App Inventor + IoT:
Flying bees with Micro:bit
Magnetometer sensor

60 min

(with Basic Connection tutorial completed)

Level: advanced

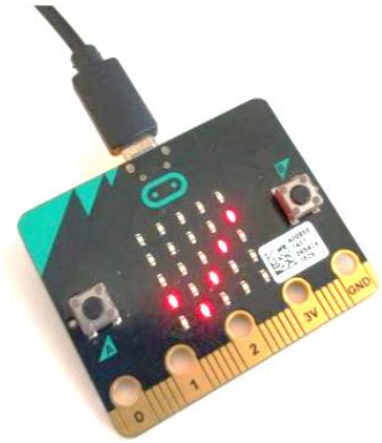
This tutorial will help you get started with App Inventor + IoT and the magnetometer sensor on a BBC micro:bit controller.

Hardware

You only need one **BBC micro:bit** to get started with this project.

Paring with Micro:bit

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.



App Inventor

This app can let you control a small Bee icon in the app to move around according the magnetic field around micro:bit. Now log into **MIT App Inventor site** and create a new project.

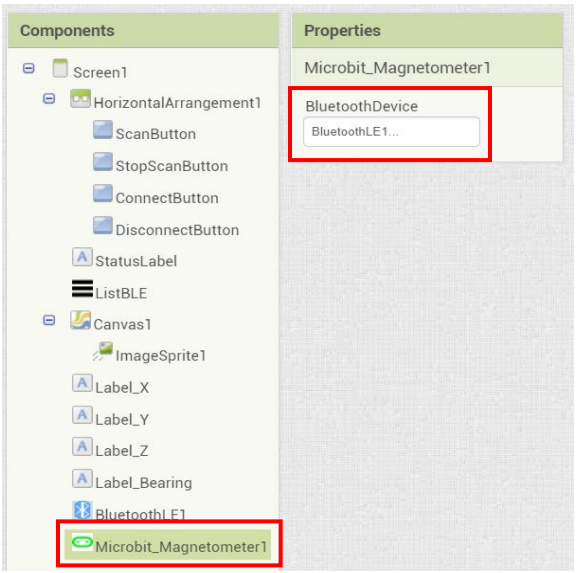
You should complete the **App Inventor + IoT Basic Connection tutorial** to make a basic connection to the micro:bit device. If you prefer, you can download the completed .aia file **here**.

- App Inventor's micro:bit magnetometer component's document

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

First, we need to add the necessary extension.

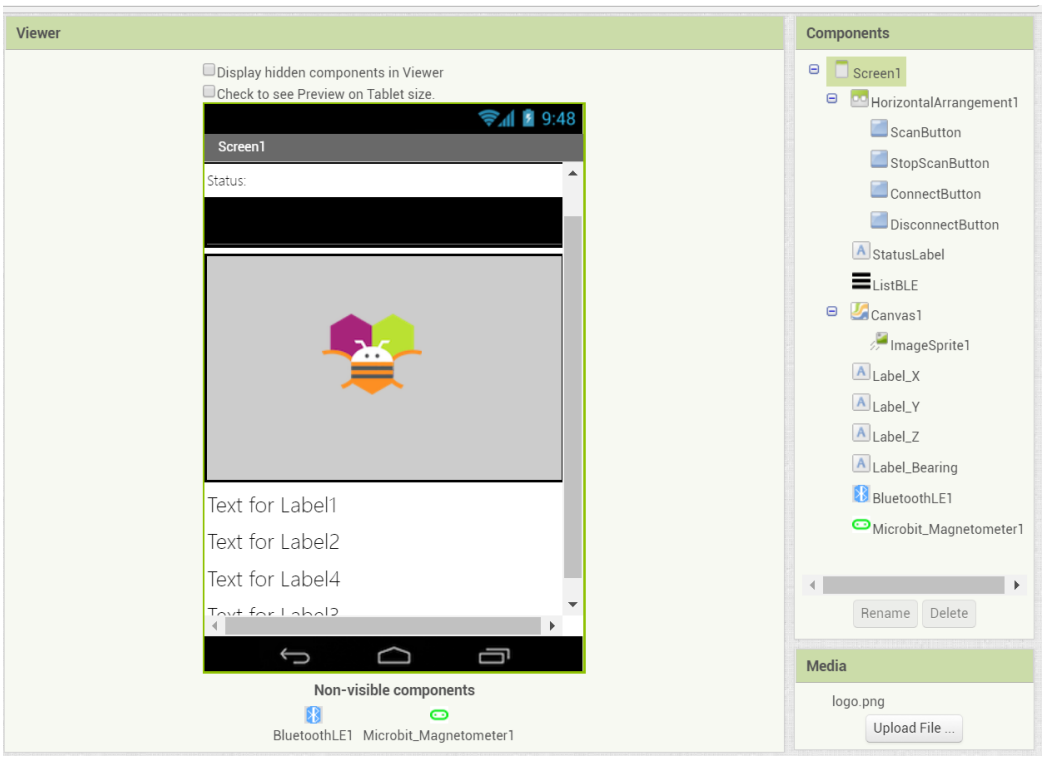
- In the Palette window, click on Extension at the bottom and then on "Import extension" and click on "URL".
    - Paste in this URL: http://iot.appinventor.mit.edu/assets/com.bbc.micro:bit.profile.aix
- Add a **Microbit_Magnetometer** extension to your app by dragging it onto the Viewer, set its **BluetoothDevice** to "BluetoothLE1"(Don't forget!).



