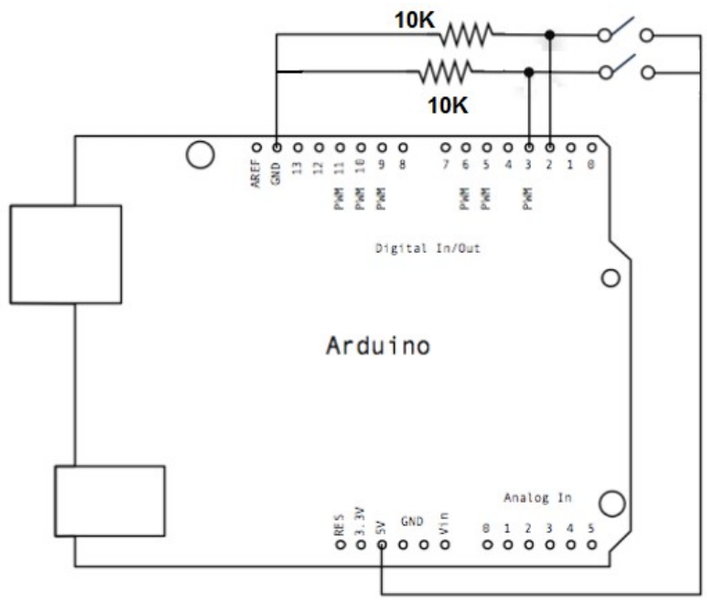
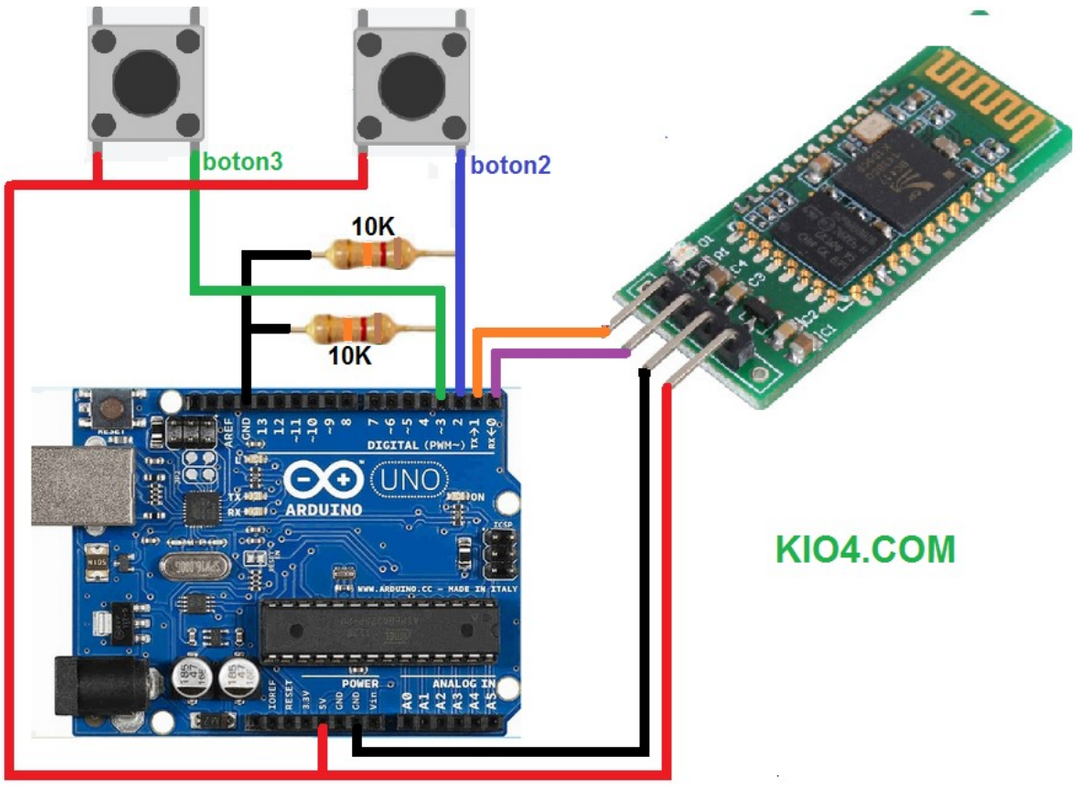
RECIBIENDO DE ARDUINO POR BLUETOOTH *(Prof: Bolaños DJ – 15-6-18)*

Conexionado del módulo Bluetooth, el Arduino y el pulsador.



