

**DAS** Departamento de Automação e Sistemas  
**CTC** Centro Tecnológico  
**UFSC** Universidade Federal de Santa Catarina

# **Desenvolvimento de um Dispositivo de Medição dos Potenciais Cardíacos**

*Monografia submetida à Universidade Federal de Santa Catarina  
como requisito para a aprovação da disciplina:*

***DAS 5511: Projeto de Fim de Curso***

***Diogo Furukawa***

*Florianópolis, fevereiro de 2014.*



# **Desenvolvimento de um Dispositivo de Medição dos Potenciais Cardíacos**

*Diogo Furukawa*

Esta monografia foi julgada no contexto da disciplina  
**DAS 5511: Projeto de Fim de Curso**  
e aprovada na sua forma final pelo  
**Curso de Engenharia de Controle e Automação**

*Prof. Jefferson Luiz Brum Marques - Eng, MSc, PhD*

---

Assinatura do Orientador



Banca Examinadora:

Lucas Casagrande Neves, Eng, MSc  
*Orientador na Empresa*

Prof. Jefferson Luiz Brum Marques, Eng, MSc, PhD  
*Orientador na UFSC*

Prof. Ricardo José Rabelo, Dr.  
*Responsável pela disciplina*



# Agradecimentos

Gostaria de agradecer à minha família, principalmente aos meus pais e irmã, por serem pessoas com quem eu sempre posso contar, em todos os momentos, e mesmo estando longe saber que há um lugar para onde ir, um motivo para seguir em frente. Também agradeço aos amigos, os que estão perto e os que estão longe, por tornarem os dias muito mais agradáveis, com as conversas, as graças, os conselhos, a companhia.

Agradeço também aos meus chefes Lucas, Gabriel e Jonatas, pela oportunidade de trabalhar em uma área tão interessante e bela da engenharia, e pela paciência e prontidão com que se dispõe a me ajudar e solucionar minhas dúvidas.

Por fim, a todos os professores que fizeram parte da minha formação. Agradeço por direta ou indiretamente fornecerem não apenas conhecimento, mas também lições de vida, formando não somente o intelecto, mas também o caráter.





# Abstract

The electrocardiogram is an exam used to evaluate the behaviour and condition of the heart from either humans or other animals. With it, it is possible for the specialist physician to detect issues like arrhythmia, infarct and other types of cardiopathies. Even though there is a high mortality rate of animals due to heart diseases, the use of such kind of exam in animals is way below ideal. That comes mainly from the lack of specialized professionals capable of evaluating these exams.

In this scenario, InPulse, an engineering enterprise based in Florianópolis - Brazil, and specialized in biomedical equipment development, started the development of a veterinary telemedicine system. This system's objective is to provide a tool that allows a remote diagnostic. With this system, the doctor that makes the diagnostic does not need to leave his place to evaluate the patients, which can be kilometers away. As a fundamental part of this system, an electrocardiograph was developed – an electronic equipment capable of acquiring ECG signals and feed it to a software for user interaction.

Therefore, this work refers to the development of the device, pursuing the portability, high resolution and flexibility as project requirements. The deployment begins at the component and modules selection and goes all the way to the development of a firmware to integrate every part of the device.

Aiming to detailing the steps on the equipment development, in this report are presented the main concepts, tools and procedures utilized during the execution of the project.



# Resumo

O exame de eletrocardiograma (ECG) é realizado para avaliar o funcionamento e estado do coração do paciente, seja ele humano ou algum outro animal. Através do exame, o médico especialista é capaz de detectar problemas como arritmias, infartos e outras cardiopatias. Apesar das altas taxas de mortalidade de animais decorrente de cardiopatias, a realização desse tipo de exame em animais é ainda muito abaixo do ideal. Isso se dá, em grande parte, pela falta de profissionais especializados capazes de laudar exames desse tipo.

Diante desse panorama, a InPulse, empresa de engenharia sediada em Florianópolis, especializada no desenvolvimento de equipamentos biomédicos, deu início ao desenvolvimento de um sistema de telemedicina veterinária, cujo objetivo é proporcionar o diagnóstico de cardiopatias à distância. Com esse sistema, não há necessidade de o médico laudista se deslocar de seu local de trabalho para diagnosticar pacientes, os quais podem estar a quilômetros de distância. Como parte fundamental desse sistema, foi desenvolvido um dispositivo eletrocardiógrafo, um equipamento eletrônico capaz de adquirir os sinais de ECG e disponibilizá-lo para o *software* de interface com o usuário.

Portanto, o presente trabalho diz respeito ao desenvolvimento desse equipamento, cujos principais requisitos de projeto são: portabilidade, alta resolução e flexibilidade. Iniciando-se na escolha dos componentes a serem utilizados, o seu desenvolvimento culmina na codificação de um *firmware* que integra todos os módulos do dispositivo.

Com o objetivo de detalhar as etapas de desenvolvimento do equipamento, neste documento são apresentados os principais conceitos, ferramentas e procedimentos utilizados durante execução desse projeto.



# Lista de Figuras

1.1	Equipamento ECG antigo. . . . .	2
1.2	Equipamentos presentes no mercado. . . . .	3
2.1	Circulação do sangue no corpo. . . . .	5
2.2	Coração Humano. . . . .	6
2.3	Potencial de ação de uma célula cardíaca. . . . .	7
2.4	Gráfico de uma onda de ECG típica relacionada às fases do ciclo cardíaco. . . . .	8
2.5	Triângulo de Einthoven e respectivas derivações. . . . .	9
2.6	Terminal Central de Wilson. . . . .	10
2.7	Exemplo de derivação aumentada de Goldberger. . . . .	10
2.8	Posicionamento dos eletrodos precordiais e bipolares. . . . .	11
2.9	Posicionamento dos eletrodos precordiais com o sistema de 4 derivações. . . . .	12
2.10	Posicionamento dos eletrodos precordiais com o sistema de 6 derivações. . . . .	12
2.11	Esquemático da conexão das linhas utilizadas na comunicação serial. . . . .	13
2.12	Formato de um frame, incluindo bits opcionais. . . . .	14
2.13	Deslocamento dos bits, em uma comunicação SPI. . . . .	15
2.14	Conexão com $\overline{SS}$ individuais para cada escravo. . . . .	15
2.15	Conexão <i>Daisy-chain</i> . . . . .	15
2.16	Esquema de conexão de hubs à uma porta USB, destacando-se o máximo de 6 camadas. . . . .	17
2.17	Conectores USB mais comuns. . . . .	17
2.18	Conexão I <sup>2</sup> C. . . . .	19
3.1	Diagrama com os principais módulos a serem implementados no dispositivo. . . . .	21
3.2	Operação de leitura de dados do ADS. . . . .	24
3.3	Protocolo <i>Bluetooth</i> implementado pelo <i>iWRAP</i> . . . . .	25
4.1	Fluxograma do comportamento desejado do ECG. . . . .	27
4.2	Detalhamento do fluxograma, referente ao loop principal do <i>firmware</i> . . . . .	28

4.3	Comportamento do dispositivo quando sem conexão. . . . .	29
4.4	Comportamento do dispositivo quando em modo <i>Idle</i> . . . . .	29
4.5	Comportamento do dispositivo em modo de Exame ou Visualização. . .	30
4.6	Conexão da placa com o PIC e o computador, através do adaptador USB-Serial. . . . .	31
4.7	Módulo <i>Bluetooth</i> conectado ao computador via adaptador USB-Serial.	31
4.8	Conexão PIC e WT11i-A. . . . .	32
4.9	Kit de desenvolvimento do ADS1298. . . . .	32
4.10	Conexão PIC e ADS1298. . . . .	33
4.11	Conexão USB do PIC18F47J53. . . . .	34
4.12	Migração para o PIC18F47J53. . . . .	34
4.13	Comportamento dos botões visto em um osciloscópio. . . . .	35
4.14	Esquemático da conexão dos botões com o PIC. . . . .	36
4.15	Esquemático da conexão PIC + BQ27541-G1. . . . .	36
4.16	Esquemático de todas as conexões com o PIC. . . . .	37
4.17	Primeira versão da placa. . . . .	38
4.18	Exemplo de uma troca de mensagens entre <i>software</i> e <i>hardware</i> . . . .	38
5.1	Equipamento aberto. . . . .	41
5.2	Conjuntos enviados para a realização de testes. . . . .	42
5.3	Sinais captados com diferentes equipamentos (derivações bipolares) .	43
5.4	Tela do programa durante aquisição (derivações bipolares). . . . .	43
5.5	Tela do programa durante aquisição (derivações aumentadas). . . . .	44
5.6	Tela do programa durante aquisição (derivações precordiais). . . . .	44
5.7	Tela de análise dos sinais (sinais promediados) . . . . .	45
5.8	Equipamento detector de sinais autonômicos. . . . .	45

# Índice

<b>Agradecimentos</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Resumo</b>	<b>v</b>
<b>Lista de Figuras</b>	<b>vii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivo . . . . .	2
1.3 A InPulse . . . . .	3
1.4 Estrutura do Trabalho . . . . .	4
<b>2 Fundamentação Teórica</b>	<b>5</b>
2.1 O Coração . . . . .	5
2.1.1 Anatomia . . . . .	5
2.1.2 Ciclo Cardíaco . . . . .	7
2.1.3 Impulso Cardíaco e Potencial de Ação . . . . .	7
2.2 O Eletrocardiograma . . . . .	8
2.2.1 Derivações Bipolares . . . . .	9
2.2.2 Derivações Unipolares e Aumentadas . . . . .	9
2.2.3 Derivações Precordiais . . . . .	10
2.3 Interfaces de Comunicação . . . . .	13
2.3.1 Interface UART . . . . .	13
2.3.2 Interface SPI . . . . .	14
2.3.3 Interface USB . . . . .	16
2.3.4 Interface I <sup>2</sup> C . . . . .	18

<b>3</b>	<b>Análise de Requisitos para os Módulos Funcionais</b>	<b>21</b>
3.1	Principais Módulos Utilizados . . . . .	22
3.1.1	Microcontrolador: PIC18F47J53 . . . . .	22
3.1.2	Front-end: ADS1298 . . . . .	22
3.1.3	Módulo de Comunicação Bluetooth: WT11i-A . . . . .	24
3.1.4	Fuel Gauge: bq27541-G1 . . . . .	25
<b>4</b>	<b>Implementação</b>	<b>27</b>
4.1	Fluxo do <i>Firmware</i> . . . . .	27
4.2	Desenvolvimento dos Módulos . . . . .	30
4.2.1	Interface UART e Módulo <i>Bluetooth</i> . . . . .	30
4.2.2	Interface SPI e ADS1298 . . . . .	32
4.2.3	Interface USB . . . . .	33
4.2.4	Migração para o PIC18F47J53 . . . . .	34
4.2.5	Adição dos Botões de Ligar/Desligar e <i>Master Clear</i> . . . . .	35
4.2.6	Interface I <sup>2</sup> C e BQ27541-G1 . . . . .	36
4.2.7	Esquemático das Conexões do PIC com os Periféricos . . . . .	37
4.2.8	Primeira Versão da Placa do Dispositivo . . . . .	37
4.3	Protocolo de Comunicação . . . . .	38
<b>5</b>	<b>Resultados</b>	<b>41</b>
<b>6</b>	<b>Conclusão e Perspectivas</b>	<b>47</b>
	<b>Referências Bibliográficas</b>	<b>49</b>



# Capítulo 1: Introdução

O mercado *pet* cresce rapidamente. Segundo dados da Abinpet, Associação Brasileira da Indústria de Produtos para Animais de Estimação, o setor faturou em 2012 R\$14,2 bilhões, um crescimento de 16,4% em relação ao ano anterior. Não somente as parcelas referentes à alimentação, beleza e entretenimento têm crescido, mas também a parcela relacionada ao bem estar e à saúde dos animais.

Doenças cardíacas são bastante comuns em cães. Estima-se que cerca de 10% dos cães apresentem algum tipo de doença cardíaca [1]. Esse número cresce com a idade dos cães: a incidência de valvopatia crônica, por exemplo, chega a 60% em cães de idade avançada [2]. A evolução do setor de cardiologia veterinária tem sido essencial para que esses animais tenham suas vidas prolongadas.

O aparelho eletrocardiógrafo é parte essencial no diagnóstico de doenças cardíacas. Entretanto, a área ainda está em crescimento no Brasil e não há número tão grande de médicos veterinários especialistas, e portanto, aptos a analisar os resultados de um eletrocardiograma. Além disso, a maioria dos equipamentos existentes no mercado deixa a desejar em alguns aspectos. Em geral, principalmente no ramo veterinário, os aparelhos comercializados apresentam baixa resolução, ou possuem dimensões que inviabilizam sua portabilidade.

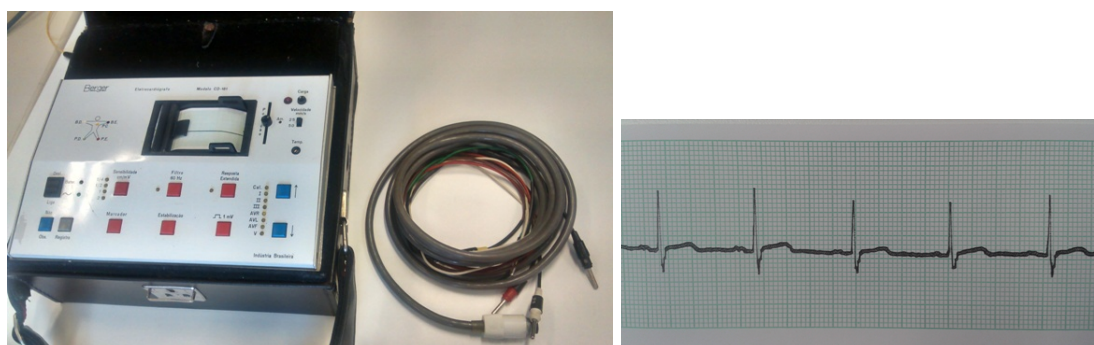
O projeto do ECG propõe um sistema de telemedicina veterinária – um sistema de diagnósticos à distância, em que o médico que analisa o exame não precisa estar no local onde é realizada a aquisição dos sinais do animal. Para isso, um *software* de aquisição com as funcionalidades requeridas para a telemedicina foi desenvolvido, em conjunto com um equipamento eletrocardiógrafo portátil e de alta resolução. Este trabalho se concentra na implementação do equipamento, mais precisamente na programação do *firmware* embarcado no aparelho.

## 1.1 Motivação

A idéia de um aparelho eletrocardiógrafo portátil se dá principalmente pela possibilidade de realização de exames *in loco*, no ambiente em que o animal está inserido e acostumado, seja sua casa no caso de animais domésticos, ou haras e fazendas no caso de animais maiores. Dessa forma elimina-se um grande problema na efetuação desse tipo de exame, que é a retirada do animal de sua zona de conforto e

consequente agitação, muitas vezes impossibilitando que os exames sejam até mesmo iniciados.

Além disso, pode-se dizer que o desenvolvimento de dispositivos do tipo está, de certa forma, estagnado. Os aparelhos presentes no mercado hoje em dia, não diferem muito daqueles produzidos há 20, 30 anos atrás. Na Figura 1.1 pode-se ver um equipamento de 25 anos, produzido pela *Berger*. Esse equipamento utiliza papel termossensível e uma agulha resistiva para traçar o sinal eletrocardiográfico. Cada derivação é captada separadamente, para isso os botões de seleção.



(a) Equipamento em maleta para mobilidade (b) Exame realizado com o equipamento

Figura 1.1: Equipamento ECG antigo.

Os dispositivos de hoje são variações desse exemplo, como pode ser visto na Figura 1.2. Há ECGs que utilizam canetas e papel milimetrado para imprimir o exame, em um processo similar ao descrito acima – a aquisição acontece antes, e pode ser vista via computador. Outros, possuem telas para visualização do sinal, mas também são aparelhos grandes e pouco portáteis. Na área veterinária, os aparelhos são adaptações dos utilizados para medicina humana, de certa forma um pouco menos sofisticados.

Nesse contexto, busca-se produzir um aparelho especificamente desenvolvido para a área veterinária, que satisfaça as necessidades dos profissionais da área e auxilie na evolução e popularização dos exames cardiológicos e diagnósticos de doenças cardíacas em animais.

## 1.2 Objetivo

Esse projeto tem como objetivo o desenvolvimento do dispositivo eletrocardiógrafo descrito acima. Busca-se uma portabilidade tal que o aparelho possa ser



(a) Com caneta



(b) Com impressora



(c) Com tela de visualização e impressora



(d) Com tela de visualização e impressora

Figura 1.2: Equipamentos presentes no mercado.

carregado no bolso, e resolução de, no mínimo, 500SPS (*Samples per second* – amostras por segundo).

Procura-se também oferecer uma maior flexibilidade na comunicação, oferecendo uma interface de comunicação bastante comum – o USB – e uma interface *wireless* que permita a realização de exames de forma diferente, possibilitando até mesmo que o animal fique solto durante a aquisição (com o dispositivo preso ao corpo) – a tecnologia *Bluetooth*.

## 1.3 A InPulse

A InPulse foi fundada em 2011 por três engenheiros de controle e automação formados pela Universidade Federal de Santa Catarina. A empresa atua na geração de soluções tecnológicas com foco em projetos de sistemas médicos para atender às necessidades dos profissionais da saúde, tanto no diagnóstico como na terapia de enfermidades de alta abrangência. A empresa é especializada no projeto de sistemas embarcados críticos voltados para a área de engenharia biomédica e no desenvolvimento de aplicações de *software* e sistemas para a área da saúde, possuindo experiência no desenvolvimento de sistemas críticos embarcados. Recentemente, a empresa iniciou suas operações para a área veterinária.

