



ETEC JORGE STREET

TRABALHO DE CONCLUSÃO DE CURSO TÉCNICO EM MECATRÔNICA

Adriano Oliveira
Antônio Junior
Eduardo Gianoto
Luan Caliel

EMPILHADEIRA ROBÓTICA

São Caetano do Sul
Junho/2017

Adriano Oliveira
Antônio Junior
Eduardo Gianoto
Luan Caliel

EMPILHADEIRA ROBÓTICA

Trabalho de Conclusão de Curso
apresentado como exigência parcial
para obtenção do Diploma do Curso
Técnico de Mecatrônica da ETEC
Escola Técnica Estadual Jorge
Street.

Professor Orientador: Ricardo Arroio

São Caetano do Sul

Junho/2017

Adriano Oliveira
Antônio Junior
Eduardo Gianoto
Luan Caliel

EMPILHADEIRA ROBÓTICA

Trabalho de conclusão de curso
apresentado como exigência parcial
para obtenção do Diploma do Curso
Técnico em Mecatrônica.

Professor Orientador: Ricardo Arroio

Aprovado pela Banca Examinadora em 06 de junho de 2017.

BANCA EXAMINADORA:

Professor(a)

Professor(a)

Professor(a)

RESUMO

A empilhadeira é um dos meios mais utilizados no dia a dia para o transporte de cargas. O meio de transporte atualmente é manual, ou seja, um operador mal treinado e que não sabe utilizar a empilhadeira pode gerar um grande problema ao seu local de trabalho como uma perda de estoque, danos à carga transportada e também até trabalhadores que estejam no local. Nesse projeto será apresentada uma empilhadeira, mostrando seu funcionamento e componentes utilizados para a sua montagem. O objetivo do projeto é realizar uma empilhadeira robótica com sistema de transporte inteligente, prático e eficaz visando sempre à rapidez e a segurança da carga, seguindo um trajeto programado onde a carga será estabelecida no local e o ponto de entrega também fazendo assim um transporte sem a necessidade de mão de obra humana. Para a realização do projeto será utilizado princípios da elétrica, mecânica, robótica e programação por meio de microcontrolador.

Palavras-chave: Empilhadeira, Transporte, Carga.

ABSTRACT

Dump truck is one of the most used means for day-to-day cargo transportation. The medium transport is currently manual, whereas, a maltreated operator who does not know how to use the forklift can cause a major problem to his/her workplace as a loss, damage to the cargo transported and also to workers who are in the place. In this project it will be presented a mini forklift, showing its operation and components used for its assembly. The project objective is to realize a robotic mini-stacker with intelligent, practical and effective transport system always aiming the speed and security of the load, following a programmed route where the load will be established in the place and the point of delivery also doing so a transport without the need of human labor. For the realization of this project, it will be used electric principles, mechanics, robotics and programming by means of microcontroller.

Keywords: Forklift. Transportation, Freight.

SUMÁRIO

1. INTRODUÇÃO	11
1.1. JUSTIFICATIVA	11
1.2. OBJETIVO.....	11
2. ESTRUTURA ELÉTRICA	12
2.1 ARDUINO.....	12
2.2 MÓDULO BLUETOOTH LOW ENERGY – HC-08	13
2.3 MÓDULO RELÉ	14
2.4 BATERIA UNIPOWER 12V 7A.h.....	16
2.5 FITA DE LED.....	17
3. ESTRUTURA MECÂNICA.....	18
3.1 MOTOR DE CORRENTE CONTÍNUA	18
3.2 FUSO TRAPEZOIDAL.....	19
3.4 ELEVAÇÃO DA CARGA	21
3.5 MOVIMENTAÇÃO E SUPORTE	24
3.6 SISTEMA DE CARGA	25
4. EXPLICAÇÃO DA PROGRAMAÇÃO	26
5. DESENVOLVIMENTO DO PROJETO	34
6. ESQUEMA ELÉTRICO.....	41
7. CONCLUSÃO.....	43
8. CRONOGRAMA.....	44
9. ESPECIFICAÇÕES.....	45
10. TABELA DE CUSTOS.....	46
11. DIAGRAMA EM BLOCOS.....	47
12. FLUXOGRAMA DE PROGRAMAÇÃO.....	48
13. PROGRAMAÇÃO DO ARDUINO	49
14. REFERÊNCIAS.....	63

LISTA DE ILUSTRAÇÕES

Figura 1 - Arduino.....	12
Figura 2 - Módulo Bluetooth Low energy HC-08.....	14
Figura 3 - Módulo relé.....	14
Figura 4 - Ponte H relé.....	15
Figura 5 - Ponte H relé.....	15
Figura 6 - Ponte H relé.....	16
Figura 7 - Bateria Unipower 12 V 7.A.h.....	16
Figura 8 – Fita de Led.....	17
Figura 9 - Motor DC 12V.....	18
Figura 10 - Fuso Trapezoidal.....	19
Figura 11 - Tarugo de alumínio.....	20
Figura 12 – Peça de alumínio.....	20
Figura 13 - Usinagem da peça com broca.....	21
Figura 14 - Suporte de alumínio.....	23
Figura 15 - Eixo guia.....	23
Figura 16 - Sistema de elevação de carga.....	24
Figura 17 - Motor de elevação.....	25
Figura 18 - Interface BlackBLE.....	28
Figura 19 - Madeira MDF.....	34
Figura 20 - Base do projeto.....	34
Figura 21 - Furos laterais.....	35
Figura 22 - Posicionamento do motor.....	35
Figura 23 - Testes com motores.....	36
Figura 24 - Posicionamento de rodas.....	36
Figura 25 - Polarização.....	37
Figura 26 - Base da empilhadeira.....	37
Figura 27 - Placa controladora.....	38
Figura 28 - Instalação dos relés.....	38
Figura 29- Fuso Trapezoidal.....	39
Figura 30 - Pás da Empilhadeira.....	39
Figura 31 - Fixação das pás de sustentação.....	40
Figura 32 – Bateria.....	40

Figura 33- Esquema elétrico.....41

LISTA DE TABELAS

Tabela 1 – Cronograma.....	44
Tabela 2 – Especificações.....	45
Tabela 3 - Tabela de custo.....	46

LISTA DE SIGLAS

CC- Corrente Continua.

CA- Corrente Alternada.

V- Volts.

A- Ampère.

mA- Mili-Ampere.

RPM- Rotações Por Minuto.

Kgf.cm- Quilograma Força Centímetro.

NA- Normalmente Aberto.

NF- Normalmente Fechado.

C- Comum.

GHz- Giga Hertz.

VCC- Tensão Corrente Continua.

GND- Terra.

IN- Entrada.

RX- Sinal de Entrada.

TX- Sinal Transmitido.

1. INTRODUÇÃO

A maioria dos processos industriais constantemente requer transporte de mercadorias que chegam a pesar toneladas dependendo do material transportado e para esse transporte é desejável um equipamento seguro e eficiente no seu procedimento de tal modo realizar essa função com eficiência a movimentações, transporte e suspensão são chamadas empilhadeiras. Evitando assim risco de acidentes como tombamentos de mercadoria onde geraria um prejuízo grande a empresa, risco de queda de carga em cima de pedestre podendo causar a morte e afastamento temporário ou definitivo por invalidez.

1.1. JUSTIFICATIVA

A ideia desse projeto surgiu visando a tornar o processo logístico mais eficiente, cada vez mais rápido e seguro. Uma empilhadeira automatizada é capaz de executar determinados movimentos em setores onde os espaços são reduzidos onde a movimentação da empilhadeira fica limitado assim evitando o risco de instabilidade da carga pela falta de espaço, e com a empilhadeira autônoma seus movimentos acabam sendo precisos para a eficiência do transporte.

1.2. OBJETIVO

Melhorar as condições de transportes de uma forma autônoma e mais eficiente com o intuito de reduzir quedas de cargas, aumentar a logística de distribuição de uma forma mais eficiente e com um menor tempo de organização da carga, evitando os atropelamentos de funcionários em locais de serviços.

1.3. METODOLOGIA

Este tópico contém a pesquisa realizada ao longo do projeto para o desenvolvimento da empilhadeira robótica. O início desta pesquisa foi iniciado buscando as informações de outra empilhadeira robótica e seu funcionamento, fazendo assim uma melhoria no preço e dando uma facilidade ao seu manuseio.

2. ESTRUTURA ELÉTRICA

A estrutura elétrica do projeto é composta por diversos componentes, entre eles temos:

- Arduino;
- Bluetooth Shield;
- Módulo Rele;
- Bateria.

