

**CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA - UNIFACCAMP**  
**ENGENHARIA ELETRÔNICA**

**PARAMETRIZAÇÃO ESTEIRA LAGARTA**

**ALLANDERSON DA SILVA SANTOS**  
**RODRIGO LIMA CARVALHO**  
**WALLISON DE SOUSA FERNANDES**

**Campo limpo paulista - SP**  
**Junho – 2020**

**Allanderson da Silva Santos**  
**Rodrigo Lima Carvalho**  
**Wallison de Sousa Fernandes**

## **PARAMETRIZAÇÃO ESTEIRA LAGARTA**

*Trabalho de conclusão apresentado ao Centro Universitário Campo Limpo Paulista – UNIFACCAMP, como requisito para a obtenção do grau de Bacharel em Engenharia Eletrônica.*

**Orientador: Prof. Dr. Alexandre Martinewski**  
**Prof. Francisco Coelho de Oliveira**

**Campo limpo paulista - SP**  
**Junho – 2020**

**CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA - UNIFACCAMP  
ENGENHARIA ELETRÔNICA**

**PARAMETRIZAÇÃO ESTEIRA LAGARTA**

**RA 25710 Allanderson da Silva Santos**

**RA 25133 Rodrigo Lima Carvalho**

**RA 25402 Wallison de Sousa Fernandes**

**Orientador: Prof. Dr. Alexandre Martinewski**

**Banca Examinadora:**

---

**Prof.  
Convidado**

---

**Prof. Dr. Alexandre Martinewski  
Orientador**

---

**Prof. Dr. Alexandre Martinewski  
Coordenador**

**Campo limpo paulista - SP  
Junho – 2020**

## **DEDICATÓRIA**

Dedicamos aos nossos pais e amigos pelo apoio incondicional em todos os momentos difíceis da nossa trajetória acadêmica. Este trabalho é dedicado a eles e a todos os meus amigos de curso, grandes companheiros de jornada.

## AGRADECIMENTO

A Deus por ter nos dado saúde e força para superar as dificuldades.

A este Centro Universitário, o corpo docente, direção e administração por oportunizarem a janela que hoje vislumbramos um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Agradecemos a todos os *professores* por nos proporcionaram o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por se dedicarem a nós, não somente por terem nos ensinado, mas por terem feito aprender.

Aos nossos pais, pelo amor, pelos incentivos nas horas difíceis, de desânimo e cansaço.

E a todos, que direta e indiretamente fizeram parte da minha formação, o nosso muito obrigado.

## EPÍGRAFE

“A vida é um grande espetáculo, só não consegue homenageá-la quem nunca penetrou dentro de seu próprio e percebeu como é fantástica a construção da sua inteligência”

**Augusto Cury**

## RESUMO

Este trabalho descreve o desenvolvimento de um sistema de parametrização sem fio para uma esteira lagarta através da aquisição dos parâmetros de correntes, tensão, e temperatura em tempo real. Estas informações devem auxiliar o piloto durante os percursos. As informações coletadas através de sensores no robô são enviadas através da tecnologia Arduino, Bluetooth e Android. Arduino é um pequeno circuito microcontrolado, onde é colocado todos os componentes necessários para que este funcione e se comunique com o computador e partir do momento que o software é gravado no ARDUINO não precisa mais do computador: o ARDUINO funciona como um computador independente e conseguirá sozinho executar o sketch criado, desde que seja ligado a uma fonte de energia. Bluetooth permite a troca de dados e arquivos entre celulares, computadores, scanners, fones de ouvido e demais dispositivos de forma rápida e segura para um dispositivo com o sistema operacional Android um sistema operacional que utiliza-se uma tela sensível ao toque para que o usuário possa manipular objetos virtuais e também de um teclado virtual, analisando informações e apresenta-la de uma forma intuitiva ao piloto, sinalizando quaisquer problemas que estejam acontecendo. Os recursos do sistema foram amplamente testados em aplicações práticas, e os resultados corresponderam aos esperados no projeto original. Nosso projeto foi desenvolvido baseado nas referências bibliográficas e na tecnologia desenvolvida no ambiente de testes para a obtenção dos resultados finais.

**Palavras chaves:** Parametrização, esteira lagarta, robô, bluetooth, Android, sensores, arduino

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 Diagrama Esteira Lagarta          | 15 |
| Figura 2 Diagrama Ponte H                  | 21 |
| Figura 3 Modulo HC-06                      | 24 |
| Figura 4 Entrega da Esteira Lagarta        | 27 |
| Figura 5 Conferindo e separando as peças   | 28 |
| Figura 6 Montagem das rodas                | 28 |
| Figura 7 Instalação da Esteira nas rodas   | 29 |
| Figura 8 Esteira montada                   | 29 |
| Figura 9 Entrega do Braço Robótico         | 30 |
| Figura 10 Esquema eletrico Esteira Lagarta | 30 |
| Figura 11 Esquema eletrico Braço Robotico  | 31 |



