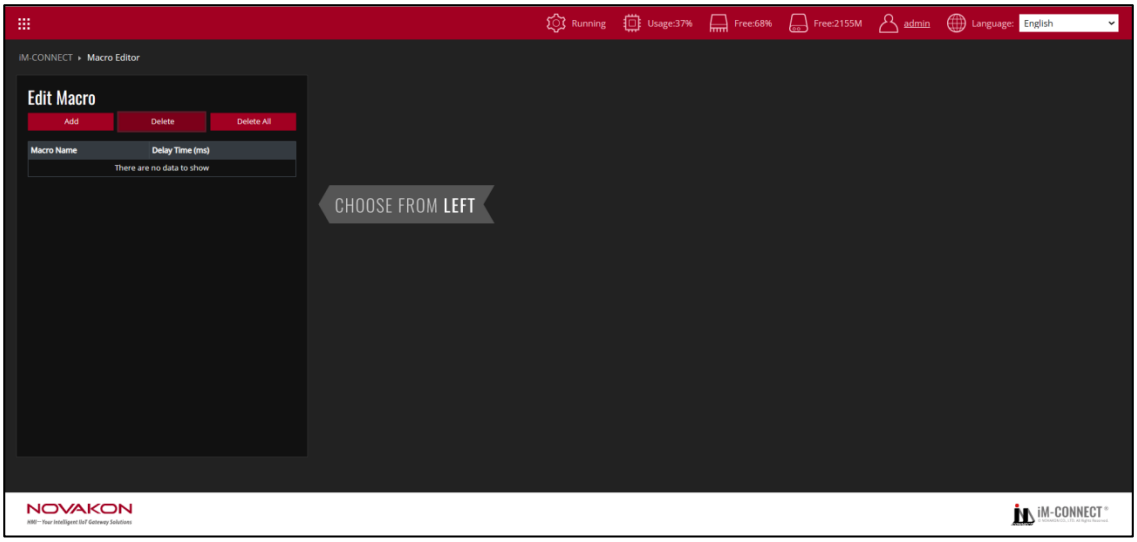
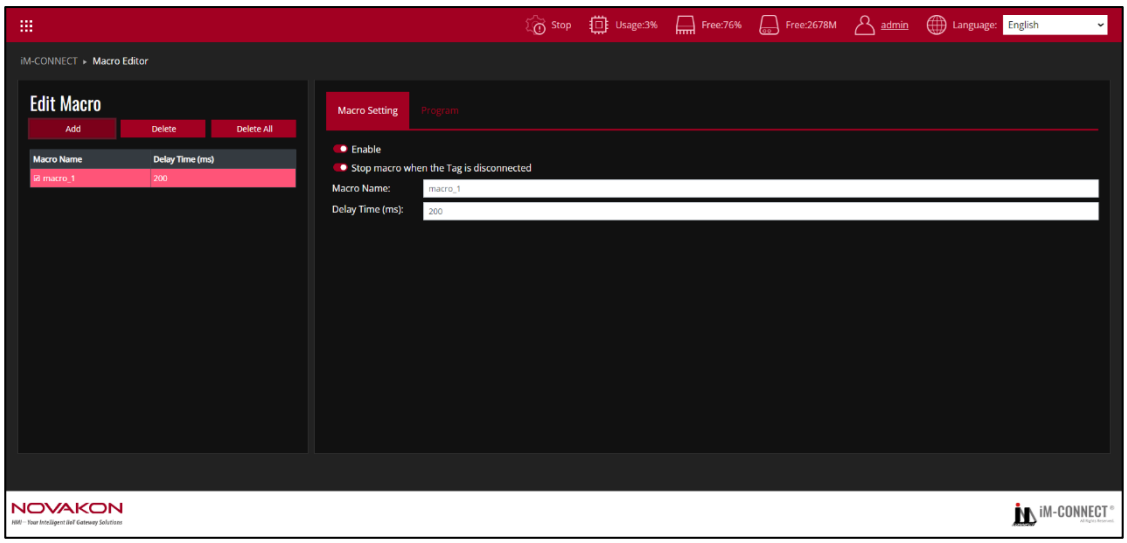
Edit Macro

Edit Marco can perform arithmetic operation, data transfer , and other actions for the Gateway’s tag.



