Contents

Internet of Things (IoT)

Connecting Raspberry Pi as a Gateway to using Node-RED – Part II

This recipe will help you connect your Raspberry Pi, as a gateway, to the Watson easy wiring approach of Node-RED

Recipes@WatsonIoT

Published on April 22, 2016 / *Updated on May 18, 2017*

Overview

Skill Level: Beginner

BEGINNER

PrerequisiteThis tutorial is continuation of the Part 1 of this recipe. So for the installation and cor follow the steps from the part 1 recipe https://developer.ibm.com/recipes/tutorials/connecting-r gateway-to-watson-iot-using-node-red/ Following steps from the Part 1 recipe must be perform configuration of Watson IoT nodes in Raspberry PiWatson IoT Node on Raspbian JessieStarting N […]

Ingredients

IBM Watson Platform Account and Watson IoT platform organization

Contents

Step-by-step

1 Prerequisite

This tutorial is continuation of the Part 1 of this recipe. So for the installation and configurat the steps from the part 1 recipe https://developer.ibm.com/recipes/tutorials/connecting-ra to-watson-iot-using-node-red/

Following steps from the **Part 1 recipe** must be performed for installation and configuratio Raspberry Pi

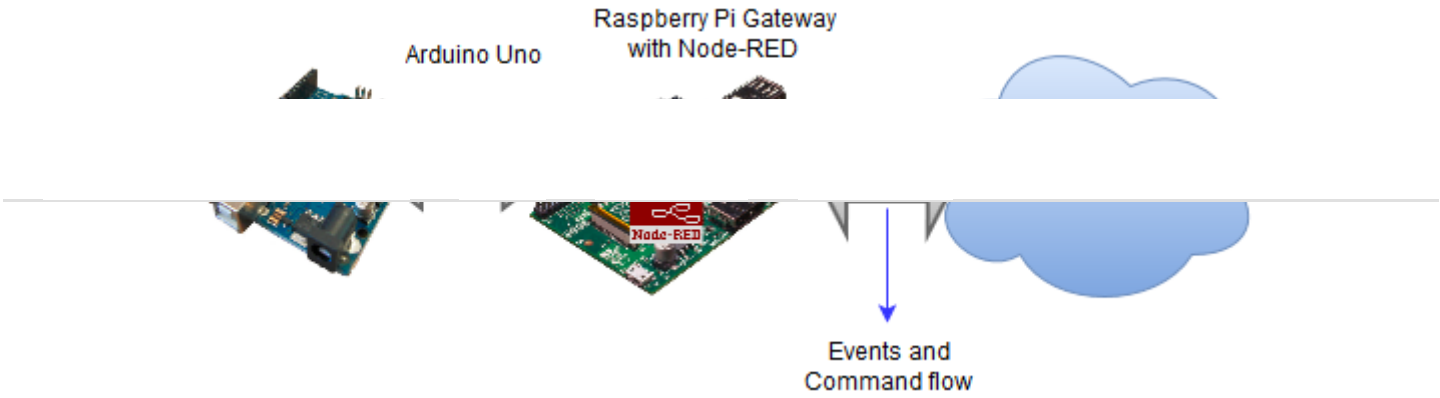
1. Watson IoT Node on Raspbian Jessie
2. Starting Node-RED.
3. Registering your Gateway In Watson IoT Platform

2 Introduction

The **previous tutorial** showcased how to use the Watson IoT node in Raspberry Pi Node-RE gateway to the IBM Watson IoT Platform, send Gateways events and receive Gateway Com

In this recipe, we will learn how to send **events on behalf of a device** from **Gateway** to IBM will learn how to get events from an Ardunio Uno device connected to Raspberry Pi, and se RED running in Raspberry Pi. The Watson IoT Node is a pair of Node-RED nodes for connec Watson Internet of Things Platform as a Device or as a Gateway.
In this recipe, you will learn

1. Connect a external Device to Raspberry Pi. We will use **Arduino Uno** in this tutorial
2. If you have no devices, use a simulator to send the device events.
3. Use Node-RED to read the events.
4. Connect the Raspberry Pi, as a gateway, to the IBM Watson IoT Platform
5. Learn how to send device events on behalf of the device to the platform
6. Learn how toReceive Device commands on behalf of a device and send it to the device.



For more information about the Watson IoT Node please **click here**.

Next, we will first learn how to upload the Sketch code to Arduino Uno and then we will Pi to receive events from Arduino Uno.

3 Configure Arduino Uno

Connect Arduino Uno to your computer using the micro USB cable and upload the sketcl operations.

In order to program Arduino Uno we need to download and setup the **Arduino IDE**. Go thro setting up the Arduino Software (IDE) on your computer.

