# Broadsens Modbus Guide

Broadsens Corporation

100 S Murphy Ave Ste 200

Sunnyvale, CA, USA, 94086

Revision 1.7

April 6, 2026

BroadVibra software version 2.9.3 SE Modbus

Broadsens

# Contents

# 1. Introduction

This document is for Modbus reference design only. It defines Modbus registers for Broadsens gateway, SVT-V series vibration & temperature sensors, and SVT200-T temperature sensor at Broadsens's wireless gateway. This document is for reference only (The register address definition can be modified by users at the gateway's control panel.) For high-speed and large data throughput applications such as SVT-A series sensors, MQTT communication protocol is recommended.

To use the Modbus example flows, and read Modbus registers, please use BroadVibra software that includes Modbus flows. The BroadVibra software that includes Modbus flows have "Modbus" letter in the file name.

Broadsens's gateway can function as both Modbus salve and Modbus master at the same time. When functioning as a slave, the gateway handles Broadsens's wireless sensors data, and save them to the registers. Another master can read the gateway's registers by following this document. When functioning as a master, the gateway can read registers from other Modbus slaves such as gateways or sensors.

# 2. Registers and addresses

This section introduces recommended Modbus register address and data conversion. The default Modbus TCP port for Broadsens gateway is 10502. The port number can be changed by referring to section 3.2.

## 2.1 Gateway information

The starting address of gateway information is 0.

Gateway information includes gateway serial ID, gateway type, CPU usage (in percentage), memory size & usage, temperature, location (latitude and longitude). The gateway information is compatible with MQTT protocol, but with limitation on the length of gateway name. Gateway information are written into holding registers. Use function code 3 to read the register values. The definition is as the following (Table 1):

*Table 1 Gateway information registers*

| **Register address** | **Description** |
|---|---|
| 0 | Gateway serial ID. |
| 1 | Gateway type |
| 2 | CPU usage (%) |
| 3 | Memory size in MB |
| 4 | Memory usage (%) |
| 5 | CPU temperature |
| 6 | Drive total space in MB |
| 7 | Drive used |
| 8 | Location latitude |
| 9 | Location longitude |
| 10-19 | Reserved |

CPU temperature can be obtained by dividing the value by 100. For example, if the register value is 5860, then the CPU temperature is 5860/100=58.60 degree Celsius. Latitude and longitude can be obtained by dividing the value by 100. Their units are degree.

**Note:** The gateway information requires user to turn on the "Gateway info" switch in the "MQTT config" page. Otherwise, the Modbus gateway information registers won't be updated.

## 2.2 SVT-V sensor data and information

Each SVT-V sensor needs 30 register addresses. The starting register address of SVT-V sensor is 100. Since the maximum number of SVT-V sensors in a gateway is 30, the highest starting register address of an SVT-V sensor is 970. The sensor data and information are written into the holding registers. Use function code 3 to read the register values.

Each SVT-V series sensor's data and information include: time stamp, sensor ID, sensor group, MAC address, RSSI, battery level, firmware version, vibration velocity RMS in x, y and z axis, acceleration RMS in x, y and z axis, and temperature measurement. Time stamp includes year, month, date, hour, minute and second, and it occupies 6 registers. Sensor ID occupies 1 register. The register definition is as the following (Table 2):

*Table 2 SVT-V sensor data and information registers*

| Register address | Description |
|---|---|
| 100-105 | Time stamp: Year, month, date, hour, minute and second |
| 106 | Sensor ID |
| 107 | Sensor group |
| 108-113 | MAC address |
| 114 | RSSI in dBm |
| 115 | Battery level |
| 116 | Firmware version |
| 117-119 | Vibration velocity RMS in x, y & z axis |
| 120-122 | Acceleration RMS in x, y & z axis |
| 123 | Temperature |
| 124 | Crest Factor (requires BroadVibra v2.9.3 or higher, SVT-V sensor firmware v3.0 or higher) |
| 125-129 | Reserved |
| 130-135 | Time stamp: Year, month, date, hour, minute and second |
| 136 | Sensor ID |
| …. | … |

**Note:** The Modbus sensor data requires user to turn on the "Save to database" switch in the home page of the gateway. Otherwise, no Modbus sensor data will be sent out.

