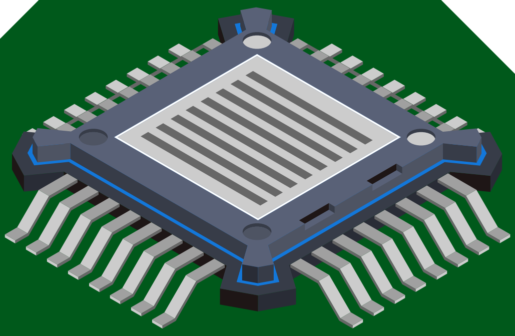
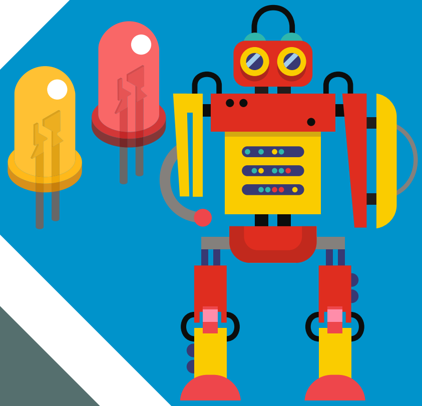
DELL Technologies

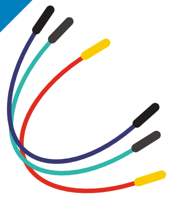
LEARNING LINKS
FOUNDATION
WHERE EDUCATION MEETS LEARNING





Emerging Technolgogies

Internet of Things



Project 1

Table of Contents

Internet of Things

Project 1: Smart Home

In this project, you will explore the basic concepts of IoT by building a circuit that responds to the inputs given by the user. You will build a smart home circuit with simple electronic components that replicate the light and a fan in a real-time scenario. Exciting? Let's get started!

Software:

1. cloud.arduino.cc

This website helps in building the IoT applications and programming the Node MCU.

2. Arduino Create Agent

This application helps in communicating with the Node MCU and the computer. It helps in dumping the code in the controller.

Components Used:

STEP by STEP Instructions

To build this project we will need the following hardware. Let's quickly look at the hardware and recall the purpose and function of the hardware in our circuit.

1. ESP8266 Node MCU

Node MCU is an open-source platform based on ESP8266 which can connect objects and let data transfer using the ESP8266 Wi-Fi module integrated in it. This board had two buttons "reset" and "flash". We can power up the board using the 3.3V DC through the "Vin" pin or 5V through a micro USB port. It has one "A0" Analog pin and 16 GPI/O pins. It has a flash memory of 4MB.

You will need 1 ESP8266 Node MCU for this project.

2. LED

LED stands for "Light Emitting Diode". It is a two terminal device. The shorter terminal is called the cathode and the longer terminal is the Anode.

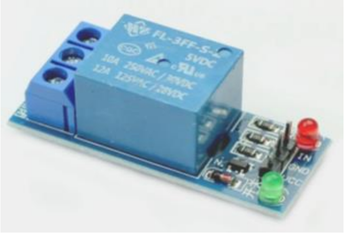
You will need 1 red coloured LED for this project.

3. Fan

A simple two terminal device which works with 5 volts input.

You will need 1 Fan for this project.

4. Relay



Relay is an electro-mechanical switch. It is used in the circuit where the switch can be controlled with low voltage. It works on the principle of electromagnetism.

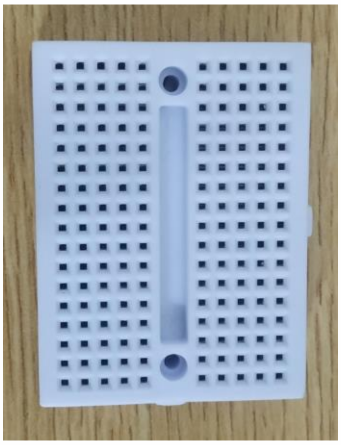
You will need 1 Relay for this project.

5. Male to Female jumper wires

Male to Female Jumper Wires are used to connect the LM35 Temperature Sensor to the Breadboard where Node MCU is mounted. These jumper wires have a male pin at one end and female pin at the other end. A male pin is commonly referred to a solid pin that stands up. A female connector is commonly referred to as a jack with a hole in it to accept the male pin.

You will need 10 Male to Female Jumpers for this project.

6. Breadboard



Breadboards are used for building circuits easily. We can easily make and break the circuit. No soldering or PCB is required.

You will need 1 Breadboard for this project.

7. Micro USB Cable.

Micro USB is used to dump the program in the Node MCU board. It is also used to power-up the board.

You will need 1 Micro USB Cable for this project.

8. Battery wire with Clip.

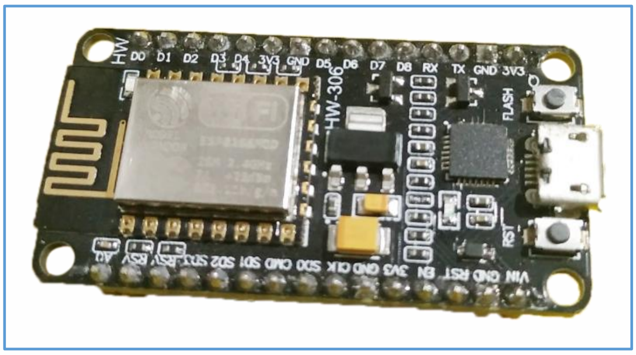
The battery supplies power to the circuits and the clip enables a proper connection between the battery and the circuit.

PART 1: Building the Circuit

Step 1

Let’s start building our "Smart Home" circuit.

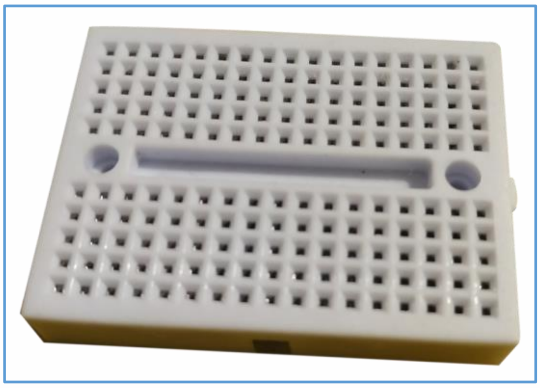
Take a Node MCU board and place it on the work table.



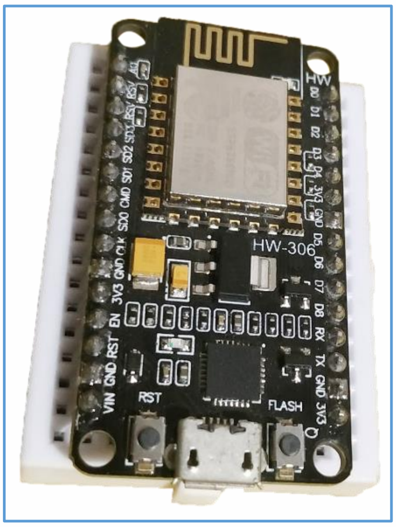
Step 2

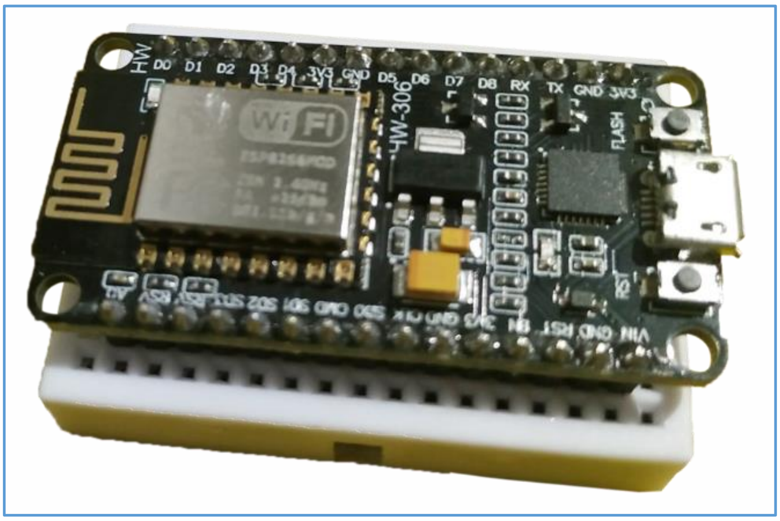
Observe the Node MCU board, it has Male pins. These male pins are not user friendly to build a circuit with different connections hence let’s place the Node MCU on a breadboard.

Take a breadboard and place it on the work table.



Then, place the Node MCU board on the Breadboard as shown in the figure.





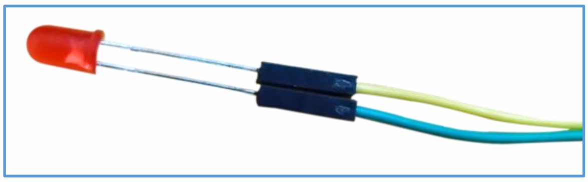
Step 3

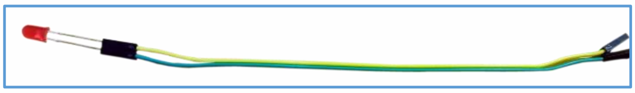
As we are building a smart home circuit that has a light and a fan. We will first begin by connecting the light which is an LED in our circuit.

Take a LED and place it on the work table.

LED has one long terminal (positive) and a short terminal (negative).

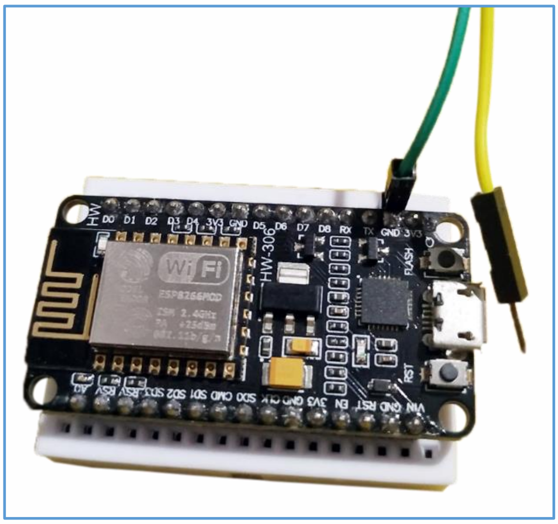
Connect the long terminal to the female end of the yellow male to female jumper wire and the short terminal to the female end of a green male to female jumper wire.



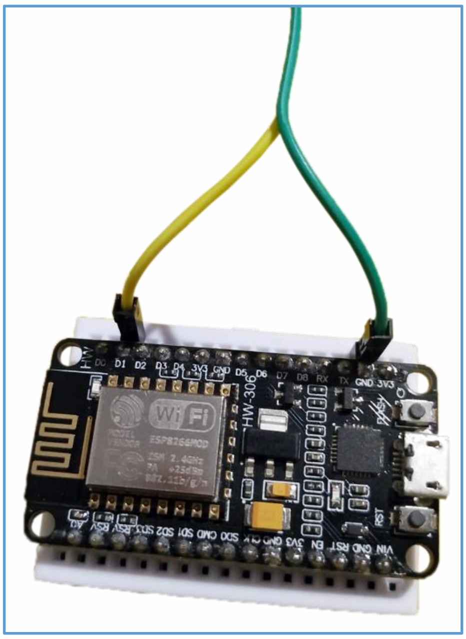


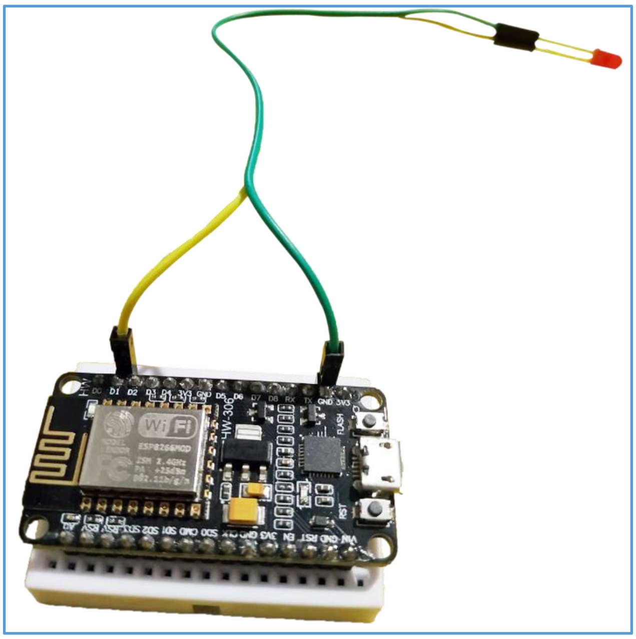
Step 4

Connect the Male end of the green jumper wire to the GND (on the flash button side, beside 3V pin) of the Node MCU.



Now connect the yellow wire of the LED to the Node MCU D1 pin as shown in the image given below.



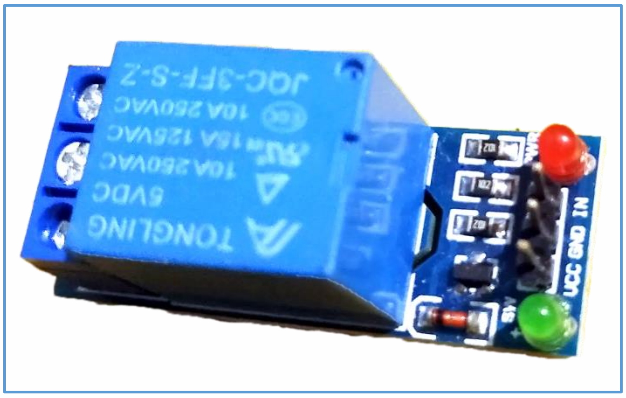


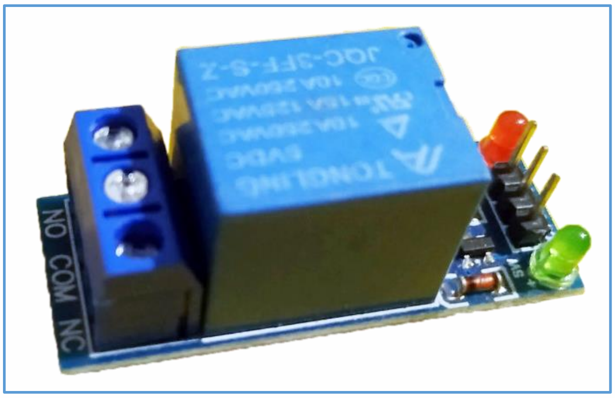
Step 5

We have completed the LED circuit. Now let’s connect the fan.

To connect the fan we will first use a relay because the fan requires more than 5 Volts but we only get 3.3 Volts from the Node MCU board.

We are using a 9V battery to power up the fan and control the switch of the fan using a relay.





The relay has 6 terminals on either side. Observe the relay closely to find the terminals.

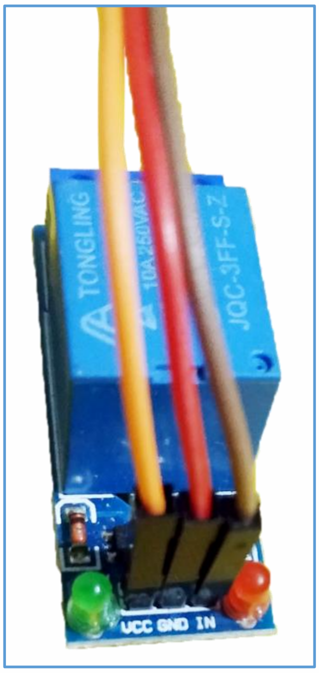
The side with VCC, GND, IN are the terminals for powering up the relay.

The side with NO, COM, NC are the terminals for connecting the switching devices like Fan, Light etc.

Step 6

Let’s complete the circuit for powering up the relay.

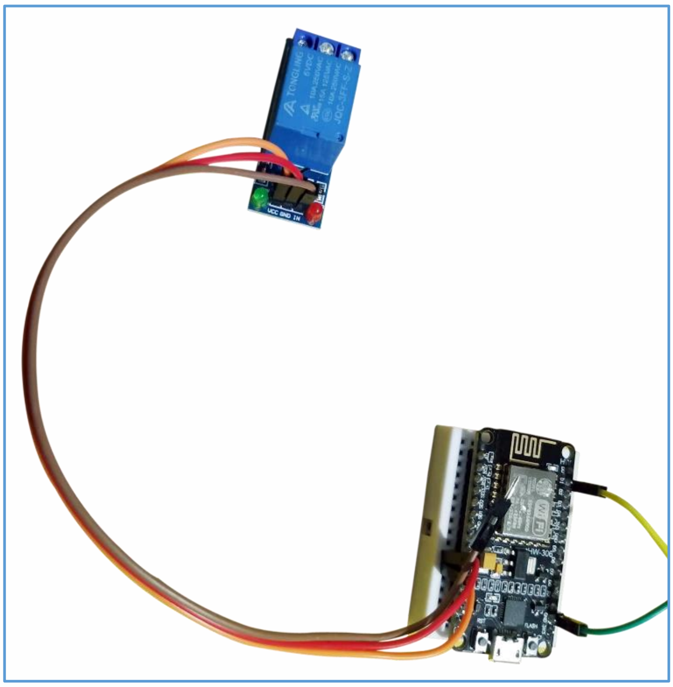
Connect the relay VCC to the Orange wire, then GND to the Red wire and IN to the Brown wire. We will use Male to female jumper wires as shown in the image given below.

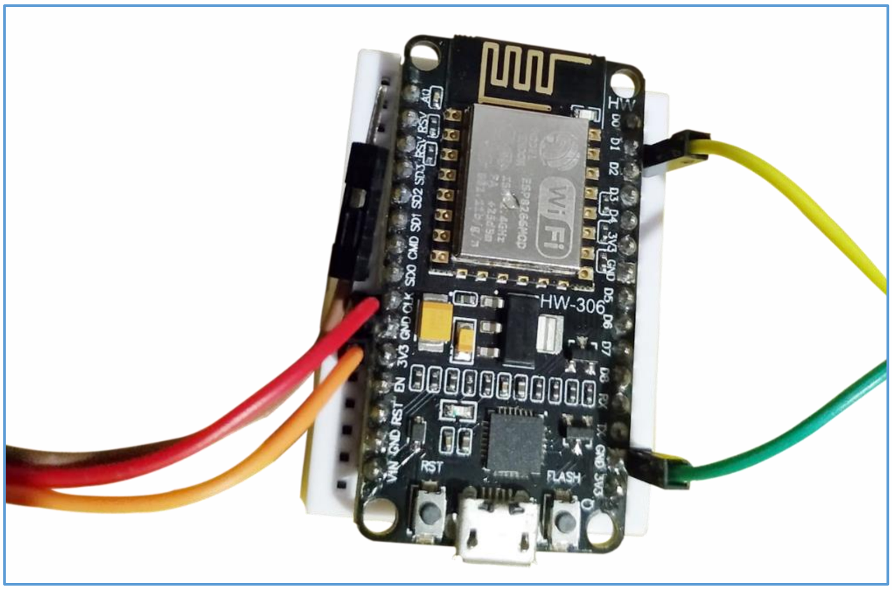


Step 7

Now, connect the Red (GND) wire of the relay to the Node MCU GND pin on the Reset button side.

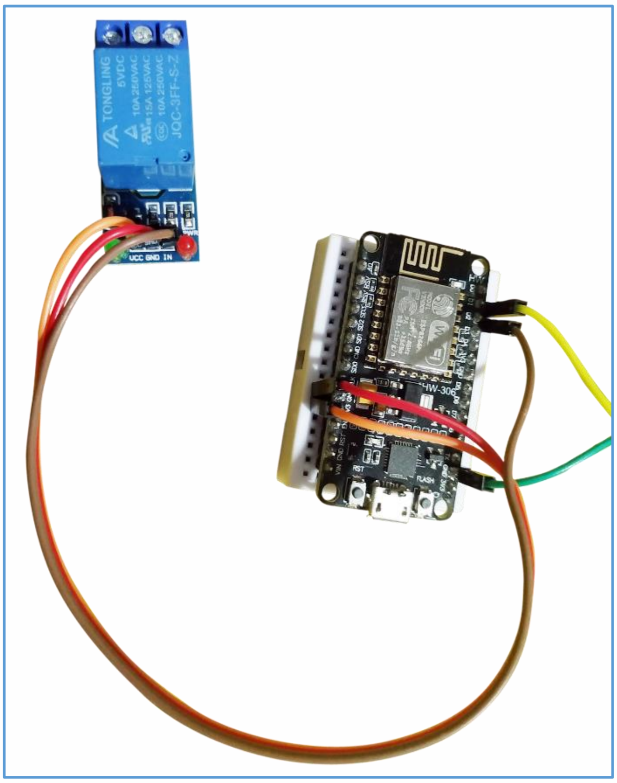
Then connect the Orange (VCC) wire of the relay to the Node MCU 3.3V pin which is beside the GND pin as shown below.

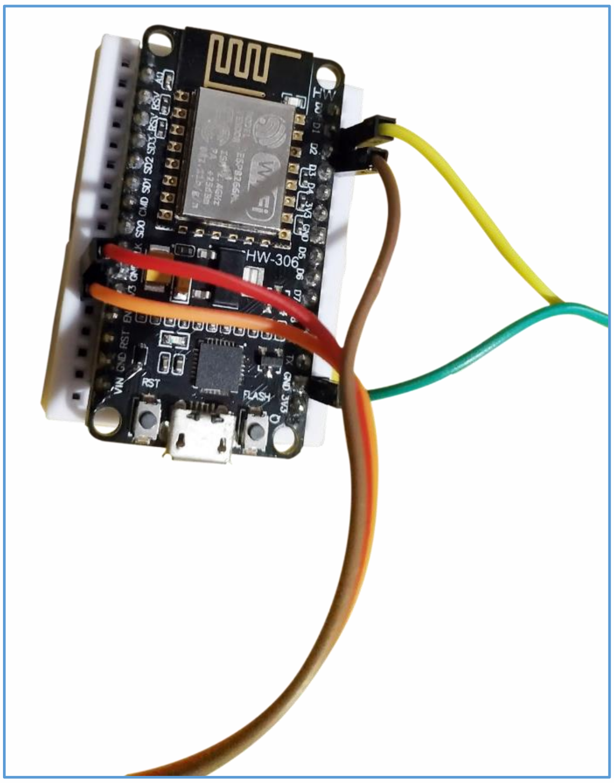




Step 8

Now let's connect the Brown wire (IN) of the relay to the Node MCU D2 pin which is present on the Flash button side, beside the D1 pin.

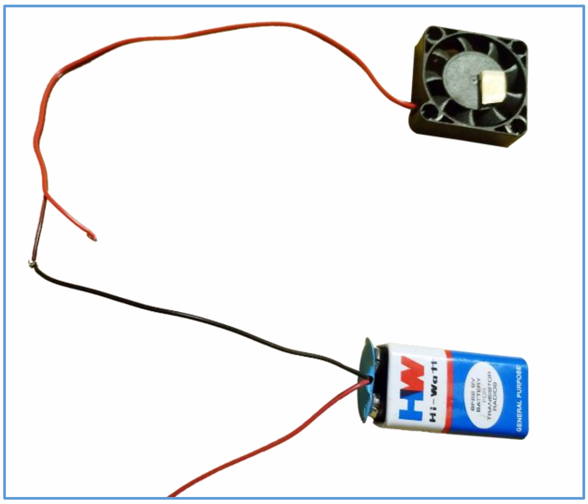




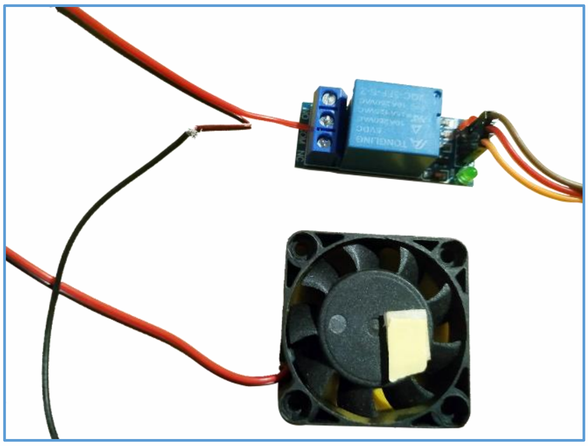
Step 9

Now let's connect the Fan to the other side of the relay. The Fan has two wires, Red and Black.

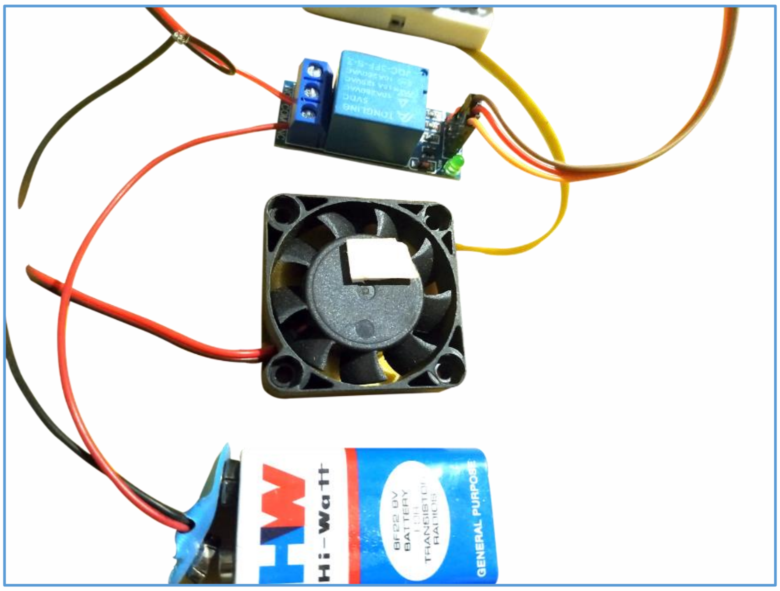
a. First, connect the black wire of the fan to the negative terminal of the 9V battery. To connect just bring the copper ends together and twist the wire.



b. Then, connect the red wire of the fan to the COM port of the relay. To connect simply unscrew the COM port, insert the wire and screw it back into the relay.



c. Connect the red wire (positive) terminal of the battery to the NC of the relay.

