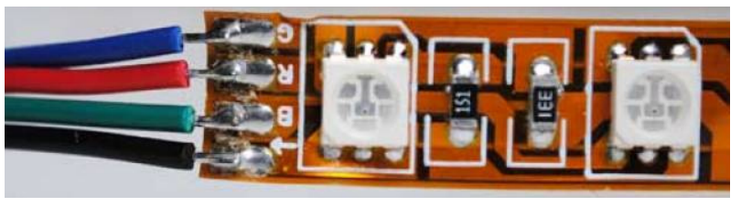
Prof: Bolaños D.J
Electrónica

Proyecto Tachos Led con Arduino *(Versión 10-9-17)*

Se desarrollará un control de Tachos Leds, mediante Arduino, el sistema podra ser controlado por PC, por control remoto infrarrojo y por Bluetooth.

El tacho estará construido con tiras de LED RGB del tipo ánodo común (o sea + a 12v los 3 colores).

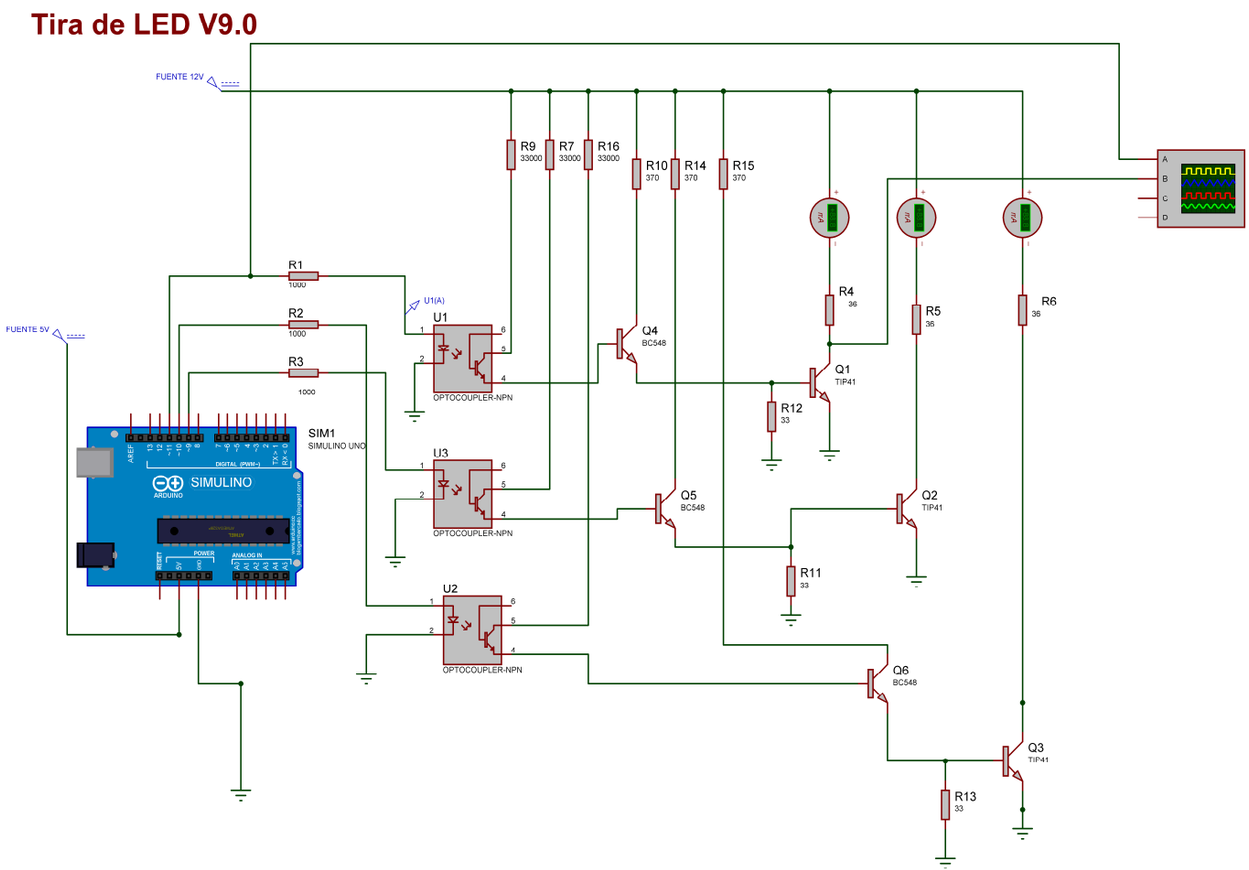


El proyecto utiliza distintos temas ya estudiados por parte separadas.

Podemos por lo tanto separar dichas partes en:

- PWM
- LEDs RGB
- Control por PC (Builder C++)
- Sensor Infrarrojo. Control remoto.
- Bluetooth

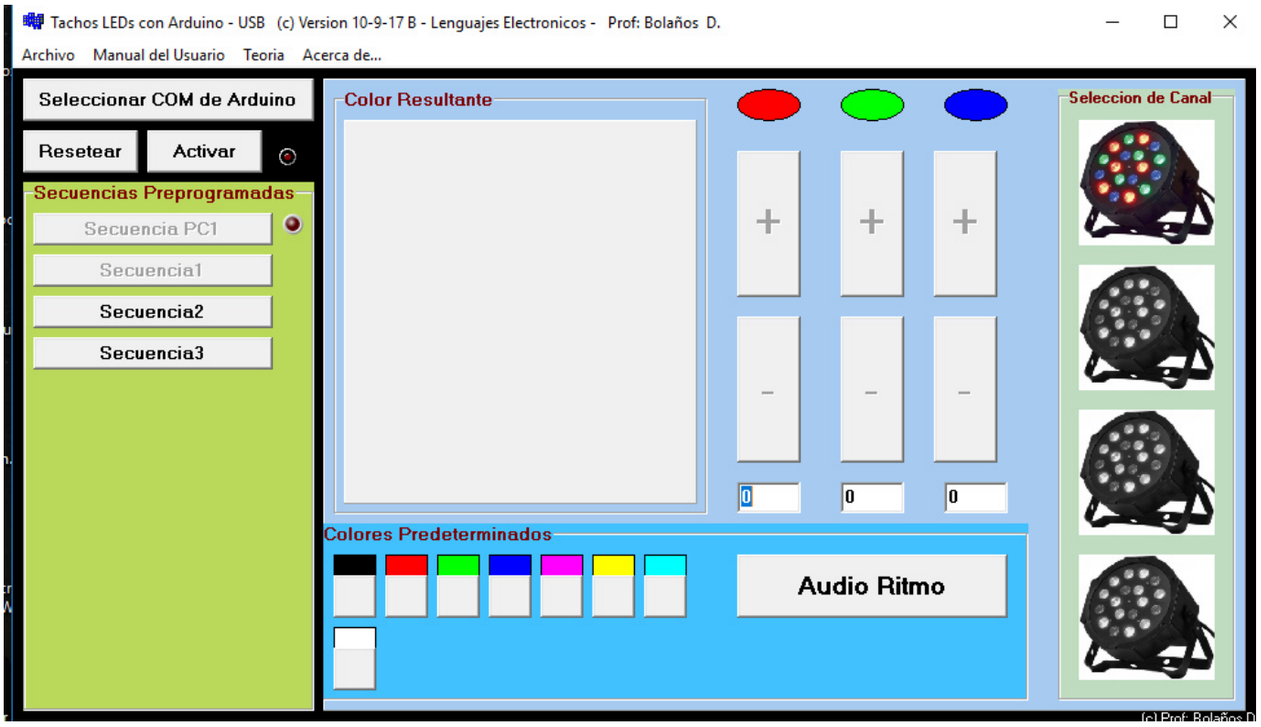
El circuito empleado tiene opto acopladores para relacionar la electrónica de control con la electrónica de potencia. Si bien es la forma correcta de trabajar, la presencia de del circuito de los opto acopladores podría evitarse para mejorar la velocidad del circuito, aunque con cierto riesgo como es imaginable.
El circuito propuesto sería el siguiente:



En este cicuito se reemplazo cada color de una tira de LED de 1 metro por una resistencia de 36 ohms, que surge de medir la corriente que toma cada color y la tension de alimentación que en este caso es de 12v.

Control por PC mediante una aplicación creada con Builder C++

Ventana de la aplicación (la misma podrá ser ampliada en el futuro).



El programa creado envía a una placa Arduino secuencias de números de solo 3 dígitos, los cuales al ser leído ejecutas las acciones necesarias. **LA PLACA ARDUINO DEBE ESTAR CONECTADA AL USB DE LA PC**.
NOTA: En esta primera etapa se controla solo un tacho de LEDs.

