Control de un LED desde App Inventor via BT

Controlando Arduino desde dispositivos móviles

Jose Luis Núñez
José Pujol

CEP Sevilla 2019

ÍNDICE

FINALIDAD DEL SISTEMA

Controlar Arduino desde el teléfono móvil mediante bluetooth, encendiendo y apagando un LED a través de un botón que crearemos en la aplicación.

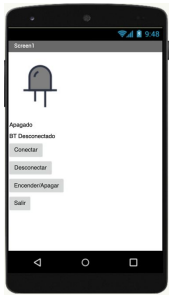
ENTRADAS

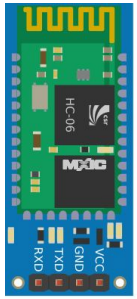
Dispositivo móvil
Módulo Bluetooth

ARDUINO

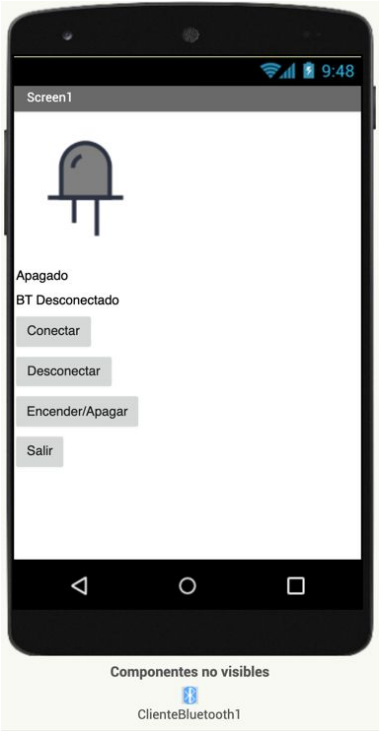
SALIDAS

LED (D)





