

# Internet of Things (IoT)

Prof. Michalis Vlachos

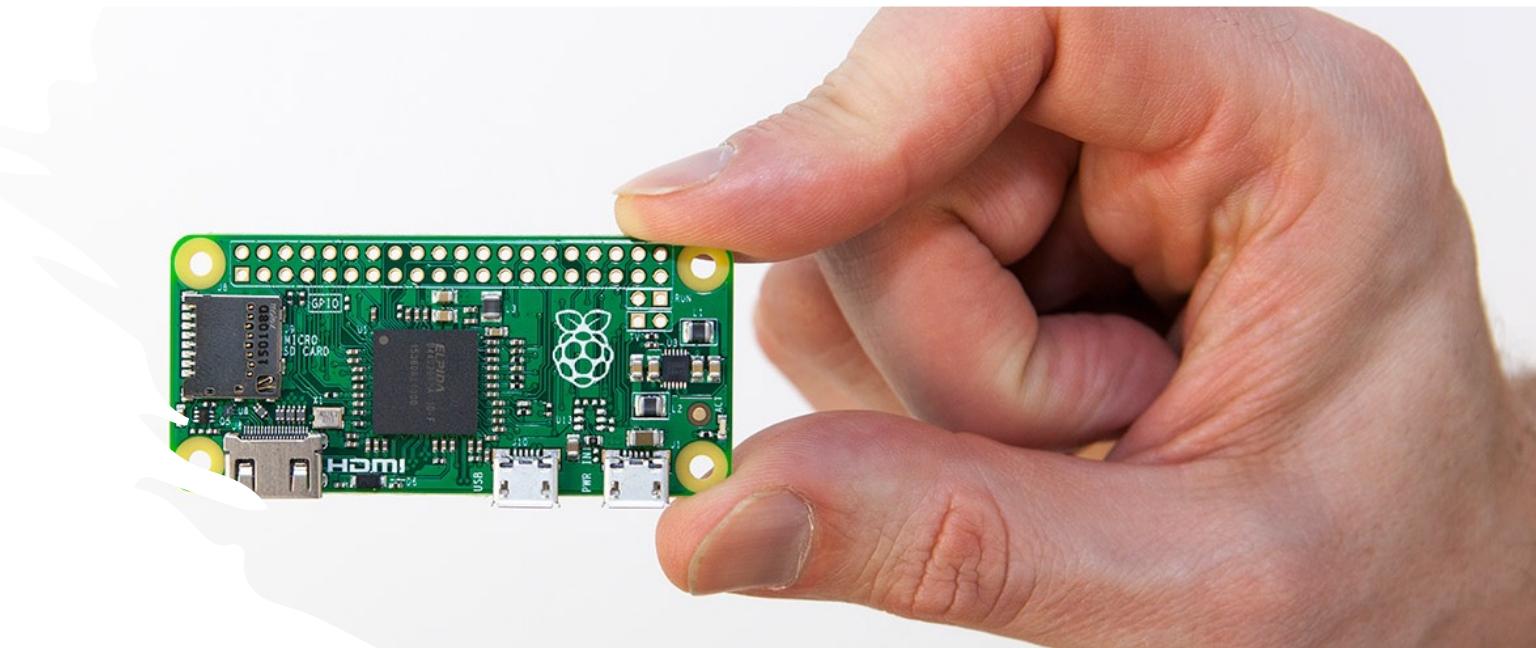


**UNIL** | Université de Lausanne

From Big  
(cloud)

to

small (IoT)



# Internet of Things (IoT)

- The **Internet of things (IoT)** describes physical objects (or groups of such objects) that are embedded with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.
- IoT is an application of existing technologies like – connectivity, security, storage, sensing – in new ways.
- Value of IoT: real commercial cases, utilitarian and engaging way.



# Why do you think we bother with IoT devices?

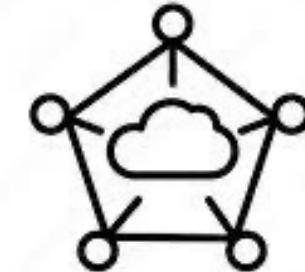
# IoT devices = Digital Tools (?)

- Humans, since the beginning of their existence, build *tools*.
- Such devices allow us to build the **tools of the digital age**.

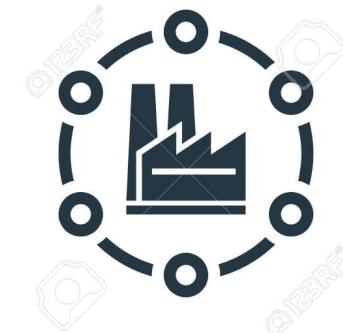


# Related terms

- **Edge computing** is “a distributed computing paradigm that brings computation and data storage closer to the sources of data. This is expected to improve response times and save bandwidth.”
- **Industry 4.0** “conceptualizes rapid change to technology, industries, and societal patterns and processes in the 21st century due to increasing interconnectivity and smart automation”



