



Carrinho de Comandado por Voz

Relatório de Atividade

ÁLVARO R. ARAÚJO, ITALO J. DIAS, SARAH C. DE OLIVEIRA, TIAGO M. BONFIM,
WILLIAN B. DA SILVA

Petianos responsáveis pelo projeto, em ordem alfabética

Resumo

O projeto consiste em um carrinho que atende aos comandos de voz imperativos, para determinar seu comportamento. O protótipo foi integrado ao conjunto de projetos do PETEE-UFMG a partir de parceria com o OptmaLab (UFMG), que o forneceu em prol da troca de conhecimentos. O Carrinho Controlado por Voz é um trabalho de robótica, montado sobre um chassi, e tem como principais componentes um microcontrolador Arduino para processamento e um sensor bluetooth para garantir a comunicação com um aplicativo de celular.

CONTEÚDO

I	Introdução	1
II	Materiais e Métodos	2
III	Resultados	9
IV	Discussão	9

I. INTRODUÇÃO

De maneira sucinta, o usuário transmite os comandos para o Carrinho Comandado por Voz por meio de um aplicativo no celular e a informação é enviada para o Arduino no protótipo através do módulo bluetooth. Ali, o mesmo decodifica as instruções, e as transmite para os motores e rodas. No Arduino encontra-se o cérebro do projeto, onde as diretrizes são traduzidas em ações via programação prévia do dispositivo.

Com o objetivo de construir um Carrinho com esse comportamento, foi necessário reunir o conhecimento já consolidado no OptmaLab(UFMG), à experiência do grupo PETEE-UFMG com carrinhos robóticos. Nesse sentido, foram propostos aperfeiçoamentos de velocidade, controle e estabilidade.

II. MATERIAIS E MÉTODOS

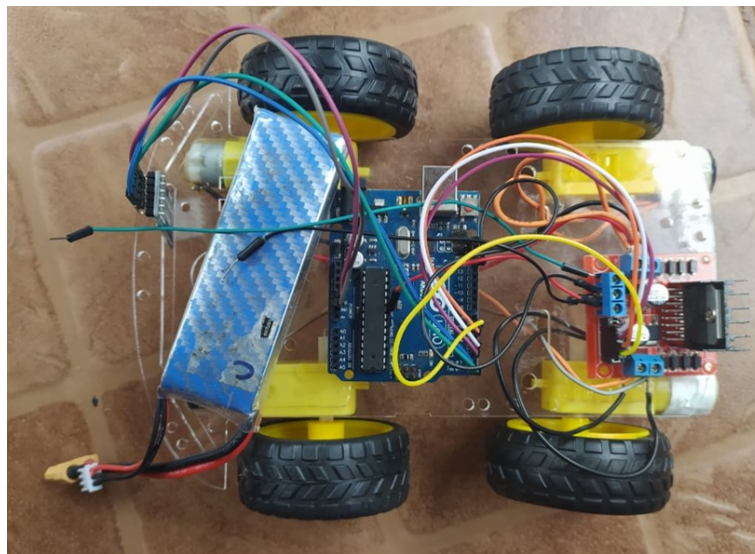


Figura 1: Carrinho Comandado por Voz

Materiais e Protótipos

Materiais usados na prototipagem e montagem do Carrinho Comandado por Voz:

- Arduino;
- Módulo Bluetooth HC-05;
- Driver Ponte H L298N;
- Bateria 12V 1300mAh;
- 4X Motores DC de 12V;
- Bateria 12V 1300mAh;
- Um resistor de 330 Ω ;
- Um resistor de 680 Ω ;
- Cabos Variados;

II MATERIAIS E MÉTODOS

- Chassi.

Os materiais usados nesta montagem possuem um valor relativamente elevado, o que pode dificultar a execução e a correção do protótipo.

O esquemático do circuito montado se encontra na Fig 2.

