

MANUAL TÉCNICO

– GRUPO 8

ARDUINO

22 AGOSTO DE 2019

ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES 1

PRÁCTICA 2

En el siguiente manual se detalla la información técnica de la funcionalidad y estructura del carro a control remoto (aplicación de Android), utilizando el microcontrolador Arduino MEGA.



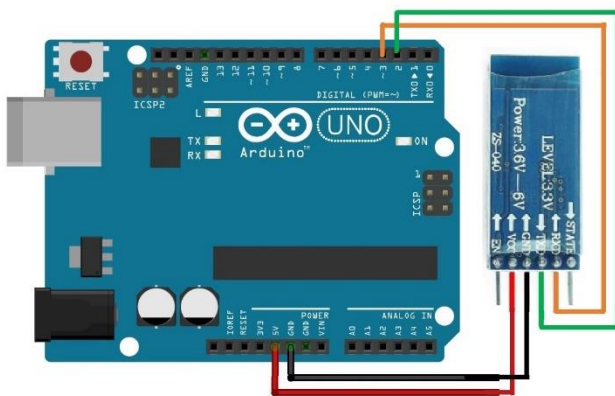
Arduino MEGA

FUNCIONALIDAD DE COMPONENTES:

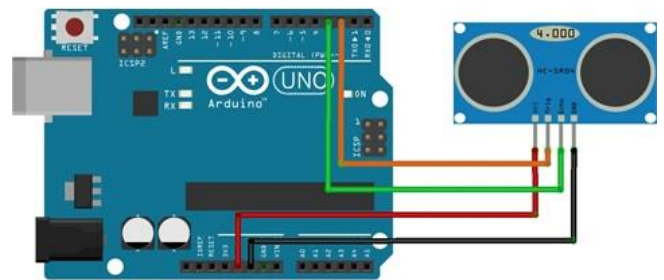
- **Aplicación para Android:** Controla los movimientos y el modo en el que se realizan.
- **Módulo Bluetooth HC-05:** Permite la conexión entre el teléfono y el Arduino.
- **Driver con Integrado L298N:** Se encarga del movimiento de las llantas para trasladar el vehículo.
- **Sensor ultrasónico HC-SR04:** Indica la proximidad a un objeto (muros o pared).
- **Sensor de color RGB TCS-230:** Detecta el color a determinada distancia, para realizar un movimiento en el modo automático. mecánico.
- **Puente H:** Se encarga de controlar el movimiento de los motores.

CONEXIÓN DE COMPONENTES:

