

Intelligent and Communicating Systems, ICS
2nd 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_heddadjji@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.

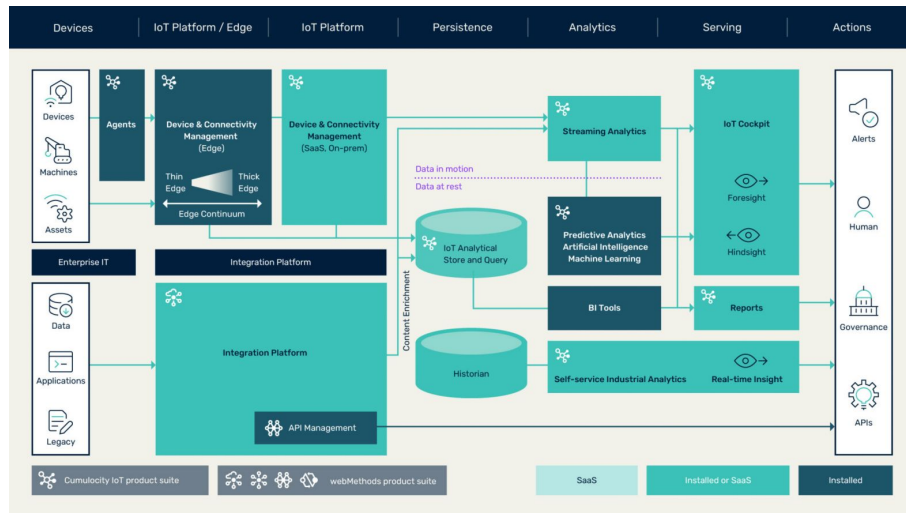


Figure 1: How does an IoT platform work?

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

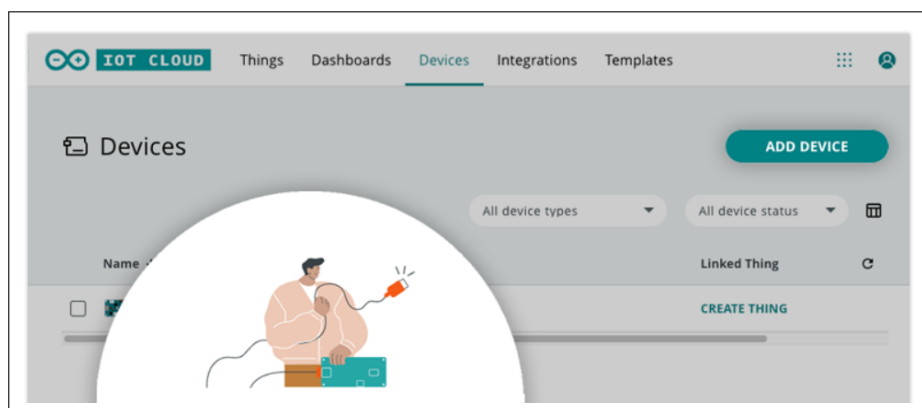
Criteria	Description
Use Case	Home automation and IoT integration
Open Source	Yes
Price Model	Open source (Free)
Platform Type	Standalone
Installation Flexibility	High (Can be installed locally)
Community Support	Strong community support
Device Compatibility	Diverse IoT devices
Customization Options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

B. Activity

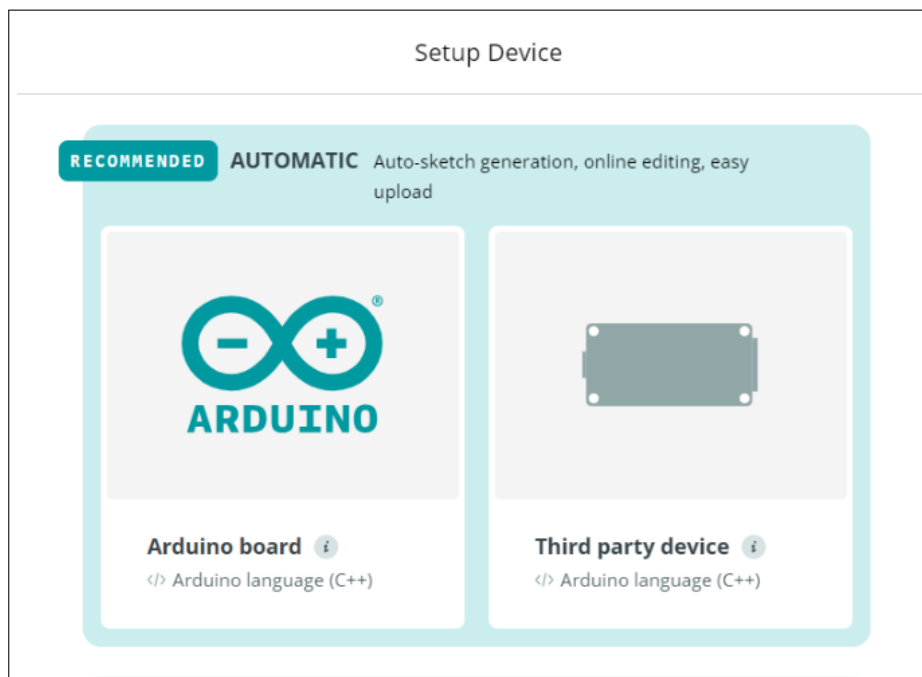
1. Arduino IoT Cloud (Web version)

