WhiteHat Jr
Live online Coding for Kids

Topic	INTRODUCTION TO RELAY	
Class Description	**Students will be introduced to Relay and they will operate appliances with Relay using Arduino IoT Cloud.**	
Class	**PRO C250**	
Class time	**50 mins**	
Goal	- Introduction to Relay - Arduino IoT Cloud	
Resources Required	- Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone - Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm-Up** **Teacher-Led Activity** **Student-Led Activity** **Wrap-Up**	**10 mins** **20 mins** **20 mins** **05 mins**

WARM-UP SESSION - 10 mins	
Teacher Action	**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Following are the WARM-UP session deliverables:**	**ESR**: Hi, thanks! Yes, I am excited about it! Click on the slide show tab

● Greet the student. ● Revision of previous class activities. ● Quizzes.	and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
Activity Details **Following are the session deliverables:** ● Appreciate the student. ● Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY ● **Introduction to Relay**	
Teacher Action	**Student Action**
Note: In this class we need to operate an End device which works on Alternating current, so while performing activity students and teachers must be careful. *We need to use one end devices, that can be anything for example: A bulb with a holder and plug, a table lamp or mosquito repellent machine.*	**ESR:** Varied!
Any doubts from the last session? *If there is any doubt, the teacher will clarify* Let's learn something new today.	**ESR:** Varied!

Can we blink a lamp/led automatically or just via giving instructions. Ok, tell me how 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. Great! And what if we want to turn them on and off automatically ? Is there any device that can do wonders like this! Fortunately, to automate your switches we must know a relay! Let's understand how a relay works!	**ESR:** Varied!
Teacher Click on Teacher Activity1	Student clicks on Student Activity 1
"A relay is an electromagnetic switch operated by a relatively small electric current that can turn on or off a much larger electric current." The heart of a relay is an **electromagnet** **Electromagnet:** A coil of wire that becomes a temporary magnet when electricity flows through it. Basically in relay two circuits are there, one for small	

electric current and the second for large current First circuit uses a small electric current to trigger a much larger current for the second circuit. When power flows through the first circuit, it activates the electromagnet generating a magnetic field that attracts a contact and activates the second circuit When the power is switched off, a spring pulls the contact back up to its original position, switching the second circuit off again. This is an example of a "normally open" (NO) relay: the contacts in the second circuit are not connected by default and switch on only when a current flows through the magnet. Other relays are "normally closed" (NC; the contacts are connected so a current flows through them by default) and switch off only when the magnet is activated, pulling or pushing the contacts apart.	
Purpose of Relay: To convert small electric current into high electric current, like DC (Direct Current) into AC (Alternating Current)	
So today our task is to make a circuit with Relay. In your WHJR kit there's black little box with a plug. In that black box relay is inside as the relay results in high output current so it's covered like this. Usually Relay circuit has 6 terminals: NC(Normally closed) NO (Normally open) Common at one side and other side VCC, GND,INPUT	

But in the black box you will see three female jumpers wire pointed out i.e basically VCC (Red) GND, (Black), INPUT(Yellow)

Other three pins are basically your Plug pins. So we just need to plug in any Ac device.

This AC device can be your Lamp, Mosquito Repellent machine, or a Bulb with plug and holder.

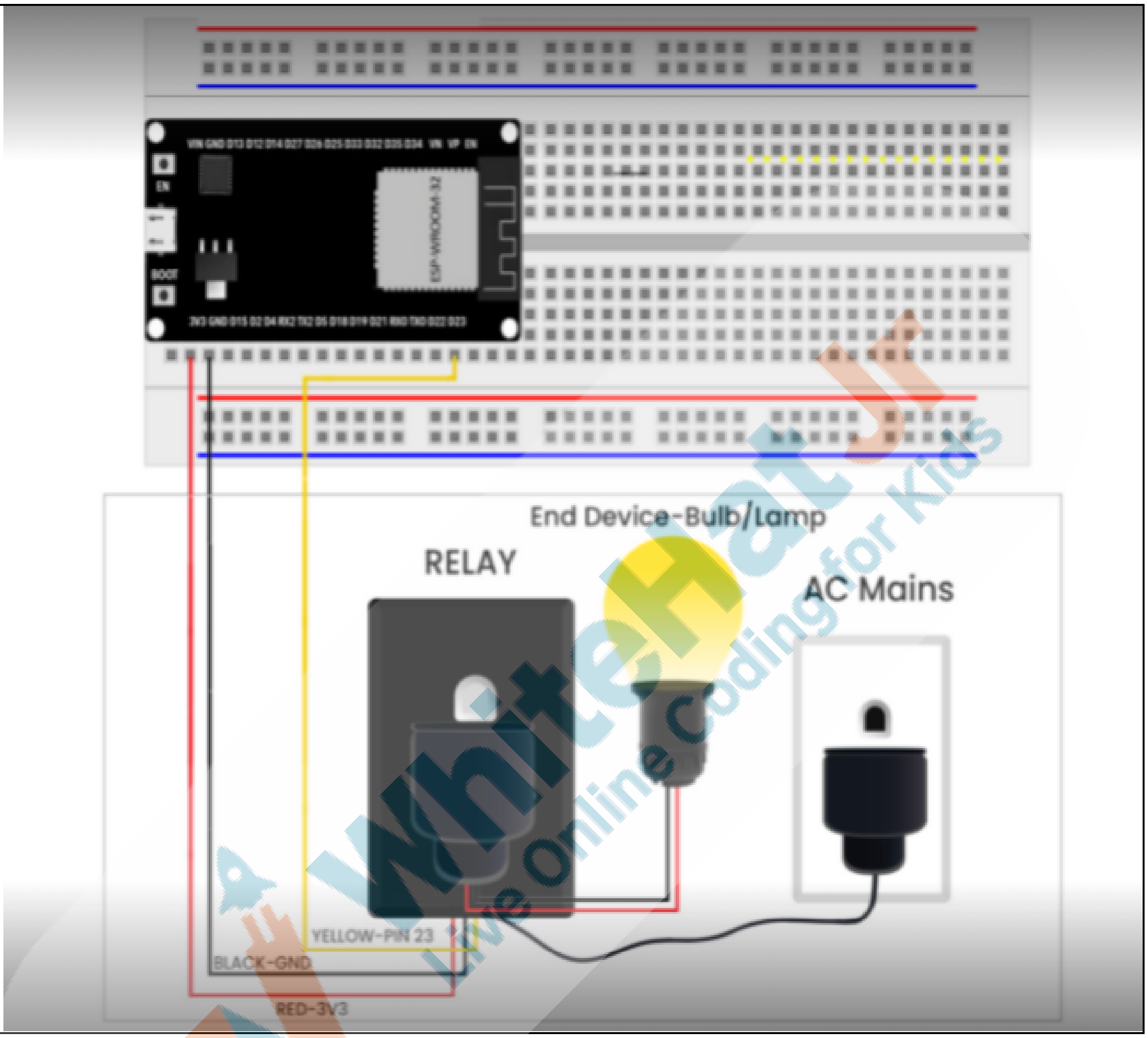
Step -1: Gather the material from the IoT kit:

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

Step -2: Let's do connections:

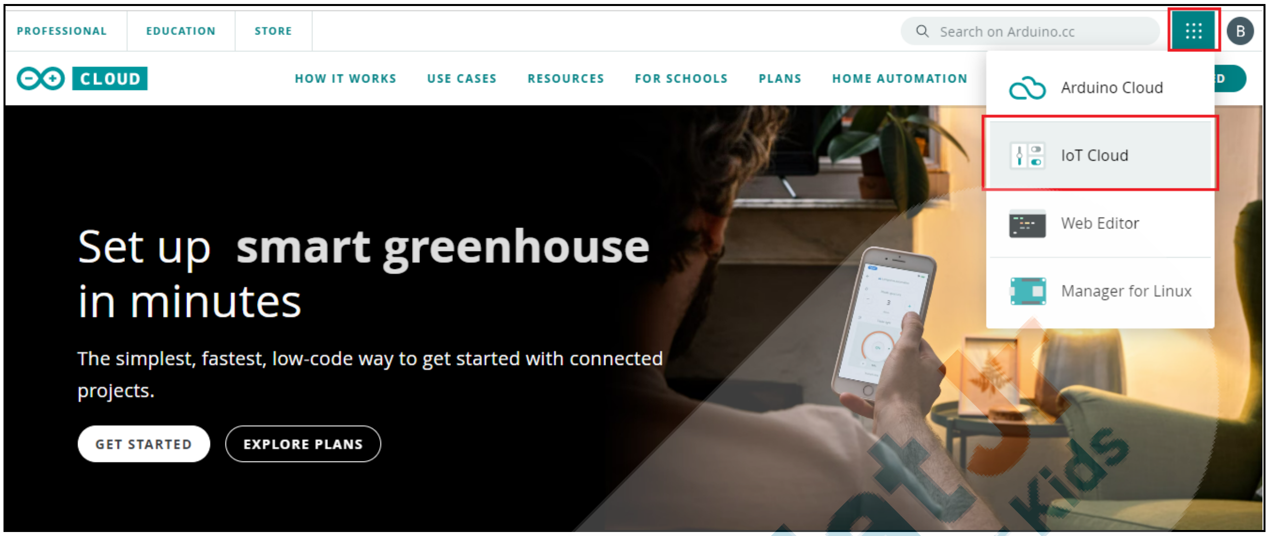
Note: Follow the same pins number as mentioned below to connect the sensor with ESP32