edge computing



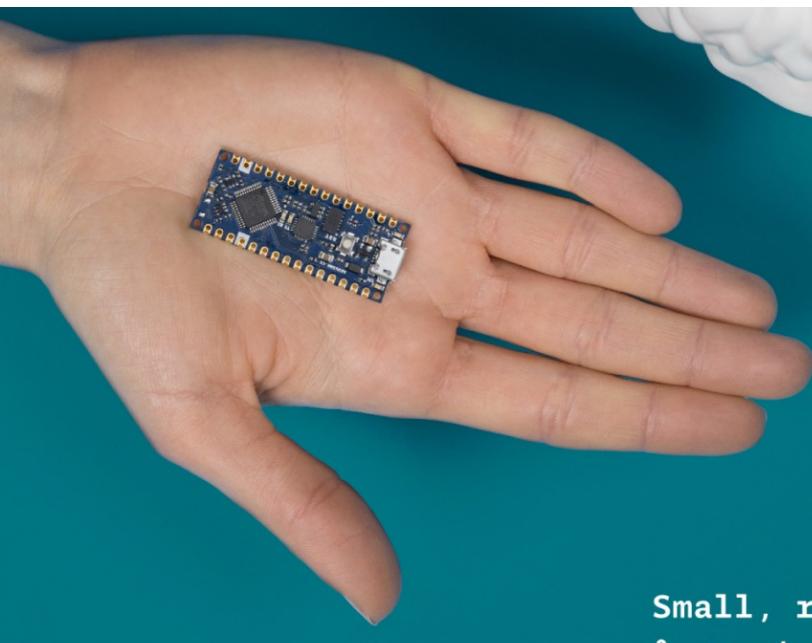
INDUSTRY 4.0

SmartHome

## Maker Economy

“Puts power in the hands of people to fund, design, prototype, produce, manufacture, distribute, market, and sell their own goods.”

# The devices



Arduino  
(microcontroller)

Code in C (typically)  
or in Python  
(Most have no WiFi)



Raspberry Pi 4

Your tiny, dual-display,  
desktop PC replacement.

Raspberry pi 4  
(just like a computer)

Code in Python  
You boot linux



ESP32

Code in Scratch  
or Python

# Comparison of devices

|                   | ARDUINO         | RASPBERRY PI Zero                      | Raspberry Pi 4                         | ESP32<br>eg M5GO or Core2 |
|-------------------|-----------------|--|--|---------------------------|
| TYPE              | Microcontroller | Microprocessor / Single Board Computer | Microprocessor / Single Board Computer | Microprocessor            |
| CPU               | ATmega328P      | ARM Cortex                             | ARM Cortex                             | Dual-Core 32-bit LX6      |
| RAM               | 48 – 256 KB     | 512 MB                                 | 1 GB – 8 GB                            | 16 MB                     |
| WIFI              | Usually not     | YES                                    | YES                                    | YES                       |
| Programming ?     | C / MicroPython | Any Language (Python, C++,...)         | Any Language (Python, C++,...)         | C, MicroPython, Blockly   |
| Goal              | Control devices | Regular Computer                       | Regular Computer                       |                           |
| Power Consumption | 40 mA (0.2W)    | 120 mA (0.7W)                          | 500 - 1000 mA (5 W)                    | 130 mA                    |
| CPU Speed         | 16 - 48 MHz     | 1 GHz                                  | 1.5 GHz                                | 240 MHz                   |

# What we will use



- **Product Features**
- ESP32-based, built-in Bluetooth, WiFi
- 16M Flash, 8M PSRAM
- Built-in speaker, power indicator, vibration motor, RTC, I2S amplifier, capacitive touch screen, power button, reset button
- TF card slot (16G Maximum size)
- Built-in lithium battery, equipped with power management chip
- Independent small board built-in 6-axis IMU, PDM microphone
- M-Bus Socket & Pins
- Program Platform: UIFlow, MicroPython, Arduino

