

**Mateus Sousa Araújo**

**Requisitos Obrigatórios e Optativos**  
**Proposta de um Sistema Embarcado para**  
**auxílio na reabilitação de pacientes com AVE**

Fortaleza/CE

2021

**Mateus Sousa Araújo**

**Proposta de um Sistema Embarcado para auxílio na  
reabilitação de pacientes com AVE**

Relatório submetido ao Laboratório de Inovação Tecnológica em Saúde (LAIS) da Universidade Federal do Rio Grande do Norte (UFRN) como requisito para atuação no projeto de Desenvolvimento Científico e Tecnológico aplicado à Esclerose Lateral Amiotrófica.

Fortaleza/CE

2021

# Resumo

Este projeto foi elaborado como proposta de um sistema embarcado voltado para auxílio na reabilitação de pacientes com atrofia muscular dos membros superiores com intuito de proporcionar melhoria na qualidade de vida de vítimas de Acidente Vascular Encefálico (AVE). O sistema será utilizado para controle de carrinhos de brinquedo em um tabuleiro de corrida a partir dos sinais musculares dos pacientes. Além do jogo, o sinal captado é tratado, filtrado e armazenado para aferição e análise de profissionais terapeutas envolvidos.

**Palavras-chave:** Sistema Embarcado. Reabilitação. Qualidade de Vida. Acidente Vascular Encefálico. Sinais Musculares. Jogo.

# Lista de ilustrações

Figura 1 – Sistema de medição de eletromiograma . . . . .	11
Figura 2 – Diagrama de blocos do biossensor conectado ao Sistema Embarcado . .	12
Figura 3 – Fonte simétrica de alimentação do Biossensor e Microcontrolador . . .	12
Figura 4 – Amplificador LM741 na configuração subtrator . . . . .	13
Figura 5 – Configuração do filtro passa-banda . . . . .	14
Figura 6 – Resposta em frequência do filtro passa-banda . . . . .	15
Figura 7 – Estágio de Amplificação e Retificação do sinal EMG . . . . .	16
Figura 8 – Topologia completa do Biossensor EMG . . . . .	17
Figura 9 – Relação SNR do EMG após aplicação de filtro digital . . . . .	18
Figura 10 – Fases de processamento do sinal muscular no Biossensor EMG . . . . .	19
Figura 11 – Representação discreta do filtro média móvel com 5 pontos . . . . .	20
Figura 12 – Medição de sinais EMG sem filtragem de um paciente . . . . .	21
Figura 13 – Medição de sinais EMG filtrados de um paciente . . . . .	22
Figura 14 – Esquemático de comunicação entre sensores, placa e computador . . . .	23
Figura 15 – Criação de portas seriais virtuais com o VSPE . . . . .	24
Figura 16 – Exemplo de plotagem em tempo real de um paciente no computador . .	26
Figura 17 – Recuperação dos dados coletados do paciente após a competição . . . .	27
Figura 18 – Modelo de funcionamento de um Autorama analógico . . . . .	28
Figura 19 – Circuito elétrico da pista de Autorama . . . . .	29
Figura 20 – Esquemático de automação do tabuleiro de corrida . . . . .	30
Figura 21 – Esquemático da fonte de tensão DC projetada para o tabuleiro . . . . .	31
Figura 22 – Ligação entre motor DC e microcontrolador . . . . .	32
Figura 23 – Modelo de configuração proposto para os carrinhos do sistema . . . . .	33
Figura 24 – Representação de modulações PWM (Duty Cycle) . . . . .	34
Figura 25 – Representação da ligação de controle do PWM com o carrinho . . . . .	35
Figura 26 – Controle de velocidade do motor em malha aberta . . . . .	36
Figura 27 – Configuração completa do tabuleiro de corrida . . . . .	37
Figura 28 – Simulação dos sinais EMG filtrados no microcontrolador . . . . .	45
Figura 29 – Simulação do Sistema de monitoramento dos 4 pacientes . . . . .	46
Figura 30 – Simulação do Sistema de Controle dos carrinhos no Proteus . . . . .	47
Figura 31 – Simulação do Sistema completo no Proteus . . . . .	47
Figura 32 – Modelo de circuito impresso dos 4 sensores conectados ao ATmega2560	48
Figura 33 – Vista frontal da placa PCB projetada . . . . .	49
Figura 34 – Vista lateral da placa PCB projetada . . . . .	49
Figura 35 – Vista traseira da placa PCB projetada . . . . .	50

# Lista de tabelas

Tabela 1 – Lista de Materiais e Orçamento do projeto . . . . .	38
Tabela 2 – Lista de Fornecedores Nacionais e Internacionais . . . . .	39

# Lista de abreviaturas e siglas

AVE	Acidente Vascular Encefálico
DDP	Diferença de Potencial elétrica
EMG	<i>Electromyography</i>
PWM	<i>Pulse Width Modulation</i>
GPIO	<i>General Purpose Input/Output</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
VSPE	<i>Virtual Serial Ports Emulator</i>
DC	<i>Direct Current</i>
LED	<i>Light-emitting diode</i>
ADC	<i>Analog-to-Digital Converter</i>
SNR	<i>Signal-to-noise ratio</i>
PCB	<i>Printed Circuit Board</i>

# Sumário

	<b>Introdução</b>	<b>8</b>
<b>1</b>	<b>CANAL DE INSTRUMENTAÇÃO</b>	<b>10</b>
1.1	Sinais musculares de Eletromiografia (EMG)	10
1.2	Construção do Biossensor EMG	11
1.2.1	Alimentação do Biossensor	12
1.2.2	Cálculo da diferença de potencial muscular entre os Eletrodos	13
1.2.3	Filtragem dos sinais EMG	14
1.2.4	Amplificação e Retificação dos sinais EMG	15
1.2.5	Simulação de sinais musculares no Biossensor por arquivos de áudio	17
1.3	Tratamento Digital do sinal Muscular	18
1.3.1	Filtro de Média Móvel	19
1.3.2	Implementação	20
1.3.3	Resultados da filtragem	21
<b>2</b>	<b>COMUNICAÇÃO DO SISTEMA</b>	<b>23</b>
2.1	Visão Geral	23
2.2	Comunicação serial UART	24
2.3	Plotagem dos dados em tempo real via PyQtGraph	25
2.4	Acesso dos sinais EMG pelo <i>Datalogger</i> do sistema	26
<b>3</b>	<b>PLATAFORMA DE CONTROLE PARA O CARRINHOS</b>	<b>28</b>
3.1	Automação do tabuleiro de corrida	28
3.1.1	Funcionamento de um Autorama analógico ( <i>Slot Car</i> )	28
3.1.2	Adição de fendas e trilhas metálicas no tabuleiro	30
3.1.3	Projeto de fonte DC para alimentação da pista	31
3.1.4	Adição de motor DC nos carrinhos de brinquedo	32
3.2	Algoritmo de Controle <i>Feedforward</i>	33
3.2.1	Modulação por Largura de Pulso (PWM)	33
3.2.2	Conexão do sinal de controle PWM com o carrinho	35
3.2.3	Código de Controle: Implementação	36
3.3	Indicativo visual para o ganhador da competição	37
<b>4</b>	<b>ORÇAMENTO DO SISTEMA PROPOSTO</b>	<b>38</b>
<b>5</b>	<b>LISTA DE FORNECEDORES</b>	<b>39</b>

REFERÊNCIAS . . . . .	40
APÊNDICE A – CÓDIGOS IMPLEMENTADOS . . . . .	41
APÊNDICE B – MEMORIAL DE CÁLCULOS . . . . .	44
APÊNDICE C – SIMULAÇÕES DO SISTEMA . . . . .	45
APÊNDICE D – MODELO DE PLACA PCB IMPLEMENTADA . .	48



# Introdução

O **Projeto V**, referente ao tema de **Sistemas embarcados**, tem como proposta a elaboração de um sistema destinado a pacientes acometidos por Acidente Vascular Encefálico (AVE) com atrofia muscular de membros superiores. Tendo em consideração as orientações repassadas para a Fase 2, foi desenvolvido um sistema completo com a criação de um canal de instrumentação para aquisição e tratamento dos sinais musculares dos pacientes e uma plataforma de controle para os carrinhos de brinquedo no tabuleiro de corrida. Um biossensor de eletromiografia foi desenvolvido para recepção dos sinais musculares que são processados e filtrados em um microcontrolador ATmega2560.

Essas informações recebidas pelo microcontrolador controlam a velocidade dos carrinhos conforme a amplitude do sinal muscular do paciente, ou seja, quanto maior o potencial do sinal muscular medido, maior a distância percorrida por cada carrinho no tabuleiro. Tal configuração proposta representa um sistema em malha aberta em que o sinal de entrada controla diretamente o comportamento final do carrinho por meio da sua velocidade. Em paralelo a este processo, foi desenvolvido um script em Python para plotagem em tempo real dos sinais musculares de até 4 pacientes no display (tela) do computador para análise dos terapeutas. Após o término da aferição dos dados, é possível armazená-los em um *datalogger* no computador para análises futuras. Durante este processo, o carrinho que ultrapassar a linha de chegada ativa um LED de indicação no tabuleiro designando o devido vencedor.

Para o tabuleiro de corrida foi proposto um modelo similar ao utilizado no Autorama analógico (*Slot Car*) utilizado largamente no Brasil na década de 70. Neste brinquedo, os carros se movem em fendas devidamente espaçadas de acordo com uma corrente elétrica controlada por um dispositivo manual. Cada carro possui um motor DC interno responsável pela rotação das rodas traseiras de cada carrinho. Neste projeto, ao invés de controlarmos a corrente elétrica de controle manualmente, é proposto um controle em malha aberta de acordo com a intensidade do sinal muscular do paciente através da modulação por largura de pulsos (PWM) do microcontrolador. Tal configuração é mais adequada porque cada paciente não fica diretamente conectado por fios ao seu carrinho de corrida uma vez que o sinal trafega pelas trilhas metálicas individualmente.

O sistema completo conta com um protocolo de segurança para que os pacientes não sofram eventuais surtos de tensão provenientes dos circuitos desenvolvidos. A alimentação dos sensores e do microcontrolador acontece por intermédio de baterias de 9V garantindo tensões e correntes elétricas incapazes de romper com a resistência elétrica do corpo. Apesar de ter sido construído uma fonte de tensão contínua para alimentação das trilhas

do tabuleiro de corrida por meio da rede elétrica, os circuitos do microcontrolador estão protegidos por meio de retificadores e diodos especiais para tensões instáveis provenientes da rede. A seguir encontra-se a estrutura e organização do trabalho ao longo deste documento:

- Todo o projeto com pontos obrigatórios e optativos encontram-se descritos neste único documento;
- Cada seção está organizada para a descrição da construção do sistema por partes;
- Cada seção contém embasamento teórico e técnico;
- Os esquemáticos dos circuitos desenvolvidos são apresentados diretamente no software Proteus e LTspice;
- Foram adotados certos níveis de abstração para a construção do sistema a nível de simulação;
- Os códigos e esquemáticos completos desenvolvidos estão presentes dentro do repositório deste projeto;
- Alguns trechos de códigos desenvolvidos para o sistema estão disponíveis para análise no Apêndice A;
- Diagramas de blocos e esquemáticos são acrescentados para melhor alusão dos conceitos propostos;
- Os modelos matemáticos explicados ao longo do trabalho encontram-se no Memorial de cálculos do Apêndice B;
- A lista de materiais/equipamentos encontra-se junto com a Tabela 1 de orçamento.

**Importante:** O orçamento da proposta assim como seus possíveis fornecedores são descritos detalhadamente ao final deste trabalho. No Apêndice C encontram-se os prints de tela da simulação do sistema de controle dos carrinhos e do sistema de processamento dos sinais EMG. No Apêndice D encontram-se imagens de um modelo PCB (*printed circuit board*) desenvolvido no software Proteus dos 4 biossensores conectados no microcontrolador ATmega2560 em uma mesma placa.

# 1 Canal de Instrumentação

De acordo com as orientações do concurso o sistema deve apresentar um canal de instrumentação para aquisição e tratamento do sinal muscular. Para este critério resolveu-se criar um biossensor simples de eletromiografia (EMG) por meio de Amplificadores Operacionais, resistores e capacitores. Esse sensor gera na saída uma tensão entre 0 e 5 Volts e é alimentado por uma tensão simétrica de +9 e -9 Volts. Quanto maior é a atividade muscular, maior é a tensão na saída. Para utilização desse biossensor é necessário utilizar 3 eletrodos (vendidos separadamente) que podem ser conectados diretamente aos amplificadores operacionais por meio de cabos. A seguir está uma breve descrição do funcionamento elétrico muscular relacionado à atividades de contração.

## 1.1 Sinais musculares de Eletromiografia (EMG)

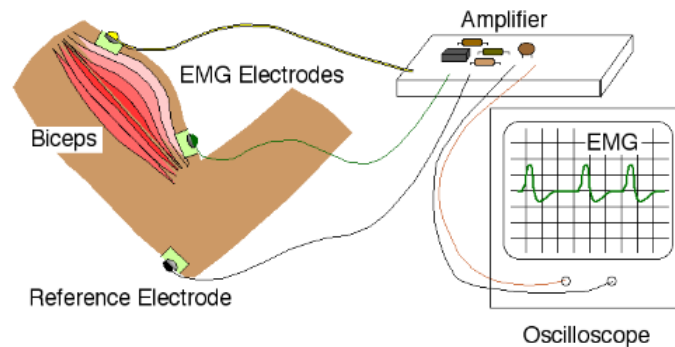
A eletromiografia (EMG) é uma técnica responsável pela medição e monitoramento da atividade elétrica em resposta à estimulação nervosa de um músculo [1]. A aferição destes sinais se dá por meio de eletrodos que captam este sinal e o “traduzem” em sinais gráficos, numéricos ou sonoros para serem interpretados por especialistas. Normalmente são utilizados 3 eletrodos para a determinação das atividades musculares em músculos pequenos, como por exemplo, o bíceps [2].

A Figura 1 representa um modelo simples de eletromiografia muscular por meio de 3 eletrodos. Nela, os 2 eletrodos principais se localizam nas duas extremidades do bíceps braquial para determinar a diferença de potencial de um estímulo nervoso. O 3º eletrodo se localiza em uma superfície óssea como uma referência de potencial em 0 Volts (terra) [3]. Geralmente os sinais são medidos na escala de miliVolts e necessitam ser amplificados externamente por um circuito eletrônico para devida análise [4].

Incluso ao sinal muscular aferido estão também presentes outros sinais que causam interferência como ruídos eletromagnéticos externos. Os sinais EMG são altamente sensíveis a esses ruídos externos por estarem localizado normalmente numa frequência entre 3 a 500  $Hz$  [1]. Para isso, é necessário fazer uma filtragem por meio de um filtro analógico ou digital que remova as frequências acima de 500  $Hz$  e abaixo de 3  $Hz$ .

Por meio dessas informações, no projeto proposto foi implementada uma fase de pré-processamento do sinal no próprio circuito do biossensor por meio de um filtro passa-faixa. Após essa filtragem, já garantimos de que o sinal é de natureza muscular. Após a fase de filtragem já podemos passar para a fase de amplificação e retificação do sinal para melhor leitura no conversor A/D do microcontrolador ATmega2560. Nesse último é

Figura 1 – Sistema de medição de eletromiograma



Fonte: Adaptado de [2].

onde teremos o algoritmo embarcado responsável pelo processamento digital do sinal. Na próxima seção encontram-se os detalhes do biossensor EMG feito no software Proteus.

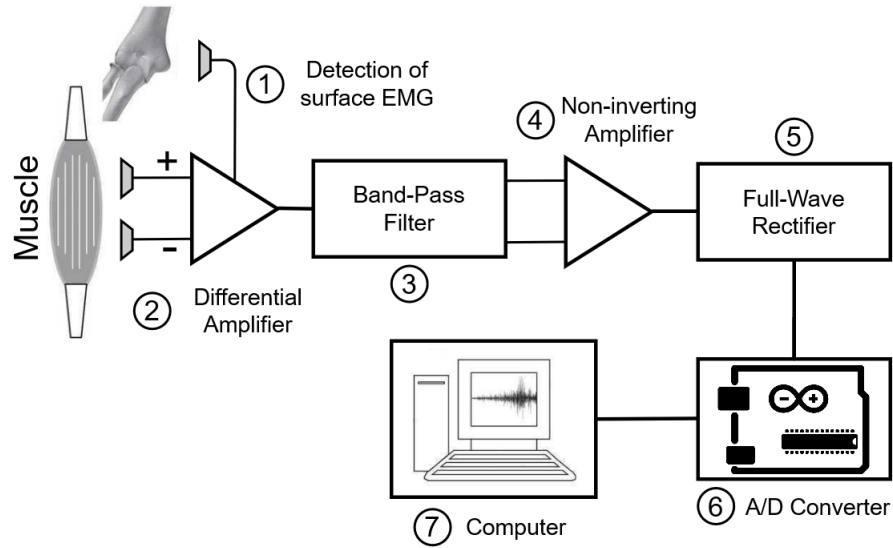
## 1.2 Construção do Biossensor EMG

A Figura 2 representa um esquemático simples do biossensor construído no software Proteus. Primeiramente temos a aquisição da diferença de potencial do sinal muscular por meio de uma configuração diferencial de um Amplificador operacional subtrator. Em seguida o sinal é pré-processado por um filtro passa-banda para a remoção de frequência fora da faixa de 3 a 500  $Hz$ . Logo após o sinal filtrado é amplificado por meio de uma configuração não-inversora de um amplificador operacional e retificado completamente para ser processado pelo microcontrolador ATmega2560.

