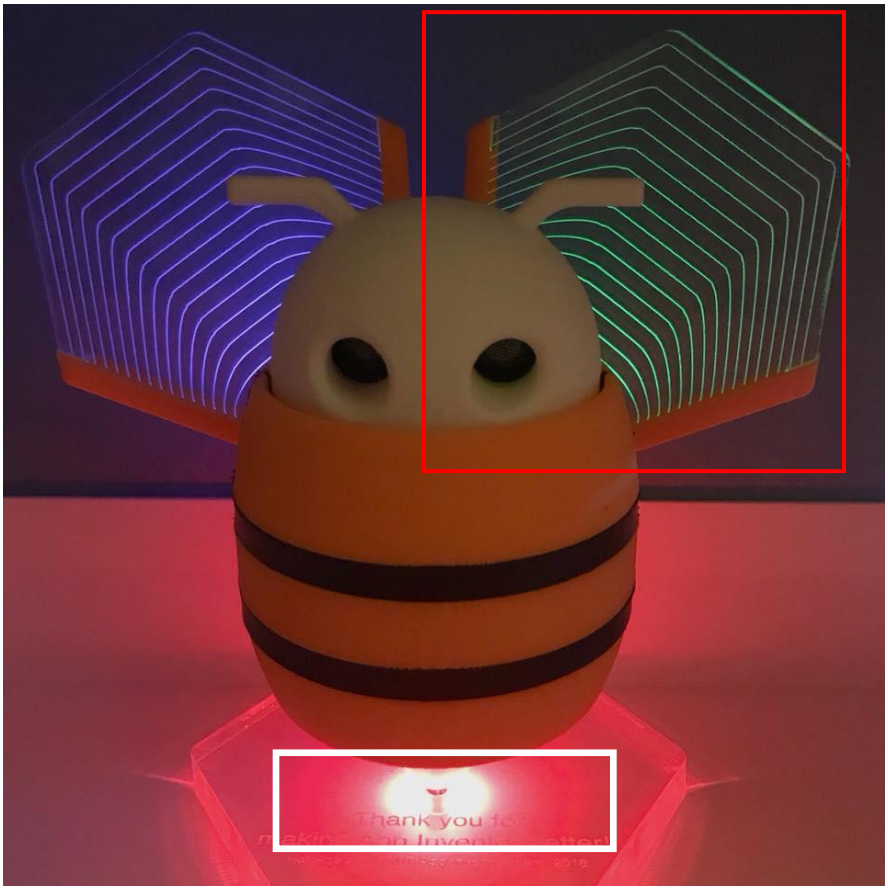
MIT App Inventor Codi Bot: LED control

60 mins

Level: advanced

This tutorial will help you get started with App Inventor + IoT, control Codi Bot LEDs by buttons and a slider. We have also provided a complete app for you to use with any Codi Bot LEDs.

- **source .ino / source .aia**
- **complete .aia**



Function description

This project will show you how to control Codi Bot LEDs with App Inventor through BLE communication, including the red light intensity of RGB LED and green LED on/off. The components used in this tutorial are buttons and a slider. There are many other components in AI2, you can also use them with the same concepts.

Hardware

Please follow this **building guide** to assemble your Codi Bot.

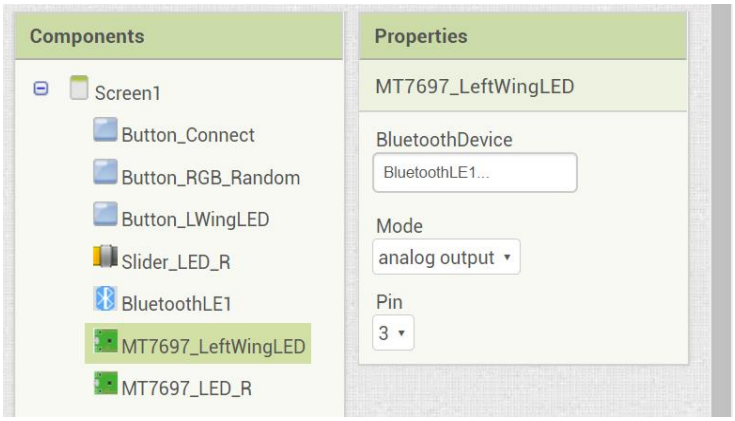
App Inventor

Now log in to your App Inventor account and create a new project, or directly import **this aia file**.

