ECOLE NATIONALE
SUPÉRIEURE
D'INFORMATIQUE

الجـــمهوريـــة الجزائريـــة الديمـــقراطيـــة الشـعبيـة
وزارة التعليـــم العـــالي والبـــحث العلمـــي
People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Intelligent and Communicating Systems, ICS

2^{nd} Year Specialty SIQ G02, 2CS SIQ2

LAB report n°08

Title:

Iot System based Platform
Cloud and standalone

Studied by:

First Name: Nour el Imane
Last Name: HEDDADJI
E-mail: jn_heddadji@esi.dz

A. Theory

1. IoT Cloud Platforms

An IoT platform serves as a foundational application or service that seamlessly **connects and orchestrates** elements within an IoT ecosystem. It encompasses device lifecycle management, communication protocols, data analytics, integration capabilities, and application enablement.

This **central hub** facilitates the provisioning, management, and automation of connected devices, simplifying the development of IoT solutions.

Operating as a **multi-layered infrastructure**, the platform connects diverse hardware to the **cloud or local servers** through flexible connectivity, robust security, and extensive data processing. For developers,

The IoT platform, whether cloud-based or standalone, aims to streamline development, reduce risk and cost, and accelerate time to market by handling non-differentiated functionalities.

1.1. How does an IoT platform work?

An IoT platform facilitates seamless connectivity and communication among all components within your IoT ecosystem through a unified application:

1. **Connect**: It links your devices and sensors, managing tasks from registering a single device to bulk connecting thousands. Remote management capabilities enable tasks like updating device firmware and software.
2. **Data Handling**: The platform then handles data transmission, applying analytics to collected data. It provides access to IoT device data at both granular and high-level views.
3. **Integration**: An IoT platform integrates device data with other business applications and record systems, empowering you to generate insights and make impactful decisions.

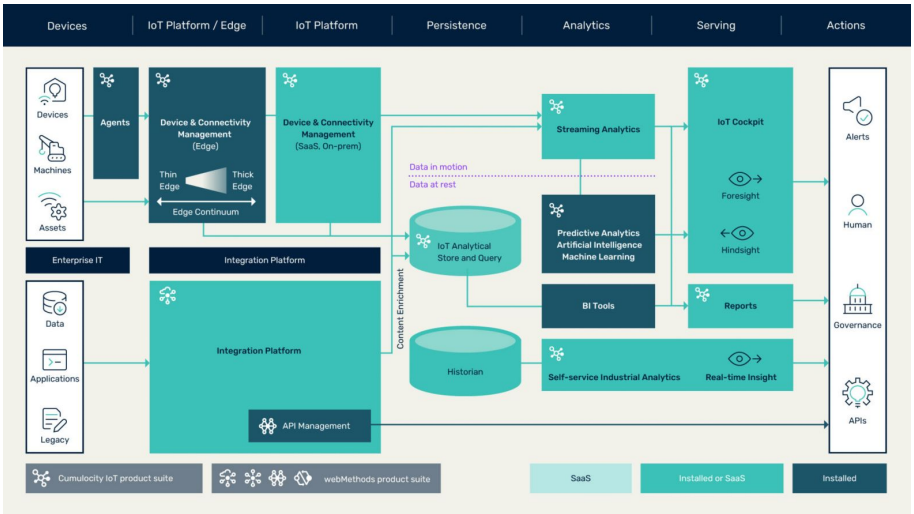


Figure 1: How does an IoT platform work?

2. Categories of IoT Platforms

2.1. IoT Cloud Platforms

IoT cloud platforms are cloud-based services that facilitate the connection, communication, and management of IoT devices. They provide a centralized infrastructure for handling data, security, and device interactions in the context of the Internet of Things.

2.2. IoT Standalone Platforms

Standalone IoT platforms, in contrast, are solutions deployable locally, independent of cloud infrastructure. Whether open-source or proprietary, these platforms provide users with the flexibility to manage their IoT devices autonomously.

3. compare 4 well-known of cloud / standalones platforms

3.1. Arduino IoT Cloud

Criteria	Description
Use Case	IoT device management within Arduino ecosystem
Open Source	No
Price Model	Freemium (Limited free usage, additional costs)
Platform Type	Cloud-based
Installation Flexibility	Limited (Primarily designed for cloud use)
Community Support	Limited (Arduino community)
Device Compatibility	Arduino devices
Customization Options	Limited
Integration Capabilities	Limited (Focused on Arduino ecosystem)
User Interface	User-friendly interface within Arduino IDE
Automation & Rules	Limited
Security Features	Standard security measures

3.2. eWeLink

Criteria	Description
Use Case	**Smart home automation**
Open Source	No
Price Model	Freemium (Limited free usage, additional costs)
Platform Type	Cloud-based
Installation Flexibility	Limited (Primarily designed for cloud use)
Community Support	Community support
Device Compatibility	Various smart home devices
Customization Options	Limited (within eWeLink ecosystem)
Integration Capabilities	Limited (Focused on eWeLink ecosystem)
User Interface	User-friendly app interface
Automation & Rules	Limited (within eWeLink ecosystem)
Security Features	Standard security measures

3.3. Blynk

Criteria	Description
Use Case	**IoT application development**
Open Source	Partial (Community Edition is open source)
Price Model	Freemium (Limited free usage, additional costs)
Platform Type	Cloud-based
Installation Flexibility	Limited (Primarily designed for cloud use)
Community Support	Active community
Device Compatibility	Wide range of IoT devices
Customization Options	Extensive
Integration Capabilities	Broad range of integrations
User Interface	User-friendly app interface
Automation & Rules	Comprehensive
Security Features	Secure connection options

3.4. openHAB

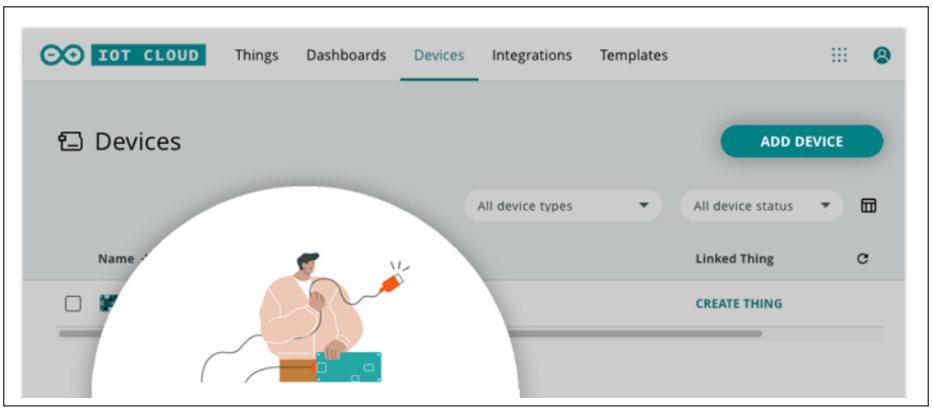
Criteria	Description
Use Case	**Home automation and IoT integration**
Open Source	**Yes**
Price Model	**Open source (Free)**
Platform Type	**Standalone**
Installation Flexibility	High (Can be installed locally)
Community Support	Strong community support
Device Compatibility	Diverse IoT devices
Customization Options	Highly customizable
Integration Capabilities	Extensive integrations
User Interface	Web-based interface
Automation & Rules	Extensive automation options
Security Features	Emphasis on security

B. Activity

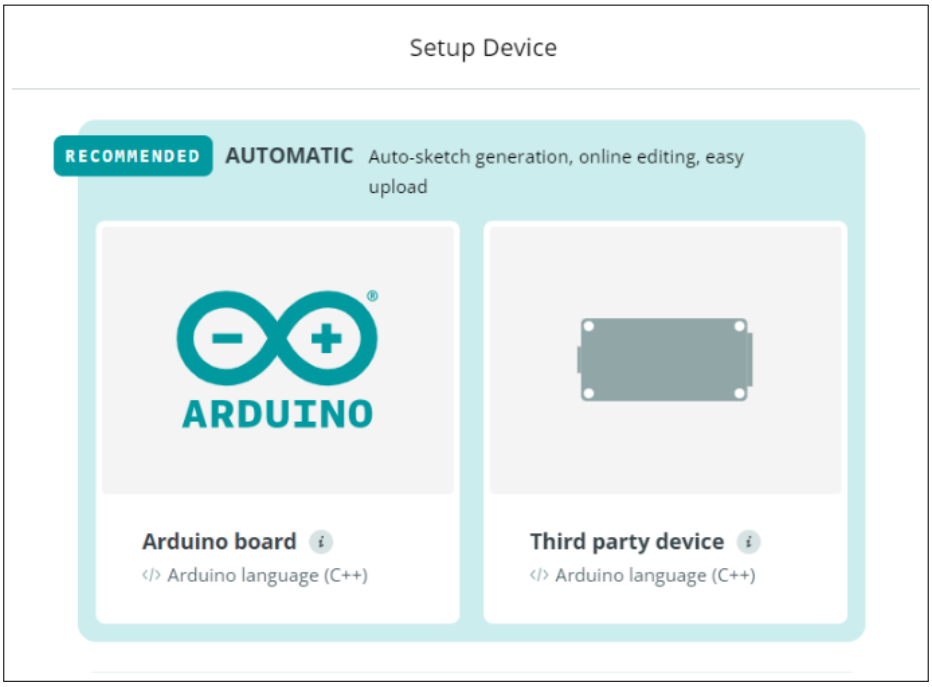
1. Arduino IoT Cloud (Web version)

