INTRODUCTION TO RELAY

What is our GOAL for this CLASS?

In this class, we were introduced to Relay and we learned to operate AC appliances with Relay as per Google Assistant instructions.

What did we ACHIEVE in the class TODAY?

- We were introduced to Relay
- We learned about Arduino IoT Cloud.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- We learned how a button works.
  - When we toggle the level to the ON position, the contact closes completes the circuit and allows current to flow through the switch.
  - When we toggle the level to the OFF position, the contact opens, restricting current to flow through the switch.
- We learned about Relay
  - A relay is an electromagnetic switch operated by a relatively small electric current that can turn on or off a much larger electric current.
  - Electromagnet: A coil of wire that becomes a temporary magnet when electricity flows through it.
  - In relay two circuits are there, one for small electric current and the second for large current
- We learned how to make a circuit with Relay.

How did we DO the activities?

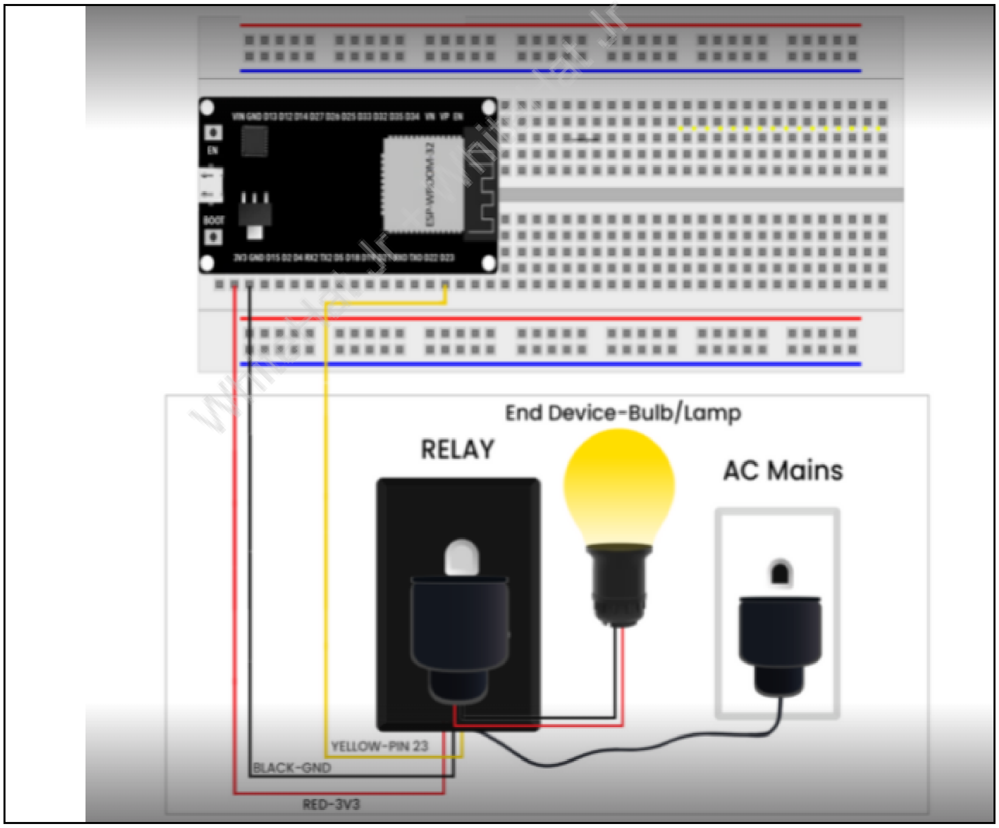
1. Gather the material from the IoT kit:

    - 1 x ESP32
    - 1 x USB Cable
    - 1 x Breadboard
    - 4 x Jumper wires
    - 1 x Relay
    - 1 x Bulb, Lamp, Mosquito Repellent Machine

2. Connections for **Circuit :**

    - **Relay VCC(Red) pin**: Connect with 5 V PIN of the ESP32
    - **Relay GND(Black) pin**: Connect with GND of the ESP32
    - **Relay Input(Yellow) pin**: Connect with GPIO PIN 23

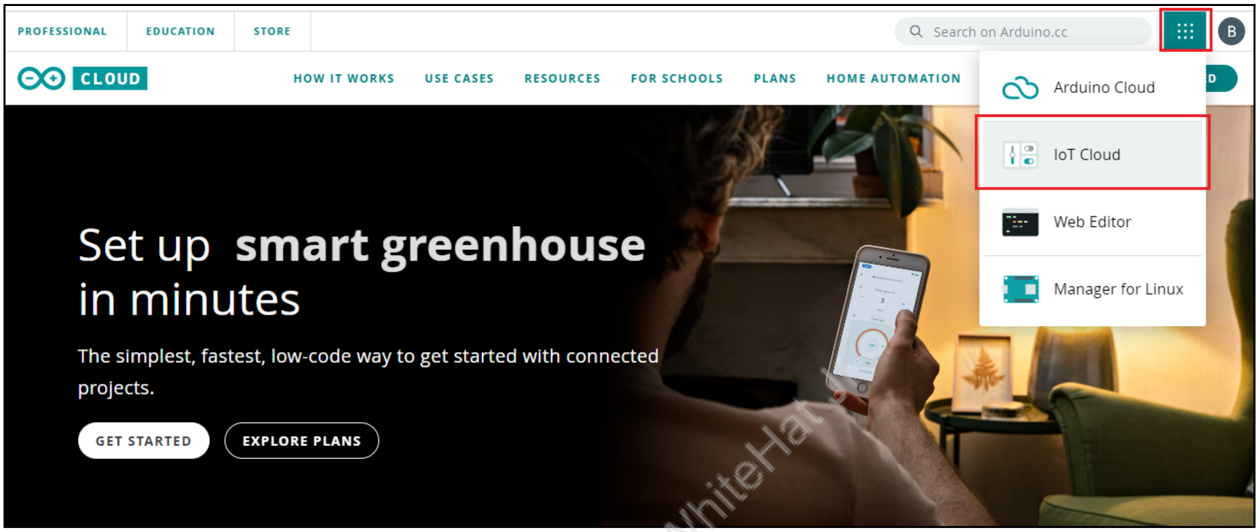
3. Reference Circuit Diagram:



4. Create an account on the Arduino IoT Cloud.

5. Once the account is created, go to the top right hand side of your screen and click on

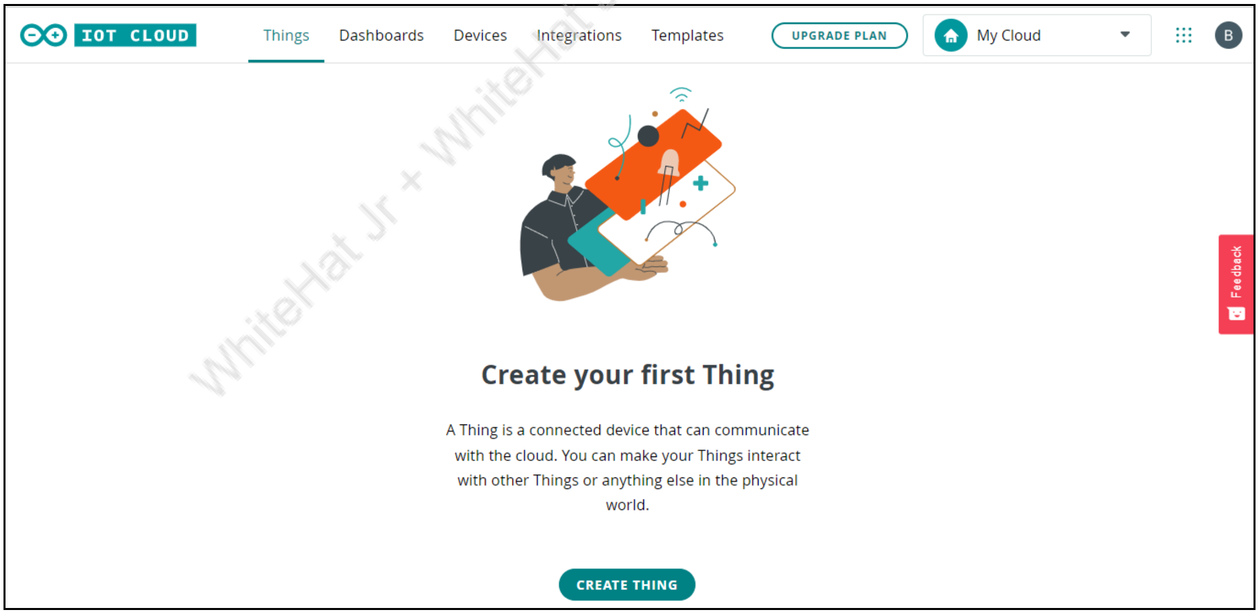
the ⠿ option.Then, click on **IoT Cloud.**



Set up smart greenhouse in minutes

The simplest, fastest, low-code way to get started with connected projects.

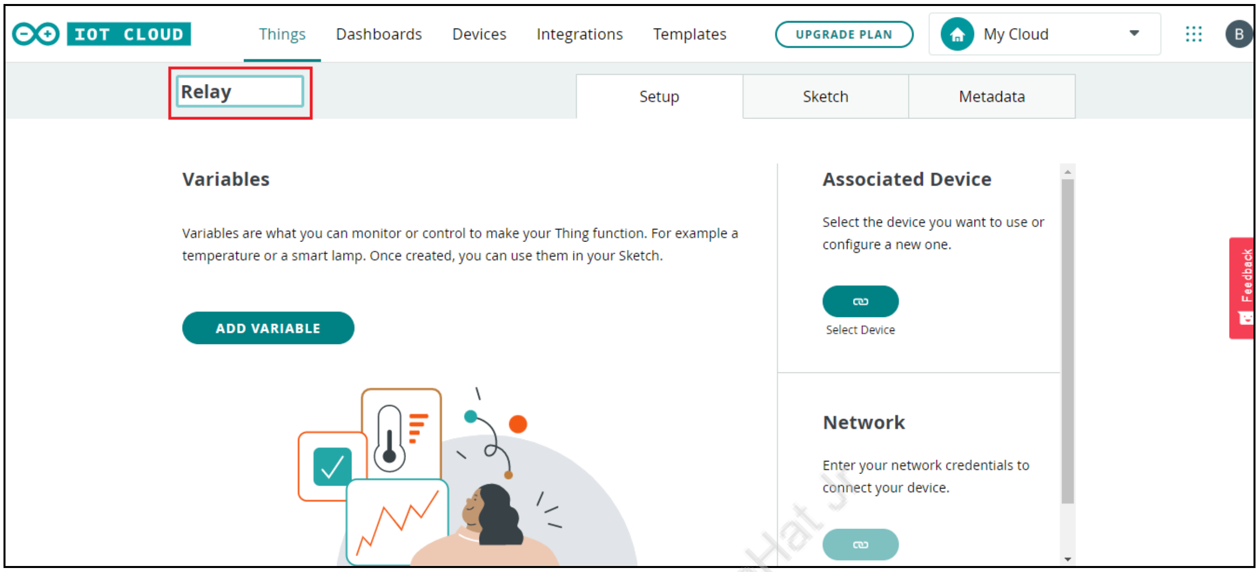
6. Create a "**thing**" for our internet of things. So, click on the **"CREATE THING"** button.



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.

7. Add a name for this project.



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.

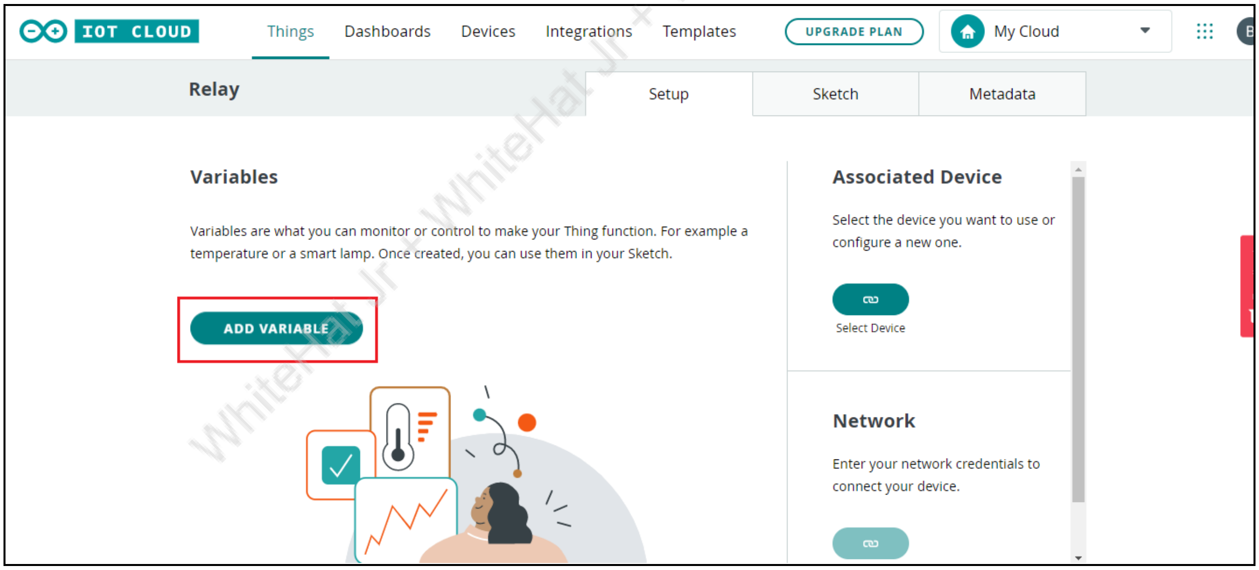
Associated Device

Select the device you want to use or configure a new one.

Network

Enter your network credentials to connect your device.

8. After that, add a **variable** named **led.** This variable will hold the status of our switch.



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.