Desde sua formação, a InPulse tem se destacado pela qualidade de suas

soluções inovadoras. Em 2011, recebeu um prêmio da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC), relativo a um de seus principais projetos: um inibidor implantável de apetite para o tratamento da obesidade. Em 2012, com a continuidade do projeto e devido ao alto potencial de sucesso do método proposto, a empresa foi novamente premiada, dessa vez pelo Programa de Apoio à Pesquisa em Empresas - PAPPE. Em 2013, a empresa teve um de seus projetos classificado como o melhor projeto do Brasil, através do programa Startup-Brasil, alcançando nota máxima pela avaliação do CNPq. Tais conquistas levaram a empresa a ser uma das selecionadas pelo SEBRAE para apresentar no Vale do Silício seus desenvolvimentos, como forma de aumentar o grau de atração de investimentos em tecnologia de ponta brasileira.

A InPulse hoje faz parte da Incubadora de Empresas MIDI Tecnológico, da Associação Catarinense de Empresas de Tecnologia (ACATE), que agrupa um conjunto de serviços de apoio em cursos de formação, assessoria de imprensa e consultorias permanentes nas áreas financeira, de marketing, jurídica e de RH.

## 1.4 Estrutura do Trabalho

Cada capítulo deste documento reúne conteúdos similares, ou que possuem o mesmo propósito. O Capítulo 2 traz uma breve fundamentação teórica, explicando conceitos de importância para o entendimento do projeto como um todo, e também especificamente para a parte desenvolvida nesse trabalho. O Capítulo 3 foca nos principais materiais utilizados no projeto. O Capítulo 4 explica como se deu a implementação do *firmware* proposto. O Capítulo 5 mostra os resultados obtidos com o desenvolvimento do dispositivo. Por fim, o Capítulo 6 traz as conclusões e perspectivas futuras de continuação do projeto como um todo.

# Capítulo 2: Fundamentação Teórica

Este capítulo busca explicar conceitos importantes que ajudam na compreensão do projeto e seu embasamento teórico.

## 2.1 O Coração

O coração é o principal órgão do sistema cardiovascular. É dele a função de bombear o sangue para todos os tecidos do corpo. Trata-se de um músculo com cavidades internas – 4 nos mamíferos – que ao se contrair e dilatar, nos movimentos conhecidos respectivamente como sístole e diástole, impulsiona o sangue para diferentes partes do corpo. Em cães o peso do coração varia com relação às diferentes raças e características como idade e sedentarismo, variando de 0,9 a 2,2% do peso do animal [3].

### 2.1.1 Anatomia

As cavidades do coração são divididas em átrios e ventrículos, direito e esquerdo. O lado direito do coração é responsável por enviar o sangue venoso aos pulmões, onde ocorrem as trocas gasosas e o sangue se torna arterial. O lado esquerdo então bombeia esse sangue para o resto do corpo, como pode ser visto na Figura 2.1.

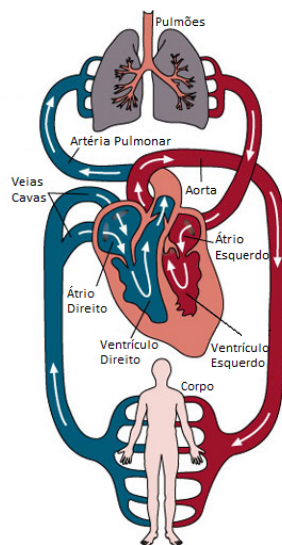


Figura 2.1: Circulação do sangue no corpo.

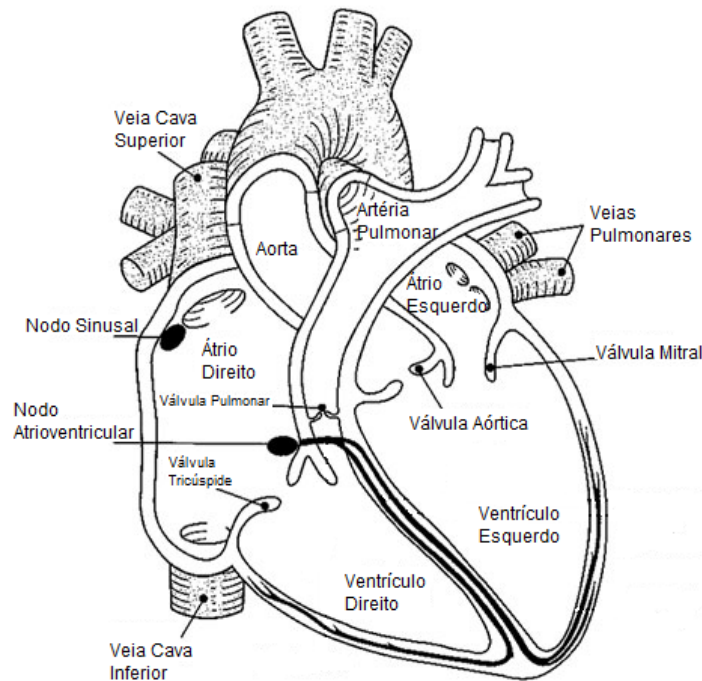


Figura 2.2: Coração Humano.

Válvulas garantem que não haja refluxo do sangue, ao mesmo tempo em que permitem sua passagem para as cavidades seguintes nos momentos corretos. As principais são: tricúspide (por apresentar 3 cúspides, ainda que em cães não haja consenso sobre a existência ou não da terceira cúspide), pulmonar, mitral e aórtica [4]. A válvula tricúspide impede o refluxo do ventrículo direito para o átrio direito, e se abre durante a sístole atrial para que o sangue flua para o ventrículo. A válvula mitral tem a mesma função do lado esquerdo do coração. Já as válvulas pulmonar e aórtica garantem que o sangue não retorne pelas artérias pulmonar e aorta, e se abrem durante a sístole ventricular para que o sangue flua para elas.

Outras estruturas de importância para este caso são os nodos sinusal e atrioventricular. No nodo sinusal é gerado o estímulo elétrico que causa a batida do coração. Ele funciona como um marca-passo biológico [3]. O nodo atrioventricular é o que coordena a relação entre as contrações atriais e ventriculares, de forma a otimizar o volume de sangue ejetado. Na Figura 2.2 pode-se ver as estruturas citadas.

Todas essas estruturas presentes no sistema cardiovascular podem ser, de uma maneira direta ou indireta, analisados pelo médico especialista através de exames específicos, dos quais a eletrocardiografia é parte fundamental.

### 2.1.2 Ciclo Cardíaco

O ciclo cardíaco consiste na sequência completa de sístole e diástole do coração. Essa sequência evolui da seguinte forma: sístole atrial em conjunto com diástole dos ventrículos, diástole atrial e sístole ventricular e finalizado pelo relaxamento completo do coração. Cada ciclo inicia-se com a geração de um estímulo no nodo sinusal. Esse impulso provoca a sístole atrial, e atinge o nodo atrioventricular, que transmite o estímulo para os ventrículos com um atraso para que os ventrículos possam ser preenchidos pelo sangue vindo dos átrios. O coração fica em repouso até o próximo impulso.

### 2.1.3 Impulso Cardíaco e Potencial de Ação

As células cardíacas possuem um potencial de repouso de  $-90\text{mV}$ . Isso se dá devido à presença de íons de potássio e sódio no interior e exterior das células. Devido à diferença nas concentrações, os íons de potássio tendem à sair das células por difusão. Já os íons de sódio tendem à entrar. Essa troca de íons termina por deixar o interior das células com carga negativa, visto que mais íons de potássio migram para o exterior do que os de sódio para o interior. Neste estado, as células se encontram polarizadas [3].

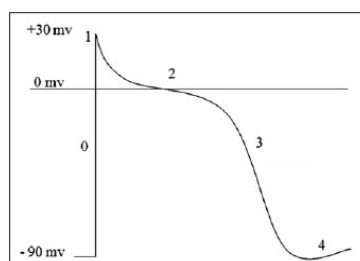


Figura 2.3: Potencial de ação de uma célula cardíaca.

Quando o estímulo gerado no nodo sinusal atinge a célula, o potencial vai, em milissegundos, de  $-90\text{mV}$  a  $+30\text{mV}$ . Tão logo ocorreu a despolarização, inicia-se a repolarização da célula, como pode ser visto na Figura 2.3. Esse processo de despolarização e repolarização é chamado potencial de ação, e é dividido em 5 fases [3]:

**Fase 0:** Despolarização, causada por uma rápida entrada de íons de sódio devido a um aumento na permeabilidade a esses íons na membrana celular;

**Fase 1:** Repolarização precoce, quando a permeabilidade ao sódio retorna ao normal;

**Fase 2:** Repolarização lenta, caracterizada pela estabilização em torno do potencial zero;

**Fase 3:** Repolarização rápida, em que há o aumento da permeabilidade para íons de potássio, que levam o potencial de volta à  $-90\text{mV}$ ;

**Fase 4:** Repouso elétrico, período em que o potencial se mantém em  $-90\text{mV}$  até que ocorra um novo estímulo.

A soma dos potenciais de ação de todas as células do coração propagados pelo corpo é o chamado sinal eletrocardiográfico.

## 2.2 O Eletrocardiograma

O eletrocardiograma (ECG) surgiu ao final do século XIX, quando fisiologistas descobriram que era possível registrar a atividade cardíaca utilizando eletrodos sobre a pele. Porém, apenas no início do século XX surgiu o primeiro eletrocardiógrafo funcional. Seu inventor foi Willem Einthoven (1860-1927), fisiologista holandês [4]. Sua máquina utilizava um cristal de quartzo ligado a eletrodos, que captavam os impulsos elétricos gerados pelas contrações do coração. Como resposta para os impulsos, o cristal obstruía um raio de luz que registrava em papel fotográfico as ondas obtidas. Como eletrodos eram utilizados baldes contendo solução salina onde os pacientes inseriam seus braços e pernas [5].

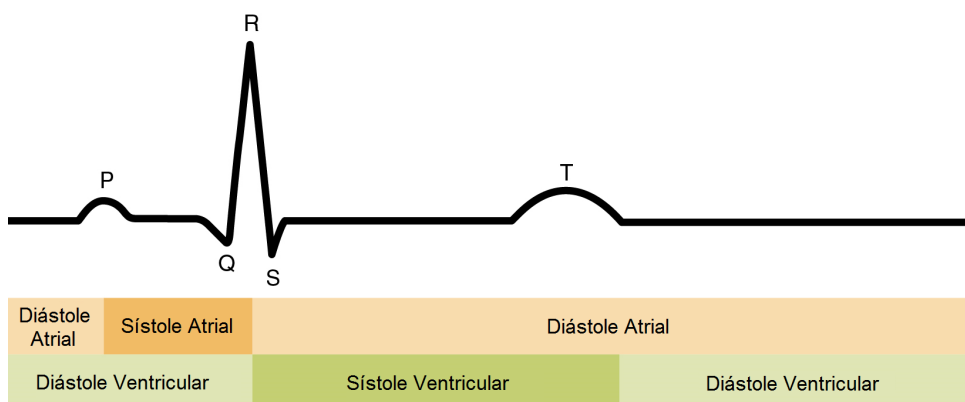


Figura 2.4: Gráfico de uma onda de ECG típica relacionada às fases do ciclo cardíaco.

O ECG é a medição dos potenciais cardíacos. Atualmente, o modo mais comum para a realização desse tipo de exame é feito utilizando eletrodos colados ou fixados sobre a pele do paciente. Com as informações obtidas no ECG é possível avaliar o funcionamento do coração e diagnosticar a presença ou ausência de cardiopatias.



Na Figura 2.4 pode-se ver uma onda de ECG teórica típica, com todos os complexos (ondas) destacados. Vale ressaltar que o exame de ECG é basicamente o mesmo para humanos e demais mamíferos, o que muda são as características observadas nos sinais obtidos.

### 2.2.1 Derivações Bipolares

As derivações bipolares saem do chamado “Triângulo de Einthoven”, um triângulo imaginário ao redor do coração, equivalente à posicionar os eletrodos nos braços e perna esquerda, como feito por Einthoven, que pode ser visto na Figura 2.5. Os lados do triângulo correspondem às derivações, que são calculadas pela subtração entre pares de eletrodos. O eletrodo de perna direita (RL) é utilizado para reduzir efeitos de ruídos de modo comum, injetando no corpo um sinal com o objetivo de anular tais ruídos.

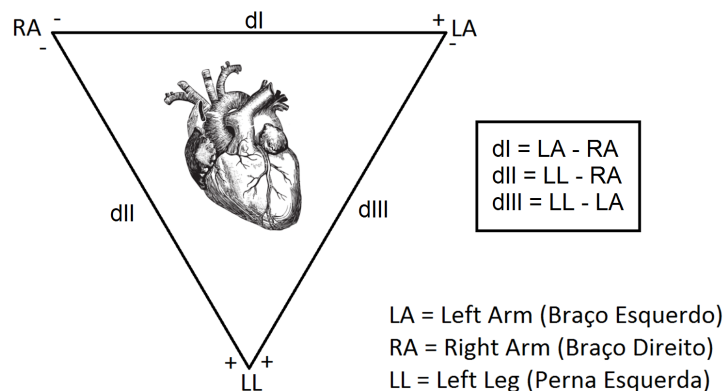


Figura 2.5: Triângulo de Einthoven e respectivas derivações.

Como as derivações captam a diferença de potencial entre dois pontos distintos do corpo, pode-se dizer que são como vistas da atividade elétrica do coração de diferentes pontos. Assim sendo, problemas encontrados em cada derivação podem ser mapeados para diferentes áreas do coração. As derivações dII e dIII são vistas da parte inferior do coração, enquanto a derivação dI é uma vista superior/lateral. Problemas em dII ou dIII podem indicar, por exemplo, um infarto da parede inferior do miocárdio [6, 7].

### 2.2.2 Derivações Unipolares e Aumentadas

Frank Norman Wilson (1890-1952), cardiologista americano, criou o chamado Terminal Central de Wilson (WCT). O WCT consiste na conexão dos 3 eletrodos do

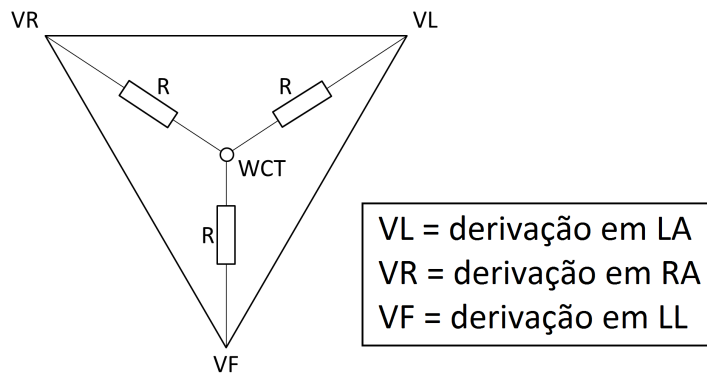


Figura 2.6: Terminal Central de Wilson.

triângulo de Einthoven através de resistências iguais, para obter um ponto de potencial considerado nulo – pode ser visto na Figura 2.6. As medições são então feitas com o WCT como referência – as chamadas derivações unipolares.

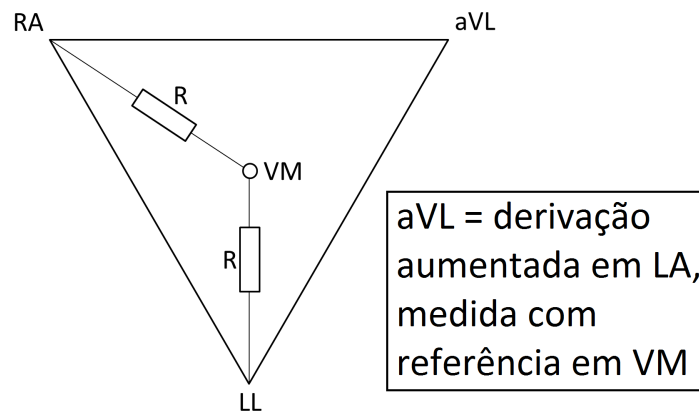


Figura 2.7: Exemplo de derivação aumentada de Goldberger.

As derivações aumentadas foram introduzidas por Emanuel Goldberger (1913-1994), cardiologista americano, com o intuito de ampliar os sinais obtidos por Wilson. Goldberger o fez desconectando o eletrodo a ser medido do WCT, e utilizando a junção dos outros dois como referência, como pode ser visto na Figura 2.7.

### 2.2.3 Derivações Precordiais

As derivações precordiais são obtidas com 6 eletrodos posicionados ao redor do coração, na superfície torácica, do 4<sup>o</sup> ao 5<sup>o</sup> espaços intercostais (entre costelas – Figura 2.8). Por estarem muito próximas ao coração, não requerem amplificação e utilizam como referência o WCT.

Em animais de pequeno porte, o mais comum é um sistema que utiliza apenas 4 derivações precordiais (Figura 2.9), mas existem médicos veterinários que defendem

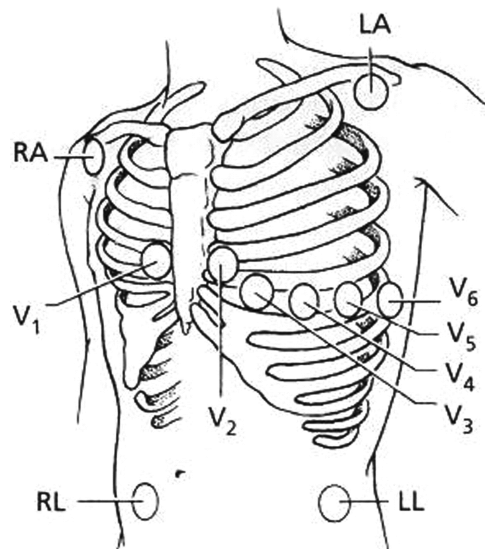


Figura 2.8: Posicionamento dos eletrodos precordiais e bipolares.

o uso de 6 derivações também para esses casos [8]. O sistema de 4 derivações em animais utiliza 4 eletrodos, V1(ou rV2), V2 e V4 localizados no 5<sup>o</sup> e 6<sup>o</sup> espaços intercostais, e V10 na região da coluna, sobre o 7<sup>o</sup> processo espinhoso dorsal [3]. Essas derivações podem ser comparadas às precordiais em humanos da seguinte maneira [9]:

- rV2 = V1.
- V2 = V2, V3.
- V4 = V4, V5.
- V10 = V6.

