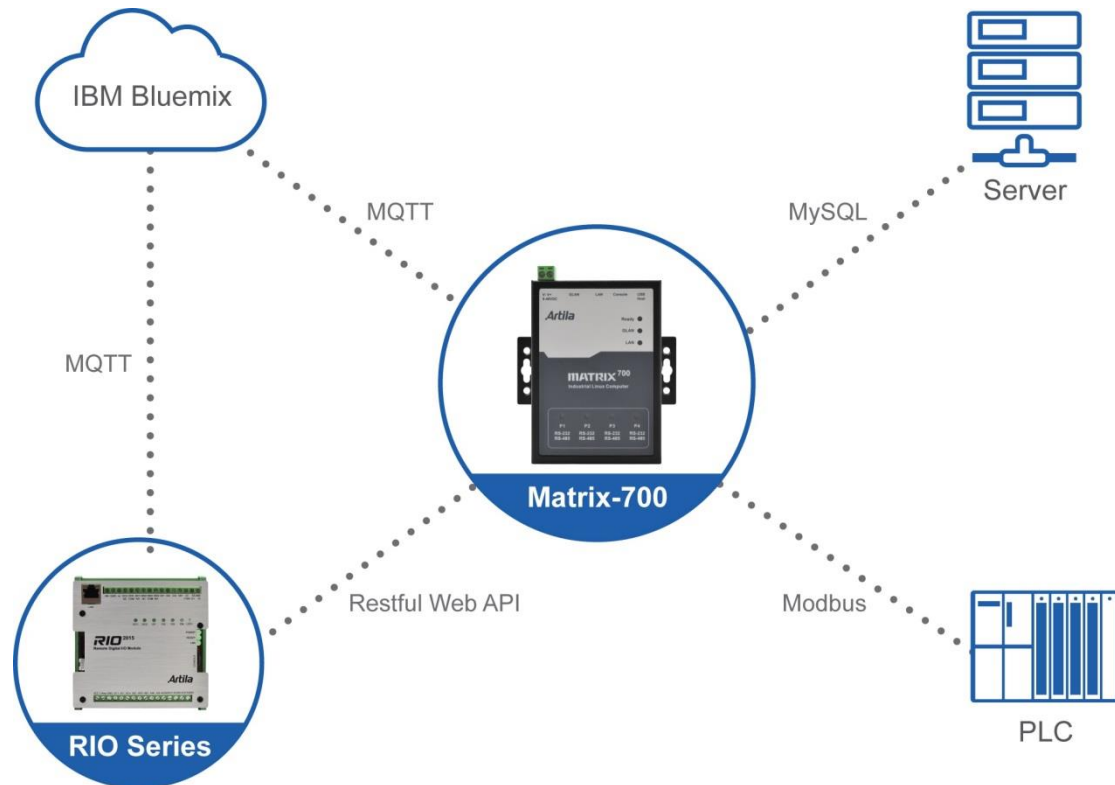


Artila's IoT gateway connects industrial devices and proactively collects the real time data, buffers it and then push it to cloud or database server. The field data can also be exchanged between I/O device and PLC cyclically by using the IoT gateway via standard Modbus, MQTT and RESTful web API. In addition, if the I/O device is smart enough, it can also push the sensor data to cloud by MQTT.



To avoid tedious low level programming in this kind of task, we use Node-RED, the open graphic programming IoT flow with nodes to reduce time of development.

To connect the Things to Internet, the IoT gateway needs to establish a configurable connection between devices. Artilla uses Node-RED for this task, a very flexible visual wiring editor for the Internet of Things.

Node-RED was developed by IBM, it is a web-based graphical tool with an intuitive user interface for wiring node with application data flow. Nodes are wired by means of a mouse with drag and drop.

Node-RED is based on node.js a platform independent runtime environment to develop web applications with server side JavaScript. You can find more information regarding Node-RED. (<http://nodered.org> current manual <http://nodered.org/docs> getting started <http://nodered.org/docs/getting-started>)

To install Node-RED in your IoT gateway, you can use commands

```
apt-get update
```

to update the repository and use

```
apt-get install nodejs-npm node-red
```

to install npm and red-red.

