Laboratory 10: Using AWS to Collect xDot data

Instructor: Young H. Cho

1. Connect the antenna to the conduit and the 9V adapter to a power source to the conduit.

2. Make a connection with the ethernet cable from the conduit to your PC.

3. Once the connection is established. Open a web browser on your PC and go to 192.168.2.1 (the default IP address for the gateway – this IP address should be verified with an IP tool).

4. Log in to the GUI, the default username:password is admin:admin. In case these credentials fail, try resetting the device. (You can also ssh admin@192.168.2.1. The password is admin)

5. If you login for the first time you will be guided through an initial setup, configure the network interface that is connected to the conduit to be a static IP address with 192.168.2.2 - 192.168.2.254

6. Click on LoRaWAN-> Network Settings to the left and change the mode to Network Server. Add changes to the configuration as so:
    a. Channel Plan: US915
    b. Frequency Band: 7
    c. Network Mode: Public LoRaWAN
    d. Join Delay: 5

    Hit Submit.

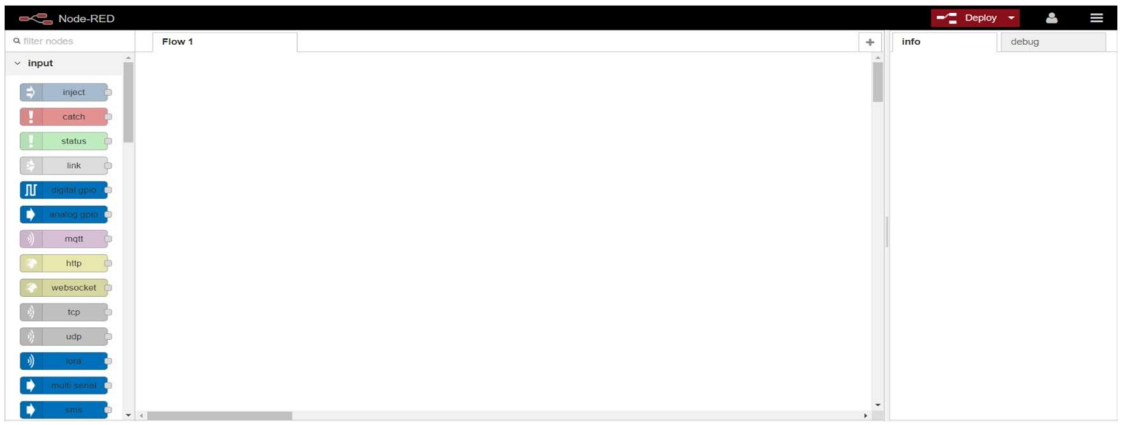
7. Navigate to LoRaWAN-> Key Management. Under Location select Local Keys. In Settings change the configuration to:
    a. Enabled: True
    b. Network ID: Name
    c. Name: <Any valid name with min 8 chars>
    d. Passphrase: <Any valid name with min 8 chars>

    Hit Submit.

8. Restart the LoRaWAN and under LoRaWAN-> Network Settings, the status of network server should be green/running

9. Now navigate to Apps and enable Node-Red Apps. Launch Node-Red and wait for the browser to load (Please be patient, the browser might reload multiple times).

10. If asked for the login details enter the default login credentials. The page should load to something like:.



11. From Input Section to the left drag the lora node/block to the flow. An instance of the lora node will be created on the flow screen. And from the output section drag the debug node to the flow. Connect both the nodes as shown below. I hope you get the basic idea:

12. Open the debug window on the right and click on Deploy to the top.

13. Let us join the network and try sending a message. For that I am making use of the AT commands which you can find here: https://www.multitech.com/documents/publications/reference-guides/S000643-mDot-AT-Command-Guide.pdf

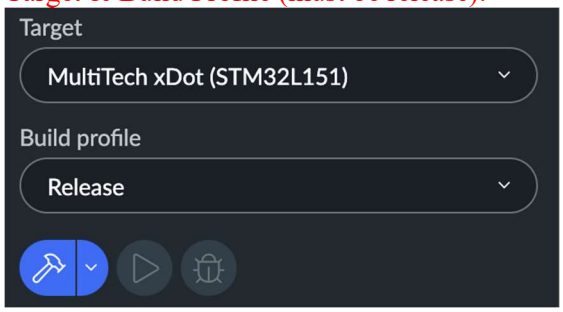
14. First we need the AT command firmware to be uploaded to the xDot (In MBedStudio)

