# Adafruit IO

**Connection**

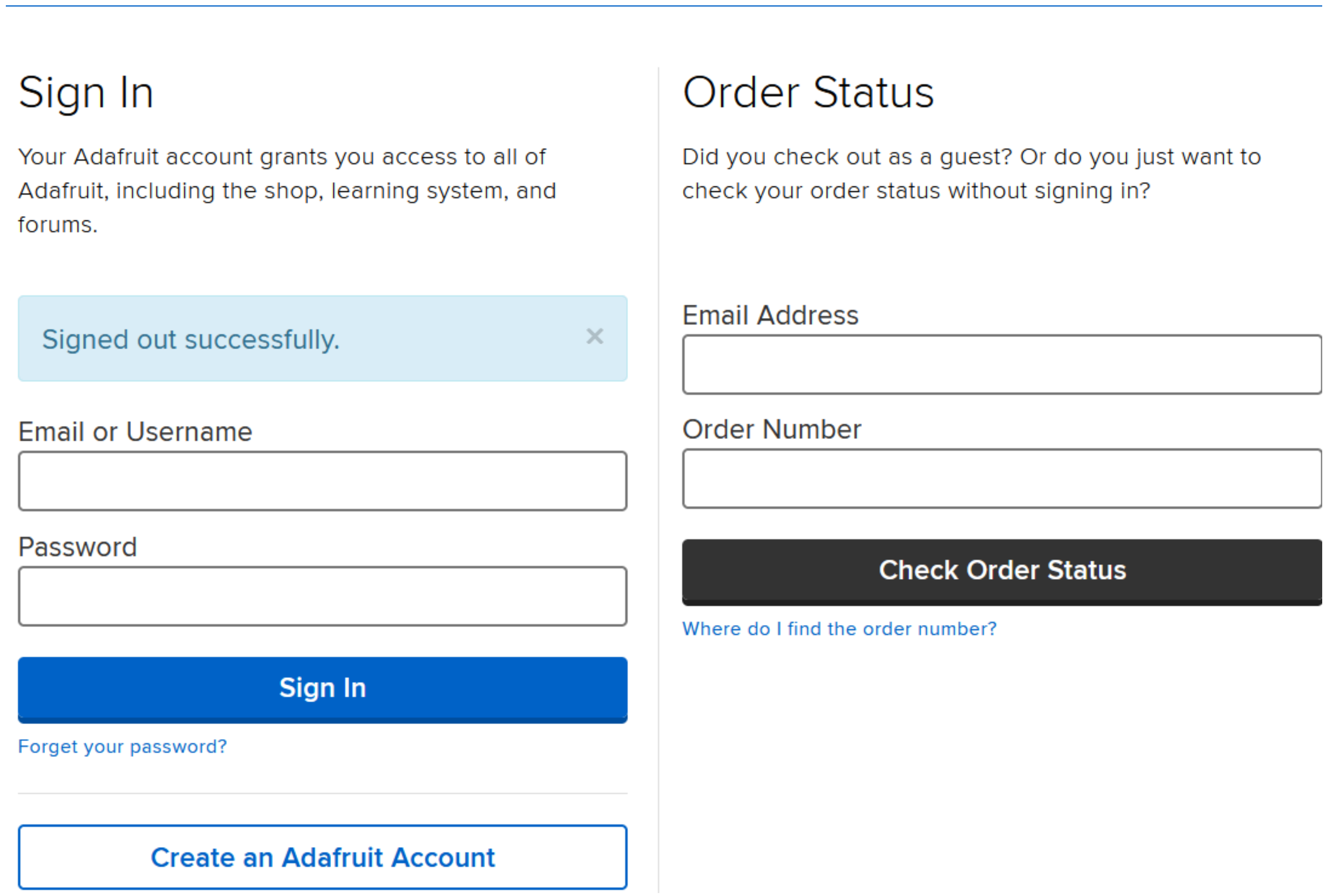
## What is Adafruit IO?

- Adafruit.io is a *cloud service* - that just means we run it and don't have to manage it. We can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

## Adafruit IO can do

- Display your data in real-time, online
- Make your project internet-connected: Control motors, read sensor data, and more!
- Connect projects to web services like Twitter, RSS feeds, weather services, etc.
- Connect your project to other internet-enabled devices
- The best part? All of the above is do-able for **free** with Adafruit IO

## Sign in


## Sign In

Your Adafruit account grants you access to all of Adafruit, including the shop, learning system, and forums.