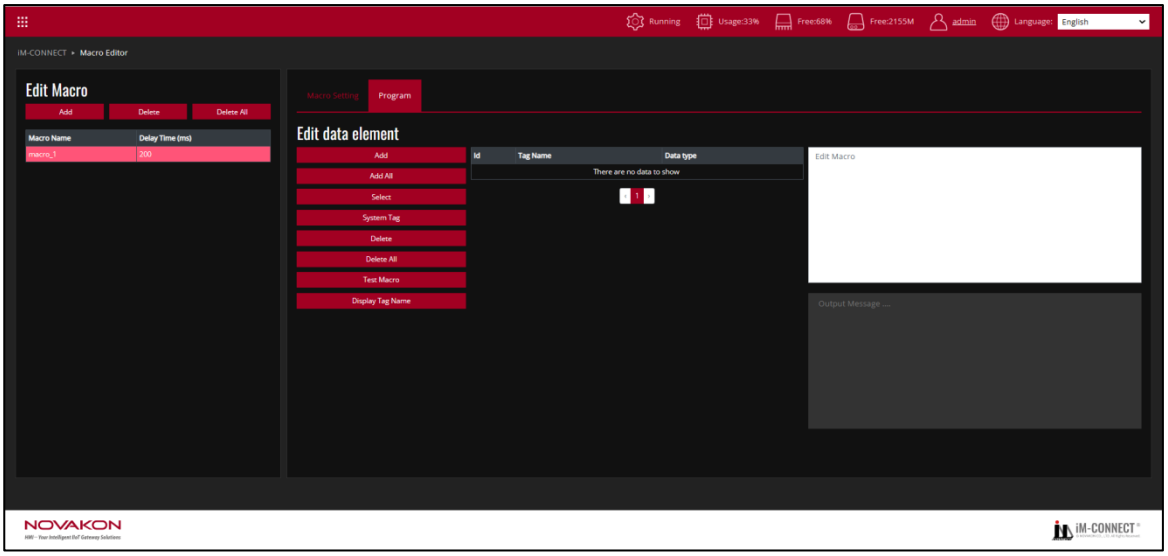
Add	Add a set of macro programming.
Delete	Delete the selected macro programming.
Delete All	Delete all the macros.

Start writing a set of macro programming after selecting[Add].

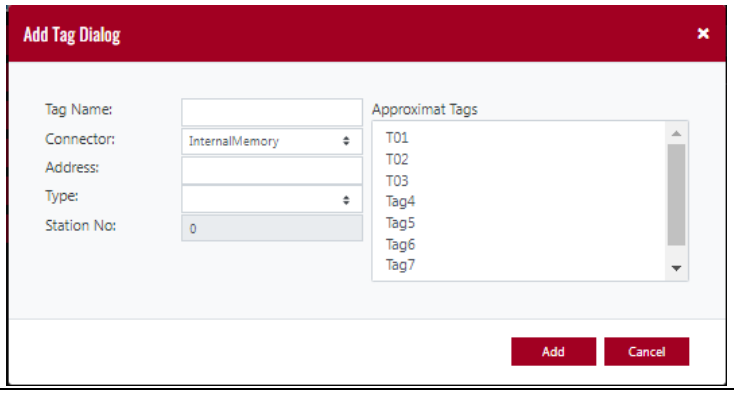


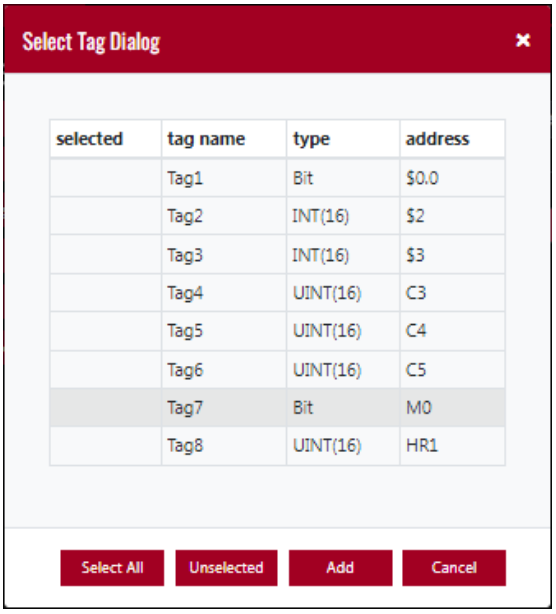
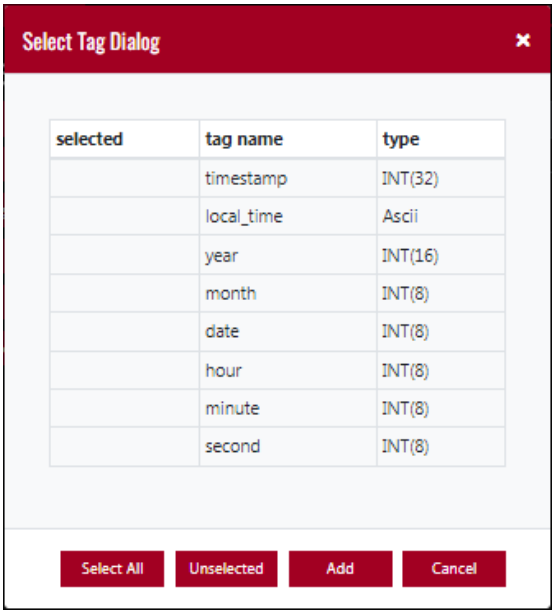
Macro Setting

