NavigateWorx

NR500 Series
Industrial Cellular VPN Router

Application Note 064

Modbus LwM2M

Version: V1.0.0
Date: Aug 2022
Status: Confidential

Directory

1. Introduction

1.1 Overview

This document contains information regarding the configuration and use of Modbus LwM2M.

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product, and of the requirements for their specific application.

1.2 Compatibility

This application note applies to:
Models Shown: NR500 series.
Firmware Version: V1.1.7(3b5122d) or newer
Other Compatible Models: None

1.3 Version

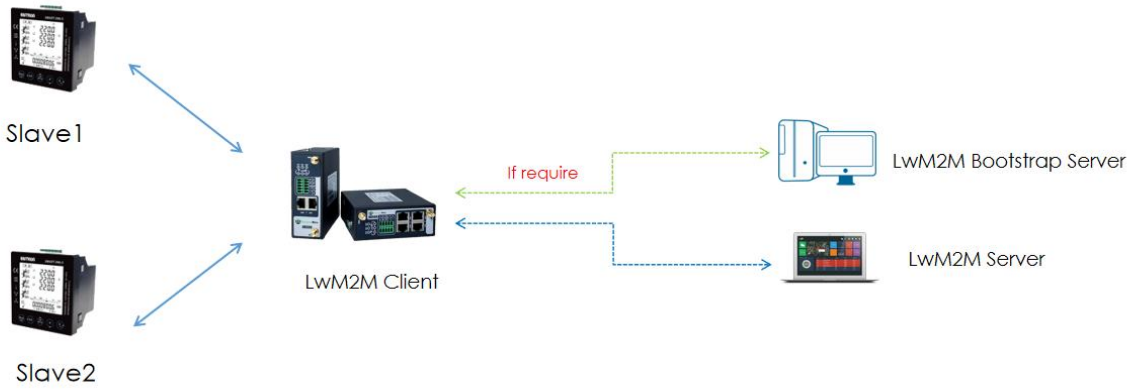
Updates between document versions are cumulative. Therefore, the latest document will include all the content of previous versions.

Release Date	Doc. Version	Firmware Version	Change Description
2022/08/24	V1.0.0	V1.1.7(3b5122d)	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@navigateworx.com

2. Topology



1. NR500 Router act as Modbus Master and LwM2M Client, and connect to Modbus Slave via Ethernet, RS232 or RS485 interface.
2. NR500 Router poll the modbus data from modbus slave and send to the LwM2M Bootstrap Server or LwM2M Server via LwM2M protocol.
3. LwM2M Bootstrap Server or LwM2M Server can write the register value to Modbus Slave.

3. Configuration

3.1 LwM2M Server Configuration

1. Setup LwM2M Server by Eclipse Leshan project, which requires JAVA running environment. Please Prepare JDK environment before the testing. Our testing environment as below:

```
E:\leshan>java -version
openjdk version "1.8.0_302"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_302-b08)
OpenJDK 64-Bit Server VM (Temurin)(build 25.302-b08, mixed mode)

E:\leshan>
```

