



**SÃO PAULO**  
GOVERNO DO ESTADO

**CENTRO PAULA SOUZA**  
**ETEC PROF. MARCOS UCHÔAS DOS SANTOS PENCHEL**  
**Técnico em Eletrônica ao Médio – Novotec Integrado**

**CARRINHO CONTROLADO VIA BLUETOOTH**  
***CART CONTROLLED VIA BLUETOOTH***

**Andrey Inacio da Silva**  
**Jeziel Nathan dos Santos Clemente**  
**Nicolas Raimundo Calioni de Oliveira**

**Orientadores:**  
**Prof. Felipe Lopes Cavalcanti**  
**Prof. Marco Antonio Guedes**

**Resumo:** O nosso projeto consiste em controlar um carrinho pelo bluetooth do celular em curtas distâncias. Este projeto tem como objetivo principal desenvolver um sistema que permita o controle remoto de um carrinho elétrico/eletrônico por meio de dispositivos móveis, como smartphones ou tablets. Utilizando a tecnologia Bluetooth, o projeto integra componentes de hardware e software para criar uma interface amigável e eficiente. O projeto representa uma excelente oportunidade para aprender conceitos de comunicação sem fio, automação e integração de sistemas, sendo popular em cursos de engenharia e áreas correlatas.

**Palavras-chave:** Carrinho. Bluetooth. Celular.

**Keywords:** Cart. Bluetooth. Phone's .

## INTRODUÇÃO

Neste projeto, vou mostrar um Robô Controlado por Bluetooth usando um módulo Bluetooth Arduino e HC-05. Vou usar um smartphone e um aplicativo Android que pode operar o carro robótico.

Este robô controlado por Bluetooth é operado simplesmente usando a tecnologia Bluetooth e também com o comando do usuário. O usuário tem que instalar um aplicativo em seu smartphone para controlar este robô.

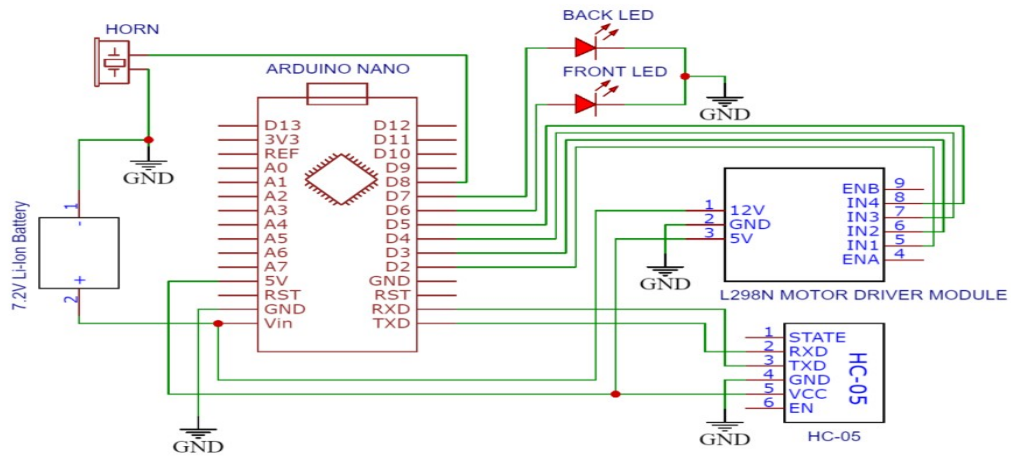
Depois de emparelhar com um smartphone e um robô controlado por Bluetooth, o usuário pode operá-lo por meio de comandos como avançar, retroceder, direita, esquerda e parar.

