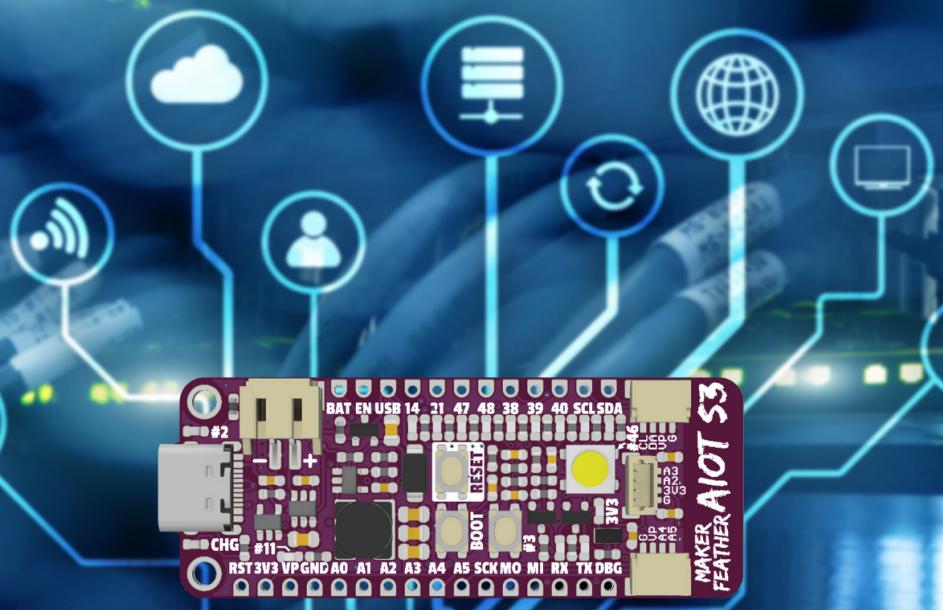


MANUAL GUIDE

IOT KIT



**MAKER FEATHER AIOT S3
(BASIC KIT)**

VITrox[®]

Cytron[®]
Technologies

V-ONE[®]

Part 1

V-One Platform

Account Registration

Hello Maker! Congrats on your first step in exploring the Internet of Things (IoT) world. If this is your first time doing an IoT project, don't worry! You will successfully create a few cool IoT projects at the end of this manual guide.

There are two main components required to run an IoT project. The first one is of course the hardware components and the other one is an IoT platform. You don't need to worry about the hardware part since everything is provided in the IoT kit box. For this kit, you will be using V-ONE for the IoT platform which has been designed by a famous electronic company in Malaysia, ViTrox. This Manual Guide will help you explore this interesting V-ONE platform with the new features released in 2022, the AloT module.

For those who prefer to watch a video instead of reading this manual guide, you can watch the video below that summarizes Part 1 and Part 2 of this manual guide. For details of each step, you can always come back here and refer to this manual guide.

Tutorial Video (Part 1 & Part 2)

[Get Started With Maker Feather S3 IoT Kit - Simplifying IoT with V-ONE](#)

First and foremost, you will need to create an account for V-ONE to get started. Once you open the IoT kit box, there will be a registration card for the V-ONE platform. Copy the URL link into a browser using your laptop or you can just simply scan the QR code with your phone.

- <https://v-one.my/subscribe/free>



There are a few simple questions that need to be answered. Once you have answered both questions, click “**Next**”.

The screenshot shows a “Subscribe” page for V-ONE. At the top right, there is a navigation bar with links: Why V-ONE, IoT Cloud, Partners, Subscribe, V-Library, Support, and a Login button. Below the navigation bar, the page title is “Subscribe”. A sub-section titled “my V-ONE” contains the text: “Congratulations on your first journey with V-ONE. Let us understand your needs, while filling the form takes only a few minutes to complete.” To the right of this, under the heading “Help us to understand you better”, there are two dropdown menus. The first dropdown, labeled “1. Your current role:”, has “Student” selected. The second dropdown, labeled “2. How will you be using V-ONE?”, has “Self-learning” selected. At the bottom of the page are “Back” and “Next” buttons.

Fill in the required information and click “**Next**” to proceed.

Note: Please remember the **Email and Password** here since it will be used to **login** into your account later.

The screenshot shows the same “Subscribe” page for V-ONE, but now at “Step 2 - Fill in your info”. The “my V-ONE” section remains the same. To the right, under the heading “Tell us about yourself”, there are four input fields arranged in a grid: “Student Email” and “Contact Number” in the top row, and “First Name” and “Last Name” in the bottom row. Below these, there are two more rows of input fields: “Create Password” and “Confirm Password”. Further down, there is a section for “Your School” with “School Name” and “Country” fields, and a “Promo Code (Optional)” field. At the bottom are “Back” and “Next” buttons.

For the last step, you must agree to the V-ONE Software License Agreement. Read all the agreements and scroll down to the bottom of the page.

V-ONE®
Why V-ONE ▾
AIoT Cloud
Partners
Subscribe
V-Library ▾
Support ▾
[Login](#)

Subscribe

Step 3 - Agree to V-ONE Software Licence Agreement

my V-ONE

Read and agree to our V-ONE Software Licence Agreement

V-ONE SOFTWARE LICENCE AGREEMENT

This V-ONE Software Licence Agreement ("Agreement") is hereby entered into between you, as an individual or an entity ("Customer") and ViTrox Technologies Sdn Bhd (Company No. 507043-P), a subsidiary of ViTrox Corporation Berhad (Company No. 649966-K) . ViTrox and the Customer shall collectively be referred to as ("Parties") and individually as ("Party"). This Agreement governs access to and use of the services provided by ViTrox Technologies Sdn Bhd ("ViTrox"). By clicking "I Agree", signing the contract for the Services or for use of the Services, downloading the Software or otherwise using the Software or Services, the Customer agrees to be bound by the terms and conditions of this Agreement.

Terms and Conditions

1. Definitions

| | |
|--------------------------|--|
| Account Data | means the account and contact information submitted to ViTrox by the Customer for use of the Services |
| Admin Account | means the administrative account provided to the Customer by ViTrox for the purpose of administering the Services |
| Confidential Information | means all information or data disclosed (whether in writing, orally or by any other means) to the Customer by ViTrox or by a third party on behalf of ViTrox and shall include, but shall not be limited to, ViTrox's products, customers, prospective customers (including lists of customers and prospective customers), methods of operation, engineering methods and processes (including any information which may be obtained by the Customer by reverse engineering, decompiling or examining any Confidential Information provided by ViTrox), programs and databases, patents and designs, billing rates, billing |

If you have no problem with the agreement, tick the box "I agree to the V-ONE Software License Agreement" and click "**Submit**".

V-ONE®
Why V-ONE ▾
AIoT Cloud
Partners
Subscribe
V-Library ▾
Support ▾
[Login](#)

15.7 Assignment. The Customer may not assign or transfer this Agreement or any rights or obligations under this Agreement without the prior written consent of ViTrox. ViTrox may assign, transfer, subcontract or novate this Agreement or any rights or obligations under this Agreement to an affiliate or in connection with any merger, acquisition, corporate reorganization or sale of all or substantially all of its assets without providing notice.

15.8 Successors bound. This Agreement shall bind the personal representatives, heirs, successors-in-title and assigns of the Customer and ViTrox respectively.

15.9 Use of data. The Customer agrees that ViTrox may collect and use technical data and related information—including but not limited to technical information about the Customer's device, system and application software, and peripherals—that is gathered periodically to facilitate the provision of software updates, product support, and other services to the Customer (if any) related to the Software and the Services. The Customer agrees that ViTrox may use this information, as long as it is in a form that does not personally identify the Customer, to improve its products or to provide services or technologies to the Customer.

15.10 Dispute resolution.

- (a) Each Party agrees to firstly resolve any dispute through amicable negotiation by contacting the other party by sending notice in accordance with Clause 15.4. If such dispute remains unresolved within thirty (30) days from the date of receipt of notice, the Customer or ViTrox may bring a formal proceeding in accordance with Clause 15.10(b).
- (b) Subject to Clause 15.10(a), any dispute, controversy or claim arising out of or relating to this Agreement, or the breach, termination or invalidity thereof, shall be settled by arbitration, Asian International Arbitration Centre (Malaysia) ("AIAC") held in Kuala Lumpur in accordance with the AIAC Arbitration Rules 2018 or such other prevailing rules practised or adopted by the AIAC. The language of arbitration shall be in English and the arbitral award shall be final and binding on both parties.
- (c) Either Party may bring a civil action against the other for injunctive relief to stop the unauthorized use or abuse of the Services or infringement of Intellectual Property Rights or Confidentiality obligations without first engaging in the informal dispute notice process under Clause 15.10(a).

15.11 Survival of terms. Clauses 5 (Fees), 6 (Intellectual Property Rights), 7 (Confidentiality), 10 (Indemnity), 13 (Limitation of liability) and 15 (Other Provisions) shall survive expiry or termination of this Agreement.

I agree to the V-ONE Software License Agreement.

[Back](#)
[Submit](#)

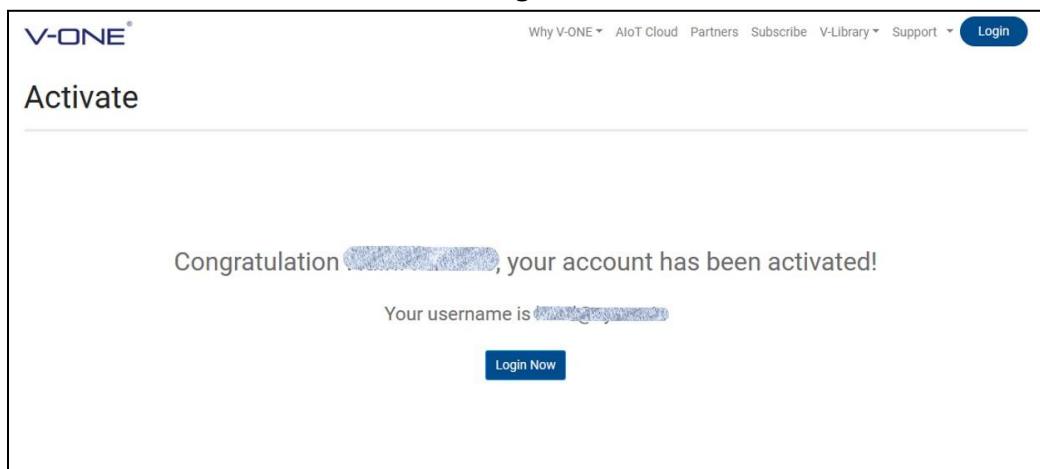
There you go, FREE V-ONE account for you! Please check your email to verify and activate your account.

The screenshot shows a 'Subscribe' page from V-ONE. At the top, there's a navigation bar with links like 'Why V-ONE', 'AloT Cloud', 'Partners', 'Subscribe', 'V-Library', 'Support', and a 'Login' button. Below the navigation, the word 'Subscribe' is prominently displayed. To the right, it says 'Step 4 - Thank you'. In the center, a large message says 'Thank you for subscription!'. Below that, smaller text reads 'Please check your email to verify and activate your account.' At the bottom, there's a link 'Go back to home page'.

Open an email from V-ONE Admin, and click the “Activate Now” button to proceed.

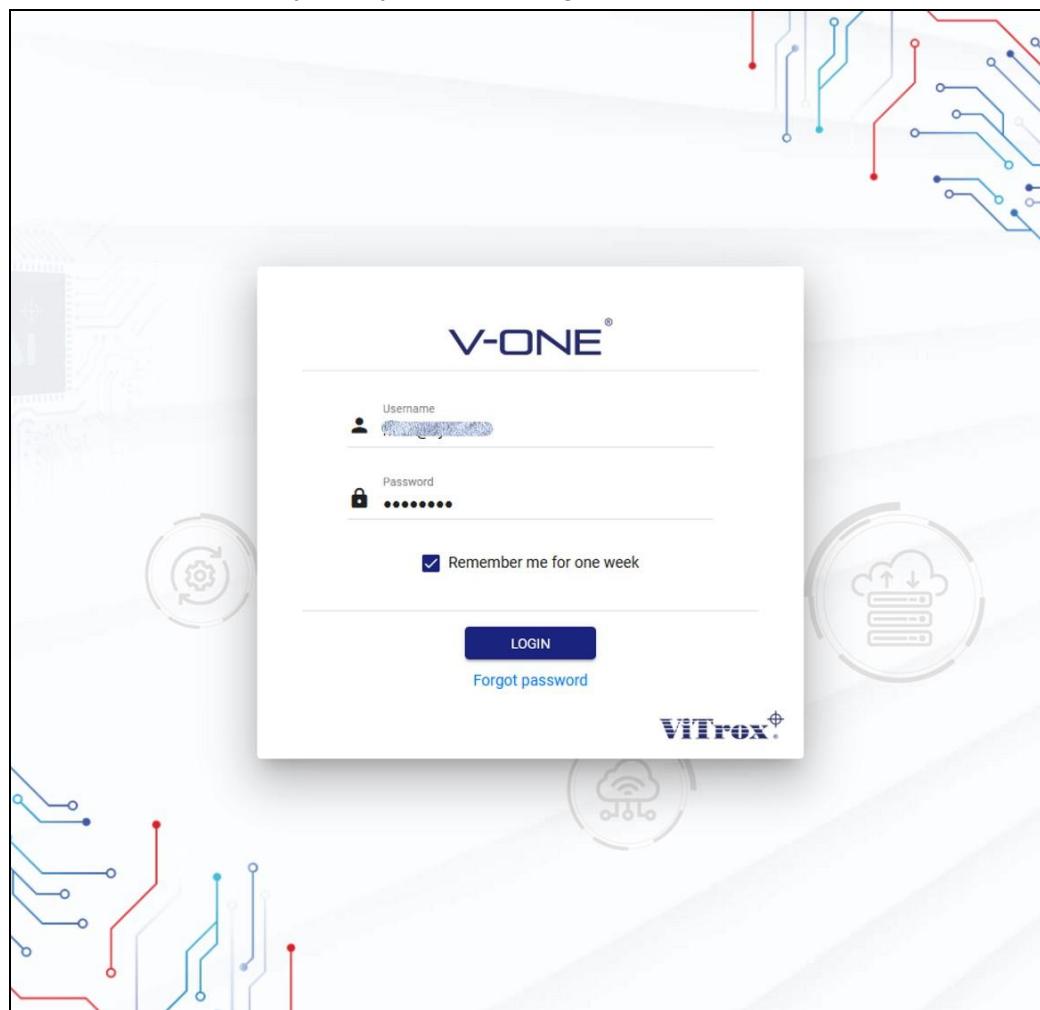
The screenshot shows an email from 'V-ONE Admin <admin@v-one.my>' to the user. The subject is '[V-ONE] New Subscription'. The email body starts with a large button labeled 'THANK YOU FOR SUBSCRIBING TO V-ONE'. Below this, it says 'Dear [REDACTED]', 'Kindly click the below button to activate your account.', and features a prominent 'Activate Now' button. At the bottom, there's contact information for the Customer Care Center, including phone numbers for V-Talk and V-Reach, and a V-Chat option. The V-ONE logo is at the very bottom.

Your account has been activated! Click the “**Login Now**” button.



This is the login page for V-ONE. Insert your username (email address) and password to login.

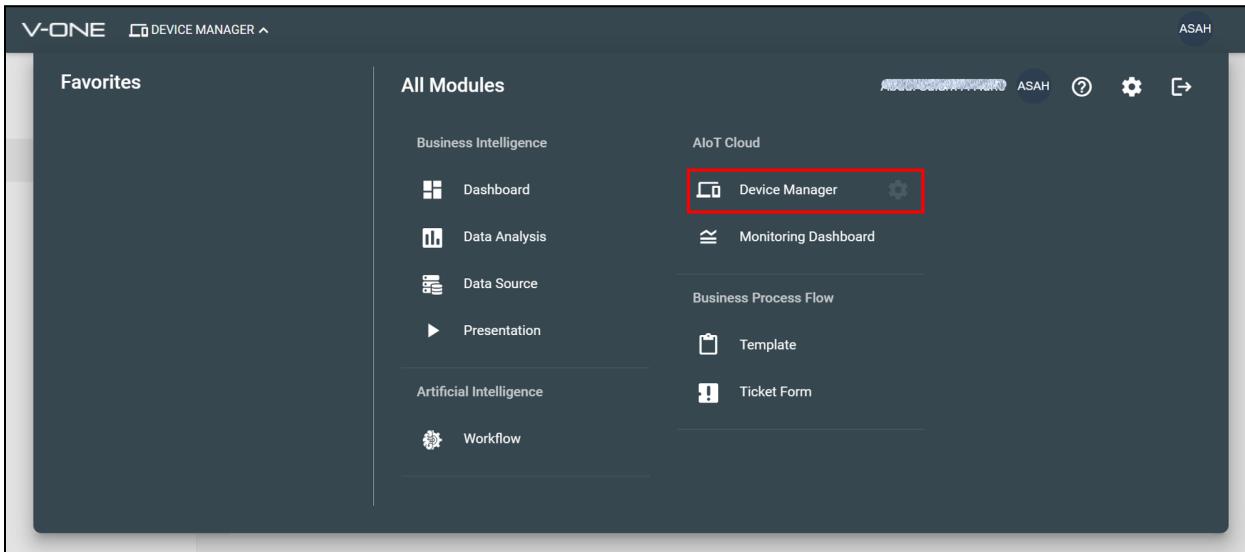
Tips: You can tick the "Remember me for one week" box so that you don't need to insert the username and password every time you want to login.



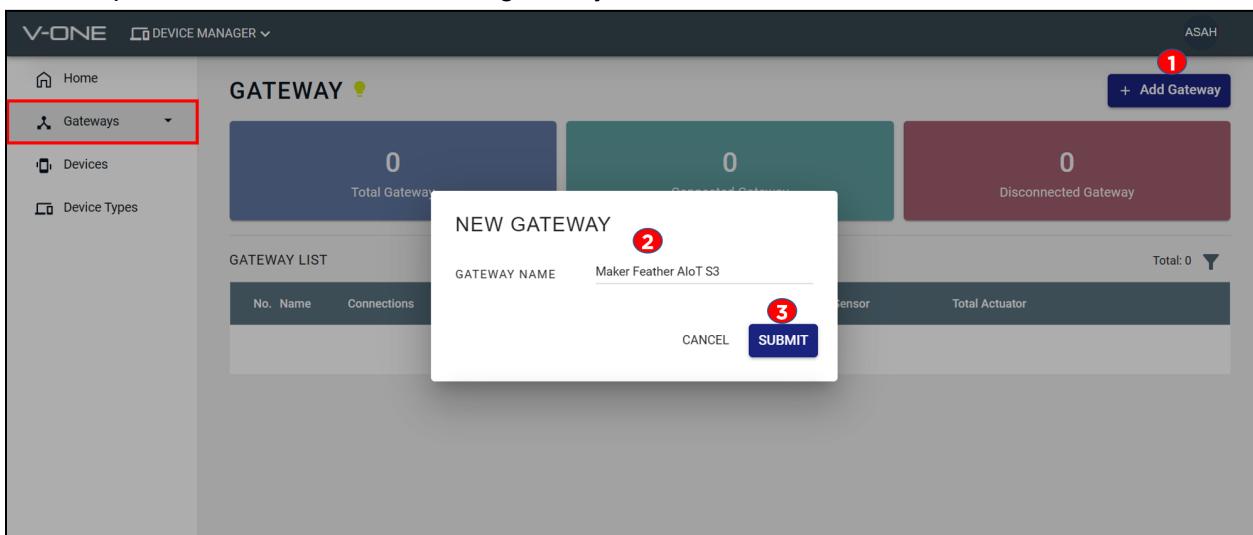
Register Gateway

Gateway here refers to the Maker Feather AloT S3 board that acts as a microcontroller to build your IoT projects. Each microcontroller will have a unique GatewayID and Access Token to function as a gateway.

To create a gateway, go to “**Device Manager**”.



In the **Gateways tab**, click “**Add Gateway**”. Then, insert the gateway name and click “**Submit**”. You can put the microcontroller as the gateway name.



With just a few clicks, you have successfully created a gateway!

Now, you can click on the eye icon to see the gateway information. Both **gatewayID** and the **Access Token** will be used in the library later (in **Part 2**).

The screenshot shows the V-ONE Device Manager interface. On the left sidebar, under the 'Gateways' section, there is a list of devices: 'Devices' and 'Device Types'. The main area is titled 'GATEWAY' with a yellow lightbulb icon. It displays three summary boxes: 'Total Gateway' (1), 'Connected Gateway' (0), and 'Disconnected Gateway' (1). Below this is a table titled 'GATEWAY LIST' with one row. The row contains the following columns:

| No. | Name | Connections | Total Connected Sensor | Total Sensor | Total Actuator | Action Buttons |
|-----|-----------------------|--------------|------------------------|--------------|----------------|----------------|
| 1 | Maker Feather AloT S3 | Disconnected | 0 | 0 | 0 | |

Under the 'Name' column, the gateway details are shown:

- ID: a776a688-6c44-49c1-aa1f-05f0e5698f8b
- Access Token: uefDBvjomcDcLM0E

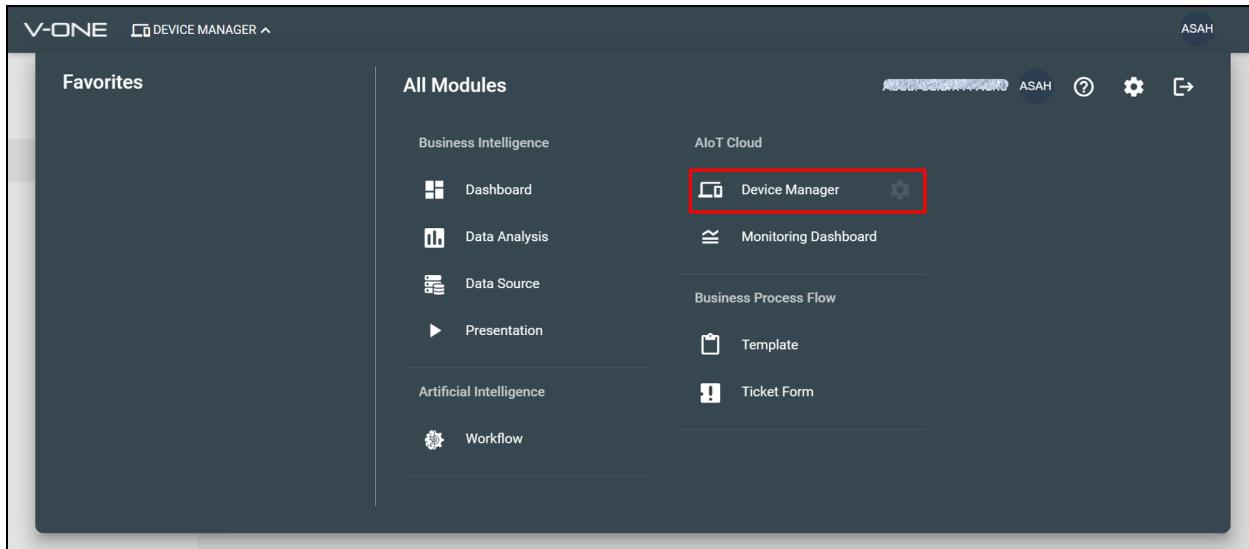
A red arrow points to the eye icon in the action column, and two red boxes highlight the 'ID' and 'Access Token' fields.

Add Device Types

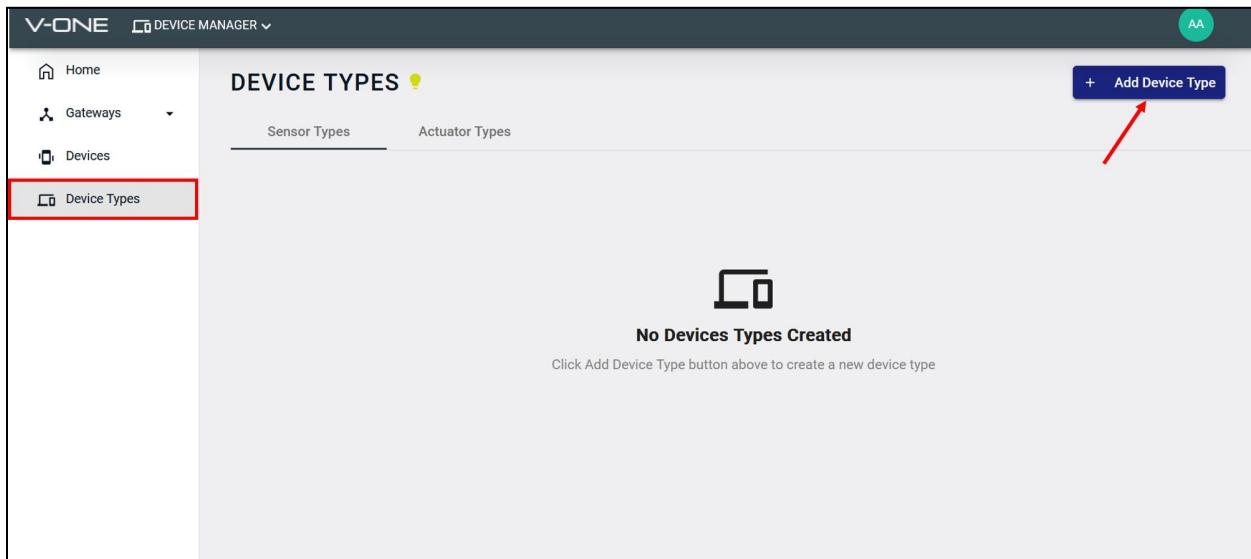
A sensor is a device that detects changes in the environment. The sensor converts a physical phenomenon into measurable digital/analog values which become a human-readable display or transmitted for reading. In this IoT Kit, there are 4 sensors for you to experiment with. The best part of this IoT kit is that it also provides actuators to make your IoT projects become more interesting! An actuator is a device that produces an output by converting energy and signals going into the system such as servo, LED, and relay.

Now you need to register all the sensors and actuators in the V-One platform. There are templates for each device, so you can just directly add them using the provided templates.

