## FAQ-APP-01: UA Web UI Function Wizard – APP Message Notify - How to set up APP Message Notify function: IFTTT Condition Trigger (Line, Twitter) ? (Use M-7055D)

The "**IFTTT Condition Trigger (Line, Twitter)**" combines the functions of the **UA and IFTTT cloud platform**. When the modules occur the special events that setting in the UA condition, it will trigger the IFTTT and send the message to the IFTTT-related cloud services (such as Line, Twitter, etc.)

The settings for sending the message to the APP with the "IFTTT Condition Trigger (Line, Twitter)" function includes two parts:

● **IFTTT Cloud Platform Setting: (It must be set before setting up the UA project)**

In the IFTTT website, set up the "**if**" side service and event (**this**: use **webhooks** for the UA), the "**then**" side service and action (**that**: user can select the service, such as the Line, twitter, etc.). And then fill the "**Event Name**" and "**Key**" getting from the IFTTT website setting into the "**Content Setting**" of the UA We HMI. Please refer **FAQ-app-02:**
http://www.icpdas.com/web/product/download/iiot/ua//faq/FAQ-en-app-02.pdf



● **UA Web Interface Setting: (**Set up via Advanced Setting > IFTTT Condition Trigger**)**

In the UA Web HMI, set up the UA controller, modules, IFTTT trigger conditions, the condition variable table, and the IFTTT event connection. (Fill the IFTTT **Event Name** and authentication **Key** in the IFTTT step of the project into the "**Content Setting**" of the UA web interface)

**[Step Box]:**
The Step Box of the [**IFTTT Condition Trigger (Line, Twitter)**] has 6 steps as below. When enabling the Step Box, it auto enters the first step setting page (The step with a bold underline means it is the current step.). The user just needs to follow the "Step Box" step-by-step and then can complete the project.
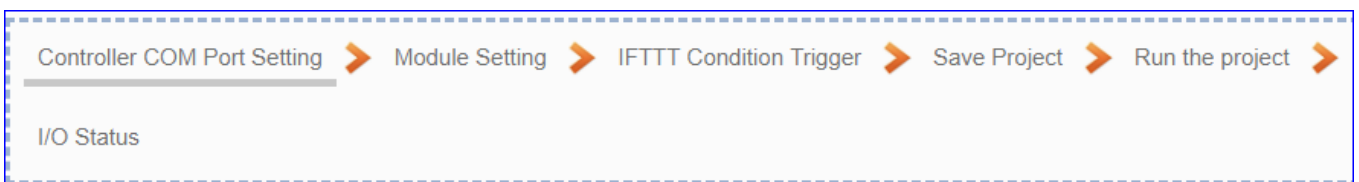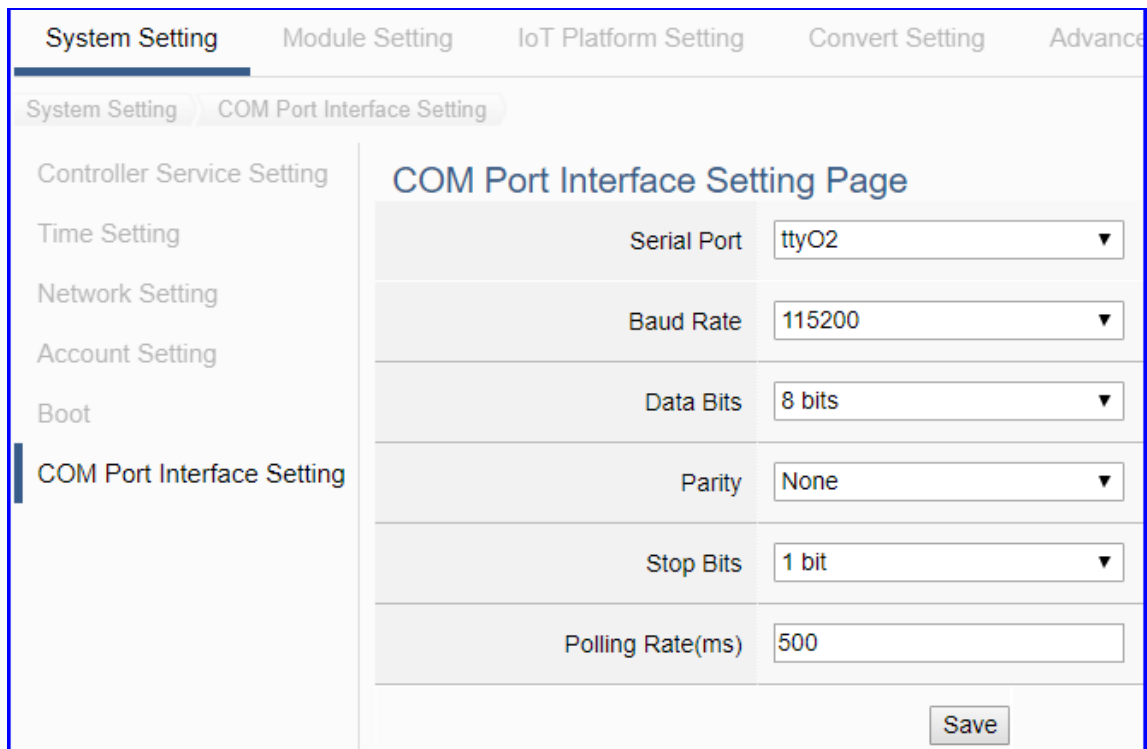
● **Step 1. Controller COM Port Setting**



