

App Inventor + IoT: Wi-Fi RGB LED control with LinkIt 7697



(including Arduino
IDE setup)

Level: advanced

This tutorial will help you get started with App Inventor + IoT, controlling LinkIt 7697 (Arduino compatible) to control a RGB LED with four different colors by buttons on your app.

- [source .ino](#) / [source .aia](#)

Hardware

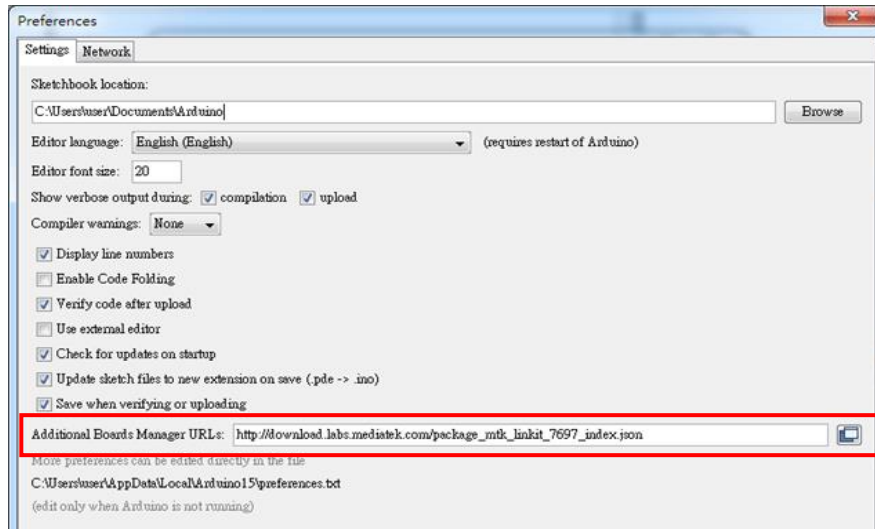
[LinkIt 7697](#) is an Arduino compatible dev board with Wi-Fi / BLE. You can use it to interface with App Inventor through its BLE communication. Here are the components we need to build this project:

- LinkIt 7697 dev board, 1
- breadboard, 1
- wires, 4
- RGB LED (common cathode), 1



Arduino IDE Setup

1. First get [Arduino IDE 1.8.x](#) version, download the .zip file, unzip and click arduino.exe to open the IDE. From **File/ Preference** menu, enter the link below in the Additional Boards Manager URLs field:
 - http://download.labs.mediatek.com/package_mtk_linkit_7697_index.json

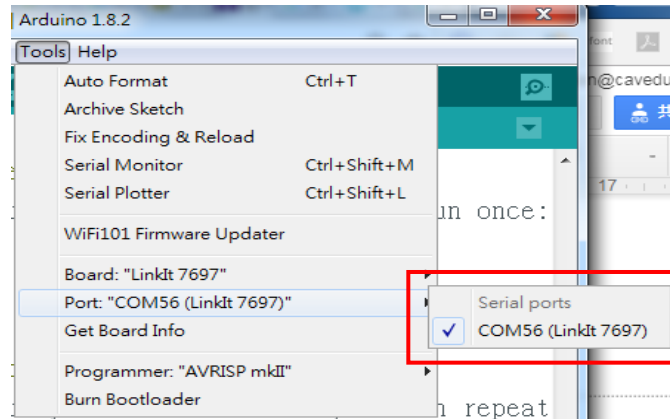


2. Open **Tools/ Board/ Board Manager**, then search “7697” and install the latest version of 7697 SDK.



3. Download and install the **CP2102N driver** ([Windows](#) / [MAC/OSX](#)), then check the COM port in your Device manager. You should see a “**Silicon Labs CP210 USB to UART Bridge(COMXX)**,” this is the COM port number of your LinkIt 7697.

Finally, go back to the Arduino IDE, check to see if it recognizes your LinkIt 7697 successfully in the **Tools/Port** menu. For MAC users, it should be something like **“/dev/tty.usbserialXXX...”** For Windows users, it should look like the figure below.



Hardware Assembly

In this project, we are going to control a RGB LED by App Inventor, which actually are three different color LEDs (red, green and blue) integrated into one.

