

### 3.2.4. Encender Led con AppInventor

En esta actividad vamos a encender/apagar un LED de manera remota utilizando nuestra Tablet o Móvil Android. Al igual que en las siguientes actividades, deberemos escribir un código para programar nuestro Arduino (mediante el Arduino IDE) y otro código para programar nuestro dispositivo Android (mediante APPInventor). Lo ideal es que hagas los códigos utilizando las instrucciones que encontrarás a continuación, pero también puedes bajártelos de nuestra web.

#### 3.2.4.1. Programa para Android (código de Bloques en AppInventor)

##### - **Diseño**

Abrimos AppInventor y le damos a crear nuevo proyecto. La aplicación que crearemos tendrá el mismo aspecto que podemos ver en la Figura 3.2.4-1. Los componentes que en ella aparecen son:

- Tres botones
  - Un ListPicker, que servirá para activar el cliente Bluetooth y conectarnos con un dispositivo de la lista de dispositivos disponibles.
  - El botón 2 mandará un mensaje al robot para encender el led
  - El botón 3 mandará un mensaje al robot para apagar el led
- Un cliente bluetooth

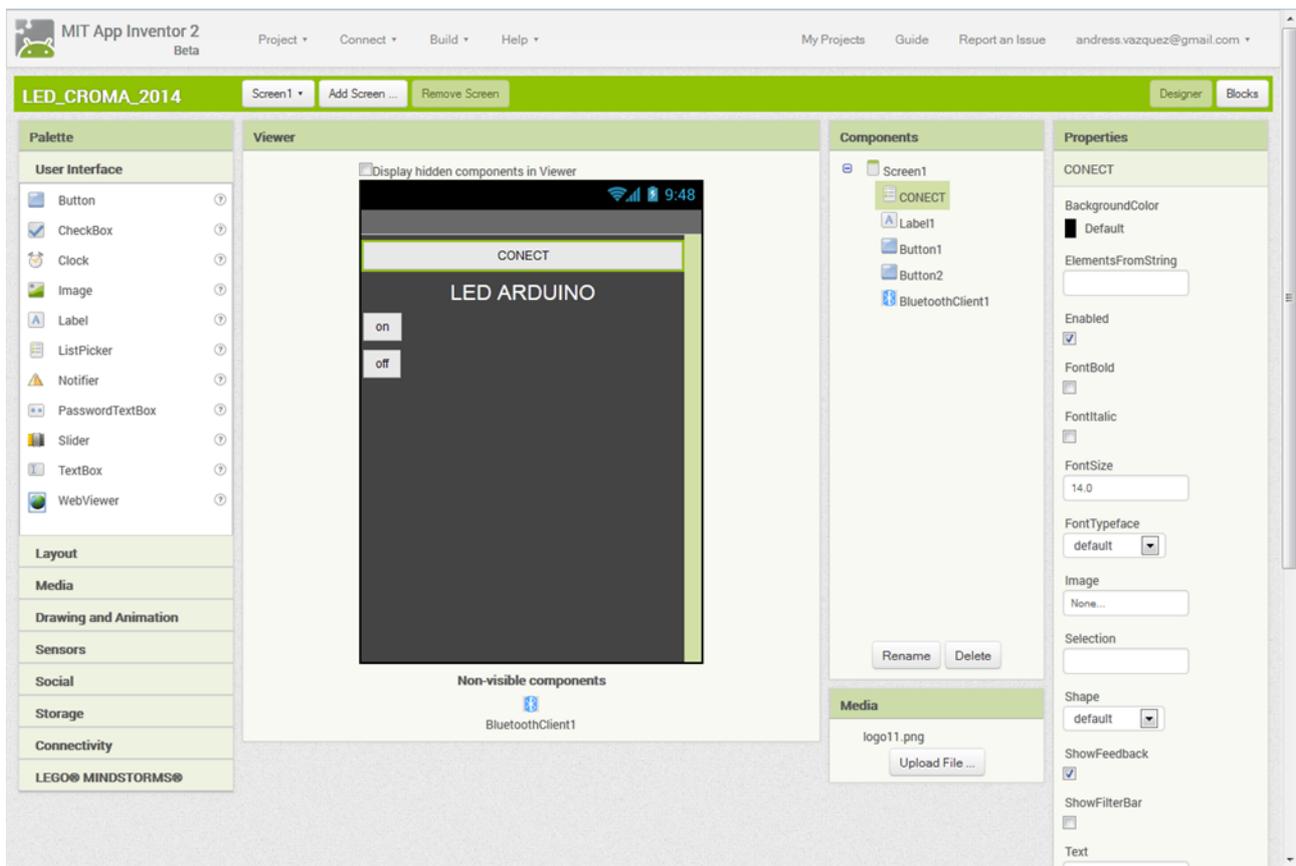


Figura 3.2.4-1 Diseño final de la aplicación de la actividad en AppInventor

## - Programación

La programación en bloques de nuestra aplicación será la que aparece en la Figura 3.2.4-2.