O sinal muscular amplificado varia entre 0 e 5 Volts devido a faixa de tensão suportada pelas portas de conversão A/D do microcontrolador. Por fim, os sinais são mostrados no display (tela) do computador em tempo real por meio de scripts em Python que estão conectados diretamente às portas seriais do ATmega2560. Como este microcontrolador possui 4 portas seriais UART, fica a critério do projetista enviar todos os dados individualmente por cada porta ou somente por uma. O sistema foi desenvolvido para aceitar essas duas possibilidades.

Dessa forma, as portas seriais do computador podem estar diretamente conectadas com o microcontrolador plotando os dados por meio da biblioteca PyQtGraph uma vez que a biblioteca Matplotlib é muito lenta para esta tarefa. Cada paciente possui um arquivo CSV no computador para armazenamento dos dados coletados durante a competição. Estes dados podem futuramente ser consultados e mostrados em um gráfico para análises específicas. A seguir serão apresentados os circuitos projetados no Proteus referente a cada etapa da Figura 2 que representa o esquemático geral do sensor.

Figura 2 – Diagrama de blocos do biossensor conectado ao Sistema Embarcado

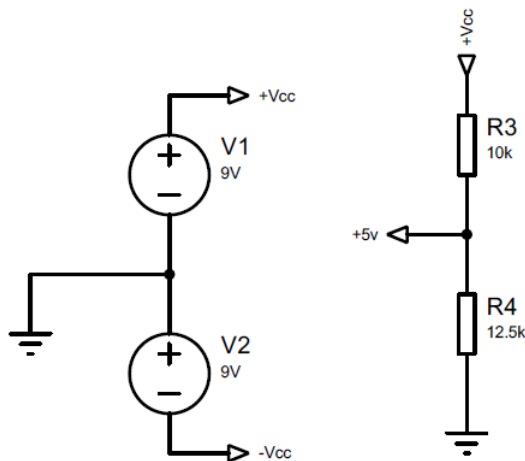


Fonte: Adaptado de [3].

### 1.2.1 Alimentação do Biossensor

A Figura 3 representa duas baterias conectadas entre si para fornecimento de uma tensão simétrica de +9V (+Vcc) e -9V (-Vcc) para alimentação dos Amplificadores Operacionais LM741 do circuito eletrônico do biossensor.

Figura 3 – Fonte simétrica de alimentação do Biossensor e Microcontrolador



Fonte: Elaborado pelo autor.

Tal tensão não é capaz de fornecer nenhum tipo de descarga elétrica nos pacientes. Seguida dessa fonte simétrica há um divisor de tensão de 5V para alimentação do microcontrolador. De acordo com [5] a relação da tensão de saída  $V_{out}$  pela tensão de entrada

$V_{in}$  descreve um divisor de tensão descrito pela equação:

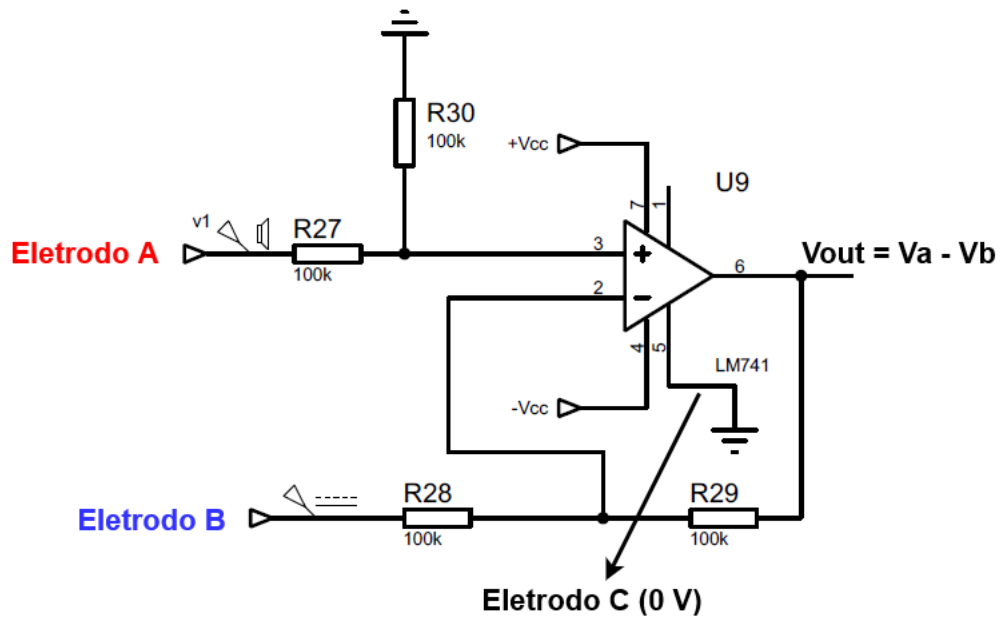
$$\frac{V_{out}}{V_{in}} = \frac{R_3}{R_3 + R_4} \quad (1.1)$$

Considerando uma tensão de saída desejada de 5V, uma tensão de entrada da fonte simétrica de +9V e um resistor  $R_3$  de 10  $k\Omega$  e utilizando a relação 1.1 para encontrar a resistência desconhecida, temos que o valor de  $R_4$  é igual a 12.5  $k\Omega$ . Esses valores de tensão garantem um bom funcionamento do sistema embarcado e do biossensor sem fornecer riscos de choques ou qualquer tipo de problema aos envolvidos.

### 1.2.2 Cálculo da diferença de potencial muscular entre os Eletrodos

Primeiramente é necessário calcular a diferença de potencial (ddp) elétrica de um estímulo muscular através dos eletrodos de conexão. Para isso, foi utilizado um Amplificador Operacional LM741 na configuração subtrator para calcular a ddp entre os sinais A e B de acordo com a Figura 4.

Figura 4 – Amplificador LM741 na configuração subtrator



Fonte: Elaborado pelo autor.

De acordo com essa configuração, temos os 3 eletrodos do paciente conectados ao Amplificador Operacional LM741 alimentado pela fonte de tensão simétrica. Dessa forma,

conforme [5], a tensão de saída  $V_{out}$  do amplificador é dada pela relação:

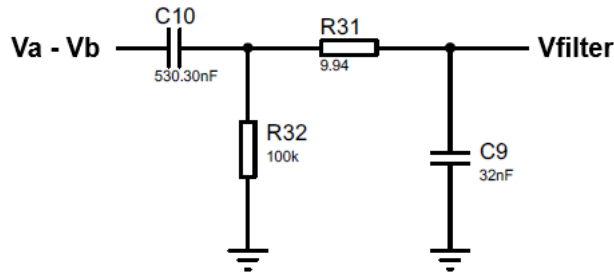
$$V_{out} = \frac{V_A \left(1 + \frac{R_{29}}{R_{28}}\right)}{\left(1 + \frac{R_{27}}{R_{30}}\right)} - V_B \frac{R_{29}}{R_{28}} \quad (1.2)$$

Pela Equação 1.2 temos que, se escolhermos todos os resistores com resistências iguais, a saída fica simplesmente a subtração entre os sinais elétricos A e B das pontas de prova do eletrodo. Sendo assim, já temos a primeira parte do biossensor implementada com a devida subtração dos potenciais elétricos do paciente. O próximo passo é filtrar estes sinais entre uma frequência de 3  $Hz$  e 500  $Hz$ .

### 1.2.3 Filtragem dos sinais EMG

Para a filtragem dos sinais EMG é necessário montar uma configuração passa-faixa ou passa-banda representada na Figura 5. Esse filtro permite a passagem dos sinais em uma determinada faixa intermediária, ou seja, ele irá atenuar sinais que estejam abaixo ou acima de uma determinada frequência [6]. Essas delimitações são também conhecidas como frequência de corte inferior e frequência de corte superior [5]. Nesta topologia, teremos uma frequência de passagem contendo de fato os sinais EMG.

Figura 5 – Configuração do filtro passa-banda



Fonte: Elaborado pelo autor.

Este filtro é construído por outras 2 configurações em sequência: filtro passa-altas e filtro passa-baixas. Para determinar as frequências de corte inferior e superior é necessário calcular as frequências de corte dos filtros passa-altas e passa-baixas, respectivamente. A frequência de corte  $F_c$  desses dois filtros é dada pela relação:

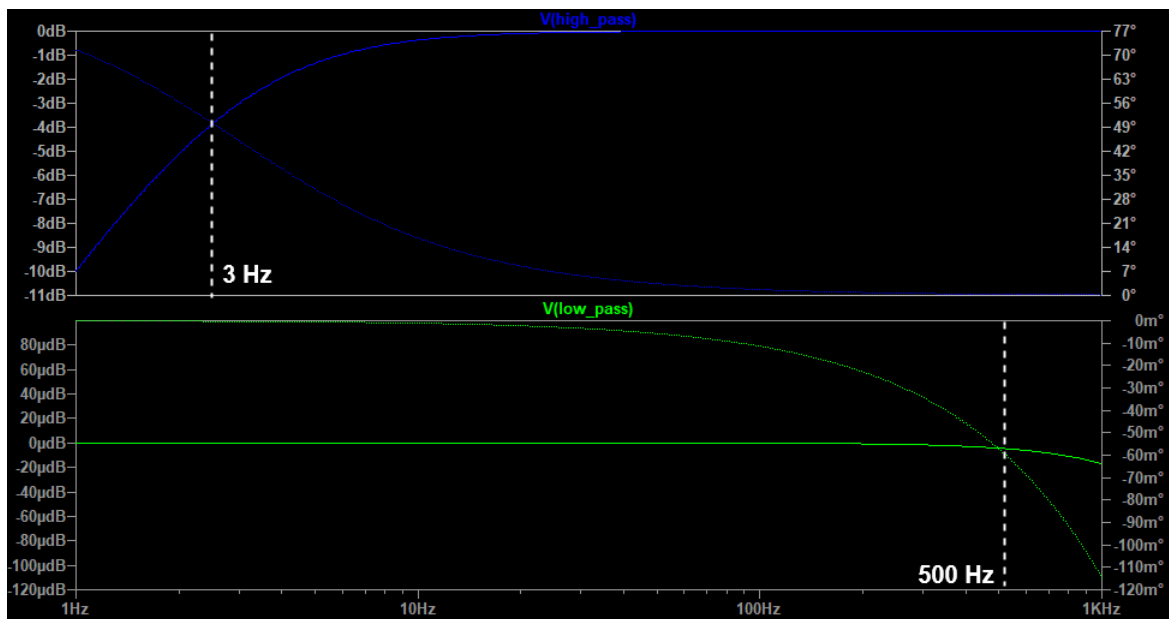
$$F_c = \frac{1}{2\pi RC}, \quad (1.3)$$

em que R e C são os valores da resistência e capacitância, respectivamente [5]. Considerando uma frequência de corte inferior de 3  $Hz$ , uma resistência de 10  $k\Omega$  e aplicando a relação

1.3, temos que o valor da capacitância é igual a  $530.30 \text{ nF}$ . Para o cálculo da frequência de corte superior de  $500 \text{ Hz}$  e considerando uma capacitância de  $32 \text{ nF}$ , temos que a resistência do componente desconhecido é cerca de  $10 \Omega$  como visto na Figura 5.

Com uso do software LTspice foi possível analisar o Diagrama de frequência para os cortes do filtro do biossensor. Como podemos observar na Figura 6, temos as frequências de corte inferior e superior dos filtros passa-alta e passa-baixa são, respectivamente, muito próximas aos cortes especificados. O biossensor trabalhando nessa configuração já garante menos processamento no sinal digital no microcontrolador.

Figura 6 – Resposta em frequência do filtro passa-banda



Fonte: Elaborado pelo autor.

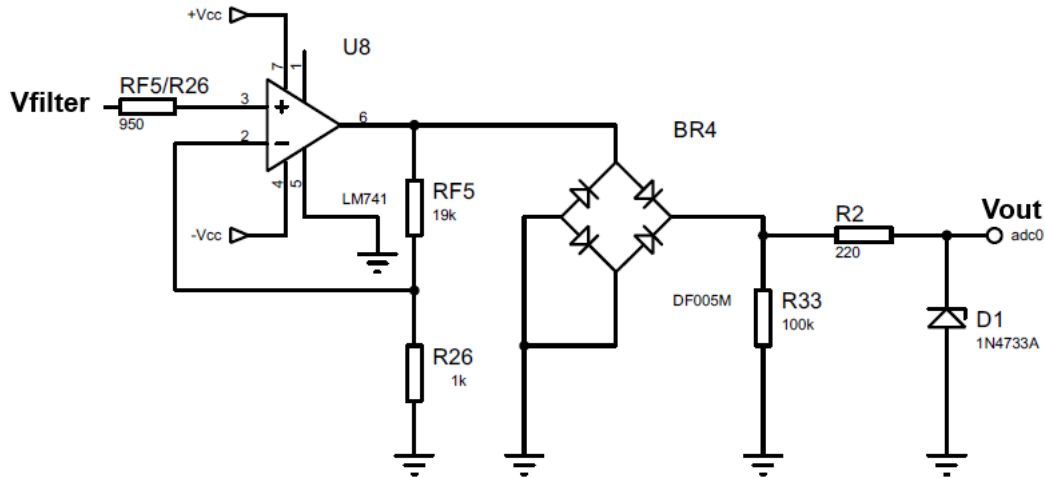
#### 1.2.4 Amplificação e Retificação dos sinais EMG

Na última etapa do biossensor se encontram os processos de amplificação e retificação completa dos sinais EMG como representado na Figura 7. Depois do filtro é necessário amplificar o sinal até um nível adequado para que este seja lido corretamente pelo conversor A/D do microcontrolador. Para isso foi utilizado uma configuração de amplificação não-inversora no LM741 para que o sinal não inverta sua fase ao decorrer do processo.

Por fim, ainda é colocado um regulador de tensão para estabilizar a saída do sensor entre as tensões de 0 a 5 Volts. Essa tensão é estabilizada de acordo com o circuito de proteção nas entradas A/D do embarcado pelo diodo Zener de 5V. Isso garante que o microcontrolador não seja danificado com tensões fora do limite permitido. Na entrada da última etapa ainda está presente um resistor de compensação  $R_{F5}/R_{26}$  para ajudar na



Figura 7 – Estágio de Amplificação e Retificação do sinal EMG



Fonte: Elaborado pelo autor.

compensação das tensões de erro do circuito [5]. Assim sendo, a topologia não-inversora do Amplificador Operacional permite amplificar o sinal de saída  $V_{out}$  acordo com a relação:

$$V_{out} = \left( \frac{R_{F5}}{R_{26}} + 1 \right) V_{in} \quad (1.4)$$

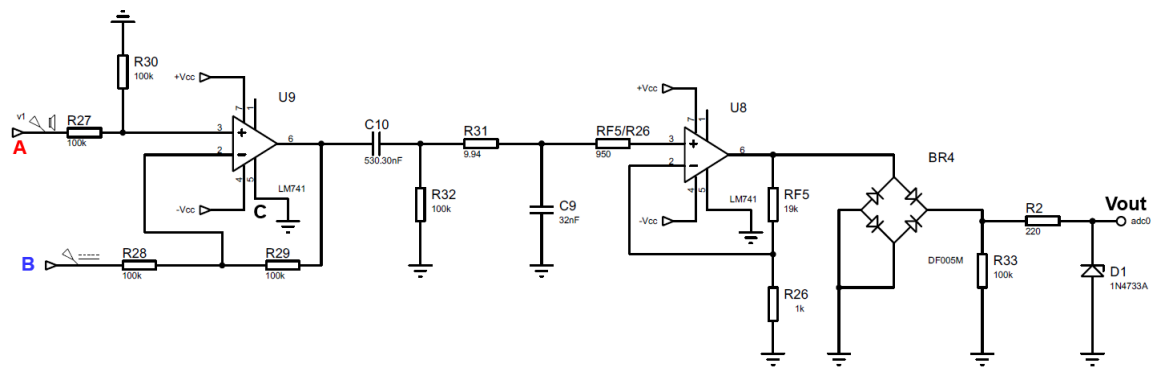
Neste projeto, foi considerado um ganho de 20 vezes o sinal de entrada de acordo com a Equação 1.4 [6]. O sinal de entrada, uma vez filtrado no estágio anterior, necessita ser amplificado para uma melhor leitura do conversor A/D de 10 bits do microcontrolador ATmega2560. Após esse estágio de amplificação, é necessário remover a parte negativa do sinal por meio de uma ponte retificadora completa para processamento posterior. Essa ponte, feita totalmente por diodos, garante uma leitura correta pelo conversor A/D do sistema embarcado. Por fim, o sinal de saída polariza um resistor de 220  $\Omega$  para que o diodo Zener regulador de tensão atue na saída do biossensor. Após todo esse processo, o sinal está pronto para ser processado digitalmente no microcontrolador.