O sistema de 6 derivações precordiais em animais é uma adaptação do sistema utilizado em seres humanos. Os eletrodos são posicionados no 5<sup>o</sup> (V1) e 6<sup>o</sup> (V2 a V6) espaços intercostais, como descrito em [10], e pode ser visto na Figura 2.10.

As derivações unipolares (precordiais e aumentadas), assim como as bipolares, proporcionam vistas de diferentes partes do coração. Assim como dII e dIII, aVF é uma vista da parte inferior do coração. V1 e V2 são relacionadas à parte anterior, mais precisamente ao septo intraventricular. V3 e V4 são vistas da parede anterior dos ventrículos, enquanto V5 e V6 em conjunto com aVL e dI formam um grupo de derivações relacionado à parte lateral esquerda do coração [6, 7].

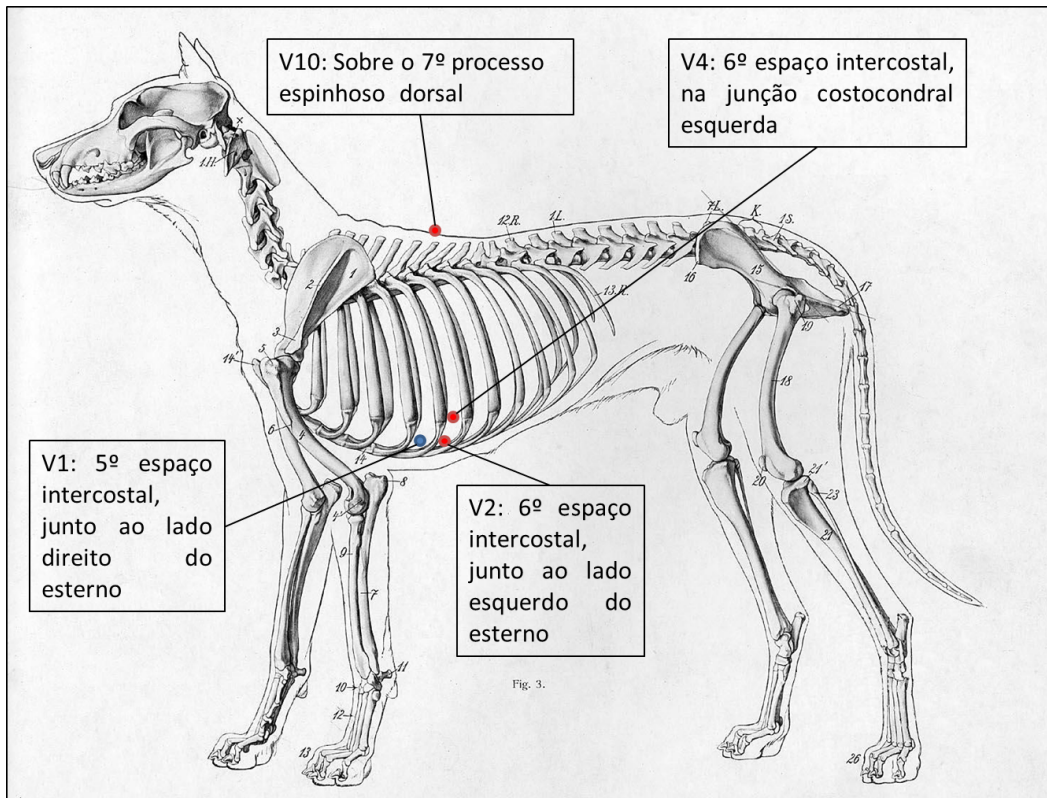


Figura 2.9: Posicionamento dos eletrodos precordiais com o sistema de 4 derivações.

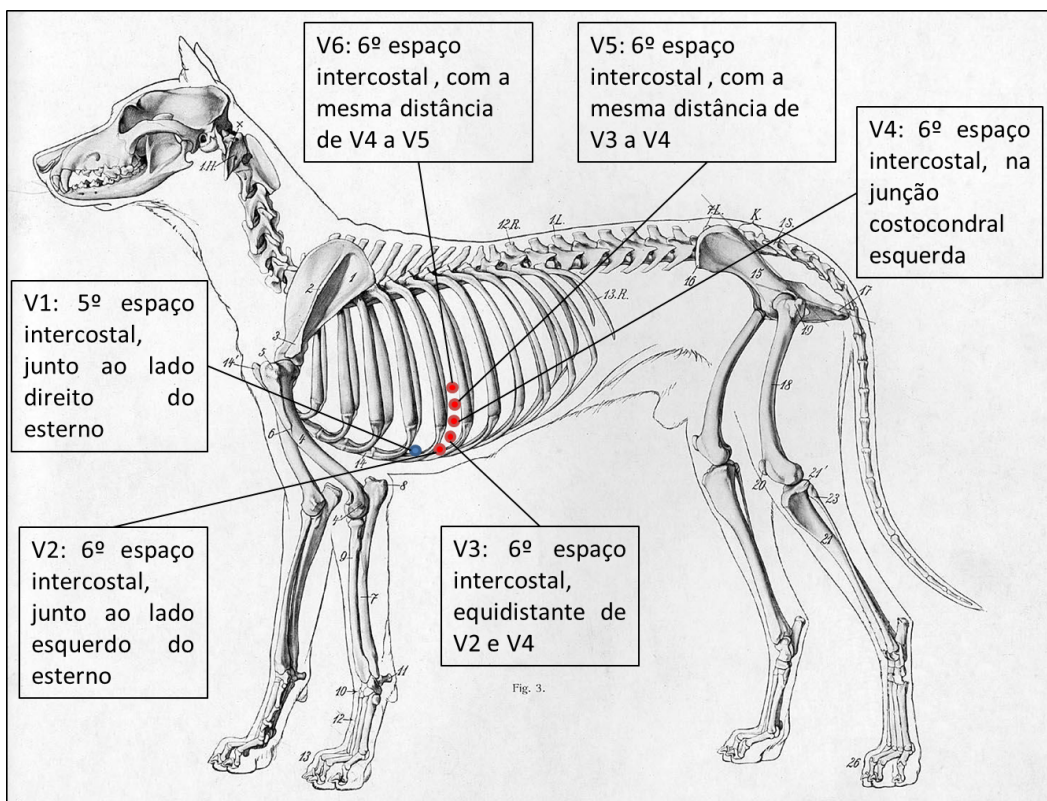


Figura 2.10: Posicionamento dos eletrodos precordiais com o sistema de 6 derivações.

## 2.3 Interfaces de Comunicação

### 2.3.1 Interface UART

UART (*Universal Asynchronous Receiver/Transmitter*, da sigla em inglês) é uma interface de comunicação serial assíncrona, que implementa uma comunicação *full-duplex* – o fluxo de dados é totalmente bi-direcional, ou seja, um dispositivo pode, ao mesmo tempo, enviar e receber dados. Duas conexões são utilizadas para isso: cada dispositivo tem uma porta de envio e uma de recebimento de dados. Como não há uma linha de clock, os dispositivos envolvidos na comunicação precisam utilizar a mesma velocidade de transmissão (*baud rate*). O protocolo utilizado pode variar, mas em geral utiliza-se mais comumente o padrão RS-232.

A implementação pode ser feita com 3 ou 5 fios, a depender das necessidades do sistema. O mais comum é utilizar um fio para transmissão (TX), um para a recepção (RX) e um para a referência/terra (GND). Quando existe a necessidade de maior controle e confiabilidade sobre a troca de dados adiciona-se o controle de fluxo por *hardware*: o *clear to send* (CTS) e *ready to send* (RTS) do protocolo RS-232. A ligação entre dispositivos tem os fios cruzados – TX de um dispositivo se conecta ao RX do outro e vice-versa, o mesmo acontece com o CTS e RTS, como visto na Figura 2.11.

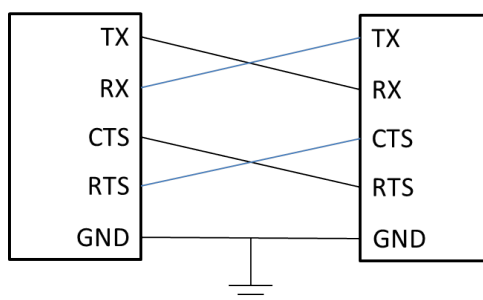


Figura 2.11: Esquemático da conexão das linhas utilizadas na comunicação serial.

A transmissão de dados se dá de forma serial, ou seja, um bit de cada vez. Os dados são enviados em frames compostos de um start bit, um ou dois stop bits, um bit de paridade opcional e 5 a 9 bits de dados. O start bit é sempre em lógica baixa, e o stop bit alta – dessa forma há sempre uma transição alto-baixo no começo de um novo frame [11]. A Figura 2.12 mostra um frame típico – nota-se que quando não há comunicação o estado deve permanecer alto. Como o clock é interno ao dispositivo, a sincronização para que a leitura dos bits seja feita corretamente se dá na detecção da descida do start bit.

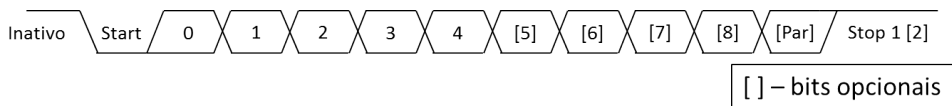


Figura 2.12: Formato de um frame, incluindo bits opcionais.

O controle de fluxo serve para que o emissor não envie mais dados do que o receptor consegue receber, evitando com isso erros de sobreescrita dos buffers de recepção e perda de dados. O protocolo CTS/RTS é uma forma de controle de fluxo que utiliza dois canais de comunicação (CTS e RTS) para fazer com que a comunicação possa ser pausada e continuada quando necessário. Cada dispositivo utiliza a linha de CTS para indicar ao outro dispositivo se está pronto para receber mais dados ou não. O CTS é ativo baixo – deve ser colocado em nível lógico baixo quando o dispositivo está apto a receber dados e alto quando não. A linha de RTS é utilizada para checar o estado do outro dispositivo e suspender a transmissão assim que verificar a linha mudar para nível lógico alto – terminando o envio do byte atual.

### 2.3.2 Interface SPI

A *Serial Peripheral Interface*, ou SPI, é uma interface de comunicação serial síncrona, e pode funcionar em barramento – com vários periféricos conectados à um mesmo dispositivo. Assim como a UART, opera em modo *full-duplex*. A comunicação é do tipo mestre-escravo, porém cada barramento pode ter apenas um mestre [12].

A comunicação SPI utiliza 4 componentes:

- Serial Clock (SCLK) é a linha de clock, controlada pelo dispositivo mestre.
- Master Output Slave Input (MOSI) é a saída de dados do dispositivo mestre.
- Master Input Slave Output (MISO) é a saída de dados do dispositivo escravo.
- Slave Select ( $\overline{SS}$ ) é a linha utilizada para selecionar qual escravo fica ativo, ou seja, com qual escravo o mestre está se comunicando. É ativo-baixa.

A transmissão de dados no protocolo SPI inicia com a seleção do escravo pelo mestre, colocando a linha  $\overline{SS}$  do escravo em lógico baixo. O mestre então ativa o SCLK, e a cada pulso o bit mais significativo do registrador de comunicação do mestre é transferido para o escravo, que o recebe na posição do bit menos significativo do seu registrador, ao mesmo tempo em que seu bit mais significativo é transferido para o registrador do mestre, também na posição menos significativa, como visto na Figura

2.13. Nessa operação todos os bits são deslocados, de forma que o segundo bit mais significativo se torna o primeiro, e assim por diante. Ao final da transmissão, mestre e escravo acabam por trocar os valores de seus registradores.

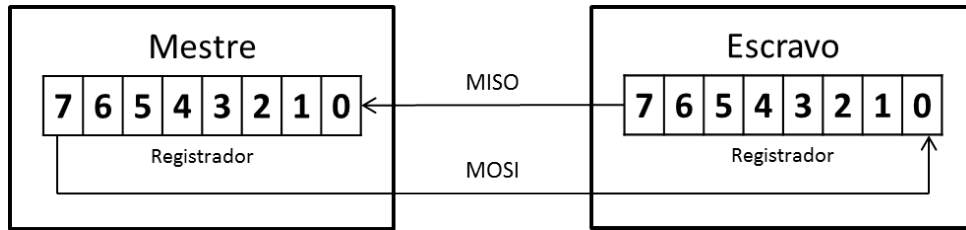


Figura 2.13: Deslocamento dos bits, em uma comunicação SPI.

Quando existem 2 ou mais escravos, existem duas possibilidades:  $\overline{SS}$  independentes para cada escravo, e compartilhamento das linhas de MOSI e MISO – a maioria dos dispositivos escravos tem saídas que se comportam como desconectadas quando o escravo não está selecionado. O dispositivo mestre seleciona um escravo por vez para realizar a comunicação (Figura 2.14).

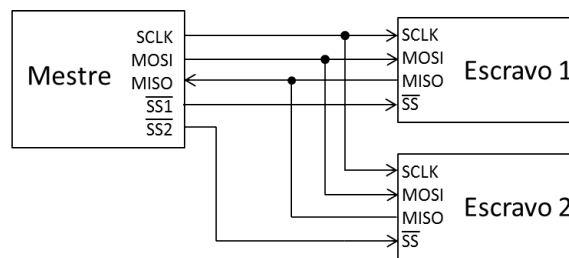


Figura 2.14: Conexão com  $\overline{SS}$  individuais para cada escravo.

Outra alternativa é conectar os dispositivos na chamada *Daisy-chain*. A saída do primeiro escravo é conectada à entrada do segundo, e assim por diante, até que a saída do último escravo seja ligada à entrada do mestre. Apenas uma linha de  $\overline{SS}$  é utilizada.

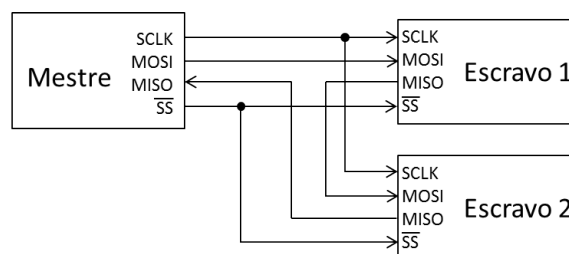


Figura 2.15: Conexão *Daisy-chain*.

Quando se inicia a transmissão, os dados de um escravo passam para o próximo, e os dispositivos devem ser capazes de, em um próximo grupo de clocks, repassar

esses dados até que os dados do primeiro escravo cheguem ao mestre. Na Figura 2.15, por exemplo, seriam necessários 16 pulsos de clock para que a informação de ambos os escravos chegasse ao mestre. Nos 8 primeiros pulsos o mestre receberia a informação do escravo 2, e nos 8 seguintes do escravo 1.

A comunicação SPI pode ser configurada em 4 modos diferentes, a depender de dois fatores: a polaridade do clock (CPOL) e a fase (CPHA). CPOL define o estado inativo do clock – alto (1) ou baixo (0). CPHA define se os dados serão amostrados e propagados na queda e subida ou subida e queda do clock, dependendo também de CPOL. Dessa forma, os 4 modos de operação ocorrem conforme a Tabela 2.1.

Modo	CPOL	CPHA	Amostra	Propagação
0	0	0	Descida	Subida
1	0	1	Subida	Descida
2	1	0	Subida	Descida
3	1	1	Descida	Subida

Tabela 2.1: Modos de funcionamento do SPI

### 2.3.3 Interface USB

O protocolo USB (*Universal Serial Bus*) é, provavelmente, o padrão mais comum de comunicação entre computadores e periféricos utilizado atualmente. Ele define os cabos e conectores a serem utilizados na conexão entre os dispositivos, bem como a forma de alimentação e o protocolo de comunicação.

Seu principal objetivo quando começou a ser desenvolvido, era obter uma solução única para a conexão de periféricos, visto que haviam diversos tipos de conectores e protocolos, como as portas seriais e paralelas, por exemplo. Outro objetivo era obter dispositivos *Plug-and-Play* (conecte e use) e *hot-pluggable*, que bastariam ser conectados para começar a funcionar, sem haver a necessidade de o usuário ter conhecimento técnico – os drivers seriam instalados e selecionados automaticamente, para cada tipo de dispositivo.

O USB funciona como um barramento “Mestre-escravo”, sendo o mestre no caso conhecido como *host*. Pode haver apenas um *host* no barramento, e a troca de dados é sempre iniciada por ele, assim como o processo de enumeração, que é o *handshake* inicial. O protocolo utiliza um endereçamento de 7 bits, o que possibilita acoplar até 127 periféricos à um único *host* (o endereço 0 é reservado) [13]. Para isso podem ser utilizados *hubs*, porém há um limite máximo de 6 “camadas” de conexões que é suportado (Figura 2.16).



