

Tecnicatura Superior en Telecomunicaciones



Electronica Microcontrolada

Profesor: C. GONZALO VERA

Profesor: JORGE E. MORALES

Tema: Proyecto de Desarrollo Auto

Ciclo lectivo: 2022

Grupo 5

En este documento primero se presentarán los objetivos del proyecto así como las diferentes fases que se han seguido para su desarrollo e implementación. Haciendo referencia a los objetivos, se describirá con detalle todo aquello que se pretende desarrollar justificando, de igual manera, las razones de su tipo de elaboración así como de aspectos más específicos.

FASES DE DESARROLLO

La primera fase consiste en determinar un plan de propuesta de desarrollo y los objetivos que se deberán cumplir.

Una vez definidos los objetivos del proyecto, se establece una planificación estimada de las tareas y un presupuesto inicial; durante la realización del proyecto, tanto la planificación como el presupuesto fijados se podrán ver alterados debido a un imprevisto o a las modificaciones que puedan surgir.

La tercera fase consta de dos partes primero conocer las necesidades que quisiéramos cubrir, y posteriormente la elaboración de un plan de requisitos en el que tener en cuenta, tanto los requisitos impuestos por nosotros, como los requisitos que el sistema deberá cumplir.

Una vez finalizada la fase de análisis, se procede al diseño del sistema, en el que se definirá la arquitectura tanto hardware como software de un modo esquemático, para su posterior implementación.

La implementación es la fase de programación del software a nivel de código, y de la construcción de todas las componentes para formar una estructura hardware. Esta tarea es la que conllevará un mayor número de horas.

Finalmente, una vez implementado el sistema, se realiza una última fase de pruebas, donde se tendrán en cuenta los requisitos establecidos y el resultado final de la implementación, para realizar todas las pruebas y verificar que el sistema funciona de forma correcta.

OBJETIVOS DEL PROYECTO

El principal objetivo del proyecto es controlar un vehículo a escala de manera que pueda utilizar un sensor de ultrasonido y realizar movimientos para evitar obstáculos, y que además pueda desplazarse de manera autónoma mediante un simple algoritmo. Para poder llevar a cabo el objetivo principal, será necesario cumplir una serie de sub-objetivos que se muestran a continuación:

- Diseño y construcción mecánica de una estructura que simule un vehículo.

La estructura deberá contener lo básico para realizar las funcionalidades de un vehículo prototipado.

- Implementación de un software para un controlador que controle los posibles movimientos del vehículo.

Para que el vehículo se pueda desplazar, son necesarias dos partes fundamentales: un controlador y un software que maneje el vehículo.

- ~~• Desarrollo de una aplicación para teléfono móvil que permita al usuario el manejo del vehículo.~~
- Creación de un modo manual y automático.

El modo manual será el modo en el que el usuario controle el vehículo, y el modo automático en el que se realice el movimiento autónomo. Este último además incluirá un sistema anti-colisiones frontales.

Por lo tanto, este proyecto consta de dos grandes bloques, cuya implementación y diseño podrán realizarse por separado, pero finalmente deben funcionar en conjunto:

- Hardware del sistema: incluye todas las componentes que forman la estructura del vehículo y todos los controladores y chips que pueda contener.
- Software del sistema: Programa cargado en el controlador.

DESCRIPCION DE HERRAMIENTAS Y MATERIALES

A continuación describiremos y daremos los datos técnicos del material que utilizaremos para esta fase.

Arduino

En la página oficial de Arduino, <http://www.arduino.cc/es/>.

“Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).



- Dos ruedas



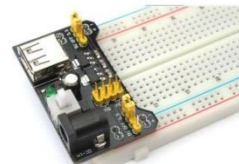
- 1 Rueda loca



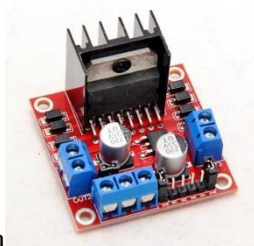
- Cables



- Fuente de 5v



- 1 Driver L298n



- Modulo-bluetooth-hc-06



SOFTWARE/ENTORNO DE PROGRAMACION

Editor de código Visual Studio Code con PlatformIO IDE y librería de Arduino.
También podríamos utilizar Arduino Uno para dicho desarrollo de programación, pero elegimos PlatformIO ya que venimos trabajando con este IDE durante la cursada realizando pequeños trabajos y simulaciones.