Onde cada um possui uma função excepcional no projeto.

2.1 ARDUINO

No projeto optou-se por a utilização do Arduino (figura 1) por ele ser responsável pela comunicação entre os relés e o motor. As informações são analisadas pelo microcontrolador e mandadas por meio de comandos para os motores, onde realizam a movimentação.



Figura 1: Arduino

Fonte: Robocore.

O Arduino basicamente é uma placa de prototipagem eletrônica de código aberto que inclui hardware e software livre, seu software é desenvolvido por meio de linguagem C/C++, onde ele é utilizado em projetos mais simples ou até em projetos mais complexos onde oferece ferramentas adaptáveis e de baixo custo para a criação de projetos, sua alimentação é feita pela porta USB onde é conectada ao computador ou por uma fonte externa de até 25 V dependendo do Arduino a ser utilizado.

O Componente escolhido no projeto é o Arduino UNO pela sua simplicidade e sua fácil compatibilidade com outros Shields.

2.2 MÓDULO BLUETOOTH LOW ENERGY – HC-08

O Bluetooth é amplamente utilizado em todo o mundo, para enviar arquivos para um celular ou até para uma impressora. A transmissão de dados é feita por meio de radiofrequência onde sua transmissão é curta (2.4 GHz a 2,5 GHz), a comunicação se dá através de uma rede chamada PICONET. Nessa comunicação o dispositivo que iniciou a conexão assume o papel de *Master*(chefe), enquanto que os demais se tornam *Slaves*(escravos) o *Master* tem como função a transmissão de dados na rede e o sincronismo entre os dispositivos.

No projeto é utilizado o Módulo Bluetooth Low Energy – HC-08, que utiliza a versão Bluetooth 4.0. Esta versão tem como funcionalidade trabalhar com dispositivos portáteis e que lidam com pouca energia. Este módulo é ideal para controlar os relés e realizar a comunicação com o celular, onde é utilizado o aplicativo *BlackBLE*, este dispositivo só se comunica com o sistema operacional iOS.

No módulo Bluetooth encontramos 4 pinos onde são caracterizados por: VCC, GND, RX E TX. O VCC e GND podem ser ligados direto no Arduino ou em uma fonte externa, já o RX e TX servem para realizar a comunicação entre o Arduino.



Figura 2: Módulo Bluetooth Low Energy – HC-08

Fonte: Robocore.

2.3 MÓDULO RELÉ

O relé basicamente é um interruptor eletromecânico, ou seja, quando energizado a corrente elétrica percorre as bobinas e cria um campo magnético que atrai a alavanca que é responsável pela mudança dos contatos.

Seu funcionamento é simples, quando o contato C é conectado com o NF estando ambos energizados, a bobina cria um campo magnético que move o contato C e conecta com o contato NA e conseqüentemente desconecta com contato NF.

O relé utilizado no projeto é um módulo relé que é um pouco diferente do relé convencional, o que muda basicamente são três pinos onde nesses pinos contém o GND, 5 V e o IN e também, temos um LED que basicamente mostra quando o relé estará ligado, ou seja, quando o contato C estiver ligado ao contato NA, os pinos GND e 5 V podem ser alimentados direto no Arduino ou em uma alimentação externa. Já o pino IN é conectado em uma saída digital do Arduino que serve para a comunicação entre eles.



Figura 3: Módulo Relé

Fonte: Robocore.

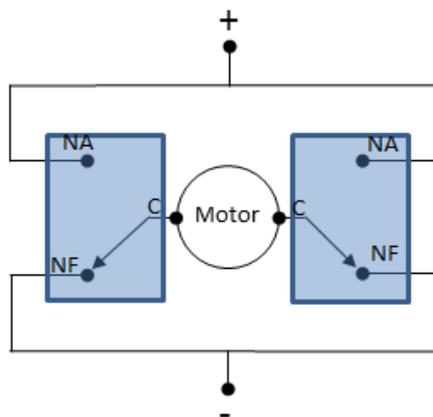


Figura 4: Ponte H com Relé

Fonte: TECNOMELQUE. Blogspot.

Com 2 relés montamos uma ponte H, ou seja, interligamos os contatos NA e também os contatos NF, nos contatos NA e NF alimentamos com uma fonte de energia externa, no contato NA ligamos o positivo e no NF o negativo, já no comum ligamos os dois cabos no C um em cada relé, neste estado o motor estará ligado pois estará recebendo negativo.

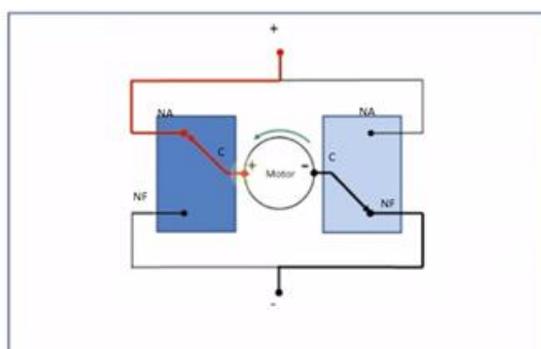


Figura 5: Ponte H com Relé

Fonte: TECNOMELQUE. Blogspot.

Já na imagem abaixo (figura 6) temos o inverso, o relé da direita é ativado e a parte da esquerda do motor passa a ser negativa e a parte direita positiva, com isso é possível fazer com que o motor gire para o sentido horário.

2.5 FITA DE LED

Essa fita de LEDs RGB SMD é ideal para aplicações em ambientes internos (classe de proteção IP20) como iluminação de destaque em vitrines, sinalizadores / indicadores, decoração de equipamentos e outras aplicações de iluminação. Seu diferencial é que possui LEDs que acendem verde, vermelho ou azul. Utilizando técnicas de PWM (com o circuito apropriado) é possível ainda misturar as três cores para se obter qualquer cor no espectro RGB.

Esta fita LED possui 60 LEDs a cada metro linear, com um consumo de 14,4W. Utilizando a eletrônica correta, pode ser dimerizável.

Para facilitar a instalação, a fita possui adesivo 3M na parte traseira.



Figura 8: Fita de led

Fonte: Robocore.

Características:

- Quantidade de LEDs por metro: 60 LEDs.
- Tensão de alimentação (Direta): 12V.
- Potência máxima consumida por tira (3 leds): 0,72W.
- Potência máxima consumida por metro (60 leds) : 14,4W.
- Classe de proteção IP20.
- Vendido por metro.

3. ESTRUTURA MECÂNICA

Primeiramente, foram estudadas as estruturas mecânicas das empilhadeiras convencionais para a confecção da Empilhadeira Robótica.

Ela deve possuir um bom acabamento para não comprometer o desempenho do sistema de modo geral. Desta forma, foram estudados modelos já existentes para a confecção e montagem do protótipo.

3.1 MOTOR DE CORRENTE CONTÍNUA

Existem dois tipos de motores os de corrente contínua (CC) e os de corrente alternada (CA), cada um com uma função diferente, os motores de corrente alternada basicamente são usados para máquinas de alto porte por receber energia diretamente da rede de distribuição de energia, já os motores de corrente contínua são usados em robôs e máquinas de porte pequeno, pois sua alimentação se dá por uma bateria.



Figura 9: Motor DC 12V

Fonte: Robocore.

Os motores de corrente contínua são constituídos pelos ímãs norte e sul, quando a corrente elétrica passa através da bobina e do campo magnético do ímã é gerada um campo magnético que produz o torque com isso resulta no acionamento do motor CC.

No projeto precisávamos de um motor com um alto torque pelo simples fato da empilhadeira robótica ter um peso de em média 6 quilos, com este fator foi escolhido o motor AK555/11.1PF12R83CE-V2 com caixa de redução onde possui um torque de 11,1 kgf.cm, uma corrente de 430 mA, picos de 1.6 A e sua velocidade é de 83rpm sendo ideal para sua utilização no projeto.

3.2 FUSO TRAPEZOIDAL

Para a função da empilhadeira utilizamos um fuso trapezoidal onde é uma barra reta formada por roscas e passos contínuos com um ângulo de 30° , é utilizado na execução de processos indústrias que exigem um movimento tanto linear quanto de posicionamento.



Figura 10: Fuso Trapezoidal

Fonte: TDTEC.

Como mostrado na imagem acima o fuso apresenta dois mancais que são dispositivos mecânicos fixos, ou seja, são usados para assegurar movimentação rotativa entre duas superfícies.