Todo o processo aqui descrito para construção do biossensor aplica-se apenas para a leitura de 1 paciente. Como o sistema requer uma competição com 4 pacientes, então o sistema requer a construção de 4 biossensores completamente idênticos. Cada biossensor pode ser fabricado em uma placa PCB separadamente ou em conjunto para o produto final. Para o sistema específico do projeto, poderíamos construir uma PCB com os 4 sensores fixos juntamente com o chip embarcado e tornar o projeto mais prático e rentável em termos econômicos como foi representado no Apêndice D. Em termos de escalabilidade, o mais adequado seria a construção de sensores individuais que fossem acoplados diretamente no microcontrolador uma vez que este possui 16 entradas analógicas para uso imediato.

### 1.2.5 Simulação de sinais musculares no Biossensor por arquivos de áudio

A Figura 8 representa a topologia completa do biossensor EMG construído no software Proteus. Nele encontra-se operações básicas de amplificação e pré-processamento analógico dos sinais musculares para que estes sejam processados no microcontrolador de forma adequada. Para isso, é necessário utilizar sinais similares a eletromiogramas reais para que o projeto esteja sustentado o mais próximo da realidade mesmo em um ambiente de simulação.

Figura 8 – Topologia completa do Biossensor EMG



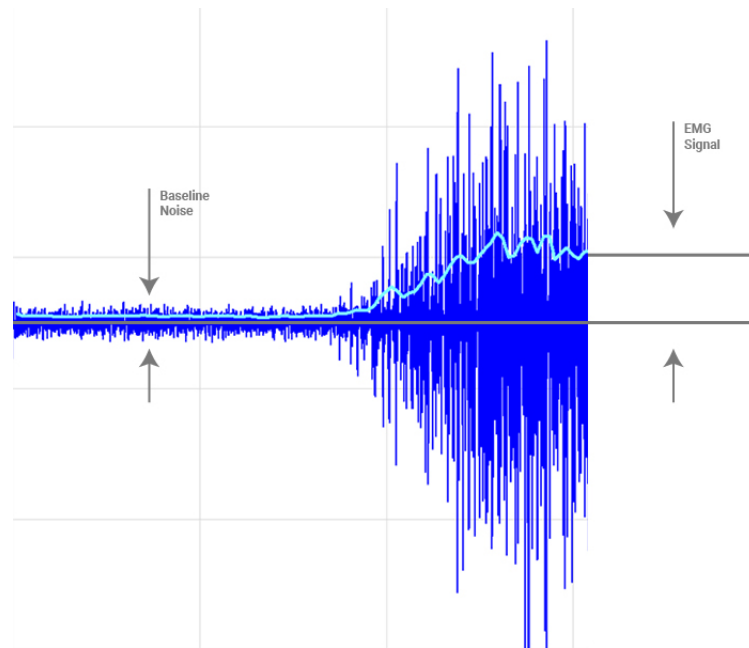
Fonte: Elaborado pelo autor.

Sinais EMG se assimilam muito com sinais de áudio devido à sua natureza de se acoplar à diferentes fontes de ruído externo [6]. Para isso, mesmo após a filtragem feita anteriormente no próprio biossensor, o sinal EMG ainda apresenta ruídos em sua composição devido a características intrínsecas do sistema como design do sensor e a quantidade de tecido gorduroso entre a pele e o músculo do paciente. A fim de amenizar essas flutuações, torna-se necessário um processamento digital no microcontrolador por meio de filtros digitais como Média móvel ou Kalman.

A melhor forma de medir a qualidade do sinal EMG é através da Relação Sinal Ruído [4]. Essa medida indica a relação do sinal EMG durante a contração muscular versus os sinais elétricos indesejados quando o músculo está relaxado (*baseline noise*) [3]. A Figura 9 indica a medição de um sinal muscular real e a relação SNR após a realização de um filtro digital. Quanto maior o SNR, mais confiável são os dados do EMG em relação à referência [2]. Por este motivo, para termos de simulação, foram utilizados para cada paciente um sinal de áudio diferente representando as contrações musculares ao decorrer do tempo.

Como o sinal de áudio é muito próximo ao comportamento de um sinal EMG real, então foram utilizados melodias como forma de abstração dos sinais coletados pelos pacientes através dos eletrodos. Estes sinais, por sua vez, passarão por todas as etapas

Figura 9 – Relação SNR do EMG após aplicação de filtro digital



Fonte: Retirado de [7].

apresentadas anteriormente do biossensor até chegarem devidamente processados nos canais A/D do microcontrolador.

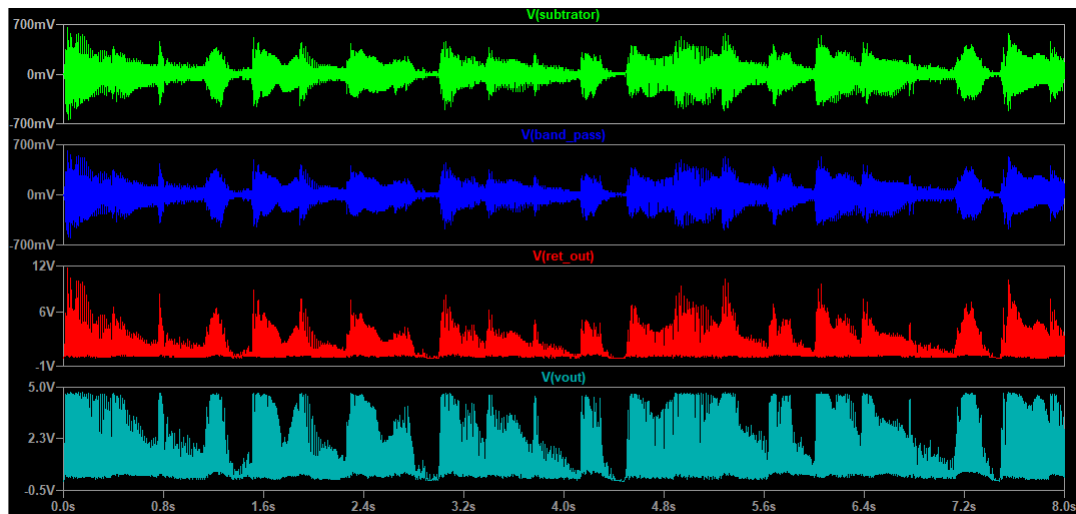
A Figura 10 representa os principais estágios de processamento do sinal EMG do paciente 1 ao longo do biossensor. Para a realização dessa figura foi necessário o uso do software LTspice para uma plotagem mais organizada em comparação com o software Proteus. O sinal na cor verde indica a saída do amplificador subtrator responsável pela ddp entre os eletrodos A e B conectados ao paciente. Este sinal encontra-se em miliVolts e representa bem a dinâmica real de um sinal EMG.

O sinal em azul, por sua vez, representa o sinal filtrado entre as frequências de 3 a 500  $Hz$ . Como o sinal está no domínio do tempo, não há muita diferença do sinal verde. No entanto, no domínio da frequência esse sinal possui harmônicos somente na faixa projetada. O sinal em vermelho indica a amplificação do EMG e sua retificação completa. Por fim, o sinal na cor verde claro está situado entre 0 a 5 Volts graças ao diodo Zener regulador de tensão. Este sinal está pronto para ser processado digitalmente no microcontrolador.

### 1.3 Tratamento Digital do sinal Muscular

Para a realização do tratamento do sinal muscular adquirido pelo biossensor foi necessário aplicar um filtro Média móvel simples para eliminar os picos indesejados e suavizá-lo para mostragem em tempo real no display (tela) do computador. Existem

Figura 10 – Fases de processamento do sinal muscular no Biossensor EMG



Fonte: Elaborado pelo autor.

diversas técnicas que são empregadas e mais eficazes do que um filtro de média móvel, porém, para este trabalho, a utilização desse filtro simplifica a compreensão de conceitos mais complexos.

### 1.3.1 Filtro de Média Móvel

O filtro de média móvel é um filtro passa baixas simples do tipo FIR (*Finite Impulse Response*) comumente usado para regular o comportamento de arrays de sinais/dados coletados [6]. Este filtro coleta  $N$  amostras de entrada e tira a média desses pontos. O resultado dessa operação segue continuamente e resulta em diversos pontos conforme mais entradas vão sendo lidas. Conforme o tamanho do filtro aumenta, a suavidade da saída também aumenta [1]. O filtro de média móvel realiza 3 operações importantes:

- Capta  $N$  pontos de entrada, calcula a média destes  $N$  pontos e produz uma saída única;
- Devido a computação envolvida no processo, o filtro introduz uma quantidade de delay na saída;
- O filtro atua como um passa-baixa e possui uma boa resposta no domínio do tempo.

É importante lembrar que o ATmega2560 possui um conversor analógico-digital (A/D) com resolução de 10 bits e clock dependendo de um fator prescaler. Esse fator tem valor padrão de 128, logo, o clock padrão do microcontrolador vale 16 MHz dividido por 128, o que resulta em 125 kHz. Uma conversão normal necessita de 13 pulsos de clock no

conversor A/D, então, o número de amostras por segundo valerá  $125\text{ kHz}$  dividido por 13 resultando em aproximadamente 9600 amostras por segundo [8].

### 1.3.2 Implementação

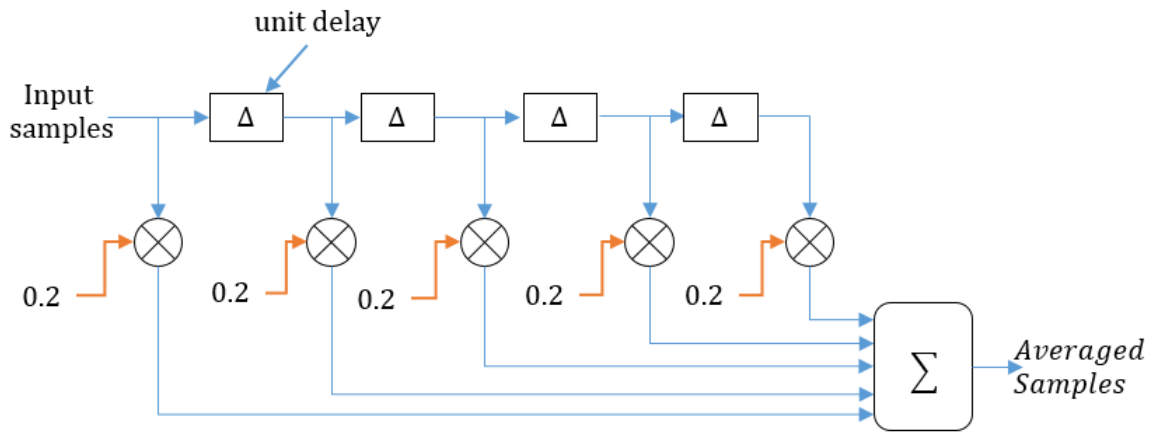
De acordo com [6], a Equação de diferenças de um filtro média móvel de  $N$  pontos com entrada representada pelo vetor  $x$  e o vetor de saída  $y$  é representada por:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (1.5)$$

Por exemplo, um filtro média móvel de 5 pontos leva a atual e últimas 4 amostras da entrada e calcula a média. Essa operação está representada na Figura 11 com as seguintes equações de diferenças relacionando a entrada e a saída em tempo discreto:

$$y[n] = \frac{1}{5} (x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4]) \quad (1.6)$$

Figura 11 – Representação discreta do filtro média móvel com 5 pontos



Fonte: Elaborado pelo autor.

A unidade de delay mostrada na Figura 11 pode ser representada pelas amostras de entrada do sensor EMG deslocadas em um array armazenado diretamente na memória do microcontrolador. Por meio de uma função é possível deslocar essas amostras em tempo real e calcular a saída do filtro digital com um número fixo de pontos definidos dentro do próprio código do sistema embarcado [6].

É bom lembrar que o filtro média móvel é aplicado juntamente com o mesmo número de amostras para todos os 4 pacientes. Conforme a leitura dos 4 canais A/D do microcontrolador são lidos, o cálculo de média móvel é executado paralelamente à

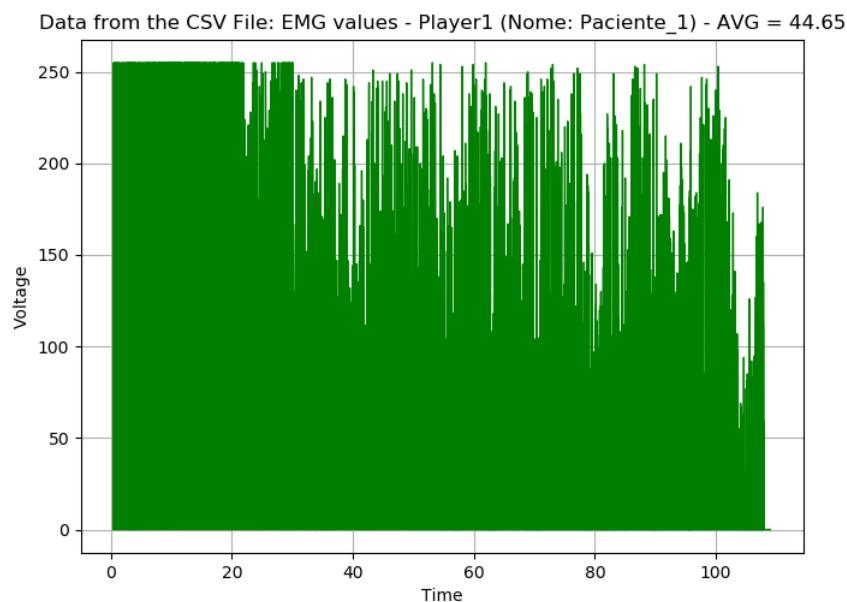
leitura dos dados mandados pelos 4 biossensores EMG. Para mais informações, o código implementado para este tipo de filtro se encontra no Apêndice A.

### 1.3.3 Resultados da filtragem

Para exemplificar o processo de filtragem de um sinal muscular pelo biossensor construído, obteve-se um sinal de teste por meio da nota Sol (G) do violão. Essa nota por sua vez representará um estímulo nervoso apenas para verificar a eficácia do filtro média móvel. Na seção do próximo capítulo será discutido como as formas de onda apresentadas nesta seção foram obtidas. Por enquanto, considere que um script em Python, conectado por uma porta serial virtual no computador esteja sendo executado.

O microcontrolador do software Proteus está conectado por uma porta serial do desktop (COM1) enquanto que o código em Python “escuta” tudo por uma COM1 virtual emulada pelo software *Virtual Serial Ports Emulator* (VSPE). Por meio desse script, foi capturado os sinais recebidos por um suposto paciente através do sensor sem a aplicação do filtro de Média móvel. Os resultados dessa aferição se encontram na Figura 12 e no Apêndice C com as simulações reais do sistema em funcionamento.

Figura 12 – Medição de sinais EMG sem filtragem de um paciente



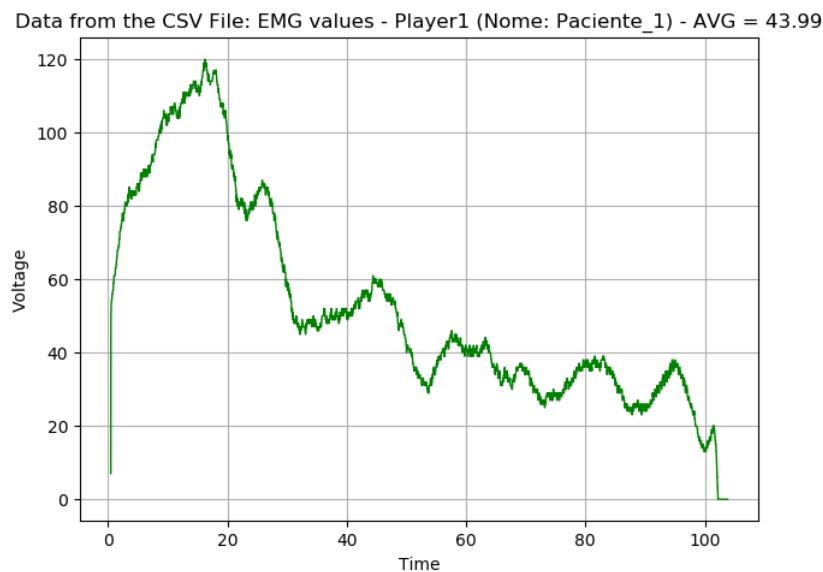
Fonte: Elaborado pelo autor.

A Figura acima representa um corte superior devido à ação do diodo Zener no Proteus. Além disso, não retira-se muita informação à respeito do sinal coletado. Em um primeiro momento, percebe-se que o paciente realizou contrações mais fortes no músculo no início da medição. Durante um certo tempo, o potencial elétrico dessa contração foi diminuindo. No entanto, ainda é possível melhorar a qualidade desse sinal com o filtro

média móvel proposto. Para isso, realizou-se o mesmo experimento com o mesmo sinal por meio do filtro média móvel.

A Figura 13 representa o sinal já filtrado digitalmente do sinal EMG no microcontrolador. Percebe-se logo de imediato uma grande diferença em comparação com a Figura 12. Primeiramente o paciente faz contrações mais fortes no início da medição e contrações menos fortes no final. A estimativa do filtro foi muito boa, uma vez que, com os dados não filtrados, já foi possível realizar esta distinção de maneira visual.

Figura 13 – Medição de sinais EMG filtrados de um paciente



Fonte: Elaborado pelo autor.

