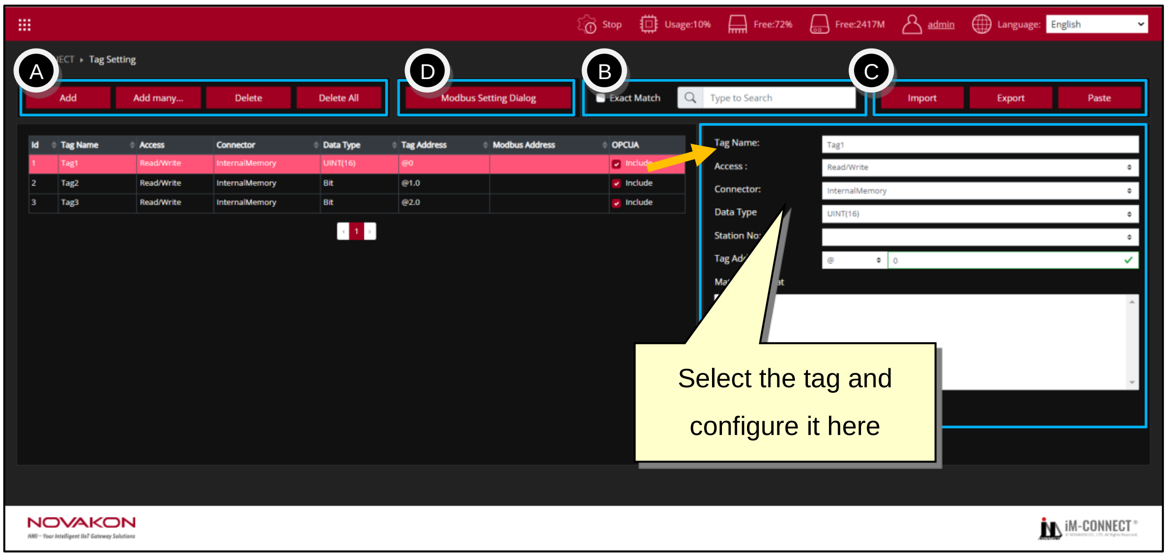
1. Tag Setting

Tag is closely related to each service, and almost all functions need well-defined tags. The way of using a tag is different for each function; however, setting a tag is quite similar. There are many benefits of using tags. One of them is to allow users to effectively summarize and organize the registers used by the controller.

Each ID represents a tag, which can correspond to the register address of a PLC or controller. If the registration number is not set, an error message would appear when entering the registered address and compiling.



Id	Tag Name	Access	Connector	Data Type	Tag Address	Modbus Address	OPCUA
1	Tag1	Read/Write	InternalMemory	UINT(16)	@0		☑ Include
2	Tag2	Read/Write	InternalMemory	Bit	@1.0		☑ Include
3	Tag3	Read/Write	InternalMemory	Bit	@2.0		☑ Include

Tag Name: Tag1
Access : Read/Write
Connector: InternalMemory
Data Type UINT(16)
Station No:
Tag Ad @ 0

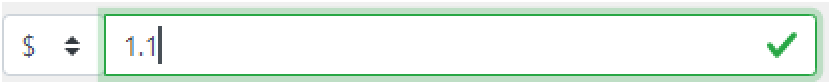
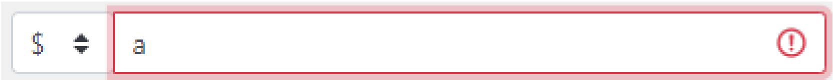
Select the tag and configure it here

A. Tag operation:

Add

One tag can be added at a time. After pressing the [Add] button, the complete content of the tag can be created in sequence.