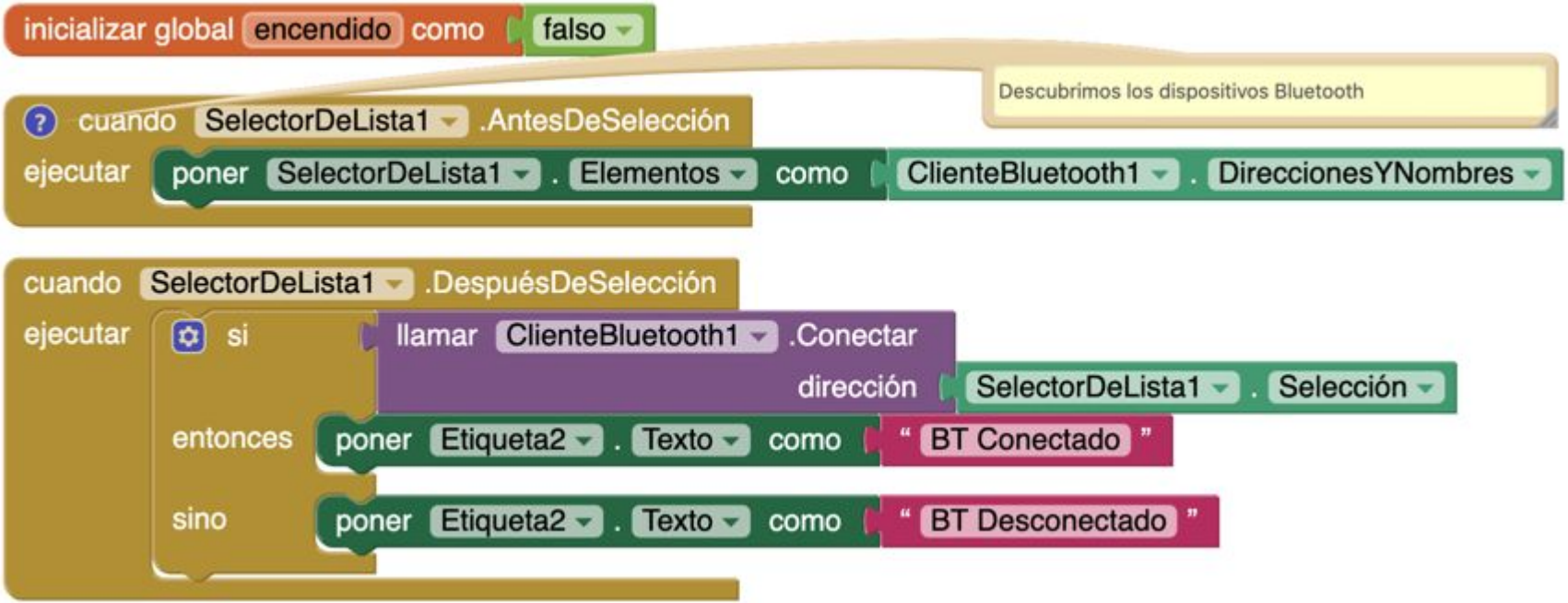
APLICACIÓN MIT APP INVENTOR



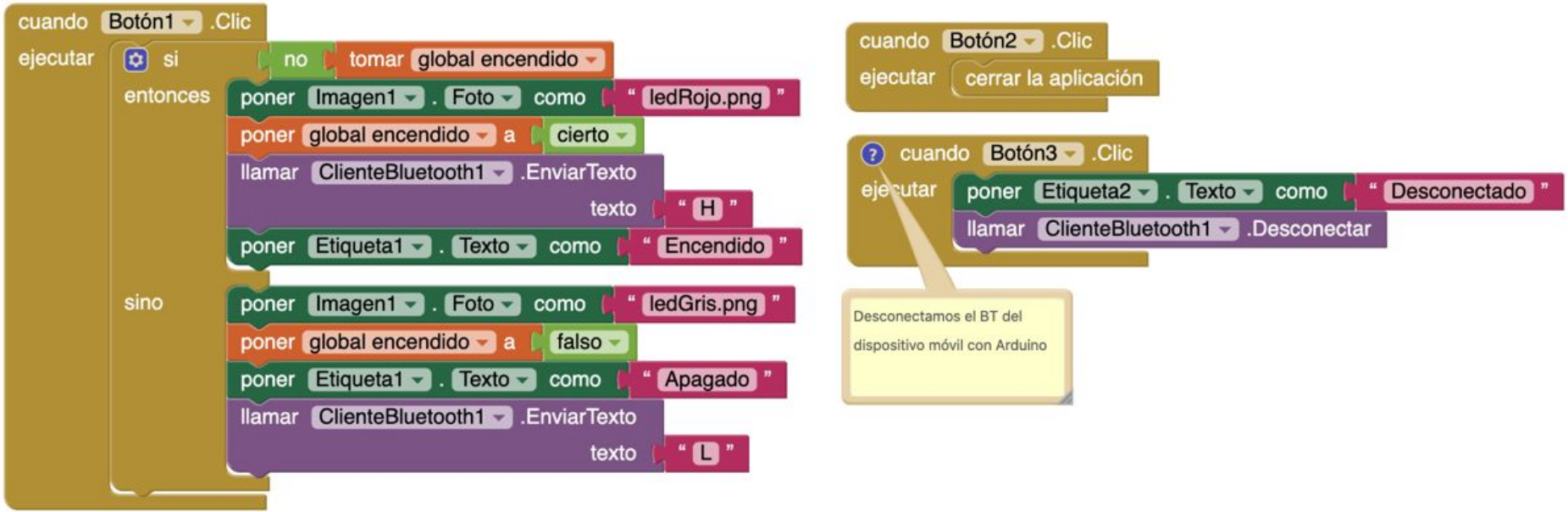
Añadimos...

- Componente ClienteBluetooth
- Etiqueta para mostrar cuando BT estará conectado/desconectado
- Selector de lista para conectar
- Botón para desconectar

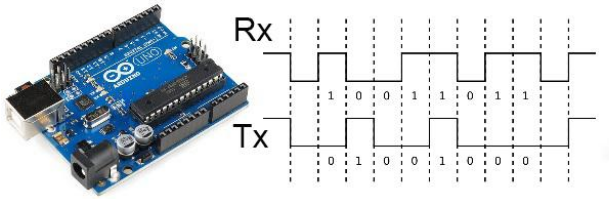
Configuración del selector bluetooth



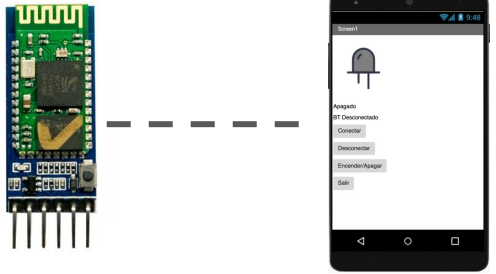
Envío de información, cierre y desconexión