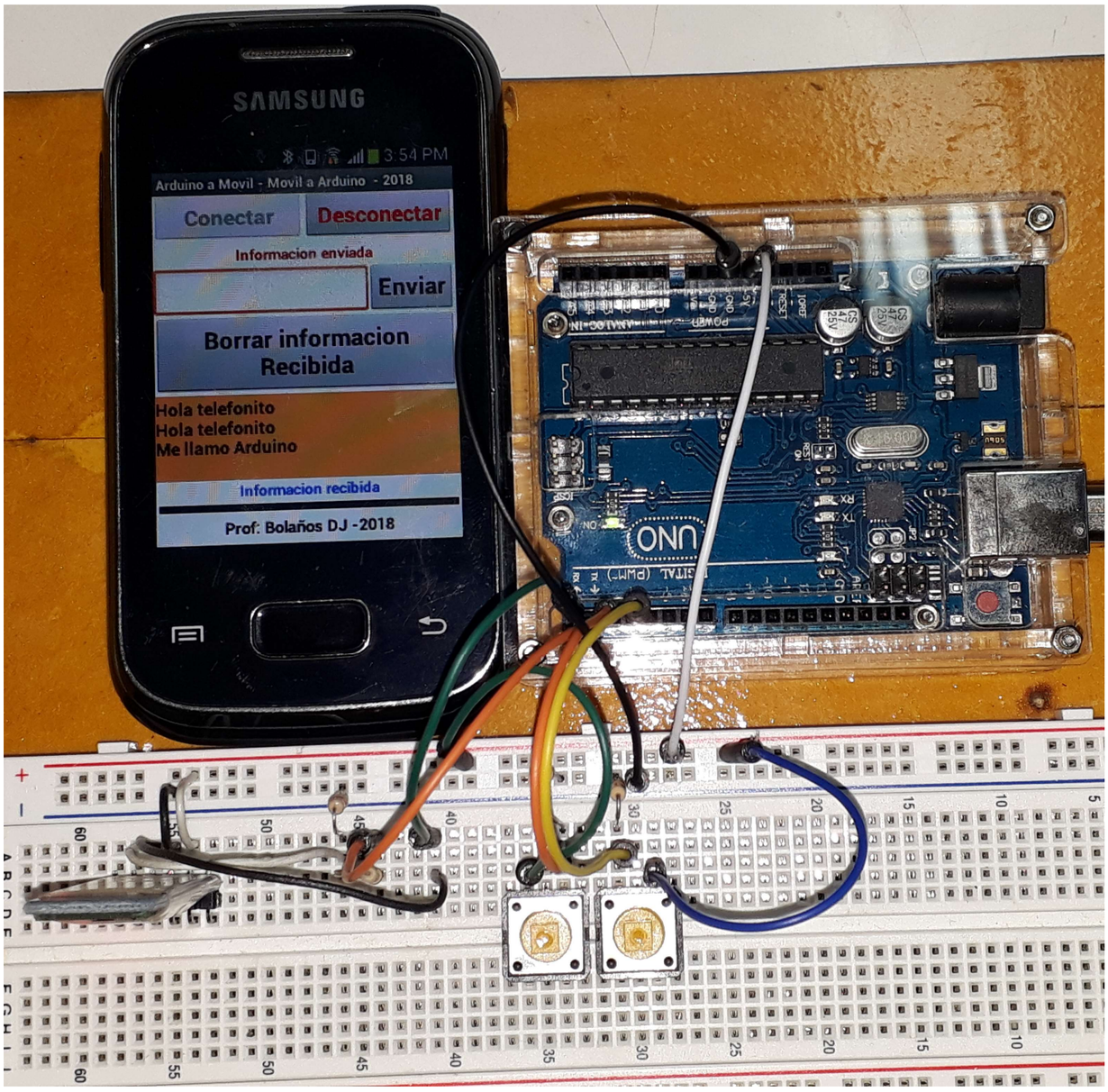
El módulo esta configurado como esclavo, tal como quedo de prácticas anteriores.

