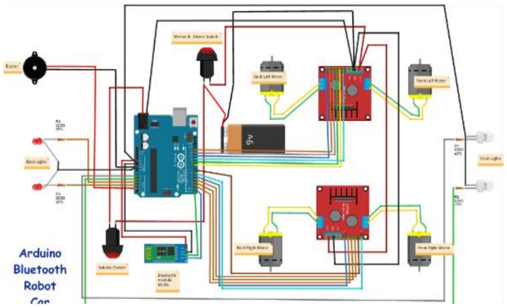
Carro con control remoto Bluetooth



- ○ *Descripción del ejercicio:*

Un automóvil robot Arduino 4WD controlado por teléfono inteligente o un robot Bluetooth Arduino. Puede moverse hacia adelante y hacia atrás, hacia la izquierda y hacia la derecha, cambiar su velocidad, encender / apagar las luces delanteras y traseras y también puede sonar.

Requisitos:

Arduino Bluetooth RC Car(PlayStore)

Soldador (genérico).

- ○ *Materiales*

  - 1 Arduino UNO
  - 1 HC-06 Bluetooth Module
  - 1 Placa de pruebas (Protoboard)
  - 1 Alambre para conexiones
  - 1 4WD Smart Robot Car Chassis Kit
  - 2 Spark Fun Dual H-Bridge motor driver (L298)
  - 4 LEDs
  - 1 Buzzer(Sumbador)
  - 4 Resistencias de 221Ω
  - 1 Pila de 9V
  - 2 Arduino Switch