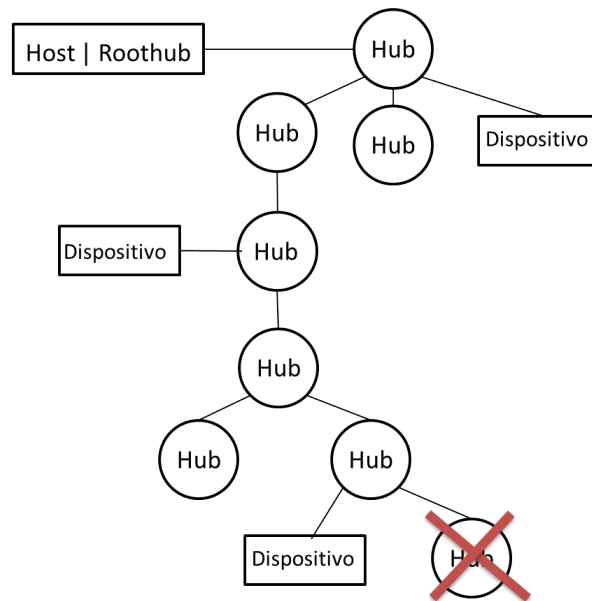


Figura 2.16: Esquema de conexão de hubs à uma porta USB, destacando-se o máximo de 6 camadas.

As conexões são realizadas com 4 fios: VBUS (5V), GND (terra), D+ e D-. VBUS pode ser utilizado para alimentar os dispositivos periféricos, enquanto D+ e D- são um par cruzado que leva os dados, referenciados em GND. Os conectores definidos na especificação do padrão são construídos de forma a evitar conexões erradas – conectores fêmeas do tipo A saem de um *host* ou *hub* e conectores fêmeas do tipo B são a “entrada” de um dispositivo ou *hub* (Figura 2.17). Isso impede que se conecte dois dispositivos *host* ou dois dispositivos “escravos” entre si.

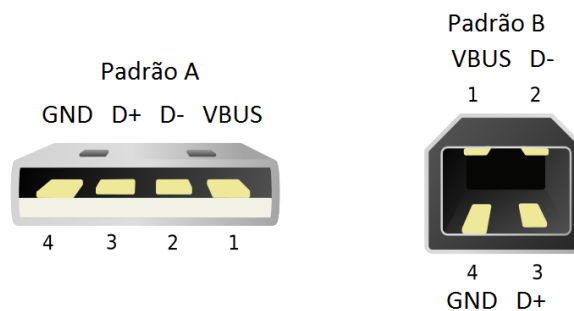


Figura 2.17: Conectores USB mais comuns.

O USB trabalha com 3 velocidades: *Low-Speed*, *Full-Speed* e *High-Speed*. *Low-Speed* e *Full-Speed* vêm do protocolo USB 1.1, e são de 1,5Mbits/s e 12Mbits/s respectivamente. *High-Speed* foi instituído pelo protocolo USB 2.0, e tem velocidade de 480Mbits/s. Já existe o protocolo USB 3.0, que traz o modo *Super-Speed*, com velocidade de 5 Gbits/s, mas que ainda não está amplamente difundido e tem apenas

começado a aparecer em novos equipamentos comercializados.

Para definir a velocidade de transmissão dos periféricos, deve-se utilizar resistores *pull-up* nas linhas D+ e D-. Para *Low-Speed* utiliza-se um resistor na linha D-, e para *Full-Speed* na linha D+. Dispositivos *High-Speed* se conectam inicialmente como *Full-Speed*, portanto o resistor também vai na linha D+. O *host* detecta qual linha está alta e determina qual a velocidade a ser utilizada.

Outra característica do protocolo USB são os modos de transmissão. São 4 ao total:

**Control:** Em geral utilizado para transmissão de comandos e status do *host* para o periférico e vice e versa. É o modo utilizado no processo de enumeração, que configura a comunicação entre os dispositivos.

**Interrupt:** É um tipo de transmissão com garantia de latência, ou seja, o *host* envia à intervalos regulares uma mensagem para verificar se o periférico tem mensagens na fila de espera para enviar. Esse intervalo tem um tempo máximo definido, de forma que o *polling* feito pelo *host* seja feito em frequência igual ou maior à determinada na configuração da comunicação (enumeração).

**Isochronous:** Esse tipo de transmissão ocorre de forma periódica e contínua. É ideal para aplicações que dependem do tempo, como transmissões de áudio ou vídeo em que é importante que haja um fluxo contínuo de dados, mas a perda de alguns dados não é tão crítica. Para isso, não há confirmação de entrega nem reenvio de pacotes, caso algum dado seja corrompido, o pacote é perdido.

**Bulk:** É um modo utilizado para transferir grandes pacotes de dados. Garante-se a entrega dos dados, mas não há garantia do uso da banda nem de latência. Pode ser utilizado em aplicações em que é necessário transferir-se grandes quantidades de dados corretos rapidamente e não há problemas muito grandes se os dados forem recebidos com atraso.

### 2.3.4 Interface I<sup>2</sup>C

O *Inter-Integrated Circuit*, ou I<sup>2</sup>C, é um padrão de comunicação serial síncrono. A comunicação é feita no modo mestre-escravo, com o mestre iniciando a comunicação. Como o I<sup>2</sup>C pode funcionar como um barramento, os escravos são selecionados por endereçamento [12, 13].

O I<sup>2</sup>C é composto por 2 linhas, uma de dados (SDA) e uma de clock (SCL). Ambas funcionam com resistores *pull up* (Figura 2.18). A linha de clock é comandada pelo mestre, e serve para sincronizar a troca de dados. Para iniciar e finalizar as transmissões, são utilizadas “sequências” de *start* e *stop*, bem como para confirmar o recebimento de dados (ACK).

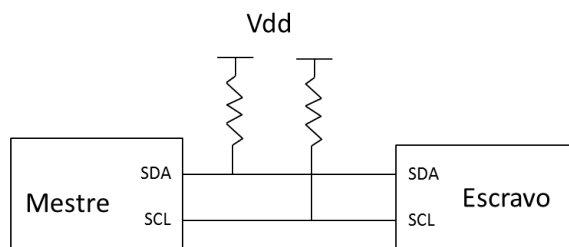


Figura 2.18: Conexão I<sup>2</sup>C.

A comunicação se inicia com o mestre enviando uma sequência de *start*. Isso fará com que os escravos saibam que uma transmissão irá acontecer, e faz com que todos passem a “ouvir” o barramento, para verificar o destinatário da mensagem. O mestre então envia o byte correspondente ao endereço do escravo com que deseja comunicar-se. O escravo deve responder com um ACK, e os outros escravos passam a ignorar o barramento. O endereço contém 7 bits, que são posicionados nos bits mais significativos do byte enviado. O último bit é reservado para definir se a operação a ser realizada é de escrita ou leitura. O mestre então procede com a transmissão dos próximos bytes, até enviar uma sequência de *stop* para finalizar a comunicação. A rotina de *repeated start* também finaliza uma comunicação, mas não libera o barramento pois inicia uma nova conexão logo em seguida.

Quando o mestre deseja ler algum dado do escravo é possível que, após o recebimento de um comando de leitura, seja necessário determinado tempo para que o escravo coloque esses dados em seu buffer de saída. Como toda a transmissão é comandada pelo mestre, e não há como saber se o escravo tem os dados prontos para serem lidos, o protocolo I<sup>2</sup>C permite que o escravo segure a linha de clock baixa. Isso evita que o mestre inicie o processo de leitura enquanto os dados não estão prontos para serem lidos. Quando o escravo libera a linha, os *pull ups* a puxam para cima e o mestre sabe que pode iniciar o processo de leitura. Esse comportamento se chama *clock stretching*.



# Capítulo 3: Análise de Requisitos para os Módulos Funcionais

O projeto do eletrocardiógrafo pode ser dividido em 3 partes principais: *hardware*, *firmware* e *software*. Esses 3 módulos focam em partes diferentes do projeto, e podem ser desenvolvidos, de certa forma, paralelamente. O *hardware* concentra-se na parte física, como por exemplo a seleção dos componentes, o projeto da placa do circuito, a fabricação do dispositivo em si – *case*, eletrodos, cabos. Já o *software* é a camada que faz a interface entre usuário e aparelho, sendo suas funções transmitir comandos do usuário ao dispositivo, processar os dados recebidos e fornecê-los de forma correta e "que façam sentido". O *firmware* lida com a programação e integração dos componentes utilizados, buscando obter o comportamento que se deseja do dispositivo. Funciona como uma interface entre *hardware* e *software*.

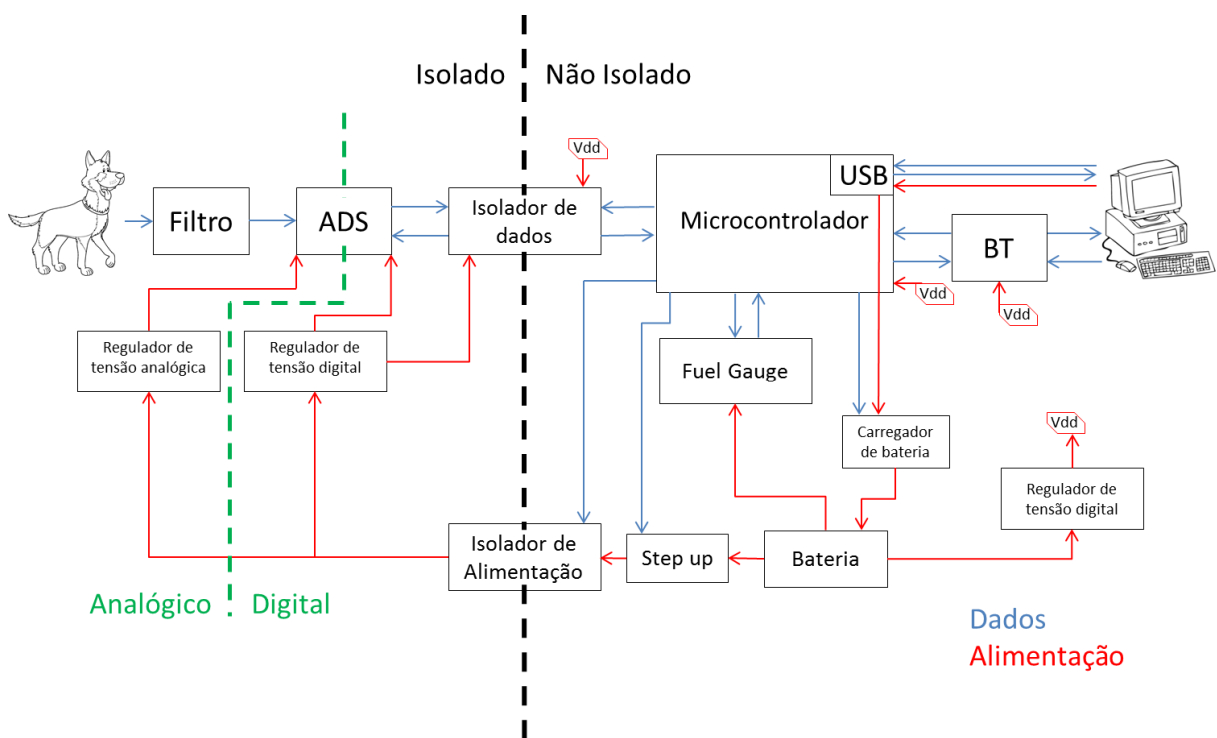


Figura 3.1: Diagrama com os principais módulos a serem implementados no dispositivo.

Em uma das primeiras etapas do projeto, foram definidos os principais módulos necessários ao funcionamento do aparelho, de forma a satisfazer os requisitos básicos. Esses módulos básicos podem ser vistos na Figura 3.1. Destacam-se também os módulos referentes à alimentação dos componentes, ressaltando a necessidade de

diferentes alimentações para partes isoladas/não isoladas e digitais/analógicas.

Após a seleção dos componentes essenciais, *hardware*, *firmware* e *software* se desenvolveram paralelamente – tanto quanto possível. Este trabalho foca na parte do *firmware*, entrando brevemente na parte de *hardware*, apenas no que diz respeito a conhecer os componentes utilizados e *software* para a elaboração do protocolo de comunicação utilizado.

## 3.1 Principais Módulos Utilizados

### 3.1.1 Microcontrolador: PIC18F47J53

O microcontrolador escolhido foi o PIC18F47J53, de 8 bits, produzido pela *Microchip Technology Inc.*. Sua escolha se baseou principalmente em 3 aspectos. Primeiramente, por ter uma tecnologia de baixo consumo, importante para um dispositivo à bateria. Também levou-se em conta seu tamanho reduzido, tendo em vista o objetivo da portabilidade do aparelho. Por fim, a presença de um módulo USB – nem todos os microcontroladores possuem essa funcionalidade – uma das características desejadas.

Outro aspecto do microcontrolador interessante ao projeto é a presença de vários módulos de comunicação adicionais ao USB. São 2 módulos que podem ser utilizados para comunicação SPI ou I<sup>2</sup>C e 2 módulos USART. Além disso, o PIC18F47J53 possui modos de descanso com alta economia de energia, para momentos em que o dispositivo não esteja em funcionamento. As principais características do PIC18F47J53 são apresentadas na Tabela 3.1.

Característica	PIC18F47J53
Número de pinos	44
Memória de Programa (Kbytes)	128
Memória de Dados (Kbytes)	3,8
Comunicação Serial	USART, SPI e I <sup>2</sup> C
Frequência de Operação	31kHz ~ 48MHz
Tensão de Alimentação	2,15 ~ 3,6V
Consumo de Corrente	15nA ( <i>Deep Sleep</i> ) ~ 20,5mA (48MHz)

Tabela 3.1: Principais características do PIC18F47J53 [13]

### 3.1.2 Front-end: ADS1298

O ADS1298, produzido pela *Texas Instruments Inc.*, é um *front-end* para aquisição de sinais biomédicos. Internamente ele possui um conversor analógico digital

delta-sigma de 8 canais simultâneos e 24 bits de resolução. Possui amplificadores de ganho programável, referência e oscilador internos. Suas características elétricas se encontram na Tabela 3.2. Ele traz todas as características normalmente requisitadas em dispositivos biomédicos, como aparelhos para eletrocardiografia, eletroencefalografia e eletromiografia [14].

Característica	ADS1298
Tensão de Alimentação (Análogica)	2,7 ~ 5,25V
Tensão de Alimentação (Digital)	1,65 ~ 3,6V
Consumo de Corrente (RLD e WCT desligados)	3,1mA

Tabela 3.2: Características elétricas do ADS1298

Especificamente para eletrocardiógrafos, o circuito integrado possui já implementados o circuito alimentador de perna direita (*Right Leg Drive* - RLD), terminal central de Wilson (WCT) e terminal central de Goldberger (GCT). O RLD é ligado ao eletrodo de perna direita, e injeta no corpo um sinal com o intuito de eliminar ruídos de modo comum. O ADS1298 permite que se configure o RLD para qualquer combinação de eletrodos que se queira, sendo a mais comum a combinação dos eletrodos dos membros (braços e pernas).

O WCT e GCT são essenciais para um ECG de 12 derivações, pois são utilizados como referência nas derivações unipolares e aumentadas, respectivamente. Dessa forma, utilizando o ADS1298 é possível obter sinais de ECG sem a necessidade de implementar extensos circuitos de aquisição. Para ativá-los basta alterar o valor de alguns registradores.

Para se obter as 12 derivações de um conversor de 8 canais, faz-se a aquisição de 2 derivações bipolares (DI e DII) e 6 precordiais. A partir de DI e DII, pode-se calcular DIII e as aumentadas (aVR, aVL e aVF).

O oscilador interno funciona a uma frequência de 2,048MHz, e a referência interna pode ser programada para 2,4 ou 4V. O ganho programável possui 7 configurações: 1, 2, 3, 4, 6, 8 ou 12, e a taxa de amostragem pode variar de 250SPS (*Samples per second*) para até 32kSPS.

O ADS1298 possui uma interface de comunicação SPI, pela qual se faz toda a comunicação com o CI. Funciona como um escravo para o barramento SPI, e é através dessa interface que a comunicação PIC-ADS é realizada. O PIC pode, através de determinados comandos, acessar os registradores internos do ADS e realizar operações de escrita ou leitura, configurando o dispositivo da forma desejada, por exemplo: configurações de ganho e taxa de amostragem como descrito acima. Também é possível enviar comandos operacionais, como iniciar ou pausar a aquisição

ou entrar em modo de economia de energia. E não menos importante, realizar a leitura dos dados convertidos.

Para sincronizar a leitura com a conversão dos dados, o ADS1298 conta com uma linha a mais do que as 4 utilizadas pela comunicação SPI. Trata-se do pino de *Data Ready*, que é colocado em lógica baixa (é um pino ativo-baixo) quando uma conversão foi terminada e os dados encontram-se no buffer de saída. Dessa forma, a leitura dos dados pelo PIC pode ser feita de forma assíncrona, baseada em uma interrupção. Cada conversão gera 216 bits, sendo 8 pacotes de 24 bits ou 3 bytes, um para cada canal e mais 3 bytes iniciais que funcionam como um cabeçalho. A transferência dos dados pelo protocolo SPI pode ser vista na Figura 3.2. Pode-se reparar que o pino de *Data Ready* retorna no primeiro pulso de clock. Além desse pino, o ADS ainda possui pinos de *Start* e *Reset*, que podem ou não ser utilizados pela aplicação.