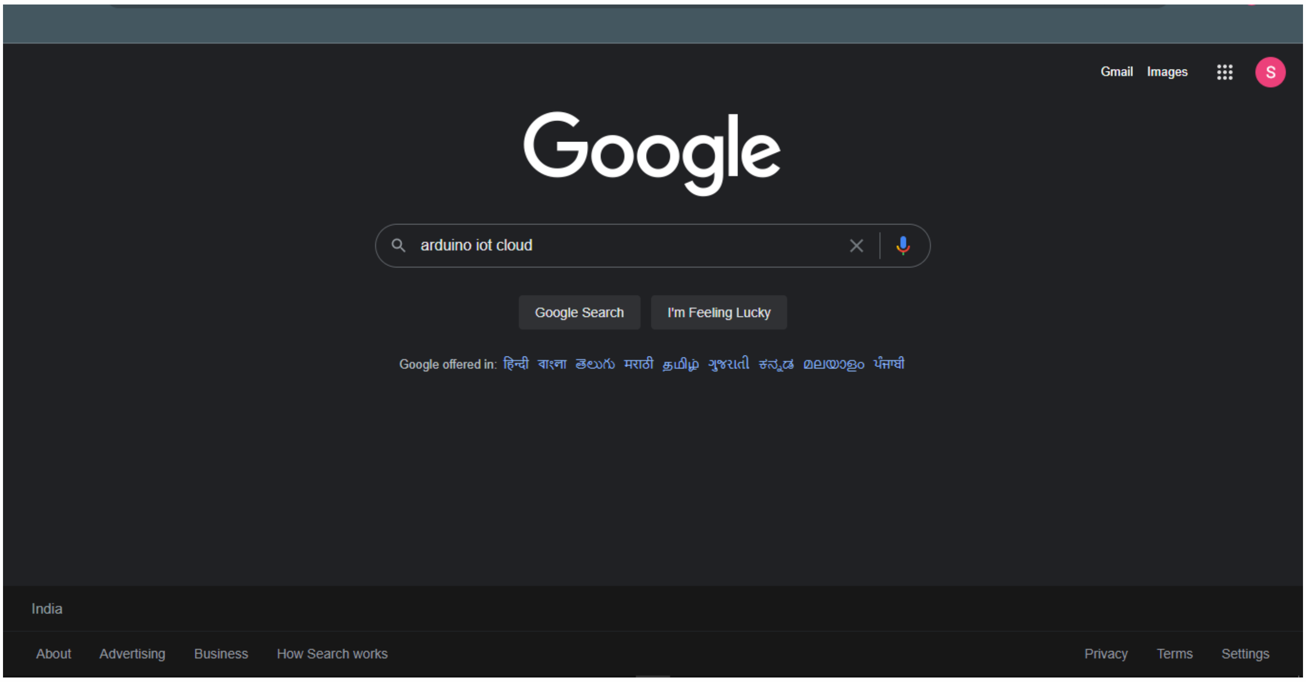


We have successfully completed the hardware connections of the circuit.

PART 2: Setting up the device

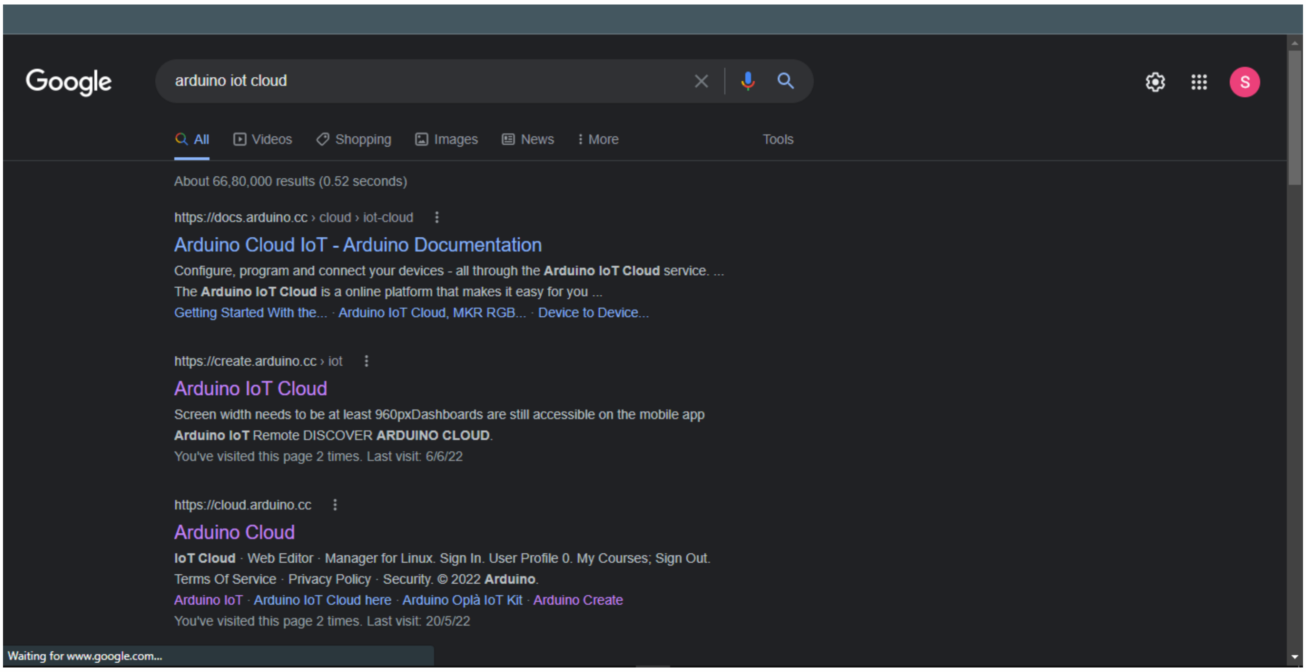
Step 10

Open any browser and then search for “Arduino IoT Cloud” in the google search bar.



arduino iot cloud

Click on the search result https://cloud.arduino.cc/ .



arduino iot cloud

About 66,80,000 results (0.52 seconds)

https://docs.arduino.cc › cloud › iot-cloud

Arduino Cloud IoT - Arduino Documentation

Configure, program and connect your devices - all through the **Arduino IoT Cloud** service. ...
The **Arduino IoT Cloud** is a online platform that makes it easy for you ...
Getting Started With the... · Arduino IoT Cloud, MKR RGB... · Device to Device...

https://create.arduino.cc › iot

Arduino IoT Cloud

Screen width needs to be at least 960pxDashboards are still accessible on the mobile app
Arduino IoT Remote DISCOVER **ARDUINO CLOUD**.
You've visited this page 2 times. Last visit: 6/6/22

https://cloud.arduino.cc

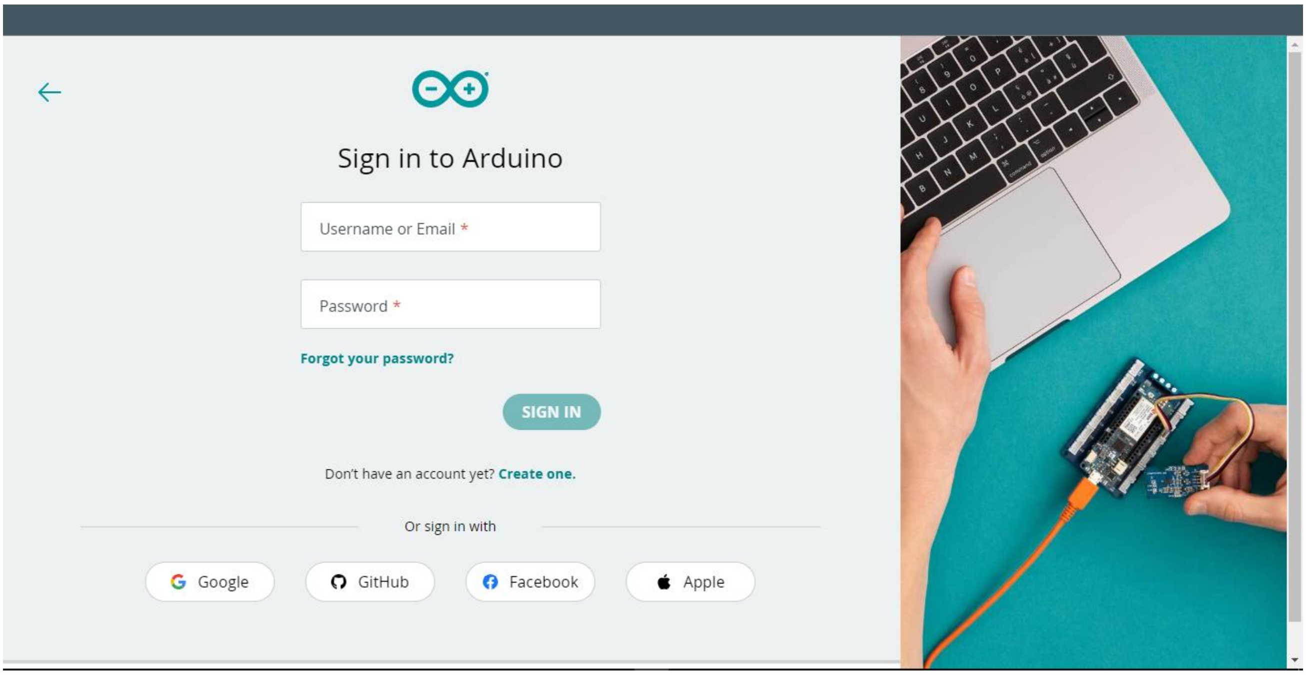
Arduino Cloud

IoT Cloud · Web Editor · Manager for Linux. Sign In. User Profile 0. My Courses; Sign Out.
Terms Of Service · Privacy Policy · Security. © 2022 **Arduino**.
Arduino IoT · Arduino IoT Cloud here · Arduino Oplà IoT Kit · Arduino Create
You've visited this page 2 times. Last visit: 20/5/22

Step 11

To sign-in, click on “Google” and sign in using your “Google Account”

Note: Seek the assistance of your teacher if you are not sure about the sign-in process.



Sign in to Arduino

Username or Email *

Password *

Forgot your password?

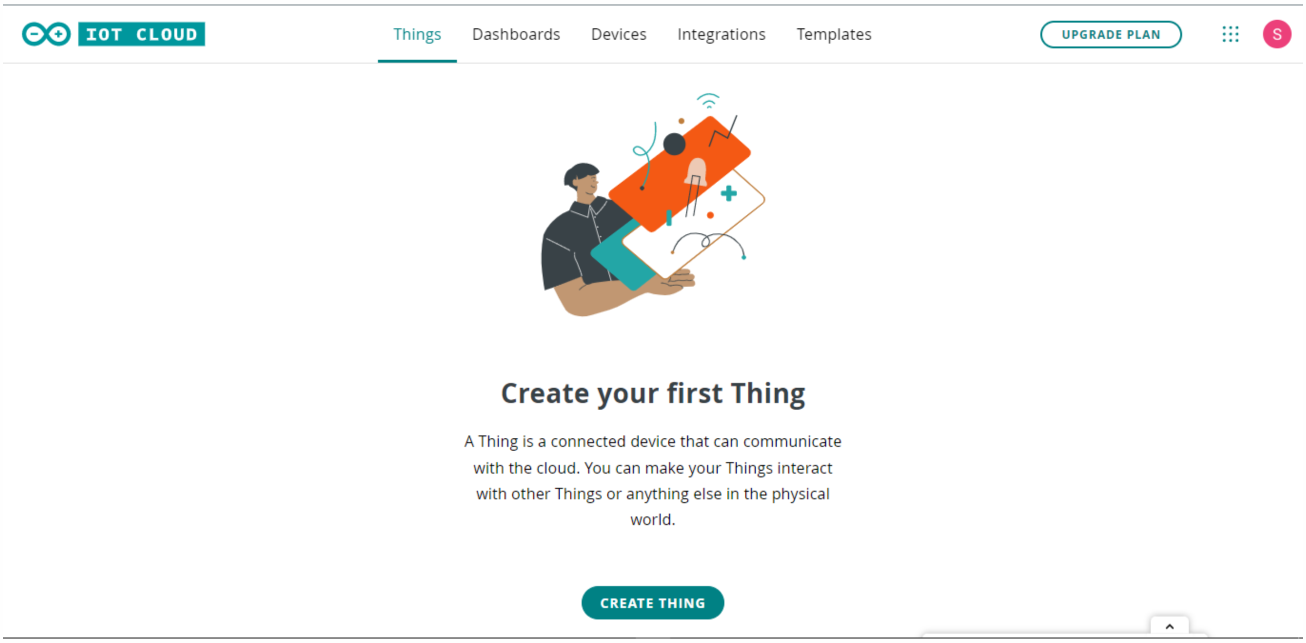
SIGN IN

Don't have an account yet? Create one.

Or sign in with

Step 12

After signing in, you will find, “Create Thing”. Click on the “Create Thing” button.



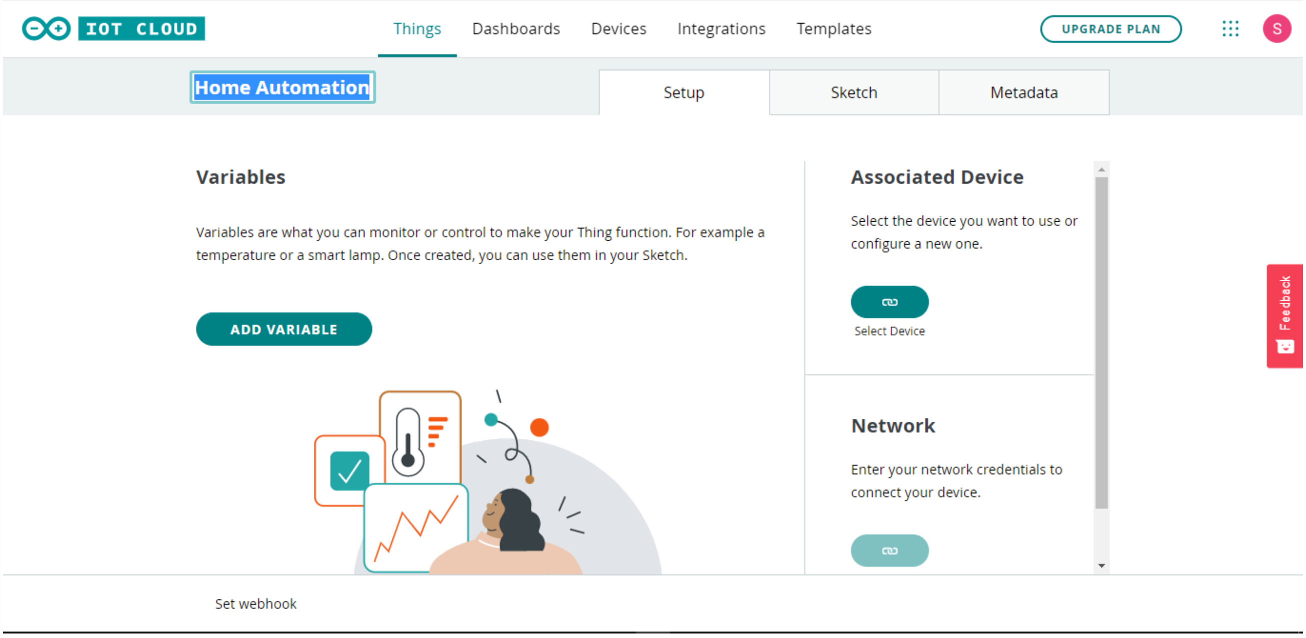
Create your first Thing

A Thing is a connected device that can communicate with the cloud. You can make your Things interact with other Things or anything else in the physical world.

CREATE THING

After clicking on you are redirected to “creating a thing” page.

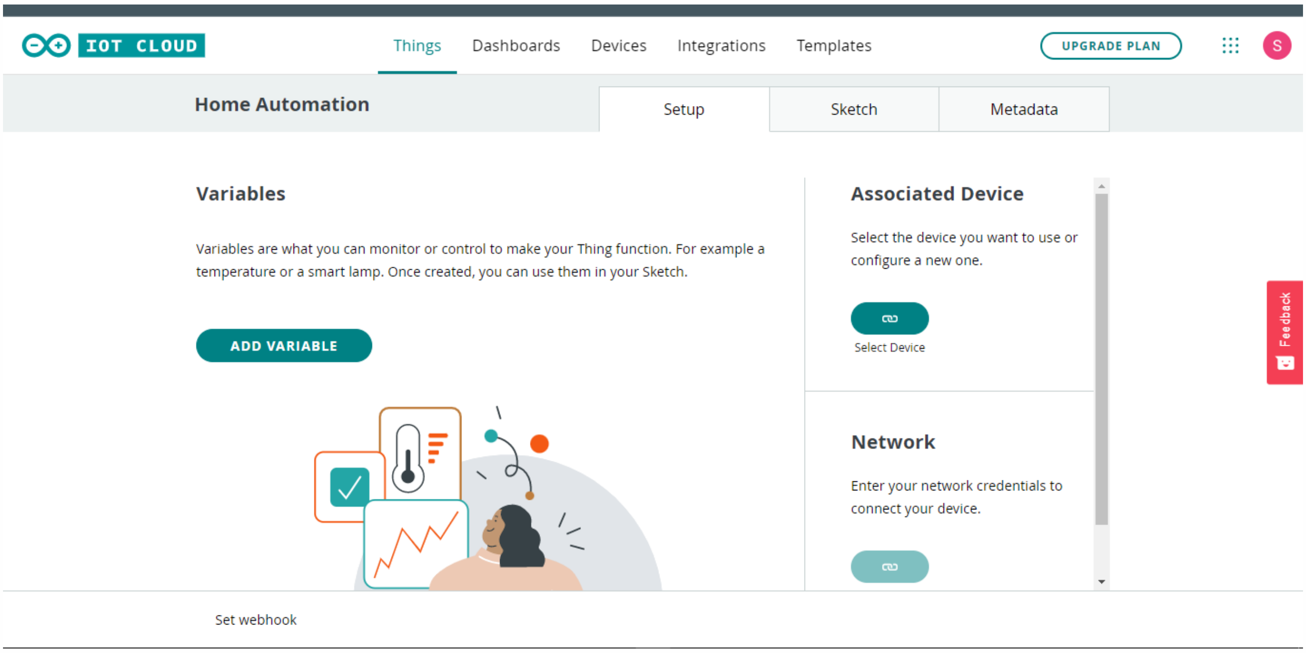
Let's first rename the thing. Click on the untitled and enter the name as “Home Automation”


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
S
Home Automation
Setup
Sketch
Metadata
Variables
Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.
ADD VARIABLE
Associated Device
Select the device you want to use or configure a new one.
Select Device
Network
Enter your network credentials to connect your device.
Feedback
Set webhook

Step 13

We need variables to store the data sent to the controller and also to get the data from the controller. So, let's create a variable for 1. Light and 2. Fan.

Click on the “Add Variable” button.


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
S
Home Automation
Setup
Sketch
Metadata
Variables
Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.
ADD VARIABLE
Associated Device
Select the device you want to use or configure a new one.
Select Device
Network
Enter your network credentials to connect your device.
Feedback
Set webhook

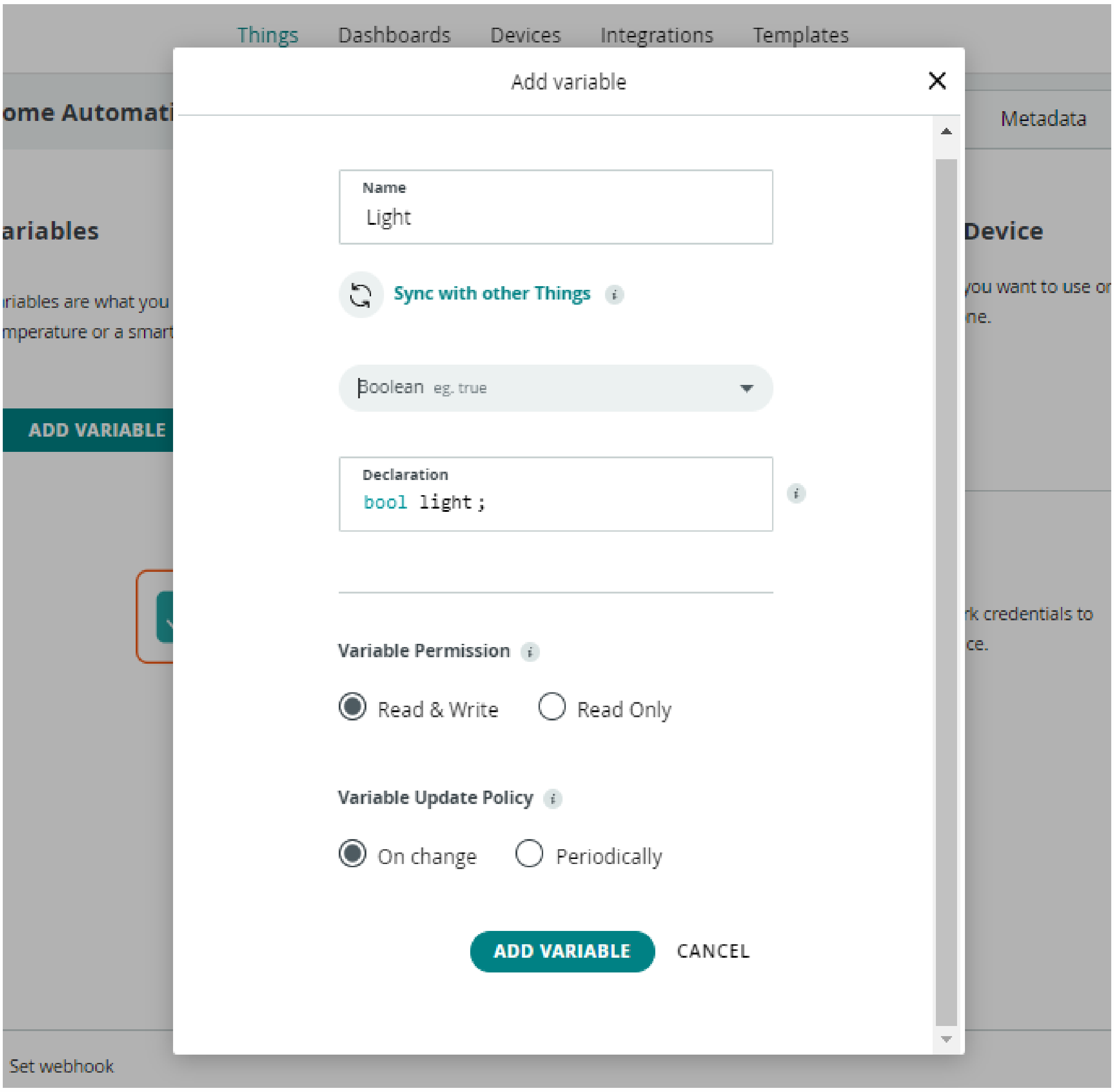
Step 14

A popup will appear asking for the details of the variable.

a. Name the variable as “Light”.

b. Type of the variable is “Boolean” because a switch can only turn on or turn off.

c. Keep the other details unchanged. Then click on the “Add Variable” button.



Name
Light

Boolean eg. true

Declaration
```
bool light;
```

Variable Permission

● Read & Write ○ Read Only

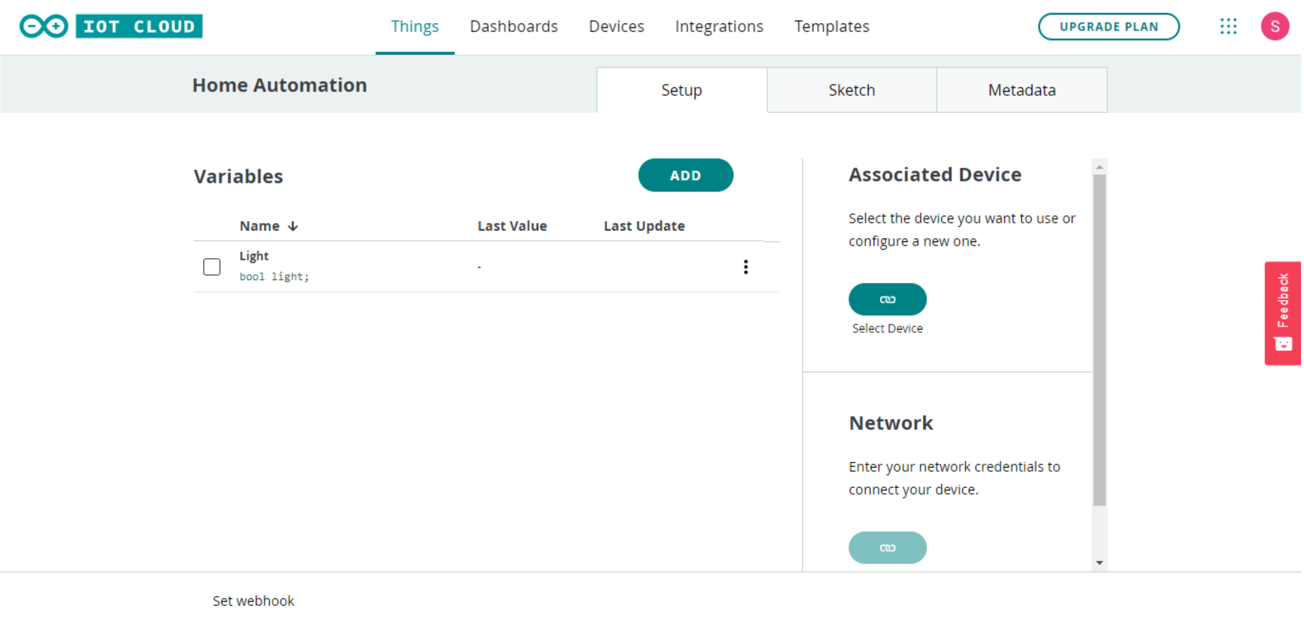
Variable Update Policy

● On change ○ Periodically

Step 15

Now let’s create one more variable for the “Fan” switch.