Here are the components you need for this project:

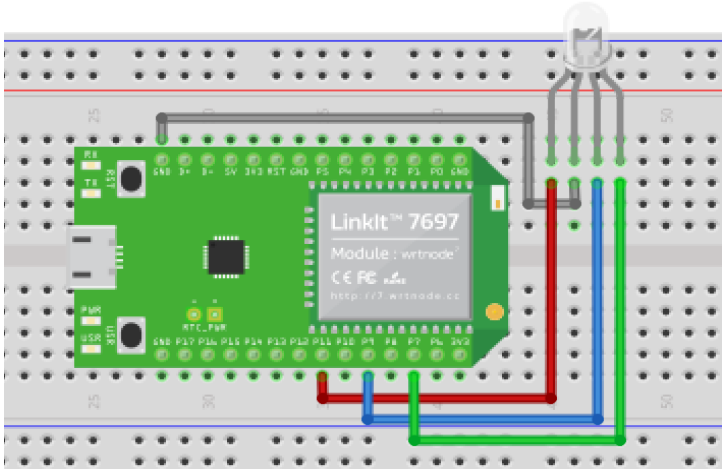
- LinkIt 7697 dev board, 1
- *breadboard, 1
- *wires, 4
- *RGB LED (common cathode), 1

First, connect LinkIt 7697 and RGB LED on the breadboard, and use wires to connect all the components according the table.

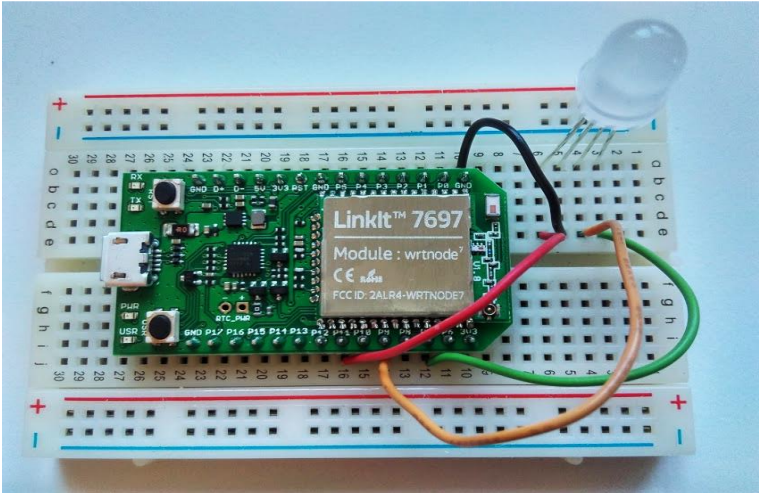
RGB LED	LinkIt 7697
R (red wire)	P11
G (green wire)	P9

B (blue wire)	P7
GND (grey wire)	any one GND pin

Your finished project should look like the picture below:



Physical photo of this project:



App Inventor

The purpose of this project is to control an RGB LED on your LinkIt 7697 board to shine in four different colors with App Inventor via Wi-Fi. The four colors are red, green, blue and white. The main idea is that App Inventor will open LinkIt 7697's URL with different command, this will light up the LED in different colors.

