Getting Started with ThingsBoard platform

ThingsBoard is an open-source IoT platform that allows users to collect, process, and visualize data from IoT devices. It supports various IoT protocols like MQTT, CoAP, and HTTP, making it flexible for different applications. Users can create dashboards, set alerts, and manage devices through a web-based interface. It's great for developers and businesses looking to monitor and control IoT environments.

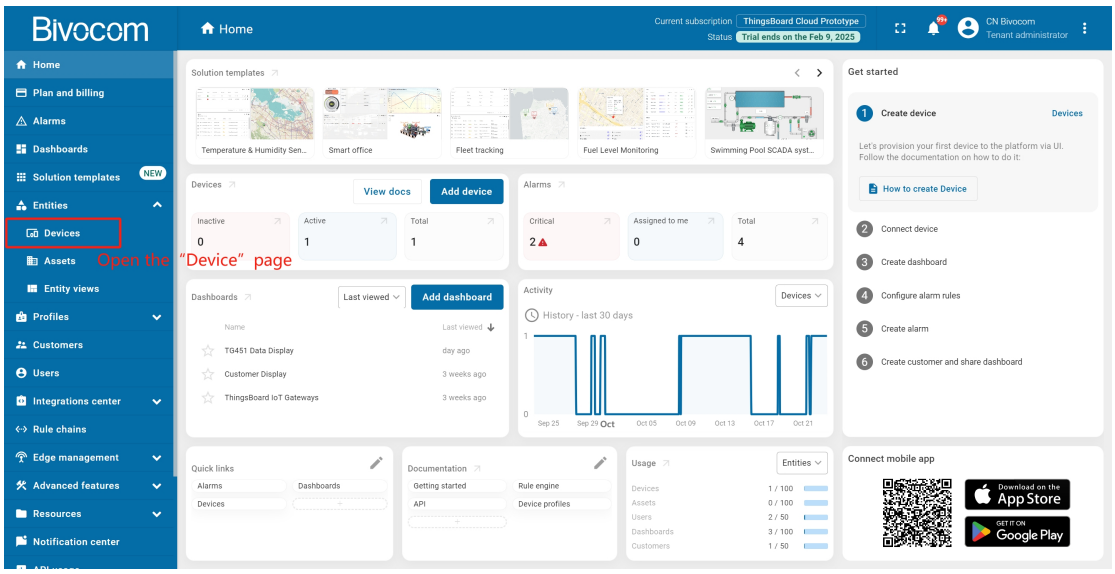
Bivocom IoT Router&Gateway can integrate with ThingsBoard IoT platform, which allows user manage device and monitor data more easily. An IoT gateway acts as a bridge between IoT devices and the cloud platform, enabling smooth data transmission and intelligent processing.

This document will demonstrate how to integrate Bivocom Router&Gateway with ThingsBoard platform. It can help you facilitate easy scaling of IoT systems by managing additional devices without overwhelming the main platform.

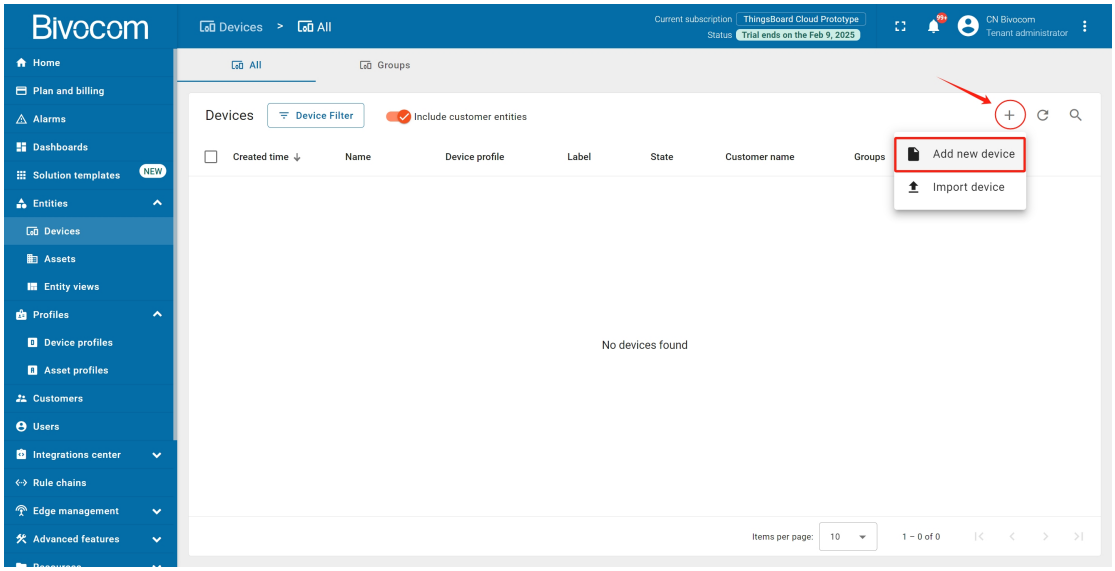
In this case, I'll use gateway TG451 to connect ThingsBoard Professional Edition. There be telemetry data collect from gateway, and transmit to platform. Then we can monitor real-time data on the dashboard, and receive the alarm notification, that can help you manage IoT devices more efficiently.

1. Provision Device

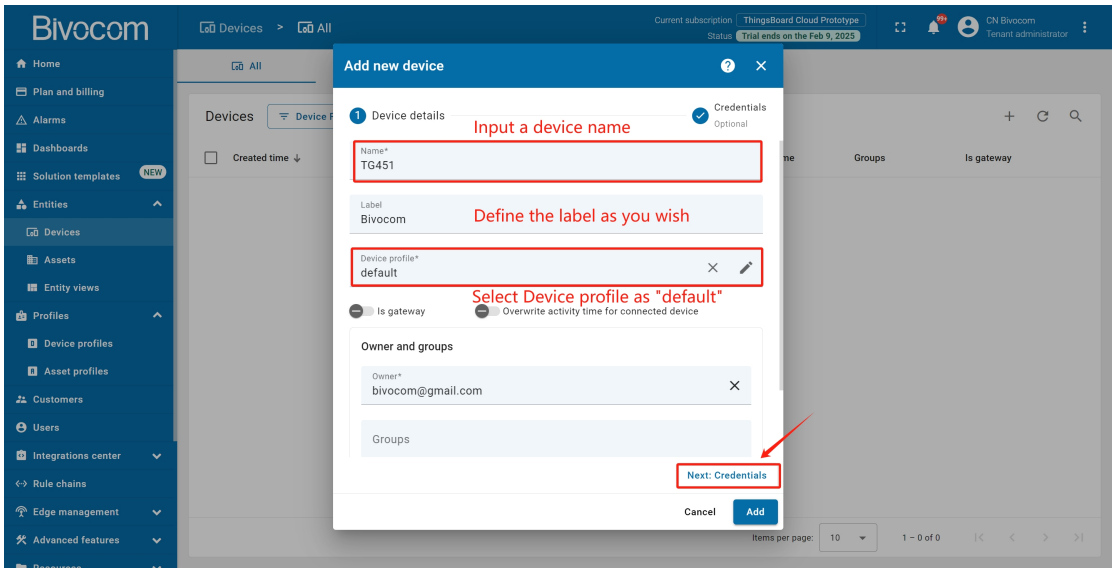
➢ Login to your ThingsBoard account and go to the "Devices" page of the "Entities" section.



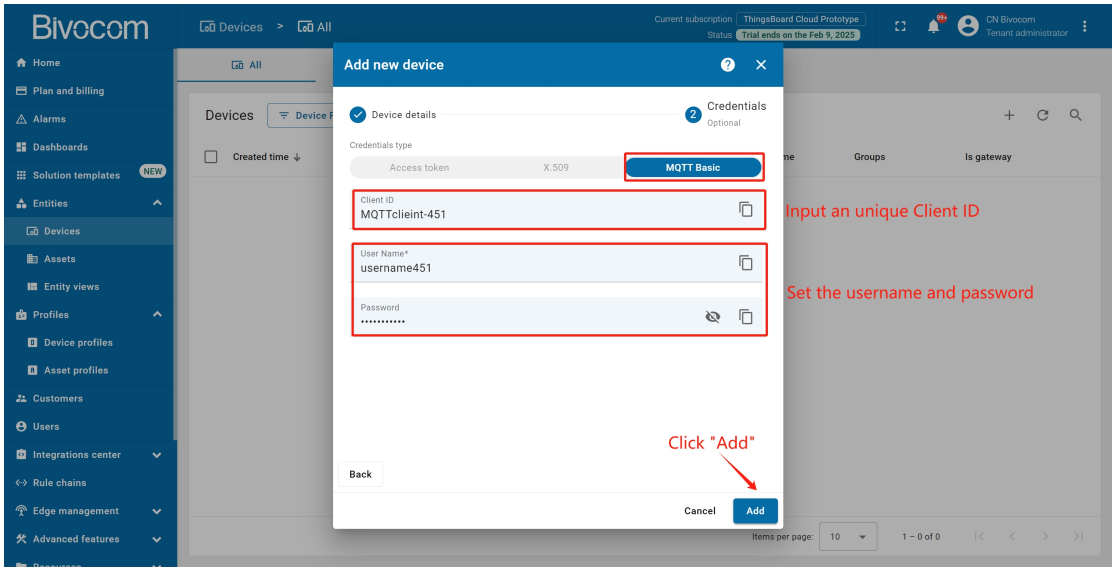
➢ By default, you navigate to the device group "All". Click on the "+" icon in the top right corner of the table and then select "Add new device" from drop-down menu.