Use

```
apt-get install gcc g++ gcc-dev gcc-symlinks g++-symlinks binutils
```

to install native C compiler and use

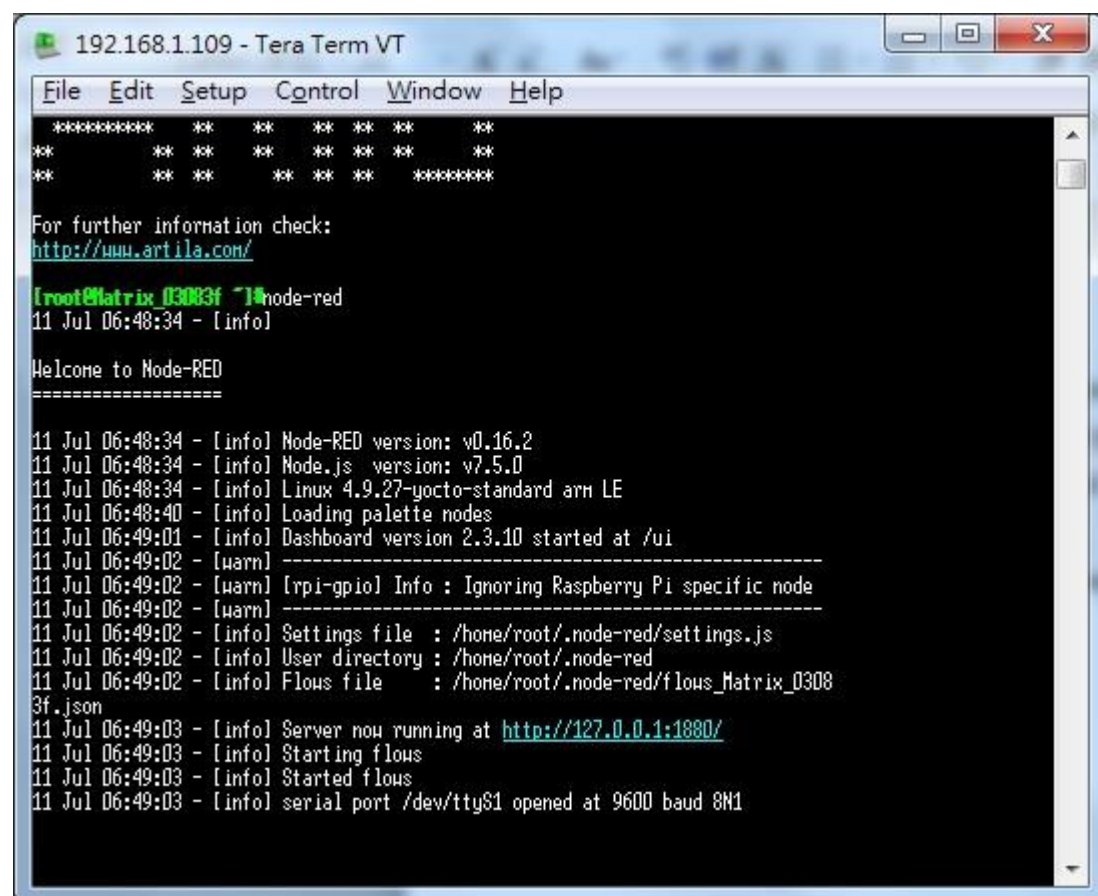
```
npm install -g node-red-dashboard node-red-node-serialport --unsafe --perm
```

to install dashboard and serial port Nodes.