1.1. Set-up instructions

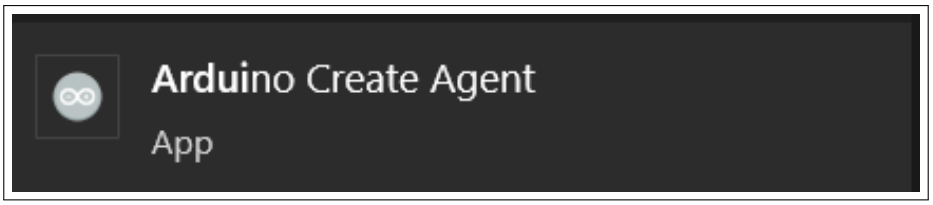
1. Go to the Arduino IoT Cloud website and create an account.
2. Open the Devices tab and click the *Add Device* button in the top-right the page

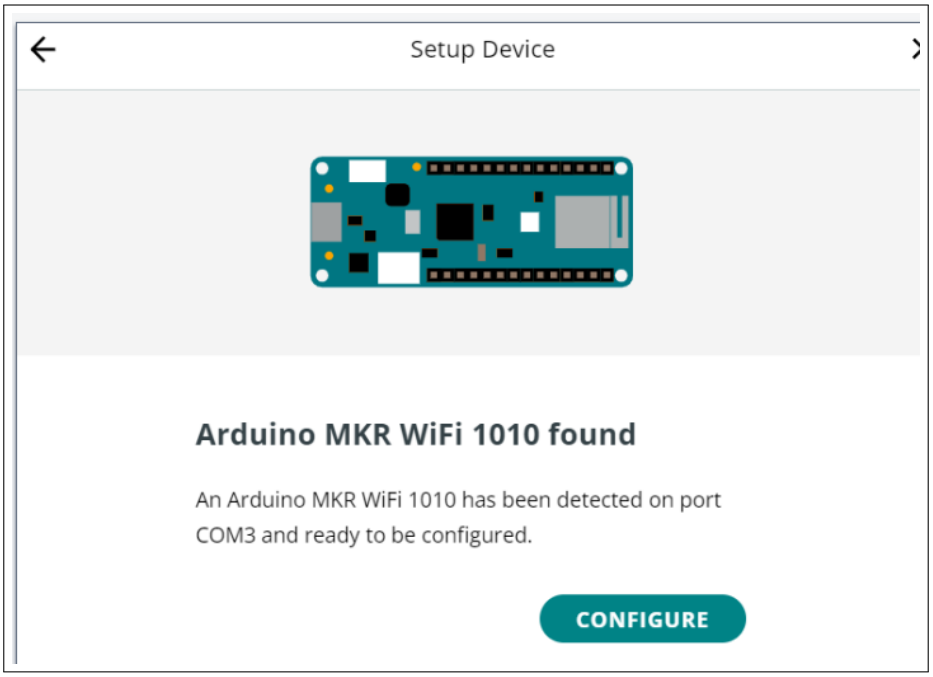


3. Select your device type

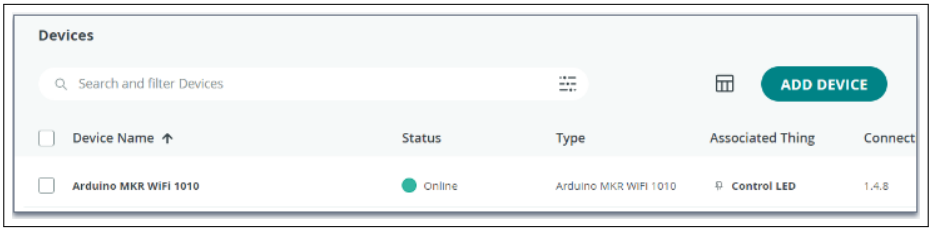


4. Install and run the *Arduino create agent* software so that the Arduino IoT cloud platform could detect the device

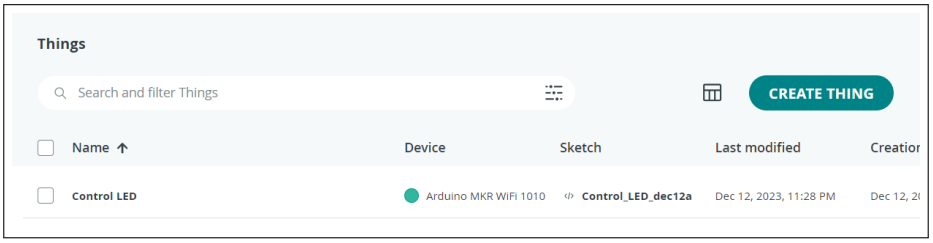


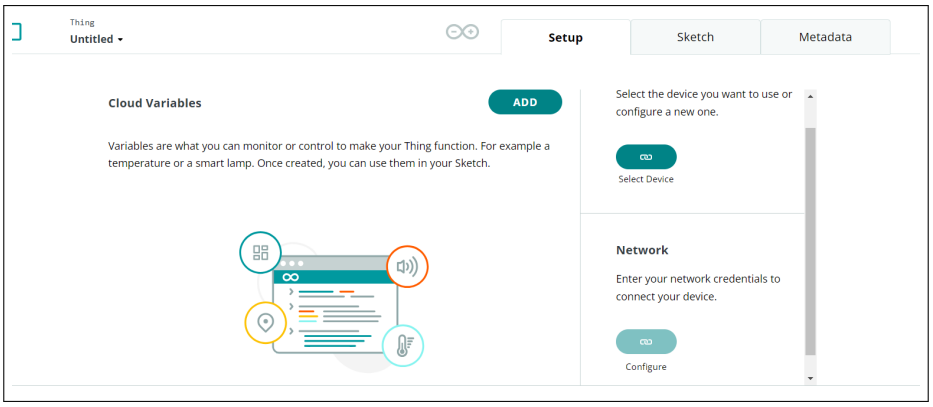


5. Proceed with the instructions to add your device.

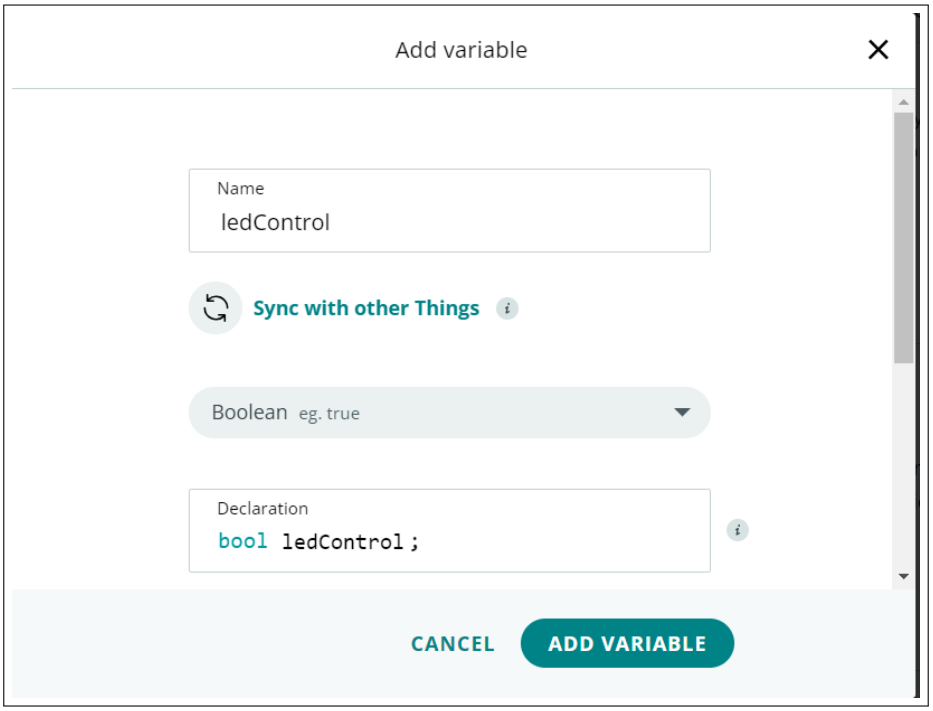


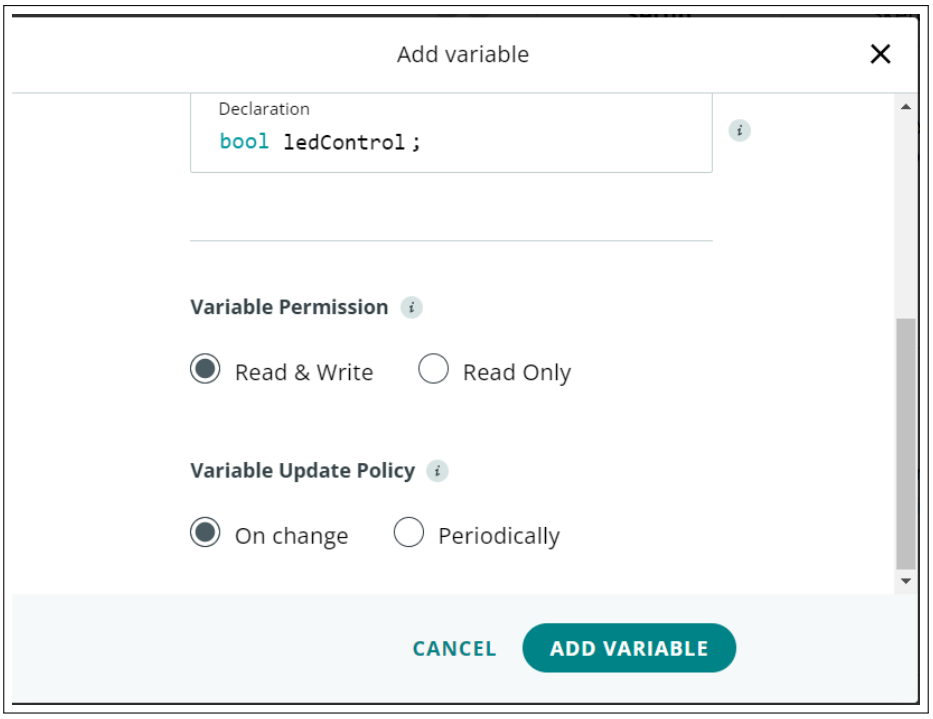
6. Go back to the left sidebar menu in your workspace and choose the ”Things” option and Click on the ”Create Thing” button