MÓDULO BT



Comunicación Serie



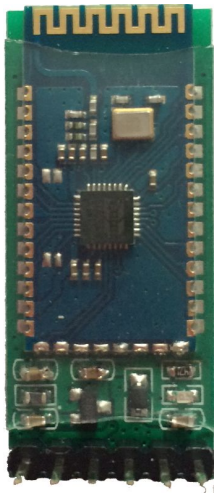
Esclavo

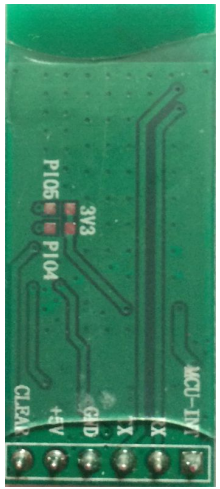
Maestro

- Usa los pines Rx y Tx para comunicarse.
- **Librería Software Serial** si queremos usar otros pines para la comunicación y así no ocupar 0 y 1.
- **Configuración** dispositivo BT mediante comandos AT.

Modulo BT

HC06





CLEAR
5V
GND
Tx
RX
MCU-INT

Configurar nombre y clave BT

- Se configura el BT mediante comandos AT
- Por defecto: nombre Arduino, clave 1234, velocidad 9600 baudios
- Al tener varios BT en una sala debemos cambiarles el nombre
- Para programarlo: cargar el siguiente código en Arduino, sin conectar el BT
- Una vez conectado el BT reiniciar Arduino. Cuando el LED L de Arduino empiece a parpadear la programación del BT ha terminado

