



ETEC JORGE STREET

**TRABALHO DE CONCLUSÃO DO CURSO TÉCNICO EM AUTOMAÇÃO
INDUSTRIAL**

Braço Robótico

**Celso Francisco dos Santos
Francisco Isnaldo Moura Macedo
Gerson Ferreira Lima
Jamil Celestino Costa
Leandro dos Santos Rodrigues
Luiz Guilherme Gomes
Rodrigo Alves
Wilson Roberto Cardoso**

**Professor (es). Orientador (es):
Renê Graminhani
Renato Machado**

**São Caetano do Sul / SP
2018**

Braço Robótico

Trabalho de Conclusão de Curso
apresentado como pré-requisito para
obtenção do Diploma de Técnico em
Automação Industrial.

**São Caetano do Sul / SP
2018**

RESUMO

Neste presente trabalho queremos demonstrar os conceitos e o funcionamento de um braço robótico controlado via arduino em um processo com esteira.

Palavras-chave: (Arduino, Braço Robótico, Esteira)

LISTA DE FIGURAS

Figura 1 – Junta e elos	13
Figura 2 – Junta P	13
Figura 3 – Junta R.....	14
Figura 4 – Junta Esférica	14
Figura 5 – Pulsos	16
Figura 6 – Servo-motor	17
Figura 7 – Arduino.....	18
Figura 8 – Módulos.....	19
Figura 9 – Shields Ethernet.....	19
Figura 10 – Tipos de Arduino	20
Figura 11 – IDE de Programação.....	20
Figura 12 – Arduino Uno	21
Figura 13 – PWM	22
Figura 14 – Esteira Transporte por Correia.....	25
Figura 15 – Tambores	25
Figura 16 – Roletes	26
Figura 17 – Processo de Detecção	27
Figura 18 – Retificação	28
Figura 19 – Regulador de Tensão.....	28
Figura 20 – Diodo Zener	29
Figura 21 – CLP CLIC 2 WEG.....	30
Figura 22 – Arduino Mega.....	30
Figura 23 – Parte Superior da Base	34
Figura 24 – Parafusos e Parte Inferior e Superior da Base.....	34
Figura 25 – Lateral da Base	35
Figura 26 – Lateral da Base	35
Figura 27 – Laterais da Base mais Hastes do Braço	36
Figura 28 – Laterais da Base mais Hastes do Braço	36
Figura 29 – Fixação do Cotovelo.....	37
Figura 30 – Fixação do Cotovelo mais Separador das Hastes do Cotovelo	37

Figura 31 – Fixação dos Parafusos	38
Figura 32 – Fixação do Servo Motor da Garra	38
Figura 33 – Fixação das Partes que compoe a garra.....	39
Figura 34 – Fixação dos Parafusos da Garra.....	39
Figura 35 – Junção da Garra com o Cotovelo.....	40
Figura 36 – Junção da Garra com o Cotovelo.....	40
Figura 37 – Circuito de Controle para Servo Motor (PWM)	41
Figura 38 – Arduino Mega.....	42
Figura 39 – Esteira.....	42
Figura 40 – Braço.....	43
Figura 41 – Tambor modificado com lixa	43
Figura 42 – Roletes para levantar esteira	44
Figura 43 – Esteira Antes da Elevação da Correia.....	44
Figura 44 – Esteira Após Elevação	44
Figura 45 – Interface de Potência	45
Figura 46 – Circuito da Interface de Potncia	45
Figura 47 – Sensor de Barreira	46
Figura 48 – Etiqueta com Informações da Fonte.....	46
Figura 49 – Vista Traseira da Fonte	47
Figura 50 – Aplicativo para Posicionamento do Braço	47
Figura 51– Aplicativo para Posicionamento do Braço	48
Figura 52 – Scripit do Aplicativo.....	48
Figura 53 – Ligação dos Servos Motores com Arduino.....	49
Figura 54 - Ligação do Arduino com o Motor da Esteira e Bluetooth	49
Figura 55 – Projeto Final	50

Sumário

Introdução	8
1 – Fundamentação Teórica.....	9
1.1 – Definições do Braço Robótico	12
1.2 – Servo Motor.....	15
1.3 – Arduino	18
1.3.1 – Módulos e Shields	18
1.3.2 – Modelos de Arduino.....	20
1.3.3 – Programação do Arduino.....	20
1.3.4 – Hardware do Arduino.....	21
1.4 – PWM.....	22
1.5 – Bluetooth	23
1.5.1 – Como Funciona o Bluetooth	23
1.6 – Esteiras Rolante	24
1.6.1 – Constituição da Esteira.....	24
1.7 – Sensores	26
1.8 – Fonte de Alimentação DC	27
2 – Planejamento	30
2.1 – Pesquisa de Componentes / Tecnologias	30
2.1.2 – Fluxograma do Processo.....	31
2.1.3 – Tabela de Custo e Cronograma	32
2.1.4 – Desenhos do Braço	34
3 – Testes.....	41
4 – Resultados Obtidos	50
Conclusão	51
Referencias	52
Apendice A.....	53

Introdução

Será utilizado um braço robótico didático para simular junto com uma pequena esteira um processo industrial. O braço robótico será controlado por uma placa de prototipagem (Arduino) e irá simular um processo industrial junto com uma esteira.

Tema e delimitação.

Pretende-se projetar um braço robótico e uma esteira, utilizada em grande escala nas indústrias em geral.

Objetivo – geral

Usar o arduino como meio de controle de um protótipo de uma esteira e um braço robótico, de melhor custo benefício.

Objetivo – específico

Demonstrar o processo de uma esteira que transportará uma peça e ao passar pelo sensor de obstáculo, acionará o braço robótico, o mesmo entenderá que a peça está pronta para ser recolhida e empilhada no local designado pelo operador ou programa.

Justificativa

O projeto desenvolvido é de fundamental importância para indústrias, cuja atividade comercial exija velocidade, precisão e facilidade de uso na maioria das aplicações.

Metodologia

Pesquisou-se sobre qual melhor projeto a ser desenvolvido e concluiu-se que o braço robótico seria o de melhor custo benefício para a equipe, bem como pesquisas sobre os componentes e qual a melhor forma de acionamento do protótipo.

1 – Fundamentação Teórica

1.1 Robôs

Robôs são uma invenção do século XX, mas sua idealização veio bem antes. Há relatos históricos bem mais antigos da construção de autômatos como um cachorro mecânico de brinquedo, por exemplo, encontrado no Egito e datado de 2.000 AC. Mesmo na *Íliada*, Homero descreve as assistentes do Deus dos ferreiros, Hephaestus, como "virgens de ouro, vivendo como mulheres, com inteligência, voz, a energia de uma serva, habilidades de imortais e montadas em tripés com rodas que lhes permitiam movimentar... uma maravilha para os olhos".

O grande catalisador do desenvolvimento de autômatos foi a Revolução Industrial. Com ela, foram desenvolvidos e aperfeiçoados dispositivos automáticos capazes de manipular e executar peças, permitindo a automatização da produção. Produzir em escala trouxe destaque à manipulação de objetos e acelerou o desenvolvimento de manipuladores, os que mais ganharam investimentos foram as estruturas que eram montadas com segmentos e junções, colocados de forma linear, que lhe davam uma aparência de braços e pernas.

Apesar da força da Revolução Industrial, a palavra Robot foi primeiramente utilizada na ficção. Ela apareceu em 1920, na peça "RUR - Rossum Universal Robot", do dramaturgo tcheco, Karel Čapek. O termo originou-se da palavra tcheca "Robota", que significa trabalho árduo, duro, e é sinônimo de trabalho escravo. Na peça, o cientista Rossum cria humanos mecanizados, os Robot, que exerciam funções repetitivas e pesadas.

Neste mesmo período, em 1924, surgiu o primeiro modelo de robô mecânico. Roy J. Wensley, engenheiro elétrico da Westinghouse, desenvolveu uma unidade de controle supervisionada. O dispositivo podia, utilizando o sistema de telefonia, ligar e desligar ou regular remotamente qualquer coisa que estivesse conectado a ele. Três anos depois, ele criou o Televox, um pequeno robô com aspecto humano que conseguia executar movimentos básicos, de acordo com os comandos de seu

operador. Os robôs ganharam ainda mais popularidade com a criação do Willie Vocalite, em 1930. Willie tinha o formato daqueles robôs que vemos nos filmes de ficção antigos. Tinha 2 metros de altura e era feito de aço e da mesma forma que o Televox podia ligar, desligar e regular dispositivos conectados a ele. A grande diferença estava no fato de fazer tudo isto sob comandos de voz, fumava, sentava, ficava de pé, movia os braços e conversava com as pessoas reproduzindo frases gravadas em discos de 78 rotações. Foi a grande sensação da exposição Mundial de Chicago em 1933.

Em 1937 nascia o Elektro, ele também era teleoperado e obedecia a comandos de voz. A grande evolução era que Elektro podia andar, mover a cabeça para cima e para baixo e para os lados. Seus braços se movimentavam separadamente e quando "falava", sua boca coordenava o Movimento das palavras. Além disso, tinha sensores fotoelétricos nos olhos, permitindo que ele distinguisse entre as cores vermelho e verde. Para fazer companhia a Elektro, os pesquisadores e engenheiros da Westinghouse Appliance Division criaram, em 1940, o cachorro robô Sparko. Os dois robôs viajaram pelos EUA em apresentações, feiras e shows. Participaram até de um filme, "Sex Kittens Go to College" em 1960.

Os robôs teleoperados continuaram sendo desenvolvidos ainda por duas décadas. George Devol recebeu as primeiras patentes sobre robótica em 1954. A primeira companhia a produzir um robô industrial foi a Unimation, fundada por George Devol e Joseph F. Engelberger em 1956, sendo baseada nas patentes originais de Devel. Os robôs da Unimation também eram chamados de "máquinas de transferência programadas", visto que sua principal função era a transferência de objetos de um ponto a outro. Elas utilizavam atuadores hidráulicos e eram programados com "conjuntos de coordenadas", podemos considerar como exemplo um robô em que os ângulos de todas as juntas são armazenados durante uma fase de aprendizagem, e então repetidos durante a operação normal. Por muito tempo o único concorrente da Unimation foi a Cincinnati Milacron, de Ohio. Isto mudou radicalmente durante os anos 70, quando grandes conglomerados japoneses começaram a produzir robôs industriais similares. A Unimation havia obtido patentes nos Estados Unidos, porém não as obteve no Japão, que se recusaram a seguir as leis de patentes internacionais, de modo que os projetos foram copiados. Em 1969

Victor Motta inventou o Robô de Stanford na Universidade de Stanford, um robô articulado de 6 eixos, totalmente elétrico, projetado de modo a permitir uma solução utilizando a anatomia de um braço. Isto permitiu que o robô fosse capaz de seguir precisamente caminhos arbitrários no espaço e aumentou as possibilidades de utilizar robôs em aplicações mais sofisticadas tais como montagem e soldagem. Scheinman em seguida projetou um segundo braço para o MIT AI Lab, chamado de "braço do MIT". Sheinman vendeu seus projetos para a Unimation, a qual o desenvolveu com o auxílio do General Motors e posteriormente o comercializou como a Máquina Programável Universal para Montagem (PUMA). Em 1973, a KUKA construiu seu primeiro robô industrial, conhecido como FAMULUS, sendo este o primeiro robô industrial articulado a possuir seis eixos controlados eletronicamente.

O interesse na robótica industrial aumentou no final dos anos 70 e muitas companhias entraram no campo, incluindo grandes empresas como a General Electric e a General Motors (que formaram o empreendimento FANUC Robotics junto com a FANUC do Japão). Dentre as empresas norte americanas temos a Automatix e a Adept Technology. No momento mais intenso do crescimento da robótica em 1984, a Unimation foi comprada pela Westinghouse por 107 milhões de dólares. A Westinghouse vendeu a Unimation para a Stäubli Faverges SCA, da França, em 1988. A Stäubli ainda fabricava robôs articulados para a indústria em geral e aplicações de limpeza até o ano de 2004 e havia inclusive comprado a divisão de robótica da Bosch no mesmo ano.

Eventualmente a visão limitada da indústria americana foi substituída pelos recursos financeiros e grande mercado interno usufruído pelas indústrias japonesas. Apenas um pequeno número de companhias não japonesas fora capaz de se manter nesta área, incluindo a Adept Technology, a Stäubli Unimation, a companhia Sueca Suíça ABB (ASEA Brown-Boveri), a companhia italiana COMAU (pertencente ao Grupo Fiat), a construtora da Áustria IGM Robotersysteme AG e a companhia Alemã KUKA Robotics.

Atualmente (início de 2007), a indústria de braços robóticos está chegando a um estado de maturidade, no qual ela pode prover velocidade, precisão e facilidade de uso suficiente para a maioria das aplicações. O controle por vídeo vem

aumentando enormemente a flexibilidade das unidades robóticas. A parte que atualmente ainda apresenta pouca flexibilidade é a mão, o manipulador fixado aos robôs é comumente uma simples garra pneumática de duas posições. Isto não permite que um robô manipule facilmente diferentes componentes, em orientações diversas. Junto com o aumento de aplicações para dispositivos programados, a calibração dos robôs está se tornando cada vez mais importante, de modo a garantir uma boa precisão no posicionamento. Outros desenvolvimentos incluem a redução no tamanho dos braços industriais para as aplicações voltadas ao consumidor e a utilização dos robôs industriais em combinação com veículos guiados automaticamente (AGVs) mais inteligentes, de modo a tornar a cadeia de automação mais flexível. Os robôs industriais possuem diferentes níveis de autonomia. Alguns robôs são programados para realizarem ações repetidamente sem nenhuma variação, com um nível elevado de precisão. Estas ações são determinadas por rotinas pré-programadas que especificam a direção, aceleração, velocidade e distância de uma série de movimentos coordenados. Outros são mais flexíveis com relação à orientação do objeto em que trabalham ou com o trabalho que realizam sobre o objeto, o qual pode eventualmente ser identificado pelo robô. Por exemplo, para uma orientação mais precisa, os robôs geralmente contêm câmeras, ligadas a computadores ou controladores. A inteligência artificial, e suas variações possuem uma importância crescente nos robôs industriais modernos.

1.1.1 Definições do braço robótico

O braço robótico também é conhecido como máquina manipuladora, com várias anatomias e graus de liberdade (números individuais de movimentos de suas articulações). Fisicamente o braço é constituído por: base, elos, juntas, efetuador final, atuadores, sensores.

Base: Fixa no chão ou em suporte;

Elos: Formam uma cadeia cinemática;

Juntas: Formado pelo punho e uma garra ou ferramenta;

Atuadores: Motores elétricos, hidráulicos ou pneumáticos;

Sensores: Encoders, tacômetros, lasers, câmeras etc.

O braço robótico possui aspectos mecânicos estruturais: Elos e juntas. Elos são os corpos da cadeia, juntas são as articulações entre os corpos que conectam os elos e permitem a realização de movimentos de um elo em relação ao elo anterior. Na figura 1 a exemplificação de juntas e elos.