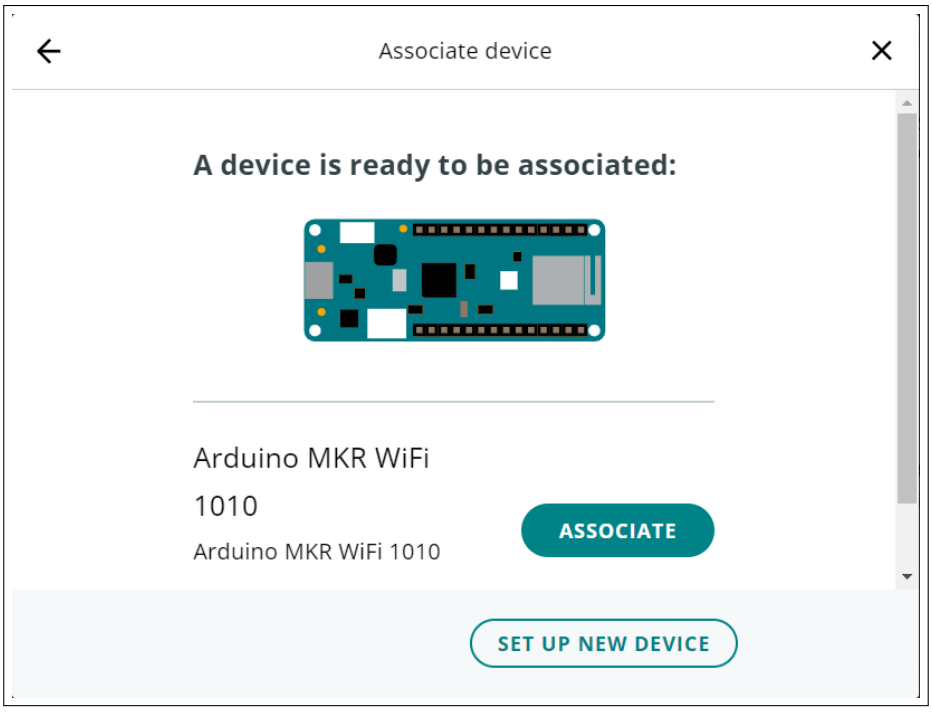


7. Add a variable.

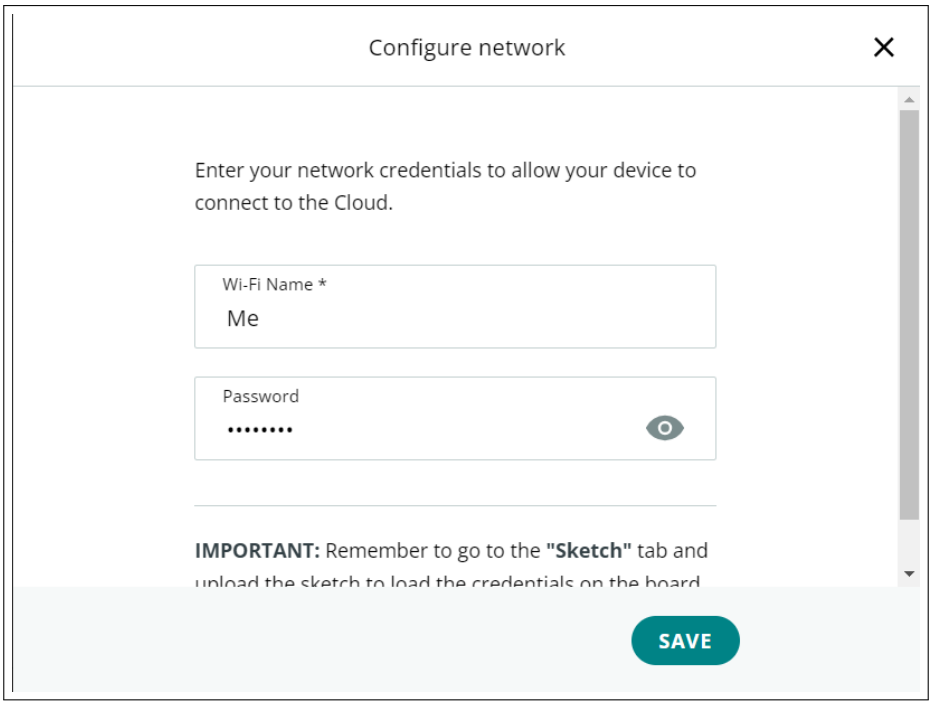




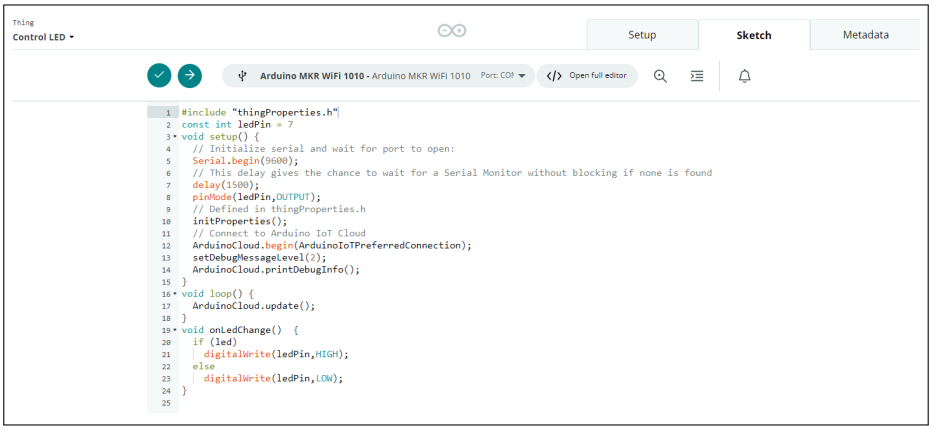
8. Select the device you added previously to associate it with the Thing.



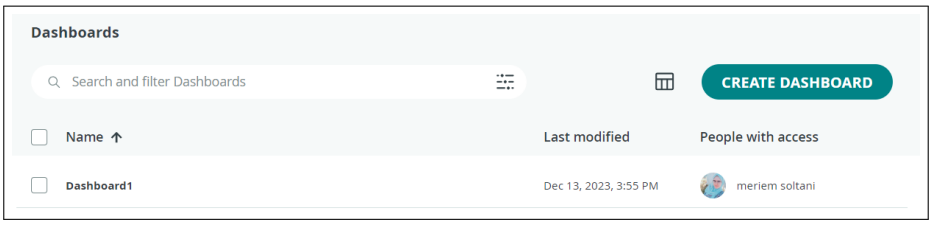
9. Enter the network credentials.



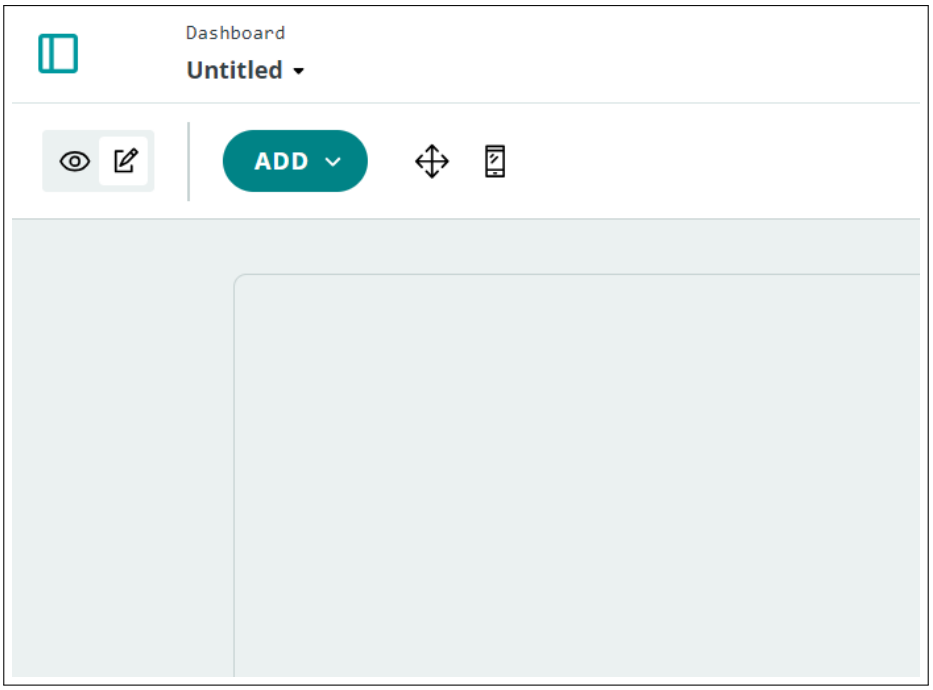
10. In the ”Sketch” section, write down the code to handle the switch and LED.