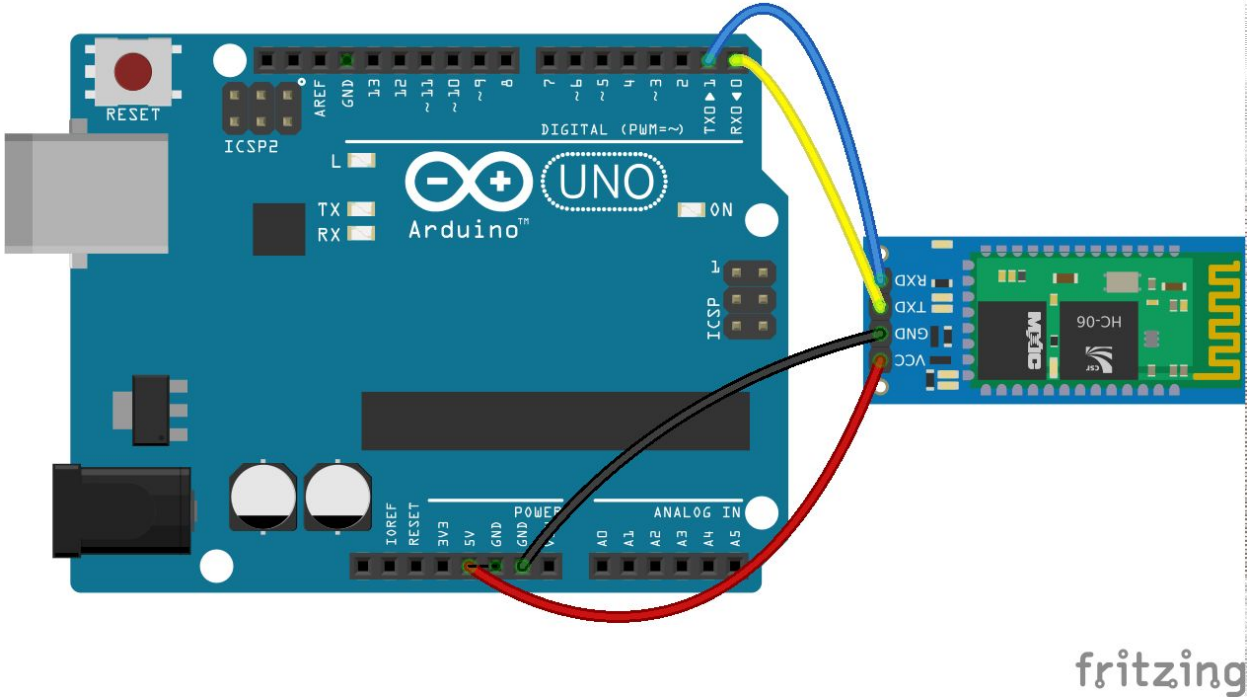
Fuente Giltesa

Software Configurar nombre y clave BT

```
// Opciones de configuración:
  char ssid[10]    = "Arduino1"; // Nombre para el modulo Bluetooth.
  char baudios     = '4';        // 1=>1200 baudios, 2=>2400, 3=>4800, 4=>9600 (por defecto), 5=>19200
  char password[10] = "1234";    // Contraseña para el emparejamiento del modulo.


void setup()
{
  Serial.begin(9600);

  // Tiempo de espera:
    pinMode(13,OUTPUT);
    digitalWrite(13,HIGH);
    delay(10000);
    digitalWrite(13,LOW);

  // Ahora se procede a la configuración del modulo:

    // Se inicia la configuración:
      Serial.print("AT"); delay(1000);

    // Se ajusta el nombre del Bluetooth:
      Serial.print("AT+NAME"); Serial.print(ssid); delay(1000);

    // Se ajustan los baudios:
      Serial.print("AT+BAUD"); Serial.print(baudios); delay(1000);

    // Se ajusta la contraseña:
      Serial.print("AT+PIN"); Serial.print(password); delay(1000);
}
```

Código

Hardware Configurar nombre y clave BT

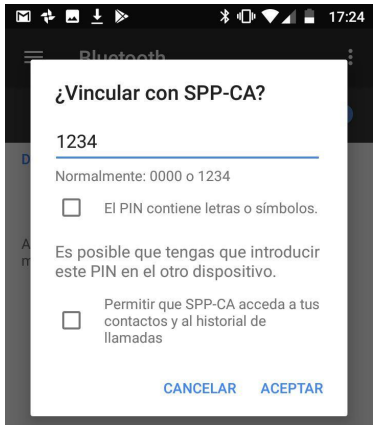


Nota:
No podemos tener el Bluetooth conectado a los pines Rx, Tx en el momento que cargamos el programa