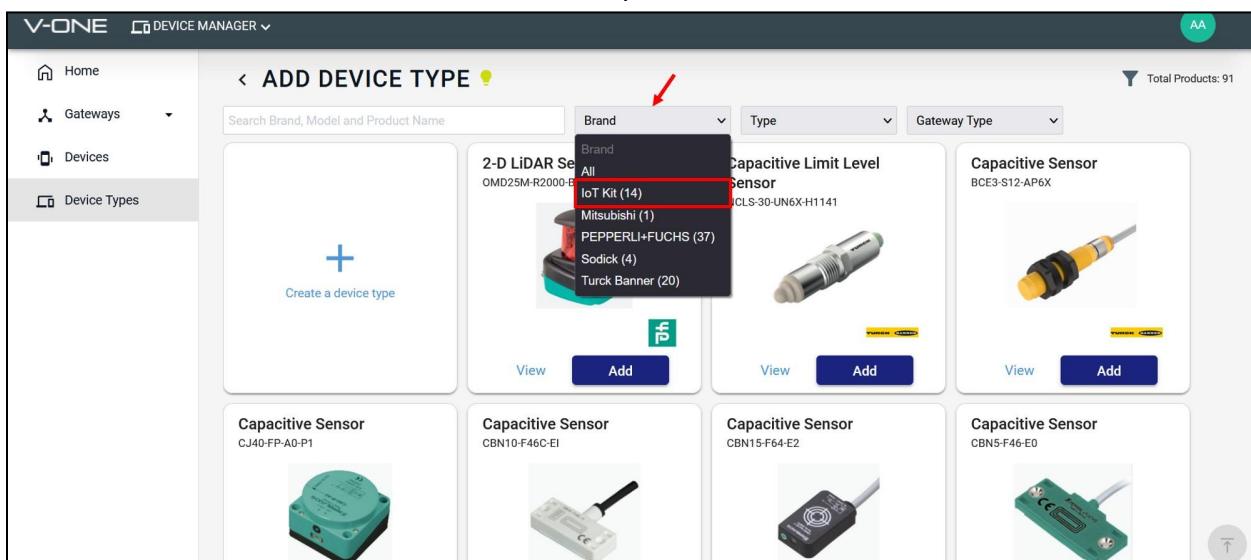
Go to “Device Manager”.



In the Device Types tab, click “Add Device Type”.



Click “Brand” and choose “IoT Kit” from the drop-down list.



Here you can see all templates are ready for every sensor and actuator provided in the IoT kit. Click “Type” to filter sensor or actuator devices.

V-ONE DEVICE MANAGER

Home Gateways Devices Device Types

IoT Kit (14) Type Gateway Type Total Products: 14

Create a device type

IoT Kit DHT11 View Add

IoT Kit Infrared sensor View Add

IoT Kit Push button - 1 View Add

IoT Kit PIR View Add

IoT Kit Water Level sensor View Add

IoT Kit Push button - 2 View Add

IoT Kit MQ2 sensor View Add

IoT Kit LED2 View Add

IoT Kit LED1 View Add

Click “View” and a new section appears showing the selected device's information.

V-ONE DEVICE MANAGER

Home Gateways Devices Device Types

IoT Kit (14) Type Gateway Type Total Products: 14

IoT Kit Infrared sensor View Add

IoT Kit Push button - 1 View Add

IoT Kit Moisture sensor View Add

IoT Kit Rain sensor View Add

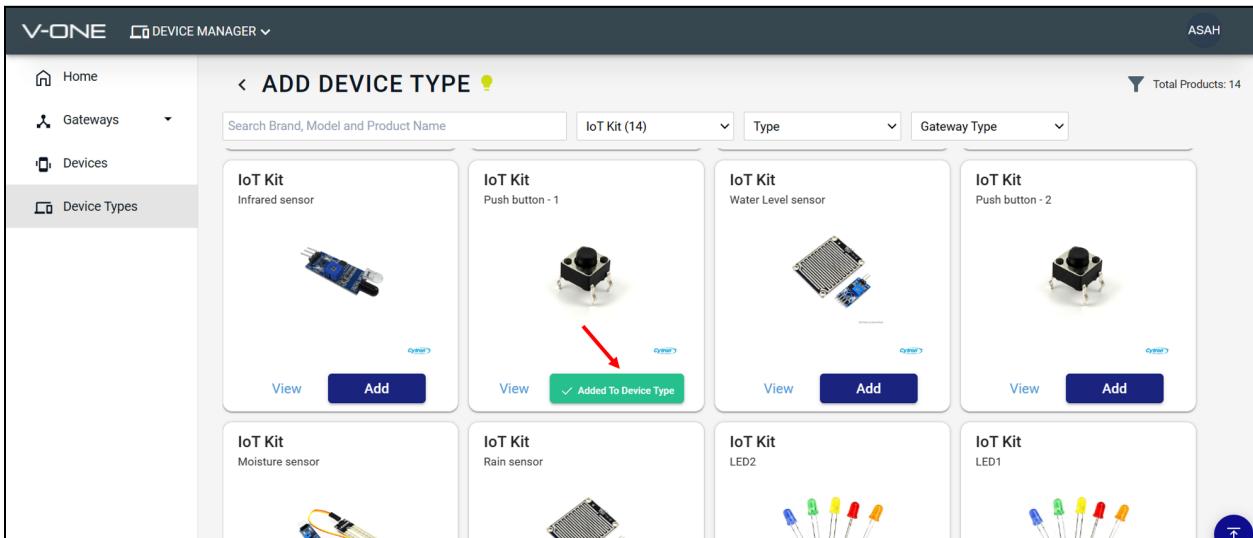
IoT Kit LED2 View Add

IoT Kit LED1 View Add

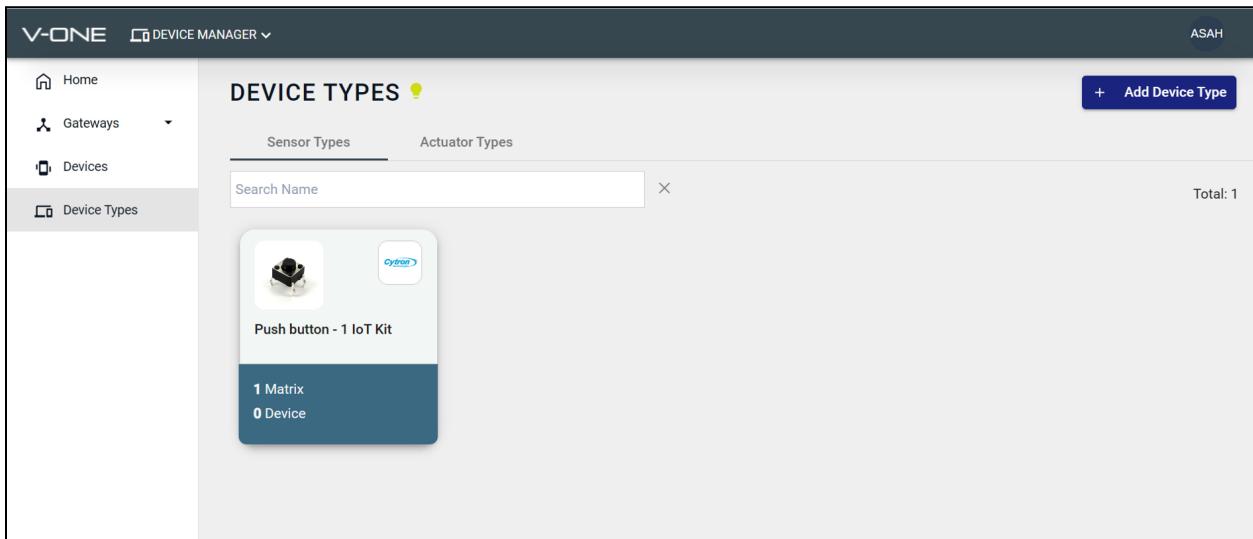
IoT Kit Data

| Key | Name | Data Type | Transmission Type |
|---------|---------|-----------|-------------------|
| Button1 | Button1 | integer | telemetry |

Lastly, you can just proceed by clicking the “**Add**” button.



Once you have added the device, it will appear on the “**Device Types**” main page. It can be under **Sensor Types** or **Actuator Types** depending on that device type.



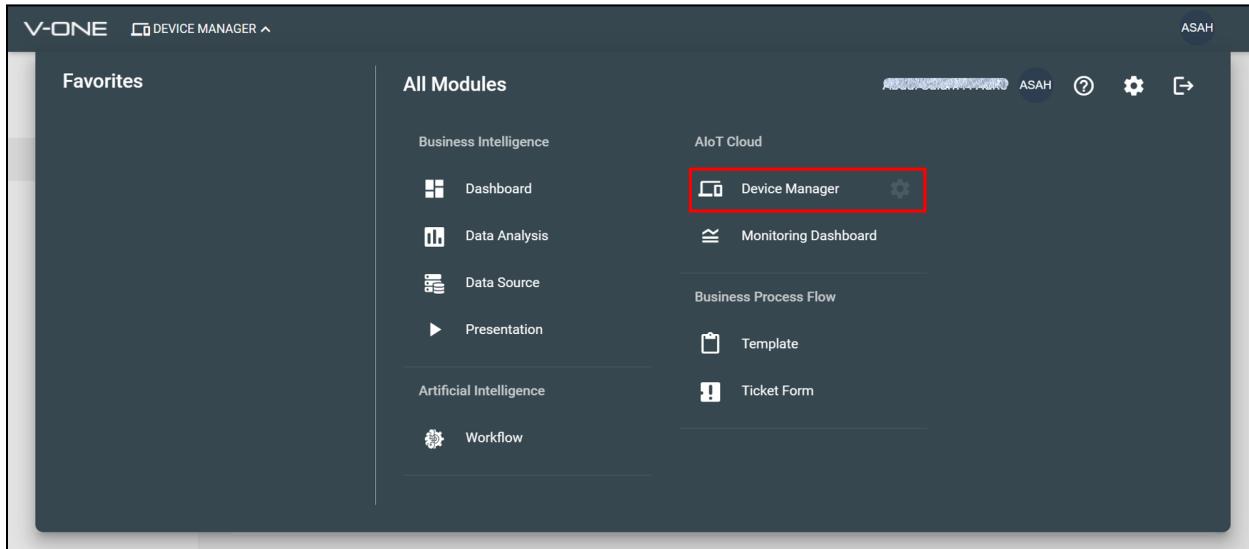
Continue to **register all the sensors and actuators** following the same steps above. These are all the required devices to be added for this **Basic Kit**:

- | | |
|--|---|
| 1. Push button - 1 2. Push button - 2 3. Moisture sensor | 4. Water Level sensor 5. Infrared sensor 6. Servo |
|--|---|

Add Device

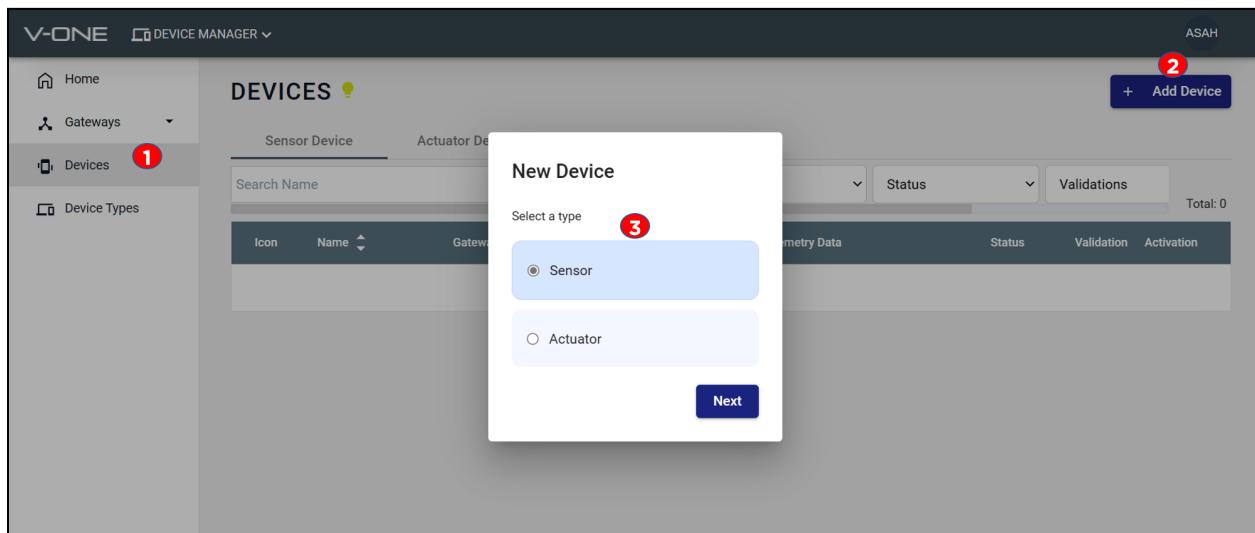
Please make sure that all the sensor and actuator types are already registered to proceed with this part. Now, you will assign all those sensors and actuators under your gateway. Every device will have a **unique devicelID** once it has been assigned under a gateway.

To add a device, go to “**Device Manager**”.



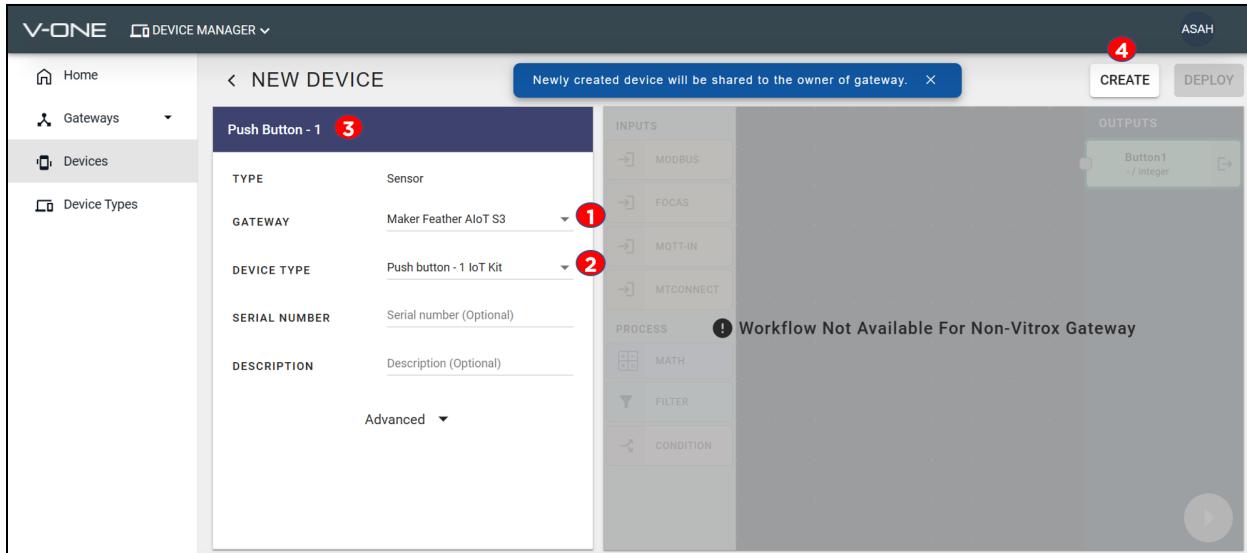
In the **Devices** tab, click “**Add Device**” and then choose “**Sensor**”.

Note: In this Basic Kit, there is only **one Actuator** which is the servo.

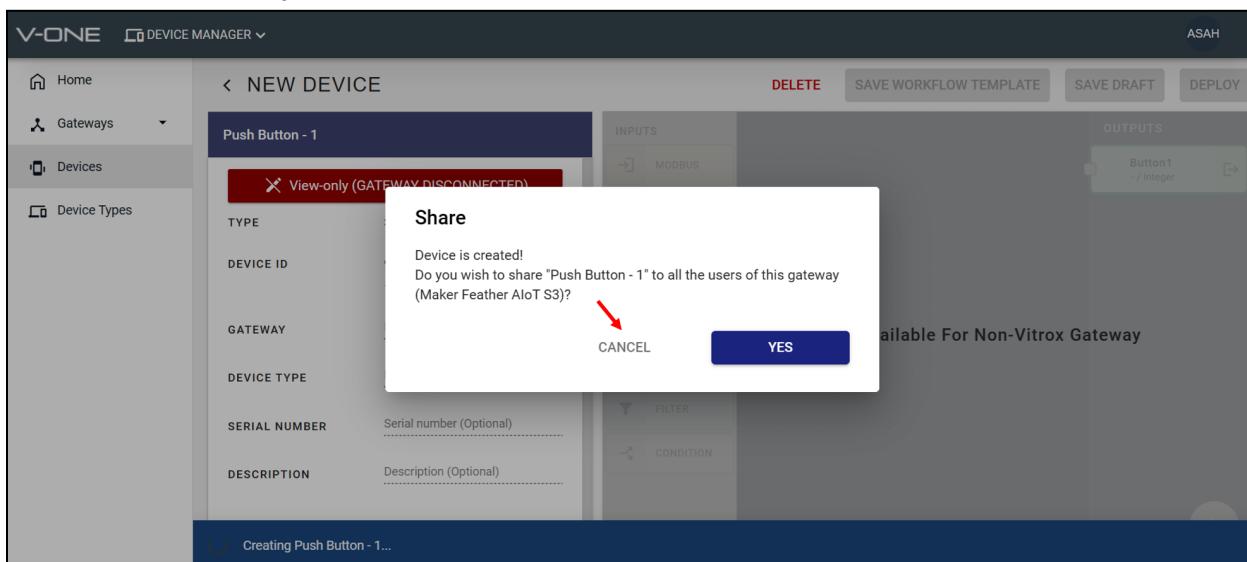


Choose the corresponding **gateway** and **device type**. Type in a name for the new device. You can use the same name as the device type to avoid confusion. Then, click the “**Create**” button.

Note: You need to type in the device name, select the gateway, and device type first to enable create button.



A pop-up will appear, just click “**Cancel**” to proceed.



You can go back to the Devices tab and check if the new device has been successfully added to the gateway. Here you will see that the added sensor is still in **Draft mode**. To deploy it, you need to **upload the code first** and make the gateway online.

Tips: Add all the sensors and actuators first, then you can simply deploy all the devices together once you upload the code later.

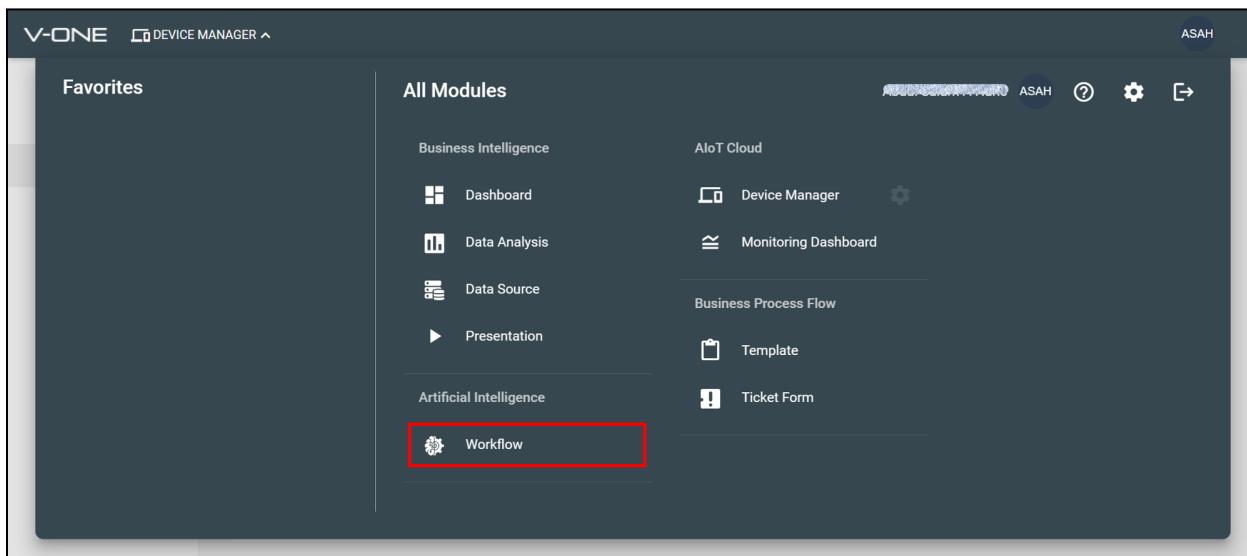
Lastly, click on the device and you can see the **deviceID** that will be used in the project code later.

Tips: Use the copy button function to make your life easier.

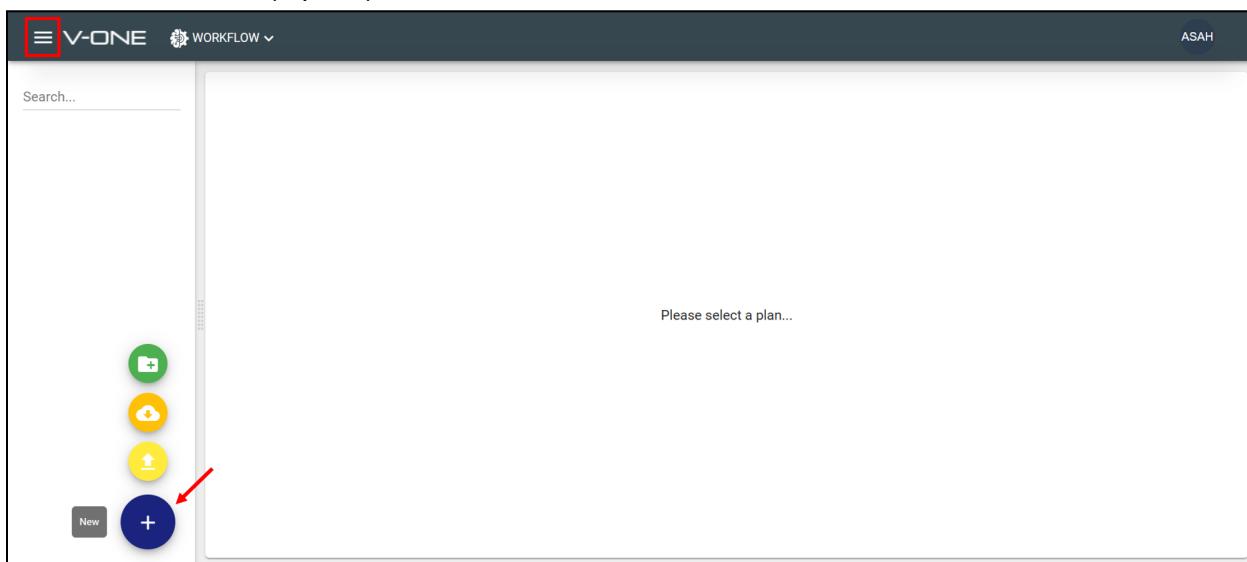
Workflow

Workflow is used for **email notifications** that are triggered once any sensor readings reach the threshold value. It becomes very useful, especially when doing IoT projects to alert the owner about the current condition of the projects.

Go to “Workflow”.

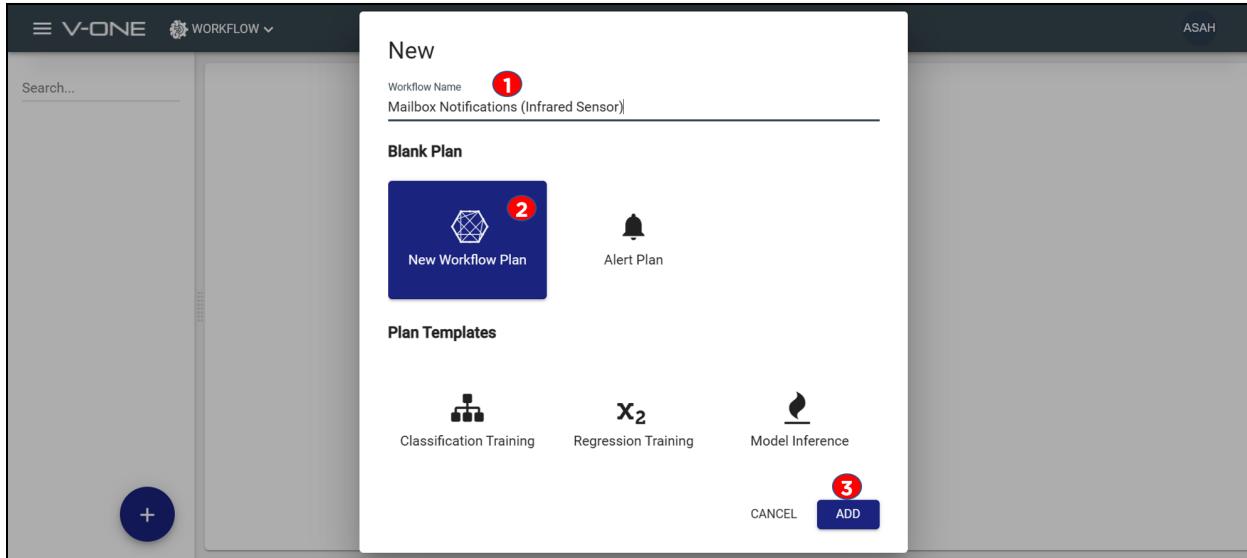


Click on the sidebar (top left) and then click the “+” button to add a new workflow.