This page allows display and set the COM port interface of the controller for the serial communication. The user can find the default communication values of our I/O modules from the module CD, manual or I/O Module website.



| COM Port Interface Setting Page | |
|---|---|
| Serial Port | Choose the serial port of UA controller that links with the I/O module. ttyO2: RS-485 ; ttyO4: RS-232 ; ttyO5: RS-485 |
| Baud Rate | Choose a baud rate to communicate with the module: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200. The UA controller and the I/O module need have the same baud rate. |
| Data Bits | The number of bits used to represent one byte of data: 7 bits or 8 bits. Default: 8 Bits. |
| Parity | Choose one way for the parity checking. Options: None, Even, and Odd. Default: None. |
| Stop Bits | Choose the number of stop bit: 1 bit or 2 bits. Default: 1. |
| Polling Rate(ms) | Set a time interval for the command. Default: 500 ms |
| Save | Click [Save] button could save the settings of this page. |

● **Step 2. Module Setting**



Click the next step, and enter the **Step 2 [Module Setting]** of the UI setting. This page is for setting the communication values with the connected modules.

First, choose the serial port that connected with the module. If use ICP DAS module, select the model to auto load the module setting. If not, give a name (Default: Name), click [ ⊕ ] button to add a module.
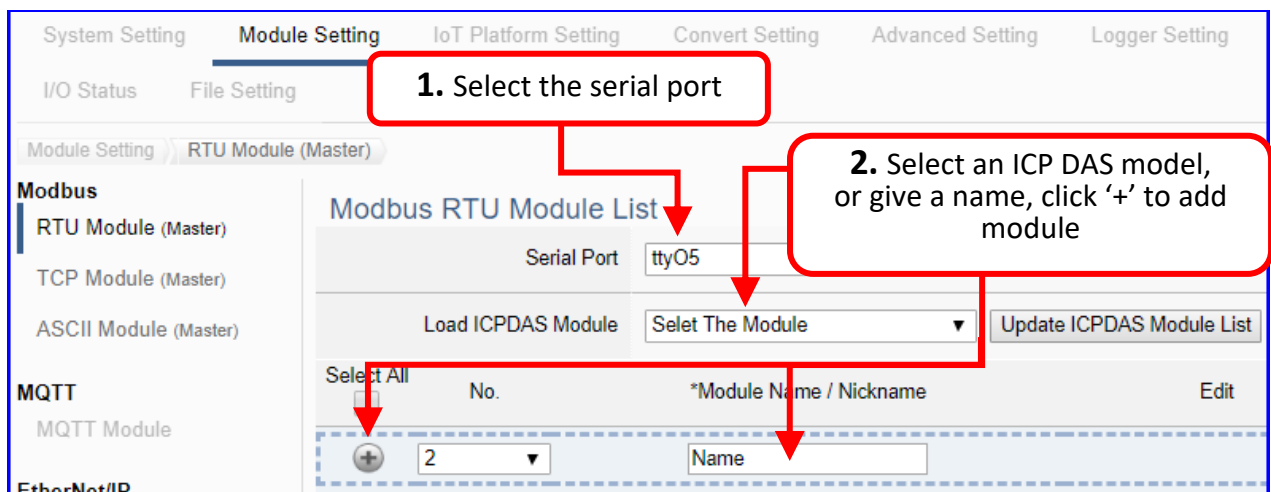


