ADVANTECH Enabling an Intelligent Planet

Advantech AE Technical Share Document

Date	2019/9/16	**SR#**	1-3968946018
Category	□FAQ ■ SOP	**Related OS**	N/A
Abstract	IAG_FAQ_ADAM-6700_Node-red application tutorial & Example		
Keyword	Node-RED, Modbus TCP, ADAM-6700, Azure, MQTT, SQL		
Related Product	ADAM-6700 series		

■ Problem Description:

Node-RED is a visual tool for wiring the Internet of Things. It's built on Node.js and support browser-based flow editing. User can download different Node-RED library to develop their own IoT application.

This document will introduce and categorize different Node-RED application with explanations and provide the sample flow for user reference.

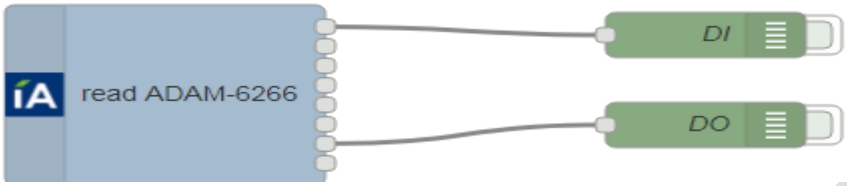
■ Application of Node-RED

Application	Node-Red Sample flow
Data Upload via different protocol	✓ Update data via Modbus ✓ Update data via MQTT
Data upload to different platform	✓ Update data to Azure ✓ Update data to SQL database ✓ Update data to Remote FTP server
I/O extension and application	✓ AI Alarm trigger DO ✓ Extend IO with ADAM-4000/6000 series
Data processing	✓ Analyze the data , scaling, calculation ✓ Data visualization by dashboard
Others	✓ Modbus TCP server ✓ Data logger function

1 Data Upload via different protocol

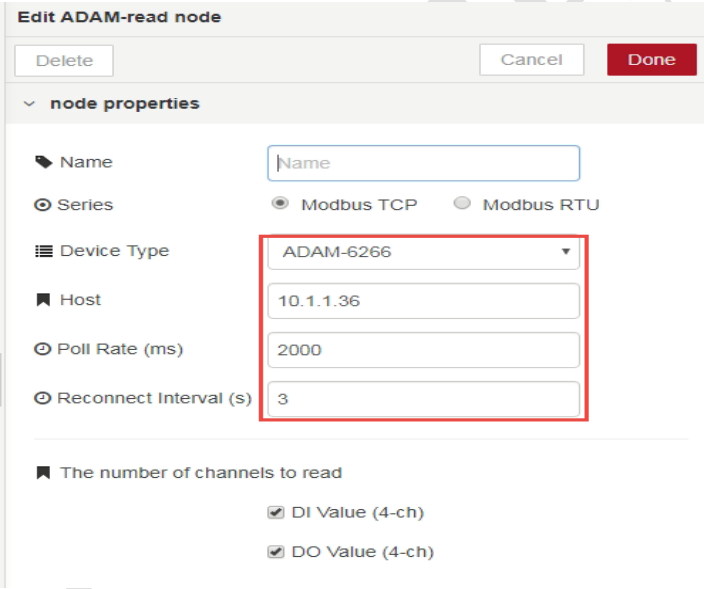
1.1 Update data via Modbus

Node-red flow example:

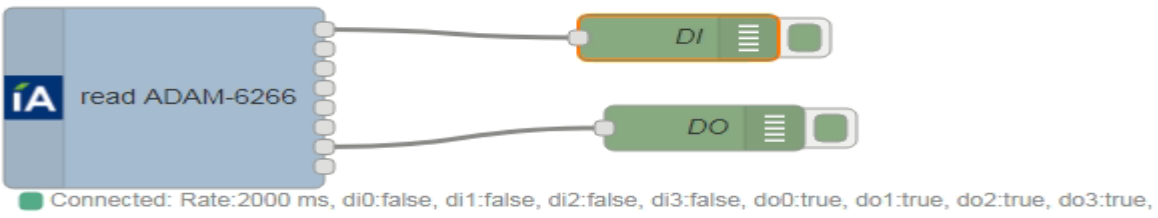


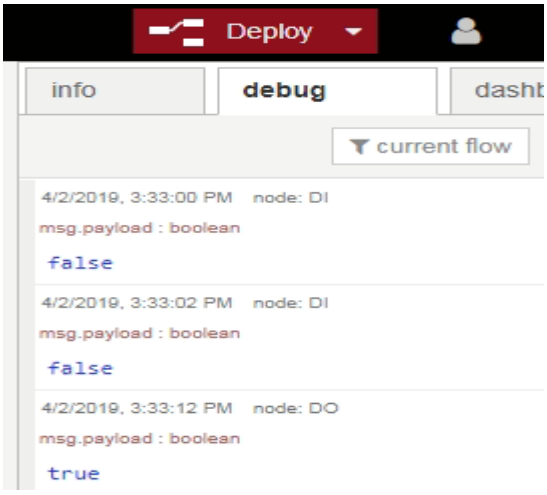
Used nodes & Steps

- Step 1: Drag above node and connect them to a flow
  - ADAM read Node: Select the module you are going to read Modbus data, Enter Module's IP, polling rate, and reconnect interval in this node
  - Debug Node: For printing the message.



- Step 2: See the result on debug column or node status
  - Node status: If connected to ADAM successfully, it will show connected also print the I/O status of each channel
  - Debug column: It will print the message of your debug node





Sample flow

[{"id":"5573d4e0.33673c","type":"ADAM-read","z":"2f8162d5.a284fe","name":"","rate":2000,"host":"10.0.0.60","serialPortCfg":"","unit_id":"","readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-6266","advDevType":"ADAM-6266","x":120,"y":180,"wires":[["a16bcbc2.bc8818"],[],[],[],[],[],["dca64651.9fa088"],[]]},{"id":"a16bcbc2.bc8818","type":"debug","z":"2f8162d5.a284fe","name":"DI","active":false,"console":"false","complete":"payload","x":390,"y":140,"wires":[]},{"id":"dca64651.9fa088","type":"debug","z":"2f8162d5.a284fe","name":"DO","active":false,"console":"false","complete":"payload","x":390,"y":200,"wires":[]}]

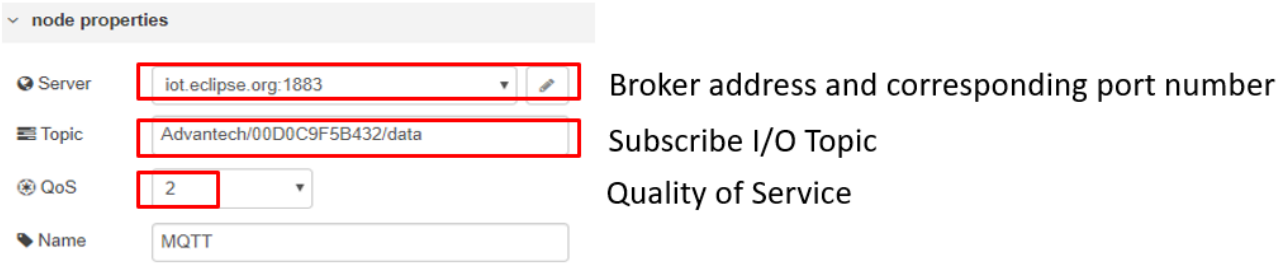
1.2 Update data via MQTT

Node-red flow example:

Used nodes & Steps

Step 1: Drag below node and connect them to a flow

- MQTT Node: Getting data from MQTT Broker
- Debug Node: For printing the message.



Step 2: See the result on debug column or node status

- Node status: If connected to broker successfully, it will show connected
- Debug column: It will print the message of your debug node

Sample flow

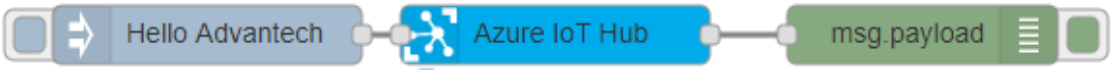
[{"id":"39a91837.2af6b8","type":"mqtt in","z":"c36940f4.544f","name":"","topic":"Advantech/00D0C9F7230E/data","qos":"2","broker":"c716450e.a0e668","x":239,"y":1000.000072479248,"wires":[["f31978b.8c1a488"]]},{"id":"f31978b.8c1a488","type":"debug","z":"c36940f4.544f","name":"","active":true,"console":"false","complete":"false","x":430,"y":1100,"wires":[]},{"id":"2a6ba98a.bf8156","type":"mqtt in","z":"c36940f4.544f","name":"","topic":"Advantech/00D0C9FAC3EA/data","qos":"2","broker":"830ddbaa.dcf0c8","x":210,"y":1240,"wires":[["5dd118ee.c9bd88"]]},{"id":"5dd118ee.c9bd88","typ

e":"debug","z":"c36940f4.544f","name":"","active":true,"console":"false","complete":"false","x":580.0234375,"y":1211.0390625,"wires":[]},{"id":"c716450e.a0e668","type":"mqtt-broker","z":"","broker":"iot.eclipse.org","port":"1883","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos":"0","birthPayload":""},{"id":"830ddbaa.dcf0c8","type":"mqtt-broker","z":"","broker":"iot.eclipse.org","port":"1883","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos":"0","birthPayload":""}]

2 Data upload to different platform

2.1 Update data to Azure

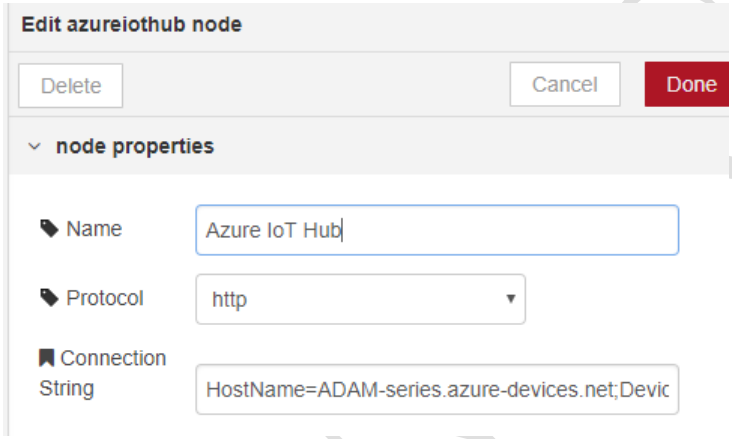
Node-red flow example:



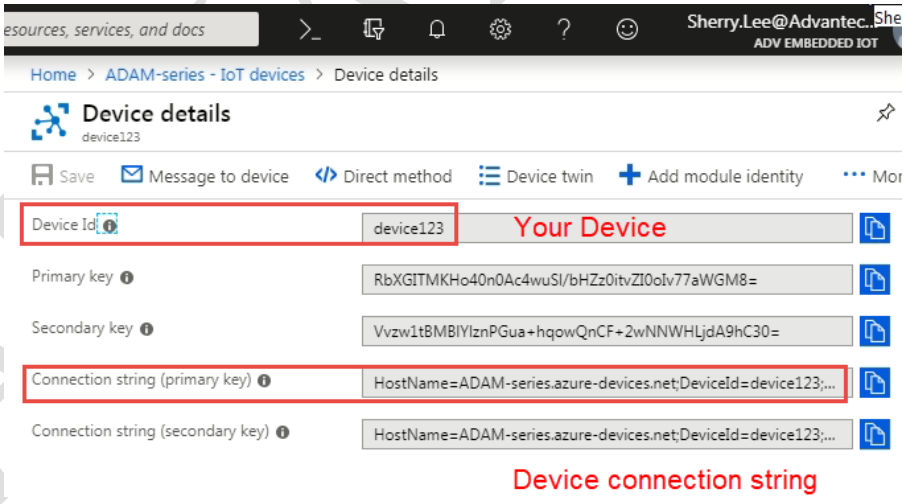
Used nodes & Steps:

Inject Node: Enter “Hello Advantech”

Azure Node: Enter the connection string of your “IoT device” created on Azure.

