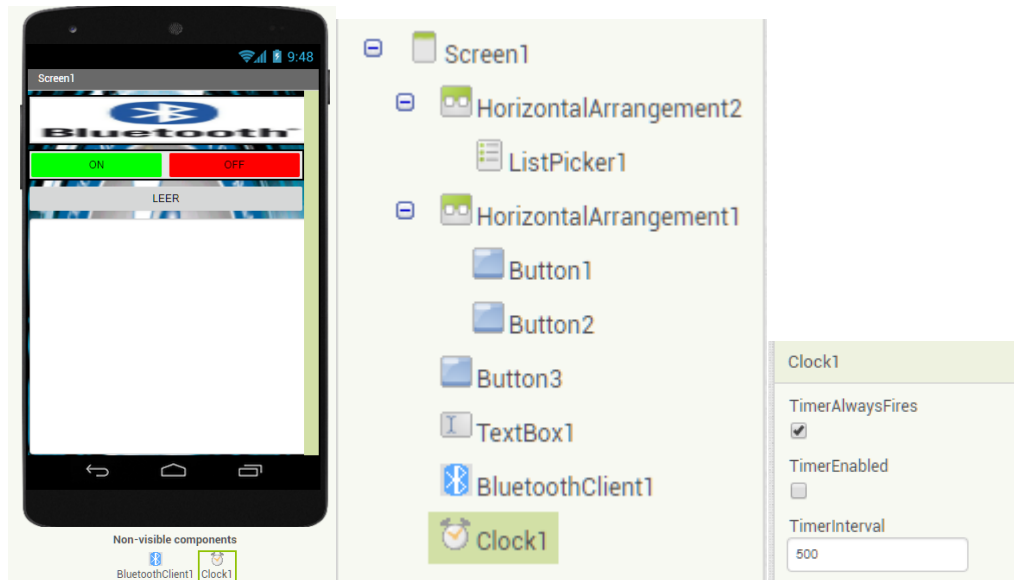


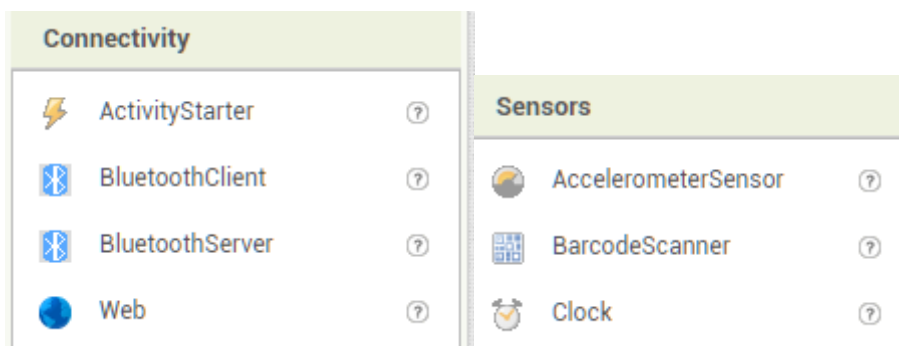
## EJEMPLO 2 – Conexión Bluetooth, envío y recepción de información

### DESIGNER

Esta sería la interface, como reto esta perfecta, sobre todo los Arreglos Horizontales que permiten colocar los botones ON y OFF en el mismo grupo Horizontal.



Además de los objetos no visibles, BluetoothClient y Clock. Que permitirán realizar las conexiones con otros dispositivos y llevar un control del tiempo, respectivamente.



## BLOCKS

Luego de diseñar la interface gráfica, vamos a darle acciones a los objetos que permiten que se ejecuten las acciones.