Add a module (Ex: **M-7055D**) as below, and then click [Edit] button to enter the "Module Content Setting" page.
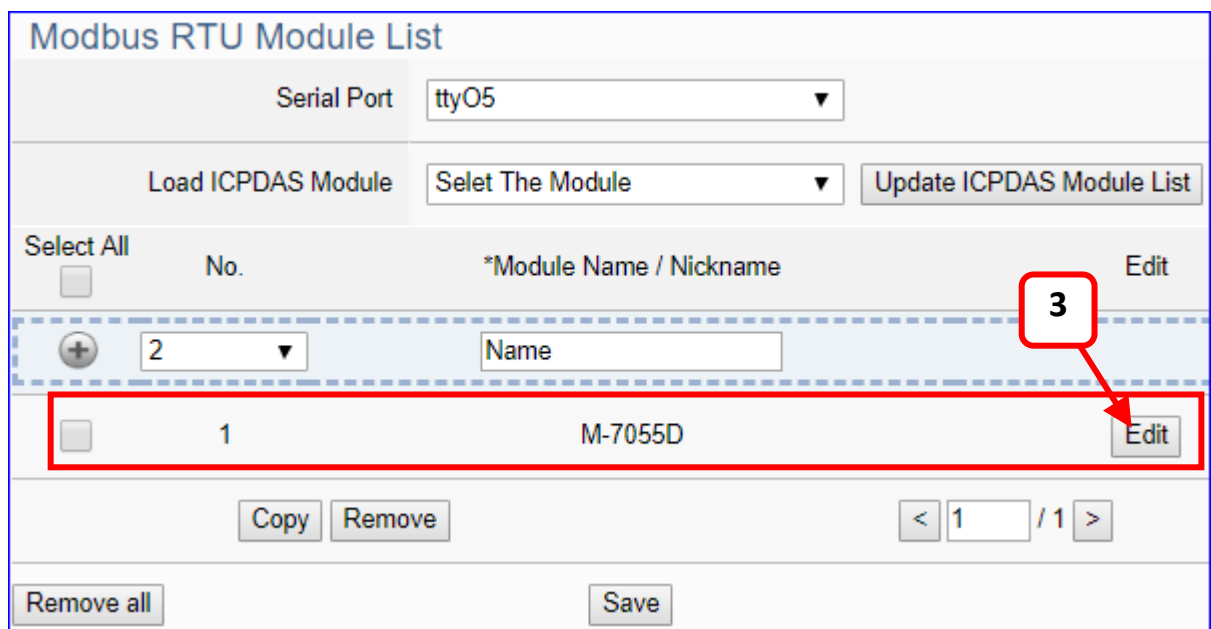
If set up a wrong module, user can click the box in the left side of the module number and click the [Remove] button to delete the module.

[**Module Content Setting**] page can set up the module and the Modbus address mapping table:

If use ICP DAS module, system can auto setup Modbus Mapping Table; if not, user needs to check Modbus address or I/O number from the module user manual.
> **Modbus Mapping Table Setting:**
Set up in the order of Data Model, Start Address and Data Number, then click "Add".
**Ex**: M-7055D has 8 Data Models of "01 Coil Status (0x)" (Mapping: DO), so select Model "01", Start Add. "0", Number "8", and click "Add".

