

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELÉTRICA  
CURSO DE ENGENHARIA ELÉTRICA

ALLAN MAGNO COSTA TEIXEIRA

**DESENVOLVIMENTO DE PROTÓTIPO PARA AQUISIÇÃO E EXIBIÇÃO  
DOS SINAIS DE OXIMETRIA DE PULSO**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2016

ALLAN MAGNO COSTA TEIXEIRA

## **DESENVOLVIMENTO DE PROTÓTIPO PARA AQUISIÇÃO E EXIBIÇÃO DOS SINAIS DE OXIMETRIA DE PULSO**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia Elétrica do Departamento Acadêmico de Elétrica – DAELE – da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Me. Artur Tsuguiyoshi Hara

PATO BRANCO

2016

## **TERMO DE APROVAÇÃO**

O trabalho de Conclusão de Curso intitulado “DESENVOLVIMENTO DE PROTÓTIPO PARA AQUISIÇÃO E EXIBIÇÃO DOS SINAIS DE OXIMETRIA DE PULSO”, do acadêmico Allan Magno Costa Teixeira foi considerado APROVADO de acordo com a ata da banca examinadora N° 127 de 2016.

Fizeram parte da banca os professores:

**Artur Tsuguiyoshi Hara**

**Fabio Luiz Bertotti**

**Edwin Choque Pillco**

## DEDICATÓRIA

Dedico este trabalho primeiramente à Deus, pois eu nada seria sem a fé que tenho nele, à minha família, que sempre me amparou durante toda essa jornada.

Às memórias de meu avô Miguel Joaquim da Costa, que me aconselhou de minha infância até a juventude, sempre se apresentando como um exemplo a ser seguido.

## **AGRADECIMENTOS**

É certo que não conseguirei agradecer a todos que contribuiriam direta e indiretamente em minha caminhada acadêmica, durante esses cinco anos na academia. Logo, peço desculpas aqueles que não estão presentes nessas palavras, mas a eles fica aqui expressa minha plena gratidão e reitero que foram essenciais para que essa fosse uma jornada próspera.

Agradeço primeiramente a Deus, aos meus pais, Edivar e Arenilza, que não mediram esforços para que fosse possível minha caminhada acadêmica, à minha irmã, Ellen, a qual sempre esteve presente me auxiliando, à minha namorada, Sílvia, que esteve ao meu lado durante os diversos momentos que presenciei em minha jornada acadêmica, aos demais familiares, que fizeram parte de minha vida antes e após o ingresso na universidade, e aos meus amigos, que sempre se mostraram solícitos e foram essenciais na minha formação acadêmica e pessoal.

Ao professor Ricardo Bernardi, por ter sugerido o tema e me auxiliado durante a elaboração desse projeto, ao meu orientador Artur Tsuguiyoshi Hara, pela orientação e apoio, ao professor César Augusto Portolann, por colocar a minha disposição o laboratório do Bloco F da UTFPR – Pato Branco, durante todo o período que desenvolvi esse trabalho, à Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, pela disponibilidade de infraestrutura e material, agradeço também, de modo geral, aos funcionários da universidade, pelo apoio e colaboração nos dias de ensino que estive no câmpus.

## RESUMO

TEIXEIRA, Allan Magno Costa. **Desenvolvimento de protótipo para aquisição e exibição dos sinais de oximetria de pulso**. 2016. 127 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Este trabalho consiste no desenvolvimento de um protótipo para aquisição e exibição dos sinais de oximetria de pulso, apresentando o valor percentual de  $SpO_2$ , da quantidade de BPM e também da curva fotopleletismografica, informações essas as quais são exibidas através de uma tela gráfica de um display 128x64 pixels, o qual teve sua biblioteca desenvolvida ao decorrer do trabalho. O protótipo de oxímetro de pulso também dispõe de um indicador sonoro dos batimentos cardíacos, com intensidade variável em função do valor percentual de  $SpO_2$ , aumentando seu nível à medida que a saturação percentual de oxigênio diminui. A unidade microcontrolador que compõe a central do protótipo é um MSP430FR5969, a qual opera: o controle da ponteira, a qual também fora desenvolvida e baseia-se em um sistema não invasivo composto por um par de LEDs (vermelho e infravermelho) e um fotodiodo OPT101; na aquisição de dados através do conversor A/D de 12 bits; o controle do visor por meio de uma interface de comunicação serial; e também a comunicação UART com o computador, utilizando-se do kit de desenvolvimento e de uma rotina de recepção de dados implementada no programa MATLAB®.

**Palavras-chave:** Oxímetro de pulso, curva fotopleletismografica,  $SpO_2$ , BPM, não invasivo, comunicação UART.

## ABSTRACT

TEIXEIRA, Allan Magno Costa. **Development of prototype for the acquisition and display of pulse oximetry signals**. 2016. 127 f. Graduation Conclusion Work (Graduation in Electric Engineering) – Electrical Engineering, Federal Technological University of Paraná. Pato Branco, 2016.

This paper presents a prototype developed for acquisition and display of pulse oximetry signals, it shows percentage SpO<sub>2</sub> value, the BPM value and photoplethysmogram, those informations are displayed using the graphical display of 128x64 pixels, a library has been developed during this work. The prototype also features an heartbeat audible indicator with intensity depending on the percentage of SpO<sub>2</sub>, increasing its level as the percent of oxygen saturation decreases. The microcontroller of the prototype is a MSP430FR5969. It take care of the acquisition of data through the 12-bit ADC; the display control through of a serial communication interface; also, the UART communication with the computer, using a development kit and using a data reception script implented in the software MATLAB®. The microprocessor also controls the probe, which was develeoped and based on a non-invasive system. The probe is composed by a couple of LEDs (red and infrared) and one phododiode OPT101.

**Palavras-chave:** Pulse oximeter, photopletismography, SpO<sub>2</sub>, BPM, noninvasive, UART communication.

## LISTA DE FIGURAS

Figura 1 - Curva de dissociação da Oxiemoglobina, para $P_{CO_2} = 46\text{mmHg}$ .....	18
Figura 2 - Absorção da Hemoglobina (Hb) e Oxiemoglobina ( $HbO_2$ ), em relação ao comprimento de onda da luz incidente.....	19
Figura 3 - Disposição dos LEDs e do fotosensor, em relação à extremidade do dedo.....	19
Figura 4 - Fotopletismograma.....	20
Figura 5 - Comparação entre a curva obtida por meio da lei de Beer-Lambert e a calibração obtida empiricamente.....	22
Figura 6 - "Picos e vales" da curva pletismográfica para determinação da saturação.....	23
Figura 7 - Diagrama da estrutura que compõe o protótipo do oxímetro.....	27
Figura 8 - Ponteira (sensor) do protótipo de oxímetro.....	28
Figura 9 - Projeto 3D da estrutura da ponteira.....	29
Figura 10 - Esquemático do sistema de acionamento dos LEDs.....	30
Figura 11 - Placa de acionamento dos LEDs.....	30
Figura 12 - Fluxograma referente ao acionamento dos LEDs.....	32
Figura 13 - Diagrama de bloco do OPT101.....	33
Figura 14 - Resposta aos espectros de frequência da luz.....	34
Figura 15 - Esquemático do circuito do fotosensor.....	35
Figura 16 - Placa do fotosensor.....	35
Figura 17 - Regulador de tensão de saída fixa.....	36
Figura 18 - Regulador de tensão ajustável.....	37
Figura 19 - Circuito de alimentação.....	38
Figura 20 - Display gráfico de LCD 128x64 pixels.....	39
Figura 21 - Fluxograma do processo para setar o registrador de instruções.....	41
Figura 22 - Fluxograma do processo para setar o registrador de endereços.....	41
Figura 23 - Estrutura do byte de envio de "RW".....	42
Figura 24 - Diagrama da operação de separação dos 4 bits mais significativos e dos 4 bits menos significativos (deslocados).....	42
Figura 25 - Fluxograma para escrever dados no display.....	43
Figura 26 - Estrutura do byte de envio de "RW" e "RS".....	43
Figura 27 - Fluxograma da inicialização do display 128x64.....	44
Figura 28 - Mapeamento do display 128x64.....	45
Figura 29 - Pastilha piezo.....	46
Figura 30 - LaunchPad MSP430FR5969.....	48
Figura 31 - Diagrama de Pinos do MSP430FR5969 48-pinos RGZ.....	49
Figura 32 - Esquemático da Placa-Mãe.....	51
Figura 33 - Placa-Mãe.....	51
Figura 34 - Diagrama de blocos de um sistema de aquisição de um sinal analógico para realização do processamento das informações.....	52
Figura 35 - Amostragem do Filtro Média Móvel.....	53

Figura 36 - Fluxograma do filtro digital média móvel.....	55
Figura 37 - Fluxograma do algoritmo de detecção de picos e vales .....	56
Figura 38 - Gerador de função, ajustado para uma tensão mínima de 1,8 V e máxima de 2,2 V, com frequência de 0,66667 Hz, e visor do protótipo de oximetria .....	58
Figura 39 - Teste de operação do protótipo .....	61
Figura 40 - Gráfico das amostras de SpO <sub>2</sub> do Indivíduo 1, anotadas ao decorrerem do tempo, com auxílio do protótipo e de um dispositivo comercial.....	62
Figura 41 - Gráfico das amostras de SpO <sub>2</sub> do Indivíduo 2, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo.....	63
Figura 42 - Gráfico das amostras de BPM do Indivíduo 1, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo.....	65
Figura 43 - Gráfico das amostras de BPM do Indivíduo 2, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo.....	66
Figura 44 - Curva fotopleletismografica enviada do microcontrolador para o computador, referente ao LED infravermelho.....	67
Figura 45 - Forma de onda na entrada do conversor A/D, correspondente ao LED Infravermelho .....	68
Figura 46 - Curva fotopleletismografica enviada do microcontrolador para o computador, referente ao LED vermelho .....	69
Figura 47 - Forma de onda na entrada do conversor A/D, correspondente ao LED Vermelho .....	69

## LISTA DE TABELAS

Tabela 1 - Lógica de acionamento dos LEDs.....	31
Tabela 2 - Dados coletados no ensaio da operação do medidor de BPM, utilizando-se de um gerador de funções.....	59
Tabela 3 – Dados de SpO <sub>2</sub> coletados do Indivíduo 1 .....	61
Tabela 4 - Dados de SpO <sub>2</sub> coletados do Indivíduo 2.....	63
Tabela 5 - Dados de BPM coletados do Indivíduo 1 .....	64
Tabela 6 - Dados de BPM coletados do Indivíduo 2 .....	65

## LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico para Digital
BPM	Batimentos Por Minuto
CI	Circuito Integrado
CS	Seleção do Chip (do original <i>Chip Select</i> )
FIR	Filtro digital de resposta ao impulso de duração finita (do original <i>Finite Impulse Response</i> )
IIR	Filtro digital de resposta ao impulso de duração infinita (do original <i>Infinite Impulse Response</i> )
LCD	Display de Cristal Líquido (do original <i>Liquid Crystal Display</i> )
LED	Diodo Emissor de Luz (do original <i>Light Emitting Diode</i> )
PWM	Modulação de Largura de Pulso (do original <i>Pulse Width Modulation</i> )
SPI	Interface Serial Periférica (do original <i>Serial Peripheral Interface</i> )
UART	Receptor/Transmissor Universal Assíncrona (do original <i>Universal Asynchronous Receiver/Transmitter</i> )
RS	Registrador de Seleção (do original <i>Register Select</i> )
RW	Ler/Escrever (do original <i>Read/Write</i> )

## LISTA DE SÍMBOLOS

$\text{CO}_2$	Dióxido de carbono
Hb	Hemoglobina molecular
$\text{HbO}_2$	Oxiemoglobina molecular
$\text{O}_2$	Dois isótopos de oxigênio
$\text{P}_{\text{O}_2}$	Pressão parcial de oxigênio
$\text{SpO}_2$	Saturação pulsátil de oxigênio
$\text{P}_{\text{CO}_2}$	Pressão parcial de dióxido de carbono

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 JUSTIFICATIVA .....	14
1.2 OBJETIVO GERAL .....	15
1.3 OBJETIVOS ESPECÍFICOS .....	15
1.4 ORGANIZAÇÃO DO TRABALHO .....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>17</b>
2.1 OXIMETRIA.....	17
2.2 PRINCÍPIOS DA OXIMETRIA DE PULSO .....	18
2.3 CÁLCULO DE SPO <sub>2</sub> .....	20
2.4 MICROCONTROLADORES.....	24
2.5 SENSORES .....	25
2.6 FILTROS DIGITAIS.....	25
<b>3 O PROJETO</b> .....	<b>27</b>
3.1 SENSOR DE OXIMETRIA.....	28
<b>3.1.1 Circuito de acionamento dos LEDs</b> .....	<b>29</b>
<b>3.1.2 Sensor OPT101</b> .....	<b>32</b>
3.2 FONTE DE ALIMENTAÇÃO.....	36
3.3 DISPLAY 128X64 .....	39
3.4 INDICADOR SONORO .....	46
3.5 MICROCONTROLADOR MSP430FR5969 .....	47
3.6 PLACA-MÃE.....	50
3.7 PROCESSAMENTO DIGITAL DOS SINAIS .....	52
<b>3.7.1 Filtro de média móvel</b> .....	<b>52</b>
<b>3.7.2 Detecção dos picos e vales</b> .....	<b>55</b>
3.8 SISTEMA DE COMUNICAÇÃO COM COMPUTADOR .....	56
3.9 RESULTADOS E DISCUSSÕES .....	57
<b>4 CONCLUSÕES</b> .....	<b>70</b>
<b>REFERÊNCIAS</b> .....	<b>73</b>
<b>APÊNDICE A - LAYOUT DAS PLACAS</b> .....	<b>75</b>
<b>APÊNDICE B - CÓDIGO PRINCIPAL “MAIN.C”</b> .....	<b>76</b>
<b>APÊNDICE C - BIBLIOTECA DESENVOLVIDA PARA O DISPLAY</b> <b>“BIBLIOTECA_LCD.C”</b> .....	<b>96</b>
<b>APÊNDICE D - “BIBLIOTECA_LCD.H”</b> .....	<b>120</b>
<b>APÊNDICE E - ARQUIVO DE INICIALIZAÇÃO DO DISPLAY</b> <b>“INICIALIZA_DISPLAY.C”</b> .....	<b>121</b>
<b>APÊNDICE F - “INICIALIZA_DISPLAY.H”</b> .....	<b>122</b>
<b>APÊNDICE G - BIBLIOTECA DESENVOLVIDA PARA COMPOR A ESTRUTURA</b> <b>DOS NÚMEROS “NUMEROS.H”</b> .....	<b>123</b>
<b>APÊNDICE H - CÓDIGO IMPLEMENTADO NO MATLAB® PARA AQUISIÇÃO E</b> <b>PROCESSAMENTO DOS DADOS ENVIADOS PELO MICROCONTROLADOR</b> .	<b>125</b>

## 1 INTRODUÇÃO

A medicina, historicamente, vem buscando meios para melhorar a capacidade de monitoramento dos sinais vitais humanos, com intuito de utilizá-los durante procedimentos cirúrgicos (e também no pós-cirúrgico), tratamentos preventivos, diagnósticos, entre outras aplicações, pertinentes à área médica. Vislumbrando esse horizonte, trabalha-se em prol do desenvolvimento de dispositivos que viabilize e otimize a aquisição de dados do organismo de pacientes, almejando disponibilizar, aos profissionais que necessitam de tais informações, equipamentos de operação prática, precisos e dinâmicos.

Visando prolongar a expectativa de vida humana, buscou-se métodos para análise e/ou monitoramento do funcionamento do sistema vital, os quais possibilitaram a detecção das causas de algum tipo de patologia, podendo assim atuar objetivamente no tratamento ou até mesmo identificar organismos que apresentavam características que indicavam pré-disposição ao desenvolvimento de tal quadro, tornando passível a implantação de um tratamento preventivo. Através da evolução desses métodos, ao decorrer de décadas, conseguiu-se melhorar a qualidade e expectativa de vida.

Frente a essa necessidade, em meio aos avanços tecnológicos, nas áreas eletrônica e digital, advindos do século XX, fez-se necessário o sombreamento entre essas áreas de conhecimento, visando impulsionar o desenvolvimento tecnológico dos métodos de análise e monitoramento de pacientes. Essa evolução fez surgir novas áreas de estudos, na qual tem-se como exemplo a bioengenharia.

No início do século XX, em um artigo escrito por Matthes (1935), fora realizado um estudo, referente ao monitoramento dos níveis de saturação do oxigênio arterial. Com base nos resultados obtidos, constatou-se a necessidade de se monitorar e registrar os níveis de saturação de oxigênio arterial. A experimentação consistiu da análise dos distúrbios detectados em cobaias animais, durante o período em que os indivíduos se encontram sob efeito do procedimento anestésico, geralmente empregado em cirurgias, mostraram que ocorrem alterações no ritmo respiratório e, conseqüentemente, geram-se distúrbios nos níveis de saturação de oxigênio (MATTHES, 1935).

Em meio à constatação da necessidade de se monitorar os níveis de oxigênio no sangue, desenvolveu-se um método para mensurar a saturação de

oxigênio do sangue, o qual fora elaborado por Hünter. Para realização da medição, era necessário a coleta de uma amostra, de maneira invasiva, que permitia apenas a análise de uma única amostragem, a qual era inviável de ser repetida muitas vezes, pelo fato de ser invasiva, e demandava de muito tempo (MATTHES, 1935).

Em decorrência da problemática levantada, surgiu, como método alternativo, o oxímetro, destacando-se por não ser invasivo. Essa nova abordagem possibilitou inúmeras amostragens, de maneira sequenciada e em curtos intervalos de tempo. A obtenção dos sinais necessários para determinação dos níveis de saturação do sangue arterial periférico é feita através da ponta do dedo ou do lóbulo da orelha (KUSHIMA, 2014).

De modo geral, a estrutura de um oxímetro digital consiste em um par de diodos emissores de luz, sendo um com emissão de luz de comprimento de onda de 660 nm (faixa vermelha) e outro emite uma onda de comprimento de onda de 895 nm (faixa infravermelho), e um fotodiodo, o qual irá fazer a detecção dos níveis de luz resultantes da emissão de feixes contra algumas partes específicas do corpo humano. Isso gera uma absorção desses comprimentos de onda, gerando uma diferença entre a oxiemoglobina e sua forma desoxigenada. Através dessa variação entre os níveis consegue-se estimar a taxa de saturação de oxigênio (SANTINI, 2010).

## 1.1 JUSTIFICATIVA

Os oxímetros digitais são frequentemente utilizados nas unidades de terapia intensiva, sendo considerado o quinto sinal vital dentro da medicina moderna. Por não ser um método invasivo e com boa precisão, o oxímetro digital torna-se cada vez mais importante na medicina. Este dispositivo também ganha mais destaque por dispor de uma velocidade considerável para se conseguir realizar medições, sendo utilizados por indivíduos que demandem de um acompanhamento dos níveis de oxigenação a serem injetados adicionalmente, tais como indivíduos com problemas respiratórios e alguns pilotos de grupos específicos de aeronaves (SANTINI, 2010).

Diante de tal cenário, faz-se relevante o desenvolvimento de um oxímetro digital de pulso que possua uma interface gráfica que apresente o sinal de fotopletismografia e a taxa percentual de saturação de oxigênio no sangue, propiciando aos profissionais que irão operar tais equipamentos, o acesso a um meio de apresentação de informações vitais dos dados do paciente mais intuitivo e de fácil

compreensão. Integrado ao sistema, é oportuno, e torna mais prático, a implantação de um sinalizador sonoro, o qual terá como função indicar o ritmo do pulso e, de modo intuitivo, através da variação na intensidade do som, de tal modo que a intensidade eleve-se conforme os níveis percentuais vão ficando críticos, possibilitando uma rápida e prática detecção de possíveis problemas com os sinais vitais do paciente monitorado.

Arelado as partes do projeto, é válido o desenvolvimento de um sistema que apresente um funcionamento apropriado, de modo que venha apresentar um sistema com bons níveis de estabilidade e confiabilidade, pois seu emprego pode ser crucial no suporte a vida humana, e também possua um consumo de energia reduzido.

## 1.2 OBJETIVO GERAL

Desenvolver protótipo de oxímetro de pulso, apresentando sinal de oximetria, as formas de ondas de fotopleletismografia, taxa percentual estimada da saturação de oxigênio do sangue arterial, informação numérica da frequência cardíaca, ambos indicados através de um visor gráfico, atrelados à indicação sonora do ritmo do pulso, com intensidade variável.

## 1.3 OBJETIVOS ESPECÍFICOS

São objetivos específicos do projeto de trabalho de conclusão de curso, os seguintes itens descritos:

- 1) Projetar e implementar sensor óptico, não invasivo, para obtenção da oximetria do pulso;
- 2) Projetar e implementar o sistema de controle do sensor óptico através de um microcontrolador;
- 3) Desenvolver e implementar código para aquisição e tratamento das informações obtidas através do sensor, por meio de microcontrolador;
- 4) Implementar algoritmo para estimar o nível percentual de saturação de oxigênio do sangue arterial;
- 5) Desenvolver e implementar interface de apresentação de dados para monitoramento, através de uma tela de cristal líquido, indicando o nível

percentual da saturação de oxigênio e exibindo a forma de onda proveniente do sinal da oximetria;

- 6) Projetar e implementar indicador sonoro do ritmo do pulso, variando gradativamente a intensidade, conforme a alteração nos níveis de oxigênio, sendo menos intensa quando está dentro do normal, e elevando-se quando o nível de oxigênio estiver reduzindo.

#### 1.4 ORGANIZAÇÃO DO TRABALHO

Das disposições dos capítulos, o Capítulo 2 consiste de uma abordagem teórica dos conceitos de fundamentação da oximetria, bem como os processos ópticos e fisiológicos envolvidos, seguido da abordagem específica da oximetria de pulso, abordado os processos fisiológicos envolvidos, e da análise matemática do levantamento da taxa de saturação de oxigênio, bem como as aproximações matemáticas para otimização do processamento dos dados adquiridos via sensor. É realizada, também nesse mesmo capítulo, a apresentação e comentários acerca dos dispositivos utilizados para a elaboração do protótipo, descrevendo e elencando as características julgadas relevantes e as opções adotadas para os equipamentos utilizados.

No Capítulo 3 é descrito o projeto e implementação dos circuitos eletrônicos e das rotinas de código utilizadas, apresentados o desenvolvimento setorizado de cada um dos periféricos que compõe o dispositivo, até o microcontrolador e a placa-mãe que integra todo o projeto, apresentando na sequência a estrutura de processamento dos sinais. Ainda no Capítulo 3, seu fechamento se dá no tópico onde se tem a apresentação dos resultados e dados obtidos.

Já no encerramento, o Capítulo 4 contém as conclusões obtidas, seguido das referências utilizadas para elaboração do trabalho e apêndices, contendo os layouts das placas produzidas e os códigos produzidos.

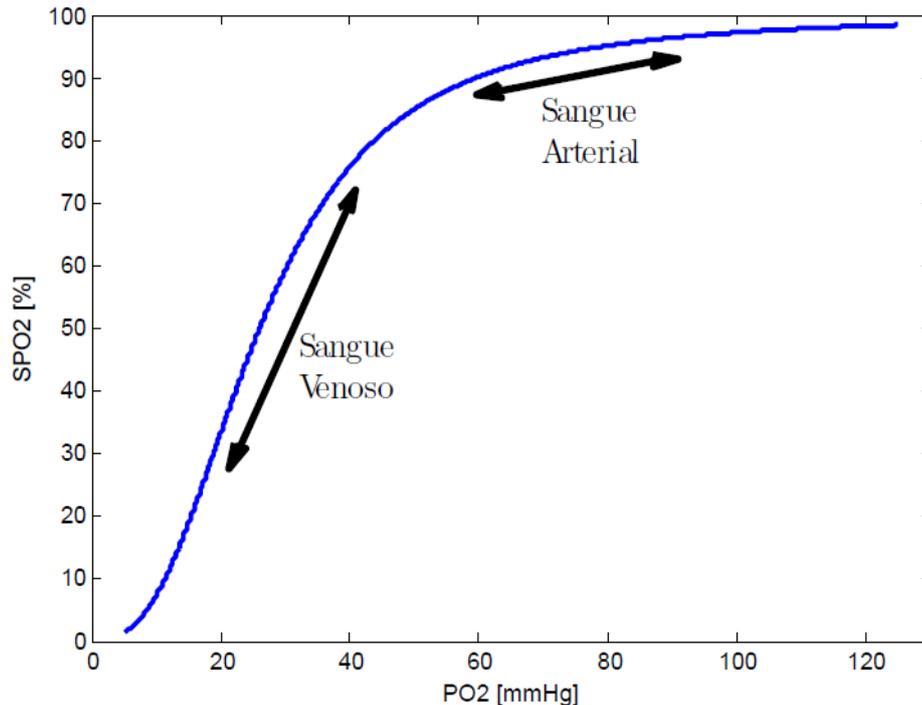
## 2 FUNDAMENTAÇÃO TEÓRICA

Para a elaboração deste trabalho, faz-se necessária a análise e revisão do material teórico referente ao assunto abordado, visando angariar conhecimento suficiente para o desenvolvimento do protótipo de oximetria de pulso.

### 2.1 OXIMETRIA

O organismo de um ser humano possui em seu processo de respiração, o estágio de oxigenação das células. Tal etapa é realizada através do transporte de gás oxigênio ( $O_2$ ) pelas moléculas de hemoglobina, as quais estão contidas nas células sanguíneas denominadas hemácias ou eritrócitos. A junção da hemoglobina com o oxigênio faz com que as moléculas recebam a denominação de oxiemoglobina,  $HbO_2$ , já as moléculas que não estão carregadas recebem o nome de hemoglobina reduzida, representada por Hb. A hemoglobina tem capacidade de transportar até quatro moléculas de oxigênio. Essa quantidade é responsável por definir o nível de saturação (KUSHIMA, 2014).

Analisando a capacidade de transporte de moléculas de oxigênio no sangue, as células contidas em uma grama de oxiemoglobinas são capazes de transportar juntas, até  $1,34 \text{ cm}^3$  de  $O_2$ . Portanto, esse volume é correspondente ao máximo percentual de saturação de oxigênio no sangue, representado pela sigla  $SpO_2$ . Porém, assim como apresentado por Santini (2010), a presença das moléculas de oxigênio no interior da estrutura das hemoglobinas é dependente da pressão parcial do próprio gás. O autor também expõe o fato de que a quantidade de dióxido de carbono ( $CO_2$ ) se apresenta de maneira aproximadamente constante nos tecidos. Portanto, existe uma relação entre a saturação percentual de oxigênio ( $SpO_2\%$ ) e a pressão parcial de oxigênio ( $P_{O_2}$ ). Em situações em que  $P_{O_2}$  seja maior que 20 mmHg e que a pressão parcial de dióxido de carbono ( $P_{CO_2}$ ) mantem-se aproximadamente constante, tal fato pode ser observado através da Figura 1 (SANTINI, 2010).



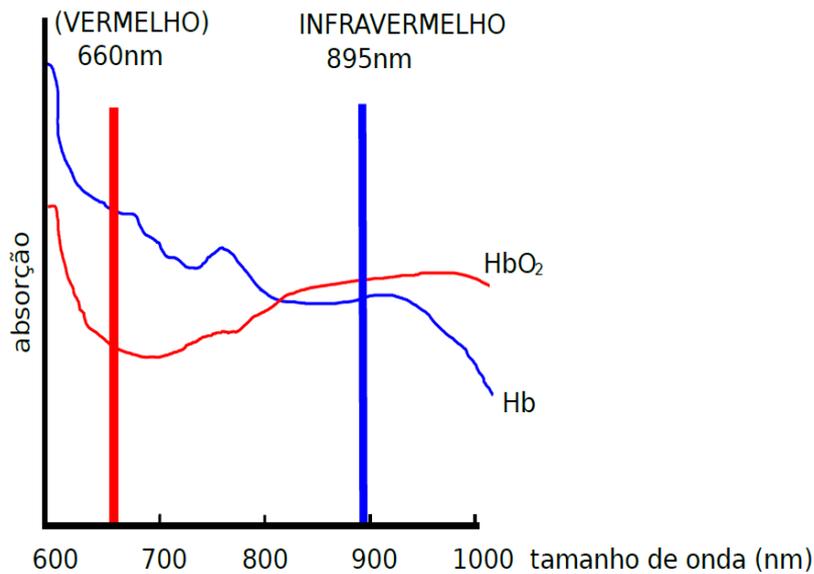
**Figura 1 - Curva de dissociação da Oxiemoglobina, para  $P_{CO_2} = 46\text{mmHg}$**   
**Fonte: Santini (2010).**

A curva apresentada na Figura 1 é conhecida como curva de dissociação da oxiemoglobina, onde se tem uma elevação de  $P_{O_2}$  em função do aumento da quantidade de moléculas de oxigênio, aumento esse que é demonstrado pela elevação do nível de  $SpO_2$  (KUSHIMA, 2014).

