

Modbus Master

1. Introduction

1.1 Overview

This document contains information regarding the configuration and use of the Modbus Master application and using it to poll or scan external Modbus TCP or RTU devices.

1.2 Compatibility

This application note applies to :

Models Shown: 641M series.

Firmware Version: V1 .2.0 (e958360) or newer

Other Compatible Models: Modbus Gateway application

1.3 Version

Updates between document versions are cumulative. Therefore, the latest document will include all the content of previous versions.

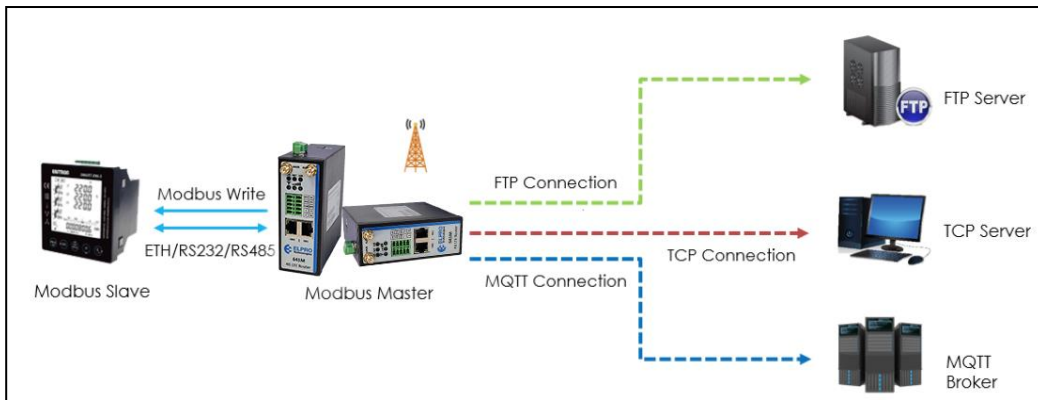
Release Date	Doc. Version	Firmware Version	Change Description
2021-12-24	V1.0	V1 .2.0 (e958360)	First released

1.4 Corrections

Appreciate for corrections or rectifications to this application note, and if any request for new application notes please email to: support@elpro.com.au

2. Topology

- 641M Router setup as a Modbus Master and connected to a Modbus Slave via Ethernet, RS232 or RS485 interface.
- 641M router polls the Modbus data from Modbus slave and can send to a remote management center via TCP, FTP or MQTT protocol.
- 641M router can write register value or coils to the Modbus Slave.

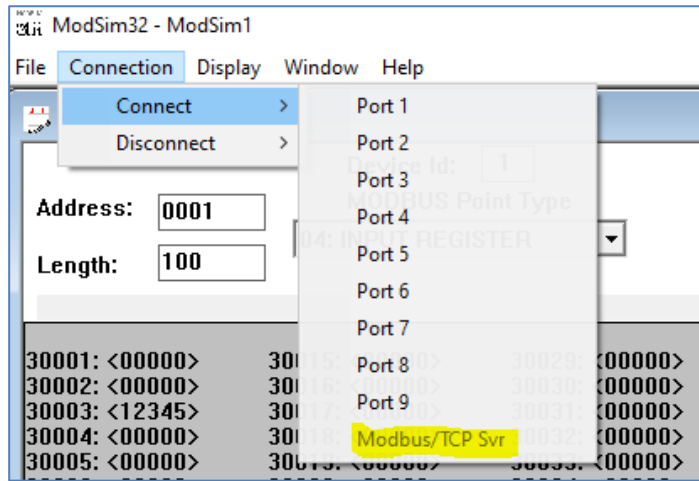


Note: For this Application Note, we will set the Connection Type as TCP as an example, which means that 641M (Modbus Master will connect to the Modbus Slave and read the value via Ethernet port. However, it can also do the same using the RS232 or RS485 serial ports.