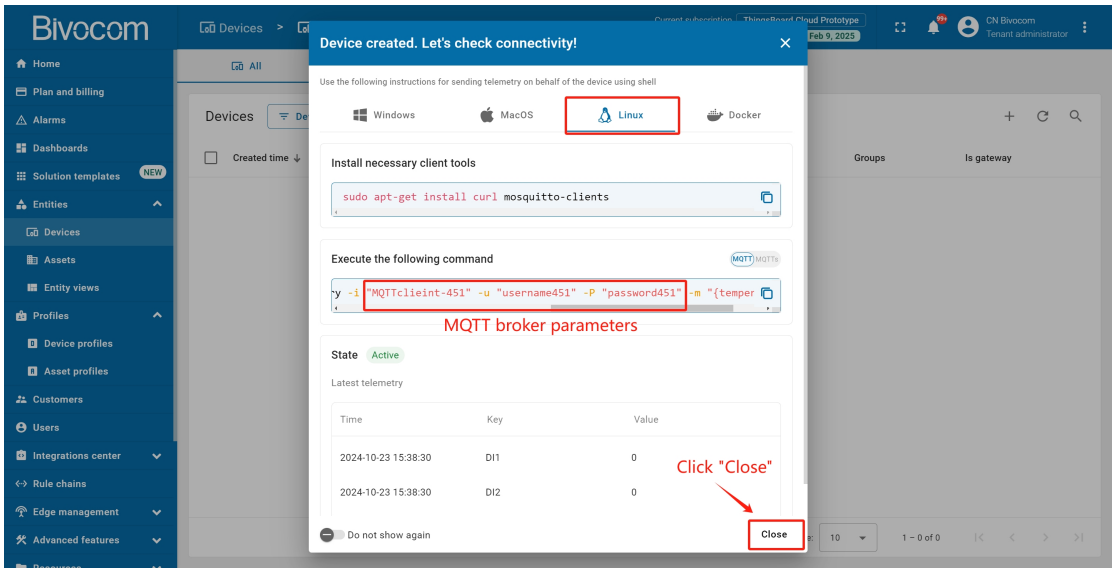
➢ Input a device name as you wish. You can define the label or not. Select “Device profile” as default. Then click "Next: Credentials".



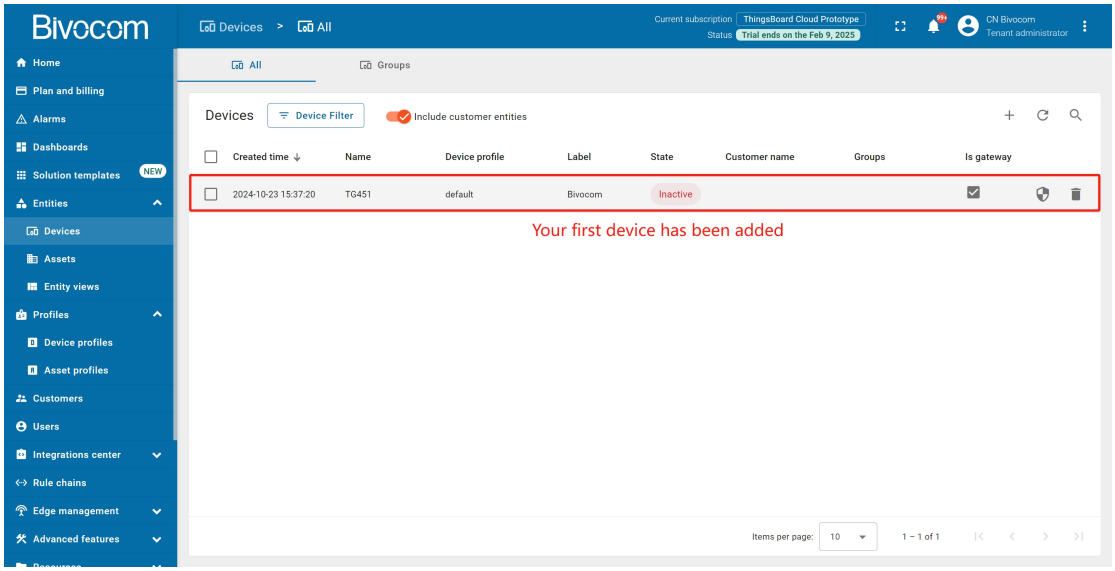
➢ In this case, the gateway is communicated to ThingsBoard via MQTT protocol. So we have to configure the credentials of MQTT broker. Click “MQTT Basic”, input an unique MQTT client ID, set the username and password of MQTT broker. Then click “Add” icon.



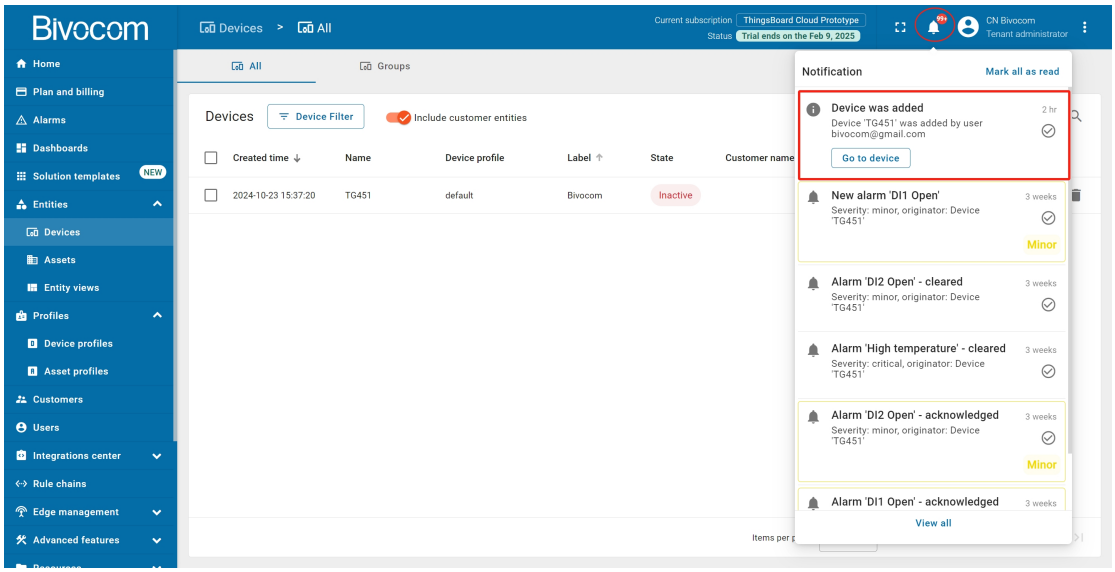
➢ A window will pop up where you can check the device's connection to ThingsBoard. Since my gateway is based on Openwrt system, the OS should be “Linux”. You can check the MQTT parameters just configured in this page. Let's close this window for now and return to checking the connection in the next step in more detail.



➢ Now your first device has been added, but it’s “Inactive” currently. As you add more devices, there will be added at the top of the table, since the table automatically sorts devices by their creation time, with the newest ones first.



➢ When adding a new device, you will receive a notification. You can view it by clicking on the "bell" icon in the top right corner.



2. Connect Device

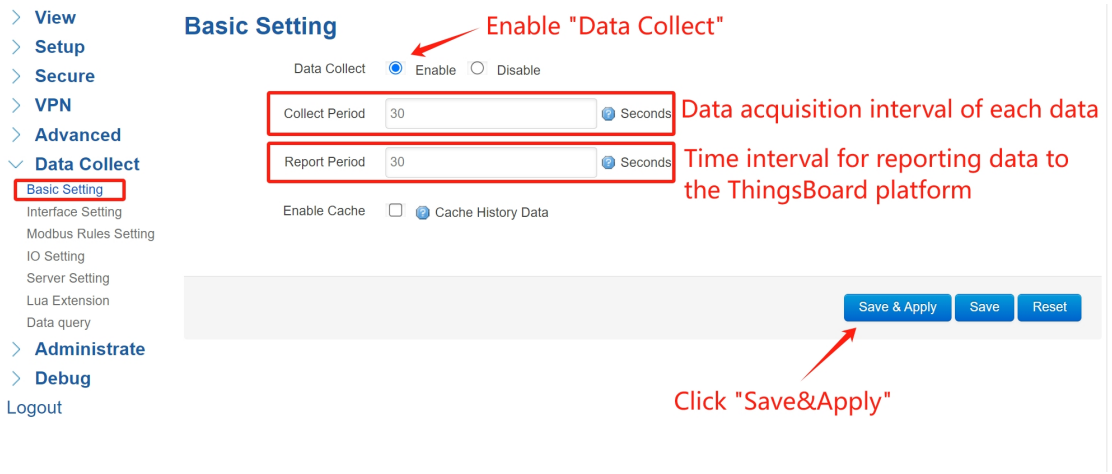
Now, let's connect gateway to the ThingsBoard platform. To accomplish this, use the "Check connectivity" functionality to publish telemetry data on behalf of your device. You can do this both while adding the device and after.

2.1 Gateway Configuration

Login to the WEBUI of your gateway via LAN IP. Default username and password are both "admin". Then click "Login" to access.

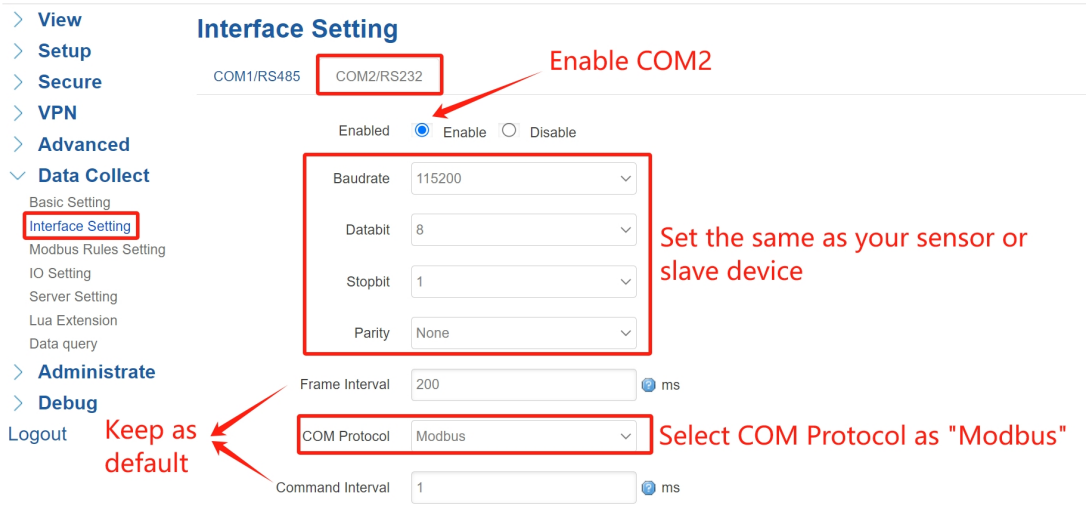
(1) Basic Setting

➢ Click Data Collect -> Basic Setting, enable "Data Collect" feature, set collect and report period according to your actual demands. Then click "Save&Apply".