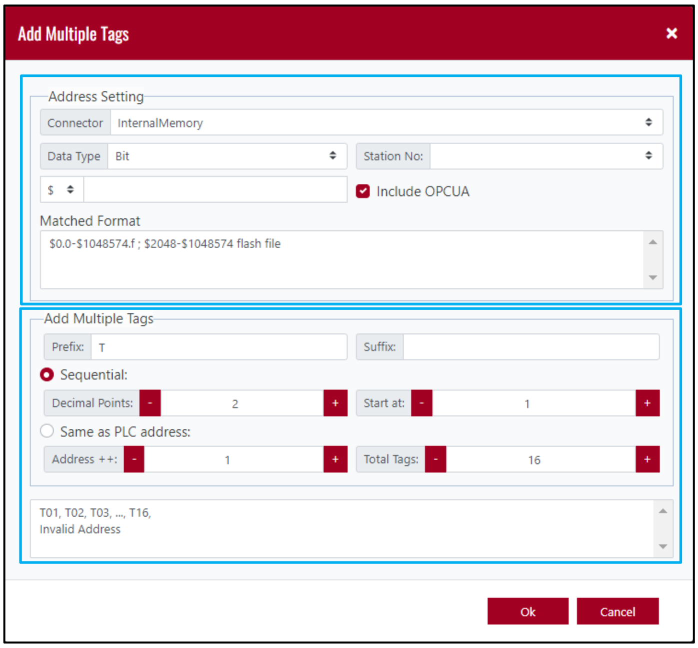
Tag_Name	Each tag can be assigned a tag name. The name supports multi-language input, and the name in English is case-sensitive (for

	example, Apple and apple represent two different tag names). The tag name must not be duplicated.
Access	Set the tag to be[Read/Write]or[Write Only].
Connector	Select the controller or the InternalMemory of Gateway to be used.
Data Type	Select the data type, including[Bit],[INT(8)],[UINT(8)], [INT(16)],[UINT(16)],[INT(32)], [UINT(32)],[FLOAT],[Ascii], [DOUBLE],[Ascii],[INT(64)],[UINT(64)]
Station No	The station number of the controller to be used. Some controllers have different station numbers, such as thermostats that use the Modbus protocol.
Modbus Address	The value of this address can be copied to the specified Modbus address.
Tag Address	The address of the register. If the format or range of the register is unknown, please refer to the prompt in the [Matched Format] window below, which display such information for the input of [Tag Address]. The figure below shows that a green frame would appear if the entered address meets the format.  $ 1.1 ✓ The figure below shows that a red frame would appear if the entered address does not meet the format.  $ a ⓘ If the selected connection is [InternalMemory], it means that no external controller is connected, and only the internal register provided by GATEWAY is used. There are currently two register

	codes that can be used: [$]:This code represents the memory retaining with power off. When GATEWAY is restarted, the original data is still saved. The range is from $0 to $1048574, of which $2048 to $1048574 are stored in Flash. [@]:This code represents the memory not retaining with power off. When GATEWAY is restarted, the original data is not saved. The range is from @0 ~ @65535. The above-mentioned two register codes can be used without actual connection to external controller as long as the setting conforms to the data type format. For example, when @0 (16-bit) is presented as a bit type, the expression is @0.0 ~ @0.f. It can be understood that it is a squad of the troop. The left side of the decimal (integer place) means the number of squads, and the right side of the decimal (decimal place) is the squad number in the squad.

Add many…

Multiple consecutive tags can be added at one time.



Add Multiple Tags

Address Setting

Connector: InternalMemory

Data Type: Bit

Station No:

$

☑ Include OPCUA

Matched Format

$0.0-$1048574.f ; $2048-$1048574 flash file

Add Multiple Tags

Prefix: T

Suffix:

● Sequential:

Decimal Points: 2

Start at: 1

○ Same as PLC address:

Address ++: 1

Total Tags: 16

T01, T02, T03, ..., T16,
Invalid Address

Address Setting

Selection of Connector, Data type, Station No, and address.