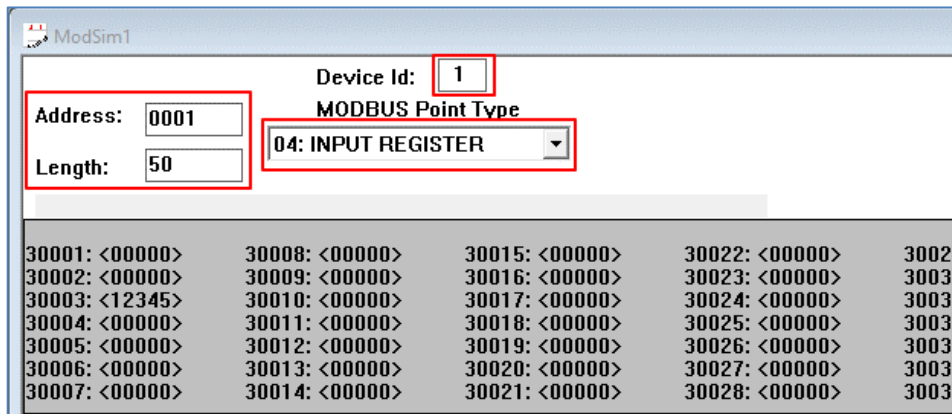
3. Configuration

3.1 Configuration of a Modbus Slave

- Here we are using some Modbus Slave software (Modsim32) to simulate the end Modbus TCP Slave device, Configure Modsim32 connection as “Modbus TCP /Svr” and it will automatically use the Computers Network IP address and Port 502. (You may need to Allow the Modbu TCP on port 502 in you computers firewall)

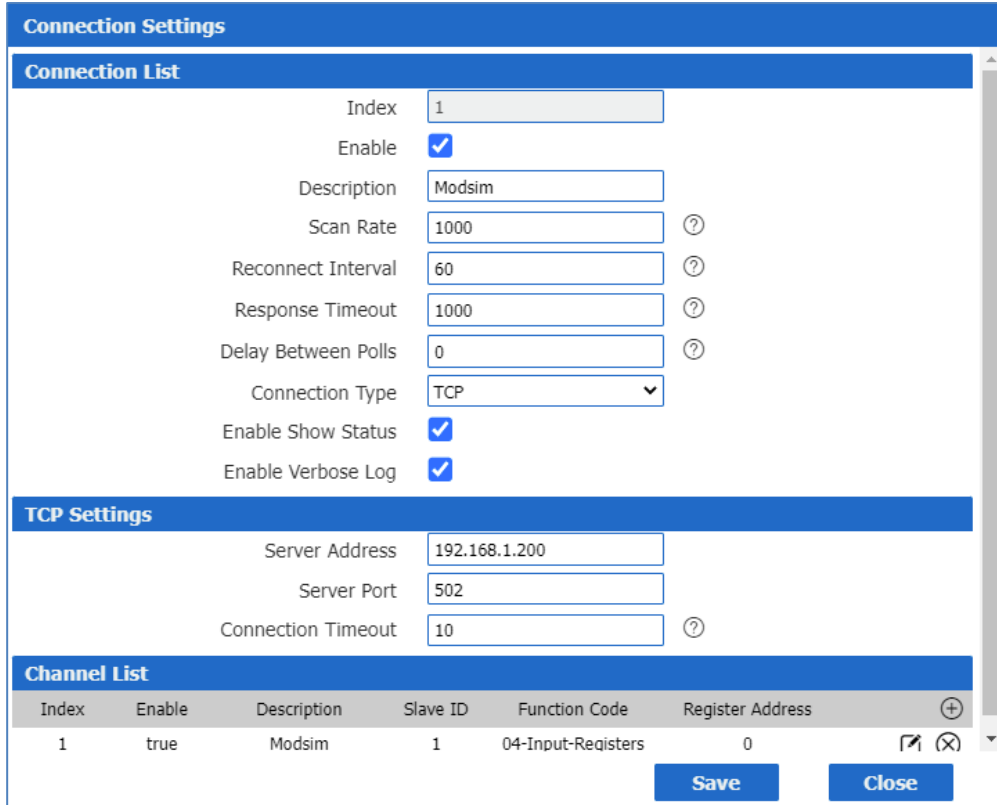


- Configure Modsim32 with the following parameters, Slave ID : 1, Function Code : 04 Input Register, a Starting address (0001) and a Length (number of consecutive registers to read) as per below setting:

