Modulos RF con Arduino (*Versión: 02-12-17*)

Arduino y radiofrecuencia 433MHz con VirtualWire

FUENTE: https://www.minitronica.com/arduino-radiofrecuencia-433mhz-virtualwire/

En este tutorial aprenderemos a **conectar 2 Arduinos de forma inalámbrica usando 2 módulos de radiofrecuencia de 433MHz**. Estos módulos se venden en parejas (un emisor y un receptor) y por muy poco dinero permiten la comunicación sin cables entre dispositivos en un rango de hasta 100 metros si no hay obstáculos por medio.
Lo que haremos será enviar una señal de un **Arduino** (emisor) a otro (receptor) para que este último encienda o apague el led que tiene en el pin 5 según la orden recibida. Esto nos dará las bases para poder desarrollar proyectos más complicados basados en estos **módulos de radiofrecuencia**.

En este mismo apunte mas adelante desarrollaremos otro ejemplo para el envío de números mediante el uso otra librería (RCSwitch.h)

Usaremos la librería **VirtualWire** de Mike McCauley, esta librería se encarga de gestionar las funciones de los **módulos RF** como envío y recepción de paquetes de datos, comprobación de errores, etc… Para iutilizarla descargamos el paquete desde su web y la instalamos como ya sabemos de otras ocaciones.

Recuerde que en la carpeta del soft del docente hay una subcarpeta (BIBLIOTECAS) donde también podrá encontrarla.

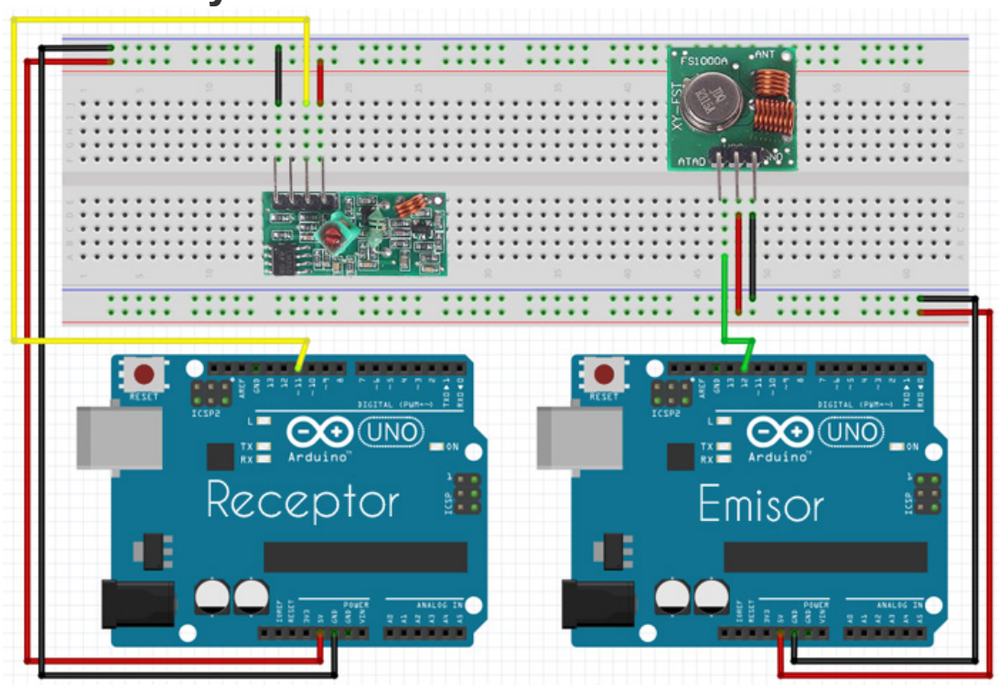
NOTA: Se incluye en otro PDF mas información sobre esta librería.

Componentes necesarios para el proyecto

Para el desarrollo de este proyecto necesitaremos los siguientes componentes:

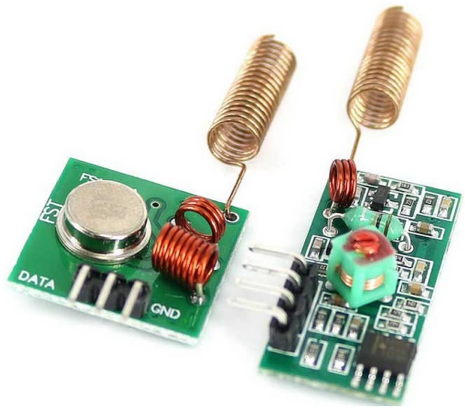
- 2x Arduino (Nano, Mega, UNO,…)
- 1x Par de emisor-receptor RF @ 433MHz
- 1x Breadboard grande
- 2x Baterías o pilas

Cableado y conexiones



Recuerde conectar un LED con su resistencia, en el PIN 5 de Arduino.

El conexionado del emisor es muy simple, solo hay que conectar 3 cables. Alimentamos el **emisor RF de 433MHz** desde la salida de 5v del Arduino y GND y conectamos el pin DATA del emisor al pin Digital 12 del Arduino. Para el **receptor RF** hacemos lo mismo, lo alimentamos desde la salida de 5v del Arduino y negativo (GND), luego conectamos el pin DATA del receptor al pin Digital 11 del **Arduino**.



CONSEJO: Si nos fijamos en las placas del receptor y del emisor veremos que tienen un pequeño agujero en una de las esquinas que esta marcado como ANT (antena), ahí podemos soldar un pequeño cable para mejorar la calidad y el alcance de la **transmisión**.