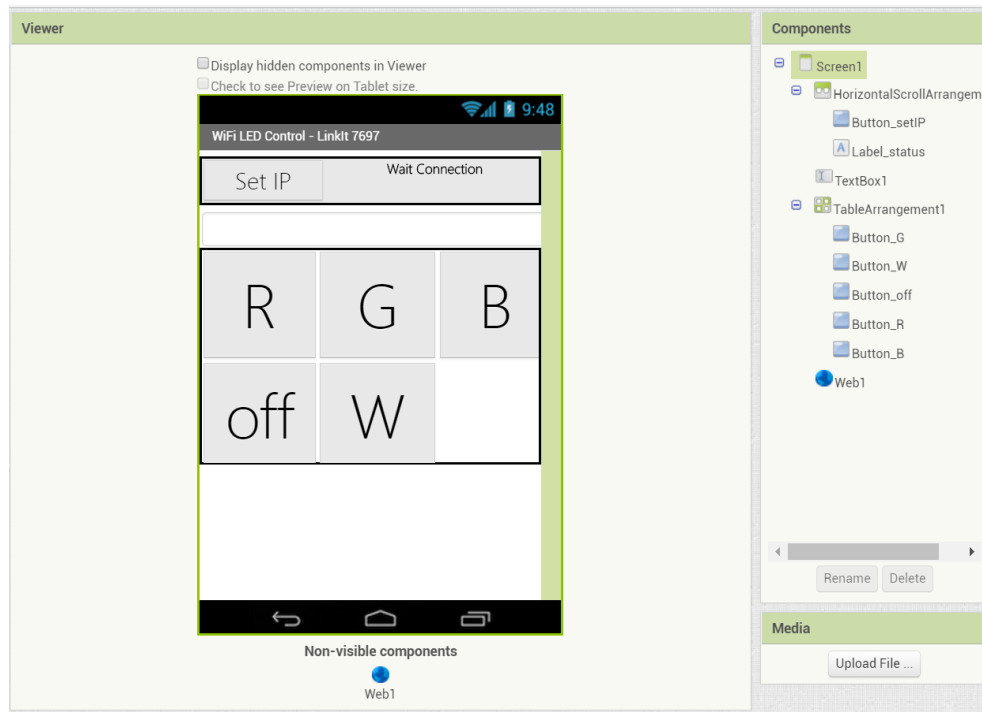
- /R: red light
- /G: green light
- /B: blue light
- /W: white light
- /o: light off

Now log in to your [MIT App Inventor](#) account and create a new project.

Designer

1. Add a **HorizontalArrangement** component.
2. Add a **button** and a **label** into previous **HorizontalArrangement** component. This button is for user to set LinkIt 7697's IP and label is to show messages.
3. Add a **Textbox** and clear the **Text** field. Set the **Hint** to "**check 7697 ip in serial monitor,**" input the user's LinkIt 7697's IP here.
4. Add a **TableArrangement** component, set its width to **Fill parent**, height to **200** pixels, Row to **2** and Column to **3**.
5. Add five **buttons** into previous **TableArrangement** component. Give each a width of 33% and a height of 100 pixels. And modify the Text to "**R**", "**G**", "**B**", "**W**" and "**off**", representing different color of RGB LED.
6. Add a **Web** component for sending text to LinkIt 7697 through Wi-Fi.

After some adjusting, your designer should look like this. It doesn't have to be exactly the same, feel free to modify the color and placement of buttons:



Blocks

Let's take a look of our blocks step by step:

1. Variable for Bluetooth address

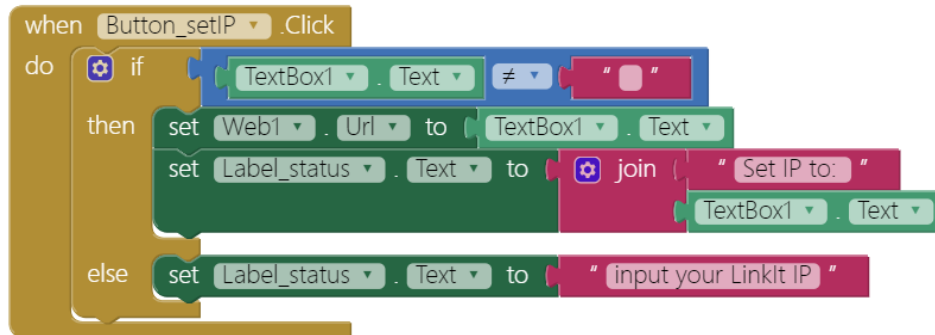
Please replace the value with what you get from Arduino's Serial Monitor. This is LinkIt 7697's IP address after connection to the Internet.

initialize global ip to " http://192.168.1.73 "

2. Setting the IP

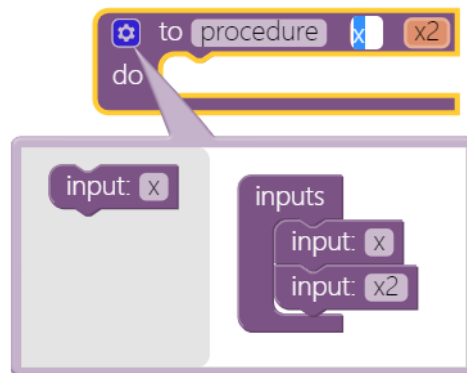
Button_setIP is used to update LinkIt 7697' IP. This is very convenient when you want to control more than one LinkIt board. When you click the button (**Button_setIP.Click** event), it will check whether the TextBox is empty. If so, then it will set the Textbox's

content to the Web component's Url (LinkIt 7697's IP address). If not, it will show a message that tells the user to input LinkIt IP in the form of "XXX.XXX.XXX.XXX", such as "192.168.1.73".