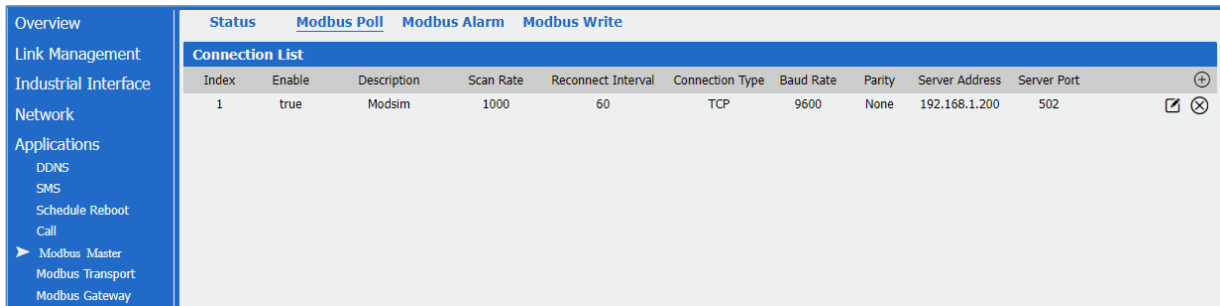


3.2 Configuration of Modbus Master Application

- Go to **Application / Modbus Master / Modbus Poll**, add a “Connection List” and specify the “Connection Type” as “TCP”, specify the “TCP Settings” to connect to Modbus Slave as per below.



- Enable “Channel List”, and specify a Name, Slave ID as “1”, Function Code as “04 Input Registers”, Register Address to “0”, then select Multiple Registers and add a Quantity. The rest can stay as default.
- Click **Save> Save> Apply**. Then Save & Apply on the Main window.

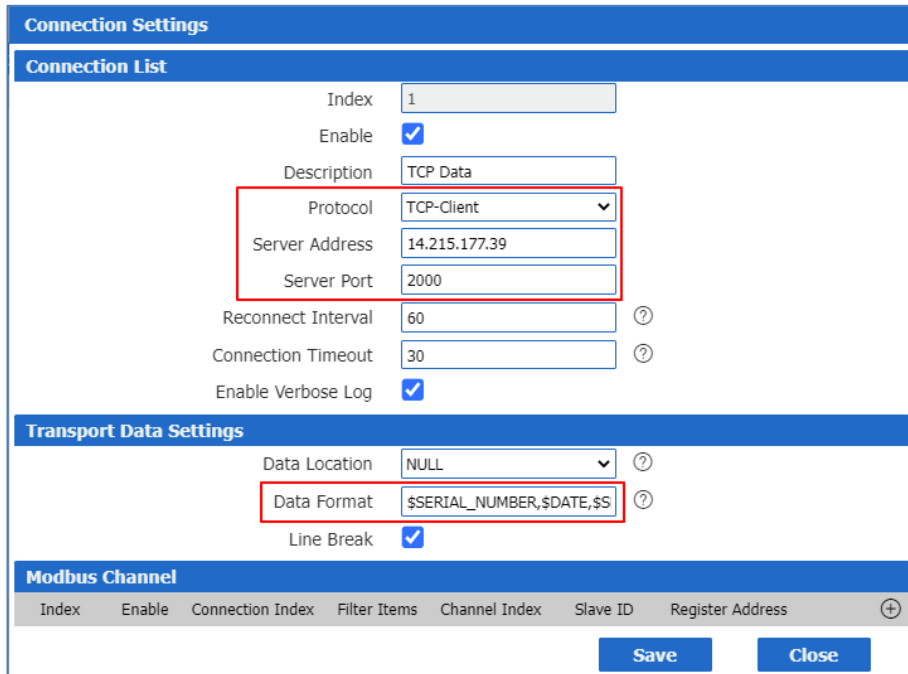


- Go to **Application>Modbus Master>Status**, then we can check the router had read the value from Modbus Slave successfully.

