ICP DAS CO., LTD.

UA-5200 Series User Manual

IIoT Communication Server

www.icpdas.com

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, starting from the date of delivery to the original purchaser.

Warning

ICP DAS CO., LTD. assumes no liability for damages consequent to the use of this product. ICP DAS CO., LTD. reserves the right to change this manual at any time without notice. The information furnished by ICP DAS CO., LTD. is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS CO., LTD. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright and Trademark Information

Trademark of Other Companies

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Technical Output:

Please contact local agent or email problem-report to output@icpdas.com .
For more product information, please refer to www.icpdas.com .

Technique: Sun Chen; Translation & Edition: Eva Li / Janice Hong; R&D Dept., ICP DAS CO., LTD.
Latest edited by: Eva Li; V.3.0.0, May 2017

Table of Contents

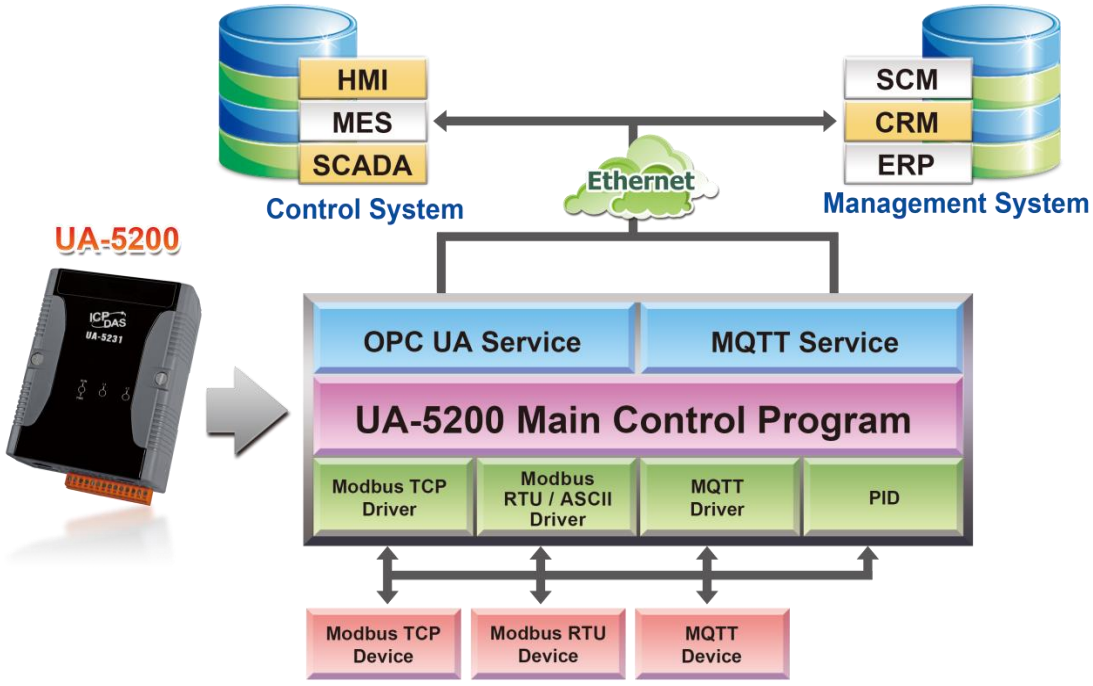
1. UA-5200 IIoT Communication Server

This chapter introduces the UA-5200 and its functions, software/hardware specifications...

1.1. Introduction

The **UA-5200** is a series of data acquisition controller and also an IIoT communication server by ICP DAS (IIoT: Industrial Internet of Things). The UA-5200 built-in **OPC UA Server** and **MQTT Client Service** support a variety of common industrial communication protocols. Its RISC-based CPU architecture has the advantages of small size and low power consumption that lets this series can be placed in a small space to fit a variety of rooms, equipment and case environment. In the hardware, it provides a variety of communication interfaces, such as Gigabit Ethernet, USB, RS-232 and RS-485… ports to connect diverse devices.

Applying the **OPC UA**, the UA-5200 can integrate the I/O products and the third-party devices, import their data to the back-end SCADA management system or the big-data analysis/decision system, to satisfy the reliability, interoperability and security needs of the Industrial 4.0 automation system. Using the **MQTT** active communications to bridge the Internet of Things (IoT) and transmit the statuses of various devices by the cloud-based interaction so that to meet the current trend of the IIoT and achieve the full smart automation system based on **Industry 4.0**.



1.2. Function Features

- **Web-based UI**

  With the Web-based User Interface, users can log in and configure the controller via a normal web browser that only need a mobile device or computer with web browsing capabilities.

- **OPC UA Server: IEC 62541 Standard**

  The OPC UA Server certified by the OPC Foundation can assist the integration for the local-end devices, actively upload data to the application system, and support to across the multiple platforms.

- **PID Logic Operation**

  The PID function can dynamically combine the remote I/O devices for the PID logic control to provide temperature control and case field solutions.

- **Support Modbus TCP/RTU/ASCII Master**

  Through the controller's RS-485, RS-232 and Ethernet ports can connect to the Modbus TCP/RTU/ASCII Slave devices. Build systems with scalability and flexibility to meet the diverse application needs and expansion at any time.

- **MQTT Broker Inside**

  Compliance with MQTT v3.1.1 protocol. Support MQTT message distribution management. Users do not need to build Broker system when using MQTT communications.

- **Support MQTT Protocol**

  Support MQTT to allow the IoT devices communicating with the OPC UA system and the UA-5200 conducting the data acquisition and management; and also can convert and publish the devices' data under the UA-5200 to the IoT system.

UA-5200 Function Overview		
Web-based UI	Built-in Web-based User Interface	
Flexible System Configuration	Variable Table/Communication Task Dynamic Editor	
OPC UA	Compliance with IEC 62541 Standard Cross-platform Data Integration (DA/AE/HDA) Transmission Security SSL Encryption Active Transmission Support Redundancy Support Remote Function Call	
MQTT Broker Inside	Built-in MQTT Broker, Compliance with MQTT V.3.1.1 Protocol	
PID Logic Operation	Dynamic Combination of I/O Devices for PID Logic Control	
Service (Output) Up to Interact with the Host	Protocol	OPC UA Server MQTT Client
	Interface	Ethernet Data Transmission
Driver (Input) Down to Interact with the I/O Modules	Protocol	Modbus RTU/ASCII/TCP MQTT
	Interface	RS-232/RS-485 Ethernet Data Transmission

1.3. Hardware Specifications

(Available Soon)

Model	UA-5231	UA-5231M	UA-5231M-3GWA
System Software			
OS	Linux Kernel 3.2.14		
Embedded Service	SFTP server, Web server, SSH		
CPU Module			
CPU	AM3354, 1GHz		
DDR3 SDRAM	512 MB		
Flash	512 MB		
FRAM	64 KB		
Expansion Flash Memory	microSD socket with one 4 GB microSD card (support up to 32 GB microSDHC card)		
RTC (Real Time Clock)	Provide second, minute, hour, date, day of week, month, year		
64-bit Hardware Serial Number	Yes, for Software Copy Protection		
Dual Watchdog Timers	Yes		
LED Indicators	4 LEDs (Power, Running and 2 user defined LEDs)		
Rotary Switch	Yes (0 ~ 9)		
VGA & Communication Ports			
VGA & Communication Ports	Yes, resolution: 640 × 480, 800 × 600, 1024 × 768, 1280 x 720		
Ethernet	RJ-45 x 1; 10/100/1000 Based-TX (Auto-negotiating, Auto MDI/MDI-X, LED indicators)		
USB 2.0 (host)	1		
Console Port	RS-232 (RxD, TxD and GND); Non-isolated		
ttyO2	RS-485 (Data+, Data-); Non-isolated		
ttyO4	RS-232 (RxD, TxD and GND); Non-isolated		
ttyO5	RS-485 (Data+, Data-); 2500 V_{DC} isolated		
Mechanical			
Dimensions (W x L x H)	91 mm x 132 mm x 52 mm	117 mm x 126 mm x 58 mm	
Installation	DIN-Rail Mounting		
Environmental			
Operating Temperature	-25 ~ +75°C		
Storage Temperature	-40 ~ +80°C		
Ambient Relative Humidity	10 ~ 90% RH (non-condensing)		
Power			

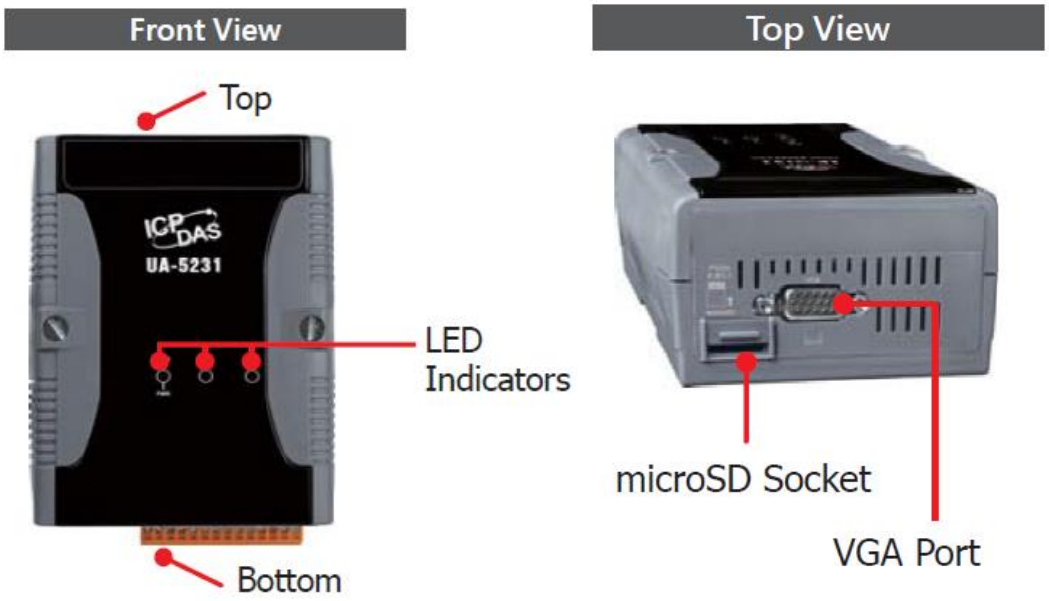
Model	UA-5231	UA-5231M	UA-5231M-3GWA
Input Range	+12 ~ +48 VDC		
Consumption	4.8 W		6.5 W
GSM System			
Frequency Band	-		GSM: 850/900/1800/1900 MHz
GPRS Connectivity	-		GPRS class 12/10; GPRS station class B
Data GPRS	-		Downlink transfer: Max. 85.6 kbps; Uplink transfer: Max 42.8kbps
3G System			
Frequency Band	-		WCDMA 850/900/1900/2100 MHz
Data Transmission	-		WCDMA / HSPA+ Download: Max. 14.4 Mbps; Upload: Max. 5.76Mbps

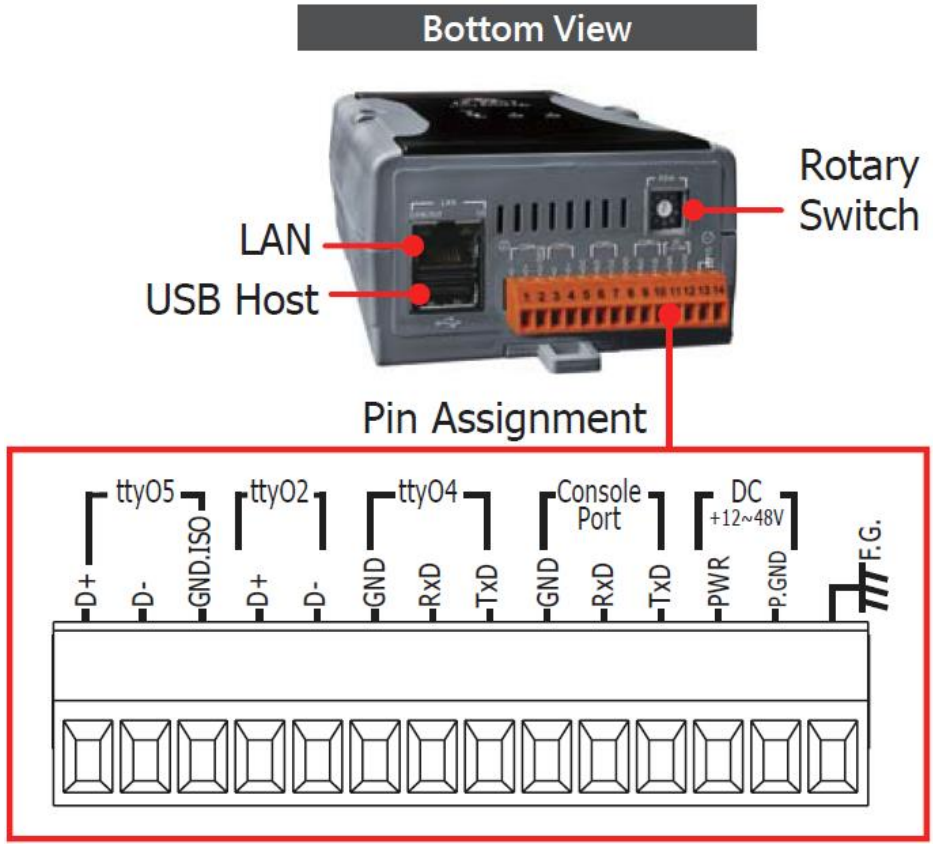
1.4. Software Specifications

Model	UA-5200Series
OPC UA	
OPC UA Server	● OPC Unified Architecture: 1.02 ● Core Server Facet ● Data Access Server Facet ● Method Server Facet ● Client Redundancy Facet ● UA-TCP UA-SC UA Binary ● User Token User Name Password & X509 Certificate ● Security Policy ◦None ◦Basic128Rsa15 • Sign • Sign & Encrypt ◦Basic256 • Sign •Sign & Encrypt
Modbus Master	
Modbus TCP	To read or control the devices that support standard Modbus TCP Slave protocol. Recommend to keep the maximum number of devices within 100 connections.
Modbus RTU/ASCII	A max.of 3 ports: ttyO2, ttyO4, ttyO5 to connect other Modbus RTU Slave devices (e.g. M-7000). Recommend no more than 32 devices per port for better communication quality.
MQTT	
MQTT Client	Connect the MQTT Broker to read/control the devices supporting the MQTT protocol.
MQTT Service	Connect the MQTT Broker to externally read/control the devices supporting other protocols that linking with the UA-5200 series.
MQTT Broker	Compliance with MQTT v3.1.1 protocol. Support MQTT message distribution management. Recommend to keep the connection number of Client within 400.
Virtual Device	
PID Function	Combine the remote I/O devices for the PID logic control system.

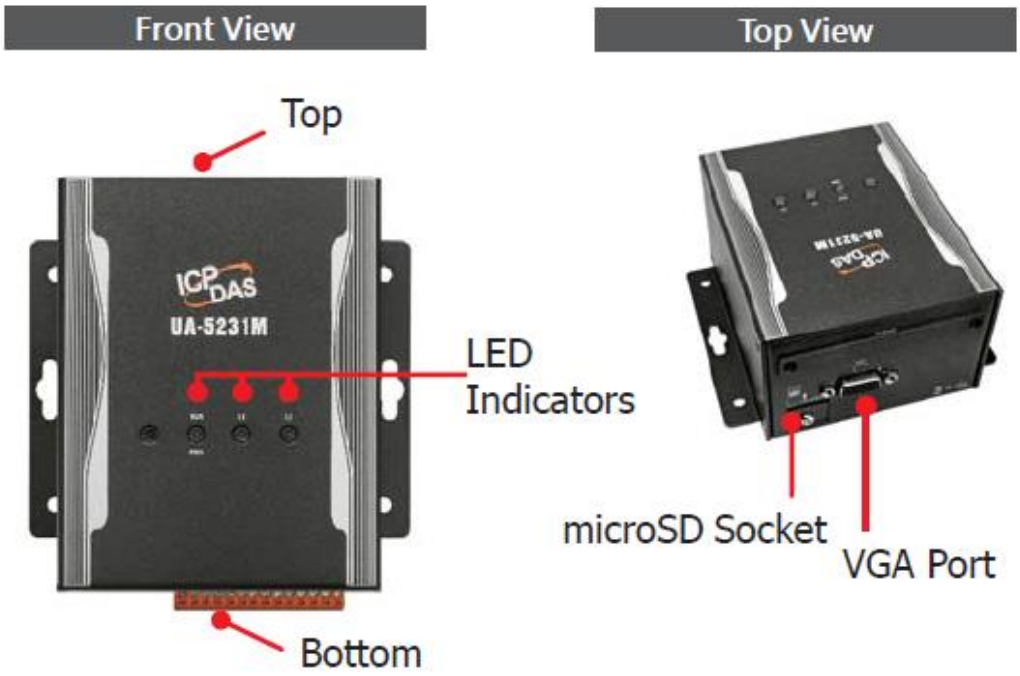
1.5. Appearance

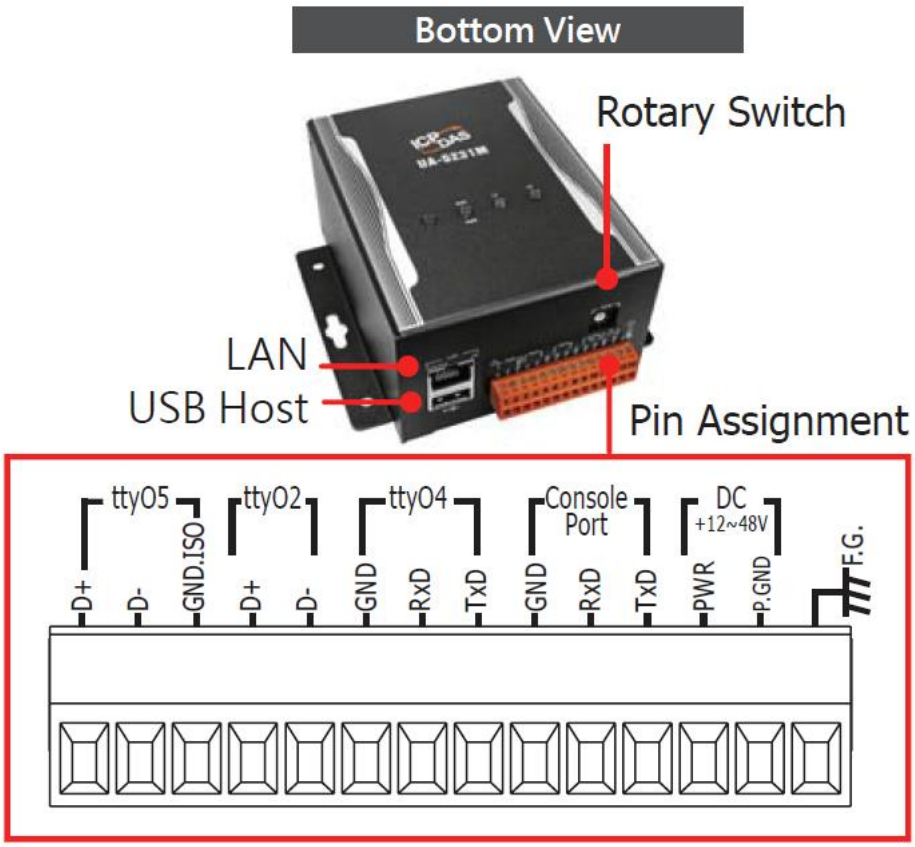
UA-5231



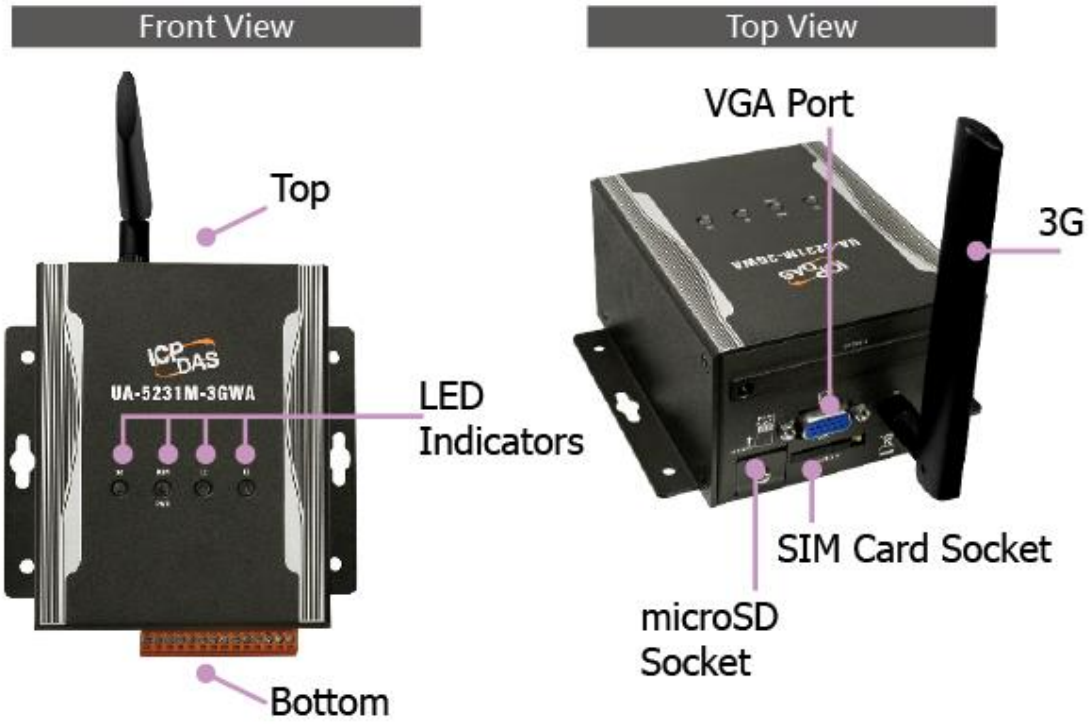


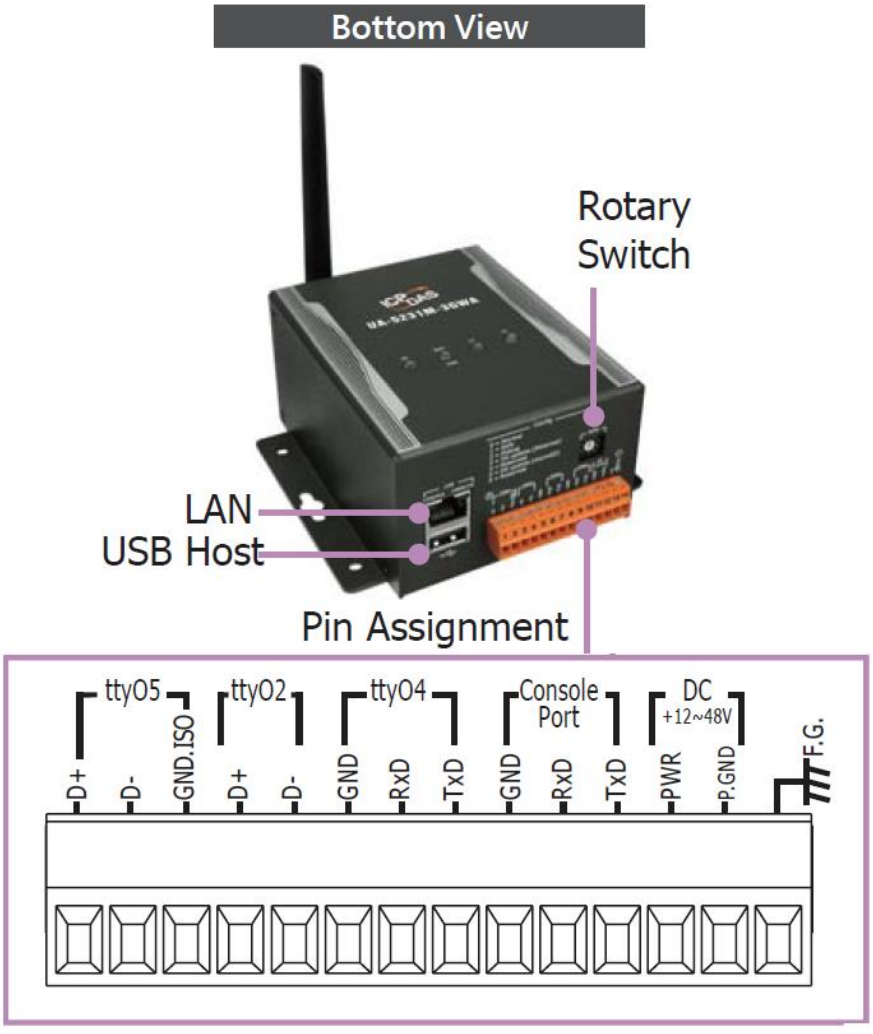
UA-5231M


Front View
Top View
Top
ICP DAS
UA-5231M
LED
Indicators
microSD Socket
VGA Port
Bottom


Bottom View
Rotary Switch
LAN
USB Host
Pin Assignment
tty05
tty02
tty04
Console Port
DC +12~48V
D+
D-
GND.ISO
D+
D-
GND
RxD
TxD
GND
RxD
TxD
PWR
P.GND
F.G.

UA-5231M-3GWA (Available Soon)

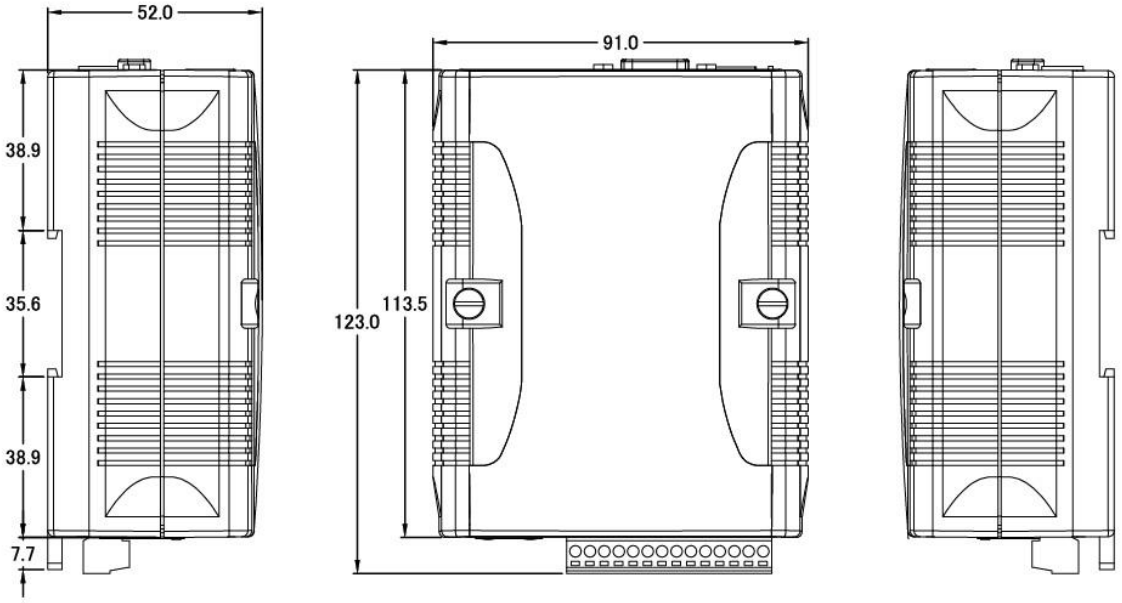




1.6. Dimensions

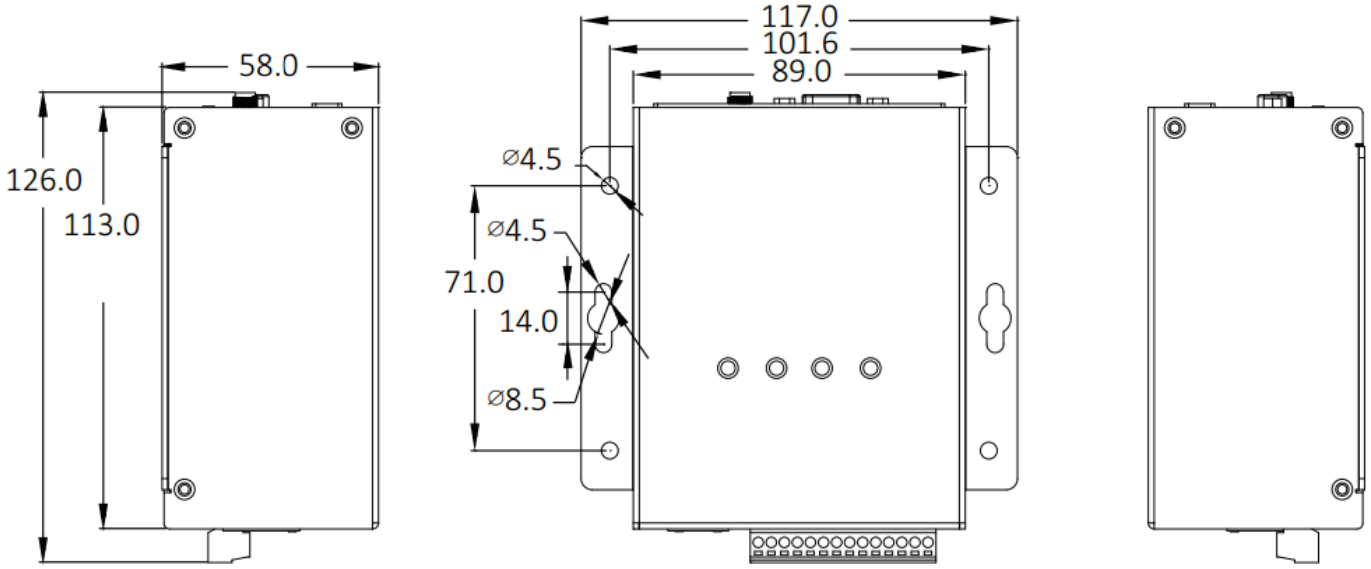
UA-5231

Unit: mm



UA-5231M/UA-5231M-3GWA

Unit: mm



2. Quick Start

This chapter describes the process of creating a UA-5200 project, including how to connect to the UA-5200 web-based UI via a browser, set web functions step-by-step, and complete a project.

