

TRABALHO DE GRADUAÇÃO

EQUIPAMENTO PARA ANÁLISE DO SISTEMA NERVOSO AUTÔNOMO COMPOSTO DE *HARDWARE* E *SOFTWARE*

**Marina Ladeira Cotta
Renata Silva Coêlho**

Brasília, 03 de julho de 2008

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**EQUIPAMENTO PARA ANÁLISE
DO SISTEMA NERVOSO AUTÔNOMO
COMPOSTO DE *HARDWARE* E *SOFTWARE***

**MARINA LADEIRA COTTA
RENATA SILVA COELHO**

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Adson Ferreira da Rocha,
PhD, ENE/UnB, (Orientador)

Prof. Janaína Gonçalves Guimarães,
Doutora, ENE/UnB

Prof. Ricardo Zelenovsky,
Doutor, ENE/UnB

Dedicatória

Dedicamos este trabalho a todos que contribuíram para o seu desenvolvimento, principalmente à professora Janaína Gonçalves Guimarães que nos acompanhou durante todo o curso de Engenharia Elétrica, sendo fundamental para a nossa formação.

Agradecimentos

Agradeço à minha família, pelo carinho, principalmente a meus pais, Gilberto e Ana Rita, por todo amor e apoio e a meu irmão, Eduardo, que proporcionou mais alegria a esta caminhada. Agradeço a Deus, por me dar força e determinação e a Renata, que se tornou tão amiga e companheira nesses cinco anos e, sobretudo, durante a execução deste trabalho de graduação. Muito obrigada!

Marina Ladeira Cotta

Agradeço a Deus; a meus pais, Raimundo e Roza Maria, que sempre acreditaram em mim e me ensinaram o quanto a educação é um bem importante; ao Rodrigo e a Vera por contribuírem com toda força e incentivo nesta caminhada. Agradeço também a Marina pela amizade, por todo apoio nas atividades do curso de Engenharia Elétrica e principalmente em todo o projeto de graduação.

Renata Silva Coêlho

Agradecemos ao professor e orientador, Adson Ferreira da Rocha – que nos ajudou em todos os passos do projeto de graduação, sempre bem humorado e disposto a solucionar nossas dúvidas e problemas – contribuindo não somente com nosso crescimento intelectual, mas também pessoal.

Agradecemos também aos professores Ícaro dos Santos e Ricardo Zelenovsky pela contribuição dada a este trabalho.

Marina e Renata

RESUMO

Este trabalho apresenta o desenvolvimento de uma ferramenta para análise do Sistema Nervoso Autônomo. Ela consiste em *hardware*, que permite a captura e transformação de sinais de eletrocardiograma, temperatura e resistência galvânica da pele; e *software*, que adquire os dados transformados por meio da porta USB e mostra-os na tela do computador. A análise simultânea desses parâmetros possibilita verificar se a alteração de cada um deles está relacionada ao Sistema Nervoso Autônomo ou se ocorre devido a fatores distintos. Este projeto teve como objetivo fortalecer os conhecimentos teóricos e práticos adquiridos no curso de Engenharia Elétrica demonstrando-os através da construção do equipamento proposto.

ABSTRACT

This work shows the development of a tool to be used on the analysis of the Autonomic Nervous System. It contains a hardware – that allows the capture and transformation of electrocardiogram, galvanic skin response and temperature signs – and a software, which takes the transformed data by the USB gate and shows it on a computer's screen. The simultaneous analysis of those parameters allows one to verify if the changes on them are related with the Autonomic Nervous System or due other causes. The goal of the present project was to strengthen all the knowledge acquired on the Electric Engineering course, showing them through the creation of the proposed equipment.

SUMÁRIO

	Página
1 INTRODUÇÃO	1
2 SISTEMA NERVOSO AUTÔNOMO	3
2.1 TECNOLOGIAS PARA MEDIÇÃO.....	3
2.1.1 OBTENÇÃO DO SINAL DO BATIMENTO CARDÍACO.....	4
2.1.2 MEDIÇÃO DE RESISTÊNCIA GALVÂNICA DA PELE.....	4
2.1.3 MEDIÇÃO DE TEMPERATURA	5
3 HARDWARE	7
3.1 MATERIAL E MÉTODO	7
3.1.1 MATERIAL UTILIZADO	7
3.1.2 PROCEDIMENTOS	7
3.2 PROJETO E DESENVOLVIMENTO.....	8
3.2.1 ELETROCARDIOGRAMA.....	8
3.2.2 RESISTÊNCIA GALVÂNICA DA PELE	12
3.2.2 TEMPERATURA	15
4 SOFTWARE	17
4.1 MATERIAL E MÉTODO.....	18
4.1.1 MATERIAL UTILIZADO	18
4.1.2 PROCEDIMENTOS	18
4.1.2.1 AQUISIÇÃO DOS DADOS.....	18
4.1.2.2 CÓDIGO DE PROGRAMAÇÃO.....	19
5 MEDIÇÕES E ANÁLISE	24
6 CONCLUSÕES	28
REFERÊNCIAS BIBLIOGRÁFICAS	29
ANEXOS	30

LISTA DE FIGURAS

2.1	Forma padrão da onda de um batimento cardíaco	4
2.2	Representação do triangulo de Einthoven	5
3.1	Esquemático do circuito ECG	8
3.2	Circuito que compõem o primeiro bloco, INA 118	8
3.3	Representação do circuito do eletrocardiograma	10
3.4	Representação do circuito que mede a resistência galvânica da pele	13
3.5	Gráfico da Resistência galvânica da pele versus a Tensão medida	14
3.6	Representação do circuito que mede a temperatura	15
4.1	Interface do Programa desenvolvido	17
4.2	DLP-IO8-G - Dispositivo de aquisição de dados	18
4.3	Funcionamento do programa dividido em blocos	19
4.4	Fluxograma relativo à inicialização do programa	20
4.5	Fluxograma relativo à simulação	20
4.6	Propriedades da porta serial	21
4.7	Fluxograma relativo à aquisição de dados	22
4.8	Fluxograma relativo ao desenho nos gráficos	23
5.1	Sinal de ECG antes de passar pelo circuito limitador de tensão	24
5.2	Forma da onda de batimento cardíaco. (Figura 2.1)	25
5.3	Sinal de ECG após passar pelo circuito limitador de tensão.	25
5.4	Onda triangular de freqüência 15Hz	26
5.5	Onda triangula de freqüência 5Hz dados	26
5.6	Dados de ECG adquiridos e plotados no programa	26
5.7	Dados de ECG simulados e plotados no programa	27
A.1	Trilha do circuito impresso criada no TraxMaker 3 Pro	29
A.2	Vista superior do <i>hardware</i>	30
A.3	Vista frontal do equipamento fechado	31
A.4	Cabos e conectores do eletrocardiograma em forma trançada	32
A.5	Cabos e eletrodos de medição da resistência da pele	32
A.6	Cabos e termistor para a medição da temperatura	32
A.7	Cabos e conectores montados para a condução de sinal	32

1 INTRODUÇÃO

O trabalho apresentado teve início em um estudo dos principais sintomas e características provenientes de alterações físicas ou psíquicas no corpo humano. A parte do Sistema Nervoso que tem a função de manter estável o organismo frente às necessidades de adaptação aos meios externos e internos é o chamado Sistema Nervoso Autônomo. Tal sistema controla funções como circulação, respiração, temperatura, digestão e outras, estando diretamente relacionado com a variação da resistência galvânica da pele, dilatação das pupilas, contração do intestino, aceleração cardíaca e respiratório entre outros. De todos os diversos fatores passíveis de alteração pelo Sistema Nervoso Autônomo foram escolhidos para análise a variação dos batimentos cardíacos, da temperatura corporal e da resistência galvânica da pele.

O eletrocardiograma, exame utilizado por profissionais da área de medicina para detectar alterações no ritmo do batimento cardíaco, foi o escolhido para implementação. A tensão correspondente à variação cíclica da quantidade de íons de sódio relativos à atividade elétrica do coração é capturada por eletrodos, que serão colocados em pontos específicos do corpo. O equipamento construído medirá as alterações de potencial elétrico devido à despolarização e repolarização das células cardíacas.

A resistência da pele, proteção do corpo humano contra qualquer alteração de estado, pode ser obtida ao se passar uma pequena corrente galvânica, contínua, entre dois dedos da mão do paciente e medir a diferença de tensão entre eles.

A temperatura, também alterada em qualquer mudança de estado do corpo humano, é medida a partir de um circuito acoplado a um termistor que relaciona a resistência do mesmo a uma determinada temperatura. Assim, medindo a voltagem obtida nos terminais desse resistor estima-se a temperatura do corpo naquele momento.

O objetivo desse trabalho é construir um sistema completo – *hardware* e *software* – de captura de dados e plotagem dos sinais de eletrocardiograma, temperatura e resistência galvânica da pele, para então permitir uma análise do Sistema Nervoso Autônomo.

O *hardware* foi construído sobre uma placa de circuito impresso e é composto basicamente de resistores, capacitores, amplificadores operacionais, além dos sistemas de aquisição de dados e de isolamento. Quando conectado a determinados pontos no indivíduo, os sensores permitem que o equipamento capture os sinais descritos anteriormente. O *software* foi desenvolvido na linguagem *Visual Basic* e permite que os sinais sejam adquiridos por meio da porta USB e mostrados na tela do computador. As técnicas de medidas e análise são mostradas e comentadas de forma a facilitar o entendimento de utilização do equipamento.

O capítulo 2 trata do Sistema Nervoso Autônomo, o alvo do presente projeto, explicando suas características e a tecnologia usada para a medição do mesmo.

No capítulo 3 é explicado em detalhes o projeto, planejamento, criação e montagem do *hardware* e no capítulo 4 o mesmo é feito para o *software*.

A descrição dos resultados obtidos e a análise do desempenho do equipamento são feitas no capítulo 5 e as conclusões acerca de todo o projeto são descritas no capítulo 6.

2 SISTEMA NERVOSO AUTONOMO

Este capítulo explica o sistema humano a ser estudado e as características a serem medidas.

Elementos indispensáveis à vida são tratados em diversas teses e debates. Algumas dessas partes são, de certa forma, deixadas de lado sem perder sua importância no equilíbrio do Ser Humano. Nosso corpo é formado por um sistema responsável por tomada de decisões, transporte de sensações, envio de informações e ordens, o chamado Sistema Nervoso é em grande parte relevante à vida.

A divisão em duas partes possibilita o melhor funcionamento de todo complexo nervoso. O Sistema Nervoso Central recebe, analisa e integra informações. A sua localização é no encéfalo e na medula espinhal, onde ocorre a tomada de decisões e o envio de ordens. Já o Sistema Nervoso Periférico, composto de nervos e gânglios, carrega informações dos órgãos sensoriais para o sistema nervoso central e deste para os órgãos efetores, músculos e glândulas.^[10]

É no Sistema Nervoso Periférico que se situa o importante alvo desse estudo, o Sistema Nervoso Autônomo. Este é o principal responsável pelo controle automático do corpo, respostas reflexas, frente às diversidades do ambiente, sendo responsável pelo controle da musculatura lisa, da musculatura cardíaca e das glândulas exócrinas. É também composto por duas porções distintas, cujas ações são antagônicas. As fibras nervosas dos sistemas Simpático e Parassimpático secretam principalmente dois principais neurotransmissores: noradrenalina ou acetilcolina.^[10]

O sistema Simpático estimula ações que mobilizam energia, permitindo ao organismo responder a situações de estresse, medo e nervosismo para uma luta ou uma fuga. As respostas desencadeiam ações como o aumento da adrenalina e do metabolismo corporal, a aceleração dos batimentos cardíacos, a dilatação dos brônquios, o aumento da pressão arterial, a concentração de açúcar no sangue e a dilatação das pupilas.

O sistema Parassimpático evidencia atividades relaxantes. As ações características são, normalmente, contrárias às do sistema Simpático. As duas vertentes atuam normalmente em simultâneo sendo do equilíbrio entre a força de ação de cada uma delas que nasce a extrema capacidade regulatória do Sistema Nervoso Autônomo, essas ações estendem-se a diversos domínios biofisiológicos do nosso organismo.

2.1 TECNOLOGIA PARA MEDIÇÃO

Como o próprio nome sugere, Autônomo, o sistema estudado não pode ser voluntariamente controlado, portanto, para analisar se o Simpático ou o Parassimpático está atuando em prioridade, deve-se analisar as características sintomáticas produzidas por cada um desses subsistemas.

A frequência cardíaca é um importante fator de análise não só de patologias, mas principalmente da situação psicológica em que o indivíduo se encontra. A atuação do Simpático acelera o batimento cardíaco e a do Parassimpático reduz o mesmo.

A temperatura do corpo humano também é um importante parâmetro para estudo dos sistemas Simpático e Parassimpático. Ela se adequa ao tipo de situação externa vivida pelo indivíduo.

Outra característica sintomática da atuação do sistema nervoso autônomo é a resistência galvânica da pele. Em uma situação de estresse seu valor é diminuído pelo Simpático; da mesma forma, em uma situação de tranquilidade, o Parassimpático aumenta o mesmo.

Dessa forma, as tecnologias utilizadas no projeto compreendem as três avaliações mencionadas.

2.1.1 OBTENÇÃO DO SINAL DO BATIMENTO CARDÍACO

O eletrocardiograma é o registro dos fenômenos elétricos que se originam durante a atividade cardíaca por meio de um aparelho denominado eletrocardiógrafo. Este é composto por um galvanômetro que mede a intensidade de corrente que é recolhida a partir de eletrodos dispostos em determinados pontos do corpo humano.

A atividade normal de um coração é composta pela seguinte forma de onda:

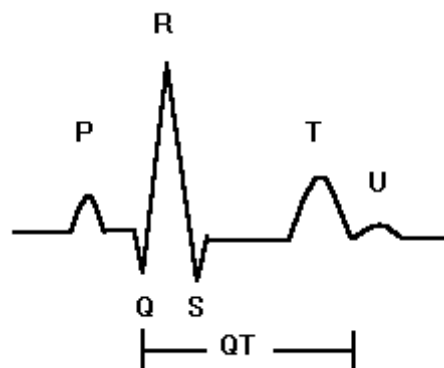


Figura 2.1: Forma padrão da onda de um batimento cardíaco.

