

# MODBUS TCP OVER ETHERNET

## 1. Introduction

Senquip devices feature an integrated Modbus RTU interface that communicates over RS232 or RS485. However, many industrial applications use Modbus TCP over Ethernet or Wi-Fi for networked communication.

Modbus TCP over Wi-Fi can be implemented on a Senquip device without the need for an adapter. This is covered in [APN0024 – Modbus TCP In a Script](#).

This application note demonstrates how a Senquip device can interface with a remote Modbus TCP sensor using a cost-effective RS485-to-Ethernet converter. The [PUSR USR-TCP232-304](#) RS485-to-Ethernet converter was selected for its affordability and widespread availability.



Figure 1 - PUSR USR-TSP232-304

## 2. Introduction to Modbus TCP 3.

### 2.1. What is Modbus TCP?

Modbus TCP is a widely used industrial communication protocol that enables Ethernet-based data exchange between devices such as sensors, controllers, and monitoring systems. It is an extension of the Modbus RTU protocol, which operates over RS232 or RS485, but instead utilizes TCP/IP as the transport layer.

- Key Features of Modbus TCP:
- Uses Ethernet/IP instead of serial communication.
- Follows a client-server architecture.
- Allows a single client to communicate with multiple servers (sensors, meters, or RTUs).
- Enables remote access over local networks or the internet.
- Uses port 502 as the default communication port.

### 2.2. Modbus TCP Client-Server Model

In Modbus TCP, devices follow a client-server model:

**Client** – Initiates communication by sending requests to read or write data. Examples: PLC, SCADA system, Senquip Device

**Server** – Holds data and responds to client requests. Examples: Sensors, meters, remote I/O modules

|                                   |                 |                    |                   |
|-----------------------------------|-----------------|--------------------|-------------------|
| Document Number<br>APN0038        | Revision<br>1.0 | Prepared By<br>NGB | Approved By<br>NB |
| Title<br>Modbus TCP over Ethernet |                 |                    | Page<br>2 of 8    |

Most Modbus TCP sensors act as servers, waiting for a client (such as a Senquip QUAD) to request data at regular intervals. The client polls the sensor for measurements like temperature, pressure, or voltage and uses the received data for monitoring or control.

Example Communication Flow:

1. Client (Senquip QUAD) → Server (Modbus TCP Sensor): "Provide the value of register 1 (temperature reading)."
2. Server (Modbus TCP Sensor) → Client (Senquip QUAD): "Register 1 = 25°C"

This request-response cycle ensures that the client receives up-to-date sensor readings.

In some applications, a single Modbus TCP server may support multiple clients (e.g., both a SCADA system and a PLC requesting data from the same sensor). However, the most common scenario involves a single client polling multiple sensors.

### 2.3. Differences Between Modbus TCP and Modbus RTU

Since **Modbus TCP** runs over Ethernet, it supports multiple clients and servers, remote access, and faster communication speeds compared to **Modbus RTU**.

| Feature        | Modbus RTU (RS232/RS485)     | Modbus TCP (Ethernet)                     |
|----------------|------------------------------|-------------------------------------------|
| Physical Layer | RS232/RS485 (Serial)         | Ethernet (TCP/IP)                         |
| Data Format    | Binary (RTU)                 | TCP/IP packets                            |
| Addressing     | Device ID (1-247)            | IP Address                                |
| Communication  | Master-Slave (single master) | Client-Server (multiple clients possible) |
| Speed          | Typically, up to 115.2 kbps  | Up to 1 Gbps or higher                    |
| Network Size   | Limited (serial bus)         | Scalable (IP network)                     |

## 3. Connecting the RS485 to Ethernet Interface

In this example, the RS485-to-Ethernet converter was connected to the RS485 port of a Senquip QUAD, with an Ethernet cable linking the converter to a PC. The converter was powered using the included power adapter.