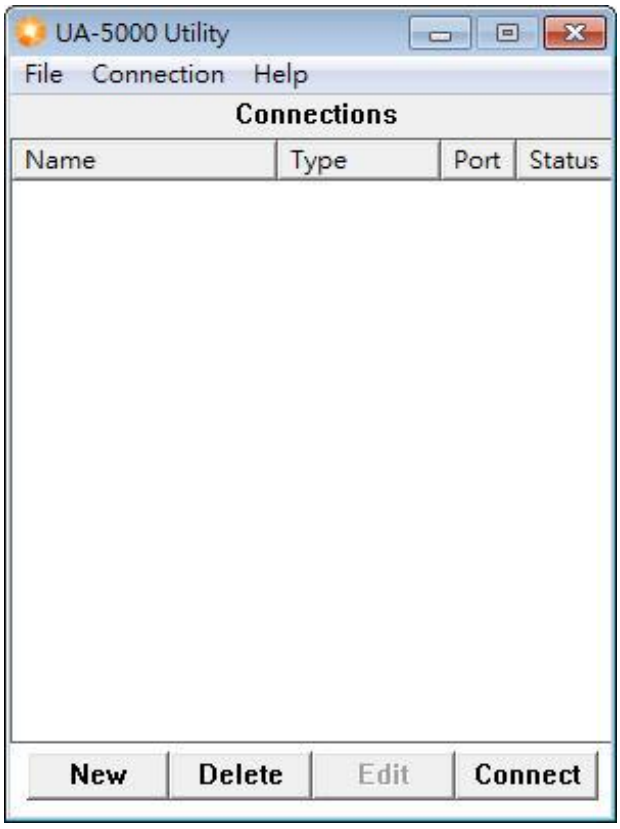
2.1. Link to UA-5200 Web-based UI

The following steps will show you how to connect to the UA-5200 web interface.

Using the **UA-5000 Utility** (named "UA-5000utility.exe") at the path of the companion CD (i.e., **CD:/UA-5000/Utility/**).Please copy this file to your PC, and then run it to connect the device.

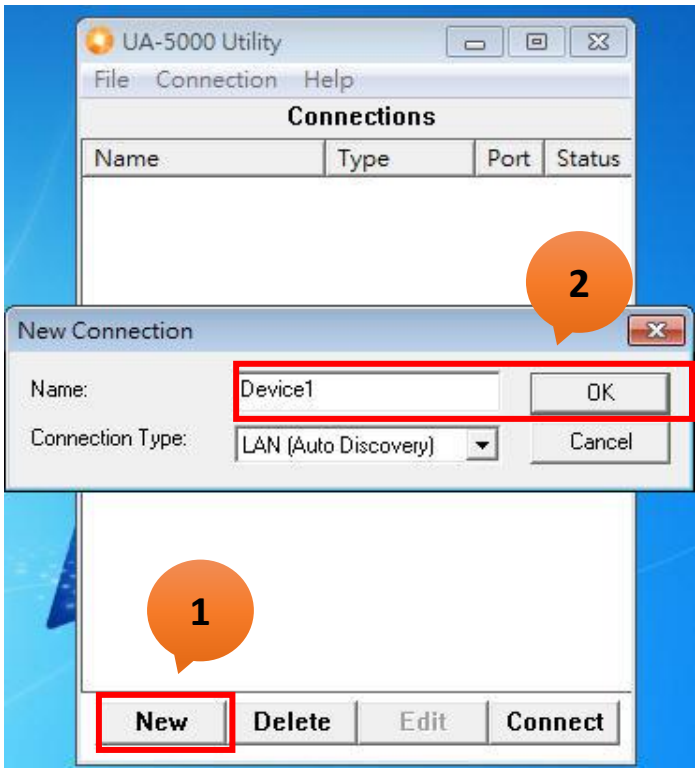
Step 1

Run the UA-5000 Utility (file name: **UA-5000utility.exe**) to install the Utility program.



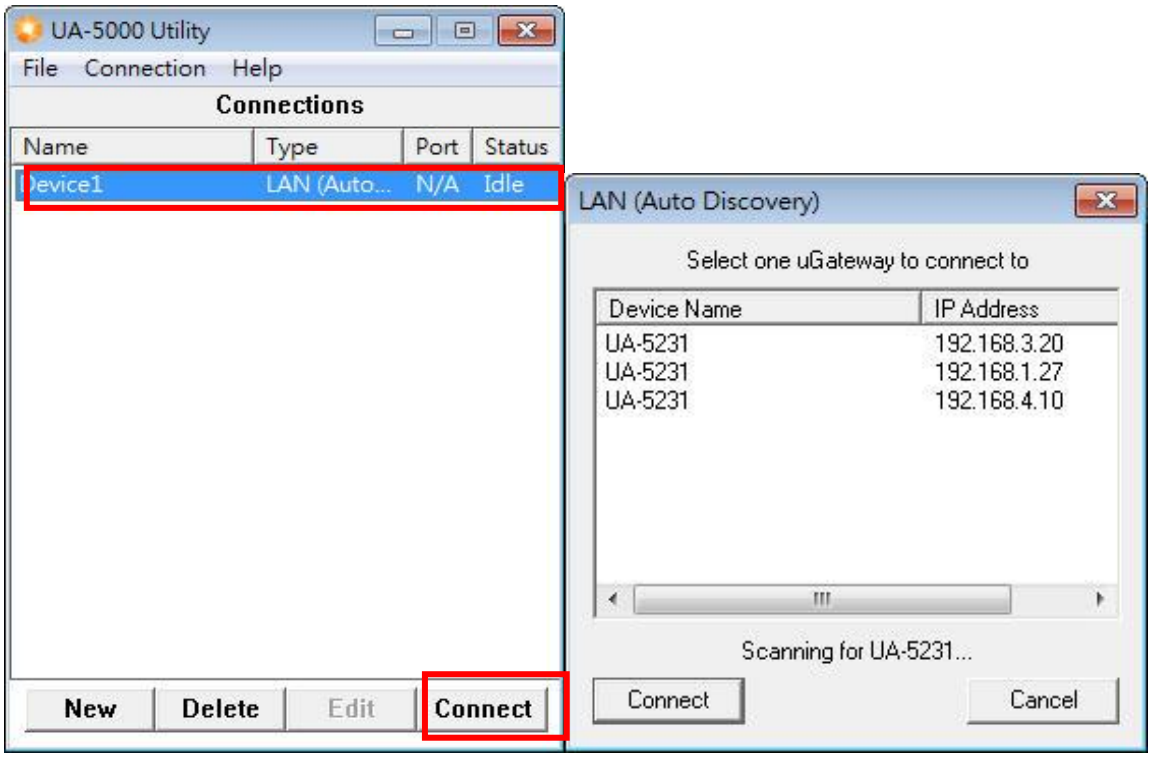
Step 2

Add a connection item and give a name for it.



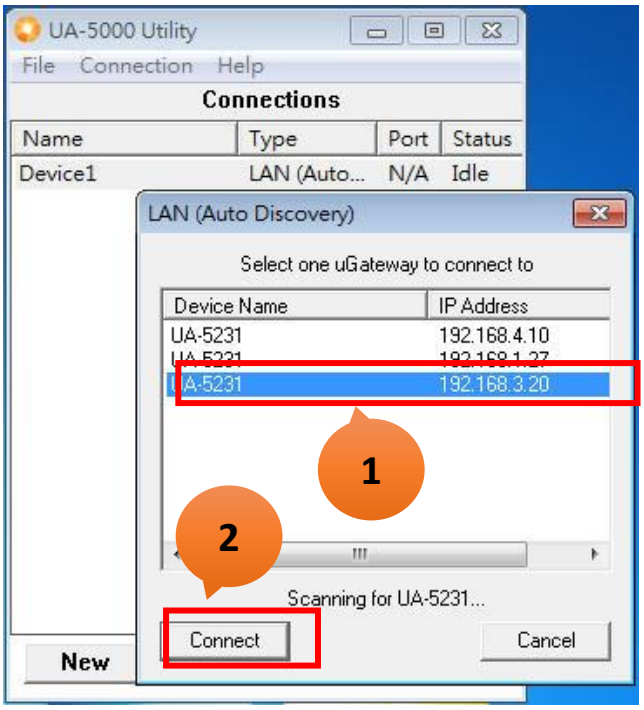
Step 3

Mouse double-click on the name you created (or single-click and then click the “Connect” button), this utility will scan and list all UA-5200 devices over the network.



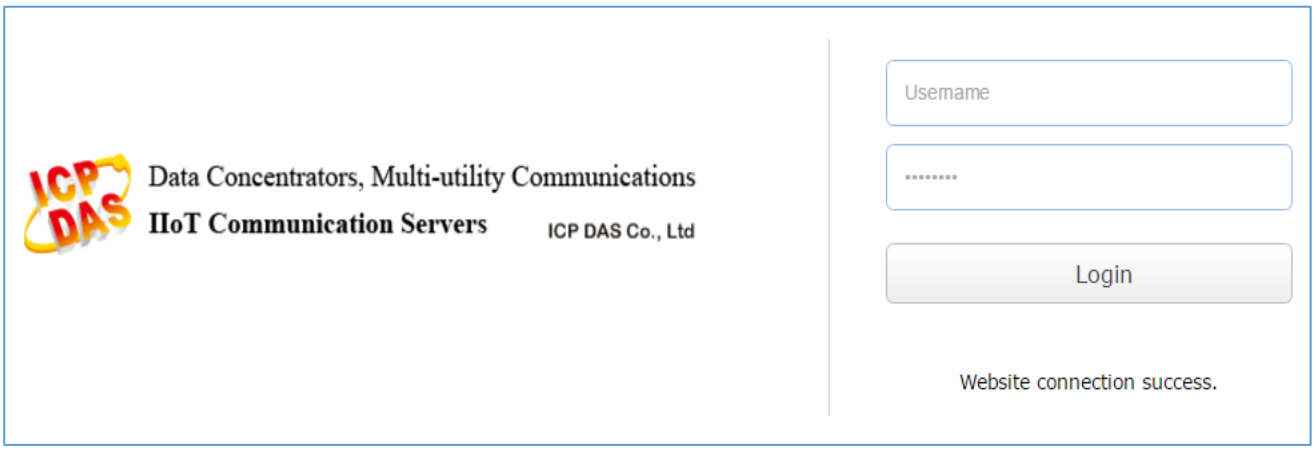
Step 4

Click the device name you want to connect to, and then click the “Connect” button. It will connect to the UA-5200 webpage via the default Web browser (IE/Chrome…).



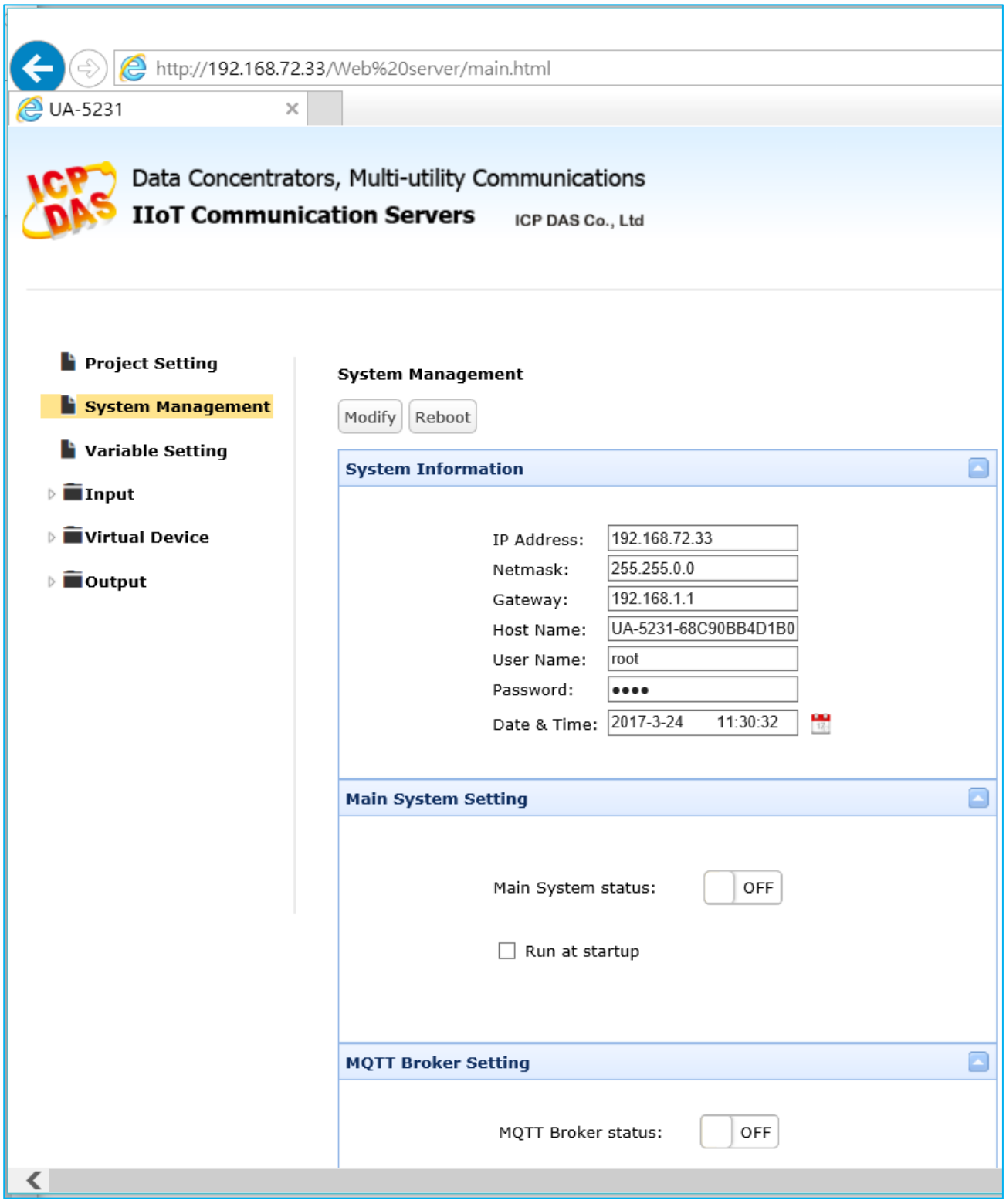
Step 5

A login dialog box will appear, entering your user name and password, and then click “OK”.
The factory default user name and password are “root”.



Finish

When login into the web interface, the UA-5200 default home page (the main configuration screen) will be displayed as below, and will automatically read setting of that UA-5200 to the webpage.



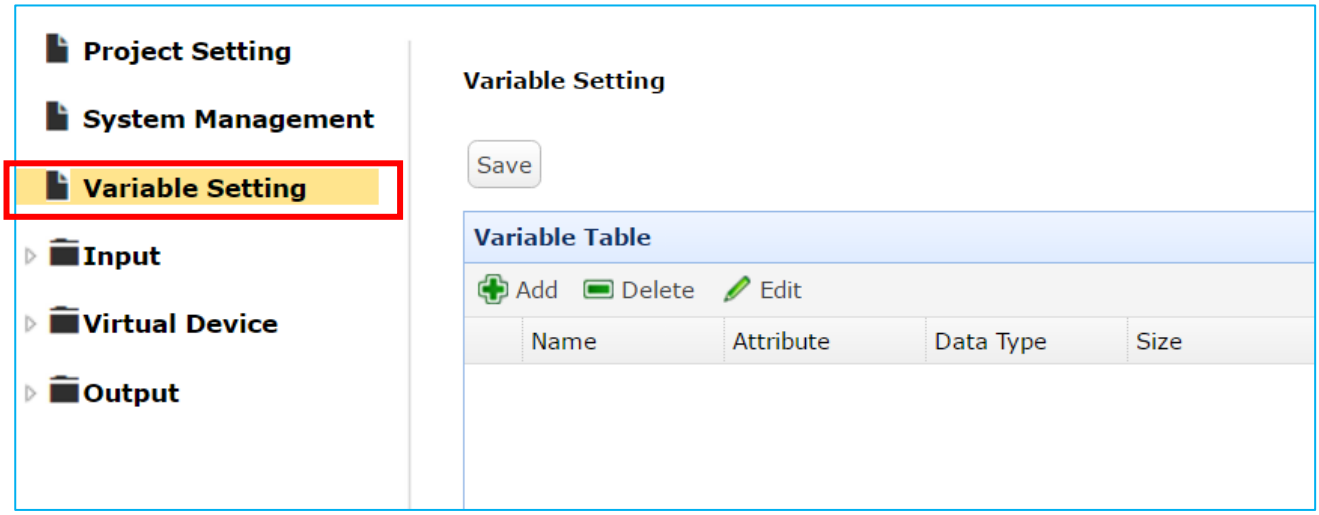
2.2. Add Variables in the Variable Table

After login, the user can set up the project by clicking the main menu on the left side of the web interface. (Refer to 3. System Function for more.)

First, set up the variables. The following steps will show how to add variables in the variable table.

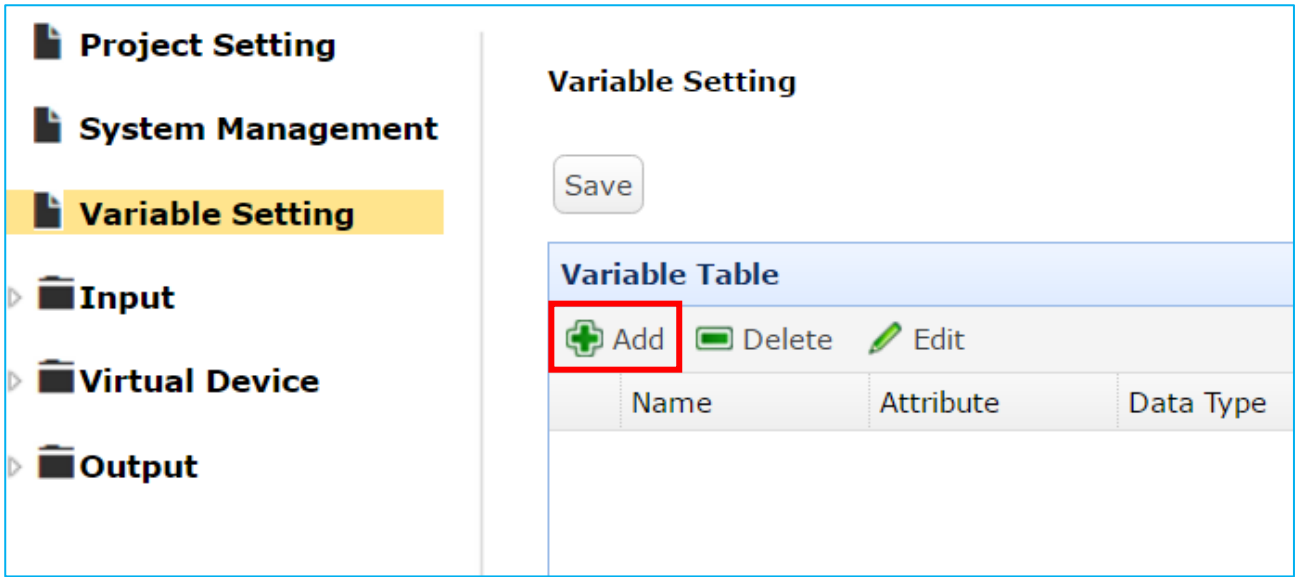
Step 1

Click “Variable Setting” on the left main menu to open the Variable Setting page.



Step 2

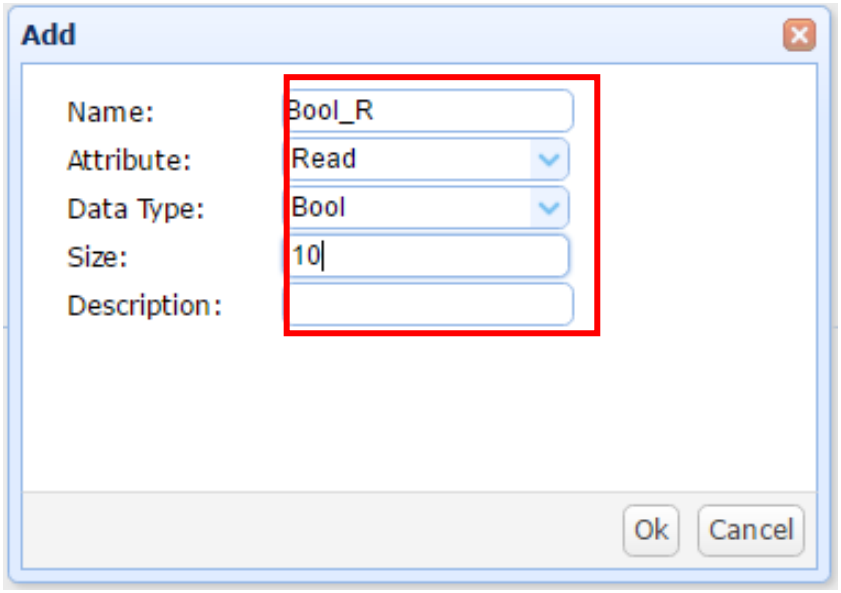
Click “Add” to add the needed variable.



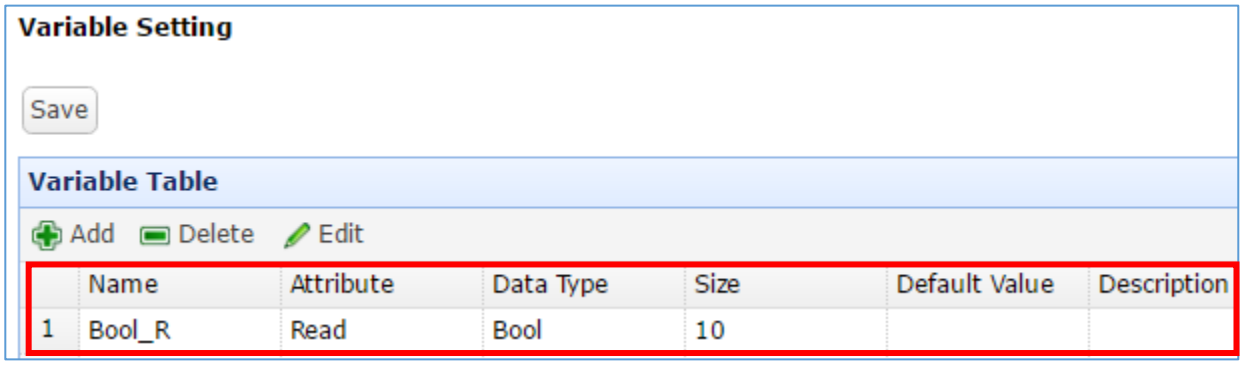
Step 3

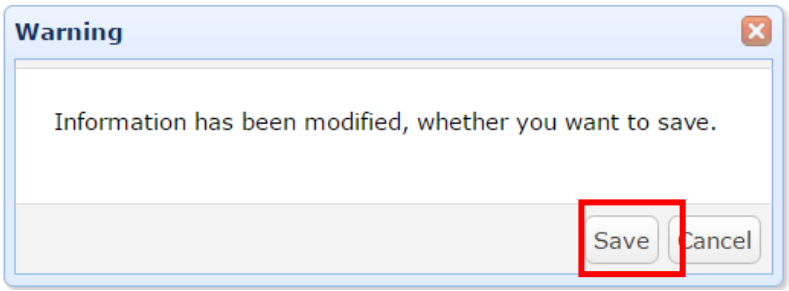
Enter all related information for this variable and then click “OK”.

Ex: set the Name of the variable as "Bool_R", the Attribute as "Read", the Data Type as "Bool", the Size as "10", etc.



Now your variable is set up as below. Press “Save” when system pops up a “Warning” box.



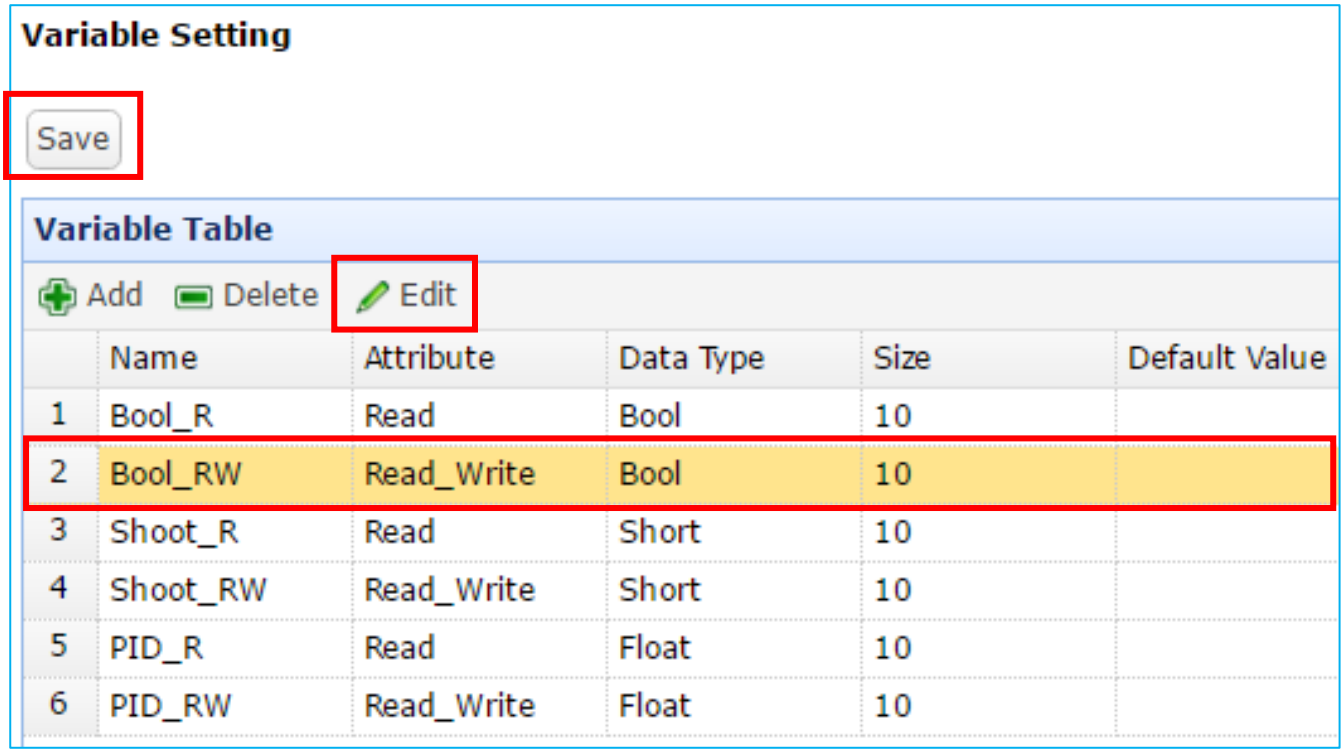


Step 4

Repeat the previous steps to add more wanted variables.

When need to change the variable, click that variable and then click “Edit” to change it.

At last, click the “Save” button to save all variable settings.



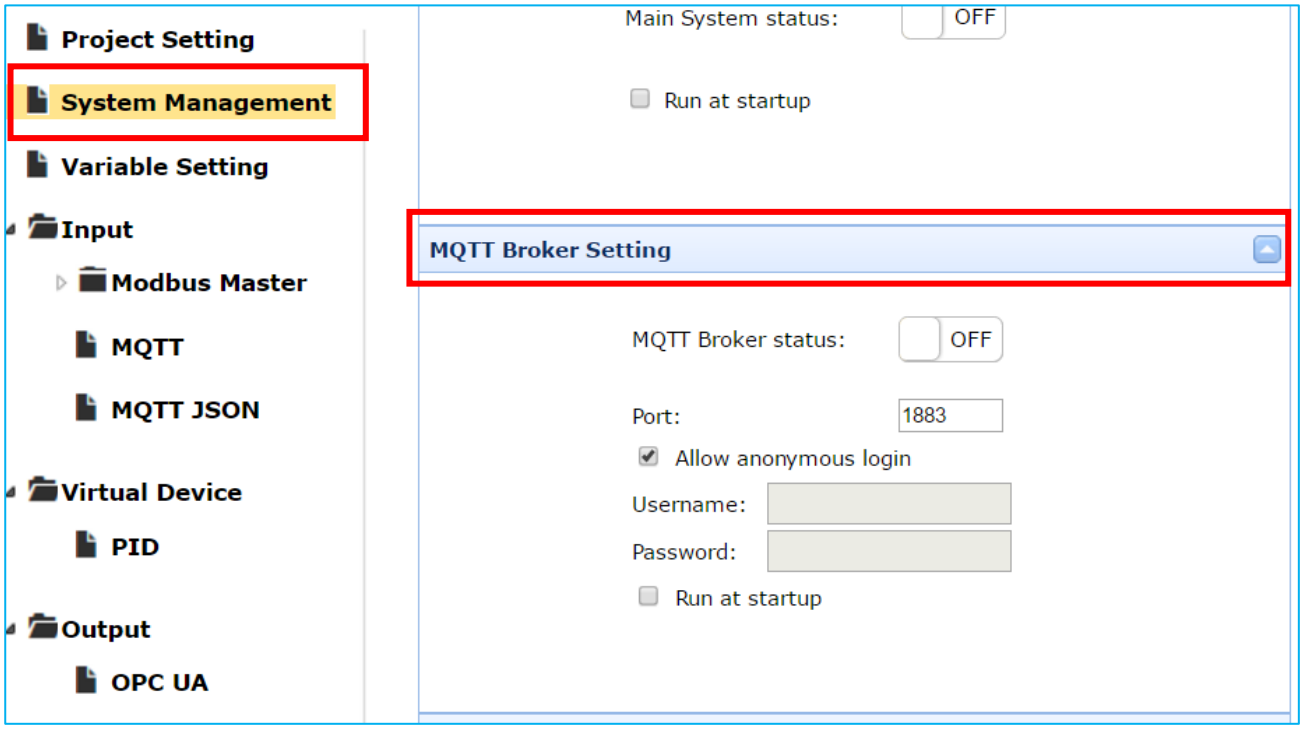
	Name	Attribute	Data Type	Size	Default Value
1	Bool_R	Read	Bool	10	
2	Bool_RW	Read_Write	Bool	10	
3	Shoot_R	Read	Short	10	
4	Shoot_RW	Read_Write	Short	10	
5	PID_R	Read	Float	10	
6	PID_RW	Read_Write	Float	10	

2.3. Start a Built-in MQTT Broker

If the MQTT and MQTT JSON communication of the project use the built-in MQTT Broker, you need to enable it before setting communication. The enabling function is in the “System Management”.

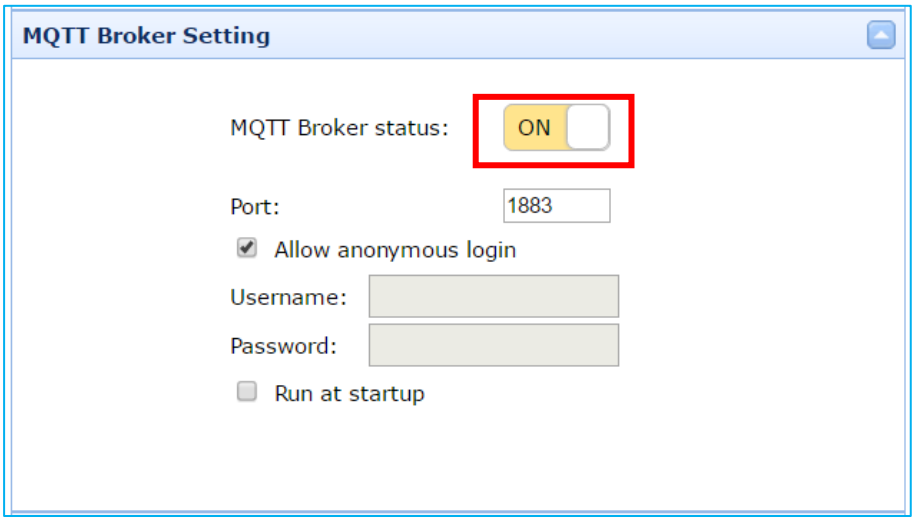
Step 1

Click “System Management” on the left tree-menu, and then scroll down the right webpage to the “MQTT Broker Setting” box that is the settings for the built-in MQTT Broker.