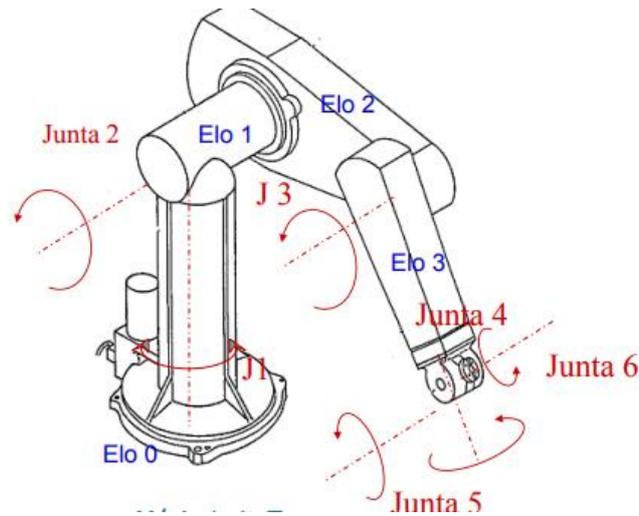


Figura 1 – Junta e elos

Juntas são compostas por dois elos adjacentes e existem dois tipos básicos: Rotação e Prismática (de translação).

A quantidade de juntas equivale aos graus de liberdade do braço. Existem alguns tipos de juntas como veremos a seguir:

Juntas prismáticas (lineares) “P”

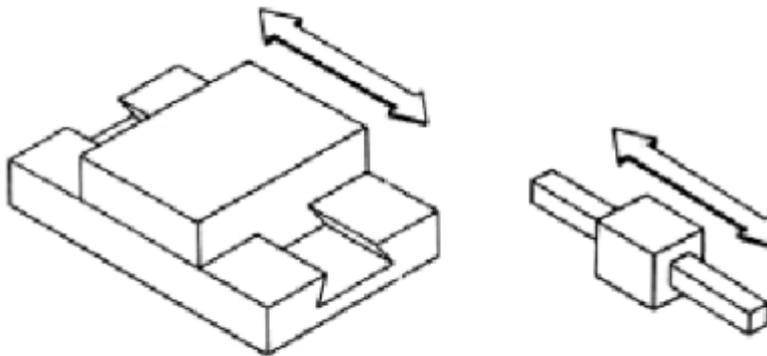


Figura 2 – Junta P

Juntas de Rotação (revolução) “R”

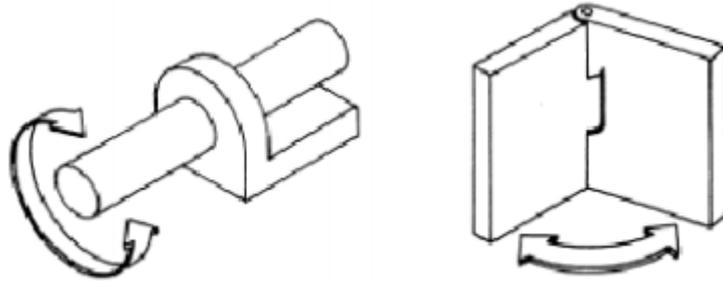


Figura 3 – Junta R

Junta Esférica (rótula)

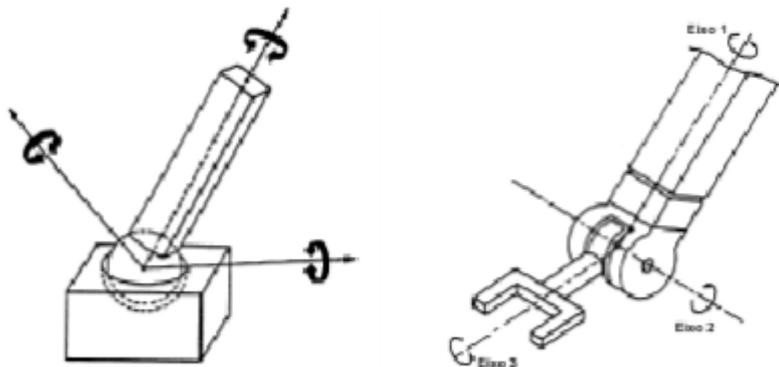


Figura 4 – Junta Esférica

Os braços podem ser classificados por sua estrutura mecânica: Coordenadas cartesianas, pórtico, cilíndricas, esféricas, scara, articulado ou antropomórfico e paralelo.

Ainda existem órgãos terminais: dispositivos fixados no punho do robô que permite o mesmo realizar uma tarefa específica, alguns usam garras para pegar objetos ou ferramentas utilizadas para realizar um trabalho sobre uma peça específica.

1.2 Servo Motor

É um dispositivo eletromecânico que através de um pulso elétrico enviado à sua entrada de sinal, altera seu posicionamento num ângulo que vai de 0° a 180° .

O servo-motor possui um potenciômetro em seu eixo que possibilita o circuito monitorar o ângulo do eixo do servo motor. Recebendo o pulso elétrico, o potenciômetro compara o sinal ao ângulo do eixo do motor, se estiver correto o motor permanece parado, se estiver diferente do sinal recebido, o eixo alterará o posicionamento até chegar ao ângulo desejado.

Faz parte do servo-motor as engrenagens que, por sua vez possuem um limitador que atua no ângulo do eixo, sua função é variar o ângulo entre 0 e 180° e são constituídos por:

Circuito de Controle: Tem por função receber os pulsos elétricos, monitorar a posição do potenciômetro e controlar o motor de acordo com o pulso recebido.

Potenciômetro: Dispositivo ligado ao eixo da saída do servo motor e monitora o eixo do mesmo.

Motor: Para movimentar as engrenagens e o eixo.

Engrenagens: Tem a função de reduzir a rotação do motor, transferindo mais força ao eixo de saída e movimentam o potenciômetro junto com o eixo.

Um ângulo é determinado a partir da duração da largura do pulso enviado à entrada do sinal, este pulso pode ser de 0v ou 5v, esse sinal é monitorado pelo circuito em intervalos de 20ms, se neste intervalo ocorrer qualquer alteração do sinal durante 1ms e 2ms, ele altera a posição para o ângulo desejado, como demonstrado na figura 5.

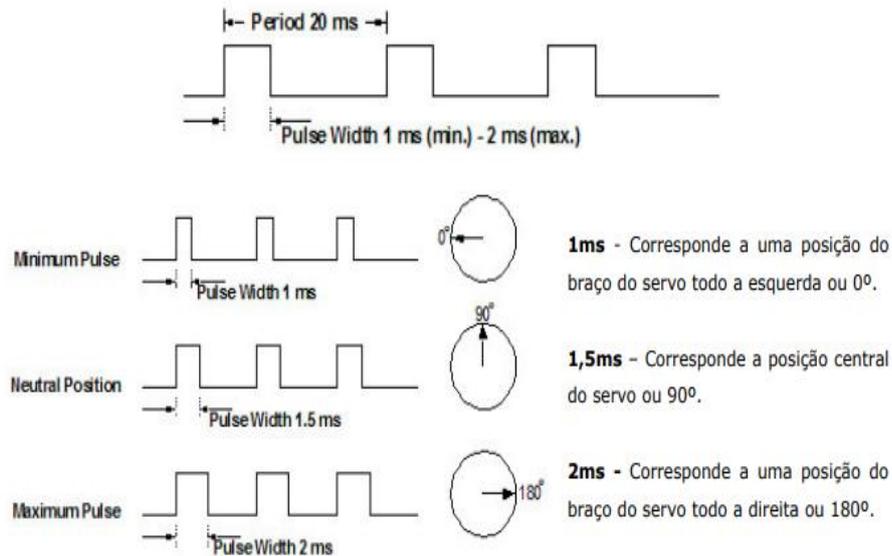


Figura 5 – Pulsos

Ao receber um sinal, o potenciômetro analisa se o eixo está na posição correta, se não estiver, o circuito aciona o motor até o potenciômetro estar na posição desejada, a direção da rotação dependerá da posição do potenciômetro, ou seja, o motor girará as engrenagens para a direção que entender ser mais rápido chegar a posição correta, abaixo a figura 6 a descrição das partes de um servomotor.

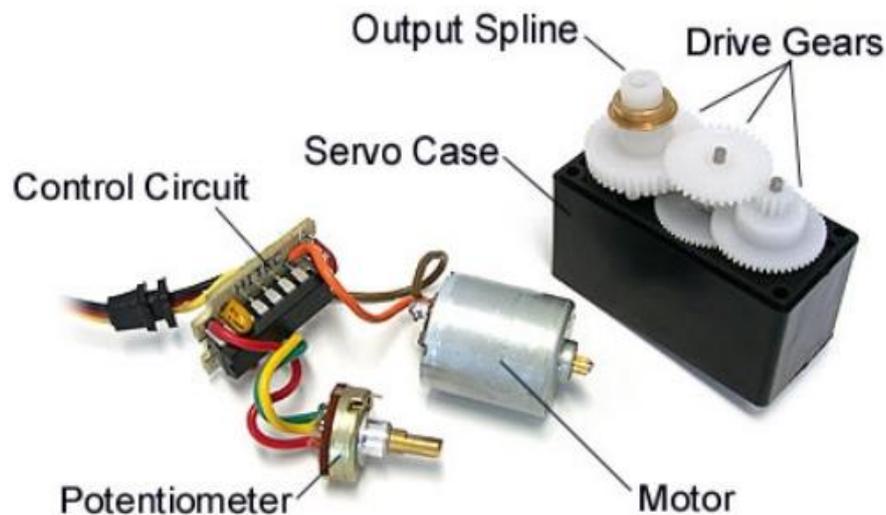


Figura 6 – Servo-motor

O servo-motor é muito utilizado em controle de precisão em projetos de automação industrial. Com custos cada vez mais baixos, o servo-motor vem se tornando uma excelente alternativa em substituição a acionamentos com motores de indução, atuadores hidráulicos e pneumáticos.

Existem dois tipos de servo-motor: os assíncronos e síncronos.

ASSÍNCRONO: Motor de indução, trifásico ou monofásico, por corrente alternada AC, seu nome se dá por conta do seguinte fato: O circuito quando alimentado induz uma corrente no rotor que, pela Lei de Lenz, tende a se opor ao campo que o gerou, desta forma, o campo gerado acompanha o campo gerador. Se a velocidade do rotor for igual a do campo gerador (ou também chamado campo magnético girante), a indução no rotor é zero e o motor não funciona. Esta diferença de velocidade é chamada **ESCORREGAMENTO** e sua velocidade está ligada ao número de pares de polos.

SÍNCRONO: Como o seu nome diz, a velocidade do rotor é igual a velocidade do campo magnético.

1.3 Arduino

O Arduino foi criado em 2005 pelos pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, com o objetivo era elaborar um dispositivo barato, funcional, de fácil de programação, acessível a estudantes e projetistas amadores. Foi adotado o conceito de hardware livre, ou seja, qualquer um pode montar, modificar, melhorar e personalizar o Arduino. Foi criada uma placa composta por um microcontrolador Atmel, circuitos de entrada e saída, utiliza uma linguagem baseada em C/C++, usando apenas um cabo USB ligado ao computador, a seguir na figura 7 a imagem de um arduino uno.



Figura 7 – Arduino

Uma vez programado, o microcontrolador pode ser usado para controlar qualquer coisa seja, um robô, um ventilador, as luzes da sua casa, a temperatura do ar condicionado, etc.

1.3.1 Módulos e Shields

O Arduino possui muitos sensores e componentes para utilização nos projetos e grande parte do material utilizado está disponível em módulos, que são pequenas placas que contém os sensores e outros componentes auxiliares como resistores, capacitores e leds, abaixo na figura 8 a imagem de módulos para arduino.



Figura 8 - Módulos

Existe também os Shields, que são placas para encaixe no Arduino para expandir suas funcionalidades. A imagem abaixo mostra um Arduino Ethernet Shields encaixado no Arduino Mega 2560, ele permite o acesso a uma rede ou até mesmo à internet, mantêm os demais pinos livres para utilização, por exemplo, utilizar os pinos para controlar temperatura e umidade de um ambiente, e consultar esses dados de qualquer lugar do planeta. Na figura 9 temos um exemplo de shields.



Figura 9 – Shields Ethernet

1.3.2 Modelos de Arduino

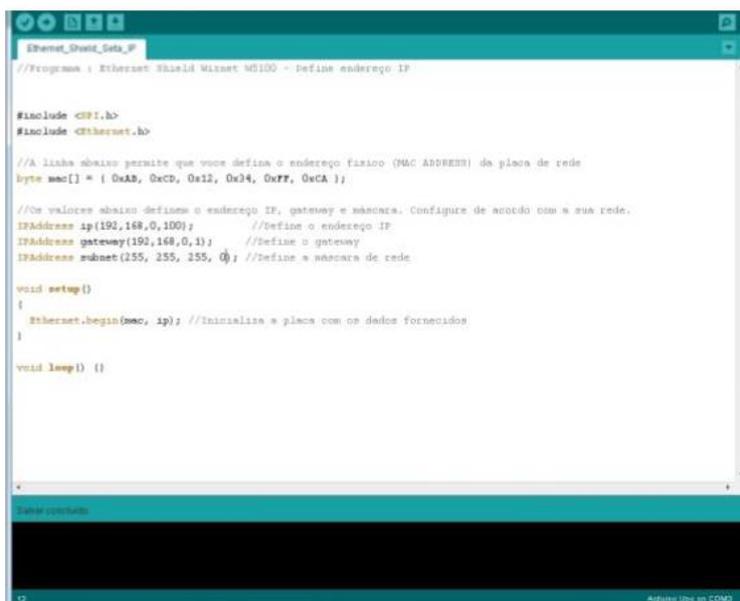
O tipo de placa que você vai utilizar depende do projeto a ser desenvolvido. As opções vão das mais comuns, como o Arduino Uno até as placas com maior poder de processamento, como o Arduino Mega, o Arduino Due, abaixo na figura 10 a imagem de tipos de arduino.



Figura 10 – Tipos de Arduino

1.3.3 Programação do Arduino

É preciso conectá-lo ao computador por meio de um cabo USB e utilizar um ambiente de programação chamado IDE, lá você digita o programa, faz os testes para encontrar eventuais erros e transfere o programa para o dispositivo. Na figura 11 temos a IDE já com um programa

A screenshot of the Arduino IDE showing a C++ program for an Ethernet shield. The code is as follows:

```
Ethernet_Shield_Geta_IP
//Programa : Ethernet Shield Wiznet W5100 - Define endereço IP

#include <SPI.h>
#include <Ethernet.h>

//A linha abaixo permite que voce defina o endereço fixado (MAC ADDRESS) da placa de rede
byte mac[] = { 0xAB, 0xCD, 0x12, 0x34, 0xFF, 0xCA };

//Os valores abaixo definem o endereço IP, gateway e máscara. Configure de acordo com a sua rede.
IPAddress ip(192,168,0,100); //Define o endereço IP
IPAddress gateway(192,168,0,1); //Define o gateway
IPAddress subnet(255, 255, 255, 0); //Define a máscara de rede

void setup()
{
  Ethernet.begin(mac, ip); //Inicializa a placa com os dados fornecidos
}

void loop() {}
```

Figura 11 – IDE de programação

Uma vez feito o programa, transfira-o para o Arduino e ele começará a funcionar. Você não precisa ser expert em linguagem, você pode começar um programa utilizando a estrutura básica do Arduino, que é composta por duas partes:

Setup – Nessa parte você configura as opções iniciais: os valores iniciais de uma variável, se uma porta será utilizada como entrada ou saída, etc.

Loop – Nessa parte você repete uma estrutura de comandos de forma contínua ou até que algum comando de “parar” seja enviado ao Arduino.