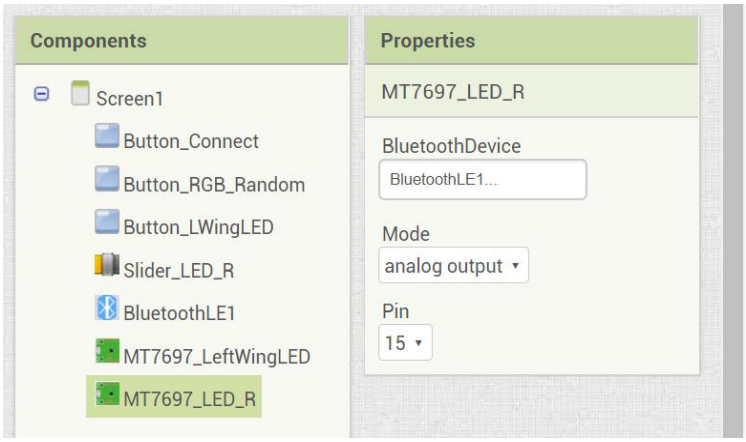
Designer

1. We need to import two extensions from this URL:
    - **Bluetooth low energy**: http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.ble.aix
    - **MT7697pin**: http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.iot.mt7697.aix
2. Add a **BluetoothLE** component to your project. We will use it to send commands to the Codi Bot through Bluetooth communication.
3. Add two **MT7697pin** components to your project. We will use them to control different pins of LinkIt 7697 - in this case the green LED in the left wing and the red pin of the RGB LED at the base.
    - Rename one **MT7697pin** component as "**MT7697_LeftWingLED**". Set its **BluetoothDevice**

property to BluetoothLE1 (Step 2.), Mode to "**analog output**" and set Pin to **3**. This is because we have connected the red pin of RGB LED to the LinkIt 7697 #3 pin.



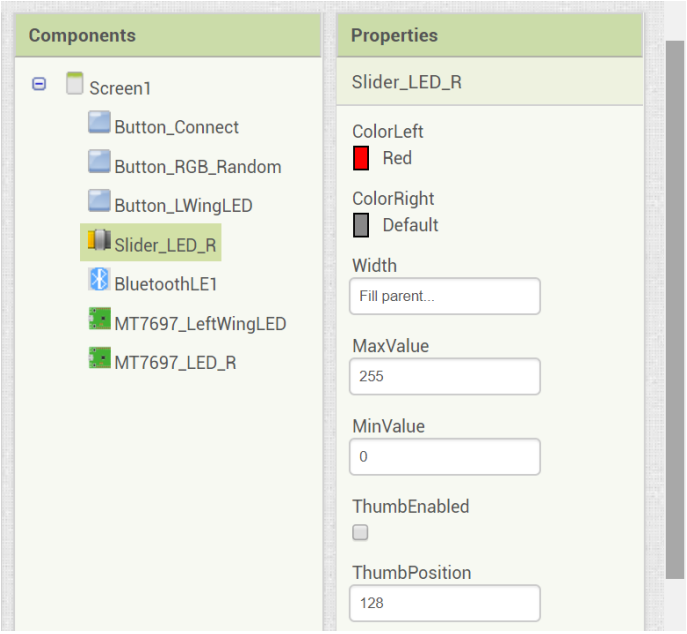
- Rename the other **MT7697pin** component as "**MT7697_LED_R**". Set the BluetoothDevice property to **BluetoothLE1** (Step 2.), Mode to “**analog output**” and the Pin to **15**. This is because we have connected the red pin of RGB LED to the LinkIt 7697 #15 pin.



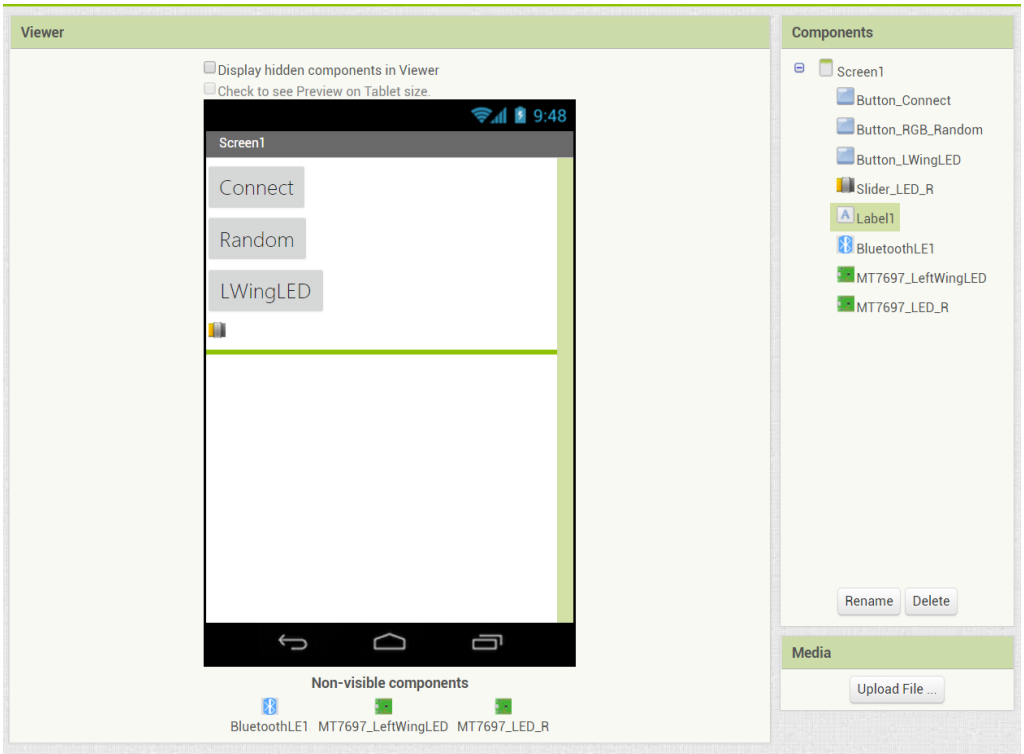
4. Add a button to establish Bluetooth connection between your Android phone and LinkIt 7697. Rename it to "**Button_Connect**" and set Text to "**Connect**".
5. Add another two buttons to control the LEDs. Rename them to "**Button_RGB_Random**" and “**Button_LWingLED**”. Set their Text to “**Random**” and “**LWingLED**” accordingly.
6. Add a slider to control the red light intensity of RGB LED.