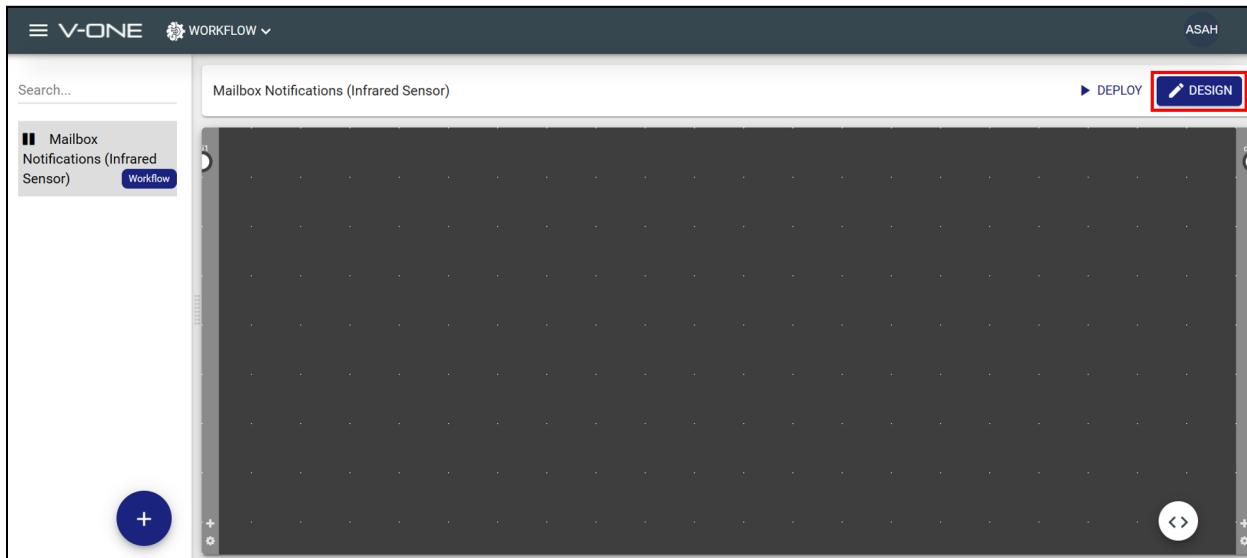


Insert **workflow name** and then click the “**Add**” button.

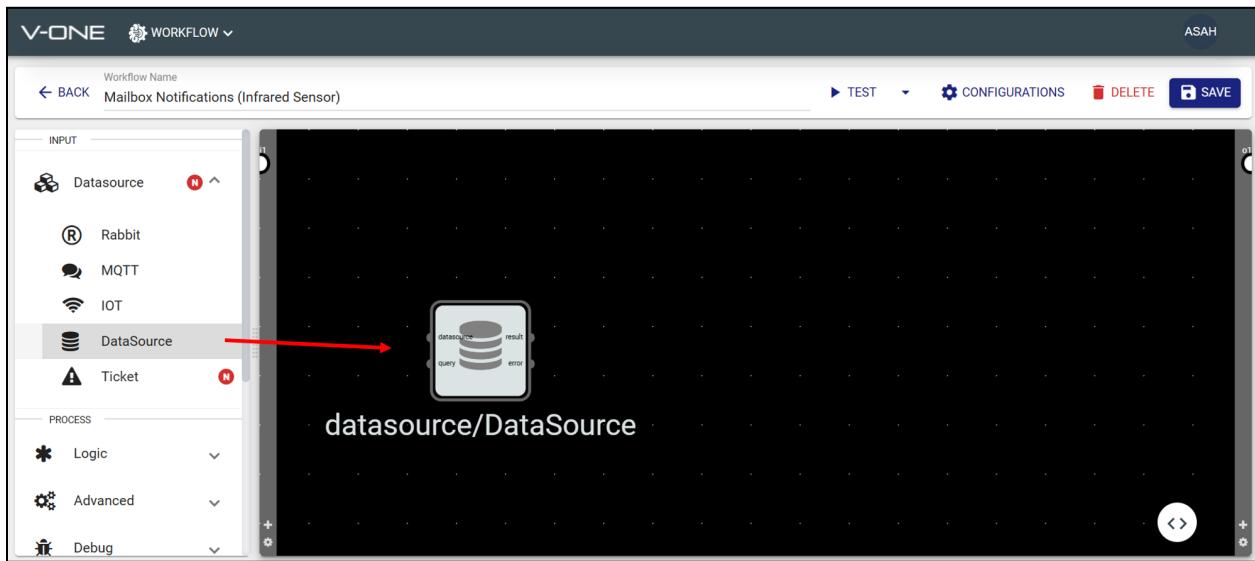
Tips: ‘Mailbox Notifications (Infrared Sensor)’ is just an example of the workflow name. You can put anything and it can be general such as ‘Project 1 & Project 2’ since one workflow can be used for multiple sensors/projects.



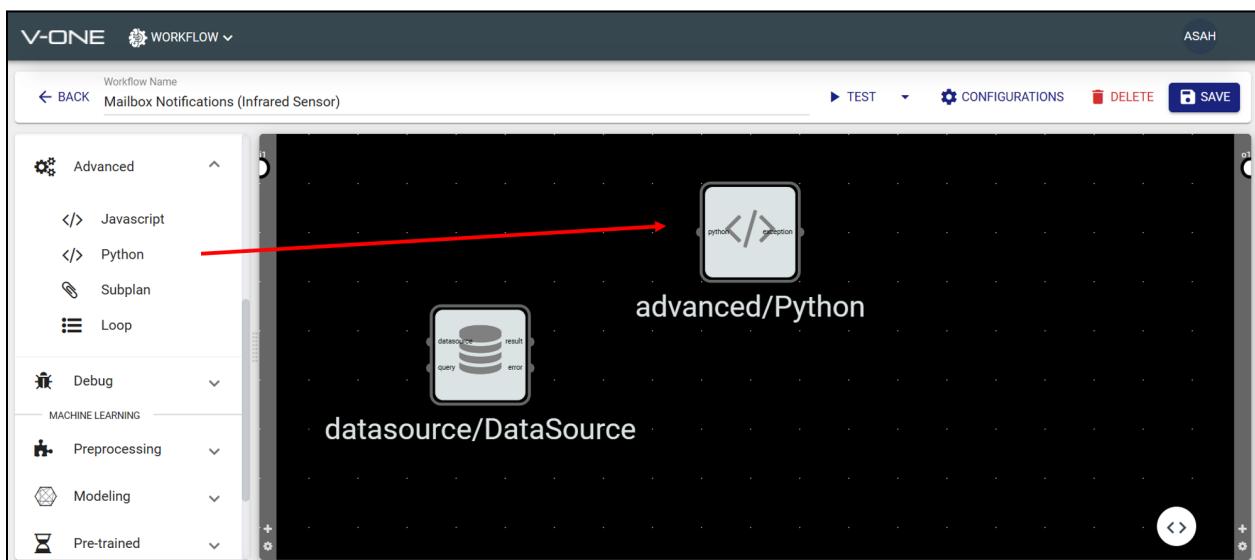
Click the “**Design**” button to start creating the workflow.



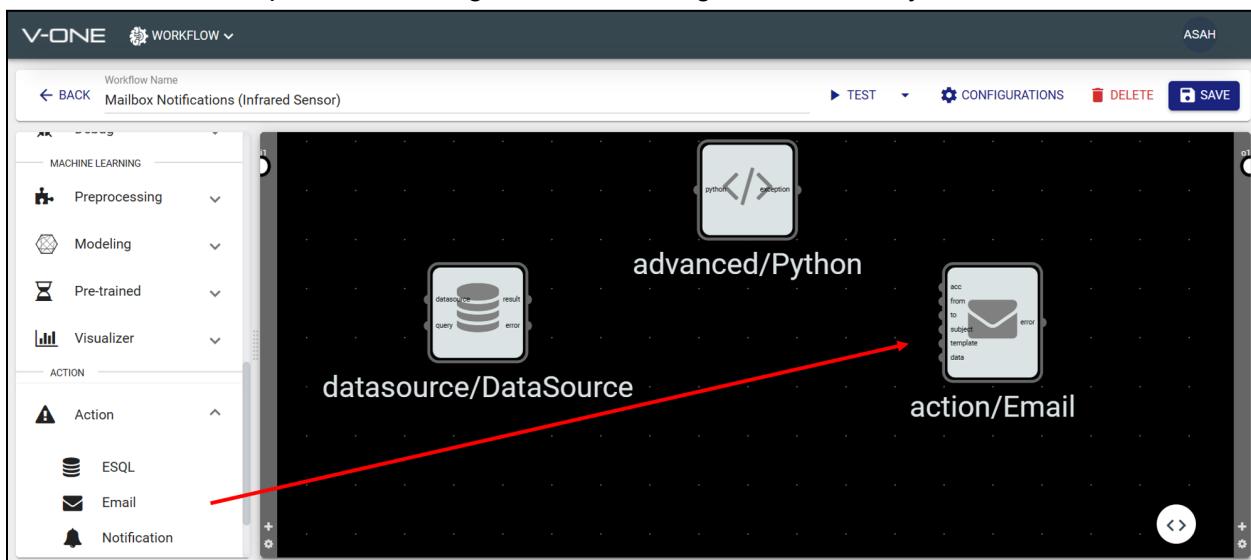
Under **Datasource** drop-down list, drag “**Datasource**” to the workflow space.



Then, under the **Advanced** drop-down list, drag “**Python**” to the right side of the Datasource icon.



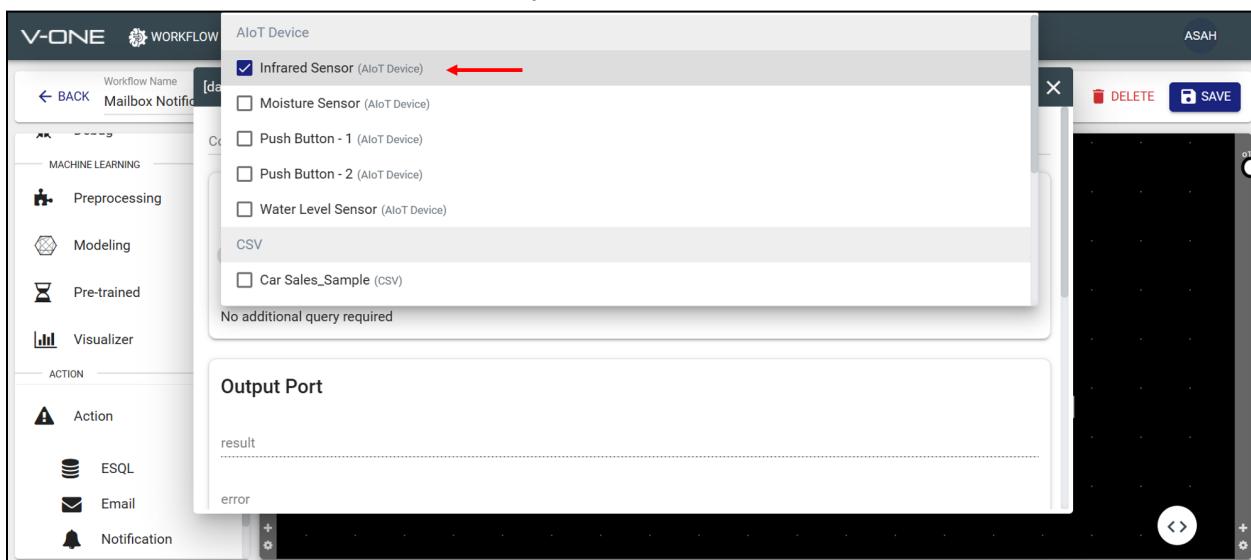
From the **Action** drop-down list, drag “Email” to the right side of the Python icon.



Click on the **Datasource icon**, and a small window will pop up. Under “Input Port”, choose the related sensor for the datasource. The workflow will trigger based on this datasource value.

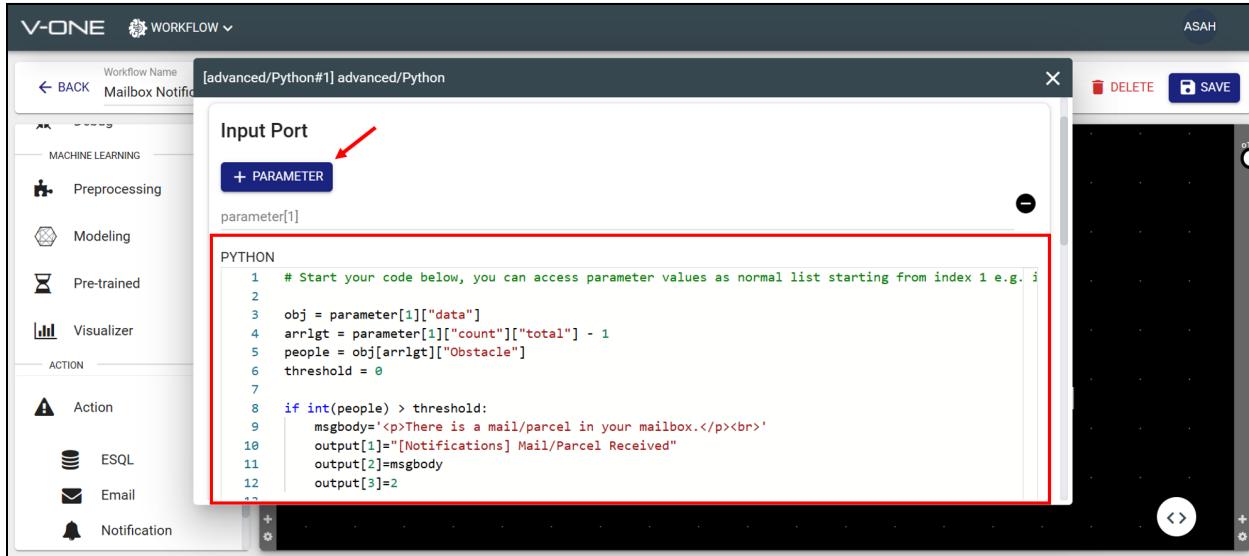
Close the Datasource window.

Tips: One datasource for one sensor only!



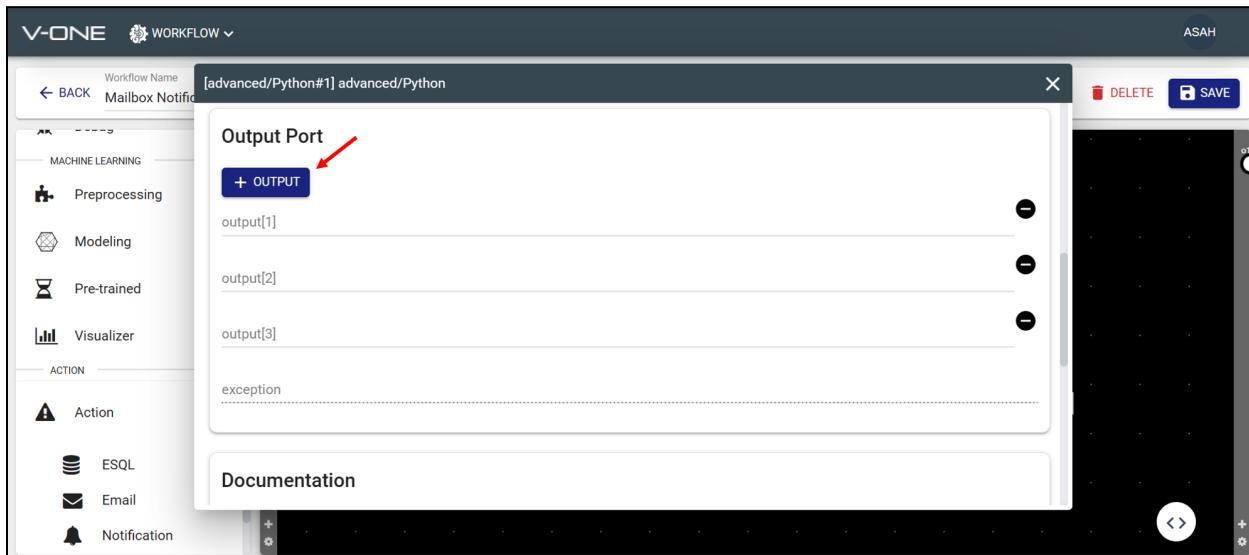
Then, click on the **Python icon**. Under **Input Port**, add **one** parameter by clicking on the “Parameter” button.

For workflow code, you can just copy it from our [GitHub page](#). It is also provided in **Part 3** later for each IoT project.



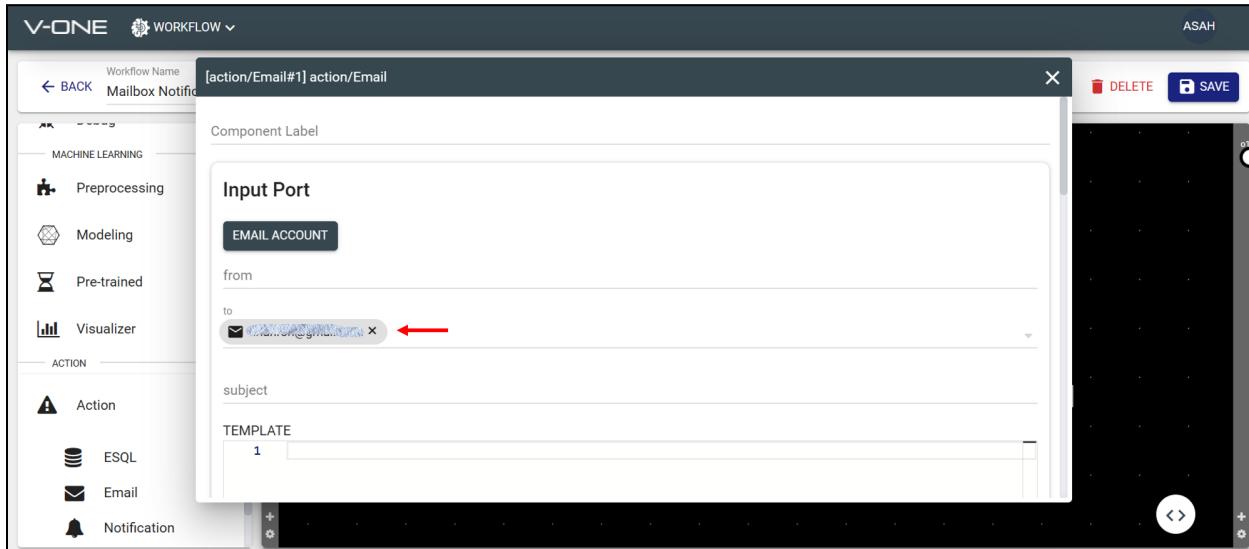
Scroll down and you will see the “Output Port” part. Add **three outputs** by clicking the “Output” button three times.

Close the Python window.



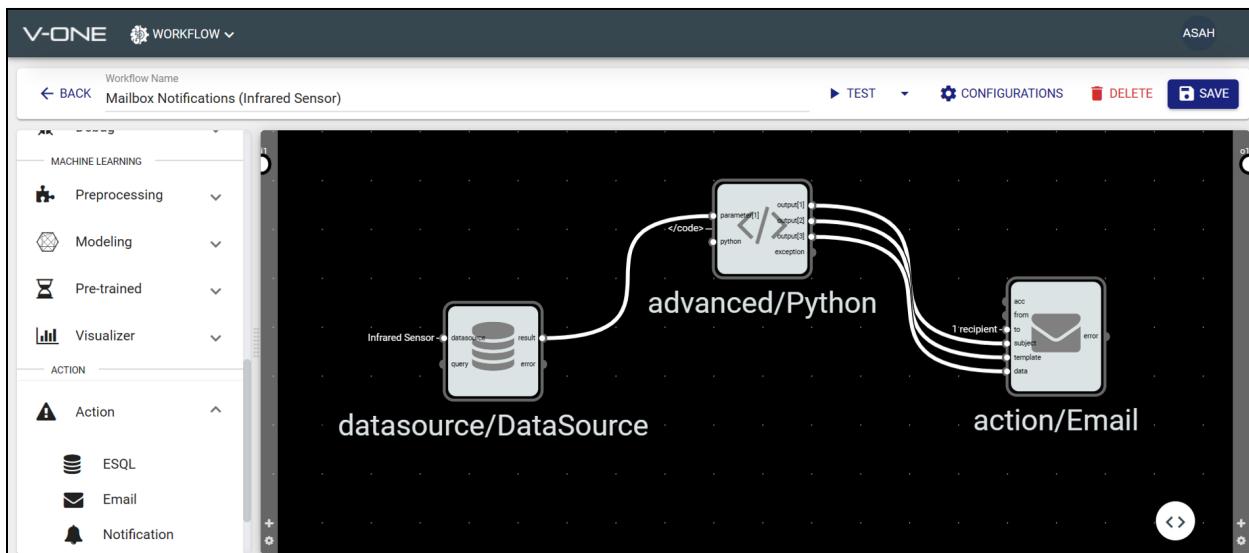
Open the **Email window**. Under Input Port, type in **your email** under the “to” section, and leave the “from” section empty.

Note: Press **Enter** once you have done type in your email address.



Lastly, create connections between the boxes as follows.

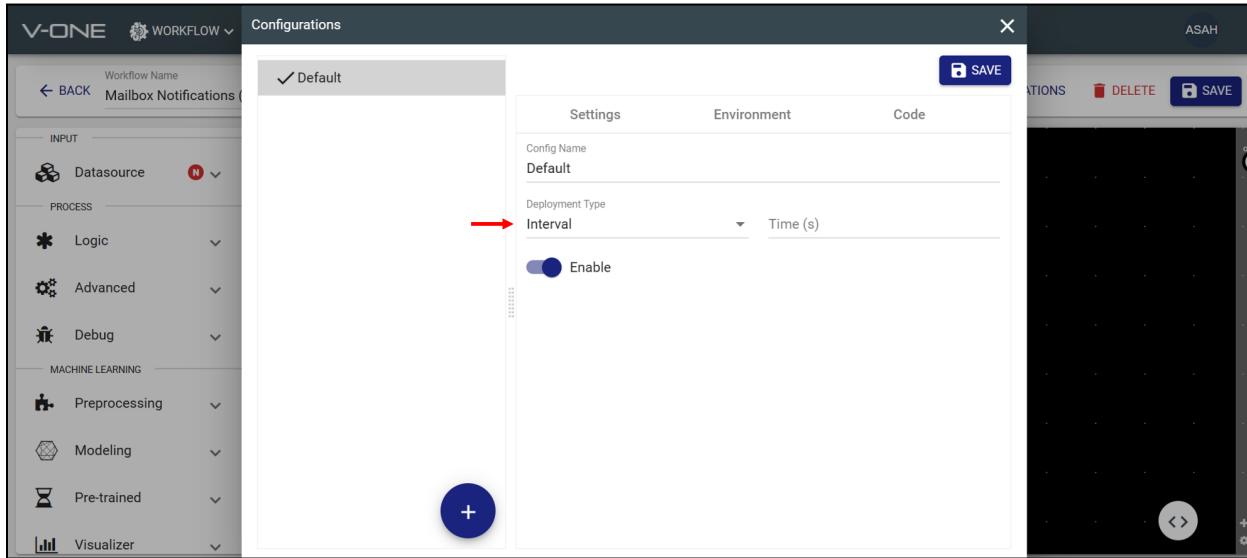
- result > parameter [1]
- output [1] > subject
- output [2] > template
- output [3] > data



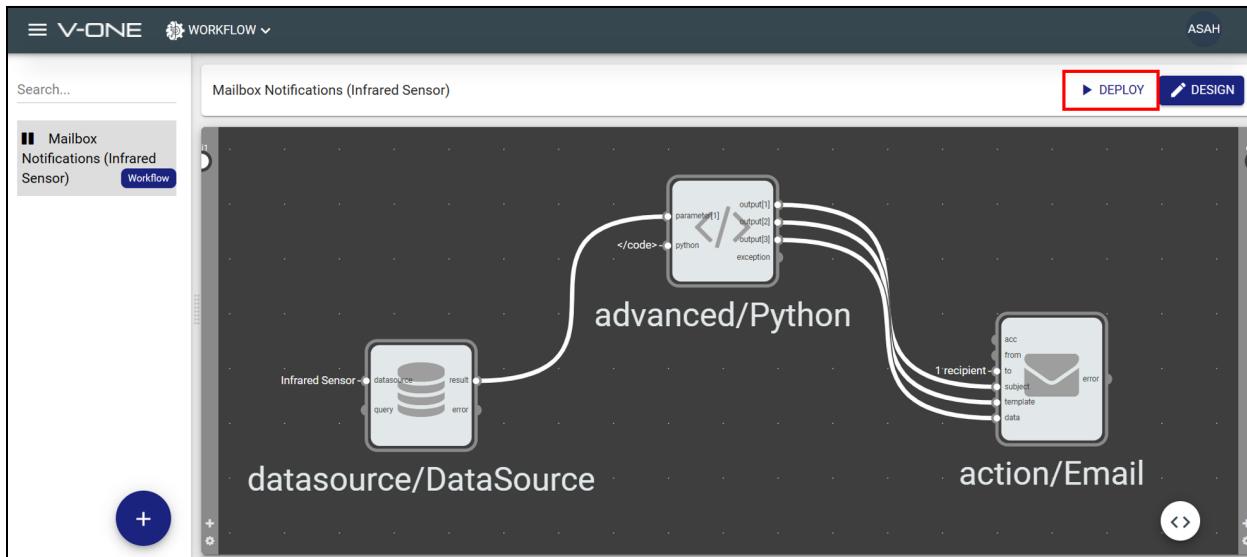
Open the configurations and you can change the “**Deployment Type**” according to your project preferences.

Tips: **Interval** and **Scheduled** will always be the best options since you don't need to always deploy the workflow when running the IoT projects.