## 2.2 PRINCÍPIOS DA OXIMETRIA DE PULSO

Para a obtenção dos sinais de oximetria de maneira não invasiva, o oxímetro de pulso consiste em um sensor composto por diodos emissores de Luz (LED, do original *Light Emitting Diode*), tipicamente um vermelho e outro infravermelho, e um fotosensor, o qual possua sensibilidade luminosa para abranger ambos os espectros e que apresente uma boa resposta à incidência luminosa.

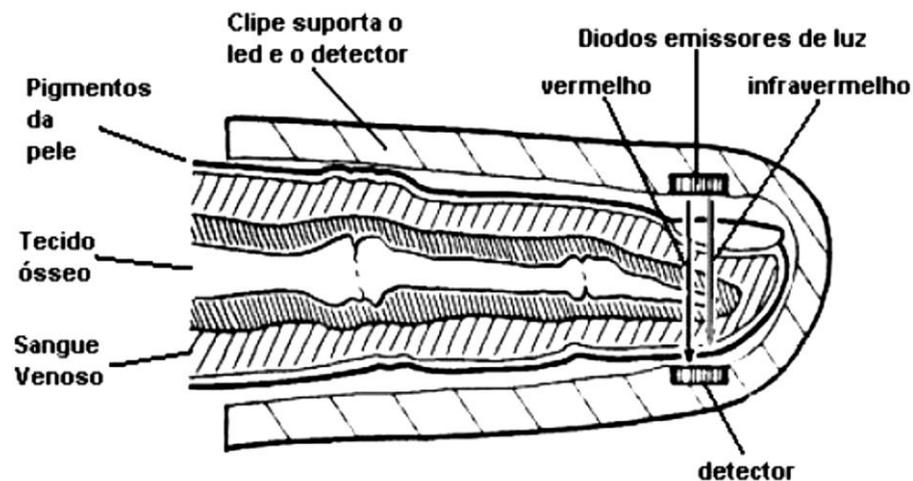
O comportamento de absorção da hemoglobina oxigenada e da desoxigenada varia de acordo com a frequência da luz incidente. Essa peculiaridade é explorada pelo princípio de medição do oxímetro de pulso, uma vez que se tem fatores de absorção distintos em algumas faixas do espectro (MARTINS, 2010). Essa constatação pode ser feita através da análise da Figura 2.



**Figura 2 - Absorção da Hemoglobina (Hb) e Oxihemoglobina (HbO<sub>2</sub>), em relação ao comprimento de onda da luz incidente**  
 Fonte: Adaptado de Santini (2010).

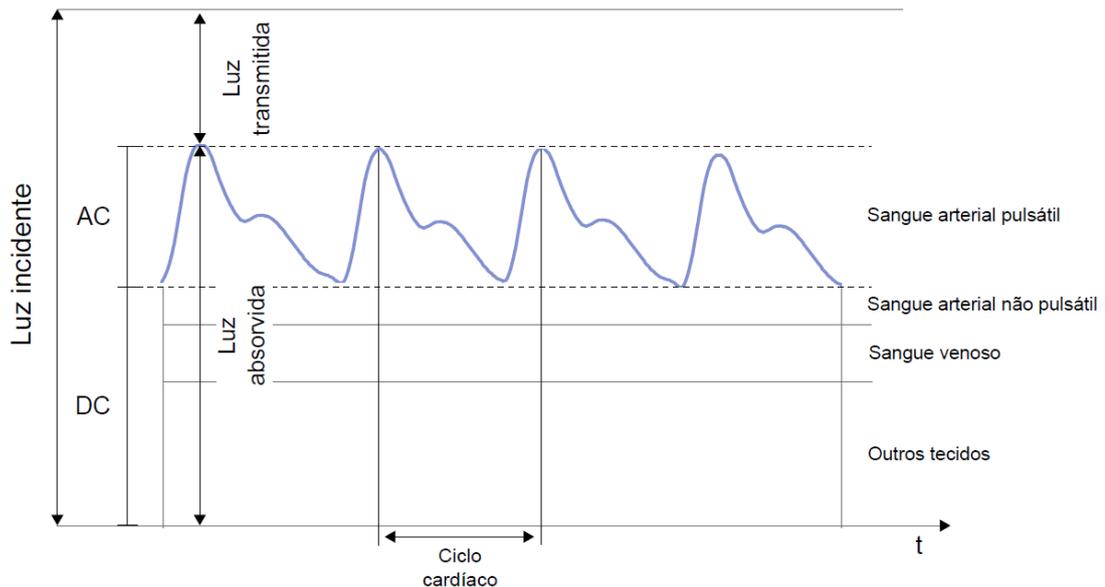
Atualmente existem dois grupos de sensores que dispõem da capacidade de operar como fotodetector nesse tipo de aplicação, os fotodiodos e os fototransistores, ambos possuem a capacidade de operar dentro do espectro utilizado e fornecem uma corrente de polarização proporcional a incidência luminosa (MARQUES; JÚNIOR; CRUZ, 1996).

A disposição dos componentes que formam o sensor, em relação à extremidade utilizada para realizar a medição, de modo geral, a ponta do dedo, dá-se como apresentado no esquema da Figura 3.



**Figura 3 - Disposição dos LEDs e do fotosensor, em relação à extremidade do dedo**  
 Fonte: Adaptado de Andrade (2009).

O feixe de luz que é emitido pelos LEDs é parcialmente absorvido pelos diversos tipos de tecidos presentes no caminho até o detector. A luz transmitida passa através do sistema vascular periférico pulsátil, o que origina o termo oxímetro de “pulso”, e tem parte do sinal absorvido de forma variável, em decorrência da pulsação arterial, assim como exposto na Figura 4.



**Figura 4 - Fotopletismograma**  
**Fonte: Martins (2010).**

Através da absorção dos comprimentos de onda pelo sangue arterial pulsátil, torna-se possível estimar no nível de  $SpO_2$ . Essa premissa fica evidente quando observada a Figura 4, a qual apresenta a onda fotopletismografica, onde se tem a indicação de que a variação do nível de absorção da luz incidente é causada pelo sangue arterial pulsátil.

O fotopletismograma também carrega, em si, informações referentes ao ciclo cardíaco, tornando-se possível estimar a frequência cardíaca do indivíduo, bastando identificar a frequência com que o sinal oscila (MARTINS, 2010). Tal fato está indicado na Figura 4, onde se tem o período indicado pela seta denominada “Ciclo cardíaco”.

### 2.3 CÁLCULO DE $SPO_2$

Durante o processo de absorção dos feixes de luz, existe o nível de absorção não constante, gerado principalmente pela pulsação arterial. Isso implica em

uma variação de aproximadamente 1 % a 2 %. O efeito da dilatação dos vasos acarreta na variação do comprimento do caminho que o feixe percorre até chegar ao sensor. Essa situação gera a necessidade de conhecer a distância do caminho percorrido no tecido, bem como a intensidade luminosa incidente, segundo a lei de Beer-Lambert (LIMA, 2009).

O princípio definido pela lei de Beer-Lambert, em um relato sucinto, relaciona o nível de absorção de um feixe de luz com as propriedades moleculares do material que o recebe, em uma abordagem experimental. Essa relação depende de algumas variáveis, como o comprimento de onda do feixe incidente, o comprimento de onda que o material absorve e a distância que o feixe percorre até transpassar o material. É válido ressaltar que, ao se adotar a estratégia de transpassar um material com um feixe luminoso, o material não deve ser totalmente opaco, do contrário isso inviabilizaria a análise (KUSHIMA, 2014).

Para minimizar a necessidade de se conhecer a intensidade emitida, a determinação do nível percentual de saturação de oxigênio baseia-se nos valores da intensidade luminosa detectados pelo sensor. Porém, como se têm uma diferença na amplitude dos sinais obtidos através da excitação proporcionada pelo LED infravermelho e vermelho (sendo o primeiro de maior amplitude), faz-se necessária a normalização dos valores. Esse procedimento é realizado através da divisão do valor amostrado pelo valor máximo. Feito isso, para determinar a razão entre as absorbâncias da luz, calcula-se o logaritmo natural da razão entre as absorbâncias de cada tipo de frequência, em seu nível máximo e mínimo, conseguindo assim a independência do caminho óptico. Dessa forma, a razão entre as absorbâncias pode ser dada por (LIMA, 2009):

$$R = \frac{\varepsilon_{Hb}(\lambda_V) + [\varepsilon_{HbO_2}(\lambda_V) - \varepsilon_{Hb}(\lambda_V)] \times SpO_2}{\varepsilon_{Hb}(\lambda_{IV}) + [\varepsilon_{HbO_2}(\lambda_{IV}) - \varepsilon_{Hb}(\lambda_{IV})] \times SpO_2} \quad (1)$$

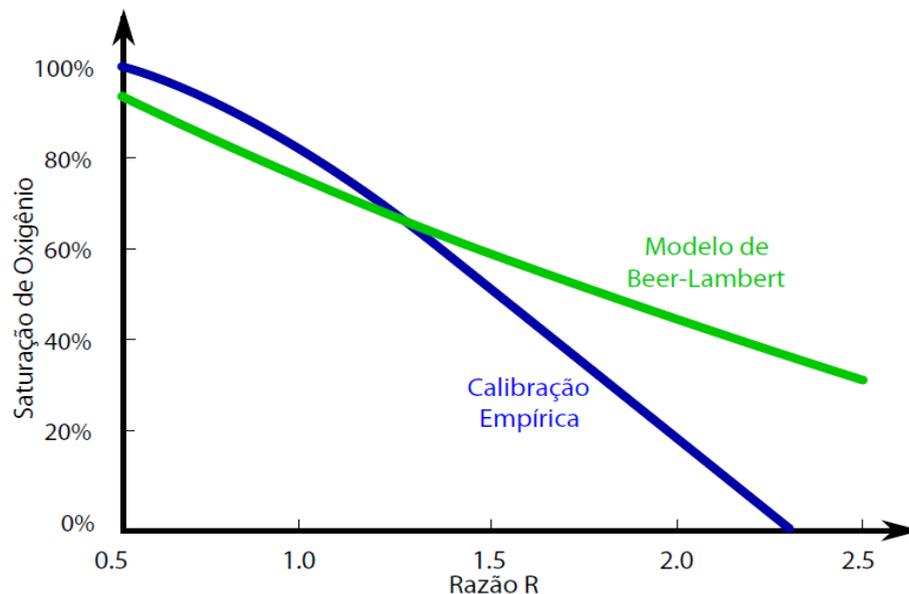
onde  $\varepsilon_{Hb}(\lambda_V)$  é o coeficiente de extinção da substância Hb em um comprimento de onda  $\lambda_V$ ;  $\lambda_V$  é o comprimento de onda da cor vermelha;  $\varepsilon_{Hb}(\lambda_{IV})$  é o coeficiente de extinção da substância Hb em um comprimento de onda  $\lambda_{IV}$ ;  $\lambda_{IV}$  é o comprimento de onda da cor infravermelho;  $\varepsilon_{HbO_2}(\lambda_V)$  é o coeficiente de extinção da substância HbO<sub>2</sub> em um comprimento de onda  $\lambda_V$ ;  $\varepsilon_{HbO_2}(\lambda_{IV})$  é o coeficiente de extinção da substância

HbO<sub>2</sub> em um comprimento de onda  $\lambda_{IV}$ ;  $SpO_2$  é a saturação de oxigênio; e  $R$  é a razão entre as absorvâncias dos comprimentos de onda.

Assim, têm-se que, para determinação do nível de saturação percentual do sangue através do modelo teórico, a equação expressa com dependência dos seguintes termos (LIMA, 2009):

$$SpO_2 [\%] = \frac{\varepsilon_{Hb}(\lambda_V) - \varepsilon_{Hb}(\lambda_{IV}) \times R}{\varepsilon_{Hb}(\lambda_V) - \varepsilon_{HbO_2}(\lambda_V) + [\varepsilon_{HbO_2}(\lambda_{IV}) - \varepsilon_{Hb}(\lambda_{IV})] \times R} \times 100\% \quad (2)$$

O modelo teórico apresentado é demasiadamente dispendioso e apresenta a necessidade de se conhecer uma quantidade significativa de variáveis. Em contrapartida, a calibração do oxímetro pode ser realizada através de uma curva empírica, a qual relaciona a razão “R” com a taxa percentual de oxigênio (SANTINI, 2010). A comparação das duas curvas está apresentada na Figura 5.



**Figura 5 - Comparação entre a curva obtida por meio da lei de Beer-Lambert e a calibração obtida empiricamente**  
**Fonte: Santini (2010).**

Para se determinar a saturação de oxigênio com base na curva empírica, utiliza-se o método de “pico e vale”, onde se precisa determinar os valores de máximo e mínimo para cada uma das curvas levantadas através dos dois LEDs (KUSHIMA, 2014). A disposição desses pontos na curva do sinal pletismográfico está apresentada na Figura 6.

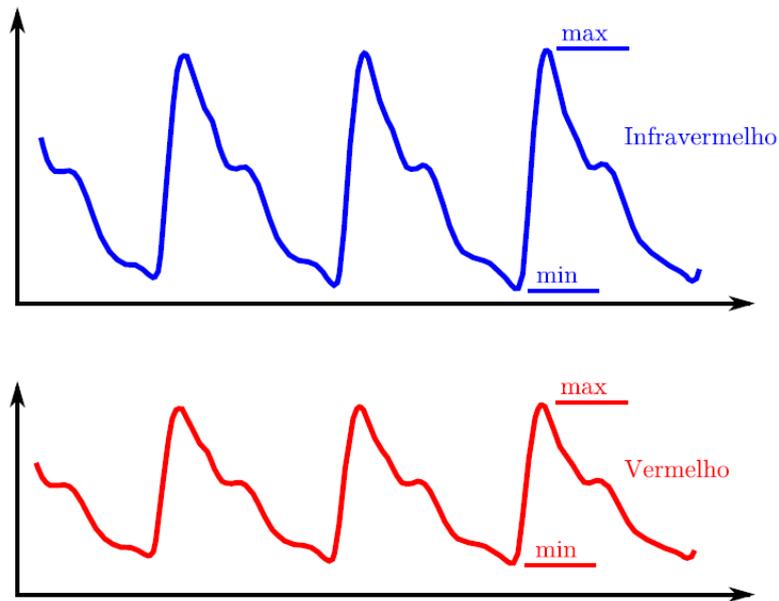


Figura 6 - "Picos e vales" da curva pletoemográfica para determinação da saturação  
Fonte: Santini (2010).

Para determinar o valor de R, basta utilizar a equação (3), adaptada de Kushima (2014):

$$R = \frac{\ln\left(\frac{I_{\text{vermelho}_{\min}}}{I_{\text{vermelho}_{\max}}}\right)}{\ln\left(\frac{I_{\text{infravermelho}_{\min}}}{I_{\text{infravermelho}_{\max}}}\right)} \quad (3)$$

onde  $I_{\text{vermelho}_{\min}}$  é a mínima intensidade luminosa do feixe vermelho;  $I_{\text{vermelho}_{\max}}$  é a máxima intensidade luminosa do feixe vermelho;  $I_{\text{infravermelho}_{\min}}$  é a mínima intensidade luminosa do feixe infravermelho; e  $I_{\text{infravermelho}_{\max}}$  é a máxima intensidade luminosa do feixe infravermelho.

Tendo obtida a expressão para determinar o valor de R, Kushima (2014) realizou o levantamento das interpolações da curva empírica, normalmente utilizadas pela literatura, a qual fornece alguns polinômios tipicamente utilizados na determinação da saturação de oxigênio, os quais são expressos por (KUSHIMA, 2014):

$$SpO_2[\%] = 110 - 25 \times R \quad (4)$$

$$SpO_2[\%] = \frac{1000 - 550 \times R}{900 - 350 \times R} \times 100 \quad (5)$$

$$SpO_2[\%] = 10,0002 \times R^3 - 52,887 \times R^2 + 26,871 \times R + 98,283 \quad (6)$$

Segundo Kushima (2014), cada uma das equações de calibração empírica apresenta um comportamento particular, em função de R, porém todas se aproximam dos valores teorizados de saturação percentual de oxigênio, respeitando uma faixa de valores que R pode assumir.

Sendo assim, as equações de calibração empírica devem ter seu desempenho analisado, de modo que se determine qual equação melhor se adapta ao equipamento, fazendo com que os valores indicados sejam mais próximos possíveis dos teóricos.

## 2.4 MICROCONTROLADORES

A evolução da eletrônica possibilitou a criação de componentes cada vez menores, chegando a dimensões nanométricas, o que possibilitou a junção de inúmeros dispositivos em um único encapsulamento. Em decorrência dessa evolução, a área da microeletrônica possibilitou o desenvolvimento de unidades programáveis, designadas para o controle e supervisão de processos lógicos.

Dentro da gama de unidades programáveis, surgiu a linha dos microcontroladores, os quais são compostos por portas de comunicação, as quais possibilitam o controle e/ou supervisão de periféricos, memória e uma unidade central de processamento, empregada na execução de instruções e processamento das informações. Alguns microcontroladores também dispõem de periféricos dentro do próprio encapsulamento, tais como: conversores A/D, sensores de temperatura, entre outros.

O fato de se ter um único encapsulamento com diversos componentes, em escala reduzida em seu interior, reduz o custo de produção e torna esse dispositivo economicamente atrativo.

## 2.5 SENSORES

A escolha do tipo de sensor dos espectros emitidos pelos LEDs fora feita com base na característica de operação. Existem dois grupos de sensores para esse tipo de aplicação, os fotodiodos e os fototransistores. Em uma análise comparativa, pode-se determinar qual componente teria um melhor desempenho e um funcionamento adequado à aplicação. Para isso levantou-se as características mais relevantes dos dois fotosensores e fez-se um comparativo.

O fototransistor destaca-se por ser um dispositivo semicondutor mais sensível, em contrapartida, sua curva de operação é pouco linear, o que acarretaria em uma maior complexidade de operação e tratamento das informações obtidas através do sensor (MARQUES; JÚNIOR; CRUZ, 1996).

No caso do fotodiodo, esse componente dispõe de uma curva com uma considerável região linear de operação. Portanto, a corrente reversa tem a mesma forma de onda da amplitude da intensidade luminosa que incide sobre sua janela, essa relação é de grande valia para a aplicação no oxímetro de pulso, uma vez que, para determinação da saturação de oxigênio, é necessário conhecer a incidência luminosa sobre o sensor. Neste caso, o componente já fornece a conversão diretamente proporcional da intensidade luminosa para a corrente de polarização reversa (MARQUES; JÚNIOR; CRUZ, 1996).

Como tipo de sensor, utiliza-se então, o fotodiodo, porém, para que se consiga fazer a aquisição desse sinal, faz-se necessária a conversão da corrente reversa para uma grandeza elétrica compatível com o conversor A/D (Analógico para Digital) do microcontrolador. Para realizar tal adequação de sinal, faz-se necessário um transdutor de corrente para tensão, essa necessidade é atendida através de um circuito com amplificador de transimpedância.

## 2.6 FILTROS DIGITAIS

Dentro da teoria de filtros, aplicada no processamento de sinais digitais, tem-se que levar em consideração o tipo de filtro digital e a forma de implementação. Entre os tipos de filtros digitais, a literatura faz a abordagem de dois principais, o filtro digital de resposta ao impulso de duração infinita (IIR, do original *Infinite Impulse*

*Response*) e o filtro digital de resposta ao impulso de duração finita (FIR, do original *Finite Impulse Response*) (INGLE; PROAKIS, 2011).

Os filtros do tipo IIR são normalmente implementados em processadores digitais de sinal (DSP, do original *Digital Signal Processor*), utilizando-se de funções racionais ou equações diferenciais elaboradas, o que acaba demandando de uma arquitetura mais específica para aplicação (INGLE; PROAKIS, 2011).

Com relação aos filtros do tipo FIR, esse tipo de filtro digital baseia-se no cálculo de média móvel, restringindo-se a operações computacionais reduzidas. Esse tipo de filtro é mais utilizado, em relação ao filtro IIR, por apresentar um comportamento estável e demandar de um hardware menos específico para a aplicação. Por esses motivos, opta-se pela escolha do filtro digital do tipo FIR, para aplicação nesse trabalho. A modelagem matemática do filtro FIR, para se obter a filtragem do sinal, é definida por (INGLE; PROAKIS, 2011):

$$y(n) = b_0 * x(n) + b_1 * x(n - 1) + \dots + b_{M-1} * x(n - M + 1) \quad (7)$$

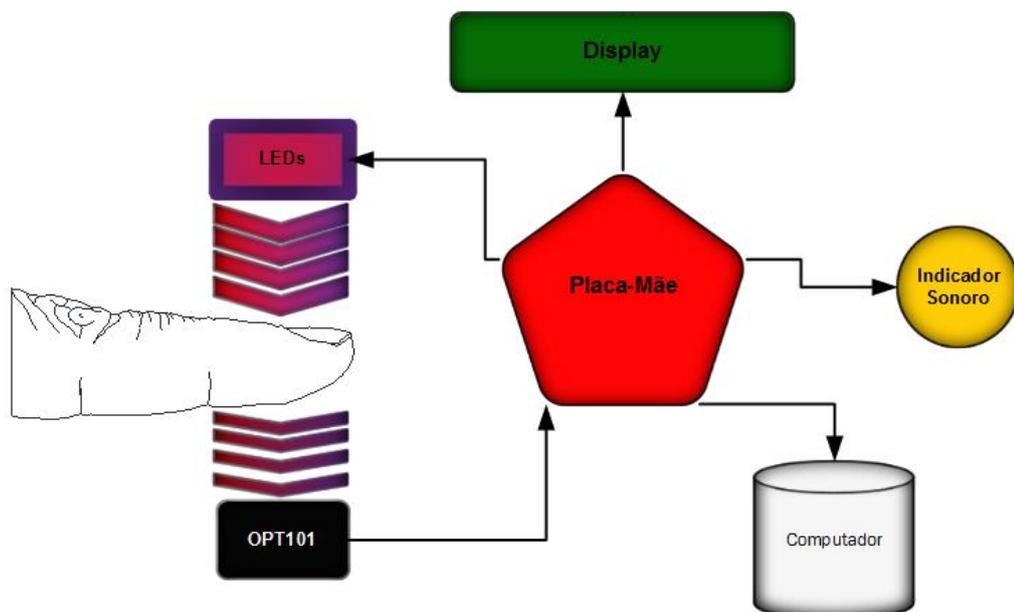
onde  $y(n)$  é o sinal de saída;  $b_0, b_1, \dots, b_{M-1}$  são os coeficientes do filtro;  $M$  é a ordem do filtro.

### 3 O PROJETO

Para o desenvolvimento do protótipo, fora necessário o levantamento dos segmentos de *hardware* que estruturam o dispositivo. Em decorrência de tal necessidade, elaborou-se uma estratégia de desenvolvimento, a qual consiste no desenvolvimento da estrutura de sensoriamento, composta pelos circuitos de acionamento dos LEDs e do OPT101, pelo circuito de alimentação de todos os componentes, pela interface de apresentação de dados ao usuário, constituída pelo visor gráfico e por um circuito sonoro.

Todas as estruturas apresentadas, estão interconectadas através da placa-mãe, a qual possui em seu núcleo o microcontrolador, que tem como função o processamento das estruturas de código desenvolvidas, com a finalidade de tornar a integração dos periféricos, um sistema operante em conjunto e capaz de realizar a aquisição, processamento e apresentação dos dados da curva pletismografica, batimentos cardíacos e percentual de oxigenação do sangue arterial periférico. É valido ressaltar o fato de que, o desenvolvimento das partes do projeto possui interdependências entre si, fazendo-se necessária a revisão sucessiva.

De modo a contextualizar e exemplificar as estruturas que compõe o protótipo, tem-se o diagrama apresentado na Figura 7, o qual ilustra de maneira simplificada as principais partes do sistema.



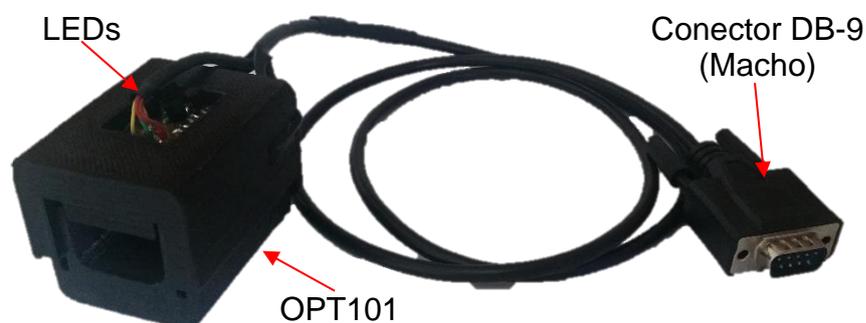
**Figura 7 - Diagrama da estrutura que compõe o protótipo do oxímetro**  
Fonte: Autoria própria.

Através da análise do diagrama apresentado na Figura 7, percebe-se que o sistema pode ser analisado através de cinco principais pontos de estudo, sendo: o sensor, composto pelos blocos “LEDs” e “OPT101”; a interface de exibição, composta pelo bloco “Display”; o sistema sonoro, representado por “Indicador Sonoro”; a central de processamento, no bloco “Placa-Mãe”; e, por fim, o sistema de comunicação com o computador, representado pelo bloco “Computador”, o qual tem por finalidade apenas complementar a análise e estudo dos dados a serem adquiridos durante a análise experimental.

### 3.1 SENSOR DE OXIMETRIA

O dispositivo responsável pela captação do sinal que propicia a determinação da taxa de saturação percentual de oxigênio no sangue é composto basicamente por dois blocos, o circuito de chaveamento dos LEDs (5 mm de alto brilho), responsável pela emissão dos feixes luminosos, e o circuito de captura da luminosidade não absorvida pelo tecido. Essas duas estruturas encontram-se descritas nos subcapítulos 3.1.1 e 3.1.2.

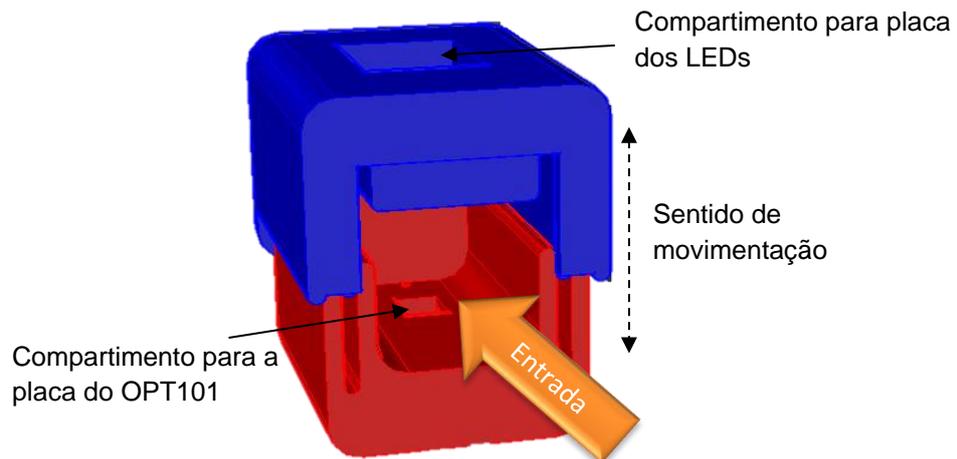
A estrutura completa da ponteira do sensor de oximetria do equipamento, desenvolvida especificamente para o trabalho, encontra-se ilustrada na Figura 8. A ponteira é constituída pela estrutura de suporte para a extremidade do dedo, com as placas dos LEDs e do OPT101, um cabo blindado e um conector “macho” do tipo DB-9.



**Figura 8 - Ponteira (sensor) do protótipo de oxímetro**  
**Fonte: Autoria própria.**

A estrutura de suporte ao dedo fora projetada em *software* de modelagem 3D, visando a impressão de objeto em plástico, através de uma impressora tridimensional, de modo que fosse possível comportar as placas dos emissores e do

fotodiodo. A estrutura da ponteira, além de fornecer suporte físico as placas, também tem como finalidade reduzir a entrada de luz externa, visando a minimização da interferência luminosa do ambiente. Uma ilustração do projeto encontra-se apresentada na Figura 9, bem como algumas informações explicativas.