2. Add a "modules" folder, and create **.XML** file to create instances and object resources.

The configuration of **31024.xml** like below:

```
<?xml version="1.0" encoding="UTF-8"?>

<LWM2M  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://openmobilealliance.org/tech/profiles/LWM2M-v1_1.xsd">
    <Object ObjectType="MODefinition">
        <Name>Test Object</Name>
        <Description1>Tommy Test object.</Description1>
        <ObjectID>31024</ObjectID>
        <ObjectURN>urn:oma:lwm2m:x:31024</ObjectURN>
        <LWM2MVersion>1.1</LWM2MVersion>
        <ObjectVersion>1.0</ObjectVersion>
        <MultipleInstances>Multiple</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Resources>
            <Item ID="1">
                <Name>boolean 1</Name>
                <Operations>RW</Operations>
                <MultipleInstances>Single</MultipleInstances>
                <Mandatory>Optional</Mandatory>
                <Type>Boolean</Type>
                <RangeEnumeration></RangeEnumeration>
```

```
        <Units></Units>
        <Description>boolean value</Description>
    </Item>
    <Item ID="2">
        <Name>boolean 2</Name>
        <Operations>RW</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Boolean</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>boolean value</Description>
    </Item>
    <Item ID="3">
        <Name>int16</Name>
        <Operations>RW</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Integer</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>int 16 value</Description>
    </Item>
    <Item ID="4">
        <Name>uint16</Name>
        <Operations>RW</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Unsigned Integer</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>uint 16 value</Description>
    </Item>
    <Item ID="5">
        <Name>int32</Name>
        <Operations>RW</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Integer</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>int 32 value</Description>
    </Item>
    <Item ID="6">
```

```
            <Name>float</Name>
            <Operations>RW</Operations>
            <MultipleInstances>Single</MultipleInstances>
            <Mandatory>Optional</Mandatory>
            <Type>Float</Type>
            <RangeEnumeration></RangeEnumeration>
            <Units></Units>
            <Description>float value</Description>
        </Item>
        <Item ID="7">
            <Name>double</Name>
            <Operations>RW</Operations>
            <MultipleInstances>Single</MultipleInstances>
            <Mandatory>Optional</Mandatory>
            <Type>Float</Type>
            <RangeEnumeration></RangeEnumeration>
            <Units></Units>
            <Description>double value</Description>
        </Item>
    </Resources>
    <Description2></Description2>
  </Object>
</LWM2M>
```

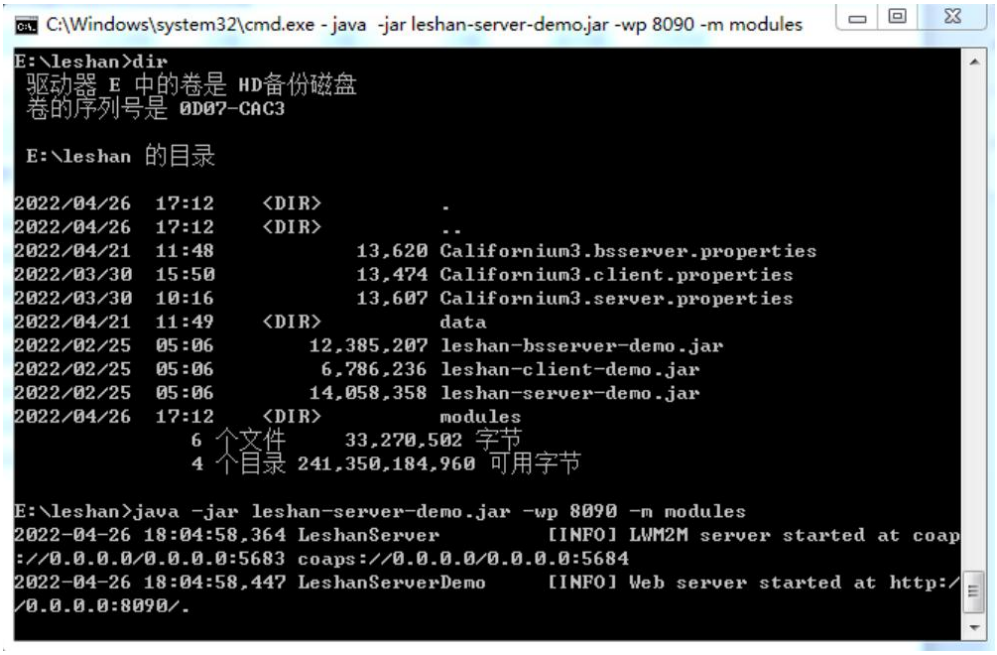
3. Install LwM2M Server or LwM2M Bootstrap Server, here we test with LwM2M Server.