Sketch del emisor

Este es el sketch del emisor, como podéis ver envía “Encender” y “Apagar” en intervalos de 1 segundo. (Archivo EMISOR1)

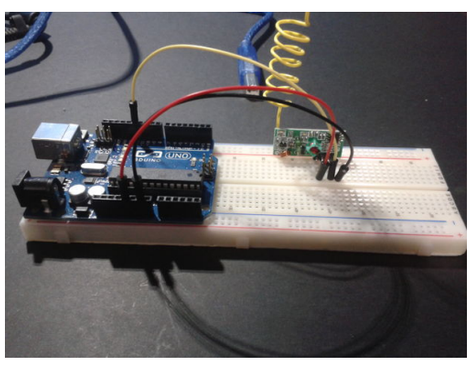
```
/*
EMISOR1

Envia los mensajes "Apagar" y "Encender" en intervalos de un segundo

*/

#include <VirtualWire.h>

void setup()
{
  //Iniciamos el Serial y la comunicacion por radio
  Serial.begin(9600);
  Serial.write("Sistema encendido\n");
  vw_setup(2000);
}
void loop()
{
  send("Encender");
  delay(1000);
  send("Apagar");
  delay(1000);
}

//Funcion para enviar el mensaje
void send (char *message)
{
  vw_send((uint8_t *)message, strlen(message)); //Envia el mensaje
  vw_wait_tx(); //Espera hasta que se haya acabado de transmitir todo

  Serial.println(message); //Muestra el mensaje por Serial
}
```

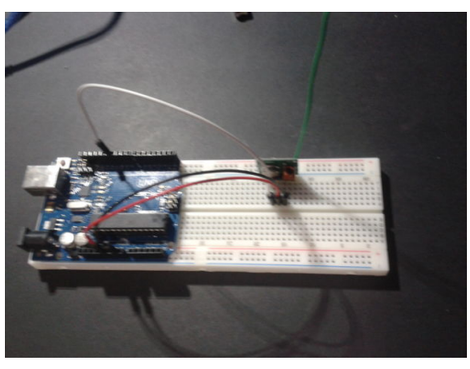
Sketch del receptor

Gracias al módulo RF recibimos un montón de carácteres que se guardan en un array, entonces comprobamos si ese array es igual a ENCENDER o APAGAR y procedemos a encender o apagar el led en pin digital 5 del Arduino.

```
/*
RECEPTOR1
 Apaga el LED 5 si recibe el mensaje "Apagar"
Enciende el LED 5 si recibe el mensaje "Encender"
*/
 #include <VirtualWire.h>
//Creamos un mensaje
//La constante VW_MAX_MESSAGE_LEN viene definida en la libreria
byte message[VW_MAX_MESSAGE_LEN];
byte messageLength = VW_MAX_MESSAGE_LEN;
void setup()
{
  pinMode(5, OUTPUT); //Configuramos el pin 5
  Serial.begin(9600);//Iniciamos el Serial
  Serial.println("Iniciando...");
  vw_setup(2000);
  vw_rx_start();
}
void loop()
{
  if (vw_get_message(message, &messageLength))
  {
      if(comparar("Encender") == 0){
        digitalWrite(5, HIGH);
        Serial.write("LED Encendido\n");
      }
      else if(comparar("Apagar") == 0)
      {
        digitalWrite(5,LOW);
        Serial.write("LED Apagado\n");
      }

  }
}

char comparar(char* cadena) {
  //Esta funcion compara el string cadena con el mensaje recibido.
  //Si son iguales, devuelve 1. Si no, devuelve 0.

  for(int i = 0; i<messageLength; i++)
  {
    if(message[i] != cadena[i])
    {
      return 1;
    }
  }

    return 0;
}
```

Y con esto ya lo tendríamos todo listo para alimentar los **Arduinos** y ver como se enciende y apaga el LED. En la página web del autor podéis ver una lista completa de las funciones que implementa la librería, se incluye el documento traducido junto a este material.

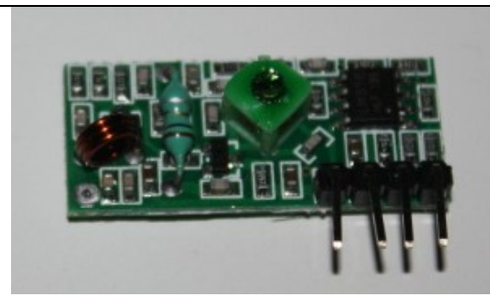




Repaso de Conexiones

Las conexiones de cada una de las placas Arduino con los módulos serán:

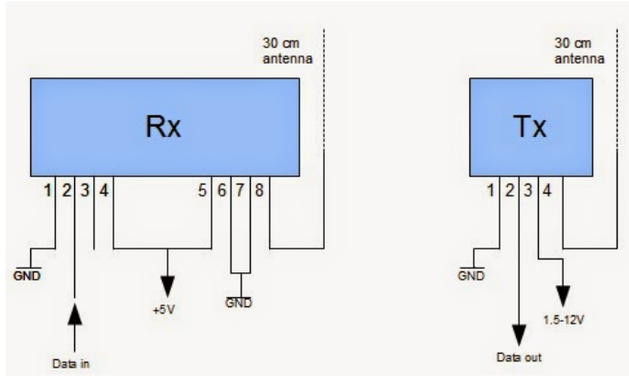
EMISOR : -Arduino 5V – Módulo VCC -Arduino GND – Módulo GND -Arduino Digital 12 – Módulo DATA	

RECEPTOR: -Arduino 5V – Módulo VCC -Arduino GND – Módulo GND -Arduino Digital 11 – Módulo DATA (tiene dos pines DATA: cualquiera de los dos vale) -Arduino VCC – Positivo -Arduino GND – Negativo	

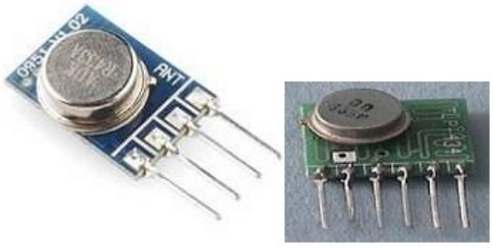
Desarrollo con el uso de otra biblioteca

Enviando datos a través de modulo RF 433Mhz

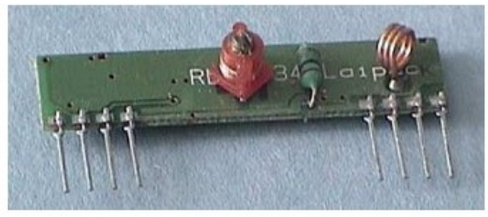
Notemos que usaremos módulos de otro fabricante, los cuales demostraron ser mas eficientes y además tienen un PIN para incluirles la antena.