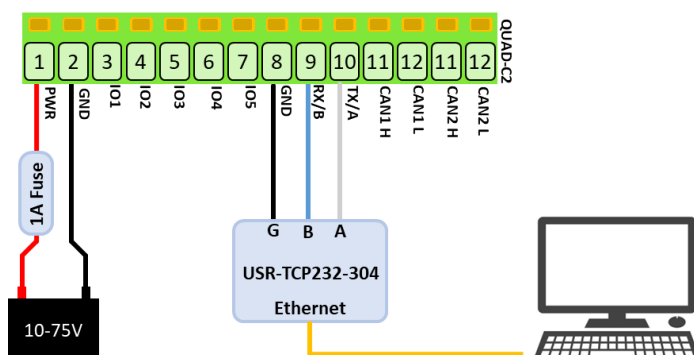


Figure 2 - RS485 to Ethernet Connections

The configuration software, USR-M0, shown in Figure 3, was [downloaded](#) from the PUSR website. The converter was assigned an available IP address (169.254.203.199) within the same subnet as the PC's Ethernet connection. The PC's

|                                   |                 |                    |                   |
|-----------------------------------|-----------------|--------------------|-------------------|
| Document Number<br>APN0038        | Revision<br>1.0 | Prepared By<br>NGB | Approved By<br>NB |
| Title<br>Modbus TCP over Ethernet |                 |                    | Page<br>3 of 8    |

Ethernet IP address was determined using the ipconfig command from a command prompt. A static IP address must be used in this case as the PC is not running a DHCP server.

```
Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : lan
Link-local IPv6 Address . . . . . : fe80::2108:5723:ea9c:d0dc%17
Autoconfiguration IPv4 Address. . : 169.254.203.184
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
```

Figure 3 - PC IP Address on Ethernet Adapter

[Modbus Slave](#) was used as a Modbus TCP server to which the interface will connect.

The settings for the RS485 to ethernet interface are shown in Figure 4. The most important settings updates are:

3. The interface is setup as a TCP client.
4. The RemoteIP is set as the IP address of the PC running the Modbus Server software.
5. The port is specified as the port on which the Modbus server software will be listening.
6. The baud rate and serial port settings match the QUAD.
7. The ModBus Type is set as Modbus TCP/RTU.

The interface (shown in Figure 5) also has a built in webserver through which the device can be configured. The webserver can be accessed at the allocated IP address using username: admin and password: admin.

The settings for the Modbus Slave software are shown in Figure 6. The most important settings are:

1. Mode is set as TCP.
2. The port is set as 502 to match that configured on the interface.
3. A Slave ID of 1 is chosen.
4. Registers 1 and 2 are set as 12 and 13 respectively.

Modbus Slave offers a handy communications log in which incoming and sent data can be seen.

The serial port settings for the Senquip QUAD are shown in Figure 7 and Figure 8. The most important settings are:

1. The serial port settings must match those specified for the RS485 to ethernet interface.
2. The Mode is set to Modbus.
3. Two reads are configured for registers addresses 1 and 2 on a slave device with address 1.

The image shows the USR-M0 V2.2.6.1 software interface. The top menu bar includes 'File', 'Language', and 'Help'. The main window is divided into several sections. On the left, there's a 'Device List' table with columns for Device IP, Device Name, MAC, and Ver... The table contains one entry: 169.254.203.1..., USR-TCP232-3..., D4 AD 20 7B 7C ..., 4020. Below the table is a 'Search Device' button. The bottom left section displays 'Operation Log' and 'Hex Streams'. The right side of the interface is dedicated to configuration parameters, organized into sections: 'Base Param' (IP Type, ModuleStaticIP, SubnetMask, Gateway, DNS Address, Reset Timeout, Clear Buffer Data Before Connected, UART Set Parameter, HTTP Port, User Name, Password, Device Name, Index, Reset, Link, RFC2217), 'Port Param' (Parity/Data/Stop, Module work mode, RemoteIP, Short Connection time, PackTime, Baudrate, Local Port, Remote Port, Tcp connect num, PackLen), 'Register' (Register Packet Type), 'Heartbeat' (Heartbeat Packet Type), and 'ModBus' (Modbus Type). At the bottom right, there are buttons for 'Save Config' and 'DataDebug'.

Figure 4 – RS485 to Ethernet Interface Settings

|                          |          |             |             |
|--------------------------|----------|-------------|-------------|
| Document Number          | Revision | Prepared By | Approved By |
| APN0038                  | 1.0      | NGB         | NB          |
| Title                    |          |             | Page        |
| Modbus TCP over Ethernet |          |             | 5 of 8      |

Firmware Version: V4020

中文



**USR**  
-IOT Experts-

Be Honest, Do Best!

