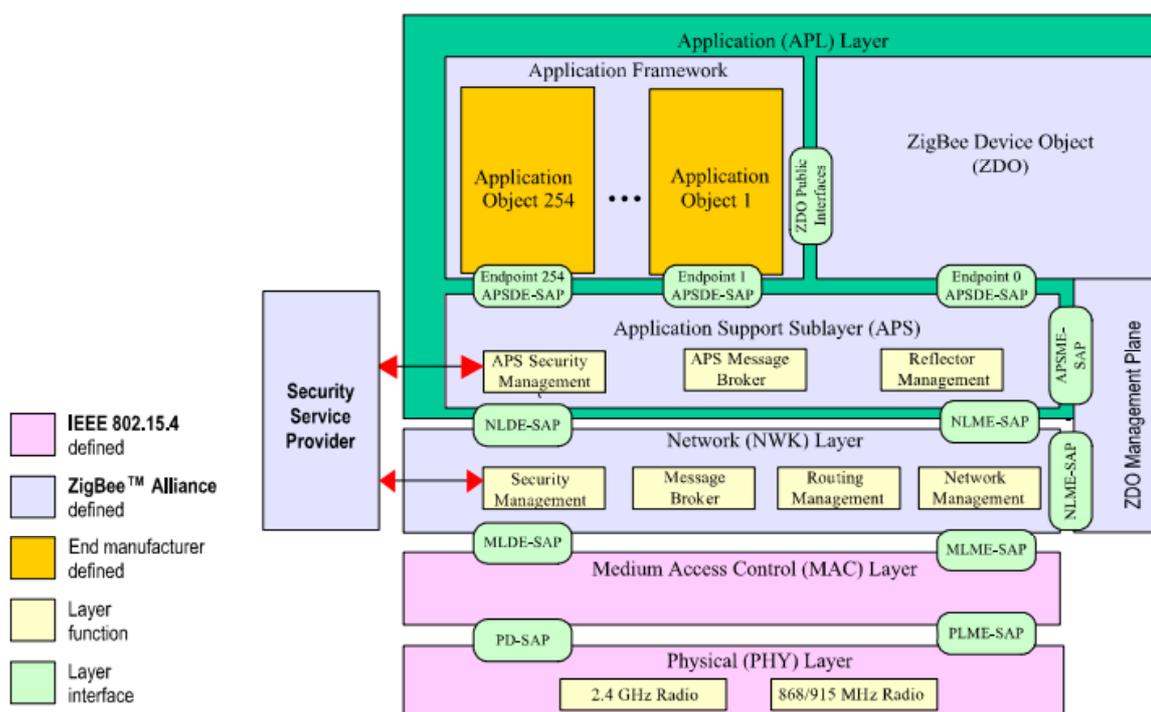


Figura 2.1 - Arquitetura da *Stack* ZigBee (ZigBee Alliance Organization, n.d.)

A camada APL é a camada que implementa a aplicação do dispositivo e contém três subcamadas, a subcamada de suporte à aplicação (APS – *Application Support Sublayer*), é responsável pela gestão da segurança e por realizar a interligação da camada APL com a camada imediatamente inferior, a camada NWK. A subcamada ZDO (*ZigBee Device Object*), é responsável pela configuração do dispositivo *ZigBee* e a subcamada AF (*Application Framework*) fornece os meios de configuração dos elementos lógicos de aplicação do protocolo (*Endpoints*). Cada aplicação *ZigBee* permite a configuração de 240 *EndPoints*, onde cada *Endpoint* define uma entidade de comunicação dentro do dispositivo, através do qual é implementada uma determinada função. O *Endpoint 0* está reservado para a aplicação ZDO que permite enviar comandos de controlo e gestão para o dispositivo. Um *Endpoint* é definido por um conjunto de entradas e saídas de dados, denominados *Clusters*. A *ZigBee Alliance* garante interoperabilidade de determinadas aplicações dentro de uma área específica através de perfis de aplicação (*Application Profiles*). Estes perfis de aplicação descrevem uma coleção de dispositivos que são utilizados numa aplicação específica e implicitamente descrevem o esquema de mensagens entre esses dispositivos. Nestes perfis de aplicação são descritos os *Clusters* de entrada e saída que cada aplicação deve ter por forma a garantir a interoperabilidade. Cada *Cluster* define ainda um conjunto de atributos (*Attributes*) que podem conter dados ou estados necessários à aplicação. A Figura 2.2, representa de forma esquemática, um exemplo da estrutura interna de um dispositivo *ZigBee*.

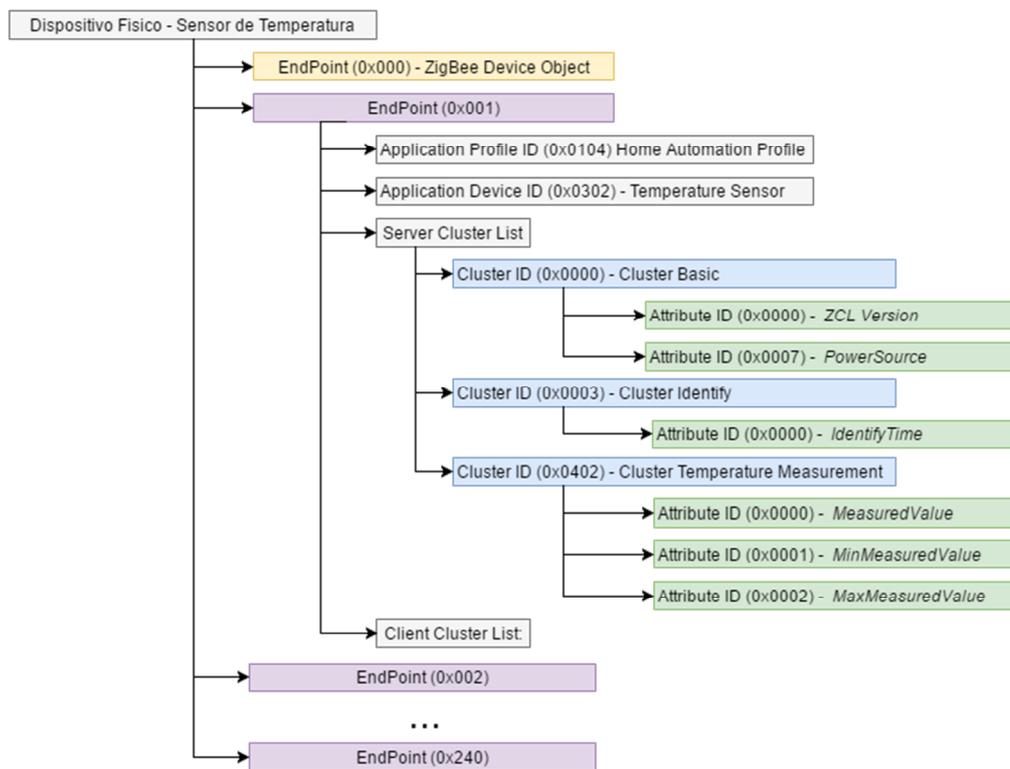


Figura 2.2 - Exemplo da estrutura interna de um dispositivo *ZigBee*

Para que seja possível criar uma rede fiável de vários dispositivos, estão definidos três tipos de dispositivos *ZigBee*:

- *ZigBee Coordinator*;
- *ZigBee Router*;
- *ZigBee End Device*;

O *ZigBee Coordinator* é responsável por criar e gerir a rede *ZigBee*, o *ZigBee Router* permite fazer a interligação entre dispositivos que se encontrem fora do alcance de comunicação entre si, e o *ZigBee End Device* é um dispositivo terminal da rede, não desempenhando qualquer função necessária à manutenção desta, normalmente é um sensor ou um atuador.

2.3. Tecnologia 6LoWPAN

A tecnologia 6LoWPAN está a ser desenvolvida pela IETF 6LoWPAN WG (*Internet Engineering Task Force 6LoWPAN Working Group*) (Olsson, n.d.) , e implementa uma rede sem fios WPAN que utiliza o protocolo de internet versão 6 (IPv6 - *Internet Protocol Version 6*) para endereçamento dos nós da rede.

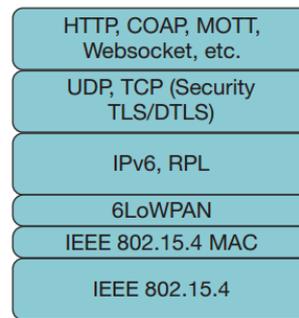


Figura 2.3 – Ilustração do modelo OSI das camadas protocolares do 6LoWPAN (Olsson, n.d.)

Apesar não ter sido ainda publicada a versão final desta especificação foi possível identificar um conjunto de implementações da sua arquitetura. O funcionamento da rede é garantido pela norma IEEE 802.15.4 – 2003, que como nas redes *ZigBee*, define as camadas de base dos dispositivos que formam a rede. O IEEE 802.15.4 – 2003 define três bandas de funcionamento da rede em 868MHz, 915Mhz e 2.4GHz, e permite atingir velocidades de transferência de dados até 250Kb\s. Uma vez que os pacotes de dados utilizados em redes IPv6 ocupam muito espaço, para serem enviados através de redes IEEE 802.15.4 (IEEE, 2008), é utilizada uma camada imediatamente acima da camada de controlo de acesso ao meio (MAC), a camada 6LoWPAN, que faz a compressão do cabeçalho dos pacotes IPv6 e fragmenta os pacotes IPv6 em vários pacotes com tamanhos suportados pelas redes IEEE 802.15.4. A utilização deste endereçamento permite ligar as redes 6LoWAPN a outras redes que também utilizem o endereçamento IPv6, como por exemplo, WLAN (WLAN – *Wireless Local Area Network*) ou LAN.(LAN – *Local Area Network*).

2.4. Ciclo de Desenvolvimento dos Processadores ARM

Os processadores ARM têm origem em Inglaterra em 1984. A sua designação original era *Acorn RISC Machine*, sendo mais tarde alterada para *Advanced RISC Machine*. Estes processadores foram o produto da equipa de desenvolvimento da *Acorn Computer*. A Acorn foi um dos principais nomes no mercado britânico de computadores pessoais (Ray, O’Dea Chad, Hughes Bob, Schmertz Taneika, 2012). O primeiro produto desta empresa foi o computador doméstico britânico, o *Atom*, com um processador de 1 MHz, 12kbytes de ROM e RAM. A partir daqui a empresa passou a trabalhar com a *British Broadcasting Corporation* (BBC) para produzir e comercializar este computador de uso pessoal. O produto resultante, o micro BBC, tornou-se um sucesso logo no seu lançamento em 1982. No entanto, outros computadores como, por exemplo, o *Apple Lisa*, que combinou o primeiro ambiente gráfico para PC com um processador de 16 bits, mostravam um aumento de desempenho considerável em relação às máquinas de 8 bits existentes. Como resultado direto, foi criada uma secção de investigação e desenvolvimento avançado da *Acorn*, para tentar desenvolver um projeto especial de forma a criar um processador com um conjunto de instruções reduzido (RISC). O resultado deste projeto de investigação foi o primeiro processador ARM, o *Acorn RISC Machine*. As primeiras amostras foram entregues em 1985, foram fabricadas com uma tecnologia VLSI a 3µm. A

experiência adquirida ao desenvolver o ARM1, mostrou que existiam algumas áreas onde o conjunto de instruções poderia ser melhorado de forma a maximizar a performance dos sistemas neles baseados. Foram adicionadas algumas facilidades no que toca ao processamento de sinais em tempo real, bem como uma interface de coprocessador, a qual permitia acelerar o cálculo de números de vírgula flutuante. Em 1987, um computador pessoal, o *Archimedes*, foi lançado como o primeiro computador a utilizar o ARM2 de 8MHz. Depois do lançamento do *Archimedes*, a *Acorn* continuou a desenvolver o processador lançando novas versões com melhor performance. Em 1989 foi lançado o ARM3 que melhorou significativamente a frequência de oscilador para 25MHz. Logo após o desenvolvimento do ARM3 a *Acorn* desenvolveu uma variante de baixo consumo energético com uma lista de características que tornaram os processadores ARM mais atrativos para os utilizadores e programadores com interesse no desenho de dispositivos portáteis de baixo custo, bem como dispositivos de comunicação. O interesse na família ARM estava a crescer cada vez mais á medida que os engenheiros prestavam mais atenção a esta arquitetura RISC. A *Acorn RISC Machines Ltd* passou a ser a *Advanced RISC Machine Ltd*, com a missão de promover o desenvolvimento dos processadores ARM e facilitar o seu uso pelos engenheiros deste tipo de sistemas embebidos. O primeiro desenvolvimento da ARM Ltd foi o processador ARM6, que é um processador de 32-bit e constitui o passo seguinte depois do ARM3. A ARM Ltd, continuou a evolução do processador ARM tendo já lançado no mercado o ARM7, o ARM8, o *Strong ARM*, o ARM9, ARM10 e atualmente está na versão ARM11. (LTD, n.d.), (Markus Levy, n.d.)

Ao longo dos últimos seis a sete anos uma das principais tendências no design de sistemas que utilizem microcontroladores é a adoção dos ARM7 e ARM9 como o CPU para microcontrolador de uso geral. Hoje em dia existem mais de 240 microcontroladores baseados em ARM disponíveis a partir de uma grande variedade de fabricantes (Trevor, 2009).

A família ARM *Cortex* é uma nova geração de microprocessadores que fornecem uma arquitetura padrão para uma ampla gama de requisitos técnicos. Ao contrário de outros processadores ARM, a família ARM Cortex é composta por três perfis principais:

- ARM-Cortex A – Este perfil é otimizado para processadores com sistemas operativos complexos e com aplicações de alto desempenho.
- ARM-Cortex R – Este perfil é otimizado para sistemas de tempo real.
- ARM-Cortex M – Este perfil é otimizado para microcontroladores, e aplicações de baixo custo e baixo consumo energético.

Na designação do tipo de cada processador é acrescentado um número (ex. ARM *Cortex* M4) indicando o nível de performance do processador, atualmente a família ARM Cortex M contem os seguintes níveis de performance:

- ARM Cortex M0
 - ARM Cortex M0+
 - ARM Cortex M1
-

- ARM Cortex M3
- ARM Cortex M4
- ARM Cortex M7

O perfil de performance selecionado para o desenvolvimento deste projeto foi o ARM Cortex M4, que proporciona um bom equilíbrio entre performance e baixo consumo energético.

2.5. Comparação com Dispositivos Similares

Nesta secção é apresentada uma comparação entre o protótipo desenvolvido e plataformas semelhantes, existentes no mercado. Na pesquisa realizada por plataformas semelhantes ao protótipo desenvolvido, apenas foi encontrada uma plataforma semelhante, a plataforma Bitalino.

2.5.1. Plataforma Bitalino

A plataforma Bitalino (Bitalino, 2013) é uma plataforma, desenvolvida por uma equipa portuguesa, e é utilizada principalmente como plataforma de desenvolvimento e aprendizagem. A plataforma consiste num conjunto de módulos, representado na Figura 2.4, tendo a capacidade de medir sinais Bio potenciais e de os enviar através de uma interface *Bluetooth* para um dispositivo remoto.

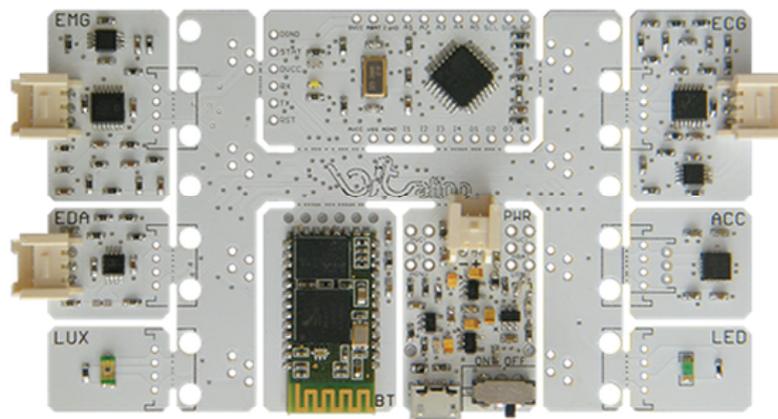


Figura 2.4 – Ilustração da plataforma Bitalino Board kit (Bitalino, 2013)

As características da plataforma Bitalino e do protótipo desenvolvido estão detalhadas na Tabela 2.1.

Tabela 2.1 – Características da plataforma Bitalino Vs protótipo desenvolvido

	Características Plataforma Bitalino		Protótipo
Microcontrolador	<ul style="list-style-type: none"> • Core: AVR328 Arduino • Clock: 16 MHz • SRAM: 1Kb 	Microcontrolador	<ul style="list-style-type: none"> • Core: STM32F401RE, ARM Cortex M4 32-bit • Clock: 84MHz • SRAM: 96Kb

	<ul style="list-style-type: none"> • EEPROM: 2Kb 		<ul style="list-style-type: none"> • Flash: 512Kb
eletromiografia (EMG)	<ul style="list-style-type: none"> • Número de Eléttodos: 3 • CMRR: - 110dB • Impedância de Entrada: 1000Gohm 3pF • Limites:0 - 3.3mV • Ganho: 1000 • Largura de Banda: 10 – 400Hz • Resolução: 10 bits 	<p>Eletromiograma (EMG)</p> <p>Eletrocardiograma (ECG)</p> <p>Encefalograma (EEG)</p> <p>SPO2</p>	<ul style="list-style-type: none"> • Módulo de aquisições bio - potenciais configurável • Número de Eléttodos: 8 (possibilidade de expandir até 32 com 4 módulos de aquisição) • CMRR: - 110dB • Ganho: programável de 1, 2,4,6,8. 24 • Resolução: 24 bits
Resposta galvânica da pele (EDA)	<ul style="list-style-type: none"> • Número de Eléttodos: 2 • Largura de Banda: 0 – 3Hz • Limites: 1Mohm • Resolução: 10 bits 	Temperatura	<ul style="list-style-type: none"> • Resolução: 12 bits • Limite de operação: -25 e + 85 °C
Luz	360-970nm	Humidade Relativa	<ul style="list-style-type: none"> • Resolução: 14 bits • Precisão: 2%
Eletrocardiograma (ECG)	<ul style="list-style-type: none"> • Número de Eléttodos: 2 • CMRR: 110dB • Impedância de Entrada: 100Gohm 3pF • Limites:0 - 3mV • Ganho: 1100 • Largura de Banda: 0.5 – 40Hz • Resolução: 10 bits 	Pressão Atmosférica	<p>Resolução: 12bits</p> <p>Precisão: 0,02 hPa</p>
Acelerómetro	<ul style="list-style-type: none"> • Largura de Banda: 0- 50Hz • Limites: +/- 3g 	IMU	Giroscópio Acelerómetro e bussola.
Comunicação	Bluetooth 2.0	Comunicação	Bluetooth 2.0/4.0 LE, UART
Tamanho	100x60mm	Tamanho	60mmx60mm

3. Elementos das Arquitetura do Sistema

Neste capítulo pretende-se fazer uma introdução aos fundamentos teóricos necessários para a realização deste projeto e para a criação da arquitetura do sistema, ilustrada na Figura 3.1. Deste modo é feita uma introdução à tecnologia *Bluetooth Low Energy*, bem como uma introdução à *stack* de *firmware* do processador ARM, que foi utilizada no desenvolvimento da aplicação de *firmware*.

São também introduzidos os fundamentos teóricos dos protocolos de comunicação entre dispositivos de sistemas embutidos, que foram utilizados neste projeto. Por fim, é feita uma introdução aos tipos de sinais que se pretende adquirir.

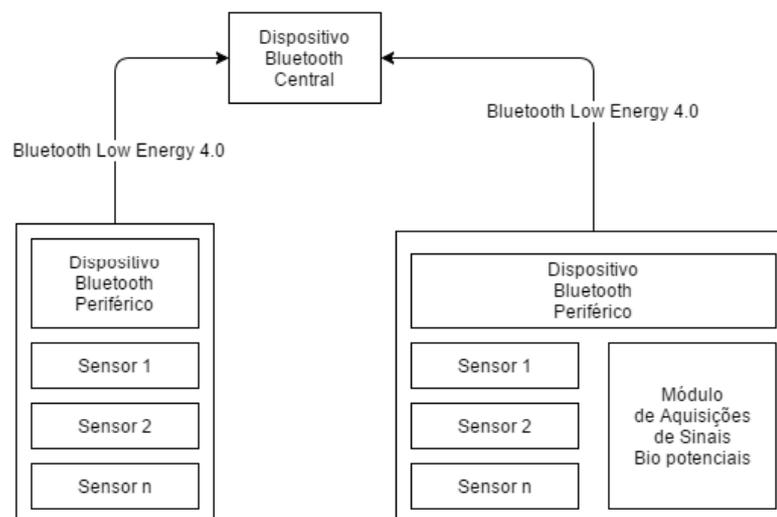


Figura 3.1 – Esquema geral da arquitetura do protótipo

3.1. Tecnologia Bluetooth Low Energy 4.0

A tecnologia Bluetooth é reconhecida pelo público para efetuar ligações de rede sem fios. Esta tecnologia é projetada e comercializada pela corporação, sem fins lucrativos, *Bluetooth Special Interest Group* (SIG) (Bluetooth SIG, 1998). Embora, inicialmente, esta tecnologia tenha sido projetada para uso na área da saúde, *fitness*, segurança, e indústrias dedicadas ao entretenimento, atualmente é reconhecida como sendo uma tecnologia *legacy*, estando presente na maioria dos dispositivos que utilizamos no dia-a-dia.

Desde a primeira versão da especificação do *Bluetooth* (*Core Version 1.0*), publicada em 1999, o *Bluetooth SIG* adotou formalmente várias melhorias para o *Bluetooth Core Specification*, as melhorias implementadas eram apenas ao nível da velocidade de transmissão e da largura de banda.

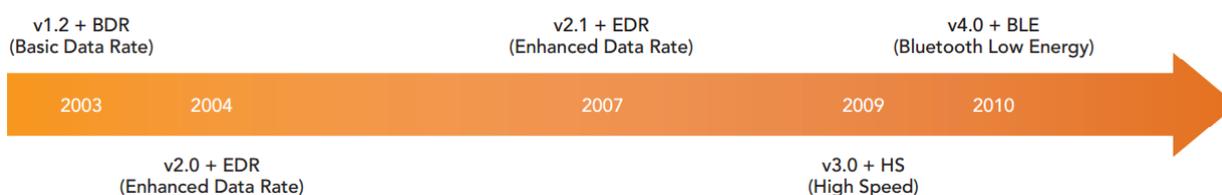


Figura 3.2 – Evolução da especificação do Bluetooth(S.L, 2014)

A Figura 3.2 representa a evolução das versões da especificação do *Bluetooth*, no tempo. No entanto, a partir de 2010, esta tecnologia adotou um novo paradigma e anunciou formalmente uma nova extensão, o *Bluetooth Low Energy* (BLE). Em comparação com o *Bluetooth classic*, o BLE requer consideravelmente menos energia, e consegue apresentar prestações de comunicação similares e por vezes superiores. O BLE está classificado como uma rede de área pessoal sem fios (WPAN), as comunicações são efetuadas numa gama de frequências designada por ISM (*Industrial, Scientific and Medical*), que se destina a aplicações industriais, científicas e médicas.

A especificação *Bluetooth 4.0* define duas tecnologias sem fio distintas e dois novos tipos de dispositivos *Bluetooth*:

- O *Bluetooth Classic* (BR/EDR), que é a tecnologia padrão, sem fios que evoluiu com a especificação *Bluetooth* desde a versão 1.0.
- O *Bluetooth Low Energy* (BLE), que é a tecnologia sem fios, de baixo consumo que foi introduzida com a versão da especificação *Bluetooth 4.0*.

É deste modo possível identificar três tipos de dispositivos *Bluetooth* definidos, onde cada tipo de dispositivo define qual é a tecnologia com que este pode interagir. Os tipos de dispositivos *Bluetooth* são os seguintes:

- Os dispositivos com denominação *Bluetooth* (BR/EDR), suportam apenas a tecnologia *Bluetooth Classic*, e apenas podem comunicar com outros dispositivos que suportem esta tecnologia.
- Os dispositivos *Bluetooth Smart Ready* (BR/EDR/BLE) podem comunicar com qualquer tecnologia *Bluetooth*, seja uma, dispositivo com tecnologia *Bluetooth Classic*, ou um com a tecnologia *Bluetooth Low Energy*.
- Os dispositivos *Bluetooth Smart* (BLE), apenas podem comunicar com outros dispositivos que implementem a mesma tecnologia.

A escolha do tipo de dispositivo *Bluetooth* a utilizar dependerá do tipo de aplicação. Sendo que a implementação de um dispositivo *Bluetooth Smart Ready* para dispositivo central terá mais vantagens pois neste período de transição, permitirá a utilização de ambas as tecnologias. Os dispositivos com a tecnologia *Bluetooth Classic* terão uma implementação mais comum em

aplicações no qual o consumo energético não seja um fator importante. Para esses casos, deverá ser utilizado o dispositivo *Bluetooth Smart*. A Figura 3.3 representa o modelo de arquitetura de *firmware* de cada tipo de dispositivo *Bluetooth*.

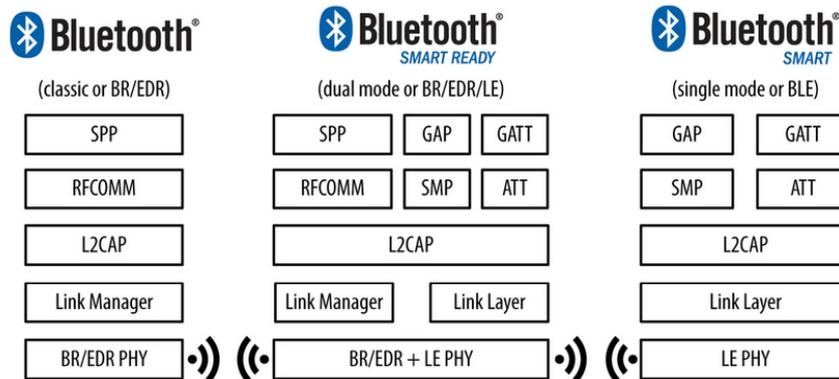


Figura 3.3 - Arquiteturas de *firmware* dos diferentes tipos de dispositivos *Bluetooth* (Sig, 2010).

3.1.1. Topologia de Redes Bluetooth

A topologia da rede *Bluetooth* básica é referida como *Piconet*, consiste num mestre e de um a sete dispositivos escravos ativos. “Uma rede *ad-hoc multihop* de *piconets* em que alguns dos dispositivos presentes estão em mais do que uma *piconet* é referido como uma *scatternet*” (Har-Shai, Kofman, Zussman, & Segall, 2002) Um exemplo de uma *scatternet* é ilustrado na Figura 3.4.

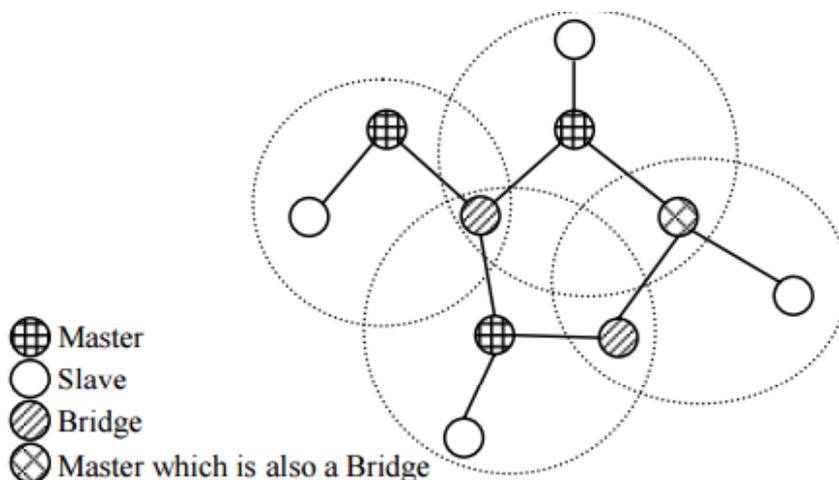


Figura 3.4 - Exemplo de uma *Scatternet Bluetooth* (Har-Shai et al., 2002)

Um dispositivo que é membro de mais do que uma *piconet* é referido como o *Bridge*, este dispositivo deve marcar a sua presença em todas as *piconets* de que é membro. Nas especificações do *Bluetooth* (Sig, 2010), a atribuição da largura de banda pelo mestre para cada *link* na sua *piconet* é deixada em aberto. O mestre agenda o tráfego dentro de uma *piconet* por meio de *polling* e determina como a largura de banda deve ser distribuída entre os escravos. É

de realçar que a *piconet* é a topologia mais utilizada, pela rede *Bluetooth* e apenas em casos muito específicos será utilizado a topologia *scaternet*.

3.1.2. Camadas de Arquitetura de Sistema

Esta seção apresenta a *stack* do protocolo BLE, e descreve os principais mecanismos e características de cada camada. “Como no *Bluetooth classic* (Bisdikian, 2001), a *stack* do protocolo BLE é composta de duas partes principais: o *Host* e o *Controller*. O *Controller* compreende a camada física e a camada de *Link*, e é tipicamente implementado como um pequeno *System-on-Chip* (SOC) com um rádio integrado. O *Host* é executado no processador da aplicação e inclui uma camada de funcionalidade que contem o *Logical Link Control and Adaptation Protocol* (L2CAP), o *Attribute Protocol* (ATT), o *Generic Attribute Profile* (GATT), o *Security Manager Protocol* (SMP) e o *Generic Access Profile* (GAP).” (Gomez, Oller, & Paradells, 2012). A comunicação entre o *Host* e o *Controller* é conhecido como um padrão denominado por *Host Controller Interface* (HCI), a Figura 3.5 representa um exemplo da interligação entre o *Host* e o *Controller*.

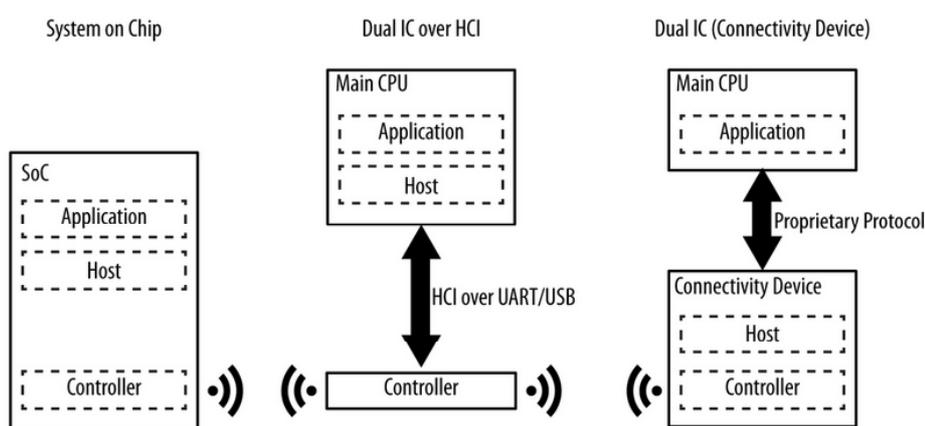


Figura 3.5 - Comunicação entre *Host* e *Controller* (Sig, 2010).

A protocolo *Bluetooth Low Energy* está organizado segundo o modelo representado pela Figura 3.6, dividindo as várias camadas entre o *Host* e o *Controller*.

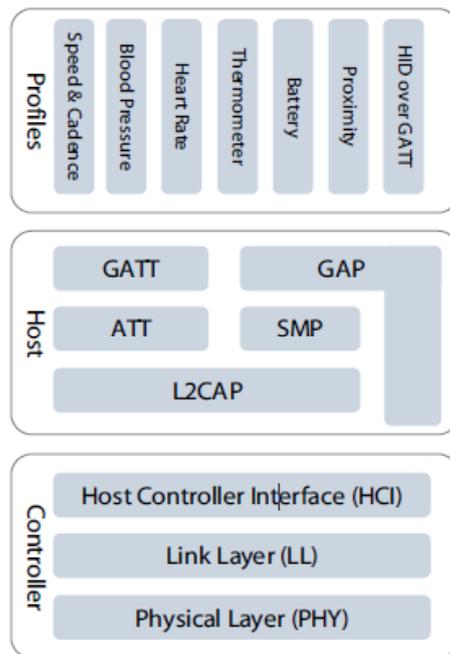


Figura 3.6 – Arquitetura de firmware de um dispositivo *Bluetooth Low Energy* (Nordic Semiconductor, 2014).

3.1.2.1. Physical Layer

Como referido anteriormente, “o *Bluetooth* opera na gama de frequências industrial, científica e médica (ISM) e define 40 canais de radio frequência com espaçamento de 2 MHz entre canais. Existem dois tipos de canais BLE, os canais de *Advertising* e os canais de dados”.(Gomez et al., 2012) Os canais de *Advertising* são utilizados na descoberta de dispositivos, estabelecimento da ligação e transmissão em *Broadcast*, onde os canais de dados são utilizados para a transmissão bidirecional entre dois dispositivos ligados. Estão definidos 3 canais de *Advertising* dos 40 que podem ser utilizados pela BLE. Estes canais foram atribuídos a frequências centrais de modo a minimizar a sobreposição com os canais IEEE 802.11, sendo estes os canais 37, 38 e 39, os quais são bastante comuns em vários países. Os restantes 37 canais são utilizados para transmissão de dados. A Figura 3.7 ilustra uma representação visual da distribuição dos canais utilizados na comunicação *Bluetooth*.

