

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Julio de Siqueira

**Desenvolvimento de Interface Homem-Máquina (IHM)
configurável para monitorização de sinais fisiológicos**

Itajubá, abril de 2013.

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Julio de Siqueira

**Desenvolvimento de Interface Homem-Máquina (IHM)
configurável para monitorização de sinais fisiológicos**

**Dissertação submetida ao Programa de Pós-Graduação
em Engenharia Elétrica como parte dos requisitos para
obtenção do Título de Mestre em Ciências em
Engenharia Elétrica.**

Área de Concentração: Automação e Sistemas Elétricos
Industriais

Orientador: Prof. Dr. Paulo César Crepaldi

Coorientador: Dr. Roberto Castro Júnior

Abril de 2013

Itajubá - MG

Ficha catalográfica elaborada pela Biblioteca Mauá –
Bibliotecária Margareth Ribeiro- CRB_6/1700

S618d

Siqueira, Julio de

Desenvolvimento de Interface Homem-Máquina (IHM) confi-
gurável para monitorização de sinais fisiológicos / Julio de Si-
queira. -- Itajubá, (MG) : [s.n.], 2013.

82 p. : il.

Orientador: Prof. Dr. Paulo César Crepaldi.

Coorientador: Prof. Dr. Roberto Castro Júnior.

Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. Monitorização de sinais fisiológicos. 2. Interface homem-
máquina. 3. Eletrocardiografia. I. Crepaldi, Paulo César, orient.
II. Castro Júnior, Roberto, coorient. III. Universidade Federal de
Itajubá. IV. Título.

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Julio de Siqueira

**Desenvolvimento de Interface Homem-Máquina (IHM)
configurável para monitorização de sinais fisiológicos**

Dissertação aprovada por banca examinadora em 26 de abril de 2013, conferindo ao autor o título de **Mestre em Ciências em Engenharia Elétrica**.

Banca examinadora:

Prof. Dr. Paulo César Crepaldi (Orientador)

Dr. Roberto Castro Júnior (Coorientador)

Prof. Dr. José Feliciano Adami

Prof. Dr. Ismael Noronha

Itajubá

2013

À minha família.

Agradecimentos

Primeiramente, gostaria de agradecer à minha família que, apesar da distância, sempre me estimulou e apoiou incondicionalmente.

Agradeço também aos meus amigos pelo constante suporte e pelos bons momentos de reflexão e descontração.

Aos orientadores, Prof. Dr. Paulo César Crepaldi e Dr. Roberto Castro, agradeço pelo apoio, confiança e constante auxílio no desenvolvimento deste trabalho.

Ao Grupo de Microeletrônica da Universidade Federal de Itajubá, agradeço por disponibilizar sua estrutura e a seus membros pelo acolhimento e companheirismo durante o período em que estive lá.

Gostaria de agradecer ainda à empresa Minas IC pela confecção dos pôsteres apresentados no XXIII CBEB (outubro de 2012).

Por fim, agradeço à CAPES pelo apoio financeiro que possibilitou o desenvolvimento deste trabalho.

“Saber que se sabe o que se sabe é, na verdade, saber que não se sabe o que não se sabe. Eis a verdadeira ciência.”

Confúcio

Resumo

Este trabalho apresenta o desenvolvimento de uma interface homem-máquina (IHM) configurável e amigável para monitorização de sinais fisiológicos (temperatura, eletrocardiograma e frequência cardíaca calculada a partir do sinal eletrocardiográfico) e físicos (acelerometria), sem a utilização de fios, de seres humanos durante atividades físicas, bem como a caracterização das condições ambientes através de medições de temperatura ambiente e umidade relativa do ar.

O sistema proposto poderá ser utilizado como ferramenta no auxílio à análise do desempenho físico de atletas e paratletas durante treinamento, sessões de reabilitação física e motora e onde mais se justifique a monitorização de sinais físicos e fisiológicos durante atividades físicas.

O sistema de monitorização é constituído por dois módulos de dimensões reduzidas, garantindo portabilidade e facilidade de manuseio. O primeiro módulo acompanha o usuário e é capaz de receber e transmitir, via ondas de rádio, os diversos sinais a serem monitorizados. O segundo módulo é responsável pela recepção dos sinais enviados pelo primeiro módulo e transmissão dos mesmos ao computador pessoal para tratamento final, exibição e armazenamento.

O programa de computador possibilita ao operador: o acompanhamento, em tempo real, da evolução dos sinais de interesse; a configuração de alguns parâmetros relativos ao desempenho do programa e exibição das grandezas medidas; o armazenamento adequado dessas informações para consultas posteriores. Dessa maneira, tem-se um sistema de monitorização de grande flexibilidade e altamente configurável.

Para explicar o projeto da interface homem-máquina, são abordados, primeiramente, os desafios e escolhas realizadas a respeito da estrutura do programa desenvolvido.

Após isso, são mostrados e analisados os resultados pertinentes às diferentes etapas da implementação da interface, a partir de dados simulados e dados reais obtidos de banco de dados específico para o desenvolvimento de sistemas para aplicação médica (PhysioNet). Também é apresentado o *layout* final da interface homem-máquina.

Finalmente, é mostrada a estrutura do programa obtido, representada através de um diagrama de classes, e são apresentadas conclusões sobre o trabalho descrito.

Palavras-Chave: Monitorização de sinais fisiológicos, interface homem-máquina, eletrocardiografia.

Abstract

This work presents the development of a friendly and configurable Human-Machine Interface (HMI) for monitoring physiological (temperature, electrocardiogram and calculated heart rate) and physical (accelerometry) signals, wirelessly, of human beings during physical activities, as well as the characterization of environmental conditions through measurements of room temperature and relative air humidity.

The proposed system can be utilized as a tool to aid physical performance analysis of athletes and parathletes during training, physical and motor rehabilitation sessions and every other situation when monitoring physical and physiological signals during physical activities is justified.

The monitoring system is comprised by two small-sized modules, ensuring portability and ease of usage. The first module accompanies the user and is able to receive and transmit, via radio waves, the several signals monitored. The second module is responsible for the reception of the information sent by the first module and transmission of such information to the personal computer for final processing, exhibition and storage.

The proposed software allows the operator to: visualize, in real time, the evolution of the signals of interest; configure parameters relative to the software's performance and display of the measured signals; proper storage of the information for further consultation. Thus, it is envisioned a highly flexible and configurable monitoring system.

In order to explain the human-machine interface's design, the first topics covered refer to the challenges and choices made regarding the developed software's structure.

Later, results are shown and analyzed concerning the different phases of the interface's implementation. To do so, simulated data and real data were used. In order to obtain reliable real data, PhysioNet data bank was used. The human-machine interface's final layout is also presented.

Finally, the software's structure is shown, represented through a class diagram, and conclusions about this work are presented.

Keywords: Physiological signals monitoring, human-machine interface, electrocardiography.

Sumário

Agradecimentos	6
Resumo	6
Abstract.....	10
Sumário.....	11
Índice de Figuras	13
Índice de Tabelas	15
Lista de Abreviaturas.....	16
Lista de Símbolos	17
1 Introdução.....	16
1.1 Considerações iniciais.....	16
1.2 Justificativa	17
1.3 Objetivos	18
1.4 Estrutura do trabalho.....	19
2 Fundamentação Teórica	21
2.1 Considerações iniciais.....	21
2.2 Introdução à fisiologia cardíaca	21
2.1.1 Estrutura do coração	21
2.1.2 Potenciais de ação.....	22
2.1.3 Ciclo Cardíaco	23
2.1.4 Eletrocardiograma normal	23
2.1.5 Derivações eletrocardiográficas	24
2.1.6 Determinação da frequência cardíaca.....	26
2.2 Introdução à programação orientada a objetos	26
2.2.1 Visão Geral	26
2.2.2 Conceitos básicos	27
2.2.3 Diagrama de classes.....	28
2.2.4 <i>Multi-threading</i>	31
3 Metodologia e revisão bibliográfica.....	32
3.1 Considerações iniciais.....	32
3.2 Comunicação USB.....	32

3.3	Decodificação do protocolo	33
3.4	Estrutura dos buffers	34
3.5	Exibição de múltiplas grandezas para múltiplos usuários	35
3.6	Processamento adicional	36
3.6.1	Filtro digital	36
3.6.2	Algoritmo <i>Turning Point</i>	37
3.6.3	Detecção de complexo QRS e cálculo de frequência cardíaca.....	37
3.6.4	Linearização de temperatura corpórea.....	38
3.7	Interface visual.....	38
3.8	Estrutura do <i>software</i>	40
4	Resultados	42
4.1	Considerações iniciais.....	42
4.2	Comunicação USB.....	42
4.3	Decodificação do protocolo	44
4.4	Estrutura dos buffers	45
4.5	Exibição de múltiplas grandezas para múltiplos usuários	46
4.6	Processamento adicional	49
4.6.1	Filtros digitais	49
4.6.2	Algoritmo de <i>Turning Point</i>	50
4.6.3	Detecção de complexo QRS e cálculo de frequência cardíaca.....	53
4.6.4	Linearização de temperatura.....	64
4.7	Interface visual.....	65
4.8	Estrutura do <i>software</i>	70
	Conclusão e Trabalhos Futuros	73
	Referências Bibliográficas.....	74
	Apêndice A – Publicações	76

Índice de Figuras

Figura 1.1 – Sistema proposto para a monitorização multiparamétrica sem fios de sinais fisiológicos de múltiplos usuários com destaque para o trecho abordado neste trabalho.	17
Figura 2.1 – Estrutura do coração.....	22
Figura 2.2 – Excitação elétrica do coração.....	23
Figura 2.3 – Eletrocardiograma normal.....	24
Figura 2.4 – Ligações entre eletrocardiógrafo e corpo humano para as três derivações bipolares padrão.....	25
Figura 2.5 – Sinais eletrocardiográficos das três derivações padrão.....	25
Figura 2.6 – Exemplo de herança.	28
Figura 2.7 – Diagrama detalhando cada classe do sistema exemplo.....	29
Figura 2.8 – Diagrama mostrando as relações entre as classes do sistema exemplo.	30
Figura 4.1 – Arquitetura de <i>drivers</i> D2XX.....	43
Figura 4.2 – Pacote, ou <i>frame</i> , genérico do protocolo.....	44
Figura 4.3 – Visualização tridimensional dos <i>buffers</i>	45
Figura 4.5 – Ondas eletrocardiográficas do primeiro usuário.	47
Figura 4.6 – (a) Onda eletrocardiográfica do segundo usuário; (b) ondas eletrocardiográficas de dois usuários exibidas simultaneamente.	48
Figura 4.7 – Respostas de magnitude e fase do filtro <i>notch</i> de 60 Hz.	49
Figura 4.8 – Representação das decisões do algoritmo <i>Turning Point</i> [18].....	50
Figura 4.9 – Sinal eletrocardiográfico original.....	51
Figura 4.10 – (a) Sinal eletrocardiográfico submetido ao algoritmo <i>Turning Point</i> uma vez; (b) sinal eletrocardiográfico submetido ao algoritmo <i>Turning Point</i> duas vezes.....	52
Figura 4.11 – (a) Sinal após o filtro passa-faixa; (b) sinal após a diferenciação; (c) sinal após a transformação quadrática; (d) sinal após o integrador de média móvel.	53
Figura 4.12 – (a) Sinal após filtro passa-baixas; (b) sinal após filtro passa-altas.....	56
Figura 4.13 – (a) Sinal derivado; (b) sinal após a transformação quadrática; (c) sinal após integrador de janela móvel.	57
Figura 4.14 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.	58
Figura 4.15 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.	60
Figura 4.16 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.	61
Figura 4.17 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.	62
Figura 4.18 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.	63
Figura 4.19 – Interface visual final.....	65
Figura 4.20 – Aba destinada ao gráfico de ECG.	67

Figura 4.21 – (a) Aba destinada aos gráficos dos três eixos do sinal de acelerometria; (b) aba destinada à frequência cardíaca.	68
Figura 4.22 – (a) Aba destinada ao sinal de temperatura corpórea; (b) aba destinada aos sinais do ambiente (temperatura e umidade relativa do ar).	69
Figura 4.23 – Diagrama detalhando as classes do programa.....	70
Figura 4.24 – Diagrama mostrando as relações entre as classes do programa.....	72

Índice de Tabelas

Tabela 4.1 – Valores de temperatura medidos e linearizados	64
------------------------------------------------------------------	----

Lista de Abreviaturas

IHM/HMI	-	Interface Homem-Máquina/ <i>Human Machine Interface</i> .
USB	-	<i>Universal Serial Bus</i> .
PC	-	<i>Personal Computer</i> .
ANSI	-	<i>American National Standards Institute</i> .
ECG	-	Eletrocardiograma.
S-A	-	Sinoatrial.
A-V	-	Átrio-ventricular.
QRS	-	Complexo que envolve as ondas eletrocardiográficas Q, R e S.
OOP	-	<i>Object-Oriented Programming</i> .
NTC	-	<i>Negative Temperature Coefficient</i> .
UART	-	<i>Universal Asynchronous Receiver/Transmitter</i>
VCP	-	<i>Virtual COM Port</i> .
COM	-	Designação de portas de comunicação serial.
API	-	<i>Application Programming Interface</i> .
D2XX	-	Driver USB proprietário da FTDI Ltd.
ID	-	Identificação de usuário.
CRC	-	<i>Cyclic Redundancy Check</i> .
FCS	-	<i>Frame Checking Sequence</i> .
SNR	-	<i>Signal-to-Noise Ratio</i> .
BPM	-	Batimentos por minuto.

Lista de Símbolos

f_c	-	Denota frequência cardíaca
n_{QRS}	-	Denota quantidade de complexos QRS.
T	-	Denota o período de tempo utilizado para o cálculo da frequência cardíaca.
$x(n)$	-	Denota a posição n do vetor de entrada.
$y(n)$	-	Denota a posição n do vetor de saída.
N	-	Denota a quantidade de amostras utilizadas na janela móvel.
t_{QRS}	-	Denota tempo de interesse para detecção de complexos QRS.
T_s	-	Denota período de amostragem.

1 Introdução

1.1 Considerações iniciais

A monitorização de sinais fisiológicos abrange diversas situações, mais notavelmente em residências e hospitais tanto para prevenção quanto para acompanhamento do desenvolvimento de quadros clínicos [1].

Outra área de interesse para a monitorização de sinais fisiológicos é a da reabilitação motora, uma vez que, sem a referida monitorização, não há como quantificar o desempenho físico do paciente durante a atividade. Dessa forma, a avaliação do desempenho do paciente fica dependente apenas da experiência do profissional de saúde [2]. Visando minimizar estes problemas, vários sistemas sem fio voltados para a supervisão de atividades de reabilitação motora já foram propostos [3].

A área esportiva também está se tornando cada vez mais científica e conta com auxílio tecnológico tanto para identificar situações de risco potencial para atletas quanto para otimização de treinamentos, que acabam por melhorar o desempenho de atletas em competições [4]. Consequentemente, sistemas de monitorização sem fio voltados para utilização por atletas têm sido desenvolvidas, como em [5].

Entretanto, a monitorização de sinais fisiológicos muitas vezes é realizada em ambientes controlados, limitando a duração e o realismo dos ensaios, como no caso de atletas realizando atividades em laboratórios que não correspondem à realidade de seus esportes [6]. Além disso, quando o objetivo é a monitorização durante a realização de atividades físicas, é interessante que o sistema seja biomecanicamente neutro ao atleta [7].

Com o desenvolvimento tecnológico, sistemas de monitorização sem fio têm atraído grande atenção por parte de pesquisadores e desenvolvedores de produtos. Tais sistemas de monitorização têm sido direcionados a diversas áreas, como: monitoramento de doenças crônicas, idosos, reabilitação de pacientes, esportes [8].

O sistema que engloba o trabalho apresentado nessa dissertação foi concebido para realizar a monitorização de sinais fisiológicos de diferentes indivíduos durante a realização de atividades físicas. O sistema consiste de dois módulos de circuitos eletrônicos, protocolo

específico para esta aplicação e programa de computador dedicado que funciona como interface homem-máquina. O primeiro módulo, de dimensões reduzidas, é acoplado a cada indivíduo, formando redes de sensores sem fio para a realização da transdução e transmissão, através de protocolo específico, dos sinais fisiológicos de interesse. O segundo módulo é conectado a um computador pessoal (PC) através de uma porta USB para possibilitar que os dados adquiridos pelos sensores possam ser visualizados e armazenados adequadamente. Finalmente, o programa dedicado finalizará o processamento dos dados e cuidará de sua exibição e armazenamento adequados [9].

O sistema descrito é representado na figura 1.1 abaixo.

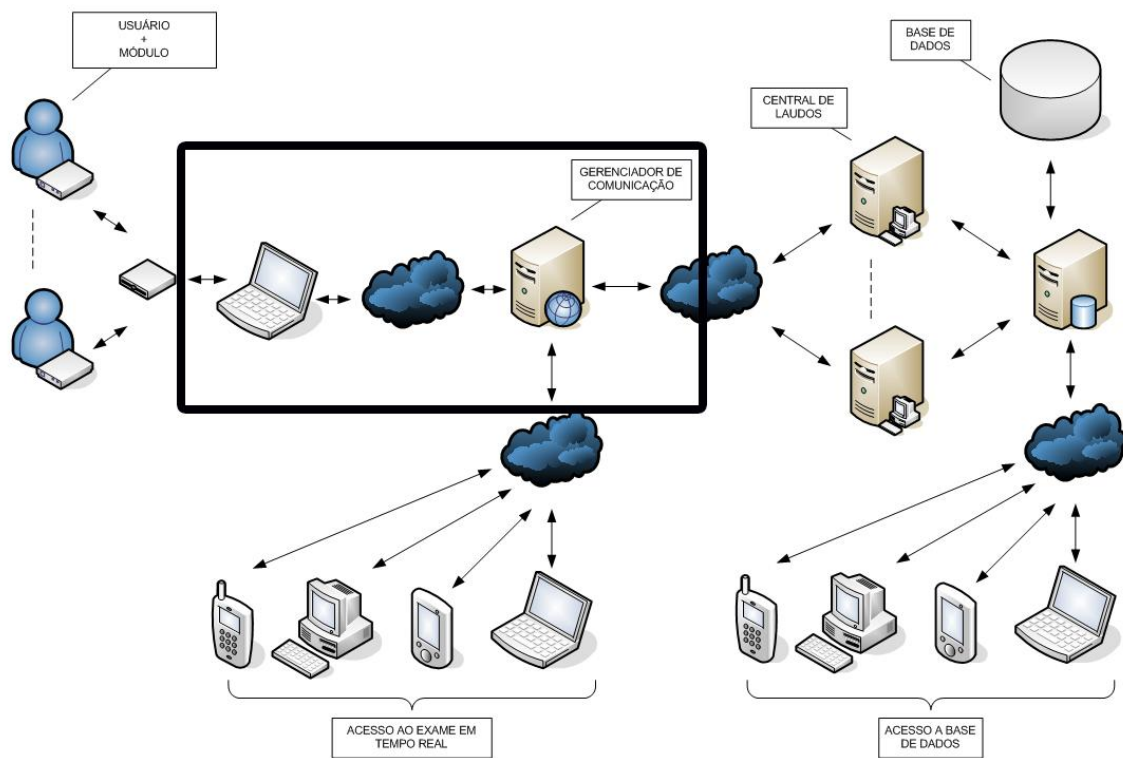


Figura 1.1 – Sistema proposto para a monitorização multiparamétrica sem fios de sinais fisiológicos de múltiplos usuários com destaque para o trecho abordado neste trabalho.

Fonte: Castro Jr. (2010).

1.2 Justificativa

Idealmente, um equipamento de monitorização de sinais fisiológicos, especialmente um voltado para utilização durante a execução de atividades físicas, não deveria interferir na mobilidade do usuário nem causar desconforto. Com tecnologias cada vez mais portáteis e

eficientes, o desenvolvimento de sistemas portáteis e sem fio que corroboram com essa premissa tem se popularizado [8].

Muitos desses sistemas visam a monitorização de sinais fisiológicos durante algum tipo de atividade física. Dentre as áreas para as quais tais projetos têm se voltado estão: *health-care*, terapia (reabilitação), pesquisa, esportes e outras [1, 2, 3, 4, 5, 6, 7].

O sistema aqui descrito, além de levar em conta essas premissas, se propõe a atender múltiplos usuários simultaneamente (visando a monitorização durante a prática de esportes coletivos, por exemplo). Para tanto, o alcance de comunicação deve ser significativo. Com isso em mente, a comunicação sem fio se dá por meio do protocolo ZigBee, que possibilita que cada membro da rede possa agir como repetidor, aumentando o alcance da comunicação, além da garantia de privacidade e integridade de informação. Tal escolha já se mostrou eficiente em sistemas semelhantes [8, 10].

1.3 Objetivos

O projeto do sistema referido anteriormente foi dividido em três partes:

- Desenvolvimento de circuitos eletrônicos tanto do módulo que acompanha o usuário quanto do módulo que se conecta ao computador;
- Desenvolvimento de protocolo específico para comunicação de sinais fisiológicos sobre rede ZigBee;
- Desenvolvimento de interface homem-máquina configurável para monitorização de sinais fisiológicos.

Essa divisão foi feita de modo que cada uma das partes do projeto originasse uma dissertação de mestrado.

Esta dissertação trata do desenvolvimento de uma interface homem-máquina configurável para monitorização de sinais fisiológicos.

A interface homem-máquina em que se baseia este trabalho tem por objetivo cumprir os requisitos descritos a seguir para que, uma vez integrada ao restante do projeto, permita que o mesmo cumpra sua proposta de maneira satisfatória.

O acompanhamento da atividade física monitorizada deve ser feito por um profissional de saúde, seja ele médico, profissional de educação física ou outro profissional capacitado para desempenhar tal função. Por isso, a interface deve ser amigável e intuitiva, possibilitando fácil configuração e utilização. O programa deve ser capaz de receber os dados que chegam pela porta USB do computador pessoal, ainda encapsulados pelo protocolo, tratá-los, exibi-los graficamente e armazená-los de maneira adequada. Entretanto, mesmo realizando todas essas tarefas o programa deve continuar respondendo aos comandos do operador.

A seção seguinte posiciona o leitor com relação às etapas necessárias para atingir os objetivos anteriormente estabelecidos e como elas se dispõem ao longo do texto.

1.4 Estrutura do trabalho

O presente trabalho consiste de cinco capítulos, sendo o primeiro de cunho introdutório e cujo objetivo é posicionar o leitor quanto ao trabalho que será melhor explicitado nos capítulos seguintes.

O segundo capítulo traz uma breve fundamentação teórica, abordando conceitos referentes à fisiologia cardíaca e programação orientada a objetos que são necessários à compreensão deste trabalho.

O terceiro capítulo aborda em detalhes as diferentes questões e decisões do projeto e os métodos utilizados no desenvolvimento do mesmo.

O quarto capítulo apresenta os resultados obtidos nas diversas fases do projeto. Por fim, visando estar em acordo com as normas relacionadas a equipamentos eletromédicos, mostra resultado do processamento de sinal eletrocardiográfico obtido a partir de sinais reais retirados do banco de dados livre PhysioNet. O portal Physionet oferece acesso gratuito a grandes coleções de dados fisiológicos gravados com o objetivo de fornecer ferramentas de testes para o desenvolvimento de pesquisas e equipamentos médicos, e é referência para a comunidade de pesquisadores da área biomédica [11]. Conforme indicado pelo portal, foram

adotadas recomendações da ANSI/AAMI (*American National Standards Institute/Association for the Advancement of Medical Instrumentation*) apresentadas no documento que trata de monitores cardíacos [12].

Finalmente, o quinto e último capítulo apresenta as conclusões sobre o desenvolvimento deste trabalho e deixa sugestões para trabalhos futuros.

2 Fundamentação Teórica

2.1 Considerações iniciais

Para o entendimento deste trabalho, são necessários alguns conceitos referentes à fisiologia do coração e seu funcionamento, bem como a compreensão do ciclo cardíaco e sua relação com as ondas eletrocardiográficas.

Também são necessários alguns conceitos básicos referentes à programação orientada a objetos para que o leitor possa entender a estrutura do programa desenvolvido.

Este segundo capítulo visa esclarecer esses conceitos para que o leitor possa compreender as diferentes etapas do desenvolvimento deste projeto.

2.2 Introdução à fisiologia cardíaca

Esta seção busca introduzir conceitos básicos referentes à fisiologia cardíaca para que o leitor possa compreender ondas eletrocardiográficas e o processamento computacional relacionado às mesmas.

2.1.1 Estrutura do coração

O coração, ilustrado na figura 2.1, é na verdade, formado por duas bombas separadas: o coração direito, que bombeia sangue para os pulmões, e o coração esquerdo, que bombeia sangue para os órgãos periféricos [13].

O coração consiste de quatro câmaras. Os átrios são antecâmaras que armazenam sangue durante a contração ventricular e, em seguida, bombeiam este sangue para o interior dos ventrículos. O bombeamento principal é realizado pelos ventrículos, enviando sangue ao sistema vascular corporal [14].