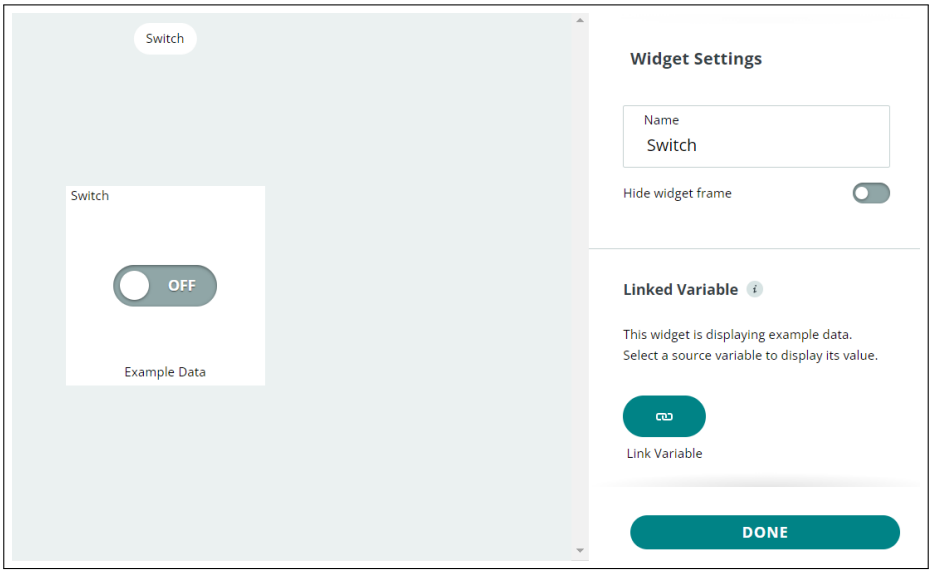
11. Go back to the left sidebar menu and choose the ”Dashboard” option then click on ”Create a Dashboard.”

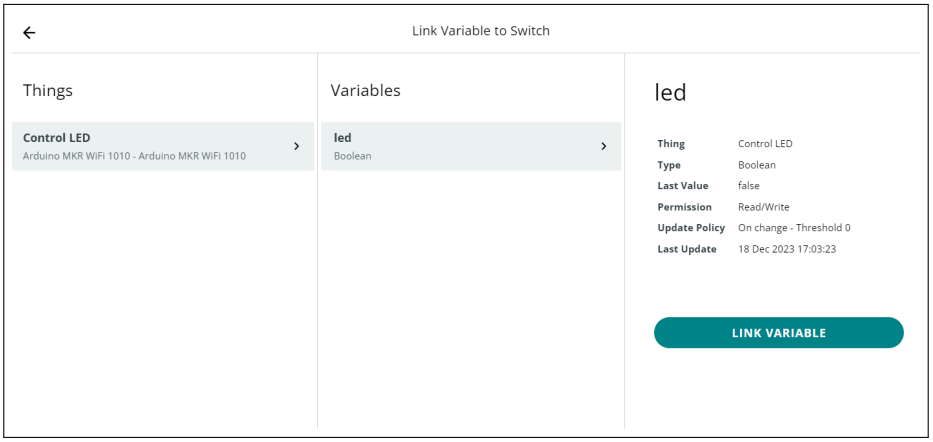


12. Add a switch element to the dashboard.

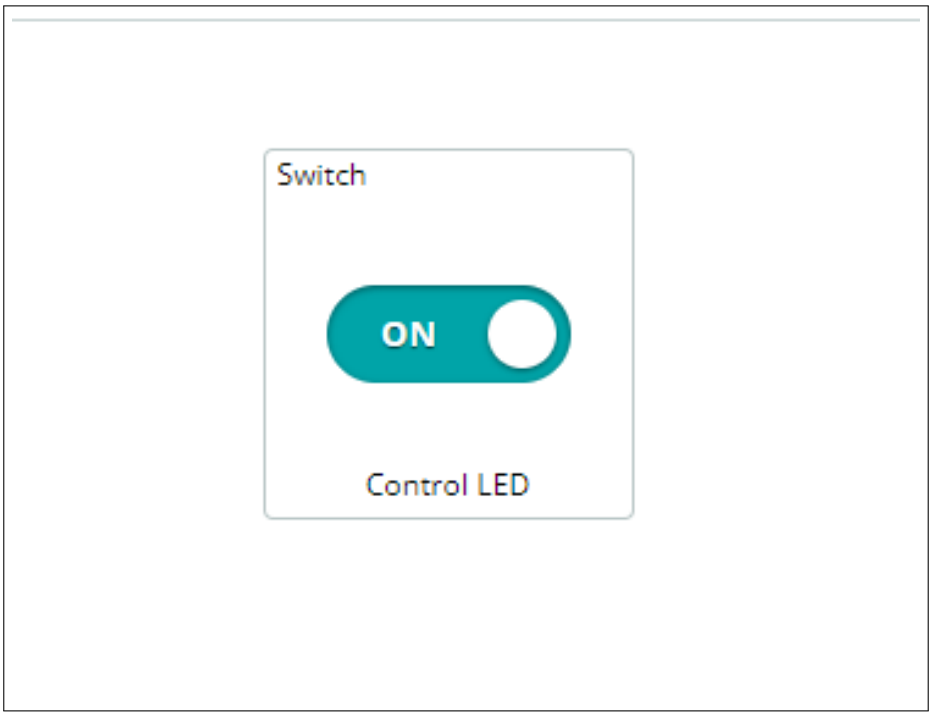


13. Link the switch to the variable you created in the Thing.





14. Turn the switch on and off to test the circuit. The LED should respond accordingly.



Hardware Setup

Software

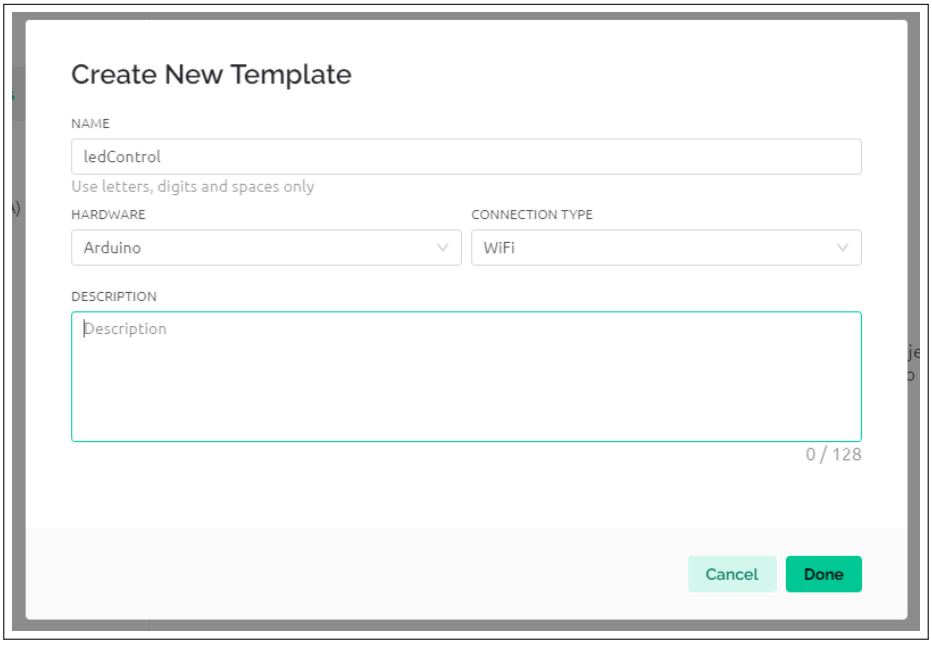
```
#include "thingProperties.h"

const int ledPin = 7;

void setup() {
  Serial.begin(9600);
  delay(1500);
  pinMode(ledPin, OUTPUT);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
}

void onLedChange() {
  if (led)
    digitalWrite(ledPin, HIGH);
  else
    digitalWrite(ledPin, LOW);
}
```

Listing 1: the code to handle the switch and LED

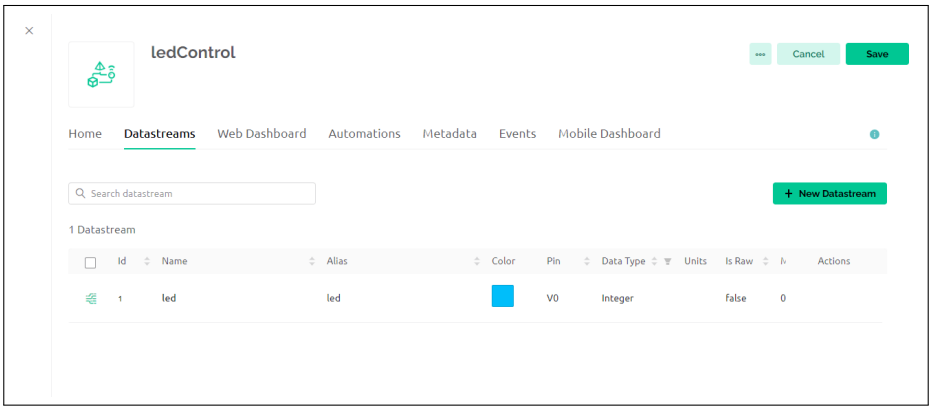
2. Blynk

2.1. Set-up instructions

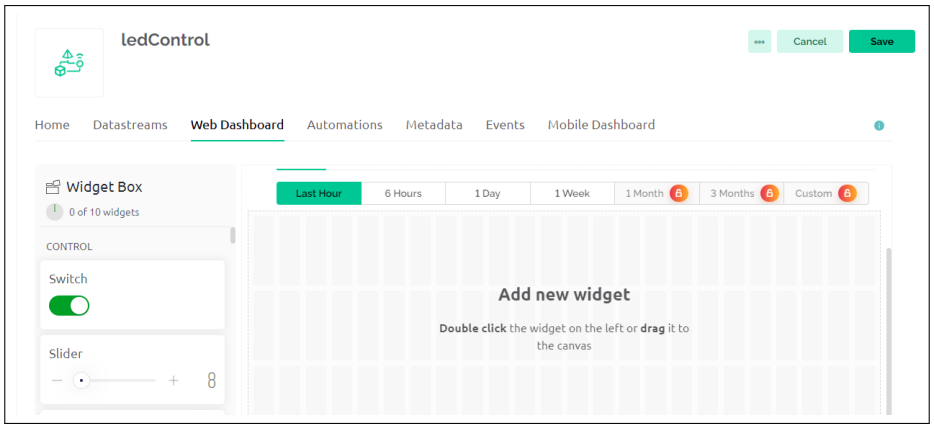
1. In Blynk, Create a new template with connection type 'WIFI'

