

Laboratory 7: Android Phone based IoT interactions with Node-Red and AWS IoT

Instructor: Young H. Cho T.A.: Arthur Win

Due Oct 19 at 11:59 PM (Report), Oct 21 at 11:59 PM (Video)

You will use an Android phone that has various sensors such as GPS, accelerometer, gyro, barometer, light sensor, camera, mic, etc. to generate the data, Node-RED to capture and process the data with least amount of coding, and use AWS services to visualize the data in a distributable dashboard.

1. Install Termux & Termux API

Download these two required apps from the play store, following are the links.

<https://play.google.com/store/apps/details?id=com.termux>

<https://play.google.com/store/apps/details?id=com.termux.api>

2. Install Node-RED & Termux API Module on Termux

Now open the Termux App and run the following commands to install Node-RED

```
apt update
apt upgrade
apt install termux-api
apt install coreutils nodejs
npm i -g --unsafe-perm node-red
```

If you are running the latest version of Android you might face an error on the “**npm I -g --unsafe-perm node-red**” line saying “**cannot read property ‘length’ of undefined**”. Run the following commands if you face them.

```
apt-get install yarn
yarn global add npm
```

3. Launch Node-RED

Now launch Node-RED

```
node-red
```

Node-RED is now running on localhost on the Android phone; it will be showing the IP address and port it is running on in Termux.

On your phone you can point a browser to localhost:1880 and it will open up Node-RED flows editor. But for practical purposes you'll have to open it up on a bigger screen such as a laptop.

To access the flows editor on a browser from a laptop running on the same network as the phone, you can do it in two ways.

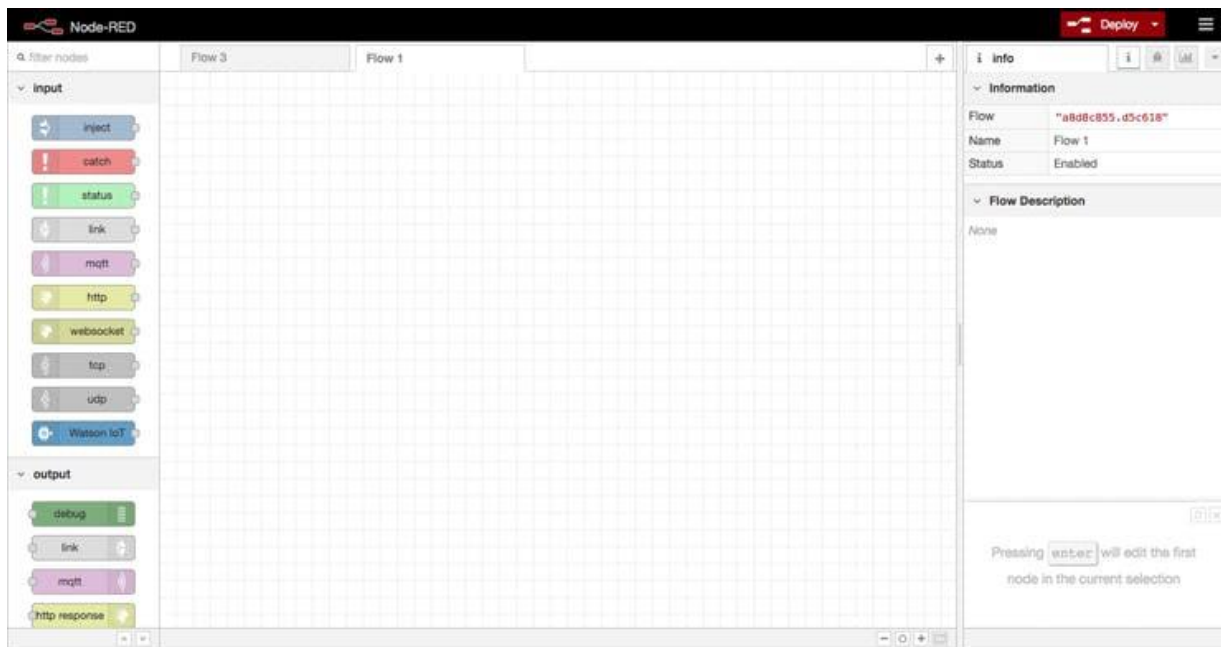
One, your laptop/desktop has to be connected to the same WiFi router as the phone is connected to.

Two, you can run a WiFi hotspot from your phone and connect to the hotspot from your laptop.

Now type in ifconfig in Termux and get your phone's IP address.



Add a **colon** and port number of “**1880**” to the ip address and enter it into a browser from the laptop/desktop. In my case it was 192.168.100.2:1880



Node-RED Blank Flow

4. See All the Sensors

Open a new session in Termux; swipe left drawer and click new session to open a new session. Switch between the session from the same left drawer menu. You can discontinue a termux-process running in the current session by pressing phone's **Volume Down** + **c** in the keyboard.