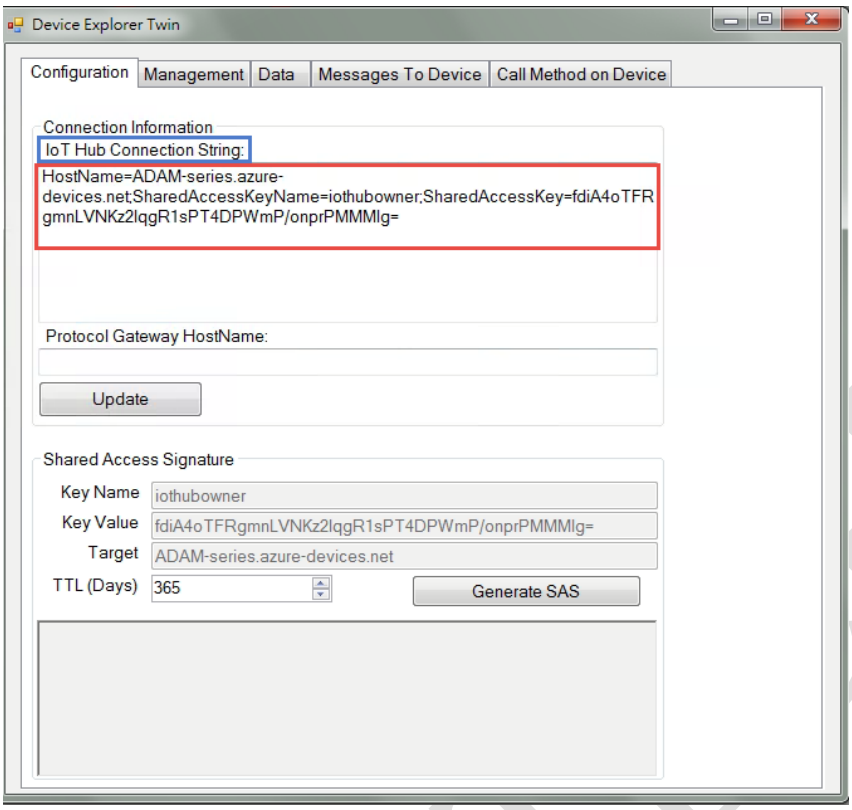


You can find the device connection string as below picture on Azure

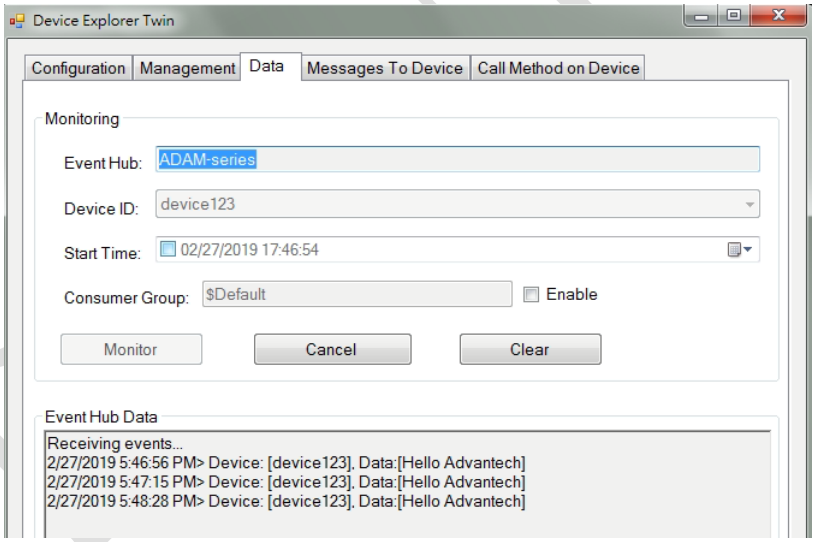


Use device explorer to check

Update the IoT Hub connection String on Device explorer



See the send data from Azure node on Device explorer



Note: SOP for create IoT device on Azure

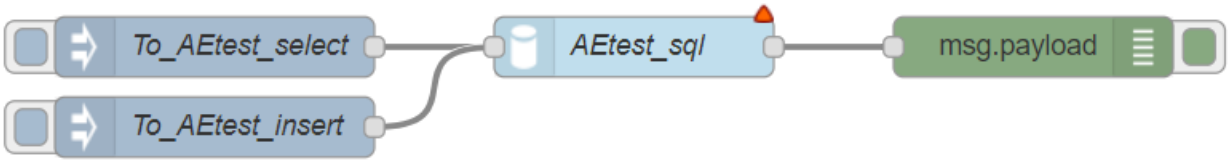
http://forum.adamcommunity.com/viewthread.php?tid=96451&extra=page%3D1

Sample flow

[{"id":"64652724.c72748","type":"azureiothub","z":"c36940f4.544f","name":"Azure IoT Hub","protocol":"http","x":1000,"y":140,"wires":[["2239a513.ee473a"]]},{"id":"2239a513.ee473a","type":"debug","z":"c36940f4.544f","name":"","active":true,"console":"false","complete":"false","x":1230,"y":140,"wires":[]},{"id":"844dd784.9e9f48","type":"inject","z":"c36940f4.544f","name":"","topic":"","payload":"Sherry","payloadType":"str","repeat":"","crontab":"","once":false,"x":810,"y":140,"wires":[["64652724.c72748"]]}]

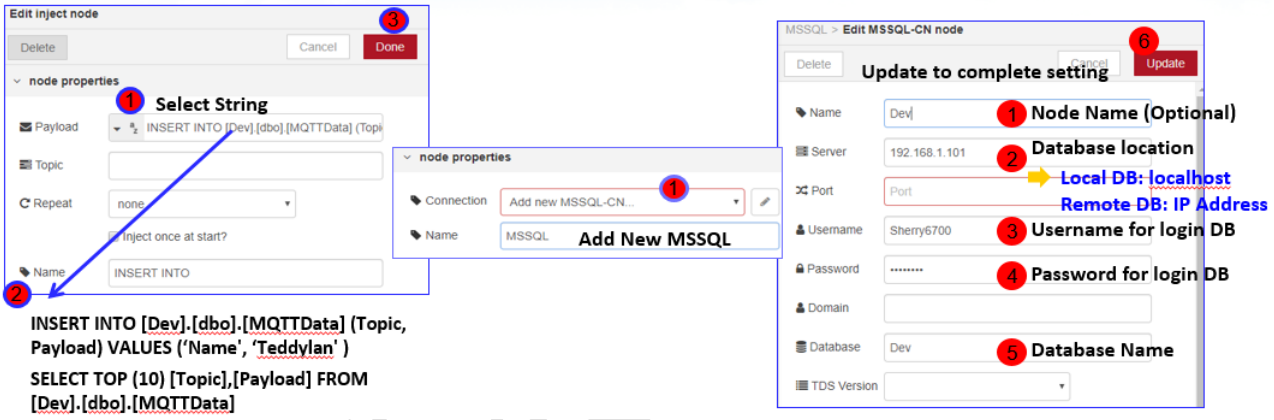
2.2 Update data to SQL database

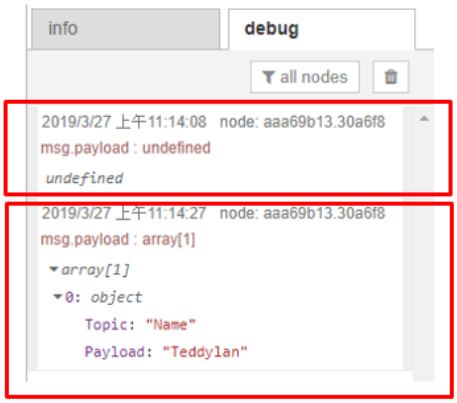
Node-red flow example:



Used nodes & Steps:

- Drag above node and connect them to a flow
  - Inject Node: Enter SQL command
  - MSSQL Node: Refer to below settings
  - Debug Node: For printing the message.





1. After inject message to MSSQL, it should show this message.

2. After inject SELECT node, it should show the message you just inject.

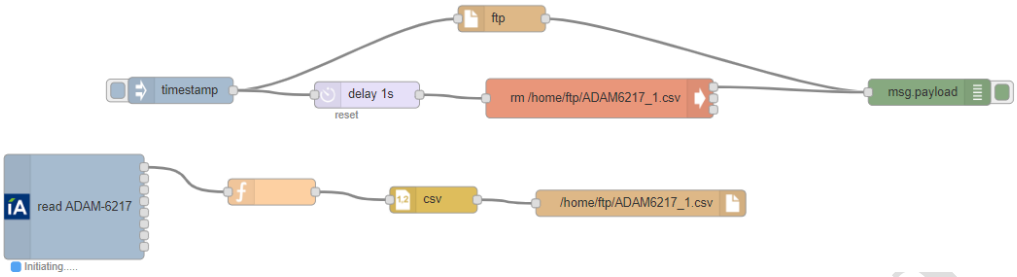
Sample flow

[{"id":"e7774694.0c3a18","type":"inject","z":"760d96cd.7330e8","name":"To_AEtest_select","topic":"","payload":"SELECT TOP (3) [Topic],[Payload] FROM [AEtest].[dbo].[MQTTdata]","payloadType":"str","repeat":"","crontab":"","once":false,"x":160,"y":60,"wires":[["b8f1916f.26163"]]},{"id":"a4a52abe.bb4438","type":"debug","z":"760d96cd.7330e8","name":"","active":true,"console":"false","complete":"false","x":570,"y":40,"wires":[]},{"id":"ab3b3fd2.27a93","type":"i

nject","z":"760d96cd.7330e8","name":"To_AEtest_insert","topic":"","payload":"INSERT INTO [AEtest].[dbo].[MQTT Data] (Topic, Payload) VALUES ('NAME', 'Sherry0917')","payloadType":"str","repeat":"","crontab":"","once":false,"x":160,"y":100,"wires":[["b8f1916f.26163"]]},{"id":"b8f1916f.26163","type":"MSSQL","z":"760d96cd.7330e8","mssqlCN":"1cfde562.4490cb","name":"AEtest_sql","query":"","outField":"payload","x":390,"y":40,"wires":[["a4a52abe.bb4438"]]},{"id":"1cfde562.4490cb","type":"MSSQL-CN","z":"","name":"MSSQL_nb_AEtest","server":"172.16.16.67","port":"","encyption":false,"database":"AEtest","useUTC":false,"connectTimeout":"","requestTimeout":"","cancelTimeout":"","pool":""}]

2.3 Update data to Remote FTP server

Node-red flow example:



This flow will help to get I/O data from ADAM-6717, and then store into a csv file with column name “channel0” in ADAM-6717 local side. Then we use a timestamp node to trigger the ftp node send the local file from ADAM-6717 to remote FTP server. Since time I/O data will keep updating to the local csv file inside ADAM-6717, so we use the exec node to delete the original file for a certain period. This flow is just a concept, User can change to the node-red flow logic based on their application need. Please note that user have to install the 3rd party FTP node into ADAM-6717 https://flows.nodered.org/node/node-red-contrib-ftp

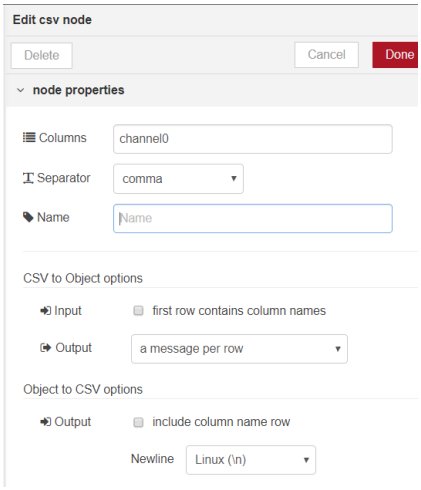
Used nodes & Steps

Step 1: Drag below node and connect them to a flow

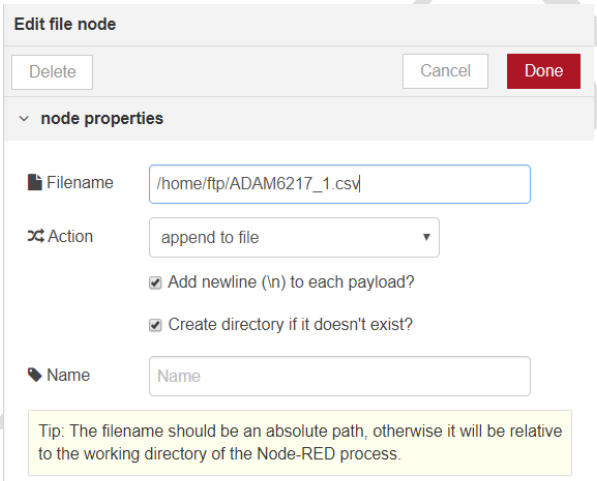
- ADAM Node: Get remote I/O data from ADAM-6217, please make sure you enter the correct IP of ADAM module
- Function Node: Convert AI value to string
- CSV Node: Converts between a CSV formatted string and its JavaScript object representation, user can also define the column name in the CSV file.
- FTP Node: For send file to FTP server, please enter the IP of FTP server.
- File Node: For create a file on ADAM-6717 Local directory
- Exec Node: Runs a system command and returns its output, this is used for delete the local file inside ADAM-6717.
- Delay Node: For delay time setting.

Step 2: Check each node setting

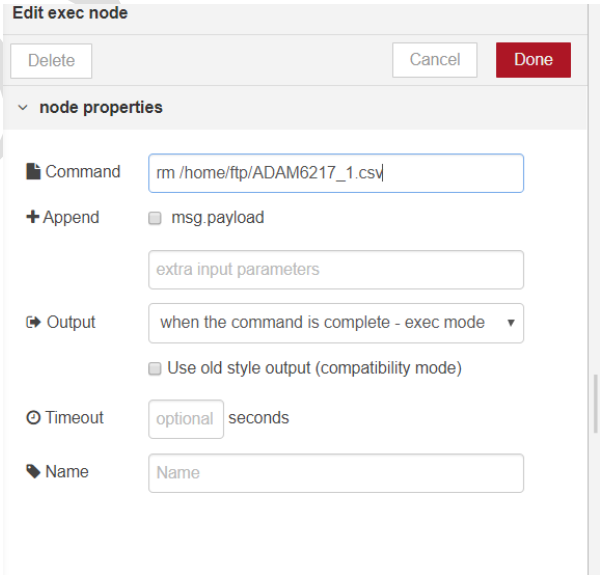
- CSV Node: Enter the “Column name” of AI raw data, choose “comma” for the separator.



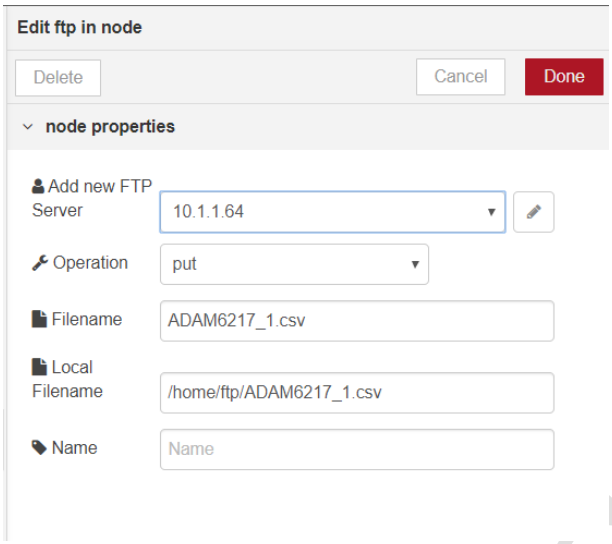
– File Node: Enter the filename, file type and directory you wish to create in ADAM-6717