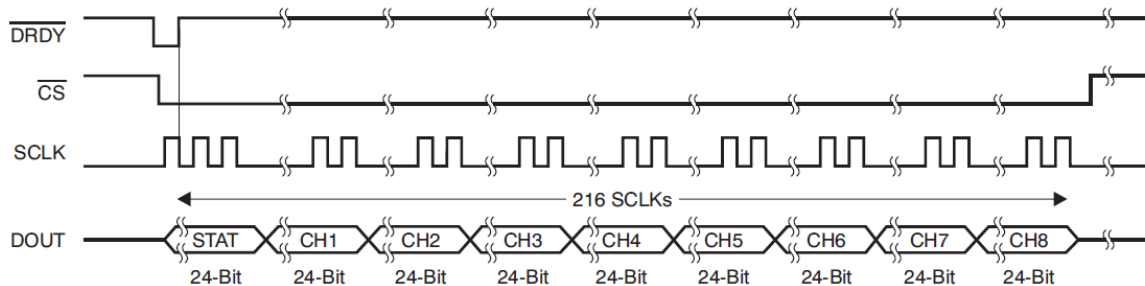


Figura 3.2: Operação de leitura de dados do ADS.

### 3.1.3 Módulo de Comunicação Bluetooth: WT11i-A

Uma das especificações do projeto é a comunicação *wireless*. Para tal, a tecnologia *Bluetooth* foi selecionada. O módulo WT11i-A, da *Bluegiga Technologies*, foi escolhido, por trazer diversas vantagens ao projeto. Por ser um dispositivo de classe 1, tem um alcance maior e transmite o sinal com maior potência [15]. Dessa forma, diminuem-se os problemas com interferência de objetos no caminho e até mesmo do *case* na comunicação, já que o mesmo é confeccionado em alumínio. A Tabela 3.3 traz as características de consumo do módulo.

Característica	WT11i-A
Tensão de Alimentação	2,7 ~ 3,6V
Consumo de Corrente máximo	180mA
Consumo de Corrente médio	72mA
Consumo de Corrente ( <i>Deep Sleep</i> )	50uA

Tabela 3.3: Características de consumo do WT11i-A.



O WT11i-A traz integrado um *firmware* que implementa o protocolo *Bluetooth*. É o *iWRAP*, que utiliza sua interface serial UART, por onde é possível controlar o módulo (Figura 3.3). Esse controle pode ser feito pelo envio de comandos em ASCII, que são interpretados pelo processador interno do WT11i-A e permite que o usuário não precise implementar as camadas mais baixas e complexas do protocolo.

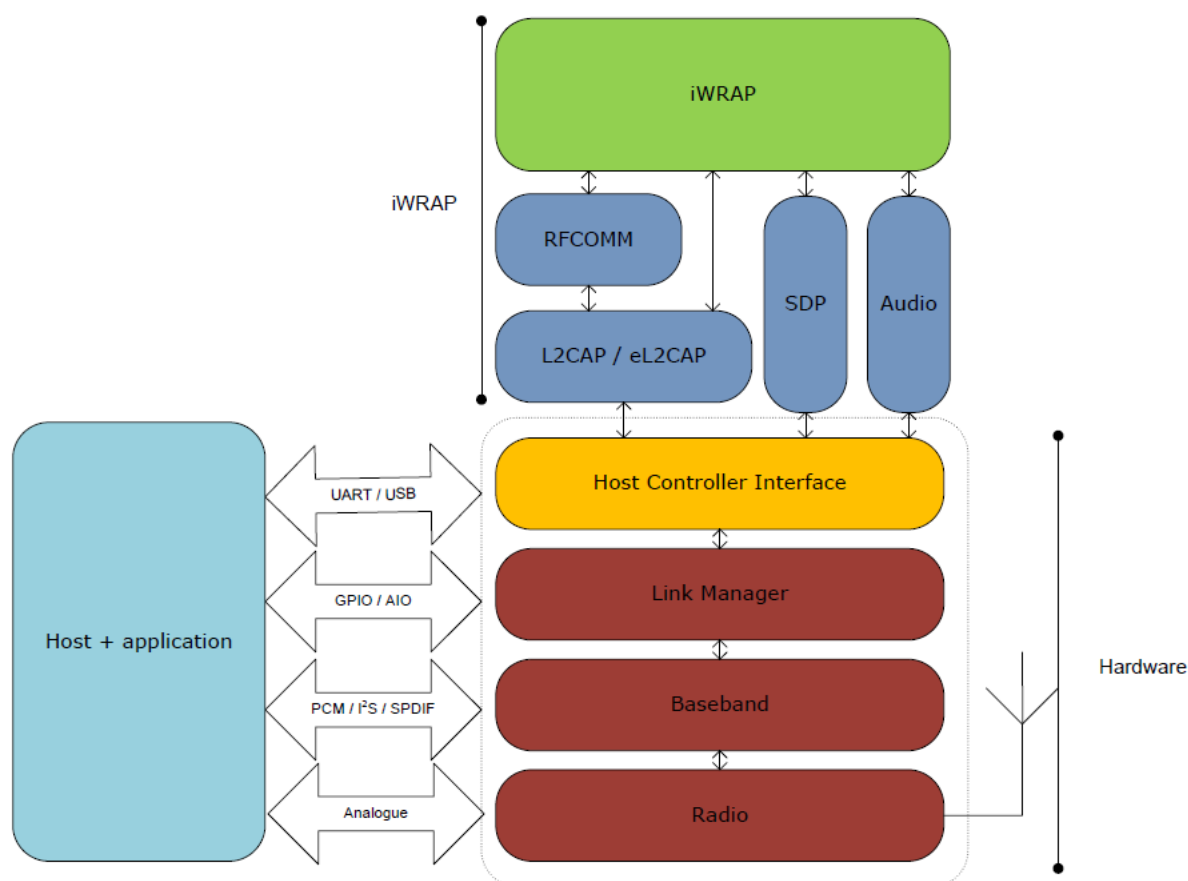


Figura 3.3: Protocolo *Bluetooth* implementado pelo *iWRAP* (imagem retirada do guia de usuário do *iWRAP*) [16].

O *iWRAP* possui 2 modos, o modo de comando e o modo de dados. Quando se inicia uma conexão, entra-se no modo de dados. Existem comandos para migrar entre os dois modos. No modo comando, é possível configurar diversos parâmetros da comunicação, como *baud rate*, nome do dispositivo, etc. O *iWRAP* traz 13 possíveis perfis de comunicação do protocolo *Bluetooth*, incluindo o utilizado no projeto – RFCOMM – que simula uma porta serial.

### 3.1.4 Fuel Gauge: bq27541-G1

Como o ECG será alimentado por uma bateria, é importante que se tenha um *feedback* sobre o estado de carga da bateria, para evitar possíveis problemas durante o

uso. O *fuel gauge* é um dispositivo que monitora o estado da bateria, ao medir variáveis como tensão, corrente e temperatura. O bq27541-G1, da *Texas Instruments Inc.*, é um *fuel gauge* para baterias de íons de lítio (Li-Ion) de uma célula. Esse dispositivo fornece informações como capacidade restante (mAh), estado de carga (%), tempo restante, tensão e temperatura [17].

Sua escolha se deu justamente por oferecer uma grande quantidade de informações sobre a bateria, sem a necessidade de cálculos posteriores pois possui um algoritmo interno para determinação das variáveis, característica não presente em todos os CIs do tipo. Também levou-se em conta o consumo e a interface de comunicação. A Tabela 3.4 mostra as características de consumo do CI.

Característica	BQ27541-G1
Tensão de Alimentação	2,7 ~ 5,5V
Consumo de Corrente (Operação Normal)	131uA
Consumo de Corrente ( <i>Sleep</i> )	60uA
Consumo de Corrente ( <i>Full Sleep</i> )	21uA
Consumo de Corrente ( <i>Hibernate</i> )	6uA

Tabela 3.4: Características de consumo do BQ27541-G1.

O bq27541-G1 utiliza uma interface I<sup>2</sup>C para comunicação – permitindo configurar o dispositivo e fazer a leitura de dados medidos através de comandos pré-definidos. Assim como no caso do ADS1298, há comandos de controle que possibilitam alterar parâmetros do algoritmo de monitoramento: é possível alterar variáveis como tensão de carga total, tensão limiar de uso da bateria, e também se pode realizar a calibração do dispositivo.

# Capítulo 4: Implementação

Este capítulo tem por objetivo descrever o desenvolvimento da solução para o dispositivo ECG, e está dividido em 3 partes: desenvolvimento dos módulos, fluxo do *firmware* e protocolo de comunicação.

## 4.1 Fluxo do *Firmware*

O ECG deve funcionar de forma a interagir com um *software* e responder corretamente à estímulos do usuário, via *software* ou botão. Esse funcionamento é buscado no desenvolvimento do *firmware* que é embarcado no aparelho. O comportamento desejado para o dispositivo pode ser visto nos fluxogramas das Figuras 4.1, 4.2, 4.3, 4.4 e 4.5.

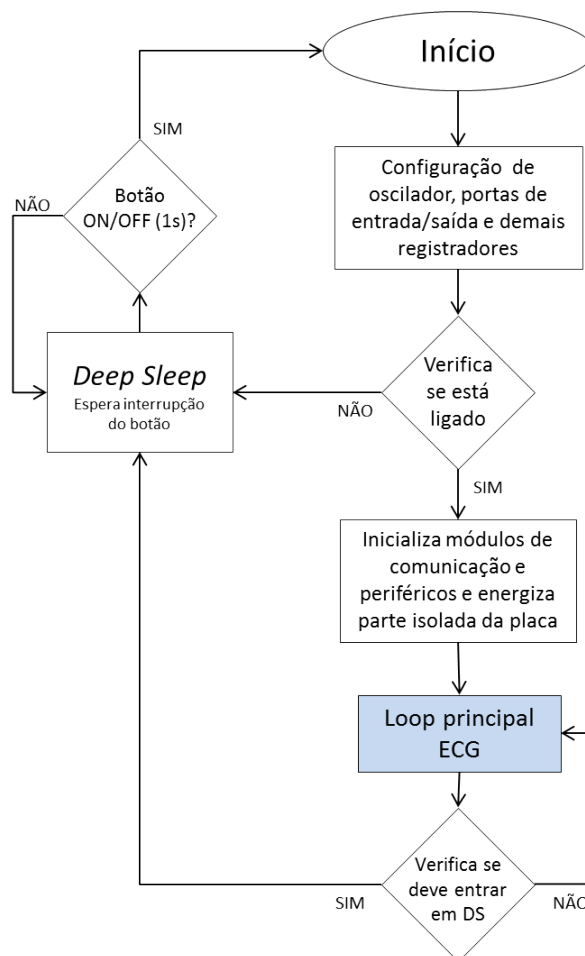


Figura 4.1: Fluxograma do comportamento desejado do ECG.

O aparelho inicia seu funcionamento configurando o PIC18F47J53, bem como as portas de comunicação com seu periféricos. Uma verificação é então feita sobre o atual estado do dispositivo – ligado ou desligado – que fica guardado em uma variável persistente. Quando o aparelho deveria estar desligado, o sistema simplesmente retorna ao *Deep Sleep*, estado de máxima economia de energia. Caso esteja ligado, faz-se a inicialização dos módulos de comunicação: USB, UART, SPI e I<sup>2</sup>C, e alimenta-se a parte isolada da placa – ativando o Isolador de Alimentação (Figura 3.1). A partir de então, o ECG entra em seu *loop* principal.

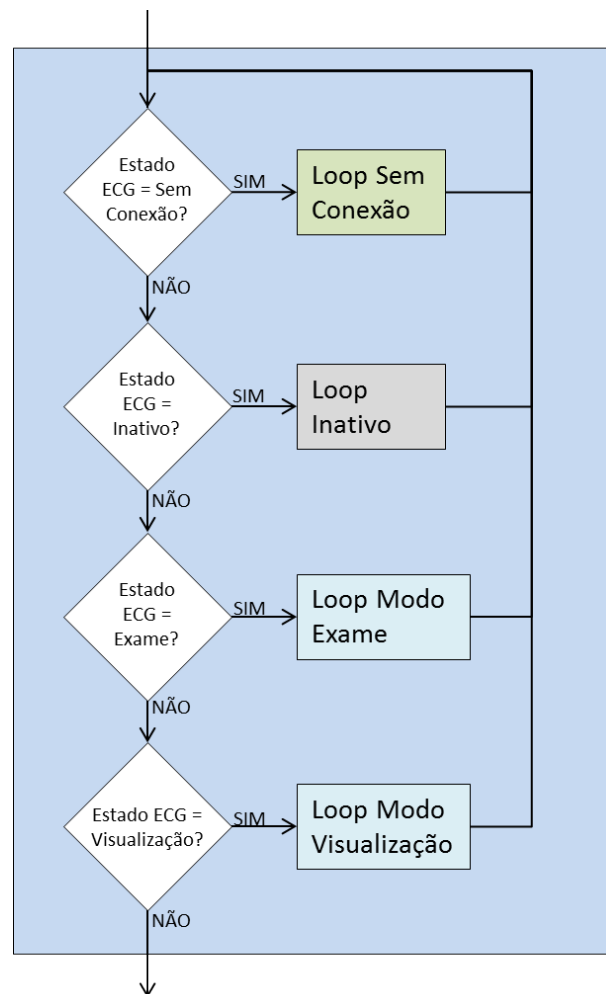


Figura 4.2: Detalhamento do fluxograma, referente ao loop principal do *firmware*.

O *loop* principal consiste em 4 *loops*, para 4 modos diferentes em que o dispositivo pode se encontrar. Há o estado sem conexão, em que o aparelho se encontra ligado, porém desconectado do computador/*software* – apenas aguardando uma conexão *Bluetooth* ou USB. Há então o estado *Idle*, em que já foi realizada uma conexão com o computador/*software*, mas o dispositivo não performa nenhuma função de conversão de dados. Nesse estado, ele já responde à comandos de configuração. Os modos

Exame e Visualização são os modos em que o dispositivo encontra-se pronto a realizar aquisições, em mais alta (Exame) ou baixa resolução (Visualização).

Quando não há conexão o dispositivo responde apenas à mensagens de pedido de conexão, descartando qualquer outro tipo de mensagem (Figura 4.3). O buffer de entrada recebe dados de qualquer um dos tipos de comunicação do aparelho com o *software*, que deve enviar uma mensagem de pedido de conexão, após realizar a conexão “física”.

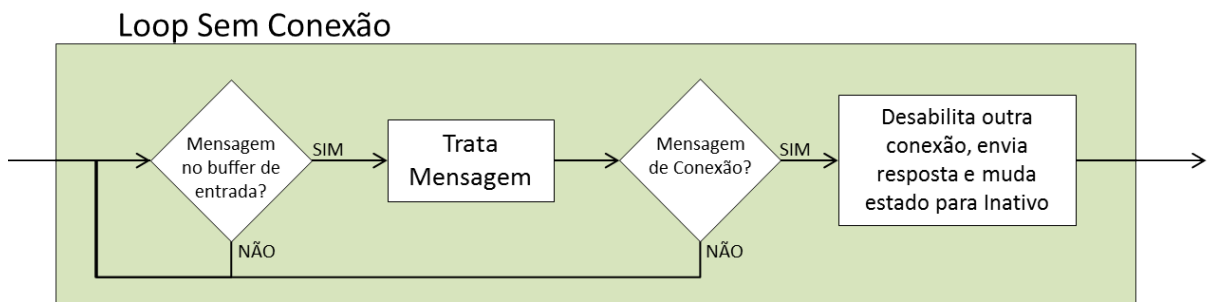


Figura 4.3: Comportamento do dispositivo quando sem conexão.

Já no modo *Idle*, o dispositivo responde à mensagens de controle, sem realizar aquisições (Figura 4.4), mudando de estado quando solicitado pelo usuário.

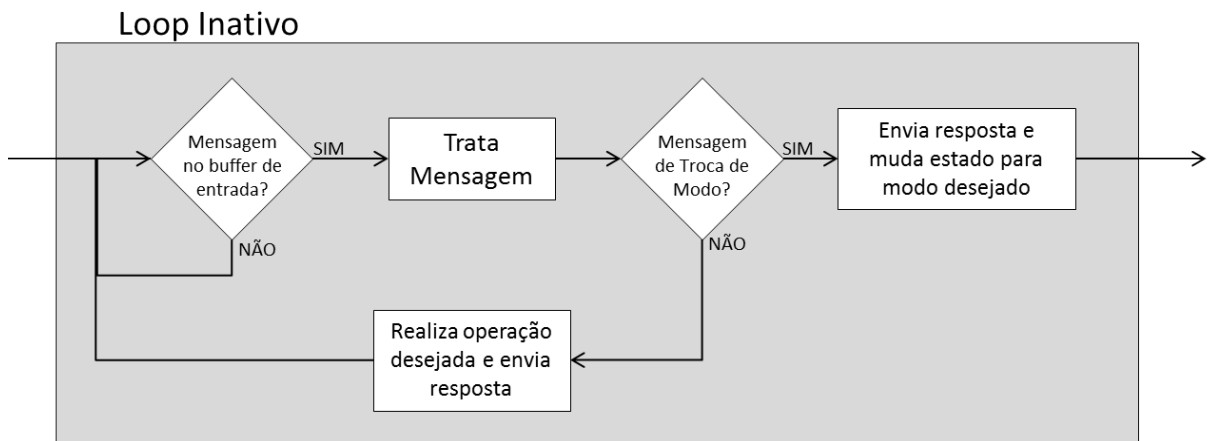


Figura 4.4: Comportamento do dispositivo quando em modo *Idle*.

Nos modos Exame e Visualização, o ECG verifica constantemente a situação de seus *buffers* de entrada e saída, de forma que processe tão rápido quanto possível as requisições do *software*, ao mesmo tempo em que não permita que o buffer de dados de saída seja completamente preenchido, o que levaria à perda de dados.