A contração coordenada entre átrios e ventrículos ocorre graças a um padrão específico de ativação elétrica na musculatura cardíaca [14].

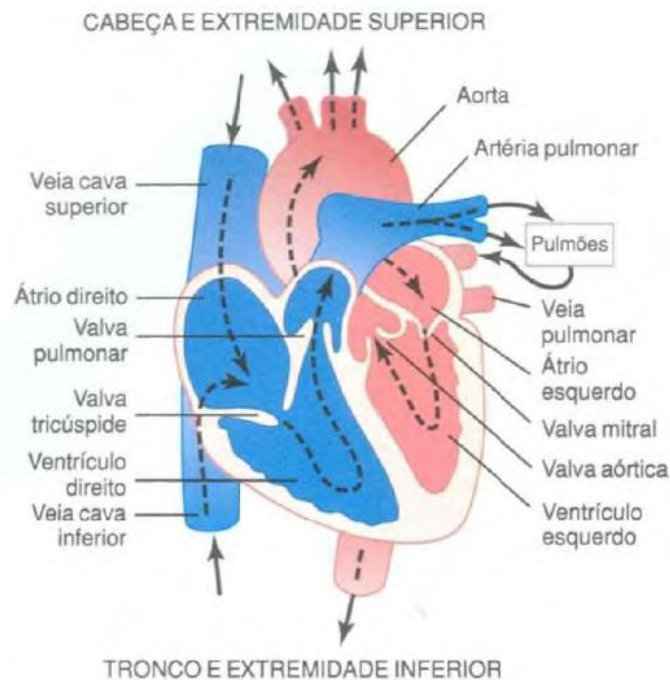


Figura 2.1 – Estrutura do coração.

Fonte: Guyton e Hall (2006).

2.1.2 Potenciais de ação

Antes da excitação, uma típica célula ventricular possui um potencial em repouso de aproximadamente -85 [mV]. A fase inicial de despolarização rápida possui uma taxa de subida geralmente superior a 150 [V/s]. Essa fase é seguida por uma rápida repolarização inicial que leva à manutenção de uma região (platô) de despolarização que dura cerca de 200 a 300 [ms]. Uma fase final de repolarização restaura o potencial da membrana ao nível de repouso que é mantido pelo restante do ciclo cardíaco [14].

Os mecanismos que originam o potencial de ação no músculo cardíaco são conhecidos como canais de sódio rápidos e canais de cálcio lentos ou canais de cálcio-sódio. Tais mecanismos controlam a passagem de íons cálcio e sódio e, conseqüentemente, as polarizações, despolarizações da musculatura cardíaca que ocasionam momentos de contração e relaxamento muscular [13].

2.1.3 Ciclo Cardíaco

O conjunto de eventos cardíacos que ocorre entre o início de um batimento e o início do próximo é denominado ciclo cardíaco. De maneira geral, é composto por um período de relaxamento (diástole) seguido por um período de contração (sístole) [13].

O impulso cardíaco rítmico se origina em células especializadas do nodo sinoatrial (também conhecido como nodo sinusal ou nodo S–A), localizado na junção entre a veia cava superior e o átrio direito. O impulso passa do no S–A de maneira organizada através de tratos condutores especializados nos átrios para ativar primeiro o átrio direito e depois o átrio esquerdo. A passagem do impulso é retardada no feixe A–V antes de continuar até os ventrículos [14].

A figura 2.2 mostra as estruturas responsáveis pela excitação elétrica do coração.

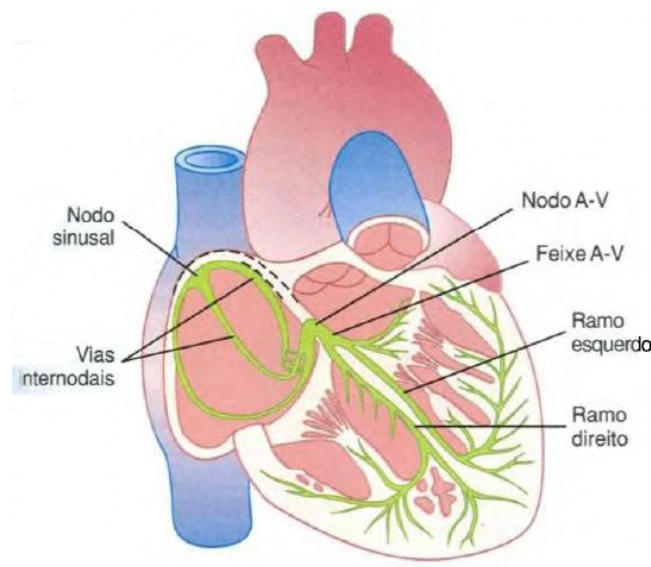


Figura 2.2 – Excitação elétrica do coração.

Fonte: Guyton e Hall (2006).

2.1.4 Eletrocardiograma normal

A ativação elétrica do coração leva à produção de correntes de ação que fluem pelo condutor torácico (considerado um meio puramente passivo, sem a presença de fontes ou drenos elétricos). Potenciais medidos na superfície externa desse meio – ou seja, a superfície do corpo – são conhecidos como eletrocardiogramas (ECG) [14].

Fatores importantes da forma de onda eletrocardiográfica são: (1) ondas individuais (P, Q, R, S e T), (2) duração das ondas, e (3) intervalos de tempo específicos (intervalos P–R, S–T e Q–T) [14].

A onda P é produzida pela despolarização atrial, o complexo QRS é produzido primariamente pela despolarização ventricular e a onda T pela repolarização ventricular. Manifestações da repolarização atrial são geralmente encobertas pelo complexo QRS. Os intervalos P–R e S–T normalmente têm potencial nulo e o intervalo P–R é causado principalmente pelo atraso de condução no feixe A–V. O segmento S–T se relaciona à duração média de regiões de platô de células ventriculares individuais [14].

Dessa forma, o eletrocardiograma é formado por ondas de despolarização e repolarização [13].

Pode-se observar um eletrocardiograma normal na figura 2.3 abaixo.

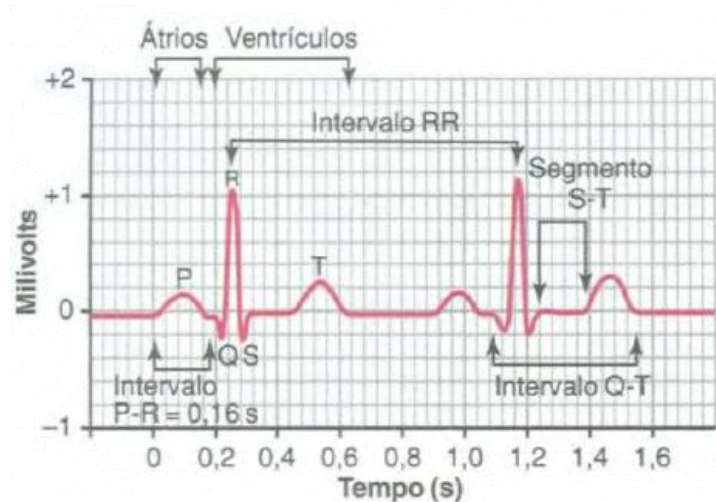


Figura 2.3 – Eletrocardiograma normal.

Fonte: Guyton e Hall (2006).

2.1.5 Derivações eletrocardiográficas

A figura 2.4 representa as conexões elétricas entre os membros do paciente e o eletrocardiógrafo para a obtenção dos sinais eletrocardiográficos provenientes das três derivações bipolares padrão. O termo *bipolar* refere-se ao fato de os eletrodos serem colocados em lados opostos do coração. Dessa maneira, uma derivação não é apenas um fio conectado ao corpo, mas o conjunto de dois fios e seus eletrodos conectados ao corpo de maneira a fechar um circuito completo entre corpo e eletrocardiógrafo.

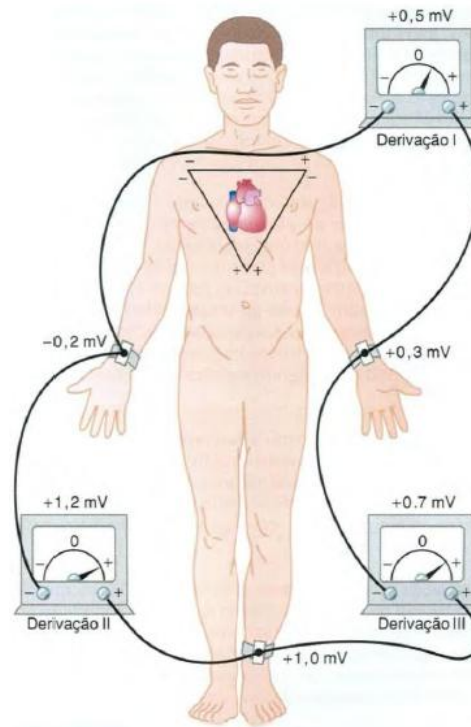


Figura 2.4 – Ligações entre eletrocardiógrafo e corpo humano para as três derivações bipolares padrão.

Fonte: Guyton e Hall (2006).

Por apresentarem disposições de eletrodos diferentes, os sinais eletrocardiográficos captados por cada uma das derivações são diferentes. A figura 2.5 ilustra as diferenças entre os sinais eletrocardiográficos oriundos das diferentes derivações padrão.



Figura 2.5 – Sinais eletrocardiográficos das três derivações padrão.

Fonte: Guyton e Hall (2006).

2.1.6 Determinação da frequência cardíaca

A frequência cardíaca pode ser determinada conhecendo-se a quantidade de complexos QRS que ocorrem em certo período de tempo, como mostra a equação (1):

$$f_c = \frac{n_{QRS}}{T} \quad (2.1)$$

Onde n_{QRS} indica a quantidade de complexos QRS detectados no período de tempo T e o valor de frequência cardíaca obtido é denotado por f_c .

2.2 Introdução à programação orientada a objetos

Esta seção busca introduzir conceitos básicos referentes à programação orientada a objetos para que o leitor possa compreender questões estruturais do programa apresentado.

2.2.1 Visão Geral

No mundo real, estamos cercados de objetos, sejam eles animados ou inanimados. Em consequência disso, o ser humano tende a pensar em termos de objetos como árvores, casas, aviões etc. Outra parte importante desse raciocínio são os atributos de cada um desses objetos (por exemplo, uma bola rola, salta, incha, esvazia; um bebê chora, dorme, pisca; um carro acelera, freia, muda de direção). A partir desse raciocínio, o ser humano é capaz de traçar relações entre múltiplos objetos como bebês e adultos, carros e caminhões, entre outros [15].

A *programação orientada a objetos (OOP, object-oriented programming)* modela objetos do mundo real com duplicatas em *software*. Ela se aproveita das relações de *classe*, nas quais objetos de uma mesma classe – tal qual uma classe de veículos – têm as mesmas características. Também tira proveito de relações de *herança*, nas quais classes de objetos recém-criadas são derivadas e absorvem características de classes já existentes, além de adicionar características suas. Um objeto da classe conversível certamente tem características da classe mais genérica automóvel, mas apenas a capota da classe conversível é capaz de subir e descer [15].

A programação orientada a objetos apresenta uma maneira mais intuitiva para a visualização do processo de programação, modelando objetos reais, seus atributos e

comportamentos. Também modela a comunicação entre objetos, que é feita através de mensagens [15].

Por esse motivo, a programação orientada a objetos facilita a reutilização e manutenção de código, e acaba proporcionando boas bases para o desenvolvimento de aplicações que envolvam análises complexas de dados [15, 16].

Buscando aproveitar as vantagens desse modelo, optou-se pelo desenvolvimento da IHM em uma linguagem orientada a objetos. Dessa maneira, vale apresentar ao leitor uma revisão dos conceitos básicos deste paradigma de programação.

2.2.2 Conceitos básicos

Classes e objetos: Classes são como a planta de casa, ou seja, contém os atributos de uma casa, mas não passam de uma abstração. Entretanto, com o auxílio de uma planta (classe), um construtor pode construir uma casa real (objeto). Dessa forma, objetos são instâncias de classes e carregam seus atributos (dados, propriedades, métodos) [15].

Encapsulamento: A OOP encapsula dados (atributos) e métodos (comportamento) em pacotes conhecidos como classes [15]. Essas informações não são visíveis a outros objetos, deixando os detalhes de implementação ocultos dentro do objeto em questão. As vantagens do encapsulamento incluem modularidade e isolamento de partes do código [16]. Pensando novamente na analogia apresentada há pouco, certamente é possível dirigir um carro sem ser especialista no funcionamento de motores, transmissões e escapamentos [15].

Herança: É uma forma de reutilização de *software* na qual novas classes são criadas a partir de classes existentes, absorvendo seus atributos e comportamentos e as adequando para os recursos que as novas classes necessitam [15, 16]. A figura 2.6 a seguir apresenta um exemplo simples de herança.

Polimorfismo: O polimorfismo permite o projeto e implementação de sistemas facilmente extensíveis. Os programas podem ser escritos para processar genericamente objetos de todas as classes pertencentes a uma hierarquia [15].

Método: O nome para procedimentos e funções pertencentes a um objeto. Métodos são chamados de funções-membro em C++ [16].

Propriedade: Uma entidade que atua como um acesso para os dados e o código contidos em um objeto. *Propriedades* preservam o usuário final dos detalhes de implementação de um objeto [16].

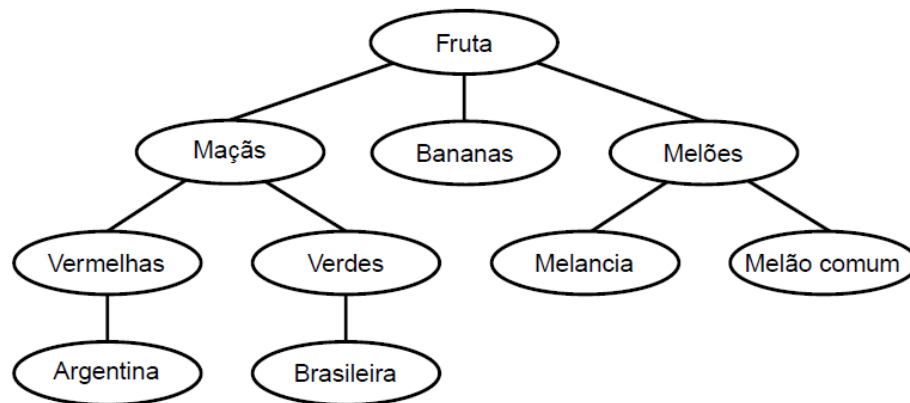


Figura 2.6 – Exemplo de herança.

Fonte: Teixeira e Pacheco (2000).

2.2.3 Diagrama de classes

É possível modelar as classes de um sistema e seus respectivos relacionamentos a través de *diagramas de classes*.

Cada classe é modelada por um retângulo que pode ser dividido em três partes, sendo que: (1) a parte superior contém o nome da classe; (2) a parte intermediária contém os atributos da classe; (3) a parte inferior contém os métodos da classe [15]. A figura 2.7 representa tal modelo de classes para o caso de um elevador em um prédio de dois andares, melhor explicado um pouco mais adiante.

Nesta representação, percebemos que o elevador carrega informações sobre sua posição, direção, capacidade, tempo gasto para chegada, se está em movimento (atributos). Além disso, também percebemos as ações relacionadas à operação do elevador, como passagem de tempo, entrada e saída de pessoas, chamada do elevador e sua saída (métodos). Aplicando raciocínio análogo, se pode entender o comportamento de todos os objetos do sistema exemplo [15].

Quando é desejável denotar os relacionamentos entre classes, o sistema de três divisões é suprimido com o objetivo de deixar o diagrama mais legível. Nesse diagrama, linhas cheias representam associações, que podem receber nomes, entre classes. Os números junto às linhas denotam multiplicidade, indicando quantos objetos de uma classe participam da associação. Losangos cheios presos às linhas de associação indicam relação de

composição. Setas vazadas presas às linhas de associação denotam relações de herança. A figura 2.8 a seguir exemplifica essa representação [15].



Figura 2.7 – Diagrama detalhando cada classe do sistema exemplo.

Fonte: Modificado de Deitel e Deitel (2001).

Pelo diagrama, é possível notar que, no sistema exemplo, o prédio possui dois pisos, um elevador, um relógio e uma agenda; o elevador possui uma campainha, um botão e uma porta; cada piso possui iluminação e botão próprios. Também se pode perceber que a agenda cuida de verificar para qual piso o elevador deve se dirigir em seguida, e, que uma pessoa pode estar no elevador ou em um dos pisos. Por fim, se conclui que o botão do elevador e o botão do piso são descendentes da classe mais genérica botão [15].

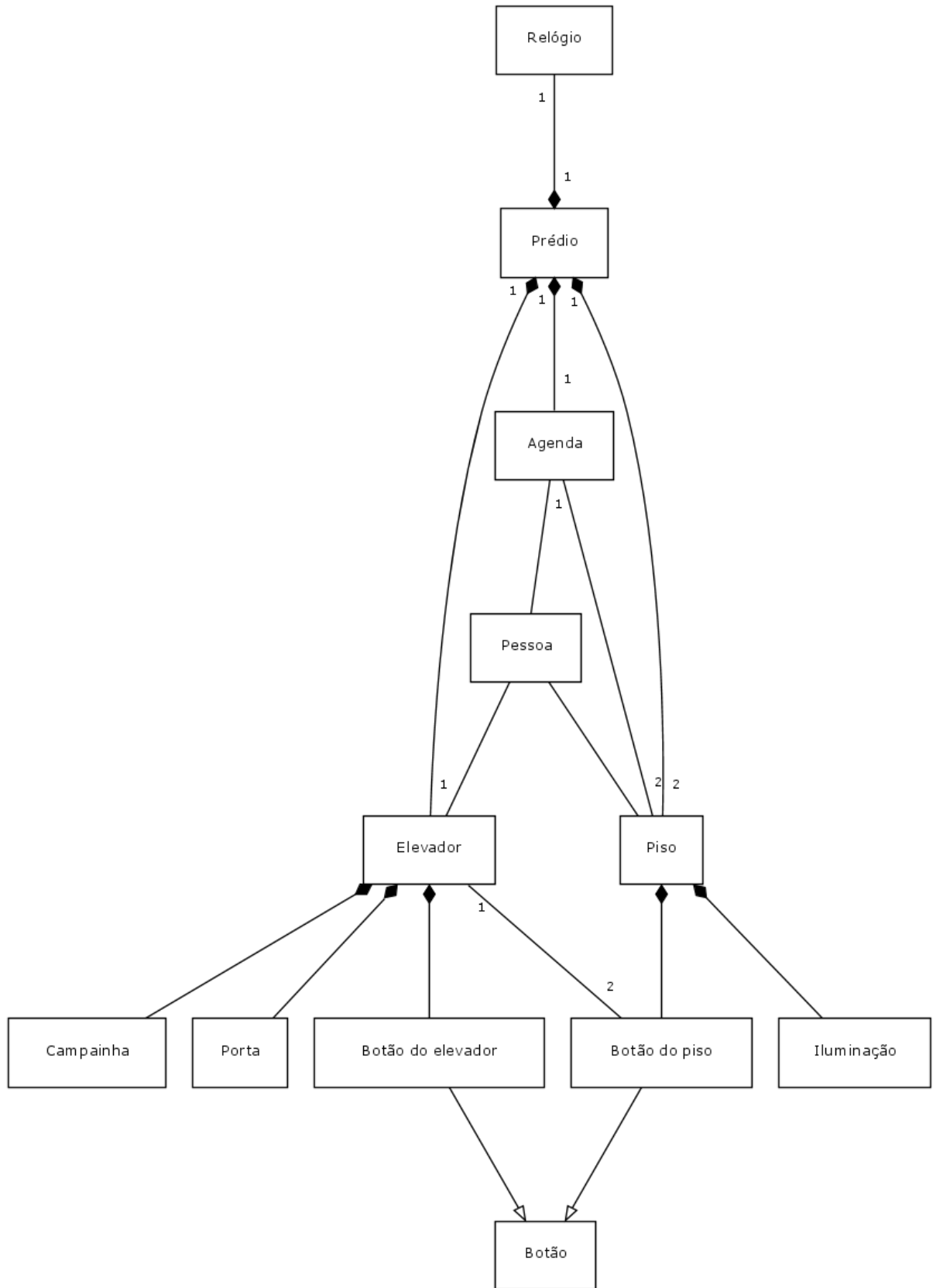


Figura 2.8 – Diagrama mostrando as relações entre as classes do sistema exemplo.

Fonte: Modificado de Deitel e Deitel (2001).

2.2.4 Multi-threading

Thread é um objeto do sistema operacional que representa um caminho de execução de código dentro de um determinado processo [16].

Multi-threading refere-se à possibilidade de escolha de múltiplos *threads* (ou caminhos) de execução em uma aplicação. Cada aplicação tem, no mínimo, um *thread* (*thread* default ou *thread* principal), mas pode criar outros threads para realizar outras tarefas [16].

Os threads permitem que diversas rotinas de código sejam executadas “simultaneamente”. Entretanto, a execução simultânea só ocorre, de fato, se houver vários núcleos de processamento na máquina que esteja executando a aplicação. De qualquer maneira, o sistema operacional emite “fatias” de tempo para cada aplicação. Na verdade, o que acontece é a comutação de tarefas em curtos intervalos de tempo, de modo a gerar a impressão de que várias tarefas estejam sendo executadas ao mesmo tempo [16].

A adoção desse método previne situações como aquelas em que o sistema parecer ter travado quando na verdade apenas entrou em um loop de operações mais longo [16].

3 Metodologia e revisão bibliográfica

3.1 Considerações iniciais

Este capítulo tem o objetivo de expor os requisitos, desafios e escolhas encontrados nas diferentes etapas do projeto da interface homem-máquina abordada nesse trabalho.

3.2 Comunicação USB

O protocolo USB (*Universal Serial Bus*) é uma solução adequada sempre que se deseja realizar a comunicação entre um computador e um dispositivo externo. Tal interface serve bem tanto a dispositivos produzidos em massa quanto a dispositivos especializados feitos sob-medida [17].

A utilização do protocolo USB para a realização da comunicação entre os módulos de circuitos eletrônicos e o computador pessoal apresenta aspectos vantajosos, tanto do ponto de vista de implementação quanto do ponto de vista de utilização. Alguns desses aspectos são [17]:

- A mesma interface serve a múltiplos dispositivos.
- A configuração ocorre de maneira automática, uma vez que o sistema operacional reconhece o dispositivo conectado. Talvez seja necessária a instalação de *driver* específico na primeira vez em que o dispositivo seja conectado, mas isso não ocorre nas vezes seguintes.
- Há grande facilidade de conexão, pois um computador típico possui várias portas USB.
- Cabos compactos comparados a outros padrões como RS-232. Há especificações para garantir que os cabos USB funcionem de maneira confiável.
- Possibilidade de dispositivos sem fio que se comuniquem com o computador através do protocolo USB.

- Usuários podem conectar e desconectar os dispositivos quando lhes convier.
- Alimentação de dispositivos que necessitem de tensões inferiores a 5 V e correntes inferiores a 500 mA, para o padrão USB 2.0, e 900 mA para o padrão USB 3.0.

USB é confiável, devido a suas especificações de *drivers*, *receivers* e cabos garantirem uma interface com baixo ruído elétrico que poderia causar erros nos dados. Os protocolos USB permitem a detecção de erros nos dados recebidos e notificam a necessidade de retransmissão. Com isso, desenvolvedores não precisam se preocupar com rotinas de checagem de erros na comunicação USB [17].

Dispositivos USB também apresentam baixo custo, devido ao *host* USB do computador realizar a maior parte do controle da interface, e baixo consumo, graças a circuitos e protocolos desenvolvidos com este objetivo em mente [17].

Em termos velocidade, a interface USB pode trabalhar com transmissões de dados de até 480 Mbps para o padrão USB 2.0 e 5 Gbps para o padrão USB 3.0, garantindo largura de banda para aplicações que não podem ter atrasos na transmissão de dados [17].

As características da interface USB apresentadas nesta seção justificam sua escolha como interface entre os dispositivos da rede de sensores para monitorização de sinais fisiológicos e o programa de computador que desempenha a função de interface homem-máquina no sistema proposto.

3.3 Decodificação do protocolo

Muitas aplicações, inclusive as de áreas correlatas à medicina, devem minimizar ao máximo a tolerância a erros de comunicação, tais quais erros ou perdas de dados. Além disso, sistemas de monitorização podem ter sua capacidade limitada pela largura de banda disponível, algo que pode ser melhorado com um controle eficiente do fluxo de dados [18].

Levando em conta também as características e objetivos do sistema proposto (monitorização multiparamétrica de sinais fisiológicos de múltiplos usuários), ficou constatada a necessidade de elaboração de protocolo para encapsulamento de dados e checagem de erros.

Vale lembrar que a concepção e elaboração de tal protocolo não foram objetivo desse trabalho, conforme explicitado na última seção do primeiro capítulo.