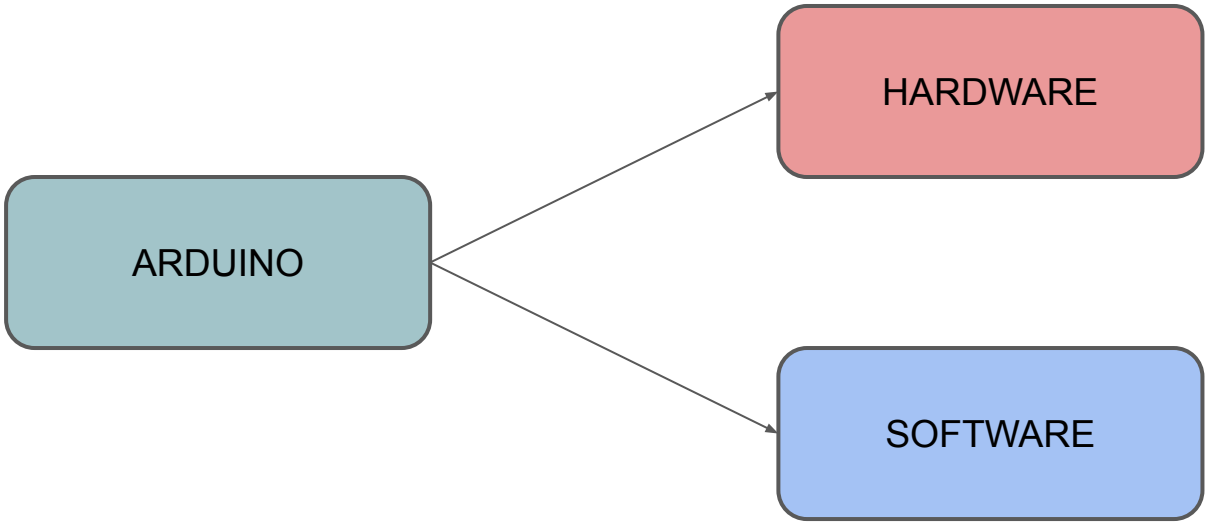
Vinculación BT con el teléfono

TELÉFONO MÓVIL

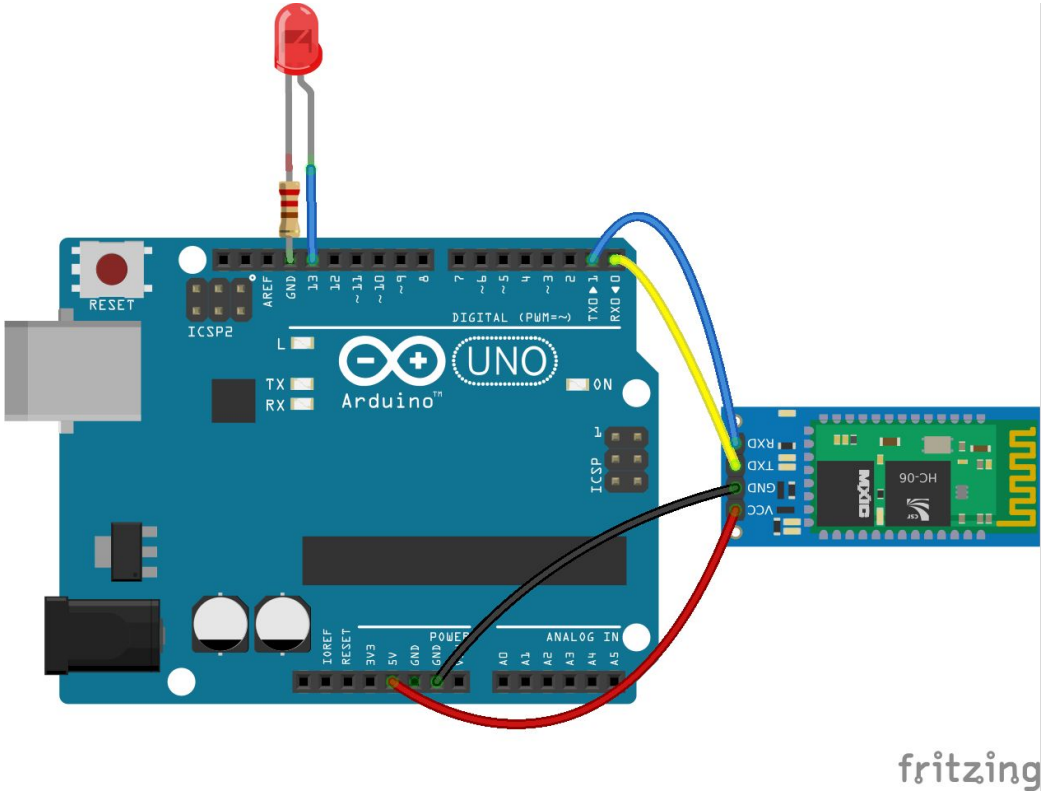
6. Activar bluetooth del teléfono móvil y buscar dispositivos.
7. Vincular con NOMBRE, e introducir la clave: 1234.



ARDUINO



Hardware

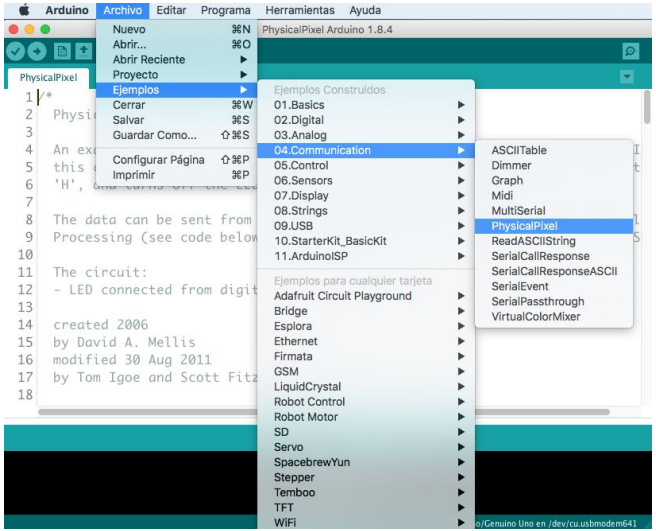


HC06	ARDUINO
GND	GND
5V	5V
Tx(entr)	Rx(sal)
Rx(sal)	Tx(ent)