## DESENVOLVIMENTO

### Projeto



## Diagrama de Circuito



CIRCUITDIAGRAMS.IN

ELECTRO GADGET

## Componentes

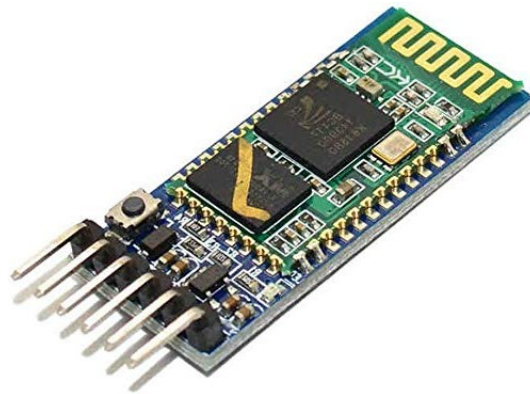
- Arduino uno
- HC-05 Bluetooth Module
- L298N Motor Driver
- BO Motor (x2)
- LED
- Smartphone
- Bluetooth Controlled App
- Jumper Wires
- Mini Breadboard
- Chasis
- 9V Battery(2x)

## Arduino Uno



O **microcontrolador** Arduino Uno é uma plataforma de prototipagem eletrônica baseada em um microcontrolador ATmega328P fabricado pela empresa italiana Arduino.

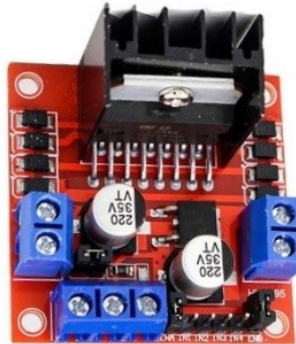
### HC-05 Bluetooth Module



O Módulo Bluetooth HC-05 é responsável por capacitar a comunicação Bluetooth entre o Arduino e o telefone Android. Já possui uma taxa de transmissão padrão de 9600 bps. Basta conectar RX e TX com um conversor serial do Arduino e pronto.

O módulo Bluetooth HC-05 possui dois modos, um em modo mestre e outro em modo escravo. Podemos definir qualquer modo usando alguns comandos AT. Até podemos definir as configurações do módulo usando o comando AT. Primeiro, precisamos entrar no modo AT com uma taxa de transmissão de 38.400 bps pressionando o botão Enable no módulo Bluetooth ou fornecendo um nível ALTO no pino Enable.

## L298N Motor Driver



O módulo acionador do motor L298N é responsável por fornecer a corrente de acionamento vital aos motores do carro robótico.

NB: Estou usando um driver de motor L298N neste projeto, mas você também pode usar um driver de motor L293D, pois usamos apenas dois motores BO aqui. Mas no caso de quatro motores BO, devemos usar o driver do motor L298N.

### Conexão de Circuito

Para construir este robô controlado por Bluetooth, primeiro devemos procurar o módulo Bluetooth HC-05 onde os pinos +5V e GND estão conectados diretamente à fonte de alimentação de 5V ou à fonte de 5V do Arduino.

Depois disso, precisamos conectar o pino TX do módulo Bluetooth ao pino RX do Arduino, pois sabemos que só devemos transmitir dados do smartphone para o módulo Bluetooth do robô controlado por Bluetooth e não precisamos receber nenhum tipo de informação do Arduino.

Este pino RX do Arduino depende da biblioteca SoftwareSerial (o pino 0 e o pino 1 são organizados como pinos RX e TX do Arduino).

Atualmente, os pinos 2 ao pino 5 de E/S digital do módulo driver do motor L298N são projetados para os pinos de saída do driver do motor e estão associados a IN1 a IN4 do módulo driver do motor L298N.

**Arduino Nano**

**HC-05 Module**

+5V

VCC

GND

GND

TX

RX

RX

TX

**Arduino Nano**

**L298N Motor Driver**

D2

IN1

D3

IN2

D4

IN3

D5

IN4

## Aplicativo android para controlar o robô

Aqui usei um aplicativo simples controlado por Bluetooth chamado “Arduino Bluetooth RC Car” disponível na Playstore, que pode se comunicar com um módulo Bluetooth usando comunicação serial. Após instalar o aplicativo, precisamos ligar o Bluetooth e a localização no smartphone para que funcione corretamente.