Click on the “Add” button.

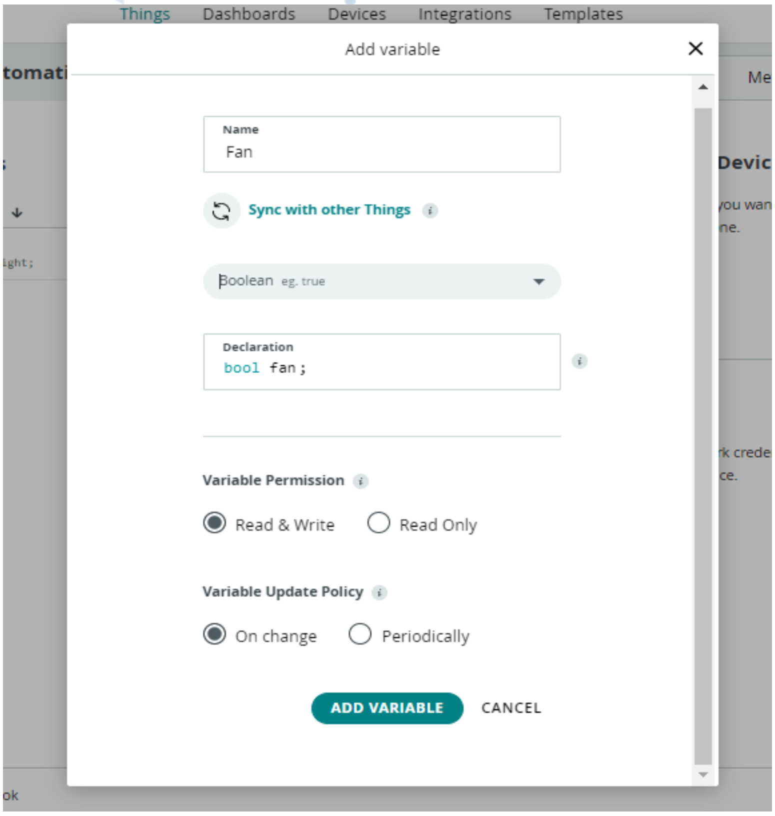


Step 16

Now let's add the details of the fan variable.

1. Name the variable as “Fan”
2. Type of the variable is “Boolean” again because the fan switch can only turn on or turn off.
3. Variable Permission as “Read/Write”
4. Variable update policy as “On change” because when the button is pressed the data has to be changed.

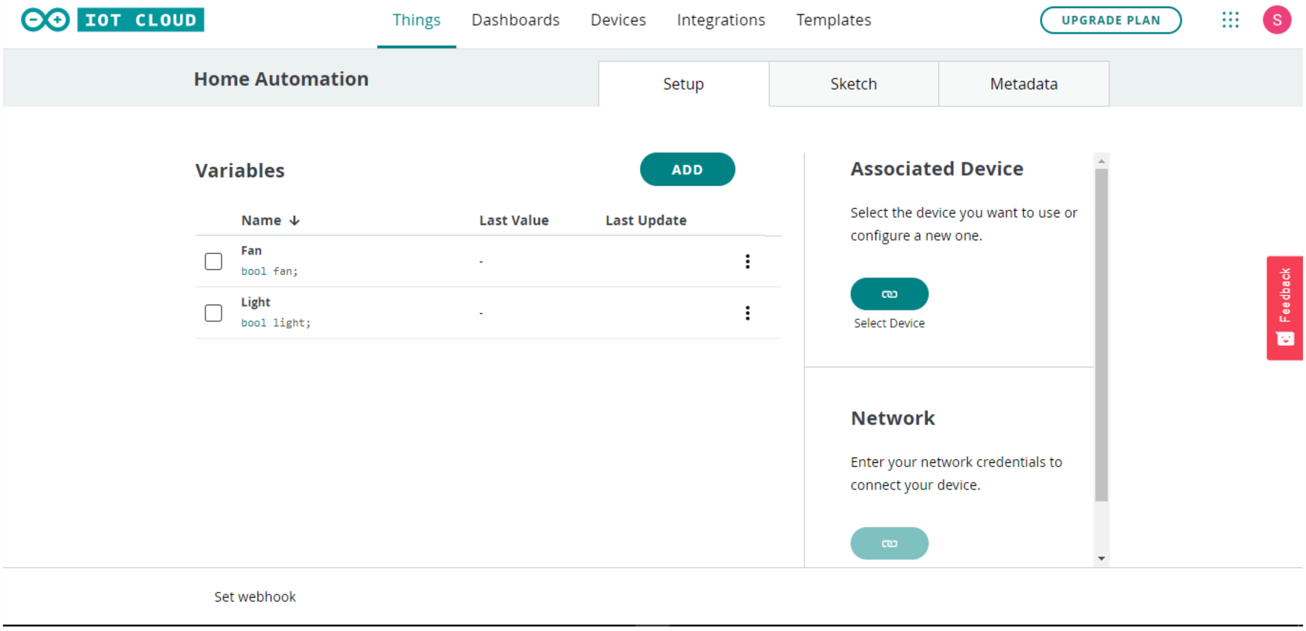
Now click on “Add Variable”



Step 17

Now let’s link the circuit as a device.

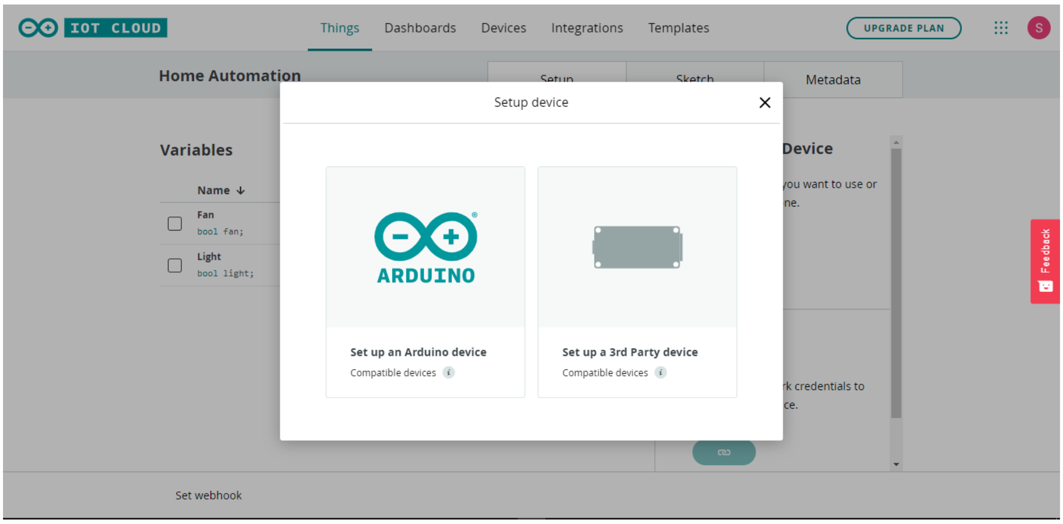
Click on the “Select device” button then a popup will open.



Step 18

Now in the popup, we have to select the type of board. Here we are using “Node MCU” which is a 3rd party device.

So click on "Set up a 3rd Party device".

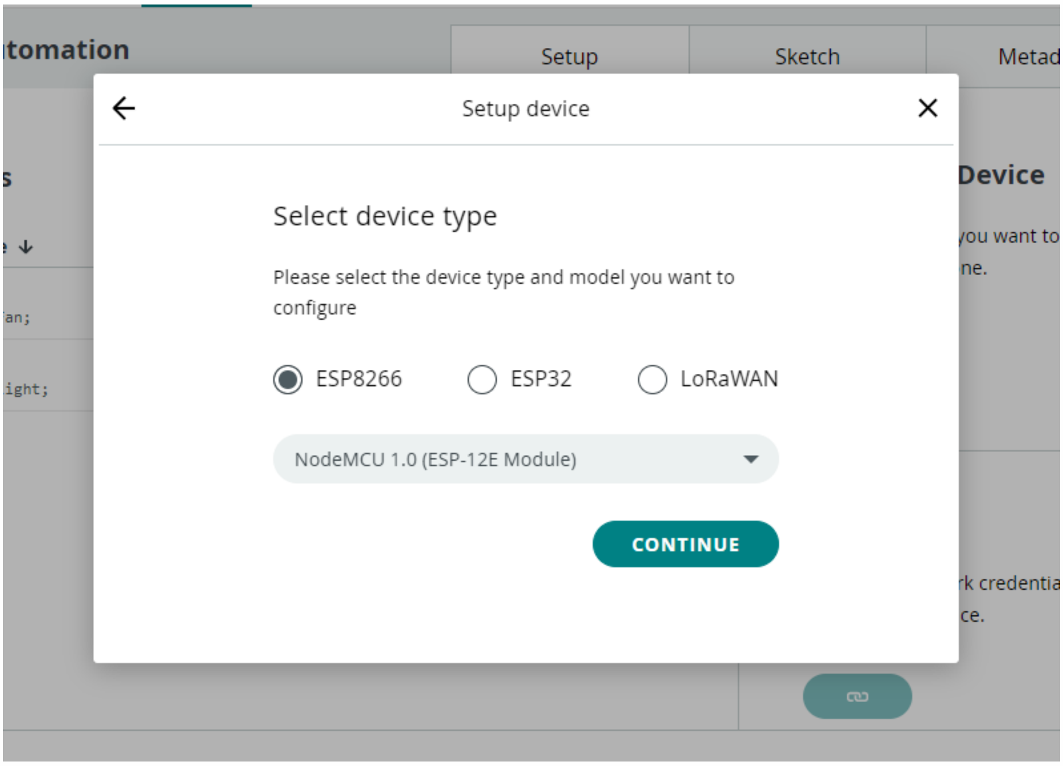


Setup device

Set up an Arduino device
Compatible devices

Set up a 3rd Party device
Compatible devices

In the dropdown search for Node MCU 1.0 board and click on "Continue."



Setup device

Select device type

Please select the device type and model you want to configure

● ESP8266 ○ ESP32 ○ LoRaWAN

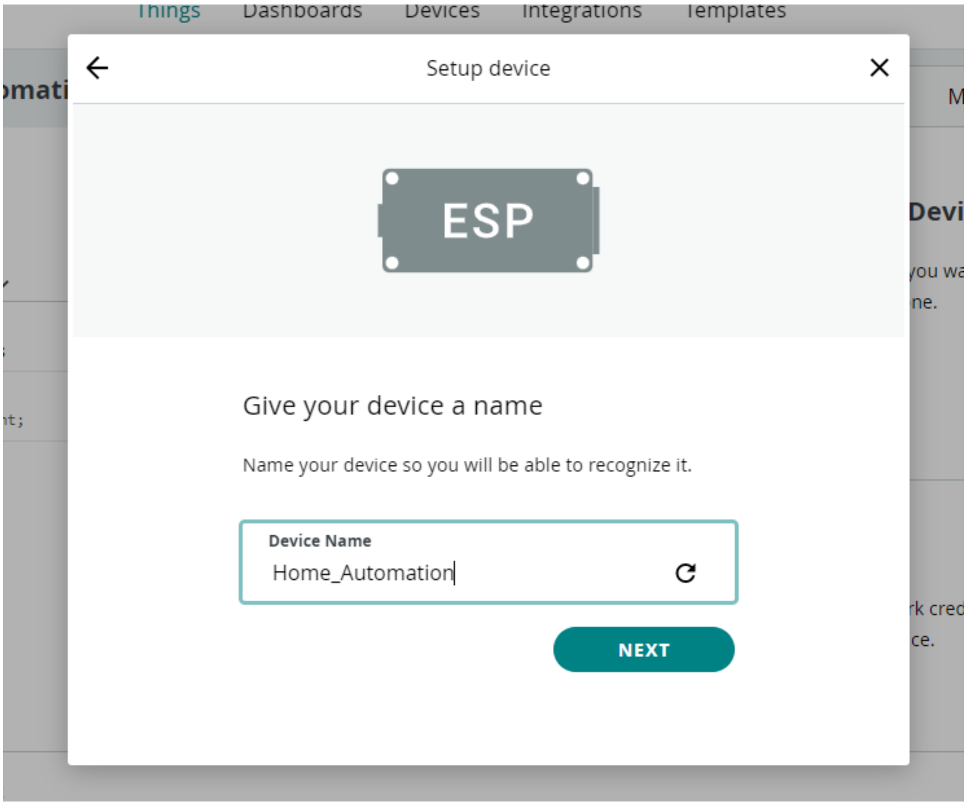
NodeMCU 1.0 (ESP-12E Module)

CONTINUE

Step 19

Now let's name the device.

a. A random name is assigned to the device. Delete the name and enter the name as "Home_Automation" and then click on the "Next" button.



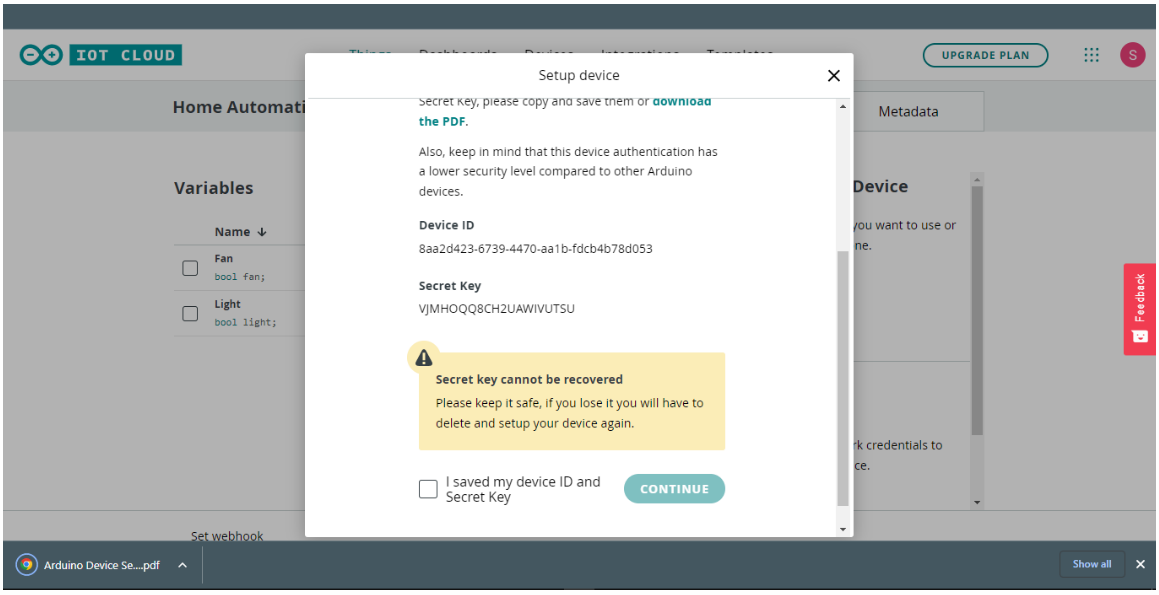
Setup device

Give your device a name

Name your device so you will be able to recognize it.

Device Name
Home_Automation

b. After clicking on "Next" a new page will open with "Device ID" and "Security Key". These two details are very important, so click on the "download pdf" and save the file.



Setup device

Secret Key, please copy and save them or download the PDF.

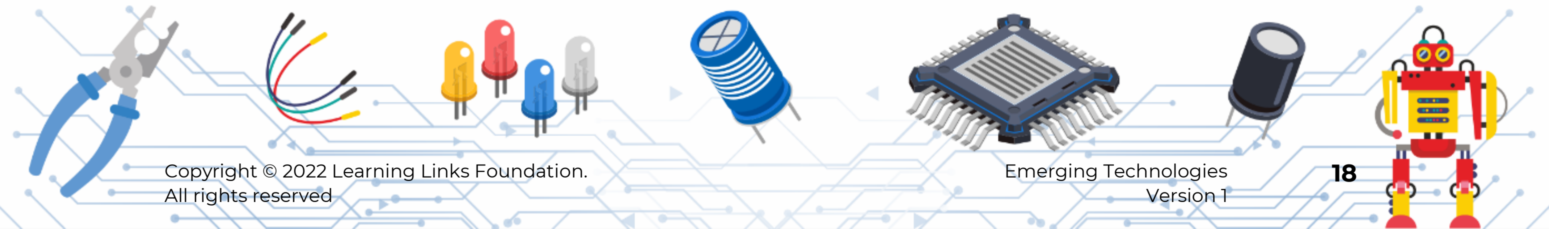
Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID
8aa2d423-6739-4470-aa1b-fdcb4b78d053

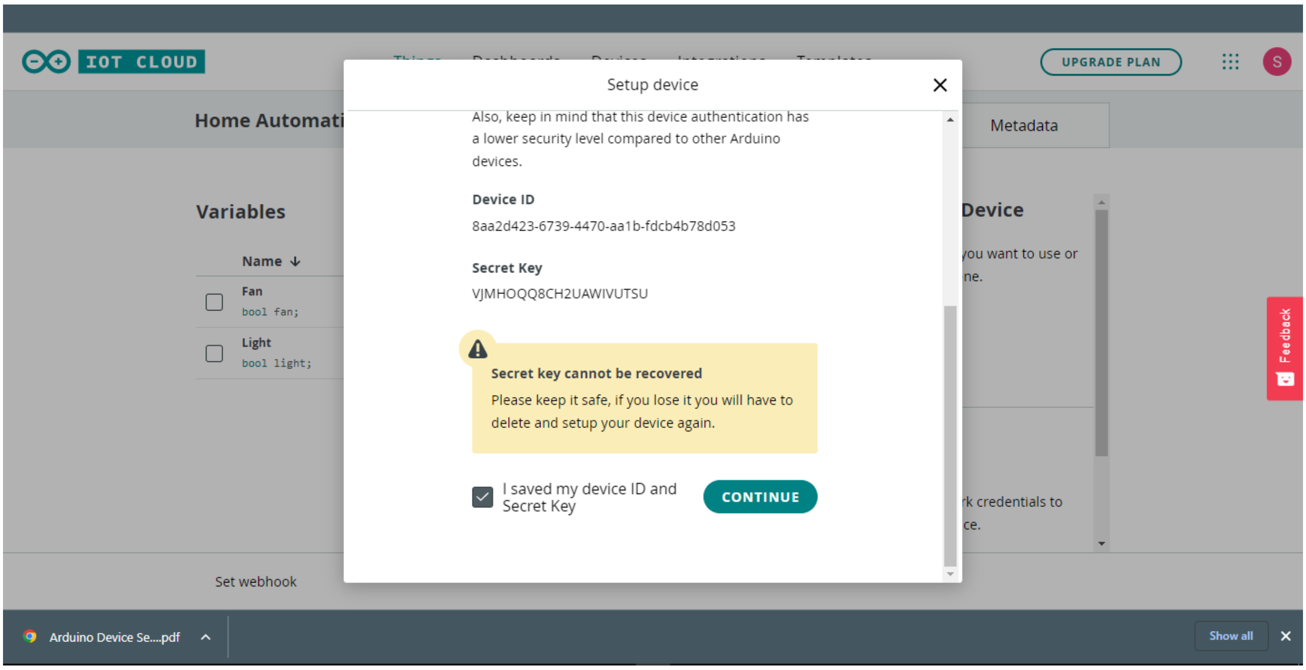
Secret Key
VJMHOQQ8CH2UAWIVUTSU

Secret key cannot be recovered
Please keep it safe, if you lose it you will have to delete and setup your device again.

□ I saved my device ID and Secret Key



c. Then check the "I saved my device ID and Secret key" check box and then click on "Continue".



Setup device

Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID
8aa2d423-6739-4470-aa1b-fdcb4b78d053

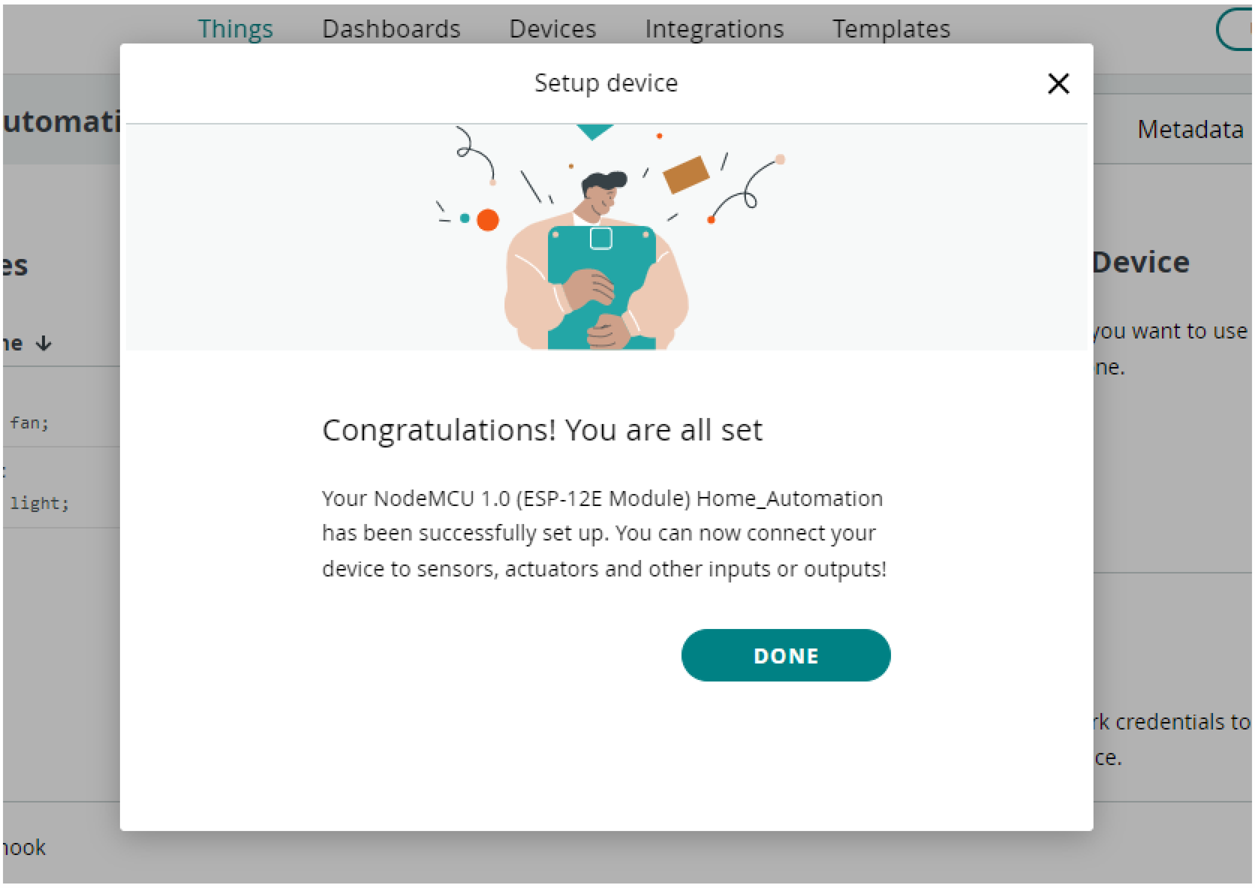
Secret Key
VJMHOQQ8CH2UAWIVUTSU

Secret key cannot be recovered
Please keep it safe, if you lose it you will have to delete and setup your device again.

☑ I saved my device ID and Secret Key

CONTINUE

Congratulations! you device set-up is successful.



Setup device

Congratulations! You are all set

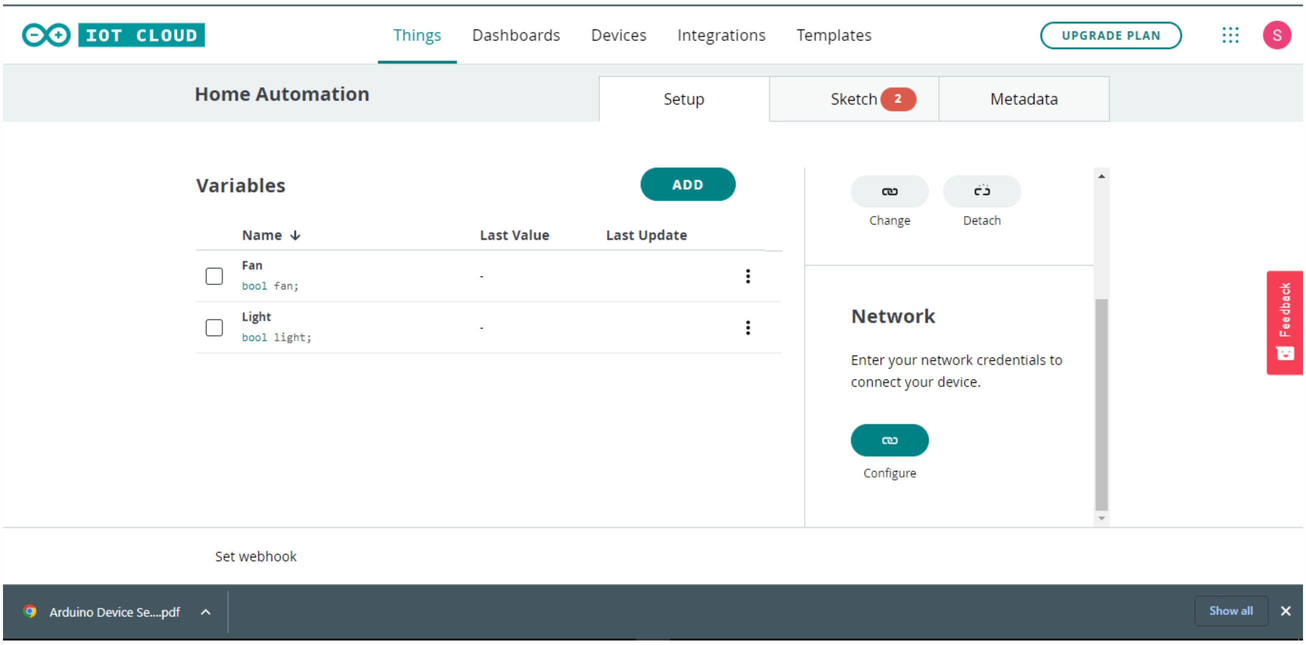
Your NodeMCU 1.0 (ESP-12E Module) Home_Automation has been successfully set up. You can now connect your device to sensors, actuators and other inputs or outputs!