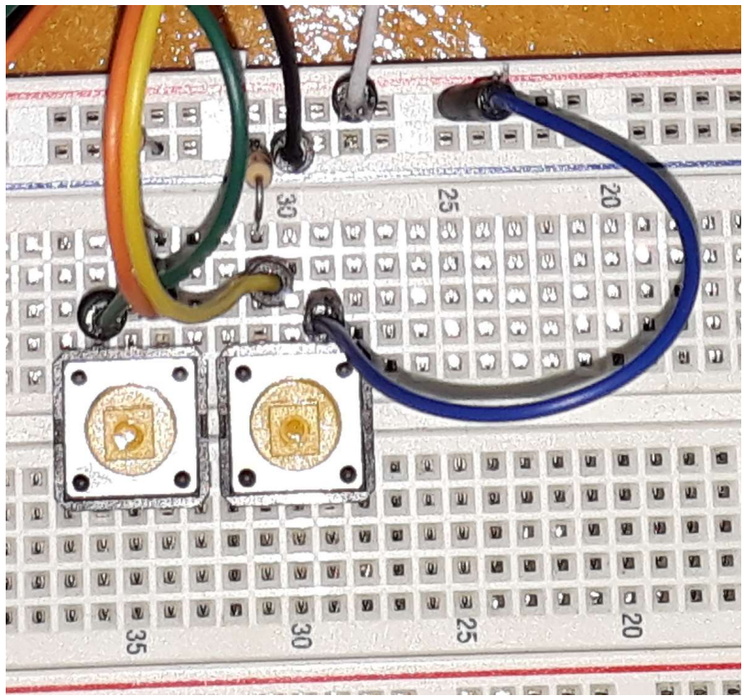
Atención: Usaremos divisor resistivo para el módulo Bluetooth (3,5v).

