

**FELIPE BARBOSA DA SILVA**

**Desenvolvimento de eletrônica para controle de motores DC com escova**

**Felipe Barbosa da Silva**

**Desenvolvimento de eletrônica para controle de motores DC com escova**

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador (a): Prof. Dr. Samuel Euzédice de Lucena

Silva, Felipe Barbosa da  
S586d Desenvolvimento de eletrônica para controle de Motores DC com escova. /  
Felipe Barbosa da Silva. – Guaratinguetá, 2018.  
54 f : il.  
Bibliografia: f. 36

Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual  
Paulista, Faculdade de Engenharia de Guaratinguetá, 2018.  
Orientador: Prof. Dr. Samuel Euzedice de Lucena

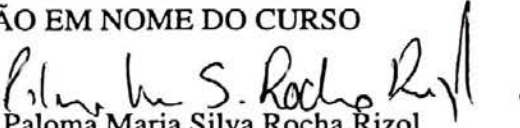
Motores elétricos de corrente contínua. 2. Motores elétricos - Controle  
eletrônico. 3. Sistemas eletrônicos I. Título

CDU 621.313.291


**FELIPE BARBOSA DA SILVA**

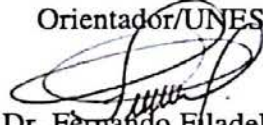
ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO  
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE  
“GRADUADO EM ENGENHARIA ELÉTRICA”


APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE  
GRADUAÇÃO EM NOME DO CURSO

  
Paloma Maria Silva Rocha Rizol  
Coordenadora

**BANCA EXAMINADORA:**

  
Prof. Dr.. Samuel Euzédice de Lucena  
Orientador/UNESP-FEG

  
Prof. Dr. Fernando Filadelfo  
UNESP-FEG

  
Prof. Dr. Daniel Julien Barros da Silva Sampaio  
UNESP-FEG

Dezembro 2018



Dedico este trabalho de modo especial, à minha família, à equipe FEG – Robótica e a todos os amigos que me assistiram nos últimos anos.

## **AGRADECIMENTOS**

Em primeiro lugar agradeço a Deus. Agradeço pela minha vida, minha inteligência, minha família e meus amigos,

ao meu orientador, Prof. Dr. Samuel Euzédice de Lucena e demais professores da instituição que de alguma forma colaboraram com a minha formação, em especial ao Daniel Julien Barros da Silva Sampaio e ao Prof. Dr. Leonardo Mesquita.

às funcionárias e aos funcionários da Faculdade de Engenharia do Campus de Guaratinguetá de maneira geral que sempre foram atenciosos e prestativos.

aos meus pais Gilson Barbosa da Silva e Josimara Aparecida Prezotto da Silva, que apesar das dificuldades enfrentadas, sempre me incentivaram em todas minhas escolhas de vida.

ao grande amigo, e mais que um companheiro de FIFA, que me assistiu em todos os momentos desde que entrei na universidade, Zizinho (Francisco de Assis Vasconcelos Miranda Neto).

aos membros da equipe Feg-Robótica que dividiram comigo a paixão pela robótica, e a sede de conhecimento.

à minha companheira de vida, que me deu foco, força e paciência para que chegasse vitorioso no fim dessa jornada, Paula Góis de Carvalho.

## RESUMO

Este trabalho fundamenta-se no desenvolvimento do hardware completo de uma placa controladora de motor de corrente contínua com escovas, com grande flexibilidade de controle, uma vez que possui entrada para os mais diversos tipos de sensores e comunicações, além de possuir inúmeros métodos de proteção, os quais serão explicitados adiante.

Feito na ideia de um sistema embarcado, tendência mundial, este projeto faz uso de um microcontrolador amplamente difundido, o ATMEGA328, o que facilita a implementação pelos mais diversos tipos de usuários, sejam “hobbistas”, ou profissionais de tecnologia.

A ideia inicial do projeto era fazer um controle de motor de corrente contínua que pudesse controlar a velocidade e sentido de rotação do mesmo, mas ao finalizar a placa e os testes foram observados diversos pontos que poderiam ser incrementados para que se pudesse aumentar a robustez e confiabilidade da placa, sendo assim foi feita uma segunda versão onde mais formas de proteção foram implementadas, e o objetivo foi alcançado com louvor.

A cada passo que a sociedade dá em busca de formas alternativas de energia, o uso de motores de corrente contínua aumenta, sendo assim, é imprescindível que sejam estudadas formas mais eficientes e versáteis de controle dessas máquinas.

**PALAVRAS-CHAVE:** Ponte H. Motor de corrente contínua. Motor escovado. Sistemas embarcados.



## **ABSTRACT**

This work is based on the development of the complete hardware of a motor controller board of direct current with brushes, with great flexibility of control, since it has input for the most diverse types of sensors and communications, besides having numerous methods of protection, which will be explained later.

Made in the idea of an embedded system, worldwide trend, this project makes use of a widely diffused microcontroller, the ATMEGA328, which facilitates the implementation by the most diverse types of users, be they hobbyists or technology professionals.

The initial idea of the project was to make a control of DC motor that could control the speed and direction of rotation of the same, but at the end of the construction and tests, were observed several points that could be improved, so that it could increase the robustness and reliability of the board, so a second version was made where more forms of protection were implemented, and the objective was achieved with praise.

At every step that society looks for alternative forms of energy, the use of DC motors increases, so it is imperative that more efficient and versatile ways of controlling these machines are studied.

**KEYWORDS:** H Bridge. Direct current motor. Brushed motor. Embedded systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Construção de um motor DC com escovas .....	11
Figura 2 – Simplificação de um motor DC .....	12
Figura 3 – Alinhamento das bobinas .....	13
Figura 4 – Circuito equivalente de um motor DC .....	13
Figura 5 – Controle de velocidade com potenciômetro.....	15
Figura 6 – Controle de velocidade com transistor e PWM.....	15
Figura 7 – PWM – Duty Cicle: 99%, 50%, 2% .....	16
Figura 8 – Ponte H com Relês .....	17
Figura 9 – Ponte H com chaves .....	18
Figura 10 – Visão 3D do projeto .....	19
Figura 11 – Visão inferior do projeto .....	20
Figura 12 – TBJ como chave fechada .....	21
Figura 13 – TBJ como chave aberta .....	21
Figura 14 – Mosfet como chave aberta .....	22
Figura 15 – Mosfet como chave fechada.....	23
Figura 16 – Meia Ponte H com driver dedicado.....	24
Figura 17 – Filtro de capacitores .....	25
Figura 18 – Regulador chaveado .....	26
Figura 19 – Diodo TVS .....	26
Figura 20– Regulador chaveado expandido .....	27
Figura 21 – Reguladores de tensão.....	27
Figura 22 – Arduino Nano.....	28
Figura 23 – Módulo Bluetooth HC-05 .....	29
Figura 24 – Trilha rompida.....	31
Figura 25 – BTN8982.....	32
Figura 26 – Visão 3D da segunda versão do projeto.....	33
Figura 27 – Visão inferior da segunda versão do projeto.....	33
Figura 28 – Visão 3D dos BTNs e dos conectores dos motores .....	34
Figura 29 – Placa confeccionada industrialmente .....	34

## LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico Digital
AC	Alternate Current
CI	Circuito Integrado
FEG	Faculdade de Engenharia de Guaratinguetá
LED	Light Emitting Diode
LiPo	Lithium Polymer
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PWM	Pulse Width Modulation
TVS	Transient Voltage Suppressor
USB	Universal Serial Bus

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>10</b>
<b>2</b>	<b>DESENVOLVIMENTO (REVISÃO TEÓRICA)</b> .....	<b>11</b>
2.1	MOTORES DE CORRENTE CONTÍNUA COM ESCOVA.....	11
2.2	FORMAS DE ACIONAR UM MOTOR DC.....	14
<b>2.2.1</b>	<b>Controle de velocidade</b> .....	<b>14</b>
<b>2.2.2</b>	<b>PWM</b> .....	<b>15</b>
<b>2.2.3</b>	<b>Controle de sentido de rotação</b> .....	<b>16</b>
<b>2.2.4</b>	<b>Ponte H</b> .....	<b>17</b>
<b>3</b>	<b>DESENVOLVIMENTO</b> .....	<b>19</b>
3.1	PRIMEIRA VERSÃO UTILIZANDO COMPONENTES DICRETOS .....	19
<b>3.1.1</b>	<b>Dispositivos seccionadores</b> .....	<b>20</b>
<b>3.1.2</b>	<b>Transistor TBJ</b> .....	<b>20</b>
<b>3.1.3</b>	<b>Transistor MOSFET</b> .....	<b>22</b>
<b>3.1.4</b>	<b>Driver IR2104 para MOSFET</b> .....	<b>23</b>
<b>3.1.5</b>	<b>Dispositivos de proteção</b> .....	<b>25</b>
<b>3.1.6</b>	<b>Alimentação</b> .....	<b>26</b>
<b>3.1.7</b>	<b>Circuito Lógico</b> .....	<b>28</b>
<b>3.1.8</b>	<b>Comunicação Bluetooth</b> .....	<b>29</b>
<b>4</b>	<b>APRIMORAMENTO – SEGUNDA VERSÃO</b> .....	<b>30</b>
4.1	APRIMORAMENTOS.....	31
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>35</b>
	<b>REFERÊNCIAS</b> .....	<b>36</b>
	<b>BIBLIOGRAFIA CONSULTADA</b> .....	<b>37</b>
	<b>ANEXO A - CÓDIGO</b> .....	<b>38</b>
	<b>ANEXO B - ESQUEMA ELÉTRICO DO PRIMEIRO PROJETO</b> .....	<b>53</b>
	<b>ANEXO C - ESQUEMA ELÉTRICO DO SEGUNDO PROJETO</b> .....	<b>54</b>

## 1 INTRODUÇÃO

Apesar de combustíveis fósseis ainda serem predominantes como forma de energia, novas formas limpas de energia estão ganhando cada vez mais espaço, devido aos problemas ambientais que estão cada vez mais severos, e por esses motivos equipamentos que fazem uso dessas novas formas de energia estão ganhando cada vez mais espaço no mercado, como exemplo pode-se citar os carros elétricos, que em diversos países desenvolvidos já estão sendo preteridos.

O principal dispositivo que faz uso dessas novas formas de energia são os motores elétricos de corrente contínua, o qual possui grande versatilidade no controle de velocidade e sentido de rotação, e sua aplicação se estende desde pequenos brinquedos, até motores de carros elétricos, por isso o estudo de novas formas de acionamento do mesmo é imprescindível nesse novo cenário global.

A topologia de ponte H trata-se de um circuito de potência capaz de controlar de forma eficaz o sentido de rotação desses motores e as suas velocidades, o presente trabalho faz uso dessa topologia de duas formas diferentes, na primeira com componentes discretos e na segunda com circuitos integrados dedicados para esse tipo de tarefa.

A primeira parte deste trabalho foi fundamental para o aprendizado, pois todos os conceitos básicos puderam ser analisados, assim como os detalhes que compõe o desenvolvimento de um novo hardware, por isso a escolha de componentes discretos como transistor e “drivers” de MOSFETS.

Ao passo que não existe uma receita para se desenvolver tamanho circuito eletrônico, é preciso ir a busca de diversas soluções e informações para os problemas que aparecem no caminho, pode se citar alguns problemas na parte de proteção da placa, os quais muitas vezes não são previstos nos livros, e cabe ao desenvolvedor pensar em soluções e formas de se aumentar a robustez de seu projeto.