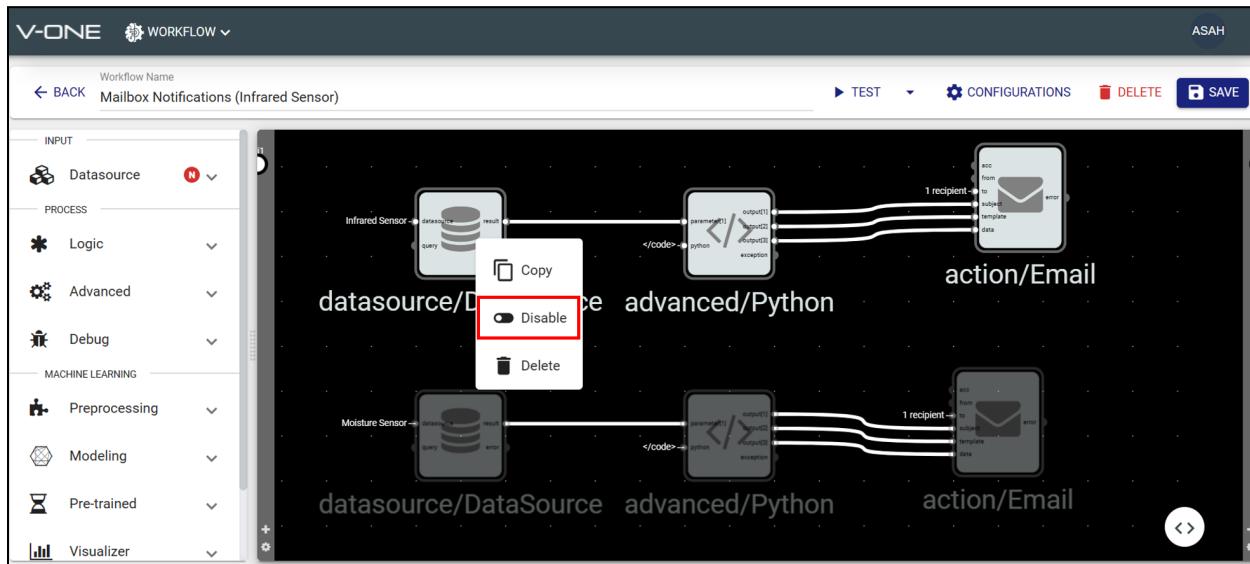
Don't forget to click the **save** button when you are done.



Note: Before you click the “Deploy” button, please make sure that **your project is already running**. You can just create all the workflow first and come back here to deploy it later once your IoT project is complete.



For the free V-ONE account, only **three workflows** are allowed. However, you can create multiple lines for each sensor/project in one workflow but you need to **disable** other lines that are not being used.



Tips: If you decide to use more than one line in a single Workflow, you can rename each block so that you will not confuse about which block is for which projects.

This screenshot shows a detailed view of a component within the V-ONE Workflow interface. The component label is '[advanced/Python#1] advanced/Python'. The component is titled 'Mailbox Notifications (Infrared Sensor)'. The 'Input Port' section shows a parameter named 'parameter[1]'. The 'PYTHON' section contains the following code:

```

1 # Start your code below, you can access parameter values as normal list starting from index 1 e.g. parameter[1]
2
3 obj = parameter[1]["data"]
4 arrlgt = parameter[1]["count"]["total"] - 1
5 people = obj[arrlgt]["Obstacle"]
6 threshold = 0
7
8 if int(people) > threshold:
9     msgbody='<p>There is a mail/parcel in your mailbox.</p><br>'
```

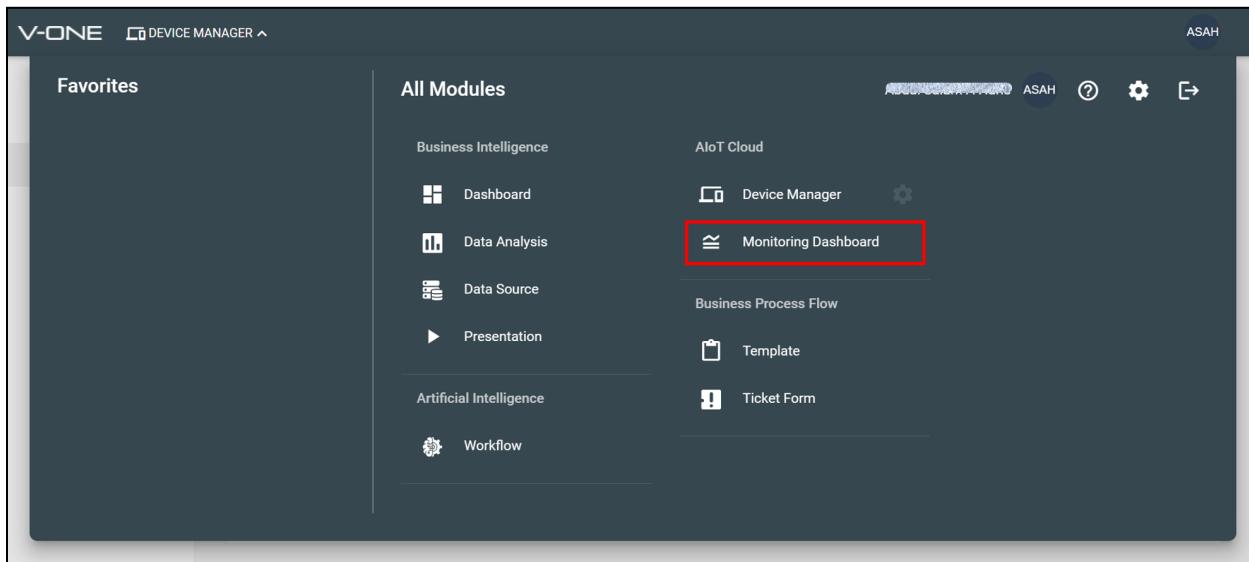
Monitoring Dashboard

For this section, every step is based on the sample IoT Projects to ease your understanding.

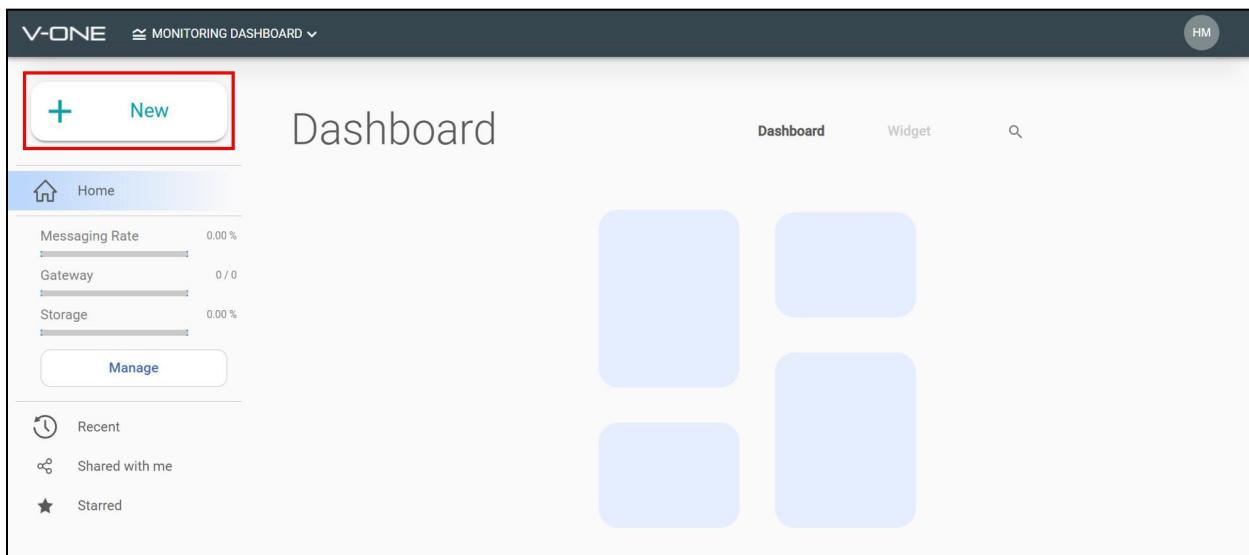
Project 5: Smart Agriculture

This project has both sensors and an actuator, so we will refer to it while learning how to set up the dashboard and widget.

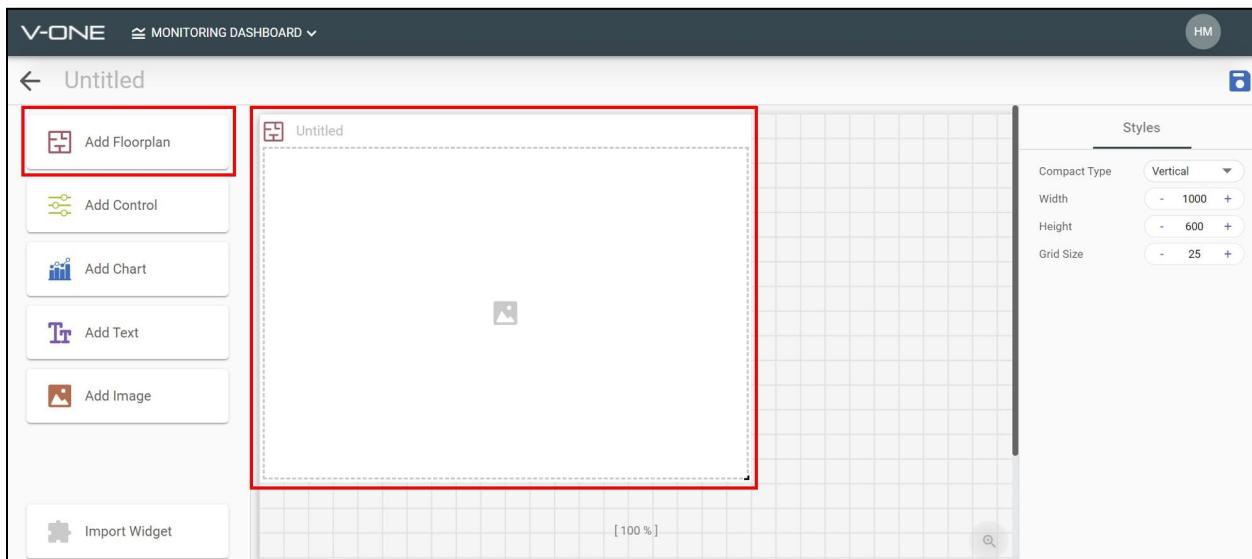
Make sure that you already registered the gateway and all devices used in this project. Then, go to “Monitoring Dashboard”.



Create a new “Dashboard”.



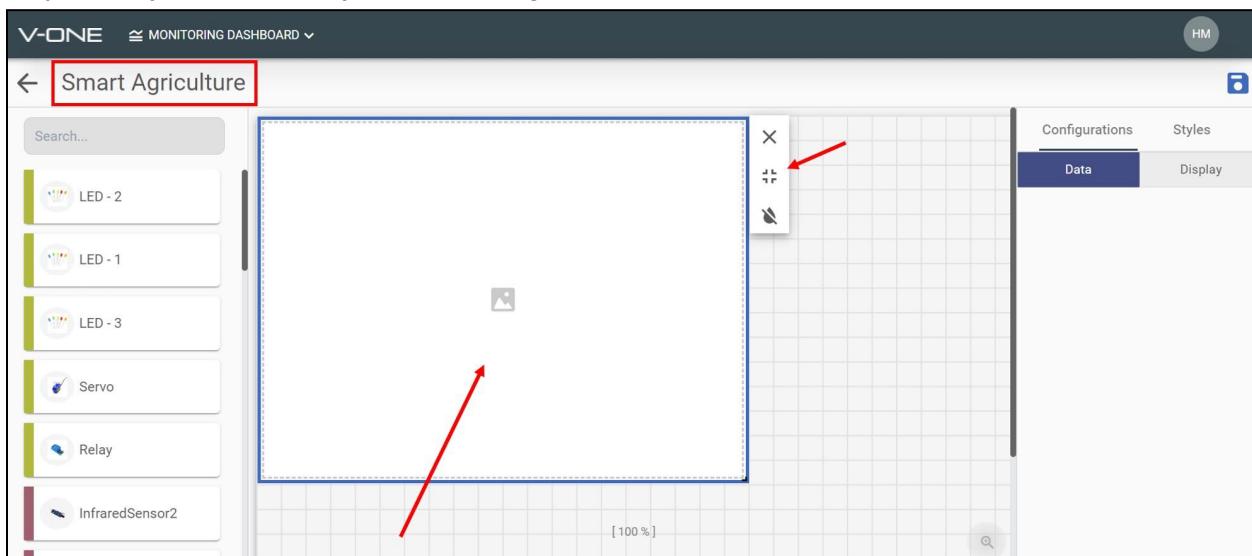
Click “**Add Floorplan**” and you will see a blank white box appears.



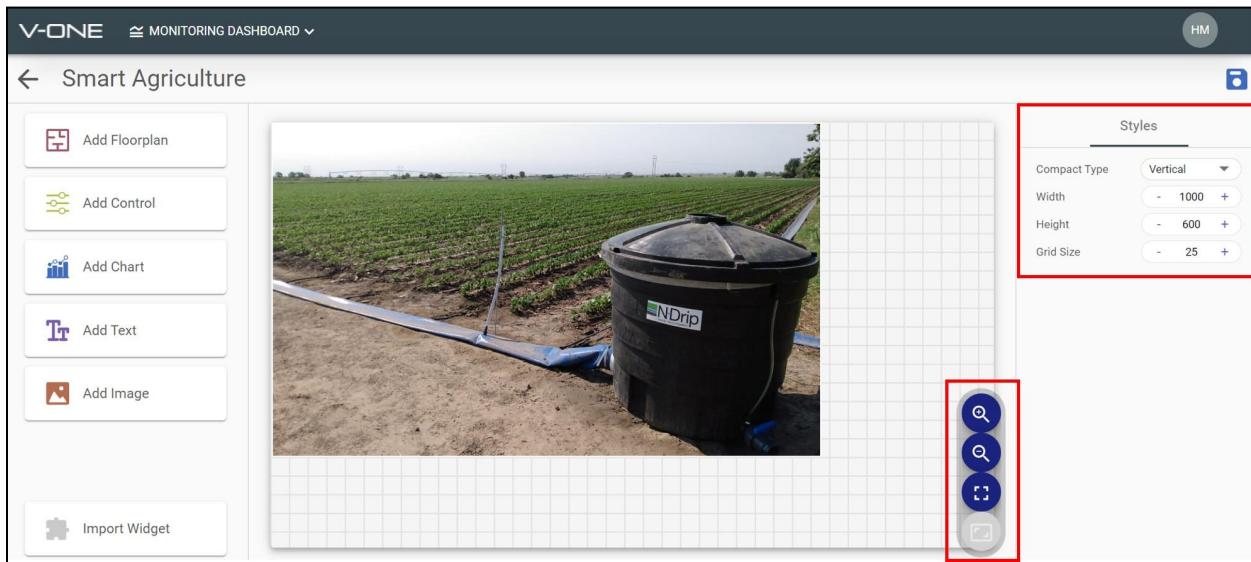
Give your monitoring dashboard a nice **title**. For this example we're using **Project 5** which is related to agriculture; so let's name our dashboard **Smart Agriculture**.

Tips: You can click on the **fullscreen icon** to hide the small title since the dashboard already has a main title.

Click the **gallery icon** to upload a background image. Kindly choose an image that is suitable for your project. As for Project 5, the image of a farm with a water tank is the most suitable.



Click on the grid background and you can play around with the dashboard settings in the red box shown below.

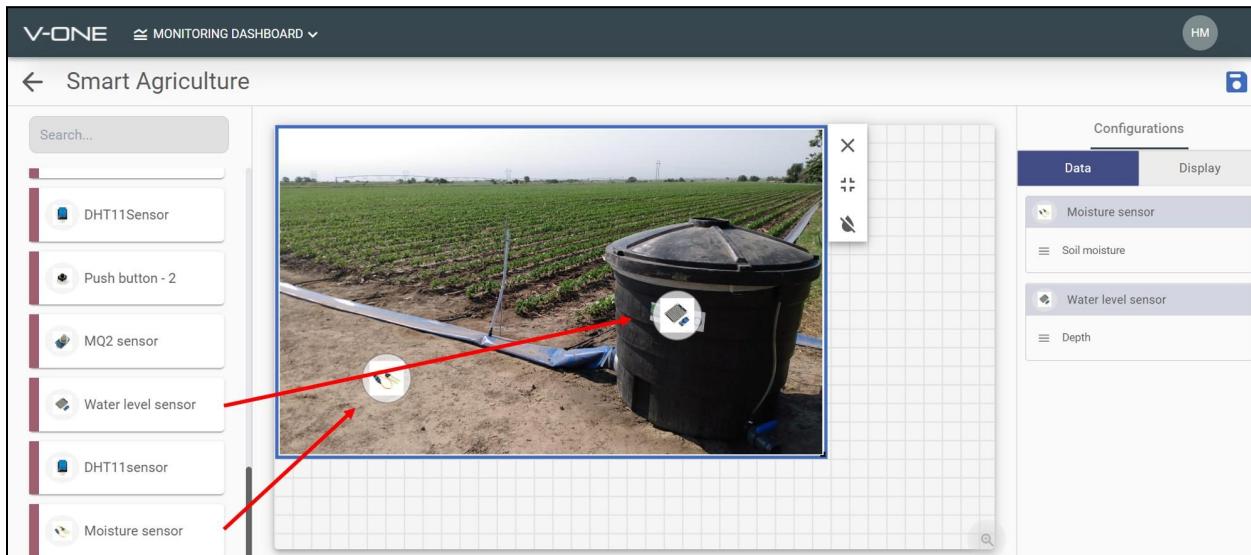


To **add sensors**, click on the uploaded image and you will see all the available devices on the left side of the screen. Drag each sensor that will be used in the project. To make it more interesting, you can put the sensor widget on the area of the image that it wants to measure.

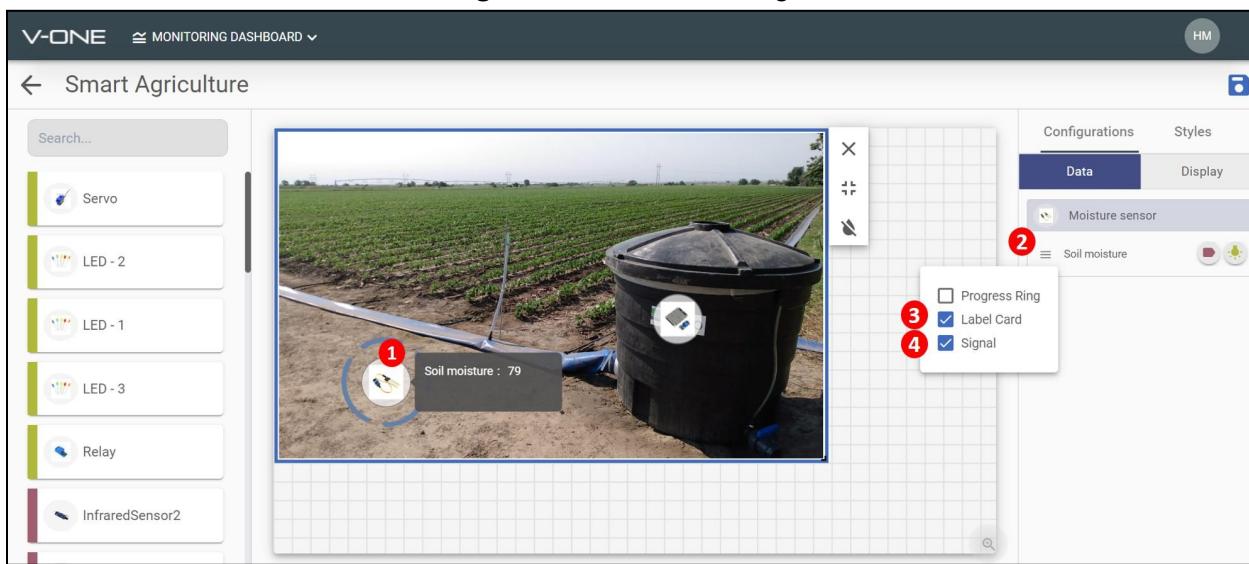
For example (**Project 5**):

Moisture Sensor - Place it on the soil area to indicate the soil moisture.

Water Level Sensor - Place it on the tank to indicate the water level in the tank.

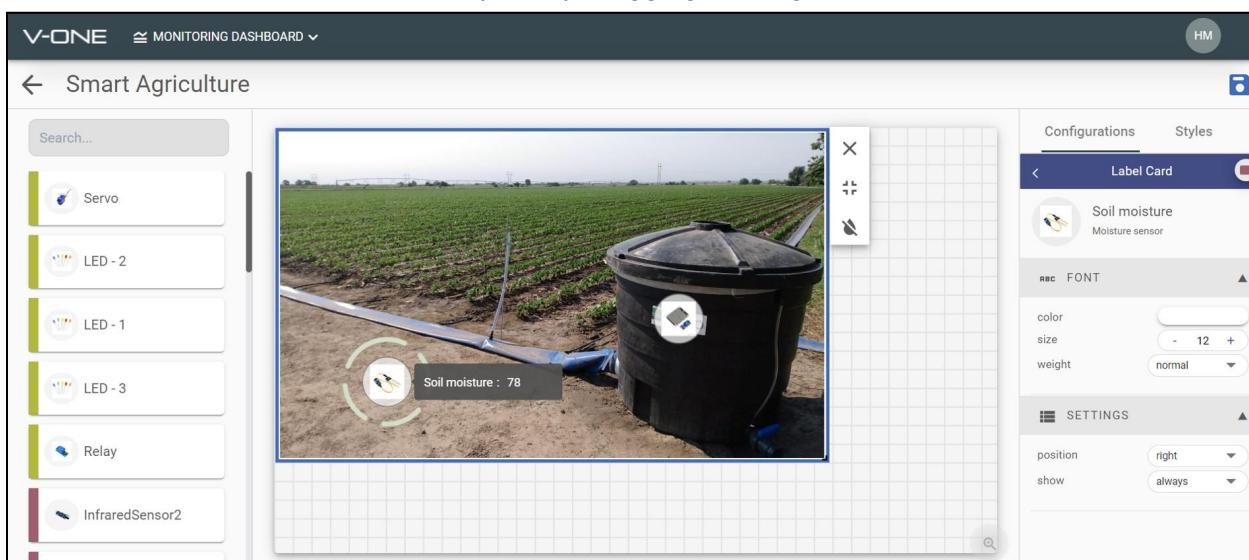


Then, enable the **Label Card** and **Signal** for the sensor widget.

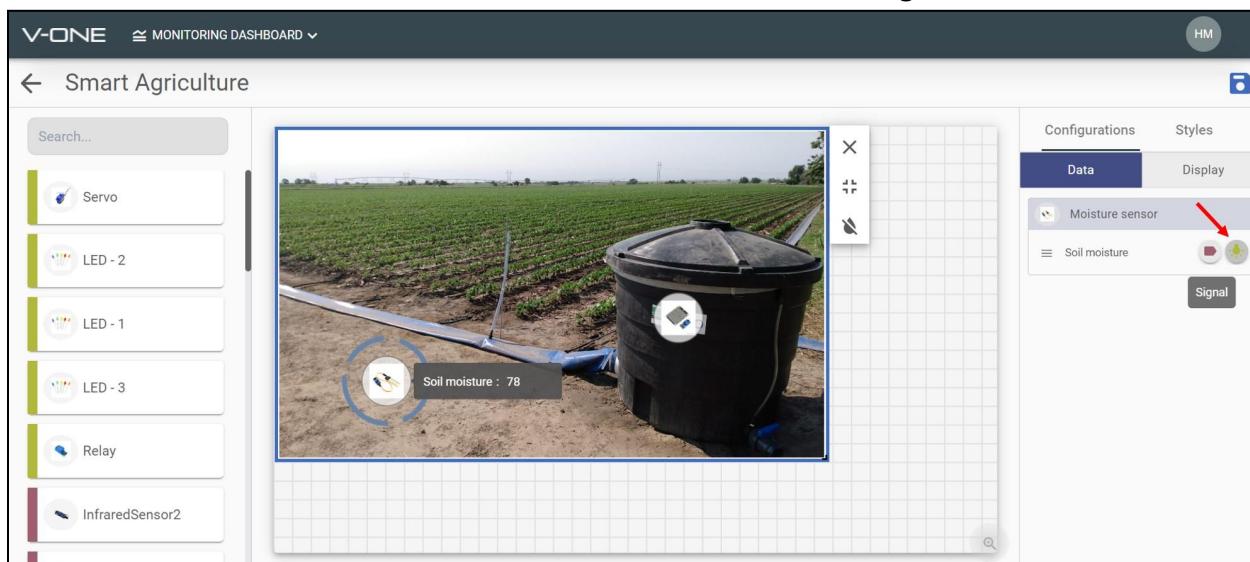


Go to the Label Card settings if you like to change anything.

Tips: You can resize the label card by easily dragging the edge of the label card.



Go back to the “Data” tab of the soil moisture sensor, and click the **Signal icon**.



Here, you can set the validation to turn on the signal or LED indicator. For the Signal settings, you can change the color of the LED indicator and set the validation or threshold value.