1.3.4 Hardware do Arduino

Existem alguns tipos de Arduino, assim suas formas são distintas umas das outras, vamos verificar abaixo na figura 12 o hardware de um Arduino Uno:

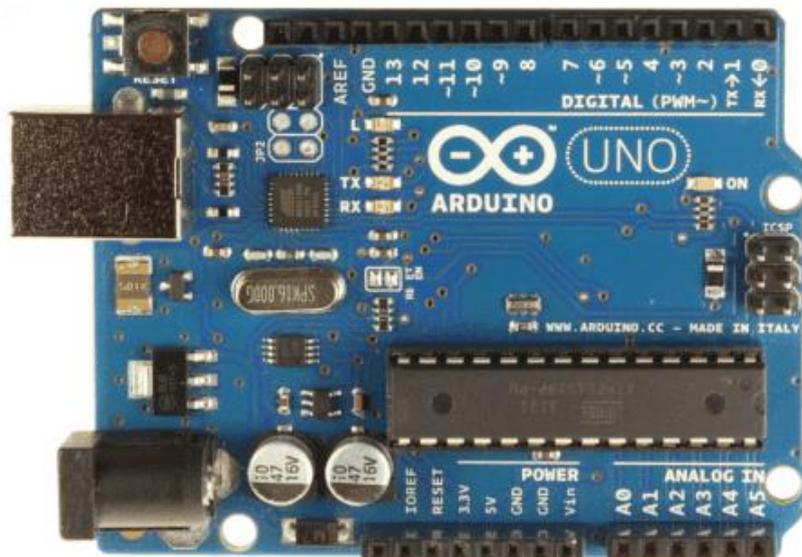


Figura 12 – Arduino Uno

Conforme visto acima o arduino uno é constituído por:

- 14 pinos de entrada e saída digital (pinos 0-13):

Esses pinos podem ser utilizados como entradas ou saídas digitais de acordo com a necessidade do projeto e conforme foi definido no sketch criado na IDE.

- 6 pinos de entradas analógicas (pinos A0 - A5):

Esses pinos são dedicados a receber valores analógicos, por exemplo, a tensão de um sensor. O valor a ser lido deve estar na faixa de 0 a 5V onde serão convertidos para valores entre 0 e 1023.

- 6 pinos de saídas analógicas (pinos 3, 5, 6, 9, 10 e 11):

São pinos digitais que podem ser programados para serem utilizados como saídas analógicas, utilizando modulação PWM.

A alimentação da placa pode ser feita a partir da porta USB do computador ou através de um adaptador AC. Para o adaptador AC recomenda-se uma tensão de 9 a 12 volts.

1.4 PWM

PWM (Pulse Width Modulation) ou Modulação de Largura de Pulso é uma técnica para variar o valor médio de uma forma de onda, ou seja, podemos controlar o tempo em que uma onda quadrada fica em nível lógico alto, mantendo sua frequência. É como se o circuito, tivesse uma chave que fica abrindo e fechando em uma determinada frequência.

Na figura 13 abaixo a onda fica 50% do tempo fechado e 50% do tempo aberta.

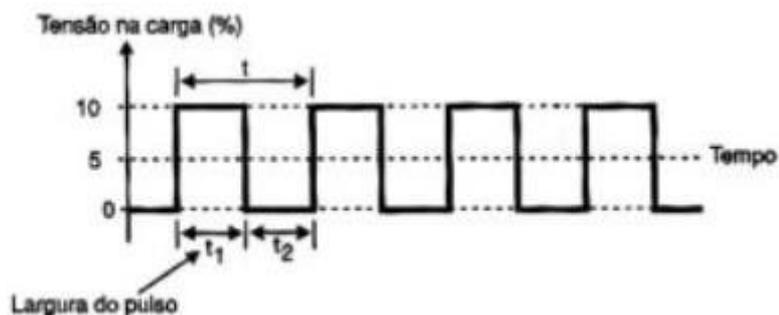


Figura 13 –PWM

É isso que o PWM faz, pode-se regular o tempo em que o sinal estará em nível lógico alto em uma determinada frequência. Esse tempo também é chamado de Duty Cycle ou Ciclo Ativo.

Em suma, controlamos o PWM variando a largura do pulso para controlar o ciclo do sinal e sua potência.

1.5 Bluetooth

O Bluetooth é uma tecnologia de comunicação sem fio que permite a troca de dados e arquivos entre celulares, computadores e demais dispositivos de forma rápida e segura.

Bluetooth foi o nome dado à tecnologia em homenagem ao antigo rei da Dinamarca e Noruega, Harold Blatand (em inglês, Harold Bluetooth), com a façanha de unificar as tribos norueguesas, suecas e dinamarquesas, já que a tecnologia é uma forma de unificação de dispositivos. O logotipo e símbolo do Bluetooth também são baseados no nome de Harold, já que é formado às iniciais do nome do rei, “H” e “B”, respectivamente.

1.5.1. Como funciona o Bluetooth?

O sistema utiliza uma frequência de rádio de onda curta (2.4 GHz) para criar comunicação entre aparelhos habilitados. Como seu alcance é curto, só permite comunicação entre dispositivos próximos e consome pouca energia.

A comunicação do Bluetooth se dá através de uma rede chamada piconet, ela permite conexão de até oito dispositivos, porém, para aumentar a quantidade, é possível sobrepor mais piconets, capacitando o aumento de conexões através do método scatternet.

Embora já existam classes de Bluetooth com alcance de 100 metros, os dispositivos em sua maioria contam com alcance de 1 a 10m, embora seja uma desvantagem, ajuda na segurança dos usuários.

Outra garantia de ambiente seguro, é que antes de efetuar trocas de dados e arquivos entre aparelhos que dispõem de tecnologia Bluetooth, normalmente deve-se ativar a função através das configurações dos dispositivos.

1.6 Esteiras rolantes

Esteiras rolantes estão cada vez mais incluídas no nosso dia a dia, como nos terminais de aeroportos, terminais de carga, terminais rodoviários, linhas de produção, facilitando o deslocamento de usuários em supermercados onde a garagem é projetada em pavimentos inferiores ao piso do estabelecimento, a esteira rolante é a candidata ideal para viabilizar o transporte do carrinho de compras até o estacionamento onde se encontra o automóvel do cliente.

O surgimento da patente de esteira usada para transportar pessoas é datado em 1871 pelo inventor Alfred Speer, já a primeira a ser construída foi em Chicago, operada pela empresa Columbian Movable Company, projetada para transportar até 31.680 pessoas por hora, porém um trágico incêndio destruiu a esteira no ano seguinte.

Em 1900, Speer e Max Schmidt desenvolveram outra esteira, a exemplo de sua antecessora, para uma exposição em Paris com a ideia de melhorar o tráfego de pessoas na cidade, porém, houve uma preocupação com a manutenção do equipamento e medidas de segurança para as pessoas utilizarem o transporte na chuva ou na neve, viabilizando os trens e ônibus urbanos e acabou fazendo os projetos de esteiras rolantes serem esquecidos. Meio século depois, com o crescente tráfego de pessoas nos centros urbanos, foi reconsiderado o projeto de transporte de pessoas por esteiras rolantes, mas dessa vez com o objetivo de auxiliar em trajetos de curtas distâncias, a primeira esteira concebida sob este novo objetivo foi instalada em 1954 na cidade de Jersey, construída pela Goodyear.

1.6.1 Constituição da esteira

A esteira é composta por uma correia, motor DC, tambores e roletes. As correias são para alimentação ou saída de produtos de máquinas e/ou dispositivos.

A correia fica entre dois tambores apoiando-se sobre roletes durante o percurso, abaixo na figura 14 a imagem de um exemplo de esteira.



Figura 14 - Esteira transporte por correia

Tambores: são peças que impulsionam a correia transportadora a deslizar pelos roletes, abaixo na figura 15 a imagem dos tambores.



Figura 15 – Tambores

Roletes: Conjunto de rolos, geralmente cilíndricos, é utilizada para dar suporte à movimentação da correia e guia-la, abaixo na figura16 a imagem de um tipo de rolete.



Figura 16 – Roletes

1.7 Sensores

Sensor é um dispositivo que faz a detecção e responde com eficiência a algumas entradas provenientes de um ambiente físico, por exemplo: Uma luz, o calor, movimento, umidade, pressão ou qualquer variável detectável em um ambiente, assim, quando um sensor recebe uma entrada específica proveniente do ambiente, é emitido um sinal capaz de ser convertido e interpretado por dispositivos como CLP ou uma IHM capaz de mostrar esta informação em uma tela por exemplo, ou seja, ao ser convertido, o sinal pode ser lido por um processador ou ser transmitido eletronicamente por uma rede de dados.

É comum quando falamos de sensores, também falamos dos transdutores, eles são utilizados juntos quando tratamos medições. O primeiro termo (sensores), é muito popular na América, já o transdutor é mais popular na Europa.

A palavra sensor significa “perceber” e a palavra transdutor significa “levar a frente”. Se procurarmos esses os termos no dicionário, saberemos que o primeiro significa “dispositivo que detecta uma mudança no ambiente físico e transforma em um sinal que pode ser medido e gravado, enquanto que transdutor é definido como dispositivo que transfere a energia de um sistema para outro que pode ser da mesma forma ou de forma convertida (diferente do original).

Uma distinção entre eles é que o sensor é aplicado para a detecção em si, já o transdutor aplica-se para o elemento de detecção associado a qualquer circuito. Sendo assim, podemos dizer que todos os transdutores possuem um sensor e a maioria dos sensores são transdutores.

A Figura17 abaixo mostra um processo de detecção, em termos de conversão de energia. A forma do sinal de saída, será muitas vezes semelhante ao sinal de entrada, embora algumas vezes pode ser uma forma de onda cuja frequência seja proporcional a entrada ou um trem de pulsos que contem a informação. No caso da Figura coloco um exemplo com termopar sendo convertido o sinal por um transdutor sendo que este transdutor converte o sinal do termopar em uma saída

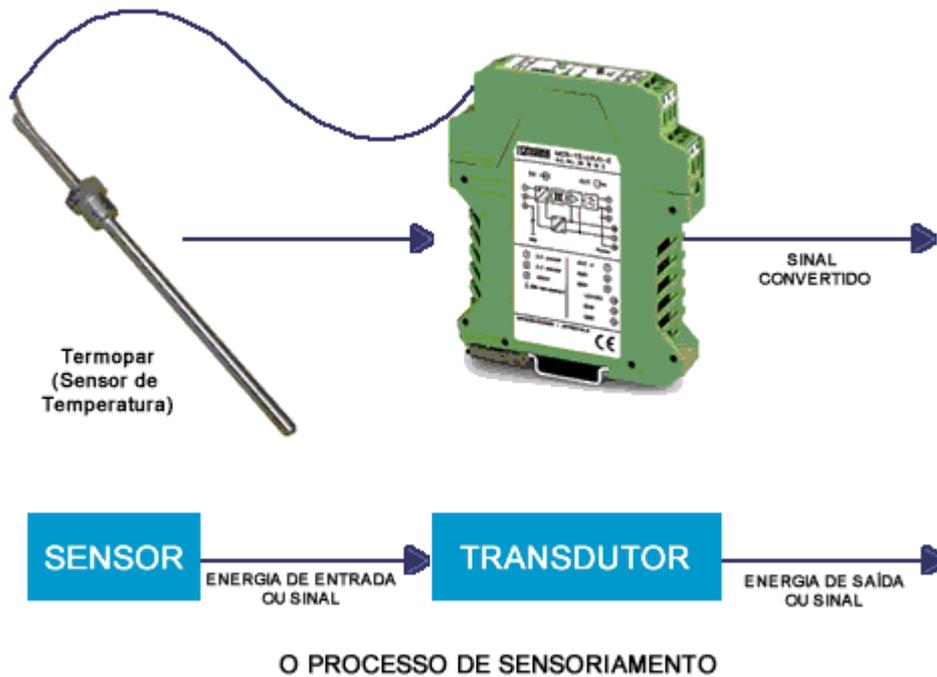


Figura 17 – processo de detecção

1.8 Fonte de alimentação DC.

A **fonte de alimentação** é o dispositivo que transforma tensão alternada em tensão DC, através de um redutor de tensão que transforma, por exemplo, 127VAC em 5 VAC, retificando essa tensão que se torna tensão contínua pulsante formada pelos semciclos alternados que são conduzidos pelo diodo, depois de retificada essa tensão passa por filtros para que fica parecendo com a figura 18 abaixo.

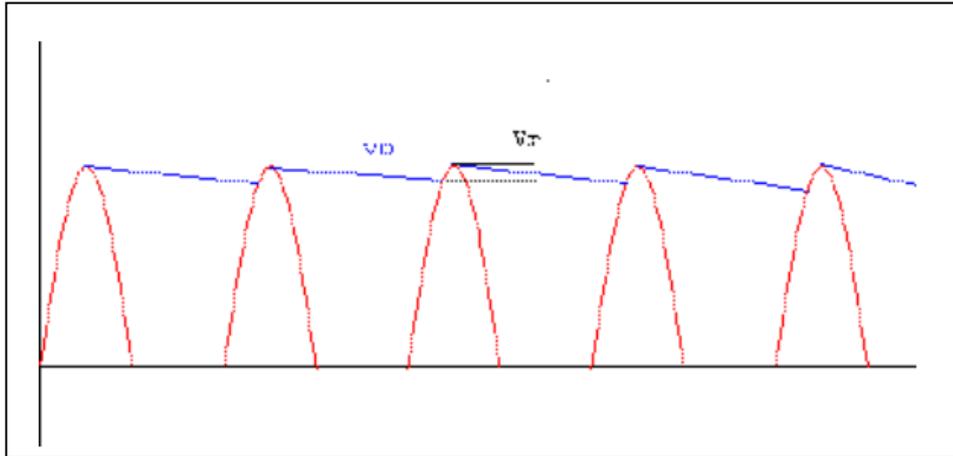


Figura 18 – Retificação

Em seguida é necessário se estabilizar a tensão, é possível de duas maneiras através de um diodo zener ou um regulador de tensão da família LM78XX como mostra nas figuras 19 e 20.

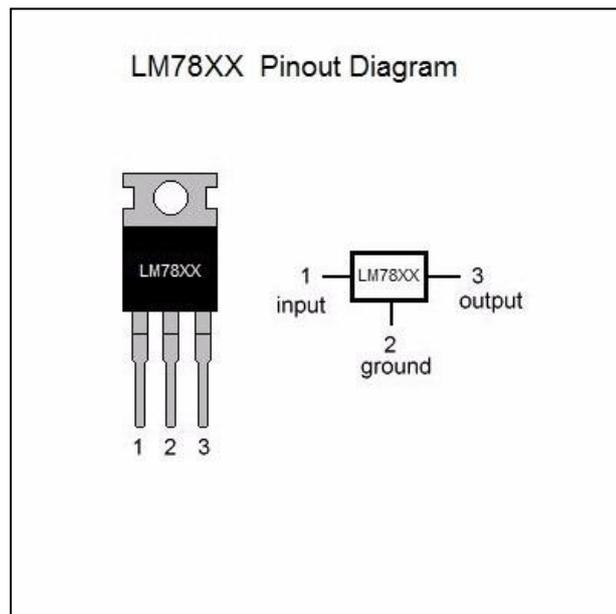


Figura 19 – Regulador de tensão



Figura 20 – Diodo Zener

2 Planejamento

O Primeiro passo foi decidir qual seria o dispositivo para o acionamento e controle do braço e da esteira, CLP ou o Arduino. O arduino foi escolhido como a melhor opção, porque possui controles de PWM e pelo melhor custo benefício, o CLP teria que utilizar circuitos adicionais de para controle dos servos motores.

2.1 Pesquisas de Componentes/Tecnologias

Seria utilizado o modelo de CLP CLICK2 do fabricante WEG pela facilidade de programação, a seguir na figura 21 a imagem do CLP.



