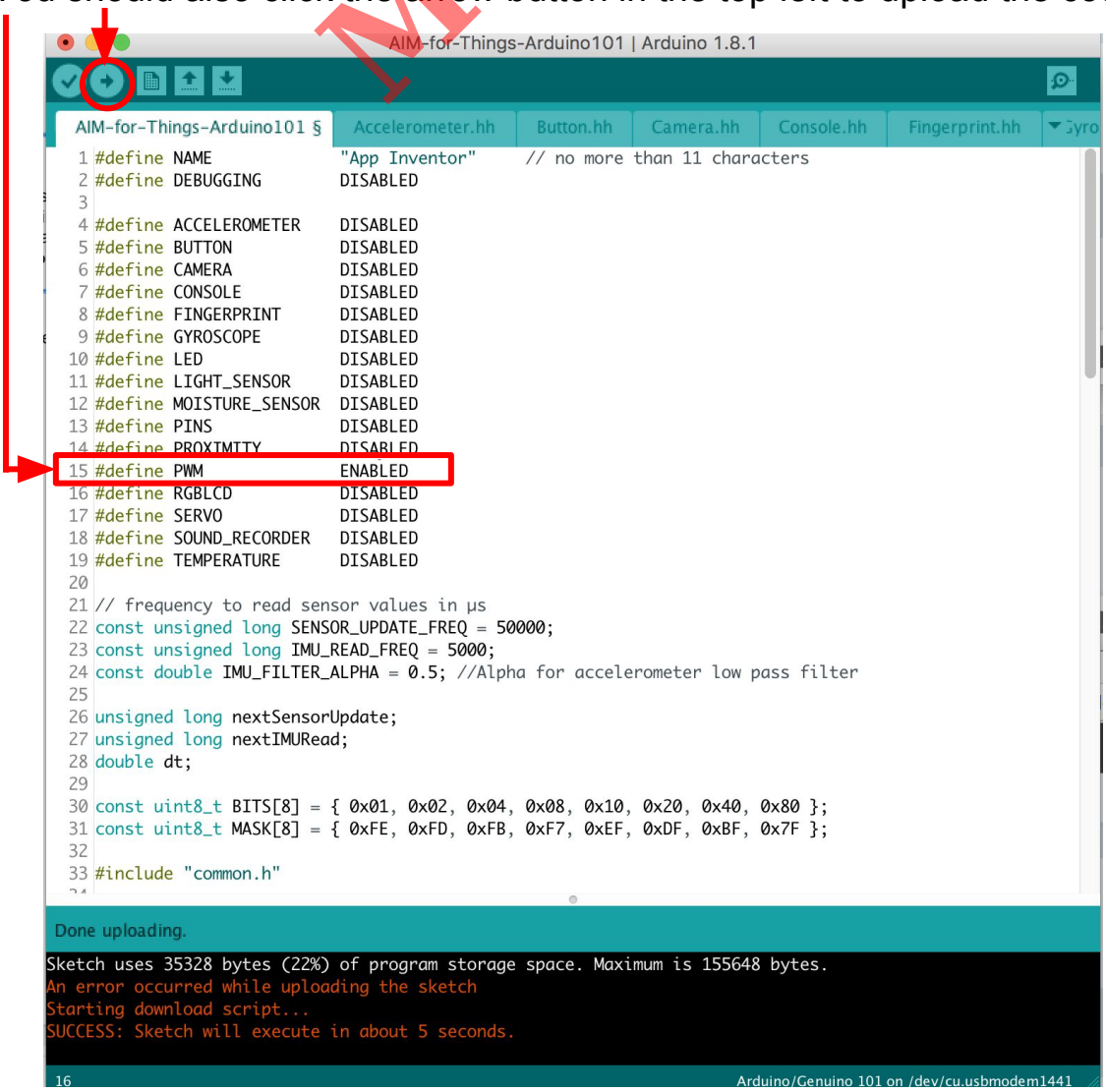
App Inventor + IoT: PWM Motor

This tutorial will help you get started with App Inventor + IoT and a PWM Motor.

Before you start you should have completed the App Inventor + IoT Setup tutorial. This tutorial builds on the App Inventor + IoT Basic Connection tutorial. We suggest you build the basic connection tutorial at least once, but you can also download the completed .aia file to get you started here. We are also using a Seeed Grove board for this tutorial. You do not need to use the board, but it does make things easier.