Current Status

Local IP Config

Serial Port

Expand Function

Misc Config

Reboot

parameter

IP type: Static IP

Static IP: 169 . 254 . 203 . 199

Submask: 255 . 255 . 255 . 0

Gateway: 192 . 168 . 1 . 1

DNS Server: 8 . 8 . 8 . 8

Save

Cancel

Help

- IP type:** StaticIP or DHCP
- StaticIP:** Module's static ip
- Submask:** usually 255.255.255.0
- Gateway:** Usually router's ip address
- DNS IP:** DNS gateway or Router's IP

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved

website: [www.pusr.com](http://www.pusr.com)

Figure 5 - Webserver

Document Number  
APN0038

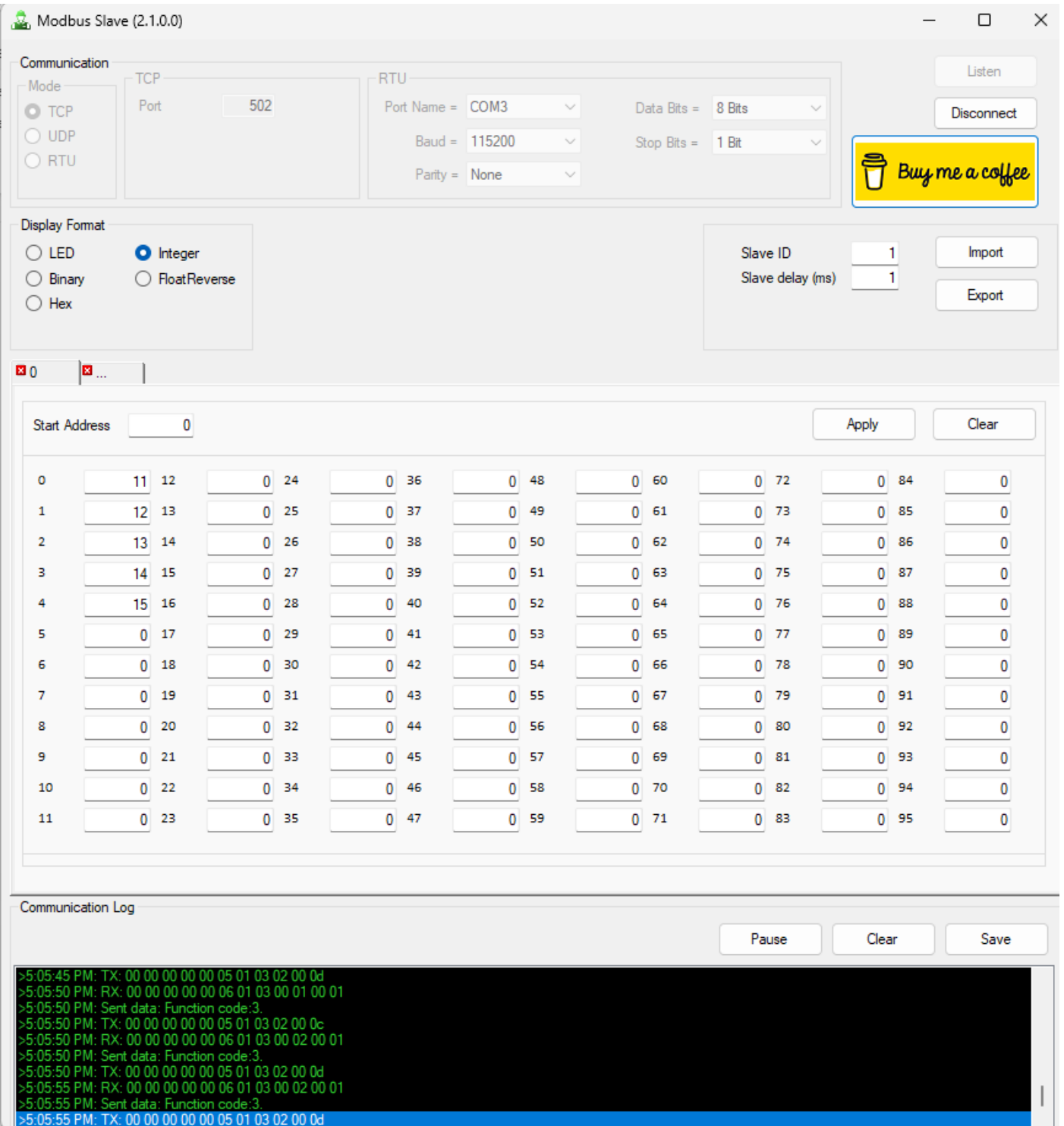
Revision  
1.0

Prepared By  
NGB

Approved By  
NB

Title  
Modbus TCP over Ethernet

Page  
6 of 8



**Modbus Slave (2.1.0.0)**

**Communication**

Mode: ☒ TCP ☐ UDP ☐ RTU

TCP Port: 502

RTU Port Name: COM3 Baud: 115200 Parity: None Data Bits: 8 Bits Stop Bits: 1 Bit

Buttons: Listen, Disconnect

Display Format: ☐ LED ☒ Integer ☐ Binary ☐ Hex ☐ FloatReverse

Slave ID: 1 Slave delay (ms): 1 Buttons: Import, Export

Start Address: 0 Buttons: Apply, Clear

|    |    |    |   |    |   |    |   |    |   |    |   |    |   |    |   |
|----|----|----|---|----|---|----|---|----|---|----|---|----|---|----|---|
| 0  | 11 | 12 | 0 | 24 | 0 | 36 | 0 | 48 | 0 | 60 | 0 | 72 | 0 | 84 | 0 |