Relay Pins	**Wiring Connections**
VCC(Red)	Connect with **3V3 PIN** of the ESP32
GND(Black)	Connect with **GND** of the ESP32
INPUT(Yellow)	Connect with GPIO **PIN 23**

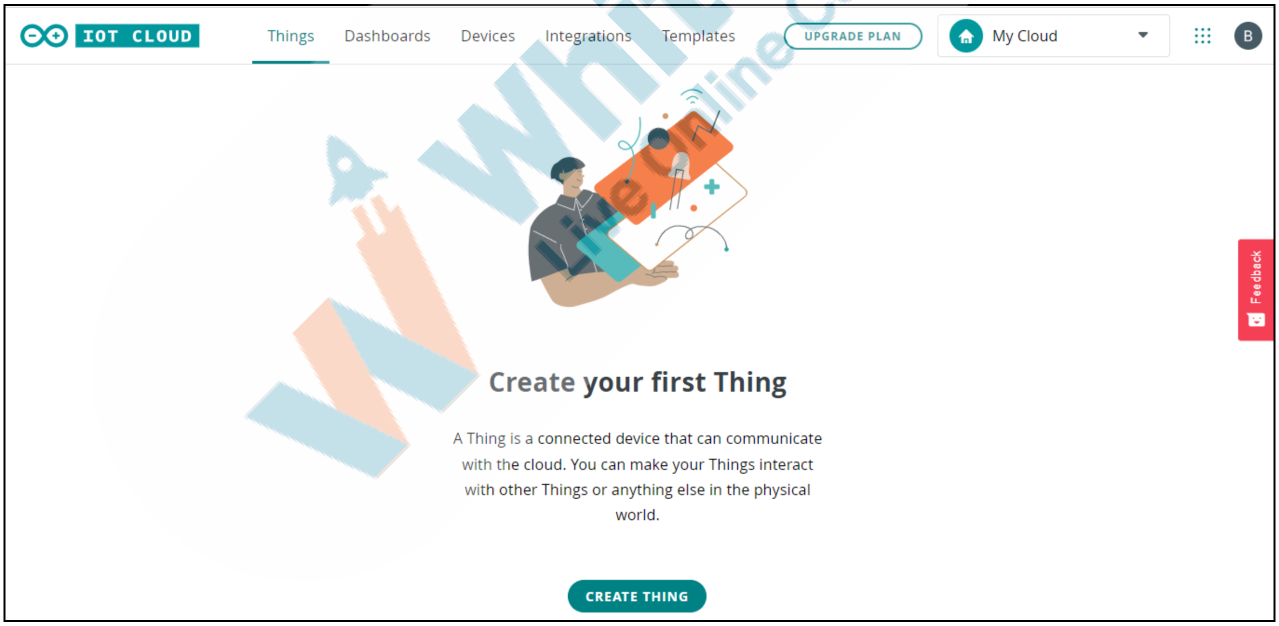


Before writing a program we need to do some more steps We want to make a smart system where on our command electronic appliances will be turned "ON" or "OFF" The first and foremost step is to create an account on the Arduino IoT Cloud.	
I want you to work on the project on your end.	

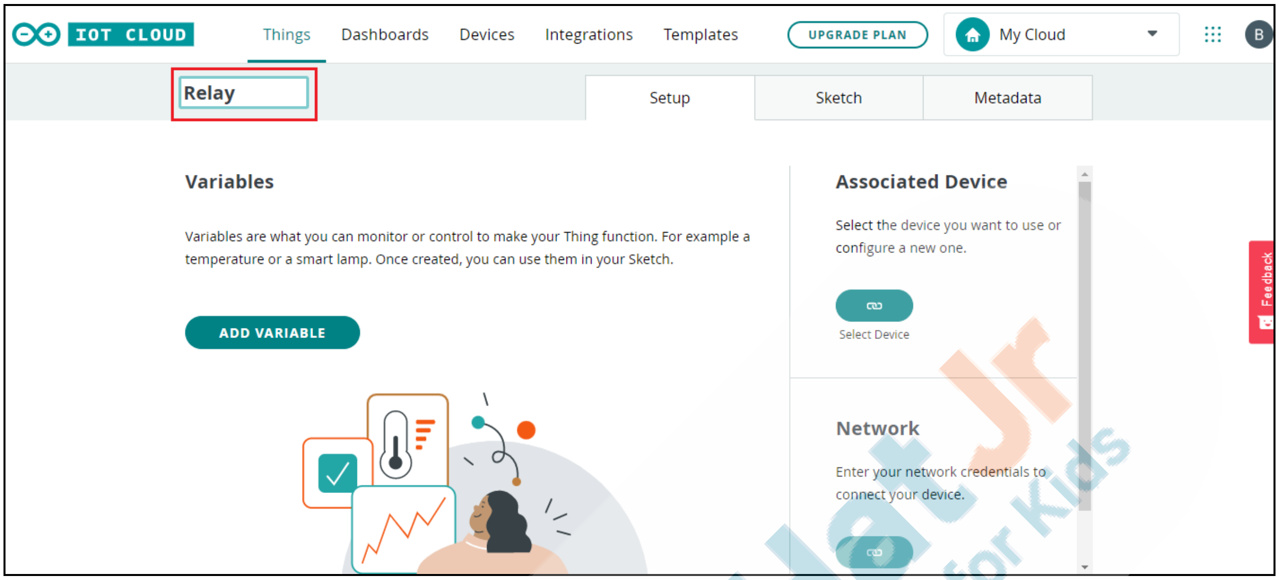
Are you ready? Let's start!	**ESR:** Yes.
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
STUDENT-LED ACTIVITY - 20 mins	
● **Ask the student to press the ESC key to come back to the panel.** ● **Guide the student to start Screen Share.** ● **The teacher gets into Full Screen.**	
Student Initiates Screen Share	
ACTIVITY ● **Student Activity description (in bullet points).**	
Teacher Action	**Student Action**
Teacher asks the student to open Student Activity 2. Teacher helps the student to create an account on https://cloud.arduino.cc/	Student clicks on Student Activity 2
1. Go to - https://cloud.arduino.cc/	
2. Go to the top right hand side of your screen and click on the ⠿ option.Then, click on **IoT Cloud.**	



3. First we need to create a "**thing**" for our internet of things. So, click on the **"CREATE THING"** button.


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
My Cloud
B
Feedback
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

4. Now, you will add a name to this project.



IOT CLOUD Things Dashboards Devices Integrations Templates UPGRADE PLAN My Cloud B

Relay

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.

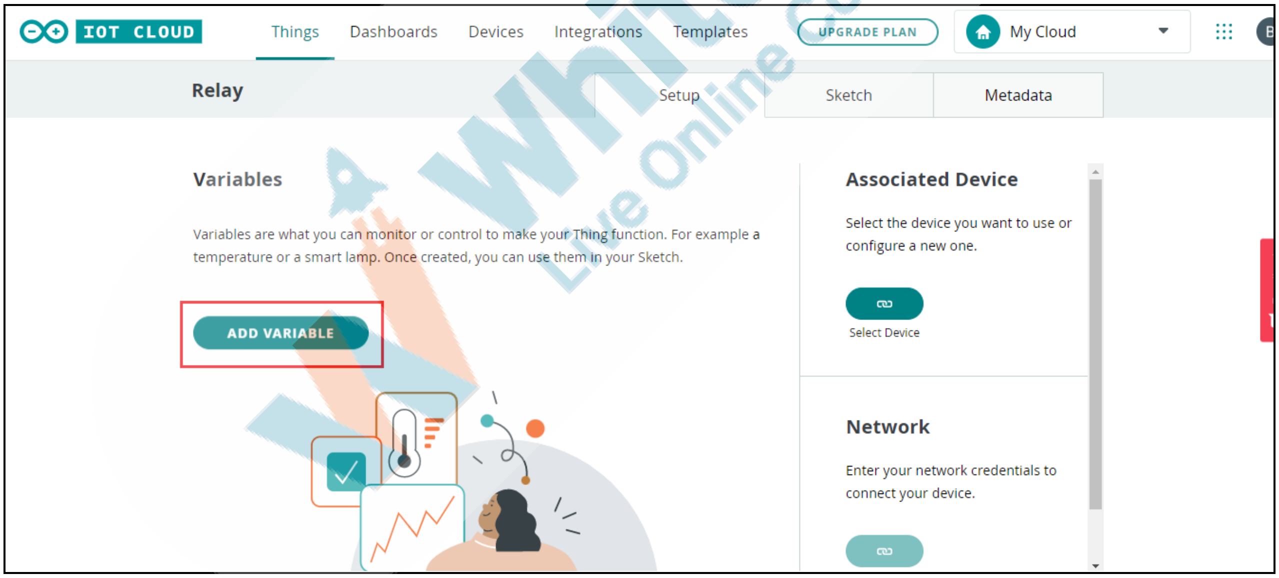
Select Device

Network

Enter your network credentials to connect your device.