– Exec Node: Enter the remove file linux command after certain delay time



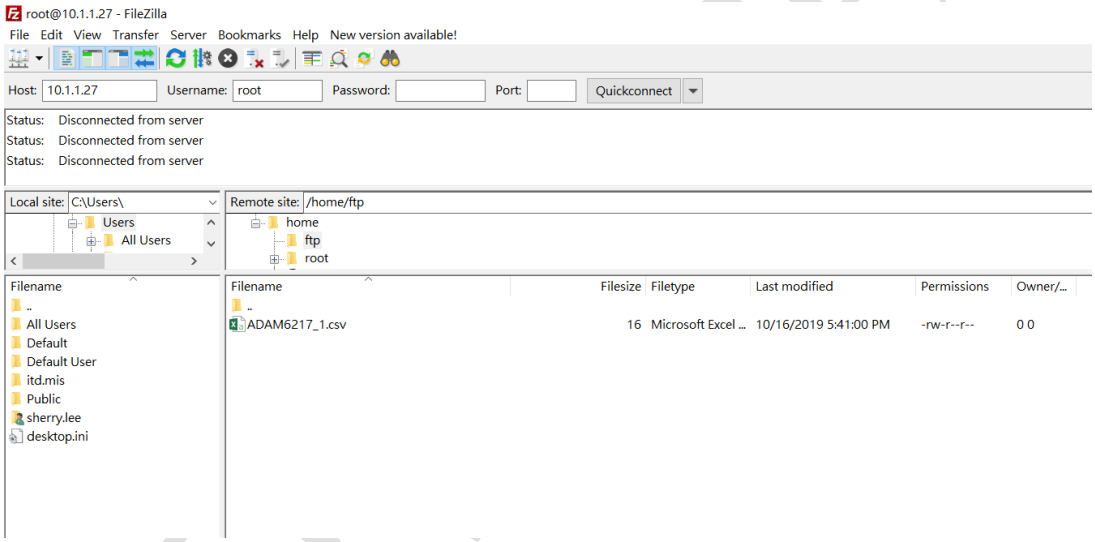
– FTP Node: Enter the IP of FTP server



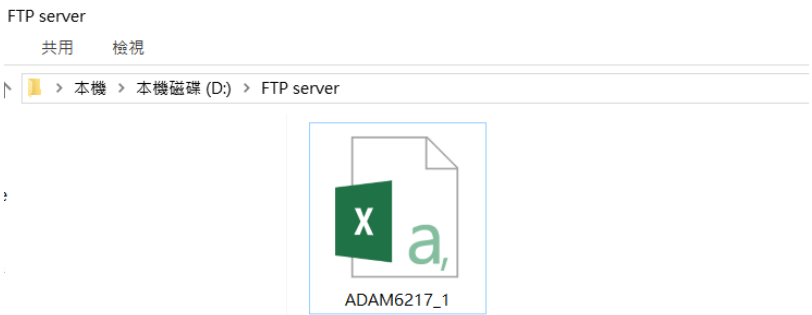
Step 3: See the result

Below is the file created in ADAM-6717 with below directory,

/home/ftp/ADAM6217_1.csv



Below is the file send to the remote FTP server with filename ADAM6217_1.csv



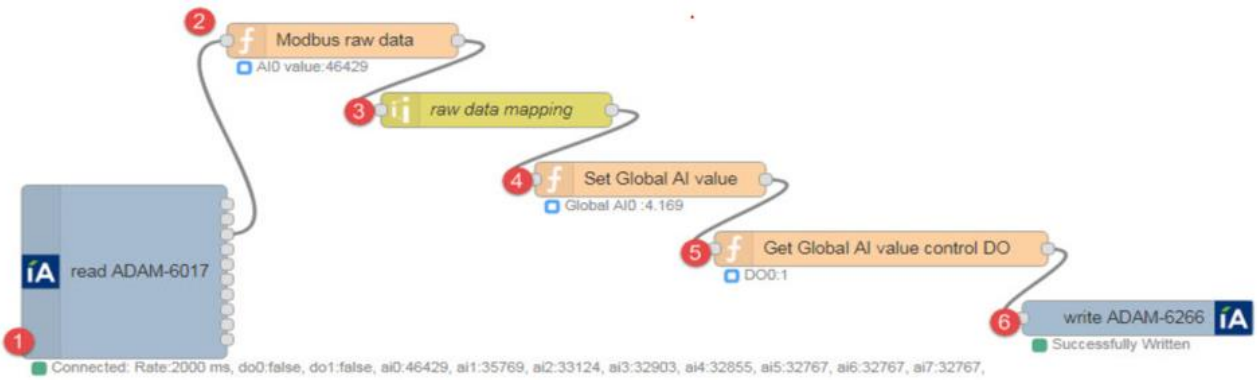
Sample flow

[{"id":"4f31f50b.13866c","type":"inject","z":"bcf408e6.aad5f8","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"x":272,"y":158,"wires":[["36c62094.e7051","307ff56a.f58c7a"]]},{"id":"33f90e6e.88be52","type":"file","z":"bcf408e6.aad5f8","name":"","filename":"/home/ftp/ADAM6217_1.csv","appendNewline":true,"createDir":true,"overwriteFile":"false","x":797.9999313354492,"y":287.0000057220459,"wires":[]},{"id":"36c62094.e7051","type":"ftp in","z":"bcf408e6.aad5f8","ftp":"4a16a252.1e894c","operation":"put","filename":"ADAM6217_1.csv","localFilename":"/home/ftp/ADAM6217_1.csv","name":"","x":638.9999923706055,"y":76.0000319480896,"wires":[["dacebd67.7a7d"]]},{"id":"5fa20cb5.da9fa4","type":"csv","z":"bcf408e6.aad5f8","name":"","sep":",","hdrin":false,"hdrout":false,"multi":"one","ret":"\\n","temp":"channel0","x":560.9999885559082,"y":283.0000057220459,"wires":[["33f90e6e.88be52"]]},{"id":"dacebd67.7a7d","type":"debug","z":"bcf408e6.aad5f8","name":"","active":true,"console":"false","complete":"false","x":1128,"y":160.00003242492676,"wires":[]},{"id":"307ff56a.f58c7a","type":"delay","z":"bcf408e6.aad5f8","name":"","pauseType":"delay","timeout":"1","timeoutUnits":"seconds","rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x":485.0000534057617,"y":163.00002574920654,"wires":[["c3223653.62f3d8"]]},{"id":"c3223653.62f3d8","type":"exec","z":"bcf408e6.aad5f8","command":"rm /home/ftp/ADAM6217_1.csv","addpay":false,"append":"","useSpawn":"false","timer":"","oldrc":false,"name":"","x":750.9999809265137,"y":167.00006103515625,"wires":[["dacebd67.7a7d"],[],[]]},{"id":"d5d62951.2e9c08","type":"function","z":"bcf408e6.aad5f8","name":"to String","func":"var num=msg.payload;\nmsg.payload=num.toString();\nreturn msg;","outputs":1,"noerr":0,"x":386.0000343322754,"y":274.00000190734863,"wires":[["5fa20cb5.da9fa4"]]},{"id":"9cccc6c8.f3ffd8","type":"ADAM-read","z":"bcf408e6.aad5f8","name":"","rate":2000,"host":"10.0.0.217","serialPortCfg":"","unit_id":1,"readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-6217","advDevTypeRTU":"ADAM-4015","advDevType":"ADAM-6217","x":150.08332061767578,"y":291.11458587646484,"wires":[["d5d62951.2e9c08"],[],[],[],[],[],[],[]]},{"id":"4a16a252.1e894c","type":"ftp","z":"","host":"10.1.1.64","port":"","secureOptions":"","user":"Sherry","connTimeout":"","pasvTimeout":"","keepalive":""}]

3 I/O extension and application

3.1 AI Alarm trigger DO

Node-red flow example:



Used nodes & Steps:

- **Step 1:** Make sure "***read ADAM node***" is connected to ADAM-6017 (IP =10.1.1.16)
- **Step 2:** Use function node "***function***" to write script and print the **node status** to show AI raw data
- **Step 3**: Use function node "***range***" to convert Modbus raw data into engineering data
- **Step 4**: Use function node "***function***" to Set Global AI value also Fixed decimal place for better present
- **Step 5:** Use function node "***function***" to Get Global AI value as the condition for controlling DO
- **Step 6:** Use "***ADAM write node***" to control DO channel of ADAM-6266 (IP=10.1.1.19)

Sample flow

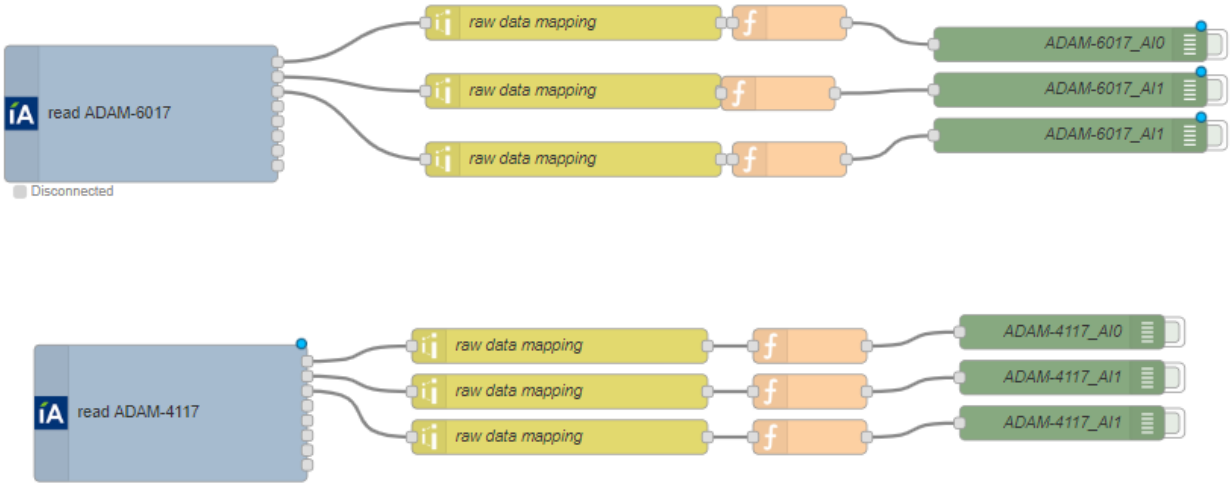
[{"id":"14b33a72.3548e6","type":"ADAM-read","z":"21056b5e.9abb64","name":"","rate":2000,"host":"10.1.1.16","serialPortCfg":"","unit_id":1,"readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-6017","advDevTypeRTU":"ADAM-4015","advDevType":"ADAM-6017","x":120,"y":320,"wires":[[],[],["b2f073fb.77df1"],[],[],[],[],[],[],[]]},{"id":"ec3151da.4a6d6","type":"function","z":"21056b5e.9abb64","name":"Set Global AI value","func":"var AI0 = msg.payload;\nglobal.set(\"AI0\",AI0); // this is now available to other nodes\nmsg.payload = AI0.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"Global AI0 :\"+AI0.toFixed(3)});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":490,"y":240,"wires":[["863db8fd.f92098","1a7e3f4e.487231","e489efe1.acd6d"]]},{"id":"863db8fd.f92098","type":"function","z":"21056b5e.9abb64","name":"Get Global AI value control DO","func":" var DO0 = 0;\n var ai0 = global.get(\"AI0\"); \n\n // ai0\n if (ai0 > 2){\n   DO0 = 1;\n }\n\n node.status({fill:\"blue\",shape:\"ring\",text:\" DO0:\"+DO0});\n\n//node.status({fill:\"blue\",shape:\"ring\",text:\"Global AI0 :\"+AI0.toFixed(3)});\n var result =

[DO0];\nmsg.payload =result;\n \n//return msg;\n\nnode.send(msg);\n\t\t\n\n\n/*\n\nvar aa=0;\n\nvar timerID; \t\nclearInterval(timerID);\ntimerID = null; \nif (!timerID) { \n timerID = setInterval(function () { \n\t\taa++;\n\t\t//node.status({fill:\"blue\",shape:\"ring\",text:aa });\n\t\t\n var DO0 = 0;\n var ai0 = global.get(\"AI0\"); \n var ai1 = global.get(\"AI1\"); \n var ai2 = global.get(\"AI2\"); \n // ai0\n if (ai0 > 60){\n DO0 = 1;\n }\n else {\n DO0 = 0;\n }\n \n // ai1\n if (ai1 > 400){\n DO1 = 1;\n \n }\n else {\n DO1 = 0;\n }\n \n // ai2\n \n if (ai2 > 300){\n DO2 = 1;\n \n }\n else {\n DO2 = 0;\n }\n \n //node.status({fill:\"blue\",shape:\"ring\",text:\"[times:\"+aa+\"]\"+\" DO0:\"+DO0+\" DO1:\"+DO1+\" DO2:\"+DO2 });\n node.status({fill:\"blue\",shape:\"ring\",text:\"AI0 value:\"+ai0});\n\n //var result = msg.payload;\n var result = [DO0,DO1,DO2];\n msg.payload =result;\n \n //return msg;\n node.send(msg);\n\t\t\n\t\t\n\t\t\n }, 500);\n} \t\n*/\n","outputs":1,"noerr":0,"x":670,"y":300,"wires":[["e3f6aec.211fe5"]]},{"id":"c9588b46.df55f8","type":"range","z":"21056b5e.9abb64","minin":"0","maxin":"65535","minout":"-10","maxout":"10","action":"scale","round":false,"name":"raw data mapping","x":370,"y":180,"wires":[["ec3151da.4a6d6"]]},{"id":"b2f073fb.77df1","type":"function","z":"21056b5e.9abb64","name":"Modbus raw data","func":"//var ai0 = global.get(\"AI0\"); \nvar value = Number(msg.payload);\n//msg.payload = value.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"AI0 value:\"+value});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":250,"y":120,"wires":[["c9588b46.df55f8"]]},{"id":"e3f6aec.211fe5","type":"ADAM-write","z":"21056b5e.9abb64","name":"","host":"10.1.1.19","serialPortCfg":"","unit_id":1,"write_ch":0,"write_ch_type":"write_do_n","reconnecttimeout":"5","Series":"mbtcp","advDevTypeTCP":"ADAM-6266","advDevTypeRTU":"ADAM-4022T","advDevType":"ADAM-6266","x":870,"y":360,"wires":[]},{"id":"1a7e3f4e.487231","type":"ui_chart","z":"21056b5e.9abb64","name":"6017_AI0_Voltage","group":"398ef60e.ae593a","order":4,"width":"6","height":"4","label":"AI Voltage","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","ymin":"0","ymax":"10","removeOlder":"5","removeOlderPoints":"","removeOlderUnit":"60","cutout":"","colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"x":860,"y":120,"wires":[]},{"id":"e489efe1.acd6d","type":"ui_gauge","z":"21056b5e.9abb64","name":"6017_AI0_Voltage","group":"398ef60e.ae593a","order":3,"width":"6","height":"4","gtype":"gage","title":"Voltage","label":"Volt","format":"{{value}}","min":"0","max":"10","colors":["#00b500","#e6e600","#ca3838"],"seg1":"4","seg2":"6","x":850,"y":160,"wires":[]},{"id":

"398ef60e.ae593a","type":"ui_group","z":"","name":"Hands-on 2-2","tab":"15992465.c9d5ac","disp":true,"width":"6"},{"id":"15992465.c9d5ac","type":"ui_tab","z":"","name":"Leapcamp","icon":"dashboard"}]

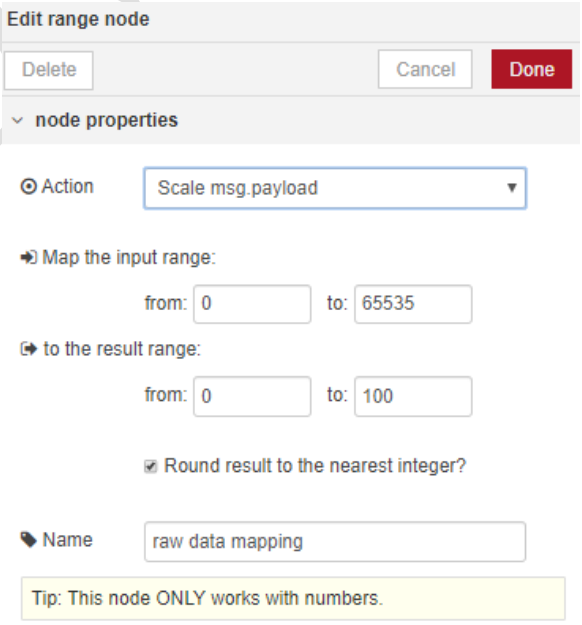
3.2 Extend IO with ADAM-4000/6000 series

Node-red flow example:



Used nodes & Steps:

- Step 1: Drag above node and connect them to a flow
  - ADAM read Node: Select the module you are going to read Modbus data, Enter Module's IP, polling rate, and reconnect interval in this node
  - Range Node: For Modbus raw data mapping to real data output range. In other words, this node can be also used to scale the raw data.



- Debug Node: For printing the message.

