



Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

Sistemas Operativos Avanzados

Trabajo Práctico Sistemas Embebidos Android

Control Remoto - RC

Año 2016 – 2 Cuatrimestre

Integrantes:

Bartellone, Matias
Maulella, Pablo

DNI 37.123.993
DNI 34.374.330

Índice

Objetivo	3
Tareas realizadas	4
Información técnica	5
Sketch utilizado	10
Armado de la Aplicación Android para la comunicación con Arduino	11
Botones Principales	12

Objetivo

Realizar un auto con un sistema de control remoto, a partir de un sistema embebido compuesto por módulo de bluetooth, conectado a la placa Arduino y el mismo junto a un Puente H que controlar los motores del auto.

Hardware utilizado

- Placa Arduino UNO
- Modulo bluetooth HC-06 (Esclavo)
- Puente H
- LED's
- Cables macho – macho/ Macho- Hembra
- Protoboard 400 puntos
- Baterías 9v
- Interruptor
- Conexión externa alimentación Arduino a 9v
- Carcasa Auto Tyco 6x6

Software utilizado

- Arduino 1.6.11
- Android Studio 2.2

Videos Youtube

Arduino LED - Android - Bluetooth HC-06

- <https://www.youtube.com/watch?v=eqORIPH5X3E&feature=youtu.be>

Prueba Control Remoto RC – Bluetooth – Android

- <https://www.youtube.com/watch?v=X51dZgEghoQ&feature=youtu.be>

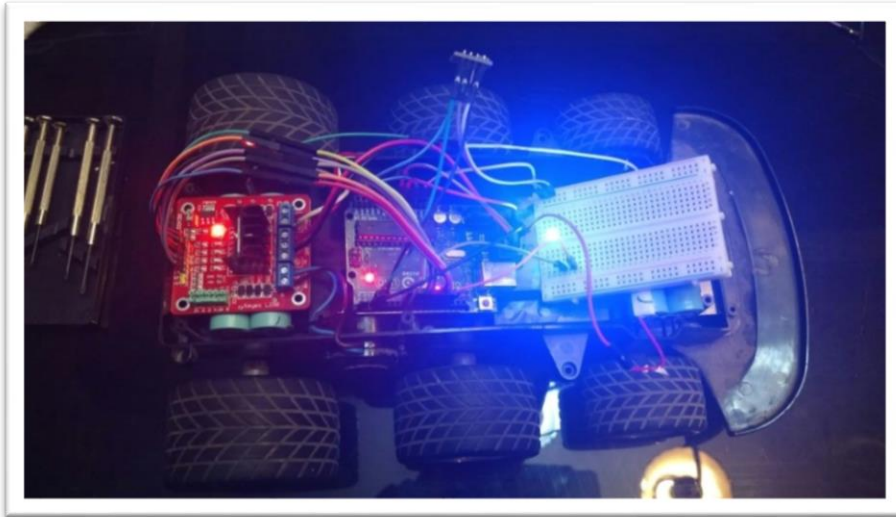
Código

<https://drive.google.com/open?id=0BxyttFY4yEd-SU5MT0xYUzJkREU>

Tareas realizadas

Armado de circuito y pruebas.

1. Iniciamos la conexión del Puente H con la placa Arduino.
2. Establecimos los circuitos que alimentan al puente H para poder controlar los motores.
3. Conectamos el módulo bluetooth para la recepción de los mensajes a la placa.



Auto Cerrado y Conectado.



Sketch utilizado

```
#include <SoftwareSerial.h>

#define IR 4
#define RE 12
#define ST 5
#define DE 3
#define DEBUG true

SoftwareSerial HC06(0,1);

int ENA=11;
int IN1=9;
int IN2=8;
int ENB=10;
int IN3=7;
int IN4=6;
int detection = HIGH;

void derecha(int val){
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(ST,LOW);
  analogWrite(ENA,val);
  analogWrite(ENB,val);
}

void izquierda(int val){
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN4,HIGH);
  digitalWrite(ST,LOW);
  analogWrite(ENA,val);
  analogWrite(ENB,val);
}
```



```

}

void adelante(int val){
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(IN2,HIGH);
    digitalWrite(IN1,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
    digitalWrite(ST,LOW);
    analogWrite(ENA,val);
    analogWrite(ENB,val);
}

void parar(){
    analogWrite(ENA,0);
    analogWrite(ENB,0);
    digitalWrite(ENA,LOW);
    digitalWrite(ENB,LOW);
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,LOW);
    digitalWrite(ST,HIGH);
    digitalWrite(RE,LOW);
}

boolean hayObstaculo(){
    detection = digitalRead(IR);
    if(detection == LOW){
        Serial.println("Hay un obstaculo!\n");
        return true;
    }
    else{
        Serial.println("No hay obstaculo!\n");
        return false;
    }
}