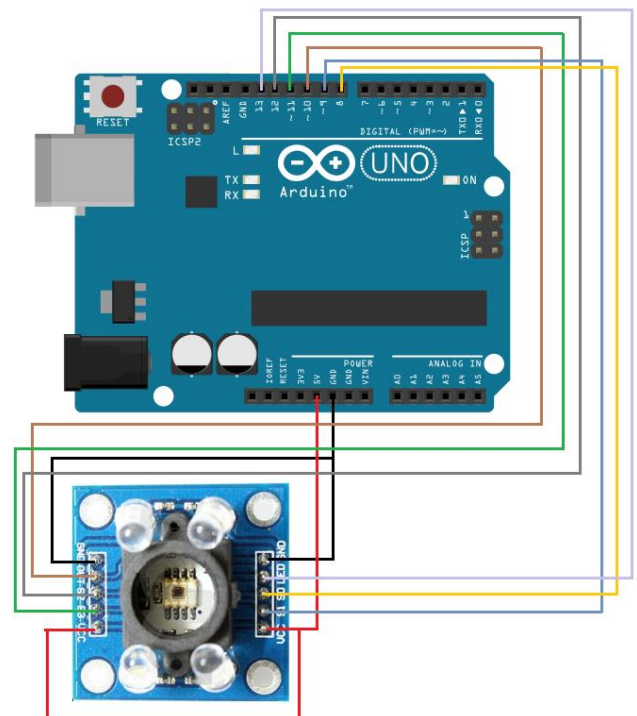
Módulo Bluetooth HC-05



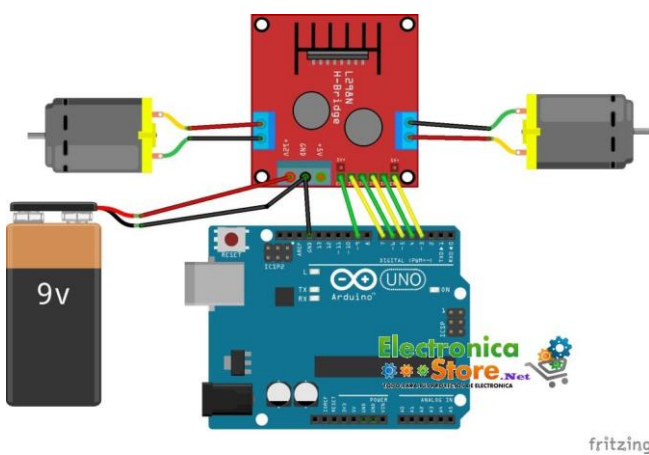
Sensor ultrasónico HC SR04



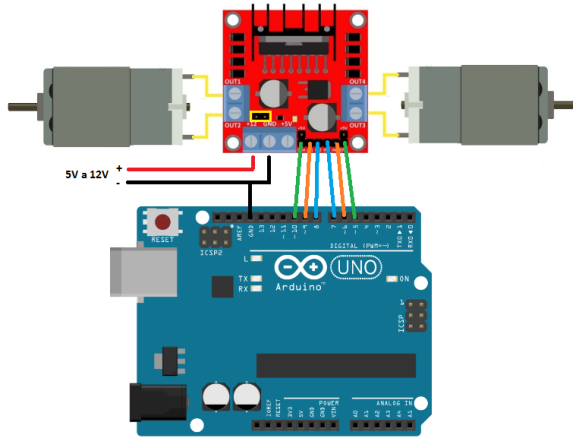
Sensor de color RGB TCS-230



Driver con Integrado L298N



Puente H



LIBRERÍAS ARDUINO UTILIZADAS

- SoftwareSerial
- NewPing



CÓDIGO ARDUINO

```
1 #include <SoftwareSerial.h>
2 //sensor proximidad
3 const int Trigger = 6; //Pin digital 2 para el Trigger del sensor
4 const int Echo = 7; //Pin digital 3 para el Echo del sensor
5
6 //motor
7 int motorControl1A = 4;
8 int motorControl1B = 3;
9
10 int motorControl1C = 10;
11 int motorControl1D = 9;
12
13 int motorControl1E = 6;
14 int motorControl1F = 5;
15
16 int motorControl1G = 13;
17 int motorControl1H = 12;
18
19 int motorSpeed1 = A1;
20 int motorSpeed2 = 7;|
21 int motorSpeed3 = 8;
22 int motorSpeed4 = 11;
23 //Color
24 const int s0 = 23;
25 const int s1 = 24;
26 const int s2 = 25;
27 const int s3 = 26;
28 const int out = 20;
29 // LED pins connected to Arduino
30 int redLed = 32;
31 int greenLed = 33;
32 int blueLed = 34;
33 // Variables
34 int red = 0;
35 int green = 0;
36 int blue = 0;
37 //Fin Color
38
39 SoftwareSerial modBT(2,3);
40 void setup() {
41 // put your setup code here, to run once:
42 Serial.begin(9600);
43 modBT.begin(9600);
44 Serial1.begin(9600);
45 Serial.println(">> START<<");
46
47 pinMode(motorControl1A, OUTPUT);
48 pinMode(motorControl1B, OUTPUT);
49 pinMode(motorControl1C, OUTPUT);
50 pinMode(motorControl1D, OUTPUT);
51 pinMode(motorControl1E, OUTPUT);
52 pinMode(motorControl1F, OUTPUT);
53 pinMode(motorControl1G, OUTPUT);
54 pinMode(motorControl1H, OUTPUT);
55 //Color
56 pinMode(s0, OUTPUT);
57 pinMode(s1, OUTPUT);
58 pinMode(s2, OUTPUT);
59 pinMode(s3, OUTPUT);
60 pinMode(out, INPUT);
61 pinMode(redLed, OUTPUT);
62 pinMode(greenLed, OUTPUT);
63 pinMode(blueLed, OUTPUT);
64 digitalWrite(s0, HIGH);
65 digitalWrite(s1, HIGH);
66 //Fin Color
67 //sensor
68 pinMode(Trigger, OUTPUT); //pin como salida
69 pinMode(Echo, INPUT); //pin como entrada
70 digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
71 }
72
73 void loop() {
74 if(Serial1.available() > 0)
75 {
76 Serial.println("here");
77 byte data;
78 data = Serial1.read();
79 Serial1.write(Serial1.read());
80
81 Serial.println(data);
82 switch (data)
83 {
84
85 case 49: //FORWARD
86 digitalWrite(motorControl1A, HIGH);
87 digitalWrite(motorControl1B, LOW);
88 analogWrite(motorSpeed1, 255);|
```



```

89     digitalWrite(motorControl1C, HIGH);
90     digitalWrite(motorControl1D, LOW);
91     analogWrite(motorSpeed2, 255);
92