- Step 2: See the result on debug column or node status
  - Node status: If connected to ADAM successfully, it will show connected also print the I/O status of each channel
  - Debug column: It will print the message of your debug node

Sample flow

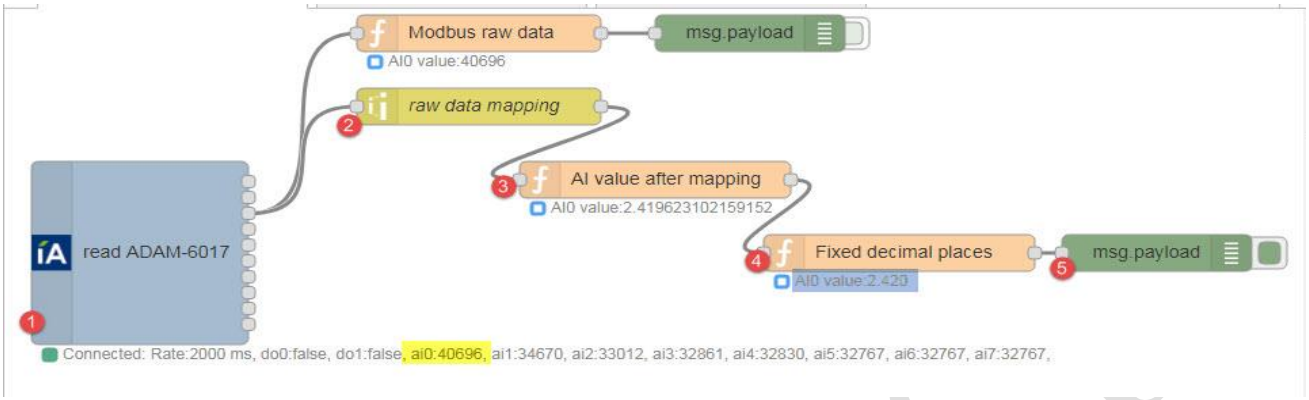
[{"id":"fdeeffd2.2ccf8","type":"ADAM-read","z":"569e21d4.26ac7","name":"","rate":"1000","host":"127.0.0.1","serialPortCfg":"de95c959.e1ba78","unit_id":"1","readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbrtu","advDevTypeTCP":"ADAM-6015","advDevTypeRTU":"ADAM-4117","advDevType":"ADAM-4117","x":2148.0000610351562,"y":1699.0000095367432,"wires":[["b53ad.1a982c54"],["c80b7321.bea7f"],["cbb15274.0b60f"],[],[],[],[],[]]},{"id":"b53ad.1a982c54","type":"range","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"14","action":"scale","round":false,"name":"raw data mapping","x":2490,"y":1640,"wires":[["58c3554a.f44c2c"]]},{"id":"58c3554a.f44c2c","type":"function","z":"569e21d4.26ac7","name":"","func":"var value = Number(msg.payload);\nmsg.payload = value.toFixed(3);\nreturn msg;","outputs":1,"noerr":0,"x":2710,"y":1640,"wires":[["c7088005.70b86"]]},{"id":"c80b7321.bea7f","type":"range","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"5000","action":"scale","round":false,"name":"raw data mapping","x":2490,"y":1680,"wires":[["3a51d155.f92cfe"]]},{"id":"3a51d155.f92cfe","type":"function","z":"569e21d4.26ac7","name":"","func":"var value = Number(msg.payload);\nmsg.payload = value.toFixed(3);\nreturn msg;","outputs":1,"noerr":0,"x":2710,"y":1680,"wires":[["39625c38.757544"]]},{"id":"cbb15274.0b60f","type":"range","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"50","action":"scale","round":false,"name":"raw data mapping","x":2490,"y":1720,"wires":[["8c751efb.9f551"]]},{"id":"8c751efb.9f551","type":"function","z":"569e21d4.26ac7","name":"","func":"var value = Number(msg.payload);\nmsg.payload = value.toFixed(3);\nreturn msg;","outputs":1,"noerr":0,"x":2710,"y":1720,"wires":[["760d7511.599ecc"]]},{"id":"2bb832d9.cd02ee","type":"range","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"600","action":"scale","round":true,"name":"raw data mapping","x":2502,"y":1416,"wires":[["9ce95c34.91109"]]},{"id":"dae8b2fa.70b1b","type":"range","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"500","action":"scale

","round":true,"name":"raw data
mapping","x":2502,"y":1476,"wires":[["c0441e01.b4a24"]]},{"id":"5c17515b.333c7","type":"range
","z":"569e21d4.26ac7","minin":"0","maxin":"65535","minout":"0","maxout":"100","action":"scal
e","round":true,"name":"raw data
mapping","x":2502,"y":1356,"wires":[["760d72ca.88b3ec"]]},{"id":"9ce95c34.91109","type":"funct
ion","z":"569e21d4.26ac7","name":"","func":"var value = Number(msg.payload);\n//msg.payload
= value.toFixed(3);\nreturn
msg;","outputs":1,"noerr":0,"x":2682,"y":1418,"wires":[["d9d1fae1.8a9988"]]},{"id":"c0441e01.b4
a24","type":"function","z":"569e21d4.26ac7","name":"","func":"var value =
Number(msg.payload);\n//msg.payload = value.toFixed(3);\nreturn
msg;","outputs":1,"noerr":0,"x":2692,"y":1476,"wires":[["6919ac9f.da3424"]]},{"id":"f7034a8.ab41
ab8","type":"ADAM-
read","z":"569e21d4.26ac7","name":"","rate":"1000","host":"10.0.0.17","serialPortCfg":"","unit_i
d":1,"readDI":true,"readDO":false,"readAI":true,"readAO":true,"reconnecttimeout":"5","mboutpu
tStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-
6017","advDevTypeRTU":"ADAM-4015","advDevType":"ADAM-
6017","x":2122,"y":1436,"wires":[["5c17515b.333c7"],["2bb832d9.cd02ee"],["dae8b2fa.70b1b"],[]
,[],[],[],[]]},{"id":"760d72ca.88b3ec","type":"function","z":"569e21d4.26ac7","name":"","func":"va
r value = Number(msg.payload);\n//msg.payload = value.toFixed(3);\n\nreturn
msg;","outputs":1,"noerr":0,"x":2692,"y":1356,"wires":[["ff69e3f3.ae7c4"]]},{"id":"d9d1fae1.8a99
88","type":"debug","z":"569e21d4.26ac7","name":"ADAM-
6017_AI1","active":false,"console":"false","complete":"payload","x":2939.0000915527344,"y":141
5,"wires":[]},{"id":"ff69e3f3.ae7c4","type":"debug","z":"569e21d4.26ac7","name":"ADAM-
6017_AI0","active":false,"console":"false","complete":"payload","x":2939.0000915527344,"y":137
5,"wires":[]},{"id":"6919ac9f.da3424","type":"debug","z":"569e21d4.26ac7","name":"ADAM-
6017_AI1","active":false,"console":"false","complete":"payload","x":2939.0000915527344,"y":145
5,"wires":[]},{"id":"39625c38.757544","type":"debug","z":"569e21d4.26ac7","name":"ADAM-
4117_AI1","active":false,"console":"false","complete":"payload","x":2931.666748046875,"y":1667
.6666259765625,"wires":[]},{"id":"c7088005.70b86","type":"debug","z":"569e21d4.26ac7","name
":"ADAM-
4117_AI0","active":false,"console":"false","complete":"payload","x":2931.666748046875,"y":1627
.6666259765625,"wires":[]},{"id":"760d7511.599ecc","type":"debug","z":"569e21d4.26ac7","nam
e":"ADAM-
4117_AI1","active":false,"console":"false","complete":"payload","x":2931.666748046875,"y":1707
.6666259765625,"wires":[]},{"id":"de95c959.e1ba78","type":"ADAM-serial-
port","z":"","name":"","serialPort":"COM1","serialBaudrate":"9600","serialDatabits":"8","serialSto
pbits":"1","serialParity":"none","unit_id":"2","clientTimeout":"1000","reconnecttimeout":"2000"}]

4 Data processing

4.1 Analyze the data , scaling, calculation

Node-red flow example:



Used nodes & Steps:

- **Step1:** Make sure "***read ADAM mode***" is connected to ADAM-6017 (IP =10.1.1.16)
- **Step2:** Use function node "***range***" to convert Modbus raw data into engineering data
- **Step3**: Use function node "***function***" to write script and print the **node status** to show the mapping result
- **Step4**: Use function node "***function***" to Fixed decimal place for better present
- **Step5:** Use output node "***Debug***" for printing the message.

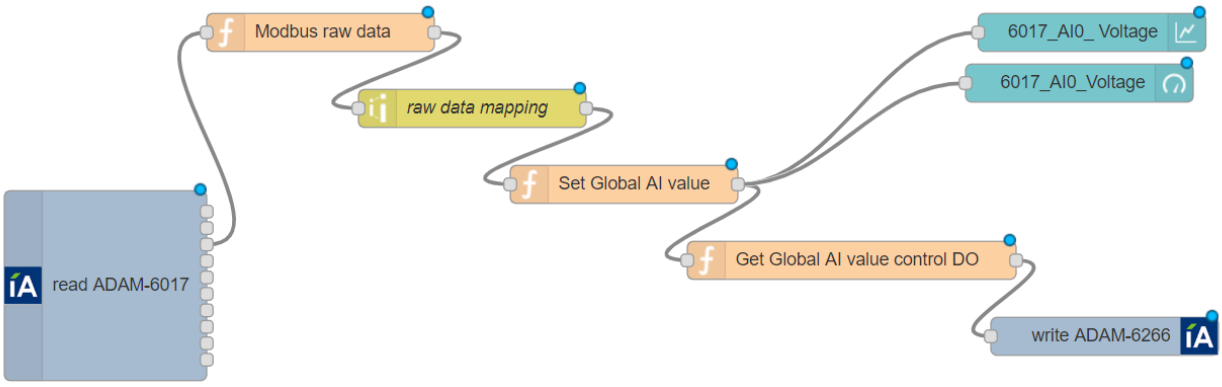
Sample flow

[{"id":"ec0a0e95.10542","type":"ADAM-read","z":"c36940f4.544f","name":"","rate":2000,"host":"10.1.1.16","serialPortCfg":"","unit_id":1,"readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-6017","advDevTypeRTU":"ADAM-4015","advDevType":"ADAM-6017","x":120,"y":440,"wires":[[],[],["c6ceb75d.4be5e8","f58ba9be.a87e98"],[],[],[],[],[],[],[]]},{"id":"c6ceb75d.4be5e8","type":"function","z":"c36940f4.544f","name":"Modbus raw data","func":"//var ai0 = global.get(\"AI0\"); \nvar value = Number(msg.payload);\n//msg.payload = value.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"AI0 value:\"+value});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":430,"y":340,"wires":[["ce817b71.1b83b8"]]},{"id":"f58ba9be.a87e98","type":"range","z":"c36940f4.544f","minin":"0","maxin":"65535","minout":"-10","maxout":"10","action":"scale","round":false,"name":"raw data mapping","x":430,"y":400,"wires":[["23f1f222.45f8be"]]},{"id":"23f1f222.45f8be","type":"function","z":"c36940f4.544f","name":"AI value after mapping","func":"var value = Number(msg.payload);\nmsg.payload = value.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"AI0

value:\"+value});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":560,"y":460,"wires":[["a92096aa.627cd8"]]},{"id":"dfd4d413.7df498","type":"debug","z":"c36940f4.544f","name":"","active":false,"console":"false","complete":"false","x":990,"y":520,"wires":[]},{"id":"a92096aa.627cd8","type":"function","z":"c36940f4.544f","name":"Fixed decimal places","func":"var value = Number(msg.payload);\nmsg.payload = value.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"AI0 value:\"+value.toFixed(3)});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":780,"y":520,"wires":[["dfd4d413.7df498"]]},{"id":"ce817b71.1b83b8","type":"debug","z":"c36940f4.544f","name":"","active":false,"console":"false","complete":"false","x":630,"y":340,"wires":[]}]