4. Run leshan-server-demo.jar file with port 5683(CoAP) or port 5684(CoAPS) in CMD, the command as below:

java -jar leshan-server-demo.jar -wp 8090 -m modules

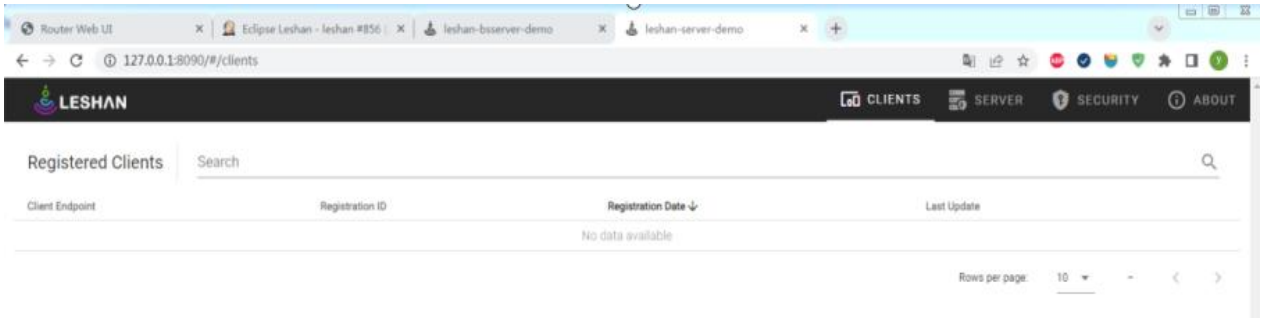
***Note**: a)*Run the server demo with webserver port 8090, because the default port 8080 will conflict when the Bootstrap server run.

*b)*If Run LwM2M Bootstrap Server with port 6683(CoAP) or port 6684(CoAPS) in CMD, the command: java -jar leshan-bsserver-demo.jar -lp 6683 -slp 6684

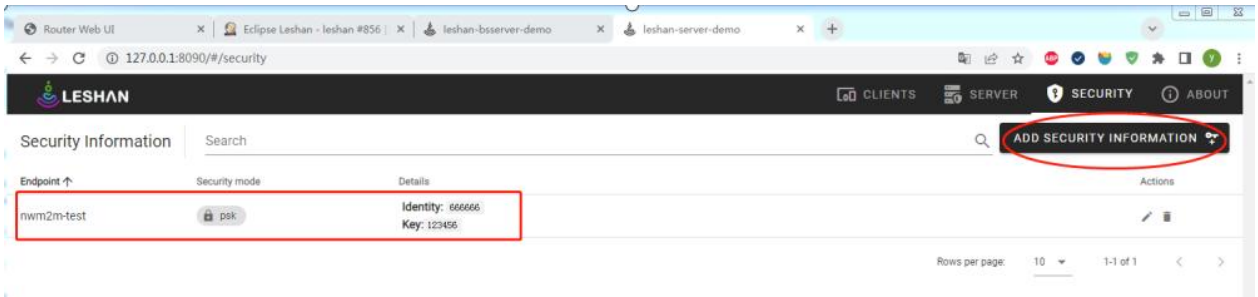


Running successfully.