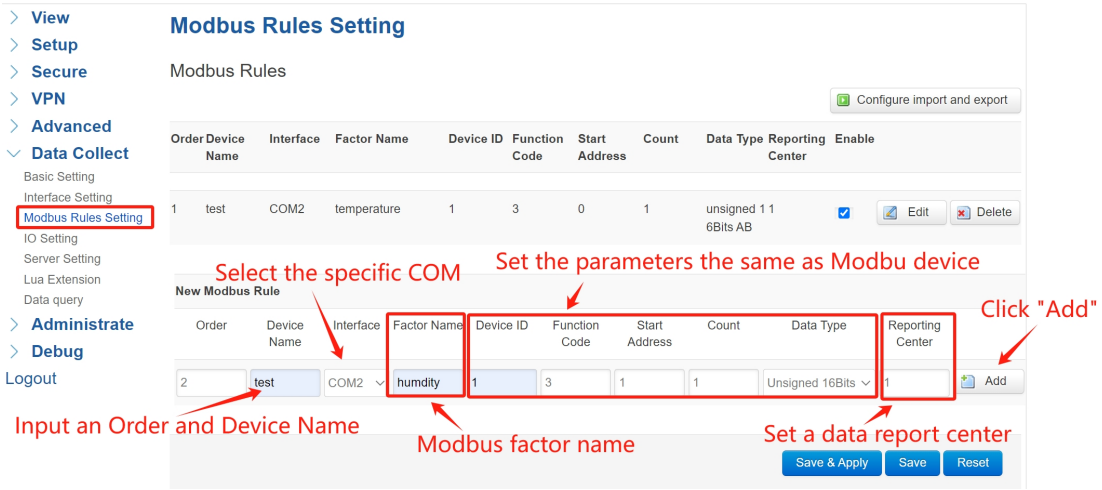
(2) Interface Setting

➢ Click Data Collect -> Interface Setting, select a COM which connect with your slave device. In this case, I use COM2-RS232 to connect gateway with slave. Enable COM2, configure the following parameters according to your sensor or slave setting. I'll collect Modbus RTU data using my gateway, so I select COM Protocol as "Modbus". Keep "Frame Interval" and "Command Interval" as default. After configured, click "Save&Apply".

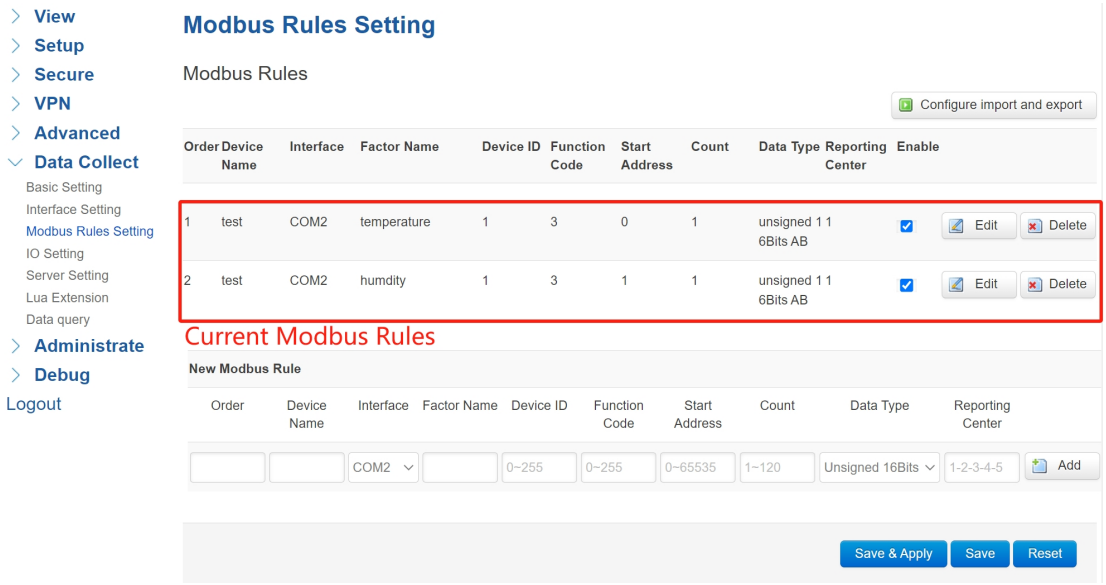


(3) Modbus Rules Setting

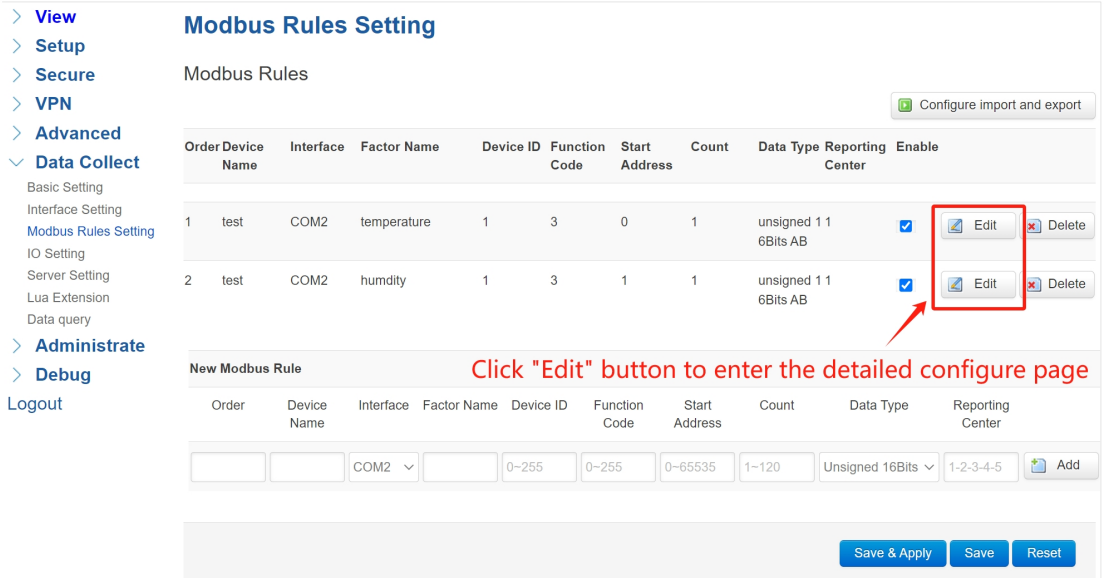
- In this case, the gateway is connected with typo device via RS232, so the communication protocol is Modbus RTU. We need to configure the Modbus rule according to Modbus device.
- Click Data Collect -> Modbus Rules Setting, add new Modbus rule for your device. Set the "Order" and "Device Name" as you wish. "Interface" select "COM2" . Define a Modbus factor name. Set the Modbus parameters the same as your sensor or slave device. Select a data reporting center according to your "Server Setting". After configured all parameters, click "Add" icon.



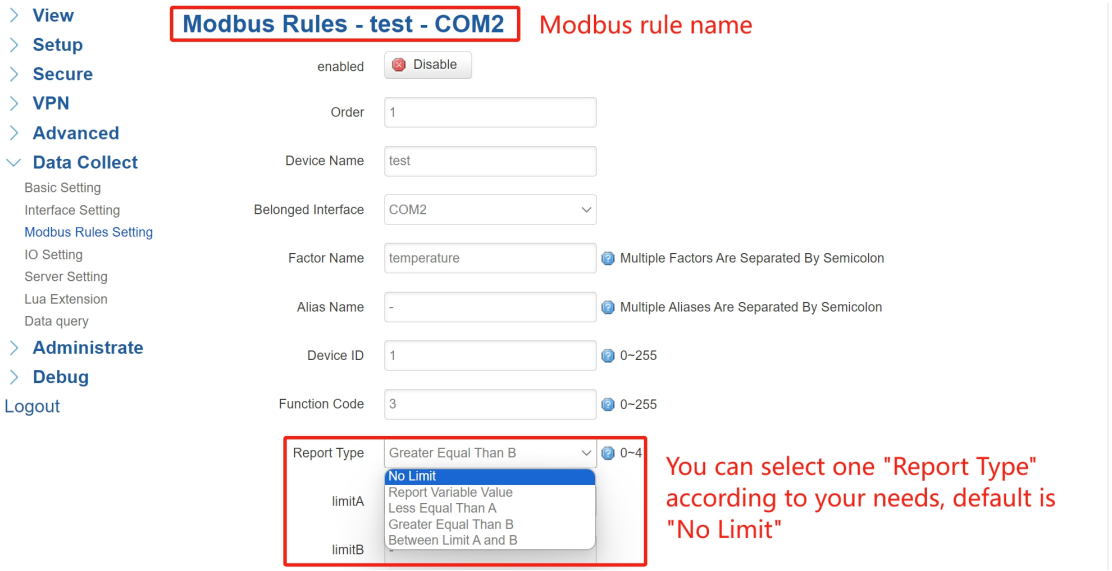
- After the new rules are added, they will be listed above in the order in which they were added.



