

## **DISPOSITIVOS MOVILES**

**PROYECTO:** Comunicación Android-Arduino vía Bluetooth.

### **INTRODUCCIÓN.**

Diseñar una aplicación móvil desde Android que permita encender los LEDs de una tarjeta Arduino, vía una comunicación Bluetooth. Los LEDs poseen una configuración equivalente desde la aplicación, Adelante, Izquierda, Centro, Derecha y Atrás.

### **DESARROLLO.**

Capturar el código Java y XML en un proyecto del IDE Android Studio con la plantilla Empty Views Activity.

El código de Arduino es LEDs.ino. Al final de este documento se indican los pines de conexión del Arduino, HC-05 y los LEDs.

Enseguida se muestran los códigos correspondientes a los archivos Java y XML. Copiar cada código a su correspondiente archivo.

#### **MainActivity.java**

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import java.util.UUID;
import android.bluetooth.*;
import android.os.*;
import android.view.View;
import android.widget.*;
import java.io.*;

public class MainActivity extends AppCompatActivity {
    EditText edtTextoOut;
    ImageButton btnEnviar, btnAdelante, btnIzquierda, btnStop, btnDerecha, btnReversa;
    TextView tvtMensaje;
    Button btnDesconectar;
    Handler bluetoothIn;
    final int handlerState = 0;
    private BluetoothAdapter btAdapter = null;
    private BluetoothSocket btSocket = null;
    private StringBuilder DataStringIN = new StringBuilder();
    private ConnectedThread MyConexionBT;
    // Identificador único de servicio - SPP UUID
    private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    // String para la dirección MAC
    private static String address = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bluetoothIn = new Handler() {
            public void handleMessage(android.os.Message msg) {
                if(msg.what == handlerState) {
                    char MyCaracter = (char) msg.obj;
                    if(MyCaracter == 'a') {
                        tvtMensaje.setText("ACELERANDO");
                    }
                    if(MyCaracter == 'i') {
                        tvtMensaje.setText("GIRO IZQUIERDA");
                    }
                    if(MyCaracter == 'd') {
                        tvtMensaje.setText("GIRO DERECHA");
                    }
                    if(MyCaracter == 'r') {
                        tvtMensaje.setText("RETROCEDIENDO");
                    }
                }
            }
        };
    }
}
```

```

        if(MyCaracter == 's'){
            tvtMensaje.setText("DETENIDO");
        }
    }
};

btAdapter = BluetoothAdapter.getDefaultAdapter();
verificarEstadoBT();
edtTextoOut = findViewById(R.id.edtTextoOut);
btnEnviar = findViewById(R.id.btnEnviar);
btnAdelante = findViewById(R.id.btnAdelante);
btnIzquierda = findViewById(R.id.btnIzquierda);
btnStop = findViewById(R.id.btnStop);
btnDerecha = findViewById(R.id.btnDerecha);
btnReversa = findViewById(R.id.btnReversa);
tvMensaje = findViewById(R.id.tvMensaje);
btnDesconectar = findViewById(R.id.btnDesconectar);

btnEnviar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {    // GetDat es cualquier dato String
        String GetDat = edtTextoOut.getText().toString();
        MyConexionBT.write(GetDat); // ENVIA GetDat AL PUERTO SERIE
    }
});

btnAdelante.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MyConexionBT.write("A");
    }
});

btnIzquierda.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MyConexionBT.write("I");
    }
});

btnStop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MyConexionBT.write("S");
    }
});

btnDerecha.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MyConexionBT.write("D");
    }
});

btnReversa.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MyConexionBT.write("R");
    }
});

btnDesconectar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(btSocket!=null){
            try{btSocket.close();}
            catch(IOException e){
                Toast.makeText(getApplicationContext(),
                    "ERROR",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
});

```