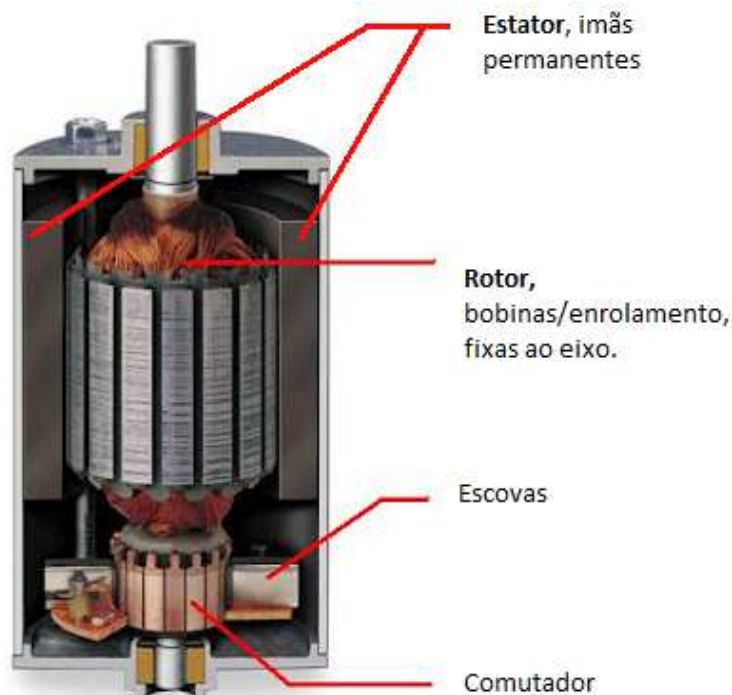
Já na segunda parte do presente trabalho a intenção foi de usar tudo o que já havia sido adquirido de conhecimento com a primeira versão e substituir as áreas mais susceptíveis a falhas por componentes mais modernos e seguros, por isso a escolha de componentes dedicados.

## 2 DESENVOLVIMENTO (REVISÃO TEÓRICA)

### 2.1 MOTORES DE CORRENTE CONTÍNUA COM ESCOVA

Para se desenvolver um controlador de motor de corrente contínua escovado é primeiro necessário entender a construção, Figura 1, e funcionamento, Figura 2, Figura 3, e Figura 4, do mesmo.

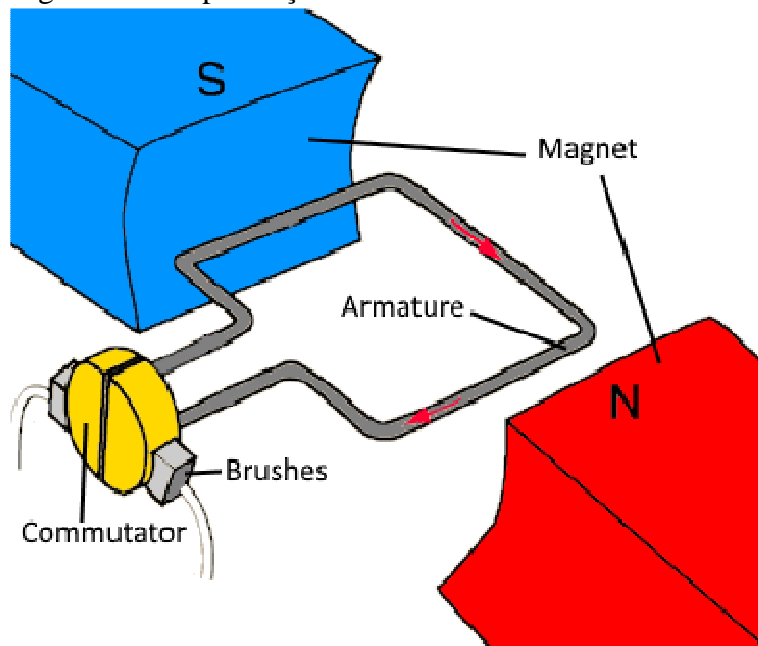
Figura 1 – Construção de um motor DC com escovas



Fonte: Eporsche website (2012).

Esse tipo de motor é um dispositivo eletromecânico que é constituído por três elementos principais, estator, rotor e comutador. E para entender cada um desses componentes é necessário analisar seu funcionamento por partes, em primeiro momento com o motor desligado, sem nenhuma fonte de energia ligada em seus terminais (Escovas), a única força resultante dentro do motor é oriunda dos ímãs permanentes que forma o estator, com um pólo Norte e um pólo Sul, como as bobinas não estão energizadas o rotor (eixo) pode girar livremente sobre seus rolamentos.

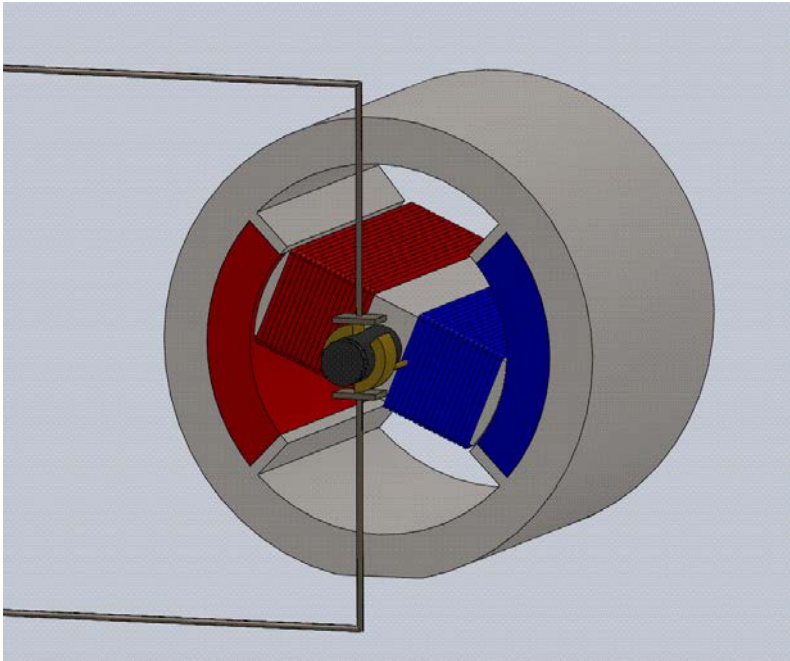
Figura 2 – Simplificação de um motor DC



Fonte: Progressive Automations website (2017).

Agora para entender a utilidade do comutador é necessário imaginar como seria o funcionamento do motor DC caso esse componente não existisse. As bobinas seriam alimentadas diretamente, transformando um lado dos enrolamentos em pólo norte e outro em pólo sul, e então os ímãs do estator atrairiam seus opostos, e o rotor iria parar em uma posição de 180°, e não iria girar mais, por isso, para que o motor continue a girar é necessária uma forma de que a corrente nas bobinas inverta seu sentido a cada ciclo em que o rotor alinhe sua polaridade com a polaridade dos ímãs fixos no estator, e então gere torque no motor. Nesse aspecto é introduzido o comutador, que faz a inversão do sentido da corrente de forma mecânica, sem que seja necessário o uso de alguma forma de controle eletrônico para manter o motor em contínuo movimento angular.

Figura 3 – Alinhamento das bobinas

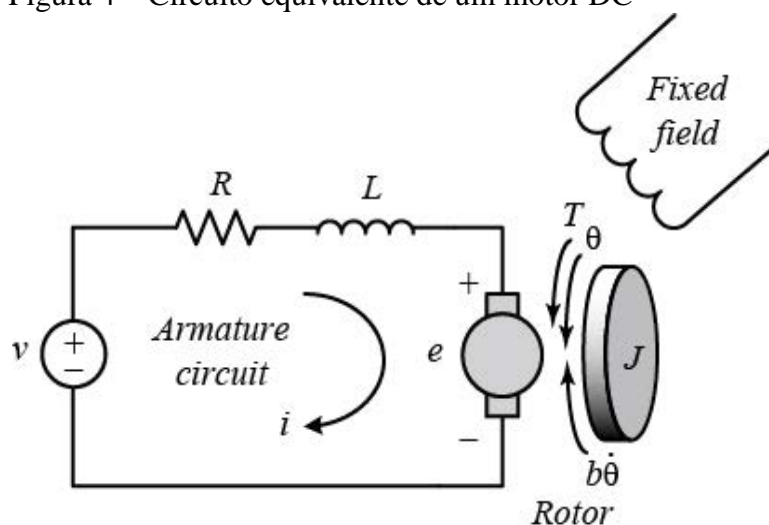


Fonte: Giphy website (2018).

O comutador possui chapas metálicas ligadas às bobinas do rotor, e essas chapas entram em contato com as escovas por atrito, e essas escovas normalmente são feitas de algum composto de carbono, normalmente carvão. Sendo esse atrito um dos fatores que diminuem a eficiência desse tipo de motor.

É possível modelar um motor DC, e representá-lo através de um circuito RL em série com uma fonte de tensão, e esses valores são diretamente proporcionais a velocidade.

Figura 4 – Circuito equivalente de um motor DC



Fonte: Ctms website (2018).



Pela figura pode se extrair a seguinte equação:

$$v = L \cdot \frac{di_a}{dt} + R \cdot i_a + K \cdot \omega$$

Onde:

- $v$  é a tensão aplicada nos terminais do motor;
- $i_a$  é a corrente fluindo através do motor;
- $K$  é a constante que dita a tensão gerada nos terminais do motor quando é multiplicada pela rotação do eixo  $\omega$  (velocidade angular do eixo);

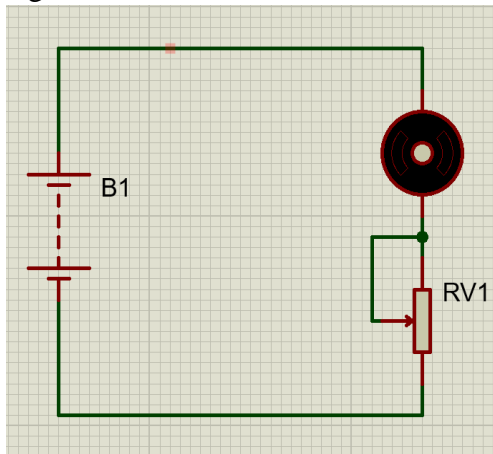
## 2.2 FORMAS DE ACIONAR UM MOTOR DC

Uma vez entendido o funcionamento e construção do motor DC, fica fácil concluir que para se fazer um motor DC girar basta se aplicar uma tensão em seus terminais, e a velocidade do motor é diretamente proporcional à essa tensão, caso queira inverter a rotação desse motor é necessário gerar uma tensão negativa, que pode ser feito invertendo a polaridade da fonte de tensão.

### 2.2.1 Controle de Velocidade

Uma abordagem simples para se fazer o controle de velocidade, seria utilizar um potenciômetro em serie com o motor, Figura 5, e ao variar a resistência iria variar a tensão em cima do mesmo, sendo assim iria mudar sua velocidade também. O problema dessa abordagem é a baixa eficiência, pois a energia que não é utilizada pelo motor é dissipada em forma de calor pelo potenciômetro. Por exemplo, se utilizar um motor com uma resistência interna de  $0,5\Omega$ , e uma fonte de 24V, para deixar a velocidade do motor em 50% o potenciômetro deve atingir  $0,5\Omega$  para que a tensão seja dividida de forma igual, sendo assim a energia dissipada em cima do potenciômetro seria de 12W, o que é um grande desperdício de energia se o calor dissipado não for utilizado para nenhuma outra finalidade.

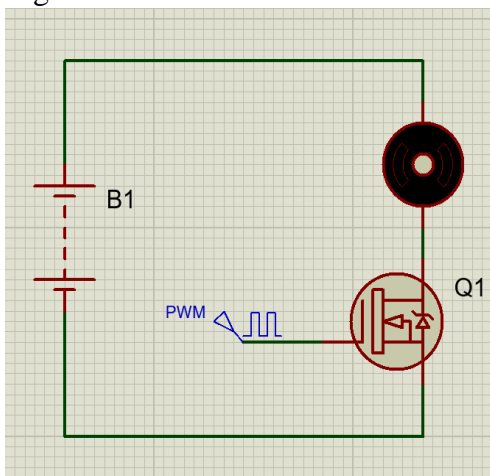
Figura 5 – Controle de velocidade com potenciômetro



Fonte: Produção do próprio autor

Para se aumentar a eficiência pode se utilizar no lugar do potenciômetro uma chave eletrônica, como um transistor TBJ, ou um MOSFET, Figura 6, e utilizar a técnica de PWM, para controlar a velocidade.

Figura 6 – Controle de velocidade com transistor e PWM



Fonte: Produção do próprio autor

### 2.2.2 PWM

Para controlar a velocidade de giro de um motor DC, usa-se um conceito conhecido como PWM (Pulse Width Modulation, Modulação por Largura de Pulso). Esse método consiste em ligar e desligar o motor, numa frequência fixa, através de um dispositivo seccionador, normalmente algum tipo de transistor (bipolar ou MOSFET), fazendo com que o motor gire numa velocidade proporcional à relação entre o tempo ligado ( $T_{on}$ ) e período da

forma de onda (T). Essa relação é chamada de Duty Cycle (D), Figura 7, e, se multiplicada pela tensão de alimentação do motor, temos uma tensão média que equivale à tensão DC que deveria ser aplicada para fazer o motor girar à mesma velocidade.