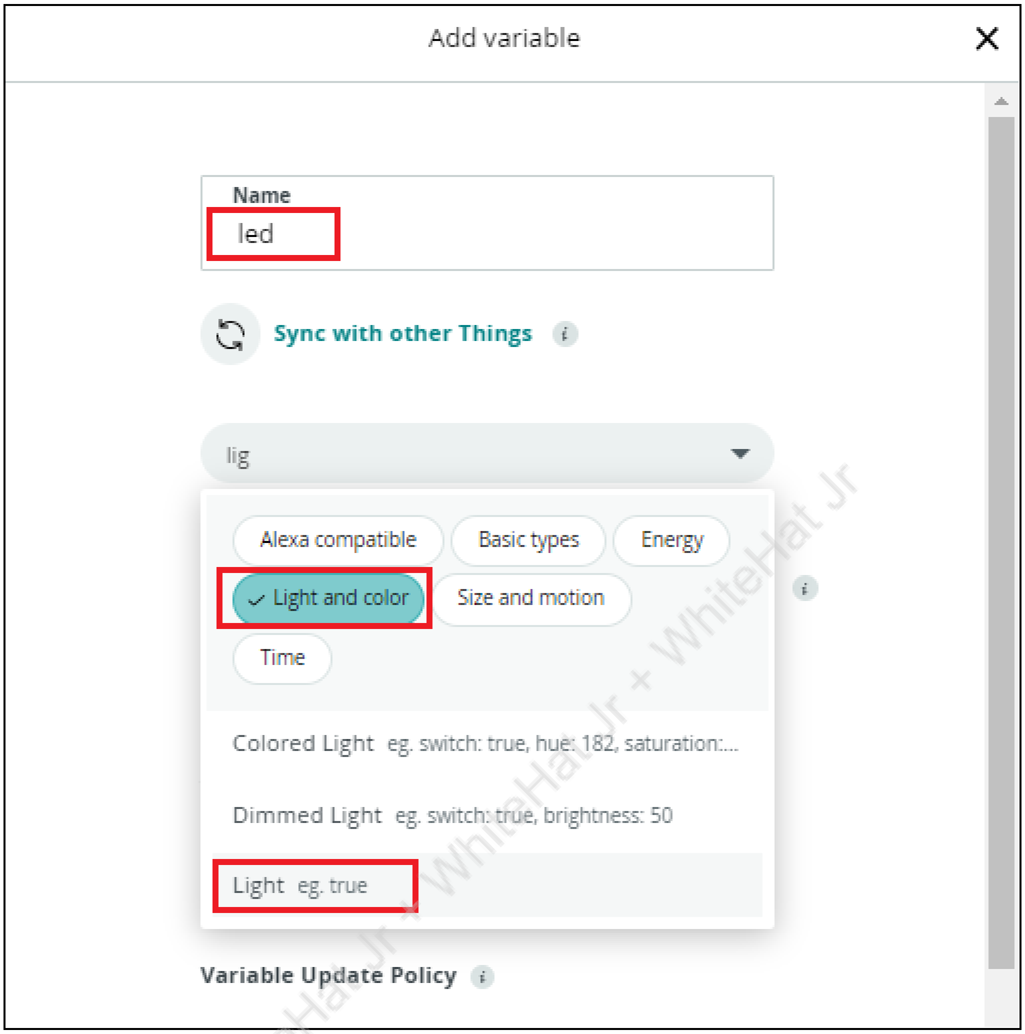
Associated Device

Select the device you want to use or configure a new one.

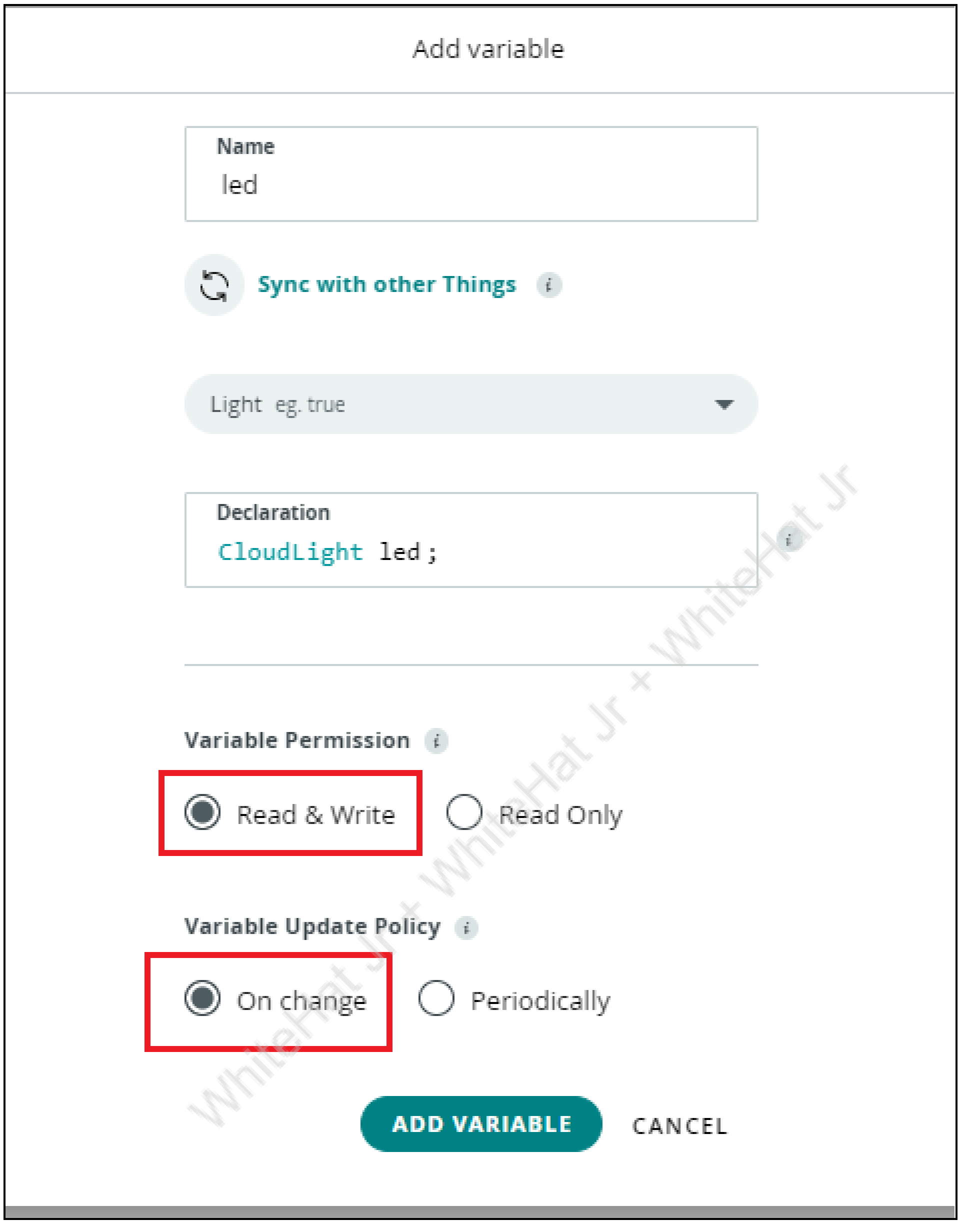
Network

Enter your network credentials to connect your device.

9. Change the name of the variable to **led.** Select variable type as “**Light**”.



10. To switch the relay on and off in the program, assign **Read & Write** permissions to this variable.
Also, to update this variable with a switch, set the **Variable Update Policy** to **On Change.**