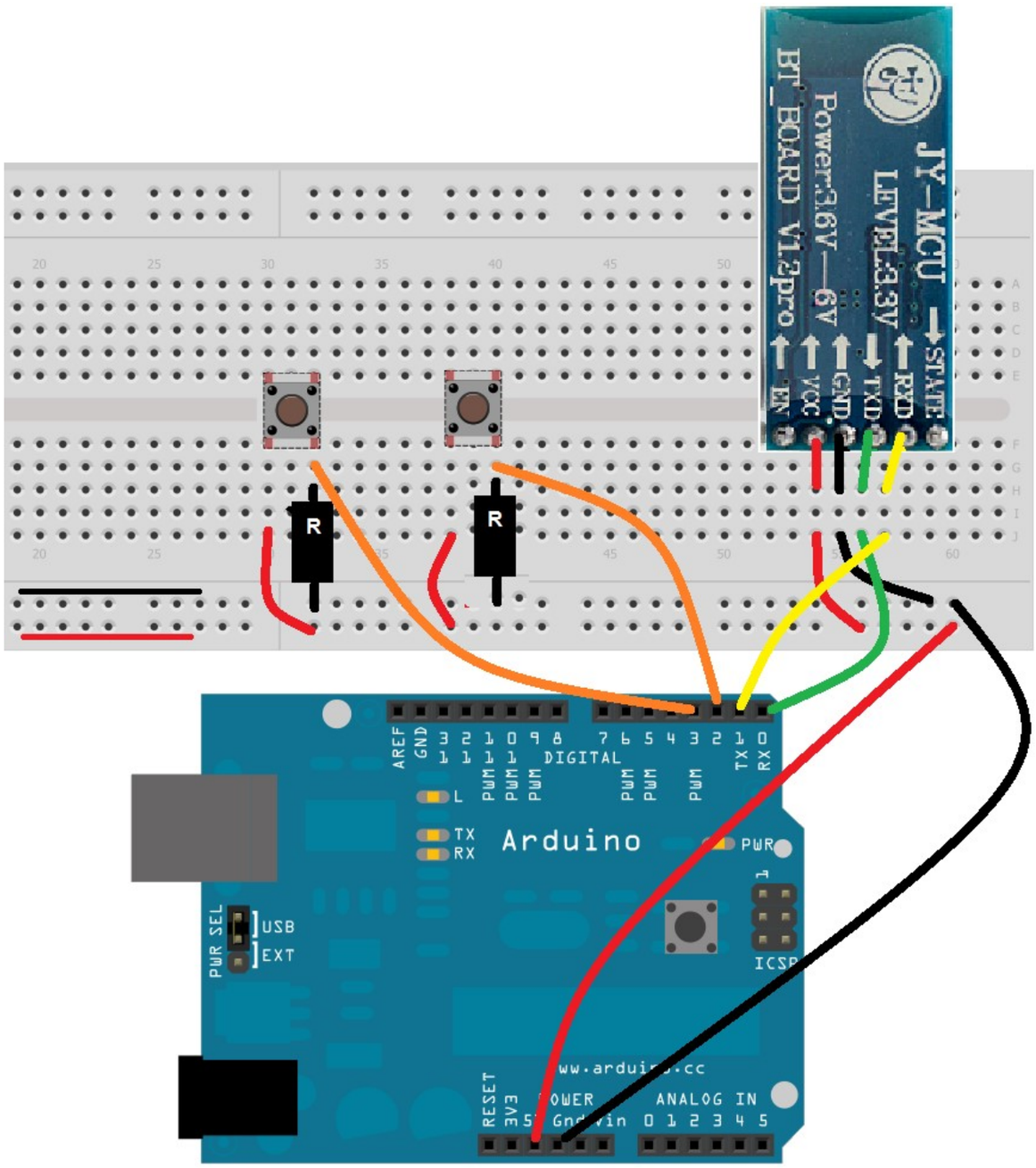


Debemos alimentar el módulo Bluetooth y además llevar tensión al pulsador como indica la figura superior. Utilizamos una resistencia de 10 K o 1K.

-Un pulsador irá al terminal 2 y el otro al terminal 3 del Arduino.


SAMSUNG
3:54 PM
Arduino a Movil - Movil a Arduino - 2018
Conectar
Desconectar
Informacion enviada
Enviar
Borrar informacion Recibida
Hola telefonito
Hola telefonito
Me llamo Arduino
Informacion recibida
Prof: Bolaños DJ -2018
UNO





ATENCION: Usamos divisor de tensión resistivo para el módulo Bluetooth. Ver al final.

El cable de un pulsador va al terminal 2 y el otro cable del otro pulsador al terminal 3 del Arduino (son los dos naranjas de la foto).

Código del Arduino.

IMPORTANTE: Es posible que si está conectado el módulo Bluetooth, el programa no se carge en el Arduino, así que cuando vayamos a cargar el programa al Arduino, quitamos los cables de la conexión de alimentación del módulo Bluetooth, cargamos el programa y luego volvemos a conectar los cables de alimentación del módulo Bluetooth.

- En este código cuando pulsemos uno de los dos botones, se enviará un texto al móvil.

Si pulsamos un Botón enviará: "Hola telefonito"

Si pulsamos el otro Botón enviará: "Me llamo Arduino"

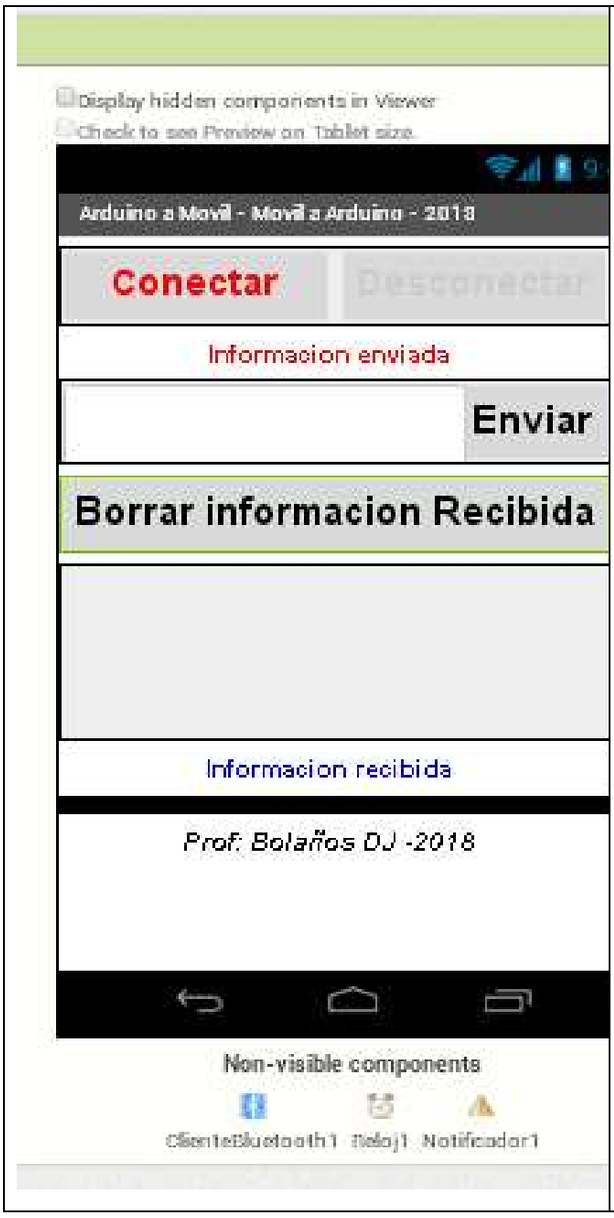
Estos mensaje son lo únicos que también se podrán ver en el monitor serie del IDE

