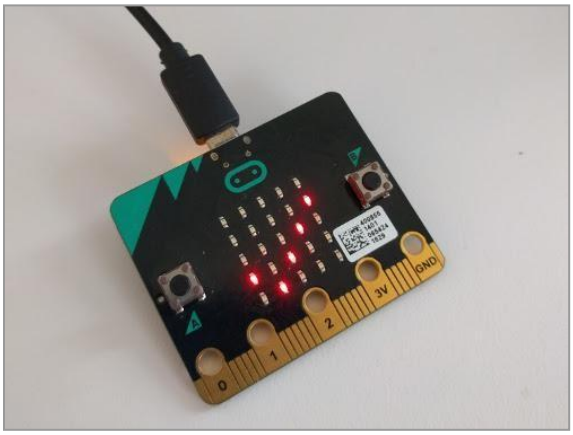
App Inventor + IoT: Micro:bit Button

This tutorial will help you get started with App Inventor + IoT and the two buttons on a micro:bit controller.



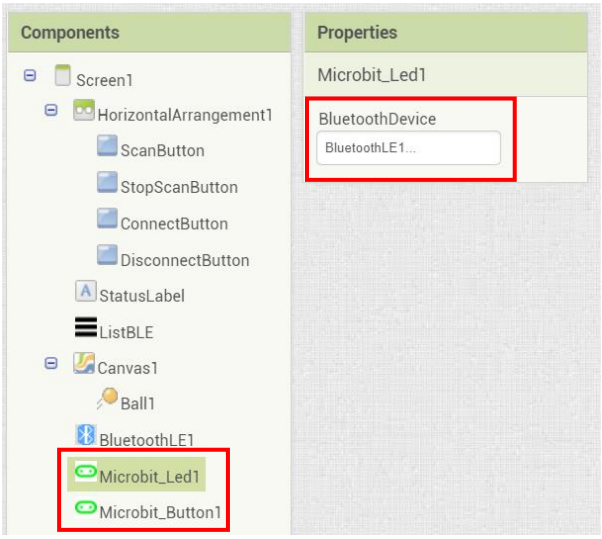
First, you will need to pair your phone or tablet to the micro:bit controller, using these directions. Your device must be paired with the micro:bit in order for the app to work.

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.

App Inventor's micro:bit button component's document

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.

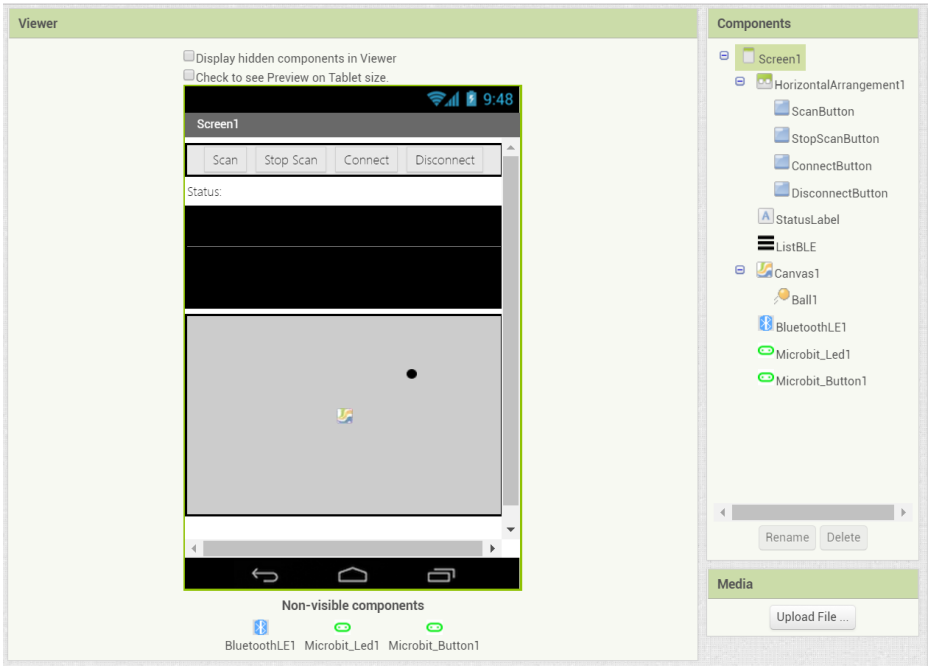
- In the Palette window, click on Extension at the bottom and then on "Import extension" and click on "URL".
    - Paste in this URL: http://iot.appinventor.mit.edu/assets/com.bbc.micro:bit.profile.aix
- Add a **Microbit_Buttons** extension to your app by dragging it onto the Viewer, set its *BluetoothDevice* to "BluetoothLE1".
- Add a **Microbit_Led** extension, also set its *BluetoothDevice* to "BluetoothLE1".



Let’s add more components to our app to receive the micro:bit buttons’ statuses.

- From the Drawing and animation drawer in the Palette, drag in a **Canvas** and a **Ball**. Set Canvas’s height to 320 pixels, width to fill parent (or any parameters you like).