Listado del programa

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma link "CPort"
#pragma link "CPortCtl"
#pragma link "ILLed"
#pragma link "LPComponent"
#pragma link "LPDrawLayers"
#pragma link "LPTransparentControl"
#pragma link "SLComponentCollection"
#pragma link "ULBasicControl"
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::FormCreate(TObject *Sender)
{


}

//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
     ComPort1->ShowSetupDialog();
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button2Click(TObject *Sender)
{

        ComPort1->BaudRate = br9600;
        ComPort1->Parity->Bits = prNone;
        ComPort1->DataBits = dbEight;
        ComPort1->StopBits = sbOneStopBit;
        ComPort1->Open();
        Timer1->Enabled=true;  //Habilita Botones

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
  Button3->Enabled=true;
  Button4->Enabled=true;
  Button5->Enabled=true;
  Button6->Enabled=true;
  Button7->Enabled=true;
  Button8->Enabled=true;
 //colores predeterminados
  Button9 ->Enabled=true;
  Button10 ->Enabled=true;
  Button11 ->Enabled=true;
  Button12 ->Enabled=true;
  Button13 ->Enabled=true;
  Button15 ->Enabled=true;

//Secuencias
  Button14 ->Enabled=true;

 //Inicialmente color negro

   Button9->Click();

  Timer1->Enabled=false;

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Label1Click(TObject *Sender)
{

   if(Edit4->Visible==false)

        {

        //instrucciones se ejecutan si condición se cumple
         Edit4->Visible=true;
         Edit5->Visible=true;
         Edit6->Visible=true;

         }
else
         {

         //instrucciones se ejecutan si condición NO se cumple
         Edit4->Visible=false;
         Edit5->Visible=false;
         Edit6->Visible=false;

         }


}
```

```
//---------------------------------------------------------------------------

void __fastcall TForm1::Button3Click(TObject *Sender)
{

     int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (R <= 245)
        {
        R = R + 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("111");
          }

  Panel2->Color=RGB(R,G,B);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button4Click(TObject *Sender)
{

     int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (R >=5)
        {
        R = R - 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("444");
          }

  Panel2->Color=RGB(R,G,B);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button5Click(TObject *Sender)
{

 int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (G <= 245)
        {
        G = G + 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("222");
          }

  Panel2->Color=RGB(R,G,B);


}
```

```
//---------------------------------------------------------------------------

void __fastcall TForm1::Button7Click(TObject *Sender)
{
  int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (B <= 245)
        {
        B = B + 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("333");
          }

  Panel2->Color=RGB(R,G,B);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button6Click(TObject *Sender)
{
  int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (G >=5)
        {
        G = G - 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("555");
          }

  Panel2->Color=RGB(R,G,B);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button8Click(TObject *Sender)
{

 int R,G,B;

      R = StrToFloat(Edit1->Text);;
      G = StrToFloat(Edit2->Text);;
      B =StrToFloat(Edit3->Text);;

      if (B >=5)
        {
        B = B - 5;

        Edit1->Text= R;
        Edit2->Text= G;
        Edit3->Text= B;

        ComPort1->WriteStr("666");
          }

  Panel2->Color=RGB(R,G,B);

}
//---------------------------------------------------------------------------


void __fastcall TForm1::Panel3Click(TObject *Sender)
```

```
{

  ComPort1->WriteStr("700");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button9Click(TObject *Sender)
{
   Panel2->Color=RGB(0,0,0);

   Edit1->Text="0";
   Edit2->Text="0";
   Edit3->Text="0";

  ComPort1->WriteStr("700");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button10Click(TObject *Sender)
{

   Panel2->Color=RGB(255,0,0);

   Edit1->Text="255";
   Edit2->Text="0";
   Edit3->Text="0";

  ComPort1->WriteStr("750");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button11Click(TObject *Sender)
{
    Panel2->Color=RGB(0,255,0);

   Edit1->Text="0";
   Edit2->Text="255";
   Edit3->Text="0";

  ComPort1->WriteStr("800");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button12Click(TObject *Sender)
{
    Panel2->Color=RGB(0,0,255);

   Edit1->Text="0";
   Edit2->Text="0";
   Edit3->Text="255";

  ComPort1->WriteStr("850");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button13Click(TObject *Sender)
{
   Panel2->Color=RGB(255,255,255);

   Edit1->Text="255";
   Edit2->Text="255";
   Edit3->Text="255";

  ComPort1->WriteStr("900");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button14Click(TObject *Sender)
{
```

```
   if(ILLed1->Value==true)

        {
        ILLed1->Value=false;// APAGA LED


        }

else
         {
         ILLed1->Value=true;// ENCIENDE LED


         }

    Button9->Click();
    Timer2->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{

   Button9->Click();

   Timer3->Enabled=true;
   Timer2->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer3Timer(TObject *Sender)
{

    Button10->Click();
    Timer4->Enabled=true;
    Timer3->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer4Timer(TObject *Sender)
{

    Button11->Click();
    Timer5->Enabled=true;
    Timer4->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer5Timer(TObject *Sender)
{

    Button12->Click();

   if(ILLed1->Value==true)
        {
         Timer2->Enabled=true;
         }

else
         {
         Timer2->Enabled=false;
         }

    Timer5->Enabled=false;

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button15Click(TObject *Sender)
{

  ComPort1->WriteStr("130");

}
//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::Button16Click(TObject *Sender)
{
   ComPort1->WriteStr("135");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button21Click(TObject *Sender)
{
   ComPort1->WriteStr("140");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button17Click(TObject *Sender)
{
   ComPort1->WriteStr("999");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button18Click(TObject *Sender)
{
   Panel2->Color=RGB(255,0,255);

   Edit1->Text="255";
   Edit2->Text="0";
   Edit3->Text="255";

  ComPort1->WriteStr("145");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button19Click(TObject *Sender)
{
    Panel2->Color=RGB(255,255,0);

   Edit1->Text="255";
   Edit2->Text="255";
   Edit3->Text="0";

  ComPort1->WriteStr("150");

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button20Click(TObject *Sender)
{
   Panel2->Color=RGB(0,255,255);

   Edit1->Text="0";
   Edit2->Text="255";
   Edit3->Text="255";

  ComPort1->WriteStr("155");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button22Click(TObject *Sender)
{
   ComPort1->WriteStr("888");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::ManualdelUsuario1Click(TObject *Sender)
{
    ShowMessage("Manual en proceso de creacion");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Teoria1Click(TObject *Sender)
{
   ShowMessage("Vea Modulo Color en Verificar 3 y Tutor de Arduino");
}
//---------------------------------------------------------------------------
```

El programa anterior se dijo que envía datos conformados por 3 dígitos decimales que estan asociados a distintas órdenes. Se puede ver el listado de estas a continuación:

Dato enviado	Orden	Dato enviado	Orden
750	ROJO	700	NEGRO
800	VERDE	130	SECUENCIA A1
850	AZUL	135	SECUENCIA A2
150	AMARILLO	140	SECUENCIA A3
145	MAGENTA	111	+ ROJO
900	BLANCO	444	- ROJO
155	CYAN - *CELESTE*	222	+ VERDE
888	AUDIORITMO	555	- VERDE
999	RESET	333	+ AZUL
		666	- AZUL

Nota: Secuencia PC1 es una secuencia creada mediante el envío de varias ordenes a la placa Arduino mediante **Click** en los Botones de colores disponibles, si bien funciona, no es eficiente, tan solo se incluye a titulo informativo.

Placa Arduino

El circuito de conexión de la placa Arduino Uno al circuito de potencia se puede observar en una figura anterior.

El programa Arduino

```
//En esta version TARRGB6. Se combinara con Remoto Infrarrojo
//SE CAMBIO EL PIN 11 POR EL 6 YA QUE TENIA PROBLEMAS
//PARA EL CONTROL DEL RGB
//SE AGREGAN SECUENCIAS Y SENSOR DE SONIDO
//Se va a llevar la seccion de analisis de la variable global (num) a una
//funcion
//llamada analisisnum()
//Se adicionara el modulo Bluetooth para control por movil Android.