In this section, you will learn what the sketch program does and upload it to the Ardunio Ur

```
 /**
 * This Arduino Sketch is written to demonstrate the Gateway Support in
 * IBM Watson IoT Platform, and this sketch represents the device code
 * that connects to Raspberry Pi Gateway.
 *
 */

char deviceEvent[30]="";
// LED PIN is 13
const int LEDPIN = 13;

void setup() {
 Serial.begin(9600);
 pinMode(LEDPIN,OUTPUT); // LED actuator
 delay(500);
}
```

```
/**
 * This method does the following,
```

```
 * 2. Sends the temperature status to the Gateway in the following forma
 *
 * { "temp" : n }
 *
 *
 */

void loop() {
 if (Serial.available()) {
 int value = Serial.parseInt();
 // Sometime we see a garbage number, so restrict to a lower number
 if(value > 100) {
 value = 5;
 }
 blink(value);
 }

 strcpy(deviceEvent, "");

 char val[10];
 strcat(deviceEvent,"{"temp":");
 dtostrf(getTemp(),1,2, val);
 strcat(deviceEvent,val);
 strcat(deviceEvent,"}");
 Serial.println(deviceEvent);
 delay(2000);
}

void blink(int n){
 for (int i = 0; i < n; i++) {
 digitalWrite(LEDPIN, HIGH);
 delay(100);
 digitalWrite(LEDPIN, LOW);
 delay(100);
 }
}
```

```
/*
```

```
double getTemp(void) {
 unsigned int wADC;
 double t;
 ADMUX = (_BV(REFS1) | _BV(REFS0) | _BV(MUX3));
 ADCSRA |= _BV(ADEN); // enable the ADC
 delay(20); // wait for voltages to become stable.
 ADCSRA |= _BV(ADSC); // Start the ADC
 // Detect end-of-conversion
 while (bit_is_set(ADCSRA,ADSC));
 // Reading register "ADCW" takes care of how to read ADCL and ADCH.
 wADC = ADCW;
 // The offset of 324.31 could be wrong. It is just an indication.
 t = (wADC - 324.31 ) / 1.22;
 // The returned temperature is in degrees Celcius.
 return (t);
}
```

Functions in Sketch Program

1. **Setup** – It sets up the serial communication and the LED pin mode.
2. **loop** – This function performs 2 actions

   a. 1. listens for serial communication. We will use this to listen for commands from Rasp
   function receives the command, will invoke the blink function with the count.

   b. 2. Sends the temperature event every 2 seconds.

3. **blink** – This function powers the LED by writing on PIN 13 when it receives a non-zero
4. **getTemp** – This function is used to read the temperature sensor value present in the Ar

Upload the Sketch

The complete sketch code is present **here**. Copy the entire Sketch code from the link and p
and upload the code to Arduino Uno.

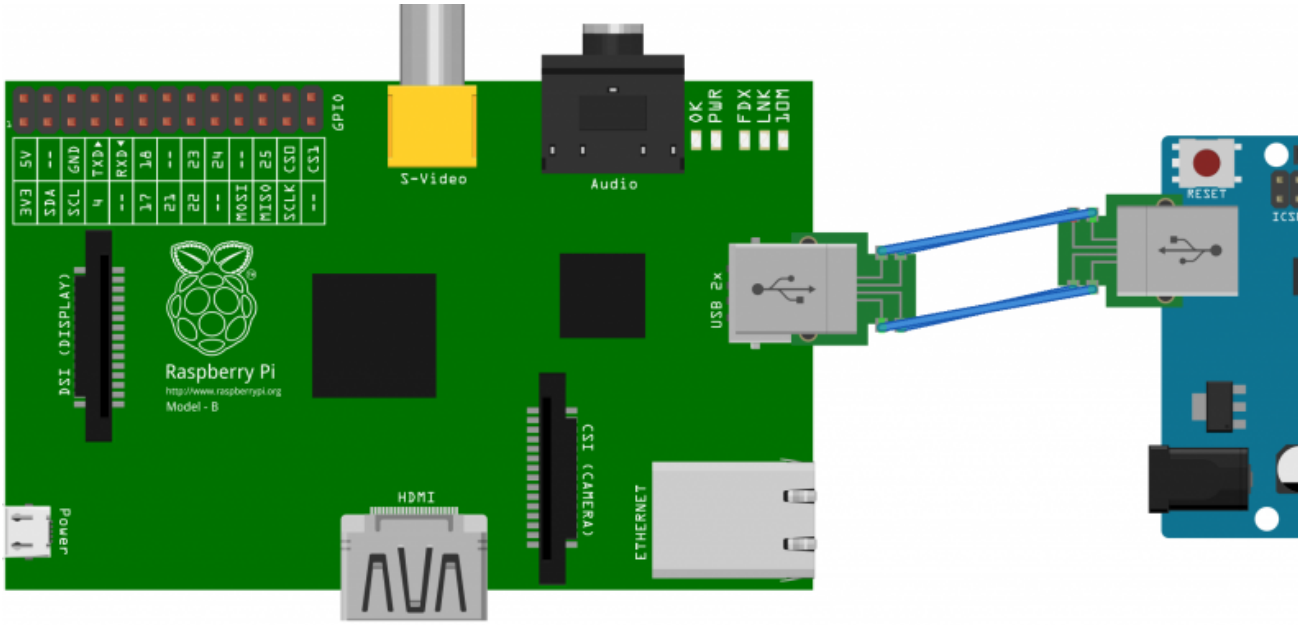
You can verify the code by observing the result in the Serial Monitor. Go to **Tools -> Serial M**
to see the temp event every 2 seconds.

Now we have successfully configured the Arduino Uno. Next we will learn how to connec Raspberry Pi.

4 Connect Arduino Uno to Raspberry Pi

In this step, we will learn how to connect Arduino Uno with Raspberry Pi.

There are many ways of connecting the Raspberry Pi and Arduino, such as using the **GPIO a I2C**. But connecting through the USB cable is one of the easiest way to get them talking, be required is minimal: all you will need is a micro USB cable that comes with the Arduino.