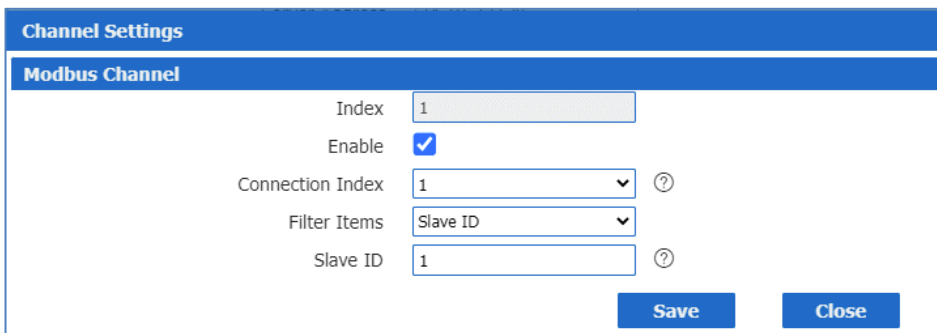
Channel Status									
Index	Description	Connection Index	Type	Slave ID	Register Address	Function Code	Status	Value	
1	Modsim	1	TCP	1	0	4	Read successfully	0, 0, 12345, 0, 0,...	

3.3 Configuration of Modbus Transport

1. Go to **Application>Modbus Transport> Modbus Transport**, enable “Connection List”, and specify the remote TCP server IP address and port to send the data to. The Data Format should be defined accordingly or set it as default “\$SERIAL_NUMBER,\$DATE,\$SLAVE_ID,\$REGISTER_ADDER,\$VALUE”.