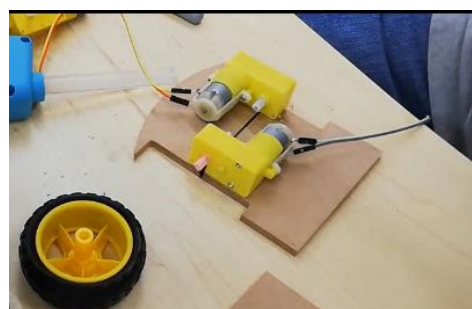
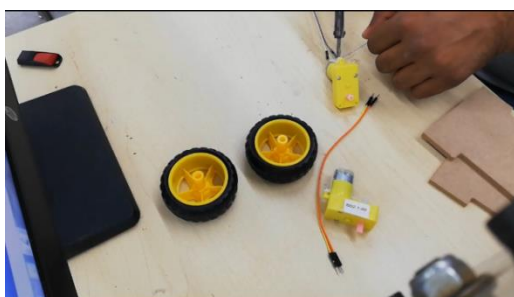
Arduino
El proceso de compilado y enlace del programa se hace cada que se cambia el código, así como el proceso de flasheado
La compilación del programa puede demorar más entre más librerías se incluyan
El archivo de programa se compila a lenguaje máquina, lo que lo hace más eficiente, pero menos portable

Una vez en el desarrollo del programa definiremos los puertos por los cuales van a trabajar los dos motores, sabiendo que la tarjeta trae 4 puertos donde podremos utilizar el puerto 1 para el motor de la derecha y el puerto 3 para el motor de la izquierda; para el sensor de distancia vamos a utilizar el pin 13 como disparo y el pin 2 como ECO. Luego definiremos variables para calcular la duración y la distancia que nos encontramos de los objetos, en el LOOP activamos el sensor y dependiendo de la distancia a los objetos efectuaremos el giro de las ruedas ya sea para cambiar su dirección o seguir su camino sino encuentra ningún objeto. Para esto nos ayudaremos con un condicional IF donde detectara si no hay un objeto a 10 cm continuara su camino o en caso contrario ejecutara la detención de los motores, activándolos a mitad de potencia y haciendo girar la rueda izquierda hacia atrás y la rueda derecha hacia adelante; saldrá de ese bucle y se volverá a disparar el sensor y medir la distancia, evaluara si es mayor o menor a esos 10 cm, si es menor que siga girando, girando y girando hasta que supere los 10 cm.

PUESTA EN PRÁCTICA: (DESARROLLO)

Ensamblado de Auto:

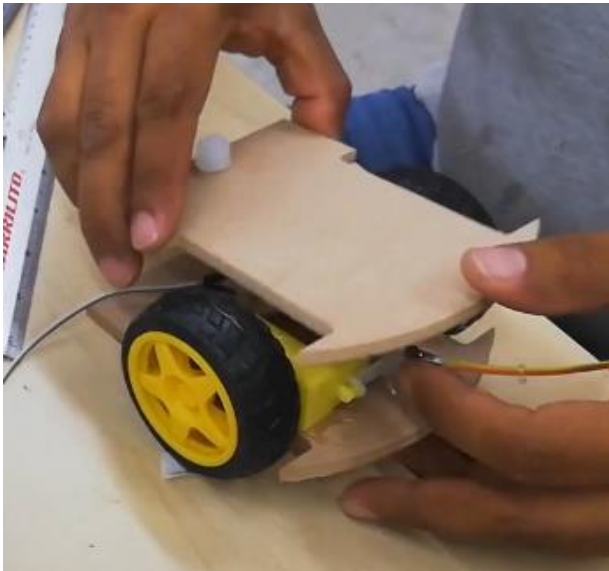
Soldamos los Motores con sus cables.



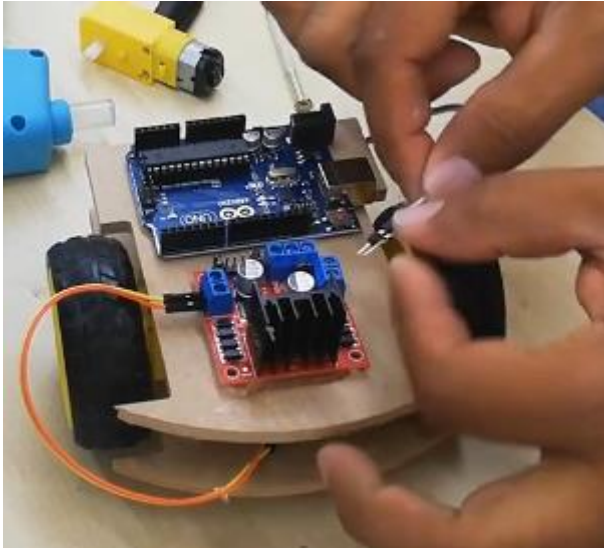
Colocamos la rueda loca



Unimos la parte superior



Colocamos y hacemos las conecciones del modulo arduino con el Driver L298n



Para los dos motores asignamos de la siguiente forma:

Motor1: Pines 4, 5 y 6 (pwm)

Motor2: Pines 7, 8 y 9 (pwm)

Los pines de arduino y el modulo son:

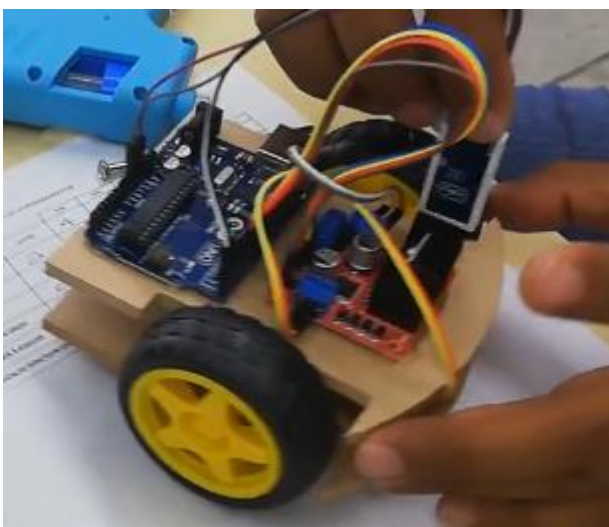
Pin 0 : RX con TX del modulo BT

Pin 1: TX con RX del mosulo BT

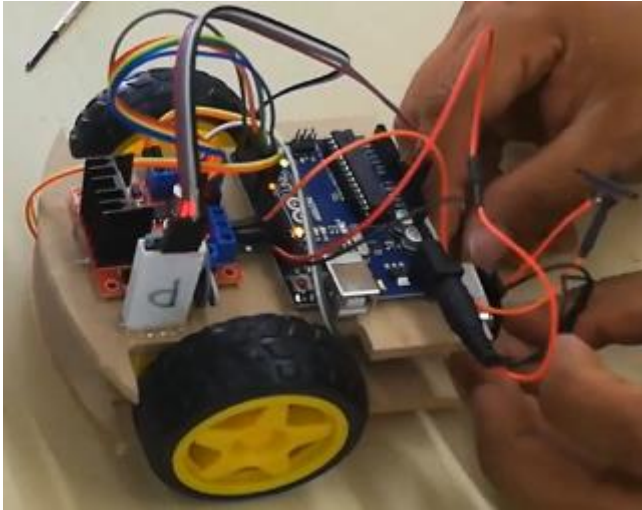
3.3V con VCC del modulo

GNG:GNG

Colocamos el modulo Bluetooth

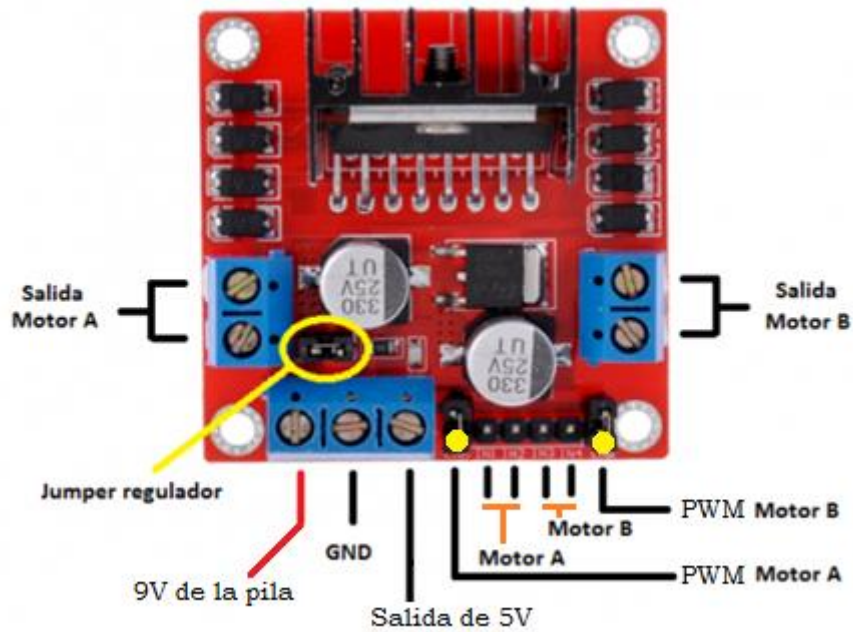


Y la fuente de alimentacion



Puente H L298N

Diagrama de conexiones



PWM Motor A y B: es el pin de control de velocidad. Es necesario retirar los jumpers de estos pines para usarlos.