Para os testes realizados com este sinal foram utilizados 50 pontos de amostras de entrada para realizar a média. É importante destacar também que nas Figuras 12 e 13 temos a média das amplitudes coletadas representadas pela sigla AVG. Essa foi uma característica implementada para se ter uma noção de qual paciente pode ganhar a corrida, uma vez que os carrinhos se movem proporcionalmente de acordo com a amplitude do sinal muscular. O paciente com uma maior média de amplitude ganhará a corrida como consequência.

Outra característica importante é que os gráficos foram plotados de acordo com o valor inteiro de tensão coletado pelo conversor A/D do microcontrolador no eixo Y. Para convertê-los em tensão real, é necessário multiplicar cada ponto por  $5.0/1023$  uma vez que o canal analógico-digital do microcontrolador possui 10 bits e pode representar até 5 Volts. No próximo capítulo será discutido como os dados são capturados para plotagem em tempo real e como é feita a comunicação entre o microcontrolador e o computador.

## 2 Comunicação do Sistema

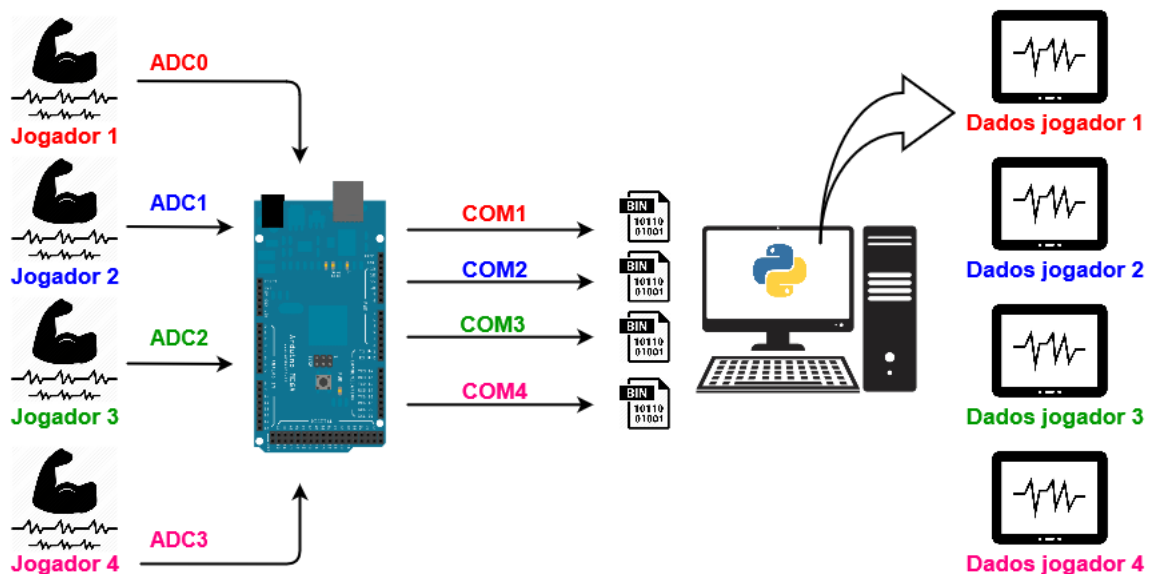
Neste capítulo serão apresentados alguns princípios básicos de comunicação do sistema no que diz respeito ao fluxo de dados entre as plataformas utilizadas. Os dados são apresentados em tempo real no display do computador e armazenados em um *Datalogger*.

### 2.1 Visão Geral

O sistema foi construído utilizando protocolos simples de comunicação e leitura de dados. Basicamente o microcontrolador ATmega2560 foi programado para receber dados de eletromiograma de até 4 pacientes pelas portas de conversão ADC0, ADC1, ADC2 e ADC3. Após o processamento digital dos dados, estes são encaminhados por 4 canais individuais de comunicação serial entre a placa e o computador. É bom lembrar que o sistema também foi desenvolvido para que todos os dados sejam transmitidos apenas por uma porta serial. Cada uma das 4 portas são adequadas para esta configuração.

Os dados coletados por meio do sensor desenvolvido são processados em um arquivo Python no computador para que ocorra a plotagem do desempenho individual de cada paciente em tempo real. Estes dados são armazenados em arquivos CSV no computador. O esquema básico de comunicação do sistema está representado na Figura 14.

Figura 14 – Esquemático de comunicação entre sensores, placa e computador



Fonte: Elaborado pelo autor.



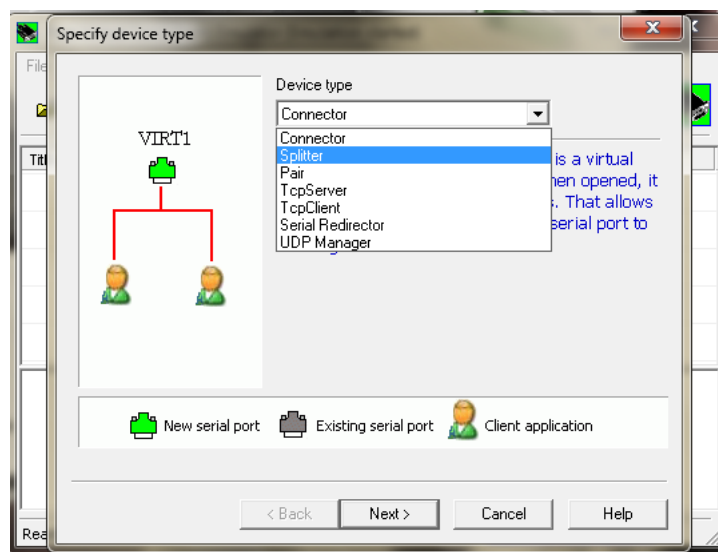
## 2.2 Comunicação serial UART

A finalidade da comunicação UART (*Universal asynchronous receiver/transmitter*) é possibilitar a transmissão e a recepção de dados entre as portas paralelas e serial de um computador [8]. O processamento desse protocolo transmite um byte de dados em série, bit a bit. Esses dados são então convertidos de volta em um byte completo na outra extremidade da conexão. O sistema embarcado do projeto permite a comunicação ponta a ponta do microcontrolador presente no software proteus com o próprio computador [9].

Uma interface COM é simplesmente uma interface de E/S que permite a conexão de um dispositivo serial a um computador [9]. Essas portas são interfaces assíncronas que podem transmitir um bit de dados por vez, quando conectadas a um dispositivo serial [10]. Por padrão, o microcontrolador ATmega2560 possui 4 conexões seriais livres, permitindo com que, dessa forma, possamos interagir com os dados em tempo real processados no sistema embarcado [11]. É por meio do protocolo UART que conseguimos comunicar o microcontrolador com o computador pelas suas portas seriais.

Uma vez que cada UART do microcontrolador se conecta a uma porta COM do computador, é impossível que outros dispositivos acessem essa mesma conexão existente no sistema. Para que possamos acessar diretamente os dados dos sinais EMG por meio de scripts externos é necessário a criação de portas seriais virtuais. Uma porta serial virtual é uma construção de software que emula a funcionalidade de uma interface serial para permitir a comunicação com dispositivos serial usando uma rede ou outra porta física [11]. É por meio dessa possibilidade que foi utilizado o software *Virtual Serial Ports Emulator* (VSPE) como mostra a Figura 15.

Figura 15 – Criação de portas seriais virtuais com o VSPE



Fonte: Elaborado pelo autor.

Este software, na qual possui licença gratuita, permite criar portas seriais virtuais a partir de pares com portas seriais já conectadas. Por exemplo, se criarmos um par COM1 e COM6, tudo que acontecer na porta COM1 refletirá na porta COM6. A partir disso, é possível estabelecer uma conexão entre um script em Python ou JavaScript para acesso direto aos dados da COM1 sem interferir no seu funcionamento.

## 2.3 Plotagem dos dados em tempo real via PyQtGraph

Um script em Python foi desenvolvido para coletar e plotar em tempo real os dados recebidos dos sensores de cada paciente por meio de 4 portas seriais virtuais ou apenas uma. Para isso, foi utilizado uma biblioteca chamada PyQtGraph para que o plot em tempo real fosse feito de forma mais rápida do que outras bibliotecas como o Matplotlib. Essa última não é recomendada para plotagem em tempo real. O script feito em Python encontra-se ao final deste trabalho no Apêndice A.

Enquanto os dados são processados no microcontrolador, é necessário que o script em Python esteja sendo executado ao mesmo tempo. Isso garante que o script conecte-se a todo instante às portas seriais virtuais do sistema e o plot aconteça em tempo real no display do computador. Ao mesmo tempo que o plot acontece, os dados são armazenados em arquivos CSV para que especialistas possam analisar os dados coletados futuramente de cada paciente.

Um outro script feito em Python permite que profissionais da saúde vejam na forma de gráficos os mesmos dados coletados e apresentados durante a competição. Tal funcionalidade foi proposta como um *Datalogger* para que os dados coletados e processados pelo microcontrolador sejam salvos no computador. Esse segundo script também encontra-se no Apêndice A para consulta.

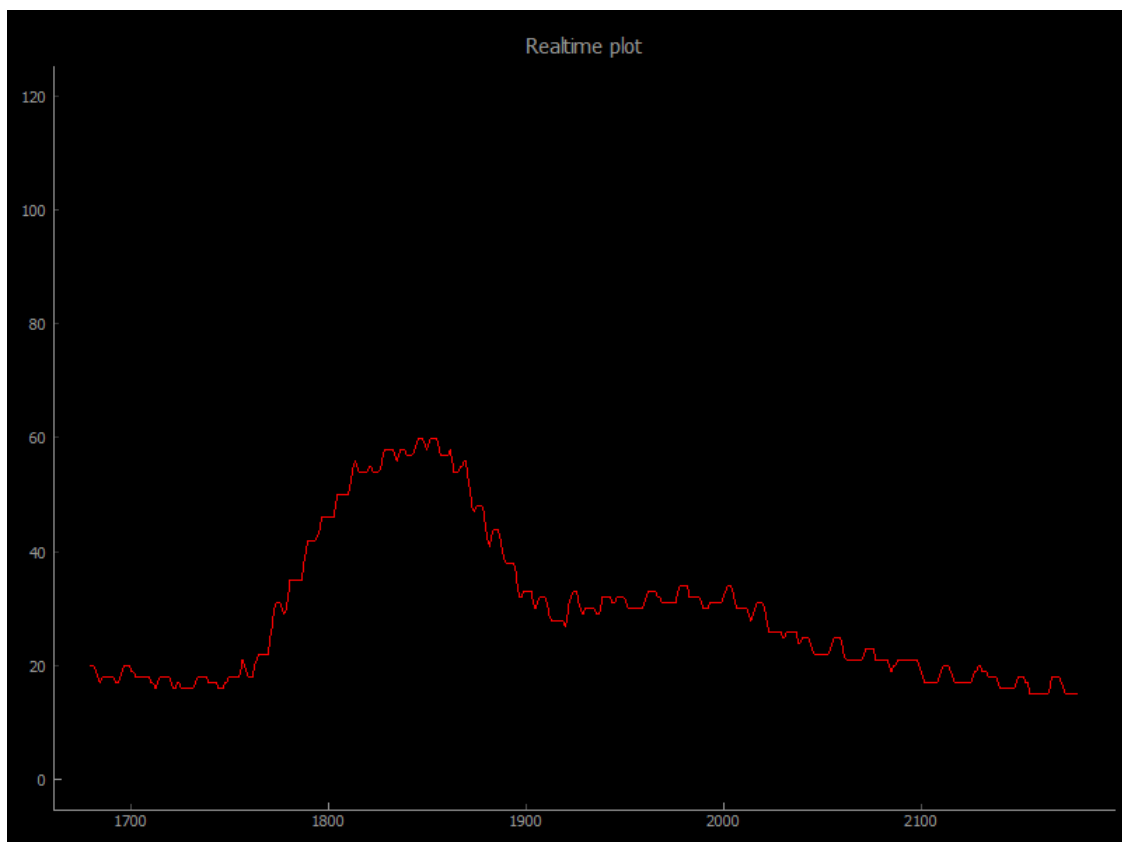
Os scripts presentes no Apêndice A mostram apenas os códigos para coleta de 1 paciente. Para que a coleta dos 4 pacientes sejam feitas de forma simultânea, é necessário executar os scripts de coleta dos dados separadamente considerando a utilização completa das 4 portas ou apenas um script considerando apenas uma porta. A coleta dos 4 sensores é feita apenas por 1 código no microcontrolador. Para plotagem posterior dos dados, apenas 1 script é necessário. Os códigos completos da proposta encontram-se no repositório do projeto.

A Figura 16 representa o gráfico do desempenho muscular de um paciente já processado digitalmente pelo microcontrolador sendo mostrado em tempo real no display (tela) do computador. O script em Python abre automaticamente uma janela aonde o gráfico vai atualizando a cada instante de tempo. Por meio dessa janela os profissionais de saúde podem acompanhar o progresso de cada paciente durante a competição. No sistema final, 4 janelas abrem individualmente para análises como representado na Figura 29. Esse

mesmo sinal é o que foi representado na Figura 10. O filtro média móvel implementado considerou 100 pontos para cálculo da média para todos os sinais EMG dos pacientes.

Na Figura 16 percebe-se que o paciente mantém um leve aumento de contração muscular durante um determinado período e retorna ao normal logo após. Neste mesmo período, o carrinho deveria se movimentar mais rápido uma vez que a amplitude do sinal aumentou. O controle implementado para a movimentação do carrinho é explicada no próximo capítulo assim como a proposta de algumas modificações no tabuleiro de corrida.

Figura 16 – Exemplo de plotagem em tempo real de um paciente no computador



Fonte: Elaborado pelo autor.

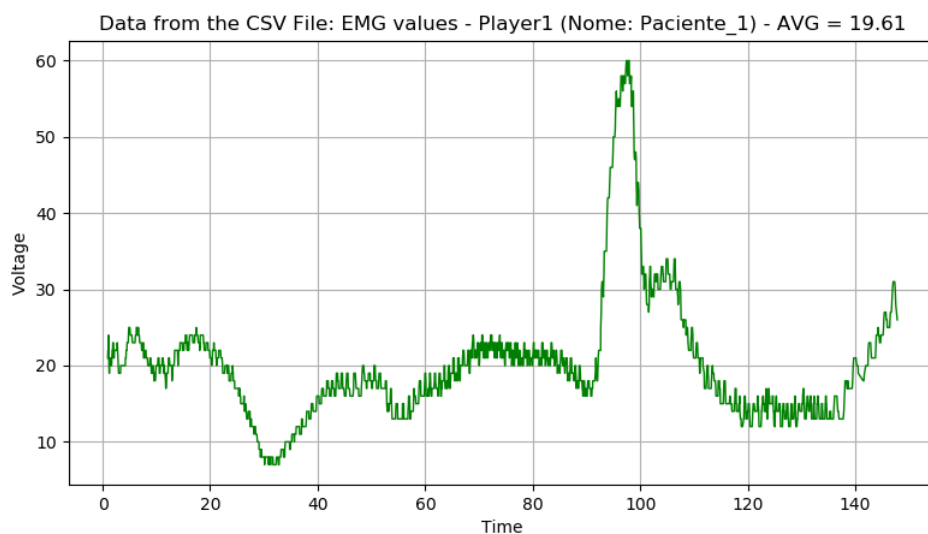
## 2.4 Acesso dos sinais EMG pelo *Datalogger* do sistema

Conforme o sinal vai sendo exibido na tela do computador, o mesmo script em Python responsável pelo plot do sinal em tempo real também armazena os dados coletados em diferentes arquivos CSV. Como são 4 pacientes diferentes, foi considerado 4 arquivos individuais de backup para eventuais consultas posteriores. Esses arquivos são acessados e plotados por outro script em Python por meio da biblioteca Matplotlib. Como os dados são salvos nos arquivos CSVs durante a realização da competição, então não é necessário

utilizar o PyQtGraph para esta tarefa uma vez que os dados já encontram-se armazenados. O profissional responsável pode consultar estes dados individualmente e salvá-los em seu prontuário com o nome individual de cada paciente.

A Figura 17 representa o resultado do desempenho individual do paciente 1 logo após realizar a competição. Nela, percebe-se o trecho representado na Figura 16 referente ao gráfico de tempo real. Apenas para critérios de simulação, a média de amplitudes também foi colocada com a sigla AVG para termos uma noção de quem ganhou a corrida durante o tempo medido.

Figura 17 – Recuperação dos dados coletados do paciente após a competição



Fonte: Elaborado pelo autor.

Diante do que foi exposto, percebe-se que o sistema conta com uma medição e tratamento adequado dos dados musculares ao longo do tempo. Além disso, o sistema também conta com ferramentas adequadas para representação dos sinais em tempo real assim como um *datalogger* para armazenamento destes dados. Após a etapa de instrumentação e armazenamento dos dados, agora é o momento de propor adaptações ao tabuleiro de corrida. No próximo capítulo serão discutidas algumas propostas de melhoria à plataforma de controle dos carrinhos de brinquedo e o algoritmo em malha aberta desenvolvido para controle dos mesmos por meio de sinais PWM.

