

# Proyecto Multitacho EXPO 2017

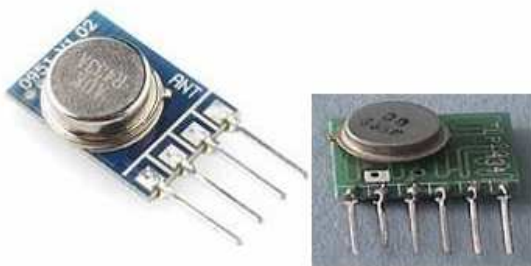
(versión 8-3-18)

En presentaciones anteriores del tutor de Arduino ya hemos tratado el tema de manejo de Tachos LEDs mediante 3 formas de control:

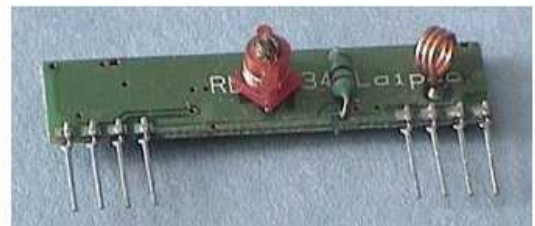
- Control Remoto Infrarrojo.
- Bluetooth por medio Aplicación Android desde celular o tablet, o por PC con Bluetooth.
- Control por puerto USB via cable desde una PC.

También hemos presentado la posibilidad de controlar varios tachos a la vez, a los cuales hemos llamado tachos esclavos. En este caso de un tacho principal (que podemos llamar consola) se transmiten ordenes desde algunas de las 3 formas de control antes mencionadas a otros tachos que podrán trabajar en forma conjunta con el tacho principal o en forma independiente (ejecutando una secuencia por ejemplo) pero siempre ordenada desde el tacho principal o consola. La ordenes desde el Tacho Principal llegan al o los tachos esclavos por medio del uso de Modulos RF Wenshing.

MODULO EMISOR



MODULO RECEPTOR



Vamos a resumir a continuación el avance de dicho proyecto y que se presento en la EXPO 2017 de la Semana de la Educación Técnica en la EEST5 de Tigre.

Uno de los cambios en el Tacho Principal es que se suplanto el modulo detector de sonido Arduino que viene en el KIT básico, por un amplificador de audio mas eficiente construido con un LM 386 alimentado con 5v.