now you can use command

```
node-red
```

to start Node-RED



```
192.168.1.109 - Tera Term VT
File Edit Setup Control Window Help
***** ** ** ** **
**      ** ** ** **
**      ** ** ** **

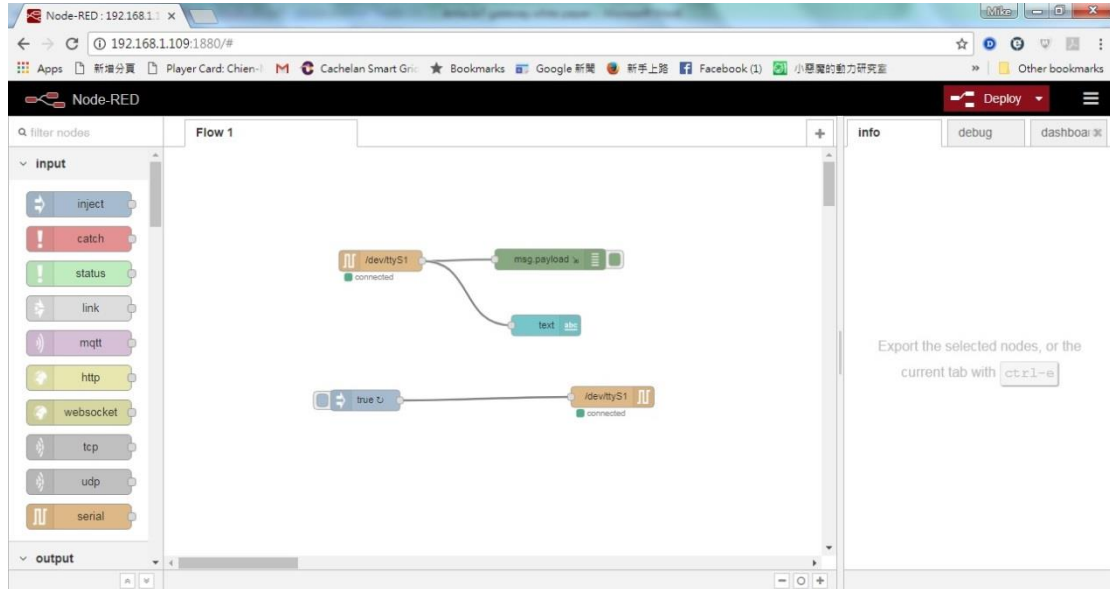
For further information check:
http://www.artila.com/

[root@Matrix:03083f ~]# node-red
11 Jul 06:48:34 - [info]

Welcome to Node-RED
=====

11 Jul 06:48:34 - [info] Node-RED version: v0.16.2
11 Jul 06:48:34 - [info] Node.js version: v7.5.0
11 Jul 06:48:34 - [info] Linux 4.9.27-yocto-standard arm LE
11 Jul 06:48:40 - [info] Loading palette nodes
11 Jul 06:49:01 - [info] Dashboard version 2.3.10 started at /ui
11 Jul 06:49:02 - [warn] -----
11 Jul 06:49:02 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
11 Jul 06:49:02 - [warn] -----
11 Jul 06:49:02 - [info] Settings file : /home/root/.node-red/settings.js
11 Jul 06:49:02 - [info] User directory : /home/root/.node-red
11 Jul 06:49:02 - [info] Flows file   : /home/root/.node-red/flows_Matrix_0308
3f.json
11 Jul 06:49:03 - [info] Server now running at http://127.0.0.1:1880/
11 Jul 06:49:03 - [info] Starting flows
11 Jul 06:49:03 - [info] Started flows
11 Jul 06:49:03 - [info] serial port /dev/ttyS1 opened at 9600 baud 8N1
```

It will take about three minutes to start Node-RED in your gateway. Once done, you can use browser to visit <https://localhost:1880> where localhost is the IP address of your gateway. Now you can see the flow editing window on screen as follow:

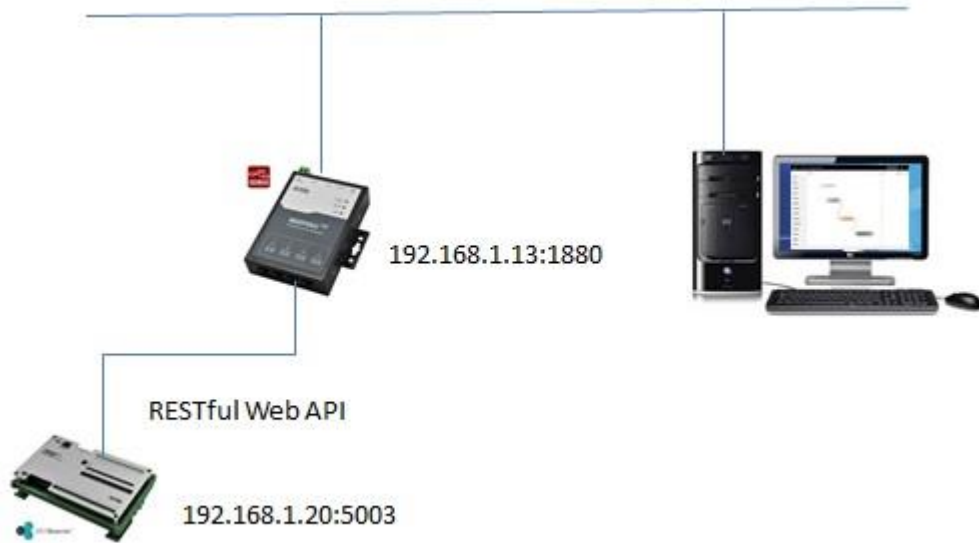


A node can have input and/or output and can connect with other nodes. Data transport between nodes is called *msg* object. The *msg* object contains two properties: *.topic* and *.payload*.

The following categories of node exist in Node-RED:

- Input
- Output
- Function
- Social
- Storage
- Analysis
- Advanced
- Dashboard

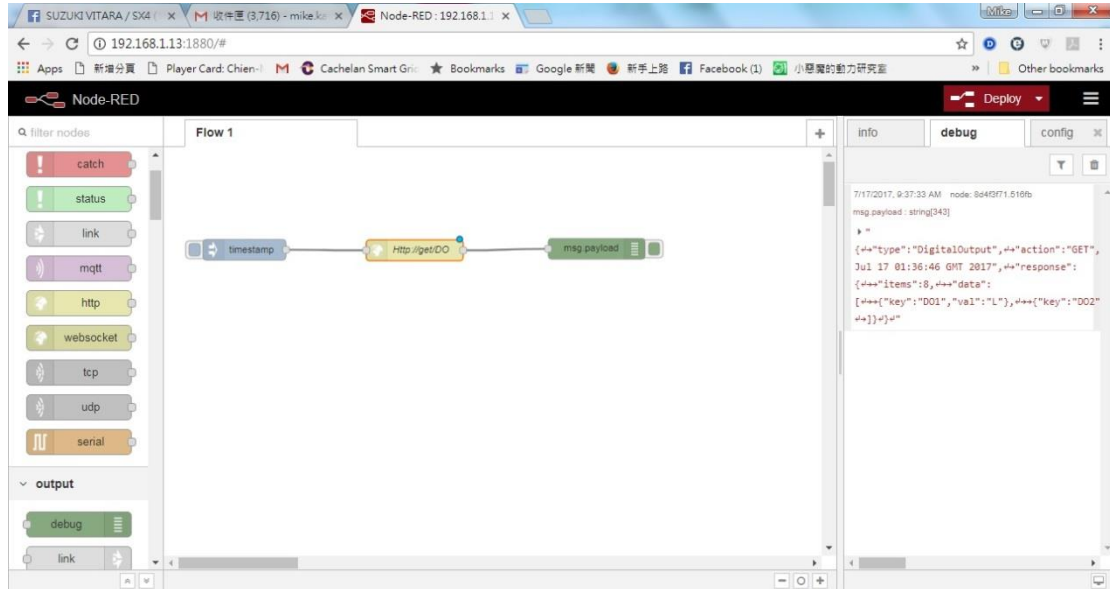
A demo flow of Node-RED is constructed as the following diagram. A Matrix-700 which is pre-installed Node-RED connects to a RIO-2010BM and access the remote I/O via a RESTful Web API. Therefore the computer can use Web browser to configure and run Node-RED flow.