O médico alemão *Einthoven* foi o imaginador do coração como um triângulo equilátero cujos vértices são representados pelo braço direito, braço esquerdo e perna esquerda do ser humano. A partir da determinação, surgiu uma técnica de medidas de eletrocardiograma por meio de derivações bipolares. Com base na Segunda Lei de *Kirchoff*, a soma das correntes em um mesmo nó resulta em zero, a soma dessas derivações em um mesmo triângulo resulta em zero. Sendo essas derivações a relação de um ponto a outro, onde são colocados os eletrodos.^[4]

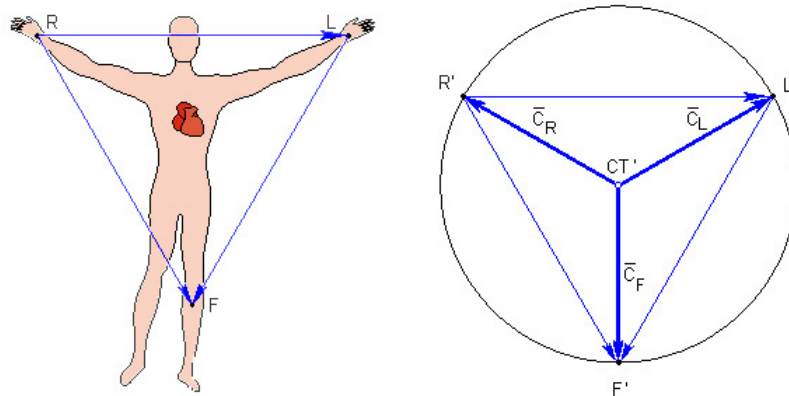


Figura 2.2: Representação do triângulo de *Einthoven*.

2.1.2 MEDIÇÃO DE RESISTÊNCIA GALVÂNICA DA PELE

A pele de um indivíduo possui certa resistência, ou seja, elementos opostos à condução de corrente elétrica. Denominada de resistência galvânica da pele, ou até mesmo de impedância cutânea, o obstáculo se torna maior para correntes de baixa frequência.

A variação de fatores como temperatura, pilosidade, gordura, espessura da pele, suor e umidade alteram diretamente a resistividade da pele. Dessa forma, qualquer atividade característica do sistema nervoso autônomo pode ser percebida por meio das medidas dessa resistência.

O equipamento desenvolvido emite, através de eletrodos, corrente elétrica, entregando energia eletromagnética à pele. Da mesma forma, ele recebe essa energia e, em forma de tensão elétrica, analisa as perdas associando-as a resistência contida na pele humana.

2.1.3 MEDIÇÃO DE TEMPERATURA

A temperatura é a propriedade da matéria que reflete a média de energia cinética das moléculas de um corpo, ou seja, está relacionada com o movimento dos átomos de uma substância. As formas de vibrações variam de acordo com essa energia cinética e com o deslocamento das moléculas. Quanto mais rápido esse movimento, mais quente se encontra o corpo, e quanto mais lento o movimento, mais frio o mesmo se apresenta. Esta condição pode ser descrita como um potencial térmico ou como uma energia efetiva da substância. Na

prática a temperatura é representada em uma escala numérica, onde, quanto maior o seu valor, maior é a energia cinética média dos átomos do corpo em questão. ^[4]

A mudança de temperatura afeta diretamente qualquer tipo de processo alterando o comportamento do corpo humano. A medição da mesma é feita com base na mudança dos efeitos elétricos ou físicos produzidos sobre os tecidos, órgãos e sistemas corpo. Sobre a pele é colocado um termistor que permite captar a energia proveniente desses efeitos elétricos e transmiti-la ao circuito principal. Este circuito relaciona a quantidade de energia recebida com os valores pré-estabelecidos pela escala térmica. ^[4]

3 HARDWARE

Este capítulo explica o projeto, planejamento, criação e montagem do circuito físico.

3.1 MATERIAL E MÉTODO

3.1.1 MATERIAL UTILIZADO

- *Protoboard*, fonte, osciloscópio e multímetro para testes;
- Circuitos integrados: TL 064 e INA 118;
- Baterias 9V, resistores, capacitores e diodos;
- Reguladores de tensão: MC 7905 e MC 7805;
- Placa de fibra de vidro coberta por fina camada de cobre;
- Material para confecção da placa de circuito impresso;
- *Software TraxMaker 3 Pro*;
- Material para soldagem e construção da caixa externa.

3.1.2 PROCEDIMENTOS

O projeto apresentado exigiu a construção de um dispositivo físico e eletrônico, o *hardware*, que permitisse a captura dos sinais de resistência galvânica da pele, de temperatura e de batimentos cardíacos.

Primeiramente, para criá-lo, foi elaborado um projeto de cálculos e gráficos que permitissem capturar, amplificar e disponibilizar os sinais interessantes ao projeto sem corrompê-los. Dessa forma, este projeto foi testado e aprimorado em laboratório.

Criado e aprovado o projeto do *hardware*, o programa *TraxMaker* foi utilizado para elaborar todo o circuito em forma de uma trilha. O desenho foi impresso em uma placa coberta por uma fina camada de cobre. A parte negativa à trilha é corroída, deixando o cobre formar o caminho por onde passa a corrente e os componentes do circuito. Em formato de circuito impresso, e os componentes são soldados à mesma.

Por fim, foi criada uma caixa externa que abrange o circuito com todas as entradas e saídas. Também foram feitos os cabos e conectores responsáveis pela condução dos sinais entre o corpo humano e o circuito.

Fotos detalhadas do *hardware* assim como o desenho da placa encontram-se presentes no Anexo I.

3.2 PROJETO E DESENVOLVIMENTO

3.2.1 ELETROCARDIOGRAMA

O projeto da placa física que forma todo o circuito proposto é dividido em duas etapas. Foi construída primeiramente uma placa com o circuito que compõe o analisador de batimentos cardíacos, o medidor de resistência da pele e o medidor de temperatura. Logo em seguida foi feita uma placa que limita a tensão, entre 0V e 5V, de cada saída do circuito para que pudessem ser corretamente adquiridas pelo *software*.

Para uma melhor compreensão, o circuito correspondente ao ECG é representado em blocos, na figura 3.1, abaixo:

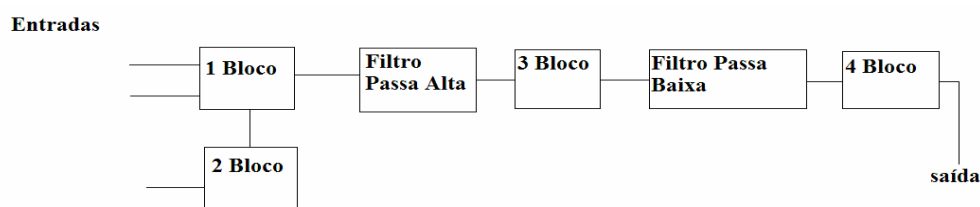


Figura 3.1: Esquemático do circuito ECG.

O primeiro bloco corresponde à entrada ligada diretamente na parte superior do paciente. São dois eletrodos como entradas que capturam diferentes frequências do batimento cardíaco. Dentro deste primeiro bloco ocorre a derivação de uma entrada pela outra. Para isso, é utilizado um circuito com amplificadores que fornecem alto ganho e alta largura de banda.

Em biomédica existem amplificadores de grande importância que propiciam precisão, acurácia em altas larguras de banda e ganho. O circuito integrado INA 118 é bastante utilizado e considerado em aplicações de engenharia biomédica, já que rejeita ruídos relativos à 60Hz, conduzidos pelo corpo humano. Com alto fator atuante, o INA 118 foi usado no circuito de ECG como um amplificador de instrumentação, em substituição ao circuito mostrado na figura 3.2.^[8]

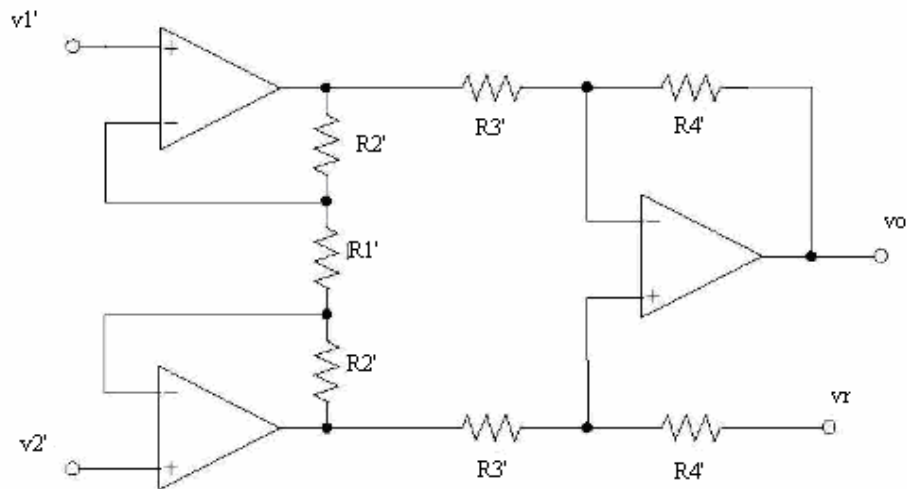


Figura 3.2: Circuito que compõem o primeiro bloco, INA 118.

A expressão resultante ao circuito proposto dentro do INA 118 é:

$$v_O - v_R = \frac{R_4'}{R_3'} \cdot \left(1 + 2 \cdot \frac{R_2'}{R_1'} \right) \cdot (v_1' - v_2')$$

O circuito implementado apresenta alta impedância de entrada e alto RRMC. O ganho foi ajustado com as resistências propostas entre as entradas inversora e não inversora: $R_G/2 = 2,7k\Omega$. Dessa forma, o INA 118 é composto por:

- dois terminais de alimentação, sendo um positivo (a ser alimentado com 5V) e um negativo (no qual é ligado à tensão constante de -5V);
- dois terminais de entrada, sendo um positivo, V_{in}^+ , e um negativo, V_{in}^- ;
- um terminal de saída, cuja tensão é dada por $V_s = A (V_{in}^+ - V_{in}^-)$.

Como o sistema é alimentado por baterias de 9 volts, para se manter uma fonte constante de 5 volts na entrada do sistema e também na alimentação positiva do amplificador INA 118P é usado os reguladores de tensão 7805 e LP2905 (*National Semiconductor*). Já para garantir uma tensão constante de -5 volts na alimentação negativa do amplificador é usado o regulador 7905.

Dessa forma, em cada entrada, inversora e não inversora, foram adaptadas resistências $R_1 = R_2 = 2,7k\Omega$. Essas resistências são ligadas em um mesmo ponto de saída, que está conectado à entrada do segundo bloco.

O segundo bloco corresponde à terceira entrada, que está ligada à parte inferior esquerda do corpo humano. É composta por um eletrodo que captura um sinal de frequência cardíaca como referência de todo o circuito. Este sinal, dentro do circuito, é derivado, por meio de

amplificadores e resistências, em relação às duas primeiras entradas. Obtendo essas derivações em conjunto, pode ser obtido pelo circuito um sinal da forma e frequência da onda do batimento cardíaco.

Este bloco é composto por um amplificador, utilizando o circuito integrado TL 064, que propicia alto ganho. Este amplificador tem geralmente em seu retorno a perna esquerda ou a parte de baixo das costelas do paciente como referência. Para este ramo foi colocada uma resistência de proteção, no valor de $R_5 = 390k\Omega$, que impede o retorno de alta tensão para o corpo do ser humano.

No mesmo amplificador, os resistores foram considerados no cálculo do ganho.

$$A = \frac{R_4}{R_3} = \frac{390k}{10k} = 39V/V$$

Em seguida ao primeiro e segundo blocos, foi colocado um filtro passa-alta com frequência de corte de 0,1 Hz. A principal função deste é eliminar tensões DC.

Cálculo o filtro passa alta:

$$f_c = 0,1Hz$$

$$\frac{1}{R_6 C_1} = 2 \cdot \pi \cdot f$$

$$R_6 C_1 = 1,59$$

$$R_6 = 159k\Omega$$

$$C_1 = 10\mu F$$

A resistência encontrada foi dividida em dois valores de resistência comerciais em série: $R_6' = 150k\Omega$ e $R_6'' = 10k\Omega$.

O terceiro bloco corresponde ao ganho do sinal que é capturado na ordem de milivolts e precisa ser amplificado. Para isto, está sendo utilizado um amplificador operacional e uma chave que altera o valor das resistências e, conseqüentemente, o valor do ganho. O terminal não inversor do amplificador é ligado ao circuito e o terminal inversor é ligado em paralelo à resistência de $R_7 = 1k\Omega$ e à chave. Na chave, R_8 , estão alternadas as resistências de $50k\Omega$, $100k\Omega$, $150k\Omega$ e $200k\Omega$, proporcionando ganhos de 50 V/V, 100 V/V, 150 V/V e 200 V/V, respectivamente.

Após o terceiro bloco e antes do quarto, foi implementado um filtro passa-baixas com frequência de corte de 20 Hz. O filtro é utilizado para não deixar que a faixa de frequência a 60 Hz da rede elétrica influencie no sinal obtido pelo batimento cardíaco, além de outros ruídos provenientes do ambiente externo e do próprio corpo. Não foi escolhida uma frequência de corte de menor valor para que o sinal de saída do circuito não fosse deformado. O cálculo dos componentes que formam o filtro encontra-se abaixo:

$$f_c = 20\text{Hz}$$

$$R_9 C_2 = \frac{1}{2 \cdot \pi \cdot f_c} = 7,9\text{m}$$

$$R_9 = 79,6\text{k}\Omega$$

$$C_2 = 1\mu\text{F}$$

O quarto bloco especificado é um sistema estabilizador do circuito. Para isso foi utilizado um *buffer*, sistema no qual a saída realimenta a entrada inversora do amplificador operacional sem qualquer resistência, ou seja, não possui ganho; permitindo, contudo, um bom isolamento do circuito a possíveis interferências. Para este, utilizamos o circuito integrado TL064. [2]

Na figura 3.3 é possível visualizar e analisar o circuito com os seus elementos e componentes.

