App Inventor + IoT:
Read analog pin data with LinkIt 7697

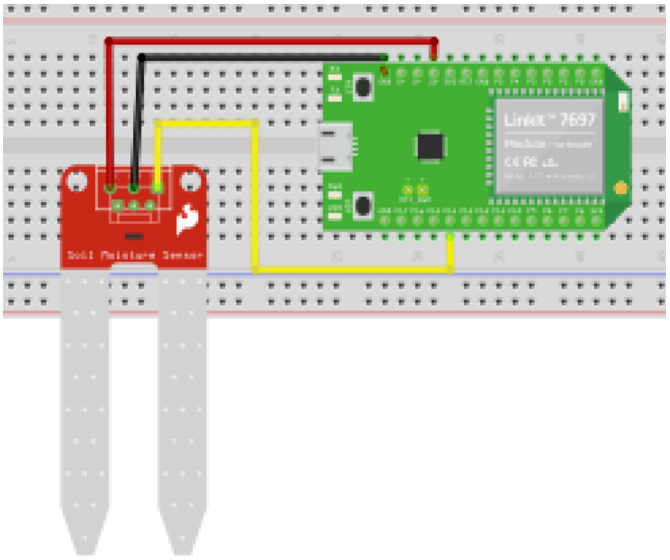
This tutorial will help you get started with App Inventor + IoT. We are going to show LinkIt 7697(Arduino compatible)'s analog pin status on App Inventor screen. Here we are using **Seeed Studio's soil moisture sensor**, but other moisture sensor or any analog input component like potentiometer, photoresistor, thermo-resistor should be fine.

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

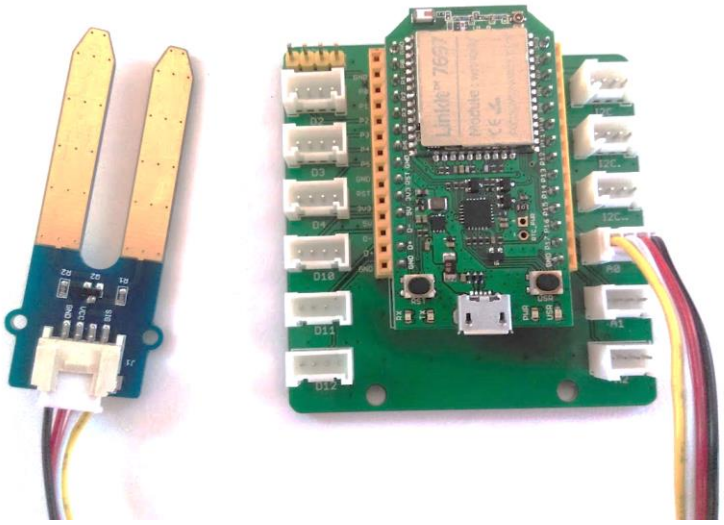
Hardware

LinkIt 7697 is an Arduino compatible dev board with Wi-Fi / BLE. You can use it like just like any other Aruinos and interfacing with App Inventor through its BLE commutation.

If you use general 3-pin analog sensor, please connect SIG pin to LinkIt 7697's P14(A0), Vcc to 5V, GND to GND, finished as below:

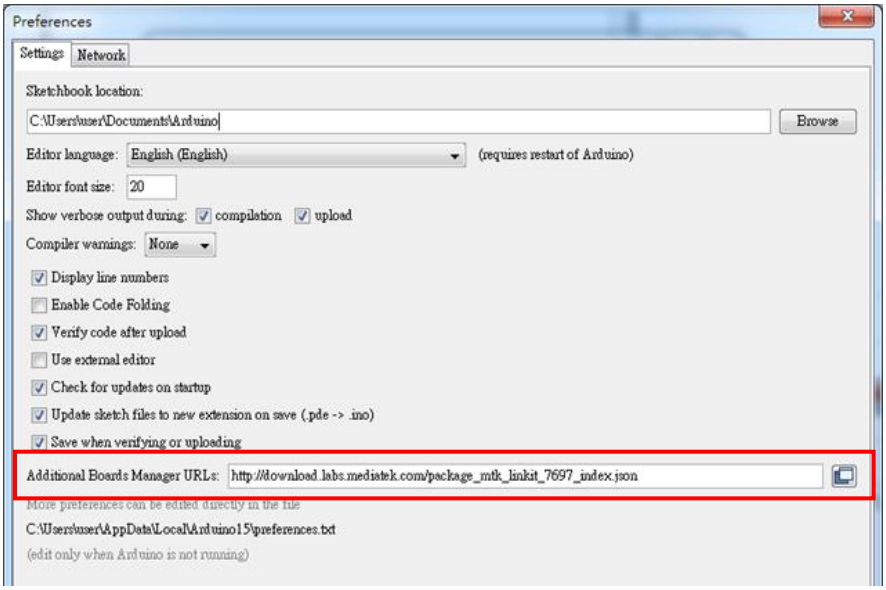


Or you can use Seeed Studio's extension board to connect its **Grove soil moisture sensor**, like below:

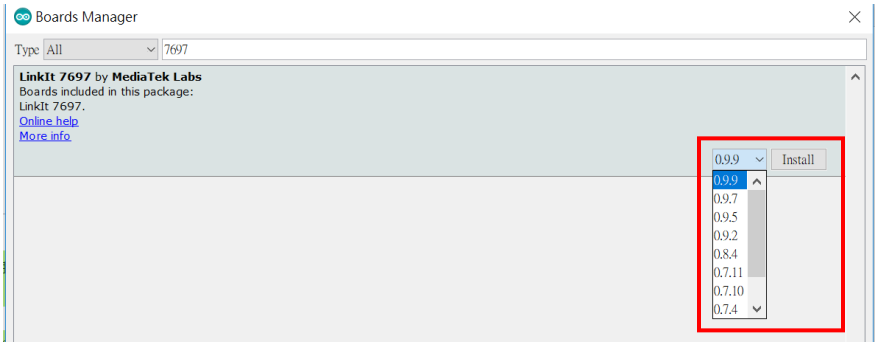


Arduino IDE Setup

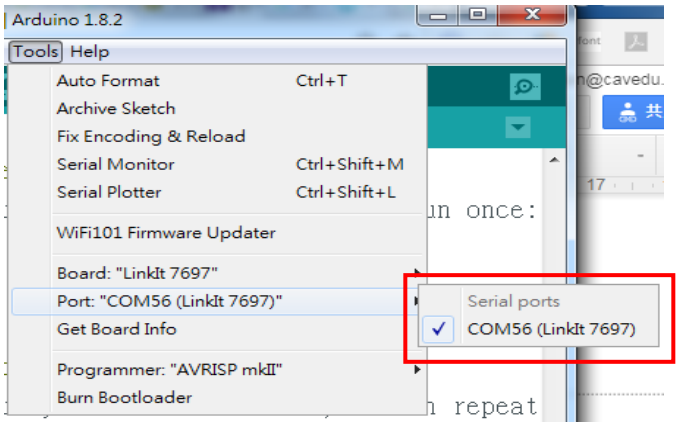
1. Frist get Arduino IDE 1.8.x version, download the .zip file, unzip and click arduino.exe to open the IDE. From File → Preference menu, enter the link below to Additional Boards Manager URLs field:
    - http://download.labs.mediatek.com/package_mtk_linkit_7697_index.json



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



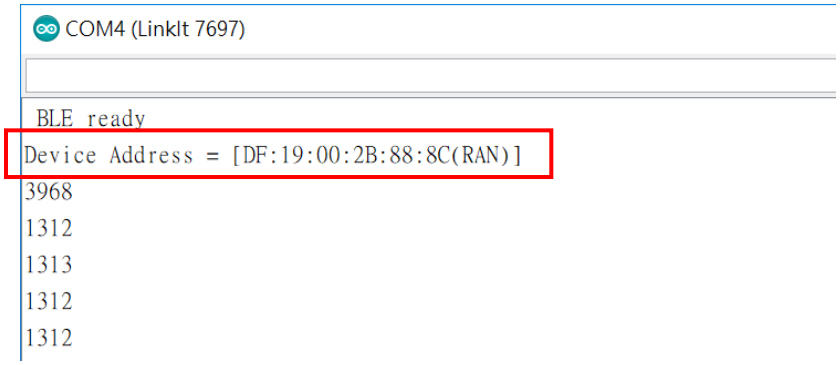
3. Download and install **CP2102N driver**(Windows / MAC/OSX) , then check the COM port in your Device manager. Check if you can see a “**Silicon Labs CP210 USB to UART Bridge(COMXX)**”, this is the COM port number of your LinkIt 7697.
   Finally go back to Arduino IDE, check if IDE had recognized your LinkIt 7697 successfully from **Tools/Port** menu. For MAC user, it should be something like “**/dev/tty.usbserialXXX**…”



Run Arduino sketch

1. For safety reason, not every board marked its Bluetooth address on its board (Arduino 101 is an exception). In Arduino IDE, first set the board to “**LinkIt 7697**” then open **source .ino** of this project.