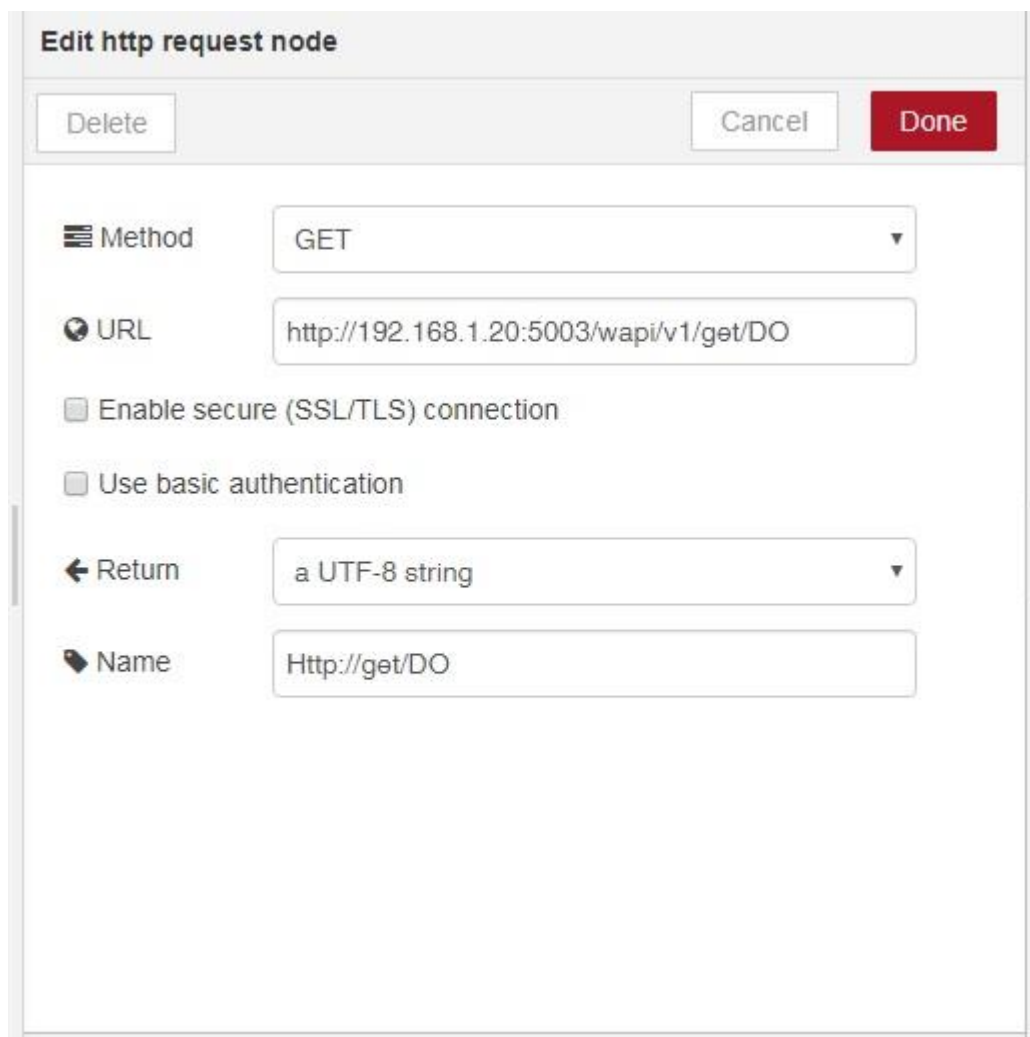
Node-RED uses `http` input Node and method `GET` to get the digital output data of RIO-2010BM. The Web API is as followed:

```
{
  "type": "DigitalOutput",
  "action": "GET",
  "date": "Mon Sep 05 21:14:17 GMT 2016",
  "response": {
    "items": 8,
    "data": [
      {"key": "D01", "val": "L"},
      {"key": "D02", "val": "L"},
      {"key": "D03", "val": "L"},
      {"key": "D04", "val": "L"},
      {"key": "D05", "val": "L"},
      {"key": "D06", "val": "L"},
      {"key": "D07", "val": "L"},
      {"key": "D08", "val": "L"}
    ]
  }
}
```

The response data is in JSON format.



Drag and connect *injet*, *http*, and *debug* Nodes and edit *http* Node as follow:



Click *Deploy* button, you can get the digital output channel status of the RIO-2010BM

and output to the `debug` Node.

There are more Nodes and Flows available at <http://flow.nodered.org>

Find the Node you need and install it to your IoT gateway by `npm`

Install

Run the following command in the root directory of your Node-RED install

```
npm install node-red-contrib-modbus
```

Run the following command for global install

```
npm install -g node-red-contrib-modbus
```

try these options on npm install to build, if you have problems to install

```
--unsafe-perm --build-from-source
```

Please refer to <http://nodered.org/docs> for more features of Node- RED.