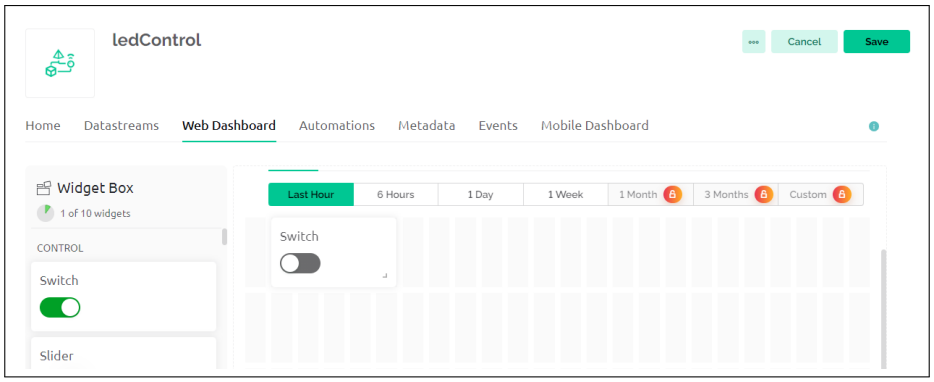


2. Add a new datastream

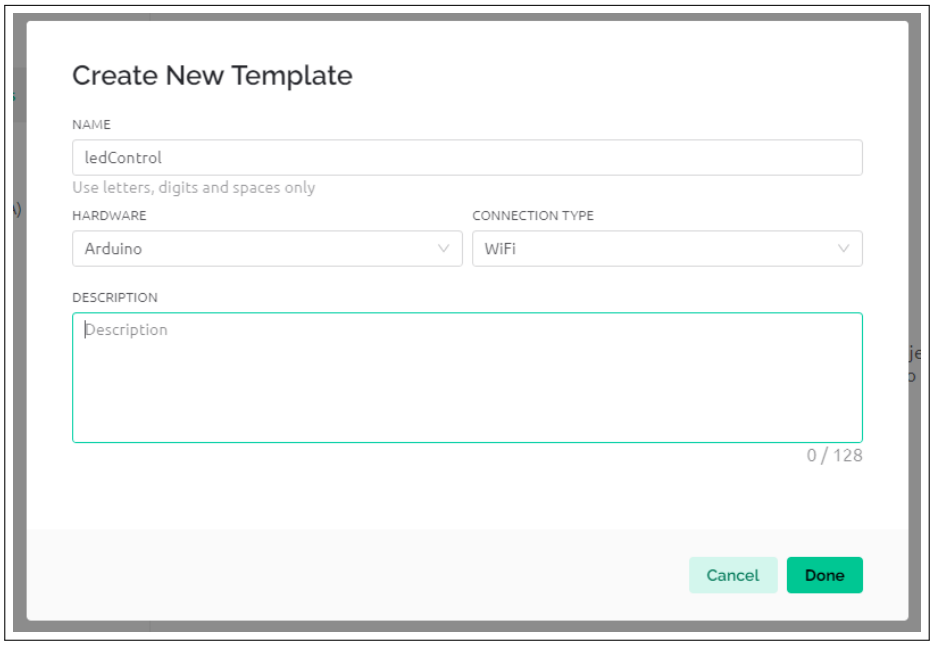


3. Add a new widget (Switch)

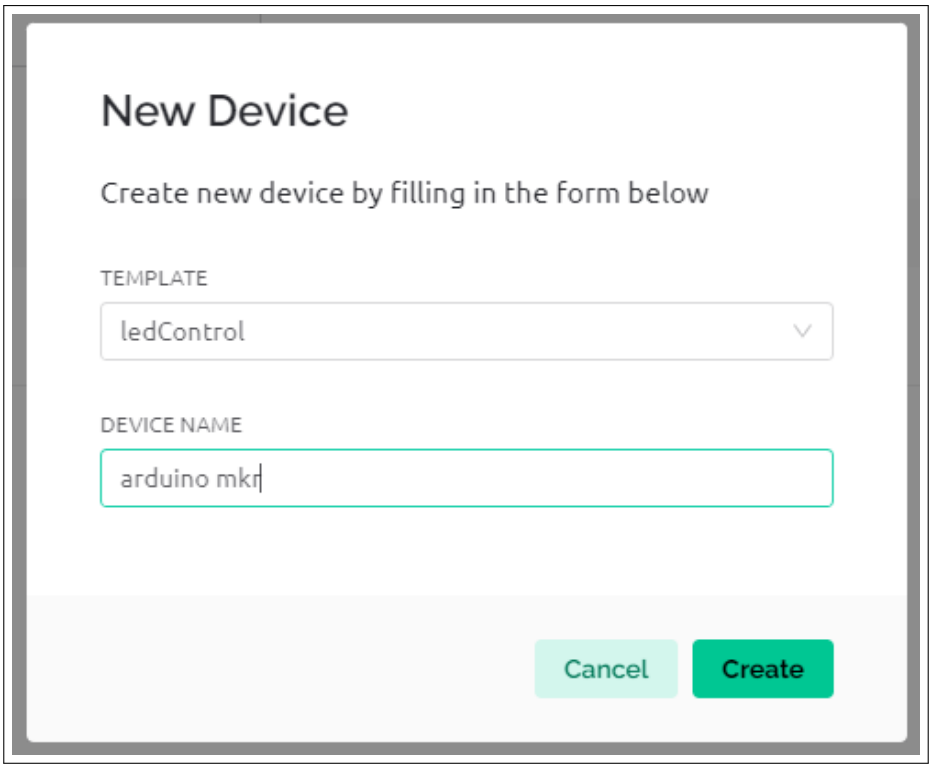


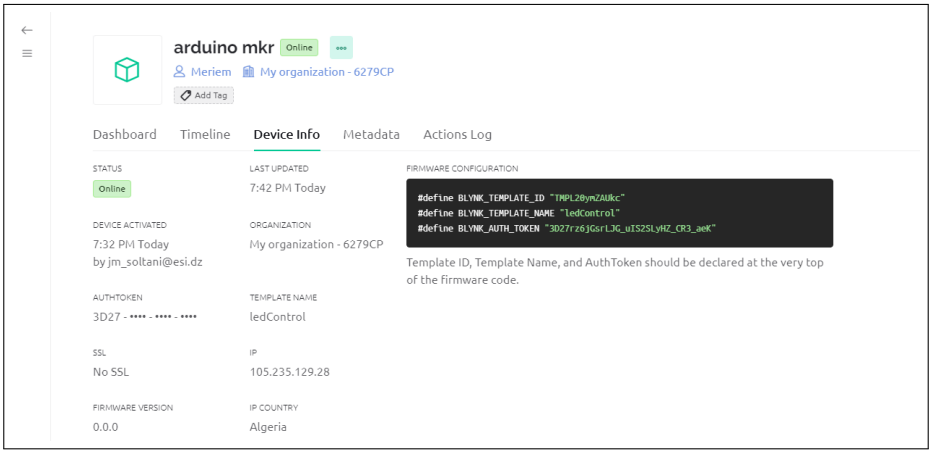


4. Install the libary 'Blynk' in your arduino Ide



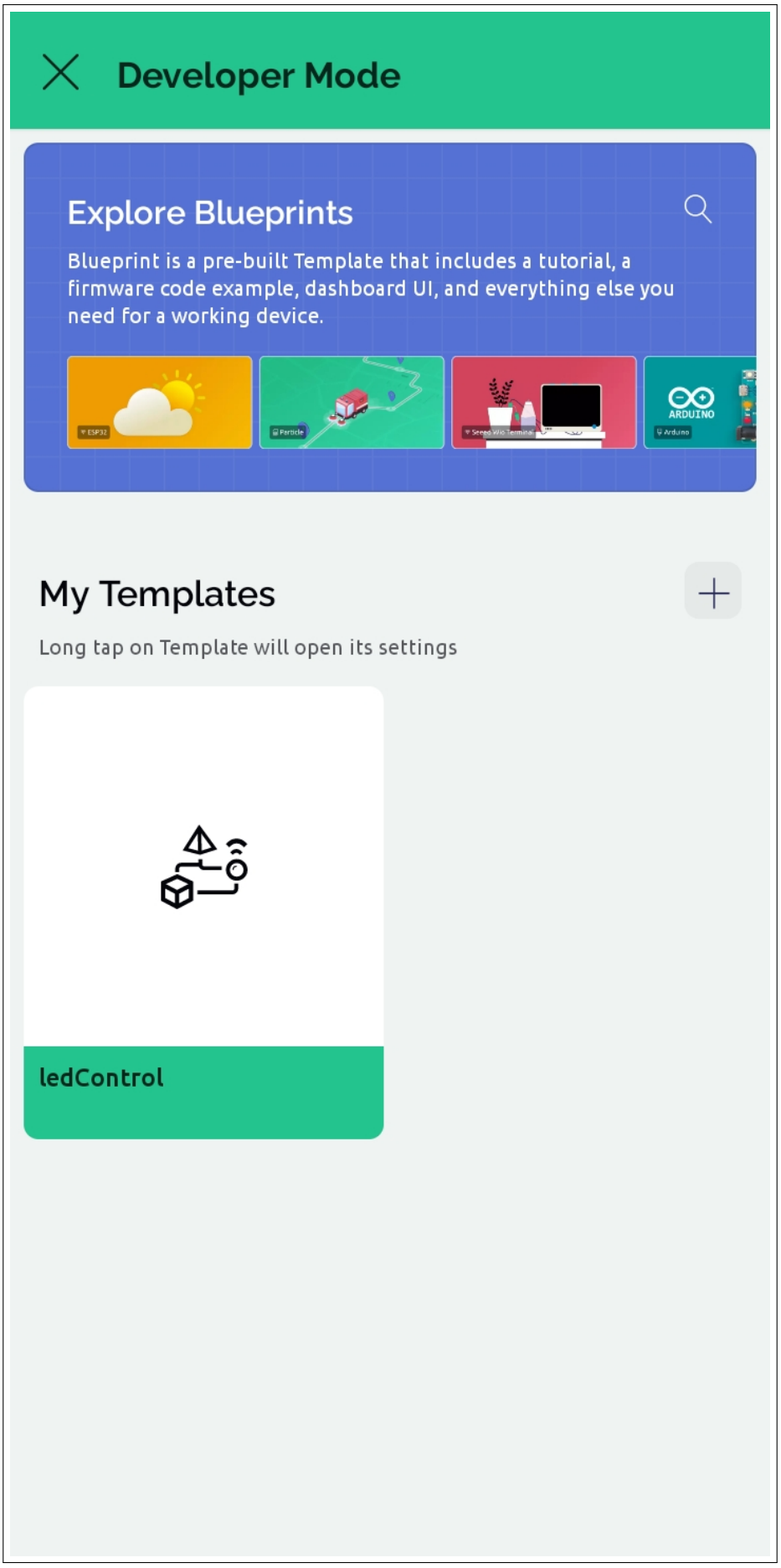
5. Go to the Devices tab and add your device