**Figura 9 - Projeto 3D da estrutura da ponteira.**  
**Fonte: Autoria própria.**

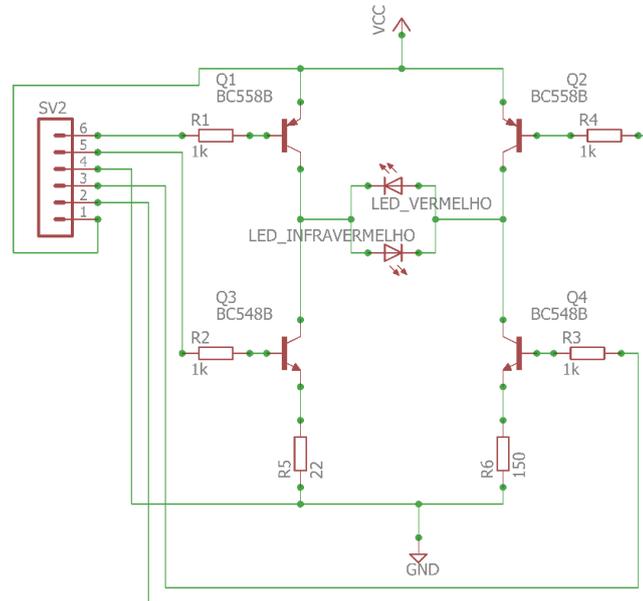
Para possibilitar a conexão desses sistemas periféricos com a placa principal, optou-se por um cabo blindado com a extremidade composta por um conector macho DB-9. Tal escolha tem como intuito a redução da interferência de ruído externo, através da malha de proteção, e atender a necessidade de 9 (nove) vias de conexão, sanada pelos 9 (nove) cabos que encontram-se no interior da malha de blindagem.

### 3.1.1 Circuito de acionamento dos LEDs

O sistema de acionamento dos LEDs, consiste em acende-los em momentos diferentes, seguidos de intervalos em que ambos estejam desligados. Para tal efeito, fez-se o uso de um grupo de transistores em formato de ponte H, onde se tem a polarização de um transistor NPN e um PNP, permitindo a polarização direta de um dos diodos emissores de luz, e polarizando inversamente o outro (KUSHIMA, 2014). Tal acionamento é realizado através da combinação de chaveamento de quatro pinos do microcontrolador, assim como descrito no código apresentado no APÊNDICE B.

A estrutura e a disposição dos componentes que compõe o sistema de acionamento dos LEDs está apresentada na Figura 10, a qual fora elaborada através

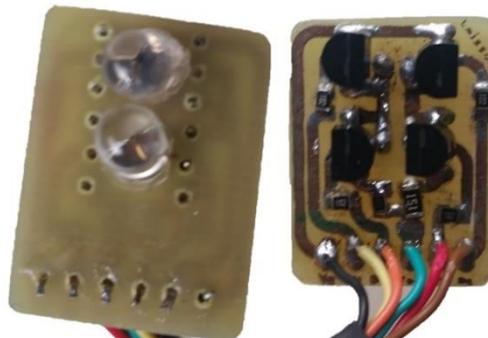
do *software* EAGLE®, na qual consta o esquemático do circuito eletrônico de chaveamento dos emissores, baseado no circuito desenvolvido por Kushima (2014).



**Figura 10 - Esquemático do sistema de acionamento dos LEDs**  
Fonte: Autoria própria.

É importante ressaltar que fora necessário realizar a alteração do valor do resistor R5, para  $22\ \Omega$ , em decorrência da constatação empírica da baixa intensidade luminosa captada pelo fotodiodo, tornando necessária a elevação da potência luminosa gerada pelo LED vermelho, bastando reduzir o valor do resistor, visando o aumento da corrente que passa pelo LED vermelho.

Através do esquemático, e utilizando o *software* de elaboração de *layout* de placas de circuito impresso (vide APÊNDICE A), fora produzida a placa física, com a finalidade de implementação do protótipo. Na sequência está apresentada a imagem da placa com os componentes, vide a Figura 11.



**Figura 11 - Placa de acionamento dos LEDs**  
Fonte: Autoria própria.

As terminações da placa do circuito de acionamento dos LEDs encontram-se conectadas aos pinos do microcontrolador, os quais estão configurados como portas de saída. O acionamento de cada um dos LEDs, bem como o desligamento de ambos, é realizado através de uma combinação lógica dos níveis de tensão dos quatro pinos do controlador, responsáveis pelo chaveamento. Para melhor ilustrar o funcionamento do circuito, têm-se a lógica apresentada na Tabela 1.

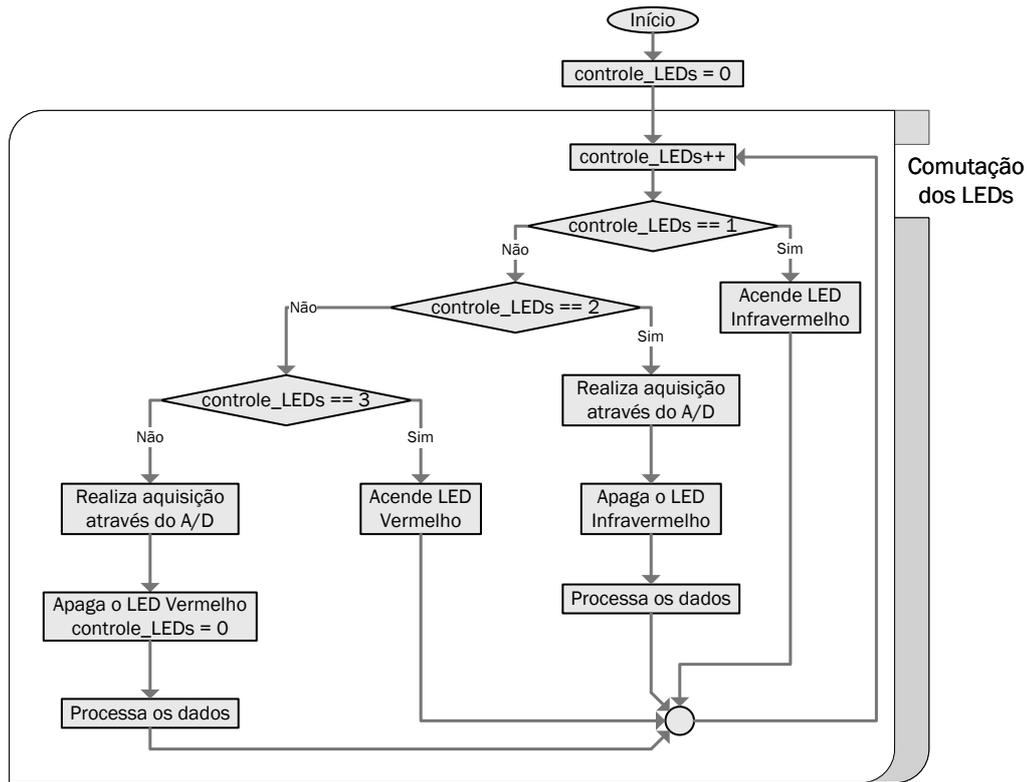
**Tabela 1 - Lógica de acionamento dos LEDs**

Terminal 6	Terminal 5	Terminal 3	Terminal 2	LED Vermelho	LED Infravermelho
1	1	0	0	Ligado	Desligado
0	0	1	1	Desligado	Ligado
1	0	0	1	Desligado	Desligado

Para realizar o chaveamento do transistor, é emitido um sinal lógico em nível alto, com amplitude de aproximadamente 3,6 V, o qual injeta uma corrente na base do transistor, fazendo com que o componente passe a conduzir corrente pelos outros dois terminais. Quando se aplica nível lógico baixo, próximo dos 0 V, o transistor não é polarizado, impedindo a passagem de corrente. Essa metodologia de acionamento é utilizada nos quatro componentes, os quais são acionados como demonstrado na Tabela 1, em cada período do ciclo de chaveamento.

A definição dos níveis lógicos dos pinos é realizada através de uma rotina responsável pelo acionamento dos LEDs, atrelada à amostragem do nível de tensão do sensor. Tal rotina de código está apresentada no APÊNDICE B.

Para uma melhor compreensão da forma com que opera o algoritmo de comutação dos LEDs e, paralelamente, em que momento são realizadas a aquisição do sinal através do A/D e o processamento dos dados, tem-se, portanto, o fluxograma apresentado na Figura 12.



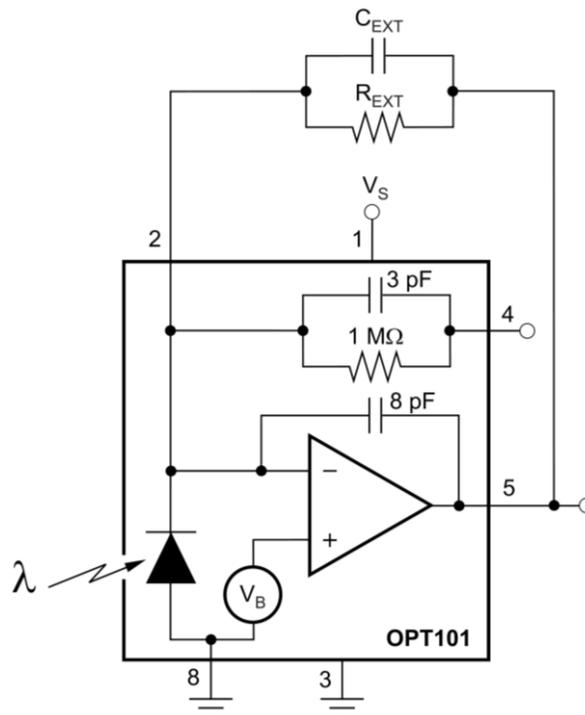
**Figura 12 - Fluxograma referente ao acionamento dos LEDs**  
**Fonte: Autoria própria.**

Através da análise da estrutura apresentada na Figura 12, pode-se perceber que, basicamente, o sistema consiste em acionar um dos LEDs (no primeiro ciclo), na sequência realizar a aquisição, desativação dos LEDs e processamento dos dados obtidos referente ao LED acionado (no segundo ciclo), ativar o outro LED (no terceiro ciclo), e na sequência realizar uma nova aquisição, desativação dos LEDs e processamento dos dados referentes ao segundo LED (no quarto e último ciclo), de modo que se conclua toda a rotina e que seja reiniciada, fazendo com que sejam realizados novamente todos os quatro ciclos, de modo que a amostragem ocorra à uma frequência de aproximadamente 256 Hz, o que garante uma amostragem maior do que duas vezes a frequência do sinal, a qual pode chegar a 4 Hz, em condições normais de um ser humano.

### 3.1.2 Sensor OPT101

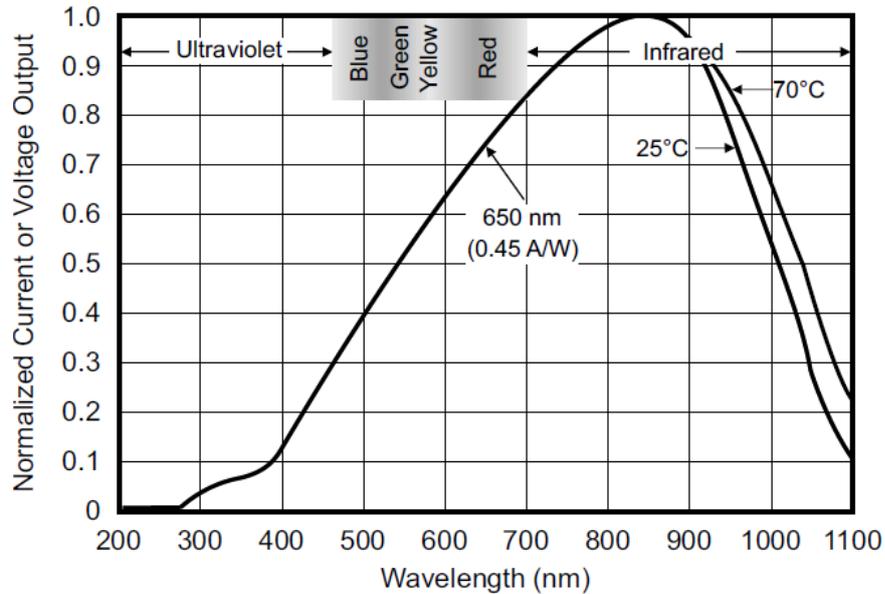
A empresa Texas Instruments Incorporated (2015) dispõe ao mercado um circuito integrado composto por um fotodiodo e um amplificador de transimpedância. Esse componente é indicado para diversas áreas de aplicação, tais como instrumentação biomédica, analisadores fotográficos, instrumentação laboratorial,

entre outras. O CI (Circuito Integrado) possui, dentre suas características, uma faixa de alimentação de 2,7 V até 36 V e também apresenta uma reposta elevada à incidência de um feixe no espectro vermelho (0,45 A/W para 650 nm) (TEXAS INSTRUMENTS INCORPORATED, 2015). O esquema do encapsulamento, bem como os componentes que estão integrados no circuito, estão apresentados na Figura 13.



**Figura 13 - Diagrama de bloco do OPT101**  
**Fonte: TEXAS INSTRUMENTS INCORPORATED (2015).**

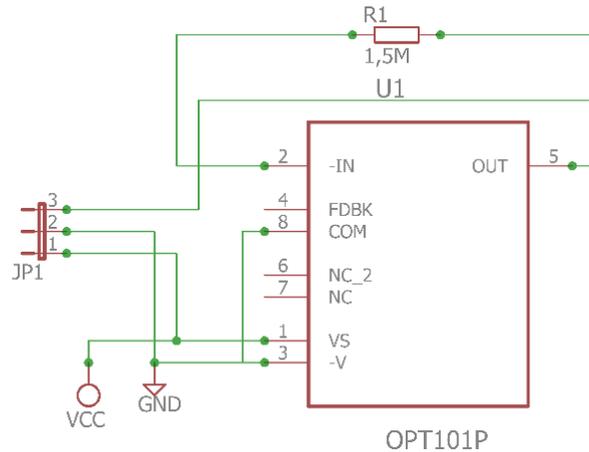
Como se tem a necessidade de trabalhar com os dois LEDs de espectro diferente, faz essencial a análise da abrangência da resposta do sensor à estimulação através dos dois espectros de onda. Para tal conferência, utilizou-se a curva de resposta ao espectro da Figura 14.



**Figura 14 - Resposta aos espectros de frequência da luz**  
**Fonte: TEXAS INSTRUMENTS INCORPORATED (2015).**

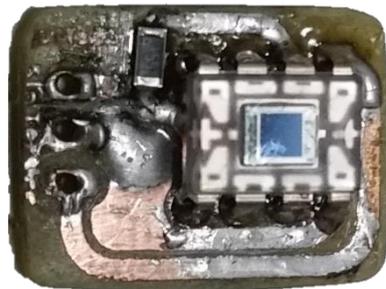
Como os espectros utilizados estão próximos das faixas de 660 nm (vermelho) e 895 nm (infravermelho), ao analisar a Figura 14, conclui-se que o CI proporciona um fator de resposta maior que 70%, o que possibilita a captação dos comprimentos de onda de ambos emissores.

A captura do sinal luminoso gerado pelos LEDs, e que não fora absorvida pelo tecido, é realizada pelo OPT101, o qual dispõe de pinos que permitem a conexão de um resistor que proporciona o ajuste do ganho do sinal de tensão gerado pelo fotodiodo. Através da análise das especificações técnicas do fabricante, constatou-se a necessidade de se conectar apenas o resistor, não se fazendo necessária a conexão de um capacitor, uma vez que a folha de dados dispensa o uso do capacitor para valores de resistor maior, ou igual, a 1 MΩ e frequência inferior, ou igual, a 23 kHz (TEXAS INSTRUMENTS INCORPORATED, 2015). A disposição esquemática do circuito do fotosensor está apresentada na Figura 15.



**Figura 15 - Esquemático do circuito do fotosensor**  
**Fonte: Autoria própria.**

Através do circuito apresentado, fora confeccionada a PCI do fotosensor, a qual possui seu layout apresentado no APÊNDICE A. Com isso, obteve-se o circuito físico ilustrado na Figura 16.



**Figura 16 - Placa do fotosensor**  
**Fonte: Autoria própria.**

O valor do resistor utilizado no OPT101 é ajustado experimentalmente, uma vez que existem um número demasiadamente grande de variáveis envolvidas com o sinal de saída do sensor, tais como distância do caminho percorrido pelos feixes de luz, densidade do tecido do corpo, pigmentação da pele, entre outras. Portanto, com relação a luminosidade incidente, necessita-se da constatação de qual ganho apresenta uma melhor operação para que se consiga um sinal com uma amplitude razoavelmente alta (próxima a tensão máxima de referência do A/D, definida em 2,5 V). Porém, o ganho não deve saturar a saída do amplificador operacional, contido no interior do encapsulamento.

O sistema de obtenção do sinal de oximetria é estruturado pelo fotosensor, o qual atua em paralelo com os emissores de luz, e por um pino do microcontrolador, o qual dispõe de um conversor A/D de 12 bits. Esse conversor irá transformar o sinal

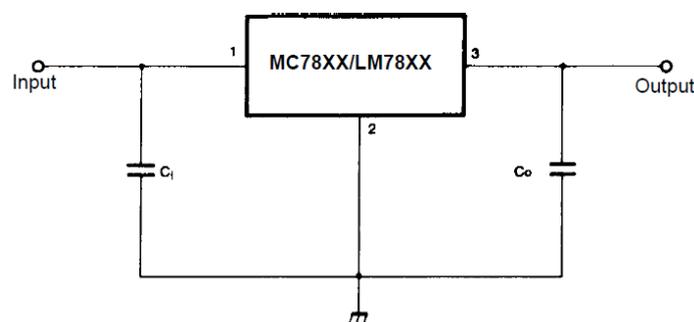
de tensão produzido pelo amplificador de transimpedância em informação digital, a qual será tratada e processada pelas rotinas contidas no controlador, com a finalidade de obter a curva de oximetria, os batimentos por minuto (BPM) e a taxa percentual de saturação de oxigênio arterial periférico.

Para a demultiplexação dos sinais provenientes dos LEDs vermelho e infravermelho, como se utiliza o mesmo sensor, é realizada a leitura do nível de tensão enviado pelo terminal 3, da Figura 15, de maneira intercalada, atribuindo o valor lógico a duas variáveis distintas, bastando realizar a atribuição da leitura à variável correspondente ao LED que encontrasse sensibilizando o fotodiodo, permitindo que os sinais sejam adquiridos separadamente. A rotina responsável pela lógica aqui descrita, está contida em um trecho do APÊNDICE B.

### 3.2 FONTE DE ALIMENTAÇÃO

O circuito digital do protótipo demanda de dois níveis distintos de tensão, de 3,6 V e 5 V, sendo responsáveis pela alimentação, respectivamente, do circuito de acionamento dos LEDs e microcontrolador, e do display e sensor OPT101.

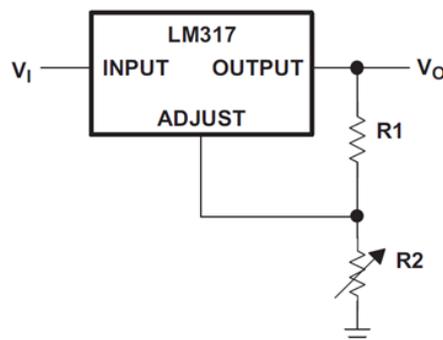
Para se conseguir obter os níveis de tensão demandados, utilizou-se um regulador de tensão fixa, de 5 V, denominado LM7805, o qual, para valores de entrada acima da tensão de saída de 5 V, garante uma operação consideravelmente estável do nível de tensão de saída. O circuito típico para esse tipo de regulador está apresentado na Figura 17 e tem-se a necessidade de acoplar capacitores na entrada e saída, para minimizar as flutuações de tensão e reduzir o nível de ruído.



**Figura 17 - Regulador de tensão de saída fixa**  
**Fonte: Adaptado de FAIRCHILD (2001).**

O circuito do sistema primário de regulação de tensão, composto em sua essência pelo LM7805, permite que a entrada de alimentação possa excursionar entre 7,5 V até 35 V, sem que haja prejuízos à operação do protótipo (FAIRCHILD, 2001).

Na confecção do segundo circuito de fornecimento de tensão, fora utilizado outro tipo de CI regulador de tensão, o LM317, o qual possibilita, através de um circuito típico, o ajuste da tensão de saída, bastando variar o valor da resistência existente entre seu pino de ajuste e a referência. A configuração do componente, para atuar como um regulador ajustável, tipicamente sugerida pelas fabricantes, está apresentada na Figura 18.



**Figura 18 - Regulador de tensão ajustável**  
**Fonte: Adaptado de TEXAS INSTRUMENTS INCORPORATED (2014).**

Para se conseguir variar o valor da resistência  $R_2$ , indicado na imagem que antecede, fora utilizado um trimpot. Ressaltando que fora necessário realizar o ajuste da resistência com o circuito regulador desconectado dos componentes que desejasse alimentar, visando proteger tais componentes de possíveis valores de tensão que não sejam admitidos para sua operação, podendo danificá-los.

Segundo a folha de dados da fabricante National Semiconductor (2004), o cálculo dos componentes, para utilização do LM317 na configuração regulador ajustável, segue a relação determinada por:

$$V_{OUT} = 1,25 \times \left( 1 + \frac{R_2}{R_1} \right) + I_{ADJ} \times R_2 \quad (8)$$

onde  $V_{OUT}$  é a tensão de saída do regulador, dada em V (Volts);  $R_2$  é o valor da resistência ajustável, em  $\Omega$  (Ohm);  $R_1$  é o resistor associado com  $R_2$ , dado em  $\Omega$

(Ohm); e  $I_{ADJ}$  é a corrente intrínseca de ajuste do componente, em A (Ampere), tipicamente entorno de  $50 \mu A$ .

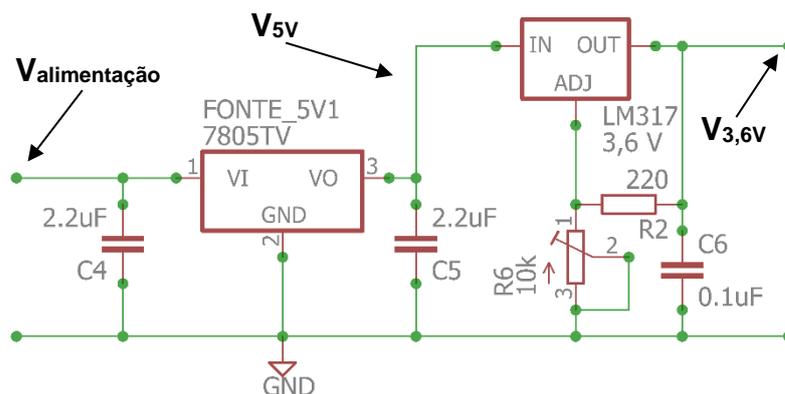
Reajustando a (8), visando isolar o termo  $R_2$ , e assumindo  $V_{OUT} = 3,6 V$ ,  $R_1 = 220 \Omega$  e  $I_{ADJ} = 50 \mu A$ . Logo, obtém-se:

$$R_2 = \frac{3,6 - 1,25}{\frac{1,25}{220} + 50 \times 10^{-6}} \cong 409,99 [\Omega] \quad (9)$$

Portanto, observado o valor de  $R_2$  definido em (9), utiliza-se um potenciômetro para ajustar tal valor de resistência, ressaltando que ele deverá ser maior que  $R_2$ . Logo, optou-se por um potenciômetro de  $10 k\Omega$ .

No circuito regulador de tensão ajustável também fora inserido capacitores, semelhantes aos utilizados no circuito do LM7805, com a mesma finalidade do regulador anteriormente apresentado.

Portanto, para a composição do sistema de alimentação, faz-se o uso dos dois componentes de regulação de tensão, de modo que estejam associados em série, onde a fonte alimenta o LM7805 (disponibilizando uma tensão de  $5 V$ ), o qual, na sequência, alimenta o LM317 (disponibilizando uma tensão ajustada em  $3,6 V$ ). O diagrama esquemático do sistema de alimentação está apresentado na Figura 19.



**Figura 19 - Circuito de alimentação**  
Fonte: Autoria própria.

Após a confecção da placa, necessita-se do ajuste do potenciômetro, uma vez que o valor calculado leva em consideração parâmetros ideais do componente. Portanto, deve-se utilizar um multímetro para acompanhar o nível de tensão na saída

do circuito de regulação da tensão de 3,6 V, de modo que se excursiona o potenciômetro até atingir o valor desejado.

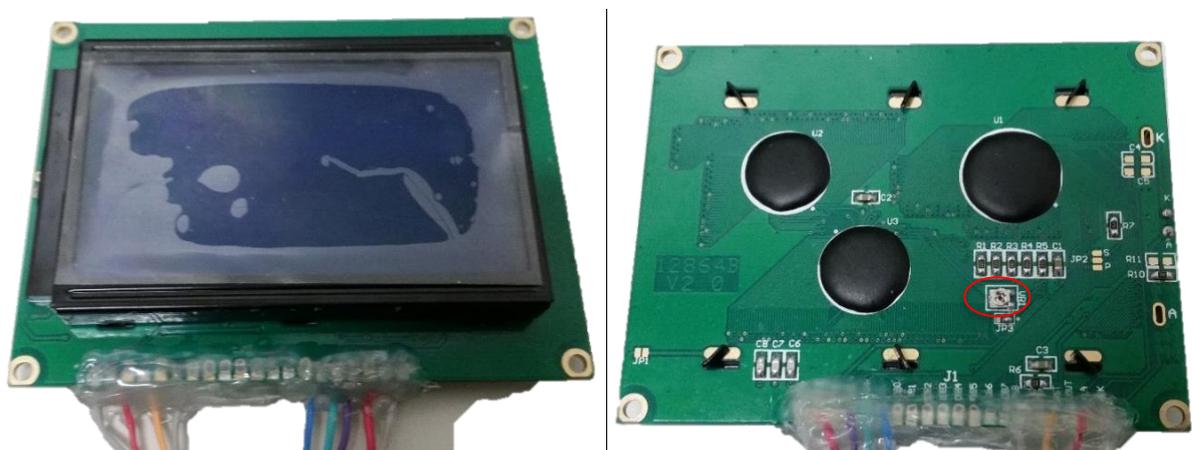
Ambos os circuitos de alimentação compõem parte da placa-mãe, a qual encontra-se apresentada no Subcapítulo 3.6.

### 3.3 DISPLAY 128X64

Com o intuito de proporcionar ao usuário uma interface de monitoramento da operação do protótipo, buscou-se utilizar um visor de cristal líquido para exibir a curva pletismográfica e os valores de batimentos cardíacos e saturação percentual de oxigenação. Definiu-se o display de modo que suas dimensões físicas e a resolução de pixels fossem suficientes para proporcionar fácil visualização das informações pelo usuário. Isto posto, definiu-se o display de resolução 128x64, o qual pode operar no modo SPI (Interface Serial Periférica, do original *Serial Peripheral Interface*), visando a redução no uso das portas do microcontrolador.

O display adquirido possui um controlador ST7920, o qual é responsável pelo processamento das instruções e dados enviados via serial pelo microcontrolador. Esse dispositivo possibilita a escrita de dados na memória DDRAM, endereçando em 64 blocos de 16-bit cada, os quais são correspondentes a matriz de pixels da tela (SITRONIX, 2008).

A imagem do Display de Cristal Líquido (LCD, do original *Liquid Crystal Display*) gráfico 128x64 pixels, para finalidade de identificação, está exposta na Figura 20.



**Figura 20 - Display gráfico de LCD 128x64 pixels**  
Fonte: Autoria própria.

O display também possui, como característica de operação, uma faixa de alimentação que corresponde entre os valores de 2,7 V até 5,5 V e um potenciômetro integrado a placa, identificado na Figura 20 por um círculo em vermelho, o qual permite o ajuste do contraste do visor (SITRONIX, 2008).

Para a operação do display, como não utilizou-se bibliotecas prontas, uma vez que o enfoque é também ampliar a capacidade de desenvolvimento do acadêmico, necessitou-se da elaboração de uma rotina de inicialização do equipamento, como descrito na folha de dados fornecida pelo fabricante. Tal rotina tem por finalidade inicializar o visor e também o configurar, de maneira condizente com a aplicação e modo de operação que o desenvolvedor opte por utilizar. A rotina de inicialização está apresentada no APÊNDICE E.

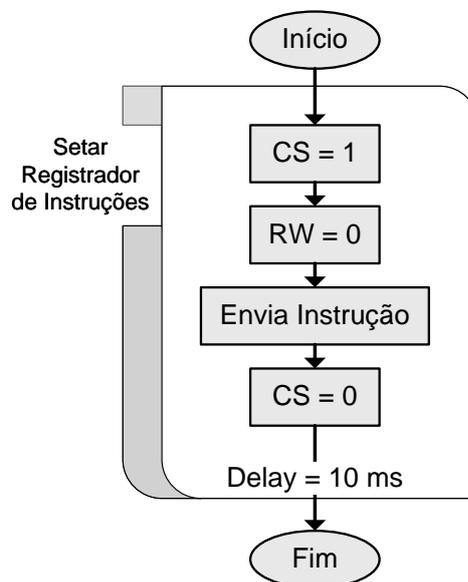
Visando reduzir o número de portas utilizado por esse periférico, optou-se por uma comunicação SPI, a qual demanda de poucos pinos, sendo necessários apenas quatro para a operação do display. Com base nesse cenário, o display irá atuar como um dispositivo “escravo”, o qual apenas irá receber dados e comando do microcontrolador, o qual assume a posição de “mestre”. A conexão do display com o restante do circuito encontra-se apresentada na Figura 32, a qual encontra-se exposta no tópico “Placa-mãe”.

