App Inventor + IoT: LCD RGB

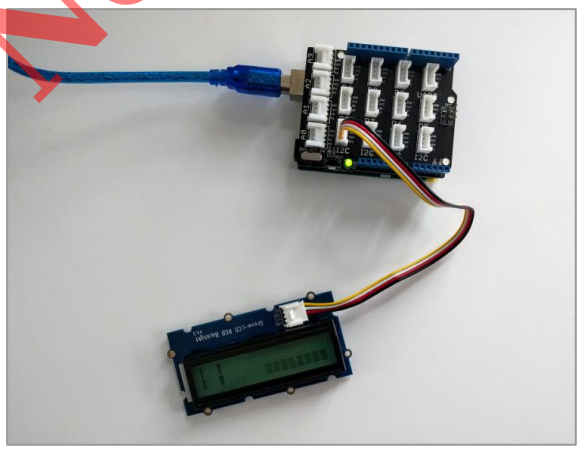
30 min

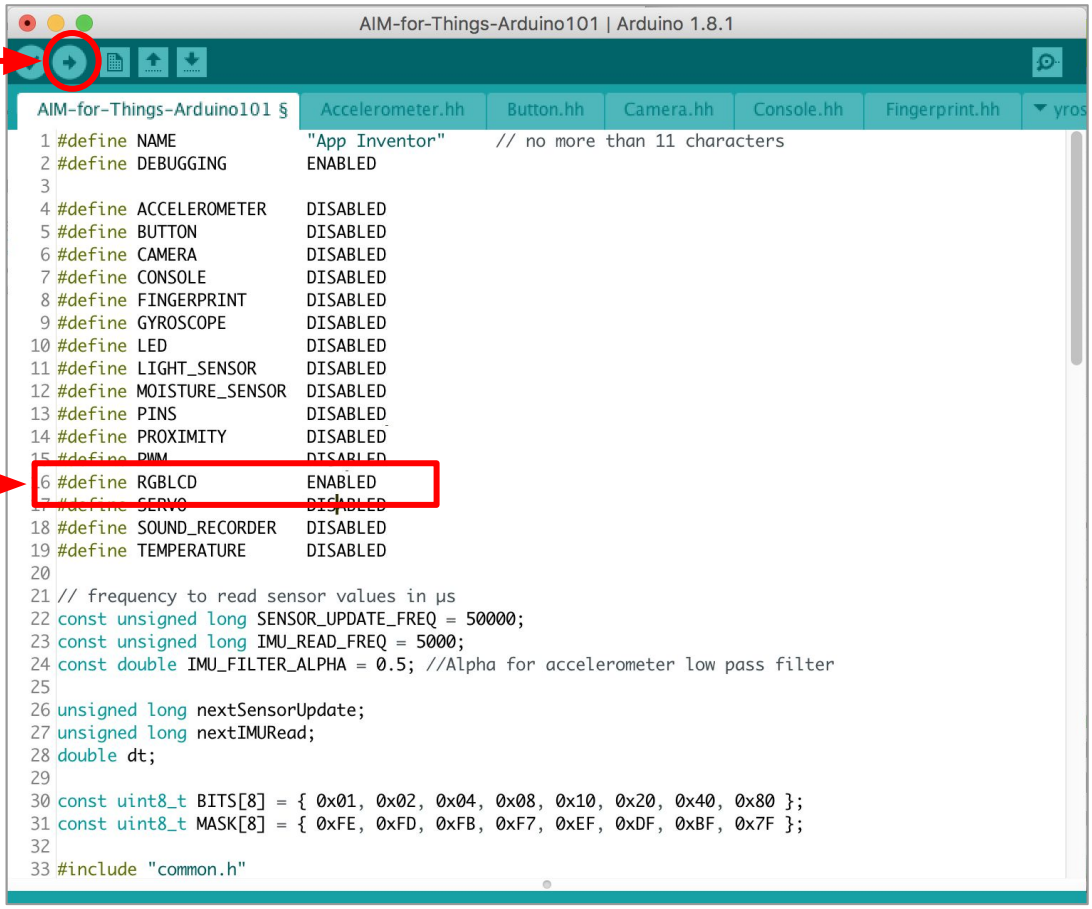
(with IoT Setup and Basic Connection tutorials completed)

This tutorial will help you get started with App Inventor + IoT and a RGB LCD on an Arduino 101 controller. An RGB LCD is a color liquid crystal display, where text can be displayed. We are also using a Seeed Grove shield for this tutorial. You do not need to use this board, but it does make things easier. The LCD we recommend is the Grove LCD RGB Backlight.