    Select File -> Import Program
    In URL window, type: http://os.mbed.com/teams/MultiTech/code/Dot-AT-Firmware/
    Then click on Add Program

    Once the program is completely loaded, Select File -> Add Library to Active Program…
    In Git or os.mbed.com URL window, type: https://github.com/MultiTechSystems/libxDot.git
    Then click on Next>>
    Select the master branch then click on Finish

15. Compile the Dot-AT-Firmware program by clicking on the hammer

    Target & Build Profile (must be release):

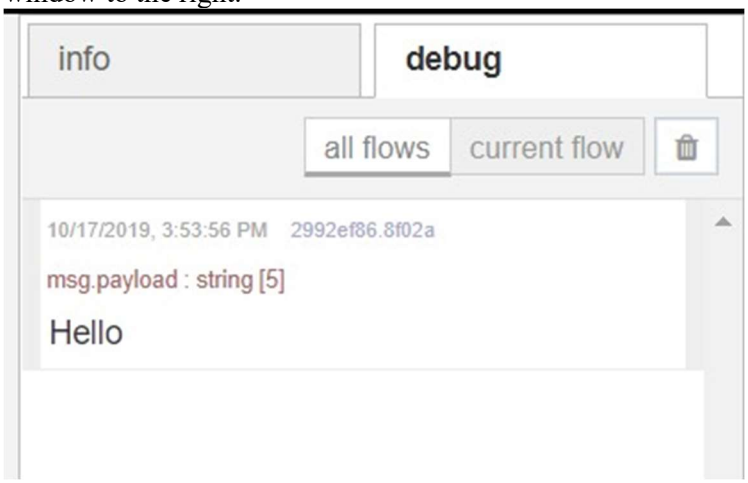


Once bin file is successfully created, copy the bin file to the xDot (Make sure the transmission did not fail). Open a serial connection to the USB UART (**not the debug port**) (port COM3, Windows needs to install serial driver like Lab8) with a baud of 115200 (or the baud you specified for serial communication in main.cpp). Hit the reset button on the xDot and type AT. The xDot must return an OK which determines a successful connection and a working AT firmware. If you are having a too hard of a time getting the firmware to compile correctly, you may have to use the following precompiled binary:
https://www.multitech.net/developer/downloads/#xdot-US915-mbed-os-6.8.0.bin

16. Now let's configure the network parameters on the xDot using the AT commands.
    at+ni=1,<The name of the Lora network provided in step 6>
    at+nk=1,<The passphrase for the Lora network provided in step 6>
    at+fsb=7
    at+jd=5
    at&w
    atz
    at+join
    at+send=Hello

```
AT

OK
at+ni=1,multitech123
Set Network Name: multitech123

OK
at+nk=1,multitech123
Set Network Passphrase: multitech123

OK
at+fsk=7
Command not found!

ERROR
at+fsb=7

OK
at

OK
at+jd=5

OK
at&w

OK
atz

OK
at+join
Successfully joined network

OK
at+send=Hello

OK
```

17. If you navigate back to the Node-Red app you will see the Hello message under the Debug window to the right.



20. And for the last part we will be sending the data collected to the LoRa conduit and check the output on the Node-Red debug window.

21. For the LoRa configuration on the xDot, use the xDot Examples program OTA Example skeleton.

    As before, import xDot Example with URL
    http://os.mbed.com/teams/MultiTech/code/Dot-Examples/

22. Click on import program under Dot-Examples and import the program to the compiler. Import the library as you did in the earlier steps.

23. Now in the compiler's project explorer navigate to Dot-Examples-> examples-> example_config.h. Study the file and make sure the OTA_EXAMPLE is your ACTIVE_EXAMPLE.

24. Now navigate to Dot-Examples-> examples-> src-> ota_example.cpp and make the following changes:
    a. network_name = <The name of the Lora network provided in step 7>
    b. network_passphrase = <The passphrase for the Lora network provided in step 7>
    c. frequency_sub_band = 7
    d. join_delay = 5

25. Send EE542 to the conduit by making the following changes
    a. Commented lines 132-139

```
char arr[] = "EE542";
while (true) {
    uint16_t light;
    std::vector<uint8_t> tx_data;

    // join network if not joined
    if (!dot->getNetworkJoinStatus()) {
        join_network();
    }
    for(int i=0;i<sizeof(arr)-1;i++){
        tx_data.push_back(arr[i]);
        }

    send_data(tx_data)
```