| **Module Content Setting** | |
|---|---|
| No. | The module number in the module list (Not editable here) |
| Module Name | Give a name, e.g. model number or name. Default: Name. |
| Slave ID | Set the module Slave ID of the UA. (Range: 1 ~ 247) |
| Timeout | Set the timeout value for the module. Default: 500 ms |
| **Modbus Mapping Table Setting** | |
| Data Model | System provides 4 Modbus data models "01" ~ "04" for mapping to address of DO, DI, AO and AI. (ex. 01: DO channels, 02: DI, 03: AO, 04: AI) |
| Start Address | The start address of the Modbus command. **Note:** the Start Address of UA is bass on 0, even if some modules are bass on 1, here it needs to follow UA to set bass on 0. |
| Data Number | The number of the Modbus address. Need to give enough number for the DO, DI, AO, AI channels of the module. Default: 1. |
| Type | This item only when the data model is 03 or 04. Choose the suitable data type: 16-bit Short, 16-bit Unsigned Short, 32-bit Long, 32-bit Unsigned Long, 32-bit Float, 64-bit Double. |
| Create Tables | Click [Add] button, it will add a table in the Modbus mapping table. |

The finished Modbus Mapping Table as below is in order of DO, DI, AO and AI.

**Address:**
Display and edit the Modbus Mapping Table.



If user selects ICP DAS module, the system will auto set up the Modbus Mapping Table. If not, user needs to check the module Modbus address or I/O number from the module user manual.

| Modbus Mapping Table – Address Setting | |
|---|---|
| Address Setting | The "Address Setting" page of the Modbus Mapping Table |
| Nickname Setting | Click can switch to the The "Nickname Setting" page of the Modbus Mapping Table. (Next page) |
| Modbus Mapping Table | Coil Status(0x): Mapping to DO Modbus address<br>Input Status(1x): Mapping to DI Modbus address<br>Holding Registers(4x): Mapping to AO Modbus address<br>Input Registers(3x): Mapping to AI Modbus address |
| Address | The start address of the Modbus command. Default: 0. **Note:** the Start Address of UA is bass on 0, even if some modules are bass on 1, here it needs to follow UA to set bass on 0. |
| Number | The number of the Modbus address. Need to give enough number for the DO, DI, AO, AI channels of the module. At least 1. |
| Type | DO/DI type: Bool (Boolean)<br>AO/AI type: depend on setting of [Modbus Mapping Table Setting] |
| Edit | Click to change the address and Number. |
| Delete | Click to delete this address table. |
| Save | Click to save and exit this table editing. |
| Cancel | Click to exit without saving and back to the module list page. |
| OK | Click to save this page settings and back to the module list page. |

**Nickname:**

Setting the variable nickname and description.



| Modbus Mapping Table – Nickname Setting | |
|---|---|
| Modbus Mapping Table | Coil Status(0x): Mapping to DO Modbus address |
| | Input Status(1x): Mapping to DI Modbus address |
| | Holding Registers(4x): Mapping to AO Modbus address |
| | Input Registers(3x): Mapping to AI Modbus address |
| Table Display | Click [Show] to display all fields, click [Hide] to hide some fields. |
| Address | Modbus address. System auto arrange. |
| Variable name | The variable name of the mapping address. Default: Tag0 and auto arrange the number. User can define the name. |
| Data Type | Display data type of the variable. (Not editable) |
| Swap | Check to swap the byte order (Lo-Hi/Hi-Lo) for 4-byte or 8-byte. |
| Description | Write a note for this variable. |
| OK | Click to save this page settings and back to the module list page. |

**Scaling:**

**Scaling is only available in the AI/AO settings of Modbus RTU/TCP.** When the variable value needs to be scaled or converted before output, click the "**Advanced Setting**" button of the variable on the **Scaling** page, input the **Min**./**Max**./**Offset** of the Reference/Output items, add a description, and check "**Enable**" box, The Scaling conversion function will be activated.

The M-7055D has no AI/AO, so here uses the screen of DL-302 for an example.



| Modbus Mapping Table – Scaling | |
|---|---|
| Modbus Mapping Table | Holding Registers(4x): Mapping to AO Modbus address<br>Input Registers(3x): Mapping to AI Modbus address<br>**Scaling do not support 01 Coil Status(0x):DO** & **02 Input Status(1x):DI** |
| Table Display | Click [Show] to display all fields, click [Hide] to hide some fields. |
| Address | Modbus address. System auto arrange. |
| Reference | The I/O variable of the Modbus address. |
| Output | The scaling variable for scaling output. User can define the variable name. |
| Scaling | Click [Show Detail] to set up the Scaling parameters, and click [Hide Detail] to hide the parameters.<br>Fill in the Min/Max range values of the source in the Reference column. Fill in the Min/Max range values after scaling in the Output column. If needs offset, fill the offset value in the Offset item. Remember check "Enable" box. |
| Enable | Check the box of the variable can enable just that variable for scaling. |
| Description | Write a note for this variable. |
| OK | Click to save this page settings and back to the module list page. |