Para a construção das imagens a serem exibidas no visor, foram elaboradas bibliotecas e rotinas de escrita. Logo, fez-se necessária a elaboração de uma biblioteca constituída de uma imagem gráfica de inicialização do display e de operação, contida em um vetor hexadecimal, e de rotinas de escrita em tela, cada uma com finalidade e característica específica. Essas rotinas estão apresentadas e comentadas no APÊNDICE C.

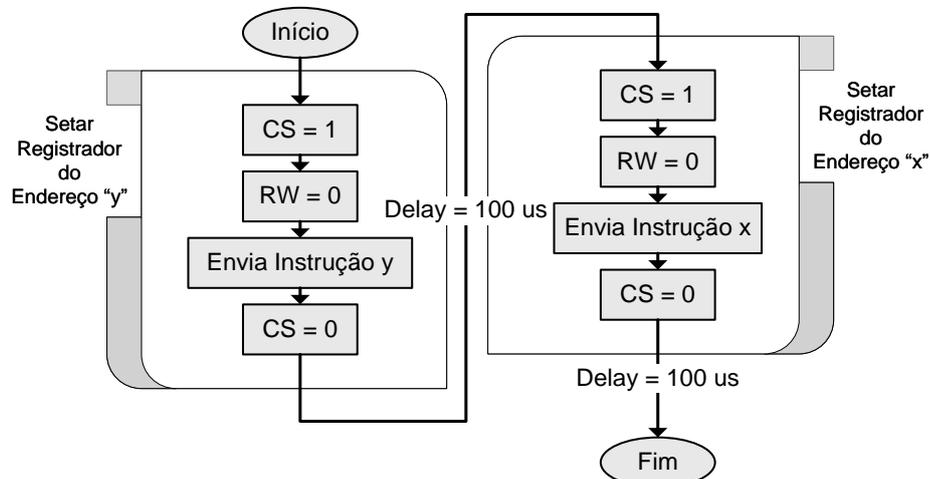
Fora necessário, também, o desenvolvimento de uma biblioteca específica para conter os dígitos utilizados na apresentação dos valores dos batimentos cardíacos e taxa percentual de oxigenação, tal biblioteca encontra-se contida no APÊNDICE G.

A escolha da elaboração das bibliotecas também fora tomada com base no uso consciente de memória do microcontrolador, uma vez que a utilização de uma biblioteca convencional, disponibilizada por grupos de usuários, apresenta uma diversidade de número de caracteres maior do que o necessário para a aplicação, tornando tais bibliotecas “maiores” do que o necessário.

Através da análise da folha de dados, disponibilizada pela fabricante Sitronix (2008), pode-se identificar os procedimentos e códigos necessários para que se possa estabelecer a comunicação entre o microcontrolador e o display, e como realizar a escrita de dados na tela. Tal análise identificou os procedimentos necessários para realizar 3 (três) tipos de instruções, essenciais para o envio de dados ao display na forma serial, as quais são: setar o registrador de instruções, descrito na Figura 21; setar os registradores de endereço, descrito na Figura 22, e envio de dados, apresentado na Figura 25.

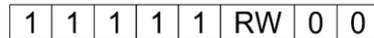


**Figura 21 - Fluxograma do processo para setar o registrador de instruções**  
**Fonte: Autoria própria.**



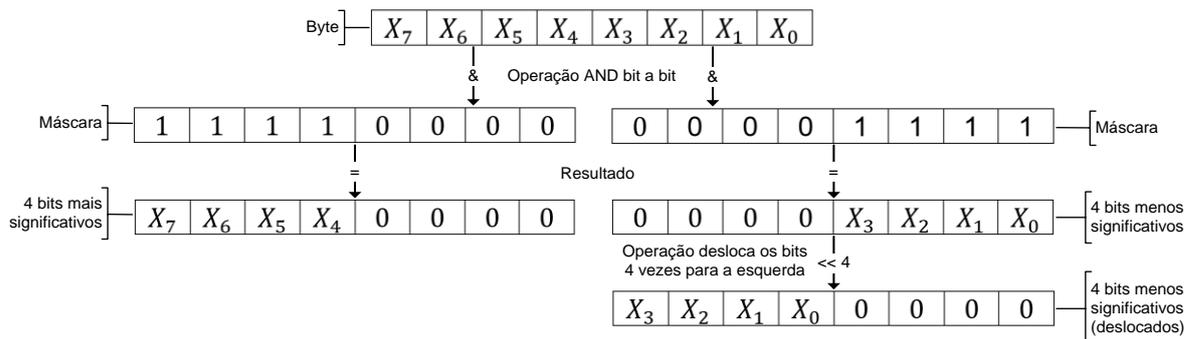
**Figura 22 - Fluxograma do processo para setar o registrador de endereços**  
**Fonte: Autoria própria.**

Como o sistema de comunicação fora definido como sendo serial, deve-se ter em mente o fato de que os dados dos blocos “RW” e “Envia Instrução” devem ser enviados em blocos de 8 (oito) bits sequências. É necessário ressaltar que, por padrão da fabricante, o bit RW deve ser enviado em um byte com a seguinte estrutura apresentada na Figura 23.



**Figura 23 - Estrutura do byte de envio de "RW"**  
**Fonte: Autoria própria.**

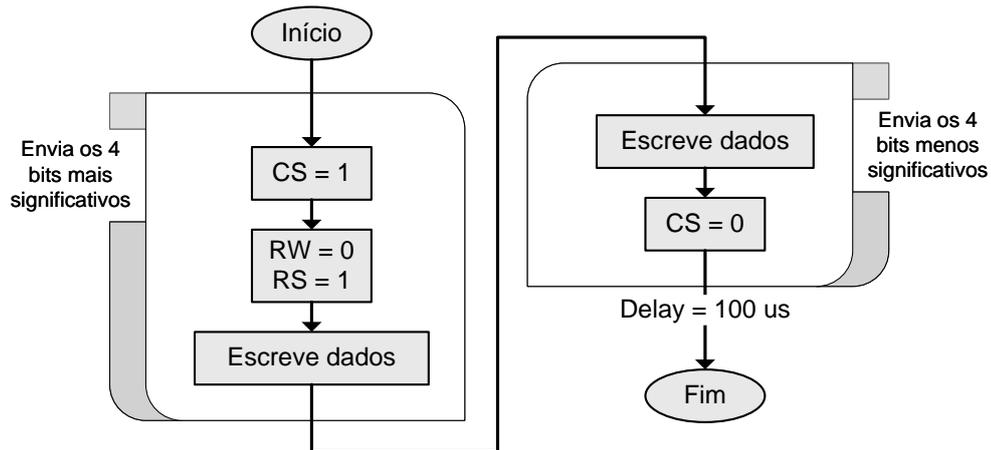
Existe também uma peculiaridade no envio do byte de instrução, onde deve-se enviá-lo separadamente em dois grupos de 4 (quatro) bits de informação, ordenado primeiro pelo envio da parte dos 4 (quatro) bits mais significativos e, na sequência, pelos demais. A operação para envio e separação em dos blocos de bits encontra-se melhor descrita na Figura 24.



**Figura 24 - Diagrama da operação de separação dos 4 bits mais significativos e dos 4 bits menos significativos (deslocados)**  
**Fonte: Autoria própria.**

Tendo executada a separação dos 2 (dois) blocos de 4 (quatro) bits, seguindo as indicações contidas na Figura 24, basta enviá-los em sequência ao display, para concluir o processo de envio de instrução.

Para o envio de dados que se deseja escrever no display, deve-se executar os procedimentos descritos no fluxograma apresentado na Figura 25, onde encontram-se descritas as sequências dos procedimentos e configurações necessárias.



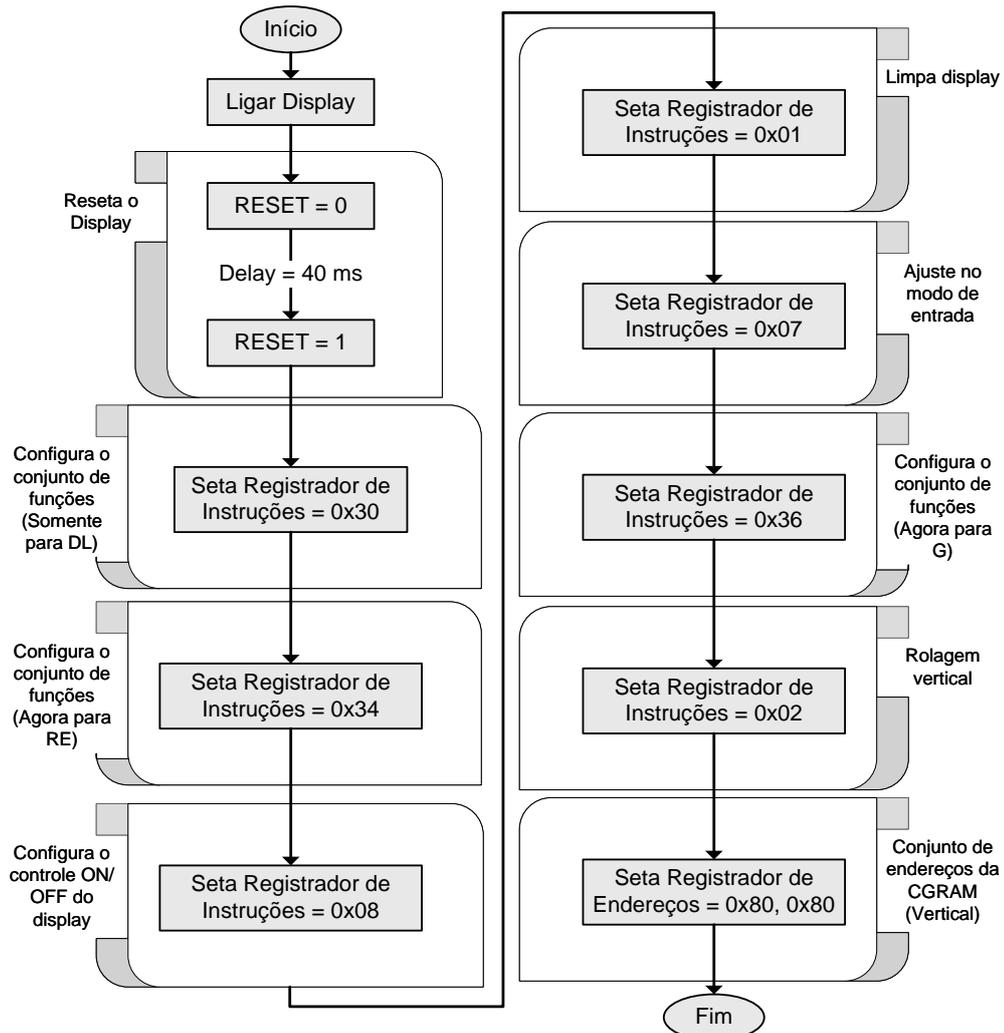
**Figura 25 - Fluxograma para escrever dados no display**  
**Fonte: Autoria própria.**

Nos procedimentos descritos na Figura 25, percebe-se que é necessária a separação do byte, a ser escrito, em 2 (dois) blocos de 4 (quatro) bits, seguindo os procedimentos indicados na Figura 24, apresentada anteriormente. Com relação ao envio dos valores de “RW” (Ler/Escrever, do original *Read/Write*) e “RS” (Registrador de Seleção, do original *Register Select*), deve-se seguir a estrutura de construção do byte segundo apresentado na Figura 26.



**Figura 26 - Estrutura do byte de envio de "RW" e "RS"**  
**Fonte: Autoria própria.**

Tendo apresentada as principais operações necessárias para realizar o controle da operação do display, deve-se então, sendo especificações definidas pela Sitronix (2008), realizar algumas rotinas de instruções e comandos, para que se inicialize o display e possa, enfim, escrever informações na tela, tais procedimentos encontram-se explicitados em um fluxograma apresentado na Figura 27.



**Figura 27 - Fluxograma da inicialização do display 128x64**  
 Fonte: Autoria própria.

Concluído o processo de inicialização do display, o dispositivo torna-se disponível para receber as informações a serem escritas na tela, sendo necessário fornecer, primeiramente, os dados da posição de memória correspondente ao bloco dos 16 (dezesesseis) pixels, e, na sequência, os dados a serem escritos na tela, ressaltando que o valor binário “0” equivale ao pixel apagado e “1” refere-se ao pixel acesso.

Para que torne-se possível a identificação de quais posições de memória corresponde aos pixels da tela, fora necessário a realização do mapeamento dos pixels da tela, o qual encontra-se ilustrado na Figura 28, com indicadores explicativos dos trechos da estrutura.

Para uma melhor experiência de interface homem-máquina, criou-se estruturas gráficas para tela de inicialização, de operação e de números (e indicadores dos modos especiais de operação), através da elaboração de bibliotecas

correspondentes, as quais encontram-se apresentadas no APÊNDICE C e no APÊNDICE G.

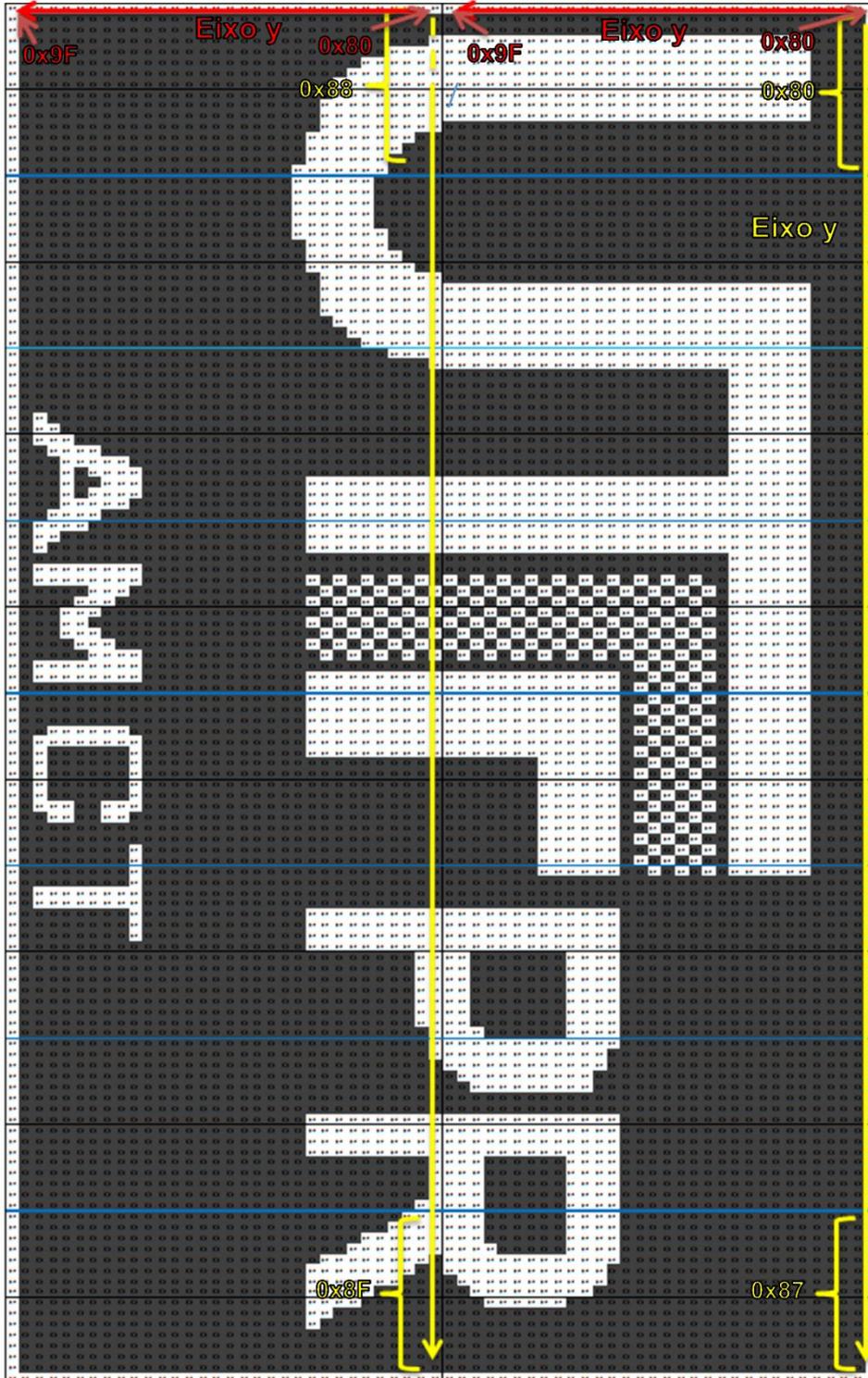


Figura 28 - Mapeamento do display 128x64  
Fonte: Autoria própria.

### 3.4 INDICADOR SONORO

O sistema sonoro é constituído basicamente por uma pastilha piezoelétrica, o qual passa a produzir um sinal sonoro quando excitado por uma forma de onda oscilatória, operando dentro de determinada faixa de frequência, arbitrariamente emitindo som audível quando sensibilizado por um sinal oscilatório próximo de 2 kHz.

Para tal aplicação, fora utilizada uma pastilha que possui um diâmetro de 20 milímetros, a qual está apresentada na Figura 29.



**Figura 29 - Pastilha piezo**  
**Fonte: Autoria própria.**

Para o acionamento do indicador sonoro, faz-se o uso de um temporizador, com finalidade exclusiva para tal função, disparando a geração de um sinal modulado por largura de pulso (PWM, do original *Pulse Width Modulation*) de aproximadamente 2 kHz, por um curto período, de modo que coincida com o pico da forma de onda proveniente do sinal de oximetria, operando assim como indicador audível dos batimentos cardíacos.

Visando a distinção da variação da intensidade sonora do equipamento, com base na alteração do nível de saturação de oxigênio do paciente, utiliza-se a equação (10), a qual varia em função do nível percentual de saturação de oxigênio.

$$N_{passos} = \frac{101 - SpO2\%}{2} \quad (10)$$

onde  $N_{passos}$  corresponde à quantidade de períodos de amostragem do sinal do A/D que o PWM ficará ativo acionando o piezo;  $SpO2\%$  refere-se a taxa percentual de saturação de oxigênio.

A equação (10) tem por finalidade determinar o período que o trem de pulsos ficará ativo, com base na quantidade de novas amostragens que o A/D realiza, lembrando que cada nova amostragem ocorre a uma frequência de aproximadamente

256 Hz. Desse modo, quanto menor a taxa de saturação de oxigenação, o PWM ficara ativos durante mais períodos de amostragem do A/D, e, de maneira complementar, quanto mais a taxa se aproxima dos 100%, menor será a duração do sinal PWM.

### 3.5 MICROCONTROLADOR MSP430FR5969

Para o acionamento dos periféricos de sensoriamento, faz-se necessário quatro pinos para o chaveamento dos diodos emissores de luz, atrelados a aquisição através de um conversor A/D. O procedimento de sensoriamento irá demandar de uma sincronização entre a comutação do acionamento e a aquisição de uma amostra, logo tem-se a necessidade de um temporizador, para determinar os intervalos entre as ações.

Para a comunicação com o display, visando a utilização de um número reduzido de portas, é passível de se utilizar o modo de comunicação serial, através da comunicação SPI, logo o controlador e o display devem dispor de tal capacidade de operação.

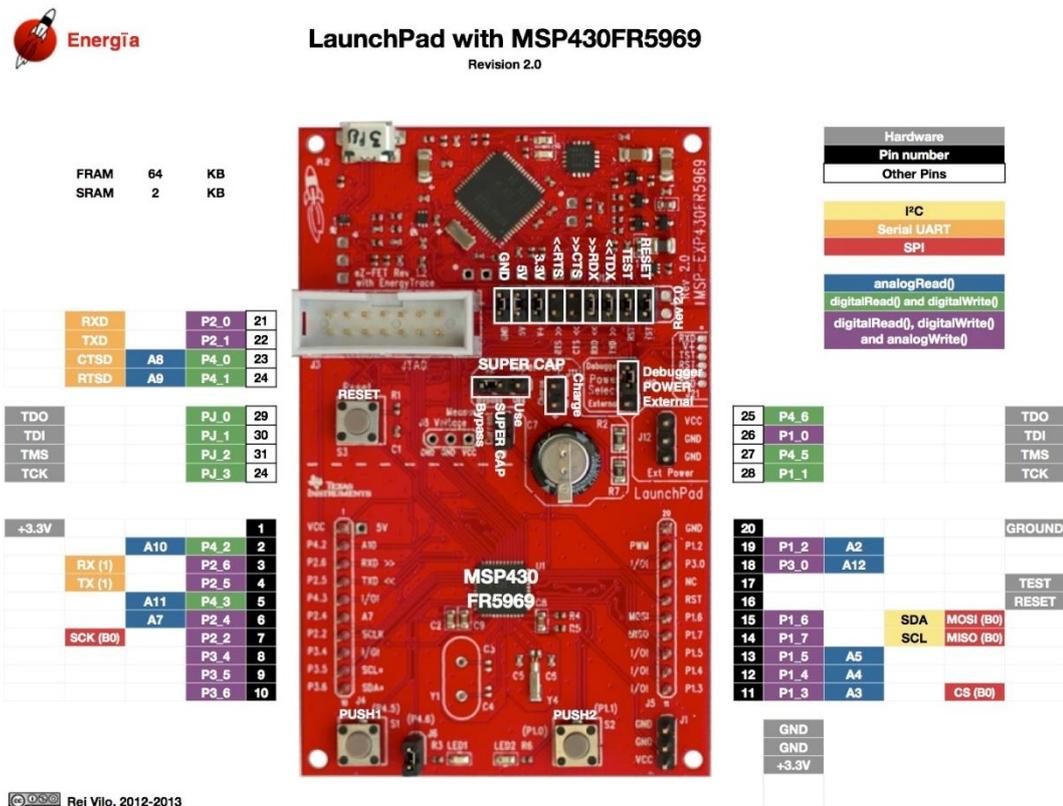
A construção do projeto fora estruturada na utilização do microcontrolador da família MSP430FR59xx, produzido pela Texas Instruments Incorporated, uma vez que o dispositivo apresenta características compatíveis com as demandadas pelo projeto, tais como conversor A/D, pinos de entrada e saída, comunicação SPI e temporizadores. A fabricante produz o *Kit LaunchPad*, para programação, e disponibiliza um ambiente de programação e depuração gratuito, para programação em linguagem C.

O microcontrolador escolhido é denominado MSP430FR5969, que dispõe das seguintes características, aqui julgadas como mais relevantes para a aplicação, sendo elas (TEXAS INSTRUMENTS INCORPORATED, 2015):

- Arquitetura RISC 16-bit;
- 4 (quatro) portas com 8 (oito) pinos cada, totalizando 32 (trinta e dois) pinos de entrada/saída;
- Tensão de operação de 1,8V até 3,6V;
- Frequência de operação da CPU de até 16MHz;
- Memória FRAM de 64 kB e SRAM de 2 kB;
- Frequência de operação da memória FRAM de até 8 MHz;

- 5 (cinco) temporizadores de 16-bit;
- 2 (dois) canais de comunicação Receptor/Transmissor Universal Assíncrona (UART, do original *Universal Asynchronous Receiver/Transmitter*);
- 3 (três) interfaces de comunicação serial SPI, UART e I2C;
- Conversor A/D 12-bit com dezesseis canais.

A ferramenta de desenvolvimento da fabricante fornece um hardware com conexão USB ao computador, para que se possa realizar os procedimentos de gravação dos códigos elaborados, bem como, utilizar do ambiente de depuração, o qual facilita o acompanhamento da execução da rotina e, caso necessário, auxilia na identificação de falhas e/ou inconsistências nos trechos de código. A disposição dos periféricos presentes no *LaunchPad* e sua pinagem, juntamente com as funções/características presentes em cada um dos pinos conectados ao microcontrolador, está apresentada na Figura 30.

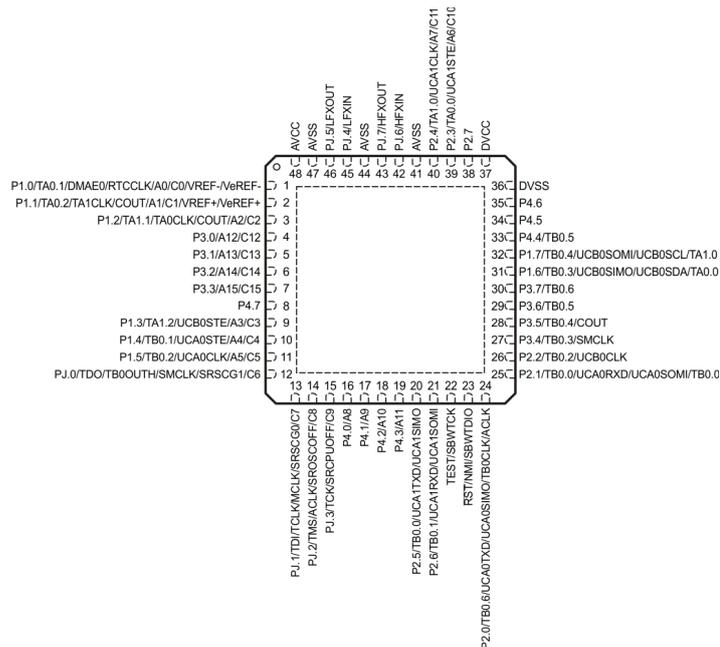


**Figura 30 - LaunchPad MSP430FR5969**  
Fonte: Energia (2016).

A ferramenta de desenvolvimento da Texas Instruments Incorporated coloca à disposição do desenvolvedor uma interface com terminais de conexão à alguns pinos do microcontrolador e também disponibiliza uma conexão USB com os pinos de comunicação UART do dispositivo, permitindo ao usuário explorar tal interface de comunicação, tornando possível a troca de dados entre o microcontrolador e o computador, por exemplo.

A central responsável pelo processamento das informações obtidas pelo sensor, bem como o acionamento dos periféricos, é composta pelo microcontrolador MSP430FR5969. O qual possui, em sua essência, rotinas de código, os quais, associado aos periféricos, possibilitam o desenvolvimento e execução de tarefas. Entretanto, também são necessárias algumas outras estruturas para permitir seu funcionamento, como a alimentação de 3,6 V, dos meios de conexão com os periféricos e também de um pino de *reset*, o qual é responsável por reiniciar o microcontrolador, sem a necessidade de retirar a alimentação do CI.

Salienta-se a necessidade de se conectar um resistor de *pull-up*, para que o componente não sofra inúmeras reinicialização, devido a flutuação no valor de tensão do pino de *reset*, indicado na Figura 31 como o pino 23 do controlador. Também fora necessária a utilização de uma chave tátil, para se consiga alterar o nível de tensão do pino de *reset* para zero, visando a reinicialização do controlador.



**Figura 31 - Diagrama de Pinos do MSP430FR5969 48-pinos RGZ**  
**Fonte: TEXAS INSTRUMENTS INCORPORATED (2015).**

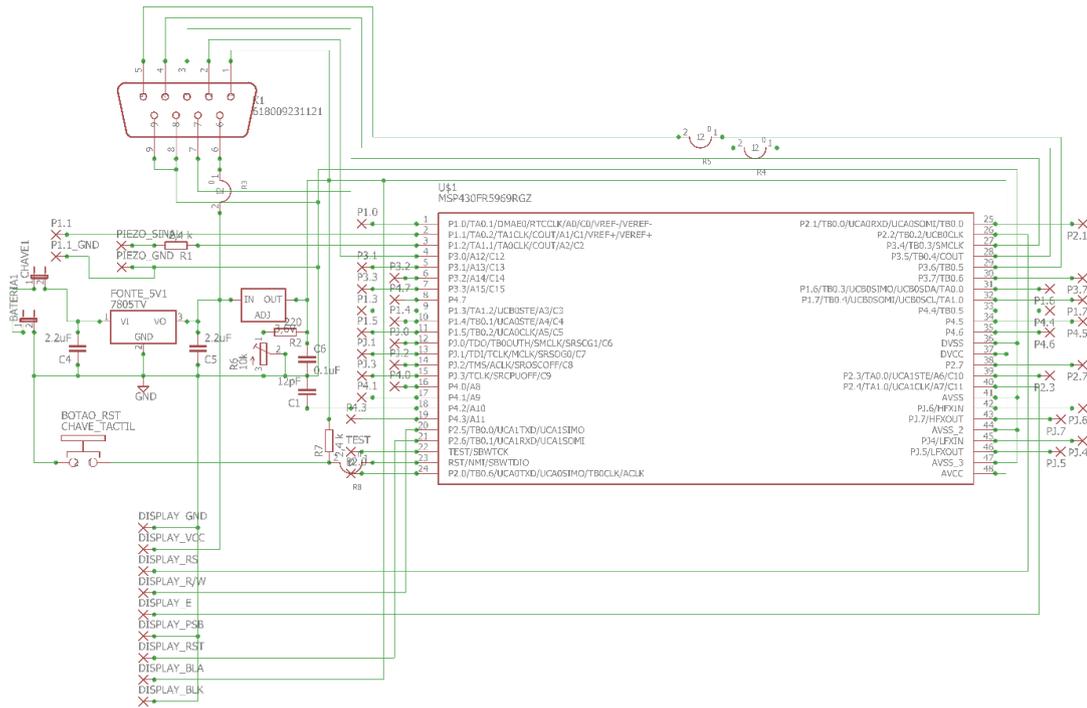
Experimentalmente, percebeu-se a necessidade de se inserir um tempo de espera no início do código principal, após a inicialização do protótipo, o que não fora observado durante as análises de operação do microcontrolador no kit. Tal medida foi necessária em virtude do fato de que o display não inicializa corretamente assim que se ligava o equipamento, passando a operar corretamente somente após uma reinicialização através do botão de *reset*. Essa observação induziu a atribuir a causa ao sistema de alimentação do protótipo, mais especificamente ao período transitório gerado pelo fechamento do contato entre a bateria e o restante do circuito. Essa especificação e outras, estão apresentadas no APÊNDICE B.

