# Lab 6: Controlling the BLEPad from Android via PD

In this weeks lab we will be introducing the board that will be used for your project. So far we have learned how to create a GUI for our app and how to send data from this GUI to our pd patch and from our patch to our GUI. Today we are going to learn how to control an LED attached to the BLEPad via a button click.

## Setup

Download lecture6.pdf from here: <https://github.com/Hydroxate/Notes-Performance-Tech-1>

You must read this.

Download the example from Github:

<https://github.com/Hydroxate/Android-BLEPad>

Rename the containing folder to firstname\_surname\_lab06.

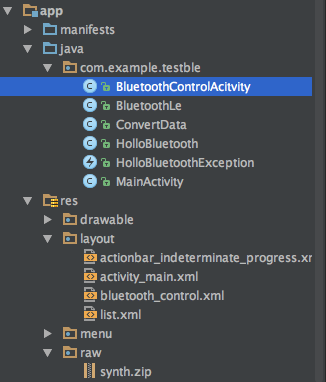
(This step is important as otherwise another student may accidentally delete your project.)

Open the provided template in android studio.

## Part 1: BLE-Template

### Familiarization

When the project has been opened, look at the project pane.



**NOTE:**

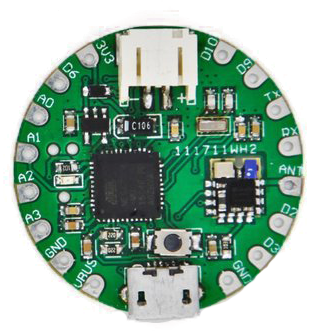
**The only file you edit is BluetoothControlActivity.java and it’s respective Layout bluetooth\_control.xml**

**All other files are to be left as they are as they are vital to the bluetooth process.**

Previously, we would edit MainActivity but this is the default screen to search for BLE (bluetooth low-energy) devices. Upon clicking the device you wish to connect to (in this case, BLEPad), the screen that is shown is the BluetoothControlActivity screen. There is a lot of Bluetooth-handling code under the hood here so be careful where you put your listeners in onCreate. Remove any bluetooth code and I will murder you.

## Part 2: The BLEPad

### Layout



Above you see the BLEpad. It is similar in function to the Arduino but it’s size and shape allows easier use for wearable technology. **Marked with an X are the pins you do not clip to.**

The following are the pins we will use:

**GND** – Ground, used to close a circuit loop

**A0,A1,A2,A3** – Analog input/output pins. These support input values from 0 – 1023 and output values of 0 – 255.

**D2,D3,D6,D9,D10** – Digital input/ output pins. These support input and output values of 0 and 1.

We will be using crocodile clips for the lab to connect devices to the BLEPad.