93     digitalWrite(motorControl1E, HIGH);
94     digitalWrite(motorControl1F, LOW);
95     analogWrite(motorSpeed3, 255);
96
97     digitalWrite(motorControl1G, HIGH);
98     digitalWrite(motorControl1H, LOW);
99     analogWrite(motorSpeed4, 255);
100     Serial.println("herex2");
101     break;
102
103 case 50: //REVERSE
104     digitalWrite(motorControl1A, LOW);
105     digitalWrite(motorControl1B, HIGH);
106     analogWrite(motorSpeed1, 255);
107
108     digitalWrite(motorControl1C, LOW);
109     digitalWrite(motorControl1D, HIGH);
110     analogWrite(motorSpeed2, 255);
111
112     digitalWrite(motorControl1E, LOW);
113     digitalWrite(motorControl1F, HIGH);
114     analogWrite(motorSpeed3, 255);
115
116     digitalWrite(motorControl1G, LOW);
117     digitalWrite(motorControl1H, HIGH);
118     analogWrite(motorSpeed4, 255);
119     Serial.println("herex3");
120     break;
121
122 case 51: //FORWARD LEFT
123     digitalWrite(motorControl1A, HIGH);
124     digitalWrite(motorControl1B, LOW);
125     analogWrite(motorSpeed1, 255);
126
127     digitalWrite(motorControl1C, LOW);
128     digitalWrite(motorControl1D, HIGH);
129     analogWrite(motorSpeed2, 255);
130
131     digitalWrite(motorControl1E, LOW);
132     digitalWrite(motorControl1F, HIGH);
133     analogWrite(motorSpeed3, 225);
134
135     digitalWrite(motorControl1G, LOW);
136     digitalWrite(motorControl1H, HIGH);
137     analogWrite(motorSpeed4, 225);
138     Serial.println("her4");
139     break;
140
141 case 52: //FORWARD RIGHT
142     digitalWrite(motorControl1A, LOW);
143     digitalWrite(motorControl1B, HIGH);
144     analogWrite(motorSpeed1, 225);
145
146     digitalWrite(motorControl1C, HIGH);
147     digitalWrite(motorControl1D, LOW);
148     analogWrite(motorSpeed2, 225);
149
150     digitalWrite(motorControl1E, HIGH);
151     digitalWrite(motorControl1F, LOW);
152     analogWrite(motorSpeed3, 225);
153
154     digitalWrite(motorControl1G, HIGH);
155     digitalWrite(motorControl1H, LOW);
156     analogWrite(motorSpeed4, 225);
157     Serial.println("herex5");
158     break;
159
160 case 53:
161
162     long t; //timepo que demora en llegar el eco
163     long d; //distancia en centimetros
164
165     digitalWrite(Trigger, HIGH);
166     delayMicroseconds(10); //Enviamos un pulso de 10us
167     digitalWrite(Trigger, LOW);
168
169     t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
170     d = t/59; //escalamos el tiempo a una distancia en cm
171
172
173 if(d <= 5) {
174     Serial.print("Distancia: ");
175     Serial.print(d); //Enviamos serialmente el valor de la distancia

