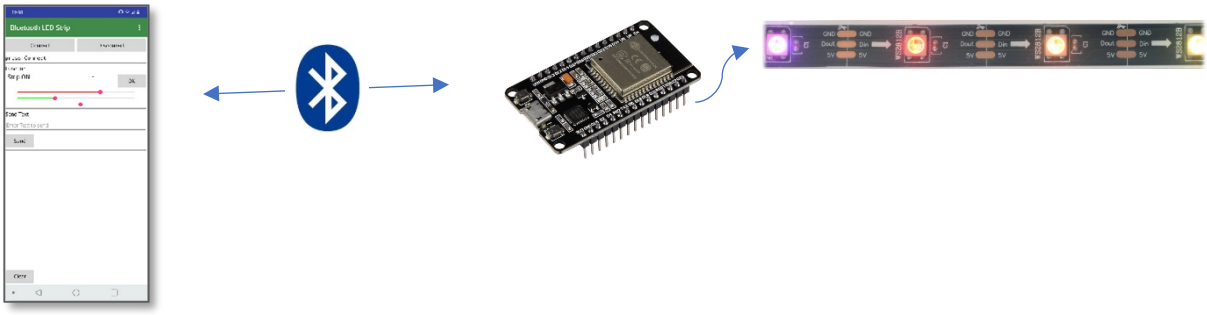
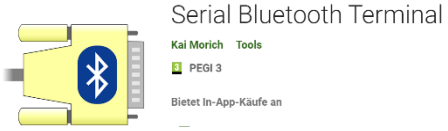
Programming a mobile app for the LED-Strip Module

The LED Strip Module is based on a NodeMCU-ESP32 microcontroller board with a simple firmware which can be controlled by a mobile app sending commands via a serial Bluetooth connection.



You can test the LED Strip Module with a Bluetooth Terminal App like the Serial Bluetooth Terminal in the Google Play App Store
The Bluetooth device name is ErasmusLEDs*XY*

(XY being some number)



https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal

The commands must be sent in the following format:

```
command:parameter;
```

for example:

```
light:on;
```

to switch on all the leds of the strip showing the default color. If there are no parameters just send

```
command:;
```

The following commands are implemented:

Command	Parameters	Description
light	on off	to switch on or off all the leds of the led strip
led	on off	to switch the single led on the board on or off
r g b	intensity	to set the color of the rgb leds, the intensity of the three colors red, green and blue can be set. The intensity has to be between 0 (black/off) and 70 (bright). e.g. to set the intensity of green to 25 send g:25;
run	on off	switch the running light on and off.
rc	led count	to set the number of leds which are switched on at the same time during the running light
delay	time in ms	to set the delay/speed for the running or blinking light. delay:20; is fast, delay:1000; is slow
blink	on off mode(1,2)	switch the blinking mode on and off or set it to a specific blink mode. Currently only mode 1 and 2 are implemented (send blink:2;)