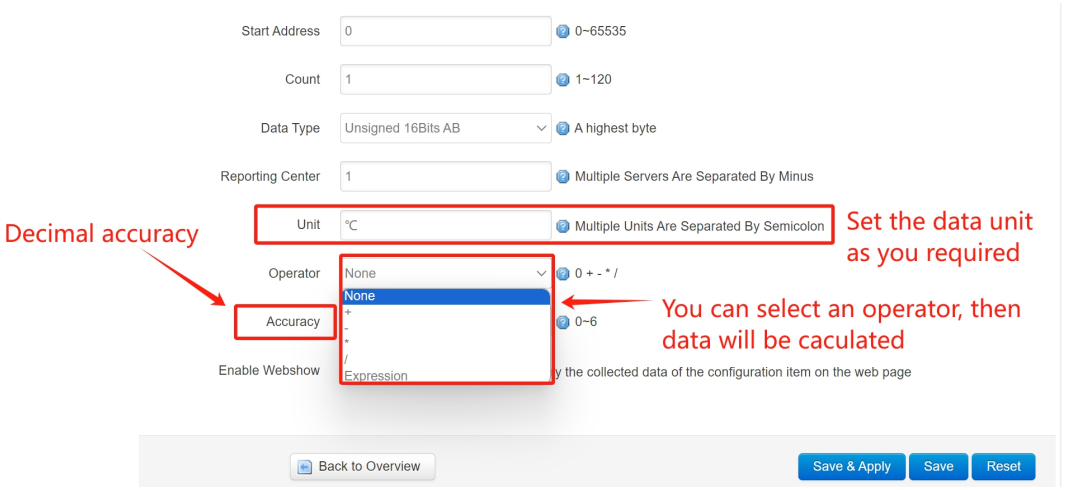
➢ Click “Edit” button, it’ll enter the detailed configure page for the selected Modbus rule.



➢ You can configure the extra parameters in this page. “Report Type” is set “No Limit” as default, you can configure other types and set the limit value according to your actual demands.



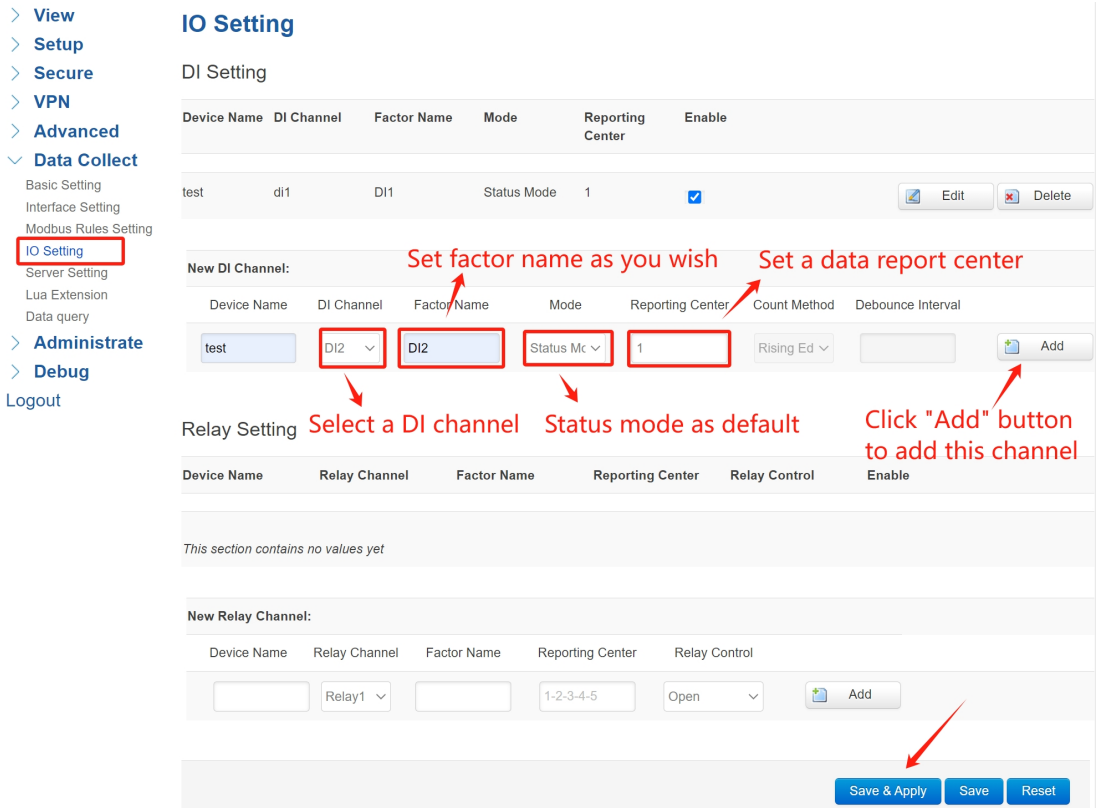
➢ You can set the data unit as you required, in this case, I set "℃" for temperature unit. Then you can select an operator for this rule.



➢ For example, I select the "Operator" as "+", "Operand" set as "10", then the calculated data should be "x=x+10". After setup all extra parameters, click "Save&Apply".

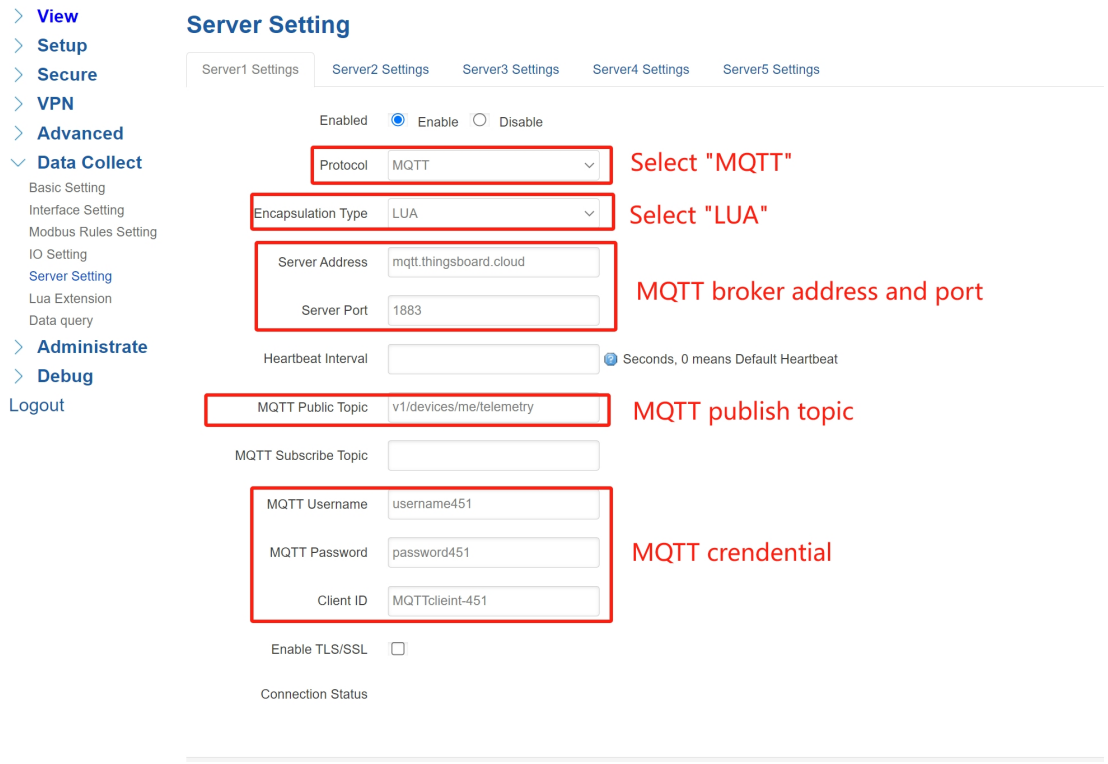
(4) IO Setting

➢ Click Data Collect -> IO Setting, in this case, I'll add 2 DI channels for monitoring the input signal. Select the DI channel of gateway, set a factor name as you wish, select "Status Mode" as default, set a data report center, then click "Add" button to add this channel. After added, click "Save&Apply".