Software

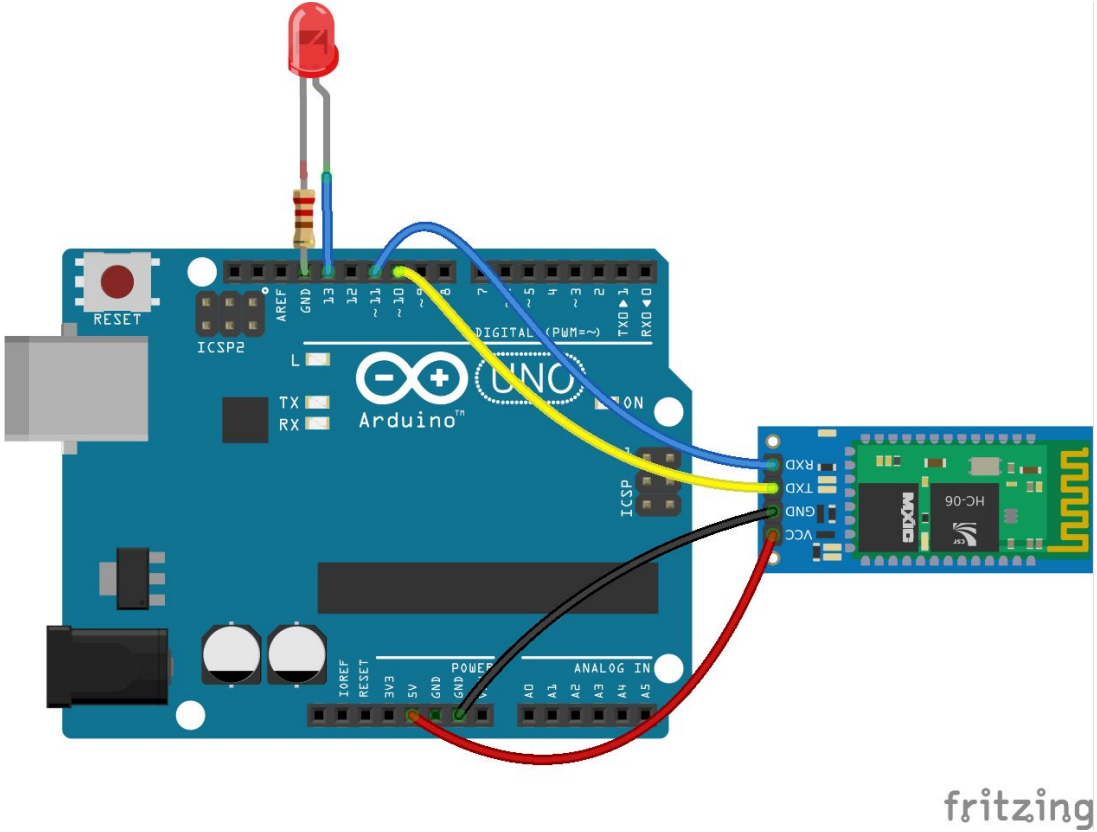
ARDUINO

- Cargar programa Physical Pixel en Arduino.
- Ejemplos⟶ Communication⟶ PhysicalPixel. Y subir a la placa.



Nota:
No podemos tener el Bluetooth conectado a los pines Rx, Tx en el momento que cargamos el programa

