App Inventor + IoT: Micro:bit Temperature



This tutorial will help you get started with App Inventor + IoT and the temperature sensor on a <u>micro:bit</u> controller.

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

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



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.microbit.profile.aix

• Add the **Microbit_Temperature** extension to your app by dragging it onto the Viewer.

	Palette	Viewer	Components	Properties
	User Interface	Display hidden components in Viewer Check to see Preview on Tablet size,	B Screen1	Screen1
	Layout	9:48 🗐 🕄 😒	HorizontalArrangement1 ButtonScan	AboutScreen
	Media Drawing and Animation	Screen1	ButtonStopScan	
		Scan Stop Scan Connect Disconnect	ButtonConnect	AlignHorizontal
	Sensors	Status:	ButtonDisconnect	AlignVertical
	Social	Data:	A LabelStatus	Top : 1 👻
	Storage		ListBLE	AppName
	Connectivity		BluetoothLE1	BackgroundColor
	LEGO® MINDSTORMS®		Microbit_Temperature1	White
	Experimental			BackgroundImage
	Extension			None
	Import extension			CloseScreenAnimation Default -
	BluetoothLE 🕐 🔣			Icon
	💿 Microbit 🕐 🔣			None
	💿 Microbit_Accelerometer 🛞 📓			OpenScreenAnimation
	🖸 Microbit_Button 🕐 📓		Rename Delete	ScreenOrientation
	💿 Microbit_Led 🕐 🕱		Media	Unspecified -
	💿 Microbit_Magnetometer 🕐 📓	Non-visible com, onents		Scrollable
	💿 Microbit_Temperature 💿 📓	BluetoothLE1 Microbit_Temperature1	Upload File	ShowListsAsJson

- Click on **Microbit_Temperature1** in the Components pane.
- In the Properties tab for the **Microbit_Temperature1**
 - Set *BluetoothDevice* to "BluetoothLE1". -

omponents	Properties
Screen1	Microbit_Temperature1
HorizontalArrangement1	BluetoothDevice
ButtonScan	BluetoothLE1
ButtonStopScan	
ButtonConnect	
ButtonDisconnect	
A LabelStatus	
A LabelData	
ListBLE	
Bluetoothi E1	

- Drag a Label from the User Interface Palette and drop it between LabelStatus and ListBLE
 - Rename the Label "LabelData".-
 - Change its text to "Data: ".

wer		Components	Properties
	Display hidden components in Viewer Check to see Preview on Tablet size.	Screen 1 Screen 1 Screen 1 SuttonScan ButtonStopScan ButtonDisconnect LabelStatus LabelData	LabelData
	Screen1 Scan Stop Scan Connect Disconnect		BackgroundColor None FontBold
	Status: Data:		FontItalic FontSize
		ListBLE	FontTypeface default ✓ HTMLFormat
			☐ HasMargins ✔ Height
			Automatic Width
			Automatic
		Rename Delete	Data:
		Media	TextAlignment
Non-visible components		Upload File	TextColor Black

Now switch to the Blocks Editor view

First, we want to request data updates when the sensor value on the micro:bit changes.

 from Microbit_Temperature1 in the Blocks pane, add call Microbit_Temperature1.RequestTemperatureUpdates to the existing when BluetoothLE1.Connected block from the Basic Connection tutorial.



Next, we need to store the data we receive from the sensor. From the Variables drawer in the docs pane, drag an **initialize global name to** block and name it "Temperature". From the Math drawer add a number block and set it to "0". We'll use this to keep track of the sensor value.

initialize global Temperature to

Let's make a new procedure to display the current readings in the **LabelData** when we get new data. You can create a procedure by dragging out a purple procedure block from the Procedures drawer in the Blocks pane. Let's rename it **updateDataLabel.**

- from LabelData in the Blocks pane, add set LabelData.Text to.
- from the Text drawer connect a **join** block.
 - From the Text drawer, connect a text block and type
 "Temperature: "
 - From the Variables drawer connect a get global Temperature.



Finally, we need to call the procedure when this data is received.

- From the Microbit_Temperature1 drawer in the Blocks pane, drag when Microbit_Temperature1.TemperatureReceived
 - from the Variables drawer, add **set global Temperature.**
 - Hover over the orange "temperature" in .TemperatureReceived to see the get temperature block. Drag the get temperature block from this window and snap to set global Temperature.
 - from the Procedures drawer, add call updateDataLabel.

whe	peratureReceived	
 temperature		
do	set global Temperature - to	get temperature
	call updateDataLabel	•

Your app should now be working! Connect your micro:bit device using the companion (if you haven't already). Test it out by closing your hand around the micro:bit. If it is working, you should see the data label change (but probably very slowly).