```

        }
        finish();
    }
});
}
private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws
IOException{
    // Crea una conexion de salida segura para el dispositivo usando el servicio UUID
    return device.createRfcommSocketToServiceRecord(BTMODULEUUID);
}
@Override
public void onResume(){
    super.onResume();
    Intent intent = getIntent();
    address = intent.getStringExtra(DispositivosVinculados.EXTRA_DEVICE_ADDRESS);
    BluetoothDevice device = btAdapter.getRemoteDevice(address); // Setea la direccion
MAC
    try{
        btSocket = createBluetoothSocket(device);
    } catch(IOException e){
        Toast.makeText(getBaseContext(), "La creaci3n del Socket fall3",
Toast.LENGTH_LONG).show();
    }
    try{ // Establece la conexion con el socket Bluetooth.
        btSocket.connect();
    } catch(IOException e){
        try{
            btSocket.close();
        } catch(IOException e2){}
    }
    MyConexionBT = new ConnectedThread(btSocket);
    MyConexionBT.start();
}
@Override
public void onPause(){
    super.onPause();
    try{ // Cuando se sale de la aplicaci3n esta parte permite que no se deje abierto
el socket
        btSocket.close();
    } catch(IOException e2){}
}

private void verificarEstadoBT(){
    if(btAdapter == null){ // Comprueba que el Bluetooth est3 disponible y solicita se
active si est3 desactivado
        Toast.makeText(getBaseContext(), "El dispositivo no soporta bluetooth",
Toast.LENGTH_LONG).show();
    } else{
        if(btAdapter.isEnabled()){
        } else{
            Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}

private class ConnectedThread extends Thread{ // Clase que permite crear el evento de
conexion
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    public ConnectedThread(BluetoothSocket socket){
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

```

```

        try{
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch(IOException e){}
        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run(){
        byte[] byte_in = new byte[1];
        while(true){ // Se mantiene en modo escucha para determinar el ingreso de datos
            try{
                mmInStream.read(byte_in);
                char ch = (char) byte_in[0];
                bluetoothIn.obtainMessage(handlerState, ch).sendToTarget();
            } catch(IOException e){
                break;
            }
        }
    }
    public void write(String input){ // Envío de trama
        try{
            mmOutStream.write(input.getBytes());
        } catch(IOException e){ // si no es posible enviar datos se cierra la conexión
            Toast.makeText(getApplicationContext(), "La conexión falló",
                Toast.LENGTH_LONG).show();
        }
    }
}
}
}

```

#### activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnDesconectar"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="#E8E8E8"
        android:text="@string/strdesc"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/tvtMensaje"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="#340BCD4"
        android:hint="@string/strResp"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/btnDesconectar"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <EditText
        android:id="@+id/edtTexttoOut"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

```

```

        android:ems="10"
        android:hint="@string/strescriba"
        android:inputType="text"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/btnEnviar"
        app:layout_constraintEnd_toStartOf="@+id/btnEnviar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
<ImageButton
    android:id="@+id/btnEnviar"
    android:layout_width="100dp"
    android:layout_height="40dp"
    android:background="#FFFFFF"
    android:backgroundTint="#548BC34A"
    android:contentDescription="@string/strconten"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/enter" />
<ImageButton
    android:id="@+id/btnAdelante"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#FFFFFF"
    android:contentDescription="@string/strconten"
    android:contextClickable="false"
    android:scaleType="fitStart"
    app:layout_constraintBottom_toTopOf="@+id/btnStop"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edtTextoOut"
    app:srcCompat="@drawable/arriba" />
<ImageButton
    android:id="@+id/btnIzquierda"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#FFFFFF"
    android:contentDescription="@string/strconten"
    android:scaleType="fitStart"
    app:layout_constraintBottom_toBottomOf="@+id/btnStop"
    app:layout_constraintEnd_toStartOf="@+id/btnStop"
    app:layout_constraintStart_toStartOf="parent"
    app:srcCompat="@drawable/izquierda" />
<ImageButton
    android:id="@+id/btnStop"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#FFFFFF"
    android:contentDescription="@string/strconten"
    android:scaleType="fitStart"
    app:layout_constraintBottom_toTopOf="@+id/tvtMensaje"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edtTextoOut"
    app:srcCompat="@drawable/centro" />
<ImageButton
    android:id="@+id/btnDerecha"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#FFFFFF"
    android:contentDescription="@string/strconten"
    android:scaleType="fitStart"
    app:layout_constraintBottom_toBottomOf="@+id/btnStop"

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/btnStop"
        app:srcCompat="@drawable/derecha" />
<ImageButton
    android:id="@+id/btnReversa"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:background="#FFFFFF"
    android:contentDescription="@string/strconten"
    android:scaleType="fitStart"
    app:layout_constraintBottom_toTopOf="@+id/tvtMensaje"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnStop"
    app:layout_constraintVertical_bias="0.521"
    app:srcCompat="@drawable/abajo" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

### DispositivosVinculados.java