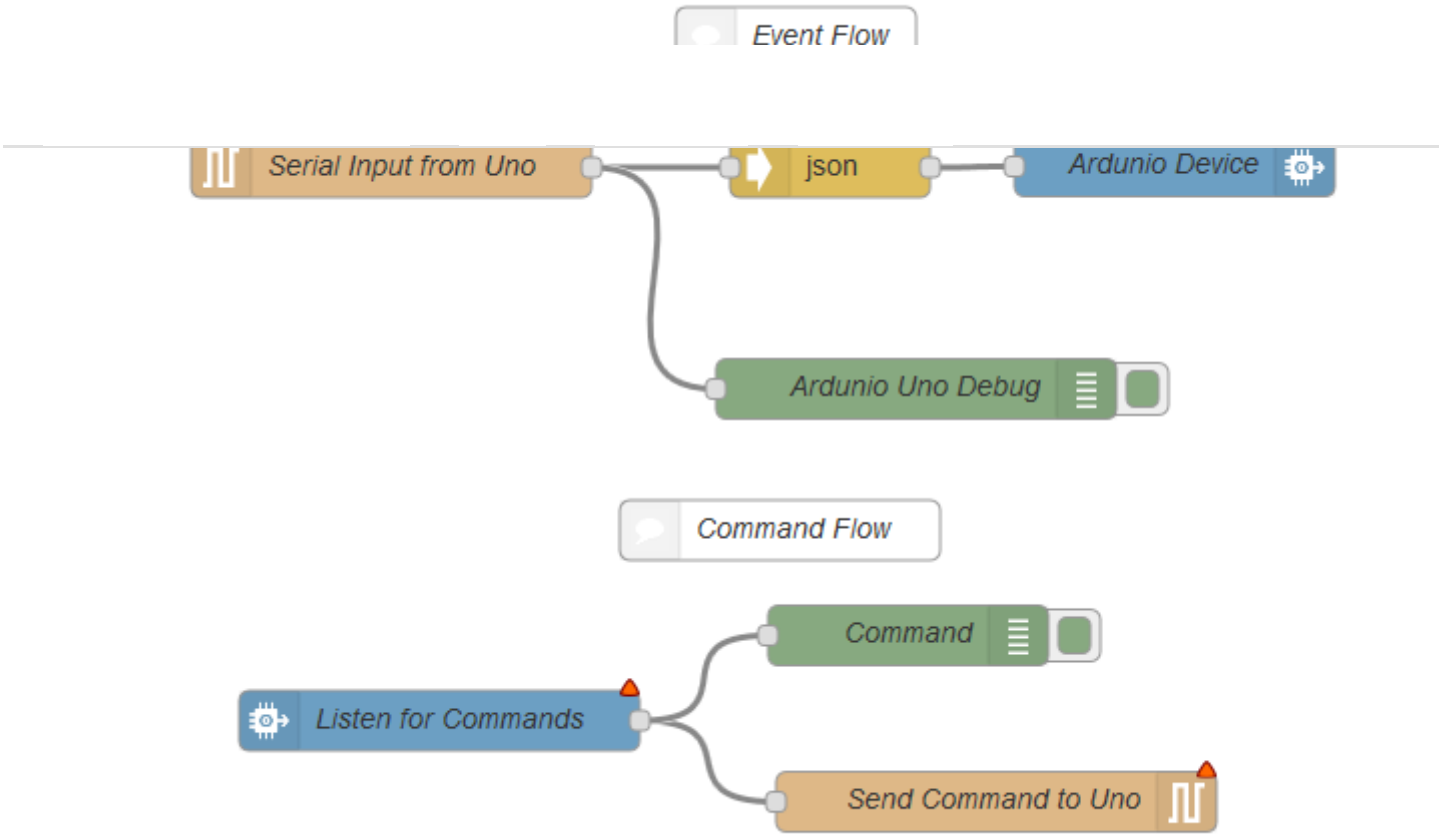


In this step, we have successfully connected Arduino to Raspberry Pi Using the micro US

5 Configuring Node-RED flow

In this step, we will configure the Node-RED flow to setup the Ardunio Uno serial connectic Connection details.

1. Open the Node-RED editor in the browser.
2. Click on **Menu(Top Right Corner) > Import -> Clipboard.**
3. Copy the JSON below and paste it in the clipboard.
   `[{"id":"507cb3dd.2da064","type":"serial in","z":"95e6b532.2f8f18","na`
4. Click **OK**.



