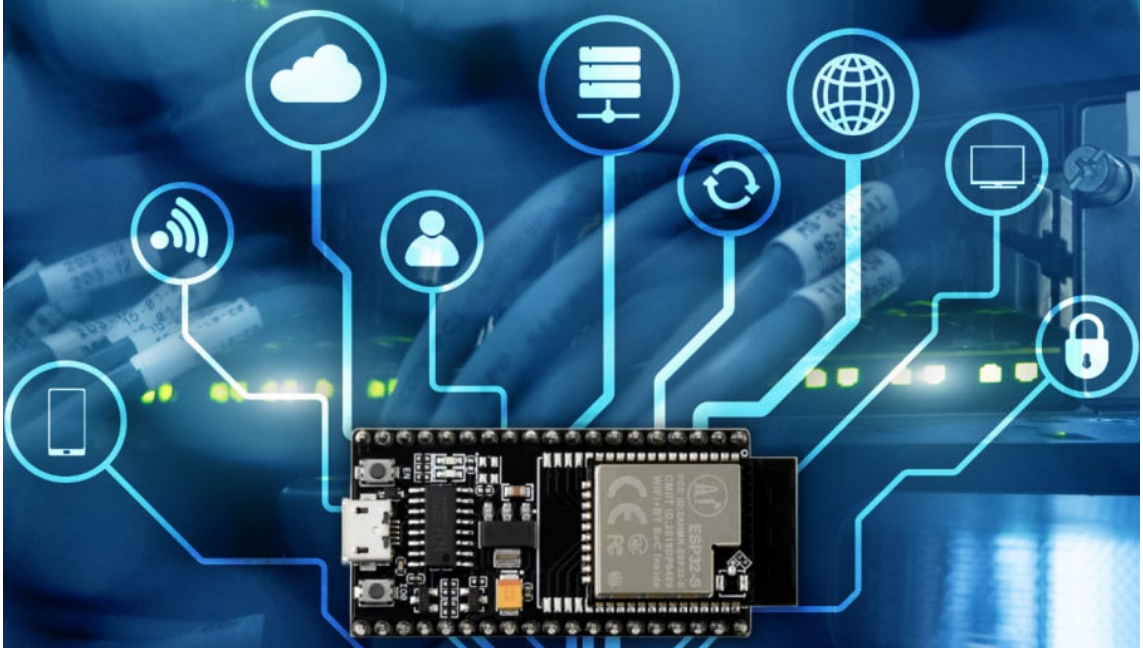
MANUAL GUIDE

IOT KIT



NODEMCU ESP-32S

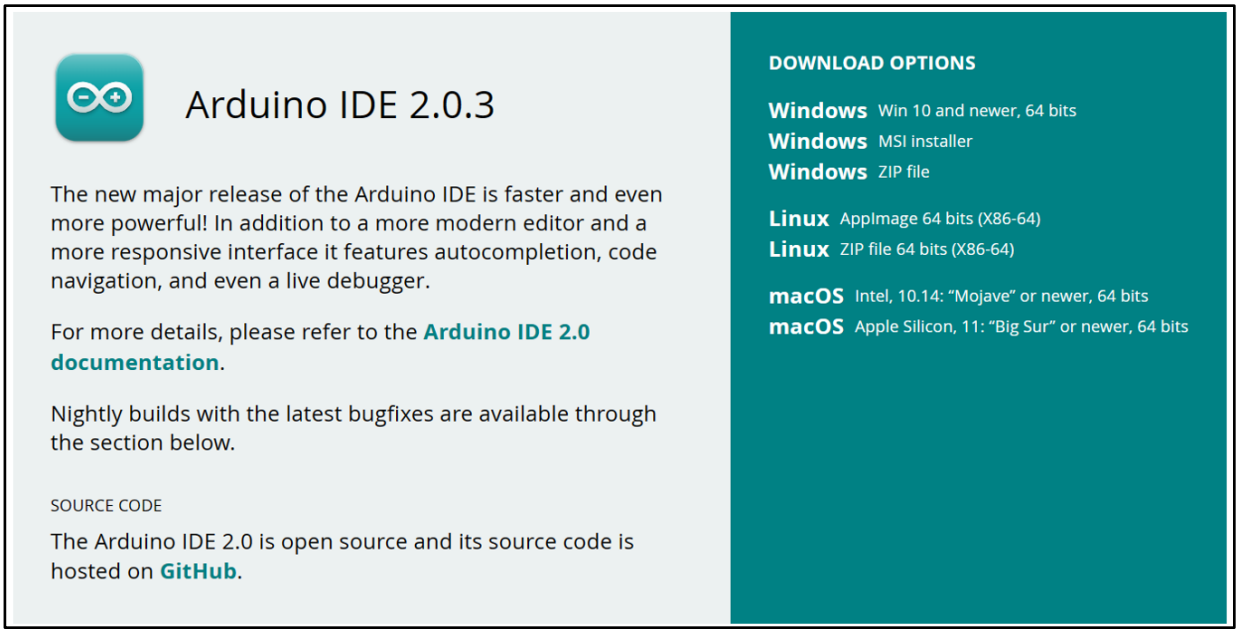
ViTrox Cytron Technologies V-ONE®

Part 2

Arduino IDE

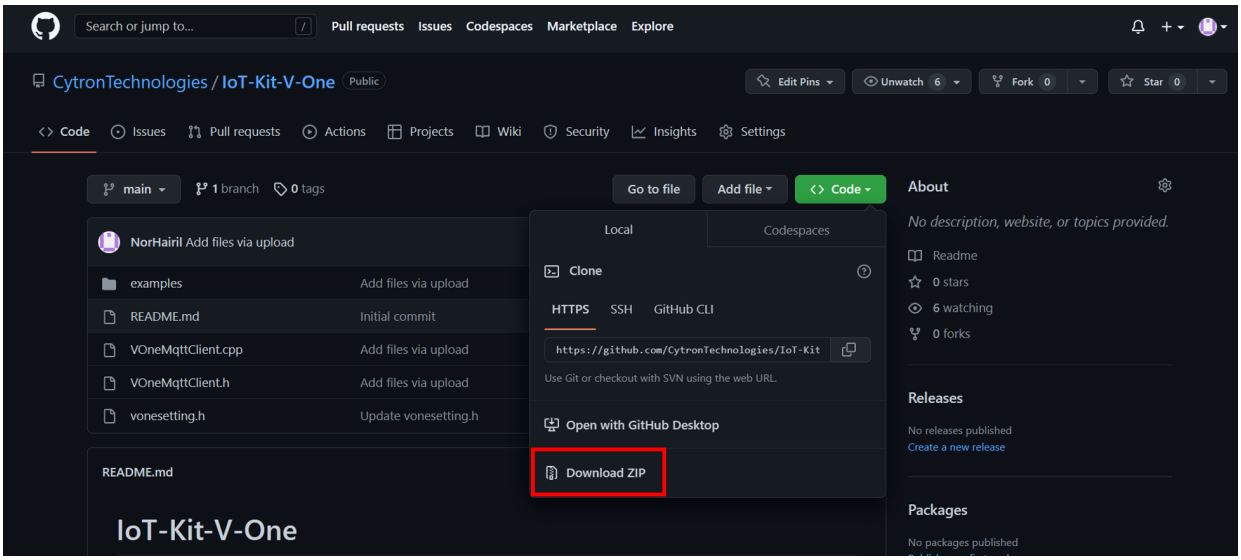
Github Page

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)**.



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

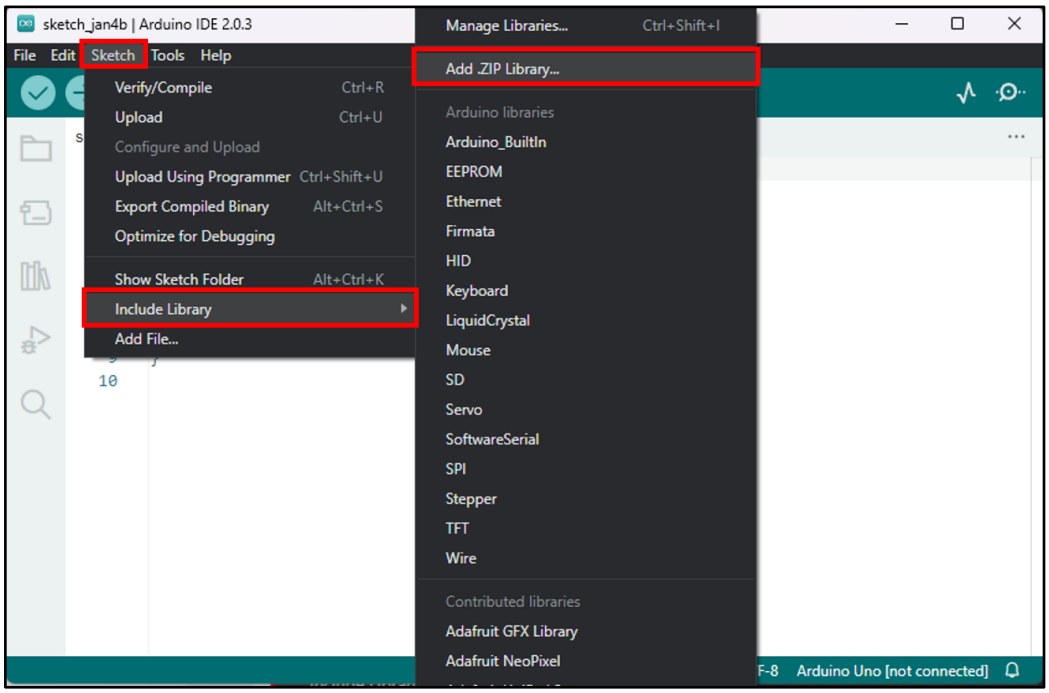
- GitHub - CytronTechnologies/IoT-Kit-V-One



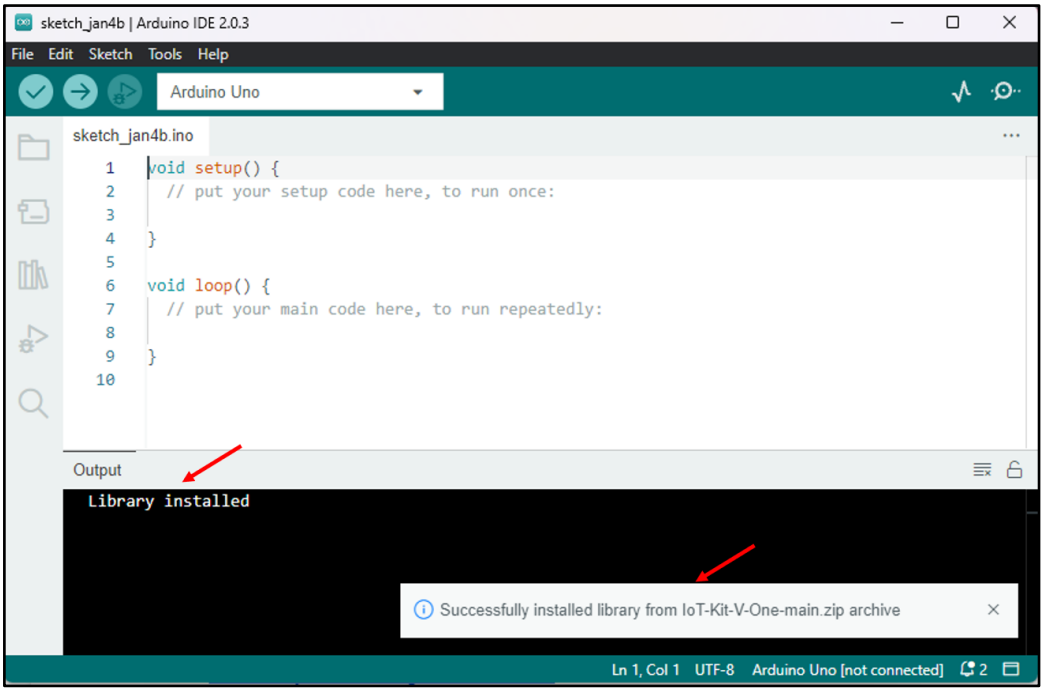
All resources for the IoT Kit are included in this ZIP folder including **V-One libraries**, **sensor code**, **10 IoT projects code**, and **workflow code** (used for email notifications).

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.



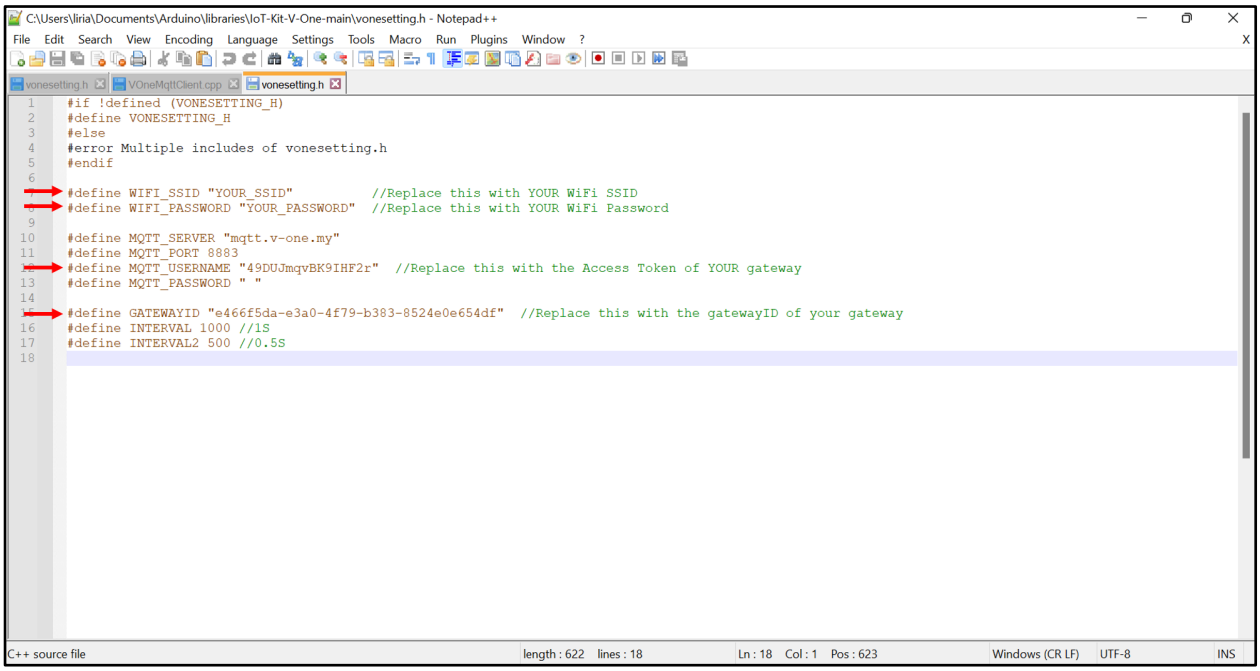
Navigate to the V-One libraries to modify the **WiFi and gateway credentials**. The libraries are usually stored in this path:

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**.