Agora toque na guia de configurações e faça uma varredura para encontrar o módulo Bluetooth correto. Existem algumas teclas básicas listadas na guia de configurações que incluí no código do Arduino para executar suas funções básicas que são avançar, retroceder, direita, esquerda, etc. Também podemos controlar este robô usando controle por gestos com este mesmo aplicativo.

Function	Code
Forward	F
Backward	B
Left	L
Right	R
Forward Left	G
Forward Right	I
Backward Left	H
Backward Right	J
Stop	S

<b>Function</b>	<b>Code</b>
Front Light ON	W
Front Light OFF	w
Back Light ON	U
Back Light OFF	u
Horn ON	V
Horn OFF	v

## **Princípio de funcionamento do robô controlado por Bluetooth**

Para começar, para este robô controlado por Bluetooth, utilizei 7 teclas para frente, para trás, esquerda, direita, luz ON/OFF, buzina e velocidade.

No momento em que uma tecla é pressionada pelo usuário, as informações de comparação são transmitidas ao módulo Bluetooth do smartphone através da comunicação serial Bluetooth.

No código Arduino, o Arduino obtém qualquer uma dessas informações do módulo Bluetooth (de acordo com o mapeamento de teclas) e executa uma atividade simples de switch case, onde cada caso é relacionado com direções adequadas para os pinos de entrada do driver do motor.

Por exemplo, se a tecla avançar for pressionada no aplicativo do smartphone, o Arduino ativará IN1 e IN3 do módulo acionador do motor para mover o robô na direção direta enquanto IN2 e IN4 estão no modo de desativação.



## **Aplicação de robô controlado por Bluetooth**

- Robô móvel de baixo alcance controlado por Bluetooth.
- Aplicativo de robô espião militar.
- Robôs de automação residencial como uma cadeira de rodas.

## **Desvantagens do robô controlado por Bluetooth**

- Como sabemos, o alcance de conectividade da comunicação Bluetooth é de aproximadamente 10 metros, portanto o alcance de controle deste robô também é limitado.
- Se a energia não for suficiente para todos os módulos, especialmente o módulo Bluetooth, todo o processo não funcionará. Além disso, não podemos conectar mais de 3 baterias ao chassi pequeno. Portanto, precisamos de baterias poderosas.

## **Desenvolvimento futuro**

- No futuro, poderemos adicionar mais sensores para funcionar de forma eficiente.
- Também podemos adicionar uma câmera sem fio e um display para visualizar onde o robô estará se movendo.

## **Arduino Code**

```
int motorRF = 2; //right front motor
int motorRB = 3; //Left front motor
int motorLF = 4; //Right back motor
int motorLB = 5; //Left back motor
int lightF = 6; //Front light

int lightB = 7; //Back light
int horn = 8; //Horn
int ena = 10; //Enable A
int enb = 11; //Enable B
```

```

int spd = 26; //Speed
char val;
void setup()
{
  pinMode(motorRF, OUTPUT); //D2 set to IN1
  pinMode(motorRB, OUTPUT); //D3 set to IN2
  pinMode(motorLF, OUTPUT); //D4 set to IN3
  pinMode(motorLB, OUTPUT); //D5 set to IN4
  pinMode(lightF, OUTPUT); //D6 set to Front Light
  pinMode(lightB, OUTPUT);
  pinMode(motorLB, OUTPUT);
  pinMode(motorLB, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  while (Serial.available() > 0)
  {
    val = Serial.read();
    Serial.println(val);
  }
  if ( val == 'F') //Forward
  {

    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, HIGH);
    digitalWrite(motorLF, HIGH);
    digitalWrite(motorRB, LOW);
  }
}