Feedback

5. After that we need to add a **variable** named **led**. This variable will hold the status of our switch.



IOT CLOUD Things Dashboards Devices Integrations Templates UPGRADE PLAN My Cloud B

Relay

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

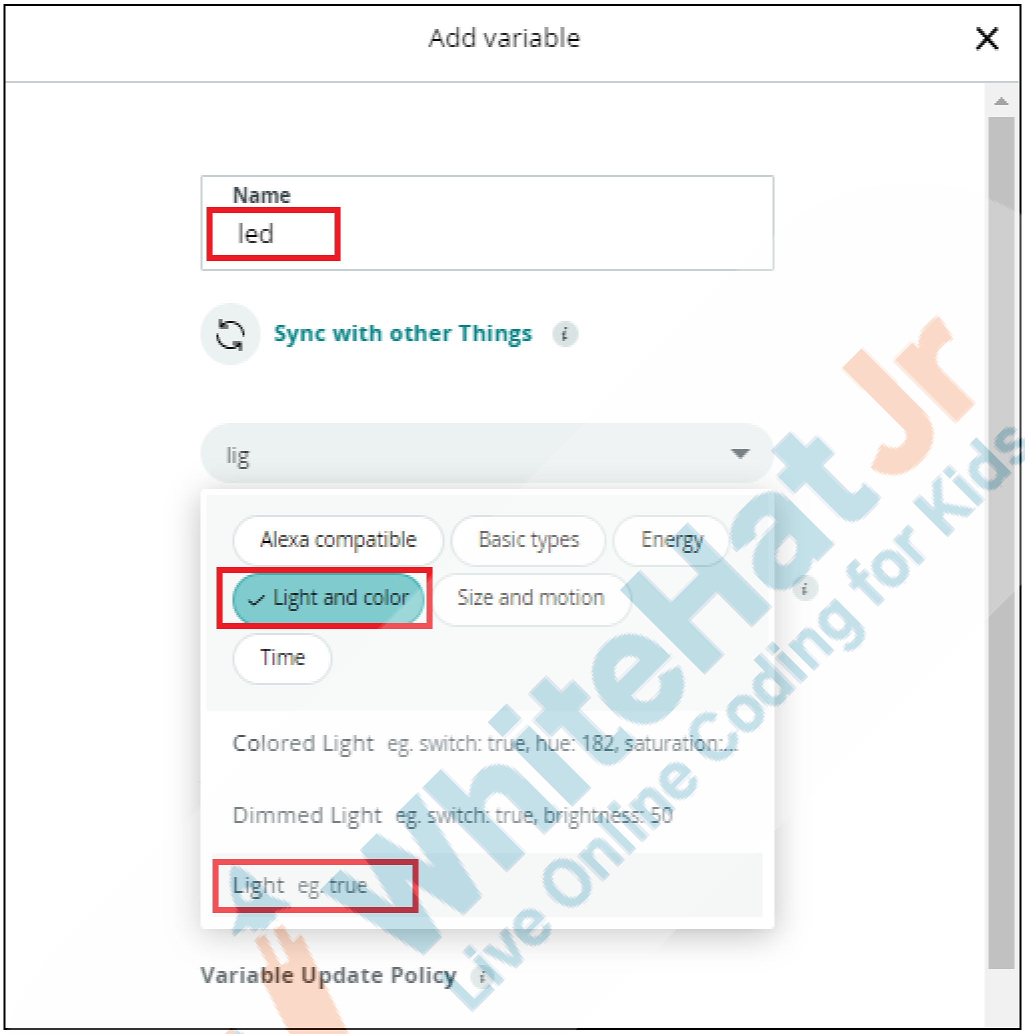
Select the device you want to use or configure a new one.

Select Device

Network

Enter your network credentials to connect your device.

6. Change the name of the variable to **led.** Select variable type as "**Light**".



Add variable

Name
led

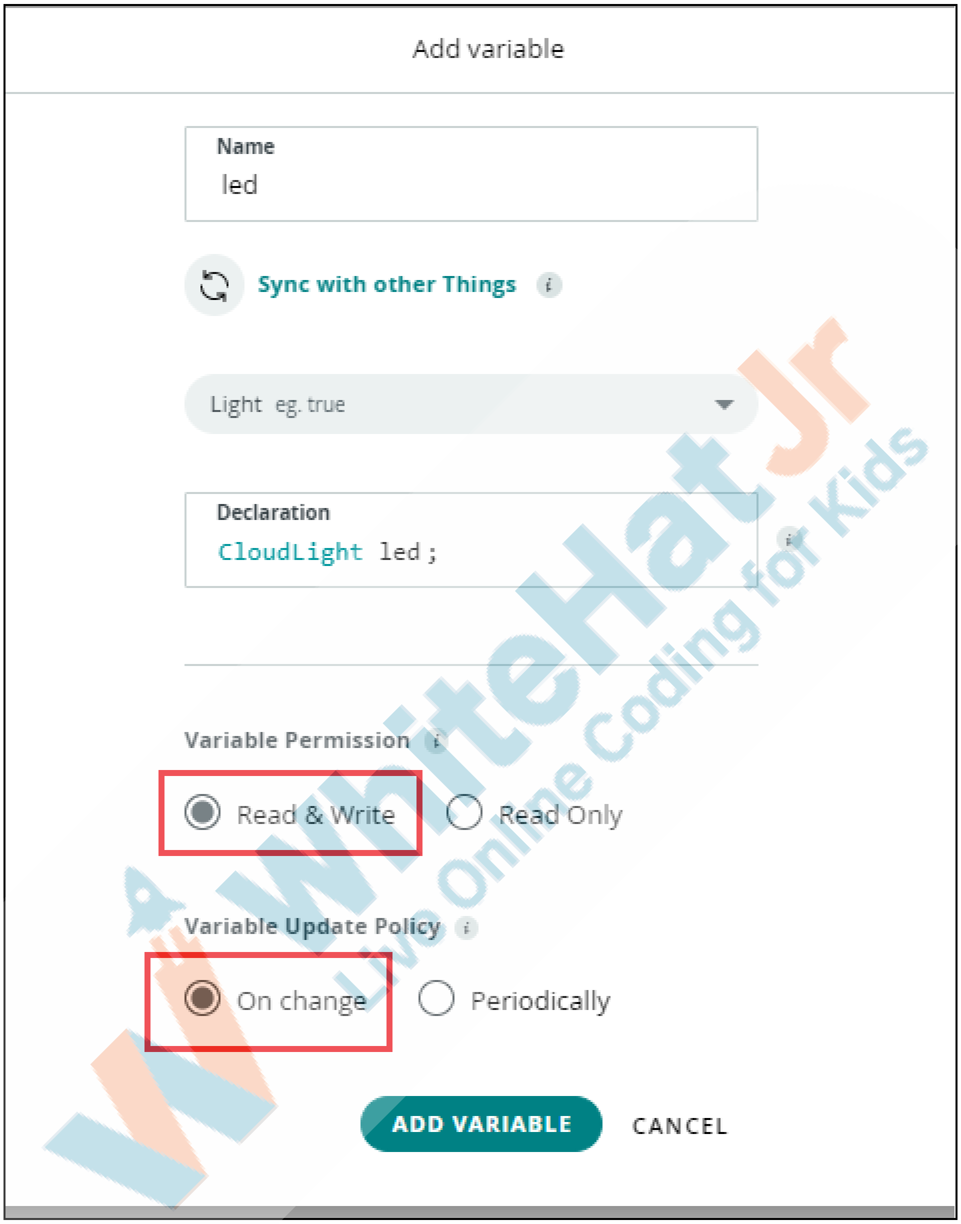
lig

Colored Light eg. switch: true, hue: 182, saturation:...

Dimmed Light eg. switch: true, brightness: 50

Light eg. true

7. As we want to switch it on and off in the program, we will assign **Read & Write** permissions to this variable.
   Also, we would want to update this variable with a switch. Hence, the **Variable Update Policy** will be set to **On Change.**



