

App Inventor + IoT: LED Blink with LinkIt 7697 (Bluetooth version)



(including Arduino IDE setup)

Level: advanced

This tutorial will help you get started with App Inventor + IoT, using LED of LinkIt 7697 (Arduino compatible) board and Google Speech recognition.

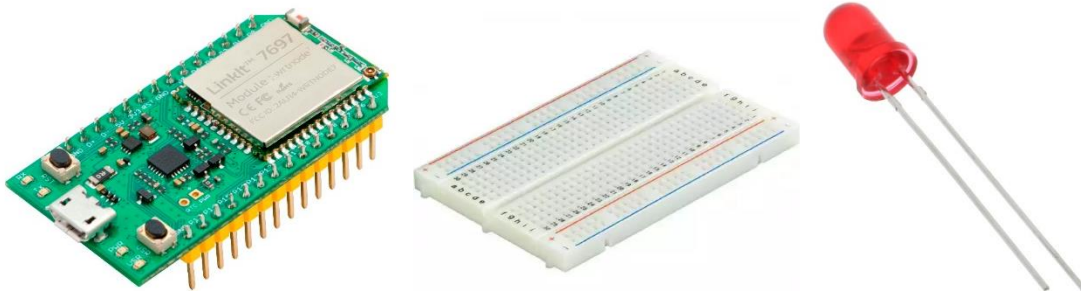
- [source.ino](#) / [source.aia](#)

Hardware

[LinkIt 7697](#) is an Arduino compatible dev board with Wi-Fi / BLE. You can use it to interface with App Inventor through its Bluetooth communication.

In this project, we are going to control this LED with App Inventor. Here are the components we need to build this project:

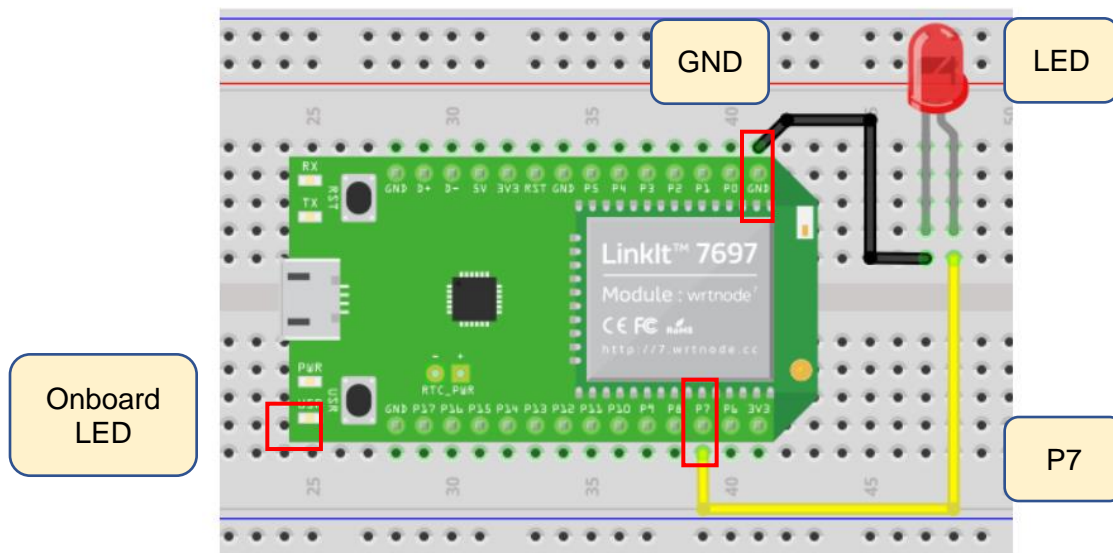
- LinkIt 7697 dev board, 1
- *breadboard, 1
- *wires, 2
- *5mm LED (any color), 1



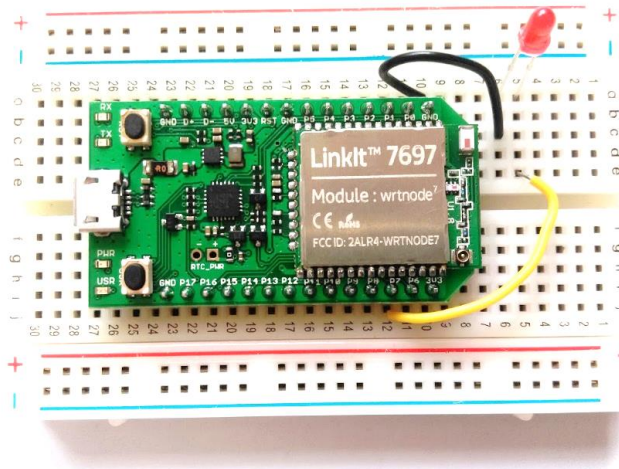
Except for the LinkIt board, other components with * are optional since LinkIt has an onboard LED.