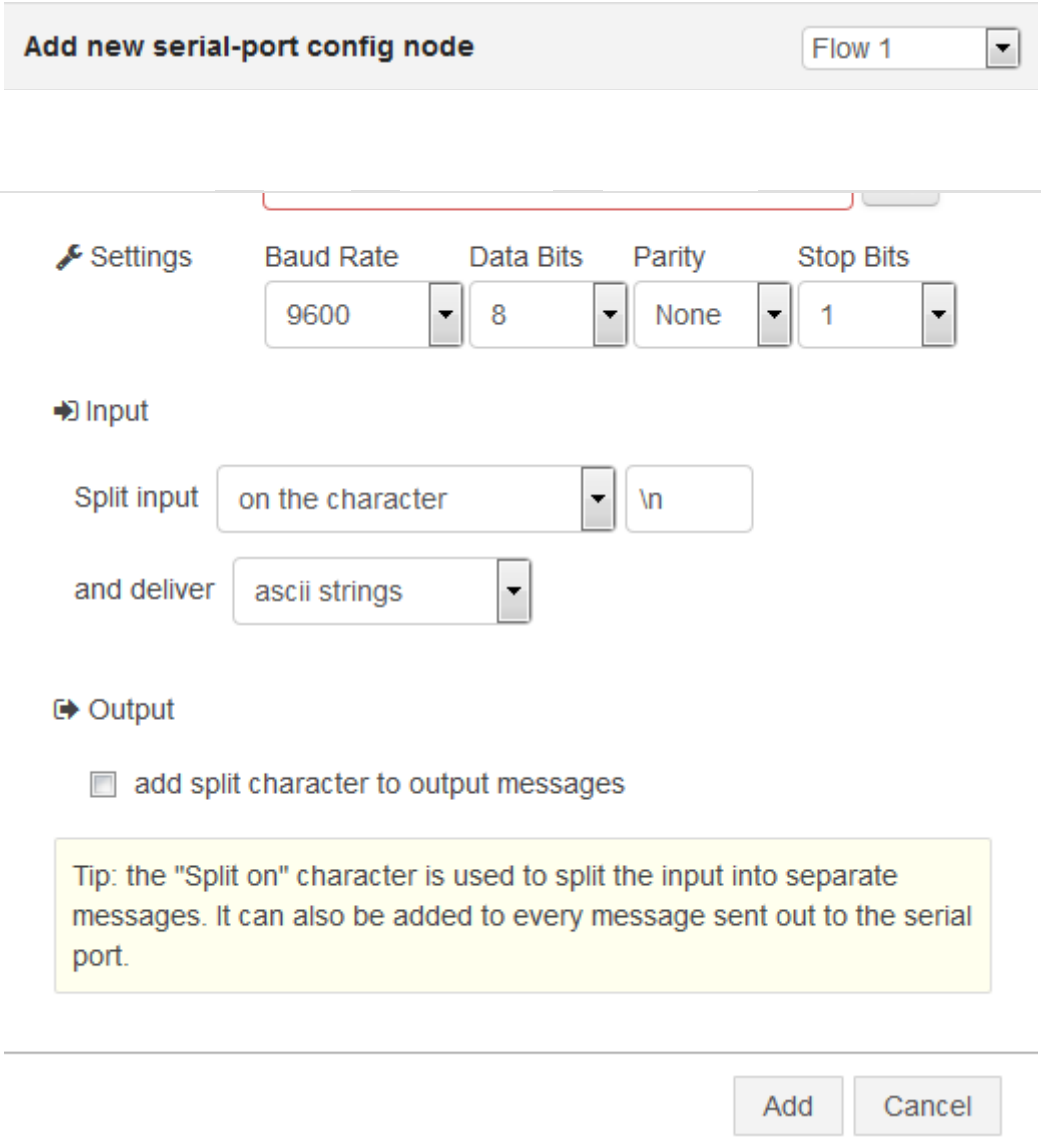
Nodes of Node-RED

1. **Arduino Device / Listen For Commands** – Watson IoT Input and output nodes. Used to subscribe for commands
2. **Serial Input from Uno / Send Command to Uno** – Serial Nodes. Used to read/write fro
3. **Arduino Uno debug / Command** – Debug nodes

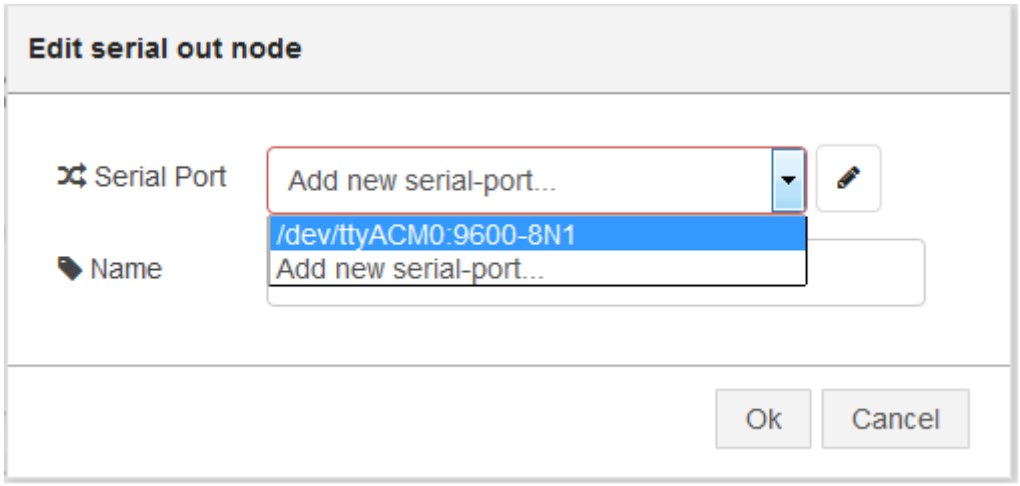
Nodes are still not configured, so you will observe the red triangles on top of the nodes tha

Configure Serial nodes

1. Double Click "**Serial Input from Uno**" node to configure the serial node(Arduino Uno).
2. Click to add a new serial connection in Node-RED
3. Click the **Search** icon to find all the serial ports connected in your Raspberry Pi
4. Select "**/dev/ttyACM0**". (0 can be any number based on the number of serial connectio
5. Select the **Baud rate** to 9600.
6. Click **Add**.

7. 