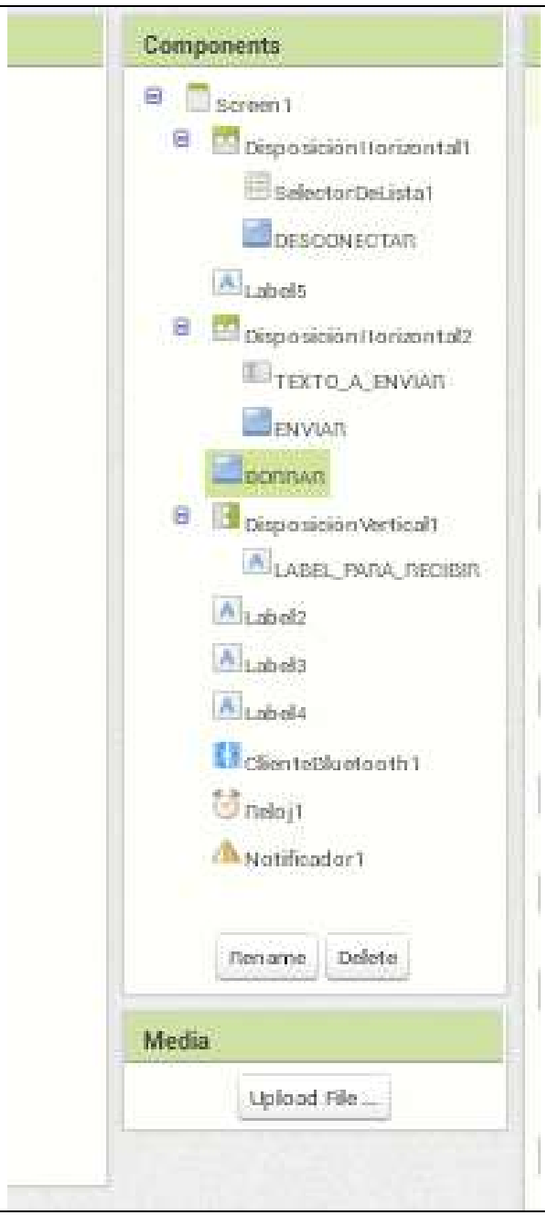
Además también recibe información, es decir, cuando escribamos un 0 o un 1 en el móvil y pulsemos el botón de enviar, se encenderá/apagará el LED 13, como vimos en el código anterior.

```
//Enviar -Recibir por Bluetooth
//Arduino a Movil - Movil a Arduino
//********************************************************
//Proyecto original de Juan Antonio Villalpando -juana1991@yahoo.com
//Adaptado y probado por Prof: Bolaños DJ - 2018
//********************************************************
char val;
int ledPin13 = 13;

//********** Arduino a Android
const int boton2 = 2;
const int boton3 = 3;

int buttonState2 = 0;
int buttonState3 = 0;

void setup()
{
Serial.begin(9600);
pinMode(boton2, INPUT);
pinMode(boton3, INPUT);
pinMode(ledPin13, OUTPUT);
}

void loop()
{

buttonState2 = digitalRead(boton2);
buttonState3 = digitalRead(boton3);

if (buttonState2 == HIGH)
{
Serial.println("Hola telefonito");
delay(200);
}
if (buttonState3 == HIGH)
{
Serial.println("Me llamo Arduino");
delay(200);
}

//************* Android a Arduino
/// LED 13
if( Serial.available() )
  {
  val = Serial.read();

  if( val == '0' )
    {
    digitalWrite(ledPin13, LOW);
    }

if( val == '1' )
{
digitalWrite(ledPin13, HIGH);
}
  }

}
```

En el código anterior se aconseja probar otros tiempos de delay para evitar el efecto rebote.