Notice: Almost every Arduino board has an onboard LED connected to D13 in case you are accessible to an external LED and still able to test. However, LinkIt 7697's onboard LED is D7.

Please use one wire to connect the LED's longer pin to the LinkIt 7697's D7 pin. And another wire to connect the LED's shorter pin to LinkIt 7697's GND pin. There are three GND pins on the LinkIt 7697, you can choose any one of them. Our finished circuit should look like this:

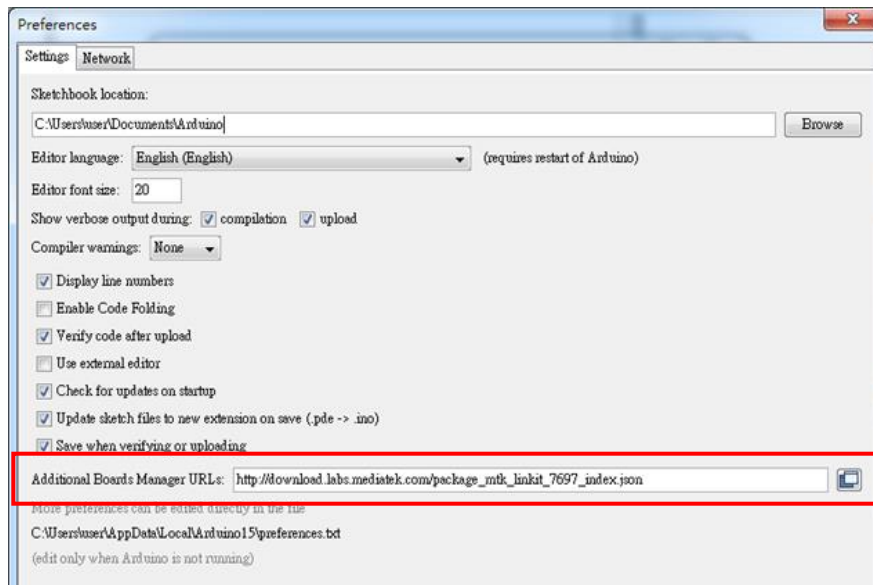


Take a look of the physical photo of our project:



Arduino IDE Setup

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

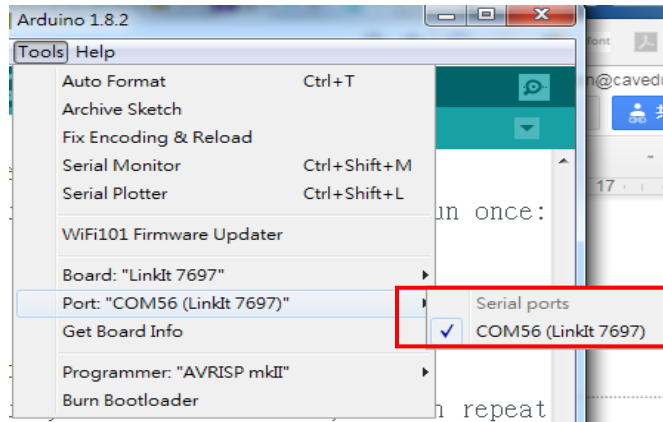


3. Open **Tools/ Board/ Board Manager**, then search for "7697" and install the latest version of 7697 SDK.



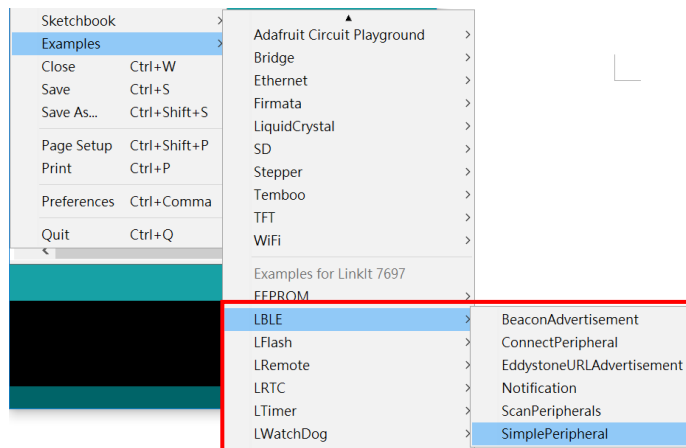
4. Download and install the **CP2102N driver** ([Windows](#) / [MAC/OSX](#)), then check the COM port in your Device manager. Look for "**Silicon Labs CP210 USB to UART Bridge(COMXX)**", this is the COM port number of your LinkIt 7697.