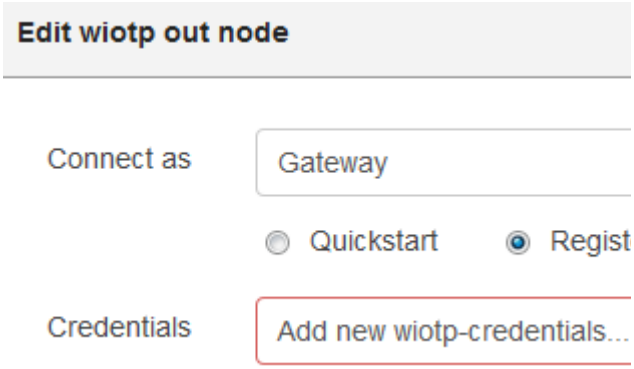
8. Now you have configued the input serial node for your Arduino Uno. Next we will config
9. Double Click "**Send Command to Uno**".
10. In **Serial Port** field, you can select the configuration we added in the previous step.



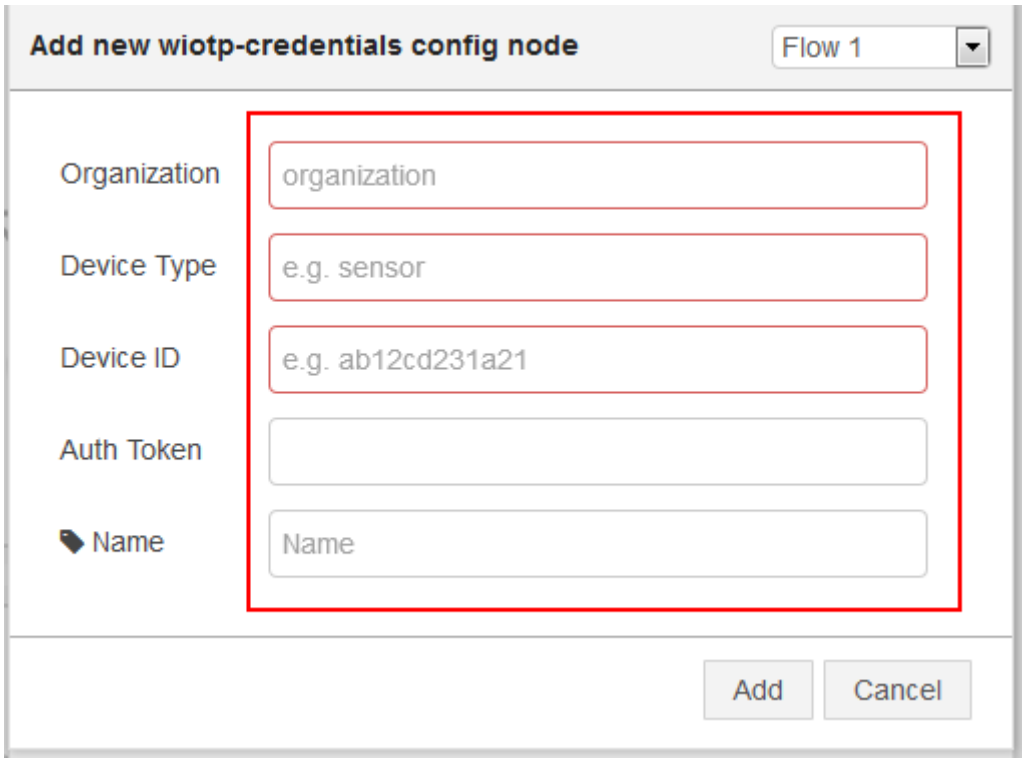
11. Click **OK**

Now you will observe that the red triangle on the serial nodes disappeared as we have succ nodes.

