

3 de Septiembre del 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
ARQUITECTURA DE COMPUTADORES Y
ENSAMBLADORES I

Ing. Otto Escobar

Aux. David Jonathan Gonzáles Gámez

Aux. Andhy Lizandro Solís Osorio

Proyecto 1

“APLICACION ANDROID Y ARDUINO UTILIZANDO COMUNICACIÓN BLUETOOTH”

Fernando Mauricio Gómez Santos-----201901849

Walter Manolo Martínez Mateo-----201544295

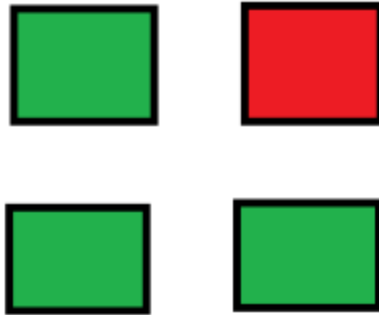
Gabriel Orlando Ajsivinac Xicay----- 201213010

DESCRIPCION

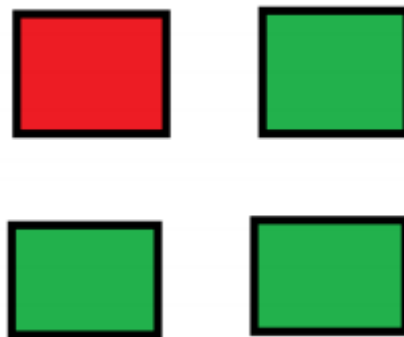
Este proyecto consiste en un carro simulado en proteus en cual será controlado por una aplicación Android a través de Bluetooth.

CARRO

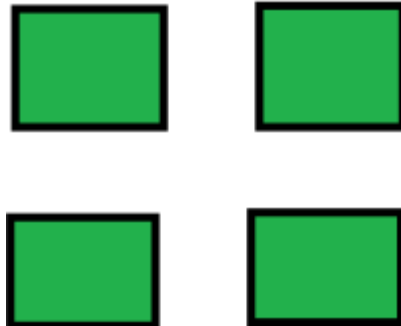
- Control de Velocidad: La velocidad se controla desde la aplicación Android.
 - Rápido
 - Lento
- Izquierda: Esta se controla desde la aplicación Android utilizando el botón respectivo, el cual toma el comportamiento de: 3 ruedas en funcionamiento, la rueda delantera derecha estará sin movimiento, además se prende la luz led izquierda.



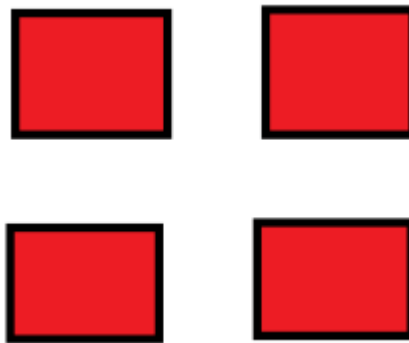
- Derecha: Se controla desde la aplicación Android utilizando el botón respectivo, el cual toma el comportamiento de: 3 ruedas en funcionamiento, la rueda delantera izquierda estará sin movimiento, además se prende la luz led derecha.



- Adelante y Reversa: Esta se controlará desde la aplicación Android utilizando el botón respectivo, el cual toma el comportamiento de: 4 ruedas estarán en funcionamiento, y la dirección de la rotación dependerá del sentido del movimiento.



- Detenerse: Desde la aplicación Android se presiona el botón respectivo, que tomara el siguiente comportamiento: 4 ruedas sin movimiento.



- Sensor Ultrasónico: El funcionamiento del sensor será el siguiente:
 - Si detecta algo diferente en $x \leq 20\text{cm}$
 - Los motores frenarán.
 - En la aplicación Android mostrara "El carro llegó a la meta".
 - El led trasero se activará.

APLICACIÓN ANDROID



Botón iniciar: Inicia la comunicación bluetooth con arduino.

Adelante: Le envía al Arduino la orden que el carro siga avanzando.

Reversa: El carro en este caso retrocederá.

Izquierda: El carro girará a la izquierda.

Derecha: El carro girará a la derecha.

Detenerse: El carro se detendrá. Apagando todos los motores.

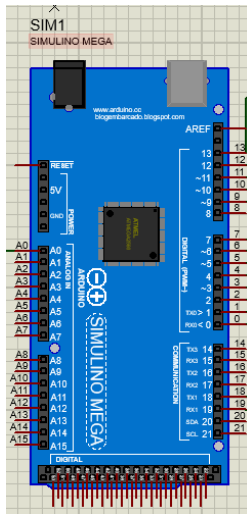
Frenar: Los motores giran lento.

Acelerar: Los motores giran rápido.

Etiqueta de llegada a la meta: Estará invisible y al momento de llegar a la meta se activará, o se podrá utilizar una alerta/mensaje al llegar a la meta.