Para isso, também é proposta a inclusão de motores DC de 12 Volts individualmente a cada carrinho para que estes se movam por uma fenda individual semelhante a um jogo de Autorama analógico. A pista, por sua vez, possuirá pequenos trilhos metálicos (fita adesiva de cobre) alimentados por uma fonte de tensão DC de 12 Volts conectado à rede elétrica à fim de fechar o circuito quando o PWM for acionado. Dessa forma os carrinhos se movem proporcionalmente à força muscular imposta pelos pacientes.

## 3 Plataforma de controle para o carrinhos

Neste capítulo serão discutidas algumas propostas e melhorias para o funcionamento do tabuleiro e carrinhos de corrida. Também será apresentado o algoritmo de controle desenvolvido para movimentação dos carrinhos.

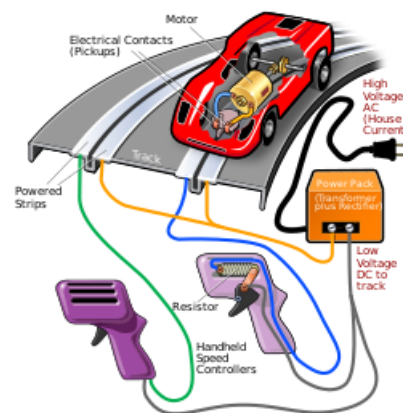
### 3.1 Automação do tabuleiro de corrida

Para que os pacientes não se movam durante a movimentação dos carrinhos, foi proposta a utilização de um modelo similar ao utilizado em Autoramas analógicos. Isso garante que os carrinhos se movimentem livremente pela pista sem a conexão de fios conectados diretamente. Para isso, é necessário entender primeiramente como um autorama analógico funciona e como este poderia nos ajudar a automatizar o nosso sistema da melhor forma possível.

#### 3.1.1 Funcionamento de um Autorama analógico (*Slot Car*)

O Autorama ou Automobilismo de fenda é um modelismo elétrico de carrinhos de corrida em miniaturas de pistas especialmente fabricadas para o esporte ou entretenimento. Esse tipo de diversão foi bastante difundida no Brasil ao longo da década de 70 e que possui um forte adepto até os dias de hoje [12]. A Figura 18 representa o funcionamento básico de um autorama analógico.

Figura 18 – Modelo de funcionamento de um Autorama analógico



Fonte: Retirado de [12].

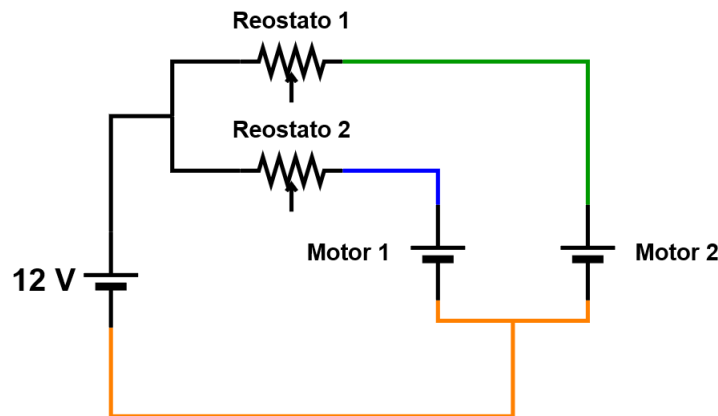
Primeiramente, o tabuleiro de corrida é separado por pistas. Para cada pista um carro movimenta-se por uma fenda entre dois trilhos metálicos. Um trilho, por sua vez, é alimentado por uma tensão de referência (0 Volts) externa, enquanto que o outro, é conectado a dispositivos manuais de controle. Esses dispositivos em formato de pistola, por sua vez, são alimentados com uma tensão contínua geralmente de 12 Volts [12].

Cada dispositivo de controle funciona por meio de um reostato regulado manualmente por uma espécie de gatilho. Conforme o gatilho se movimenta, o reostato varia sua resistência por meio do seu comprimento [12]. Conforme a 2ª Lei de Ohm, representada pela equação 3.1, conforme o comprimento de um reostato varia, a sua resistência também varia proporcionalmente [6]. De acordo com a 1ª Lei de Ohm, sabemos que a resistência e a corrente são inversamente proporcionais. Dessa forma, o competidor, ajustando a resistência do dispositivo, aumenta ou diminui a corrente elétrica que alimenta os trilhos onde os carros disputam a corrida. Cada carro possui um contato elétrico entre as trilhas e o motor pela peça denominada de *pickup* conforme demonstrado na Figura 18 [12].

$$R = \frac{\rho L}{A} \quad (3.1)$$

Cada carrinho, por sua vez, possui um motor DC acoplado que movimenta as suas rodas traseiras conforme a intensidade de corrente elétrica liberada pelo reostato. Essa corrente alimenta o outro trilho do carrinho, fechando o circuito completamente e girando o motor DC. Quanto mais o competidor aperta o gatilho, menor o reostato fica e maior é a corrente elétrica liberada para a trilha, variando a velocidade de rotação do motor do respectivo carrinho [5]. A Figura 19 representa o circuito elétrico da Figura 18. Cada reostato é conectado a uma trilha que possui um motor DC que se movimenta proporcionalmente à corrente emitida por cada dispositivo de controle. Cada trilha é conectada em paralelo à fonte de alimentação.

Figura 19 – Circuito elétrico da pista de Autorama



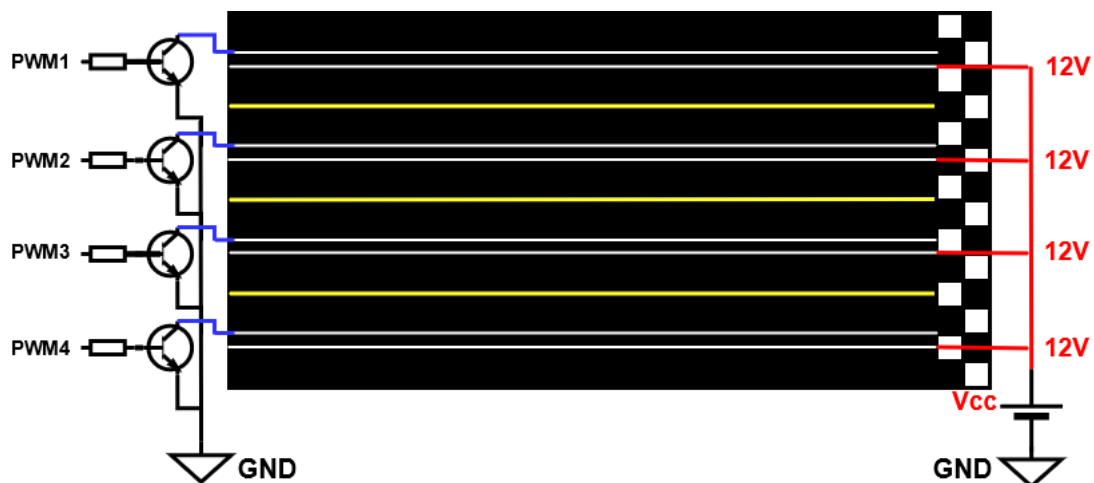
Fonte: Elaborado pelo autor.

### 3.1.2 Adição de fendas e trilhas metálicas no tabuleiro

Como foi discutido, o modelo de movimentação dos carrinhos pelo Autorama é feito de forma muito elegante. Através dessa característica é proposto o mesmo modelo de trilhos e fendas para cada carrinho uma vez que os sinais de controle trafegam nos trilhos metálicos da pista. Cada carrinho terá sua própria fenda e dois trilhos. Um trilho será alimentado com 12 Volts de uma fonte DC contínua enquanto que o outro trilho estará conectado nos pinos de controle por PWM do microcontrolador.

A Figura 20 representa o processo de automação proposto. As demarcações em amarelo são as divisórias das pistas de cada carrinho. As linhas brancas são os trilhos metálicos na qual circula o sinal PWM do microcontrolador (ligação de cor azul) e a alimentação da fonte DC (ligação de cor vermelha). Entre as duas linhas brancas de cada pista encontra-se uma fenda para que cada carro acople os contatos elétricos do seu motor nos trilhos por meio de uma peça similar a *pickup* da Figura 18.

Figura 20 – Esquemático de automação do tabuleiro de corrida



Fonte: Elaborado pelo autor.

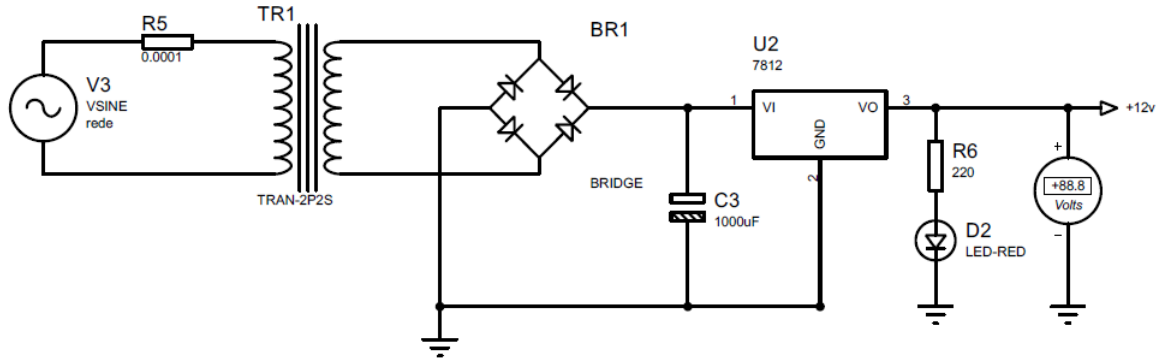
Enquanto a fonte de tensão DC alimenta as 4 trilhas dos 4 carrinhos, o microcontrolador conecta suas 4 saídas PWM nas outras 4 trilhas do tabuleiro. Essa conexão fecha o circuito quando o transistor Darlington polariza a sua base por meio de um sinal PWM. Dessa forma, uma corrente regulada flui pelas trilhas do tabuleiro movimentando o motor DC do respectivo carrinho através do coletor do transistor.

O algoritmo de controle será explicado com mais detalhes na próxima seção uma vez que o algoritmo proposto regula a corrente elétrica de forma similar ao Autorama analógico. Neste trabalho, ao invés do controle ser feito de forma manual, é realizado um controle automático de acordo com a amplitude do sinal EMG recebido. No próximo tópico é discutido o projeto da construção de uma fonte DC de 12 Volts para a pista.

### 3.1.3 Projeto de fonte DC para alimentação da pista

Para alimentação das trilhas do tabuleiro foi proposto a construção de uma fonte DC que transforma a tensão da rede elétrica alternada de 220 Volts para tensão contínua de 12 Volts. O projeto da fonte foi realizado no Software Proteus conforme representado na Figura 21.

Figura 21 – Esquemático da fonte de tensão DC projetada para o tabuleiro



Fonte: Elaborado pelo autor.

Para a construção da fonte DC de tensão foi necessário a utilização de um pequeno transformador 220V/12V, diodos, capacitor eletrolítico e um regulador de tensão LM7812. Primeiramente, a tensão alternada da rede elétrica é reduzida de acordo com a relação entre a tensão e a indutância das bobinas primárias  $V_1$  e  $L_1$  e secundárias  $V_2$  e  $L_2$  pela relação:

$$\left(\frac{V_1}{V_2}\right)^2 = \frac{L_1}{L_2} \quad (3.2)$$

Após a redução de tensão nas bobinas, é necessário retificar a onda senoidal completamente por meio de uma ponte de diodos conforme mostra a Figura 21. Após a onda ser retificada, o sinal que antes era alternado agora passa a ser contínuo. Para que um nível DC seja extraído da onda retificada, é necessário a utilização de um capacitor eletrolítico na saída do retificador de onda completa.

Para que o nível DC de 12 Volts tenha uma saída constante é necessário um regulador de tensão LM7812 conectado no capacitor eletrolítico. O regulador de tensão LM7812 pode ser usado em fontes de alimentação, carregadores e circuitos em geral, fornecendo uma tensão fixa de 12V na saída [5]. Essa tensão alimenta todas as 4 trilhas do tabuleiro de corrida constantemente. Em seguida, no circuito, temos um LED de funcionamento da fonte juntamente com um leitor de tensão na sua saída. No próximo tópico é discutido uma leve modificação nos carrinhos de brinquedo.

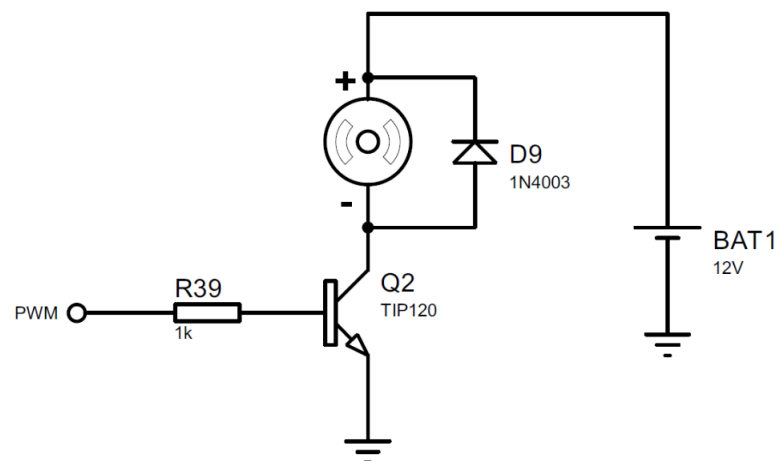


### 3.1.4 Adição de motor DC nos carrinhos de brinquedo

Para que os sinais de controle percorram pelas trilhas, é necessário que o circuito seja fechado por um motor de corrente contínua. Basicamente, para que um motor DC funcione corretamente ligado a um microcontrolador, é necessário que este esteja separadamente conectado a uma fonte externa de 12 Volts. Atrelado a este fator, é necessário que o sinal do microcontrolador seja amplificado por um transistor para que o motor seja acionado.

Tal configuração é representada na Figura 22. O pino negativo e positivo do motor deve estar ligado no coletor do transistor e polo positivo da bateria, respectivamente. Conforme o sinal PWM polariza a base do transistor, uma corrente é produzida no sentido coletor-emissor, fechando o circuito. Como forma de proteção do circuito, um diodo roda-livre é colocado em paralelo com a carga para que não haja retorno de corrente para a porta do microcontrolador.

Figura 22 – Ligação entre motor DC e microcontrolador

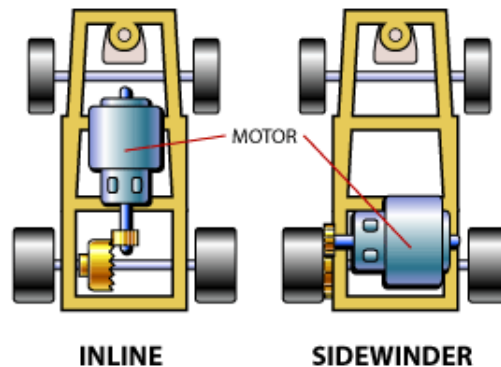


Fonte: Elaborado pelo autor.

A configuração acima justifica as ligações feitas no tabuleiro de corrida da Figura 20. Para que o motor rotacione, é necessário que uma corrente emissor-coletor seja produzida. Por este motivo, é necessário a adição interna de motores em cada carrinho. No Autorama analógico, existem diversas configurações de carrinhos de corrida. A principal delas, conhecida como *Inline*, fixa o motor DC no centro do chassi do carro por meio de uma conexão direta com as rodas traseiras do carrinho como representado na Figura 23.

Para a realização desse trabalho, foi proposta a configuração *Inline* para cada um dos carros envolvidos na competição. Essa configuração permite, por sua simplicidade, manter uma conexão direta dos terminais do motor com os trilhos do tabuleiro e movimentar adequadamente os carrinhos por meio de um sinal de controle configurado

Figura 23 – Modelo de configuração proposto para os carrinhos do sistema



Fonte: Retirado de [12].

no microcontrolador. Na próxima seção, será discutido sobre o algoritmo de controle de velocidade dos carrinhos por meio de sinais PWM.

## 3.2 Algoritmo de Controle *Feedforward*

O algoritmo de controle proposto foi desenvolvido no próprio microcontrolador a partir dos sinais EMG coletados pelo biossensor desenvolvido. A ideia básica é aumentar a velocidade do carrinho à medida que o sinal EMG aumenta sua amplitude. Para realizar esse tipo de algoritmo, é importante perceber que o controle do sistema atua em malha aberta, ou seja, não possui realimentação.

No sistema de controle em questão não estamos interessados que um erro ocorra para que o controle atue diretamente. Queremos que um controle simples atue em *feedforward* para aumentar ou diminuir a rotação dos motores dos carrinhos de acordo com os dados lidos pelo conversor A/D do microcontrolador. Tal controle possui forte dependência temporal com a amplitude dos sinais musculares lidos ao decorrer do tempo, indicando por sua vez, um problema clássico de controle em malha aberta. A seguir será explicado o funcionamento básico do PWM para o controle proposto.