Figura 7 – PWM – Duty Cycle: 99%, 50 % e 2%



Fonte: Produção do próprio autor

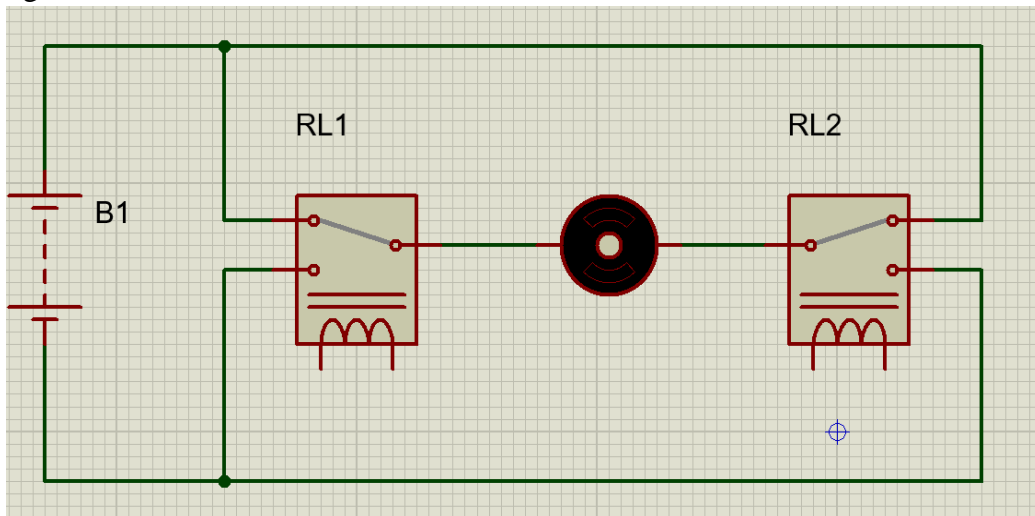
Naturalmente, se a tensão média entre os terminais do motor é reduzida de seu valor nominal de acordo com o Duty Cycle, a corrente média que circula por ele também é reduzida conforme o mesmo parâmetro. Logo, se o motor nunca operar com um Duty Cycle maior que certa porcentagem, podem-se dimensionar os demais componentes e condutores do circuito para atenderem este valor de corrente ao invés dos valores nominais de corrente do motor, o que pode aumentar o leque de dispositivos que possam ser usados ou mesmo aumentar a margem de segurança considerada.

### 2.2.3 Controle de sentido de rotação

Uma vez escolhida a forma de controlar a velocidade der rotação do motor, pode se pensar na forma de controlar o sentido de rotação do mesmo, como dito anteriormente, para inverter o sentido de rotação deve inverter o sentido da corrente através do motor.

Uma alternativa simples e barata seria fazer uso de dois relês, Figura 8, onde o acionamento independente de cada um deles faria o motor girar em um sentido, e no caso dos dois estarem ligados ou desligados ao mesmo o motor seria freiado, pois a energia seria dissipada em cima do mesmo, isso é conhecido como freio eletrônico.

Figura 8 – Ponte H com relés



Fonte: Produção do próprio autor

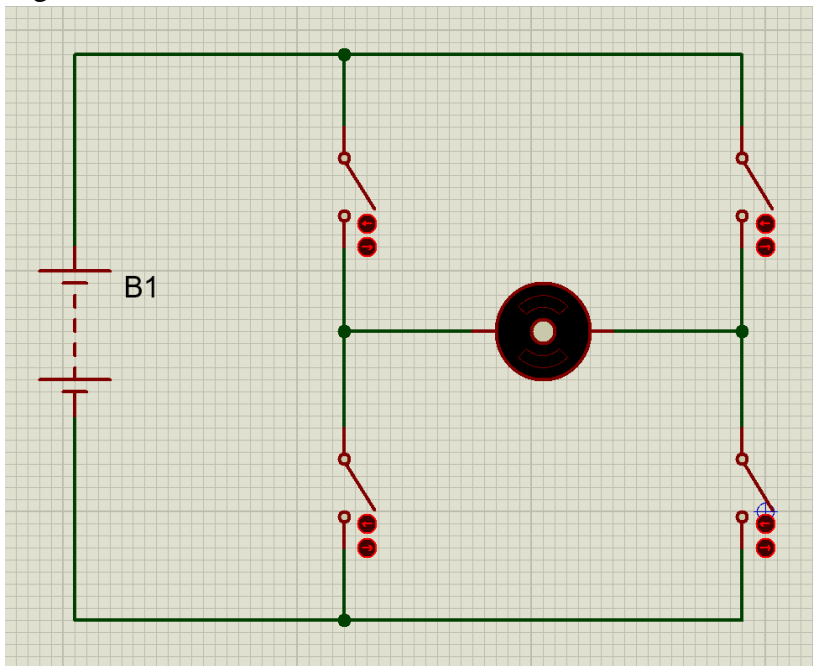
Existem alguns problemas com essa técnica, primeiro é que a reversão instantânea do sentido de rotação de um motor DC gera altos picos de corrente e tensão que diminuem a vida útil dos relês, o segundo problema é que para controlar motores que demandam altas correntes são necessários relês que possuem grandes dimensões físicas que inviabilizam um projeto de PCB. Outro problema a se citar é a lentidão de resposta desses dispositivos, uma vez que são eletromecânicos.

Uma alternativa para uso de relês é utilizar uma topologia de ponte H, com transistores, que com os avanços da microeletrônica tem possibilitado o controle de grandes correntes em pequenos encapsulamentos.

#### 2.2.4 Ponte H

A ponte H, Figura 9, é usada para controlar o sentido de rotação de um motor DC. Uma ponte H básica é composta por quatro chaves mecânicas ou eletrônicas posicionadas formando a letra “H”, sendo que cada uma localiza-se em um extremo e o motor é posicionado no meio.

Figura 9 – Ponte H com chaves



Fonte: Produção do próprio autor

Para que o motor funcione, basta acionar um par de chaves diagonalmente opostas, o que faz com que a corrente flua do pólo positivo para o negativo, atravessando o motor e fazendo-o girar em um determinado sentido. Para inverter a rotação, desligamos essas chaves e acionamos o outro par de chaves, o que faz com que a corrente siga na direção oposta e, conseqüentemente, invertendo o sentido da rotação do motor.

Caso se deseje parar o motor por completo, pode-se:

- 1- Fechar ambas as chaves superiores da ponte H
- 2- Fechar ambas as chaves inferiores da ponte H
- 3- Abrir todas as chaves da ponte H

Os métodos 1 e 2 efetivamente põem os terminais do motor em curto-circuito (em um mesmo potencial), fazendo com que a corrente gerada pelo campo magnético resultante do motor circule por ele mesmo de modo a tender o motor girar no sentido oposto ao sentido de quando estava ligado, fazendo-o parar rapidamente. Este modo de parada é chamado de freio dinâmico.

O método 3 fará com que toda corrente circulante pelo motor anteriormente pare instantaneamente, porém o motor reduzirá sua velocidade até zero de forma lenta, pois contará apenas com a força de atrito do rotor ao invés de uma corrente reversa de descarga como no freio dinâmico. Tal modo de parada é chamado de coasting.

Deve-se tomar muito cuidado para não acionar as chaves de um mesmo lado da ponte “H” simultaneamente. Isso faz com que o fluxo de corrente vá direto do pólo positivo para o negativo, causando um curto-circuito na fonte de alimentação, gerando correntes muito altas e que podem danificar componentes do circuito, tal efeito é conhecido como “Shoot-through”.

### 3 DESENVOLVIMENTO

#### 3.1 PRIMEIRA VERSÃO UTILIZANDO COMPONENTES DISCRETOS

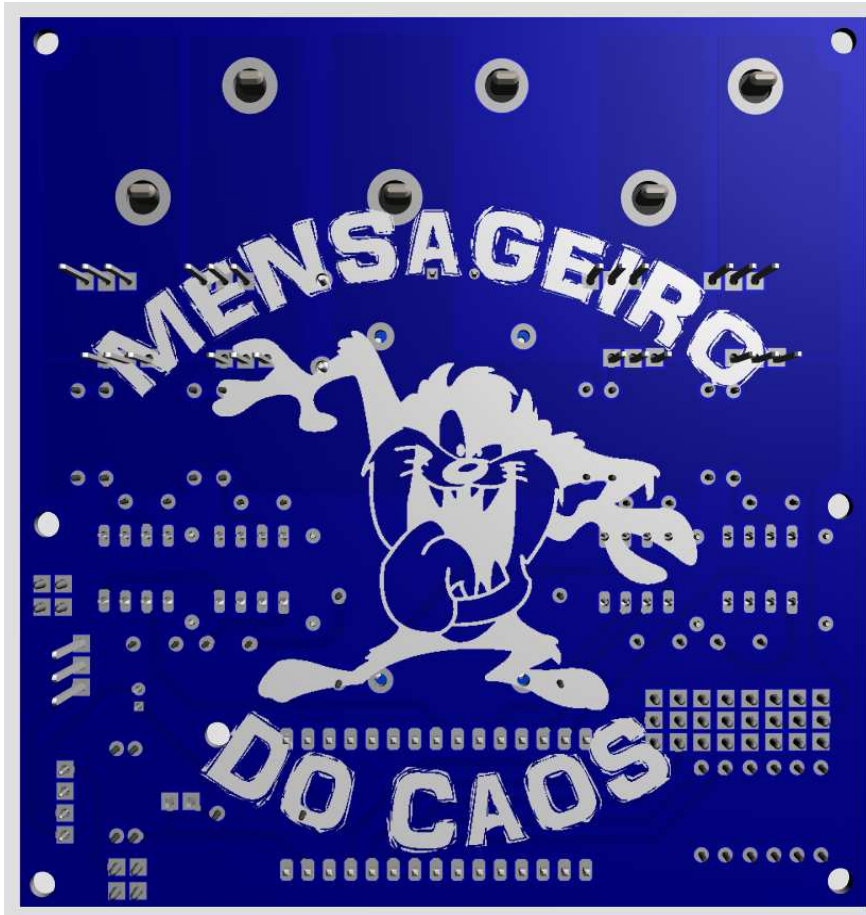
A primeira abordagem tomada foi de desenvolver uma eletrônica utilizando apenas componentes discretos, Figura 10, e Figura 11 como MOSFET, diodos, capacitores, resistores e etc., reduzindo o uso de circuitos integrados, uma vez que isso iria agregar mais conhecimento à formação acadêmica. Ao passo que a primeira versão foi concluída com sucesso, observou-se durante os testes uma oportunidade de melhoria em termos de dispositivos de proteção e em confiabilidade do sistema, o que resultou no desenvolvimento de uma segunda versão do projeto que será explicada mais adiante. Por hora segue a descrição de todo aprendizado adquirido com esse primeiro projeto.

Figura 10 – Visão 3D do projeto



Fonte: Produção do próprio autor

Figura 11 – Visão inferior do projeto



Fonte: Produção do próprio autor

### 3.1.1 Dispositivos seccionadores

Para o desenvolvimento de uma ponte H, o primeiro passo é saber as características do motor que pretende se controlar, para o devido projeto o motor que foi usado de base para os cálculos foi um Maxon RE35 de 24V, que atinge uma corrente máxima de 41.1A com eixo travado, esse valor de corrente e tensão serão utilizados como base para a escolha dos dispositivos seccionadores e dispositivos de proteção.