5. Then access the server via http://127.0.0.1:8090. (LwM2M Bootstrap Server http://127.0.0.1:8080.)

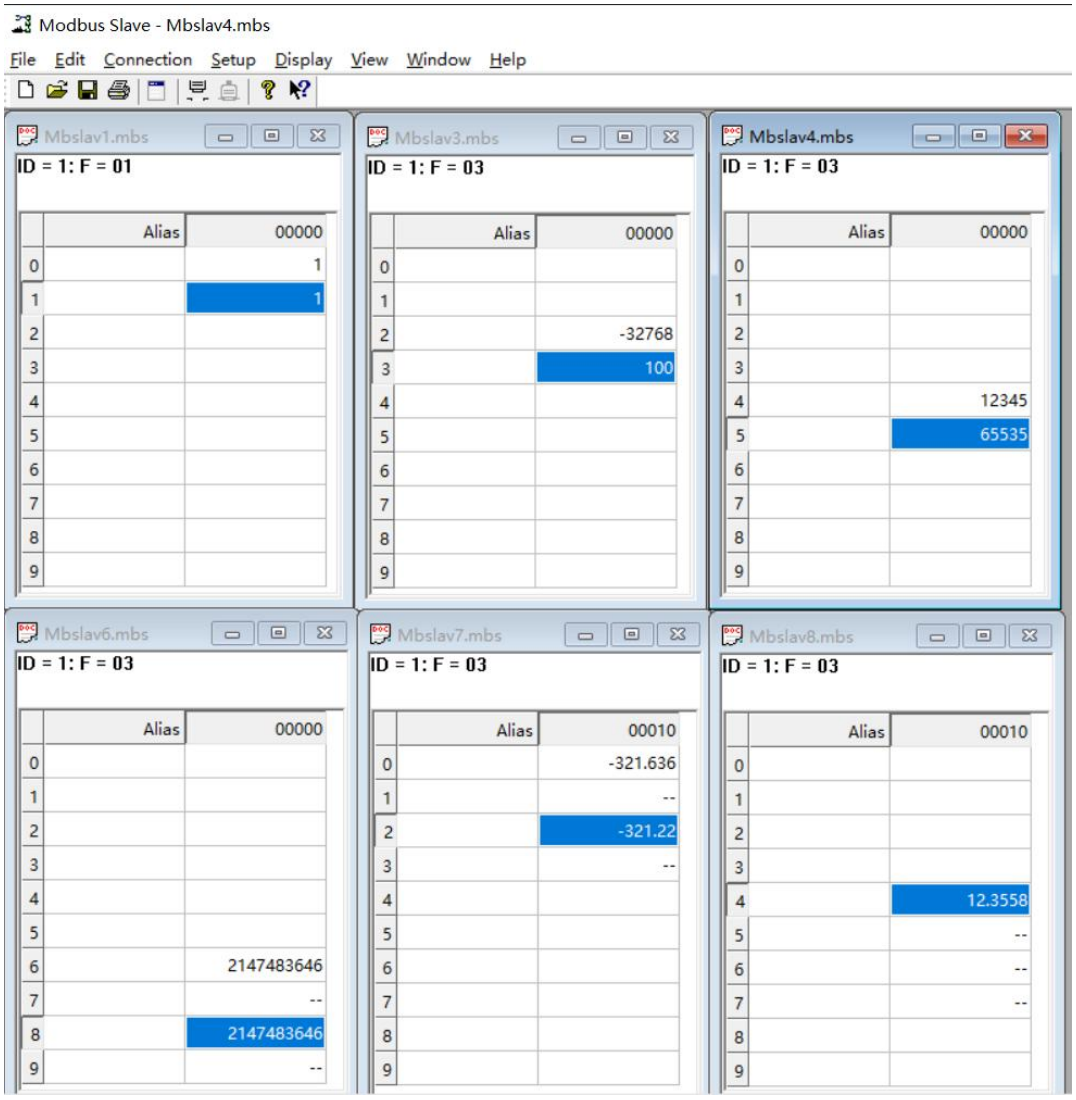


6. Add the Security information on LwM2M Server, please make sure that match LwM2M Client configuration. Add Endpoint Name, select Security mode, and add the PSK Identity and PSK Key.