Figura 21 – CLP CLIC 2 WEG

Decidiu-se pelo Arduino Mega pela sua quantidade entradas e saídas, como mostra na figura 22 a imagem abaixo:

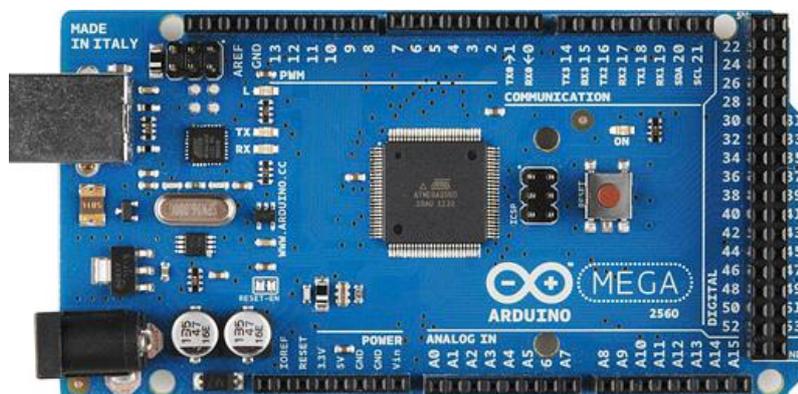
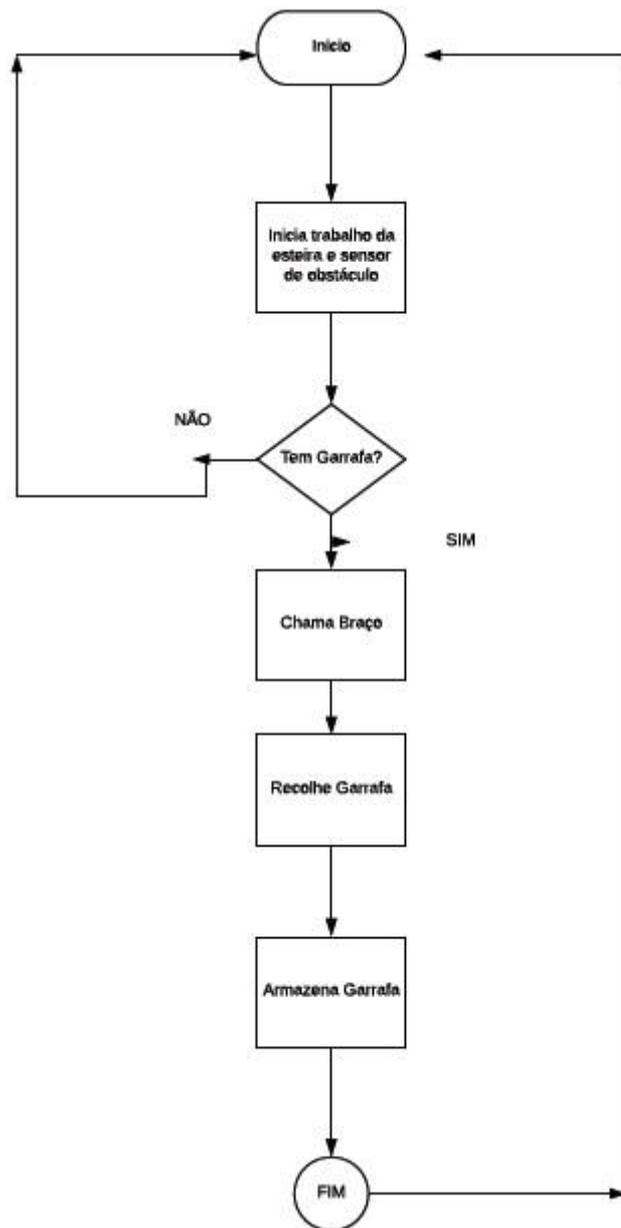


Figura 22 – Arduino Mega

2.1.2 Fluxograma do processo e cronograma



2.1.3 Tabela de custo

Tabela de Custo Projeto			
Item	Quantidade	Valor Unitário	Valor Total
Braço em MDF	1	R\$ 74,00	R\$ 74,00
Esteira	1	R\$ 148,35	R\$ 148,35
Caixa de comando em MDF	1	R\$ 14,00	R\$ 14,00
Shield Bluetooth	1	R\$ 35,00	R\$ 35,00
Arduino Mega	1	R\$ 300,00	R\$ 300,00
Placa Circuito impresso	1	R\$ 5,00	R\$ 5,00
Cabo de Rede	2	R\$ 2,00	R\$ 4,00
Cabo Serial	1	R\$ 25,00	R\$ 25,00
Servos	4	R\$ 12,00	R\$ 48,00
Cabo Arduino Macho	2	R\$ 16,00	R\$ 32,00
Cabo Arduino Fêmea	2	R\$ 16,00	R\$ 32,00
Motor DC 3-6v 80RPM com caixa de redução	1	R\$ 15,90	R\$ 15,90
Fonte Chaveada	1	R\$ 35,90	R\$ 35,90
Espaguete Termo Retrátil	2	R\$ 3,00	R\$ 6,00
Espaguete Termo Retrátil	3	R\$ 0,16	R\$ 0,48
Frete	1	R\$ 55,00	R\$ 55,00
Resistores 1k	6	R\$ 0,20	R\$ 1,20
Botão de emergência	1	R\$ 45,50	R\$ 45,50
Botão liga desliga	1	R\$ 4,80	R\$ 4,80
Contatos NA\NF	4	R\$ 7,50	R\$ 30,00
Caneleta	1	R\$ 12,50	R\$ 12,50
Sensor de Barreira	1	R\$ 12,00	R\$ 12,00
Aluguel do armário	1	R\$ 65,00	R\$ 65,00
Jogo de mão Francesa	1	R\$ 4,90	R\$ 4,90
Botão Pulsador	1	R\$ 2,50	R\$ 2,50
Cadeado	1	R\$ 20,00	R\$ 20,00
Barra Roscada	2	R\$ 4,50	R\$ 9,00
Porca 1/8"	30	R\$ 0,05	R\$ 1,50
Arruela 1/8"	30	R\$ 0,05	R\$ 1,50
Placa de MDF encampada 63,5 x 46	1	R\$ 78,95	R\$ 78,95
Rolo de Fita Isolante	1	R\$ 16,00	R\$ 16,00
Serra com suporte	1	R\$ 17,00	R\$ 17,00
Diodo 1n4007	2	R\$ 0,12	R\$ 0,24
EDT003 - Interface de Potência com Relé	1	R\$ 20,00	R\$ 20,00
Abraçadeira Nylon 100 x 1,5	1	R\$ 3,00	R\$ 3,00
Banner	1	R\$ 70,00	R\$ 70,00
Total de peças	113	R\$ 1.141,88	R\$ 1.246,22

ETEC JORGE STREET - CRONOGRAMA PLANEJAMENTO DO TRABALHO DE CONCLUSÃO DE CURSO (CCC - 1º SEMESTRE 2018)
TURMA: TÉCNICO EM AUTOMAÇÃO INDUSTRIAL

DATA: 11 /06 /2018

Nº	ATIVIDADE	Projeto : Braço Robótico e esteira												ANDAMENTO (%)				RESPONSÁVEL			
		Março			Abril			Maio			Junho			20	40	60	80		100		
		S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A	S E M A N A										
1	Programação do CLP	P	R	X																Todos	
2	Programação do Arduino	P	R		X																Todos
3	Pesquisa de materiais para conexão do braço	P	R			X															Todos
4	Pesquisa de materiais para conexão da esteira	P	R				X														Todos
5	Aquisição da esteira, braço e PWM	P	R				X														Todos
6	Montagem do braço	P	R					X													Todos
7	Montagem Da esteira	P	R						X												Todos
8	Montagem do braço	P	R							X											Todos
9	Montagem do PWM	P	R								X										Todos
10	Testes do braço com CLP	P	R								X										Todos
11	Testes do braço com Arduino	P	R							X											Todos
12	Teste do Braço no Arduino com a Esteira	P	R								X										Todos
13	Avaliação dos Testes	P	R									X									Todos
14	Ajustes Finais	P	R										X								Todos
15	Entrega da monografia	P	R											X							Todos
16	Banca	P	R												X						Todos
17		P	R																		
18		P	R																		
19		P	R																		
20		P	R																		

ALUNOS E RESPONSABILIDADES DESTES PROJETO

NO.	RM	NOME
1	26104	Lider do Grupo:Wilson Roberto Cardoso
2	26345	Diário de Bordo:Jamili Celestino Costa
3	26167	Monografia:Ceiso Francisco dos Santos
4	26370	Croquis Completo:Francisco Israldo Moura Macedo
5	26320	Lista de materiais:Geison Ferreira Lima
6	26433	Custos Completo:Rodrigo Alves da Silva
7	26288	FMEA:Luz Guilherme Gomes
8	26324	Cronograma para o PTCC:Leandro dos Santos Rodrigues
9	26345	Cronograma para o DTCC:Jamili Celestino Costa
10		

BANCA Recuperação

2.1.4 Desenhos da esteira e do braço

Nas bases figuras a seguir os desenhos dos das partes que compõe o braço.