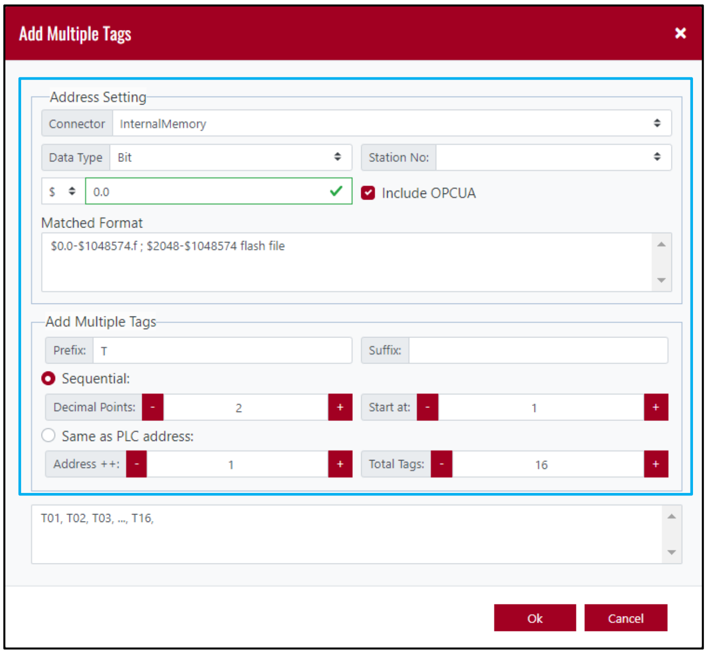
Connector	Selection of the controller to be used when adding new addresses.
Data Type	Selection of the type of data to be used when adding new addresses.
Station No	Selection of the station number of the controller to be used when adding new addresses.
Address	Enter the starting register address of the new tag.
Matched Format	This displays the format of the register address that the user can reference.

Add Multiple Tags

Set the format of add multiple tags.

Prefix	Set the prefix of tag name.
Suffix	Set the suffix of tag name.
Sequential	The setting of custom prefixes and suffixes for the names of

	consecutively added multiple tags...
Decimal Points	The setting of the number of digits in the middle of the prefix and suffix.
Start at	The starting number of [Decimal Points].
Address ++	The number of increments of the register number in each step.
Total Tags	The total number of tags to be added.
Same as PLC address	Check this box, and the new tag name would be the same as the address of the PLC register.



Address Setting

Connector: InternalMemory

Data Type: Bit

Station No:

$ 0.0

☑ Include OPCUA

Matched Format

$0.0-$1048574.f ; $2048-$1048574 flash file

Add Multiple Tags

Prefix: T

Suffix:

● Sequential:

Decimal Points: - 2 +

Start at: - 1 +

○ Same as PLC address:

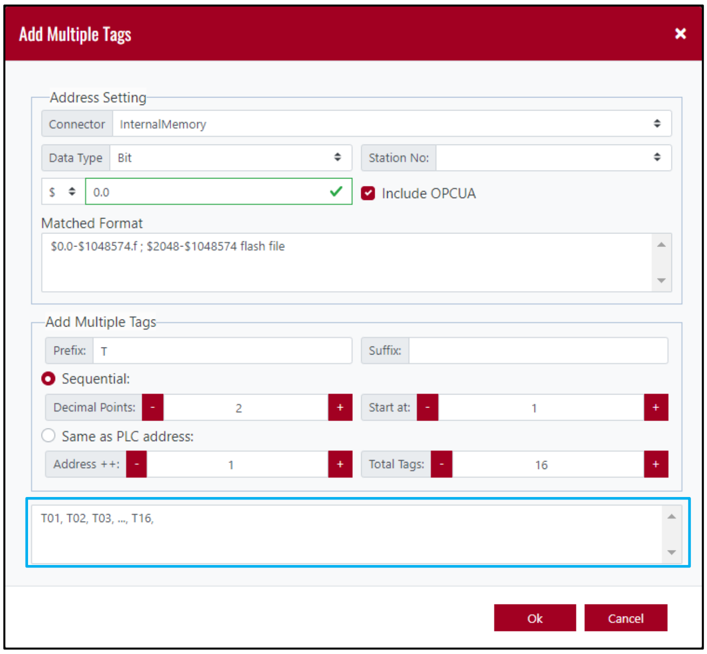
Address ++: - 1 +

Total Tags: - 16 +

T01, T02, T03, ..., T16,

For example, in the figure shown above, the controller is set a [InternalMemory](internal memory), and the data type is [Bit], the address is [$0.0]. The prefix is set to [T], the suffix is not set, the number of digits is [2], and the initial value is [1], the address increment is [1], and the number of tags is [16].