The sensor's starting register address follows the order from the SVT-V sensor information table at the gateway. For example, there are five SVT-V sensors in the gateway, with sensor id: 206, 679, 1035, 2078 and 100069, then the starting register address of sensor 206 is 100, starting register address of sensor 679 is 130, …, and the starting register address of sensor 10069 is 220.

The data is in 16-bit unsigned data format (big endian). The data conversion are explained in the following.

The time stamp can be converted directly to year, month, date, hour, minute, and second. For example, Registers values [2024,1,18,20,29,38] means the data was taken at 20:29:38 on January 18, 2024.

The MAC address should be converted to HEX value. For example, the register values [209,162,75,66,9,105] corresponds to MAC address of d1:a2:4b:42:09:69.

The RSSI value is a negative number and can be obtained by subtracting value by 65535. The unit is dBm. For example, value of 65478 corresponds to -57 dBm (65478-65535).

The battery level can be obtained by dividing the number by 100. The unit is volt. For example, value of 364 corresponds to 3.64V (364/100).

The firmware version can be obtained by diving the value by 10. For example, number of 28 stands for version 2.8 (28/10).

The vibration velocity RMS can be obtained by dividing the value by 500. The unit is either in mm/s (Metric) or inch/s (Imperial) based on the unit selection from the gateway setup page. For example, assume that user selects Metric unit at the gateway, and values of 17,15,26 means that:

Velocity RMS in x axis is: 17/500=0.034 mm/s
Velocity RMS in y axis is: 17/500=0.034 mm/s
Velocity RMS in z axis is: 17/500=0.034 mm/s

The vibration acceleration can be obtained by dividing the value by 1000. The unit is g (9.81m/s$^2$). For example, values of 8,5,11 means that:

Acceleration RMS in x axis is: 8/1000=0.008 g
Acceleration RMS in y axis is: 5/1000=0.005 g
Acceleration RMS in z axis is: 11/1000=0.011 g

Temperature value should be scaled by 0.0078125 to obtain the correct reading. If the number is less than 32768, then user can multiply the value directly by 0.0078125 to obtain the temperature. If the number is larger than or equal to 32768, then it is negative temperature, user should minus 65535, then scale the result with 0.0078125. For example, the register value of 2859 corresponds to temperature of 2859*0.0078125=22.336 $^0$C. A register value of 63079 corresponds to temperature of (63079-65535)*0.0078125=-19.188 $^0$C.

Crest factor was added since BroadVibra v2.9.3. It requires SVT-V sensor firmware version v3.0 or higher, and gateway firmware version v2.9.0 or higher. Crest factor value can be obtained by dividing the value by 1000. For example, value of 1362 at register address 24 means that crest factor of the sensor is equal to 1.362.

Next SVT-V sensor's data and information starts from address 130. Each SVT-V sensor needs 30 register addresses. The order of the SVT-V sensor location is based on the sensor's ID. The smallest ID starts from address 100. The sensor with the largest ID occupies the last position.

Because the registers are updated in real time and the update speed could be much faster than Modbus TCP read interval, please handle this accordingly when user read the registers through Modbus.

## 2.3 SVT-T temperature sensor data and information

SVT-T sensor support is added starting from document version 1.5, gateway software version 2.8.1 SE Modbus.

Each SVT-T sensor occupies 20 registers. The starting register address of SVT-T sensor is 1,000. Since there are up to 100 SVT-T sensors in a gateway. The register address of SVT-T sensor is from 1,000 to 2,999. The register definition of SVT-T sensor is as the following (Table 3).

*Table 3 SVT-T sensor data and information registers*

| Register address | Description |
|---|---|
| 1000-1005 | Time stamp: Year, month, date, hour, minute and second |
| 1006 | Sensor ID |
| 1007 | Sensor group |
| 1008-1013 | MAC address |
| 1014 | RSSI in dBm |
| 1015 | Battery level |
| 1016 | Firmware version |
| 1017 | Temperature value |
| 1018-1019 | Reserved |
| 1020-1025 | Time stamp: Year, month, date, hour, minute and second |
| 1026 | Sensor ID |
| … | … |

The sensor's starting register address follows the order from the SVT-T sensor information table at the gateway. For example, there are four SVT-T sensors in the gateway, with sensor id: 29, 179, 180, and 3069, then the starting register address of sensor 29 is 1000, starting register address of 179 is 1020, …, and the starting register address of sensor 3069 is 1060.

