**FACULDADE IMPACTA DE TECNOLOGIA**

**Daniel Roberto 1800479**

**João Vitor Paulino 1801021**

**Milton Lima 1801234**

**Ramon C. Pires 1800260**

**Tiago Beneteli 1800804**

**INTRODUÇÃO A INTERNET DAS COISAS**

**SÃO PAULO**

**2018**

**Objetivo:**

O trabalho tem por objetivo a criação de um carrinho de controle remoto controlado via Bluetooth utilizando a plataforma Arduino. A escolha dos materiais é livre tendo que respeitar somente algumas regras impostas pelo organizador. Haverá também um campeonato entre as classes onde dois carrinhos tentam se empurrar para fora de um ringue tendo um tempo definido para tal feito.

**Projeto:**

O trabalho foi particionado em partes distintas que posteriormente foram adicionadas ao projeto final, sendo elas:

* **Montagem da parte mecânica do carro:** Foi necessário a utilização de ferramentas como chaves de fenda para prender e apertar os parafusos que mantes as peças mecânicas do carro fixas, fitas dupla-face e foi necessário também a utilização de ferramentas de soldagem elétrica para soldar peças elétricas nos motores DC e no Arduino.
* **Desenvolvimento do Firmware do Microcontrolador:** A parte lógica da aplicação que fará o sistema de movimento do carrinho funcionar foi programada em C na IDE Arduino, consistiu na confecção de um algoritmo que compara a entrada Serial (RX, TX) com o devido comando estipulado.
* **Conexão Bluetooth com o smartphone:** Depois de implementado o algoritmo do controle de movimentos foi necessário a entrada de dados via Bluetooth para tornar o carrinho wireless definitivamente. Até então, os testes estavam sendo efetuados via conexão serial USB, a comunicação do modulo Bluetooth HC06 pode ser feita apenas conectando-o a pinagem RX e TX do modulo as portas 1 e 0 respectivamente, e o smartphone pode ser utilizado como percursor dos comandos, usamos o aplicativo para celular chamado “Bluetooth Arduino RC Car”.

**Materiais e componentes utilizados:**

1x - Placa Arduino UNO R3;

1x - Motor Shield L293D Driver Ponte H;

1x - Chassi 2WD (2 rodas);

1x - Modulo Bluetooth HC-06;

1x - Bateria 9 volts Recarregável 450maH;

1x - Clip de Bateria/Pilha;

4x - Pilhas AA alcalinas;

1x - Case para 4 pilhas AA;

Xx - Jumpers diversos;

**Placa Arduino UNO R3:** A plataforma que utilizamos para processar o algoritimo de movimento do carrinho.

**Motor Shield L293D Driver Ponte H:** A Shield contém um microcontrolador e duas pontes H modelo L293D, ela serve para habilitar e desabilitar trilhas passagem de corrente elétrica em motores DC, fazendo-o rotacionar em direções distintas descritas pela linha de execução do algoritmo do Arduino, é necessário usar uma biblioteca chamada AFMotor para que haja o correto funcionamento da shield.

**Chassi 2WD:** Uma plataforma em acrílico onde são colocados todos os componentes do carrinho, ela tem três rodas, sendo uma de apoio e duas com motores DC com caixas de redução para gerar os movimentos.

**Modulo Bluetooth HC-06:** Módulo que faz a comunicação Bluetooth com o mundo exterior, seu uso é bem amplo podendo ser usado para receber dados de diversas fontes que detenham a mesma banda de conexão que ela. Será usado no projeto para receber dados do smartphone.

**Bateria 9 volts Recarregável:** A Shield utilizada concede a opção de se utilizar uma fonte independente de energia elétrica, a bateria será utilizada para alimentar os motores independente da alimentação do Arduino e módulo Bluetooth.

**Clip de Bateria/Pilha:** Usado para plugar as fontes de alimentação no Arduino e na Shield.

**Pilhas AA alcalinas:** Usada para alimentar o Arduino e o modulo independentemente da alimentação dos motores.

**Case para 4 pilhas AA:** Usada para acoplar o conjunto de pilhas AA no Chassi.

**Jumpers diversos:** Usada para fazer a conexão do modulo Bluetooth com o pino da placa Arduino.

**MONTAGEM DA PARTE ELÉTRICA**

BATERIA 9v

(FONTE)

SHIELD L293D

MOTOR 2

(ATUADOR)

MOTOR 1

(ATUADOR)

MÓDULO BT HC06

(SENSOR)

ARDUINO UNO R3

4x PILHAS AA

(FONTE)

**FLUXOGRAMA DO ALGORITMO**

**INICIO**

**Iniciar comunicação externa**

**não**

**Enquanto recebe dados**

**sim**

Variavel recebe valor externo

SE variavel = F

**não**

**não**

**sim**

SE variavel = B

**não**

Movimenta motor esquerdo e direito para frente

SE variavel = L

**não**

**sim**

Movimenta motor esquerdo para frente e direito para trás

Movimenta motor esquerdo e direito para trás

**sim**

SE variavel = R

**sim**

Movimenta motor esquerdo para trás e direito para frente

SE variavel = S

**sim**

Mantem os dois motores imóveis

**FIM**

**Código fonte:**

/\*

Obs: para o correto funcionamento é recomendável a instalação do aplicativo para Android "Arduino Bluetooth RC Car"

https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller

Obs2: para ocorrer a conexão bluetooth somente é necessário conectar saída tx do módulo na porta rx (0) do Arduino,

não sendo necessário nenhuma configuração extra.

\*/

#include <AFMotor.h> //importando a biblioteca da Shield L923D

/\*Seção de declaração das variáveis e declaração de entrada dos motores\*/

AF\_DCMotor Motor1(1);//criação do objeto Motor1 e Motor2 da classe AF\_DCMotor

AF\_DCMotor Motor2(2);//e definição de suas portas de entrada e saída da shield (M1 e M2)

char buf;//char que receberemos a entrada do bluetooth que compararemos com o comando

void setup(){

Serial.begin(9600);//Define a banda da comunicação serial do modulo HC06

}

void loop(){

while(Serial.available() > 0){ //enquanto a porta serial estiver recebendo qualquer valor

Motor1.setSpeed(255);//o método setspeed está especificado dentro da biblioteca

Motor2.setSpeed(255);//serve para definir a velocidade dos motores, 255 é o valor maximo da biblioteca

buf = Serial.read();//recebendo o valor do modulo bluetooth

switch (buf){ // comparando a entrada com o devido comando

case 'S':

Motor1.run(RELEASE);//motor 1 mantem parado

Motor2.run(RELEASE);//motor 2 mantem parado

break;

case 'F':

Motor1.run(FORWARD);//motor 1 mantem à frente

Motor2.run(FORWARD);//motor 2 mantem à frente

break;

case 'L':

Motor1.run(BACKWARD);//motor 1 mantem à trás

Motor2.run(FORWARD); //motor 2 mantem à frente o que o faz girar para a esquerda

break;

case 'R':

Motor1.run(FORWARD); //motor 1 mantem à frente

Motor2.run(BACKWARD);//motor 2 mantem à trás o que o faz girar para a direita

break;

case 'B':

Motor1.run(BACKWARD); //motor 1 mantem à trás

Motor2.run(BACKWARD); //motor 2 mantem à trás

break;

}

}

}