Step 2

Click the switch of the “MQTT Broker status” to enable the Broker. When it shows “ON” that means the Broker is running. According to user’s needs, the user can set up other setting items.



2.4. Set Up the Input, Virtual Device and Output

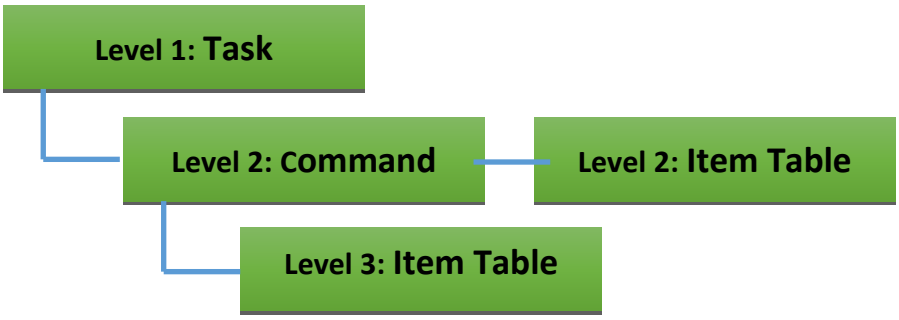
The UA-5200 is a Communication Server that down to interact with the I/O modules (Modbus/MQTT) and up to interact with the host (OPC UA/MQTT). The "**Input**" setting of tree-menu is mainly related to the communication with the down side I/O modules. The "**Output**" setting of tree-menu is mainly related to the communication with the up side host. The input data via the UA-5200 converting and programming will be sent to the up side host for the application platform/software/system/cloud /equipment/database that support client-end OPC UA/MQTT communication.

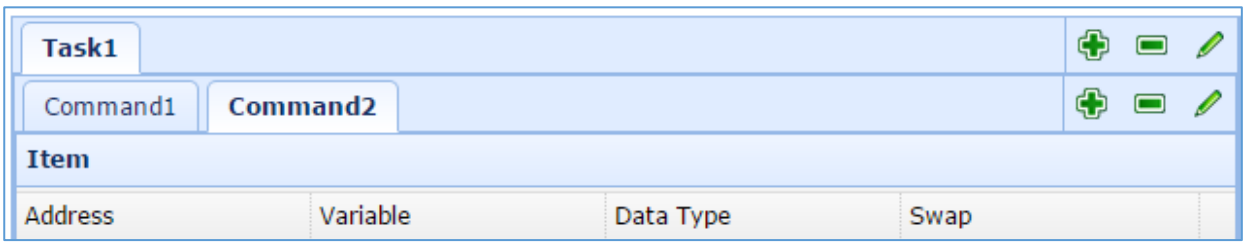
This section will describe the Input, Virtual Device, and Output settings. The user can also refer the Chapter 3: System Function Description to view all features, properties and configuration notices.

Section 3.4 Input
Section 3.5 Virture Device
Section 3.6 Output

Description of the Setting Steps:

All function settings follow the task-oriented and level logic architecture. First, the user can add a connection **Task** depends on device properties, and then add **Command** or **Item** table under the Task. Finally, you can complete two-level or three-level settings.





Description of the tool button:

: Add a task, command, or item.

: Delete a task, command, or item.

: Modify a task, command, or item.

2.4.1. Set up the Input

The **Input** setting is divided into the **Modbus Master**, **MQTT** and **MQTT JSON**. It is mainly for the UA-5200 to communicating with the I/O modules. Here will show how to set up Modbus and MQTT in the following 2 sections.

A. Modbus Master Setting Example

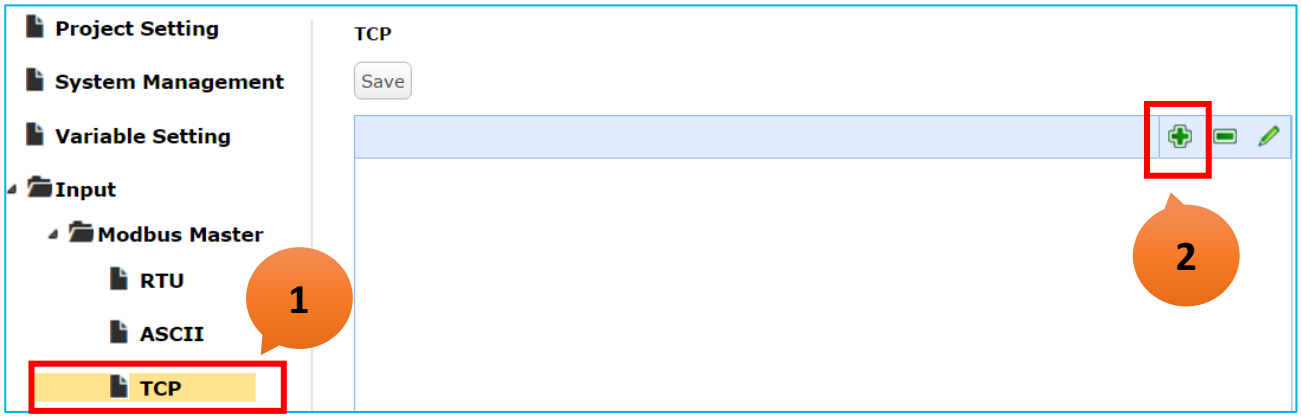
It provides Modbus RTU/ASCII/TCP 3 protocols. We use the **Modbus TCP** to conduct the setting example. The user can also refer the Chapter 3 - System Function Description - Section 3.4 Input to view all features, properties and configuration notice for each item mentioned in this section.

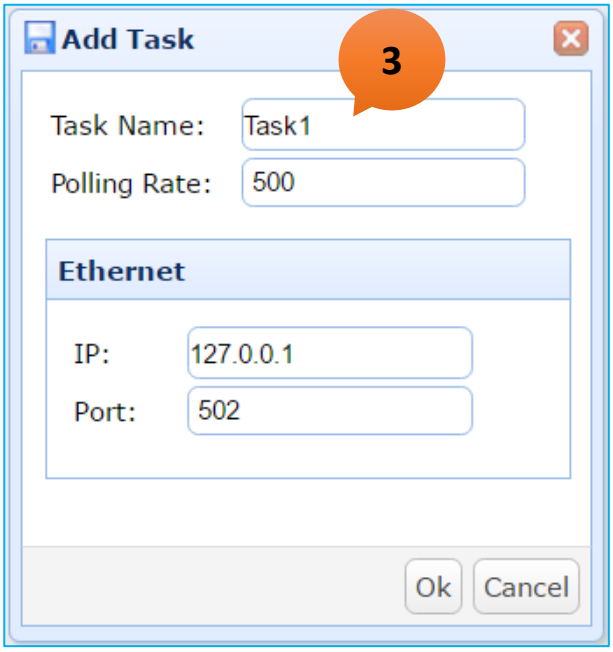
Step 1

Add all needed variables in the variable table. (See Section 2.2)

Step 2

On the left tree menu, click “Input -> Modbus Master -> TCP”, and add a connection **Task**.



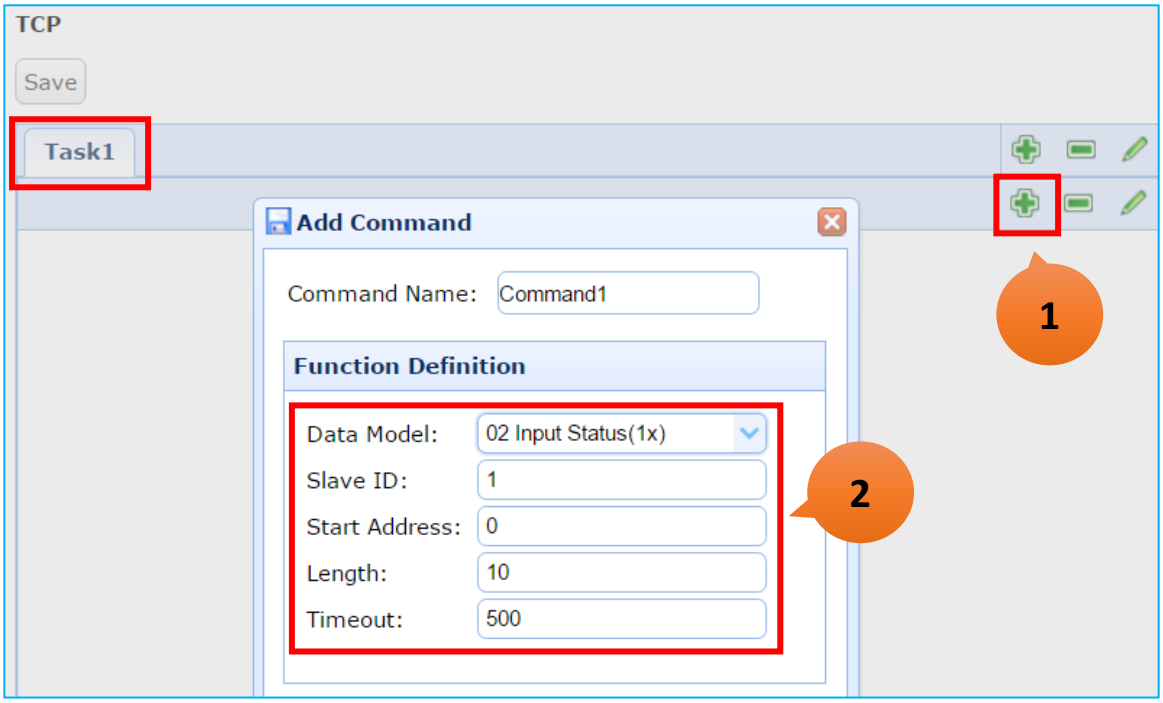


Set the TCP Task:

* Task Name: give the Task a name
* Polling Rate: set the interval time for each command
* IP: The IP address of the connected device
* Port: The port number for Modbus TCP

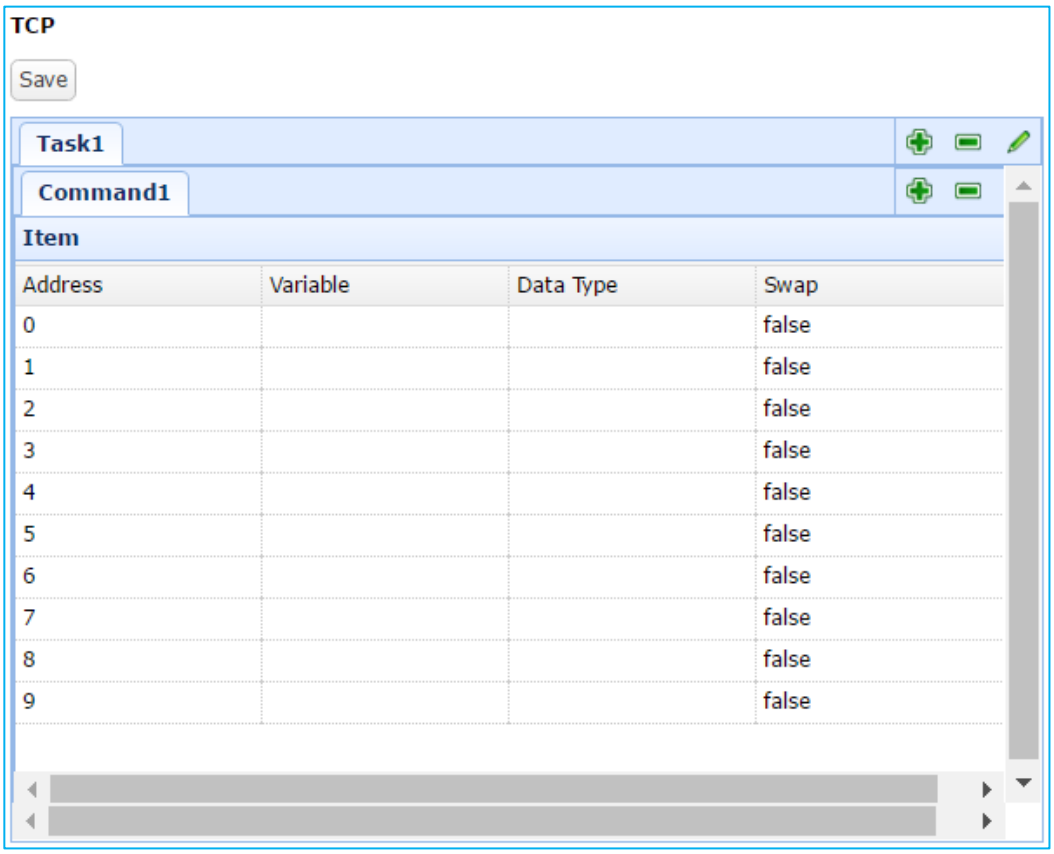
Step 3

In the Task table, add a Modbus command and enter all parameters.



Step 4

After completing it, you can see this two-level (Task -> Command) table.



Address	Variable	Data Type	Swap
0			false
1			false
2			false
3			false
4			false
5			false
6			false
7			false
8			false
9			false

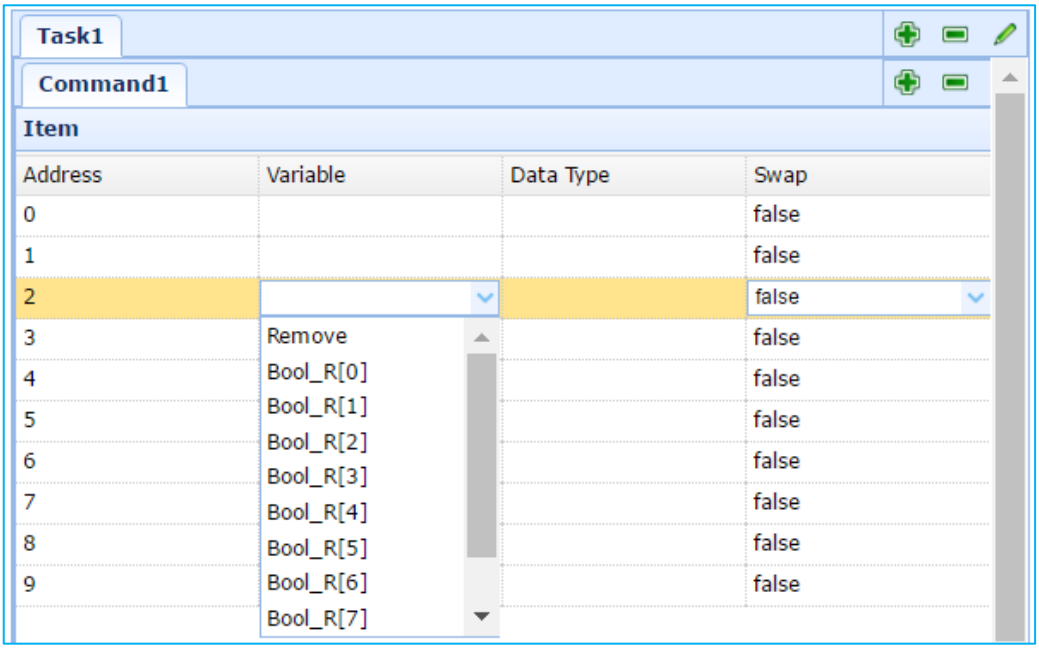
Step 5

Then, choose the proper variables for the Modbus Address in the **Item** table.

EX: Double click the Variable cell of Address 2, and assign the variable “Bool_R[0]”.

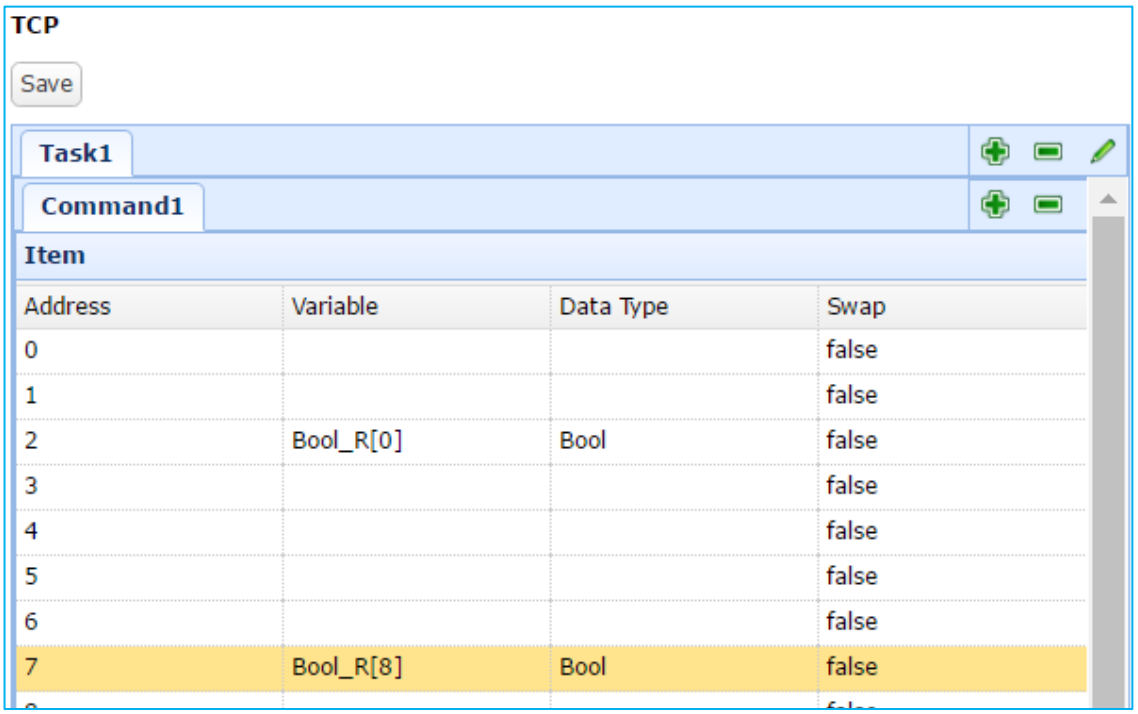
Double click the Variable cell of Address 7, and assign the variable “Bool_R[8]”.

If want to delete or change a setting, just double click and select “remove” or other variable.



Step 6

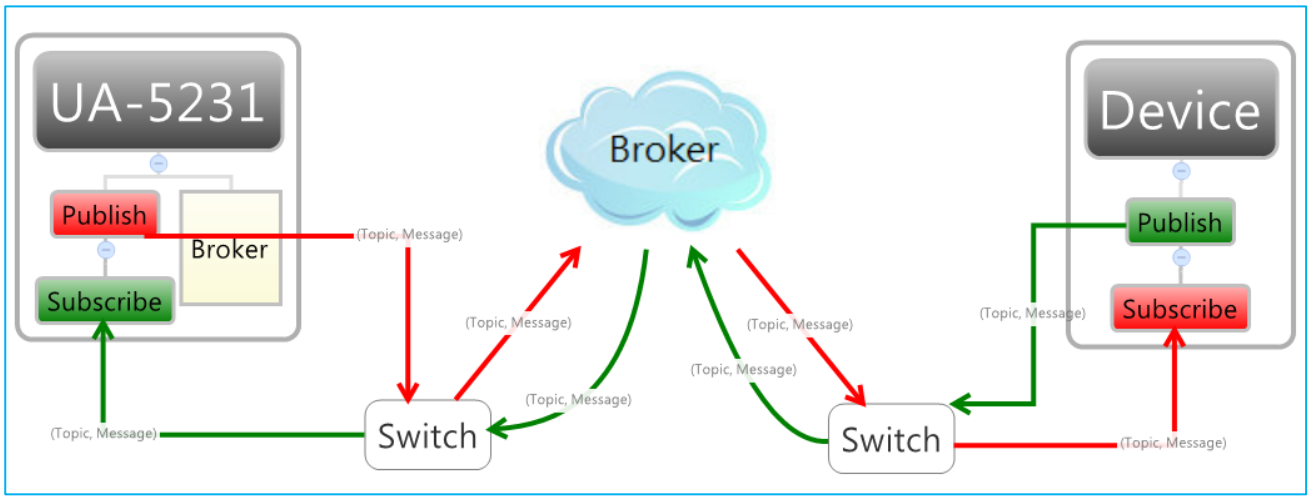
Repeat the previous steps to set up this table. At last, click “Save” for all variable settings.



B. MQTT Setting Example

Here we will describe how to set the MQTT Input of UA-5200 (i.e., UA-5231) to communicate with the MQTT device. The user can refer the Chapter 3. System Function Description - Section 3.4 Input to view all features, properties and configuration notice for each item mentioned in this section.

For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.



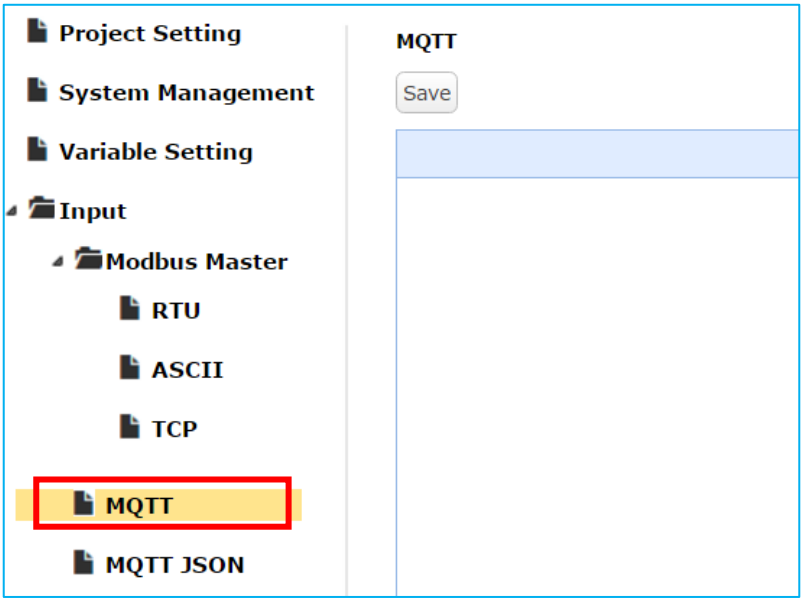
Step 1

Add all needed variables in the variable table. (See Section 2.2)

If the MQTT communication uses the built-in MQTT broker, enable the broker first. (See Section 2.2)

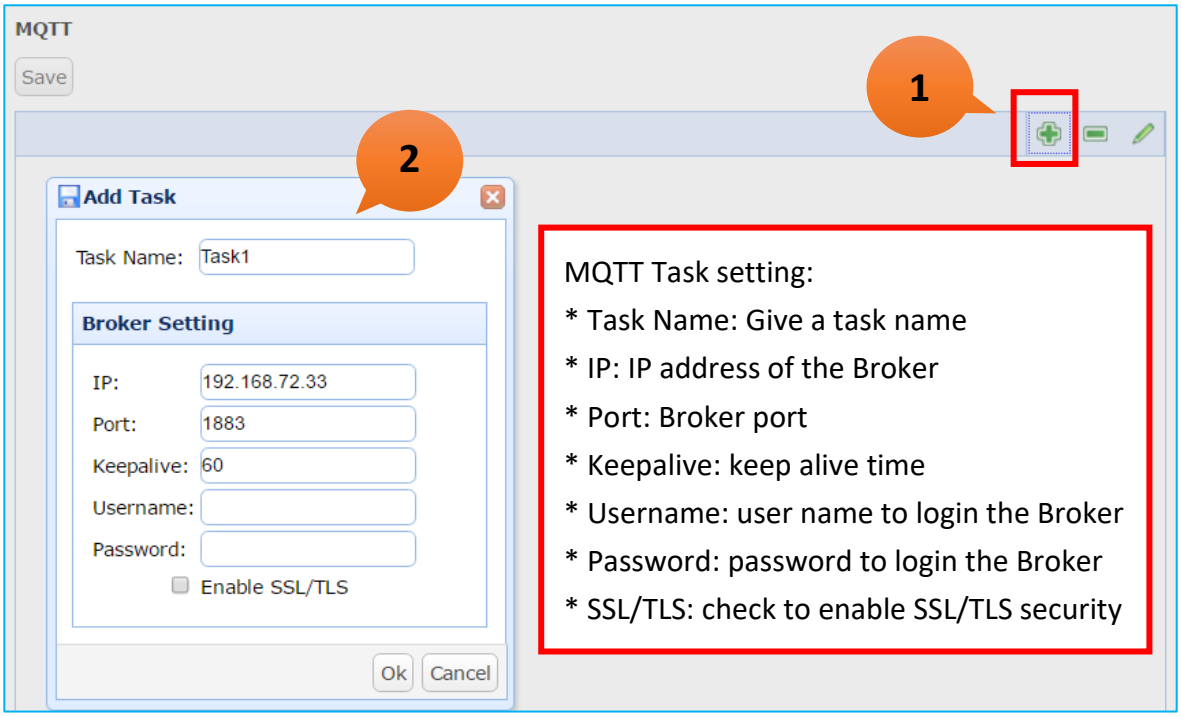
Step 2

Click on “Input” -> “MQTT” of the left tree-menu to open the MQTT Task setting box.



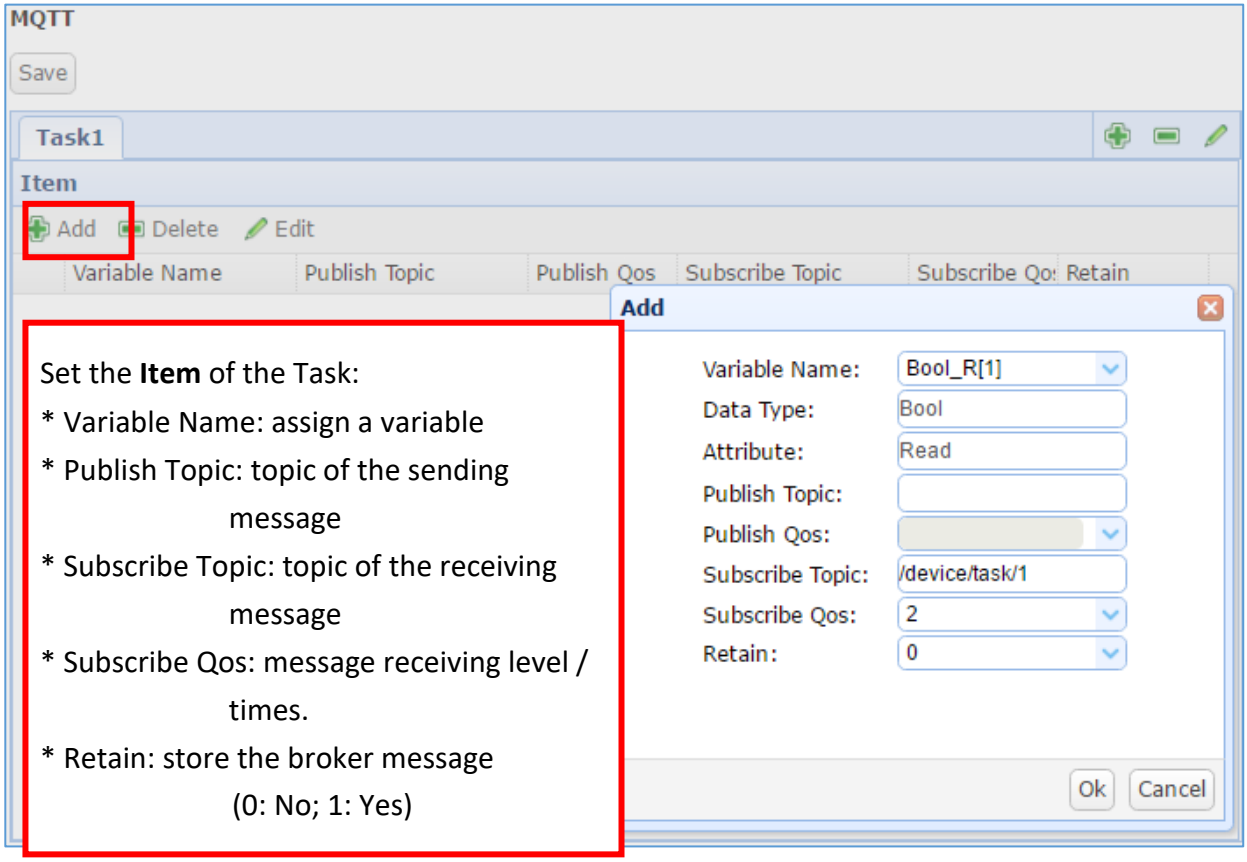
Step 3

Add a MQTT Task. Set Task name and Broker information according to your project.



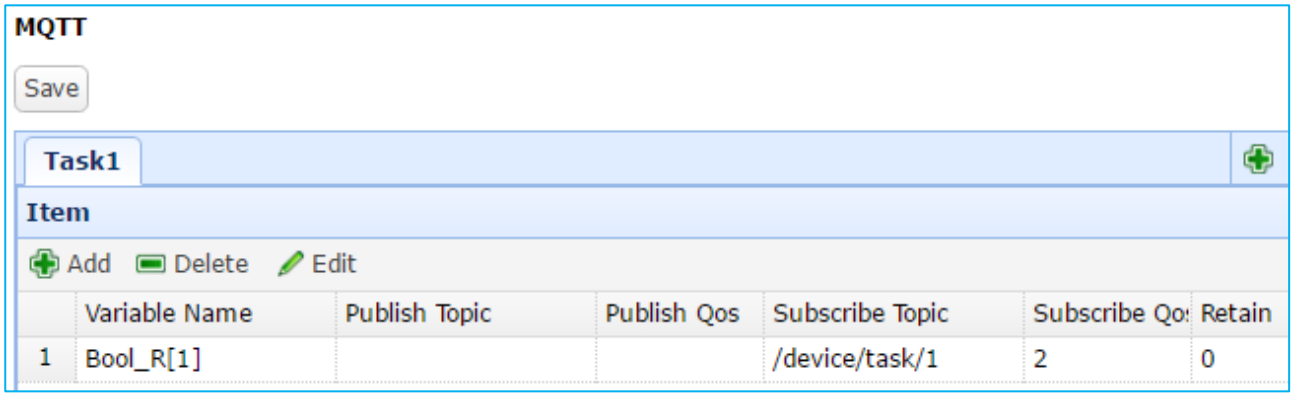
Step 4

In the **Item** table, click “Add” to assign a variable and edit its topic related settings.



