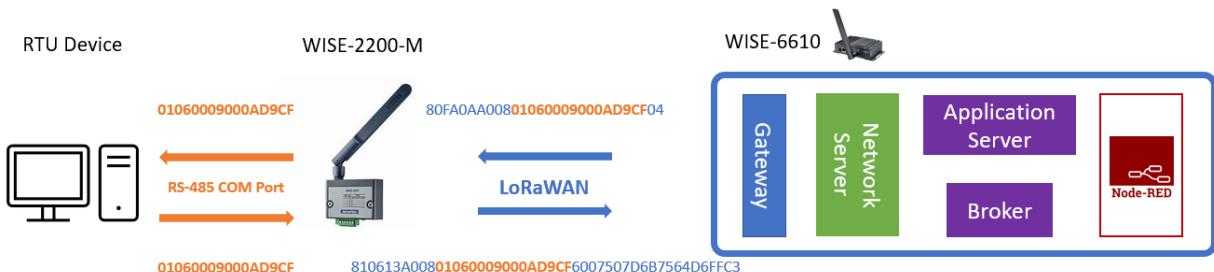


Advantech AE Technical Share Document

Date	2023/04/07	SR#	1-5206932612
Category	<input checked="" type="checkbox"/> FAQ <input type="checkbox"/> SOP	Related OS	N/A
Abstract	How to Use WISE-2200-M Transparent Mode to Send Uplink or Downlink of COM Port Data of WISE-2200-M ?		
Keyword	Downlink, COM Port		
Related Product	WISE-2200-M, WISE-6610		

■ Brief Description

WISE-2200-M is a LoRaWAN device supporting Modbus read and write functions. Besides, WISE-2200-M supports transparent mode, which means WISE-2200-M can encapsulate the received frames on COM port into the uplink message and forward it to the LoRaWAN Gateway. It also can decapsulate received downlink message to get and forward transmitted frames to connected device through COM port. Just like below figure.



Therefore, this document will describe how to use transparent mode of WISE-2200-M and take an experiment of above topology.

■ Brief Solution

This document will be divided into two sections shown as below.

1. What meaning of each parameter of transparent mode setting is on WISE-2200-M.
2. Experiment: Using transparent mode to communicate with Modbus RTU client of WISE-2200-M through WISE-6610

Section 1: What meaning of each parameter of transparent mode setting on WISE-2200-M

The below figure shows what parameter should be set for transparent mode on WISE-2200-M.

The path is **IO Status**. And there are descriptions for each parameters shown as below.

The screenshot shows the 'IO Status' configuration page for COM1. The 'Protocol' is set to 'Transparent'. Other settings include Baud rate (9600 bps), Data Bit (8 bit), Parity (None), Stop Bit (1 bit), Response Timeout (1500 ms), Minimum Payload Length (5), Start Character, End Character, Length (0), Break Time (500 ms), and Maximum Length of Under Processing Payload (256). A note indicates to send or drop if over the maximum length. A 'Submit' button is at the bottom right.

Parameter	Value	Description
a Protocol	Transparent	Protocol type
b Baud rate	9600 bps	Baud rate
c Data Bit	8 bit	Data Bit
d Parity	None	Parity
e Stop Bit	1 bit	Stop Bit
f Response Timeout	1500 ms	Response Timeout
g Minimum Payload Length	5	Minimum Payload Length
h Start Character		Start Character
i End Character		End Character
j Length	0	Length
k Break Time	500 ms	Break Time
l Maximum Length of Under Processing Payload	256	Maximum Length of Under Processing Payload
m Note	<input checked="" type="radio"/> Send if over the Maximum Length of Under Processing Payload <input type="radio"/> Drop if over the Maximum Length of Under Processing Payload	Note for payload length
n ✓ Submit		Submit button