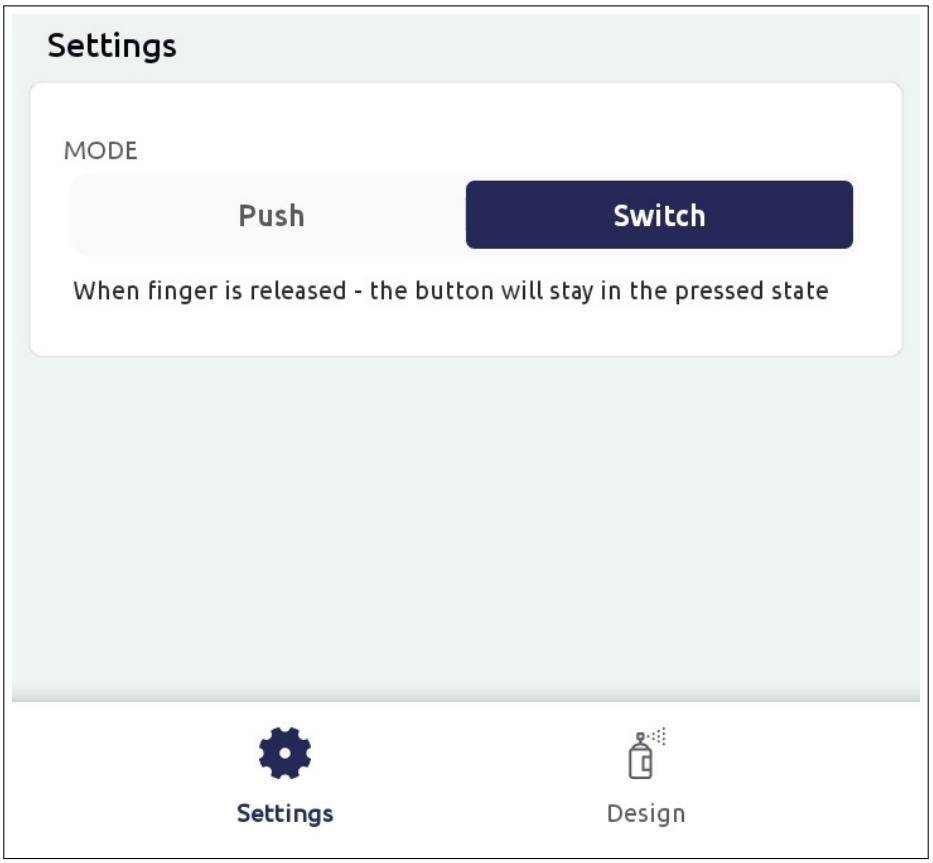


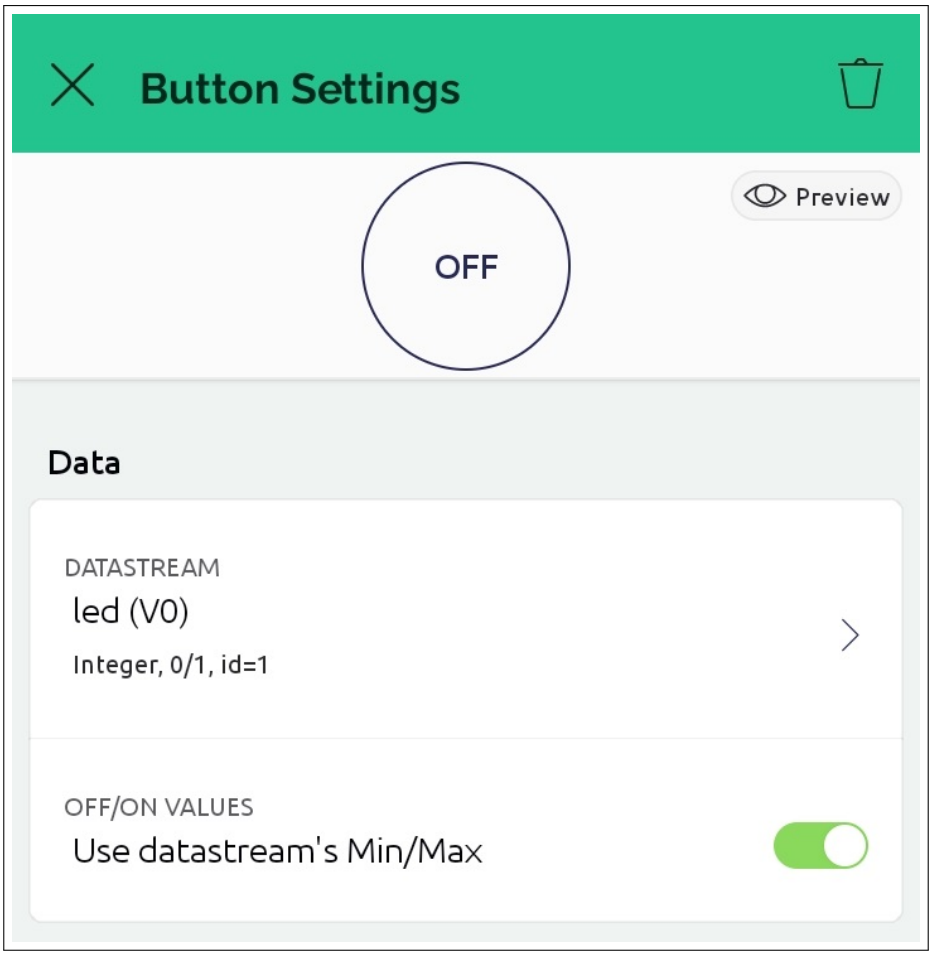
6.

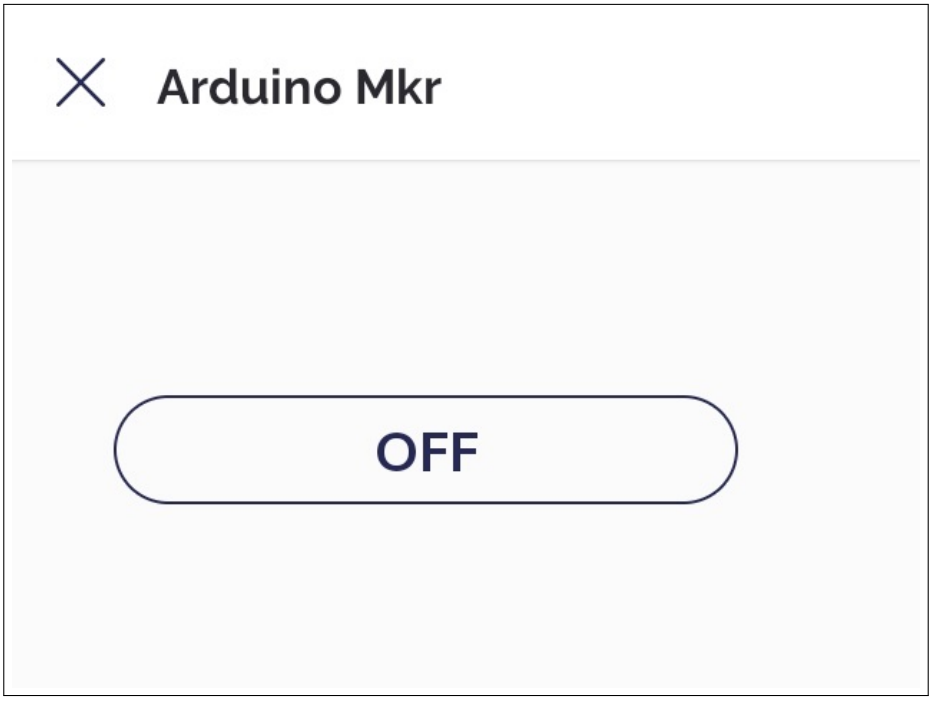
7. in your mobile app, go to my templates and look for the led control.