1.1. Set-up instructions

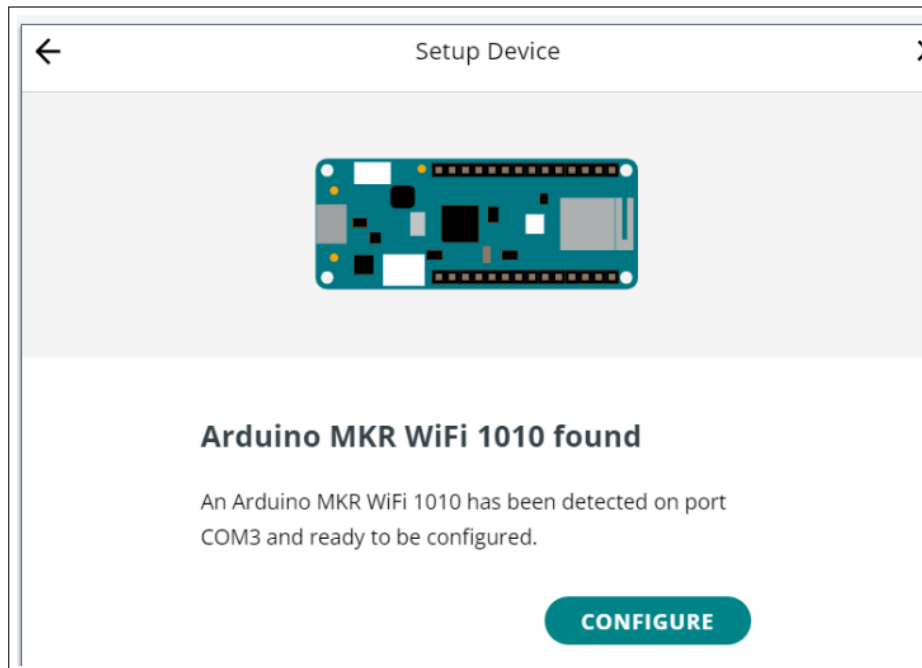
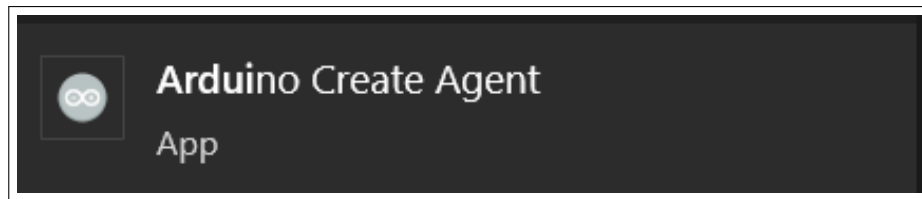
1. Go to the Arduino IoT Cloud website and create an account.
2. Open the Devices tab and click the *Add Device* button in the top-right the page



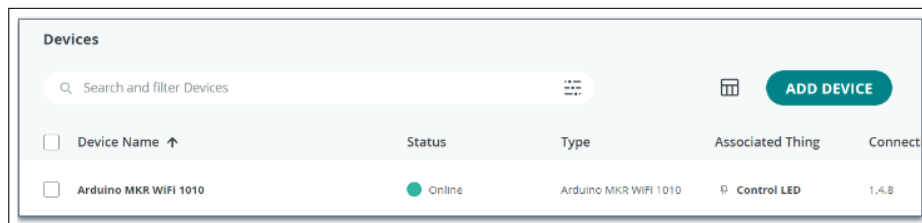
3. Select your device type



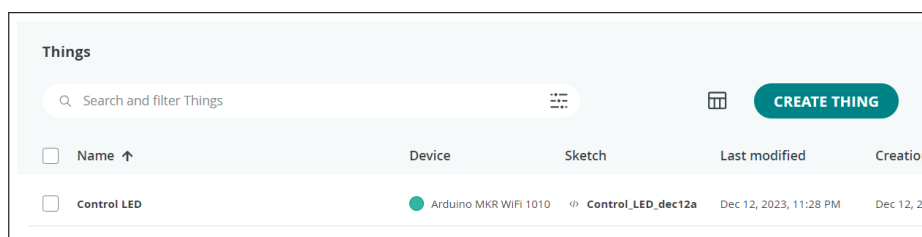
4. Install and run the *Arduino create agent* software so that the Arduino IoT cloud platform could detect the device

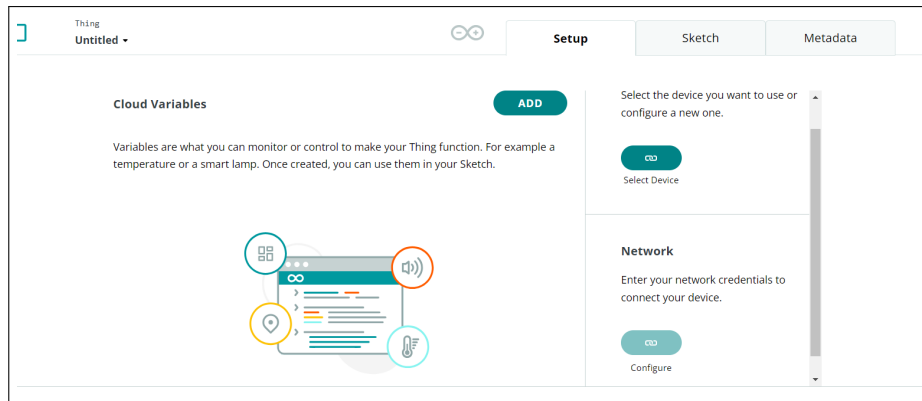


5. Proceed with the instructions to add your device.

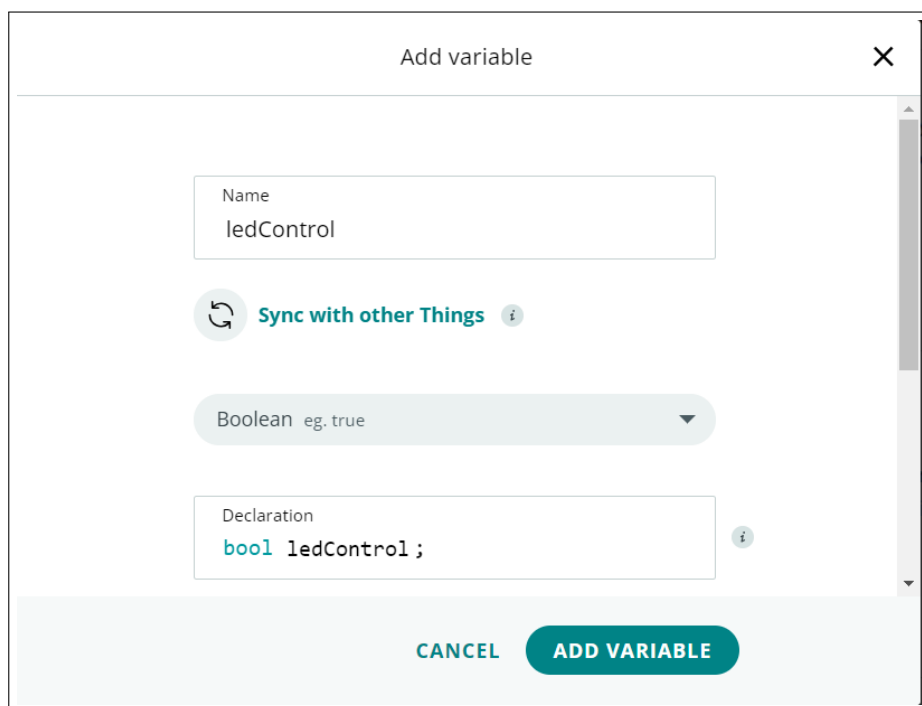


6. Go back to the left sidebar menu in your workspace and choose the "Things" option and Click on the "Create Thing" button