Step 5

Enter the needed parameters and click “OK”, it will show the Item table as below.

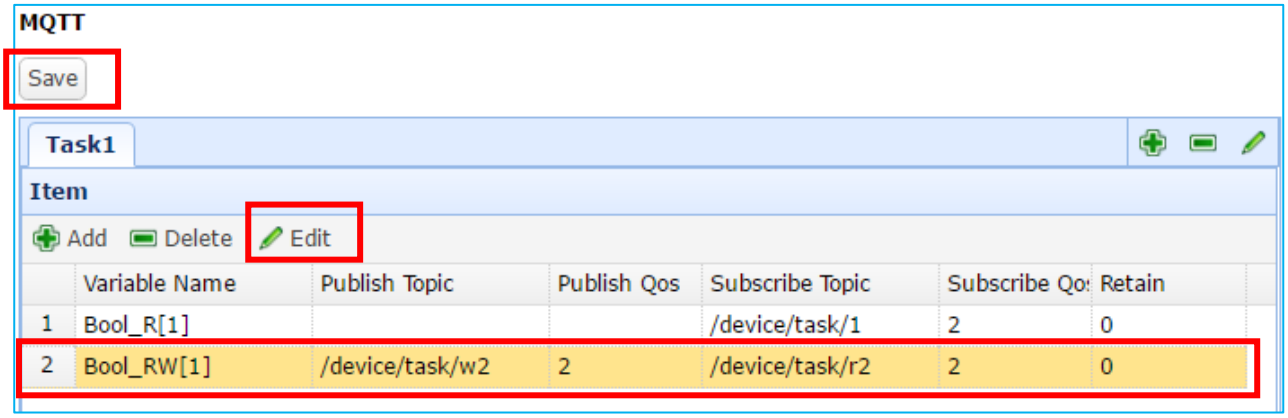


Step 6

Repeat the previous steps to set up more items to the table.

If need to change the setting, click that item (color changed) and then click “Edit” to change data.

At last, click “Save” for all items settings.



2.4.2. Set Up the Virtual Device

This Virtual Device function allows the user to simulate various devices with the real I/O. This function includes the PID function. Proportional-Integral-Derivative control is the most widely used in industrial control systems. A regulator which is controlled in accordance with Proportional, Integral and Derivative is called PID control for short, also called PID regulator. When the user cannot fully grasp or measure parameters of the control system, the PID regulator is the best solution.

A. PID Setting Example

In this section, we will describe the PID settings.
The user can also refer the Chapter 3. System Function Description - Section 3.5 Virture Device to view all features, properties and configuration notice for each item mentioned in this section.

Step 1

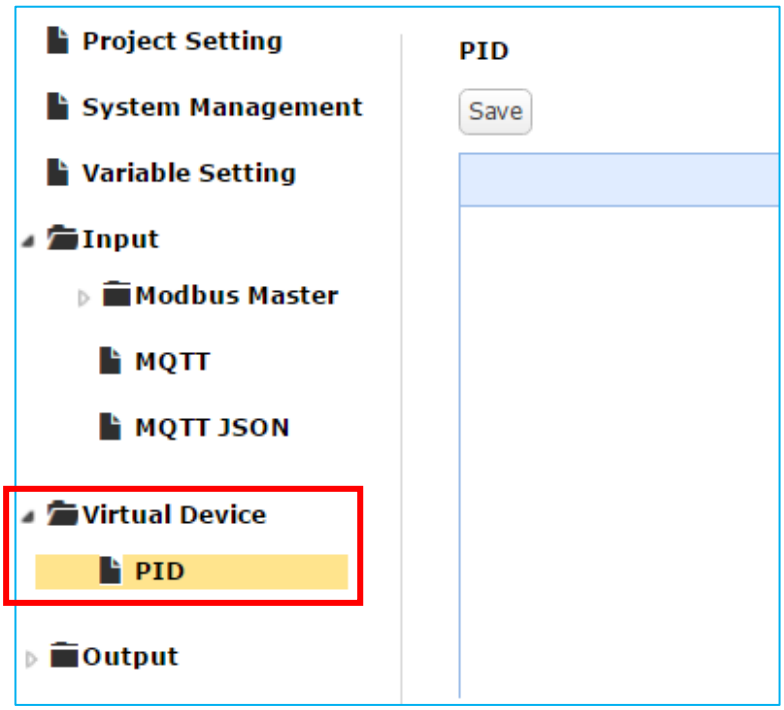
Add all needed variables in the variable table. (See Section 2.2)
Note: The inputs/outputs type of PID must be floating point, please set variables as “Float” type.

Step 2

Set up the Input for the corresponding physical I/O. (See Section 2.4.1)

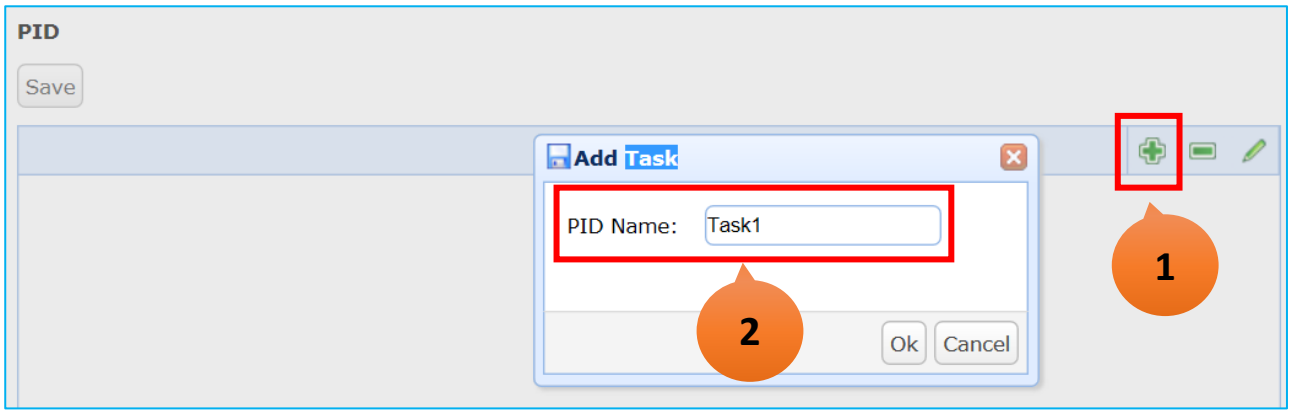
Step 3

Click on “Virtual Device” -> “PID” to set up the PID.



Step 4

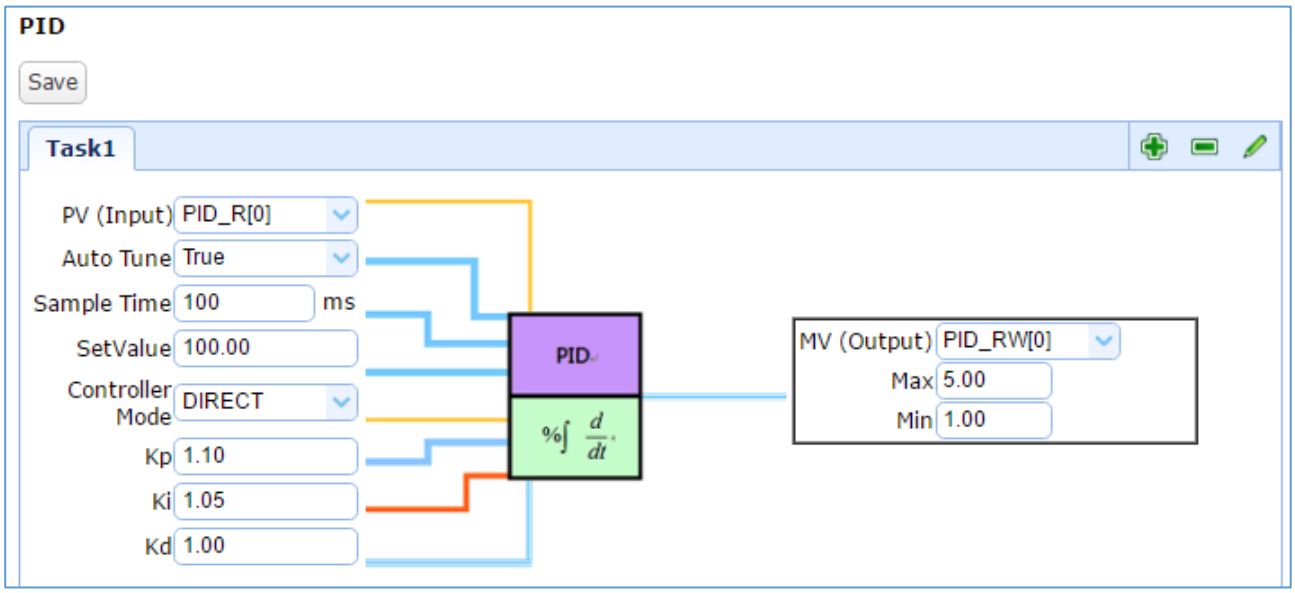
Add one PID **Task**.



Step 5

Set up the PID parameters. Assign an input variable to PV, an output variable to MV, and set other parameters according to the project needs. The user can set the “Auto Tune” as “true” for automatic PID regulator, but still has to enter some important parameters such as “Sample Time”, “SetValue”… (See Section 3.5.1 PID for details) .

At last, click “Save” for all settings.

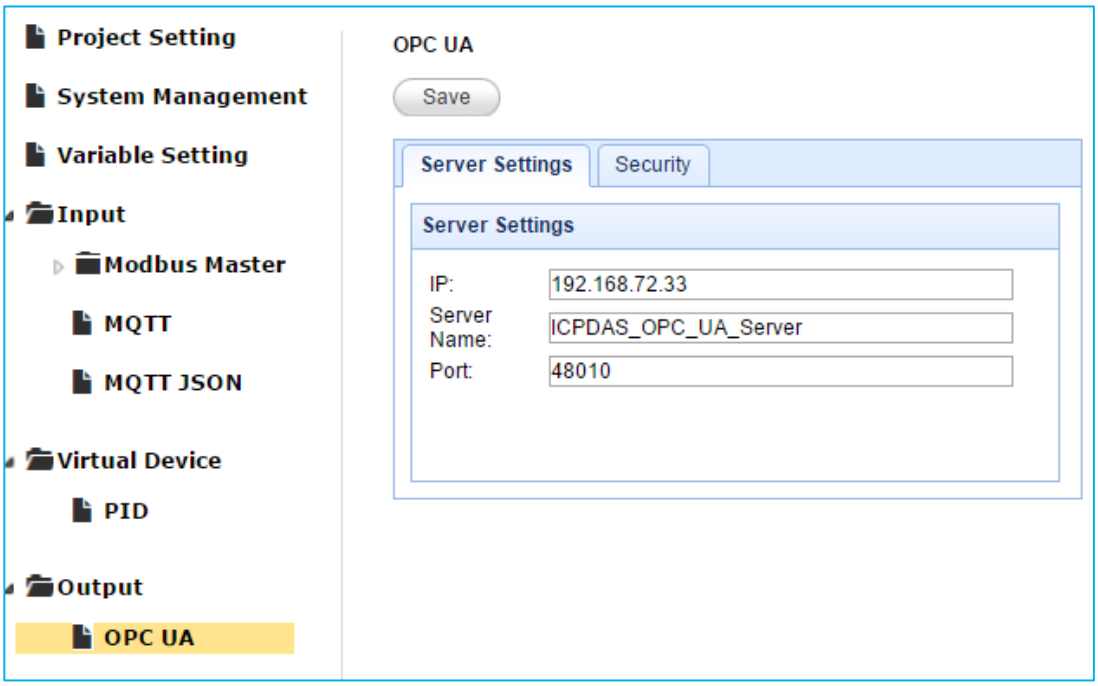


2.4.3. Set Up the Output

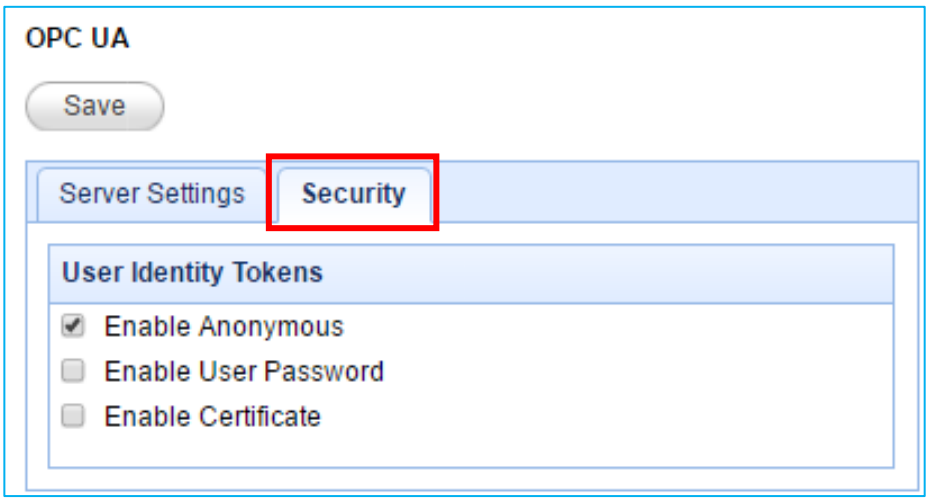
The **Output** setting includes the OPC UA, MQTT and MQTT JSON. It is mainly for the UA-5200 to communicating with the up side host. Here will show how to set up Modbus and MQTT in the following 2 sections.

A. OPC UA Output Setting Example

The OPC UA Server is system enabled by default. When the user assigns variables in the Input and Virtual Device, the configuration for the OPC UA Server will also be done.
Click on “Output” -> “OPC UA” to see the settings.

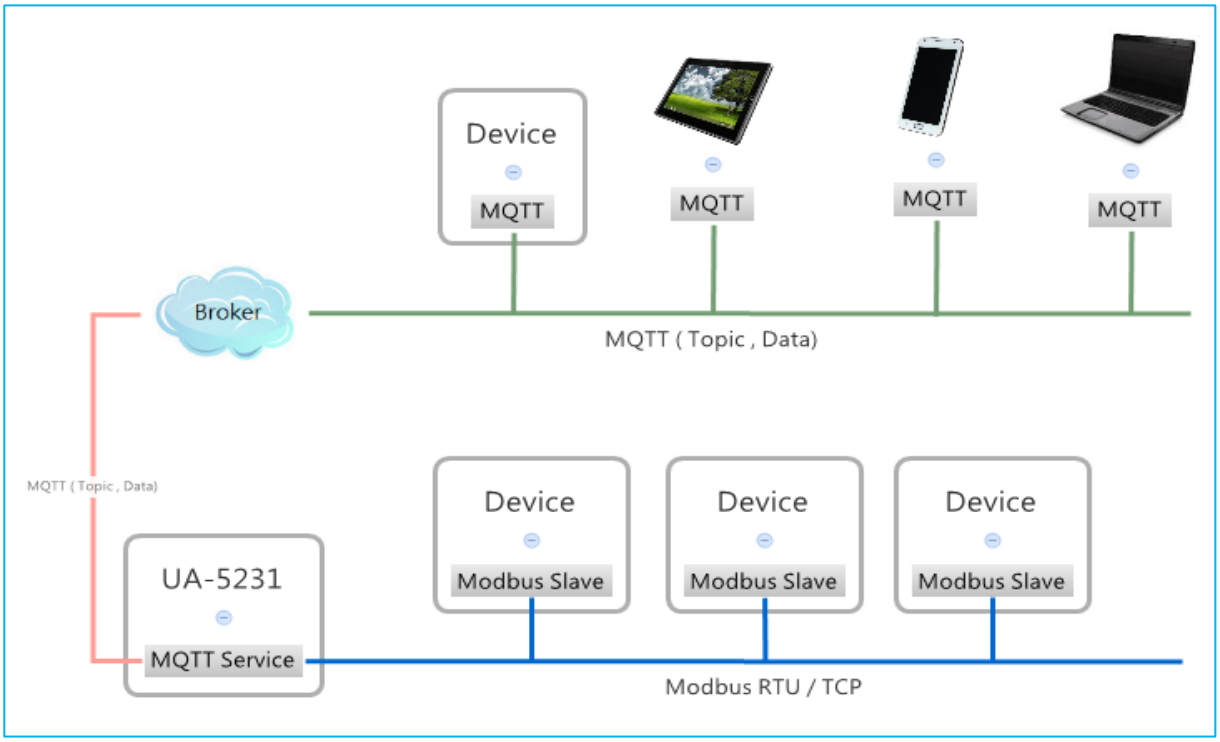


Click on the “Security” page can enable the anonymous, password or certificate login.
At last, click “Save” for all settings.
For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.)



B. MQTT Output Setting Example

The UA-5200's MQTT Output is used to convert other Input's data into the MQTT message, using a user-defined topic as an index to receive the data sending from other MQTT devices.



The following steps will show you the way to convert the Modbus Master Input's data into the MQTT message.

The user can also refer the Chapter 3. System Function Description - Section 3.6 Output to view all features, properties and configuration notice for each item mentioned in this section.

Step 1

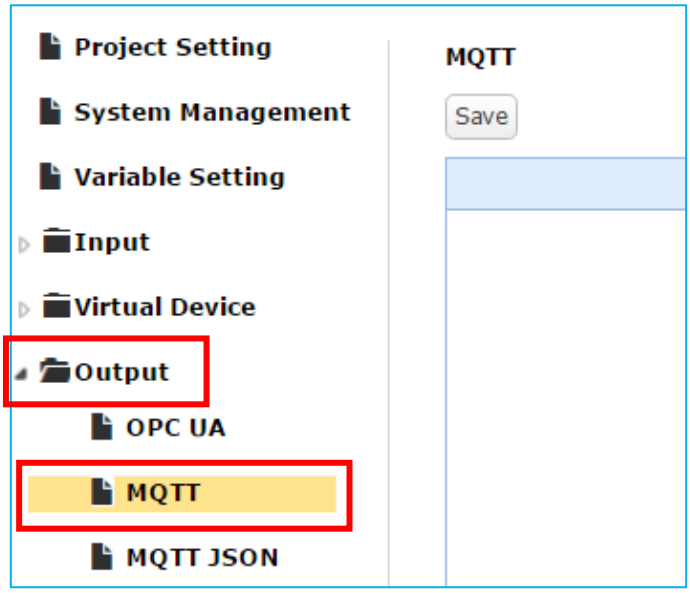
Add all needed variables in the variable table. (See Section 2.2)

Step 2

Set up the Modbus Input, and assign variables to the Input. (See Section 2.4.1)

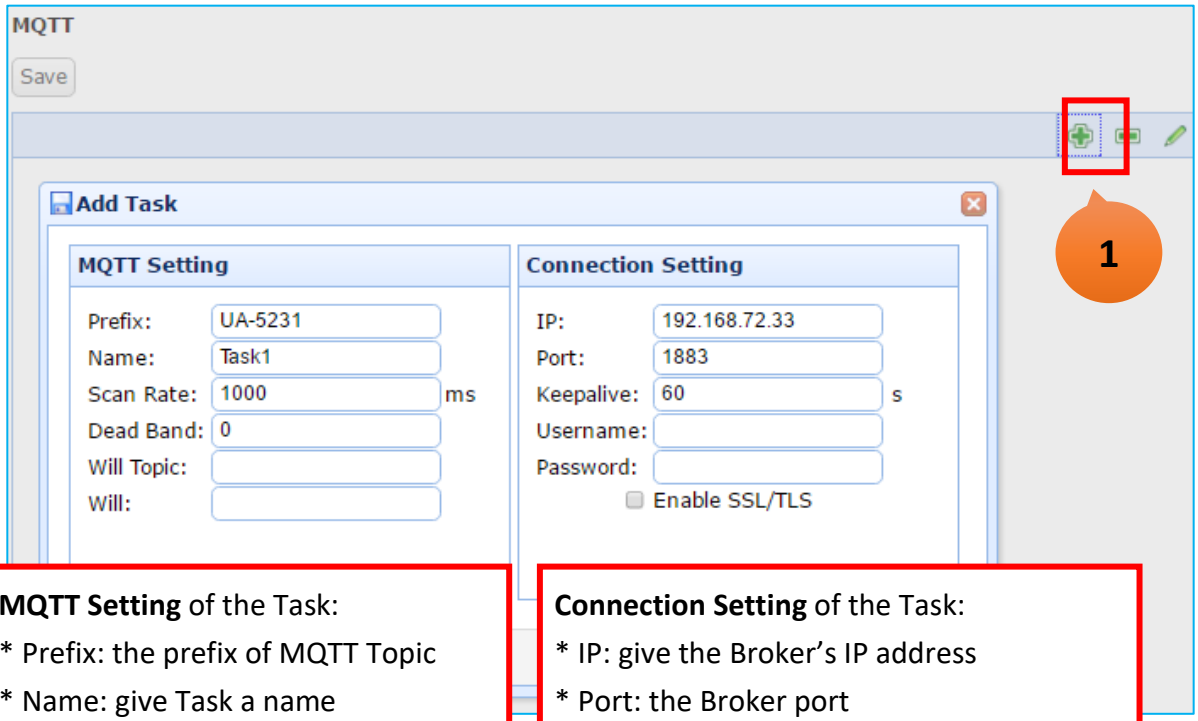
Step 3

Set up the MQTT Output. Click “Output” -> “MQTT” on the left tree-menu.



Step 4

Add a **Task** for this MQTT Output as below. Detail descriptions see Section 3.6.2 MQTT .



MQTT Setting of the Task:

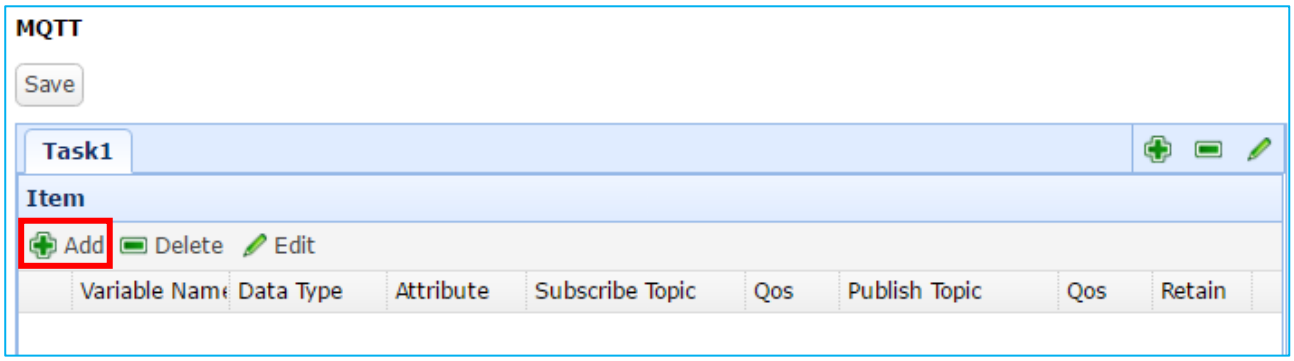
* Prefix: the prefix of MQTT Topic
* Name: give Task a name
* Scan Rate: Task update frequency
* Dead Bend: Give a dead bend value for updating a float signal.
* Will Topic: title of the disconnect notice
* Will: the disconnect notice

Connection Setting of the Task:

* IP: give the Broker’s IP address
* Port: the Broker port
* Keep alive: Set a time to check if the connection to the Broker is working
* Username: user name to login the Broker
* Password: password to login the Broker
* Enable SSL/TLS: check to enable SSL/TLS security

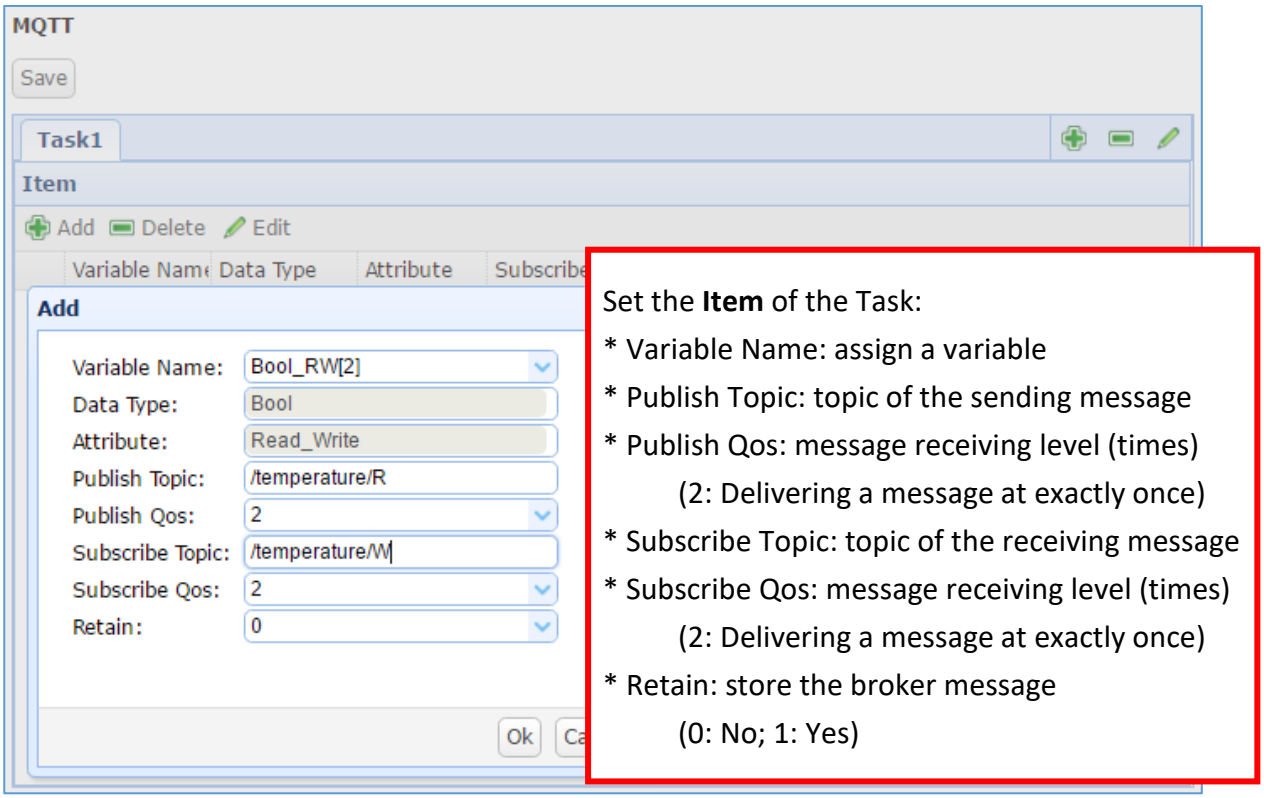
Step 5

Click the “Add” button under the “Task” to add a Topic item.



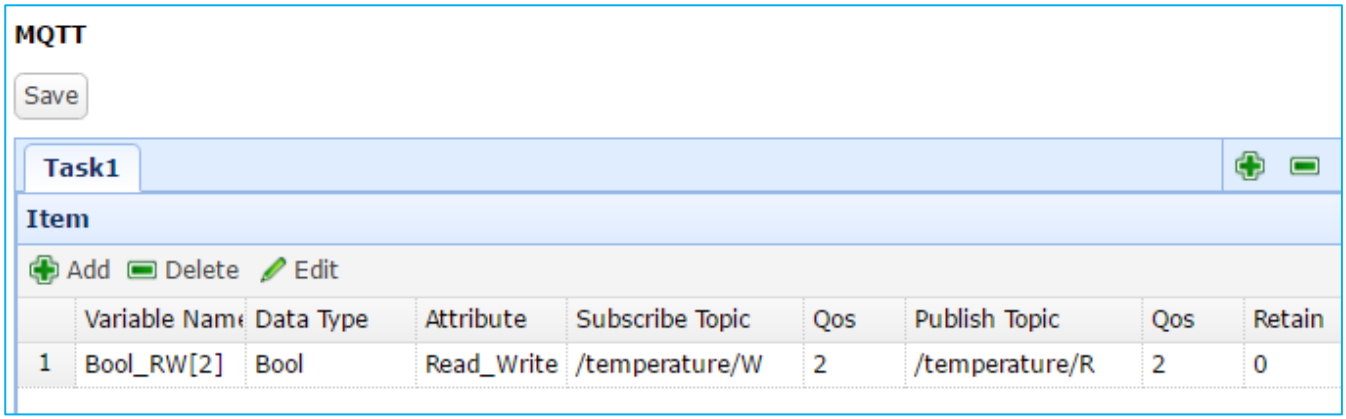
Step 6

Enter the needed variable and parameters as below. Detail descriptions see Section 3.6.2 MQTT.



Step 7

Now, you have successfully added this item.

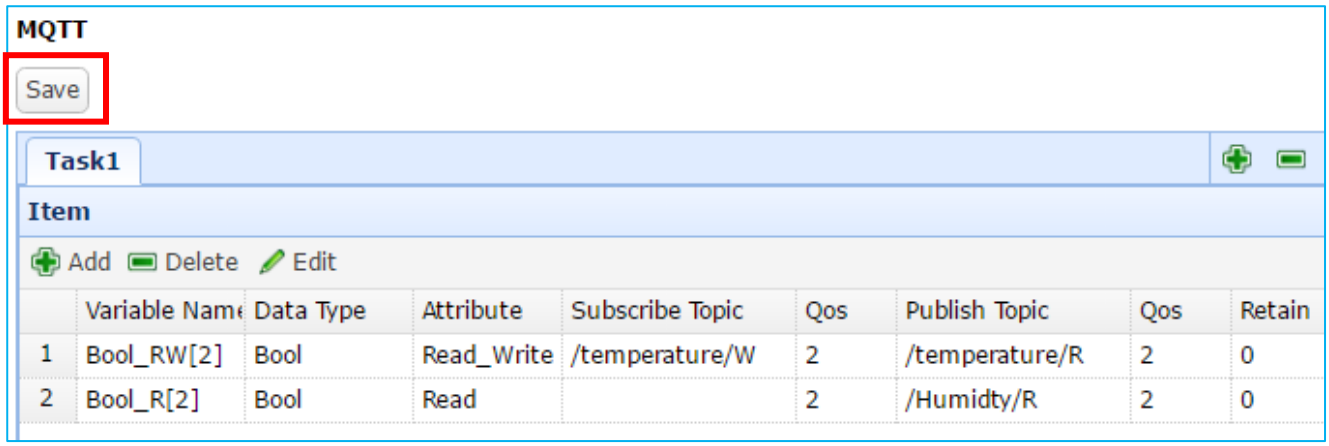


	Variable Name	Data Type	Attribute	Subscribe Topic	Qos	Publish Topic	Qos	Retain
1	Bool_RW[2]	Bool	Read_Write	/temperature/W	2	/temperature/R	2	0

Step 8

Repeat the previous steps to add several items.