```

```
    digitalWrite(motorLB, LOW);
}
else if (val == 'B') //Backward
{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, LOW);
    digitalWrite(motorRB, HIGH);
    digitalWrite(motorLB, HIGH);
}
else if (val == 'L') //Left
{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, HIGH);
    digitalWrite(motorRB, LOW);
    digitalWrite(motorLB, LOW);
}
else if (val == 'R') //Right

{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, HIGH);
    digitalWrite(motorRB, LOW);
```

```
    digitalWrite(motorLB, LOW);
}
else if (val == 'S') //Stop
{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, LOW);
    digitalWrite(motorRB, LOW);
    digitalWrite(motorLB, LOW);
}
else if (val == 'I') //Forward Right
{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, HIGH);
    digitalWrite(motorRB, LOW);
    digitalWrite(motorLB, LOW);
}
else if (val == 'J') //Backward Right

{
    analogWrite(ena, spd);
    analogWrite(enb, spd);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorLF, LOW);
    digitalWrite(motorRB, LOW);
    digitalWrite(motorLB, HIGH);
```

```
}  
else if (val == 'G') //Forward Left  
{  
  analogWrite(ena, spd);  
  analogWrite(enb, spd);  
  digitalWrite(motorRF, HIGH);  
  digitalWrite(motorLF, LOW);  
  digitalWrite(motorRB, LOW);  
  digitalWrite(motorLB, LOW);  
}  
else if (val == 'H') //Backward Left  
{  
  analogWrite(ena, spd);  
  analogWrite(enb, spd);  
  digitalWrite(motorRF, LOW);  
  digitalWrite(motorLF, LOW);  
  digitalWrite(motorRB, HIGH);  
  digitalWrite(motorLB, LOW);  
}  
else if (val == '1')  
{  
  
spd = 26;  
}  
else if (val == '2')  
{  
  spd = 51;  
}  
else if (val == '3')
```

```
{  
    spd = 77;  
}  
else if (val == '4')  
{  
    spd = 102;  
}  
else if (val == '5')  
{  
    spd = 128;  
}  
else if (val == '6')  
{  
    spd = 153;  
}  
else if (val == '7')  
{  
    spd = 179;  
}  
else if (val == '8')  
{  
  
    spd = 204;  
}  
else if (val == '9')  
{  
    spd = 230;  
}  
else if (val == 'q')
```

```
{
    spd = 255;
}
else if (val == 'W') //front Light ON
{
    digitalWrite(lightF, HIGH);
}
else if (val == 'w') //front Light OFF
{
    digitalWrite(lightF, LOW);
}
else if (val == 'U') //back Light ON
{
    digitalWrite(lightB, HIGH);
}
else if (val == 'u') //back Light OFF
{
    digitalWrite(lightB, LOW);
}
else if (val == 'V') //Horn ON

{
    digitalWrite(horn, HIGH);
}
else if (val == 'v') //Horn OFF
{
    digitalWrite(horn, LOW);
}}
```

## CONCLUSÃO:

O projeto do carrinho controlado via Bluetooth demonstra como tecnologias acessíveis podem ser aplicadas em sistemas de automação e controle remoto. Utilizando um módulo Bluetooth integrado à plataforma Arduino, foi possível estabelecer uma comunicação eficiente entre o carrinho e dispositivos móveis, proporcionando uma experiência prática e funcional.

Além de ser uma solução de baixo custo e fácil implementação, o projeto também abre portas para diversas aplicações em robótica e automação. Como continuidade, pode-se incluir sensores para aumentar a autonomia e funcionalidades do carrinho, bem como explorar novas interfaces de controle e protocolos de comunicação. Dessa forma, o projeto reforça o potencial educacional e prático da integração entre hardware e software, contribuindo para o avanço de tecnologias baseadas em IoT e sistemas móveis.

**Abstract:** Our project involves controlling a cart using your cell phone's Bluetooth over short distances.

## REFERÊNCIAS:

USINAINFO. **Carrinho Controlado via Bluetooth**. 2024. Disponível em: <https://www.usinainfo.com.br/blog/carrinho-arduino-controlado-por-bluetooth-e-sistema-android/>. Acesso em: 22 abr. 2024.