| 1  | 12 | 13 | 0 | 25 | 0 | 37 | 0 | 49 | 0 | 61 | 0 | 73 | 0 | 85 | 0 |
| 2  | 13 | 14 | 0 | 26 | 0 | 38 | 0 | 50 | 0 | 62 | 0 | 74 | 0 | 86 | 0 |
| 3  | 14 | 15 | 0 | 27 | 0 | 39 | 0 | 51 | 0 | 63 | 0 | 75 | 0 | 87 | 0 |
| 4  | 15 | 16 | 0 | 28 | 0 | 40 | 0 | 52 | 0 | 64 | 0 | 76 | 0 | 88 | 0 |
| 5  | 0  | 17 | 0 | 29 | 0 | 41 | 0 | 53 | 0 | 65 | 0 | 77 | 0 | 89 | 0 |
| 6  | 0  | 18 | 0 | 30 | 0 | 42 | 0 | 54 | 0 | 66 | 0 | 78 | 0 | 90 | 0 |
| 7  | 0  | 19 | 0 | 31 | 0 | 43 | 0 | 55 | 0 | 67 | 0 | 79 | 0 | 91 | 0 |
| 8  | 0  | 20 | 0 | 32 | 0 | 44 | 0 | 56 | 0 | 68 | 0 | 80 | 0 | 92 | 0 |
| 9  | 0  | 21 | 0 | 33 | 0 | 45 | 0 | 57 | 0 | 69 | 0 | 81 | 0 | 93 | 0 |
| 10 | 0  | 22 | 0 | 34 | 0 | 46 | 0 | 58 | 0 | 70 | 0 | 82 | 0 | 94 | 0 |
| 11 | 0  | 23 | 0 | 35 | 0 | 47 | 0 | 59 | 0 | 71 | 0 | 83 | 0 | 95 | 0 |

**Communication Log**

Buttons: Pause, Clear, Save

```
>5:05:45 PM: TX: 00 00 00 00 00 05 01 03 02 00 0d
>5:05:50 PM: RX: 00 00 00 00 00 06 01 03 00 01 00 01
>5:05:50 PM: Sent data: Function code:3.
>5:05:50 PM: TX: 00 00 00 00 00 05 01 03 02 00 0c
>5:05:50 PM: RX: 00 00 00 00 00 06 01 03 00 02 00 01
>5:05:50 PM: Sent data: Function code:3.
>5:05:50 PM: TX: 00 00 00 00 00 05 01 03 02 00 0d
>5:05:55 PM: RX: 00 00 00 00 00 06 01 03 00 02 00 01
>5:05:55 PM: Sent data: Function code:3.
>5:05:55 PM: TX: 00 00 00 00 00 05 01 03 02 00 0d
```