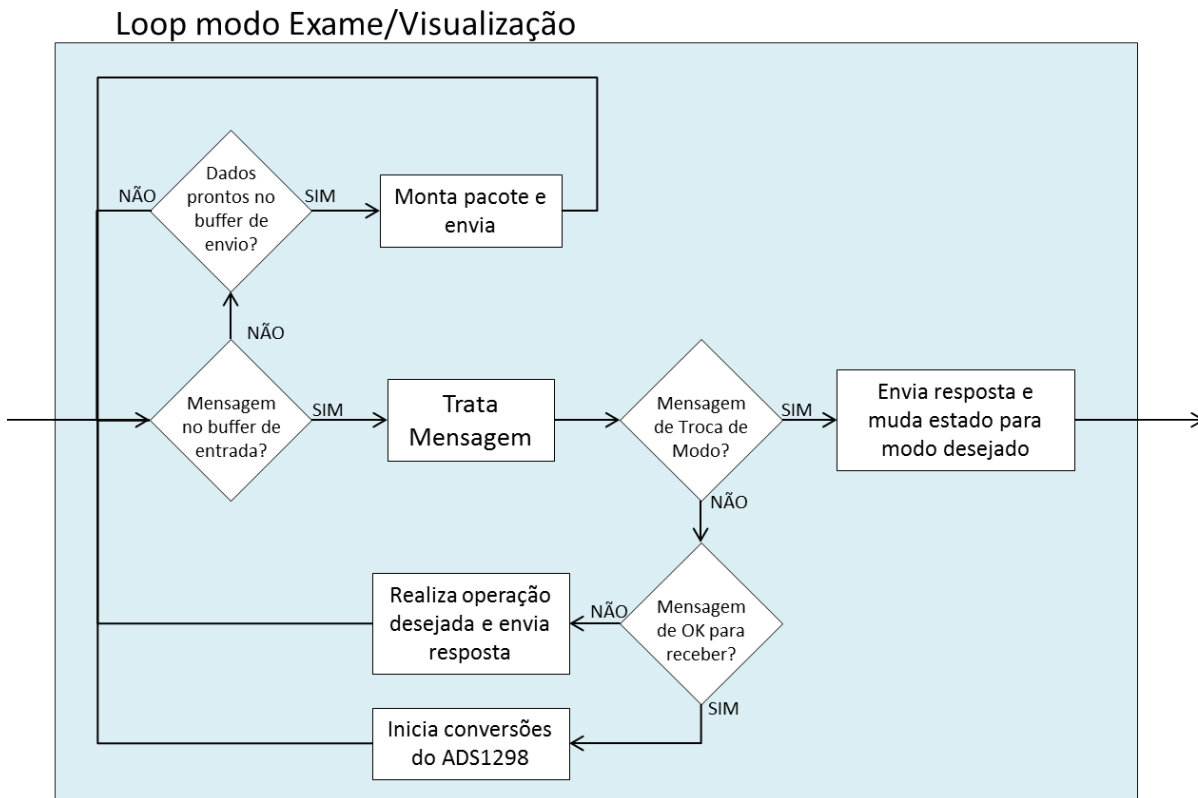


Figura 4.5: Comportamento do dispositivo em modo de Exame ou Visualização.

## 4.2 Desenvolvimento dos Módulos

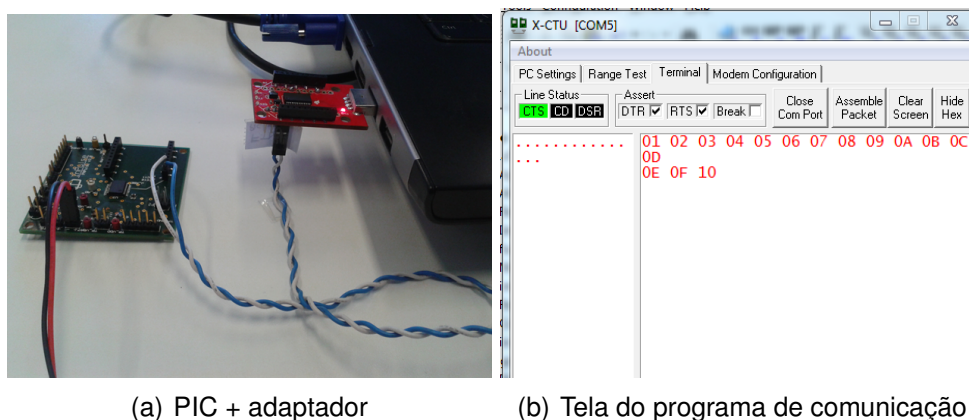
O dispositivo é controlado pelo PIC18F47J53. O microcontrolador deve coordenar as ações de cada um dos periféricos, buscando um comportamento que se encaixe no proposto. Porém, antes de desenvolver o *firmware* do ECG, o qual realiza todas as ações planejadas, foi necessário estudar cada um dos módulos utilizados, e desenvolver bibliotecas de comunicação para cada um deles.

Apesar de o projeto utilizar um PIC18F47J53, o desenvolvimento inicial se deu sobre o PIC 16LF1829, pela experiência dos supervisores e facilidade de acesso que havia à esse componente na empresa. Foi utilizada uma placa contendo o PIC16LF1829 e sistema de alimentação para realizar os primeiros testes. Posteriormente, o sistema foi migrado para o PIC18F47J53.

### 4.2.1 Interface UART e Módulo *Bluetooth*

A primeira biblioteca a ser desenvolvida foi a da comunicação serial UART – em conjunto com o módulo *Bluetooth*. Inicialmente foi necessária uma familiarização maior com a interface serial, portanto utilizou-se um adaptador USB-Serial para as primeiras

comunicações realizadas pelo PIC (Figura 4.6(a)).

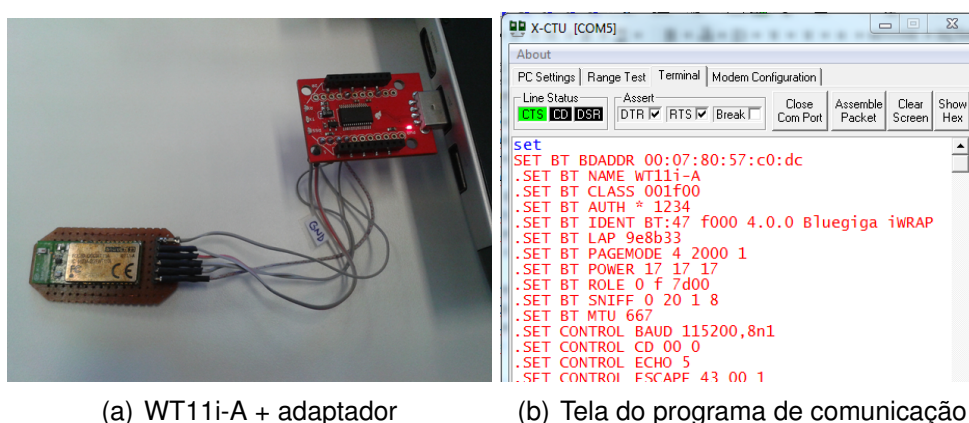


(a) PIC + adaptador

(b) Tela do programa de comunicação

Figura 4.6: Conexão da placa com o PIC e o computador, através do adaptador USB-Serial.

Através de um *software* do tipo *Hyperterminal* foi possível depurar os erros e acertos na programação do PIC, com o envio de dados conhecidos (Figura 4.6(b)). A partir daí foi criada uma biblioteca de funções para configurar e utilizar a comunicação UART. Para configurar o módulo *Bluetooth* a mesma abordagem foi utilizada. Para isso, o módulo bluetooth foi conectado diretamente à placa conversora USB-Serial para permitir a comunicação com o terminal (Figura 4.7). A biblioteca referente ao WT11i-A para o PIC foi então construída, utilizando a biblioteca UART e os comandos de controle do módulo.



(a) WT11i-A + adaptador

(b) Tela do programa de comunicação

Figura 4.7: Módulo *Bluetooth* conectado ao computador via adaptador USB-Serial.

Por fim, com as bases da comunicação PIC - WT11i-A criadas, foram realizados testes nos quais, de um lado estava presente o PIC integrado ao WT11i-A, e do outro lado um computador recebendo os dados de um módulo bluetooth WT11i-A conectado a um conversor USB. A ligação entre o PIC e o WT11i-A pode ser vista na Figura 4.8.

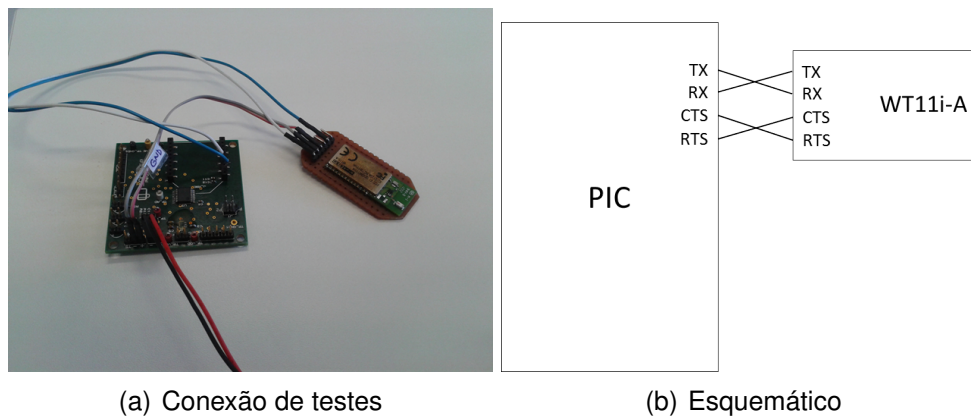


Figura 4.8: Conexão PIC e WT11i-A.

## 4.2.2 Interface SPI e ADS1298

Para o estudo do ADS1298, optou-se pela compra do kit de desenvolvimento fornecido pela fabricante, para que se pudesse entender seu funcionamento de forma mais clara. O kit acompanha um *software* de aquisição próprio para eletrocardiogramas. Com o kit pôde-se comprovar o funcionamento do componente, bem como de seus circuitos internos como o compensador de perna direita (RLD) e terminal central de Wilson (WCT). Na Figura 4.9 pode-se ver o kit conectado ao computador e a tela do *software* de aquisição da *Texas Instruments Inc.* O *software* possibilita obter todas as 12 derivações – e daí retirou-se uma primeira configuração do ADS1298 funcional – que posteriormente foi ajustada para o dispositivo construído.

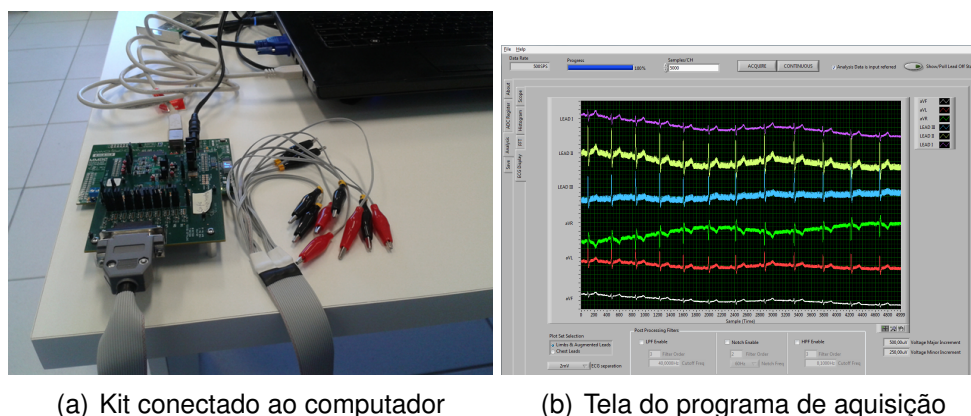


Figura 4.9: Kit de desenvolvimento do ADS1298.

Para desenvolver a comunicação entre PIC e ADS foi utilizada então a parte do kit referente ao ADS, e foram feitas as conexões necessárias com o PIC. Com a falta de um conversor USB-SPI para depurar o sucesso/insucesso na tentativa de configuração da interface SPI do PIC e posterior comunicação com o ADS, continuou-se utilizando a



interface serial, já funcional da etapa anterior do projeto. Na Figura 4.10 encontra-se a placa do PIC conectada à placa do ADS, e o conversor USB-Serial conectado ao computador. Criou-se então uma biblioteca de funções referentes à comunicação SPI e, com base nela, uma biblioteca para a comunicação com o ADS.

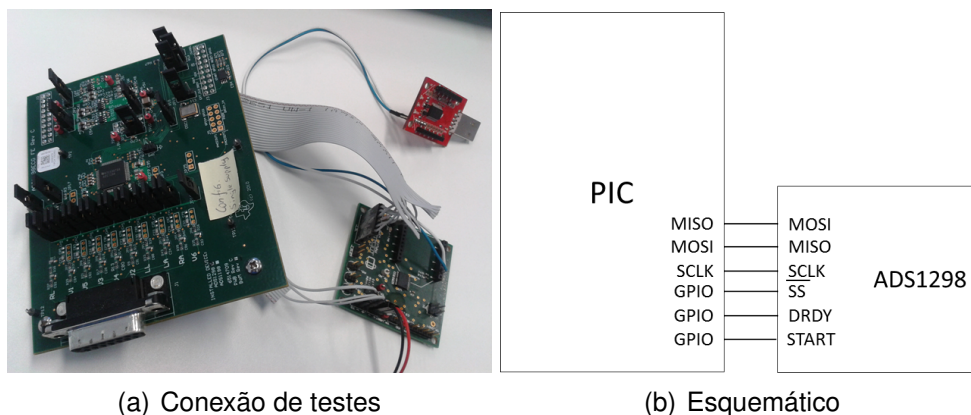


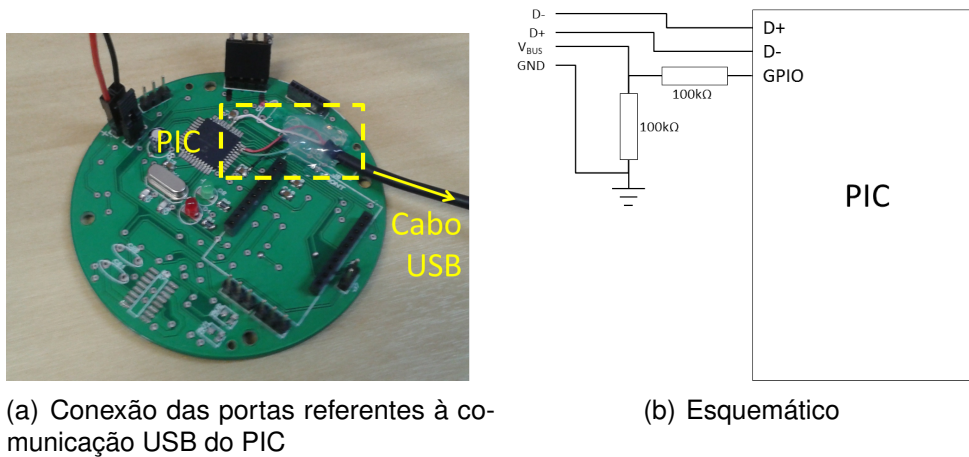
Figura 4.10: Conexão PIC e ADS1298.

### 4.2.3 Interface USB

O USB é uma interface completamente *user-friendly*. Um dos objetivos do USB-IF, *USB-Implementers Forum* – entidade criada para organizar e publicar as especificações do padrão, era obter uma solução única para os diversos padrões existentes de conexão de periféricos ao computador. Essa solução deveria ser tal que não requeresse nenhum conhecimento técnico do usuário para instalar um novo dispositivo. Dessa forma, a implementação do protocolo USB é extremamente complexa, e por isso optou-se por utilizar uma biblioteca fornecida pelo fabricante *Microchip Technologies Inc.*

O desenvolvimento da interface USB se deu diretamente no PIC18F47J53, visto que o PIC16F não possuía portas destinadas à essa comunicação. Utilizou-se uma outra placa, contendo o PIC18, para realizar os testes. Com as conexões corretas no PIC18F47J53 foi possível testar a interface USB e adaptar a biblioteca sem precisar estudar o protocolo USB à fundo. Na Figura 4.11 pode-se ver o cabo USB ligado ao PIC, e o esquemático referente à essa conexão. Nota-se que não há resistor *pull-up* em nenhuma das linhas de dados, como descrito na Seção 2.3.3, pois o PIC possui esse resistor internamente.

A biblioteca escolhida implementa a *Communications Device Class*, uma classe de dispositivos USB que é interpretada pelo computador como uma porta serial virtual, funcionando como um adaptador USB-Serial.



(a) Conexão das portas referentes à comunicação USB do PIC

(b) Esquemático

Figura 4.11: Conexão USB do PIC18F47J53.