Rename it to "**Slider_LED_R**". Set its MaxValue to **255**, MinValue to **0** and ThumbPosition to **128**.

7. Add a label to show red light intensity (0 - 255). Set its Text to " "" (no text).



After some adjusting, your designer should look similar to the image below. It doesn’t have to be exactly the same. Feel free to modify the component’s background color, position and text size.



Blocks

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

1. **Variable for Bluetooth address**

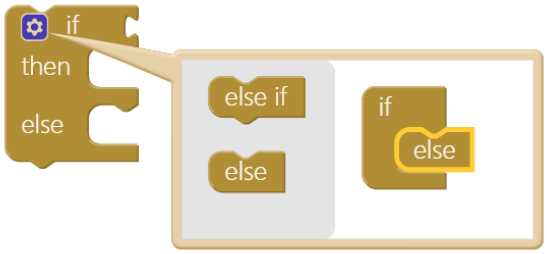
Please replace the **addr** variable value with what you get from Arduino Serial Monitor. This is the Bluetooth address of LinkIt 7697. We will show you how to check this information in [Arduino IDE and sketch] section.

2. **Initialize app and scan for nearby Bluetooth devices**

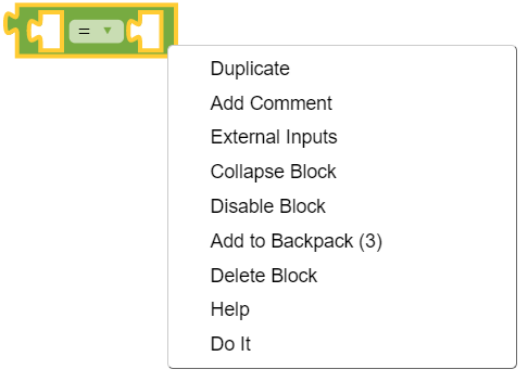
In **Screen1.Initialize** event, we tell the **BluetoothLE** component to scan for BLE devices nearby (**BluetoothLE1.StartScanning**).

3. **Connect and disconnect from Bluetooth device**

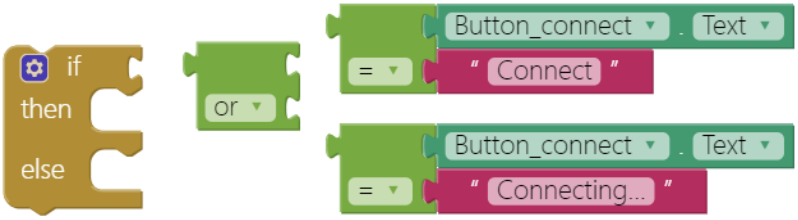
In **Button_Connect.Click** event, we are going to connect to or disconnect from Bluetooth device depending on the button text. First, add an **if** condition, then click its blue gear to add an **else**.



Add an **or** command from logic block, then right-click it and select "**External Inputs**". This will make it into more rectangular shape with input on the right-hand side.