Signed out successfully. ×

Email or Username

Password

## Order Status

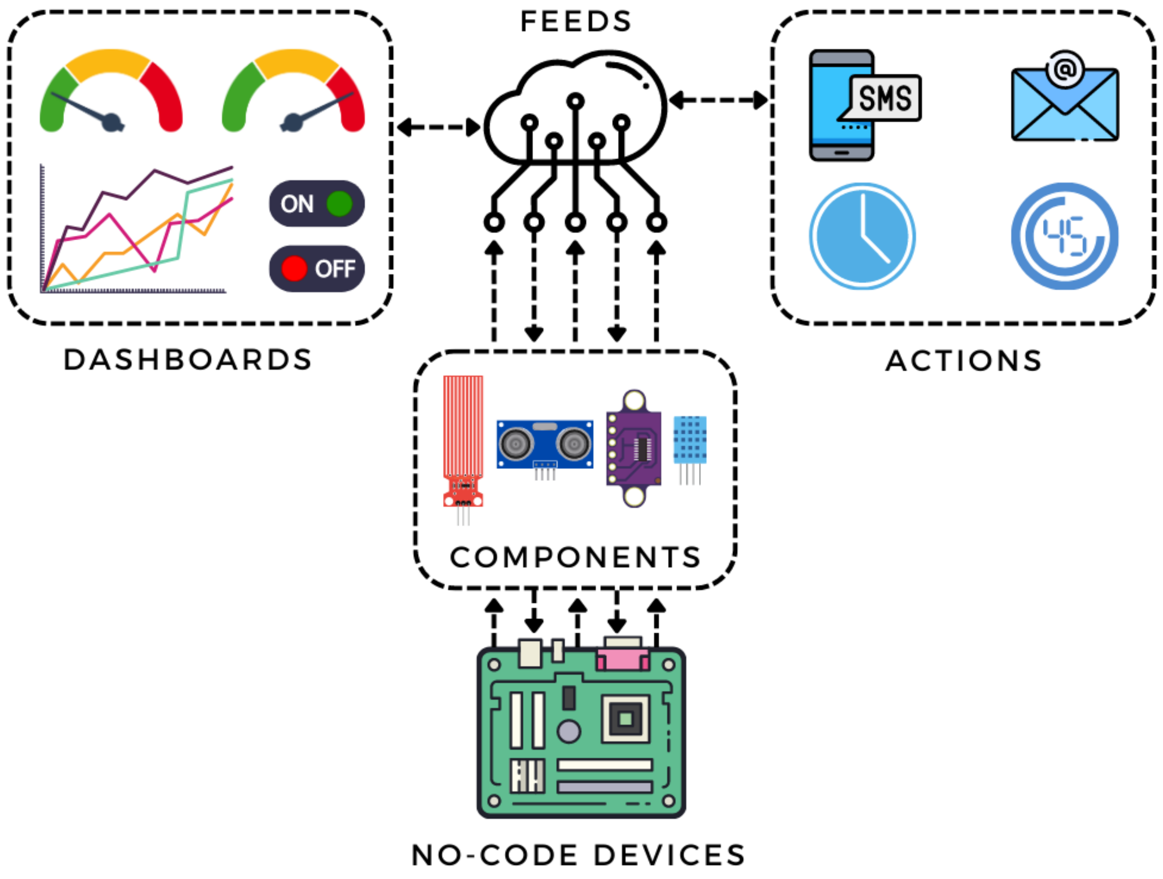
Did you check out as a guest? Or do you just want to check your order status without signing in?