Add variable

Name
led

Light eg. true

Declaration
```
CloudLight led;
```

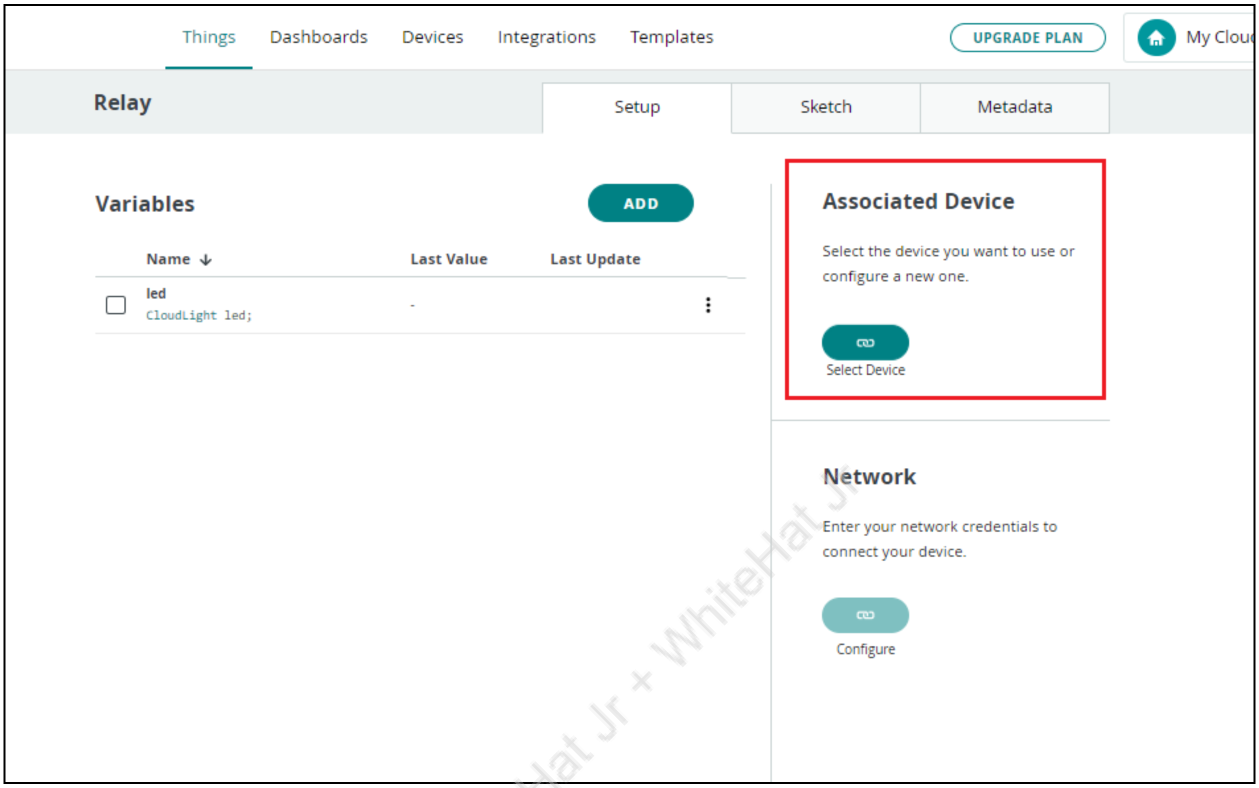
Variable Permission

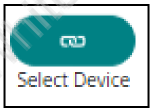
● Read & Write ○ Read Only

Variable Update Policy

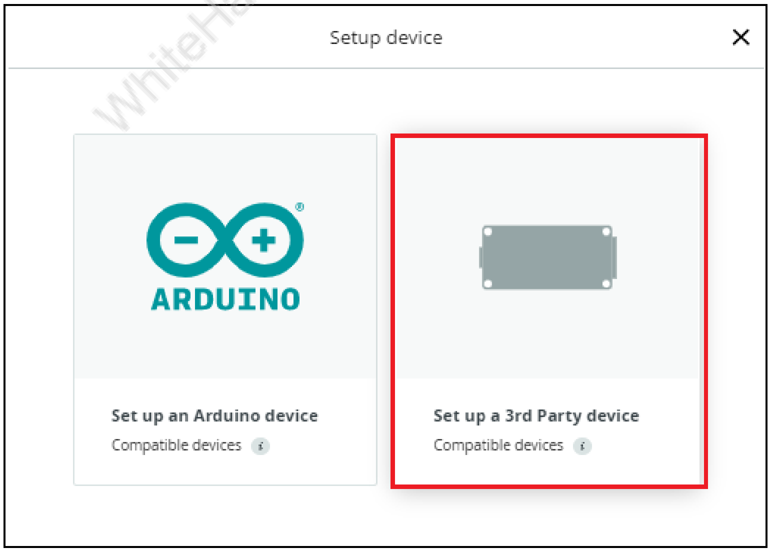
● On change ○ Periodically

11. Set up the **Associated Device**.



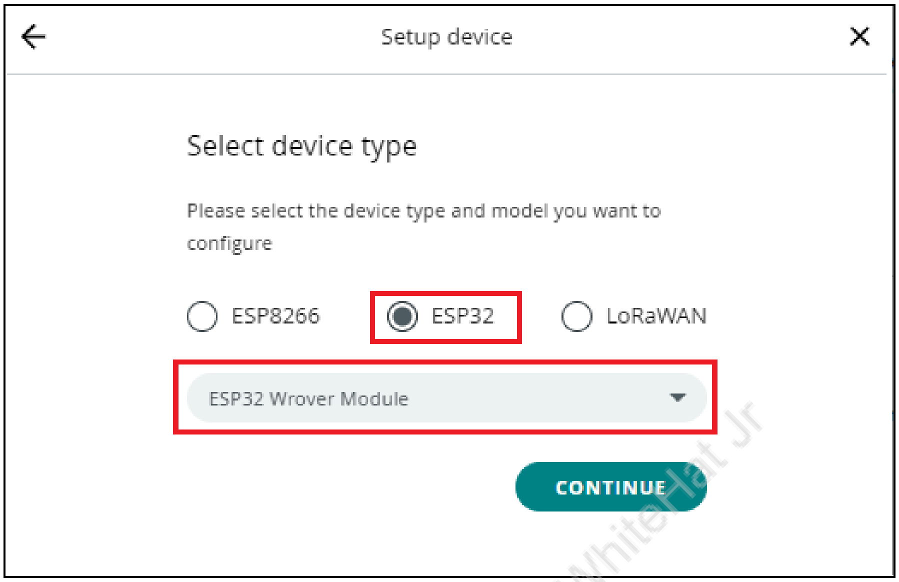
a. Click on the **Select Device**  button.

b. Click on **Set Up a 3rd Party device.**



c. Select the device type as **ESP32** and set the module as **ESP32 Wrover**

Module.