### 3.2.1 Modulação por Largura de Pulso (PWM)

O sinal de PWM é utilizado para controlar a potência média de uma carga em corrente contínua [6]. Com o PWM é possível controlar a tensão e corrente fornecidas a uma carga ao ligar e desligar (chavear) o fornecimento de energia entre a fonte e a carga em uma taxa muito rápida [5]. Quanto mais tempo a alimentação permanece ligada, em comparação com o tempo desligada, maior a quantidade total de potência fornecida à

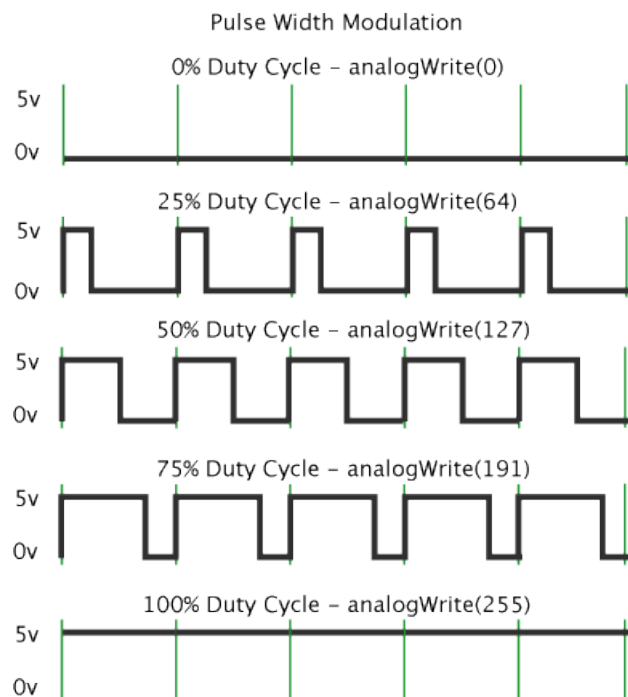
carga. Assim, se quisermos, por exemplo, fazer um motor girar com, digamos, 75% de sua velocidade máxima, podemos conseguir com o uso do PWM [9].

O PWM funciona modulando o ciclo ativo (duty cycle) de uma onda quadrada. O conceito de funcionamento é simples [6]. O controlador (fonte de tensão com PWM) entrega uma série de pulsos, gerados em intervalos de igual duração, que pode ser variada. Quanto mais largo o pulso, maior a quantidade de corrente fornecida à carga [5].

O duty cycle diz respeito à proporção de tempo com fornecimento de energia (on) em relação ao período de tempo sem energia (off). Um duty cycle baixo significa pouca potência fornecida, pois a energia está desligada a maior parte do tempo [2]. Expressamos o duty cycle em porcentagem, sendo que o valor 100% significa “totalmente ligado” e 0%, “totalmente desligado”.

O microcontrolador ATmega2560 possui 15 pinos para saída PWM de 8 bits com uma frequência de 480  $Hz$  [11]. Para auxiliar na manipulação desses pinos a plataforma possui uma função que auxilia na escrita de valores de duty cycle. O valor que deve ser repassado para esta função deve ser de 0 a 255 onde com 0 a saída permanece sempre em nível baixo e 255 a saída permanece sempre em nível alto. Um valor intermédio equivale a valores de duty cycle entre 0 e 100%. A Figura 24 representa o funcionamento do duty cycle nos pinos do microcontrolador por meio da função **analogWrite**.

Figura 24 – Representação de modulações PWM (Duty Cycle)



Fonte: Retirado de [11].

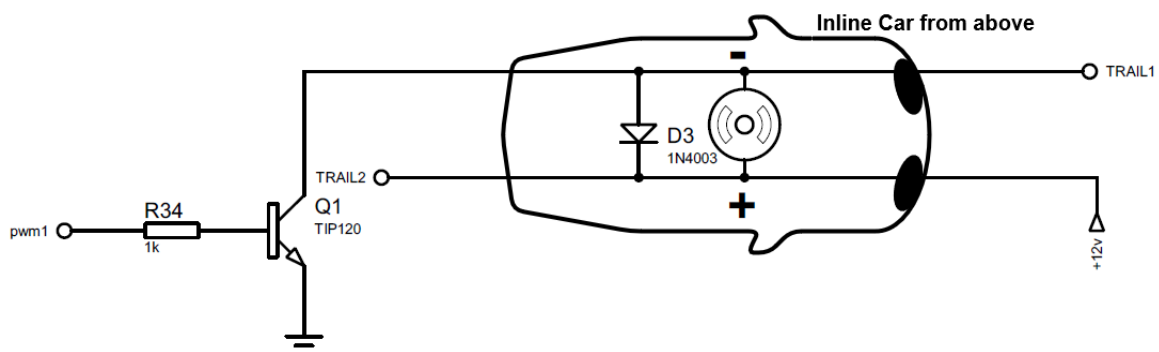
Tendo em mente que a variação da largura do pulso varia a corrente elétrica do pino PWM, então a configuração da Figura 22 deve ser utilizada para controlar cada um dos motores dos carrinhos de cada paciente. Dependendo do RPM nominal do motor (quantidade de giros máximos quando ligado a 12 Volts) e do duty cycle, temos como variar a sua rotação de acordo com o valor do ciclo de trabalho configurado.

Tendo como controlar a rotação dos motores através do PWM, podemos construir um Autorama modificado para o sistema voltado para pacientes com AVE. Ao invés de realizar o controle manual, o controle é feito automaticamente pela variação do duty cycle de cada pino PWM proporcionalmente à amplitude do sinal EMG recebido pelo conversor A/D. De acordo com a tensão traduzida do sinal muscular pelo conversor analógico-digital podemos acionar um sinal na base do transistor para modificar a velocidade do motor proporcional a este valor.

### 3.2.2 Conexão do sinal de controle PWM com o carrinho

A Figura 25 representa a ligação do sinal PWM de controle com as trilhas do carrinho. Nessa figura, percebe-se o carrinho de modelo *Inline* visto de cima e o seu motor interno conectado sobre as trilhas. Conforme a base do transistor é polarizada pelo sinal de PWM, uma corrente elétrica no sentido coletor-emissor ocorre, movimentando o carro de corrida. Quanto maior a amplitude do sinal EMG (tensão lida pelo conversor A/D), maior o duty cycle do PWM (maior corrente no coletor) e maior velocidade de rotação do motor. Essa configuração permite que o carrinho se desloque semelhantemente a um carrinho de Autorama analógico ao longo de sua trajetória.

Figura 25 – Representação da ligação de controle do PWM com o carrinho



Fonte: Elaborado pelo autor.

O PWM, por se tratar de uma modulação, encarrega-se de transportar a informação através de seu duty cycle a partir dos valores lidos dos biossensores desenvolvidos por valores de tensão. A seguir será discutido sobre o código de controle desenvolvido.

### 3.2.3 Código de Controle: Implementação

O código feito para controle de rotação dos motores se encontra no Apêndice A deste projeto. O primeiro trecho de código representa um algoritmo de controle em malha aberta tendo os valores analógicos provenientes dos biossensores como entrada. O conversor A/D do microcontrolador traduz os valores de tensão analógicos dos biossensores em valores discretos por um processo conhecido como quantização e digitalização [6].

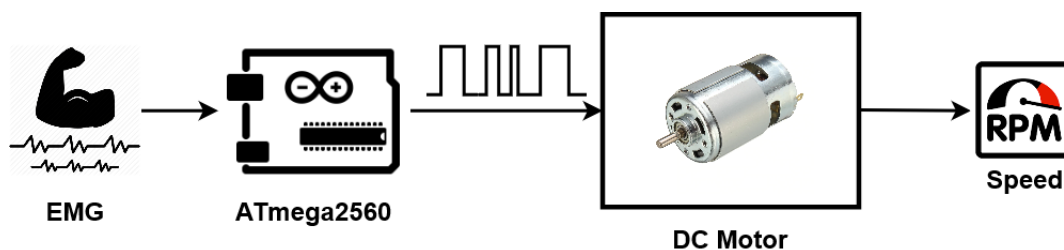
Cada valor analógico de tensão é representado entre 0 e 1024, faixa essa que corresponde aos 10 bits de resolução dos canais A/D do sistema embarcado. Quanto mais próximo o valor de tensão é de 5 Volts, mais próximo o valor discreto analógico correspondente fica de 1024. Quanto mais próximo o valor de tensão é de 0 Volts, mais próximo o valor discreto analógico correspondente fica em 0.

Por meio dessa característica, é possível mensurar em valores discretos os valores de tensão dos sinais musculares fornecidos pelo biossensor. A ideia básica é transmitir esse valor correspondente discreto por meio dos pinos PWM do microcontrolador. O valor de tensão discretizado, por sua vez, equivale ao tempo ativo da onda (duty cycle) PWM que polariza a base do transistor.

Para que o valor lido pelo conversor A/D tenha a mesma proporcionalidade do duty cycle do PWM, é necessário realizar uma normalização entre os canais. Enquanto que o conversor A/D do ATmega2560 possui 10 bits de resolução, o seu canal PWM possui apenas 8. Para compensar essa diferença, basta dividir os valores lidos dos biossensores por 4 e jogar diretamente na função **analogWrite**.

Cada valor lido e normalizado será o equivalente a uma porcentagem do valor de duty cycle da onda de controle. Quanto maior for a tensão lida (mais contração muscular), maior o duty cycle do PWM (mais corrente no coletor) e mais rápido o carro se movimentará. A Figura 26 representa um fluxo de dados para controle de rotação dos motores em malha aberta. Conforme os dados são recebidos, estes são normalizados pelo microcontrolador e jogados nas trilhas por meio do PWM para o controle de velocidade dos carrinhos.

Figura 26 – Controle de velocidade do motor em malha aberta



Fonte: Elaborado pelo autor.

### 3.3 Indicativo visual para o ganhador da competição

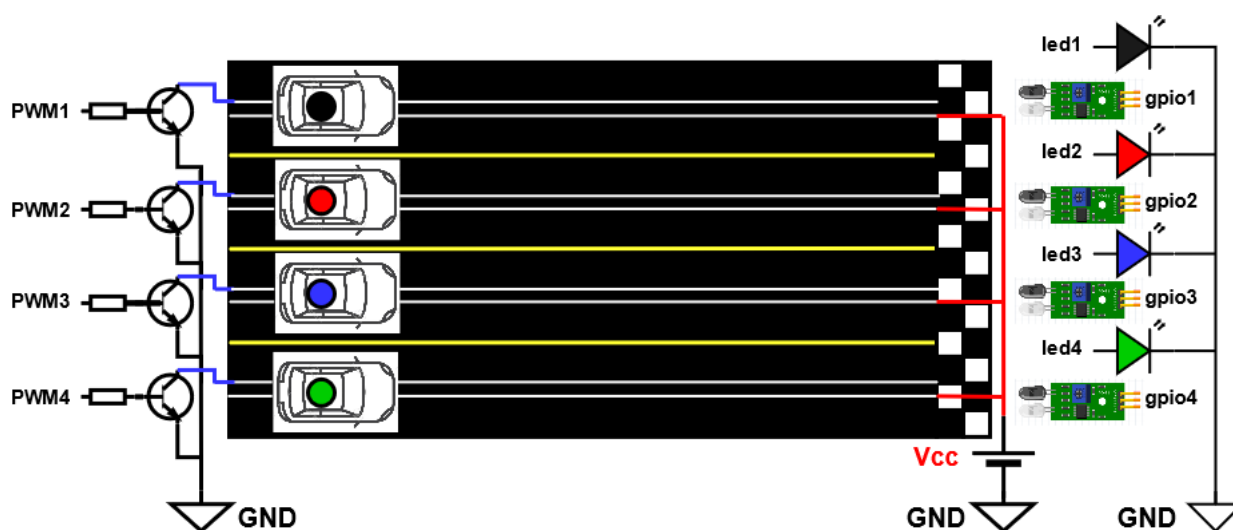
Como último requisito obrigatório foi pedido um indicativo visual para o primeiro carrinho que ultrapassar a linha de chegada. Para isso foram utilizados 4 sensores de obstáculos IR colocados prontamente no final de cada pista para a detecção do vencedor quando este tocar a linha de chegada. O sensor de obstáculo é um circuito composto por um emissor e um receptor IR, mais o CI comparador LM393, que facilita sua conexão com o ATmega2560. Esse sensor é ajustável por meio de um trimpote entre 0 a 30 *cm*.

Cada sensor IR é conectado ao microcontrolador por fios e são programados por meio de interrupções. Quando um carro toca a linha de chegada, o microcontrolador dispara uma interrupção acendendo o LED do respectivo ganhador. Automaticamente, quando isso acontece, nenhum LED pode ser mais acionado pois o carrinho vencedor já foi detectado. Esse tipo de indicação visual apesar de simples é muito útil para identificar o ganhador uma vez que o sensor IR funciona de forma digital.

A Figura 27 demonstra um esquemático completo do tabuleiro de corrida juntamente com os carrinhos e o sistema de identificação do vencedor. Para cada carrinho, existe um sensor IR calibrado no final da pista para detectar sua possível vitória. Cada sensor e LED de identificação é conectado por uma porta de GPIO simples. Todos os 4 sensores são programados por interrupção e os 4 LEDs são conectados por fios com o microcontrolador.

Para uma visualização mais detalhada dos circuitos apresentados, por favor, verificar os esquemáticos completos elaborados no repositório do projeto. O próximo capítulo trata da lista de componentes do sistema, orçamento e possíveis fornecedores.

Figura 27 – Configuração completa do tabuleiro de corrida



Fonte: Elaborado pelo autor.

## 4 Orçamento do sistema proposto

A Tabela 1 demonstra detalhadamente o orçamento dos materiais necessários para a construção do sistema proposto. Os preços foram baseados em fornecedores nacionais e internacionais de acordo com a cotação do dólar no dia 09/03/2021 (1\$ = 5,85 R\$). Essa mesma tabela equivale a lista de materiais/equipamentos utilizados no projeto. De acordo com o orçamento feito, o total do projeto custa em torno de **R\$ 254,6** podendo ser barateado com a troca do ATmega2560 por um ATmega328P. O próximo capítulo sugere outra tabela com os fornecedores dos equipamentos com os respectivos sites para consulta.

Tabela 1 – Lista de Materiais e Orçamento do projeto

Materiais	Quantidade	Preço Unitário	Preço Total
ATMega2560	1	R\$ 31,32	R\$ 31,32
Cristal de Quartzo 16 <i>MHz</i>	1	R\$ 0,94	R\$ 0,94
Eletrodos de Superfície Neuroline	12	R\$ 7,41	R\$ 88,92
Resistor 100 <i>kΩ</i>	25	R\$ 0,06	R\$ 1,50
Resistor 10 <i>kΩ</i>	10	R\$ 0,06	R\$ 0,60
Resistor 1 <i>kΩ</i>	15	R\$ 0,06	R\$ 0,90
Resistor 220 $\Omega$	10	R\$ 0,14	R\$ 1,40
Capacitor eletrolítico 22 <i>pF</i>	2	R\$ 0,11	R\$ 0,22
Capacitor eletrolítico 100 <i>pF</i>	20	R\$ 0,43	R\$ 8,60
Capacitor eletrolítico 500 $\mu F$	10	R\$ 0,14	R\$ 1,40
Diodos 1N4003	25	R\$ 0,07	R\$ 1,75
Diodos Zener 1N4733A	4	R\$ 0,10	R\$ 0,40
Amplificador operacional LM741	8	R\$ 1,57	R\$ 12,56
Transistor TIP120	4	R\$ 1,76	R\$ 7,04
LEDs alto brilho 10 <i>mm</i>	5	R\$ 0,64	R\$ 3,20
Regulador de Tensão LM7812	1	R\$ 1,20	R\$ 1,20
Motor DC 12V	4	R\$ 5,32	R\$ 21,28
Bateria 9V	2	R\$ 3,56	R\$ 7,12
Transformador 220V/12V	1	R\$ 8,89	R\$ 8,89
Sensores IR	4	R\$ 2,25	R\$ 9,00
Fita adesiva de cobre 10 <i>m</i>	1	R\$ 8,64	R\$ 8,64
Engrenagens plásticas	10	R\$ 0,30	R\$ 3,00
Placa PCB de Fenolite (5 <i>cm</i> x 7 <i>cm</i> )	5	R\$ 1,55	R\$ 7,75
Percloroeto de Ferro 500 <i>ml</i>	1	R\$ 17,23	R\$ 17,23
Jumpers M-F 20 <i>cm</i>	20	R\$ 0,22	R\$ 4,40
Cabo USB serial	1	R\$ 5,34	R\$ 5,34
<b>TOTAL:</b>			<b>R\$ 254,6</b>

Fonte: Elaborado pelo autor.

## 5 Lista de Fornecedores

A Tabela 2 fornece os possíveis fornecedores de equipamentos para a construção completa do sistema. A escolha por fornecedores nacionais e internacionais facilitam na redução de custos com o projeto. Ao lado de cada fornecedor está o link de acesso para os respectivos equipamentos.

No apêndice D estão os modelos PCB desenvolvidos dos 4 biossensores conectados em uma mesma placa embarcada como proposta para o sistema de reabilitação de pacientes com AVE. Essa mesma PCB representa uma plataforma de aquisição e tratamento dos dados musculares recebidos assim como pede o documento de orientação do projeto.