2. Compile and upload to your LinkIt 7697 then open Arduino IDE's Serial Monitor, should see similar image like below. The [**XX:XX:XX:XX:XX:XX**] 12-digit string following number between ~4095 (analog pin status). The 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 to receive command from App Inventor.



App Inventor

The purpose of this project is to interact with LinkIt 7697 dev board with App Inventor through BLE communication. The main idea is to receive 7697's analog pin status(moisture sensor in our case) and change images according to the pin status.

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

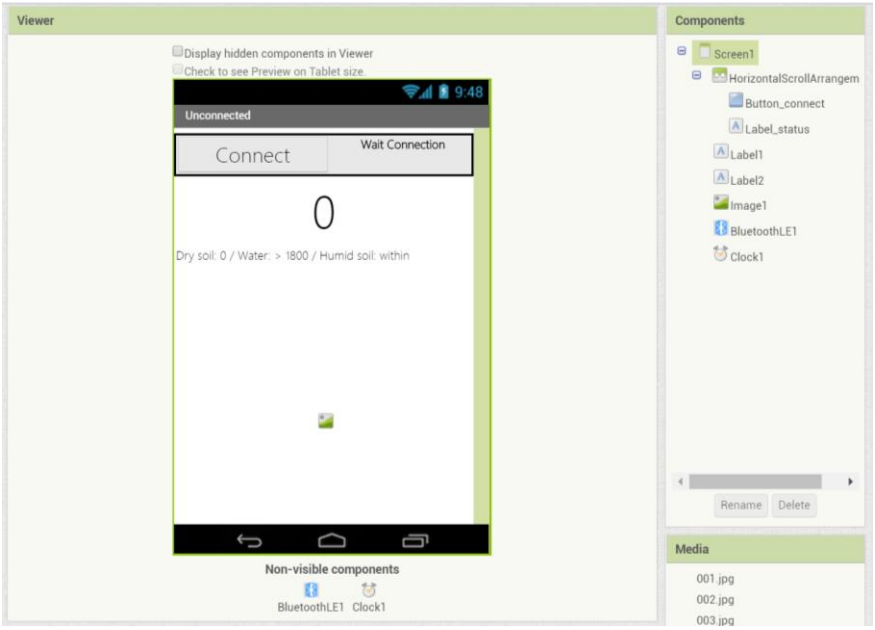
Designer

1. The most used components in this project are buttons (to trigger actions) and labels (to show related messages).
2. And we have to import BLE extension from URL:
   - http://iot.appinventor.mit.edu/assets/com.bbc.micro:bit.profile.aix
   - add a BLE extension by dragging it into Viewer.
3. Add one **HorizontalArrangement** component from Layout drawer and set its width to "**Fill Parent…**"
4. Add one **button** and one **label** the put them into

HorizontalArrangement component. The button is used to connect/disconnect with your Bluetooth device.

5. Add two Labels, one for displaying sata fom LinkIt 7697, the other is for message (Modify its text to ”**Dry soil: 0 / Water: > 1800 / Humid soil: within**”).
6. Add a **Clock** from **Sensor** drawer, and set its **TimerInterval** to 100 (milliseconds). We use it to ask for LinkIt 7697’s pin data 10 times per second.
7. Add an **Image** from **Media** drawer.
8. Upload three images you like to represent three statuses of your plant.

After some adjusting, your designer should be like this. Don’t have to be exactly the same, feel free to modify:



Blocks

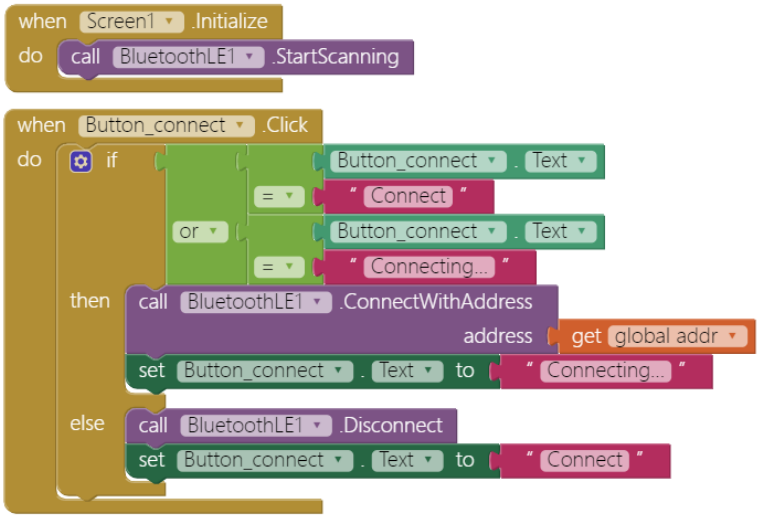
Let’s take a look of our blocks step by step:

1. **Variable for Bluetooth address**

   Please replace the value with what you get from Arduino’s Serial Monitor.

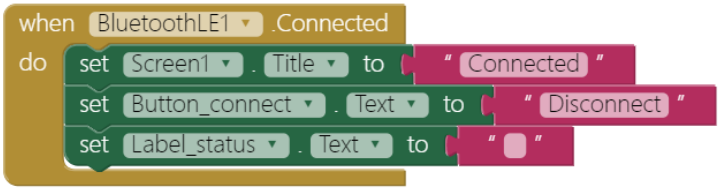
2. **Initialize and connect**

   The app will start scanning for BLE devices nearby. In **Button_connect** event, we will check current connect status then decide to connect or disconnect.



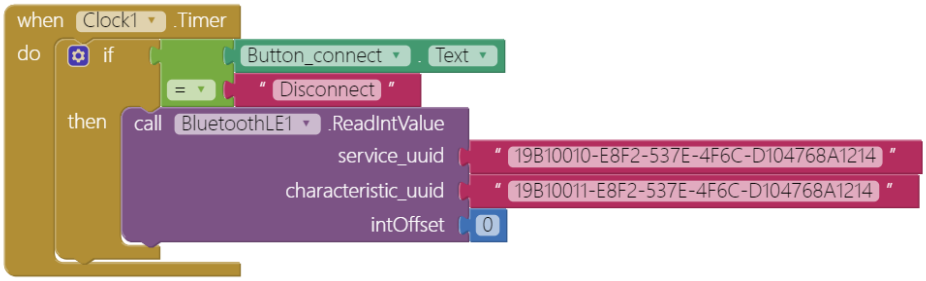
3. **BLE Connected**

   When connected successfully (**BluetoothLE.Connected** event), we show related messages on several components.



4. **Ask for LinkIt 7697's data periodically**

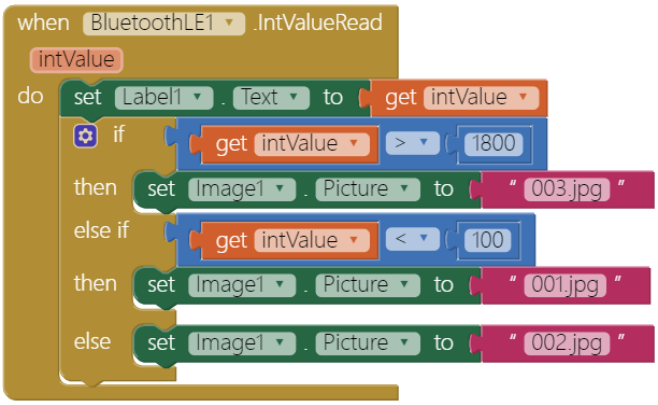
   In **Clock.Timer** event, we first check the connection is still there and ask to read analog pin data from LinkIt 7697 using **BluetoothLE.ReadInt** method. Notice that the **service_uuid** and **characteristic_uuid** must be identical with what in Arduino sketch.



5. **Show received data on label and change images**

   If read successfully, **BluetoothLE.IntValueRead** event will be triggered. In the event, we first show the received data on Label1, and change **Image1** component's Picture according. If the value is greater than 1800, means the soil is too wet, than change to **003.jpg**; if smaller than 100, means too dry, than change to **001.jpg**. If it is within the 100 to 1800, means the soil is well moisturized, then change to **002.jpg**. Check the last section for the screenshots.