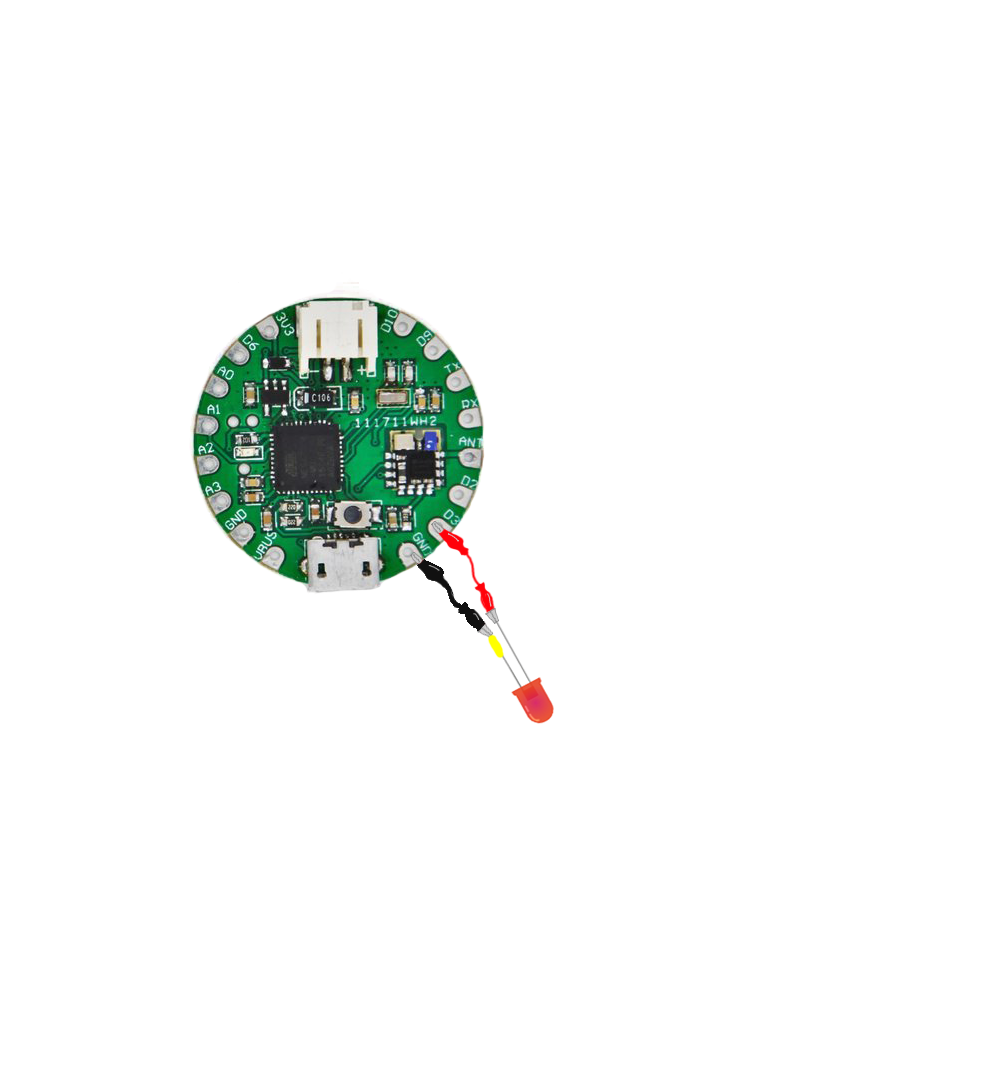
**Crocodile Clip:**



To reiterate from the lecture 6 notes, **connect the outputs/inputs first without power.** Before you connect the blePAD to power, check with myself or Nick to make sure the board does not go up in smoke.

The boards cost €20 and take 4 weeks to arrive from China.

In this lab we will be using a red LED. The circuit is as following:



The D3 pin is connected to the leg of the LED without the yellow wire wrap. GND is connected to the leg with the yellow wire wrap (This is the positive leg that has a 75ohm resistor attached).

### Testing LED

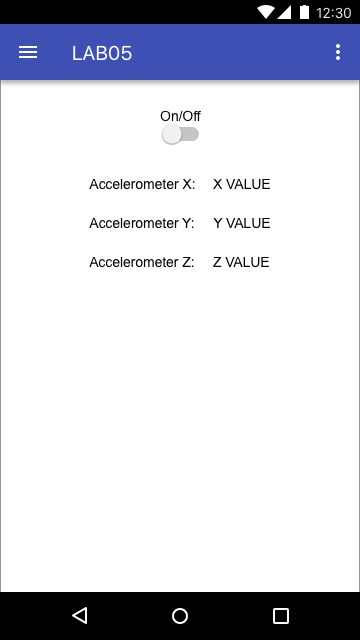
To test the LED, attach the red crocodile clip to the 3.3V and the LED should turn on.

Reattach the red crocodile clip to the D3 pin.

## Part 3: Adding the switch to the GUI.

### Step 1: Add the widget graphically

1. Click on bluetooth\_control\_activity.xml in the project pane, it is located under res/layout.
2. Add a slider widget, with ID onOffSwitch.