Email Address

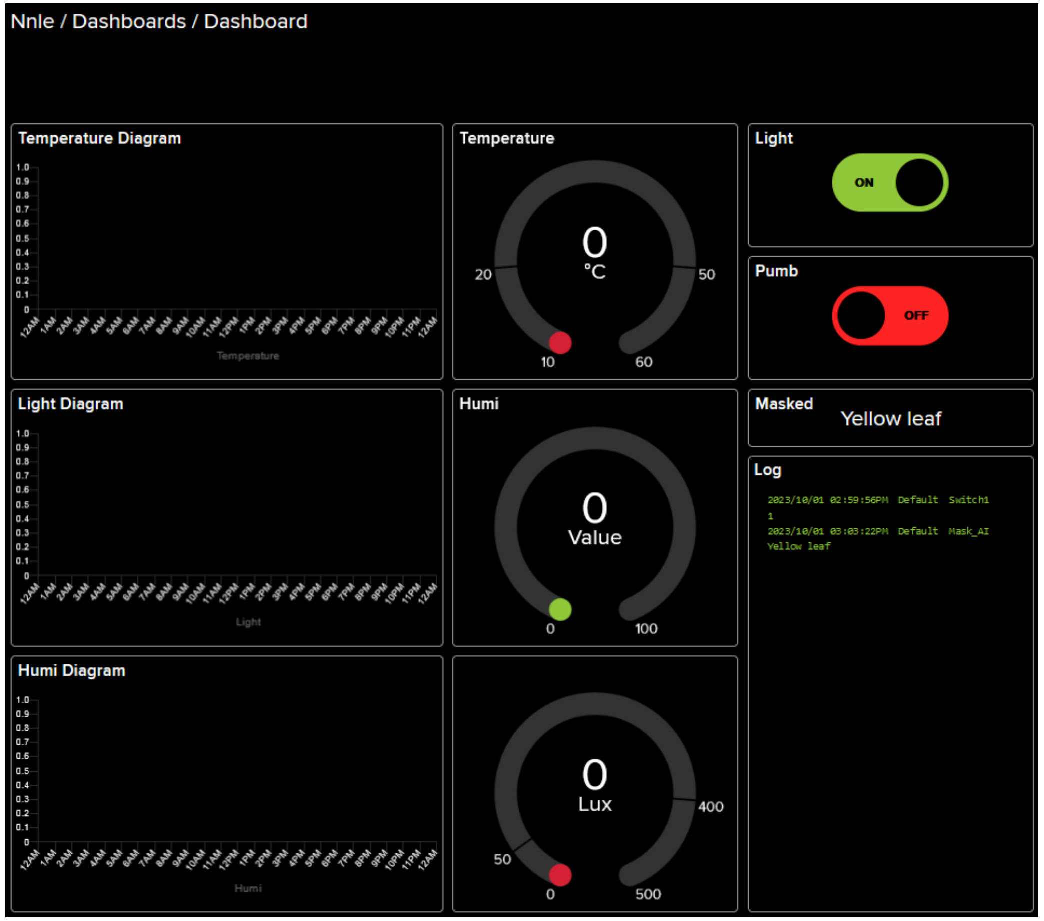
Order Number

INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

## Adafruit IO can do



# Dashboard Design

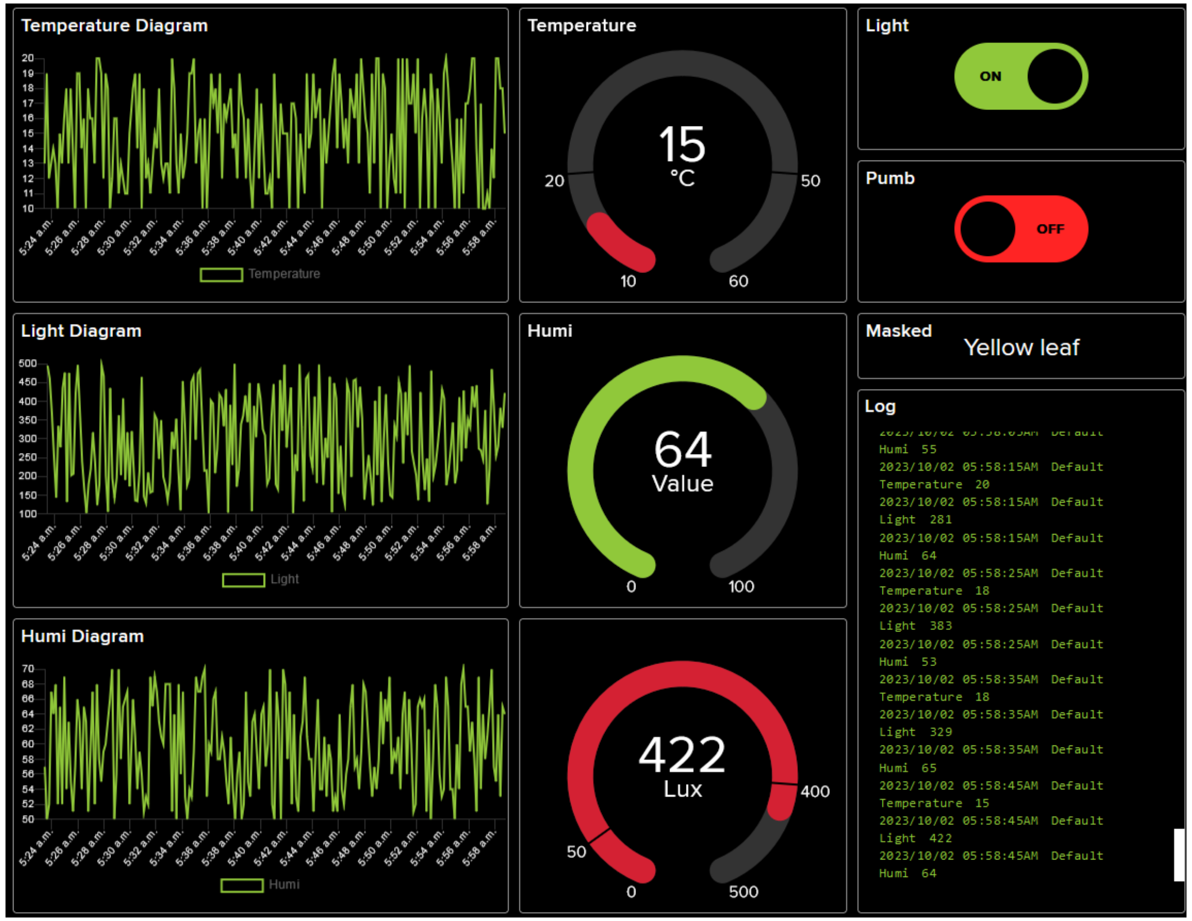