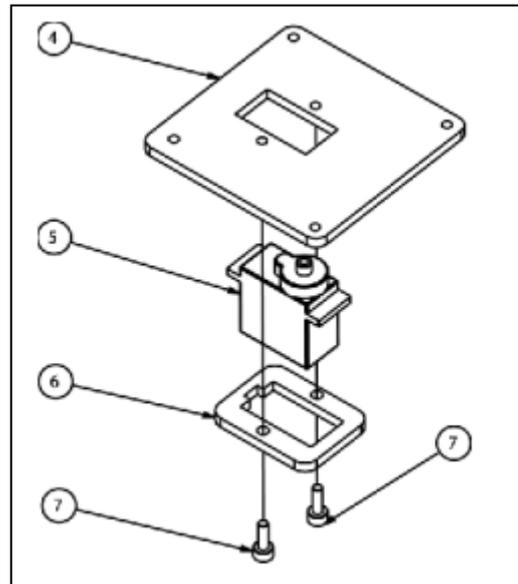


Figura 23 – parte superior da base

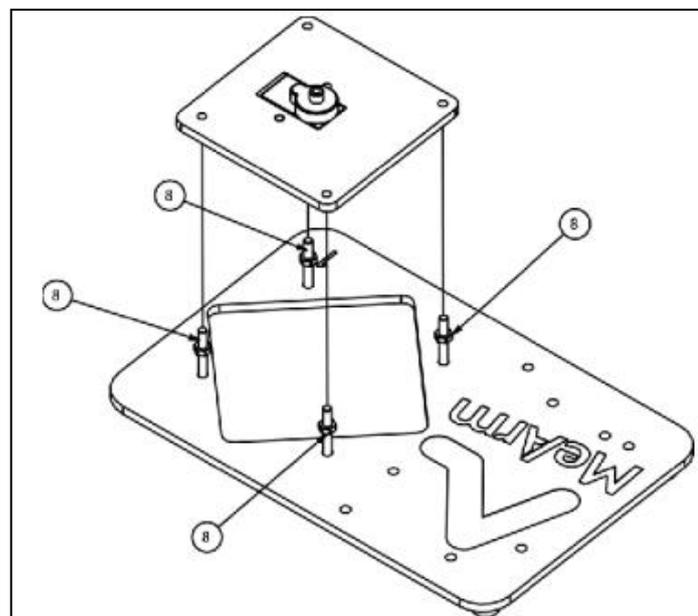


Figura 24 – Parafusos e parte inferior e superior da base

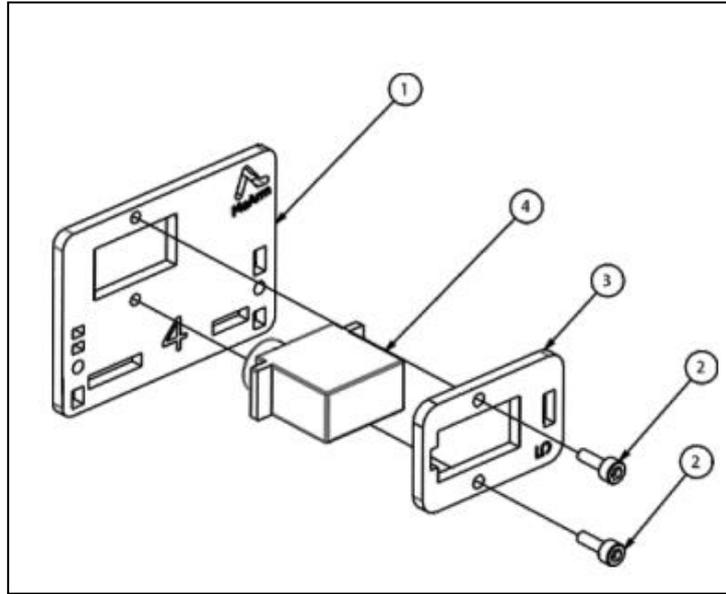


Figura 25 – Lateral da base

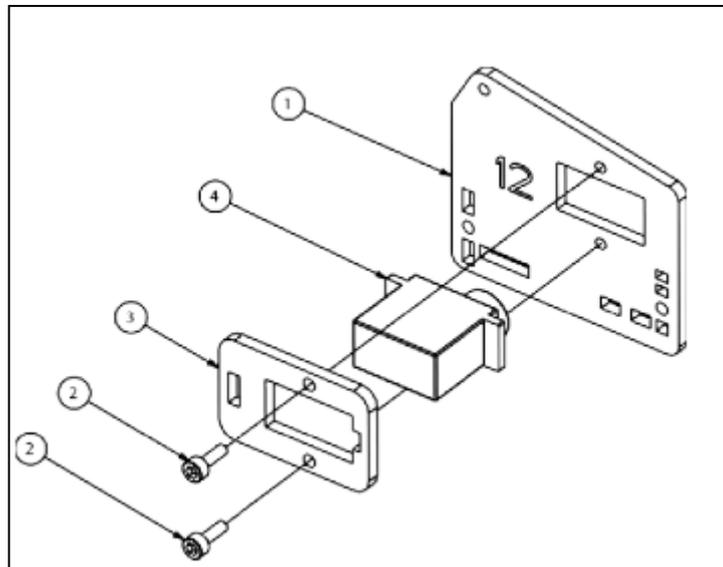


Figura 26 – Lateral da base

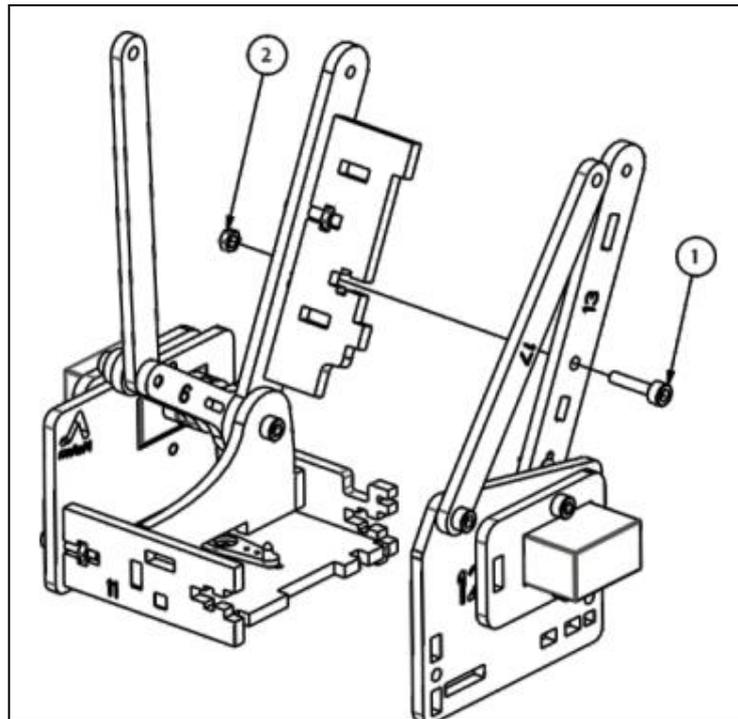


Figura 27 – Laterais da base mais hastes dos braços

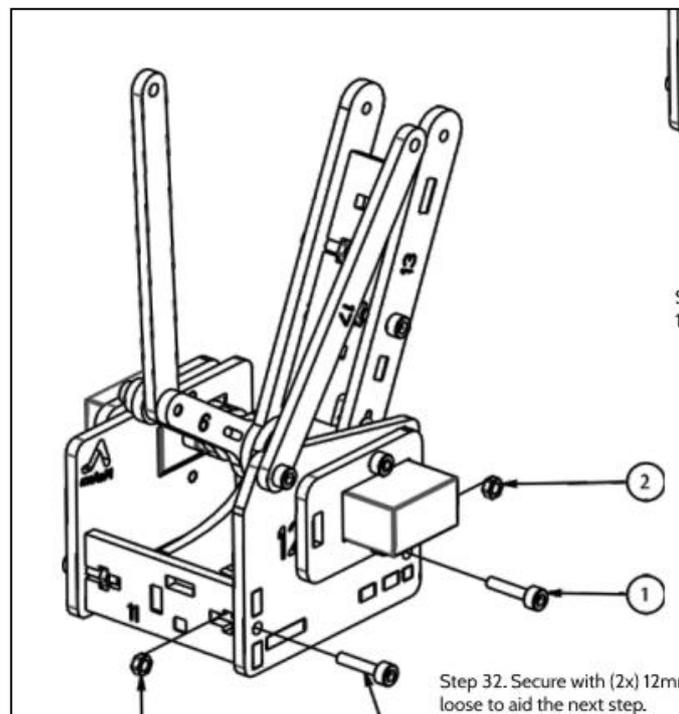


Figura 28 – Laterais da base mais hastes dos braços

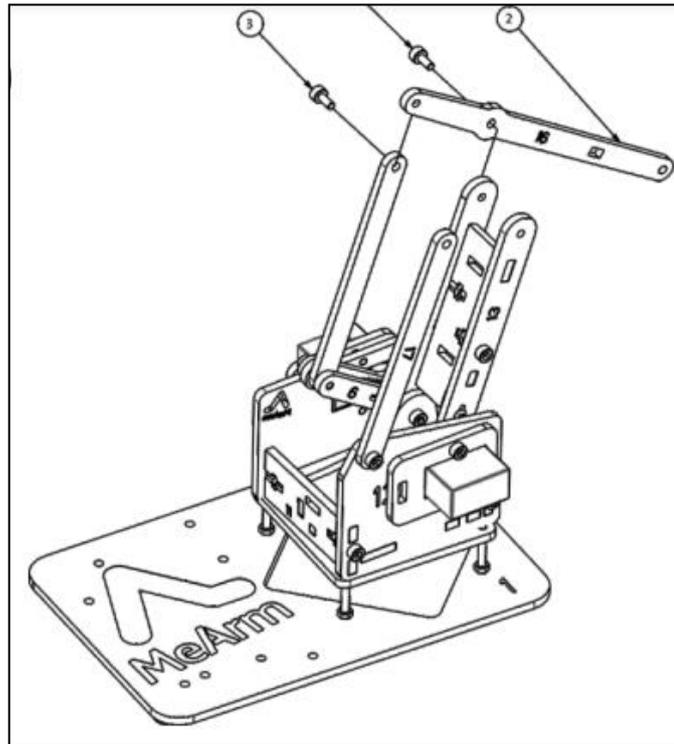


Figura 29 – Fixação do Cotovelo

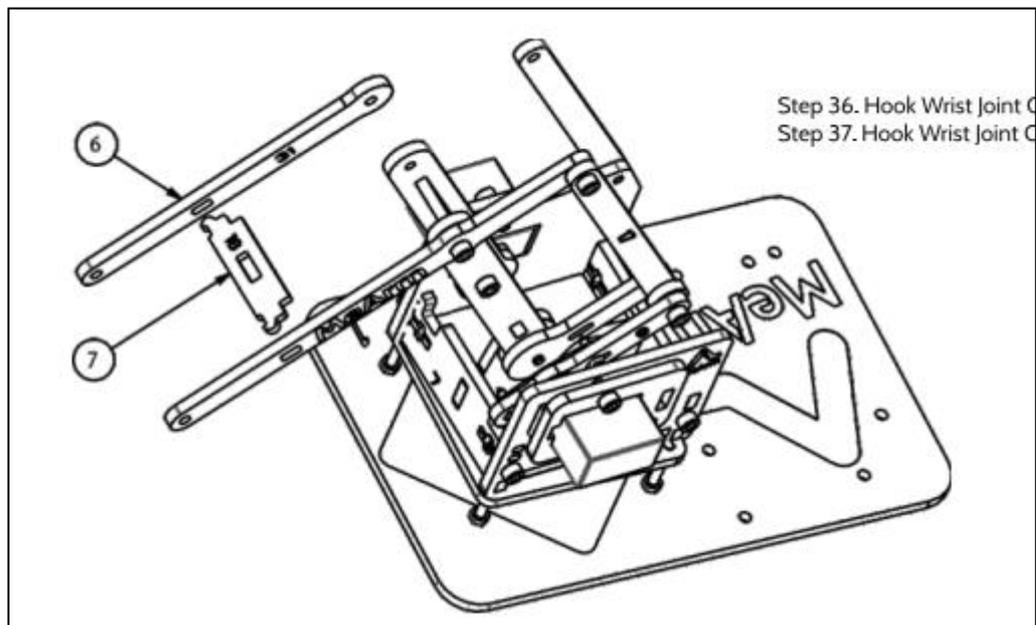


Figura 30 – Fixação do Cotovelo mais separador das hastes do cotovelo

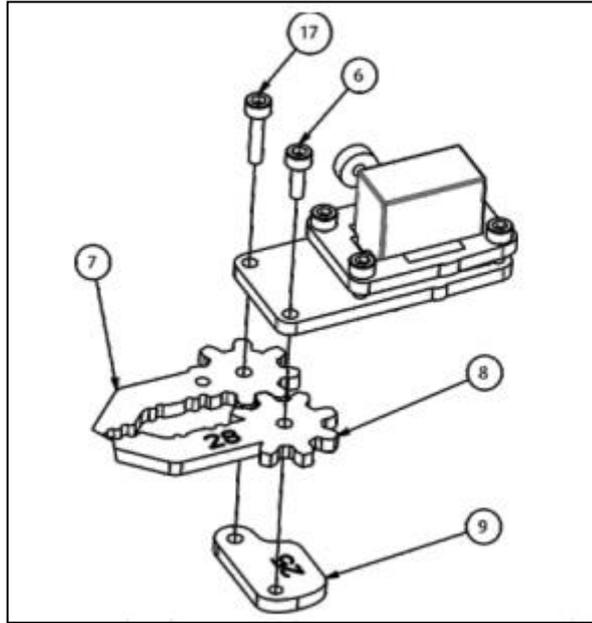


Figura 33 – Fixação das partes que compõe a garra

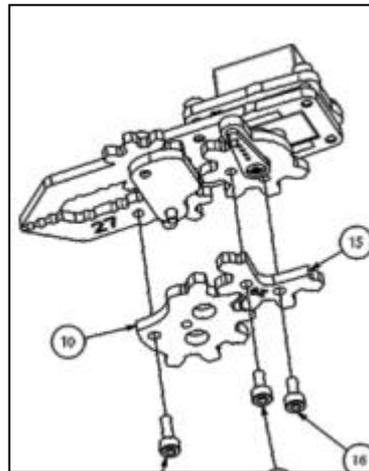


Figura 34 – Fixação dos parafusos da garra

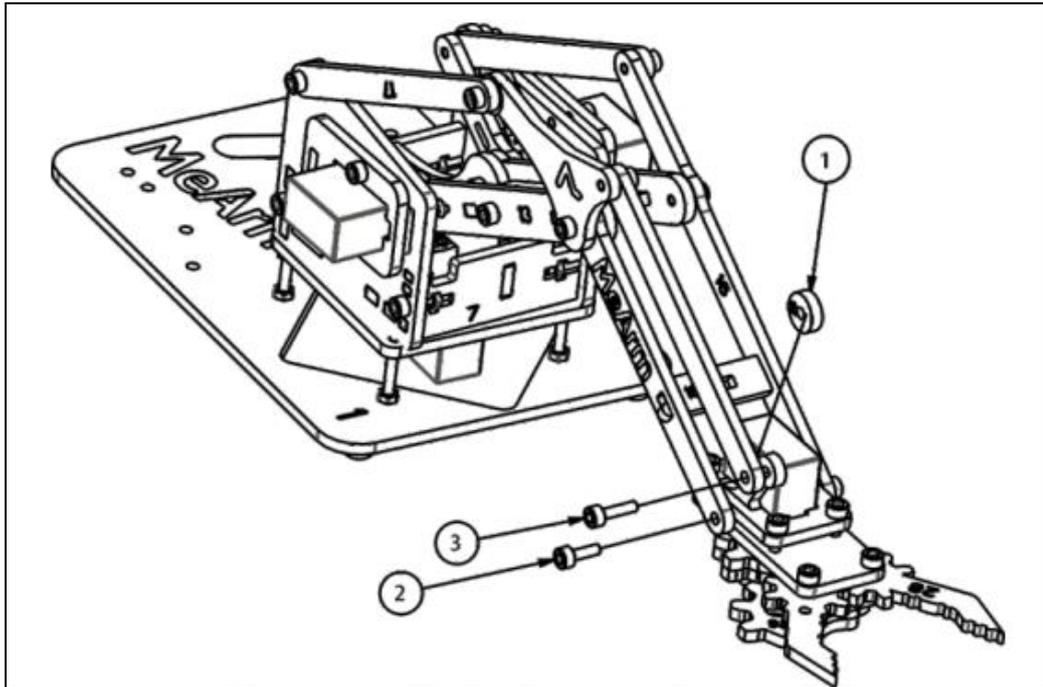


Figura 35 – Junção da garra com o cotovelo

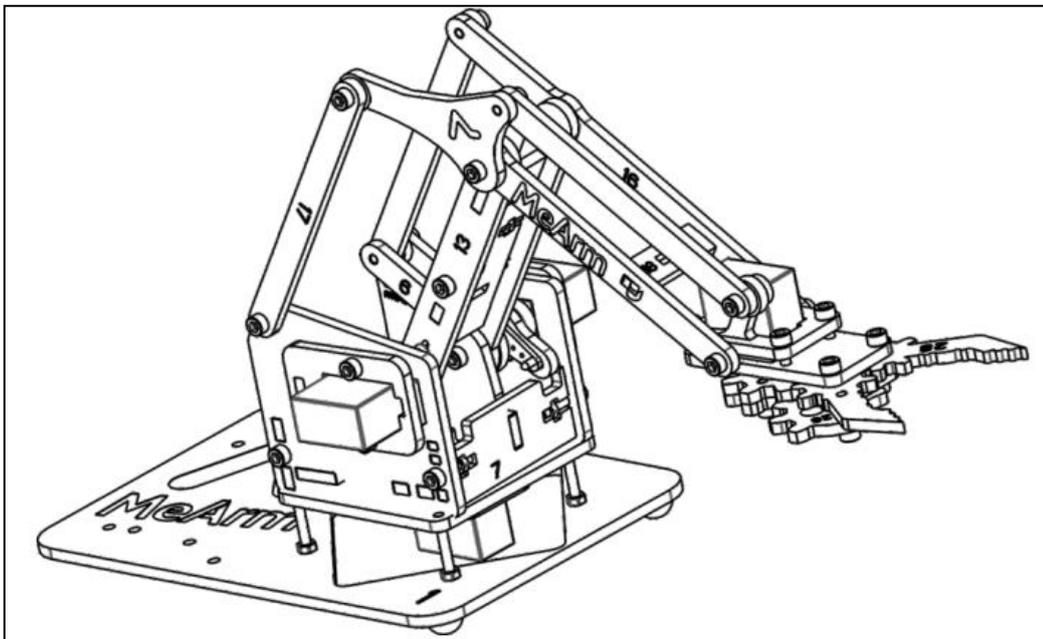


Figura 36 – Junção da garra com o cotovelo

3 Testes

Inicialmente foi utilizado o CLP para realizar o controle do braço robótico, porém, a velocidade em que o braço operava era muito alta e para conseguir controlar essa velocidade seria necessário a utilização de um controlador PWM, abaixo segue a figura 20 da imagem do controlador PWM que seria utilizado no CLP para controle de velocidade dos servo-motores.

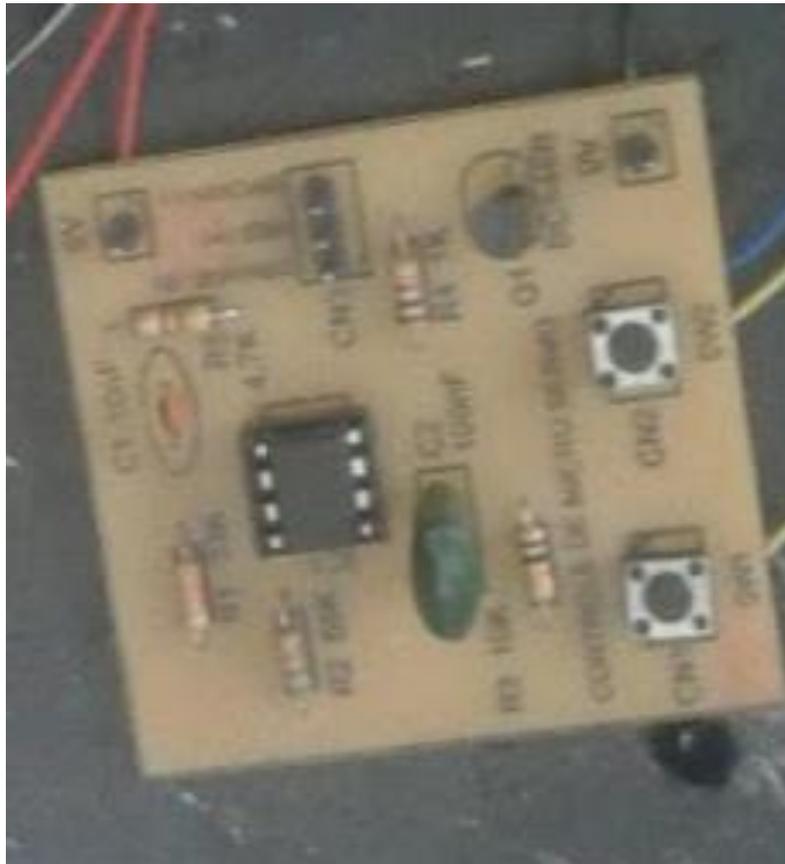


Figura 37 – Circuito de Controle para servo motor (PWM).

Percebeu-se que o braço perdia força, decidiu-se então pela utilização do Arduino Mega, pois este já possui controlador PWM embutido, então conseguiria controlar a velocidade do motor, reduzindo o custo com componentes, a seguir na figura 38 abaixo uma imagem do Arduino utilizado no projeto.



Figura 38 – Arduino Mega.

Adquiriu-se um protótipo de um braço robótico e de uma esteira conforme na figura 39 e 40 nas imagens a seguir.