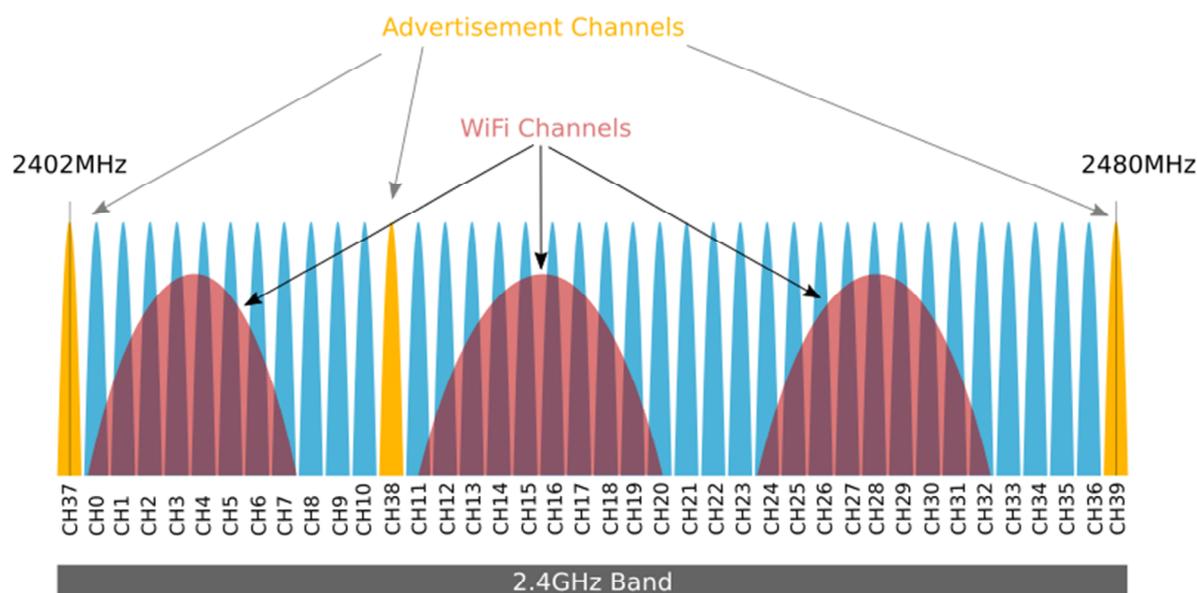


Figura 3.7 - Bluetooth Low Energy Channels (Argenox, 2015)

A camada física utiliza um mecanismo de frequência adaptativa, nos canais de dados de modo a diminuir os problemas com a interferência e propagação sem fios, como o *fading* ou *multipath*. Este mecanismo seleciona um dos 37 canais de dados disponíveis para comunicação por um tempo determinado. Todos os canais utilizam a modulação *Gaussian Frequency Shift Keying* (GFSK).

3.1.2.2. Link Layer

No BLE quando um dispositivo apenas necessita de realizar uma transmissão de dados por *Broadcast*, os dados são transmitidos em pacotes através dos canais de *Advertising*. Qualquer dispositivo que transmite pacotes de *Advertising* é designado de *Advertiser*. A transmissão dos pacotes de dados através dos canais de *Advertising* é feita em intervalos de tempo designados de *Advertising events*. “Em cada *Advertising event*, o *Advertiser* usa sequencialmente cada um dos canais de *Advertising* para a transmissão do pacote. Os dispositivos, que apenas recebem dados através dos canais de *Advertising* são designados de *scanners*” (Gomez et al., 2012).

A criação de uma ligação entre dois dispositivos é um procedimento assimétrico no qual um *Advertiser* anuncia através dos canais de *Advertising*, que um dispositivo está disponível para uma ligação, sendo referido como o iniciador (*Initiator*), este escuta os eventos de *Advertisement*. Quando um iniciador encontra um *Advertiser*, este pode transmitir uma mensagem de pedido ao *Advertiser*, que cria uma ligação ponto a ponto entre os dois dispositivos. Ambos os dispositivos podem comunicar utilizando os canais de dados. Os pacotes de dados para esta ligação serão identificados com um código de acesso de 32-bit gerado aleatoriamente. O BLE define dois papéis para os dispositivos na *Link Layer* para a criação de uma ligação, o mestre e o escravo. Estes são os dispositivos que atuam como *Initiator* e *Advertiser* durante a criação da ligação, respetivamente. Um mestre pode gerir múltiplas

ligações simultâneas com escravos diferentes, no entanto, cada escravo apenas pode estar ligado a um mestre. De modo a reduzir o consumo energético, os escravos entram em modo de baixo consumo de energia (*Sleep*) e acordam periodicamente para escutar por possíveis pacotes de dados que tenham sido enviados pelo mestre. O mestre determina os instantes nos quais os escravos devem escutar, e coordena o acesso ao meio de transmissão através de um esquema TDMA, (*Time Division Multiple Access*). O mestre também fornece ao dispositivo escravo a informação necessária para a supervisão da ligação e o mapa de canais de dados a ser utilizado. Uma vez que a ligação tenha sido estabelecida entre o mestre e o escravo o canal físico é dividido em unidades de tempo, designados por eventos de ligação. Dentro de um evento de ligação todos os pacotes de dados são transmitidos utilizando o mesmo canal de dados e a mesma frequência. Todos os eventos de ligação começam com uma transmissão de um pacote de dados por parte do mestre. No entanto o mestre pode não enviar um pacote de dados quando recebe um pacote de um dispositivo escravo. Quando não existir mais dados para transmitir, por parte de nenhum dos dispositivos, o evento de ligação é terminado e o escravo não será obrigado a escutar até ao início do próximo evento de ligação. Quando existir um novo evento de ligação, o mestre e escravo utilizam um novo canal de dados. As ligações da *Link Layer* utilizam um mecanismo de controlo de fluxo *Stop-and-Wait*.

3.1.2.3. Logical Link Control And Adaptive Protocol Layer

A camada L2CAP é utilizada no BLE e é uma versão otimizada e simplificada do protocolo *Bluetooth classic L2CAP*. No BLE, o objetivo principal do L2CAP é o de multiplexar os dados das 3 camadas superiores do protocolo, ATT, SMP e o controlo da *Link Layer*. Os dados destes serviços são manipulados pela camada L2CAP, e não usam os mecanismos de controlo de fluxo, que estão disponíveis em outras versões do Bluetooth. A segmentação dos pacotes de dados não é utilizada, uma vez que as camadas superiores do protocolo fornecem unidades de dados que não excedem o tamanho máximo dos dados enviados pelo L2CAP, que é de 23 bytes no BLE.

3.1.2.4. Attribute Protocol Layer

A camada ATT define a comunicação entre dois dispositivos com os papéis de servidor e cliente, no topo de um canal L2CAP dedicado. O servidor mantém um conjunto de atributos. Um atributo é uma estrutura de dados que guarda a informação, gerida pelo GATT, o protocolo que opera na parte superior do ATT. O papel de cliente ou servidor é determinado pelo GATT, e é independente dos papéis de mestre ou escravo. O cliente pode aceder aos atributos do servidor através do envio de pedidos, que ativam as mensagens de resposta do servidor. Para uma maior eficiência, o servidor pode também enviar para o cliente dois tipos de mensagem não solicitadas que contêm atributos:

- As notificações, que não são confirmadas;
- As indicações, que requerem que o cliente envie uma confirmação.

O cliente pode também enviar comandos para o servidor para escrever os valores dos atributos. As transações de Pedidos/Respostas e as Indicações/Confirmação seguem o esquema *stop-and-wait*.

3.1.2.5. Generic Access Profile Layer

As únicas camadas que necessitam de interagir com a camada de aplicação são as camadas GAP e GATT. No BLE os perfis definem o funcionamento da aplicação, o GAP (*Generic Access Profile*) fornece a estrutura para todos os perfis, definindo como é que dois dispositivos descobrem e estabelecem uma ligação. Para conseguir isso o GAP define a função do dispositivo, existem quatro funções distintas, cada função tem requisitos específicos para o controlador subjacente:

- *Broadcaster*
- *Observer*
- *Peripheral*
- *Central*

Um dispositivo com o papel de *Broadcaster* apenas envia dados em modo de *Broadcast* pelos canais de *Advertisement* e não suporta ligações com outros dispositivos. O papel de *Observer* é complementar ao papel do *Broadcaster*, isto é, tem o propósito de receber dados transmitidos pelo *Broadcaster*. O dispositivo *Central* é projetado para um dispositivo capaz de iniciar e gerir múltiplas ligações, por outro lado os dispositivos com o papel de *Peripheral* são projetados para uma única ligação com o dispositivo com papel de *Central*. Como consequência, o dispositivo *Central* e o dispositivo *Peripheral* requerem que o controlador do dispositivo suporte tanto o papel de *mestre* como de *escravo*. Um dispositivo pode suportar vários papéis, no entanto apenas pode assumir um papel de cada vez.

Uma vez, que certos tipos de aplicação podem beneficiar da utilização de uma funcionalidade comum, podem ser construídos *Bluetooth Profiles* no topo do GAP. O *Bluetooth* segue uma hierarquia de perfis e como tal, pode ser definido um novo perfil que inclua todos os requerimentos de um perfil existente. Um perfil de alto nível que especifica como é que as aplicações devem operar, é chamado de *Application Profile*. Os *Application Profiles*, são também especificados pelo *Bluetooth SIG* (SIG, 2015), para favorecer a interoperabilidade entre dispositivos de diferentes de fabricantes.

3.1.2.6. Generic Attribute Profile layer

A camada *Generic Attribute Profile* (GATT) define a estrutura que usa os atributos para a descoberta dos serviços, e para a troca de características de um dispositivo para o outro. Um serviço é uma coleção de dados e comportamentos associados a uma função particular de um dispositivo. Um serviço pode referenciar outros serviços primários e secundários e/ou um conjunto de características que compõe o serviço. Existem dois tipos de serviços: o primário e o secundário. O serviço primário fornece uma funcionalidade primária ao dispositivo. Um

serviço secundário fornece uma funcionalidade auxiliar ao dispositivo e é referenciado, por pelo menos um serviço primário do dispositivo.

	Handle	UUID	Permissions	Value
Service	0x0001	SERVICE	READ	HRS
Characteristic	0x0002	CHAR	READ	HRM
	0x0003	HRM	READ/NOTIF	80 bpm
Descriptor	0x0004	DESC	READ	NOTIFY

Figura 3.8 - Estrutura utilizada pelo GATT

Uma característica é um valor utilizado por um serviço em conjunto com propriedades e configurações de informação sobre, como é que esse valor é acessado, mostrado ou representado. A definição de uma característica, contém a declaração da característica, das suas propriedades e do seu valor. Pode também conter descritores (*Descriptors*) que descrevem o valor ou permitem a configuração do servidor com o respetivo valor da característica. (Bluetooth SIG, 2010). A Figura 3.8 representa o modelo utilizado pelo GATT para definir serviços e características.

3.2. Camada de Abstração de Hardware STM32Cube

O desenvolvimento de software para sistemas embebidos, é normalmente, desenvolvido sobre uma camada de software capaz de abstrair o *hardware* do sistema embebido. Em aplicações específicas por vezes esta camada de abstração de *hardware* não pode ser utilizada, devido ao nível de otimização necessário para a aplicação, no entanto, a utilização desta camada de abstração de hardware torna-se muito vantajosa, pois torna a aplicação de software, mais fácil de portar para outros sistemas embebidos, com um tipo de hardware diferente. Com este pressuposto foi criado o CMSIS (*Cortex Microcontroller Software Interface Standard*) (ARM, 2013). O CMSIS é uma camada de abstração de hardware, independente de fabricantes de microprocessadores, que agrupa características comuns de um tipo de núcleo de microprocessador, por exemplo, o NVIC (*Nested Vector Interrupt Controller*), temporizadores, e interfaces de *debug* padrão e disponibiliza-as sobre a forma de uma API (*Application Programming Interface*) para os fabricantes de microprocessadores. Torna-se interessante a realização de um pequeno estudo da arquitetura deste padrão, uma vez que a sua utilização é comum a todos os processadores com núcleos ARM. A Figura 3.9 representa essa mesma arquitetura.

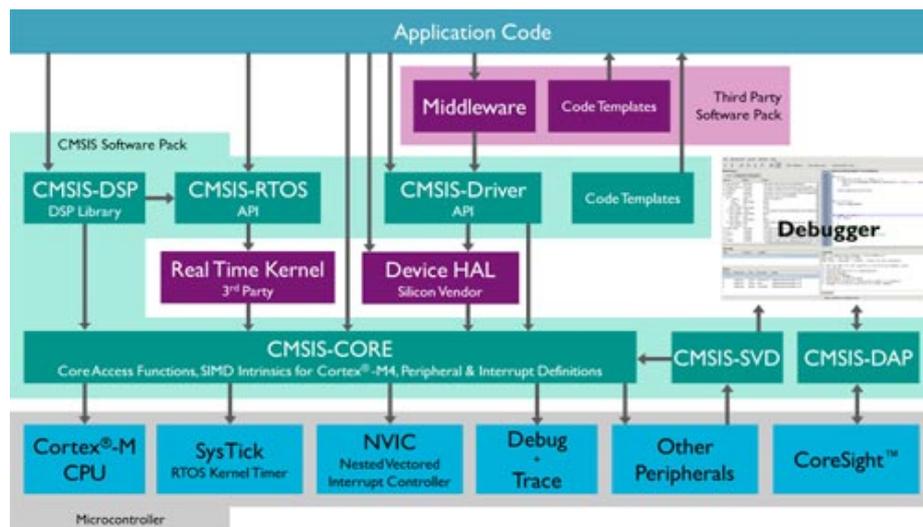


Figura 3.9 – Arquitetura *Cortex Microcontroller Standard Interface* (ARM, 2013)

O CMSIS consiste nos seguintes componentes:

- O CMSIS-CORE, é uma API para o processador com núcleo Cortex-M e os seus periféricos, esta fornece uma interface padrão para o Cortex-M0, Cortex-M3, Cortex-M4, SC00 e SC300.
- O CMSIS-Driver, define um driver para as interfaces genéricas para *middleware*, sendo reutilizável através de todos os dispositivos suportados. A API (*Application Programming Interface*) é independente do RTOS (*Real Time Operating System*) e liga os periféricos do microcontrolador com o *middleware* que são implementadas, por exemplo, em *stacks* de comunicação, como USB ou TCP/IP.
- O CMSIS-DSP: contem uma biblioteca que fornece funções otimizadas para processamento de sinais nos processadores Cortex-M0, Cortex-M3 e Cortex-M4.
- O CMSIS-RTOS API, contém uma API comum para sistemas operativos de tempo real, e fornece uma interface de programação portátil para muitos RTOS.
- O CMSIS-PACK é um ficheiro de descrição XML que contem uma coleção chamada de *software pack* que inclui os ficheiros de programação bibliotecas e documentação, para cada dispositivo.
- O CMSIS-SVD *System View Description* descreve os periféricos do dispositivo num ficheiro de XML, que pode ser utilizado para identificar esse periférico e os seus registos por parte dos *debuggers*.
- O CMSIS-DAP, *Debug Access Port* é um *firmware* padrão que liga com o *CoreSight Debug Access Port* para realizar *debug* de uma unidade.

Esta arquitetura permite aos fabricantes e vendedores de dispositivos com núcleos ARM, criar uma abstração de *hardware* que utiliza as API's do CMSIS. Esta camada de Abstração é conhecida como Device HAL. A maior parte dos fabricantes disponibilizam, uma camada de abstração de hardware para um tipo de microprocessadores ou para uma família que tenham

disponível no seu portfólio. A iniciativa *STM32Cube* (ST Microelectronics, 2015c) foi criada pela *ST Microelectronics* (ST Microelectronics, n.d.-a) para diminuir o esforço de desenvolvimento, tempo e custo de um projeto de sistemas embebidos. O *STM32Cube* cobre todo o portfólio de processadores da família STM32 este inclui o seguinte software:

- O STM32CubeMx, (ST Microelectronics, 2015d) que constitui uma ferramenta de *software* gráfico, que permite configurar e gerar código C para a configuração e inicialização básica dos periféricos de um microprocessador da família STM32.
- O STM32Cube HAL, que é uma camada de *software* para processadores utilizados em aplicações embebidas, que realiza a abstração do hardware.

À data de realização deste trabalho a *ST Microelectronics* disponibiliza dois conjuntos de software para a família STM32, a SPL (*Standard Peripheral Library*) e o STMCube, no entanto o conjunto de software SPL não terá mais atualizações futuras e irá deixar de ser suportada num futuro próximo. Na realização deste trabalho optou-se pela utilização do *STM32Cube*, pois será a versão que terá suporte no futuro. A solução de *software STM32Cube* foi construída á volta de três níveis independentes que podem facilmente interagir uns com os outros. A Figura 3.10 expõe a estrutura organizacional do pacote de *software STM32Cube*.

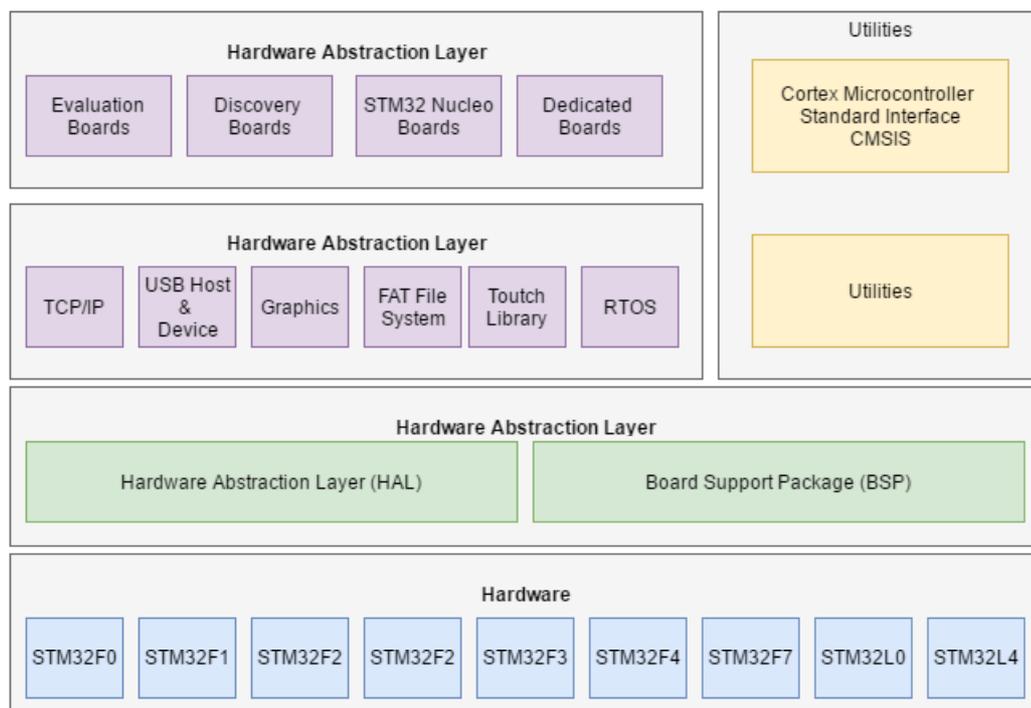


Figura 3.10 - Arquitetura STM32Cube (ST Microelectronics, 2015a)

A Figura 3.11 representa a interligação existente entre as várias camadas do *SMT32Cube*. Em seguida, será feita uma breve descrição das várias camadas e dos seus blocos principais, assim como uma breve descrição da sua funcionalidade.

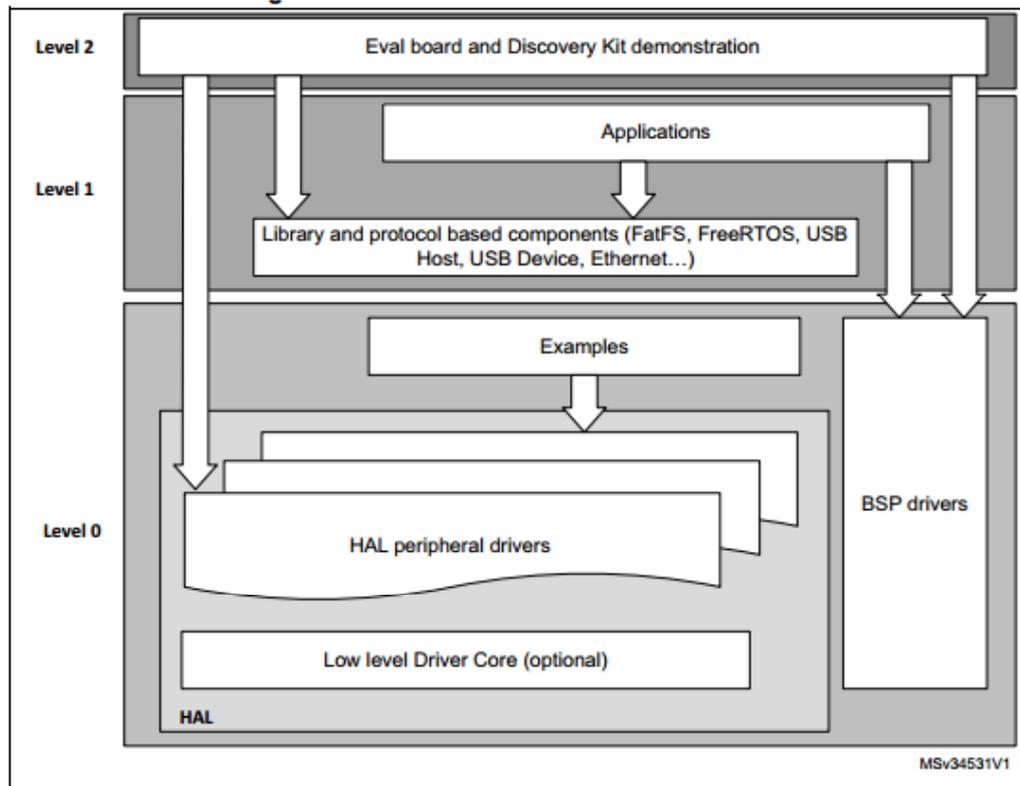


Figura 3.11 – Interligação entre os vários níveis do *STM32Cube* (ST Microelectronics, 2015a)

Nível 0 (*Level 0*) – Este nível encontra-se dividido em três subcamadas:

- A camada BSP (*Board Support Package*) – que oferece um conjunto de API's relacionada com os componentes de hardware disponíveis nas placas de desenvolvimento, encontrando-se dividida em duas partes:
 - *BSP Driver* que disponibiliza, API's específicas para cada placa de desenvolvimento.
 - *Component* que disponibiliza API's específicas para os componentes externos ao *BSP Driver* e pode ser portado para qualquer placa de desenvolvimento.
- A camada HAL (*Hardware Abstraction layer*) fornece os drivers de baixo nível e os métodos de interface de *hardware* com as camadas superiores.
- A camada *Basic Peripheral Usage Examples* contém, como o próprio nome indica exemplos de operação básica dos periféricos, e utiliza apenas os recursos da camada HAL e BSP.

Nível 1 (*Level 1*) – Este nível está subdividido em duas camadas:

- A camada de componentes de *Middleware*, que contém um conjunto de bibliotecas para *stacks* de comunicação, por exemplo USB, TCP/IP, e também suporte para outro tipo de *middleware* como RTOS (*Real Time Operating Systems*); por exemplo o *FreeRTOS*.

- A camada com exemplos de aplicação de componentes de *Middleware*, que demonstram o funcionamento de um determinado componente de através de uma aplicação de exemplo.

Nível 2 (*Level 2*) – Este nível é composto apenas por uma camada, que contem uma demonstração global, gráfica em tempo real, com base na camada *middleware*.

Neste projeto são apenas utilizadas a camada HAL (*Hardware Abstraction Layer*) e a camada designada como *Utilities*, a qual contem métodos de acesso ás API's do CMSIS.

3.3. Meios de Comunicação em Sistemas Embebidos

A interação com aplicações de sistemas embebidos é realizada através de meios de comunicação, sobre os quais estão implementados protocolos que permitem o controlo do fluxo da comunicação. Nas secções seguintes foi feito o estudo dos protocolos e meios de comunicação mais usuais em aplicações de sistemas embebidos com aplicação neste projeto.

3.3.1. Universal Synchronous Asynchronous Receiver Transmitter

O meio de comunicação USART (*Universal Synchronous Asynchronous Receiver Transmitter*) é também designado como SCI (*Serial Communications Interface*) é um tipo de comunicação serie e pode ser dividida em dois grupos: comunicações síncronas e comunicações assíncronas.

A interface serie síncrona utiliza sempre pares de comunicação nas linhas de dados, uma para o envio e outra para a receção, normalmente denominados de Tx e Rx, respetivamente. Além disso utiliza também um sinal de relógio, de modo a que todos os dispositivos tenham a mesma referência de relógio (*Clock*). A utilização da interface serie síncrona, permite transferências mais rápidas, mas requer pelo menos uma ligação extra entre os dispositivos de comunicação. Existe também uma ligação síncrona que utiliza duas ligações extra entre dispositivos e apenas um par de ligação para o envio de dados, as duas linhas auxiliares são denominadas de RTS (*Request To Send*) e CTS (*Clear To Send*), quando um dispositivo pretende enviar dados para outro coloca a linha RTS no estado alto, e aguarda que o dispositivo coloque a linha CTS no estado alto, nesse momento a transmissão de dados pode ser executada.

A interface serie assíncrona, transfere dados sem o apoio de linhas auxiliares, e apenas necessita de um par de linhas para a transmissão de dados. Este método de transmissão minimiza o número de I/O (Inputs/Outputs), que é necessário para realizar a ligação entre os dois dispositivos. Como não existe um sinal de relógio no funcionamento assíncrono, a transmissão e receção podem ocorrer ao mesmo tempo, esta operação é conhecida como uma ligação *full-duplex*.

O modo de uso mais comum do USART é o modo assíncrono UART (*Universal Asynchronous Receiver Transmitter*) e é normalmente utilizado para realizar comunicações entre uma aplicação de sistemas embebidos e um PC (*Personal Computer*), para controlar o fluxo da ligação é utilizado o protocolo RS-232. A porta serie pode ser utilizada para realizar

comunicação entre dois dispositivos de sistemas embebidos. Em seguida, será apenas descrito o protocolo serie de transferência assíncrona, pois é o mais comum e o que será utilizado neste projeto.

O protocolo serie assíncrono tem uma série de mecanismos que ajudam a garantir a transferência de dados, de forma robusta e livre de erros. Estes mecanismos são:

- Número de *bits* de dados (*Data Bits*)
- Número de *bits* de sincronização (*Stop Bits*)
- Número de *bits* de paridade (*Parity*)
- Número de *bits* transmitidos num segundo (*Baudrate*)

A taxa de transmissão de dados (*Baudrate*), indica o quão rápido os dados são enviados através das linhas de dados. O valor é expresso na unidade bps (*bits per second*). O inverso deste valor indica o tempo que é necessário para transmitir um único *bit*. Este valor determina também o tempo que um dispositivo transmissor deve manter a linha de dados no nível lógico um e no nível Lógico zero, e qual o período com o qual o dispositivo recetor deve realizar a amostragem dos bits de dados. Cada bloco de dados (normalmente 1 Byte) transmitido, é enviado numa estrutura de *bits*. As estruturas de *bits* são criadas ao anexar os *bits* de sincronização e de paridade aos *bits* de dados. A Tabela 3.1, representa o formato da estrutura utilizada, para transmissão de dados através de uma interface serie.

Tabela 3.1 – Estrutura de dados utilizado na transmissão serie

	Start Bit	Data Bits	Parity Bits	Stop Bits
Número de bits	1	5-9	0-1	1-2

O campo *Data Bits* pode conter de 5 a 9 *bits*. No entanto, o tamanho de dados padrão é o *byte*, com uma base de 8 *bits*. Um segmento de dados de 7 *bits* pode ser mais eficiente do que 8, especialmente se estamos apenas a transferir caracteres ASCII (*American Standard Code for Information Interchange*) de 7 *bits*. Além do tamanho do campo *Data Bits* da estrutura dados a enviar, é necessário especificar também a ordenação dos *bits* (*endianess*). Os bits podem ser enviados com o *bit* mais significativo, ou MSB (*Most Significant Bit*) em primeiro (*Big endian*), ou com o *bit* menos significativo ou LSB (*Least Significant Bit*) em primeiro (*Little Endian*). Caso nenhuma opção seja indicada o envio dos dados será em *Big Endian*.

Os campos *Start Bit* e *Stop Bits* são campos especiais que são transferidos em cada estrutura de dados. Tal como o seu nome indica, estes *bits* indicam o inicio e o fim da estrutura de dados. Apenas existe um *Start Bit* mas o número de *Stop Bits* é configurável para um ou dois, normalmente utiliza-se apenas um *Stop Bit*. O *Start Bit* é sempre indicado pela transição da linha de dados do nível alto (*High*) para o nível baixo (*Low*), enquanto que o *Stop Bit(s)* irá repor a linha no estado de repouso.

Os *bits* de paridade são uma forma muito simples de verificação de erros, de baixo nível. A paridade pode ser, par ou ímpar. Para produzir o bit de paridade, todos os 5-8 bits do byte de dados são somados, e a uniformidade da soma decide se o bit é definido ou não. A utilização do bit de paridade, é opcional, pode ser útil quando o meio de transmissão é suscetível de produzir erros devido á existência de ruído.

A interface serie é composta por apenas duas ligações, uma para o envio de dados e outra para a receção, além destas duas ligações é também necessário a ligação dos planos de GND (*Ground*) de ambos os dispositivos. Na Figura 3.12, esta representado a interface UART entre dois dispositivos.



Figura 3.12 – Interface serie entre dois dispositivos

3.3.2. Inter-Integrated Circuit

O I2C (*Inter-Integrated Circuit*) foi originalmente desenvolvido em 1982 pela *Philips*, e desde 2006 pela NXP Semiconductors, (NXP Semiconductors, 2007). A especificação original permitia apenas comunicações de 100kHz, e fornecia apenas endereços de 7 bits, o que limitava o número de dispositivos no barramento a 112 dispositivos. Em 1992, foi publicada a primeira especificação, adicionando o modo rápido de 400kHz, bem como um espaço de endereçamento expandido a 10 bits. Atualmente os modos disponíveis para I2C estão descritos na Tabela 3.2.

Tabela 3.2 – Modos de Operação I2C (NXP Semiconductors, 2007)

Modo de Operação	Comunicação até
Standard Mode	400 KHz
Fast Mode	1 Mbit/s
Fast Mode Plus	1 Mbit/s
High Speed Mode	3.4 Mbit/s
Ultra Fast Mode (unidirecional)	5Mbit/s

Cada barramento I2C necessita de dois sinais, o SCL e o SDA. O SCL é o sinal de relógio, e o SDA é o sinal de dados. Um barramento I2C pode suportar até 112 dispositivos

com endereços de 7 bits e 1008 com endereços de 10 bits dispositivos escravos. A Figura 3.13, representa um exemplo de um barramento I2C, com múltiplos dispositivos.

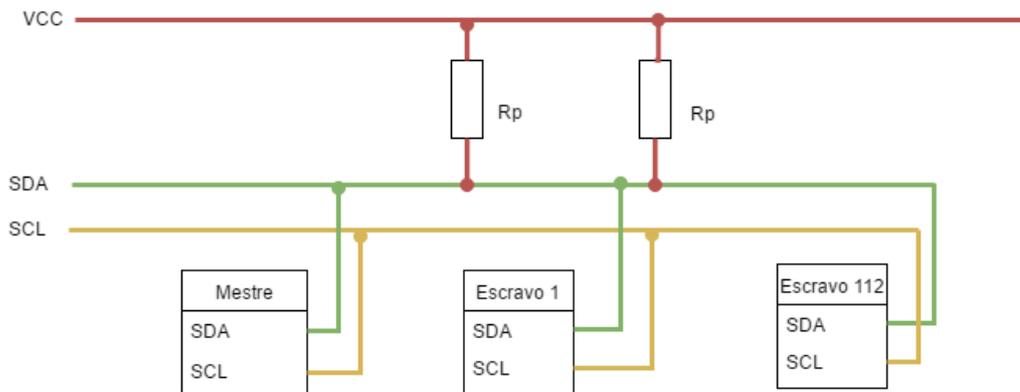


Figura 3.13 – Exemplo da interface com barramento I2C

Os drivers de I2C presentes em cada dispositivo são *Open-drain*, o que significa que estas podem colocar as linhas dos sinais no nível lógico zero, mas não as conseguem colocar num nível lógico um. Assim sendo, cada linha tem uma resistência de *Pull-Up* (R_p), para restaurar o sinal ao nível alto depois de um dispositivo o ter colocado no nível baixo.

As comunicações via I2C são um pouco mais complexas do que as comunicações com UART ou SPI. As transferências de dados são divididas em dois tipos de estruturas: uma estrutura de endereço, onde o mestre indica ao escravo para o tipo de estrutura que vai ser enviada em seguida, e uma ou mais estruturas de dados, que são estruturas de dados de 8 bits passados do mestre para escravo ou vice-versa. Os dados são colocados na linha SDA logo após a passagem da linha SCL para o nível lógico zero, e a amostragem desses dados, é realizada a seguir à passagem da linha de SCL para o nível lógico um.