Entretanto, para que a aplicação final seja capaz de tratar e exibir os dados adequadamente, ela precisa, primeiro, ser capaz de receber e desencapsular os dados com sucesso.

Sob essa perspectiva, a interface homem-máquina apresentada nesse trabalho realiza o desencapsulamento dos dados de acordo com as especificações de protocolo recebidas e assume que tal protocolo esteja funcionando corretamente.

3.4 Estrutura dos buffers

Uma vez que os dados tenham sido recebidos e desencapsulados, necessitam ser armazenados em buffers adequados para que possam ser utilizados para as etapas subsequentes de processamento realizadas pelo programa final.

A primeira consideração a ser feita relaciona-se ao fato de que sistema propõe a monitorização de diversos sinais físicos e fisiológicos. A princípio, pensou-se em um sistema capaz de monitorar sinais de eletrocardiografia, temperatura corpórea, múltiplos sinais de acelerometria e frequência cardíaca (calculada a partir do sinal eletrocardiográfico) para cada usuário. Além disso, também se definiu o objetivo de caracterização das condições do ambiente, através a coleta de dados de temperatura ambiente e umidade relativa do ar.

Desse ponto de vista, é intuitiva a adoção de um buffer para armazenar os dados amostrados referentes a cada uma dessas grandezas.

A segunda consideração importante trata da multiplicidade de usuários envolvidos na rede de monitorização. Tal multiplicidade se estende aos sinais coletados dos usuários, ou seja, sinais de eletrocardiografia, temperatura, frequência cardíaca e quantos forem os sinais de acelerometria coletados por usuário.

Por exemplo, suponha uma situação onde haja três usuários conectados à rede, cada um com dois acelerômetros posicionados em locais de interesse. Nesse caso, haverá a coleta e

transmissão de três sinais de eletrocardiografia, três sinais de temperatura corpórea e seis sinais de acelerometria. Haverá, ainda, o cálculo de três frequências cardíacas.

Para lidar com esta situação, foi pensada uma estrutura tridimensional para cada buffer de sinal coletado dos usuários, ou seja, um buffer tridimensional para sinais eletrocardiográficos, um buffer tridimensional para sinais de temperatura corpórea, um buffer tridimensional para as frequências cardíacas calculadas e um buffer tridimensional para cada sinal de acelerometria. Dessa maneira, cada buffer faz referência a: (1) o usuário a quem pertence o sinal medido; (2) o número da amostra ou instante discreto em que foi realizada a medida; (3) o valor medido da grandeza em determinado instante.

3.5 Exibição de múltiplas grandezas para múltiplos usuários

Como se trata de um sistema cujo objetivo principal é a monitorização de grandezas fisiológicas dos usuários, se faz necessária a exibição gráfica da evolução temporal dos sinais medidos.

Visando obtenção de uma interface amigável e de fácil de utilização, a exibição dessas grandezas foi pensada de modo a agrupar curvas que representam a evolução de um mesmo sinal para diferentes usuários e, ainda, permitir a exibição seletiva de tais curvas. Em outras palavras, todas as curvas referentes a eletrocardiogramas devem ser traçadas no mesmo gráfico e permitir ao operador do programa a visualização individual de cada curva ou a visualização simultânea de duas ou mais curvas, de acordo com o que julgar melhor.

O programa também deve permitir que a seletividade da exibição de curvas esteja ativa e responda aos comandos do operador durante todo o período de traçado. Ou seja, o operador deve poder alternar a visualização entre as diversas curvas disponíveis quando bem entender.

Essa abordagem deve ser aplicada a todos os grupos de curvas que representam a mesma grandeza física ou fisiológica para usuários diferentes. Dessa forma, as evoluções temporais dos sinais de temperatura corpórea também devem ficar juntas, assim como os diferentes sinais de acelerometria e as frequências cardíacas calculadas para os múltiplos usuários.

3.6 Processamento adicional

Apesar de os circuitos desenvolvidos para este sistema já envolverem etapas de pré-processamento, algum processamento além do traçado gráfico ainda precisa ser realizado pelo programa de computador.

Esta seção busca esclarecer os motivos por trás desse processamento adicional.

3.6.1 Filtro digital

Filtros digitais exercem a mesma função que filtros analógicos, porém, ao invés de serem implementados através de circuitos eletrônicos, são implementados através de equações recursivas em programas computacionais. A natureza dessa implementação gera algumas vantagens para os filtros digitais sobre os analógicos, como [19]:

- Alta imunidade a ruídos;
- Sua precisão depende apenas de arredondamentos matemáticos;
- A alteração de características operacionais do filtro (frequências de corte) costuma ser simples e barata;
- Diferentemente de filtros analógicos, o desempenho de filtros digitais não é dependente de fatores como envelhecimento de componentes eletrônicos, variação de temperatura e tensão de alimentação. Esta característica é de especial importância para aplicações médicas, onde muitos sinais possuem baixas frequências que podem ser distorcidas devido a deriva (*drift*) em circuitos analógicos.

Para a aplicação proposta, ruído proveniente da rede elétrica é uma grande fonte de interferência [19]. Com o objetivo de minimizar o efeito de possíveis ruídos provenientes da rede elétrica, foi decidido que seria conveniente a implementação de um filtro digital com o objetivo de rejeitar ruídos com frequência de 60 [Hz] (filtro *notch* centralizado em 60 [Hz]). Pelo seu potencial de apresentar bandas de corte estreitas, decidiu-se pela utilização de um filtro IIR (*Infinite Impulse Response*) [19].

3.6.2 Algoritmo *Turning Point*

O programa proposto deve realizar múltiplas tarefas simultaneamente, sendo algumas delas: receber dados que chegam pela porta USB, decodificar e realocar esses dados, realizar cálculos e realizar múltiplos traçados gráficos. Dependendo da capacidade de processamento da máquina que estiver sendo utilizada para executar o programa, isso pode significar uma sobrecarga e comprometer o desempenho do programa.

Com o objetivo de aliviar essa situação, optou-se por implementar algoritmo *Turning Point* que reduz o esforço computacional através da redução de pontos a serem traçados em um gráfico [19]. Esse algoritmo poderá ser aplicado a todos os gráficos traçados pelo programa e será explicado no próximo capítulo.

Esse algoritmo pode ser aplicado em cascata e traz a resposta gráfica do programa cada vez mais próxima de um traçado em tempo real.

3.6.3 Detecção de complexo QRS e cálculo de frequência cardíaca

A frequência cardíaca também é um sinal de interesse para a monitorização do sistema proposto, entretanto, não é utilizado nenhum método de medição de frequência cardíaca. Ao invés disso, optou-se por calcular a frequência cardíaca a partir do sinal eletrocardiográfico.

Para que isso seja possível, se faz necessária a detecção eficiente e confiável dos complexos QRS presentes no eletrocardiograma [19].

Em [20], os autores apresentam uma solução para esta questão que reduz consideravelmente o número de falsos positivos e falsos negativos, tornando esta solução bastante confiável. Este método será discutido em mais detalhes no capítulo seguinte.

3.6.4 Linearização de temperatura corpórea

Decidiu-se pelo sensor MA300 da GE Thermometrics para medição de temperatura corpórea. Trata-se de um termistor de coeficiente de temperatura negativo (NTC, *Negative Temperature Coefficient*) cuja resposta não é totalmente linear [21].

O sensor será usado para medir uma faixa reduzida de temperaturas, visto que a temperatura do corpo humano pode variar em torno de 36°C a 40°C, dependendo das condições do ambiente e de intensidade de exercícios físicos [13].

Portanto, para melhorar a resposta do sensor para esta aplicação, julgou-se adequada a linearização da resposta fornecida pelo mesmo. Tal linearização é feita a partir de uma reta delimitada pelos valores máximo e mínimo de temperatura de interesse e os valores de tensão correspondentes registrados pelo sensor.

Entretanto, com o objetivo de diminuir o volume de processamento dos microcontroladores embarcados nas placas de circuitos projetados para o sistema, optou-se pela realização da linearização dos sinais de temperatura no programa final. Por este motivo, a linearização não será feita a partir de tensões analógicas, mas sim de seus correspondentes valores digitais.

O processo de linearização será explicado em mais detalhes no próximo capítulo.

3.7 Interface visual

Segundo [22], alguns princípios de organização visual devem ser levados em consideração durante a elaboração de uma interface homem-máquina de modo a melhorar a compreensão do ambiente virtual e facilitar sua utilização, engrandecendo, assim, a experiência do usuário do *software*.

Alguns desses princípios são mostrados a seguir.

Adoção de elementos visuais: A utilização de elementos gráficos do *design* visual da IHM facilita a compreensão da mesma. Estes elementos visuais devem ser compactos e

genéricos, permitindo que o usuário os relacione com experiências anteriores, bem como seja capaz de distinguir diferentes funções do programa utilizado.

Evitar desordem visual: O *design* visual da IHM deve minimizar a utilização de elementos visuais, mantendo funções importantes para o usuário e fazendo uso de formas geométricas simples e cores pouco saturadas, por exemplo, para que o usuário tenha uma experiência confortável e não seja distraído pelo excesso de elementos visuais na IHM.

Distinção entre plano de fundo e elementos principais: Pessoas tendem a encarar elementos luminosos como sendo principais e, em contrapartida, elementos apagados como plano de fundo. Deve-se tomar cuidado para que não seja dada ênfase ao plano de fundo e para que elementos importantes não fiquem imperceptíveis. Isso pode ser obtido através da adoção de proporções, disposição espacial e escolha de cores adequadas.

Organização espacial: Deve ser feita a divisão das diferentes áreas da interface visual e planejamento dos elementos visuais e integração do ambiente, para transmitir ao usuário a sensação de unidade. Espaços vazios também devem existir para aumentar o conforto do usuário.

Adoção de cores: Cores são elementos importantes para guiar e influenciar a experiência do usuário. A adoção apropriada de cores pode ajudar o usuário a encontrar informações relevantes, atrair atenção, aumentar a velocidade de navegação, demonstrar a relação entre conteúdos etc.

Layout: Uma interface visual bem estruturada depende de um bom *layout*. Elementos de funções similares devem ser agrupados, ícones de formas geométricas similares podem ser agrupados de maneira horizontal, vertical etc.

3.8 Estrutura do *software*

O primeiro passo para se definir a estrutura do programa consiste da identificação das classes básicas que são necessárias para que o mesmo cumpra sua função [15].

Para proporcionar a interação com o usuário, são necessários objetos visuais de classes básicas, como botões, menus etc. A quantidade e disposição destes objetos depende de como a interface é organizada visualmente, como discutido na seção anterior.

Além disso, são necessários objetos de classes não visíveis para desempenhar funções de cálculo e gerenciamento do programa.

Durante a formulação do projeto, foram identificadas algumas classes importantes que são mostradas a seguir.

Interface visual: Janela principal do programa e meio de interação entre operador e máquina.

Usuário: Armazena identificação e cor associada aos dados e traçados gráficos de um usuário.

Thread de controle: O objetivo deste *thread* é gerenciar o tempo dedicado para cada trecho de código executado pelo programa, para que este possa desempenhar múltiplas tarefas simultaneamente e continuar respondendo aos comandos do usuário, conforme discutido anteriormente.

Driver: Realiza a interface entre a aplicação de alto nível e os *drivers* de mais baixo nível, possibilitando a comunicação através do protocolo USB.

Comunicação: Cuida dos aspectos referentes à comunicação USB.

Protocolo: Esta classe visa o desencapsulamento dos dados e tomada de decisões adequadas de acordo com as informações do cabeçalho de cada pacote.

Processamento: Tem por objetivo a realização das rotinas de processamento adicional discutidas na seção 3.6.

Gráfico: Esta classe deve cuidar de todos os aspectos necessários à geração e traçado dos múltiplos gráficos tratados pelo programa.

Arquivo: Trata dos métodos responsáveis pelo tratamento de arquivos. Buscando a utilização de arquivos compactos e de fácil tratamento, optou-se pela geração de arquivos contendo os dados binários recebidos através da porta de comunicação USB.

4 Resultados

4.1 Considerações iniciais

O objetivo deste capítulo é apresentar, de forma prática, como foram implementados os pontos discutidos no capítulo anterior.

O primeiro aspecto de importância prática a ser ressaltado é a escolha da linguagem de programação. Dentre várias opções disponíveis, optou-se pelo desenvolvimento do *software* no ambiente *Delphi*, que utiliza a linguagem *Object-Pascal*. Essa escolha deve-se à ótima combinação entre programação orientada a objetos e programação visual proporcionada pelo ambiente [23].

A versão utilizada foi a *Embarcadero Delphi 2010*.

A seguir, são apresentadas as implementações dos diferentes estágios do projeto da interface homem-máquina.

4.2 Comunicação USB

Com o objetivo de simplificar e agilizar o projeto do sistema de monitorização proposto, foi adotada a solução USB integrada oferecida pela empresa FTDI Ltd. O circuito integrado FT232RL, com o auxílio de *drivers* disponibilizados pelo fabricante, permite a realização de uma interface de comunicação entre USB e UART serial, simplificando significativamente os projetos de *hardware* e *software*, e aumentando a velocidade de desenvolvimento do projeto como um todo [24].

Nesta seção, a utilização de tal ferramenta será abordada do ponto de vista do desenvolvimento da aplicação de *software*.

O fabricante oferece duas maneiras para a realização da interface USB-UART [25]:

- Um *driver* que leva o sistema operacional a perceber a porta USB como uma porta serial virtual (VCP, *Virtual COM port*) e permite que a aplicação utilize

APIs (*Application Programming Interface*) do próprio sistema para realizar operações com a porta;

- Um *driver* proprietário (D2XX) com funções voltadas à comunicação USB que permite que a aplicação realize operações na porta a partir de comandos específicos para este *driver*;

A figura 4.1 mostra a arquitetura de *drivers* disponibilizados pelo fabricante.

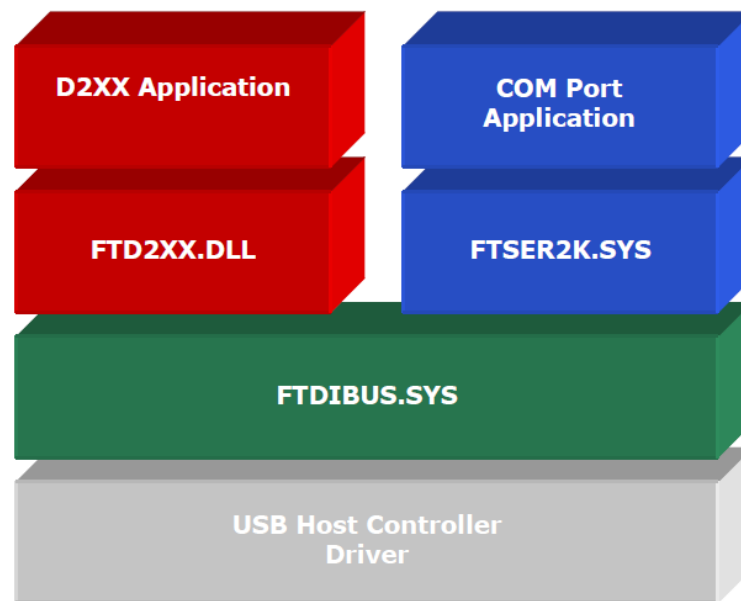


Figura 4.1 – Arquitetura de *drivers* D2XX.

Fonte: FTDI Chip (2012).

Faz-se necessária, ainda, a implementação de código capaz de desempenhar o papel de interface entre a aplicação de alto nível e os *drivers* de baixo nível.

Com o objetivo de preservar as características de comunicação possibilitadas pelo protocolo USB, foi escolhida a implementação através do *driver* D2XX.

Finalmente, implementou-se no programa uma classe voltada ao tratamento dos eventos relacionados à comunicação.

4.3 Decodificação do protocolo

Conforme mencionado no capítulo anterior, o desenvolvimento do protocolo de comunicação não foi objetivo deste trabalho. Entretanto, é necessário que o programa aqui apresentado seja capaz de desencapsular os dados recebidos através da porta USB.

Para que o leitor possa entender como isso foi feito, é necessário que ele seja apresentado à estrutura básica do protocolo especificada até o momento.

A figura 4.2 mostra um pacote, ou *frame*, genérico do protocolo.

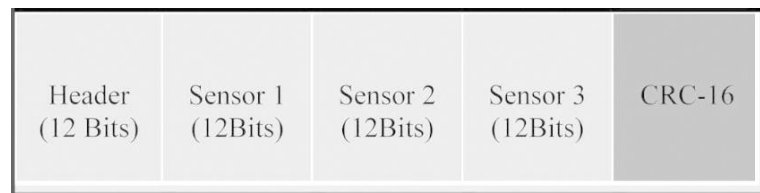


Figura 4.2 – Pacote, ou *frame*, genérico do protocolo.

Cada pacote possui 64 bits, divididos entre cabeçalho (12 bits), dados amostrados de três sensores (12 bits para cada) e 16 bits voltados para checagem de erro.

Os 12 bits do cabeçalho incluem as seguintes informações:

- Versão do protocolo – 2 bits (b0 e b1);
- Tipo de pacote – 3 bits (b2 a b4);
- Identificação (ID) do usuário – 7 bits (b5 a b11).

A versão do protocolo prevê a possibilidade de atualizações no protocolo que demandem tratamento diferente, permitindo retrocompatibilidade. Os tipos de pacote denotam as origens dos dados contidos no pacote (ECG, acelerômetro, temperatura, frequência cardíaca). Já a ID de usuário permite que se saiba de qual usuário foram feitas as leituras dos dados contidos no pacote.

Uma vez conhecida a estrutura dos pacotes do protocolo, foi criada uma sequência de operações com o propósito compreender as informações contidas no cabeçalho. Essas operações consistem de máscaras lógicas e rotações de bits que visam isolar e possibilitar a interpretação das diferentes informações do cabeçalho. Uma vez conhecidos os valores referentes à versão do protocolo, ao tipo do pacote e à identificação do usuário, pode-se iniciar tomadas de decisões que levem ao processamento adequado dos dados de cada pacote.

Por fim, realiza-se a checagem de erros de transmissão através do método CRC (*Cyclic Redundancy Check*).

O CRC é um dos métodos mais comuns e poderosos para checagem de erros de transmissão de dados. Dado um bloco de bits, ou mensagem, de tamanho k , o transmissor gera uma sequência de bits de tamanho $(n - k)$, conhecida como sequência de checagem de pacote (FCS, *frame checking sequence*), tal que o pacote resultante, que consiste de n bits, é divisível por um número pré-determinado. O receptor então divide o pacote recebido e, se não houver resto, assume que não houve nenhum erro de transmissão [26].

Foi implementada uma classe no programa para lidar com as questões relativas ao tratamento do protocolo.

4.4 Estrutura dos buffers

No capítulo anterior, foram abordadas as razões que explicam a estrutura tridimensional dos *buffers* de sinais coletados dos usuários.

Nesta seção, são mostradas diferentes maneiras de representar estes *buffers* para facilitar a compreensão da estrutura.

A figura 4.3 mostra os *buffers* representados como estrutura tridimensional.

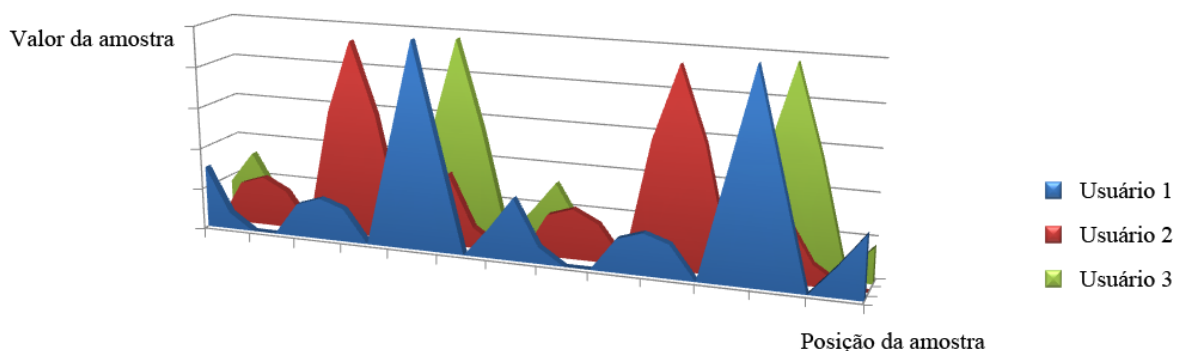


Figura 4.3 – Visualização tridimensional dos *buffers*.

Os *buffers* também podem ser vistos como vetores de matrizes, onde cada vetor contém matrizes de informações referentes à evolução temporal de um sinal de um dos usuários.

Em termos de indexação, a estrutura foi tratada da seguinte maneira:

$$\text{Valor da amostra} = \text{Buffer} [\text{usuário}, n]$$

onde *buffer* pode representar qualquer um dos *buffers* tridimensionais do sistema (ECG, acelerômetros, temperatura, frequência cardíaca) e *n* representa o número da amostra.

A linguagem Delphi não oferece tratamento nativo para estruturas tridimensionais como esta, portanto foi necessária a criação de um tipo de variável que consiste do encadeamento de estruturas matriciais, ou seja, um arranjo (*array*) de arranjos de inteiros.

4.5 Exibição de múltiplas grandezas para múltiplos usuários

A disposição dos gráficos foi definida através da utilização de abas para agrupar sinais que representam a mesma grandeza para diferentes usuários. Dessa maneira, tem-se uma aba para exibição dos sinais de ECG de todos os usuários, outra para a exibição dos sinais de temperatura corpórea de todos os usuários e assim sucessivamente. Há ainda uma aba que contempla os gráficos de temperatura ambiente e umidade relativa do ar.

A fim de permitir a exibição seletiva dos diferentes gráficos de cada grupo, optou-se pela utilização de vetores de *bitmaps* para armazenar as informações dos gráficos dos múltiplos usuários.

Bitmaps são informações binárias organizadas em um padrão de bits que representa uma imagem gráfica [16]. Por este motivo, não são visíveis a menos que recebam uma instrução para serem desenhadas. Dessa forma, é possível armazenar as informações gráficas de todos os usuários sem que haja perda quando um gráfico deixar de ser exibido. Outra vantagem dessa abordagem é que há redução do esforço computacional quando não ocorre a exibição de todos os gráficos.

Quando a exibição simultânea se faz necessária, ocorre a realização de uma máscara lógica entre todos os *bitmaps* que devem ser exibidos, gerando um novo *bitmap* que contém todos os gráficos selecionados para exibição. Em seguida, esse *bitmap* é traçado possibilitando a visualização dos gráficos selecionados.

Com o objetivo de exemplificar a exibição de curvas, foi gerado um arquivo contendo dados de ECG encapsulados pelo protocolo. Esse arquivo foi então lido pelo programa que, após realizar a decodificação do protocolo, traçou a onda eletrocardiográfica resultante. Para que também fosse possível a visualização de mais de uma curva sendo traçada, foi criado um *buffer* adicional contendo a mesma onda eletrocardiográfica, porém deslocada no tempo. Os resultados são mostrados juntamente com a técnica de exibição seletiva de curvas através das figuras 4.5 e 4.6 a seguir.

À esquerda é possível observar quais usuários estão selecionados para visualização e na área destinada ao traçado gráfico, são mostradas as respectivas curvas.

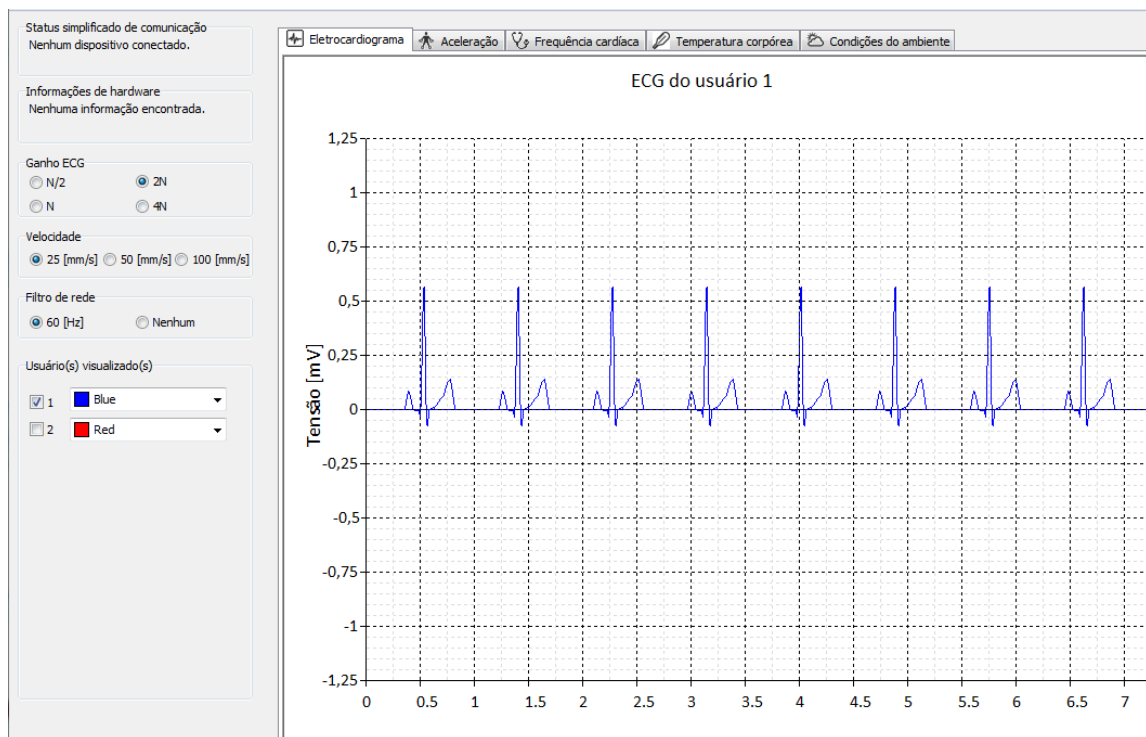


Figura 4.4 – Ondas eletrocardiográficas do primeiro usuário.