Before you start you should first complete the App Inventor + IoT Setup tutorial to set up your Arduino device.

- Connect the LCD RBG to the Grove board in the any I2C pin connector.
- For this tutorial make sure **RGBLCD** 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.

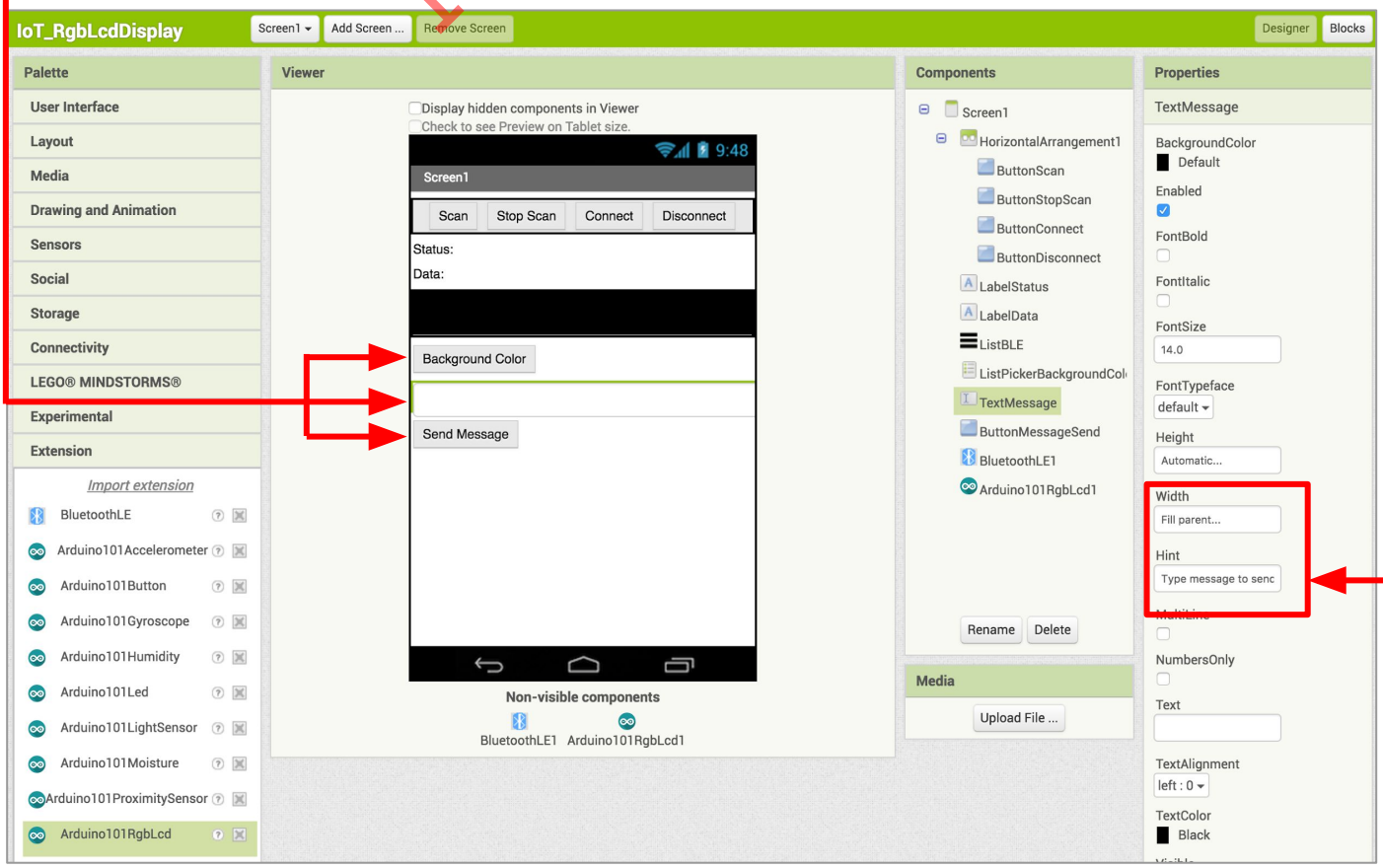




Next, you should complete the App Inventor + IoT Basic Connection tutorial to make a basic connection to the Arduino device. If you prefer, you can download the completed .aia file here.