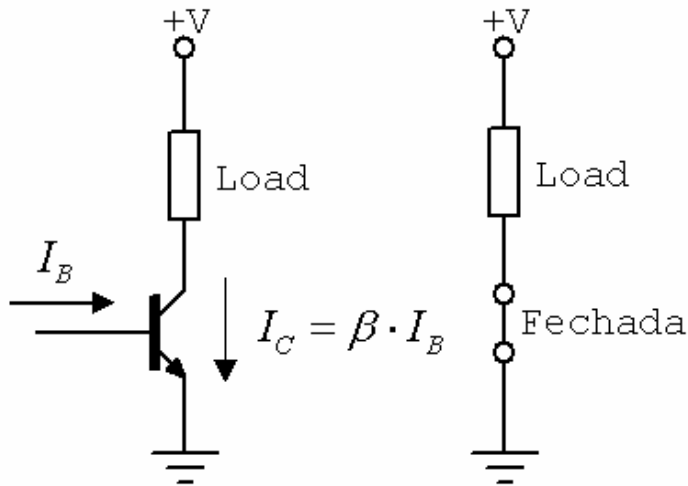
### 3.1.2 Transistor TBJ

A primeira opção a se analisar é o transistor bipolar de junção, por ter sido o primeiro transistor a ter sido criado é o de construção mais simples e barato. Seu funcionamento é bem simples, a corrente que flui entre seus terminais, coletor e emissor é diretamente proporcional a corrente de base, sendo assim um amplificador de corrente, com um ganho dado pela constante  $\beta$ . Um uso muito comum para esse dispositivo é o de chave eletrônica, onde ele



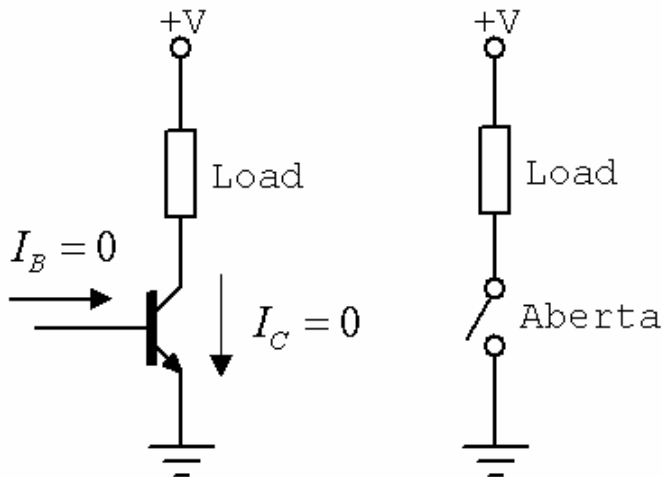
trabalha somente nas regiões de corte e saturação, funcionando como uma chave aberta, Figura 12, quando a corrente em sua base é nula, e como chave fechada, Figura 13 quando a corrente de base é suficientemente grande para acionar a carga.

Figura 12 – TBJ como chave fechada



Fonte: PUC Rio website (2004).

Figura 13 – TBJ como chave aberta



Fonte: PUC Rio website (2004).

Para analisarmos a viabilidade do uso de um TBJ para este tipo de aplicação temos que observar a perda de potência que este dispositivo oferece, uma vez que a corrente necessária pra carga já está determinada, que é a corrente máxima que o motor pode atingir, os 41,1A com eixo travado, e a queda de tensão entre o coletor e emissor fica entre 0,5V e 1V, tem-se

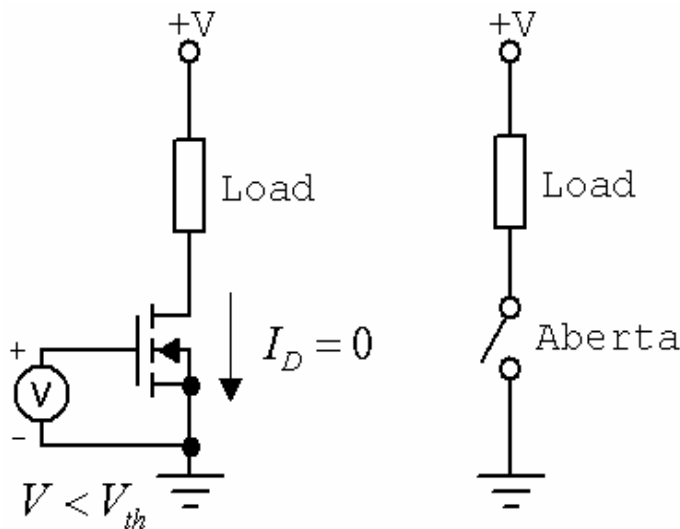
uma perda máxima de  $P(W) = 1V * 41,1A = 41,1W$  , o que iria demandar dissipadores de calor relativamente grande e pesados, além do fato que seria um desperdício de energia.

Outro fator importante a se salientar é que para se atingir esse alto valor de corrente no coletor, seria necessário uma corrente de base muito grande, pois o ganho desse tipo de transistor de potência fica por volta de  $\beta = 50$  , e seria necessário mais um estágio de amplificação de corrente, pois a parte lógica do projeto seria feita por um microcontrolador onde a corrente de suas portas de entrada e saída podem atingir no máximo 50mA sem danificar o componente.

### 3.1.3 Transistor MOSFET

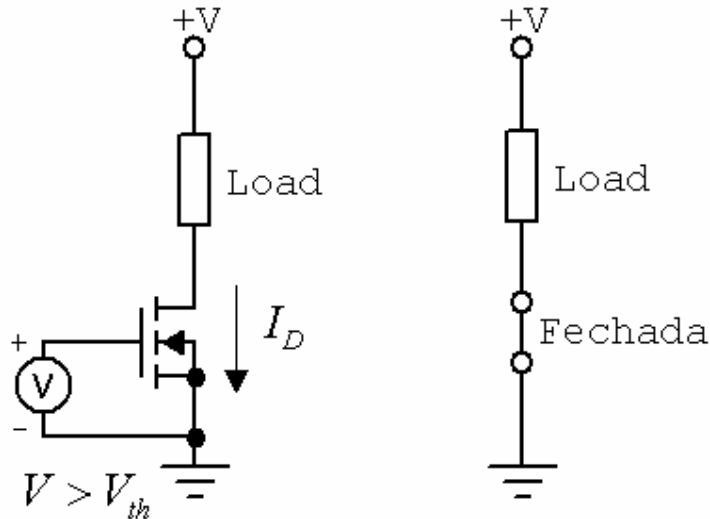
O próximo dispositivo a se analisar é o MOSFET, diferente do TBJ seu acionamento é feito por tensão e não por corrente, o que torna seu acionamento muito mais simples. Para se utilizar um MOSFET como chave fechada, Figura 15, basta que a tensão no terminal de gate atinja o valor de tensão limiar ( $V_{th}$ ), e para usar como chave aberta, Figura 14, a tensão no gate tem que ser menor que essa tensão limiar.

Figura 14 – Mosfet como chave aberta



Fonte: PUC Rio website (2004).

Figura 15 – Mosfet como chave fechada



Fonte: PUC Rio website (2004).

Assim como foi feito para o TBJ, para analisar a viabilidade do MOSFET para este tipo de aplicação é necessário ver principalmente a perda de potência por condução. Quando este dispositivo está conduzindo ele se comporta basicamente como um resistor em série com a carga, resistência essa conhecida como  $R_{ds(on)}$  (Resistência Dreno-Source no estado ligado). Na primeira versão deste projeto foi utilizado o MOSFET IRF1405, onde o valor dessa resistência era igual a  $5,3\text{m}\Omega$ , com isso a perda por condução para o caso do motor escolhido era de  $P(W) = 41,1^2 \cdot 5,3 \cdot 10^{-3} = 8,95\text{W}$ , praticamente 4 vezes menor do que a perda em um TBJ.

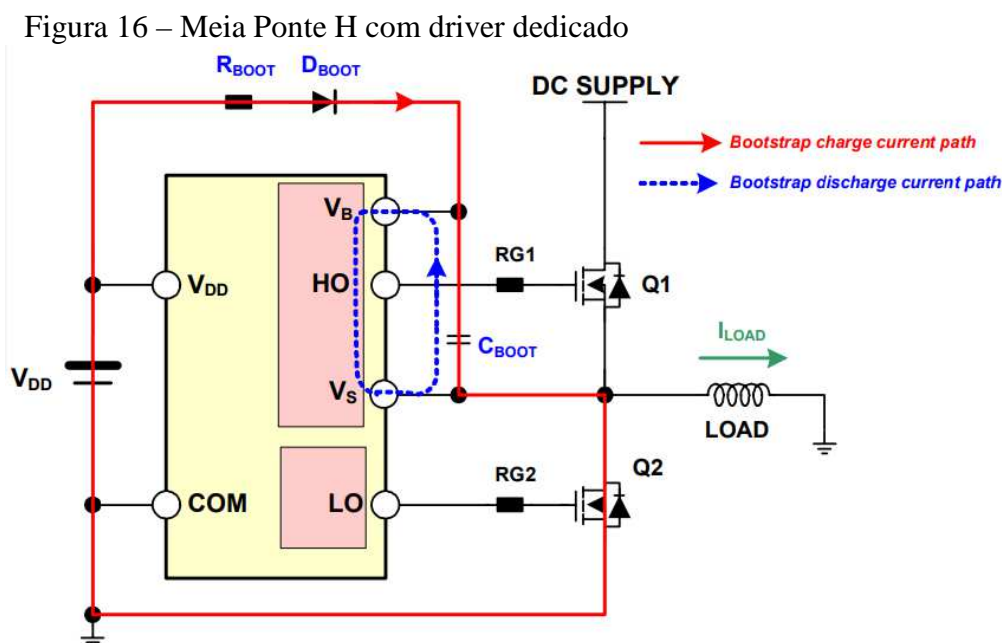
Para a primeira versão do projeto onde foram utilizados componentes discretos, foi necessário o uso de drivers de MOSFETs para fazer a interface com o microcontrolador, pois o mesmo não conseguiria atingir os valores de tensão limiar necessário para ligar um MOSFET.

### 3.1.4 Driver IR2104 para MOSFET

Como já foi dito para fazer a interface entre a ponte H de MOSFET e o microcontrolador foi utilizado um circuito integrado dedicado a fazer o controle de meia ponte H, sendo assim foram utilizados dois IR2104, Figura16, para cada ponte H.

Um detalhe importante a se notar quando se projeta uma ponte H com MOSFET é o fato de que os transistores superiores não possuem uma tensão no terminal “source” fixa, uma vez que este está ligado ao motor, e então fica complicado ter o controle da tensão  $V_{GS}$  para controlar o estado de ligado e desligado do componente, e por esse motivo esses drivers necessitam de um circuito complementar, conhecido como “bootstrap”, composto de um diodo, um capacitor, e em alguns casos de um resistor, que é usado para garantir o valor de tensão limiar para ligar o MOSFET superior.

O funcionamento do circuito de “bootstrap” é simples, mas se seu projeto não for bem dimensionado pode causar mal funcionamento, ou até mesmo o colapso do projeto. Ele funciona da seguinte forma, no primeiro momento o MOSFET inferior quando acionado carrega o capacitor, e quando ocorre a inversão do ciclo, o capacitor já carregado se descarrega no MOSFET superior, fazendo dessa forma com que ele seja ligado, o diodo serve para que o capacitor não se descarregue na fonte de energia.



Fonte: Fairchild website (2014).

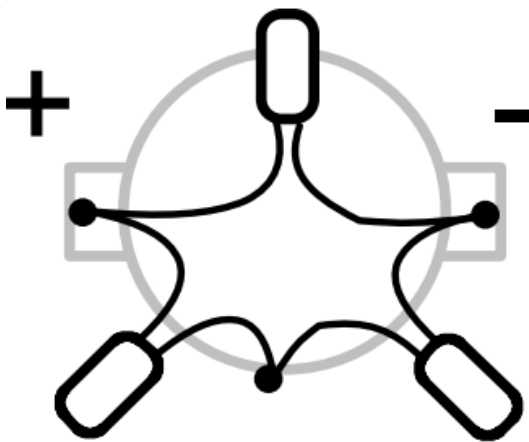
Para o cálculo do capacitor foi utilizado o “Application Note – AN-6076” da “Fairchild Semiconductor” como referência, e o valor escolhido foi de 1nF.

### 3.1.5 Dispositivos de proteção

Quando se faz acionamento de motores DC existem algumas precauções que devem ser tomadas para proteger o circuito eletrônico dos picos de tensão e corrente oriundos das escovas do motor, sendo os momentos mais críticos quando se faz a inversão de rotação e a partida.

A primeira medida a se tomar é soldar capacitores em uma configuração delta entre os terminais e a carcaça do motor, Figura 17, esses capacitores irão filtrar parte do ruído gerado pelas escovas.