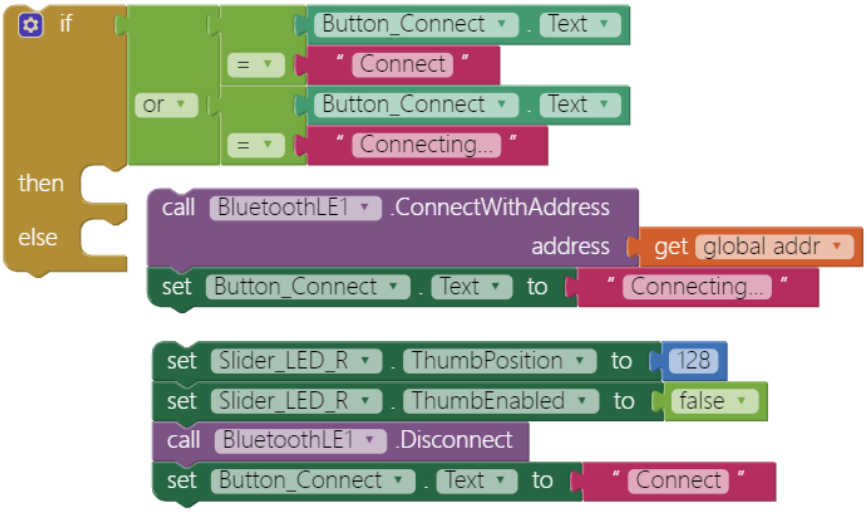
Now we want to check whether the **Button_Connect.Text** status is "**Connect**" OR "**Connecting...**", this is how App Inventor decides to connect or disconnect Bluetooth connection with LinkIt 7697. Please combine these blocks.



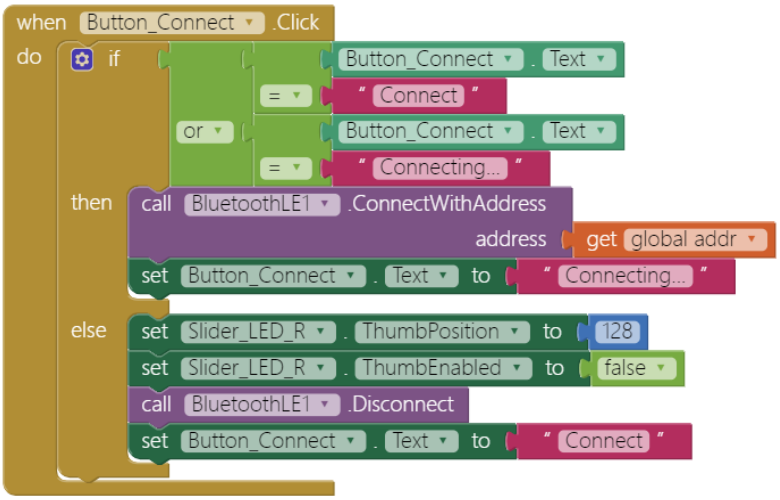
Good! When the **Button_Connect**'s text reads "**Connect**" or "**Connecting...**", the app will connect to the specified Bluetooth device (**BluetoothLE1.ConnectwithAddress**), which is our

LinkIt 7697.

If the text does not read "**Connect**", then set slider to default position/disabled, disconnect (**BluetoothLE1.Disconnect**) and show message on Button_Connect.



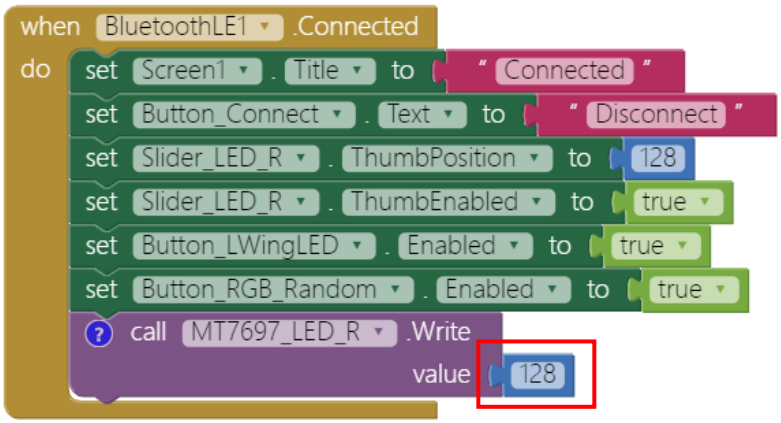
Put everything into **Button_Connect.Click** event, and finish like this:



4. **BLE Connected**

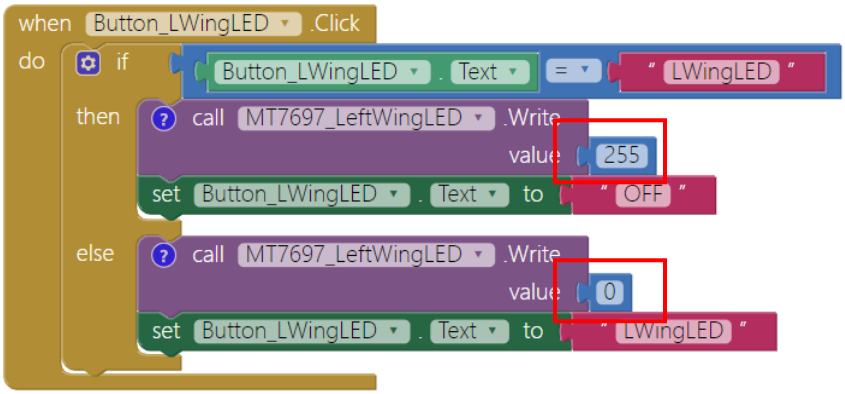
   When connected successfully (**BluetoothLE.Connected** event), we will see related messages on several components. This also enables **Button_LED** to be clicked and Slider_LED_R to middle position and to be dragged. Finally we use

MT7697_LED_R.Write method with value **128** to light on the red light with half of the maximum intensity (255), indicating we've connected to Codi Bot successfully.