La APP para Android



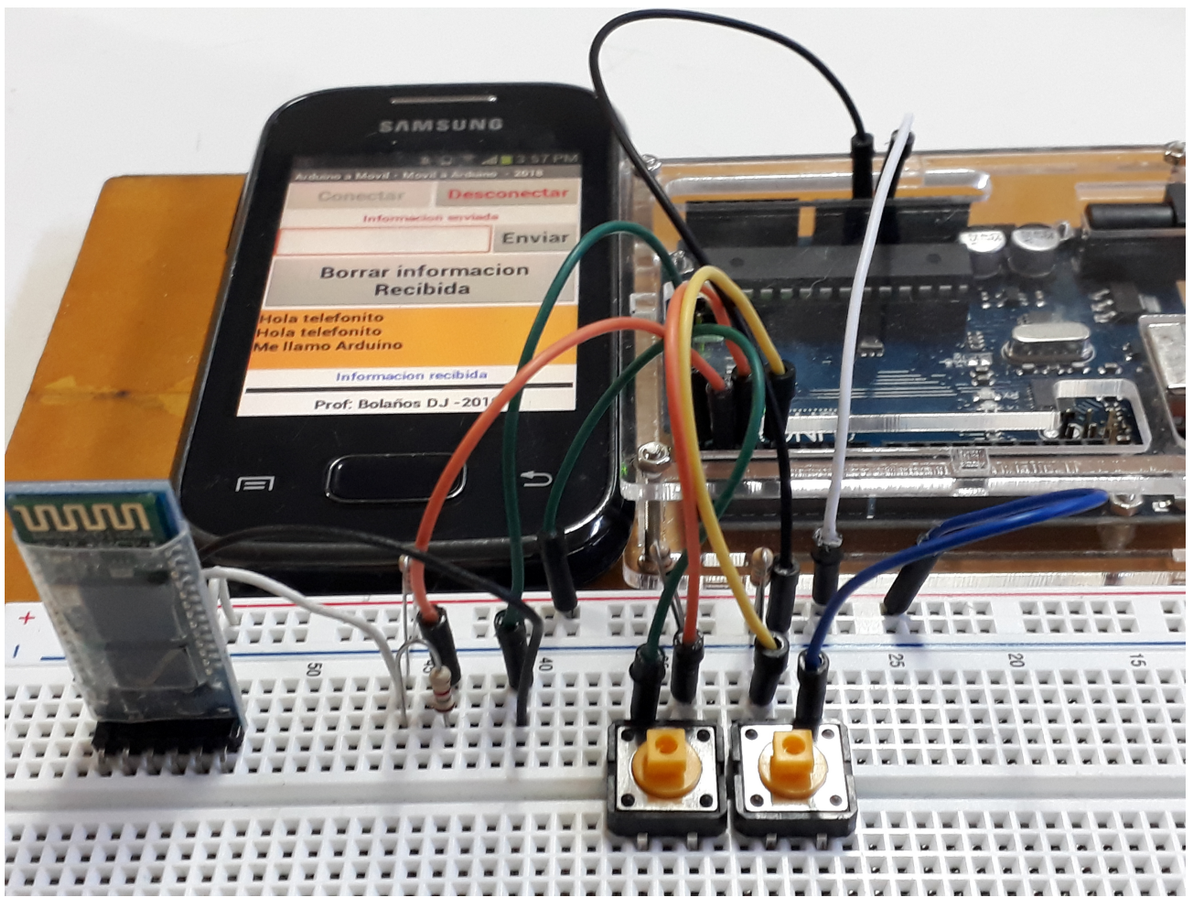


Enviando 0 o 1 y se apagará o encenderá el LED13 del Arduino.