At last, click “Save” to save all settings.

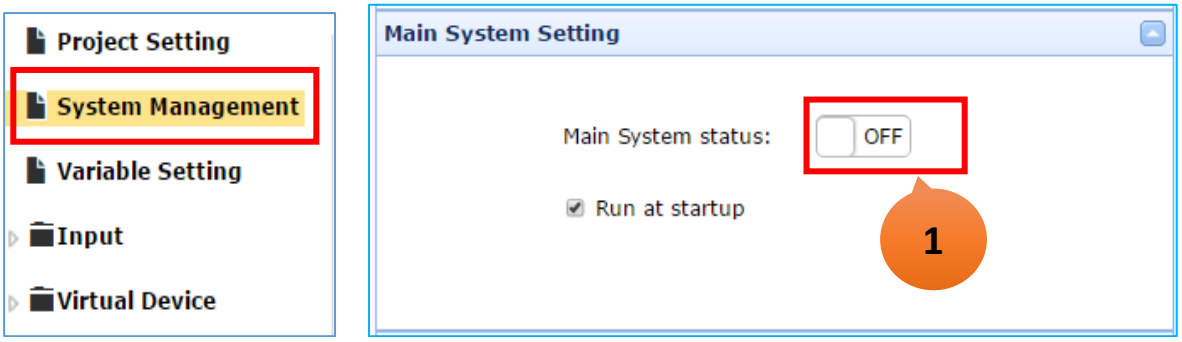


	Variable Name	Data Type	Attribute	Subscribe Topic	Qos	Publish Topic	Qos	Retain
1	Bool_RW[2]	Bool	Read_Write	/temperature/W	2	/temperature/R	2	0
2	Bool_R[2]	Bool	Read		2	/Humidty/R	2	0

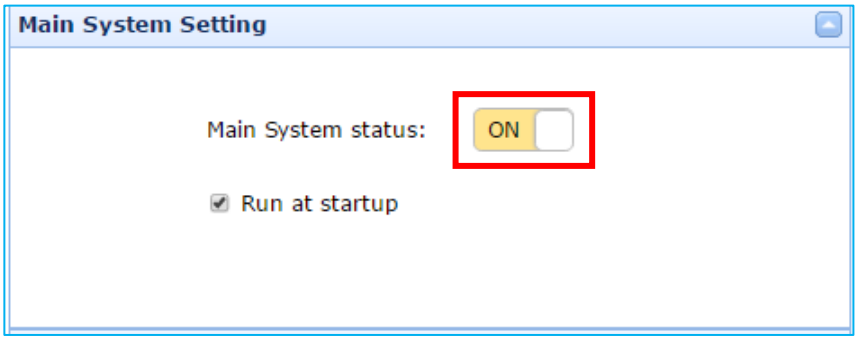
2.5. Run the Main System and Project

This section will describe how to start the main system and the project of the UA-5200 series.

When the user finish the project setting and want to start the system and project, simply click on the “System Management” on the left tree-menu, scroll down the webpage to the “Main System Setting” box, and then click the “status” switch.



While “ON” is displayed, it means the new project is running in the UA-5200 system.



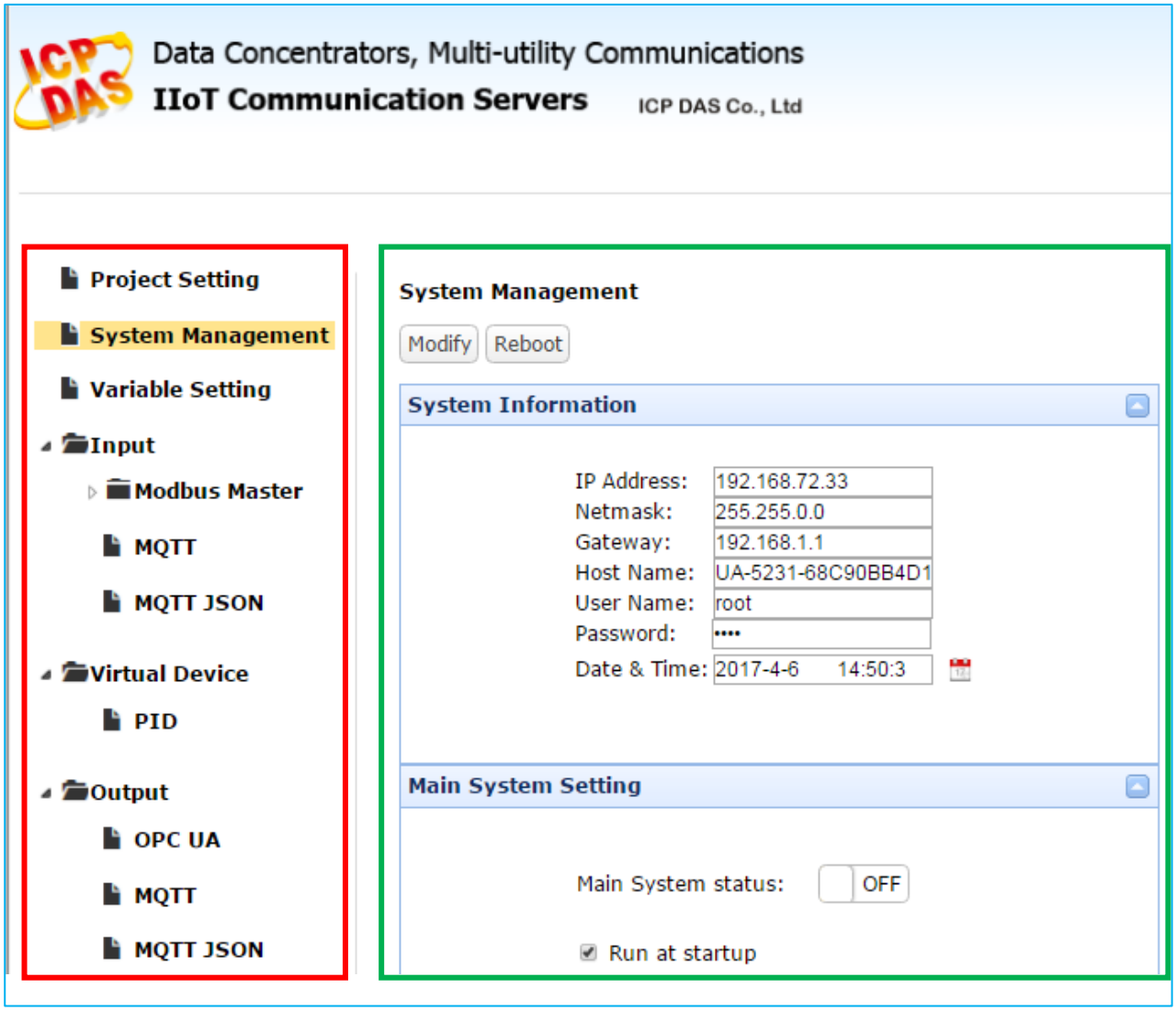
3. System Function Description

In this chapter, we will describe all functions and parameters on the following topics that listed in the UA-5200's Web UI (as the figure below).

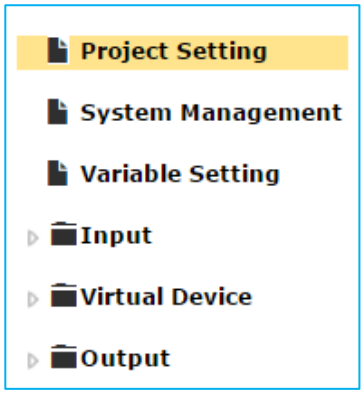
System function includes two parts:

Function Tree-Menu: On the left webpage (**Red** box). Function menu is in the multi-level tree structure. Click the left button can expand or collapse the lower level functions. This chapter describes each setting of all system functions.

Parameter Setting: On the right webpage (**Green** box). When clicking the function item on the left Tree-Menu, the related setting window will show up to set the parameters. Scroll down the window can set more settings if needs.



3.1. Project Setting

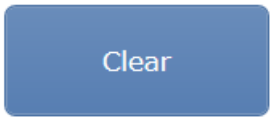


This section describes how to use the “Project Setting” function to delete, export or import a project.

It includes the “Clear”, “Export” and “Import”.

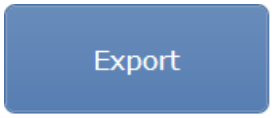
3.1.1. Clear

The “Clear” function can delete all project settings, includes the settings of “Variable”, “Input”, “Virtual Device” and “Output”.



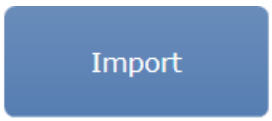
3.1.2. Export

The “Export” function can compress and save the project of the UA-5200 to a “**.tar**” file. The user can select the path, folder and assign the file name. The export items include the “Variable Setting”, “Input”, “Virtual Device” and “Output”.

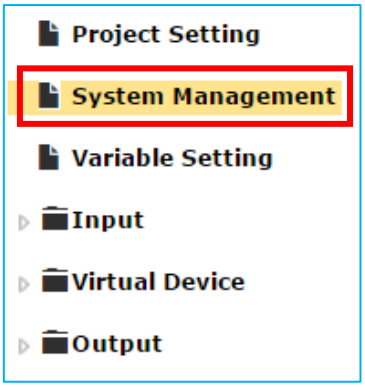


3.1.3. Import

The “Import” function can import the project into the UA-5200. The extension file name must be “**.tar**”. The user can select the file name, folder and path. The import items include the “Variable Setting”, “Input”, “Virtual Device” and “Output”.



3.2. System Management

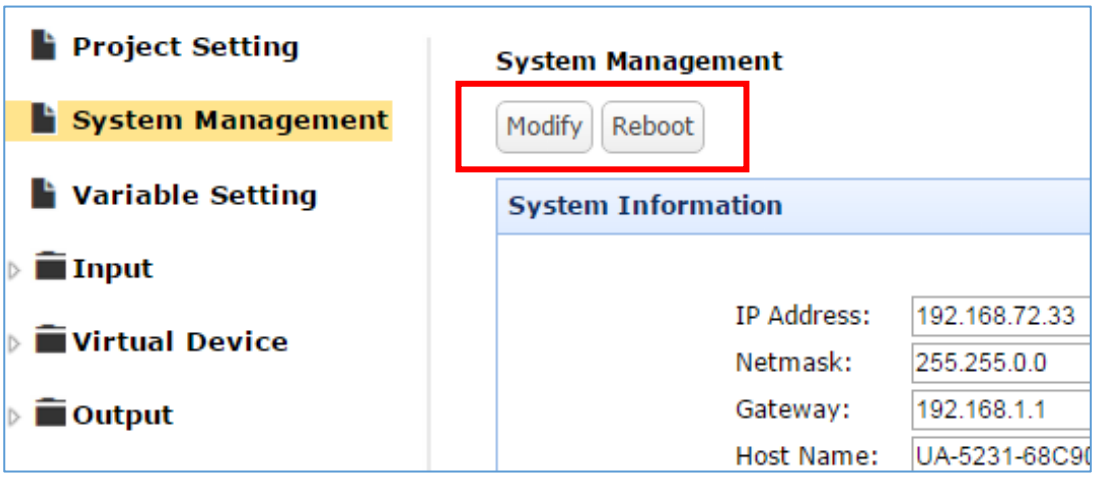


This section describes how to use the “System Management” function to manage system setting of the UA-5200.

It includes the settings:

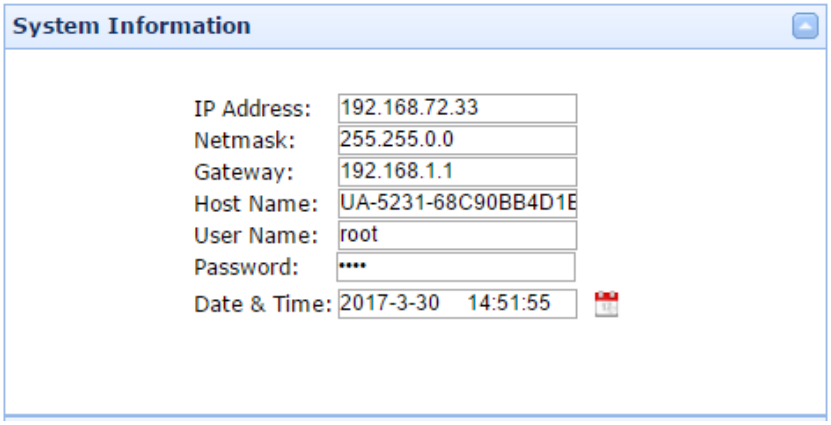
- System Information
- Main System Setting
- MQTT Broker Setting
- Dynamic DNS Setting

After setting or changing, remember to scroll up and click the “Modify” button in the top of the webpage to save the project. Click “Reboot” can restart the Run Time of the UA-5200.



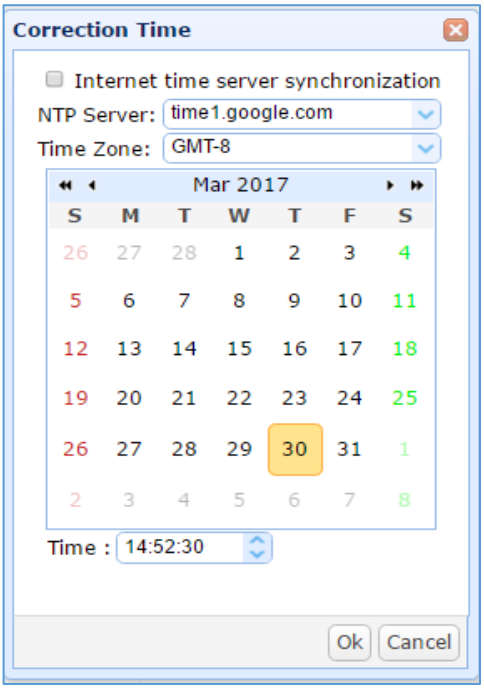
3.2.1. System Information

The setting window displays the system information and allows to modify the system.



Function items	Description	Default
IP Address	The IP address of the UA-5200.	System value
Netmask	The mask address of the UA-5200.	System value
Host Name	The host name of the UA-5200.	System value
User Name	The login name for the UA-5200's Web UI.	
Password	The login password for theUA-5200's Web UI.	
Date	Time/Time zone setting, NTP network time synchronization.	System value

Click the small button beside the "Date & Time" can set the time and the time synchronization.

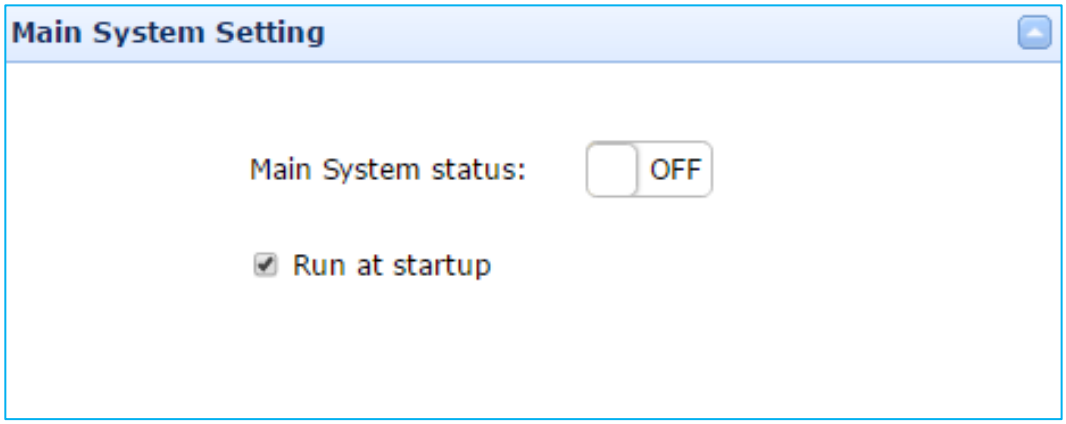


 :

Function items	Description	Default
Internet time server synchronization	Enable the internet time synchronization	Uncheck
NTP Server	Set up the IP address or domain name of the NTP time server.	
Time Zone	Set up the time zone	
Time	Set up the date and time	

3.2.2. Main System Setting

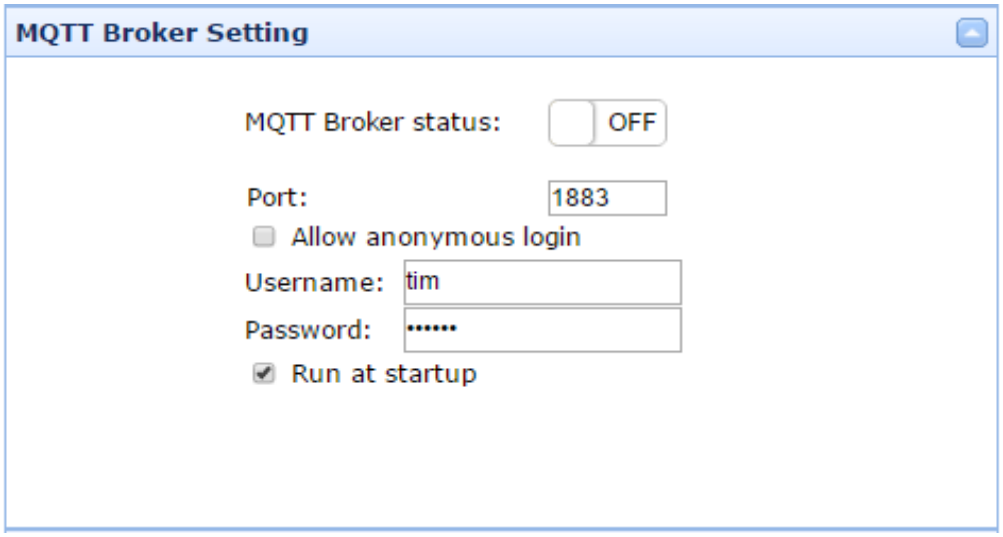
The “Main System Setting” can display and modify the running status of the main system.



Function items	Description	Default
Main System status	Display the status of the main system and allows switching this function to Run/Stop the main system. ON: Run the main system OFF: Stop the main system	OFF
Run at startup	Whether to run at startup.	Uncheck

3.2.3. MQTT Broker Setting

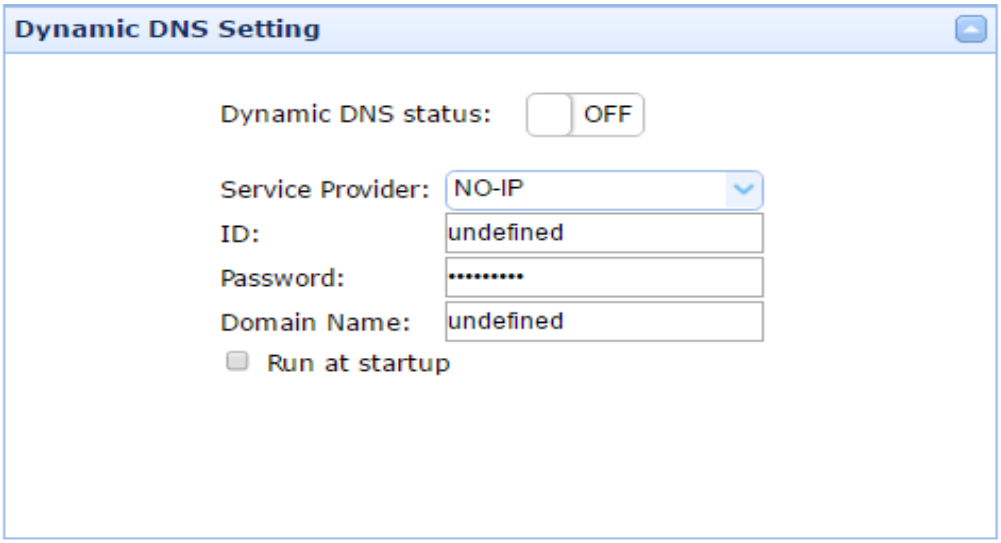
This setting window can display, set up and modify the status for the MQTT Broker.



Function items	Description	Default
MQTT Broker status	Display the status of the Broker and allows switching this function to ON/OFF the MQTT Broker. ON: Run the MQTT Broker OFF: Stop the MQTT Broker	OFF
Port	MQTT Broker's COM port.	1883
Allow anonymous login	Check to allow anonymous login	Check
Username	Set up the login user name	
Password	Set up the login password	
Run at startup	Whether to run at startup.	Uncheck

3.2.4. Dynamic DNS Setting

This window displays the current status of the DDNS Client and allows user to change the settings.

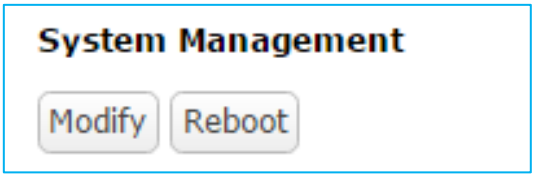


Function items	Description	Default
Dynamic DNS status	Display the current status of the DDNS Client and allows switching this function to ON/OFF the DDNS. ON: Run the DDNS OFF: Stop the DDNS	OFF
Service Provider	Select the company of the DDNS service. Supports: NO-IP, ChangeIP.com, DynDNS, FreeDNS	NO-IP
ID	Set up the login user ID	undefined
Password	Set up the login password	
Domain Name	Define the parked domain name of the DDNS	undefined
Run at startup	Whether to run at startup.	Uncheck

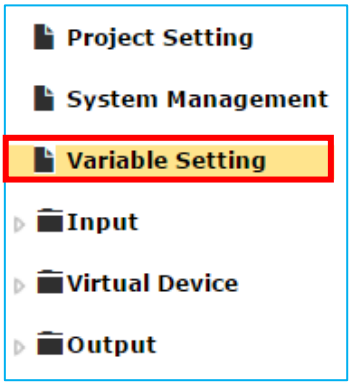
3.2.5. Save the System Management Settings

Click the “Modify” button to save the current settings.

Click “Reboot” can restart the UA-5200.

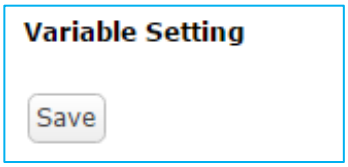


3.3. Variable Setting



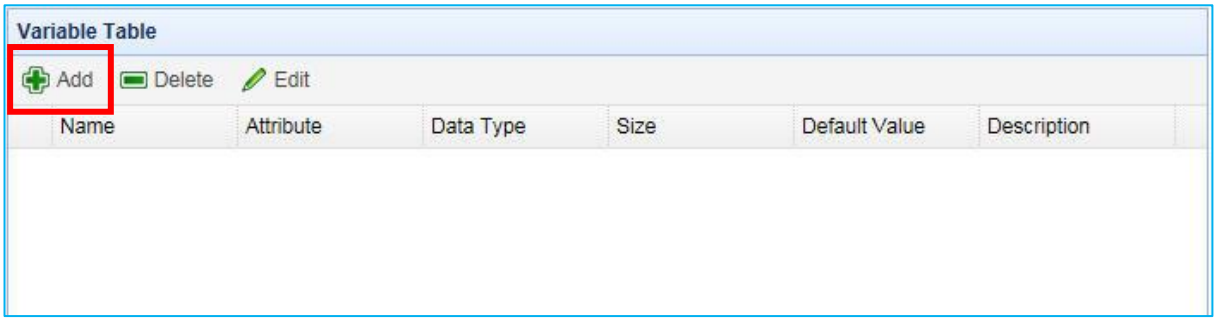
This section describes the variable configuration in the “Variable Setting” function, and then save the settings.

After setting, click “Save” to save all variables.



Steps:

1. Use tool buttons to add the needed variables in the Variable Table. (as the figure below). Click the “Add” to add a variable. If need, edit or delete the existing variable.



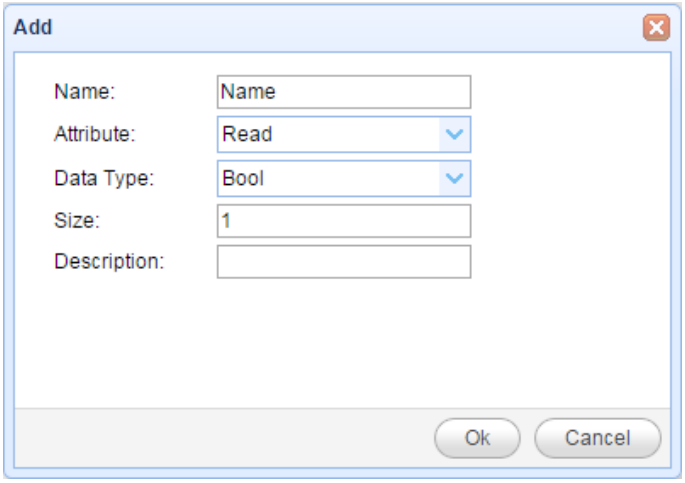
Description of the tool button:

: Add a variable

: Delete the selected variable

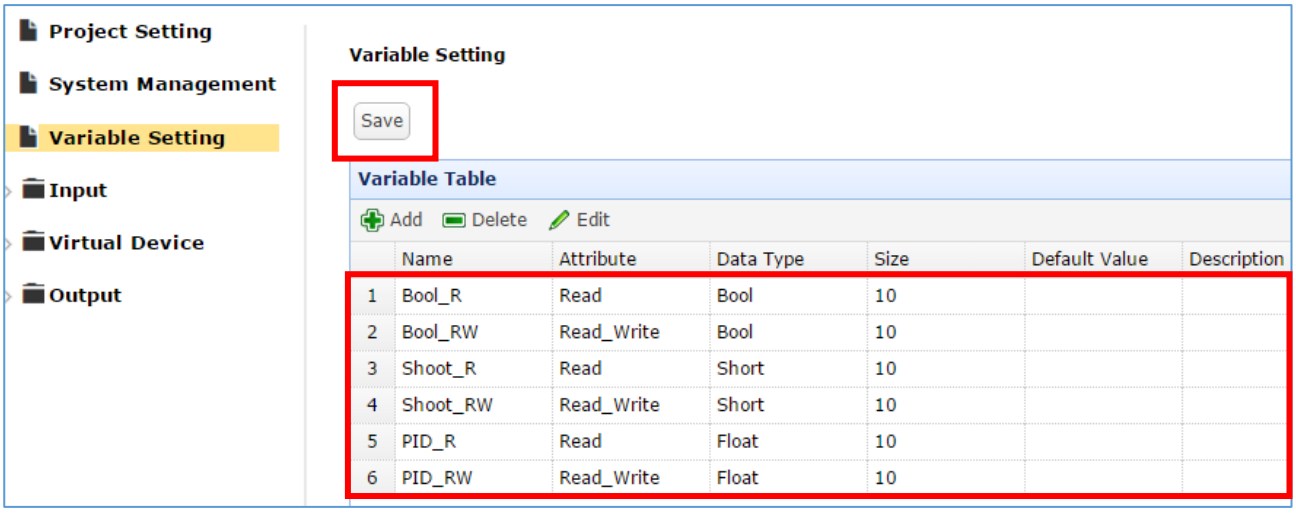
: Edit the selected variable

2. Set up the parameters of the variable.

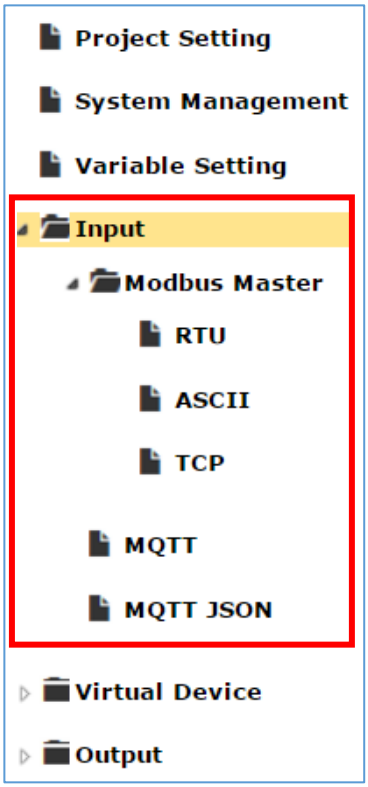


Function items	Description	Default
Name	Variable name.	Name
Attribute	Variable attribute. Options: Read, Write, Read_Write	Read
Data Type	Variable data type. Options: Bool, Short, Unsigned Short, Long, Unsigned Long, Float, Double, String	Bool
Size	Variable size. If this value is greater than 1, this variable will be declared as an array.	1
Description	Write a note for this variable.	

3. Using the tool buttons set up the needed variables as following Variable Table.
   At last, remember to click “Save” to save all settings.



3.4. Input



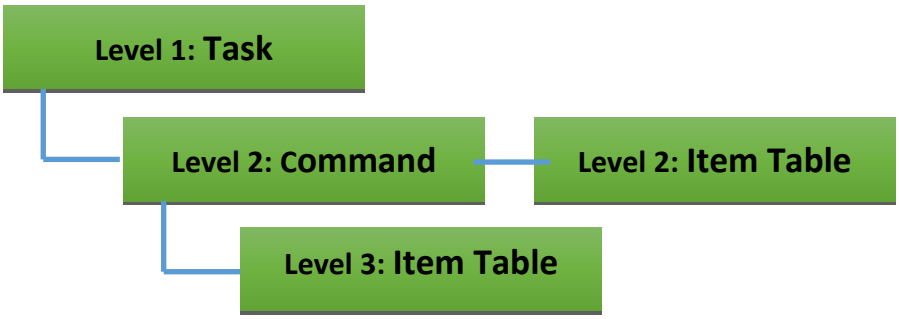
This section describes the Input setting and all related parameters for the UA-5200 series.

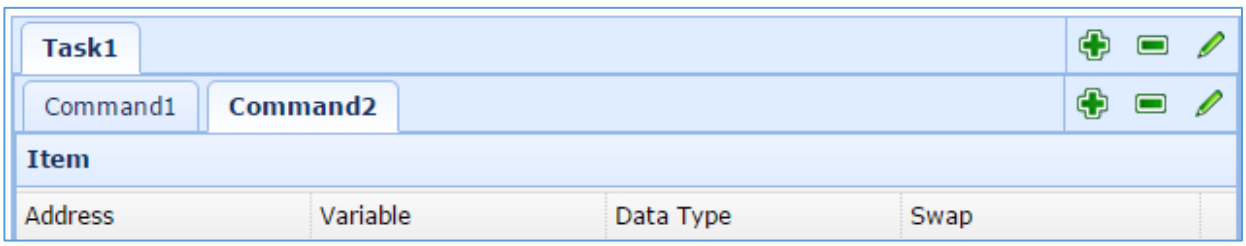
This topic includes:

- Modbus Master (RTU, ASCII and TCP)
- MQTT
- MQTT JSON

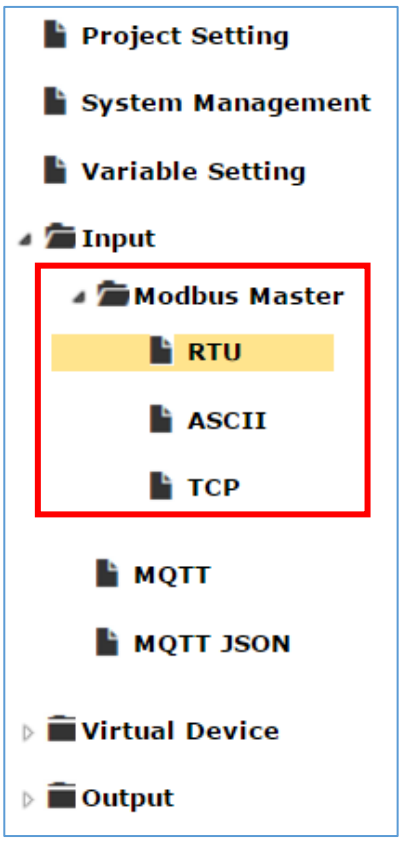
Description of the Setting Steps:

All function settings follow the task-oriented and level logic architecture. First, the user can add a connection **Task** depends on device properties, and then add **Command** or **Item** table under the Task. Finally, you can complete two-level or three-level settings.