Tabela 2 – Lista de Fornecedores Nacionais e Internacionais

<b>Materiais</b>	<b>Fornecedores</b>	<b>Site da Loja</b>
ATMega2560	Aliexpress Chip Store	<a href="#">Link 1</a>
Cristal de Quartzo 16 <i>MHz</i>	Hu Infinito	<a href="#">Link 2</a>
Eletrodos de Superfície Neuroline	Kandel	<a href="#">Link 3</a>
Resistor 100 <i>kΩ</i>	Baú da Eletrônica	<a href="#">Link 4</a>
Resistor 10 <i>kΩ</i>	Baú da Eletrônica	<a href="#">Link 5</a>
Resistor 1 <i>kΩ</i>	Baú da Eletrônica	<a href="#">Link 6</a>
Resistor 220 $\Omega$	Baú da Eletrônica	<a href="#">Link 7</a>
Capacitor eletrolítico 22 <i>pF</i>	Baú da Eletrônica	<a href="#">Link 8</a>
Capacitor eletrolítico 100 <i>pF</i>	Baú da Eletrônica	<a href="#">Link 9</a>
Capacitor eletrolítico 500 $\mu F$	Baú da Eletrônica	<a href="#">Link 10</a>
Diodos 1N4003	Chinese Super Eletronic	<a href="#">Link 11</a>
Diodos Zener 1N4733A	Letron Chip Store	<a href="#">Link 12</a>
Amplificador operacional LM741	Baú da Eletrônica	<a href="#">Link 13</a>
Transistor TIP120	Baú da Eletrônica	<a href="#">Link 14</a>
LEDs alto brilho 10 <i>mm</i>	Baú da Eletrônica	<a href="#">Link 15</a>
Regulador de Tensão LM7812	Smart Kits	<a href="#">Link 16</a>
Motor DC 12V	Parts Hub Store	<a href="#">Link 17</a>
Bateria 9V	Baú da Eletrônica	<a href="#">Link 18</a>
Transformador 220V/12V	Shop Store	<a href="#">Link 19</a>
Sensores IR	RoarKit Store	<a href="#">Link 20</a>
Fita adesiva de cobre 10 <i>m</i>	Nicelux Store	<a href="#">Link 21</a>
Engrenagens plásticas	Factorysale Store	<a href="#">Link 22</a>
Placa PCB de Fenolite (5 <i>cm</i> x 7 <i>cm</i> )	Shenzhen IC Store	<a href="#">Link 23</a>
Percloroeto de Ferro 500 <i>mL</i>	Baú da Eletrônica	<a href="#">Link 24</a>
Jumpers M-F 20 <i>cm</i>	Baú da Eletrônica	<a href="#">Link 25</a>
Cabo USB serial	Acelex Store	<a href="#">Link 26</a>

Fonte: Elaborado pelo autor.



# Referências

- 1 GRAUPE, D.; CLINE, W. K. Functional separation of emg signals via arma identification methods for prosthesis control purposes. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, n. 2, p. 252–259, 1975. Citado 2 vezes nas páginas 10 e 19.
- 2 JAMAL, M. Z. Signal acquisition using surface emg and circuit design considerations for robotic prosthesis. *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*, InTech, v. 18, p. 427–448, 2012. Citado 4 vezes nas páginas 10, 11, 17 e 34.
- 3 MACVIEIRA, T. G. Surface electromyography: Why, when and how to use it. *Revista Andaluza Medicina Deporte*, v. 4, n. 1, p. 17–28, 2011. Citado 3 vezes nas páginas 10, 12 e 17.
- 4 FARINA, D.; MERLETTI, R.; ENOKA, R. M. The extraction of neural strategies from the surface emg. *Journal of applied physiology*, American Physiological Society, v. 96, n. 4, p. 1486–1495, 2004. Citado 2 vezes nas páginas 10 e 17.
- 5 BOYLESTAD, R. L.; NASHELSKY, L. *Electronic devices and circuit theory*. [S.l.]: Pearson, 2014. Citado 7 vezes nas páginas 12, 14, 16, 29, 31, 33 e 34.
- 6 OPPENHEIM, A. V.; BUCK, J. R.; SCHAFER, R. W. *Discrete-time signal processing. Vol. 2*. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 2001. Citado 9 vezes nas páginas 14, 16, 17, 19, 20, 29, 33, 34 e 36.
- 7 FUKUDA, O. et al. A human-assisting manipulator teleoperated by emg signals and arm motions. *IEEE transactions on robotics and automation*, IEEE, v. 19, n. 2, p. 210–222, 2003. Citado na página 18.
- 8 LIPPS, C. et al. Proof of concept for iot device authentication based on sram pufs using atmega 2560-mcu. In: IEEE. *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. [S.l.], 2018. p. 36–42. Citado 2 vezes nas páginas 20 e 24.
- 9 KARL, H.; WILLIG, A. *Protocols and architectures for wireless sensor networks*. [S.l.]: John Wiley & Sons, 2007. Citado 2 vezes nas páginas 24 e 34.
- 10 FRENZEL, L. E. *Handbook of serial communications interfaces: a comprehensive compendium of serial digital input/output (I/O) standards*. [S.l.]: Newnes, 2015. Citado na página 24.
- 11 WARREN, J.-D.; ADAMS, J.; MOLLE, H. Arduino for robotics. In: *Arduino robotics*. [S.l.]: Springer, 2011. p. 51–82. Citado 2 vezes nas páginas 24 e 34.
- 12 KANE, S.; SCOTT, J. B. The slot car stig: Performance and consistency of a slot car driven by a heuristic algorithm in an embedded microcontroller. In: ELECTRONICS RESEARCH, DEPARTMENT OF PHYSICS, UNIVERSITY OF OTAGO. *The 16th Electronics New Zealand Conference*. [S.l.], 2009. Citado 3 vezes nas páginas 28, 29 e 33.

# APÊNDICE A – Códigos implementados

Listing A.1 – Filtro Média móvel para sensores EMG

---

```

1  #include <stdint.h>
2  #include <Arduino.h>
3  #define N 100          // Number of points of the moving average
4
5  int valuesAvg1[N];    // Vector to store the moving average output
6  long moving_average(uint8_t sensorValue, uint8_t *number);
7
8  int main()
9  {
10     // Normalize data (1024/256 = 4) – ADC (10 bits) to PWM (8 bits)
11     float rawSensor1 = analogRead(EMG1) / 4;
12     // Calls moving average function
13     float dataFiltered1 = moving_average(rawSensor1, valuesAvg1);
14     // Send the filtered data through the COM1 serial
15     Serial.print(rawSensor1); // Send raw data first
16     Serial.print(","); // separate data by comma
17     Serial.println(dataFiltered1); // send full string
18     // Triggers PWM with read analog value
19     digitalWrite(PWM1, dataFiltered1);
20 }
21
22 long moving_average(uint8_t sensorValue, uint8_t *number)
23 {
24     // Accumulator to add the points of the moving average
25     uint32_t acc = 0;
26     // Shifts the elements of the moving average vector
27     for(int i = N - 1; i > 0; i--) number[i] = number[i-1];
28     // Initial position of the vector receives the original reading
29     number[0] = sensorValue;
30     // Sum the number of points
31     for(int i = 0; i < N; i++) acc += number[i];
32     // Returns the moving average
33     return acc/N;
34 }

```

---

Listing A.2 – Script para coleta e plotagem de dados em tempo real

---

```

1 from numpy import *
2 from pyqtgraph.Qt import QtGui, QtCore
3 import pyqtgraph as pg
4 import serial
5 import time
6 import csv
7
8 start_time = time.time()    # Start timer
9 portName = "COM6"         # Create object serial port
10 baudrate = 115200         # Baudrate
11 ser = serial.Serial(portName, baudrate)
12 app = QtGui.QApplication([])    # START QtApp
13 win = pg.GraphicsWindow(title="Signal from serial port")
14 p = win.addPlot(title="Realtime plot")
15 curve = p.plot(pen=(255,0,0), name="Red X curve")
16 windowWidth = 500         # width of the window displaying the curve
17 Xm = linspace(0,0,windowWidth) # array with relevant time series
18 ptr = -windowWidth        # set first x position
19
20 def update():    # Realtime data plot
21     global curve, ptr, Xm
22     Xm[:-1] = Xm[1:]    # shift data
23     value = ser.readline() # read line from the serial port
24     Xm[-1] = float(value) # vector with instantaneous values
25     ptr += 1    # update x position for displaying the curve
26     curve.setData(Xm) # set the curve with this data
27     curve.setPos(ptr,0) # set x position in the graph to 0
28     QtGui.QApplication.processEvents() # process the plot now
29
30     # save the serial data in real-time to a .csv file (Datalogger)
31     decoded_bytes = float(value[0:len(value)-2].decode("utf-8"))
32     with open("dataPlayer1.csv", "a") as f:
33         writer = csv.writer(f, delimiter=",")
34         writer.writerow([decoded_bytes, (time.time() - start_time)])
35
36 while True: update()    # Main program
37 pg.QtGui.QApplication.exec_()
38 }