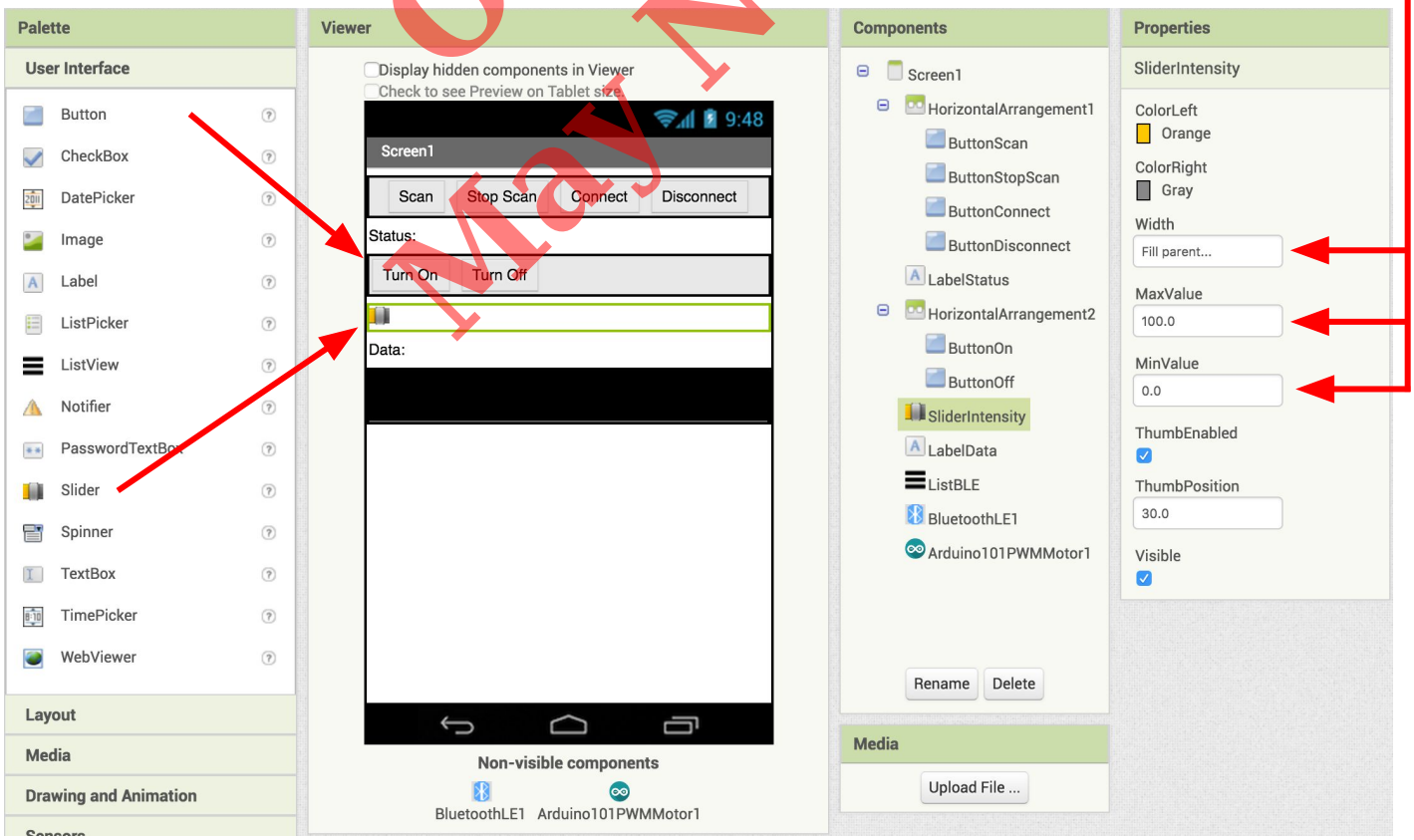
First, we need to make sure we have the correct Arduino code running. Plug in your Arduino and open the AIM-for-Things-Arduino101.ino file (from the Setup tutorial above).