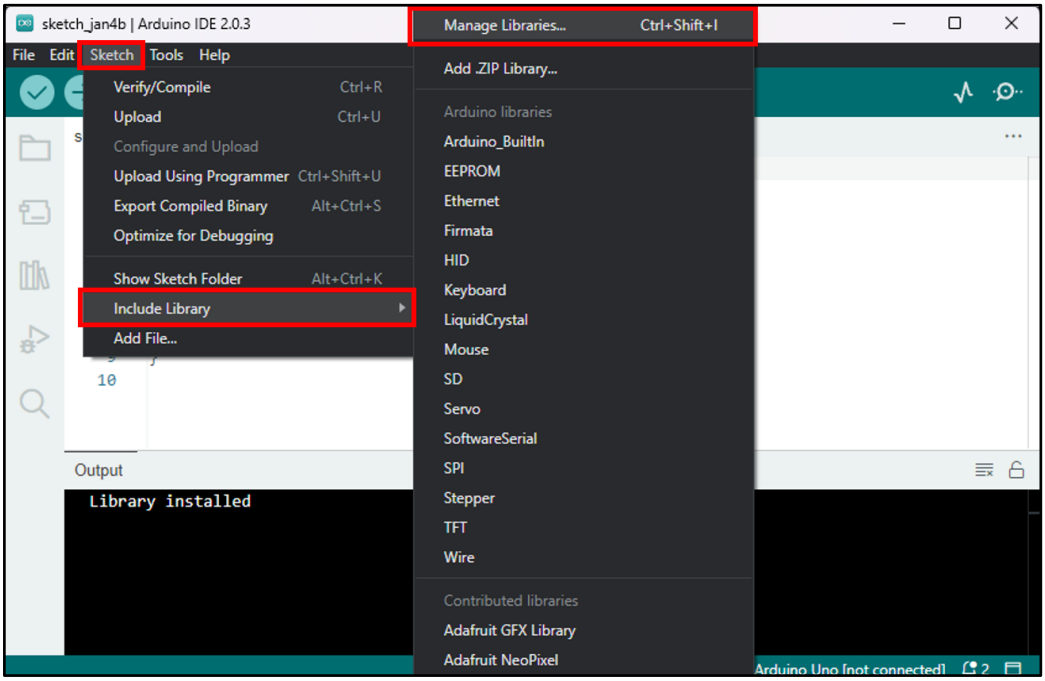
Note: Access Token and GatewayID can be found in the V-One platform at **Device Manager > Gateways**.

Then, click the save button and close the file.

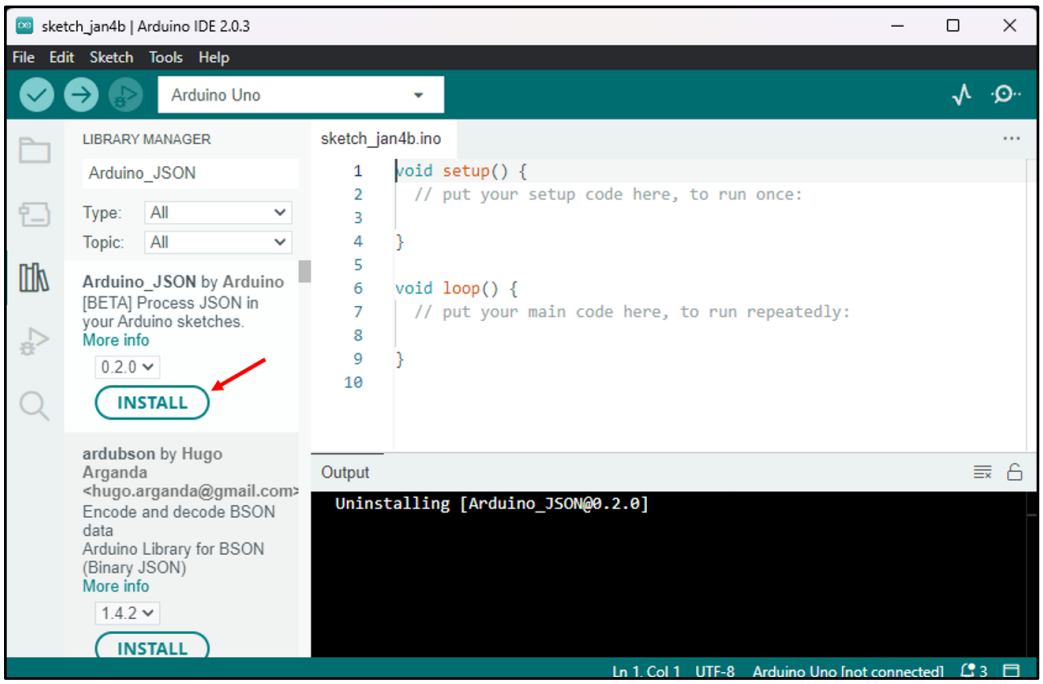


Add Libraries

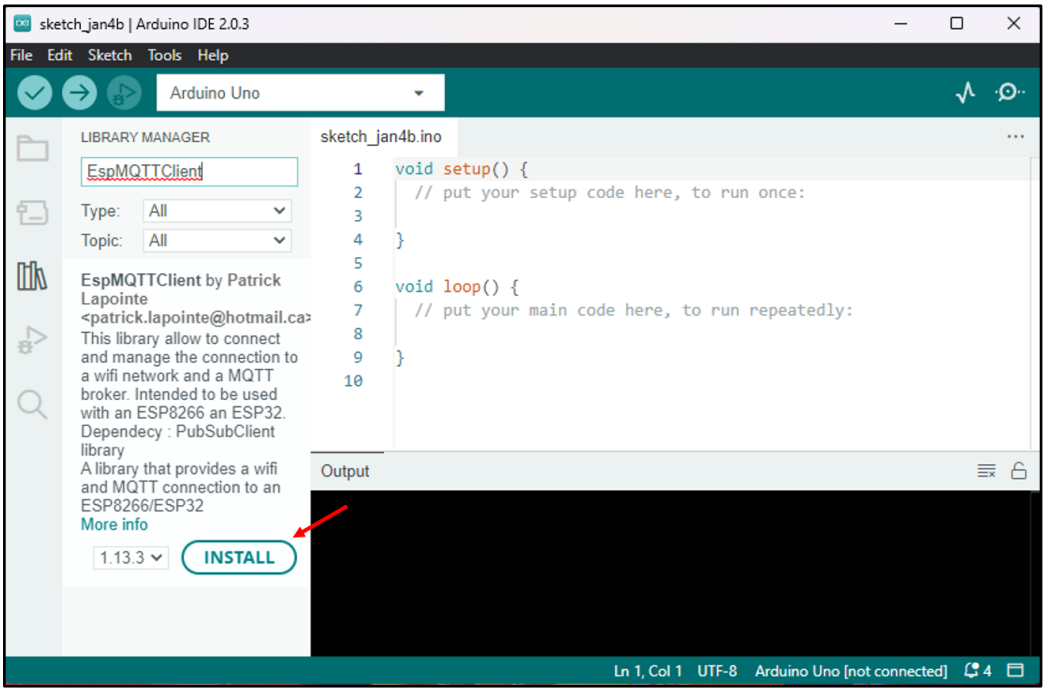
Go to **Sketch** > **Include Library** > **Manage Libraries**.



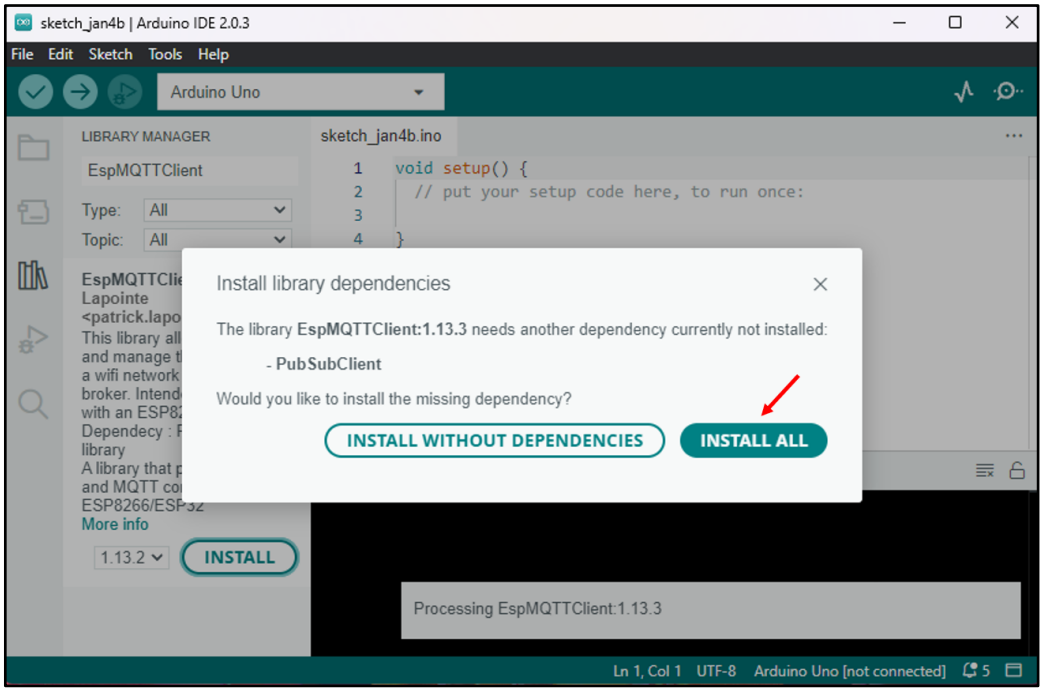
Search **Arduino_JSON** and install the library.



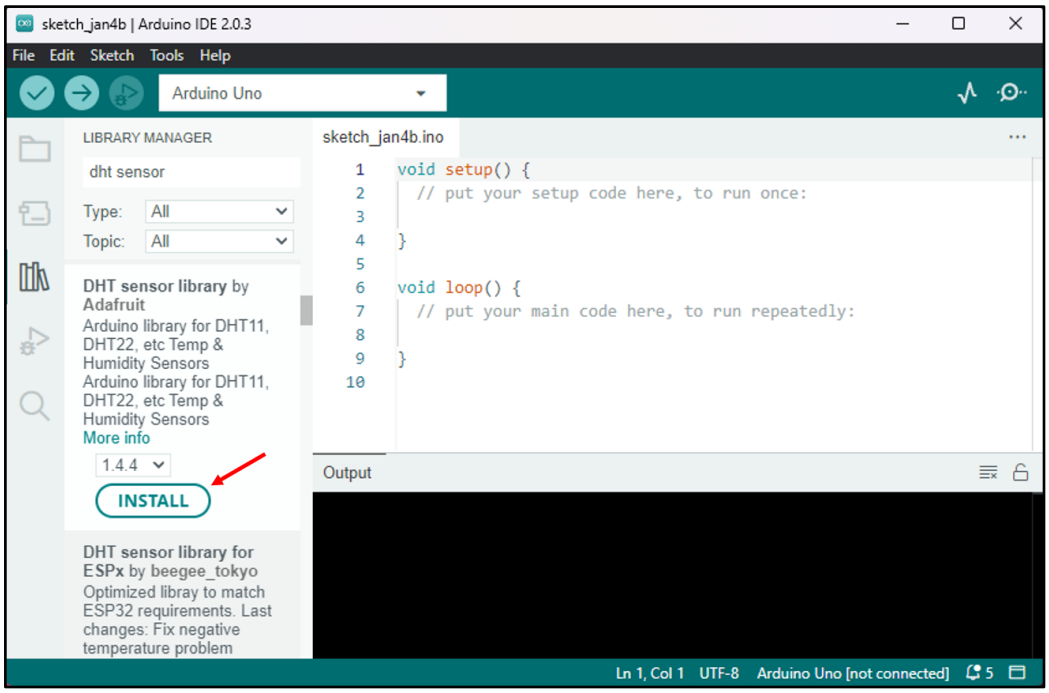
Then, search for **EspMQTTClient**. Install that library.