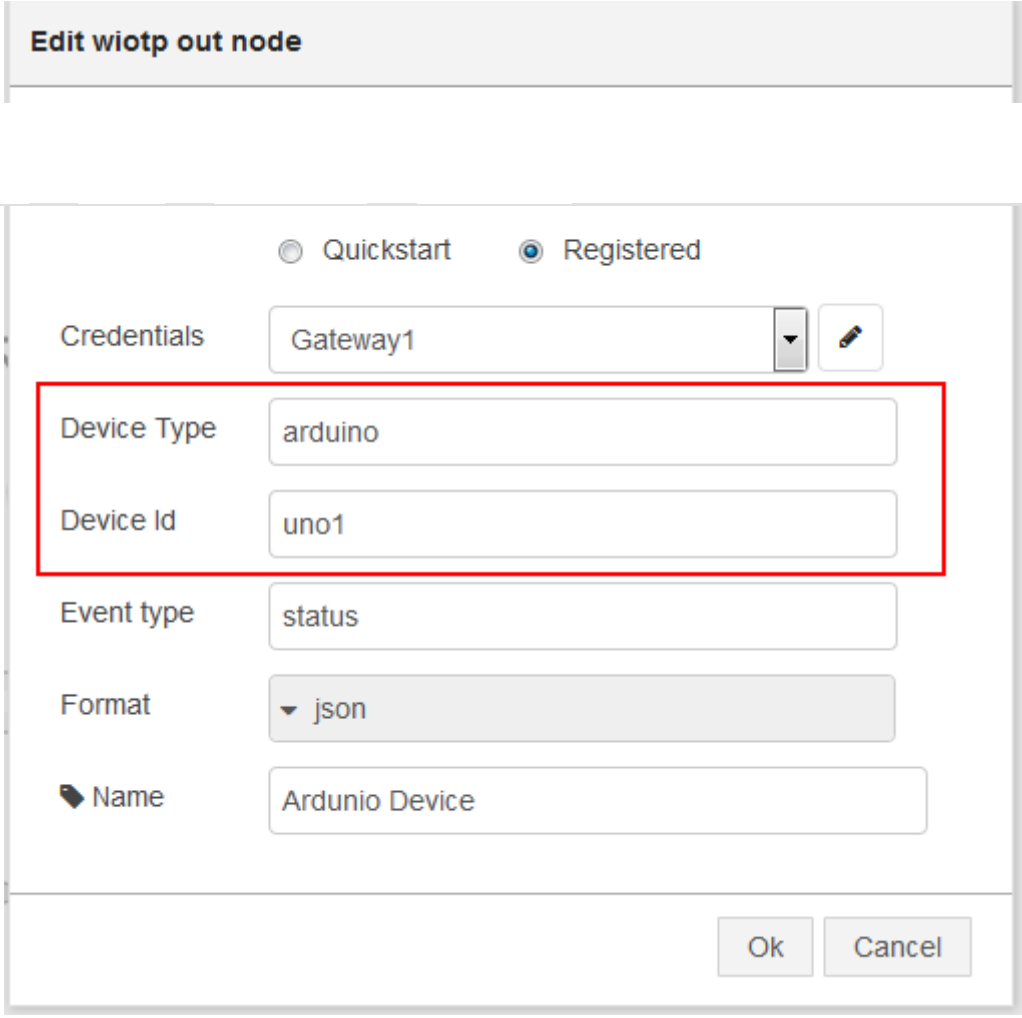
Configure Watson IoT Nodes

2. Select "**Gateway**" in Connect Field.
3. Select connection as **Registered** .
4. 

   Select Add button to add new Credentials.
5. Enter your credentials details for your Gateway and Click **Add**.



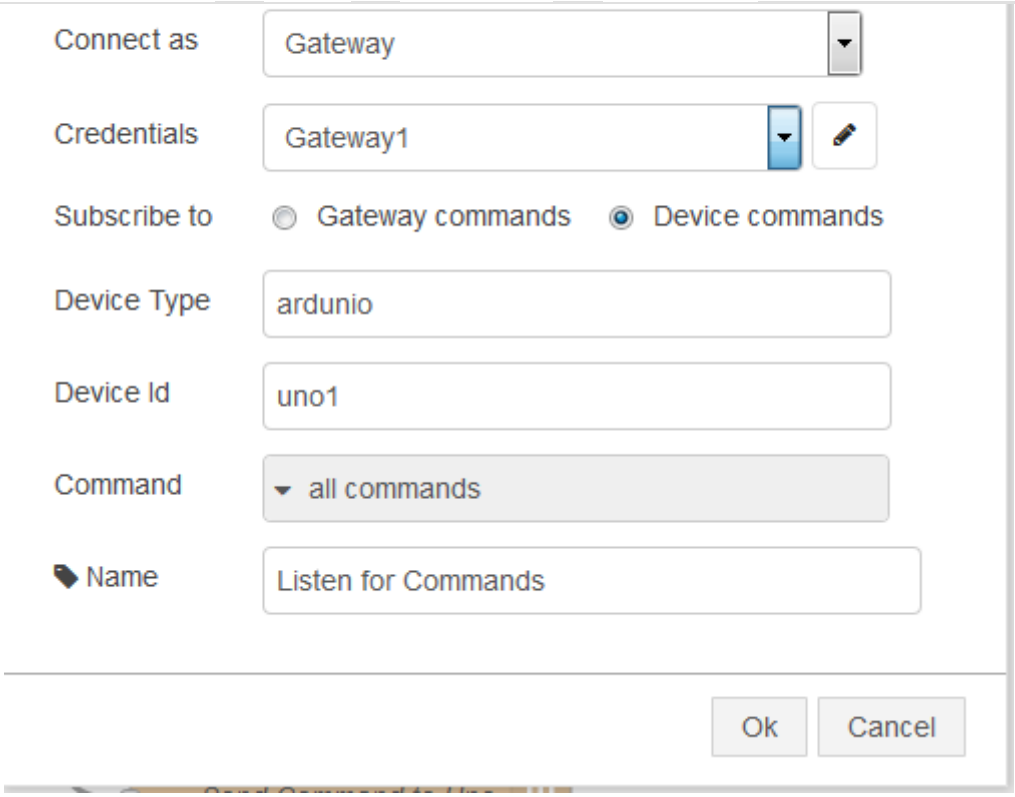
6. Now we have configured the Gateway. Next we will configure the Device
7. Enter the Type and ID for your Ardunio Uno device. In this tutorial, we will use it as "arc
   respectively. **Note:** The Watson IoT platform will auto-register this device for you basec



ID of device.

8. In the **Event Type**, enter the type of event.
9. Now we have successfully configured the Output Watson IoT(events) node. Next lets cc node(Commands)
10. Double Click "**Listen for Commands**" Node.
11. Select **Connect As** "**Gateway**".
12. Select the **Credentials** we created in the previous step.
13. Select the **Device Commands** radio button.