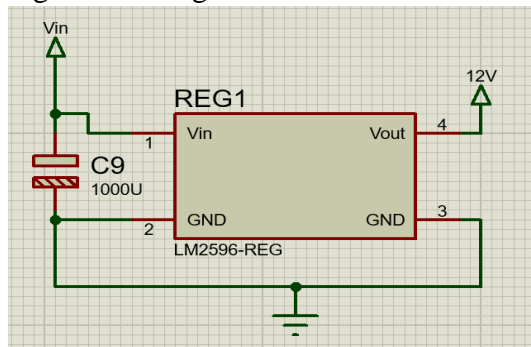
Figura 17 - Filtro de capacitores



Fonte: Produção do próprio autor

Um outro problema que se encontra quando está acionando motores DC de alta corrente, é no momento que ocorre variações bruscas de velocidade e torque, a demanda de corrente do motor aumenta, causando quedas de tensão na fonte de energia, o que pode ocasionar um “reset” da parte lógica do circuito, para evitar que isso ocorra, é colocado em paralelo com a alimentação, bem próximo da entrada do circuito, um capacitor de valor alto, Figura 18, no presente projeto foi utilizado um capacitor de 1000uF, ele funciona melhor que uma bateria pra prover energia por um curto espaço de tempo, mantendo assim o funcionamento estável do circuito. Vale ressaltar que o valor desse capacitor não pode ser muito alto a ponto de causar arcos voltaicos ao se conectar uma bateria, ou caso seja, é necessário usar um circuito para carregar o capacitor antes de ligar a fonte de energia.

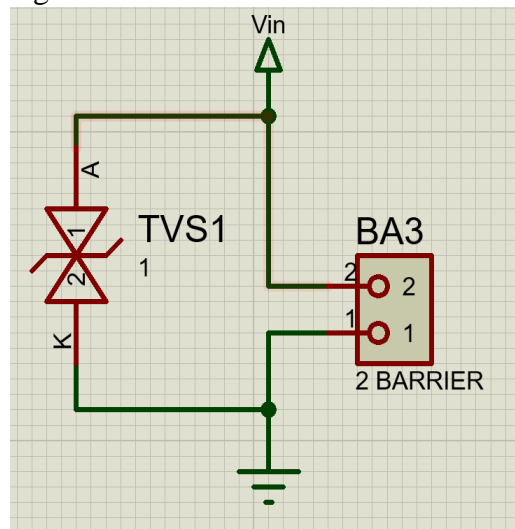
Figura 18 - Regulador chaveado



Fonte: Produção do próprio autor

Agora indo no sentido contrário, além da queda de tensão, pode também ocorrer o efeito de sobre tensão, já que o motor também funciona como um gerador, e para proteger o circuito desse efeito, é adicionado um diodo supressor de tensão (TVS), em paralelo com a alimentação, Figura 19, cortando qualquer surto de tensão acima do valor especificado, para o presente projeto foi utilizado o diodo 1.5KE35CA, que suprime picos de tensão acima de 35V.

Figura 19 - Diodo TVS

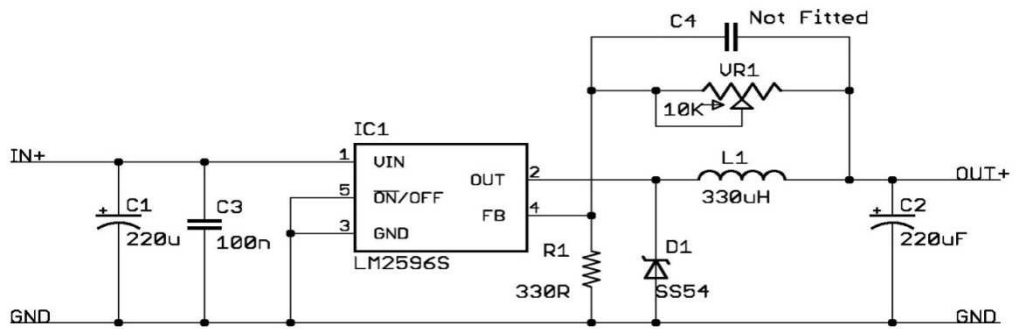


Fonte: Produção do próprio autor

### 3.1.6 Alimentação

Este projeto foi feito tendo em vista de que a alimentação seria feita através de uma bateria, podendo esta ter valores entre 12V e 34V, optou-se por utilizar um regulador chaveado do tipo “Step-Down Bulk”, o LM2596, Figura 20, as características desse regulador se encaixavam perfeitamente ao projeto, uma vez que ele pode ter tensões de entrada de até 37V, e consegue fornecer até 3A de saída.

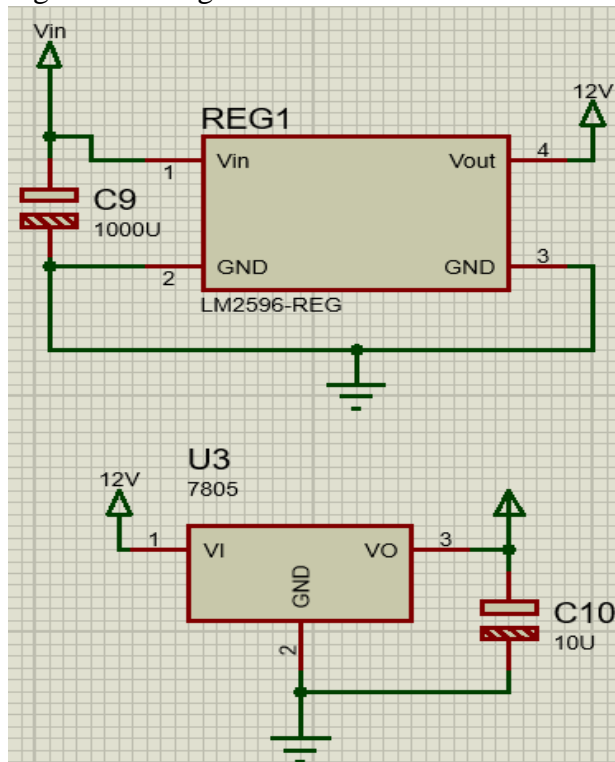
Figura 20 - Regulador Chaveado expandido



Fonte: Minikits website (2018).

Um inconveniente dos reguladores chaveados é o “ripple” que podem oferecer, por esse motivo foi escolhido utilizar um regulador linear na saída do regulador chaveado, Figura 21, somente para a parte do microcontrolador para garantir uma alimentação limpa, sem ruídos, para essa tarefa foi escolhido o LM7805, componente barato, de simples funcionamento e fácil de ser encontrado.

Figura 21 – Reguladores de tensão

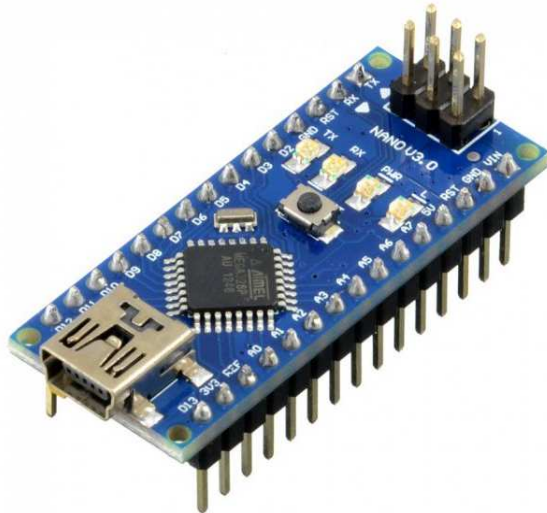


Fonte: Produção do próprio autor

### 3.1.7 Circuito Lógico

Para fazer a interface homem-máquina e para controlar os sinais de PWM, foi utilizado uma placa Arduino Nano, Figura 22, com um microcontrolador ATMEGA328P, o motivo da escolha dessa placa foi pelo ótimo custo benefício e facilidade de programação, uma vez que para a gravação da mesma não é necessário nenhum gravador externo diminuindo assim o custo do projeto.

Figura 22 – Arduino Nano



Fonte: Produção do próprio autor

Essa placa possui 6 pinos dedicados a PWM, sendo que parte deles funcionam com uma frequência de 490Hz e a outra parte com uma frequência de 980Hz, essa diferença de frequência não foi problema para o desenvolvimento desse projeto, pois as mudanças que ocorrem com frequências diferentes é o som que o motor emite ao chavear suas bobinas, e na dissipação térmica, o qual não se notou diferenças significativas.

Essa placa também possui 8 canais analógicos de 10 bits, que possibilita para o usuário o de uso diversos tipos de sensores como input (respeitando os limites de tensão de entrada, que no caso é 5V), para controlar suas aplicações, foi adicionado resistores de pull-up à essas entradas, pois muitos sensores precisam desse tipo de circuito para garantir o correto funcionamento.

### 3.1.8 Comunicação Bluetooth

Mais um diferencial desse projeto é o uso de comunicação Bluetooth, Figura 23, para facilitar e auxiliar, aumentando ainda mais as possibilidades de interação homem-máquina,



com a comunicação Bluetooth é possível fazer o controle dos motores às distâncias de até 10m, podendo assim utilizar o projeto em áreas que homem não pode se fazer presente.

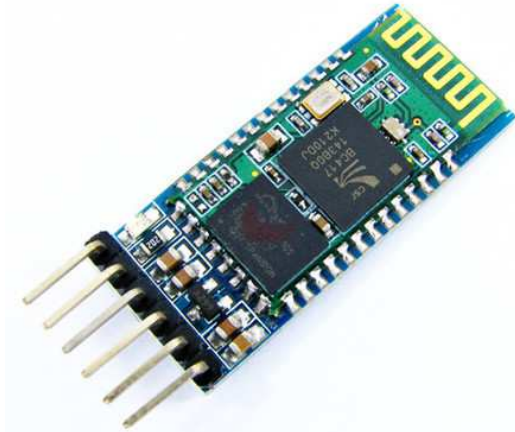


Figura 23 - Módulo Bluetooth HC-05

Fonte: Produção do próprio autor

Para incluir essa funcionalidade no projeto foi utilizado um módulo Bluetooth HC-05, que utiliza um padrão de comunicação serial UART, o que facilitou a integração com a placa Arduino, um detalhe importante é que como tanto a gravação do arduino, como o Bluetooth utilizam a comunicação serial do Arduino é imprescindível que no momento de descarregamento de um novo código a placa, o módulo Bluetooth esteja desconectado da mesma, para que não haja conflito de informações, para isso existe 2 jumpers que devem ser removidos no processo de gravação, e recolocados na hora de usar.

Existem diversos aplicativos prontos na PlayStore(loja de aplicativos da Google), que realizam esse tipo de comunicação, o que facilita ainda mais o uso da mesma. Sendo necessário apenas saber a informação que o aplicativo envia de acordo com cada comando(por exemplo, um botão clicado na tela de um smartphone ou tablet), e condicionar esse comando à uma ação no código(por exemplo girar um motor em um determinado sentido e uma determinada velocidade ao se apertar um botão no celular).

A velocidade de comunicação utilizada foi de 9600bps, esse valor assim como outras configurações do módulo, podem ser alteradas através dos chamados “comandos AT”, que são comandos específicos do módulo destinados as configurações internas do mesmo.

#### **4 APRIMORAMENTO – SEGUNDA VERSÃO**

O primeiro projeto que foi explicado acima foi utilizado durante a maior competição de robótica da América Latina, o Winter Challenge realizado pela empresa Robocore, e como o próprio slogan do evento diz, a tecnologia foi colocada à prova. A placa foi implementada em um robô de sumo autônomo, cujo objetivo desse robô é tirar o seu oponente de uma arena circular, feita de chapa de aço de 5mm, o que faz com todos os projetos utilizem ímãs de neodímio em suas bases, para aumentar suas forças normais, e assim ganhar tração e se defender, esse feito acaba por exigir bastante dos motores, e das eletrônicas que os controlam. Infelizmente durante umas das partidas uma das trilhas da placa se rompeu, Figura 24, por causa do excesso de corrente que acarretou num sobre aquecimento localizado, destruindo o cobre, essa trilha rompida ocasionou na queima dos Mosfets de um lado da ponte H, isso ocorreu, pois, a espessura de cobre na placa não era ideal, assim como a largura da trilha não foi suficiente. Devido a esse problema decidiu-se desenvolver um novo projeto que pudesse se prevenir esse tipo de problema, e assim surgiu a segunda versão, que será explicada brevemente, pois ainda está em fase de desenvolvimento.