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

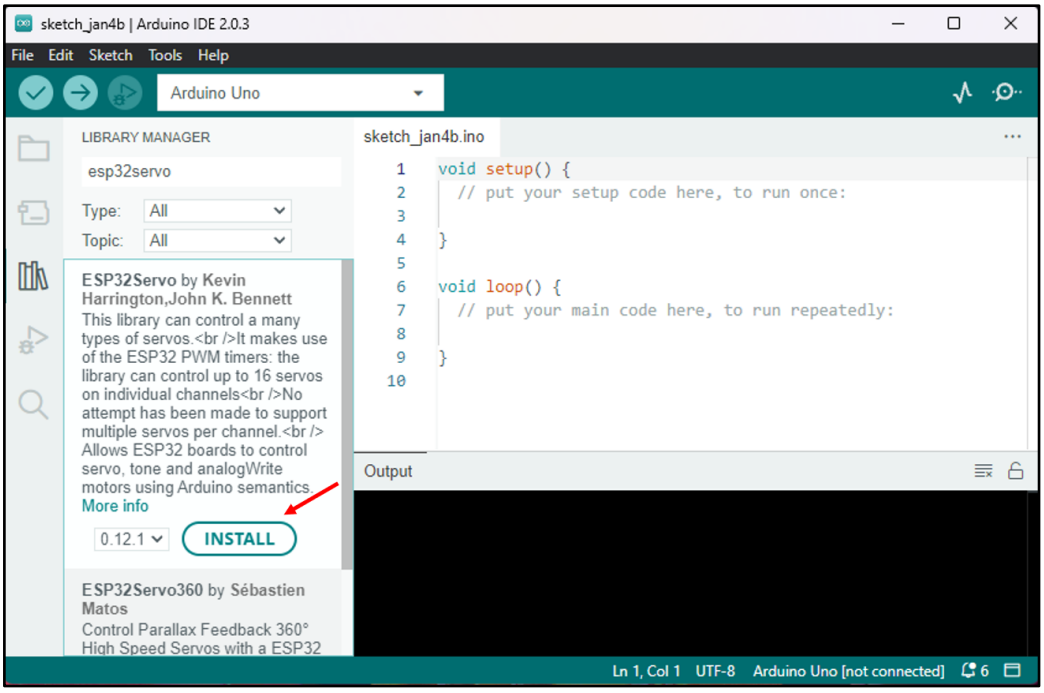


For projects that will be using any **DHT sensor**, please ensure the DHT library is also installed. Click **install all**.



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

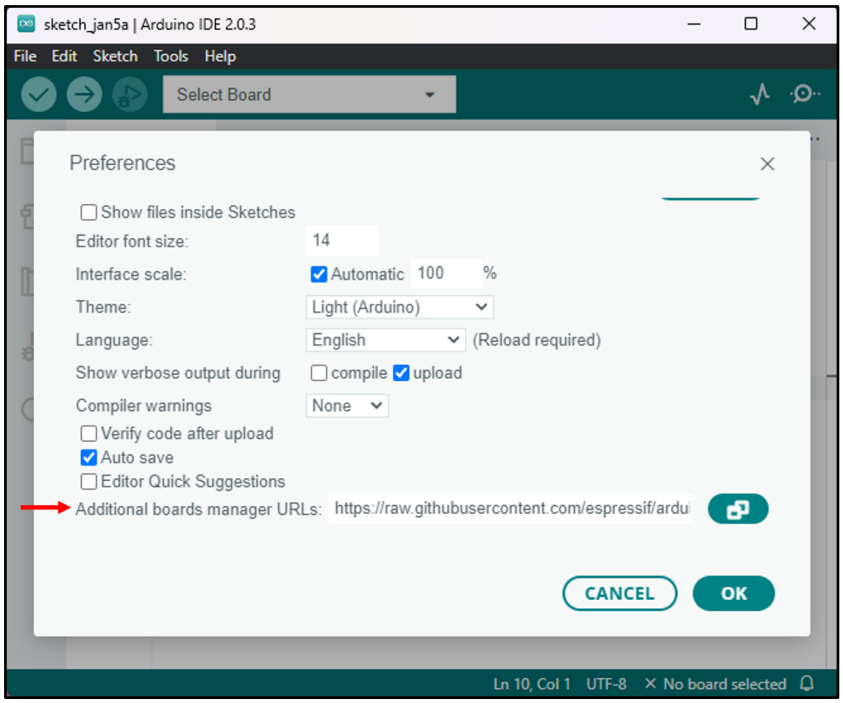
Note: Please type in "**esp32servo**" to ensure you find the right library. The ESP32 does not support the standard servo library.



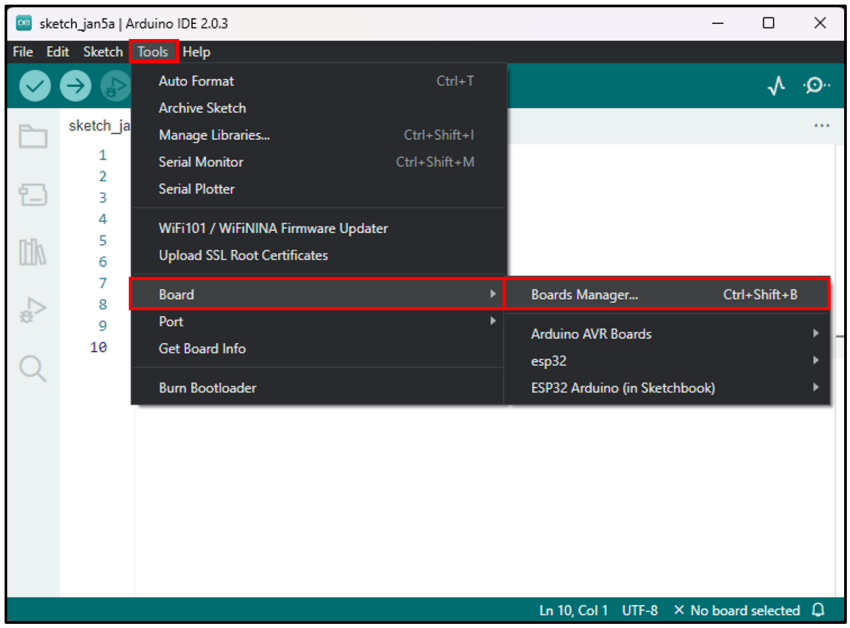
Upload Code

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.

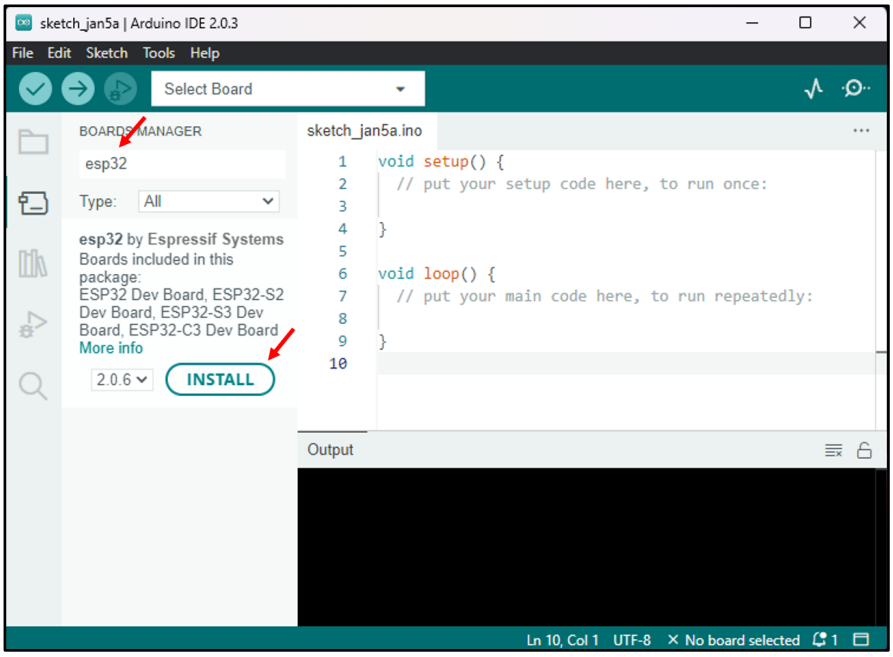
- **https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json**



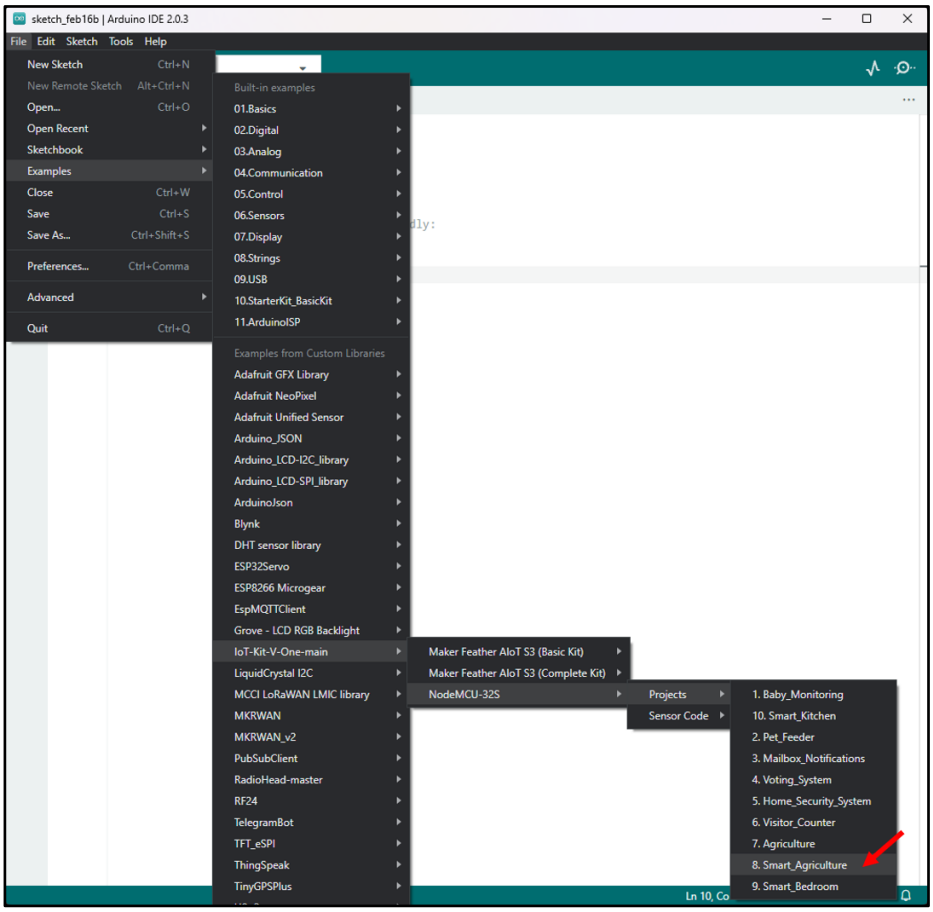
Go to **Tools** > **Board** > **Boards Manager**.



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 > NodeMCU-32S > Projects** and choose any project.

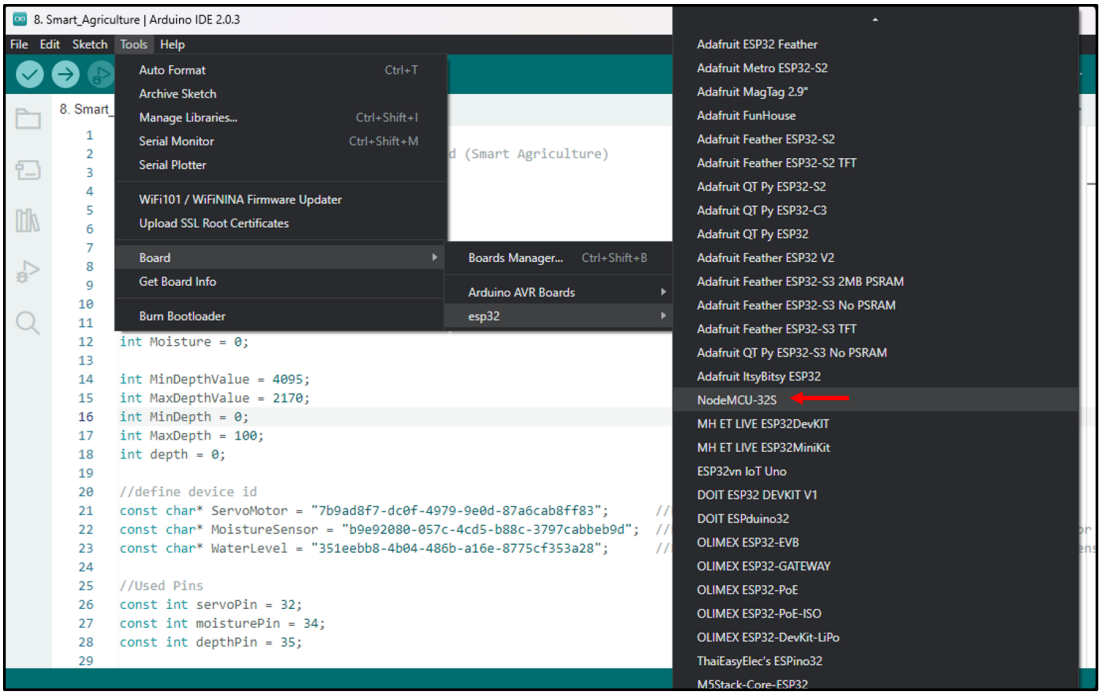


You will need to edit only one part of the project code. Find the lines that define the **deviceID** (in the red box) and replace them with **your deviceID** for each sensor.