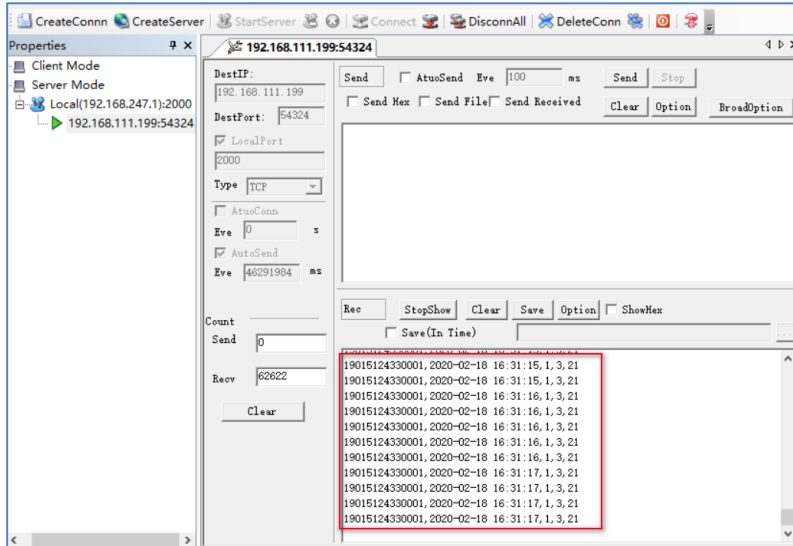


2. Enable and configure a “Modbus Channel”, this is where the Modbus Master will send the “Transport Data”.



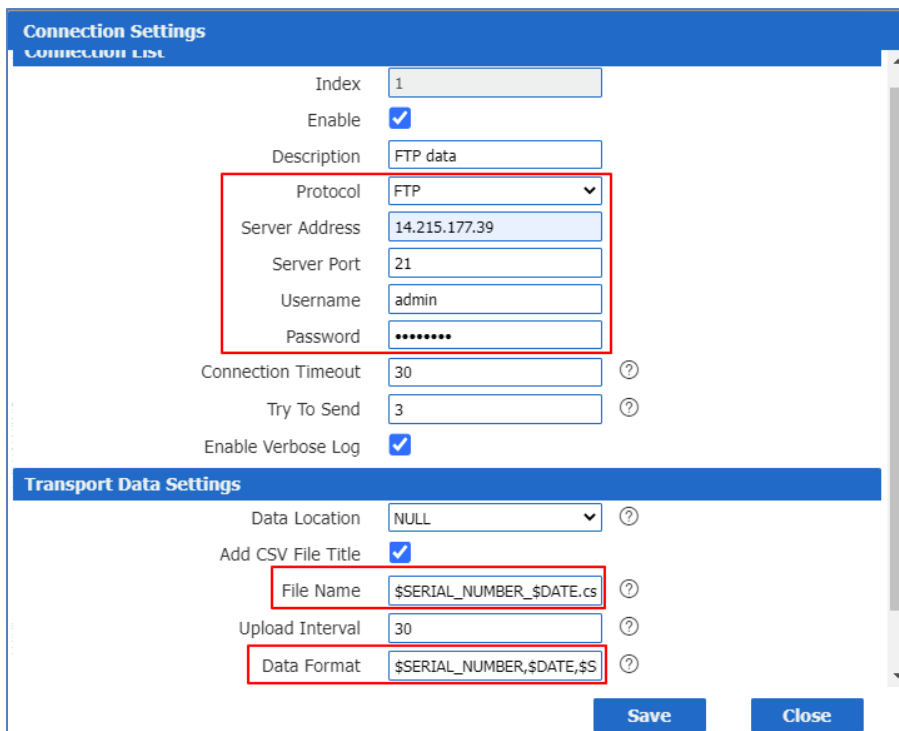
3. Click Save>Save>Apply.
4. Go to **Application>Modbus Transport>Status**, will show the connection status and uptime for the connection.

Status					
Modbus Transport					
X.509 Certificate					
Connection Status					
Index	Enable	Description	Protocol	Status	Uptime
1	true	TCP Data	TCP Client	Connected	00:01:38

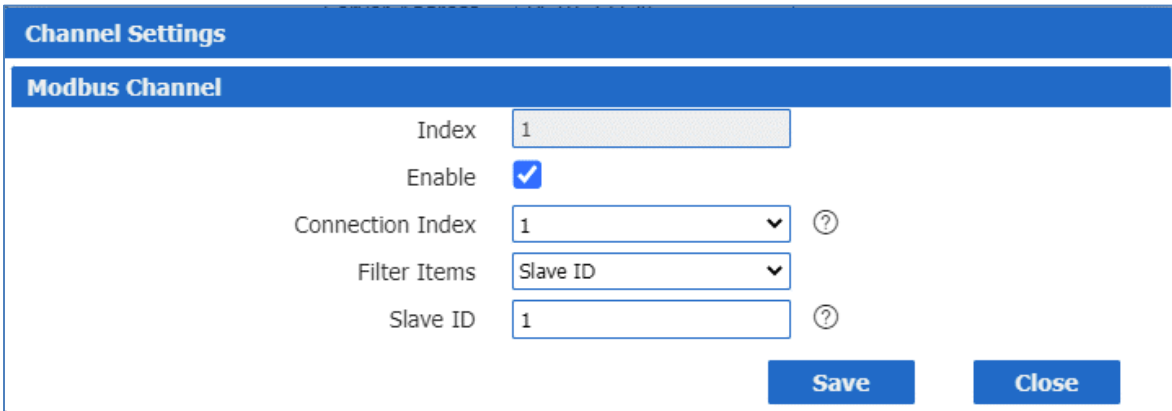


4. Transport via FTP

1. Please refer to the Configuration of the Modbus Slave and Modbus Master Polling in section 3.1 and 3.2 above.
2. Navigate to **Application>Modbus Transport> Modbus Transport**, enable “Connection List”, and specify FTP, Server IP address, port , username and password of the remote FTP server that the data will be sent to. The File Name and Data Format could be defined accordingly or left at the default in the “Transport data Settings” below.