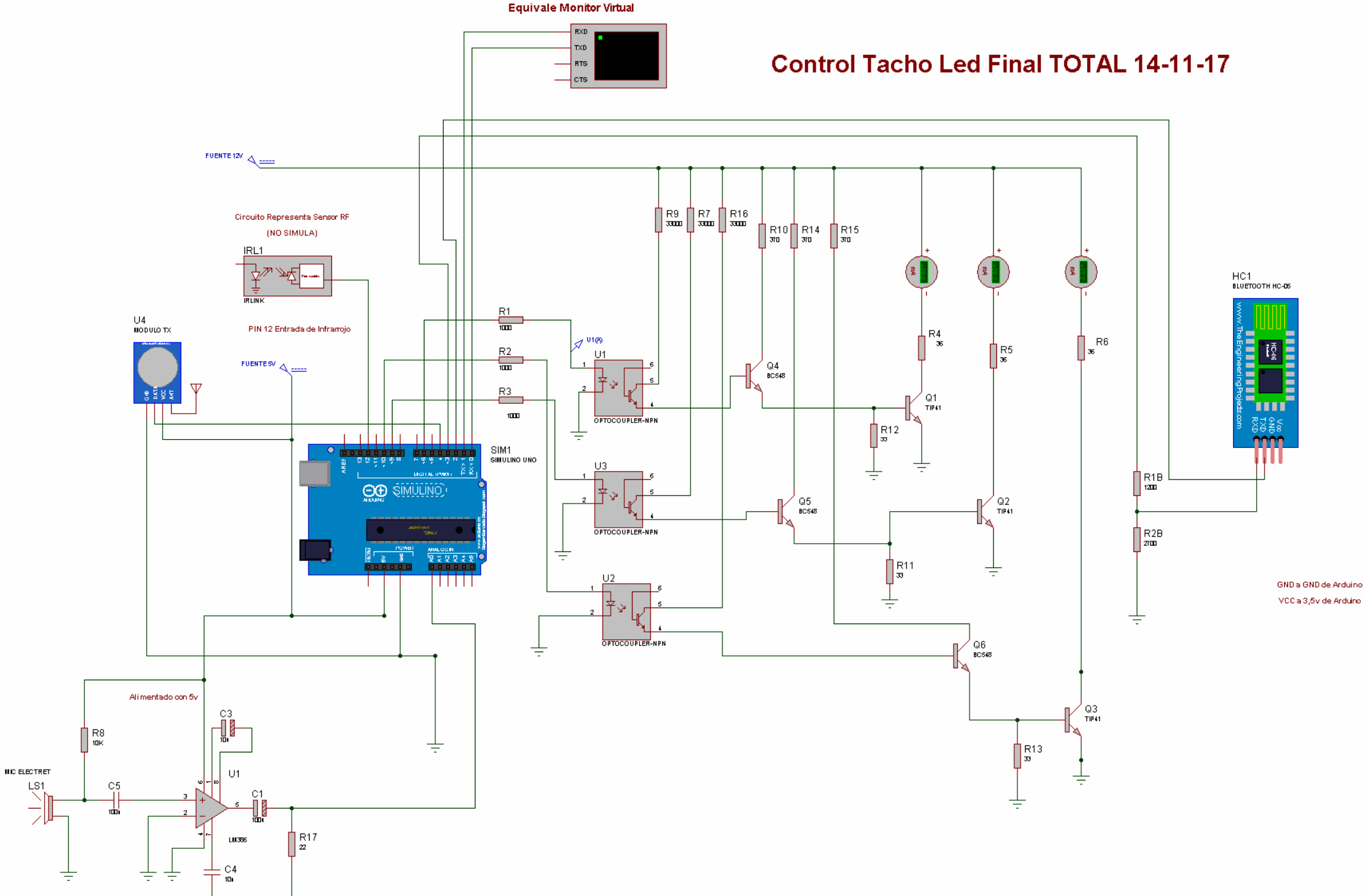
	POR	<p>Amplifier with Gain = 20 Minimum Parts</p>	
--	-----	---	--

En el esquemático total del circuito Multitacho se puede observar los componentes usados para el caso del circuito del LM386.

También se agrego la SECUENCIA 4, esta secuencia es una sucesión infinita de las otras secuencias existentes, de ella se sale con RESET POR SOFTWARE. Se restaura la función **BAILE CON EL MOVIL** que en versión anterior de Multitacho se había desactivado por falta de estabilidad del funcionamiento de la misma. Recuerde que esta función utiliza el acelerómetro del móvil para enviar valores del control de los colores. Se soluciono activando la misma solo si el TACHO PRINCIPAL esta operando y el esclavo desactivado. Además se mejoró la APP de Android (**TachoBluetooth8.apk**)

A continuación se observa el circuito total del tacho principal:

# Control Tacho Led Final TOTAL 14-11-17



## Programa Arduino del Tacho Principal

```
//REMOTO25 16-11-17
//Se agregar SECUENCIA 4 (LAZO INFINITO)
//Muchos de los comentarios son solo significativos para el autor del proyecto
//-Se cambio sensor sonido LM386 a 5v- Se busca activar baile
//FUNCIONO
//En esta version POSTERIOR A TARRGB6. Se combinara con Remoto Infrarrojo
//Avances de Remoto10
//SE CAMBIO EL PIN 12 POR EL 6 YA QUE TENIA PROBLEMAS
//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.
//Se agrega la funcion selecttacho() encargada de dirigir los datos al tacho elegido
/*Se asigna un numero de 3 digitos a cada tacho (idtacho)
*Un valor determinado que llega a todos los tachos fija la variable -tacho-
*Datoenviado variable: tacho Accion en tacho idetacho TACHO
*230 125 sigue ordenes 125 1
*235 0 deja de seguir ordenes 125 1
*240 127 sigue ordenes 127 2
*245 0 deja de seguir ordenes 127 2
*250 129 sigue ordenes 129 3
*255 0 deja de seguir ordenes 129 3
*
*El cambio de tacho se admite por 3 formas de control (si no hay incompatibilidad)
*/

//La version REMOTO23 presenta problemas con la funcion de Baile con el
//telefono. Se observa que el envio de datos necesarios es superior al
//que admite el envio por RF.
//Al hacer el RESET hay que presionar reiteradas veces para que se envíe
//la orden por RF.
//Todo se probó con RECEPTOR7-----
//CORREGIR LO ANTERIOR EN PROXIMAS VERSIONES---
//---Para Modulos RF-----
//Vamos a enviar la variable num de 3 digitos por medio de Modulo RF-----
#include <RCSwitch.h>

RCSwitch myRF = RCSwitch(); // Asignamos el nombre a la libreria para el uso
//de las funciones que esta nos permite, yo la
//he llamado myRF -Es solo un cambio de nombre

//myRF.enableTransmit(4); // Habilitamos y Establecemos el pin digital
//para el envío de datos en este caso el 4
//Se ha probado el 12 y funciona

//---Fin Modulos RF -----

//--Para Bluetooth--
//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;
int idetacho= 125; //Esta variable identifica al tacho - NUNCA CAMBIA-
int tacho = 125 ; //Este tacho que es el principal arranca aceptando ordenes
```

```

        //En los demas tacho asegurarse de fijar a 0 (cero)--
        //La variable tacho controla si el tacho debe seguir las
        //ordenes o no.
        //Si tacho = idtacho sigue ordenes---Si tacho = 0 no sigue.

int baile=0; //La variable se reserva para poder interactuar solo con Tacho principal
            //Se busca que la funcion RF multitacho se pueda desactivar
            //de modo que el programa se comporte como Remoto 23
            //baile=0 multitacho baile= 1 monotacho
            //Desde control baile= 1 se establece con num= 400 y baile=0 con num=410

//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 comunicaci3n 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 recepci3n

//Fin para infrarrojos-----
//Para Modulos RF-----
myRF.enableTransmit(4); // Habilitamos y Establecemos el pin digital
                        //para el env3o de datos en este caso el 4
                        //Se ha probado el 12 y funciona
}

//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 recepci3n
  //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 2351064443://Tecla 4 SecuenciaA4

Serial.print(results.value);

SecuenciaA4();
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 infrarrojos 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

}

//--Fin secuencia A3-----
//
//----Secuencia A4-----Se activa con num=142--

void SecuenciaA4()
{

do

```

```

    {
        SecuenciaA2();
        SecuenciaA1();
    } while (num >0);

}

//--Fin secuencia A4-----

//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 audioritmico----
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 > 350)
    AUDIOPICO();

if ((valor > 170) && (valor < 250 ))
    AUDIOMEDIO();
if ((valor > 80) && (valor < 150 ))
    AUDIOBAJO();

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

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

} //Fin funcion audioritmico

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

Color(255,0,255);//magenta