3.3 PROCESSO DE USINAGEM

A fabricação do mecanismo de elevação de carga foram feitas das seguintes maneiras:

- Divisão do tarugo De alumínio em três partes;



Figura 11: Tarugo de Alumínio
Fonte: Autoria própria

- Fresamento das faces e do topo das peças até às medidas finais;



Figura 12: Peça de Alumínio
Fonte: Autoria Própria.

Usando a furadeira de bancada foram feitos furos na peça para fixação dos mancais onde usamos uma broca de 28 mm. Foram feitos também os furos de fixação dos eixos usando uma broca de diâmetro 4,2 mm para ser usado o macho M5x0,8 fazendo assim a rosca M5.

Usinagem das peças superior e inferior na fresadora vertical usando a fresa de topo de 10 mm de diâmetro. Após a usinagem giramos o cabeçote da fresadora em 5 graus para ser feito um ângulo de perfeito encaixe do mancal, em seguida com o cabeçote da fresadora retornado ao 0 grau e usando a mesma ferramenta foram feitos os rebaixos.



Figura 13: Usinagem da peça com a broca

Fonte: Etec Jorge Street.

Na furadeira de bancada foram feitos os furos dos eixos guias onde usamos uma broca de 8 mm na parte inferior e superior da peça, onde também foram feitos os furos de 13 mm de diâmetro para o acoplamento das buchas. No torno mecânico foi utilizado uma broca de 8,8 mm de diâmetro para a perfuração das buchas. Foi utilizado um alargador manual com 9mm de diâmetro dente helicoidal H7 para um melhor deslizamento e menos atrito. Foram feitos 5 furos na peça inferior para fixação no suporte de madeira do carrinho, em seguida foi feita a junção das peças para a montagem do sistema mecânico de elevação de carga.

3.4 ELEVAÇÃO DA CARGA

O sistema mecânico de elevação foi baseado em um sistema muito utilizado em máquinas cnc, impressora 3D etc. Usamos esse método por ser mais relevante ao material do projeto pois dessa forma a fixação do sistema de levantamento na posição vertical foi feito diretamente na peça inferior, tendo como objetivo a facilidade de retirada para manutenção.

O modelo conta com uma torre de elevação de até 200 mm, por ter um esquema mecânico de fuso de 300 mm, ele apenas fornece uma altura abaixo de seu tamanho.

Os materiais utilizados no sistema de elevação foram:

- Fuso trapezoidal 300 mm Tr8-8m;
- 2 guias lineares 8 mm;
- Mancal KP08;
- 2 Castanhas;
- 2 bucha 9 mm interno;
- Suporte de pillow block de 10mm (sc10uu);
- 2 parafusos M3 45 mm com porca;
- 4 parafusos M3 50 mm com porca;
- 5 parafusos M4 40 mm com arruelas e porcas;
- 6 parafusos m4 80 mm com arruelas e porcas;
- 4 parafusos M5 45 mm;
- 4 parafusos M5 15 mm com arruelas e porcas;
- 4 parafusos M5 10 mm;
- 5 parafusos;
- 1 acoplador 6 x 8;
- 2 peças de alumínio 13 mm x 120 mm x 50 mm (peças inferior e superior);
- 1 peça de alumínio 17 mm x 120 mm x 50 mm (peça meio).



Figura 14: Suporte de alumínio feitas na Fresadora
Fonte: Aatoria própria.

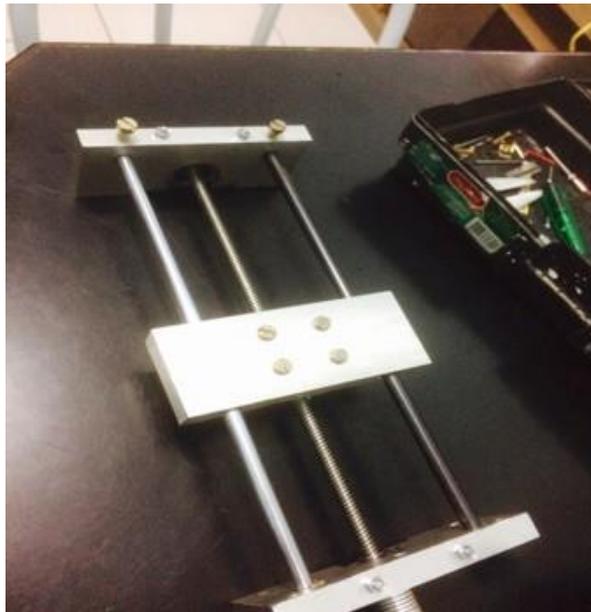


Figura 15: Eixos guias com fuso central para elevação de cargas.
Fonte: Aatoria própria.



Figura 16: Sistema de levantamento fixado na sua determinada posição.

Fonte: A autoria própria.

Com as peças devidamente acopladas o sistema tem o seu ganho em menos atrito, pois as buchas dentro da peça de alumínio do meio são com diâmetros internos maiores que o diâmetro dos guias, sendo assim deixando certo atrito diretamente no fuso trapezoidal.

3.5 MOVIMENTAÇÃO E SUPORTE

Com relação à sua movimentação, este sistema conta com 4 motores DC na horizontal, fornecendo para o carrinho cerca de 40 kgf.cm. No suporte desses motores foram utilizadas braçadeiras, parafusos e porcas, essas braçadeiras foram fixadas no chassi onde foi feito um esquema de madeira com furos laterais onde foi acoplado um HUB SEXTAVADO com 40 mm de comprimento para que os eixos ultrapassassem o MDF para fixação das rodas.

Os motores foram isolados com uma caixa de MDF para que não ocorra nenhum risco de conflito entre os fios do sistema elétrico. O carrinho foi desenhado para que os componentes eletrônicos estejam seguros.

É importante ressaltar que neste sistema a velocidade dos motores é reduzida e nunca chegará ao seu RPM total, que é de 83 RPM por conta do peso final do projeto e também pelas rodas extremamente aderentes fazendo com que todo peso do carrinho seja absorvido.

3.6 SISTEMA DE CARGA

O motor do sistema mecânico de carga foi fixado na vertical onde também foram usadas braçadeiras para fixação, o sistema conta com um acoplador de eixos 6x8 mm para ser feito a junção do fuso trapezoidal e o eixo do motor.



Figura 17: Motor de elevação da carga

Fonte: Autoria própria

4. EXPLICAÇÃO DA PROGRAMAÇÃO

A programação da empilhadeira robótica é simples o seu funcionamento, os pontos chaves estão tanto no bluetooth quanto no módulo relé. Na primeira linha da programação mandamos incluir a biblioteca `SoftwareSerial.h` onde, esta biblioteca é necessária para utilizar o RX e TX sem ser 0 e 1 dos pinos digitais do Arduino, ou seja, com isso utilizamos outros pinos digitais para a comunicação serial, logo em seguida definimos esses pinos utilizados com a função `SoftwareSerial bluetooth (10, 11)`.

Depois disso declaramos e demos um nome para as variáveis onde os pinos IN do módulo relé estão conectados.

```
int vez;
const int N1=2; // motores da direita
const int N2=3; // motores da direita
const int N3=4; // motores da esquerda
const int N4=5; // motores da esquerda
const int N5=6; // motores da empilhadeira
const int N6=7; // motores da empilhadeira
const int ledpin=13;
const int ledpin1=12;
int incomingByte;
```

Ou seja, o N1 está conectado no pino 2 da porta digital do Arduino e assim por diante demos um nome e declaramos uma variável para realizar a leitura dos dados.

```
void setup() {
  bluetooth.begin(9600);
  pinMode(N1, OUTPUT);
  pinMode(N2, OUTPUT);
  pinMode(N3, OUTPUT);
```

```
pinMode(N4, OUTPUT);
pinMode(N5, OUTPUT);
pinMode(N6, OUTPUT);
pinMode(ledpin, OUTPUT);
```

No void setup, declaramos a taxa de transmissão de dados o baud rate que colocamos 9600 em um segundo, e depois declaramos que o N1 ao N6 são pinos de saída o que os leva a um estado de baixa impedância, de modo que eles possam fornecer corrente para outros circuitos.

```
void loop ()
  if (bluetooth.available() > 0)
    incomingByte = bluetooth.read();
    if (incomingByte == 'D')
      digitalWrite(N1, HIGH);
      digitalWrite(N2, LOW);
      digitalWrite(N3, HIGH);
      digitalWrite(N4, LOW);
      digitalWrite(ledpin, HIGH);
      bluetooth.println("Ré");
```

No void loop nós temos a lógica da programação começando com a função if(bluetooth.available() > 0) onde ele irá ler se tem dados entrando ou não.