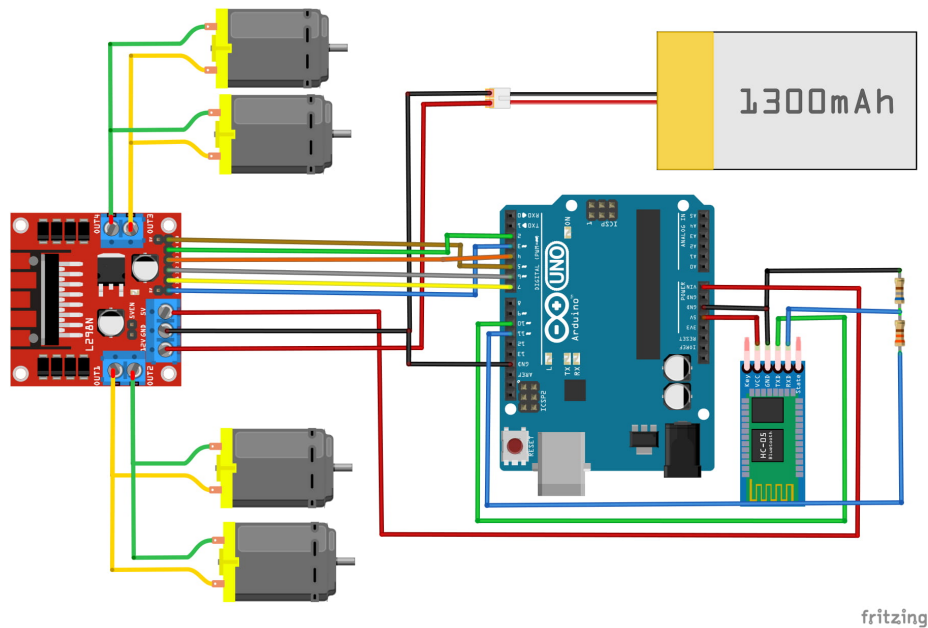


Figura 2: *Esquemático Circuito*

Foram feitos alguns protótipos do circuito antes que o carrinho funcionasse. Os principais erros na estruturação destes foram a montagem das partes do circuito, a configuração do aplicativo de comunicação Bluetooth do celular e o código do Arduino. Após o funcionamento do protótipo alteramos o chassi para uma melhor adequação do circuito.

Funcionamento do Circuito

O circuito do carrinho controlado por voz possui três partes essenciais, o Arduino, o Módulo Bluetooth HC-05 e o Driver Ponte H L298N. O Módulo HC-05, Fig 3, é responsável por permitir uma comunicação via bluetooth, sem a necessidade de fios, entre o Arduino (receptor) e o usuário (transmissor). Um diagrama do circuito é mostrado a seguir:

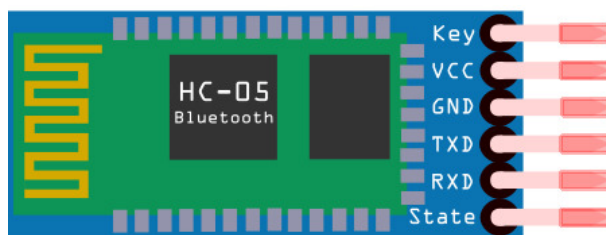


Figura 3: Diagrama do HC-05

O HC-05 possui 6 pinos, sendo estes:

- +5V - Pino de alimentação, conectado a 5V;
- GND - Pino de terra, conectado a 0V;
- TX - Pino de transmissão dos dados da placa bluetooth;
- RX - Pino de recepção dos dados que vem do Arduino;
- Key - Pino que altera entre o modo de dados e o modo de comandos AT; temporização do circuito.
- State - Pino conectado ao LED vermelho, permite informar se a comunicação bluetooth esta funcionando.

O módulo HC-05, ao receber 5V no pino Key, entra em modo de comandos AT, já ao receber 0V neste, o módulo entra em modo de dados. Logo este pino não é necessário estar conectado a nada já que, por padrão, ele fica no modo de dados.

Note que o pino RX, de recepção dos dados do arduino para o módulo bluetooth, necessita de uma alimentação de aproximadamente 3,3V. Sendo a saída dos pinos do arduino projetada para um sinal de 5V é necessário realizar um divisor de tensão entre o pino 11 do arduino e o pino RX do HC-05 que possibilite um sinal de 3,3V. Esse circuito divisor de tensão ocorre na associação de resistores de 330 Ω e 680 Ω .

A troca de dados entre o Arduino e o módulo em questão se dá através de uma comunicação serial. Ou seja, a transferência de dados é feita bit a bit (um de cada vez).

A forma do usuário se comunicar com o módulo se dá por meio de aplicativos de celular como o Arduino BlueControl e o Arduino-Bluetooth-voice-control. Estes aplicativos recebem a voz do usuário e codificam as palavras ou frases ditas como uma string. Transmitindo, assim, estes dados através da rede bluetooth criada pelo HC-05 para o arduino.

O arduino ao obter a informação do HC-05 ira, de acordo com a string recebida, emitir um sinal para a ponte H de forma a realizar uma ação no carrinho, seja essa acelerar, freiar, esquerda, direita, etc. Veja a seguir o código utilizado no arduino para interpretação do sinal recebido do módulo bluetooth:

II MATERIAIS E MÉTODOS

Código Arduino

```
1
2 #include <SoftwareSerial.h>
3 SoftwareSerial voicecontrol(10, 11); //TX, RX
4 String readvoice;
5
6 int IN1 = 7; // IN1
7 int IN2 = 6; // IN2
8 int IN3 = 4; // IN3
9 int IN4 = 2; // IN4
10 int ENA = 3; // PWM 1
11 int ENB = 5; // PWM 2
12
13 void setup() {
14   voicecontrol.begin(9600);
15   Serial.begin(9600);
16   pinMode (IN1, OUTPUT);
17   pinMode (IN2, OUTPUT);
18   pinMode (IN3, OUTPUT);
19   pinMode (IN4, OUTPUT);
20   pinMode(ENA,OUTPUT);
21   pinMode(ENB,OUTPUT);
22 }
23
24 void loop() {
25   while (voicecontrol.available()){
26     delay(10);
27     char c = voicecontrol.read();
28     readvoice += c;
29   }
30
31   if (readvoice.length() > 0) {
32
33     Serial.println(readvoice);
34
35     if(readvoice == "para frente"){
36       digitalWrite (IN1, LOW);
37       digitalWrite (IN2, HIGH);
38       digitalWrite (IN3, HIGH);
39       digitalWrite (IN4, LOW);
40       analogWrite(ENA,100);
41       analogWrite(ENB,110);
42       delay(100); }
43
44     else if(readvoice == "para trás"){
45       digitalWrite (IN1, HIGH);
46       digitalWrite (IN2, LOW);
47       digitalWrite (IN3, LOW);
48       digitalWrite (IN4, HIGH);
49       analogWrite(ENA,100);
50       analogWrite(ENB,110);
```

```
51     delay(100); }
52
53 else if (readvoice == "acelera"){
54     digitalWrite (IN1, LOW);
55     digitalWrite (IN2, HIGH);
56     digitalWrite (IN3, HIGH);
57     digitalWrite (IN4, LOW);
58     analogWrite(ENA,150);
59     analogWrite(ENB,150);
60     delay (100);}
61
62 else if (readvoice == "freio"){
63     digitalWrite (IN1, LOW);
64     digitalWrite (IN2, HIGH);
65     digitalWrite (IN3, HIGH);
66     digitalWrite (IN4, LOW);
67     analogWrite(ENA,60);
68     analogWrite(ENB,60);
69     delay (100);}
70
71 else if (readvoice == "esquerda"){
72     digitalWrite (IN1, LOW);
73     digitalWrite (IN2, HIGH);
74     digitalWrite (IN3, HIGH);
75     digitalWrite (IN4, LOW);
76     analogWrite(ENA,50);
77     analogWrite(ENB,100);
78     delay (100);}
79
80 else if ( readvoice == "direita"){
81     digitalWrite (IN1, LOW);
82     digitalWrite (IN2, HIGH);
83     digitalWrite (IN3, HIGH);
84     digitalWrite (IN4, LOW);
85     analogWrite(ENA,100);
86     analogWrite(ENB,70);
87     delay (100);}
88
89 else if (readvoice == "stop"){
90     digitalWrite (IN1, LOW);
91     digitalWrite (IN2, LOW);
92     digitalWrite (IN3, LOW);
93     digitalWrite (IN4, LOW);
94     delay (100);}
95
96 else if (readvoice == "dança"){
97     digitalWrite (IN1, LOW);
98     digitalWrite (IN2, HIGH);
99     digitalWrite (IN3, LOW);
100    digitalWrite (IN4, LOW);
101    delay (400);
102
103    digitalWrite (IN1, HIGH);
```

II MATERIAIS E MÉTODOS

```
104 digitalWrite (IN2, LOW);
105 digitalWrite (IN3, HIGH);
106 digitalWrite (IN4, LOW);
107 delay(600);
108
109 digitalWrite (IN1, LOW);
110 digitalWrite (IN2, HIGH);
111 digitalWrite (IN3, HIGH);
112 digitalWrite (IN4, LOW);
113 delay (500);
114
115 digitalWrite (IN1, HIGH);
116 digitalWrite (IN2, LOW);
117 digitalWrite (IN3, LOW);
118 digitalWrite (IN4, HIGH);
119 delay (500);
120
121 digitalWrite (IN1, LOW);
122 digitalWrite (IN2, HIGH);
123 digitalWrite (IN3, LOW);
124 digitalWrite (IN4, LOW);
125 delay (400);
126
127 digitalWrite (IN1, HIGH);
128 digitalWrite (IN2, LOW);
129 digitalWrite (IN3, HIGH);
130 digitalWrite (IN4, LOW);
131 delay(600);
132
133 digitalWrite (IN1, LOW);
134 digitalWrite (IN2, HIGH);
135 digitalWrite (IN3, HIGH);
136 digitalWrite (IN4, LOW);
137 delay (500);
138
139 digitalWrite (IN1, HIGH);
140 digitalWrite (IN2, LOW);
141 digitalWrite (IN3, LOW);
142 digitalWrite (IN4, HIGH);
143 delay (500);
144
145 digitalWrite (IN1, LOW);
146 digitalWrite (IN2, HIGH);
147 digitalWrite (IN3, LOW);
148 digitalWrite (IN4, LOW);
149 delay (400);
150
151 digitalWrite (IN1, HIGH);
152 digitalWrite (IN2, LOW);
153 digitalWrite (IN3, HIGH);
154 digitalWrite (IN4, LOW);
155 delay(600);
156
```

```
157   digitalWrite (IN1, LOW);  
158   digitalWrite (IN2, HIGH);  
159   digitalWrite (IN3, HIGH);  
160   digitalWrite (IN4, LOW);  
161   delay (500);}  
162  
163   readvoice="";  
164   }  
165 }
```