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

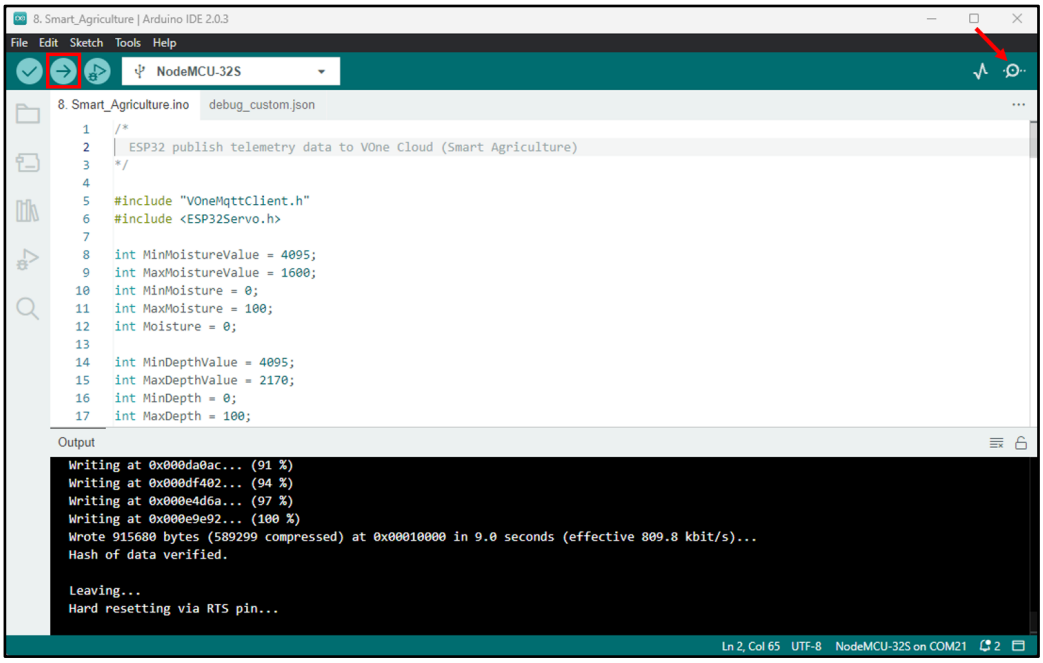
Choose the corresponding **board** and **COM port** before uploading the code.

- **Tools** > **Boards** > **esp32** > **NodeMCU-32S**
- **Tools** > **Port** > **Your COM port**



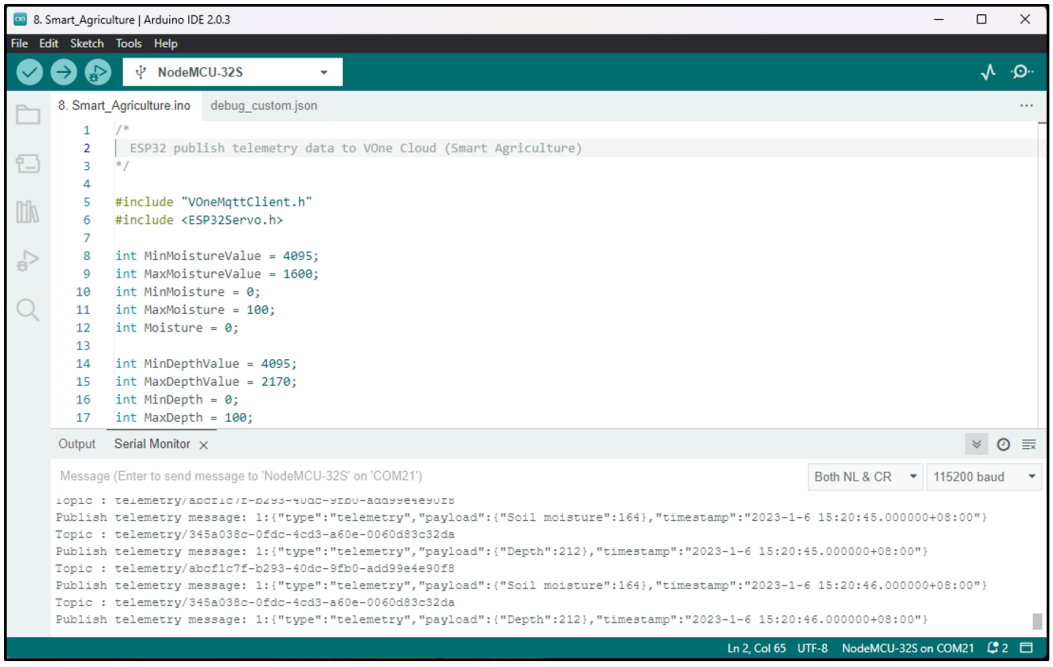
Click the Upload button and **it will take some time** to finish uploading. Once the code has been successfully uploaded, click the Serial Monitor button on the top right of the screen.

Note: Please refer to the project tutorials and make sure the hardware is connected properly before uploading the code.



Here you can see if the sensor data is successfully sent to the V-One platform or not.

Note: For the Serial Monitor, kindly choose **115200** for the baudrate. Plus, if you cannot see the data in V-One, please check again the **deviceID** of each sensor or the **hardware connections**.



Part 3

IoT Projects

Project 1: Baby Monitoring

Overview

As parents, it is very frightening to leave your baby while doing something, isn't it? We often want to check on them but at the same time, we need to focus on our tasks. With this simple project, we can monitor if there is movement from our baby using a motion sensor. Once the motion sensor detects a movement, the user will receive a notification.

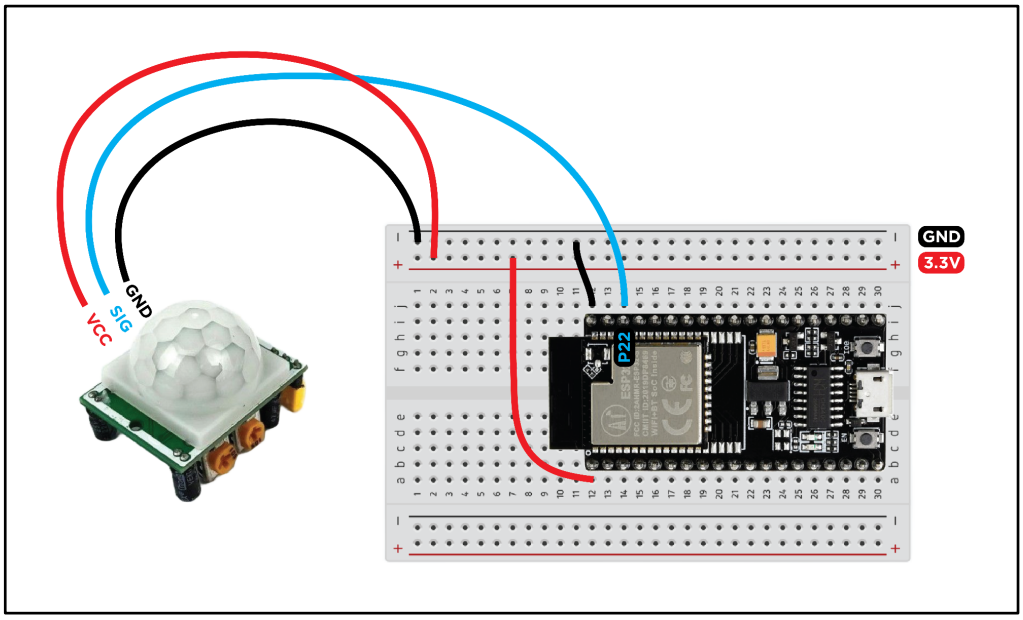
Tutorial Video

[IoT-Based Baby Monitoring System]

Required Components

- NodeMCU-32S
- Breadboard
- PIR Sensor
- Jumpers

Circuit Diagram



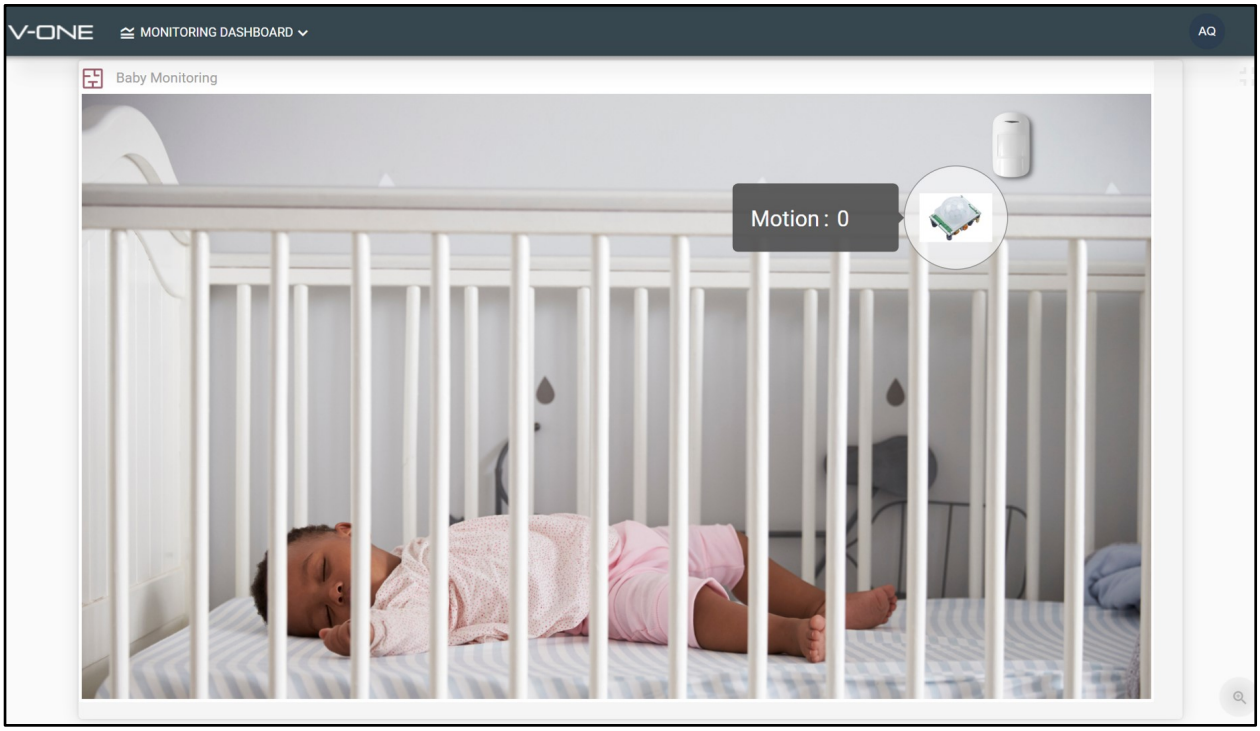
Project Code

- Baby_Monitoring.ino

Workflow Code

- For PIR Sensor

V-One Dashboard



Tips and Tricks

- Change the **Workflow deployment type** from 'immediate once' to 'intervals, 5 seconds' so that the sensor will always check for movement every 5 seconds.
- Kindly calibrate the PIR sensor first by adjusting the variable resistors, CH1 for delay time and RL2 for detection range.
- You can change the message in the email notifications by modifying the Python code in Workflow:

```
if int(motion) > threshold:
    msgbody='<p>Your baby awake. Please go check on him.</p><br>'
    output[1]="[Alert]  Motion Detected! "
    output[2]=msgbody
    output[3]=2
```

Project 2: Pet Feeder

Overview

When you go to work don't forget about your lovely pet! They must be waiting for their food. Even if you are working in the office, you still can take care of your pet. But how is that even possible? This project is mainly about how to feed your pet remotely. Using an infrared sensor, you can know the presence of your pet near the food container. When the infrared sensor detects the pet, you will get a notification which means that your pet is hungry! You can directly open the food container remotely by using the V-One dashboard. Then, change to another compartment of the container for the next meal with the help of a servo.

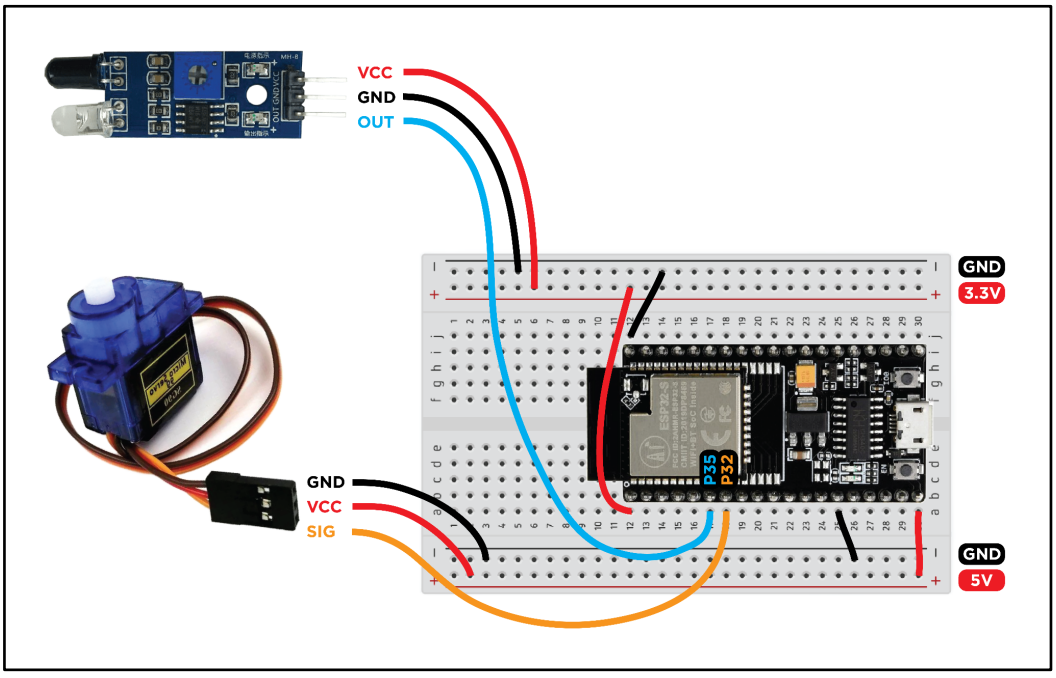
Tutorial Video

[IoT-Based Pet Feeder Using NodeMCU-32S]