Masked: Yellow leaf

Log

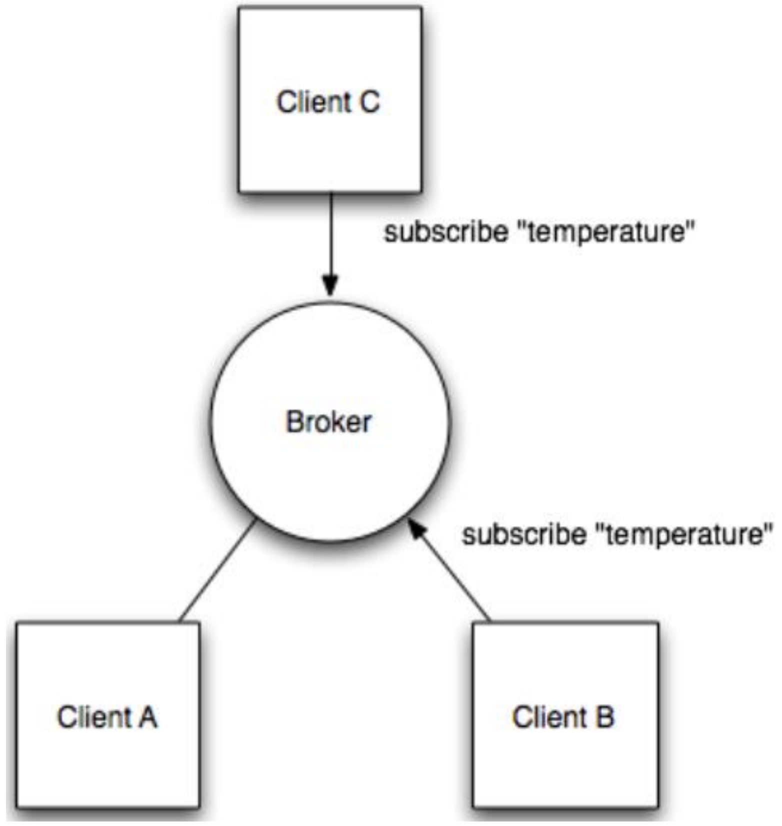
```
2023/10/01 02:59:56PM  Default  Switch1
1
2023/10/01 03:03:22PM  Default  Mask_AI
Yellow leaf
```