```

```

delay (200);

Color(255,255,255);//blanco
delay (200);

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

void AUDIOMEDIO()
{
Color(255,255,0);//amarillo
delay(100);
Color(0,255,0);//verde
delay(100);
Color(0,255,255);//cyan
delay (200);
Color(255,0,0);//rojo
delay (100);

Color(255,150,0);//naranja
delay (100);

Color(0,0,0);
}

void AUDIOBAJO()
{
Color(155,0,0);//rojo palido
delay(100);

Color(255,0,0);//rojo
delay(100);

Color(0,0,0);
}

void RESETTEOPC() //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

    }

}

}

}

//-----

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();
}

if (num==142)
{
    SecuenciaA4();
}

//-----
//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();
        }
//-----baile-----

        if (num ==400)
        {
            baile=1; //Monotacho
        }

        if (num ==410)
        {
            baile=0; //Multitacho
        }
//----- baile-----

    }//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(); // Antes llamaba directamente al analisis de variable num---

    selectacho();// Ahora va previamente a seleccionar tacho-----

    }//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
//-----Funcion selectacho()-----

void selectacho()

    {
        //Activa o Desactiva el tacho-----
        if(num== 230)
            tacho=125;
        if(num== 235)
            tacho=0;

//---Envio por RF el valor de la variable num-----

if (baile==0) //Si no debe bailar entonces envia num por RF
    {

        myRF.send(num, 10); // envía el dato de la variable valor1 asignando
                            //un tamaño de 10 bits- Es fundamental estimar
                            //el tamaño del numero que quiero enviar

    }

//--Fin del envio de num por RF-----

    if(tacho == idetacho)
        {
            //Llama la funcion analisisnum-----
            analisisnum();
        }
}

//-----FIN Funcion selectacho()-----

```

Puede obtener el archivo de texto de este programa en **Remoto25.txt**

Este programa es del Tacho Principal.

**ES IMPORTANTE PARA QUIEN TRATE DE INTERPRETAR EL CODIGO DEL PROGRAMA, NOTAR QUE HAY MUCHAS INSTRUCCIONES QUE FUERON DESACTIVADAS HACIENDOLAS COMENTARIOS – TENERLO EN CUENTA-**

Hay que destacar la tabla de los números de 3 cifras que controlan todo el sistema y que son transmitidos por algunas de las 3 formas mencionadas.



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
142	SECUENCIA A4	666	- AZUL

El tacho principal es el único que esta preparado para recibir ordenes por alguno de estos 3 modos, pero es posible usarlo como transito para otros canales.

### Concepto a utilizar en el programa:

Se asigna un número de 3 dígitos a cada tacho (**idtacho**)

Un valor determinado que llega a todos los tachos fija la variable -tacho- y de acuerdo a ello el tacho obedece o no las ordenes que llegan.

Ejemplo para el caso de 3 tachos, donde el 1 es el principal.

Dato enviado	variable: tacho	Acción en tacho	idetacho	TACHO
230	125	sigue ordenes	125	1
235	0	deja de seguir ordenes	125	1
240	127	sigue ordenes	127	2
245	0	deja de seguir ordenes	127	2
250	129	sigue ordenes	129	3
255	0	deja de seguir ordenes	129	3

**Esta implementado hasta el tacho 2. Recuerde que Tacho 1 es donde esta la consola y Tacho 2 lo hemos llamado Tacho Esclavo 1**