- a. **Protocol:** This parameter indicates which mode WISE-2200-M is on and, so far, there are two modes, which are **Modbus/RTU (Client) mode** and **Transparent mode**. In this document, the option is set as **Transparent**.
- b. **Baud rate:** This parameter refers to speed of communication over a RS-485 (COM port). In this document, the option is set as **9600bps**.
- c. **Data Bit:** The parameter refers to data is transmitted in N-bit bytes, one bit at a time. In this document, the option is set as **8 bit**.
- d. **Parity:** The parameter is used to detect errors that may occur during transmission of data. The Parity can be set to none, odd or even. In this case, the parameter is set as **None**.
- e. **Stop Bit:** The parameter is used to indicate the end of a data frame. In this document, the option is set as **1 bit**.
- f. **Response Timeout (ms):** The parameter refers to the maximum time WISE-2200-M wait for a reply from the connected device through COM port. In this document, the parameter

is set as “**1000**” ms.

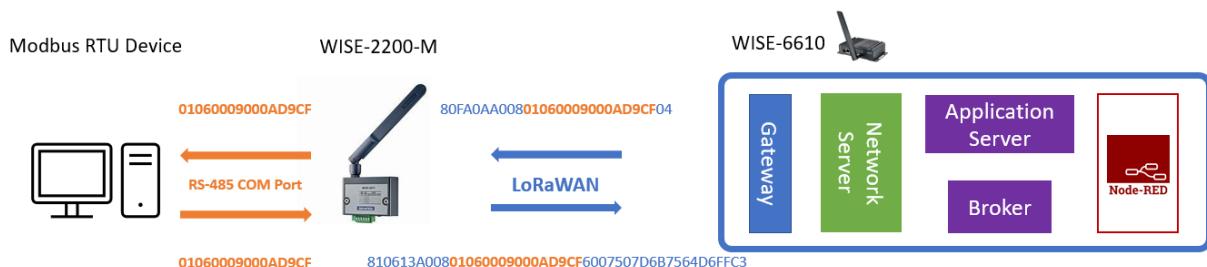
Note: The parameter is used when WISE-6610 sends a downlink to ask WISE-2200-M to send serial frame to connected RTU device.

- g. **Minimum Payload Length (bytes):** The parameter refers to the minimum length of payload on COM port the WISE-2200-M allows; otherwise, WISE-2200-M will ignore the data frames. In this document, the value is set as “**5**”.
- h. **Start Character:** The parameter is used to let WISE-2200-M identify whether the payload is valid depending on if initial character(s) of payload match the Start Character. In this document, the parameter leaves **blank**, meaning disable this function.
- i. **End Character:** The parameter is used to let WISE-2200-M identify whether the payload is valid depending on if final character(s) of payload match the End Character. In this document, the parameter leaves **blank**, meaning disable this function.
- j. **Length:** The parameter is used to let WISE-2200-M identify whether the payload is valid depending on if length of payload is equal to or larger than the Fixed Length. In this document, the parameter leaves **0**, meaning disable this function.
- k. **Break Time (ms):** The parameter is used to let WISE-2200-M identify whether the payload transmission is finished depending on if WISE-2200-M has wait for signal for the Break Time. In this document, the parameter is set as “**500**” ms.
- l. **Maximum Length of Under Processing Payload (bytes):** The parameter is used to let WISE-2200-M identify whether the payload is complete message depending on if length of payload is shorter than to the Maximum Length. In this document, the parameter is set as **256**.
- m. **Send or Drop Large Payload:** The user can choose whether WISE-2200-M send the payload whose length is greater than **Maximum Length of Under Processing Payload**. In this document, the parameter is set as **send**.
- n. **Submit:** The button is used to save setting of WISE-2200-M.

Note: If one of **End Character**, **Fixed Length**, or **Break Time** is fulfilled, WISE-2200-M will identify the serial frame is finished transmitted.