Figure 6 - Modbus Slave Settings

Document Number  
APN0038

Revision  
1.0

Prepared By  
NGB

Approved By  
NB

Title  
Modbus TCP over Ethernet

Page  
7 of 8

### Serial 1 (Modbus TCP) ✓

Name

Modbus TCP

Interval

1

Type

RS485

Termination Resistor

☒ Enabled

Mode

Modbus

Baud Rate

57600

Settings

8N1

Powered by Output 1

☐ Enabled

#### Capture

Start String

Start String

Idle Time Before Start

0

Milliseconds

End String

End String

Request String

Request String

Max Time

5

Seconds

Max Chars

500

Capture Alert

#### Modbus RTU

Slave Timeout

0.2

Seconds

Delay Between Reads

15

Milliseconds

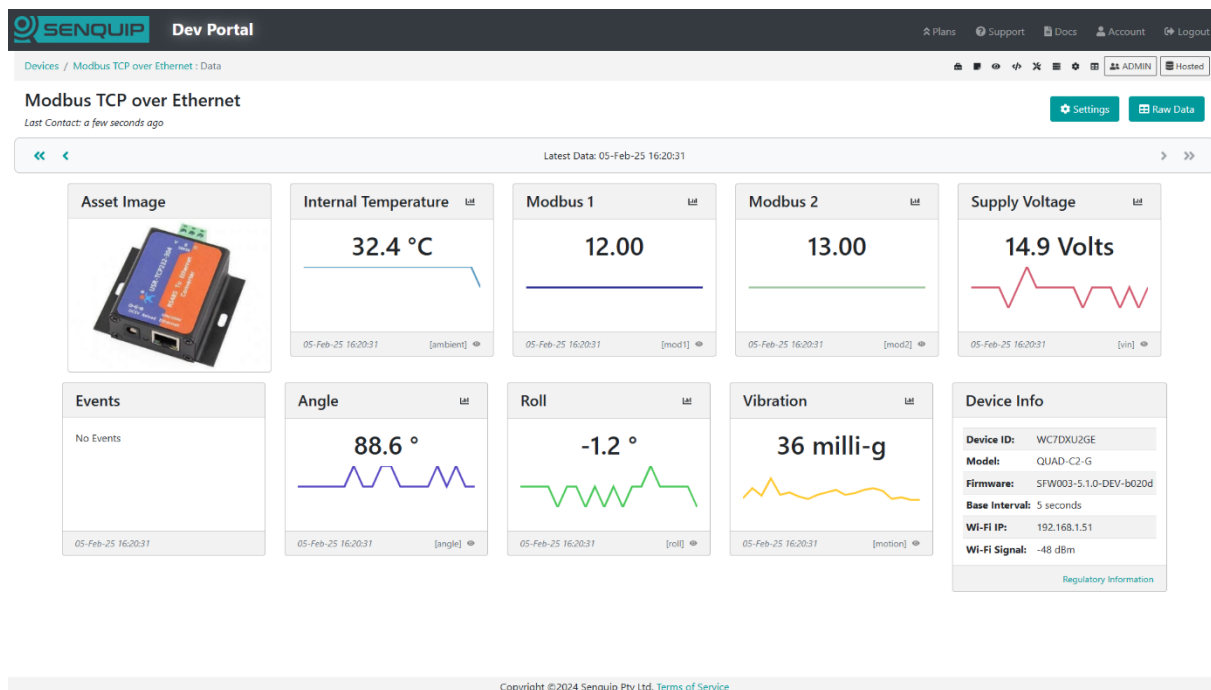
Figure 7 - Senquip QUAD Serial Port Settings

| Modbus                                                                                            |                         |               |                                    |                  |
|---------------------------------------------------------------------------------------------------|-------------------------|---------------|------------------------------------|------------------|
| Configure up to 50 reads from Modbus RTU slaves in the table below. Select a cell to edit values. |                         |               |                                    |                  |
| ID                                                                                                | Name                    | Slave Address | Function                           | Register Address |
| 1                                                                                                 | <span>✗</span> Modbus 1 | 1             | 3: Read Unsigned Holding (16-bits) | 1                |
| 2                                                                                                 | <span>✗</span> Modbus 2 | 1             | 3: Read Unsigned Holding (16-bits) | 2                |

Figure 8 - Senquip QUAD Modbus Requests

## 4. Results

Modbus request packets can be seen arriving on the Modbus Slave Software. Responses are seen to be sent. Modbus reads 1 and 2 are shown on the Senquip Portal.



## 5. Conclusions

This application note demonstrates how a Senquip QUAD can communicate with a remote Modbus TCP sensor using an RS485-to-Ethernet converter. By configuring the PUSR USR-TCP232-304 as a TCP client, assigning a static IP address, and setting up the Modbus Slave software as a TCP server, seamless data exchange was established.

Successful Modbus requests were observed in the Modbus Slave software's communication log, confirming proper transmission and response of data packets. The Senquip Portal displayed the requested register values, verifying correct configuration and operation.

This approach provides a cost-effective solution for integrating RS485-based Modbus RTU devices into Ethernet-based Modbus TCP networks, enabling remote monitoring and control in industrial applications.