# Sumario

|        |                                    |    |
|--------|------------------------------------|----|
| 1      | INTRODUÇÃO .....                   | 11 |
| 1.1.   | Objetivos .....                    | 12 |
| 1.1.1  | Objetivo geral.....                | 12 |
| 1.1.2  | Objetivos Especificos.....         | 12 |
| 1.2.   | Problema.....                      | 12 |
| 1.3.   | Metodologia.....                   | 13 |
| 1.3.1. | Metodologia Cientifica .....       | 13 |
| 1.3.2. | Metodologia do Projeto.....        | 13 |
| 1.3.3. | Organização do texto.....          | 14 |
| 2.     | REFERENCIAL TEORICO.....           | 15 |
| 2.1.   | Princípio da esteira lagarta ..... | 15 |
| 2.2.   | Esteira lagarta robotizada .....   | 16 |
| 2.2.1. | Componentes utilizados .....       | 16 |
| 2.3.   | Placa de Arduino uno .....         | 17 |
| 2.3.1. | Vantagem .....                     | 17 |
| 2.3.2. | Descrição da Placa.....            | 18 |
| 2.3.3. | Software do ARDUINO .....          | 18 |
| 2.3.4. | Hardware do ARDUINO.....           | 18 |
| 2.3.5. | Ciclo de programação.....          | 19 |
| 2.3.6. | IDE do ARDUINO .....               | 19 |
| 2.4.   | Ponte H L298N.....                 | 20 |
| 2.4.1. | Finalidade da ponte H L298N .....  | 20 |

|        |   |    |
|--------|---|----|
| 2.4.2. | Vantagem .....  | 20 |
| 2.4.3. | Especificações do chip L298N.....                                       | 20 |
| 2.4.4. | Descrição Detalhada da Ponte H L298N .....                              | 21 |
| 2.5.   | Modulo bluetooth hc-06.....   | 23 |
| 2.5.1. | Finalidade Modulo Bluetooth .....                                       | 23 |
| 2.5.2. | Modulo HC-06.....   | 24 |
| 2.5.3. | Especificação do modulo HC-06.....                                      | 24 |
| 2.6.   | Android.....  | 25 |
| 2.6.1. | Sistema Android .....   | 25 |
| 2.6.2. | Vantagens.....  | 25 |
| 3.     | MONTAGEM DA ESTEIRA LARGATA COM BRAÇO ROBOTICO .....                    | 27 |
| 3.1.   | Esteira Lagarta.....  | 27 |
| 3.2.   | Braço Robotico .....  | 30 |
| 3.3.   | Ligação dos componentes eletrônicos.....                                | 30 |
| 3.3.1. | Esquema elétrico da Esteira Lagarta via Bluetooth .....                 | 30 |
| 3.3.2. | Esquema elétrico do Braço Robotico controlado por Joystick .....        | 31 |
| 4.     | APRESENTAÇÃO DOS RESULTADOS.....  | 32 |
| 4.1.   | Aplicativo .....  | 32 |
| 4.2.   | Estrutura mecânica .....  | 32 |
| 4.3.   | Materiais.....  | 32 |
| 5.     | CONSIDERAÇÕES FINAIS .....  | 33 |
|        | REFERÊNCIAS BIBLIOGRÁFICAS.....   | 34 |
|        | APENDICE - SKETCH de programação da Esteira Lagarta via Bluetooth ..... | 35 |
|        | SKETCH de programação do Braço Robotico .....                           | 37 |

# 1 INTRODUÇÃO

É de conhecimento que para trabalhar na área de desenvolvimento de dispositivos robóticos são necessários muitos investimentos com o tempo, em pesquisas, em conhecimentos e experiência, em capital, e várias outras áreas, como mecânica, elétrica e computação. Por esse motivo, tem-se notado um aumento considerável no incentivo à participação e desenvolvimento na área de robótica (eletrônica) nos principais centros de estudos como universidades, escolas e até mesmo empresas, que estimulam seus funcionários a estudarem e realizarem implementações de soluções nesta área.

O intuito desse trabalho foi para desenvolver o monitoramento de parâmetros de extrema importância para o funcionamento do robô, que são as correntes elétricas dos motores e eletrônicas, as tensões de controle a operação e as temperaturas dos principais dispositivos que estão sendo utilizados. Estes parâmetros informam ao piloto do robô qual a situação interna do funcionamento do mesmo em tempo real.

Com esse conceito de monitoramento, os parâmetros citados anteriormente do robô são monitorados e enviados para um smartphone com sistema operacional Android, onde tornara disponível ao piloto essas informações de forma simples e direta.

Utilizando um dispositivo Arduino interligado com sensores de corrente, tensão, temperatura e um módulo adicional Bluetooth para a conexão remota sem fio deste sistema, é possível receber os dados relevantes a operação do robô diretamente no smartphone do piloto.

Todos os circuitos foram cuidadosamente implementados para a realização dos experimentos práticos em laboratório, e demonstraram grande eficácia na aquisição e transferência das informações para o piloto, que analisava os dados diretamente em smartphone.

## **1.1. Objetivos**

### **1.1.1 Objetivo geral**

Foi o desenvolvimento de um sistema de parametrização seguro para a aquisição de parâmetros de corrente, tensão e temperatura dos componentes essenciais de operação de um robô, visando aumentar consideravelmente a expectativa de durabilidade e funcionalidade do equipamento.

### **1.1.2 Objetivos Especificos**

- Desenvolver um sistema de coleta de informações e apresentação simplificada de conteúdo direcionado ao piloto e assistente em plataforma Android;
- Estudar a tecnologia Bluetooth para a utilização na comunicação de dados entre o robô, sistema de controle e apresentação dos parâmetros;
- Promover a comunicação automática dos dispositivos com o console que processa as informações e demonstra ao piloto e assistente.

## **1.2. Problema**

Os principais problemas apresentados com exceções de quebras mecânicas, são sinalizados previamente por sobrecorrentes, quedas de tensão e variações brusca de temperatura em pontos específicos. Com o acesso a estas informações em tempo real, o piloto pode tomar algumas decisões para poupar determinados recursos ou até prever quebras, como controlar determinadas manobras.

## **1.3. Metodologia**

### **1.3.1. Metodologia Científica**

O desenvolvimento do trabalho foi definido que o sistema de telemetria seria composto basicamente por cinco partes, os sensores que se encarregam de obter as informações passantes pelo circuito, o modulo Arduino onde é colocado todos os componentes necessários para que este funcione e se comunique com o computador e partir do momento que o software é gravado no Arduino não precisa mais do computador, o Bluetooth que tem a função de transmitir as informações para um terminal remoto, o smarthphone como receptor do sinal, que também possui seu modulo Bluetooth que irá parrear com o modulo da interface de aquisição, e o aplicativo desenvolvido para Android que possui entre outras a função de analisar e armazenar as informações recebidas, tornando-as dados de fácil compreensão e consistentes para o piloto do robô.