5. **Button_LWingLED turn on the green LED**

   We use same button to toggle the green LED on left wing. When **Button_LWingLED** is touched (**Button_LWingLED.Click** event), we turn on green LED (**MT7697_LeftWingLED.Write** with 255) or turn it off (with 0) according the button text.



6. **Button_RGB_Random to randomize red light intensity**

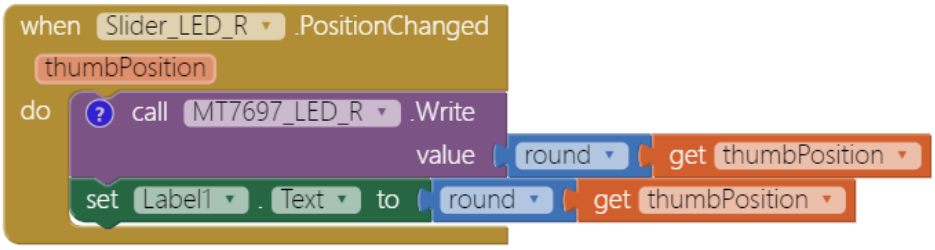
   For a more interesting effect, we use a button to randomize the red light intensity. First create a variable named **lightIntensity** with number **0**.

   When **Button_RGB_Random** is touched (**Button_RGB_Random.Click** event), we will set **lightIntensity**

variable value with **random integer** (0, 255), set the red light intensity, modify label text and move slider thumbposition according to this variable value.

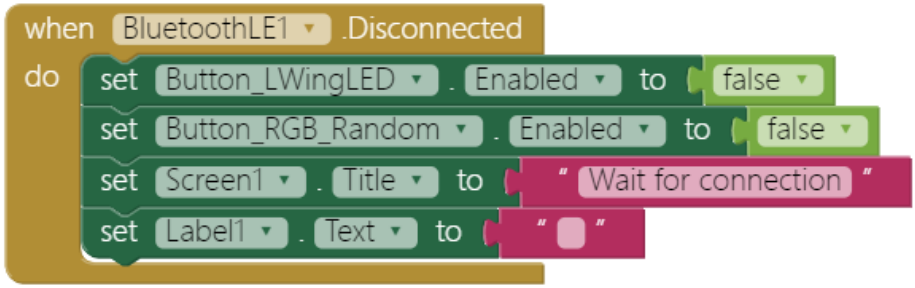
7. **Drag slider to change red light intensity**

   When the slider is dragged (**Slider_LED_R.PositionChanged** event), we set the red light intensity and label text according to the slider thumbpostion.



8. **Disconnect**

   After Bluetooth communication is closed successfully (trigger in Step1), we reset the app to its initial state to wait for next connect request in **BluetoothLE1.disconnected** event.



Arduino IDE and sketch

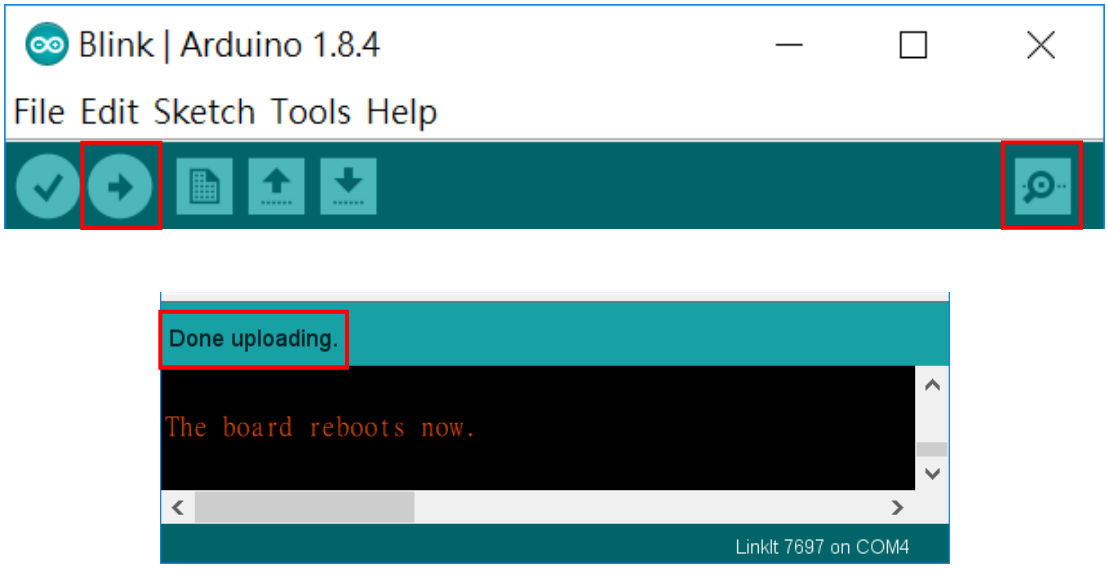
Make sure your computer has Arduino IDE installed and LinkIt 7697 SDK/driver are ready. If not, please check Codi Bot Standalone tutorial.