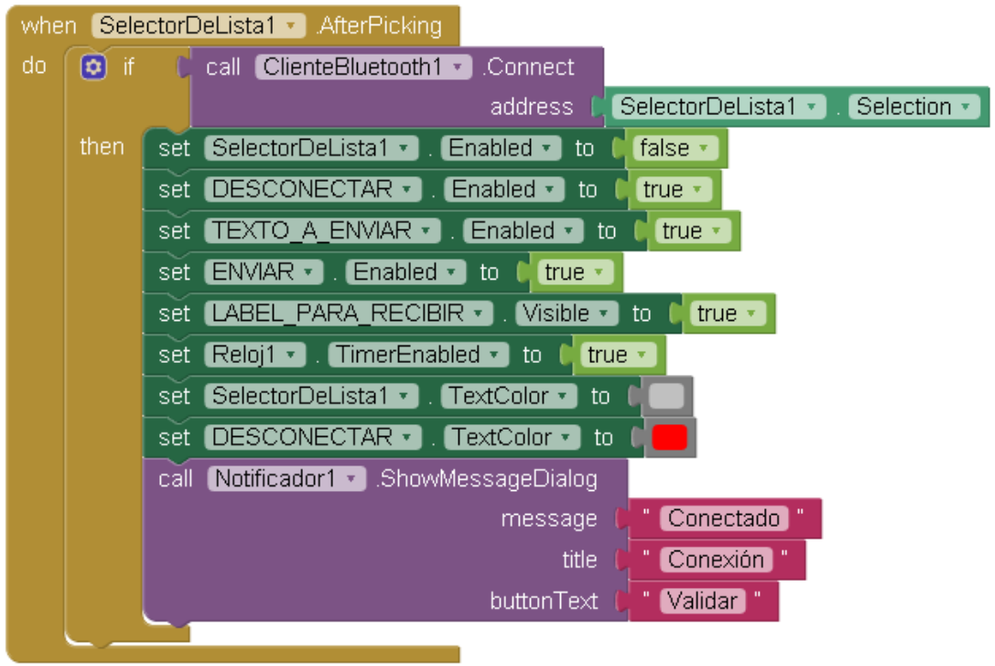
Si pulsamos un botón del Arduino, se escribirá un mensaje en el móvil.

DESCRIPCION DE LOS OBJETOS Y FUNCIONES DE LA APP

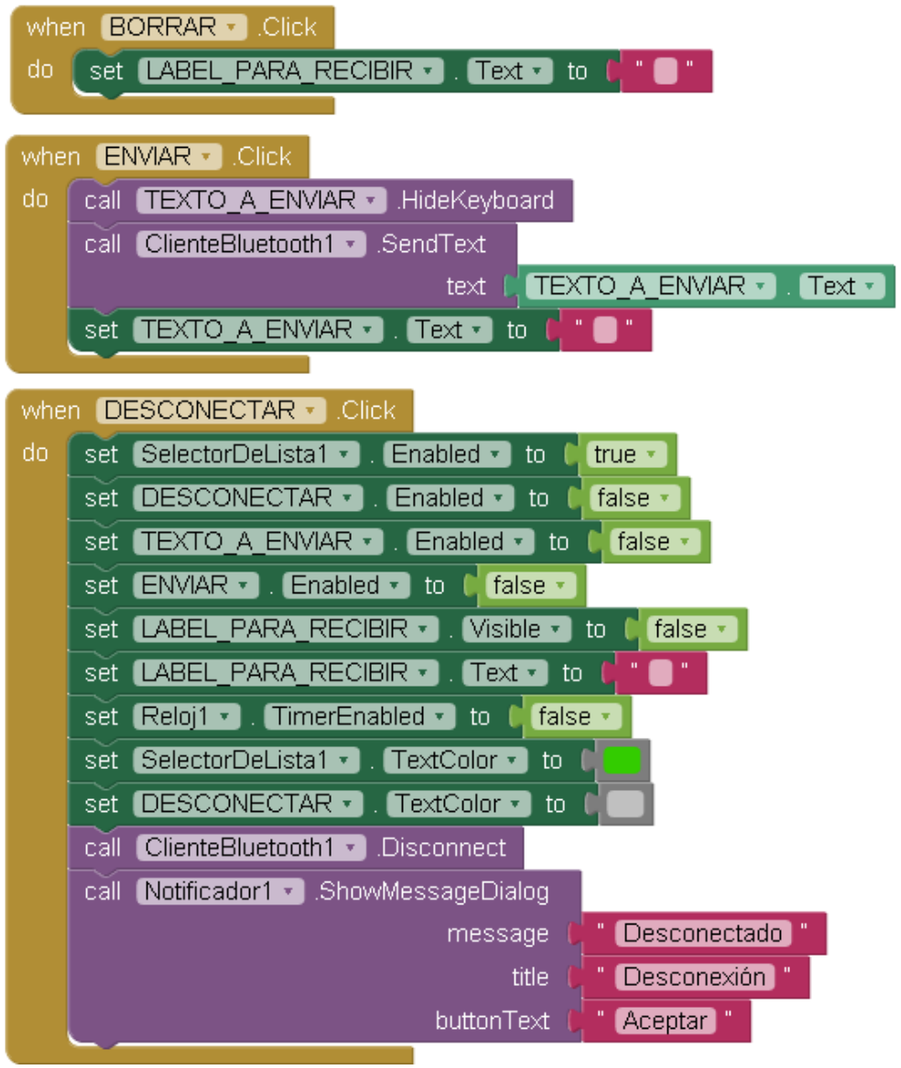
- El botón Conectar llama a un Selector De Lista, mediante éste se cargan los dispositivos Bluetooth encontrados. Elegimos nuestro modulo, se entra al proceso de conexión. El Reloj tiene Intervalo de Tiempo = 10 milisegundos
- Cuando pulsamos un botón en el Arduino, se envía un texto por Bluetooth.
- El Reloj Temporizador, está continuamente chequeando si hay Bytes disponibles recibidos, cada Intervalo deTiempo = 10, hace este chequeo.
- Así que el Reloj está continuamente en funcionamiento chequeando.
- En caso que se detecte Bytes disponibles recibidos, éstos se escriben en la Etiqueta, Es el texto enviado por la placa Arduino.
- El Botón Borrar información Recibida, borra la información del Label.
- El Botón Enviar: si escribimos en el casillero un 1 o un 0, se enviará al Arduino y se encenderá/apagará el LED13.
- El Botón Desconectar, realiza la desconexión Bluetooth.
- Al ensayarlo puede ocurrir el efecto rebote, es decir el Arduino está enviando el texto durante el tiempo de pulsado del botón, en cada pulsado envía varios mensajes correspondientes a ese botón, textos repetidos, ese es el resultado del efecto rebote.
- Para solucionarlo, podemos cambiar el delay en el código del Arduino o agregar un fin de texto a cada mensaje y cuando lo reciba el móvil que que la APP no admita otro hasta pasado un tiempo.
- Esta es una conexión asíncrona, es decir el Reloj y por consiguiente el móvil está continuamente chequeando si hay información, esto no es una buena método ya que consume recursos, sería mejor utilizar un proceso síncrono, esto es que el Arduino enviara una señal y el móvil la reconociera, sin estar constantemente chequeando. Pero esto implica más programación.