Section 2: Experiment: Using transparent mode to communicate with Modbus RTU client of WISE-2200M trough WISE-6610

For this experiment, the user will learn how to use Application Server of WISE-6610 to send downlink of serial frame to Modbus RTU device, and use Node-Red in WISE-6610 to check the response sent from WISE-2200-M. The topology of this experiment is shown as below.



For this experiment, it will be divided into three parts:

Part 1: Build connection between WISE-2200-M and WISE-6610

Part 2: Send write function of Modbus RTU frame from WISE-6610 Application

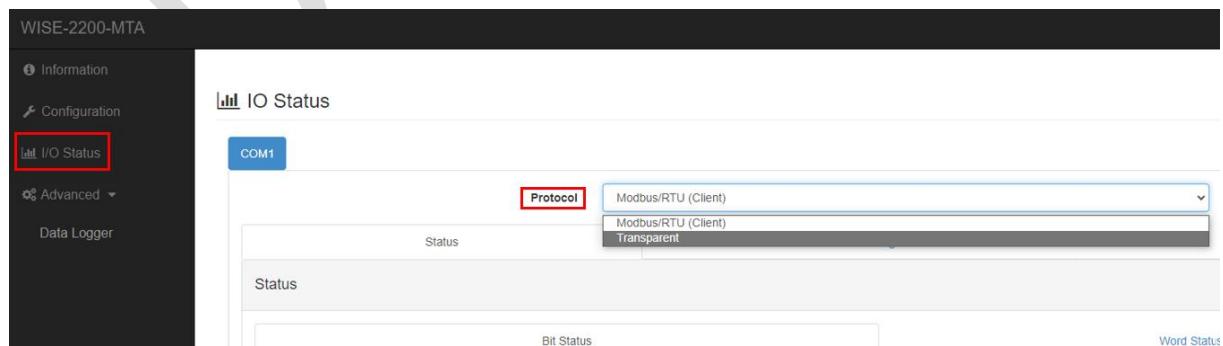
Part 3: Use Node-Red in WISE-6610 to check response of Modbus RTU frame.

Part 1: Build connection between WISE-2200-M and WISE-6610

Step 1: Please refer to below FAQ to learn how to fast connect WISE-2200-M to the WISE-6610 by using the Wizard tool of WISE-6610.

FAQ: <https://www.advantech.com/en/support/details/faq?id=1-29KPEKB>

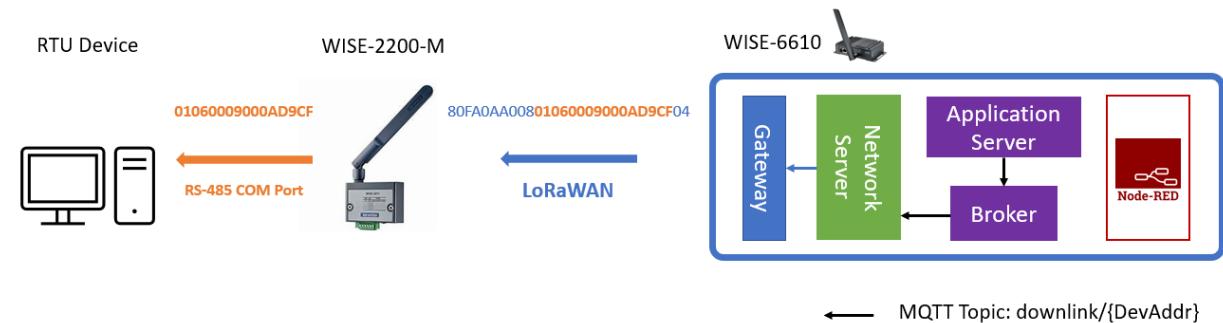
Step 2: On **IO Status** page of WISE-2200-M, please set WISE-2200-M as Transparent mode. Just like below figure.



Step 3: After converting the WISE-2200-M to Transparent mode, please refer to Section 1 of this document to check what meaning of each parameter is and, for this experiment, what value we set for each parameter.

