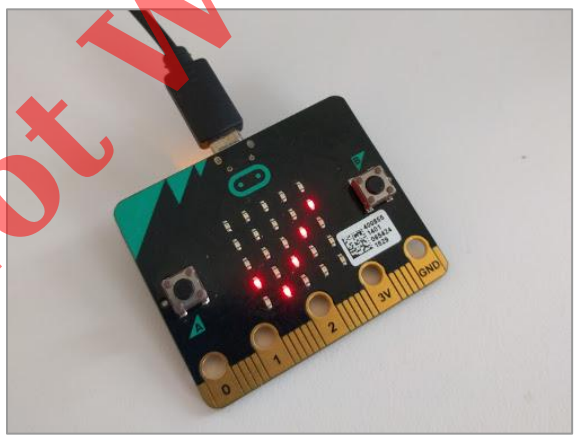
App Inventor + IoT: Micro:bit LED

This tutorial will help you get started with App Inventor + IoT and the LED light grid (light emitting diode… basically a small light) on a micro:bit controller.



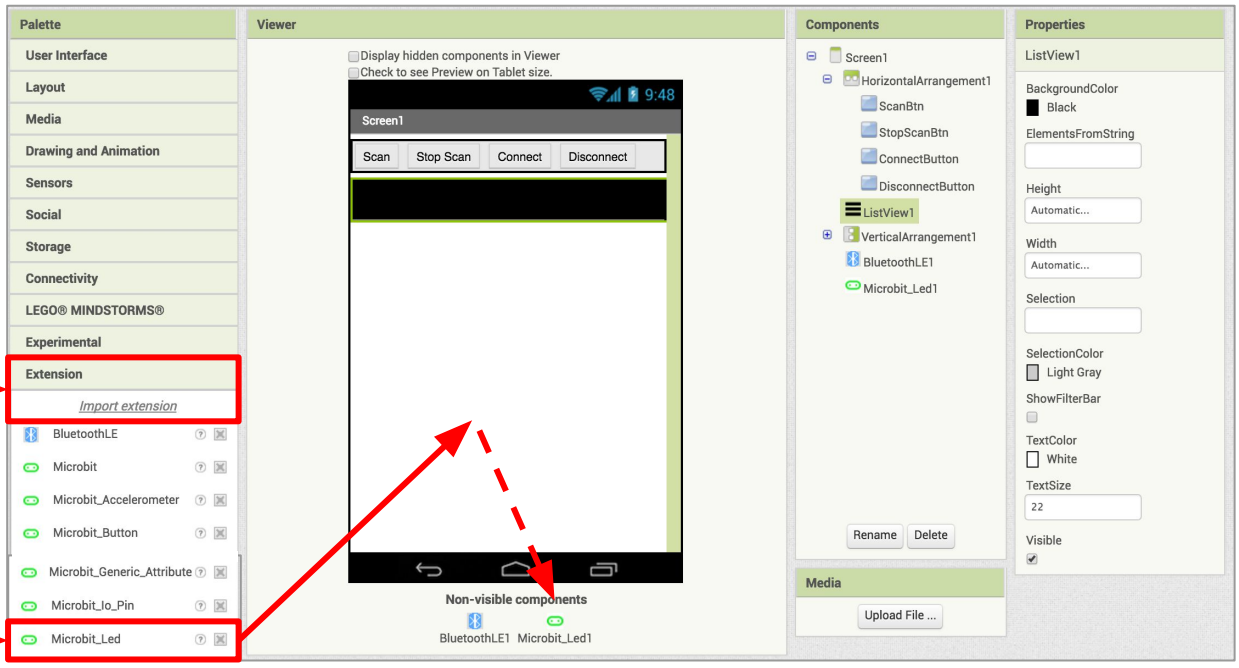
First, you will need to pair your phone or tablet to the micro:bit controller, using these directions. Your device must be paired with the micro:bit in order for the app to work.

Next, you should complete the App Inventor + IoT Basic Connection tutorial to make a basic connection to the micro:bit device. If you prefer, you can download the completed .aia file here.