Figura 24 - Trilha rompida



Fonte: Produção do próprio autor

#### 4.1 APRIMORAMENTOS

Durante pesquisas sobre os mais modernos dispositivos seccionadores voltados para motor DC escovado, foi descoberto os circuitos integrados da família BTNxxxx, fabricados pela Infineon, que são circuitos de meia ponte H, já com drivers de mosfets, e também diversos dispositivos de proteção integrados no mesmo encapsulamento. Um dos componentes dessa família, o BTN8982, chamou atenção pela alta capacidade de corrente suportada, que pode chegar à 90A, e este mesmo foi escolhido para integrar o novo projeto.

Dentre as proteções internas deste componente, pode-se citar, proteção contra sobrecorrente, proteção contra sobre e sob tensão, proteção contra aumento de temperatura, proteção contra curto circuito, e em todos esses casos para qualquer comportamento adverso o componente se desliga automaticamente, reporta em um de seus terminais uma flag/sinal indicando o tipo de falha que aconteceu. Além dessas proteções ele também indica em um de seus terminais o valor de corrente que está sendo drenado pela carga, e essa informação pode ser lida por um microcontrolador, através de uma porta analógica. O acionamento da carga, que no caso é um motor DC, pode ser feito através de um sinal PWM que ditam a velocidade de rotação do motor.

Como este componente trata-se de uma meia ponte H, é necessário 2 do mesmo para contruir uma ponte H com capacidade de controlar a rotação e velocidade do motor, sendo assim no projeto foram utilizados 4 BTN8982, Figura 25, para que fosse possível controlar 2 motores com total autoridade.

Figura 25 - BTN8982

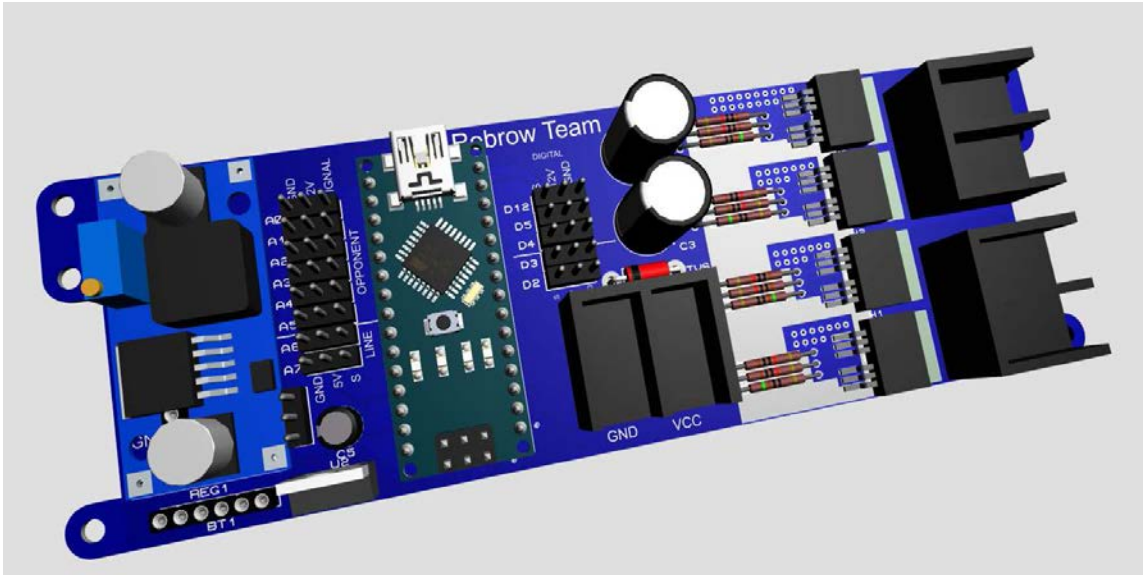


Fonte: Produção do próprio autor

Outra mudança no projeto foi a adição de mais um capacitor eletrolítico em paralelo à alimentação, pois durante o uso da placa aconteceram resets na parte lógica, devido aos altos de pico de corrente e tensão oriundos dos motores, essa melhoria já foi testada e aprovada.

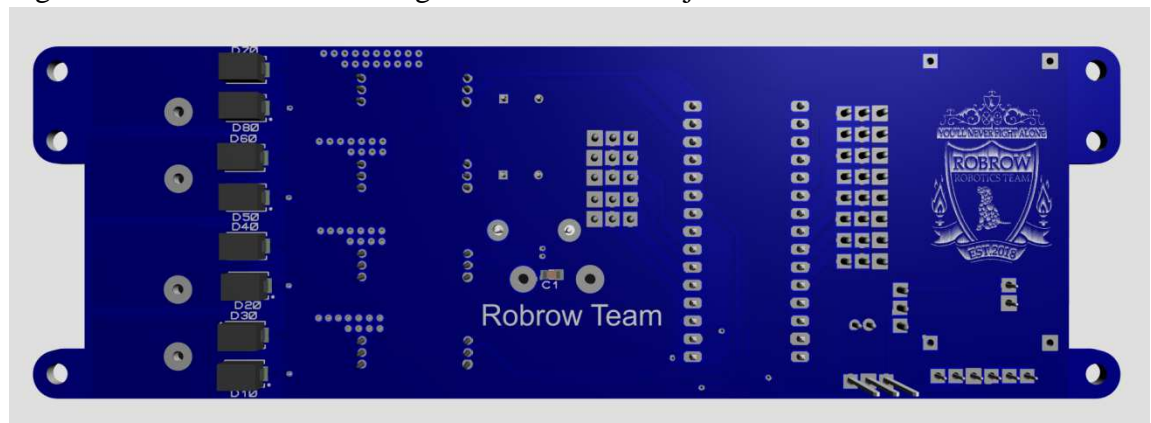
A parte lógica do circuito continuou sendo feita através de um Arduino Nano, pois o mesmo se mostrou eficiente, e atendeu a todos os requisitos do projeto. Também foram mantidas as funcionalidades para conexão de sensores externos, com resistores de pull up, assim como a conexão Bluetooth, essas implementações continuam a garantir ampla gama de aplicações para o presente projeto, Figura 26, e Figura 27.

Figura 26 - Visão 3D da Segunda Versão do Projeto



Fonte: Produção do próprio autor

Figura 27 - Visão inferior da Segunda Versão do Projeto

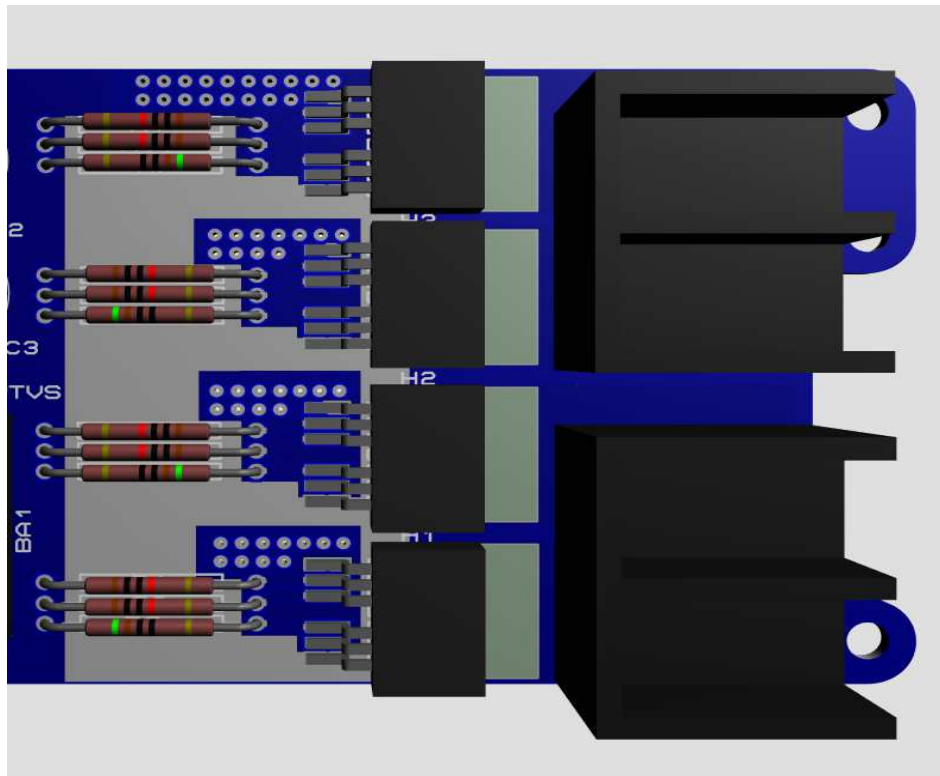


Fonte: Produção do próprio autor

Foi adicionada também a opção de se alimentar a parte lógica através de uma fonte de energia diferente da parte de potência, para casos em que a carga esteja gerando muitos ruídos e implicando em mau funcionamento do código.

Para garantir a capacidade total de circulação de corrente através do cobre da placa, foi feita uma exposição do cobre na região onde circula maior corrente, Figura 28, evitando o uso de verniz, para que pudesse ser adicionado depósito de estanho aumentando assim a área da seção transversal por onde circula corrente, e evitando sobre aquecimento das trilhas.

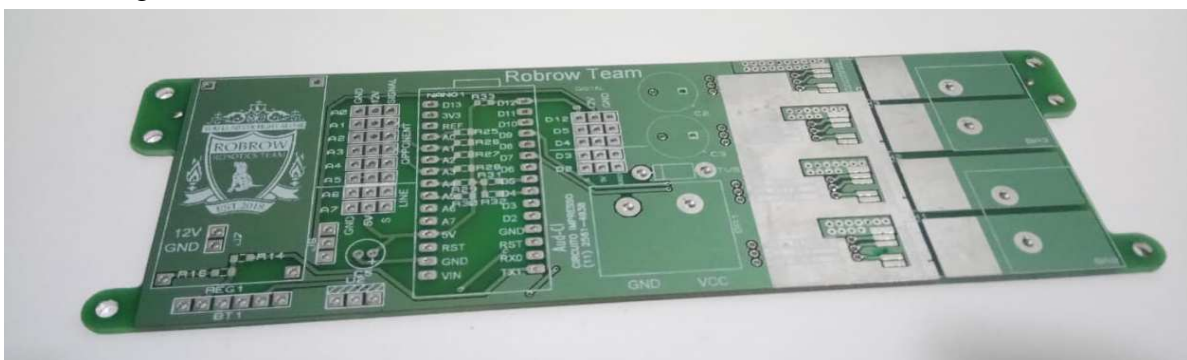
Figura 28 - Visão 3D dos BTNs e dos Conectores dos Motores



Fonte: Produção do próprio autor

Os aprimoramentos feitos nesse projeto visam garantir uma maior robustez e ampliação das aplicações possíveis para a placa. Sendo que todas essas mudanças já foram testadas e aprovadas individualmente, resultando na placa final confeccionada, Figura 29.

Figura 29–Placa confeccionada industrialmente



Fonte: Produção do próprio autor

## 5 CONCLUSÃO

O desenvolvimento de novos circuitos eletrônicos é parte fundamental da evolução humana, sempre que o panorama energético global muda com surgimento de novas formas de energia, a sociedade precisa se reinventar e muitas vezes se antecipar na criação e adaptação de dispositivos que façam uso dessas novas fontes energéticas.

O objetivo desse trabalho foi criar uma placa eletrônica que fosse versátil, capaz de controlar os mais diversos tipos de motores de corrente contínua com escova, com as mais diversas demandas de correntes e tensão, que fosse robusta e oferecesse confiabilidade e segurança ao usuário no momento do uso, além de ser interativa, com inúmeras possibilidades de comunicação entre homem e máquina.

Todos os requisitos foram atendidos com louvor, uma vez que a placa é compatível com motores de até 32V de tensão nominal, e que tenham demanda de corrente de no máximo 50A contínuos e 90A de pico; a placa possui circuitos de proteção contra sobre tensão, sobre corrente, e sobre temperatura; já para interação homem e máquina o projeto conta com inúmeras entradas analógicas e digitais, que permitem o uso de sensores diversificados, além de possuir também uma conexão Bluetooth, aumentando o campo de aplicações da placa, já que pode ser acionada à distância, garantindo a segurança do usuário em casos em que o ser humano não pode estar presente.