Note: Please make sure Data+ pin and Data- pin of WISE-2200 are connected to Data+ pin and Data- pin of Modbus RTU client respectively.

Part 2: Send write function of Modbus RTU frame from WISE-6610 Application Server



Step 4: On the User Module > LoRaWAN Gateway > Application Server > Status page on WISE-6610, the user should see WISE-2200-M information after WISE-2200-M successfully connects to WISE-6610. And please hit the **Setting** button to set for downlink message. Just like below figure.

The screenshot displays two main pages of the WISE-6610 web interface:

- LoRaWAN Gateway Settings:** Shows the **Application Server Status** section with the following details:

Index	DevAddr	Battery	Model	Received	Fcnt	Packet Loss	Rssi	Bad Signal	Action
1	FF622FA1	Unknown	WISE2200-M	2023-06-06T06:52:56Z	45	0.00(%) from fcnt 2	-27	0.00(%)	Setting
- Application Server:** Shows the **Status** section with a link to **Settings**.
 - Navigation:** Includes links to Router, Wizard, LoRaWAN Radio, Network Server, MQTT, Storage, Application Server (with **Settings** and **Status** sub-links), Modbus Mapping Table, Payload Engine, and Licenses.
 - Application Log:** Buttons for Refresh and Clear log.

Step 5: When setting page of downlink for WISE-2200-M pop up, the user can choose different type of downlink message. Please choose **Transparent** for this experiment. Just like below figure

Navigation		LoRaWAN Gateway Settings	
		Node Setting Select	
		Devaddr	FF622FA1
		Model	WISE2200-M
		Function	Transparent
			<input type="button" value="Select"/> <input type="button" value="Return"/>

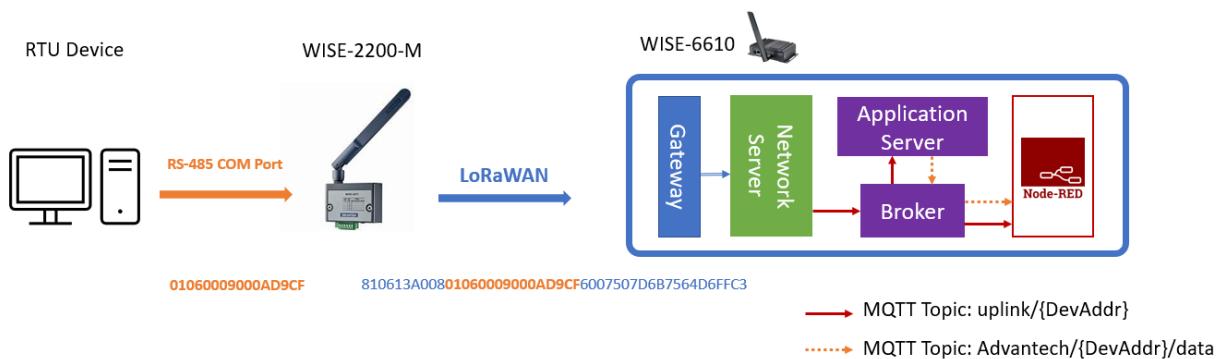
After choosing the transparent type of downlink message, there are some parameters that should be filled in. Just like below figure, and there is also description for each parameter.

Navigation		LoRaWAN Gateway Settings	
		Information	
		Devaddr	FF622FA1
		Time a	Immediately(only Class C Support)
		Confirmed b	Unconfirmed Data
		Function c	Transparent
		Transparent	
		Data	d
			<input type="button" value="Set"/> <input type="button" value="Return"/>