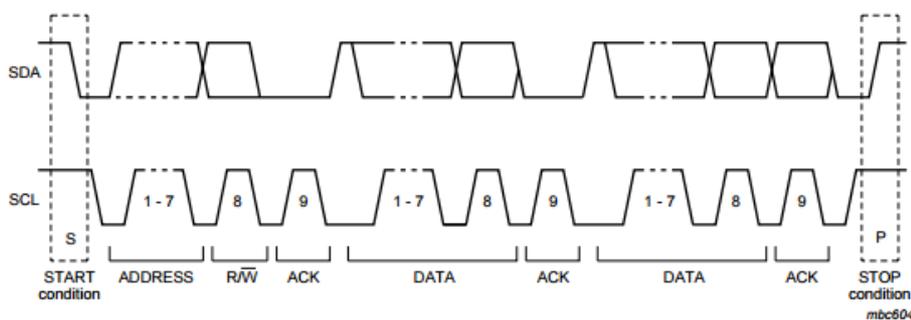


Figura 3.14 - Sequência completa de transferência I2C(NXP Semiconductors, 2004)

A Figura 3.14 representa a sequência de uma transferência completa através de I2C e enumera os principais comandos utilizados na transmissão I2C.(NXP Semiconductors, 2004) Para iniciar o envio de uma estrutura do endereço, o dispositivo mestre coloca a linha SCL (*Serial Clock Line*) no nível lógico um e coloca a linha SDA (*Serial Data Line*) no nível lógico zero, esta operação é denominada por *Start Condition* e quando ocorre indica a todos os

dispositivos escravos que uma transmissão está prestes a começar. Se dois dispositivos mestres desejarem utilizar o barramento ao mesmo tempo, o primeiro dispositivo que colocar a linha SDA no nível baixo, é o dispositivo que fica com o controlo sobre o barramento. É possível emitir *Start Conditions* repetidas, dando início a uma nova sequência de comunicação sem perder o controlo do barramento para outros dispositivos mestre.

A estrutura do endereço é sempre a primeira a ser enviada, em qualquer nova sequência de comunicação. Para um endereço de 7 bits, o bit mais significativo (MSB) é amostrado em primeiro lugar, seguido por um bit de R/W que indica se esta é uma operação de escrita (1), ou uma operação de leitura (0). O nono bit da estrutura é o bit NACK/ACK, em todas as estruturas de dados ou de endereços. Após o envio dos 8 primeiros bits da estrutura terem sido enviados, o dispositivo recetor passa a controlar a linha SDA. Se o dispositivo recetor não puxar a linha SDA para o nível lógico zero, antes do nono pulso do relógio (clock), pode-se inferir que o dispositivo de recetor ou não conseguiu receber os dados ou não conseguiu analisar os dados da estrutura enviada. Nesse caso, a troca de dados termina, e cabe ao dispositivo mestre decidir como proceder. Após o envio da estrutura de endereço, o dispositivo mestre pode começar a transmitir a estrutura de dados. O dispositivo mestre tem de continua a gerar pulsos de relógio com um intervalo regular, e os dados são colocados na linha SDA pelo dispositivo mestre ou pelo dispositivo escravo, dependendo se o bit R/W indicou uma operação de leitura ou de escrita. O número de estruturas de dados é arbitrário, e a maioria dos dispositivos escravos auto incrementa o endereço do registo interno. Depois de todas as estruturas de dados serem enviadas, o dispositivo mestre irá gerar uma *Stop Condition*. As *Stop Conditions* são definidas por uma transição do nível lógico zero para o nível lógico um, na linha SDA, após uma transição do nível logico zero para o nível lógico um na linha SCL, com SCL restante a nível lógico um.

3.3.3. Serial Peripheral Interface

O SPI (*Serial Peripheral Interface*) é um barramento síncrono com uma interface de comunicação serie e é utilizado para comunicações de curta distancia. A interface foi desenvolvida pela Motorola, tendo-se tornado um *de facto standard*, na comunicação entre processadores, periféricos, sensores e outros dispositivos. A comunicação entre dispositivos SPI é realizada em modo *full-duplex*, e utilizam o modelo mestre-escravo, num único barramento, onde apenas existe um dispositivo mestre. (Motorola Inc, 2003). O dispositivo mestre inicia a leitura e escrita. No barramento, são suportados múltiplos dispositivos escravos, onde cada um tem uma linha dedicada para que o dispositivo mestre possa realizar a seleção do mesmo, afim de comunicar com este. A Figura 3.15 representa um exemplo de uma interface SPI, o barramento SPI utiliza quatro sinais lógicos distintos:

- SCLK (*Serial Clock*)
- MOSI (*Master OutPut, Slave Input*)
- MISO (*Master Input, Slave Output*)
- NSS (*Slave Select*)



Figura 3.15 – Exemplo de uma interface SPI

A linha MISO é configurada como sendo uma entrada para o dispositivo mestre, e como uma saída para o dispositivo escravo, e é uma de duas linhas utilizada para a transferência de dados série, neste caso do escravo para o mestre. A linha MISO quando não é utilizada é colocada no estado de alta impedância, se o dispositivo escravo não estiver selecionado. A linha MOSI é configurada como saída do dispositivo mestre e como entrada do dispositivo escravo e é também uma linha de transmissão de dados, neste caso a linha de comunicação utilizada para enviar dados do mestre para o escravo. A linha SCLK é utilizada para sincronizar a transferência dos dados de entrada e de saída do dispositivo através das linhas MOSI e MISO. Os dispositivos, mestre e escravo são capazes de transmitir um *Byte*, de informação durante uma sequência de 8 ciclos de relógio (*Clock*). Como o relógio é gerado pelo dispositivo mestre, esta linha é uma entrada no dispositivo escravo. A linha \overline{NSS} é uma linha utilizada para selecionar o dispositivo escravo. A linha \overline{NSS} é uma linha de saída utilizada para selecionar o dispositivo escravo. A linha \overline{NSS} deve ser colocada no nível lógico zero, imediatamente antes da transmissão de dados, e após a transmissão de dados deve ser colocada no nível lógico um (Texas Instruments, 2010).

O SPI é um barramento de dados síncrono onde a transmissão é realizada quando os dispositivos estão sincronizados. O sinal de relógio fornece indica ao dispositivo receptor, quando este deve realizar uma amostragem dos bits na linha de dados. Para iniciar a comunicação, o mestre do barramento configura o relógio, utilizando uma frequência que seja suportada pelo dispositivo escravo. Durante cada ciclo do relógio SPI, ocorre uma transmissão de dados *full-duplex*. O dispositivo mestre envia um *bit* na linha MOSI e o escravo lê esse *bit*, ao mesmo tempo o dispositivo escravo, envia na linha MISO um bit e o mestre lê esse bit, a leitura de cada bit é indicado pela transição do sinal de relógio. Este processo mantém-se inalterado mesmo quando é realizada apenas uma transmissão unidirecional. Quando não é necessário transferir mais dados o dispositivo mestre para de gerar o sinal de relógio e tipicamente, remove a seleção do escravo, colocando a linha \overline{NSS} no nível alto (ST Microelectronics, 2013). O dispositivo mestre fornece o sinal de relógio e para além de definir a sua frequência, também é necessário definir qual a polaridade e fase do mesmo, no que diz respeito aos dados. O Guia do bloco SPI da *Freescale Semiconductor* (Freescale Semiconductor, 2004) designou estas duas propriedades de “CPOL” e “CPHA”, respetivamente, a maioria dos fornecedores adotaram esta convenção. As combinações da

polaridade e da fase são muitas vezes referidas como um modo de operação, e estão comumente numeradas de acordo com a convenção da Tabela 3.3.

Tabela 3.3 – Modos de operação SPI (Motorola Inc, 2003)

Modo de Operação	CPOL	CPHA
0	0	1
1	0	0
2	1	1
3	1	0

Um barramento SPI suporta a ligação de múltiplos escravos onde, cada escravo tem uma linha de sinal NSS individual, para que possa ser seleccionado individualmente afim de realizar uma transmissão. A Figura 3.16 representa um exemplo de uma interface com múltiplos escravos ligados a um barramento SPI.

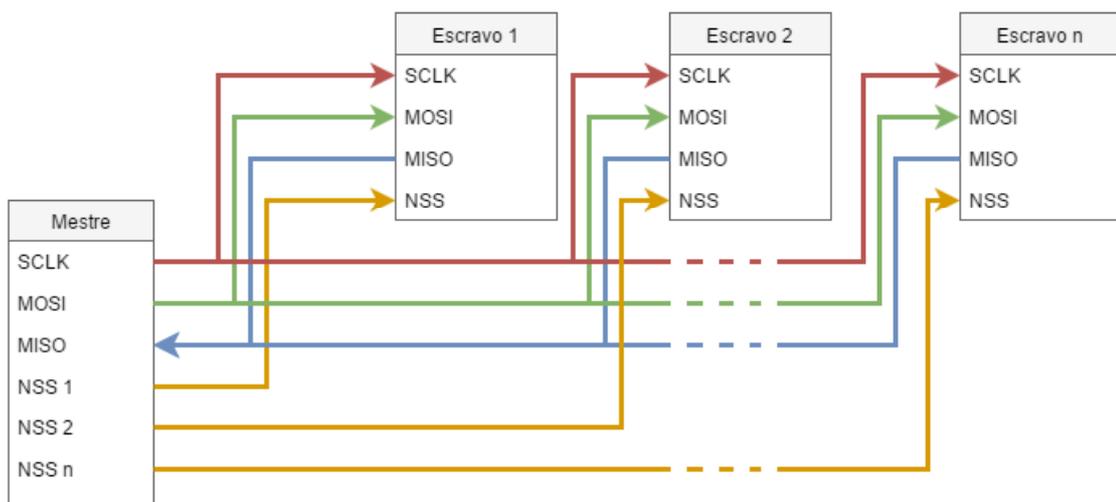


Figura 3.16 – Interface barramento SPI múltiplos escravos

No entanto, este método para grandes números de escravos utiliza demasiados GPIOs (*General Purpose Input Output*). A Figura 3.17 representa uma alternativa a este método, utilizando apenas uma linha de sinal NSS.

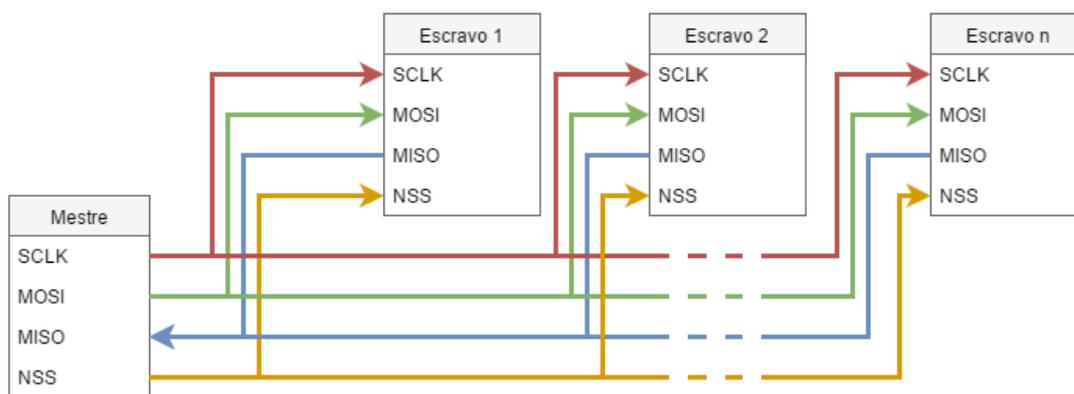


Figura 3.17 – Interface barramento SPI com múltiplos escravos, Daisy Chain

Esta configuração é designada por *Daisy Chain*, neste tipo de ligação os dados passam de um escravo para o próximo, portanto, para enviar dados para qualquer escravo, é necessário transmitir dados suficientes para chegar a todos. Além disso, os primeiros dados transmitidos vão para o último escravo. Se for necessário enviar dados para o dispositivo mestre, se os dados forem enviados pelo escravo número um, os dados terão de passar por todos os escravos seguintes, antes de chegar ao dispositivo mestre. Este tipo de disposição é normalmente utilizado em situações somente de saída, onde não é necessário receber quaisquer dados de volta.

3.4. Sinais e Grandezas

Esta secção é feita uma pequena introdução aos tipos de sinais e grandezas a adquirir neste projeto, evidenciando as principais dificuldades e métodos de aquisição de cada tipo de sinal.

3.4.1. Sinais Bio Potenciais

Uma parte crítica e importante do diagnóstico médico é a monitorização de sinais bio potenciais. Muitos dos órgãos do corpo humano, tais como o coração, cérebro, músculos e os olhos manifestam a sua função através da atividade elétrica (L. A. Geddes, 1989). O coração, por exemplo, produz um sinal de chamado eletrocardiograma (ECG). Na Figura 3.18 estão representados exemplos para os três tipos de sinais de interesse para este projeto.

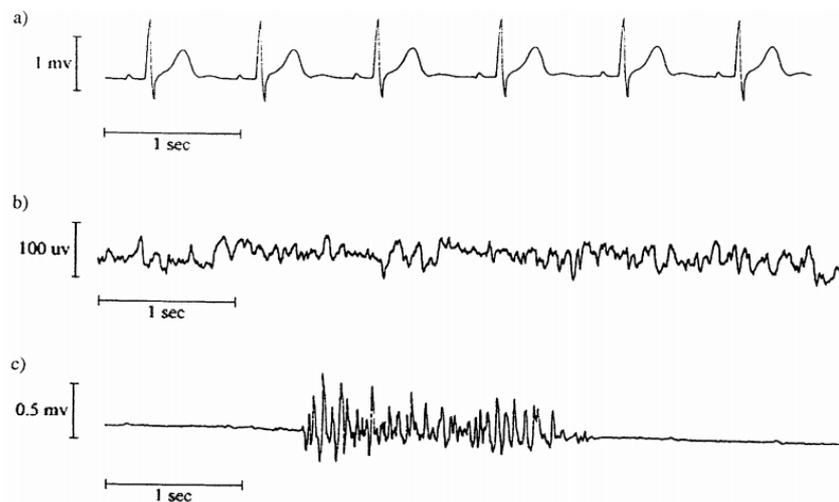


Figura 3.18 - Amostras dos sinais, (a) ECG; (b) EEG (c) EMG(Luca & G.De, 2000)

O cérebro produz um sinal de chamado eletroencefalograma (EEG) e a atividade dos músculos, como a contração e a relaxação, produzem um eletromiograma (EMG). As medições destes e outros sinais elétricos a partir do corpo podem proporcionar pistas para funções vitais como normais ou patológicas de órgãos. Por exemplo, batimentos cardíacos anormais ou arritmias podem ser facilmente diagnosticada a partir de um ECG. Os neurologistas podem interpretar sinais de EEG para identificar eventos ataques epiléticos. Os sinais EMG podem ser úteis na avaliação da função muscular, bem como doenças neuromusculares. As origens desses bio potenciais podem ser atribuídos à atividade elétrica ao nível celular (Plonsey, 1999). Os sinais bio potenciais provenientes dos órgãos são diversos, a Tabela 3.4 lista apenas os sinais de interesse para este projeto, e os seus principais índices de medida.

Tabela 3.4 – Propriedades dos sinais bio potenciais (Luca & G.De, 2000)

Tipo de Sinal	Amplitude em (mV)	Largura de Banda (Hz)
ECG	1 - 5	0.05 – 100
EEG	0.001 – 0.01	0.5 – 40
EMG	0.01 – 0.1	20 - 2000

As características mais notáveis de bio potenciais são: (L. A. Geddes, 1989)

- Amplitudes pequenas (10uV a 10mV)
- Gama baixa frequência de sinais

Os problemas mais notórios dessas aquisições são:

- Presença de interferência biológica (proveniente da pele, elétrodos, movimento)

- Ruído a partir de fontes ambientais (Linhas de energia, radio frequência, eletromagnética)

Estes desafios de aquisição de sinal e problemas para cada um dos bios potenciais são considerados em maior detalhe nas secções seguintes.

3.4.1.1. Eletrocardiograma

Os sinais de ECG são adquiridos, colocando elétrodos diretamente no tronco, braços e pernas, Figura 3.19. A atividade sobre a superfície do corpo, reflete a atividade do músculo cardíaco por baixo e na sua proximidade. Um sistema de elétrodos clinicamente aceite, é chamado de sistema de 12 derivações (G., 2000). Os sinais de ECG na superfície do corpo são pequenos em amplitude, o que torna as medições suscetíveis a erros (Webster, 1984), gerados pelo movimento relativo entre o elétrodo e a pele, bem como pela atividade dos músculos adjacentes. Uma consideração importante na boa aquisição do sinal do ECG é a utilização de elétrodos de alta qualidade (Thakor & Webster, 1985).

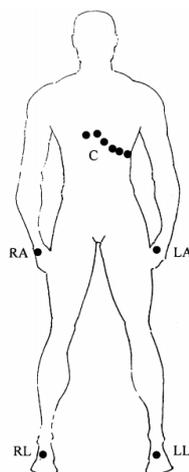


Figura 3.19 - ECG de 12 derivações é adquirido usando os elétrodos braço direito (RA), braço esquerdo (LA), perna esquerda (LL), referência perna direita (RL), e seis no peito (C).(Luca & G.De, 2000)

É recomendado o uso de elétrodos feitos de revestido com cloreto de prata ou de material de sintetizados Ag-AgCl prata. É utilizado um gel eletrolítico para melhorar a condução entre a pele e o elétrodo de metal de modo a reduzir os artefactos produzidos pelo contato elétrodo-pele, bem como a interferência eletromagnética de todas as fontes.

3.4.1.2. Eletromiograma

As fibras musculares geram atividade elétrica sempre que os músculos estão ativos (Luca & G.De, 2000). Os sinais EMG são registrados, colocando elétrodos próximos a um grupo muscular, por exemplo, um par de elétrodos colocados sobre os bíceps e outro par colocado sobre o tríceps pode capturar os sinais EMG gerados quando estes músculos contraem, Figura 3.20.

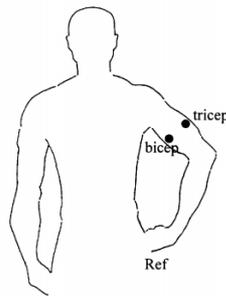


Figura 3.20 - Colocação dos eletrodos sobre os bíceps e tríceps com uma referência independente para aquisição de um EMG(Luca & G.De, 2000)

Os eletrodos usados para tais aplicações devem ser pequenos, bem seguros e devem fornecer aquisições livres de artefactos, estes podem ser de cloreto de prata-prata, banhados a ouro ou eletrodos de aço inoxidável. A aquisição atividade diretamente das fibras musculares si pode ser clinicamente valiosa na identificação de doenças neuromusculares.

3.4.1.3. Eletroencefalograma

Os sinais de EEG são caracterizados pelas suas amplitudes extremamente pequenas (na gama dos microvolts), os eletrodos a utilizar devem ser banhados a ouro e colocados muito firmemente no couro cabeludo para fazer um contato de muito baixa resistência. Um sistema de derivações clinicamente aceite, que inclui vários eletrodos colocados uniformemente em torno da cabeça, é o sistema de 10-20 derivações (Aminof, 2000).

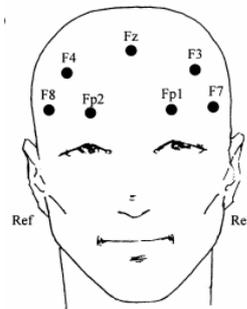


Figura 3.21 – Colocação dos eletrodos do sistema de 10-20 derivações utilizando as orelhas como referência(Luca & G.De, 2000)

Os sinais de EEG são de difícil interpretação, já que representam a atividade global dos neurónios, transmitidos através dos tecidos cerebrais, fluidos, e couro cabeludo (Niedermeyer & da Silva, 2005). Na própria forma de onda, é possível analisar, por exemplo, picos de ataques interictais ou uma convulsão completa (como pequeno mal e grande mal epilético) (Niedermeyer & da Silva, 2005). A análise do espectro do EEG na frequência pode revelar alterações na potência de sinal, em frequências diferentes, que são produzidas durante as várias fases do sono, assim como o resultado dos efeitos da anestesia, e por vezes, o resultado de uma lesão cerebral (Aminof, 2000). Os problemas práticos e desafios associados com aquisições do

sinal EEG surgem de fontes de ruído fisiológicos, ambientais e eletrônicos. Fontes fisiológicas de interferência são os artefactos do movimento, ruído muscular, movimentos oculares, e por vezes até mesmo sinais de pulsação. A interferência elétrica surge a partir das fontes habituais, como as linhas elétricas 50 Hz, frequências de rádio (RF), e a interferência induzida eletricamente ou magneticamente. Além disso, os componentes eletrônicos do circuito amplificador também podem contribuir com ruído, por nos componentes integram a fonte de alimentação do mesmo. No entanto, a utilização de algumas técnicas de projeto e de medição podem mitigar os efeitos de tais ruídos e interferências.

3.4.1.4. SpO2

Para os doentes em risco de insuficiência respiratória, é importante controlar a eficiência da troca gasosa nos pulmões, isto é, quão bem o sangue arterial é oxigenado. De preferência, tais informações devem estar disponíveis para os médicos numa base contínua. Ambos estes requisitos podem ser satisfeitos de forma não invasiva com a tecnologia de oximetria de pulso. A técnica está bem estabelecida e é um recurso regular durante anestesia ou terapia intensiva. A oximetria de pulso também é utilizada na monitorização de doenças pulmonares em adultos e na investigação de distúrbios do sono. O princípio de operação da oximetria de pulso é baseado no facto de que a componente de sangue arterial é pulsátil por natureza (varia com o tempo). Então quando a luz de um LED é incidida no corpo humano (por exemplo, no dedo), a quantidade de luz que passa através do dedo, após a atenuação de vários componentes, como por exemplo, o tecido, artérias e veias, também contém uma componente pulsátil, sobre uma componente constante. O objetivo da oximetria de pulso é medir a percentagem de HbO₂ (hemoglobina oxigenada) para o total de Hb (Hemoglobina oxigenada mais a desoxigenada) no sangue arterial, isto é referido como SPO₂. A hemoglobina oxigenada no sangue é distintivamente vermelha, onde a hemoglobina desoxigenada no sangue tem uma colocação característica azul escuro. As propriedades óticas, do sangue, no espectro visível, encontram-se entre 400 e 700 nm, e perto do espectro de infravermelho, entre 700 e 1000nm, as regiões espectrais dependem da quantidade de oxigénio que é transportado pela hemoglobina.

O método do SPO₂ tira partido do facto, da Hb ter um coeficiente de absorção na região vermelha do espectro, cerca de 660nm, quando comparada com o HbO₂, como se encontra representado na Figura 3.22.

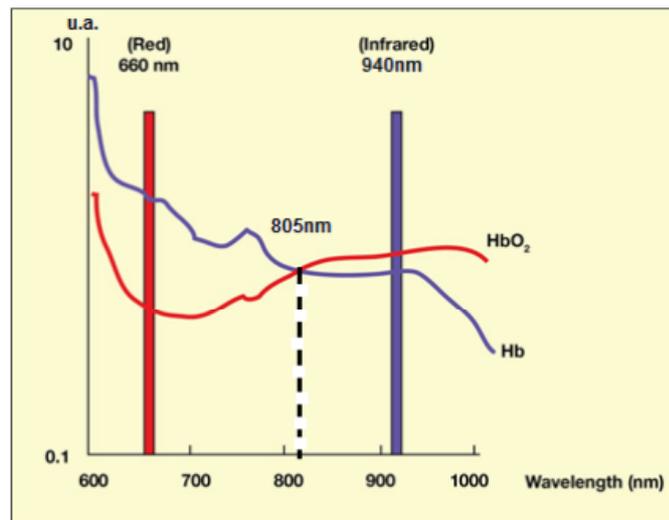


Figura 3.22 – Absorção da luz infravermelha e Vermelha no sangue oxigenado versus sangue desoxigenado. (Texas Instruments, 2014b)

O sistema ótico de medição para a medição do SPO2 consiste em dois LEDs que emitem luz e um foto díodo que recebe essa luz, o foto díodo converte a luz incidente num sinal elétrico que é proporcional á intensidade de luz. Este sinal é denominado de PPG (Photoplethysmogram), e contém a periodicidade da frequência de pulsação. A magnitude do sinal PPG depende da quantidade de sangue que é injetado a partir do coração, a cada ciclo sistólico, da absorção ótica do sangue, da pele e dos vários componentes dos tecidos.(Texas Instruments, 2014b)

Durante a sístole, quando a pulsação arterial está no seu máximo, o volume de sangue nos tecidos aumenta. O sangue adicional absorve mais luz, reduzindo a intensidade de luz que é deflectida. Durante a diástole, está presente menos sangue nos tecidos, aumentando a intensidade de luz deflectida. A parte pulsátil do sinal PPG é considerada como uma componente AC, e a parte não pulsátil é considerada uma componente DC. O desvio na intensidade do LED ou um desvio na sensibilidade do detetor podem alterar a intensidade de luz que é recebida pelo sensor. Esta dependência entre a intensidade da luz que é transmitida e deflectida, pode ser compensada pela utilização de uma técnica de normalização onde a componente AC é dividida pela componente DC, e é dada pela equação (1) indicada abaixo.

$$\frac{R}{IR} = \frac{AC_R}{DC_R} / \frac{AC_{IR}}{DC_{IR}} \quad (1)$$

Então a absorção é independente do tempo devido à absorção do sangue venoso, ou devido aos tecidos que os rodeiam, não tendo assim qualquer efeito na medição. A maior parte dos oxímetros de pulso medem a absorção nos dois tipos de comprimentos de onda e são calibrados mediante a utilização de oxímetro invasivos (SaO2). Esta calibração utiliza a equação (2) empírica indicada abaixo.

$$S_a O_2 \% = A - B * (R/IR) \quad (2)$$

Onde o (R/IR) é a normalização da componente pulsátil (AC) dividida pela correspondente componente não pulsátil (DC) de cada comprimento de onda, e o A e o B são coeficientes de regressão linear os quais estão relacionados com coeficientes de absorção linear específicos da Hb e da HbO₂. Estas constantes são determinadas empiricamente durante a calibração *in-vivo* pela correlação entre o rácio calculado pelo oxímetro de pulso contra as amostras de sangue arterial obtidas por um oxímetro invasivo (SaO₂). (Nellcor, 2003)

3.4.2. Sinais e Grandezas ambientais

As medições das grandezas ambientais, como a temperatura, a Humidade Relativa e a pressão atmosférica, podem fornecer informação extra acerca do ambiente em redor do paciente, durante a sua monitorização contínua. Todos os anos um grande número de hospitalizações e mortes ocorrem com associação à exposição a extremos de temperaturas ambiente. Existe evidencia de que a taxa de mortalidade aumenta em e imediatamente depois de dias muito frios ou muito quentes, tendo sido realizados vários estudos que relacionam a exposição à alta temperatura a uma mortalidade mais elevada (Basu & Samet, 2002)(Basu, 2009). A aquisição destes dados torna-se vantajosa pois pode fornecer informação destas grandezas de modo a realizar o ajuste desse mesmo ambiente, quando possível, ou apenas para fornecer o alerta de que as condições ambientais não são as ideais.

3.4.2.1. Temperatura e Humidade Relativa

O teor de água em torno do ar é um fator importante para o bem-estar dos seres humanos e animais. O nível de conforto é determinado por uma combinação de dois fatores:

- Humidade relativa
- Temperatura ambiente

A temperatura é uma medida da energia cinética numa amostra de material e é expressa em unidades de graus numa escala padrão. A humidade pode ser medida por instrumentos chamados higrómetros. O termo humidade refere-se ao teor de vapor de água nos gases. A humidade relativa é definida como a razão entre a pressão atual do vapor de ar a qualquer temperatura até ao máximo da saturação da pressão do vapor à mesma temperatura. Humidade relativa (Norton, 1996) em percentagem é definido como:

$$Hr(\%) = 100 \frac{p_w}{p_s} \quad (3)$$

Onde P_w é a pressão parcial do vapor de água e o P_s é a pressão do vapor de água saturado a uma determinada temperatura. O valor H exprime o conteúdo de vapor de água como a percentagem de concentração necessária para causar a saturação do vapor, isto é a formação de gotas de água a essa temperatura.

4. Desenvolvimento de Hardware

Este capítulo descreve a seleção de *hardware* e o desenvolvimento da plataforma de hardware necessária para a aquisição das grandezas ambientais e bio potenciais. A seleção do hardware a utilizar, sempre que possível, recaiu sobre *hardware* que disponibilizam placas de desenvolvimento ou placas de *Break-out*, deste modo é possível acelerar o desenvolvimento da plataforma. Caso contrário seria necessário desenvolver um protótipo de *hardware* que com todos os componentes a utilizar.

A arquitetura da plataforma de hardware, define três tipos de dispositivos, um dispositivo *Bluetooth* que atua como dispositivo central na rede *Bluetooth*, este dispositivo poderá ser por exemplo a interface *Bluetooth* disponível na maioria dos *smartphones*, ou computadores. O dispositivo central tem como função receber os dados provenientes dos dispositivos *Bluetooth* periféricos (*Peripheral*). Além deste dispositivo são considerados dois outros tipos de dispositivos que atuam como dispositivo *Bluetooth* periférico e realizam a recolha de dados. A Figura 4.1 representa a arquitetura de *hardware* de um dos dispositivos periféricos, onde este contém apenas um módulo *Bluetooth* e sensores com comunicação I2C

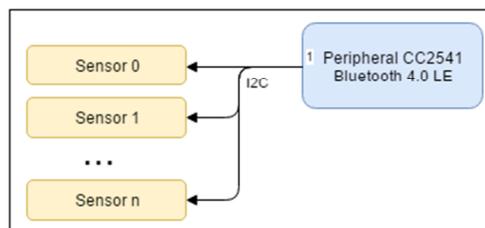


Figura 4.1 – Arquitetura da plataforma de hardware, dispositivo periférico apenas com sensores

A Figura 4.2 representa a arquitetura do segundo dispositivo periférico. Este segundo dispositivo será descrito e maior detalhe nas próximas secções devido á sua maior complexidade. Este dispositivo tem presente um módulo de aquisição de sinais bio potenciais configurável, permitindo assim adquirir vários tipos de sinais bio potenciais com características diferentes. Além da aquisição dos sinais bio potenciais este dispositivo suporta também a leitura de sensores através do barramento I2C e o controlo dos Leds utilizados em sensores de SPO2. Em termos de comunicação este dispositivo suporta comunicação através de *Bluetooth* e UART.

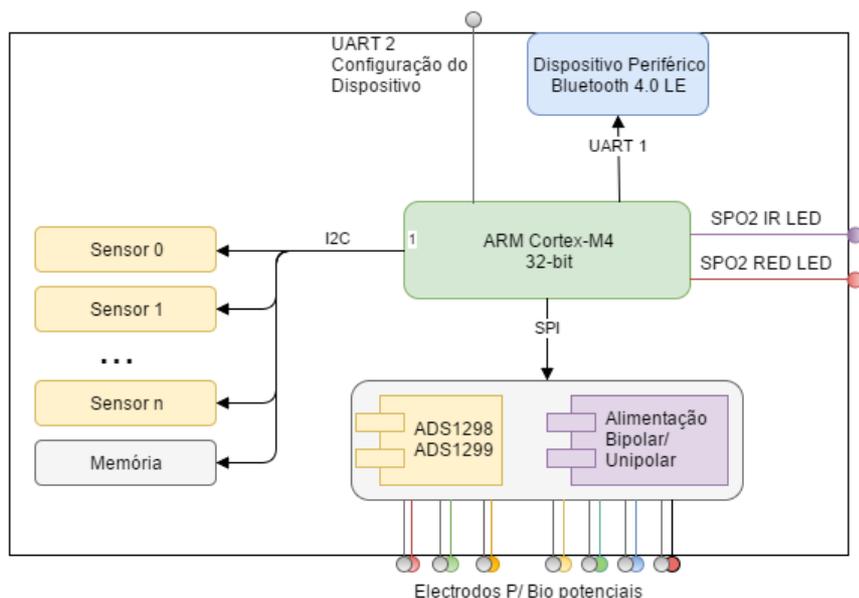


Figura 4.2 - Arquitetura da plataforma de hardware, para aquisição de sinais Bio potenciais

As próximas secções descrevem os vários estágios de desenvolvimento da plataforma do protótipo realizado neste projeto.

4.1. Sensores

Nesta secção são descritas as principais características dos sensores que foram selecionados para realizar este projeto. Para os sensores de monitorização ambiental foi possível encontrar, placas de desenvolvimento com características relevantes, para este projeto. Estes sensores de monitorização ambiental utilizam a interface I2C, possibilitando assim a interligação entre o microprocessador e estes sensores com apenas quatro ligações.

4.1.1. Sensor de Temperatura

O sensor de temperatura selecionado foi o sensor TMP102 (Texas Instruments, n.d.-b), da *Texas Instruments*. E encontra-se disponível na placa de desenvolvimento da Figura 4.3.