Dentre as configurações realizadas no controlador, ressalta-se o fato de que a CPU foi configurada para operar à 16 MHz e a frequência de operação da memória FRAM foi fixada em 8 MHz, visando agilizar o processamento dos dados.

### 3.6 PLACA-MÃE

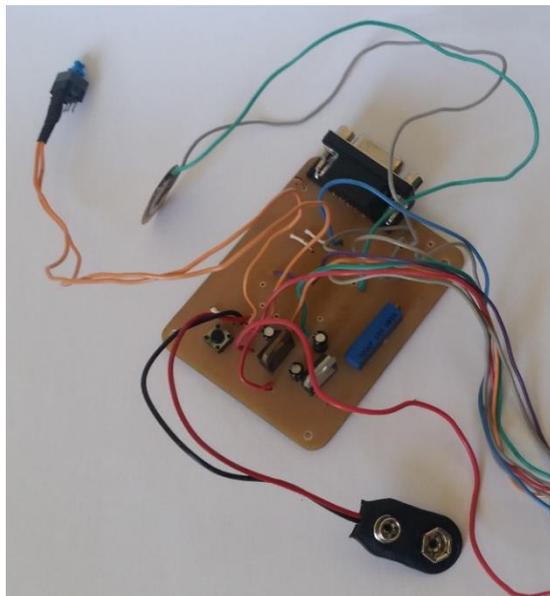
Para integração dos periféricos com a central de processamento, têm-se como vias de interligação o circuito da placa-mãe, onde estão presentes o terminal de conexão com o sistema de sensoriamento, constituído pelo sistema de acionamento dos LEDs e pelo fotosensor, o canal de conexão com o visor, as fontes de alimentação de 5 V e 3,6 V, e o canal de saída do sinal de acionamento do sistema sonoro.

O esquemático do circuito da placa-mãe está apresentado na Figura 32, disposta na sequência, indicando os componentes que compõe a placa e as terminações de conexão, devidamente indicadas.



**Figura 32 - Esquemático da Placa-Mãe**  
 Fonte: Autoria própria.

Com base no esquemático e no layout denominado “PLACA MÃE” (apresentado no APÊNDICE A), o qual fora elaborado por meio do programa EAGLE®, realizou-se a confecção da placa de circuito impresso, a qual está apresentada na Figura 33, com a finalidade de experimentação do protótipo operando de maneira independente, sem o auxílio do kit, assim como um produto produzido comercialmente.



**Figura 33 - Placa-Mãe**  
 Fonte: Autoria própria.

### 3.7 PROCESSAMENTO DIGITAL DOS SINAIS

O sinal analógico gerado pelo sensor OPT101, correspondente ao índice de absorção dos comprimentos de onda emitidos pelos LEDs vermelho e infravermelho, é convertido em informação digital através do conversor A/D, tornando possível o tratamento digital desses dados.

Imediatamente após a aquisição, como o sinal adquirido possui interferência ruidosa (tendo como exemplo o ruído irradiado por lâmpadas), faz-se necessária a filtragem do sinal pletoisomografico. O método de filtragem do sinal encontra-se descrito no tópico 3.7.1.

Após o tratamento do sinal digital, deve-se realizar o processamento dos dados, visando detectar a amplitude dos picos e vales, bem como a frequência com que são gerados os picos. A abordagem melhor detalhada encontra-se apresentada no tópico 3.7.2.

#### 3.7.1 Filtro de média móvel

Em um sistema que necessite da aquisição de sinais analógicos, através de conversores A/D, para que possam ser processados digitalmente, tem-se como estrutura típica uma determinada sequência de procedimentos a serem realizados, os quais estão apresentados no diagrama de blocos da Figura 34.



**Figura 34 - Diagrama de blocos de um sistema de aquisição de um sinal analógico para realização do processamento das informações**  
Fonte: Autoria própria.

Nesse tipo de sistema, torna-se necessária a estruturação da etapa de tratamento do sinal digital, gerado após a conversão das informações analógicas em dados digitais. Essa etapa tem como objetivo a eliminação de dados referentes a interferências ruidosas, de modo que se consiga realizar a filtragem do sinal almejado.

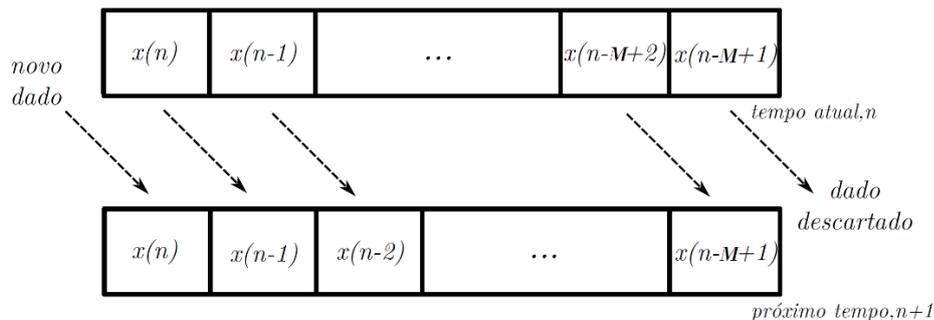
O fato de se ter um (ou mais) sinais indesejados contidos na informação que se busca processar, torna necessária, idealmente, a retirar de todas as

informações que não pertencem ao sinal que deseja-se processar. Como solução, tem-se a possibilidade de realizar a filtragem do sinal digital, visando atenuar o ruído.

Visando a atenuação do ruído em frequências acima dos batimentos cardíacos, os quais são inferiores a 5 Hz, e priorizando a redução da possível interferência (irradiada) causada por lâmpadas conectadas a rede de energia elétrica, de frequência fundamental 60 Hz, optou-se por um filtro de média móvel do tipo *notch* que atenua principalmente a frequência de 60 Hz e seus múltiplos inteiros. Nesse caso, tem-se que os coeficientes do filtro possuem o mesmo valor, podendo representar a equação (7) na forma da equação (11) (SANTINI, 2010).

$$x(n) = \sum_{k=(n-M+1)}^n \frac{x(k)}{M} \quad (11)$$

A equação (11) determina que  $x(n)$  será o sinal de saída, em um primeiro momento, porém em uma segunda amostragem, o sinal de saída da operação anterior passa a compor parte do cálculo da saída seguinte, como pode ser observado na Figura 35.



**Figura 35 – Amostragem do Filtro Média Móvel**  
**Fonte: Adaptado de Santini (2010).**

Para o cálculo dos coeficientes do filtro *notch*, utiliza-se a expressão de projeto que define a primeira frequência a ser atenuada, a qual pode-se calcular por meio de:

$$f_c = \frac{f_a}{M} \quad (12)$$

onde  $f_c$  é a frequência de corte;  $f_a$  é a frequência de amostragem do sinal que deseja-se filtrar; e  $M$  é a ordem do filtro e o termo do coeficiente do filtro.

Para o determinar o valor de  $M$ , optou-se por uma frequência de corte acima da frequência do sinal e próxima de um múltiplo inteiro de 60 Hz, e de modo que seja um múltiplo de 2 (dois), visando otimizar o processamento dos dados pelo microcontrolador para que se possa utilizar o deslocamento de bits para realizar as divisões. Assim, tem-se a expressão:

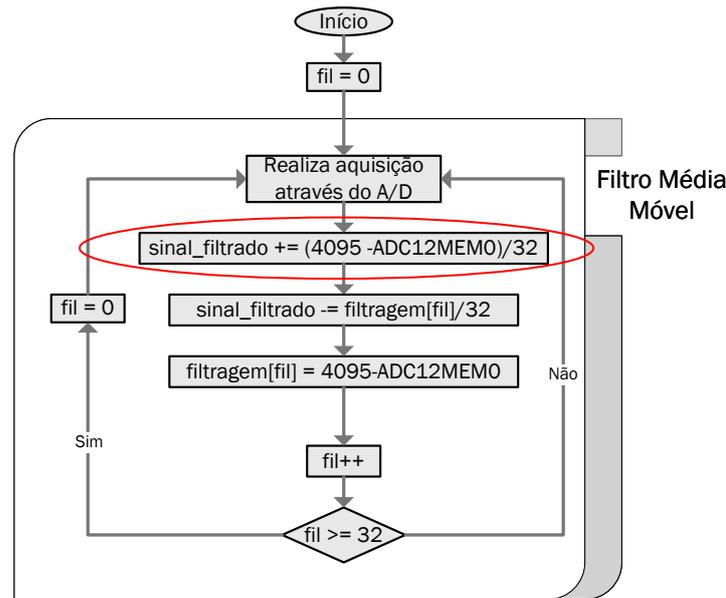
$$M = \frac{f_a}{f_c} = \frac{256}{8} = 32 \quad (13)$$

Portanto, tendo determinado o valor de  $M$ , obtém-se a expressão:

$$y(n) = \frac{1}{32} * x(n) + \frac{1}{32} * x(n - 1) + \dots + \frac{1}{32} * x(n - 32 + 1) \quad (14)$$

Após a determinação do polinômio do filtro, adotou-se estratégias de cálculo que otimizem o processamento dos dados, por parte do processador. Uma das estratégias adotadas fora de utilizar a operação de deslocamento de bits para realizar a divisão por 32 (trinta e dois), onde tem-se a necessidade de deslocar os bits das variáveis em cinco posições para a direita.

Algumas outras otimizações e estratégias de execução do processamento dos dados, para obter a filtragem do sinal, estão apresentadas na forma de fluxograma, o qual encontra-se exposta na Figura 36.



**Figura 36 - Fluxograma do filtro digital média móvel**  
**Fonte: Autoria própria.**

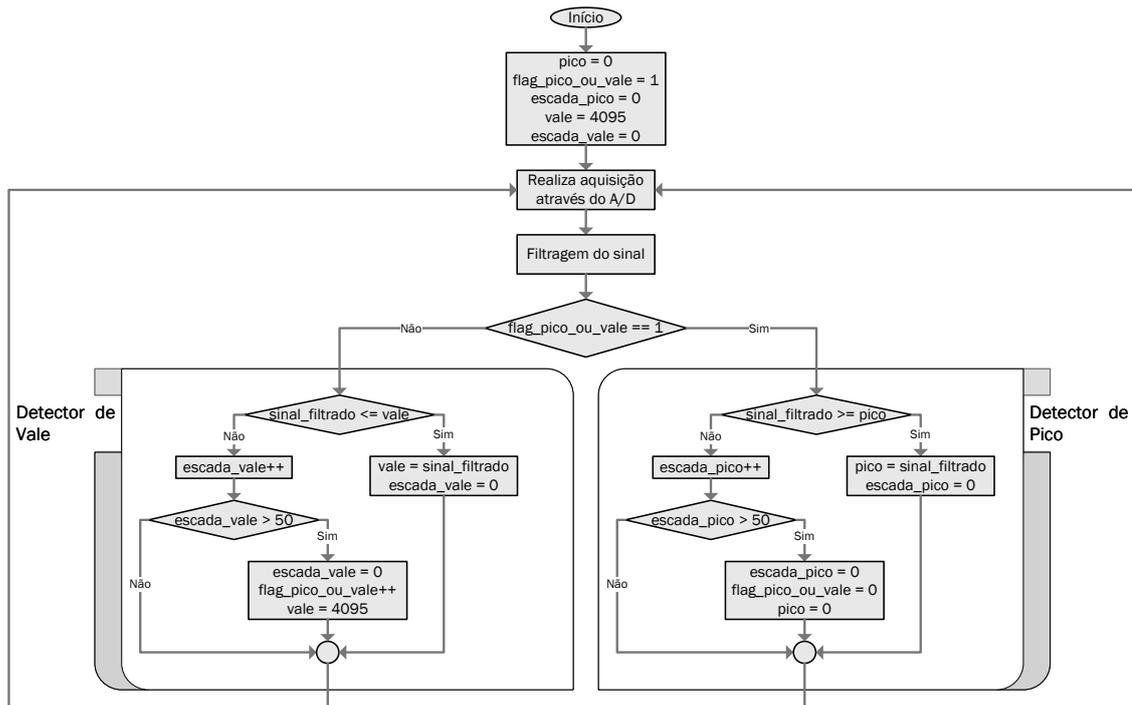
Observando o trecho circulado em vermelho na Figura 36, percebe-se que há uma inversão do sinal adquirido pelo A/D, através da subtração do valor do registrado pelo valor máximo que ele pode atingir. Esse ajuste é necessário em razão do fato que a forma de onda fotopletoisomografica é baseada na intensidade luminosa absorvida pelo tecido, porém o fotodiodo capta a luz que transpassou e não fora absorvida, portanto, faz-se necessária tal inversão.

### 3.7.2 Detecção dos picos e vales

Informação necessária para o cálculo da saturação percentual de oxigênio e da frequência cardíaca, os valores das amplitudes do pico e do vale, bem como a frequência com que ocorrem, devem ser obtidas através do processamento dos dados adquiridos via conversor A/D.

A necessidade de se detectar os picos e vales do sinal da curva fotopletoisomografica demandou de um algoritmo capaz de varrer os dados adquiridos e identificar quando ocorre um pico e quando ocorre um vale em tempo real. Para a elaboração de tal algoritmo, parte-se da premissa que o sinal assemelha-se a uma forma de onda senoidal e que todo pico é sucedido por um vale, portanto, leva-se em consideração a existência de dois possíveis cenários, um onde o algoritmo está buscando um pico e no outro o algoritmo está buscando um vale.

Para que a rotina compreenda que passou por um pico, utiliza-se uma variável que armazena o maior valor analisado até o momento, até que se identifique uma sequência de 50 (cinquenta) amostras inferiores ao valor do *buffer*. De maneira análoga, tal procedimento é realizado para detecção do vale. Esse processo está didaticamente apresentado no fluxograma da Figura 37.



**Figura 37 - Fluxograma do algoritmo de detecção de picos e vales**  
**Fonte: Autoria própria.**

Para estimar a frequência cardíaca, faz-se o uso do algoritmo de detecção de pico e, assim que se constata a identificação de um pico, é disparado um contador, o qual irá ser interrompido assim que se detectar um novo pico. É válido ressaltar o fato de que o cálculo dos BPM é anulado, caso o intervalo de tempo esteja fora da faixa de valores admitidos como válidos, a qual compreende os valores entre 38 BPM até 220 BPM.

### 3.8 SISTEMA DE COMUNICAÇÃO COM COMPUTADOR

De modo a complementar a análise experimental e enriquecer a qualidade da análise do processo de operação do projeto, adotou-se a estratégia de comunicação de dados do protótipo para o computador, via interface UART. Como o

kit de desenvolvimento fornecido pela Texas Instruments Incorporated dispõe de interface de conexão USB com o computador, tal processo fora viabilizado.

Para acionar o modo de comunicação, utilizou-se de uma chave tátil conectada ao pino P1.1, para indicar ao protótipo que o usuário deseja entrar em um dos modos de operação de comunicação com o computador. Existem dois tipos de modo de comunicação:

- MODO\_ENGENHARIA I: esse modo é selecionado bastando apenas manter a chave tátil pressionada, por alguns milésimos de segundo, enquanto o dispositivo é iniciado. Tal opção habilita o envio dos dados da curva, baseado na resolução apresentada no display, dos batimentos e da saturação de oxigênio;
- MODO\_ENGENHARIA II: esse modo é selecionado bastando apenas manter a chave tátil pressionada, por alguns segundos, enquanto o dispositivo é iniciado. Esse modo habilita o envio apenas do sinal fotopleletismografico, porém sem os ajustes realizados para exibição no display, fornecendo ao computador os dados referentes a amplitude do sinal.

A interface de comunicação UART fora configurada para envio de dados a uma velocidade de 115.200 bps e 8 bits de dados, com um bit de sinalização de parada. O envio é realizado cada vez que a curva percorre todo o visor, trabalhando em paralelo com a operação do protótipo.

Visando a recepção e retrabalho dos dados enviados pelo equipamento ao computador, realizou-se a elaboração de um algoritmo, na ferramenta MATLAB®, o qual tem por finalidade estabelecer a comunicação e tratar as informações, de modo que a curva fotopleletismografica venham a ser apresentadas graficamente ao usuário, e que também sejam enviados os dados referentes a saturação percentual de oxigênio e a frequência cardíaca.

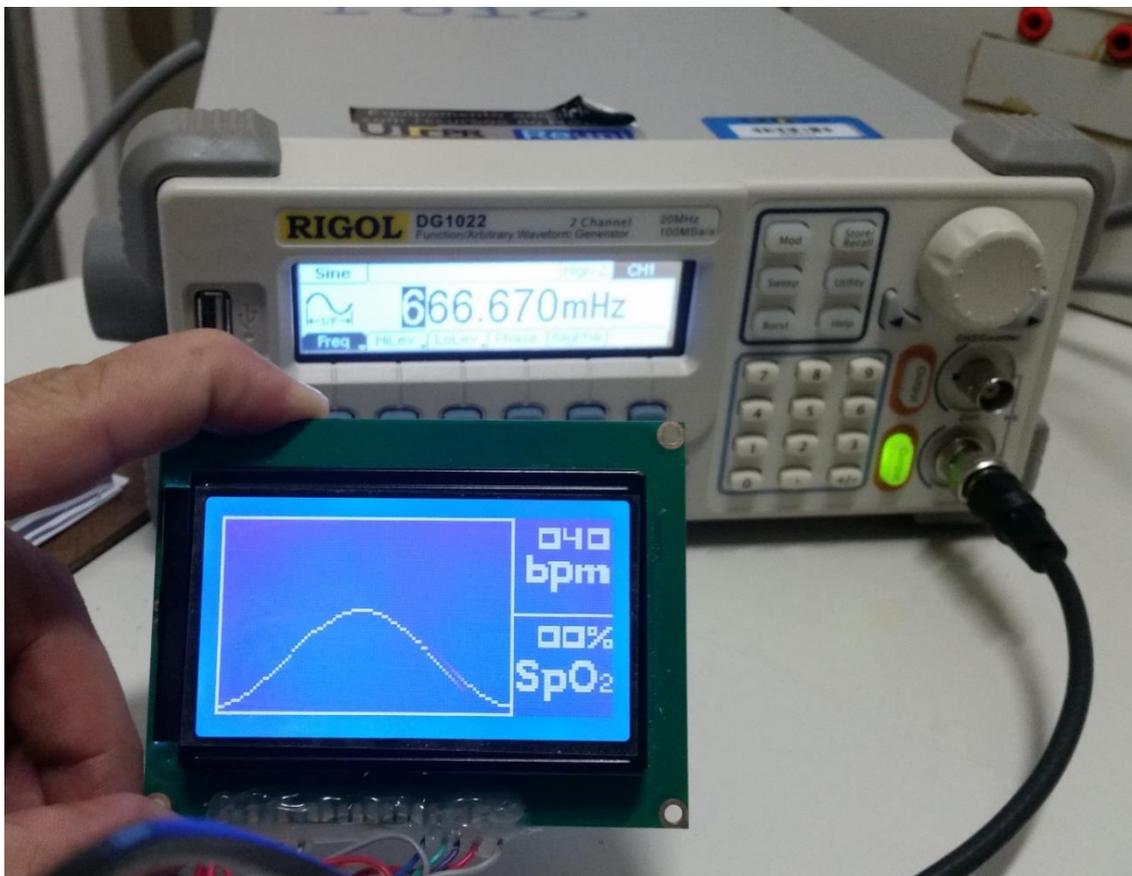
### 3.9 RESULTADOS E DISCUSSÕES

Visando analisar e identificar a capacidade de operação do algoritmo de cálculo do BPM, fora montado um circuito com o auxílio de um gerador de função, o qual passou a fornecer o sinal ao conversor A/D do microcontrolador, para que o

protótipo possa analisar e processar a onda conhecida, tornando-se viável a identificação do comportamento do dispositivo para diferentes frequências de entrada.

Para o teste experimental de aferição do medidor de frequência do protótipo, utilizou-se uma forma de onda senoidal, uma vez que assemelha-se à uma curva fotopletismografica, com tensão mínima de 1,8 V e máxima de 2,2 V.

Utilizou-se então, para a execução de sucessivas medições de frequências distintas, a identificação da frequência do sinal por parte do gerador de função e do protótipo, com respaldo da medição através do osciloscópio, observando o visor do equipamento desenvolvido para tal feito, assim como indicado na Figura 38.



**Figura 38 - Gerador de função, ajustado para uma tensão mínima de 1,8 V e máxima de 2,2 V, com frequência de 0,66667 Hz, e visor do protótipo de oximetria**  
Fonte: Autoria própria.

Tendo estruturada e apresentada a estratégia utilizada para a análise experimental do protótipo desenvolvido, parte-se para os testes de medição de frequência, onde iniciou-se com uma forma de onda de 0,7 Hz, equivalente à 42 BPM, e foram acrescentados 0,1 Hz nas medições posteriores, até que se atingisse a frequência

de 3,7 Hz, equivalente à 220 BPM. As informações coletadas encontram-se apresentadas na Tabela 2.

**Tabela 2 - Dados coletados no ensaio da operação do medidor de BPM, utilizando-se de um gerador de funções**

<b>Frequência (Hz)</b>	<b>Batimentos (BPM)</b>	<b>Valor no Display (BPM)</b>	<b>Erro (%)</b>
0,7	42	42	0,00%
0,8	48	48	0,00%
0,9	54	54	0,00%
1,0	60	60	0,00%
1,1	66	65	1,52%
1,2	72	72	0,00%
1,3	78	78	0,00%
1,4	84	84	0,00%
1,5	90	90	0,00%
1,6	96	96	0,00%
1,7	102	102	0,00%
1,8	108	108	0,00%
1,9	114	113	0,88%
2,0	120	120	0,00%
2,1	126	126	0,00%
2,2	132	132	0,00%
2,3	138	138	0,00%
2,4	144	144	0,00%
2,5	150	150	0,00%
2,6	156	157	0,64%
2,7	162	163	0,62%
2,8	168	168	0,00%
2,9	174	174	0,00%
3,0	180	180	0,00%
3,1	186	187	0,54%
3,2	192	192	0,00%
3,3	198	198	0,00%
3,4	204	204	0,00%
3,5	210	210	0,00%
3,6	216	216	0,00%
3,7	222	218	1,80%

**Fonte: Autoria própria.**

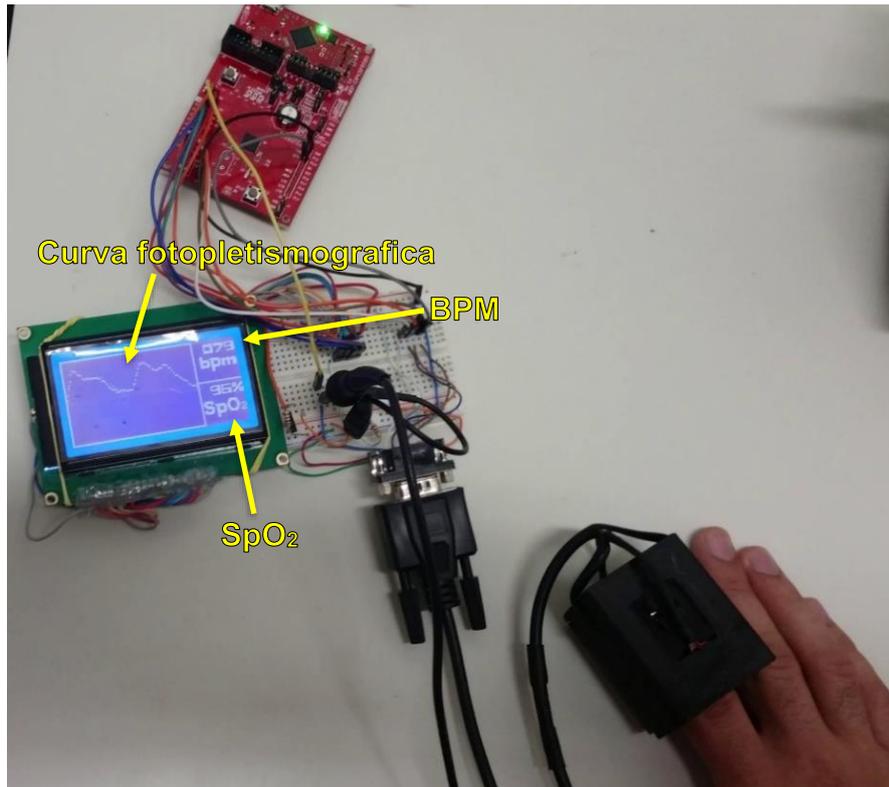
Na Tabela 2 estão apresentadas nas colunas, da esquerda para a direita, a frequência (em hertz) do sinal ajustado no visor do gerador de função, o valor da frequência do gerador convertido em BPM, o valor da frequência (em BPM) medido pelo oxímetro, e, por fim, o erro percentual absoluto da medição realizada pelo protótipo, em relação ao valor ajustado no gerador de função.

Analisando os valores correspondentes aos erros apresentados nas medições expostas na Tabela 2, tendo como base o gerador, percebe-se medições com um nível elevado de qualidade, onde tem-se, na maioria das vezes, erro nulo entre os valores indicados no gerador de função e do indicado no display do protótipo. Alguns pontos que apresentam divergência do valor esperado, podem ter seu erro gerado em função de interferências nas medições e/ou por conta de arredondamentos nos cálculos de frequência, uma vez que o dispositivo exibe apenas os dígitos de centena, dezena e unidade, o que pode implicar em um erro menor do que o indicado.

A medição que apresentou maior erro fora a última, realizada para um valor de frequência correspondente a 222 BPM, grande parte da contribuição para tal erro deve-se em virtude do fato de que o protótipo está projetado para operar em uma faixa de 38 BPM até 220 BPM, e de que o sinal passa a se aproximar consideravelmente do valor da frequência de corte do filtro, uma vez que essa frequência cardíaca corresponde a aproximadamente 3,7 Hz.

Identificada a capacidade de medição da frequência do sinal de entrada no conversor A/D, passa-se para o processo de análise da capacidade do protótipo em estimar o nível SpO<sub>2</sub>%. Para se obter valores comparativos de base, fora realizada a aquisição de um oxímetro de pulso da fabricante Quanta, modelo QTOX201, o qual indica tanto os valores de SpO<sub>2</sub> como os valores de BPM.

Para a análise da capacidade de operação do protótipo, utilizou-se de sucessivas medições do nível de SpO<sub>2</sub>, como indicado na Figura 39.



**Figura 39 - Teste de operação do protótipo**  
**Fonte: Autoria própria.**

Em intervalos de 10 segundos, anotando os valores apresentados tanto pelo equipamento comercial quanto pelo protótipo, de modo que fora possível compor a Tabela 3.