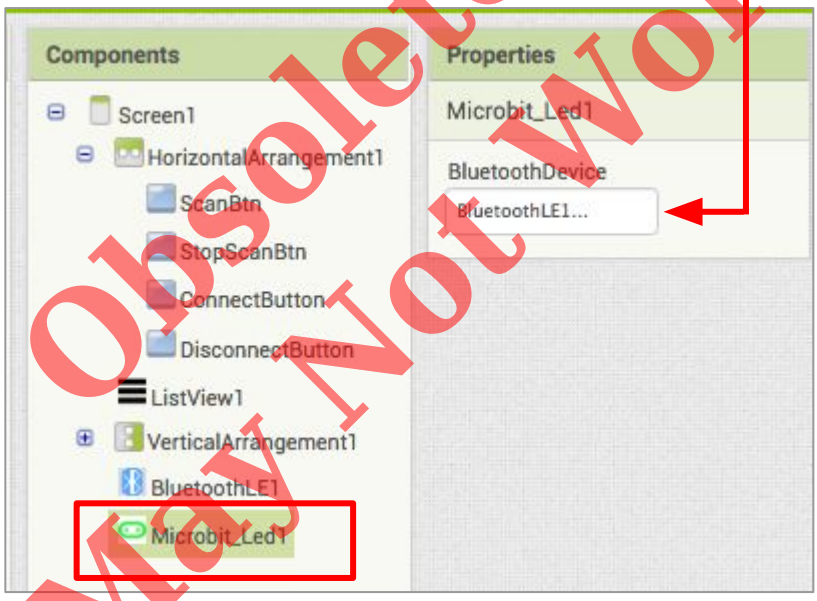
The remaining steps all build off of the the starter code for Basic Connection tutorial and .aia.

First, we need to add the necessary extension.

- In the Palette window, click on Extension at the bottom and then on "Import extension" and click on "URL".
  - Paste in this URL: http://iot.appinventor.mit.edu/assets/com.bbc.micro:bit.profile.aix
- Add the **Microbit_Led** extension to your app by dragging it onto the Viewer.

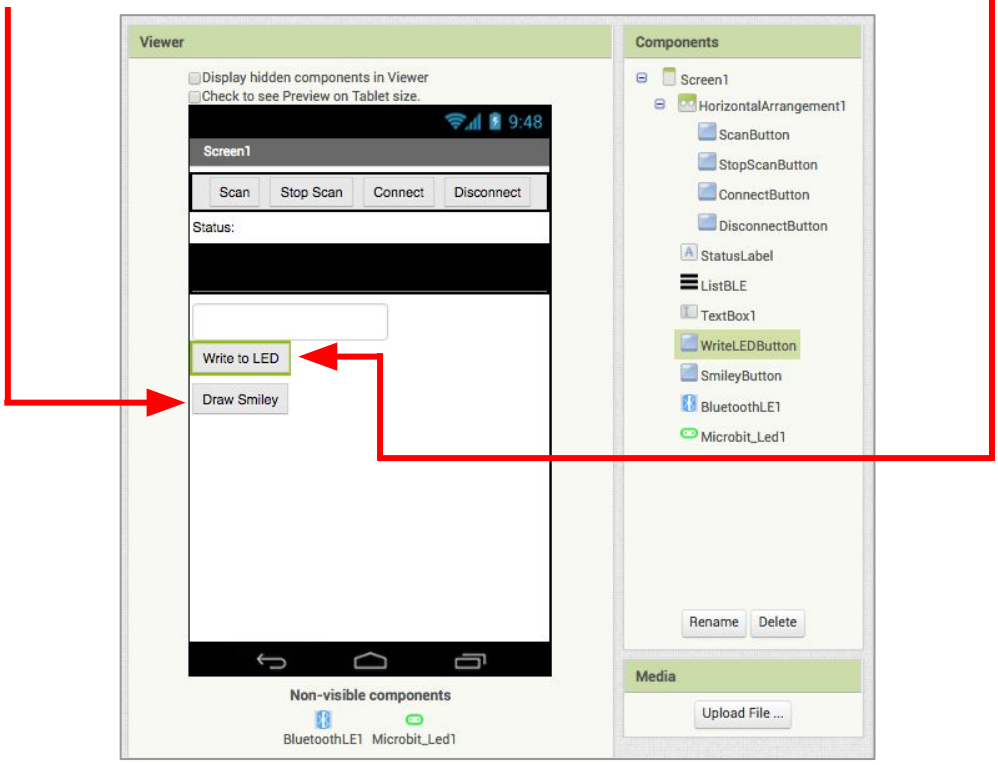


- Click on **Microbit_Led1** in the Components pane.
- In the Properties tab for the **Microbit_Led1**
    - Set *BluetoothDevice* to "BluetoothLE1".