COMPONENTES UTILIZADOS

- Arduino



- Celular Android



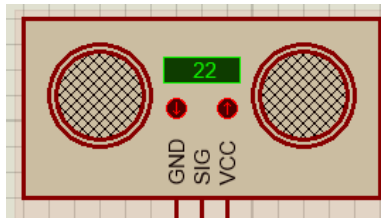
- 4 Motores DC



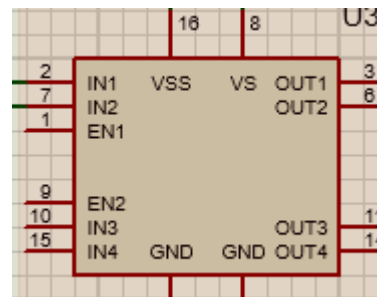
- Diodos Led



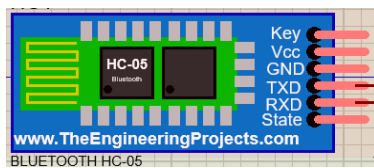
- 1 Sensor Ultrasonico



- L29Rd (como puente H)



- Modulo Bluetooth



CODIFICACION DE ENSAMBLAJE ARDUINO-PROTEUS

Variables globales utilizadas para en funcionamiento de motores y sensor.

```
int lento = 100;
int rapido = 255;
int valor;
int vel=255;
String test="";
int temporal=255;

//pin a utilizar del sensor
int pinSensor = A0;
```

Pines utilizados der arduino como salida.

```
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
pinMode(6,OUTPUT);
pinMode(5,OUTPUT);
pinMode(3,OUTPUT);
//leds
pinMode(7,OUTPUT);
pinMode(4,OUTPUT);
pinMode(2,OUTPUT);
```

Ciclo que obtendrá lo que la App le envíe.

```
while (Serial.available()){
    delay(10);
    char c = Serial.read();
    test=c;
}
```

Ciclo el cual ejecutara la acción respecto a lo que la App envié.

```
if(test.length()>0){

    if (test=="1"){
        adelante(temporal);
    }else if (test=="2"){
        atras(temporal);
    }else if (test=="3"){
        izq(temporal);
    }else if (test=="4"){
        der(temporal);
    }else if (test=="5"){
        parar();
    }else if (test=="6"){
        temporal=lento;
    }else if (test=="7"){
        temporal=rapido;
    }
}
test="";
```

Porción de código que determina la distancia del sensor ultrasónico.

```
int cm = 0.5 * Distancia(pinSensor,pinSensor) * 0.0344 + 1;
escuchar(cm);
delay(800);
```

Método el cual lee lo que está capturando el sensor ultrasónico.

```
long Distancia (int Trigger, int Echo){
    pinMode (Trigger,OUTPUT);
    digitalWrite(Trigger,LOW);
    delayMicroseconds(2);
    digitalWrite(Trigger,HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger,LOW);
    pinMode (Echo,INPUT);

    return pulseIn(Echo,HIGH);
}
```

Método por el cual se ejecuta la acción de detener motores DC después de que la distancia del sensor ultrasónico ≤ 20

```
int escuchar(int numero){
    if(numero <= 20){
        parar();
    }else{
        Serial.println("Siga bajando la distancia para parar motores");
    }
}
```

Código el cual se controla el giro de los motores “ADELANTE”

```
int adelante(int n){  
  
    analogWrite(11, n);  
    digitalWrite(10, LOW);  
    analogWrite(9, n);  
    digitalWrite(6, LOW);  
    analogWrite(5, n);  
    digitalWrite(3, LOW);  
    //leds  
    digitalWrite(7, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(2, LOW);  
}
```

Código el cual se controla el giro de los motores “ATRAS”

```
int atras(int n){  
    digitalWrite(11, LOW)  
    analogWrite(10, n);  
    digitalWrite(9, LOW);  
    analogWrite(6, n);  
    digitalWrite(5, LOW);  
    analogWrite(3, n);  
    //leds  
    digitalWrite(7, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(2, LOW);  
}
```

Código el cual se controla el giro de los motores “DETENER”

```
void parar(){  
    digitalWrite(11, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(9, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(3, LOW);  
    //leds  
    digitalWrite(7, HIGH);  
    digitalWrite(4, LOW);  
    digitalWrite(2, LOW);  
}
```


Código el cual se controla el giro de los motores “IZQUIERDA”

```
int izq(int n){  
  digitalWrite(11, LOW);  
  digitalWrite(10, LOW);  
  analogWrite(9, n);  
  digitalWrite(6, LOW);  
  analogWrite(5, n);  
  digitalWrite(3, LOW);  
  //leds  
  digitalWrite(7, LOW);  
  digitalWrite(4, LOW);  
  digitalWrite(2, HIGH);  
  
}
```

Código el cual controla el giro de los motores “DERECHA”

```
int der(int n){  
  analogWrite(11, n);  
  digitalWrite(10, LOW);  
  digitalWrite(9, LOW);  
  digitalWrite(6, LOW);  
  analogWrite(5, n);  
  digitalWrite(3, LOW);  
  //leds  
  digitalWrite(7, LOW);  
  digitalWrite(4, HIGH);  
  digitalWrite(2, LOW);  
  
}
```

DIAGRAMA DEL CIRCUITO