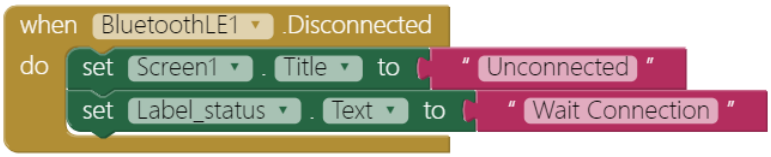
   Note: the analog pin data range from 0 to 4095(12-bit resolution). A bit different from Arduino (0 to 1023, 10-bit).



6. **Disconnect**

   The connection will be disconnected if you click the **Button_connect** or pressed the USS button(D6) of LinkIt 7697. This will reset the app to initial state and wait for next connect

request.



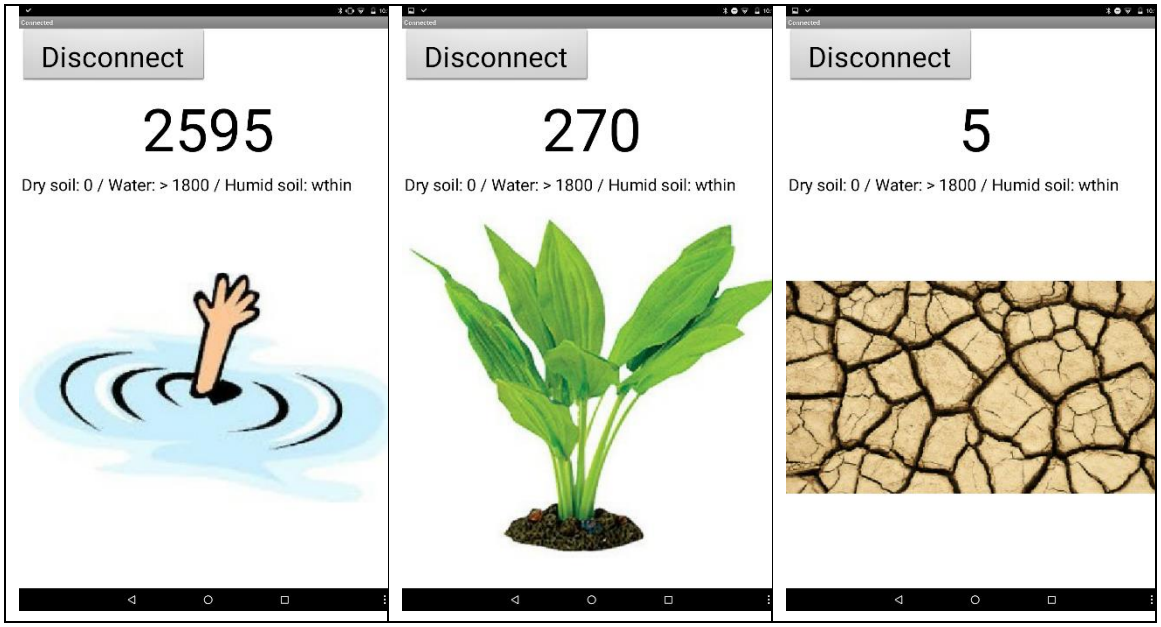
Arduino code

```
#include <LBLE.h>
#include <LBLEPeriphral.h>

int data;

LBLEService AService("19B10010-E8F2-537E-4F6C-D104768A1214");
LBLECharacteristicInt ARead("19B10011-E8F2-537E-4F6C-D104768A1214",
LBLE_READ | LBLE_WRITE);

void setup()
{
  Serial.begin(9600);
  LBLE.begin();
  while (!LBLE.ready()) { delay(100); }
  Serial.println("BLE ready");

  Serial.print("Device Address = [");
  Serial.print(LBLE.getDeviceAddress());
  Serial.println("]");

  AService.addAttribute(ARead);

  LBLEPeripheral.addService(AService);
  LBLEPeripheral.begin();
  LBLEAdvertisementData advertisement;
  advertisement.configAsConnectableDevice("BLE Ana");
  LBLEPeripheral.advertise(advertisement);
}

void loop()
{
  data = analogRead(A0);
```

```
    Serial.println(data);
    ARead.setValue(data);
    delay(1000);
}
```

Have Fun!

Make sure your LinkIt 7697 is running correctly as a BLE peripheral. Open your app and click **Connect** button, you should see the larger number in the middle of screen is varying according to your moisture sensor. Try to get a cup a water and put the moisture sensor into the water and see how the value and picture change.



Brainstorming

1. Use orientation sensor to turn on/off the LED.
2. Add two more buttons to trigger another LED on LinkIt 7697 (hint: more cases in Arduino sketch!)