### **1.3.2. Metodologia do Projeto**

Como o projeto trata na sua maior parte sobre o desenvolvimento físico de um hardware, com implementação do circuito que analisa os parâmetros importantes definidos pelo projeto, foi desenvolvido uma metodologia baseada em as etapas para contextualizar a problematização e fornece uma base tecnológica para o desenvolvimento do projeto.

Em um segundo momento, foram adquiridos todos os recursos de hardware para execução do projeto e o desenvolvimento do sistema de telemetria. Juntamente com essa etapa também foi realizado o desenvolvimento do aplicativo para armazenamento e visualização dos dados recebidos pelo smarthphone, que fornece condições em tempo real ao piloto para serem tomadas decisões mais assertivas em relação ao comportamento do robô.

### **1.3.3. Organização do texto**

O documento está dividido em 4 capítulos. No primeiro capítulo apresenta-se o processo do projeto, expondo uma breve contextualização e apresentando a problemática vislumbrada, assim como os componentes, objetivo geral e específicos.

No segundo capítulo é realizada uma revisão sobre a Esteira e o Braço Robótico promovendo um maior detalhamento do processo de montagem, esquemas elétricos e programações.

O terceiro capítulo apresenta e discute os resultados obtidos assim como a estrutura mecânica, materiais e aplicativo utilizado considerando a proposição do trabalho.

Por fim, o sexto capítulo contém as considerações finais.

## 2. REFERENCIAL TEORICO

### 2.1. Princípio da esteira lagarta

A lagarta ou rastro consiste num esteira que se acopla às rodas de certos veículos terrestres com a finalidade de lhes aumentar a aderência ao solo e, conseqüentemente, a sua capacidade de tração em terreno difícil.

As lagartas são movidas por rodas motrizes dentadas, impulsionadas pelo motor e engatadas em aberturas ou cavilhas existentes nos elos da lagarta. Tipicamente, a roda motriz é montada bastante acima da área de contato com o solo, permitindo-lhe ocupar uma posição fixa. Uma roda não motriz é colocada no extremo oposto da lagarta, primariamente para lhe dar uma inclinação que lhe permita superar obstáculos, mas também para lhe dar uma tensão adequada.

Este sistema impede os veículos a lagartas de se afundarem no solo, em terrenos onde veículos de rodas convencionais ficariam atolados. Exemplo na figura 1.

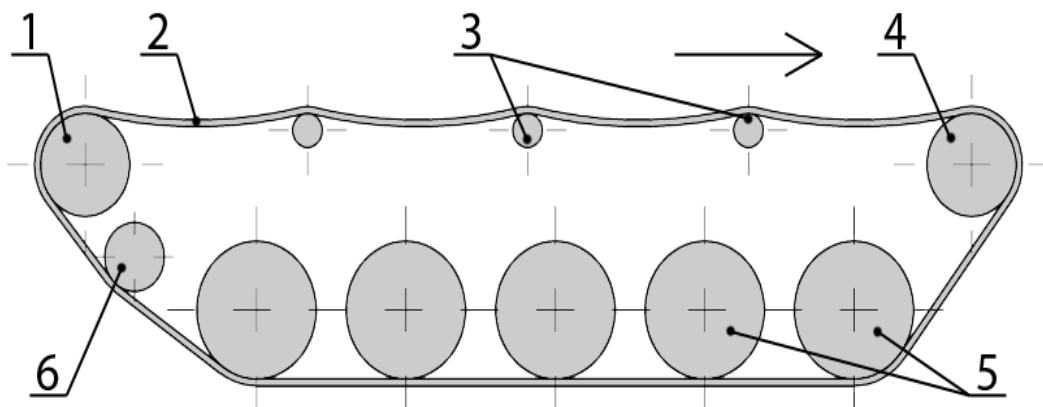


Figura 1 - Diagrama Esteira Lagarta

## Diagrama da suspensão da esteira

- (1) Roda motriz traseira,
- (2) Lagarta,
- (3) Roletes de apoio superior,
- (4) Roda motriz dianteira,
- (5) Roda de apoio inferior,
- (6) Roda tensora,

## 2.2. Esteira lagarta robotizada

### 2.2.1. Componentes utilizados

- Placa de Arduino Uno
- Ponte H L298N
- Jumpers
- Bateria 9v
- Bateria 12v
- Modulo Bluetooth HC-06
- Sistema Android
- Botão Joystick
- Motores DC 9v
- Motores DC 5v



## 2.3. Placa de Arduino uno

ARDUINO é uma plataforma flexível open-source de hardware e software para prototipagem eletrônica. ARDUINO pode receber entradas de uma variedade de sensores e pode atuar o ambiente ao seu redor por controladores de iluminação, motores e uma infinidade de atuadores.

O microcontrolador presente na placa é programado usando a linguagem de programação do ARDUINO (baseada em *wiring*) e o ambiente de desenvolvimento ARDUINO (baseada em *processing*). Os projetos com ARDUINO podem ser STANDALONE ou comunicar com um software rodando no PC.

### 2.3.1. Vantagem

- É um ambiente multiplataforma, podendo ser executado em Windows, Macintosh e Linux;
- Tem por base um ambiente de fácil utilização baseado em processing;
- Pode ser programado utilizando um cabo de comunicação USB onde geralmente não é necessária uma fonte de alimentação;
- Possui hardware e software open-source, facilitando a montagem do seu próprio hardware sem precisar pagar nada aos criadores originais;
- Hardware de baixo custo;
- Grande comunidade ativa de usuários;
- Ambiente educacional, ideal para iniciantes que desejam resultados rápidos.