- For this tutorial make sure **PWM** 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



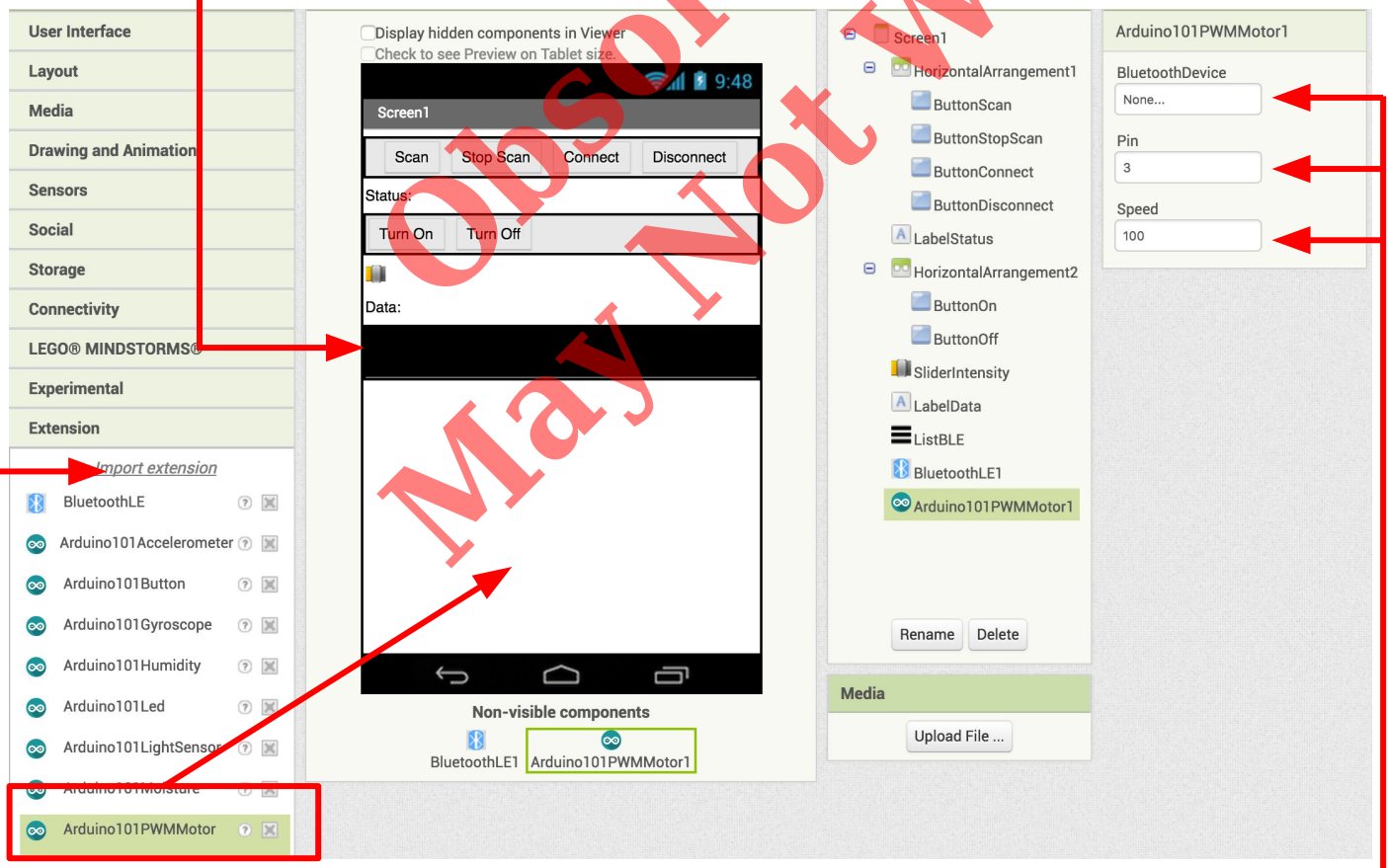
The next steps all build off of the the starter code for the Basic Connection.aia:

- Drag a *Horizontal Alignment* from the Layout Palette below **LabelStatus**
  - Drag two **Buttons** into the *Horizontal Arrangement*
    - Rename the first "ButtonOn" and change its text to "Turn On"
    - Rename the second "ButtonOff" and change its text to "Turn Off"
- Below the *Horizontal Arrangement*, drag in a **Slider** from the Layout Palette
  - Set the *Width* to "Fill parent"
  - Set *MaxValue* to 100 and *MinValue* to 0



Drag a **Label** from the User Interface Palette and drop it between **SliderIntensity** and **ListBLE**

- Rename the **Label** "LabelData"