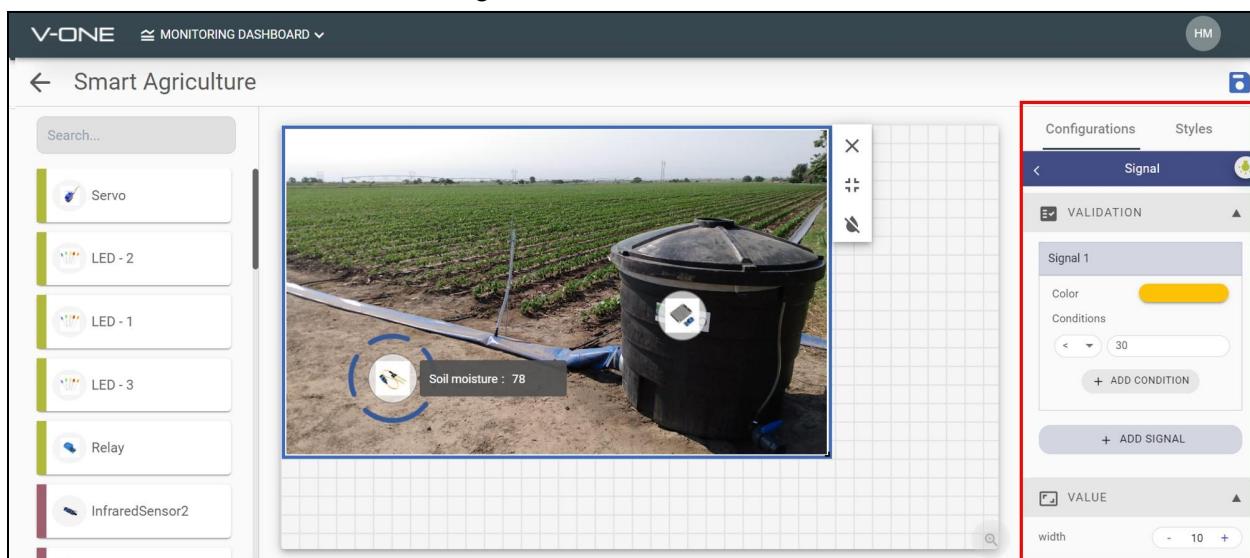
For example:

No LED - The soil moisture reading is above 30%

Yellow Color LED - The soil moisture reading is less than 30%

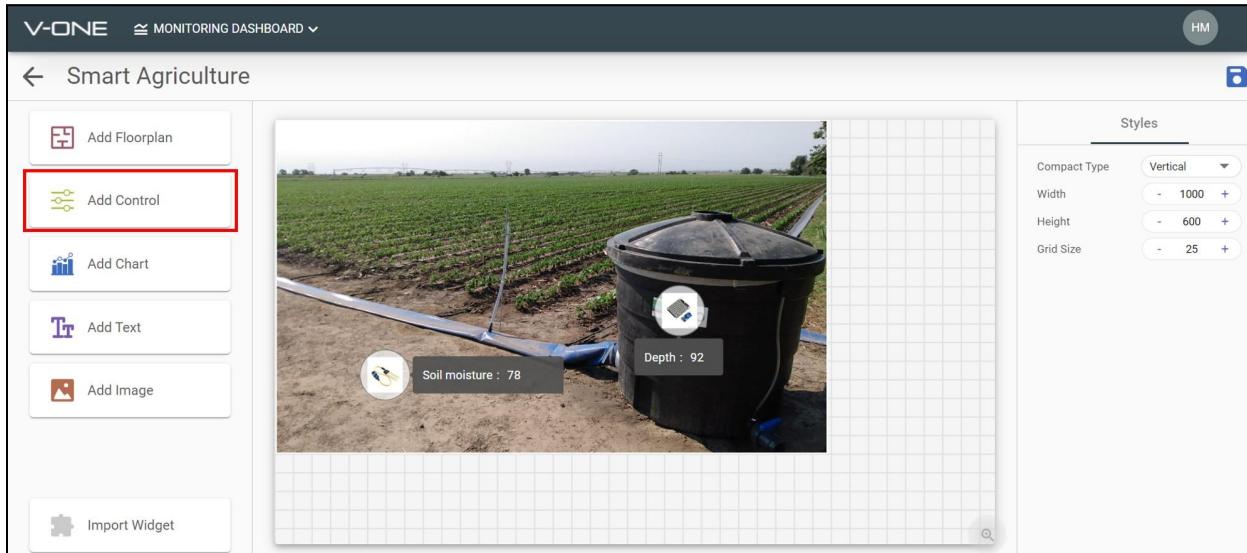
Change the width value to get a larger indicator.

Note: You can add more than one signal for each sensor!

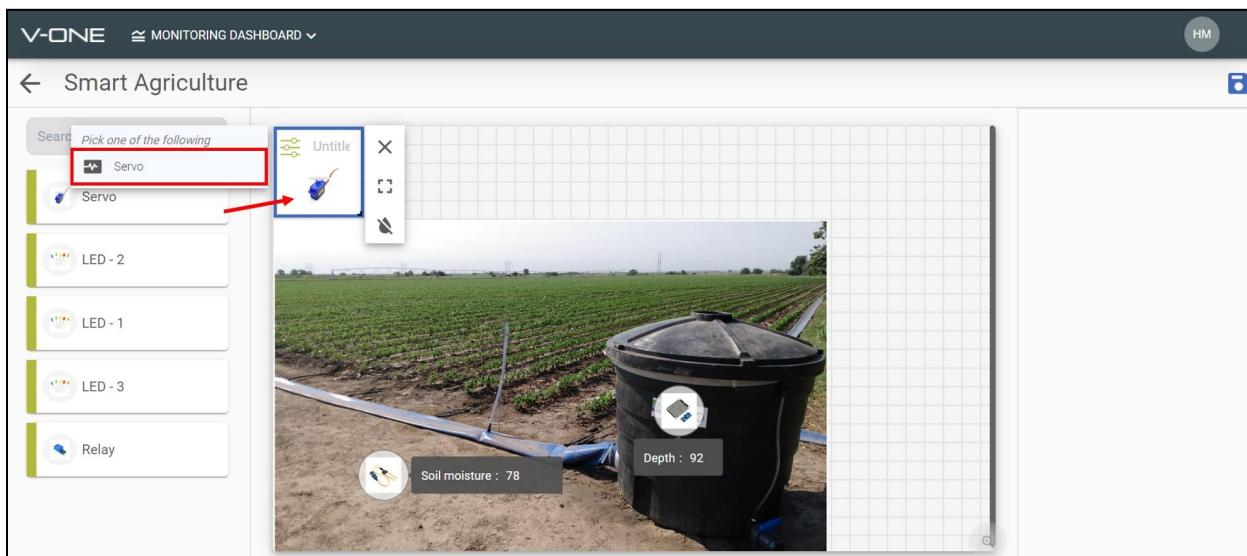


Done with the sensors part, let's move on to the actuator part. Click “**Add Control**”.

Note: Widget for actuators cannot be inserted in the background image. So, you can place it in the grid area.



For the actuator in **Project 5**, a servo is used to symbolize the water sprinkler switch. Drag **Servo** to the actuator box and pick “**Servo**” for the options.

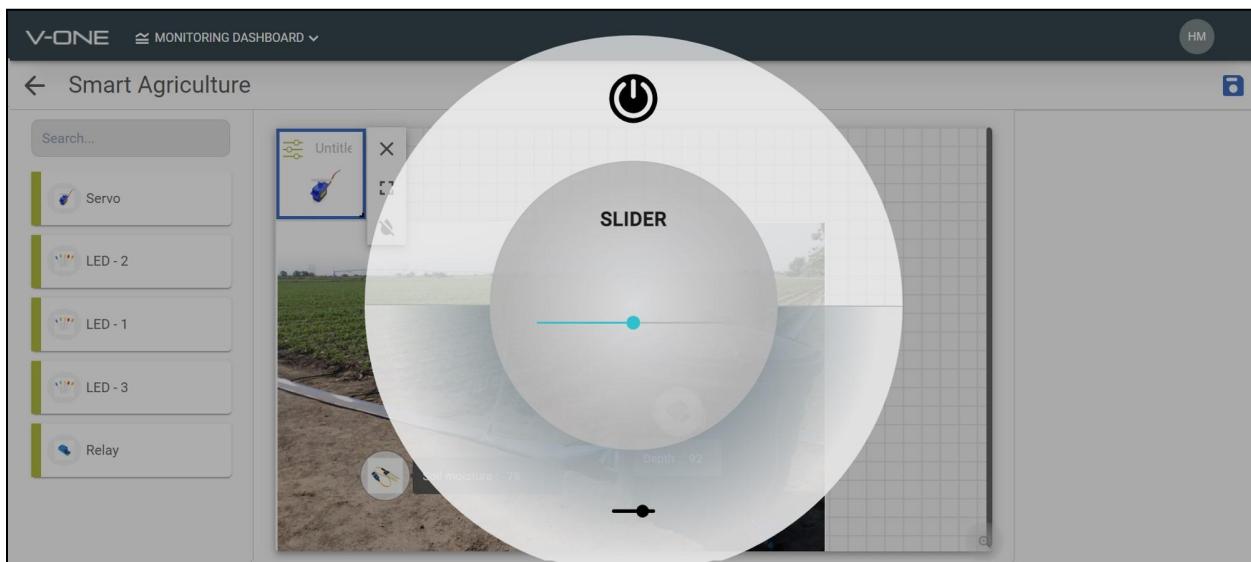


There are two choices of widgets available for the servo.

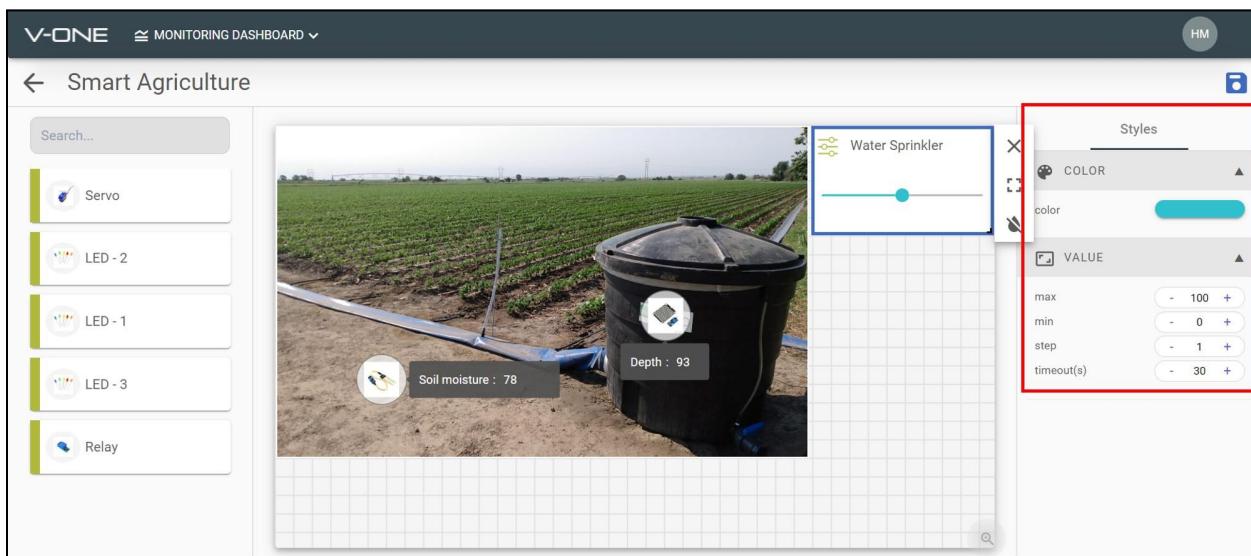
Slider - slide to simply change to any value.

Button - fix at one value to turn ON/OFF.

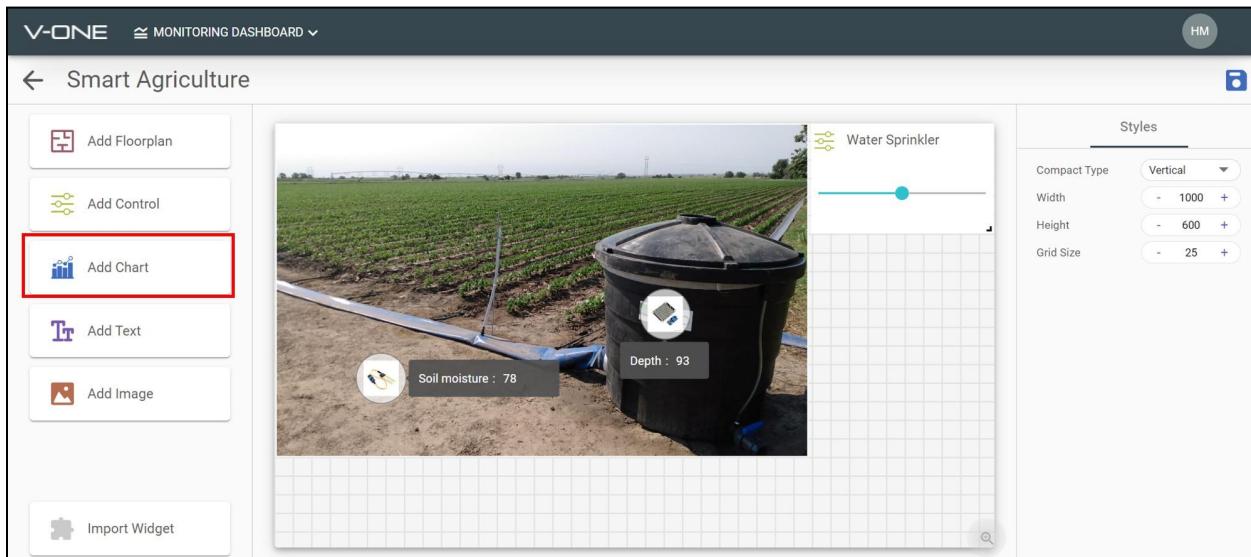
Let's proceed with Slider first.



Click on the slider widget. There you can change the color and value settings. Since this servo support 180 degrees of rotation, change the **max value** to **180**.



Last but not least, let's add a chart to the dashboard so that you can observe and analyze all the data collected in this project.

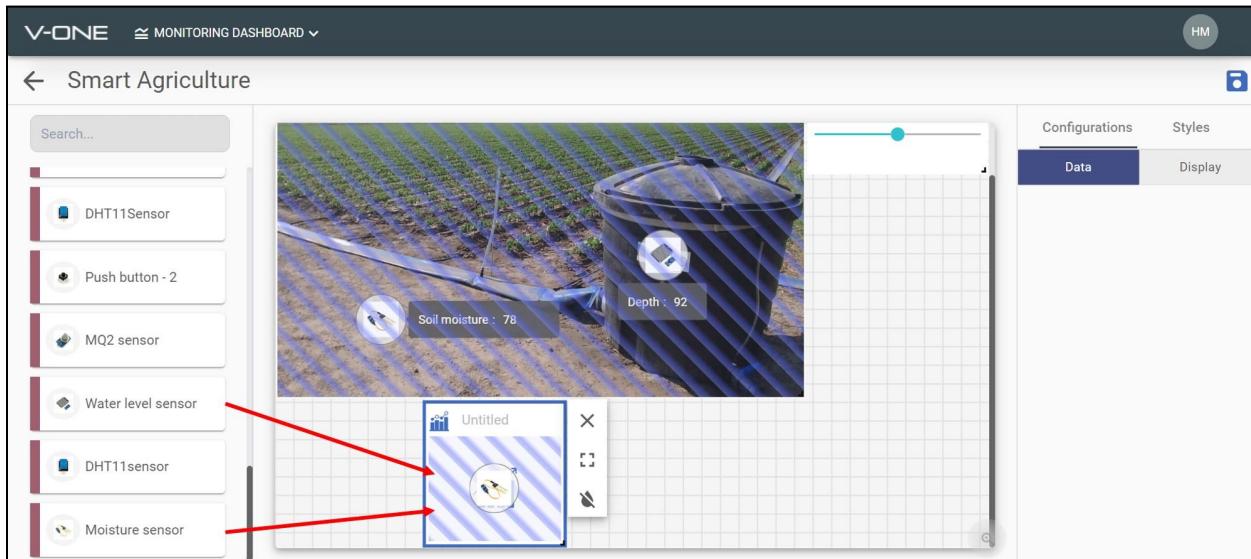


Choose the “**Series**” type chart.

Tips: For parameter like the water level sensor, the **gauge** will be a good option.

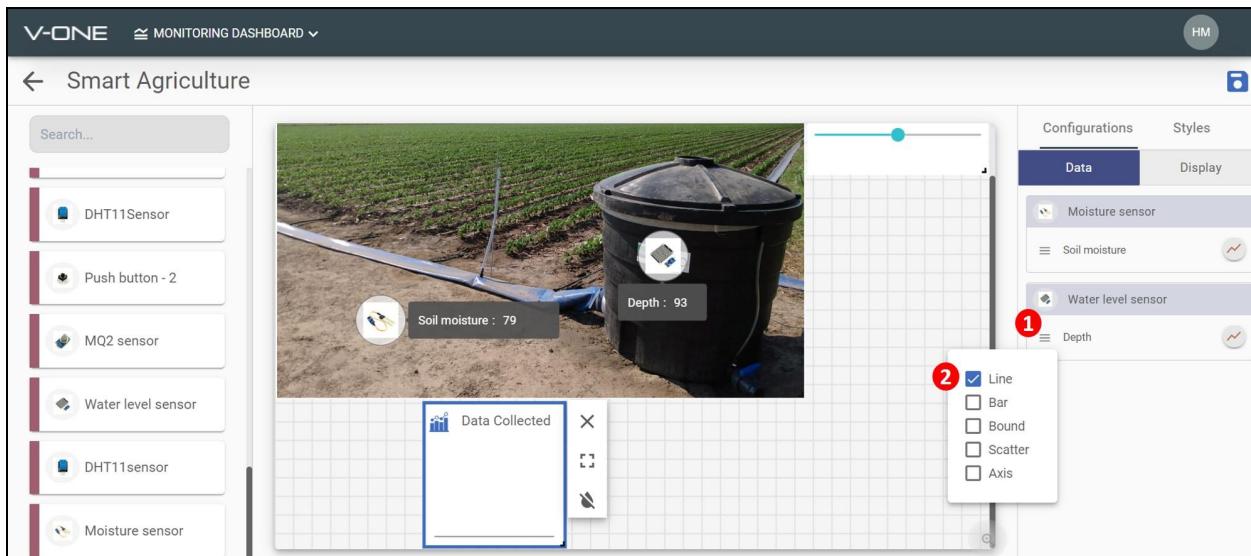


Similar to the actuator setup before this, drag any sensor to the chart box. You can drag more than one type of sensor into the chart and it will automatically plot the graph for you.



Enable the “Line” graph for both sensors.

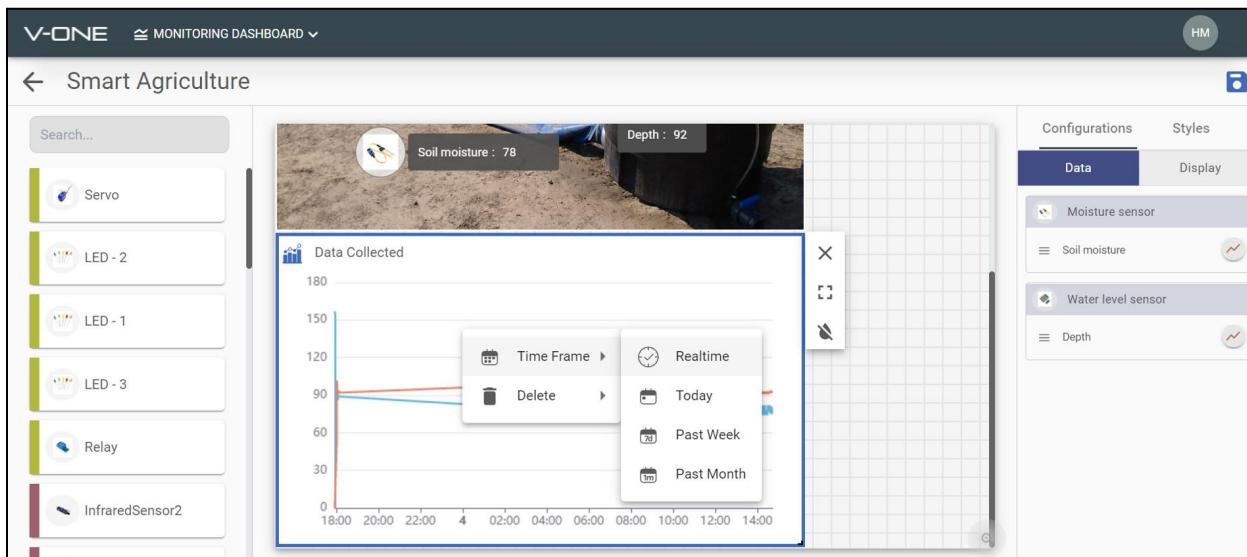
Tips: You can also customize the **y-axis** range by enabling the “**Axis**”.



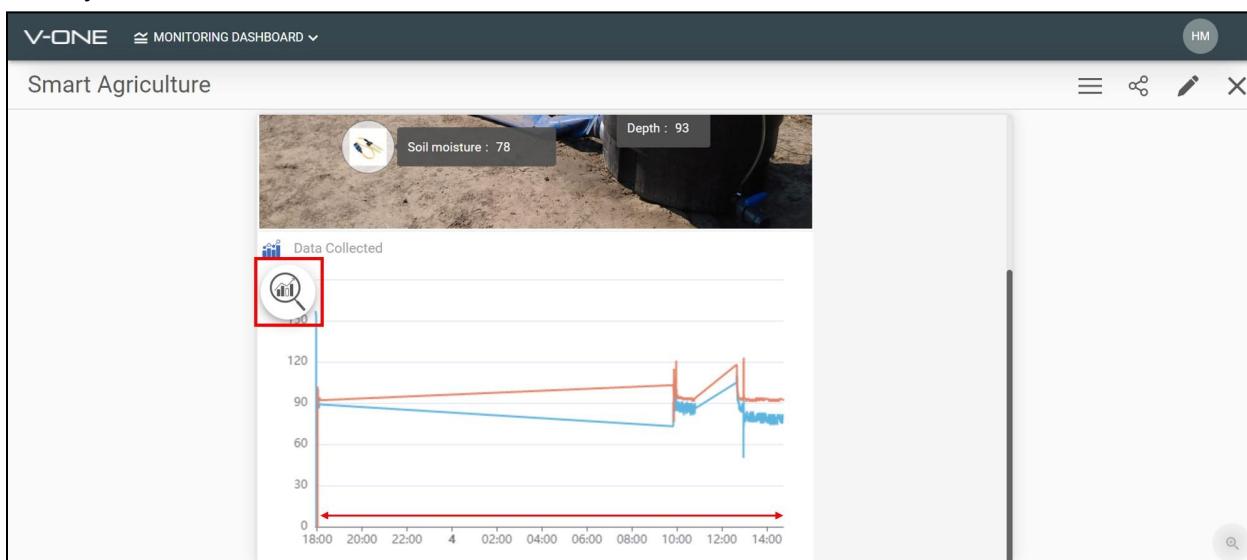
There you can see that a graph is created automatically, plotting real-time readings of the sensors.

Right-click on the chart to change the “Time Frame”. Then, click the save icon.

Note: You can change the color of each graph in the “Line” chart settings (line graph icon).



In view mode, you can zoom on any part of the chart by clicking on the magnifying glass icon. Scroll your mouse to zoom in and out.



Some beneficial connectivity features of the V-ONE dashboard:

- The sensor widget will have a red icon on the top left if the sensor is disconnected.
- The slider will turn into a grey color and you cannot move the slider if the servo is disconnected.
- The color of button widget will become dim if the actuator is disconnected.

Congratulations! You have successfully built the final dashboard. Now you can proceed to the next part to start creating IoT projects.



Part 2

Arduino IDE

Github Page

For this IoT kit, we will be using **Arduino IDE** software to upload the code. For those that still don't have the Arduino IDE yet, you can download it [here](#).

Note: Please install the **latest version (2.0.X)**.

The screenshot shows the Arduino IDE 2.0.3 download page. On the left, there's a teal icon with two overlapping circles and a plus sign. Below it, the text "Arduino IDE 2.0.3" is displayed. A paragraph describes the new features of the release. Below that, a link points to the "Arduino IDE 2.0 documentation". Another section mentions "Nightly builds" with bugfixes. Under "SOURCE CODE", it says the IDE is open source and links to GitHub. On the right, a teal sidebar titled "DOWNLOAD OPTIONS" lists download links for Windows (Win 10 and newer, 64 bits), Windows (MSI installer), Windows (ZIP file), Linux (AppImage 64 bits (X86-64)), Linux (ZIP file 64 bits (X86-64)), macOS (Intel, 10.14: "Mojave" or newer, 64 bits), and macOS (Apple Silicon, 11: "Big Sur" or newer, 64 bits).

Kindly visit our GitHub page and **download the ZIP file** for this IoT kit.

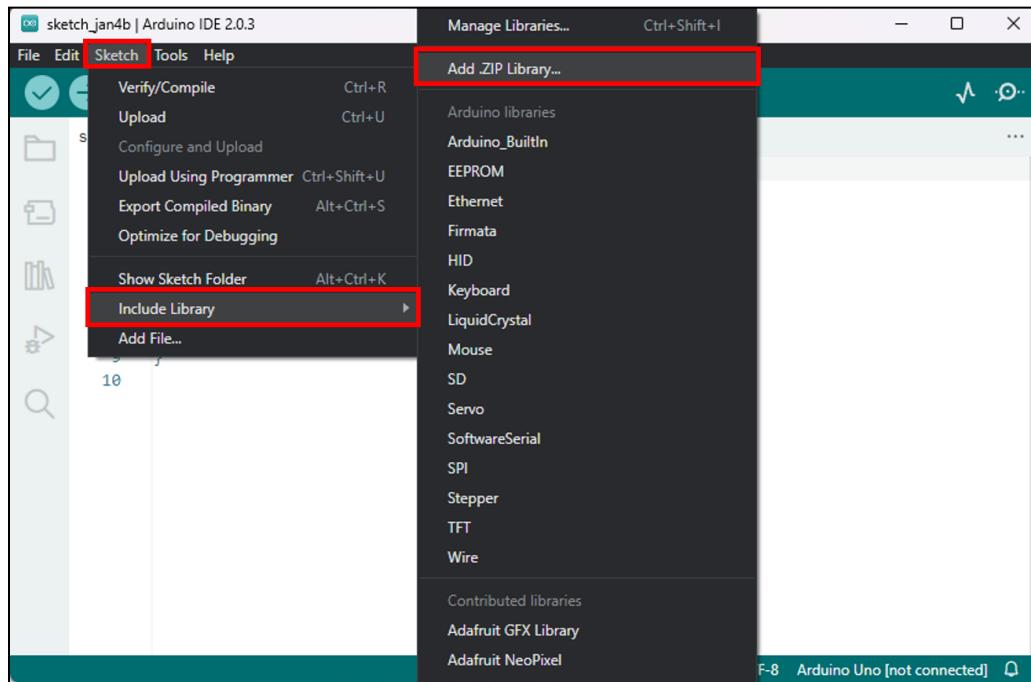
- [GitHub - CytronTechnologies/IoT-Kit-V-One](#)

The screenshot shows the GitHub repository page for "CytronTechnologies / IoT-Kit-V-One". The repository has 1 branch and 0 tags. The main file listed is "NorHairil Add files via upload". On the right, there's a "Code" dropdown menu with "Clone" options (HTTPS, SSH, GitHub CLI) and a "Download ZIP" button, which is highlighted with a red box. Other sections include "About" (no description), "Releases" (no releases), and "Packages" (no packages). The repository has 0 stars, 6 watchers, and 0 forks.