//--Para Bluetootth---
//A través de la librería SoftwareSerial
//se pueden cambiar los pines  RX y TX  a otros pines
//para establecer la comunicación con el Modulo Bluetooth
//ya que utilizaremos el puerto serie RX TX para comunicacion
//con la PC, ya sea para cargar o depurar el programa o comunicarse
//via PC con Arduino

#include <SoftwareSerial.h>
SoftwareSerial BT(2,3); // Cambio RX | TX para conectar Modulo Bluetooth
                        //en  pines 2 y 3 (yo elijo llamarlo BT)

long bps=9600; // Como comodidad para definir la velocidad de la comunicacion

//--Fin para Bluetooth----


//Para tachos--------------
//Declaro variables globales

int red=0,green=0,blue=0;
int num;
//Fin de tachos--------------
```

```
//Para infrarrojos-----------
#include <boarddefs.h>
#include <IRremote.h>
#include <IRremoteInt.h>
#include <ir_Lego_PF_BitStreamEncoder.h>

int RECV_PIN = 12; //Pin que recibe el envio infrarrojo

IRrecv irrecv(RECV_PIN);

decode_results results; //results contiene el valor llegado
//Fin para infrarrojos-------

void setup()
{
   pinMode(5, OUTPUT); // configura ‘pin’ como salida, para visualizar
                       //llegada de desconocido

  pinMode(12, INPUT);//Configura PIN12 como entrada

  Serial.begin(9600);   //Iniciamos comunicación con el puerto serie

   BT.begin(bps);//Iniciar serial para Modulo BT

//Para tachos-------
//Paneo de canales---
  analogWrite(9, 255);
  delay(1000);
  analogWrite(9, 0);
  analogWrite(10, 255);
  delay(1000);
  analogWrite(10, 0);
  analogWrite(6, 255);
  delay(1000);
  analogWrite(6, 0);
//Fin paneo de colores---
//Fin para tachos-----

//Para infrarrojos---
  irrecv.enableIRIn(); // Empezamos la recepción

//Fin para infrarrojos-----

}

//Funcion  que muestra resultado en el Serie Monitor
//Tambien manda pulso al LED testigo para saber si llego el rayo infrarrojo
//Cualquier rayo infrarrojo

void dump(decode_results *results) {
  // Dumps out the decode_results structure.
  // Call this after IRrecv::decode()

 // Serial.print("(");
  //Serial.print(results->bits, DEC);
  //Serial.print(" bits)");

  if (results->decode_type == UNKNOWN) {
    //Serial.print("Unknown encoding: ");

      digitalWrite(5,HIGH );//Pulso de llegada desconocido
      delay(100);
      digitalWrite(5,LOW );//Termina pulso

  }

  //Serial.print(results->value, HEX);
  //Serial.print(" (HEX) , ");
```

```
  //Serial.print(results->value, BIN);
 // Serial.println(" (BIN)");

} // Fin funcion que muestra por Monitor Serial si se desea y LED testigo

//Fin de funcion que muestra resultado-----------

//*****************************************
void(* resetFunc) (void) = 0; // esta es la funcion concreta de reseteo
//*****************************************


void loop() {

//Para infrarrojos----------
  if (irrecv.decode(&results)) {
   dump(&results);
    //irrecv.resume(); // empezamos una nueva recepción
    //Serial.print("LEIDO");
    //Serial.print(results.value, HEX);
   //Serial.println(results.value);

   //En result.value tenemos el valor del codigo que llego por infrarrojo
   //A continuacion las acciones dependeran de ese codigo
   //Hay acciones que son exclusivas de manejo por infrarrojos
   //Hay algunas acciones que son llamadas por medio de funciones creadas
   //Esas funciones son accesibles tambien por el control desde PC.

   switch(results.value)
      {
        case 3238126971://Tecla 0

       //Serial.print(results.value); Activarlo si se quiere ver en Monitor
Serial
        SecuenciaA1();
        break;

        case 3810010651://Tecla ch-  Rojo

       //Serial.print(results.value);Activarlo si se quiere ver en Monitor
Serial

        Color(255,0,0);
        break;

       case 5316027://Tecla ch  -Verde

       //Serial.print(results.value);Activarlo si se quiere ver en Monitor
Serial

        Color(0,255,0);
        break;

       case 4001918335://Tecla ch+ Azul

       //Serial.print(results.value);

        Color(0,0,255);
        break;

   case 3622325019://Tecla NEXT-Negro

       //Serial.print(results.value);Activarlo si se quiere ver en Monitor
Serial

        Color(0,0,0);
        break;

    case 553536955://Tecla Play Pause - Lila
```

```
      //Serial.print(results.value);

       Color(255,0,255);
       break;

    case 1386468383://Tecla Play Pause -Amarillo

      //Serial.print(results.value);

       Color(255,255,0);
       break;

    case 3855596927://Tecla EQ -Blanco

      //Serial.print(results.value);

       Color(255,255,255);
       break;

   case 2534850111://Tecla 1

       //Serial.print(results.value);
        SecuenciaA2();
        break;

  case 4039382595://Tecla 200 subidas varios colores

       //Serial.print(results.value);

        SecuenciaA3();
        break;

 case 465573243://Tecla 8 AUDIORRITMICO

       //Serial.print(results.value);
       //Entrar modo audiorritmico
        AUDIORITMO();
        break;

     } //Fin de acciones determinadas por  result.value

  irrecv.resume(); // empezamos una nueva recepción

  }

//Fin relacion infrarrojos tachos---

//Inicio para control desde la PC con programa creado en Builder----

 /*
   * Evaluamos el momento en el cual recibimos un caracter
   * a través del puerto serie
   */
  if (Serial.available()>0) {

    //A continuacion acciones realizadas solo desde PC--

  //Delay para favorecer la lectura de caracteres
    delay(22);
    //Se crea una variable que servirá como buffer
    String bufferString = "";
    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */
    while (Serial.available()>0) {
      bufferString += (char)Serial.read();
    }
```

```
    num = bufferString.toInt();

    analisisnum();  //Llamos funcion que analiza valores de variable num

//-------------------------------------------------
  }  //FIN de acciones realizadas solo por control desde PC---

ordenesdemovil(); //va controlar si hay envios desde movil

} //Llave final del void loop()----

//*************************************************
//Funciones creadas----------------------
//Son accesibles tanto por infrrarrojos como por control por PC.

void Menos_blue()
      {
            blue = blue - 5;
            if (blue == -5)
             {
              blue = blue + 5;
             }
            analogWrite(6, blue) ;// blue -blue  PIN 11
       }
//--------------------
void Menos_green()
      {
           green= green - 5;
           if (green == -5)
            {
            green = green + 5;
            }
            analogWrite(10, green) ;   // Green - Verde  PIN 10
       }
//--------------------

void Menos_red()
      {
             red= red - 5;
             if (red == -5)
             {
             red= red +5;
             }
             analogWrite(9, red) ;   // Rojo  PIN 9
      }

//-----------------------
void Mas_blue()

    {
     //Serial.print(num);

     blue = blue + 5;

    // Serial.print(blue);

     if (blue > 255)
      {
     blue = blue -5;

      }

     analogWrite(6, blue);
     }

//--------------------
void Mas_green()
      {
            green= green + 5;
```

```
            if (green > 255)
            {
            green = green - 5;
            }
            analogWrite(10, green) ;   // Green - Verde  PIN 10
     }
//--------------------
void Mas_red()
     {
             red= red +5;
             if (red > 255)
             {
             red= red-5;
             }
             analogWrite(9, red) ;   // Rojo  PIN 9
     }

//-----------------------------
void Color(int R, int G, int B)
    {
          analogWrite(9 , R) ;   // Rojo
          analogWrite(10, G) ;   // Green - Verde
          analogWrite(6, B) ;   // blue - blue
    }

//-------------------------
void SecuenciaA1()
   {
     for (int i =0 ; i<255 ; i++)
           {
            Mas_red();//Sube a maximo rojo
            delay(20);
            Color(0, 0, 0); //negro
            delay(25);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }
      Color(0, 0, 0); //negro

     for (int i =0 ; i<255 ; i++)
           {
            Mas_green();//Sube a maximo verde
            delay(20);
            Color(0, 0, 0); //negro
            delay(25);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }
         Color(0, 0, 0); //negro

      for (int i =0 ; i<255 ; i++)
           {

            Mas_blue();//Sube a maximo blue
            delay(20);
            Color(0, 0, 0); //negro
            delay(25);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();
            }


for (int i =0 ; i<25 ; i++)
           {
         Color(0, 0, 0); //negro
```

```
            delay (100);
         Color(255, 0, 0);
            delay (100);
         Color(0, 0, 255);
            delay (300);
          Color(0, 255, 0);
            delay (100);
          Color(0, 0, 0); //negro
           delay (200);
          Color(0, 255, 0);
            delay (100);
          Color(0, 0, 0); //negro
          RESETEOINFR();
          RESETEOPC();
          RESETEOBT();

         }
   }

void SecuenciaA2()
   {

for (int i =0 ; i<25 ; i++)
           {
         Color(255, 0, 255);
            delay (100);
         Color(0, 255, 0);
            delay (100);
         Color(255, 0, 0);
            delay (100);
          Color(255, 255, 0);
            delay (100);
          Color(255, 255, 255); //blanco
           delay (200);
          Color(255, 0, 0);
            delay (100);
          Color(0, 0, 0); //negro

          RESETEOINFR();
          RESETEOPC();
          RESETEOBT();

         }

   }

void SecuenciaA3()
   {
     for (int i =0 ; i<255 ; i++)
           {
            Mas_red();//Sube a maximo rojo
            delay(2);
            Color(0, 0, 0); //negro
            delay(20);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }

      Color(255, 0, 0);
      delay(200);

      Color(0, 0, 0); //negro

     for (int i =0 ; i<255 ; i++)
           {
            Mas_green();//Sube a maximo verde
            delay(2);
            Color(0, 0, 0); //negro
```

```
            delay(20);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }

          Color(0, 255, 0);
          delay(200);
          Color(0, 0, 0); //negro

       for (int i =0 ; i<255 ; i++)
            {

            Mas_blue();//Sube a maximo blue
            delay(2);
            Color(0, 0, 0); //negro
            delay(20);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }
        Color(0, 0, 255);
        delay(200);
        Color(0, 0, 0); //negro


    for (int i =0 ; i<255 ; i++)
           {

            Mas_red();//Sube  red
            Mas_green();//Sube  verde

            delay(2);
            Color(0, 0, 0); //negro
            delay(20);
            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();