MODULO EMISOR



MODULO RECEPTOR



*Frequency Range: 433.92 MHZ
*Modulate Mode: ASK
*Circuit Shape: LC
*Date Rate: 4800 bps
*Selectivity: -106 dB
*Channel Spacing: 1MHZ
*Supply Voltage: 5V
* High Sensitivity Passive Design.
*Simple To Apply with Low External Count.

Pin Assignment

Pin	Function
1	GND
2	Data in
3	Vcc
4	ANT

Ya hicimos una práctica enviando mensajes a través de un modulo de RadioFrecuencia y dos Arduinos. Pero en ese caso simplemente estábamos enviando un mensaje de tipo char mediante la librería **VirtualWire.h.**
Lo que vamos a hacer ahora es totalmente distinto ya que vamos a enviar datos de tipo Int empleando la librería **RCSwitch.h.**
<RCSwitch.h> Es una librería de Arduino que nos permite el envío y recepción de datos vía RF con módulos de 315/433MHz.

Podemos descargarla en el siguiente link:

https://code.google.com/p/rc-switch/downloads/detail?name=RCSwitch.zip&can=3&q=

Una vez descargada la instalada en nuestro Arduino IDE y nos ponemos manos a la obra.

NOTA: Recuerde que en la carpeta BIBLIOTECAS del soft del docente podra encontrar un copia de las bibliotecas empleadas.

Desarrollo del ejemplo EMISOR7 - RECEPTOR7

Nos proponemos un ejercicio donde mediante el Monitor Serie del Emisor tipearemos un valor numérico no mayor a 1023 (ya que estableceremos en envío de datos de hasta 10 bits).

El número escrito se carga en una variable **valor1** para luego ser enviada al RECEPTOR

Sketch del emisor

EMISOR7

```
//EMISOR7 Envia un numero desde el Monitor Serie que luego se recibe en
RECEPTOR7 y

//se muestra en Monitor Serie de Arduino con RECEPTOR7

#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

int valor1=0; //me interesa solo usar valor1 como variable
```

```
void setup()
{
  Serial.begin(9600);
  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
}
void loop()
 {
//Se van a tomar los valores que escribo en Monitor Serial
   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();
    }
  valor1 = bufferString.toInt(); //Se transforma el buffer a un número entero
                                 //Se carga lo leido en la variable valor1
    }
 if(valor1 > 0) //Va a enviar el numero contenido en valor1 si es > 0
    {
```

```
myRF.send(valor1, 10); // envía el dato de la variable valor1 asignando
                       //un tamaño de 10 bits- Es fundamental estimar
                       //el tamaño del numero que quiero enviar
Serial.println("Enviado");//Muestro en Monitor Serie (del EMISOR)
Serial.println(valor1);//Muestro en Monitor Serie (del EMISOR)
delay(1000);//Dejamos un tiempo
valor1=0;//Se hace valor1 cero para que no se siga mostrando en Monitor Serie
 }
}
```

Destaquemos lo importante:

```
RCSwitch myRF = RCSwitch();

/* Asignamos el nombre a la librería para el uso de las funciones que esta nos permite,
yo la he llamado myRF */
```

```
myRF.enableTransmit(4);

/* Habilitamos y Establecemos el pin digital para el envío de datos en este caso el  */
//DEBE IR DENTRO DE  void setup()  ya  que se trata de configuración..
```

```
myRF.send(valor1, 10);

// envía el dato de la variable valor1 asignando un tamaño de 10 bits
//Solo podremos enviar números hasta 1023 en decimal
```

Sketch del receptor

RECETOR7

En el receptor se reciben el dato y se muestra en Monitor Serie. Solo si llega 125 enciende el LED verde, otro valor lo apaga.

```
//Muestra por Monitor Serie los numeros que le llega desde EMISOR7

//Este programa admite el envio de numeros de hasta 10 bits y mayores que 0

//Por lo tanto el mumero mas grande que se puede enviar es 1023

#include <RCSwitch.h>

RCSwitch myRF = RCSwitch();

int datob,valor1;  // definimos las variables

void setup()

{

   pinMode(5, OUTPUT); // configura 'pin' como salida.LED testigo



    Serial.begin(9600);

    myRF.enableReceive(0);

 /* Habilitamos el receptor en la interrupción 0 => Pin 2 Digital(Para Arduino
UNO) */

}

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)

        {

   valor1=myRF.getReceivedValue();

                               // almacenamos en la variable valor1 el dato recibido

     if (valor1 >0 )

          {

      Serial.println("Valor1");
```

```
      Serial.println(valor1);

          }

       }

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

//Le agrego led testigo- Enciende cuando llega  125

if (valor1==125)

   {

digitalWrite(5, HIGH);

   }

else

  {

digitalWrite(5, LOW);

  }

   }

delay(100);

}
```

Destaquemos lo importante:

```
myRF.enableReceive(0);

 /* Habilitamos el receptor en la interrupción 0 => Pin 2 Digital(Para Arduino UNO) */
```

```
 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 10 Bits en nuestro ejemplo. */
```