### 2.3.2. Descrição da Placa

- **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 0 - 5):** 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 5 V 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 ser 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 volts com pino redondo de 2,1 mm e centro positivo.

### 2.3.3. Software do ARDUINO

O software para programação do ARDUINO é uma IDE que permite a criação de Sketches para a placa ARDUINO. A linguagem de programação é modelada a partir da linguagem *processing*. Quando pressionado o botão upload da IDE, o código escrito é traduzido para a linguagem C e é transmitido para o compilador *avr-gcc*, que realiza a tradução dos comandos para uma linguagem que pode ser compreendida pelo microcontrolador.

### 2.3.4. Hardware do ARDUINO

A placa do ARDUINO é um pequeno circuito microcontrolado, onde é colocado todos os componentes necessários para que este funcione e se comunique com o computador.

### 2.3.5. Ciclo de programação

O Ciclo de programação do ARDUINO pode ser dividido da seguinte maneira:

1. Conexão da placa a uma porta USB do computador;
2. Desenvolvimento de um sketch com comando para a placa;
3. Upload do sketch para a placa, utilizando a comunicação USB.
4. Aguardar a reinicialização, após ocorrerá à execução do sketch criado.

A partir do momento que o software é gravado no ARDUINO não precisa mais do computador: o ARDUINO funciona como um computador independente e conseguirá sozinho executar o sketch criado, desde que seja ligado a uma fonte de energia.

### 2.3.6. IDE do ARDUINO

O IDE é dividido em três partes: A Toolbar no topo, o código ou a Sketch Window no centro, e a janela de mensagens na base. Sob a Toolbar há uma guia, ou um conjunto de guias, com o nome do arquivo do sketch. Também há um botão posicionado no lado direito que habilita o serial monitor.

Ao longo do topo há uma barra de menus, com os itens File, Edit, Sketch, Tools e Help. Os botões na Toolbar fornecem acesso conveniente às funções mais utilizadas dentro desses menus.

Identificação dos ícones de atalho da IDE:

- **Verify:** Verifica se existe erro no código digitado.
- **Upload:** Compila o código e grava na placa Arduino se corretamente conectada;

- **New:** Cria um novo sketch em branco.
- **Open:** Abre um sketch, presente no sketchbook.
- **Save:** Salva o sketch ativo
- **Seria monitor:** Abre o monitor serial.

Os demais comandos presentes na barra de menus podem ser consultados através do help do IDE.

## **2.4. Ponte H L298N**

### **2.4.1. Finalidade da ponte H L298N**

O uso de Ponte H é interessante para diversas implementações com motores, porque estes componentes permitem uma série de facilidades. Então, o primeiro ponto a citar é que a ponte H nos permite girar o motor para ambos os lados.

### **2.4.2. Vantagem**

Uma das principais vantagens de se ter um circuito assim está associado a ele permitir uma maior dissipação de potência em relação ao acionamento direto por portas de circuitos microprocessados, que só conseguem fornecer alguns mA de corrente, contra 2A por canal da ponte H L298N.

### **2.4.3. Especificações do chip L298N**

- Tensão de Operação: 4.5 V~46 V
- Controle de 2 motores DC ou 1 motor de passo

- Corrente de Operação máxima: 2 A por canal ou 4 A total.
- Tensão lógica: 4.5 V ~7 V
- Corrente lógica: 0~36 mA
- Limites de Temperatura: -20 a +135 °C
- Potência Máxima: 25 W
- Dimensões: 43 x 43 x 27 mm.

#### 2.4.4. Descrição Detalhada da Ponte H L298N

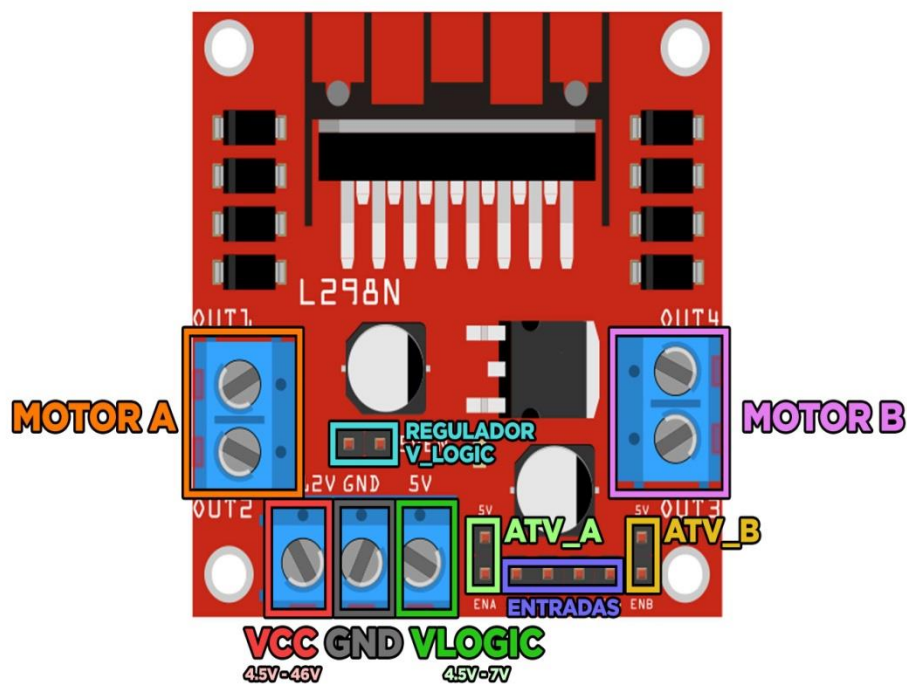


Figura 2- Diagrama Ponte H

#### ▪ Vcc:

Este borne é responsável pela alimentação dos motores, com isso a tensão colocada neste borne de Vcc será copiada para os motores. Por isso, muita atenção: Se você possui um motor que opera somente

até 12V, por exemplo, não alimente Vcc com mais de 12V, pois danificará o motor.