### Step 2: Create an instance in code

At this stage our widgets only exist as graphical objects, essentially a picture on the canvas. We have finished step 1 of our 2 step process. On to step two, “create an instance of the widget type in code with a unique name and link this to the graphical object we just made.”

1. Double click the BluetoothControlActivity.java file in the project pane. This is located under the java folder.

Lets declare and instantiate our switch. Go to line 209 in the OnCreate method and type the below:

Switch onOffSwitch = (Switch) findViewById(R.id.*onOffSwitch*);//declared the switch here pointing to id onOffSwitch

Now let’s create the listener of the switch under the previous line in the OnCreate() method. It will look like the below:

//Check to see if switch1 value changes  
onOffSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 float val = (isChecked) ? 1.0f : 0.0f; // value = (get value of isChecked, if true val = 1.0f, if false val = 0.0f)  
 sendFloatPD("onOff", val); //send value to patch, receiveEvent names onOff  
  
 }  
});

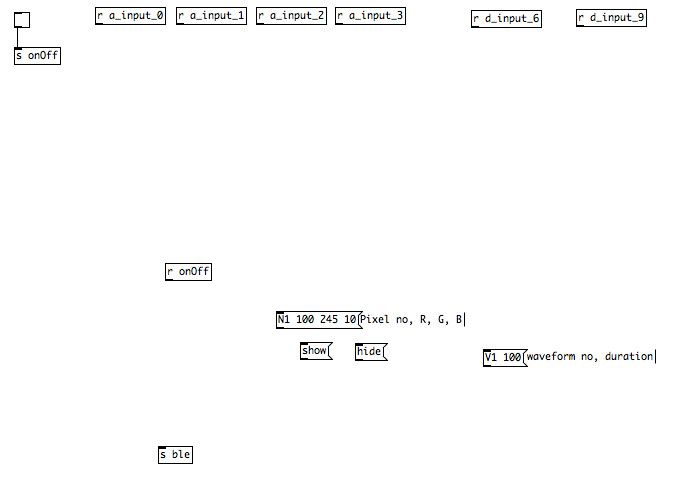
We have now completed step 2 of our two step process. We have created a switch instance with a unique name and linked it to the graphical switch object that we placed on the canvas in step 1.

## Part 4: Pure Data Patch

### Step 1: BLE enabled Patch

1. Right-click on synth.zip folder and click “Reveal in Finder”.
2. Double-click the zip to extract and open synth.pd.

You will see the following:



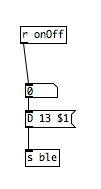
This patch is also the patch you will be extending for your final project.

It has [receive] objects for all the analog and digital inputs, and [s ble] object for the outputs of the BLEPad. Since our LED is connected to D3 on the BLEPad, we will use the [s d\_ouput\_3] object to turn the LED on and off.

|  |  |
| --- | --- |
|  |  |

### Step 2: Add the functionality for turning LED on/off

1. Add the [receive] object for onOff.
2. Add a number object.
3. Connect the output of the [r onOff] to the input of the number object.
4. Connect the output of the number object to the a message [D 13 $1]. The value passed into the message will replace the $1.
5. Attach the message to the input of the [s ble] object as shown below



1. Zip the synth.pd.
2. Delete the synth.zip .
3. Rename the synth.pd.zip to synth.zip
4. Delete synth.pd.

### Step 3: Updating pin of LED

The 13 in the message in the patch here represents the on-board LED . First try controlling this with the Switch in the GUI. Then we’ll try switching it to 3 and with a physical LED.

Run the application in Android Studio.

Upon clicking the switch in the GUI, the LED should switch on and off respectively.

## Going Onwards

For the remainder of this lab, get creative with the output of LED flashing, introduce counters, metros, lines etc.

The homework for this lab will be sent out shortly.