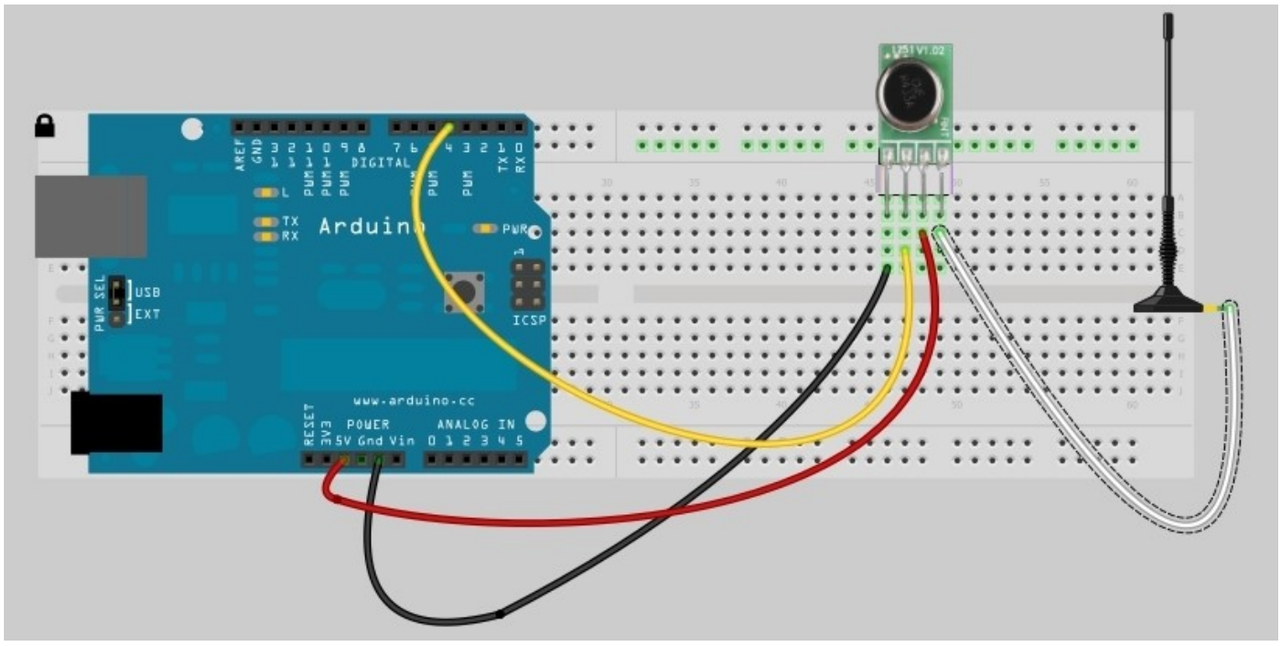
```
 valor1=myRF.getReceivedValue();

// almacenamos en la variable valor1 el dato recibido
```

```
 myRF.resetAvailable();

 // para terminar hacemos un reset- Detiene lectura hasta próxima llegada
```

CIRCUITO

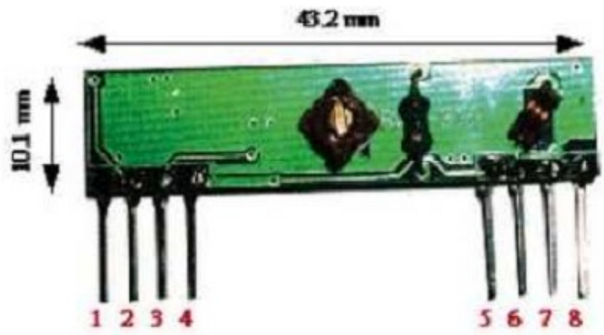
Recordar colocar un cable de 30 cm como antena tanto para el EMISOR como para el RECEPTOR.