Add variable

Name
led

Sync with other Things

Light eg. true

Declaration
```
CloudLight led ;
```

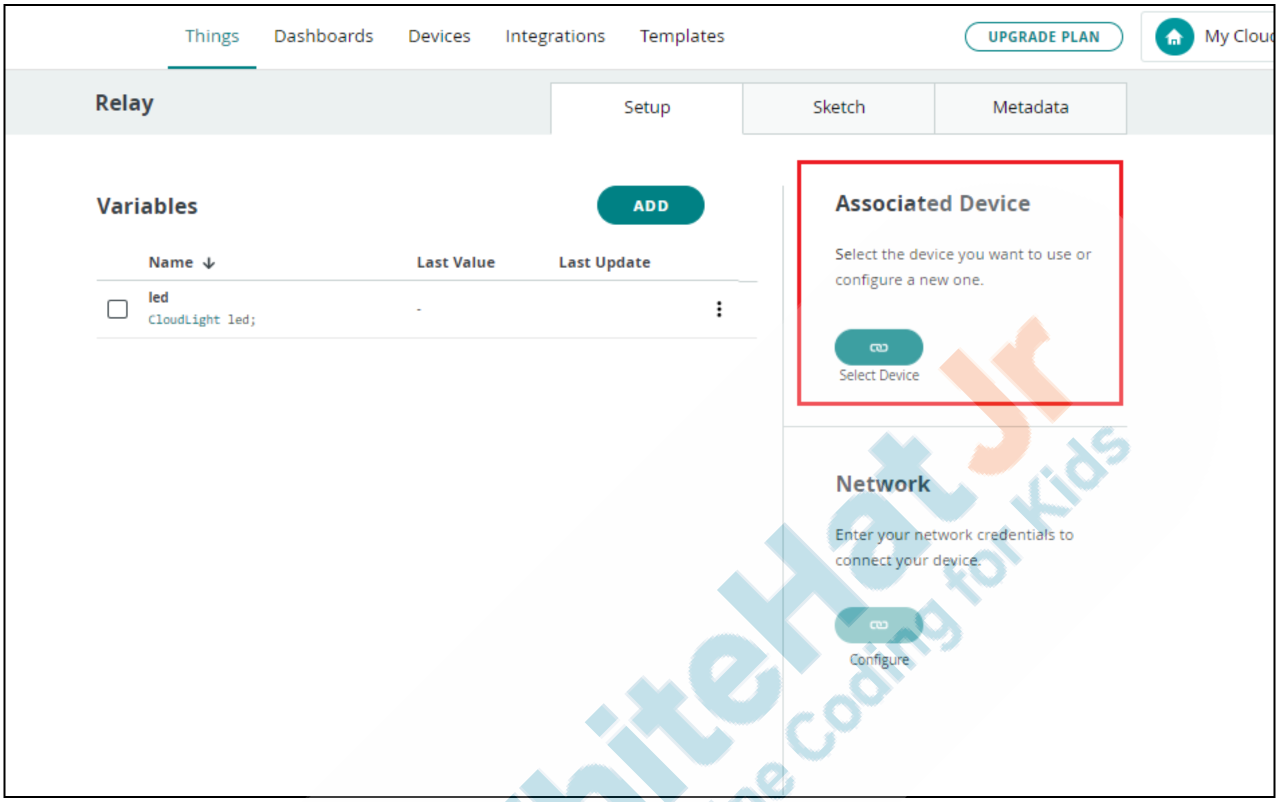
Variable Permission

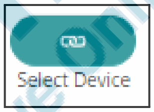
● Read & Write ○ Read Only

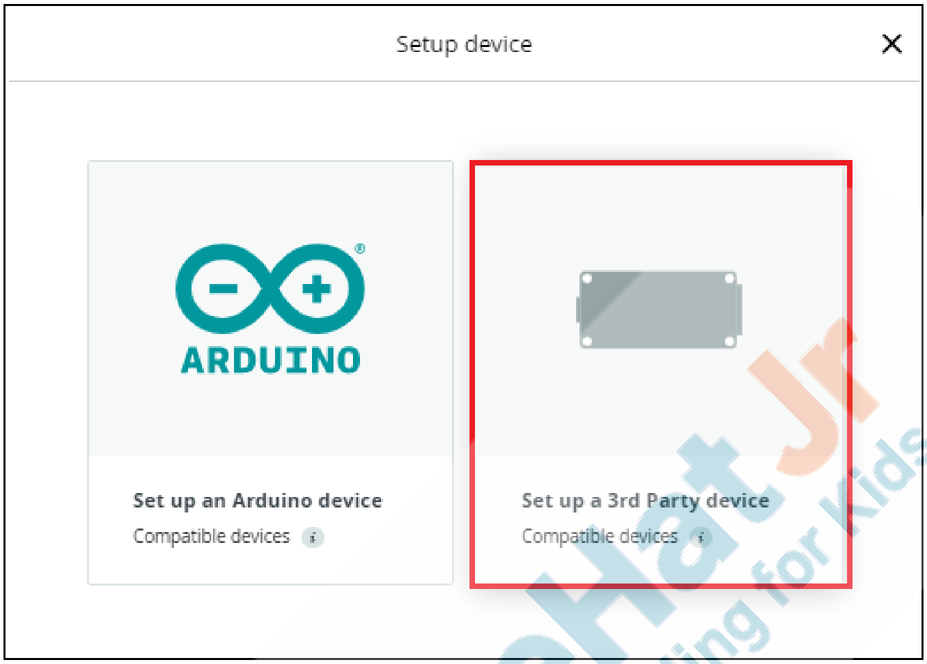
Variable Update Policy

● On change ○ Periodically

8. Now, we will set up the **Associated Device**.



a. Click on the **Select Device**  button.
b. Click on **Set Up a 3rd Party device**.