5. Enable and configure the “Modbus Channel”, this is where the Modbus Master will send the “Transport Data”.



Channel Settings

Modbus Channel

Index: 1

Enable:

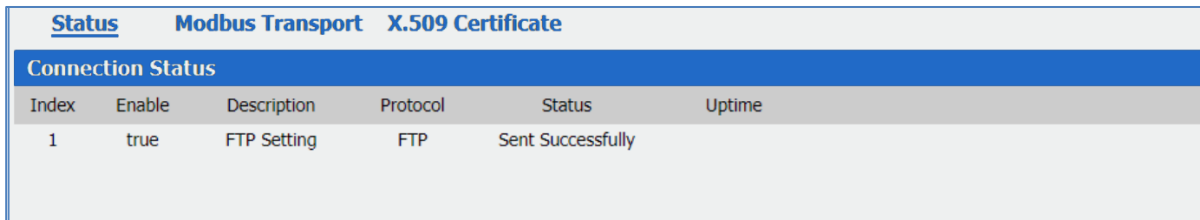
Connection Index: 1

Filter Items: Slave ID

Slave ID: 1

Buttons: Save, Close

3. Click Save>Save>Apply.
4. Go to **Application>Modbus Transport>Status**, this will show if the connection was successful sent or Failed.

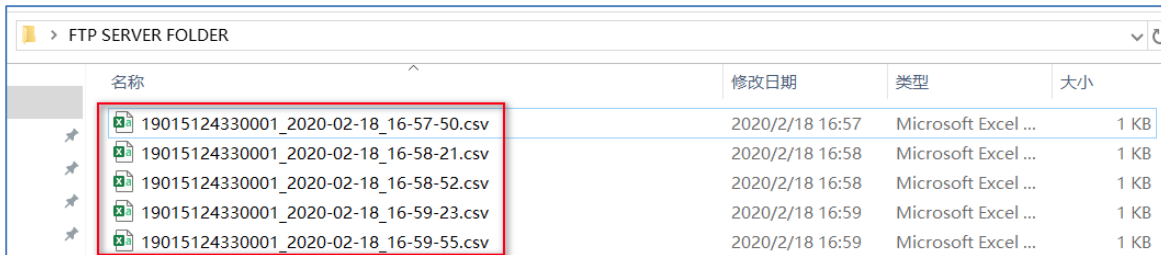


Status **Modbus Transport** **X.509 Certificate**

Connection Status

Index	Enable	Description	Protocol	Status	Uptime
1	true	FTP Setting	FTP	Sent Successfully	

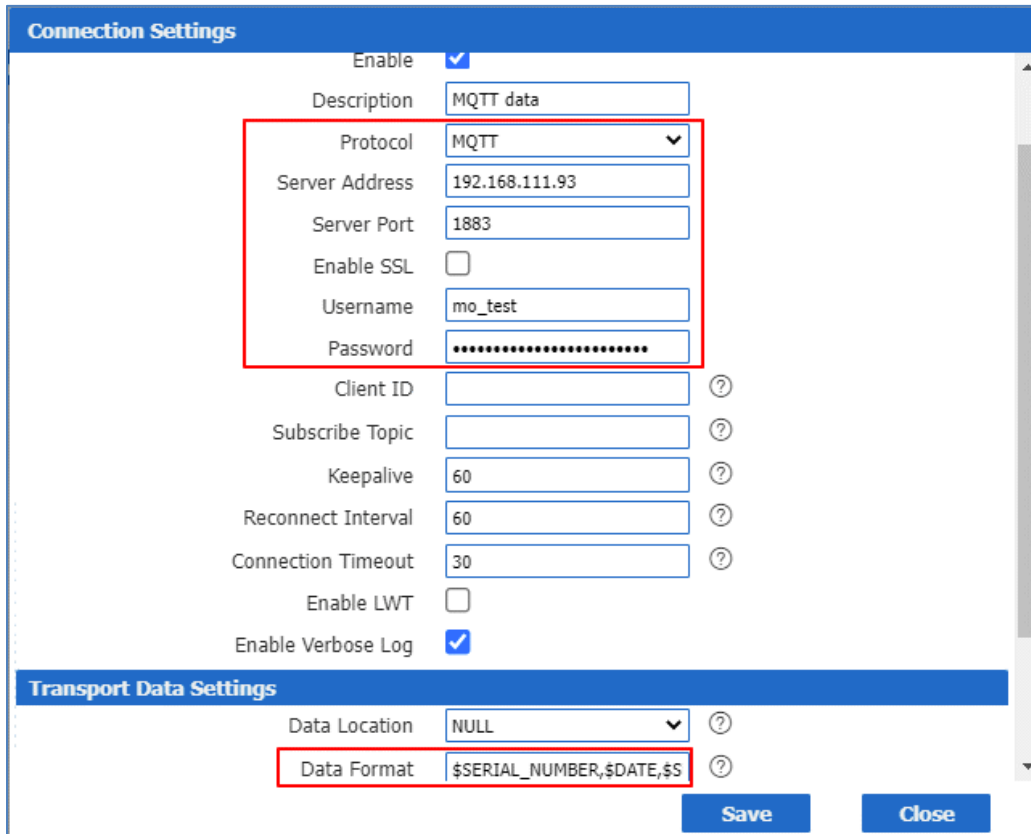
5. Remote FTP Server should have received the CSV file/s.