Figura 4.3 - Placa de desenvolvimento TMP102

Este sensor da temperatura ambiente, é capaz de detetar variações de $.0625^{\circ}\text{C}$, e tem uma resolução de 12bits, a sua gama de funcionamento fica entre -25 e $+85^{\circ}\text{C}$. O seu consumo de

energia é de apenas 10uA, o que faz com que este seja um sensor ideal para aplicações portáteis, com alimentação a bateria. O sensor TMP102 é um dispositivo com interface I2C, e utiliza o pino (ADD0) para selecionar o seu endereço no barramento I2C. Os endereços I2C disponíveis para este tipo de sensor encontram-se disponíveis na Tabela 4.1.

Tabela 4.1 - Endereços sensor de temperatura TMP102 (Texas Instruments, n.d.-b)

Endereço TMP102	Ligação do Terminal ADD0
0x48	GND
0x49	VCC
0x4A	SDA
0x4B	SCL

A Figura 4.4 representa a sequência de transmissões I2C que efetuadas entre o MCU (*Microcontroller Unit*) e o sensor para o MCU obter o valor da temperatura ambiente. São apenas necessárias duas transmissões I2C para obter o valor da temperatura.

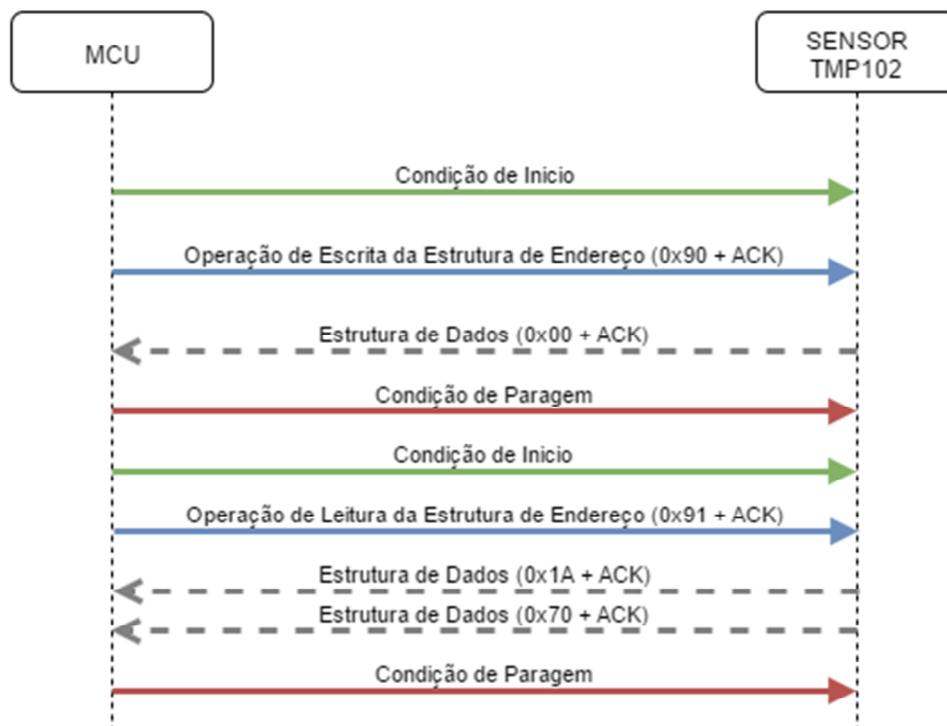


Figura 4.4 - Sequência de transições I2C, TMP102

4.1.2. Sensor de Pressão Atmosférica

O sensor de medição da pressão atmosférica selecionado foi o BMP180 (BOSCH, n.d.), disponibilizado pela *BOSCH*. Existe uma placa de desenvolvimento para este sensor que se ilustra na Figura 4.5.



Figura 4.5 - Placa de desenvolvimento BMP180

Este sensor é um barômetro digital, de baixo consumo de energético, oferece uma gama de medição pressão de 300 a 1100 hPa (*hectopascal*) com uma precisão até 0,02 hPa em modo de resolução máxima. É baseado na tecnologia piezo-resistiva de alta precisão, é um sensor robusto e com grande estabilidade. Este sensor utiliza também a interface I2C para enviar e receber dados. Para poder obter a medição da pressão atmosférica é necessário realizar uma leitura dos parâmetros de calibração do sensor a Figura 4.6.

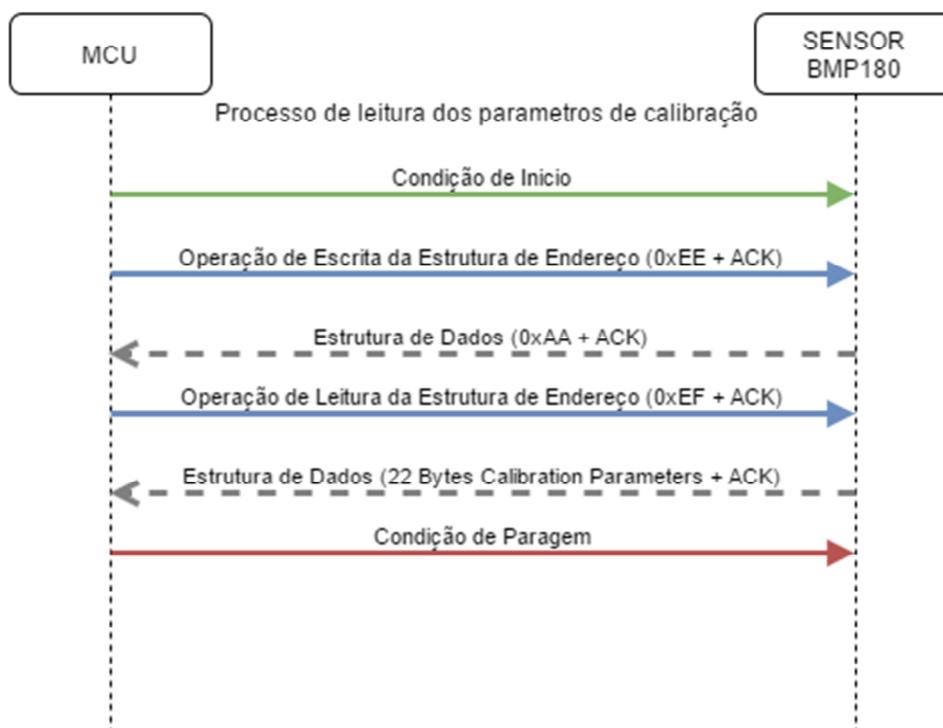


Figura 4.6 – Sequência de leitura dos parâmetros de calibração do sensor BMP180

Este sensor além do valor da pressão atmosférica fornece também o valor da temperatura, no entanto, a leitura do valor da temperatura não é direta, o valor lido é um valor

que não está calibrado, o valor real da temperatura é calculado á custa, de alguns parâmetros de calibração. A sequência de transmissões I2C necessárias para obter o valor da temperatura do sensor BMP180 está representado na Figura 4.7. Para obter o valor da pressão atmosférica real é necessário realizar também um processo de calibração, após a obtenção do valor da pressão atmosférica não regulado, a partir do sensor. Esta sequência de transmissões I2C é ilustrada na Figura 4.8.

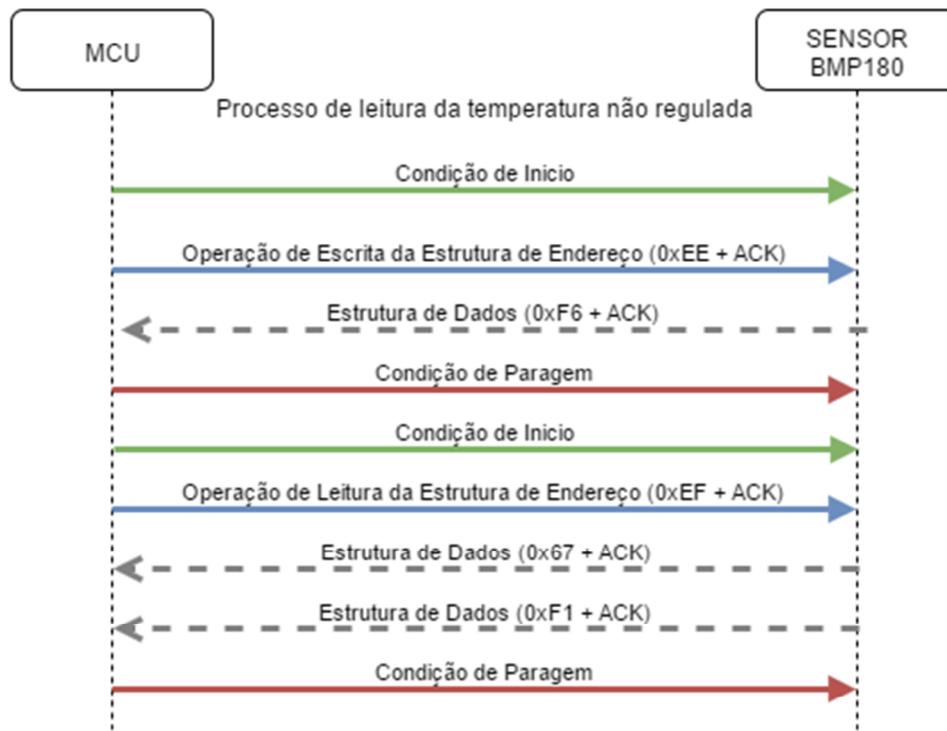


Figura 4.7 – Leitura da temperatura não regulada do sensor BMP180

o resultado com o que é enviado pelo sensor na estrutura de dados I2C, se os dois valores forem iguais, os dados recebidos são válidos, caso contrário, o valor lido não deve ser considerado válido. O mesmo processo deve ser repetido quando é realizada uma leitura da humidade relativa. A sequência de transmissões I2C para realizar uma leitura da humidade relativa encontra-se representada na Figura 4.11.

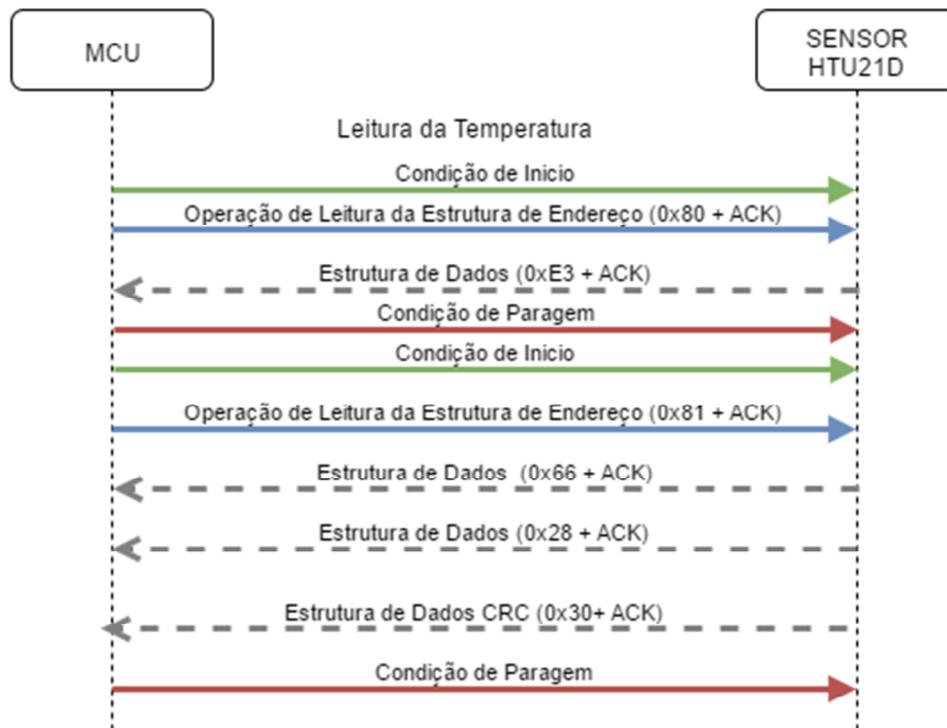


Figura 4.10 – Sequência de transmissões I2C, leitura da temperatura HTU21D

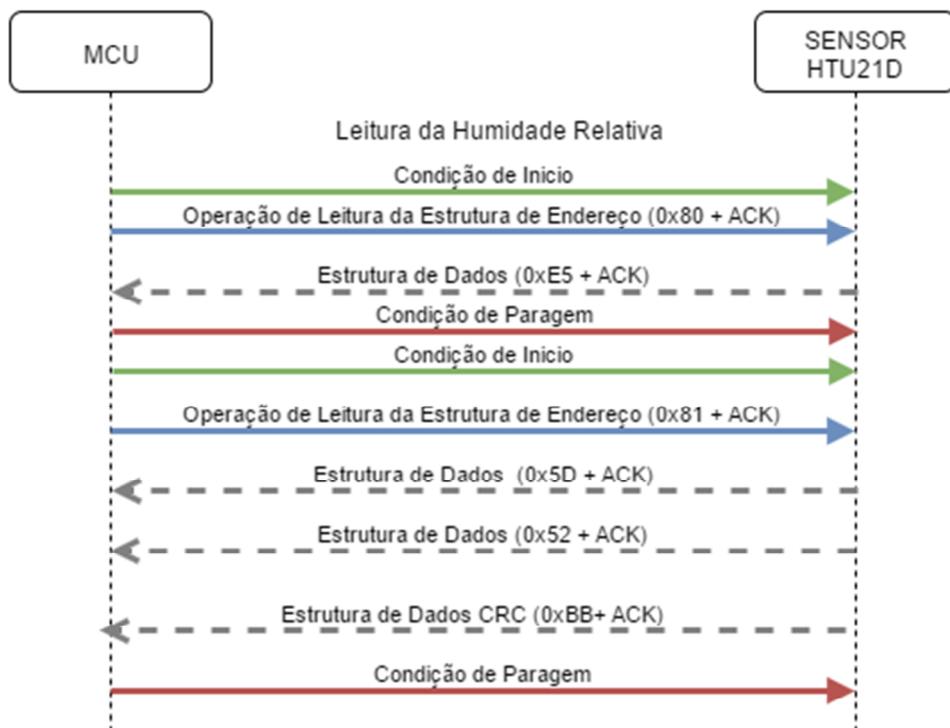


Figura 4.11 – Sequência de transmissões I2C, leitura da umidade relativa HTU21D

Após a execução das sequências representadas nas Figura 4.10 e Figura 4.11 e caso a recepção do CRC seja validado, é possível obter uma leitura da temperatura ambiente e da umidade relativa.

4.1.3. Oxímetro de Pulso

O sensor selecionado para realizar a medição de SPO₂, foi o *Nellcor DS-100A* este sensor é reutilizável e pode ser utilizado para a monitorização da perfusão normal de pulso. As ligações deste sensor são compatíveis com a maioria dos modelos utilizados por vários fabricantes de sistemas de monitorização de pacientes.



Figura 4.12 - Oxímetro de pulso NellCor DS-100A

Um sensor SPO₂ contém dois leds e um foto díodo, estes são colocados no dedo do paciente. Um dos leds emite uma luz vermelha, e o outro emite uma luz perto do infravermelho. Os LED's são controlados através de dois sinais, os tempos para ambos os sinais estão representados na Figura 4.13.

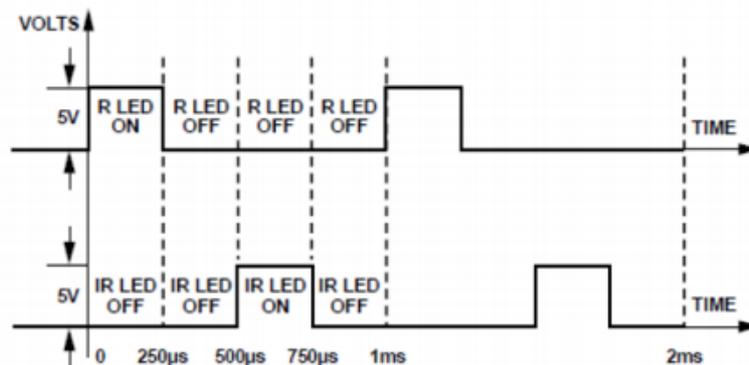


Figura 4.13 - Sinais de controlo do Oxímetro de pulso

O controlo destes sinais é feito através de dois GPIO's do microcontrolador. Estes GPIO's são capazes de fornecer 20mA de corrente, não sendo por isso necessário a utilização de drivers de corrente para a alimentação destes dois LED's. O detetor está sincronizado para capturar a luz de cada LED uma vez que é transmitida através do tecido. A luz que incide no foto díodo produz uma pequena corrente que é convertida por um amplificador de transimpedância, Figura 4.14.

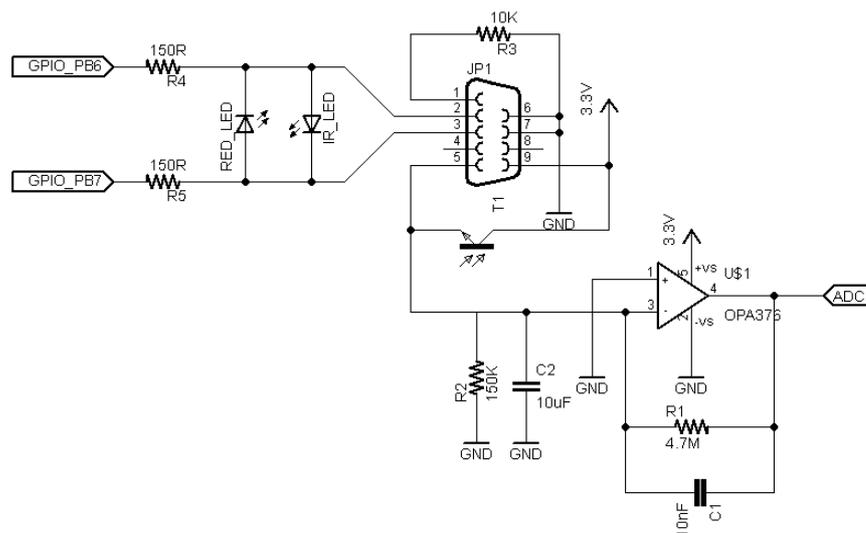


Figura 4.14 - Esquema de ligação oxímetro de pulso

Um amplificador de transimpedância, converte corrente para tensão. A sua entrada idealmente tem uma impedância igual a zero e a sua saída pode ter baixa impedância. Como o sinal de saída é expresso em tensão e o sinal de entrada é em corrente, o ganho, ou rácio da entrada saída, é expresso em ohms. A resistência R1, do circuito representado na Figura 4.14, é referida como transresistência. Como a entrada do amplificador operacional tem a corrente igual a zero, toda a corrente que entra no nó tem de fluir através de R1. Assim sendo se a corrente que flui através da resistência R1 for denominada por I_1 , a tensão de saída é igual a $I_1 * R1$. A tensão de saída corresponde, à quantidade de fótons recebidos pelo foto díodo, que por sua vez converte estes fótons numa corrente, essa corrente é convertida pelo amplificador de

transimpedância para uma tensão. A qual poderá ser capturada por um conversor analógico-digital (ADC).

4.1.4. Aquisição de Sinais Bio Potenciais

Um dos blocos de *hardware* necessário para este projeto é um bloco capaz de realizar aquisições de múltiplos sinais bio potenciais, de forma simultânea, estes sinais são caracterizados por possuírem amplitudes da ordem dos milivolts, portanto torna-se necessário que o bloco de aquisição seja capaz de realizar amostragem com boa resolução. Esta secção descreve o bloco de hardware para aquisição de bio potenciais, estes módulos de conversão de analógico para digital são normalmente referidos como AFE (*Analog Front End*). O componente principal deste bloco é o ADS1298/9 (Texas Instruments, 2013), produzido pela *Texas Instruments*. O ADS1299 conversor analógico-digital (ADC) de 24 bits, delta-sigma ($\Delta\Sigma$) com um baixo nível de ruído, 8 canais capazes de realizar amostragens simultâneas, um amplificador embutido de ganho programável (PGA), referência de tensão e referência de relógio internos. Este componente é especialmente produzido com características necessárias para a medição de bio potenciais permitindo assim a criação de sistemas de instrumentação médica escaláveis com um tamanho reduzido, baixo custo e baixo consumo energético. A Figura 4.15 contem uma representação simplificada dos blocos internos do ADS1299.

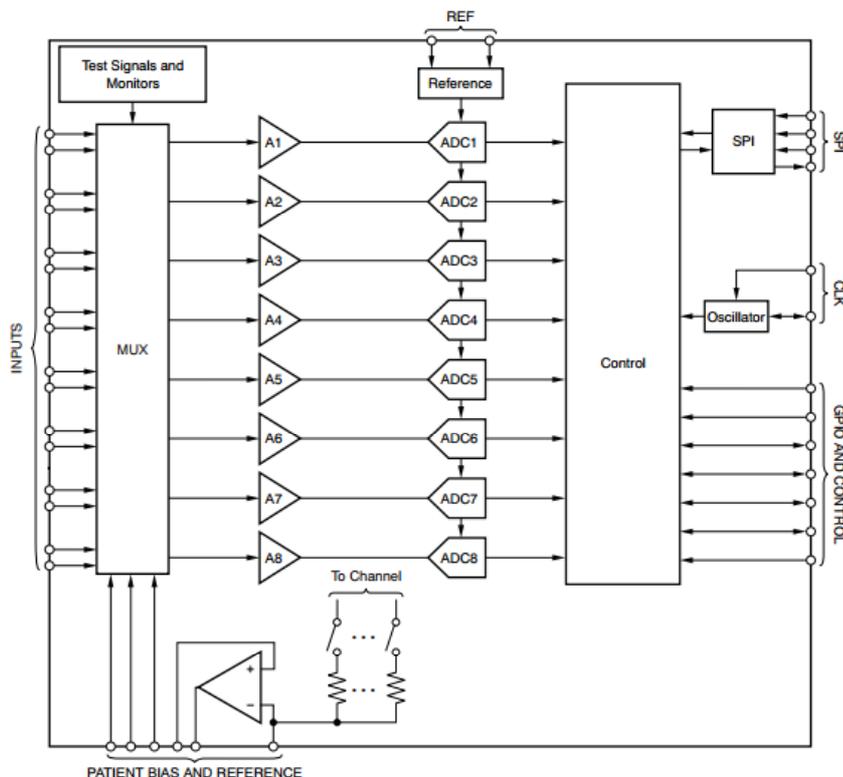


Figura 4.15 - Diagrama de Blocos internos ADS1299 (Texas Instruments, 2013)

O ADS1299 tem um *multiplexer* de entrada por canal que pode ser ligado de forma independente a sinais gerados internamente para teste, temperatura e detecção da ligação dos

elétrodos. (*Lead-off Detection*), sendo, portanto, bastante flexível. Adicionalmente, o *multiplexer* permite quaisquer elétrodos de entrada serem programados como unidade de referência do paciente (*Patient Bias Output Signal*). O ADS1299 opera a taxas de transferência de 250 amostras por segundo até 16 mil amostras por segundo. Outra vantagem deste dispositivo é que pode funcionar em modo de cascata para obter um maior número de canais de entrada, para isso o ADS1299 disponibiliza a opção de ligação com uma configuração *Daisy-Chain*. Além disso, este dispositivo integra várias funções específicas do EEG que o torna adequado para aplicações de eletroencefalografia (EEG) escalável. O dispositivo também pode ser utilizado como um sistema de aquisição de sinais analógicos de alta performance, multicanal, bastando para isso desligar o circuito específico do EEG, o que para este projeto constitui uma vantagem pois a aquisição do sinal proveniente do sensor SPO2, poderá ser convertido através do uso de um canal deste dispositivo, não necessitando assim de mais hardware externo. O ganho PGA pode ser escolhido a partir de uma das sete definições (1, 2, 4, 6, 8, 12 e 24). A comunicação para o dispositivo é realizada usando uma interface SPI. O dispositivo fornece quatro de entrada (GPIOs) para uso geral. Adicionalmente este dispositivo contém ainda uma referência de tensão interna pode ser programado para 4,5 V e um oscilador interno que gera um relógio de 2,048 MHz. O ADS1299 vem com um invólucro com a *footprint* TQFP64, sendo que por isso será necessário o desenvolvimento de uma placa de protótipo, para o bloco de hardware de aquisição de sinais bio potenciais. O circuito projetado para esta placa de aquisição encontra-se no Anexo I, nesta secção serão apenas mencionadas as partes mais relevantes, do processo de desenvolvimento do *hardware*. Como auxílio ao desenho de *hardware* foi utilizado o manual de utilizador do kit de demonstração (Texas Instruments, 2014a) deste dispositivo, o qual contém informação adicional sobre o funcionamento do dispositivo.

Para o desenho do esquema do circuito necessário para o ADS1299, e para o desenho do PCB (*Printed Circuit Board*) foi utilizado o *software Cadsoft Eagle 7.2.0*. (CadSoft Computer, n.d.). Foram criadas bibliotecas para este programa de modo a poder desenhar o esquema e o PCB.

4.1.4.1. Entradas Analógicas

As entradas analógicas dos ADS1299 são entradas diferenciais, e assumindo que o PGA tem o ganho de um, a entrada ($A_{INxP} - A_{INxN}$) pode variar entre as tensão de $-V_{ref}$ e $+V_{ref}$, Figura 4.17 e Figura 4.16. Existem dois métodos de utilização das entradas analógicas do ADS1299, ou em modo *Single-ended* ou em modo diferencial. Quando as entradas estão em modo *single-ended*, a entradas A_{INxN} é mantida no mesmo nível da tensão de modo comum (*Common-mode voltage*), que preferencialmente se situa a meio da tensão de alimentação. Quando a entrada é diferencial, a tensão de modo comum é dada por:

$$\frac{A_{INxP} - A_{INxN}}{2} \quad (4)$$

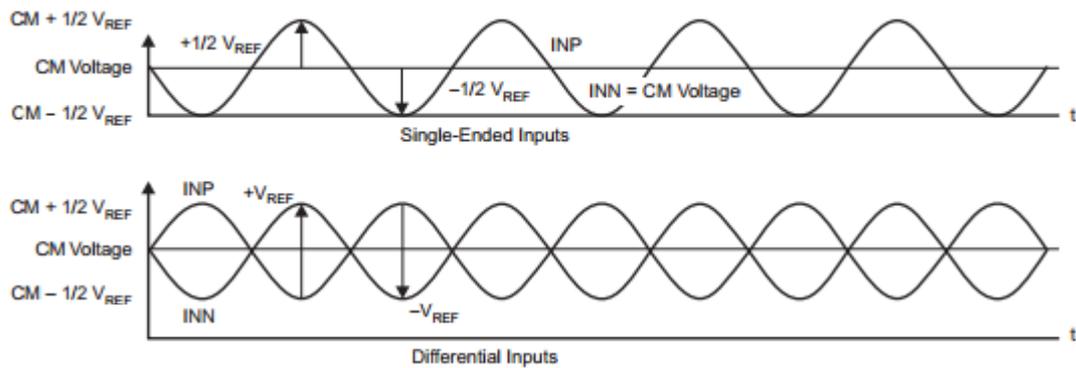


Figura 4.16 - Modos de operação das entradas analógicas ADS1299(Texas Instruments, 2013)

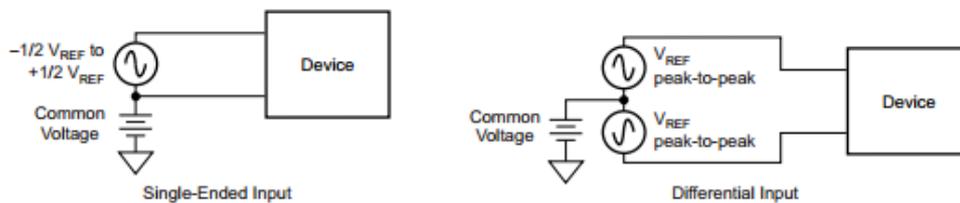


Figura 4.17 - Modos de operação das entradas analógicas (Texas Instruments, 2013)

4.1.4.2. Circuito de Alimentação

O ADS1299 tem três alimentações, o AVDD, o AVDD1, e o DVDD. O AVDD e AVDD1 devem conter o menor ruído possível. O AVDD1 fornece a alimentação ao bloco de *Charge Pump* interno do AD1299, e tem transitórios á frequência de oscilação do ADS1299 (fclk). Assim sendo, o AVDD1 e o AVSS1 devem estar ligados em estrela ao AVDD e ao AVSS, de forma a eliminar as diferenças entre os dois, por motivos de ruído. Cada alimentação deve ter um condensador cerâmico de 10uF (*microFarad*) e um de 0.1uF. Os condensadores utilizados para realizar a filtragem do ruído de alimentação (*decoupling*) são condensadores *multilayer* cerâmicos. O condensador VCAP1 terá de ser de tântalo, uma vez que a placa poderá estar sujeita a frequência de baixa vibração. Os dielétricos dos condensadores devem ser de classe 2 ou 3 (X7R, X5R). O ADS1299 pode ser alimentado com uma tensão unipolar ou bipolar, os esquemas recomendados, para o circuito de alimentação estão representados nas Figura 4.18 e Figura 4.19, respetivamente.

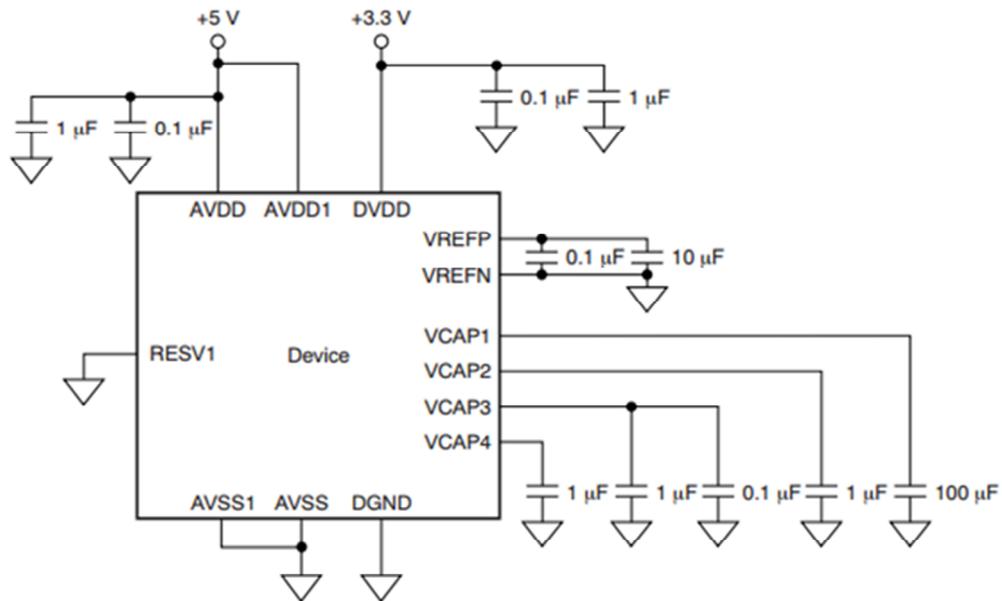


Figura 4.18 - Esquema de alimentação, para a operação com alimentação unipolar (Texas Instruments, 2013)

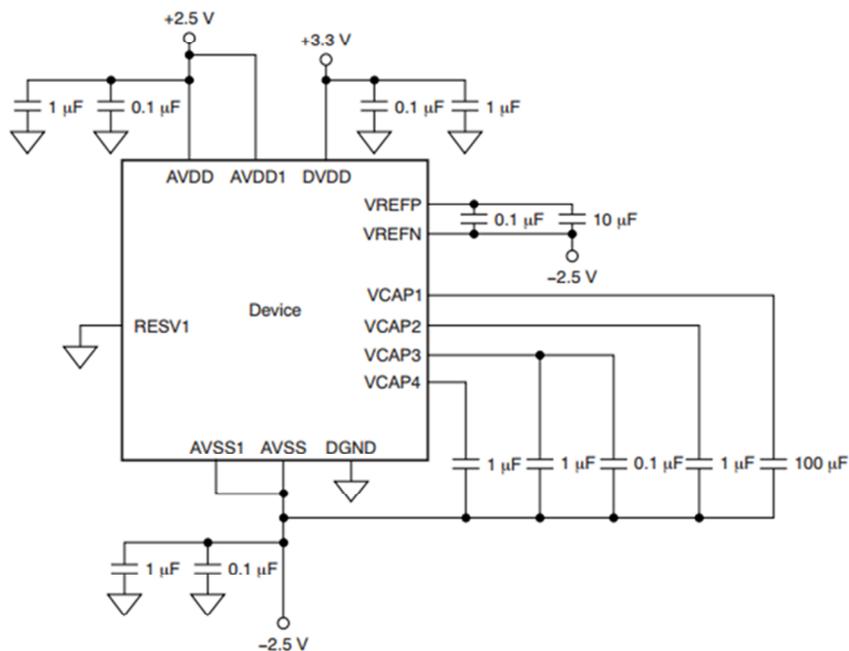


Figura 4.19 - Esquema de alimentação, para a operação com alimentação bipolar (Texas Instruments, 2013)

No PCB será possível realizar a seleção do tipo de alimentação, a seleção é feita mediante o posicionamento de duas resistências de 0R. Para obter uma alimentação unipolar de 0 a 5V, devem ser colocadas as resistências R17 e R18 e remover as resistências R16 e R19. Para obter uma alimentação bipolar de -2.5V a 2.5V, deve ser feito o contrário, as resistências

R16 e R19 devem estar colocadas e as resistências R17 e R18 devem ser removidas, a Figura 4.20 tem representado o circuito que permite realizar esta seleção. Esta seleção será útil para permitir avaliar qual o melhor método de alimentação para o circuito, na versão final esta seleção poderá ser removida.