Required Components

- NodeMCU-32S
- Breadboard
- Infrared Sensor
- Servo
- Jumpers

Circuit Diagram



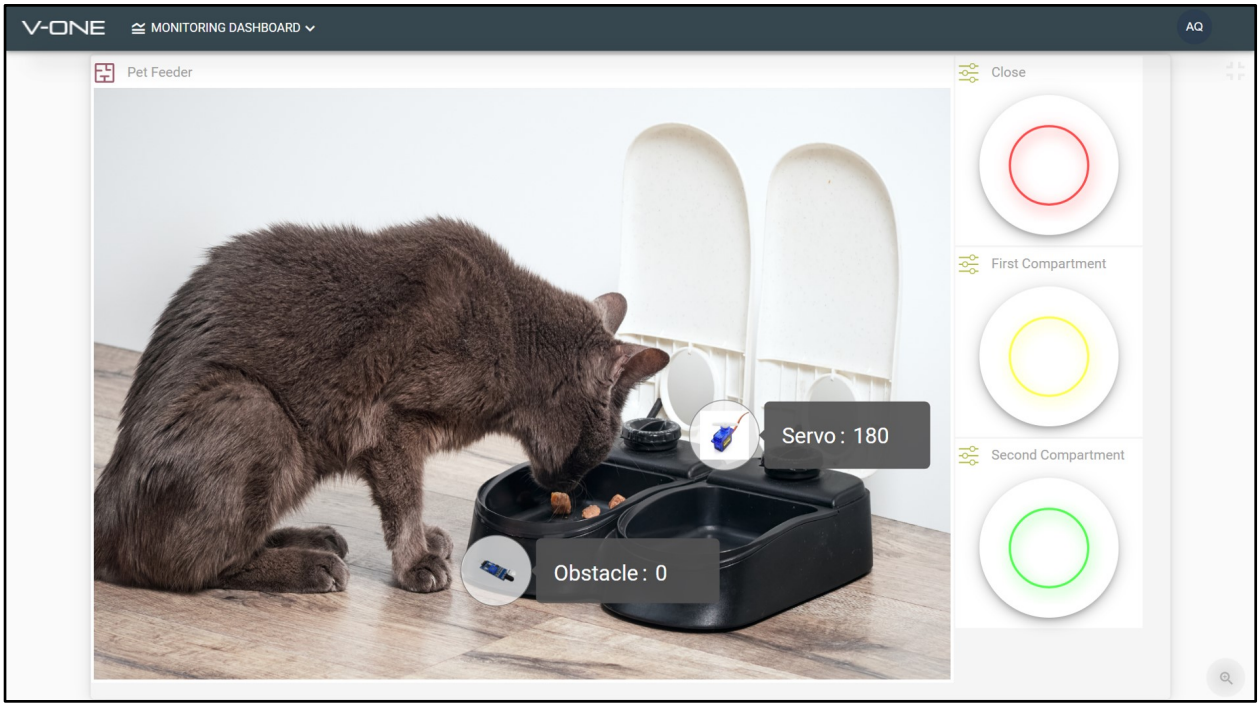
Project Code

- 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.
- If you have a 3D printer, try to print the pet feeder container using this 3D design.

Project 3: Mailbox Notifications

Overview

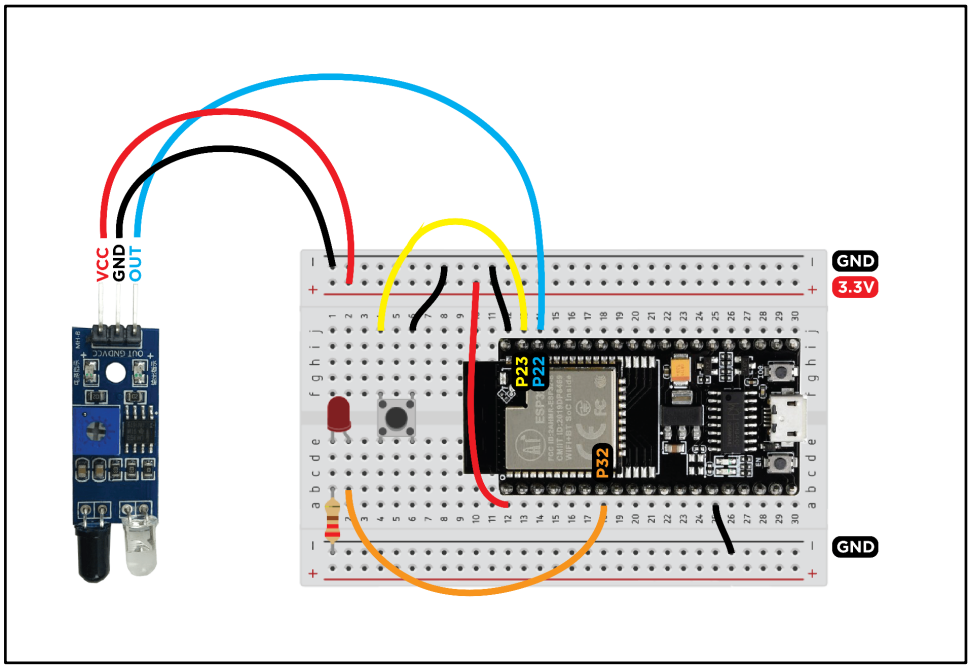
Have you always forgotten to check your mailbox and find your old important mail sitting there waiting for you? After a hectic day, we may have overlooked our mailbox and did not collect the mail sent before during the day. How to ensure we know when there is mail available in the mailbox?

Fret not! With this IoT project, you will be able to receive notifications via V-One Cloud to notify you if there is any mail inserted in the mailbox and its quantity. The infrared sensor will detect inserted mail and count its value, then the data will be sent to you. After collecting it, you may reset the count by pushing the button.

Required Components

- NodeMCU-32S
- Breadboard
- Infrared Sensor
- LED
- Button
- 220 Ohm Resistor
- Jumpers