```

---

Listing A.3 – Script para plotagem dos sinais após a competição

---

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import sys
4
5 # Amplitude average function
6 def average_column (csv):
7     list_num = []
8     with open (csv, "rb") as annotate:
9         for col in annotate:
10             ann = col.decode().split(",")
11             list_num.append(float(ann[0]))
12     avg = sum(list_num) / len(list_num)
13     return avg
14
15 if len(sys.argv) <= 1: # Receives patient's name
16     print('Insufficient arguments')
17     sys.exit()
18
19 # Performs the average amplitude of the EMG signal
20 amp_avg = average_column('dataPlayer1.csv')
21
22 # Separate data separated by ',' in CSV
23 y, x = np.loadtxt('dataPlayer1.csv', delimiter=',', unpack=True)
24
25 # Plot patient data
26 plt.plot(x, y, 'g-', linewidth=1.0, label='EMG filtered')
27
28 plt.title('Data from the CSV File: EMG values - \\
29           Player1 (Nome: {}) - AVG = {:.2f}'.format(sys.argv[1], amp_avg))
30
31 plt.xlabel('Time')
32 plt.ylabel('Voltage')
33
34 plt.grid()
35 plt.show()
```

---

## APÊNDICE B – Memorial de Cálculos

Divisor de Tensão

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (\text{B.1})$$

Amplificador subtrator

$$V_{out} = \frac{V_A \left(1 + \frac{R_4}{R_3}\right)}{\left(1 + \frac{R_1}{R_2}\right)} - V_B \frac{R_4}{R_3} \quad (\text{B.2})$$

Amplificador não-inversor

$$V_{out} = \left(\frac{R_2}{R_1} + 1\right) V_{in} \quad (\text{B.3})$$

Frequência de Corte

$$F_c = \frac{1}{2\pi RC} \quad (\text{B.4})$$

Filtro média móvel discreto

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (\text{B.5})$$

1ª Lei de Ohm

$$V = R i \quad (\text{B.6})$$

2ª Lei de Ohm

$$R = \frac{\rho L}{A} \quad (\text{B.7})$$

Equação do transformador

$$\left(\frac{V_1}{V_2}\right)^2 = \frac{L_1}{L_2} \quad (\text{B.8})$$

Corrente de coletor no Transistor

$$I_c = \beta I_b \quad (\text{B.9})$$

Cálculo do *Duty Cycle*

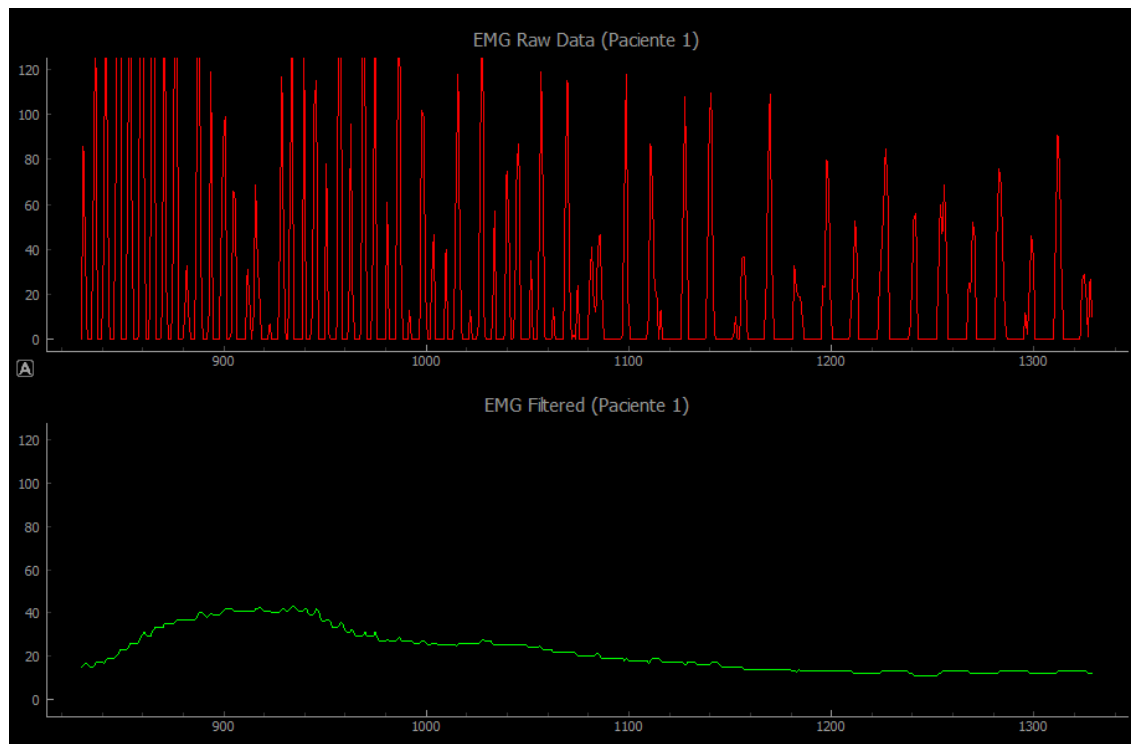
$$Duty = \frac{T_H}{T} * 100 \quad (\text{B.10})$$

Tensão média na carga

$$V_m = Duty * T_{aplicada} \quad (\text{B.11})$$

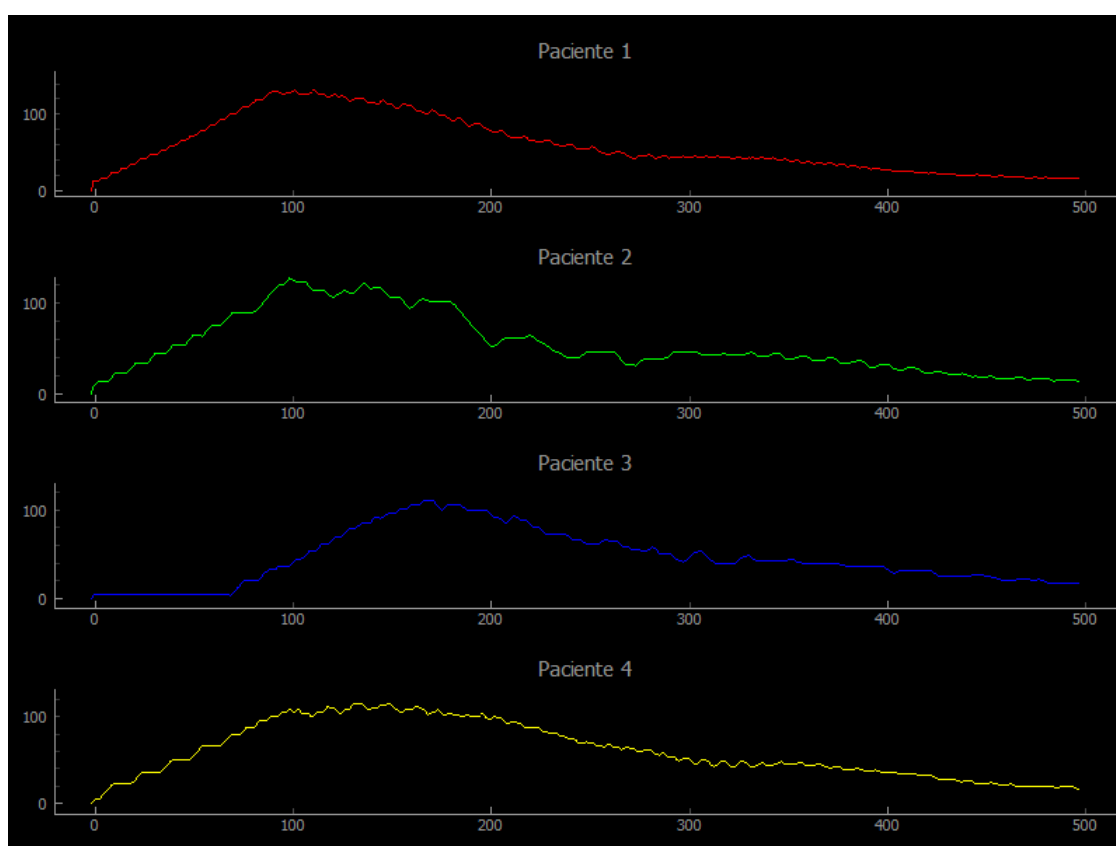
## APÊNDICE C – Simulações do Sistema

Figura 28 – Simulação dos sinais EMG filtrados no microcontrolador



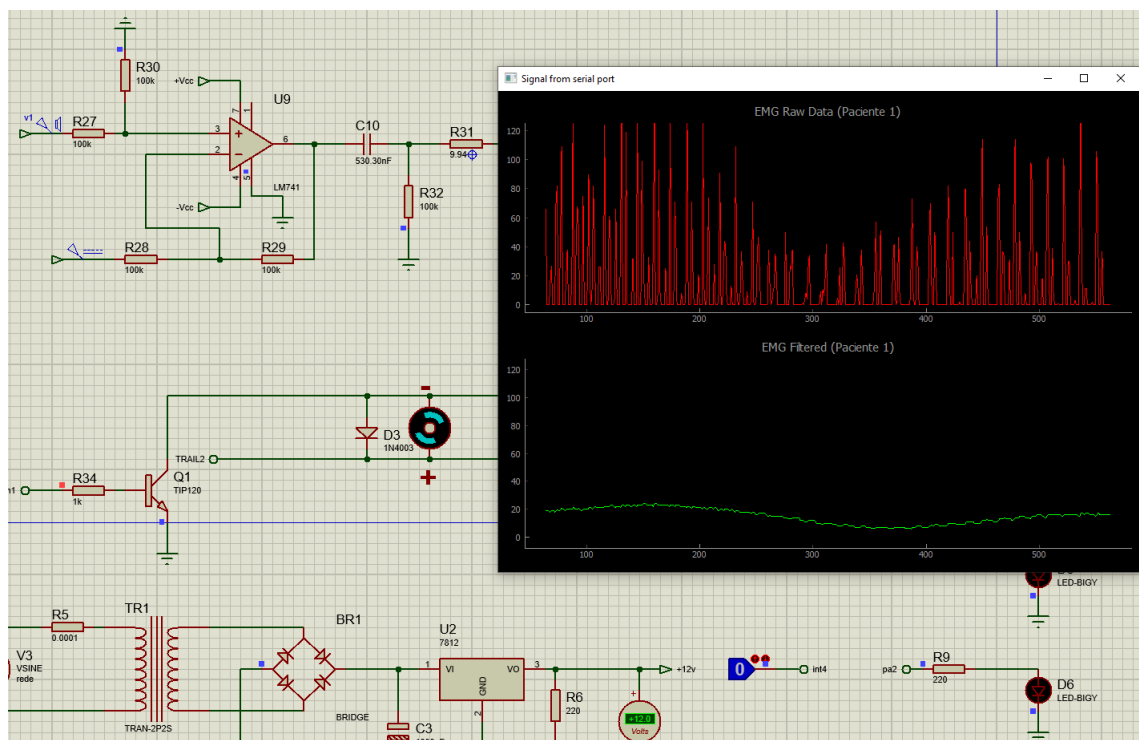
Fonte: Elaborado pelo autor.

Figura 29 – Simulação do Sistema de monitoramento dos 4 pacientes



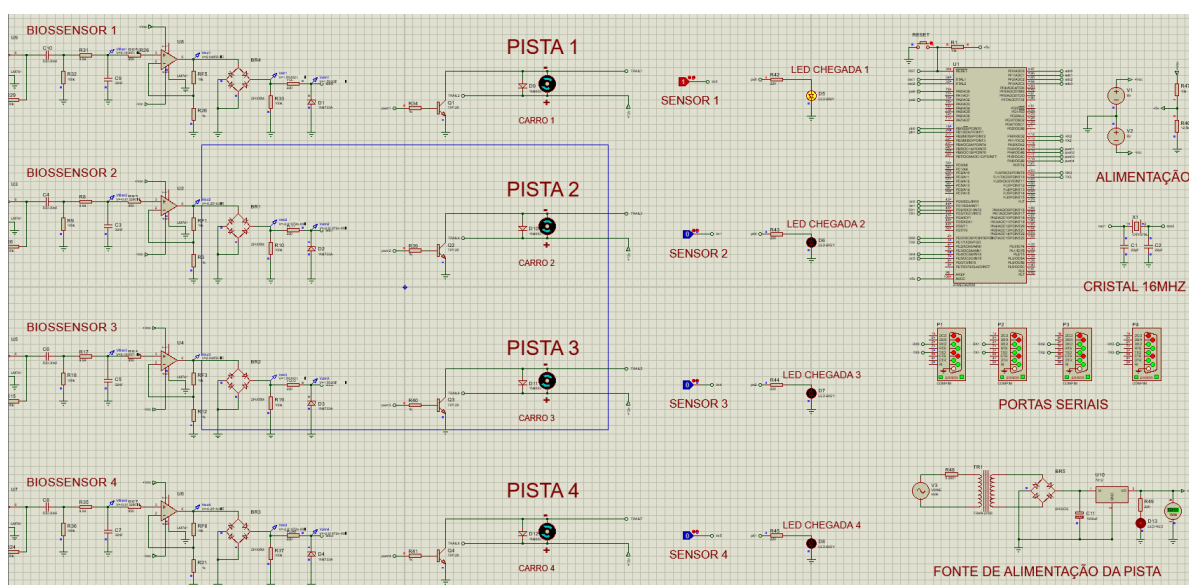
Fonte: Elaborado pelo autor.

Figura 30 – Simulação do Sistema de Controle dos carrinhos no Proteus



Fonte: Elaborado pelo autor.

Figura 31 – Simulação do Sistema completo no Proteus

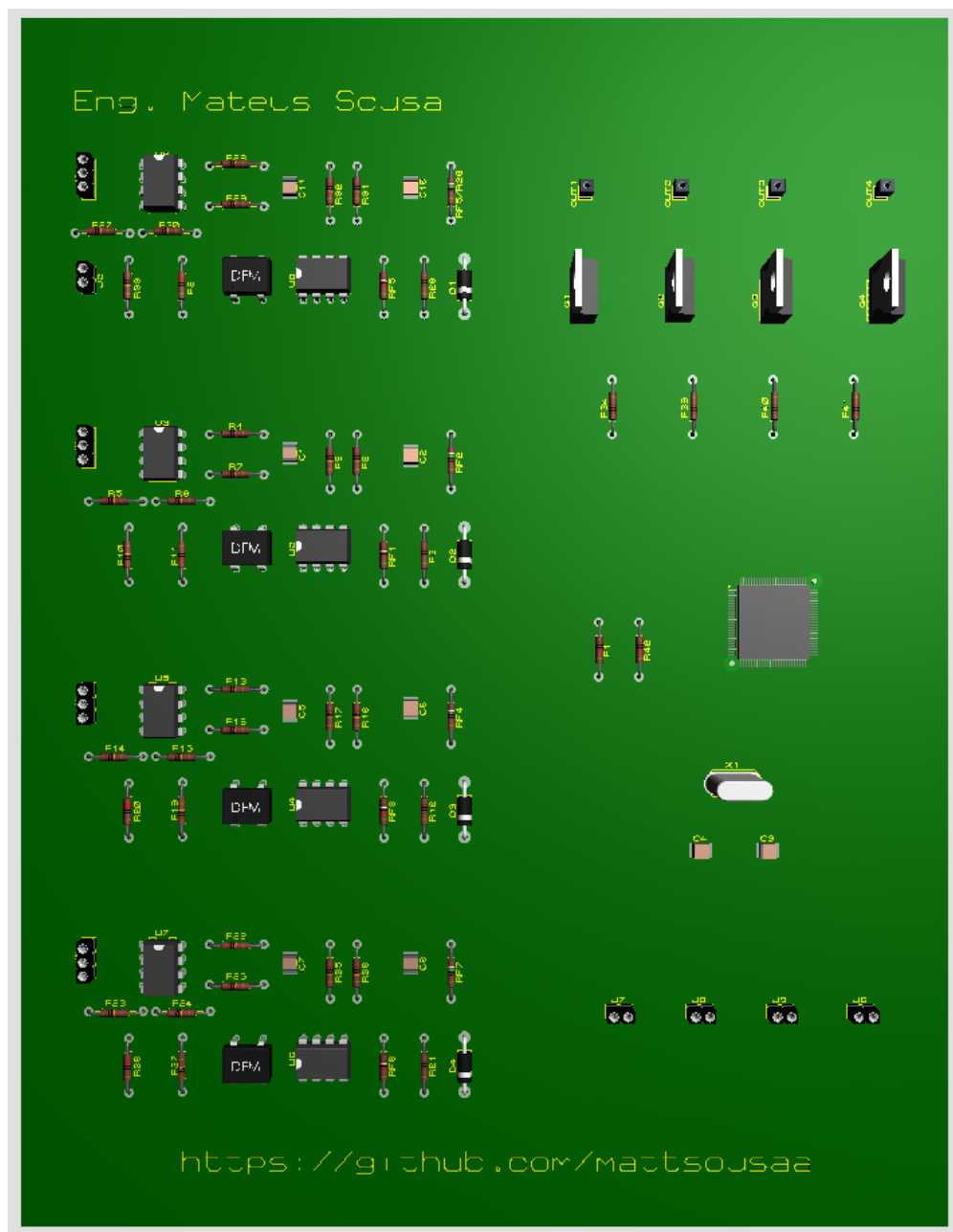


Fonte: Elaborado pelo autor.



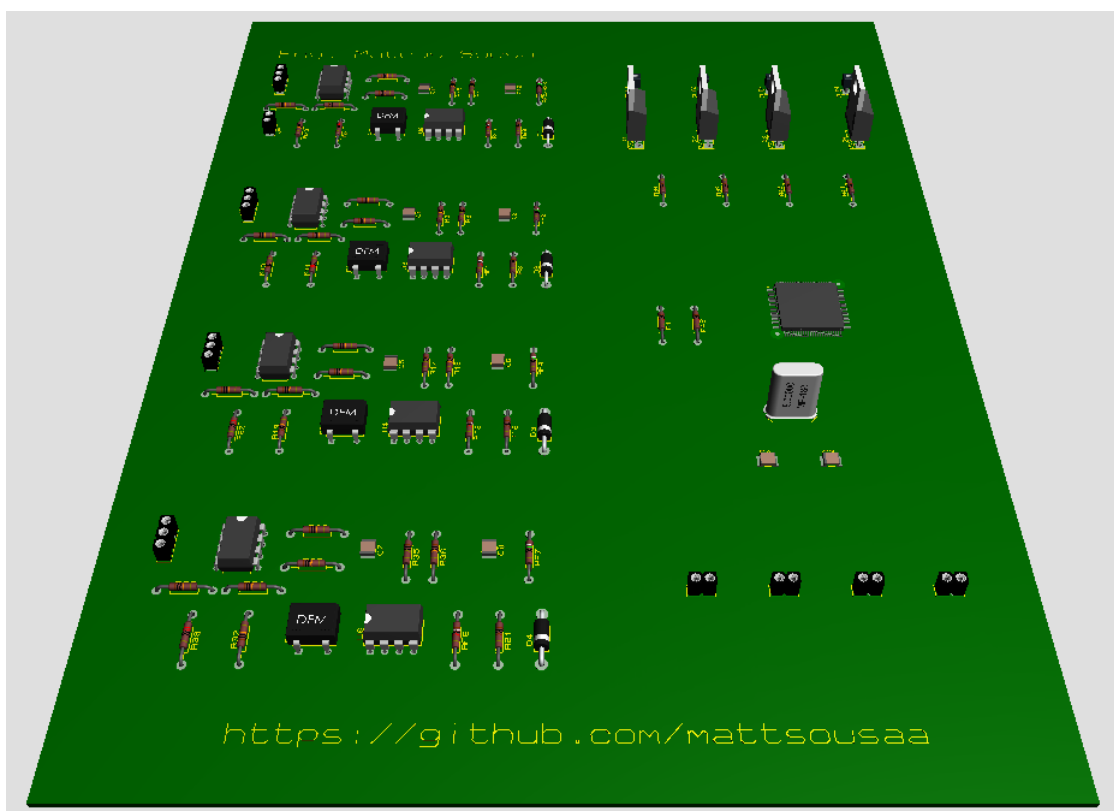
# APÊNDICE D – Modelo de placa PCB Implementada

Figura 32 – Modelo de circuito impresso dos 4 sensores conectados ao ATmega2560



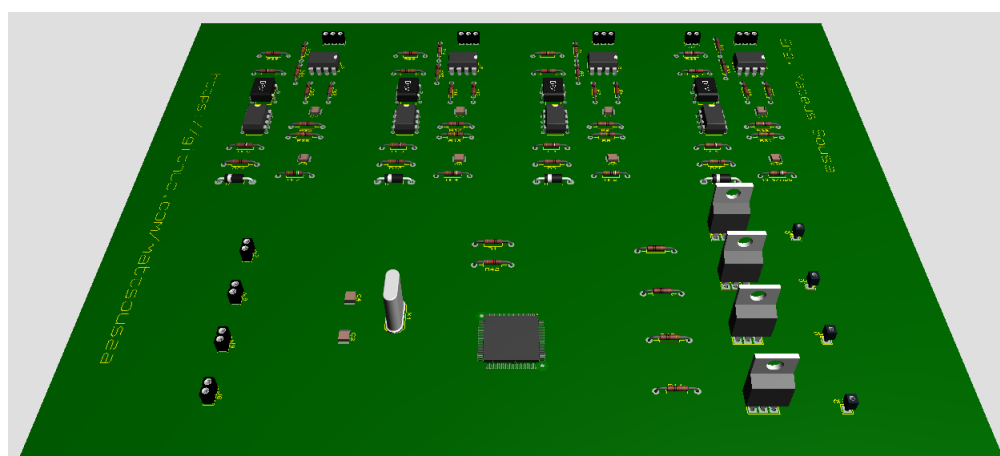
Fonte: Elaborado pelo autor.

Figura 33 – Vista frontal da placa PCB projetada



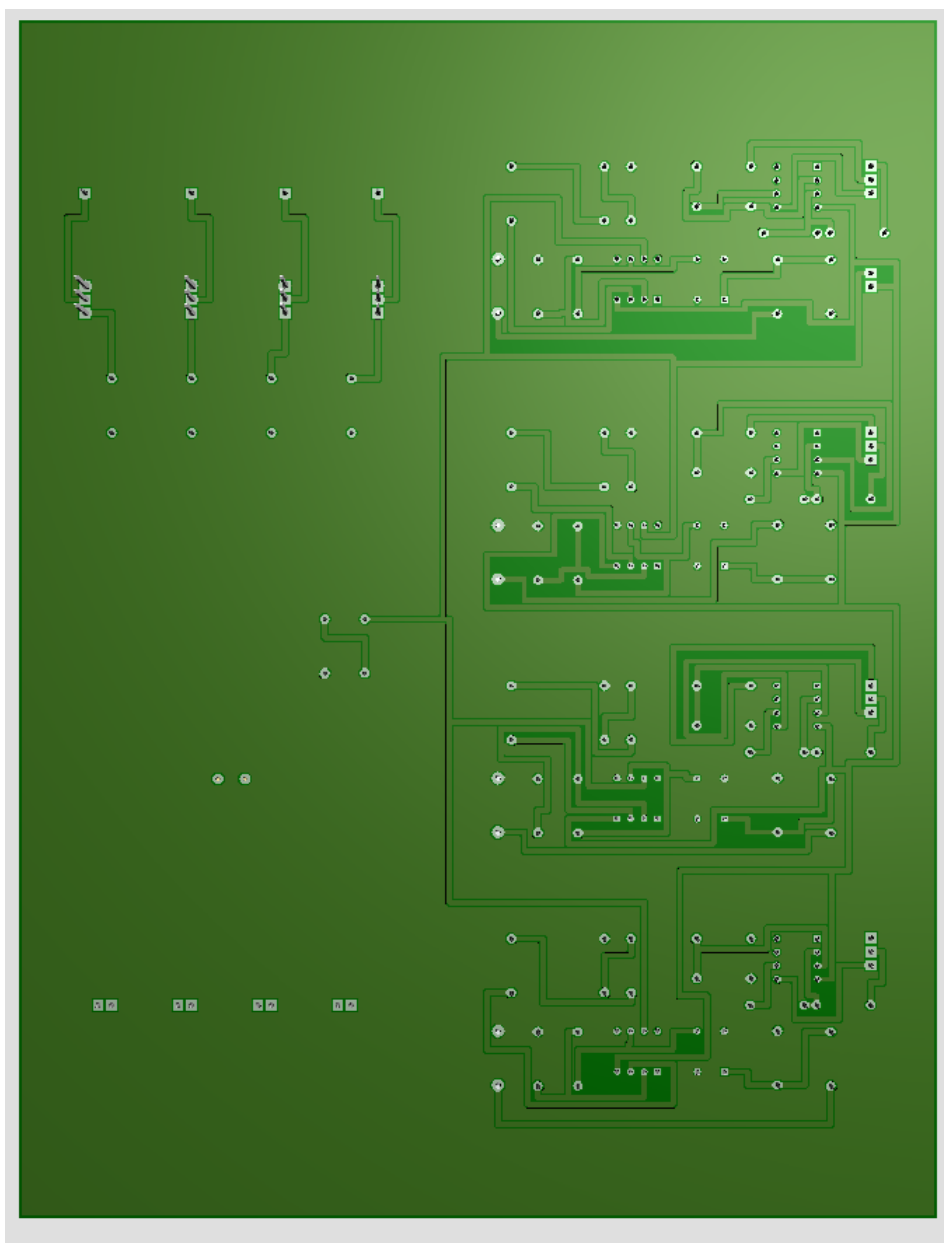
Fonte: Elaborado pelo autor.

Figura 34 – Vista lateral da placa PCB projetada



Fonte: Elaborado pelo autor.

Figura 35 – Vista traseira da placa PCB projetada



Fonte: Elaborado pelo autor.