Hardware Software Serial



HC06	ARDUINO
GND	GND
5V	5V
Tx(entr)	10
Rx(sal)	11

Software

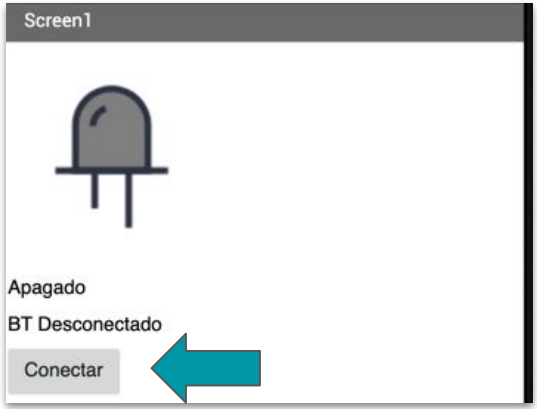
```
#include <SoftwareSerial.h>
SoftwareSerial I2CBT(10, 11);
// El TX del módulo BT va al pin 10 del Arduino
// El RX del módulo BT va al pin 11 del Arduino

const int ledPin = 13; // pin al que el led esta conectado
int incomingByte;      // variable para leer los bytes de entrada

void setup() {
  // inicializamos el led como pin digital salida
  pinMode(ledPin, OUTPUT);
  // inicializamos la comunicacion serie BT
  I2CBT.begin(9600);
}

void loop() {
  // comprobamos si hay datos de entrada
  if (I2CBT.available() > 0) {
    // lectura del byte mas antiguo del buffer serial
    incomingByte = I2CBT.read();
    // si el byte es H (ASCII 72) enciende el LED
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // si el byte es L (ASCII 76) apaga el LED
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

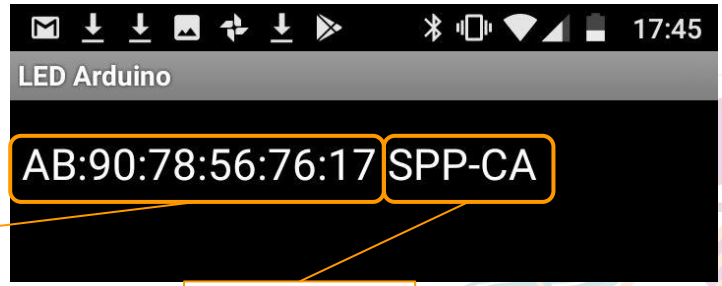
Código

FUNCIONAMIENTO

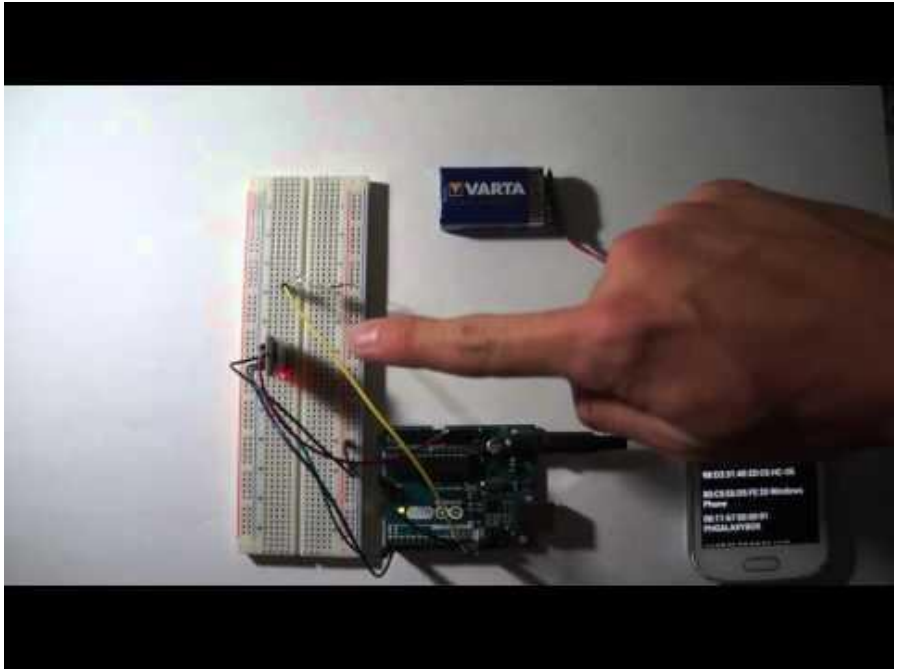
TELÉFONO

1. Abrir APP.
2. Presionar Conectar
3. De la lista de dispositivos seleccionar Nombre BT, nos debe aparecer el mensaje BlueTooth Conectado“