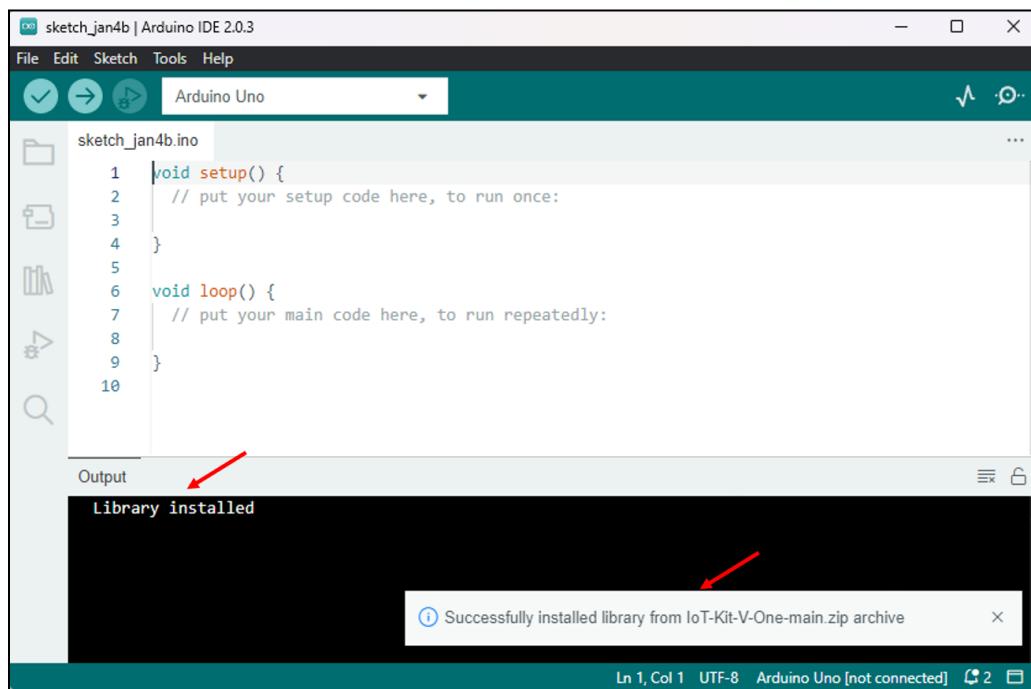
All resources for the IoT Kit are included in this ZIP folder including **V-One libraries**, **sensor code**, **5 IoT projects code**, and **workflow code** (used for email notifications).

V-One Libraries

Open Arduino IDE. Go to **Sketch > Include Library > Add ZIP Library**. Choose the ZIP file that you have downloaded. The file name should be “**IoT-Kit-V-One-Main**”.



You have successfully added the ZIP file if you got something like this.



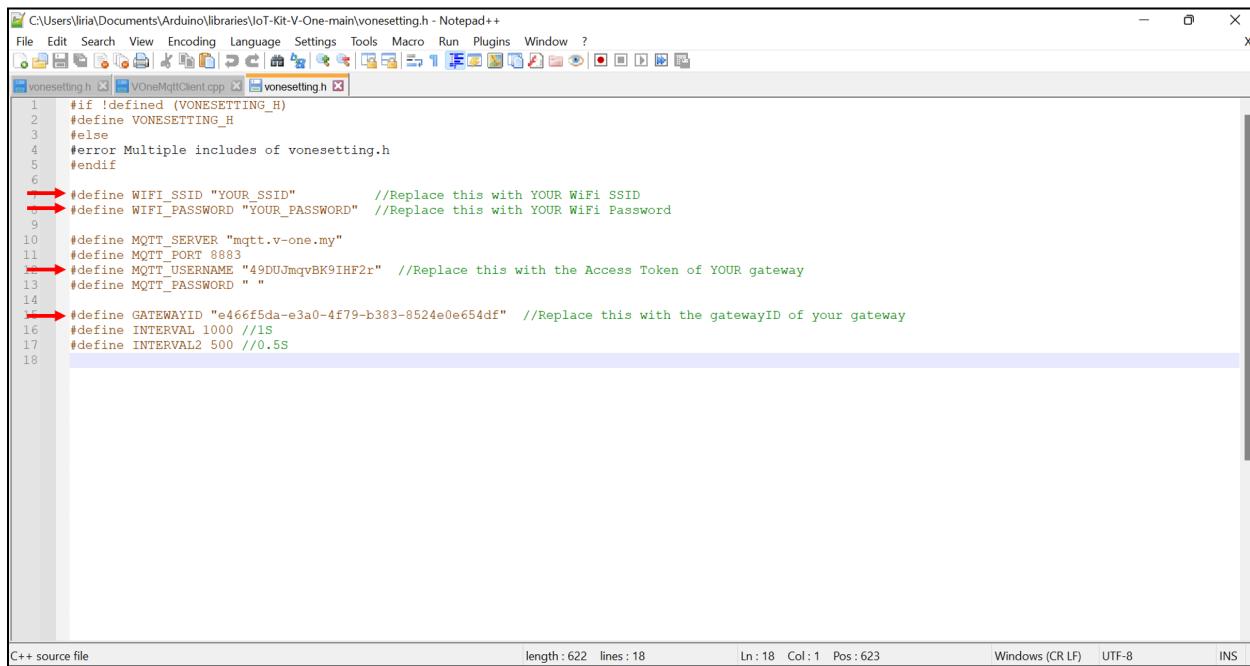
Navigate to the V-One libraries to modify the **WiFi and gateway credentials**. The libraries for Arduino IDE are usually stored in this path:

Documents\Arduino\libraries\IoT-Kit-V-One-main

Open the “**vonesetting**” file using Notepad or any text editor. Change the **WiFi SSID**, **WiFi Password**, **Gateway Access Token**, and **GatewayID**.

Note: Access Token and GatewayID can be found in the V-One platform at **Device Manager > Gateways**.

Then, click the save button and close the file.



```

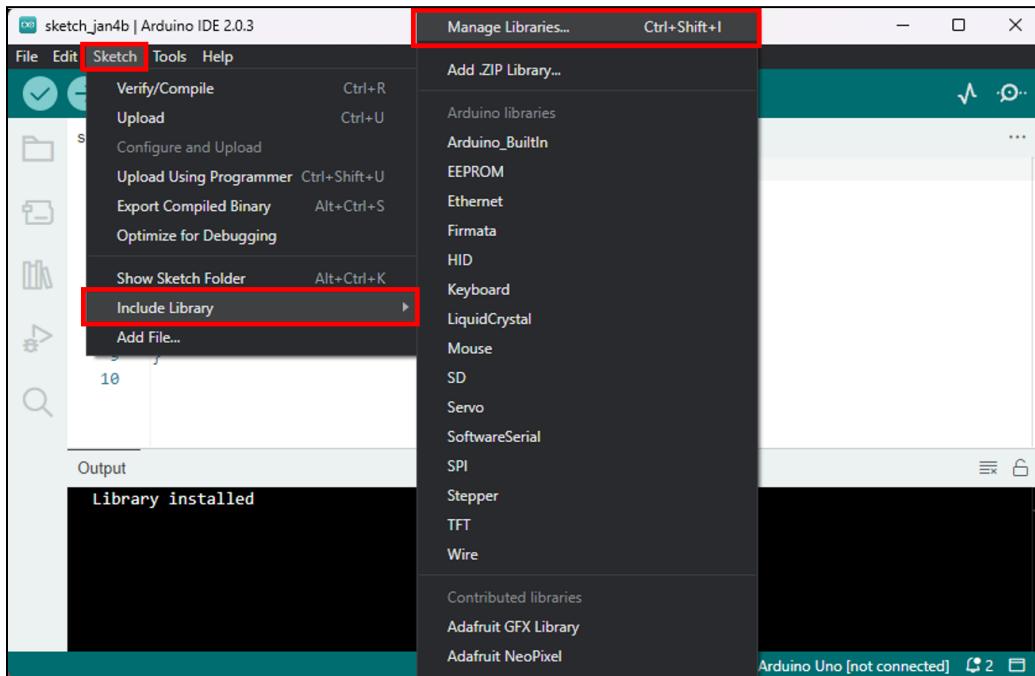
C:\Users\liria\Documents\Arduino\libraries\IoT-Kit-V-One-main\vonesetting.h - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
voneMqttClient.cpp vonesetting.h
1 #if !defined (VONESETTING_H)
2 #define VONESETTING_H
3 #else
4 #error Multiple includes of vonesetting.h
5 #endif
6
7 #define WIFI_SSID "YOUR_SSID"           //Replace this with YOUR WiFi SSID
8 #define WIFI_PASSWORD "YOUR_PASSWORD"  //Replace this with YOUR WiFi Password
9
10 #define MQTT_SERVER "mqtt.v-one.my"
11 #define MQTT_PORT 8883
12 #define MQTT_USERNAME "49DUJmqvBK9IHf2r" //Replace this with the Access Token of YOUR gateway
13 #define MQTT_PASSWORD " "
14
15 #define GATEWAYID "e466f5da-e3a0-4f79-b383-8524e0e654df" //Replace this with the gatewayID of your gateway
16 #define INTERVAL 1000 //1s
17 #define INTERVAL2 500 //0.5s
18

```

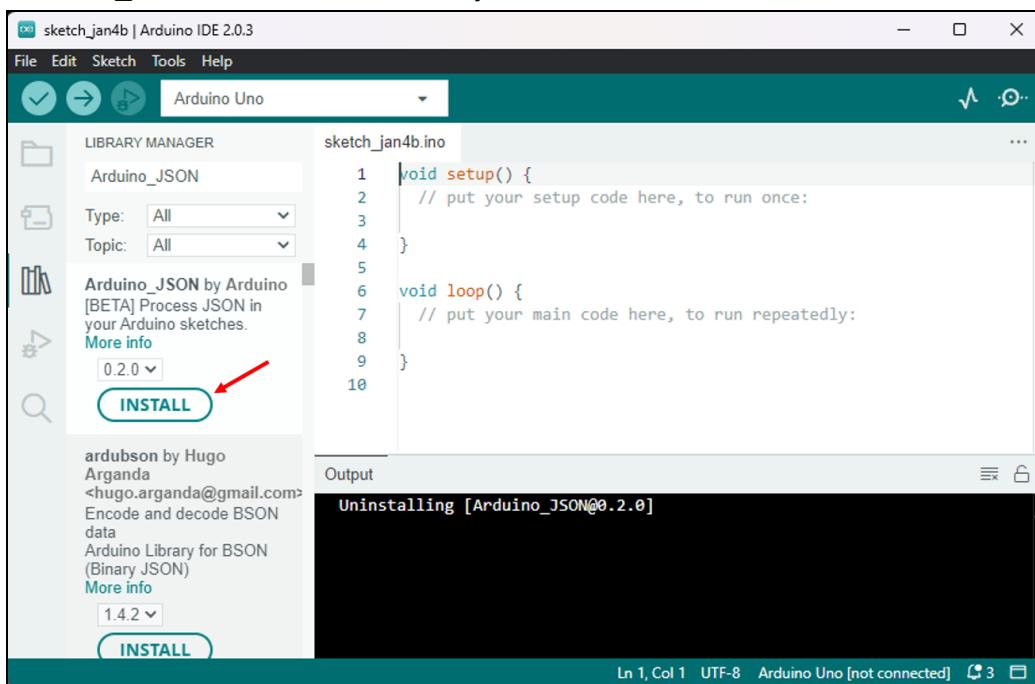
C++ source file length : 622 lines : 18 Ln : 18 Col : 1 Pos : 623 Windows (CR LF) UTF-8 INS

Add Libraries

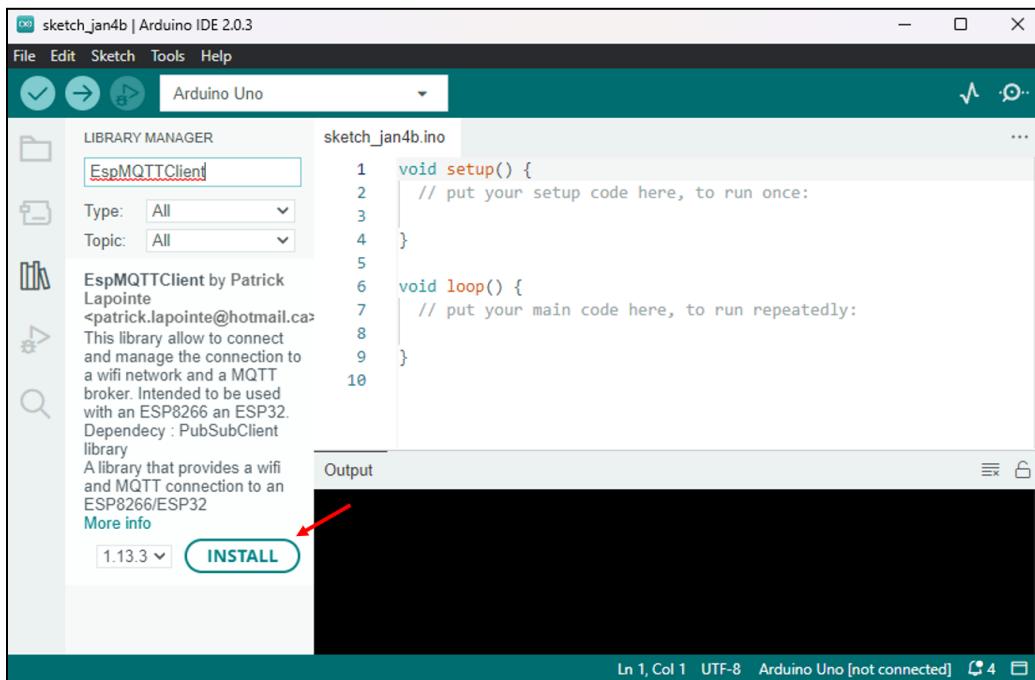
Go to Sketch > Include Library > Manage Libraries.



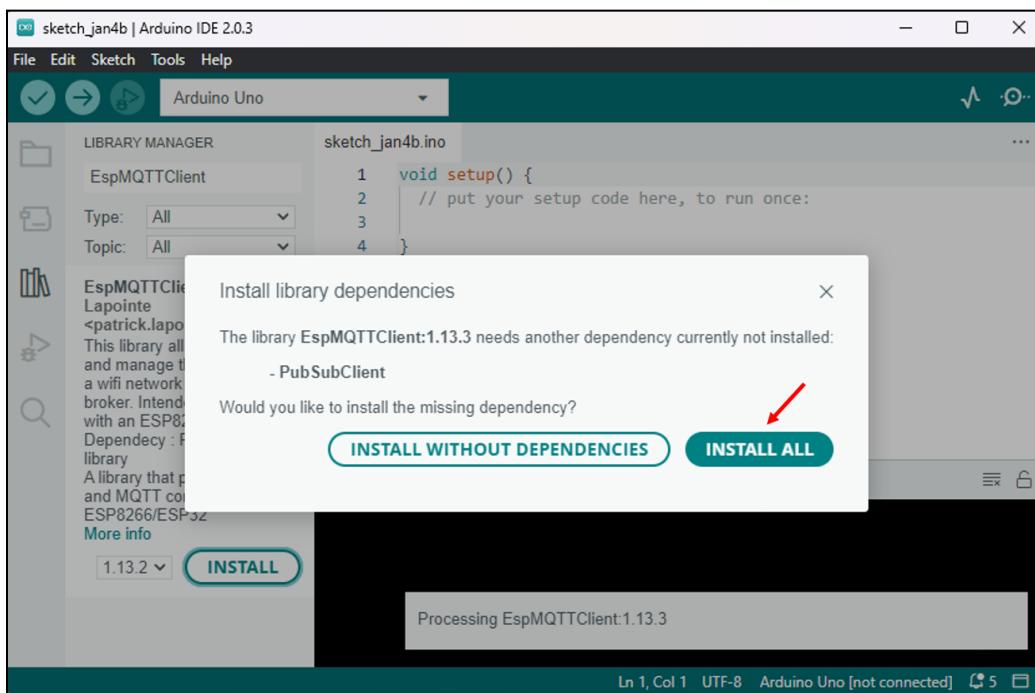
Search **Arduino_JSON** and install the library.



Then, search for **EspMQTTClient**. Install that library.

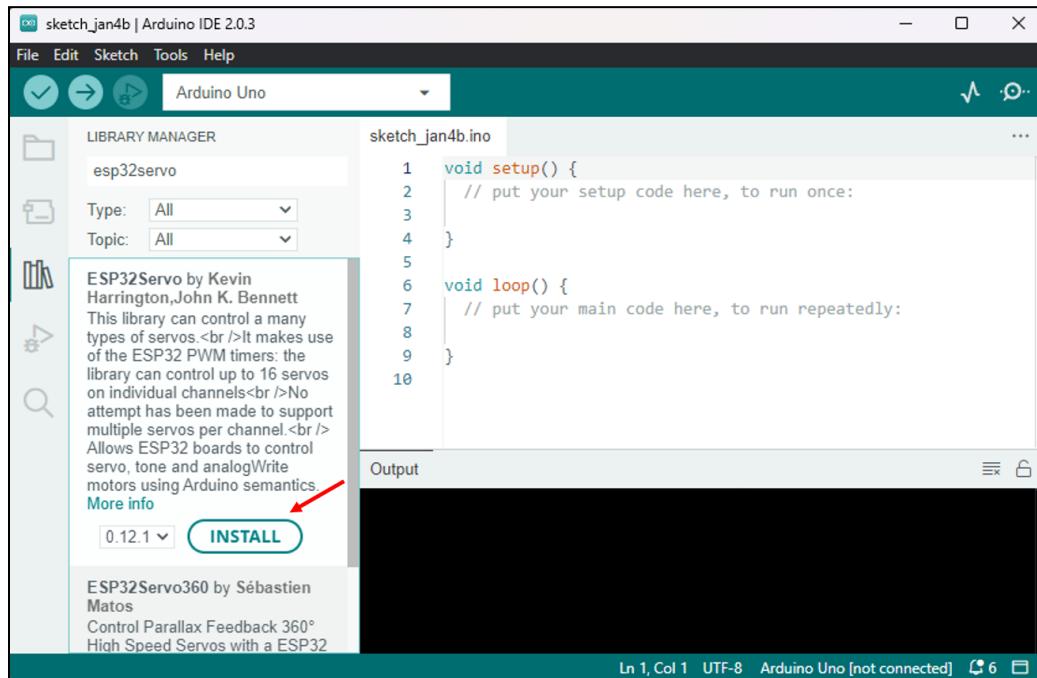


This window will pop up. Click **Install All**.

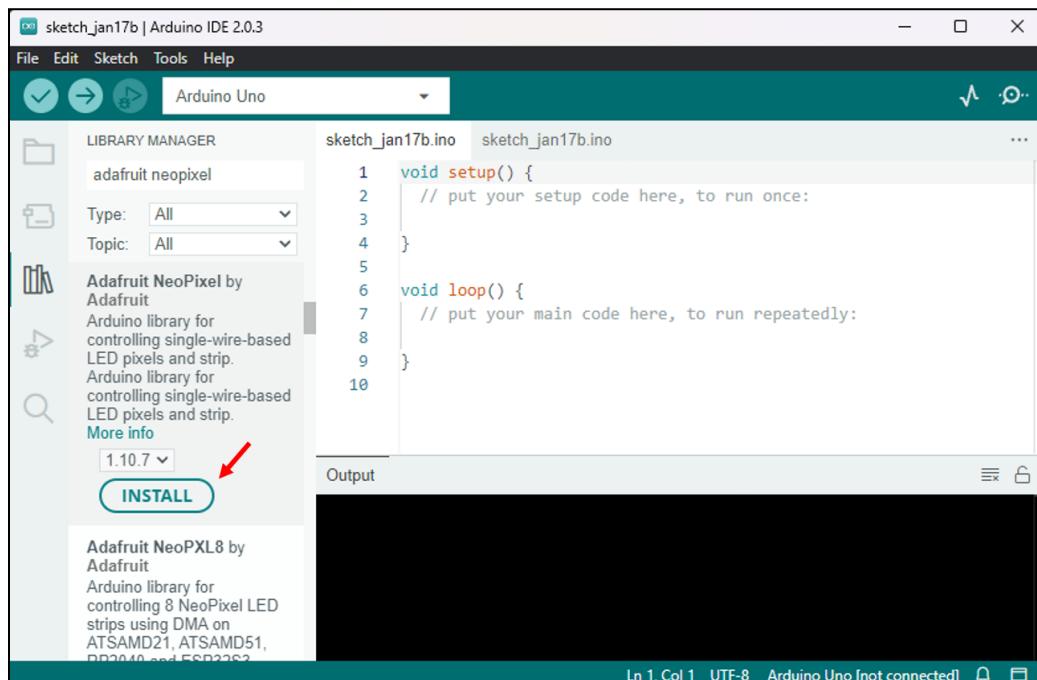


For projects with servo, ensure the servo library is installed.

Note: Please type in “**esp32servo**” to ensure you find the right library. The ESP32 does not support the standard servo library.



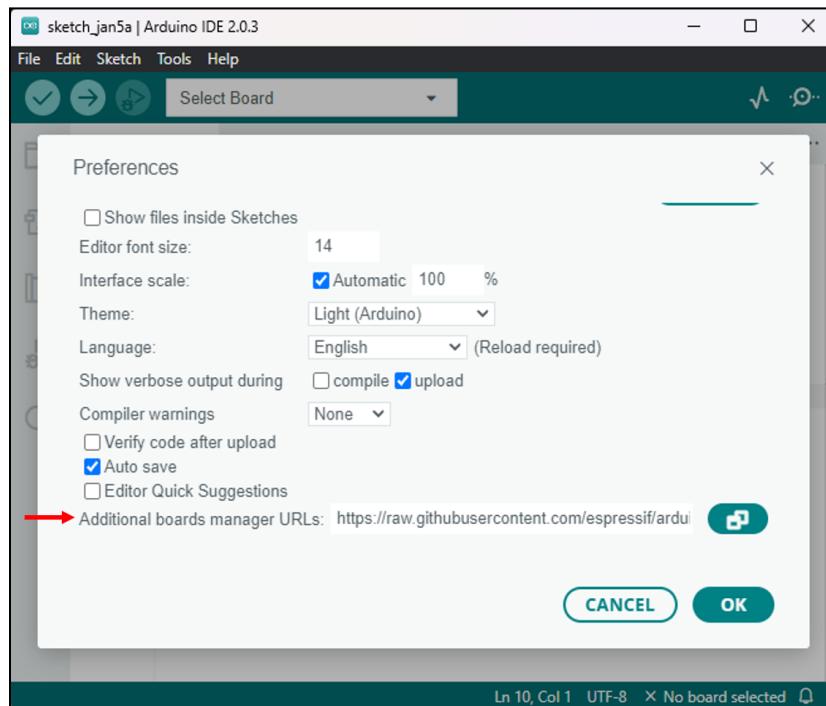
Lastly, this library is a must for Maker Feather AloT S3 users to use the onboard Neopixel. Search for the **Adafruit Neopixel** library.



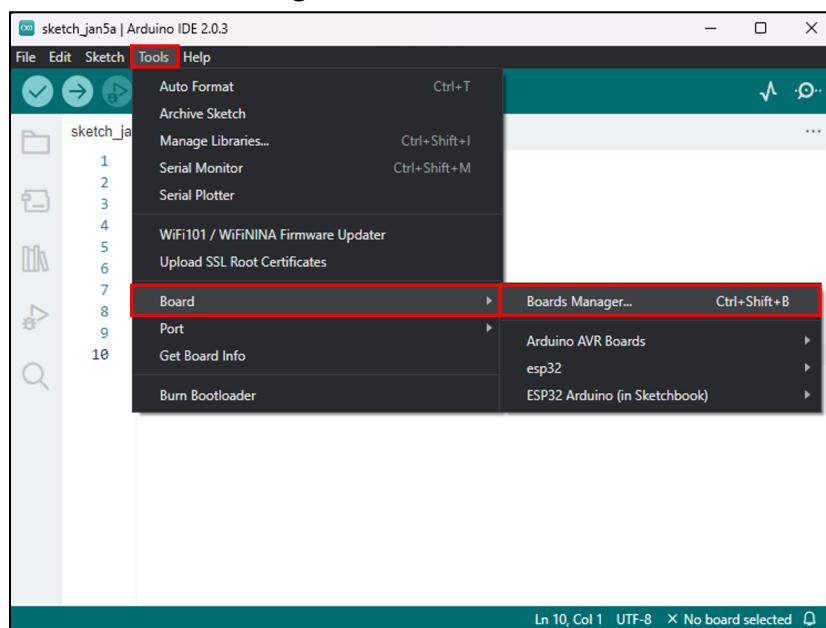
Upload Code

Go back to the Arduino IDE. For this time, we need to include the ESP32 library. Go to **File > Preferences**. Then, copy the link below and paste it into the **Additional Boards Manager URLs** box.