The data is also unsigned 16-bit integer (big endian), and the data conversion is the same as the SVT-V sensor. For example, temperature value should be scaled by 0.0078125 to obtain the correct reading. If the number is less than 32768, then then user can multiply the value directly by 0.0078125 to obtain the temperature. If the number is larger than or equal to 32768, then it is negative temperature, user should minus 65535, then scale the result with 0.0078125.

For example, the register value of 2859 corresponds to temperature of 2859*0.0078125=22.336 0C. A register value of 63079 corresponds to temperature of (63079-65535)*0.0078125=-19.188 0C.

### 2.4 SVT-A sensor data and information

Because SVT-A series sensors transmit raw triaxial data, many DAQ modes (such as real time, batch, multiDAQ, live FFT) of SVT-A sensors require much higher speed and space than Modbus can handle. So MQTT protocol is recommended. But there are certain DAQ modes that Modbus could be able to process for a small number of SVT-A series sensors. The possible DAQ modes include single DAQ mode and single FFT mode.

In these modes, each SVT-A sensor's data could occupy up to 50k addresses. Please contact Broadsens if user want help to transmit SVT-A sensor's data & information with Modbus, which may require a turn key custom solution.

## 3. Gateway Modbus Nodes

**Attention: This section is for advanced users only.**

### 3.1 Modbus node install or update

For users who need Modbus communication, Modbus node is preinstalled at Broadsens's gateways. The name of the node is called "node-red-contrib-modbus". User can also install this node from the control panel of the gateway. Please refer to Appendix 4 of the gateway's manual on how to access the control panel:

https://www.broadsens.com/download/manuals/Wireless_gateway_operation_manual.pdf

The Modbus node information can be found at GitHub:

https://github.com/biancoroyal/node-red-contrib-modbus

Modbus node can be updated in the control panel too.

### 3.2 Chang default port number

The default Modbus port number is 10502. The Modbus TCP port number can be modified by going to the control panel of the gateway, and follow the steps below.

Step 1. Update Modbus server node port number. Open the control panel (please refer to Appendix 4 of the gateway manual on how to access the control panel). In the Modbus flow, double click the Modbus server (Figure 1) to modify the server setup (Figure 2). The default port number "10502" is shown in Figure 2. User can change the port number to the desired value such as 502.

Please note that in default, the port number should be larger than 1024 for security protection. If the default port number 10502 is used, or a port number large than 1,000 is used, then there is no need to modify the security setting. The gateway OS disabled ports below 1024 for security reason. For example, to enable port 502 access, please SSH to the gateway, and paste the following command:

```
sudo setcap 'cap_net_bind_service=+ep' `which node`
```

Restart the gateway, then port number below 1,000 is enabled. Users are recommended to use the a port number larger than 1,000 for enhanced security.

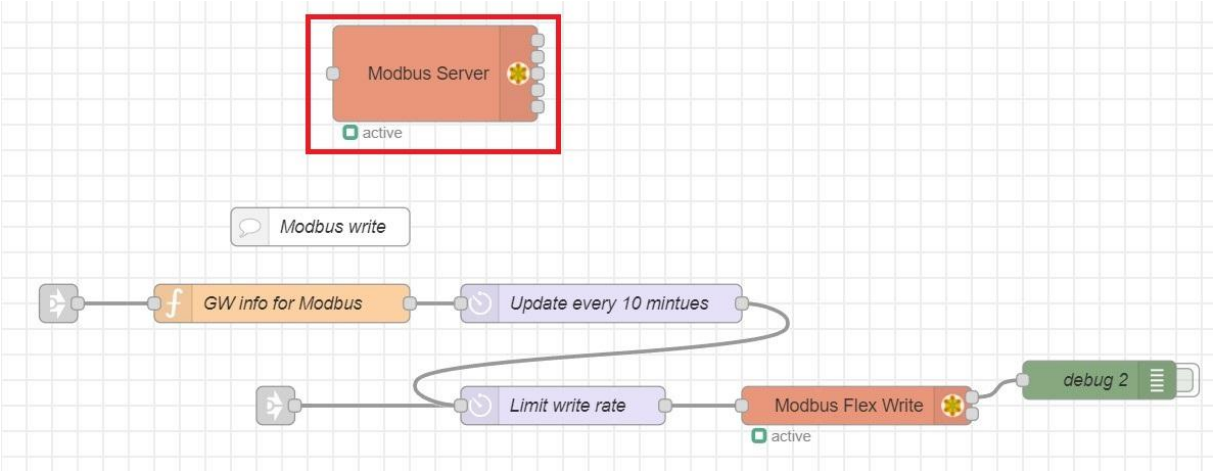


*Figure 1 Modbus server in the control panel*

Please click "Done" button to change the server node.