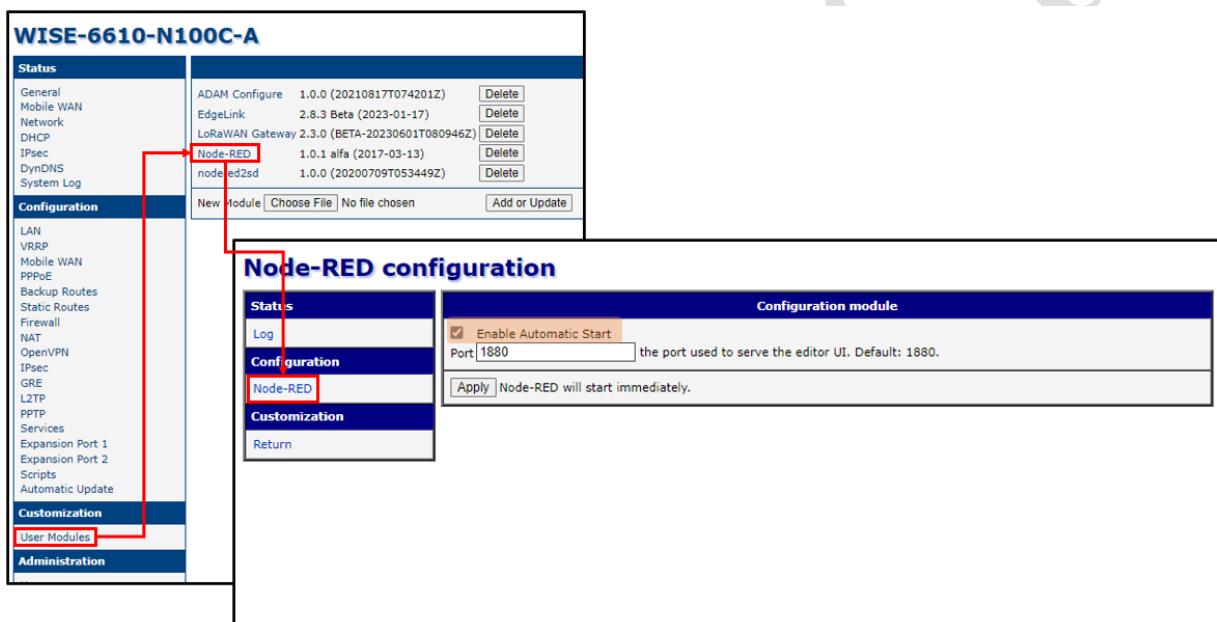
- a. **Time:** This parameter refers to whether the downlink message will be sent immediately. Otherwise, WISE-6610 will queue the downlink message until next WISE-2200-M uplink message. In this case, the parameter is set as **Immediately(only Class C Support)**.
- b. **Confirmed:** This parameter refers to whether WISE-6610 asks WISE-2200-M reply for this downlink message. In this case, the parameter is set as **Unconfirmed Data**.
- c. **Function:** This parameter refers to which type of function of downlink application the user uses. In this case, the parameter was set as **Transparent**.
- d. **Data:** This parameter refers to the serial frame of RS-485 will be sent from WISE-2200-M after WISE-2200-M receive this downlink message. In this case, WISE-2200-M will send Modbus write function to Modbus RTU device, so the parameter is set as **“01060009000AD9CF”**.

Note: The “01060009000AD9CF” means that Modbus RTU client writes the value 10 to the address 10, which is of length 1, for the Client ID 1.

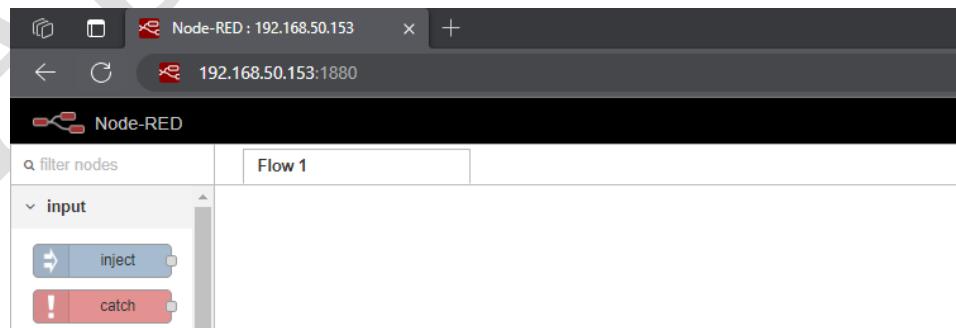
Part 3: Use Node-Red in WISE-6610 to check response of Modbus RTU frame.