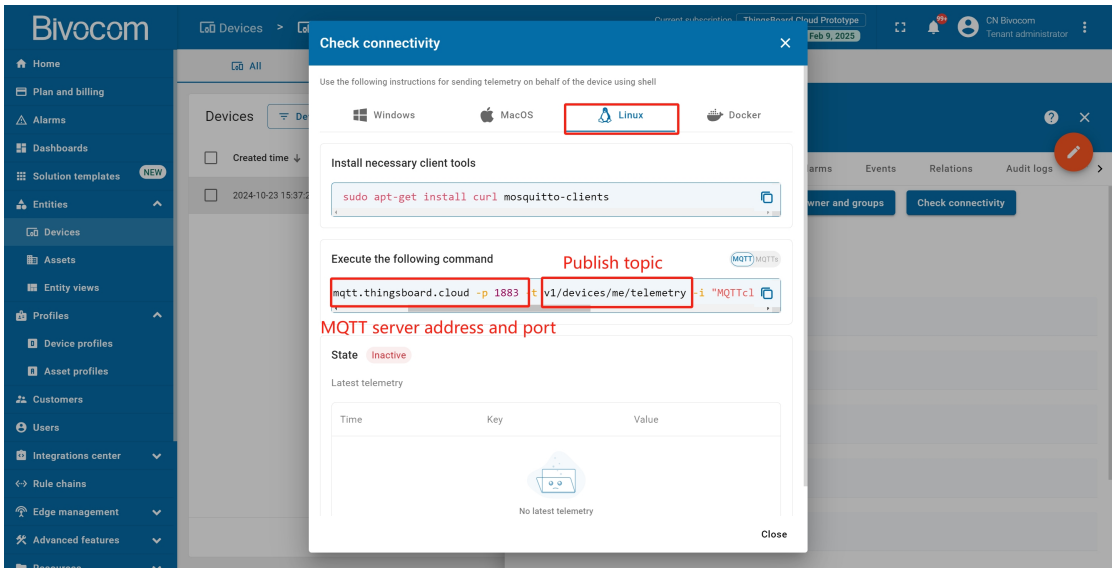


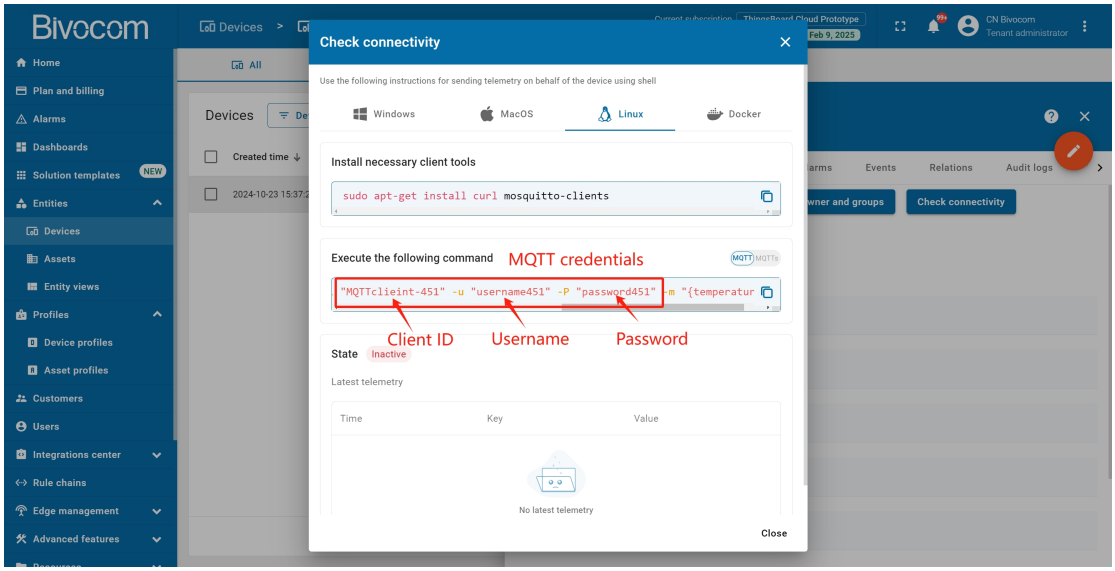
(5) Server Setting

➢ Click Data Collect -> Server Setting, here you can configure up to 5 data report servers. Enable a server setting, "Protocol" select "MQTT", "Encapsulation Type" select "LUA". MQTT Server Address: "mqtt.thingsboard.cloud", Server Port: "1883". Set MQTT parameters according to ThingsBoard setting.

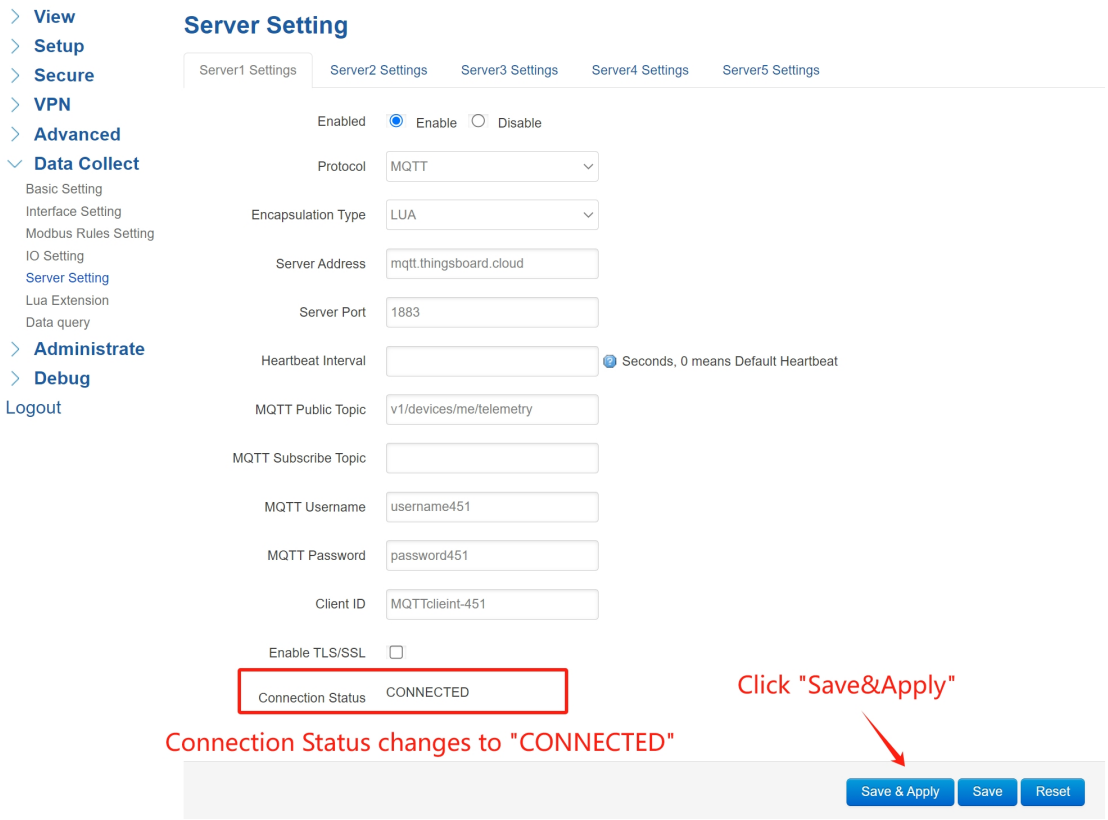


➢ MQTT parameters of ThingsBoard platform:





➢ After configured MQTT parameters, click "Save&Apply" icon, then the "Connection Status" will change to "CONNECTED", it indicates the gateway has connected to the ThingsBoard platform successfully.

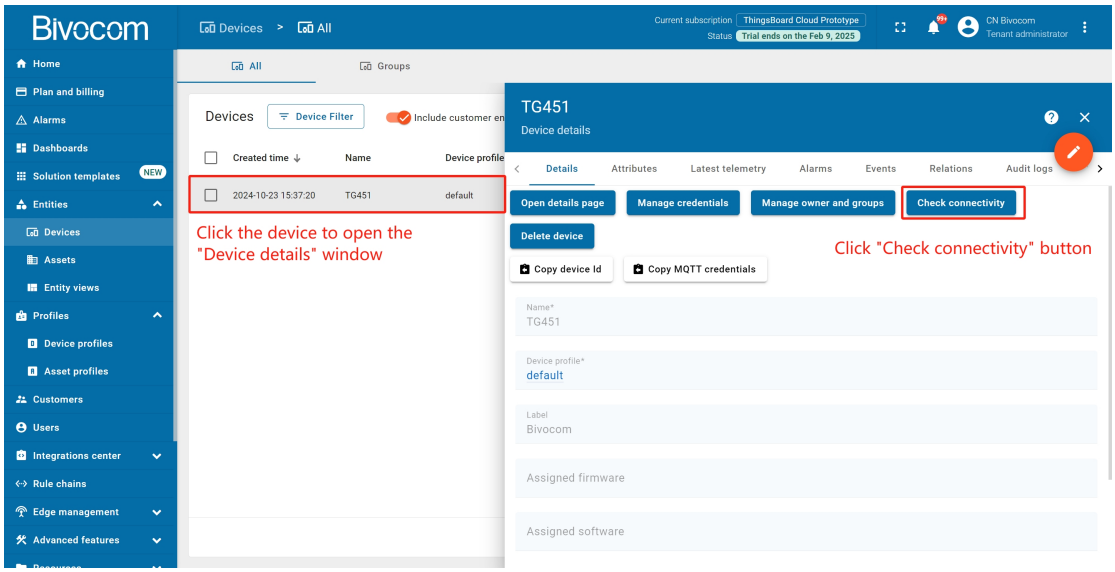


***Note:** Since the ThingsBoard platform has specific requirements for the data reporting format, the standard JSON of gateway cannot meet the requirements, so I modified the gateway's reporting format as "LUA" to adapt to the platform's demands. This means that special firmware for gateway needs to be customized to connect to the platform.

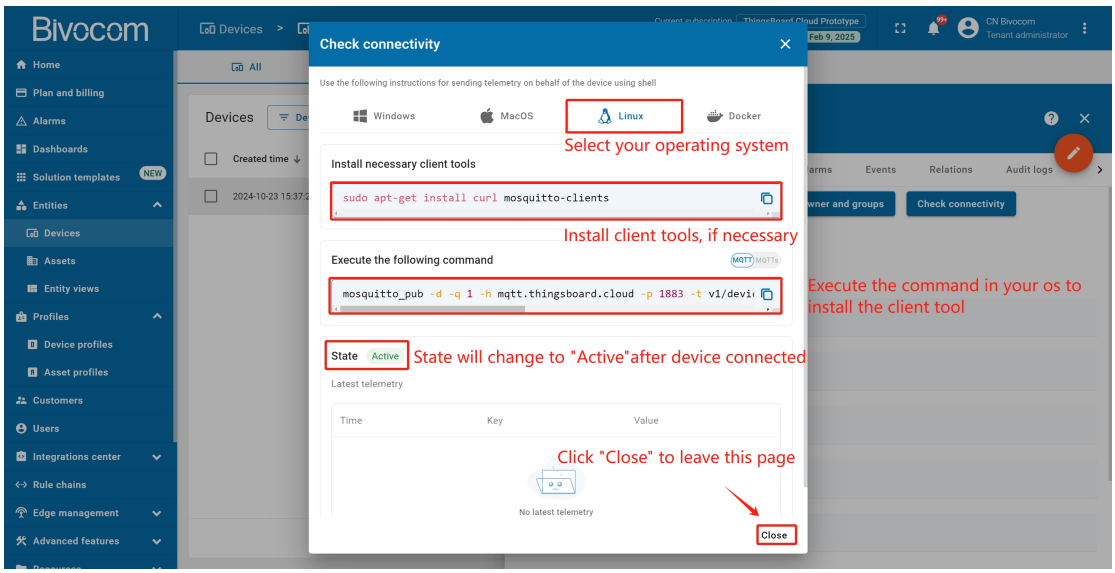
2.2 Check Connectivity

Now, the gateway has connected to ThingsBoard platform via MQTT successfully, next we have to check the connection on platform side, and test the communication between gateway and platform.