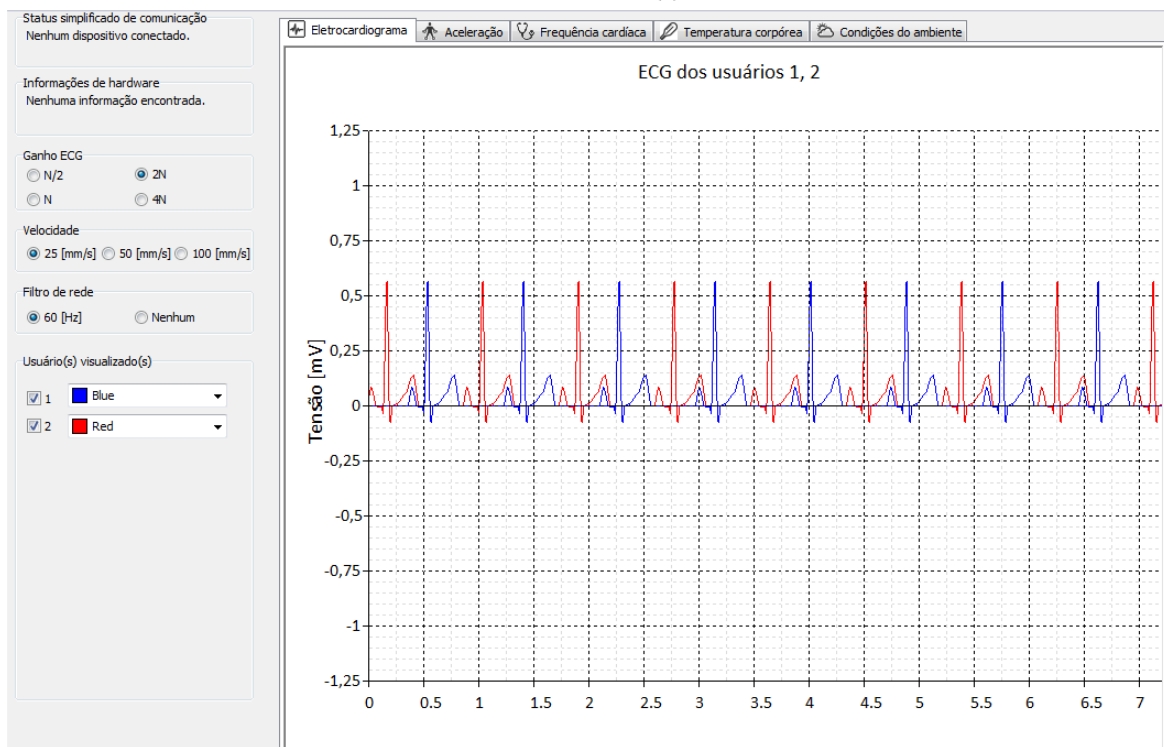
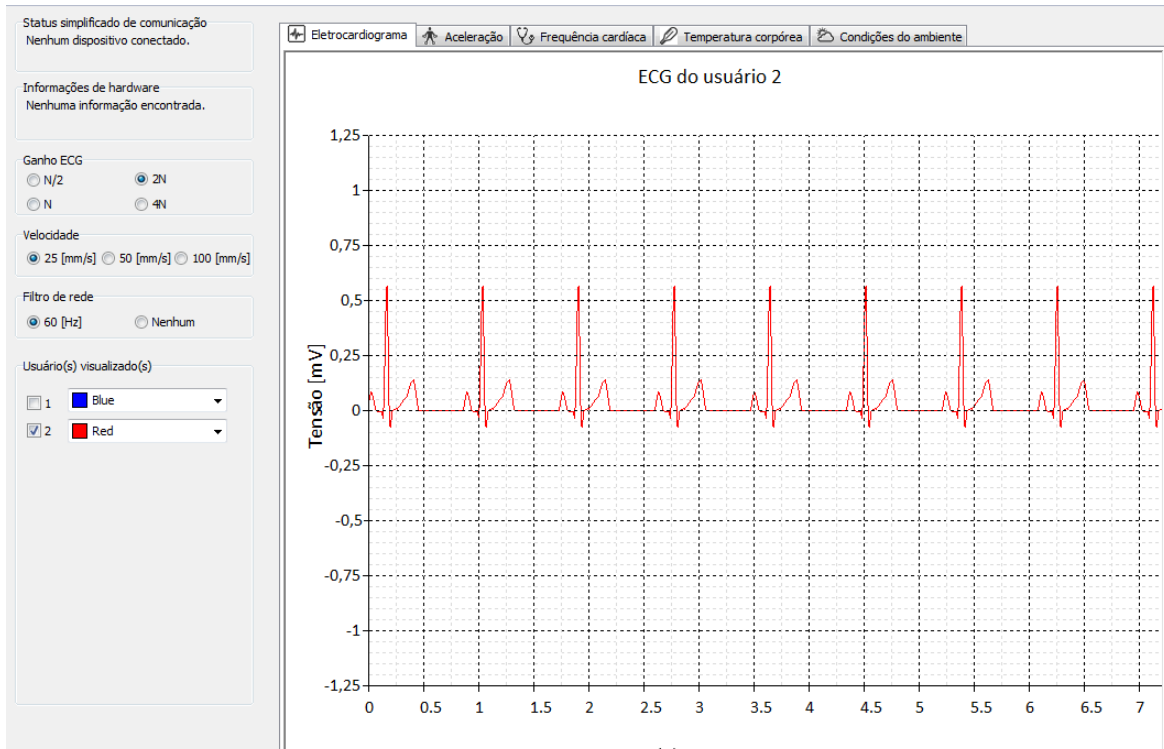


Figura 4.5 – (a) Onda eletrocardiográfica do segundo usuário; (b) ondas eletrocardiográficas de dois usuários exibidas simultaneamente.

4.6 Processamento adicional

4.6.1 Filtros digitais

O filtro apresentado a seguir foi projetado com o auxílio do software MatLab®, através da ferramenta FDATOOL. A seguir, as características do referido filtro são apresentadas.

Filtro *notch* com centro em 60 [Hz]:

O projeto do filtro resultou na seguinte equação recursiva:

$$y(n) = 0,994x(n) - 1,449x(n - 1) + 0,994x(n - 2) + 1,449y(n - 1) + 0,987y(n - 2) \quad (4.1)$$

Onde x representa valores de entrada e y representa valores de saída do filtro.

O ganho deste filtro é unitário.

A figura 4.7 apresenta as respostas de magnitude e fase do filtro.

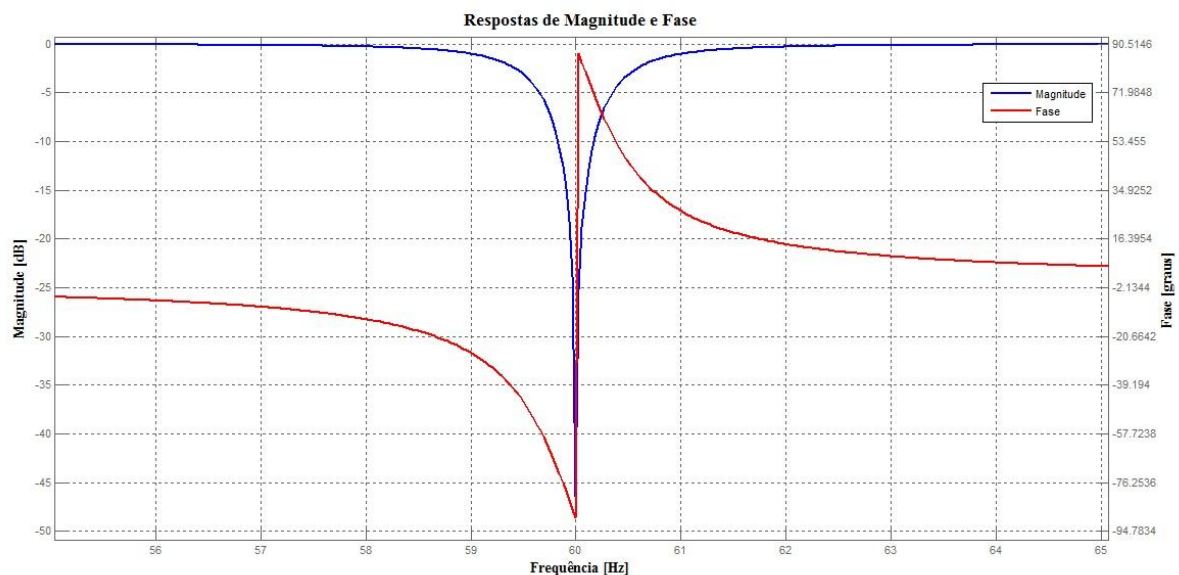


Figura 4.6 – Respostas de magnitude e fase do filtro *notch* de 60 Hz.

4.6.2 Algoritmo de *Turning Point*

O algoritmo de *Turning Point* processa três pontos por vez. O primeiro é o ponto de referência X_0 , que já foi traçado. O algoritmo então decide se deve manter X_1 ou X_2 dependendo de qual dos pontos preserva a inflexão do sinal original [19]. Em outras palavras, o algoritmo procura por pontos de máximo e mínimo e, sempre que há ocorrência destes pontos, eles são selecionados para o traçado. Caso não sejam encontrados pontos de máximo ou mínimo, o ponto mais distante é utilizado no traçado gráfico.

A figura 4.8 mostra todas as configurações possíveis para três pontos consecutivos e o ponto que deve ser escolhido em cada situação.

1		4		7	
2		5		8	
3		6		9	

Figura 4.7 – Representação das decisões do algoritmo *Turning Point* [18].

Com o objetivo de traduzir esse critério matematicamente, calcula-se a diferença entre os dois grupos de pontos consecutivos – $(X_1 - X_0)$ e $(X_2 - X_1)$ – e multiplicam-se os resultados. Se a o resultado desta multiplicação for negativo, escolhe-se X_1 . Em todos os outros casos, escolhe-se X_2 .

A figura 4.9 mostra o sinal original para que possa ser comparado com os resultados obtidos traçando-se a mesma curva com a utilização do algoritmo *Turning Point*, mostrados na figura 4.10.

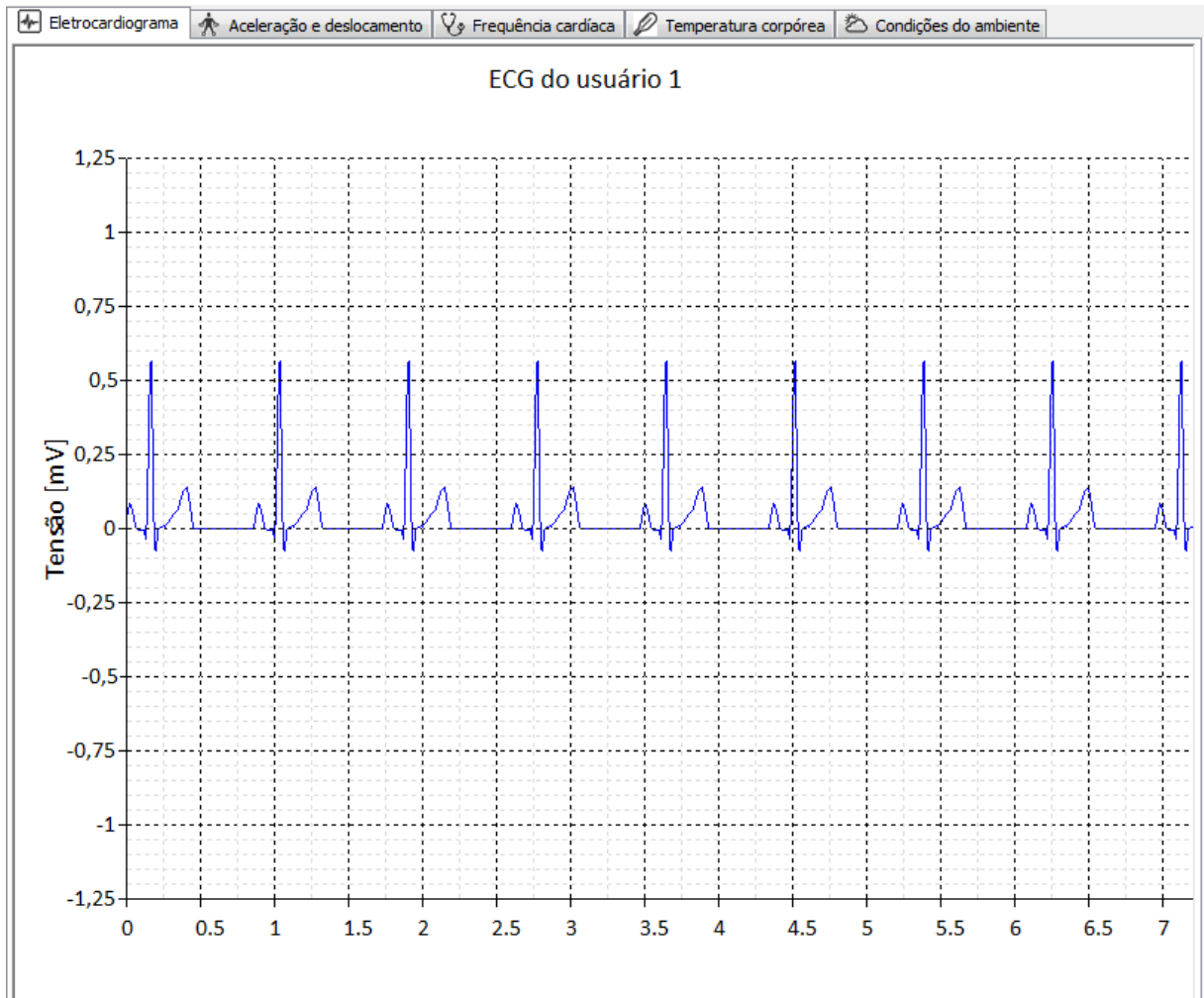


Figura 4.8 – Sinal eletrocardiográfico original.

É possível notar, especialmente quando o algoritmo é aplicado duas vezes em cascata, uma ligeira diminuição na fidelidade da curva quando comparada com a original. Isso é de se esperar, uma vez que o algoritmo *Turning Point* reduz a quantidade de pontos traçados visando a diminuição do esforço computacional. Entretanto, fica claro que o algoritmo mantém, dentro das devidas proporções, as características da forma de onda original.

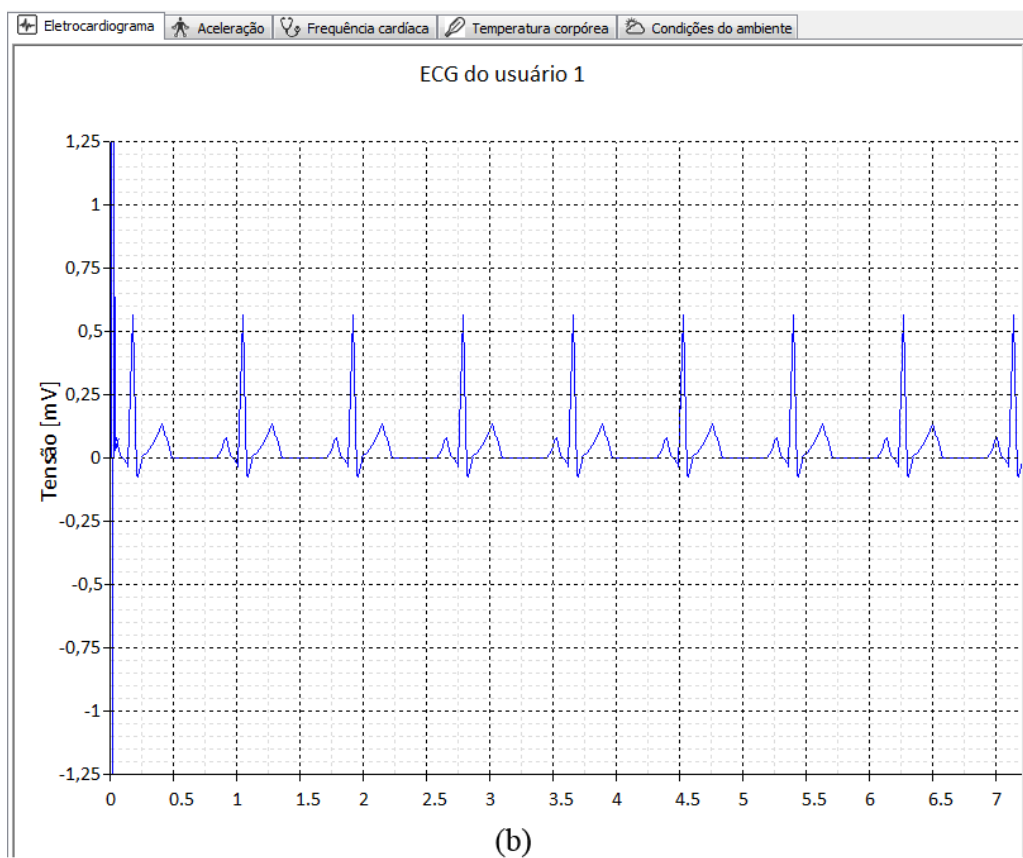
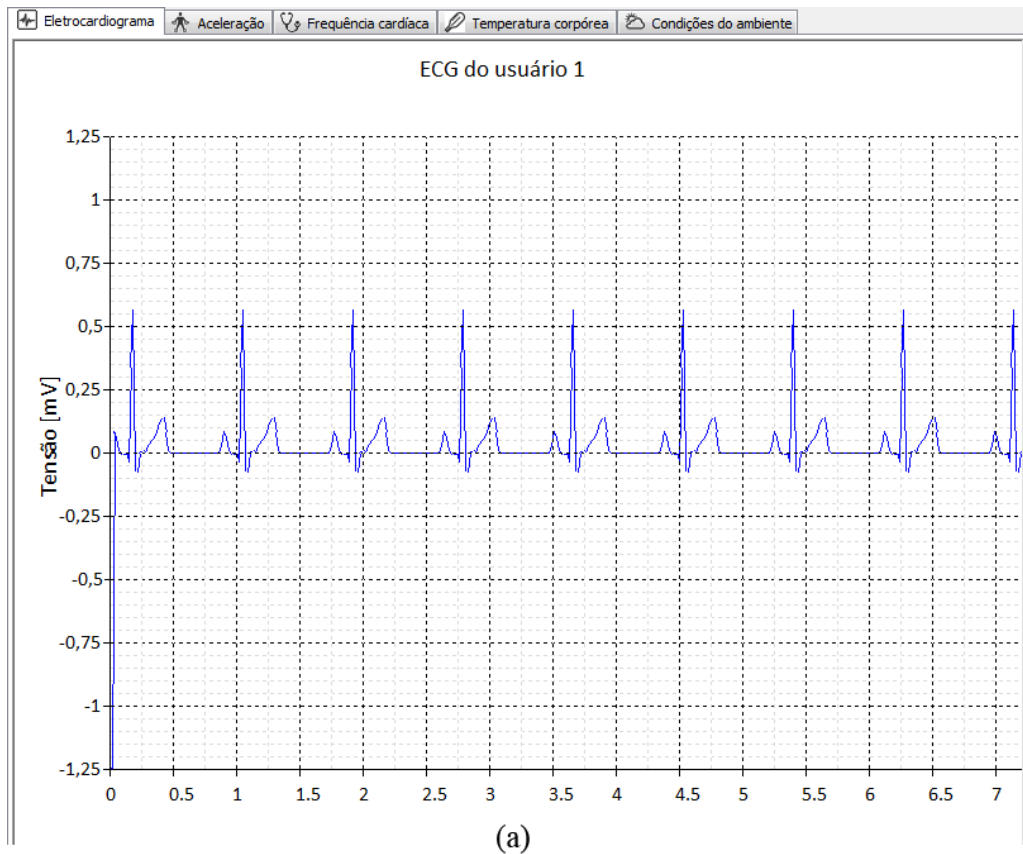


Figura 4.9 – (a) Sinal eletrocardiográfico submetido ao algoritmo *Turning Point* uma vez; (b) sinal eletrocardiográfico submetido ao algoritmo *Turning Point* duas vezes.

4.6.3 Detecção de complexo QRS e cálculo de frequência cardíaca

O algoritmo para detecção de complexo QRS descrito em [20] se baseia em análises de inclinação, amplitude e largura e é descrito a seguir.

Um filtro passa-faixa centrado em 10 [Hz] isola a energia predominante do complexo QRS, atenuando ruídos de baixa (ondas P e T, ruído de linha de base) e alta frequências (ruídos eletromiográfico e da rede elétrica). Em seguida, o sinal é diferenciado para encontrar inclinações mais fortes, típicas do complexo QRS. Após a diferenciação, o sinal é elevado ao quadrado, retificando os dados e acentuando ainda mais o resultado obtido pela diferenciação. Então, o sinal passa por um integrador de janela móvel, visando caracterizar a duração do complexo. Por fim, limites (*thresholds*) baseados nos picos de sinal e de ruído são utilizados para validar a detecção do complexo QRS e um sinal pulsado é gerado para indicar a ocorrência dos complexos QRS [19].

A figura 4.11 ilustra o processo.

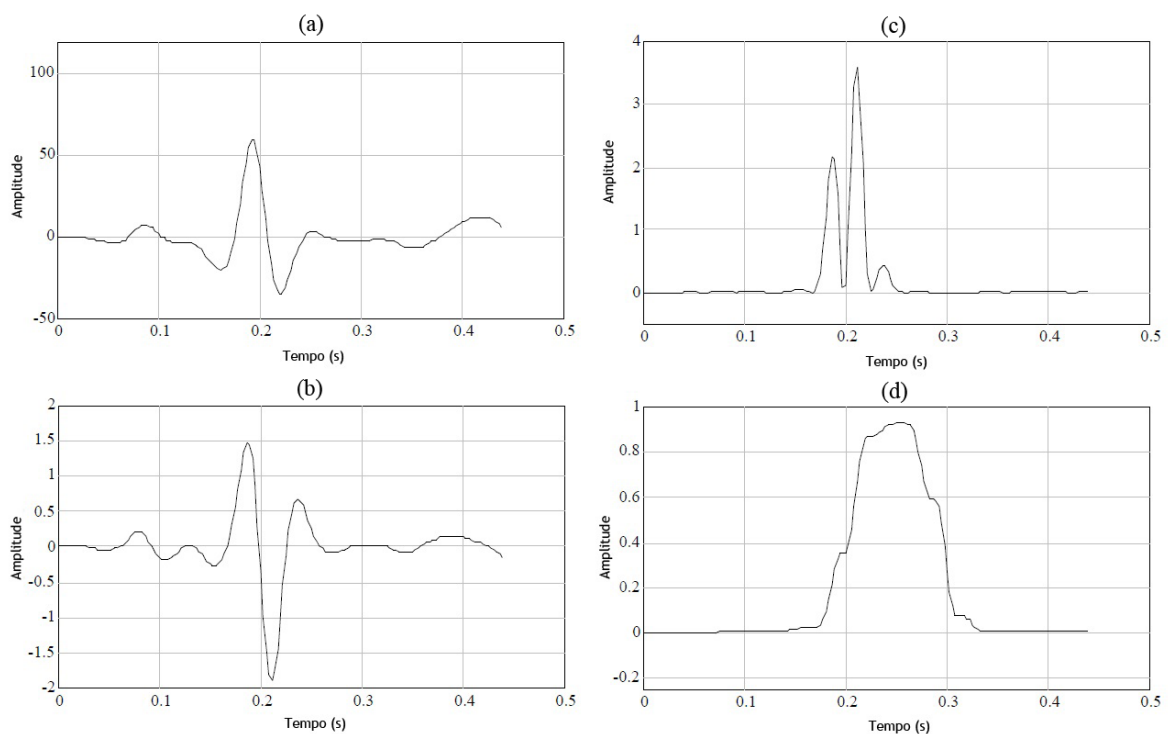


Figura 4.10 – (a) Sinal após o filtro passa-faixa; (b) sinal após a diferenciação; (c) sinal após a transformação quadrática; (d) sinal após o integrador de média móvel.

Seguindo a abordagem sugerida em [19], foram implementados um filtro passa-baixas seguido de um passa-altas, para, em cascata, funcionarem como um filtro passa-faixa.