- **GND:**

É o GND da placa, que deverá ser o mesmo GND do Arduino ou de outro microcontrolador. Por isso, é importante interligar os GNDs sempre com fios. Serão demonstrados na seção de exemplos.

- **Vlogic:**

Este borne é responsável pela alimentação lógica da ponte H. O chip L298N presente na placa recebe comandos do Arduino, e para ele conseguir operar os comandos e atuar na saída ele precisa de operar como um componente lógico (como o Arduino), e para isso precisa atuar em uma faixa bem restrita de tensão, que no caso é 4.5V a 7V. Como foi dito a placa possui um regulador de tensão de 5V, que a função dele é justamente fazer a regulação de Vcc para a alimentação lógica do chip em 5V. Por isso, se o regulador estiver ativo, este pino Vlogic terá a tensão de 5V, pois, é a tensão presente na saída do regulador.

- **Regulador Vlogic:**

Este jumper é responsável por ativar ou desativar o regulador de tensão da alimentação de Vlogic. Quando o jumper está encaixado entre os terminais, o regulador de tensão está ativado e a Vlogic como a ser regulado em 5V a partir da tensão de Vcc. Por isso, é muito importante saber manusear esse jumper do regulador.

- **Atv\_A:**

Este jumper é responsável por desativar ou ativar o motor A para ser controlado pelo Arduino ou outra placa.

- **Atv\_B:**

Este jumper é responsável por desativar ou ativar o motor B para ser controlado. O funcionamento é exatamente idêntico ao jumper Atv\_A.

- **Entradas:**

As entradas de controle da placa possuem os pinos IN1, IN2, IN3 e IN4, sendo que a entrada IN1 e IN2 são as entradas que controlam os bornes do motor A, e as entradas IN3 e IN4 controlam os bornes do Motor B.

- **Motor A e Motor B:**

Estes bornes são de fato onde os motores DC são conectados. A posição dos fios do motor na conexão com o borne altera o sentido de rotação do motor. Se for invertido os fios no borne, o sentido de rotação também é invertido no acionamento.

## **2.5. Modulo bluetooth hc-06**

### **2.5.1. Finalidade Modulo Bluetooth**

Bluetooth é um protocolo padrão de comunicação primariamente projetado para baixo consumo de energia com baixo alcance, (dependendo da potência: 1 metro, 10 metros, 100 metros) baseado em microchips transmissores de baixo custo em cada dispositivo.

O Bluetooth possibilita a comunicação desses dispositivos uns com os outros quando estão dentro do raio de alcance. Os dispositivos usam um sistema de comunicação via rádio, por isso não necessitam estar na linha de visão um do

outro, e podem estar até em outros ambientes, contanto que a transmissão recebida seja suficientemente potente.

### 2.5.2. Módulo HC-06

O módulo Bluetooth HC-06, dispõe de apenas 4 pinos de entrada e saída de dados, sendo dois deles para alimentação do sistema, e os outros dois para transmissão e recepção de dados via **serial**. Logo abaixo temos a especificação do dispositivo:



Figura 3 - Módulo HC-06

### 2.5.3. Especificação do módulo HC-06

- Bluetooth: 2.0V
- Tensão de funcionamento: 3.3v~5v
- Taxa de transmissão: 2Mbps
- Frequência: 2,4 Ghz
- Nível lógico: 3.3v



- Pinos: VCC, GND, TXD, RXD;
- Perfis suportados: Escravo (slave) e Mestre (master)

## **2.6. Android**

### **2.6.1. Sistema Android**

Android é o sistema operacional móvel do Google. Presente em múltiplos aparelhos de diversas fabricantes, como Samsung, Motorola, LG, e Sony, é a plataforma mobile mais popular do mundo. É conhecido por ser baseado no núcleo do Linux, ter um código aberto e uma série de possibilidades de personalização.

O sistema nasceu em 2008, e desde então vem sendo aperfeiçoado pelo Google com diversas novidades. Suas atualizações, que não ocorrem com periodicidade exata, recebem sempre nomes de decas.

Um aspecto a ser ressaltado é que celulares não são os únicos a serem equipados com Android. Tablets, consoles móveis, e agora uma série de novos produtos, como TVs, computadores e dispositivos vestíveis, já estão sendo fabricados com ele. Esta fragmentação é uma das grandes vantagens desse sistema operacional para sua popularização, mas também uma desvantagem pelo fato de ele não ser unificado.

O sistema é uma das grandes armas do Google nos últimos anos, pois é extremamente popular, ganhou público tanto com os modelos mais simples como com os tops de linha, e não há dúvida de que deve manter-se no topo ainda por um bom tempo.

### **2.6.2. Vantagens**

- Sistema operacional aberto;
- Funciona de forma leve junto aos aparelhos com mais hardware;

- Possui touch sensível em análise a outros aparelhos antigos;
- Aplicativos grátis na Android Market;
- Conexão a redes sociais;
- Conexão a internet;
- Aparelhos mais baratos nas operadoras;