EMISOR



PIN	MODULO	ARDUINO 1
1	GND	GND
2	DATA IN	PIN 4
3	VCC	5V
4	ANT	CABLE DE 30 cm

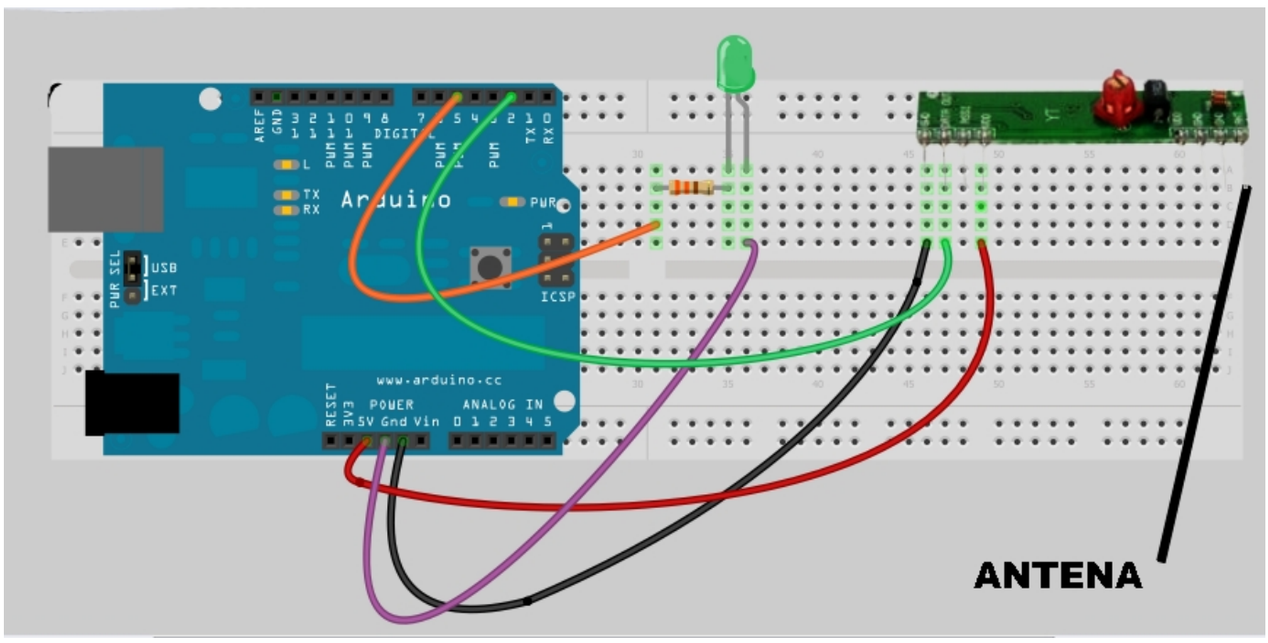
RECEPTOR



pin 1 : Gnd
pin 2 : Digitak Output
pin 3 : Linear Output
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 :ANT (About 30 - 35 cm)

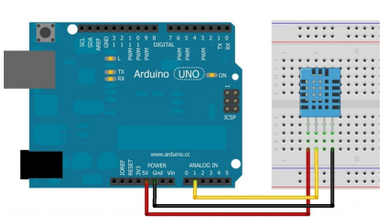
Moulation : AM
Supply Voltage : 5v dc

PIN	MODULO RECEPTOR	ARDUINO 1
1	GND	GND
2	DIGITAL OUTPUT	PIN 2
3	LINEAR OUTPUT	NINGUNO
4	VCC	5V
5	VCC	NINGUNO
6	GND	NINGUNO
7	GND	NINGUNO
8	ANT	CABLE DE 30 cm



EJEMPLO 2 PROPUESTO

SENSOR DE TEMPERATURA Y HUMEDAD INALAMBRICO.



EJEMPLO 3

TACHO LED VARIOS CANALES

Se propone partiendo del EJEMPLO TACHO LED 3 CONTROLES (PC – INFRARROJO – ANDROID), comandar otros tachos LED independientes entre si.

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

REMOTO23

En este programa agregamos la capacidad de enviar las órdenes que le llegan al tacho principal y dirigirlas a alguno de los tachos o a todos a la vez.

```
//REMOTO23

//FUNCIONO

//En esta version TARRGB6. Se combinara con Remoto Infrarrojo

//Avances de Remoto10

//SE CAMBIO EL PIN 12 POR EL 6 PAEA EL RGB 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 selectacho()  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)

*/

//Esta 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 envie

//la orden por RF.

//Todo se probo 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 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;
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.
//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-----

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

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

 }

//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 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
    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)
            {
```

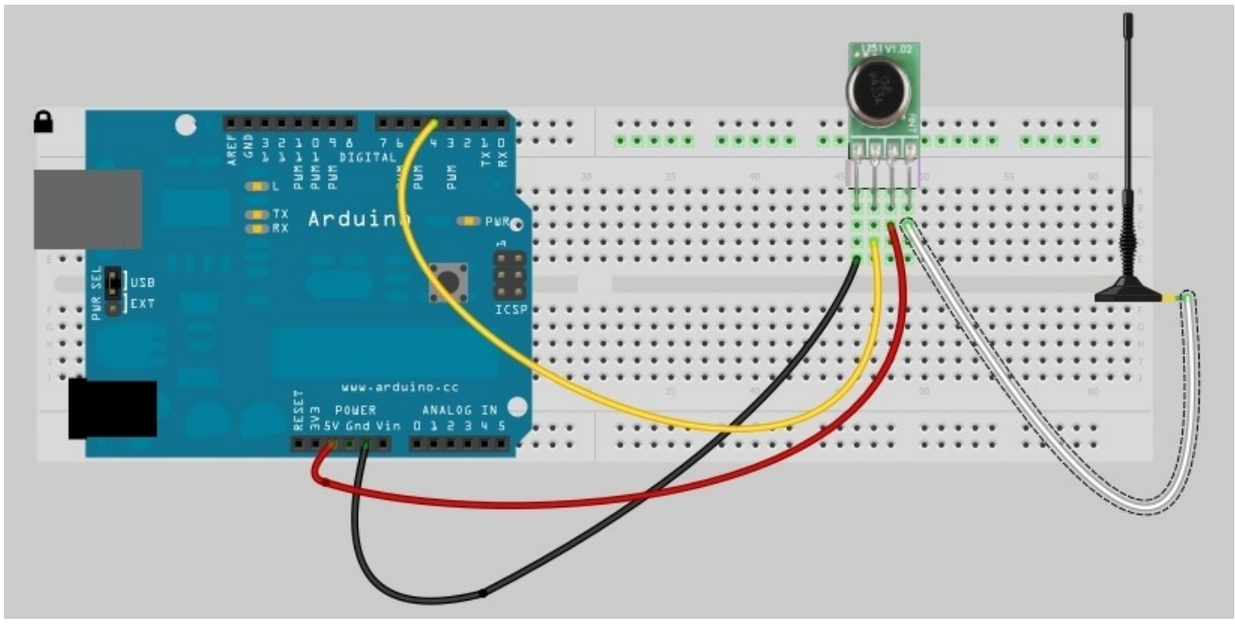
```
            Menos_green();
            }
         if (num ==666)
            {
            Menos_blue();
            }
//Fin bajando intensidad de a pasos-----------------
          if (num ==888)
            {
            AUDIORITMO();
            }
   }//llave fina de  funcion analisisnum---
//Fin 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(); // 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)
```