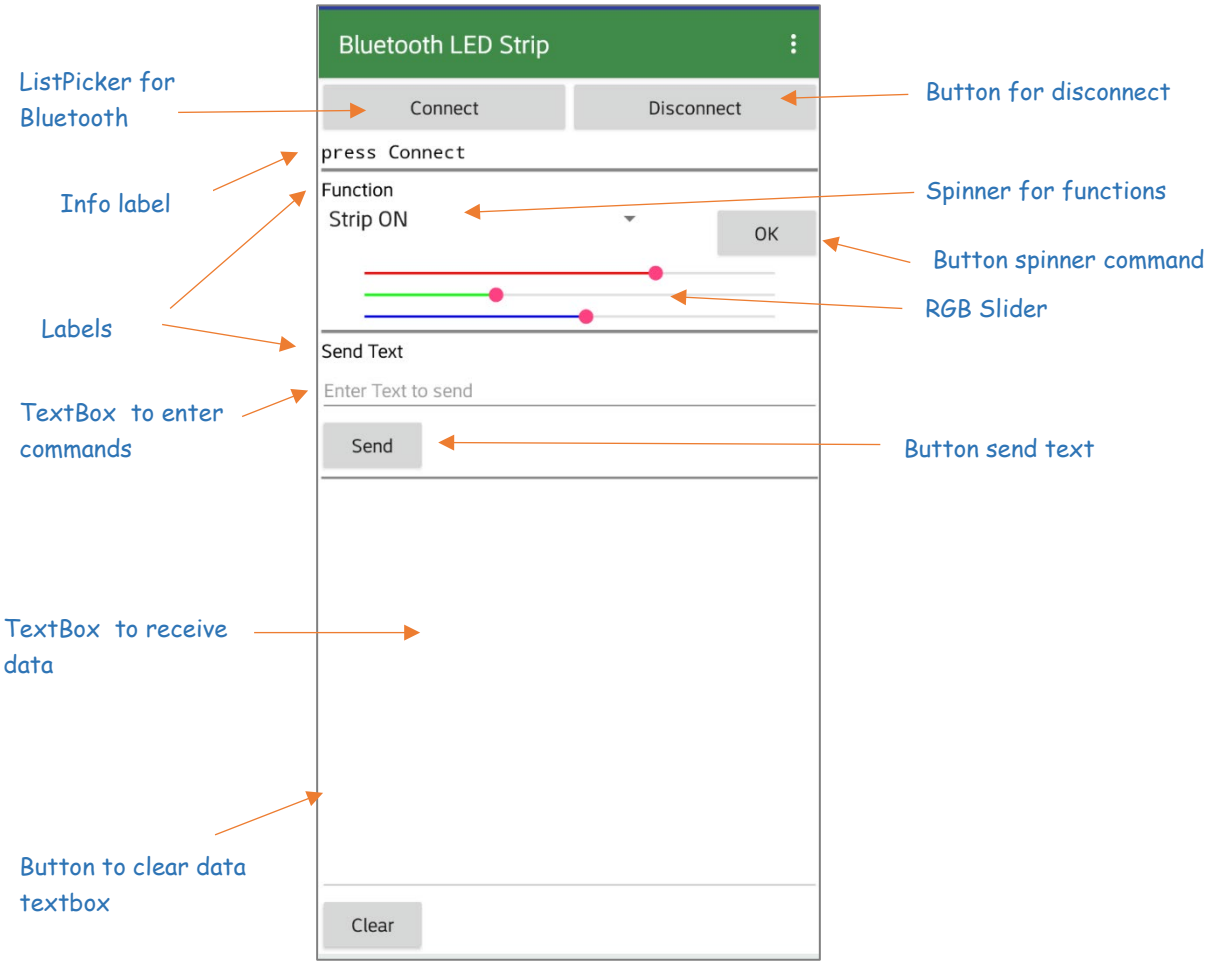
Command	Parameters	Description
r2 g2 b2	intensity	to set the intensity of the second color for the blinking mode.
p	number	set one specified led/pixel on or off. leds are numbered from 0 to 9. p:4; switches the 5th led on or off. The currently active color is used.
clear		switches the leds off
info		sends some info back to the connected device
d		to switch debug info on or off (default off)

Try out and test the led module using the serial terminal and different commands.