5. For MAC users, it should be something like `"/dev/tty.usbserialXXX..."` and for Windows users, please check the picture below:



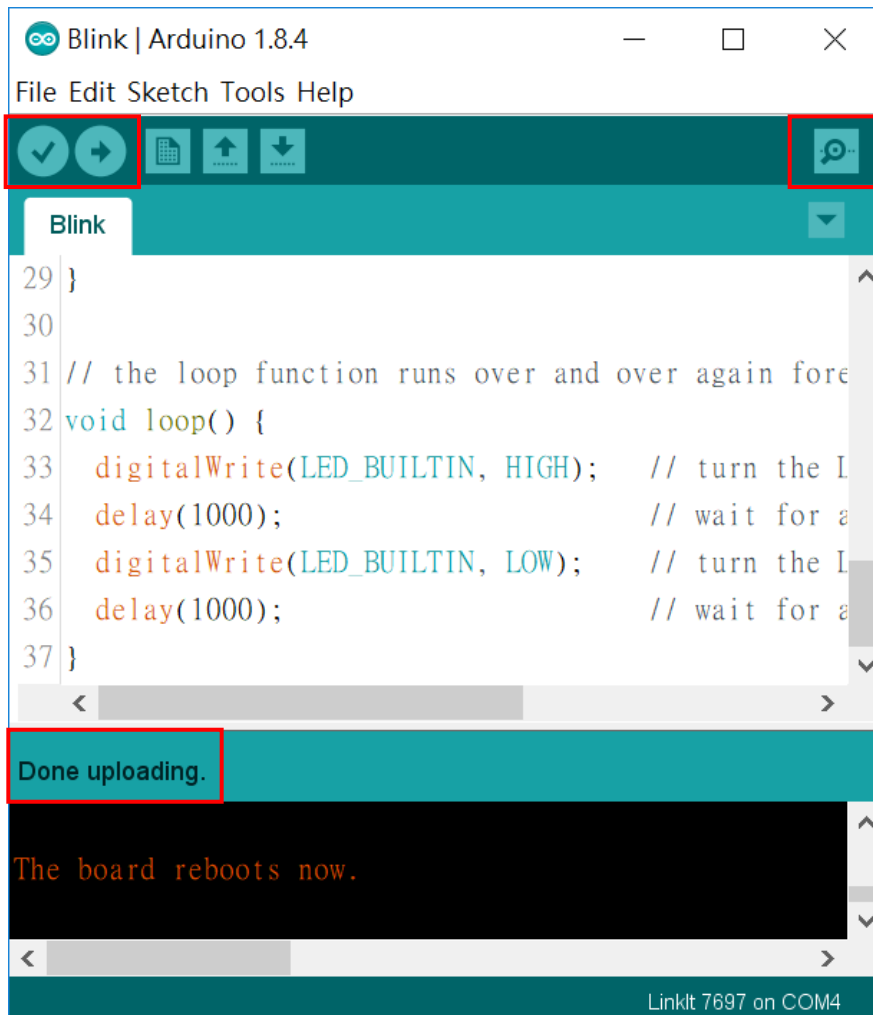
Get the BLE address of LinkIt 7697

1. For safety reasons, not every board is marked with its Bluetooth address on the board (Arduino 101 is an exception). In Arduino IDE, first set the board to "**LinkIt 7697**" then open example "**SimplePeripheral**" from **File/Examples/LBLE** menu.