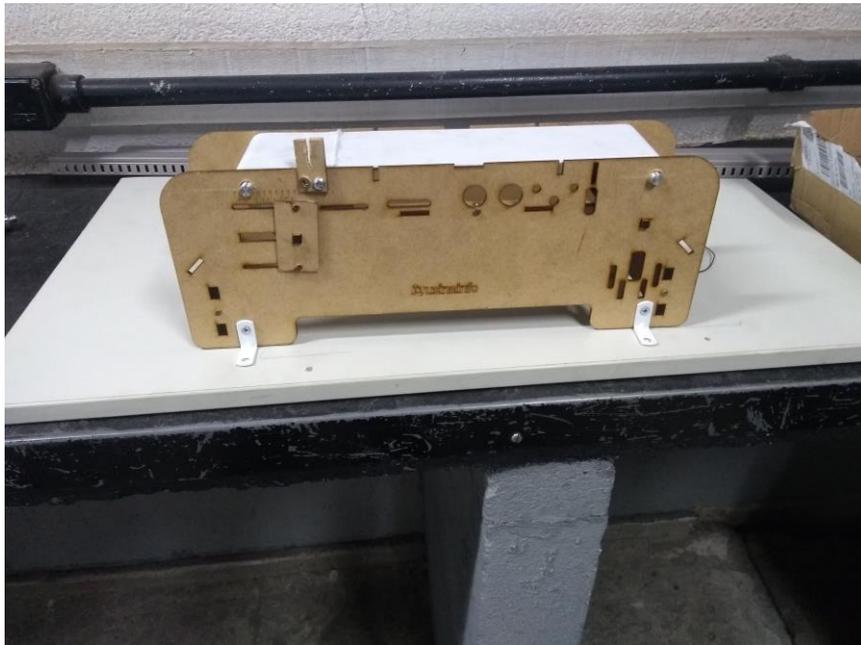


Figura 39 – Esteira

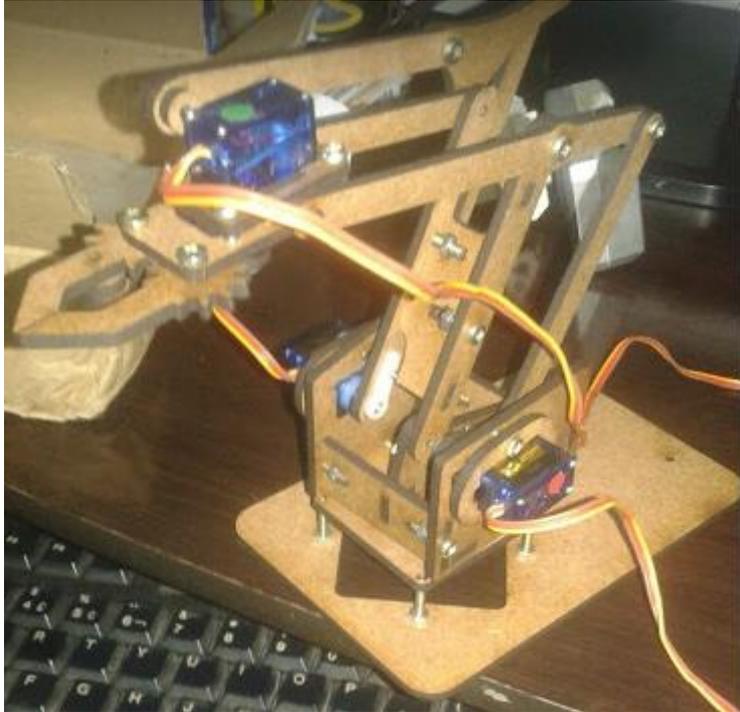


Figura 40 – Braço

O braço robótico e a esteira são fabricados em MDF e cortadas a laser, a esteira passou por algumas modificações para que a correia pudesse entrar em atrito com os tambores e roletes, a correia teve que ser remanejada para que o braço não encontrasse dificuldades para pegar os objetos na esteira, a seguir nas figuras 41, 42, 43 e 44 as imagens das modificações.



Figura 41 – Tambor modificado com lixa



Figura 42 – Roletes para levantar esteira



Figura 43 – Esteira antes da elevação da correia

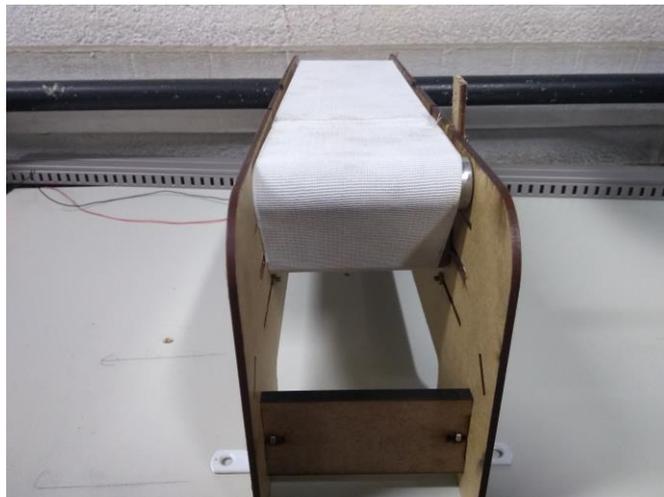


Figura 44 – Esteira com a elevação da correia

Usou – se a interface de potência para fazer com que a esteira pare quando a garrafa chegar ao sensor. Na figura 45 a imagem da interface.

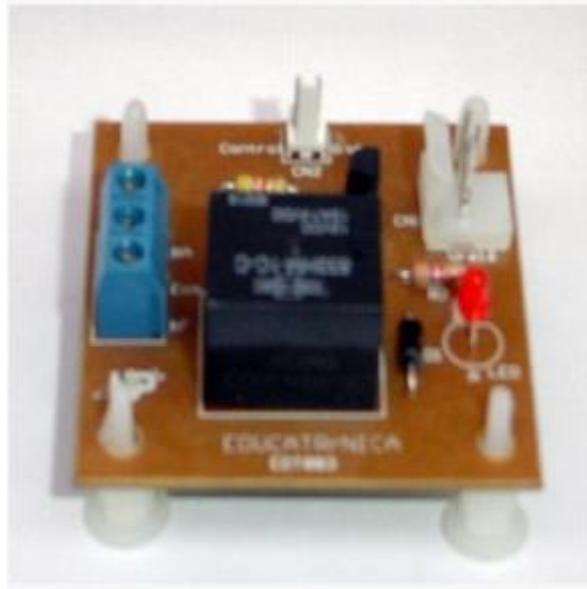


Figura 45 – Interface de potência

Na figura 46 a imagem do circuito elétrico da interface.

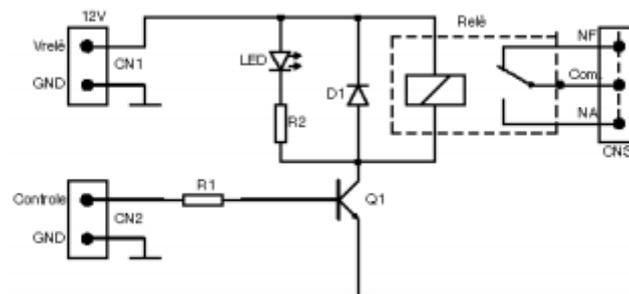


Figura 46 – Circuito da Interface de potência

Utilizou – sensor de obstáculo abaixo para indicar ao braço robótico que o objeto chegou ao final da esteira, como mostra a figura 47 o exemplo do sensor utilizado.

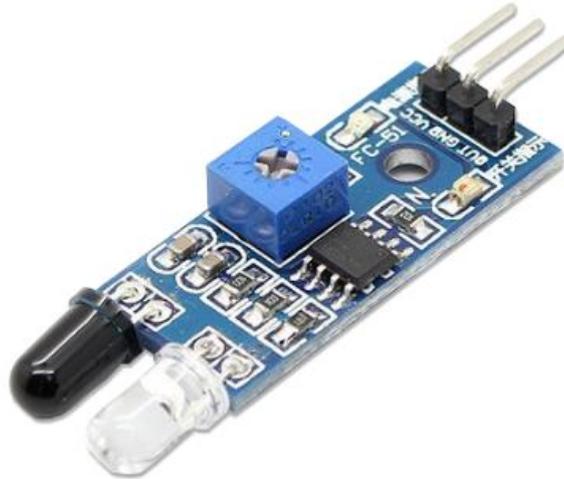


Figura 47 – Circuito da Interface de potência

Usamos a fonte ATX a seguir nas figuras 48 e 49 para alimentar todo o projeto a alimentação de 5V e corrente 25A.

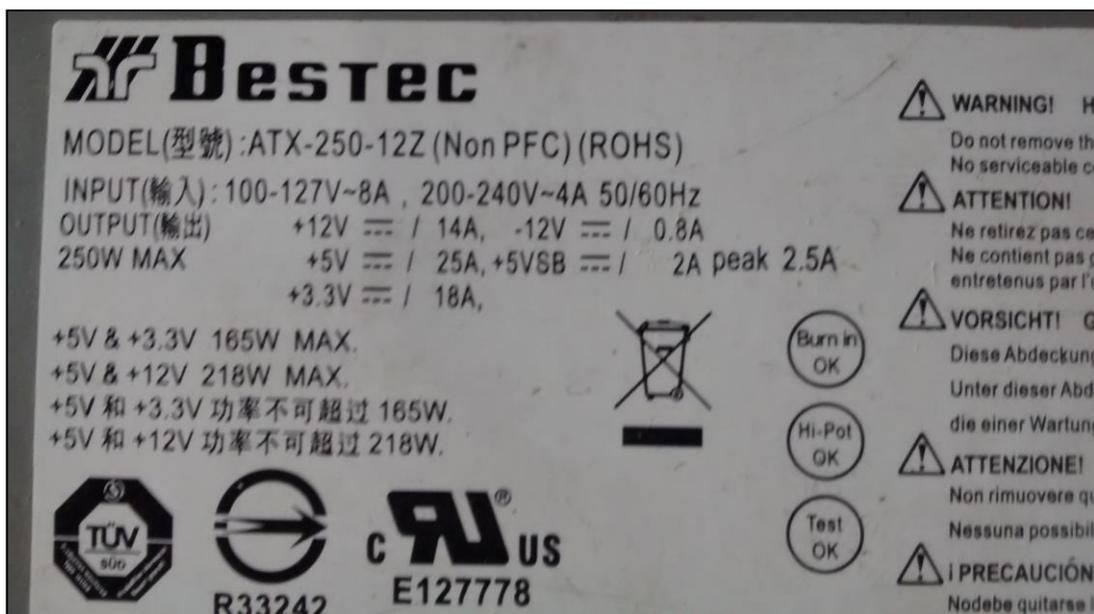


Figura 48 – Etiqueta com as informações da fonte



Figura 49 – Vista Traseira da fonte

Foi utilizado o site app inventor para desenvolver o aplicativo android de posicionamento do braço nas figuras 50 e 51 mostra o aplicativo desenvolvido para posicionar o braço, nas figuras 52 o script do aplicativo.

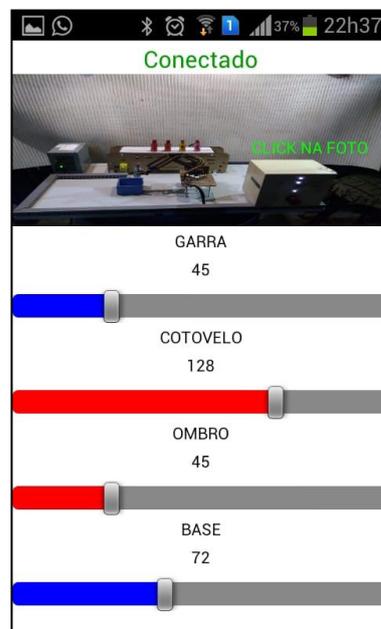


Figura 50 – Aplicativo para posicionamento do braço.



Figura 51 – Aplicativo para posicionamento do braço.

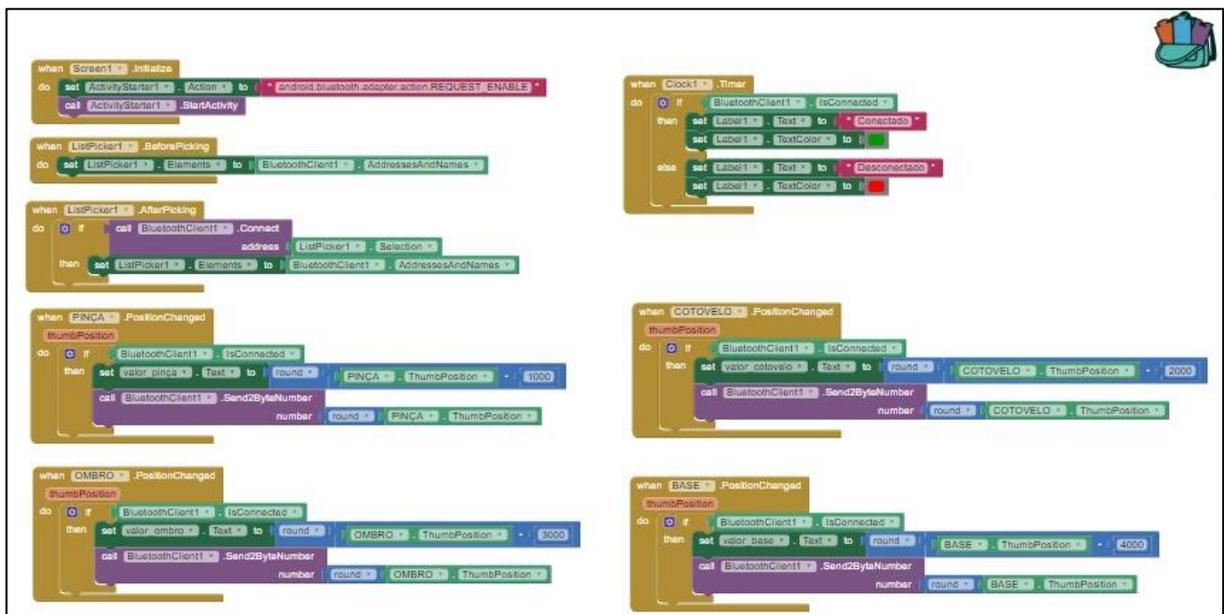


Figura 52 – Script do aplicativo

Nas Figuras 53, 54 as ligações do arduino com os servos-motores, motor da esteira e bluetooth.

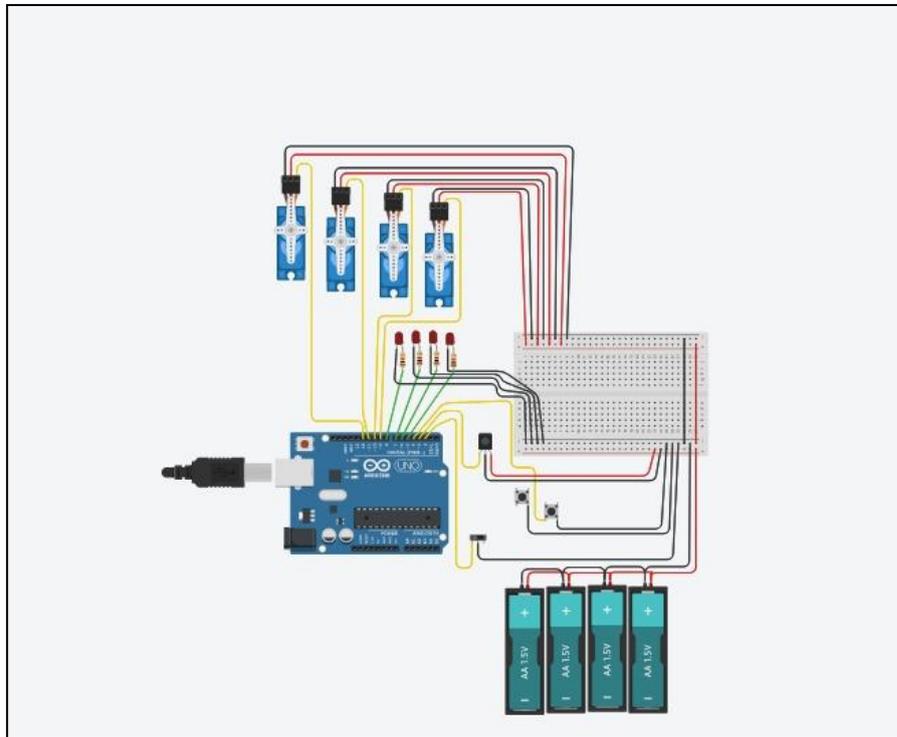


Figura 53 – Ligação dos servos-motores com arduino.