            }
        Color(255, 255, 0);
        delay(200);
        Color(0, 0, 0); //negro


  for (int i =0 ; i<255 ; i++)
           {

            Mas_blue();//Sube  azul
            Mas_red();//Sube  rede

            delay(2);
            Color(0, 0, 0); //negro
            delay(20);

            RESETEOINFR();
            RESETEOPC();
            RESETEOBT();
            }
        Color(255, 0,255);
        delay(200);
        Color(0, 0, 0); //negro

  }
//Funcion reseteo infrarrojo----
void RESETEOINFR()
   {
//Para infrarrojos----------
```

```
        delay(100);
  if (irrecv.decode(&results)) {
   dump(&results);

   switch(results.value)
      {
        case 1053031451://Tecla 9

        resetFunc();  // llamo funcion especifica de reseteo
        break;
      }

 }

  irrecv.resume(); // empezamos una nueva recepción
 }

// Fin Funcion reseteo infrarrojo----

//Funcion audiritmico----
void AUDIORITMO()
   {

int AUD=1;// Esta variable es para entrar a WHILE, nunca cambia

    while (AUD== 1) // testea si AUD =1
{
  int valor;
  valor = analogRead(0); // asigna a valor lo que lee
                         // en la entrada ´pin'

if (valor > 526)
     AUDIOPICO();

if ((valor > 523) && (valor < 525 ))
     AUDIOMEDIO();
if ((valor > 520) && (valor < 522 ))
     AUDIOBAJO();

//Serial.println(valor); Activar para seguimiento diseñador
    RESETEOINFR();
    RESETEOPC();
    RESETEOBT();

    Color(0,0,0);
    delay(20);
}

   }   //Fin funcion audioritmico

//Funciones para AUDIO
void  AUDIOPICO()
{
Color(255,0,0);
delay(200);
Color(0,255,0);
delay(200);
Color(0,0,255);
delay (200);
Color(0,0,0);
}

void AUDIOMEDIO()
{
Color(255,255,0);
delay(100);
Color(255,255,0);
delay(100);
Color(0,0,255);
```

```
delay (200);
Color(255,0,0);
delay (100);
Color(0,0,0);
}

void AUDIOBAJO()
{
Color(255,0,0);
delay(100);
Color(0,0,0);
}

void RESETEOPC()  //Funcion reseteo por PC
{

if (Serial.available()>0) {

   //Delay para favorecer la lectura de caracteres
    delay(22);
    //Se crea una variable que servirá como buffer
    String bufferString = "";
    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */
    while (Serial.available()>0) {
      bufferString += (char)Serial.read();
    }

    num = bufferString.toInt();

//Analiza si llego orden de resetear----

     if (num==999)
            {

             resetFunc();  // llamo funcion especifica de reseteo

            }

}//Fin RESETEOPC

}

void analisisnum()
   {
//Colores fijos

     if (num==700)//Negro
            {

             analogWrite(9, 0) ;   // Rojo  PIN 9
             analogWrite(10, 0) ;   // Green - Verde  PIN 10
             analogWrite(6, 0) ;   // blue - blue  PIN 11
            red= 0;
            green= 0;
            blue= 0;
            }

    if (num==750)  //Rojo
            {

             analogWrite(9, 255) ;   // Rojo  PIN 9
             analogWrite(10, 0) ;   // Green - Verde  PIN 10
             analogWrite(6, 0) ;   // blue - blue  PIN 11
            red= 255;
            green= 0;
```

```
            blue= 0;
            }

    if (num==800) //Verde
            {

             analogWrite(9, 0) ;   // Rojo  PIN 9
             analogWrite(10, 255) ;   // Green - Verde  PIN 10
             analogWrite(6, 0) ;   // blue - blue  PIN 11
            red= 0;
            green= 255;
            blue= 0;
            }

   if (num==850)  //blue
            {

             analogWrite(9, 0) ;   // Rojo  PIN 9
             analogWrite(10, 0) ;   // Green - Verde  PIN 10
             analogWrite(6, 255) ;   // blue - blue  PIN 11
            red= 0;
            green= 0;
            blue= 255;
            }

 if (num==900)  //Blanco
            {

             analogWrite(9, 255) ;   // Rojo  PIN 9
             analogWrite(10, 255) ;   // Green - Verde  PIN 10
             analogWrite(6, 255) ;   // blue - blue  PIN 11
            red= 255;
            green= 255;
            blue= 255;
            }

if (num==145)  //Lila  (Magenta)
            {

             analogWrite(9, 255) ;   // Rojo  PIN 9
             analogWrite(10, 0) ;   // Green - Verde  PIN 10
             analogWrite(6, 255) ;   // blue - blue  PIN 11
            red= 255;
            green= 0;
            blue= 255;
            }


if (num==150)  //Amarillo
            {

             analogWrite(9, 255) ;   // Rojo  PIN 9
             analogWrite(10, 255) ;   // Green - Verde  PIN 10
             analogWrite(6, 0) ;   // blue - blue  PIN 11
            red= 255;
            green= 255;
            blue= 0;
            }


if (num==155)  //Celeste (Cyan)
            {

             analogWrite(9, 0) ;   // Rojo  PIN 9
             analogWrite(10, 255) ;   // Green - Verde  PIN 10
             analogWrite(6, 255) ;   // blue - blue  PIN 11
            red= 0;
            green= 255;
            blue= 255;
            }
```

```
//Secuencias ---
 if (num==130)
            {
             SecuenciaA1();
            }
 if (num==135)
            {
             SecuenciaA2();
            }


  if (num==140)
            {
             SecuenciaA3();
            }

//-------------------------------------------
//Se ordena el encendido de los LEDs por pasos
//Va de 0 a 255 con pasos de 5
// Subiendo intensidad----------

        if (num==111)
            {
             Mas_red();
            }

        if (num ==222)
            {
             Mas_green();
            }

         if (num ==333)
            {
            //Serial.print(num);
            //Serial.print(blue);
            Mas_blue();

            }
//Fin subiendo intensidad--------------

//---   Bajando intensidad de a pasos-------------

        if (num==444)
            {
            Menos_red();
            }

        if (num ==555)
            {
            Menos_green();
            }

         if (num ==666)
            {
            Menos_blue();
            }
//Fin bajando intensidad de a pasos-----------------
          if (num ==888)
            {
            AUDIORITMO();
            }

   }//llave fina de  funcion analisisnum---
//Fin funcion analisisnum--------------------------

//--Funcion que revisa ordenes provenientes de un movil
void ordenesdemovil()

   {
```

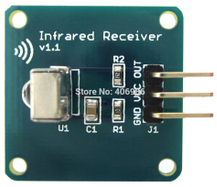
```
if(BT.available()>=1)// Me refiero a la comunicacion con Modulo BT
   {

       //Delay para favorecer la lectura de caracteres

    delay(22);

    //Se crea una variable que servirá como buffer
    String bufferString = "";

    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */

    while (BT.available()>0) {
      bufferString += (char)BT.read();
                          }

    num = bufferString.toInt(); //Se carga lo leido en la variable global num

     // Serial.println(num);//Muestro contenido de variable entrada

   analisisnum();
     }//Llave final de BT.available
   } //Llave final de funcion ordenesmovil
//-----------------------------------------------------

void RESETEOBT()  //Funcion reseteo por BT*******************
{

if (BT.available()>0) {

   //Delay para favorecer la lectura de caracteres
    delay(22);
    //Se crea una variable que servirá como buffer
    String bufferString = "";
    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */
    while (BT.available()>0) {
      bufferString += (char)BT.read();
                    }

    num = bufferString.toInt();