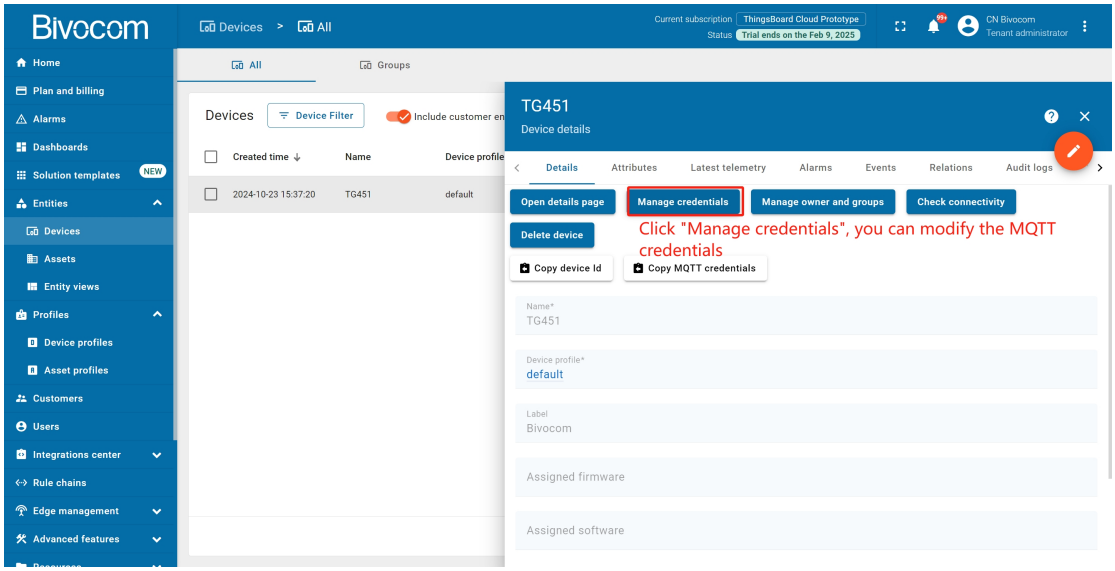
- Click on your device, and click the "Check connectivity" button in the "Device details" window.



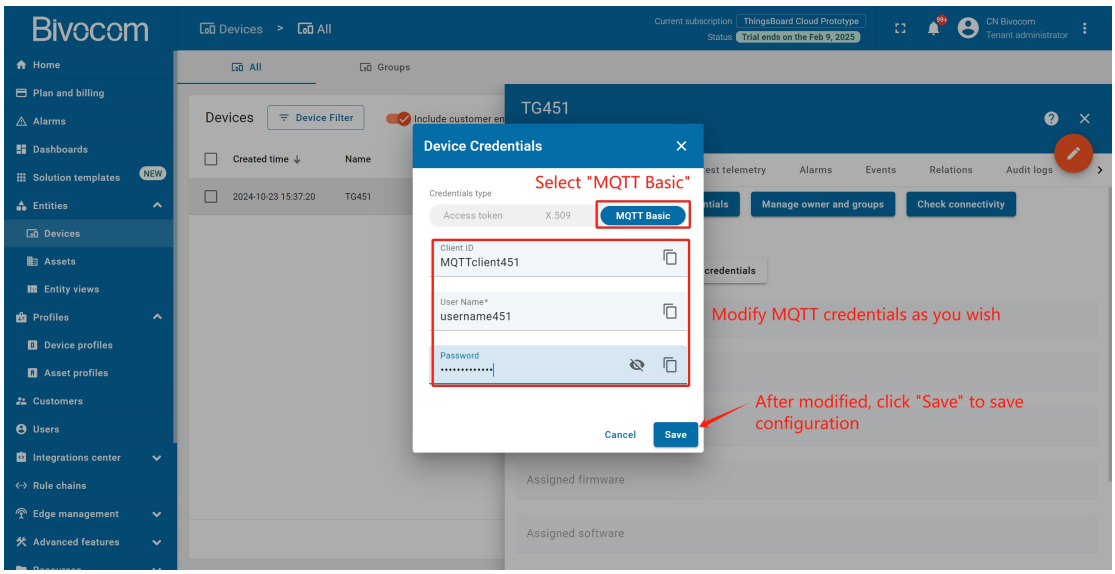
- In the opened window, select the operating system of your device. Execute the command and install the client tools in your OS if necessary. We already connected the gateway to platform in previous steps, then the state will change to “Active” after connected. After checked the connectivity, close the connectivity window.



- Click “Manage credentials”, you can modify the MQTT credentials.



➢ It'll enter "Device Credentials" page, select "MQTT Basic", then you can modify the MQTT Client ID, Username and Password as you wish. After modified, click "Save" to save the configuration.



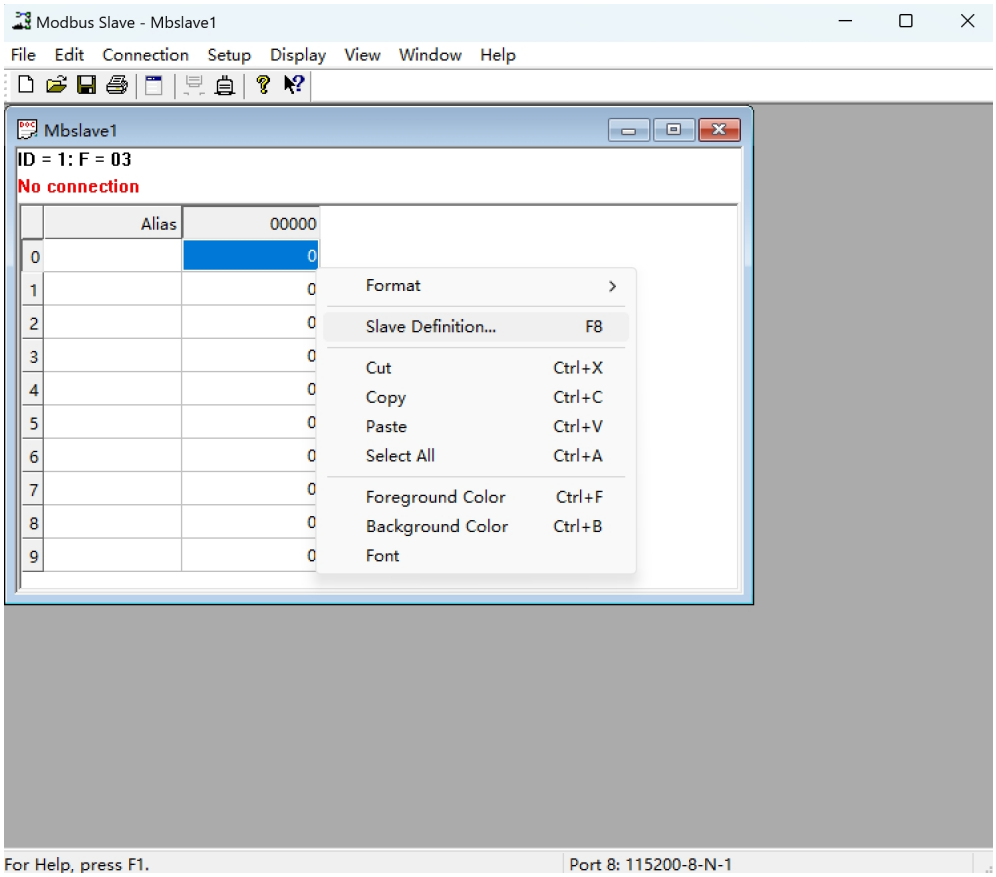
3. Data Transmission

Now, the gateway has already connected to ThingsBoard platform. Meanwhile, the gateway can work as a data logger to collect sensor data. Next step, we have to publish attribute/telemetry data from gateway to platform for testing the data transmission. In this case, I'll use "Mbslave" to simulate Modbus data. If you connect your gateway with an actual sensor or slave device, you can collect real data. In the above steps, we already added the Modbus rules for gateway configuration. Now, it's time to configure the Modbus slave.

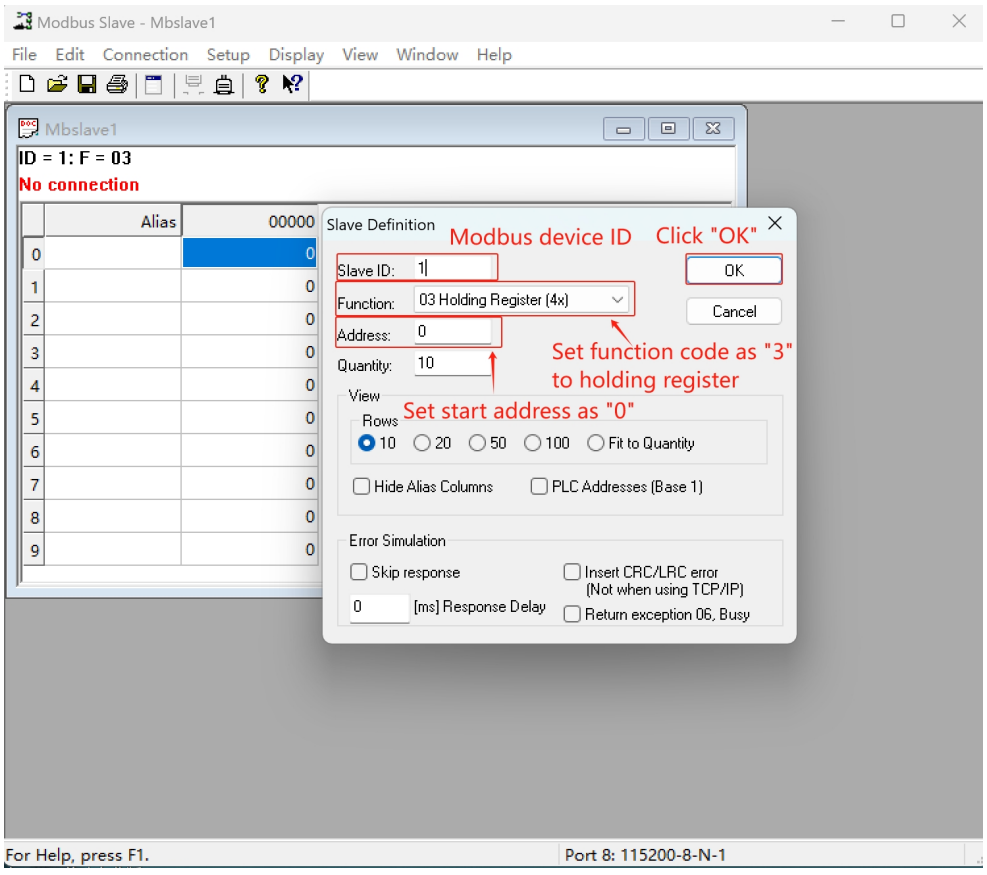
3.1 Mbslave Setting

(1) Slave Definition

➢ Right click the mouse, it'll pop up a window, click "Slave Definition".