4. Encender y apagar el LED con pulsador de la APP.





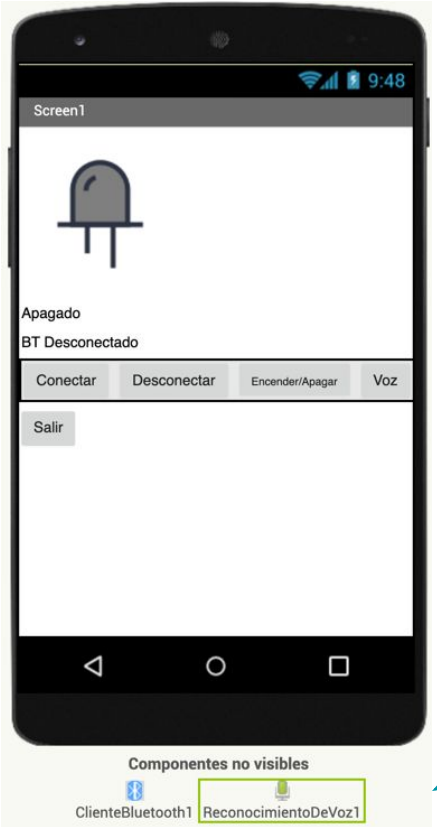
Funcionamiento



Propuesta de actividad

- Añadir reconocimiento de voz proporcionado por MIT App Inventor para encender/apagar el led utilizando un botón en nuestra app.

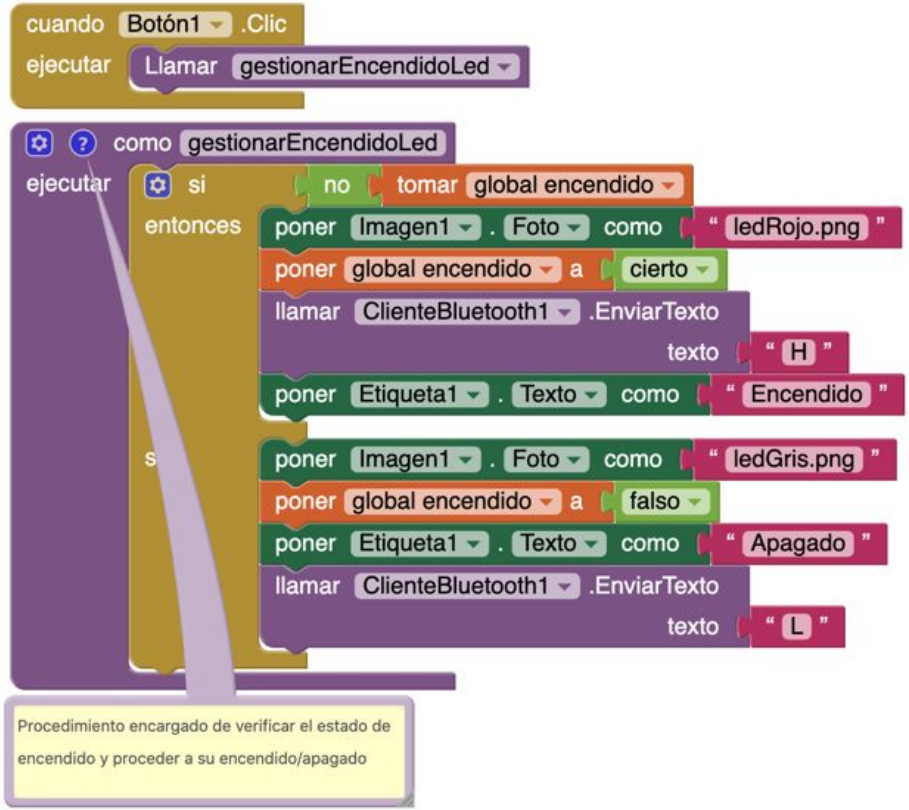
Encendido del LED mediante Voz



Añadimos una disposición horizontal que permita alinear los botones..¿Cómo podríamos separarlos?

En los componentes dedicados a *Medios* encontraremos ReconocimientoDeVoz que nos permitirá hablar y controlar el led

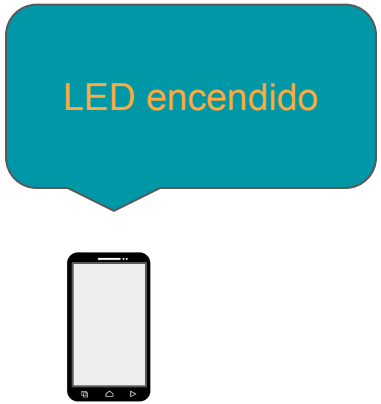
Bloques que permiten la mejora (1/2)



Bloques que permiten la mejora (2/2)

Desafío

- Finalmente, añade el componente TextoAVoz para que una vez encendido/apagado el LED nuestra app nos lo comunique mediante una frase, por ejemplo, *“LED encendido”*.



LICENCIA

Las diapositivas son obra de Jose Pujol y Jose Luis Núñez creadas para el curso “Controlando Arduino desde el teléfono móvil” para el CEP de Sevilla y han sido creadas a partir del material elaborado para el curso “Tech Project: Arduino en el aula” que fue realizado por Jose Antonio Vacas y Jose Pujol en colaboración con Avante s.l.