To get a list of available sensor in the phone, type in Termux:

```
termux-sensor -l
```



```
Welcome to Termux!
Wiki:      https://wiki.termux.com
Community forum: https://termux.com/community
IRC channel: #termux on freenode
Gitter chat: https://gitter.im/termux/termux
Mailing list: termux+subscribe@groups.io

Search packages: pkg search <query>
Install a package: pkg install <package>
Upgrade packages: pkg upgrade
Learn more:     pkg help
$ termux-sensor -l
{
  "sensors": [
    "BMI160 accelerometer",
    "BMI160 gyroscope",
    "BMI150 magnetometer",
    "BMP280 pressure",
    "BMP280 temperature",
    "RPR0521 Proximity Sensor",
    "RPR0521 Light Sensor",
    "Orientation",
    "BMI160 Step detector",
    "Significant motion",
    "Gravity",
    "Linear Acceleration",
    "Rotation Vector",
    "Geomagnetic Rotation Vector",
    "Game Rotation Vector",
    "Pickup Gesture",
    "Tilt Detector",
    "BMI160 Step counter",
    "BMI150 magnetometer (uncalibrated)",
    "BMI160 gyroscope (uncalibrated)",
    "Sensors Sync",
    "Double Twist",
    "Double Tap",
    "Device Orientation",
    "BMI160 accelerometer (uncalibrated)"
  ]
}
$
```

This list will vary based on handset model.

Termux will print the value of all the sensors continuously if you type

```
termux-sensor -a
```

This consumes a lot power. To print it once, type in

```
termux-sensor -a -n 1
```

To get all the list of options available, just type in:

```
termux-sensor
```

5. Filter Light Sensor

Light sensor is typically placed beside front camera (selfie camera) along with proximity sensor, and pretty much available in all handset.

Take the light sensor data to showcase how to extract and pipe this data.

To filter out the light sensor in termux-sensor, find the name from the list of printed sensor, in my case I picked "**RPR0521 Light Sensor**" in my Nexus 5X. To get a filtered data with this sensor, type in

```
termux-sensor -s "RPR0521 Light Sensor" # User your sensor name here
```

You will see data from the selected sensor.

6. AWS IoT Setup

In the AWS IoT Core console, create a new policy in the “Policies” tab under “Secure”.

In the “Action field, enter: **"iot:*"** for all IoT actions, and under the “Resource ARN” field, enter **"*"** for all devices and topics. Check “Allow” for the “Effect” field.

Back in the AWS IoT Core console, create a new device in the “Things” tab under “Manage”. Name your device, create a new type, and a new group for your IoT device.

Select “One-click certificate creation” and download your certificate, public, and private keys for your device. Make sure to click on “Activate” for your root certificate and click on the download link. This will not actually download anything and will redirect you to a download page with several root certificates. Download either Amazon Root CA 1 or CA 3. Lastly, attach the policy you created to your device.

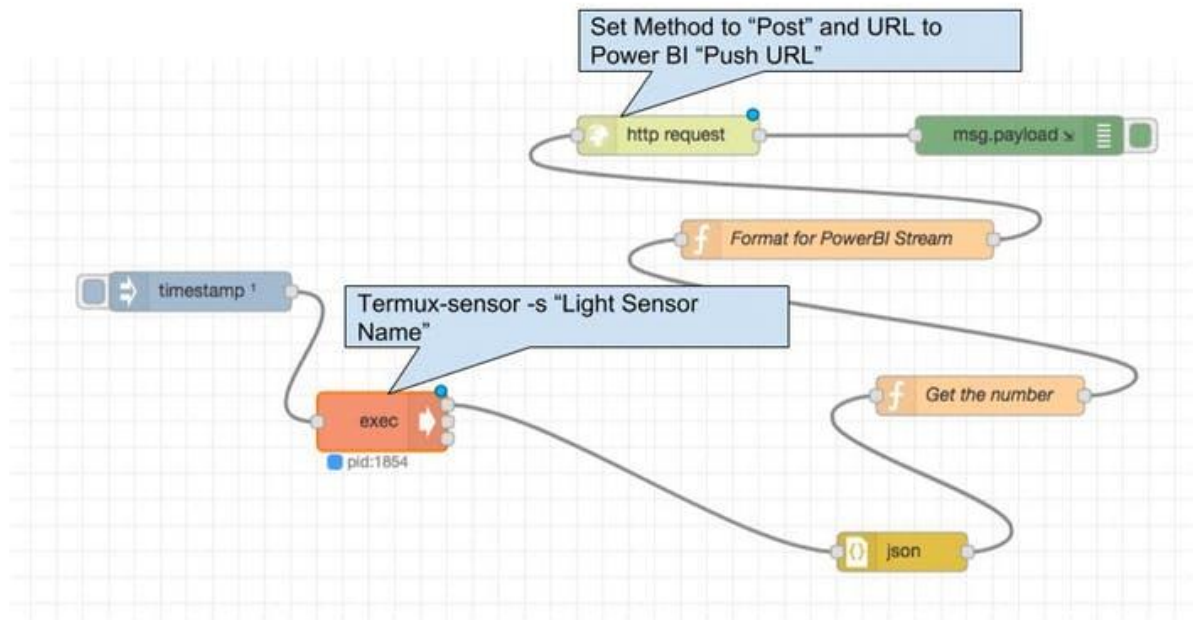
7. Post Data From the Phone to AWS IoT

Use Node-RED's "exec" node to run a terminal command and get the sensor data into Node-RED's flow editor, then convert the data into JSON, and finally post the data to AWS IoT Core.