when SelectorDeLista1 .BeforePicking
do set SelectorDeLista1 . Elements to ClienteBluetooth1 . AddressesAndNames


when SelectorDeLista1 .AfterPicking
do if call ClienteBluetooth1 .Connect
address SelectorDeLista1 . Selection
then set SelectorDeLista1 . Enabled to false
set DESCONECTAR . Enabled to true
set TEXTO_A_ENVIAR . Enabled to true
set ENVIAR . Enabled to true
set LABEL_PARA_RECIBIR . Visible to true
set Reloj1 . TimerEnabled to true
set SelectorDeLista1 . TextColor to
set DESCONECTAR . TextColor to
call Notificador1 .ShowMessageDialog
message " Conectado "
title " Conexión "
buttonText " Validar "


when Reloj1 .Timer
do while test call ClienteBluetooth1 .BytesAvailableToReceive > 0
do set LABEL_PARA_RECIBIR . Text to join LABEL_PARA_RECIBIR . Text
call ClienteBluetooth1 .ReceiveText
numberOfBytes call ClienteBluetooth1 .BytesAvailableToReceive



FUENTES:
http://kio4.com/appinventor/291B_extension_notificacion.htm

http://kio4.com/appinventor/9bluetootharduino.htm

ANEXO:

En primer lugar tenemos la salida del módulo (TX, pin 1) en ella tendremos un nivel de 0 para el nivel lógico 0 y aproximadamente 3.3V para el nivel 1. Como según las características del ATMega328P el Vih es 0.6Vcc, y nuestro Vcc es de 5V, tendremos que Vih es aproximadamente de 3V y por tanto es perfectamente válido conectar el TX del HC-05 con el RX del Arduino.

Algo totalmente diferente es la salida del Arduino hacia la entrada (RX, pin 2) del módulo. En el datasheet del HC-05 no he visto que indique que las entradas sean tolerantes a 5V, y por otro lado en el datasheet del ATMega328P indica que Voh es, como mínimo, de 4.2V, por lo que, por precaución, deberíamos adaptar los niveles. Hay muchos métodos para hacer la adaptación de niveles, aunque, para mí, en este caso la solución más fácil es realizar un divisor resistivo, de forma que solo le entreguemos una parte de la tensión que esté dentro de sus márgenes de tolerancia al módulo bluetooth.

Para ampliar el tema vea el apunte: **ConfigHC05.pdf**