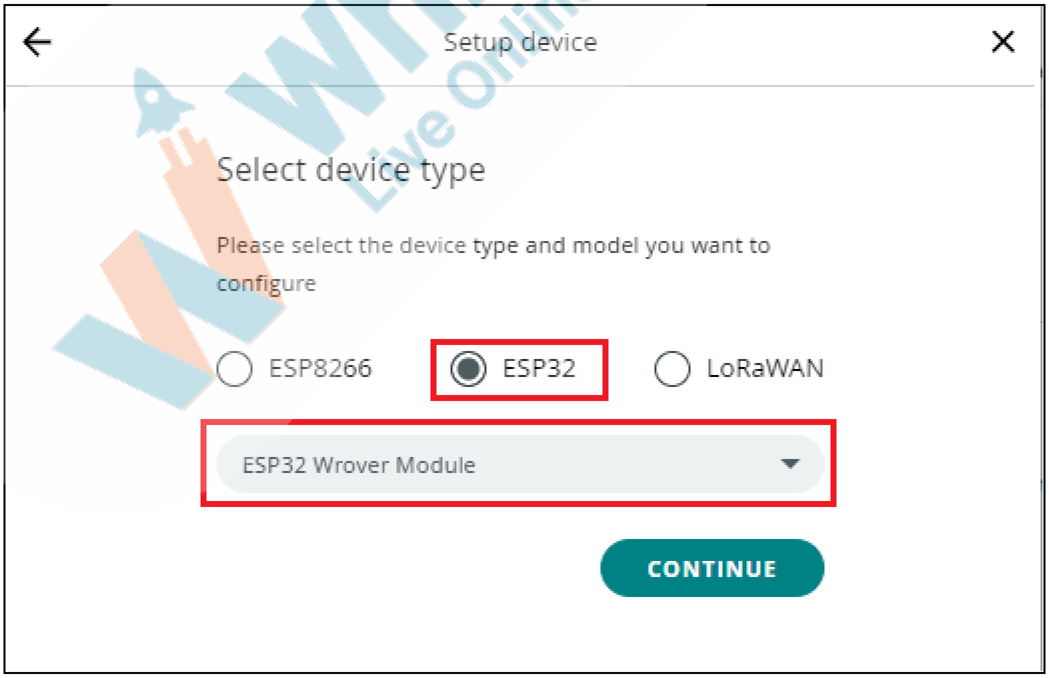


Setup device

Set up an Arduino device
Compatible devices

Set up a 3rd Party device
Compatible devices

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



Setup device

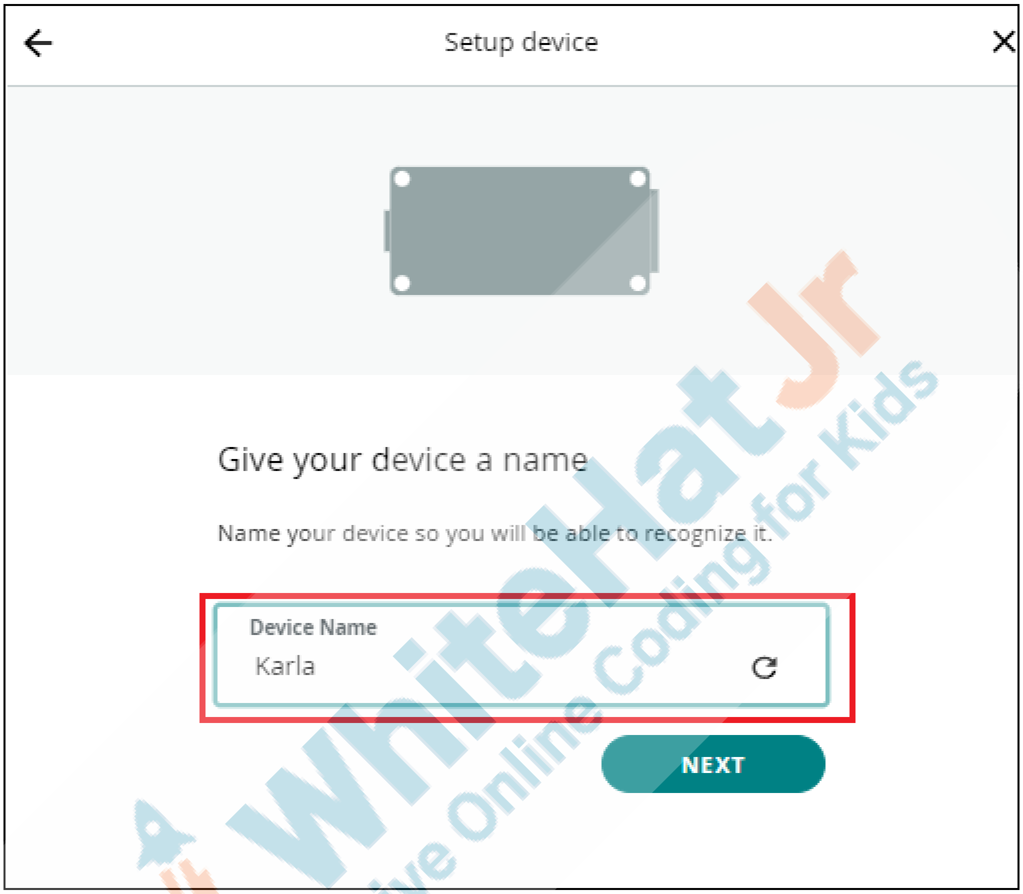
Select device type

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.**



Setup device

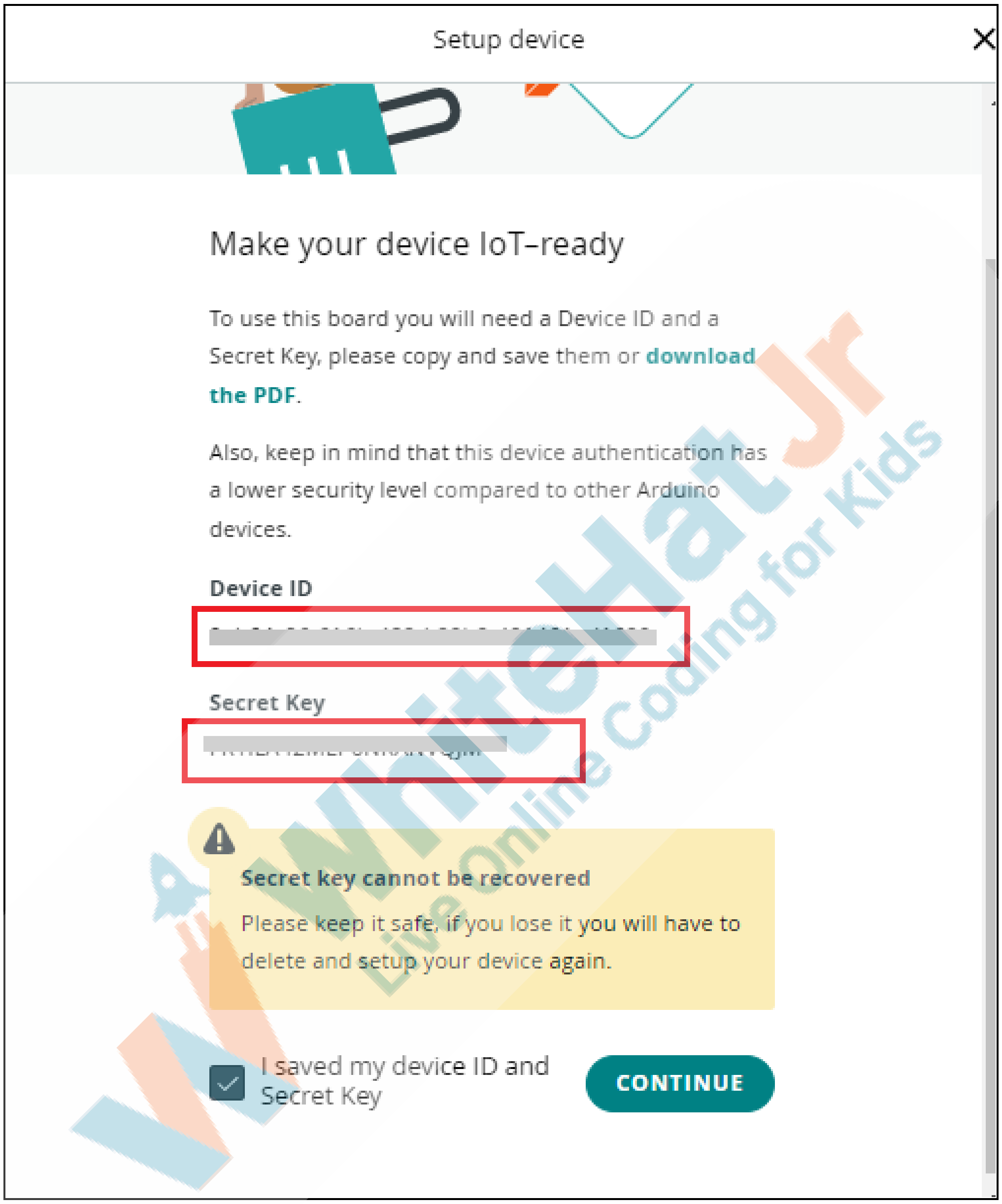
Give your device a name

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**. Make sure you store it 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]