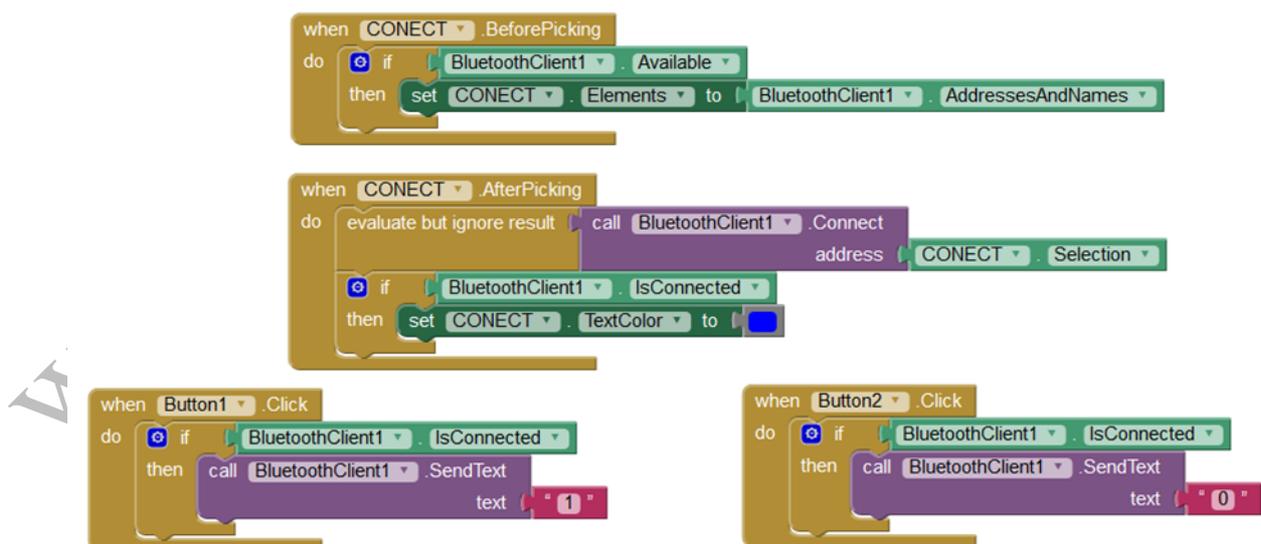


Figura 3.2.4-2 Programa en bloques en AppInventor de la actividad

El funcionamiento de la aplicación, desglosado por bloques, aparece a continuación:

El bloque “when CONECT BeforePicking” incluye todo lo necesario para que, cuando el usuario pulse el primer botón, se muestre por pantalla una lista con todos los dispositivos Bluetooth visibles desde nuestro móvil o tablet. Se debe tener en cuenta que dicho botón es el que en la figura Figura 3.2.4-1 aparece con el texto “Conect”.

Cuando el usuario selecciona un dispositivo de la lista, se activa el bloque “when CONECT afterpicking”. Dentro de este bloque está todo lo necesario para conectarse al dispositivo seleccionado.

El bloque “when Button1 Click” envía por el Bluetooth el caracter 1 (uno). El código de arduino deberá examinar el carácter recibido y si es “1”, se deberá encender el led.

El bloque “when Button1 Click” envía por el Bluetooth el caracter 0 (cero). El código de arduino deberá examinar el carácter recibido y si es “0”, se deberá encender el led.

### 3.2.4.2. Programa en Arduino (con el IDE de Arduino)

El programa en Arduino que debemos crear, abrirá una comunicación serie y permanecerá a la espera de cualquier mensaje que se mande desde la aplicación Android creada en la sección 3.2.4.1. El programa de Arduino estudiará el mensaje recibido y actuará en consecuencia, si recibe un “1” enciende el led, si recibe un “0” lo apaga. El código que permite hacer todo esto es el que aparece a continuación. Ten en cuenta que este programa funcionará en cualquier placa ZUM-BT.

```
/*
conexion con appinventor. Para la zumbt con Bluetooth integrado
*/
#include <SoftwareSerial.h>
char comando;
int led = 13;

void setup()
{
  // Abrimos el Puerto serie
  Serial.begin(19200);
  pinMode(led, OUTPUT);
}
void loop()
{
  if (Serial.available())
  {
    comando=Serial.read();
    //Según el dato recibido
    switch (comando){
      case '1':
        digitalWrite(led, HIGH);
        break;
      case '0':
        digitalWrite(led, LOW);
        break;
    }
  }
}
```

Si utilizas un Arduino UNO o compatible y un módulo HCO5 como el de la Figura 3.2.2-1 o el de la Figura 3.2.2-2, deberás cambiar la configuración del puerto serie, modificando la velocidad a 9600 baud o 38400 respectivamente. También, deberás indicar los pines donde has conectado el RX y el TX del módulo. El código queda así:

```
#include <SoftwareSerial.h>
char comando;
int led = 13;
SoftwareSerial mySerial(4, 5); //en este ejemplo hemos conectado al pin 4 el tx
y el rx al pin 5.

void setup()
{
  pinMode(led, OUTPUT);
  mySerial.begin(9600); //Si es el modulo de BQ pon aqui 38400
}
if (mySerial.available())
{
  comando = mySerial.read();
  //Según el dato recibido
  switch (comando) {
    case '1':
      digitalWrite(led, HIGH);
      break;
    case '0':
      digitalWrite(led, LOW);
      break;
  }
}
```

**RECUERDA** estos cambios para configurar correctamente el puerto serie en las siguientes actividades en función del módulo bluetooth que estés utilizando.

Como vemos, cuando hay datos en el puerto serie los lee y los guarda en una variable que hemos llamado “comando”. Si “comando” vale 1 entonces se enciende el LED, si “comando” es 0 (cero) lo apaga.

Para probar esta actividad sólo necesitas subir a tu móvil o Tablet el programa en AppInventor, tal y como explicamos en la Sección 3.1.2, y subir a tu tarjeta Arduino UNO o compatible el programa cliente. Recuerda que antes tienes que tener el bluetooth configurado (como vimos en la sección 3.2.1) y vinculado con tu dispositivo Android.