- 1- Cuando se selecciona el botón ListPicker se ejecutan dos eventos, BeforePicking(AntesDe) y AfterPicking(DespuesDe).  
 En el primero se cargan sus elementos después de verificar que BluetoothClient1 está disponible, esto solo ocurre si el dispositivo móvil tiene activa la conectividad por bluetooth.  
 En el segundo se conecta el dispositivo con el dispositivo seleccionado en el primer paso.

```

when ListPicker1 .BeforePicking
do
  if BluetoothClient1 . Available
  then
    set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
    
```

```

when ListPicker1 .AfterPicking
do
  if BluetoothClient1 . Available
  then
    set ListPicker1 . Selection to call BluetoothClient1 .Connect
    address ListPicker1 . Selection
    
```

- 2- Una vez hecha la conexión, vamos a enviar datos desde el dispositivo móvil al otro dispositivo

```

when Button1 .Click
do
  if BluetoothClient1 . Available
  then
    call BluetoothClient1 .Send1ByteNumber
    number 97
    
```

```

when Button2 .Click
do
  if BluetoothClient1 . Available
  then
    call BluetoothClient1 .Send1ByteNumber
    number 112
    
```

Cuando se haga click en *Button1*, se usa la condición *if* para verificar que el bluetooth esté disponible, si esto es verdad, se llama la función *SendText* para enviar el texto "H". De la misma forma se envía el texto "L" cuando se hace click en el *Button2*.

- 3- También podemos recibir información desde el dispositivo. Para activar o desactivar la lectura necesitamos una variable que va a cambiar de 0 a 1 y de 1 a 0. Inicialmente la variable contendrá un 0.

```
initialize global leyendo to 0
```

- 4- Cuando se haga clic en el *Button3*, se usará una condicional para saber si el contenido de la variable *leyendo* es 0 o 1, si es 0, se activara el reloj, que previamente se ha configurado en el modo *Designer* con un intervalo de 500 mili segundos.

Ademas se cambiará el texto del botón a "DEJAR DE LEER" y se colocara de color rojo su propiedad *Background*, por último la variable se cambiará el valor a 1. Si el valor no es 0, se deshabilitara el *Clock1*, y se cambiará la apariencia del *Botton3*.

```
when Button3 . Click
do
  if
  then
    set Clock1 . TimerEnabled to true
    set Button3 . Text to "DEJAR DE LEER"
    set Button3 . BackgroundColor to red
    set global leyendo to 1
  else
    set Clock1 . TimerEnabled to false
    set Button3 . Text to "LEER"
    set Button3 . BackgroundColor to green
    set global leyendo to 0
```

- 5- El ultimo paso en la lectura es usar el evento *Clock1.Timer*, que ocurre cada intervalo de tiempo (para nuestro caso, cada 500 milisegundos), si el Bluetooth esta conectado, se ejecuta nuevamente otra condicional que pregunta si el *TextBox1* tiene "", osea si esta vacio, si es así usa las propiedades *BackgroundColor* y *Text* para cambiar la apariencia a verde y el texto "ESPERANDO INFORMACIÓN"

Luego hace llama la función *BluetoothClient1.BytesAvailableToReceive* para saber si hay datos en el buffer, en caso de haberlos, usa *join* para ir

```
when Clock1 . Timer
do
  if BluetoothClient1 . IsConnected
  then
    if
    then
      set TextBox1 . BackgroundColor to green
      set TextBox1 . Text to "ESPERANDO INFORMACIÓN ..."
    while test
    do
      call BluetoothClient1 . BytesAvailableToReceive > 0
      set TextBox1 . Text to join TextBox1 . Text
      call BluetoothClient1 . ReceiveText numberOfBytes
      call BluetoothClient1 . BytesAvailableToReceive
```

concatenando lo que se vaya recibiendo por el buffer y reescribiendo el dato en el *TextBox1.Text*.

**RETO**

Diseñar y construir un circuito eléctrico (recomendado Arduino + HC 06 + sensores y actuadores)

- 1- Que responda a las señales de los botones ON OFF desde el programa
- 2- Que cambie el color de la pantalla si recibe una señal desde el circuito.

Diseño:

<b>Explicación de funcionamiento</b>	<b>Boceto Físico – eléctrico</b>