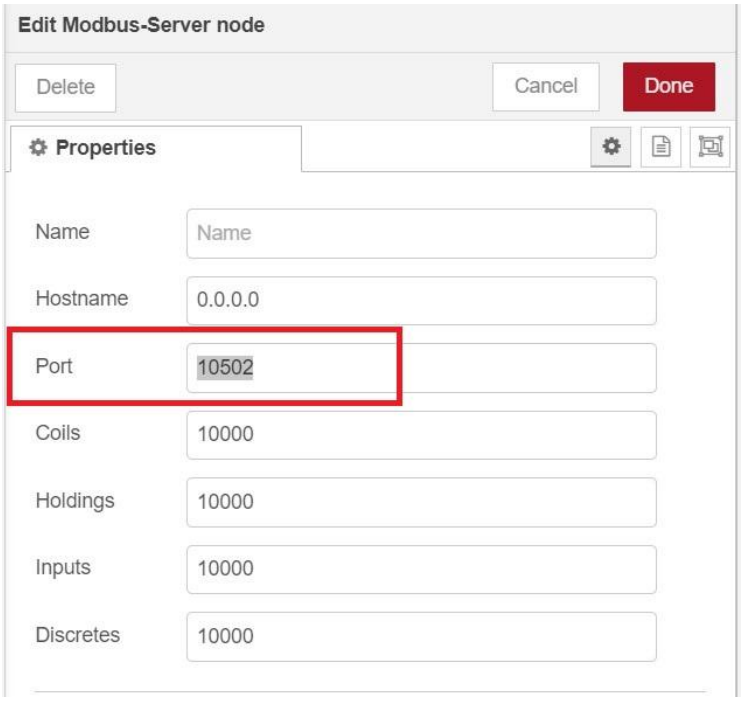


*Figure 2 Modbus TCP port number*

Step 2. Update write node port number. Double-click the “Modbus Flex Write” node (Figure 3), which opens the write node window (Figure 4). Click on the pencil sign beside “Modbus server” (Figure 4).