14. Enter the **same Device Type and Device ID** we entered in the previous step.

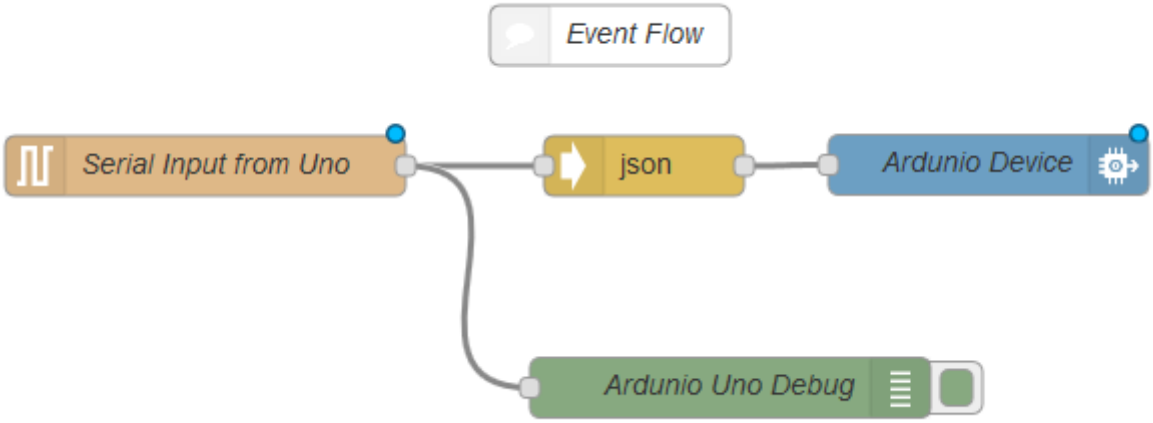


15. Now we have configured both the serial and Watson IoT nodes.

16. Click on **Deploy** to save the flow and activate it.

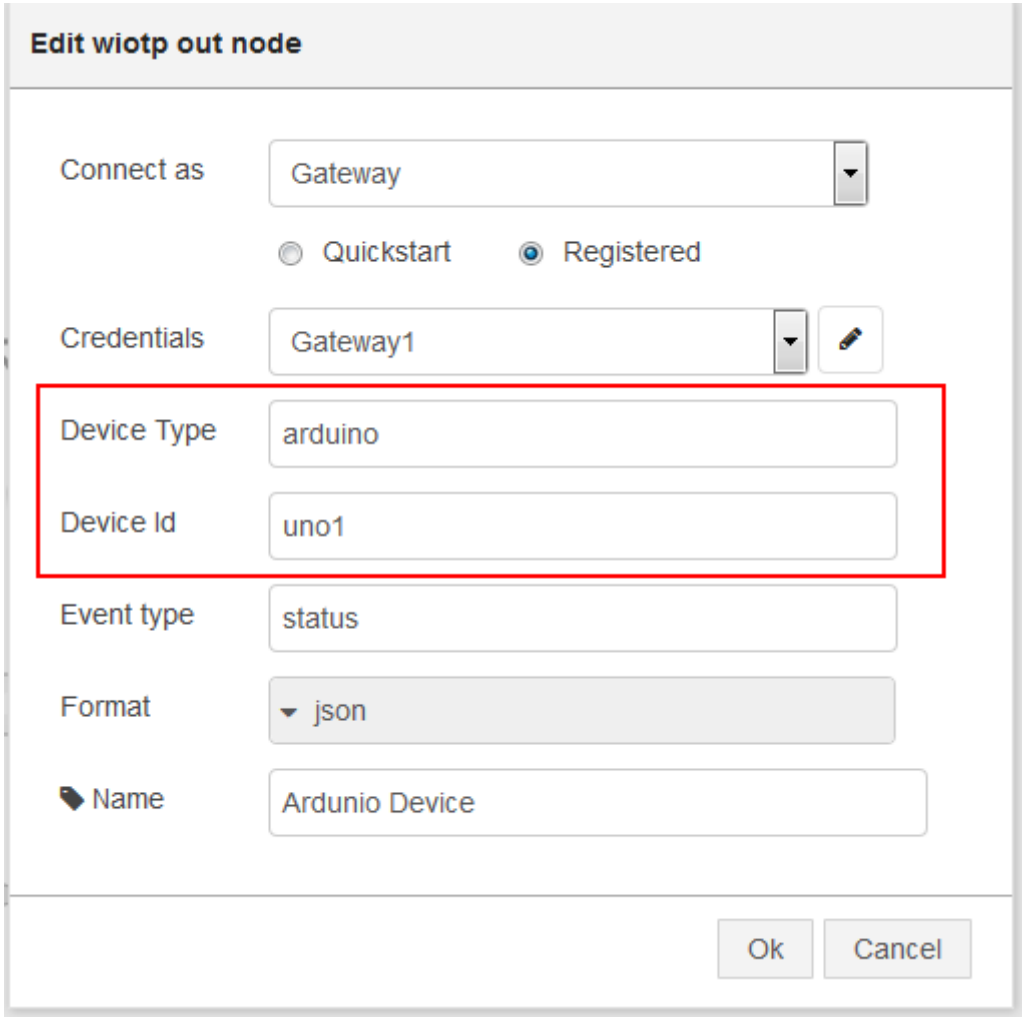
Now Node RED nodes are successfully configured. Next we will learn how the nodes pub to commands

6 Sending Device(Arduino Uno) Events to Watson IBM IoT P



This flow read the events from Arduino Uno and uses the Gateway option of the Watson IoT events to IBM Watson IoT Platform.

1. **Serial Input from Uno** read the serial inputs from Arduino Uno and passes it to the flow

1. 

a.