3. Procedure to update Url and connect

Here we use a procedure to manage what we send and message update when we click each button. Please add a procedure and click the blue gear to add two parameters. Rename this procedure to "**sendData**", and two parameters as "**message**" and "**command**", which means the **message** parameter **is used to** update the label and **command** parameter is used to send different command to LinkIt 7697.



The **command** parameter of the sendData procedure will be combined with the IP variable into a complete URL, like "**http://192.168.1.73/R.**" The last letter is a different command sent to LinkIt 7697. Finally we use the Web component to get this URL, by sending out a specific text to LinkIt 7697.

```

to sendData message command
do
  set Web1 . Url to join get global ip get command
  set Label_status . Text to join Web1 . Url " : " get message
  call Web1 . Get

```

4. Buttons to control LED

Every time we click a button, we are calling **sendData** procedures with different parameters. Using procedures will make your code easier to read and manage.

- /R: red light
- /G: green light
- /B: blue light
- /W: white light
- /o: light off

```

when Button_R .Click
do
  call sendData
  message " Red "
  command "/R "

```

```

when Button_W .Click
do
  call sendData
  message " White "
  command "/W "

```

```

when Button_G .Click
do
  call sendData
  message " Green "
  command "/G "

```

```

when Button_off .Click
do
  call sendData
  message " light off "
  command "/o "

```

```

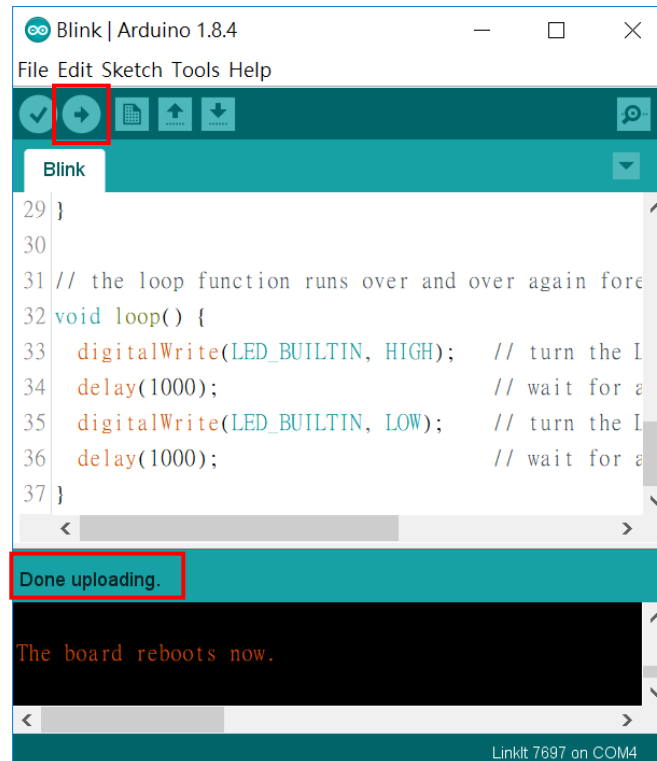
when Button_B .Click
do
  call sendData
  message " Blue "
  command "/B "

```


Arduino code

Please download code [here](#) and upload to your LinkIt 7697.