//Analiza si llego orden de resetear----

     if (num==999)
            {

             resetFunc();  // llamo funcion especifica de reseteo

            }
}

}//Fin RESETEOBT
```

Sección Control Remoto Infrarrojo

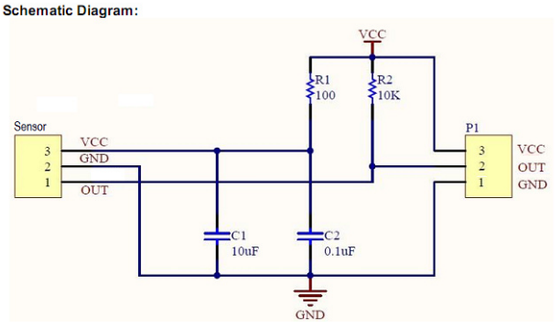
Se uso el control remoto que viene en kit de Arduino

Conexiones entre Arduino y modulo receptor IR

Las conexiones son simples el sensor tiene un pin VCC el cual se alimenta con 5V un pin GND y un pin de DATA, que es una salida digital el cual conectaremos al pin 11 del Arduino.

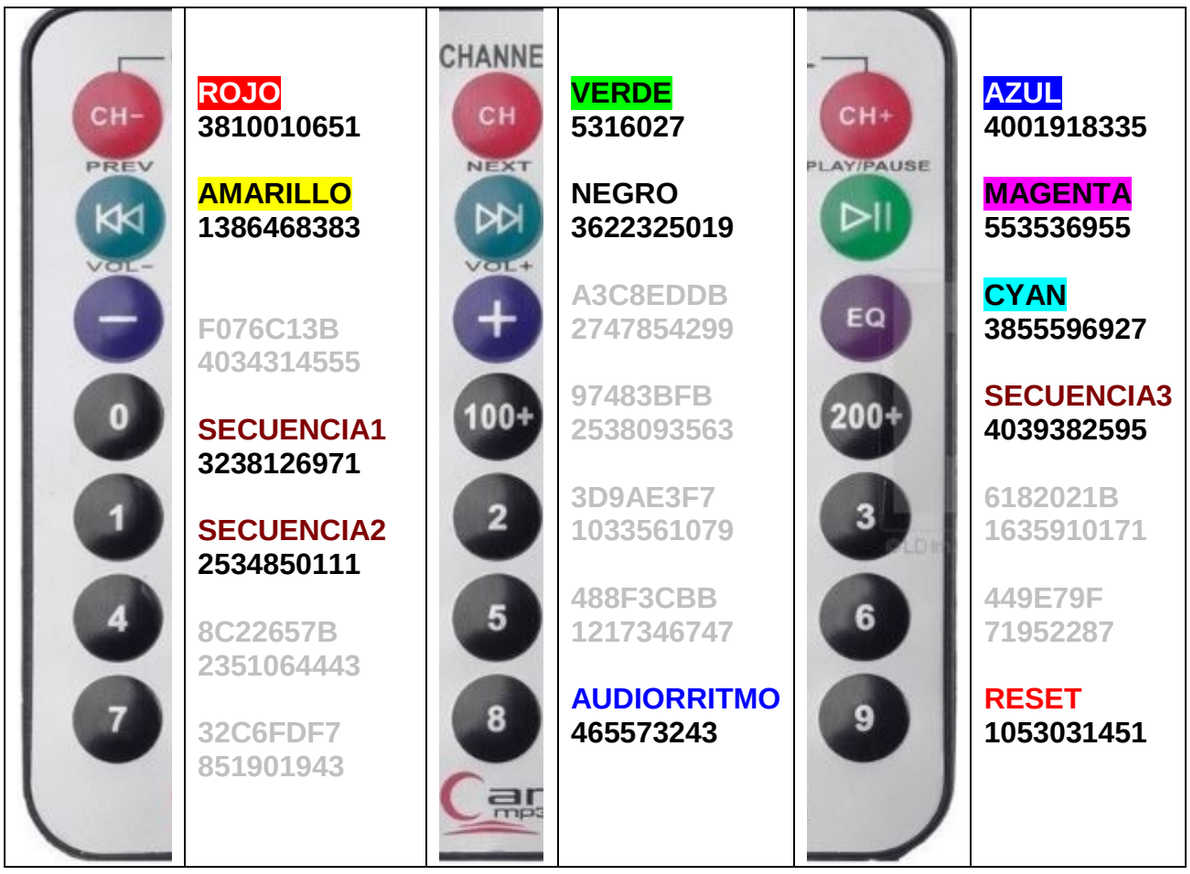


Schematic Diagram:



Recuerde que si el sensor que cuenta en su kit no incluye la interfaz en el modulo Keyes, entonces deberá implementarla como mencionamos antes.

Los botones que tienen función esta indicado en el siguiente gráfico:



ROJO
3810010651

AMARILLO
1386468383

F076C13B
4034314555

SECUENCIA1
3238126971

SECUENCIA2
2534850111

8C22657B
2351064443

32C6FDF7
851901943

VERDE
5316027

NEGRO
3622325019

A3C8EDDB
2747854299

97483BFB
2538093563

3D9AE3F7
1033561079

488F3CBB
1217346747

AUDIORRITMO
465573243

AZUL
4001918335

MAGENTA
553536955

CYAN
3855596927

SECUENCIA3
4039382595

6182021B
1635910171

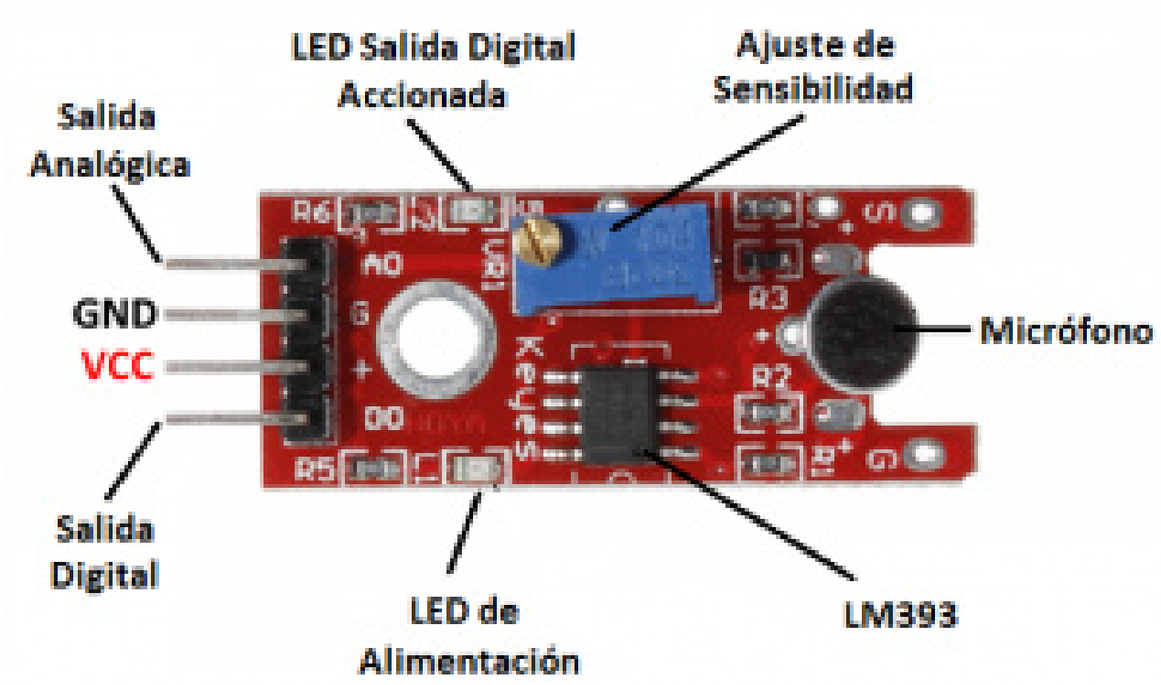
449E79F
71952287

RESET
1053031451

IMPORTANTE: Una vez que se entra a las secuencias o a la función Audiorritmo se debe salir, solo se puede salir con la orden **RESET**.

Funcion Audioritmo

Para esta funcion se utilizo el sensor de sonido que viene en el KIT de Arduino. No es muy eficiente, se recomienda adaptar a un amplificador integrado como el LM386



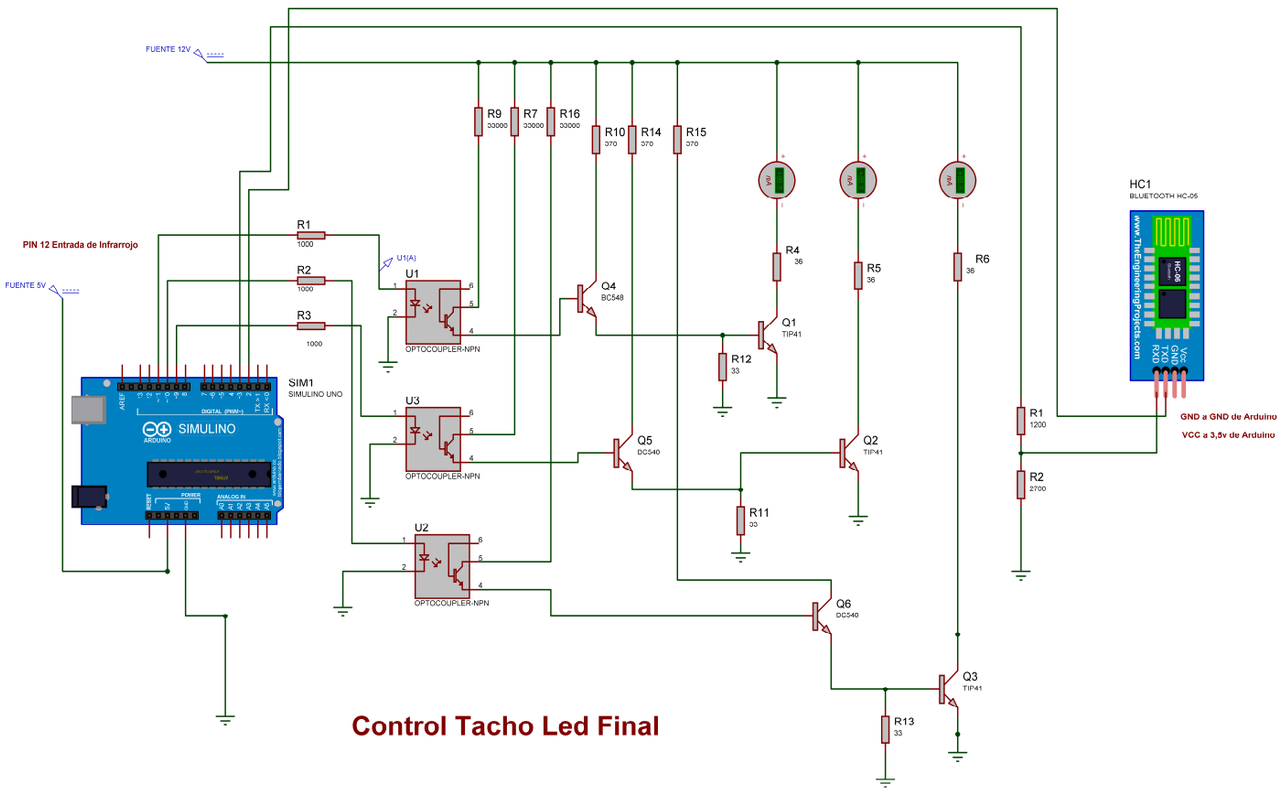
Usamos la salida analógica del modulo conectada al pin A0 del Arduino.

