How to Connect Real-world Data to an AT&T M2X Device

Introduction

This technical note shows you how to send real-world data, in this example a temperature in degrees F, to an M2X device.

What you'll need:

- An account on m2x.att.com
- A configured M2X device and its API Key. In this example, the M2X device we're using is called **OptoBrainTest**, with a Stream ID called **LampTemperature**.
- An Opto 22 SNAP PAC controller with its RESTful server enabled. In this example, we're communicating with a SNAP-PAC-R1 that's part of a SNAP-PACLC Learning Center. This Learning Center hardware also includes a temperature probe.
- A Node-RED flow to connect the Opto 22 hardware to the M2X device.

It's as easy as 1-2-3:

1. Configure the Opto 22 PAC hardware

2. Configure the Node-RED flow to connect the Opto 22 PAC to M2X

3. Check out your data in the M2X cloud!

1. Configure the Opto 22 PAC hardware

Assemble and program the SNAP-PAC-R1 as described in the *SNAP PAC Learning Center User's Guide*, Opto 22 form 1638.

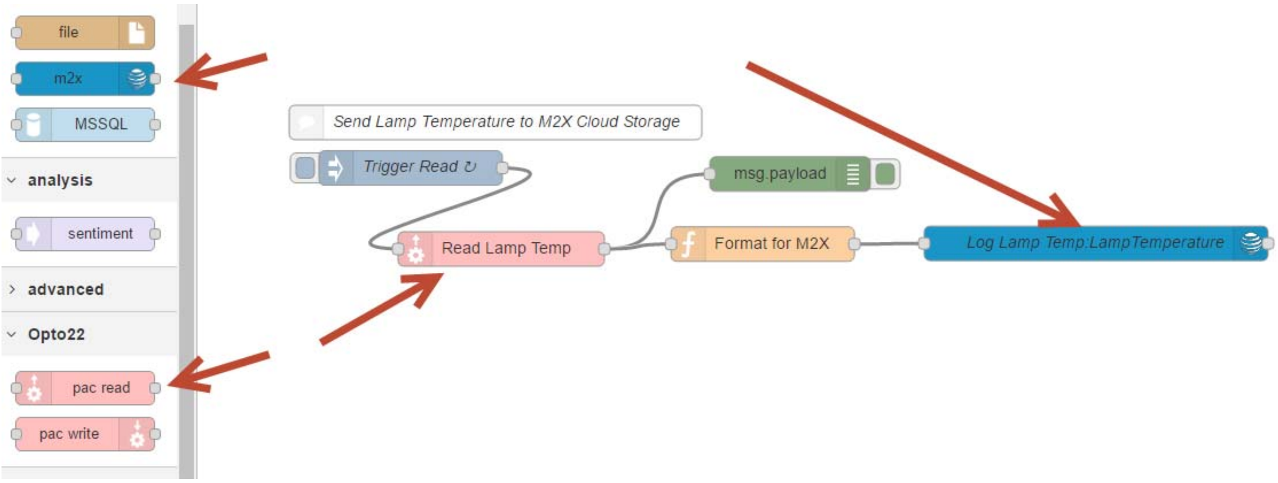
Note: for this example, we renamed the temperature input in the control strategy from Store_Temperature to Lamp_Temperature. We also added an output to turn on a Lamp so we could remotely increase that temperature by turning on the lamp. While this document only describes how to read/monitor the temperature, controlling an output, such as turning on the lamp, can also be done via similar steps.