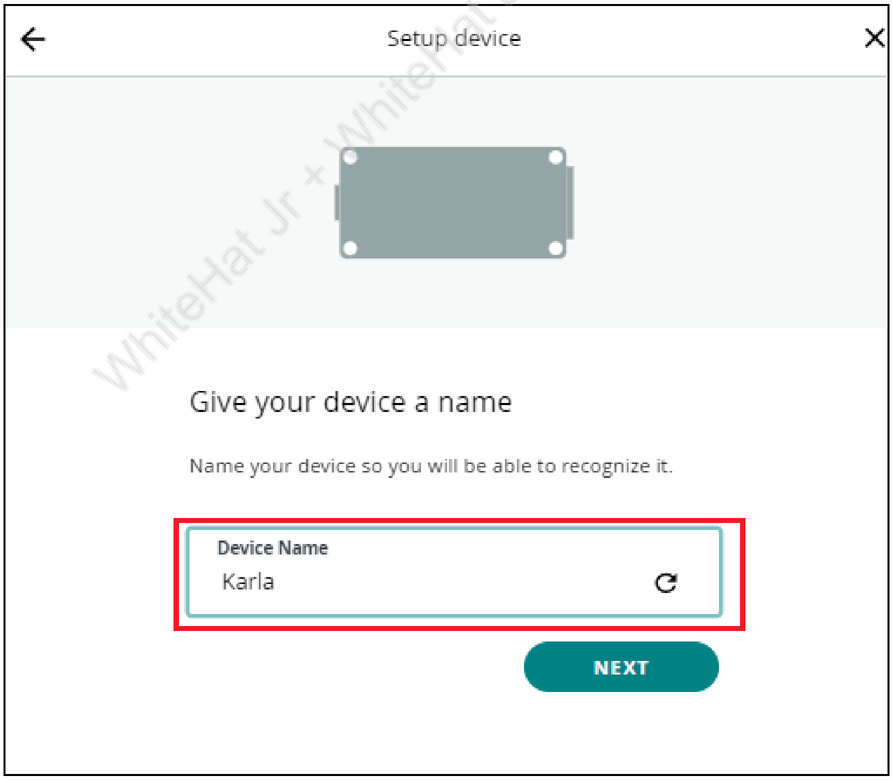
Select device type

Please select the device type and model you want to configure

○ ESP8266 ● ESP32 ○ LoRaWAN

ESP32 Wrover Module

d. Click **CONTINUE** and add a **Device Name.**

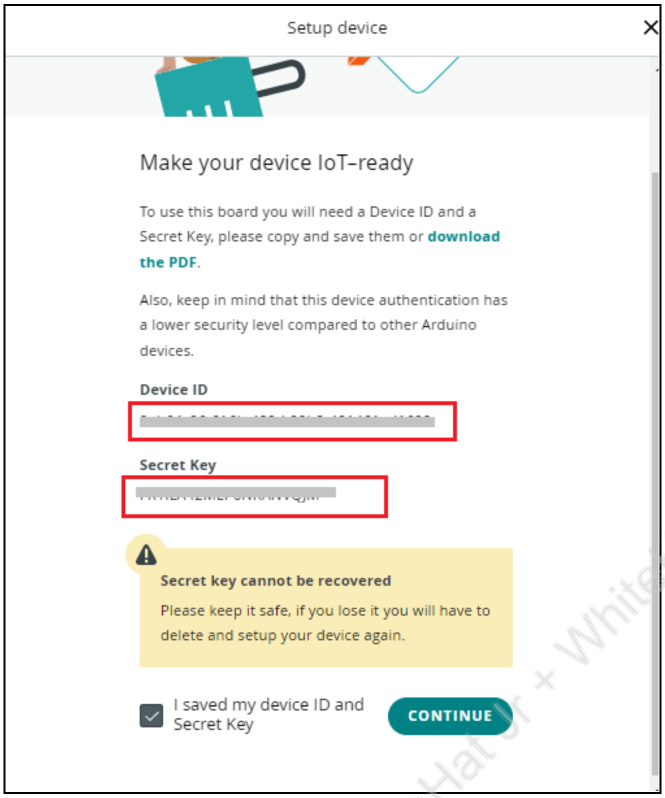


Give your device a name

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

Device Name: Karla

e. Now, it will show the **Device ID** and **Secret Key.** Copy the **Device ID** and **Secret Key.**Store the keys somewhere before proceeding with the next steps.



Setup device

Make your device IoT-ready

To use this board you will need a Device ID and a 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

[redacted]

Secret Key

[redacted]

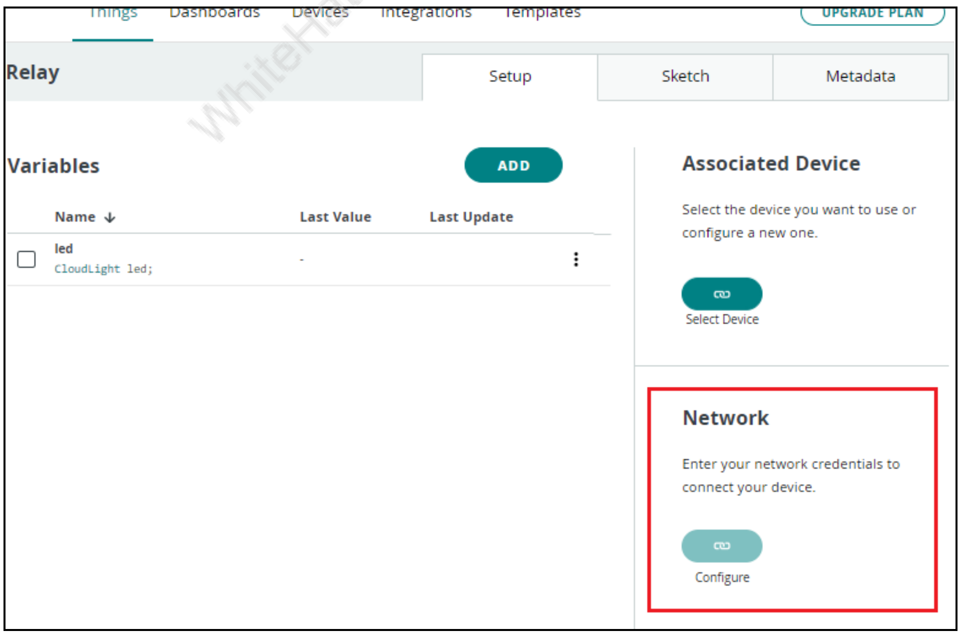
Secret key cannot be recovered

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

f. Click on **CONTINUE.**

12. Once the **Associated Device** is set, go to the Network section and click on the configure button.



Relay

Variables

□	Name ↓	Last Value	Last Update
□	led `CloudLight led;`	-	

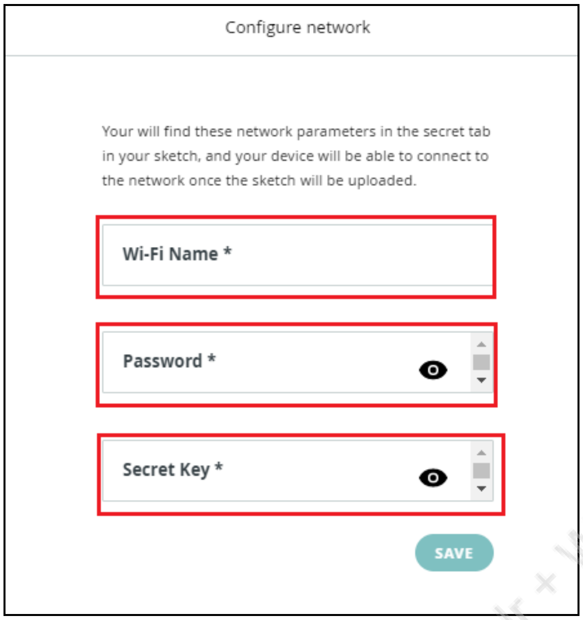
Associated Device

Select the device you want to use or configure a new one.

Network

Enter your network credentials to connect your device.

a. Add **Wi-Fi Name** and **Password.** Also, add the **Secret Key** generated while adding **Associated Device.**



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.