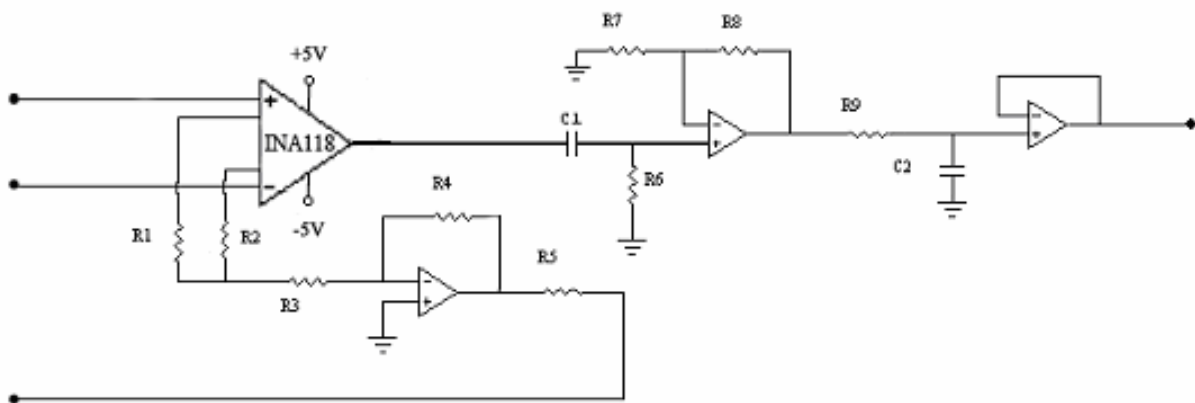


Figura 3.3: Representação do circuito do eletrocardiograma

O circuito mostrado na figura 3.3 já pode fazer a medição como um eletrocardiógrafo, pois possui todos os itens exigidos: amplificação, derivação, filtragem e proteção. No entanto, para que este processo seja capturado por um *software*, utilizaremos um equipamento que somente permite em sua entrada sinais dentro da faixa de 0 e 5 volts. Por isso, como o sinal do eletrocardiograma varia entre $\pm 5\text{V}$, a segunda etapa é imprescindível e consiste em elaborar um circuito limitador da tensão de saída.

Portanto foi realizado um circuito que envolve um amplificador TL 064, seguido por um resistor e, limitando as tensões se encontram 2 diodos. O primeiro diodo captura para si correntes cujas tensões são menores que 0 volt, ou seja, só permite passagem de tensões positivas. O segundo captura para si correntes cujas tensões são maiores que 5 volts, ou seja, só permite passagem de tensões menores que esse valor. Esta foi uma forma simples e eficiente forma encontrada para limitar a tensão de saída e não danificar o equipamento.

Além dos dois filtros usados, para evitar que a onda de frequência 60 Hz interfira no sinal de ECG, os cabos que capturam esse sinal do paciente, presos a conectores do tipo “jacaré” devem ser trançados (figura A.4), criando assim uma blindagem ao sinal de qualquer fonte externa, 60Hz, presente dentro das paredes, cabos que alimentam estabelecimentos com energia elétrica. Isto para que não haja interferência no sinal de ECG.

3.2.2 RESISTÊNCIA GALVÂNICA DA PELE

O princípio básico usado para medir a resistência galvânica da pele é, por meio de eletrodos (figura A.5) colocados nos dedos do indivíduo, fazer passar uma pequena corrente e medir o valor de tensão entre eles, encontrando dessa forma o valor da resistência.

Considerando que a corrente máxima permitida a passar pelos dedos de uma pessoa seja de $50 \mu A$, temos então que garantir que tal corrente não seja ultrapassada. Para isso, é colocado um resistor de proteção em série com a resistência a ser medida, separando a mesma da fonte de tensão. Considerando uma tensão constante de 5 volts, o resistor de proteção deve ser de no mínimo $R_p = 5V/50\mu A = 100k\Omega$.

Dado um valor tão pequeno de corrente, a tensão entre os terminais dos eletrodos é bastante pequena e, para que possa ser mais bem observada, deve ser amplificada. O amplificador usado é o mesmo utilizado no eletrocardiograma, o INA118P, de baixa potência e freqüentemente usado em aplicações de instrumentação biomédica.

Novamente o INA 118 teve os seus terminais de alimentação positivo e negativo mantidos com 5 e -5 volts, respectivamente. O sinal de entrada positivo é chamado V_{in}^+ e o negativo, V_{in}^- . A saída é dada pela tensão $V_s = A (V_{in}^+ - V_{in}^-)$. Por último, mas muito importante, o ganho A é regulado pelo resistor R_{ga} da forma $A = 1 + 50k\Omega / R_{ga}$.

Para a tensão de saída variar entre os limites desejados – 0 e 5 volts –, utiliza-se um divisor de tensão formado pelos resistores R_1 e R_2 para garantir uma tensão V_{in}^- tal que esses limites sejam obedecidos.

No circuito são utilizados reguladores de tensão, que permitem a regulação da tensão das baterias de 9 volts para a tensão constante desejada. Foi utilizado para garantir a tensão constante de 5 volts que alimenta toda a placa os reguladores 7805 e LP2905 e, para tensão negativa de -5 volts, o regulador 7905.

Dessa forma, foi montado o circuito mostrado na figura 3.4.

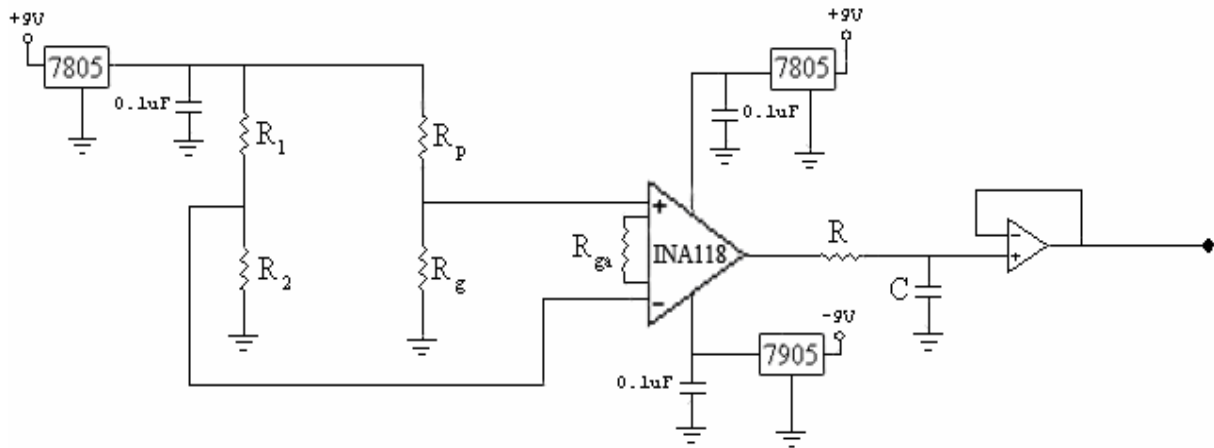


Figura 3.4: Representação do circuito que mede a resistência galvânica da pele.

Do circuito observamos a seguinte relação entre o valor da resistência da pele, R_g , e a tensão nos seus terminais, V_g :

$$V_g = \frac{R_g}{R_g + R_p} \cdot 5$$

Como o valor de R_g varia entre 50Ω e $1,5M\Omega$:

$$V_g^{\text{máx}} = \frac{1,5 \cdot 10^6}{100 \cdot 10^3 + 1,5 \cdot 10^6} \cdot 5 = 4,6875V$$

$$V_g^{\text{mín}} = \frac{50}{100 \cdot 10^3 + 50} = 0,0025V$$

Assim, V_g deve variar entre $0,0025V$ e $4,6875V$. Sendo a tensão de saída especificada para variar entre os limites de 0 e 5 volts, temos que:

$$\frac{V_g - V_g^{\text{mín}}}{V_g^{\text{máx}} - V_g^{\text{mín}}} = \frac{V_s - 0}{5 - 0}$$

$$V_s = 5 \cdot \frac{V_g - 0,0025}{4,6875 - 0,0025} = 1,0672 \cdot V_g - 0,002668$$

Como $V_s = A(V_{in}^+ - V_{in}^-)$, onde $V_{in}^+ = V_g$, temos que $A=1,0672$ e $V_{in}^- = 0,0025V$. Assim, pela fórmula do ganho, $1,0672 = 1 + 50k\Omega / R_{ga}$, logo $R_{ga} = 744k\Omega$. E pelo divisor de tensão formado por R_1 e R_2 temos que, para o valor de V_{in}^- encontrado, $R_1 = 1999 \cdot R_2$. Fixando $R_2 = 100k\Omega$ temos que $R_1 \approx 2M\Omega$.

A equação que relaciona o valor da tensão de saída com o da resistência da pele é

$$V_s = 5,336 \cdot \frac{R_g}{R_g + 100k} + 0,002688 \text{ e apresenta o gráfico mostrado a seguir.}$$

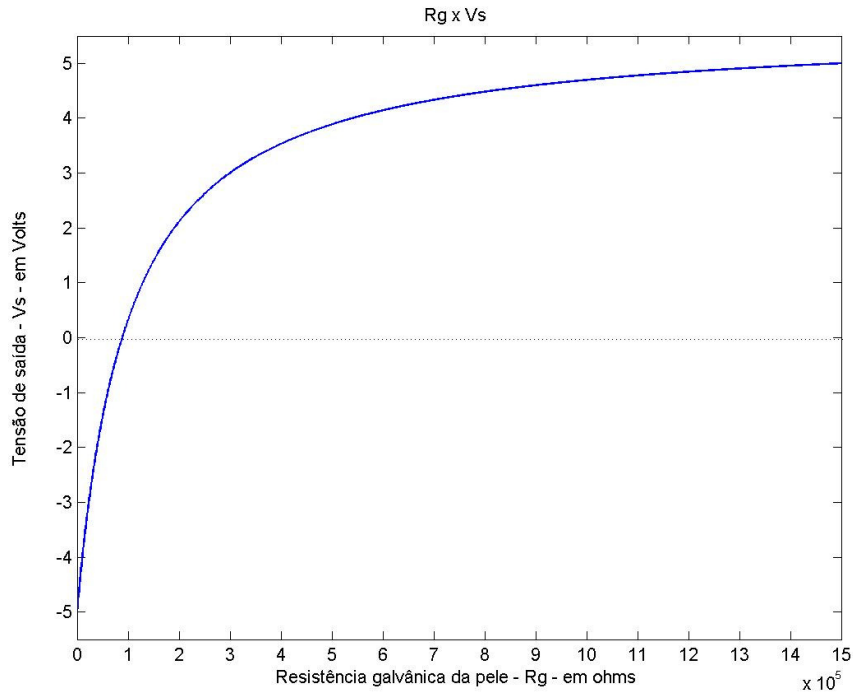


Figura 3.5: Gráfico da Resistência galvânica da pele versus a Tensão medida.

Como pode ser observado, para valores superiores a $5 \cdot 10^5 \Omega$, a tensão de saída varia muito pouco mesmo com grandes variações do valor da resistência da pele.

Para melhorar a qualidade do sinal de saída do sistema, considerando que sua variação é lenta, foi adicionado após o amplificador INA118 um filtro passa-baixa de 5Hz, cujas componentes têm o cálculo especificado a seguir:

$$f_c = 20Hz$$

$$RC = \frac{1}{2 \cdot \pi \cdot f_c} = 31,85m$$

$$R_9 = 32k\Omega$$

$$C_2 = 1\mu F$$

Em seguida ao filtro, um *buffer* foi feito utilizando o amplificador operacional TL064. Este sistema não proporciona ganho, porém favorece a estabilidade e equilíbrio do sistema, isolando-o de possíveis interferências.

3.2.3 TEMPERATURA

Um dispositivo muito comum para a medição de temperatura é o termistor (figura A.6). Ele é composto de uma extremidade sensível que varia sua resistência de acordo com a mudança de temperatura do ambiente em sua superfície. Assim, é montado um circuito (figura 3.6) muito similar àquele usado para medir a resistência da pele, no entanto, no lugar desta é inserido um termistor de resistência variável R_T . Além disso, são recalculados os valores dos resistores R_{ga} , R_1 e R_2 .

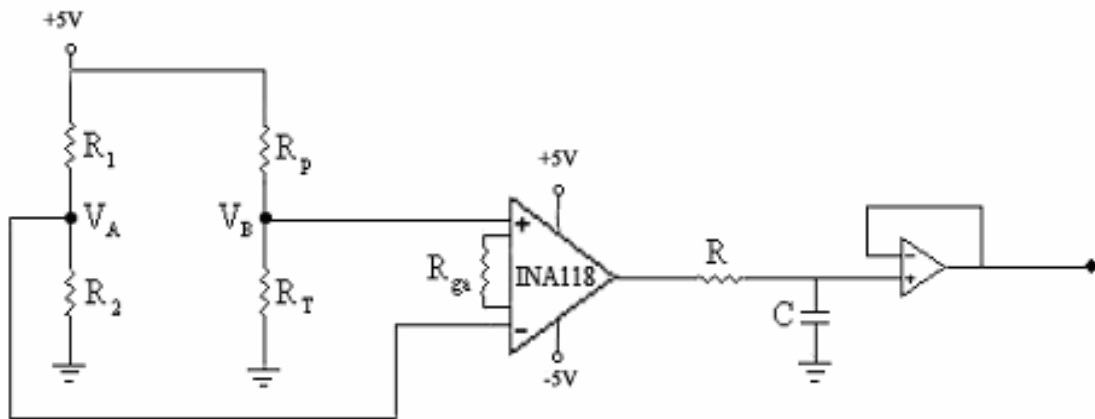


Figura 3.6: Representação do circuito que mede a temperatura.

O termistor escolhido foi o RS NTC10. Estudando o seu *Datasheet* e fazendo sua calibração encontramos a seguinte relação: $R_{T_2} = R_{T_1} \cdot e^{B \left(\frac{1}{T_2} - \frac{1}{T_1} \right)}$. Considera-se, pela tabela característica contida no *Datasheet* $B=3555$, sabendo que para a temperatura de 25°C (ou 298K) o valor da resistência é de 10k Ω .^[10]

Devem ser satisfeitas as seguintes condições:

- a tensão de saída deve variar no limite de 0 a 5 volts;
- a tensão $V_s = 2,5V$ deve indicar uma temperatura de 37°C ou 310K.