When [Continuous] is selected, the tag name is set in the sequence of T01, T02, T03,..., T16, and the corresponding address is $0.0,$0.1,$0.2,...,$0.f.



Add Multiple Tags

Address Setting

Connector: InternalMemory

Data Type: Bit

Station No:

$ 0.0

☑ Include OPCUA

Matched Format

$0.0-$1048574.f ; $2048-$1048574 flash file

Add Multiple Tags

Prefix: T

Suffix:

● Sequential:

Decimal Points: 2

Start at: 1

○ Same as PLC address:

Address ++: 1

Total Tags: 16

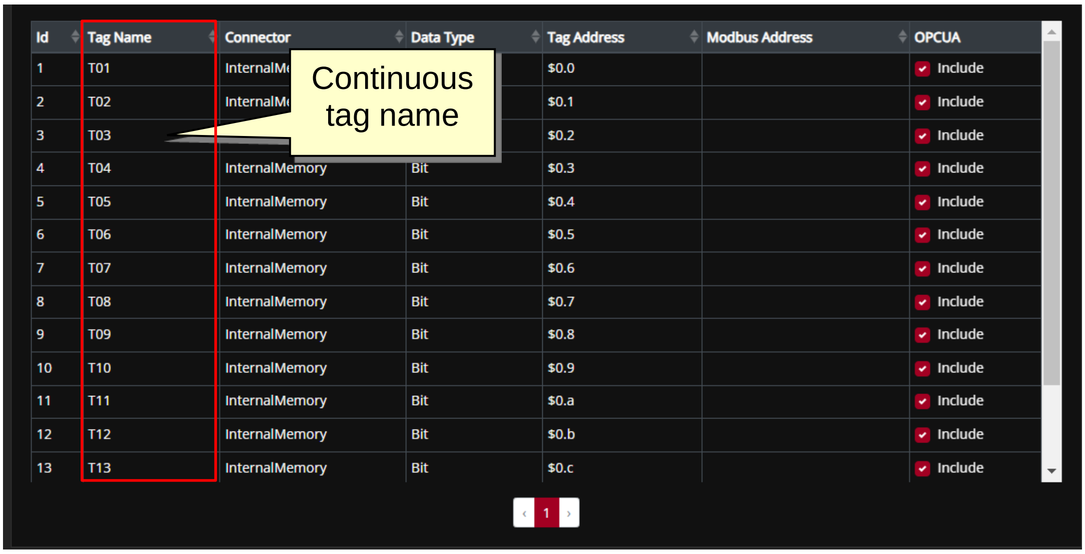
T01, T02, T03, ..., T16,

According to the new tag content set, the added tag name would be generated immediately to provide a preview at the bottom of the dialog box. If the setting is wrong, the preview window prompts the message of [Invalid Address].

Click [OK] after completion, and GATEWAY would automatically add consecutive tags.

Note: All objects in the edit correspond to the tag name, not the address of the controller.