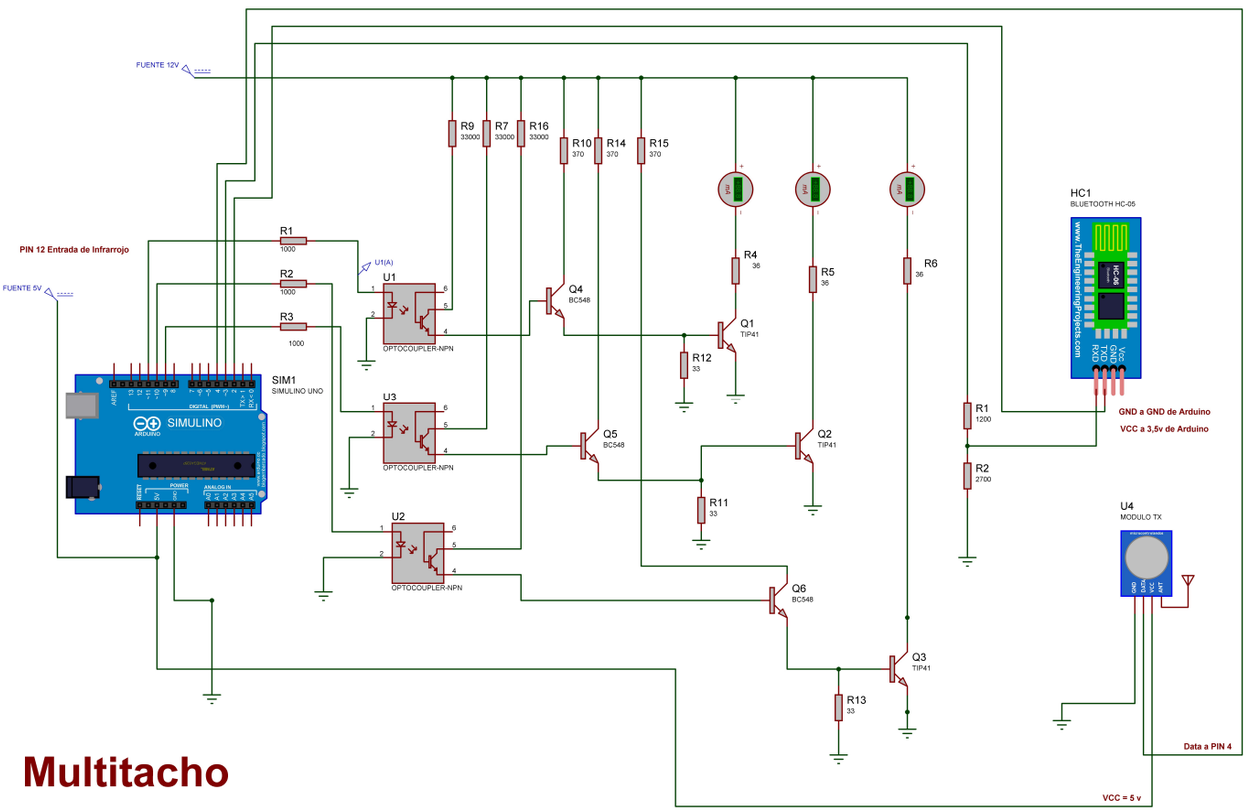
```
                tacho=125;

        if(num== 235)

                tacho=0;



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

 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()--------
```

Para el envío de datos al Modulo transmisor de RF se utiliza el PIN4 del Arduino.

La siguiente figura muestra solo la conexión al modulo RF.



Circuito total con modulo RF



Multitacho

TACHOESCLAVO1A

Este es programa que se debe cargar en el Arduino del Tacho1. Los demas tachos tienen programas similares, pero con los códigos correspondientes a esos Tachos.

```
//TachoEsclavo1A

//Se va a llevar la seccion de analisis de la variable global (num) a una funcion

//llamada analisisnum()

//Se agrega la funcion selectacho()  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                12             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);
            RESETEOTACHO();
            }
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
          RESETEOTACHO();
         }
   }


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
          RESETEOTACHO();
         }


   }
```

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

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

    RESETEOTACHO();