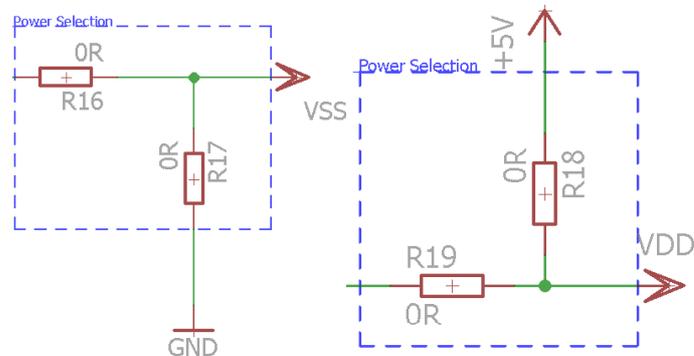


Figura 4.20 - Circuito de seleção de alimentação Unipolar ou Bipolar

O circuito de alimentação necessita de ser alimentado por tensões positivas de 5V e 3.3V. A conversão para alimentação unipolar é feita pelos componentes existentes na PCB, através do uso de dois reguladores de tensão o TPS73225 (Texas Instruments, n.d.-a), que fornece uma alimentação positiva de +2.5V e o TPS72325 (Texas Instruments, n.d.-a) que fornece uma alimentação negativa de -2.5V. Estes dois reguladores são reguladores *Low Drop Out* (LDO) e ambos são capazes de fornecer correntes até 250mA. As tensões fornecidas têm ruídos muito baixos, da ordem dos 60 μV_{rms} , sendo, portanto, ideais para a utilização em circuitos analógicos. Para fornecer a alimentação negativa na configuração de alimentação bipolar, foi necessário inverter a alimentação positiva de 5 volts, essa operação é garantida pelo TPS60403. Este dispositivo é um circuito designado de *charge pump*, a sua função neste circuito é inverter a tensão de alimentação que é aplicada à sua entrada, Figura 4.22. Utilizando o esquema representado na Figura 4.21 é possível perceber o princípio de operação deste dispositivo. Durante o primeiro meio ciclo, os interruptores S2 e S4 estão abertos, e os interruptores S1 e S3 fecham, e o condensador, C_{fly} (C1 no circuito, realizado neste trabalho), carrega até a tensão de entrada (V_i). Durante a segunda parte do ciclo, S1 e o S3 estão abertos e o S2 e S4 estão fechados. Isto liga o terminal positivo do C_{fly} ao GND e o negativo à saída (V_o). O condensador C_o (C12) é carregado negativamente através da sua ligação em paralelo com V_o . O valor da tensão à saída é mais positivo do que $-V_i$, uma vez que os interruptores S1 e o S4 tem resistência e a carga drena alguma energia do C_o (C12).

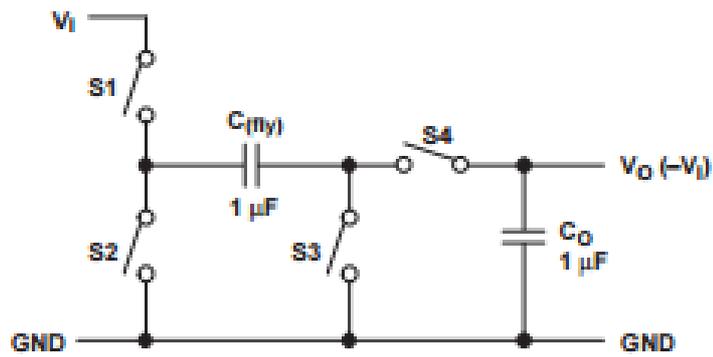


Figura 4.21 - Esquema do princípio de operação do dispositivo TPS60403 (Texas Instruments, 2015)

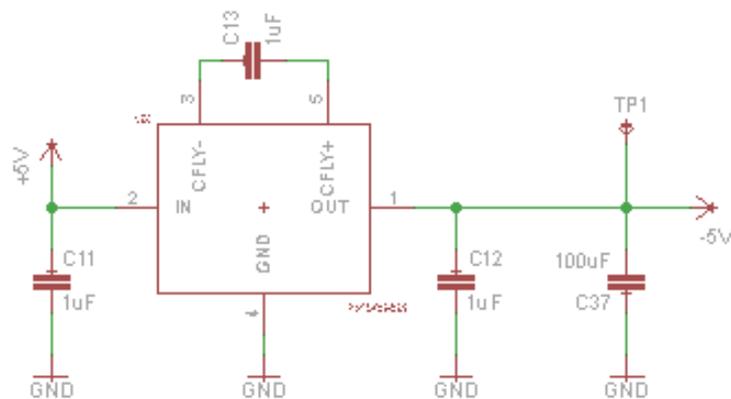


Figura 4.22 - TPS60403 Circuito implementado

4.1.4.3. Referência de Tensão Interna

O ADS1299 dispõe de duas formas para obter a tensão de referência interna. Esta referência pode ser gerada a partir de uma fonte externa ou gerada internamente pelo próprio ADS1299. Um esquema simplificado da referência interna do ADS1299 encontra-se ilustrado na Figura 4.23.

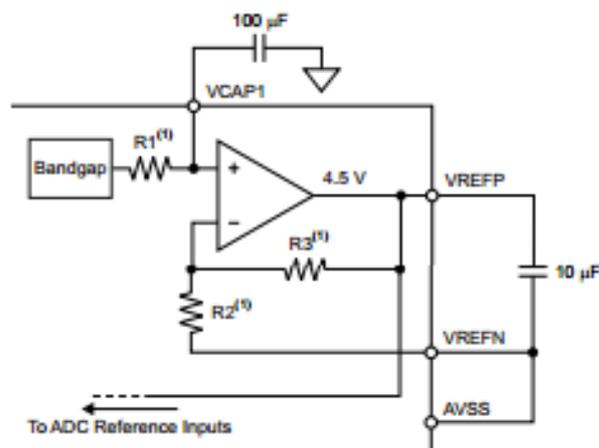


Figura 4.23 - Referência Interna, ADS1299 (Texas Instruments, 2013)

A referência interna é de 4.5V e é gerada relativamente ao AVSS, de modo a utilizar a referência interna o terminal VREFN deve estar ligado ao AVSS. Para esta aplicação, escolheu-se o uso da referência interna em detrimento da referência de tensão externa, devido a esta ser uma aplicação portátil alimentada a baterias e ao tamanho da placa final ser um fator a ter em conta, pois para gerar a tensão de referência externa seria necessário adicionar mais componentes, o que aumentaria o consumo de energia e o tamanho da PCB final. Ao iniciar o ADS1299 este acorda no modo de referência externa, pelo que é necessário configura-lo para utilizar a referência de tensão interna.

4.1.4.4. Referência de Oscilador

Tal como a referência de tensão, existem dois modos de fornecer o sinal de oscilação ao ADS1299, ou através do uso do oscilador interno, ou através do uso de um oscilador externo. O *clock* interno é ideal para circuitos de baixo consumo, alimentados a bateria como é o caso desta aplicação, no entanto a exatidão do sinal de oscilação varia com a temperatura, segundo as características elétricas do ADS1299, varia $\pm 2.5\%$ da frequência de *clock* gerada (2.048MHz), e para o efeito no circuito de teste foi colocada a *footprint* para a utilização de um cristal externo, se necessário. A seleção do *clock* é controlada pelo terminal CLKSEL e pelo bit CLK_EN do registo CONFIG1. A é uma tabela de verdade entre estes dois bits.

Tabela 4.2 - Configuração da referência do *clock* ADS1299

CLKSEL Pin	CONFIG1 CLK_EN Bit	Clock Source	CLK Pin Status
0	X	External Clock	Input
1	0	Internal Clock	3-State
1	1	Internal Clock	Output

No decorrer deste trabalho não foi possível, a produção da PCB desenhado para a aquisição de sinais bio potenciais. No entanto, de modo a ser possível obter uma visão geral do aspeto da placa e das suas dimensões, (6 cm x6 cm), foi criado um modelo da placa em 3D. As figuras resultantes são ilustradas, na Figura 4.24.

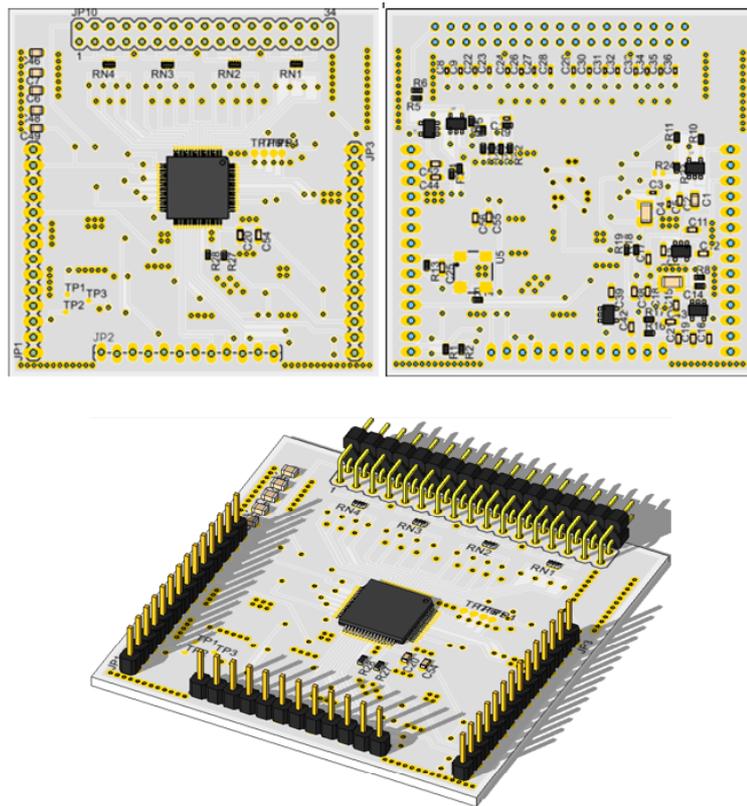
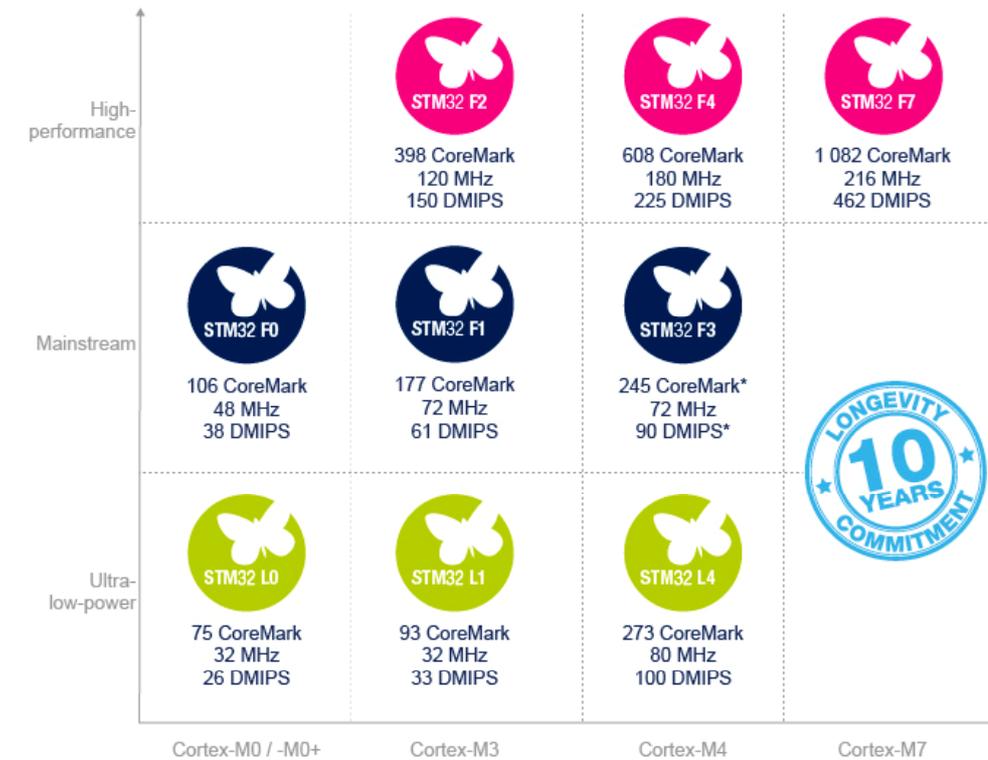


Figura 4.24 – Representação 3D da PCB de aquisição de bio potenciais

4.2. Microcontrolador

Para a criação do protótipo relativo a este projeto foi selecionado o microcontrolador STM32F401RE. Este microcontrolador encontra-se na gama de alta performance (*High-performance*), da família de microcontroladores STM32F4, da *ST Microelectronics*. Estes processadores são de 32 bits e são baseados no núcleo ARM Cortex®-M4, esta gama junta a alta performance com a capacidade de processamento de sinais digitais e baixo consumo de energia, com a operação em baixa tensão. A família STM32 será suportada durante os próximos dez anos, fornecendo, portanto, uma ótima base de hardware para este projeto. A família STM32 está dividida em 3 perfis, que se ilustram na Figura 4.25, onde cada perfil é otimizado para um determinado tipo de aplicação. O perfil definido pelas subfamílias STM32L0/1/4 é otimizado para aplicações de baixo consumo de energia. O perfil definido pelas subfamílias STM32F0/1/3 são otimizados para fornecer um equilíbrio entre boa performance e o baixo consumo energético.



* from CCM-SRAM

Figura 4.25 - Família de microcontroladores STM32 (ST Microelectronics, n.d.-b)

A família STM32F2/4/7 encontra-se otimizada para sistemas de alta performance, a capacidade de processamento seja importante.

A *ST Microelectronics* disponibiliza para a família de microcontroladores STM32, as placas de desenvolvimento da serie Núcleo. Estas placas de desenvolvimento tem um custo relativamente baixo, e um desenho de hardware que permite criar rapidamente protótipos com a um microprocessador STM32 MCU. Esta serie de placas de desenvolvimento partilha esquema de conectores com as placas *Arduíno*, o que facilita o uso de extensões de *hardware* (*Arduino Shields*). Além desses conectores compatíveis com *Arduíno*, contem outros, chamados de *ST Morpho*, que permite com que as placas STM32 Núcleo possam ser facilmente ser utilizadas num grande número de aplicações relacionados ao *hardware*. As placas STM32 Núcleo possuem com o *ST-Link* que é um *debugger*/programador integrado, fazendo com que não haja necessidade de ter um programador externo, reduzindo assim o custo de aquisição da plataforma de desenvolvimento.

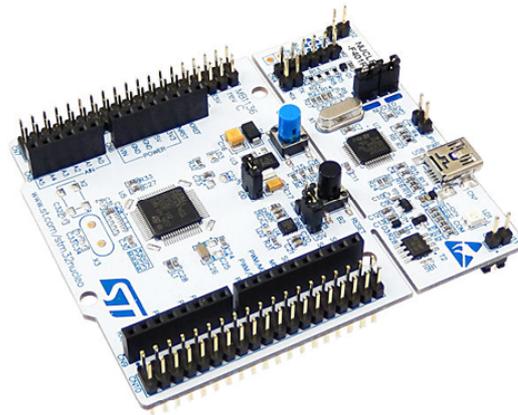


Figura 4.26 - Placa de desenvolvimento STM32F401RE Núcleo

A Figura 4.27 representa as características principais do microcontrolador selecionado para a realização deste projeto.

System	ART Accelerator™	512-Kbyte Flash memory	Control			
	ARM Cortex-M4 CPU 84 MHz	96-Kbyte SRAM		5x 16-bit timer		
		80-byte backup data		1x 16-bit motor control PWM synchronized AC timer		
		Connectivity	Floating Point Unit (FPU) Nested Vector Interrupt Controller (NVIC) JTAG/SW debug Embedded Trace Macrocell (ETM) Memory Protection Unit (MPU)	2x 32-bit timer		
				3x PC	Analog	
				3x USART LIN, smartcard, IrDA, modem control		
				4x SPI (2x with FS)		
				SDIO		
				USB 2.0 OTG FS		
				1.2 V internal regulator POR/PDR/PVD/BOR	AHB-Lite bus matrix	APB bus
		32 kHz + 4 ~26 MHz	Temperature sensor			
	32 kHz + 16 MHz					
	PLL					
	Clock control					
RTC/AWU						
2x watchdogs (independent and window)						
36/50/81 I/Os						
Cyclic Redundancy Check (CRC)						
96-bit unique ID						
Voltage scaling						

Figura 4.27 – Características STM32F401RE.

A Figura 4.28 representa a configuração de pinos necessária para a realização deste projeto.

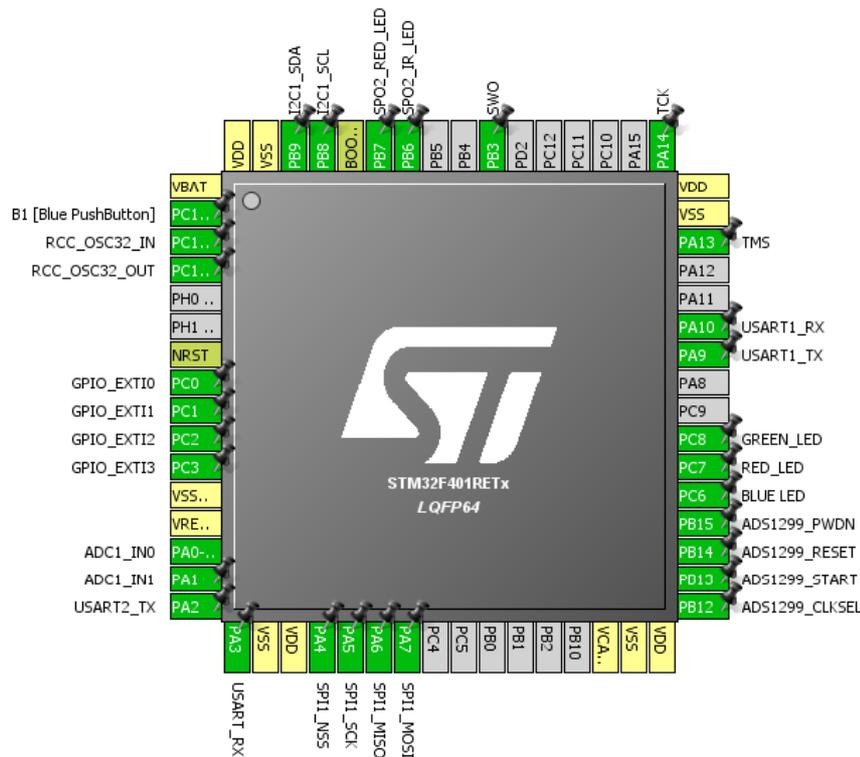


Figura 4.28 - Configuração de pinos utilizada

Atualmente a escolha de um microcontrolador para uma determinada aplicação enfrenta uma necessidade de fazer um compromisso entre a eficiência de energética e alta performance, especialmente com os recentes microcontroladores de alto desempenho, como os da série STM32F4. Esta família de microcontroladores assim como muitas outras implementa alguns modos de baixo consumo de modo a reduzir o consumo médio, durante o tempo de vida da aplicação. Após o *Reset* ou um *Power-on*, o microcontrolador STM32F4 fica no estado *Run mode*, neste modo o microprocessador fica totalmente ativo, consumindo assim muita energia mesmo quando as tarefas a executar são menos exigentes. Os modos de baixo consumo de energia são utilizados para economizar energia quando o processador não precisa de estar em funcionamento. A Figura 4.29, representa uma possível utilização dos modos de consumo de um microprocessador para reduzir o seu consumo de energia médio, ou seja, sempre que o processador necessitar de executar uma determinada função, o processador é colocado no modo ativo (*Run mode*), quanto maior for a sua frequência de operação maior será o seu consumo de energia, já nos modos de baixo consumo de energia o processador é desligado ou são apenas mantidas ativas as suas funções essenciais, o que reduz o seu consumo energético. Através de uma troca dinâmica dos modos de consumo energético, realizada por parte da aplicação de *firmware* é possível obter resultados médios de consumo menores.

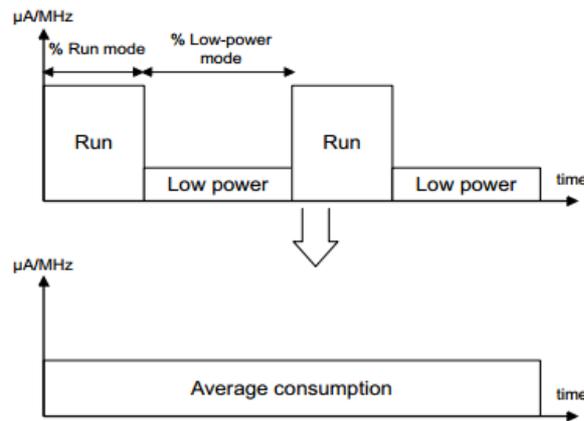


Figura 4.29 - Troca de modos de consumo de energia no microprocessador.

Os dispositivos STM32F4 tem quatro modos de gestão do consumo de energia:

- *Sleep mode*: O CPU e o oscilador de sistema são parados. Os periféricos são mantidos em execução.
- *Stop mode*: Este modo é o que proporciona menor consumo de energia, enquanto todos os registos da SRAM são mantidos. O PLL (Phase Locked Loop), HSI (High Speed Internal Oscilator), HSE (High Speed External Oscilator) são desativados. Todos os blocos, com alimentação de 1.2 volt são deligados.
- *Standby mode*: Consumo energético mais baixo possível. Todos os blocos, com alimentação de 1.2 volt são deligados. O conteúdo dos registos da SRAM são perdidos expeto os que se encontram na região de *Backup*.
- *VBAT mode*: A alimentação VDD é desligada. O circuito é alimentado através do pino VBAT deve estar ligado a uma fonte de alimentação externa (bateria).

Os consumos, em *Run mode* e em *Sleep mode* dependem da frequência de operação do microprocessador.

4.3. Módulos de comunicação Bluetooth 4.0

Os módulos de *Bluetooth* selecionados para uso neste projeto foram os módulos HM10/11, estes módulos tem o circuito integrado CC2541 e todo o circuito necessário para que este funcione corretamente. Estes módulos disponibilizam também acesso aos I/Os do circuito integrado CC2541. O módulo encontra-se representado na Figura 4.30.

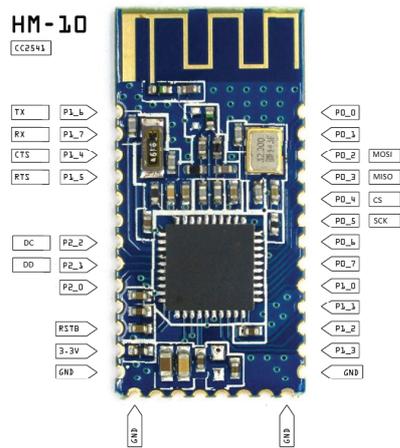


Figura 4.30 - Modulo HM-10 Bluetooth

Dependo do *firmware* utilizado estes módulos podem atuar como dispositivo central ou periférico numa rede *Bluetooth*. Outro módulo *Bluetooth* que foi utilizado é baseado no mesmo circuito integrado, o CC2141, no entanto este módulo fornece, uma entrada USB. Este módulo foi utilizado como o *hardware* de dispositivo central, Figura 4.31.

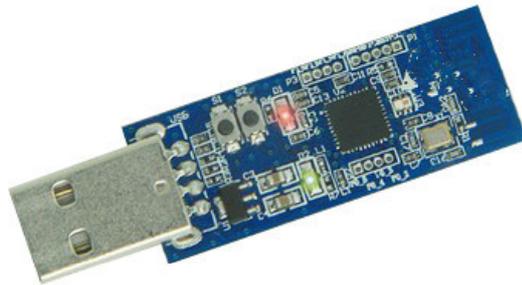


Figura 4.31 - Modulo Bluetooth para o dispositivo central

5. Desenvolvimento de Firmware

Nesta secção descreve a aplicação de *firmware* e a arquitetura implementada neste projeto. Este capítulo encontra-se dividido duas partes uma parte descreve a inicialização dos periféricos, utilizando o *software STM32CubeMX*, e a segunda parte a aplicação de *firmware* desenvolvida, utilizando o ambiente integrado de desenvolvimento *Eclipse Mars* e as ferramentas de desenvolvimento OpenSTM32 workbench. São ainda descritas as principais estruturas de código utilizadas no desenvolvimento da aplicação de *firmware*.

Todas as ferramentas de desenvolvimento de *firmware* são, ferramentas de código aberto (*Open source*) facilitando assim a sua utilização neste projeto.

5.1. Inicialização do Projeto e Periféricos

O conjunto de software *STM32Cube* disponibiliza uma ferramenta gráfica, o *STM32CubeMX* (ST Microelectronics, 2015d). Esta ferramenta ajuda o utilizador no processo de inicialização e configuração dos periféricos do microprocessador. A disponibilização deste tipo de ferramentas é muito vantajosa, pois não só proporciona uma familiarização mais rápida com o hardware, mas também reduz o tempo e esforço necessário na criação da aplicação, reduzindo assim os custos.

Ao utilizar a ferramenta *STM32CubeMX* para a criação de um novo projeto, é apresentado ao utilizador a interface, representada pela Figura 5.1, na qual este pode seleccionar e configurar quais os periféricos que pretende utilizar, assim como seleccionar os terminais de saída do microcontrolador para esse periférico.

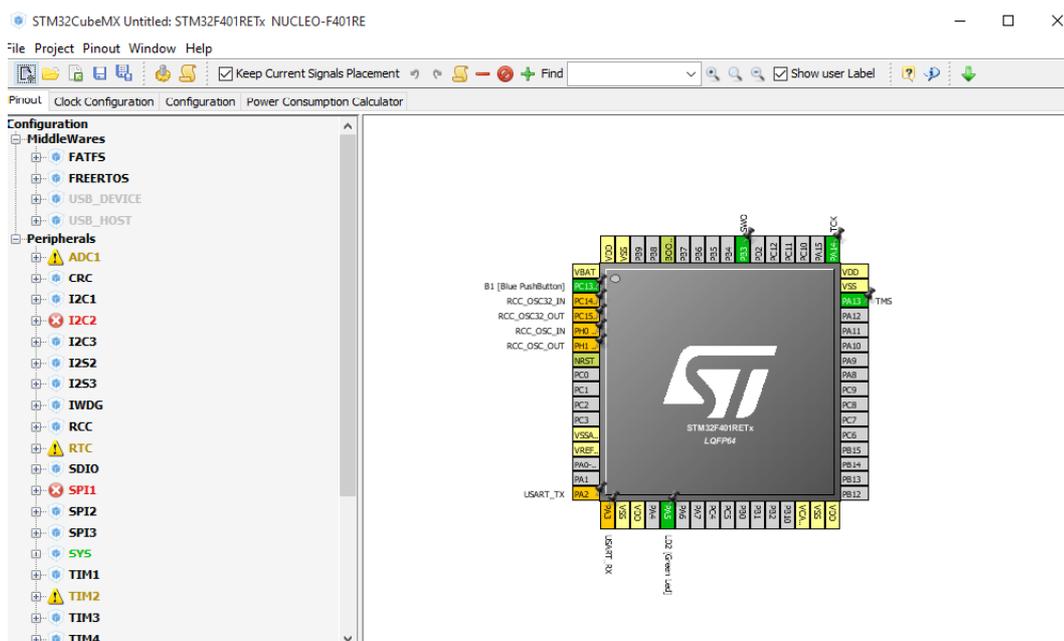


Figura 5.1 - Interface Gráfica STM32F4CubeMX

Outra funcionalidade que o *software* permite, é a configuração das fontes de sinal utilizadas para gerar o sinal de oscilador do sistema e dos periféricos. É possível selecionar como fontes de sinal o HSI (*High Speed Internal Oscilator*), HSE (*High Speed External Oscilator*) ou MCO (*Microcontroller Clock Output*) gerado pelo *ST-Link*.

Através de uma visualização geral da configuração dos PLL (*Phase Locked Loops*) e das fontes que geram os sinais do *oscilador*, (HSE, HSI ou MCO) é possível selecionar para obter a frequência de *oscilador* do processador pretendida. A configuração completa do *oscilador* do processador e dos periféricos, para o protótipo encontra-se ilustrado na Figura 5.2.

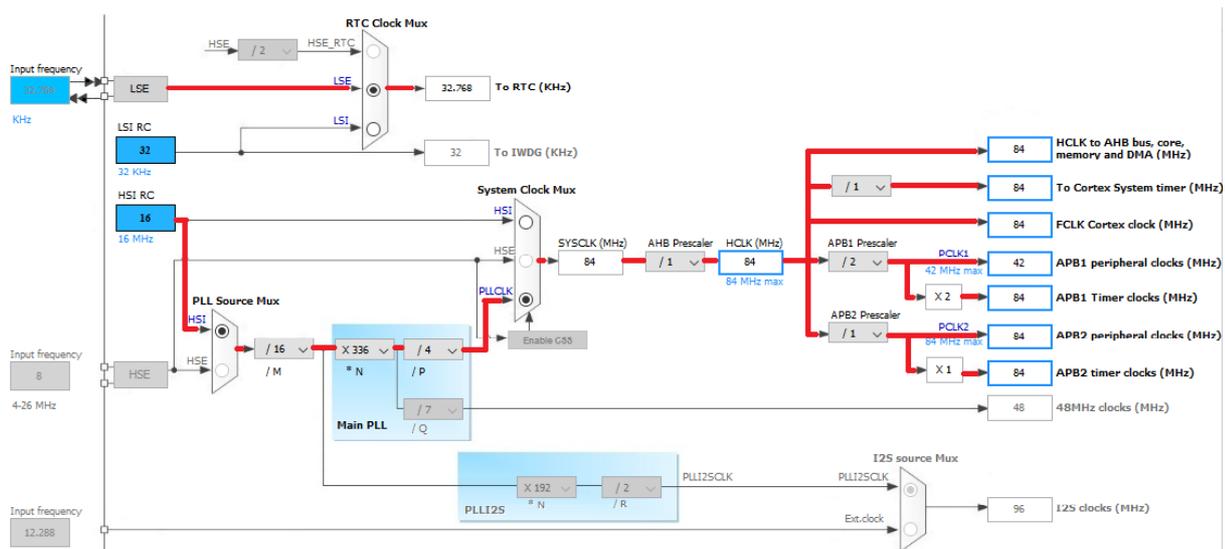


Figura 5.2 - Configuração do *oscilador* para o processador STM32F401RE

O *oscilador* selecionado foi de 84 MHz que é o *oscilador* máximo suportado pelo processador, a fonte de sinal do *oscilador* será o HSI, que é um oscilador interno. O uso do oscilador interno foi selecionado, em detrimento do oscilador externo (HSE), pois as únicas aplicações onde é desaconselhado o uso do HSI, são aplicações que tirem partido do uso do USB ou Ethernet. No caso deste protótipo, não é utilizado o USB ou Ethernet, como tal o uso deste oscilador é viável para esta aplicação.

Foi criado um projeto neste programa com as configurações básicas necessárias para o protótipo, as configurações mais relevantes são as seguintes:

- 1 x UART, utilizada para debug/configuração do dispositivo;
- 1 x UART com Interrupções, utilizada para comunicação com periféricos externos (Modulo *Bluetooth*);
- 1 x SPI com Interrupções, para acesso ao ADS1299;
- 1 x I2C com Interrupções, para acesso aos sensores;

- 3 x Timers com Interrupções (1 para a geração do PWM necessário para os LEDs SpO2, dois timers para uso geral);
- Configuração do *System oscilador*, para 84MHz;
- Configuração do *Watchdog*;

Depois de gerar o código de inicialização com as configurações anteriores, é necessário criar um projeto de aplicação de *firmware* para o processador em questão, para o qual é necessário um IDE (*Integrated Development Environment*), o ambiente integrado de desenvolvimento escolhido foi o *software Eclipse Mars* (The Eclipse Foundation, 2015), Figura 5.3, com os *plugins*, *System Workbench for STM32 - Bare Metal Edition* (ST Microelectronics, 2015b) e *GNU ARM Packs Manager* (GNU ARM Eclipse, 2014), sendo que o primeiro contém o STM32F4Cube e um conjunto de ferramentas como o *Open On-Chip Debugger* (OpenOCD), o *GNU Project Debugger* (GDB) entre outras ferramentas necessárias para programar, compilar e realizar *debug* neste tipo de processadores, e o segundo permite a visualização e a mudança de valores, em tempo real, dos registos do microprocessador e dos seus periféricos.