# What we will use

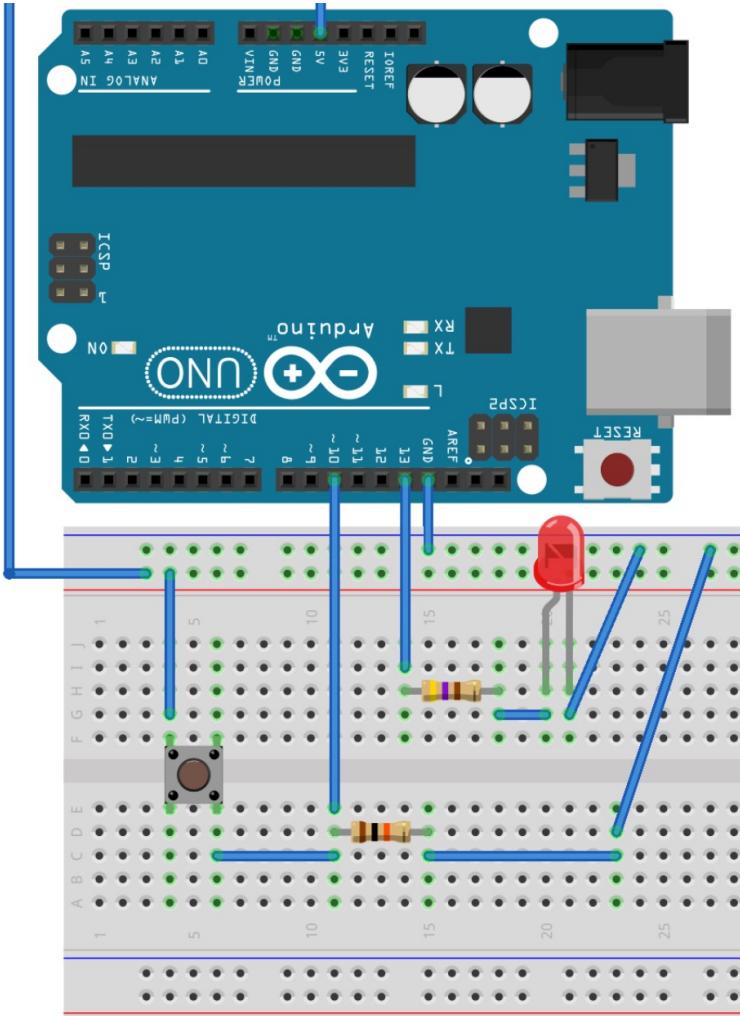


|                       |   |
|-----------------------|---|
| ESP32                 | 240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth |
| Flash Memory          | 16MB  |
| PSRAM                 | 4MB   |
| Power Input           | 5V @ 500mA  |
| Port                  | TypeC x 1,<br>GROVE(I2C+I/O+UART) x 1                               |
| IPS Screen            | 2 inch, 320x240 Colorful TFT LCD, ILI9342C                          |
| Speaker               | 1W-0928   |
| Button                | Custom button x 3   |
| MEMS                  | BMM150 + MPU6886  |
| Battery               | 500 mAh @ 3.7V  |
| Operating Temperature | 32°F to 104°F ( 0°C to 40°C )                                       |
| Size                  | 54.2 x 54.2 x 30.5mm  |
| Weight                | 62.3g   |
| Case Material         | Plastic ( PC )  |

# Development Environments

- Arduino IDE
- Python
- Micropython
- Scratch

# Connecting to the outside world!



The appeal of those devices is that you can connect it to and interact with the outside world

ESP32 (M5Stack)

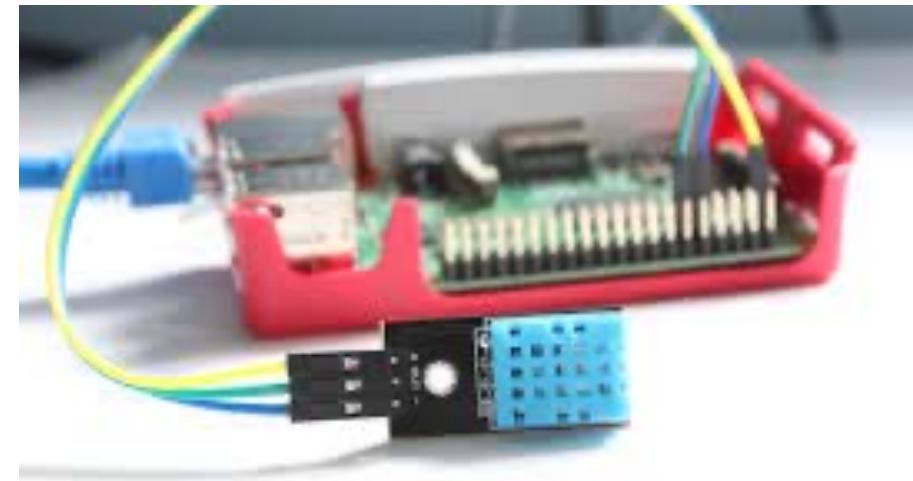
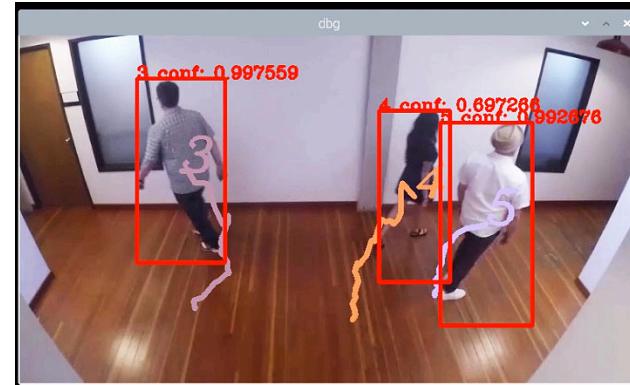


# Connectivity

Screen/e-ink



Camera



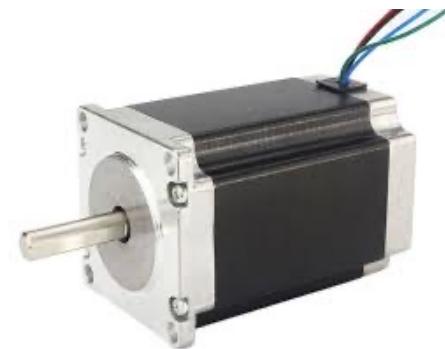
CO2/temperature  
sensors



GPS



Relay Switch  
(control high  
voltage devices)



Step  
motors

# Talk to your neighbor

- Brainstorm on some project you would like to do with such a device.