名称	修改日期	类型	大小
19015124330001_2020-02-18_16-57-50.csv	2020/2/18 16:57	Microsoft Excel ...	1 KB
19015124330001_2020-02-18_16-58-21.csv	2020/2/18 16:58	Microsoft Excel ...	1 KB
19015124330001_2020-02-18_16-58-52.csv	2020/2/18 16:58	Microsoft Excel ...	1 KB
19015124330001_2020-02-18_16-59-23.csv	2020/2/18 16:59	Microsoft Excel ...	1 KB
19015124330001_2020-02-18_16-59-55.csv	2020/2/18 16:59	Microsoft Excel ...	1 KB

5. Transport via MQTT

1. Please refer to the 3.1 Configuration on Modbus Slave and 3.2 Configuration on Modbus Poll to finish and setting.
2. Go to **Application>Modbus Transport> Modbus Transport**, enable “Connection List”, and specify the MQTT Broker IP Address, port, username and password to wish to publish the Topic to. The Data Format could be defined accordingly or left as the default in the “Transport Data Settings”.



Connection Settings

Enable

Description

Protocol

Server Address

Server Port

Enable SSL

Username

Password

Client ID ?

Subscribe Topic ?

Keepalive ?

Reconnect Interval ?

Connection Timeout ?

Enable LWT

Enable Verbose Log

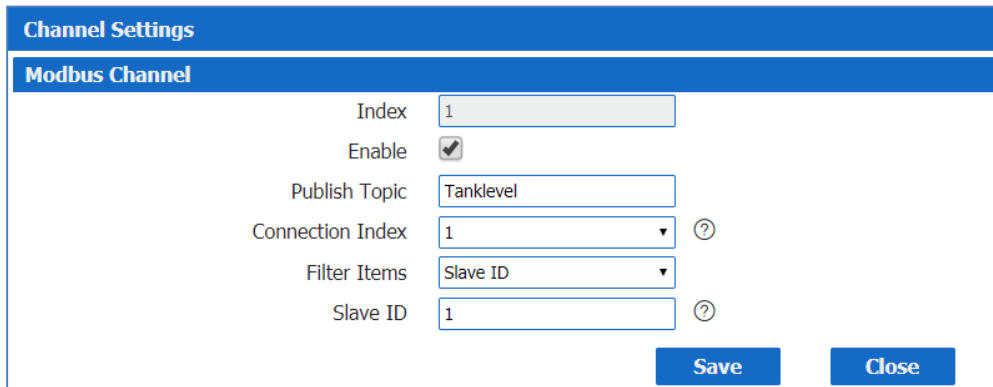
Transport Data Settings

Data Location ?

Data Format ?

Save **Close**

3. Enable and configure the “Modbus Channel”, Define the “Topic” to publish to the MQTT Broker with Modbus data



Channel Settings

Modbus Channel

Index

Enable

Publish Topic

Connection Index ?

Filter Items ?