2. Configure the Node-RED flow to connect the Opto 22 PAC to M2X

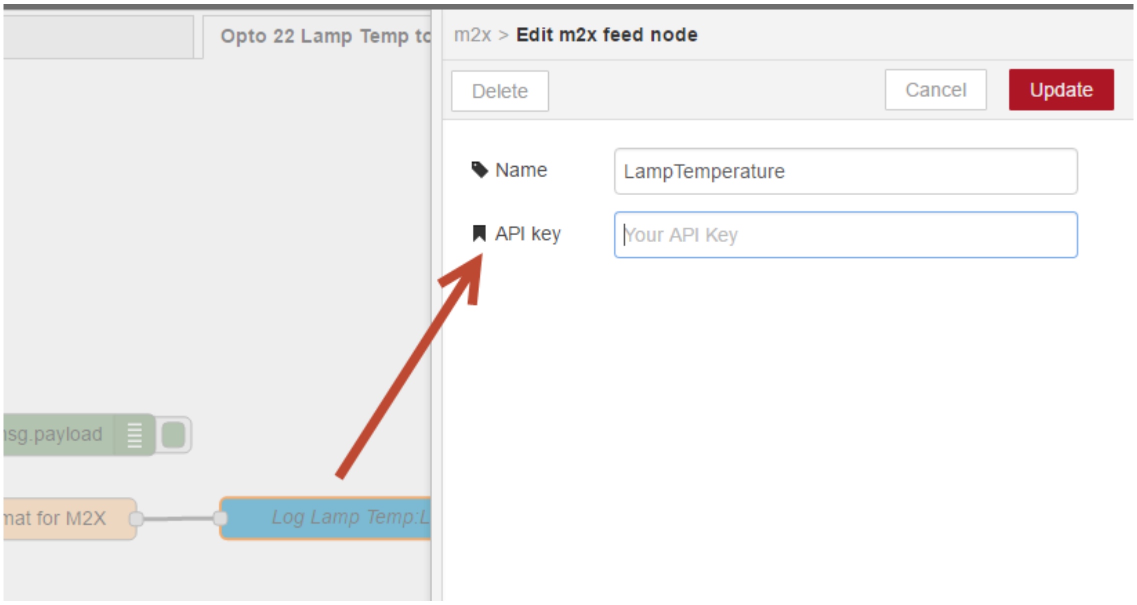
1. Install Node-RED, including the SNAP PAC nodes, as described on this Node-RED for SNAP-PAC page on Opto 22's developer website (developer.opto22.com).
2. Make sure you enable the RESTful server on the PAC, which the SNAP PAC Node-RED nodes use to read or write to the controller.
3. Install the Node-RED mx2 node.

4. Create or import a flow. For example, the flow we used is shown below as the importable JSON and as a screenshot of the resulting flow:

```
o[{"id":"20311063.16361","type":"m2x
feed","z":"","apiKey":"fbb31b1e50130ab4ff299d035111fbbc","name":"M2X"}
,{"id":"db122dcb.144e6","type":"pac-device","z":"","address":"restpac.
groov.com","protocol":"http"},{"id":"46f6c401.9d0b2c","type":"pac-read
","z":"33885e25.21d752","device":"db122dcb.144e6","dataType":"ana-inpu
t","tagName":"Lamp_Temperature","tableStartIndex":"","tableLength":"",
"name":"Read Lamp
Temp","x":220.0994415283203,"y":869.727294921875,"wires":[["11d27a7b.d
19536","809a96c0.f08538"]]},{"id":"11d27a7b.d19536","type":"debug","z"
:"33885e25.21d752","name":"","active":true,"console":"false","complete
":"false","x":472.0937042236328,"y":804.5795593261719,"wires":[]},{"id
":"f0be75c2.ce89c8","type":"inject","z":"33885e25.21d752","name":"Trig
ger
Read","topic":"","payload":"","payloadType":"date","repeat":"","cronta
b":"","once":false,"x":131.09376525878906,"y":799.2727432250977,"wires
":[["46f6c401.9d0b2c"]]},{"id":"53cc89e2.ccd3e8","type":"m2x","z":"338
85e25.21d752","feed":"20311063.16361","name":"Log Lamp
Temp","x":689.0994110107422,"y":864.8722229003906,"wires":[[]]},{"id":
"809a96c0.f08538","type":"function","z":"33885e25.21d752","name":"Form
at for M2X","func":"var fromPAC = msg.payload;\nmsg = {\n    topic:
\"devices\",\n    action: \"setStreamValue\",\n    topic_id:
\"d6b700985c63ac83d4b0159a93c1df8e\",\n    sub_topic_id:
\"LampTemperature\",\n    payload: { \"value\": fromPAC}\n};\n\nreturn
msg;","outputs":1,"noerr":0,"x":448.24147033691406,"y":865.47729492187
5,"wires":[["53cc89e2.ccd3e8"]]},{"id":"4a8d65cb.3a136c","type":"comme
nt","z":"33885e25.21d752","name":"Send Lamp Temperature to M2X Cloud
Storage","info":"","x":215.0994415283203,"y":759.7273254394531,"wires"
:[]}]
```