Step 6: First, please install Node-Red User Module on WISE-6610 and make sure the checkbox, Enable Automatic Start, is checked.

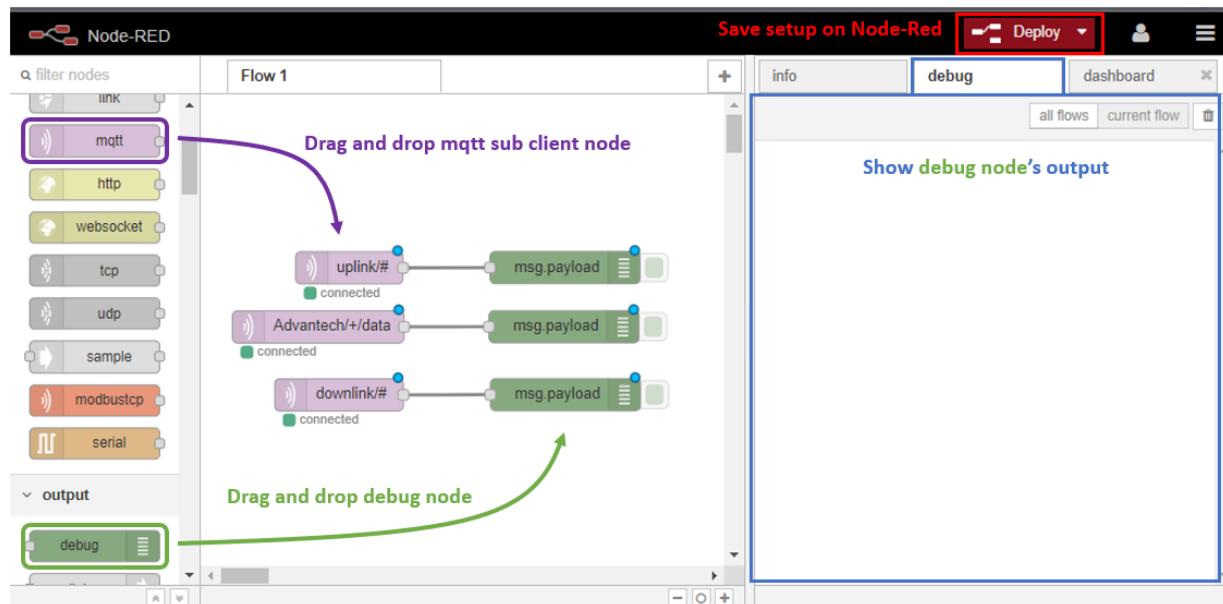


Step 7: Restart the WISE-6610 and go to Node-Red page by browsing {IP Address of WISE-6610}:1880, just like below picture.

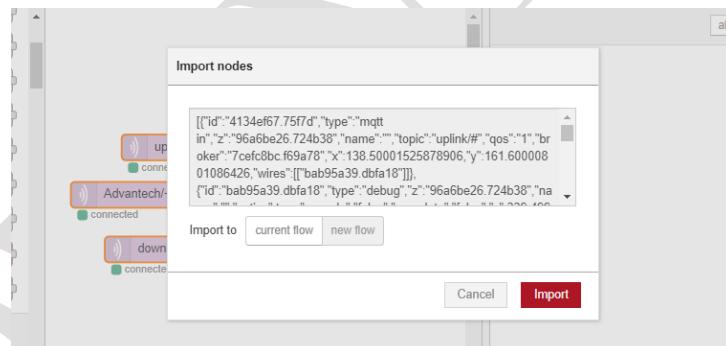


Step 8: Please drag and drop mqtt node and debug node like below screenshot in order to subscribe WISE-6610 internal broker and get message transmitted between WISE-2200-M and WISE-6610. And on right side, Node-Red will print out transmitted data.