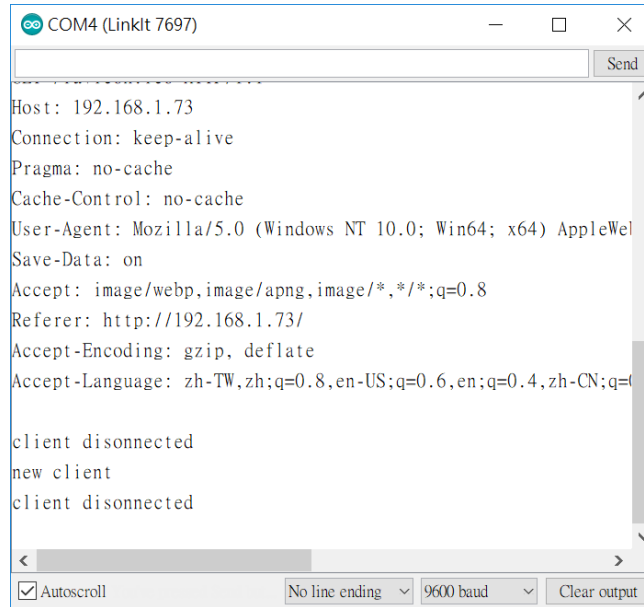
Press the "**Upload**" right-arrow button, this will compile and upload the Arduino sketch in Arduino IDE to your LinkIt 7697. Please make sure you see the "**done uploading**" message in the console below.



Tips

Make sure your LinkIt 7697 is running correctly as a Webserver. Open Serial Monitor in Arduino IDE, check the IP address of your LinkIt 7697.

Modify App Inventor's **IP** variable value with this, then click each button to see the different colors lighting up!



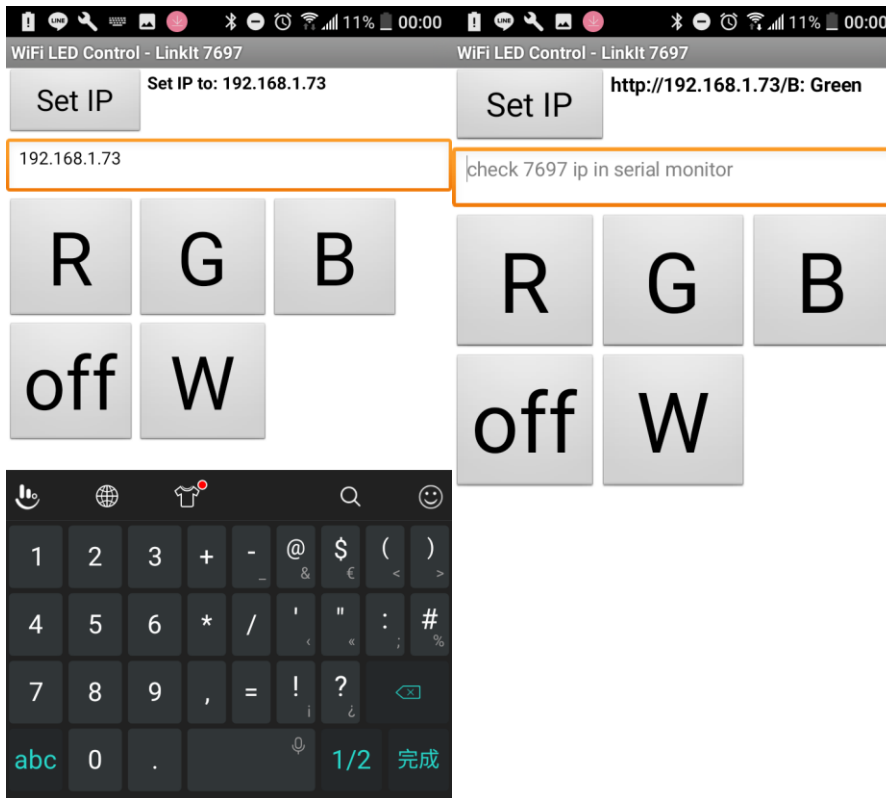
Outside of App Inventor, open a web browser with LinkIt 7697's ip address and you can see a webpage hosted by LinkIt 7697. Click each "here" to control the LED light color. Note: Your PC or any device to open this webpage and LinkIt 7697 must under the same network.



- Click [here](#) to light red
- Click [here](#) to light green
- Click [here](#) to light blue
- Click [here](#) to light white
- Click [here](#) to light off

Tips

Open your app. You can see messages in the lower left corner of the screen when you have set the IP correctly. Click the buttons to light up different colors and the label in the upper right hand corner will show corresponding messages.



Brainstorming

1. Try to light up RGB LED with different colors, such as red and blue lighting up together to shine in purple.
2. Try to control the LED with your voice (hint: add a **SpeechRecognizer** component).