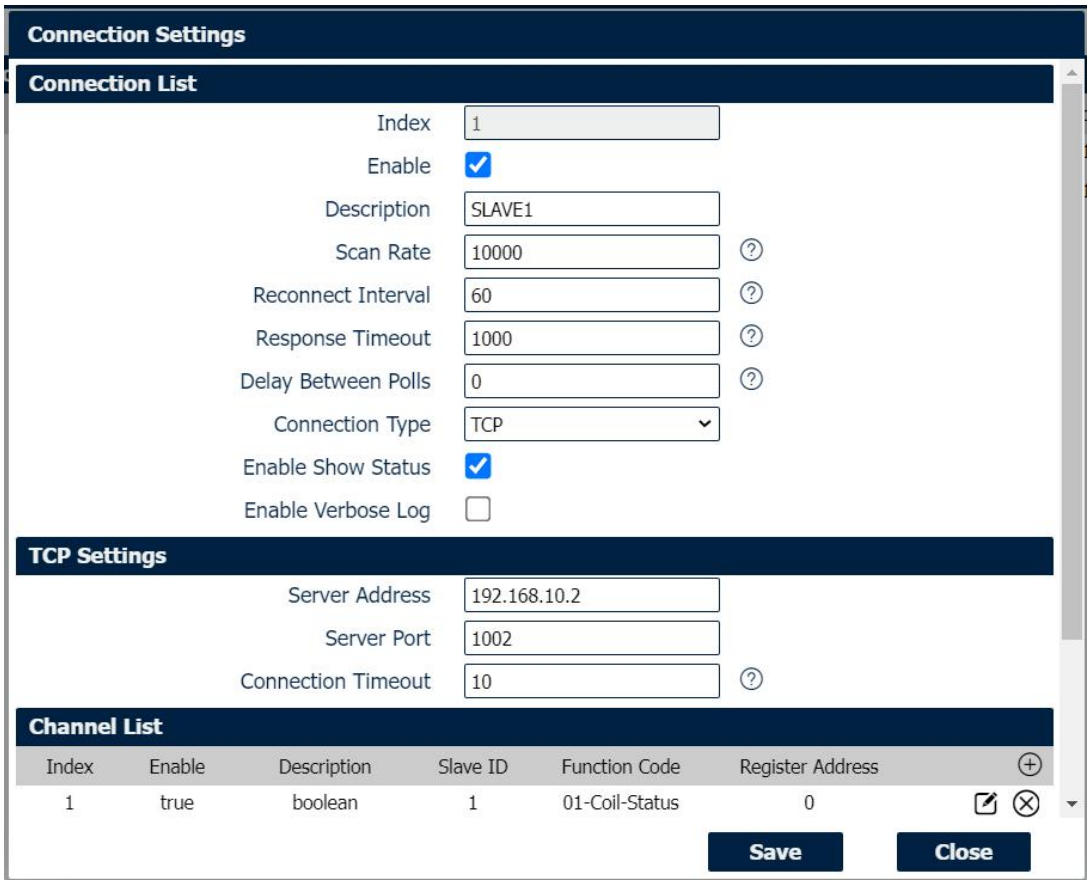
3.2 Configuration on Modbus Slave

1. Here we use "Modbus Slave" software to simulate the end device (Modbus Slave device), and the **TCP Port: 1002, Slave ID: 1, like below setting:**



3.3 Configuration on Modbus Poll

1. Go to **Application>Modbus Master>Modbus Poll**, add a "Connection List" and specify the "Connection Type" as **"TCP"**, specify the "**TCP Setting**" to connect to Modbus Slave, like below:

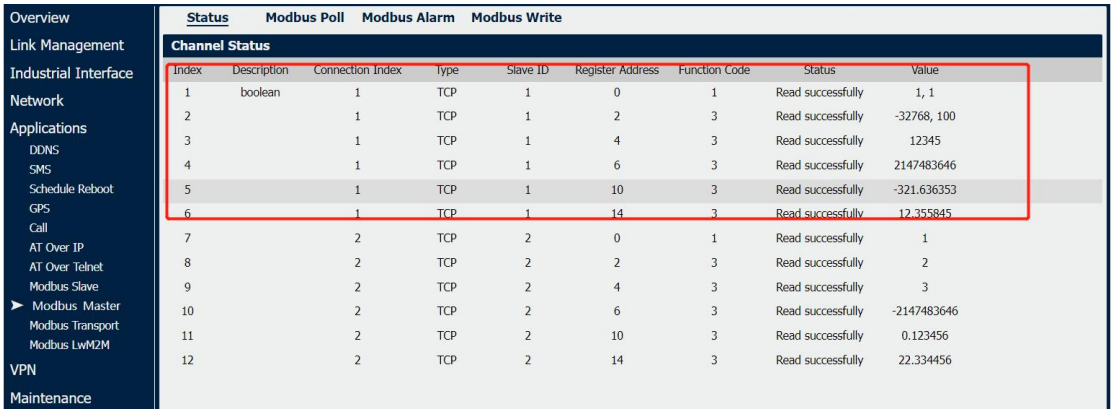


2. Click Save>Apply.

3. Enable "Channel List", and specify the Slave ID as " **1** ", Function Code according slave1, Register Address like below, then it will poll the value from register specially address of Modbus Slave:

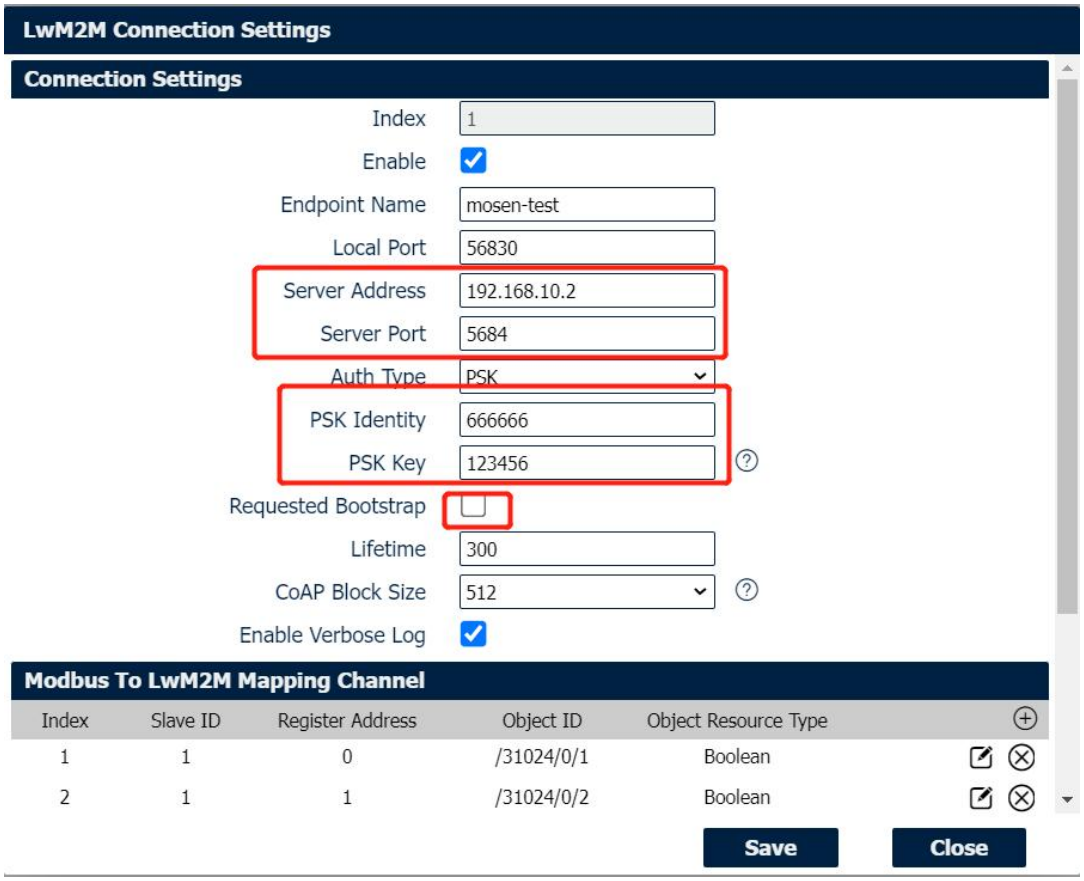
4. Click Save>Save>Apply. *(Note: This is a secondary list, it needs to double click save)*