5. Configure the M2X node for your specific M2X Device's API key and Stream ID:

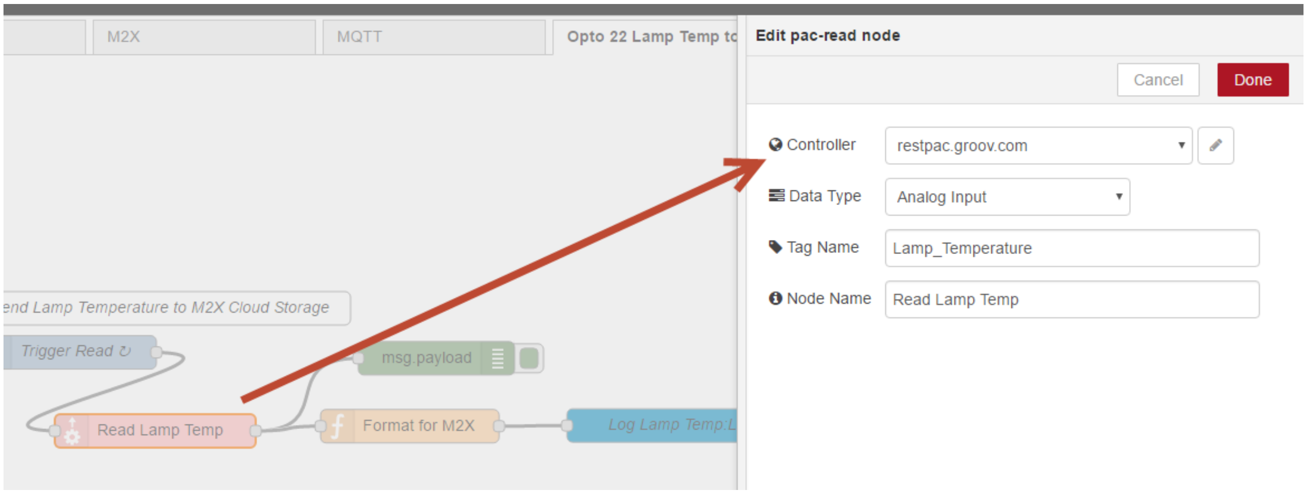


Name: LampTemperature

API key: Your API Key

6. Configure the SNAP PAC read node for the IP address or URL of the Opto 22 SNAP PAC controller.

Note: in this case we used a SNAP-PAC-R1 that happens to be on the Internet at address restpac.groov.com, but typically this would be the IP address of the controller on the same internal network/LAN as the Node-RED server.