Connect your computer and LinkIt 7697 with a microUSB cable.

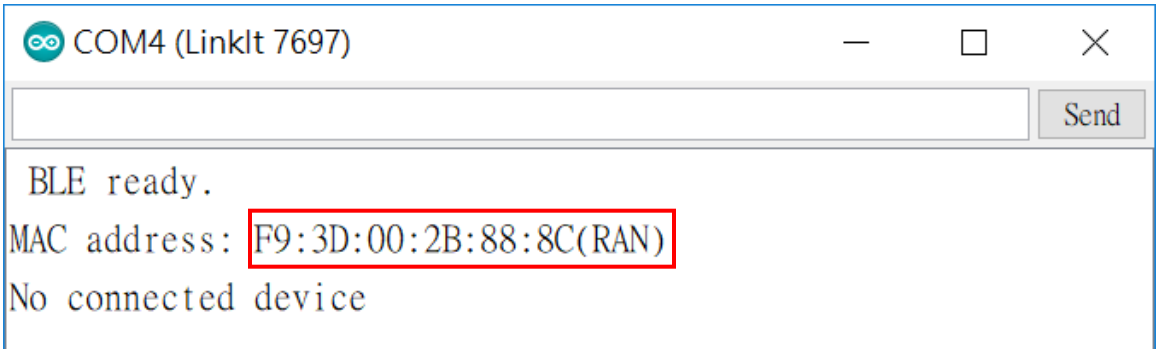
Please download the Arduino sketch from **here** and open in your Arduino IDE. This sketch can be used for all following Codi Bot projects, let you focus on building App Inventor projects.

Press the "**Upload**" right-arrow button of Arduino IDE, this will

compile and upload the Arduino sketch to your LinkIt 7697. Please make sure you see the "**done uploading**" message in the console.



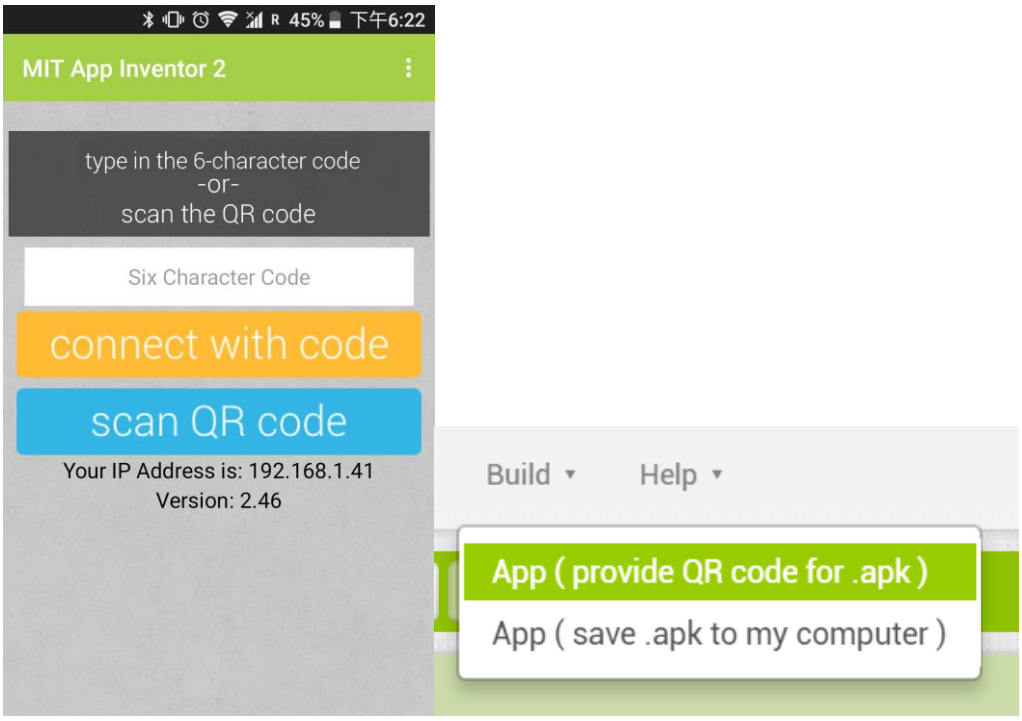
Click the magnifier icon at the up-right corner of Arduino IDE, you should see a message in the pop-up window. The [**XX:XX:XX:XX:XX:XX**] 12-digit string is the Bluetooth address of your LinkIt 7697. We have to modify the **addr** variable value of your AI2 project.