Circuit Diagram



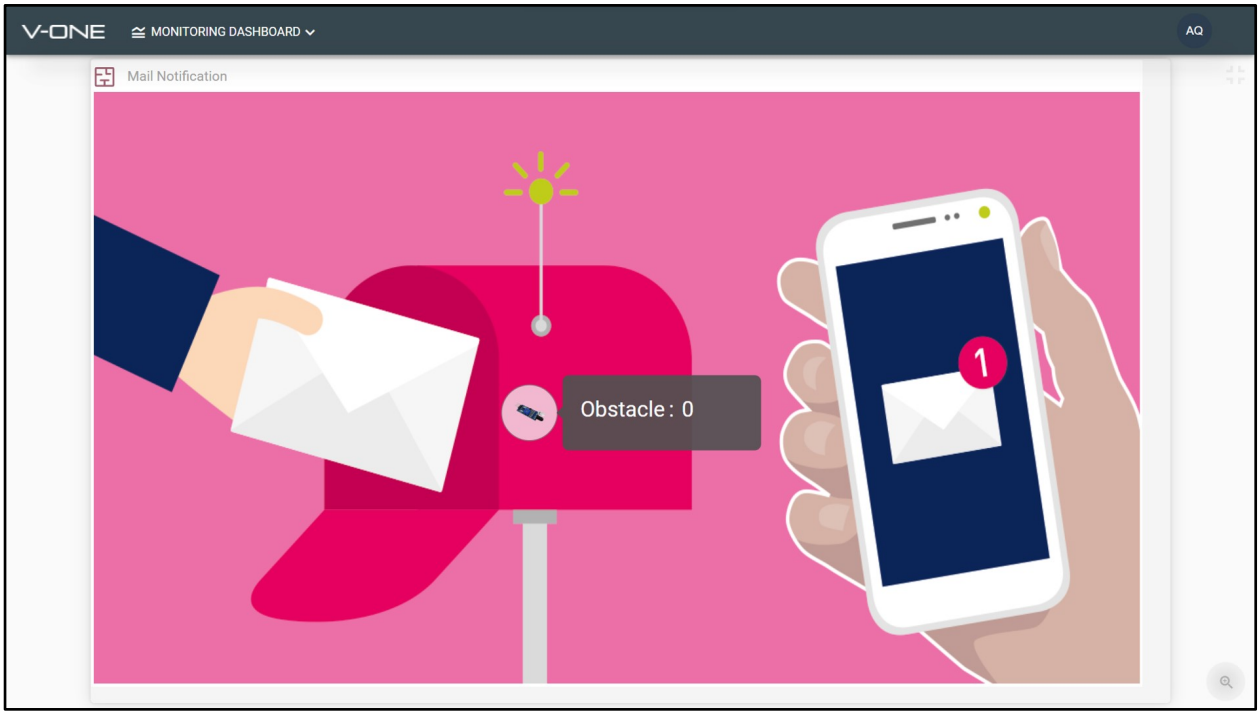
Project Code

- Mailbox_Notifications.ino

Workflow Code

- For Infrared Sensor

V-One Dashboard



Tips and Tricks

- Make sure you place your Infrared Sensor properly. If the mail moves too fast, it can't detect it and if the mail is stuck on the sensor, the counting will keep increasing.
- 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 the 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
```

Project 4: Voting System

Overview

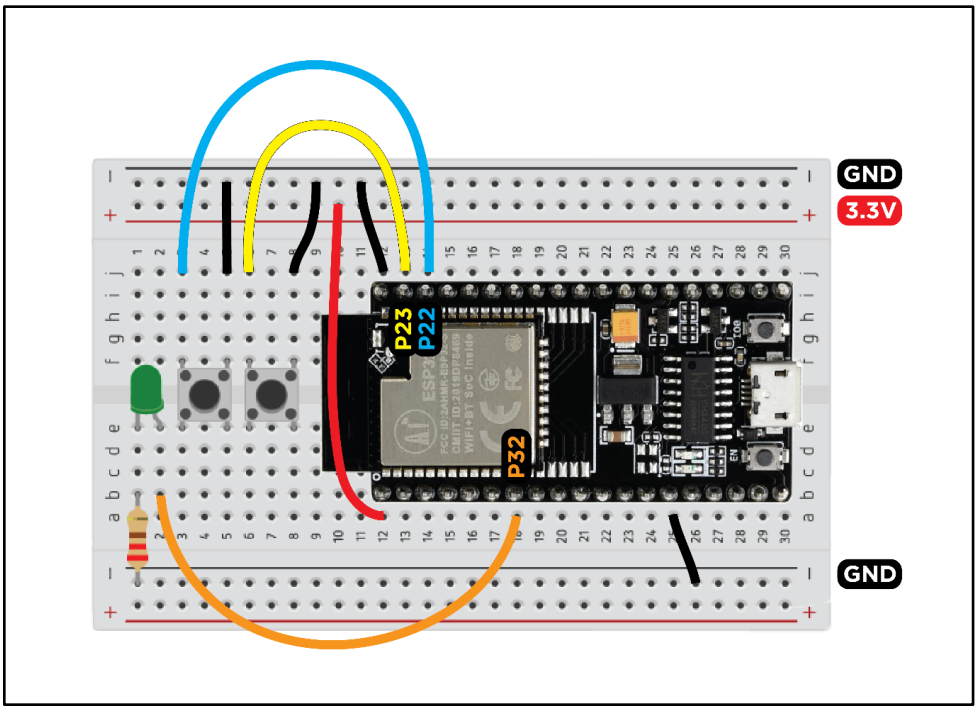
Don't you feel that sometimes voting is taking too much time? With all the errors that might occur if the voting is done manually with pen and paper and human calculating it.

With this project, the voting system can be automated in an embedded system by pressing the button and the LED will blink to validate the vote. Not only can this system be used to receive voting, but it also can automatically calculate the result and transfer the data to the admin via V-One Cloud. This could prevent spoilt votes and provide transparency.

Required Component

- NodeMCU-32S
- Breadboard
- 2 x Push Button
- LED
- 220 Ohm Resistor
- Jumpers

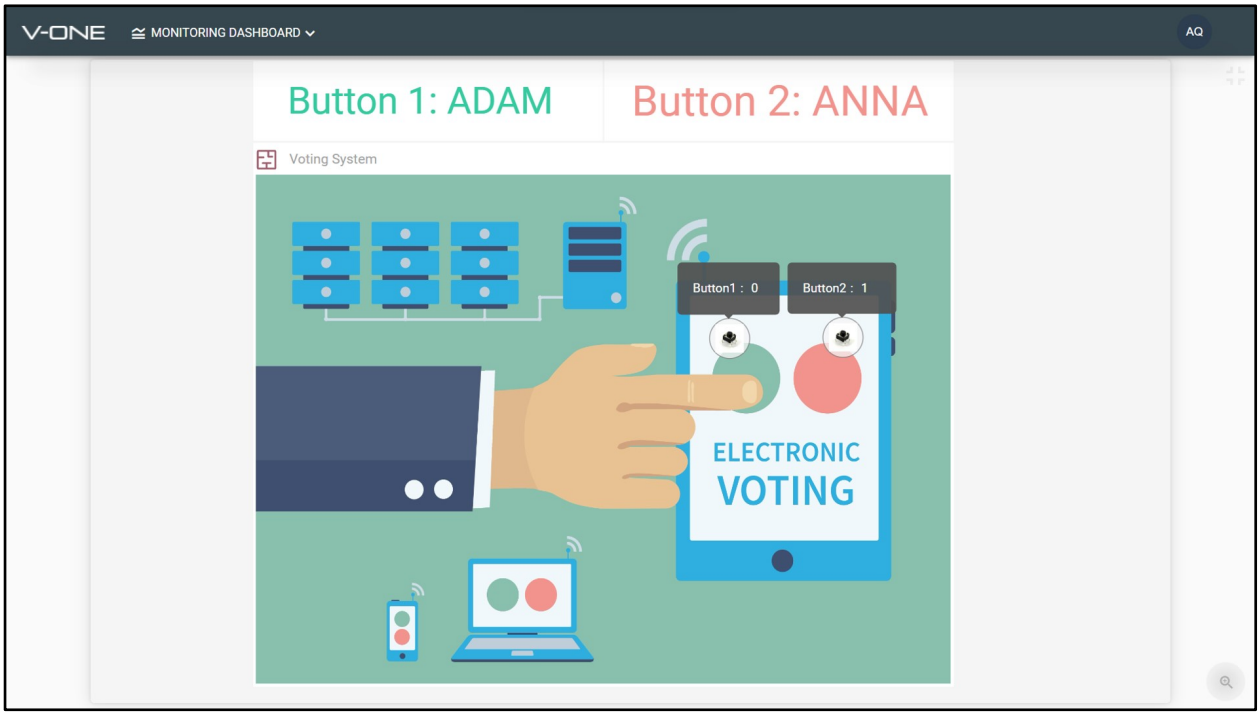
Circuit Diagram



Project Code

- Voting_System.ino

V-One Dashboard



Tips and Tricks

- 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 5: Home Security System

Overview

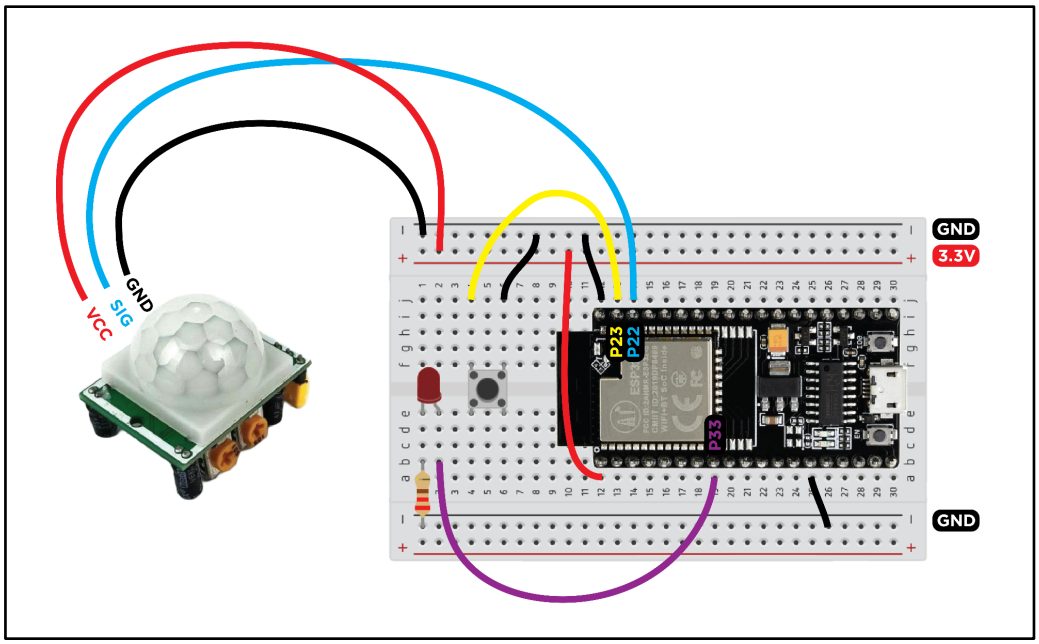
Home is a place where we will feel safest the most. But with the crimes that might occur, there is always risk anywhere. How to build a good home security system?

When this project is turned on, the PIR motion sensor will detect if there is a motion at where the sensor is placed, or when someone triggers the button. The LED that is placed in the house will turn on and the house owner will be notified about the intruder via V-One Cloud.

Required Components

- NodeMCU-32S
- Breadboard
- PIR Motion Sensor
- LED
- Push Button
- 220 Ohm Resistor
- Jumpers

Circuit Diagram



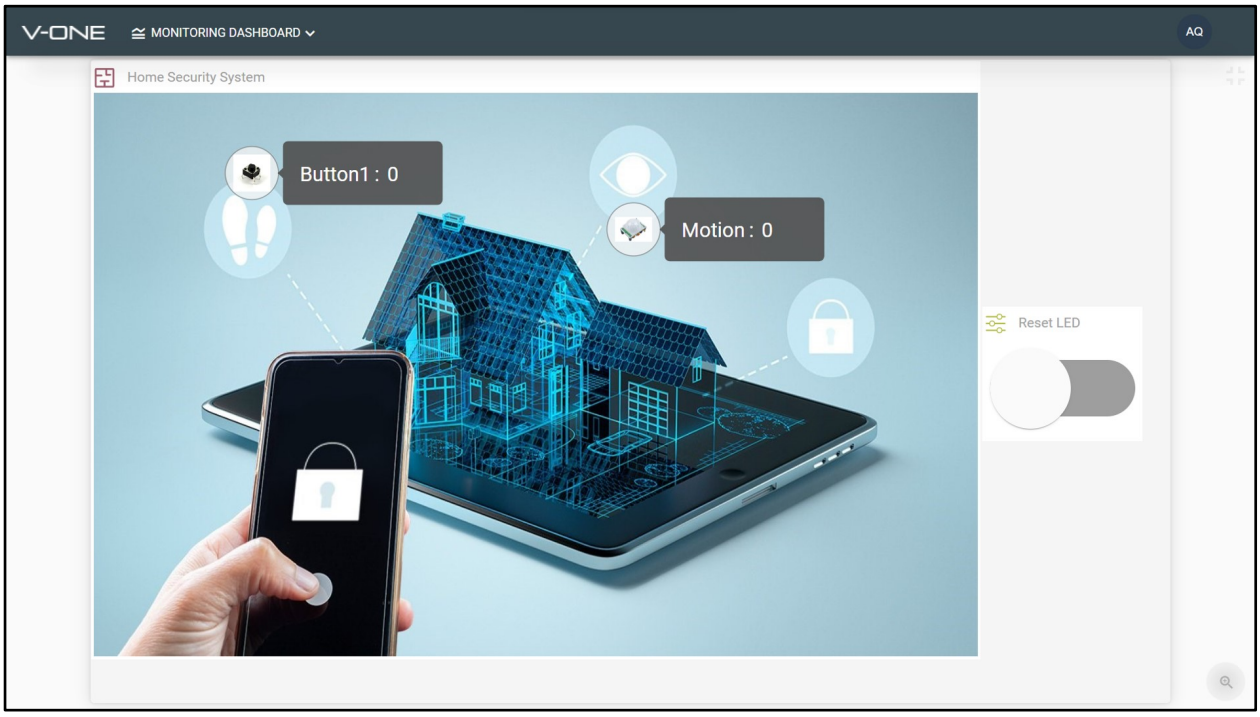
Project Code

- Home_Security_System.ino

Workflow Code

- For PIR Sensor
- For Push Button

V-One Dashboard



Tips and Tricks

- The LED will not turn off automatically after it is triggered. This is like a safety feature where you must confirm first there is no intruder around then you can reset the LED using the V-ONE dashboard.
- Since this project involves two parameters, you are preferred to use two workflows. One for the button and one for the PIR sensor.

Project 6: Visitor Counter

Overview

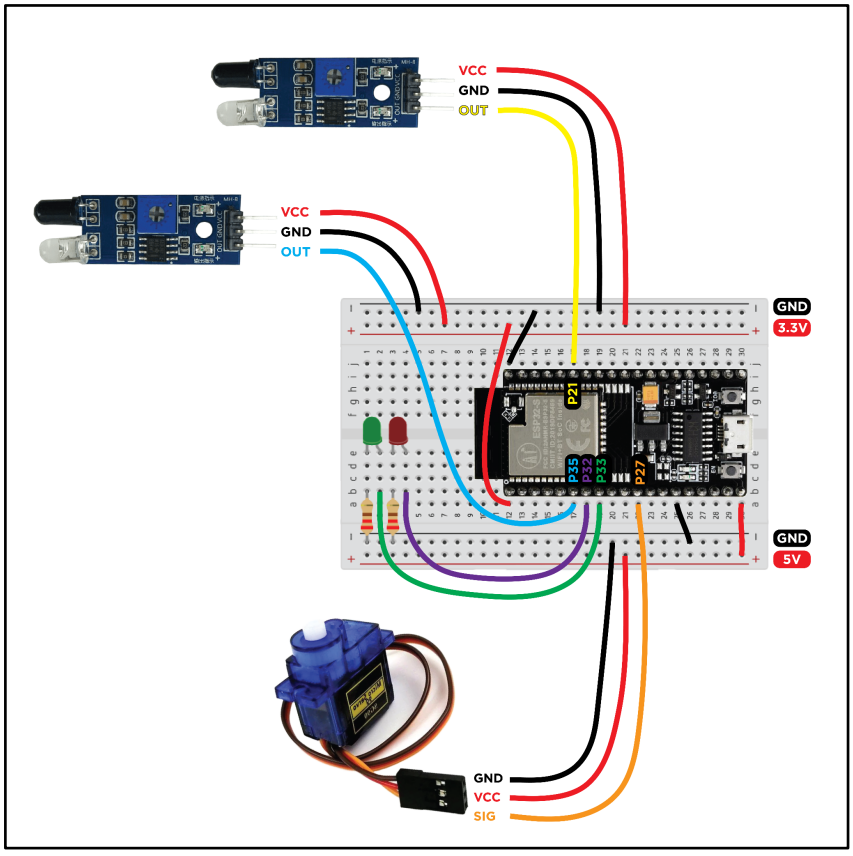
If you have a store and you want to limit the number of customers in the premise, how can you do it? Or if you want to track down the timing trend of customers coming into your store, is it possible?

This project is equipped with two infrared sensors that can be placed at the entrance and exit of your store. It can detect any person coming in or going out. We can get the value of the people in the store. With the servo, we also can shut the door if the number of people inside the store is exceeding the limit.

Required Components

- NodeMCU-32S
- Breadboard
- 2 x Infrared Sensor
- Micro Servo
- 2 x LED
- 2 x 220 Ohm Resistor
- Jumpers

Circuit Diagram



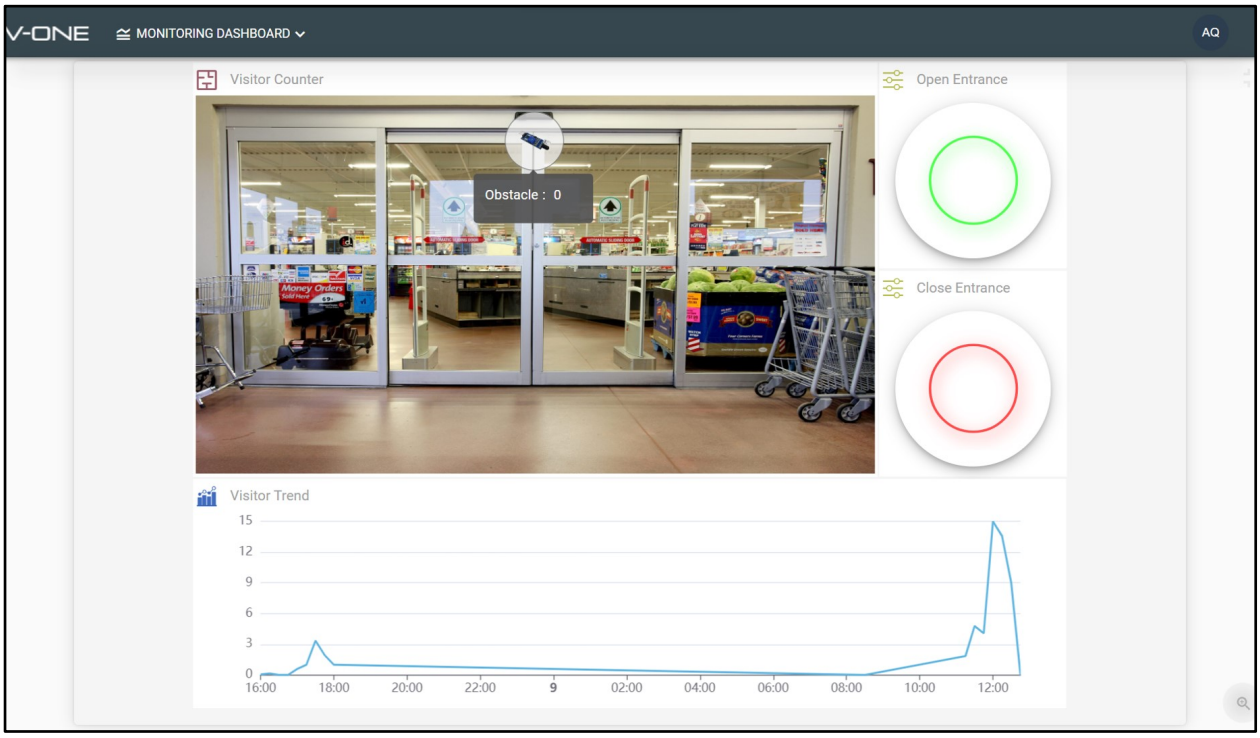
Project Code

- Visitor_Counter.ino

Workflow Code

- For Infrared Sensor

V-One Dashboard



Tips and Tricks

- The first infrared sensor increases the value while another one decreases the value. The last output will be the remaining visitor in the store.
- 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.