To import any external flow into Node-RED, click on the Hamburger Menu on the Top Right, then click **Import > Clipboard**, you'll get an import dialogue box, then copy paste the following JSON content into the box and hit **Import**.

```
[{"id":"c12e9e18.651ae","type":"tab","label":"Flow 3","disabled":false,"info":{}},{id":"bec0b05.54ba45","type":"inject","z":"c12e9e18.651ae","name":"","topic":"","payload":"","payloadType":"date","repeat":"1","crontab":"","once":true,"onceDelay":"","x":170,"y":200,"wires":[["a9c3b25c.f6cc1"]]},{"id":"a9c3b25c.f6cc1","type":"exec","z":"c12e9e18.651ae","command":"","addpay":true,"append":"","useSpawn":"false","timer":"","oldrc":false,"name":"","x":210,"y":300,"wires":[["dda5983.8467f68"],[],[]]},{id":"dda5983.8467f68","type":"json","z":"c12e9e18.651ae","name":"","property":"payload","action":"","pretty":false,"x":550,"y":380,"wires":[["3fed63c8.f80ccc"]]},{"id":"3fed63c8.f80ccc","type":"function","z":"c12e9e18.651ae","name":"Get the number","func":"msg.payload.light_sensor = msg.payload[\"RPR0521 Light Sensor\"][\"values\"][0];\nreturn msg;","outputs":1,"noerr":0,"x":610,"y":260,"wires":[["a8cd33f3.1c049"]]},{"id":"a8cd33f3.1c049","type":"http request","z":"c12e9e18.651ae","name":"","method":"POST","ret":"txt","url":"","tls":"","x":370,"y":60,"wires":[["e7c0faa5.356a38"]]},{"id":"a8cd33f3.1c049","type":"function","z":"c12e9e18.651ae","name":"Format for PowerBI Stream","func":"var timestamp = new Date();\nmsg.payload = JSON.stringify([{\n  \"light_sensor\":msg.payload.light_sensor,\n  \"datetime\":timestamp\n}]);\nreturn msg;","outputs":1,"noerr":0,"x":500,"y":140,"wires":[["6d1fabef.8f4bf4"]]},{"id":"e7c0faa5.356a38","type":"debug","z":"c12e9e18.651ae","name":"","active":true,"tosidebar":true,"console":true,"tostatus":false,"complete":"payload","x":640,"y":60,"wires":[]}]
```

Node-RED Flow Import Process



Now, click on **exec** node and put in **termux-sensor -s "Light Sensor Name"-n 1**, make sure to put in your light sensor name. Also, because this command outputs a value with a single number, the rest of the flow nodes are designed to handle that single value; if you call for any sensor that return multiple values, the later nodes will break. You will also have to modify the **"Get the number"** node to the name of your light sensor.

Go back into AWS IoT Core and copy the endpoint URL found in the “Settings” tab.

In the **HTTP request** node, set method to "Post" and paste the endpoint URL that we got from AWS IoT. Append the following text in yellow to your URL so it looks like the one shown below.

XXXX-ats.iot.us-east-1.amazonaws.com:8443/topics/myTopic?qos=1

You can change “myTopic” in the URL to the name of any topic name of your choosing.

Check the box for “Enable secure SSL/TLS connection” and leave it as a TLS connection. Click on the pencil to open the “Properties” menu.

Upload the root certificate, private key, and certificate files for your device in the fields listed. Save your changes to the node and exit the configuration menu.

Click on the **Deploy** button on the upper right corner of Node-RED flows window.

If you click on the bug icon on the upper right corner you should see the debug window and all the debug messages.

You should begin to see messages in the following format in the debug window

```
"{"message":"OK","traceId":"XXXX"}"
```

If you see “OK” as shown above, your packet was successfully sent to AWS IoT Core.

In AWS IoT Core, go into the “Test” tab. Subscribe to your topic using your chosen topic name and verify that IoT Core is reading your packets correctly.

8. Visualize Data in AWS IoT

Install ThingsBoard CE on an EC2 instance following the instructions in the link below. You are free to use any supported OS.

<https://thingsboard.io/docs/user-guide/install/installation-options/>

Send your data stream to your instance and create a dashboard in ThingsBoard. This dashboard must display a real-time graph of your light sensor value on the y-axis and time on the x-axis. Tutorials on using ThingsBoard are in the following link.

<https://thingsboard.io/docs/guides/>

Note that there are multiple ways you may route your stream to your instance. Some possible methods include modifying your Node-RED flow or republishing your stream from AWS IoT. You are free to achieve this in anyway you like, with any combination of AWS tools available.

Sources

<https://nodered.org/docs/platforms/android>

<https://nodered.org/docs/node-red-admin>

<https://github.com/termux/termux-packages/issues/1855>

<https://www.hackster.io/mainul/build-an-iot-viz-with-your-android-phone-s-sensors-cc241e>