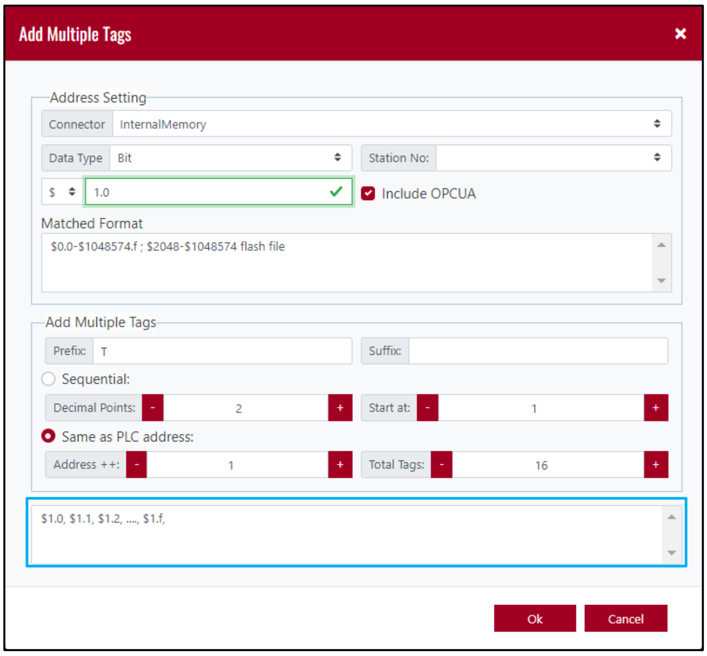
The following figure shows the content of the tag created when [Continuous] is selected.



Id	Tag Name	Connector	Data Type	Tag Address	Modbus Address	OPCUA
1	T01	InternalM		$0.0		☑ Include
2	T02	InternalM		$0.1		☑ Include
3	T03			$0.2		☑ Include
4	T04	InternalMemory	Bit	$0.3		☑ Include
5	T05	InternalMemory	Bit	$0.4		☑ Include
6	T06	InternalMemory	Bit	$0.5		☑ Include
7	T07	InternalMemory	Bit	$0.6		☑ Include
8	T08	InternalMemory	Bit	$0.7		☑ Include
9	T09	InternalMemory	Bit	$0.8		☑ Include
10	T10	InternalMemory	Bit	$0.9		☑ Include
11	T11	InternalMemory	Bit	$0.a		☑ Include
12	T12	InternalMemory	Bit	$0.b		☑ Include
13	T13	InternalMemory	Bit	$0.c		☑ Include

Continuous tag name

If [Same as PLC address] is selected, the tag name would be the same as the PLC address, as shown in the figure below.



Add Multiple Tags

Address Setting

Connector: InternalMemory

Data Type: Bit

Station No:

$ 1.0 ✓

☑ Include OPCUA

Matched Format

$0.0-$1048574.f ; $2048-$1048574 flash file

Add Multiple Tags

Prefix: T

Suffix:

○ Sequential:

Decimal Points: - 2 +

Start at: - 1 +

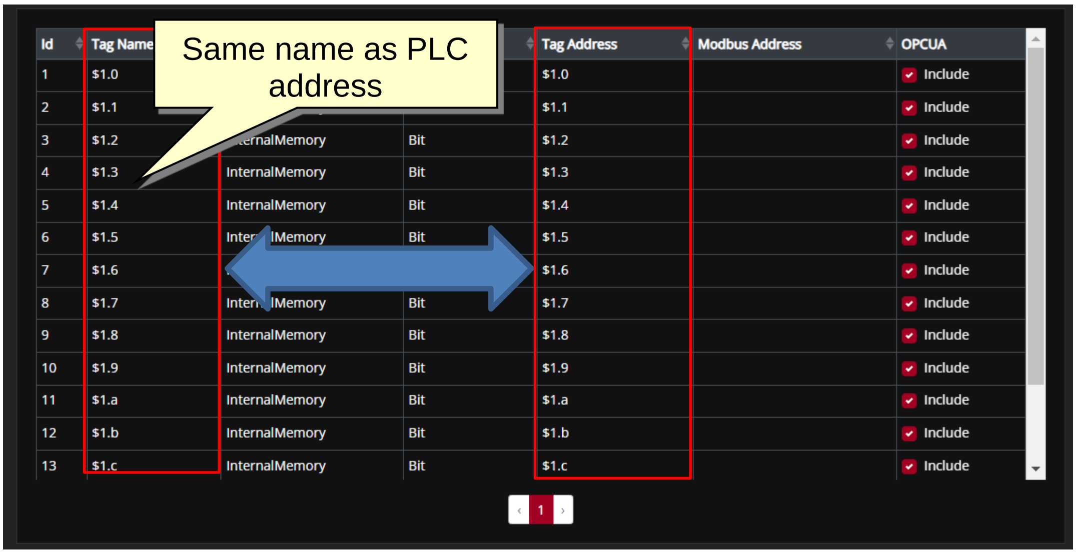
● Same as PLC address:

Address ++: - 1 +

Total Tags: - 16 +

$1.0, $1.1, $1.2, ..., $1.f,

The following figure shows the created tag.



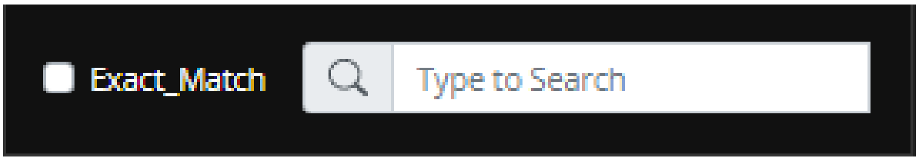
Same name as PLC address

Id	Tag Name			Tag Address	Modbus Address	OPCUA
1	$1.0			$1.0		☑ Include
2	$1.1			$1.1		☑ Include
3	$1.2	ernalMemory	Bit	$1.2		☑ Include
4	$1.3	InternalMemory	Bit	$1.3		☑ Include
5	$1.4	InternalMemory	Bit	$1.4		☑ Include
6	$1.5	Inter lMemory	Bit	$1.5		☑ Include
7	$1.6			$1.6		☑ Include
8	$1.7	Inter lMemory	Bit	$1.7		☑ Include
9	$1.8	InternalMemory	Bit	$1.8		☑ Include
10	$1.9	InternalMemory	Bit	$1.9		☑ Include
11	$1.a	InternalMemory	Bit	$1.a		☑ Include
12	$1.b	InternalMemory	Bit	$1.b		☑ Include
13	$1.c	InternalMemory	Bit	$1.c		☑ Include

Delete	Click this button to delete a tag(s). Move the cursor to the leftmost serial number column, press, hold and drag the mouse's left button to select multiple tags, and click [Delete] to delete multiple tags simultaneously.
Delete All	Click this button to delete all the tags.