7. Add a variable.



Add variable

Declaration

`bool ledControl;`

Variable Permission

☒ Read & Write

☐ Read Only

Variable Update Policy

☒ On change

☐ Periodically

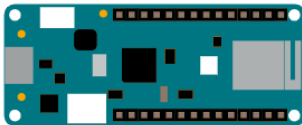
CANCEL

ADD VARIABLE

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

Associate device

A device is ready to be associated:



Arduino MKR WiFi
1010
Arduino MKR WiFi 1010

ASSOCIATE

SET UP NEW DEVICE

9. Enter the network credentials.

Configure network

Enter your network credentials to allow your device to connect to the Cloud.

Wi-Fi Name *

Me

Password

.....

IMPORTANT: Remember to go to the "Sketch" tab and upload the sketch to load the credentials on the board

SAVE

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

Thing

Control LED

Setup

Sketch

Metadata

Arduino MKR WIFI 1010 - Arduino MKR WIFI 1010

Port: COM

Open full editor

```

1 #include "thingProperties.h"
2 const int ledPin = 7
3 void setup() {
4   // Initialize serial and wait for port to open:
5   Serial.begin(9600);
6   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
7   delay(1500);
8   pinMode(ledPin, OUTPUT);
9   // Defined in thingProperties.h
10  initProperties();
11  // Connect to Arduino IoT Cloud
12  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
13  setDebugMessageLevel(2);
14  ArduinoCloud.printDebugInfo();
15 }
16 void loop() {
17   ArduinoCloud.update();
18 }
19 void onLedChange() {
20   if (led)
21     digitalWrite(ledPin, HIGH);
22   else
23     digitalWrite(ledPin, LOW);
24 }
25

```

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

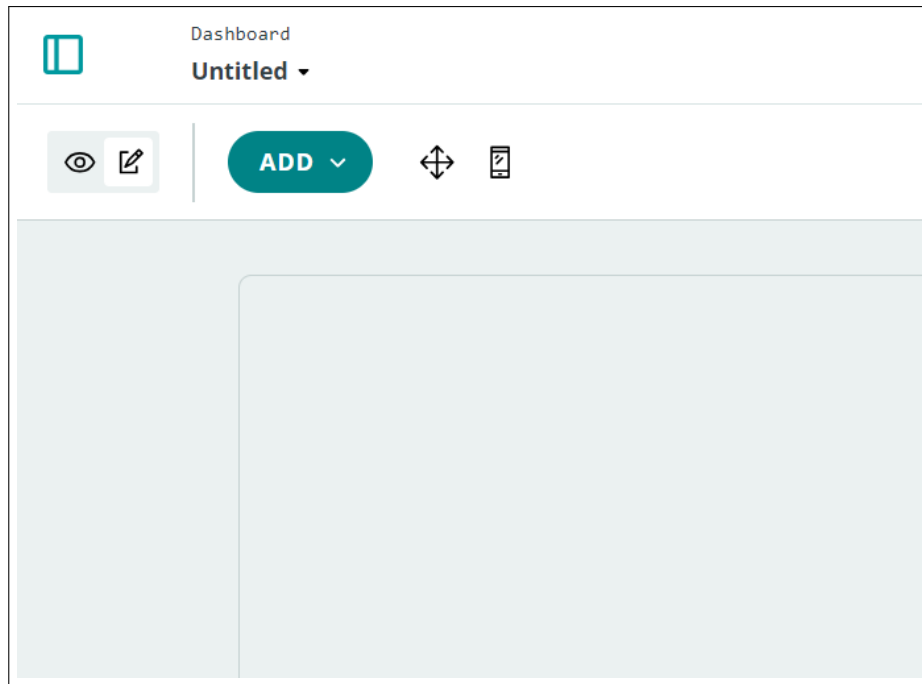
Dashboards

Search and filter Dashboards

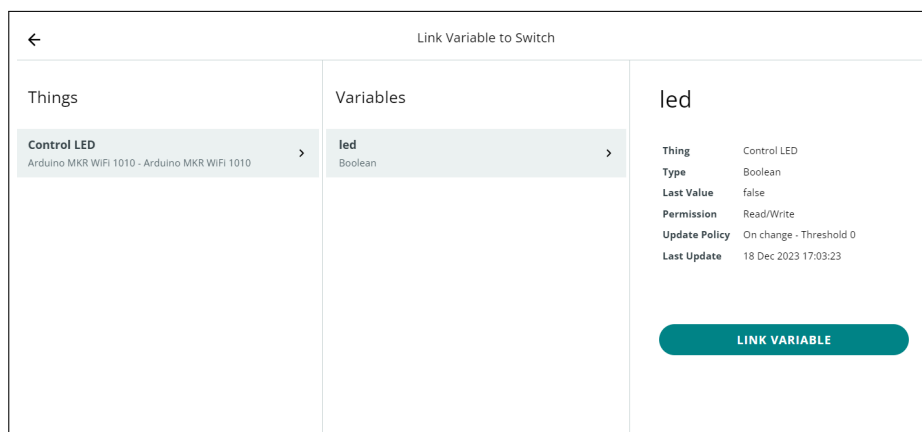
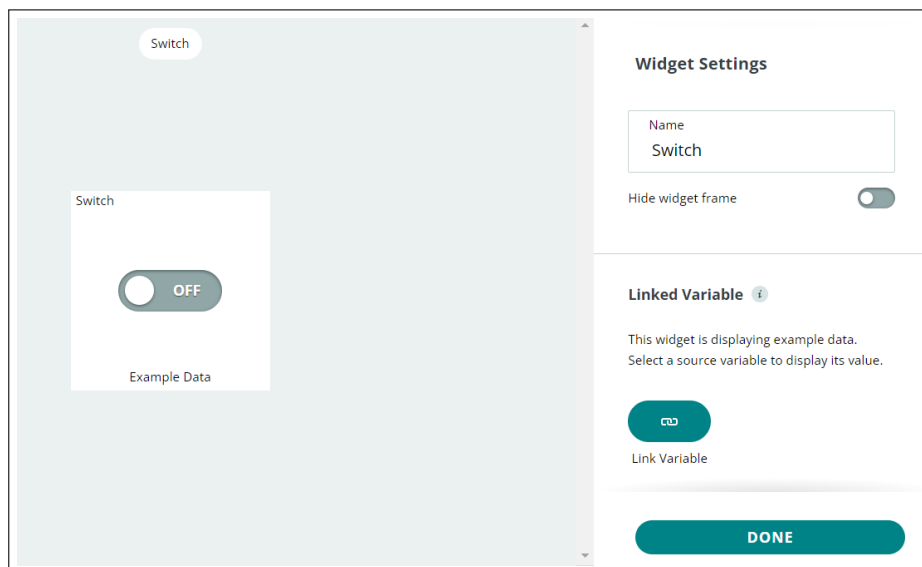
CREATE DASHBOARD