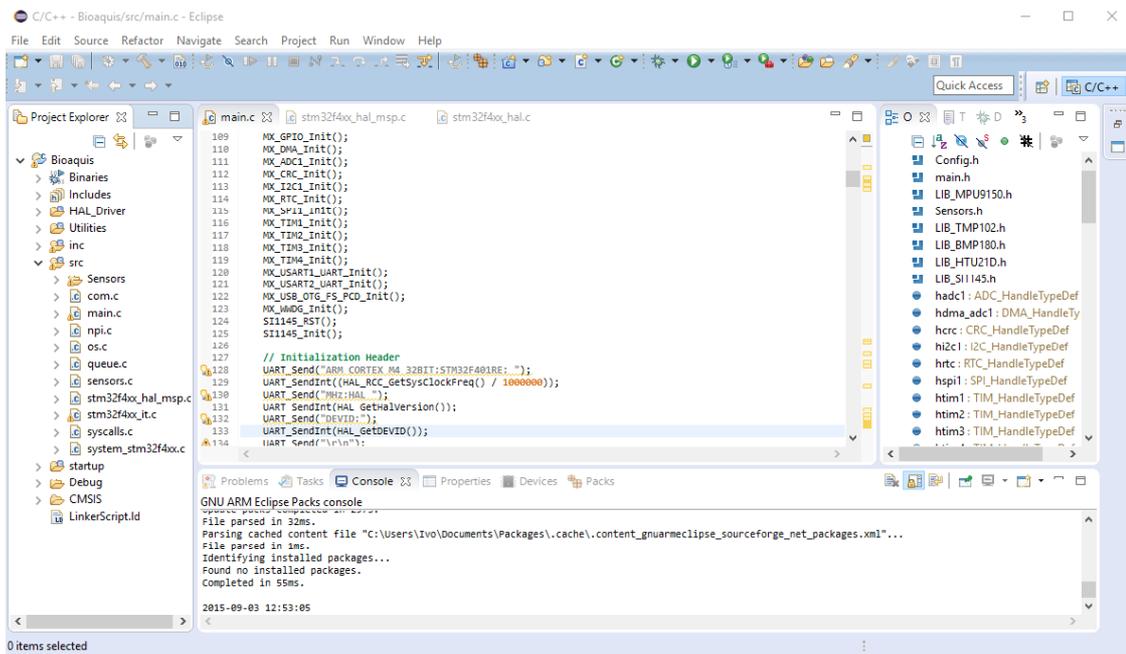


Figura 5.3 - Ambiente de Desenvolvimento Integrado Eclipse

Através do *plugin System Workbench for STM32 - Bare Metal Edition* criou-se um novo projeto de um modelo para o microprocessador escolhido. Como já foi referido este modelo inclui o CMSIS, o HAL Drivers e os BSP Drivers para a serie STM32F4, que são os componentes mínimos requeridos para poder desenvolver código para uma determinada placa de desenvolvimento.

É ainda necessário ativar a *Float Pointing Unit* FPU, presente no *hardware* deste microcontrolador. O uso da FPU tem vantagens quando se realizam cálculos entre quaisquer dois números (ST Microelectronics, 2012). Por exemplo é necessário:

- Alinhar os números
- Executar a operação
- Arredondar o resultado
- Codificar o resultado

Num processador sem FPU, todas estas operações são feitas pelo *software* através de uma biblioteca do compilador C e não são visíveis ao programador, no entanto a performance é muito baixa. Já num processador com FPU, todas as operações são inteiramente feitas pelo *hardware* num único ciclo, para a maior parte das instruções, aumentando assim a performance do processador quando este efetua cálculos matemáticos. A Figura 5.4 ilustra a configuração da FPU para o processador STM32F401RE.

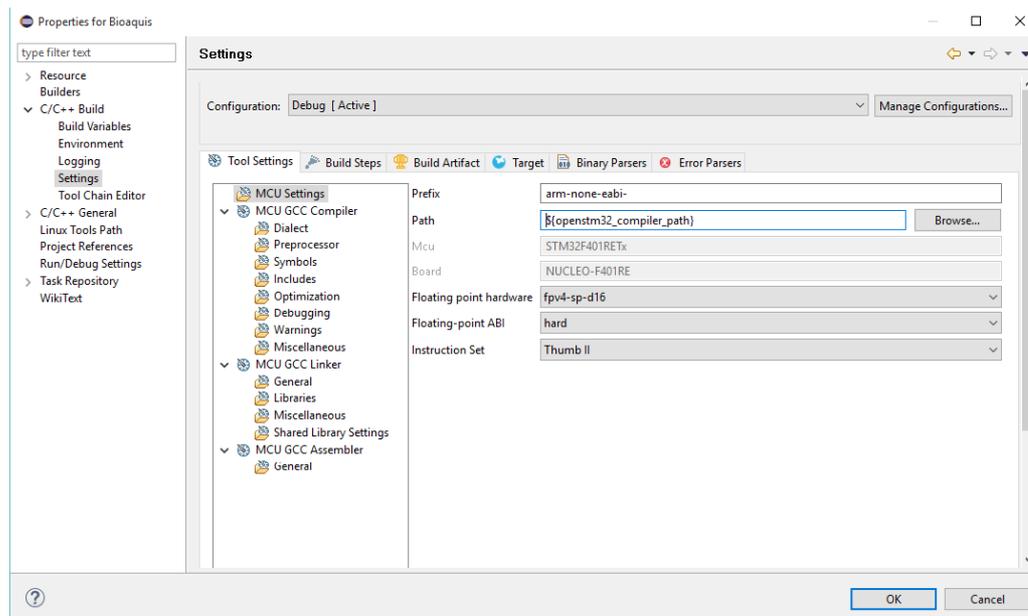


Figura 5.4 - Configuração da FPU

Após a criação do projeto a partir do modelo, necessitamos apenas de adicionar ao projeto o código de inicialização dos periféricos e do *oscilador*, gerado pelo STM32F4Cube, o projeto fica agora compilável. Para compilar e realizar o processo de *debug* do projeto o do *plugin System Workbench for STM32 - Bare Metal Edition* proporciona o openOCD ou o GDB, para essas operações.

5.2. Arquitetura

Os principais componentes da arquitetura encontram-se ilustrados na Figura 5.5.

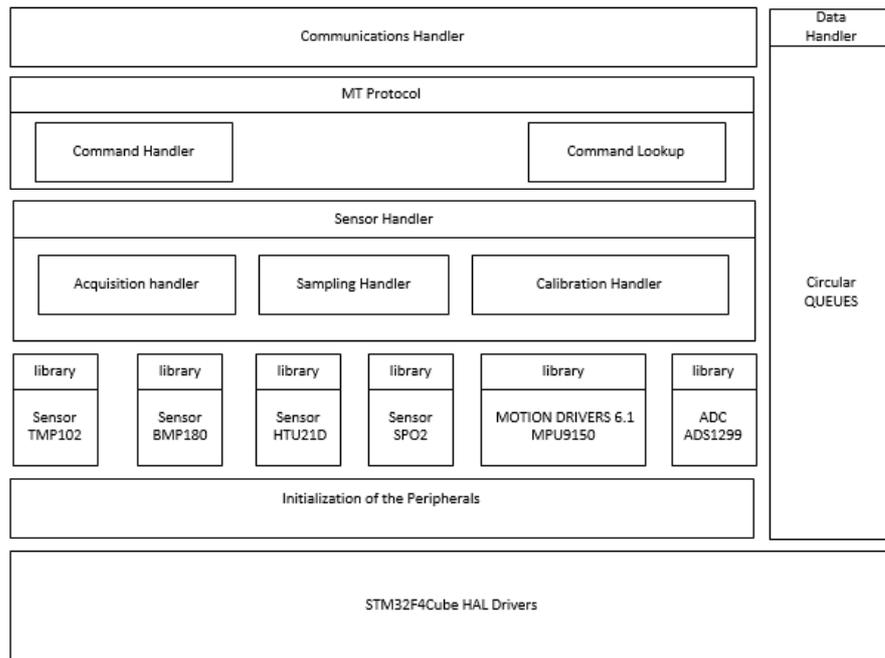


Figura 5.5 - Arquitetura de *firmware* para o STM32F401RE

Initialization of the Peripherals: Este módulo de inicialização dos periféricos contém o projeto criado pelo STM32F4CubeMX.

Library: os módulos que se encontram identificados como *Library*, são utilizados para realizar a interface com componentes externos neste caso, os sensores. Cada sensor tem a sua própria biblioteca e todos os métodos necessários para obter, dados da grandeza que este mede. Estes módulos de *firmware* foram criados com base nos *datasheets* de cada sensor.

Sensor Handler: Este módulo é utilizado para gerir os dados de cada um dos sensores, segundo a configuração de cada um deles. Cada sensor encontra-se identificado segundo a Figura 5.6. Este módulo gere ainda quando devem ser feitas as amostragens dos sensores e quando é que as aquisições devem ser enviadas, através do *communication Handler*.

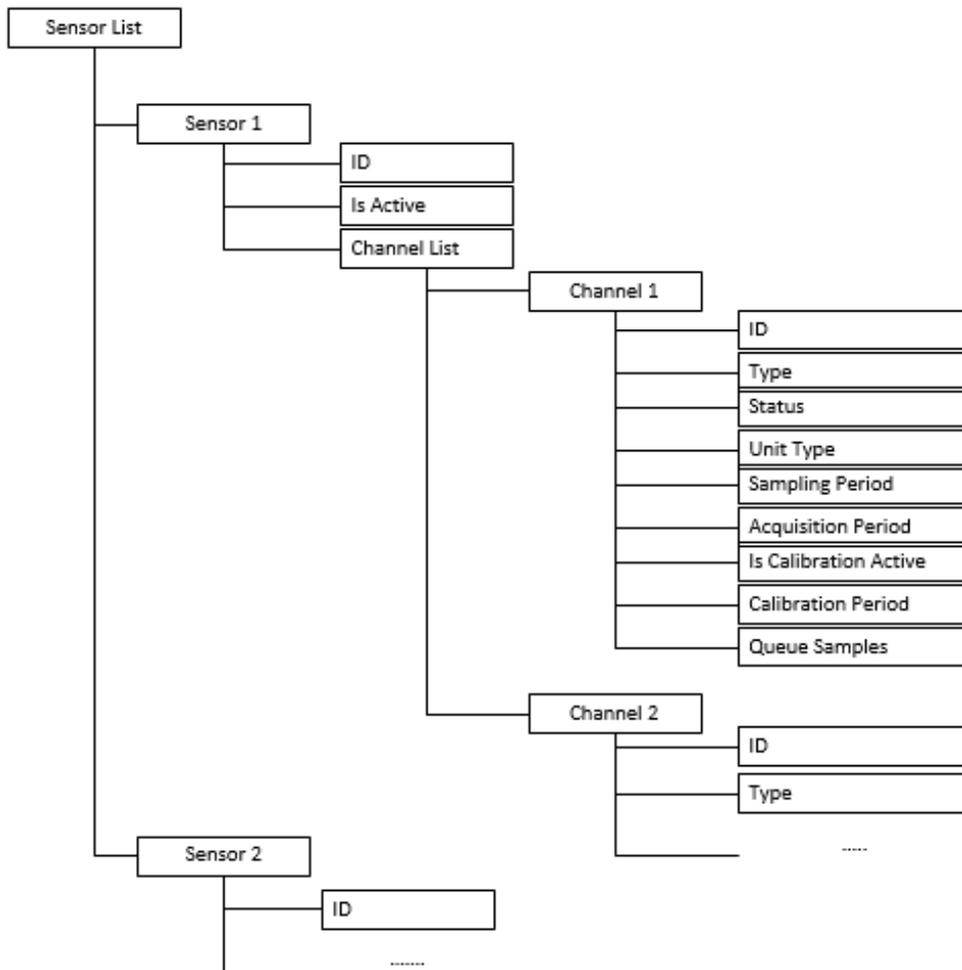


Figura 5.6 - Estrutura de identificação dos sensores e dos seus canais

De forma a poder obter os dados de uma forma estruturada dos sensores ligados ao protótipo, implementou-se a estrutura indicada na Figura 5.6. A funcionalidade dos parâmetros encontra-se descrita a seguir:

- *Sensor ID*: Este parâmetro identifica o sensor na lista de sensores, presentemente ligados ao dispositivo.
- *Sensor Is Active*: indica se o sensor se encontra ativo (1) ou inativo (0). Caso este parâmetro seja igual a inativo, o sensor está no estado inativo e não adquire dados de nenhum canal, independentemente do estado dos *Channel Ids*.
- *Channel ID*: Este parâmetro identifica o canal, dentro da estrutura do sensor. Um sensor pode ter múltiplos canais. Por exemplo, o sensor de humidade relativa (HTU21D) fornece dois tipos de grandezas, sendo elas a temperatura ambiente e a humidade relativa, nesta estrutura este sensor teria dois canais diferentes.

- *Channel Type*: Este parâmetro é uma enumeração dos tipos de canais pertencentes aos sensores que foram utilizados neste protótipo, sendo possível adicionar mais tipos de canais posteriormente até um máximo de 255. Os tipos de canais disponíveis encontram-se representados na Tabela 5.1.
- *Channel Unit*: Este parâmetro é uma enumeração dos tipos de unidades da grandeza medida pelo canal. Os tipos de unidades encontram-se associados com o tipo de canal e estão representados na Tabela 5.1.
- *Sampling Period*: Este parâmetro indica com que período deve ser realizada uma amostragem da grandeza fornecida pelo canal. Para os sensores ambientais tem o limite mínimo de 1 segundo, e para os sensores fisiológicos de 0,1 segundos.
- *Aquisition period*: este parâmetro indica o período de aquisição, isto é o período com o qual as grandezas obtidas dos sensores devem ser enviadas.
- *Is calibration Active*: Este parâmetro indica se o parâmetro *Calibration Active* está ativo ou não.
- *Calibration Period*: alguns sensores necessitam de executar uma calibração dos seus parâmetros com uma certa periodicidade, como por exemplo o sensor de pressão atmosférica (BMP180), este parâmetro indica o valor da periodicidade com que a calibração deve ser executada, o valor mínimo é de um Segundo.
- *Queue Samples*: Este parâmetro contém uma *queue* circular, com as amostras de dados e os seus *Timestamps*. Correntemente o tamanho destas *queues* tem de ser limitado, apenas permite guardar um máximo de 60 samples, pois encontra-se na SRAM deste processador. A descrição da estrutura da *queue* será descrita no módulo *Data Handler*.

Tabela 5.1 - Identificação do tipo de canal e das suas unidades

Tipo de canal	Channel Type	Unidade	Unit Type
Temperatura	0x00	Celcius (C)	0x01
Humidade relativa	0x01	Porcentagem (%)	0x02
Pressão atmosférica	0x02	Milibar (mbar)	0x03
Altitude	0x03	Metros (m)	0x04
Índice ultravioleta	0x04	Sem unidade	0x00
Luz Visível	0x05	Lux (lx)	0x05
Luz Infravermelha	0x06	Lux (lx)	0x05
Acelerómetro	0x07	Sem unidade	0x00
Giroscópio	0x08	Sem unidade	0x00
Campo magnético	0x09	Microtesla (uT)	0x06
Pedómetro	0x0A	Sem unidade	0x00
Frequência cardíaca	0x0B	BPM	0x07
ECG	0x0C	Volts	0x08
EMG	0x0D	Volts	0x08

Communications Handler: Este módulo gere as comunicações, orientando os pacotes de dados na direção correta. Cada ligação (UART, SPI) tem um byte de identificação, para identificar o pacote de dados. Neste momento apenas a ligação RS232 está terminada.

Monitor and Test Protocol: este módulo permite o envio e receção de mensagens segundo o protocolo de transporte *Monitor and test*. Um protocolo de transporte é necessário para que as mensagens possam ser trocadas entre dois dispositivos através de uma ligação RS-232/SPI. O propósito do protocolo de transporte é o de encapsular as mensagens em pacotes de dados para uma transmissão e receção correta e a integridade da mensagem. Os campos contêm múltiplos bytes e são transmitidos com o byte mais significativo em primeiro (MSB). Não existe aprovisionamento nem retransmissão de pacotes de dados perdidos. O pacote de dados serie são enviados para a aplicação de PC e para o dispositivo Bluetooth. Estes contêm um SOF (*Start Of Frame*), seguido de um pacote MT com comprimento variável, e terminado com um FCS (*Frame Check Sequence*).

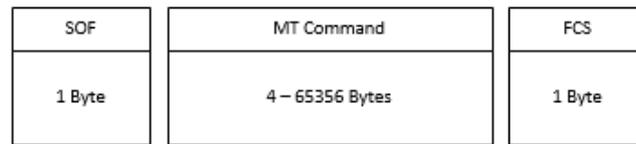


Figura 5.7 - Pacote serial Geral

SOF (*Start of Frame*): Este campo é um único byte com o valor igual a 0xFE, que define o início de cada pacote de dados

MT Command (*Monitor Test Command*): Este campo contém a estrutura da mensagem a enviar, o tamanho pode variar de 4 – 65356.

FCS (*Frame Check Sequence*): Este campo contém um único byte que é utilizado para assegurar a integridade do pacote de dados. Este campo é obtido calculando o XOR de todos os bytes nas mensagens a começar pelo SOF. O recetor volta a calcular o XOR da mensagem recebida e da mesma forma e compara o valor FCS obtido com o que vem na mensagem, se ambos os *Bytes* forem diferentes, o pacote recebido está corrompido. O método para calcular o FCS encontra-se representado no Anexo IV.

A estrutura *MT Command* é composta por 2 bytes que indicam o tamanho dos dados a enviar (*Payload*), 2 bytes que identificam o comando e pelos dados a enviar que pode tem um tamanho variável de 0 a 65531. A estrutura do *MT Command* está representada na Figura 5.8.

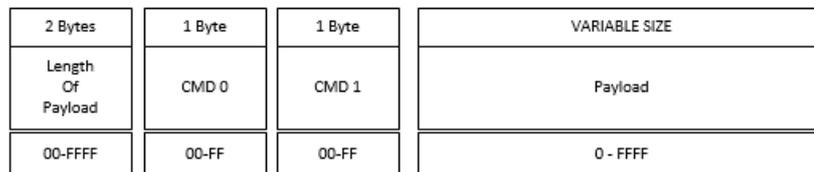


Figura 5.8 - Estrutura *MT Command*

Neste protótipo apenas existem 3 tipos de mensagens diferentes, podendo ser adicionadas mais posteriormente conforme seja necessário. Os tipos de mensagens e a sua identificação estão representadas na Tabela 5.2.

Tabela 5.2 - Identificação das mensagens MT

CMD 0	CMD 1	Descrição
0x01	0x02	<i>Configure Notification</i>
0x01	0x03	<i>Sensor Data Notification</i>
0x01	0x04	<i>Discover</i>

A mensagem *Configure Notification* permite configurar os tempos de amostragem, aquisição e calibração de um canal, pertencente a um sensor disponível. O período de amostragem indica o período com o qual será realizada uma amostragem da grandeza fornecida pelo respetivo canal.

Além destes três parâmetros este comando permite ainda configurar o estado do canal, isto é, se este se encontra ativo ou inativo, o mesmo para o estado da calibração. A resposta a este tipo de mensagem é indicada pela identificação do comando, no entanto, os dados a enviar são apenas 1 byte que representa se o pedido de configuração foi bem realizado ou não. Caso o pedido esteja correto é enviado o byte 0x06 (ACK), caso contrário é enviado o byte 0x15 (NAK). Se a configuração da notificação estiver correta, é enviado com a periodicidade do *Acquisition Period* uma mensagem do tipo *Sensor Data Notification Request* com os dados do canal configurado pela mensagem *Configure Notification*. Caso exista mais do que uma Notificação ativa são enviadas duas mensagens do tipo *Sensor Data Notification Request*. A estrutura desta mensagem e da sua resposta encontra-se representada pela Figura 5.9.

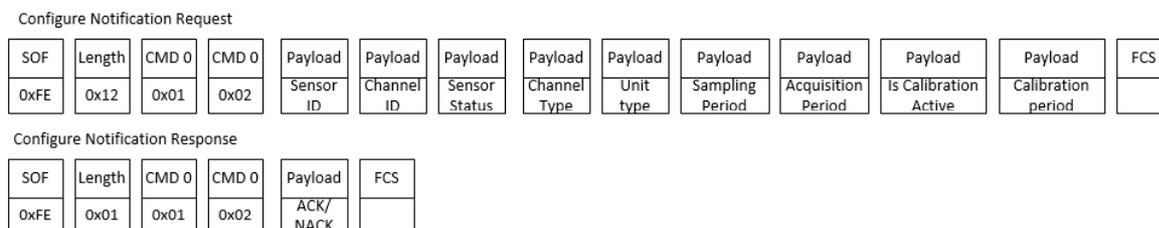


Figura 5.9 - Estrutura da mensagem *Configure Notification*

A mensagem *Sensor Data Notification* serve para pedir a ultima amostra a um canal. A estrutura da mensagem e da resposta a enviar está representada na Figura 5.10.

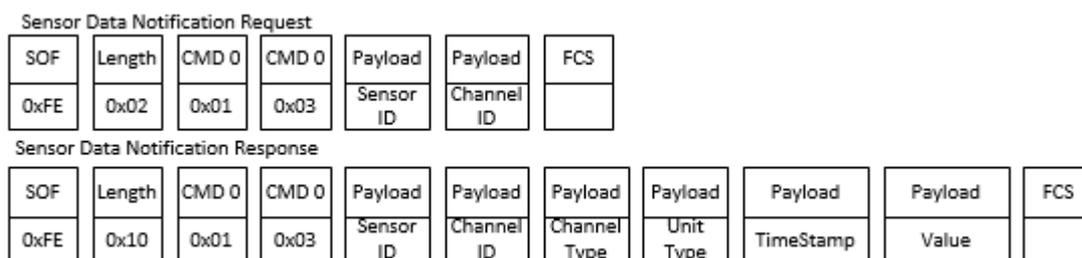


Figura 5.10 - Estrutura da mensagem *Sensor Data Notification*

A mensagem *Discover* é utilizada para identificar a estrutura de sensores presente no dispositivo, ao enviar esta mensagem com a estrutura representada na Figura 5.11, são enviadas tantas mensagens de resposta quantos os canais presentes na lista de canais. Cada uma das mensagens contem todos os parâmetros da estrutura para que seja possível pelo lado do recetor identificar o tipo de configuração existente para cada sensor e para cada canal desse sensor.

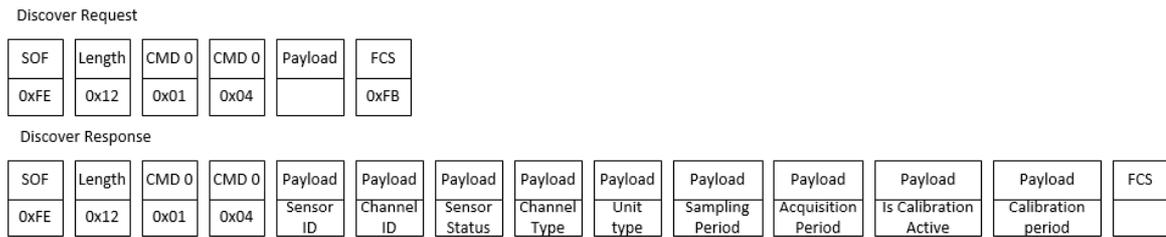


Figura 5.11 - Estrutura da mensagem *Discover*

A Figura 5.12, ilustra as seqüências de mensagens que podem ser trocadas entre a aplicação de *software* para PC e o Microcontrolador para obter os valores das aquisições.

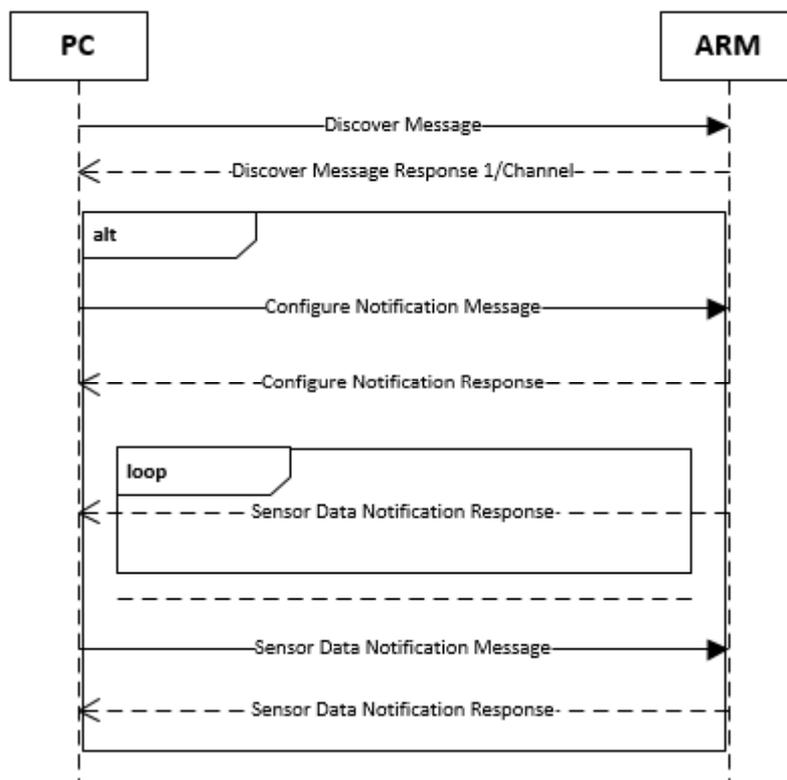
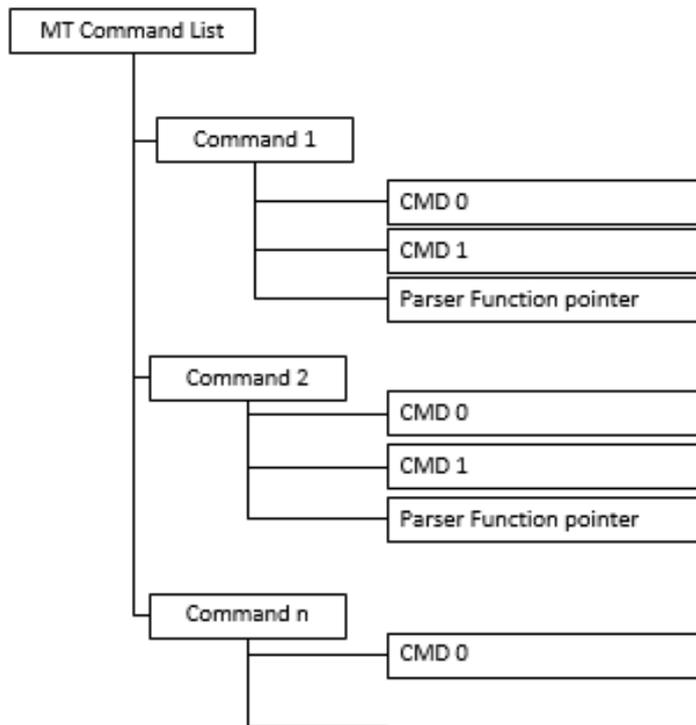


Figura 5.12 - Diagrama de seqüência de mensagens entre o PC e o protótipo

O módulo *MT Protocol*, contém ainda um modulo designado de *Command handler*, que tem a função de verificar se as mensagens recebidas são validas ou não, uma das verificações é a verificação da identificação dos MT Command. Os comandos suportados pelo dispositivo, Tabela 5.2, encontra-se organizados numa lista. Todos os comandos têm de ter uma função associada para realizar a descodificação da mensagem, essa associação é feita através de ponteiros para funções. Assim sendo todos os comandos seguem a estrutura representada pela Figura 5.13, contendo a identificação do *MT command* e um ponteiro para a sua função de descodificação.

Figura 5.13 - Estrutura da identificação dos *MT Commands*

O módulo Data Handler fornece as funções para criar e gerir as filas circulares. As filas são bastante utilizadas em aplicações de *firmware* e *software*, uma vez que quando estamos a executar uma determinada tarefa é muitas vezes necessário esperar por algum evento de modo a poder ter acesso a alguma coisa, por exemplo, apenas podemos processar um comando depois de termos todos os bytes pertencentes a esse comando. Uma fila é uma lista na qual todas as adições de elementos, a essa lista são realizadas em uma das terminações e todos os elementos removidos são feitos na terminação contrária. As filas também são chamadas de listas *First in, First Out* (FIFO). (Ryba, 2000) A estrutura geral das filas utilizada encontra-se representada na Figura 5.14.

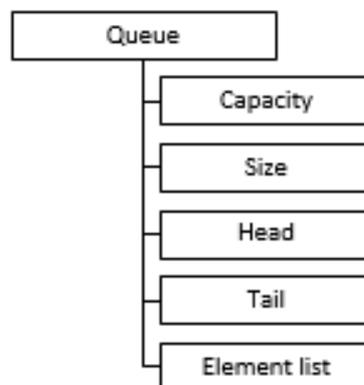


Figura 5.14 - Estrutura geral de uma fila FIFO

- *Capacity*: Este campo indica o número máximo de elementos que a filas pode ter na lista de elementos.
- *Size*: indica o número de elementos na lista de elementos.
- *Head*: Índice do primeiro elemento que entrou na lista de elementos;
- *Tail*: Índice do último elemento que entrou na lista de elementos.
- *Element List*: Este campo é a lista de elementos, que contem todas as entradas, nas filas.

Para um processamento eficiente das filas circulares, são necessários dois índices, para que se possa manter o controle de ambas as partes, a frontal e a parte posterior da fila sem mover nenhuma das entradas. Ambos os índices iniciam com o valor zero, ou seja, no primeiro elemento da lista. Para adicionar uma entrada à fila, basta incrementar a *Head* e colocamos a entrada na posição indicada pelo *Head*. Quando queremos remover uma entrada da fila, lê-se o valor indicado pela *Tail* incrementando-a posteriormente. Se o valor dos índices, *Head* ou *Tail* atingir o valor da capacidade máxima, estes voltam a ter o valor zero, fazendo assim com que a fila além de FIFO seja também circular. O uso de filas circulares permite ultrapassar a ineficiência do uso do espaço, bastando para isso pensar no vetor de dados como sendo um círculo, em vez de uma linha direita. Um exemplo da implementação deste método encontra-se representado na Figura 5.15.

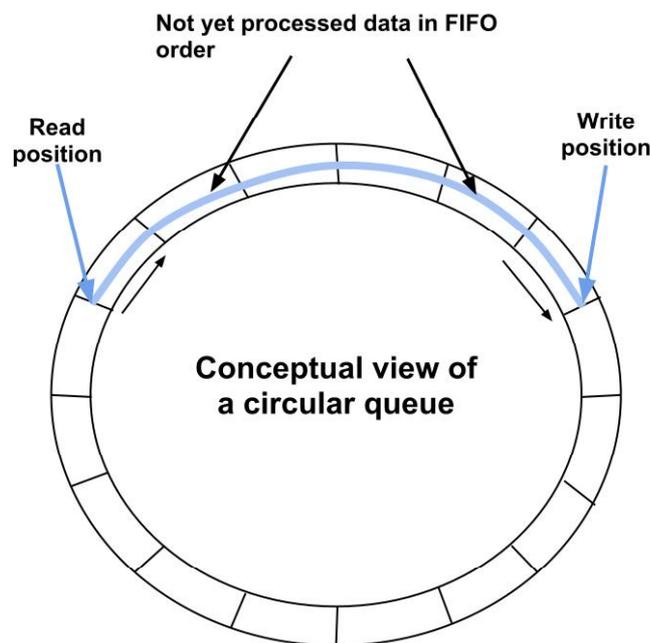


Figura 5.15 – Fila Circular

De modo a guardar os valores das amostragens realizadas e dos *Time stamps* para cada amostra, a estrutura utilizada para a fila é diferente, esta encontra-se representada na Figura 5.16.

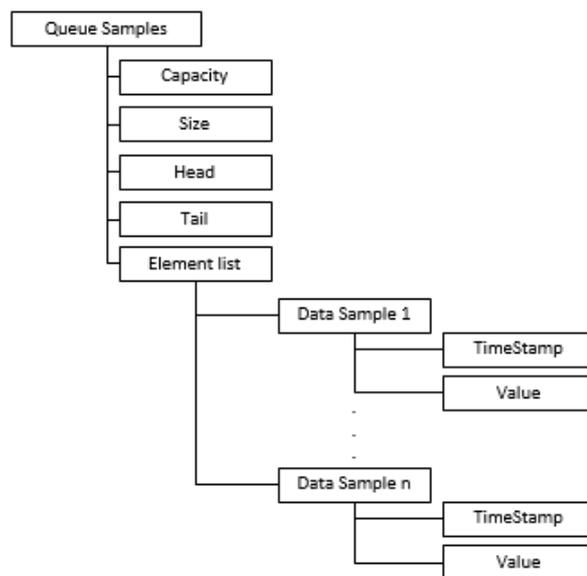


Figura 5.16 - Estrutura da fila para guardar amostras dos sensores

6. Desenvolvimento do Software de Apoio

Este capítulo descreve de forma resumida a aplicação de *software* desenvolvida no decorrer deste projeto. Esta aplicação é uma implementação em .NET, através do uso da linguagem de programação C#, no ambiente integrado de desenvolvimento *Visual Studio 2015 Community Edition* (Microsoft, 2015), de uma possível solução para realizar o interface com a aplicação de firmware desenvolvida no Capítulo 5.