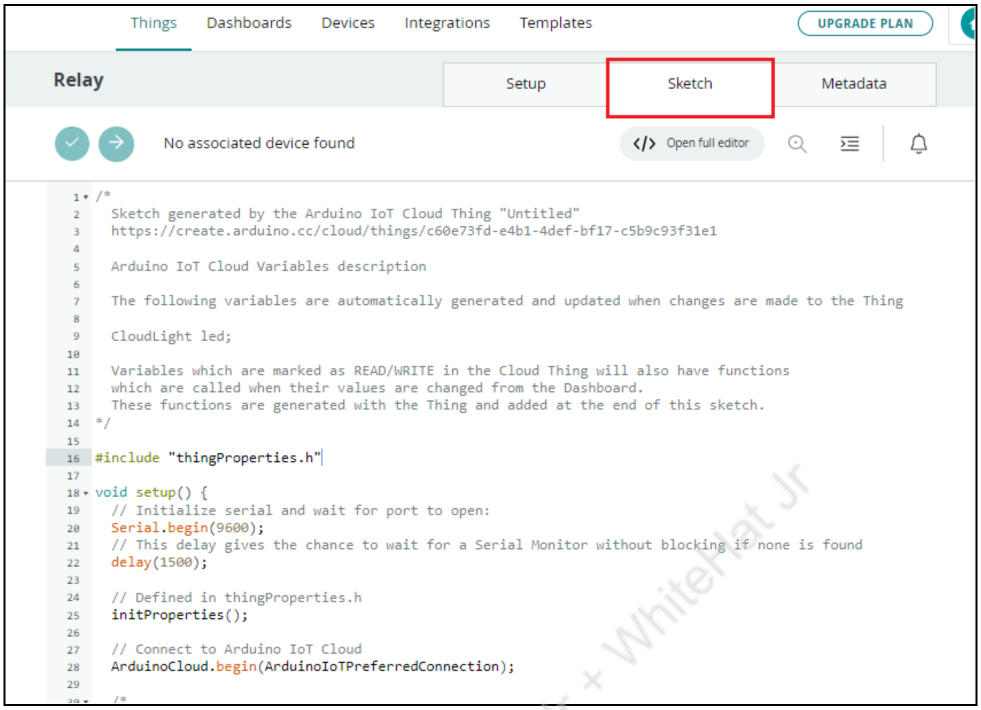
Wi-Fi Name *

Password *

Secret Key *

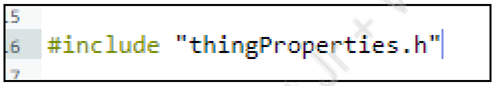
13. Let's write the program now

a. Click on the **Sketch** tab where the predefined code will automatically be added.



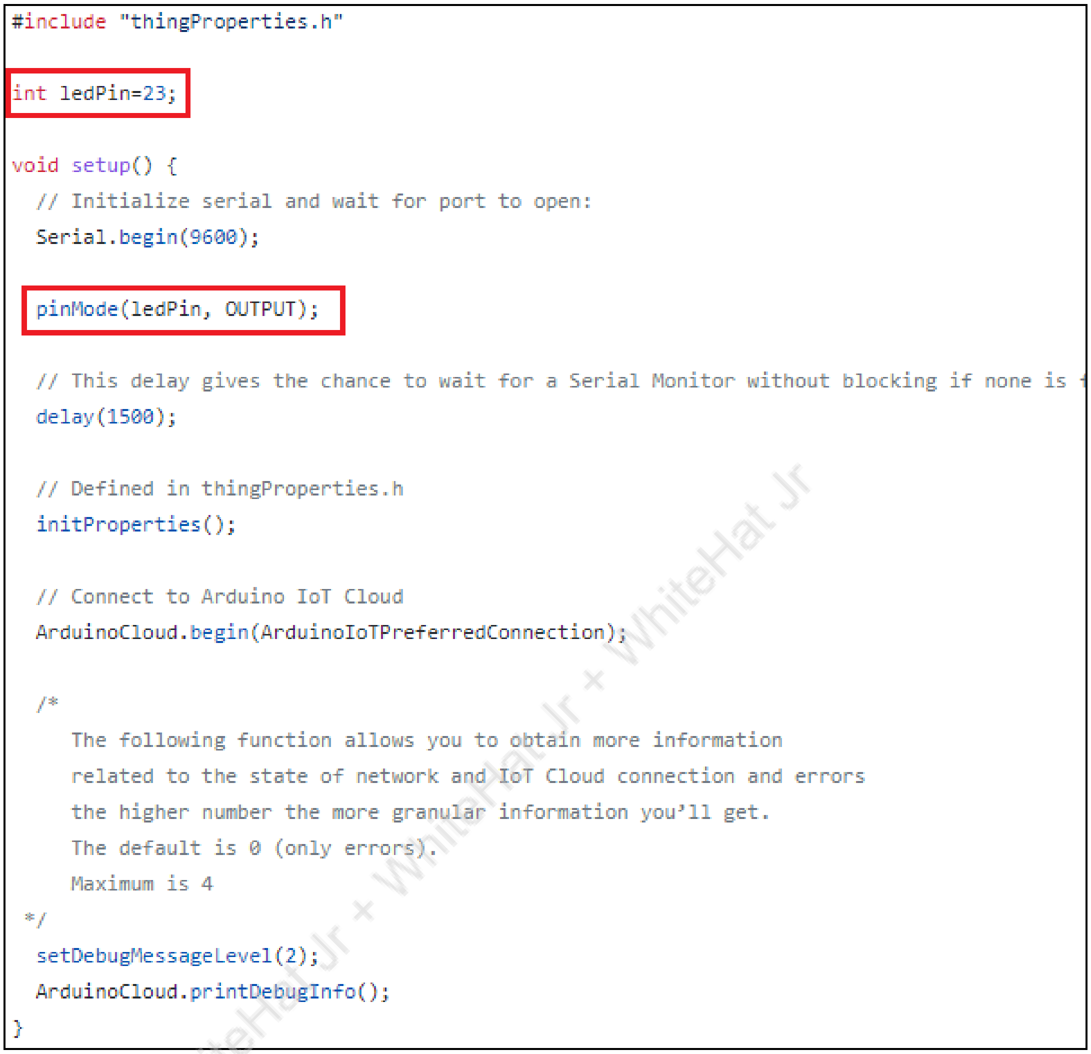
```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled"
  https://create.arduino.cc/cloud/things/c60e73fd-e4b1-4def-bf17-c5b9c93f31e1

  Arduino IoT Cloud Variables description

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

  CloudLight led;

  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);

  // Defined in thingProperties.h
  initProperties();

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

  /*
```

b. Let's observe the code. We will understand the code and add new functionalities to this project. Initially, **thingProperties.h** header file is included.



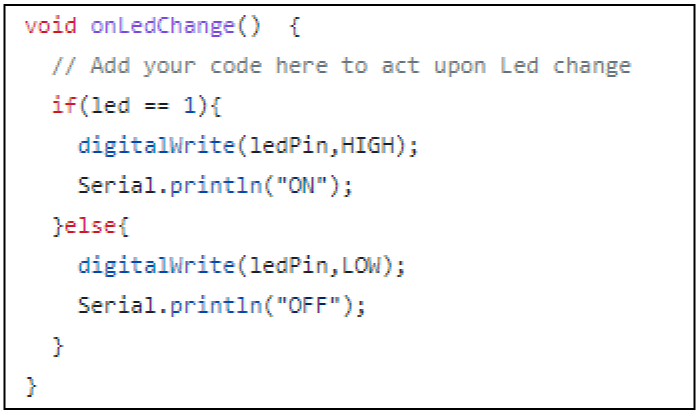
```
#include "thingProperties.h"
```

c. After that **setup()** method is defined. Here, we will define the **pinMode()** for pin number 23.



```
#include "thingProperties.h"

int ledPin=23;

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

  pinMode(ledPin, OUTPUT);

  // This delay gives the chance to wait for a Serial Monitor without blocking if none is
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

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

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you’ll get.
     The default is 0 (only errors).
     Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}
```