### Programa para el Tacho Esclavo 1

```
//TachoEsclavo1B
//Se agrega SecuenciaA4 que son A1 y A3 en loop--
//Se va a llevar la seccion de analisis de la variable global (num) a una funcion
//llamada analisisnum()

//Se agrega la funcion selecttacho() encargada de dirigir los datos al tacho
elegido
/*Se asigna un numero de 3 digitos a cada tacho (idtacho)
*Un valor determinado que llega a todos los tachos fija la variable -tacho-
*Datoenviado variable: tacho Accion en tacho idetacho TACHO
*230 125 sigue ordenes 125 1
*235 0 deja de seguir ordenes 125 1
*240 127 sigue ordenes 127 2
*245 0 deja de seguir ordenes 127 2
*250 129 sigue ordenes 129 3
*255 0 deja de seguir ordenes 129 3
*
*El cambio de tacho se admite por 3 formas de control (si no hay incompatibilidad)
*/

//---Para Modulos RF-----
```

```

//Vamos a enviar la variable num de 3 digitos por medio de Modulo RF-----
#include <RCSwitch.h>

    RCSwitch myRF = RCSwitch(); // Asignamos el nombre a la libreria para el uso
                                //de las funciones que esta nos permite, yo la
                                //he llamado myRF -Es solo un cambio de nombre

//---Fin Modulos RF -----

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

int red=0,green=0,blue=0;
int num;
int idetacho= 127; //Esta variable identifica al tacho - NUNCA CAMBIA-
int tacho = 0 ; //Este tacho es el TACHO 2 arranca NO aceptando ordenes
                //En los demas tacho asegurarse de fijar a 0 (cero)--
                //La variable tacho controla si el tacho debe seguir las
                //ordenes o no.
                //Si tacho = idtacho sigue ordenes---Si tacho = 0 no sigue.
int datob; // variable para el tamaños del dato a recibir por RF

//Fin de tachos-----

void setup()
{

//Para tachos-----
//Paneo de canales---
    analogWrite(5, 255);
    delay(1000);
    analogWrite(5, 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 Modulos RF-----
Serial.begin(9600); //Iniciamos comunicación con el puerto serie
myRF.enableReceive(0); /* Habilitamos el receptor en la interrupción 0 => Pin 2
Digital(Para Arduino UNO) */

}

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

void loop() {

    if (myRF.available())
    {
        datob = myRF.getReceivedBitlength(); //Mediante esta función leemos el
                                             //tamaño del paquete que estamos
                                             //recibiendo y lo almacenamos en
                                             //la variable datob, de esta forma
                                             //hacemos una criba de los paquetes

```

```

//recibidos en función de su
//tamaño. En este caso es 10 Bits.

if(datob==10)
{
    num =myRF.getReceivedValue(); // almacenamos en la variable num el dato
recibido

}

//Serial.println(num); //Muestra dato que llego por Monitor Serie
myRF.resetAvailable(); // para terminar resepcion hacemos un reset
selectacho();

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

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

//*****
//Funciones creadas-----

void Menos_blue()
{
    blue = blue - 5;
    if (blue == -5)
    {
        blue = blue + 5;
    }
    analogWrite(6, blue) ;// blue -blue PIN 6
}
//-----

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(5, red) ; // Rojo PIN 5
}

//-----
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(5, red) ;    // Rojo  PIN 5
}

//-----

void Color(int R, int G, int B)
{
    analogWrite(5 , 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);

        RESETEOTACHO();

    }
    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);

        RESETEOTACHO();

    }
}

```

```

    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);

        RESETTEOTACHO();

    }

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

    RESETTEOTACHO();

}

}

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

    RESETTEOTACHO();

}

}

void SecuenciaA3()
{
    for (int i =0 ; i<255 ; i++)
    {

```

```

    Mas_red();//Sube a maximo rojo
    delay(2);
    Color(0, 0, 0); //negro
    delay(20);

    RESETTEOTACHO();

}

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);

    RESETTEOTACHO();

}

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);

    RESETTEOTACHO();

}
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);

    RESETTEOTACHO();

}
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);

        RESETTEOTACHO();

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

}
//---Fin SecuenciaA3-----

//---Secuenica A4-----Se activa con num=142-----

void SecuenciaA4()
{
    do
    {
        SecuenciaA2();
        SecuenciaA1();
    } while (num >0);

}

//--Fin secuencia A4-----

//Funcion audiritmico----
void AUDIORITMO()
{
int AUD=1;// Esta variable es para entrar a WHILE, nunca cmabia

    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

        RESETTEOTACHO();

        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 analisisnum()
{
//Colores fijos

    if (num==700)//Negro
        {

            analogWrite(5, 0) ; // Rojo PIN 5
            analogWrite(10, 0) ; // Green - Verde PIN 10
            analogWrite(6, 0) ; // blue - blue PIN 6
            red= 0;
            green= 0;
            blue= 0;
        }

    if (num==750) //Rojo
        {

            analogWrite(5, 255) ; // Rojo PIN 5
            analogWrite(10, 0) ; // Green - Verde PIN 10
            analogWrite(6, 0) ; // blue - blue PIN 6
            red= 255;
            green= 0;
            blue= 0;
        }

    if (num==800) //Verde
        {

            analogWrite(5, 0) ; // Rojo PIN 5
            analogWrite(10, 255) ; // Green - Verde PIN 10
            analogWrite(6, 0) ; // blue - blue PIN 6
            red= 0;