DONE

PART 3: Setting up the Network & Building the Dashboard

Step 20

Click on the “Configure” button then a popup will appear.



	Name	Last Value	Last Update
□	Fan bool fan;	-	
□	Light bool light;	-	

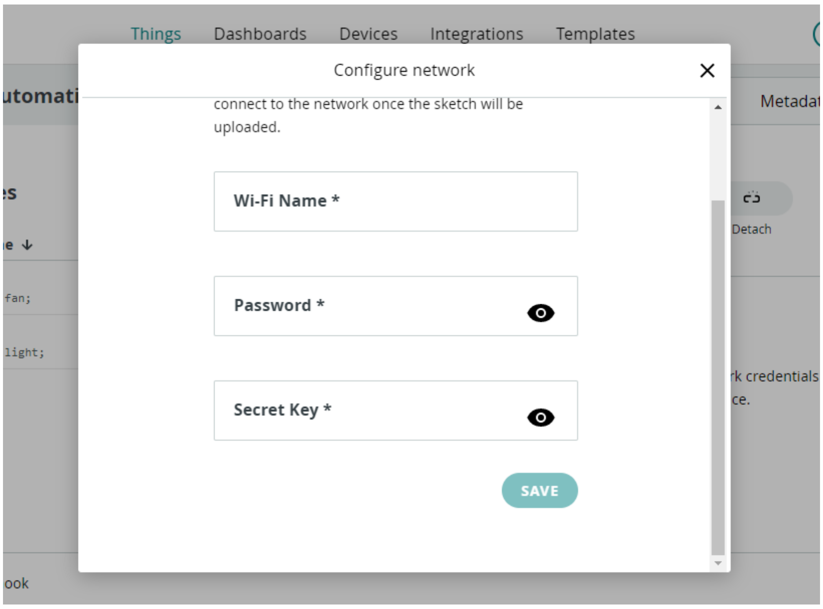
Enter your network credentials to connect your device.

Enter the details of the network.

Wifi Name–

Password–

Secret Key– (Available in the pdf downloaded)



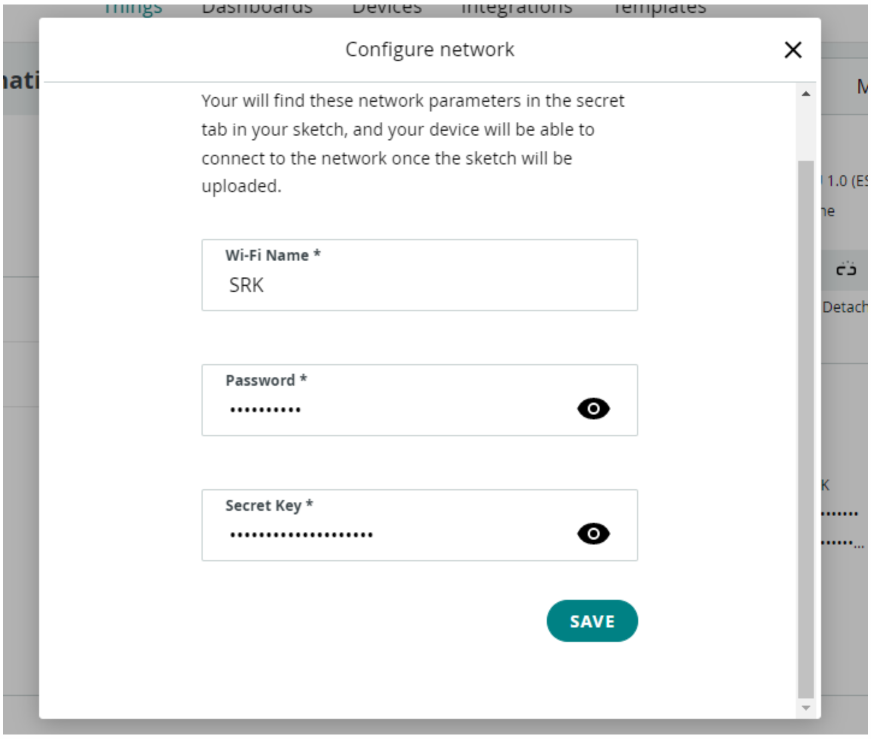
connect to the network once the sketch will be uploaded.

Wi-Fi Name *

Password *

Secret Key *

After entering all the details click on the “Save” button.



Configure network

Your will find these network parameters in the secret tab in your sketch, and your device will be able to connect to the network once the sketch will be uploaded.

Wi-Fi Name * SRK

Password * ••••••••••

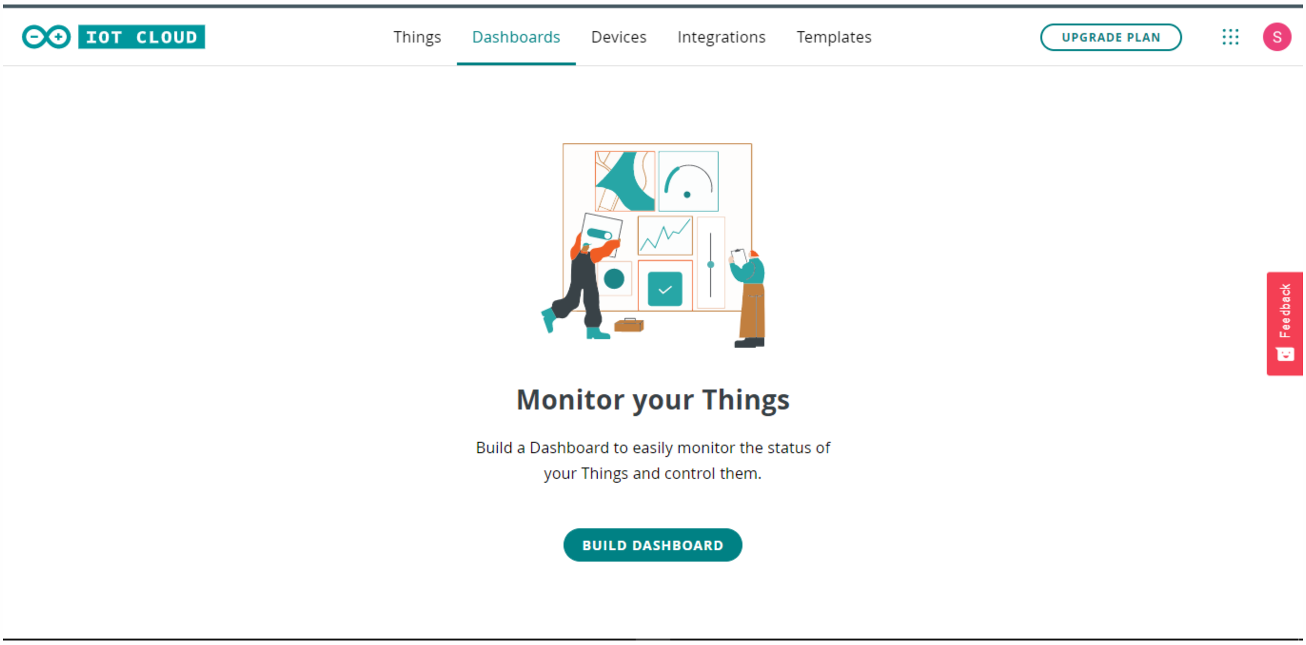
Secret Key * ••••••••••••••••••••

Step 21

Building the Dashboard

Now let’s build the dashboard. Click on the “Dashboards” button beside the “Things” button.

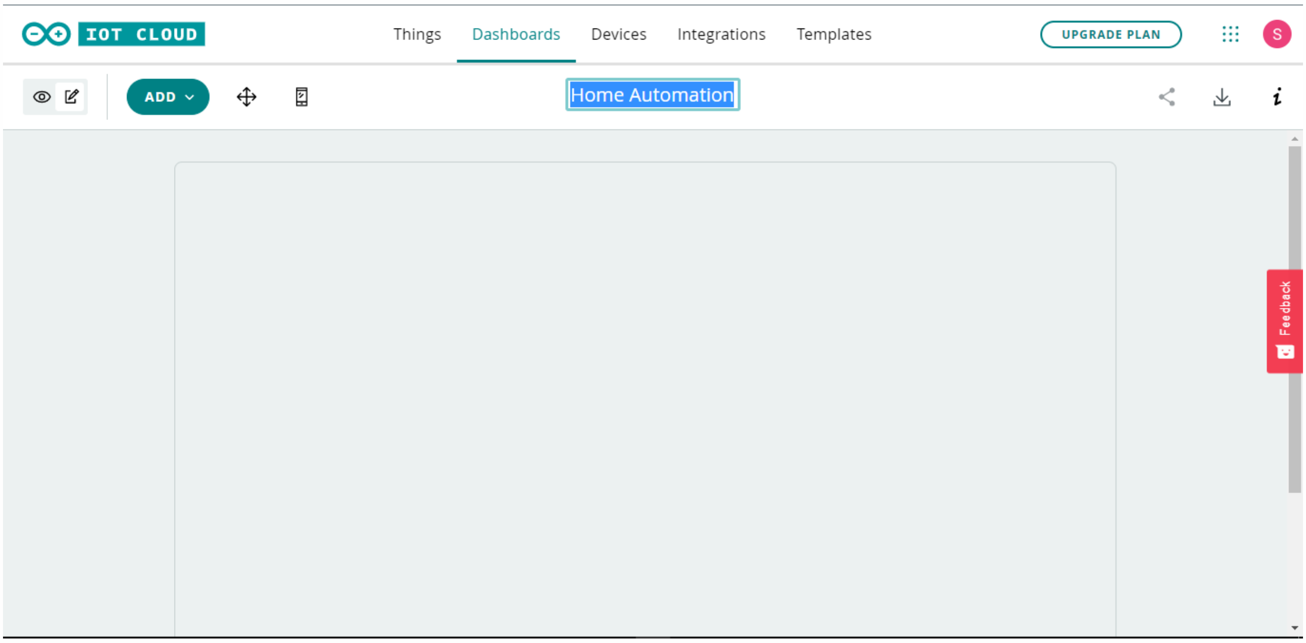
After clicking on Dashboard the page will look like this.



Monitor your Things

Build a Dashboard to easily monitor the status of your Things and control them.

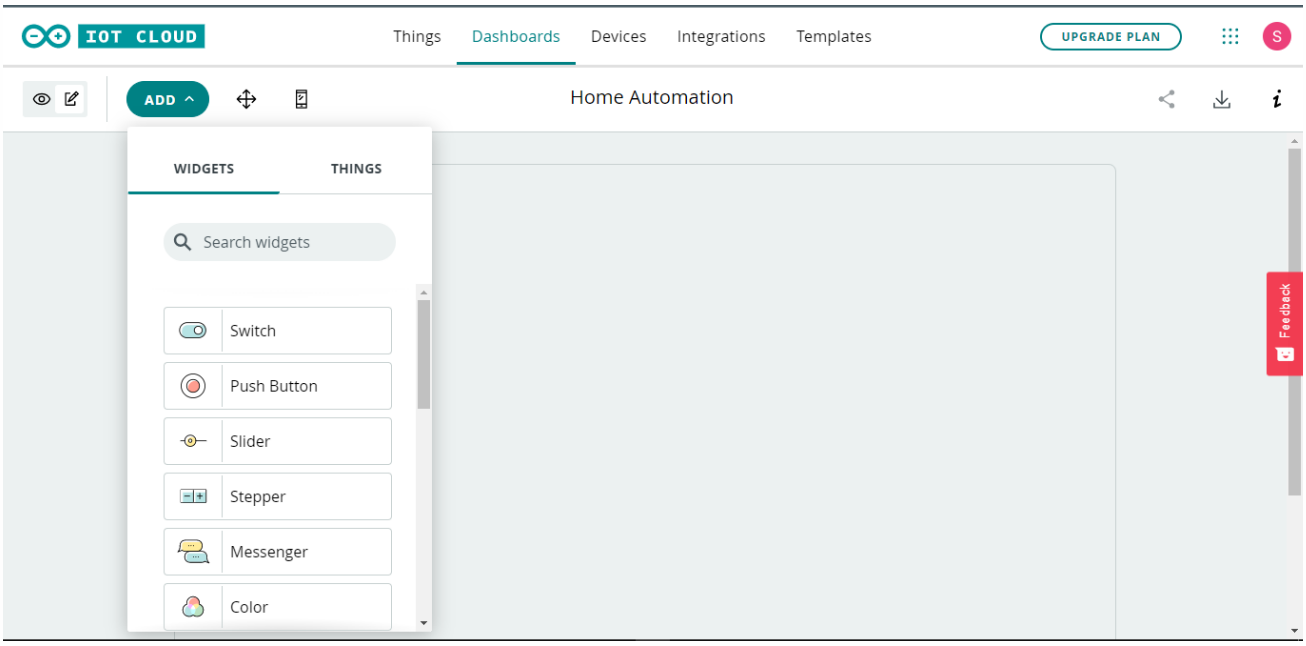
a. Click on "Build Dashboard" button. Then a new page will appear. Click on the untitled Dashboard name and enter the name as "Home Automation"



Home Automation

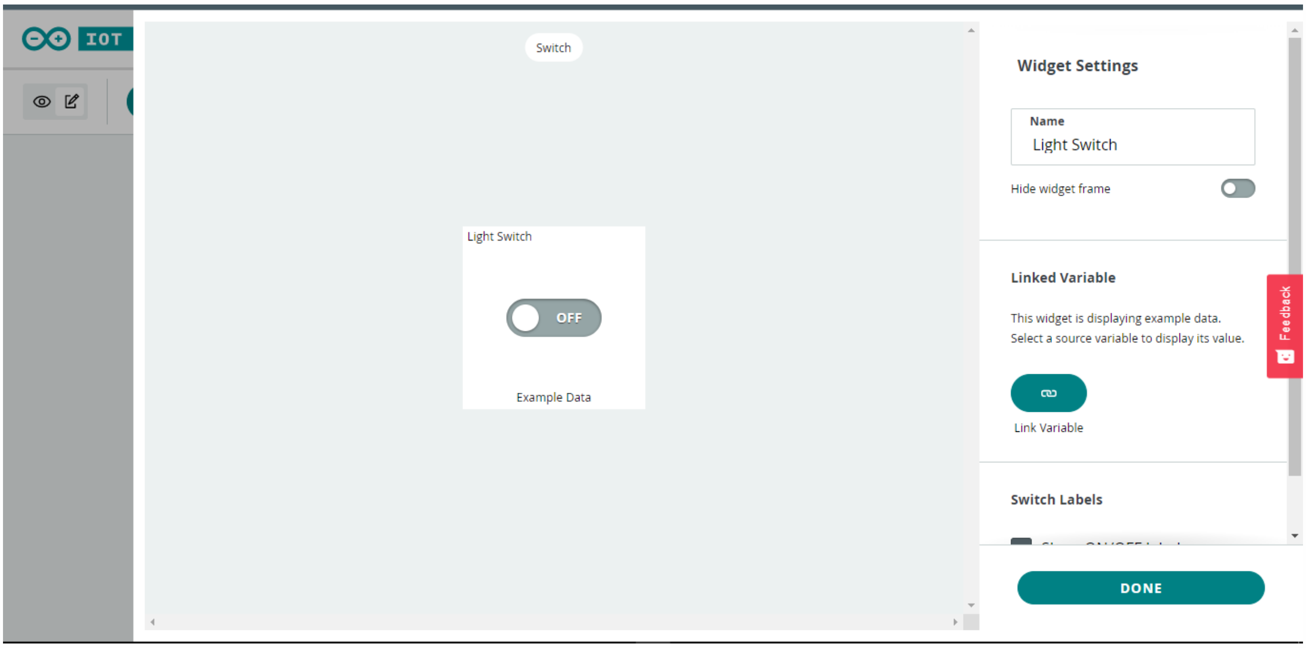
b. Now let's add a switch in the dashboard.

We need a switch to turn on/turn off the Light. Click on the "Add" dropdown and select the switch option from the drop-down list.



Home Automation

c. Name the switch as “Light Switch”. Then click on “Link Variable”



Widget Settings

Name: Light Switch

Hide widget frame

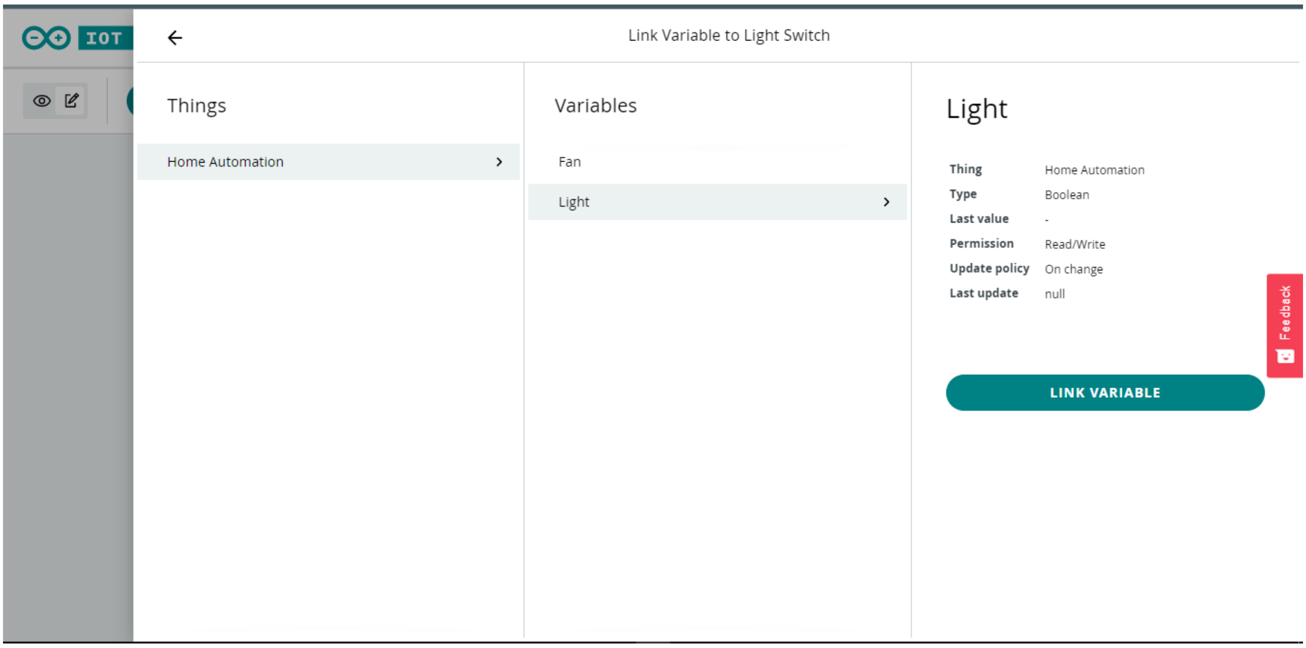
Linked Variable

This widget is displaying example data. Select a source variable to display its value.

Link Variable

Switch Labels

d. After clicking on Link Variable a new page will appear. Select the Light variable from the list and then click on the “Link Variable”.



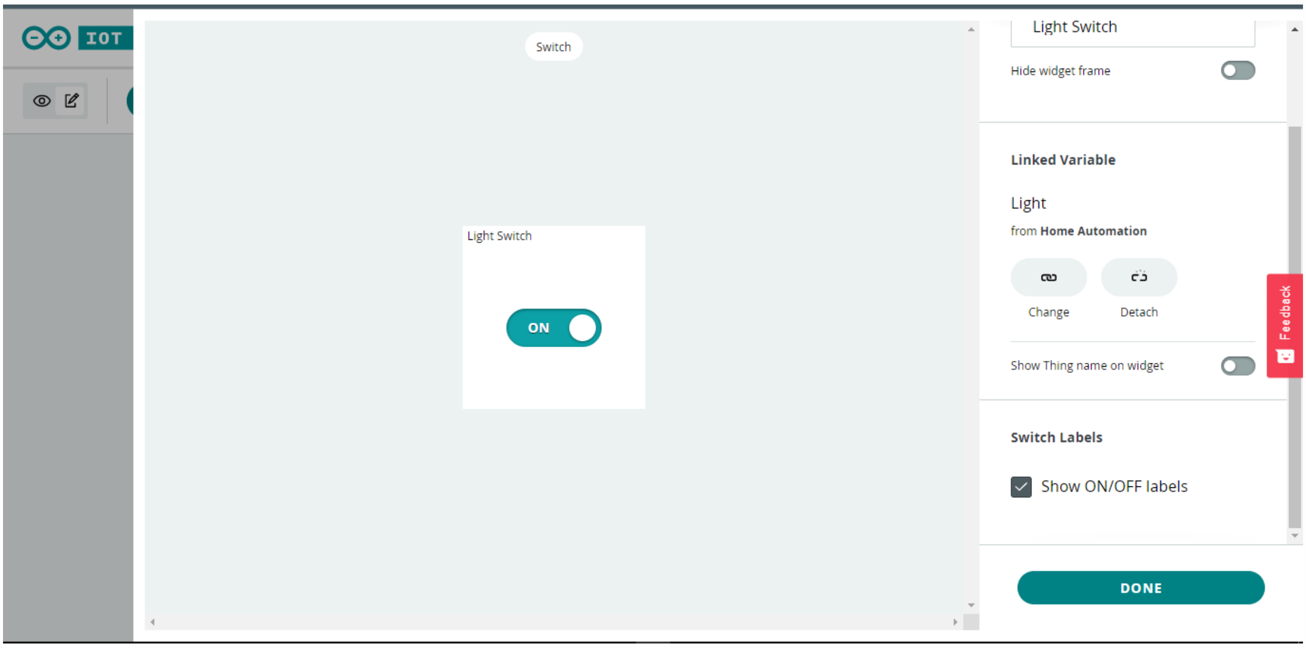
Link Variable to Light Switch

Things	Variables
Home Automation	Fan
	Light

Light

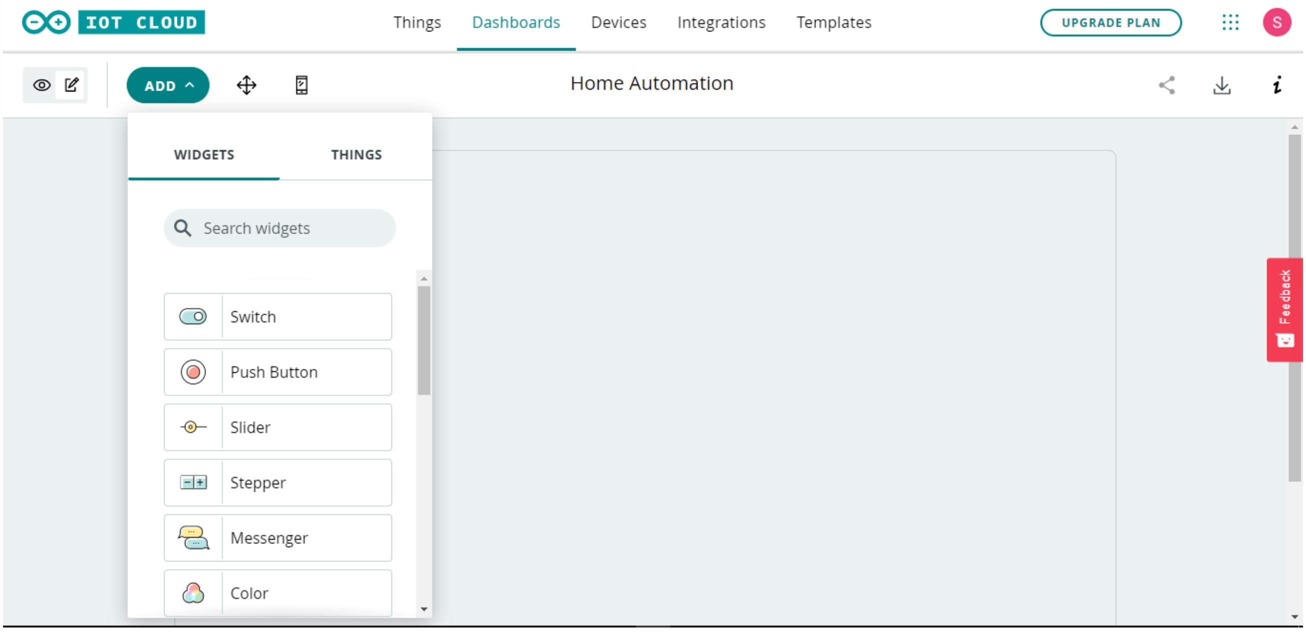
Thing	Home Automation
Type	Boolean
Last value	-
Permission	Read/Write
Update policy	On change
Last update	null

e. Now click on the “Done”.