- ○ *El código que usaremos será el siguiente*

```
#define light_FR 14 //LED Front Right pin A0 for Arduino Uno
#define light_FL 15 //LED Front Left pin A1 for Arduino Uno
#define light_BR 16 //LED Back Right pin A2 for Arduino Uno
#define light_BL 17 //LED Back Left pin A3 for Arduino Uno
#define horn_Buzz 18 //Horn Buzzer pin A4 for Arduino Uno
#define ENA_m1 5 // Enable/speed motor Front Right
#define ENB_m1 6 // Enable/speed motor Back Right
#define ENA_m2 10 // Enable/speed motor Front Left
#define ENB_m2 11 // Enable/speed motor Back Left
#define IN_11 2 // L298N #1 in 1 motor Front Right
#define IN_12 3 // L298N #1 in 2 motor Front Right
#define IN_13 4 // L298N #1 in 3 motor Back Right
#define IN_14 7 // L298N #1 in 4 motor Back Right
#define IN_21 8 // L298N #2 in 1 motor Front Left
#define IN_22 9 // L298N #2 in 2 motor Front Left
#define IN_23 12 // L298N #2 in 3 motor Back Left
#define IN_24 13 // L298N #2 in 4 motor Back Left
int command; //Int to store app command state.
int speedCar = 100; // 50 - 255.
int speed_Coeff = 4;
boolean lightFront = false;
boolean lightBack = false;
boolean horn = false;
void setup() {
pinMode(light_FR, OUTPUT);
pinMode(light_FL, OUTPUT);
pinMode(light_BR, OUTPUT);
pinMode(light_BL, OUTPUT);
pinMode(horn_Buzz, OUTPUT);
pinMode(ENA_m1, OUTPUT);
pinMode(ENB_m1, OUTPUT);
pinMode(ENA_m2, OUTPUT);
pinMode(ENB_m2, OUTPUT);
pinMode(IN_11, OUTPUT);
pinMode(IN_12, OUTPUT);
pinMode(IN_13, OUTPUT);
pinMode(IN_14, OUTPUT);
pinMode(IN_21, OUTPUT);
pinMode(IN_22, OUTPUT);
pinMode(IN_23, OUTPUT);
pinMode(IN_24, OUTPUT);
Serial.begin(9600);

}
void goAhead(){
digitalWrite(IN_11, HIGH);
digitalWrite(IN_12, LOW);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, LOW);
digitalWrite(IN_14, HIGH);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, LOW);
digitalWrite(IN_22, HIGH);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, HIGH);
digitalWrite(IN_24, LOW);
analogWrite(ENB_m2, speedCar);
}
void goBack(){
digitalWrite(IN_11, LOW);
digitalWrite(IN_12, HIGH);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, HIGH);
digitalWrite(IN_14, LOW);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, HIGH);
digitalWrite(IN_22, LOW);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, LOW);
digitalWrite(IN_24, HIGH);
analogWrite(ENB_m2, speedCar);
}
void goRight(){
digitalWrite(IN_11, LOW);
digitalWrite(IN_12, HIGH);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, HIGH);
digitalWrite(IN_14, LOW);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, LOW);
digitalWrite(IN_22, HIGH);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, HIGH);
digitalWrite(IN_24, LOW);
analogWrite(ENB_m2, speedCar);
}
void goLeft(){
digitalWrite(IN_11, HIGH);
digitalWrite(IN_12, LOW);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, LOW);
digitalWrite(IN_14, HIGH);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, HIGH);
digitalWrite(IN_22, LOW);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, LOW);
digitalWrite(IN_24, HIGH);
analogWrite(ENB_m2, speedCar);
```

```
}
void goAheadRight(){
digitalWrite(IN_11, HIGH);
digitalWrite(IN_12, LOW);
analogWrite(ENA_m1, speedCar/speed_Coeff);
digitalWrite(IN_13, LOW);
digitalWrite(IN_14, HIGH);
analogWrite(ENB_m1, speedCar/speed_Coeff);
digitalWrite(IN_21, LOW);
digitalWrite(IN_22, HIGH);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, HIGH);
digitalWrite(IN_24, LOW);
analogWrite(ENB_m2, speedCar);
}
void goAheadLeft(){
digitalWrite(IN_11, HIGH);
digitalWrite(IN_12, LOW);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, LOW);
digitalWrite(IN_14, HIGH);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, LOW);
digitalWrite(IN_22, HIGH);
analogWrite(ENA_m2, speedCar/speed_Coeff);
digitalWrite(IN_23, HIGH);
digitalWrite(IN_24, LOW);
analogWrite(ENB_m2, speedCar/speed_Coeff);
}
void goBackRight(){
digitalWrite(IN_11, LOW);
digitalWrite(IN_12, HIGH);
analogWrite(ENA_m1, speedCar/speed_Coeff);
digitalWrite(IN_13, HIGH);
digitalWrite(IN_14, LOW);
analogWrite(ENB_m1, speedCar/speed_Coeff);
digitalWrite(IN_21, HIGH);
digitalWrite(IN_22, LOW);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, LOW);
digitalWrite(IN_24, HIGH);
analogWrite(ENB_m2, speedCar);
}
void goBackLeft(){
digitalWrite(IN_11, LOW);
digitalWrite(IN_12, HIGH);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, HIGH);
digitalWrite(IN_14, LOW);
analogWrite(ENB_m1, speedCar);
```

```
digitalWrite(IN_21, HIGH);
digitalWrite(IN_22, LOW);
analogWrite(ENA_m2, speedCar/speed_Coeff);
digitalWrite(IN_23, LOW);
digitalWrite(IN_24, HIGH);
analogWrite(ENB_m2, speedCar/speed_Coeff);
}
void stopRobot(){
digitalWrite(IN_11, LOW);
digitalWrite(IN_12, LOW);
analogWrite(ENA_m1, speedCar);
digitalWrite(IN_13, LOW);
digitalWrite(IN_14, LOW);
analogWrite(ENB_m1, speedCar);
digitalWrite(IN_21, LOW);
digitalWrite(IN_22, LOW);
analogWrite(ENA_m2, speedCar);
digitalWrite(IN_23, LOW);
digitalWrite(IN_24, LOW);
analogWrite(ENB_m2, speedCar);
}
void loop(){
if (Serial.available() > 0) {
command = Serial.read();
stopRobot(); //Initialize with motors stopped.
if (lightFront) {digitalWrite(light_FR, HIGH); digitalWrite(light_FL,
HIGH);}
if (!lightFront) {digitalWrite(light_FR, LOW); digitalWrite(light_FL,
LOW);}
if (lightBack) {digitalWrite(light_BR, HIGH); digitalWrite(light_BL,
HIGH);}
if (!lightBack) {digitalWrite(light_BR, LOW); digitalWrite(light_BL,
LOW);}
if (horn) {digitalWrite(horn_Buzz, HIGH);}
if (!horn) {digitalWrite(horn_Buzz, LOW);}
switch (command) {
case 'F':goAhead();break;
case 'B':goBack();break;
case 'L':goLeft();break;
case 'R':goRight();break;
case 'I':goAheadRight();break;
case 'G':goAheadLeft();break;
case 'J':goBackRight();break;
case 'H':goBackLeft();break;
case '0':speedCar = 100;break;
case '1':speedCar = 115;break;
case '2':speedCar = 130;break;
case '3':speedCar = 145;break;
case '4':speedCar = 160;break;
case '5':speedCar = 175;break;
case '6':speedCar = 190;break;
case '7':speedCar = 205;break;
case '8':speedCar = 220;break;
case '9':speedCar = 235;break;
case 'q':speedCar = 255;break;
case 'W':lightFront = true;break;
case 'w':lightFront = false;break;
case 'U':lightBack = true;break;
case 'u':lightBack = false;break;
case 'V':horn = true;break;
case 'v':horn = false;break;
}
}
}
```