4.2 Data visualization by dashboard

Node-red flow example:

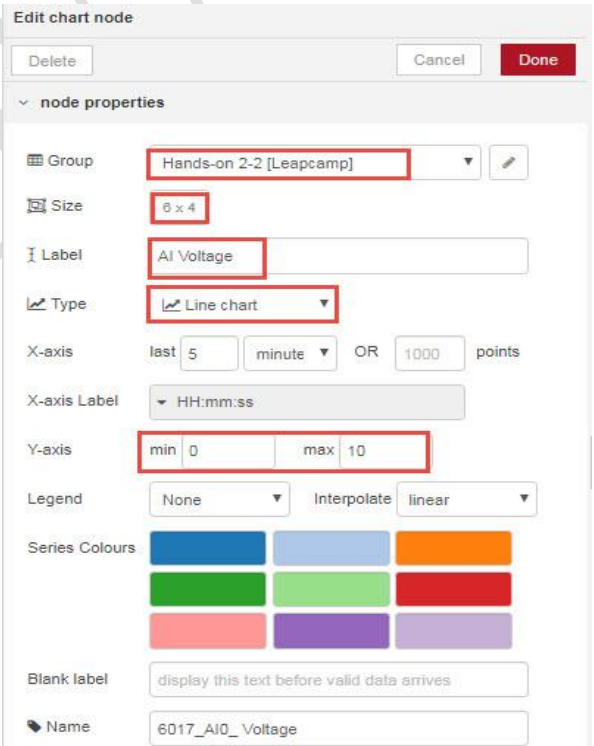


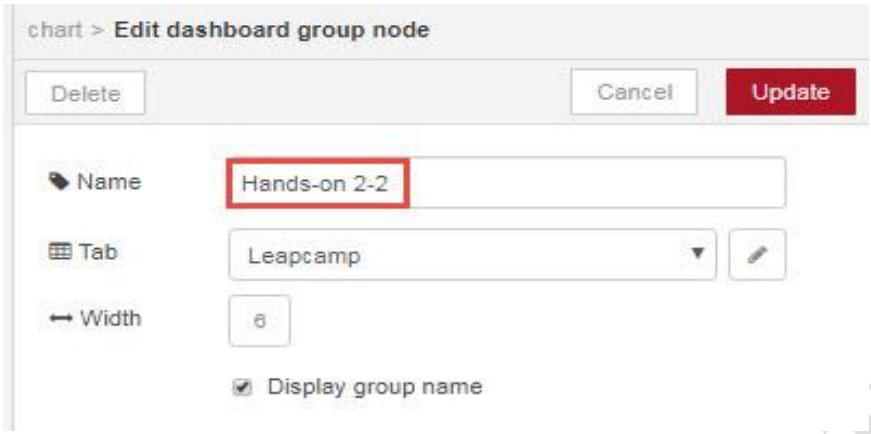
Used nodes & Steps:

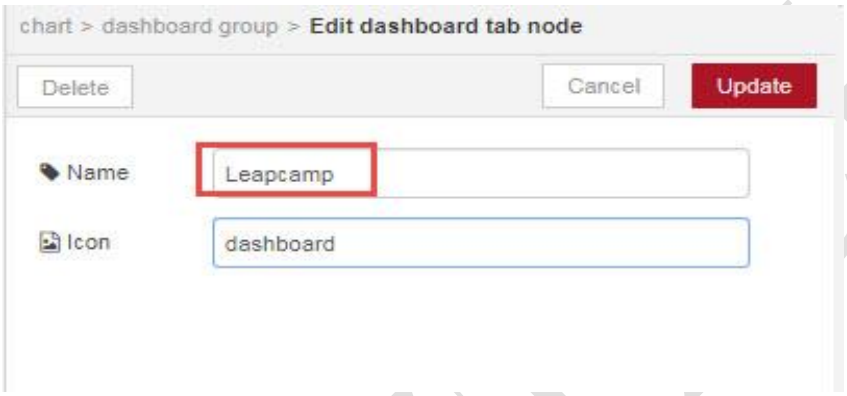
Step 1: Drag the dashboard node to display specific value of a node-red node, each dashboard node is represent a dashboard widget

Step 2: Edit Group, tab name and bind this name to your widget

Enter http://localhost:1880/ui on browser to launch the screen with 2 in total of the dashboard node







Sample flow

[{"id":"b0fd0e3b.0eefb","type":"ADAM-read","z":"c36940f4.544f","name":"","rate":2000,"host":"10.1.1.16","serialPortCfg":"","unit_id":1,"readDI":true,"readDO":true,"readAI":true,"readAO":true,"reconnecttimeout":"","mboutputStyle":"Multiple Outputs","Series":"mbtcp","advDevTypeTCP":"ADAM-6017","advDevTypeRTU":"ADAM-4015","advDevType":"ADAM-6017","x":180,"y":1640,"wires":[[],[],["c4f9c1ae.53442"],[],[],[],[],[],[],[]]},{"id":"2535880b.8d9e78","type":"function","z":"c36940f4.544f","name":"Set Global AI value","func":"var AI0 = msg.payload;\nglobal.set(\"AI0\",AI0); // this is now available to other nodes\nmsg.payload = AI0.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"Global AI0 :\"+AI0.toFixed(3)});\n\n\nreturn msg;","outputs":1,"noerr":0,"x":590,"y":1560,"wires":[["ee7696f7.8e3db8","94625da9.36fc5","3ca38259.42e63e"]]},{"id":"ee7696f7.8e3db8","type":"function","z":"c36940f4.544f","name":"Get Global AI value control DO","func":"   var DO0 = 0;\n   var ai0 = global.get(\"AI0\"); \n\n   // ai0\n   if (ai0 > 2){\n   DO0 = 1;\n   }\n\n node.status({fill:\"blue\",shape:\"ring\",text:\" DO0:\"+DO0});\n\n//node.status({fill:\"blue\",shape:\"ring\",text:\"Global AI0 :\"+AI0.toFixed(3)});\n   var result = [DO0];\nmsg.payload =result;\n   \n//return msg;\n\nnode.send(msg);\n\t\t\n\n/*\n\nvar aa=0;\n\nvar timerID;\n\tclearInterval(timerID);\ntimerID = null; \nif (!timerID) { \n

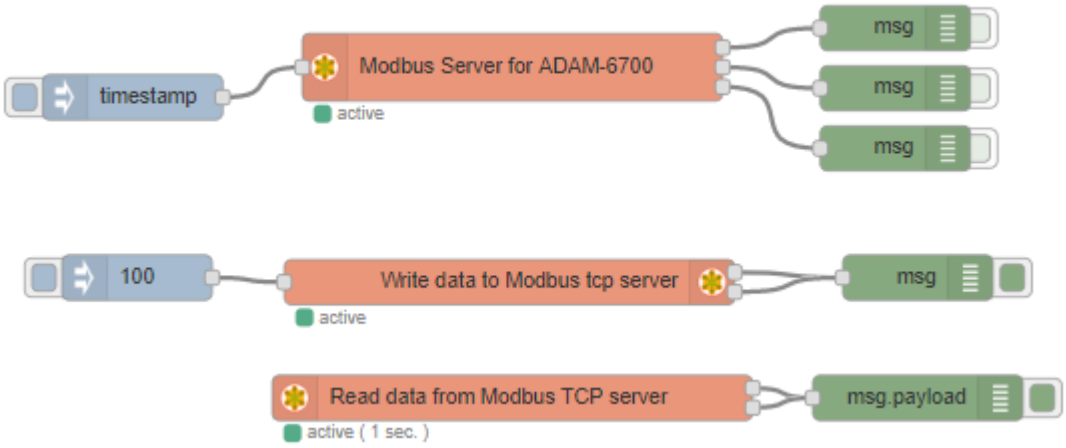
timerID = setInterval(function ()
{ \n\t\taa++;\n\t\t//node.status({fill:\"blue\",shape:\"ring\",text:aa });\n\t\t\n
var DO0 = 0;\n var ai0 = global.get(\"AI0\"); \n var ai1 = global.get(\"AI1\"); \n
var ai2 = global.get(\"AI2\"); \n // ai0\n if (ai0 > 60){\n DO0 =
1;\n }\n else {\n DO0 = 0;\n }\n \n //
ai1\n if (ai1 > 400){\n DO1 = 1;\n \n }\n
else {\n DO1 = 0;\n }\n \n // ai2\n \n
if (ai2 > 300){\n DO2 = 1;\n \n }\n else {\n
DO2 = 0;\n }\n \n
//node.status({fill:\"blue\",shape:\"ring\",text:\"[times:\"+aa+\"]\"+\" DO0:\"+DO0+\"
DO1:\"+DO1+\" DO2:\"+DO2 });\n node.status({fill:\"blue\",shape:\"ring\",text:\"AI0
value:\"+ai0});\n\n //var result = msg.payload;\n var result = [DO0,DO1,DO2];\n
msg.payload =result;\n \n //return msg;\n
node.send(msg);\n\t\t\n\t\t\n\t\t\n\t\t\n }, 500);\n}
\t\n*/\n","outputs":1,"noerr":0,"x":770,"y":1620,"wires":[["84fe137b.841e5"]]},{"id":"fd83406d.8
2f2d","type":"range","z":"c36940f4.544f","minin":"0","maxin":"65535","minout":"-
10","maxout":"10","action":"scale","round":false,"name":"raw data
mapping","x":470,"y":1500,"wires":[["2535880b.8d9e78"]]},{"id":"c4f9c1ae.53442","type":"functi
on","z":"c36940f4.544f","name":"Modbus raw data","func":"//var ai0 = global.get(\"AI0\"); \nvar
value = Number(msg.payload);\n//msg.payload =
value.toFixed(3);\nnode.status({fill:\"blue\",shape:\"ring\",text:\"AI0
value:\"+value});\n\n\nreturn
msg;","outputs":1,"noerr":0,"x":350,"y":1440,"wires":[["fd83406d.82f2d"]]},{"id":"84fe137b.841e
5","type":"ADAM-
write","z":"c36940f4.544f","name":"","host":"10.1.1.19","serialPortCfg":"","unit_id":1,"write_ch":
0,"write_ch_type":"write_do_n","reconnecttimeout":"5","Series":"mbtcp","advDevTypeTCP":"AD
AM-6266","advDevTypeRTU":"ADAM-4022T","advDevType":"ADAM-
6266","x":970,"y":1680,"wires":[]},{"id":"94625da9.36fc5","type":"ui_chart","z":"c36940f4.544f","
name":"6017_AI0_
Voltage","group":"5216a502.a1fb2c","order":4,"width":"6","height":"4","label":"AI
Voltage","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata"
:"","ymin":"0","ymax":"10","removeOlder":"5","removeOlderPoints":"","removeOlderUnit":"60","
cutout":"","colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9
467bd","#c5b0d5"],"x":960,"y":1440,"wires":[[],[]]},{"id":"3ca38259.42e63e","type":"ui_gauge","z
":"c36940f4.544f","name":"6017_AI0_Voltage","group":"5216a502.a1fb2c","order":3,"width":"6",
"height":"4","gtype":"gage","title":"Voltage","label":"Volt","format":"{{value}}","min":"0","max":"1
0","colors":["#00b500","#e6e600","#ca3838"],"seg1":"4","seg2":"6","x":950,"y":1480,"wires":[]},{"
id":"5216a502.a1fb2c","type":"ui_group","z":"","name":"Hands-on 2-
2","tab":"4a0e9ea4.78174","disp":true,"width":"6"},{"id":"4a0e9ea4.78174","type":"ui_tab","z":""
,"name":"Leapcamp","icon":"dashboard"}]