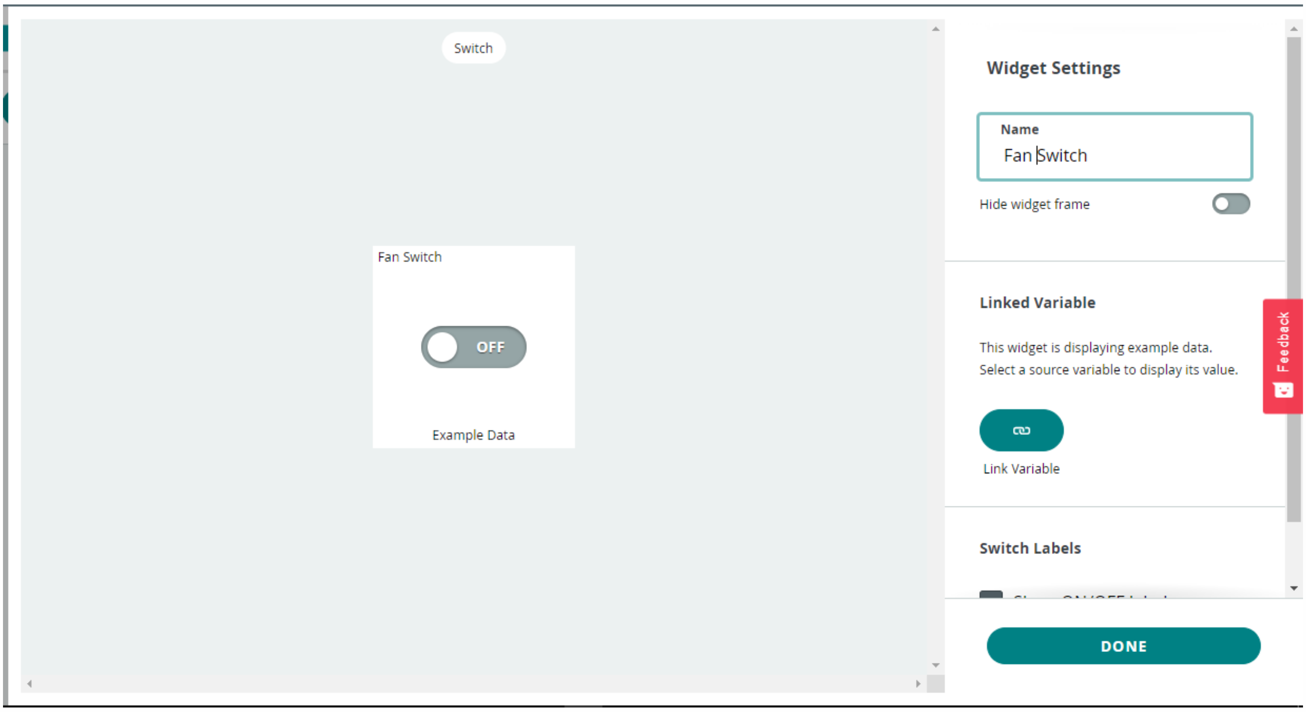


Step 22

Now, we will create one more switch to control the fan. Click on the “Add” button and select the switch option from the drop-down list.



a. Name the Switch as "Fan Switch" and then click on "Link Variable" button.



Widget Settings

Name: Fan Switch

Hide widget frame

Linked Variable

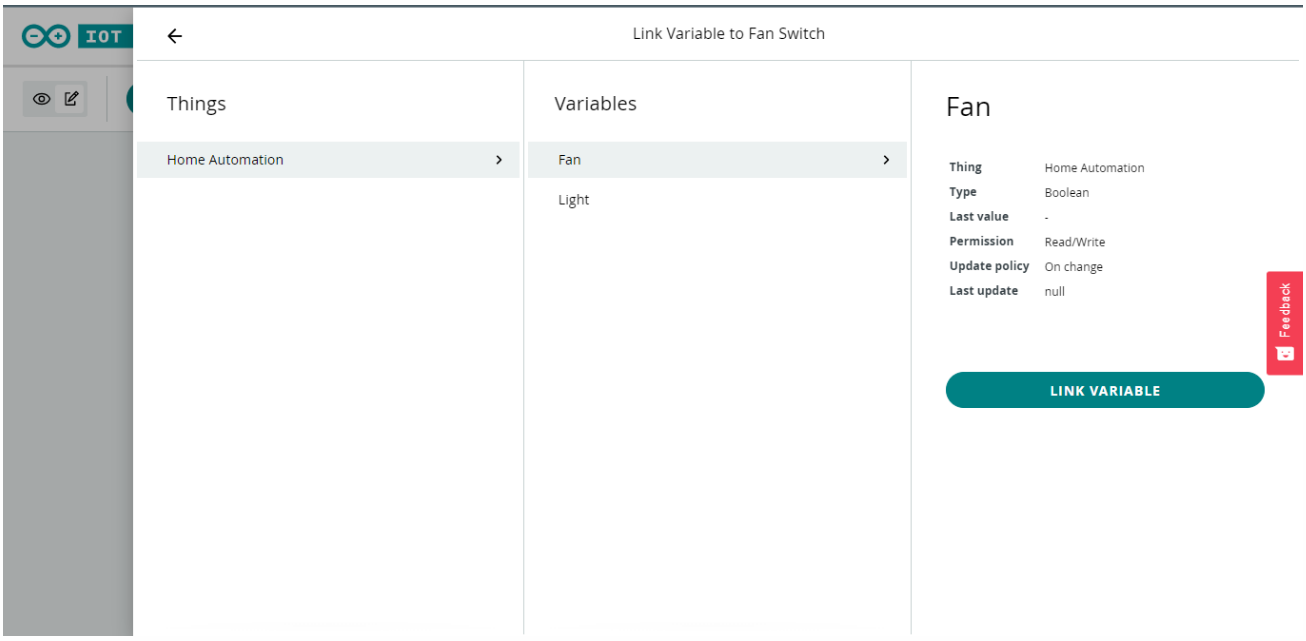
This widget is displaying example data. Select a source variable to display its value.

Link Variable

Switch Labels

DONE

b. Now choose the fan variable from the list and then click on the "Link Variable"



Link Variable to Fan Switch

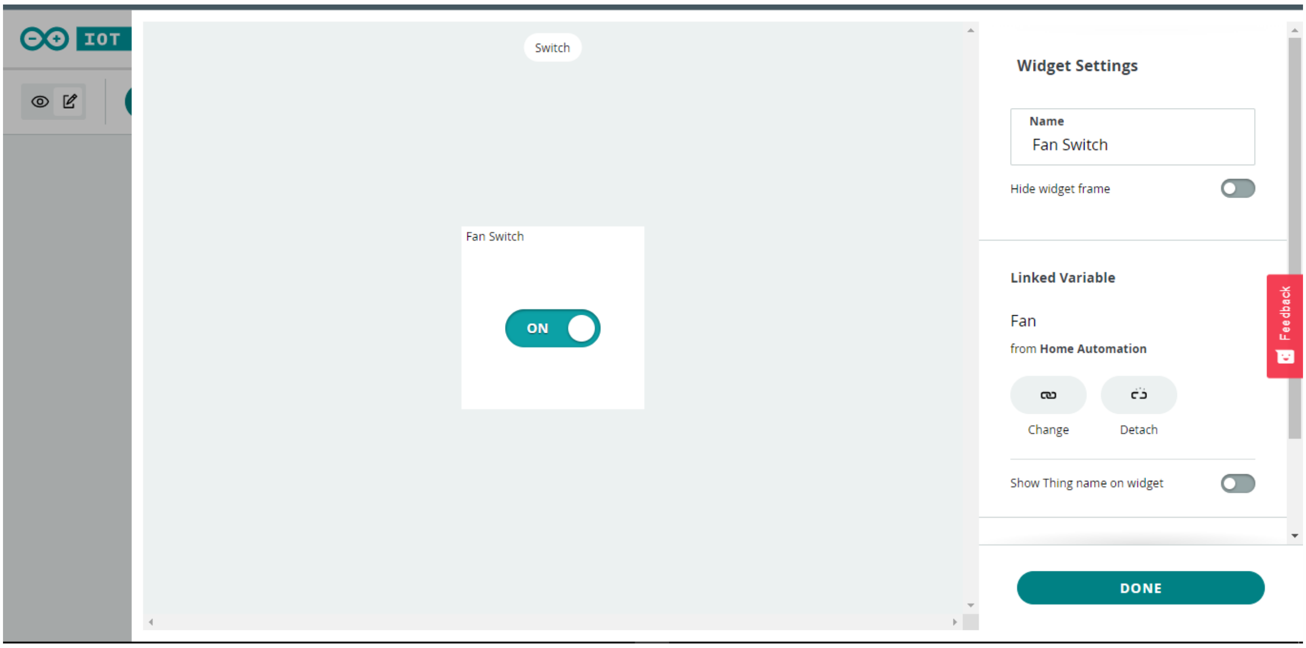
Things	Variables
Home Automation	Fan
	Light

Fan

Thing	Home Automation
Type	Boolean
Last value	-
Permission	Read/Write
Update policy	On change
Last update	null

LINK VARIABLE

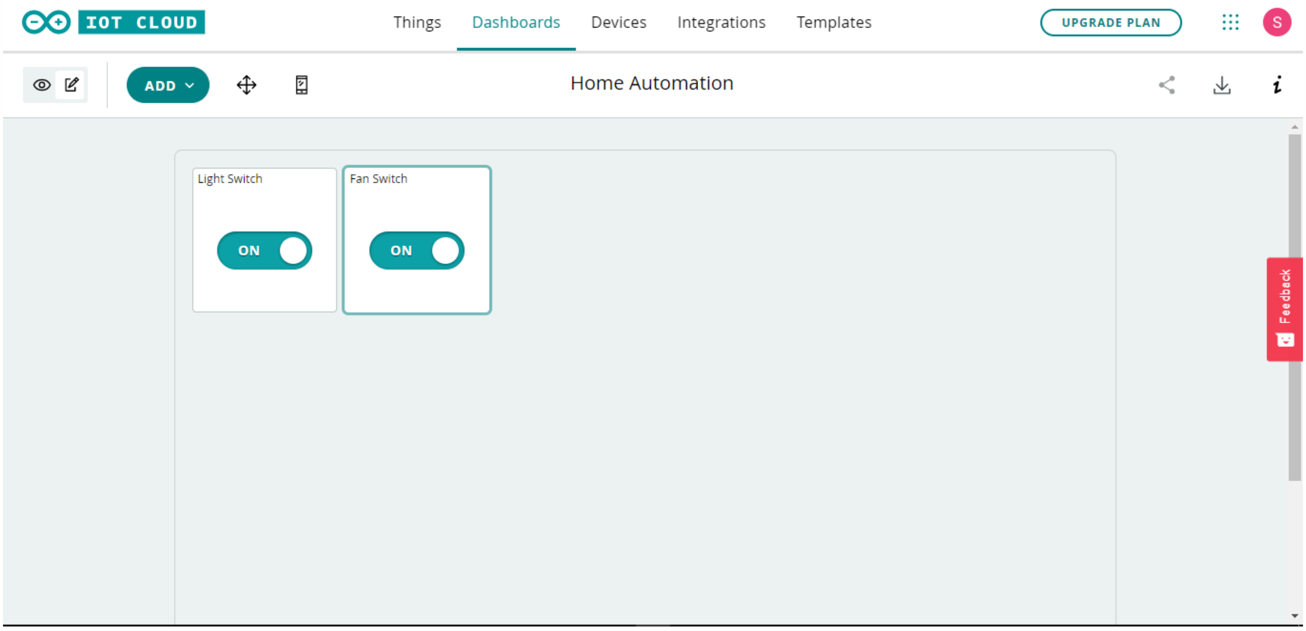
c. Then, click on "Done".



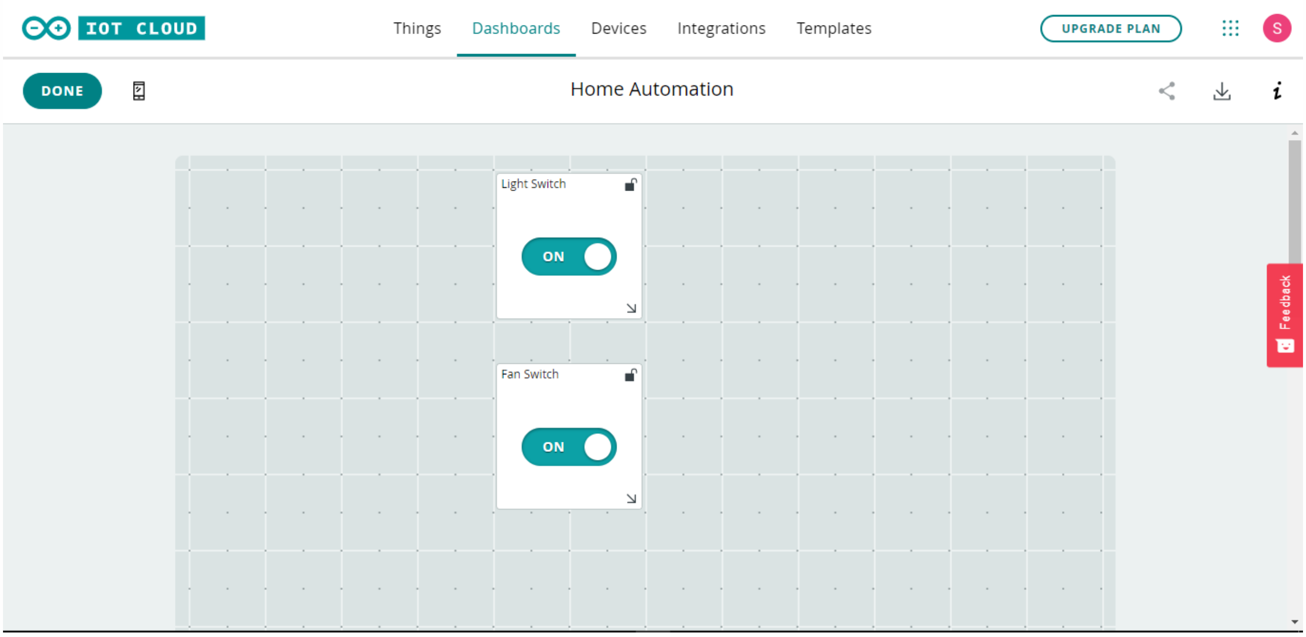
Step 23

Adjusting the position of the buttons

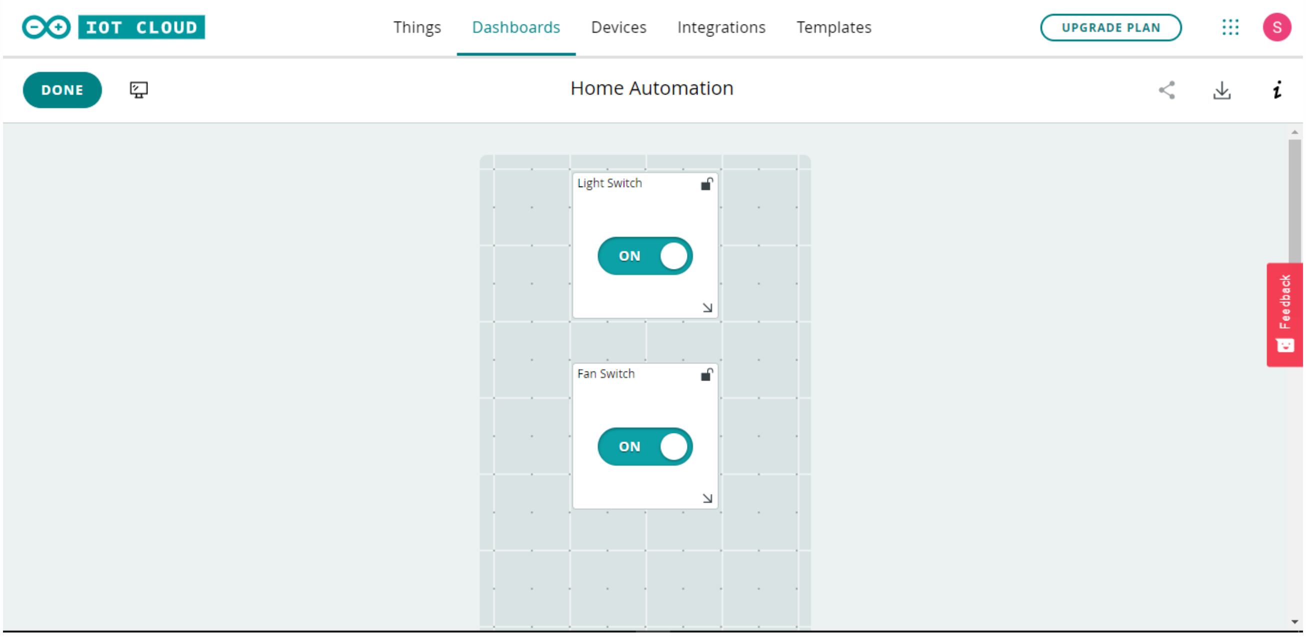
Click on the "Arrange Widgets" button which is present beside the "Add" button.



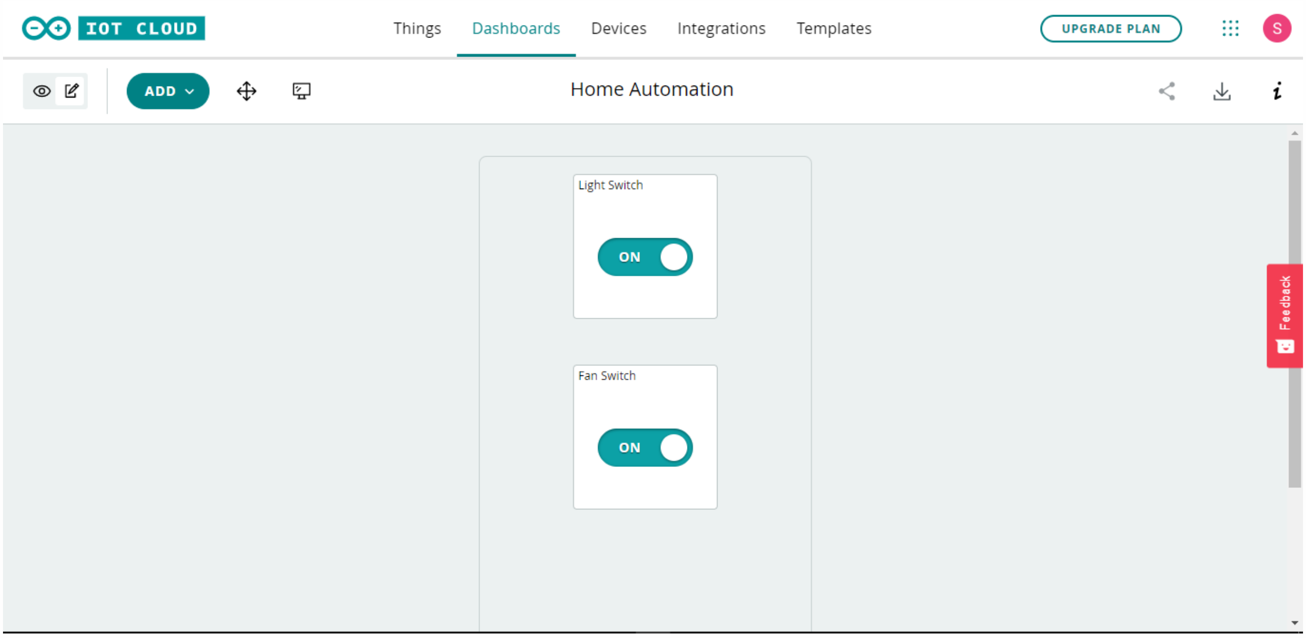
a. Move the widgets as per your requirement and then click on the "Mobile" icon beside the "done" for the mobile view.

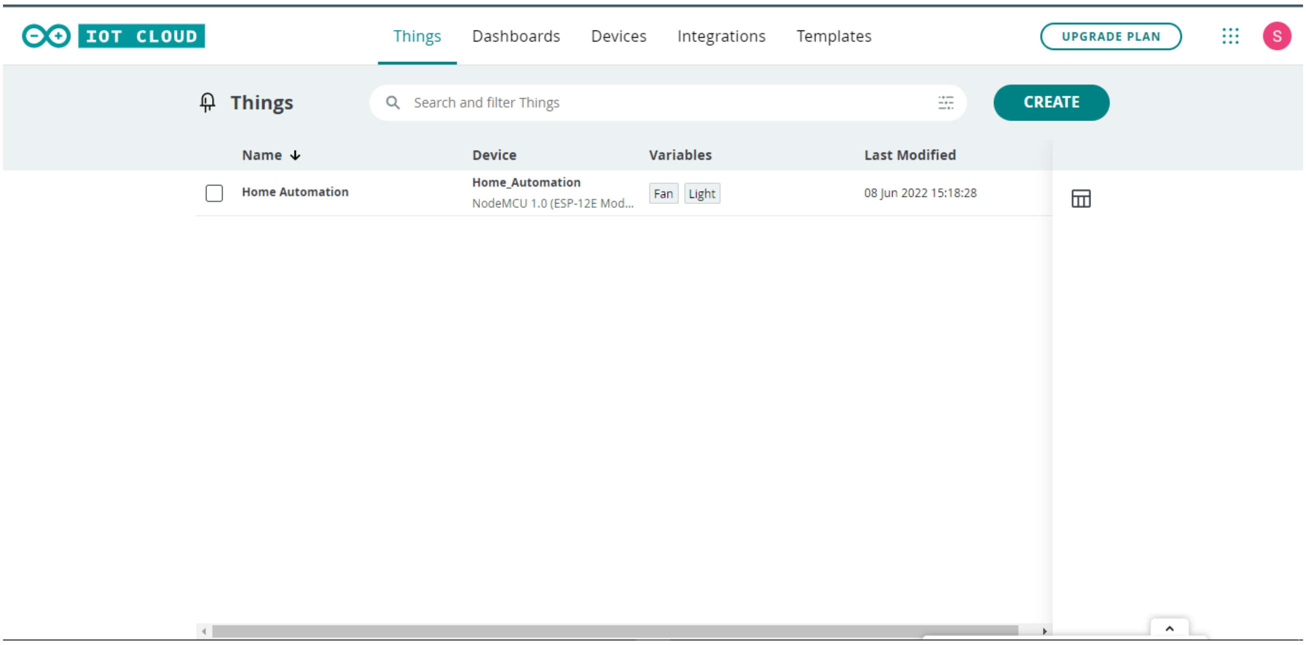


b. Adjust the position and click on "Done".



c. Now let’s move on to the Things page. Click on the "Things" page beside the "Dashboards" page.



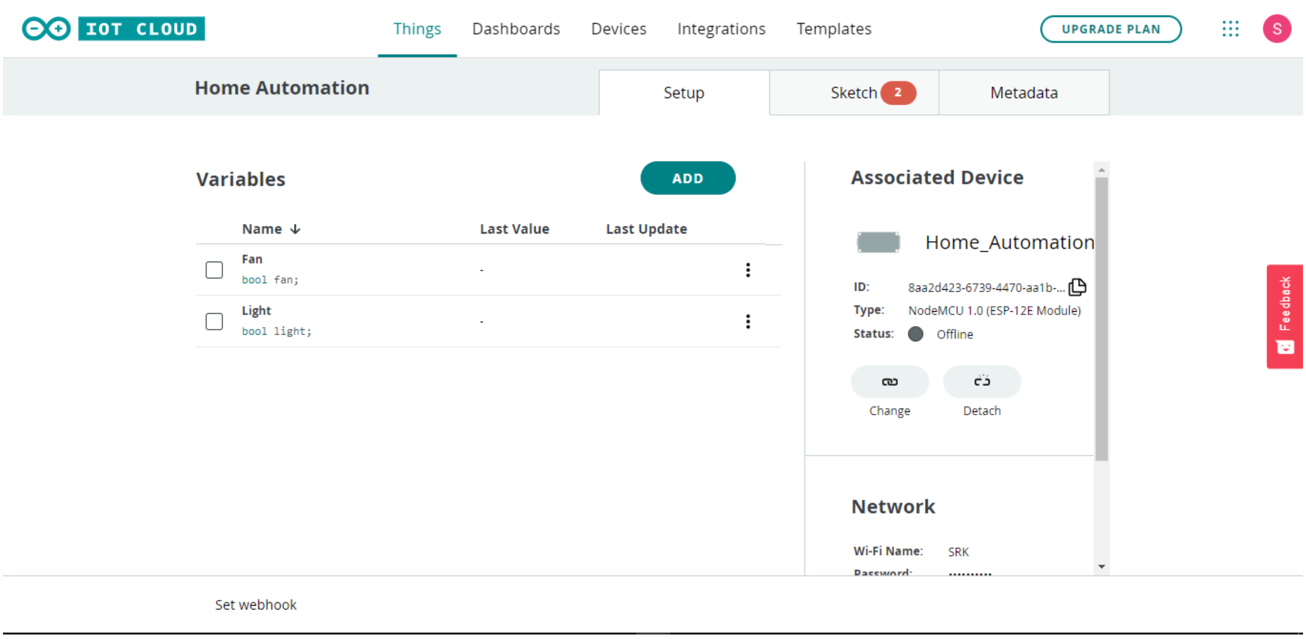


Name ↓	Device	Variables	Last Modified
□ Home Automation	Home_Automation NodeMCU 1.0 (ESP-12E Mod...	Fan Light	08 Jun 2022 15:18:28

PART 4: Writing the code

Step 24

Click on the “Sketch” page.



Home Automation

Variables

Name	Last Value	Last Update
Fan bool fan;	-	
Light bool light;	-	

Associated Device

Home_Automation

ID: 8aa2d423-6739-4470-aa1b-...