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



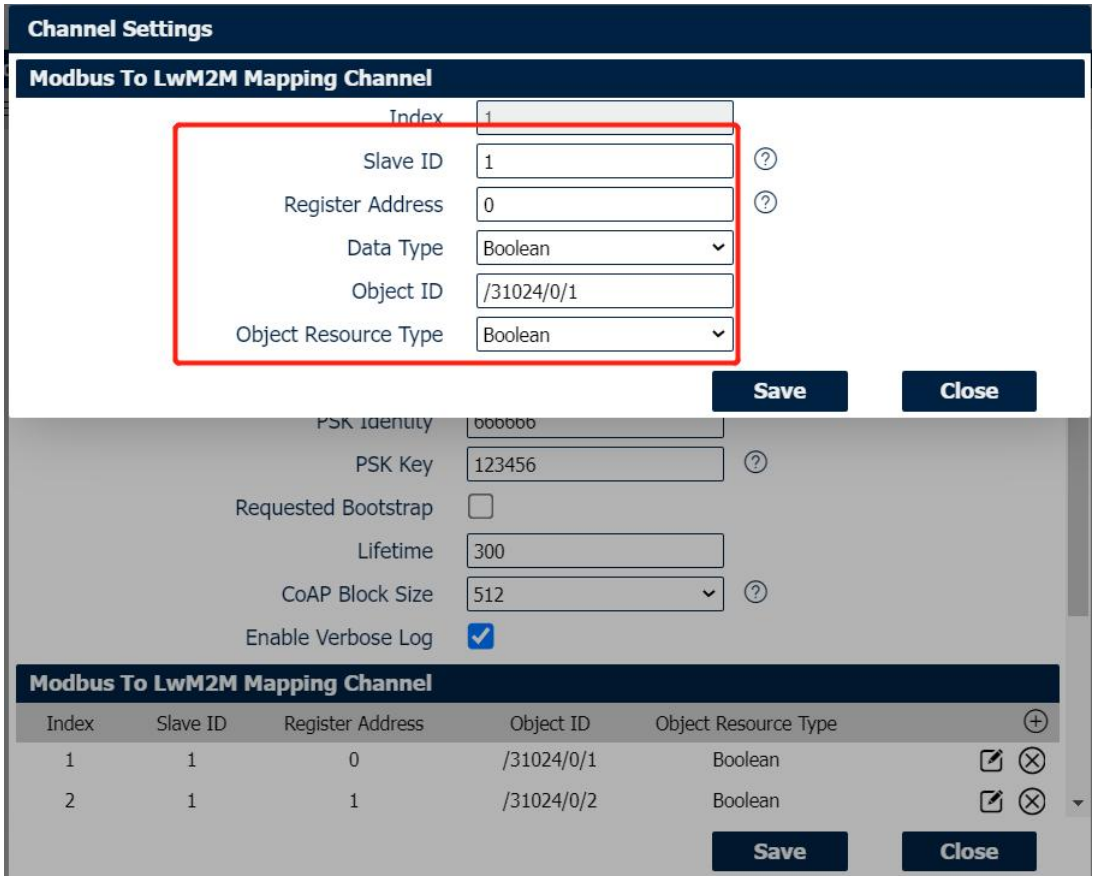
3.4 Configuration on Modbus LwM2M

1. Go to **Application>Modbus LwM2M>Modbus to LwM2M**, enable "Connection List", and specify Server address and Server port and Endpoint Name, Endpoint Name and PSK Identity and PSK key should be match the settings on LwM2M Server.



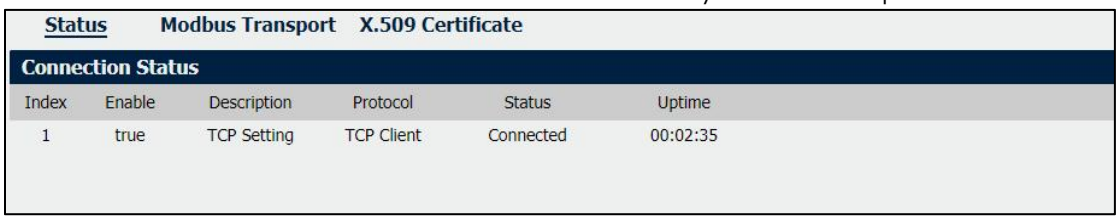
Note: If connect LwM2M Bootstrap Server, please enable "Requested Bootstrap", set port to 6683(CoAP) or port 6684(CoAPS).

2. Enable "Modbus To LwM2M Mapping Channel", Modbus Master will select the value send to the remote LwM2M server from Modbus Slave.



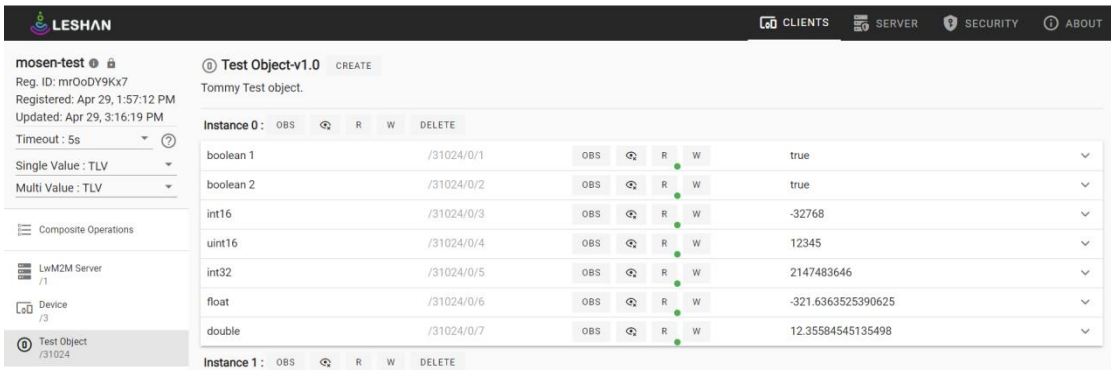
3. Click Save>Save>Apply. (Note: This is a secondary list, it needs to double click save)