Fazendo $T_1=298K$, $R_{T_1}=10k\Omega$ e $T_2=310K$, temos que:

$$R_{T_2} = 10 \cdot 10^9 \cdot e^{3555 \cdot \left(\frac{1}{310} - \frac{1}{298} \right)}$$

$$R_{T_2} = 6,3016m\Omega$$

Sabendo que :

$$V_B = V_T = \frac{6,3016 \cdot 10^3}{100 \cdot 10^3 + 6,3016 \cdot 10^3} \cdot 5$$

$$V_B = 0,2964V$$

$$V_A = \frac{R_2}{R_1 + R_2} \cdot 5 = V_B + 0,25V$$

$$V_A = 0,5464V$$

Escolhendo $R_1 = 100k\Omega$ temos que $R_2 = 12,269k\Omega$.

Com determinado ganho de 10 para o INA118 devemos fazer $R_{ga} = 5,55k\Omega$.

4 SOFTWARE

Este capítulo explica o projeto, planejamento e programação computacional da parte de captura e desenho dos sinais analisados.

Cada um dos três sistemas possui uma saída de tensão. Tais saídas analógicas podem ser observadas por meio de um osciloscópio, que é um equipamento de difícil manuseio para leigos e relativamente caro – os mais simples custam em média R\$ 1.500,00 e os mais sofisticados chegam a R\$ 10.000,00. Para uma visualização mais prática e viável dos sinais de saída, foi feito um *software* que plota os dados na tela do computador, após capturá-los através de uma de suas portas USB.

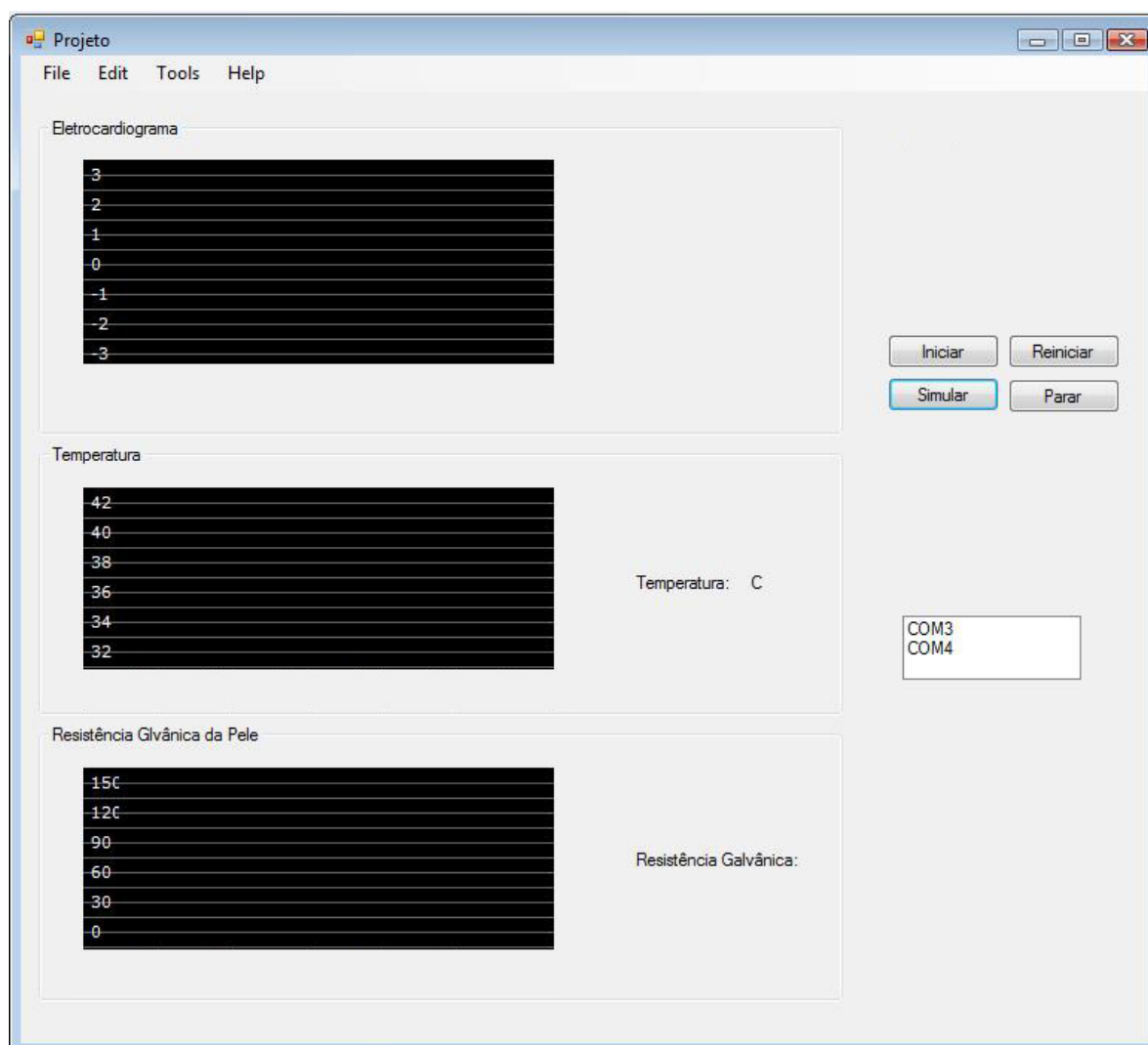


Figura 4.1: Interface do Programa desenvolvido

4.1 MATERIAL E MÉTODO

4.1.1 MATERIAL UTILIZADO

- DLP-IO8-G, para a aquisição de dados;
- *Microsoft Visual Basic 2008 Express Edition* ^[7].

4.1.1 PROCEDIMENTOS

4.1.2.1 AQUISIÇÃO DOS DADOS

O *hardware* foi construído para apresentar, como saída, valores analógicos de tensão na faixa de 0 a 5 volts. No entanto, para que os dados possam ser processados pelo computador e trabalhados no *software*, eles devem primeiramente passar por uma conversão analógico/digital.

Dessa forma, foi utilizado o DLP-IO8-G, um dispositivo da marca DLP Design usado para aquisição de sinais. Tal dispositivo é mostrado na figura 4.2 e apresenta oito canais de entrada analógica. Cada sinal analógico de entrada é convertido em digital e transmitido pelo canal USB do computador a uma taxa de 115200 bauds – a máxima possível. Isso porque, mesmo o sinal entrando pela porta USB a comunicação ocorrida é serial. O DLP-IO8-G, assim como qualquer outro periférico que se comunica com a CPU dessa maneira, faz uso da comunicação serial assíncrona. Tal forma de comunicação é interessante devido sua simplicidade, pois bastam três fios para que ela ocorra: um para a transmissão, um para recepção e outro como referência de tensão, sendo para isso ligado ao terra.

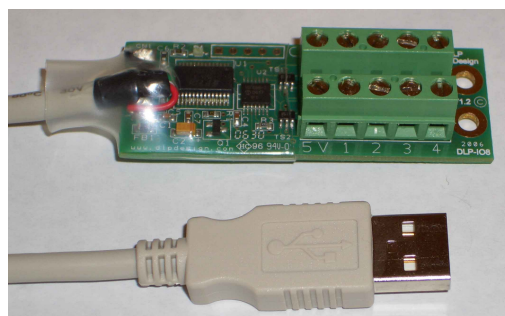


Figura 4.2: DLP-IO8-G - Dispositivo de aquisição de dados.

Como na forma assíncrona não existe relógio para sincronizar a transmissão, é necessário que, antes do início da comunicação seja definida a janela de tempo na qual o transmissor envia (e o receptor recebe) cada *bit* ^[6]. Para isso, são de fato transmitidos 10 *bits*, pois além dos oito que contêm a informação são transmitidos também um *bit* de partida e um de parada.

Quando a transmissão vai ser iniciada, a linha é colocada em nível baixo, sinalizando o bit de partida; após transmitir o último *bit* a linha de transmissão é colocada em nível alto, indicando o *bit* de parada. A janela de tempo de cada um desses *bits* corresponde ao inverso da taxa de comunicação (*baud rate*). Uma desvantagem da comunicação serial, no entanto, é sua velocidade, pois os 8 *bits* são transmitidos sequencialmente por um único caminho. Outras formas de comunicação que fazem uso da porta USB, com protocolos mais sofisticados, podem chegar a uma taxa de transmissão da ordem de 480 Mbps. O USB utiliza um conector universal que permite ao usuário instalar e remover periféricos sem sequer abrir o computador, podendo ele ser instalado o removido a qualquer momento ^[6].

Antes da primeira utilização do dispositivo no computador é necessária a instalação dos *drivers* R2176 e dlp-io8, disponibilizados pela DLP Design em seu *site* ^[8]. O primeiro permite o reconhecimento do dispositivo, enquanto último *driver* citado deve ser usado para verificar qual é a porta serial COM utilizada durante a comunicação. A aquisição de dados funciona da seguinte forma: é enviado um comando de um *byte* indicando qual a porta deseja-se ler e em seguida são recebidos dois *bytes*, mais significativo e menos significativo, respectivamente, com os dados lidos.

Por especificação do fabricante, os sinais de entrada devem estar contidos na faixa de -0,3 a 5,3 volts, faixa essa diferente daquela do sinal de saída do circuito de ECG do *hardware*. Por esse motivo, foi implementado um sistema limitador para mudar a saída analógica para a faixa de 0 a 5 volts. A partir do *hardware* entram nos canais 1, 3 e 5 do DLP-IO8 os sinais de Eletrocardiograma, Temperatura e Resistência galvânica da pele, respectivamente.

4.1.2.2 CÓDIGO DE PROGRAMAÇÃO

O *software* foi desenvolvido em *Visual Basic*, uma linguagem de programação produzida pela *Microsoft* e compilada no programa *Microsoft Visual Basic 2008 Express Edition*. Sendo um aperfeiçoamento do *Basic* – linguagem dirigida por eventos –, possui também um ambiente de desenvolvimento integrado (*IDE - Integrated Development Environment*) totalmente gráfico, o que facilita a construção da interface do programa. O desenvolvimento do *software* foi dividido em quatro partes: a inicialização do programa, a simulação de dados, a aquisição dos dados de cada canal e a plotagem dos dados nos gráficos respectivos.

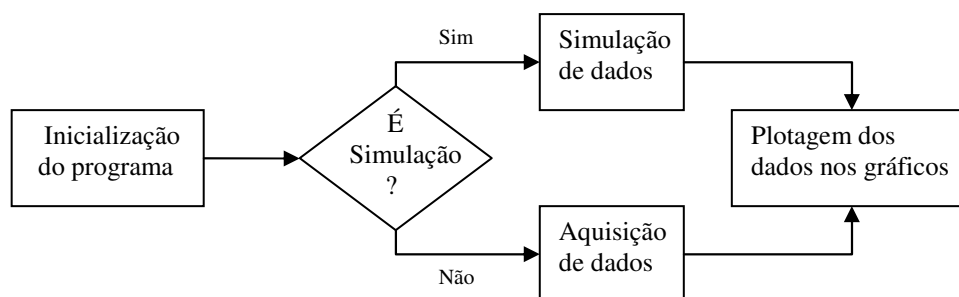


Figura 4.3: Funcionamento do programa dividido em blocos.

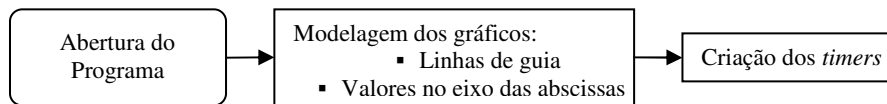


Figura 4.4: Fluxograma relativo à inicialização do programa.

Ao ser inicializado, o programa deixa as figuras já prontas para receberem o gráfico com os sinais observados, desenhando as linhas guia e os valores de tensão fixos no eixo das ordenadas. Para economizar tempo, nesta etapa os quatro *timers* usados são criados, permanecendo, porém, ainda desabilitados. Quando o botão “Iniciar” é ativado, ocorre a habilitação dos mesmos e o preparo da porta serial pelos comandos: `PortaSerial.Open()` e `PortaSerial.WriteLine("\n")`, que indica que os dados devem ser recebidos em valor binário.

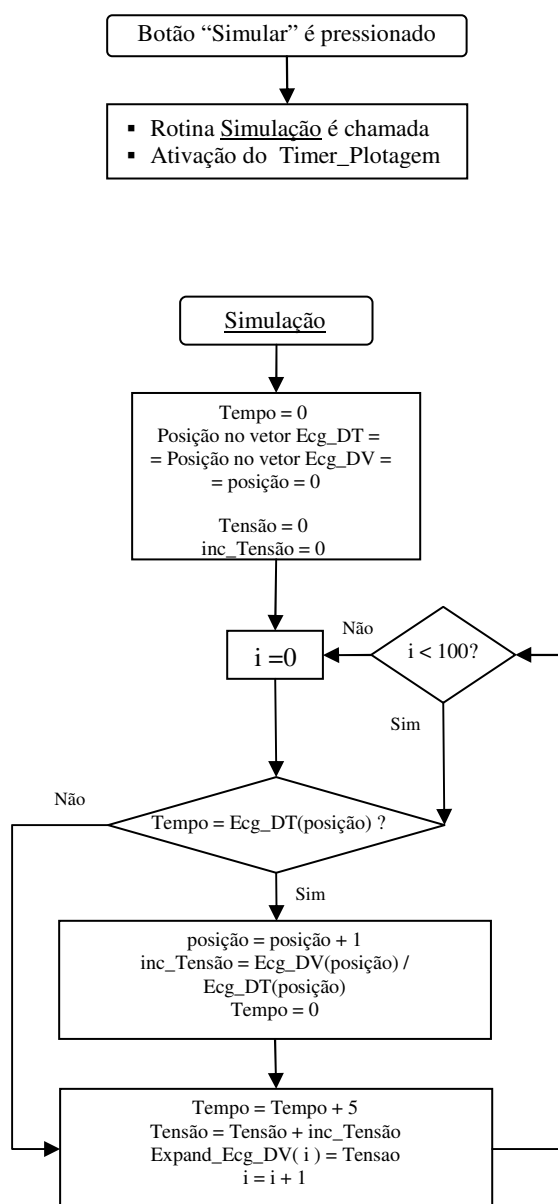


Figura .4.5: Fluxograma relativo à simulação.