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

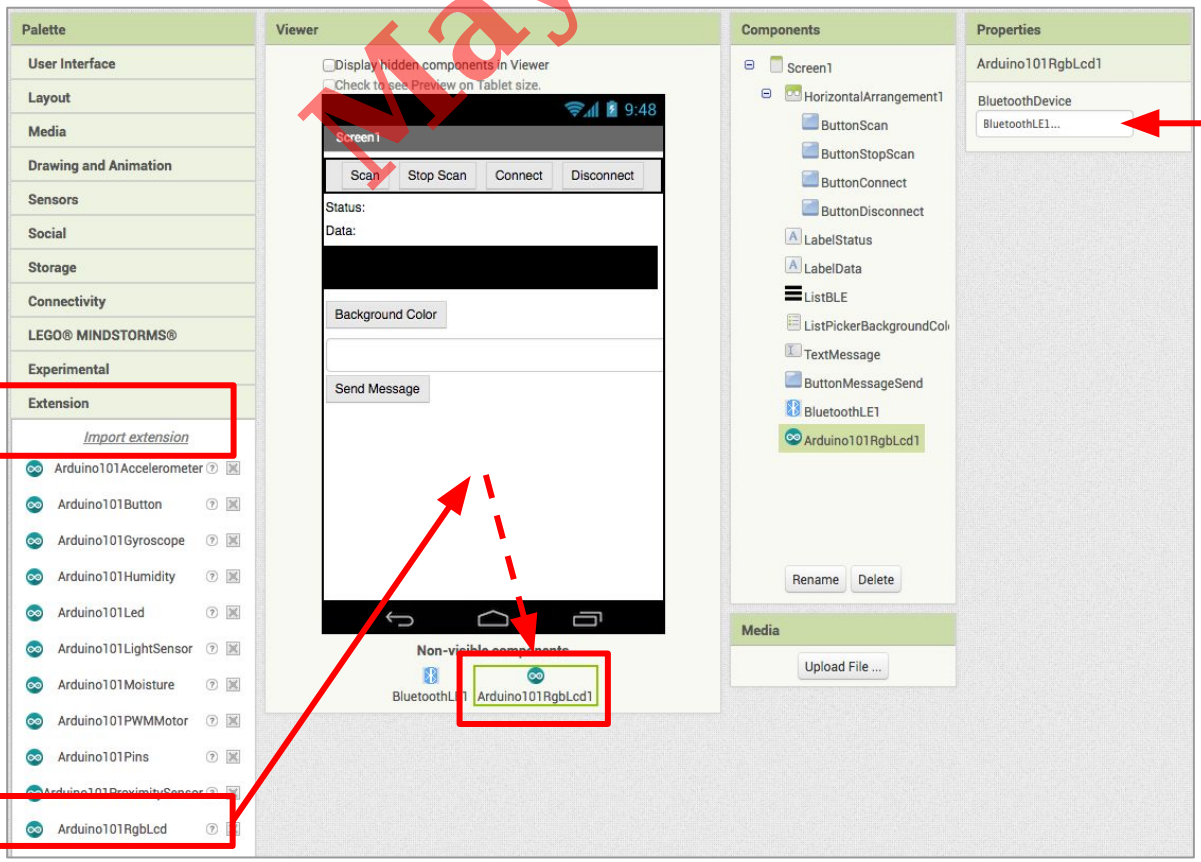
- Drag a **ListPicker**, a **TextBox** and a **Button** from the User Interface Palette and drop them underneath **ListBLE**.
    - Rename the **ListPicker** "ListPickerBackgroundColor", the **TextBox** "TextMessage" and the **Button** "ButtonMessageSend"
- Set the Text of the **ListPicker** to "Background Color"
- Set the width of the **TextBox** to "Fill Parent" and the Hint to "Type message to send".
- Set the Text of the **Button** to "Send Message".



Now let's install the RGB LCD extension to our app.

- 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 **Arduino101RgbLcd** extension to your app by dragging it onto the Viewer.
- Click on **Arduiono101RgbLcd** in the Components pane.
- In the Properties pane, click on **BlueToothDevice** and select “BluetoothLE1”.