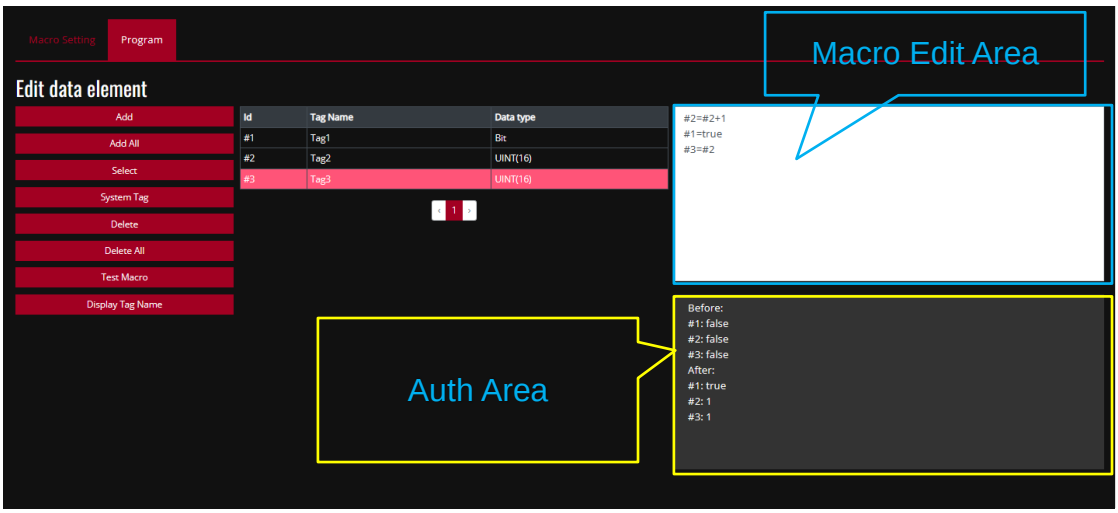
Enable	Checking this box enables the execution of the macro program.
Stop macro when the Tag is disconnected	Checking this box causes the entire macro to stop running if the tag disconnects during its execution; otherwise, the macro will continue to execute.
Macro Name	Set the name of the set of macro programming.
Time (Secs)	Set how long it delays after a regular loop of macro programming complete before starting the next loop.



Edit data element

Add	Add the tag(s) needed in the macro programming. The newly added tag(s) will also be set in the Tag Setting page. 
Add All	Add all the tags existing in the project.
Select	Select the tag(s) needed in the Macro Programming from the list of tags.

	
System Tag	Select the system tag(s) required in the macro programming from the list of tags. 
Delete	Delete the selected set of macro programming.
Delete all	Delete all the macro programming.
Test Macro	Verify the results of the macro programming performance test.
Display Tag Name	Change the display from ID number to Tag Name in Macro Edit Area. (Marco Edit Area becomes read only when it is changed to display Tag Name)



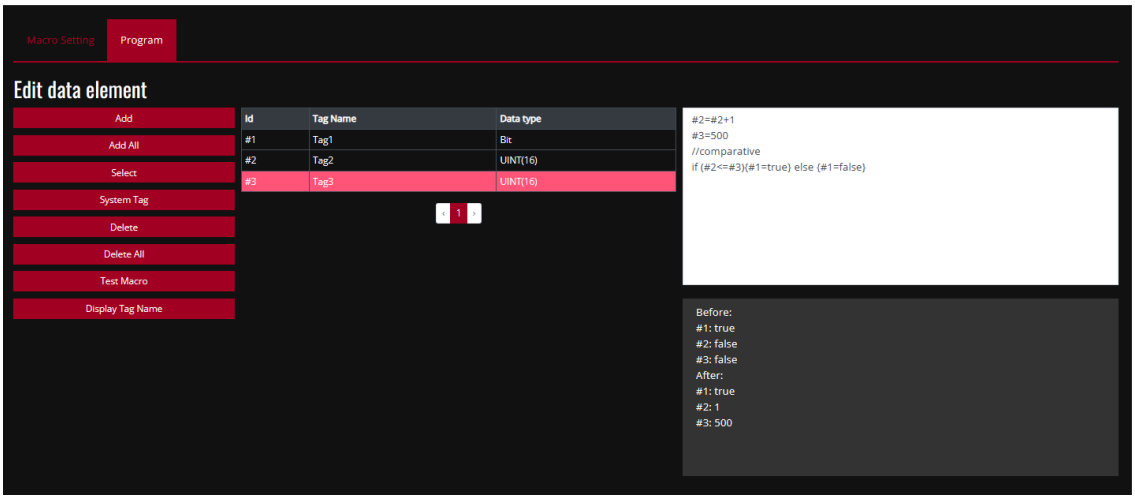
Macro Edit Area	The user can write the macro programming here.
Auth Area	The results from the macro programming performance can be learned in the Auth Area after the[Test Macro]is selected.