Isso foi feito porque os filtros foram projetados para terem coeficientes inteiros, possibilitando a utilização de rotações de bits para a realização de multiplicações e divisões em potências de dois. Dessa maneira, o esforço computacional é reduzido e o algoritmo proporciona uma resposta mais rápida.

Em seguida são mostradas as respostas de cada estágio do processo. Note que são exibidas apenas as formas de onda, sem escalas, pois o algoritmo busca analisar a morfologia do complexo QRS e alguns estágios do processamento inserem ganhos e *offsets*, fazendo com que os valores absolutos percam significado para outro tipo de análise. Além disso, é importante ressaltar que essas formas de onda são mostradas apenas para comprovar a implementação do algoritmo, uma vez que, quando o sistema está em funcionamento, essas etapas somente servem à detecção de complexos QRS e posterior cálculo de frequência cardíaca, não sendo traçadas em momento algum.

As equações de cada uma dessas etapas são apresentadas a seguir, onde, em cada estágio, x representa valores de entrada e y representa valores de saída.

Filtro passa-baixas:

$$y(n) = 2y(n - 1) - y(n - 2) + x(n) - 2x(n - 6) + x(n - 12) \quad (4.2)$$

Este filtro apresenta um ganho de 36. Para que não haja saturação, os valores de saída são divididos por 36.

Filtro passa-altas:

$$y(n) = x(n - 16) - \frac{1}{32} [y(n - 1) + x(n) - x(n - 32)] \quad (4.3)$$

Derivação:

$$y(n) = 0,1 [2x + x(n - 1) - x(n - 3) - 2x(n - 4)] \quad (4.4)$$

Função quadrática:

$$y(n) = [x(n)]^2 \quad (4.5)$$

Integrador de janela móvel:

$$y(n) = \frac{1}{N} [x(n - N - 1) + x(n - N - 2) + \dots + x(n)] \quad (4.6)$$

Onde N é o número de amostras que define o intervalo da janela de integração.

A janela do integrador de média móvel é definida para que o intervalo de integração compreenda complexos QRS de duração longa, mas não se sobreponha à onda T.

Segundo [19], é possível calcular N de acordo com a equação 4.7 a seguir.

$$N = \frac{t_{QRS}}{T_s} \quad (4.7)$$

Onde t_{QRS} representa o tempo que permita a detecção de complexos QRS longos mas não incorpore ondas T (150 [ms]) e T_s representa o período de amostragem. Como a frequência de amostragem foi definida 500 [Hz] para este projeto, T_s vale 2 [ms] e N vale 75.

As respostas dos filtros são mostradas a seguir.

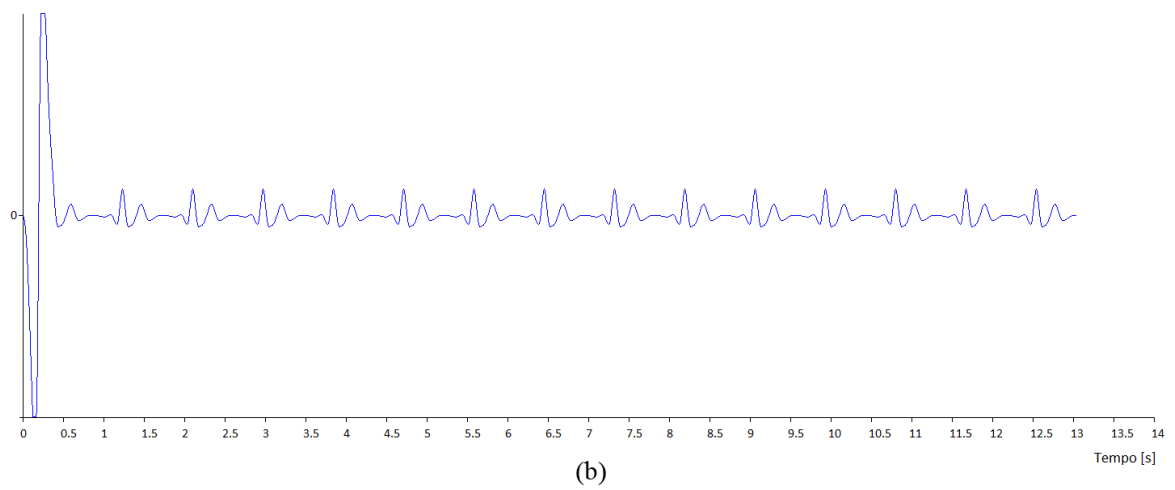
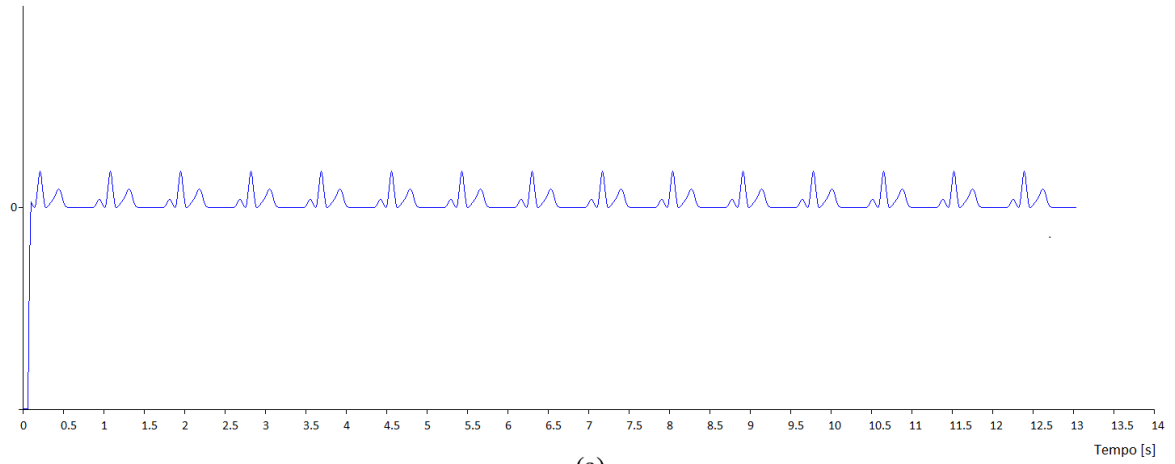
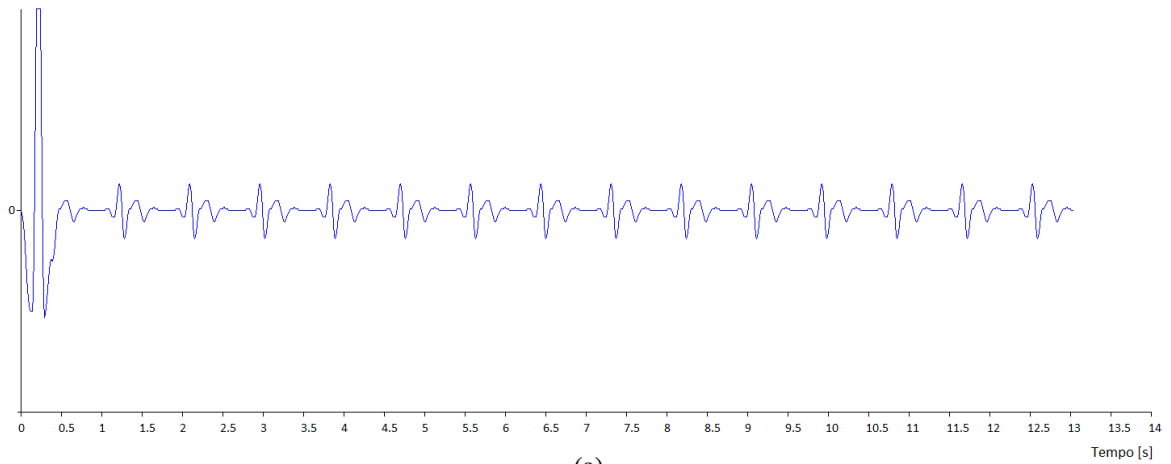
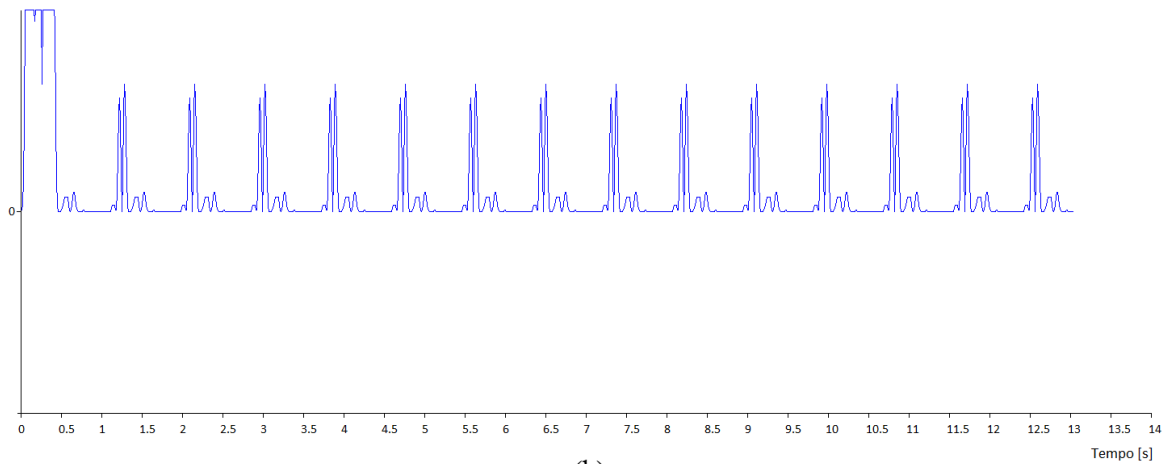


Figura 4.11 – (a) Sinal após filtro passa-baixas; (b) sinal após filtro passa-altas.

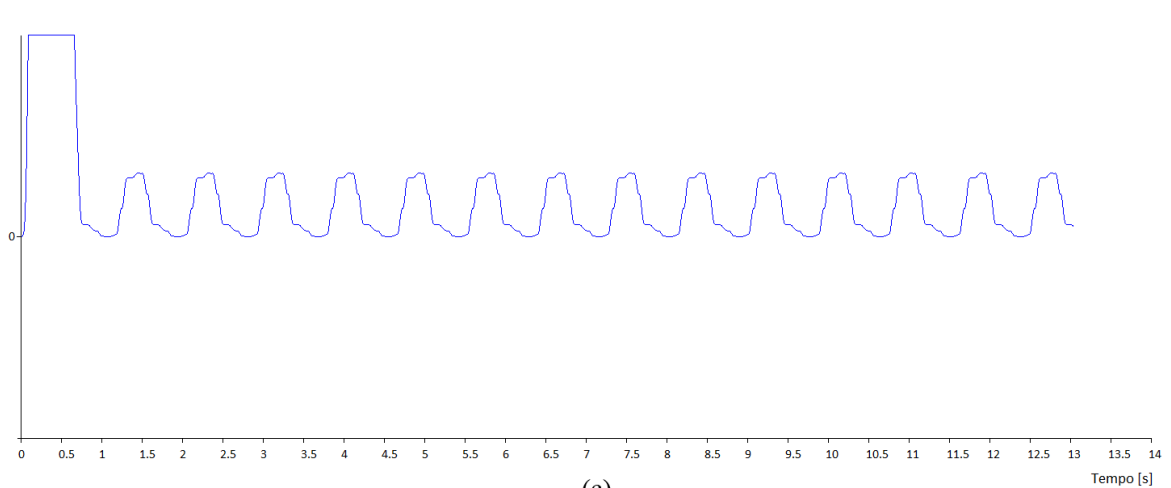
A equação e a resposta do estágio de diferenciação são mostradas a seguir, bem como as repostas da transformação quadrática e do integrador de média móvel.



(a)



(b)



(c)

Figura 4.12 – (a) Sinal derivado; (b) sinal após a transformação quadrática; (c) sinal após integrador de janela móvel.

O grupo de limites é definido de tal forma que picos de sinal (complexos QRS válidos sejam detectados). Após o sinal ter sido filtrado, seu SNR (*Signal-to-Noise Ratio*) aumenta, permitindo a utilização de *thresholds* logo acima dos picos de ruído (ondas T, ruído muscular), melhorando a sensibilidade do detector.

Em seguida são mostradas as curvas do ECG original e da detecção de complexos QRS.

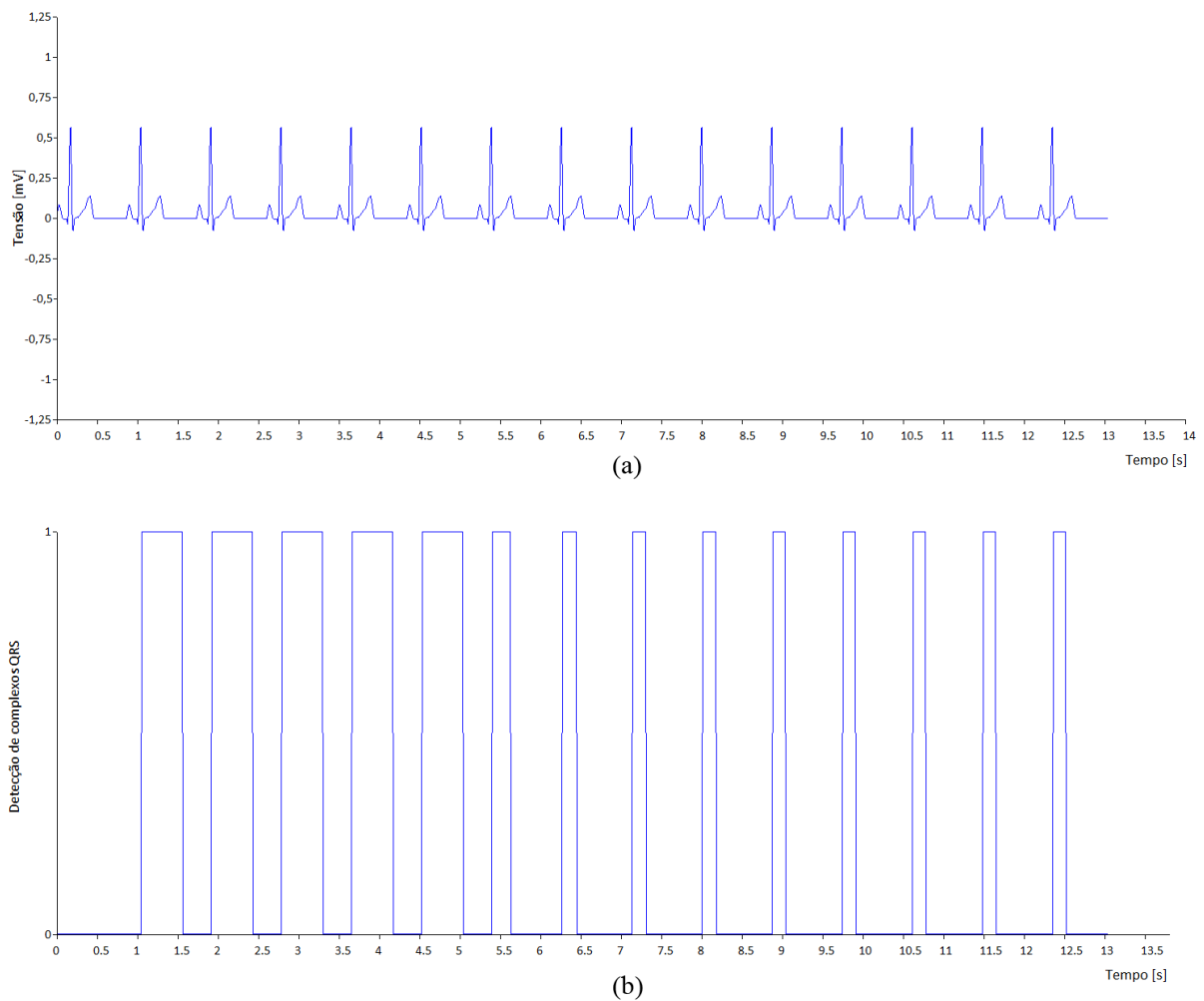


Figura 4.13 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.

Os primeiros pulsos do sinal de saída do algoritmo de detecção de complexos QRS são mais largos que os seguintes por se tratar ainda da fase de aprendizado do mesmo. Nesse período, há inicialização de variáveis e os *thresholds* adaptativos ainda não atingiram a faixa de valores que representa o regime permanente do algoritmo. Além disso, o olhar atento é capaz de perceber ligeiro atraso entre os pulsos de detecção de complexos QRS e momento

em que tais complexos, de fato, ocorrem. Este pequeno atraso é devido ao processamento do algoritmo de detecção. De qualquer forma, tal atraso não mostra efeito nocivo à monitorização de sinais de ECG.

Após a detecção de complexos QRS, a frequência cardíaca é calculada com base na quantidade de complexos QRS detectados em períodos de 4 segundos. A escolha do período de 4 segundos para cálculo de frequência cardíaca foi feita para garantir a detecção correta de assistolias (frequência cardíaca inferior a 15 BPM), a arritmia mais grave que um ser humano pode apresentar.

Seguindo a recomendação [12] e lendo-se o valor de frequência cardíaca calculado após 20 segundos (tempo que garante a estabilização do algoritmo), foi obtido o valor de 80 [BPM] (batimentos por minuto), que corresponde ao valor obtido através de inspeção visual.

Devido à necessidade da utilização de arquivos contendo dados eletrocardiográficos reais para a realização de testes relativos à detecção de complexos QRS e cálculo de frequência cardíaca, optou-se pela utilização de dados disponibilizados gratuitamente no portal PhysioNet. O próprio portal referencia o documento [12] como fonte para uma metodologia de testes a serem realizados com esses dados.

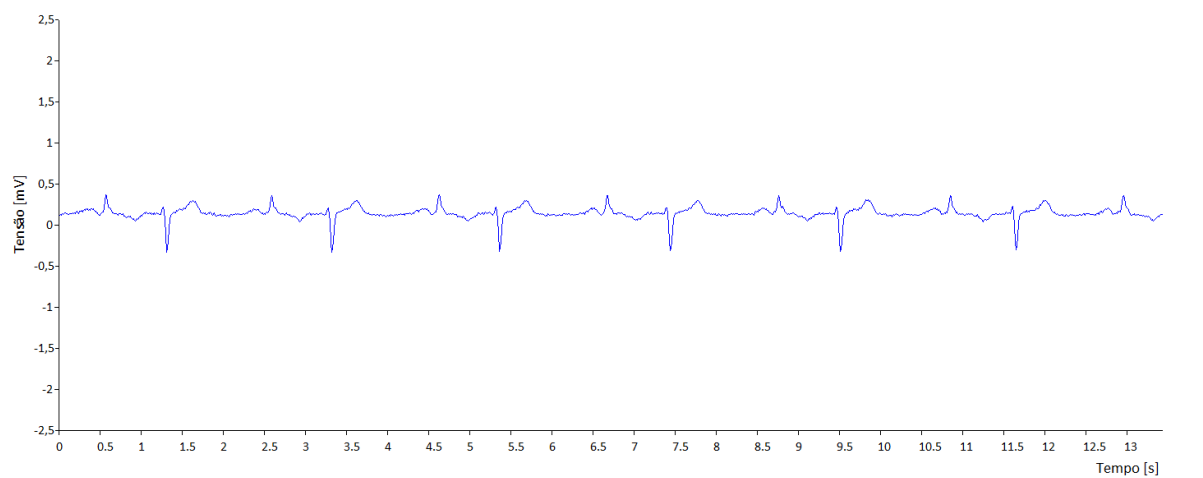
Em seguida, de acordo com as indicações [12], foram baixados arquivos binários específicos do portal PhysioNet que contém ondas eletrocardiográficas obtidas através de medidas reais.

São quatro ondas indicadas para o teste de desempenho de eletrocardiógrafo com cálculo de frequência cardíaca. Cada uma dessas ondas representa uma condição cardíaca específica, para que seja possível testar a confiabilidade da detecção de complexos QRS e cálculo de frequência cardíaca. As ondas são apresentadas a seguir, com os respectivos valores de frequência cardíaca obtidos.

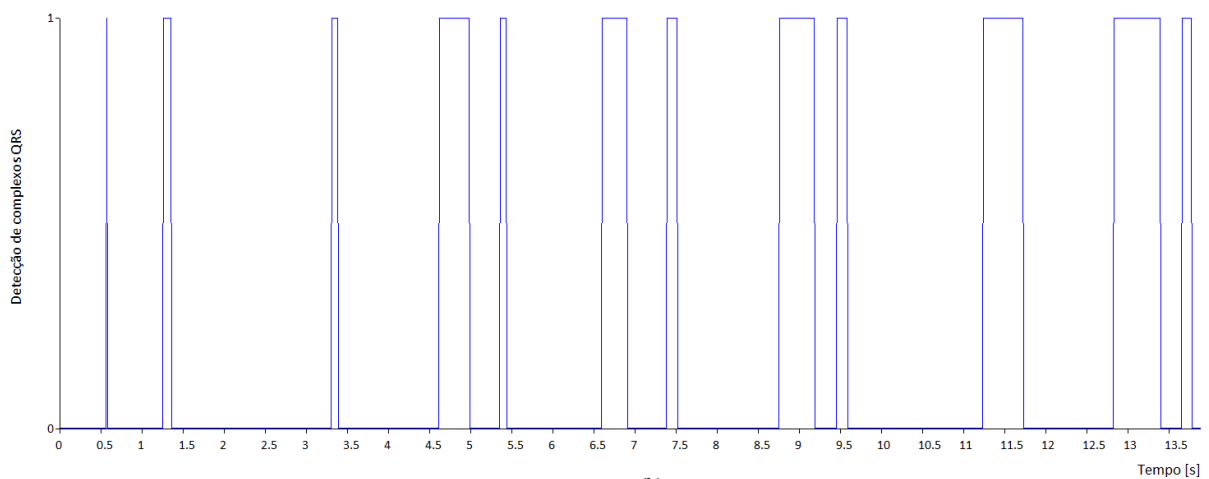
Bigeminismo ventricular:

A primeira das ondas contendo cardiopatias a ser utilizada para teste é a que contém bigeminismo ventricular.

Após o período de 20 segundos, a frequência cardíaca registrada foi de 80 [BPM], indicando que todos os complexos QRS foram contados, concordando com os resultados esperados [12].



(a)



(b)

Figura 4.14 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.

Bigeminismo ventricular alternado lento:

A segunda onda eletrocardiográfica destinada a testes contém a cardiopatia conhecida como bigeminismo alternado lento.

Após o período de 20 segundos, a frequência cardíaca registrada foi de 60 [BPM], indicando que todos os complexos QRS foram contados, concordando com os resultados esperados [12].