## REFERÊNCIAS

PROGRESSIVE AUTOMATIONS. **Brushed vs brushless dc motor:** an overview. Disponível em: <<https://www.progressiveautomations.com/brushed-vs-brushless-dc-motor-an-overview/>>. Acesso em: 15 nov. 2018.

MATLAB. **DC motor speed:** system modeling. Disponível em: <<http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling>>. Acesso em: 15 nov. 2018.

EPORSHE. **Motor break-in:** seating the brushes. Disponível em: <<http://eporsche911.blogspot.com/2012/04/motor-break-in-seating-brushes.html>>. Acesso em: 15 nov. 2018.

MINIKITS. **DC-DC modules.** Disponível em: <<https://www.minikits.com.au/components/power-supply/dc-dc-modules/LM2596-PSU-02>>. Acesso em: 15 nov. 2018.

GIPHY. **DC motor gif.** Disponível em: < <https://giphy.com/gifs/dc-motor-operation-hPqoa0nqdh292>>. Acesso em: 15 nov. 2018.

MAIMON, F. **Projeto de um sistema eletrônico para o controle de motores de alta potência por PWM.** Rio de Janeiro: Pontífica Universidade Católica, 2004.

FAIRCHILD SEMICONDUCTOR CORPORATION. **Design and application guide of bootstrap circuit for high-voltage gate-drive IC.** Disponível em:



<<https://www.fairchildsemi.com/application-notes/AN/AN-6076.pdf> >. Acesso em: 15 nov. 2018.

TEAM DELTA ENGINEERING. **Application note #1:** reducing motor noise. Disponível em: <<https://www.robotshop.com/media/files/pdf/motor-noise-reduction.pdf>>. Acesso em: 15 nov. 2018.

## BIBLIOGRAFIA CONSULTADA

SANTOS, F. M. V. **Robótica Industrial**. Disponível em  
<<http://www.ece.ufrgs.br/~rventura/RoboticaIndustrial.pdf>> Acesso em: 09 fev. 2017.

TCHAPEK, K. **A fábrica de robôs**. São Paulo: Hedra, 2010.

NORTON, R. L. **Projeto de máquinas: uma abordagem integrada**. 4. ed. Porto Alegre: Bookman, 2013.

MARGOLIS, M. **Arduino cookbook**. 2. ed. Sebastopol: O'Reilly, 2012.

MEGGIOLARO, A.M. **Tutorial em robôs de combate**. Disponível em  
<[http://www.robot.bmstu.ru/files/books/\[Robotic\]%20Tutorial%20RioBotz.pdf](http://www.robot.bmstu.ru/files/books/[Robotic]%20Tutorial%20RioBotz.pdf)> Acesso em:  
09 fev. 2017.

## ANEXO A – CÓDIGO

```

/*****

```

Vista Superior

Frente

```

*****
* F      E *
*****
*  C    B  *
*          *
* D      A *
*          *
*          *
*          *
*****

```

sensor\_word = 0-0-F-E-D-C-B-A (8-bits)

```

*****/

```

```

#define motor_A 11

```

```

#define motor_B 10

```

```

#define motor_C 6

```

```

#define motor_D 5

```

```

#define SDA 8

```

```

#define SDB 2

```

```

#define sensor_A A0

```

```

#define sensor_B A1

```

```

#define sensor_C A2

```

```

#define sensor_D A3

```

```

#define sensor_E A6

```

```

#define sensor_F A7

```

```
unsigned long tempo_antigo = 0;
int tempo_limite_wait = 5000;
int tempo_wait = 500;
int sensor_word = 0;

int found = 0;
int found_once = 0;

int threshold_A = 400;
int threshold_B = 400;
int threshold_C = 400;
int threshold_D = 400;
int threshold_E = 400;
int threshold_F = 400;

char EST_I = 'p';
char EST_D = 'a';
char EST_L = 'D';

int start = 0;
boolean LIGA = false;

int velocidade_ataque = 255;
int velocidade_giro = 200;
int velocidade_busca = 200;

int tempo_de_giro = 400;

int ultimo_estado = 0;

void motor_R(int velocidade, char sentido)
{
    if(sentido == 'f')
```

```
{
  analogWrite(motor_A, velocidade);
  analogWrite(motor_B, 0);
}
if(sentido == 't')
{
  analogWrite(motor_A, 0);
  analogWrite(motor_B, velocidade);
}
}

void motor_L(int velocidade, char sentido)
{
  if(sentido == 'f')
  {
    analogWrite(motor_C, velocidade);
    analogWrite(motor_D, 0);
  }
  if(sentido == 't')
  {
    analogWrite(motor_C, 0);
    analogWrite(motor_D, velocidade);
  }
}

void trata_sensores()
{
  if (analogRead(sensor_A)<threshold_A){ bitSet(sensor_word, 0); }
    else{ bitClear(sensor_word,0); }
  if (analogRead(sensor_B)<threshold_B){ bitSet(sensor_word, 1); }
    else{ bitClear(sensor_word,1); }
  if (analogRead(sensor_C)<threshold_C){ bitSet(sensor_word, 2); }
    else{ bitClear(sensor_word,2); }
  if (analogRead(sensor_D)<threshold_D){ bitSet(sensor_word, 3); }
```

```

        else{ bitClear(sensor_word,3); }
    if (analogRead(sensor_E)<threshold_E){ bitSet(sensor_word, 4); }
        else{ bitClear(sensor_word,4); }
    if (analogRead(sensor_F)<threshold_F){ bitSet(sensor_word, 5); }
        else{ bitClear(sensor_word,5); }
}

int delay_sensor(int tempo)
{
    found_once = 0;
    tempo_antigo = millis();
    while(((millis() - tempo_antigo)<=tempo))
    {
        if (analogRead(sensor_A)<threshold_A){ bitSet(sensor_word, 0); }
            else{ bitClear(sensor_word,0); }
        if (analogRead(sensor_B)<threshold_B){ bitSet(sensor_word, 1); }
            else{ bitClear(sensor_word,1); }
        if (analogRead(sensor_C)<threshold_C){ bitSet(sensor_word, 2); }
            else{ bitClear(sensor_word,2); }
        if (analogRead(sensor_D)<threshold_D){ bitSet(sensor_word, 3); }
            else{ bitClear(sensor_word,3); }
        if (analogRead(sensor_E)<threshold_E){ bitSet(sensor_word, 4); }
            else{ bitClear(sensor_word,4); }
        if (analogRead(sensor_F)<threshold_F){ bitSet(sensor_word, 5); }
            else{ bitClear(sensor_word,5); }

        if (sensor_word > 0)
        {
            found_once = 1;
            break;
        }
    }
    return found_once;
}

```

```
void Pica_Pau(char lado)
{
  switch (lado)
  {
    case 'D':
      motor_L(220,'f');
      motor_R(220,'t');
      if(delay_sensor(40))break;
      break;
    case 'E':
      motor_L(220,'t');
      motor_R(220,'f');
      if(delay_sensor(40))break;
      break;
    case 'F':
      break;
  }
  while(true)
  {
    motor_L(125,'f');
    motor_R(125,'f');
    if(delay_sensor(60))break; // se encontrar adversário ou linha sai do loop
    motor_L(0,'f');
    motor_R(0,'f');
    if(delay_sensor(2000))break; // se encontrar adversário ou linha sai do loop
  }
}

void Curva(char lado)
{
  switch(lado)
  {
    case 'D':
      motor_L(220,'f');
```

```

    motor_R(220,'t');
    if(delay_sensor(40))break;
    while(sensor_word==0)
    {
        trata_sensores();
        motor_L(150,'f');
        motor_R(200,'f');
    }
    break;
case 'E':
    motor_L(220,'t');
    motor_R(220,'f');
    if(delay_sensor(40))break;
    while(sensor_word==0)
    {
        trata_sensores();
        motor_L(200,'f');
        motor_R(150,'f');
    }
    break;
case 'F':
    while(sensor_word==0)
    {
        trata_sensores();
        motor_L(200,'f');
        motor_R(200,'f');
    }
    break;
}
}

void Agressiva()
{
    trata_sensores();

```



```
if (sensor_word >15 && sensor_word < 32)
{
    motor_L(velocidade_ataque,'t');
    motor_R(velocidade_ataque,'t');
    delay_sensor(100);
    motor_L(velocidade_giro,'t');
    motor_R(velocidade_giro,'f');
    delay_sensor(50);
}
else if(sensor_word >= 32 && sensor_word <=47)
{
    motor_L(velocidade_ataque,'t');
    motor_R(velocidade_ataque,'t');
    delay_sensor(100);
    motor_L(velocidade_giro,'f');
    motor_R(velocidade_giro,'t');
    delay_sensor(50);
}
else if(sensor_word>47)
{
    motor_L(velocidade_ataque,'t');
    motor_R(velocidade_ataque,'t');
    delay_sensor(100);
    motor_L(velocidade_giro,'f');
    motor_R(velocidade_giro,'t');
    delay_sensor(50);
}
else
{
    switch(sensor_word)
    {
        case 0:
            if (ultimo_estado==2)
            {
```

```
    motor_L(velocidade_giro,'f');
    motor_R(velocidade_giro,'t');
    delay_sensor(tempo_de_giro);
}
if(ultimo_estado == 4)
{
    motor_L(velocidade_giro,'t');
    motor_R(velocidade_giro,'f');
    delay_sensor(tempo_de_giro);
}
motor_L(velocidade_busca,'f');
motor_R(velocidade_busca,'f');
break;
case 1: motor_L(velocidade_giro,'f');
        motor_R(velocidade_giro,'t');
        delay_sensor(tempo_de_giro);
        break;
case 2: motor_L(velocidade_ataque,'f');
        motor_R(velocidade_ataque,'f');
        break;
case 3: motor_L(velocidade_ataque,'f');
        motor_R(velocidade_ataque,'f');
        break;
case 4: motor_L(velocidade_ataque,'f');
        motor_R(velocidade_ataque,'f');
        break;
case 5: motor_L(velocidade_giro,'f');
        motor_R(velocidade_giro,'t');
        break;
case 6: motor_L(velocidade_ataque,'f');
        motor_R(velocidade_ataque,'f');
        break;
case 7: motor_L(velocidade_giro,'f');
        motor_R(velocidade_ataque,'t');
```

```
        break;
    case 8: motor_L(velocidade_giro,'t');
            motor_R(velocidade_giro,'f');
            delay_sensor(tempo_de_giro);
            break;
    case 9: break;
    case 10: motor_L(velocidade_giro,'t');
            motor_R(velocidade_giro,'f');
            break;
    case 11: motor_L(velocidade_ataque,'f');
            motor_R(velocidade_ataque,'f');
            break;
    case 12: motor_L(velocidade_ataque,'f');
            motor_R(velocidade_ataque,'f');
            break;
    case 13: motor_L(velocidade_ataque,'f');
            motor_R(velocidade_ataque,'f');
            break;
    case 14: motor_L(velocidade_ataque,'t');
            motor_R(velocidade_giro,'f');
            break;
    case 15: break;
}
}
ultimo_estado = sensor_word;

}

void Wait(char lado)
{
    unsigned long t= millis();
    unsigned long t_1 = millis();

    int flag_ataque = 0;
```

```
switch (lado)
{
  case 'D':
    motor_L(220,'f');
    motor_R(220,'t');
    if(delay_sensor(40))break;
    break;
  case 'E':
    motor_L(220,'t');
    motor_R(220,'f');
    if(delay_sensor(40))break;
    break;
  case 'F':
    break;
}
while(((millis() - t)<= tempo_wait) || (millis() - t_1)<=tempo_limite_wait || !flag_ataque)
{
  trata_sensores();
  switch(sensor_word)
  {
    case 0: t = millis();
      break;
    case 1: motor_L(velocidade_giro,'f');
      motor_R(velocidade_giro,'t');
      break;
    case 2: motor_L(velocidade_giro,'f');
      motor_R(velocidade_giro,'t');
      break;
    case 3: motor_L(velocidade_giro,'f');
      motor_R(velocidade_giro,'t');
      break;
    case 4: motor_L(velocidade_giro,'t');
      motor_R(velocidade_giro,'f');
      break;
  }
}
```

```
case 5: break;
case 6: motor_L(velocidade_ataque,'f');
        motor_R(velocidade_ataque,'f');
        flag_ataque = 1;
        break;
case 7: motor_L(velocidade_giro,'f');
        motor_R(velocidade_giro,'t');
        break;
case 8: motor_L(velocidade_giro,'t');
        motor_R(velocidade_giro,'f');
        break;
case 9: break;
case 10: motor_L(velocidade_giro,'t');
        motor_R(velocidade_giro,'f');
        break;
case 11: motor_L(velocidade_giro,'f');
        motor_R(velocidade_giro,'t');
        break;
case 12: motor_L(velocidade_giro,'t');
        motor_R(velocidade_giro,'f');
        break;
case 13: motor_L(velocidade_giro,'t');
        motor_R(velocidade_giro,'f');
        break;
case 14: motor_L(velocidade_giro,'t');
        motor_R(velocidade_giro,'f');
        break;
case 15: break;
}
}
}

void Zigue_Zague(char lado){ }
```

```
void estrategia(char inicio, char durante, char lado)
{
  start++;
  if(start==1)
  {
    switch(inicio)
    {
      case 'p': Pica_Pau(lado);  break;
      case 'z': Zigue_Zague(lado); break;
      case 'c': Curva(lado);    break;
      case 'w': Wait(lado);     break;
    }
  }
  switch(durante)
  {
    case 'a': Agressiva();  break;
    case 'p': Pica_Pau(lado);  break;
    case 'c': Curva(lado); break;
  }
}

void setup()
{
  Serial.begin(9600);
  // Serial.print("*SR255G255B255*");
  // Serial.print("*IR255G255B255*");
  // Serial.print("*DR255G255B255*");
  // Serial.print("*LR255G255B255*");
  pinMode(motor_A, OUTPUT);
  pinMode(motor_B, OUTPUT);
  pinMode(motor_C, OUTPUT);
  pinMode(motor_D, OUTPUT);
  pinMode(SDA, OUTPUT);
  pinMode(SDB, OUTPUT);
  digitalWrite(SDA,HIGH);
```

```
digitalWrite(SDB,HIGH);
motor_L(0,'f');
motor_R(0,'f');
}

void loop()
{
  if(LIGA)
  {
    estrategia(EST_I, EST_D, EST_L);
  }
  else
  {
    motor_L(0,'f');
    motor_R(0,'f');
  }
}

void serialEvent()
{
  if(Serial.available())
  {
    char BluetoothData = (char)Serial.read();
    if (BluetoothData == 'L' && LIGA == false)
    {
      Serial.print("*SR0G255B0*");
      delay(4800);
      LIGA = true;
      digitalWrite(3,HIGH);
      //break;
    }
    if (BluetoothData == 'I' && LIGA == true)
    {
      digitalWrite(3,LOW);
      Serial.print("*SR255G0B0*");
    }
  }
}
```