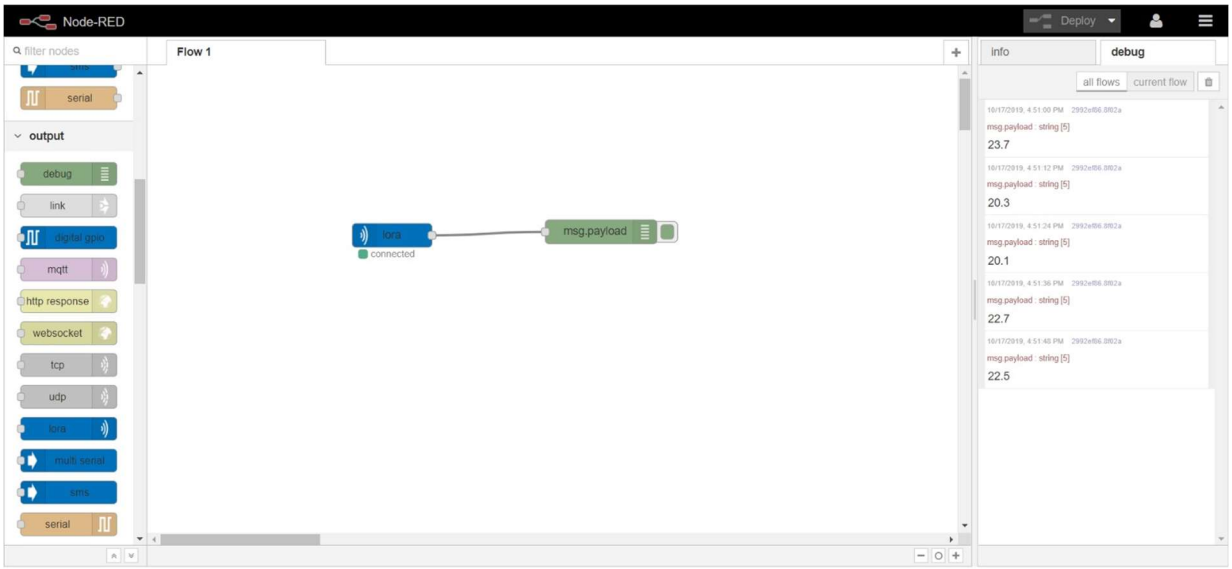
26. Verify that you are receiving the message on the Node-Red debug window. Also study the ota_example.cpp to better understand the flow and changes needed to-be made for sending the sensor data. You can also connect to the serial debug port of xDot (baud 115200) to monitor/capture logs.

27. Integrate the code from Lab8 part I to the ota_example to send the sensor data through LoRa.

```
#define BETA 3975
AnalogIn sensor(PB_0);
DigitalOut vcc(GPIO0);

double Thermistor(int RawADC) {
  double Temp;
  Temp= (float) 10000.0 * ((65536.0 / RawADC) - 1.0);
  Temp = (1/((log(Temp/10000.0)/BETA) + (1.0/298.15)));
  Temp = Temp - 273.15;             // Convert Kelvin to Celcius
   //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
   return Temp;
}

```

```
155     char out[5];
156
157     while (true) {
158         std::vector<uint8_t> tx_data;
159         // join network if not joined
160
161         if (!dot->getNetworkJoinStatus()) {
162             logInfo("Joining Network");
163             join_network();
164             logInfo("Check Status");
165         }
166         val=sensor.read_u16();
167         double temp =  Thermistor(val);
168         //pc.printf("%f\n",temp);
169         snprintf(out, 5, "%f", temp);
170         for(int i=0;i<5;i++){
171             tx_data.push_back(out[i]);
172             }
173         send_data(tx_data);
174
```

Also don't forget to include necessary headers. In order to use snprintf you will need to include stdlib.h



28. In order to use conduit as the gateway for your xDot nodes to the Internet, you will need to configure the conduit to connect to your home router that is connected to the Internet. Once you set the conduit up to get an IP address via your router's built-in DHCP server (you should be able to see which IP address was assigned to it by logging on to your Internet router and view its DHCP table), the conduit should get assigned a different IP address when you connect it to the router with an Ethernet cable. Then you should be able to get the same webpage access with your PC through the router using the IP address assigned by the router. At the same time, your conduit should have the access to the Internet.
29. Configure Node-Red in a similar way that your configured Node-Red on the phone to send your xDot data to AWS.
30. Add dash board widgets in your thingsboard to visualize your xDot data over time.