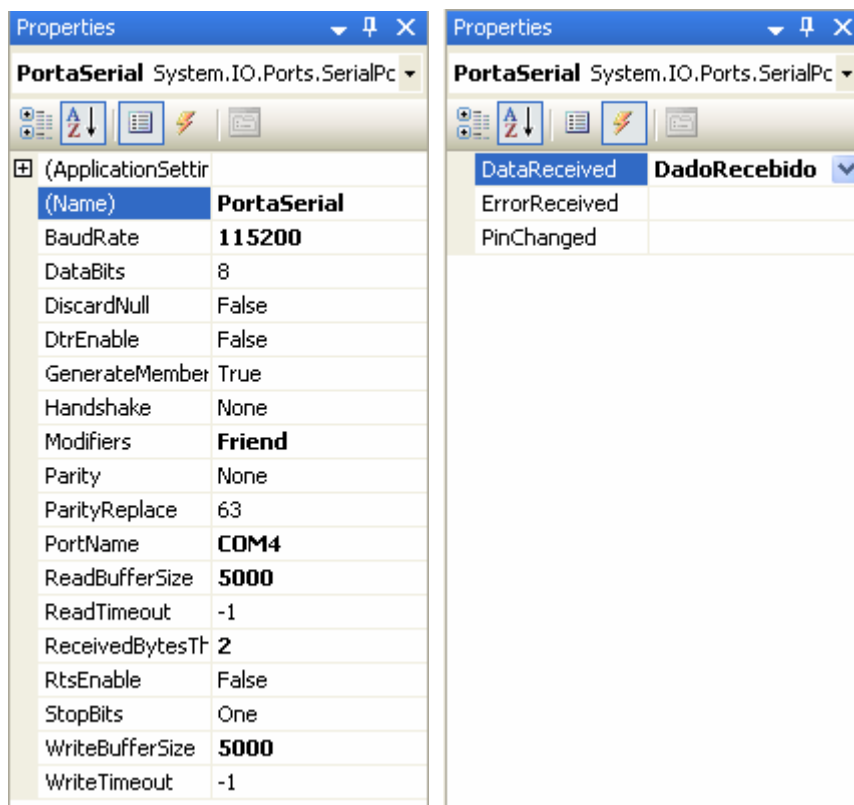
Para testar o funcionamento da parte gráfica do programa independentemente da aquisição de dados, foi feito um código que simula a recepção dos sinais. Os dados para a simulação são obtidos a partir de dois vetores, um de variação de tensão e outro correspondendo ao intervalo de tempo em que essa variação do sinal desejado deve ocorrer. Assim, dividindo-se o primeiro pelo segundo obtém-se o valor a ser incrementado à tensão a cada intervalo de tempo unitário. Para o sinal de ECG é considerado um valor inicial de 70 *pixels*, indicando no gráfico a tensão de zero volt. A cada *loop* o valor inicial é incrementado pela razão acima explicada e a variável Tempo é aumentada em 5 unidades; quando a mesma atingir o valor presente na posição do vetor a ser considerada ocorre o seguinte processo: o Tempo é zerado, a posição aumentada em uma unidade e o novo valor do incremento (razão entre tensão e tempo) é calculado. Quando o botão “Simular” é ativado, a rotina de simulação é chamada e o *timer* relativo ao desenho do gráfico é habilitado. Tal desenho ocorre a cada 5 ms. Os vetores com os dados de um sinal padrão de ECG usados na simulação encontram-se a seguir.

```
Dim Ecg_DT() As Byte = {200, 10, 20, 10, 50, 10, 20, 20, 10, 40, 30, 20, 50, 20, 50, 220, 219}
```

```
Dim Ecg_DV() As SByte = {0, 10, 0, -10, 0, -50, 110, -100, 35, 15, 6, 1, 0, -5, -30, -10, 0}
```

A parte relativa à captura dos dados lidos pela porta serial dá-se principalmente por meio da componente *SerialPort*, presente na *Toolbox* do programa *Visual Basic 2008 Express Edition*. As propriedades dessa componente – tais como a taxa de transmissão/recepção, número de bits transmitidos, número de *bytes* recebidos, etc. – são facilmente ajustadas, como é mostrado na figura 4.6.

Uma característica importante dessa componente é a facilidade de se trabalhar com a porta serial, não sendo necessária a interação direta com *hardware* do PC. O coração da interface serial do PC é o circuito 8250, ou seu equivalente funcional ^[6]. No entanto, não é necessário criar um código para controlar os pinos desse circuito para se ter uma interrupção sempre que algum dado é recebido, pois a componente *SerialPort* já o faz, devendo-se apenas escrever o nome desejado para a interrupção no campo *DataReceived*, mostrado na figura 4.6 b.



(a)

(b)

Figura 4.6: Propriedades da porta serial.

Para cada canal do DLP-IO8 a ser lido é criado um *timer*. Ele conta, decrescentemente, continuamente e em milissegundos, o intervalo de tempo ocorrido entre cada leitura do canal. Esse intervalo nada mais é do que o período de amostragem, T_s . Quando o tempo

programado chega a zero é chamada uma interrupção – *Timer_Ecg_Elapsed*, *Timer_Temp_Elapsed* ou *Timer_Rgp_Elapsed*, a depender de qual *timer* a gerou – nela são atribuídos às variáveis de controle *CanalLido* e *Controle* os valores que indicam qual canal será lido, além de escrever no *DLP_IO8* o comando pedindo os dados do canal que chamou a interrupção. Tal comando é do tipo `PortaSerial.WriteLine("Z")`, no qual a letra entre aspas indica o canal a ser lido, sendo “Z” para o canal 1, “C” para o canal 2 e “B” para o 5. Quando o dado pedido é recebido pela porta serial, ocorre outra interrupção – chamada *DadoRecebido*. Nessa interrupção são lidos os *bytes* mais significativo (MSB) e menos significativo (LSB), nessa ordem. O MSB é transladado 8 bits para a esquerda e é então somado ao LSB para a obtenção do valor do sinal lido. A variável em que esse valor será guardado dependerá do controle feito pelos comandos `If/ElseIf` e pelas variáveis de controle, que serão zeradas ao final da interrupção. A seqüência dos valores lidos é guardada num vetor de 100 posições e posteriormente lida pela rotina de desenho gráfico. Como os *timers* e as rotinas são independentes e como é possível a execução de partes do programa em tempo compartilhado, não é necessário que um valor seja plotado para que ocorra a aquisição do próximo dado, proporcionando assim uma certa agilidade ao programa. A representação dessa parte relativa a captura dos dados convertidos encontra-se no fluxograma da figura 4.7.

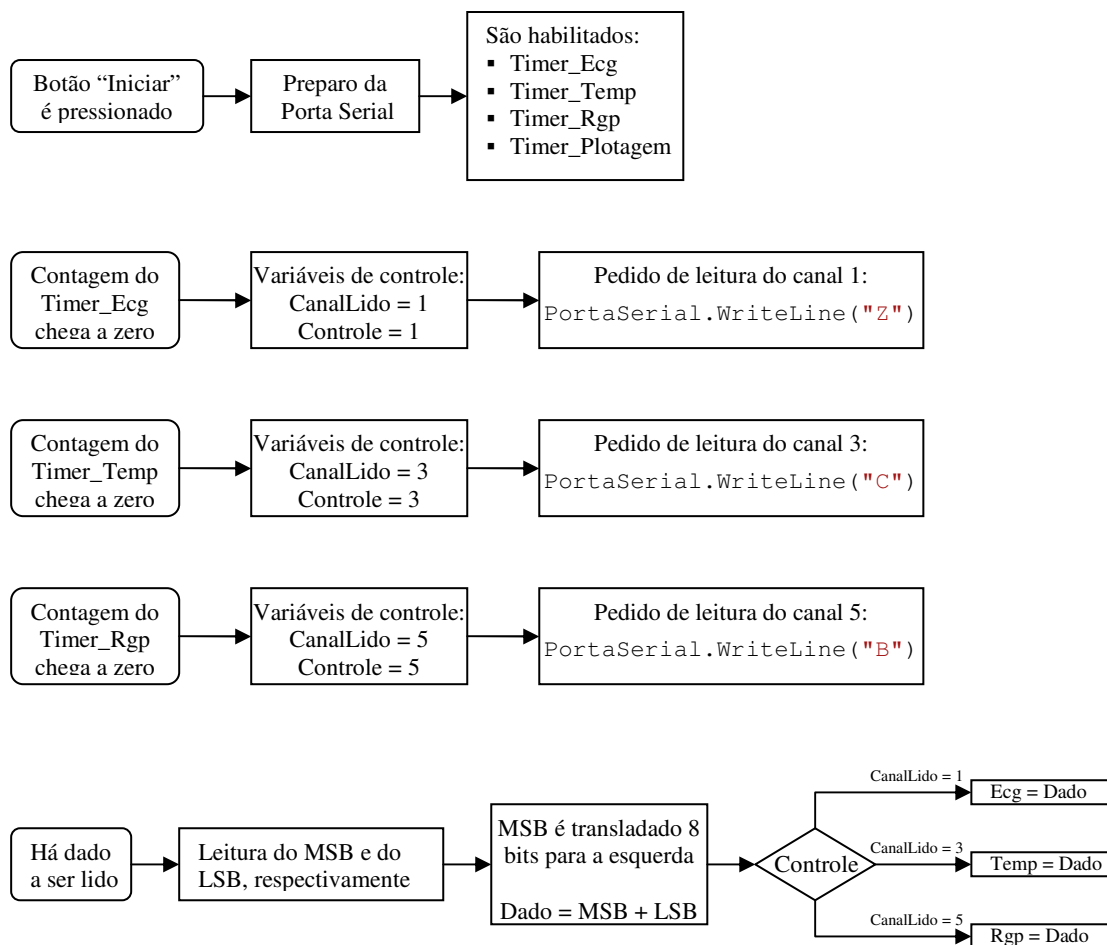


Figura 4.7: Fluxograma relativo à aquisição de dados.

Sabe-se, pelo Teorema de Nyquist que a relação entre o período de amostragem, T_s , e a banda do sinal amostrado, B , em hertz, é tal que $T_s < 1/2B$. A frequência cardíaca normal de uma pessoa adulta saudável varia entre 60 e 100 batimentos por minuto, ou seja, numa frequência de 1 a 1,667 Hz – para crianças e adolescentes esse valor é um pouco maior. No entanto, a onda QRS apresenta uma rápida variação e a maior parte de sua energia é concentrada na faixa de 5 a 15 Hz. Assim, a banda B considerada para os cálculos é de 15 Hz, ou seja, o período de amostragem para o sinal de eletrocardiograma deve ser menor que 33,33 ms. Já os sinais de temperatura e resistência da pele apresentam uma variação mais lenta, podendo tal período ser de até 100 ms sem prejudicar a qualidade do sinal.

Com os dados adquiridos, já é possível plotá-los no gráfico. Como a linguagem *Visual Basic* permite o compartilhamento temporal da CPU das instruções do código do programa, a rotina de desenho no gráfico não precisa esperar cada dado ser adquirido para ocorrer, assim é criado também para ela um *timer*, que gera a interrupção que contém as instruções de trabalhar com os gráficos, chamando a rotina *DrawGuidelinesAndChart*.

Nessa interrupção a última versão do gráfico é recuperada e transladada um pixel para a esquerda. No espaço que ficou vago à direita as linhas de guia são complementadas e também é desenhada uma linha ligando o ponto do último valor do sinal com o ponto do mais novo valor. Tais valores são adquiridos do vetor criado na rotina *DadoRecebido*.

A relação entre a tensão medida e o valor lido pela porta serial local mantém a proporção de um para duzentos, ou seja, 1 volt lido pelo dispositivo corresponde a um valor de 200 na variável que o armazena. Como o objeto gráfico usado para mostrar o sinal de ECG, numa faixa de 5 volts, possui 140 *pixels* de altura, então 1 volt equivale a 28 *pixels*. Assim, para que o sinal seja devidamente representado no gráfico, o valor armazenado deve ser dividido por $5 \cdot 200 / 140 \cong 7,14$. Como as janelas de temperatura e resistência galvânica da pele possuem 122 *pixels* de altura, a divisão deve ser feita por 8,2.

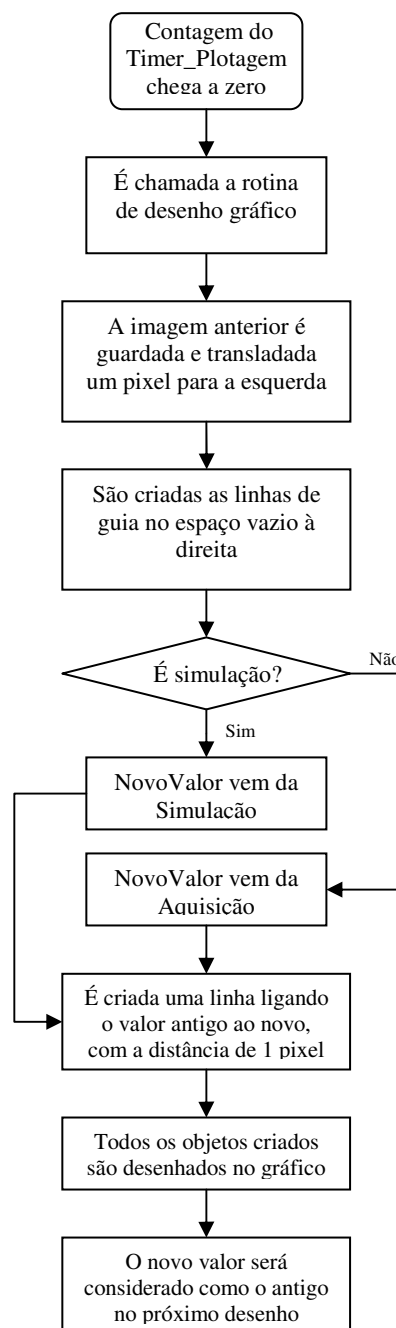


Figura 4.8: Fluxograma relativo ao desenho nos gráficos.