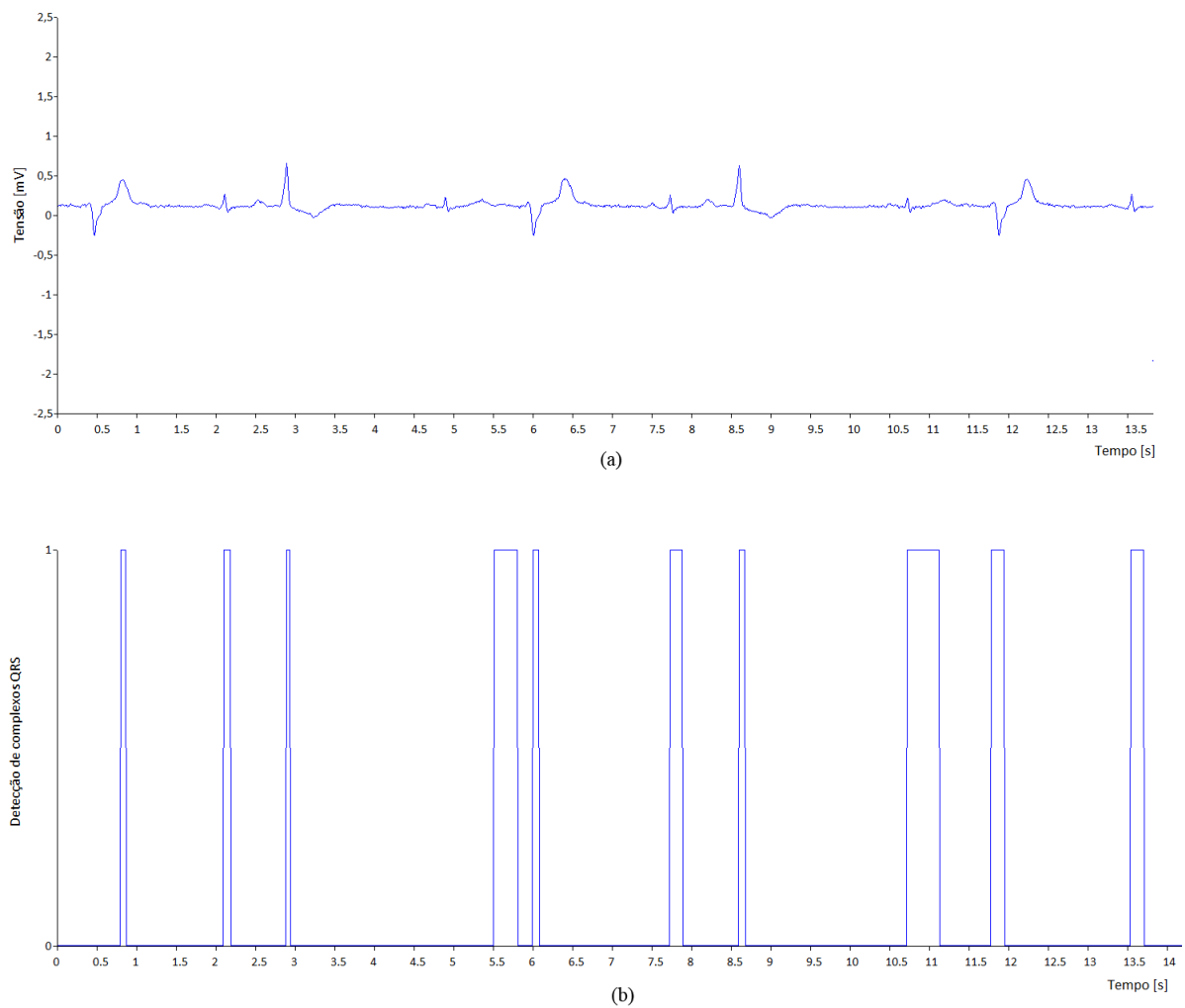
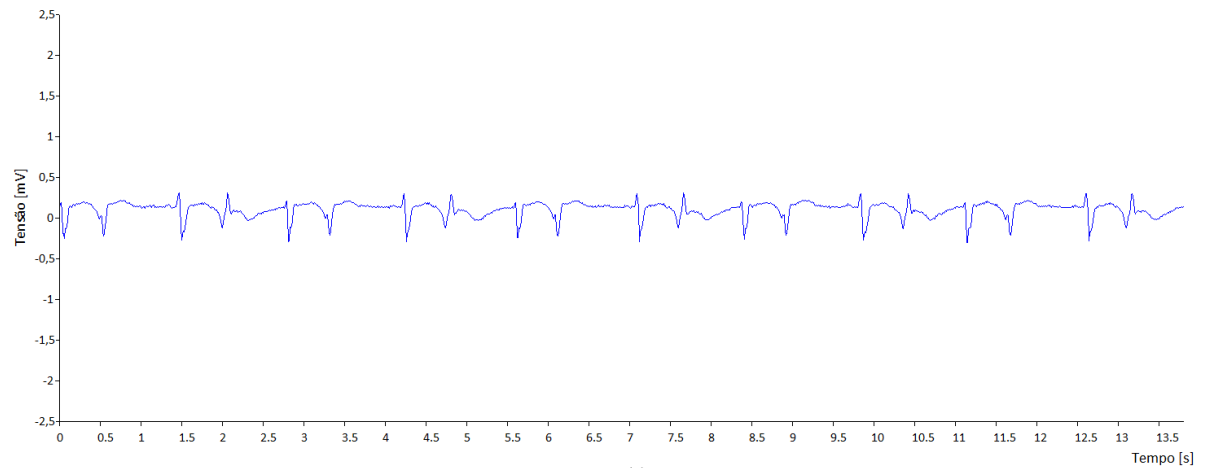


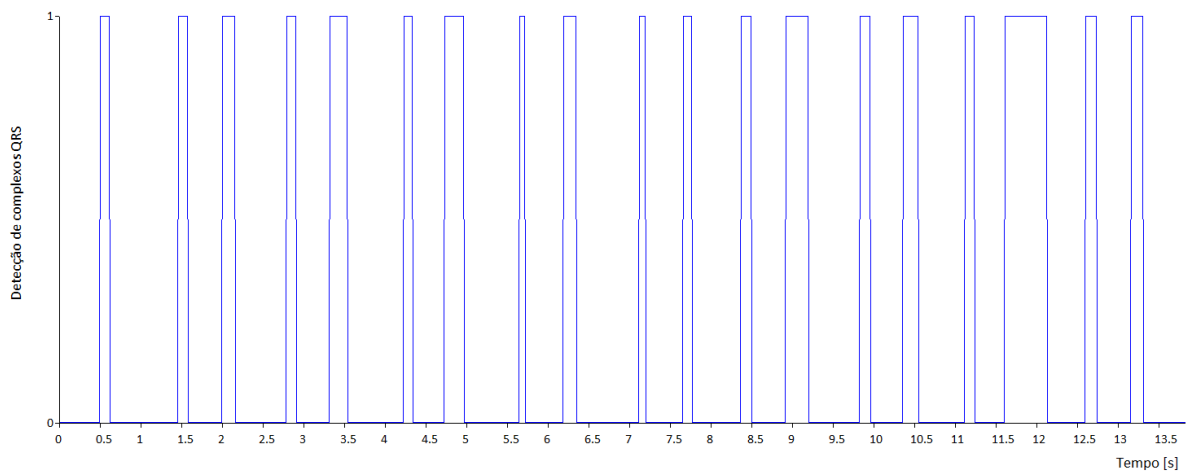
Figura 4.15 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.

Bigeminismo ventricular alternado rápido:

Após o período de 20 segundos, a frequência cardíaca registrada foi de 120 [BPM], indicando que todos os complexos QRS foram contados, concordando com os resultados esperados [12].



(a)

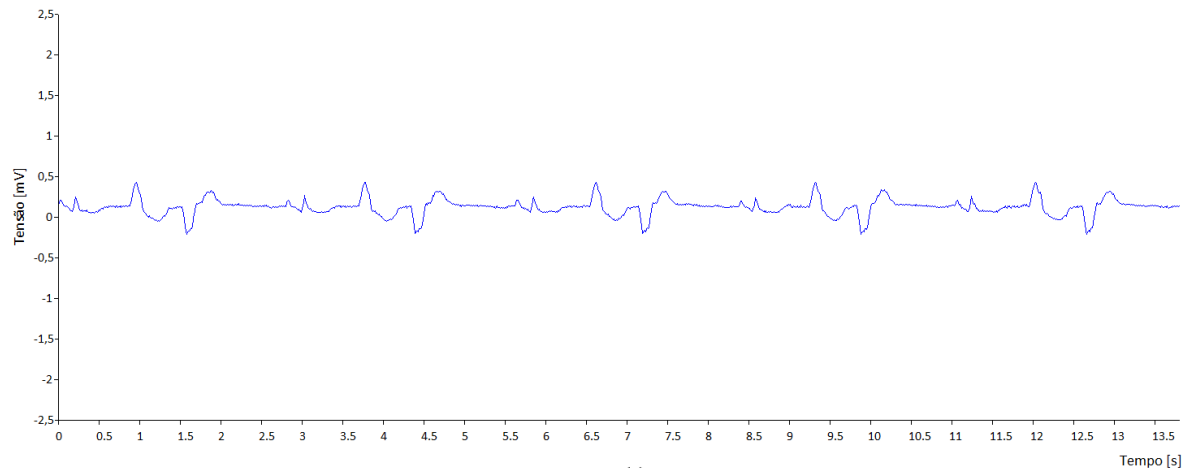


(b)

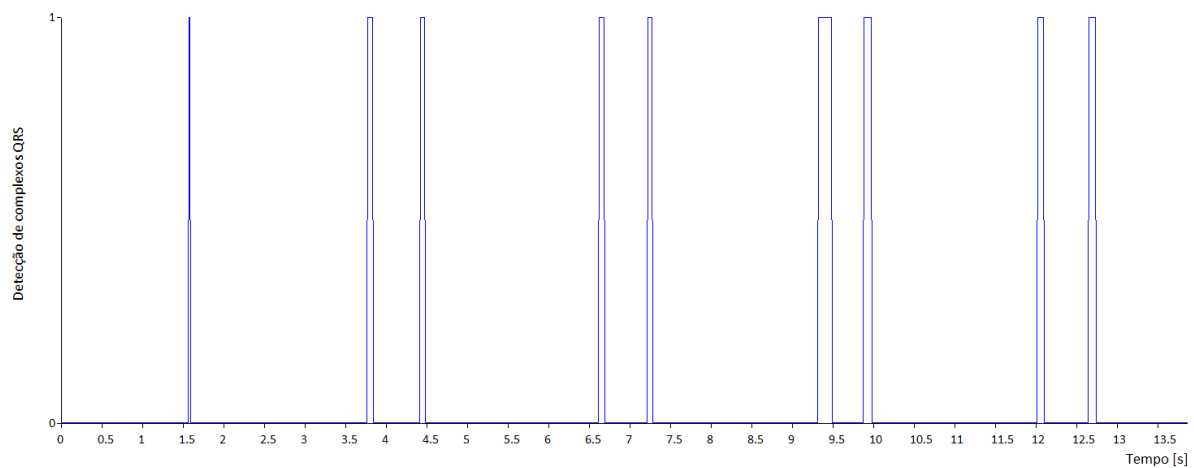
Figura 4.16 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.

Sístoles bidirecionais:

Após o período de 20 segundos, a frequência cardíaca registrada foi de 45 [BPM], indicando que as ondas R maiores foram contadas, concordando com os resultados esperados [12].



(a)



(b)

Figura 4.17 – (a) Sinal eletrocardiográfico original; (b) onda quadrada que representa a detecção de complexos QRS.

É possível perceber, então, que a detecção de complexos QRS e o cálculo de frequência cardíaca se mostraram confiáveis.

4.6.4 Linearização de temperatura

Com o auxílio de valores medidos e do fundo de escala determinado para esta aplicação (tabela 4.1), estabeleceu-se retas que linearizam a temperatura adequadamente, para diferentes faixas dinâmicas, proporcionando uma resposta mais exata.

Fazendo uso do programa Microsoft Excel, foram realizadas linearizações das curvas de temperatura através de projeções lineares.

Tabela 4.1 – Valores de temperatura medidos e linearizados

T [°C]	R [Ω]	V 10k [V]	V _{Out} [V]	I _{NTC} [uA]	Palavra digital	Equação de linearização	Temperatura linearizada [°C]	Erro [%]
20	6246	0,0799	2,142	7,990	2658,0		19,95	0,23%
21	5971	0,0798	2,106	7,980	2613,4		21,01	0,06%
22	5709	0,0799	2,070	7,990	2568,7	82,893943584 - 0,0236788469642452*	22,07	0,32%
23	5461	0,0799	2,038	7,990	2529,0	palavra	23,01	0,05%
24	5225	0,0799	2,006	7,990	2489,3		23,95	0,20%
25	5000	0,0798	1,976	7,980	2452,0		24,95	0,22%
26	4786	0,0799	1,890	7,990	2345,3		26,04	0,17%
27	4583	0,0799	1,812	7,990	2248,5	50,1592627 - 0,0102827229481294*	27,04	0,14%
28	4389	0,0799	1,735	7,990	2153,0	palavra	28,02	0,07%
29	4204	0,0799	1,662	7,990	2062,4		28,95	0,16%
30	4029	0,0799	1,593	7,990	1976,8		29,95	0,18%
31	3861	0,0799	1,526	7,990	1893,6		31,03	0,10%
32	3702	0,0799	1,463	7,990	1815,5	55,6958727 - 0,0130256310619338*	32,05	0,15%
33	3549	0,0799	1,402	7,990	1739,8	palavra	33,03	0,10%
34	3404	0,0799	1,346	7,990	1670,3		33,94	0,18%
35	3266	0,0799	1,291	7,990	1602,0		34,98	0,05%
36	3134	0,0799	1,239	7,990	1537,5		36,06	0,16%
37	3008	0,0799	1,194	7,990	1481,6	61,6865931 - 0,0166699830928156*	36,99	0,03%
38	2888	0,0799	1,148	7,990	1424,6	palavra	37,94	0,16%
39	2773	0,0799	1,095	7,990	1358,8		39,04	0,09%
40	2664	0,0799	1,053	7,990	1306,7		39,93	0,18%
41	2559	0,0799	1,011	7,990	1254,6		41,04	0,09%
42	2459	0,0799	0,973	7,990	1207,4	67,71635 - 0,021267264878713542	42,04	0,09%
43	2363	0,0799	0,934	7,990	1159,0	* palavra	43,07	0,16%
44	2272	0,0799	0,899	7,990	1115,6		43,99	0,02%
45	2185	0,0799	0,863	7,990	1070,9		44,94	0,13%

Através da tabela acima é possível conferir, além das equações que possibilitam a linearização, alguns valores de temperatura após a linearização dos valores medidos e seus erros.

A linearização feita apresentou erros inferiores a 0,1 [°C]. Em termos percentuais, os erros foram inferiores a 0,35% para todos os valores apresentados na tabela acima. A média de erro foi de 0,13%, mantendo boa proximidade com os valores esperados e indicando que se trata de uma boa linearização que atende a padrões comerciais.

4.7 Interface visual

Uma vez determinadas as informações e os controles que deveriam estar disponíveis ao operador do programa, foi feito o *layout* do mesmo. O resultado é apresentado na figura 4.19 e explicado em seguida.

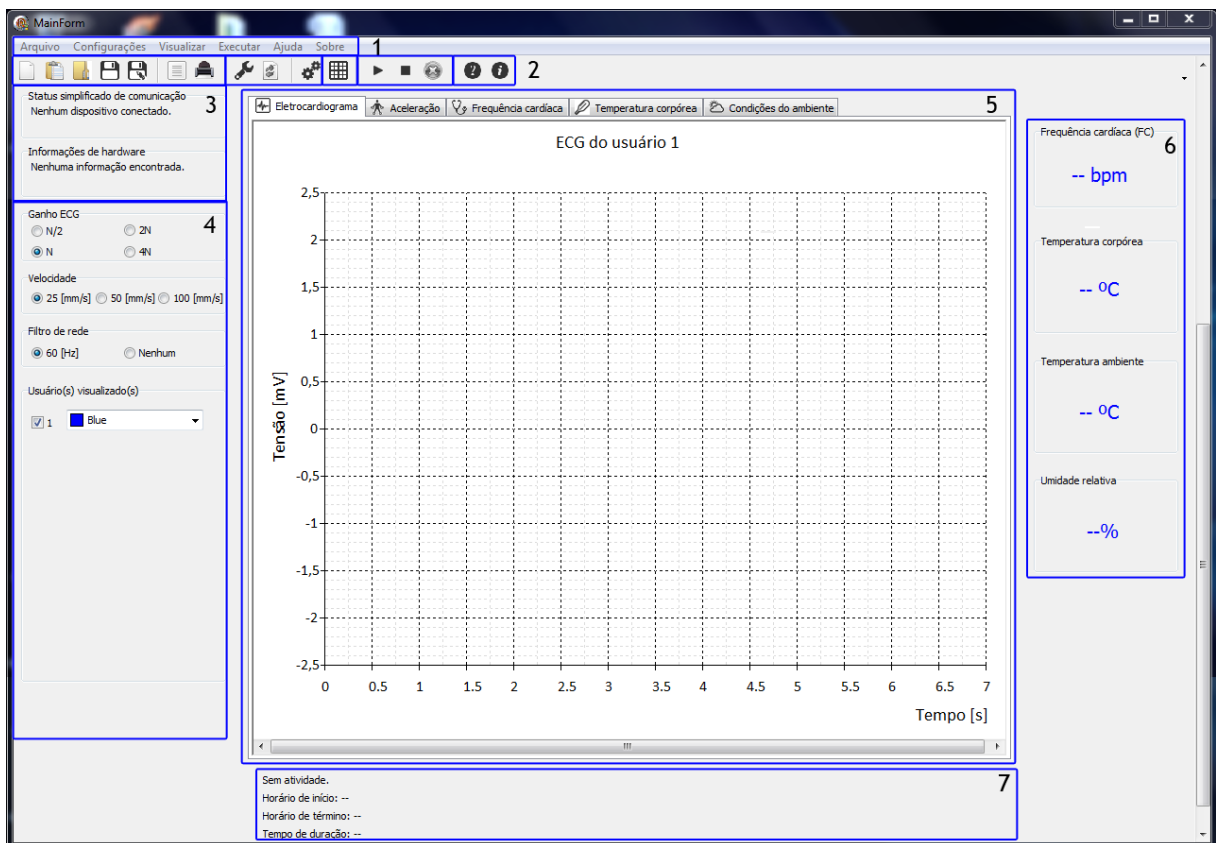


Figura 4.18 – Interface visual final.

- 1) **Menu:** Contém comandos referentes ao tratamento de arquivos, configurações do sistema, comandos referentes à execução da atividade monitorizada e também informações sobre o sistema.

- 2) **Barra de ferramentas:** Teclas de acesso rápido para comandos referentes ao tratamento de arquivos, configurações do sistema, exibição de *grid* nos gráficos, execução da atividade monitorizada e informações sobre o sistema.
- 3) **Quadros de *status*:** Informações referentes aos dispositivos conectados ao computador e *status* da comunicação.
- 4) **Quadros para configuração em tempo real:** Controle de ganho do ECG, aplicação do algoritmo *Turning Point*, filtros de rede e vídeo, e escolha de usuários visualizados e suas respectivas cores associadas.
- 5) **Área para traçados gráficos:** área destinada aos traçados dos diferentes gráficos. Cada aba serve ao traçado de uma grandeza diferente na seguinte ordem: ECG, aceleração (três eixos), frequência cardíaca, temperatura corpórea e condições do ambiente (temperatura ambiente e umidade relativa do ar).
- 6) **Quadros para exibição de valores instantâneos:** Algumas grandezas podem ser interpretadas a partir de valores instantâneos, sem a necessidade de acompanhar a evolução de suas curvas. Estes quadros exibem valores de frequência cardíaca, temperatura corpórea, temperatura ambiente e umidade relativa do ar, possibilitando a visualização destes valores enquanto se acompanha outras curvas, como, por exemplo, ECG e acelerometria.
- 7) **Registros temporais:** Essa área exibe informações pertinentes à ocorrência ou não de atividade e horários de início e término, além da duração da atividade até o presente momento e depois de finalizada.

A seguir são mostradas as disposições dos gráficos, seus respectivos fundos de escala e traçados que exemplificam a funcionalidade do programa desenvolvido.

A figura 4.20 mostra a região determinada para o gráfico de ECG. Para este gráfico, foi implementado um controle de ganho que permite a ampliação do sinal para melhorar a visualização da onda. Esse ganho pode ser aplicado até quatro vezes, sempre dobrando a resolução do eixo y da onda. Consequentemente, os valores de escala em y podem ser: (1) -5 a 5 mV; (2) $-2,5$ a $2,5$ mV, valor *default* visto na figura 25; (3) $-1,25$ a $1,25$ mV; (4) $-0,625$ a $0,625$ mV. O sinal apresentado é o mesmo descrito anteriormente após a decodificação do protocolo.

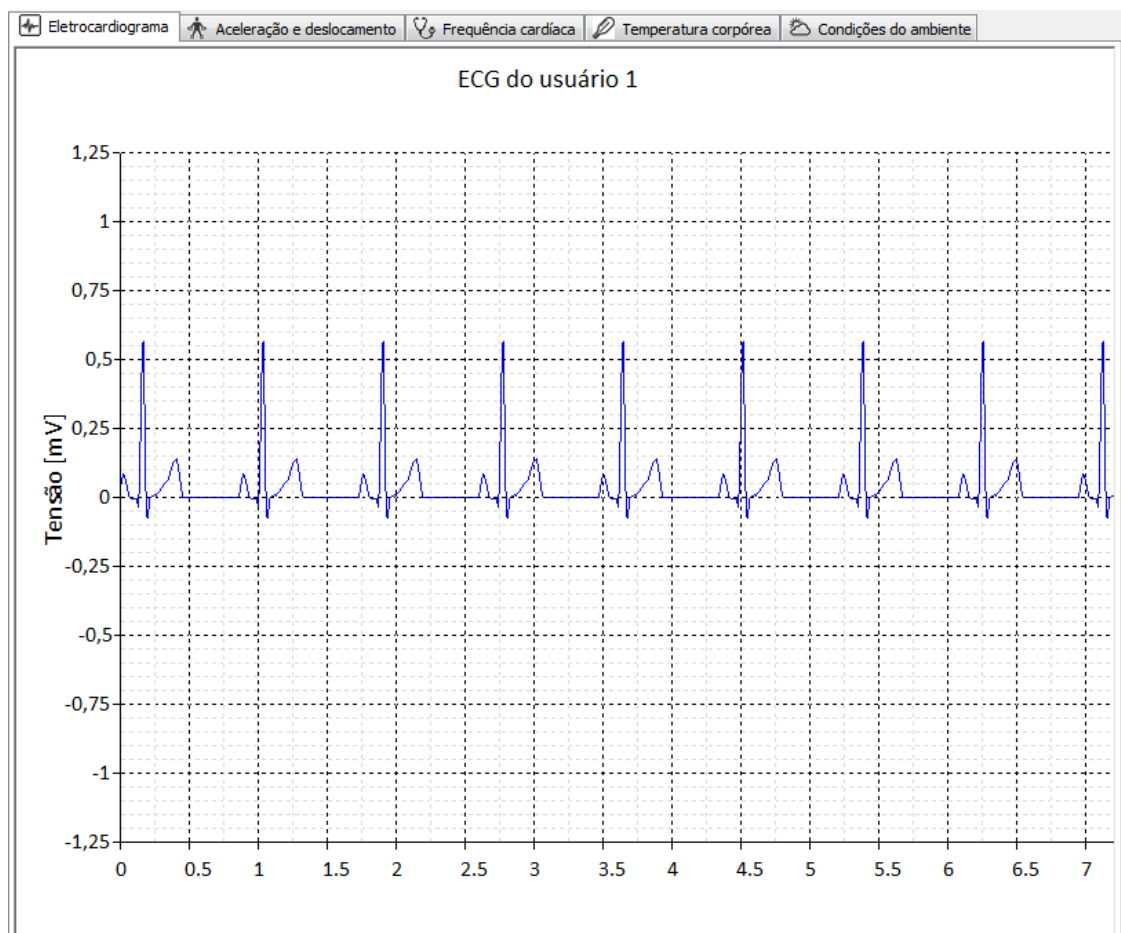


Figura 4.19 – Aba destinada ao gráfico de ECG.

A figura 4.21 mostra as disposições dos gráficos destinados aos sinais de acelerômetros e frequência cardíaca. Para melhor visualização, foram simuladas e exibidas ondas de acelerometria. Vale notar que essas ondas são meramente ilustrativas e não representam quaisquer sinais coletados. Já na porção referente ao gráfico de frequência cardíaca, é mostrada a onda referente à frequência cardíaca calculada a partir do sinal eletrocardiográfico mostrado na figura 4.20.