    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();
            }
```

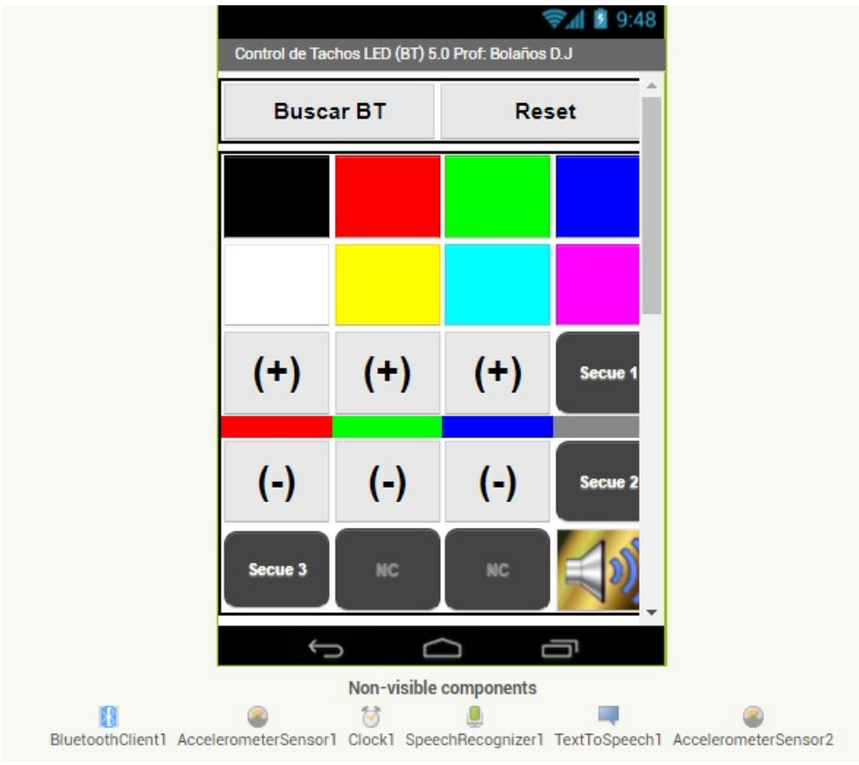
```
  if (num==140)
            {
             SecuenciaA3();
            }
//-------------------------------------------
//Se ordena el encendido de los LEDs por pasos
//Va de 0 a 255 con pasos de 5
// Subiendo intensidad----------
        if (num==111)
            {
             Mas_red();
            }
        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
}
```

A continuación desarrollaremos la APP con la variante necesaria para controlar el Tacho principal y los Tachos Esclavos.
(Se desactivará la función de Baile con el Teléfono dado que no es eficiente el envio de datos desde esa función, se tratará en futuras versiones volver a activarla).

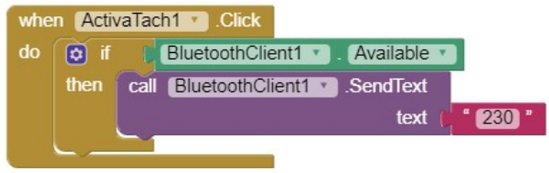
DISEÑO DE LA APP TachoBluetooth5.apk