5 Others

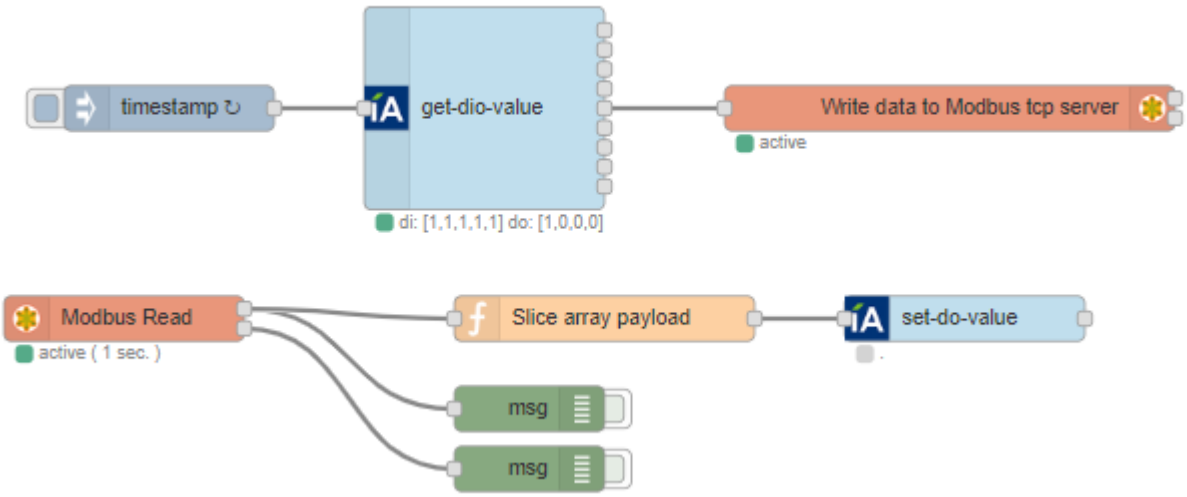
5.1 Modbus TCP server

Node-red flow example:

Create Modbus TCP server



Get Local DIO data and write to Modbus TCP server



Used nodes & Steps:

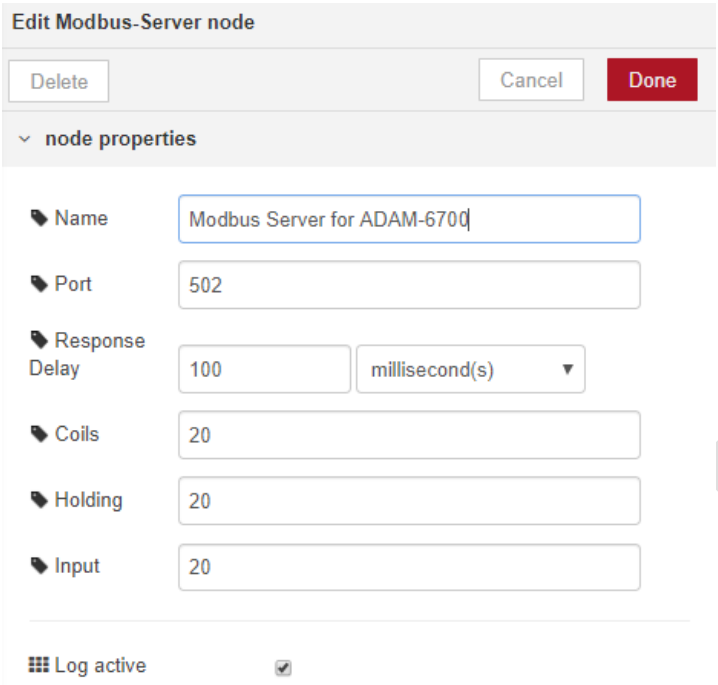
Step1: Use Modbus-Server node to create a space for Modbus TCP server.

For example, user can create the Modbus address for certain amount. In this example, we create below address.

Output 1: holding Buffer, type, msg

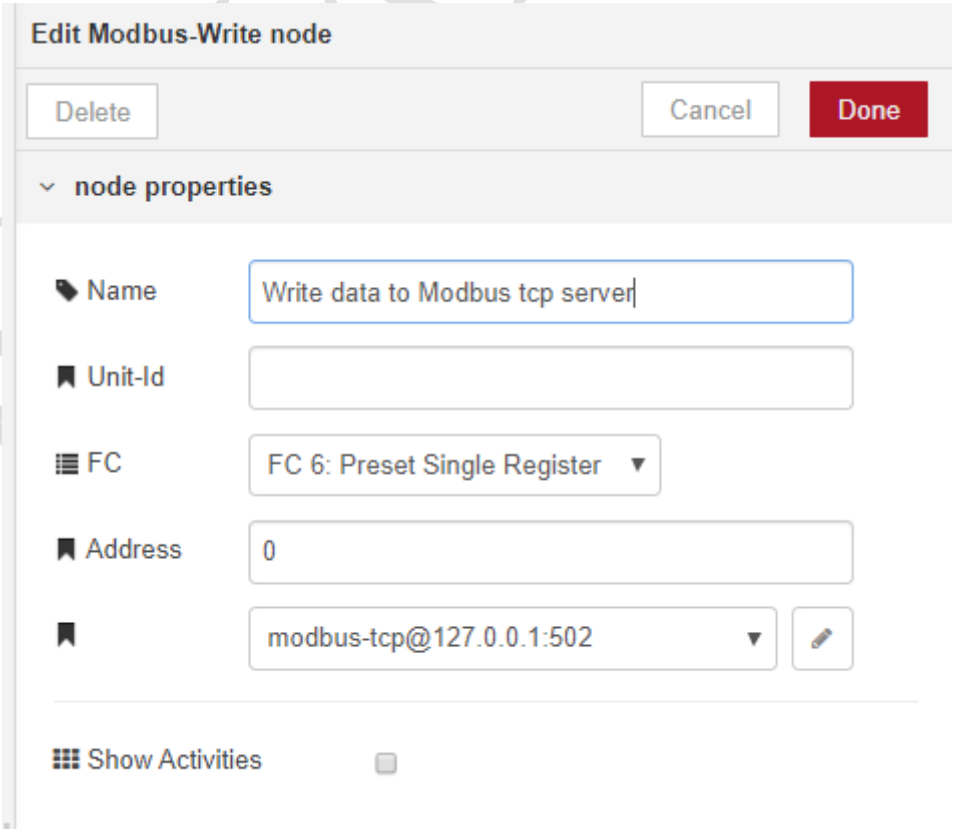
Output 2: coils Buffer, type, msg

Output 3: input Buffer, type, msg

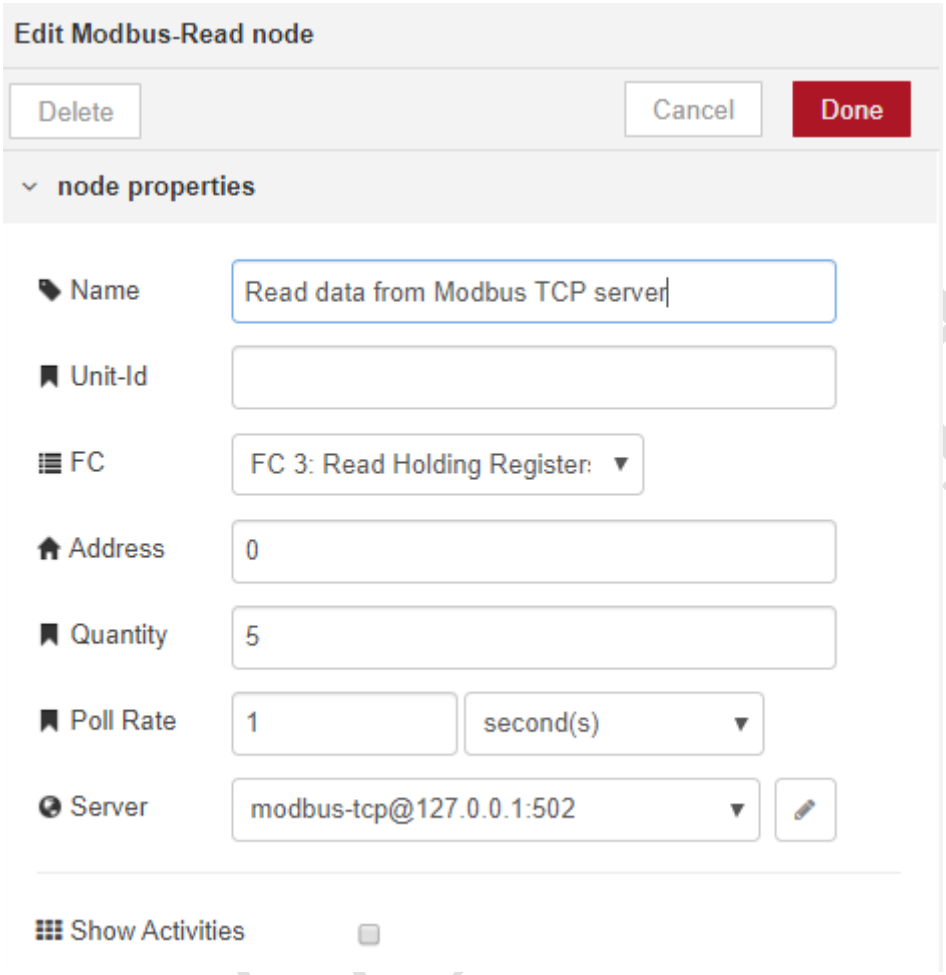


Step2: After you create Modbus address range, you can write data to the Modbus address you just create

You can use the Modbus write nodes (FC) to write data to the server buffers.
You can use the Modbus read nodes (FC) to read data from the server buffers.