<input type="checkbox"/>	Name ↑	Last modified	People with access
<input type="checkbox"/>	Dashboard1	Dec 13, 2023, 3:55 PM	meriem soltani

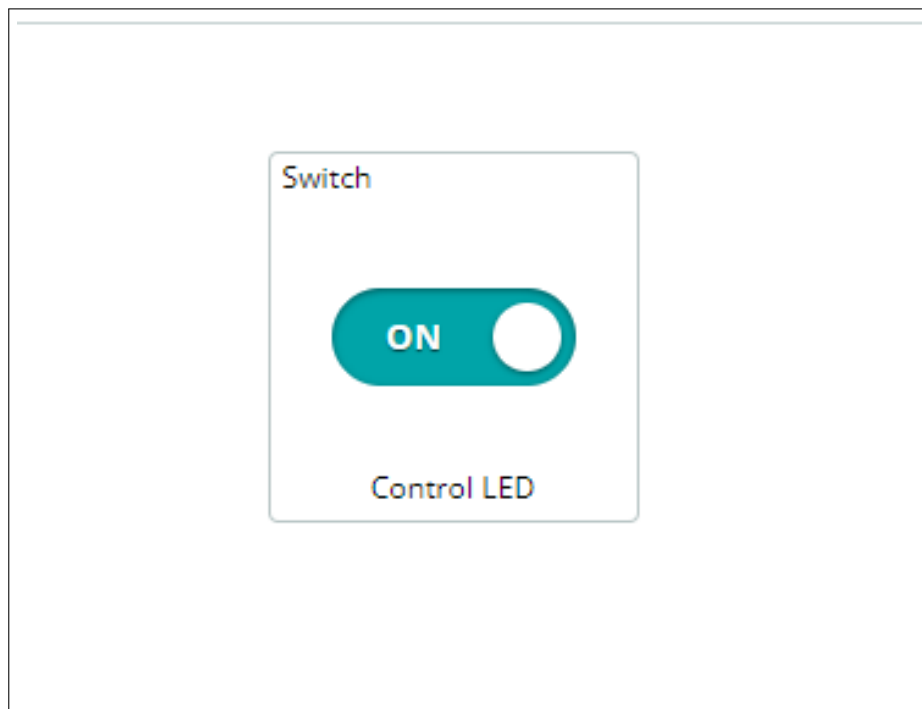
12. Add a switch element to the dashboard.



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

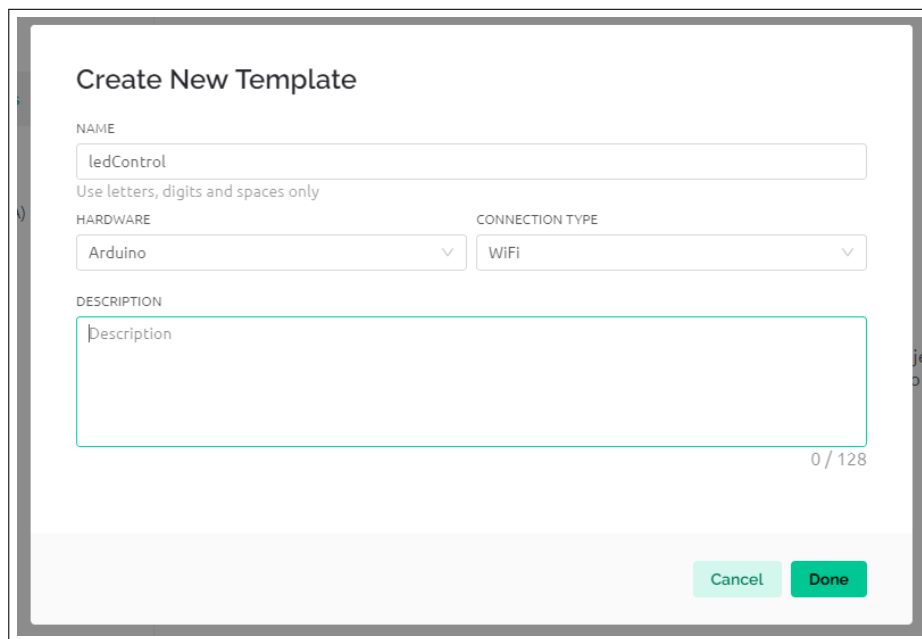
```
1 #include "thingProperties.h"
2
3 const int ledPin = 7;
4
5 void setup() {
6   Serial.begin(9600);
7   delay(1500);
8   pinMode(ledPin, OUTPUT);
9   initProperties();
10  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
11  setDebugMessageLevel(2);
12  ArduinoCloud.printDebugInfo();
13 }
14
15 void loop() {
16   ArduinoCloud.update();
17 }
18
19 void onLedChange() {
20   if (led)
21     digitalWrite(ledPin, HIGH);
22   else
23     digitalWrite(ledPin, LOW);
24 }
```

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'



Create New Template

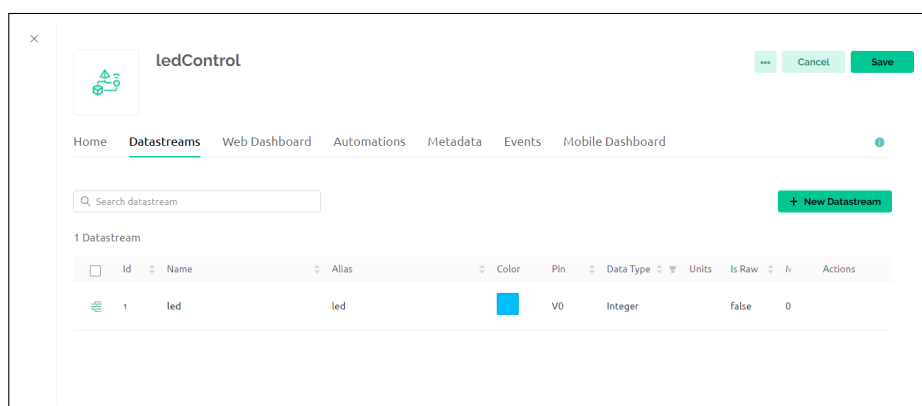
NAME
ledControl
Use letters, digits and spaces only