3.4.1. Modbus Master



This section describes the Input setting of the Modbus Master. It includes RTU, ASCII and TCP protocols.

The parameters for setting the RTU and ASCII are similar, here will describe them together.

A. RTU / ASCII

B. TCP

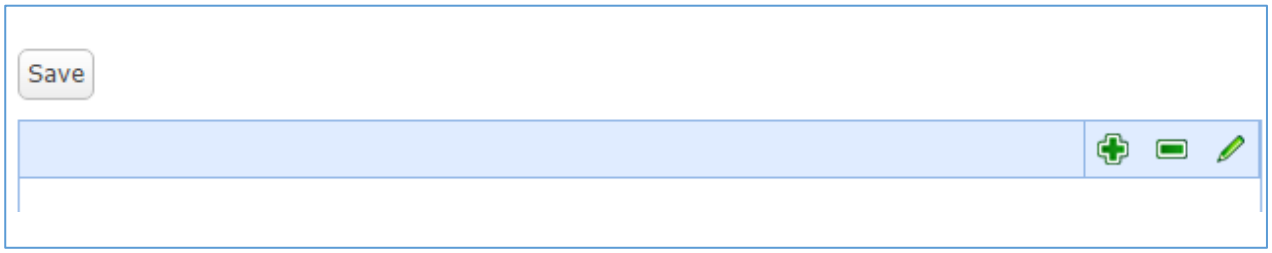
Description of the tool button:

: Add a task, command, or item.

: Delete a task, command, or item.

: Modify a task, command, or item.

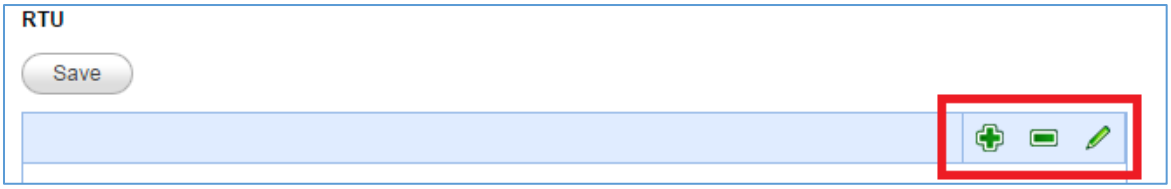
Save : After setting, click “Save” to save all.

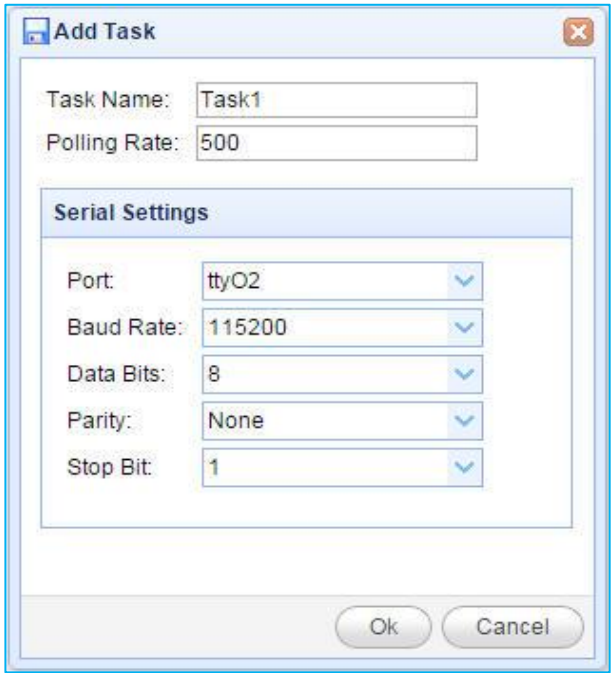


A. RTU/ASCII

Here will show how to add, edit and delete the Modbus RTU/ASCII Master Input in the RTU page.

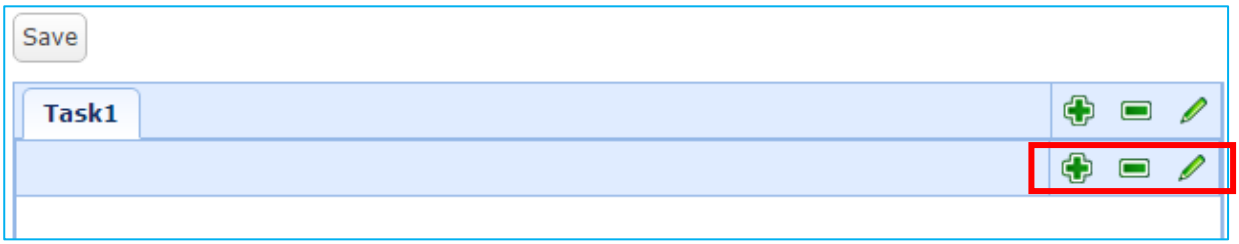
1. On the “Input” -> “Modbus Master” -> “RTU” or “ASCII” page, add a Task and edit it.

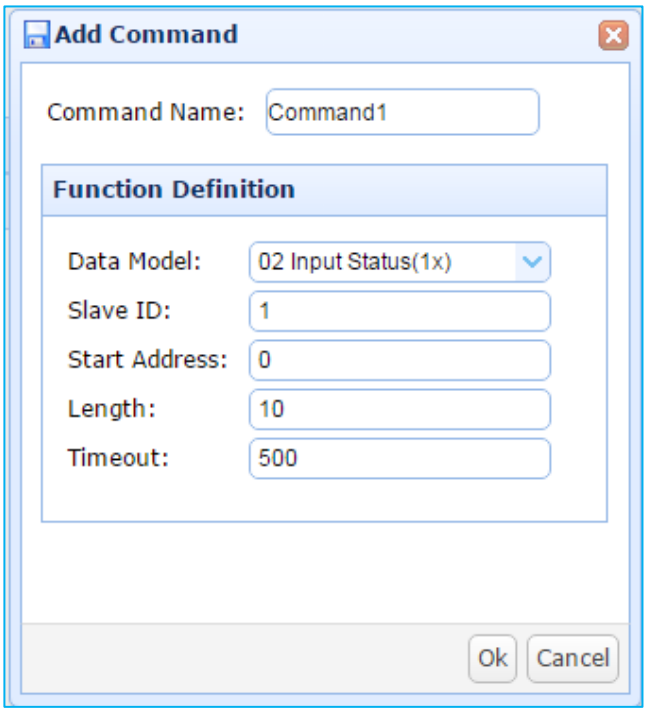




Function items	Description	Default
Task Name	Give a task name.	Task1
Polling Rate	Set a time interval for the command.	500
	Serial Settings	
Port	Choose a serial port number. Please check which RS-232/485 port is in use. Note: The wrong setting will cause the communication error.	TtyO2
Baud Rate	Choose a baud rate. Make sure the module’s baud rate is correct. Note: The wrong setting will cause the communication error.	115200
Data Bits	The number of bits used to represent one byte of data.	8
Parity	Choose one way for the parity checking. Options: None, Even, and Odd.	None
Stop Bit	Choose the number of stop bit.	1

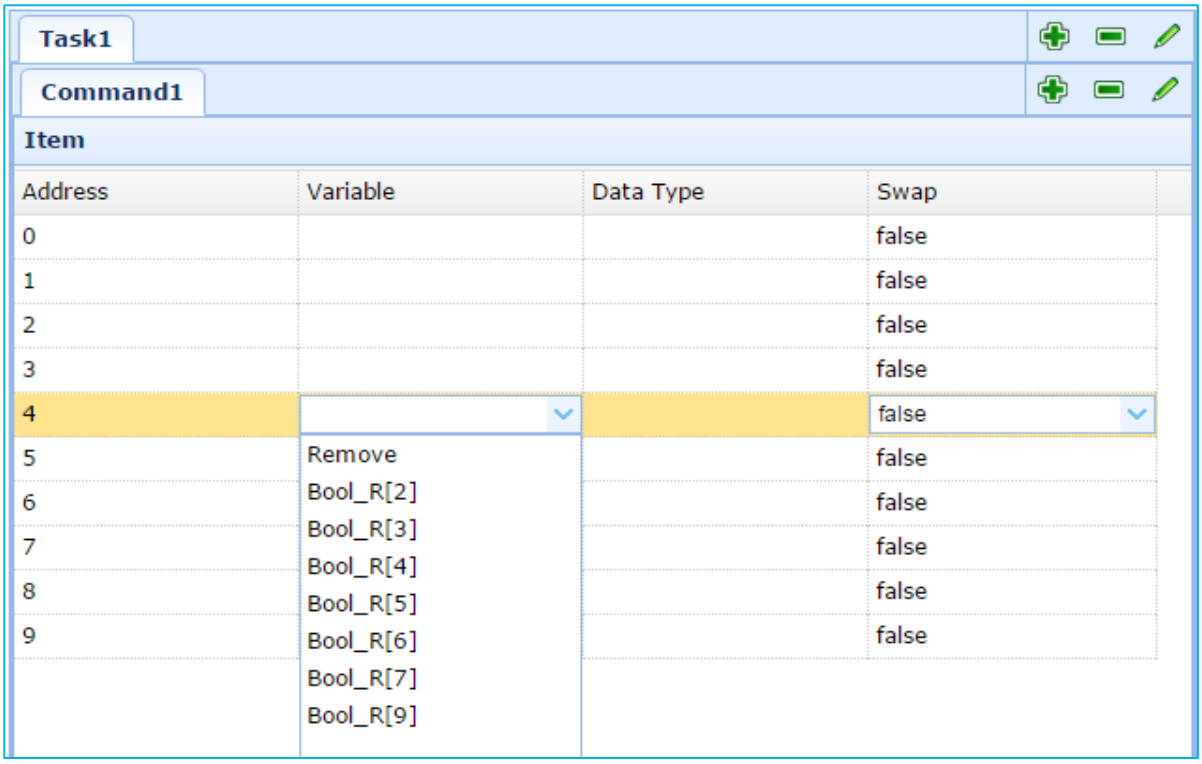
2. Click the tool buttons under the task tab to add and configure the command.





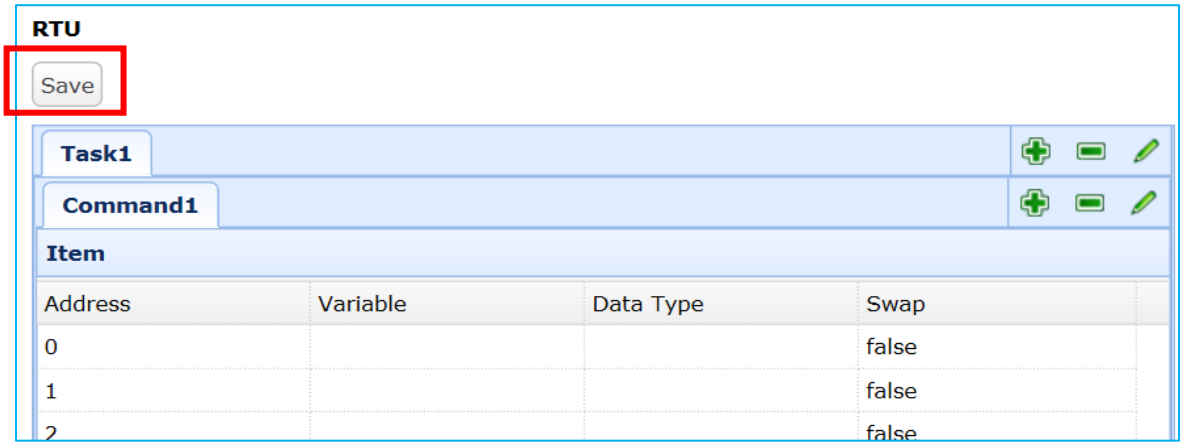
Function items	Description	Default
Command Name	Give a command name.	Command1
Function Definition		
Data Model	Choose the data type for the Modbus command.	02 Input Status(1x)
Slave ID	Set the Slave ID of the UA-5200. (Range: 1 ~ 247)	1
Start Address	The start address of the Modbus command.	0
Length	The number of the Modbus address.	10
Timeout	Set the timeout value for the module.	500

3. Double click the Variable and Swap row to set up the items in the command table.

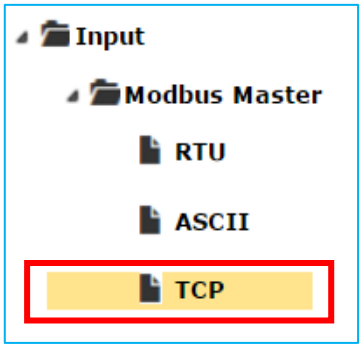


Function items	Description	Default
Address	Modbus address.	Auto arrange
Variable	Choose the variable you set before. (See 3.2 Variable Setting)	
Data Type	After selecting the variable, its data type will automatically display. (Not editable)	
Swap	To swap the byte order (Lo-Hi/Hi-Lo) for a 4-byte or 8-byte value. (False: disable/True: enable)	False (disable)

4. Click “Save” to save current settings.

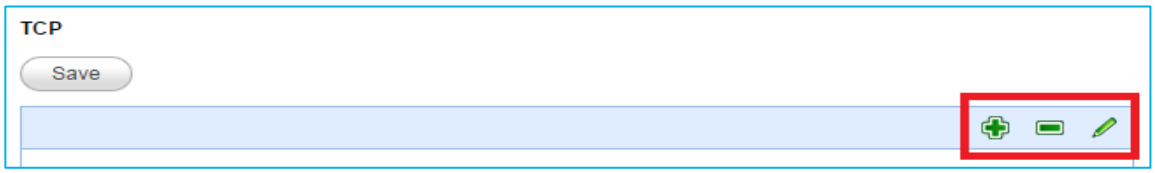


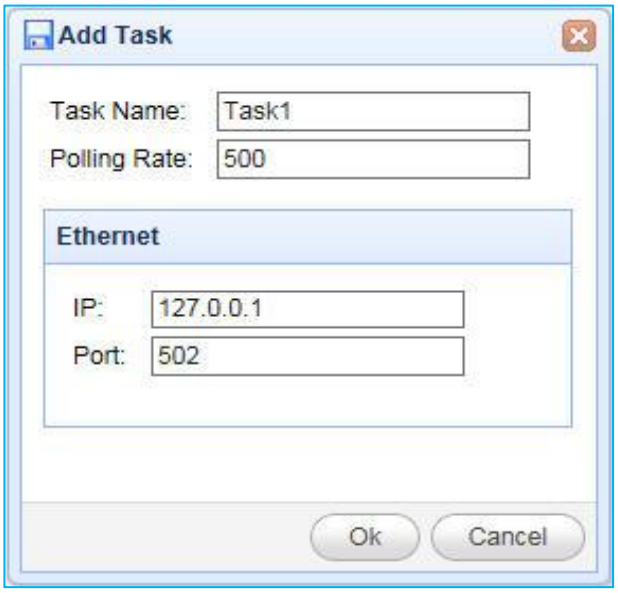
B. TCP



This section describes the Input settings of the Modbus TCP Master.

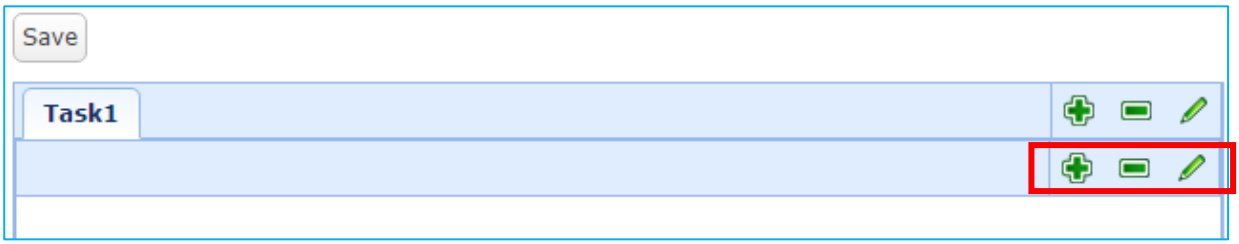
1. On the “Input” -> “Modbus Master” -> “TCP” page, add a Task and edit the setting.

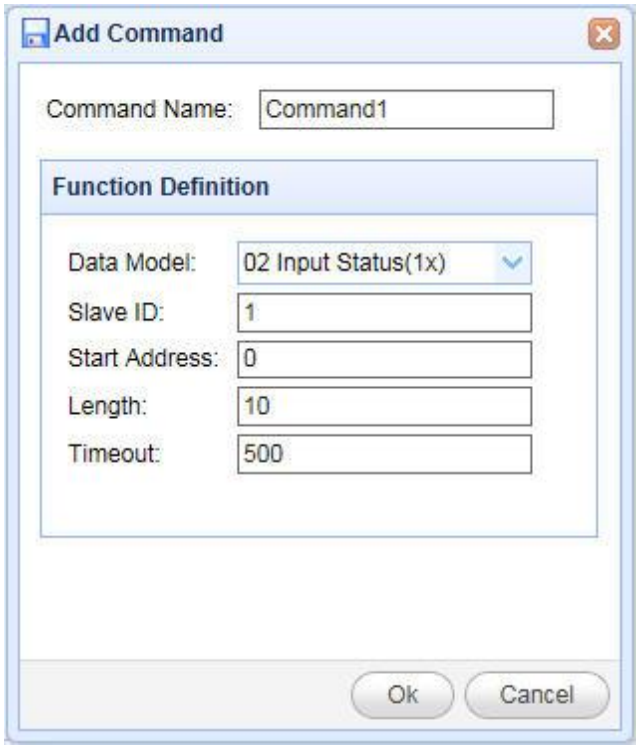




Function items	Description	Default
Task Name	Give a task name.	Task1
Polling Rate	Set an interval time for each command.	500
Ethernet		
IP	The IP address of the connected device.	127.0.0.1
Port	The port number for Modbus TCP.	502

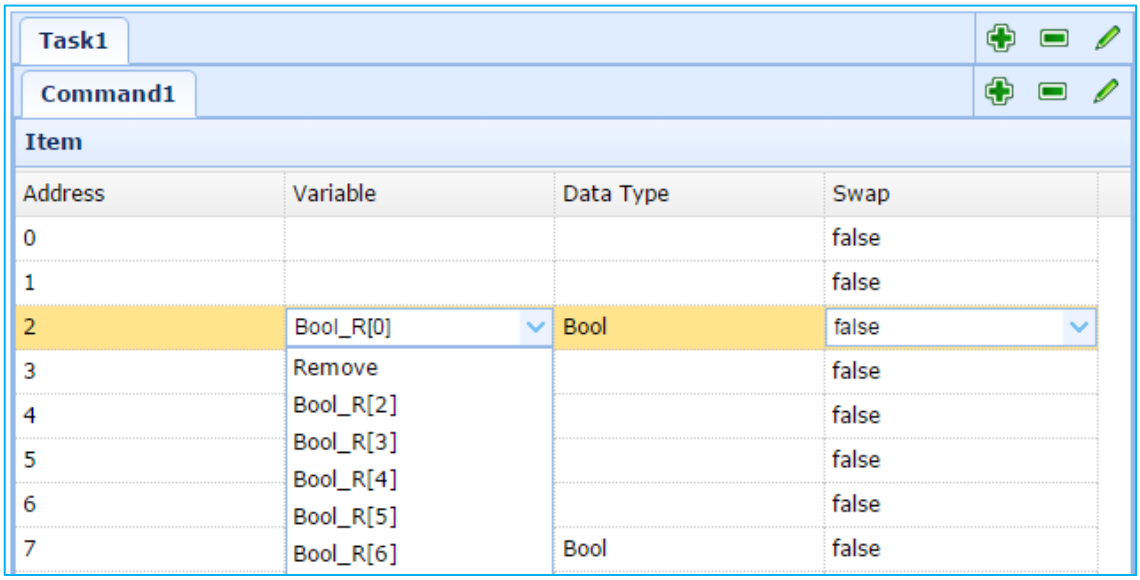
2. Click the tool buttons under the Task tab to add and configure the command.





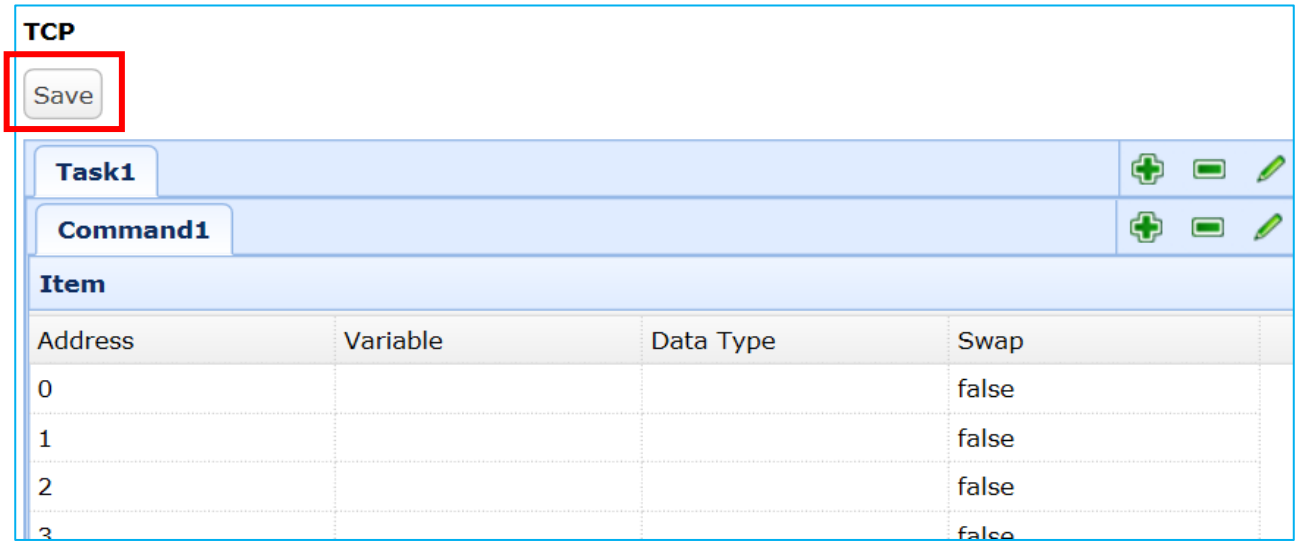
Function items	Description	Default
Command Name	Give a command name.	Command1
Function Definition		
Data Model	Choose the data type for the Modbus command.	02 Input Status(1x)
Slave ID	Set the Slave ID of the UA-5200. (Range: 1 ~ 247)	1
Start Address	The start address of the Modbus command.	0
Length	The number of the Modbus address.	10
Timeout	Set the timeout value for the module.	500

3. Double click the Variable and Swap row to set up the items in the command table.



Function items	Description	Default
Address	Modbus address.	Auto arrange
Variable	Choose the variable you set before. (See 3.2 Variable Setting)	
Data Type	After selecting the variable, its data type will automatically display. (Not editable)	
Swap	To swap the byte order (Lo-Hi/Hi-Lo) for a 4-byte or 8-byte value. (False: disable/True: enable)	False (disable)

4. Click “Save” to save the current settings.



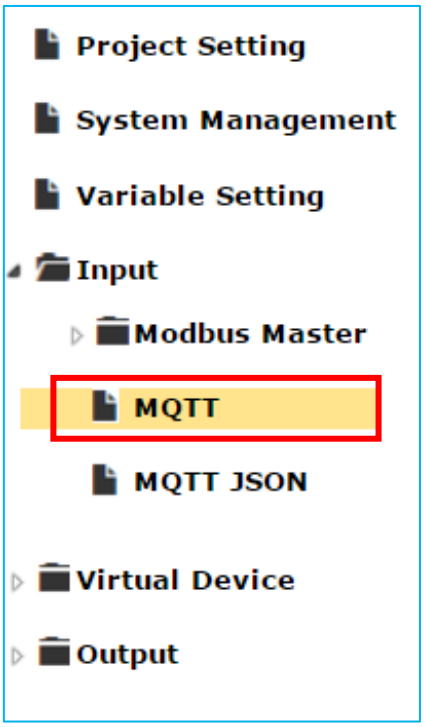
3.4.2. MQTT

This section describes the Input setting of the MQTT communication.

MQTT setting sequence of Input function:

1. Set up the MQTT client to connect to the Broker.
2. Map the contents of the Topic messages published and subscribed by other external MQTT devices to the UA-5200 user's configured variables.
3. Convert the contents of the MQTT device to become the contents of other communication protocol data.

For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.



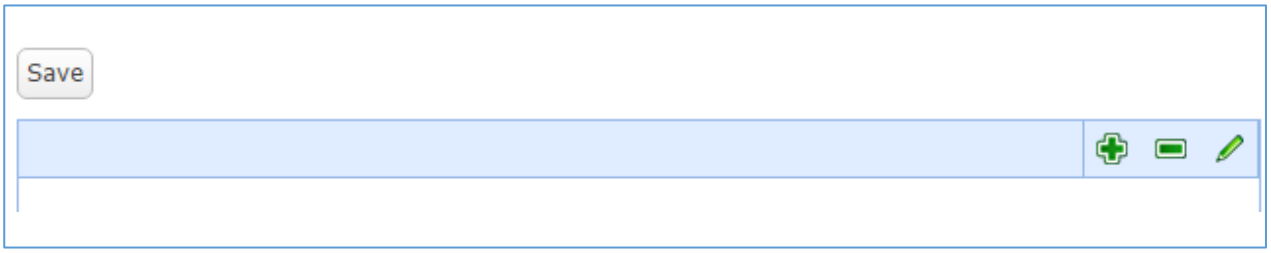
Description of the tool button:

: Add a task, command, or item.

: Delete a task, command, or item.

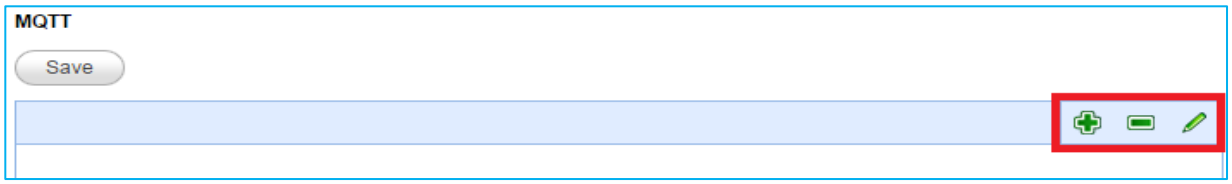
: Modify a task, command, or item.

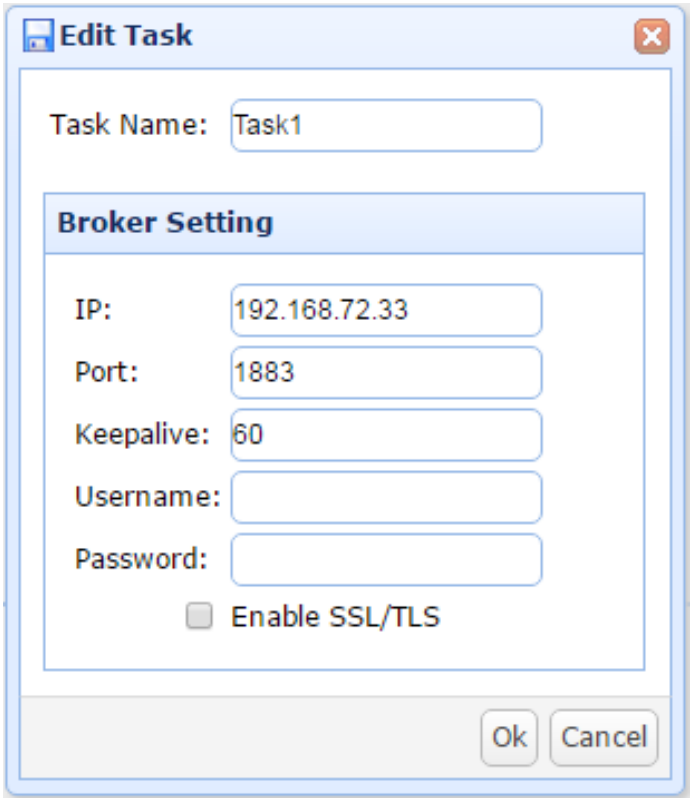
Save : After setting, click “Save” to save all.



Steps:

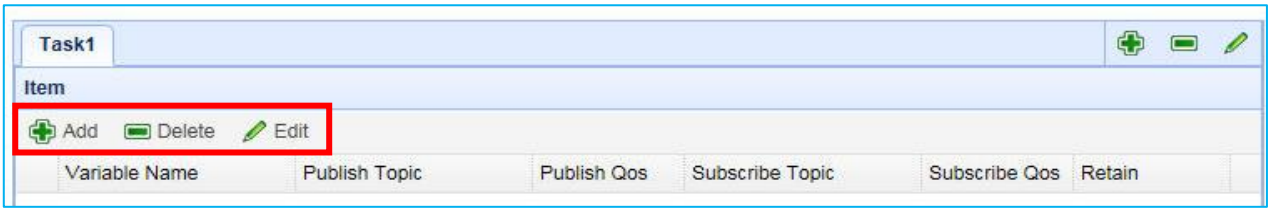
1. On the “Input” -> “MQTT” page, add a Task and edit the setting.