```

import androidx.appcompat.app.AppCompatActivity;
import android.bluetooth.*;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.*;
import java.util.Set;
public class DispositivosVinculados extends AppCompatActivity {
    // Depuración de LOGCAT
    private static final String TAG = "DispositivosVinculados";
    // Declaración de ListView
    ListView IdLista;
    // String que se enviará a la actividad principal, MainActivity
    public static String EXTRA_DEVICE_ADDRESS = "device_address";
    private BluetoothAdapter mBtAdapter;
    private ArrayAdapter mPairedDevicesArrayAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dispositivos_vinculados);
    }
    @Override
    public void onResume() {
        super.onResume();
        verificarEstadoBT(); // Inicializa la array de la lista de los dispositivos
        vinculados.
        mPairedDevicesArrayAdapter = new ArrayAdapter(this,
        R.layout.dispositivos_encontrados);
        IdLista = (ListView) findViewById(R.id.IdLista);
        IdLista.setAdapter(mPairedDevicesArrayAdapter);
        IdLista.setOnItemClickListener(mDeviceClickListener);
        mBtAdapter = BluetoothAdapter.getDefaultAdapter();// Obtiene el adaptador local
        Bluetooth adapter
        Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();
        if(pairedDevices.size() > 0){ // Agrega un dispositivo emparejado al array
            for(BluetoothDevice device : pairedDevices){
                mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
                device.getAddress());
            }
        }
        // Configura un(on-click) para la lista
    }
}

```

```

        private AdapterView.OnItemClickListener mDeviceClickListener = new
AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView av, View v, int arg2, long arg3){
        // Obtener la direccion MAC del dispositivo, que son los últimos 17 caracteres
en la vista
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);
        finishAffinity();
        // Realiza un intent para iniciar la siguiente actividad
        // mientras toma un EXTRA_DEVICE_ADDRESS que es la dirección MAC.
        Intent intend = new Intent(DispositivosVinculados.this, MainActivity.class);
        intend.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(intend);
    }
};

private void verificarEstadoBT(){ // Comprueba que el dispositivo tiene Bluetooth y que
está encendido.
    mBtAdapter = BluetoothAdapter.getDefaultAdapter();
    if(mBtAdapter==null){
        Toast.makeText(getApplicationContext(), "El dispositivo no soporta Bluetooth",
Toast.LENGTH_SHORT).show();
    } else{
        if(mBtAdapter.isEnabled()){
            Log.d(TAG, "...Bluetooth activado ...");
        } else{ // Solicita al usuario que active Bluetooth
            Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}
}
}

```

### activity\_dispositivos\_vinculados.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".DispositivosVinculados">
    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="DISPOSITIVOS VINCULADOS"
        android:textAlignment="center"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <ListView
        android:id="@+id/IdLista"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:paddingStart="20dp"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintTop_toBottomOf="@+id/textView" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

### dispositivos\_encontrados.xml

```

<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</TextView>

```

### strings.xml