Type: NodeMCU 1.0 (ESP-12E Module)

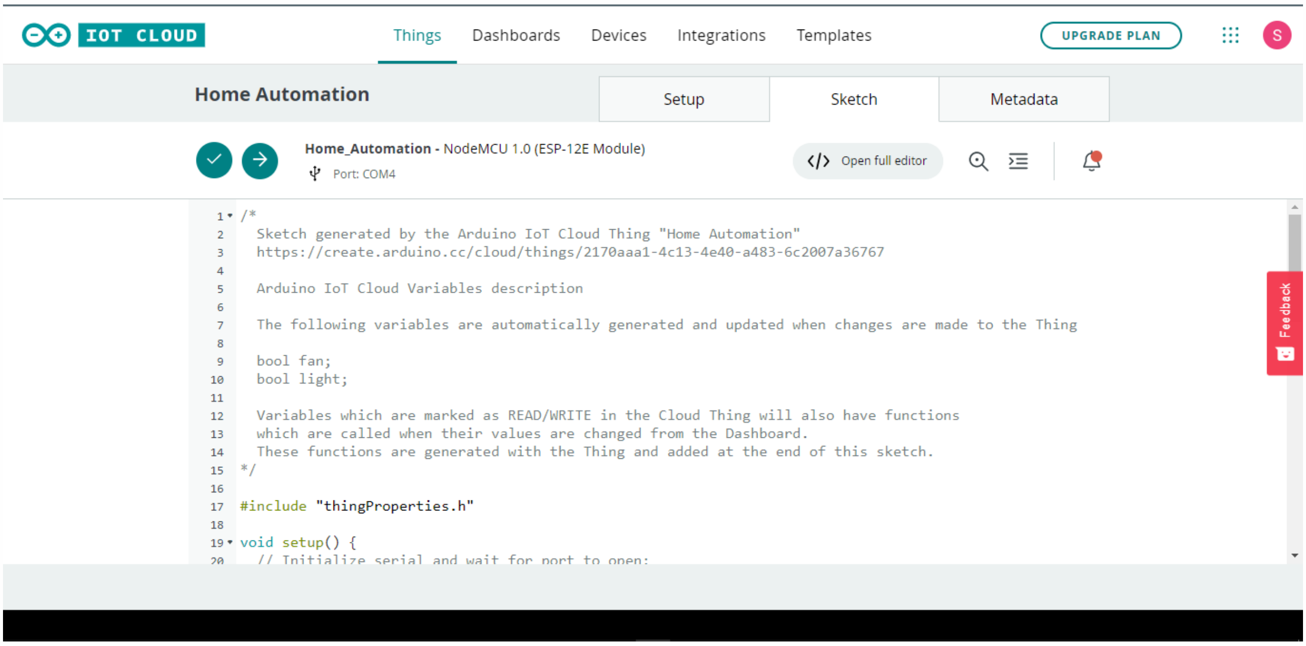
Status: Offline

Network

Wi-Fi Name: SRK

Password: ....

Now you are on the sketch page. Scroll down to the “void setup” function.



Home Automation

Home_Automation - NodeMCU 1.0 (ESP-12E Module)

Port: COM4

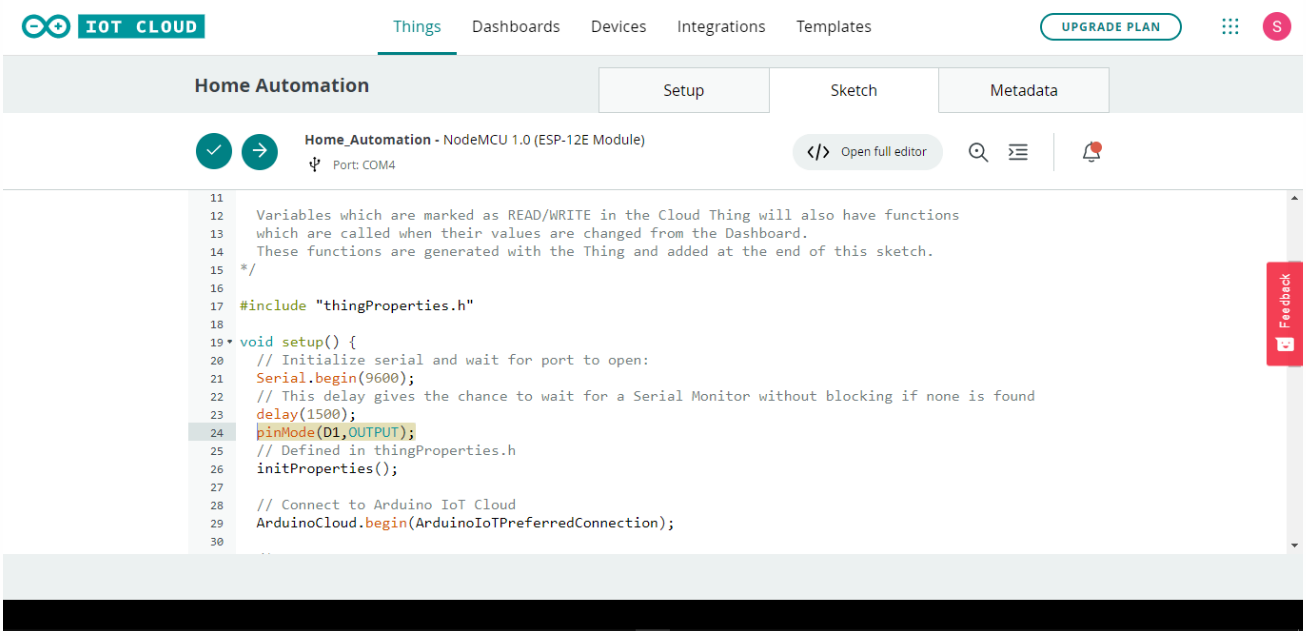
```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Home Automation"
  https://create.arduino.cc/cloud/things/2170aaa1-4c13-4e40-a483-6c2007a36767

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  bool fan;
  bool light;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
```

Step 25

Now let’s write the code for setting up the pin modes.

a. We know that we have connected our Light to D1 pin on the Node MCU and the Light is an output device.

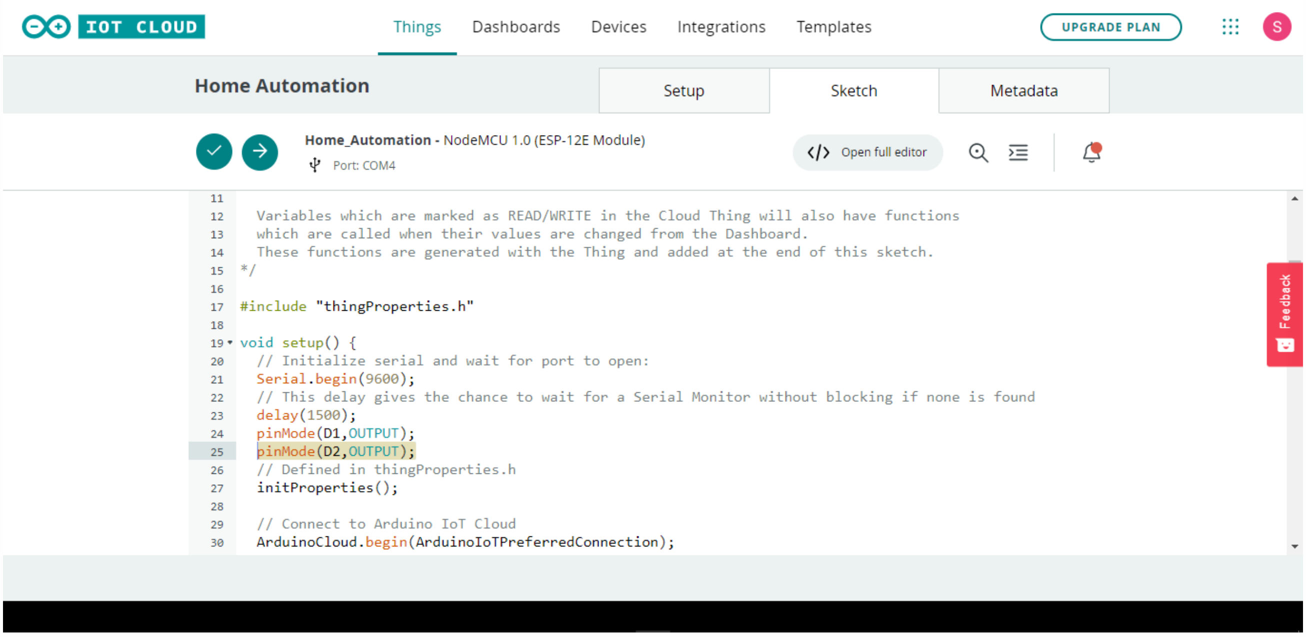
So let’s write the code: **pinMode(D1,OUTPUT);**



```
  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(D1,OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

b. We have connected our Fan relay to the D2 pin of the Node MCU.

So let's write the code: **pinMode(D2,OUTPUT);**



```
  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(D1,OUTPUT);
  pinMode(D2,OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

Step 26

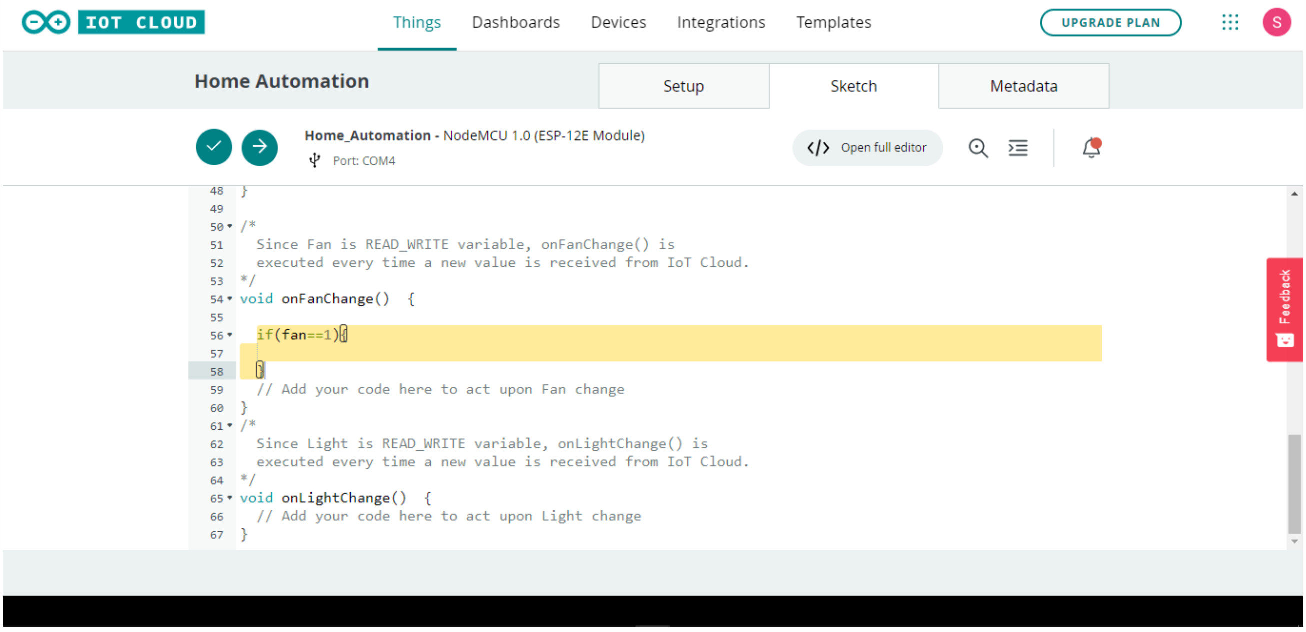
Now let's write the logic to turn on the fan.

Go to the "void onFanChange()" function and write the if block as shown below

If () {

}

Here we are using the “if” block because when the fan switch is turned on the fan has to run and when it is turned off the light has to be turned off.

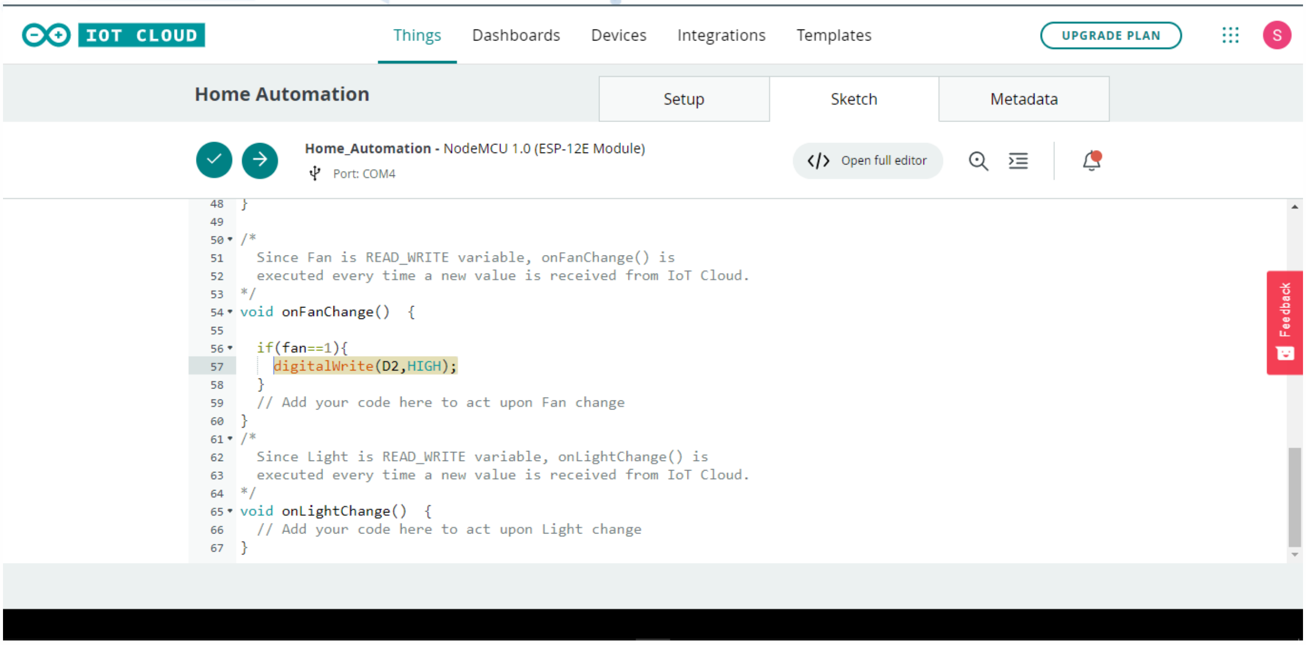


```
}

/*
  Since Fan is READ_WRITE variable, onFanChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onFanChange()  {

  if(fan==1){

  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {
  // Add your code here to act upon Light change
}
```

Step 27

Write the condition as “fan==1” in the “if” function which indicates that the user requested to turn on the fan.

Write **digitalWrite(D2,HIGH);** *(This line of code turns on the fan relay)*



```
}

/*
  Since Fan is READ_WRITE variable, onFanChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onFanChange()  {

  if(fan==1){
    digitalWrite(D2,HIGH);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {
  // Add your code here to act upon Light change
}
```

Step 28

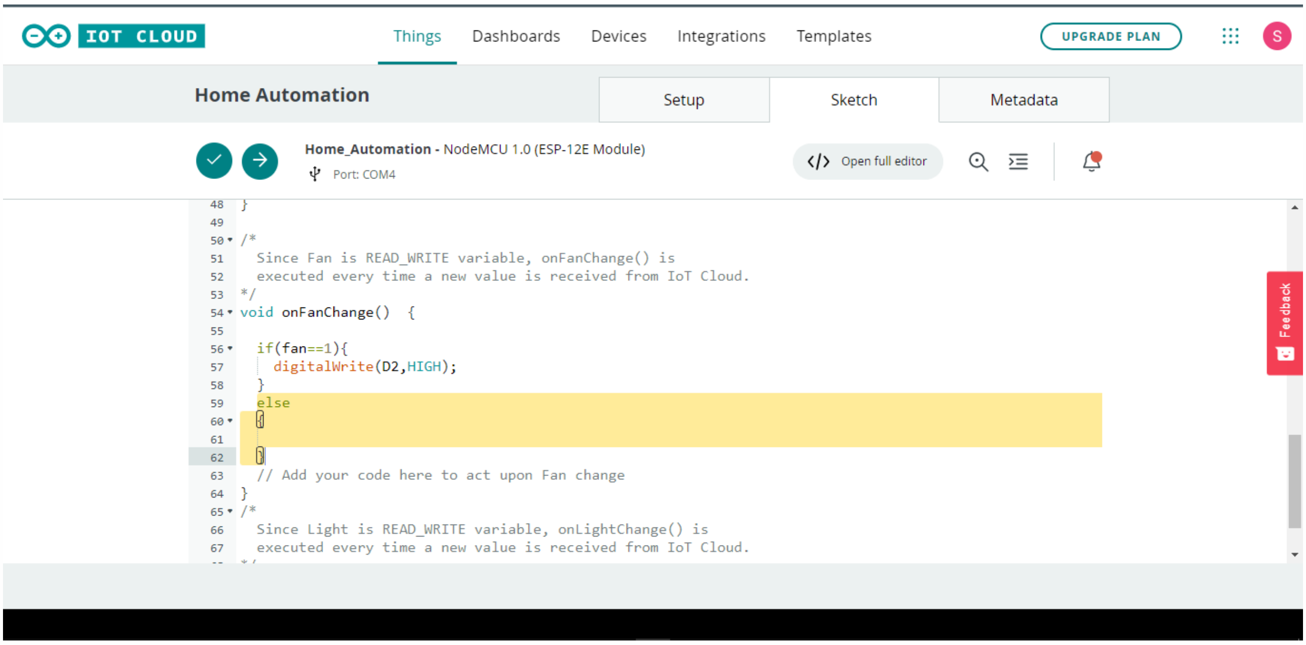
Let’s write the condition for turning off the Fan.

So when the switch is off the fan has to be stopped. For this, we will use an “else” function. Write the function as shown below.

else

{

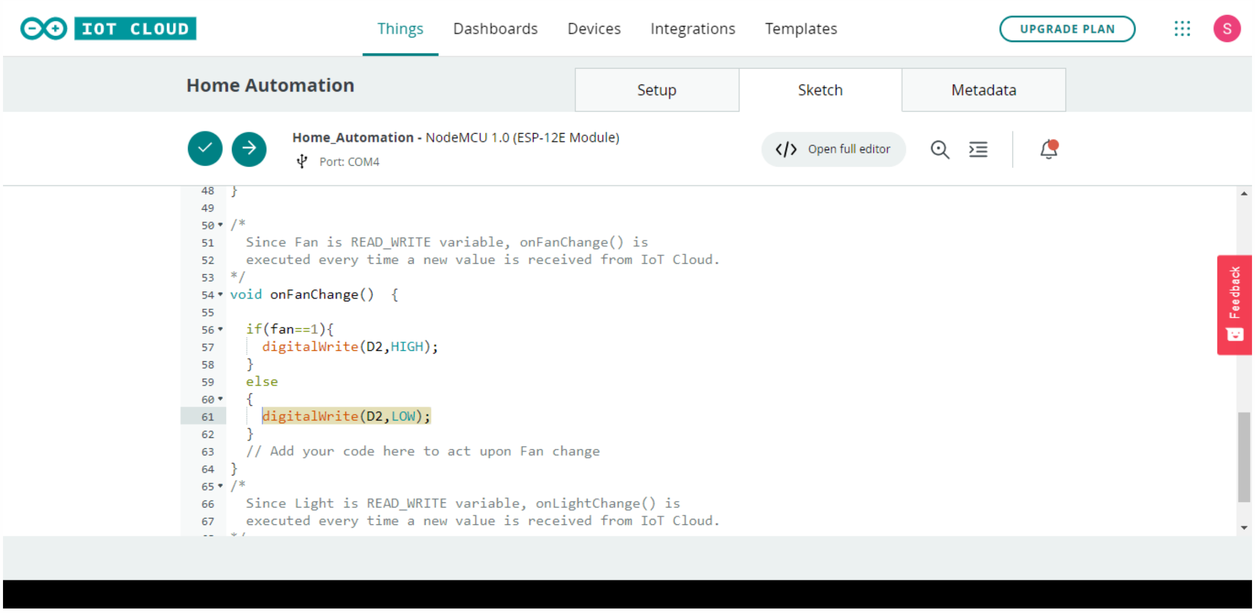
}



```
}

/*
  Since Fan is READ_WRITE variable, onFanChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onFanChange()  {

  if(fan==1){
    digitalWrite(D2,HIGH);
  }
  else
  {

  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
```

For turning off the fan we will write the code in the "else" block as

digitalWrite(D2,LOW);



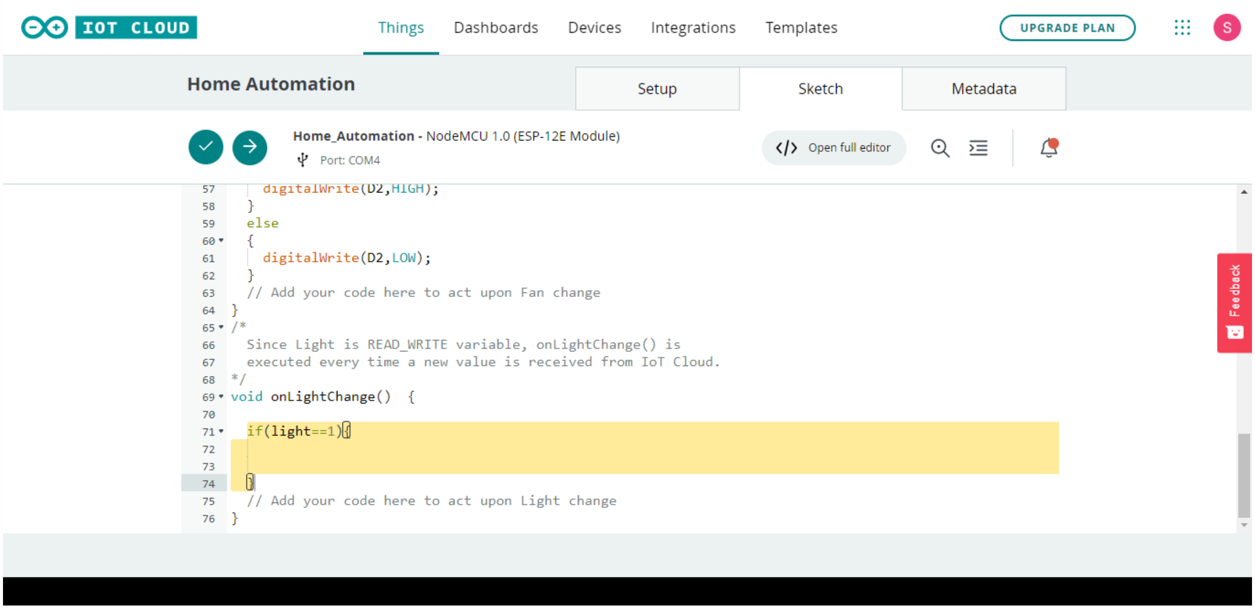
```
}

/*
  Since Fan is READ_WRITE variable, onFanChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onFanChange()  {

  if(fan==1){
    digitalWrite(D2,HIGH);
  }
  else
  {
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
```

Step 29

Let’s write the same code for the light also. Go to the “void onLightChange()” function, Then write the

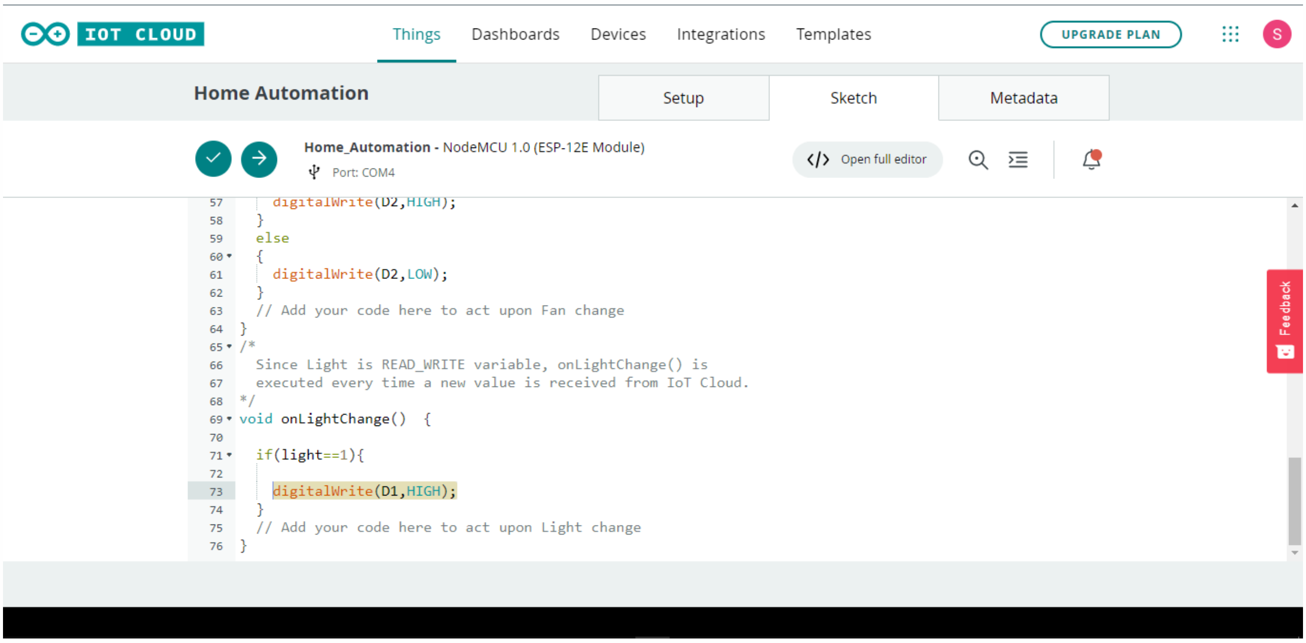
if(light==1){

}



```
    digitalWrite(D2,HIGH);
  }
  else
  {
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {

  if(light==1){


  }
  // Add your code here to act upon Light change
}
```

Write the code as **digitalWrite(D1,HIGH);** *(this line of code turns on the light when the switch is turned on)*



```
    digitalWrite(D2,HIGH);
  }
  else
  {
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {

  if(light==1){

    digitalWrite(D1,HIGH);
  }
  // Add your code here to act upon Light change
}
```

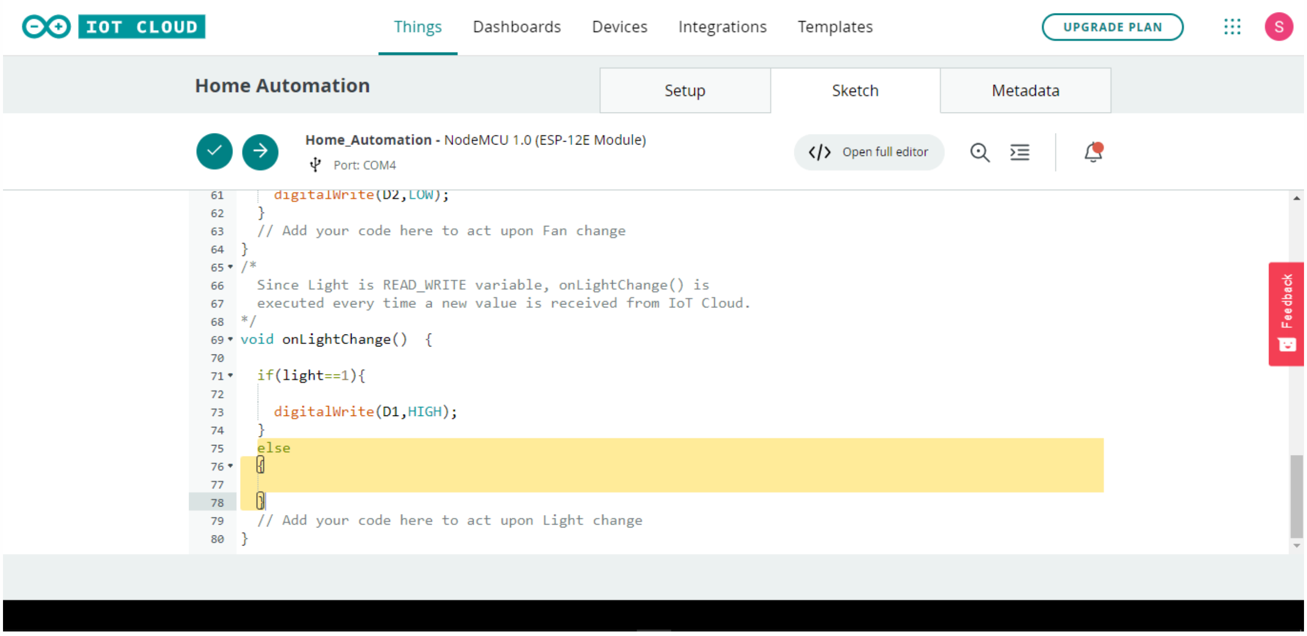
Step 30

Let's write the condition for the state - if the light is not turned on.