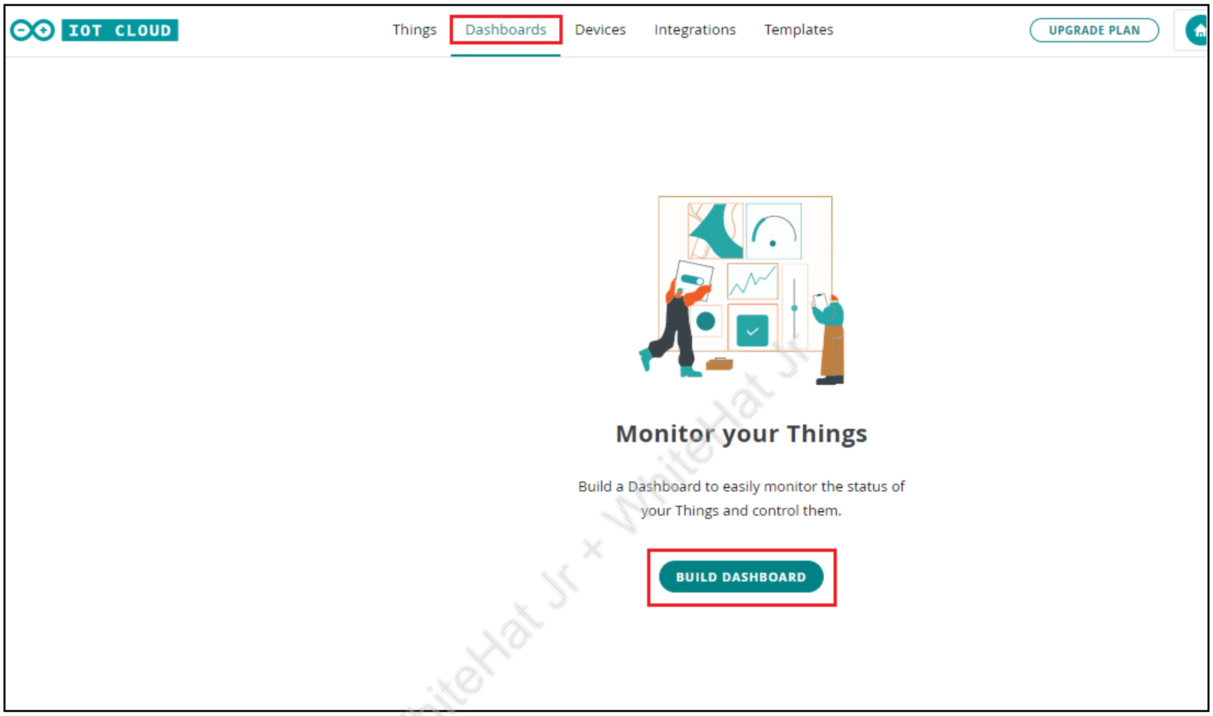
d. Now, find the **onLedChange()** method at the bottom. Write the code to set the pin 23 to HIGH and LOW depending on the **led** variable.



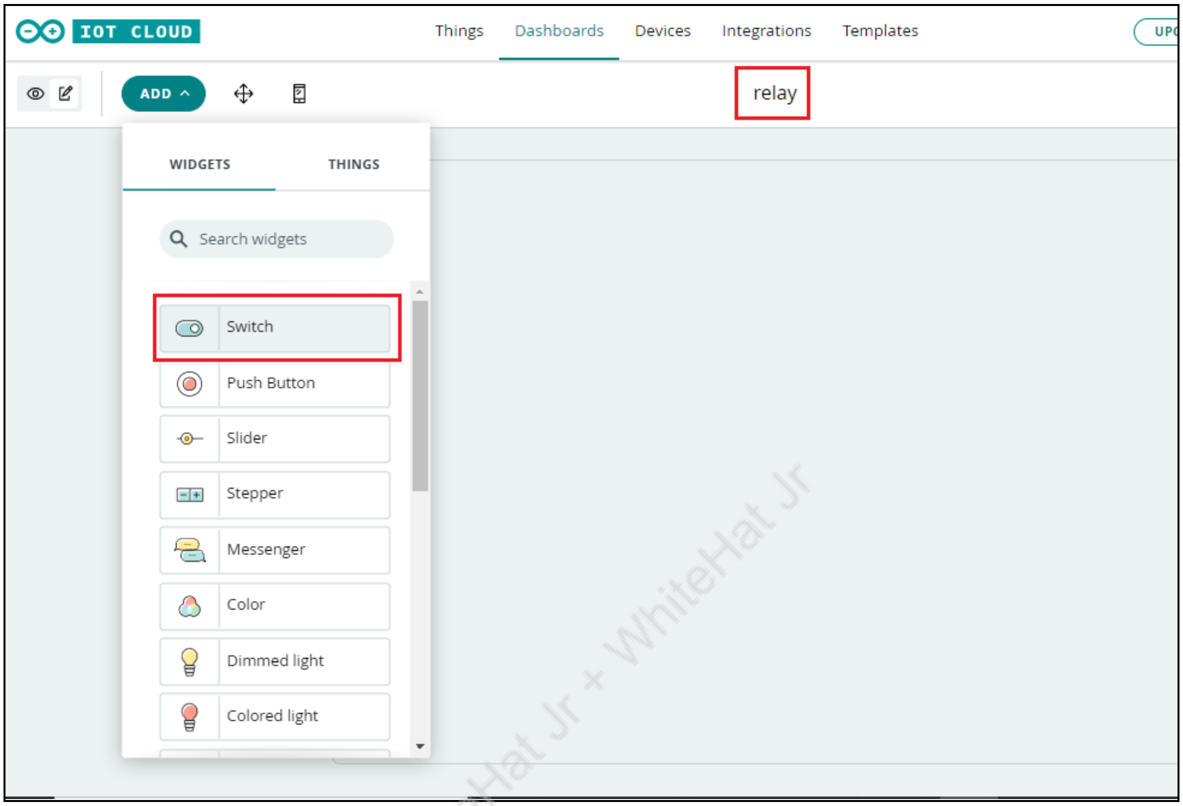
```
void onLedChange()  {
  // Add your code here to act upon Led change
  if(led == 1){
    digitalWrite(ledPin,HIGH);
    Serial.println("ON");
  }else{
    digitalWrite(ledPin,LOW);
    Serial.println("OFF");
  }
}
```