Let’s add more components to our app to read the magnetometer status.

- From the Drawing and animation drawer in the Palette, drag in a **Canvas** and an **ImageSprite**. Set Canvas’s height to 320 pixels, width to fill parent (or any size you like).
    - Set **ImageSprite**’s Picture to some cute image (no bigger than the canvas).
- Add four Label to show Magnetometer’s X, Y, Z axis and bearing value.

Your designer page should look like this:


Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Screen1
Status:
Text for Label1
Text for Label2
Text for Label4
Non-visible components
BluetoothLE1
Microbit_Magnetometer1
Components
Screen1
HorizontalArrangement1
ScanButton
StopScanButton
ConnectButton
DisconnectButton
StatusLabel
ListBLE
Canvas1
ImageSprite1
Label_X
Label_Y
Label_Z
Label_Bearing
BluetoothLE1
Microbit_Magnetometer1
Rename
Delete
Media
logo.png
Upload File ...

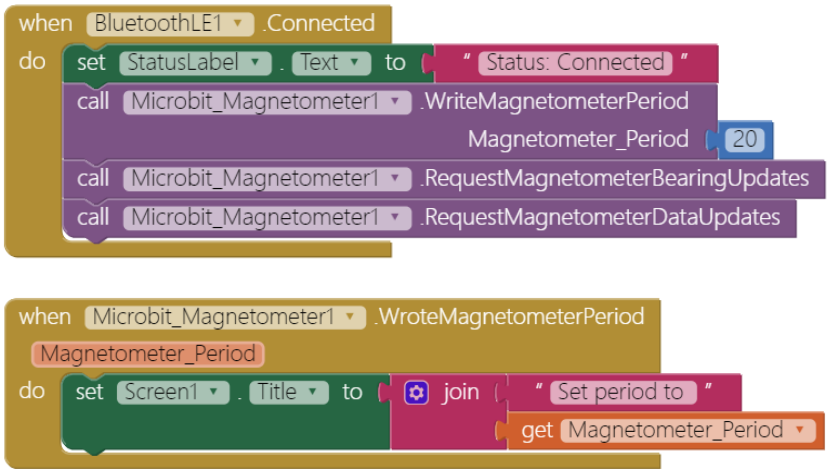
Now switch to the Blocks Editor view

We would like to control ImageSprite's heading by the Z-axis movement of magnetometer on micro:bit controller. Let's begin:

STEP 1: Request updates when connected

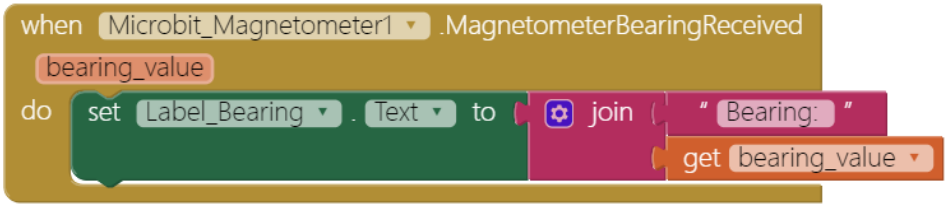
In the **BluetoothLE1.Connected** event, we show messages and tell the micro:bit to update the magnetometer's state.

And in the **Microbit_Magnetometer1.WroteMagnetometerPeriod** event, we show the related message and the period value just set.



STEP2: Show bearing value

Whenever the Micro:bit's bearing value is changed(**Microbit_Magnetometer1.MagnetoBearingReceived** event), we show the bearing value on the label.

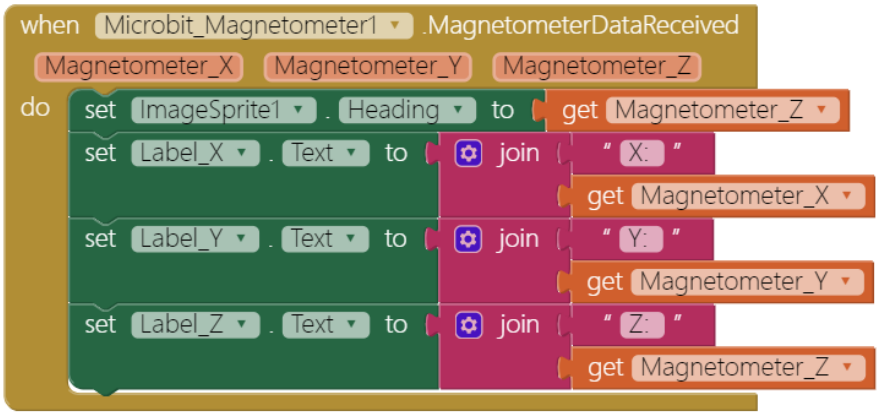


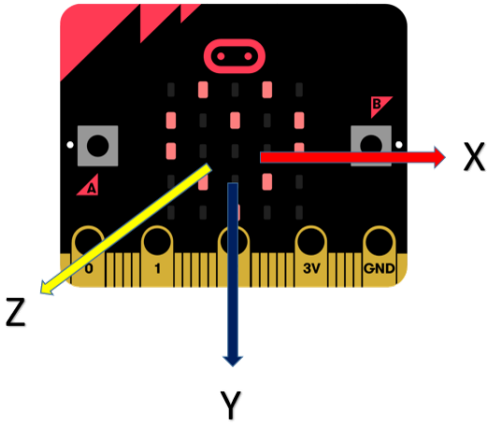
STEP3: Show XYZ data and control ImageSprite

Same idea, whenever the magnetic situation around Micro:bit is changed (**Microbit_Magnetometer1.MagnetoDataReceived** event),

we do things below:

- Set **ImageSprite**'s heading to **Magnetometer_Z** value.
- Show X, Y and Z axis value on corresponding label, check image below for micro:bit's axis.





(From https://makecode.microbit.org)

Tips

Your app should now be working! Make sure you have paired the Bluetooth on your Android device to your micro:bit. Then test it out by connecting your micro:bit device using the MIT AI2 Companion (if you haven't already) or installing it by .apk. Try to shake or flip around your micro:bit or have a small magnet (not too strong or it may influence your device!) point toward it, you should see the App Inventor logo turning and turning!

Brainstorming

1. Try to add more cute movement into your app, for example, you can use X and Y axis values to make ImageSprite move left and right and show something when it bumps the edge of the Canvas. (Refer to our **Micro:bit button tutorial**).
2. Add some sound effects when the magentosensor value have exceeded certain level.