Write an else block below the if block as shown in the image.

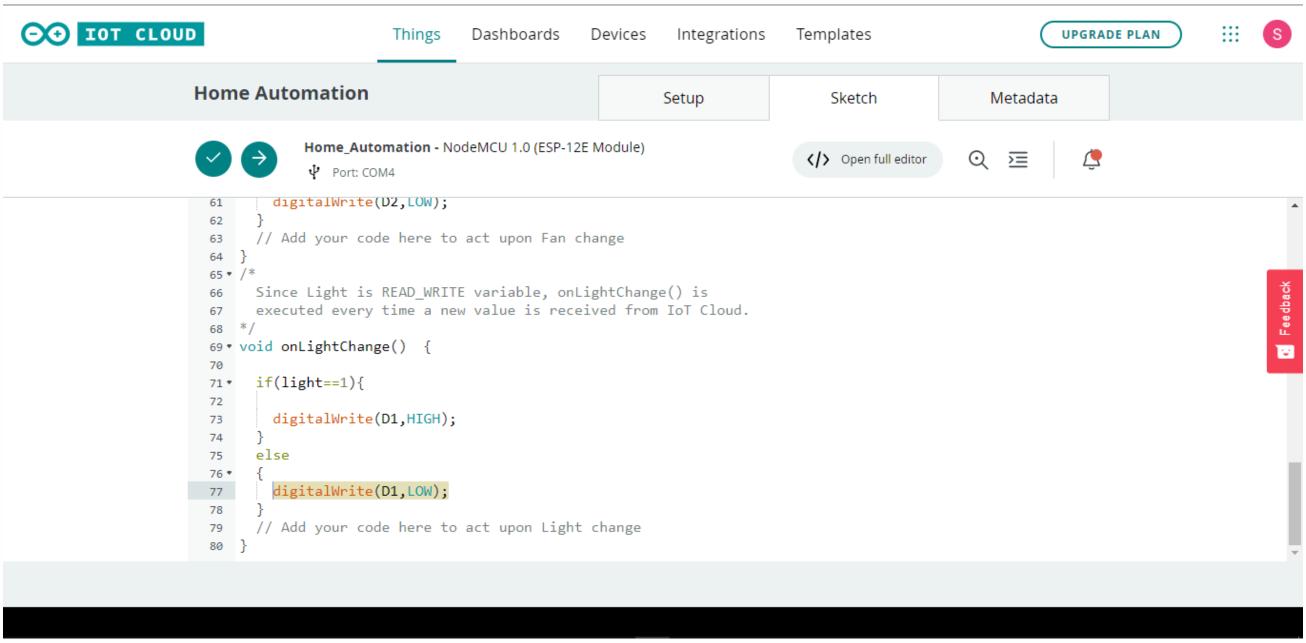
else {

}



```
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {

  if(light==1){

    digitalWrite(D1,HIGH);
  }
  else
  {

  }
  // Add your code here to act upon Light change
}
```

Write the code as **digitalWrite(D1,LOW);**

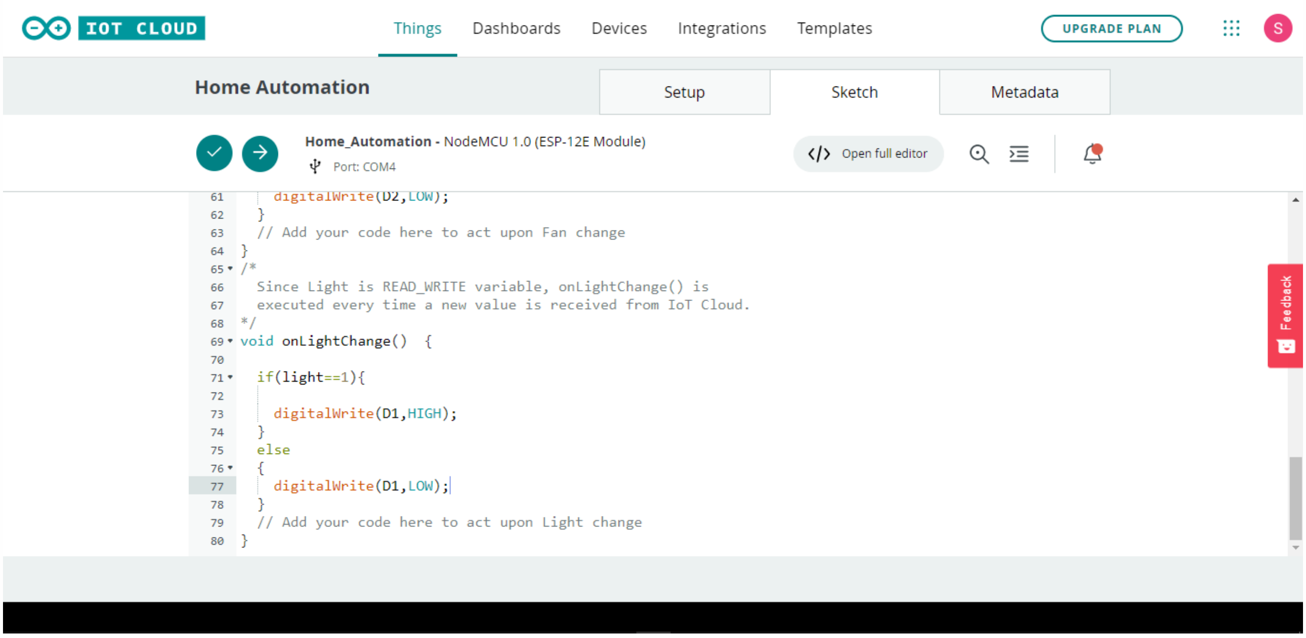


```
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {

  if(light==1){

    digitalWrite(D1,HIGH);
  }
  else
  {
    digitalWrite(D1,LOW);
  }
  // Add your code here to act upon Light change
}
```

Step 31

Now let's dump the code into the controller. To transfer the code, we need to select the port and board.

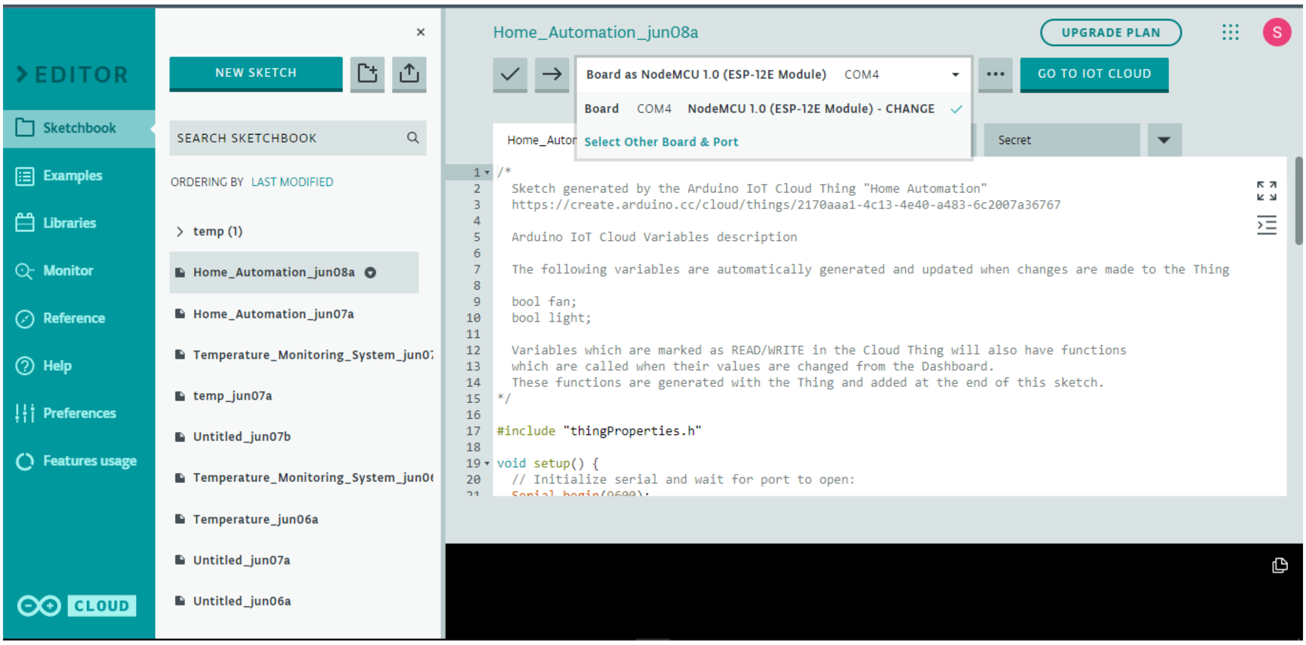
a. Now let's "open full editor" click on the open full editor button



```
    digitalWrite(D2,LOW);
  }
  // Add your code here to act upon Fan change
}
/*
  Since Light is READ_WRITE variable, onLightChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLightChange()  {

  if(light==1){

    digitalWrite(D1,HIGH);
  }
  else
  {
    digitalWrite(D1,LOW);
  }
  // Add your code here to act upon Light change
}
```

b. Before selecting the port we have to download the Arduino Create Agent application and Install it. Click on the link and download the application.

Now click on the “select boards” option.

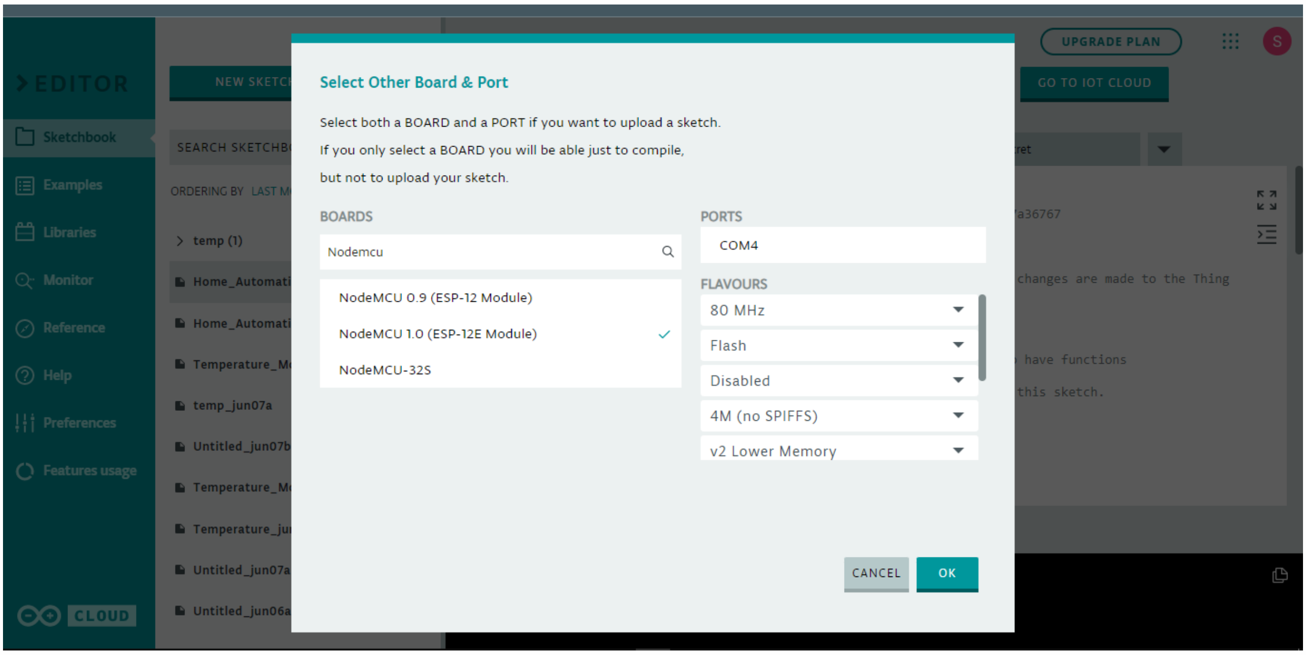


```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Home Automation"
  https://create.arduino.cc/cloud/things/2170aaa1-4c13-4e40-a483-6c2007a36767

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  bool fan;
  bool light;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
```

c. Search for Node MCU in the search bar and also select the COM port.

Then click on ok.

Note: Make sure the NodeMCU board is connected to the computer before choosing the port



Select Other Board & Port

Select both a BOARD and a PORT if you want to upload a sketch.
If you only select a BOARD you will be able just to compile,
but not to upload your sketch.

BOARDS

Nodemcu

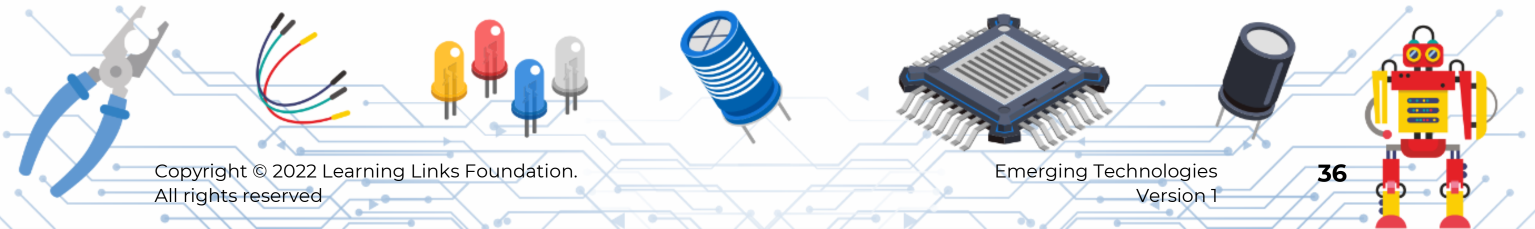
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module) ✓
NodeMCU-32S

PORTS

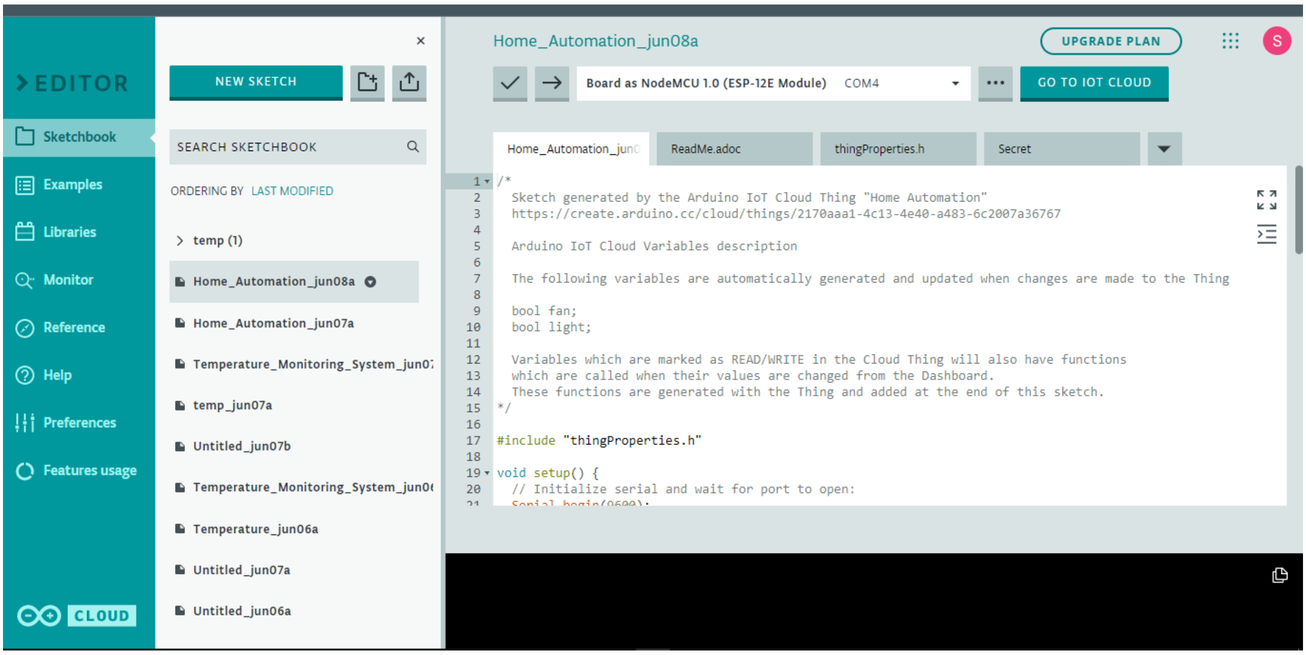
COM4

FLAVOURS

80 MHz
Flash
Disabled
4M (no SPIFFS)
v2 Lower Memory

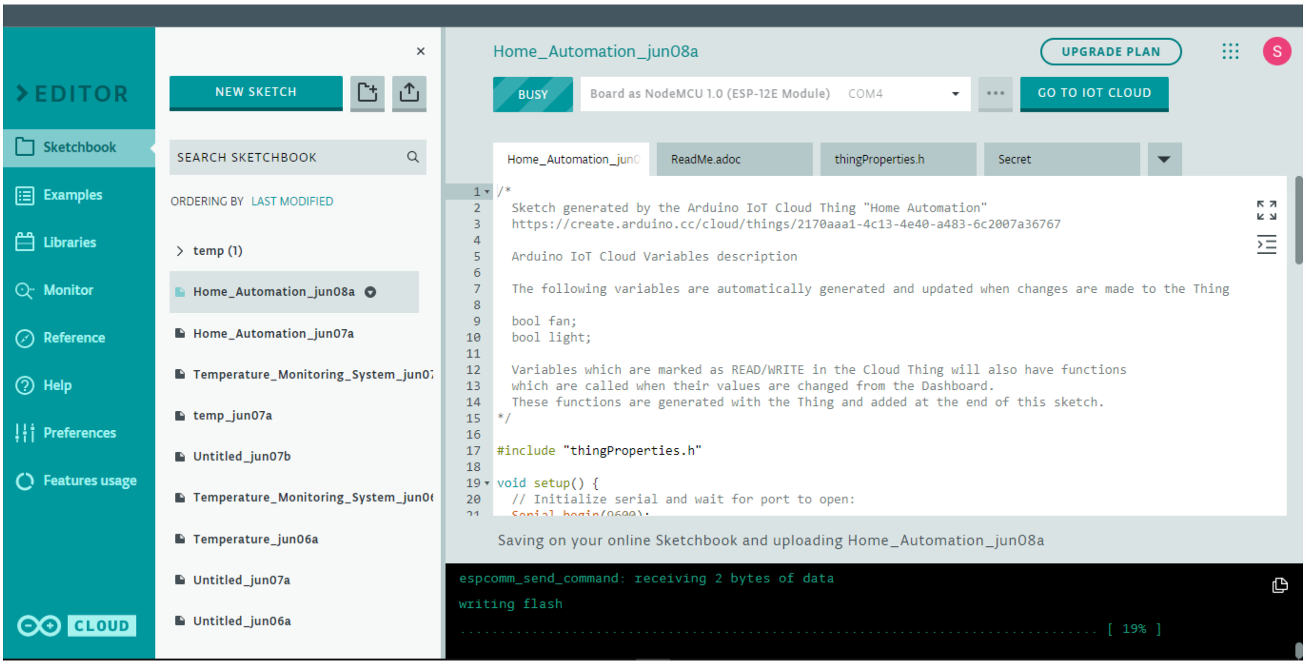


d. Now click on the “Save and upload button”



```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Home Automation"
  https://create.arduino.cc/cloud/things/2170aaa1-4c13-4e40-a483-6c2007a36767

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  bool fan;
  bool light;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
```

The program will be uploaded in a few seconds.



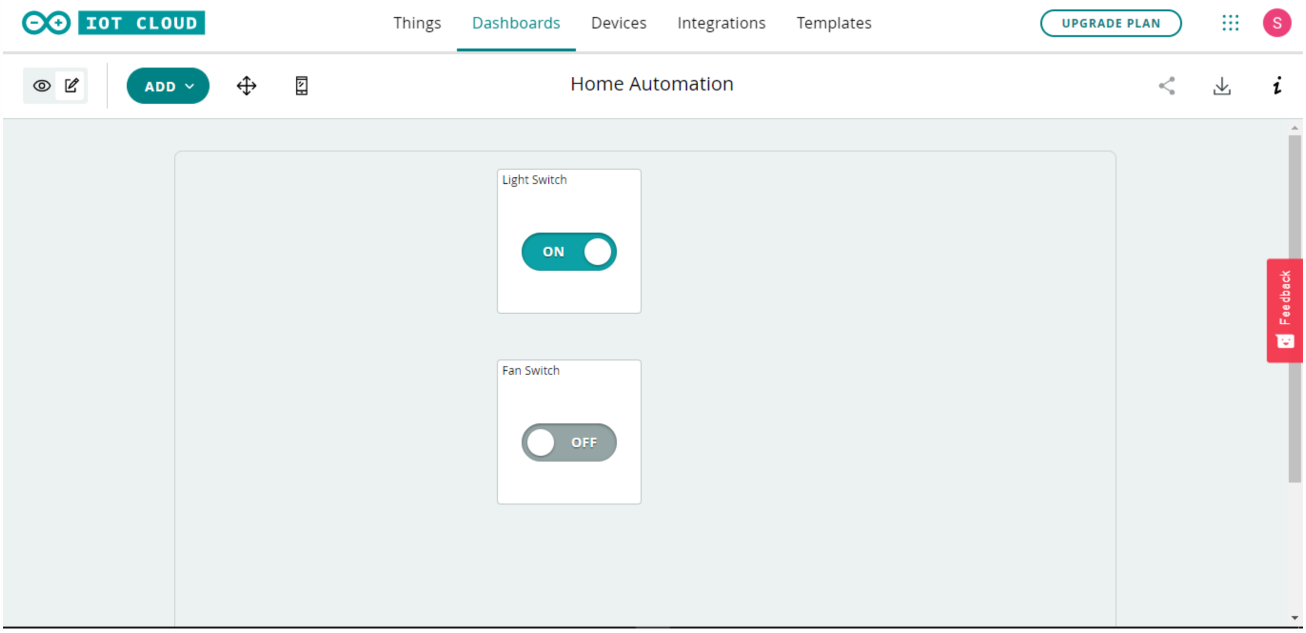
```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Home Automation"
  https://create.arduino.cc/cloud/things/2170aaa1-4c13-4e40-a483-6c2007a36767

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  bool fan;
  bool light;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
```

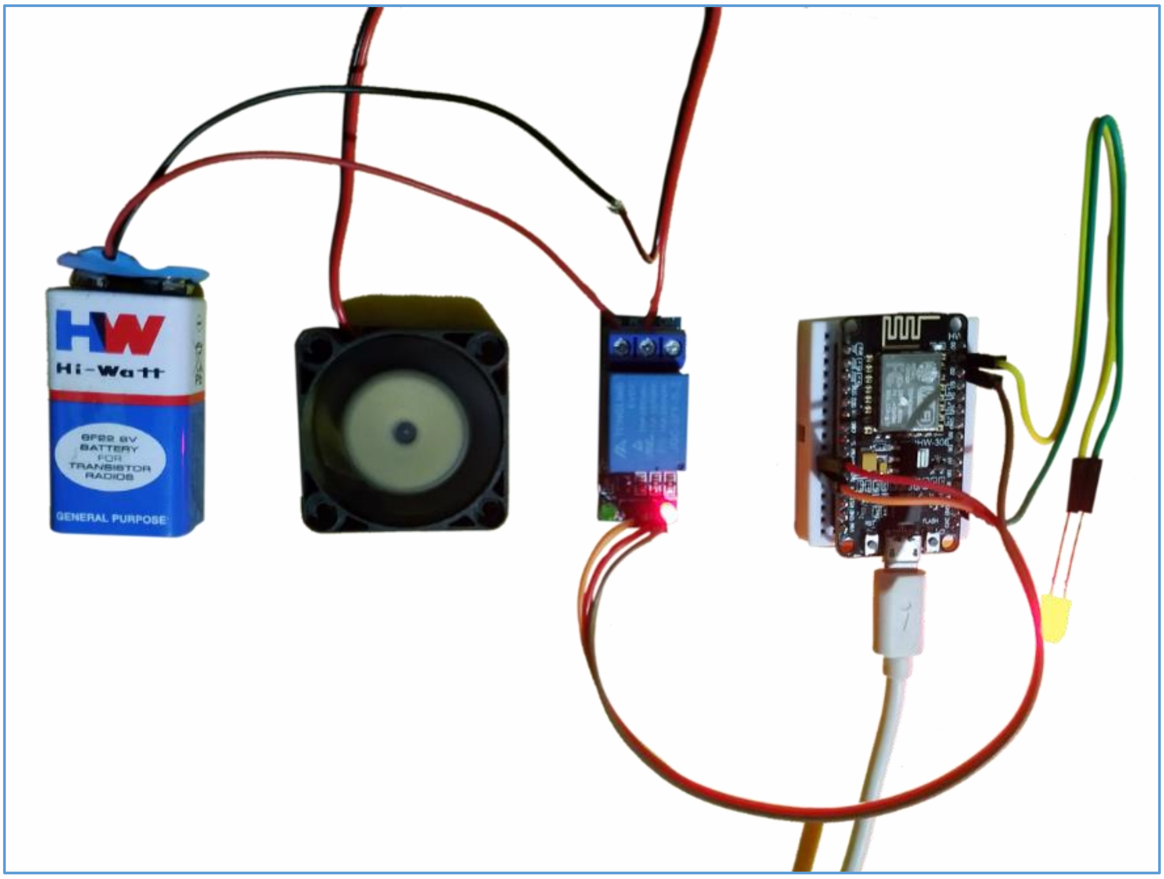
Saving on your online Sketchbook and uploading Home_Automation_jun08a

```
espcomm_send_command: receiving 2 bytes of data
writing flash
.......................................................................... [ 19% ]
```

Step 32

After uploading click on Dashboard and try testing the code with the circuit, when you turn on the switches the connected device should turn on and vice versa.