14. Add the switch to control the **led** variable:

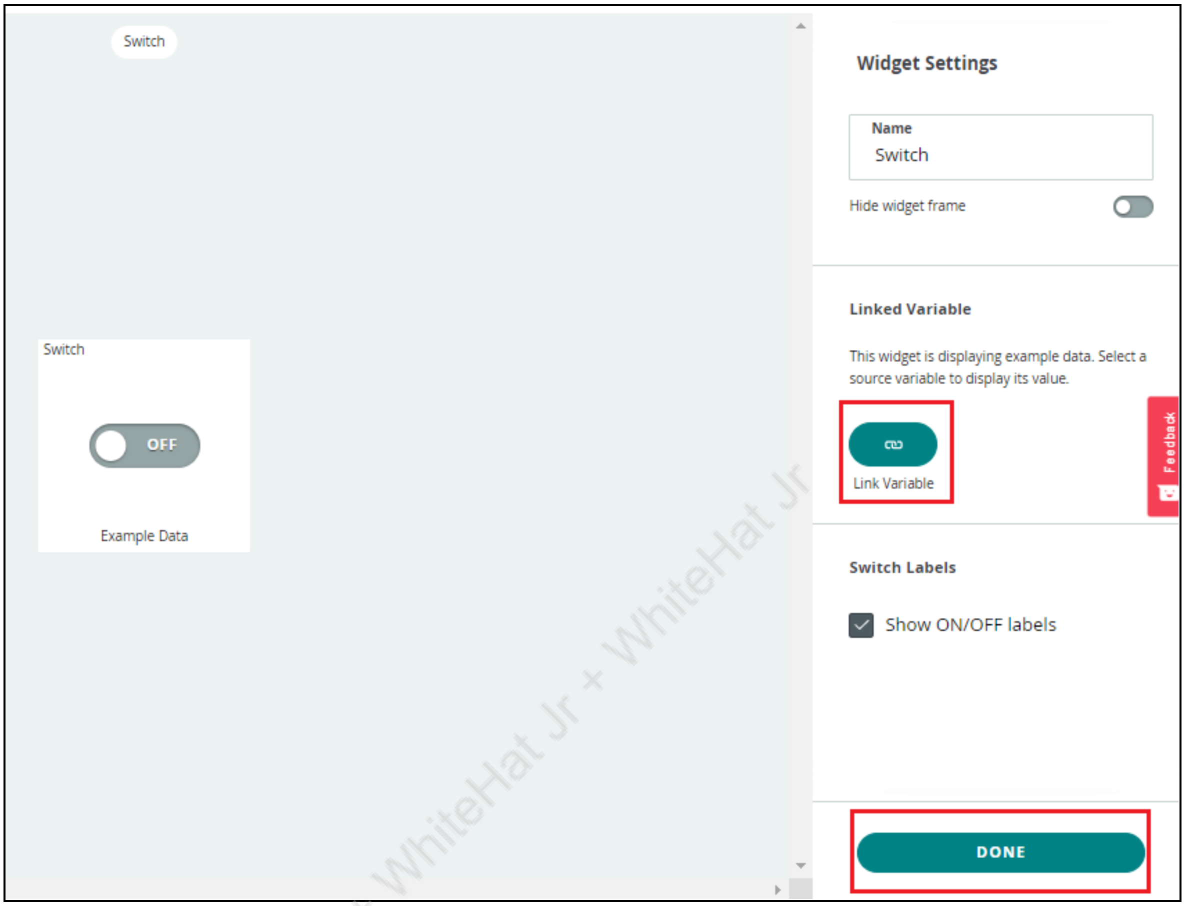
   a. Go to the **Dashboards** tab and click on the **BUILD DASHBOARD** button.



   b. Add a name and add a switch to the dashboard.



c. After that, let’s link the **led** variable to the **switch**. Click on the **Link Variable** button and click on **DONE.**



Widget Settings

Name
Switch

Hide widget frame

Linked Variable

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

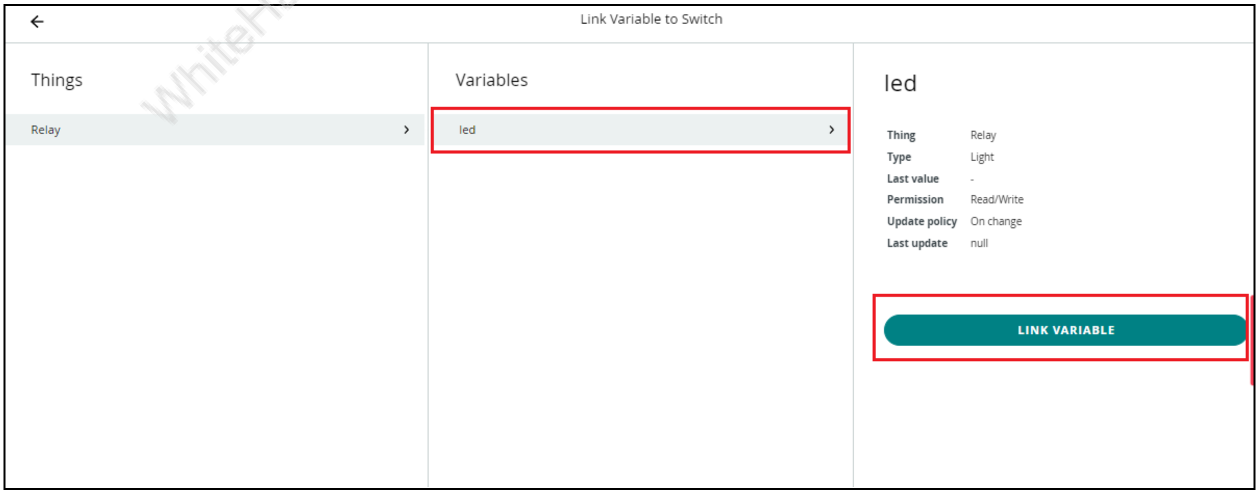
Link Variable

Switch Labels

☑ Show ON/OFF labels

DONE

d. Select the **led** variable and click on **LINK VARIABLE.**



Link Variable to Switch

Things
Relay ›

Variables
led ›

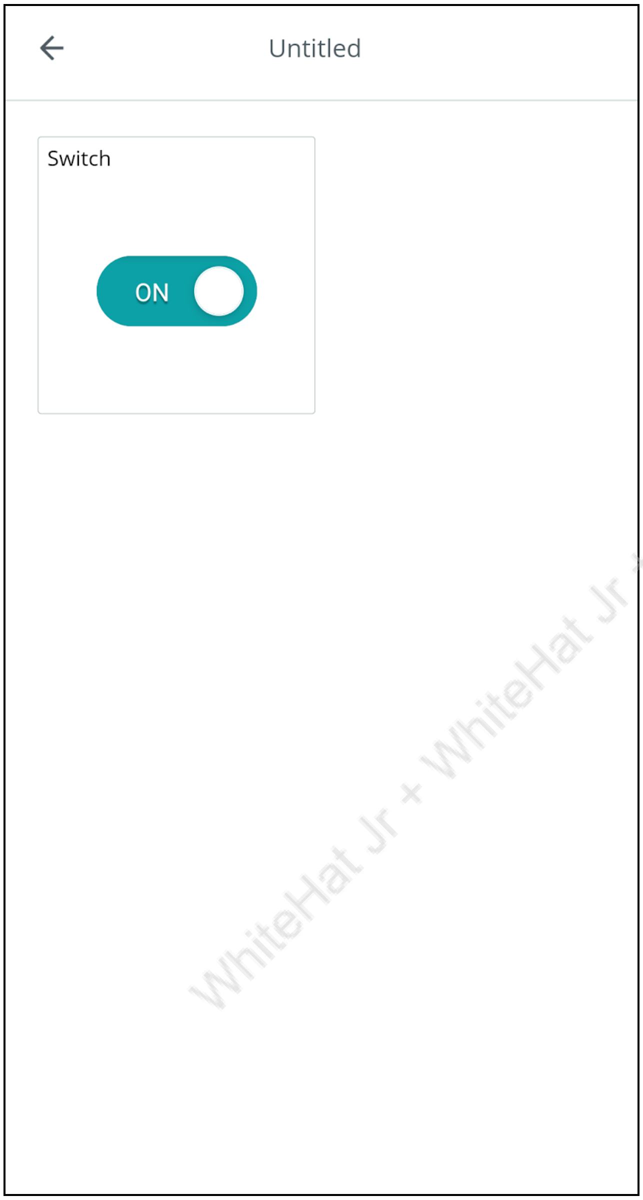
led

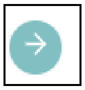
Thing	Relay
Type	Light
Last value	-
Permission	Read/Write
Update policy	On change
Last update	null

LINK VARIABLE

15. After that, download the Arduino Iot Cloud Remote app on your phone from the playstore.

Once the app is downloaded, login with the same credentials to control the device from your phone now.



16. Go back to things again, open your project and go to **Sketch**. Upload the sketch to your **ESP32** board by clicking on this button .
17. Once it is uploaded, go to the **Dashboard** and control the light with the switch.

What's NEXT?

In the next class, we will learn about the **Weather Monitoring System.**

Expand Your Knowledge

To know more about **Relays** click here.