Logo em seguida temos:

```
incomingByte = bluetooth.read()
```

No começo do programa declaramos o incomingByte e nesta função estamos dizendo que ele é a mesma coisa que o bluetooth.read que é uma função para ler um Byte.

```
if (incomingByte == 'D')
  digitalWrite(N1, HIGH);
```

```
digitalWrite(N2, LOW);
digitalWrite(N3, HIGH);
digitalWrite(N4, LOW);
digitalWrite(ledpin, HIGH);
```

Se o Byte for igual a D eu irei executar a função digitalWrite, esta função irá ligar os relés ou seja, se ele ler D será acionado o N1 e N3 fazendo assim energizar meus módulos relés e com isso mover os motores para trás.

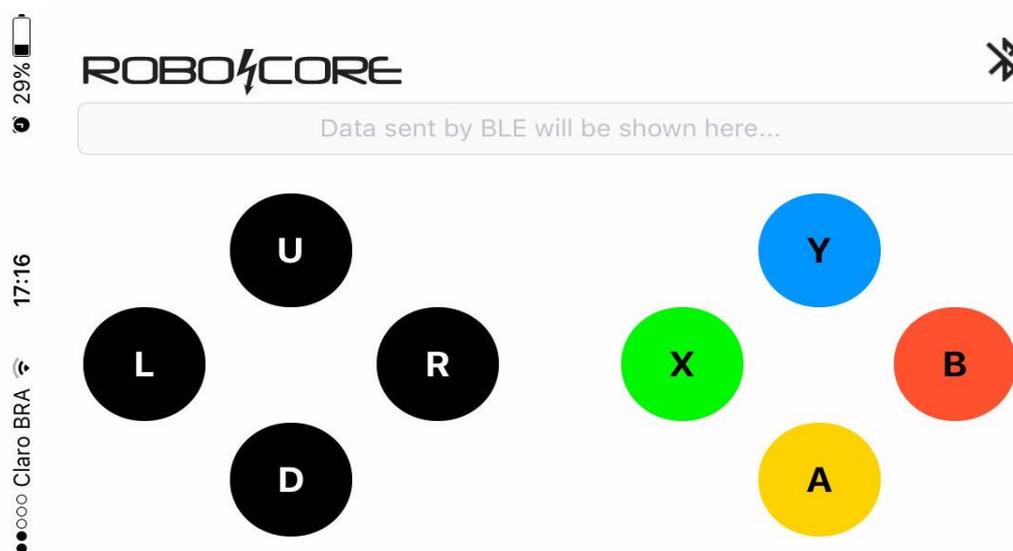


Figura 18: Interface BlackBLE

Fonte: BlackBle

No aplicativo desenvolvido para iPhone temos esta configuração, onde com a nossa programação quando selecionamos no D enviamos um byte onde com a função `incomingByte = bluetooth.read` irá ler o valor que foi apertado.

```
bluetooth.println("Ré").
```

Logo em seguida temos esta função que quando pressionado o D mostra na barra da tela do iphone a mensagem Ré.

Quando ele ler U irão ligar o N2 e N4 fazendo assim meus motores irem para frente.

```

if (incomingByte == 'U')
    digitalWrite(N1, LOW);
    digitalWrite(N2, HIGH);
    digitalWrite(N3, LOW);
    digitalWrite(N4, HIGH);
    digitalWrite(ledpin, LOW);
    bluetooth.println("Frente");

```

Aqui iremos ligar o N2 e N3 fazendo assim a empilhadeira girar em torno do seu próprio eixo e girar pra direita.

```

if (incomingByte == 'R')
    digitalWrite(N2, LOW);
    digitalWrite(N1, HIGH);
    digitalWrite(N4, HIGH);
    digitalWrite(N3, LOW);
    digitalWrite(ledpin, LOW);
    bluetooth.println("Direita");

```

Se ele ler o L iremos ligar o N1 e N4 e conseqüentemente irar girar em torno do seu próprio eixo, girar para a esquerda.

```

if (incomingByte == 'L')
    digitalWrite(N2, HIGH);
    digitalWrite(N1, LOW);
    digitalWrite(N4, LOW);
    digitalWrite(N3, HIGH);
    digitalWrite(ledpin, LOW);
    bluetooth.println("Esquerda");

```

Quando ler o B irá parar todos os motores.

```

if (incomingByte == 'B')

```

```

digitalWrite(N1, HIGH);
    digitalWrite(N2, HIGH);
    digitalWrite(N3, HIGH);
    digitalWrite(N4, HIGH);
    digitalWrite(ledpin, LOW);
    bluetooth.println("Parar tudo: ON");

```

Quando o Y selecionado o N5 é ativado e o motor do fuso sobe.

```

if (incomingByte == 'Y')
    digitalWrite(N5, HIGH);
    digitalWrite(N6, LOW);
tone(buzzer,2300);
bluetooth.println("Empilhadeira sobe");

```

Quando o A selecionado o N6 é ativado e o motor desce.

```

if (incomingByte == 'A')
    digitalWrite(N5, LOW);
    digitalWrite(N6, HIGH);
    tone(buzzer,2300);
bluetooth.println("Empilhadeira desce");

```

Quando o X acionado todos os motores irão parar.

```

if (incomingByte == 'X')
    digitalWrite(N5, HIGH);
    digitalWrite(N6, HIGH);
    noTone(buzzer);
    bluetooth.println("Para tudo ");

```

Se for F ligo meu led da frente.

```

if (incomingByte == 'I') {

```

```
digitalWrite(ledpin1, LOW );
```

Se for I desligo meus led da frente.

```
if (incomingByte == 'H') {
```

```
if (vez != 1) {
```

```
vez = 1;
```

```
tone(buzzer,2300);
```

```
delay(500);
```

```
noTone(buzzer);
```

```
delay(500);
```

```
tone(buzzer,2300);
```

```
delay(500);
```

```
noTone(buzzer);
```

```
delay(500);
```

```
tone(buzzer,2300);
```

```
delay(500);
```

```
noTone(buzzer);
```

```
delay(500);
```

```
tone(buzzer,2300);
```

```
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);
```

```
digitalWrite(ledpin1, LOW);  
digitalWrite(N1, HIGH);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N5, HIGH);  
digitalWrite(N6, HIGH);
```

```
delay(2000);
```

Quando a função H executa a forma autônoma será executada apenas uma vez quando clicada na letra H e reiniciar se eu voltar a clicar, a programação funciona por delay, onde calculamos o tempo e os lugares determinados para a empilhadeira pegar algum material ou um objeto. Para sua operação temos um som que avisa que a empilhadeira irá começar a trabalhar.

5. DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento do projeto foi feito um estudo de empilhadeiras autônomas ou controladas por operadores, com as pesquisas finalizadas decidimos os hardwares e sua estrutura.

1° - Foram feitos cortes na madeira (MDF) de acordo com o desenho de uma empilhadeira convencional.

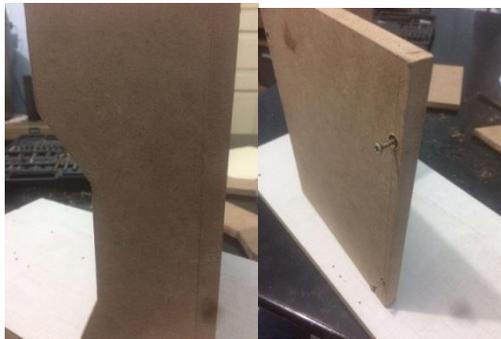


Figura 19: Madeira MDF

Fonte: Autoria própria.

2° - Após as peças prontas e posicionadas, obtivemos a base do projeto.



Figura 20: Base do Projeto

Fonte: Autoria própria.

3° - Após a base, hastes e teto prontos foram efetuados furos nas laterais para posicionamento dos motores e das rodas.



Figura 21: Furos laterais

Fonte: Autorial própria.

4° - Após os furos realizados, posicionamos os motores;



Figura 22: Posicionamento dos motores

Fonte: Autorial própria.

5° - Iniciamos os testes com os motores individualmente;



Figura 23: Testes com os motores

Fonte: Autoria própria.

6° - Posicionamento das rodas nos motores;



Figura 24: Posicionamento das Rodas

Fonte: Autoria própria.