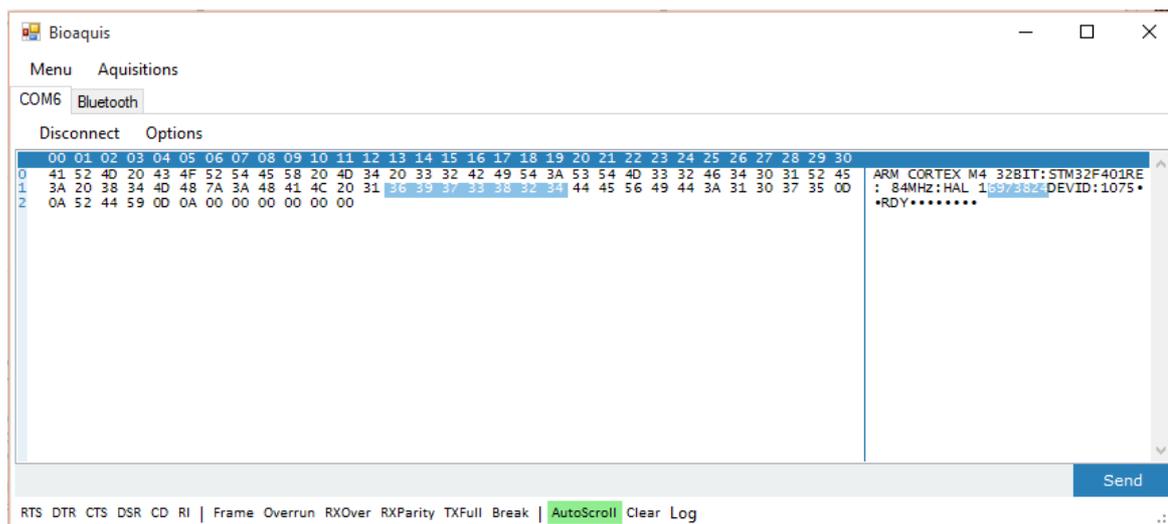


Figura 6.1 - Software desenvolvido

6.1. Arquitetura

Nesta secção, pretende-se descrever a arquitetura do *software* utilizada, esta descrição não será exaustiva uma vez que o software é extenso. O *software* utiliza como meio de transmissão a porta serie, com dois protocolos sobrepostos, sendo que um é o protocolo MT, utilizado para receber dados dos sensores listados na aplicação de *firmware*, e o outro o protocolo HCI que fornece dados para a criação, gestão e manutenção da rede BLE.

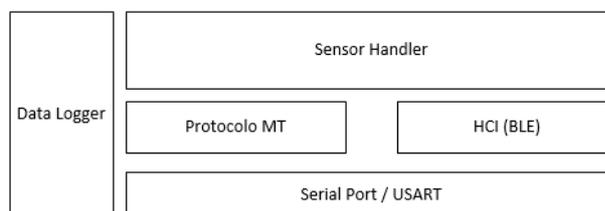


Figura 6.2 - Camadas de Software implementadas

A camada “*Serial Port*” é responsável por gerir a ligação da porta serie e por enviar e recolher dados. Esta fornece dados às camadas superiores do Protocolo MT e HCI (BLE), estas por sua vez analisam os dados para verificar se existem comandos dos respetivos protocolos. A camada “*Serial Port*” além de permitir a configuração da porta serie, permite também a

aquisição dos eventos de erro que possam ocorrer durante a transmissão, mostrando-os ao utilizador, Figura 6.3.

RTS DTR CTS DSR CD RI | Frame Overrun RXOver RXParity TXFull Break | AutoScroll Clear Log

Figura 6.3 - Barra de estado

O modo de agregação dos dados é baseado no esquema produtor consumidor, no entanto, contem múltiplos consumidores, a Figura 6.4, representa um esquema simplificado do modelo implementado no programa.

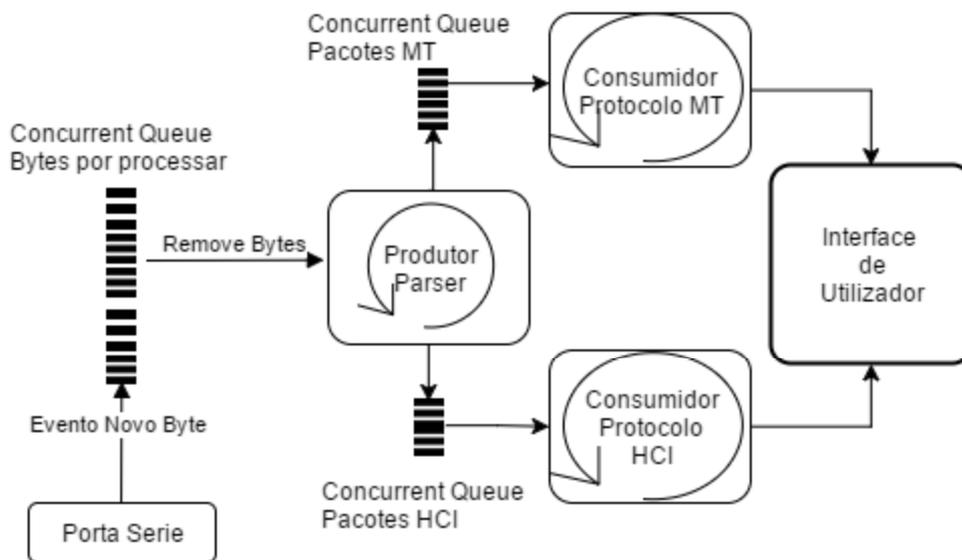


Figura 6.4 – Esquema simplificado produtor múltiplos consumidores

O produtor, que recebe os dados, é executado numa *Thread* separada da aplicação principal, de modo a conseguir agregar todos os dados produzidos pelos eventos de dados recebidos proveniente da porta serie. Na ocorrência destes eventos, os dados são colocados numa *Concurrent Queue* (Microsoft, n.d.) , que são filas de espera *first in first out* (FIFO) especialmente construídas para evitar a ocorrência de *deadlocks*, no acesso aos dados (objetos) por parte de múltiplas *Threads*. Como o produtor corre numa *Thread* separada, isto permite que o reenaminhamento dos pacotes de dados para as camadas superiores seja efetuado mais rapidamente. Para a transmissão dos pacotes de dados é também utilizado um esquema produtor consumidor, em que o consumidor corre numa *Thread* separada para realizar o envio de dados pela porta serie. Esses dados encontram-se mais uma vez guardados numa *Concurrent Queue*. Qualquer camada pode inserir dados nesta fila de espera, portanto, existem múltiplos produtores e apenas um consumidor. Quando existirem dados na fila de espera de saída o consumidor que corre também numa *Thread* separada, retira esses dados da fila de espera, tenta realizar o envio através da porta serie, caso ocorra um erro é mostrado um aviso para voltar a tentar enviar de forma a não perder os dados. Por fim a camada “*sensor handler*” permite analisar e configurar as aquisições e amostragens dos sensores e a sua estrutura é idêntica à camada representada no

capítulo 5. A camada Protocolo MT permite identificar e enviar comandos os comandos MT, funcionando de modo análogo à camada implementada no capítulo *firmware*. A camada denominada como HCI/BLE foi criada com o intuito de disponibilizar os comandos para criar e gerir os dados da rede Bluetooth. A *Texas Instruments* disponibiliza um *software* para isso o *BTool*, no entanto o código fonte não se encontra disponível. Existia, no entanto, uma classe com todos os comandos Bluetooth, devido a forma como foi estruturada o uso desta torna-se difícil. No decorrer deste projeto essa classe foi reestruturada, foram criados *namespaces* para cada um dos blocos presentes (GATT, L2CAP) em que cada um contem os comandos e eventos referentes a eles próprios. Foi criada também uma nova classe “*BLEHost*” que permite a identificação e a criação dos comandos, tanto para envio como para receção.

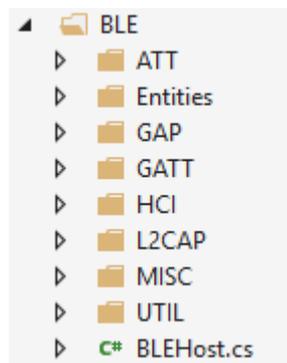


Figura 6.5 – Estrutura final da camada HCI/BLE para envio/receção de comandos Bluetooth

Esta biblioteca encontra-se dividida em blocos, os mesmos que existem na *stack* de *firmware*, cada bloco contem os comandos e eventos referentes a esse bloco.

6.2. Funcionalidades

O programa além de permitir visualizar os dados recebidos pela porta serie permite também a criação de ficheiros de registo desses mesmos dados, permitindo assim guardar as aquisições para análise posterior.

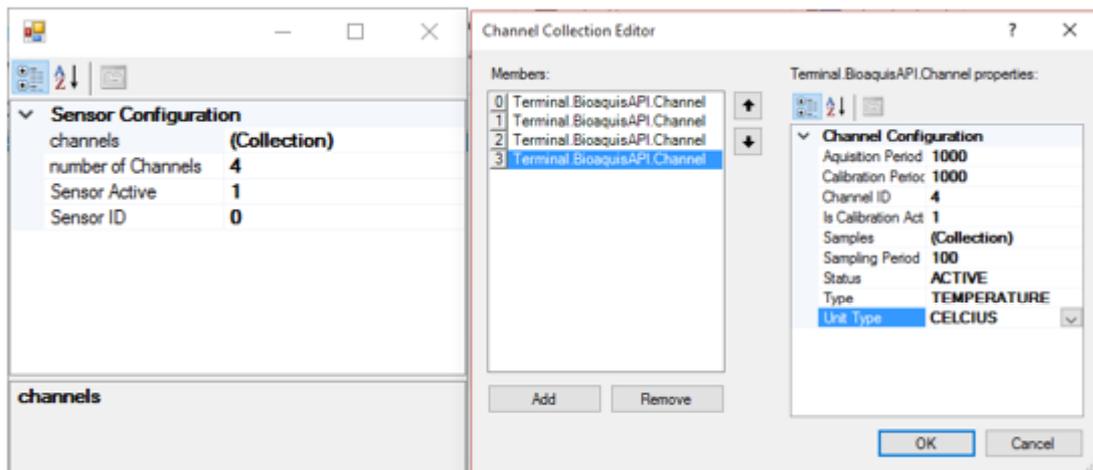


Figura 6.6 - UI de configuração das aquisições dos sensores

Além destas funcionalidades, o programa permite ainda a configuração dos sensores identificados no *firmware* desenvolvido.

7. Resultados

O objetivo principal deste projeto seria a realização de aquisições de dados provenientes dos sensores, quer sejam fisiológicos ou ambientais. Durante a realização deste projeto não foi possível obter as aquisições fisiológicas, pois seria necessário utilizar o circuito desenvolvido para o *analog front End* (AFE) ADS1299, para as realizar. O PCB encontra-se preparado para produção, mas durante a realização desta tese não foi possível produzi-lo em tempo útil. No entanto, foi possível obter dados dos sensores que medem grandezas ambientais.

7.1. Aquisições

Os dados do gráfico da Figura 7.1 á Figura 7.4, foram obtidos utilizando a plataforma desenvolvida (*Hardware, firmware e software*) para realizar as aquisições. O gráfico da Figura 7.1 representa a aquisição de temperatura, durante um período de 24 horas, com períodos de amostragem de 1s e períodos de aquisição de 1 minuto, o valor final da aquisição é a média das aquisições obtidas durante esse minuto.

Como se pode observar, existem três sensores de temperatura, embora dois deles sejam utilizados como parâmetro auxiliar para realização dos cálculos da pressão atmosférica, e da humidade relativa, pode-se comparar a sensibilidade dos vários sensores realizando amostragens simultâneas destes sensores. Neste caso a variação da temperatura nas leituras simultâneas dos vários sensores não é significativa, e as diferenças entre os valores obtidos devem-se à diferença entre as resoluções dos sensores.

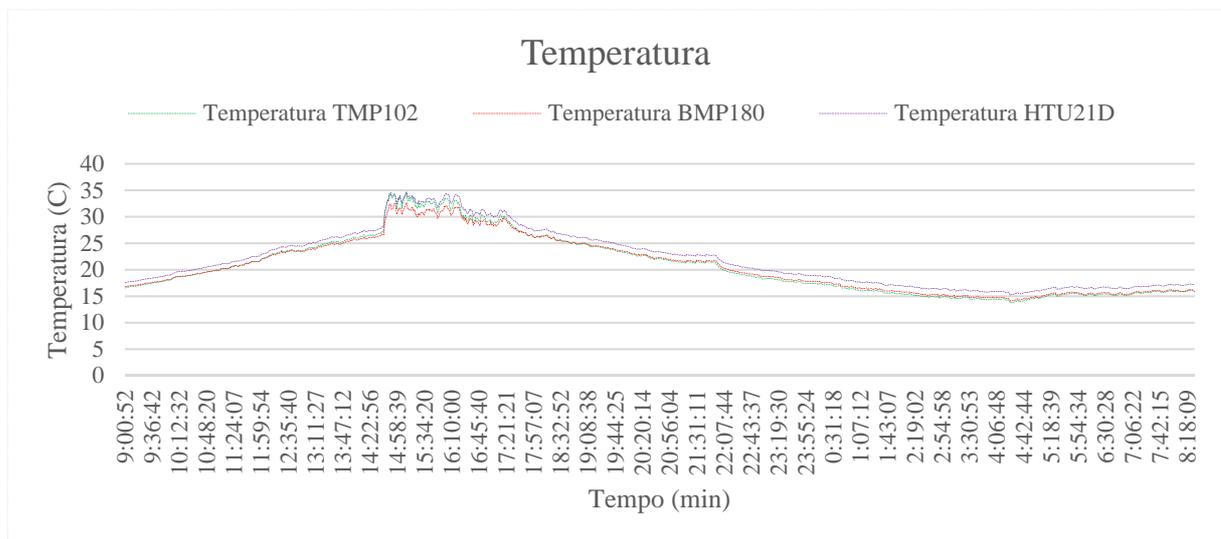


Figura 7.1 - Aquisição da temperatura ambiente (Período de 24horas)

Assim com o gráfico da temperatura ambiente, foi obtido um gráfico da humidade relativa durante o mesmo período, observando o gráfico da temperatura, Figura 7.1 e o gráfico da humidade relativa, Figura 7.2 verifica-se que com o aumento da temperatura existe uma

diminuição da humidade relativa, sendo que no período noturno a humidade relativa tem um valor bastante superior ao período diurno.

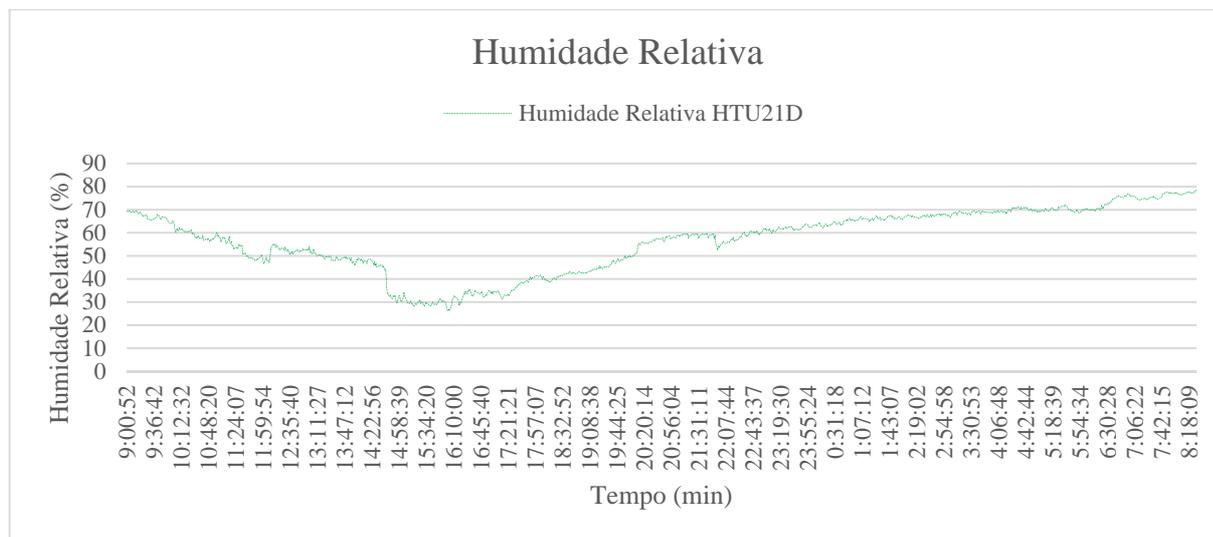


Figura 7.2 - Aquisição da Humidade Relativa (Período de 24horas)

Outro dos valores ambientais obtidos foi a pressão atmosférica, como se pode observar pelo gráfico da Figura 7.3, não existe grande variação deste parâmetro sem que exista uma variação da altitude ou da temperatura.

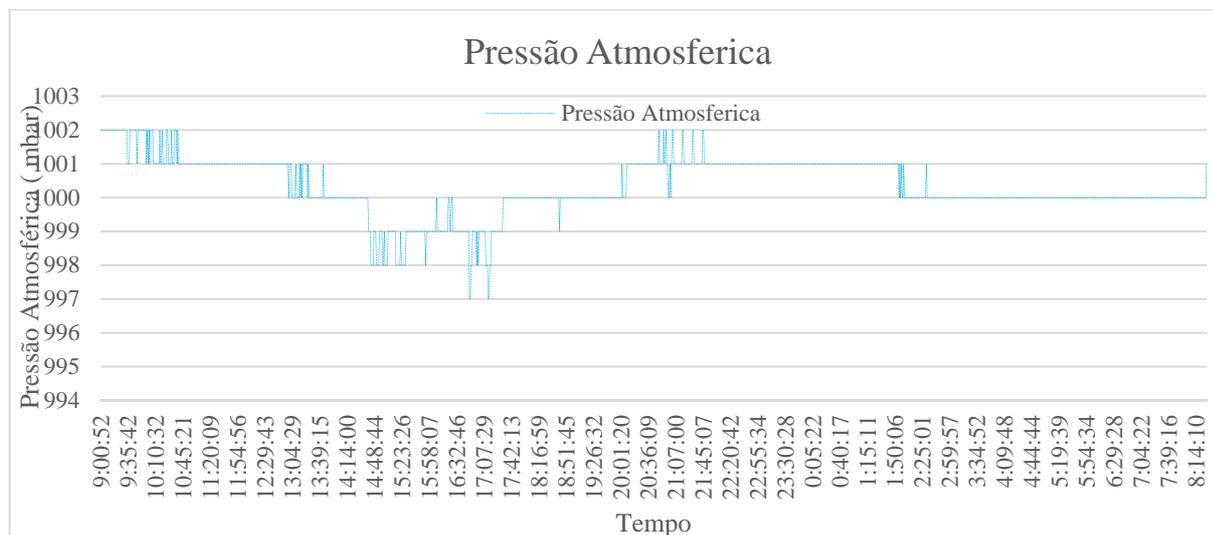


Figura 7.3 - Aquisições da Pressão Atmosférica (Período de 24 horas)

A Figura 7.4 apresenta um segmento do sinal obtido com o oxímetro de pulso, estão apenas representados 2 minutos do sinal, para que seja mais fácil a sua visualização. A figura apresenta ainda uma janela de 20 segundos desse sinal, para que se consiga verificar melhor as características do sinal.

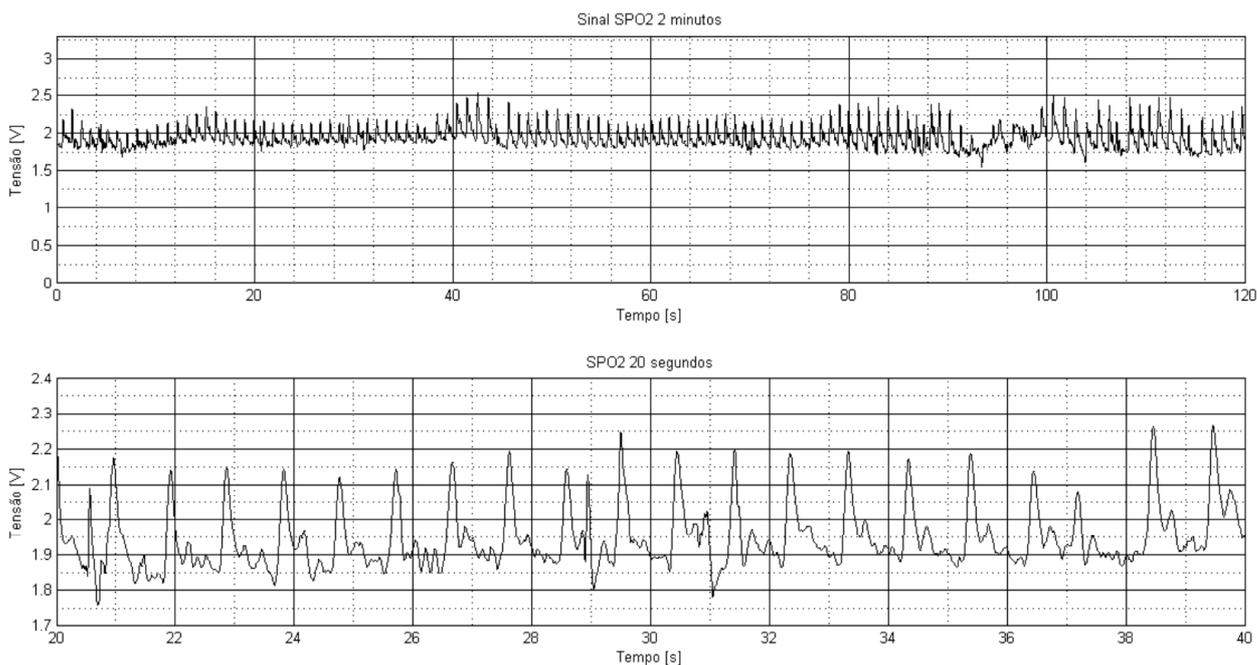


Figura 7.4 - Sinal obtido do Oxímetro de pulso

Depois de obtido o sinal foi submetido a um filtro digital, passa baixo, *Butterworth* de ordem 10, com uma frequência normalizada de 0,1 Hz de forma a remover o ruído. Para criar os gráficos a partir das amostras obtidas, e devido ao seu grande número, foi utilizado o *software Matlab*. Observando a figura, verifica-se que o sinal corresponde ao esperado, apresentando um sinal com uma componente de frequência fundamental AC de aproximadamente 1Hz, que depende da frequência cardíaca. Depois da sístole (período de contração muscular das câmaras cardíacas), existe um aumento do volume de sangue nas artérias, o que reduz a intensidade de luz recebida. Durante a diástole (período de relaxamento muscular), o volume de sangue nas artérias diminui e, conseqüentemente, existe um aumento da transmissão de luz. Assim, o sinal aparece com uma componente AC. (Allen, 2007) Esta componente AC sobrepõe-se ainda a uma componente DC que se relaciona com os tipos de tecidos e o volume médio do sangue. Esta componente DC também varia lentamente devido à respiração, e a atividade motora. O tempo e o período de cada impulso são ditados pela pulsação e a amplitude pela concentração dos vários constituintes do sangue arterial e o caminho através do qual a luz passa.

8. Certificação e Qualificação de dispositivos de Rádio Frequência

De modo a poder vender um produto Bluetooth LE no mercado da Europa ou dos Estados Unidos da América, um equipamento necessita de obedecer a alguns requerimentos regulatórios da FCC (*Federal Communications Committee*), da Comissão Europeia (R&TTE) e *Industry Canada* (IC). Para um dispositivo Bluetooth existe ainda a necessidade da certificação de Bluetooth *Compliance*.(Instruments, 2015)

8.1. Processo de Qualificação de um dispositivo Bluetooth

No 1 de Fevereiro de 2014, a Bluetooth SIG lançou um processo de qualificação simplificada e estrutura de taxas revista para todas as novas ofertas de produtos Bluetooth.(Bluetooth SIG, 2014a). Antes do lançamento de um produto Bluetooth no mercado, a solução necessita de ser qualificada. Existem projetos de referência, por exemplo, CC2540EM/CC2541EM disponibilizados pela Texas Instruments, que se forem estritamente seguidos, sem que exista nenhuma mudança nos perfis do Bluetooth, não é necessário passar pelo processo de qualificação. Os perfis Bluetooth proprietários não fazem parte do processo de qualificação, no entanto é necessário que estes sigam o protocolo disponibilizado pelo GATT. As soluções já referenciadas e listadas são designados por *Qualified Design Listings* (QDL) (Bluetooth SIG, 2014b) com a correspondente *Qualified Design Identification* (QDID). Quando não se tem a certeza se o projeto de referência foi estritamente seguido, é necessário contactar uma entidade para ajudar a aconselhar qual o caminho a seguir, estas entidades denominam-se por *Bluetooth Qualification Experts* (BQE). Um BQE é individualmente reconhecido pelo Bluetooth SIG para fornecer a um membro do Bluetooth SIG, serviços relacionados com a qualificação, os BQE ajudam ainda na compreensão do processo e da política de qualificação. O recurso a BQE para realização do processo de qualificação é opcional. O Bluetooth SIG disponibiliza um guia de acesso rápido para realizar o processo de qualificação. Os passos a seguir para realizar o processo de qualificação são os seguintes:

1. É necessário ser um membro do *Bluetooth SIG* (Sig, 2010).
2. Em seguida é necessário comprar uma *Declaration ID*, para realizar a listagem do dispositivo, a taxa de listagem depende do tipo de assinatura.
 - *Adopter*: \$8,000
 - *Associate*: \$4,000
 - *Innovation Incentive Program*: \$2,500
3. Referenciar um QDID e listar o produto.
4. Por fim, é assinada a declaração de conformidade (DoC)

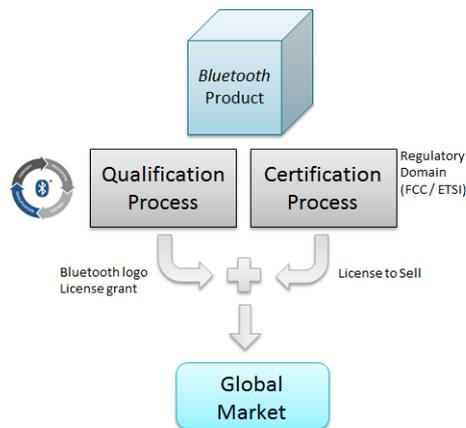


Figura 8.1 - Processo de colocação de um novo produto Bluetooth no mercado

8.2. Processo de Certificação de FCC

Todos os transmissores devem estar registados e certificados pela FCC (FCC, n.d.-c) para assegurar que estes, não causam emissões perigosas e interferências nas frequências reguladas. Existem dois modos de obter uma certificação FCC, ou recorremos diretamente à entidade FCC ou através de um *Telecommunications Certification Body* (TCB). Para quase todos os dispositivos é possível escolher o uso da FCC ou o uso de um TCB para a certificação, a exceção é quando o equipamento utilizada uma nova tecnologia ou quando os métodos de teste não estão bem definidos ou não são claros, então a única entidade que pode fornecer a certificação é a FCC.

O FCC-ID (FCC, n.d.-a) é um identificador único, com 4 a 17 caracteres, para equipamentos que consistem de dois elementos, para o equipamento consiste de dois elementos, o código de garantia e o código de produto do dispositivo. O código de garantia é um conjunto de três caracteres alfanuméricos, este código começa sempre com um caracter alfanumérico e não contem os números zero e um. O Código de garantia é assignado de forma permanente pela FCC a uma empresa, para autorização de todos os equipamentos de rádio frequência. O código de produto é a porção do código FCC ID que não faz parte do código de garantia e começa após os três primeiros caracteres. O código de produto pode também incluir hífen (-). O código de produto consiste no mínimo de 1 caracter e no máximo de 14, por exemplo, FCC ID: XXXXXABC123 O FCC-ID deve ser permanentemente marcado, ou diretamente no transmissor ou através do uso de uma etiqueta permanentemente afixada neste. O FCC-ID deve poder ser lido diretamente na altura da compra do produto.(FCC, n.d.-b) A Figura 8.2 ilustra a possível configuração da marcação/etiqueta utilizada por um dispositivo certificado.



Figura 8.2 - Etiqueta/marcação FCC-ID

Quando se submete um produto a um processo de certificação da FCC, é necessário entregar alguma documentação, de modo a que o equipamento possa ser testado e certificado. A documentação a entregar no início do processo é a seguinte:

- Carta de apresentação
- Relatório de testes do laboratório
- Manual de utilizador do produto
- Esquemático do produto com uma listagem dos componentes utilizados (BOM)
- Diagrama de Blocos do produto
- Fotografia com a configuração de teste
- Fotos internas/externas
- Indicação do posicionamento do FCC-ID
- Descrição operacional do produto

É possível modificar um produto depois ser certificado, sem ter que passar pelo processo de certificação de um novo equipamento, existem várias alterações permissivas (PC) / modificações (FCC, n.d.-d) que se podem realizar no entanto, sempre que existam modificações na determinação da frequência básica na estabilização do circuito, incluindo o oscilador e a taxa de transmissão de dados, estágios de multiplicação de frequência, a adição de circuitos moduladores básicos ou a potencia máxima de transmissão, será sempre necessário obter um novo FCC-ID e uma autorização para um novo equipamento. Existem 3 classes de alterações permissivas, sendo elas as seguintes:

- Classe 1: Modificações que não degradam as características reportadas á FCC e não requerem a modificação da concessão.
- Classe 2: Modificações que degradam a performance reportada á FCC, mas ainda estão em conformidade com os limites, é necessário a modificação da concessão.
- Classe 3: Modificações ao *software* do transmissor, é necessária a alteração da concessão.

As modificações da classe 1 podem ser alterações, por exemplo, nas pistas do PCB. Mesmo não existindo a necessidade de requerer uma concessão nova para o produto, é necessário documentar as alterações realizadas. As modificações da classe 2 podem ser, por exemplo a substituição de um circuito integrado, ou a substituição de porções do circuito do transmissor, que realizem algumas subfunções como por exemplo um circuito integrado amplificador ou por

exemplo um oscilador. A classe 2 requer a descrição das modificações efetuadas, bem como a realização de um novo teste para mostrar que o dispositivo continua em conformidade com as regras fornecidas pela FCC. A classe 2 requer uma nova concessão, mas, não existiram modificações no FCC-ID.

8.3. Processo de Certificação CE

A marca CE é uma marca de conformidade obrigatória para os produtos colocados no mercado europeu. De modo a colocar um dispositivo rádio no mercado europeu, é necessário demonstrar que este se encontra em conformidade com a norma R&TTE.(CE, n.d.) A marca CE indica que o fabricante declarou que o produto ou equipamento está conforme com todas as normas Europeias aplicáveis, e possibilita o movimento livre do produto dentro do mercado europeu. A norma R&TTE depende da harmonização de padrões desenvolvidos pela *European Standarts Organizations* (ESTI, CEN). Estes padrões foram harmonizados e definem as características técnicas, as quais podem ser utilizadas para alcançar os requerimentos das normas, como:

- Proteção da saúde e segurança
- Compatibilidade Eletromagnética
- Uso efetivo do espectro de rádio

Para dispositivos que operam na gama ISM as normas harmonizadas (ESTI, n.d.) são:

- EN 50371 – Conformidade com a exposição humana aos campos eletromagnéticos
- EN 300328 – Teste de equipamentos rádio para transmissão de dados na gama 2.4GHz ISM.
- EN 301489 – Compatibilidade eletromagnética

Existem vários caminhos para demonstrar a conformidade dos produtos com a norma R&TTE, para a maioria dos casos que cumpram com os requisitos das normas harmonizadas relevantes permite presumir a conformidade exigida, assim, permitindo-lhe para assinar uma Declaração de Conformidade com confiança. Um laboratório de testes irá fornecer o caminho e ajudar na colocação do produto no mercado.

8.4. Processo de Certificação IC

A IC (*Industry of Canada*) (Canada, n.d.) é a autoridade canadiana que autoriza a certificação de produtos e fornece as normas. Como para a certificação FCC a certificação IC pode ser realizada através da IC ou através de um TCB. Para uma certificação IC é necessário ter um representante no Canada, e a IC requer uma carta de confirmação dos representantes com a submissão da aplicação de certificação. Da mesma forma que um equipamento com radio, no mercado americano necessita de ter a marcação FCC-ID, um equipamento no mercado canadiano necessita de ter a marcação IC-ID. O IC-ID consiste em dois pares, o número da empresa (CN) e um número único de produto (UPN), o qual pode ser escolhido pelo fabricante. Para obter um CN é necessário utilizar o sistema de listagens (Industry of Canada, n.d.-a) Um

exemplo de um IC-ID seria o seguinte IC ID: XXXXZ-YYYYYYYYY. Quando se submete um produto a um processo de certificação da IC, é necessário entregar alguma documentação, de modo a que o equipamento possa ser testado e certificado. A documentação a entregar no início do processo é a seguinte:

- Uma cópia preenchida do formulário *Appendix A — Application and Agreement for Certification Services* (Industry of Canada, n.d.-b), uma carta de apresentação a explicar o tipo de serviços de certificação requeridos e uma breve descrição do equipamento rádio.
- Uma cópia preenchida e assinada *Appendix B — Test Report Cover Sheet*(Industry of Canada, n.d.-b).
- Um relatório detalhado do teste dos requerimentos técnicos das especificações radio aplicáveis (RSS).
- Fotografias e literatura do modelo do produto.
- Um esquemático e diagramas de blocos do produto.
- Ilustração ou amostra da etiqueta do produto que irá conter o IC-ID.

Quando um equipamento já se encontra certificado pela FCC, segue as seguintes condições:

- O relatório de teste tem menos de um ano.
- O laboratório que realizou o teste deve ter uma câmara anecoica aprovada pela IC.
- Deve ser entregue uma tabela cruzada que mostra que o equipamento está conforme com todos os requerimentos aplicáveis pela IC.

Não é necessário realizar nenhum teste de RF extra, quando se realiza a aplicação para uma certificação IC.