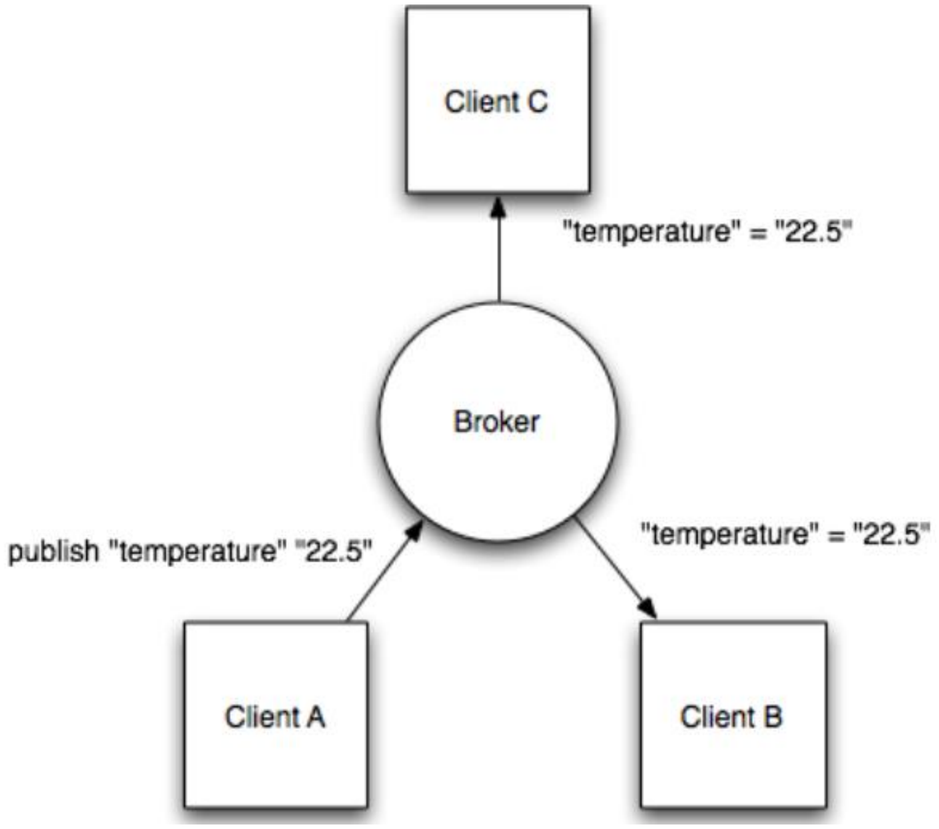
# Interact with your data



```
Humi  55
2023/10/02 05:58:15AM  Default
Temperature  20
2023/10/02 05:58:15AM  Default
Light  281
2023/10/02 05:58:15AM  Default
Humi  64
2023/10/02 05:58:25AM  Default
Temperature  18
2023/10/02 05:58:25AM  Default
Light  383
2023/10/02 05:58:25AM  Default
Humi  53
2023/10/02 05:58:35AM  Default
Temperature  18
2023/10/02 05:58:35AM  Default
Light  329
2023/10/02 05:58:35AM  Default
Humi  65
2023/10/02 05:58:45AM  Default
Temperature  15
2023/10/02 05:58:45AM  Default
Light  422
2023/10/02 05:58:45AM  Default
Humi  64
```

## MQTT Protocols





## Installing Adafruit IO Python Library

- Install the Adafruit IO Python Client Library to communicate with Adafruit IO.

```
pip3 install adafruit-io
```

## Library

- import sys
- from Adafruit_IO import MQTTClient

- AIO_FEED_ID = ""
- AIO_USERNAME = ""
- AIO_KEY = ""

## Define function

```
def connected(client):
    print("Ket noi thanh cong ...")
    client.subscribe(AIO_FEED_ID)
```

## Define function

```
def subscribe(client , userdata , mid , granted_qos):
    print("Subscribe thanh cong ...")
```

## Define function

```
def disconnected(client):
    print("Ngat ket noi ...")
    sys.exit (1)
```

## Define function

```
def message(client , feed_id , payload):
    print("Nhan du lieu: " + payload)
```

## Run

```
client = MQTTClient(AIO_USERNAME , AIO_KEY)
client.on_connect = connected
client.on_disconnect = disconnected
client.on_message = message
client.on_subscribe = subscribe
client.connect()
client.loop_background()
while True:
    pass
```

## Read and write with Serial port

- Install library: pyserial

## Read and write with Serial port

- import serial.tools.list_ports
- import time

## Read and write with Serial port

```
def getPort():
    ports = serial.tools.list_ports.comports()
    N = len(ports)
    commPort = "None"
    for i in range(0, N):
        port = ports[i]
        strPort = str(port)
        if "USB Serial Device" in strPort:
            splitPort = strPort.split(" ")
            commPort = (splitPort[0])
    return commPort
```

# Read and write with Serial port

- `ser = serial.Serial( port=getPort(), baudrate=9600)`
- `mess = ""`

# Read and write with Serial port

```
def processData(client,data):
    data = data.replace("!", "")
    data = data.replace("#", "")
    splitData = data.split(":")
    print(splitData)
    if splitData[1] == "T":
        client.publish("nhietdo", splitData[2])
    elif splitData[1] == "H":
        client.publish("doam", splitData[2])
```

# Read and write with Serial port

Receive data from device send to and then transmit to server

```
def readSerial():
    bytesToRead = ser.inWaiting()
    if (bytesToRead > 0):
        global mess
        mess = mess + ser.read(bytesToRead).decode("UTF-8")
        while ("#" in mess) and ("!" in mess):
            start = mess.find("!")
            end = mess.find("#")
            processData(mess[start:end + 1])
            if (end == len(mess)):
                mess = ""
            else:
                mess = mess[end+1:]
```

## Read and write with Serial port

```
def writeData2Hercule(data):
    // Your code here
```