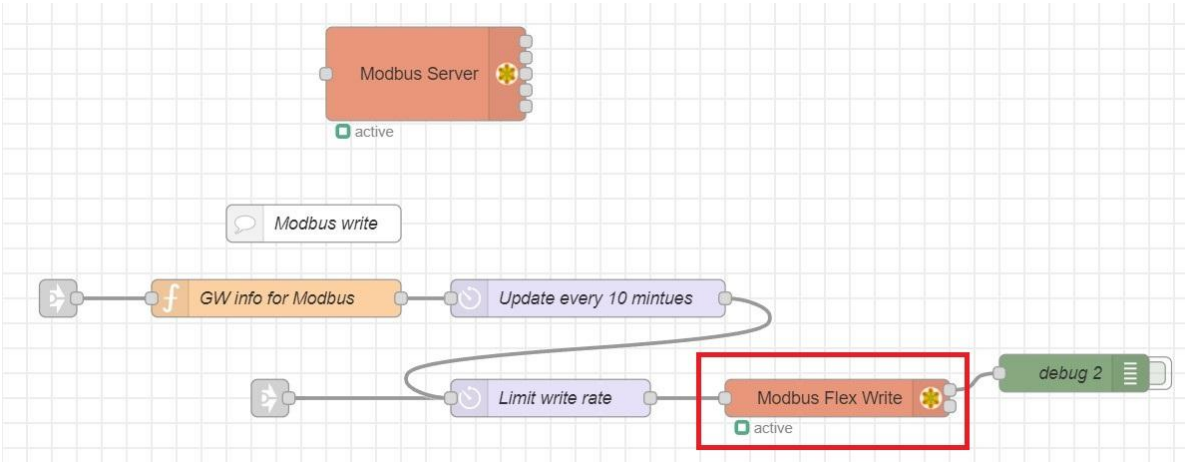


*Figure 3 Modbus Flex write node*

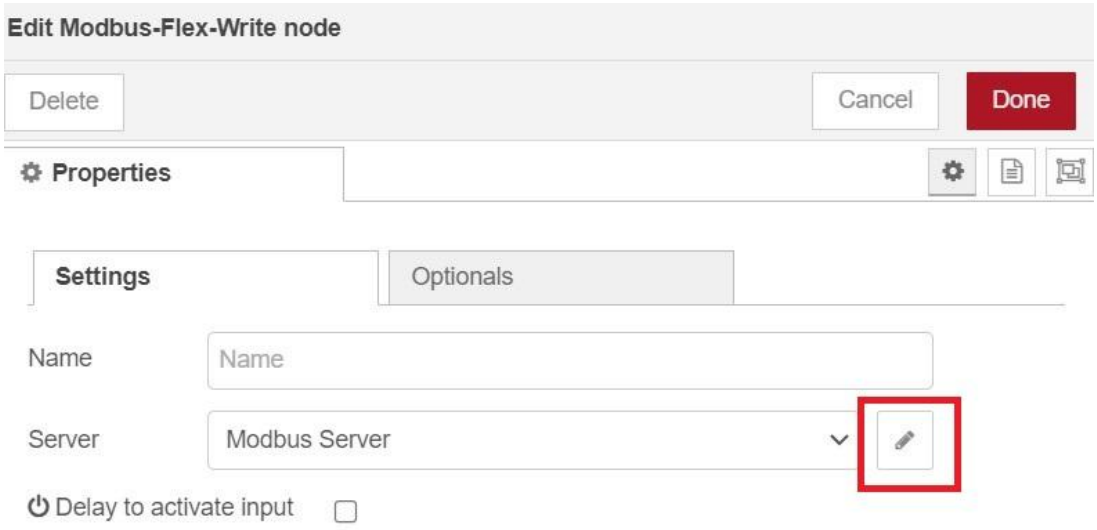


*Figure 4 Edit Modbus server inside Flex write node*

This brings up a new window to modify the writing port address (Figure 5). Change this port number to the desired port number too. Click “Update” button on the top right corner.