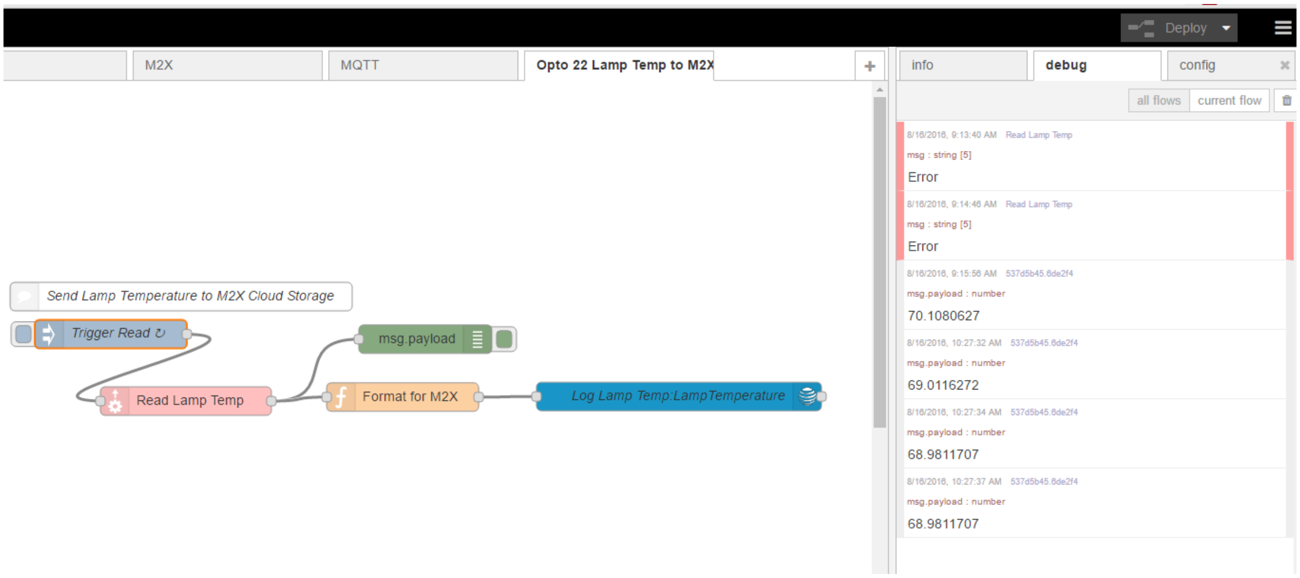
Controller: restpac.groov.com

Data Type: Analog Input

Tag Name: Lamp_Temperature

Node Name: Read Lamp Temp

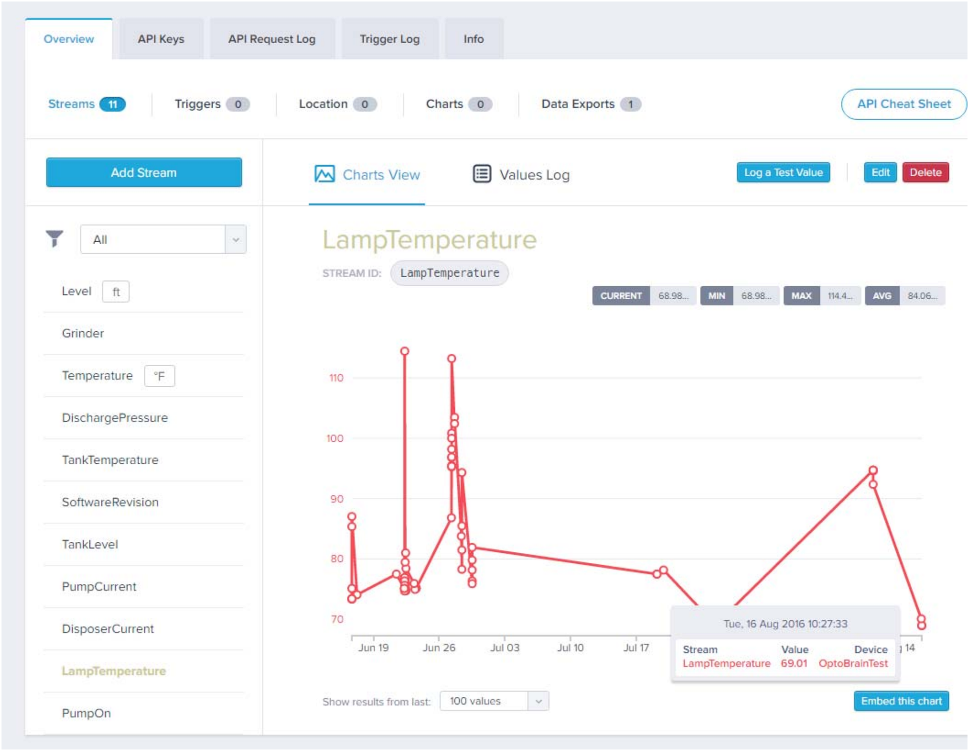
7. Deploy and debug the flow:



```
8/16/2016, 9:13:40 AM   Read Lamp Temp
msg : string [5]
Error
8/16/2016, 9:14:46 AM   Read Lamp Temp
msg : string [5]
Error
8/16/2016, 9:15:56 AM   537d5b45.6de2f4
msg.payload : number
70.1080627
8/16/2016, 10:27:32 AM   537d5b45.6de2f4
msg.payload : number
69.0116272
8/16/2016, 10:27:34 AM   537d5b45.6de2f4
msg.payload : number
68.9811707
8/16/2016, 10:27:37 AM   537d5b45.6de2f4
msg.payload : number
68.9811707
```

3. Check out your data in the M2X cloud!

The data is shown here in the M2X cloud's Charts View:



CURRENT 68.98... MIN 68.98... MAX 114.4... AVG 84.06...

Tue, 16 Aug 2016 10:27:33

Stream	Value	Device
LampTemperature	69.01	OptoBrainTest