Note: Please, finally hit **Deploy** button to save setup for this experiment.



Note: The user can directly use below Json words into your Node-RED on WISE-6610.



```
[{"id": "4134ef67.75f7d", "type": "mqtt", "in": "z": "96a6be26.724b38", "name": "", "topic": "uplink/#", "qos": "1", "broker": "7cefc8bc.f69a78", "x": 138.50001525878906, "y": 161.60000801086426, "wires": [{"id": "bab95a39.dbfa18"}], {"id": "bab95a39.dbfa18", "type": "debug", "z": "96a6be26.724b38", "name": "", "active": true, "console": "false", "complete": "false", "x": 339.49999237060547, "y": 161.3999969959259, "wires": []}, {"id": "3a041f1.57723e", "type": "mqtt", "in": "z": "96a6be26.724b38", "name": "", "topic": "Advantech+/data", "qos": "1", "broker": "7cefc8bc.f69a78", "x": 109.50001525878906, "y": 216.6000099182129, "wires": [{"id": "cbdf0ee5.c0b43"}]}, {"id": "cbdf0ee5.c0b43", "type": "debug", "z": "96a6be26.724b38", "name": "", "active": true, "console": "false", "complete": "false", "x": 338.49999237060547, "y": 216.4000129699707, "wires": []}, {"id": "b44a33c7.66631", "type": "mqtt", "in": "z": "96a6be26.724b38", "name": "", "topic": "downlink/#", "qos": "1", "broker": "7cefc8bc.f69a78", "x": 128.89584350585938, "y": 279.8888912200928, "wires": [{"id": "c3fc0657.527168"}]}, {"id": "c3fc0657.527168", "type": "debug", "z": "96a6be26.724b38", "name": "", "active": true, "console": "false", "complete": "false", "x": 339.8957748413086, "y": 279.6888790130615, "wires": []}, {"id": "7cefc8bc.f69a78", "type": "mqtt-broker", "z": "", "broker": "127.0.0.1", "port": "1883", "clientid": "", "usetls": false, "compatmode": true, "keepalive": "60", "cleansession": true, "willTopic": "", "willQos": "0", "willPayload": "", "birthTopic": "", "birthQos": "0", "birthPayload": ""}]
```

Step 9: After setup for Node-Red on WISE-6610, please carry out Step 5 again and the user can see the result from downlink message, uplink message and parsed uplink message.

The screenshot shows the Node-Red interface with the 'debug' tab selected. It displays three log entries:

- Downlink message sent to WISE-2200-M**
6/7/2023, 1:30:57 PM c3fc0657.527168
downlink/FF622FA1 : msg.payload : string [69]
{"data":"80000AA00801060009000AD9CF04","port":1,"time":"immediately"}
- Uplink message sent from WISE-2200-M**
6/7/2023, 1:30:58 PM bab95a39.dbfa16
uplink/FF622FA1 : msg.payload : string [176]
{"appargs":"WISE-2200-M","data":"810513A00801060009000AD9CF60075012168064DFFF84","datetime":"2023-06-07T05:30:58Z","devaddr":"FF622FA1","fcnt":5,"lsnr":7.8,"port":1,"rssl":-16}
- Parsed uplink message sent from WISE-2200-M**
6/7/2023, 1:30:58 PM cbdf0ee5.c0b43
Advantech/FF622FA1/data : msg.payload : string [83]
>{"TransParent":01060009000AD9CF,"Device":{"Time":1686115858}}