```



```

176 Serial.print(d); //Enviamos serialmente el valor de la distancia
177 Serial.print("cm");
178 Serial.println();
179 color();
180 Serial.print("R Intensity:");
181 Serial.print(red, DEC);
182 Serial.print(" G Intensity: ");
183 Serial.print(green, DEC);
184 Serial.print(" B Intensity : ");
185 Serial.print(blue, DEC);
186 //Serial.println();
187
188 if (red < blue && red < green && red < 20)
189 {
190 Serial.println(" - (Red Color)");
191 digitalWrite(redLed, HIGH); // Turn RED LED ON
192 digitalWrite(greenLed, LOW);
193 digitalWrite(blueLed, LOW);
194 }
195
196 else if (blue < red && blue < green)
197 {
198 Serial.println(" - (Blue Color)");
199 digitalWrite(redLed, LOW);
200 digitalWrite(greenLed, LOW);
201 digitalWrite(blueLed, HIGH); // Turn BLUE LED ON
202 }
203
204 else if (green < red && green < blue)
205 {
206 Serial.println(" - (Green Color)");
207 digitalWrite(redLed, LOW);
208 digitalWrite(greenLed, HIGH); // Turn GREEN LED ON
209 digitalWrite(blueLed, LOW);
210 }
211 else{
212 Serial.println();
213 }
214 delay(300);
215 digitalWrite(redLed, LOW);
216 digitalWrite(greenLed, LOW);
217 digitalWrite(blueLed, LOW);
218 }
219 break;
220
221 default: //If bluetooth module receives any value not listed above, both motors turn off
222 digitalWrite(motorControl1A, LOW);
223 digitalWrite(motorControl1B, LOW);
224 analogWrite(motorSpeed1, 0);
225
226 digitalWrite(motorControl1C, LOW);
227 digitalWrite(motorControl1D, LOW);
228 analogWrite(motorSpeed2, 0);
229
230 digitalWrite(motorControl1E, LOW);
231 digitalWrite(motorControl1F, LOW);
232 analogWrite(motorSpeed3, 0);
233
234 digitalWrite(motorControl1G, LOW);
235 digitalWrite(motorControl1H, LOW);
236 analogWrite(motorSpeed4, 0);
237 Serial.println("nope");
238 break;
239 }
240 }
241 }
242 void color()
243 {
244 digitalWrite(s2, LOW);
245 digitalWrite(s3, LOW);
246 //count OUT, pRed, RED
247 red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
248 digitalWrite(s3, HIGH);
249 //count OUT, pBLUE, BLUE
250 blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
251 digitalWrite(s2, HIGH);
252 //count OUT, pGreen, GREEN
253 green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
254 }