**Bitwise:**

**Bitwise is only available in the AI/AO settings of Modbus RTU/TCP.** When the data needed to take out the value of the specified bit, fill in the variable name in the specified Bit# of the required address, and the value of the bit can be output to the filled variable.
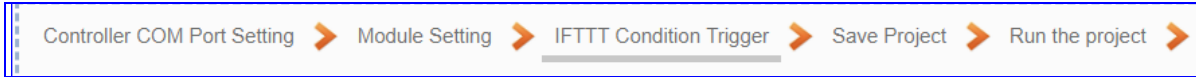
The M-7055D has no AI/AO, so here uses other module's setting screen as an example.



| Modbus Mapping Table – Bitwise | |
|---|---|
| Modbus Mapping Table | Holding Registers(4x): Mapping to AO Modbus address<br>Input Registers(3x): Mapping to AI Modbus address<br>**Bitwise do not support 01 Coil Status(0x):DO & 02 Input Status(1x):DI**<br>**Bitwise do not supports 32-bit Float & 64-bit Double data types.** |
| Table Display | Click [Show] to display all fields, click [Hide] to hide some fields. |
| Address | Modbus address. System auto arrange. |
| Reference | The Bit# variables of the Modbus address. |
| Bitwise | Set up the variables for Bitwise. Click [Advanced Settings] to set up the Bitwise parameters, and click [Hide] to hide the parameters.<br>Fill in the variable names to the Bit# that wanted to do the Bitwise. The value in the fixed bit number will be assigned into the variable. |
| OK | Click to save this page settings and back to the module list page. |

● **Step 3. IFTTT Condition Trigger**



Click the next step, and enter the **Step 3 [IFTTT Condition Trigger ]**.
This page is for the APP message related setting, e.g. IFTTT event name, key, trigger condition, I/O variables ….

We select the "**IFTTT Condition Trigger (Line, Twitter)**" at the beginning, so this step will auto enter the setting page [**Advanced Setting > IFTTT Condition Trigger**]. The "Step Box" will prevent the user from selecting the wrong platform.
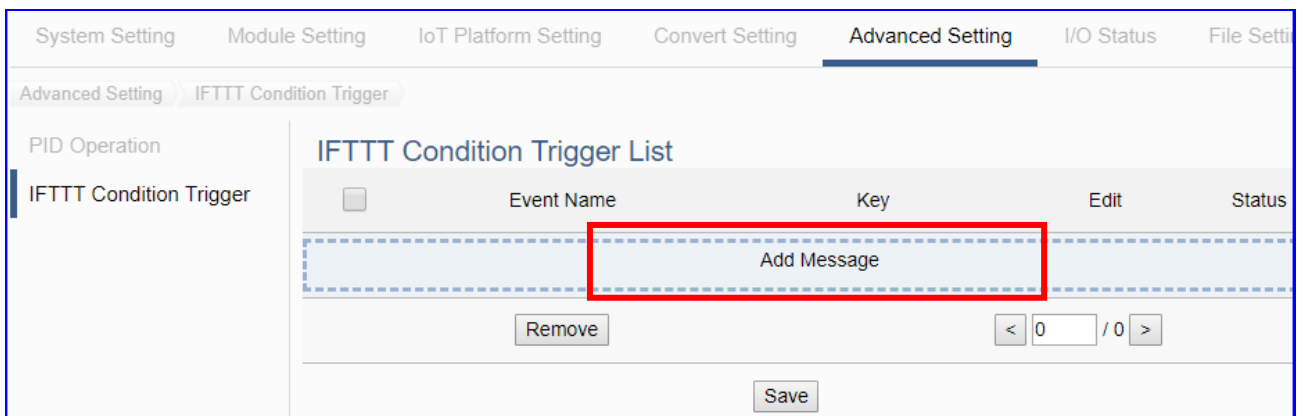