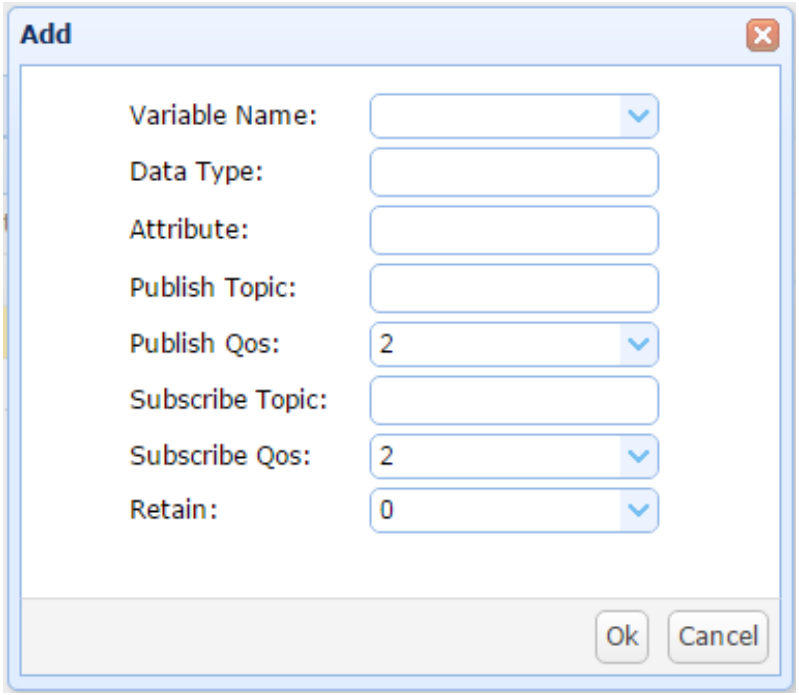




Function items	Description	Default
Task Name	Give a task name.	Task1
Broker Setting		
IP	The IP address of the Broker.	System value
Port	The Broker port.	1883
Keepalive	Keep alive time.	60
Username	The user name to login the Broker.	Null
Password	The password to login the Broker.	Null
Enable SSL/TLS	Check to enable the supporting of SSL/TLS security communication.	Uncheck

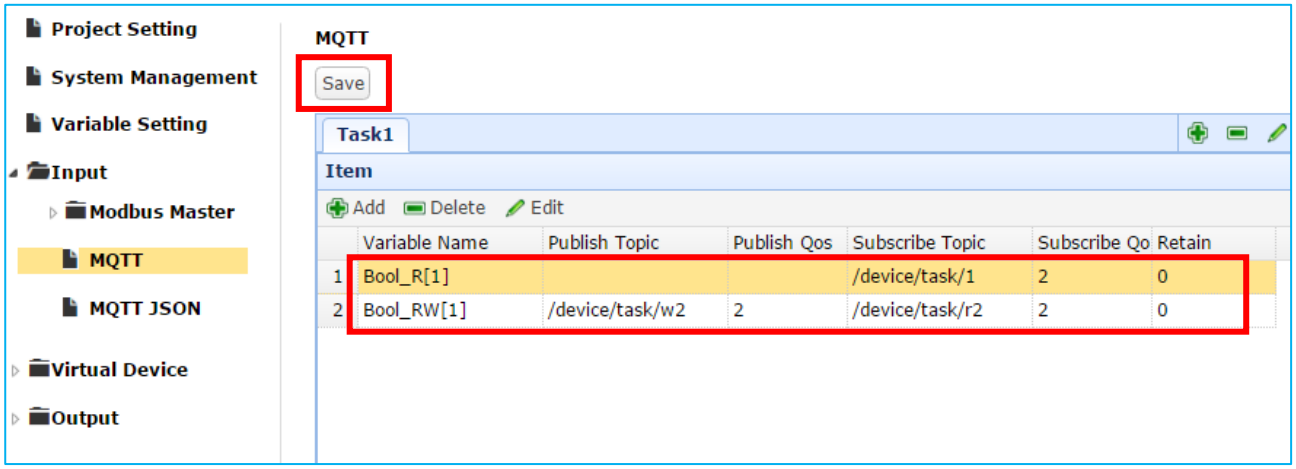
2. Add and set up the variable and related parameters under the **MQTT** task tab.





Function items	Description	Default
Variable Name	Choose a variable which pre-defined in the variable table.	
Data Type	Not editable. It will show the data type of a variable.	System value
Attribute	Not editable. It will show the variable attribute.	System value
Publish Topic	The topic of sending data or publishing messages.	
Publish Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Subscribe Topic	The topic of receiving data or subscribing messages.	
Subscribe Qos	The subscribe Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Retain	Whether to store a broker message. (0: No ; 1: Yes)	0

3. After configuration, the MQTT Task table is similar to the following picture.
   At last, click “Save” to save the current settings.



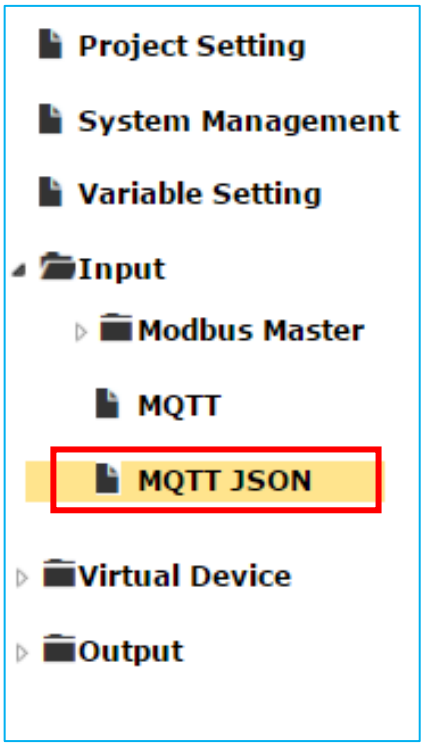
3.4.3. MQTT JSON

This section describes the Input setting of the MQTT JSON communication.

MQTT JSON setting sequence of Input function:

1. Set up the MQTT client to connect to the Broker.
2. Map the contents of the Topic messages published and subscribed by other external MQTT devices that support the UA-5200 JSON format to the UA-5200 user's configured variable group.
3. Convert the contents of the MQTT device to become the contents of other communication protocol data.

For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.



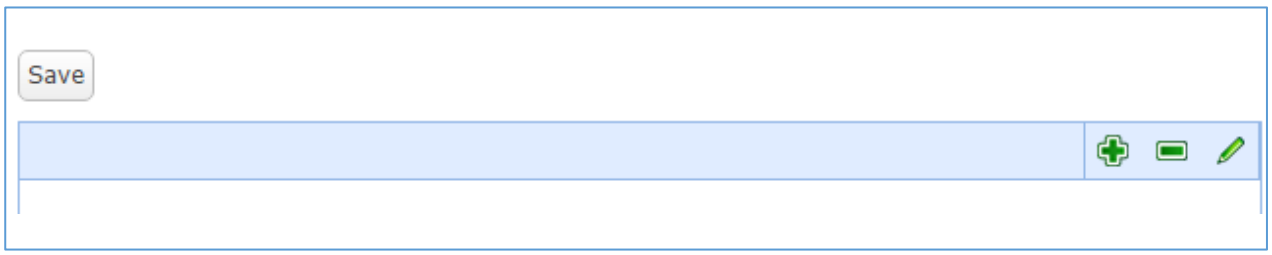
Description of the tool button:

: Add a task, command, or item.

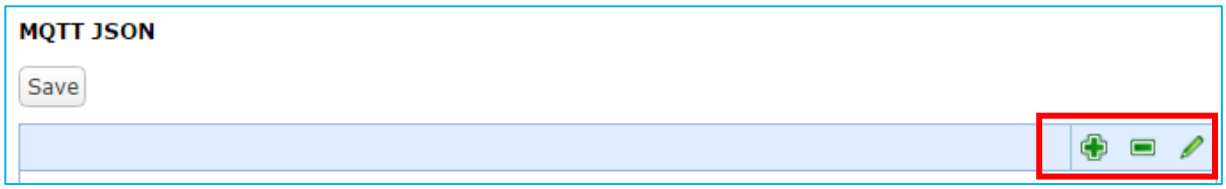
: Delete a task, command, or item.

: Modify a task, command, or item.

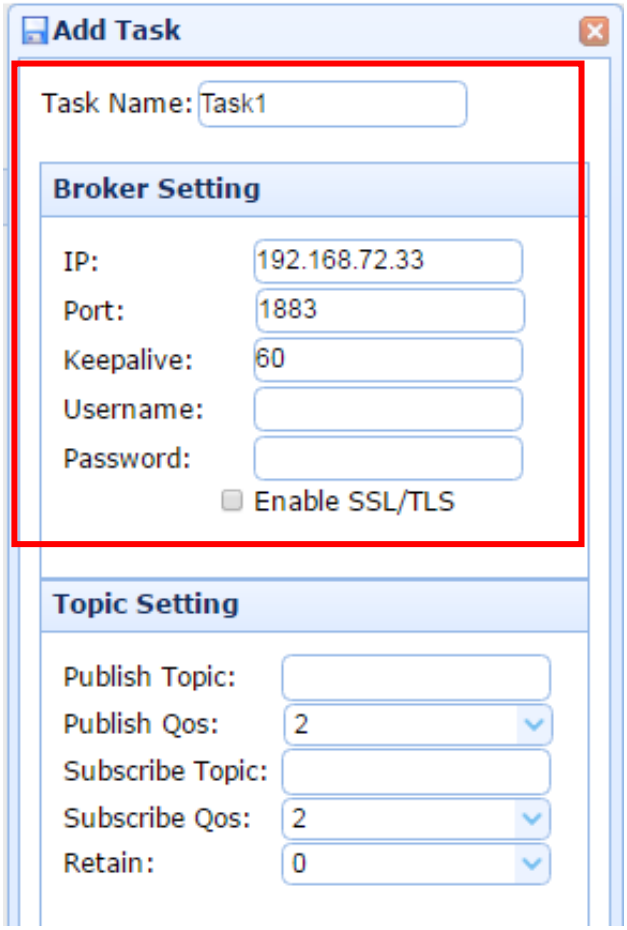
Save : After setting, click “Save” to save all.



1. On the “Input” -> “MQTT JSON” page, add a **Task** and edit the setting.

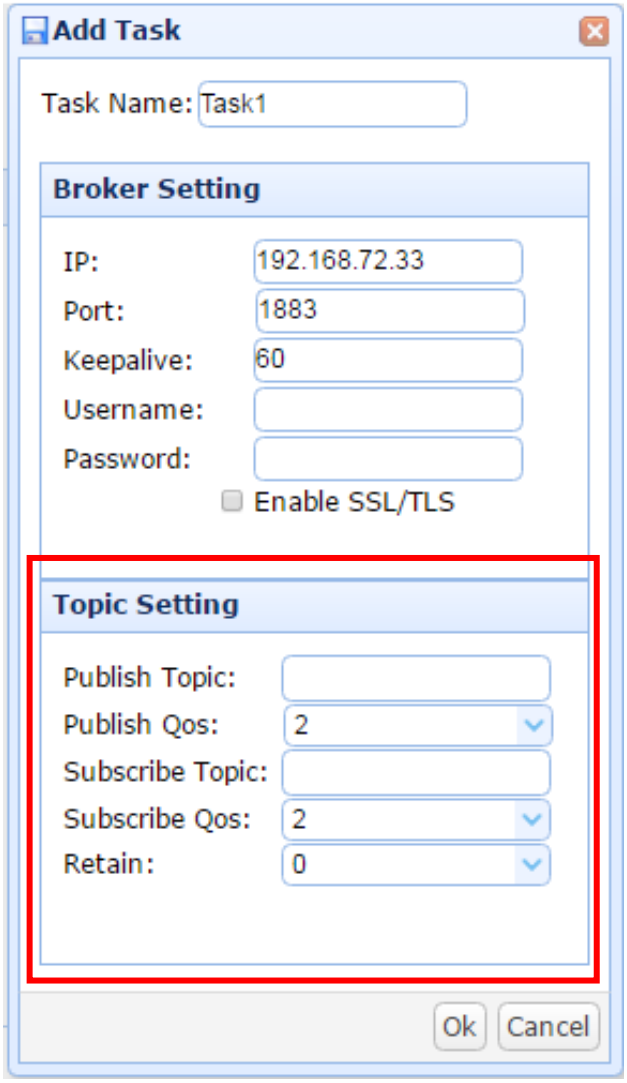


Broker Setting of the MQTT JSON Task:



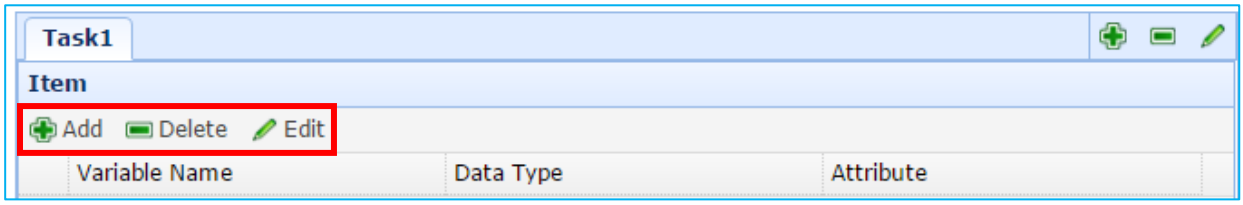
Function items	Description	Default
Task Name	Give a task name.	Task1
Broker Setting		
IP	The IP address of the Broker.	System value
Port	The Broker port.	1883
Keepalive	Keep alive time.	60
Username	The user name to login the Broker.	Null
Password	The password to login the Broker.	Null
Enable SSL/TLS	Check to enable the supporting of SSL/TLS security communication.	Uncheck

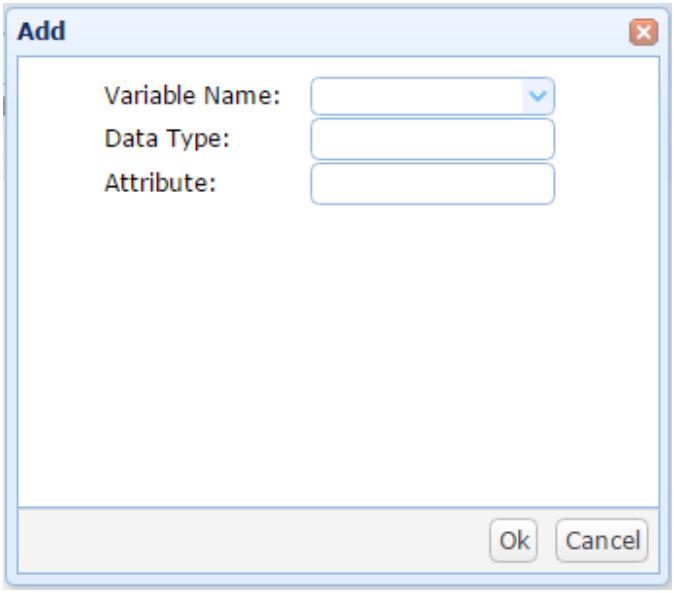
Topic Setting of the MQTT JSON Task:



Function items	Description	Default
Topic Setting		
Publish Topic	The topic of sending data or publishing messages.	
Publish Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Subscribe Topic	The topic of receiving data or subscribing messages.	
Subscribe Qos	The subscribe Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Retain	Whether to store a broker message. (0: No ; 1: Yes)	0

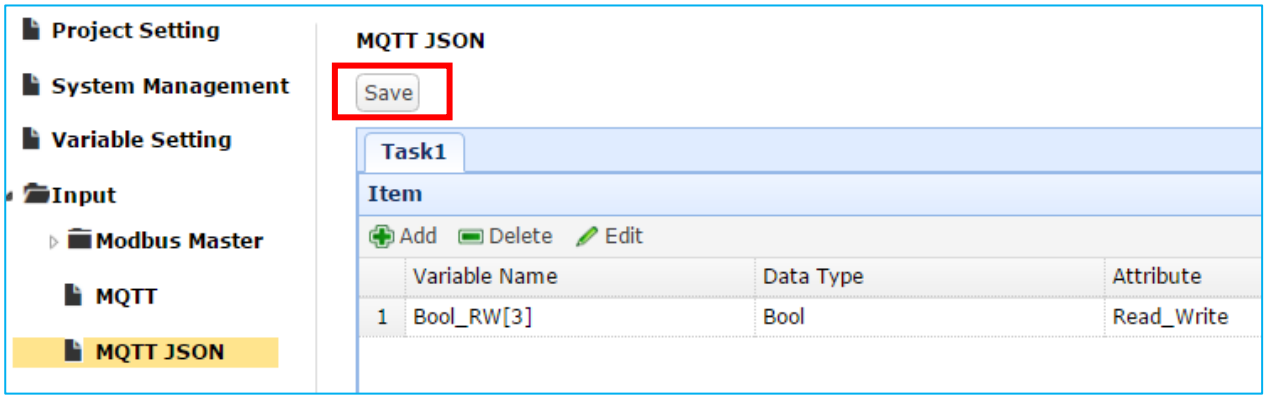
2. Under the MQTT JSON Task, add the MQTT JSON Items and assign the variables.





Function items	Description	Default
Variable Name	Choose a variable which pre-defined in the variable table.	
Data Type	Not editable. It will show the data type of a variable.	System value
Attribute	Not editable. It will show the variable attribute.	System value

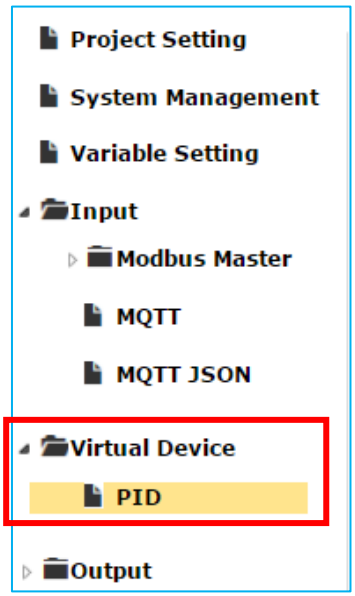
3. After configuration, the MQTT JSON Task table is similar to the following picture.
   At last, click “Save” to save the current settings.



3.5. Virtual Device

This Virtual Device function allows the user to simulate various devices with the real I/O by using the PID tuning function. This article includes the PID function.

3.5.1. PID



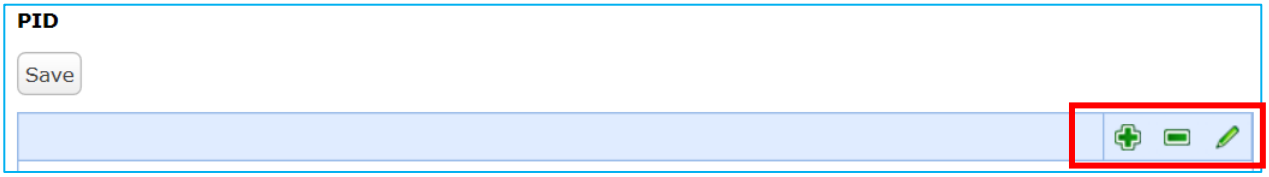
This section describes the "**PID**" settings of "Virtual Device" function.

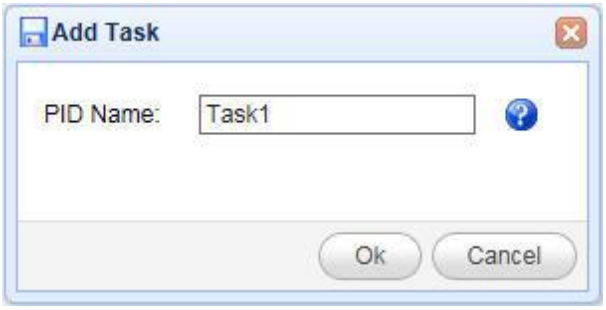
PID (Proportional-Integral-Derivative) control is the most widely used in industrial control systems. A regulator which is controlled in accordance with Proportional, Integral and Derivative is called PID control for short, also called PID regulator. When the user cannot fully grasp or measure parameters of the control system, the PID regulator is the best solution.

Description of the tool button:

: Add a task, command, or item.
: Delete a task, command, or item.
: Modify a task, command, or item.

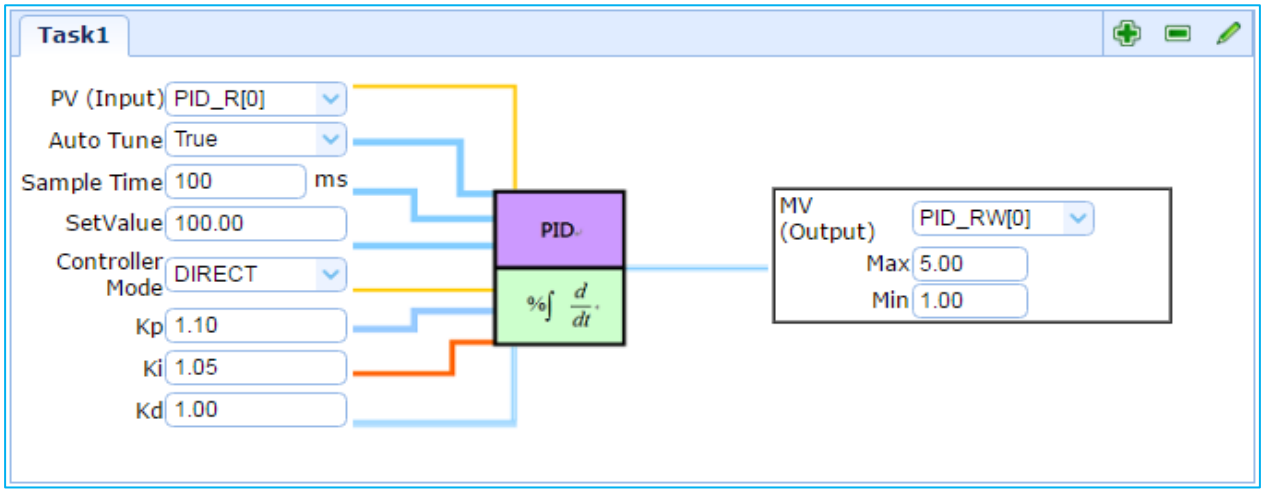
1. Add a Task and give a PID task name.





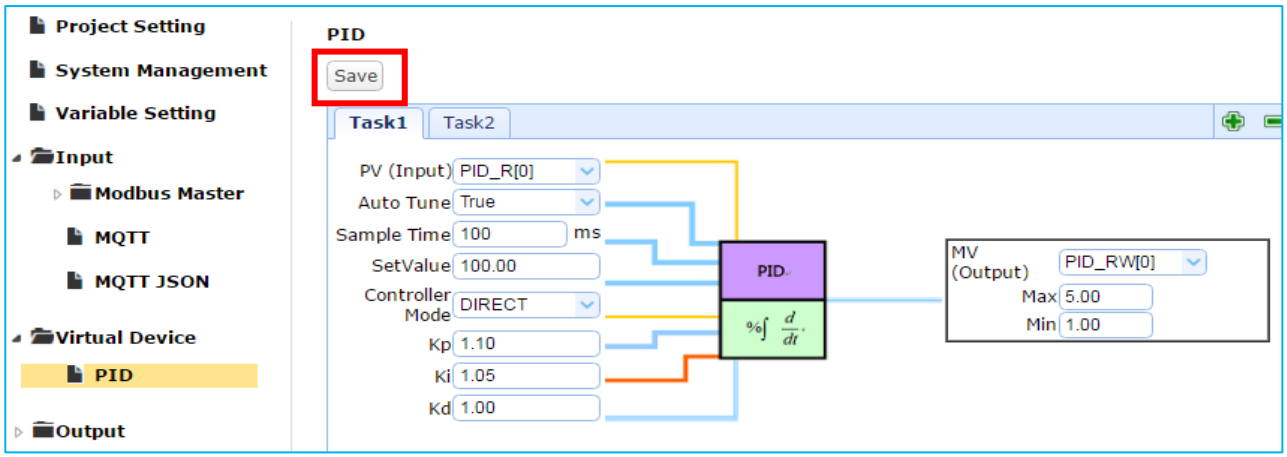
Function items	Description	Default
PID Name	Give a PID task name.	Task1

2. Configure related parameters for the PID device in the Task tab.

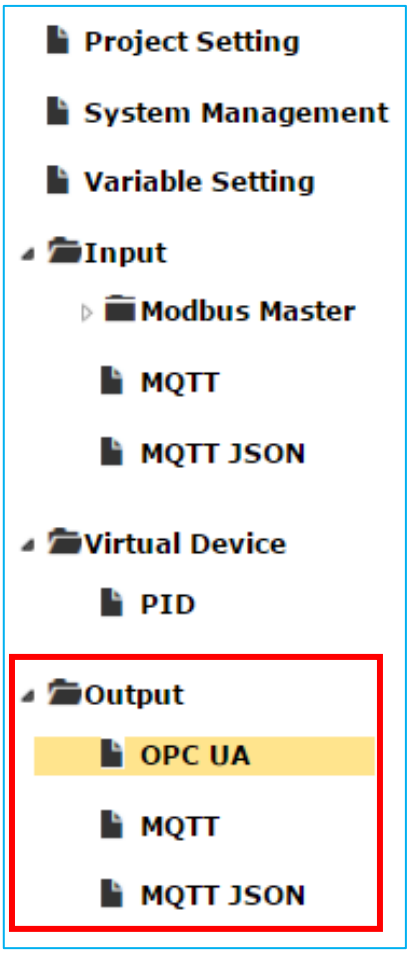


Function items	Description	Default
PV(Input)	Choose a predefined float variable as the input parameter.	
Auto Tune	True: Auto-tuning PID parameters for your system. False: Tuning PID parameters manually.	True
Sample Time	Set the sampling time.	500
Set value	The target value for PID control.	0
Controller mode	DIRECT: Set it as positive output value. REVERSE: Set it as reverse output value.	DIRECT
Kp	Set the Proportional gain.	1.0
Ki	Set the Integral gain.	1.0
Kd	Set the Derivative gain.	1.0
MV(Output)	Choose a preset floating variable as output.	
Max	Set the upper-limit value for the variable.	0
Min	Set the lower-limit value for the variable.	0

3. Click “Save” to save the current settings.



3.6. Output



This section describes the configuration the “Output” function of the UA-5200.
It includes the OPC UA, MQTT and MQTT JSON.

Output Functions:

- OPC UA (Server and Security)
- MQTT
- MQTT JSON

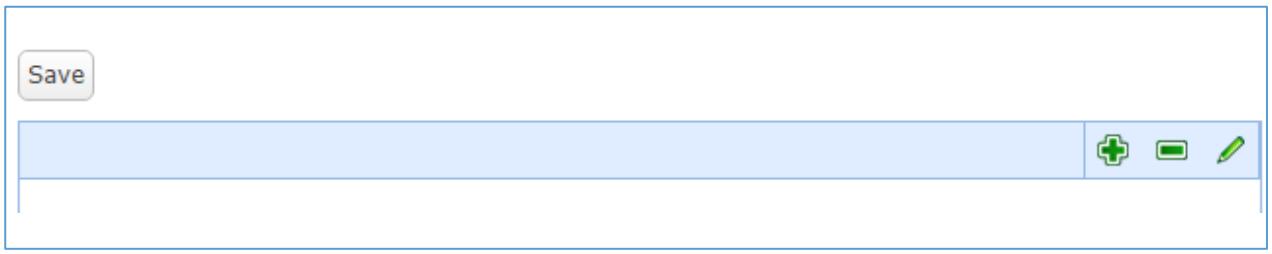
Description of the tool button:

: Add a task, command, or item.

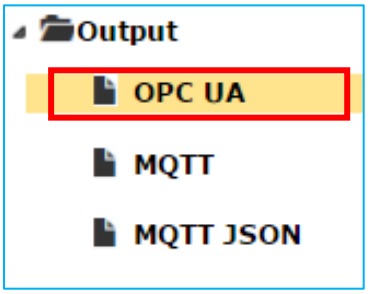
: Delete a task, command, or item.

: Modify a task, command, or item.

Save : After setting, click “Save” to save all.



3.6.1. OPC UA



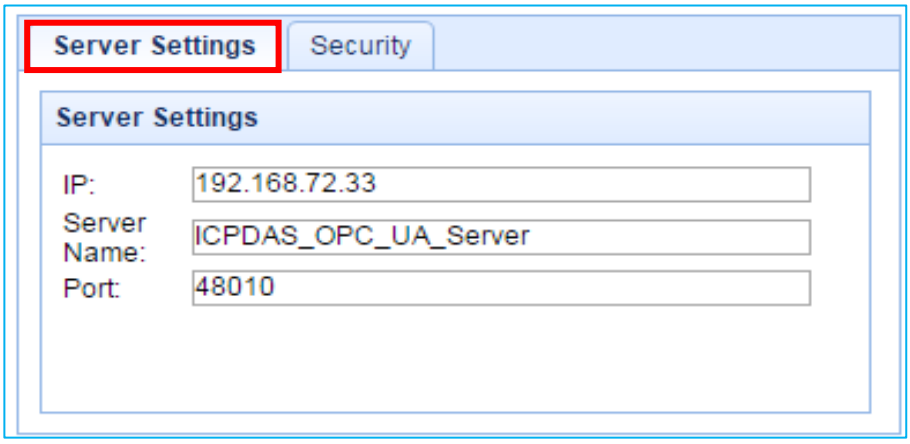
This section describes the “OPC UA” configuration of the “Output” function. It includes:

A. Server Settings

B. Security

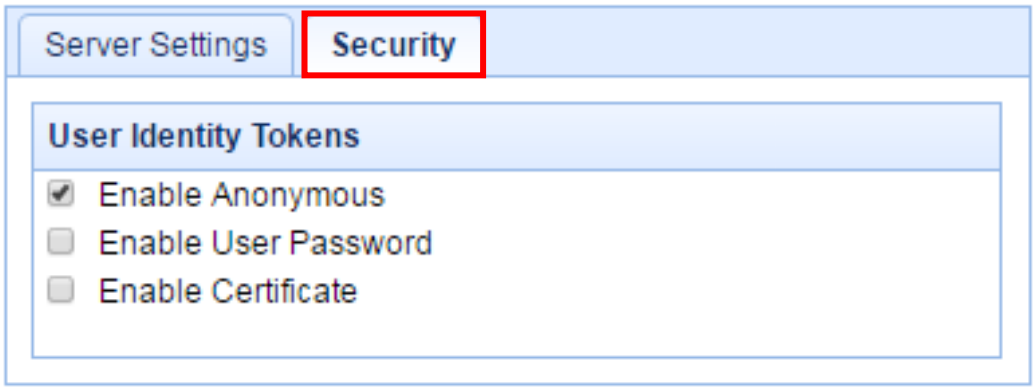
For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.