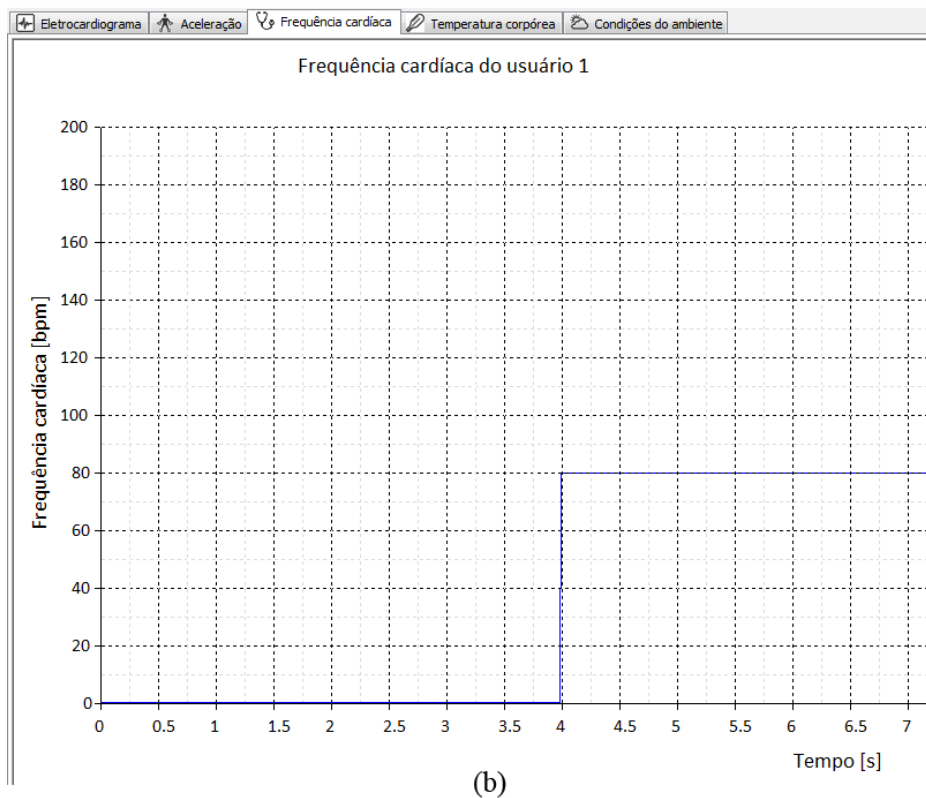
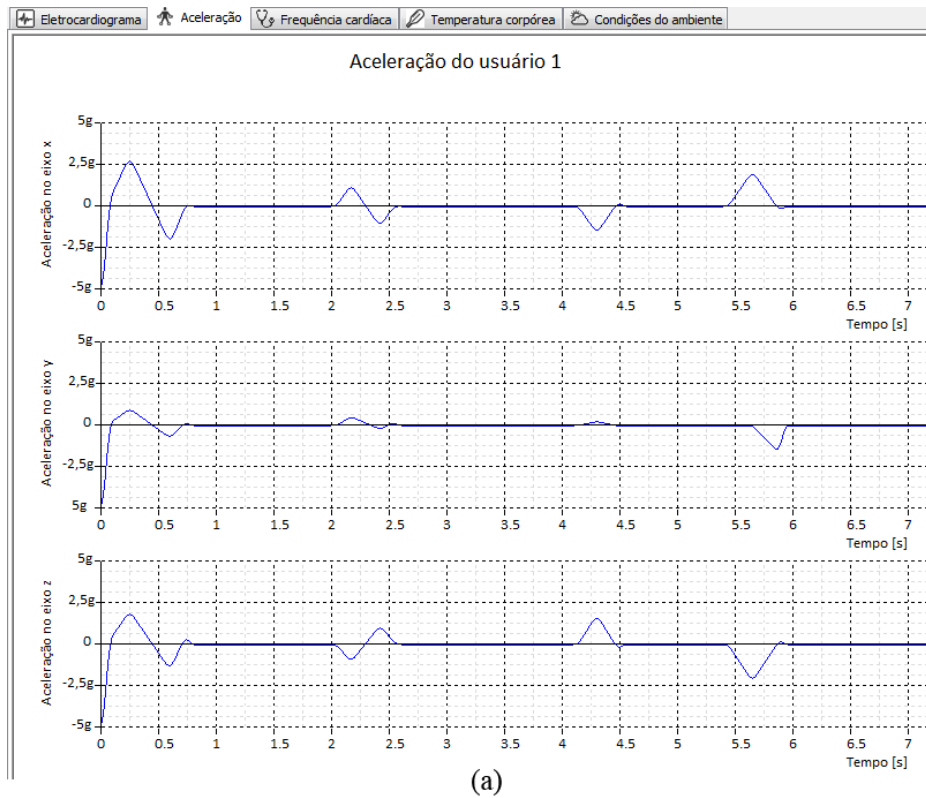


Figura 4.20 – (a) Aba destinada aos gráficos dos três eixos do sinal de acelerometria; (b) aba destinada à frequência cardíaca.

A figura 4.22 mostra as disposições dos gráficos destinados à exibição dos sinais de temperatura e caracterização do ambiente (temperatura ambiente e umidade relativa do ar). Para melhor visualização, foram simuladas e exibidas ondas de temperatura e umidade. Vale

notar que essas ondas são meramente ilustrativas e não representam quaisquer sinais coletados.

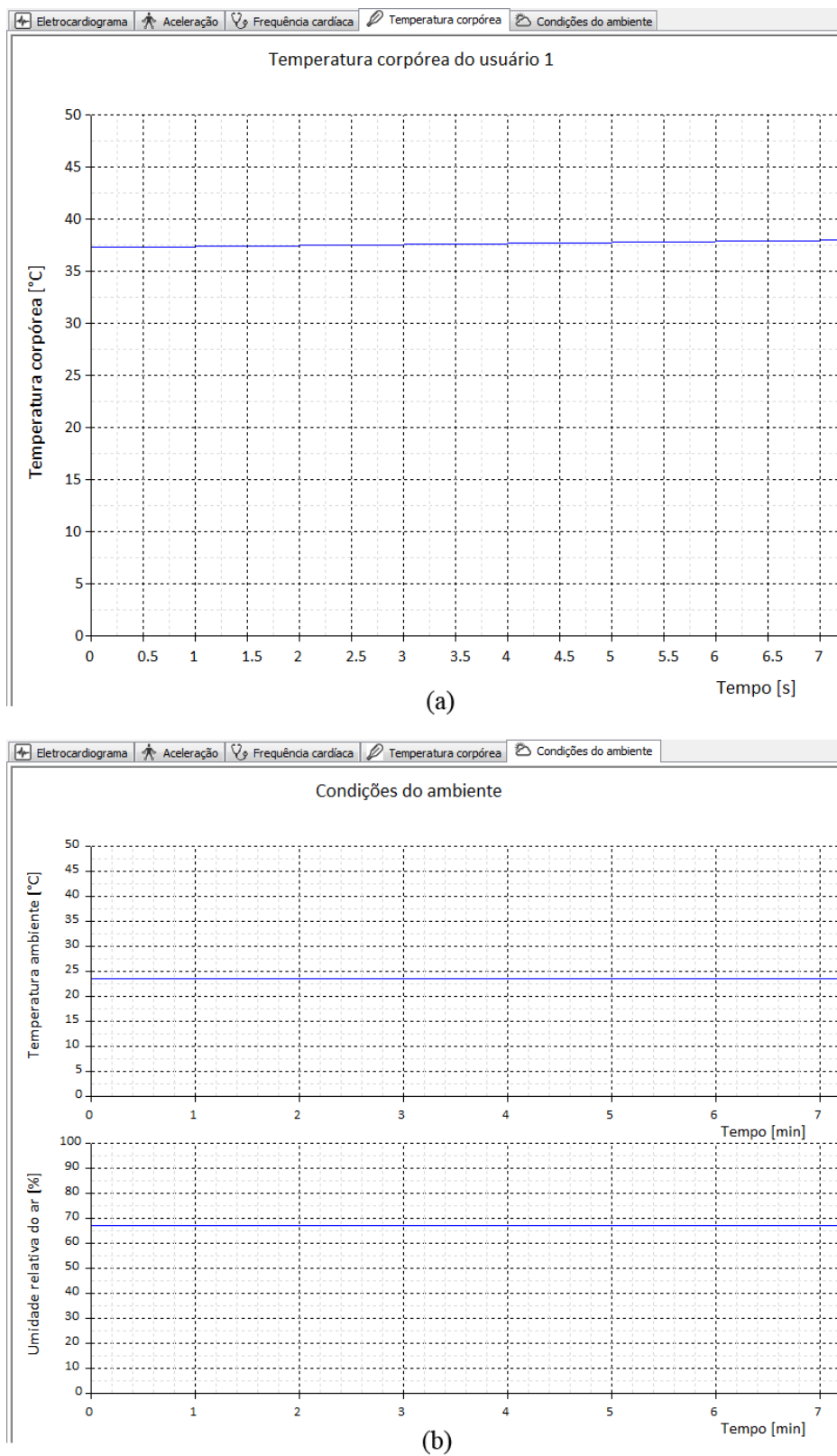


Figura 4.21 – (a) Aba destinada ao sinal de temperatura corpórea; (b) aba destinada aos sinais do ambiente (temperatura e umidade relativa do ar).

4.8 Estrutura do *software*

As figuras 4.23 e 4.24, que seguem, mostram os diagramas de classe que representam o programa implementado. Também são explicadas as relações entre os objetos do programa quando o mesmo está em funcionamento.

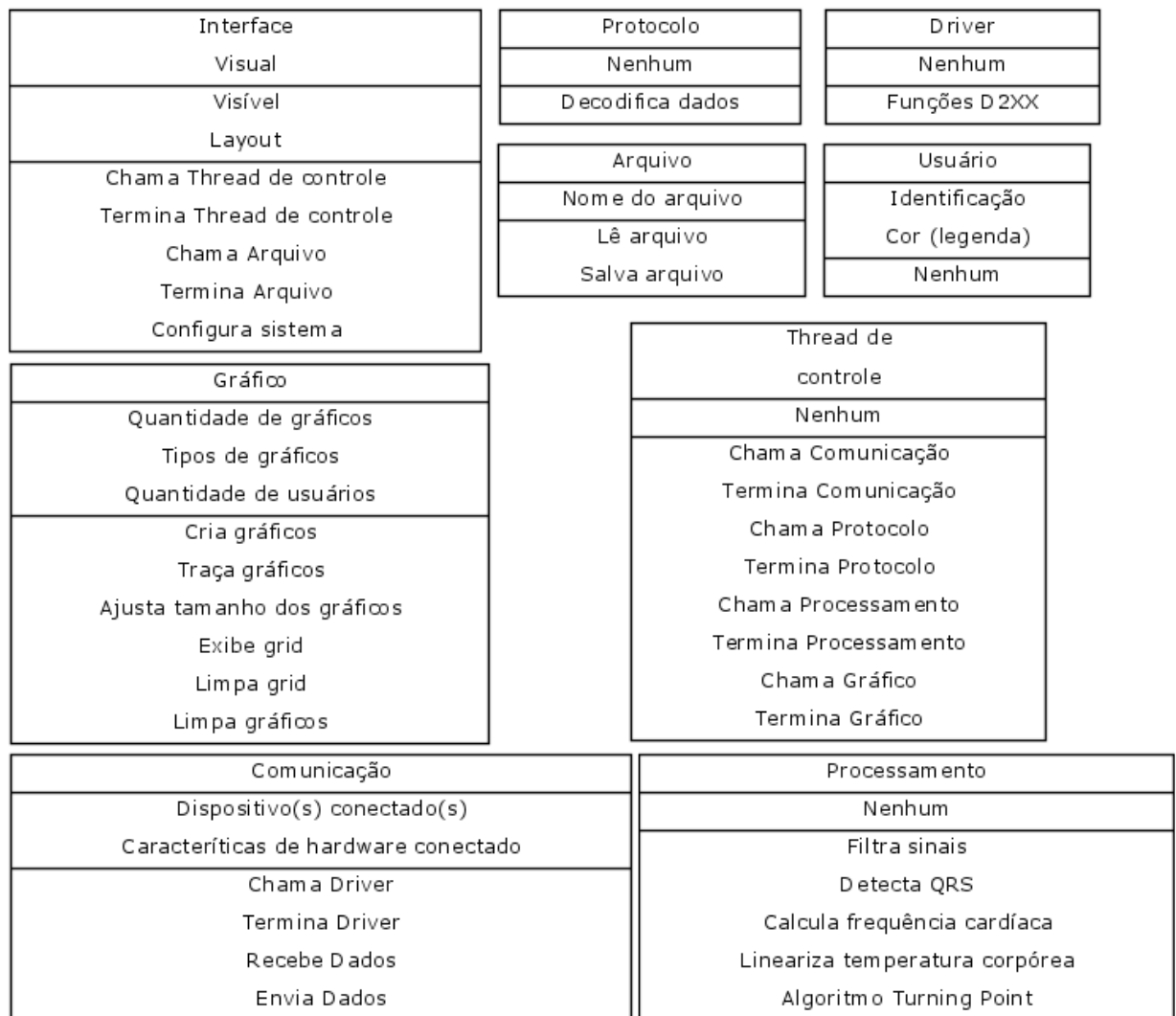


Figura 4.22 – Diagrama detalhando as classes do programa.

Interface visual: responsável pela interação com o operador do programa. Cuida dos itens configuráveis e da relação com os objetos Thread de controle e Arquivo, quando o operador desejar iniciá-los ou finalizá-los.

Thread de controle: garante a sincronia entre as tarefas relativas à comunicação, decodificação do protocolo, pós-processamento e traçado gráfico. Após iniciado, chama o objeto Comunicação para que seja feito o recebimento de dados. Em seguida encaminha os dados recebidos para serem decodificados pelo objeto Protocolo. Uma vez decodificados, os dados são transmitidos ao objeto Processamento para que sejam devidamente processados. Por fim, direciona os dados ao objeto Gráfico para que sejam exibidos.

Arquivo: se encarrega da leitura e armazenamento de arquivos, usando métodos das classes Protocolo, Processamento e Gráfico para o primeiro caso, e dados provenientes do objeto Comunicação para o segundo.

Comunicação: faz uso do objeto para realizar a comunicação USB.

Driver: contém as funções D2XX, necessárias para a realização da comunicação USB.

Protocolo: contém os métodos necessários à decodificação do mesmo.

Processamento: guarda os métodos necessários ao pós-processamento dos sinais.

Gráfico: cuida dos métodos para a geração e exibição dos diferentes gráficos do sistema na Interface visual.

Usuário: guarda as informações referentes à identificação, inclusive visual, de cada usuário. Dessa maneira, permite que os objetos Interface visual, Protocolo, Processamento e Gráfico façam as atribuições adequadas a cada conjunto de dados.

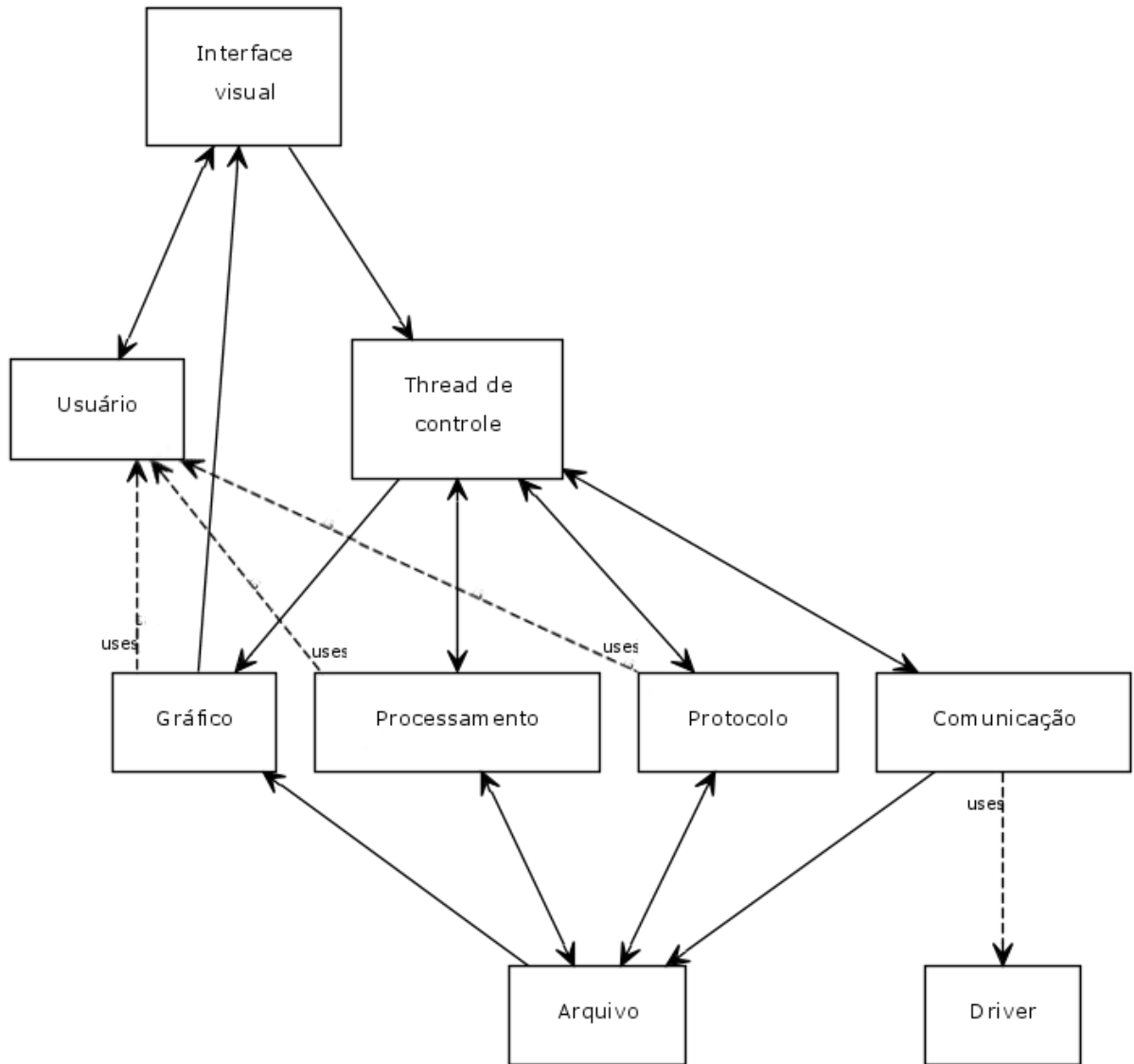


Figura 4.23 – Diagrama mostrando as relações entre as classes do programa.

Conclusão e Trabalhos Futuros

O objetivo deste trabalho foi o desenvolvimento de uma interface homem-máquina amigável e configurável para a monitorização de múltiplos sinais físicos e fisiológicos que fosse expansível a múltiplos usuários. A interface desenvolvida conta com capacidades de comunicação, pós-processamento de sinais, tratamento e exibição gráfica, além de gerenciamento de arquivos.

Foram abordados os desafios encontrados na concepção do projeto e as escolhas feitas ao longo do caminho que tornaram sua implementação viável, bem como a literatura que embasou o trabalho.

O programa obtido sedimenta as escolhas de tecnologias e estruturas realizadas ao longo do desenvolvimento, pois se mostrou funcional, amigável e intuitivo. Dessa forma, é possível concluir que o objetivo do trabalho foi atingindo.

Entretanto, é importante ressaltar que foram encontrados alguns entraves durante o desenvolvimento do projeto, como a dificuldade para a obtenção de recursos e realização de compras de componentes eletrônicos em tempo hábil. Tais dificuldades ocasionaram atraso considerável no cronograma do projeto e culminaram por impossibilitar a realização de testes integrados que envolvessem todo o sistema globalmente proposto (aquisição de sinais, transmissão sem fio, tratamento e apresentação de dados através da interface homem-máquina) e sua validação clínica.

Como sugestão para trabalhos futuros fica a realização de testes integrados que possibilitem que ajustes e calibrações sejam feitos, com o objetivo de proporcionar um desempenho satisfatório ao sistema de monitorização como um todo. Uma vez atingido desempenho satisfatório, o sistema poderá ser submetido à avaliação clínica, para viabilizar sua utilização em situações reais de treinos esportivos e atividades terapêuticas.

Vale ainda lembrar que, uma vez que o sistema consiga validação clínica, ele guarda grande potencial para contribuições nos âmbitos: esportivo, ao possibilitar análises de desempenho de atletas em seu ambiente de treino; terapêutico, ao permitir a monitorização de pacientes submetidos a tratamentos que envolvam atividades físicas; social, ao auxiliar o tratamento de pessoas com sequelas físicas, melhorando sua qualidade de vida.

Referências Bibliográficas

- [1] B. Massot, C. Gehin, R. Nocua, A. Ditmar, E. McAdams, “A wearable, low-power, health-monitoring instrumentation based on a Programmable System-on-Chip™”, 31st Annual International Conference of the IEEE EMBS, p. 4852-4855, September 2-6, 2009.
- [2] M. Zhang, A. A. Sawchuk, “A customizable framework of body area sensor network for rehabilitation”, Proc. Of 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, p.1-6, 2009.
- [3] A. Hadjidj, A. Bouabdallah, Y. Challal, “Rehabilitation supervision using wireless sensor networks”, Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, p. 1-3, 2011.
- [4] A. Dhamdhere, H. Chen, A. Kurusingal, V. Sivaraman, A. Burdett, “Experiments with wireless sensor networks for real-time athlete monitoring”, 5th IEEE International Workshop on Practical Issues in Building Sensor Network Applications, p. 938-945, 2010.
- [5] X. Yong, B. Tingting, B. Chunhua, S. Qinghong, M. Qianli, “Design of the athlete’s electrocardiogram monitoring and evaluation system based on wireless network”, First International Workshop on Complexity and Data Mining, p. 60-63, 2011.
- [6] C. A. Boano, M. Lasagni, K. Römer, T. Lange, “Accurate Temperature Measurements for medical Research using Body Sensor Networks”, 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, p. 189-198, 2011.
- [7] D. A. James, N. Davey, T. Rice, “An Accelerometer Based Sensor Platform for Insitu Elite Athlete Performance Analysis”, Proc. IEEE Sensors, V. 3, p. 1373-1376, 2004.
- [8] A. Pantelopoulos, N. G. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis”, *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, V. 40, N. 1, p. 1-12, January 2010.
- [9] R. Castro Jr., Sistema de Monitorização de Sinais Biológicos e Físicos à Distância para Atletas com Deficiências Físicas, Prova Científica UNIFEI, Itajubá – MG, 2010.
- [10] K. Malhi, S. C. Mukhopadhyay, J. Schnepper, M. Haefke, H. Ewald, “A ZigBee-Based Wearable Physiological Parameters Monitoring System”, *IEEE Sensors Journal*, V. 2, N. 3, p. 423 – 430, March 2012.
- [11] PhysioNet. Disponível em: <www.physionet.org>, Acesso em 23 de outubro de 2012.
- [12] *Cardiac monitors, heart rate meters, and alarms* [American National Standard (ANSI/AAMI EC13:2002)]. Arlington, VA: Association for the Advancement of Medical Instrumentation, 2002.
- [13] A. C. Guyton, J. E. Hall, *Tratado de Fisiologia Médica*, 11^a Ed, Rio de Janeiro, Elsevier, 2006.
- [14] J. G. Webster, *Medical Instrumentation: Application and Design*, 4th Ed, New York, John

Wiley and Sons, 2010.

- [15] H. M. Deitel, P. J. Deitel, *C++ Como Programar*, 3^a Ed, Porto Alegre, Bookman, 2001.
- [16] S. Teixeira, X. Pacheco, *Delphi 5 – Guia do desenvolvedor*, 1^a Ed, Rio de Janeiro, Campus, 2000.
- [17] J. Axelson, *USB Complete: The Developer's Guide*, 4th Ed, Madison – WI, Lakeview Research, 2009.
- [18] M. A. Garcia, R. S. G. Villar, A. L. R. Cardoso, J. C. T. B. Moraes, “A Low-Cost and High-Reliability Communication Protocol for Remote Monitoring Devices”, *IEEE Transactions on Instrumentation and Measurement*, V. 53, N. 2, p. 612 – 618, April 2004.
- [19] W. J. Tompkins, *Biomedical Digital Signal Processing*, 2nd Ed, New Jersey, Prentice Hall, 2000.
- [20] J. Pan, W. J. Tompkins, “A Real-Time QRS Detection Algorithm”, *IEEE Transactions on Biomedical Engineering*, V. BME-32, N. 3, p. 230 – 236, March 1985
- [21] MA series thermistor specifications, GE Thermometrics. Disponível em: <<http://biibs.sdsu.edu/images/datasheets/Thermistor.pdf>>, Acesso em 6 de junho de 2012.
- [22] C. Gong, “Human-Machine Interface: Design Principles of Visual Information in Human-Machine Interface Design”, International Conference on Intelligent Human-Machine Systems and Cybernetics, p. 262 – 265, 2009.
- [23] M. Cantú, *Mastering Delphi 7*, 1st Ed, Alameda – CA, SYBEX, 2003.
- [24] FT 232R datasheet, FTDI Ltd. Disponível em <http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf> Acesso em: 17 de fevereiro de 2012.
- [25] FTDI Programmer's Guide, FTDI Ltd. Disponível em <[http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide\(FT_000071\).pdf](http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf)>, Acesso em: 17 de fevereiro de 2012.
- [26] W. Stallings, *Data and Computer Communications*, 8th Ed, New Jersey, Pearson Prentice Hall, 2007.

Apêndice A – Publicações

O sistema que originou esta dissertação, além das duas outras mencionadas, gerou dois artigos publicados e apresentados sob a forma de pôster no XXIII Congresso Brasileiro de Engenharia Biomédica – CBEB – realizado entre os dias 01 a 05 de outubro de 2012 em Porto de Galinhas – PE.

Os artigos são apresentados, na íntegra, a seguir.

CIRCUITOS PARA AQUISIÇÃO E CONDICIONAMENTO DE SINAIS PARA WIRELESS BODY AREA SENSOR NETWORK (WBASN)

M. D. Aleixo, Jr.; G. Leal, J. Siqueira, P. C. Crepaldi e R. Castro, Jr.

Instituto de Engenharia e Tecnologias da Informação / Universidade Federal de Itajubá

marcos.aleixo.jr@gmail.com

Abstract: This paper envisions describing circuitry developed to assemble a ZigBee wireless body area sensor network. The goal of this network is to acquire physiological data from the subject without limiting movement. Through the acquisition of signals of body temperature, electrocardiography (ECG), triaxial acceleration, speed and displacement it is possible to provide accurate information about one's clinical improvement. By doing so, we aim to technically enhance the rehabilitation process for people with physical sequels.

Palavras-chave: ZigBee, Redes de sensores sem fio, condicionamento de sinais fisiológicos.

Introdução

A utilização de comunicação sem fio para implementação de rede de sensores vem adquirindo espaço em campos como: reabilitação [1], monitorização [2] e *home care* [3].

O sistema proposto visa coleta de sinais fisiológicos de um indivíduo durante execução de atividades físicas, sem que sua mobilidade seja comprometida. Para tanto, foi escolhida comunicação utilizando ondas de rádio.