7° - Polarização dos motores;



Figura 25: Polarização

Fonte: Autoria própria.

8° - Carroceria, motores e rodas todos posicionados em seus respectivos locais;

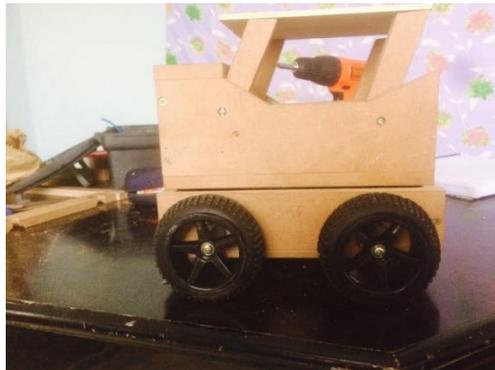


Figura 26: Base da empilhadeira

Fonte: Autoria própria.

9º - Iniciar Fixação de placa controladora na base da carroceria;



Figura 27: Placa controladora
Fonte: Autoria própria.

10º- Instalação dos Relés.



Figura 28: Instalação dos Relés
Fonte: Autoria própria.

11º- Fixação do fuso trapezoidal em sua base de apoio;

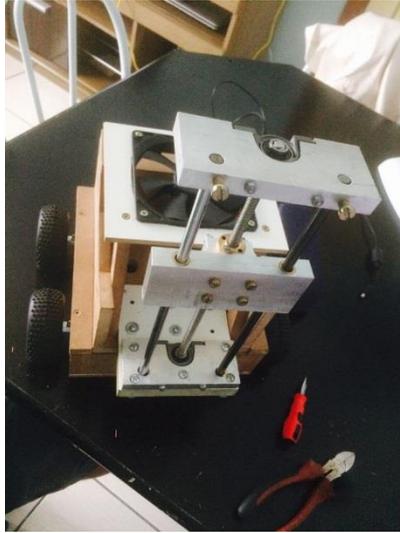


Figura 29: Fuso Trapezoidal

Fonte: Autoria própria.

12º- Confeção das pás de sustentação de carga;



Figura 30: Pás da Empilhadeira

Fonte: Autoria própria.

13º- Fixação das pás de sustentação de carga;

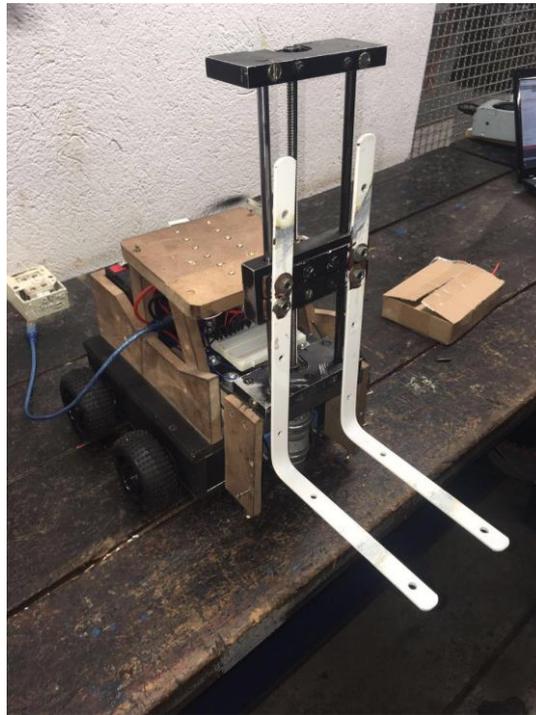


Figura 31: Fixação das pás de sustentação

Fonte: Autoria própria.

14º- Fixação da bateria;



Figura 32: Bateria

Fonte: Autoria própria

6. ESQUEMA ELÉTRICO

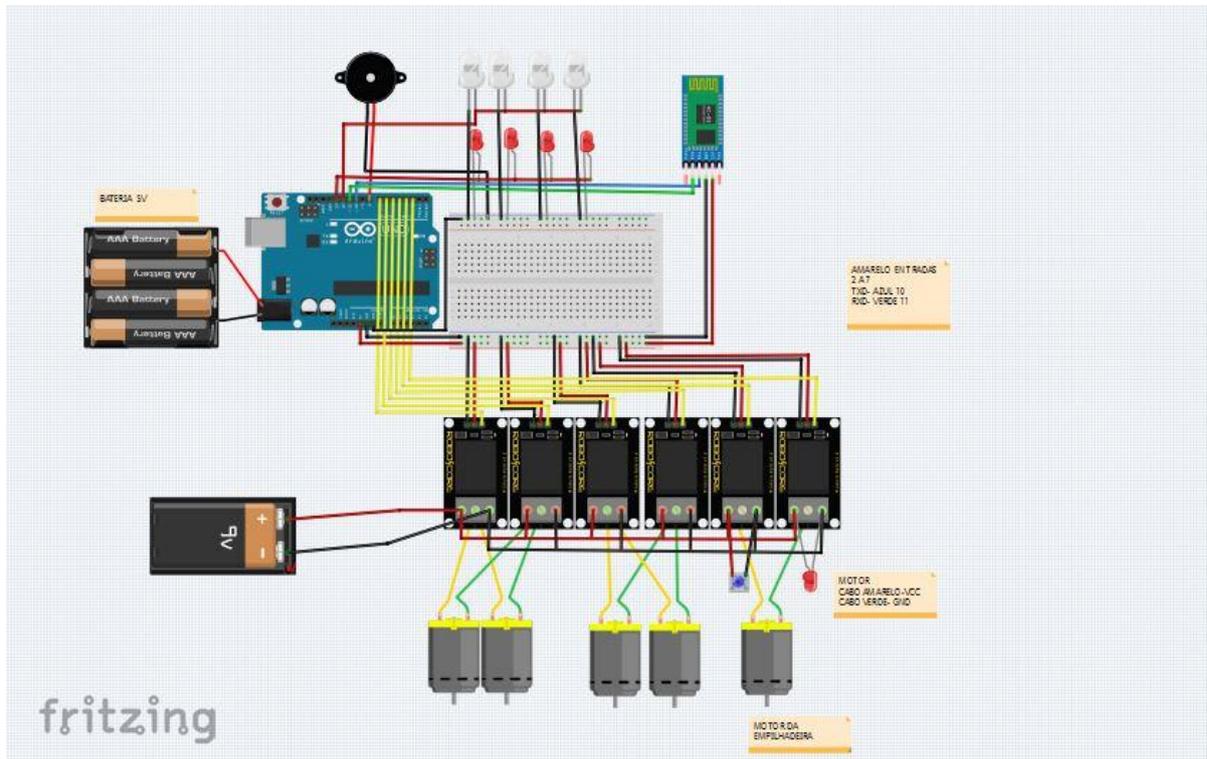


Figura 33: Esquema elétrico

Fonte: Fritzing.

Dentro do esquema elétrico temos os seguintes componentes:

- 6 módulos relé;
- 5 motores Akiyama 12V;
- Arduino Uno;
- Módulo Bluetooth Low Energy – HC-08;
- Protoboard;
- Fonte de alimentação 5V, 9V.
- Cabos amarelos – Pino IN Módulos Relé;
- 5 V – Cabo Vermelho;
- GND – Cabo Preto;
- RX – Azul;
- TX – Verde.

Neste esquema elétrico os meus pinos IN estão sendo ligados as saídas digitais do Arduino na 2,3,4,5,6 e 7, já meu VCC e GND dos módulos dos relés estão sendo todos conectados na trilha da protoboard do positivo e negativo.

No Arduino está sendo puxado do pino de 5V e do pino de GND dois cabos e sendo ligados na minha Protoboard onde está alimentando tanto o módulo relé quando o bluetooth.

No NA do módulo bluetooth temos a interligação dos contatos igualmente com o NF onde é conectada uma bateria que no desenho foi representada 9V mas no projeto foi utilizada uma de 14,8V.

Já no bluetooth sai o cabo vermelho e preto onde é ligado na protoboard e mais dois cabos, o RX e TX que são ligados na saída digital do Arduino 10 e 11.

Os 5 motores são conectados no C do módulo relé onde os motores da direita são juntados com os fios, como visto no esquema assim como os motores da esquerda.

Quando ligamos o Arduino, os pinos de 5V e GND passam a alimentar a protoboard onde nela estão conectados o 5V e o GND do módulo relé, com isso temos o acionamento dos módulos relés e bluetooth, já os cabos IN servem para a comunicação entre o Arduino e módulo relé, e os cabos RX E TX do bluetooth servem também para a comunicação entre o Arduino.