Interpretação do Código

Observamos que a biblioteca incluída no código é a `SoftwareSerial.h`. Esta biblioteca permite que outros pinos I/O do arduino sejam portas serial.

Após a inclusão da biblioteca vemos a definição do objeto `voicecontrol` associada ao `SoftwareSerial`. Este define os pinos 10 e 11 do arduino como transmissor e receptor serial, respectivamente. Na linha 4 é definido a string que irá receber a palavra dita pelo usuário.

Nas linhas 6 a 11 do código é definido os pinos associados as entradas da ponte H. Quatro dos pinos são responsáveis pela direção tomada pelo carrinho e dois pinos são responsáveis pelo PWM, associado a aceleração e desaceleração do carrinho. Das linhas 13 a 22 os pinos definidos são inicializados bem como a taxa de transferência de bits por segundo das portas serial.

Nas linhas 25 a 28 é recebido a frase dita pelo usuário. Na linha 25 o arduino verifica se existe algum dado na porta serial `voicecontrol`. Enquanto existir dados para serem lidos, na linha 27, uma variável do tipo `char` recebe o caractere que se encontra na porta serial. Na linha 26 a string `"acumula"` os caracteres de forma a construir a frase dita pelo usuário.

Após não existir mais dados para serem lidos o código irá interpretar, por meio dos `if's` e `else if's`, a frase dita. Aqui iremos explicar apenas o funcionamento do primeiro comando `"para frente"` pois as outras ações são análogas a esta.

Nas linhas 36 a 42 o arduino interpreta e executa a ação `"pra frente"` dita pelo usuário. Na linha 36 o código verifica se a string `readvoice` é igual ao comando `"pra frente"`. Caso seja verdade das linhas 36 a 39 os pinos de saída digital possuem valores correspondentes a ação de andar para frente interpretada pela ponte H, que neste caso é LOW, HIGH, HIGH e LOW. As linhas 40 e 41 são responsáveis pelo PWM emitido pelo arduino através da função `analogWrite`. Esta função possui como parâmetro o pino de saída e o valor do `dutycycle` do PWM.

REFERÊNCIAS

III. RESULTADOS

Os resultados apresentados pelo projeto foram satisfatórios. O objetivo do projeto era que o carrinho recebesse um comando de voz por meio do celular e realizasse a ação correspondente. Sendo assim, o objetivo foi alcançado, ou seja, dado um comando de voz, dentre os comandos descritos em código, o carrinho é capaz de realizar a ação correspondente. Por exemplo: quando um comando de frente é acionado, a mensagem é enviada ao carrinho que possui uma ação correspondente ao comando, dessa forma, o microcontrolador irá mandar um sinal para seus atuadores, que por sua vez, realizaram a ação do comando proposto.

Um ponto a ser considerado durante a execução do projeto é em relação a forma de captura de voz. O comando de voz passado é capturado por meio de um aplicativo, e esse aplicativo utiliza do google para reconhecer as palavras. Dessa forma, a escolha do aplicativo em que o comando será passado é parte essencial do projeto, podendo melhorar a qualidade da aplicação dependendo do quão bem funciona o aplicativo.

O projeto apresenta nível de complexidade baixa, logo é possível utilizar o protótipo do carrinho comando por voz para fins didáticos. Dessa forma, sua utilização em eventos ,como a mostra de profissões, se faz interessante.

IV. DISCUSSÃO

O carrinho comandado por voz se mostrou um protótipo funcional e teve bons resultados. O preço elevado de seus componentes dificulta tanto a realização de testes para a melhoria de suas funcionalidades, quanto a possível reprodução do protótipo. No entanto, o modelo atual funciona bem e os aperfeiçoamentos propostos com relação à versão do OptmaLab foram atingidos. No futuro, é desejável que novas mudanças sejam feitas, para permitir que o controle seja mais apurado.

REFERÊNCIAS

Entendendo Módulo Bluetooth HC-05:

<<http://mundoprojetado.com.br/modulo-bluetooth-comunicando-c-arduino-parte-1/>>

<<http://mundoprojetado.com.br/modulo-bluetooth-criando-aplicativo-parte-2/>>