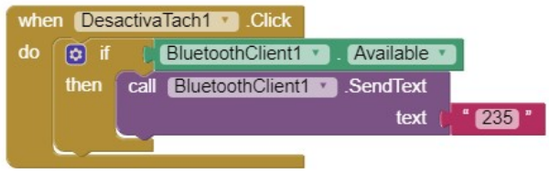


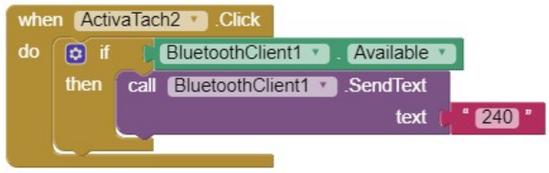
NOTA: Recuerde que el archivo APK de la APP y el fuente de la misma AIA estan disponibles dentro del Tutor de Arduino (RECURSOS).

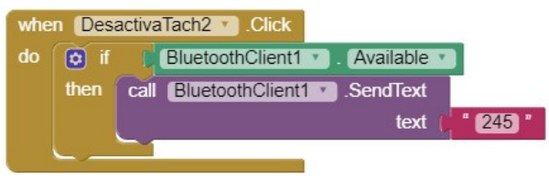
A la APP **TachoBluetooth4.apk** que se puede repasar en el Tutor de Arduino, se le agrega los botones para la selección de Tacho.

Así el diagrama de bloques adicional seria el siguiente:

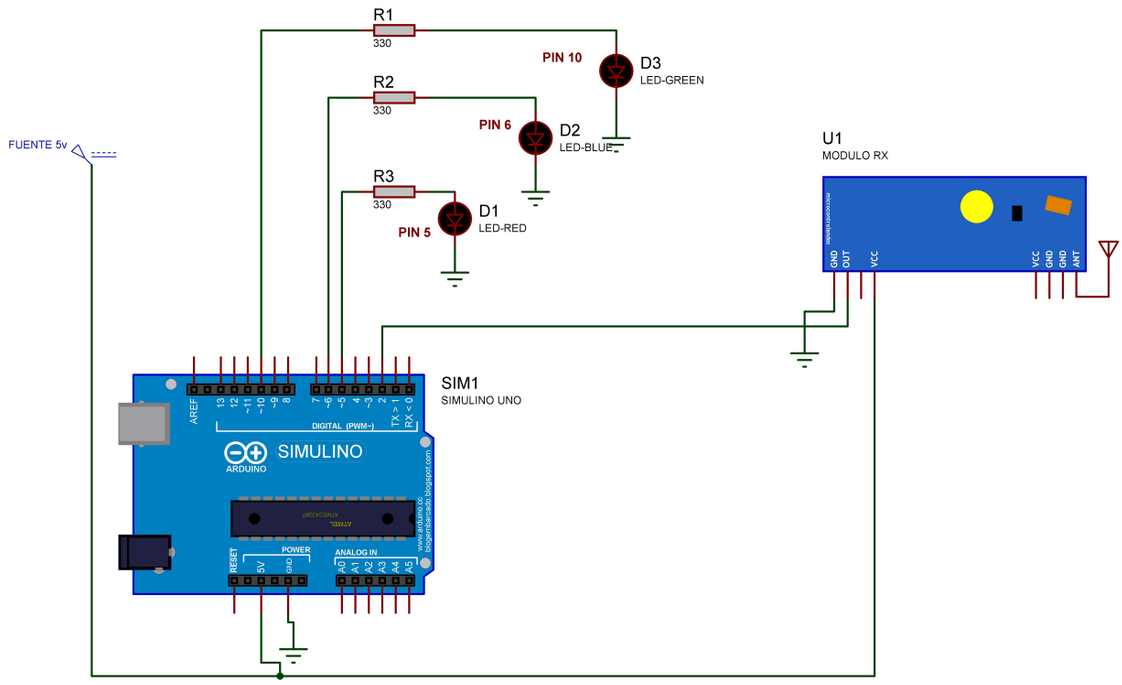




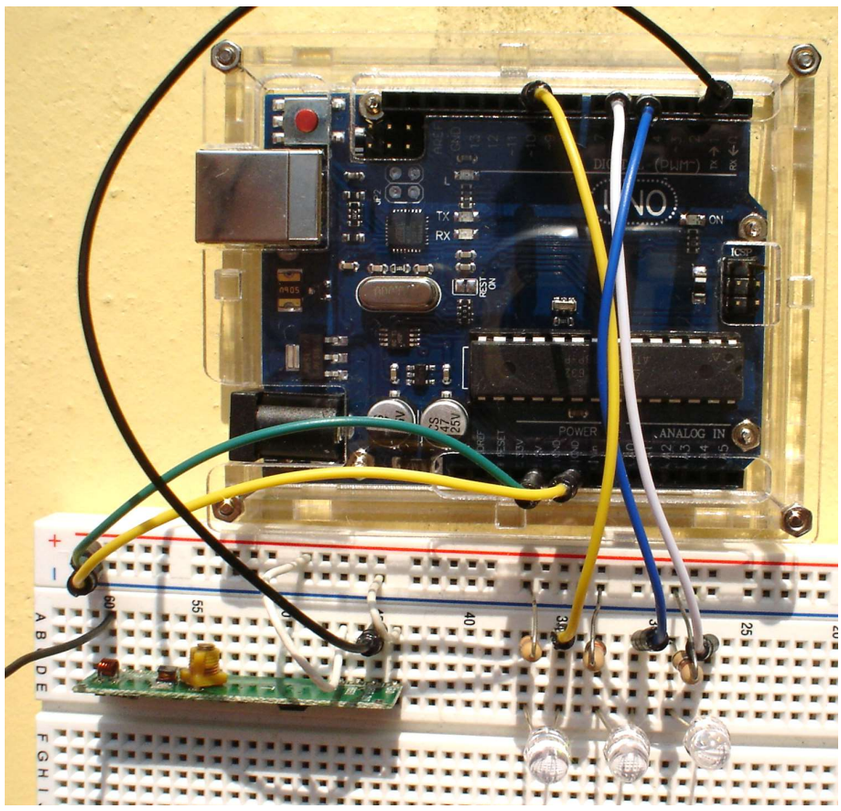


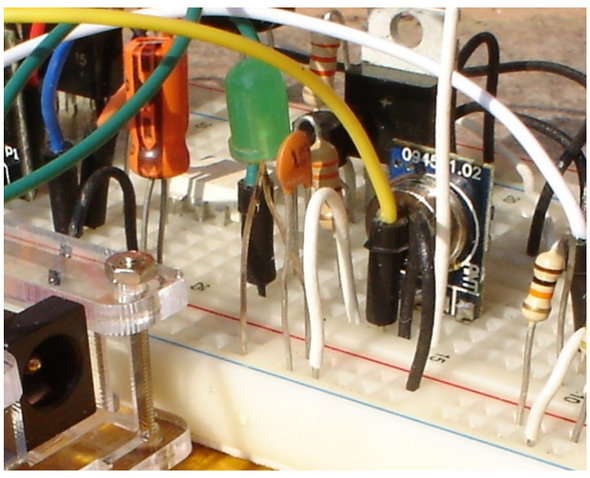


El circuito de tacho LEDs ESCLAVO seria el siguiente:



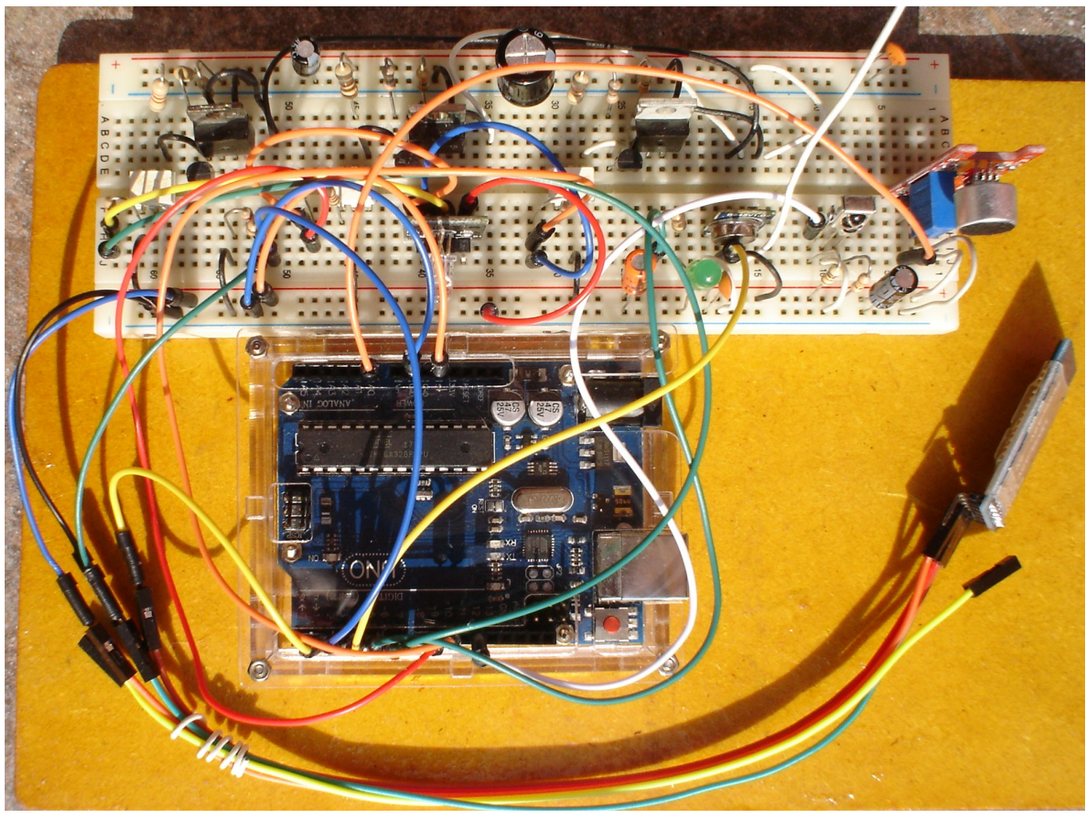
Fotografía del circuito receptor real





Acercamiento al modulo transmisor

Fotografía circuito completo del tacho principal incluyendo al transmisor.



Continuara…