### 3. MONTAGEM DA ESTEIRA LARGATA COM BRAÇO ROBOTICO

#### 3.1. Esteira Lagarta



Figura 4 - Entrega da Esteira Lagarta



Figura 5 - Conferindo e separando as peças

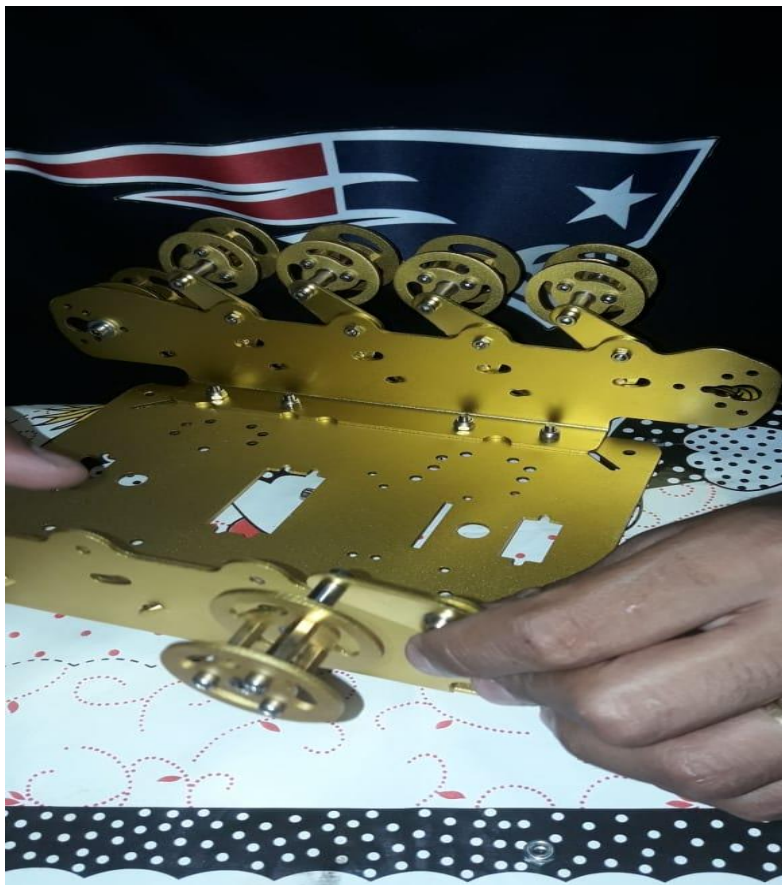
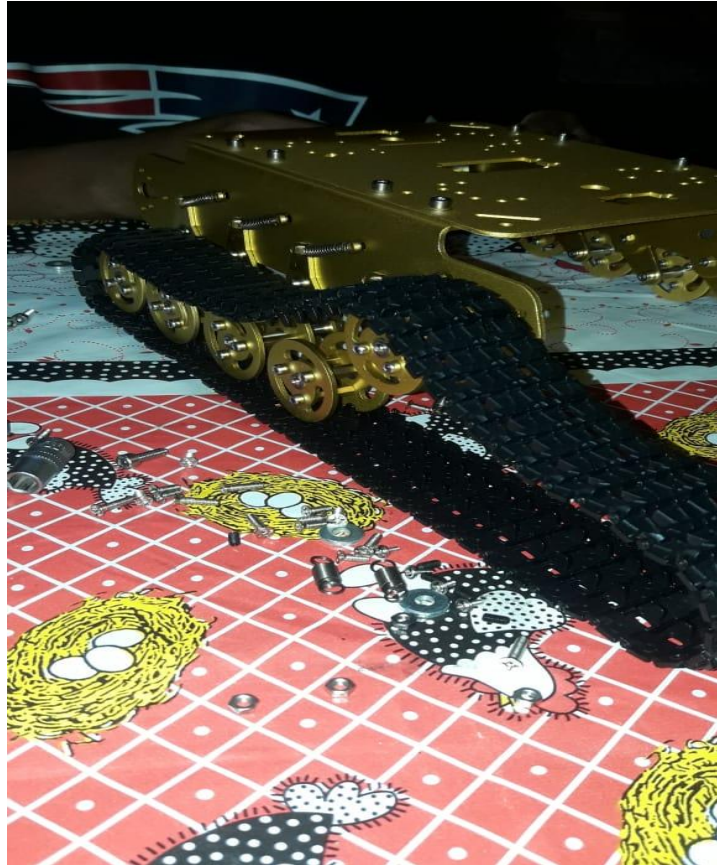