Let's add more components to our app to control the LED matrix.

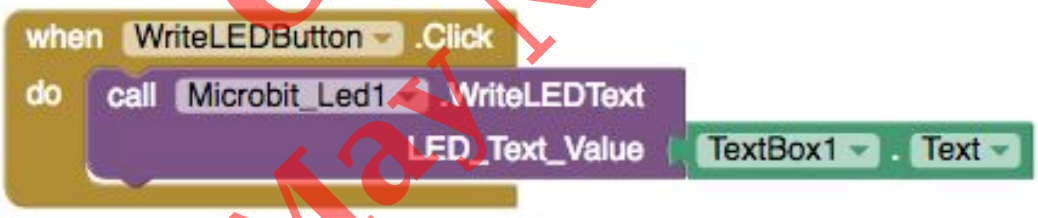
- From the User Interface drawer in the Palette, drag in a **Textbox**, and two **Buttons**.
- Rename the first button "WriteLEDButton" and change its Text property to "Write to LED".
- Rename thte second button, "SmileyButton", and change its Text property to "Draw Smiley.



Now switch to the Blocks Editor view

Each button will perform a task to display something on the LED matrix on the micro:bit. First we'll code the **WriteLEDButton**.

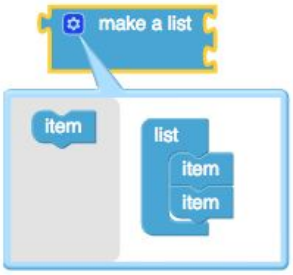
- From **WriteLEDButton** in the Blocks pane, drag out a **when WriteLEDButton.Click** block.
  - From **Microbit_Led1** in the Blocks pane, drag out, a **Microbit_Led1.WriteLEDText** block.
  - From **Textbox1** in the Blocks pane, drag out a **Textbox1.Text** and snap to the **Microbit_Led1.WriteLEDText** block.



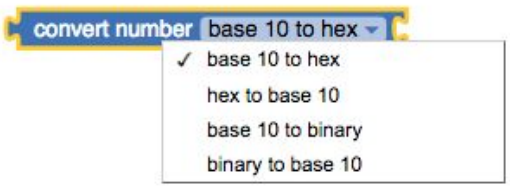
For the **SmileyButton**, we're going to create a procedure to draw a smiley face on the LED matrix.

- From the **Procedures** drawer, drag out a to procedure block, and rename the procedure "DrawSmiley".
- From the **Microbit_Led1** drawer, drag out a **Microbit_Led1.WriteLEDMatrixState** block and snap into the procedure.

  - From the **Lists** drawer, drag out a **make a list** block.
  - Click on the blue gear icon, and drag three **item** blocks to the **make a list** block so there are 5 empty slots.



  - From the **Math** drawer, drag out a **convert number** block.
  - Pull down the dropdown in the **convert number** block, and select "binary to base 10".
  - Drag out 4 more **convert number** blocks and select "binary to base 10".
  - From the Text drawer, drag out 5 blank text blocks.



- In each of the text blocks, type in the following text strings: (01010, 01010, 00000, 10001, 01110)

Your final procedure should look like the code below.

Each string represents a row in the matrix. A “1” signifies an “on” light, and a “0” signifies an “off” light.

The last thing is to code the button.

- From **SmileyButton** in the Blocks pane, drag out a **when SmileyButton.Click** block.
- From the Procedure drawer in the Blocks pane, drag out a **call DrawSmiley** block.

Your app should now be working! Test it out by connecting your micro:bit device using the MIT AI2 Companion (if you haven't already). Make sure you have paired the Bluetooth on your Android device to your micro:bit first! You should be able to see the Smiley face appear, as well as scrolling text you type into the text box, after pressing the appropriate buttons in the app.