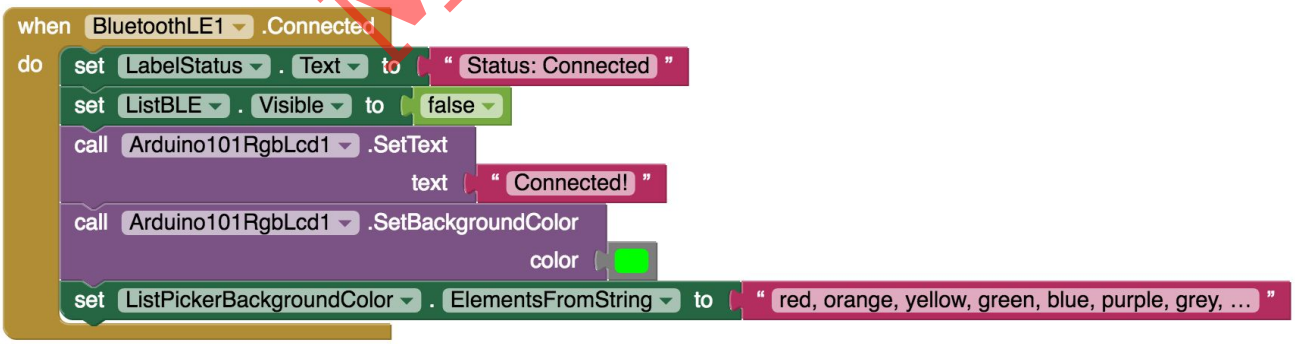
Unlike many other components, RgbLcd doesn’t need to define a pin in the designer. You just need to plug the RgbLcd component into any of the I2C slots on the Grove.



Now switch to the Blocks Editor view

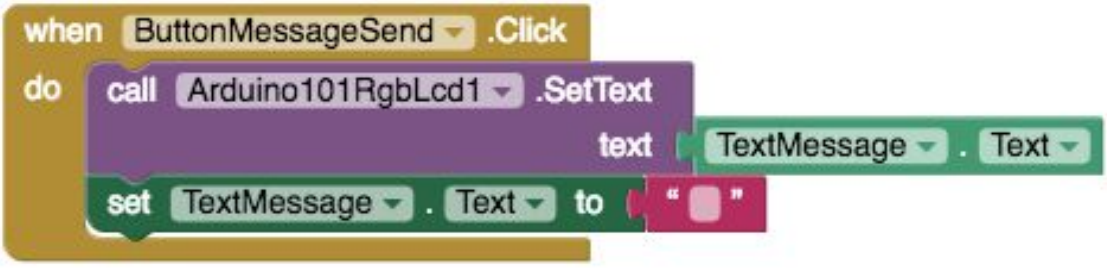
Next we want to set it up so that we set the text and background color when we first connect to the Arduino.

- Find the existing **when BluetoothLE1.Connected** block you made in the Basic Connection tutorial.
- from the Arduino101RgbLcd1 drawer in the Blocks pane, add **call Arduino101RgbLcd1.setText.**
  - from the Text drawer, add a text block and type “Connected!”.
- from the Arduino101RgbLcd1 drawer, add a **call Arduino101RgbLcd1.SetBackgroundColor.**
  - From the Color drawer, add a color block (we used green below).
- From the ListPickerBackgroundColor drawer in the Blocks pane, drag out a **set ListPickerBackgroundColor.ElementsFromString.**
  - From the Text drawer, add a text block and type “red, orange, yellow, green, blue, purple, grey, white”. This will give you the different options for the **ListPicker**.



Then, we want to update the LCD's text with what the user has typed into the textbox when they click the "Send Message" button.

- from the ButtonMessageSend drawer in the Blocks pane, drag out a **when ButtonMessageSend.Click**.
- from the Arduino10RgbLcd1 drawer, drag out a **call Arduino101RgbLcd1.SetText.**
  - from TextMessage in the Blocks Pane, add **TextMessage.Text**.
- From TextMessage in the Blocks Pane, add **set TextMessage.Text to.**
  - From the Text draw, add an empty text block.
    This will clear the TextBox after you send the message.



Finally, we want to be able to change the the background color of the RGB LCD. We'll use the ListPicker, "Background Color", to initiate this change.

- From the ListPickerBackgroundColor drawer in the Blocks pane, add **when ListPickerBackgroundColor.AfterPicking.**
- From the Control drawer, add an **if-then** block.
- Click on the blue gear iconto get a popup window. Drag 7 **else if** blocks into the **if-then** block to make a very large **if-then-else-if** block.

- Drag a **= (equals)** block from the Logic drawer and add to the first **if.**
  - from the ListPickerBackgroundColor drawer, add **ListPickerBackgroundColor.Selection** and snap into the left side of the **=** block.
  - from the Text drawer, drag a text block, type in "red" and snap that block into the right side of the **=** block.
- From the Arduino101RgbLcd1 drawer, add **call Arduino101RgbLcd1.SetBackgroundColor** to the **then** part of the **if** block.
- From the Color drawer, add a **red** block and snap in.

- In each of the remaining else if parts of the **if-else-if** block, repeat what you have done for the color red. Just change the name of the color and the corresponding color block from the Color drawer.

Your final code should look like this:

Your app should now be working! Test it out by connecting your Arduino device using the companion (if you haven't already) and sending some text or changing the LCD color.