Considerando a possibilidade de atividades em grupo, como prática de esportes coletivos, foi escolhido o protocolo *ZigBee*. Este protocolo apresenta maior facilidade de expansão, uma vez que cada elemento da rede pode operar como uma estação de repetição de informações.

Este projeto contempla a aquisição de sinais de temperatura corpórea, eletrocardiografia (ECG), aceleração, velocidade e deslocamento em até três eixos.

Para realizar a tarefa de coletar e transmitir dados fisiológicos, foram desenvolvidos circuitos de condicionamento de sinais, adequados para os transdutores utilizados.

A Figura 1 ilustra o posicionamento idealizado dos sensores para a montagem da rede.

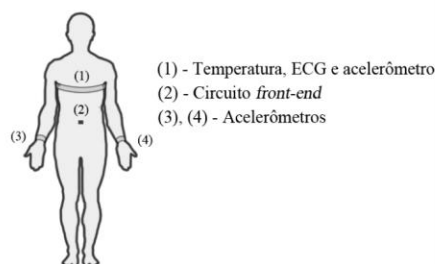


Figura 1: Posicionamento dos circuitos no corpo do usuário.

Estes circuitos se comunicarão com uma unidade transceptora localizada no paciente, denominada circuito *front-end*, que encaminhará os dados para outra unidade transceptora, denominada Unidade Transceptora de Gerenciamento de Comunicação (UTGC), conectada a um computador pessoal (PC).

Cada circuito terá como elemento de pré-processamento e gerenciamento de comunicação um microcontrolador. Cada microcontrolador possuirá *firmware* específico, todos ainda em fase de desenvolvimento.

O protocolo desenvolvido para a comunicação garantirá que cada sensor terá identificação única. Isto deverá ocorrer mesmo no caso da existência de múltiplos usuários.

No PC, os dados serão processados, exibidos e armazenados, possibilitando consulta local ou remota.

Materiais e Métodos

Para a implementação do sistema proposto foram desenvolvidos cinco circuitos:

Circuito para aquisição e condicionamento de sinal de aceleração, velocidade e deslocamento – Os sinais serão gerados a partir de um acelerômetro triaxial da *Analog Device*: o *ADXL325*. Este componente gera sinais analógicos de tensão para cada eixo.

Suas saídas são conectadas, individualmente, a filtros passa-baixa de frequência de corte igual a 100[Hz] [4]. Tais filtros consistem em apenas capacitores,

seguindo especificações fornecidas pelo próprio fabricante do acelerômetro.

O sinal já filtrado é, então, enviado ao conversor A/D do microcontrolador.

A Figura 2 ilustra o diagrama em blocos deste circuito.



Figura 2: Condicionamento do sinal do acelerômetro.

Circuito para aquisição e condicionamento de sinal de temperatura corpórea – Estes sinais são obtidos através de um Termistor de Coeficiente Negativo (*Negative Thermistor Coefficient* – NTC). Este elemento apresenta tolerância de $\pm 0.1[^\circ\text{C}]$ no intervalo de 35 a 39[$^\circ\text{C}$] [5].

Em circuitos para monitorização de temperatura, deve-se observar o fluxo de corrente pelo NTC [6]. Para tanto, o elemento sensor é alimentado por uma fonte de corrente controlada. O controle é realizado através de uma tensão obtida a partir de um resistor série localizado no mesmo ramo do circuito do sensor.

Este valor de tensão é comparado a um valor de tensão de referência estável. Este valor estável de tensão é obtido através de um circuito integrado regulador de tensão e um divisor resistivo. O resultado desta comparação determina a intensidade da fonte de corrente.

A corrente que flui através do NTC gera uma tensão diferencial que tem sua magnitude amplificada em cerca de cinquenta vezes.

Esta amplificação é dividida em dois estágios, sendo o primeiro realizado por um amplificador de instrumentação e o segundo por um amplificador não inversor.

O sinal amplificado passa, então, por um filtro passa-baixa, com frequência de corte de 5[Hz]. Por fim o sinal filtrado é enviado ao conversor A/D do microcontrolador.

A Figura 3 ilustra o diagrama em blocos deste circuito.

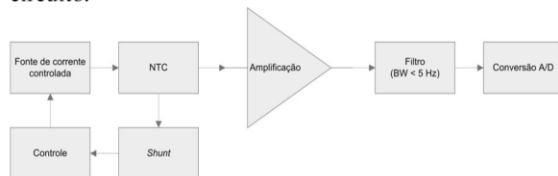


Figura 3: Condicionamento do sinal de temperatura corporal.

Circuito para aquisição e condicionamento de sinal de ECG – o circuito desenvolvido segue algumas observações:

- O circuito deve possuir sensibilidade a sinais entre $\pm 0,25[\text{mV}]$ e $\pm 5[\text{mV}]$;
- Alta impedância de entrada;

- Baixa corrente de fuga;
- Capacidade de condicionamento de sinais com frequências entre 0,5[Hz] e 40[Hz];
- Alta rejeição das tensões de modo comum.

O circuito recebe sinais provenientes de derivações polares [7]. Fisicamente, os eletrodos são confeccionados utilizando um tecido condutivo, a ser posicionado ao redor do tórax do usuário.

Os sinais são captados de dois pontos conectados ao tecido condutivo, operando como as terminações de braço direito (*right arm* – RA) e de braço esquerdo (*left arm* – LA).

Este sinal é pré-amplificado por um amplificador de instrumentação. Este estágio apresenta ganho de baixa ordem ($G = 3$) a fim de evitar saturação do dispositivo.

O amplificador de instrumentação é utilizado por possuir alta taxa de rejeição de modo comum (*Common Mode Rejection Ratio* – CMRR). Esta característica contribui para a eliminação de parte das tensões de modo comum. O restante destas tensões é eliminado com o auxílio do circuito *Driven Right Leg* (DRL) [8].

A saída do estágio de pré-amplificação é conectada a um filtro passa-alta (FPA) de frequência de corte igual a 0,5[Hz]. Sua função é eliminar tensões DC oriundas da interface entre o tecido condutivo e a pele do usuário.

A saída do filtro passa-alta é conectada a um segundo amplificador. Este estágio possui ganho superior ao anterior ($50 < G < 100$), resultando em valores mais compatíveis com a faixa dinâmica do conversor A/D presente no microcontrolador utilizado.

Em seguida, há um filtro passa-baixa (FPB) com frequência de corte igual a 40[Hz], de forma a eliminar componentes de alta frequência do sinal.

O sinal é, por fim, encaminhado ao seu respectivo canal de conversão A/D do microcontrolador.

No microcontrolador será implementado um filtro rejeita faixa digital. Os dados serão, então, transmitidos por ondas de rádio com o auxílio de antenas. O *firmware* a ser implementado no microcontrolador ainda se encontra em fase de desenvolvimento.

A Figura 4 ilustra o diagrama em blocos de tal circuito.

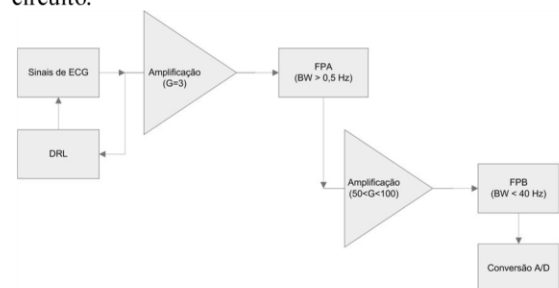


Figura 4: Condicionamento do sinal de ECG.

Circuito Front-End – Este circuito será responsável pela recepção dos sinais enviados pelos microcontroladores localizados nos circuitos de

condicionamento de sinais dos sensores. Também será sua responsabilidade encaminhar estes sinais à UTGC.

Este encaminhamento será realizado com auxílio de um circuito integrado de amplificação de sinais de rádio, garantindo maior alcance da comunicação.

UTGC – Este circuito será responsável pela recepção de dados enviados pelo circuito *front-end*. Para isto, este circuito estará equipado com o mesmo amplificador de RF especificado para o circuito *front-end*. Também realizará a coleta de dados referentes ao ambiente (temperatura e umidade).

Estes dados serão obtidos por meio da utilização de um sensor comercial dedicado à transdução destes sinais.

Todos os dados supracitados serão encaminhados a um PC, através da porta USB. A comunicação será realizada através de circuito integrado específico para comunicação via protocolo USB.

No PC haverá um programa dedicado para processamento, exibição e armazenamento das informações.

Tecnologias e componentes escolhidos – A Tabela 1 apresenta os componentes escolhidos para o desenvolvimento dos circuitos descritos anteriormente.

Tabela 1: Tamanho de Dispositivos especificados.

Componente	Modelo	Fabricante
Microcontrolador (circuitos dos sensores)	CC2531	Texas Instruments
Microcontrolador (circuito <i>front-end</i>)	CC2531	Texas Instruments
Microcontrolador (circuito UTGC)	CC2533	Texas Instruments
Acelerômetros	ADXL325	Analog Devices
NTC	MA300TA 502B	GE Thermometrics
Tecido Condutivo	MedTex	Shieldex
Sensor de temperatura e umidade relativa do ar (UTGC)	SHT25	Sensirion
Amplificador de ondas de rádio	CC2591	Texas Instruments
CI – comunicação USB	FT232RL	FTDI Ltd.
Amplificadores operacionais	AD8609	Analog Devices
Amplificadores de instrumentação	AD8553	Analog Devices

Resultados Esperados

Com este trabalho, espera-se o desenvolvimento de equipamento dedicado à monitorização de indivíduos com necessidades físicas especiais durante execução de suas atividades físicas.

Discussão

Este artigo apresenta as idéias utilizadas para o desenvolvimento de circuitos para aquisição e condicionamento de sinais para serem utilizados em uma rede de sensores sem fio.

A escolha de componentes foi feita buscando ponto ótimo entre custo e confiabilidade, fator importante em equipamentos eletromédicos.

Conclusão

O projeto apresentado possui o potencial necessário para auxiliar reabilitação de pessoas portadoras de sequelas físicas, contribuindo para melhorar sua qualidade de vida.

As tecnologias e descrições apresentadas neste artigo indicam que o desenvolvimento científico de um equipamento para realizar monitorização dos parâmetros apresentados é possível e realizável.

Agradecimentos

Agradecemos à Universidade Federal de Itajubá, aos Grupos de Engenharia Biomédica e de Microeletrônica pela estrutura e auxílio.

Agradecemos também à CAPES, à FAPEMIG e ao CNPQ pelo fomento.

Referências

- [1] A. Hadjidj, A. Bouabadallah, and Y. Challal (2011), "Rehabilitation supervision using wireless sensor networks", In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1 – 3.
- [2] A. Pantelopoulos, and N. G. Bourbakis (2010), "A survey on wearable sensor-based systems for health monitoring and prognosis", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, v. 40, n. 1, p. 1 – 12.
- [3] R. S. Dilmaghani, H. Bobarshad, M. Ghavami, S. Choobkar, and C. Wolfe (2011), "Wireless sensor networks for monitoring physiological signals of multiple patients", *IEEE Transactions on Biomedical Circuits and Systems*, v. 5, n. 4, p. 347 – 356.
- [4] ADXL325 specifications, Analog Devices, www.analog.com.
- [5] MA series thermistor specifications, GE Thermometrics, www.thermometrics.com
- [6] Pat Lyons, Phil Waterworth (2003), "The Use of NTC Thermistors as sensing devices for TEC controllers and temperature control Integrated Circuits", Application Note Measurements Specialties. Disponível em: www.meas-spec.com/downloads/Thermistor_TEC_application_notes.pdf
- [7] Guyton, A. C. and Hall, J. H. (1997) "O Eletrocardiograma Normal", In: *Tratado de Fisiologia Médica*, 9ª Edição Rio de Janeiro: Guanabara p. 105.
- [8] Neuman, M.R. (1998) "Biopotential Amplifiers", In: *Medical Instrumentation: Application and Design*, Ed.: J. G. Webster, New York: John Wiley and Sons, p. 241 – 292.

PROTOCOLO DE COMUNICAÇÃO DE SINAIS FISIOLÓGICOS TRAFEGANDO EM UMA REDE *ZIGBEE*

G. Leal; J. Siqueira; M. D. Aleixo, Jr.; P. C. Crepaldi e R. Castro, Jr.

Instituto de Engenharia de Sistemas de Tecnologia da Informação / Universidade Federal de Itajubá

guilherme@universopcs.com.br

Abstract: This article seeks to present a protocol that runs on a ZigBee wireless network, allowing data from physiological sensors connected to the body of an individual, storing in the computer that hosts the data. A method is developed for data transmission between sensor and computer, whose main target is to allow the energy consumption, is divided by several devices. As a result we obtain a links at distances reaching more than a thousand meters, without the need for high gain of the transmitter antenna. Through the acquisition of signals from body temperature, electrocardiogram (ECG), a triaxial acceleration, the velocity in real time, we aim to provide subsidies to assess clinical improvement, assisting in the rehabilitation process of persons with physical sequels, between other possible applications.

Palavras-chave: ZigBee, Redes de sensores sem fio, CRC-16, IEEE 802.15.2 WPAN.

Introdução

Amostrar em tempo real sinais fisiológicos de um indivíduo, durante a execução de atividades físicas, abre caminho para a avaliação do processo de reabilitação de pessoas com algum tipo de seqüela [1]. Conceito similar pode ser aplicado para monitorar indivíduos em suas casas, desempenhando atividades esportivas, ou até mesmo, no monitoramento durante atividades profissionais de risco [2].

Tais sinais são amostrados e transmitidos através de um protocolo que trafega sobre uma rede sem fio Zigbee.

O protocolo que estamos desenvolvendo permite identificar uma pessoa com a sua respectiva rede de sensores presos ao corpo. Informações sobre temperatura e ECG, só para citar alguns sinais, podem ser posteriormente armazenados e analisados em um computador pessoal ligado a rede.

O padrão IEEE 802.15.4 WPAN (Wireless Personal Area Network), ao qual a rede Zigbee pertence, possui atributos que convergem para requisitos de nosso projeto: comunicação sem fio entre dispositivos, baixo consumo de energia, complexidade reduzida, custo muito atraente são algumas das características chave, para sistemas de monitoramento [3]. Dentro do cenário apre-

sentado, uma rede ZigBee, se mostra como um meio de transporte ideal para os sinais fisiológicos. O padrão proporciona condições, para o desenvolvimento de redes de sensores e serviços de acompanhamento e controle. Estabilidade adequada e segurança para a aplicação que estamos desenvolvendo são outros fatores determinantes para a escolha do Zigbee[4].

IEEE 802.15.2 WPAN - Este padrão é um subconjunto do modelo OSI. Trabalhando numa gama de projetos particulares, sendo que algumas das características principais são as seguintes: topologias de rede Peer-to-peer e estrela; baixa potência, ampliando de forma razoável a vida útil das baterias; operação de curto alcance, com transmissão de dados via rádio a uma taxa de 250kbps; protocolos simplificados e flexíveis; aplicação de algoritmos CSMA-CD, *Carrier Sense Multiple Access* [3];

As três camadas apresentadas no referido padrão, quando comparadas ao ISO/OSI [3] conectam o aplicativo a camada três (rede), que conectar-se a camada 2 (Data Link) e finalmente, camada 1 (Física). A camada física atinge o hardware responsável pelo link de rádio.

Para o IEEE 802.15.4, a camada 1 oferta dois serviços para os níveis superiores: ativação / desativação do transceptor de rádio; canais de rádio; detecção de nível de energia e qualidade do sinal recebido, permitindo a transmissão e recepção dos pacotes de dados através da banda de frequência de 2,4 GHz[3].

Padrão Zigbee – Construído sobre o padrão IEEE 802.15.4 fornecendo segurança adicional para o serviço disponível na camada MAC, a solução de rede ZigBee preserva inúmeras características do padrão 802.15.4, com a vantagem da padronização dos serviços de nível superior. Essa padronização está disponível sob a forma de API [4], oferecendo vários "perfis de aplicação" com um conjunto padrão de mensagens e comandos para que desenvolvedores escrevam aplicações para transitar os dados pela rede.

Esta norma fornece alta compatibilidade entre fornecedores, sendo que esta característica ajuda os desenvolvedores a integrar itens de diferentes fabricantes, acarretando um forte impacto sobre o custo de um projeto [4].

O conceito piconet - Uma concepção interessante na topologia de comunicação, que utilizaremos amplamen-

te em nosso projeto, é a piconet[4]. Uma piconet consiste no mínimo de dois dispositivos que compartilham o mesmo canal físico. A observação imediata é que ambos os dispositivos são sincronizados para um mesmo sinal de clock.

Cada piconet possui um canal físico diferente e somente um dispositivo atuará como a figura do mestre, em outras palavras, o dispositivo tem como responsabilidade, a sincronização dos outros participantes em que estão inseridos.

Um dispositivo pode participar de mais de uma piconet, mas somente poderá ser mestre de uma.

Scatternet - Diferentes piconets podem trabalhar integradas, sendo esta topologia batizada de scatternet [4]. Um dispositivo inserido em uma scatternet não significa que as capacidades de roteamento de rede estão disponíveis no dispositivo, no entanto, temos acesso a um determinado conjunto de rotinas que representará um papel chave.

Materiais e Métodos

O projeto do protocolo envolve a concepção de três pilares principais: diferentes quadros de dados trafegando pela rede Zigbee, rodando em um sistema com um microprocessador baseado num dispositivo da Texas Instruments, modelo CC253X; um método para rotear os dados buscando minimizar o consumo de energia de todos os dispositivos; uma concepção para detectar a saúde dos dispositivos do sistema.

Compartilhando consumo de energia - Na prática, para atingirmos distâncias de até mil metros, ou mais, na comunicação entre os usuários e o computador pessoal, estamos desenvolvendo em nosso projeto um método, que busca o equilíbrio entre os ganhos nas antenas dos rádios, através de um subsistema que será responsável pela propagação da informação por entre as piconets, num nível superior scatternet, chegando finalmente ao computador pessoal.

Em última instância podemos dizer, que o consumo de energia, de uma comunicação será compartilhada por tantos membros da rede ZigBee que forem necessários, dependendo da localização geográfica de um usuário e o computador.

Formato dos quadros – Nosso protocolo define o formato dos quadros que trafegarão sobre a rede. O comprimento de cada quadro será de 64 bits com uma taxa de 2[ms] por quadro, sendo esta configuração adequada para uso em projetos com o microcontroladores CC2531 / CC2533, escolhidos para o protótipo.

Header (12 Bits)	Sensor 1 (12Bits)	Sensor 2 (12Bits)	Sensor 3 (12Bits)	CRC-16

Figura 1: Formato geral de um quadro de dados

A estrutura geral de cada quadro é também mostrada

na **figura 1**, onde é possível observar três categorias de informações: cabeçalhos, valores digitais a partir de sensores e finalmente dados de verificação de redundância cíclica (CRC).

Cabeçalho - O cabeçalho contém informações para identificar os dados que trafegam pela rede, como pode ser visto na Tabela 1. Os campos são definidos como descrito a seguir: a versão do nosso protocolo, código de identificação do tipo de quadro enviado e, finalmente, a identificação do atleta monitorado.

Tabela 1: Cabeçalho de um Quadro.

Bit	Descrição
0, 1	Versão do Protocolo
2 até 4	Tipo de Quadro
5 até 11	ID do usuário

Os dois primeiros bits, 0-1, do campo de cabeçalho são projetados para permitir atualizações do nosso protocolo, assim, gerações diferentes do sistema podem coexistir na mesma rede, sendo totalmente identificados pelo computador que armazena dados para análise posterior.

O último campo da **Tabela 1** é chamado ID usuário representando, a identificação do indivíduo monitorado. O conjunto de bits, 2 a 4, identifica o tipo de quadro, sendo este valor discriminado na **tabela 2**. O referido campo poderá informar diferentes classes de informações: conjunto de sensores, status de um dispositivo e alerta de um sensor, indicando que uma avaliação e /ou manutenção deve alertar a todos.

Deteção de erros - Os tipos de quadros desenvolvidos permitem o acesso, e armazenamento dos dados de uma forma segura através de algoritmos eficientes para a correção de erros. O algoritmo escolhido, devido à relativa simplicidade de programação é o CRC-16 [5]. Os erros são completamente detectados quando ocorre qualquer evento em um único bit, ou em número ímpar de bits num bloco de dados. Rajadas de erros podem também ser detectadas com uma eficiência ao redor de 98% [5].

Tabela 2: Tipos de Quadro.

Tipo de Quadro	Dados presentes
0	Status, ID e dados de sensores.
1	ECG, Acelerômetros 1 e 2.
2	ECG, Acelerômetros 3 e 4.
3	ECG, Termômetro 1 e 2.
4	Alerta de Sensor

O protocolo ZigBee possui o seu próprio sistema de controle de erro relacionado com a transmissão [4] no entanto, nosso protocolo rodando sobre a referida rede,

possui uma estrutura que visa evitar que dados amostrados incompletos ou inconsistentes, vindos dos conversores A/D do protótipo, utilizando microprocessador CC253X, sejam transmitidos e armazenados no computador, aumentando a precisão da análise a ser executada posteriormente.

Header (12 Bits)	Status Register (8Bits)	Reserved (4 Bits)	Sensor ID (8Bits)	Sensor Level (12Bits)	CRC-16
---------------------	----------------------------	----------------------	----------------------	--------------------------	--------

Figura 2: Quadro tipo 0.

A **Figura 2** mostra o frame 0, que apresenta uma estrutura, que informa o estado de saúde do dispositivo microprocessado e seus sensores. Estamos atualmente identificando um conjunto de parâmetros e desenvolvendo um algoritmo para avaliar as condições do sistema. De acordo com os resultados da referida avaliação, o sistema será capaz de detectar automaticamente se ele pode trabalhar corretamente ou não.

Tabela 3: Id dos Sensores.

Código	Descrição
01	Bateria
11	ECG
21	Acelerômetro 1
22	Acelerômetro 2
23	Acelerômetro 3
24	Acelerômetro 4
31	Termômetro 1
32	Termômetro 2
99	Computador

O Campo Sensor ID pode assumir os valores mostrados na **Tabela 3**. Para cada sensor poderemos recuperar o valor respectivo do Registro de Status, conforme é apresentado na Tabela 4.

Tabela 4: Registro de Status.

Código	Descrição
00	OK
01	Checar Sensor/Dispositivo
100	Falha do computador
200	Falha de comunicação com o sensor

O software no computador pessoal pode fornecer feedback para o operador, oferecendo subsídios para a tomada de decisão se algum tipo de manutenção no sistema, caso seja necessário.

Header (12 Bits)	Sensor 1 (12Bits)	Reserved (12Bits)	Reserved (12Bits)	CRC-16
---------------------	----------------------	----------------------	----------------------	--------

Figura 3: Formato do quadro tipo 4.

Resultados Esperados

Esperamos testemunhar o quanto é eficiente e eficaz o nosso protocolo, bem como determinar estratégias para eventuais aprimoramentos ampliando, o leque de aplicações.

Discussão

Este artigo propõe um protocolo de comunicação para sistemas sem fio, monitorarem sinais fisiológicos, sem a limitação dos movimentos dos usuários. Os estudos mostram que o projeto é viável, podendo evoluir para o nascimento de um produto comercial.

Conclusão

O protocolo proposto, integrado a um sistema microprocessado de monitoramento de sinais fisiológicos, apresenta diversas utilidades que vão do auxílio em reabilitação motora de pessoas com sequelas físicas, monitoramento de pessoas em ambientes insalubres ou avaliação de desempenho físico de usuários de um modo geral. Um consequente vínculo entre este trabalho e a melhoria da qualidade de vida de indivíduos assistidos, é algo que esperamos alcançar.

Agradecimentos

Agradecemos à Universidade Federal de Itajubá, aos Grupos de Engenharia Biomédica e Microeletrônica pela estrutura e auxílio.

Referências

- [1] Jarochoowski, B.P.SeungJung Shin; DaeHyun Ryu ; HyungJun Kim (2007), "Ubiquitous Rehabilitation Center: An Implementation of a Wireless Sensor Network Based Rehabilitation Management System", In: *Convergence Information Technology, 2007. International Conference on*, pp. 2349 - 2358.
- [2] Silva, Joao C. Arsenio, Artur M.; Garcia, Nuno M. (2011), "Context-awareness for mobility management: A systems survey for healthcare monitoring", In: *Broadband and Biomedical Communications (IB2Com), 6th International Conference on*, pp. 18 – 23.
- [3] IEEE 802.15™: "Wireless personal area networks (WPAN's)". Disponível em <http://standards.ieee.org/about/get/802/802.15.html>. Acessado em 18 de Jun. 2011.
- [4] Zigbee Alliance; "Zigbee Alliance Standards". Disponível em [http:// www.zigbee.org/Standards/Overview.asp](http://www.zigbee.org/Standards/Overview.asp). Acessado em 18 de Jun. 2011.
- [5] S. Shukla, and Bergmann (2004), "Single bit error correction implementation in CRC-16 on FPGA", In: *Proceedings of IEEE International Conference on Field-Programmable Technology*, pp. 319–322.