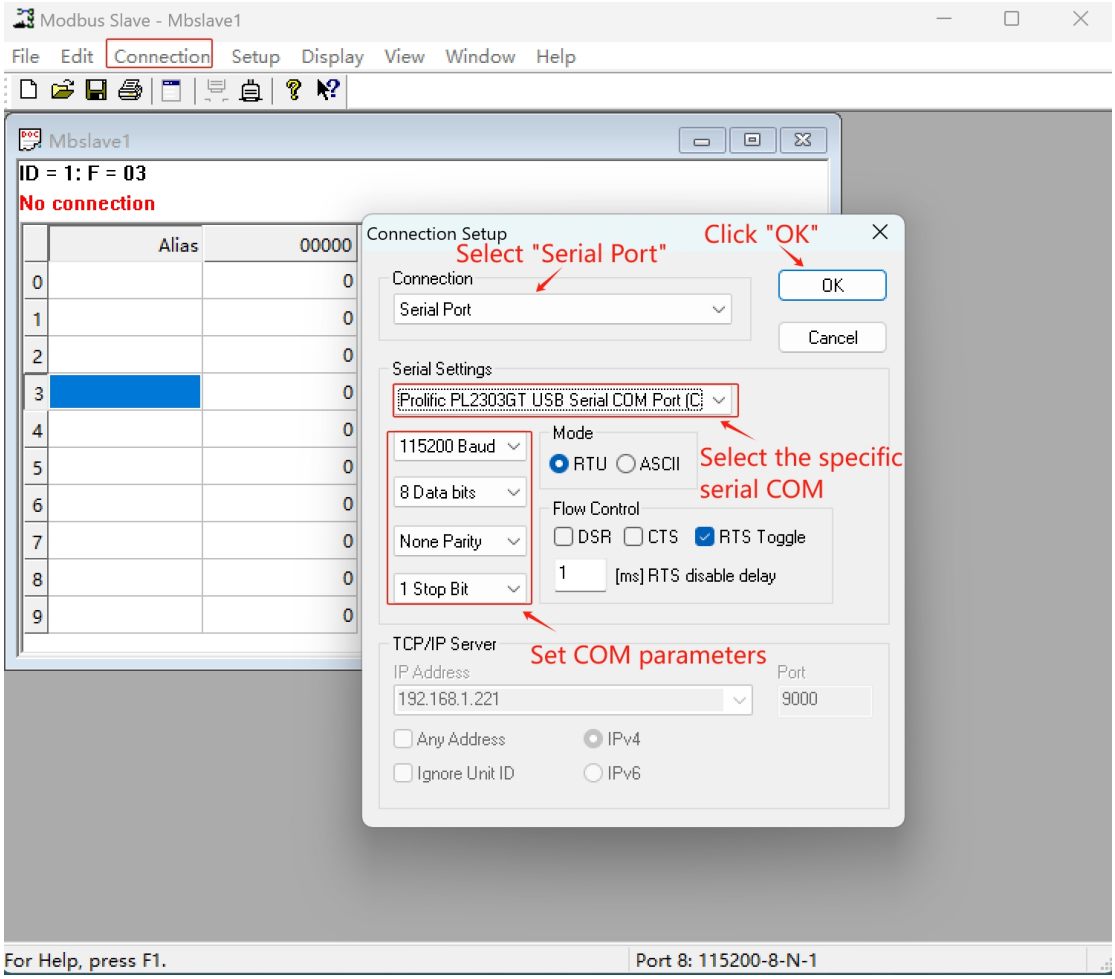


➢ Set Modbus device ID as “1”, set function code as “3” for holding register, set Modbus start address as “0” or others as you wish. After setup, click “OK” to save modification.



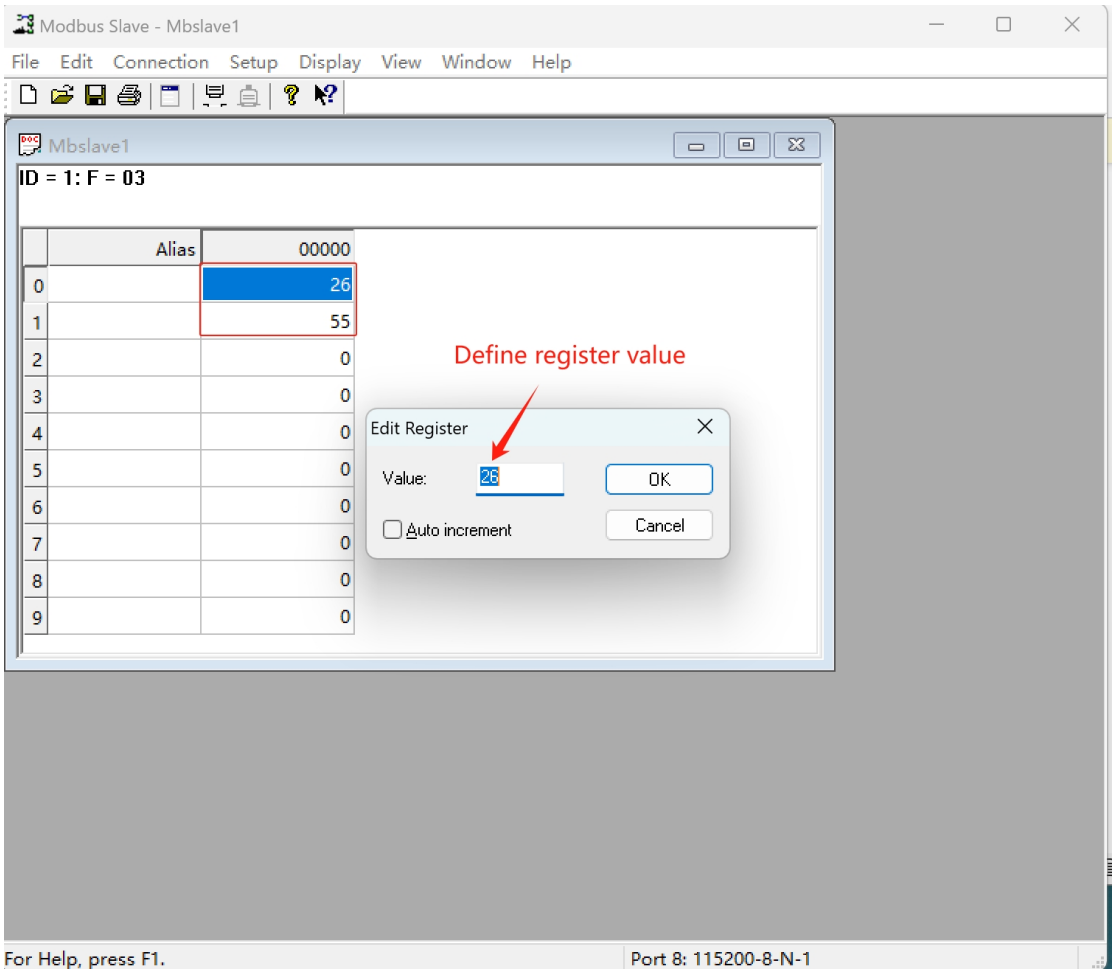
(2) Connection Setup

➢ Click “Connection” menu, it’ll show “Connection Setup”. Since I use RS232 to connect gateway with Modbus slave, the “Connection” should select “Serial Port”. Select the specific serial COM that you connected. Set COM parameters according to your needs.

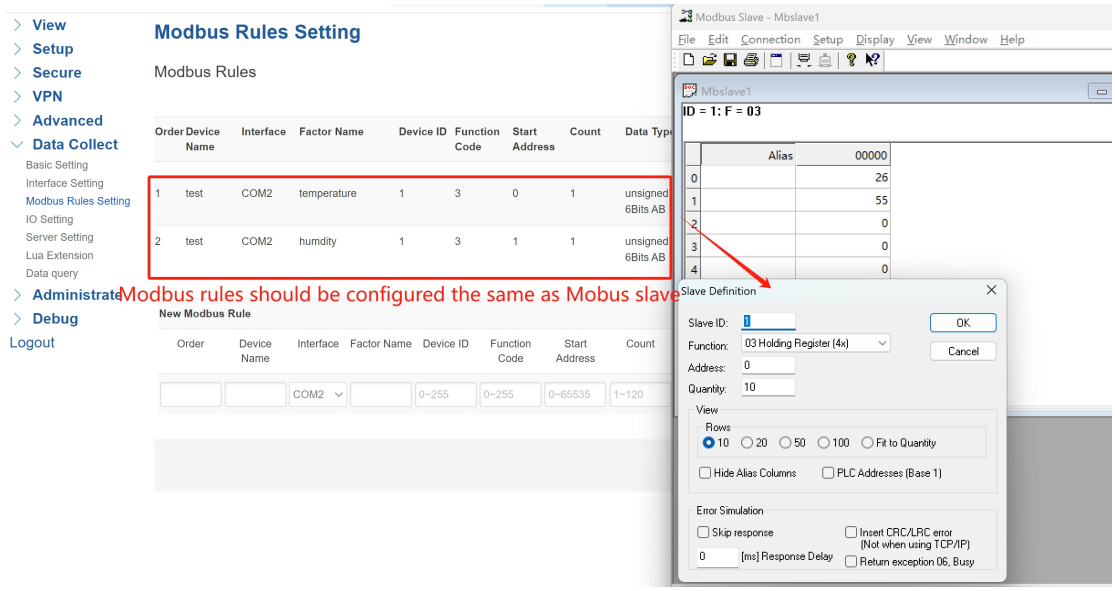


(3) Define Register Value

➢ Edit the specific register, define the register value as you wish. In this case, I set 2 registers, register 0 and 1.