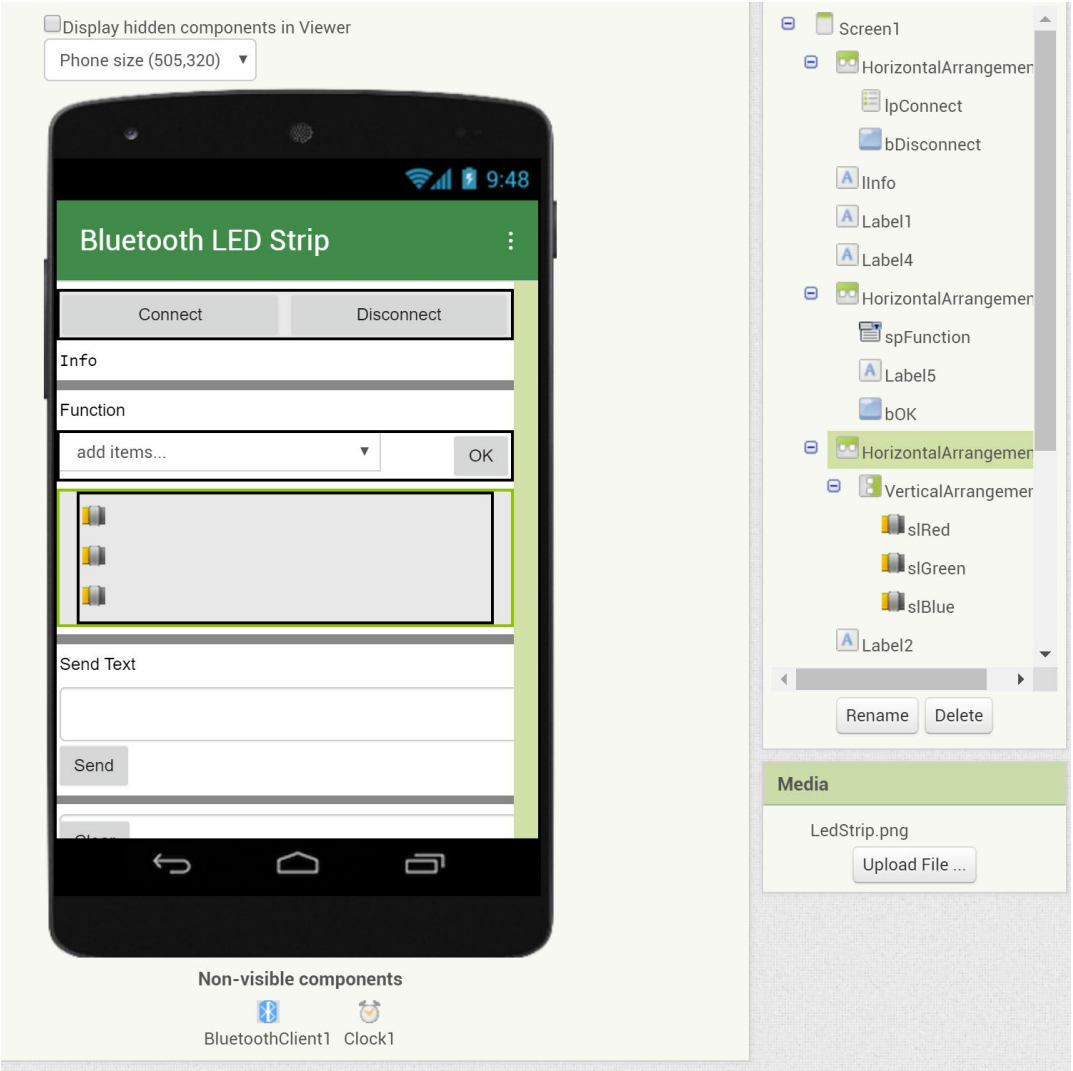
Mobile App

After Testing the Led module let's build the mobile app using the MIT App Inventor.
http://ai2.appinventor.mit.edu/

Start with the Designer and the user interface. It should look like this:



In the App Inventor it looks more like this:

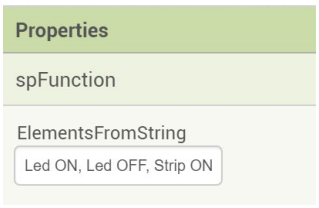


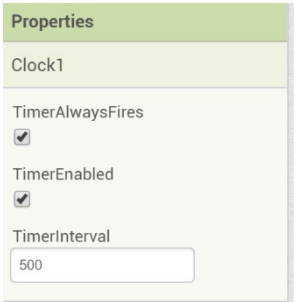
There are also two invisible components. Add the BluetoothClient (Connectivity) and a Clock (Sensors) to the Screen. Note the arrangement of the sliders to prevent them from sticking to the edge of the layout. Set labels and titles according to the screenshots.