```



CÓDIGO APLICACIÓN ANDROID

```
1 package com.example.practicaapp;
2
3 import android.annotation.SuppressLint;
4 import android.content.Intent;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.bluetooth.BluetoothAdapter;
8 import android.bluetooth.BluetoothDevice;
9 import android.bluetooth.BluetoothSocket;
10 import android.util.Log;
11 import android.view.MotionEvent;
12 import android.view.View;
13 import android.widget.Button;
14 import android.widget.CheckBox;
15 import android.widget.EditText;
16 import android.widget.TextView;
17 import android.widget.Toast;
18
19 import java.io.IOException;
20 import java.io.OutputStream;
21 import java.util.ArrayList;
22 import java.util.Set;
23 import java.util.UUID;
24
25 public class MainActivity extends AppCompatActivity {
26     private final String DEVICE_ADDRESS = "98:D3:32:11:24:2E";
27     private final UUID PORT_UUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
28
29     private BluetoothDevice device;
30     private BluetoothSocket socket;
31     private OutputStream outputStream;
32
33     Button upb, downb, rightb, leftb, btconnect, autobt, rfward, fward;
34     CheckBox recordcheck;
35     TextView txt;
36     EditText name;
37     ArrayList<Recorrido> recorrido=new ArrayList<Recorrido>();
38     String command, nombre;
39     boolean acname=false;
40     @SuppressWarnings("ClickableViewAccessibility")
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_main);
45
46         upb= (Button)findViewById(R.id.button);
47         downb= (Button)findViewById(R.id.button2);
48         rightb= (Button)findViewById(R.id.button4);
49         leftb= (Button)findViewById(R.id.button3);
50         btconnect= (Button)findViewById(R.id.button6);
51         name= (EditText)findViewById(R.id.editText);
52         autobt= (Button)findViewById(R.id.button5);
53         rfward= (Button)findViewById(R.id.button9);
54         fward= (Button)findViewById(R.id.button8);
55         recordcheck=(CheckBox)findViewById(R.id.checkBox);
56
57         txt=(TextView)findViewById(R.id.textView);
58
59
60         //OnTouchListener code for the forward button (button long press)
61         upb.setOnTouchListener(new View.OnTouchListener() {
62             @Override
63             public boolean onTouch(View v, MotionEvent event) {
64
65                 if(recordcheck.isChecked()){
66                     acname=true;
67                     recorrido.add(new Recorrido("Arriba",49,5000));
68                 }
69                 if (event.getAction() == MotionEvent.ACTION_DOWN) //MotionEvent.ACTION_DOWN is when you hold a button down
70                 {
71                     command = "1";
72
73                     try
74                     {
75                         nombre=name.getText().toString();
76                         txt.setText(command.getBytes().toString());
77                         outputStream.write(49); //transmits the value of command to the bluetooth module
78                         //outputStream.write(nombre.getBytes());
79                     }
80                     catch (IOException e)
81                     {
82                         txt.setText(e.toString());
83                     }
84                 }
85                 else if(event.getAction() == MotionEvent.ACTION_UP) //MotionEvent.ACTION_UP is when you release a button
86                 {
87                     command = "10";
88                     txt
89                 {
90
```

```

89         {
90             txt.setText(command.getBytes().toString());
91             outputStream.write(54);
92         }
93         catch(IOException e)
94         {
95             txt.setText(e.toString());
96         }
97     }
98 }
99
100     return false;
101 }
102
103 }):
104 //OnTouchListener code for the reverse button (button long press)
105 downb.setOnTouchListener(new View.OnTouchListener() {
106     @Override
107     public boolean onTouch(View v, MotionEvent event)
108     {
109         if(recordcheck.isChecked()) {
110             recorrido.add(new Recorrido("Abajo",50,5000));
111         }
112         if(event.getAction() == MotionEvent.ACTION_DOWN)
113         {
114             command = "2";
115
116             try
117             {
118                 txt.setText("pues aqui si entra x2");
119                 txt.setText(command.getBytes().toString());
120                 outputStream.write(50);
121             }
122             catch (IOException e)
123             {
124                 txt.setText(e.toString());
125             }
126         }
127         else if(event.getAction() == MotionEvent.ACTION_UP)
128         {
129             command = "10";
130             try
131             {
132                 txt.setText(command.getBytes().toString());
133                 outputStream.write(54);
134             }
135             catch(IOException e)
136             {
137             }
138         }
139     }
140 }
141     return false;
142 }
143 }):
144
145 //OnTouchListener code for the forward left button (button long press)
146 leftb.setOnTouchListener(new View.OnTouchListener() {
147     @Override
148     public boolean onTouch(View v, MotionEvent event)
149     {
150         if(recordcheck.isChecked()) {
151             recorrido.add(new Recorrido("Izquierda",51,5000));
152         }
153         if(event.getAction() == MotionEvent.ACTION_DOWN)
154         {
155             command = "3";
156
157             try
158             {
159                 txt.setText("pues aqui si entra x3");
160                 txt.setText(command.getBytes().toString());
161                 outputStream.write(51);
162             }
163             catch (IOException e)
164             {
165                 txt.setText(e.toString());
166             }
167         }
168         else if(event.getAction() == MotionEvent.ACTION_UP)
169         {
170             command = "10";
171             try
172             {
173                 txt.setText(command.getBytes().toString());
174                 outputStream.write(54);
175             }
176             catch(IOException e)
177             {
178