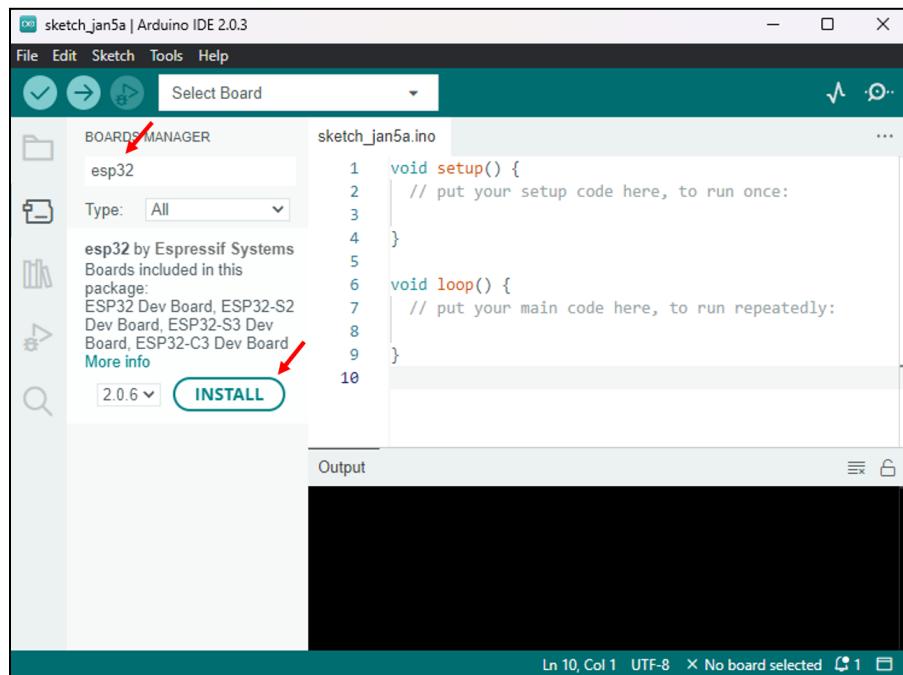
- https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



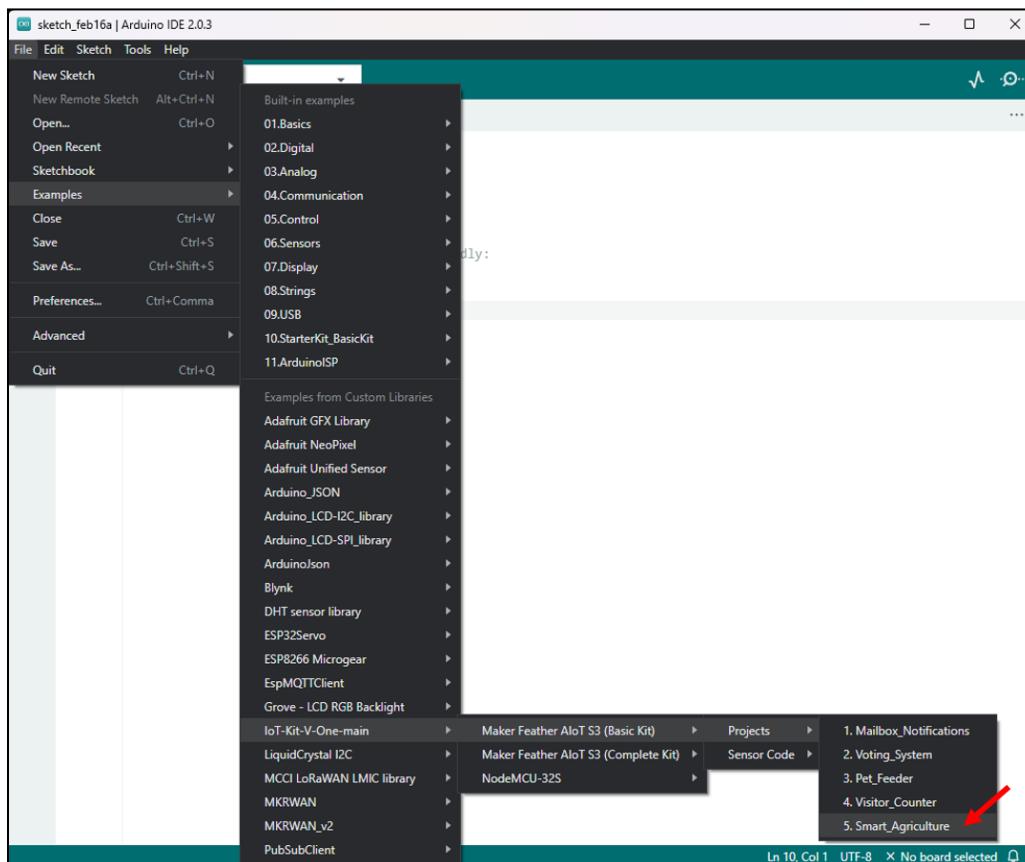
Go to **Tools > Board > Boards Manager**.



Type in “**esp32**” and install the esp32 board manager by Espressif Systems.

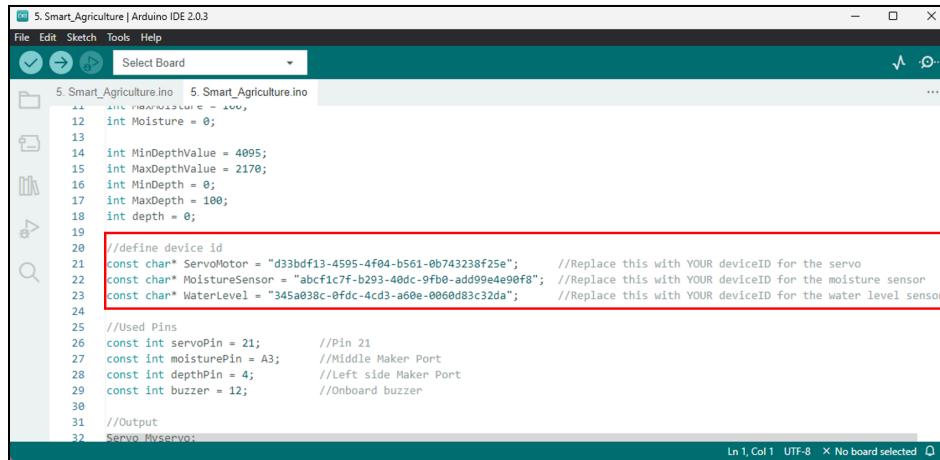


For the project codes, you may directly go to **File > Examples > IoT Kit-V-One-main > Maker Feather AloT S3 (Basic Kit) > Projects** and choose any project.



You will need to edit only one part of the project code. Find the lines that define the **deviceID** (in the red box) and replace them with **your deviceID** for each sensor.

Note: You can get the deviceID from the V-One platform, **Device Manager > Devices > click on the related device**.



```

5. Smart_Agriculture | Arduino IDE 2.0.3
File Edit Sketch Tools Help
Select Board ...
5. Smart_Agriculture.ino 5. Smart_Agriculture.ino
11  #include <Wire.h>
12  int moisture = 200;
13
14  int MinDepthValue = 4095;
15  int MaxDepthValue = 2170;
16  int MinDepth = 0;
17  int MaxDepth = 100;
18  int depth = 0;
19
20 //define device id
21 const char* ServoMotor = "d33bdf13-4f04-b561-0b743238f25e"; //Replace this with YOUR deviceID for the servo
22 const char* MoistureSensor = "abfc1c7f-b293-40dc-9fb0-add99e4e90f8"; //Replace this with YOUR deviceID for the moisture sensor
23 const char* WaterLevel = "345a038c-0fdc-4cd3-a60e-0060d83c32da"; //Replace this with YOUR deviceID for the water level sensor
24
25 //Used Pins
26 const int servoPin = 21; //Pin 21
27 const int moisturePin = A3; //Middle Maker Port
28 const int depthPin = 4; //Left side Maker Port
29 const int buzzer = 12; //Onboard buzzer
30
31 //Output
32 Servo MyServo;

```

Before you start to upload the code, you will need to enter the **ROM Bootloader mode** of the Maker Feather IoT S3. Follow these steps:

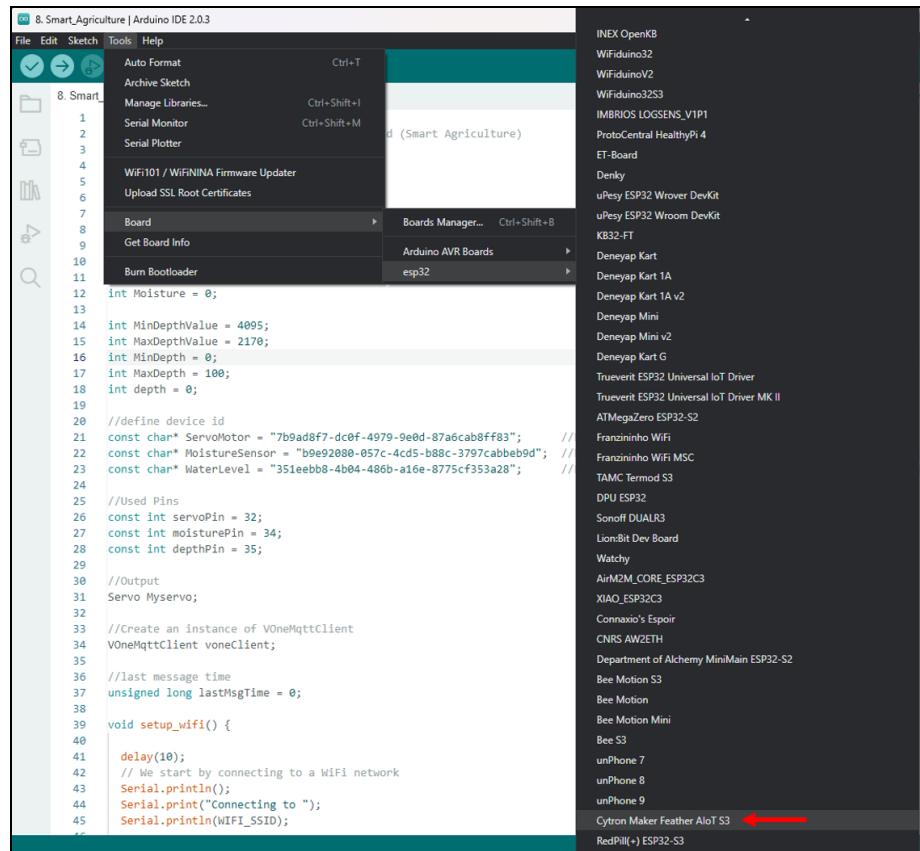
- Connect the Maker Feather IoT S3 to the computer.
- Press and hold the **BOOT** button.
- Press and release the **RESET** button. Make sure the **BOOT** button is still pressed while resetting the board.
- Now you can release the **BOOT** button. You should see a new COM port on your computer.

Note: You need to enter ROM Bootloader mode for **first-time use only**.

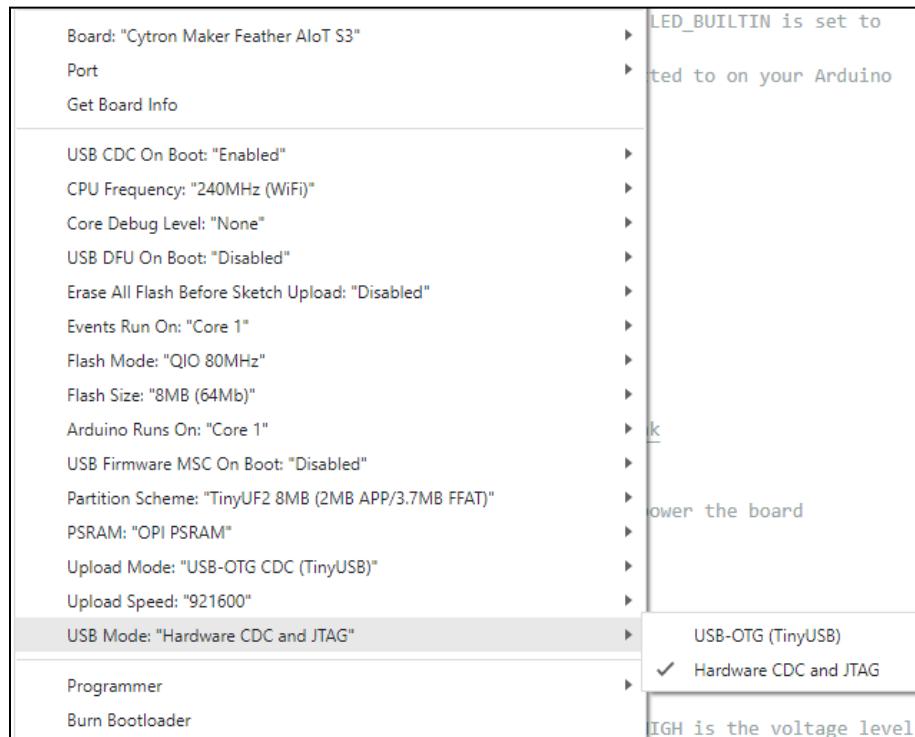
Choose the corresponding **board** and **COM port** before uploading the code.

- **Tools > Boards > esp32 > Cytron Maker Feather IoT S3**
- **Tools > Port > Your COM port**

Note: For the COM port, you will notice it will show a **random name of an ESP32 board** instead of the Cytron Maker Feather IoT S3, it's okay and you may ignore that name and just select that port.



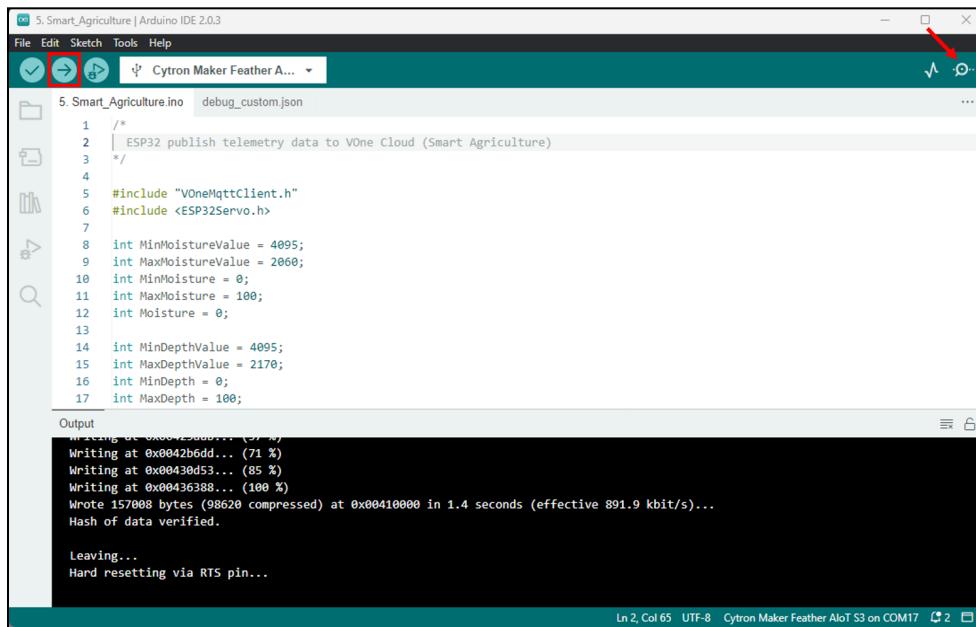
Change the USB Mode to Hardware CDC and JTAG.



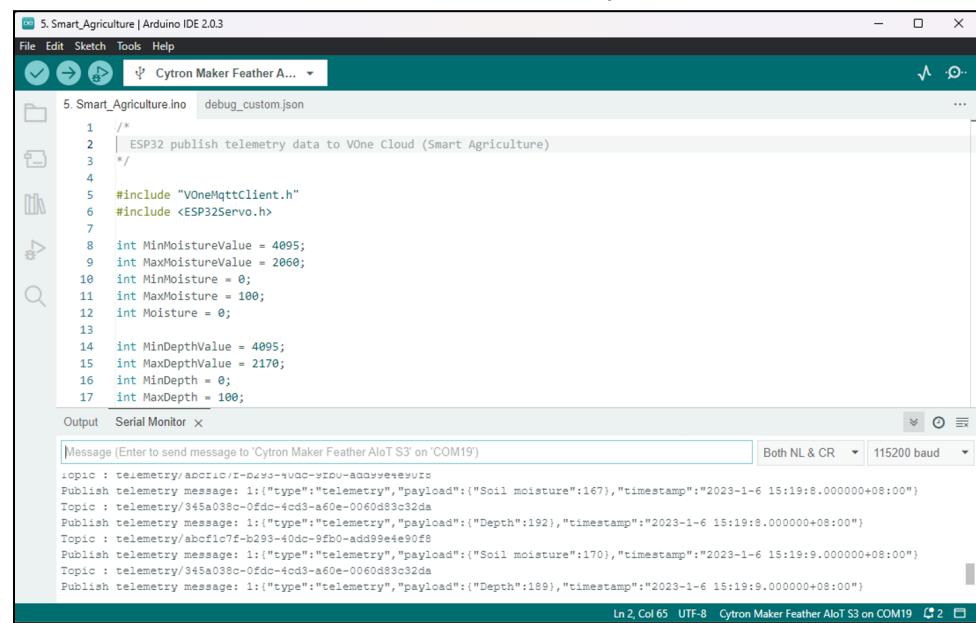
Click the Upload button and it will take some time to finish uploading. Once the code has been successfully uploaded, press the **RESET** button on your Maker Feather AloT S3 board and your Maker Feather AloT S3 will start to run the code.

Note: Choose the **COM port** again after you pressed the **RESET** button (**for first-time use only**).

Then, click the **Serial Monitor** button on the top right of the screen.



Here you can see whether the sensor data is successfully sent to the V-One platform.



Part 3

IoT Projects

In Part 3 of this manual guide, you will be directed to experiment with a variety of sample IoT projects using the setup outlined in Part 1 and Part 2. We recommend following the project sequence, as it progresses from **simple to more complex** projects.

Project 1: Mailbox Notifications

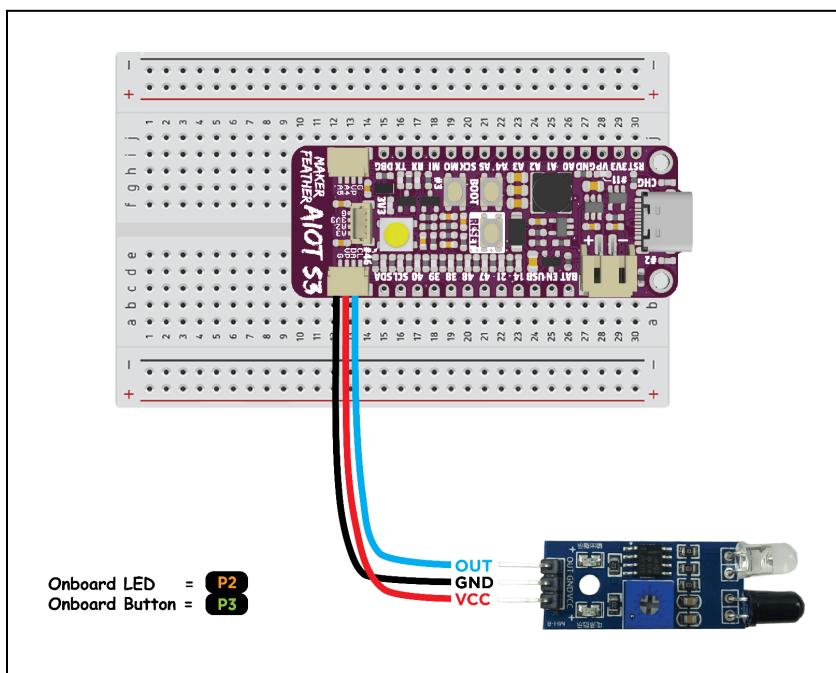
Overview

Do you often forget to check your mailbox and miss important mail? After a long day, it's easy to overlook your mailbox and miss out on important letters and packages. But with this simple IoT project, you'll never miss another piece of mail again. An infrared sensor will detect when mail has been inserted into your mailbox and count how many items are inside. You'll receive real-time notifications via V-ONE Cloud, so you'll always know when you have mail waiting for you. Once you've collected your mail, simply push the button to reset the count and start fresh. No more missed deliveries or forgotten mail - this project has got you covered.

Required Components

- Maker Feather AIoT S3
- Breadboard
- Infrared Sensor
- Onboard LED
- Onboard Button
- Qwiic Cable

Circuit Diagram



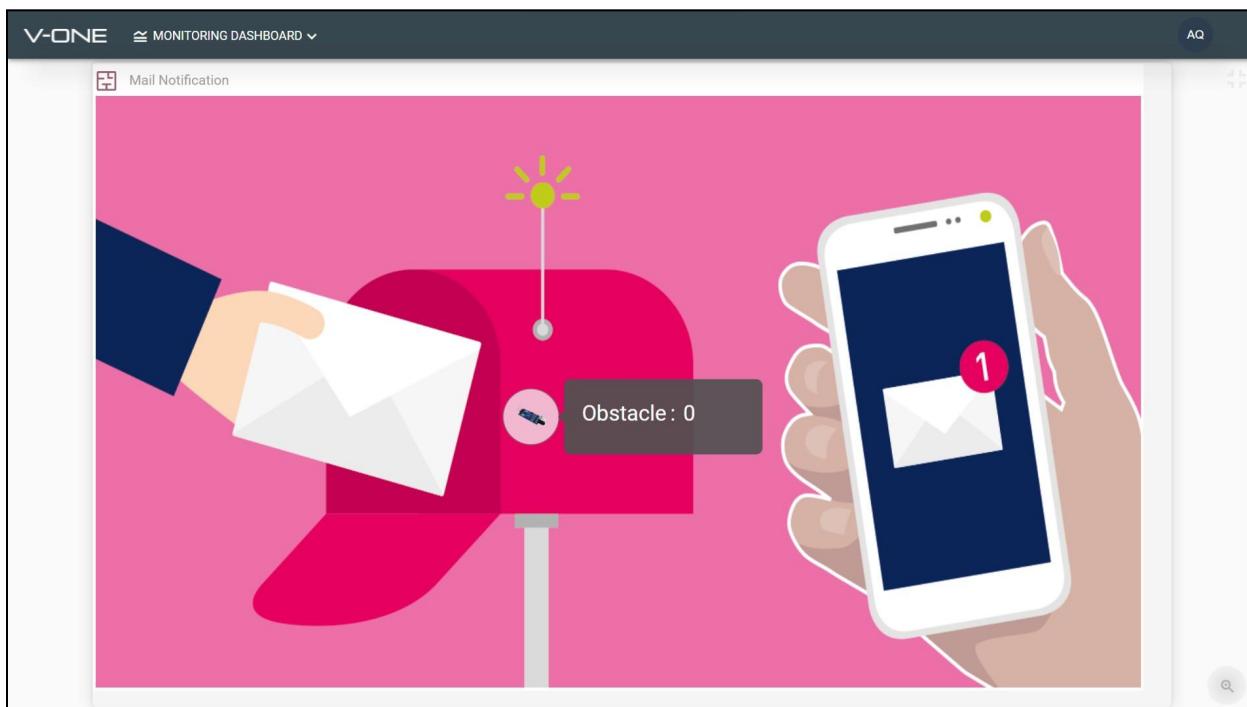
Project Code

- [Mailbox_Notifications.ino](#)

Workflow Code

- [For Infrared Sensor](#)

V-One Dashboard



Tips and Tricks

- Place your infrared sensor properly. If the mail moves too fast, it can't detect it, and if the mail stuck on the sensor, the counting will keep increasing.
- You can adjust the variable resistor on the infrared sensor with a screwdriver for better sensitivity.
- From the **workflow** template given, the default **threshold value** is **0**. It means that if the mail count is more than 0, it will send email notifications to the user. You can modify or change the number of mails that will trigger the email notifications in the workflow.

```

3  obj = parameter[1]["data"]
4  arrlgt = parameter[1]["count"]["total"] - 1
5  button = obj[arrlgt]["Button1"]
6  threshold = 0 ←

```

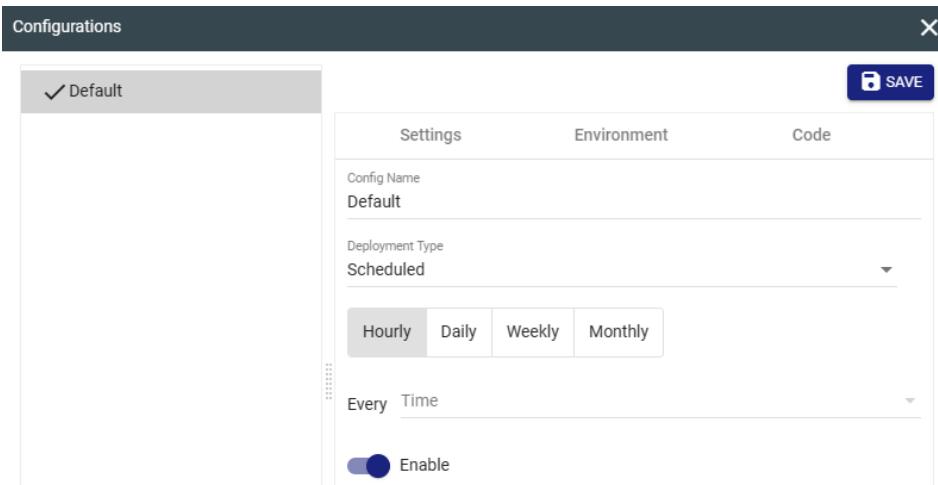
- You can change the message in the email notifications by modifying the Python code in Workflow.

```

8  if int(people) > threshold:
9      msgbody=<p>There is a mail/parcel in your mailbox.</p><br>
10     output[1]=[Notifications] Mail/Parcel Received"
11     output[2]=msgbody
12     output[3]=2

```

- Change the **Workflow deployment type** from ‘Immediately Once’ to ‘Scheduled’. Now, you can choose to receive email notifications hourly, daily, weekly, or monthly.