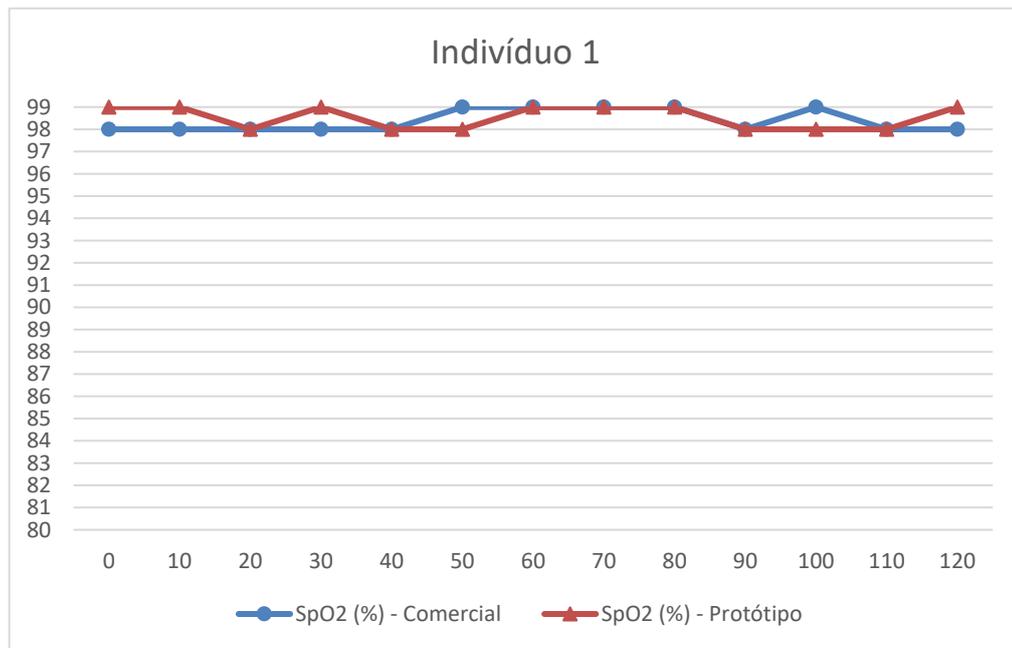
**Tabela 3 – Dados de SpO<sub>2</sub> coletados do Indivíduo 1**

<b>Tempo (s)</b>	<b>SpO<sub>2</sub> (%) - Comercial</b>	<b>SpO<sub>2</sub> (%) - Protótipo</b>	<b>Erro (%)</b>
0	98	99	1,02%
10	98	99	1,02%
20	98	98	0,00%
30	98	99	1,02%
40	98	98	0,00%
50	99	98	1,01%
60	99	99	0,00%
70	99	99	0,00%
80	99	99	0,00%
90	98	98	0,00%
100	99	98	1,01%
110	98	98	0,00%
120	98	99	1,02%

**Fonte: Autoria própria.**

Na Tabela 3 estão apresentadas nas colunas, da esquerda para a direita, o tempo decorrido, o valor indicado pelo equipamento comercial, o valor indicado pelo

protótipo, e o valor do erro absoluto percentual, em relação ao valor apresentado pelo dispositivo comercial. Fazendo o uso dos dados, apresentados na referida tabela, e visando apresentar graficamente o comportamento dos valores medidos, plotou-se as curvas apresentadas na Figura 40.



**Figura 40 - Gráfico das amostras de SpO<sub>2</sub> do Indivíduo 1, anotadas ao decorrerem do tempo, com auxílio do protótipo e de um dispositivo comercial**  
**Fonte: Autoria própria.**

Os dados apresentados na Figura 40 são referentes as medições realizadas em um indivíduo adulto do sexo masculino, de cor da pele ligeiramente parda. Percebe-se que os valores apresentam pouca discrepância, apresentando um erro médio de 0,47 %, demonstrando uma operação com erros aceitáveis ao estimar os níveis de SpO<sub>2</sub>%. A existência da divergência entre os valores, muito se dá pela dinâmica lenta apresentada pelo equipamento comercial, em virtude de que o equipamento realiza uma análise média entre uma maior quantidade de amostras e possui um algoritmo de processamento diferente, o que pode acarretar na não imediata identificação de alterações repentinas no nível de saturação.

Buscando a análise de um segundo tipo de indivíduos, foram repetidos os procedimentos realizados com o indivíduo 1 no indivíduo 2, o qual está enquadrado no grupo de pessoas adultas do sexo feminino e que possuem a cor da pele consideravelmente clara (branca). Logo, fora possível elaborar a Tabela 4.

Tabela 4 - Dados de SpO<sub>2</sub> coletados do Indivíduo 2

Tempo (s)	SpO <sub>2</sub> (%) - Comercial	SpO <sub>2</sub> (%) - Protótipo	Erro (%)
0	99	98	1,01%
10	98	98	0,00%
20	98	97	1,02%
30	98	92	6,12%
40	99	93	6,06%
50	99	96	3,03%
60	99	90	9,09%
70	98	96	2,04%
80	97	98	1,03%
90	98	99	1,02%
100	98	96	2,04%
110	98	98	0,00%
120	98	98	0,00%

Fonte: Autoria própria.

A Tabela 4 apresenta a mesma estrutura da Tabela 3, sendo composta por colunas, da esquerda para a direita, o tempo decorrido, o valor indicado pelo equipamento comercial, o valor indicado pelo protótipo, e o valor do erro absoluto percentual, em relação ao valor apresentado pelo dispositivo comercial. Fazendo o uso dos dados dessa tabela, realizou-se a plotagem das curvas comparativas, as quais encontram-se apresentadas na Figura 41.

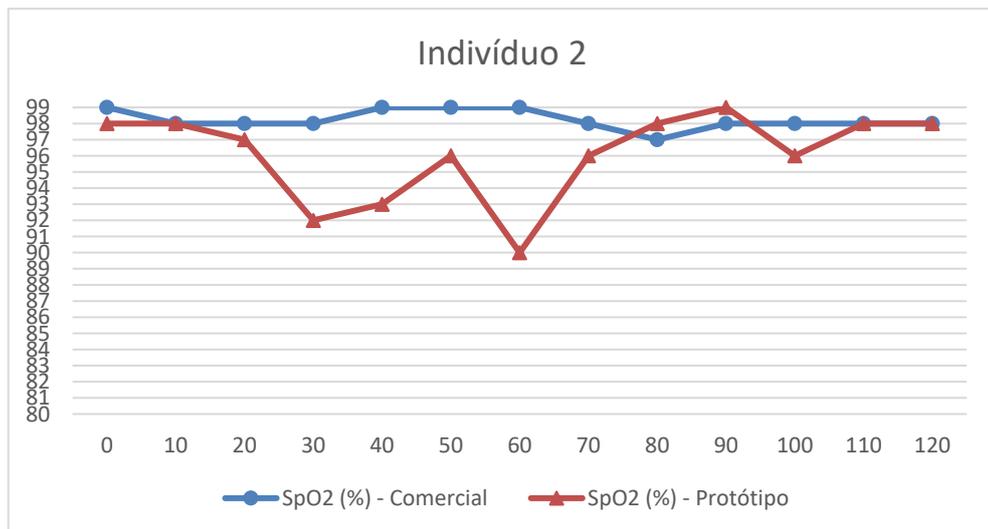


Figura 41 - Gráfico das amostras de SpO<sub>2</sub> do Indivíduo 2, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo

Fonte: Autoria própria.

Através da análise comportamental das curvas apresentadas na Figura 41, percebe-se que, em alguns momentos, a diferença entre os valores de referência e o valor do protótipo torna-se expressiva, isso se dá, principalmente, em função da cor

da pele do indivíduo 2, a qual não apresenta um nível razoável de absorção, fazendo com que o sensor trabalhe, em alguns momentos, na região de saturação, demonstrando uma incapacidade de operar com precisão em alguns indivíduos. O erro absoluto percentual manteve-se abaixo de 9,1 % e ficou em média de 2,5 %. Tal diferença também pode ser proveniente da dinâmica adotada pelo fabricante comercial, quando comparado ao protótipo (semelhante ao caso do indivíduo 1), e também deve-se levar em consideração o fato de ambos os dispositivos apresentam apenas o valor da dezena e da unidade de SpO<sub>2</sub>, o que pode acarretar em um menor erro entre as medições.

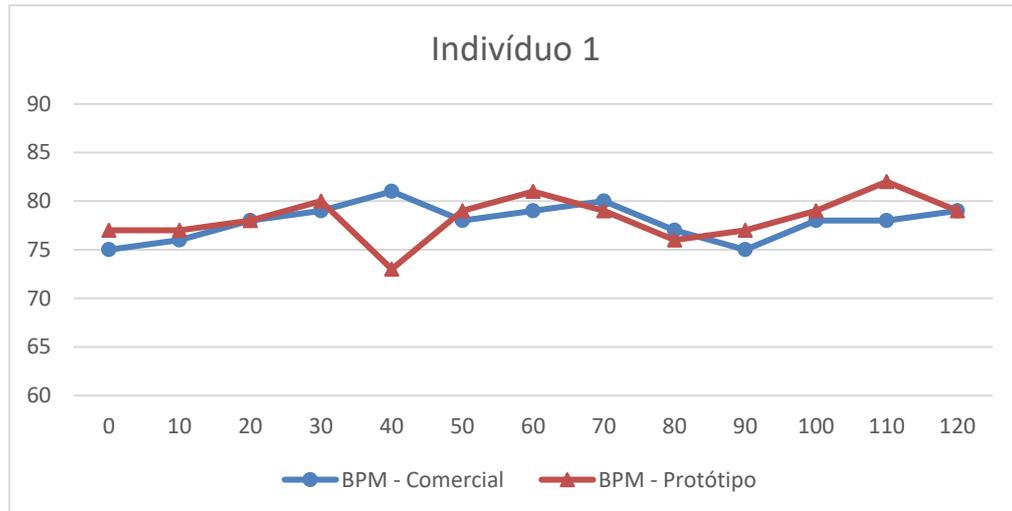
Concluída a análise da capacidade do dispositivo em estimar o valor de SpO<sub>2</sub>, busca-se analisar o desempenho do equipamento ao analisar os batimentos cardíacos dos indivíduos 1 e 2. Para tal feito, utiliza-se dos mesmos procedimentos empregados na análise da SpO<sub>2</sub>, aplicados agora na coleta dos dados referentes aos batimentos, seguindo uma sequência de amostragens ao longo do tempo. Através desse levantamento, possibilitou-se a elaboração da Tabela 5.

**Tabela 5 - Dados de BPM coletados do Indivíduo 1**

Tempo (s)	BPM - Comercial	BPM - Protótipo	Erro (%)
0	75	77	2,67%
10	76	77	1,32%
20	78	78	0,00%
30	79	80	1,27%
40	81	73	9,88%
50	78	79	1,28%
60	79	81	2,53%
70	80	79	1,25%
80	77	76	1,30%
90	75	77	2,67%
100	78	79	1,28%
110	78	82	5,13%
120	79	79	0,00%

**Fonte: Autoria própria.**

As informações apresentadas nas colunas, da esquerda para a direita, referem-se ao tempo decorrido até cada amostragem, ao valor de BPM fornecido pelo dispositivo comercial, ao valor de BPM fornecido pelo protótipo, e o erro percentual absoluto da mediação realizada pelo equipamento desenvolvido. Para que se possa melhor analisar os dados coletados, adota-se a estratégia gráfica, traçando as curvas referentes aos valores indicados pelos dispositivos ao longo do tempo, tais curvas encontram-se plotadas na Figura 42.



**Figura 42 - Gráfico das amostras de BPM do Indivíduo 1, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo**  
**Fonte: Autoria própria.**

As curvas apresentadas na Figura 42 apresentam alguns trechos de desencontro, tal comportamento fora observado e acredita-se que seja, principalmente, em razão da dinâmica do equipamento comercial, o qual apresenta um comportamento inercial demasiadamente elevada para atualização dos dados a serem exibidos. Ademais, o protótipo apresentou valores próximos dos indicados pelo equipamento comercial, aparentando um erro médio de 2,35 %.

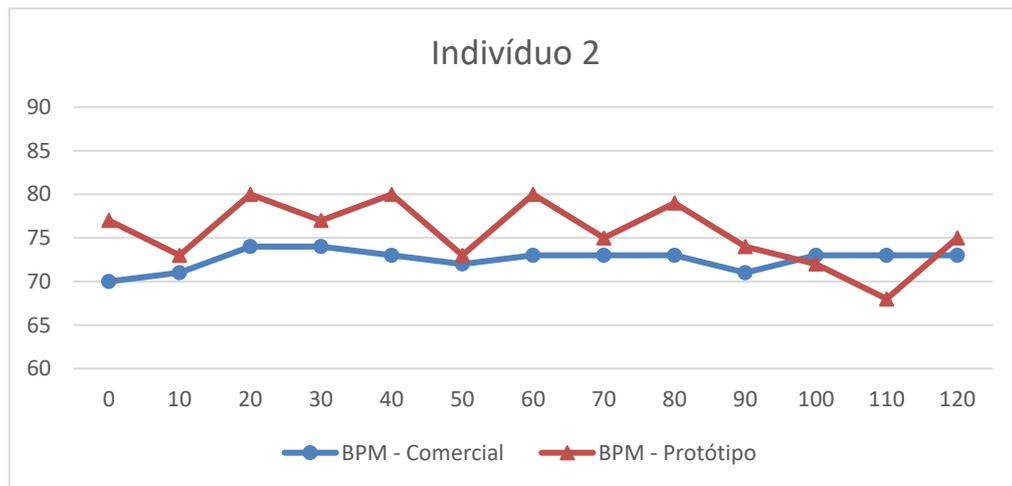
Finalizada os experimentos realizados com o indivíduo 1, passa-se para o processo de análise do desempenho dos dispositivos conectados ao indivíduo 2. Foram em então repetidos os procedimentos, de modo que obtiveram-se dados suficientes para compor a Tabela 6.

**Tabela 6 - Dados de BPM coletados do Indivíduo 2**

Tempo (s)	BPM - Comercial	BPM - Protótipo	Erro (%)
0	70	77	10,00%
10	71	73	2,82%
20	74	80	8,11%
30	74	77	4,05%
40	73	80	9,59%
50	72	73	1,39%
60	73	80	9,59%
70	73	75	2,74%
80	73	79	8,22%
90	71	74	4,23%
100	73	72	1,37%
110	73	68	6,85%
120	73	75	2,74%

**Fonte: Autoria própria.**

Tem-se apresentado nas colunas da Tabela 6, seguindo da esquerda para a direita, o intervalo de tempo decorrido até a referida amostragem, o valor de BPM anotado a partir do equipamento comercial, o valor de BPM identificado através do protótipo, e o erro percentual absoluto. Os dados que compõe a tabela, foram utilizados para traçar as curvas dos valores de BPM ao decorrer do tempo, para que se possa melhor analisar as informações, obtendo-se assim a Figura 43.



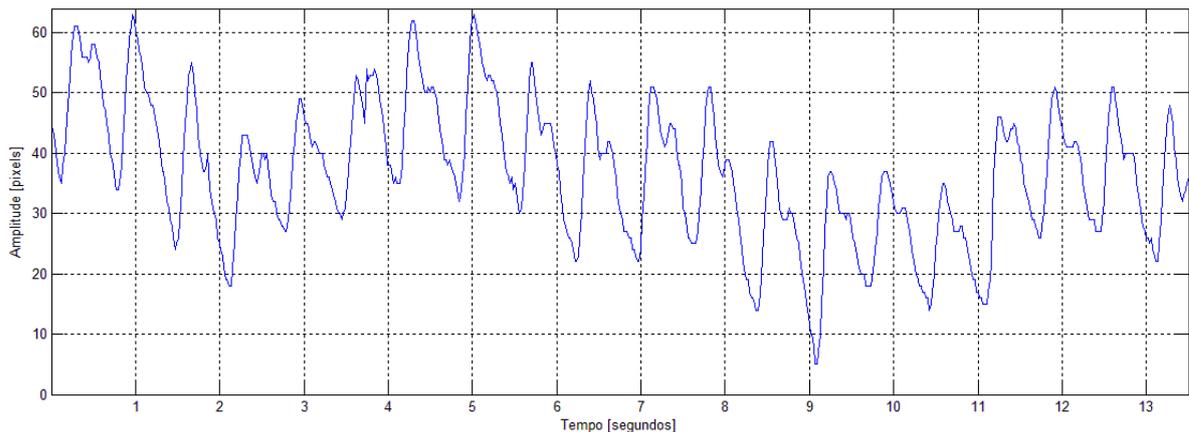
**Figura 43 - Gráfico das amostras de BPM do Indivíduo 2, anotadas ao decorrerem do tempo, com auxílio de um dispositivo comercial e do protótipo**  
**Fonte: Autoria própria.**

O comportamento da curva referente aos dados coletados por meio do protótipo, a qual está apresentada na Figura 43, apresenta divergências entre os valores indicados pelo equipamento comercial e pelo protótipo. Assim como citado no caso do indivíduo 1, além da particularidade no modo de operação do dispositivo comercial e do número de casas indicados pelos equipamentos, outro fator que contribui para o aumento desse erro é o fato de que o indivíduo possui uma pele de coloração clara, o que torna o sensor mais susceptível entrar no nível de saturação do A/D, causando imprecisão em alguns pontos. Essa imprecisão resulta em um erro médio de 5,51 %, não apresentando um bom desempenho, sendo necessários o estudo e aplicação de novos métodos em trabalhos futuros, tais como a utilização de um potenciômetro digital para ajustar o ganho do OPT101 em tempo real, de modo que se consiga obter maior resolução na aquisição do sinal.

Finalizados os ensaios e análise do desempenho do protótipo, buscou-se também, de maneira complementar, estabelecer um sistema de comunicação entre o *kit* de desenvolvimento e o computador, buscando enriquecer as informações já

apresentadas ao decorrer desse trabalho. Através do sistema de comunicação, foram realizadas aquisições de dados contidos no microcontrolador, de modo que se possa analisa-los e formalizar o registro dos valores coletados através do equipamento.

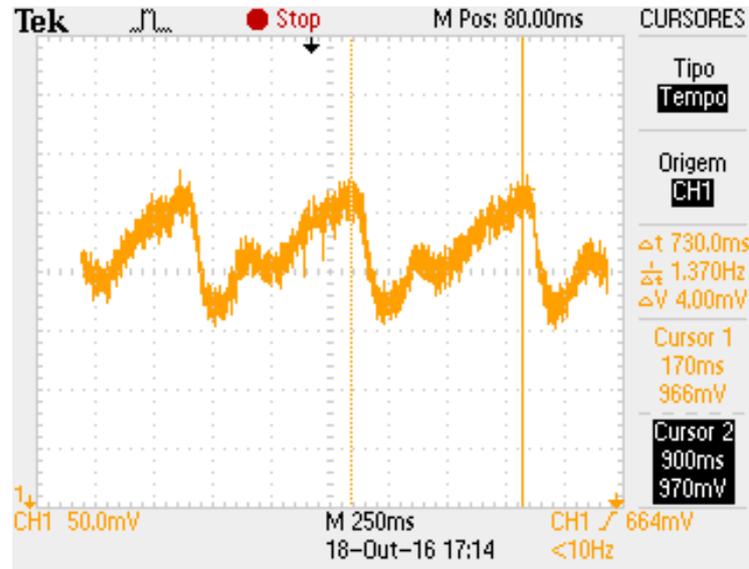
Em um primeiro momento, realizou-se a aquisição dos dados de cada uma das formas de ondas correspondentes aos 2 (dois) comprimentos de onda, infravermelho e vermelho, de modo que se possa analisar a qualidade da forma de onda que o microcontrolador processa. Para possibilitar tal análise, ajusta-se o modo de comutação dos LEDs, fixando em apenas um das cores, anulando assim o processo de comutação. É necessário ressaltar o fato de que o botão de seleção do modo engenharia seja pressionado, nesse caso seleciona-se o modo número 1, visando primeiramente a aquisição da forma de onda correspondente ao LED infravermelho, com isso, obtêm-se a forma de onda apresentada na Figura 44.



**Figura 44 - Curva fotopletismografica enviada do microcontrolador para o computador, referente ao LED infravermelho**  
**Fonte: Autoria própria.**

A estrutura vertical do eixo de plotagem é baseada na grade de pixels do visor, uma vez no “MODO ENGENHARIA 1” consiste em importar diretamente os valores que são enviados para a tela do oxímetro. A curva apresentada é resultante do ajuste automático de amplitude, o qual visa obter uma melhor interface de indicação ao operador, dando maior resolução aos dados exibidos na tela.

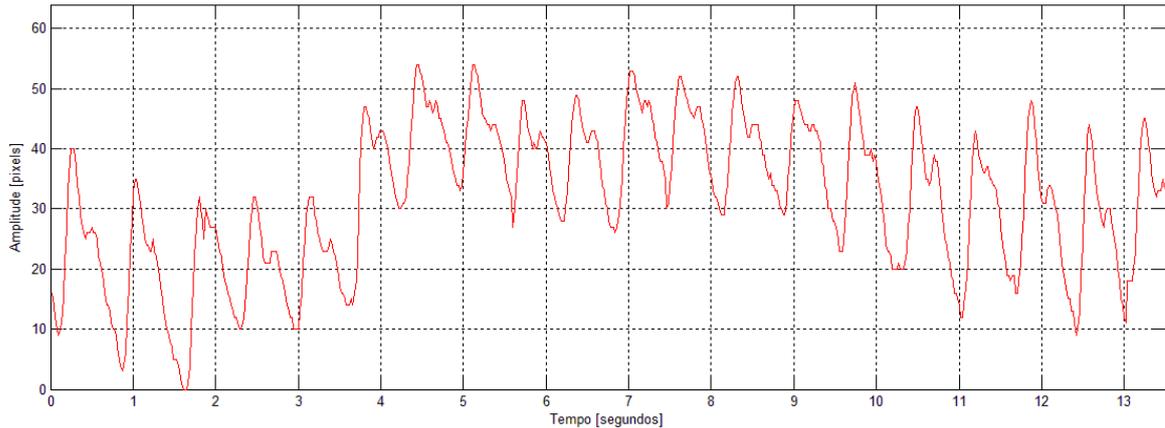
De modo a complementar e constatar a qualidade com que a forma de onda é adquirida e processada pelo dispositivo, capturou-se a onda na entrada do conversor A/D, utilizando-se de um osciloscópio, e obteve-se a Figura 45.



**Figura 45 - Forma de onda na entrada do conversor A/D, correspondente ao LED Infravermelho**  
**Fonte: Autoria própria.**

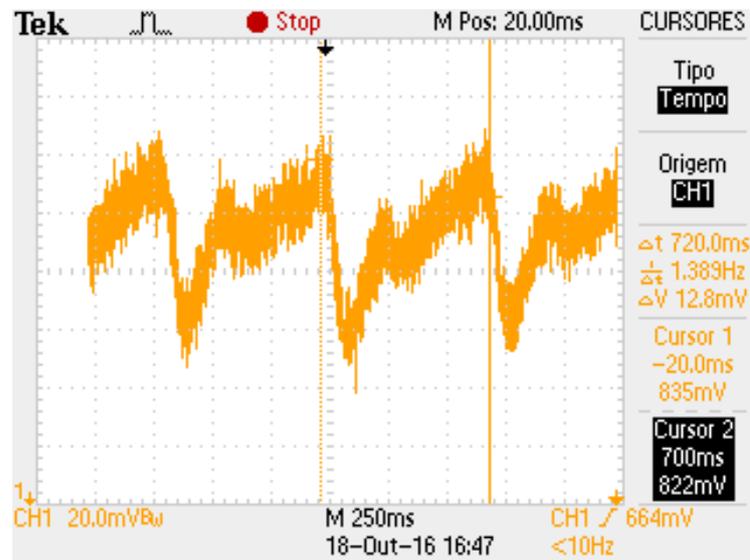
Comparando a forma de onda apresentada na Figura 44 com a da Figura 45, percebe-se que uma encontra-se espelhada em relação a outra. Esse fato se dá em razão de que a curva fotopletiométrica é composta pelo nível de absorção da luz incidente, porém o fotodiodo capta a luz que não fora absorvida pelo tecido e envia ao conversor A/D, o qual está configurado com um sinal de referência interna do microcontrolador para mínima de 0 V e máxima de 2,5 V. Portanto, faz-se necessária a inversão da forma de onda, de maneira digital, através do microcontrolador. Percebe-se também que o sinal apresentado pelo microcontrolador apresenta pouco ruído, quando comparamos com a imagem obtida através do osciloscópio, tal fato pode ocorrer em virtude da própria resolução do osciloscópio, o que acarreta na exibição de valores imprecisos.

Na sequência, faz-se os mesmos procedimentos para o LED vermelho, ajustando o sistema de acionamento dos LEDs para fixar a cor vermelha acesa ininterruptamente durante o teste, com isso, obtém-se a Figura 46.



**Figura 46 - Curva fotopletiométrica enviada do microcontrolador para o computador, referente ao LED vermelho**  
**Fonte: Autoria própria.**

Visando a comparação entre o sinal enviado pelo microcontrolador, apresentado na Figura 46, e o sinal analógico na entrada do conversor, utilizou-se de um osciloscópio para captura da forma de onda, obtendo-se a Figura 47.



**Figura 47 - Forma de onda na entrada do conversor A/D, correspondente ao LED Vermelho**  
**Fonte: Autoria própria.**

Um comparativo entre a forma de onda apresentada na Figura 47, referente ao osciloscópio, e a forma de onda da Figura 46, referente ao sinal enviado pelo protótipo ao computador, demonstra um bom nível de qualidade na exibição dos dados pelo protótipo, sendo apresentados com uma atenuação do ruído, dando qualidade ao sinal a ser processado.

## 4 CONCLUSÕES

A revisão bibliográfica realizada durante o desenvolvimento do trabalho de conclusão de curso, fomentou o conhecimento necessário para adotar e definir estratégias voltadas a elaboração do protótipo para aquisição e exibição dos sinais de oximetria de pulso, estando apresentada ao decorrer do capítulo 2, onde foram estudados trabalhos semelhantes. Esse estudo auxiliou na decisão de se utilizar um fotosensor do tipo fotodiodo, mais especificamente o OPT101, na construção da ponteira do oxímetro, bem como o comprimento de onda das cores dos LEDs utilizados para polarizar tal sensor. Essas escolhas mostraram-se eficientes na operação dos componentes como parte da ponteira do oxímetro.

A escolha do microcontrolador a ser utilizado no protótipo fora em virtude da análise técnica dos dados fornecidos pelas fabricantes, do estudo de outros trabalhos da área e, principalmente, na experimentação prática obtida ao decorrer dos trabalhos desenvolvidos para o TCC. Seguindo essa estratégia de análise, definiu-se como microcontrolador o modelo MSP430FR5969, o qual dispõe de capacidade de processamento de 16 MHz, de elevada capacidade de acesso à memória (fixada em 8 MHz), de um conversor A/D de 12 bits e da disponibilidade de se utilizar de um kit de desenvolvimento, fornecido pela fabricante Texas Instruments Incorporated, o qual proporciona ao usuário um ambiente estável para o desenvolvimento, além de possibilitar o acompanhamento em tempo real das instruções executadas pelo controlador. Esse processador atendeu as necessidades demandadas pela atividade, mostrando-se uma escolha compatível com a aplicação.

Com relação as definições traçadas na proposta, conseguiu-se desenvolver um protótipo de oxímetro de pulso, o qual apresenta a forma de onda fotopletismografica, taxa percentual estimada da saturação de oxigênio do sangue arterial, e a informação da frequência cardíaca, todos esses dados são exibidos ao usuário por meio de visor gráfico 128x64 pixels. Em paralelo com a indicação visual das informações, o equipamento também dispôs da indicação sonora do ritmo do pulso, utilizando-se de um alto-falante piezoelétrico, com variação da “intensidade” atrelada à variação do nível de saturação de oxigênio no sangue, tornando-se mais intensa quando esses níveis vão reduzindo.

Conseguiu-se, projetar e implementar a ponteira que compõe o oxímetro, para isso utilizou-se de uma estrutura de suporte dos LEDs e do OPT101 projetada

em 3D, a qual fora confeccionada através da impressão de peças em 3D. A ponteira atendeu a capacidade de atuar como unidade sensora, o que fora comprovado através dos resultados apresentados no capítulo 3.9.

Como trabalhou-se com a aquisição de sinais analógicos, fez-se necessária a criação de um algoritmo de aquisição, via conversor A/D, e a elaboração de uma rotina de filtro digital, para redução da interferência ruidosa.

Em meio a necessidade de criar uma interface de exibição de dados gráficos ao usuário, utilizou-se um display de 128x64 pixels. Para que se fosse possível o controle e operação desse tipo de display, desenvolve-se toda a biblioteca do visor, com sua estrutura voltada especificamente para a aplicação no protótipo, o que mostrou-se satisfatório ao atender as necessidades do protótipo. A estrutura disponível na biblioteca é composta por uma tela de inicialização, uma tela de operação e uma biblioteca de números, com um sistema de busca otimizado.

Portanto, em relação aos objetivos geral e específicos, conseguiu-se alcança-los, uma vez que o protótipo mostrou-se funcional e com capacidade de operação aceitável. Os dados obtidos mostraram-se próximos aos do equipamento comercial utilizado como referência, apresentando divergência significativa em algumas medições, em função das peculiaridades do paciente e limitações do protótipo.

De modo complementar aos objetivos definidos, visando enriquecer ainda mais os dados e a qualidade do trabalho, desenvolve-se um sistema de comunicação UART entre o kit de desenvolvimento e o computador, permitindo que fossem arquivados e retrabalhados os dados processados pelo protótipo, através de uma rotina de importação de dados via MATLAB®. Essa estratégia mostrou-se muito assertiva, uma vez que se a aquisição e armazenamento de dados tem sua capacidade expandida, haja vista que os computadores possuem sistemas de armazenamento de dados demasiadamente maiores do que a maioria dos microcontroladores.