HARDWARE: Arduino CONNECTION TYPE: WiFi

DESCRIPTION
Description
0 / 128

Cancel Done

2. Add a new datastream



ledControl

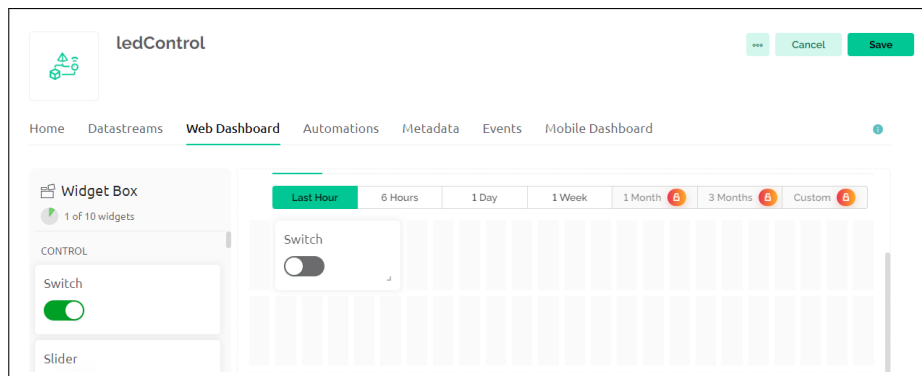
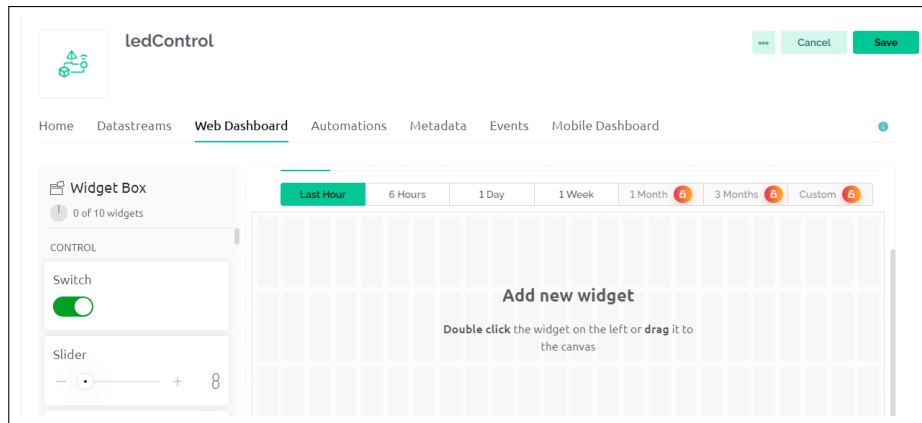
Home Datastreams Web Dashboard Automations Metadata Events Mobile Dashboard

Search datastream + New Datastream

1 Datastream

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	h	Actions
1	led	led		V0	Integer		false	0	

3. Add a new widget (Switch)



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

The screenshot shows the 'Create New Template' form. It has the following fields and options:

- NAME:** A text input field containing 'ledControl'. Below it, a note says 'Use letters, digits and spaces only'.
- HARDWARE:** A dropdown menu with 'Arduino' selected.
- CONNECTION TYPE:** A dropdown menu with 'WiFi' selected.
- DESCRIPTION:** A large text area containing the word 'Description'. A character count '0 / 128' is shown at the bottom right of the text area.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Done'.

5. Go to the Devices tab and add your device

New Device

Create new device by filling in the form below

TEMPLATE

ledControl

DEVICE NAME


arduino mkr

Cancel

Create

←

☰



arduino mkr Online ...

[Meriem](#) [My organization - 6279CP](#)

[Add Tag](#)

Dashboard

Timeline

Device Info

Metadata

Actions Log

STATUS

Online

DEVICE ACTIVATED

7:32 PM Today
by jm_soltani@esi.dz

AUTH TOKEN

3D27 - **** - **** - ****

SSL

No SSL

FIRMWARE VERSION

0.0.0

LAST UPDATED

7:42 PM Today

ORGANIZATION

My organization - 6279CP

TEMPLATE NAME

ledControl

IP

105.235.129.28

IP COUNTRY

Algeria

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPL28y2Akkc"
#define BLYNK_TEMPLATE_NAME "ledControl"
#define BLYNK_AUTH_TOKEN "3D27r26JGsrLJG_uIS2SLyH2_CR3_akk"