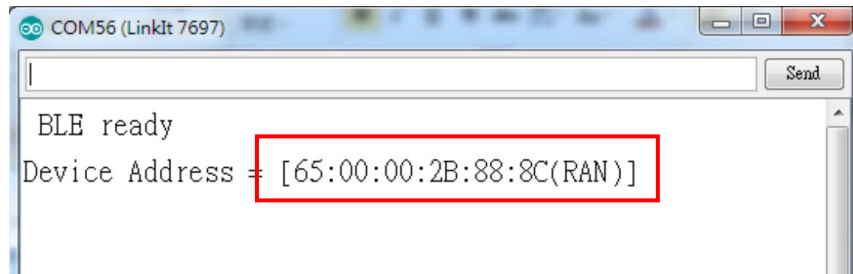
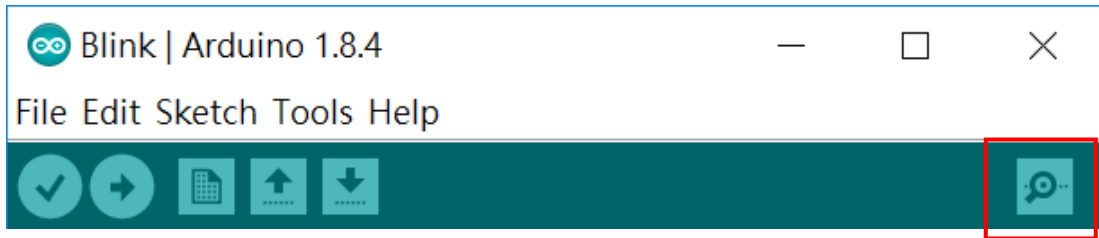


2. Set the Board from **Tools/ Board** menu to "**LinkIt 7697**" and set COM port from **Tools/ COM port** to what you've discovered in Device Manager.
3. Connect the LinkIt 7697 to your computer with a micro-USB cable.
4. Press the "**Upload**" right-arrow button, this will compile and

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



5. 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. Later we will use the same .ino file to receive commands from App Inventor.



App Inventor

The purpose of this project is to control the LED on the LinkIt 7697 dev board with App Inventor, through BLE communication. The trick here is to toggle the digital pin on/off with two buttons. There are many other components in AI2, you can also use them with the same concept. That's why we put a **SpeechRecognizer** component here. You can easily change the trigger from buttons to speech without modifying the Arduino code.

Now log in to your App Inventor account and create a new project.

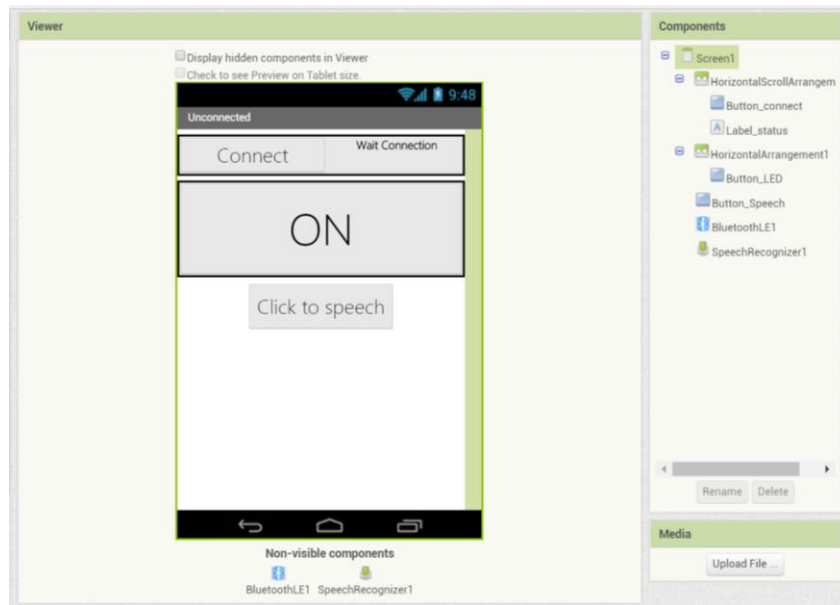
Designer

1. The most used components in this project are buttons (to trigger actions) and labels (to show related messages).
2. We need to import the BLE extension from this URL:
 - <http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.ble.aix>
 - then add it to the project by dragging it into the Viewer.
3. Add a button to send out Bluetooth message to turn the LED on and off, rename it to "**Button_LED**"
4. Add a button to start **SpeechRecognizer**, rename it to

"Button_Speech".