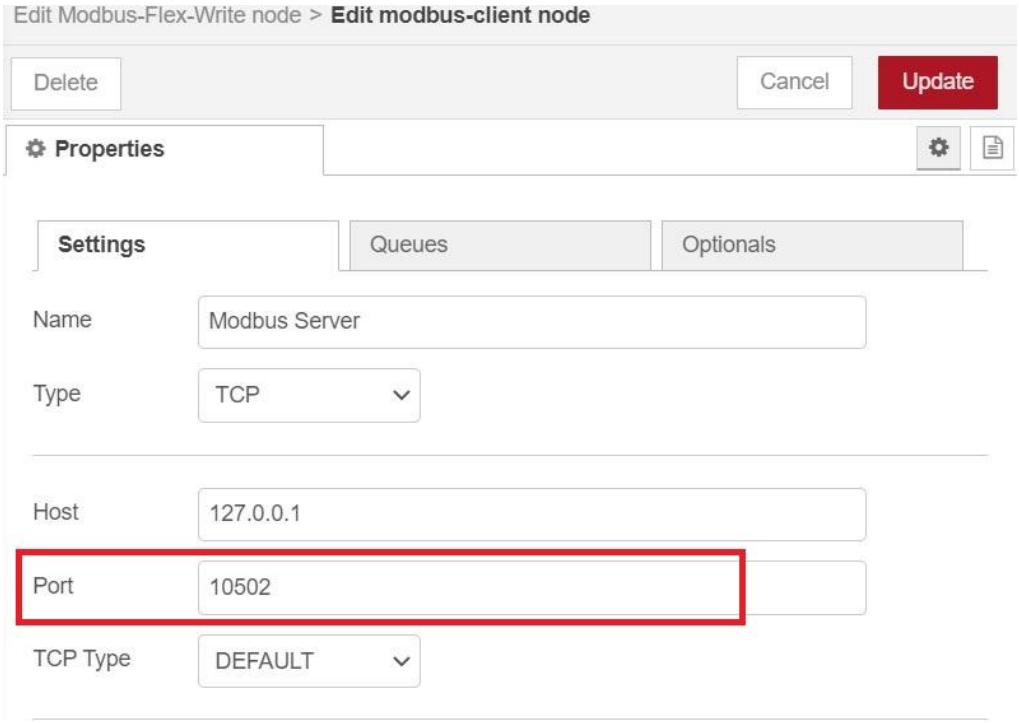


*Figure 5 Change Flex write node server port*

Step 3. Deploy changes. After the changes, the “deploy” button on the control panel’s top right corner will turn into red color (Figure 6). Please click the “deploy” button to deploy the change to the software. Then the TCP port is updated.

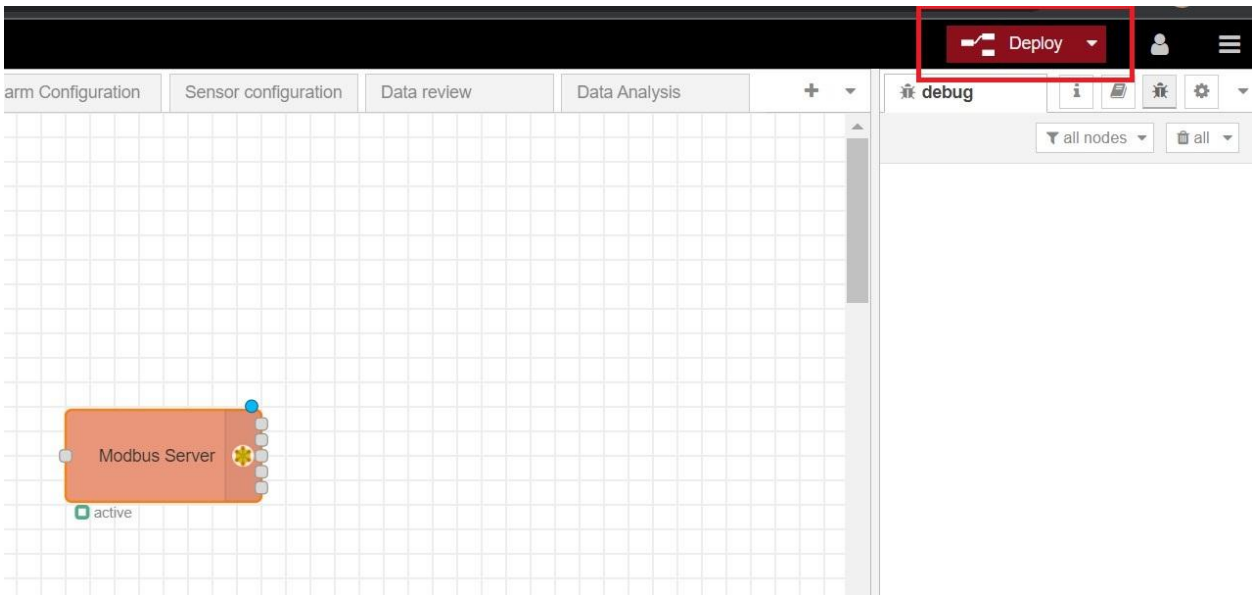


*Figure 6 "Deploy" button on the top right corner*

## 3.3 Modbus RTU slave node install

To use the gateway as Modbus RTU salve, “Modbus-RT-Slave” node is required. This node is not installed in default. To install “Modbus-RTU-Slave” node, please follow steps below:

### 3.3.1. Install Git

“Modbus-RTU-Slave” node requires to install “Git” package. Please SSH to the gateway, then enter commands below to install “Git”:

sudo apt install git

Please note that the gateway needs to access internet to install “Git” package.

### 3.3.2 Install Modbus-RTU-Slave Node

Enter gateway’s control panel (Please refer to Appendix 4 of the gateway’s manual on how to access the control panel), on the right side from the dropdown menus, click on “manage palette” option (Figure 7)→ click on “install” tab, search for “node-red-modbus-rtu-slave” node, and install the node.


*Figure 7 Manage palette in the gateway control panel*

Please note that it takes several minutes to finish installation of the node.