Project 2: Voting System

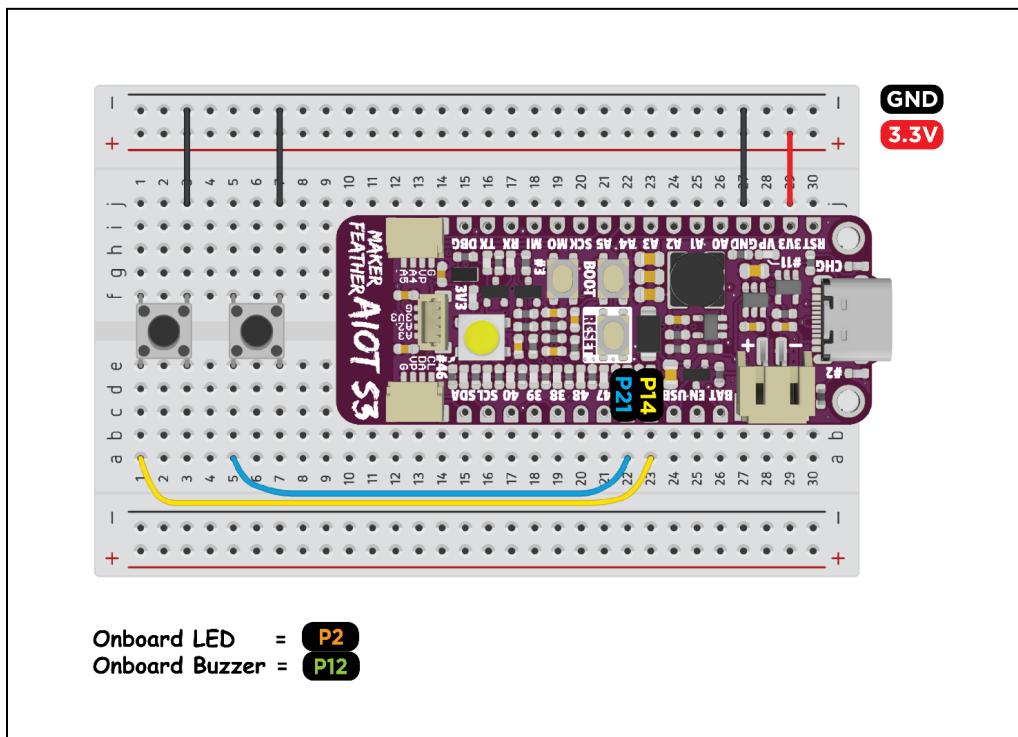
Overview

Do you find the traditional pen-and-paper voting process slow and error-prone? This IoT project offers a solution. Using an embedded system, voters can cast their votes simply by pressing a button. The LED will blink to confirm the vote, providing a simple and secure method of voting. But that's not all - this system can also automatically calculate the voting results and transfer the data to the admin via V-ONE Cloud. With this project, there's no need to worry about spoilt votes or human error, ensuring a fair and transparent voting process.

Required Component

- Maker Feather AIoT S3
- Breadboard
- 2 x Push Button
- Onboard LED
- Onboard Buzzer
- Jumpers

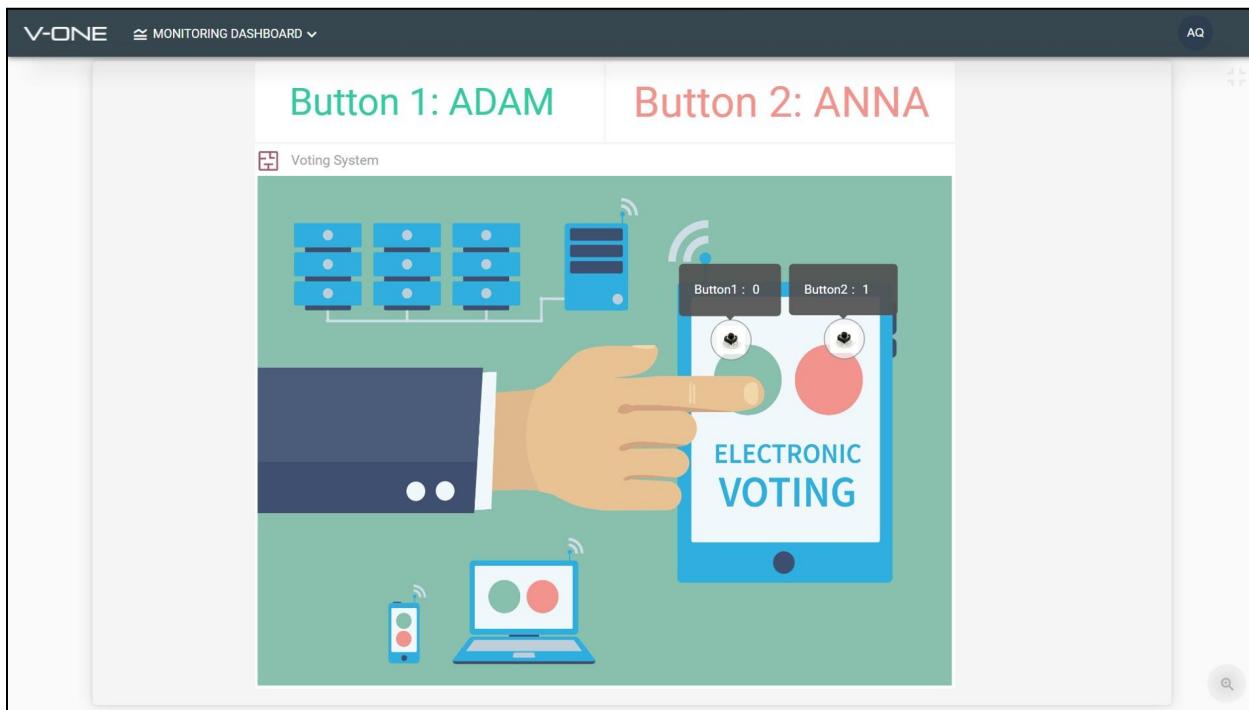
Circuit Diagram



Project Code

- [Voting_System.ino](#)

V-One Dashboard



Tips and Tricks

- When the push button is pressed, the onboard buzzer will make a sound and the onboard LED will light up indicate that the vote is received in the system.
- You can use this project for any voting activity. Have fun changing the candidate's name. You can also add more candidates. Just add more buttons to it. Please make sure to update your devices in Device Manager and the deviceID in Arduino IDE.

Project 3: Pet Feeder

Overview

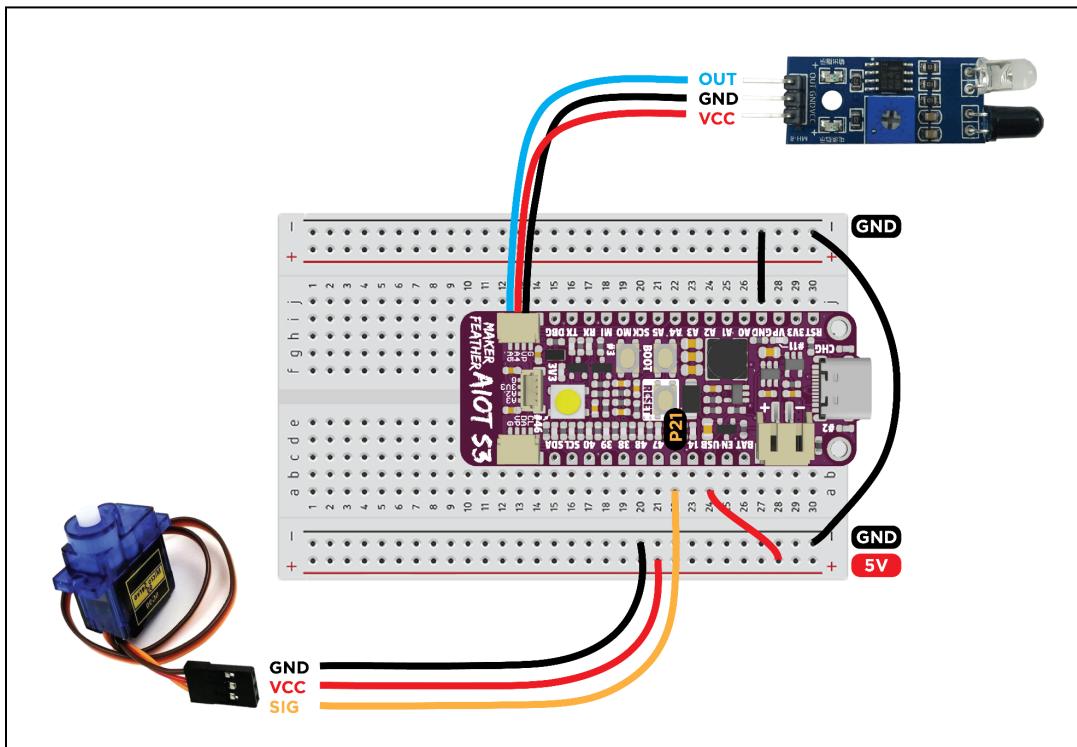
As a pet owner, it's important to ensure that your furry friend is fed and taken care of, even when you're away. With this IoT project, you can feed your pet remotely, no matter where you are. An infrared sensor will detect when your pet is near the food container, sending you an instant notification that it's time for a meal.

Using the V-ONE dashboard, you can open the food container remotely and serve your pet's meal. And with the help of a servo, you can easily switch to another compartment of the container for the next meal. This project offers a simple and convenient way to ensure that your pet is always well-fed and cared for, no matter where you are.

Required Components

- Maker Feather AloT S3
- Breadboard
- Infrared Sensor
- Micro Servo
- 2 x Qwiic Cable

Circuit Diagram



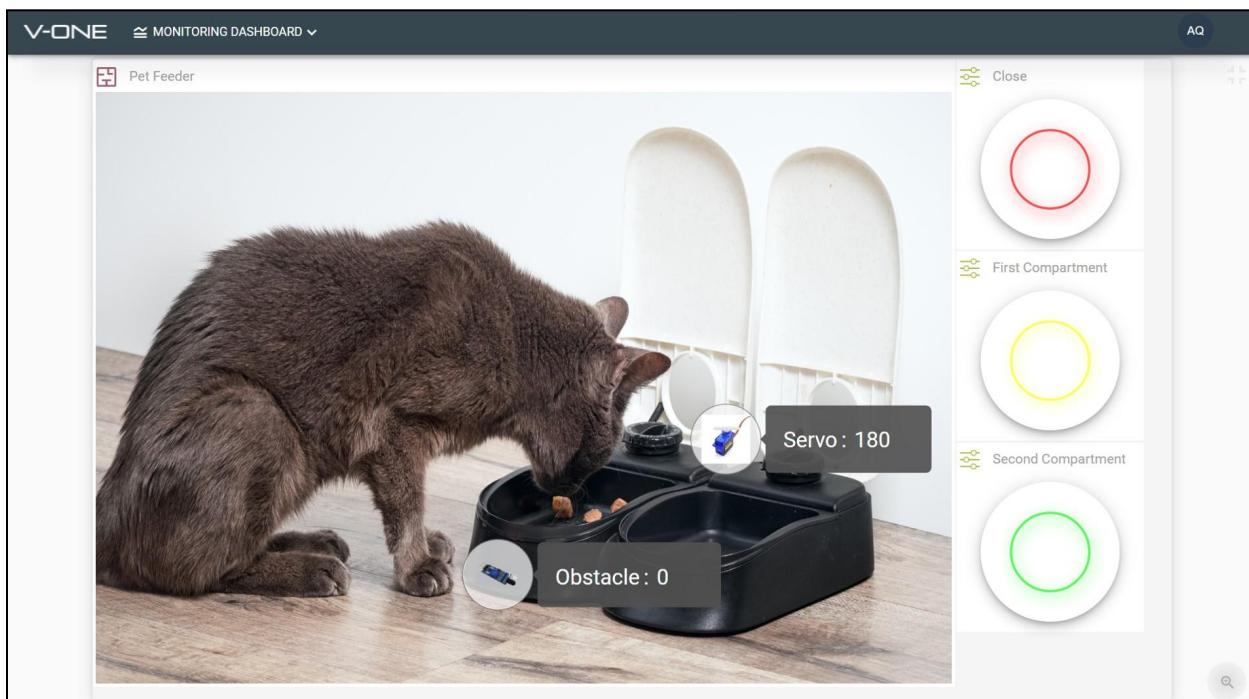
Project Code

- [Pet Feeder.ino](#)

Workflow Code

- [For Infrared Sensor](#)

V-One Dashboard



Tips and Tricks

- You can make multiple compartments for the pet feeder. Then, adjust the degree setting of the button in the Dashboard to match each compartment. For example:
 - Button 1 (Close): 0 or 180 degrees
 - Button 2 (First compartment): 60 degrees
 - Button 3 (Second compartment): 120 degrees
- If you have a 3D printer, try to print the pet feeder container using this [3D design](#).

Project 4: Visitor Counter

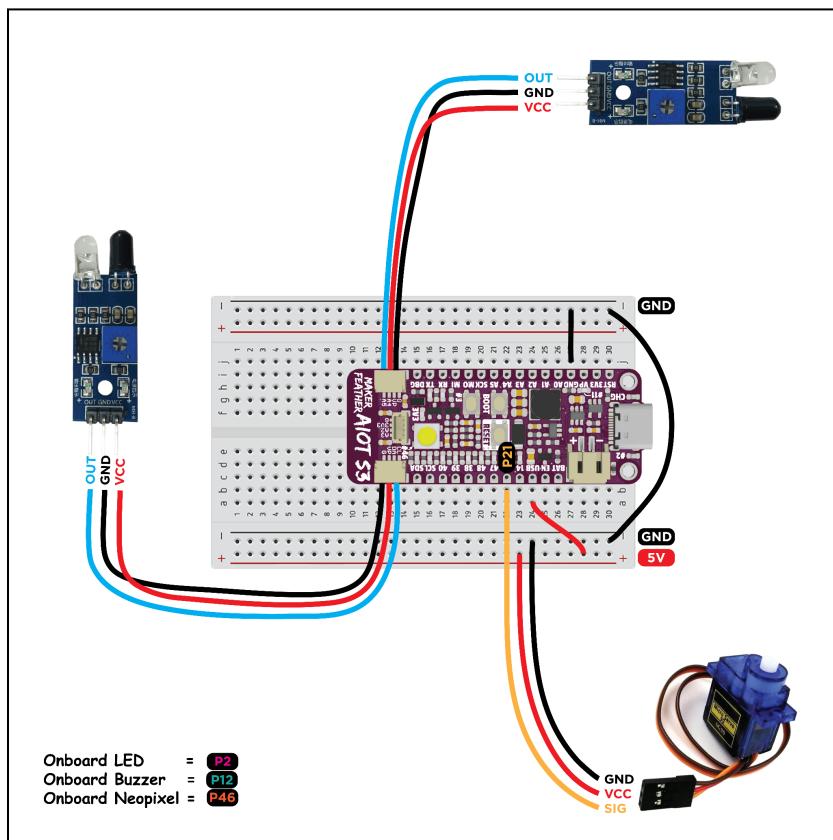
Overview

Do you run a store and want to keep track of the number of visitors? Or do you need to limit the number of people in your store at any given time? This IoT project has got you covered. With two infrared sensors placed at the entrance and exit of your store, the system can detect when someone enters or exits, providing you with real-time visitor count data. With the added feature of a servo, you can even choose to automatically close the door if the number of visitors exceeds a set limit. This project provides an easy and effective way to manage traffic in your store and ensure the safety of both your customers and employees.

Required Components

- Maker Feather ALoT S3
- Breadboard
- 2 x Infrared Sensor
- Micro Servo
- Onboard LED
- Onboard Buzzer
- Onboard Neopixel
- 2 x Qwiic Cable

Circuit Diagram



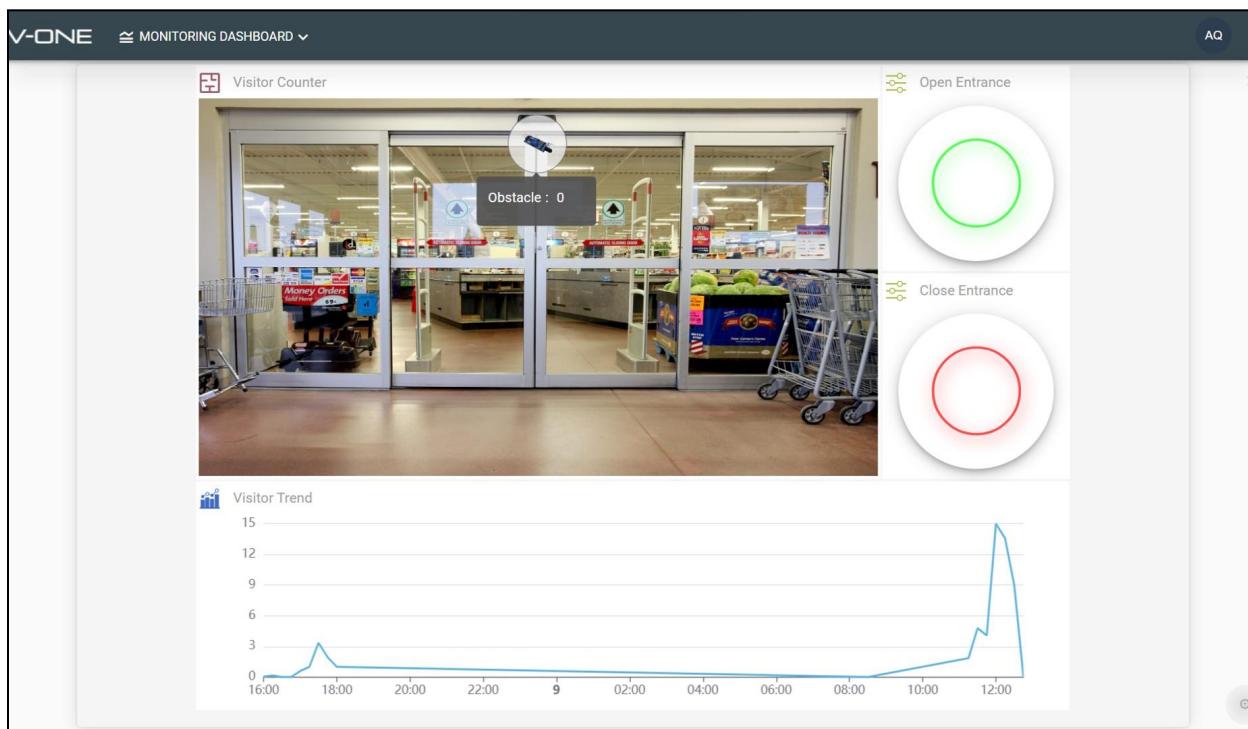
Project Code

- [Visitor_Counter.ino](#)

Workflow Code

- [For Infrared Sensor](#)

V-One Dashboard



Tips and Tricks

- Any movement going through the infrared sensors will produce a “beep” sound. The first infrared sensor increases the value while another one decreases the value. The last output will be the remaining visitor in the store.
- The onboard Neopixel will always light up indicates that the store is open for customers. Once it reaches the threshold value, the Neopixel will turn off and the onboard LED will light up. Then, the owner will receive email notifications.
- You can try to change the threshold value in the Arduino IDE code. The default threshold value is 10. Don’t forget to change it in the workflow also.

```

23 unsigned long lastMillis = 0;
24 volatile int count1 = 0;
25 int limit = 10; ←
26 int count_prev = 0;

```

Project 5: Smart Agriculture

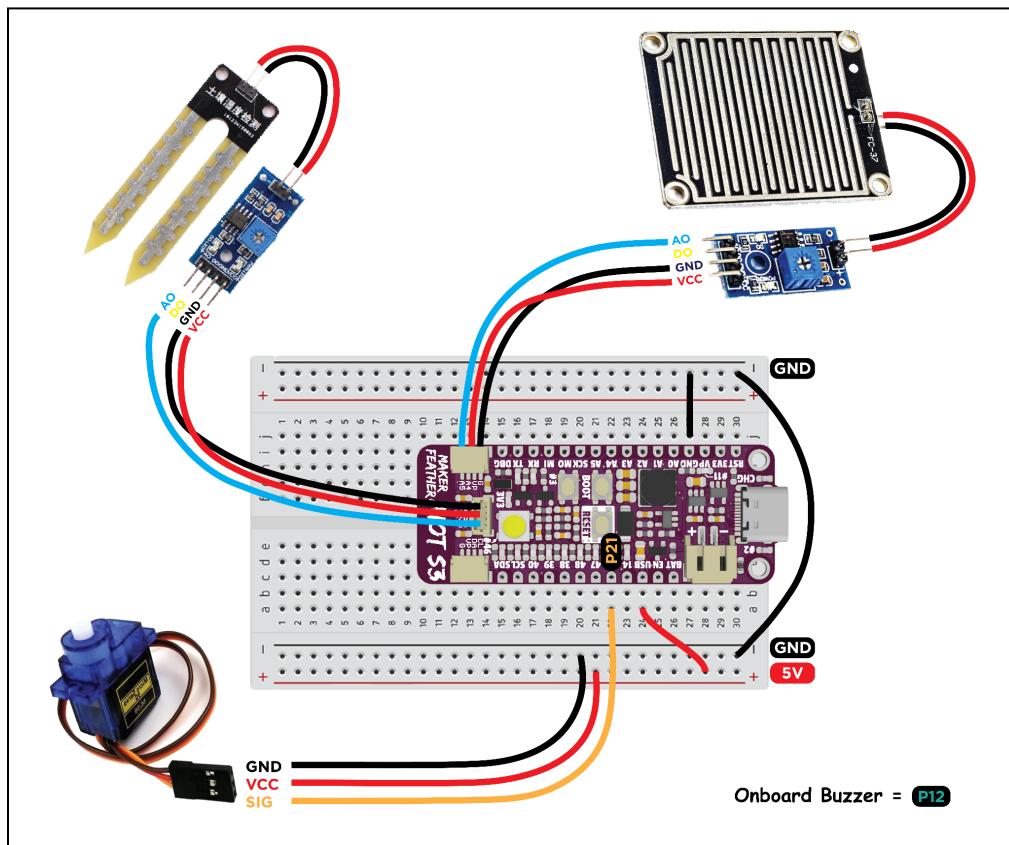
Overview

With technology nowadays, we can take care of our plants even if we are not at the farm. We just need to take action based on the collected data and go to the farm if necessary only. Examples of data that can be collected in this Smart Agriculture project are soil moisture, and water level in a tank. It is called “Smart” because you can take action remotely. For instance, you can control the water flow with a servo. If the soil moisture sensor shows that the soil is dry, we can release the water to the farm. We also will be able to monitor the level of water remaining in the tank.

Required Components

- Maker Feather AloT S3
- Breadboard
- Moisture Sensor
- Water Level Sensor
- Micro Servo
- Onboard Buzzer
- 2 x Qwiic Cable
- Jumpers

Circuit Diagram



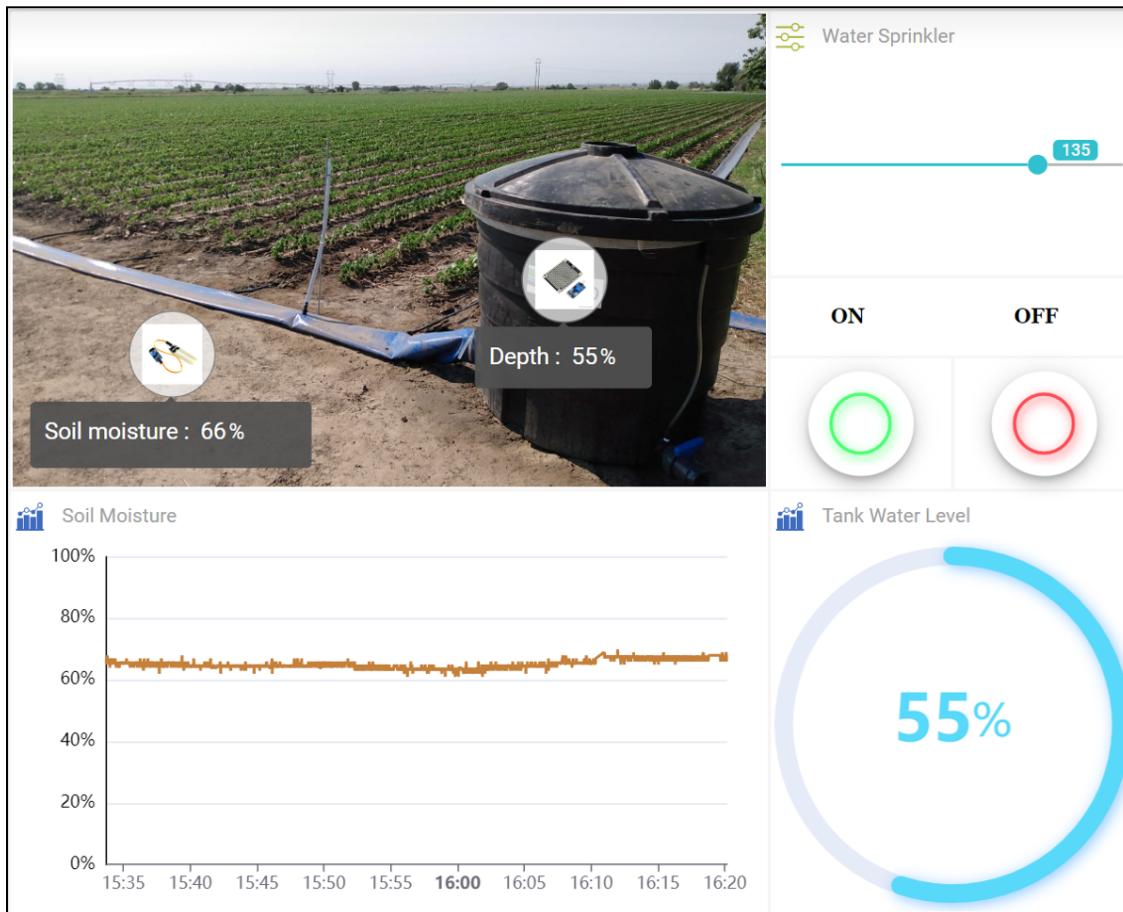
Project Code

- [Smart Agriculture.ino](#)

Workflow Code

- [For Moisture Sensor](#)
- [For Water Level Sensor](#)

V-One Dashboard



Tips and Tricks

- “Water Sprinkler” in the dashboard is represented by the servo to control the water flow. You can try changing to another component like a [water pump](#) to make your project more interesting.
- In this project, you can change the **threshold value** of soil moisture and water level for the email notifications in the workflow.

```

obj = parameter[1]["data"]
arrlgt = parameter[1]["count"]["total"] - 1
moisture = obj[arrlgt]["Soil moisture"]
threshold = 30
    
```

- If you realize that the sensors value are not very accurate, you can modify the value of the following lines in the Arduino code.
 - **MaxMoistureValue** for the moisture sensor
 - **MaxDepthValue** for the water level sensor

```
8 int MinMoistureValue = 4095;
9 int MaxMoistureValue = 2060;
10 int MinMoisture = 0;
11 int MaxMoisture = 100;
12 int Moisture = 0;
13
14 int MinDepthValue = 4095;
15 int MaxDepthValue = 2170;
16 int MinDepth = 0;
17 int MaxDepth = 100;
18 int depth = 0;
```

The **maximum** output value should be 100 (%).

Feedbacks & Questions

If there are any questions / corrections / suggestions / improvements, please email support@cytron.io or you can call **04-548 0668** for technical assistance. Thank you.