```



```

        green= 255;
        blue= 0;
    }

    if (num==850) //blue
    {

        analogWrite(5, 0) ; // Rojo PIN 5
        analogWrite(10, 0) ; // Green - Verde PIN 10
        analogWrite(6, 255) ; // blue - blue PIN 6
        red= 0;
        green= 0;
        blue= 255;
    }

    if (num==900) //Blanco
    {

        analogWrite(5, 255) ; // Rojo PIN 5
        analogWrite(10, 255) ; // Green - Verde PIN 10
        analogWrite(6, 255) ; // blue - blue PIN 6
        red= 255;
        green= 255;
        blue= 255;
    }

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

        analogWrite(5, 255) ; // Rojo PIN 5
        analogWrite(10, 0) ; // Green - Verde PIN 10
        analogWrite(6, 255) ; // blue - blue PIN 6
        red= 255;
        green= 0;
        blue= 255;
    }

    if (num==150) //Amarillo
    {

        analogWrite(5, 255) ; // Rojo PIN 5
        analogWrite(10, 255) ; // Green - Verde PIN 10
        analogWrite(6, 0) ; // blue - blue PIN 6
        red= 255;
        green= 255;
        blue= 0;
    }

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

        analogWrite(5, 0) ; // Rojo PIN 5
        analogWrite(10, 255) ; // Green - Verde PIN 10
        analogWrite(6, 255) ; // blue - blue PIN 6
        red= 0;
        green= 255;
        blue= 255;
    }

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

```

```

        {
            SecuenciaA2();
        }

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

if (num==142)
    {
        SecuenciaA4();
    }

//-----
//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 selectacho()-----