Step3: After write data into the Modbus server, you can read data from the Modbus address of the Modbus Server



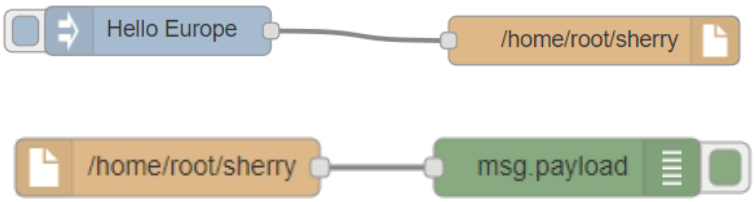
Sample flow

[{"id":"9bc81d49.0a6a9","type":"modbus-server","z":"d5035c6b.b91a8","name":"Modbus Server for ADAM-6700","logEnabled":true,"serverPort":"502","responseDelay":100,"delayUnit":"ms","coilsBufferSize":"20","holdingBufferSize":"20","inputBufferSize":"20","x":400,"y":120,"wires":[["cf3e61ad.3c3f2"],["cc9b5e9a.4f88c"],["3e6a0c7c.5ffb44"]]},{"id":"e65788ec.d6d4f8","type":"modbus-write","z":"d5035c6b.b91a8","name":"Write data to Modbus tcp server","showStatusActivities":false,"unitid":"","dataType":"HoldingRegister","adr":"0","quantity":"1","server":"bc14bc5.e80d24","x":398,"y":263,"wires":[["4eab6dda.f05b74"],["4eab6dda.f05b74"]]},{"id":"9fc9c1c5.0d0b6","type":"inject","z":"d5035c6b.b91a8","name":"","topic":"","payload":"100","payloadType":"num","repeat":"","crontab":"","once":false,"x":150,"y":260,"wires":[["e65788ec.d6d4f8"]]},{"id":"4eab6dda.f05b74","type":"debug","z":"d5035c6b.b91a8","name":"","active":true,"console":"false","complete":"true","x":670,"y":260,"wires":[]},{"id":"cf3e61ad.3c3f2","type":"debug","z":"d5035c6b.b91a8","name":"","active":false,"console":"false","complete":"true","x":65

5,"y":94,"wires":[]},{"id":"cc9b5e9a.4f88c","type":"debug","z":"d5035c6b.b91a8","name":"","active":false,"console":"false","complete":"true","x":655,"y":134,"wires":[]},{"id":"3e6a0c7c.5ffb44","type":"debug","z":"d5035c6b.b91a8","name":"","active":false,"console":"false","complete":"true","x":655,"y":174,"wires":[]},{"id":"7317ef04.5a713","type":"debug","z":"d5035c6b.b91a8","name":"","active":true,"console":"false","complete":"payload","x":670,"y":340,"wires":[]},{"id":"2dc38a0.411e476","type":"inject","z":"d5035c6b.b91a8","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"x":147,"y":140,"wires":[["9bc81d49.0a6a9"]]},{"id":"953b4fca.4c061","type":"modbus-read","z":"d5035c6b.b91a8","name":"Read data from Modbus TCP server","showStatusActivities":false,"unitid":"","dataType":"HoldingRegister","adr":"0","quantity":"5","rate":"1","rateUnit":"s","server":"bc14bc5.e80d24","x":400,"y":340,"wires":[["7317ef04.5a713"],["7317ef04.5a713"]]},{"id":"df896890.fd1eb8","type":"get-dio-value","z":"d5035c6b.b91a8","name":"","x":420,"y":560,"wires":[[],[],[],[],["73c51673.7a6368"],[],[],[],[]]},{"id":"73c51673.7a6368","type":"modbus-write","z":"d5035c6b.b91a8","name":"Write data to Modbus tcp server","showStatusActivities":false,"unitid":"","dataType":"Coil","adr":"0","quantity":"1","server":"bc14bc5.e80d24","x":730,"y":560,"wires":[[],[]]},{"id":"97333c77.66f61","type":"inject","z":"d5035c6b.b91a8","name":"","topic":"","payload":"","payloadType":"date","repeat":"0.5","crontab":"","once":false,"x":210,"y":560,"wires":[["df896890.fd1eb8"]]},{"id":"25712983.e97286","type":"set-do-value","z":"d5035c6b.b91a8","name":"","write_ch":"3","write_ch_type":"write_do_n","x":740,"y":700,"wires":[[]]},{"id":"4e01fbf9.396c44","type":"modbus-read","z":"d5035c6b.b91a8","name":"","showStatusActivities":false,"unitid":"","dataType":"Coil","adr":"0","quantity":"4","rate":"1","rateUnit":"s","server":"bc14bc5.e80d24","x":180,"y":700,"wires":[["4dff2a4d.d19c04","e63ed9cb.99d448"],["e74f6e99.36421"]]},{"id":"4dff2a4d.d19c04","type":"debug","z":"d5035c6b.b91a8","name":"","active":false,"console":"false","complete":"true","x":450,"y":760,"wires":[]},{"id":"e63ed9cb.99d448","type":"function","z":"d5035c6b.b91a8","name":"Slice array payload","func":"//var DOout = msg.payload[0];\nvar DOout = msg.payload.slice(0,4);\nmsg.payload=DOout;\nreturn msg;","outputs":1,"noerr":0,"x":500,"y":700,"wires":[["25712983.e97286"]]},{"id":"e74f6e99.36421","type":"debug","z":"d5035c6b.b91a8","name":"","active":false,"console":"false","complete":"true","x":450,"y":800,"wires":[]},{"id":"bc14bc5.e80d24","type":"modbus-client","z":"","name":"","clienttype":"tcp","bufferCommands":true,"stateLogEnabled":false,"tcpHost":"127.0.0.1","tcpPort":"502","tcpType":"DEFAULT","serialPort":"/dev/ttyUSB","serialType":"RTU-BUFFERD","serialBaudrate":"9600","serialDatabits":"8","serialStopbits":"1","serialParity":"none","serialConnectionDelay":"100","unit_id":"1","commandDelay":"1","clientTimeout":"1000","reconnectTimeout":"2000"}]

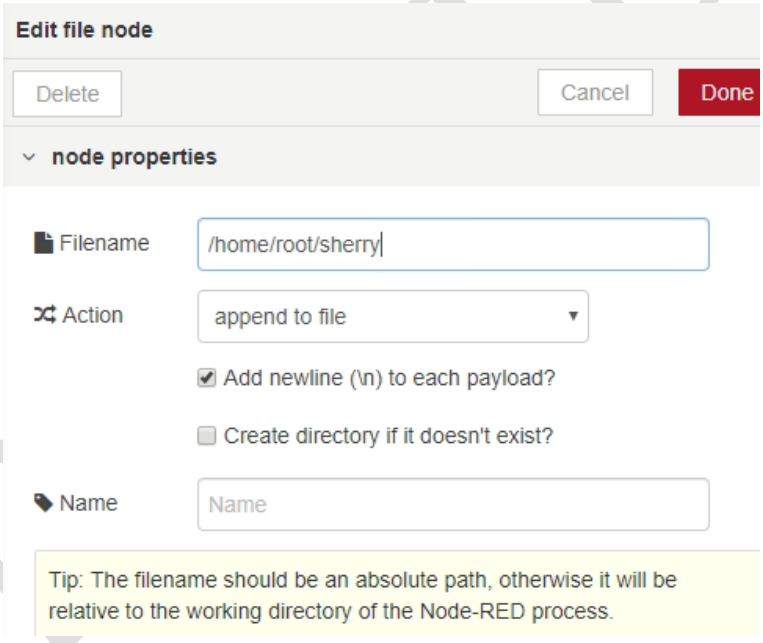
5.2 Data logger function

Node-red flow example:



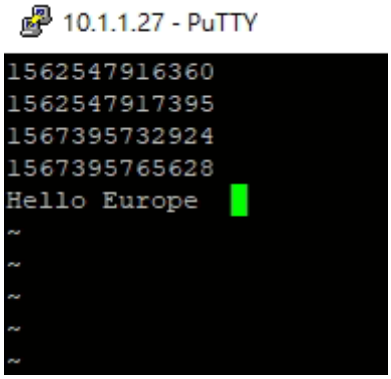
Used nodes & Steps:

- Inject Node: Enter String “Hello Europe”
- File Node: Refer to below settings

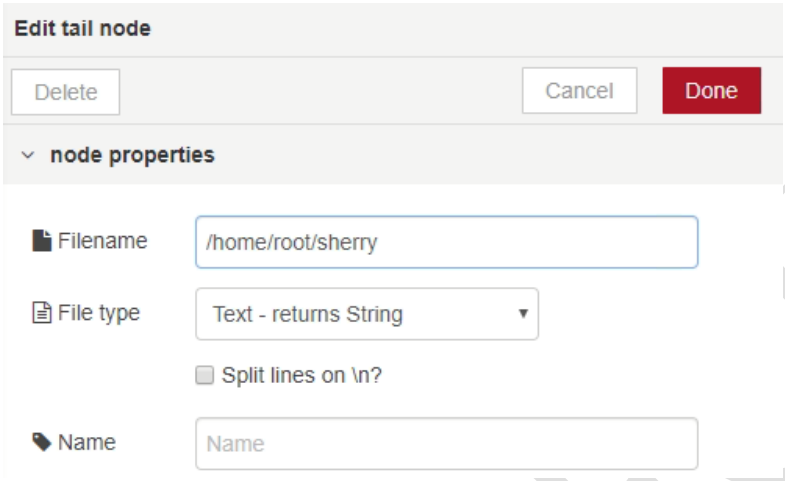


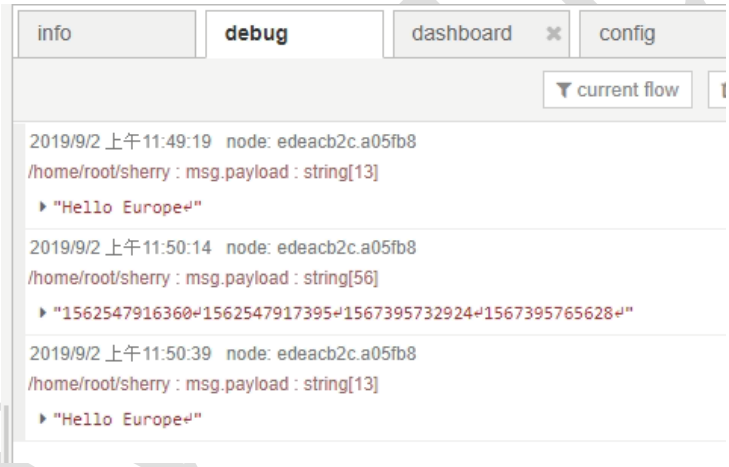
Test Result

See the file create in /home/root/sherry in ADAM-6700, and the content is as below



- Debug Node: See the result that write to the file
- Tail Node: Refer to below settings





Sample flow

[{"id":"cc93ab7b.e29e08","type":"file","z":"9b37fd57.44f52","name":"","filename":"/home/root/sherry","appendNewline":true,"createDir":false,"overwriteFile":"false","x":490,"y":160,"wires":[]},{"id":"a2d14320.fbf6e","type":"inject","z":"9b37fd57.44f52","name":"","topic":"","payload":"Hello Europe","payloadType":"str","repeat":"","crontab":"","once":false,"x":220,"y":154,"wires":[["cc93ab7b.e29e08"]]},{"id":"71928706.265c78","type":"tail","z":"9b37fd57.44f52","name":"","filetype":"text","split":false,"filename":"/home/root/sherry","x":140,"y":300,"wires":[["edeacb2c.a05fb8"]]},{"id":"edeacb2c.a05fb8","type":"debug","z":"9b37fd57.44f52","name":"","active":true,"console":"false","complete":"false","x":350,"y":300,"wires":[]}]