#### 4.2.4 Migração para o PIC18F47J53

Com as bibliotecas desenvolvidas para o PIC16LF1829, a migração para o PIC18F47J53 foi, de certo modo, simples. Em termos de código, pequenas alterações em portas de entrada/saída e nomes de registradores foram necessários, mas o cerne das funções permaneceu o mesmo. Foi feito procedimento similar ao feito com o

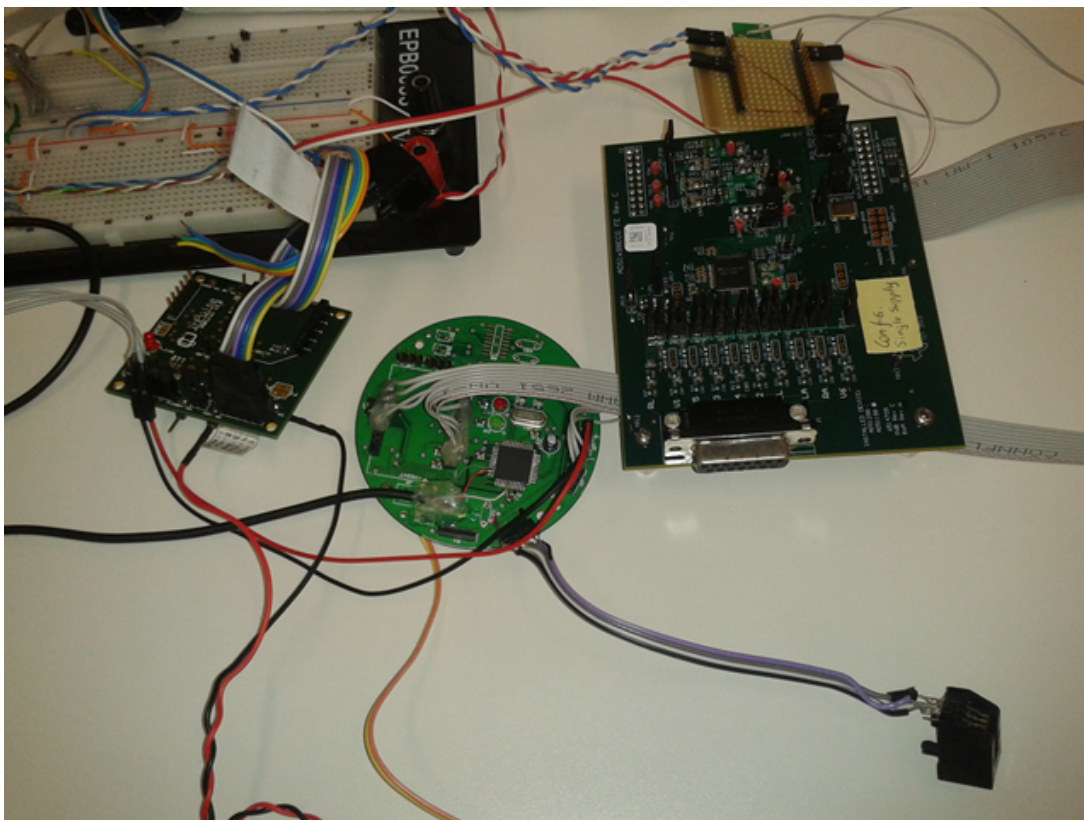


Figura 4.12: Migração para o PIC18F47J53.

PIC16, testando os módulos separadamente. Ao final, obteve-se bibliotecas igualmente

funcionais no PIC18.

#### 4.2.5 Adição dos Botões de Ligar/Desligar e *Master Clear*

O ECG, como todo aparelho, possui um botão de ligar/desligar, bem como um botão de *reset*, para reiniciar o aparelho em caso de funcionamento fora do comum. Os botão de ligar/desligar é um *push-button*, que deve gerar um sinal limpo de nível lógico alto quando pressionado, e nível lógico baixo quando solto.

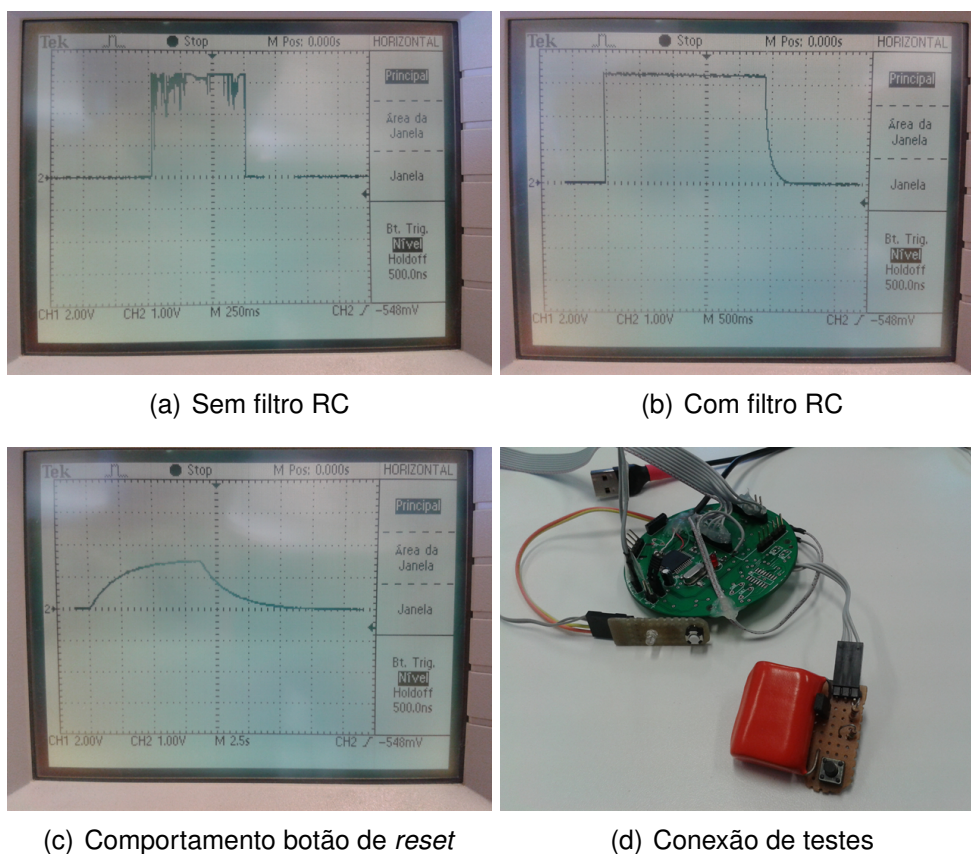


Figura 4.13: Comportamento dos botões visto em um osciloscópio.

Porém, testando esse botão com um osciloscópio, pode-se ver que muito repique é produzido ao pressioná-lo (Figura 4.13(a)). Para resolver isso, foi adicionado um filtro RC – o capacitor carrega quando o botão é pressionado e descarrega ao liberá-lo, gerando uma curva de queda do sinal mais suave (Figura 4.13(b)). O botão foi conectado à uma porta específica do PIC, para gerar interrupções ao ser pressionado – o PIC faz então uma alternância do estado ligado/desligado cada vez que o botão é pressionado.

O botão de *reset* deve ter um comportamento um pouco diferente. É um botão que deve ser pressionado por um período maior de tempo para funcionar, de modo

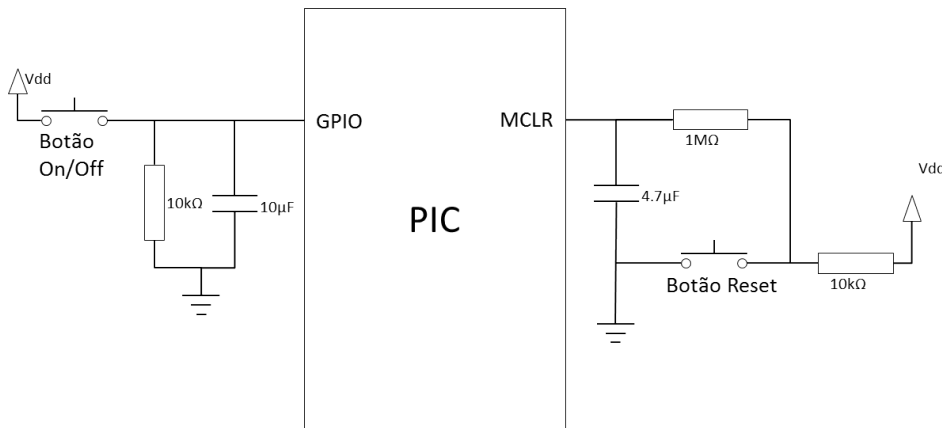


Figura 4.14: Esquemático da conexão dos botões com o PIC.

a evitar *resets* inconvenientes. Para isso foi adicionado também um circuito RC, mas com maior capacitância e relação R/C, para que o capacitor demore a ser carregado (Figura 4.13(c)). Esse botão é conectado à porta MCLR do PIC, que gera o *reset* do microcontrolador. Na Figura 4.13(d) pode-se ver a conexão de testes realizada, e na Figura 4.14 o esquemático do circuito montado.

#### 4.2.6 Interface I<sup>2</sup>C e BQ27541-G1

Para a comunicação I<sup>2</sup>C foi testado inicialmente o uso de bibliotecas internas do compilador C18, da *Microchip Technologies Inc.*, específico para PIC18. Para depurar os erros, continuou sendo utilizada a interface UART. Sem resultados positivos, o próximo passo foi criar uma biblioteca baseado nas funções já existentes. Algumas

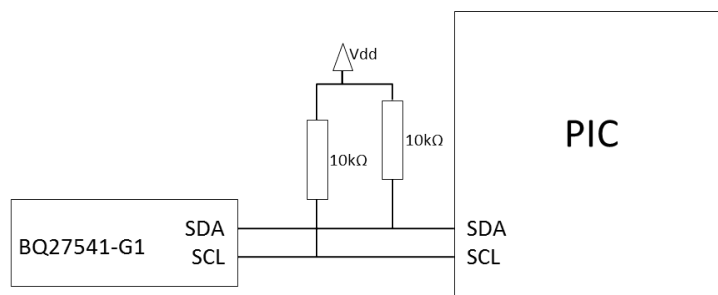


Figura 4.15: Esquemático da conexão PIC + BQ27541-G1.

alterações foram feitas para deixar o código mais específico para o PIC utilizado, e com isso foi possível estabelecer uma conexão estável com o *fuel gauge*. As conexões seguiram o padrão definido no protocolo I<sup>2</sup>C e também na datasheet do CI, e podem ser vistas na Figura 4.15.

Foi criada então uma biblioteca para o BQ27541-G1, com os comandos possivelmente úteis ao ECG, bem como alguns comandos relativos à calibração do CI e



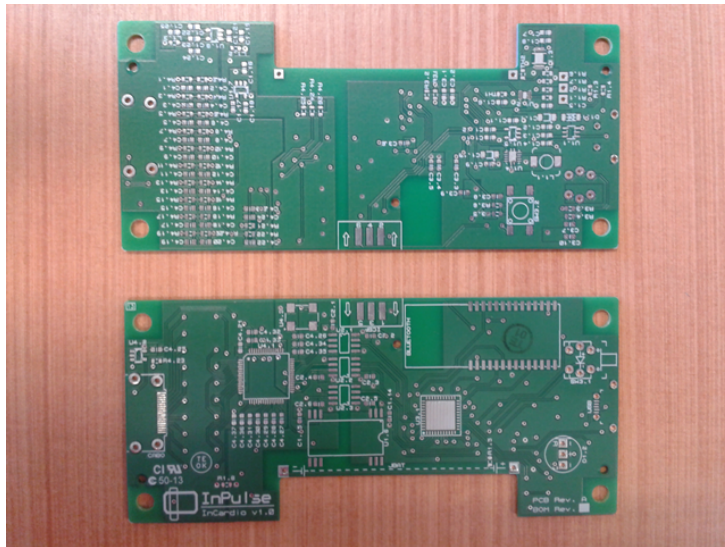


Figura 4.17: Primeira versão da placa.

### 4.3 Protocolo de Comunicação

Para que o dispositivo se comunique com o *software*, é necessário que ambos utilizem os mesmos códigos. O protocolo de comunicação é a padronização desses códigos. Nele, define-se o formato padrão para as mensagens e o fluxo que essas mensagens devem seguir.

No protocolo criado, foram definidos 3 tipos de mensagem: comandos normais, pacotes de amostras do ECG e comandos do administrador. Todos utilizam a mesma estrutura de mensagem, porém carregam mais ou menos informação. A estrutura padrão para mensagens é composta basicamente por *bytes* de controle (como um *byte* de início ou de checagem de erros), que precedem/seguem os comandos e dados a serem enviados.

Definiram-se então os *bytes* relacionados aos possíveis comandos transmitidos pelo *software*, e respostas enviadas pelo aparelho. Cada situação prevista recebeu um *byte* ou conjunto de *bytes* distinto, para que pudesse ser processada separadamente, tanto no *software* quanto no *hardware*.

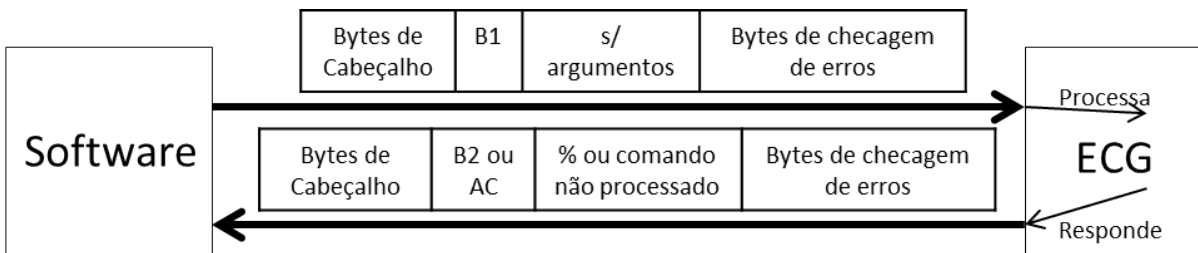


Figura 4.18: Exemplo de uma troca de mensagens entre *software* e *hardware*

Um exemplo pode ser visto na Figura 4.18, uma troca de mensagens hipotética em que o *software* faz uma requisição da carga restante na bateria, e na Tabela 4.1 os *bytes* de comando utilizados nessa operação.

<i>Byte</i>	Comando	Enviado por	Descrição
AC	NACK	Ambos	Notificação de falha no processamento da mensagem. Requer um <i>byte</i> subsequente com o comando da mensagem recebida.
B1	ASK_BAT_CHARG	<i>Software</i>	Pedido de leitura do nível de carga da bateria do dispositivo (%). Recebe MY_BAT_CHARG ou NACK como resposta.
B2	MY_BAT_CHARG	<i>Hardware</i>	Envia as informações do nível de carga da bateria do dispositivo (%). Requer um <i>byte</i> subsequente contendo um valor inteiro representando a carga da bateria.

Tabela 4.1: Exemplo de *bytes* do protocolo de comunicação





# Capítulo 5: Resultados

Ao fim desse trabalho, atingiu-se uma etapa em que o dispositivo está em uma primeira versão “completa”. A Figura 5.1 mostra a placa desenvolvida com o *case* aberto. As dimensões do dispositivo são: 12cm x 6,9cm x 1,8cm. Essas dimensões mostram como o aparelho é pequeno, ainda mais quando comparado aos equipamentos atualmente no mercado.

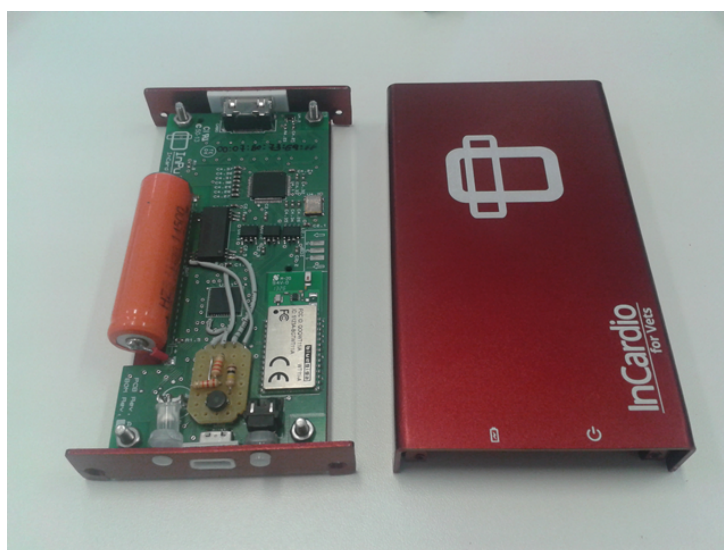


Figura 5.1: Equipamento aberto.

Com o atual dispositivo, foi possível atingir um máximo de 500SPS, apenas o mínimo desejado, devido às limitações não previstas na comunicação entre o módulo de comunicação *Bluetooth* e o PIC18F47J53. Também relativo à comunicação *Bluetooth* foi necessário o desenvolvimento de um *dongle* para o computador, pela dificuldade de utilizar módulos já presentes em computadores e também pelo fato de nem todos os computadores possuírem comunicação através dessa interface.

Buscando maior facilidade no manuseio, foram montados diversos tipos de cabo. Dependendo da quantidade de derivações desejadas pelo médico veterinário, é possível utilizar cabos com mais ou menos fios. Na Figura 5.2, pode-se ver dois conjuntos que exemplificam o que será comercializado em breve: o eletrocardiógrafo, *dongle* e vários *sets* de cabos – de 12, 6 e 1 derivações. Os dois conjuntos foram enviados no mês de fevereiro para São Paulo, para que o equipamento e o sistema sejam validados, testados e estressados pelo doutor Luís Felipe Neves dos Santos, graduado em Medicina Veterinária pela Universidade Federal de Uberlândia e mestre

em Cardiologia pela Universidade Federal de São Paulo, e atual diretor de regionais da Sociedade Brasileira de Cardiologia Veterinária.



Figura 5.2: Conjuntos enviados para a realização de testes.

Para testar o dispositivo, realizou-se a aquisição dos sinais de três formas diferentes: utilizando o kit do ADS1298, o equipamento eletrocardiógrafo da *Berger* e o equipamento desenvolvido neste trabalho. As aquisições não foram realizadas ao mesmo tempo pela possibilidade de interferência entre os equipamentos. Na Figura 5.3, pode-se ver a comparação dos sinais obtidos.

Com base nos sinais adquiridos, podemos verificar que o sinal entregue pelo equipamento desenvolvido possui qualidade tão boa quanto o equipamento comercial (*Berger*) e é coerente com os sinais obtidos pelos equipamentos já validados. Isso torna possível dizer que o sinal obtido é suficiente para que um médico cardiologista possa gerar laudos a partir de exames realizados com o equipamento.