5. Add a **SpeechRecognizer** from the **Media** drawer.

After some adjusting, your designer should look similar to this. 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's Serial Monitor, this is the Bluetooth address of LinkIt 7697.

```
initialize global addr to "7F:0C:00:2B:88:8C"
```

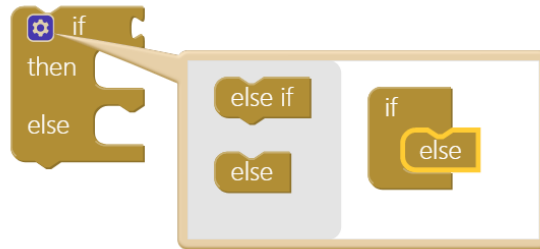
2. Initialize and connect

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

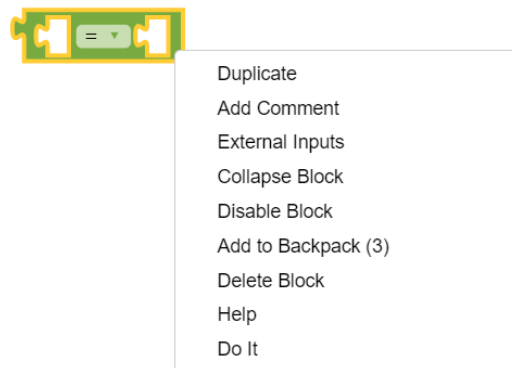
```
when Screen1 .Initialize  
do call BluetoothLE1 .StartScanning
```

3. Button to connect and disconnect from Bluetooth device

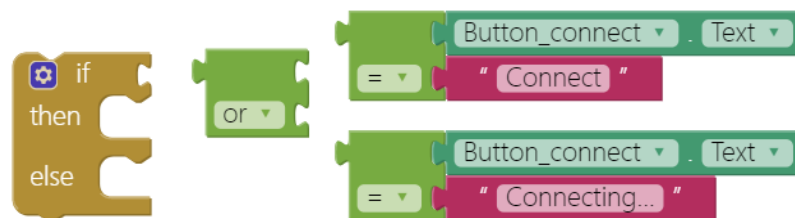
In **Button_connect.Click** event, we are going to connect or disconnect from Bluetooth device depending on the button text. First, add an **if** condition, then click its blue gear to add an **else**.



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.

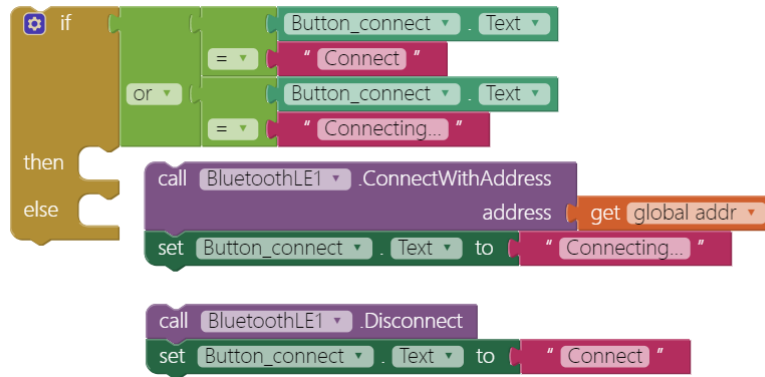


Now we want to check whether the **Button_connect.Text** status is "**Connect**" OR "**Connecting...**", this is how App Inventor decide to connect or disconnect Bluetooth connection with LinkIt 7697. please combine these blocks.



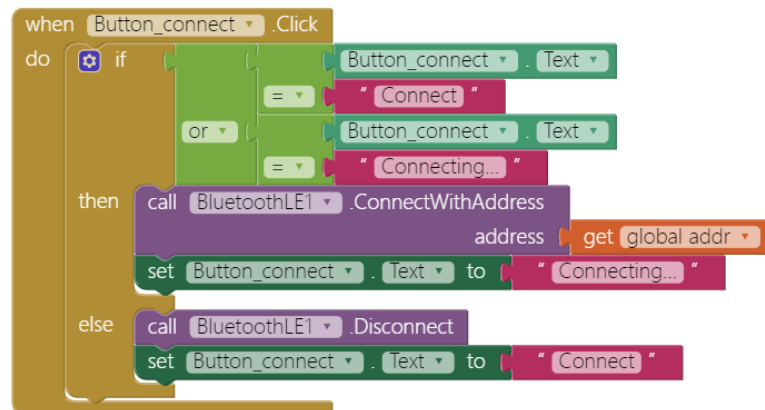
Good! When the **Button_connect**'s text is "**Connect**" OR