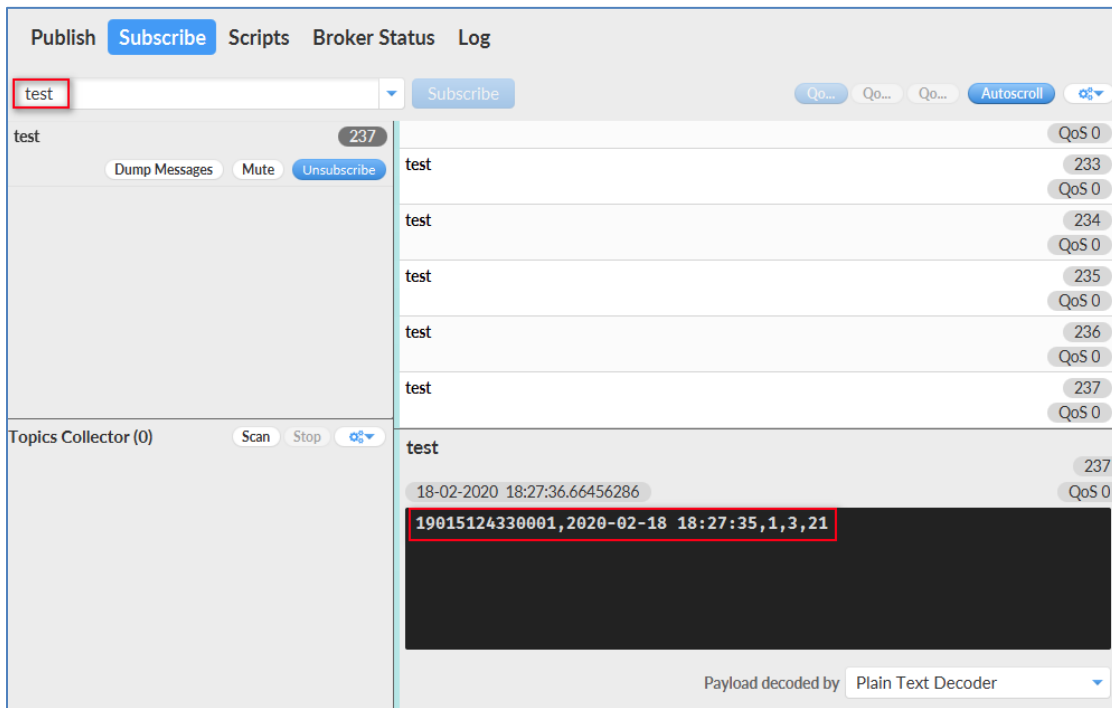
Slave ID ?

Save **Close**

4. Click Save>Save>Apply.
5. Go to **Application>Modbus Transport>Status**, 641M Modbus Master has connected to the MQTT Broker successfully.

Status Modbus Transport X.509 Certificate					
Connection Status					
Index	Enable	Description	Protocol	Status	Uptime
1	true	MQTT data	MQTT	Connecting	

6. Run the MQTT Client (MQTT Subscriber), to subscribe the topic just published to MQTT broker with Modbus data. Then we can get the Modbus data successfully.



The screenshot shows the MQTT Client interface with the following components:

- Navigation:** Publish, **Subscribe**, Scripts, Broker Status, Log
- Subscription:** A text input field containing 'test' is highlighted with a red box. A 'Subscribe' button is next to it. On the right, there are 'QoS' dropdowns and an 'Autoscroll' checkbox.
- Message List:** A list of received messages for the 'test' topic. The messages are:
 - test (QoS 0)
 - test (233, QoS 0)
 - test (234, QoS 0)
 - test (235, QoS 0)
 - test (236, QoS 0)
 - test (237, QoS 0)
- Message Detail:** The selected message shows:
 - test (237, QoS 0)
 - Timestamp: 18-02-2020 18:27:36.66456286
 - Message content: **19015124330001,2020-02-18 18:27:35,1,3,21** (highlighted with a red box)
 - Decoder: Payload decoded by Plain Text Decoder
- Topics Collector:** A section at the bottom left with 'Topics Collector (0)', 'Scan', 'Stop', and a refresh icon.