5 MEDIÇÕES E ANÁLISE

Este capítulo apresenta os testes realizados e uma análise crítica.

As medições realizadas apenas com o *hardware*, apresentaram alguns problemas iniciais. Logo após os testes de tensão pino a pino, descobriu-se que havia uma deformação na saída provocada por um filtro passa baixa que continha frequência de corte muito inferior que a desejada, cortando então uma parte do sinal. A troca da frequência de corte do filtro para 20Hz foi realizada e então foi possível obter a forma se onda esperada.

Os testes finais foram realizados com o professor orientador, que observou no osciloscópio a saída dos sinais adquiridos pelo equipamento.

O eletrocardiograma apresentou corretamente o sinal de onda do batimento cardíaco. Na figura 5.1 pode-se observar a saída, antes de passar pelo circuito limitador de tensão, e compará-la a onda teórica que esperávamos obter.

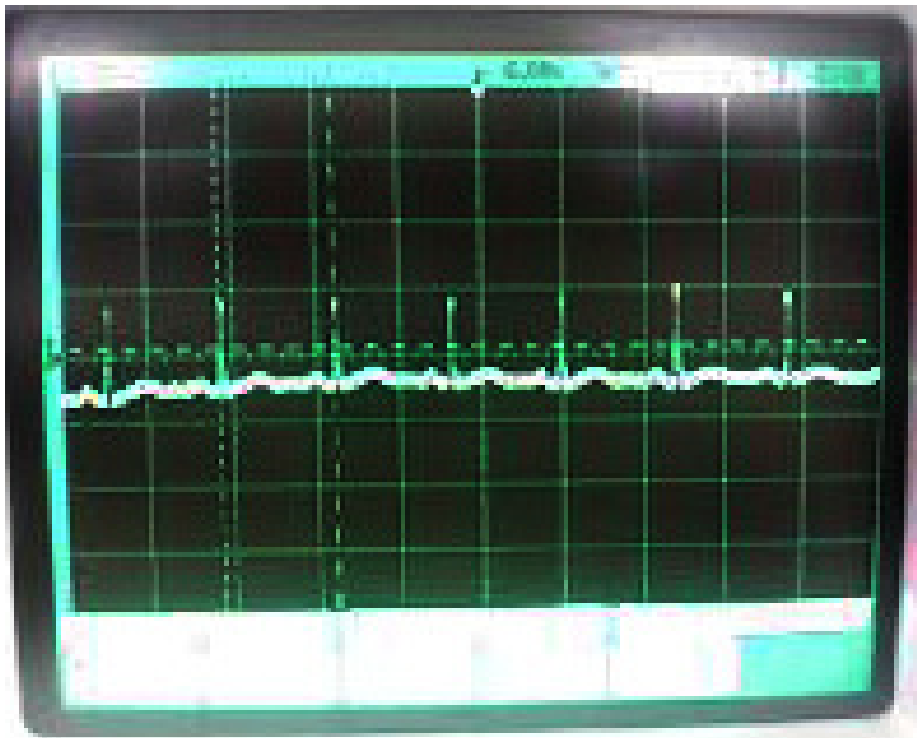
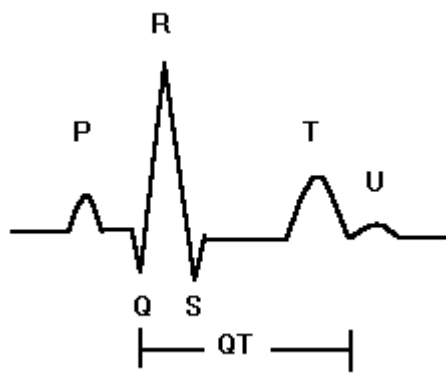
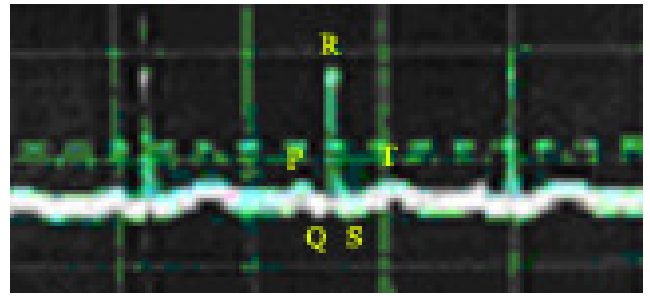


Figura 5.1: Sinal de ECG antes de passar pelo circuito limitador de tensão.



(a) Onda teórica



(b) Onda visualizada no osciloscópio

Figura 5.2: Formato das ondas de batimento cardíaco, para efeito de comparação.

O formato do pico QRS, e das formas senoidais P e T pode ser identificado.

Após passar pelo circuito limitador de tensão, o sinal do eletrocardiograma adquiriu uma tensão DC de *offset* de 2,2 volts, estando limitado na faixa desejada da 0 a 5 volts. Este sinal pode ser visto na figura 5.3.

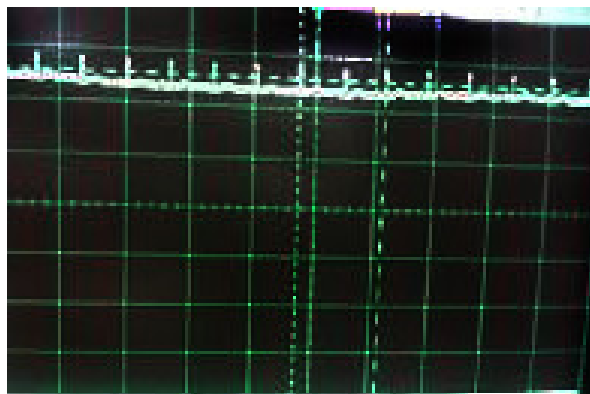
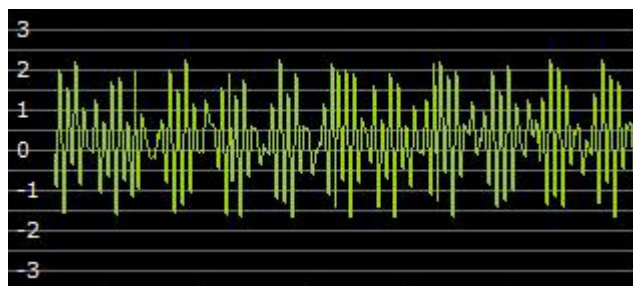


Figura 5.3: Sinal de ECG após passar pelo circuito limitador de tensão.

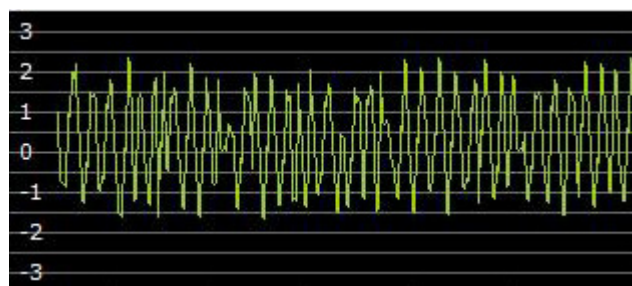
Os sinais de resistência galvânica da pele e de temperatura foram obtidos e analisados no momento de realização dos testes. Como são sinais que não apresentam grandes variações, não são apresentadas figuras com suas respostas. Os valores obtidos a partir da medição feita foi, para resistência da pele – tensão de 2,9 volts – 118,8 ohms, e para temperatura – tensão de 2,35 volts – 36,7°C.

Para testar o *software* nas partes relativas à aquisição e plotagem de dados foi colocada uma onda triangular na entrada do primeiro canal do DLP-IO8. A onda triangular foi escolhida para teste por ser esta a que mais se assemelha ao sinal de ECG, que dentre os sinais amostrados é o que possui maior frequência. Como a onda QRS do eletrocardiograma apresenta variações de até 15 Hz, a onda triangular de teste foi especificada a esta frequência. Pelo teorema de Nyquist o período de amostragem máximo deve ser de 33,33 ms. Ajustando

o *timer* do canal um para 30 ms, um pouco menor que o limite, o sinal mostrado na tela do programa apresentou claramente a existência de uma falha no *software* – figura 5.4 (a). Mudando o ajuste do período para 5 ms – mais de 6 vezes menor que o limite –, ainda assim o sinal não foi visualizado na tela do programa com precisão – figura 5.4 (b). Diminuindo-se a frequência da onda triangular para 5 Hz verifica-se uma melhor qualidade no sinal observado, porém ainda assim não há precisão – figura 5.5.

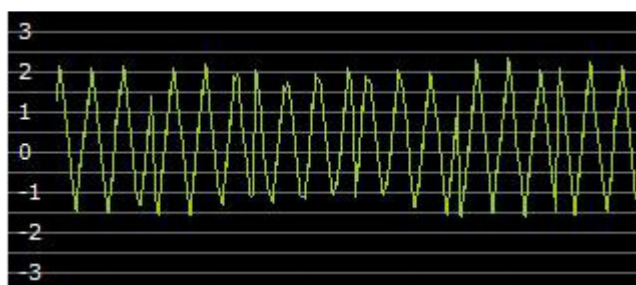


(a) Período de amostragem de 30ms.

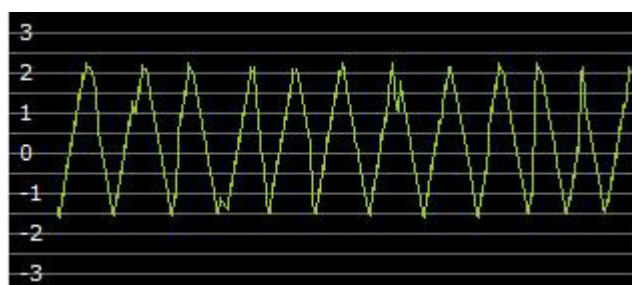


(b) Período de amostragem de 5ms.

Figura 5.4: Onda triangular de frequência 15Hz.



(a) Período de amostragem de 30ms.



(b) Período de amostragem de 5ms.

Figura 5.5: Onda triangula de frequência 5Hz.

Foram feitas algumas mudanças nos *timers* relativos aos sinais e à plotagem do gráfico, porém sem sucesso. Ligando a saída do *hardware* referente ao sinal de ECG ao *software* foi observada a forma de onda mostrada na figura 5.6.



Figura 5.6: Dados de ECG adquiridos e plotados no programa.

Para os sinais de temperatura e resistência galvânica da pele, que possuem uma variação muito lenta, o programa funcionou com sucesso.

Outra falha no sistema é que quando os três canais são ligados ao mesmo tempo ocorre uma mistura dos dados convertidos, de forma que os gráficos desenhados ficam disformes. Foram feitas várias rotinas de controle diferentes, no entanto nenhuma delas foi suficiente para atribuir o valor recém convertido à variável relacionada àquele sinal.

A formação do sinal de ECG por meio de simulação e a formação do respectivo gráfico funcionaram conforme o esperado. Dessa forma, o sinal é plotado de forma praticamente fidedigna, figura 5.7.



Figura 5.7: Dados de ECG simulados e plotados no programa.

Assim, o software funcionou parcialmente.

6 CONCLUSÕES

O *hardware* do equipamento projetado foi construído e colocado em uma caixa, cobertura externa, criada com a quantidade exata de entradas e saídas planejadas. Os testes e medidas no próprio corpo humano possibilitaram a visualização do sinal de saída de cada circuito. Neste sinal pudemos identificar o formato da onda do batimento cardíaco com frequência constante. A resistência da pele foi medida e variada de acordo com o estado do corpo. E a temperatura também proporcionou análise dentro dos valores da escala esperada.

Os componentes eletrônicos estudados foram de extrema utilidade e, portanto, facilitaram a construção da placa física. Percebemos que o amplificador INA 118 tornou o projeto mais preciso e estável para a sua utilização em engenharia biomédica.

Os resultados esperados foram obtidos com êxito, ou seja, pudemos verificar que alterações corporais provocam mudanças desejadas nos sinais de saída. Durante o desenvolvimento do projeto tivemos alguns problemas com ruídos, erros de precisão e acurácia, que foram contornados e minimizados ao final do projeto.

A linguagem de programação *Visual Basic*, e o programa *Microsoft Visual Basic 2008 Express Edition*, facilitaram consideravelmente a produção do *software*, principalmente por seu ambiente de desenvolvimento integrado gráfico e por suas ferramentas que permitem ao programador o desenvolvimento de um código mais simples, sem a necessidade de o mesmo interagir diretamente com o hardware do PC.

Como é possível a execução em tempo compartilhado de mais de uma rotina, os *timers* foram bastante úteis, pois possibilitaram uma maior independência entre os blocos de funcionamento e até mesmo dentro deles, dando uma maior agilidade ao programa. Assim, foi possível a aquisição de dados a uma frequência superior à determinada pelo Teorema de Nyquist. No entanto, apesar dessa alta taxa de amostragem, algumas partes do sinal não são capturadas. Além disso, quando todos os canais encontram-se em funcionamento simultâneo o sistema não consegue, com sucesso, discernir de qual canal veio o dado e, portanto, há uma mistura dos valores recebidos. Dessa forma, o software funcionou parcialmente.

A experiência de planejamento, criação, montagem e programação do projeto foi uma importante contribuição para o nosso conhecimento, pois utilizamos conceitos teóricos e práticos do curso de engenharia elétrica em todos os momentos desta elaboração.