Template ID, Template Name, and AuthToken should be declared at the very top of the firmware code.
```

Devices

+ New Device

Start typing

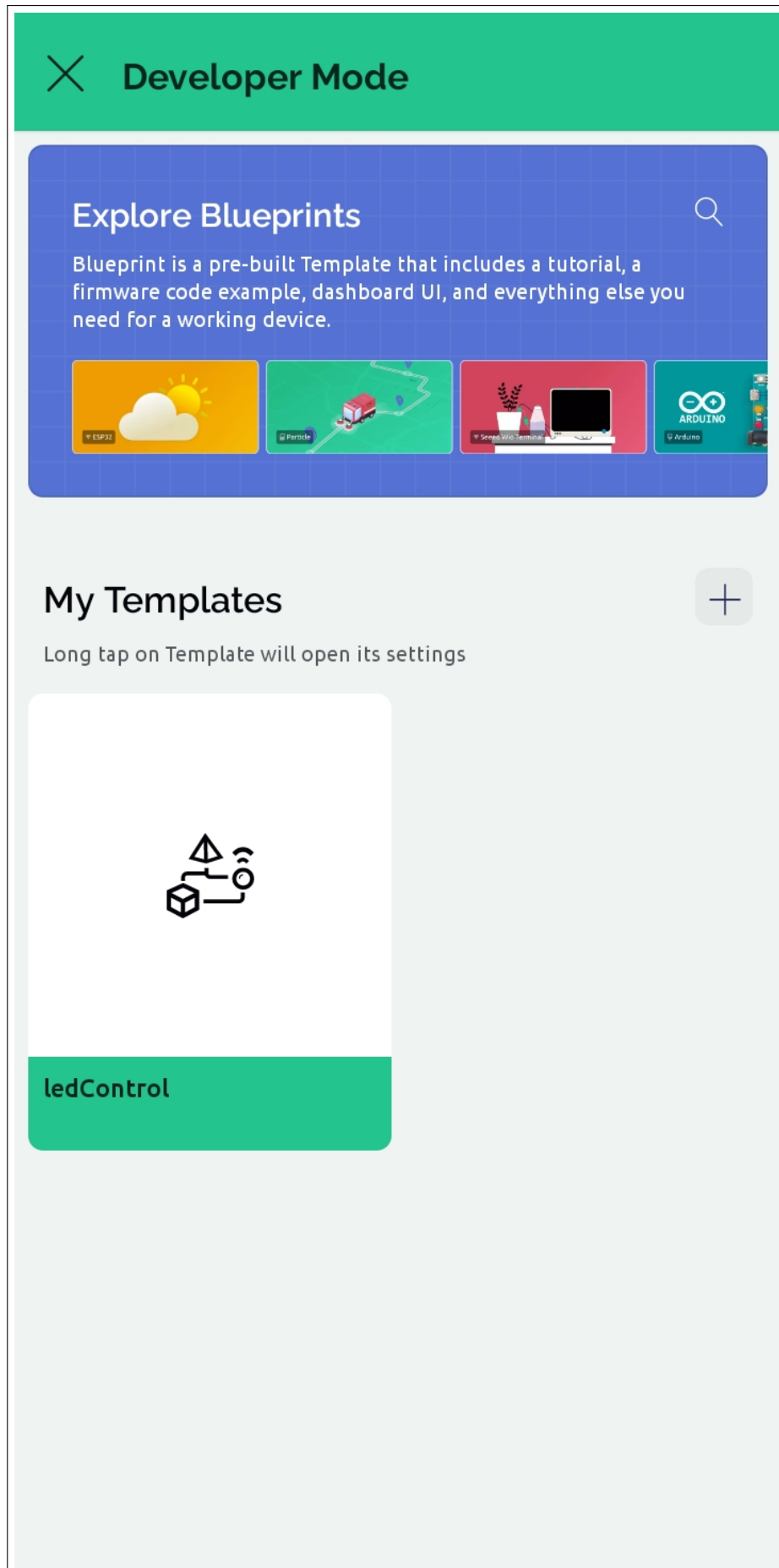
All 1

My devices 1

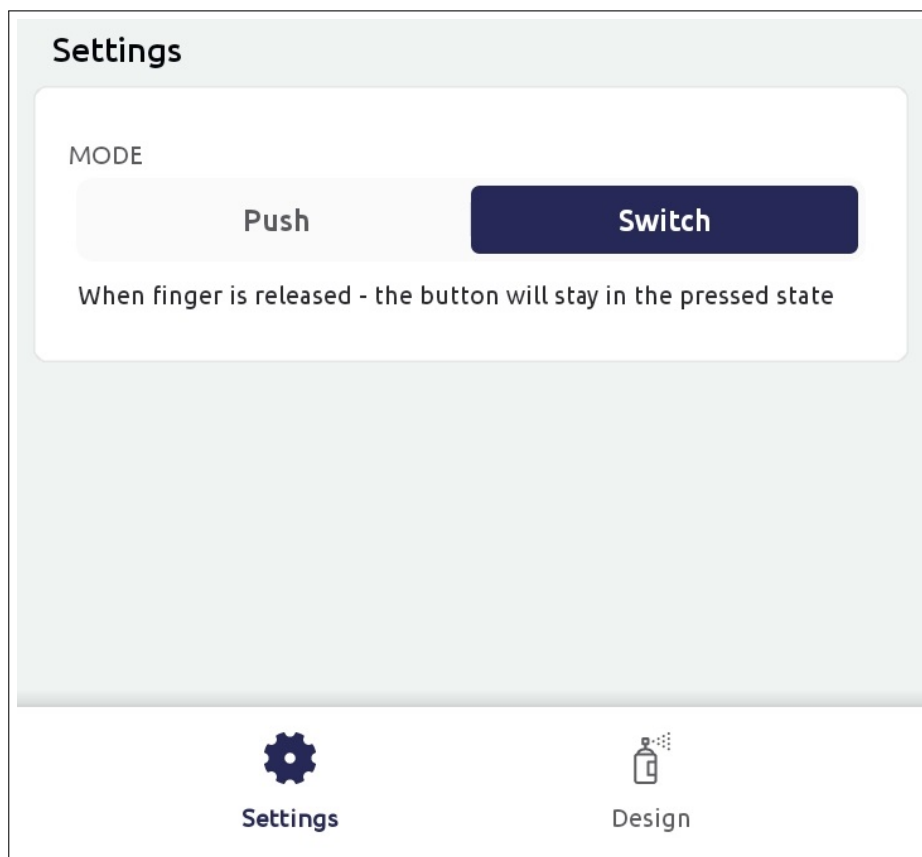
	Name	Auth Token	Device Owner	Status	Last Reported /	Actions
	arduino mkr	3D27r26JGsrLJG_uIS2SLyH2_CR3_...	jm_soltani@esi.dz (you)	Online	7:42 PM Today	



6.



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