Developer Mode
Explore Blueprints
Blueprint is a pre-built Template that includes a tutorial, a firmware code example, dashboard UI, and everything else you need for a working device.
My Templates
Long tap on Template will open its settings
ledControl

8. Choose your widget and control your led .



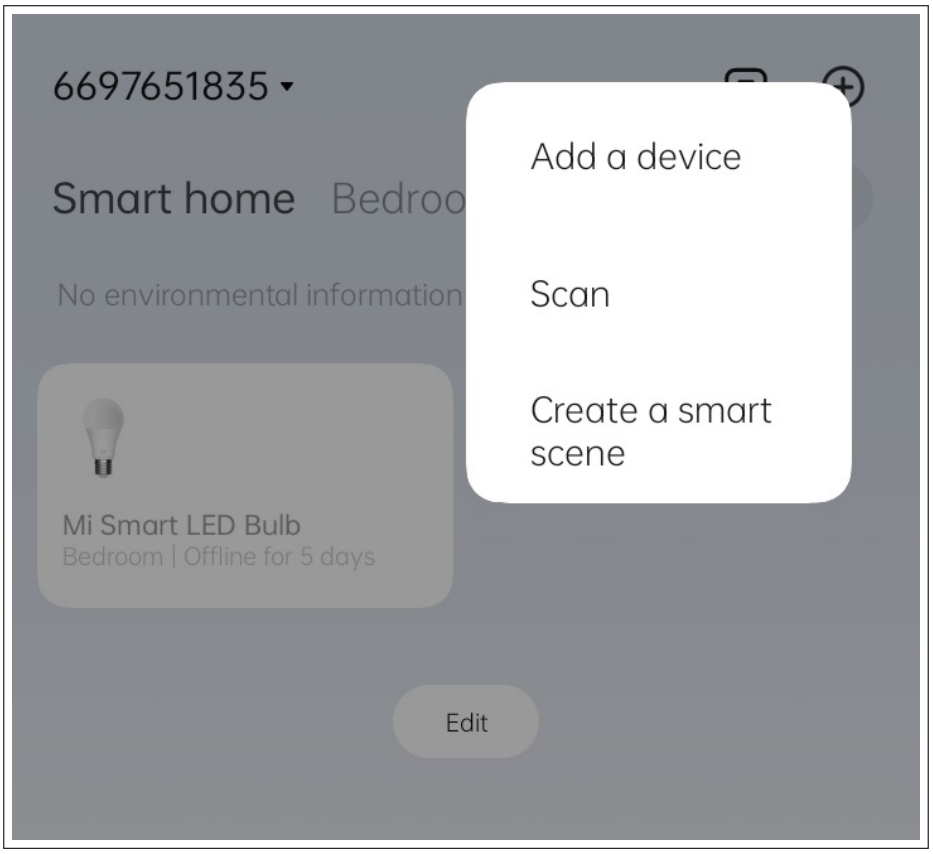

Button Settings
Preview
OFF
Data
DATASTREAM
led (V0)
Integer, 0/1, id=1
OFF/ON VALUES
Use datastream's Min/Max


Arduino Mkr
OFF

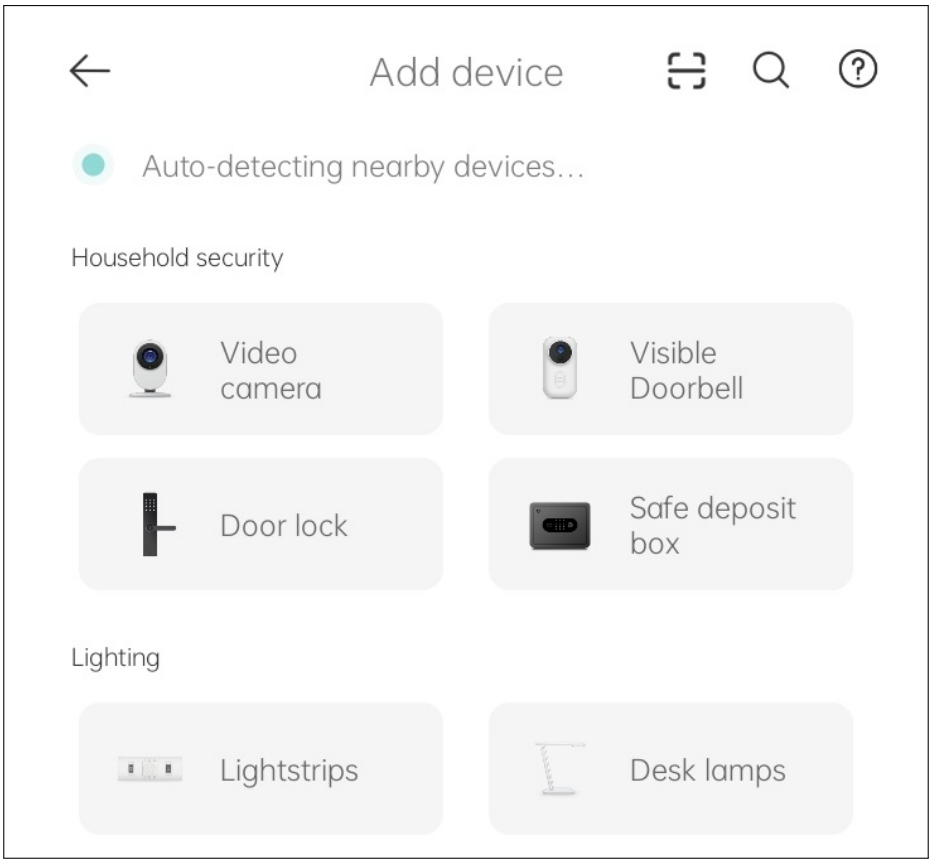
3. Xiaomi Smart Bulb

3.1. Set-up instructions

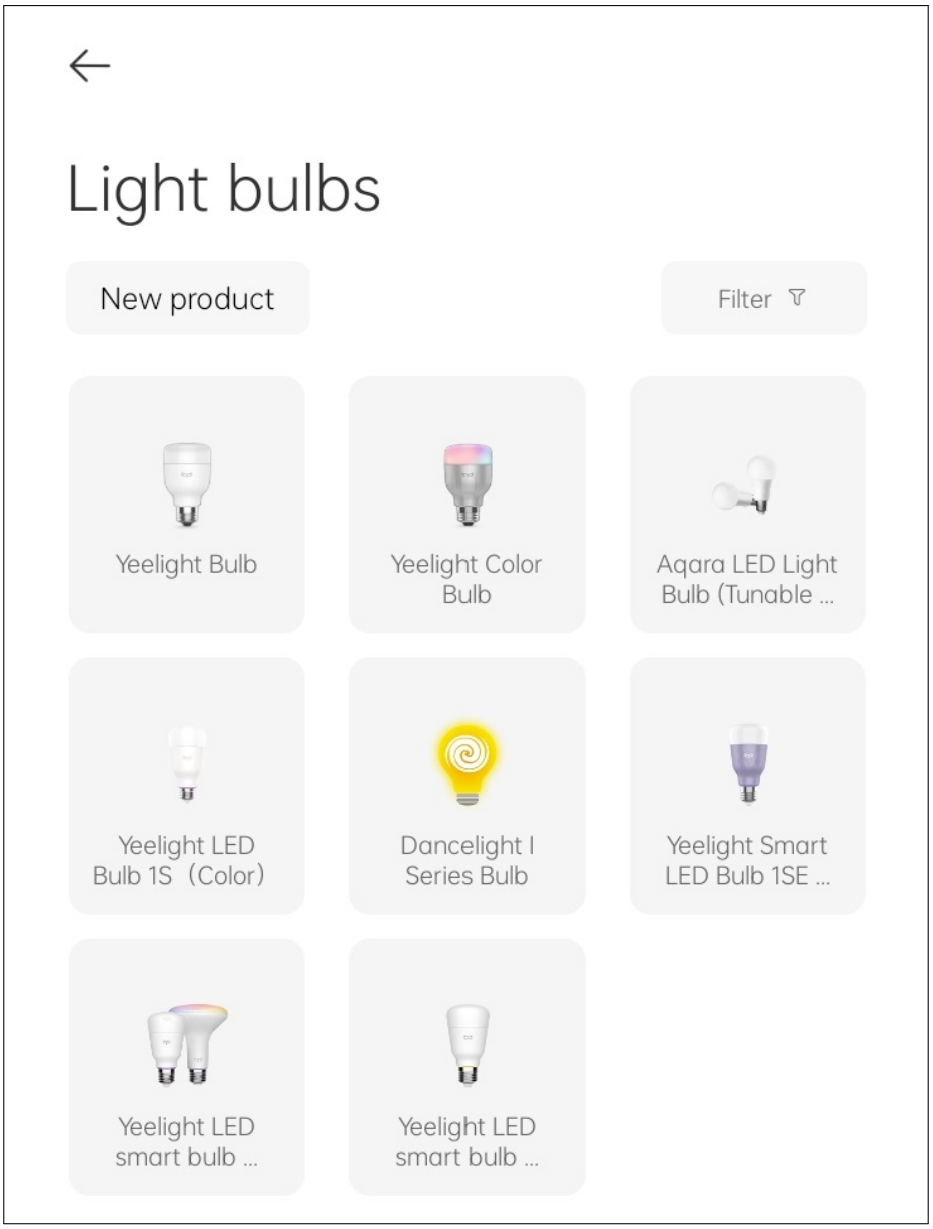
1. Install Mi Home App
2. Create or Sign In to Mi Account
3. Click on the Add device in your Mi Home