| Advanced Setting > IFTTT Condition Trigger > FTTT Condition Trigger List | |
|---|---|
| Add Message | Click to add a new IFTTT message. After setting, an IFTTT condition trigger list will show on the bottom includes left box, event name, key and status. |
| ☐ | Check the box in the left of the list is to select and to delete the list. Check the box on the top will select all lists. |
| Event Name | Display the "Event Name" setting in the IFTTT website. (FAQ-app-02) |
| Key | Display the "Key" getting from the IFTTT website. (FAQ-app-02) |
| Edit | Click [Edit] can set the IFTTT condition trigger content. |
| Status | Display the enable status of the IFTTT condition trigger list. |
| Remove | Click the left box and [remove] can delete the IFTTT list. |
| < 1 / 1 > | The page number of the IFTTT list: Current page / Total pages. Click < or > to go to the previous or next page. |
| Save | Click to save the setting of this page. |

Click [**Add Message**] botton to enter the IFTTT [Content Settings] page:

Content Setting

Event Name: UA-5200 test

Key: fkCGvasDPR-xYe2ugpgQ7

Status: ☑ Enabled

**Note: Case sensitive for Event Name and Key.**

**Note**: The "Event Name" and "Key" are set in the IFTTT website. If you are not familiar with IFTTT, please see the FAQ-005 for the setting introductions.

| Advanced Setting > IFTTT Condition Trigger > Content Setting | |
|---|---|
| Event Name | Input the "Event Name" setting in the IFTTT website. (FAQ-app-02) |
| Key | Input the "Key" getting from the IFTTT website. (FAQ-app-02) |
| Status | Check to enable the IFTTT condition trigger event. |

Condition Setting

Module Variables — Operator — Value

↓ Module Type
Modbus RTU (Master)

↓ Module Name
No.1 M-7

↓ Variable Attribute
Read

↓ Variable Name
Tag0 (Short)

= ▼

Type：User-Defined
Dead Band：1

Add

| Advanced Setting > IFTTT Condition Trigger > Condition Setting | |
|---|---|
| Module Variables | Select the module and variable for the condition trigger. Module Type: select the module type, Modbus RTU/TCP/ASCII… Module Name: select the module that set for condition trigger. Variable Attribute: select the variable attribute for condition trigger. Variable Name: select the variable name for condition trigger. |
| The following condition fields may different depending on the selected variable attribute. The condition trigger method will be described after this table. | |
| Operator | Select the operator for the trigger condition. |
| Value | Set up the value for the condition, include Type and Dead Band. |
| Status | Set up the status for the condition. Default: 0. |
| Add | Click to add a condition trigger list in the Condition Table.. |

## Condition Trigger Descriptions:

The condition trigger method will different depending on the attribute of the selected variable and the trigger will be different. There are two operation styles: **DIO** and **AIO**.