8. Choose your widget and control your led .





 **Button Settings** 


 Preview

Data

DATASTREAM
led (V0)
Integer, 0/1, id=1 

OFF/ON VALUES
Use datastream's Min/Max 

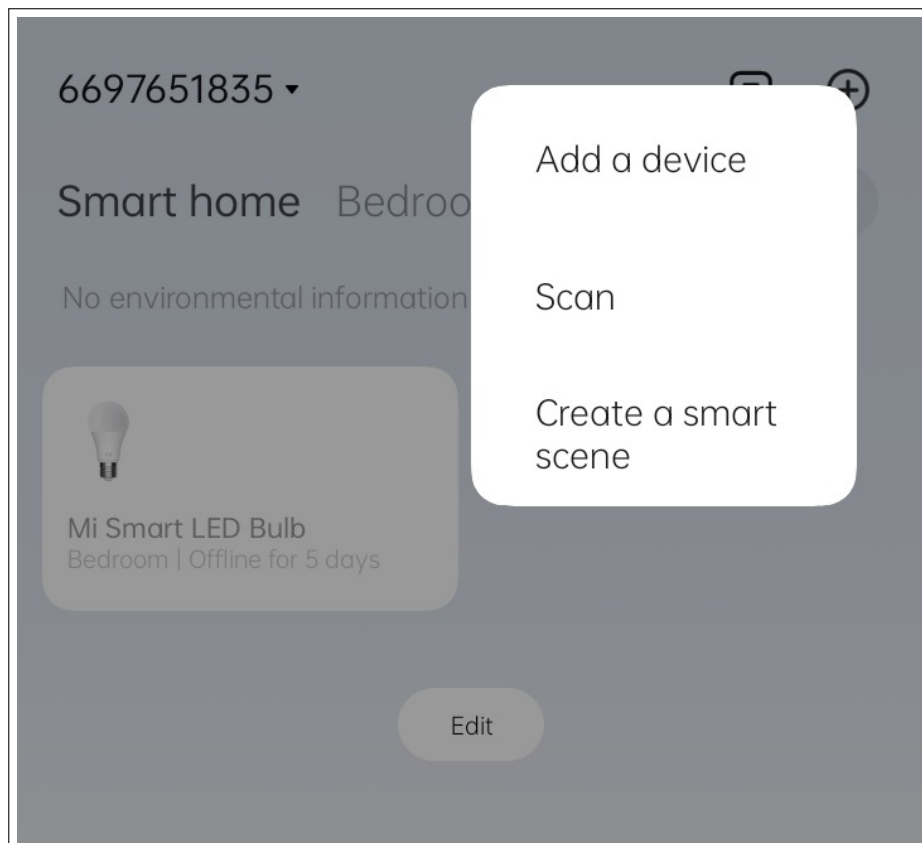
 **Arduino Mkr**



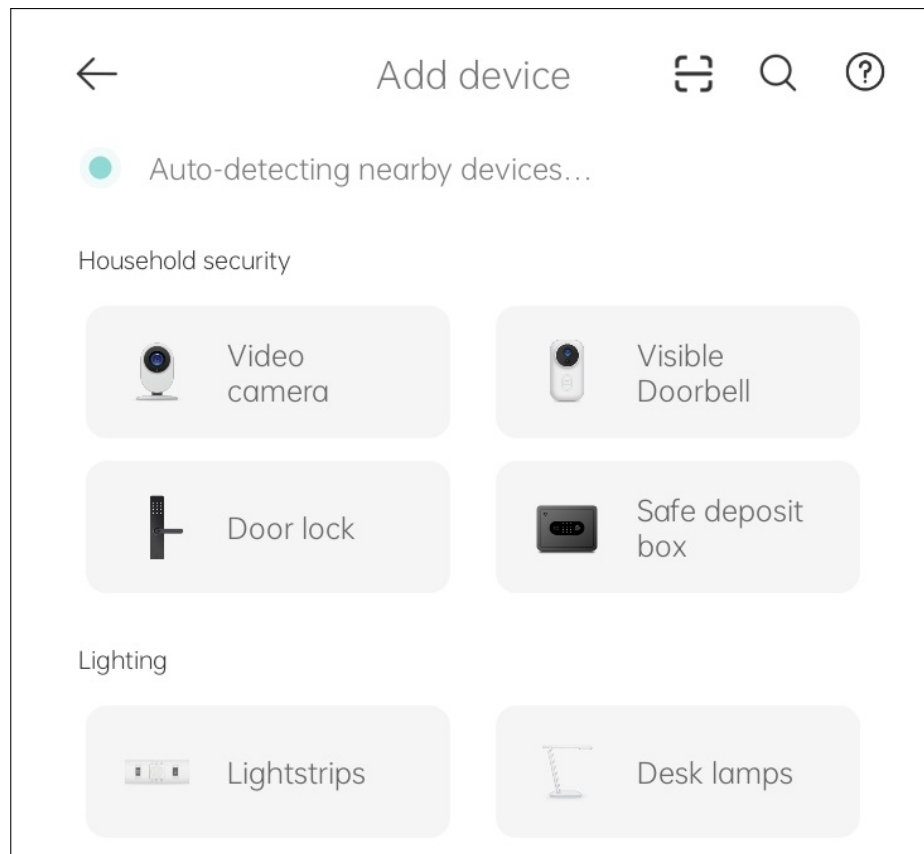
3. Xiaomi Smart Bulb

3.1. Set-up instructions

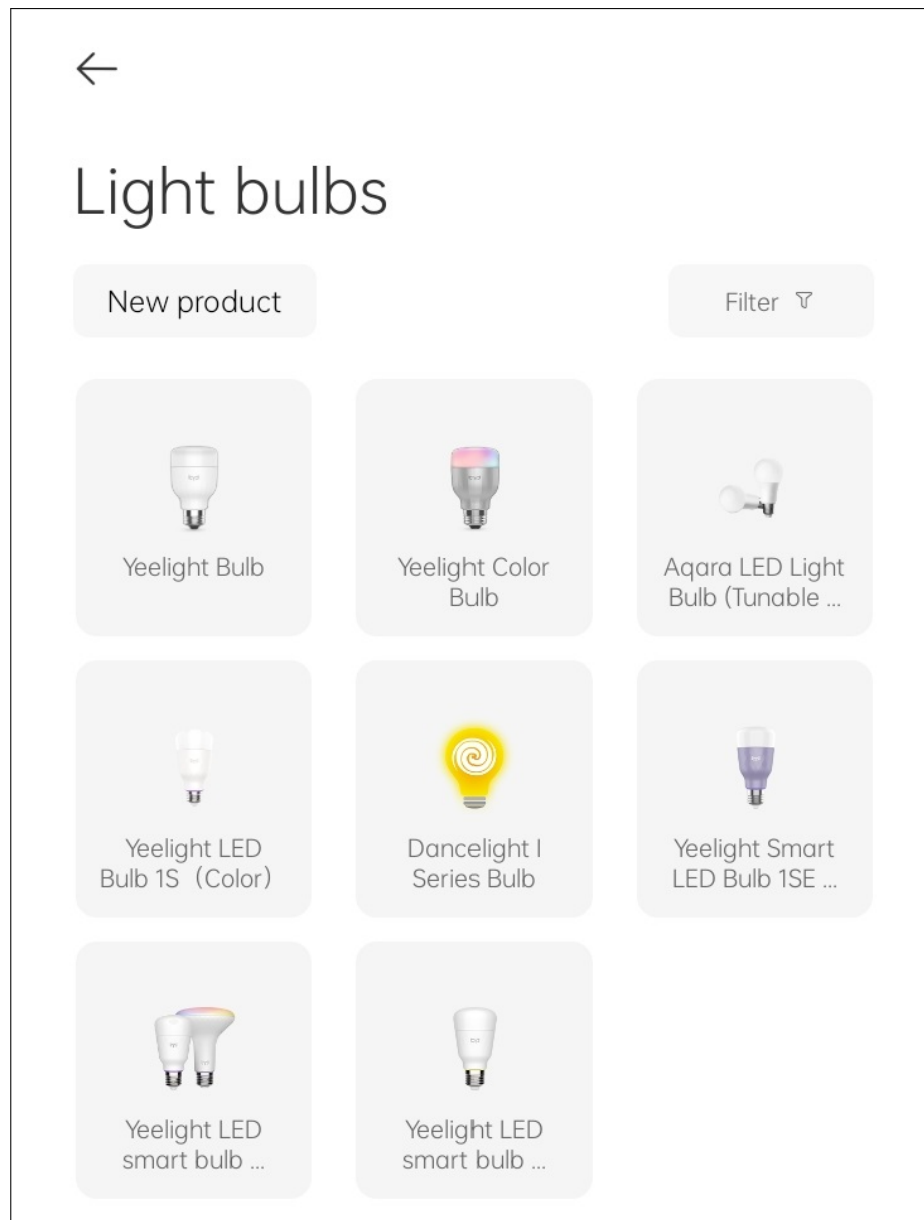
1. Install Mi Home App
2. Create or Sign In to Mi Account
3. Click on the Add device in your Mi Home



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