# Applications of IoT

Remote measurement

Smart watering

Smart Devices

Automation

Drones

Sustainability



## Smart watering

At a vast vineyard outside Modesto, Calif., E&J Gallo Winery is testing a new irrigation system developed with [IBM](#) to grow grapes using less water. The plan is to eventually apply the lessons learned to Watson so that IBM's data crunching technology can help farmers around the world.

Cutting water use can save huge amounts of money in the agriculture industry. It can also play a big role in water conservation, especially during droughts like the one that has plagued California for [several years](#).

"Water conservation has been a central focus," said Dr. Nick Dokoozlian, vice president, viticulture, chemistry, and enology at E&J Gallo. "Our stewardship of water continues."





• • •



## Smart Devices (Fridge)

**See what's inside your fridge from anywhere**

[View Inside<sup>4</sup> \(+ auto tagging of food expiration date\)](#)

# LIDAR AND ARCHAEOLOGY



## MAIN PARTS OF AN AIRBORNE LIDAR SYSTEM

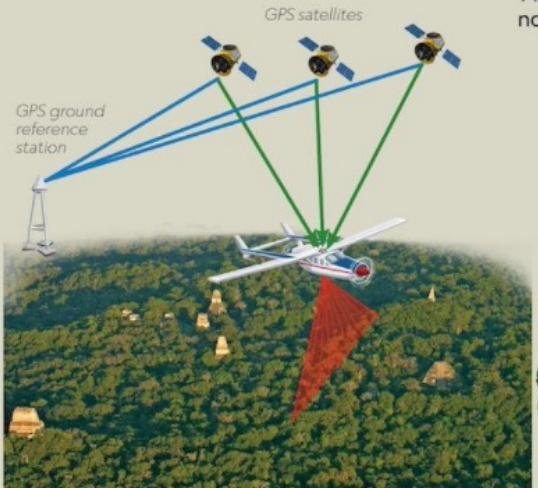
Plane/Helicopter/Drone—carries the moving technology

Laser—sends out an echo pulse

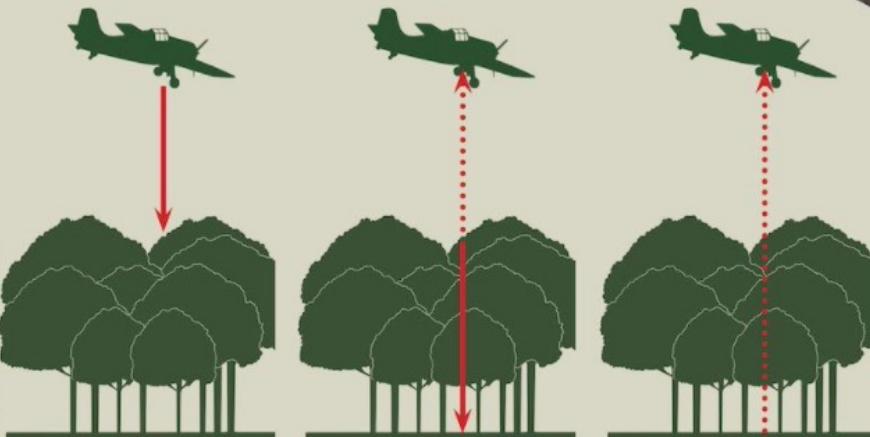
Laser Sensor—detects the reflected pulse to measure the range

Global Positioning System (GPS) ground reference and satellites—detect aircraft position

Inertial Measurement Unit system (IMU)—determines aircraft's orientation



A camera sees only the nearest objects, not what's underneath the canopy.