Figura 6 - Montagem das rodas



**Figura 7 - Instalação da Esteira nas rodas**



**Figura 8 - Esteira montada**

### 3.2. Braço Robótico



Figura 9 - Entrega do Braço Robótico

### 3.3. Ligação dos componentes eletrônicos

#### 3.3.1. Esquema elétrico da Esteira Lagarta via Bluetooth

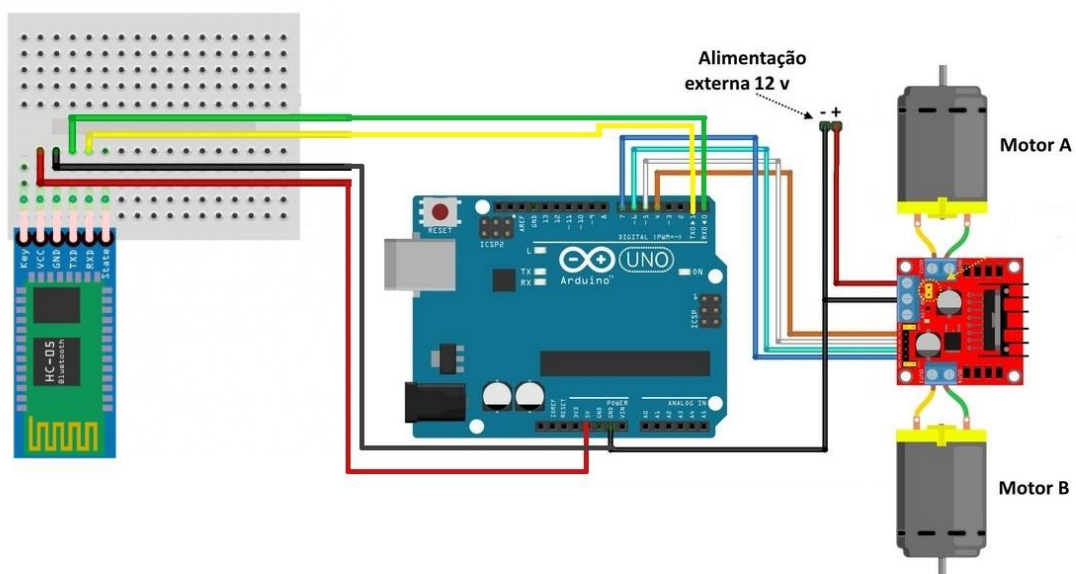


Figura 10 - Esquema elétrico Esteira Lagarta

### 3.3.2. Esquema elétrico do Braço Robótico controlado por Joystick

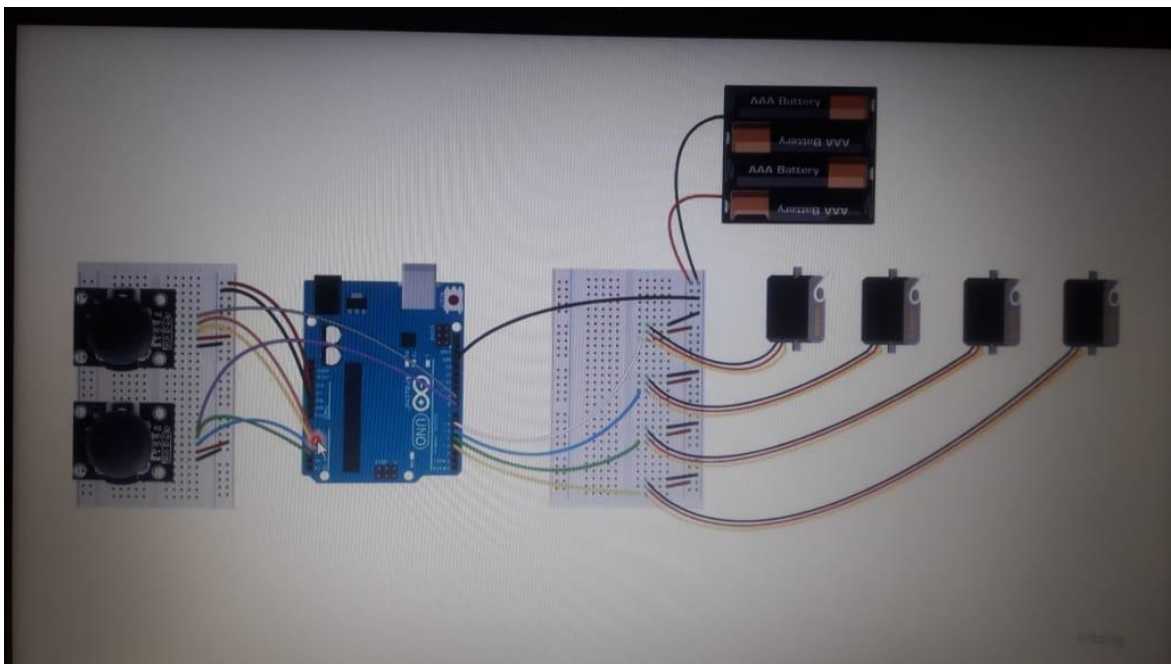


Figura 11 - Esquema elétrico Braço Robótico

## **4. APRESENTAÇÃO DOS RESULTADOS**

### **4.1. Aplicativo**

O projeto foi totalmente desenvolvido e funciona perfeitamente. A Conexão Bluetooth e o envio de dados serial operam corretamente desde que seja respeitado o limite de distância de 10 metros (limite aceitável da classe 2).

### **4.2. Estrutura mecânica**

Ao término da montagem estrutural e ao realizar o teste de movimentação da Esteira, foi observado que um dos motores da esteira estava travado e necessitava de um reparo. O problema foi sanado com a substituição de dois novos motores, sendo assim não tivemos, mas problemas com a movimentação da mesma.

### **4.3. Materiais**

Os equipamentos dimensionados neste projeto atenderam as necessidades esperadas ao longo de seu desenvolvimento. A bateria de alimentação suportou a corrente exigida pelos servos motores sem apresentar falhas ou superaquecimento. O modulo HC-06 atendeu nossos objetivos pois permitiu que adicionar o controle remoto para o Arduino, sendo assim um projeto mais tecnológico (Eliminando Joystick e Jumpers).



## 5. CONSIDERAÇÕES FINAIS

Este trabalho pode ser dividido em três partes: Aplicativo, Estrutura mecânica e Controlador.

O aplicativo denominado foi desenvolvido a partir de exemplos básicos fornecidos pela própria ferramenta.

A Estrutura mecânica proporcionou, ao longo do seu desenvolvimento, uma grande oportunidade de aprendizado, pois está correlacionado a uma área até então pouco conhecimento específico, na qual foi aprimorado durante as pesquisas para que se tornasse possível o desenvolvimento do mesmo. Esta estrutura pode ser melhorada em termos de robustez caso seja necessário que ela suporte uma carga maior.

Outro ponto que se tornou uma grande fonte de novos conhecimentos foi a programação do Arduino, pois as noções de programação na linguagem requerida pelos dispositivos eram limitadas, no qual exigiu um esforço maior para se desenvolver a comunicação entre o aplicativo. Pode-se observar que esta placa de desenvolvimento oferece as mais variadas opções para o programador, dependendo de sua criatividade.

Este projeto oferece diversas oportunidades para continuação de seu desenvolvimento como, por exemplo, a realização de seu controle via internet no qual o usuário pode controlar em qualquer lugar, desde que consiga ter acesso a mesma. Outra aplicação é o acoplamento de câmeras em uma base móvel onde este pode ser utilizado em trabalhos em risco a integridade física de seu operador.