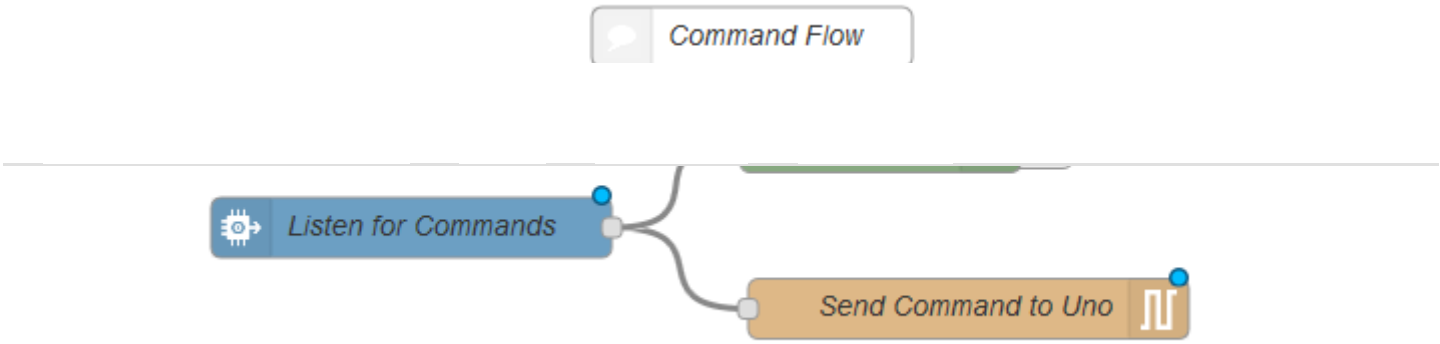
b. 2. Now this Gateway will publish data on behalf of the Arduino Device.

c. 3. It publishes the event data as the Device Type "**arduino**" and Device Id "**uno1**".

This flow sends the temperature sensor data to the platform every 2 seconds as configurec

Now you have learnt how to send device events using a Gateway. Next we will learn how commands on the Gateway.

7 Receiving Device(Arduino Uno) Commands From IBM Wat



In this flow, the Gateway subscribes to device commands on behalf of Arduino Uno.

In this tutorial, the node-RED flow will send a integer representing the number of times the
uno.

1. **Listen for Commands** node subscribes for commands from Watson IoT. When an IoT a
command for "**uno1**" device, it will be received in this node.
2. This flow will route the message to the **Send Command to Uno.** This will write to the se
3. The Arduino program listens for serial input and blinks based on the integer received.

Now you have successfully received commands on the Gateway.

8 Conclusion

We have seen how to deploy Watson IoT Node in the Raspberry Pi as a gateway and send d
device, to the IBM Watson IoT Platform and receive device commands from the Platform u
programming.

As a next step, look at the following recipes that might be helpful in building an end to end

- Visualize device events in Watson IoT Platform – Showcases how to use the cards in th
dashboard to visualize the device data in real-time.
- Real-time data analytics in Watson IoT Platform – Showcases how to create rules and a
real-time.
- Device diagnostics and management in Watson IoT Platform – Showcases how to diagr
Perform Updates & Reboots and automate the whole process on the Watson IoT Platfo

Please leave your feedback/queries on the comments section below.

TAGS **APPLICATION, ARDUINO, DEVICE, GATEWAY, IBM WATSON IOT PLATFORM, NODE-RED, NODE.JS, NODEJS, RASPBERRY P**
WATSON IOT

by Recipes@WatsonIoT

2 comments on"Connecting Raspberry Pi as a Gateway to Watso RED – Part II"

Pigio • September 04, 2016

Hi, thank you for this recipe.
Blocks in Node-red has been changed. For example in step 4 in the picture is possible to select gateway doesen't exist but this is not a big deal.
"Enter the Type and ID for your Ardunio Uno device. In this tutorial, we will use it as "arduino" and "uno1' IoT platform will auto-register this device for you based on the Values of Type "
I create a flow in Node-red and i can send JSON msg to "IBM Watson IoT Platoform", I registered only on to IoT platform without problems (and i see them).
Now i want to send messages from a device that is not registered (but I have registered before its "device gateway can auto-register devices.
I try to implement this concept but in WIoTP i don't se the "auto-registration" of this device,so I think tha properly, in fact I don't see in WIoTP the messages that i send by this new not-previously registered devic
I define a function node that set the msg.deviceId property (and other) with a new Id not yet registered ir that I define before).
var msg={

'payload': msg.payload,

'format': "json",

'deviceId': "NewIdNotregistered",

'deviceType': "Type1",

'eventOrCommandType' : "PR"
};

return msg

After this node i put an ibmioT out node.
here I set
-authentication: API KEY
-API KEY: the gateway credentials (obtained by WIoTP)
-Output type: Device event

Those characteristics are overrided so not used:
-Device Type IG_OVERRIDED

-Format json_OVERRIDED
-Data Overrided
-QoS 0
-Name Iotout

Where is the error? where I see devices that gateway auto-register?

Waiting for a reply

P.

Log in to Reply

Aby123 • March 17, 2017

Nice topic.
I would like to transfer one file in to a generator, it has one usb port only. The file is present in raspberry p
in to generator using node red.
The generator used to power the surgical equipment used for surgeries. It has USB port and serial id.We
this generator , presently we are manually copying the file from raspberry pi and connecting to generator
automatically we need to update software file from raspberry pi os.
We have the software file in raspberry that should reach at generator level through usb port . Is it possibl

Log in to Reply

Join The Discussion

You must be logged in to post a comment.

Contact Privacy Terms of use Accessibility Report Abuse Cookie Preferences Feed