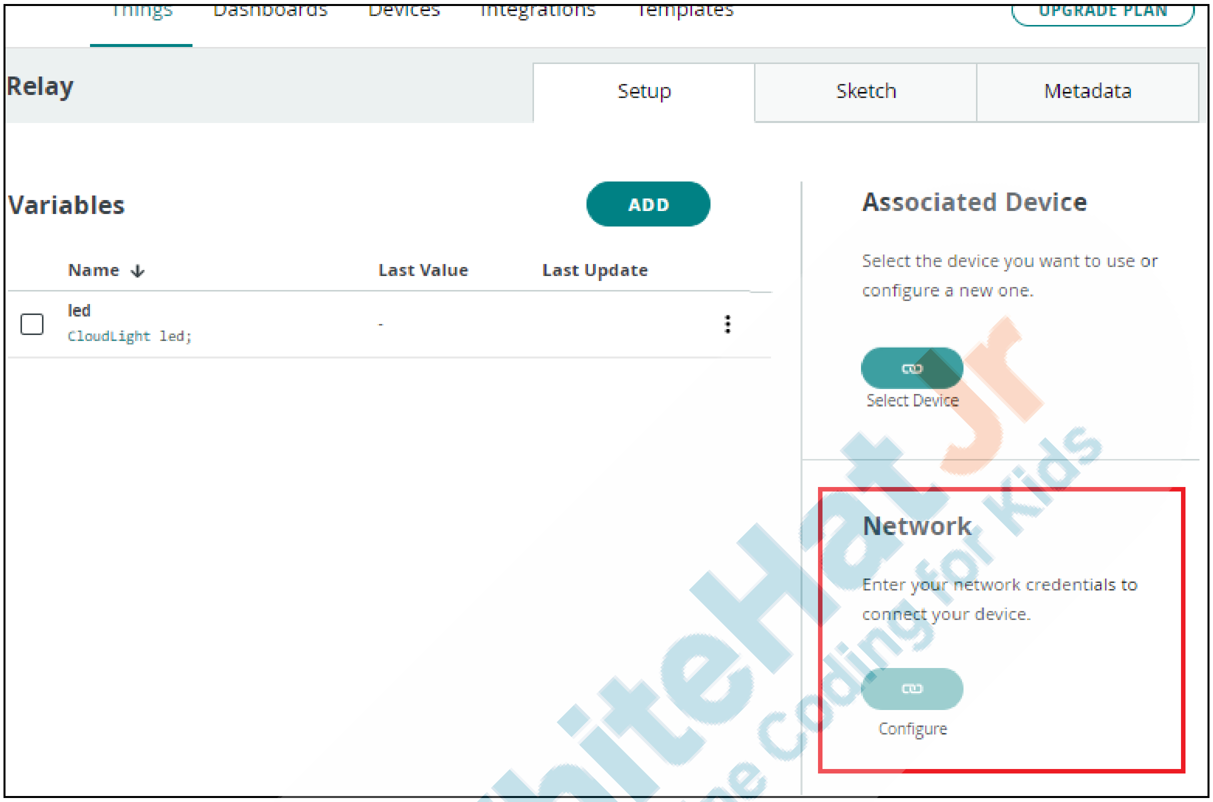
Secret key cannot be recovered

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.**

9. 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;	-	

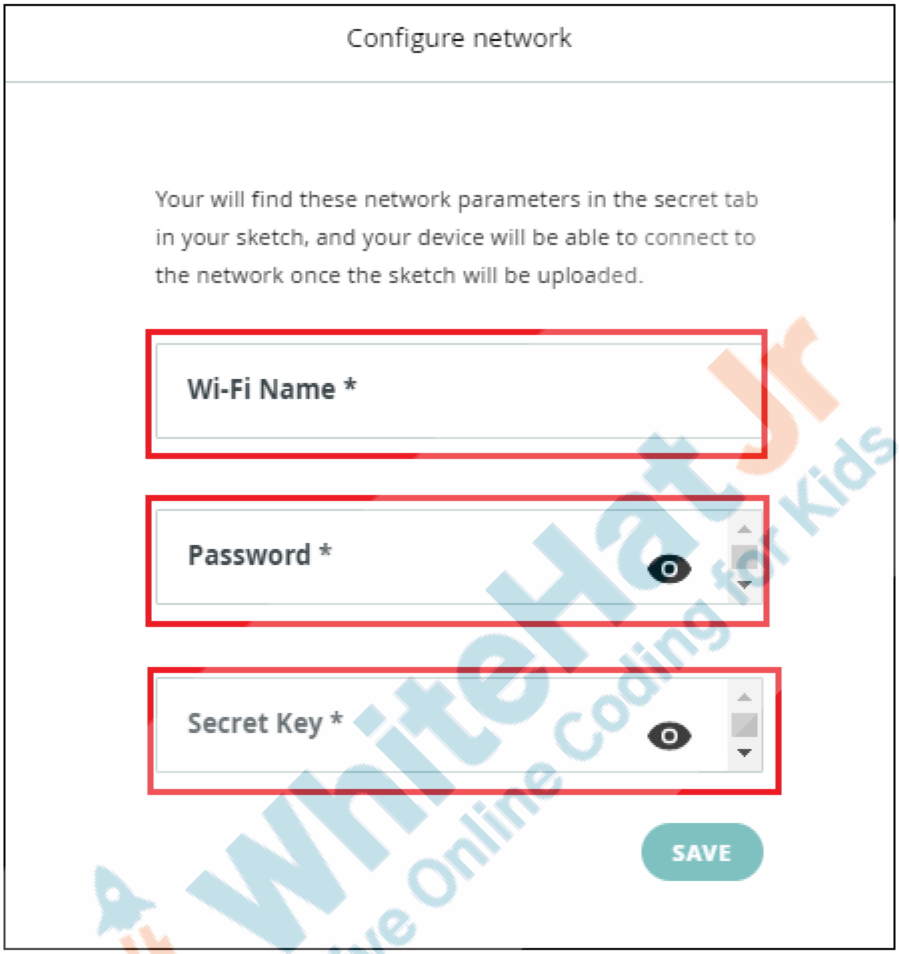
Associated Device

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**.



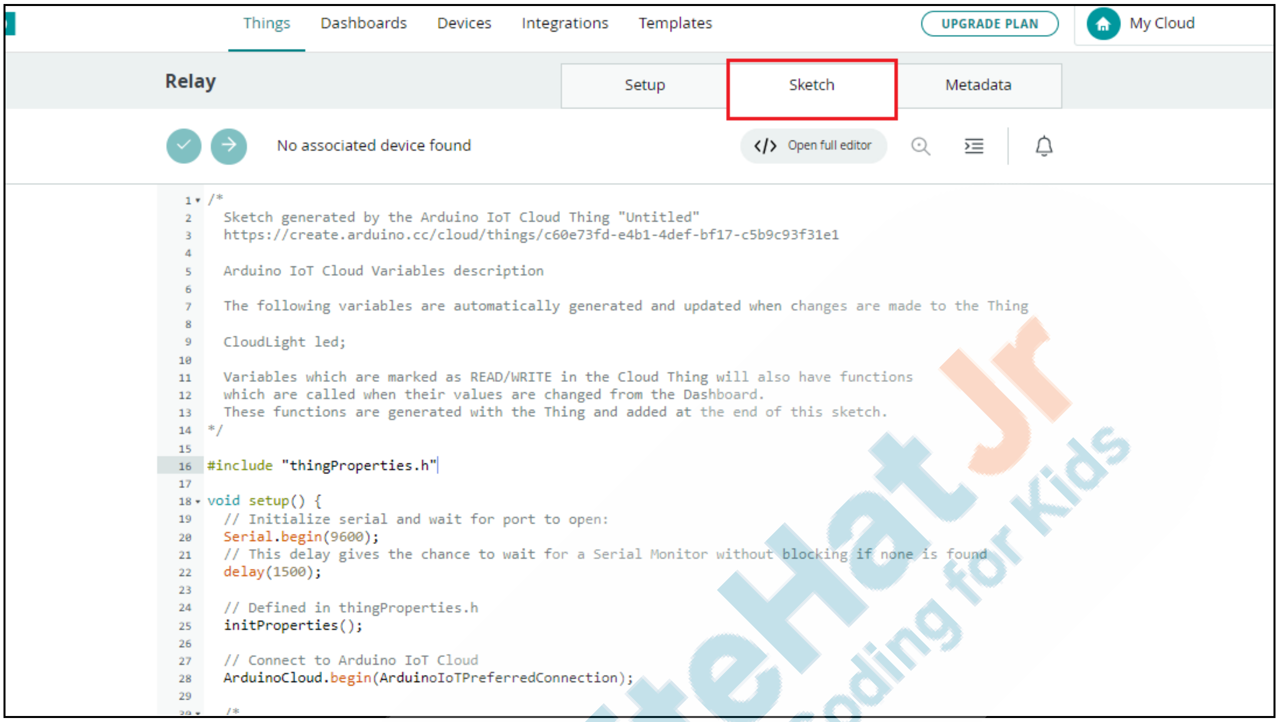
Configure network

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.

Note: Teacher/Student should use their actual WIFi Credentials.

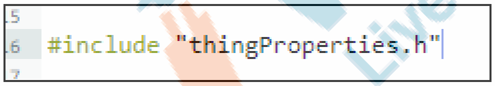
10. Let’s write the program now

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



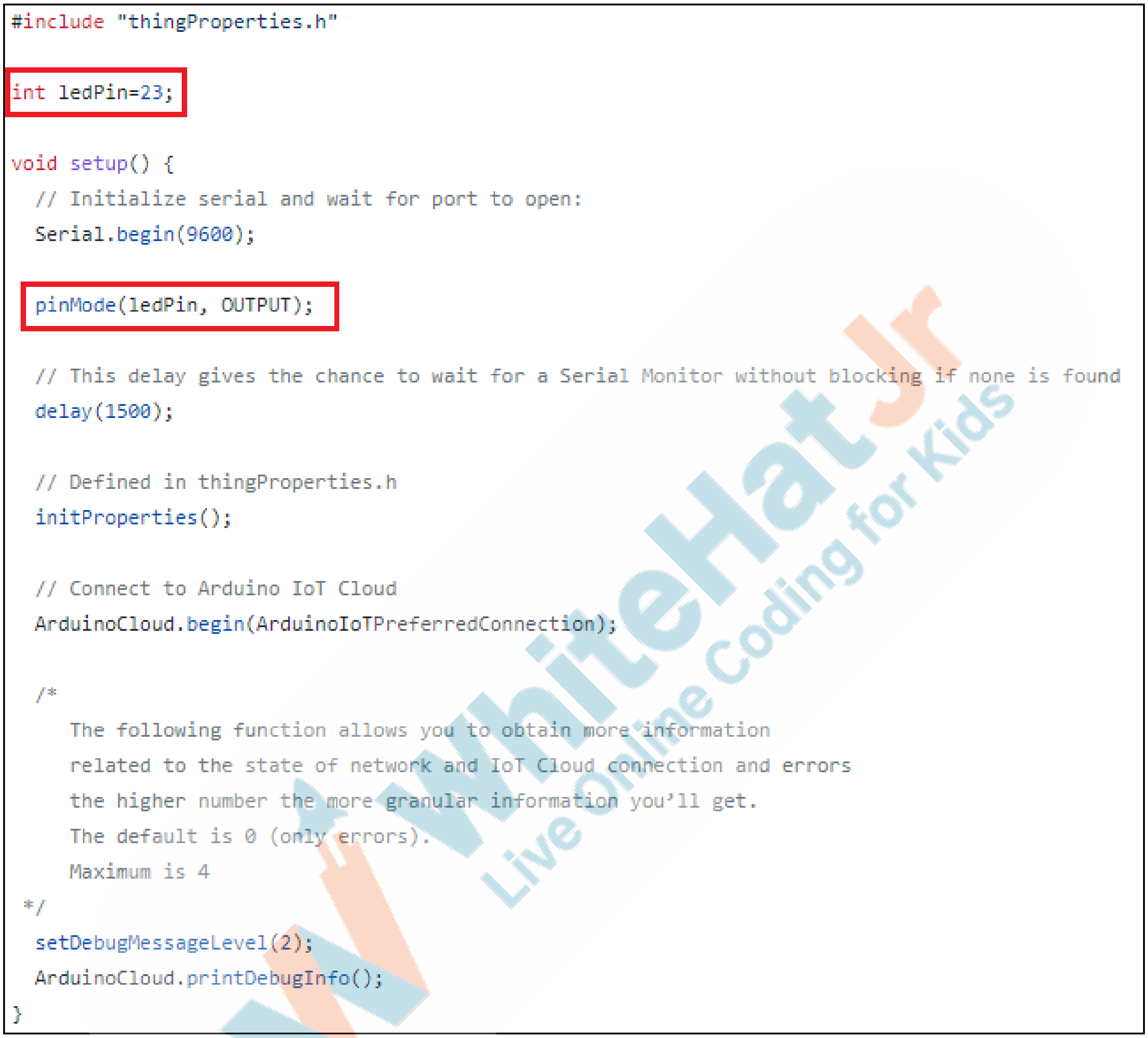
```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled"
  https://create.arduino.cc/cloud/things/c60e73fd-e4b1-4def-bf17-c5b9c93f31e1

  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.



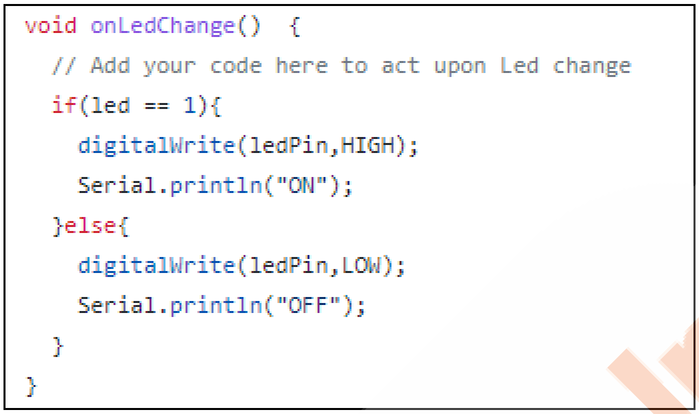
```
#include "thingProperties.h"
```

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 found
  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();
}
```