Tips

Make sure your LinkIt 7697 is running correctly. And install App Inventor project on your Android phone by clicking Build / App (provide QR code for .apk), this will show a qrcode for .apk file. Use

MIT AI2 Companion to scan this qrocde, download and install.



Open your app (Fig. 1) and click **Connect** button. You'll see a "**Connected**" message on screen title if your phone is connected with Codi Bot correctly.

Click the **LWingLED** button to see if Codi Bot left wing is lighting green. Also try to drag the slider from left to right, you should see the red light intensity RGB LED at bottom is changing according to the slider.

Remember to click **Disconnect** button when you finish with this project.

Fig 1. Initial screen	Fig 2. Press **LWingLED** button to turn on green LED on the left wing (see its text had changed to **OFF**).

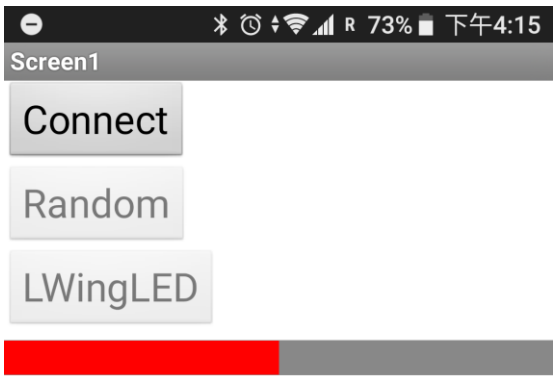
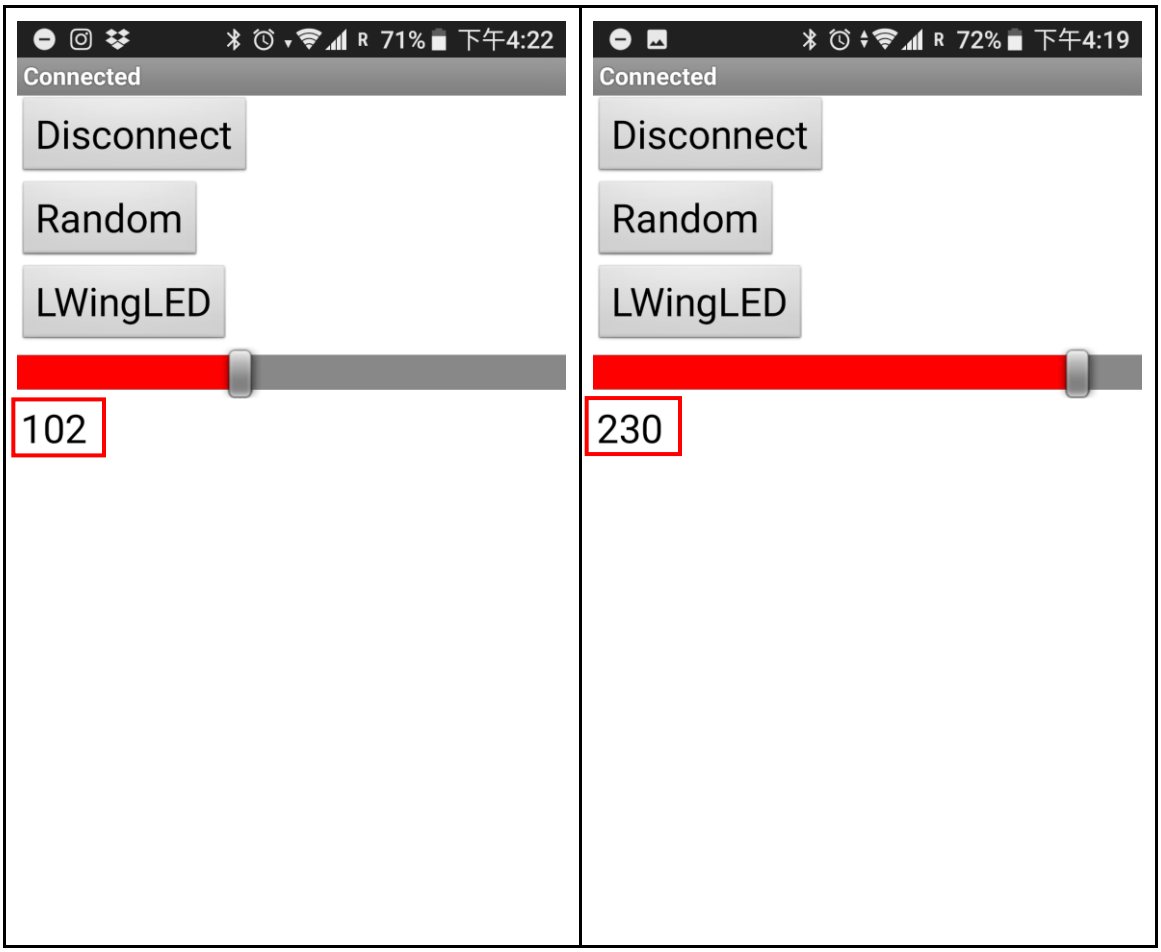
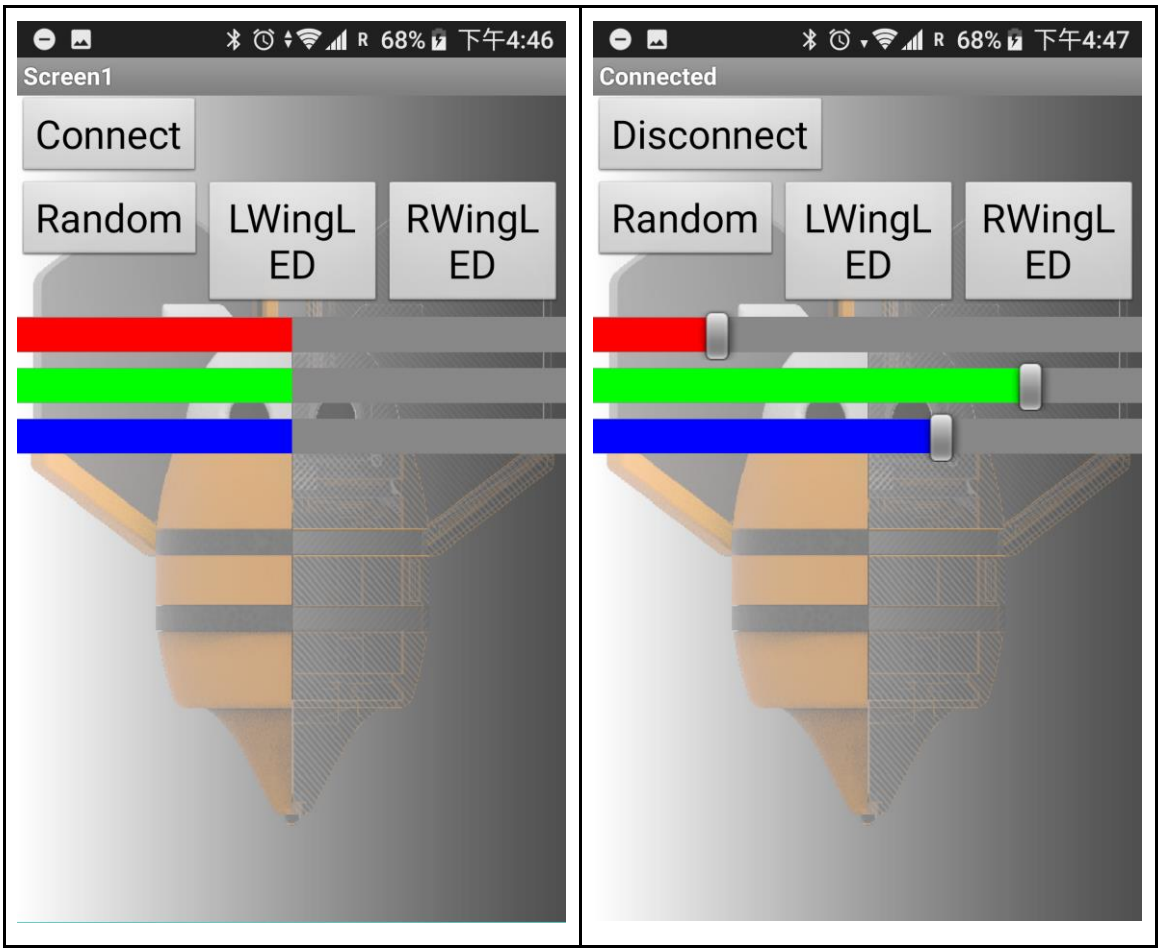


Fig 3. Drag slider to make a moderate light intensity (**102**)	Fig 4. Drag slider to make LED brighter (**230**)



Complete LED control app

We have provided a complete app to control all Codi Bot LEDs, please import this **complete .aia** to your App Inventor.


68% 下午4:46
Screen1
Connect
Random
LWingL
ED
RWingL
ED
68% 下午4:47
Connected
Disconnect
Random
LWingL
ED
RWingL
ED

Brainstorming

1. Use App Inventor **SpeechRecodnizer** component to turn on/off the green LED.
2. Replace slider with App Inventor **OrientationSensor** component to control the red light intensity.