**(A)** If select **DIO variable**, then Condition is "Status Change". When detecting the status is changed, it will trigger the event and send the assigned message. (Below is a switch detecting example.)
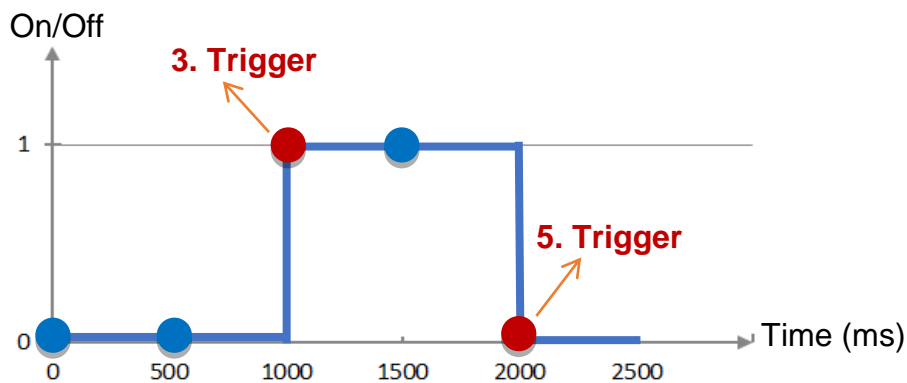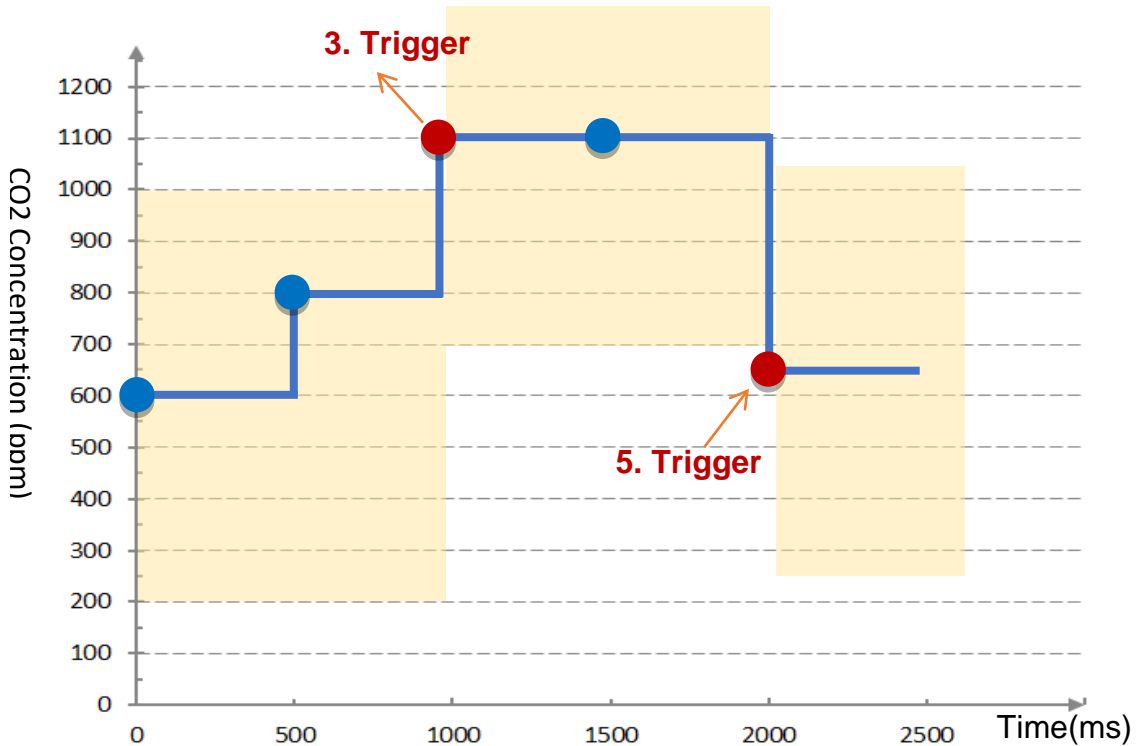


**DIO Trigger**: (Detect per 500 ms)
1. Detect initial switch status "Off" (status = 0)
2. Detect "Off" (status = 0, status no change), no trigger
3. Detect "On" (status = 1, status changed), **trigger** a message notification
4. Detect "On" (status = 1, status no change), no trigger
5. Detect "Off" (status = 0, status changed), **trigger** a message notification

**(B)** If select **AIO** **variable**, then Condition is "Value" and can set the "Dead Band". The condition will be triggered and send the message when the detected value exceeds the upper or lower Dead Band. (Below is a CO2 example. Detect per 500 ms)



**AIO Trigger:** (Detect per 500 ms. The yellow block means the Dead Band.)
1. Detect initial CO2 concentration 600 (ppm).
    Set Dead Band=400 (Initial Trigger Condition: **>= 1000 or <= 200**)
2. Detect CO2 concentration 800. It is in the range of Dead Band.
3. Detect CO2 concentration 1100. It exceeds the upper value (>= 1000) of Dead Band,
    so **trigger** a message for danger notification.
4. Detect CO2 concentration 1100. It is in the new range of Dead Band.
    Dead Band=400 (New Trigger Condition: **>= 1500 or <= 700**)
5. Detect CO2 concentration 650. It is below the lower value (<= 700) of Dead Band,
    so **trigger** a message for safety notification.

Please refer to the previous Condition Trigger Descriptions to set up your Condition. When complete, click the "Add" button. The setting will show in the Condition Table.