```
//Funcion audiritmico----
void AUDIORITMO()
   {

int AUD=1;// Esta variable es para entrar a WHILE, nunca cambia

    while (AUD== 1) // testea si AUD =1
{
  int valor;
  valor = analogRead(0); // asigna a valor lo que lee
                         // en la entrada ´pin'

if (valor > 526)
     AUDIOPICO();

if ((valor > 523) && (valor < 525 ))
     AUDIOMEDIO();
if ((valor > 520) && (valor < 522 ))
     AUDIOBAJO();

//Serial.println(valor); Activar para seguimiento diseñador
    RESETEOINFR();
    RESETEOPC();
    RESETEOBT();

    Color(0,0,0);
    delay(20);
}

   }   //Fin funcion audioritmico
```

Ver el programa completo para obtener información de las funciones mensionadas
.

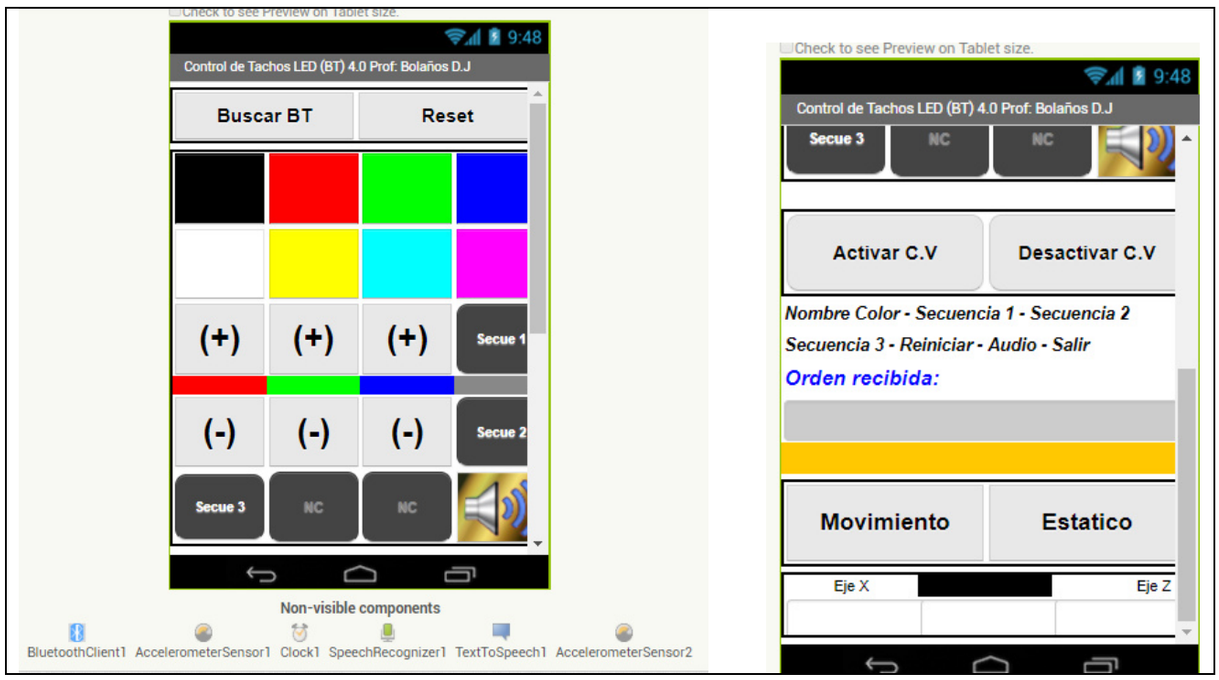
Sección Control Bluetooth

Una vez configurado el modulo Bluetooth, se puede comenzar a usar en el circuito.



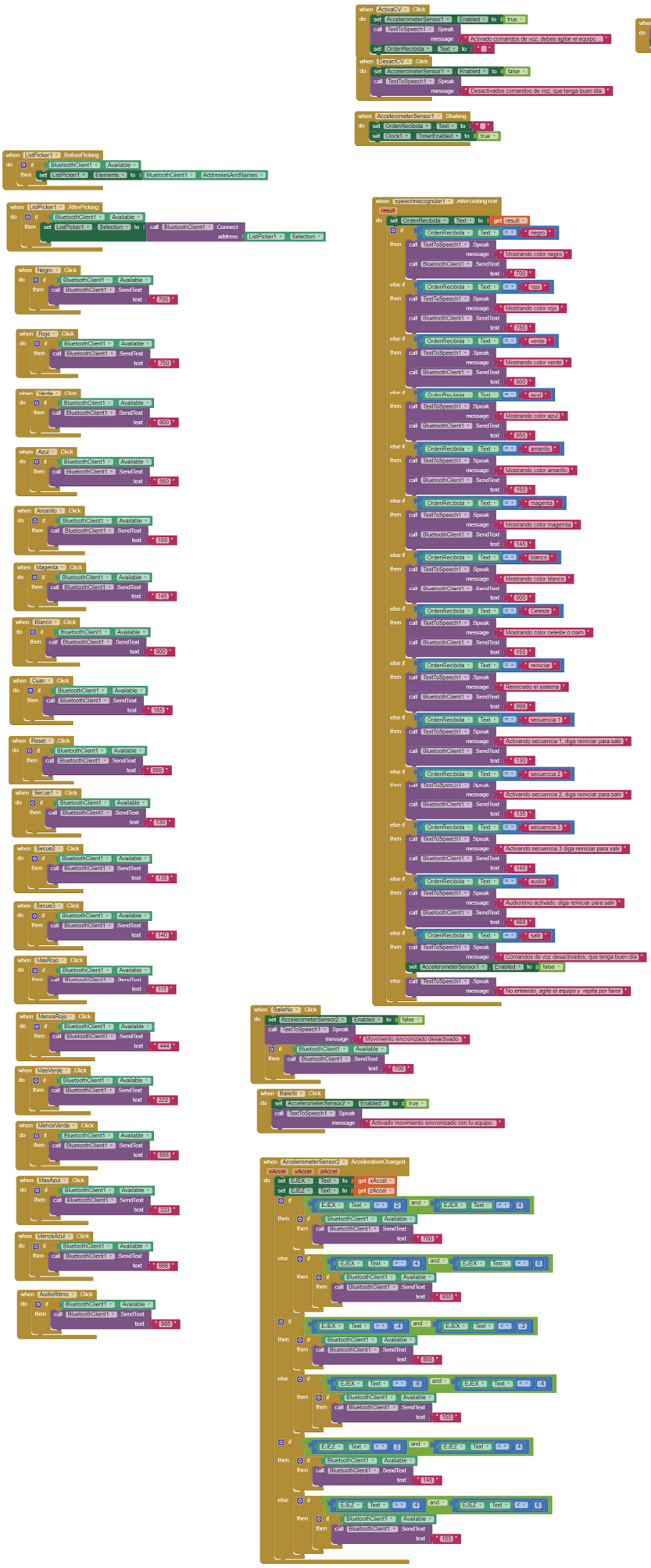
En el apunte **ConfigHC05** se trata con más detalle el tema Bluetooth.

Aplicación para el manejo por movil del modulo Bluetooth



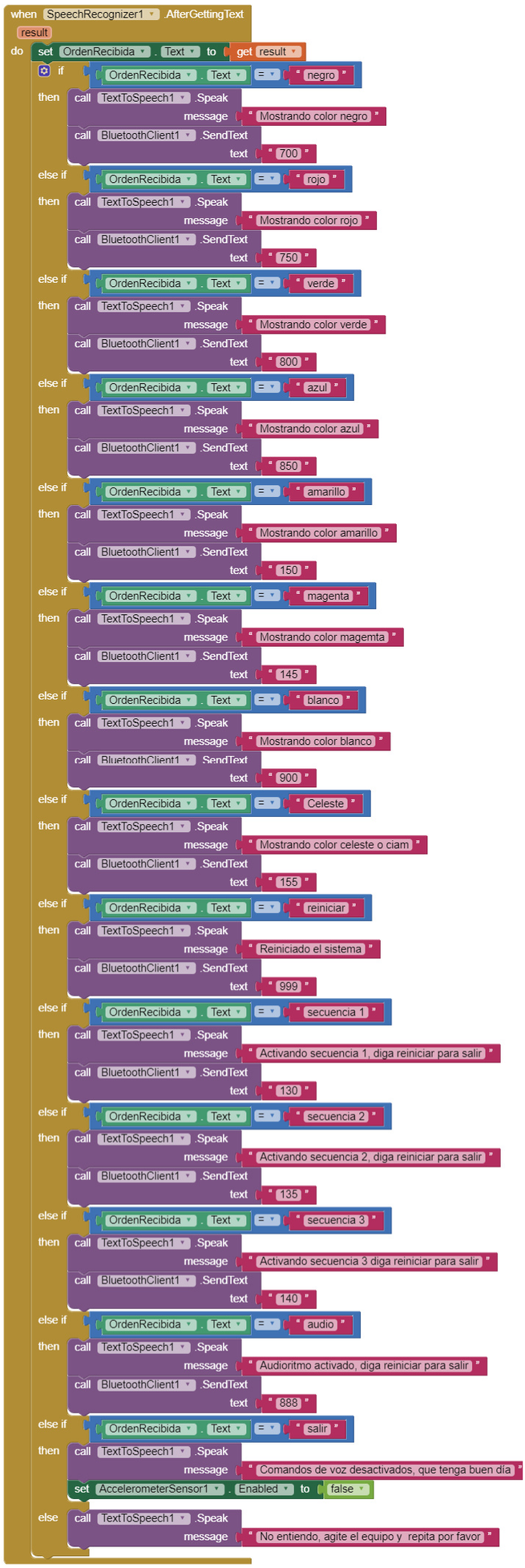
La APP permite controlar las mismas funciones que en el control por PC y mediante control remoto infrarrojo. Pero se adicion la funcion

comandos de voz y la función de variación por movimientos del portador del movil.

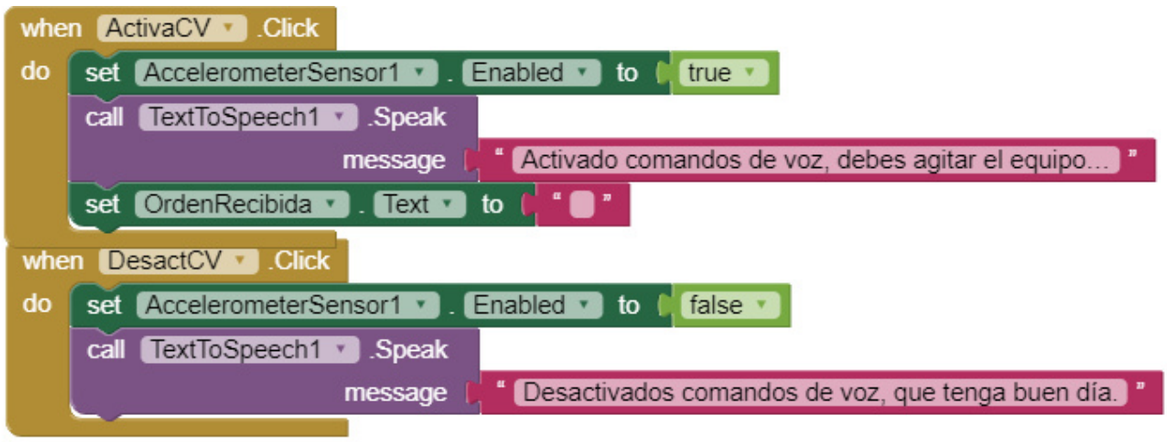


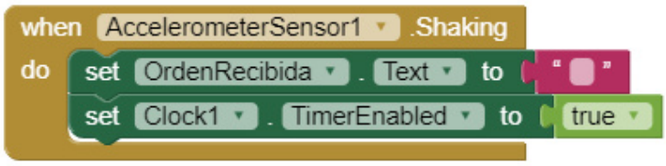
A continuación se separa en partes para su mejor apreciación.

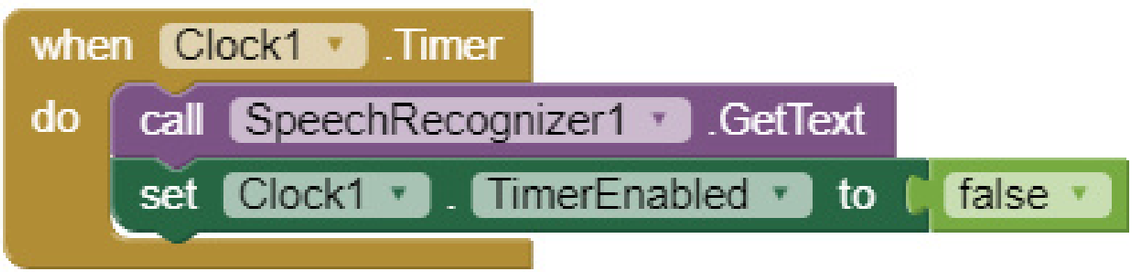
Parte 1


when SpeechRecognizer1 .AfterGettingText
result
do set OrdenRecibida . Text to get result
if OrdenRecibida . Text = " negro "
then call TextToSpeech1 .Speak message " Mostrando color negro "
call BluetoothClient1 .SendText text " 700 "
else if OrdenRecibida . Text = " rojo "
then call TextToSpeech1 .Speak message " Mostrando color rojo "
call BluetoothClient1 .SendText text " 750 "
else if OrdenRecibida . Text = " verde "
then call TextToSpeech1 .Speak message " Mostrando color verde "
call BluetoothClient1 .SendText text " 800 "
else if OrdenRecibida . Text = " azul "
then call TextToSpeech1 .Speak message " Mostrando color azul "
call BluetoothClient1 .SendText text " 850 "
else if OrdenRecibida . Text = " amarillo "
then call TextToSpeech1 .Speak message " Mostrando color amarillo "
call BluetoothClient1 .SendText text " 150 "
else if OrdenRecibida . Text = " magenta "
then call TextToSpeech1 .Speak message " Mostrando color magemta "
call BluetoothClient1 .SendText text " 145 "
else if OrdenRecibida . Text = " blanco "
then call TextToSpeech1 .Speak message " Mostrando color blanco "
call BluetoothClient1 .SendText text " 900 "
else if OrdenRecibida . Text = " Celeste "
then call TextToSpeech1 .Speak message " Mostrando color celeste o ciam "