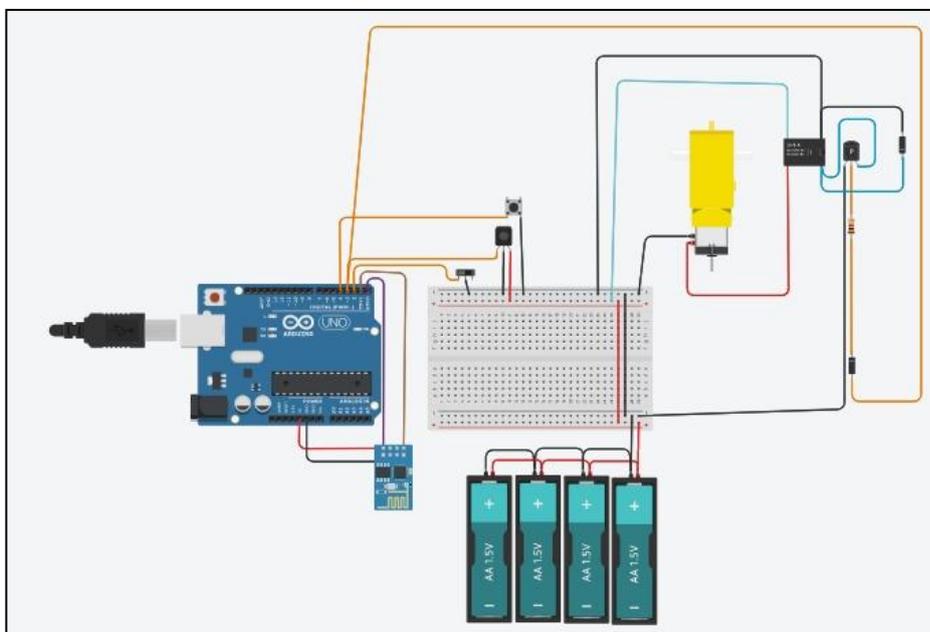


Figura 54 – Ligação do arduino com motor da esteira e bluetooth.

4 – Resultados Obtidos

O Projeto baseia – se em uma esteira para movimentação de peças utilizando um braço robótico para a transferência de peças para outra etapa do processo.

Originalmente, a esteira foi confeccionada para utilização com outros sensores, para que obtivéssemos o resultado satisfatório para o projeto foi modificada a posição da esteira confeccionado novos roletes, estes em alumínio, alterando a altura da correia para que a garra alcançasse melhor a garrafa.

Houve a necessidade de aquisição de uma fonte de alimentação para suprimir a deficiência do fornecimento da tensão, pois estava sendo alimentado pela saída USB do notebook. Na figura 55 a seguir o resultado final do projeto.

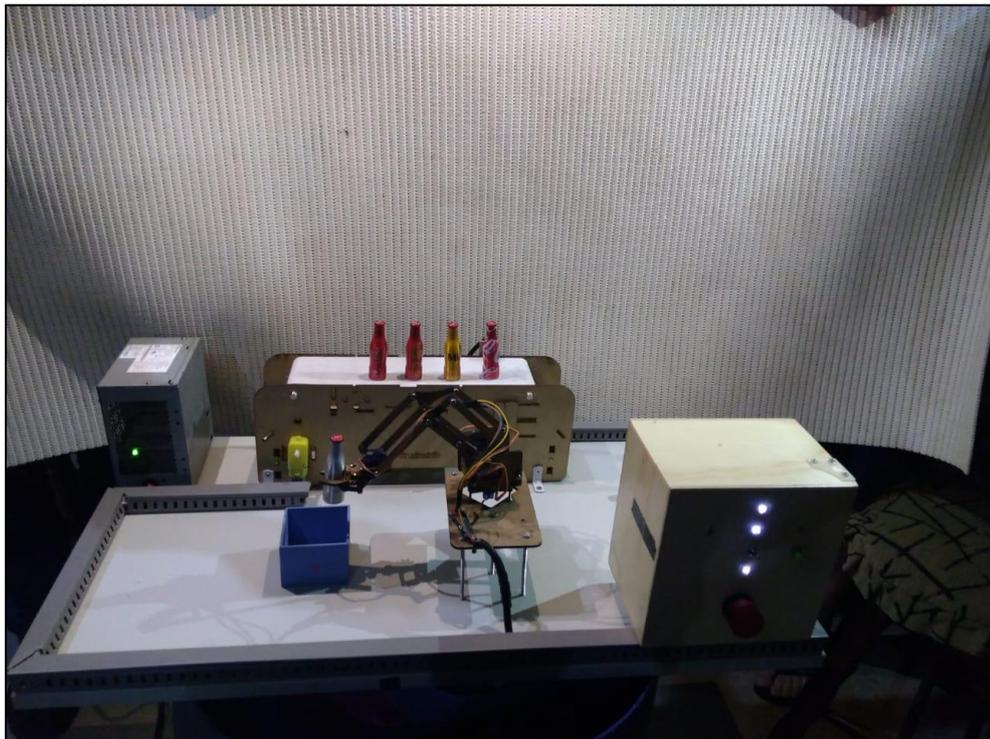


Figura 55 – Projeto final

Conclusão

O objetivo do projeto foi alcançado com êxito, o braço obedecendo de maneira satisfatória aos comandos do programa, a esteira funcionando de maneira satisfatória, sensor de obstáculo funcionando de maneira satisfatória indicado corretamente à existência de peça para o braço possa retirar a peça e levada para a próxima etapa do processo.

Referências

Cimautomacao. Braço Robótico Industrial. Disponível em: <<http://www.cimautomacao.com.br/lp/braco-robotico-industrial.php>> acesso em 27 de março de 2018 as 21:31

Edisciplinas usp. Robôs industriais. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/3265250/mod_resource/content/1/Aula_Rob%C3%B3tica_2017.pdf> acesso em 07 de Junho 2018 as 16:00

Pictronics. Servo – motor. Disponível em: <<http://www.pictronics.com.br/downloads/apostilas/servomotores.pdf>> acesso em 25 de fevereiro de 2018 as 22:00

Filipeflop. Arduino. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>> acesso em 30 de março as 11:00

Tech tudo. Bluetooth. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/01/bluetooth-o-que-e-e-como-funciona.html>> acesso em 20 de abril de 2018 as 15:00

Top componentes. Esteiras. Disponível em: <<http://www.topcomponentes.com.br/blog/2017/07/10/conheca-os-diferentes-tipos-de-esteiras-transportadoras-e-acessorios/>> acesso 05 de Maio de 2018 01:15

Apêndice A

Sketch do programa do arduino

```
[code]
#include <SoftwareSerial.h>

//int bluetoothTx = 52; // bluetooth tx to 52 pin
//int bluetoothRx = 50; // bluetooth rx to 50 pin
//SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

//Declaração das constantes

const int RELEesteira = 25; //constante led refere-se ao pino digital
const int Sensor_esteira = 2; //constante sensor de barreira
const int Botao_Ativa_Bluetooth = 3; // Botão ativa Bluetooth
const int led = 22; //Define que o pino 22 esta ativo braço em posição Automático
const int led1 = 23; //Define que o pino 23 esta ativo braço em posição Bluetooth
const int led2 = 24; //Define que o pino 24 ESTADO EM QUE O BRAÇO ROBOTICO
ESTA RETIRANDO E INDO GUARDA PEÇA
const int led3 = 26; //Define que o pino 26 ESTADO EM QUE A EMERGENCIA ESTA
ATIVA
const int led4 = 27; //Define que o pino 27 ESTADO EM QUE O ARDUINO ESTA
LIGADO
const int led5 = 28;
const int Botaostart = 7; // Botão start
const int BotaoEmergencia = 18; //Define que o pino 18 Botão de Emergência

//Variável que conterà os estados do botão (0 LOW, 1 HIGH).
int estadoSensor_esteira = 0;
int estadoBotao_Ativa_Bluetooth = 0;
int estadoBotaoEmergencia = 0;
int estadoBotaostart = 0;

// Bibliotecas auxiliares
#include <Servo.h>

// Definição de constantes
#define servoPinGarra1 12 //A
#define servoPinCotovelo1 11 //C
#define servoPinOmbro1 10 //D
#define servoPinBase1 9 //B

// Funções para movimento suave
```

```

void moveGarra1Fecha(const int tempoDelay, int fim, int inicio); //função que movimenta
o garra para fecha
void moveGarra1Abre(const int tempoDelay, int fim, int inicio); //função que movimenta o
garra para abrir
void moveCotovelo1Sobe(const int tempoDelay, int inicio, int fim); //função que movimenta
o cotovelo para cima
void moveCotovelo1Desce(const int tempoDelay, int inicio, int fim); //função que
movimenta o cotovelo para baixo
void moveOmbro1Adiantar(const int tempoDelay, int inicio, int fim); //função que
movimenta o ombro para frente
void moveOmbro1Retorna(const int tempoDelay, int inicio, int fim); //função que
movimenta o ombro para trás
void moveBase1Direita(const int tempoDelay, int inicio, int fim); //função que movimenta a
Base1 para direita (sentido horário)
void moveBase1Esquerda(const int tempoDelay, int inicio, int fim); //função que movimenta
a Base1 para esquerda (sentido antihorário)

```

```
// Funções de Manipulação
```

```

//void armOrigin(unsigned char Base1Inicio, unsigned char Base1Fim, unsigned char
Ombro1Inicio, unsigned char Ombro1Fim, unsigned char Cotovelo1Inicio, unsigned char
Cotovelo1Fim, unsigned char Garra1Int, unsigned char Garra1Fim); // unsigned char
Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char Ombro10Inicio, unsigned
char Ombro10Fim);
void armOrigin(unsigned char Base1Inicio, unsigned char Base1Fim, unsigned char
Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char Ombro1Inicio, unsigned char
Ombro1Fim, unsigned char Garra1Int, unsigned char Garra1Fim); // unsigned char
Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char Ombro10Inicio, unsigned
char Ombro10Fim);
void armOrigin2(unsigned char Base1Inicio, unsigned char Base1Fim, unsigned char
Ombro1Inicio, unsigned char Ombro1Fim, unsigned char Cotovelo1Inicio, unsigned char
Cotovelo1Fim); // unsigned char Cotovelo10Inicio,
unsigned char Cotovelo10Fim, unsigned char Ombro10Inicio, unsigned char Ombro10Fim);
void armOrigin3(unsigned char Base1Inicio, unsigned char Base1Fim, unsigned char
Ombro1Inicio, unsigned char Ombro1Fim, unsigned char Cotovelo1Inicio, unsigned char
Cotovelo1Fim, unsigned char Garra1Int, unsigned char Garra1Fim);
//void armOrigin4(unsigned char Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned
char Ombro1Inicio, unsigned char Ombro1Fim, unsigned char Base1Inicio, unsigned char
Base1Fim, unsigned char Garra1Int, unsigned char Garra1Fim); // unsigned char
Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char Ombro10Inicio, unsigned
char Ombro10Fim);
void armOrigin4(unsigned char Ombro1Inicio, unsigned char Ombro1Fim, unsigned char
Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char Base1Inicio, unsigned char
Base1Fim, unsigned char Garra1Int, unsigned char Garra1Fim);

```

```
// ##### //
```

```

// Atribuição de objetos (servos do braço)
Servo servo_Base1, servo_Ombro1, servo_Cotovelo1, servo_Garra1;

void setup()
{
  Serial.begin(9600);
  Serial.setTimeout(4);
  //Setup Bluetooth serial connection to android
  //bluetooth.begin(9600);

  pinMode(RELEesteira,OUTPUT); //Definindo pino digital 24 como de saída.
  pinMode(led,OUTPUT); //Define que o pino 22 esta ativo braço em posição automatico
  pinMode(led1,OUTPUT); //Define que o pino 23 esta ativo braço em posição Bluetooth
  pinMode(led2,OUTPUT); //Define que o pino 24 ESTADO EM QUE O BRAÇO
  ROBOTICO ESTA RETIRANDO E INDO GUARDA PEÇA
  pinMode(led3,OUTPUT); //Define que o pino 26 ESTADO EM QUE A EMERGENCIA
  ESTA ATIVA
  pinMode(led4,OUTPUT); //Define que o pino 27 ESTADO EM QUE O ARDUINO ESTA
  LIGADO
  pinMode(led5,OUTPUT);
  pinMode(Botaostart,INPUT_PULLUP); // Botão start
  pinMode(Sensor_esteira,INPUT_PULLUP); //Definindo pino digital 2 como de entrada.
  pinMode(Botao_Ativa_Bluetooth,INPUT_PULLUP); // Botão tira do modo automatico e
  ativa Bluetooth
  pinMode(BotaoEmergencia,INPUT_PULLUP); //Definindo pino digital 18 como de
  entrada.

  pinMode(servoPinGarra1, OUTPUT); //Define que o pino 12 para ativa Garra
  pinMode(servoPinCotovelo1, OUTPUT); //Define que o pino 11 para ativa Cotovelo
  pinMode(servoPinOmbro1, OUTPUT); //Define que o pino 10 para ativa Ombro
  pinMode(servoPinBase1, OUTPUT); //Define que o pino 09 para ativa Base

  servo_Garra1.attach(12); // Definido pinos do servo motor Garra
  servo_Cotovelo1.attach(11); // Definido pinos do servo motor Cotovelo
  servo_Ombro1.attach(10); // Definido pinos do servo motor Ombro
  servo_Base1.attach(9); // Definido pinos do servo motor Base

  digitalWrite(led4,HIGH); //Condição com led aceso - ARDUINO ENERGIZADO

  servo_Base1.write(10); //B // POSICIONAMENTO DOS SERVOS EM QUE NÃO
  TEM PEÇA PARA RETIRADA
  servo_Ombro1.write(45); //D
  servo_Garra1.write(49); //A
  servo_Cotovelo1.write(36); //C