Com a programação podemos realizar o acionamento dos relés, ou seja, por meio das conexões do IN e do RX e TX conseguimos executar o comando para acionar os meus relés invertendo a rotação dos motores.

7. CONCLUSÃO

Podemos concluir com o nosso projeto que a produção, a facilidade de transportar coisas ou até mesmo inovações que possam vir a fazer uma grande diferença em uma indústria se resultara em uma inovação, pois, ao colocar uma empilhadeira onde ela irá carregar determinado setor sozinho, por exemplo, irá reduzir custos do mesmo e diminuirá o tempo que um operador demoraria sozinho. Além de usar o carrinho como uma Empilhadeira é possível também alterar o seu objetivo, podemos colocar uma caixa com ferramentas em cima de seu suporte afim de agilizar a produção em setores, onde quando faltar uma peça/ferramenta por exemplo, o carrinho estará sempre à disposição do operário. O objetivo da nossa Empilhadeira é fazer com que todo o procedimento padrão de uma indústria, que seria o abastecimento de carga ou de matéria prima, seja automatizado e rápido.

8. CRONOGRAMA

FEVEREIRO	<ul style="list-style-type: none"> - ESCOLHA DO MATERIAL PARA ESTRUTURA. - ESCOLHA DE MOTORES/ COMPONENTES. - MONTAGEM DO ESCOPO.
MARÇO	<ul style="list-style-type: none"> - COMPRA DE MOTORES, RODAS E EQUIPAMENTOS ELETRICOS. - COMPRA DO FUSO. - TESTE DE MOTORES E MONTAGEM NO ESCOPO.
ABRIL	<ul style="list-style-type: none"> - FINALIZAR MONTAGEM. - INICIO DE TESTE COM PROGRAMAÇÃO. - COMPRA DE BATERIA.
MAIO	<ul style="list-style-type: none"> - INTENSIFICAÇÃO DOS TESTES COM GARGA. - MONTAGEM DA MONOGRAFIA. - REVISÃO ORTOGRÁFICA DA MONOGRAFIA. - ESTUDO PARA APRESENTAÇÃO.
JUNHO	<ul style="list-style-type: none"> - TESTES. - FINALIZAÇÃO DE MONOGRAFIA. - AJUSTES FINAIS.

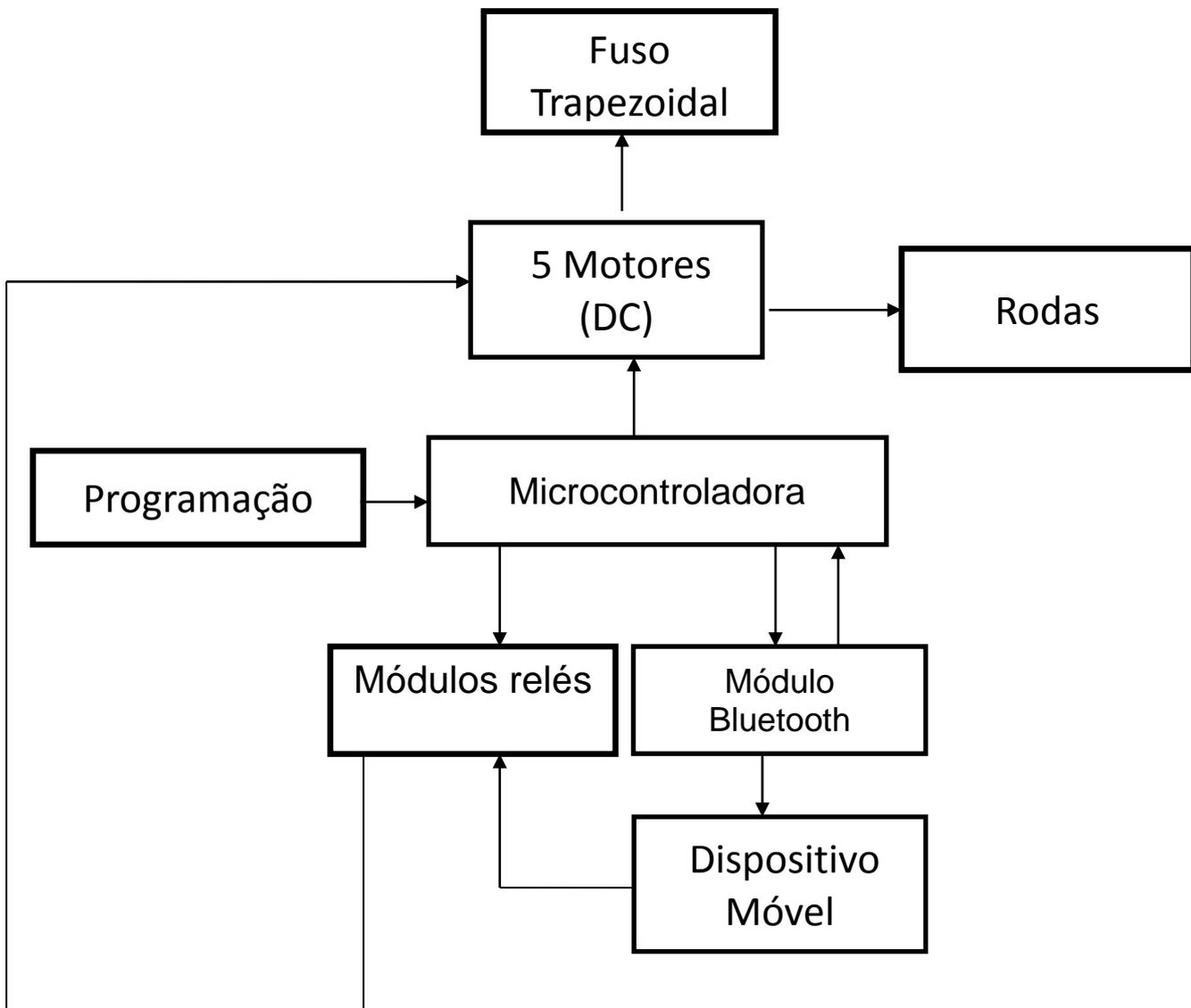
9. ESPECIFICAÇÕES

Componentes	Especificações
Arduino	<ul style="list-style-type: none"> • Alimentação 5V- 12V. • Velocidade de Clock – 16 MHz. • 14 entradas e saídas digitais. Sendo 6 analógicas.
Bateria	<ul style="list-style-type: none"> • Dimensões • Peso: 2 kg • Comprimento: 16 Cm • Largura: 7 Cm • Altura com Terminal: 10 Cm • Tensão Nominal: 12V • Capacidade Nominal: 7,0Ah
Módulo Relé	<ul style="list-style-type: none"> • Tipo: Digital. • Placa com apenas um relé e todo seu circuito de driver. • Sinal de controle: Nível TTL. • Bobina: 5VDC 75mA. • Carga nominal do relé: 12A 125VAC, 7A 250VAC. • Carga nominal do módulo: 10A. • Tempo de acionamento de contato: 10ms.
Motor Akiyama(DC)	<ul style="list-style-type: none"> • Torque de 11,1Kgf.cm. • Corrente 430mA. • Picos de corrente de 1,6A. • 83rpm.
Fuso Trapezoidal	<ul style="list-style-type: none"> • Dimensão 300mm x 100mm – TR8-8m.
Módulo Bluetooth- HC-08	<ul style="list-style-type: none"> • Baud rate padrão: 9600. • Tensão de alimentação: 5 V. • Tensão de operação: 3,3 V. • É compatível com dispositivos Apple, ou seja, sistema operacional iOS.