Em um comparativo com a maioria das bibliografias utilizadas ao decorrer desse trabalho, observou-se o fato de que os demais autores utilizaram-se de ponteiras de aquisição comerciais, a qual, neste trabalho, fora todo desenvolvida e produzida, proporcionando um maior aproveitamento dos conhecimentos e práticas. Porém, o protótipo não apresentou níveis de operação de elevada qualidade, sendo necessária a análise e incorporação de novas técnicas, para trabalhos futuros,

voltadas a melhoria na operação do equipamento em diferentes indivíduos, dentre essas técnicas pode-se destacar a utilização de um potenciômetro digital, para ajuste de ganho do sensor, e melhorias nos algoritmos de processamento dos dados, visando a eliminação de dados inválidos.

Como fora proposto desenvolver todo o sistema de sensoriamento do protótipo, não se utilizou uma ponteira de oximetria comercial, normalmente adotada nos trabalhos das bibliografias revisadas. Esse fato tornou mais complexa a aquisição do sinal de oximetria, uma vez que o nível do sinal a ser adquirido a partir do fotosensor é de ordem muito pequena, alguns milivolts, sofrendo grande interferência de fontes ruidosas que, normalmente, não apresentariam grande impacto em circuitos que trabalhem com sinais de maior ordem. Durante a análise dos sinais obtidos através do fotosensor, houve, então, a necessidade de se trabalhar com equipamentos de medição que gerassem um impacto ruidoso de ordem muito pequena, no circuito de aquisição do sinal. Em decorrência desse fato, necessitou-se de um sistema de aterramento elétrico, devidamente funcional, o qual teve-se que adaptar para que se fosse possível analisar e medir os sinais. Tal circunstância gerou atrasos na evolução do sensor óptico, uma vez que sua detecção e solução não fora imediata.

Ao decorrer do desenvolvimento desse trabalho, mostrou-se imprescindível o resgate dos conceitos e teorias abordados dentro das disciplinas da graduação, nas áreas de: digitais, englobadas pelas disciplinas de Programação A e B, Circuitos Digitais, Sistemas Digitais, Sistemas Microcontrolados e Controle Digital; eletrônica, abordada nas disciplinas de Eletrônica A, B e C; instrumentação, contemplada na disciplina de Instrumentos de Medidas Elétricas e Instrumentação Eletro-Eletrônica; e também, em nível mais genérico, na área de circuitos, na disciplinas de Análise de Circuitos Elétricos 1.

## REFERÊNCIAS

ANDRADE, Luiz A. K. de. **Sistema de medição para oximetria de pulso**. 2009. 88 f. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-Graduação em Engenharia e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2009. Disponível em: <[http://repositorio.utfpr.edu.br/jspui/bitstream/1/912/1/CT\\_CPGEI\\_M\\_Andrade%2c%20Luiz%20Augusto%20Kalva%20de\\_2009.pdf](http://repositorio.utfpr.edu.br/jspui/bitstream/1/912/1/CT_CPGEI_M_Andrade%2c%20Luiz%20Augusto%20Kalva%20de_2009.pdf)>. Acesso em: 6 mai. 2016.

ENERGIA. **Guide to the MSP430FR5969 LaunchPad (MSP-EXP430FR5969). Energia**. Disponível em: <<http://energia.nu/wordpress/wp-content/uploads/2014/06/MSP430FR5969.jpeg>>. Acesso em: 22 ago. 2016.

FAIRCHILD. **MC78XX/LM78XX/MC78XXA**. Disponível em: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/82833/FAIRCHILD/LM7805.html>>. Acesso em: 2 mai. 2016.

INGLE, Vinay K.; PROAKIS, John G.. **Digital Signal Processing Using MATLAB (R)**. 3ª. ed. Stamford: Cengage Learning, 2011. 652 p.

KUSHIMA, Mário A.. **Oxímetro de pulso para medição da oxigenação periférica e pulsação cardíaca com interface via web**. 2014. 80 f. Trabalho de Conclusão de Curso (Graduação) - Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2014. Disponível em: <[http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-26012015-141909/publico/Kushima\\_Mario\\_Augusto.pdf](http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-26012015-141909/publico/Kushima_Mario_Augusto.pdf)>. Acesso em: 30 mar. 2015.

LIMA, Daniel W. de C.. **Oxímetro de pulso com transmissão de sinal sem fios**. 2009. 79 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/24302/000736259.pdf?sequence=1>>. Acesso em: 8 mai. 2016.

MARQUES, Angelo E. B.; JÚNIOR, Salomão C.; CRUZ, Eduardo C. A.. **Dispositivos semicondutores: diodos e transistores**. São Paulo: Editora Érica, 1996.

MARTINS, Rui M. S.. **Desenvolvimento de um Sensor de Fotopletismografia para Monitorização Cardíaca para aplicação no Pulso**. 2010. 90 f. Dissertação (Mestrado em Engenharia Biomédica) - Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2010. Disponível em: <[https://estudogeral.sib.uc.pt/jspui/bitstream/10316/14086/1/Tese\\_MIEB\\_Rui%20Martins.pdf](https://estudogeral.sib.uc.pt/jspui/bitstream/10316/14086/1/Tese_MIEB_Rui%20Martins.pdf)>. Acesso em: 21 mar. 2015.

MATTHES, K.. Untersuchungen über die Sauerstoffsättigung des menschlichen Arterienblutes. **Naunyn-Schmiedebergs Archiv für experimentelle Pathologie und Pharmakologie**, Alemanha, v. 179, p. 698-711, 1935. ISSN 0365-2009. Disponível em: <<http://dx.doi.org/10.1007/BF01862691>>. Acesso em: 25 mar 2015.

NATIONAL SEMICONDUCTOR. **LM117/LM317A/LM317 3 - Terminal Adjustable Regulator**. Disponível em: <<http://pdf1.alldatasheet.com/datasheet->

pdf/view/8619/NSC/LM317/+5WQ9uIOzZKMA9z+/datasheet.pdf>. Acesso em: 2 mai. 2015.

SANTINI, Tales R. de S.. **Projeto de um Oxímetro de Pulso com comunicação USB**. 2010. 49 f. Trabalho de Conclusão de Curso (Graduação) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010. Disponível em: <[http://www.tcc.sc.usp.br/tce/disponiveis/18/180500/tce-17112011-111219/publico/Santini\\_Tales\\_Roberto\\_de\\_souza.pdf](http://www.tcc.sc.usp.br/tce/disponiveis/18/180500/tce-17112011-111219/publico/Santini_Tales_Roberto_de_souza.pdf)>. Acesso em: 28 mar. 2015.

SITRONIX. **ST7920**. Disponível em: <<http://www.hpinfotech.ro/ST7920.pdf>>. Acesso em: 7 mai. 2016.

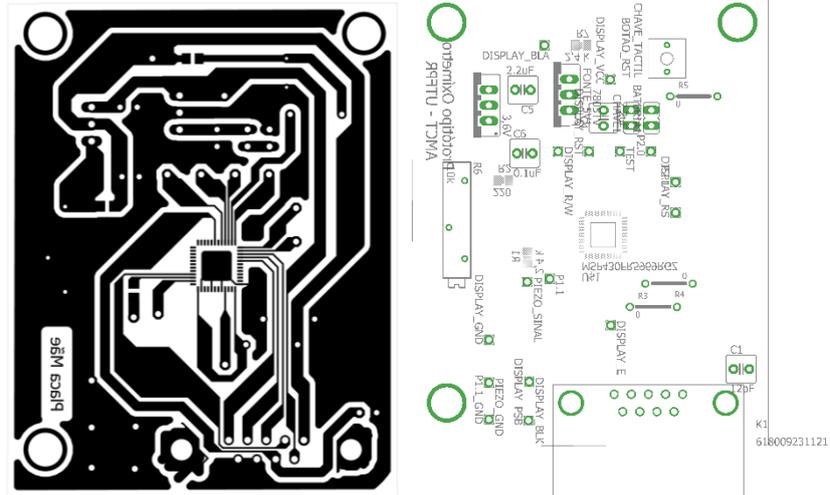
TEXAS INSTRUMENTS INCORPORATED. **LM317 3 - Terminal Adjustable Regulador**. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm317.pdf>>. Acesso em: 2 mai. 2015.

TEXAS INSTRUMENTS INCORPORATED. **MSP430FR5969 LaunchPad Development Kit. Texas Instruments Incorporated**. Disponível em: <<http://www.ti.com/lit/ds/slas704e/slas704e.pdf>>. Acesso em: 12 set. 2016.

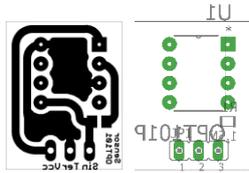
TEXAS INSTRUMENTS INCORPORATED. **OPT101 Monolithic Photodiode and Single-Supply Transimpedance Amplifier**. Disponível em: <<http://www.ti.com/lit/ds/symlink/opt101.pdf>>. Acesso em: 3 out. 2015.

## APÊNDICE A - LAYOUT DAS PLACAS

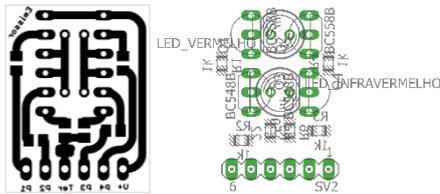
### PLACA MÃE



### FOTOSENSOR



### CIRCUITO DE ACIONAMENTO DOS LEDS



## APÊNDICE B - CÓDIGO PRINCIPAL “MAIN.C”

```

/*=====
*
*          TRABALHO DE CONCLUSAO DE CURSO
*    - Desenvolvimento de protótipo para aquisição e exibição dos sinais de
*    oximetria de pulso
*
*
*    Academico: Allan Magno Costa Teixeira
*
/*=====

/*=====INCLUSAO DE BIBLIOTECAS=====*/
#include <msp430fr5969.h>
#include <stdio.h>
#include <math.h>

#include "inicializa_display.h"
#include "biblioteca_lcd.h"
/*=====*/

unsigned int valor = 0;
int k = 0;

//Maximo 220bpm e minimo 40bpm

/*=====DEFINICOES E VARIAVEIS=====*/
/*-----Acionamento dos LEDs-----*/
unsigned char controle_LEDs = 0; /* Responsavel pela comutacao do acionamento
* dos LEDs
*/

#define VERMELHO          (0x60)
#define INFRAVERMELHO    (0x11)
#define NENHUM           (0x41)
/*-----*/

/*-----Acionamento Sonoro-----*/
unsigned char bpm_som = 0;
unsigned char intensidade_som = 20;
/*-----*/
/*-----Filtragem dos Sinais-----*/
/*- - - - - Infravermelho- - - - -*/
short int sinal_filtrado_IV = 0; // Valor do sinal IV ja filtrado
short int filtragem_IV[32];      /* Vetor que armazena os valores durante a
* filtragem
*/

unsigned char fil_IV = 0;        // Variavel de varredura do armazenamento
/*- - - - -*/

/*- - - - -Vermelho- - - - -*/
short int sinal_filtrado_V = 0; // Valor do sinal V ja filtrado
short int filtragem_V[32];      /* Vetor que armazena os valores durante a
* filtragem
*/

unsigned char fil_V = 0;        // Variavel de varredura do armazenamento
/*- - - - -*/

```







```

/*=====CONFIGURACOES/INICIALIZACOES=====*/
void Fonte_Clock(void); // Sistema de clock
void Portas_P1_P2_P3_P4(void); // Portas P1, P2, P3 e P4
void Timer_A0(void); // Timer A0
void Timer_A1(void); // Timer A1
void Timer_A2(void); // Timer A2
void Conversor_AD12bit(void); // Conversor A/D de 12-bit
void Comunicacao_SPI(void); // Comunicacao SPI
void Comunicacao_UART(void); // Comunicacao UART

void filtrar (short int *sinal_filtrado, short int *filtragem, unsigned char
*fil);
void detectar_pico (short int *sinal, short int *pico, unsigned char *flag,
unsigned char *escada_pico, unsigned char *passo, unsigned char *flag_calcular);
void detectar_vale (short int *sinal, short int *vale, unsigned char *flag,
unsigned char *escada_vale, unsigned char *flag_calcular);
/*=====*/

/*=====ROTINA PRINCIPAL=====*/
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // Para o watchdog timer

    Fonte_Clock();

    __delay_cycles(1600000); /* Delay inicial para aguardar a
    * estabilizacao da fonte de alimentacao,
    * visando eliminar o problema encontrado no
    * prototipo (sem o kit) ao acionar a chave
    * que liga a alimentacao do equipamento.
    * -> Obs.: Do contrario o display nao
    * funciona corretamente, sendo
    * necessario acionar o botao de
    * reset, para que ele funcione.
    */

    Portas_P1_P2_P3_P4();
    Comunicacao_SPI();

    if(P1IN == 0) // Verifica se o botao esta precionado
    {
        MODO_ENGENHARIA = 1; // Habilita o modo engenharia I
        Comunicacao_UART();
        __delay_cycles(1600000);

        if(P1IN == 0) // Verifica se o botao esta precionado
        {
            MODO_ENGENHARIA = 2; // Habilita o modo engenharia II
        }
    }

    ini_display_128x64();
    apresenta_tela_inicial();

    __delay_cycles(1600000);
    __delay_cycles(1600000);
}

```

```

apresenta_tela_operacao();

if (MODO_ENGENHARIA == 1)
{
    /*-----Indica o Modo Engenharia I-----*/
    num_uni_bpm = 1;
    num_dez_bpm = 11;
    num_cen_bpm = 10;
    desenha_bpm(&num_uni_bpm, &num_dez_bpm, &num_cen_bpm);
    /*-----*/
}
else if (MODO_ENGENHARIA == 2)
{
    /*-----Indica o Modo Engenharia II-----*/
    num_uni_bpm = 2;
    num_dez_bpm = 11;
    num_cen_bpm = 10;
    desenha_bpm(&num_uni_bpm, &num_dez_bpm, &num_cen_bpm);
    /*-----*/
}

Timer_A0();
Timer_A1();
Timer_A2();

Conversor_AD12bit();

__bis_SR_register(GIE); // seta o bit GIE - permite geracao de interrupcoes

do
{
    // Loop infinito
}while(1);
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DO CLOCK=====*/
void Fonte_Clock(void)
{
    FRCTL0 = FRCTLPW | NWAITS_5; //
    CSCTL0_H = CSKEY >> 8; // Senha de acesso aos registradores

    CSCTL1 = DCORSEL | DCOFSEL_4; // Seleciona o sinal de 16MHz
    CSCTL2 = SELA__LFXTCLK | SELS__DCOCLK | SELM__DCOCLK;
    CSCTL3 = DIVS__2 + DIVM__1; // Divide SMCLK e MCLK por 2 -> 8MHz
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DAS PORTAS=====*/
void Portas_P1_P2_P3_P4(void)
{
    /*-----Porta 1-----*/
    P1DIR = 0x04;
    P1REN = 0x02;
    P1OUT = 0x02;
    /*-----*/

    /*-----Porta 2-----*/
    P2DIR = 0xFF;

```

```

P2OUT = 0x00;
/*-----*/

/*-----Porta 3-----*/
P3DIR = 0xFF;
P3OUT = 0x41;
/*-----*/

/*-----Porta 4-----*/
P4DIR = 0x00;
P4OUT = 0x00;
/*-----*/

/*-----Porta J-----*/
PJDIR = 0xFFFF;
PJOUT = 0x00;
/*-----*/

/*-----Selecao das Funcoes dos Pinos-----*/
P1SEL0 |= BIT2;
P1SEL1 &= ~BIT2;

P2SEL0 &= ~(BIT0 + BIT1 + BIT4 + BIT5);    /*Seleciona UCB0SIM0/UCB0SDA,
                                           // para comunicacao

P2SEL1 |= (BIT0 + BIT1 + BIT4 + BIT5);

P4SEL0 |= BIT2;                            // Seleciona o canal A10, para entrada do
P4SEL1 |= BIT2;                            // conversor A/D
/*-----*/

PM5CTL0 &= ~LOCKLPM5;    /* Faz com que as configuracoes de porta entrem
 * em vigor, do contrario, as portas permanecem
 * em alta impedancia, para evitar possiveis
 * danos ao controlador
 */
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DO TIMER A0=====*/
void Timer_A0(void)
{
    TA0CTL = TASSEL__SMCLK + MC__UP; /* Seleciona o clock SMCLK, dividido
 * por 1 e operando no modo UP
 */
    TA0CCTL1 = OUTMOD_3;            /* Configura para o modo comparacao e
 * formato set/reset
 */

    /*
 * - CALCULO PARA DETERMINAR O NÚMERO DE CONTAGENS NECESSÁRIA PARA
 * ATINGIR A FREQUÊNCIA DESEJADA, COM BASE NO VALOR DO CLOCK UTILIZADO
 *
 * 
$$T_{comutacao} = T_{clk} * (N+1)$$

 * 
$$1/(f_{comutacao}) = (1/f_{clk}) * (N+1)$$

 * 
$$1/(4*f_{led}) = (1/f_{clk}) * (N+1)$$

 * 
$$N = f_{clk}/(4*f_{led}) - 1$$

 * 
$$N = 8.000.000/(4*256) - 1$$

 * 
$$N = 8.000.000/1024 - 1$$

 * 
$$N = 7811$$

 *
 * ∴

```

```

*          f_comutacao = fclk/(N+1)
*          f_comutacao = 8.000.000/(7811+1)
*          f_comutacao = 1024,06554Hz
*
*          f_led = f_comutacao/4
*          f_led = 1024,06554Hz/4
*          f_led = 256,01639Hz
*/

TA0CCR0 = 7811;          /* -> Ajuste do sinal para uma
*                         * frequencia de 1024Hz, produzindo uma
*                         * amostragem de aproximadamente 256Hz
*                         * para cada LED.
*/
TA0CCR1 = TA0CCR0>>1;  // Razao ciclica dos LEDs em 50%
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DO TIMER A1=====*/
void Timer_A1(void)
{
    TA1CTL = TASSEL__SMCLK + MC__STOP + TACLR + ID_1;
                /* Selecciona o clock SMCLK, dividido
                * por 2 e inicialmente pausado
                */
    TA1CCTL1 = OUTMOD_7; // Selecciona o modo reset/set

    /*
    * - CALCULO PARA DETERMINAR O NÚMERO DE CONTAGENS NECESSÁRIA PARA
    * ATINGIR A FREQUÊNCIA DESEJADA, COM BASE NO VALOR DO CLOCK UTILIZADO
    *
    *          T_comutacao = Tclk*(N+1)
    *          1/(f_comutacao) = (1/fclk)*(N+1)
    *          1/(2.000) = (1/fclk)*(N+1)
    *          N = fclk/(2.000)-1
    *          N = 4.000.000/2.000-1
    *          N = 2.000
    *          N ~ 2.048 = 1<<11
    *
    *          ∴
    *          f_comutacao = fclk/(N+1)
    *          f_comutacao = 4.000.000/(2.048+1)
    *          f_comutacao = 1.952,17179 Hz
    */

    TA1CCR0 = 1<<11;

                /* Ajusta a frequencia do sinal de
                * acionamento do piezo, em aproximadamente
                * 1.952 Hz
                */
    TA1CCR1 = TA1CCR0>>1; // Define um razao ciclica de 50%
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DO TIMER A2=====*/
void Timer_A2(void)
{
    TA2CTL = TASSEL__SMCLK + ID_3 + MC__STOP;
                /* Selecciona o clock SMCLK, dividido
                * por 8 e inicialmente pausado

```

```

        * -> 1 MHz
        */
    TA2CCTL0 = CCIE; /* Habilita a interrupcao por captura/
        * comparacao
        */
    TA2EX0 = TAIDEX_7; /* Divide o sinal de clock por 8
        * -> 125 kHz
        */

    TA2CCR0 = 65535; /* -> Ajuste do sinal para um periodo
        * maximo de 524,288 ms
        */
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DO A/D-12bit=====*/
void Conversor_AD12bit(void)
{
    ADC12CTL0 = ADC12SHT1_6 | ADC12ON; /* Ajuste do tempo de retencao e
        * amostragem para 16 ciclos
        */
    ADC12CTL1 = ADC12SHP | ADC12SHS_1 | ADC12CONSEQ_2;
        /* Configura o diparo pelo TA0CCR1,
        * com o sinal de clock de MODOSC e
        * um canal de amostragem com apenas
        * uma conversao
        */
    ADC12CTL2 = ADC12RES_2; // Seleciona a resolucao de 12-bit
    ADC12MCTL0 = ADC12INCH_10 | ADC12EOS; /* Seleciona as referencias AVCC e
        * AVSS com o canal de entrada A10,
        * para amostragem no pino 2 da
        * porta 4
        */
    ADC12IER0 |= ADC12IE0; /* Habilita a interrupcao de
        * ADC12IFG
        */

    ADC12CTL0 |= ADC12ENC | ADC12SC; /* Liga o conversor e habilita a
        * conversao
        */
}
/*=====*/

/*=====CONFIGURACAO/INICIALIZACAO DA SPI=====*/
void Comunicacao_SPI(void)
{
    UCA1CTLW0 = UCSWRST; /* Manutenção lógica de eUSCI no estado de
        * reinicialização
        */

    UCA1CTLW0 = UCCKPH + UCMSB + UCMST + UCSYNC + UCSSEL_2 + UCSWRST;
        /* Comunicacao SPI de 8bit, modo mestre com 3 pinos,
        * de forma assincrona e clock SMCLK
        */
    UCA1BR0 = 16; /* Divide a fonte de clock de 8MHz por 16,
        * resultando em um sinal de 500kHz, para fornecer
        * ao escravo (display)
        */

    UCA1BR1 = 0;
}

```

```

        UCA1CTLW0 &= ~UCSWRST;    // Reseta eUSCI liberando-o para operacao
    }
    /*=====*/

    /*=====CONFIGURACAO/INICIALIZACAO DA UART=====*/
    void Comunicacao_UART(void)
    {
        UCA0CTLW0 = UCSWRST;        /* Manutenção lógica de eUSCI no estado de
                                     * reinicialização
                                     */

        UCA0CTLW0 = UCSSEL_2;
        UCA0BR0 = 4;
        UCA0BR1 = 0;

        UCA0MCTLW |= UCOS16 | UCBRF_5;

        UCA1CTLW0 &= ~UCSWRST;    // Reseta eUSCI liberando-o para operacao
    }
    /*=====*/

    /*=====INTERRUPCOES=====*/
    /*-----Rotina de Tratamento da Interrupcao do A/D de 12-bit-----*/
    #pragma vector = ADC12_VECTOR
    __interrupt void Conversor_AD12_RTI(void)
    {
        controle_LEDs++;    // Comuta o acionamento

        valor = ADC12MEM0;

        switch(controle_LEDs)
        {
            case 1:
                P3OUT = INFRAVERMELHO;
                break;
            case 2:
                P3OUT = NENHUM;

                /*- - - - - Filtragem do Sinal Infravermelho- - - - - */
                filtrar(&sinal_filtrado_IV, &filtragem_IV[fil_IV], &fil_IV);
                /*- - - - - */

                if (MODO_ENGENHARIA == 2)
                {
                    ajuste_2++;
                    if (ajuste_2 > 3)
                    {
                        if (sinal_filtrado_IV < 0)
                        {
                            forma_onda_IV[varre_onda] = 0;
                        }
                        else if (sinal_filtrado_IV < 4095)
                        {
                            forma_onda_IV[varre_onda] =
sinal_filtrado_IV;
                        }
                    }
                }
            else
            {

```

```

        forma_onda_IV[varre_onda] = 4095;
    }
}

passo_amostra++;

/*- - -Deteccao de Pico ou Vale do Sinal Infravermelho- - - */
if (flag_pico_ou_vale_IV == 1)
{
    detectar_pico(&sinal_filtrado_IV, &pico_IV,
&flag_pico_ou_vale_IV, &escada_pico_IV, &passo_amostra,
&flag_calcular_denominador);

    if(flag_calcular_denominador > 0)
    {
        G_max += pico_IV;
        G_max = G_max>>1;

        R_denominador = pico_IV;
        pico_IV = 0; // Limpa a variavel que armazena o
valor de pico

////////////////////////////////////
if(TA2R == 0 && periodo_tempo == 0)
{
    TA2CTL |= MC_UP;
    TA2R = 0;
}
else if(TA2R < 34090 && periodo_tempo == 0) //
Condicao para eliminar amostras de BPM maiores do que 220bpm
{
    TA2R = 0;
    periodo_tempo = 0;
}
else if(periodo_tempo >= 3) // Condicao para
eliminar amostras de BPM menores que 38bpm
{
    TA2R = 0;
    periodo_tempo = 0;
}
else
{
    BPM += 75000000/(TA2R +
periodo_tempo*65536);

    BPM = BPM>>1;
    periodo_tempo = 0;
    TA2R = 0;

    TA1CTL |= MC_UP;
    bpm_som = 0;
}
////////////////////////////////////
}
}
else
{

```

```

        detectar_vale(&sinal_filtrado_IV, &vale_IV,
&flag_pico_ou_vale_IV, &escada_vale_IV, &flag_calcular_denominador);

        if(flag_calcular_denominador == 0)
        {
            G_min += vale_IV;
            G_min = G_min>>1;

            R_denominador = log(vale_IV/R_denominador);
            vale_IV = 4095;          /* Limpa a variavel que
armazena o valor                                * de pico
                                                */
        }
    }
/* ----- */

/*----- Plotagem da Forma de Onda -----*/
ajuste_janela++;

if(ajuste_janela > 3)// A onda he reamostrada a cada 4 amostra
{
    tempo++;
    ajuste_janela = 0;

    forma_onda[tempo+1] = sinal_filtrado_IV*G-OFF_SET;

    if (MODO_ENGENHARIA == 1)
    {
        forma_onda_V[tempo+1] = sinal_filtrado_V*G_V-
OFF_SET_V;

        if(forma_onda_V[tempo+1] <= 0)
        {
            forma_onda_V[tempo+1] = 0;
        }
        else if(forma_onda_V[tempo+1] > 63)
        {
            forma_onda_V[tempo+1] = 63;
        }
    }

    if(forma_onda[tempo+1] <= 0)
    {
        forma_onda[tempo+1] = 0;
    }
    else if(forma_onda[tempo+1] > 63)
    {
        forma_onda[tempo+1] = 63;
    }

    /*-----Varredura dos dados da curva a serem desenhados
- - -*/

    varre_coluna++;
    if(varre_coluna > 16)
    {
        varre_coluna = 1;
        varre_bloco++;
    }
}

```

```

        if(varre_bloco > 5)
        {
            varre_bloco = 0;
        }
    }
    /*----- */
    /*- -Varredura dos dados da curva a serem apagados- -*/
    apagar++;
    varre_coluna_apagar++;
    if(varre_coluna_apagar >= 16)
    {
        varre_coluna_apagar = 0;
        varre_bloco_apagar++;
        if(varre_bloco_apagar > 5)
        {
            varre_bloco_apagar = 0;
        }
    }
    /*----- */

    apaga_grafico(&(forma_onda[apagar+1]),
&varredura[63+128*varre_bloco_apagar-forma_onda[apagar+1]],
&varredura[127+128*varre_bloco_apagar-forma_onda[apagar+1]]);

    desenha_grafico(&(forma_onda[tempo+1]),
&varredura[63+128*varre_bloco-forma_onda[tempo+1]],
&varredura[127+128*varre_bloco-forma_onda[tempo+1]]);

    if(tempo >= 96)
    {
        tempo = 0;
        G = G_max*0.0083;
        G = G/(G_max-G_min);

        OFF_SET = G_min*G*0.975;

        num_cen_bpm = BPM/1000;
        num_dez_bpm = BPM/100 - num_cen_bpm*10;
        num_uni_bpm = BPM/10 - (num_cen_bpm*100 +
num_dez_bpm*10);

        desenha_bpm(&num_uni_bpm, &num_dez_bpm,
&num_cen_bpm);

        // Limita os valores a serem mostrados no Display
entre 70% ate 99%

        if(700 <= SpO2 && SpO2 <= 999)
        {
            num_dez_sat = SpO2/100;
            num_uni_sat = SpO2/10 - num_dez_sat*10;

            desenha_saturacao(num_uni_sat,
num_dez_sat);

            intensidade_som = (101-SpO2/10)<<2;
        }

        if (MODO_ENGENHARIA == 1)