```




```

179         }
180     }
181     }
182     return false;
183 }
184 });
185
186 //OnTouchListener code for the forward right button (button long press)
187 rightb.setOnTouchListener(new View.OnTouchListener() {
188     @Override
189     public boolean onTouch(View v, MotionEvent event)
190     {
191         if(recordcheck.isChecked()){
192             recorrido.add(new Recorrido("Derecha",52,5000));
193         }
194         if(event.getAction() == MotionEvent.ACTION_DOWN)
195         {
196             command = "4";
197
198             try
199             {
200                 txt.setText("pues aqui si entra x4");
201                 txt.setText(command.getBytes().toString());
202                 outputStream.write(52);
203             }
204             catch (IOException e)
205             {
206                 txt.setText(e.toString());
207             }
208         }
209         else if(event.getAction() == MotionEvent.ACTION_UP)
210         {
211             command = "10";
212             try
213             {
214                 txt.setText(command.getBytes().toString());
215                 outputStream.write(54);
216             }
217             catch (IOException e)
218             {
219                 e.printStackTrace();
220             }
221         }
222     }
223     return false;
224 }
225 });
226
227 autobt.setOnTouchListener(new View.OnTouchListener() {
228     @Override
229     public boolean onTouch(View v, MotionEvent event)
230     {
231         if(event.getAction() == MotionEvent.ACTION_DOWN)
232         {
233             command = "5";
234
235             try
236             {
237                 txt.setText(command.getBytes().toString());
238                 outputStream.write(53);
239             }
240             catch (IOException e)
241             {
242                 e.printStackTrace();
243             }
244         }
245         else if(event.getAction() == MotionEvent.ACTION_UP)
246         {
247             command = "10";
248             try
249             {
250                 txt.setText(command.getBytes().toString());
251                 outputStream.write(54);
252             }
253             catch (IOException e)
254             {
255                 e.printStackTrace();
256             }
257         }
258     }
259     return false;
260 }
261 });
262
263 fward.setOnTouchListener(new View.OnTouchListener() {
264     @Override
265     public boolean onTouch(View v, MotionEvent event)
266     {
267         String t="";
268         if(event.getAction() == MotionEvent.ACTION_DOWN)
269         {
270             command = "5";

```



```

269         try
270         {
271             for(int a=0;a<recorrido.size();a++){
272                 Log.d("",recorrido.get(a).movimiento);
273                 txt.setText(recorrido.get(a).movimiento);
274                 outputStream.write(recorrido.get(a).index);
275             }
276         }
277         catch (IOException e)
278         {
279             e.printStackTrace();
280         }
281     }
282     txt.setText(c);
283     return false;
284 }
285 }
286 //Button that connects the device to the bluetooth module when pressed
287 btconnect.setOnClickListener(new View.OnClickListener() {
288     @Override
289     public void onClick(View v) {
290         if(BTinit())
291         {
292             BTconnect();
293         }
294     }
295 });
296 }
297 public boolean BTinit()
298 {
299     boolean found = false;
300
301     BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
302
303     if(bluetoothAdapter == null) //Checks if the device supports bluetooth
304     {
305         Toast.makeText(getApplicationContext(), "Device doesn't support bluetooth", Toast.LENGTH_SHORT).show();
306     }
307
308     if(!BluetoothAdapter.isEnabled()) //Checks if bluetooth is enabled. If not, the program will ask permission from the user to enable it
309     {
310         Intent enableAdapter = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
311         startActivityForResult(enableAdapter,0);
312
313         try
314         {
315             Thread.sleep(1000);
316         }
317         catch (InterruptedException e)
318         {
319             e.printStackTrace();
320         }
321
322         Set<BluetoothDevice> bondedDevices = bluetoothAdapter.getBondedDevices();
323
324         if(bondedDevices.isEmpty()) //Checks for paired bluetooth devices
325         {
326             Toast.makeText(getApplicationContext(), "Please pair the device first", Toast.LENGTH_SHORT).show();
327         }
328         else
329         {
330             for(BluetoothDevice iterator : bondedDevices)
331             {
332                 if(iterator.getAddress().equals(DEVICE_ADDRESS))
333                 {
334                     device = iterator;
335                     found = true;
336                     break;
337                 }
338             }
339         }
340
341         return found;
342     }
343 }
344 public boolean BTconnect()
345 {
346     boolean connected = true;
347
348     try
349     {
350         socket = device.createRfcommSocketToServiceRecord(PORT_UUID); //Creates a socket to handle the outgoing connection
351         socket.connect();
352
353         Toast.makeText(getApplicationContext(),
354             "Connection to bluetooth device successful", Toast.LENGTH_LONG).show();
355     }
356     catch (IOException e)
357     {
358         e.printStackTrace();
359         connected = false;
360     }
361
362     if(connected)
363     {
364         try
365         {
366             outputStream = socket.getOutputStream(); //gets the output stream of the socket
367         }
368         catch (IOException e)
369         {
370             e.printStackTrace();
371         }
372     }
373
374     return connected;
375 }
376 @Override
377 protected void onStart() { super.onStart(); }
378 }

```