call BluetoothClient1 .SendText text " 155 "
else if OrdenRecibida . Text = " reiniciar "
then call TextToSpeech1 .Speak message " Reiniciado el sistema "
call BluetoothClient1 .SendText text " 999 "
else if OrdenRecibida . Text = " secuencia 1 "
then call TextToSpeech1 .Speak message " Activando secuencia 1, diga reiniciar para salir "
call BluetoothClient1 .SendText text " 130 "
else if OrdenRecibida . Text = " secuencia 2 "
then call TextToSpeech1 .Speak message " Activando secuencia 2, diga reiniciar para salir "
call BluetoothClient1 .SendText text " 135 "
else if OrdenRecibida . Text = " secuencia 3 "
then call TextToSpeech1 .Speak message " Activando secuencia 3 diga reiniciar para salir "
call BluetoothClient1 .SendText text " 140 "
else if OrdenRecibida . Text = " audio "
then call TextToSpeech1 .Speak message " Audioritmo activado, diga reiniciar para salir "
call BluetoothClient1 .SendText text " 888 "
else if OrdenRecibida . Text = " salir "
then call TextToSpeech1 .Speak message " Comandos de voz desactivados, que tenga buen día "
set AccelerometerSensor1 . Enabled to false
else call TextToSpeech1 .Speak message " No entiendo, agite el equipo y repita por favor "

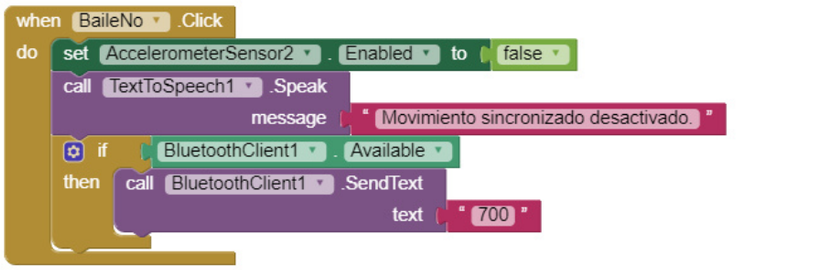
Parte 2

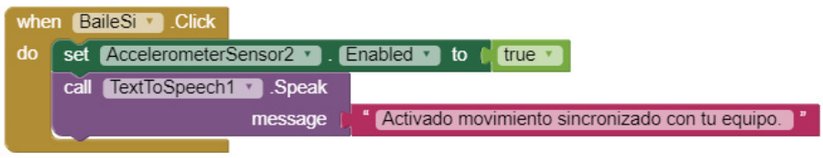

when ActivaCV .Click
do set AccelerometerSensor1 . Enabled to true
call TextToSpeech1 .Speak
message " Activado comandos de voz, debes agitar el equipo... "
set OrdenRecibida . Text to " "
when DesactCV .Click
do set AccelerometerSensor1 . Enabled to false
call TextToSpeech1 .Speak
message " Desactivados comandos de voz, que tenga buen día. "

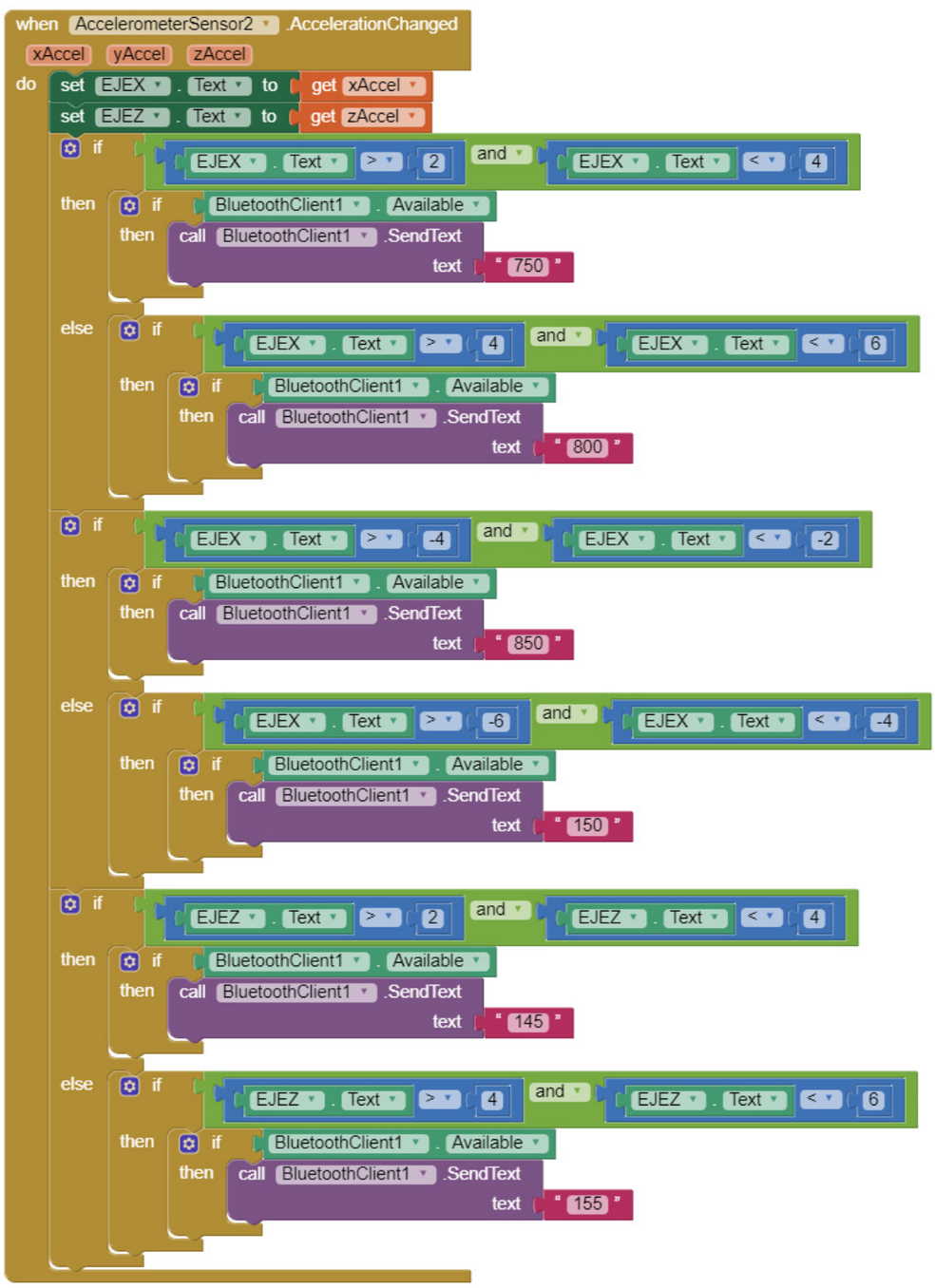

when AccelerometerSensor1 .Shaking
do set OrdenRecibida . Text to " "
set Clock1 . TimerEnabled to true


when Clock1 .Timer
do call SpeechRecognizer1 .GetText
set Clock1 . TimerEnabled to false

Parte 3

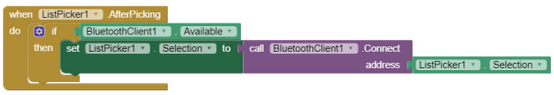

when BaileNo .Click
do set AccelerometerSensor2 . Enabled to false
call TextToSpeech1 .Speak
message " Movimiento sincronizado desactivado. "
if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 700 "

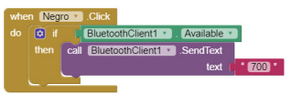

when BaileSi .Click
do set AccelerometerSensor2 . Enabled to true
call TextToSpeech1 .Speak
message " Activado movimiento sincronizado con tu equipo. "

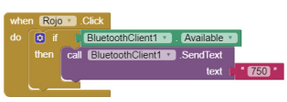

when AccelerometerSensor2 .AccelerationChanged
xAccel yAccel zAccel
do set EJEX . Text to get xAccel
set EJEZ . Text to get zAccel
if EJEX . Text > 2 and EJEX . Text < 4
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 750 "
else if EJEX . Text > 4 and EJEX . Text < 6
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 800 "
if EJEX . Text > -4 and EJEX . Text < -2