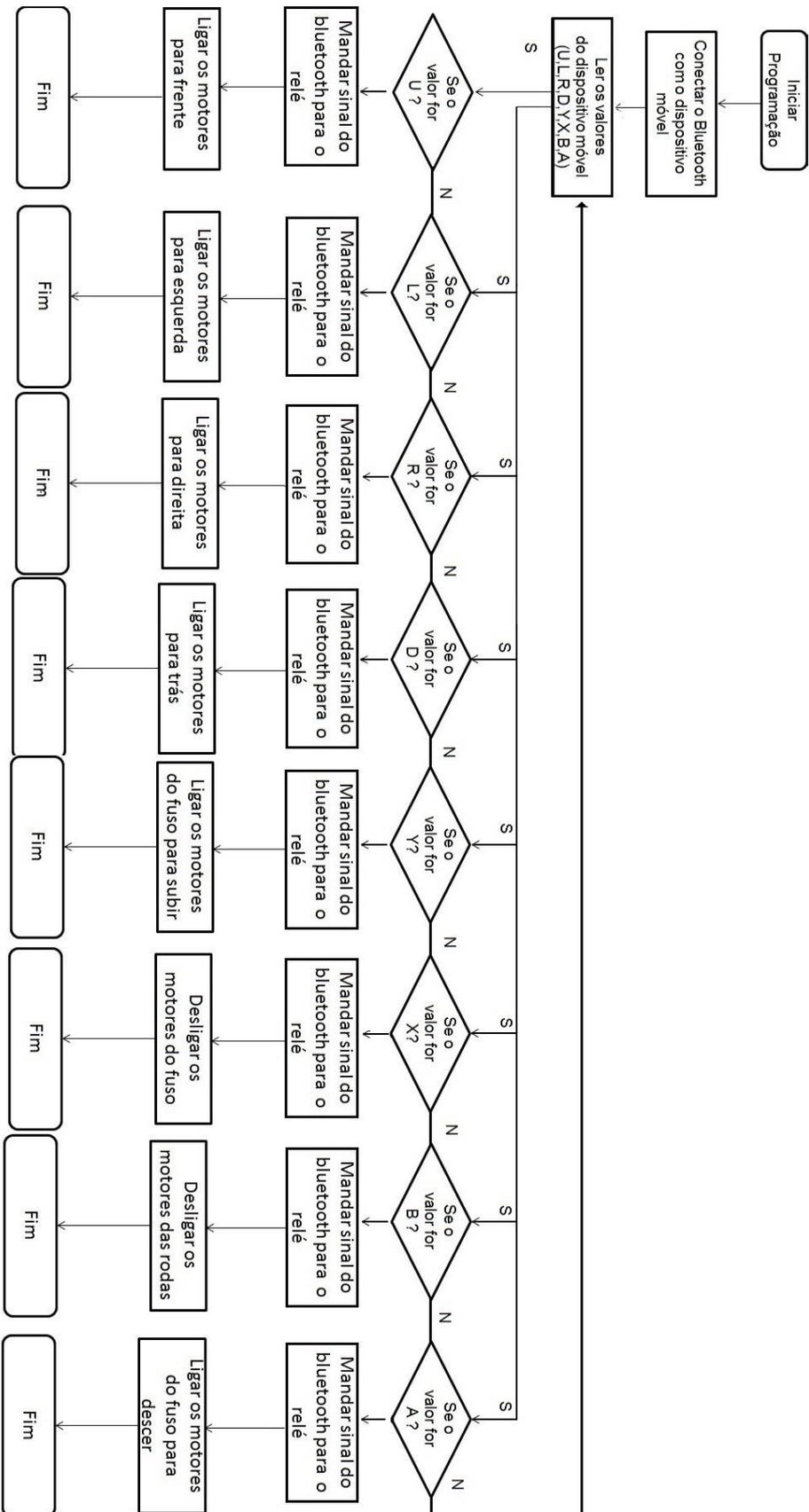
10. TABELA DE CUSTOS

Item	Quantidade	Unitário	Total
Mecânico			
Madeira MDF	1	R\$120,00	R\$120,00
Sistema completo do mancal e fuso trapezoidal	1	R\$146,00	R\$146,00
Parafusos	1	R\$22,60	R\$22,60
Braçadeira	10	R\$2,30	R\$23,00
Rodas	4	R\$42,00	R\$168,00
Hub Sextavado	4	R\$10,00	R\$40,00
Tinta para pintura	2	R\$16,00	R\$32,00
Mão de obra			R\$700,00
Eleto-Eletrônico			
Módulo Bluetooth	1	R\$49,00	R\$49,00
Bateria Unipower 12V	1	R\$61,00	R\$61,00
Arduino Uno	1	R\$49,90	R\$49,90
Módulo Relé	6	R\$12,90	R\$77,40
Motores 12V	5	R\$59,00	R\$295,00
Fita de led	1	R\$24,90	R\$24,90
Led de Alto Brilho	12	R\$0,90	R\$10,80
Buzzer	1	R\$2,00	R\$2,00
Total			R\$1.821,60

11. DIAGRAMA EM BLOCOS



12. FLUXOGRAMA DE PROGRAMAÇÃO



13. PROGRAMAÇÃO DO ARDUINO

```
#include "SoftwareSerial.h"
SoftwareSerial bluetooth(10, 11);

const int N1=2; // motores da direita
const int N2=3; // motores da direita
const int N3=4; // motores da esquerda
const int N4=5; // motores da esquerda
const int N5=6; // motores da empilhadeira
const int N6=7; // motores da empilhadeira
int incomingByte;

void setup() {
  bluetooth.begin(9600);
  pinMode(N1, OUTPUT);
  pinMode(N2, OUTPUT);
  pinMode(N3, OUTPUT);
  pinMode(N4, OUTPUT);
  pinMode(N5, OUTPUT);
  pinMode(N6, OUTPUT);
}

void loop() {
  if (bluetooth.available() > 0) {
    incomingByte = bluetooth.read();
    if (incomingByte == 'D') {
      digitalWrite(N1, HIGH);
      digitalWrite(N2, LOW);
    }
  }
}

#include "SoftwareSerial.h"
SoftwareSerial bluetooth(10, 11); //RX, TX
```

```
int vez;
const int N1=2; // motores da direita
const int N2=3; // motores da direita
const int N3=4; // motores da esquerda
const int N4=5; // motores da esquerda
const int N5=6; // motores da empilhadeira
const int N6=7; // motores da empilhadeira
const int ledpin=13;
const int ledpin1=12;
const int ledpin2=9;
const int ledpin3=A0;
const int buzzer = 8;
int incomingByte;
```

```
void setup() {
  bluetooth.begin(9600);
  pinMode(N1, OUTPUT);
  pinMode(N2, OUTPUT);
  pinMode(N3, OUTPUT);
  pinMode(N4, OUTPUT);
  pinMode(N5, OUTPUT);
  pinMode(N6, OUTPUT);
  pinMode(ledpin, OUTPUT);
  pinMode(ledpin1, OUTPUT);
  pinMode(ledpin2, OUTPUT);
  pinMode(ledpin3, INPUT);
  pinMode(buzzer,OUTPUT);
}
```

```
void loop() {
  if (bluetooth.available() > 0) {
    incomingByte = bluetooth.read();
  }
}
```

```
if (incomingByte == 'D') {
  digitalWrite(N1, HIGH);
  digitalWrite(N2, LOW);
  digitalWrite(N3, HIGH);
  digitalWrite(N4, LOW);
  digitalWrite(ledpin, HIGH);

  bluetooth.println("Ré");
}
if (incomingByte == 'U') {
  digitalWrite(N1, LOW);
  digitalWrite(N2, HIGH);
  digitalWrite(N3, LOW);
  digitalWrite(N4, HIGH);
  digitalWrite(ledpin, LOW);
  bluetooth.println("Frente");
}
if (incomingByte == 'R') {
  digitalWrite(N2, LOW);
  digitalWrite(N1, HIGH);
  digitalWrite(N4, HIGH);
  digitalWrite(N3, LOW);
  digitalWrite(ledpin, LOW);
  bluetooth.println("Direita");
}
if (incomingByte == 'L') {
  digitalWrite(N2, HIGH);
  digitalWrite(N1, LOW);
  digitalWrite(N4, LOW);
  digitalWrite(N3, HIGH);
  digitalWrite(ledpin, LOW);
  bluetooth.println("Esquerda");
}
if (incomingByte == 'B') {
```

```
digitalWrite(N1, HIGH);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(ledpin, LOW);
```

```
bluetooth.println("Parar tudo: ON");  
}  
  if (incomingByte == 'Y') {  
digitalWrite(N5, HIGH);  
digitalWrite(N6, LOW);  
digitalWrite(ledpin2, HIGH);  
digitalWrite(ledpin3, LOW);  
tone(buzzer,2300);
```

```
bluetooth.println("Empilhadeira sobE");  
}  
  if (incomingByte == 'A') {  
digitalWrite(N5, LOW);  
digitalWrite(N6, HIGH);  
tone(buzzer,2300);  
digitalWrite(ledpin3, HIGH);  
digitalWrite(ledpin2, LOW);  
bluetooth.println("Empilhadeira desce");  
}  
  if (incomingByte == 'X') {  
digitalWrite(N5, HIGH);  
digitalWrite(N6, HIGH);  
noTone(buzzer);
```

```
        digitalWrite(ledpin2, LOW);
        digitalWrite(ledpin3, LOW);
        bluetooth.println("Para tudo ");
    }
    if (incomingByte == 'F') {
        digitalWrite(ledpin1, HIGH);

        bluetooth.println("Liga Led ");
    }
    if (incomingByte == 'I') {
        digitalWrite(ledpin1, LOW );

        bluetooth.println("Liga Led ");
    }
```

```
if (incomingByte == 'H') {
```

```
if (vez != 1) {
vez = 1;
    tone(buzzer,2300);
    delay(500);
```

```
noTone(buzzer);
    delay(500);
    tone(buzzer,2300);
    delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);  
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);  
  tone(buzzer,2300);  
delay(500);
```

```
noTone(buzzer);  
delay(500);
```

```
digitalWrite(ledpin1, LOW);  
digitalWrite(N1, HIGH);
```

```
digitalWrite(N2, HIGH);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N5, HIGH);  
digitalWrite(N6, HIGH);
```

```
delay(2000);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);  
digitalWrite(ledpin1, HIGH);
```

```
delay(2000);
```

```
digitalWrite(N2, LOW);  
digitalWrite(N1, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N3, LOW);
```

```
delay(800);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