```

```

void atras(int val){
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN1,HIGH);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,HIGH);
    digitalWrite(ST,LOW);
    digitalWrite(RE,HIGH);
    analogWrite(ENA,val);
    analogWrite(ENB,val);
}

void setup()
{
    Serial.begin(9600);
    HC06.begin(9600); // your esp's baud rate might be different
    pinMode(ENA,OUTPUT);//output
    pinMode(ENB,OUTPUT);
    pinMode(DE,OUTPUT);
    pinMode(ST,OUTPUT);
    pinMode(RE,OUTPUT);
    pinMode(IN1,OUTPUT);
    pinMode(IN2,OUTPUT);
    pinMode(IN3,OUTPUT);
    pinMode(IN4,OUTPUT);
    pinMode(IR,INPUT);
    pinMode(12,OUTPUT);
    pinMode(13,OUTPUT);
    digitalWrite(ENA,LOW);
    digitalWrite(ENB,LOW);//stop driving
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);//setting motorA's directon
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,LOW);//setting motorB's directon
    analogWrite(ENA,255);//start driving motorA

```

```

    analogWrite(ENB,255);
}

char msg;

int vel=100;

boolean luz=false;

void loop()
{
    if(HC06.available()) // check if the esp is sending a message
    {
        digitalWrite(13,HIGH);
        msg=(HC06.read());
        Serial.println("Mensaje recibido: "+msg);
        if(msg == '1'){
            Serial.println("Avanzando: "+vel);
            adelante(vel);
        }
        if(msg == '2'){
            atras(vel);
            Serial.println("Retrocediendo: "+vel);
        }
        if(msg == '3'){
            Serial.println("Detenido.");
            parar();
        }
        if(msg == '4'){
            Serial.println("Doblando a la izquierda: "+vel);
            izquierda(vel);
        }
        if(msg == '5'){
            Serial.println("Doblando a la derecha: "+vel);
            derecha(vel);
        }
        if(msg == '6'){
            vel=100;
        }
    }
}

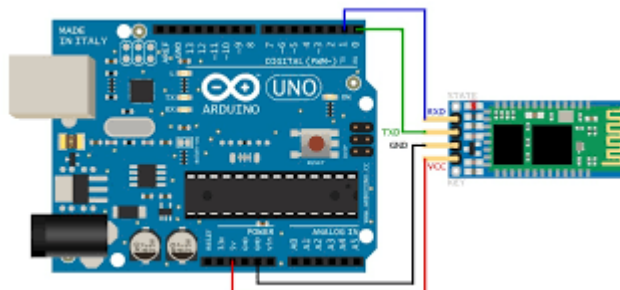
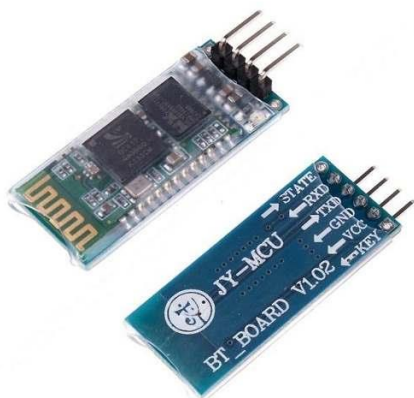
```



```
if(msg == '7'){  
    vel=150;  
}  
if(msg == '8'){  
    vel=200;  
}  
if(msg == '9'){  
    vel=255;  
}  
if(msg == '0'){  
    if (luz){  
        digitalWrite(DE,LOW);  
        luz = false;  
    }else{  
        digitalWrite(DE,HIGH);  
        luz = true;  
    }  
}  
}  
}
```

Información técnica

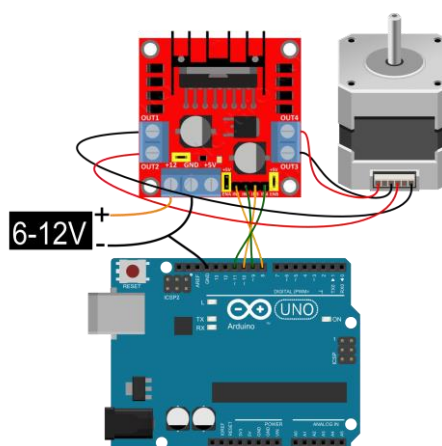
Conexión Modulo Bluetooth HC-06



El modelo HC-06 dispone de 4 pines y solo puede actuar como esclavo.

En la imagen anterior se muestra la conexión básica para el funcionamiento del modulo

Conexión Puente H L298



Este módulo basado en el chip L298N te permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.