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 850 "
else if EJEX . Text > -6 and EJEX . Text < -4
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 150 "
if EJEZ . Text > 2 and EJEZ . Text < 4
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 145 "
else if EJEZ . Text > 4 and EJEZ . Text < 6
then if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " 155 "

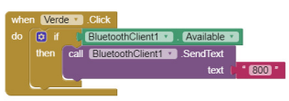
Parte 4

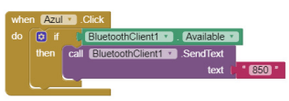

when ListPicker1 .BeforePicking
do if BluetoothClient1 .Available
then set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

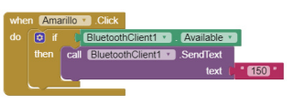

when ListPicker1 .AfterPicking
do if BluetoothClient1 .Available
then set ListPicker1 .Selection to call BluetoothClient1 .Connect
address ListPicker1 .Selection

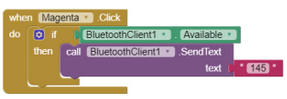

when Negro .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "700"

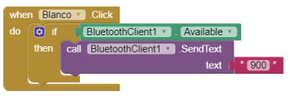

when Rojo .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "750"


when Verde .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "800"

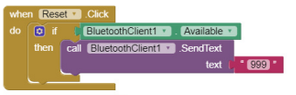

when Azul .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "850"

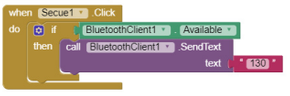

when Amarillo .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "150"

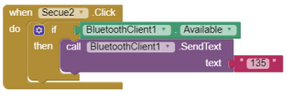

when Magenta .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "145"

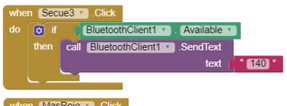

when Blanco .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "900"

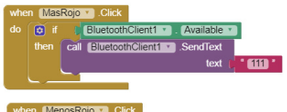

when Cyan .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "155"

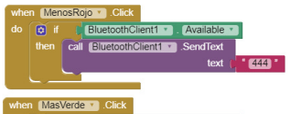

when Reset .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "999"

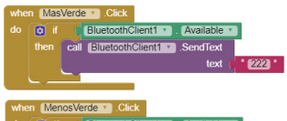

when Secue1 .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "130"

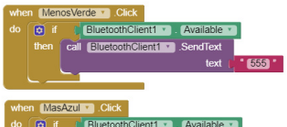

when Secue2 .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "135"


when Secue3 .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "140"

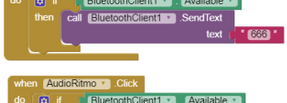

when MasRojo .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "111"

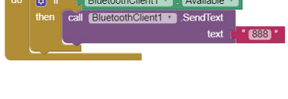

when MenosRojo .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "444"


when MasVerde .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "222"


when MenosVerde .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "555"


when MasAzul .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "333"


when MenosAzul .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "666"


when AudioRitmo .Click
do if BluetoothClient1 .Available
then call BluetoothClient1 .SendText
text "888"

Fotografía del circuito en ensayo