Your designer page should look like this:

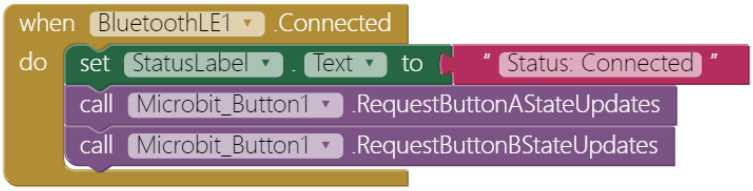


Now switch to the Blocks Editor view

We would like to control Ball component's horizontal movement by the two buttons on micro:bit controller. Let's begin:

STEP1: Request updates when connected

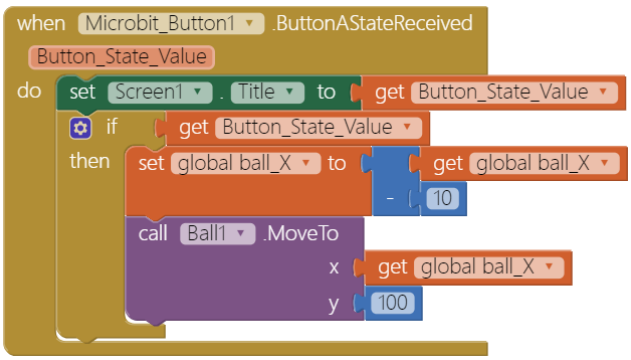
In BluetoothLE1.Connected event, we show related message and request micro:bit to update two buttons' statues.



STEP2: Micro:bit's A button pressed

In **Microbit_Button1.ButtonAStateReceived** event:

- If A button is pressed(Button_State_Value is true), then we set **ball_X** variable decrease by10.
- Make Ball1 component move to position (ball_X, 100) to make it move left by 10 pixels.



STEP3: Micro:bit's B Button pressed

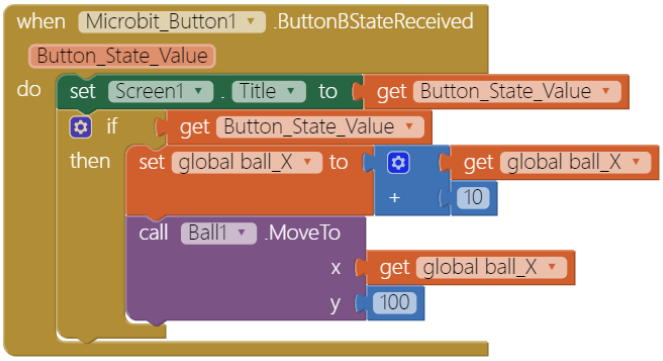
For the **Micro:bit's B Button**, things are almost the same except for the opposite direction.

In **Microbit_Button1.ButtonBStateReceived** event:

- If A button is pressed(Button_State_Value is true),

then we set **ball_X** variable increase by10.

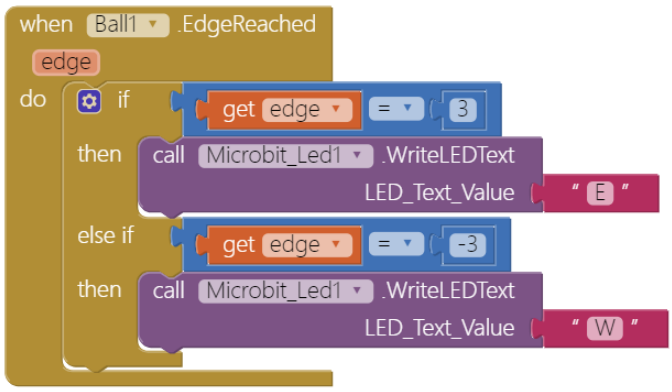
- Make Ball1 component move to position (ball_X, 100) to make it move left by 10 pixels.



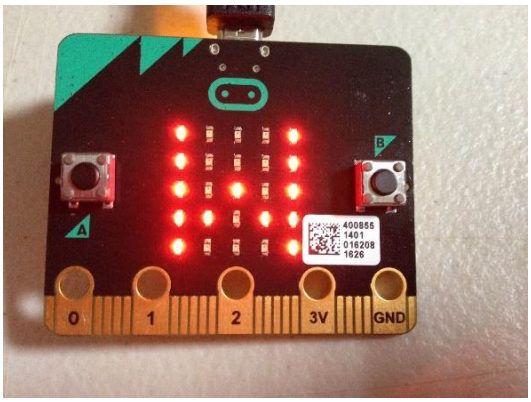
STEP4 **Ball reached canvas' edge**

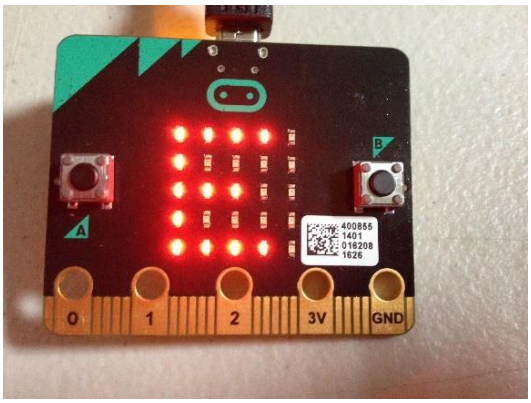
When Ball1 reached canvas's edges, we will show corresponding character on micro:bit's Led matrix.

In **Ball1.EdgeReached** event, we use if / else if to check which edge is reached, then send '**E**' and '**W**' character(means **East** and **West**) to Micro:bit using **Microbit_Led1.WriteLEDText** method.



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) or install by .apk. Make sure you have paired the Bluetooth on your Android device to your micro:bit first! Try to press the two buttons on micro:bit, and the ball on screen should move toward left or right.


A
B
0
1
2
3V
GND


A
B
0
1
2
3V
GND