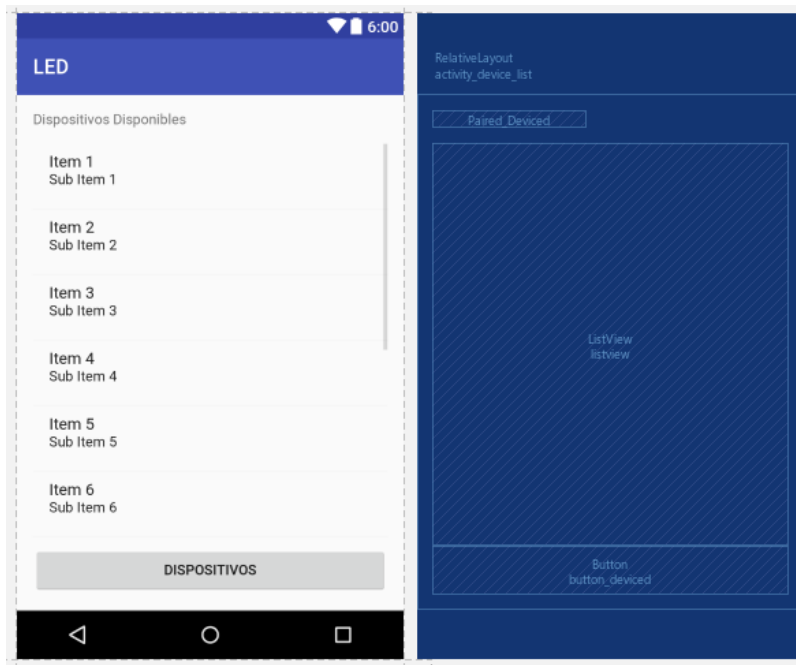
El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4.

Importante: Los pines de habilitación son ENA y ENB respectivamente. Se conectan a los pines PWM, de la placa para simular los cambios de velocidades del auto.

Armado de la Aplicación Android para la comunicación con Arduino

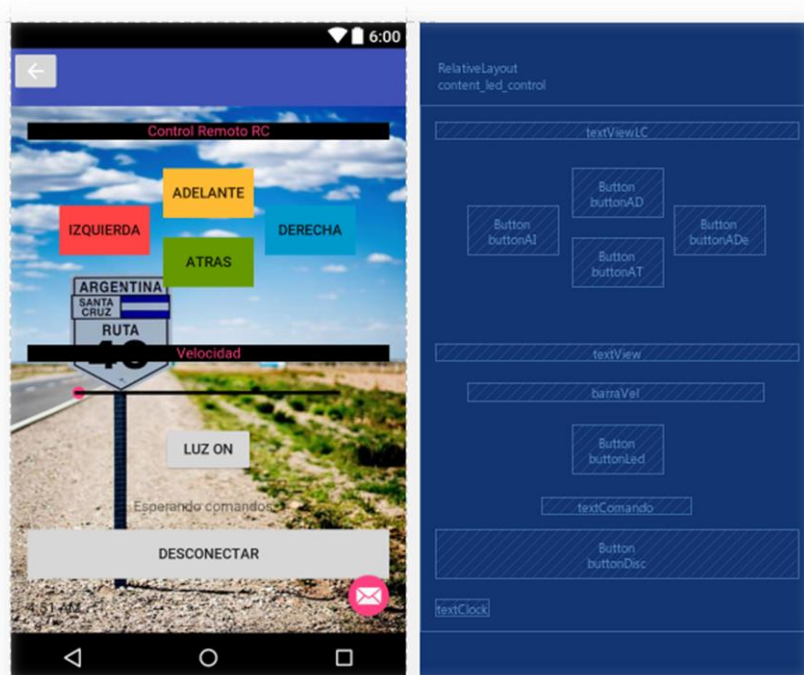
Activiy - Lista de Dispositivos

La siguiente Activity fue desarrollada para la selección de dispositivos bluetooth emparejados con el dispositivo Android.



Activity – Control Principal

Es la pantalla principal para controlar el auto de forma remota.



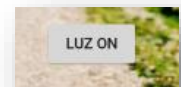
Botones Principales

Consta de 4 botones principales, los mismos fueron configurados con el evento *setOnTouchListener* que permite realizar una acción mientras se mantenga pulsado el botón.



- **Adelante:** Envía un mensaje al Arduino para que active los motores en dirección hacia adelante mientras se mantenga apretado, cuando suelta se libera el botón y se detiene.
- **Atrás:** Envía un mensaje al Arduino para que active los motores en dirección hacia atrás mientras se mantenga apretado, cuando suelta se libera el botón y se detiene.
- **Izquierda:** Envía un mensaje al Arduino para que active un motor con dirección hacia adelante y el otro en sentido opuesto para que gire a la izquierda mientras se mantenga apretado, cuando suelta se libera el botón y se detiene.
- **Derecha:** Envía un mensaje al Arduino para que active un motor con dirección hacia atrás y el otro en sentido opuesto para que gire a la derecha mientras se mantenga apretado, cuando suelta se libera el botón y se detiene.

Luz On/Off : Se configuro el evento *setOnClickListener* que permite encender los led delantero del auto y una vez presionado, cambia el estado a Led Off si esta encendido y viceversa en el caso contrario.



Velocidad: Se configuro el Componente *SeekBar* que modifica la velocidad que impulsa a los motores ya sea a la izquierda o derecha.



Desconectar: Este botón realiza la desconexión de la comunicación entre Arduino y la app Android.