If the installation is successful, then the following two additional nodes will appear in the Modbus section of the control panel nodes on the left side (Figure 8).

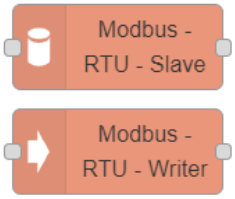


*Figure 8 Installed Modbus RTU nodes*

## 4. Modbus RTU Slave

Modbus RTU slave supports both SVT-V sensors and SVT-T sensors. To use gateway's Modbus RTU slave function, please do the following:

1. Follow section 3.3 to install Modbus RTU slave node.

2. Use provided Modbus RTU Slave installation program, in “Gateway setup” page→ “Gateway update” → “select file to update BroadVibra” to install the BroadVibra Modbus RTU slave version software.
3. Reboot the gateway after the software installation.

Then gateway functions as the Modbus RTU slave. Modbus RTU master example flows will be uninstalled to avoid conflicts. The register address definition is the same as the Modbus TCP. Please refer to section 2.1-2.3 for register details.

The Modbus RTU slave’s baud rate is 115,200, as shown in Figure 9. Other baud rates such as 9,600 can also be used. User can change the gateway’s RTU slave baud rate depending on Modbus RTU master’s setup.

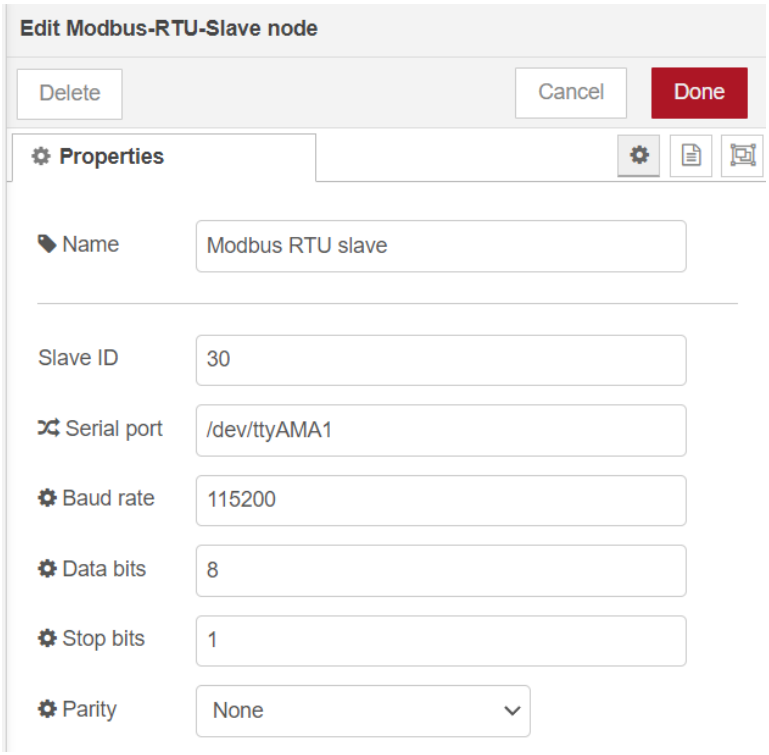


*Figure 9 Modbus RTU node setup*

To test and read the gateway’s Modbus RTU register data, please use BroadVibra Modbus SE version v2.8.3 or later. Only two wires are required to connect the gateway Modbus RTU master to the gateway Modbus RTU slave. One wire connects RS485 connector “A” to the other gateway’s connector “A”; another wire connects RS485 connector “B” to the other gateway’s connector “B”.

In the Modbus RTU master example flows (Figure 10), the gateway functions as Modbus RTU master, and can read data from a gateway that functions as Modbus RTU slave. The Modbus RTU master example flows is as the following. Please use baud rate 115,200 that matches the Modbus RTU slave’s baud rate.