"**Connecting...**", we will connect to the specified Bluetooth device (**BluetoothLE1.ConnectwithAddress**), which is our LinkIt 7697. If not, then disconnect (**BluetoothLE1.Disconnect**) and show message on Button_connect.



```
if (Button_connect.Text = "Connect" or Button_connect.Text = "Connecting...")
then
  call BluetoothLE1.ConnectWithAddress with address: get global addr
  set Button_connect.Text to "Connecting..."
else
  call BluetoothLE1.Disconnect
  set Button_connect.Text to "Connect"
```

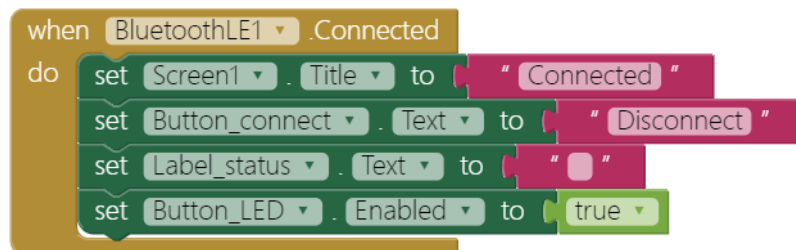
Put everything into **Button_connect.Click** event, and finish like this:



```
when Button_connect.Click
do
  if (Button_connect.Text = "Connect" or Button_connect.Text = "Connecting...")
  then
    call BluetoothLE1.ConnectWithAddress with address: get global addr
    set Button_connect.Text to "Connecting..."
  else
    call BluetoothLE1.Disconnect
    set Button_connect.Text to "Connect"
```

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 by user.

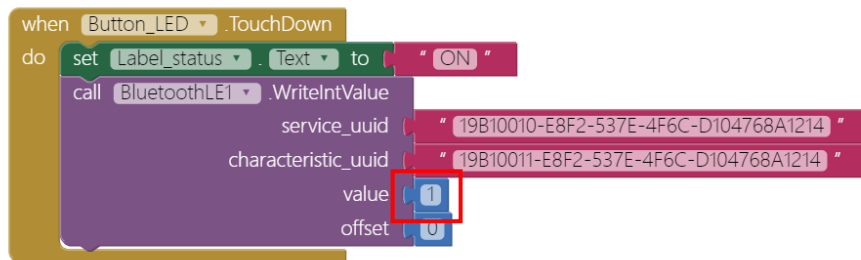


```
when BluetoothLE1.Connected
do
  set Screen1.Title to "Connected"
  set Button_connect.Text to "Disconnect"
  set Label_status.Text to " "
  set Button_LED.Enabled to true
```

5. Button_LED touch down to turn on the LED

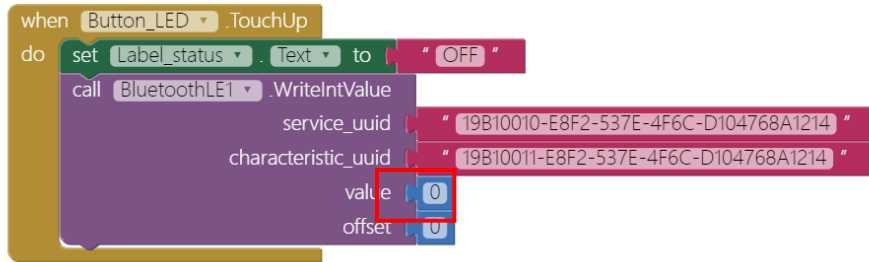
We want to send specific commands through Bluetooth to LinkIt 7697 and the easiest way is with a button. The command here can be anything, like an integer (**23**), a character (**'t'**), etc. But, we must tell LinkIt 7697 what will come from App Inventor. It will perform corresponding actions according to what it has received. You can create as many as commands you like, just make sure they are matched with each other (App Inventor and Arduino sketch).

When **Button_LED** is touched (**Button_LED.TouchDown** event), first we show a related message and send out a number **1** by **BluetoothLE.WriteIntValue** method. Notice that the **service_uuid** and **characteristic_uuid** must be identical with what is in the Arduino sketch. LinkIt 7697 will turn on the LED by putting its digital pin #7 to HIGH voltage level.