9. Conclusão e Trabalho Futuro

A pesquisa bibliográfica do capítulo 3 permitiu compreender vários aspetos importantes sobre alguns elementos da arquitetura utilizada, bem como, tecnologias alternativas, que podem ser utilizadas para a realização deste projeto. O capítulo 8 foi bastante elucidativo no percurso de certificação e qualificação de um produto com um transmissor *Bluetooth*.

A seleção do *hardware* a utilizar, bem como o desenvolvimento da placa de aquisições de sinais bio potenciais (AFE), foi bastante produtiva, pois o desenvolvimento foi levado do início ao fim, ou seja, desde a escolha dos componentes a utilizar, à criação da BOM (*Bill of Materials*), à criação das bibliotecas dos componentes para o programa de desenho de CAD, ao projeto do esquemático, e por fim, a placa de circuitos (PCB), bem como os *gerbers* para produção, e uma simulação 3D do PCB para visualização do resultado final.

Como trabalho futuro na parte de *hardware*, será necessário a produção da placa de circuito impresso que contem o AFE, pois só assim se poderá adquirir os sinais fisiológicos. Outra implementação futura, será uma modificação da plataforma de *hardware* de modo a separar cada sensor num módulo, que contenha apenas um *transceiver* Bluetooth e o próprio sensor, isto trás vantagens pois assim cada módulo terá um consumo energético menor, pois nem para todos os sensores é necessário o mesmo poder de processamento fornecido pelo processador ARM.

Outra implementação futura é a substituição do próprio módulo Bluetooth, após o início deste trabalho apareceram novos dispositivos de *hardware* Bluetooth no mercado baseados na arquitetura de processador ARM Cortex M0 32-bits, o que trás uma grande vantagem em relação aos que foram utilizados neste projeto, pois estes são baseados no processador 8051 que é um processador de 8-bits.

O *firmware* implementado no decorrer deste trabalho é funcional, e permite a configuração e a aquisição de dados provenientes dos sensores. No entanto ainda se encontra incompleto, ficando a faltar a implementação da comunicação *Bluetooth*, o teste da biblioteca desenvolvida para realizar a interface como o AFE ADS1299, e a finalização da integração da biblioteca *MotionDriver 6.12* de modo a obter valores do IMU MPU9150, neste momento a biblioteca apenas se encontra integrada no projeto de *firmware*, de modo a que o *firmware* seja compilável.

Uma das implementações futuras será o uso da solução de *hardware* da *ST Microelectronics* para *Bluetooth*, o *Bluetooth NRG* (Microelectronics, 2015), as principais vantagens do uso deste *hardware* em oposição ao da Texas CC2541, é a sua *stack* de *firmware* que além de ser diretamente utilizável no *firmware* desenvolvido para o processador ARM (STM32), também não necessita de um software com licença, como é o caso do *IAR embeded Workbench 8051*, utilizado na *stack* da *Texas Instruments*.

No *software* este não fazia parte dos objetivos iniciais, contudo no decorrer deste projeto foi necessário o seu desenvolvimento de modo a poder realizar configurações e *logs* dos dados

das aquisições. Mais tarde, iniciou-se também a implementação de uma camada para receber comando HCI *Bluetooth* de modo a que este *software* possa também realizar a criação da rede Bluetooth e a sua gestão.

No *software* o trabalho futuro a realizar, será a adição de um modo para utilização dos módulos Bluetooth internos aos dispositivos. Atualmente é utilizada uma pen USB como dispositivo central que permite a criação da rede e a sua gestão. O uso dos dispositivos de Bluetooth internos, seria uma mais-valia, pois como o *Bluetooth* é considerado uma tecnologia *legacy* existe na maioria dos dispositivos, e assim sendo não seria necessário implementar *hardware* nem *firmware* na parte do dispositivo central para a criação e gestão da rede.

Bibliografia

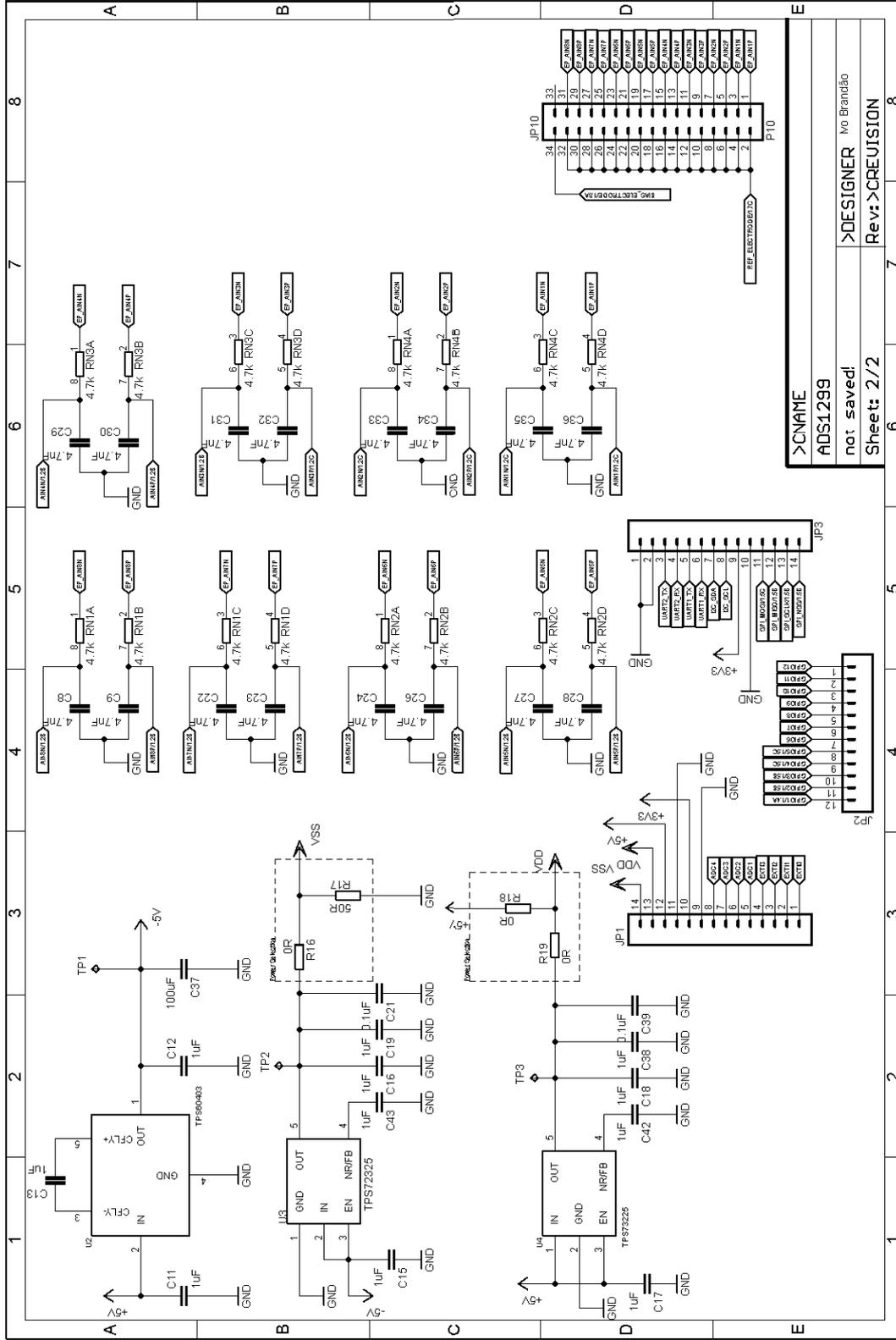
- Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, 28(3), R1.
- Aminof, A. S. G. M. J. (2000). Electroencephalography: brain electrical activity. In *Encyclopedia of Medical Devices and Instrumentation* (pp. 1084–1107).
- Argenox. (2015). BLE Advertisement Primer. Retrieved from <http://www.argenox.com/bluetooth-low-energy-ble-v4-0-development/library/a-ble-advertising-primer/>
- ARM. (2013). CMSIS - Cortex Microcontroller Software Interface Standard. Retrieved May 6, 2015, from <http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>
- Basu, R. (2009). High ambient temperature and mortality: a review of epidemiologic studies from 2001 to 2008. *Environ Health*, 8(1), 40.
- Basu, R., & Samet, J. M. (2002). Relation between elevated ambient temperature and mortality: a review of the epidemiologic evidence. *Epidemiologic Reviews*, 24(2), 190–202.
- Binkley, P. F. (2003). Predicting the potential of wearable technology, engineering in medicine and Biology Magazine, IEEE. *Engineering in Medicine and Biology Magazine, IEEE*, 22(no3), 23–27.
- Bisdikian, C. (2001). An overview of the Bluetooth wireless technology. *IEEE Commun Mag*, 39(12), 86–94.
- Bitalino. (2013). Bitalino. Retrieved February 10, 2015, from <http://www.bitalino.com/index.php/board-kit>
- Bluetooth SIG. (1998). Bluetooth Special Interest Group. Retrieved from <https://www.bluetooth.org>
- Bluetooth SIG. (2010). Generic Attribute Profile (GATT). Retrieved March 9, 2015, from <https://developer.bluetooth.org/TechnologyOverview/Pages/GATT.aspx>
- Bluetooth SIG. (2014a). Bluetooth Qualification and Declaration Processes. Retrieved March 3, 2015, from <https://www.bluetooth.org/en-us/test-qualification/qualification-overview>
- Bluetooth SIG. (2014b). Bluetooth Qualified Design Listings. Retrieved June 7, 2015, from <https://www.bluetooth.org/tpg/listings.cfm>
- BOSCH. (n.d.). *BMP180 Datasheet*. Retrieved from <https://ae-bst.resource.bosch.com/media/products/dokumente/bmp180/BST-BMP180-DS000-12~1.pdf>
- CadSoft Computer. (n.d.). Cadsoft Eagle 7.2.0. Retrieved from <http://www.cadsoftusa.com/>
- Canada, I. of. (n.d.). Industry of Canada - IC. Retrieved June 7, 2015, from <http://www.ic.gc.ca/eic/site/ceb-bhst.nsf/eng/home>
- Carrilho, M. J. C. (2015). Revista de Estudos Demográficos nº 54. *Instituto Nacional de Estatística, I.P.* Retrieved from https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=210786633&PUBLICACOESmodo=2
- CE. (n.d.). Radio and Telecommunication Terminal Equipment (R&TTE) Directive. Retrieved June 7, 2015, from <http://ec.europa.eu/growth/sectors/electrical-engineering/rtte-directive/>
- Ermes, M., Pärkkä, J., Mäntyjärvi, J., & Korhonen, I. (2008). Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1), 20–26. <http://doi.org/10.1109/TITB.2007.899496>
-

-
- ESTI. (n.d.). European Standards Organization - ESTI. Retrieved June 7, 2015, from <http://www.etsi.org/>
- FCC. (n.d.-a). FCC ID. Retrieved June 7, 2015, from <https://www.fcc.gov/fccid>
- FCC. (n.d.-b). FCC-ID HELP. Retrieved June 7, 2015, from <https://www.fcc.gov/fccid#helpSection>
- FCC. (n.d.-c). Federal Communications Commission. Retrieved June 1, 2015, from <https://www.fcc.gov/>
- FCC. (n.d.-d). *Permissive Changes*. Retrieved from [https://apps.fcc.gov/kdb/GetAttachment.html?id=2WVVe%2FJ0PoOiHR8I6M85rGg%3D%3D&desc=178919 D01 Permissive Change Policy v05r04&tracking_number=33013](https://apps.fcc.gov/kdb/GetAttachment.html?id=2WVVe%2FJ0PoOiHR8I6M85rGg%3D%3D&desc=178919%20D01%20Permissive%20Change%20Policy%20v05r04&tracking_number=33013)
- Freescale Semiconductor. (2004). *Using the Serial Peripheral Interface (SPI) eTPU Function* (No. AN2847). Retrieved from http://cache.freescale.com/files/32bit/doc/app_note/AN2847.pdf
- G., W. J.; N. V. T. (2000). Electrocardiographic monitors. In *Encyclopedia of Medical Devices and Instrumentation* (pp. 1002–1017).
- GNU ARM Eclipse. (2014). Packs Manager. Retrieved from <http://gnuarmeclipse.livius.net/blog/packs-manager/>
- Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors (Switzerland)*, *12*(9), 11734–11753. <http://doi.org/10.3390/s120911734>
- Gomez, C., & Paradells, J. (2010). Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, *48*(6), 92–101. <http://doi.org/10.1109/MCOM.2010.5473869>
- Har-Shai, L., Kofman, R., Zussman, G., & Segall, A. (2002). Inter-piconet scheduling in bluetooth scatternets. In *Proc. OPNETWORK* (Vol. 2002).
- IEEE, I. of E. and E. E. (2008). *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*.
- Industry of Canada. (n.d.-a). E-Filling. Retrieved from <http://sms-sgs.ic.gc.ca/eic/site/sms-sgs-prod.nsf/eng/home>
- Industry of Canada. (n.d.-b). RSP-100 — Certification of Radio Apparatus. Retrieved June 7, 2015, from <http://www.ic.gc.ca/eic/site/smt-gst.nsf/eng/sf01130.html#AppendixA>
- Instruments, T. (2015). How to Certify Your Bluetooth product. Retrieved July 6, 2015, from http://processors.wiki.ti.com/index.php/How_to_Certify_your_Bluetooth_product
- L. A. Geddes, L. E. B. (1989). *Principles of Applied Biomedical Instrumentation, 3rd Edition*.
- LTD, A. (n.d.). ARM ltd. Retrieved from <http://www.armltd.co.uk/products/sdt/>
- Luca, C. J.; Webster, & G. De, J. (2000). Electromyography. In *Encyclopedia of Medical Devices and Instrumentation* (pp. 1111–1120).
- Ludovici, A., Calveras, A., & Casademont, J. (2011). Forwarding techniques for IP fragmented packets in a real 6LoWPAN network. *Sensors*, *11*(1), 992–1008. <http://doi.org/10.3390/s110100992>
- Macedo, D. da C. (2010). *Sistema de Extensão de Sinais IR Usando Tecnologia ZigBee*. FEUP. Retrieved from https://web.fe.up.pt/~ee01124/files/dissertacao_deciomacedo.pdf
- Markus Levy. (n.d.). *The History of The ARM Architecture: From Inception to IPO*. Retrieved from <http://reds.heig-vd.ch/share/cours/reco/documents/thehistoryofthearmarchitecture.pdf>
- mems-exchange. (2010). MEMS & Nanotechnology Exchange. Retrieved June 9, 2015, from
-

- <https://www.mems-exchange.org/MEMS/what-is.html>
- Microelectronics, S. (2015). Very low power module for Bluetooth Smart v4.1. Retrieved August 8, 2015, from http://www.st.com/web/catalog/sense_power/FM2185/SC1898/PF261927
- Microsoft. (n.d.). Concurrent Queue. Retrieved June 7, 2015, from [https://msdn.microsoft.com/en-us/library/dd267265\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/dd267265(v=vs.110).aspx)
- Microsoft. (2015). Visual Studio 2015 Community Edition. Microsoft. Retrieved from <https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx>
- Milenković, A., Otto, C., & Jovanov, E. (2006). Wireless sensor networks for personal health monitoring: Issues and an implementation. *Comput. Commun.*, 29(13-14), 2521–2533. <http://doi.org/10.1016/j.comcom.2006.02.011>
- Motorola Inc. (2003). *Motorola SPI Block Guide V03.06*. Retrieved from ftp://ztchs.p.lodz.pl/PIK_POKL/SPI.pdf
- Nellcor. (2003). A Technology Overview of the Nellcor OxiMax Pulse Oximetry System, 8. Retrieved from [http://www.covidien.com/imageServer.aspx/doc226941.1.2.3_OxiMax whitepaper.pdf?contentID=25496&contenttype=application/pdf](http://www.covidien.com/imageServer.aspx/doc226941.1.2.3_OxiMax%20whitepaper.pdf?contentID=25496&contenttype=application/pdf)
- Niedermeyer, E., & da Silva, F. H. L. (2005). *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.
- Nieminen, J., Savolainen, T., Isomaki, M., & Shelby, Z. (2013). Transmission of IPv6 Packets over BLUETOOTH Low Energy. preparation. Retrieved from <http://tools.ietf.org/html/draft-ietf-6lowpan-btle-12#section-2.4>
- Nordic Semiconductor. (2014). *Creating Bluetooth® Low Energy Applications Using nRF51822*. Retrieved from https://www.nordicsemi.com/eng/nordic/download_resource/24020/5/80219615
- Norton, H. B. N. (1996). Handbook of transducers for electronic measuring systems.
- NXP Semiconductors. (2004). *I2C Manual* (No. AN10216-01). Retrieved from http://www.nxp.com/documents/application_note/AN10216.pdf
- NXP Semiconductors. (2007). *UM10204 I2C-bus specification and user manual. Semiconductors* (Vol. 3). Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:I2C-bus+specification+and+user+manual#0>
- Olsson, J. (n.d.). *6LoWPAN demystified*. Retrieved from <http://www.ti.com/lit/wp/swry013/swry013.pdf>
- Plonsey, R. (1999). *Bioelectric Phenomena*. <http://doi.org/10.1002/047134608X.W1403>
- Ray, O’Dea Chad, Hughes Bob, Schmertz Taneika, H. (2012). An Introduction to the ARM System Architecture. Retrieved July 7, 2015, from <https://www.cs.umd.edu/~meesh/cm411/website/proj01/arm/>
- Ryba, R. L. K. A. J. (2000). *Data Structures and program design in C*. (E. in C. M. Horton, A. E. L. Steele, P. E. R. Kernan, M. E. E. Clark, A. D. H. Scott, A. to A. D. J. Christiana, ... I. . Art Production: Blake MacLean (PreTEX, Eds.). Prentice-Hall, Inc.
- S.L., L. C. D. (2014). Bluetooth Low Energy Networking Guide. Retrieved July 7, 2015, from http://www.libelium.com/downloads/documentation/bluetooth-low-energy-networking_guide.pdf
- Sig. (2010). *Bluetooth Special Interest Group*. Retrieved from <https://www.bluetooth.org>
- SIG, B. (2015). Adopted Bluetooth Profiles, Services, Protocols and Transports. Retrieved from <https://www.bluetooth.org/en-us/specification/adopted-specifications>

-
- ST Microelectronics. (n.d.-a). ST Microelectronics. Retrieved March 10, 2015, from <http://www.st.com/web/en/home.html>
- ST Microelectronics. (n.d.-b). STM32 Microcontroller Family. Retrieved October 2, 2015, from <http://www.st.com/web/en/catalog/mmc/FM141/SC1169>
- ST Microelectronics. (2012). *Using floating-point unit (FPU)*. Retrieved from http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/DM00047230.pdf
- ST Microelectronics. (2013). *Technical note ST SPI protocol* (No. TN0897). Retrieved from http://www.st.com/st-web-ui/static/active/en/resource/technical/document/technical_note/DM00054618.pdf
- ST Microelectronics. (2015a). *Getting started with STM32CubeF4 firmware package for STM32F4 Series* (DocID025922 Rev 4 No. DM00107720). Retrieved from http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/user_manual/DM00107720.pdf
- ST Microelectronics. (2015b). Open Stm32. Retrieved from <http://www.openstm32.org/HomePage>
- ST Microelectronics. (2015c). STM32F4Cube. Retrieved March 3, 2015, from <http://www.st.com/web/en/catalog/tools/PF259243>
- ST Microelectronics. (2015d). STM32F4CubeMX. Retrieved from <http://www.st.com/web/en/catalog/tools/PF259242>
- Texas Instruments. (2010). *KeyStone Architecture Serial Peripheral Interface (SPI) User Guide* (No. SPRUGP2A). Retrieved from <http://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf>
- Texas Instruments. (2014a). *ADS1299 - EEG Front-End Performance Demonstration Kit*. Retrieved from <http://www.ti.com/lit/ug/slau443a/slau443a.pdf>
- Texas Instruments. (2014b). Miniaturized Pulse Oximeter Reference Design, (Test report), 1–24. Retrieved from <http://www.ti.com/lit/ug/tidu475/tidu475.pdf>
- Texas Instruments. (n.d.-a). *Cap-Free, NMOS, 250mA Low Dropout Regulator with Reverse Current Protection Datasheet*. Retrieved from <http://www.ti.com/lit/ds/symlink/tps732.pdf>
- Texas Instruments. (n.d.-b). *TMP102 Datasheet*. Retrieved from <http://www.ti.com/lit/ds/sbos397b/sbos397b.pdf>
- Texas Instruments. (2013). *ADS1299 Datasheet*. Retrieved from <http://www.ti.com/lit/ds/symlink/ads1299.pdf>
- Texas Instruments. (2015). *TPS6040x Unregulated 60-mA Charge Pump Voltage Inverter*. Retrieved from <http://www.ti.com/lit/ds/symlink/tps60402.pdf>
- Thakor, N. V, & Webster, J. G. (1985). Electrode studies for the long-term ambulatory ECG. *Medical and Biological Engineering and Computing*, 23(2), 116–121.
- The Eclipse Foundation. (2015). Eclipse IDE. Retrieved from <https://eclipse.org/>
- Trevor, M. S. L. (2009). The Inside's Guide To the STM32 ARM Based Microcontroller. Retrieved June 1, 2015, from <http://www.hitex.com/fileadmin/pdf/insiders-guides/stm32/isg-stm32-v18d-scr.pdf>
- Webster, J. G. (1984). Reducing motion artifacts and interference in biopotential recording. *Biomedical Engineering, IEEE Transactions on*, (12), 823–826.
- West, A. (2013). Smartphone, the key for Bluetooth low energy technology. Retrieved from <http://www.bluetooth.com/Pages/Smartphones.aspx>
- ZigBee Alliance Organization. (n.d.). ZigBee Alliance. Retrieved June 8, 2015, from <http://www.zigbee.org/>
-

ZigBee Standard Organization. (2012). *ZIGBEE SPECIFICATION*. Retrieved from <http://www.zigbee.org/download/standards-zigbee-specification/>



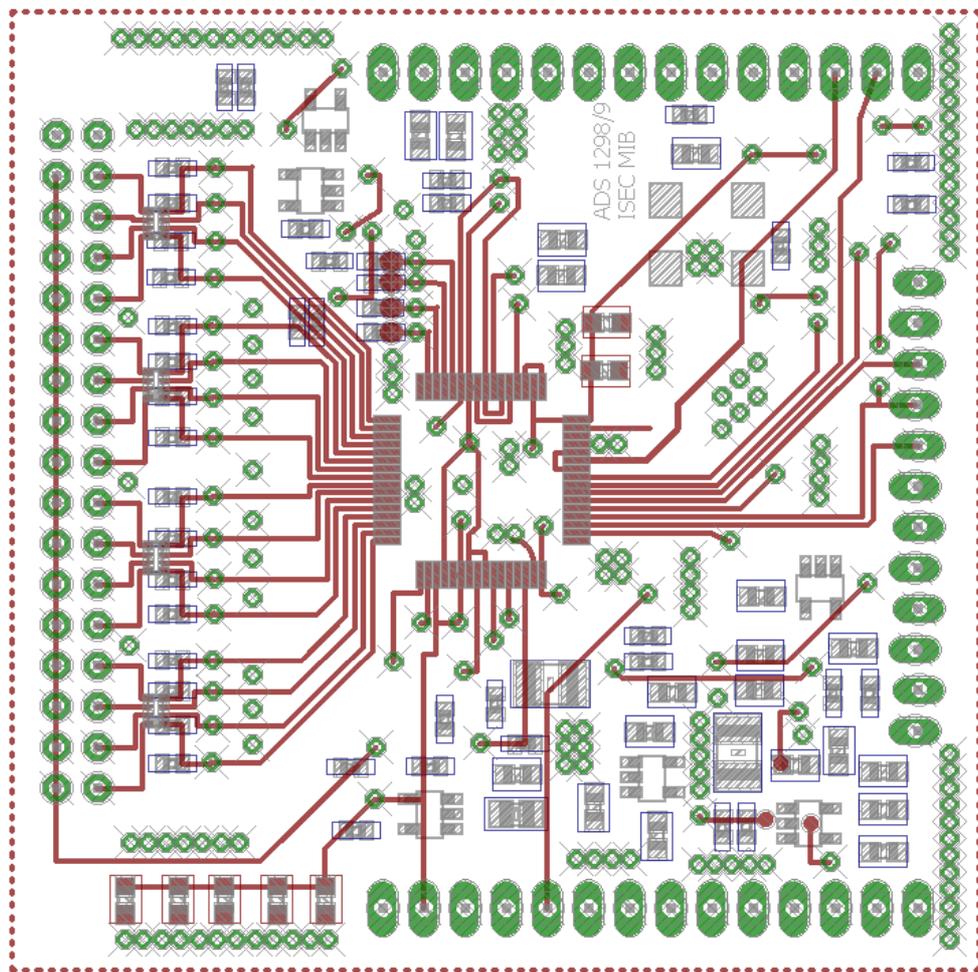
>CNAME
 ADS1299
 not saved!
 Sheet: 2/2

>DESIGNER Ivo Brandão
 >Rev: >CREUISION

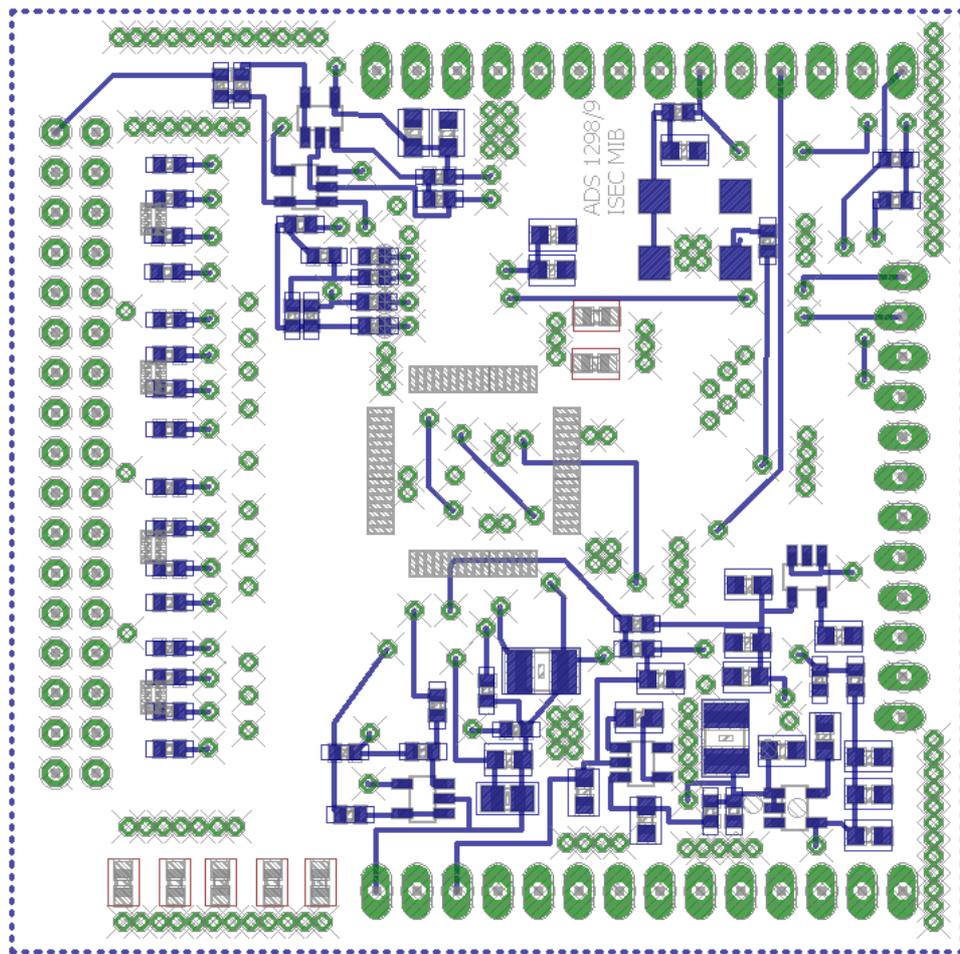
Anexo II – Bill Of Materials AFE

Qty	Line Item	Manufacturer	MPN	Schematic Reference	Description
3	OPA376AIDBVTG4-ND	Texas Instruments	OPA376AIDBVTG4	A1, A2, A3	OPA376AIDBVTG4 OP AMP, LOW NOISE, LOW LQ, SOT-23-5
2	CRCW04022M00FKED	Vishay Draloric	CRCW04022M00FKED	R3, R4	THICK FILM RESISTOR CHIP, 2MOHM 63mW, 1%
1	CRCW0402390KFKED	Vishay Draloric	CRCW0402390KFKED	R7	THICK FILM RESISTOR, 390KOHM, 63mW, 1%
2	CRCW0402000Z0EDHP	Vishay Dale	CRCW0402000Z0EDHP	R5, R6, R13, R14, R15, R16, R17, R18, R19	THICK FILM RESISTOR, 0 OHM, 0.125W
13	C1005X5R1V105K050BC	EPCOS	C1005X5R1V105K050BC	C21, C23, C6, C8, C20, C11, C12, C15, C16, C17, C18, C3, C5, C25, C42, C43	CAP, MLCC, X5R, 1UF, 35V, 0402
1	GRM219R61A106KE44D	Murata	GRM219R61A106KE44D	C1	CAP, MLCC, X5R, 10UF, 10V, 0805
1	C1005X7R1H104K050BB	EPCOS	C1005X7R1H104K050BB	C2, C19, C22, C24, C7, C9, C14	CERAMIC CAPACITOR, 0.1UF, 50V, X7R, 0402
1	LMK325BJ107MM-T	Taiyo Yuden	LMK325BJ107MM-T	C4	CAP, MLCC, X5R, 100UF, 10V, 1210
1	CC0402KRX7R9BB103	Yageo / Phycomp	CC0402KRX7R9BB103	C10	CERAMIC CAPACITOR 0.01UF, 50V, X7R, 10%, 0402
1	FXO-HC736R-2.048	Fox Electronics	FXO-HC736R-2.048	X1	OSC, 5X7MM, SMD, CER, 2.048MHZ
4	EXBN8V472JX	Panasonic	EXB-N8V472JX	RN1, RN2, RN3, RN4	RESISTOR, CCAV ARRAY, 0402X4, 4K7 5%
1	ADS1299IPAG	Texas Instruments	ADS1299IPAG	U1	ADS1299IPAG ANALOG FRONT END, BIO POTENTIAL MEASUREMENT, TQFP-64
16	CC0402KRX7R9BB472	Yageo	CC0402KRX7R9BB472	C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41	CERAMIC CAPACITOR 4700PF, 50V, X7R, 10%, 0402
1	TPS60403DBVT	Texas Instruments	TPS60403DBVT	U2	TPS60403DBVT CHARGE PUMP INVERTER, SMD, 60403
1	TPS72325DBVT	Texas Instruments	TPS72325DBVT	U3	TPS72325DBVT REGULATOR, LDO, FIXED, -2.5V, SOT-23-5
1	TPS73225DBVT	Texas Instruments	TPS73225DBVT	U4	TPS73225DBVT V REG LDO 0.25A +2.5V, SMD, 73225
3	D01-9972042	Harwin	D01-9972042	P1, P2, P3	HARWIN D01-9972042 SOCKET, SIL, 20WAY (Find Other Options)
1	MC34773	Multicomp	MC34773	P4	MULTICOMP MC34773 HEADER, 2 ROW, R/ANGLE, 16WAY

Anexo III – PCB Modulo de Aquisição de Sinais Bio – Potenciais (Top Layer)



Anexo III – PCB Modulo de Aquisição de Sinais Bio Potenciais (Bottom Layer)



Anexo IV – Função de Cálculo CRC 8 bits

Função de cálculo do CRC para a verificação dos dados recebidos do sensor HTU21D.

```
/**
 *-----
 * @brief Check CRC for the HTU21D POLYNOMIAL = 0x0131 = x^8 + x^5 + x^4 + 1
 * @param data [IN] Data that was read from the sensor
 * @param RxCRC [IN] crc received byte
 * @retval 0 - if the CRC is OK, other value if CRC is not OK
 *-----
 */
uint32_t HTU21D_Check_CRC(uint16_t data, uint8_t RxCRC) {

    uint32_t rest = (uint32_t) data << 8;
    rest |= RxCRC;

    /* 0x988000= 0x0131 polynomial shifted to the left by three bytes */
    uint32_t div = (uint32_t) 0x988000;
    uint8_t i;

    for (i = 0; i < 16; i++) {
        if (rest & (uint32_t) 1 << (23 - i))
            rest ^= div;
        div >>= 1;
    }
    /*Check if the returned value is different from zero */
    return (rest);
}
```

Anexo V – Função de Cálculo Frame Check Sequence

Função de cálculo do *frame check sequence* (FCS) utilizado na transmissão de mensagens

```
/*!
 *-----
 * @brief      Get the value of the XOR of the bytes passed in the field data
 * @param      *data [IN]   bytes from which to get the checksum
 * @param      len [IN]   number of bytes to perform the XOR
 * @retval     the XOR value of the number of bytes specified by len field in data field
 *-----
 */
uint8_t GetFrameCheckSequence(uint8_t *data, uint16_t len) {
    uint8_t Checksum = 0;

    for (int i = 0; i < len; i++) {
        Checksum ^= data[i];
    }
    return (Checksum);
}
```