```
delay(2000);
```

```
digitalWrite(N1, HIGH);  
digitalWrite(N2, HIGH);
```

```
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N5, LOW);  
digitalWrite(N6, HIGH);
```

```
delay(8000);
```

```
digitalWrite(N5, HIGH);  
digitalWrite(N6, HIGH);  
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

```
delay(800);
```

```
digitalWrite(N1, HIGH);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);
```

```
delay(1000);
```

```
digitalWrite(N6, LOW);  
digitalWrite(N5, HIGH);
```

```
delay(8000);
```

```
digitalWrite(N1, HIGH);  
digitalWrite(N2, LOW);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, LOW);
```

```
digitalWrite(ledpin, HIGH);  
digitalWrite(N5, HIGH);  
digitalWrite(N6, HIGH);
```

```
delay(2000);
```

```
digitalWrite(N2, LOW);  
digitalWrite(N1, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(ledpin, LOW);
```

```
delay(800);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

```
delay(1000);
```

```
digitalWrite(N2, HIGH);  
    digitalWrite(N1,LOW);  
    digitalWrite(N4, LOW);  
    digitalWrite(N3, HIGH);
```

```
delay(850);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

delay(2000);

digitalWrite(N1,HIGH);

digitalWrite(N2, HIGH);

digitalWrite(N3, HIGH);

digitalWrite(N4, HIGH);

digitalWrite(N5, LOW);

digitalWrite(N6, HIGH);

delay(8000);

digitalWrite(N5, HIGH);

digitalWrite(N6, HIGH);

delay(1000);

digitalWrite(N1, HIGH);

digitalWrite(N2, LOW);

digitalWrite(N3, HIGH);

digitalWrite(N4, LOW);

digitalWrite(ledpin, HIGH);

delay(2000);

digitalWrite(N1, HIGH);

digitalWrite(N2, HIGH);

digitalWrite(N3, HIGH);

digitalWrite(N4, HIGH);

digitalWrite(N6, LOW);

digitalWrite(N5, HIGH);

delay(8000);

digitalWrite(N2, HIGH);

digitalWrite(N1,LOW);

```
digitalWrite(N4, LOW);  
digitalWrite(N3, HIGH);  
digitalWrite(N6, HIGH);  
digitalWrite(N5, HIGH);
```

```
delay(850);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

```
delay(3000);
```

```
digitalWrite(N2, LOW);  
digitalWrite(N1, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N3, LOW);
```

```
delay(800);
```

```
digitalWrite(N1, LOW);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, LOW);  
digitalWrite(N4, HIGH);
```

```
delay(1500);
```

```
digitalWrite(N1, HIGH);  
digitalWrite(N2, HIGH);  
digitalWrite(N3, HIGH);  
digitalWrite(N4, HIGH);  
digitalWrite(N5, LOW);
```

```
digitalWrite(N6, HIGH);
```

```
delay(8000);
```

```
digitalWrite(N1, LOW);
```

```
digitalWrite(N2, HIGH);
```

```
digitalWrite(N3, LOW);
```

```
digitalWrite(N4, HIGH);
```

```
  digitalWrite(N5, HIGH);
```

```
digitalWrite(N6, HIGH);
```

```
delay(700);
```

```
digitalWrite(N1, HIGH);
```

```
digitalWrite(N2, HIGH);
```

```
digitalWrite(N3, HIGH);
```

```
digitalWrite(N4, HIGH);
```

```
delay(1000);
```

```
digitalWrite(N6, LOW);
```

```
digitalWrite(N5, HIGH);
```

```
delay(8000);
```

```
digitalWrite(N1, HIGH);
```

```
digitalWrite(N2, LOW);
```

```
digitalWrite(N3, HIGH);
```

```
digitalWrite(N4, LOW);
```

```
digitalWrite(N6, HIGH);
```

```
digitalWrite(N5, HIGH);
```

delay(1500);

digitalWrite(N2, LOW);
digitalWrite(N1, HIGH);
digitalWrite(N4, HIGH);
digitalWrite(N3, LOW);

delay(800);

digitalWrite(N1, LOW);
digitalWrite(N2, HIGH);
digitalWrite(N3, LOW);
digitalWrite(N4, HIGH);

delay(2500);

digitalWrite(N2, HIGH);
digitalWrite(N1, LOW);
digitalWrite(N4, LOW);
digitalWrite(N3, HIGH);

delay(850);

digitalWrite(N1, LOW);
digitalWrite(N2, HIGH);
digitalWrite(N3, LOW);
digitalWrite(N4, HIGH);

delay(1000);

digitalWrite(N1, HIGH);
digitalWrite(N2, HIGH);
digitalWrite(N3, HIGH);
digitalWrite(N4, HIGH);

```
    delay(5000);

    digitalWrite(ledpin1, LOW);
    digitalWrite(N1, HIGH);
    digitalWrite(N2, HIGH);
    digitalWrite(N3, HIGH);
    digitalWrite(N4, HIGH);
    digitalWrite(N5, HIGH);
    digitalWrite(N6, HIGH);

}

}

}
}
```

14. REFERÊNCIAS

Robocore, “Arduino”. Disponível em

<<https://www.robocore.net/loja/produtos/julieta.html>> Acesso em 28 de Outubro de 2016.

Canal Tech, “O que é arduino?”. Disponível em *<<https://canaltech.com.br/o-que-e/hardware/o-que-e-arduino/>>* Acesso em 28 de Outubro de 2016.

Techtudo, “Bluetooth: O que é e como funciona”. Disponível em

<<http://www.techtudo.com.br/artigos/noticia/2012/01/bluetooth-o-que-e-e-como-funciona.html>> Acesso em 24 de Novembro de 2016.

Robocore, “Módulo Bluetooth Low Energy - HC-08”. Disponível em

<<https://www.robocore.net/loja/produtos/modulo-bluetooth-low-energy-hc-08.html>> Acesso em 03 de Janeiro de 2017.

Vídeo explicativo sobre a ponte H do módulo relé. Disponível em

<<https://www.youtube.com/watch?v=2aH1t2bh6yM>> Acesso em 15 de Março de 2017.

Tecnomelque, “Módulo relé de 4 canais com ponte H dupla”. Disponível

em *<<http://tecnomelque.blogspot.com.br/2016/08/modulo-rele-de-4-canais-como-ponte-h.html>>* Acesso em 15 de Março de 2017.

Ponte H a partir de módulo relé. Disponível em

<<https://www.youtube.com/watch?v=i0ARkzEkc1k>> Acesso em 15 de Março de 2017.

Como funciona o Relé! Vídeo aula. Disponível em

<https://www.youtube.com/watch?v=Jd9_3caX1U> Acesso em 18 de Março de 2017.

Robocore, “Módulo relé”. Disponível em

<<https://www.robocore.net/loja/produtos/modulo-rele.html>> Acesso em 18 de Março de 2017.

Alunos online, “Baterias de chumbo usadas nos automóveis”. Disponível em <<http://alunosonline.uol.com.br/quimica/baterias-chumbo-usadas-nos-automoveis.html>> Acesso em 14 de Maio de 2017.