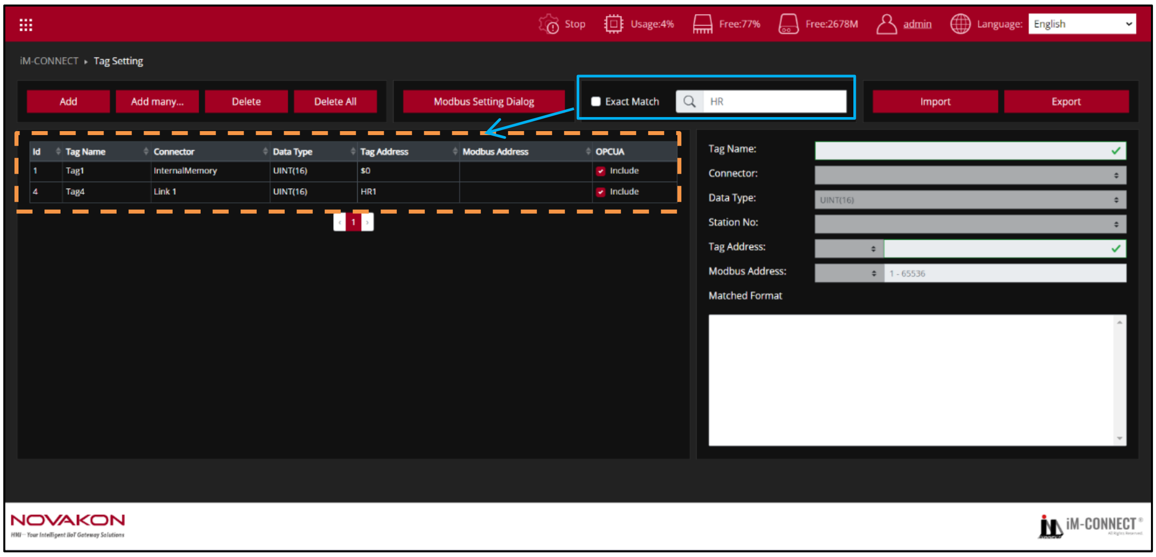
B. Search

This is used for searching for tags that are the same as the keywords according to the input content, including tag name, type, address, etc.



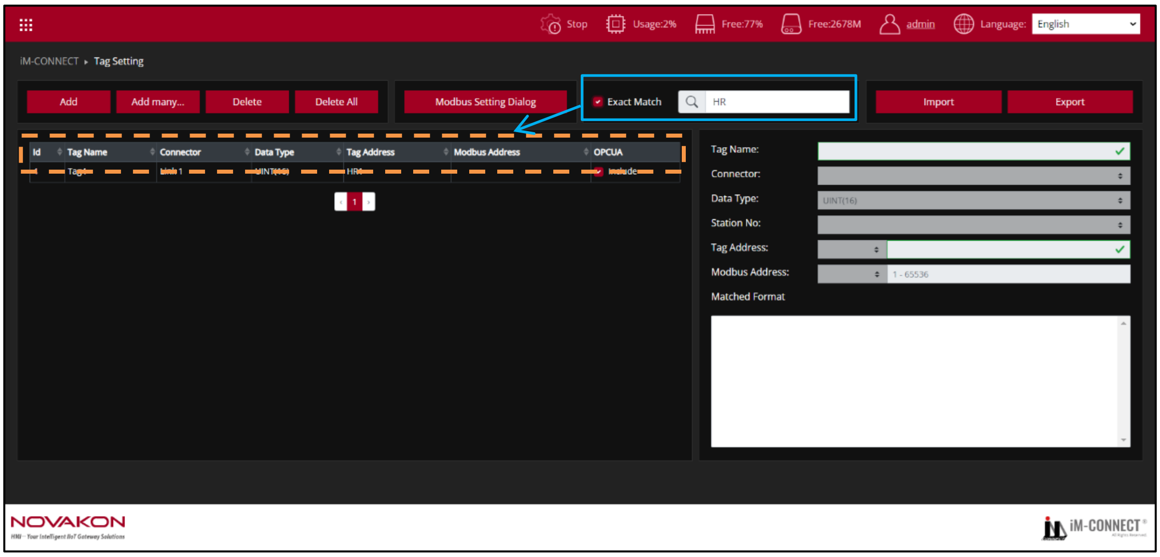
GATEWAY would automatically help us search for all matching tag content when we enter the text to be searched.

For example, after entering the search [HR] in the figure below, all tags containing [H] or [R] would be displayed, regardless of the order in which they are arranged.



Id	Tag Name	Connector	Data Type	Tag Address	Modbus Address	OPCUA
1	Tag1	InternalMemory	UINT(16)	$0		✓ Include
4	Tag4	Link 1	UINT(16)	HR1		✓ Include

But when [Exact_Match]is checked, GATEWAY would automatically list all the tag names that meet the [HR] conditions, and the arrangement position must be correct.



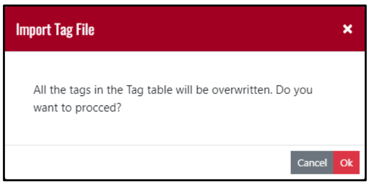
Id	Tag Name	Connector	Data Type	Tag Address	Modbus Address	OPCUA
4	Tag4	Link 1	UINT(16)	HR1		✓ Include

C. Import/Export

Export is to output the data content of the tag into a text file (default as fileName.TXT).