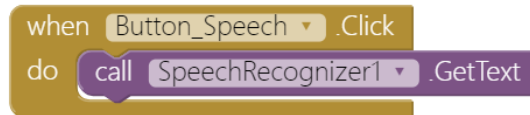
6. Button_close to turn off the LED

We are doing almost exactly the same thing here, except sending out the number **0** instead of **1** when you move your finger away from the button (**Button_LED.TouchUp** event). Then LinkIt 7697 will turn off the LED by putting its digital pin #7 to LOW voltage level.



7. Click to start recognizing speech

SpeechRecognizer will be ready for voice input once **Button_Speech** is clicked. Your screen will show a Google microphone icon, then you can give the commands "**light on**" and "**light off**".



In **SpeechRecognizer.AfterGettingText** event, we check whether the processed result by Google matches with what we set: "**light on**" and "**light off**". If it says "**light on**", then send out a number **1** by **BluetoothLE.WriteIntValue** method; and if it say "**light off**" then send out number **0**.

So, we just changed the trigger of App Inventor here, LinkIt 7697 is still waiting for 0/1 to toggle its pin. You can extend the same idea to other trigger events like an orientation sensor or an incoming phone call (**PhoneCall.PhoneCallStarted** event).

```

when SpeechRecognizer1 .AfterGettingText
  result
do
  set Screen1 . Title to get result
  if
    get result = "light on"
  then
    call BluetoothLE1 .WriteIntValue
      service_uuid "19B10010-E8F2-537E-4F6C-D104768A1214"
      characteristic_uuid "19B10011-E8F2-537E-4F6C-D104768A1214"
      value 1
      offset 0
  if
    get result = "light off"
  then
    call BluetoothLE1 .WriteIntValue
      service_uuid "19B10010-E8F2-537E-4F6C-D104768A1214"
      characteristic_uuid "19B10011-E8F2-537E-4F6C-D104768A1214"
      value 0
      offset 0

```

8. Disconnect

You can disconnect by clicking the **Button_connect** or press the USS button (D6) of LinkIt 7697. This will reset the app to its initial state to wait for next connect request.

```

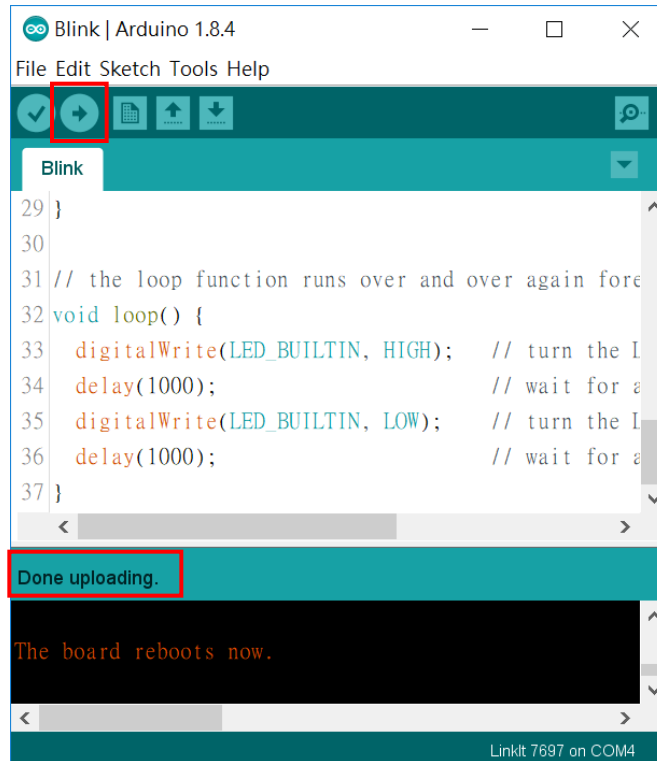
when BluetoothLE1 .Disconnected
do
  set Screen1 . Title to "Unconnected"
  set Label_status . Text to "Wait Connection"
  set Button_LED . Enabled to false

```

Arduino code

Please download from [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.



Tips

Make sure your LinkIt 7697 is running correctly as a BLE peripheral. When you open your app and click **Connect** button, you should see the ON/OFF buttons are ready to be clicked. Just click the ON button to turn on the light and the OFF button to turn it off. And click Button Speech to say "**light on**" and "**light off**" to do the same thing.

Brainstorming

1. Use App Inventor's **OrientationSensor** component to turn on/off the LED.
2. Add one more button to trigger another LED on LinkIt 7697 (*hint: add more cases in Arduino sketch!*)
3. Use a bigger LED (or relay module) with the help of a breadboard.