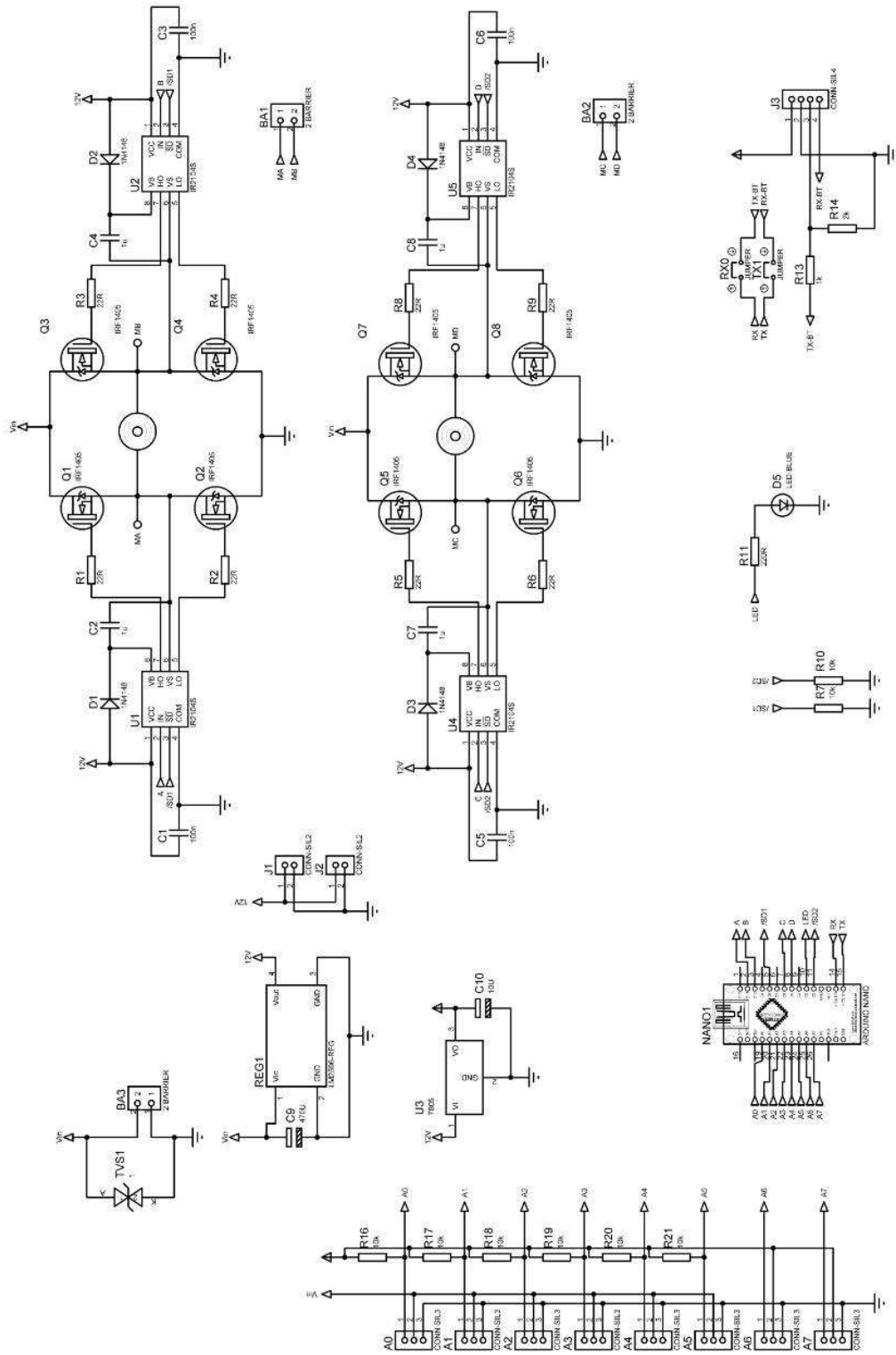
```
Serial.print("*IR255G255B255*");
Serial.print("*DR255G255B255*");
Serial.print("*LR255G255B255*");
LIGA = false;
start = 0;
//break;
}
if (BluetoothData == 'O')
{
  EST_I = 'w';
  Serial.print("*IR255G0B255*");
}
if (BluetoothData == 'A')
{
  EST_I = 'p';
  Serial.print("*IR255G0B0*");
}
if (BluetoothData == 'B')
{
  EST_I = 'z';
  Serial.print("*IR0G255B0*");
}
if (BluetoothData == 'C')
{
  EST_I = 'c';
  Serial.print("*IR0G0B255*");
}
if (BluetoothData == '1')
{
  EST_D = 'a';
  Serial.print("*DR255G0B0*");
}
if (BluetoothData == '2')
{
```



```
EST_D = 'p';
Serial.print("*DR0G255B0*");
}
if (BluetoothData == '3')
{
EST_D = 'c';
Serial.print("*DR0G0B255*");
}
if (BluetoothData == 'E')
{
EST_L = 'E';
Serial.print("*LR255G0B0*");
}
if (BluetoothData == 'D')
{
EST_L = 'D';
Serial.print("*LR0G255B0*");
}
if (BluetoothData == 'F')
{
EST_L = 'F';
Serial.print("*LR0G0B255*");
}
}
}
```

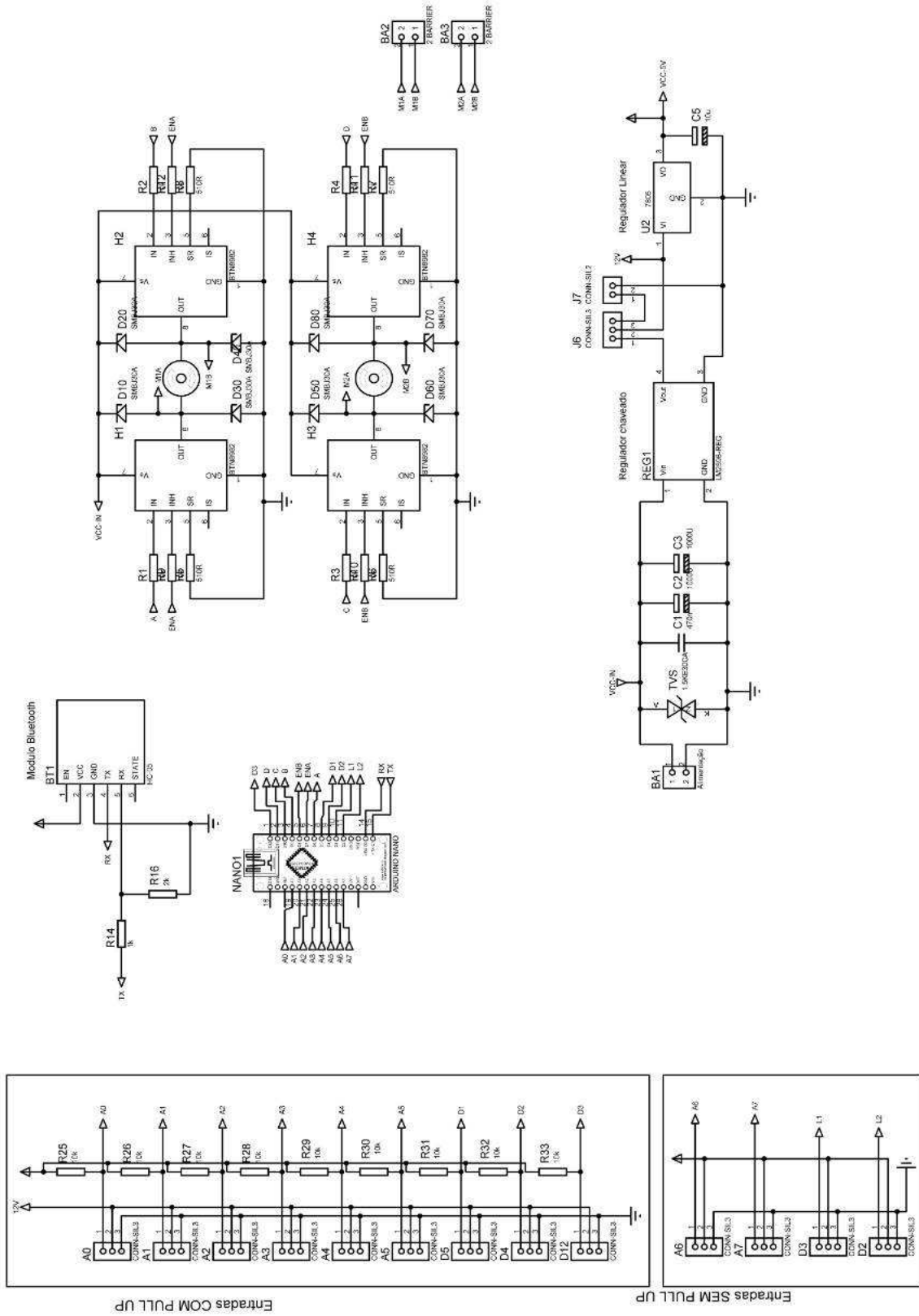
Fonte: Produção do próprio autor.

## ANEXO B – ESQUEMA ELÉTRICO DO PRIMEIRO PROJETO



Fonte: Produção do próprio autor.

## ANEXO C – ESQUEMA ELÉTRICO DO SEGUNDO PROJETO



Fonte: Produção do próprio autor.