## REFERÊNCIAS BIBLIOGRÁFICAS

<https://www.brincandocomideias.com/robotica>

<https://www.youtube.com/watch?v=yyCX9oP2cso>

<https://www.usinainfo.com.br/blog/braco-robotico-arduino-com-servo-motor-e-joystick/>

[https://pt.wikipedia.org/wiki/Lagarta\\_\(mec%C3%A2nica\)](https://pt.wikipedia.org/wiki/Lagarta_(mec%C3%A2nica))

## APENDICE - SKETCH de programação da Esteira Lagarta via Bluetooth

```
// Code by TCC Engenharia Eletronica
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial bluetooth(0, 1);
```

```
int IN1 = 4;
```

```
int IN2 = 5;
```

```
int IN3 = 6;
```

```
int IN4 = 7;
```

```
char comando;
```

```
void setup() {
```

```
  //Define os pinos como saida
```

```
  bluetooth.begin(9600);
```

```
  pinMode(IN1, OUTPUT);
```

```
  pinMode(IN2, OUTPUT);
```

```
  pinMode(IN3, OUTPUT);
```

```
  pinMode(IN4, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  while (bluetooth.available()) {
```

```
    comando = bluetooth.read();
```

```
    if (comando == 'F') {
```

```
      frente();
```

```
    }
```

```
    else if (comando == 'B') {
```

```
      tras();
```

```
    }
```

```
else if (comando == 'L') {
    esquerda();
}
else if (comando == 'R') {
    direita();
} else {
    parado();
}
}
}
```

```
void frente() {
    //Gira o Motor A e B no sentido horario
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
```

```
void tras() {
    //Gira o Motor A e B no sentido anti-horario
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}
```

```
void esquerda() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
```

```
void direita() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
}
```

```
void parado() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, HIGH);  
}
```

### **SKETCH de programação do Braço Robotico**

```
#include <Servo.h>
```

```
#define servoBase      3 // Porta Digital do Servo da Base
```

```
#define servoAlturaBraco 4 // Porta Digital do Servo da Altura do Braço
```

```
#define servoAnguloBraco 5 // Porta Digital do Servo do Ângulo do Braço
```

```
#define servoGarra     6 // Porta Digital do Servo da Garra
```

```
#define potBase        A0 // Porta Analógica do Potenciometro para Controle da  
Base
```

```
#define potAlturaBraco A1 // Porta Analógica do Potenciometro para Controle da  
Altura do Braço
```

```
#define potAnguloBraco A2 // Porta Analógica do Potenciometro para Controle do  
Ângulo do Braço
```

```
#define potGarra       A3 // Porta Analógica do Potenciometro para Controle da  
Garra.
```

```
#define botaoCongela   8 // Porta Digital do Botao Congela
```

```
#define botaoDescongela 9 // Porta Digital do Botao Descongela
```

```

//Instanciação dos Objetos de Controle dos Servos
Servo base;
Servo altura;
Servo angulo;
Servo garra;

//Variaveis para cálculo dos angulos
int valorPotBase;
int valorAngBase;
int valorPotAltura;
int valorAngAltura;
int valorPotAngulo;
int valorAngAngulo;
int valorPotGarra;
int valorAngGarra;

//Variaveis para controle dos botões
int estadoBotaoC, estadoBotaoD;
int estadoAntBotaoC = HIGH;
int estadoAntBotaoD = HIGH;
boolean congelado = false;

void setup(){
  pinMode(botaoCongela, INPUT_PULLUP);
  pinMode(botaoDescongela, INPUT_PULLUP);

  //Usa o LED do Arduino para informar estado de congelado
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);

  //Configuração das Portas dos Servos
  base.attach(servoBase);
  altura.attach(servoAlturaBraco);

```

```

angulo.attach(servoAnguloBraco);
garra.attach(servoGarra);
}

void loop() {

  if (!congelado) {
    valorPotBase = analogRead(potBase);           //Leitura do Angulo do
Potenciometro
    valorAngBase = map(valorPotBase, 0, 1024, 0, 180); //Conversão do valor do
potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    base.write(valorAngBase);                     //Envio do angulo para o Servo

    valorPotAltura = analogRead(potAlturaBraco);   //Leitura do Angulo do
Potenciometro
    valorAngAltura = map(valorPotAltura, 0, 1024, 180, 0); //Conversão do valor do
potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    altura.write(valorAngAltura);                 //Envio do angulo para o Servo

    valorPotAngulo = analogRead(potAnguloBraco);   //Leitura do Angulo do
Potenciometro
    valorAngAngulo = map(valorPotAngulo, 0, 1024, 180, 0); //Conversão do valor do
potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    angulo.write(valorAngAngulo);                 //Envio do angulo para o Servo

    valorPotGarra = analogRead(potGarra);          //Leitura do Angulo do
Potenciometro
    valorAngGarra = map(valorPotGarra, 0, 1024, 0, 90); //Conversão do valor do
potenciometro (de 0 até 1024) para o angulo (de 0 até 180)
    garra.write(valorAngGarra);                   //Envio do angulo para o Servo
  }

  //Se o botao de congelar foi apertado

```

```
estadoBotaoC = digitalRead(botaoCongela);
if (estadoBotaoC != estadoAntBotaoC) {
    congelado = true;
    digitalWrite(13, HIGH); //Informa o estado de congelado no LED interno do
    Arduino
}
estadoAntBotaoC = estadoBotaoC;

//Se o botao de descongelar foi apertado
estadoBotaoD = digitalRead(botaoDescongela);
if (estadoBotaoD != estadoAntBotaoD) {
    congelado = false;
    digitalWrite(13, LOW); //Informa o estado de congelado no LED interno do
    Arduino
}
estadoAntBotaoD = estadoBotaoD;

}
```