```

```

{
    G_V = G_max_V*0.0083;
    G_V = G_V/(G_max_V-G_min_V);

    OFF_SET_V = G_min_V*G_V*0.975;

    /*- - Envio dos dados do Infravermelho- -*/
    forma_onda[0] = 3;
    forma_onda[1] = OFF_SET_V>>4;
    forma_onda[98] = BPM/10;
    forma_onda[99] = SpO2/10;
    forma_onda[100] = G_max_V>>4;
    forma_onda[101] = G_min_V>>4;

    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = '0';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = 'I';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = 'V';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = ' ';

    for (posicao = 0; posicao <= 101;
posicao++)
    {
        Transmissao[0] =
        forma_onda[posicao]/100;
        Transmissao[1] =
        forma_onda[posicao]/10 - 10*Transmissao[0];
        Transmissao[2] = forma_onda[posicao]
        - (100*Transmissao[0] + 10*Transmissao[1]);

        while(!(UCA0IFG&UCTXIFG));
        UCA0TXBUF = ' ';
        // Transmite espacamento
        while(!(UCA0IFG&UCTXIFG));
        UCA0TXBUF = (Transmissao[0] + 48);
        // Transmite centena
        while(!(UCA0IFG&UCTXIFG));
        UCA0TXBUF = (Transmissao[1] + 48);
        // Transmite Dezena
        while(!(UCA0IFG&UCTXIFG));
        UCA0TXBUF = (Transmissao[2] + 48);
        // Transmite Unidade
    }
    /* - - - - - */

    /*- - Envio dos dados do Vermelho - - - */
    forma_onda_V[0] = 3;
    forma_onda_V[1] = OFF_SET_V>>4;
    forma_onda_V[98] = BPM/10;
    forma_onda_V[99] = SpO2/10;
    forma_onda_V[100] = G_max_V>>4;
    forma_onda_V[101] = G_min_V>>4;

    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = '0';
    while(!(UCA0IFG&UCTXIFG));

```

```

UCA0TXBUF = 'V';
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = 'E';
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = ' ';

for (posicao = 0; posicao <= 101;
posicao++)
{
    Transmissao[0] =
forma_onda_V[posicao]/100;
    Transmissao[1] =
forma_onda_V[posicao]/10 - 10*Transmissao[0];
    Transmissao[2] =
forma_onda_V[posicao] - (100*Transmissao[0] + 10*Transmissao[1]);

    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = ' ';
    // Transmite espacamento
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = (Transmissao[0] + 48);
    // Transmite centena
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = (Transmissao[1] + 48);
    // Transmite Dezena
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = (Transmissao[2] + 48);
    // Transmite Unidade
}
/* - - - - - */
}
else if (MODO_ENGENHARIA == 2)
{
    /*- - Envio dos dados do Infravermelho- -*/
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = '0';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = 'I';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = 'V';
    while(!(UCA0IFG&UCTXIFG));
    UCA0TXBUF = '2';

    for (posicao = 0; posicao <= 95; posicao++)
    {
        Transmissao[0] =
forma_onda_IV[posicao]/1000;
        Transmissao[1] =
forma_onda_IV[posicao]/100 - 10*Transmissao[0];
        Transmissao[2] =
forma_onda_IV[posicao]/10 - (100*Transmissao[0] + 10*Transmissao[1]);
        Transmissao[3] =
forma_onda_IV[posicao] - (1000*Transmissao[0] + 100*Transmissao[1] +
10*Transmissao[2]);

        while(!(UCA0IFG&UCTXIFG));
        UCA0TXBUF = Transmissao[0] + 48;
        // Transmite milhar
        while(!(UCA0IFG&UCTXIFG));

```

```

UCA0TXBUF = (Transmissao[1] + 48);
// Transmite centena
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = (Transmissao[2] + 48);
// Transmite Dezena
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = (Transmissao[3] + 48);
// Transmite Unidade
}
/* - - - - - */

/*- - - Envio dos dados do Vermelho- - - */
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = '0';
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = 'V';
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = 'E';
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = '2';

for (posicao = 0; posicao <= 95; posicao++)
{
    Transmissao[0] =
forma_onda_V_II[posicao]/1000;
    Transmissao[1] =
forma_onda_V_II[posicao]/100 - 10*Transmissao[0];
    Transmissao[2] =
forma_onda_V_II[posicao]/10 - (100*Transmissao[0] + 10*Transmissao[1]);
    Transmissao[3] =
forma_onda_V_II[posicao] - (1000*Transmissao[0] + 100*Transmissao[1] +
10*Transmissao[2]);

    while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = Transmissao[0] + 48;
// Transmite milhar
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = (Transmissao[1] + 48);
// Transmite centena
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = (Transmissao[2] + 48);
// Transmite Dezena
while(!(UCA0IFG&UCTXIFG));
UCA0TXBUF = (Transmissao[3] + 48);
// Transmite Unidade
}
/* - - - - - */
}
}

if(apagar >= 96)
{
    apagar = 0;
}
}

bpm_som++;

if (bpm_som > intensidade_som)

```

```

    {
        TA1CTL = TASSEL__SMCLK + MC__STOP + TACLK;
        P1OUT = 0x00;
    }
    /* ----- */

    break;
case 3:
    P3OUT = VERMELHO;
    break;
case 4:
    P3OUT = NENHUM;

    /* ----- Filtragem do Sinal Vermelho ----- */
    filtrar(&sinal_filtrado_V, &filtragem_V[fil_V], &fil_V);
    /* ----- */

    if (MODO_ENGENHARIA == 2)
    {
        if(ajuste_2 > 3)
        {
            ajuste_2 = 0;

            if (sinal_filtrado_V < 0)
            {
                forma_onda_V_II[varre_onda] = 0;
            }
            else if (sinal_filtrado_V < 4095)
            {
                forma_onda_V_II[varre_onda] =
sinal_filtrado_V;
            }
            else
            {
                forma_onda_V_II[varre_onda] = 4095;
            }
            varre_onda++;

            if (varre_onda > 95)
            {
                varre_onda = 0;
            }
        }
    }

    /* ----- Deteccao de Pico ou Vale do Sinal Vermelho ----- */
    if (flag_pico_ou_vale_V == 1)
    {
        detectar_pico(&sinal_filtrado_V, &pico_V,
&flag_pico_ou_vale_V, &escada_pico_V, &passo_amostra, &flag_calcular_numerador);

        if(flag_calcular_numerador > 0)
        {
            if( MODO_ENGENHARIA == 1)
            {
                G_max_V += pico_V;
                G_max_V = G_max_V>>1;
            }
            R_numerador = pico_V;
        }
    }
}

```

```

        pico_V = 0; // Limpa a variavel que armazena o
valor de pico
    }
    }
    else
    {
        detectar_vale(&sinal_filtrado_V, &vale_V,
&flag_pico_ou_vale_V, &escada_vale_V, &flag_calcular_numerador);

        if(flag_calcular_numerador == 0)
        {
            if( MODO_ENGENHARIA == 1)
            {
                G_min_V += vale_V;
                G_min_V = G_min_V>>1;
            }

            R = log(vale_V/R_numerador)/R_denominador;

            SpO2 += 100.002*R*R*R-528.87*R*R+268.71*R+982.83;
                /* Calcula 10*SpO2 e incrementa o valor
                * em SpO2
                */
            SpO2 = SpO2>>1; /* Calcula a media entre o valor
                * anterior e o novo valor calculado
                */

            vale_V = 4095; /* Limpa a variavel que armazena o
valor
                * de pico
                */
        }
    }
    /* ----- */

    controle_LEDs = 0;
    break;
}
}
/*-----*/

/*-----Rotina de Tratamento do TIMER A1-----*/
#pragma vector = TIMER2_A0_VECTOR
__interrupt void Interrupcao_TIMER2_A0(void)
{
    periodo_tempo++;
    if(periodo_tempo == 15)
    {
        periodo_tempo = 0;
        TA2CTL |= MC_STOP;
        TA2R = 0;

        apresenta_tela_inicial();

        P3OUT = NENHUM;

        __bis_SR_register(LPM4); // Seta o bit GIE
    }
}

```

```

/*-----*/
/*=====*/
/*=====FUNCOES=====*/
/*-----Filtro Digital (Media Movel)-----*/
void filtrar (short int *sinal_filtrado, short int *filtragem, unsigned char *fil)
{
    // Filtragem por media movel com M = 32 (Ordem 32)

    *sinal_filtrado += (4095-ADC12MEM0)>>5;
    *sinal_filtrado -= (*filtragem)>>5;
    *filtragem = (4095-ADC12MEM0);          /* Armazena os termos utilizados no
                                           * filtro
                                           */
    (*fil)++;          // Desloca a posicao de alocao dos termos

    if(*fil >= 32)    // Identifica quando o vetor foi totalmente preenchido
    {
        *fil = 0;    // Reinicia a posicao de alocao dos termos
    }
}
/*-----*/
/*-----Detector de Pico-----*/

void detectar_pico (short int *sinal, short int *pico, unsigned char *flag,
unsigned char *escada_pico, unsigned char *passo, unsigned char *flag_calcular)
{
    if (*sinal >= *pico)// Analise se o valor amostrado eh o maior que o atual
valor de pico
    {
        *pico = *sinal;    // Atribui o valor atual ao valor do pico
        *escada_pico = 0;  // Reseta a escada de confirmacao de passagem por
pico
    }
    else
    {
        (*escada_pico)++;    // Incrementa a variavel
        if (*escada_pico > 50) // Analisa se confirmacao por passagem de
pico ocorreu
        {
            (*flag_calcular)++;
            *escada_pico = 0; // Reseta a escada de confirmacao de
passagem por pico

            *flag = 0;      // Aciona o algoritmo para detectar vales
        }
    }
}
/*-----*/
/*-----Detector de Vale-----*/
void detectar_vale (short int *sinal, short int *vale, unsigned char *flag,
unsigned char *escada_vale, unsigned char *flag_calcular)
{
    if (*sinal <= *vale)    // Analise se o valor amostrado eh o maior que o
atual valor de pico
    {

```

```
        *vale = *sinal;    // Atribui o valor atual ao valor do pico
        *escada_vale = 0; // Reseta a escada de confirmacao de passagem por
pico
    }
    else
    {
        (*escada_vale)++; // Incrementa a variavel
        if (*escada_vale > 50) // Analisa se confirmacao por passagem de
pico ocorreu
        {
            *flag_calcular = 0;
            *escada_vale = 0; // Reseta a escada de confirmacao de
passagem por pico

            (*flag)++; // Aciona o algoritmo para detectar picos
        }
    }
}
/*-----*/
/*=====*/
```





```

// Fim do Terceiro Bloco
// Inicio do Quarto Bloco
// Fim do Quarto Bloco
// Inicio do Quinto Bloco

```

0xC0, 0x0F,  
0xC0, 0x0F,  
0xFF, 0xFF,  
0x00, 0x00,  
0x00, 0x00,  
0x00, 0x00,  
0x00, 0x00,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xE0, 0x00,  
0xE2, 0xAA,  
0xE5, 0x55,  
0xE2, 0xAA,  
0xE5, 0x55,  
0xE2, 0xAA,  
0xE5, 0x55,  
0xE2, 0xA8,  
0xE5, 0x53,  
0xE2, 0xAB,  
0xE5, 0x53,  
0xFF, 0xFF,  
0x00, 0x00,  
0x00, 0x00,  
0x00, 0x00,  
0x00, 0x00,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0xFF, 0xFF,  
0x00, 0x00,  
0xAA, 0xAA,  
0x55, 0x55,  
0xAA, 0xAA,  
0x55, 0x55,  
0xAA, 0xAA,  
0x55, 0x55,  
0x00, 0x00,  
0xFF, 0xFF,  
0xFF, 0xFF,































```

do Decimo Quinto Bloco
    0x1F, 0x31,
    0x03, 0x31,
    0x63, 0x31,
    0x7F, 0x3F,
    0x3E, 0x3F,
    0x00, 0x30,
    0x00, 0x30,
    0x00, 0x30,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00, // Fim

Inicio do Decimo Sexto Bloco
    0x00, 0x00, //
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0xE3,
    0x00, 0xA6,
    0x00, 0xEC,
    0x00, 0x18,
    0x00, 0x37,
    0x00, 0x65,
    0x00, 0xC7,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x0F, 0x80,
    0x1F, 0xC0,
    0x18, 0xC0,
    0x30, 0x60,
    0xB0, 0x66,
    0xB0, 0x69,
    0xB0, 0x61,
    0x98, 0xC6,
    0x9F, 0xC8,
    0x0F, 0x8F,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00,
    0x00, 0x00, // Fim

do Decimo Sexto Bloco
};

unsigned int instrucao = 0;
unsigned int dado = 0;
unsigned char n = 0;
int i = 0;

unsigned int x = 0;
unsigned int y = 0;

unsigned char passo = 0;

```

```

unsigned char deslocamento = 0;

unsigned char passo_apagar = 0;
unsigned char deslocamento_apagar = 1;

void envia_instrucao_24_spi(unsigned int instrucao)
{
    P2OUT ^= CS;

    UCA1TXBUF = 0xF8; // Envia 8 bits (1 1 1 1 1 W 0 0)
    while(UCA1STATW & UCBUSY);
    UCA1TXBUF = (instrucao & 0xF0) ; // Envia 4 bits mais significativos +
0x0000b
    while(UCA1STATW & UCBUSY);
    UCA1TXBUF = (instrucao & 0x0F)<<4; // Envia os 4 bits menos
significativos + 0x0000b
    while(UCA1STATW & UCBUSY);

    P2OUT ^= CS;
}

void envia_dado_24_spi(unsigned int dado)
{
    P2OUT ^= CS;

    UCA1TXBUF = 0xFA; // Envia 8 bits (1 1 1 1 1 W RS 0)
    while(UCA1STATW & UCBUSY);
    UCA1TXBUF = (dado & 0xF0) ; // Envia os 4 bits mais
significativos + 0x0000b
    while(UCA1STATW & UCBUSY);
    UCA1TXBUF = (dado & 0x0F)<<4; // Envia os 4 bits menos
significativos + 0x0000b
    while(UCA1STATW & UCBUSY);

    P2OUT ^= CS;
}

void reseta_display(void)
{
    __delay_cycles(50*ms); // Tempo de espera de 50ms
    P2OUT ^= RESET; // Seta o display
}

void seta_registrador_instrucoes(unsigned int instrucao)
{
    envia_instrucao_24_spi(instrucao); // Envia a instrucao via SPI
    __delay_cycles(1*ms); // Tempo de espera de 1ms
}

void seta_registrador_endereco(unsigned int endereco_y, unsigned int endereco_x)
{
    envia_instrucao_24_spi(endereco_y);
    __delay_cycles(100*us);

    envia_instrucao_24_spi(endereco_x);
    __delay_cycles(100*us); // Tempo de espera de 100us
}

```

```

void desenha_tela(unsigned int endereco_y, unsigned int endereco_x, unsigned int
byte_mais_sig, unsigned int byte_menos_sig)
{
    seta_registrador_endereco(endereco_y, endereco_x);

    // Escrita
    envia_dado_24_spi(byte_mais_sig);
    __delay_cycles(100*us);           // Tempo de espera de 100us

    envia_dado_24_spi(byte_menos_sig);
    __delay_cycles(100*us);           // Tempo de espera de 100us
}

void limpa_display(unsigned int instrucao)
{
    envia_instrucao_24_spi(instrucao);
    __delay_cycles(10*ms);           // Tempo de espera de 10ms
}

void desenha_tela_inicial(void)
{
    // - - - - - Toda Tela - - - - - //
    y = 0x80;
    x = 0x80;
    i = 0;
    do
    {
        desenha_tela(y, x, tela_inicial[i], tela_inicial[i+1]);

        y++;
        i+=2;

        if (y >= 160)
        {
            y = 128;
            x++;
        }
    }while(x<=143);
    // - - - - - //
}

void desenha_tela_operacao(void)
{
    // - - - - - Toda Tela - - - - - //
    y = 0x80;
    x = 0x80;
    i = 0;
    do
    {
        desenha_tela(y, x, tela_operacao[i], tela_operacao[i+1]);

        i+=2;
        y++;

        if (y >= 160)
        {
            y = 128;
            x++;
        }
    }
}

```

```

        }while(x<=143);
    }

    void desenha_bpm(unsigned char *num_unidade_bpm, unsigned char *num_dezena_bpm,
    unsigned char *num_centena_bpm)
    {
        // ----- -bpm ----- //
        n = 0;
        y = 0x83;
        x = 0x86;
        do
        {
            desenha_tela(y, x, (0x00 | tela_operacao[390+2*n]),
            (numeros[n+*num_centena_bpm*7] | tela_operacao[390+2*n])); // Centena
            y++;
            n++;
        }while(y<=0x89);

        y = 0x83;
        x = 0x87;
        n = 0;

        do
        {
            desenha_tela(y, x, (numeros[n+*num_dezena_bpm*7] |
            tela_operacao[454+2*n]), (numeros[n+*num_unidade_bpm*7] |
            tela_operacao[455+2*n])); // Dezena e Unidade
            y++;
            n++;
        }while(y<=0x89);
        // ----- -//
    }

    void desenha_saturacao(unsigned int num_unidade_saturacao, unsigned int
    num_dezena_saturacao)
    {
        // ----- Sp02 ----- //
        n = 0;
        y = 0x83;
        x = 0x8E;
        do
        {
            desenha_tela(y, x, (0x00 | tela_operacao[902+2*n]),
            (numeros[n+num_dezena_saturacao*7] | tela_operacao[903+2*n])); // Dezena
            y++;
            n++;
        }while(y<=0x89);

        y = 0x83;
        x = 0x8F;
        n = 0;

        do
        {
            desenha_tela(y, x, (numeros[n+num_unidade_saturacao*7] |
            tela_operacao[966+2*n]), (0x00 | tela_operacao[967+2*n])); // Unidade
            y++;
            n++;
        }while(y<=0x89);
    }

```

```

// - - - - - //
}

void desenha_grafico(unsigned char *amplitude, unsigned char
*varredura_mais_significativo, unsigned char *varredura_menos_significativo)
{
    // - - - - - Grafico - - - - - //

    // Escrita
    deslocamento++;

    if(deslocamento > 16)
    {
        passo++;
        deslocamento = 1;
        if(passo >= 6)
        {
            passo = 0;
        }
    }

    if(*amplitude >= 32)
    {
        *varredura_mais_significativo |= (0x80>>(deslocamento-1));
        *varredura_menos_significativo |= (0x01<<(16-deslocamento));

        desenha_tela( (191 - *amplitude), (128 + passo),
*varredura_mais_significativo, *varredura_menos_significativo);
    }
    else if((*amplitude) > 0)
    {
        *varredura_mais_significativo |= (0x80>>(deslocamento-1));
        *varredura_menos_significativo |= (0x01<<(16-deslocamento));

        desenha_tela( (159 - *amplitude), (136 + passo),
*varredura_mais_significativo, *varredura_menos_significativo);
    } // - - - - - //
}

void apaga_grafico(unsigned char *limpa, unsigned char *limpa_mais_significativo,
unsigned char *limpa_menos_significativo)
{
    // - - - - - Grafico - - - - - //
    // Escrita
    if(deslocamento_apagar >= 16)
    {
        passo_apagar++;
        deslocamento_apagar = 0;
        if(passo_apagar >= 6)
        {
            passo_apagar = 0;
        }
    }

    deslocamento_apagar++;

    if(*limpa >= 32)
    {
        // Apaga ponto da curva
        *limpa_mais_significativo &= ~(0x80>>(deslocamento_apagar-1));
    }
}

```

```

        *limpa_mais_significativo |= (tela_operacao[passo_apagar*64+126-
2*(*limpa)]);
        *limpa_menos_significativo &= ~(0x01<<(16-deslocamento_apagar));
        *limpa_menos_significativo |= (tela_operacao[passo_apagar*64+127-
2*(*limpa)]);

        desenha_tela( (191 - *limpa), (128 + passo_apagar),
*limpa_mais_significativo, *limpa_menos_significativo);
    }
    else if((*limpa) > 0)
    {
        // Apaga ponto da curva
        *limpa_mais_significativo &= ~(0x80>>(deslocamento_apagar-1));
        *limpa_mais_significativo |= (tela_operacao[passo_apagar*64+574-
2*(*limpa)]);
        *limpa_menos_significativo &= ~(0x01<<(16-deslocamento_apagar));
        *limpa_menos_significativo |= (tela_operacao[passo_apagar*64+575-
2*(*limpa)]);

        desenha_tela( (159 - *limpa), (136 + passo_apagar),
*limpa_mais_significativo, *limpa_menos_significativo);
    }
    //----- //
}

```

## APÊNDICE D - "BIBLIOTECA\_LCD.H"

```
ifndef BIBLIOTECA_LCD_H_
#define BIBLIOTECA_LCD_H_

    void reseta_display(void);
    void limpa_display(unsigned int instrucao);
    void desenha_bpm(unsigned char *num_unidade_bpm, unsigned char
*num_dezena_bpm, unsigned char *num_centena_bpm);
    void desenha_saturacao(unsigned int num_unidade_saturacao, unsigned int
int num_dezena_saturacao);
    void desenha_grafico(unsigned char *amplitude, unsigned char
*varredura_mais_significativo, unsigned char *varredura_menos_significativo);
    void apaga_grafico(unsigned char *limpa, unsigned char
*limpa_mais_significativo, unsigned char *limpa_menos_significativo);
    void desenha_tela_inicial(void);
    void desenha_tela_operacao(void);
    void seta_registrador_endereco(unsigned int endereco_y, unsigned int
endereco_x);
    void seta_registrador_instrucoes(unsigned int instrucao);

#endif
```

## APÊNDICE E - ARQUIVO DE INICIALIZAÇÃO DO DISPLAY “INICIALIZA\_DISPLAY.C”

```

#include <msp430fr5969.h>
#include "inicializa_display.h"
#include "biblioteca_lcd.h"

int grafico_1 = 0x80;
int grafico_2 = 0x80;

void ini_display_128x64(void)
{
//-----Inicializacao-----//
    reseta_display();           // Reseta o display

    seta_registrador_instrucoes(0x30); // Configura o conjunto de funcoes
(Somente para DL)

    seta_registrador_instrucoes(0x34); // Configura o conjunto de funcoes
(Agora para RE)

    seta_registrador_instrucoes(0x08); // Configura o controle ON/OFF do
display

    limpa_display(0x01);        // Limpa display

    seta_registrador_instrucoes(0x07); // Ajuste no modo de entrada
//-----FIM-----//
    seta_registrador_instrucoes(0x36); // Configura o conjunto de funcoes
(Agora para G)

    seta_registrador_instrucoes(0x02); // Rolagem vertical

    seta_registrador_endereco(0x80, 0x80); // Conjunto de enderecos da CGRAM
(Vertical)
}

void apresenta_tela_inicial(void)
{
    desenha_tela_inicial();
}

void apresenta_tela_operacao(void)
{
    desenha_tela_operacao();
}

```

## APÊNDICE F - "INICIALIZA\_DISPLAY.H"

```
#ifndef INICIALIZA_DISPLAY_H_
#define INICIALIZA_DISPLAY_H_

    void ini_display_128x64(void);
    void apresenta_tela_inicial(void);
    void apresenta_tela_operacao(void);
    void desenha_tela_inicial(void);
    void desenha_tela_operacao(void);

#endif
```

## APÊNDICE G - BIBLIOTECA DESENVOLVIDA PARA COMPOR A ESTRUTURA DOS NÚMEROS “NUMEROS.H”

```

#ifndef NUMEROS_H_
#define NUMEROS_H_

    const unsigned char numeros[84] = {

        0xFE, // Zero
        0xFE,
        0x86,
        0x86,
        0x86,
        0xFE,
        0xFE, //
        0x0C, // Um
        0x1C,
        0x3C,
        0x0C,
        0x0C,
        0x0C,
        0x0C, //
        0x7C, // Dois
        0xFE,
        0xC6,
        0x06,
        0x38,
        0x70,
        0xFE, //
        0xFE, // Três
        0x06,
        0x06,
        0x1C,
        0x06,
        0x06,
        0xFE, //
        0x8C, // Quatro
        0x8C,
        0x8C,
        0xFC,
        0x0C,
        0x0C,
        0x0C, //
        0x7E, // Cinco
        0x40,
        0x40,
        0x7E,
        0x06,
        0x06,
        0xFE, //
        0xFE, // Seis
        0x80,
        0x80,
        0xFE,
        0x86,
        0x86,
        0xFE, //
    }

```

```
0xFE, // Sete
0xFE,
0x06,
0x0C,
0x18,
0x30,
0x60, //
0xFE, // Oito
0x86,
0x86,
0xFE,
0x86,
0x86,
0xFE, //
0xFE, // Nove
0x86,
0x86,
0xFE,
0x06,
0x06,
0x7E,
0xFE, // E
0x80,
0x80,
0xFC,
0x80,
0x80,
0xFE, //
0x7E, // G
0x80,
0x80,
0x9E,
0x82,
0x82,
0x7E, //
};
```

```
#endif /* NUMEROS_H_ */
```

## APÊNDICE H - CÓDIGO IMPLEMENTADO NO MATLAB® PARA AQUISIÇÃO E PROCESSAMENTO DOS DADOS ENVIADOS PELO MICROCONTROLADOR

```

delete(instrfindall);
close all;
clear all;
clc;

s = serial('COM6'); % Seleciona a porta COM6

set(s, 'InputBufferSize', 4); % Ajusta o buffer para 4
set(s, 'FlowControl', 'hardware');
set(s, 'BaudRate', 115200);
set(s, 'Parity', 'none');
set(s, 'DataBits', 8);
set(s, 'StopBit', 1);
clc;

disp(get(s, 'Name'));
prop(1)=(get(s, 'BaudRate'));
prop(2)=(get(s, 'DataBits'));
prop(3)=(get(s, 'StopBit'));
prop(4)=(get(s, 'InputBufferSize'));

disp(['Port Setup Done!!', num2str(prop)]);

fopen(s); % Abre a porta serial
t=1;
tempo = 1;
tempo_v = 1;
w=1;
disp('Running');
x=0;
inicia_IV = 0;
inicia_V = 0;
comuta = 0;
modo = 0;

while(t < 103)
    a =fgetl(s);

    if comuta == 0
        if inicia_IV == 0
            if char(a) == 'OIV '
                inicia_IV = 1;
                modo == 1
            elseif char(a) == 'OIV2'
                inicia_IV = 2;
            end
        elseif inicia_IV == 1
            if t == 1
                ajuste_de_passo(w) = str2num(a);
            elseif t == 2
                OFF_SET(w) = 16*str2num(a);
            end
        end
    end
end

```

```

elseif t == 99
    BPM(w) = str2num(a);
elseif t == 100
    SpO2(w) = str2num(a);
elseif t == 101
    G_max(w) = 16*str2num(a);
elseif t == 102
    G_min(w) = 16*str2num(a);
    inicia_IV = 0;
    comuta = 1;
else
    x(tempo) = (1/256.01639)*4*(tempo);
    y(tempo) = str2num(a);
    tempo = tempo + 1;
end
t=t+1;
if t == 103
    t = 1;
end
a=0; % Limpa o buffer
elseif inicia_IV == 2
    x(tempo) = (1/256.01639)*4*(tempo);
    y(tempo) = (2.5/4096)*str2num(a);
    tempo = tempo + 1;
    t=t+1;
    if t == 97
        inicia_IV = 0;
        comuta = 1;
        t = 1;
    end
    a=0; % Limpa o buffer
end
else
    if inicia_V == 0
        if char(a) == 'OVE '
            inicia_V = 1;
        elseif char(a) == 'OVE2'
            inicia_V = 2;
        end
    elseif inicia_V == 1
        if t == 1
            ajuste_de_passo_V(w) = str2num(a);
        elseif t == 2
            OFF_SET_V(w) = 16*str2num(a);
        elseif t == 99
            BPM_V(w) = str2num(a);
        elseif t == 100
            SpO2_V(w) = str2num(a);
        elseif t == 101
            G_max_V(w) = 16*str2num(a);
        elseif t == 102
            G_min_V(w) = 16*str2num(a);
            inicia_V = 0;
            comuta = 0;
        else
            x_V(tempo_V) = (1/256.01639)*4*(tempo_V);
            y_V(tempo_V) = str2num(a);
        end
    end
end

```

```

        tempo_V = tempo_V + 1;
    end
    t=t+1;
    if t == 103
        t = 1;
        w = w + 1;
        if w == 5
            t = 104;
        end
    end
    end
    a=0; % Limpa o buffer
elseif inicia_V == 2
    x_V(tempo_V) = (1/256.01639)*4*(tempo_V);
    y_V(tempo_V) = (2.5/4096)*str2num(a);
    tempo_V = tempo_V + 1;
    t=t+1;
    if t == 97
        t = 1;
        inicia_V = 0;
        comuta = 0;
        w = w + 1;
        if w == 5
            t = 104;
        end
    end
    end
    a=0; % Limpa o buffer
end
end
end

fclose(s); % Fecha a porta serial

figure(1);
plot(x,y,'b',x_V,y_V,'r');
grid on;

if modo == 1
    BPM_med = BPM(1);

    for t = 2:1: numel(BPM)
        BPM_med = (BPM_med + BPM(t))/2;
    end

    c1c
    BPM_med

    SpO2_med = SpO2(1);

    for t = 2:1: numel(SpO2)
        SpO2_med = (SpO2_med + SpO2(t))/2;
    end
    end
    SpO2_med
end
end

```