4. Go to **Application>Modbus LwM2M>Status**, NR500(Modbus Master) had connected to the remote LwM2M server successfully via LwM2M protocol.

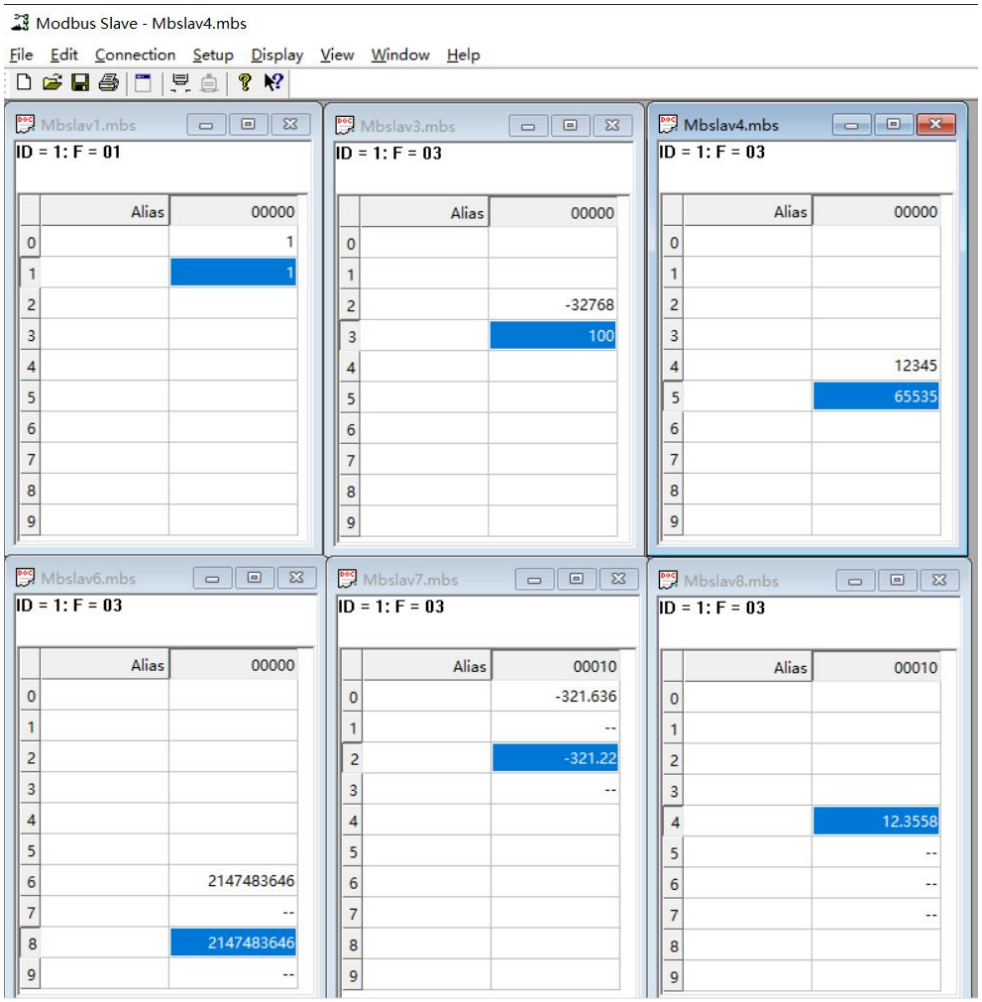


4. Testing

1. Remote LwM2M Server received the data successfully.



2. Click the "R" to read the data, Click the "W" to write data to Modbus slave. When 'OBS' is enable, value will be update in the real time.



Test successfully.