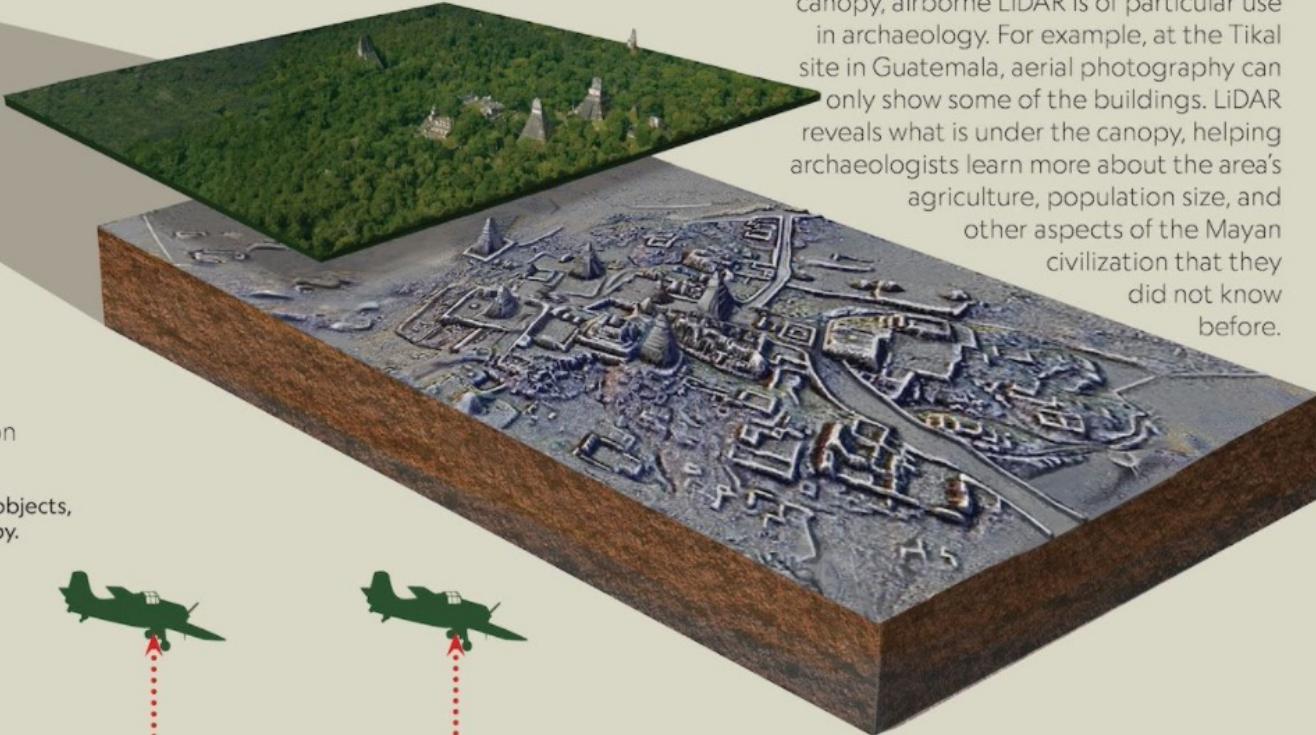


## WHAT IS LIDAR?

LiDAR stands for Light Detection And Ranging, a technology invented in the 1960s. It is a remote sensing technology system that collects data about ranges, or distances, using lasers. It has a number of benefits:

- It can more easily collect data over a large area.
- It can be used to create maps.
- It can map areas that are not visible to the human eye.

Because of its special features, LiDAR is now being used by archaeologists.



## UNCOVERING TIKAL

Because of its ability to "see" through a tree canopy, airborne LiDAR is of particular use in archaeology. For example, at the Tikal site in Guatemala, aerial photography can only show some of the buildings. LiDAR reveals what is under the canopy, helping archaeologists learn more about the area's agriculture, population size, and other aspects of the Mayan civilization that they did not know before.

## LIDAR IN ACTION

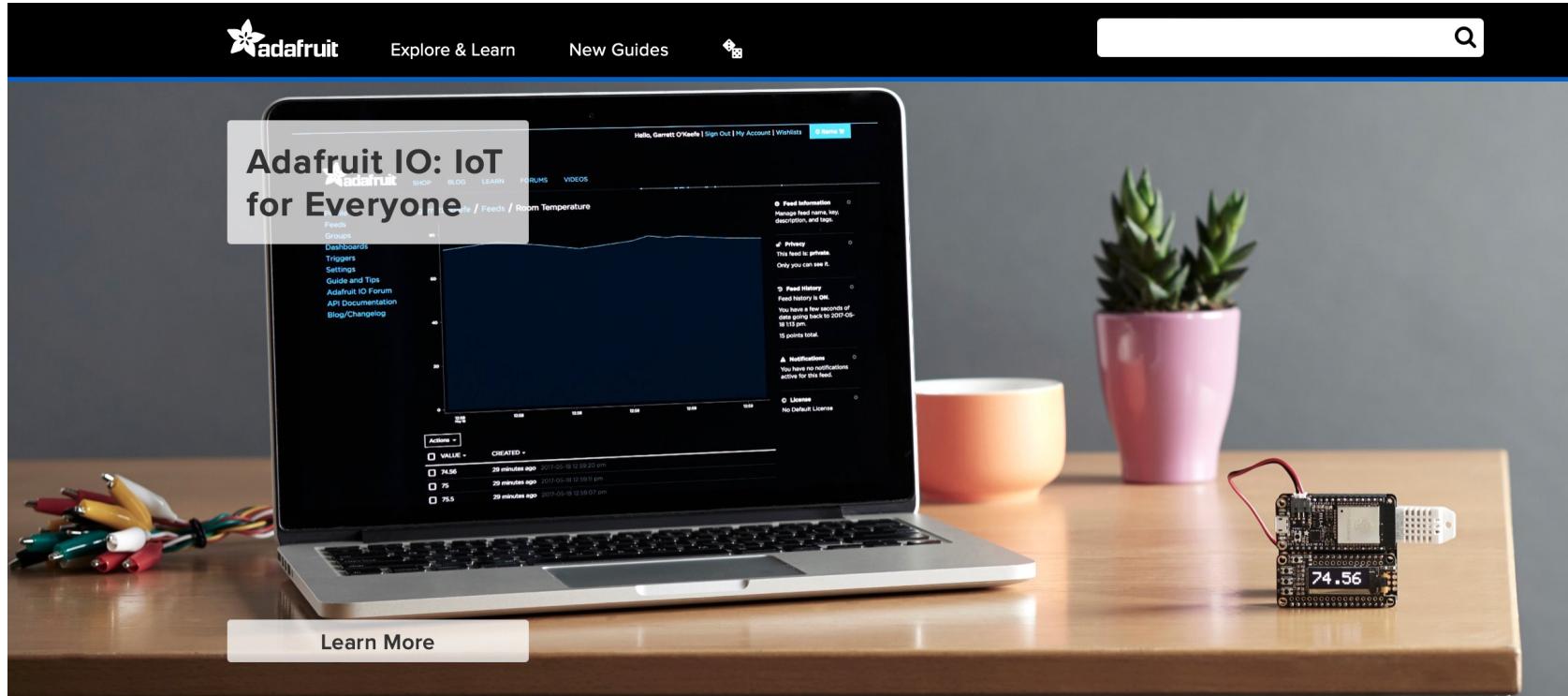
Vegetation limits aerial photography. It does not limit LiDAR. Where light can go, LiDAR can go. LiDAR pulses reflect from multiple points, starting at the first item they hit, and continue to the ground. So LiDAR can provide information about things that people cannot see.