```

<resources>
    <string name="app_name">ControlRemoto3</string>
    <string name="strconten">IMAGEN</string>
    <string name="strdesc">DESCONECTAR</string>
    <string name="strescriba">Escriba un texto</string>
    <string name="strResp">Respuesta</string>
</resources>

```

### AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.ControlRemoto3"
        tools:targetApi="31">
        <activity
            android:name=".DispositivosVinculados"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

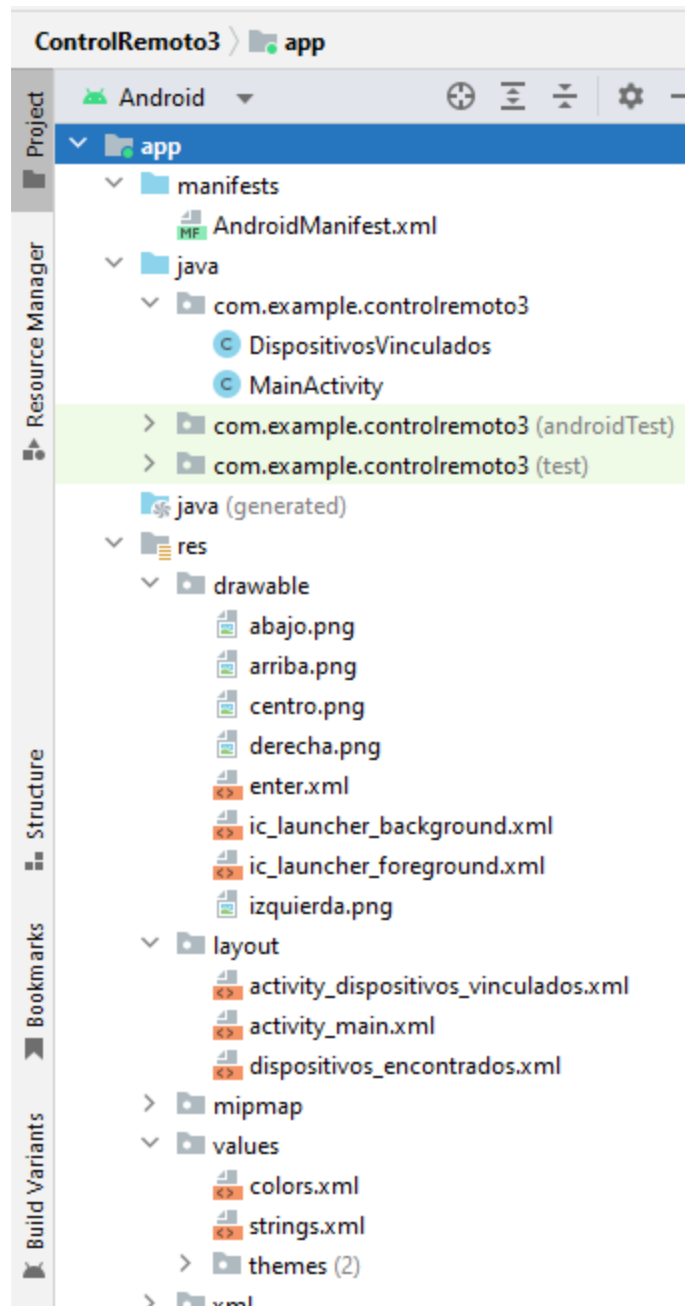
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="false" />
    </application>
</manifest>

```

La figura 1 muestra el repositorio con el contenido de las carpetas y sus correspondientes archivos en el IDE de Android Studio.

Una vez que se compila sin errores y se ejecuta la aplicación, se debe seleccionar el dispositivo Bluetooth vinculado al móvil, en este caso el HC-05, como se indica en la figura 2. Una vez establecida la comunicación se muestra la pantalla con los cinco botones de orientación para el encendido de los LEDs. En la pantalla del móvil digitar un botón y verificar el encendido de su LED correspondiente en el módulo Arduino.