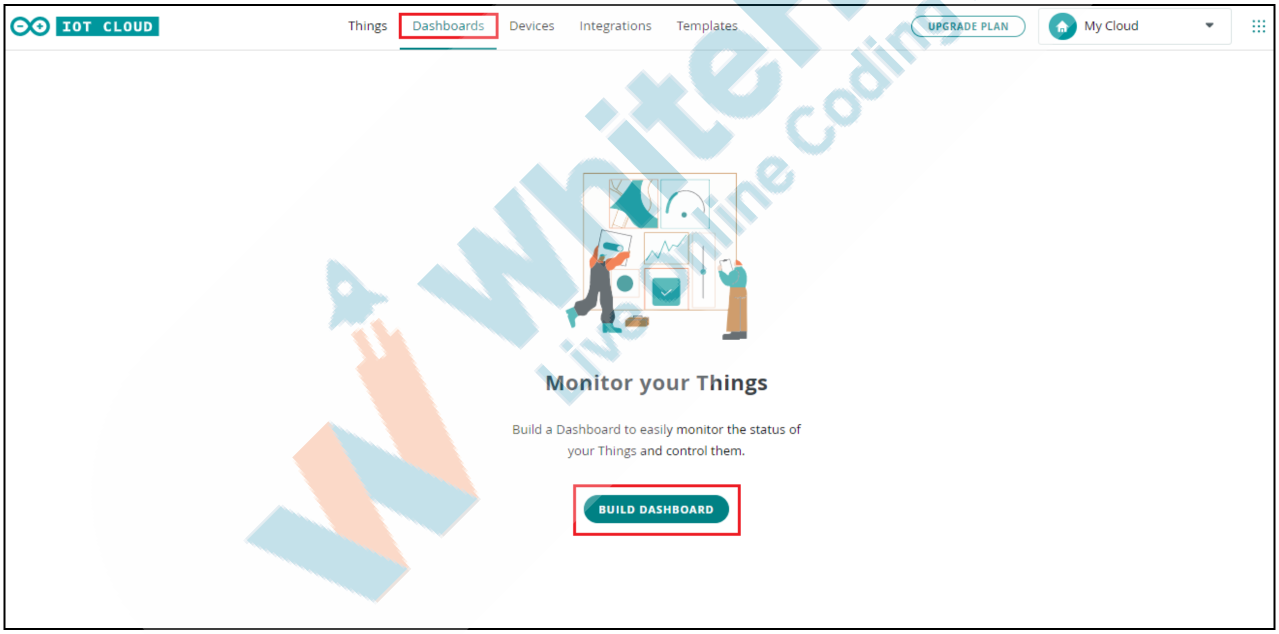
d. Now, find the **onLedChange()** method at the bottom. Write the code to set the pin 23 to HIGH and LOW depending on the **led** variable.



```
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");
  }
}
```

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

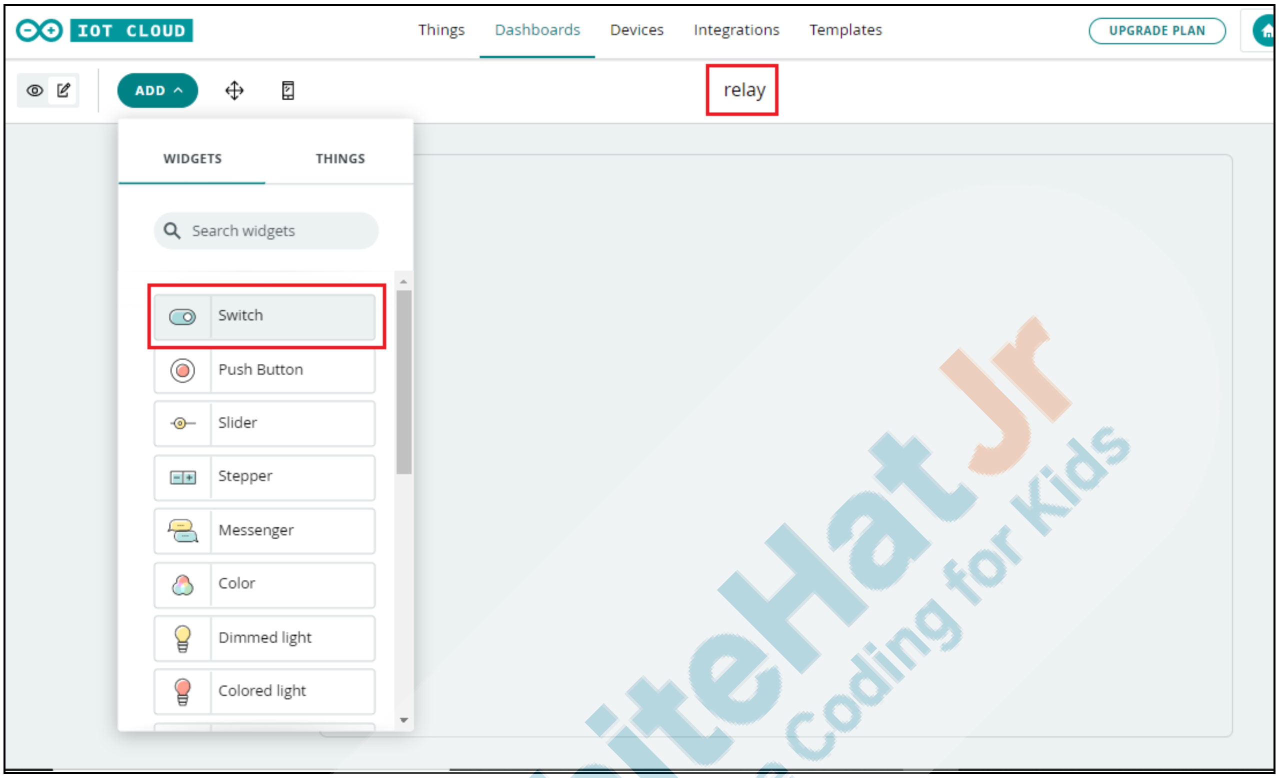
   a. Go to the **Dashboards** tab and click on the **BUILD DASHBOARD** button.