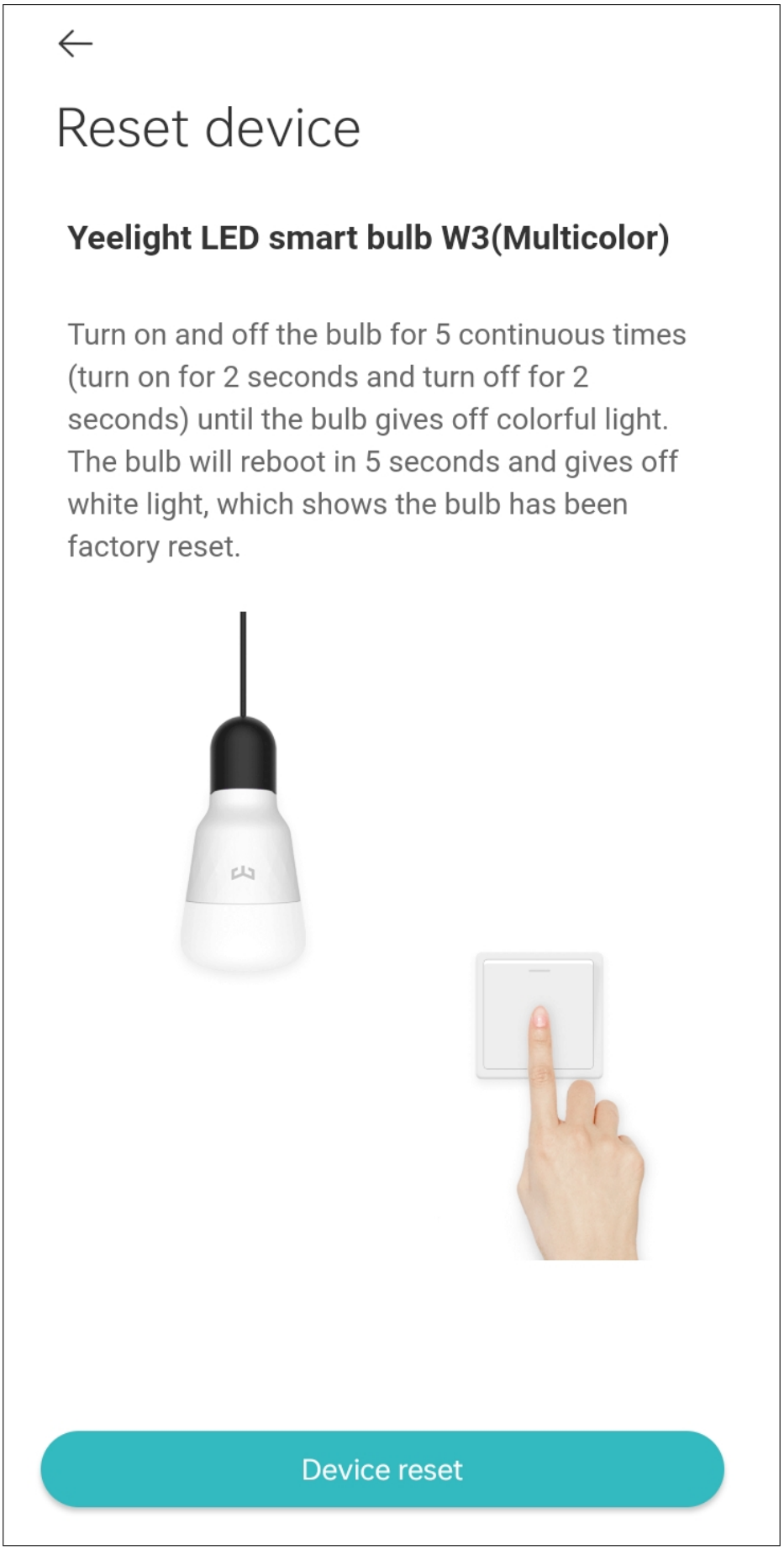
4. Select the type of your device



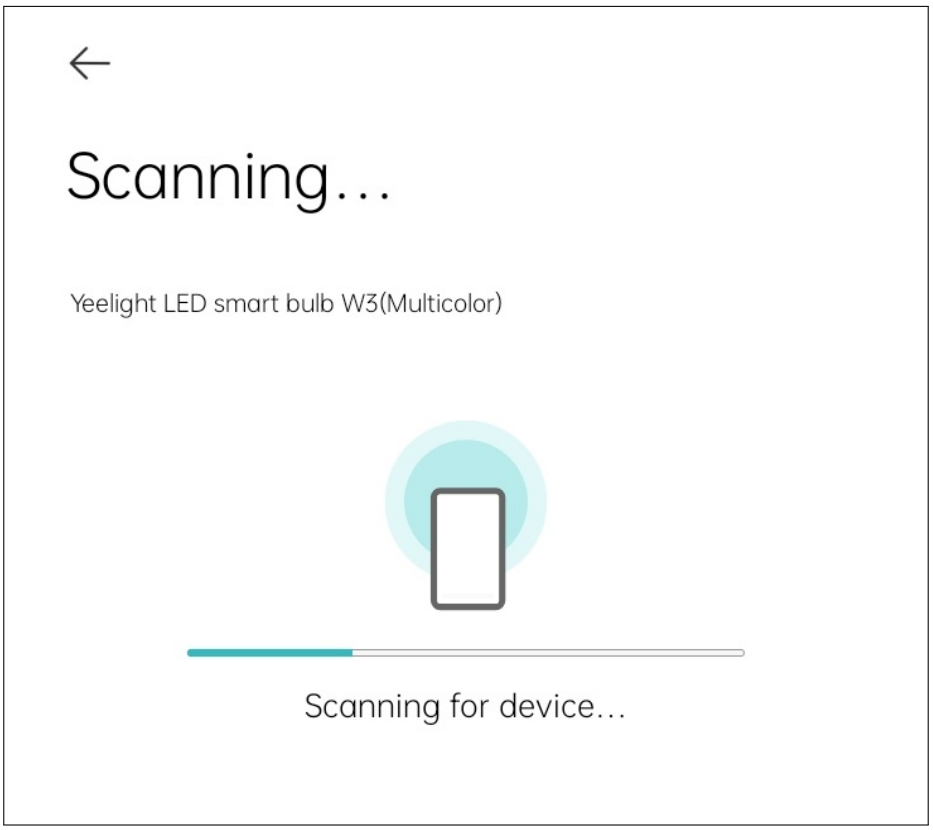
5. Now select the category of your smart bulb



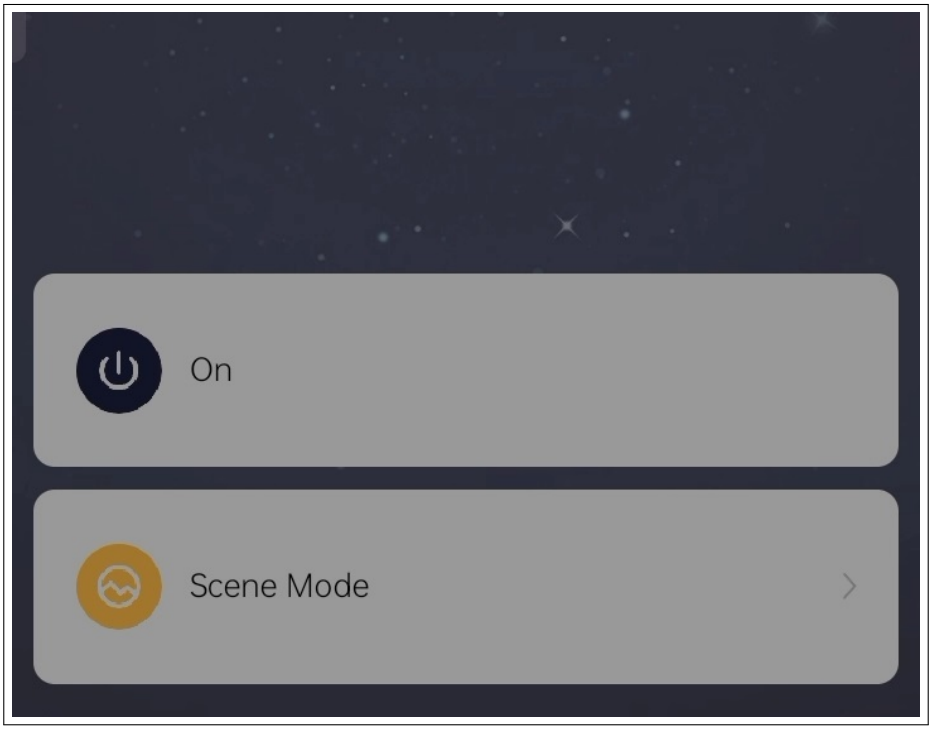
6. Follow the instructions to setup your Xiaomi smart bulb



7. Use Xiaomi scanner to scan your bulb. If it's not scanned, make sure you correctly set up your bulb



8. Control your bulb



3.2. Demo

To watch the video, click here.

C. Conclusion

This lab served as a crucial hands-on exploration into the intricate process of configuring and deploying IoT devices across diverse platforms. Our journey delved into the integration and deployment intricacies of Internet of Things (IoT) devices, with a primary focus on three distinctive platforms: Arduino IoT Cloud, Blynk, and the Xiaomi Smart Bulb.

The rich tapestry of technologies unfolded as we navigated through the unique features and advantages offered by each platform. The Arduino IoT Cloud facilitated seamless device management, emphasizing a cloud-centric approach to IoT solutions. Blynk, with its versatile template creation and widget-based interactions, showcased the potential for dynamic and user-friendly IoT interfaces. In parallel, the Xiaomi Smart Bulb demonstrated the integration of IoT into everyday life, with a user-friendly mobile app enabling effortless control of smart home devices.

The true essence of this lab lies in the realization that IoT is not a **one-size-fits-all** concept; rather, it is a dynamic landscape with platforms tailored to diverse applications. The interplay of these technologies underscored the versatility of IoT in addressing real-world challenges and opportunities. Whether in the cloud, through mobile applications, or embedded in smart home devices, IoT continues to reshape our connected world.

As we reflect on the myriad possibilities explored in this lab, it becomes evident that our hands-on experience in device configuration and interaction has laid a solid foundation for future IoT endeavors. The skills acquired here are not just technical know-how but keys to unlocking the vast potential of IoT innovation.