A. Server Settings



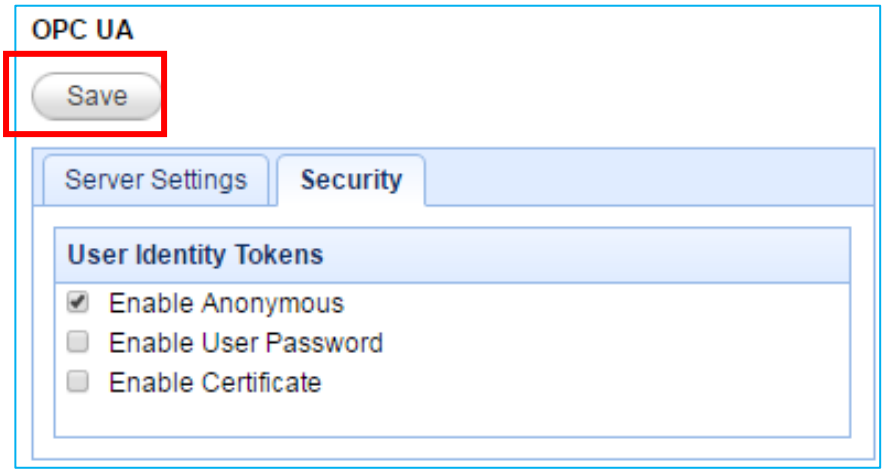
Function items	Description	Default
IP	Display the IP address of the Local host.	System value
Server Name	Display the active OPC UA Server name. Not editable.	ICPDAS_OPC_UA_Server
Port	The communication port number of the OPC UA Server.	48010

B. Security



Function items	Description	Default
User Identity Tokens		
Enable Anonymous	Check: Allow clients to use anonymous access. Uncheck: No anonymous login.	Check
Enable User Password	Check: Allow to log in with username/password. Uncheck: Not supported this way.	Uncheck
Enable Certificate	Check: Allow to log in with certificates Uncheck: Not supported this way.	Uncheck

At last, click “Save” to save the OPC UA settings.



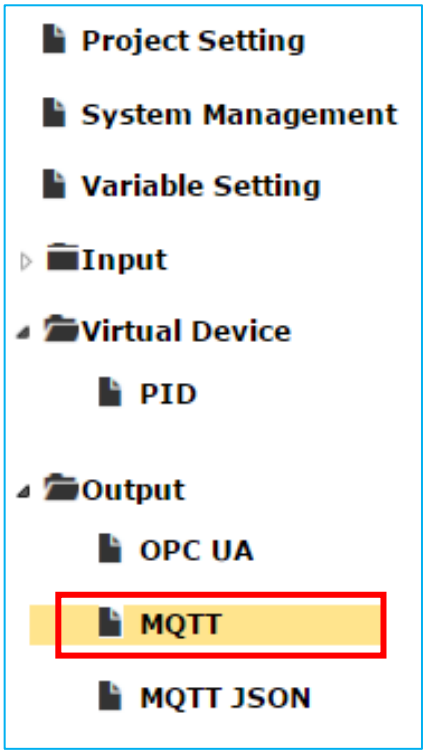
3.6.2. MQTT

This section describes the Output setting of the MQTT communication.

MQTT setting sequence of Output function:

1. Set up the MQTT client to connect to the Broker.
2. Map the read/write function of the UA-5200 user's configured variables to the Topic messages published and subscribed by the MQTT client devices.
3. Convert the contents of other communication devices to become the contents of the MQTT protocol data.

For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.



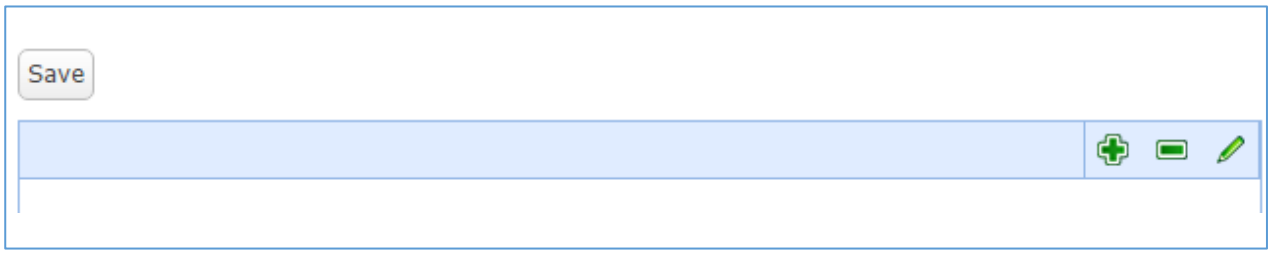
Description of the tool button:

: Add a task, command, or item.

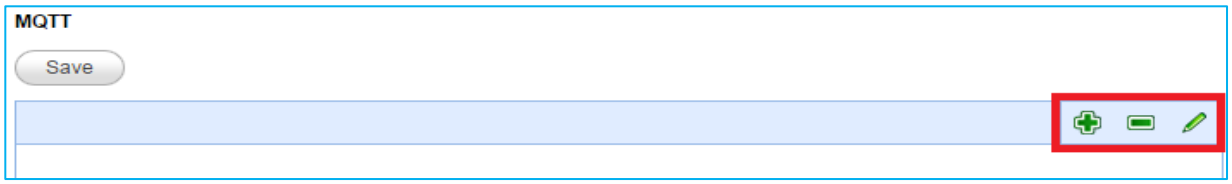
: Delete a task, command, or item.

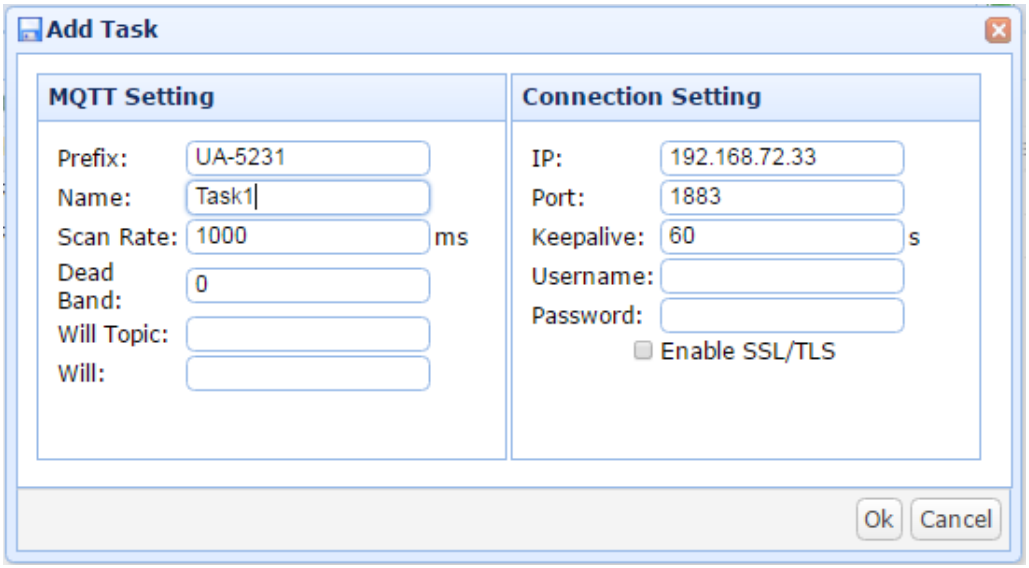
: Modify a task, command, or item.

Save : After setting, click “Save” to save all.



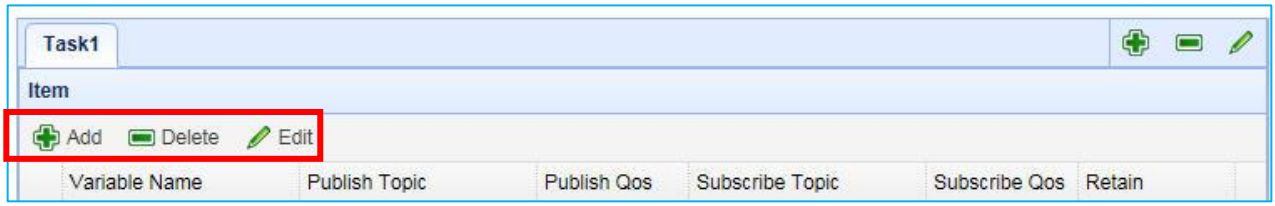
1. Add and configure a Task in the Output -> MQTT tree-menu function.

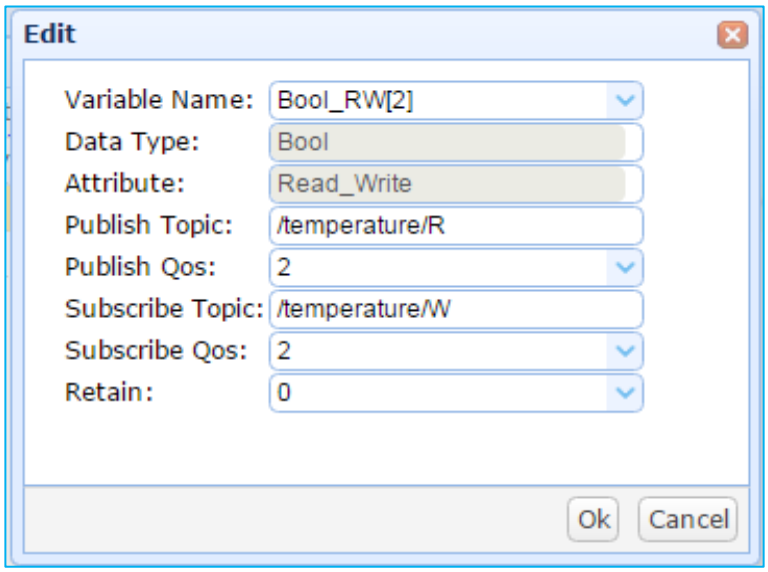




Function Items	Description	Default
MQTT Setting		
Prefix	Set up the prefix for the MQTT topic.	System value
Name	Give a task name.	Task1
Scan Rate	Set an update frequency for the task data. (Unit: ms)	1000
Dead Bend	Give a dead bend value for updating a float signal.	0
Will Topic	Enter the title of a disconnect notice.	Null
Will	Enter a disconnect notice.	Null
Connection Setting		
IP	Set the Broker's IP address.	System value
Port	Set the Broker port.	1883
Keepalive	Set a time to check whether or not the connection to the Broker is working.(Unit: second)	60
Username	The user name to login the Broker.	Null
Password	The password to login the Broker.	Null
Enable SSL/TLS	Check to enable the supporting of SSL/TLS security communication.	Uncheck

2. Add and configure the variable, topic and related parameters in the MQTT task item.





Function items	Description	Default
Variable Name	Choose a variable in the pre-defined variable table.	System value
Data Type	Not editable. It will show the data type of a variable.	System value
Attribute	Not editable. It will show the variable attribute.	System value
Subscribe Topic	The topic of receiving/subscribing data message.	
Subscribe Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Publish Topic	The topic of sending/publishing data message.	
Publish Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2

3. At last, click “Save” to save current settings of the MQTT Task.

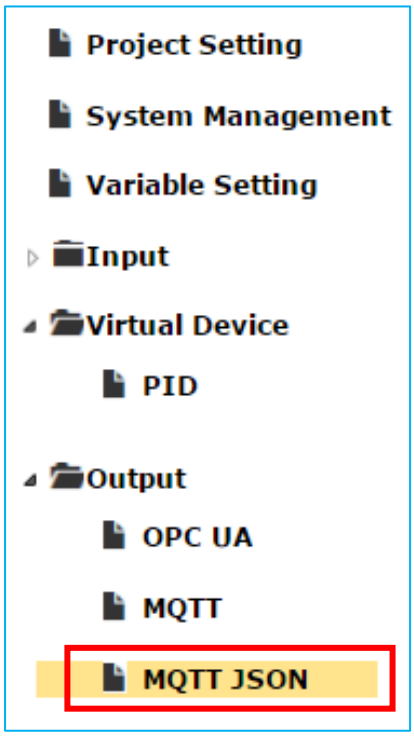
3.6.3. MQTT JSON

This section describes the Output setting of the MQTT JSON communication.

MQTT JSON setting sequence of Output function:

1. Set up the MQTT client to connect to the Broker.
2. Through the group method, map the read/write function of the UA-5200 variables to the Topic messages published and subscribed by the MQTT client devices.
3. Convert the contents of other communication devices to become the contents of the MQTT protocol data.

For the certificate and key about the SSL/TLS communication security, please refer to Chapter 5.



Description of the tool button:

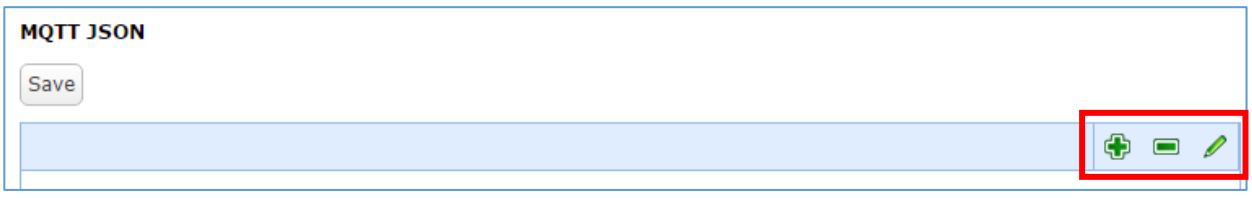
: Add a task, command, or item.

: Delete a task, command, or item.

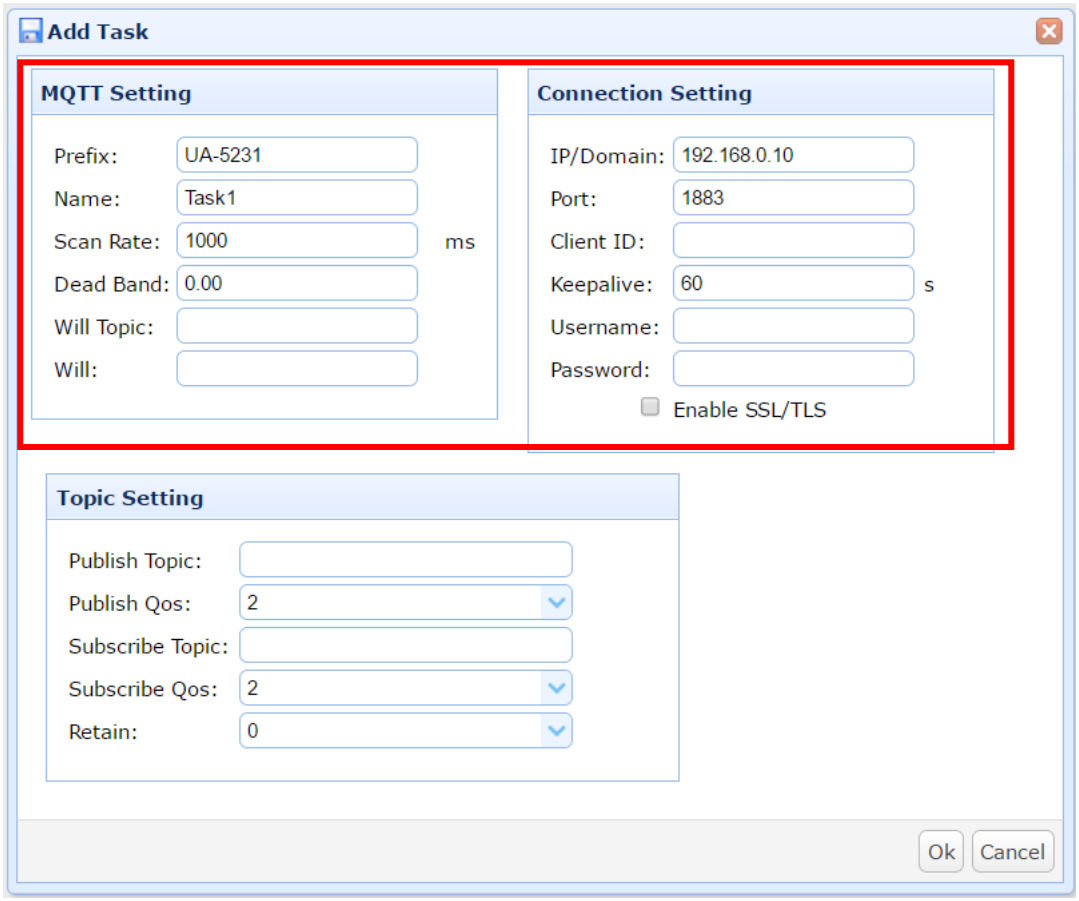
: Modify a task, command, or item.

Save : After setting, click “Save” to save all.

1. On the “Output” -> “MQTT JSON” page, add and edit a Task.

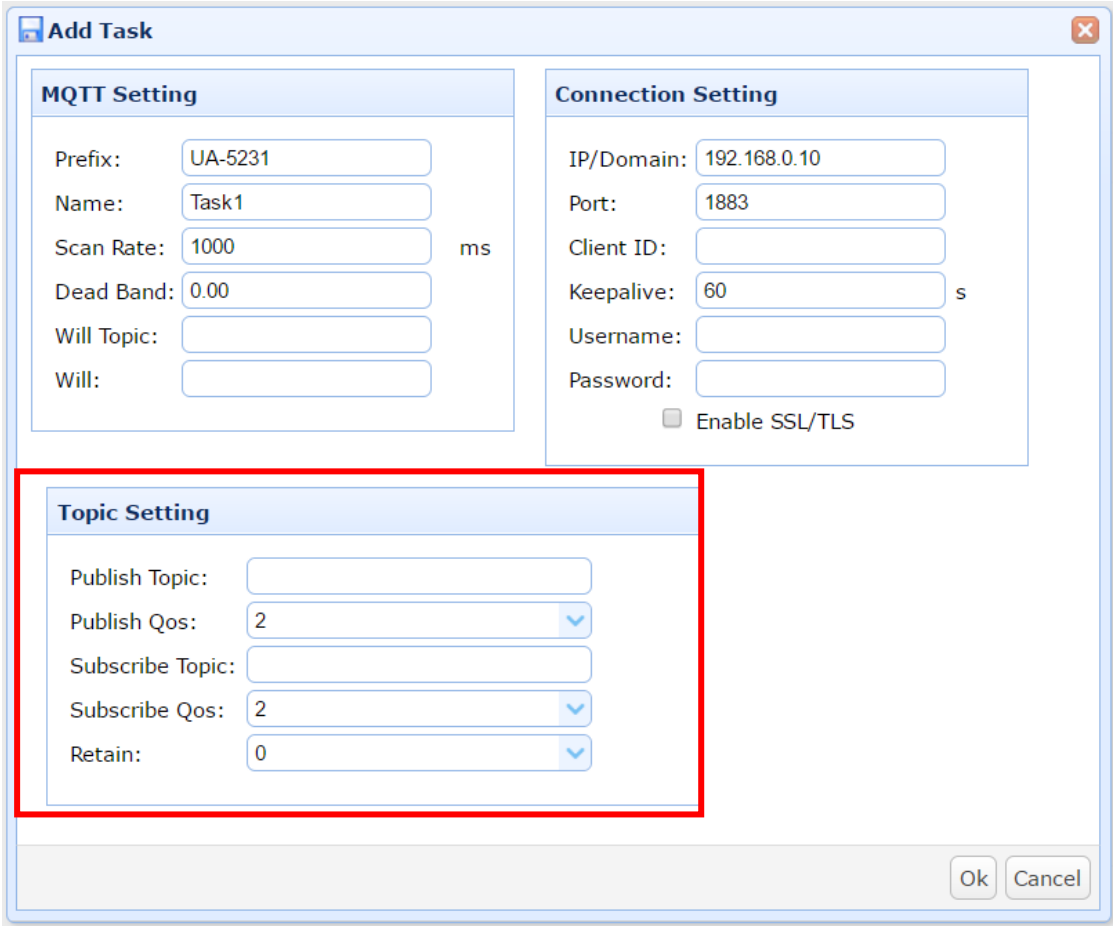


MQTT Setting and **Connection Setting** of MQTT JSON Task:



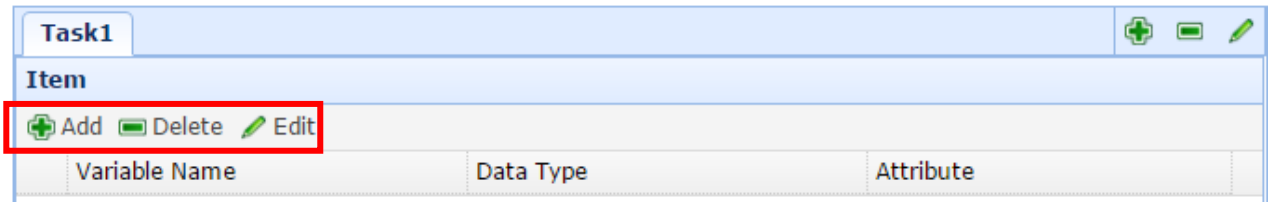
Function Items	Description	Default
MQTT Setting		
Prefix	Set up the prefix for the MQTT topic.	System value
Name	Give a task name.	Task1
Scan Rate	Set an update frequency for the task data. (Unit: ms)	1000
Dead Bend	Give a dead bend value for updating a float signal.	0
Will Topic	Enter the title of a disconnect notice.	Null
Will	Enter a disconnect notice.	Null
Connection Setting		
IP	The IP address of the Broker.	System value
Port	The Broker port.	1883
Client ID	Set the Client ID	
Keepalive	Keep alive time.	60
Username	The user name to login the Broker.	Null
Password	The password to login the Broker.	Null
Enable SSL/TLS	Check to enable the supporting of SSL/TLS security communication.	Uncheck

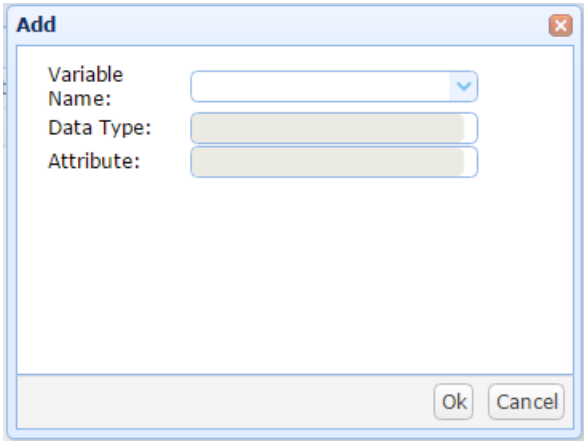
Topic Setting of the MQTT JSON Task:



Function Items	Description	Default
Topic Setting		
Publish Topic	The topic of sending/publishing data message.	
Publish Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Subscribe Topic	The topic of receiving/subscribing data message.	
Subscribe Qos	The publish Qos levels. (QoS: Quality of Service) 0: Delivering a message at most once. 1: Delivering a message at least once. 2: Delivering a message at exactly once.	2
Retain	Whether to store a broker message. (0: No ; 1: Yes)	0

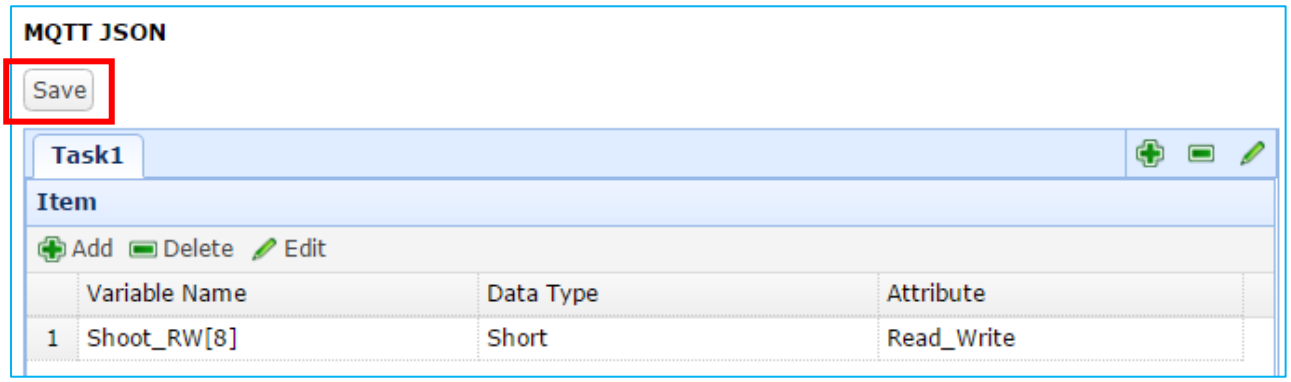
2. Under the MQTT JSON Task, add the **MQTT JSON** Items and assign the variables.





Function Items	Description	Default
Variable Name	Choose a variable which pre-defined in the variable table.	System value
Data Type	Not editable. It will show the data type of a variable.	System value
Attribute	Not editable. It will show the variable attribute.	System value

3. After configuration, the MQTT JSON Output Task table is similar to the following picture. At last, click “Save” to save the current settings



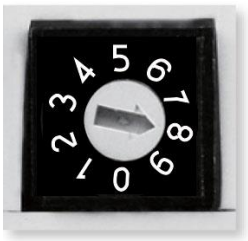
4. Recovering to Factory Setting and Updating Version

In the chapter, we will explain the functions setting by hardware Rotary Switch, including "Recovering to Factory Setting" and "Updating Version" that supported since Version 1.0.0.3.

4.1. Recovering to Factory Setting (Rotary Switch: 8)

Turn the Rotary Switch of UA-5200 series to "8" can recover to the factory setting. The steps:

1. Power off the UA-5200 hardware, and turn the Rotary Switch to "8".

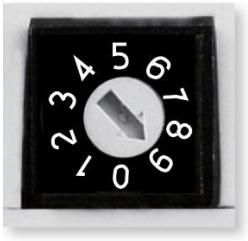


2. Reboot the UA-5200 and wait a long buzzer sound that means of doing the recovering.
3. Wait until two long buzzer sounds, and then turn the Rotary Switch to "0".
   Reboot the UA-5200 again, and the system will recover to the factory settings.

4.2. Updating the Version via USB (Rotary Switch: 9)

Turn the Rotary Switch of UA-5200 series to "9" can update the Middleware version via USB.

1. Power off the UA-5200 hardware, and turn the Rotary Switch to "9".



2. Download the Middleware package file of the UA-5200 hardware corresponding model.
   The download website: ftp.icpdas.com.tw/pub/cd/UA-5000/middleware
3. Save the Middleware package file into an empty USB drive and put to the UA-5200 USB port.
4. Reboot the UA-5200 and wait a long buzzer sound that means of doing the version updating.
5. Wait until two long buzzer sounds, and then turn the Rotary Switch to "0".
6. Reboot the UA-5200 again, and the system will update to the version of the package file.

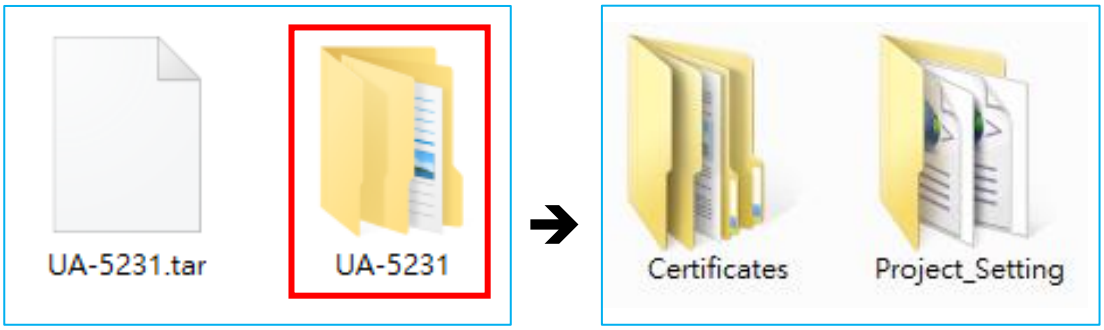
5. Security Certificate and Key Upload to UA-5200

This chapter describes how to upload the trust security certificate and private key to the UA-5200.
First, export the UA-5200 project following the way in 3.1 project setting, and decompress the file.
Next, store the certificate and private key to the specified folder of the project file.
Finally, re-compress and import the project file back to the UA-5200 device.
It is about the OPC UA and MQTT of UA-5200 setting for the SSL/TLS certificate and private key.

5.1. Export and Decompress the UA-5200 Project

1. Use the "**Project Setting**" -> "**Export"** function to save the UA-5200 project file (***.tar**).

2. Use compression software (e.g. 7.zip) to decompress the UA-5200 project to a same name folder (e.g. UA-5231 folder). There are two main sub-folders as below.



5.2. Store Trusted Root Certificate and Private Key

5.2.1. OPC UA

The user can store trusted root certificates and private key from the OPC UA client into the specific folder of UA-5200 project for setting up security communications.
The file name, path and format are as follows:

1. Store the **OPC UA** root certificates and private key to the following folder of the project.
   /Certificates/pkiserver/trusted/certs

2. The supported name is "***.der"**. The sub-file name must be "**.der**", while the file name can follow the user need.

3. The supported code format is "**DER**".

5.2.2. MQTT Client

The user can store trusted root certificates and private key from the MQTT Broker into the specific folder of UA-5200 project for setting up security communications.
The file name, path and format are as follows:

1. Store the **MQTT Broker** root certificates and private key to the following folder of the project.
   /Certificates/mqtt/client/trusted/certs

2. The supported name is "**ca.crt"**. The file name must be "**ca**", and the sub-file name must be "**.crt**".

3. The supported code format is "**PEM**".

5.3. Recompress and Import the Project

After storing the trusted root certificates and private key to the specific folder, recompress the project to the same name as export.
At last, use the "**Project Setting**" -> "**Import**" function to upload the project file back to the UA-5200 device.

Appendix A. MQTT JSON Format of the UA-5200 Series

MQTT JSON Example & Format Descriptions:

```
{
   "Variable" : [ {
      "Name" : "Bool_R[0]",
      "Attribute" : "R",
      "Datatype" : "Bool",
      "Value" : 0,
      "Quality" : "Uncertain"
   }, {
      "Name" : "Short_R[0]",
      "Attribute" : "R",
      "Datatype" : "Int16",
      "Value" : 0,
      "Quality" : "Uncertain"
   }, {
      "Name" : "Short_R[1]",
      "Attribute" : "R",
      "Datatype" : "Int16",
      "Value" : 0,
      "Quality" : "Uncertain"
   }, {
      "Name" : "Short_R[2]",
      "Attribute" : "R",
      "Datatype" : "Int16",
      "Value" : 0,
      "Quality" : "Uncertain"
   }, {
      "Name" : "Short_RW[2]",
      "Attribute" : "RW",
      "Datatype" : "Int16",
      "Value" : 0,
      "Quality" : "Uncertain"
   } ]
}
```

Name	Descriptions
Variable	The array name of JSON. Its structure includes several member data as below.
Name	The member name of the array element
Attribute	The member attribute of the array element: “R” : can read “W” : can write ”RW” : can read and write
Datatype	The member’s data type of the array element: "Bool" "Int8" "UInt8" "UInt16" "Int16" "UInt32" "Int32" "UInt64" "Int64" "Float" "Double" "String"
Value	The member’s current value of the array element
Quality	The member’s current status of the array element: "Uncertain" "Good" "Bad"

Appendix B. Technical Reference Websites

- OPC UA

https://opcfoundation.org/

- MQTT

http://mqtt.org/

- Modbus

http://modbus.org/