Nas Figuras 5.4, 5.5, 5.6, é possível ver exemplos de telas de aquisição do *software* desenvolvido. O programa implementa uma série de filtros digitais ao sinal obtido, para eliminar ruídos como os referentes à rede elétrica (60Hz) e outros inseridos por diferentes fatores como posicionamento dos eletrodos, sinais de alta frequência, entre outros.

O *software* mostra o sinal que está sendo adquirido na tela, mas para gerar laudos o usuário deve gravar o exame para que se possa realizar as análises sobre os sinais depois. Tipicamente, a tela de aquisição funciona para que se possa observar a qualidade do sinal, possibilitando alterações na realização do exame, como por exemplo

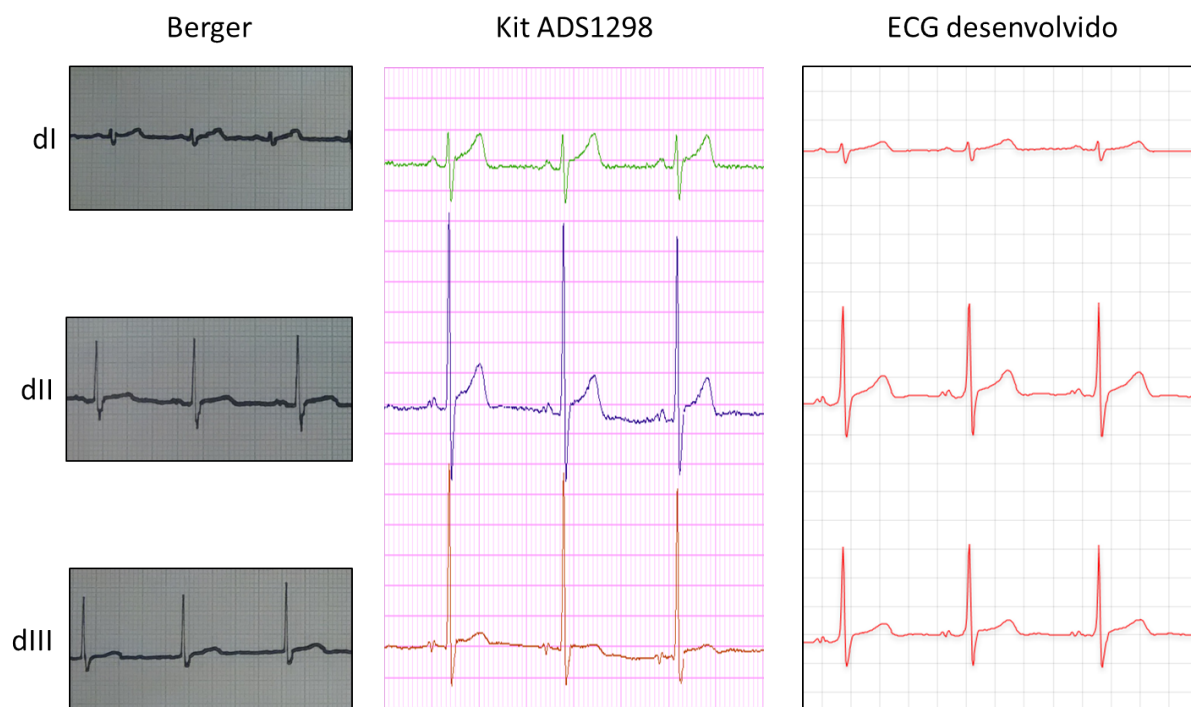


Figura 5.3: Sinais captados com diferentes equipamentos (derivações bipolares)

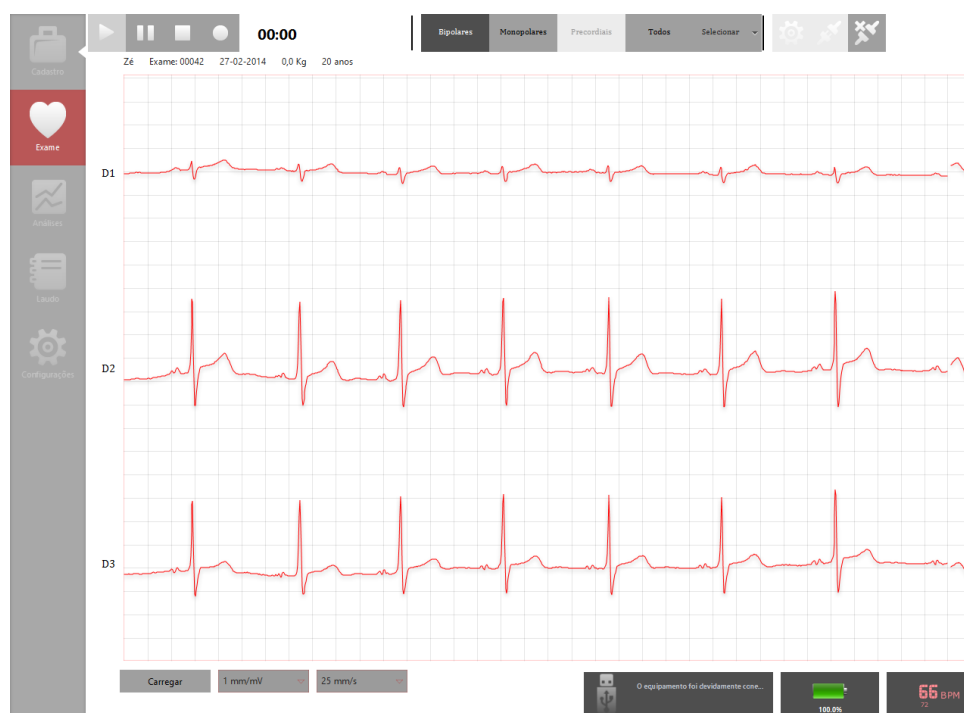


Figura 5.4: Tela do programa durante aquisição (derivações bipolares).

o reposicionamento dos eletrodos. Na Figura 5.7 pode-se ver a tela de análise dos sinais promediados, que é a média das ondas de ECG de cada derivação.

Também relacionado ao desenvolvimento do *firmware* para o microcontrolador, um segundo equipamento foi construído e programado. Trata-se de um sistema detector



Figura 5.5: Tela do programa durante aquisição (derivações aumentadas).

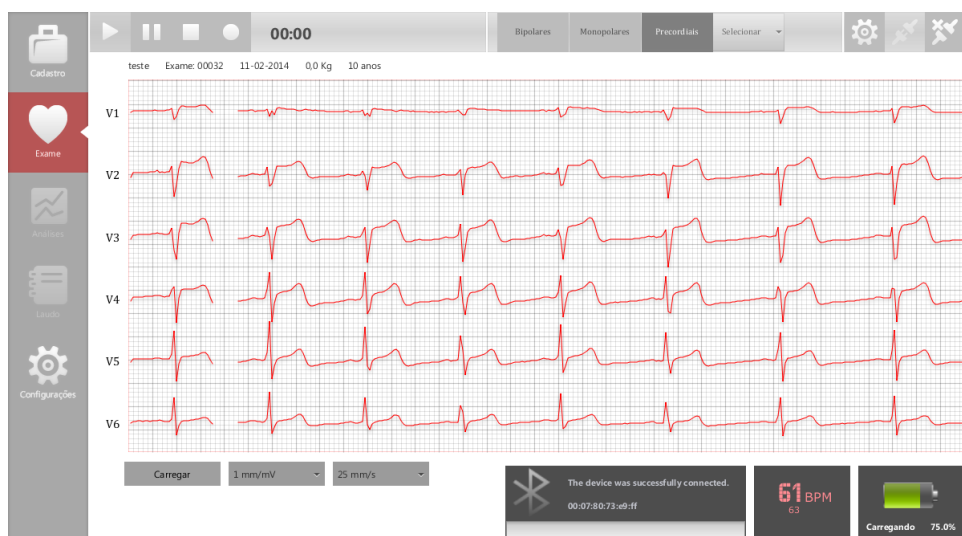


Figura 5.6: Tela do programa durante aquisição (derivações precordiais).

de neuropatia autonômica – doença no sistema nervoso autônomo, que controla as funções involuntárias do corpo. O equipamento funciona adquirindo sinais biomédicos, como sinais eletrocardiográficos, pressão sanguínea, etc. Devido ao funcionamento ser similar – aquisição de sinais biomédicos – o dispositivo foi projetado de forma também similar, utilizando o mesmo conversor analógico e microcontrolador do ECG. Dessa forma, foi possível aproveitar o desenvolvimento obtido no projeto do eletrocardiógrafo para produzir um *firmware* para o novo dispositivo. Na Figura 5.8 pode-se ver o hardware desenvolvido e em destaque os módulos utilizados em ambos os projetos.



Figura 5.7: Tela de análise dos sinais (sinais promediados)

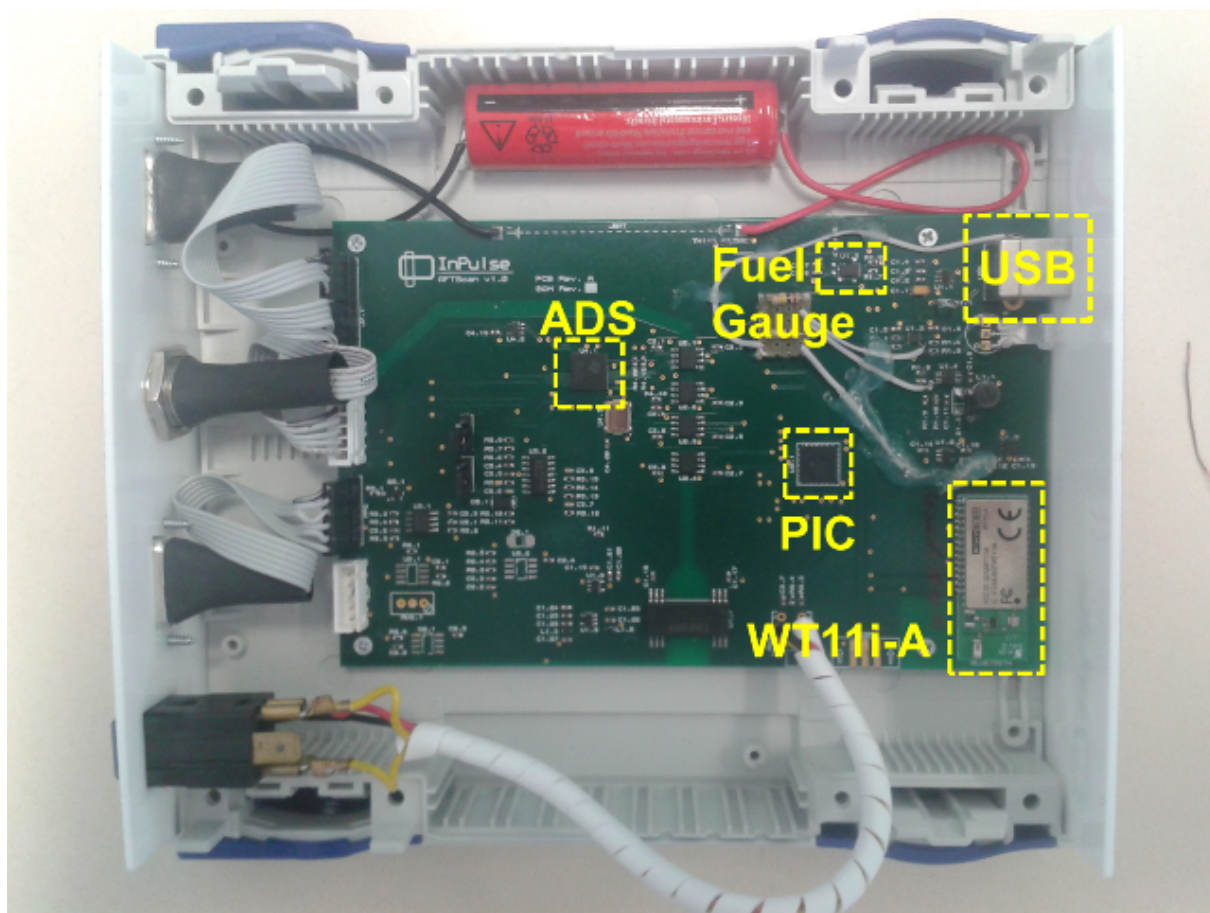


Figura 5.8: Equipamento detector de sinais autônomicos.



## Capítulo 6: Conclusão e Perspectivas

O projeto em que se inclui este trabalho previa o desenvolvimento de um dispositivo ECG portátil, flexível e de alta resolução. O desenvolvimento foi realizado do início, com a escolha dos componentes a serem utilizados, ao fim, com a implementação de um *firmware* para o microcontrolador embarcado no dispositivo, a fim de integrar os módulos presentes no aparelho. O trabalho, apesar de focado neste fim, não se limitou somente a ele, mas se estendeu sobre os outros aspectos do projeto – referentes ao desenvolvimento do *hardware* e *software* – pela relação natural de sobreposição dessas áreas.

Ao fim deste trabalho, tendo sido obtido um dispositivo funcional, pode-se dizer que os objetivos foram atingidos com sucesso. Não é possível afirmar, contudo, que o projeto realmente chegou ao fim. Como busca-se um equipamento de alta resolução, é importante atingir frequências de amostragem maiores do que a atingida neste protótipo (500SPS). Para atingir esse patamar, uma troca de processadores pode ser necessária, o que levaria a um reprojeto do equipamento.

Além disso, o projeto do ECG já abre muitas possibilidades para o futuro. A transição da área de medicina veterinária para humana sendo a mais óbvia, que possivelmente exigiria apenas pequenas alterações no *hardware* e nos processos de fabricação, para que se atingisse as requisições impostas pelos órgãos regulamentadores de equipamentos médico-hospilares, e consequentes mudanças no *firmware*. O *software* receberia uma “repaginação”, para que focasse em pontos de maior importância na análise de exames de ECGs em humanos.

O desenvolvimento de um Holter – aparelho de ECG que funciona por longos períodos de tempo (pelo menos 24h), porém com resolução mais baixa – é outra opção bastante viável. Para um projeto desse tipo, pode-se simplesmente aplicar o conhecimento adquirido com o ECG, bem como reutilizar praticamente todo o equipamento. Alguns requisitos novos estariam presentes nesse projeto, como a utilização de algum tipo de memória interna ao dispositivo, ou a questão da duração da bateria, levando à uma busca por ainda maior economia de energia.

Outra possibilidade é o desenvolvimento de um ECG de longa duração, um Holter com alta resolução. Uma análise de viabilidade seria necessária, mas um equipamento do tipo traria mais informações do que um Holter para o cardiologista, permitindo o diagnóstico de um número maior de problemas no coração do paciente.

Também fundamentado em um dispositivo de ECG, seria interessante o desenvolvimento de um dispositivo monitorador de eventos. Esse tipo de dispositivo é utilizado por períodos ainda mais longos de tempo, e serve para detectar “eventos” específicos que podem acontecer com o coração. Na ocorrência desses eventos, o aparelho salva uma janela de tempo, abrangendo determinado período decorrido antes e depois do acontecimento.



# Referências Bibliográficas

- [1] C. Atkins, J. Bonagura, S. Ettinger, P. Fox, S. Gordon, J. Haggstrom, R. Hamlin, B. Keene, V. Luis-Fuentes, and R. Stepien. *Guidelines for the Diagnosis and Treatment of Canine Chronic Valvular Heart Disease*. Blackwell Publishing Inc, 2009.
- [2] John E Rush. Chronic valvular heart disease in dogs. In *Proceedings from: 26th Annual Waltham Diets/OSU Symposium for the Treatment of Small Animal Cardiology*, 2002.
- [3] Luiz Henrique Filippi. *O Eletrocardiograma na Medicina Veterinária*. Roca, 2011.
- [4] Dee Unglaub Silverthorn. *Fisiologia humana: uma abordagem integrada*. Artmed, 5th edition, 2010.
- [5] Website do museu de ciências de Londres (visitado em 16/01/2014). <http://www.sciencemuseum.org.uk/broughttolife/people/willemeinthoven.aspx>.
- [6] D. Dubin and I.C. da Silveira. *Interpretação rápida do ecg: ... um curso programado*. Editora de Publicações Científicas, 2001.
- [7] Leslie K. Muma and Barbara Ritter. *Basics of EKG Interpretation: A Programmed Study*. 1998.
- [8] Luís Pedro Vieira da Costa Pereira. *A utilidade do eletrocardiograma de 12 derivações no diagnóstico de dilatação das câmaras cardíacas esquerdas associada à doença mixomatosa da válvula mitral em cães*. Universidade de Trás-os-Montes e Alto Douro, 2011.
- [9] Stephen J. Ettinger and Edward C. Feldman. *Textbook of Veterinary Internal Medicine*. Elsevier, 2005.
- [10] Marc S. Kraus, N. Sydney Moïse, Mark Rishniw, Nathan Dykes, and Hollis N. Erb. *Morphology of Ventricular Arrhythmias in the Boxer as Measured by 12-Lead Electrocardiography with Pace-Mapping Comparison*. Blackwell Publishing Ltd, 2002.
- [11] Frank Durda. *Serial and UART Tutorial*. FreeBSD Documentation, 1996.
- [12] *PIC16F/LF1895/1829 Data Sheet*. Microchip Technology Inc., 2010.
- [13] *PIC18F47J53 Family Data Sheet*. Microchip Technology Inc., 2010.

- [14] *ADS1298 Data Sheet*. Texas Instruments Incorporated, 2012.
- [15] *WT11i-A Data Sheet*. Bluegiga Technologies, 2013.
- [16] *iWRAP5 User Guide*. Bluegiga Technologies, 2012.
- [17] *bq27541-G1 Data Sheet*. Texas Instruments Incorporated, 2012.