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 **Arduino101PWMMotor** extension to your app by dragging it onto the Viewer
- In the Properties tab for the **Arduino101PWMMotor1**
  - Set *BluetoothDevice* to "BluetoothLE1"
  - Set *Intensity* to "100" (should already be set)
  - Set the *Pin* to match the digital ("D") one you've plugged the PWM Motor into on the Grove board (in this case D6)
    - *Note: The Motor must be plugged into the Arduino at pin D3, D5, D6, or D9, and you only type the number (6), not the letter "D".*

Now switch to the Blocks Editor view

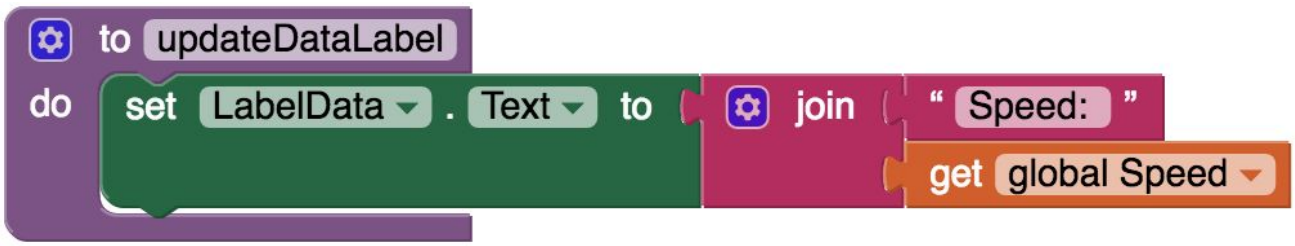
Now we need turn the LED on and off when we press our buttons.

- From the Blocks pane, from **ButtonON** drag a **when ButtonOn.Click** block in and from **Arduino101Led1** add **call ArduinoPWMMotor1.TurnOn**
- From the Blocks pane, from **ButtonOFF** drag a **when ButtonOff.Click** block in and from **Arduino101Led1** add **call ArduinoPWMMotor1.TurnOff**

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 "Speed". 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 Motor Speed.

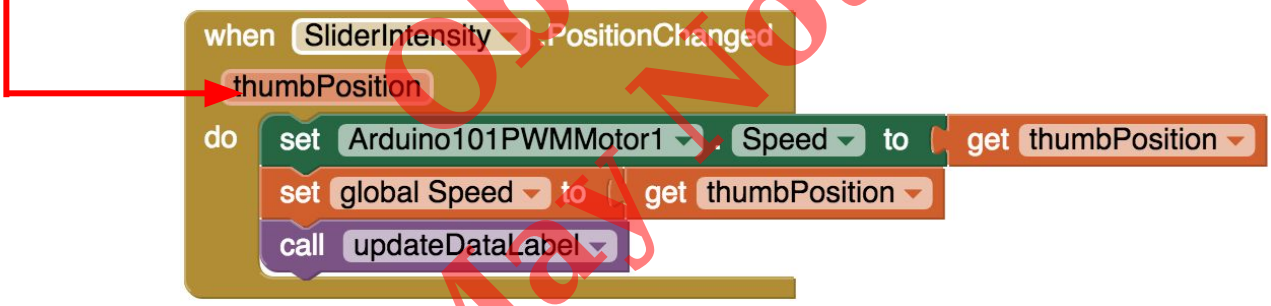
Let's make a new procedure to display the current LED intensity in the **LabelData**. You can create a procedure by dragging out a purple procedure block from the Procedures drawer in the Blocks pane. Let's rename it **updateDataLabel**

- from LabelData add **set LabelData.Text to**
- From the Text pane, connect a **join** block
  - Add a textbox with **"Speed: "**
  - And from Variables add **get global global Speed.**



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

- From **SliderIntensity** drag **when SliderIntensity .PositionChanged**
  - from variables add **set Arduino101PWMMotor1.Speed to**
    - Hover over the orange "thumbPosition" in the **when SliderIntensity .PositionChanged** to see the **get thumbPoistion** block. Connect this block to the **set Arduino101PWMMotor1.Intensity to** block
  - From Variables, drag a **set global speed to** and add another **get thumbPosition** block
  - from procedures add **call updateDataLabel**



Your app should now be working! Test it out by connecting your Arduino device using the companion (if you haven't already). Test it out by turning on the motor and moving the slider to make it turn faster or slower.