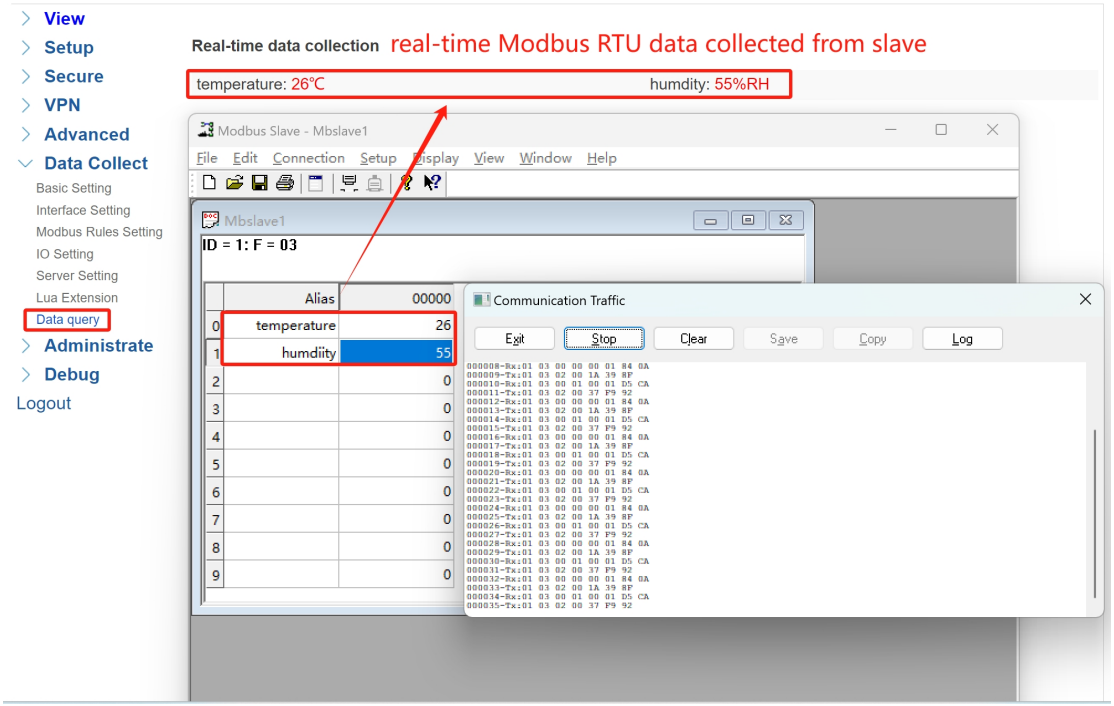
***Note:** Since the Modbus slave is a simulate tool for Modbus data, you can set the parameters as you wish. After setup Modbus slave, the Modbus rules setting of gateway must be configured the same as Mbslave.



3.2 Publish Data

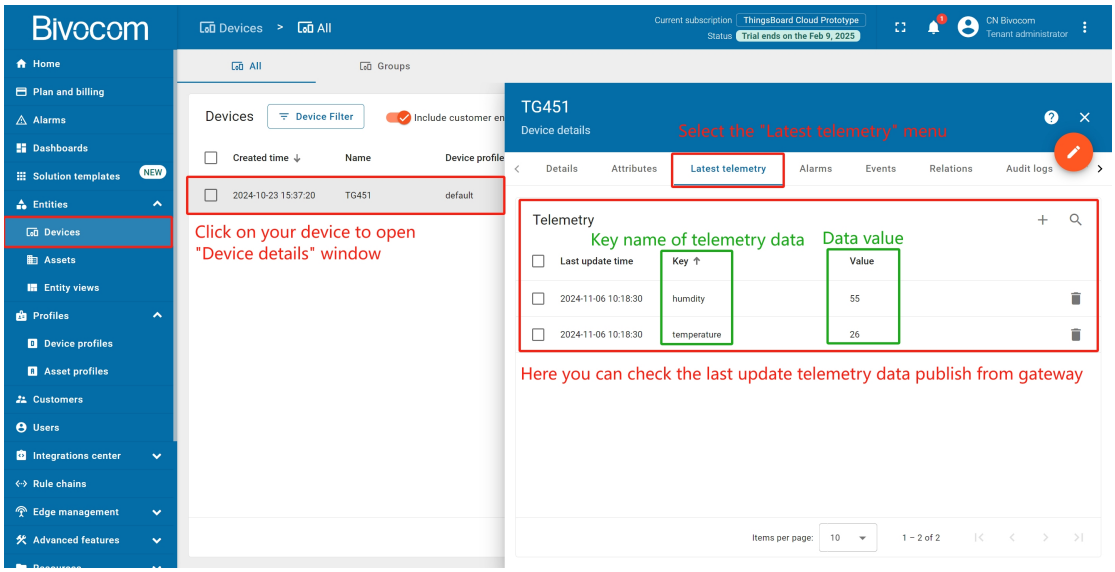
(1) Data Query

➢ Back to WEBUI of your gateway, click Data Collect -> Data query, you can check the real- time data collection in this page. We have to ensure that the gateway can collect the correct data from Mbslave.



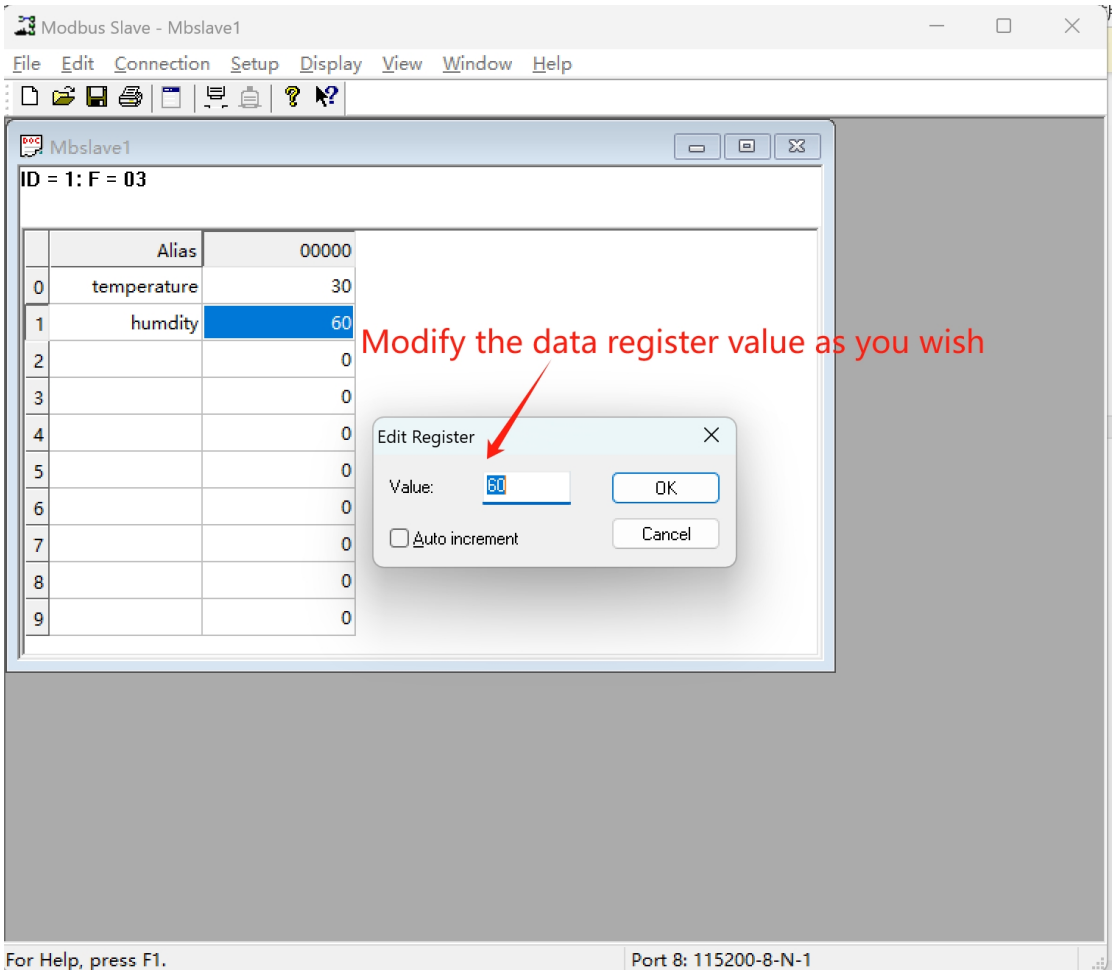
(2) Telemetry Data

➢ Come to the platform side, and go to the “Device” page. Click on your device, and click “Latest telemetry” button in the “Device details” window. Then, it’ll show the last update telemetry data in the opened window.



(3) Modify Data

➢ Now, I'm able to modify the Modbus data through the Mbslave, I can change the data to whatever I want.



➢ Back to the platform, check whether the data has been updated. Get into "Device detail" page, you can see the telemetry data has been successfully changed the same as Mbslave.