```

```
} //Fim setup
```

```
// $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ LOOP INFINITO
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ //
```

```
void loop()
{
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
```

```
    if (estadoBotaoEmergencia == HIGH)
    {
```

```
        //digitalWrite(led5,LOW); //Condição com Led aceso assumido -
emergencia ativo
```

```
        digitalWrite(led3,LOW); //Condição com Led aceso assumido -
emergencia ativo
```

```
        //delay(200);
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
//_____LEITURA
```

```
BLUETOOTH_____
```

```
//_____
```

```
    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
    estadoBotao_Ativa_Bluetooth = digitalRead(Botao_Ativa_Bluetooth); //Condição
executada enquanto botão acionado e Led aceso
```

```

//if (estadoBotao_Ativa_Bluetooth == LOW && estadoBotaoEmergencia == HIGH)
if(estadoBotao_Ativa_Bluetooth == LOW)

{
  digitalWrite(led1,HIGH); //Condição com led aceso - ativo BLUETOOTH
  digitalWrite(led,LOW); //Condição com Led apagado assumida - desligado Automatco
do braço

//Leia de bluetooth e escreva para serial USB
if(Serial.available()>= 2)
//if(blueetooth.available()>= 2 )

{
  //unsigned int servopos = bluetooth.read();
  //unsigned int servopos1 = bluetooth.read();
  unsigned int servopos = Serial.read();
  unsigned int servopos1 = Serial.read();
  unsigned int realservo = (servopos1 *256) + servopos;
  //bluetooth.println(realservo);
  Serial.println(realservo);

if (realservo >= 1000 && realservo <1180) {
  int Garra1 = realservo;
  Garra1 = map(Garra1, 1000, 1180, 0, 180);
  servo_Garra1.write(Garra1);
  Serial.println("Servo 1 ON");
  delay(1);
}
if (realservo >= 2000 && realservo <2180) {
  int Cotovelo1 = realservo;
  Cotovelo1 = map(Cotovelo1, 2000, 2180, 0, 180);
  servo_Cotovelo1.write(Cotovelo1);
  Serial.println("Servo 2 ON");
  delay(1);
}
if (realservo >= 3000 && realservo <3180) {
  int Ombro1 = realservo;
  Ombro1 = map(Ombro1, 3000, 3180, 0, 180);
  servo_Ombro1.write(Ombro1);
  Serial.println("Servo 3 ON");
  delay(1);
}
if (realservo >= 4000 && realservo <4180) {
  int Base1 = realservo;
  Base1 = map(Base1, 4000, 4180, 0, 180);
  servo_Base1.write(Base1);
  Serial.println("Servo 4 ON");
  delay(1);
}

```

```

    }
  }
}

else{
  digitalWrite(led1,LOW); //Condição com led apagado - Desligado BLUETOOTH
  digitalWrite(led,HIGH); //Condição com Led aceso - Ativo modo Automático do braço

//_____LEITURA
BLUETOOTH_____
//_____
_____

//Lendo o estado do pino 2, constante Sensor_esteira, e atribuindo
//o resultado a variável estadoSensor_esteira.
estadoSensor_esteira = digitalRead(Sensor_esteira);
estadoBotao_Ativa_Bluetooth = digitalRead(estadoBotao_Ativa_Bluetooth);
estadoBotaoEmergencia = digitalRead(BotaoEmergencia);

if (estadoSensor_esteira == LOW && estadoBotaoEmergencia == HIGH)
//if (estadoSensor_esteira == LOW)
{
  delay(1000);

if (estadoBotao_Ativa_Bluetooth == HIGH && estadoBotaoEmergencia == HIGH)
//if (estadoBotao_Ativa_Bluetooth == HIGH)
{

  digitalWrite(led2,HIGH); //ESTADO EM QUE O BRAÇO ROBOTICO ESTA
RETIRANDO E INDO GUARDA PEÇA
  delay(1);

if (estadoSensor_esteira == LOW && estadoBotaoEmergencia == HIGH)
{
  // Objeto 01

servo_Garra1.write(31); // pega objeto // GARRA
servo_Cotovelo1.write(140);
delay(300);

```

```

//B/ /C/ /D/ /A/ // POSIÇÃO PARA RETIRADA A PEÇA
armOrigin(90,63, 140,94, 35,74 , 31,65); // Preencher Origem aqui (2ºs parametros)
delay(400);

```

```

digitalWrite(led2,LOW); //ESTADO EM QUE O BRAÇO ROBOTICO ESTA
RETIRANDO E INDO GUARDA PEÇA

```

```

//C/ /D/ /B/ // RECUANDO O BRAÇO DA POSIÇÃO EM QUE
RETIROU A PEÇA DA ESTEIRA
armOrigin2(94,140 ,74,35 , 63,175); //Preencher Destino aqui (O
1ºPARAMETRO IGUAL O 2ºPARAMETRO DA ARMORIGIN DE ACORDO COM AS
LETRAS)
delay(400);

```

```

//D/ /C/ /A/ // INDO PARA POCÃO DE GUARDA A PEÇA ("1")
armOrigin3 (35,90 , 140,58 , 65,20); //Preencher Destino aqui (O
1ºPARAMETRO IGUAL O 2ºPARAMETRO DA ARMORIGIN2 DE ACORDO COM AS
LETRAS)
delay(400);

```

```

// RECUANDO O BRAÇO DA POSIÇÃO EM QUE
GUARDOU A PEÇA ("1")
//D/ /C/ /B/ /A/ // Preencher Destino aqui (O 1ºPARAMETRO IGUAL O
2ºPARAMETRO DA ARMORIGIN3 DE ACORDO COM AS LETRAS)
armOrigin4 (90,25 ,58,140 , 175,90, 20,31); // LETRA B" 1ºPARAMETRO TEM QUE
SER IGUAL O 2ºPARAMETRO DA ARMORIGIN2)
delay(200);

```

```

#####
#####
#####
#####
}
} //Fim void loop

```

```

}else {
if (estadoSensor_esteira == HIGH && estadoBotao_Ativa_Bluetooth == HIGH)
{
//void pos_home(){

```



```
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@PONTO PARA
ATUA O SENSOR @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
//##### DESENVOLVIMENTO DAS FUNÇÕES DE MANIPULAÇÃO
##### //
```

```
void armOrigin(unsigned char Base1Inicio, unsigned char Base1Fim, unsigned char
Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char Ombro1Inicio, unsigned char
Ombro1Fim, unsigned char Garra1Int, unsigned char Garra1Fim) // unsigned char
Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char Ombro10Inicio, unsigned
char Ombro10Fim);
{
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
if (estadoBotaoEmergencia == HIGH)
{
delay(1);
```

```
//EMERGENCIA ACIMA
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
// Movimenta a Base1
```

```
if(Base1Inicio < Base1Fim)
{
moveBase1Direita(200,Base1Inicio,Base1Fim);
delay(11);
}
else
{
moveBase1Esquerda(200,Base1Inicio,Base1Fim);
delay(11);
}
```

```
//Fim movimentação da Base1

// Movimenta o cotovelo
if(Cotovelo1Inicio < Cotovelo1Fim)
{
    moveCotovelo1Sobe(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
else
{
    moveCotovelo1Desce(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
// Fim movimentação do cotovelo

//Movimenta o ombro
if(Ombro1Inicio < Ombro1Fim)
{
    moveOmbro1Adiantar(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
else
{
    moveOmbro1Retorna(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
// Fim movimentação do ombro

// Movimenta DA GARRA
if(Garra1Int < Garra1Fim)
{
    moveGarra1Abre(4,Garra1Int,Garra1Fim); //fecha garra
    delay(11);
}
else
{
    moveGarra1Fecha(4,Garra1Int,Garra1Fim); // abre garra
    delay(11);
}
// Fim movimentação DA GARRA
// break;
} //Fim armOrigin
}
```

```

//
#####
##### //
void armOrigin2(unsigned char Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char
Ombro1Inicio, unsigned char Ombro1Fim, unsigned char Base1Inicio, unsigned char
Base1Fim) // unsigned char Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char
Ombro10Inicio, unsigned char Ombro10Fim)
{

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == HIGH)
        {
            delay(1);

//EMERGENCIA ACIMA

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

// Movimenta o cotovelo
if(Cotovelo1Inicio < Cotovelo1Fim)
{
    moveCotovelo1Sobe(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
else
{
    moveCotovelo1Desce(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
// Fim movimentação do cotovelo

//Movimenta o ombro
if(Ombro1Inicio < Ombro1Fim)
{
    moveOmbro1Adiantar(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
else
{
    moveOmbro1Retorna(200,Ombro1Inicio,Ombro1Fim);

```

```

    delay(11);
  }
  // Fim movimentação do ombro

  // Movimenta a Base1
  if(Base1Inicio < Base1Fim)
  {
    moveBase1Direita(200,Base1Inicio,Base1Fim);
    delay(11);
  }
  else
  {
    moveBase1Esquerda(200,Base1Inicio,Base1Fim);
    delay(11);
  }
  //Fim movimentação da Base1
    // break;
} //Fim armOrigin
}

//
#####
##### //

void armOrigin3( unsigned char Ombro1Inicio, unsigned char Ombro1Fim, unsigned char
Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char Garra1Int, unsigned char
Garra1Fim) // unsigned char Cotovelo10Inicio, unsigned char Cotovelo10Fim, unsigned char
Ombro10Inicio, unsigned char Ombro10Fim)
{

  //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
  //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
    if (estadoBotaoEmergencia == HIGH)
    {
      delay(1);
    }

  //EMERGENCIA ACIMA

  //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
  //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

```

```

//Movimenta o ombro
if(Ombro1Inicio < Ombro1Fim)
{
    moveOmbro1Adiantar(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
else
{
    moveOmbro1Retorna(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
// Fim movimentação do ombro

// Movimenta o cotovelo
if(Cotovelo1Inicio < Cotovelo1Fim)
{
    moveCotovelo1Sobe(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
else
{
    moveCotovelo1Desce(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
// Fim movimentação do cotovelo

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

// Movimenta DA GARRA
if(Garra1Int < Garra1Fim)
{
    moveGarra1Abre(4,Garra1Int,Garra1Fim); //fecha garra
    delay(11);
}
else
{
    moveGarra1Fecha(4,Garra1Int,Garra1Fim); // abre garra
    delay(11);
}

// Fim movimentação DA GARRA
// break;
} //Fim armOrigin
}

```

```
void armOrigin4(unsigned char Ombro1Inicio, unsigned char Ombro1Fim, unsigned char
Cotovelo1Inicio, unsigned char Cotovelo1Fim, unsigned char Base1Inicio, unsigned char
Base1Fim, unsigned char Garra1Int, unsigned char Garra1Fim)
{
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
    if (estadoBotaoEmergencia == HIGH)
    {
        delay(1);
    }
```

```
//EMERGENCIA ACIMA
```

```
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO BOTÃO
EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
// Movimenta o ombro
if(Ombro1Inicio < Ombro1Fim)
{
    moveOmbro1Adiantar(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
else
{
    moveOmbro1Retorna(200,Ombro1Inicio,Ombro1Fim);
    delay(11);
}
// Fim movimentação do ombro
```

```
// Movimenta o cotovelo
if(Cotovelo1Inicio < Cotovelo1Fim)
{
    moveCotovelo1Sobe(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);
}
```

```

}
else
{
    moveCotovelo1Desce(200,Cotovelo1Inicio,Cotovelo1Fim);
    delay(11);

}
// Fim movimentação do cotovelo

// Movimenta a Base1
if(Base1Inicio < Base1Fim)
{
    moveBase1Direita(200,Base1Inicio,Base1Fim);
    delay(11);

}
else
{
    moveBase1Esquerda(200,Base1Inicio,Base1Fim);
    delay(11);

}
// Fim movimentação da Base1

// Movimenta DA GARRA
if(Garra1Int < Garra1Fim)
{
    moveGarra1Abre(4,Garra1Int,Garra1Fim); //fecha garra
    delay(11);

}
else
{
    moveGarra1Fecha(4,Garra1Int,Garra1Fim); // abre garra
    delay(11);

}
// Fim movimentação DA GARRA
// break;
} //Fim armOrigin4
}

//
#####
##### //

```

```
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
//##### DESENVOLVIMENTO DAS FUNÇÕES DE MOVIMENTAÇÃO
SUAVE ##### //
```

```
//#####$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
//#####$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
void moveGarra1Abre(const int tempoDelay, int inicio, int fim)
{
```

```
    // tempoDelay sugerido: 10ms (Servo de alto torque)
    for(int i=inicio; i<fim; i++)
    {
        servo_Garra1.write(i);
        delay(8);
```

```
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
        BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
        BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
        EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
            delay(0);
            break;
        }
        //EMERGENCIA ACIMA
```

```
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
        BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
        BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
    } //Fim for
```

```
} //Fim moveGarra1Abre
```

```

void moveGarra1Fecha(const int tempoDelay, int inicio, int fim)
{
  // tempoDelay sugerido: 10ms (Servo de alto torque)
  for(int i=inicio; i>fim; i--)
  {
    servo_Garra1.write(i);
    delay(4);

    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
    if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
    EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
    {
      delay(0);
      break;
    }
    //EMERGENCIA ACIMA

    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    } //Fim for
  //Fim moveGarra1Fecha
}

void moveCotovelo1Sobe(const int tempoDelay, int inicio, int fim)
{
  // tempoDelay sugerido: 10ms (Servo de alto torque)
  for(int i=inicio; i<fim; i++)
  {
    servo_Cotovelo1.write(i);
    delay(10);

    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
    //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
    BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    estadoBotaoEmergencia = digitalRead(BotaoEmergencia);

```

```

        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
            delay(0);
            break;
        }
        //EMERGENCIA ACIMA

        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    } //Fim for
} //Fim moveCotovelo1Sobe

void moveCotovelo1Desce(const int tempoDelay, int inicio, int fim)
{
    // tempoDelay sugerido: 10ms (Servo de alto torque)
    for(int i=inicio; i>fim; i--)
    {
        servo_Cotovelo1.write(i);
        delay(30);

        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
            delay(0);
            break;
        }
        //EMERGENCIA ACIMA

        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
        //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    } //Fim for
} //Fim moveCotovelo1Desce

void moveOmbro1Adiantar(const int tempoDelay, int inicio, int fim)
{

```

```

// tempoDelay sugerido: 10ms (Servo de alto torque)
for(int i=inicio; i<fim; i++)
{
servo_Ombro1.write(i);
delay(30);

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
                delay(0);
                break;
        }
        //EMERGENCIA ACIMA

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

} //Fim for
} //Fim moveOmbro1Adiantar

void moveOmbro1Retorna(const int tempoDelay, int inicio, int fim)
{
// tempoDelay sugerido: 10ms (Servo de alto torque)
for(int i=inicio; i>fim; i--)
{
servo_Ombro1.write(i);
delay(10);

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
                delay(0);
                break;
        }
}

```

```

//EMERGENCIA ACIMA

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

} //Fim for
} //Fim moveOmbro1Retorna

void moveBase1Direita(const int tempoDelay, int inicio, int fim)
{
//tempoDelay sugerido: 10ms (Servo de alto torque)
for(int i=inicio; i<fim; i++)
{
servo_Base1.write(i);
delay(30);

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
{
delay(0);
break;

}
//EMERGENCIA ACIMA

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

} //Fim for
} //Fim moveBase1Direita

void moveBase1Esquerda(const int tempoDelay, int inicio, int fim)
{
// tempoDelay sugerido: 10ms (Servo de alto torque)
for(int i=inicio; i>fim; i--)
{
servo_Base1.write(i);
delay(5);

```

```

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

        estadoBotaoEmergencia = digitalRead(BotaoEmergencia);
        if (estadoBotaoEmergencia == LOW)//QUANDO O BOTÃO
EMERGÊNCIA FOR APERTADO FAZ PARAR ESTE COMANDO DO (FOR)
        {
            // digitalWrite(led3,HIGH); //Condição com Led aceso assumido -
emergencia ativo

                delay(0);
                break;

        }
        //EMERGENCIA ACIMA

//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
//EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-LOCAL DO
BOTÃO EMERGÊNCIA-EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

    } //Fim for
} //Fim moveBase1Esquerda

//
##### //

//void pos_home(){

    //servo_Garra1.write(40); //A // POSICIONAMENTO DOS SERVOS EM QUE NÃO
TEM PEÇA PARA RETIRADA
    //servo_Base1.write(10); //B
    //servo_Ombro1.write(35); //D
    //servo_Cotovelo1.write(40); //C

//}
[/code]

```