# Aspects to consider when developing an IoT project...

- Communications (Internet needed or not?)
- Processor Power (how much power do I need?)
- Local Storage (RAM, EEPROM, Flash)
- Power Consumption (autonomy for how many hours/days)
- Functionality (GPIO interface, sensors, touch screen, ...)
- Cost (how many devices will I need?)

# Learning Resources

<http://learn.adafruit.com>



**Start Learning Electronics**

Tutorials and series to get you on your way



**Fun Electronic Projects**

Find inspiration here



**Electronic Skills Tutorials**

Tools for every project



# Learning Resources

<https://projects.raspberrypi.org/en/projects>

The screenshot shows the Raspberry Pi Projects website interface. At the top, there is a navigation bar with links for Home, All-new paths, Projects, Code Club, CoderDojo, Raspberry Jam, Sign In, and a language selector set to English. Below the navigation bar is a search/filter section titled "Find a Project:" with dropdown menus for "Any topic", "Any hardware", "Any software", and "Any level".

**3D adventure**  
Create a 3D game, virtual world, or visual novel with a goal and one or more endings  
Unity

**3D bug**  
Design a 3D-printable bug insect with BlocksCAD  
3D Printer, BlocksCAD, Web Browser

**3D key ring**  
Design a CODER keyring that can be 3D printed.  
3D Printer, BlocksCAD, Web Browser

# Learning Resources

- <https://m5stack.hackster.io/projects>
- [https://m5stack.com/project-hub?category\\_id=1&page=1](https://m5stack.com/project-hub?category_id=1&page=1)

The screenshot shows the Hackster.io project hub interface for M5Stack projects. At the top, there's a navigation bar with a search bar, a 'Add project' button, 'Log in', and a 'Sign up' button. Below the navigation is a row of filters: 'All products', 'All categories', 'Trending', 'Any difficulty', and 'Any type'. The main content area displays eight project cards in a grid:

- M5Stick Vario** by Olivier Wolf: A project using an M5Stick to track flight data. It has 2 likes and 14 comments.
- ESP32 Audio Project - Part II: Bluetooth Receiver Add-on** by Ernst Sikora: A project using an ESP32 to create a Bluetooth receiver. It has 1 like and 260 comments.
- ESP32 Audio Project - Part III: Create Song List using IFTTT** by Ernst Sikora: A project using an ESP32 to create a song list via IFTTT. It has 63 likes and 1 comment.
- M5Stack as sensor for Prometheus** by Enrico Casti: A project using an M5Stack as a sensor for a Prometheus monitoring system. It has 1 like and 253 comments.
- ego** by Framework Labs: A project featuring a mobile robot with an M5Stick. It has 2 likes and 219 comments.
- Plant Watering System** by C1 Group 1: A project using an M5Stack to control a plant watering system. It has 2 likes and 277 comments.
- Quality Assurance with TinyAutomator** by 7 members: A project using an M5Stack for quality assurance. It has 12 likes and 600 comments.
- Plant Monitoring System with M5Stack** by 3 members: A project using an M5Stack for plant monitoring. It has 3 likes and 132 comments.