Motor A y B: son los pines donde conectaremos los pines de control de Arduino

Salida A y B: son los conectores donde se conectan los motores

9V de la pila: aquí conectamos el positivo de la batería

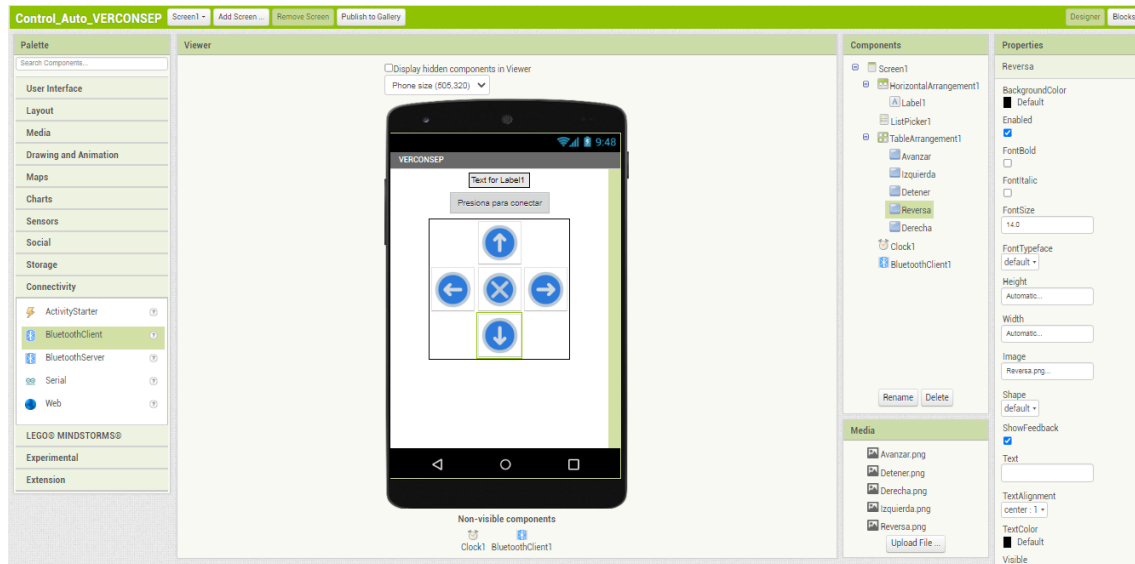
GND: aquí se conecta el negativo de la pila y el negativo de Arduino