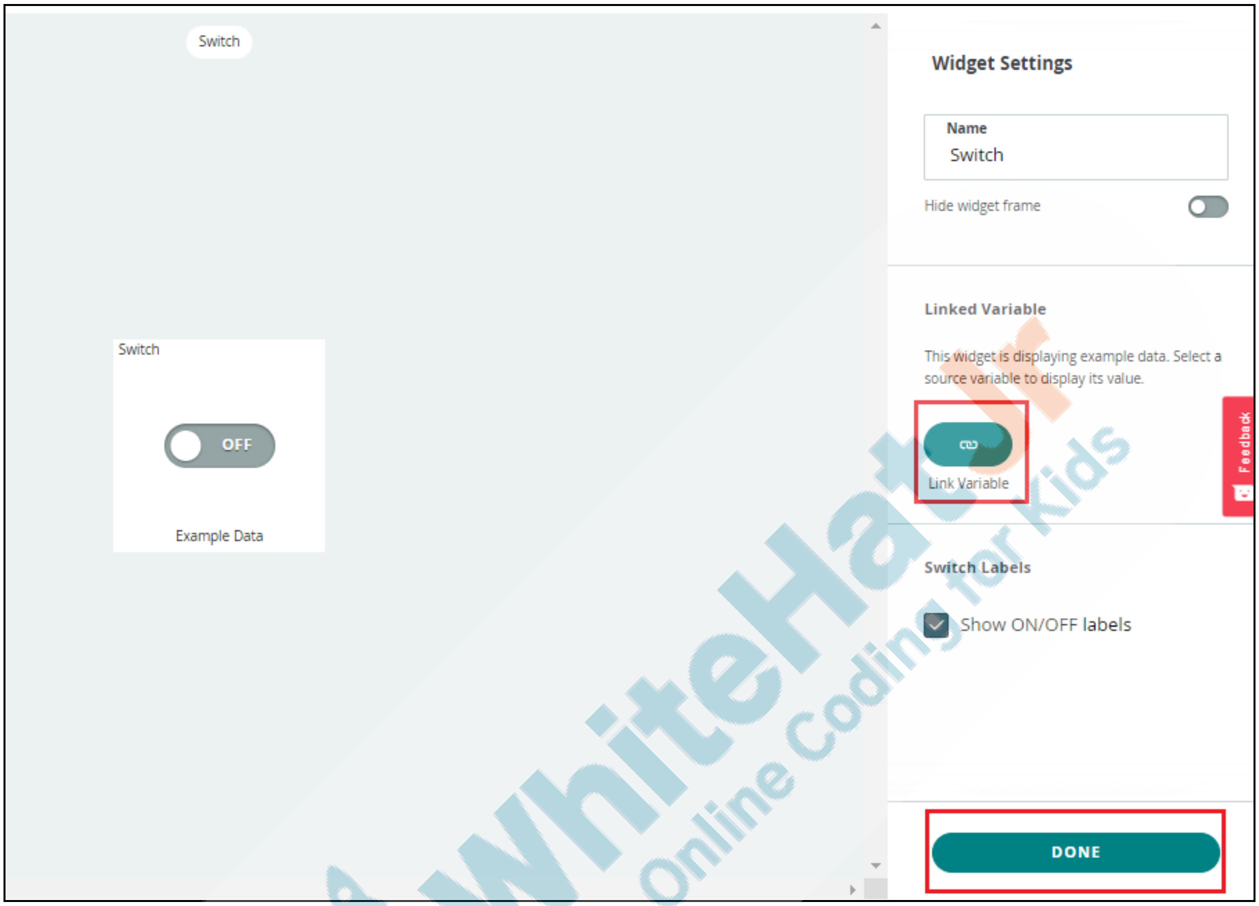
Monitor your Things

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

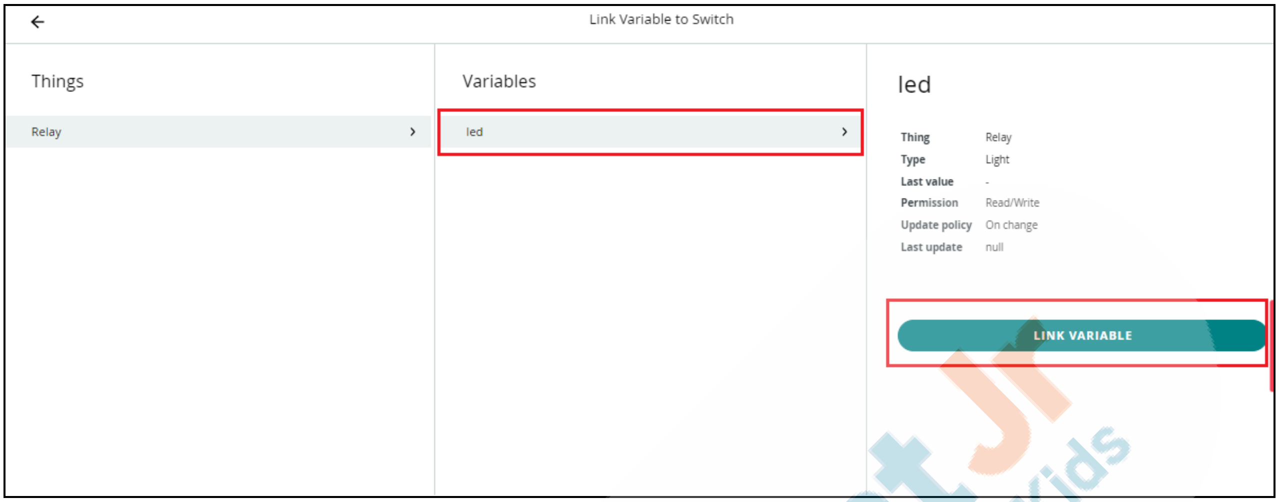
   b. Add a name and add a switch to the dashboard.



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



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



Link Variable to Switch

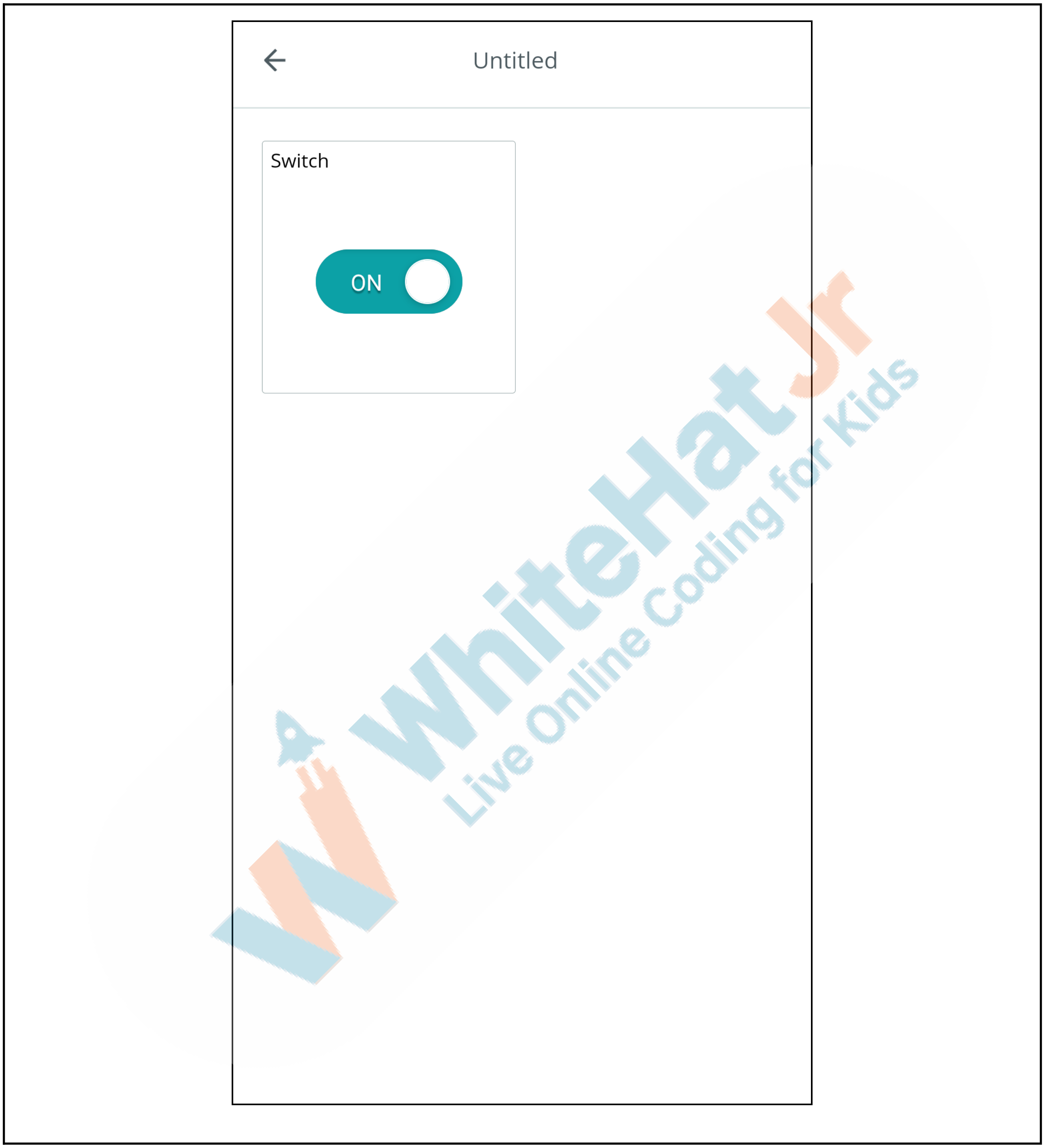
Things	Variables
Relay	led

led

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

After you create this dashboard, you can also download the **Arduino Iot Cloud Remote** app on your phone from the playstore.

Once you have downloaded the app, login with the same credentials. You should be able to control the device from your phone now.



Untitled

Switch

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

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 05 mins

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ

Click on In-Class Quiz

Continue WRAP-UP Session

Slide 19-24

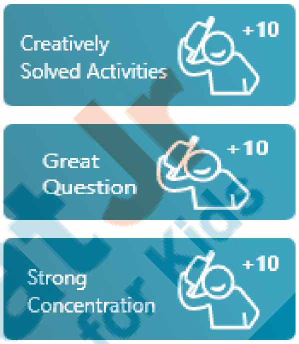
Activity Details

Following are the session deliverables:

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! In the next class, we will start building a Weather Monitoring system.	*Make sure you have given at least 2 hats-off during the class for*  Creatively Solved Activities +10 Great Question +10 Strong Concentration +10
PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections	
Teacher Clicks ✖ End Class	

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Relay Working	https://s3-whjr-curriculum-uploads.whjr.online/dad615e7-49f8-4093-b215-4cb2b3a594a4.gif
Teacher Activity 2	Arduino IoT Cloud	https://cloud.arduino.cc/
Teacher Activity 3	Reference Code	https://github.com/procodingclass/P

		RO-C250-Reference-Code
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/92185c2e-2b46-4cf4-bad9-37d02ef7f3c0.docx
Teacher Reference 2	Project Solution	https://github.com/procodingclass/PRO-C250-Project-Solution
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/e56a3fc1-97b8-48ec-b4d4-d72ec1f37811.pdf
Student Activity 1	Relay Working	https://s3-whjr-curriculum-uploads.whjr.online/dad615e7-49f8-4093-b215-4cb2b3a594a4.gif
Student Activity 2	Arduino IoT Cloud	https://cloud.arduino.cc/