Below Table is setting 2 conditions.



| Advanced Setting > IFTTT Condition Trigger > Condition Table | |
|---|---|
| Module | Display the module type and name of the condition. (Not editable here) |
| Variable | Display the variable attribute and name of the condition. (Not editable here) |
| Condition | Display the trigger condition. (Not editable here) |
| Define Message | Default Message: module code_variable code. The user can define own message in the format of English character, number, general symbol… |
| Remove | Click the left box and [remove] can delete the IFTTT list. |
| OK | Click to save this page settings and back to the module list page. |
| Cancel | Click to exit without saving and back to the module list page. |

When back to the IFTTT Condition Trigger List, the condition trigger message will show as below picture. If need more trigger conditions, click the "Add Message" again to combine the IFTTT APP message sending and the UA system. At last, click the Save button.
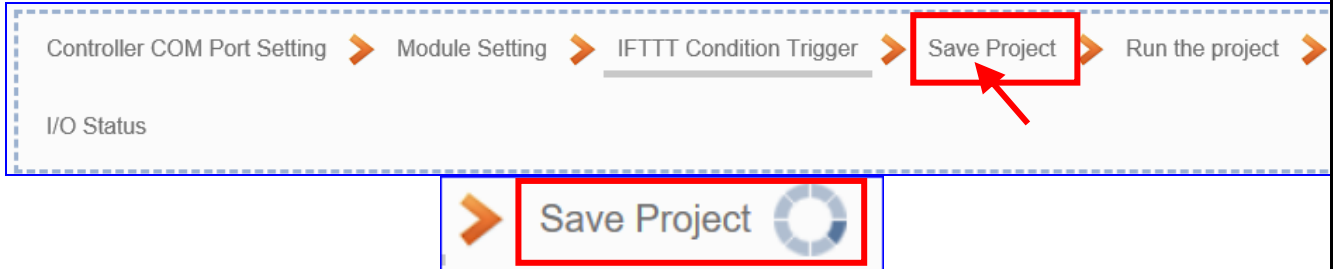
● **Step 4. Save Project**

The setting of this example is finished now. Click the next step **[Save Project],** the Step Box will show an animation as below picture, that means the project is saving. When the animation vanished, the project is saved completely.





● **Step 5. Run the Project**

The project, after saving, needs to be executed. Click the next step **[Run the Project]**. This step can also via the [**System Setting** > **Controller Service Setting** > **Run Project**] to Stop and Run the project.
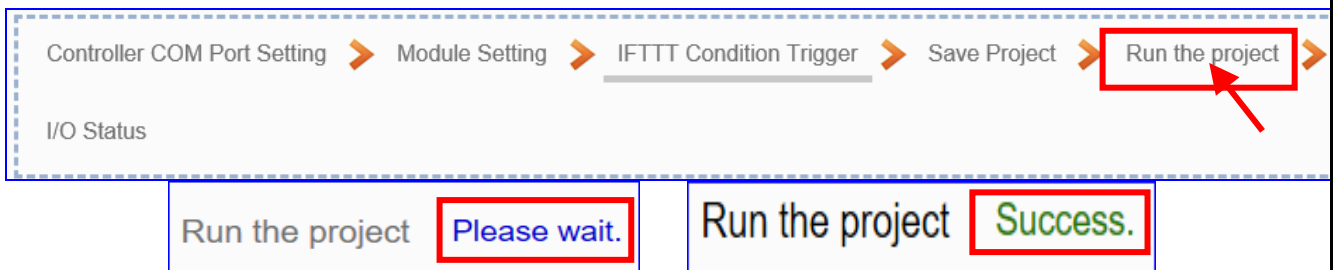




When the words "**Please wait**" disappears, the new words "**Success**" appears, that means the UA controller is running new project successfully.

● **Step 6. I/O Status**



This step will go to the Main Menu [I/O Status]. The users can click the setup module to see its real time I/O status.

Then the Step Box will disappear automatically now, and back to the first screen view of the Web UI.

The new project now completes the setting, uploading and running in the UA controller and can process the PID function. Users can see the I/O status from the menu [**I/O Status**]. For more about the Web UI settings, please refer to CH4 and CH5.

The project for APP message notifies via the **IFTTT condition trigger (Line, Twitter)** is now done.