Salida de 5V: aquí tienes una salida de 5 voltios para conectar lo que necesites

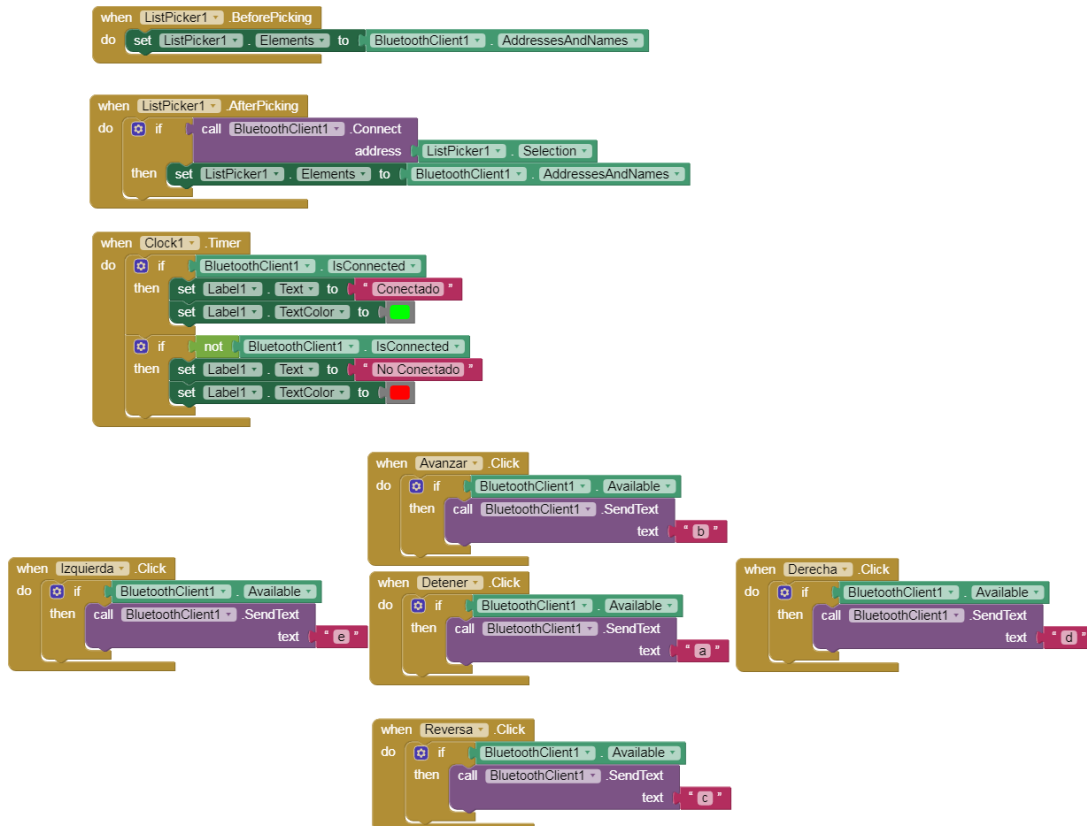
Jumper regulador: Este no se deberá de retirar si quieres tener activa la salida de 5V

Desarrollamos la aplicación de control en MIT App Inventor

DISEÑO:



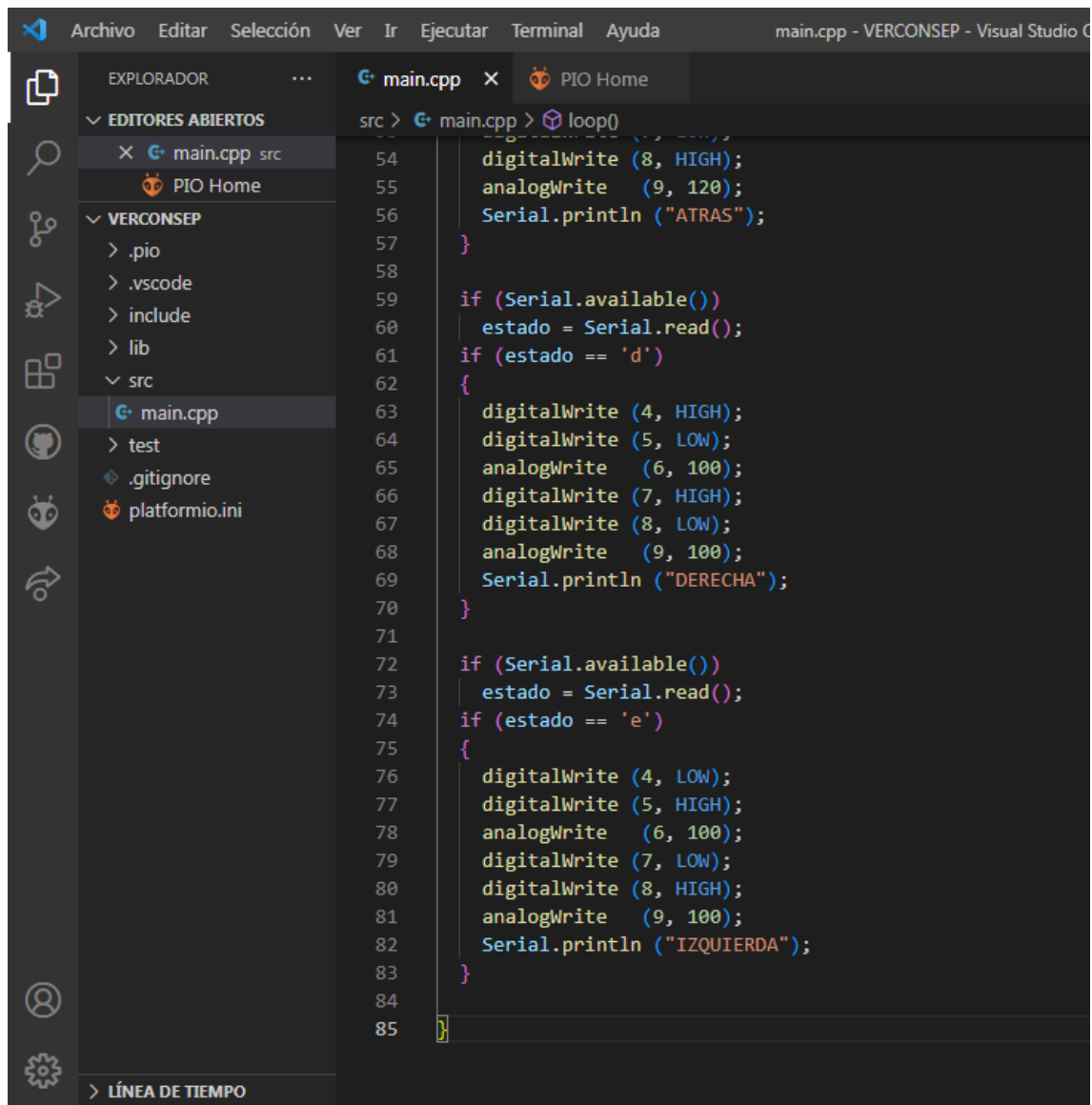
Bloques, programación de eventos:



Programación del Arduino:


```
src > main.cpp > ...
1  #include <Arduino.h>
2  #include <SoftwareSerial.h>
3
4  char estado;
5
6  void setup() {
7      pinMode(4, OUTPUT);
8      pinMode(5, OUTPUT); //MOTOR 1
9      pinMode(6, OUTPUT); //Velocidad
10
11      pinMode(7, OUTPUT);
12      pinMode(8, OUTPUT); //MOTOR 2
13      pinMode(9, OUTPUT); //Velocidad
14
15      Serial.begin(9600);
16
17  }
18
19  void loop() {
20      if (Serial.available())
21          estado = Serial.read();
22      if (estado == 'a') //DETENER
23      {
24          digitalWrite (4, LOW);
25          digitalWrite (5, LOW);
26          analogWrite  (6, 0);
27          digitalWrite (7, LOW);
28          digitalWrite (8, LOW);
29          analogWrite  (9, 0);
30          Serial.println ("DETENER");
31      }
32
33      if (Serial.available())
34          estado = Serial.read();
35      if (estado == 'h') //ADELANTE
```

```
src > main.cpp > loop()
33      if (Serial.available())
34          estado = Serial.read();
35      if (estado == 'b') //ADELANTE
36      {
37          digitalWrite (4, LOW);
38          digitalWrite (5, HIGH);
39          analogWrite  (6, 120);
40          digitalWrite (7, HIGH);
41          digitalWrite (8, LOW);
42          analogWrite  (9, 120);
43          Serial.println ("ADELANTE");
44      }
45
46      if (Serial.available())
47          estado = Serial.read();
48      if (estado == 'c')
49      {
50          digitalWrite (4, HIGH);
51          digitalWrite (5, LOW);
52          analogWrite  (6, 120);
53          digitalWrite (7, LOW);
54          digitalWrite (8, HIGH);
55          analogWrite  (9, 120);
56          Serial.println ("ATRAS");
57      }
58
59      if (Serial.available())
60          estado = Serial.read();
61      if (estado == 'd')
62      {
63          digitalWrite (4, HIGH);
64          digitalWrite (5, LOW);
65          analogWrite  (6, 100);
66          digitalWrite (7, HIGH);
```



```
54     digitalWrite (8, HIGH);
55     analogWrite  (9, 120);
56     Serial.println ("ATRAS");
57 }
58
59 if (Serial.available())
60     estado = Serial.read();
61 if (estado == 'd')
62 {
63     digitalWrite (4, HIGH);
64     digitalWrite (5, LOW);
65     analogWrite  (6, 100);
66     digitalWrite (7, HIGH);
67     digitalWrite (8, LOW);
68     analogWrite  (9, 100);
69     Serial.println ("DERECHA");
70 }
71
72 if (Serial.available())
73     estado = Serial.read();
74 if (estado == 'e')
75 {
76     digitalWrite (4, LOW);
77     digitalWrite (5, HIGH);
78     analogWrite  (6, 100);
79     digitalWrite (7, LOW);
80     digitalWrite (8, HIGH);
81     analogWrite  (9, 100);
82     Serial.println ("IZQUIERDA");
83 }
84
85 }
```

Escaneamos el código y realizamos la conexión Bluetooth junto con las pruebas de funcionamiento (en el video se pude apreciar más detalladamente este paso)