Apresentamos a proposta de aperfeiçoar o projeto do equipamento fazendo com que o *software* capture os três sinais provenientes do *hardware* e mostre-os simultaneamente na tela do computador de forma fidedigna. Este objetivo condiz com a área que optamos por seguir e pela qual tivemos grande satisfação, a de engenharia biomédica.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Machado, Ângelo (2003), *Neuroanatomia Funcional*, Editora Ateneu.2 ed.
- [2] Sedra, A.S., Smith, K.C. (2005), *Microeletrônica*, Quarta Edição, São Paulo, SP: Pearson Makron Books.
- [3] LaLond, D.E., Ross, J.A. (2000), *Princípios de Dispositivos e Circuitos Eletrônicos*, Volume2, São Paulo, SP: Makron Books.
- [4] Burton, A.C. (1997), *Fisiologia e Biofísica da Circulação*, Segunda Edição, Rio de Janeiro, RJ: Guanabara Koogan.
- [5] Stephenson, R.J., Moulin, A.M., Welland, M.E. (2000), *Mechanical Variables Measurement, Solid, Fluid and Thermal*: CRC Press LLC.
- [6] Zelenovsky, R., Mendonça, A. (2006), *PC: Um guia prático de Hardware e Interfaceamento*, Quarta Edição, Rio de Janeiro, RJ: MZ Editora Ltda.
- [7] Microsoft Press (2006), *Microsoft Visual Basic 2006 Express Edition - Build a Program Now*, Washington : Microsoft Corporation.
- [8] Datasheet do conversor DLP-IO8, retirado do site www.dlpdesign.com.
- [9] Datasheet do amplificador INA 118.
- [10] Datasheet do termistor RS NTC10 – Thermistors.

ANEXO I – Placa de circuito impresso

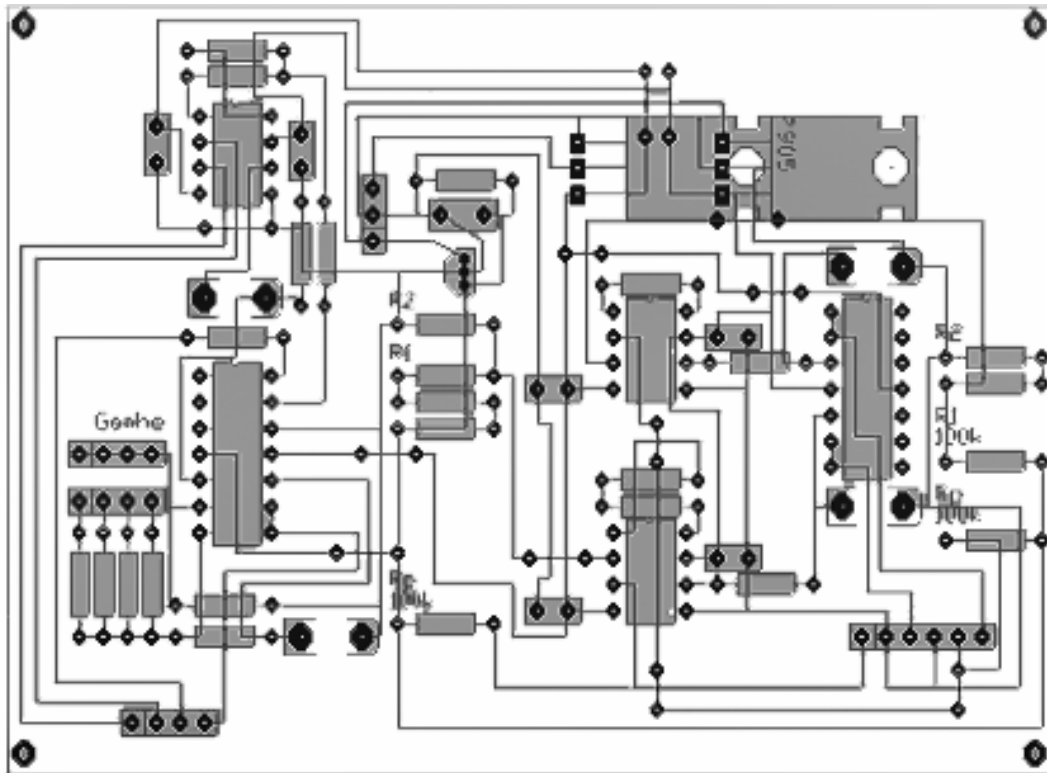


Figura A.1: Trilha de circuito impresso criada no TraxMaker 3 Pro

ANEXO II – Fotos do *Hardware*

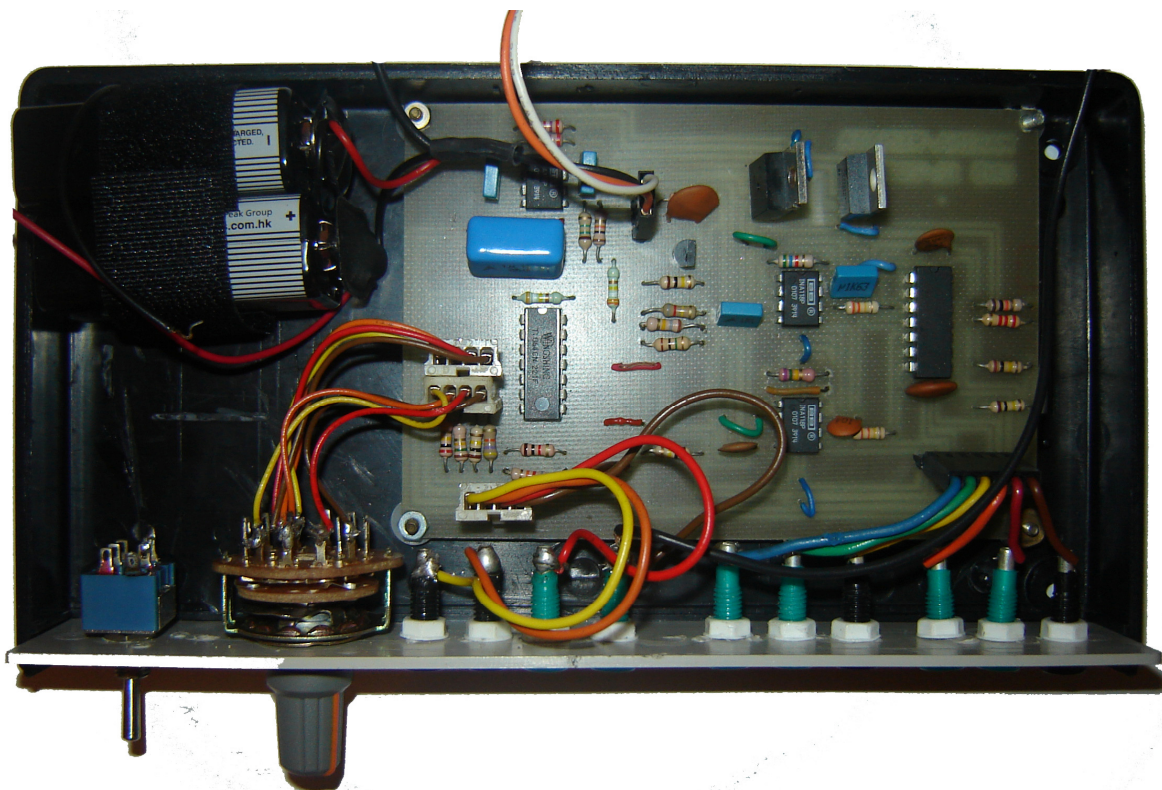


Figura A.2: Vista superior do *hardware*.



Figura A.3: Vista frontal do equipamento fechado.

ANEXO III – Fotos dos cabos montados para a medição



Figura A.4: Cabos e conectores do eletrocardiograma em forma trançada.



Figura A.5: Cabos e eletrodos de medição da resistência da pele.



Figura A.6: Cabos e termistor para a medição da temperatura.

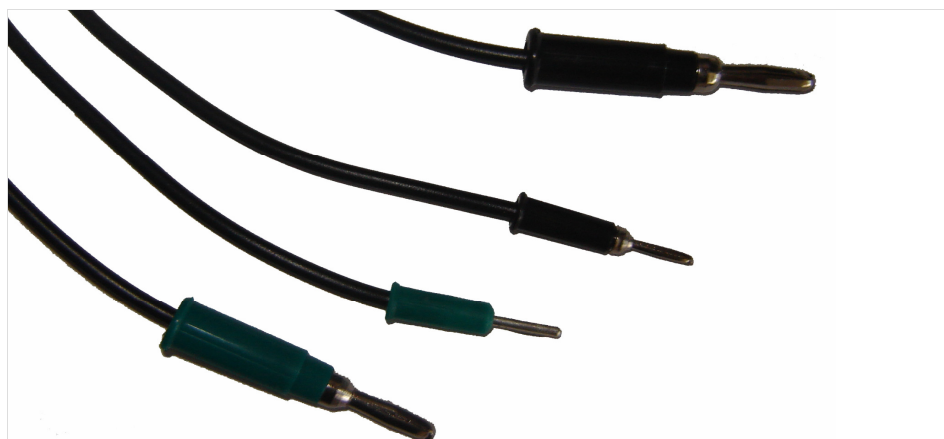


Figura A.7: Cabos e conectores montados para condução de sinal.

ANEXO IV – Código do programa

```
Imports System.Drawing.Drawing2D
Imports System.Math
Imports System.IO.Ports

Public Class Form1

    Private Declare Function rdtsc Lib "csRDTSC.dll" () As Int64

    Dim Sinal_Ecg(0 To 500) As UShort
    Dim Sinal_Temp(0 To 500) As UShort
    Dim Sinal_RGP(0 To 500) As UShort

    Dim Valor_Ecg As UShort = 70 ' Corresponde ao centro do gráfico
    Dim Valor_Temp As UShort = 70
    Dim Valor_RGP As UShort = 70

    Dim Xmove As Integer = 1 ' gráfico move 1 pixel para cada plotagem na tela

    Private OldValue_Ecg As UShort = 70 ' Corresponde ao centro do
gráfico
    Private OldValue_Temp As UShort = 70
    Private OldValue_RGP As UShort = 70

    Private NewValue_Ecg As UShort = 0
    Private NewValue_Temp As UShort = 0
    Private NewValue_RGP As UShort = 0

    Dim Timer_Ecg As New System.Timers.Timer
    Dim Timer_Temp As New System.Timers.Timer
    Dim Timer_Rgp As New System.Timers.Timer
    Dim Timer_Plotagem As New System.Timers.Timer
    Dim Timer_Simulacao As New System.Timers.Timer

    Dim Simula As Boolean = False

    Private Sub Inicio(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        Control.CheckForIllegalCrossThreadCalls = False
        'desabilita verificação de cross Thread

        DrawGuidelines(Me.PicGraph, Me.PicGraphTemp, Me.PicGraphRGP)
        DrawVerticalValues(Me.PicYAxis, Me.PicYAxisTemp, Me.PicYAxisRGP)
        DrawHorizontalValues(Me.PicXAxis, Me.PicXAxisTemp, Me.PicXAxisRGP)

        Cria_Timer_Ecg()
        Cria_Timer_Temp()
        Cria_Timer_Rgp()
        Cria_Timer_Plotagem()
        GetSerialPortNames()

    End Sub
```



```

Private Sub DrawGuidelines(ByVal PictureBox As PictureBox, ByVal PictureBoxTemp
As PictureBox, ByVal PictureBoxRGP As PictureBox)
    ' Cria um bitmap para desenhar e armazena-o no objeto gráfico

    Dim bmp As New Bitmap(PictureBox.Width, PictureBox.Height)
    Dim bmpT As New Bitmap(PictureBoxTemp.Width, PictureBoxTemp.Height)
    Dim bmpR As New Bitmap(PictureBoxRGP.Width, PictureBoxRGP.Height)
    Dim grp As Graphics = Graphics.FromImage(bmp)
    Dim grpT As Graphics = Graphics.FromImage(bmpT)
    Dim grpR As Graphics = Graphics.FromImage(bmpR)

    ' Desenha linhas horizontais no objeto gráfico.
    For i As Integer = 10 To 140 Step 10
        grp.DrawLine(Pens.DimGray, 0, i, PictureBox.Width, i)
        grpT.DrawLine(Pens.DimGray, 0, i, PictureBoxTemp.Width, i)
        grpR.DrawLine(Pens.DimGray, 0, i, PictureBoxRGP.Width, i)
    Next i

    ' Coloca o bitmap de novo no PictureBox como sua imagem.
    PictureBox.Image = bmp
    PictureBoxTemp.Image = bmpT
    PictureBoxRGP.Image = bmpR

    ' Libera os gráficos
    grp.Dispose()
    grpT.Dispose()
    grpR.Dispose()

End Sub

```

```

Private Sub DrawVerticalValues(ByVal PB As PictureBox, ByVal PB_T As
PictureBox, ByVal PB_R As PictureBox)
    ' Desenha valores no eixo vertical

    ' Passo 1
    ' Cria um bitmap para desenhar e armazena-o no objeto gráfico
    Dim bmp As New Bitmap(PB.Width, PB.Height)
    Dim bmpT As New Bitmap(PB_T.Width, PB_T.Height)
    Dim bmpR As New Bitmap(PB_R.Width, PB_R.Height)
    Dim gv As Graphics = Graphics.FromImage(bmp)
    Dim gvT As Graphics = Graphics.FromImage(bmpT)
    Dim gvR As Graphics = Graphics.FromImage(bmpR)

    ' Passo 2
    ' Desenha linhas do eixo vertical
    For i As Integer = 10 To 140 Step 10
        gv.DrawLine(Pens.DimGray, 0, i, PB.Width, i)
        gvT.DrawLine(Pens.DimGray, 0, i, PB_T.Width, i)
        gvR.DrawLine(Pens.DimGray, 0, i, PB_R.Width, i)
    Next i

    ' Passo 3
    ' Escreve valores no eixo
    ' Gráfico do ECG
    Dim NextMarker As Integer = 3
    For i As Integer = 3 To 140 Step 20
        gv.DrawString(CStr(NextMarker), New Font("Verdana", 8,
FontStyle.Regular), Brushes.WhiteSmoke, 3, i)
        NextMarker += 1
    Next
    ' Gráfico da Temperatura
    Dim NextMarker2 As Integer = 42

```

```

        For i As Integer = 3 To 140 Step 20
            gvT.DrawString(CStr(NextMarker2), New Font("Verdana", 8,
FontStyle.Regular), Brushes.WhiteSmoke, 3, i)
            NextMarker2 -= 2
        Next

        ' Gráfico da Resistência Galvânica da pele
        Dim NextMarker3 As Integer = 150
        For i As Integer = 3 To 140 Step 20
            gvR.DrawString(CStr(NextMarker3), New Font("Verdana", 8,
FontStyle.Regular), Brushes.WhiteSmoke, 3, i)
            NextMarker3 -= 30
        Next

        ' Passo 4
        ' Coloca o bitmap de novo no PictureBox como sua imagem
        PB.Image = bmp
        PB_T.Image = bmpT
        PB_R.Image = bmpR

        ' Passo 5
        ' Libera os gráficos
        gv.Dispose()
        gvT.Dispose()
        gvR.Dispose()

End Sub