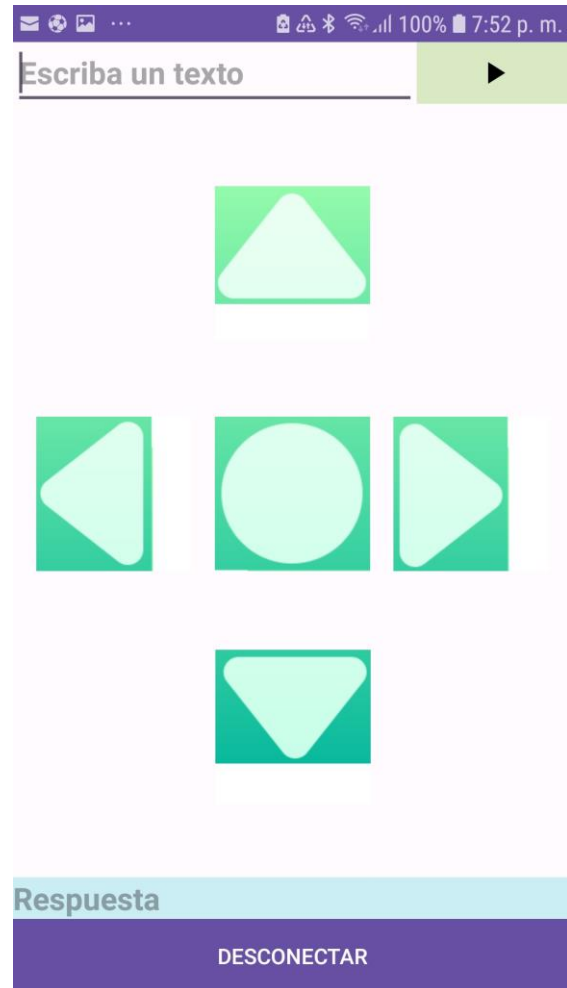




**Figura 1.** Repositorio de carpetas del proyecto, por ejemplo, controlremoto3.



**Figura 2.** Dispositivos vinculados.



**Figura 3.** Los controles de vista principal.

Enseguida se muestra el código en lenguaje C para el archivo correspondiente en el Arduino IDE.

**NOTA IMPORTANTE.** Para la realización de este proyecto se debe utilizar el módulo Arduino Pro **Mini** o Pro **Micro** y documentar, paso a paso, su programación según la opción elegida.

#### ARDUINO - LEDS.ino

```
#include<SoftwareSerial.h>
SoftwareSerial BT(10, 11); // 10 RX, 11 TX.

byte Led_1 = 14;
byte Led_2 = 15;
byte Led_3 = 16;
byte Led_4 = 17;
byte Led_5 = 18;

char DataBluetooth = ' ';

void setup() {
  Serial.begin(9600);
  BT.begin(9600);
  Serial.println("");
  Serial.println("Iniciando Control ...");

  pinMode(Led_1, OUTPUT);
  pinMode(Led_2, OUTPUT);
  pinMode(Led_3, OUTPUT);
```

```

pinMode(Led_4, OUTPUT);
pinMode(Led_5, OUTPUT);

digitalWrite(Led_1, LOW);
digitalWrite(Led_2, LOW);
digitalWrite(Led_3, LOW);
digitalWrite(Led_4, LOW);
digitalWrite(Led_5, LOW);

void loop(){
  // Si hay datos disponibles en el módulo bluetooth HC-05
  if(BT.available()){
    DataBluetooth = BT.read();
    Serial.print(DataBluetooth);
    if(DataBluetooth == 'I'){
      digitalWrite(Led_1, HIGH);
      digitalWrite(Led_2, LOW);
      digitalWrite(Led_3, LOW);
      digitalWrite(Led_4, LOW);
      digitalWrite(Led_5, LOW);
      BT.println("i");
      Serial.println("");
    }
    if(DataBluetooth == 'A'){
      digitalWrite(Led_1, LOW);
      digitalWrite(Led_2, HIGH);
      digitalWrite(Led_3, LOW);
      digitalWrite(Led_4, LOW);
      digitalWrite(Led_5, LOW);
      BT.println("a");
      Serial.println("");
    }
    if(DataBluetooth == 'D'){
      digitalWrite(Led_1, LOW);
      digitalWrite(Led_2, LOW);
      digitalWrite(Led_3, HIGH);
      digitalWrite(Led_4, LOW);
      digitalWrite(Led_5, LOW);
      BT.println("d");
      Serial.println("");
    }

    if(DataBluetooth == 'R'){
      digitalWrite(Led_1, LOW);
      digitalWrite(Led_2, LOW);
      digitalWrite(Led_3, LOW);
      digitalWrite(Led_4, HIGH);
      digitalWrite(Led_5, LOW);
      BT.println("r");
      Serial.println("");
    }

    if(DataBluetooth == 'S'){
      digitalWrite(Led_1, LOW);
      digitalWrite(Led_2, LOW);
      digitalWrite(Led_3, LOW);
      digitalWrite(Led_4, LOW);
      digitalWrite(Led_5, HIGH);
      BT.println("s");
      Serial.println("");
    }
  }
}

```

La tabla 1 muestra los pines de conexión entre el módulo Arduino, el módulo Bluetooth y los LEDs.

P I N E S		
ARDUINO	HC-05 (HC-06)	LED (Display 7 seg)
5V	5V	
GND	GND	KC
10	TX	
11	RX	
A0	-	f
A1	-	a
A2	-	b
A3	-	d
A4	-	g

**Tabla 1.** Los pines de conexión del módulo Arduino.

**NOTA.** Generar un reporte en un archivo AlumnoProyectoESPECIAL.doc y enviarlo al sitio indicado por el profesor.