Rename the components and give them meaningful names. Following names have been used in the sample app:

- lpConnect … ListPicker for the Bluetooth connection
- bDisconnect … Disconnect button
- lInfo … info label below the connection controls to show the connections status
- spFunction … Function spinner (i.e. combo box to select different functions from a drop box)
- slRed, slGreen, slBlue … for the color sliders
- bOK … OK button to send command from the spinner
- txtSend, txtData … textboxes to send and receive data
- BluetoothClient1 … default name for the Bluetooth client
- Clock1 … default name for the clock timer

Fill the function spinner with functions. Use a comma separated list for the entries and set the timer interval to 500

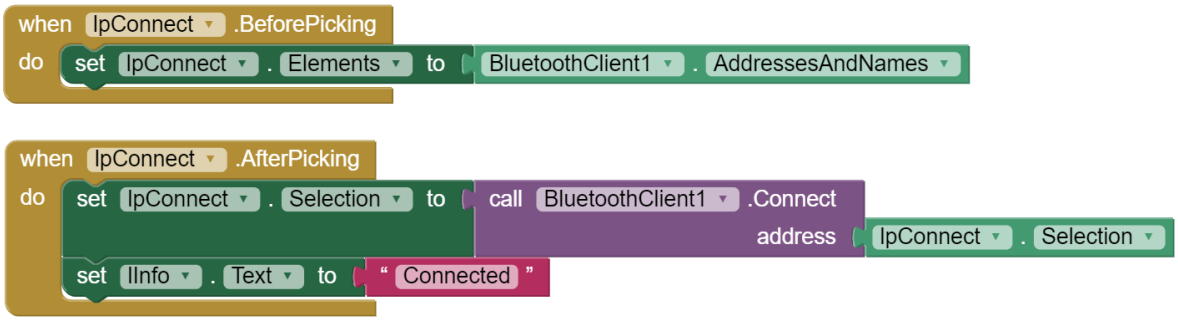




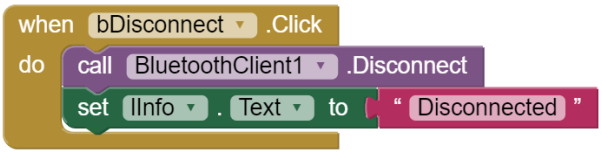
Programming Blocks

After the user interface let's start programming and to put some logic and functions behind the interface.

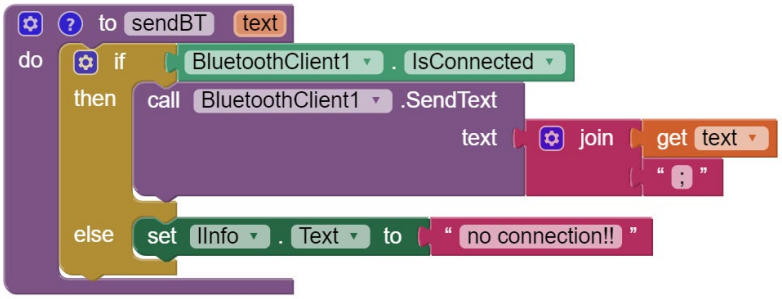
Start with the Bluetooth connection. First, we have to use the ListPicker to show available Bluetooth devices and connect to the picked device.



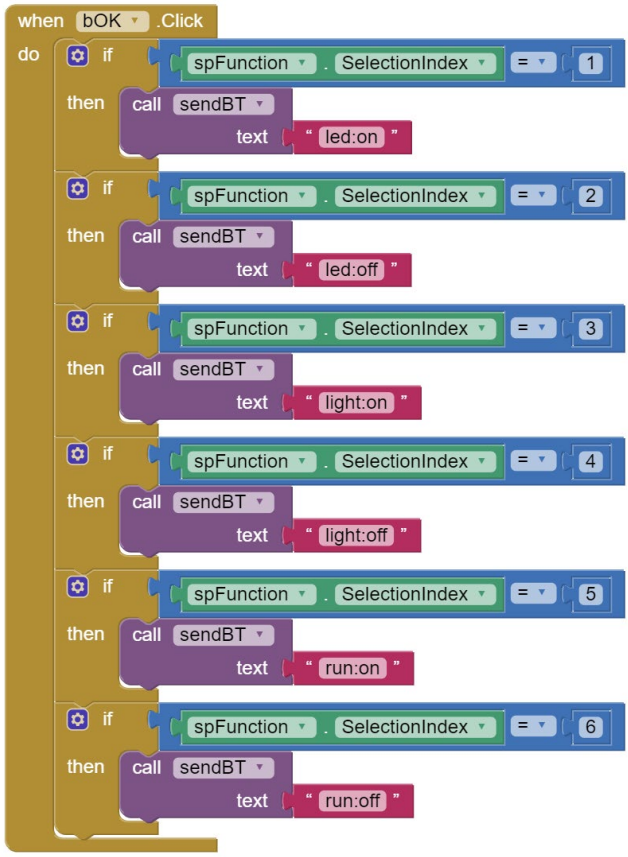
The Disconnect button needs some functionality too:



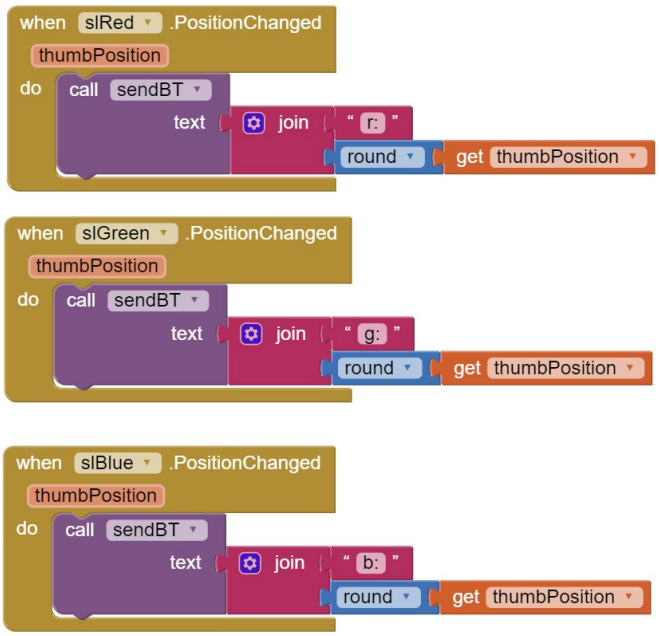
To send a command via Bluetooth define a procedure `sendBT` first. This procedure is used every time a command is sent to the LED module.



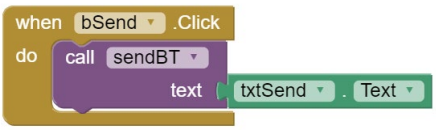
Now that we can send text via Bluetooth it is possible to use the spinner and the OK button to send the command selected with the spinner. Every entry in the spinner has as specific index which is used to send the right command when the OK button is clicked.



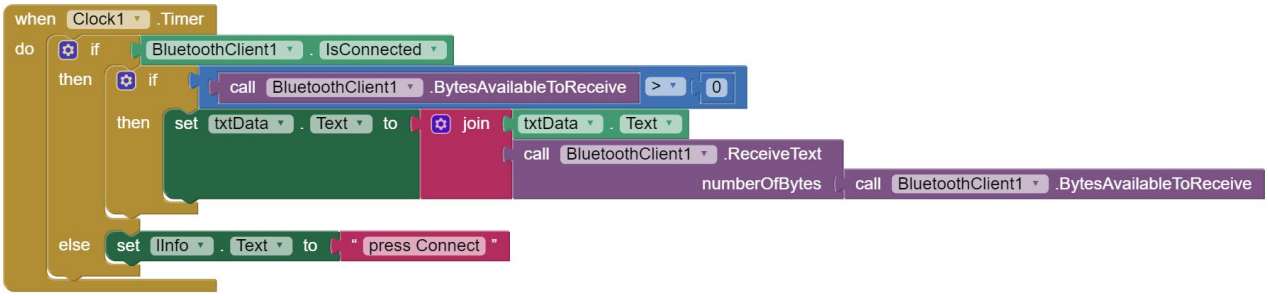
We can also send a command when the sliders are changed. Floating point numbers must be rounded.



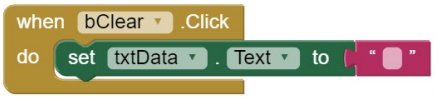
To send some commands manually using the Send-textbox and the Send-button we just send the text when the button is clicked. Commands must be entered without semicolon since the procedure `sendBT` adds the semicolon while sending the data.



To receive data from the Bluetooth client we use the timer to check in the defined interval if new data is available. If some text is available, it is written into the data textbox.



Last but not least, the Clear button clears the data textbox if necessary.



Have fun and improve the app with other features.