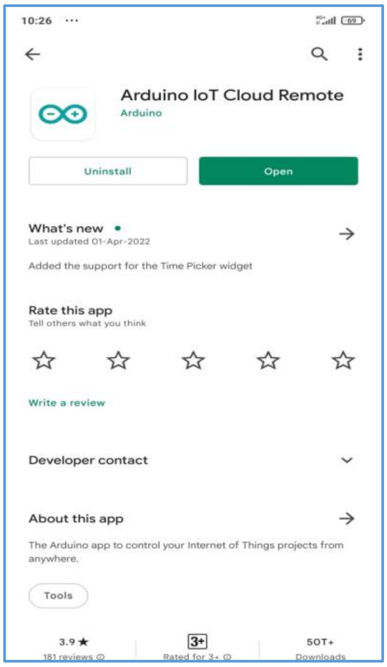


PART 5: IoT Application (Optional)

Step 33

Now let’s try controlling the devices using an android application.

Open “Playstore” on your mobile and search for “Arduino IoT” click on the Install button.



10:26 ···

Arduino IoT Cloud Remote

Arduino

Uninstall

Open

What's new •

Last updated 01-Apr-2022

Added the support for the Time Picker widget

Rate this app

Tell others what you think

☆ ☆ ☆ ☆ ☆

Write a review

Developer contact

About this app

The Arduino app to control your Internet of Things projects from anywhere.

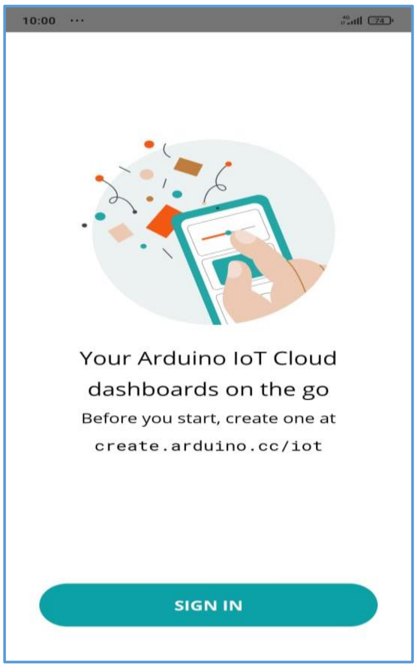
Tools

3.9 ★ 181 reviews ⓘ

3+ Rated for 3+ ⓘ

50T+ Downloads

After installation open the application and sign in.



10:00 ···

Your Arduino IoT Cloud dashboards on the go

Before you start, create one at

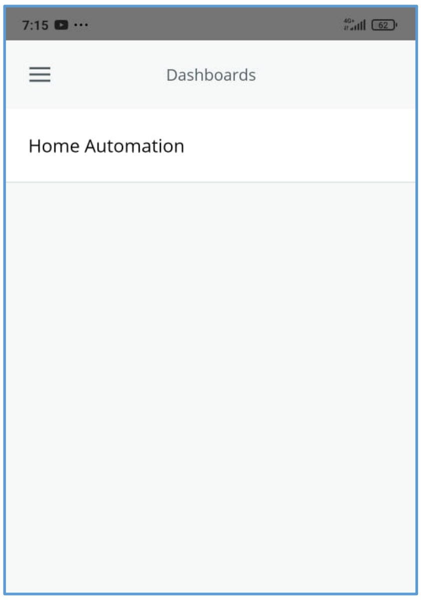
`create.arduino.cc/iot`

SIGN IN

Note: Login with the same Email ID which you have used for creating the IoT project.

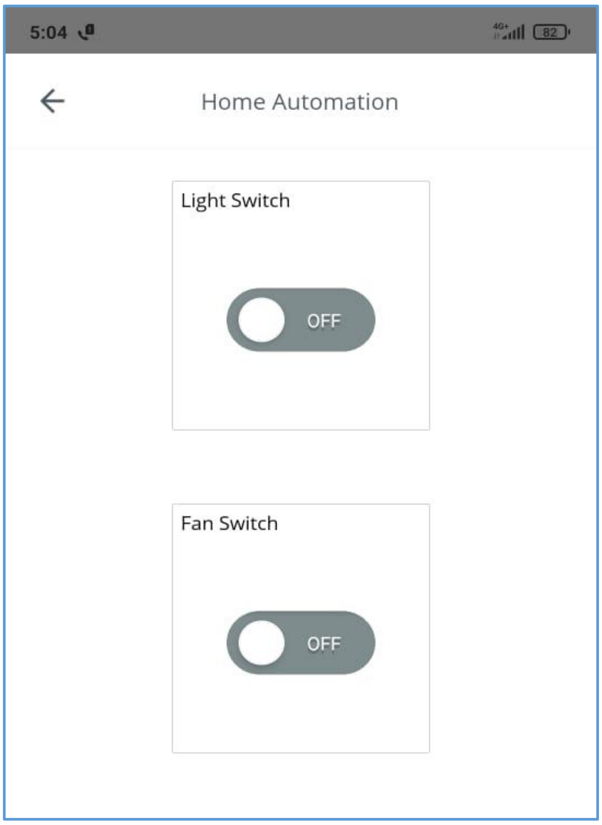
Step 34

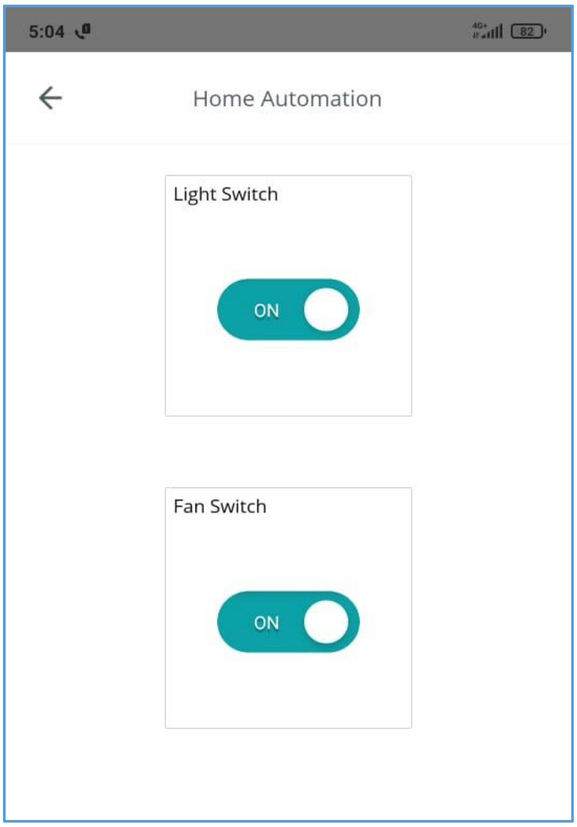
You will find the dashboard once you log in. Click on the dashboard and you will find the switches.

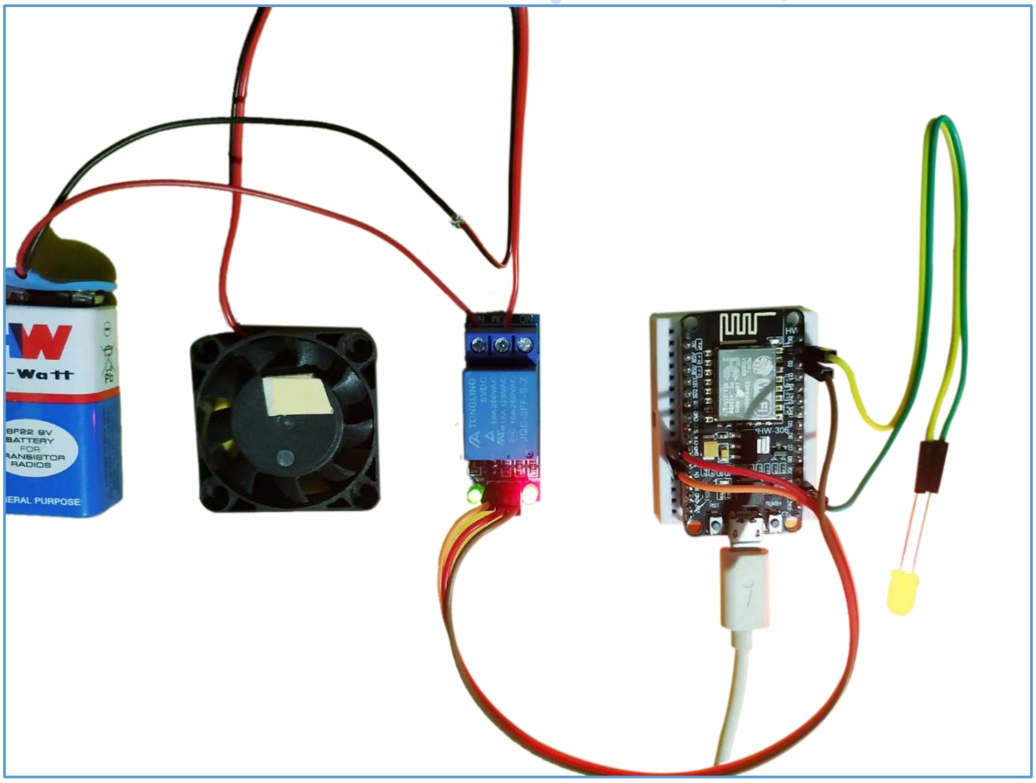


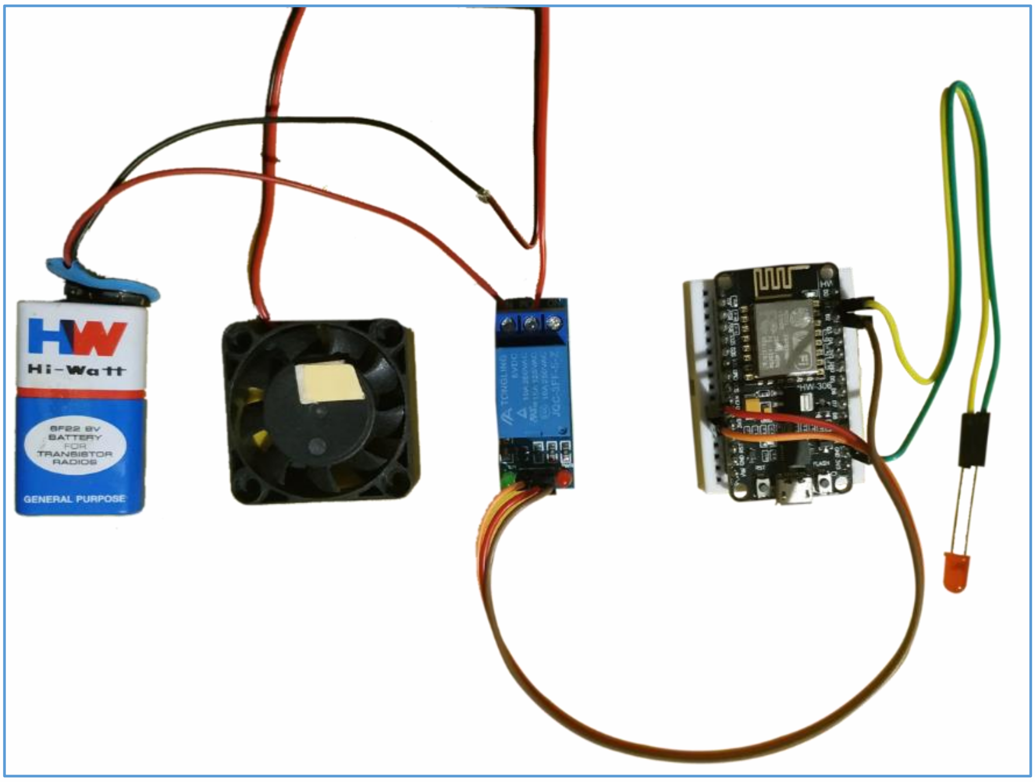
Home Automation

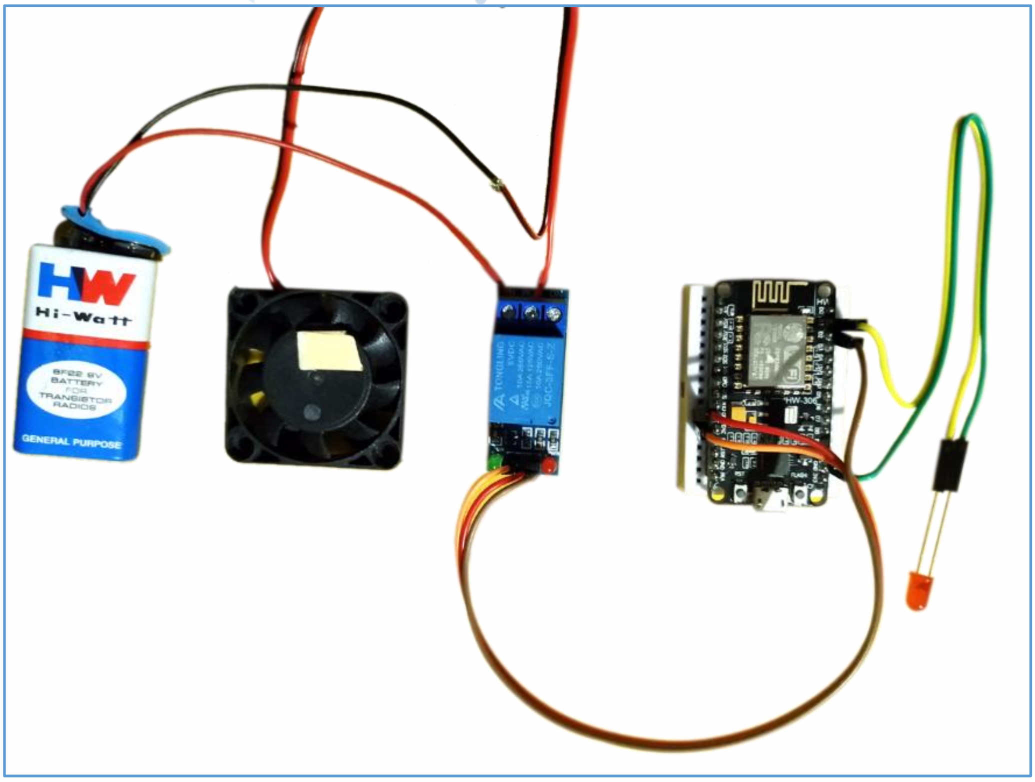
Try turning on or turning off the Fan and Light

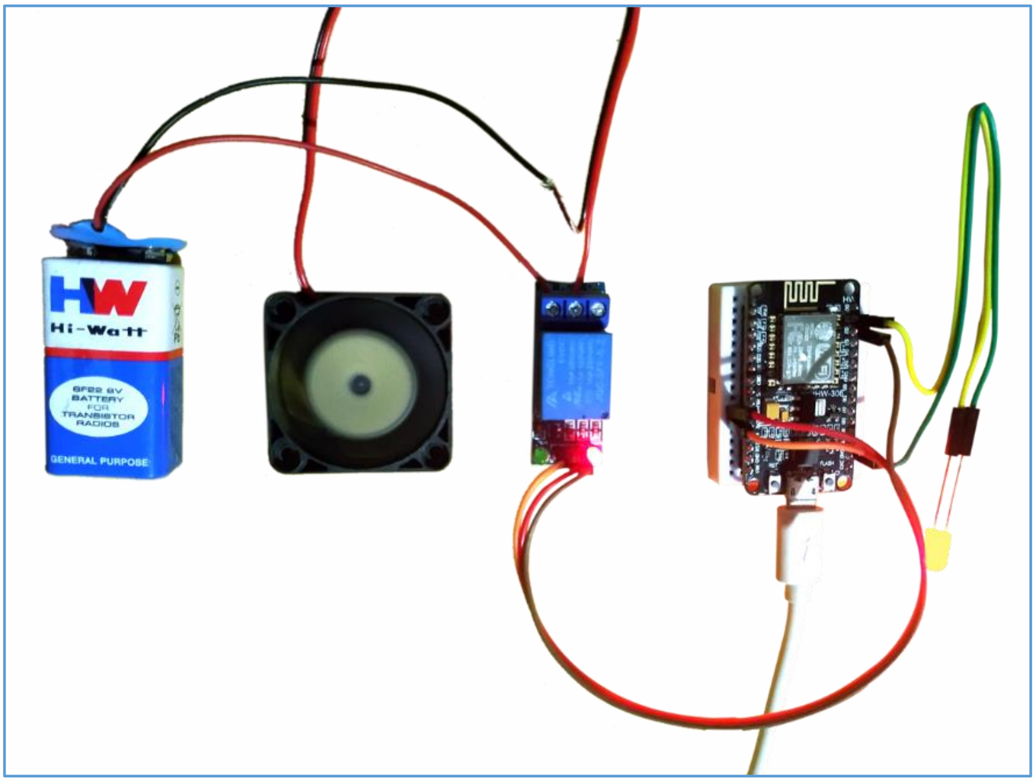


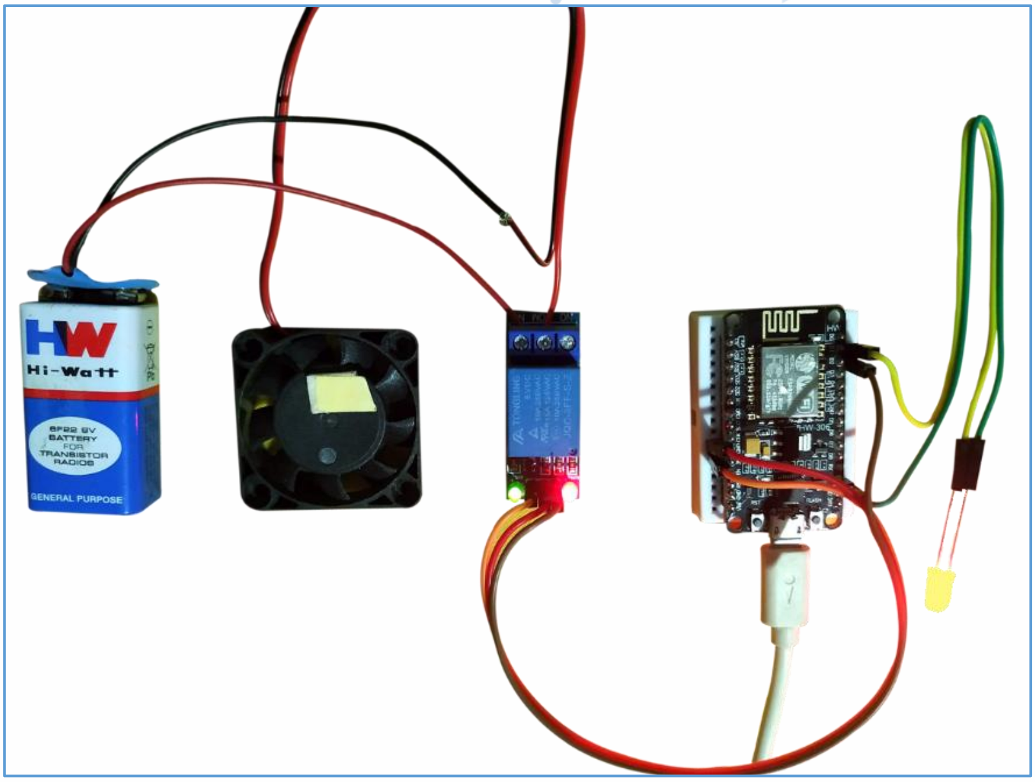


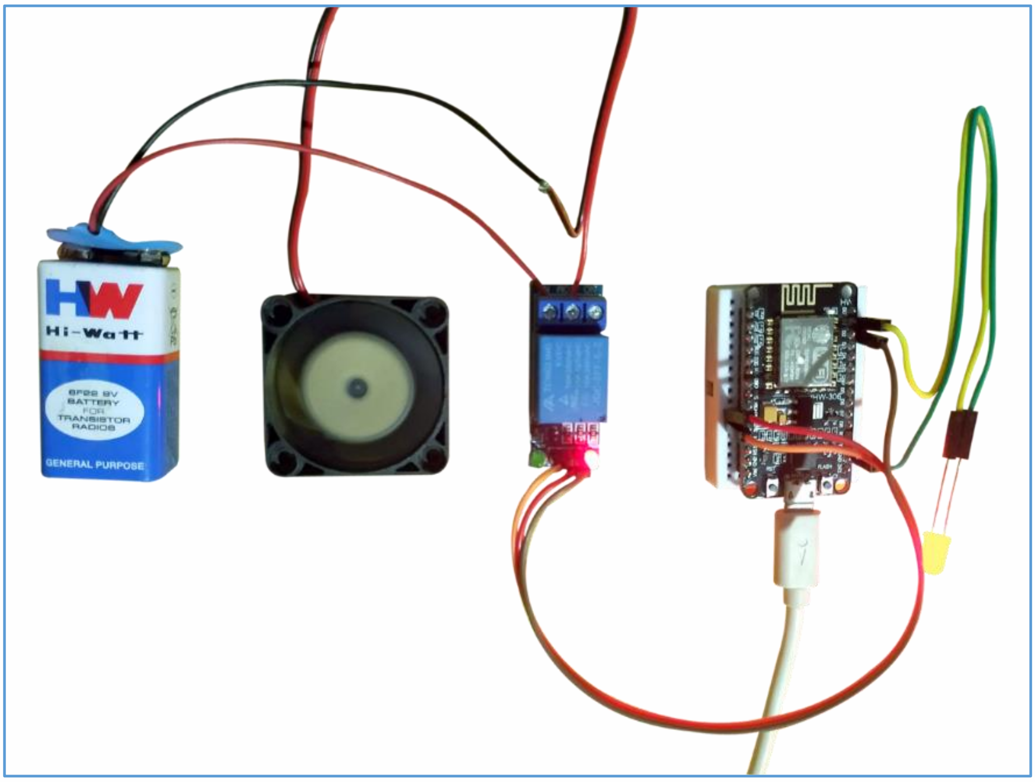





HW
Hi-Watt
6F22 9V BATTERY FOR TRANSISTOR RADIOS
GENERAL PURPOSE


HW
Hi-Watt
6F22 9V BATTERY FOR TRANSISTOR RADIOS
GENERAL PURPOSE





Step 35

Congratulations you have successfully completed the Project.

Reflection:

1. Research and find out the differences between and Arduino Uno Board and Node MCU.
2. What is the purpose of “if” function and “else” function?
3. What improvements do you recommend to this circuit?
4. What is the purpose of adding a relay? What will happen if relay is not added?
5. Have you ever seen a smart home in real-life? How do you think a smart home should be? If you were to build your own smart home, what features would like to have and why?

LEARNING LINKS
FOUNDATION
WHERE EDUCATION MEETS LEARNING

Contact us:
Learning Links Foundation
Allied House, 2nd Floor
Plot no. 5 & 6, B-7 Vasant Kunj
New Delhi 110070
contactus@learninglinksindia.org
www.learninglinksindia.org

www.facebook.com/learninglinksindia

LLF_IN

@learninglinksfoundation