App Inventor + IoT: LED

(with IoT Setup and Basic Connection tutorials completed)

30 min

This tutorial will help you get started with App Inventor + IoT and an LED (light emitting diode ... basically a small light) attached to an <u>Arduno 101</u> controller. We are also using a <u>Seeed Grove</u> shield for this tutorial. You do not need to use this board, but it does make things easier. The LED control we recommend is the <u>Grove White LED</u>. (They come in different colors).

Before you start, please complete the <u>App Inventor + IoT Setup tutorial</u> to set up your Arduino device.

- Connect the LED to the Grove board in the D5 pin connector.
- For this tutorial make sure LED is set to ENABLED and all others are set to DISABLED.
- You should also click the arrow button in the top left to upload the code.

		AIM-for-Things-Arduino101 Arduino 1.8.2		
			ø	
	AIM-for-Things-Arduino1	101 § Accelerometer.hh Button.hh Camera.hh	Console.hh 🔻 🛛 Fi	
	#define NAME #define DEBUGGING	"App Inventor" // no more than 11 characters DISABLED	1	
	<pre>#define ACCELEROMETER #define BUTTON #define CAMERA #define CONSOLE #define FUNGERPRINT #define CVPOSCOPE</pre>	DISABLED DISABLED DISABLED DISABLED DISABLED		
	<pre>#define LED #define LIGHI_SENSOR #define PINS #define PROXIMITY #define PROXIMITY #define RGBLCD #define SERVO #define SOUND_RECORDER #define TEMPERATURE // frequency to read set</pre>	ENABLED DISABLED TISABLED DISABLED DISABLED DISABLED DISABLED DISABLED DISABLED DISABLED DISABLED DISABLED		
	<pre>const unsigned long SENSOR_UPDATE_FREQ = 50000; const unsigned long IMU_READ_FREQ = 5000; const double IMU_FILTER_ALPHA = 0.5; //Alpha for accelerometer low pass filter unsigned long nextSensorUpdate; unsigned long nextIMURead; double dt; const uint8_t BITS[8] = { 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80 }; const uint8_t MASK[8] = { 0xFE, 0xFD, 0xFB, 0xF7, 0xEF, 0xDF, 0x8F, 0x7F };</pre>			

Next, you should complete the <u>App Inventor + IoT Basic Connection</u> tutorial to make a basic connection to the Arduino 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:

- Drag a *HorizontalArrangement* from the Layout drawer in the Palette window and place it below **LabelStatus**.
 - Drag two **Buttons** from the User Interface drawer and drag into the *HorizontalArrangement.*
 - Rename the first "ButtonOn" and change its text to "Turn On".
 - Rename the second "ButtonOff" and change its text to "Turn Off".
- Below the *HorizontalArrangement*, drag in a **Slider** from the User Interface drawer of the Palette window.
 - Rename the Slider to "SliderIntensity".
 - Set the Slider's Width to "Fill parent"
 - Set its *MaxValue* to 100 and *MinValue* to 0

P. L. H.			Beer of the
Palette	Viewer	Components	Properties
User Interface	Display hidden components in Viewer	😑 📃 Screen1	SliderIntensity
Eutton 🤊	⊜	HorizontalArrangement1	ColorLeft
CheckBox 🤊	Screen1	ButtonScan	
DatePicker	Scan Stop Scan Connect Disconnect	ButtonStopScan	Gray
Image 🔿	Status:	ButtonConnect	Width
A Label 2	Turn On Turn Off	A LabelStatus	Fill parent
		HorizontalArrangement2	MaxValue
	Data:	ButtonOn	MinValue
		ButtonOff	
🛕 Notifier 💿		SliderIntensity	ThumbEnabled
PasswordTextPox 🤊		A LabelData	
🚺 Slider 💿			ThumbPosition
Spinner 🕐		BluetoothLE1	No-ible
TextBox 🤊		ArdunororLear	
in TimePicker			
WebViewer 🤊			
		Rename Delete	
Layout	1 U	Media	
Media	Non-visible components		
Drawing and Animation	BluetoothLE1 Arduino101Led1	Upload File	

Drag a **Label** from the User Interface drawer in the Palette window and drop it between **SliderIntensity** and **ListBLE**.

- Rename the **Label** "LabelData".
- Change its Text property to "Data: ".



Now 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/resources/edu.mit.appinventor.iot.arduino101.aix
- Add the **Arduino101Led** extension to your app by dragging it onto the Viewer.
- In the Properties tab for the Arduino101Led1
 - Set *BluetoothDevice* to "BluetoothLE1".
 - Check that *Intensity* is set to "100."
 - Set *Pin to* the <u>digital</u> pin that matches the one the LED control is plugged into on the Grove board. (in this case D5)
 - Note: You only need to put the number (5), not the letter "D".

Now switch to the Blocks Editor view

Now we need turn the LED on and off when we press the buttons.

- From the Blocks pane, click on **ButtonON** and drag a **when ButtonOn.Click** block into the Blocks viewer.
- From Arduino101Led1 in the Blocks pane, add call ArduinoLed1.TurnOn.
- From the Blocks pane, click on ButtonOFF and drag a when ButtonOff.Click block into the Blocks viewer.
- From Arduino101Led1 in the Blocks pane, add call ArduinoLed1.TurnOff



Next we need to store the data we receive from the sensor.

- From the Variables drawer in the Blocks pane, drag an initialize global name to block and name it "Intensity".
- From the Math drawer add a number block and set it to "0". We'll use this to keep track of the slider setting for the LED brightness (intensity).

initialize global Intensity to

Let's make a new procedure to display the current LED intensity in the **LabelData**. 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 a set LabelData.Text to block.
 - From the Text drawer, add a **join** block.
 - From the Text drawer drag out a text block and type in **"Intensity: "** and snap to the **join** block.
 - From the Variables drawer, drag out a **get global Intensity** block and snap to the **join** block.



Finally, we want to change the brightness of the LED when we move the slider.

- From SliderIntensity in the Blocks pane, drag out when SliderIntensity.PositionChanged.
 - from the Variables drawer, add set Arduino101Led1.Intensity to.



Your app should now be working! Connect your Arduino device using the MIT AI2 Companion (if you haven't already). Test it out by pressing the "On" and "Off" buttons, and moving the slider left and right. The LED light on your Arduino board should respond.