Import is to read the text file (*xml, *njnx, *.TXT) back into the tag content.

Click the [Import] button, and a confirmation window as shown in the figure below would pop up.



All the tags in the Tag table will be overwritten. Do you want to procced?

Click [YES] to confirm to execute the action of importing tags, select the file you want to import, and then press [Open].

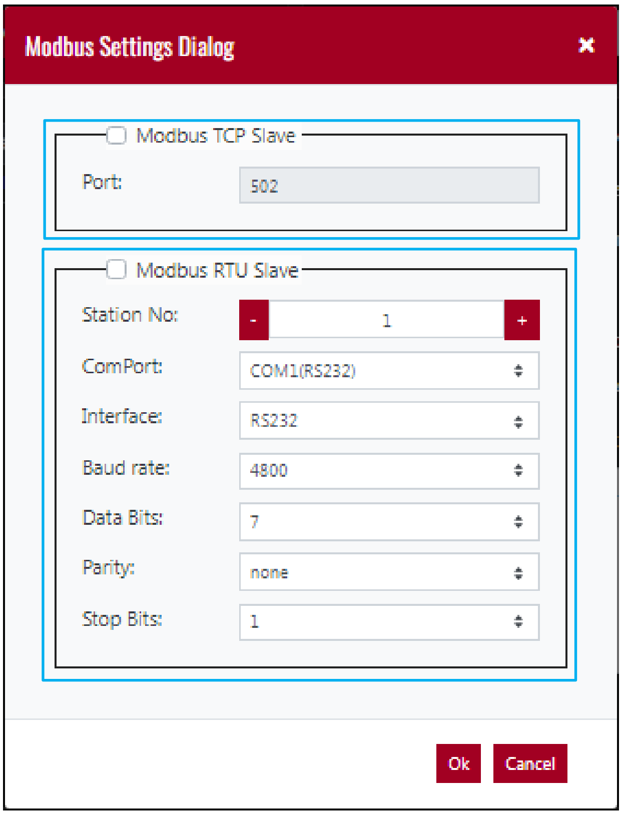
Click [NO] to exit from executing the action of importing tag.

Click the [Output] button, the system will output the data content of the tags into a text file (*.TXT) as backup or for user to edit then import later.

D. Modbus Setting Dialog

Set GATEWAY as Modbus slave.

Note:When setting the Modbus slave controller in [PLC Setting], it means GATEWAY itself is Modbus Master, and the controller is Modbus slave.



☐ Modbus TCP Slave

Port: 502

☐ Modbus RTU Slave

Station No: 1

ComPort: COM1(RS232)

Interface: RS232

Baud rate: 4800

Data Bits: 7

Parity: none

Stop Bits: 1

Modbus TCP Slave	Check the box to activate GATEWAY as[Modbus TCP Slave].
Modbus RTU Slave	Check the box to activate GATEWAY as[Modbus RTU Slave].

Test Procedure: (Reference files:TAG.dat)

1. Select PLC Setting: In the Menu, choose PLC Setting.
2. Add LinK: Click on "Add Link" to select the desired Port and input the necessary information for the PLC Vendor, Model, and Communication Format.
3. Select Tag Setting: Go back to the Menu and choose Tag Setting.
4. Press "Add": Create a new tag by clicking on "Add".
5. Enter Tag Details: Fill in the fields for Tag Name, Connector, Data Type, and Tag Address.
6. Create Additional Tags: Continue to create any additional Tags as needed. You can also use the "Add many..." option or "Import" feature to create multiple Tags at once.
7. Save and Compile: Go to the Menu, select PROJECT SETTING, and execute "Save and Compile" to compile the project.
8. Start the Project: Click "Start Project" to run the project.

9. Monitor the execution: Go back to the Menu, select ONLINE MONITOR, and choose the Tags you are using to monitor the Macro’s execution results.