```

```

void selectacho()

    {
        //Activa o Desactiva el tacho-----
        if(num== 240)
            tacho=127;
        if(num== 245)
            tacho=0;

        if(tacho == idetacho)
            {
                //Llama la funcion analisisnum-----
                analisisnum();
            }

    }

//-----FIN Funcion selectacho()-----

void RESETEOTACHO() //Funcion reseteo por orden numerica
{
    if (myRF.available())
        {
            datob = myRF.getReceivedBitlength(); //Mediante esta función leemos el
                                                //tamaño del paquete que estamos
                                                //recibiendo y lo almacenamos en
                                                //la variable datob, de esta forma
                                                //hacemos una criba de los paquetes
                                                //recibidos en función de su
                                                //tamaño. En este caso es 10 Bits.

            if(datob==10)
                {

                    num =myRF.getReceivedValue(); // almacenamos en la variable num el dato
recibido

                }

            //Serial.println(num); //Muestra dato que llego por Monitor Serie

            myRF.resetAvailable(); // para terminar resepcion hacemos un reset

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

            if (num==999)
                {

                    resetFunc(); // llamo funcion especifica de reseteo

                }

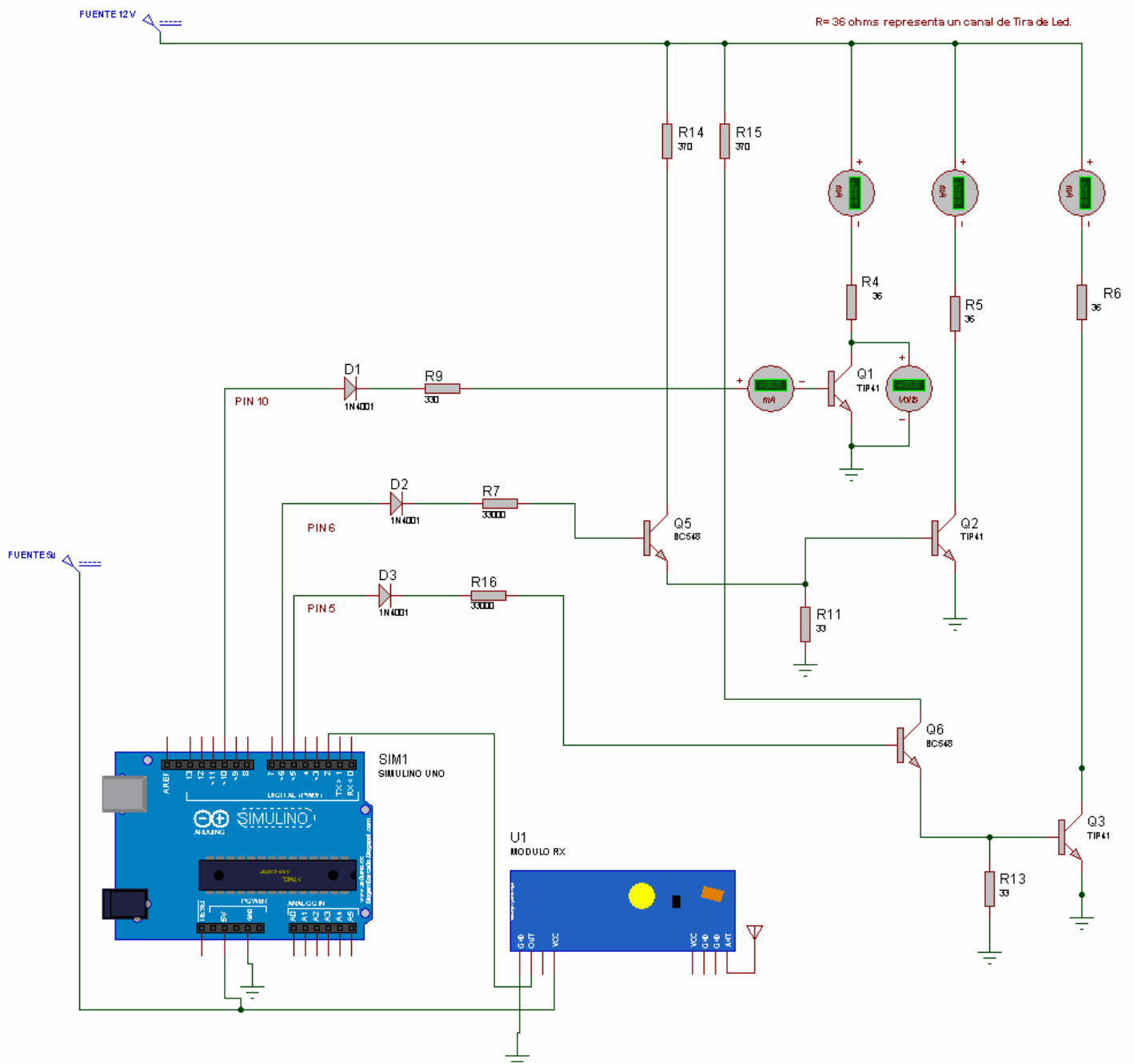
}

} //Fin RESETEOTACHO
}

```

**ES IMPORTANTE PARA QUIEN TRATE DE INTERPRETAR EL CODIGO DEL PROGRAMA, NOTAR QUE HAY MUCHAS INSTRUCCIONES QUE FUERON DESACTIVADAS HACIENDOLAS COMENTARIOS - TENERLO EN CUENTA-**

A continuación observamos el plano del tacho esclavo:



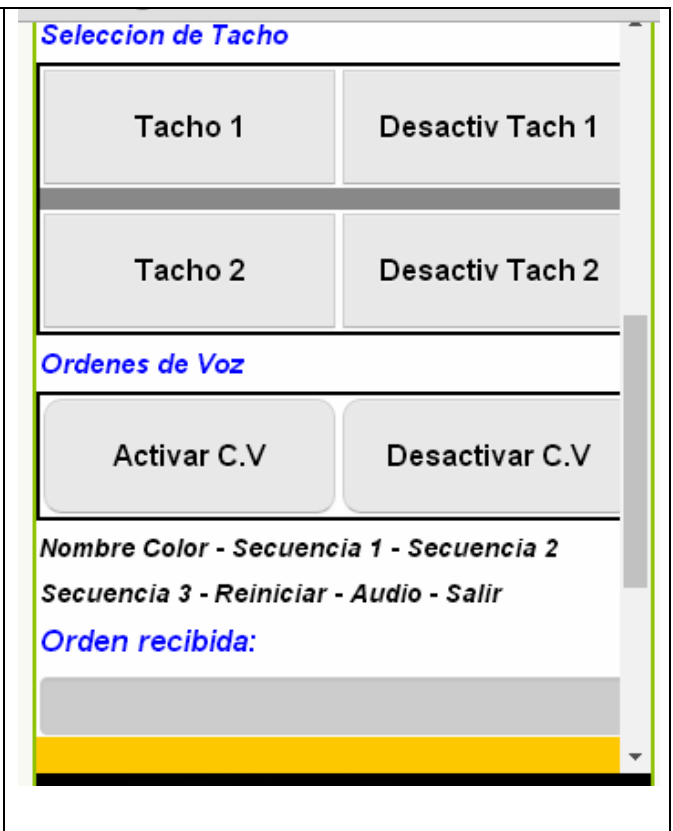
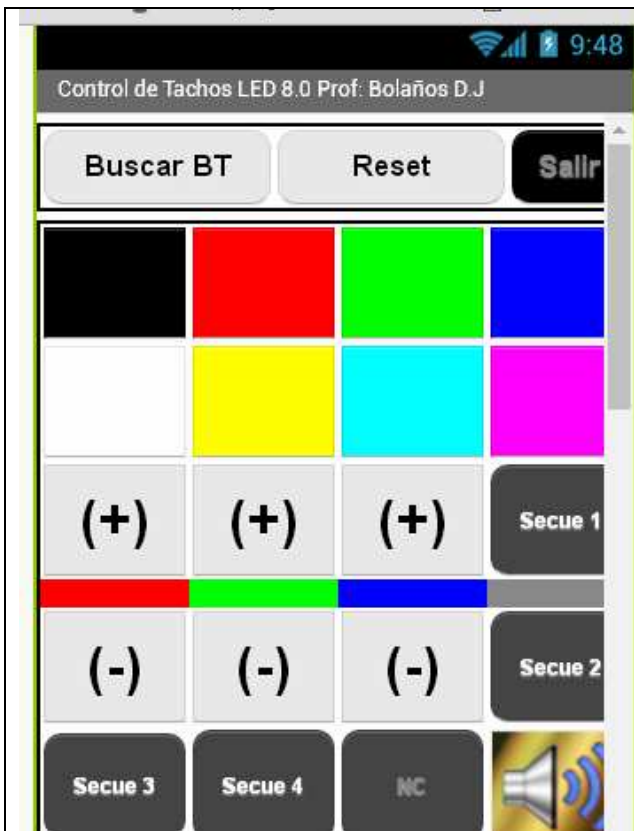
## APP Android Multitacho.

Ahora vamos a mostrar la APP para Android:.. La APP que se utiliza con REMOTO 25 es:

**TachoBluetooth8.apk**

Nota: Al cargar la APP le aparecerá como nombre de la misma TacLedBT 8.

Se mostrara la APP en 3 capturas por su característica scrollable.



Se agrego con respecto a versión anterior, Botón Salir. Se habilito la función Baile con el equipo, aunque solo es visible en Tacho 1. Se agrego Secuencia 4 que es una ejecución continua de Secuencia1, Secuencia2.y Secuencia 3.

Tanto el archivo APK como el fuente AIA los puede encontrar en la carpeta de RECURSOS.

En la figura siguiente se observa la captura del diagrama en bloques, en archivo aparte también aparecerá dicha figura.