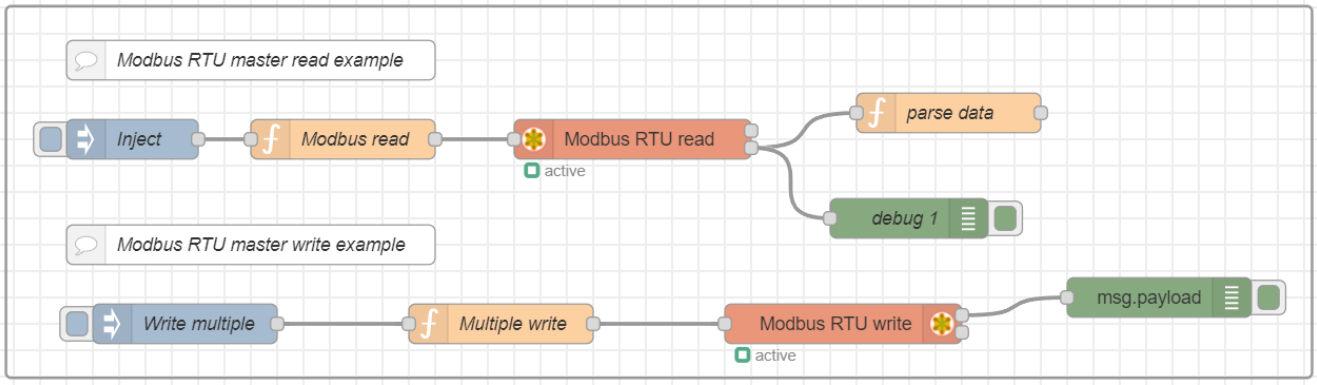


*Figure 10 Modbus RTU master read flow*

Click at “Inject” node, then the debug window will show the reading from the gateway Modbus RTU slave’s registers. If there are SVT-V sensors or SVT-T sensors connected to the gateway functioning as Modbus RTU slave, then user can see the sensor data.

For users who are interested in details of the Modbus RTU slave flows, please open the control panel of the gateway that functions as the Modbus RTU slave to see the flow details (Figure 11).

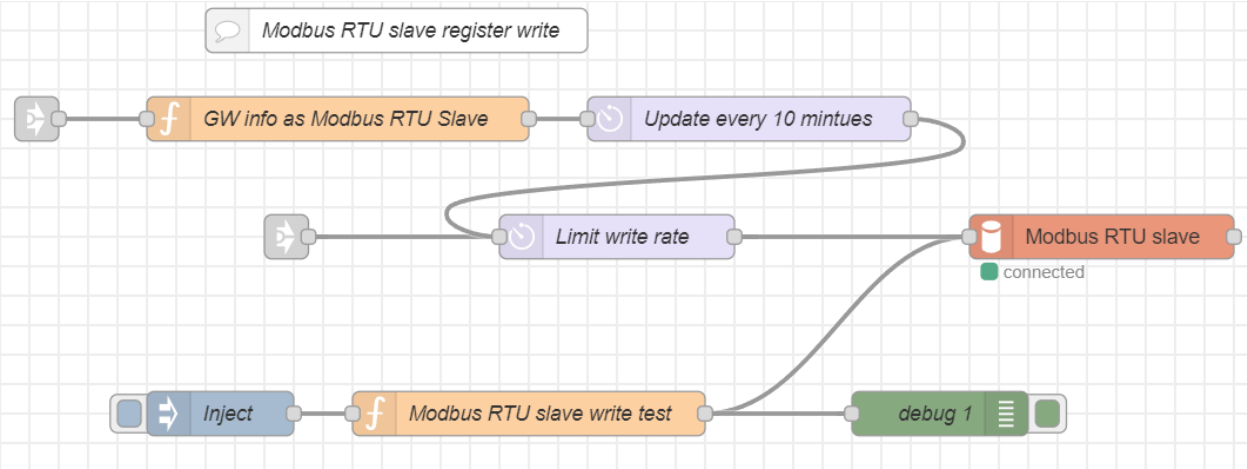


*Figure 11 Modbus RTU slave flows*

Currently, the “write rate” (Modbus RTU slave register update speed) is limited to 0.2s. User should play with the write rate to make sure that the system is stable.

The debug window shows the Modbus RTU slave node writing status. Please close the control panel after debugging, so that the gateway consumes minimum resource.

Please contact Broadsens’s tech support for details: support@broadsens.com

Broadsens

## Revision History

- V1.7. Added Crest factor support
- V1.6. Added Modbus RTU slave support documentation
- V1.5. Added SVT200-T Modbus support
- V1.1-1.4. Added Modbus TCP port change documentation. Improved Modbus TCP register update speed
- V1.0. Initial release