```

```

Sub GetSerialPortNames()
    'Mostra portas COM no ListBox
    For Each sp As String In My.Computer.Ports.SerialPortNames
        ListBox1.Items.Add(sp)
    Next
End Sub

```

```

' Vetores para simulação dos sinais
Dim Ecg_DT() As Byte = {200, 10, 20, 10, 50, 10, 20, 20, 10, 40, 30, 20,
50, 20, 50, 220, 219}
Dim Ecg_DV() As SByte = {0, 10, 0, -10, 0, -50, 110, -100, 35, 15, 6,
1, 0, -5, -30, -10, 0}

Dim Temp_DT() As Byte = {200, 50, 250, 40, 60, 30, 200, 20, 80, 20, 30,
20, 100, 20, 15, 40, 19}
Dim Temp_DV() As SByte = {0, 5, 0, -6, 0, -5, 0, 10, 0, 50, 10, 50, -
10, -50, -70, -10, 0}

Dim Rgp_DT() As Byte = {200, 50, 250, 40, 60, 30, 200, 20, 80, 20, 30,
20, 100, 20, 15, 40, 19}
Dim Rgp_DV() As SByte = {0, 5, 0, -6, 0, -5, 0, 10, 0, 50, 10, 50, -10,
-50, -70, -10, 0}

Dim Expand_Ecg_DV(0 To 200) As UInteger 'Vetor a expandir dados
Dim Expand_Temp_DV(0 To 200) As UInteger
Dim Expand_RGP_DV(0 To 200) As UInteger
Dim Tensao_Ecg As Integer = 70 'zero do gráfico (meio)
Dim Tensao_Temp As Integer = 70
Dim Tensao_Rgp As Integer = 70

```

```

Dim inc_Ecg As Integer = 0 'incremento da tensão de ECG
Dim inc_Temp As Integer = 0
Dim inc_Rgp As Integer = 0

Dim TempoE As UInteger = 0
Dim TempoT As UInteger = 0
Dim TempoR As UInteger = 0

Dim posiçãoE As UInteger = 0 'posição do vetor Ecg_DT
Dim posiçãoT As UInteger = 0
Dim posiçãoR As UInteger = 0
Dim iExp As Short = 0

Public Sub Simulação()

    For iExp As Short = 0 To 100

        If TempoE = Ecg_DT(posiçãoE) Then 'A condição será satisfeita
quando tiver transcorrido o tempo determinado naquela posição do vetor
            posiçãoE += 1
            inc_Ecg = (Ecg_DV(posiçãoE) * 5) \ (Ecg_DT(posiçãoE))
            TempoE = 0 'começa uma nova contagem
        End If
        Tensao_Ecg += inc_Ecg 'o valor da tensão anterior é
incrementado
        Expand_Ecg_DV(iExp) = Tensao_Ecg
        TempoE += 5 ' conta o tempo de 5 em 5ms

        If TempoT = Temp_DT(posiçãoT) Then
            inc_Temp = (Temp_DV(posiçãoT) * 10) \ (Temp_DT(posiçãoT))
            TempoT = 0
        End If
        Tensao_Temp += inc_Temp
        Expand_Temp_DV(iExp) = Tensao_Temp
        TempoT += 5

        If TempoR = Rgp_DT(posiçãoR) Then
            posiçãoR += 1
            inc_Rgp = (Rgp_DV(posiçãoR) * 20) \ (Rgp_DT(posiçãoR))
            TempoR = 0
        End If
        Tensao_Rgp += inc_Rgp
        Expand_RGP_DV(iExp) = Tensao_Rgp
        TempoR += 10

    Next

End Sub

' Criação dos timers
Private Sub Cria_Timer_Ecg()
    Timer_Ecg.Interval = 5
    Timer_Ecg.Enabled = False
    AddHandler Timer_Ecg.Elapsed, New
System.Timers.ElapsedEventHandler(AddressOf Me.Timer_Ecg_Elapsed)
End Sub

Private Sub Cria_Timer_Temp()
    Timer_Temp.Interval = 100
    Timer_Temp.Enabled = False

```

```

        AddHandler Timer_Temp.Elapsed, New
System.Timers.ElapsedEventHandler(AddressOf Me.Timer_Temp_Elapsed)
    End Sub

    Private Sub Cria_Timer_Rgp()
        Timer_Rgp.Interval = 50
        Timer_Rgp.Enabled = False
        AddHandler Timer_Rgp.Elapsed, New
System.Timers.ElapsedEventHandler(AddressOf Me.Timer_Rgp_Elapsed)
    End Sub

    Private Sub Cria_Timer_Plotagem()
        Timer_Plotagem.Interval = 40
        Timer_Plotagem.Enabled = False
        AddHandler Timer_Plotagem.Elapsed, New
System.Timers.ElapsedEventHandler(AddressOf Me.Timer_Plotagem_Elapsed)
    End Sub

' Variáveis de controle
Dim Canallido As Byte
Dim Controle1 As Byte
Dim Controle2 As Byte

' Rotinas chamadas quando a contagem dos timers chega a zero
    Private Sub Timer_Ecg_Elapsed(ByVal sender As System.Object, ByVal e As
System.Timers.ElapsedEventArgs)
        Canallido = 1
        Controle1 = 1
        PortaSerial.WriteLine("Z")
    End Sub

    Private Sub Timer_Temp_Elapsed(ByVal sender As System.Object, ByVal e
As System.Timers.ElapsedEventArgs)
        Canallido = 3
        Controle1 = 3
        PortaSerial.WriteLine("C")

    End Sub

    Private Sub Timer_Rgp_Elapsed(ByVal sender As System.Object, ByVal e As
System.Timers.ElapsedEventArgs)
        Canallido = 5
        Controle1 = 5
        PortaSerial.WriteLine("B")

    End Sub

    Private Sub Timer_Plotagem_Elapsed(ByVal sender As System.Object, ByVal
e As System.Timers.ElapsedEventArgs)

        DrawGuidelinesAndChart(PicGraph, PicGraphTemp, PicGraphRGP)
        GC.Collect()

    End Sub

Dim ic As Integer = 0
Dim Conv_Ecg As Short
Dim Conv_Temp As Short
Dim Conv_Rgp As Short
Dim MSB As Short

```

```

Dim LSB As Short
Dim Conv As Short
Dim MSB_Ecg As Short
Dim MSB_Temp As Short
Dim MSB_Rgp As Short
Dim LSB_Ecg As Byte
Dim LSB_Temp As Byte
Dim LSB_Rgp As Byte

Dim Vetor_Conv_Ecg(0 To 100) As UInteger
Dim Vetor_Conv_Temp(0 To 100) As UInteger
Dim Vetor_Conv_Rgp(0 To 100) As UInteger

' Rotina chamada quando um dado é recebido pela porta serial
Private Sub DadoRecebido(ByVal sender As System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
PortaSerial.DataReceived

    MSB = PortaSerial.ReadByte
    LSB = PortaSerial.ReadByte
    MSB <<= 8
    Conv = LSB + MSB

    If CanalLido = 1 Then
        If (Controle1 = 1) Then
            Conv_Ecg = Conv
        End If

    ElseIf CanalLido = 3 Then
        If (Controle1 = 3) Then
            Conv_Temp = Conv
        End If

    ElseIf CanalLido = 5 Then
        If (Controle1 = 5) Then
            Conv_Rgp = Conv
        End If

    End If

    Controle1 = 0
    CanalLido = 0

    If ic < 100 Then
        Vetor_Conv_Ecg(ic) = Conv_Ecg / 7.14
        Vetor_Conv_Temp(ic) = Conv_Temp / 8.2
        Vetor_Conv_Rgp(ic) = Conv_Rgp / 8.2
        ic += 1
    Else
        ic = 0
        Vetor_Conv_Ecg(ic) = Conv_Ecg / 7.14
        Vetor_Conv_Temp(ic) = Conv_Temp / 8.2
        Vetor_Conv_Rgp(ic) = Conv_Rgp / 8.2
    End If
End Sub
Dim ig As Byte

' Rotina que desenha os valores nos gráficos
Private Sub DrawGuidelinesAndChart(ByVal PictureBox As PictureBox, ByVal
PictureBoxTemp As PictureBox, ByVal PictureBoxRGP As PictureBox)

    On Error Resume Next

```

```

' Passo 1
' Recupera a imagem atual (a última versão do gráfico)
Dim bm As New Bitmap(PictureBox.Width, PictureBox.Height)
Dim gr As Graphics = Graphics.FromImage(bm)

Dim bmT As New Bitmap(PictureBoxTemp.Width, PictureBoxTemp.Height)
Dim grT As Graphics = Graphics.FromImage(bmT)

Dim bmR As New Bitmap(PictureBoxRGP.Width, PictureBoxRGP.Height)
Dim grR As Graphics = Graphics.FromImage(bmR)

' Passo 2
' Move as linhas à direita do objeto gráfico.
For i As Integer = 10 To 140 Step 10
    gr.DrawLine(Pens.DimGray, PictureBox.Width - Xmove, i,
PictureBox.Width, i)
    grT.DrawLine(Pens.DimGray, PictureBoxTemp.Width - Xmove, i,
PictureBoxTemp.Width, i)
    grR.DrawLine(Pens.DimGray, PictureBoxRGP.Width - Xmove, i,
PictureBoxRGP.Width, i)
Next i

' Passo 3
' Desenha esta imagem recuperada no objeto gráfico, colocando-a
Xmove pixels à esquerda da área da imagem. Nenhuma mudança aparecerá no
PictureBox ainda.

gr.DrawImage(PictureBox.Image, -Xmove, 0)
grT.DrawImage(PictureBoxTemp.Image, -Xmove, 0)
grR.DrawImage(PictureBoxRGP.Image, -Xmove, 0)

' Passo 4
' Determina quais valores assumirão as variáveis do tipo NewValue,
verificando primeiro se tal valor vem da simulação ou da aquisição de dados

If Simula = True Then ' Dados provenientes da simulação
    If iExp <= 100 Then
        NewValue_Ecg = Expand_Ecg_DV(iExp)
        NewValue_Temp = Expand_Temp_DV(iExp)
        NewValue_RGP = Expand_RGP_DV(iExp)
        iExp += 1
    Else
        ' Se a posição do vetor ultrapassar 100 unidades a contagem
recomeça do zero
        iExp = 0
        NewValue_Ecg = Expand_Ecg_DV(iExp)
        NewValue_Temp = Expand_Temp_DV(iExp)
        NewValue_RGP = Expand_RGP_DV(iExp)
    End If

Else ' Dados provenientes da aquisição
    If ig < 100 Then
        NewValue_Ecg = Vetor_Conv_Ecg(ig)
        NewValue_Temp = Vetor_Conv_Temp(ig)
        NewValue_RGP = Vetor_Conv_Rgp(ig)
        ig += 1
    Else
        ig = 0
        NewValue_Ecg = Vetor_Conv_Ecg(ig)
        NewValue_Temp = Vetor_Conv_Temp(ig)
    End If
End If

```

```

        NewValue_RGP = Vetor_Conv_Rgp(ig)
    End If
End If

' Passo 5
' Desenha no objeto gráfico a linha que ligará o novo valor ao
último valor plotado.

    gr.DrawLine(Pens.YellowGreen, PicGraph.Width - 1 - Xmove, -
OldValue_Ecg, PicGraph.Width - 1, -NewValue_Ecg)
    grT.DrawLine(Pens.YellowGreen, PicGraphTemp.Width - 1 - Xmove, -
OldValue_Temp, PicGraphTemp.Width - 1, -NewValue_Temp)
    grR.DrawLine(Pens.YellowGreen, PicGraphRGP.Width - 1 - Xmove, -
OldValue_RGP, PicGraphRGP.Width - 1, -NewValue_RGP)

' Passo 6
' O valor recém plotado passa a ser considerado com valor antigo.
OldValue_Ecg = NewValue_Ecg
OldValue_Temp = NewValue_Temp
OldValue_RGP = NewValue_RGP

' Passo 7
' Mostra os resultado colocando este Bitmap editado como imagem no
PictureBox.
PicGraph.Image = bm
PicGraphTemp.Image = bmT
PicGraphRGP.Image = bmR

' Liberação dos objetos gráficos
gr.Dispose()
grT.Dispose()
grR.Dispose()

End Sub

' Rotina chamada quando o botão Iniciar é pressionado
Private Sub Iniciar_B_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Iniciar_B.Click

    Simula = False

' Zera os vetor onde os dados convertidos serão guardados
For ic As Short = 0 To 100
    Vetor_Conv_Ecg(ic) = 0
    Vetor_Conv_Temp(ic) = 0
    Vetor_Conv_Rgp(ic) = 0
Next

' Habilita a porta serial
PortaSerial.Open()
' Especifica que os valores serão recebidos em numeros binarios
PortaSerial.WriteLine("\")

' "Liga" os timers
Timer_Ecg.Enabled = True
Timer_Temp.Enabled = True
Timer_Rgp.Enabled = True
Timer_Plotagem.Enabled = True

End Sub

```

```

' Rotina chamada quando o botão Simular é pressionado
Private Sub Simular_B_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Simular_B.Click

    Timer_Plotagem.Interval = 5
    Timer_Plotagem.Enabled = True
    Simula = True
    Simulação()

End Sub

' Rotina chamada quando o botão Parar é pressionado
Private Sub Parar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotaoPara.Click
    'Para os gráficos parando os timers
    Timer_Ecg.Enabled = False
    Timer_Temp.Enabled = False
    Timer_Rgp.Enabled = False
    Timer_Plotagem.Enabled = False
End Sub

' Rotina chamada quando o botão Recomeçar é pressionado
Private Sub Recomecar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotaoCorre.Click
    If Simula = True Then
        Timer_Plotagem.Enabled = True
    Else
        Timer_Ecg.Enabled = True
        Timer_Temp.Enabled = True
        Timer_Rgp.Enabled = True
        Timer_Plotagem.Enabled = True
    EndIf
End Sub

' Fim do programa.
End Class

```