```
19 unsigned long lastMillis = 0;
20 volatile int count1 = 0;
21 int limit = 10;
```

Project 7: Agriculture

Overview

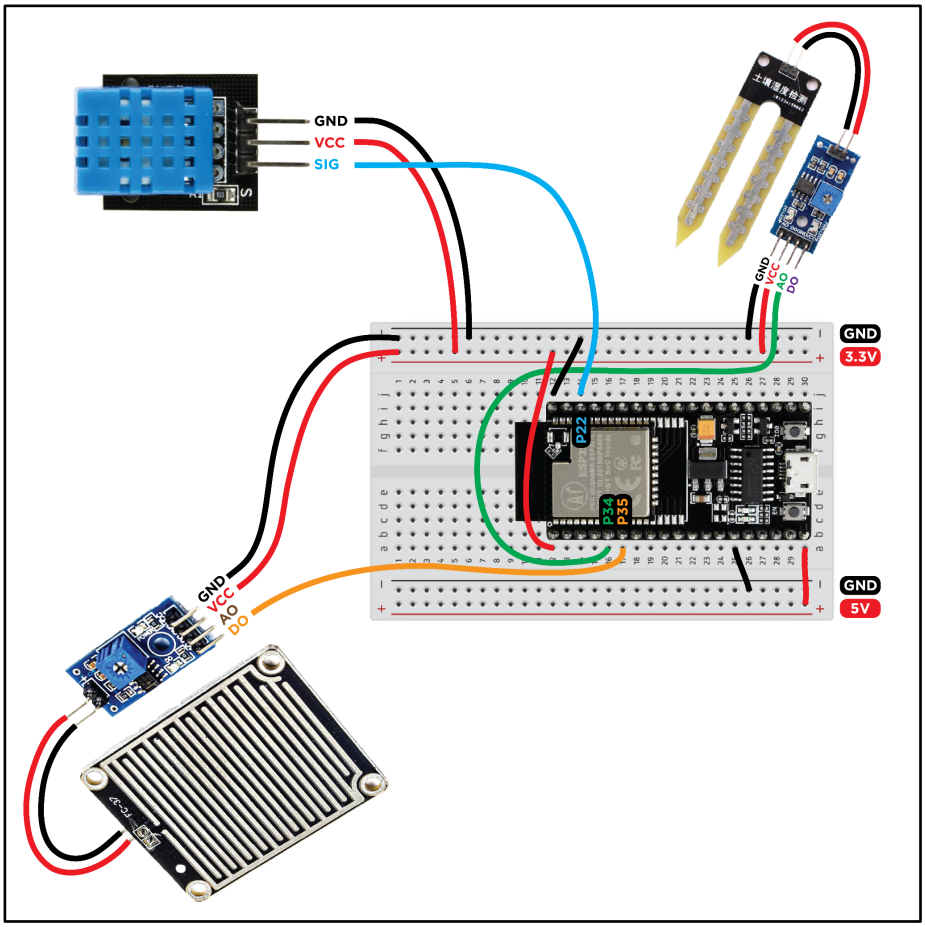
Being able to monitor the condition of your agriculture farm is really important. With technology nowadays, we can take care of our plants even if we are not at the farm. Interesting isn't it?

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 Agriculture project are soil moisture, precipitation trend, temperature, and humidity.

Required Components

- NodeMCU-32S
- Breadboard
- DHT11
- Moisture Sensor
- Rain Sensor
- Jumpers

Circuit Diagram



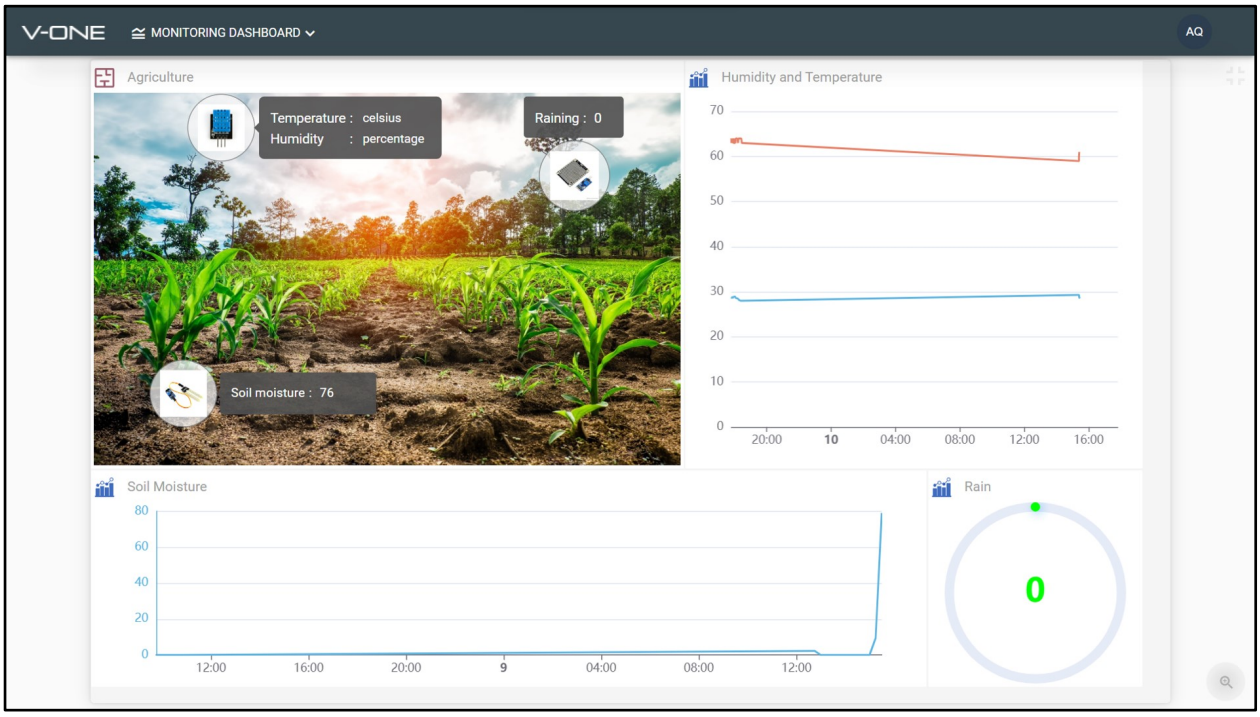
Project Code

- Agriculture.ino

Workflow Code

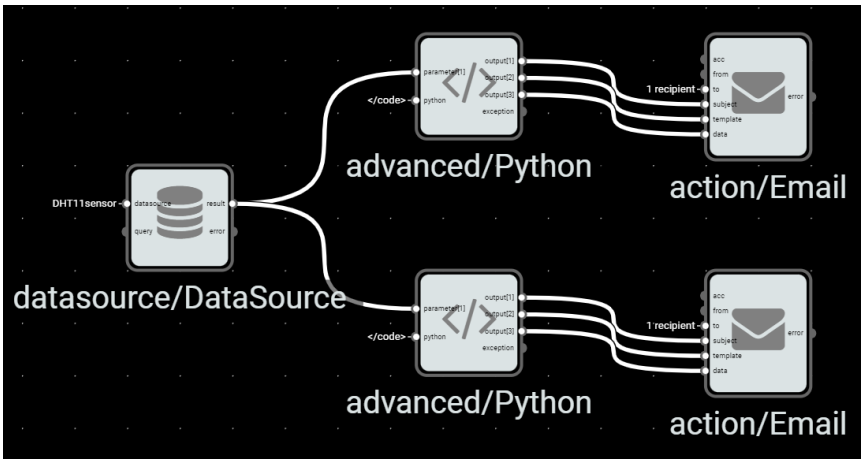
- For Temperature
- For Humidity
- For Moisture Sensor

V-One Dashboard



Tips and Tricks

- The workflow for DHT11 should look like this. Two Python sections for both humidity and temperature.



- In this project, you can change the **threshold value** of humidity, temperature, and soil moisture 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 int(moisture) < threshold:
```

- If you realize that the value of soil moisture is not very accurate, you can modify **MaxMoistureValue** in the Arduino code. The **maximum** output value should be 100 (%).

```
 8 int MinMoistureValue = 4095;
 9 int MaxMoistureValue = 1800;
10 int MinMoisture = 0;
11 int MaxMoisture = 100;
12 int Moisture = 0;
```

- It can be more interesting if you can add a water pump to this Agriculture project. If you like to buy one, you can always refer here.

Project 8: Smart Agriculture

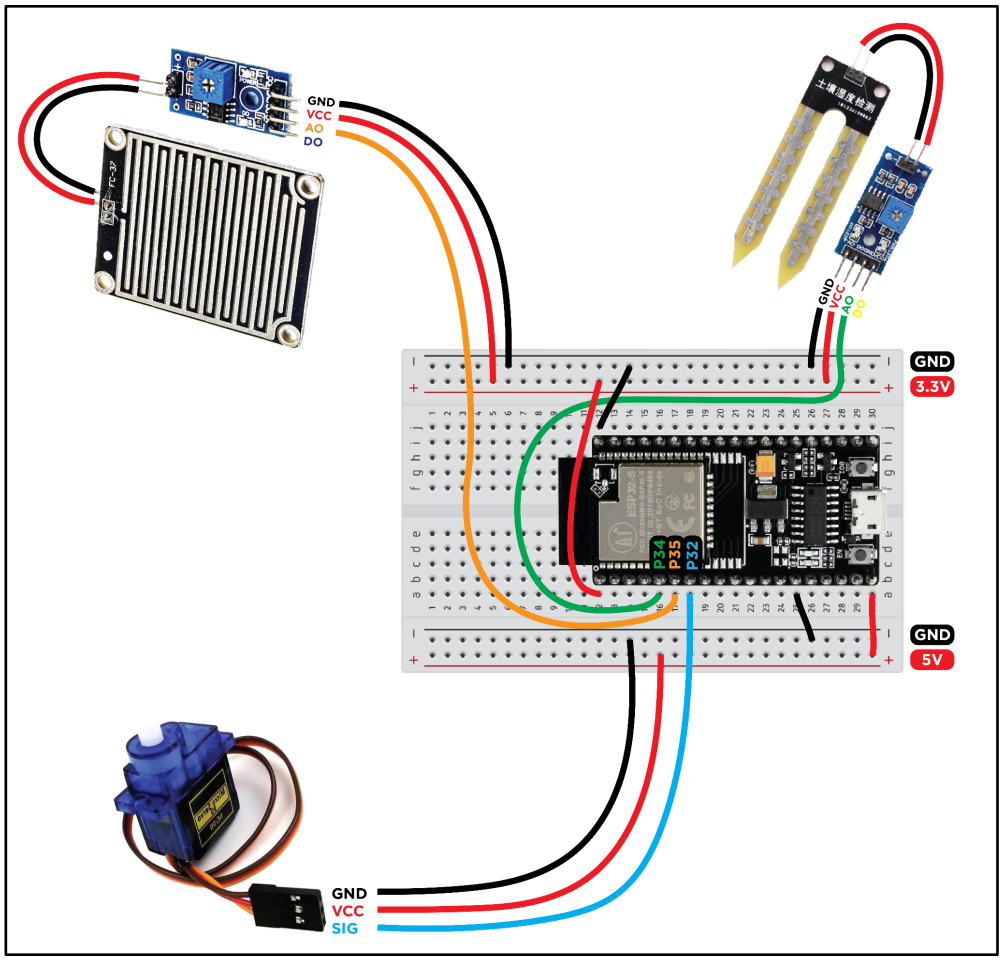
Overview

Different from the previous project, this one 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

- NodeMCU-32S
- Breadboard
- Moisture Sensor
- Water Level Sensor
- Servo
- Jumpers

Circuit Diagram



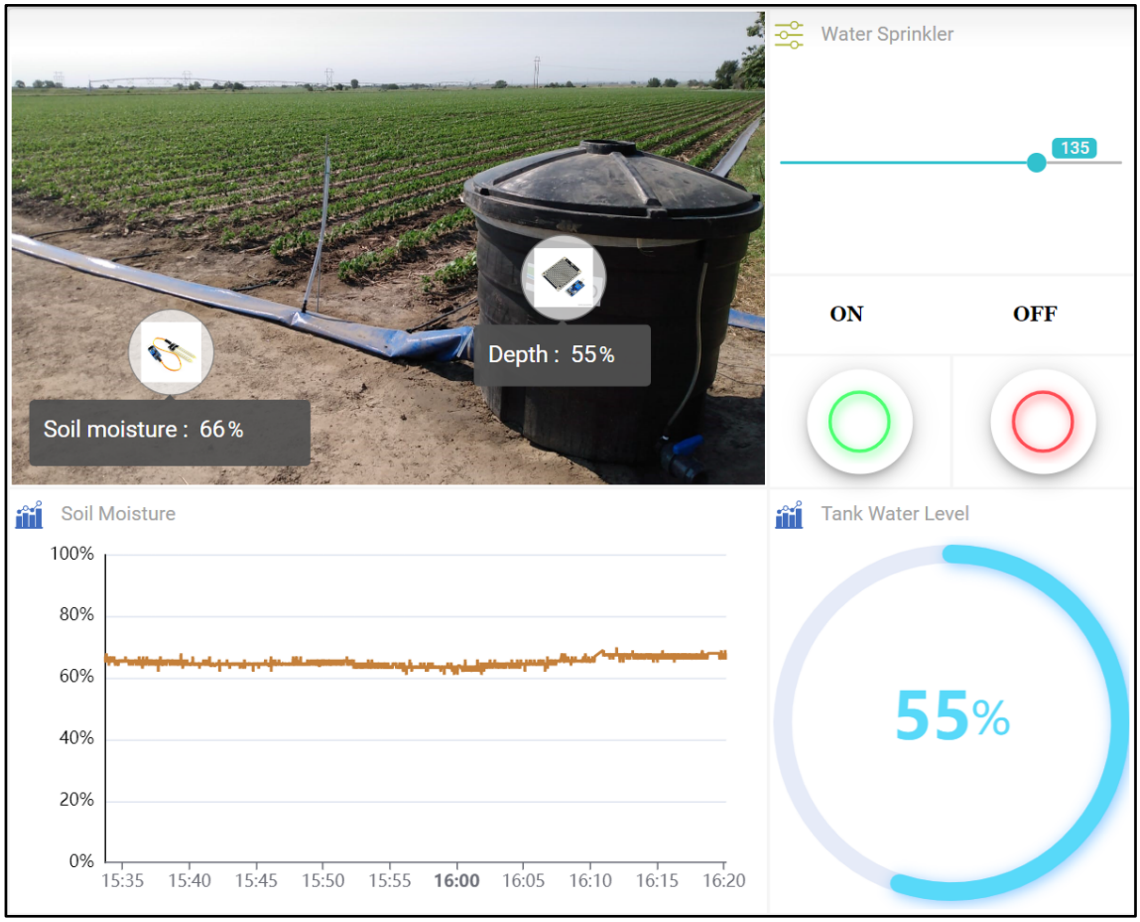
Project Code

- 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.
- The tips would be the same as in Project 7. You can change the **threshold value** in the workflow according to your preferences.
- If you got inaccurate sensor readings, modify the value of the following lines in Arduino IDE until you get the right maximum output value (100%).
    - **MaxMoistureValue** for the moisture sensor
    - **MaxDepthValue** for the water level sensor

Project 9: Smart Bedroom

Overview

When we come back home after a long day, the bedroom is the best place to rest our minds, and sometimes it is a hassle to turn on all the switches of fans and lamps. Or maybe sometimes we want to open all the fans and lamps earlier even before entering the bedroom. But how to make that possible?