# Other cool projects you can build

- Remote monitoring and automation for House/RV
- Smart supermarket
- Air quality monitor
- Portable maps for your bike (or any other vehicle)
- A speedometer for your vehicle
- ....get more ideas for projects [here](#)



# Hands-on Part

# M5Stack

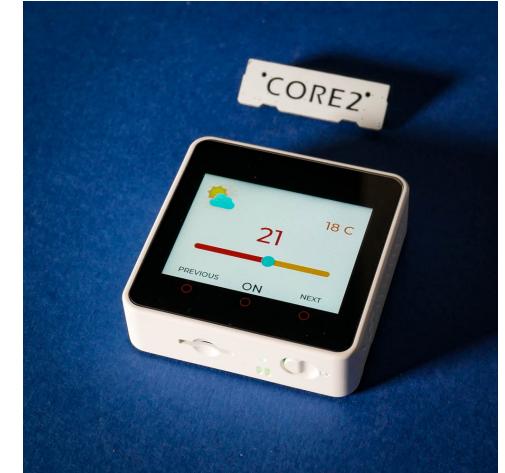
- Extendible devices
- Easy connect with sensors, etc
- Can be coded over the Internet!



M5Stick



M5Go



Core 2

# Documentation

- [https://docs.m5stack.com/en/quick\\_start/m5core/uiflow](https://docs.m5stack.com/en/quick_start/m5core/uiflow)
- [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/M5GO\\_Guide\\_English.pdf](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/M5GO_Guide_English.pdf)



| Hello M5

After completing the above steps, you can start programming with UIFlow. The following will demonstrate a simple program for you to drive the screen to display "Hello M5". (1. Place the label 2. Add the label block. 3 Click the run button in the upper right corner)

UI FLOW V1.4.5

Blockly </> Python

Execute code successfully.

Event

UI

Label

Screen

Hardwares

Units

Modules

FACES

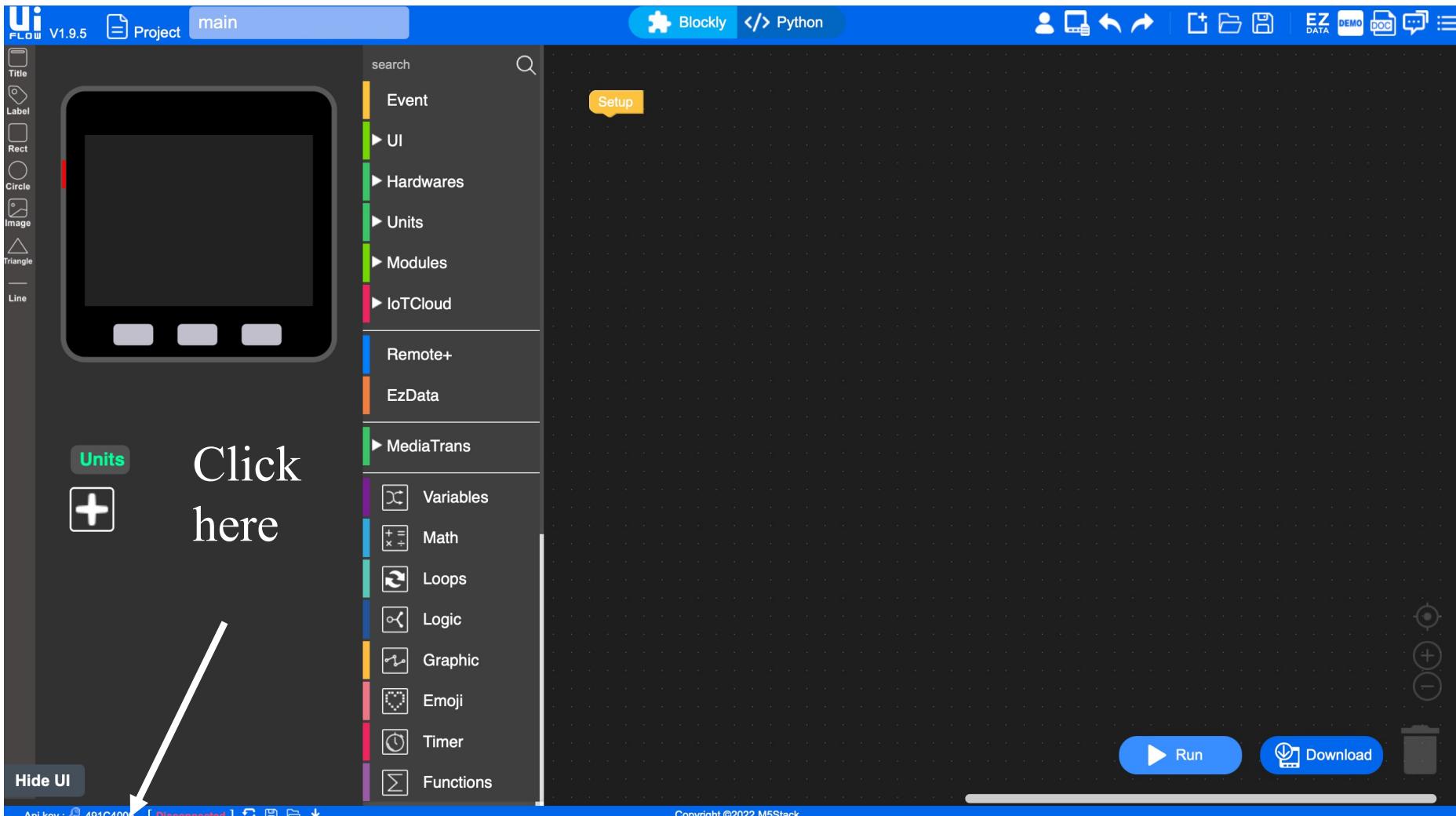
Variables

Setup

Label [label0] show "Hello M5"