Reset device

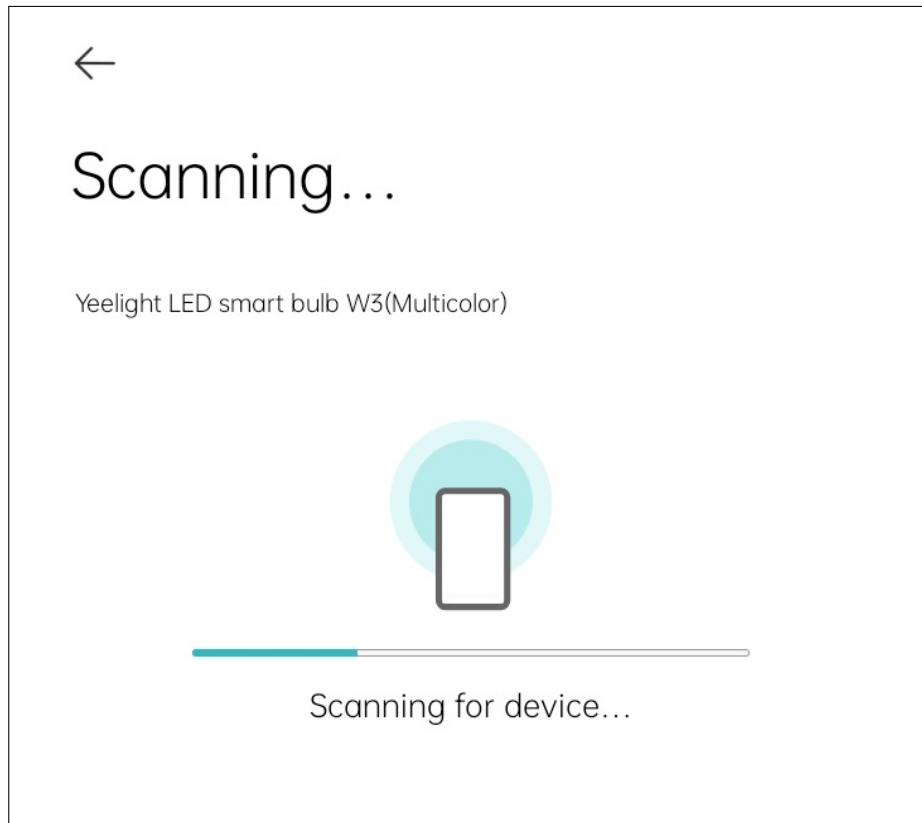
Yeelight LED smart bulb W3(Multicolor)

Turn on and off the bulb for 5 continuous times (turn on for 2 seconds and turn off for 2 seconds) until the bulb gives off colorful light. The bulb will reboot in 5 seconds and gives off white light, which shows the bulb has been factory reset.

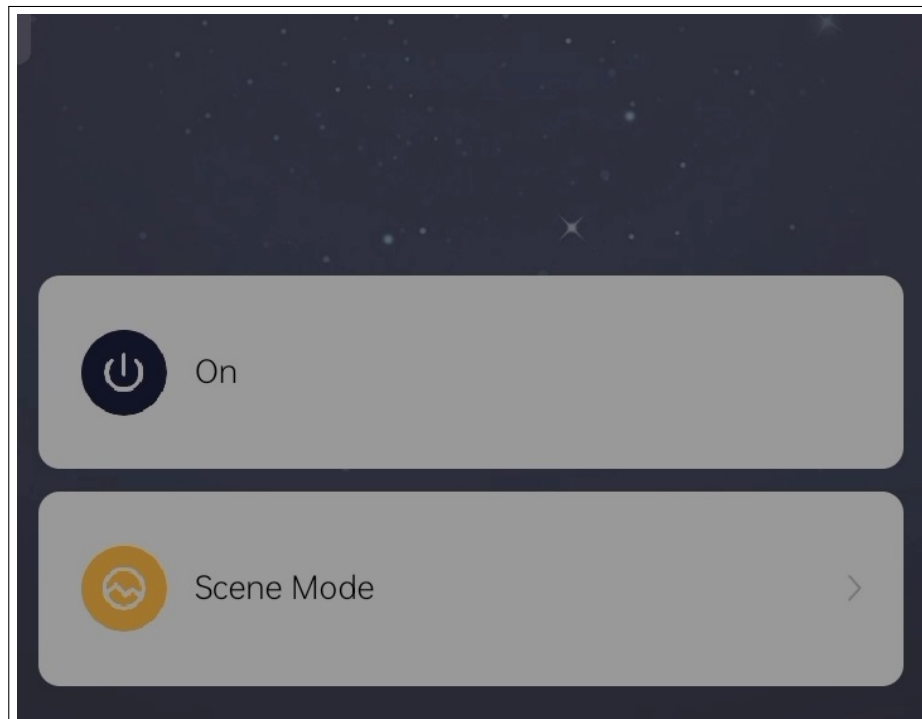


Device reset

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.