With IoT, everything is possible. In this project, we will control a fan and three lamps in the bedroom from the V-One Dashboard that can be accessed from a web browser on a smartphone, tablet, or computer.

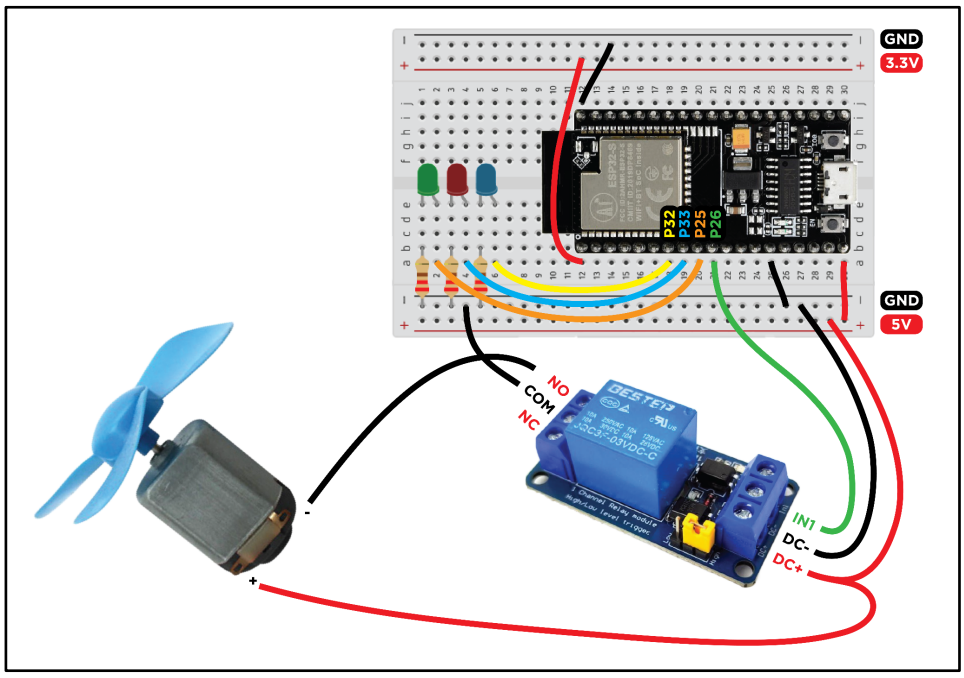
Tutorial Video

IoT-Based Smart Bedroom Using NodeMCU-32S

Required Components

- NodeMCU-32S
- Breadboard
- 3 LEDs
- Relay
- Motor & Propeller
- Resistor (3 unit 1k ohm)
- Jumpers

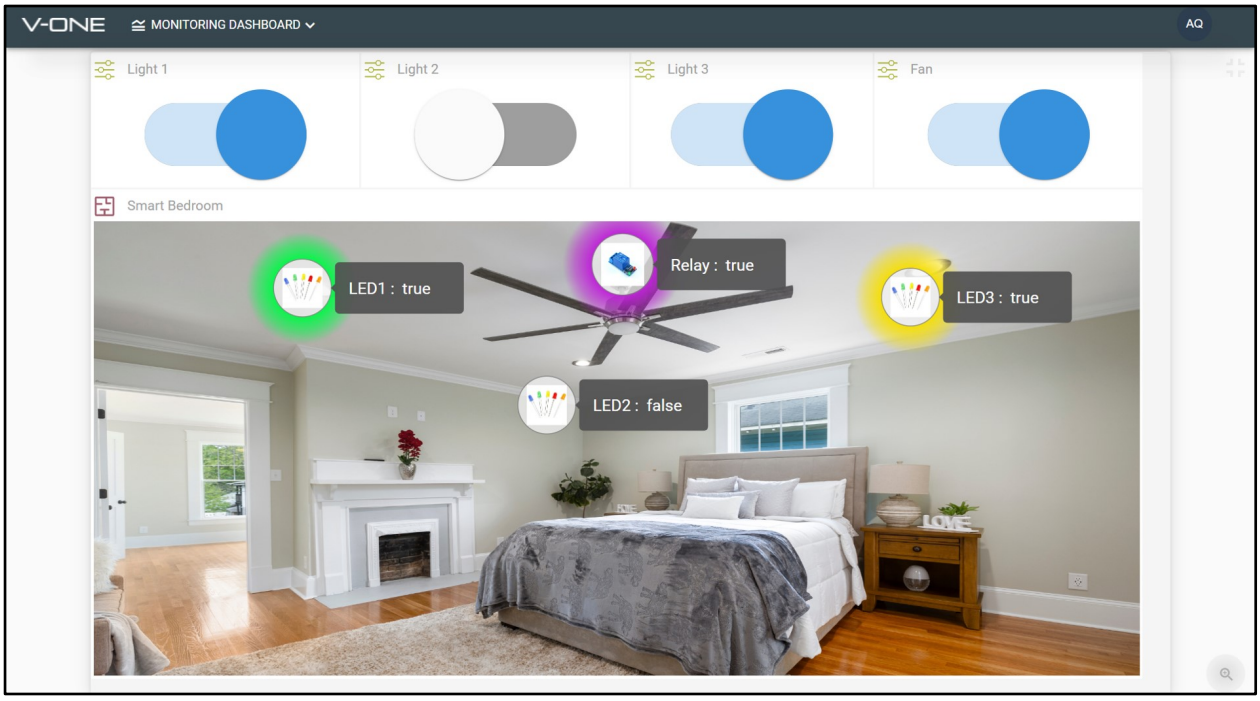
Circuit Connection



Project Code

- Smart_Bedroom.ino

V-One Dashboard



Tips and Trick

- Kindly double-check the connections of the relay before uploading the code.
- You can also use the button for the widgets instead of the switch but please be reminded that the button is a one-way function. So, you need to create two buttons for each actuator, one for turning ON and the other one for turning OFF.

Project 10: Smart Kitchen

Overview

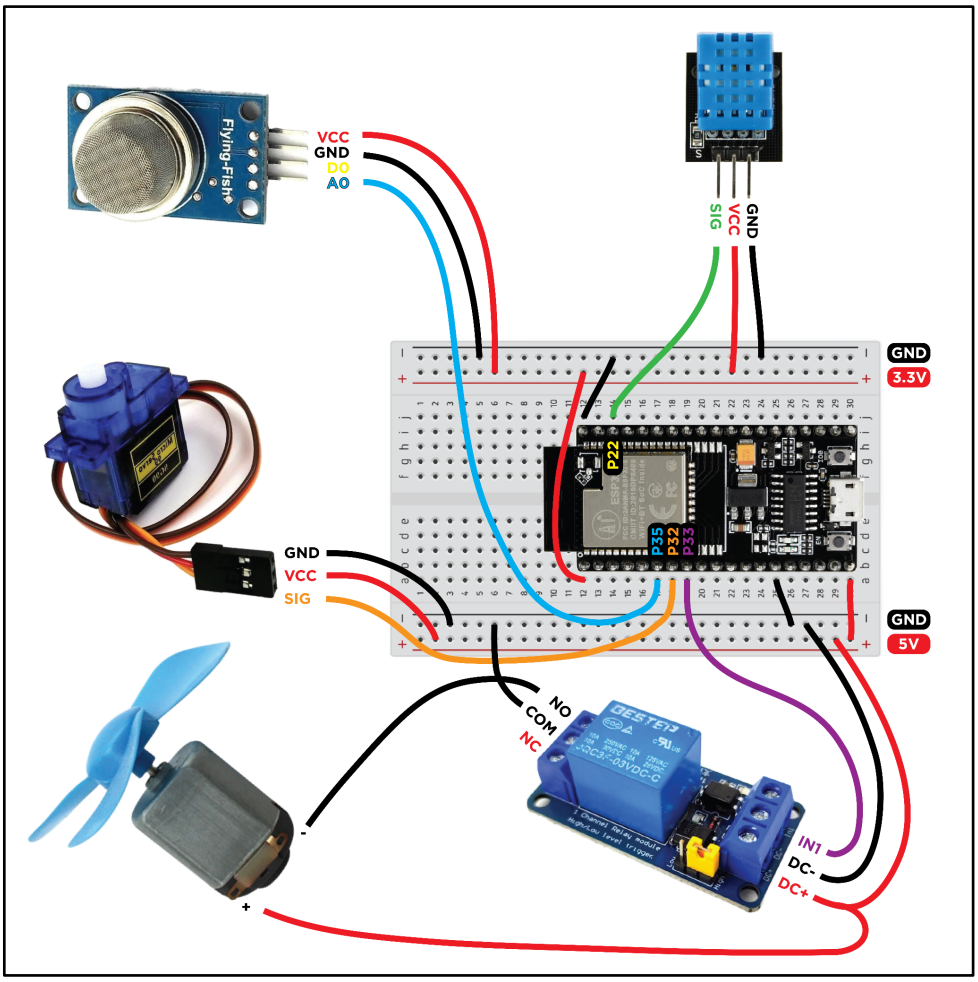
Cooking with comfort is one of the best moments in any house. However, the kitchen is also a place with a lot of risks and any accident might occur. How to control the kitchen to ensure nice surroundings while being either at home or outside?

With this smart kitchen project, the user will be notified if the DHT11 sensor shows that the temperature is high, the humidity is low and the MQ2 gas sensor indicates that the concentration of carbon monoxide in the air is high. After receiving the notification, the user can open the kitchen window and turn on the fan from the V-One Dashboard while at home, or even outside of the house.

Required Components

- NodeMCU-32S
- Breadboard
- MQ2 Gas Sensor
- DHT11 Sensor
- Servo
- Relay
- Motor + Fan Blade
- Jumpers

Circuit Connection



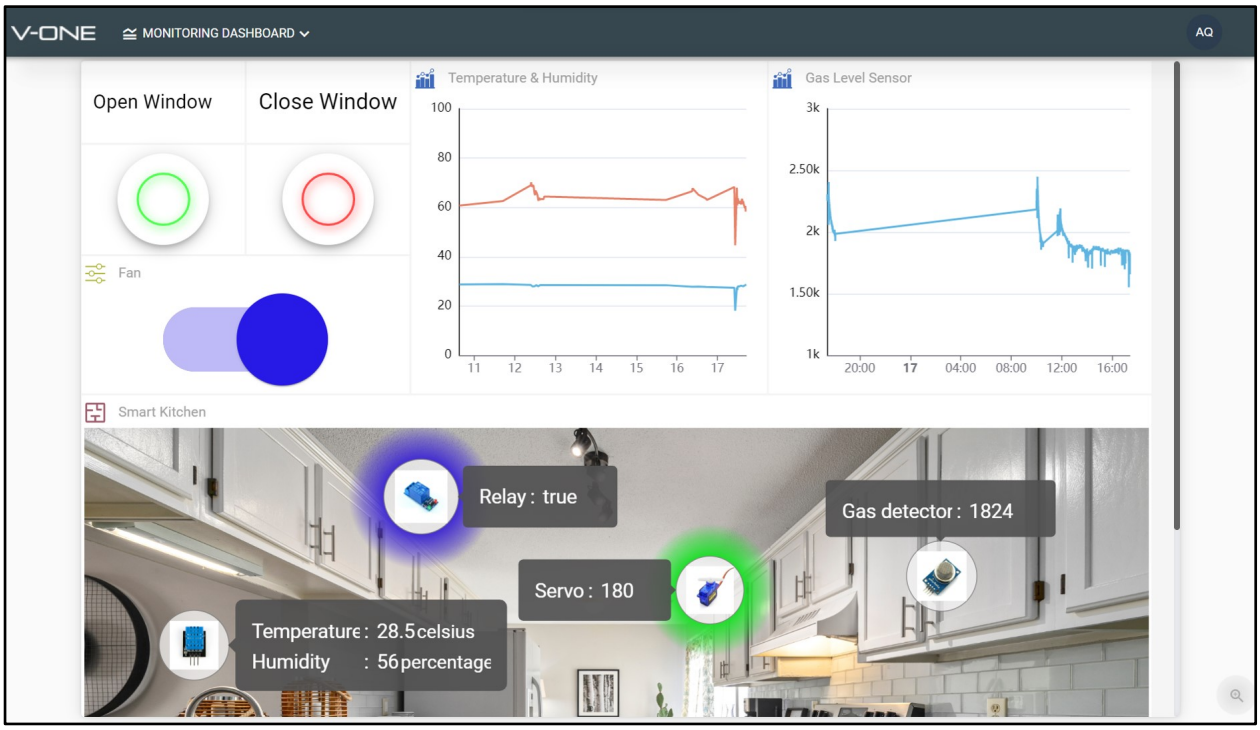
Project Code

- Smart_Kitchen.ino

Workflow Code

- For MQ2 Sensor
- For Temperature

V-One Dashboard



Tips and Tricks

- Please wait for **20 seconds** for the gas sensor to warm up after you upload the code.
- Kindly double-check the connections of the relay before uploading the code.
- Recommended to test the MQ2 sensor with a lighter.
- You can change the threshold value of **humidity**, **temperature**, and the **gas sensor** for the email notifications in the workflow.

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.