For example:As shown in the diagram above, it can edit simple expressions. For the variables (tags), the corresponding ID must be input. At the same time, if the data type is Bit (Boolean), the condition must be expressed in [true](lower case) or[false](lower case).

To add Notes, express it with [//].

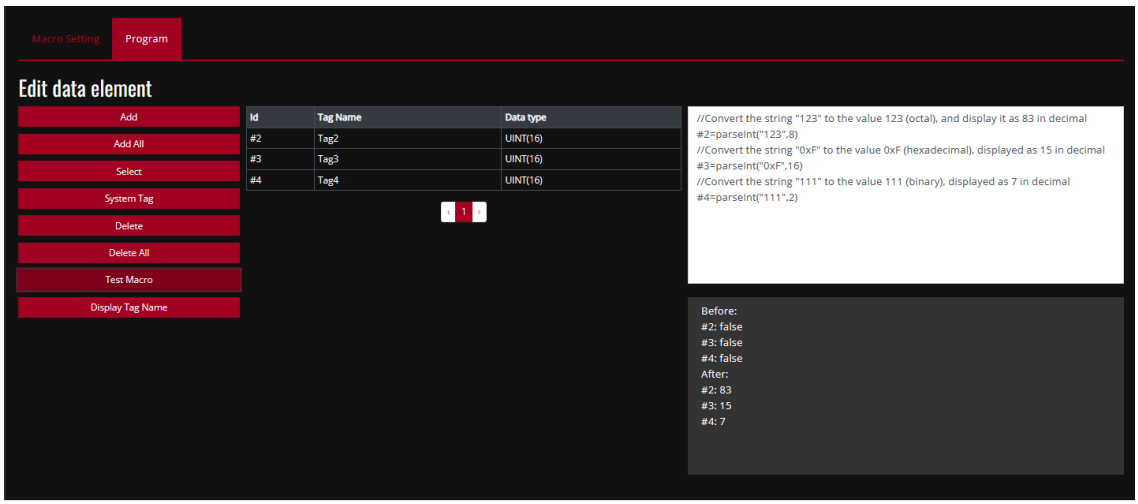
To add conditional execution, express it with [if (conditional execution body){Results executed after the conditional execution statement is set up } else {Results executed after the conditional execution statement is not set up }]

For example:



iM-connect can support elements of JavaScript, such as Number, Date, Math, etc.

The following is an example of base conversion by using JavaScript function [parseInt]:



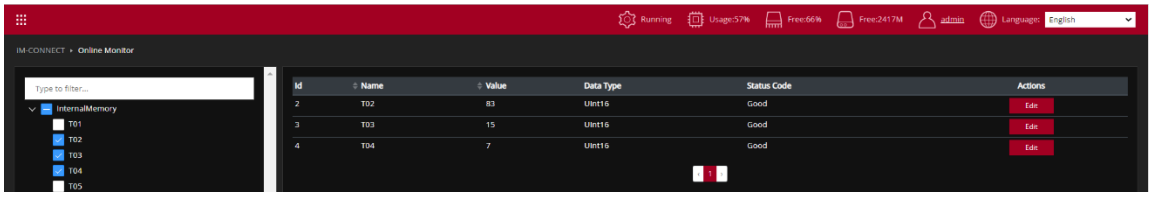
① [Add] or [Select] the needed tag.

② Write the program.

③ Go to[Project Setting]to execute[Save & Compile] and [Enable Service].

④ After selecting [Test Macro], you get the correct result.

⑤ Enter [Online Monitor] to monitor the actual tag values. The results must be consistent with the results of Test Macro.



Test Procedure: (Reference files:MACRO.dat)

1. In the Menu, choose MACRO EDITOR.
2. Press "Add": Create a new Macro.
3. Enable the Macro: Go to the Macro Setting Tab and click "Enable" to activate the Macro.
4. Optional: Enable "Stop Macro when the Tag is disconnected": Select this option if you want the Macro to stop when the associated Tag is disconnected.
5. Enter Macro Name: Provide a name for the Macro in the Macro Name field.
6. Set Delay Time: Configure the delay time for the Macro, with the unit in milliseconds (ms).
7. Switch to Program Tab: Add the required Tag or System Tag.
8. Write the Macro: Based on basic JavaScript syntax, input the Macro script that you wish to execute.
9. Test the Macro: Click on "Test Macro" to simulate and view the execution result of the Macro.
10. Save and Compile: Go to the Menu, select PROJECT SETTING, and execute "Save and Compile" to compile the project.

11. Start the Project: Click "Start Project" to run the project.
12. 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.