The image shows the UIFlow software interface. It has a sidebar with categories like QUICK START, HARDWARES, DISPLAY, EXTENSIONS, IOT-CLOUD, M5 IOT SERVICE, ADVANCED, PROGRAMM, and CUSTOM. The main workspace shows a smartphone-like screen with the text "Text". A sidebar on the right lists blocks categorized by color: Event (blue), UI (orange), Label (red), Screen (green), Hardwares (yellow), Units (purple), Modules (pink), FACES (light blue), and Variables (dark blue). A message at the bottom right says "Execute code successfully." with a green bar. A "Setup" block is visible in the workspace, containing the command "Label [label0] show \"Hello M5\"".

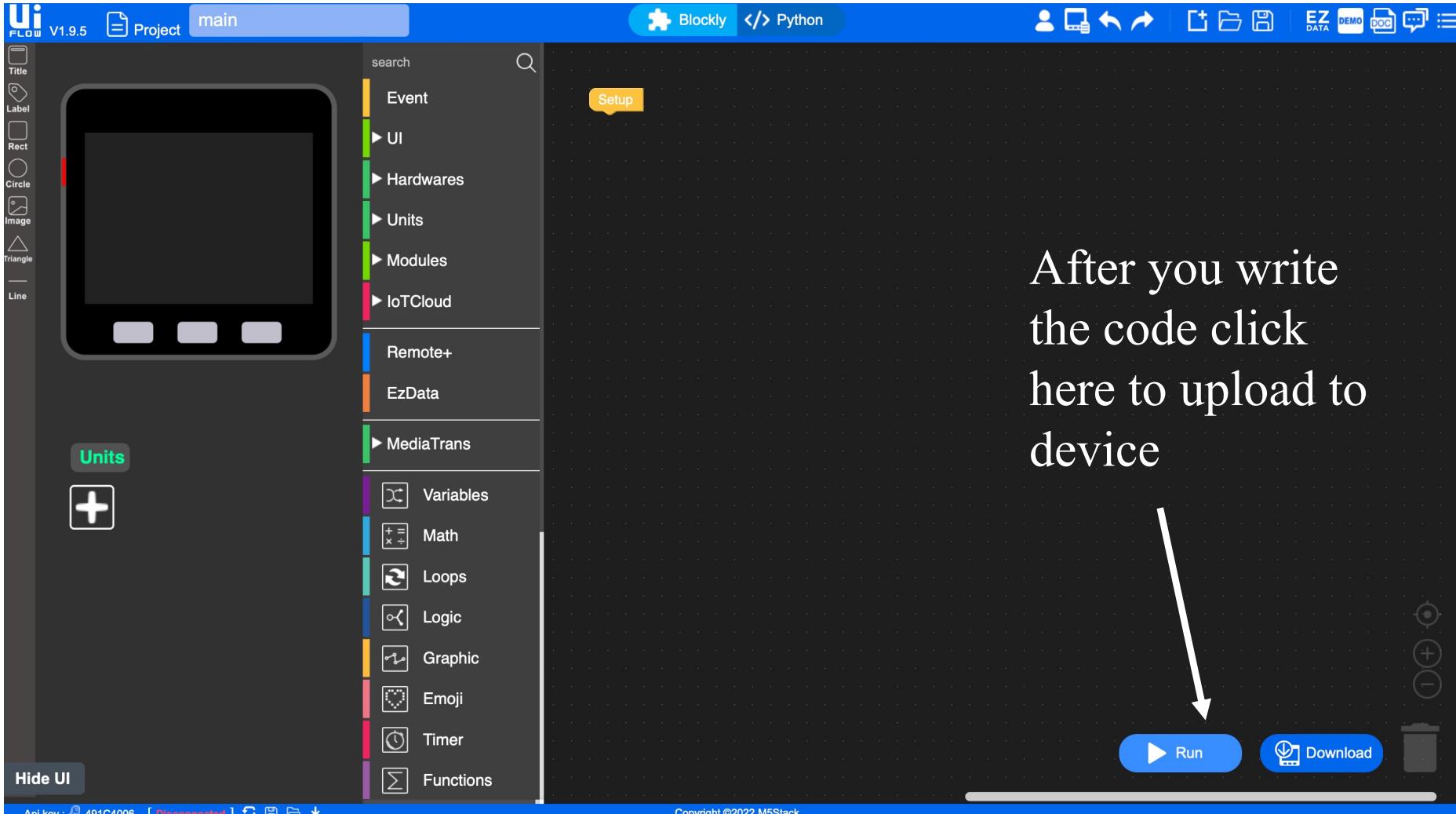
# http://flow.m5stack.com



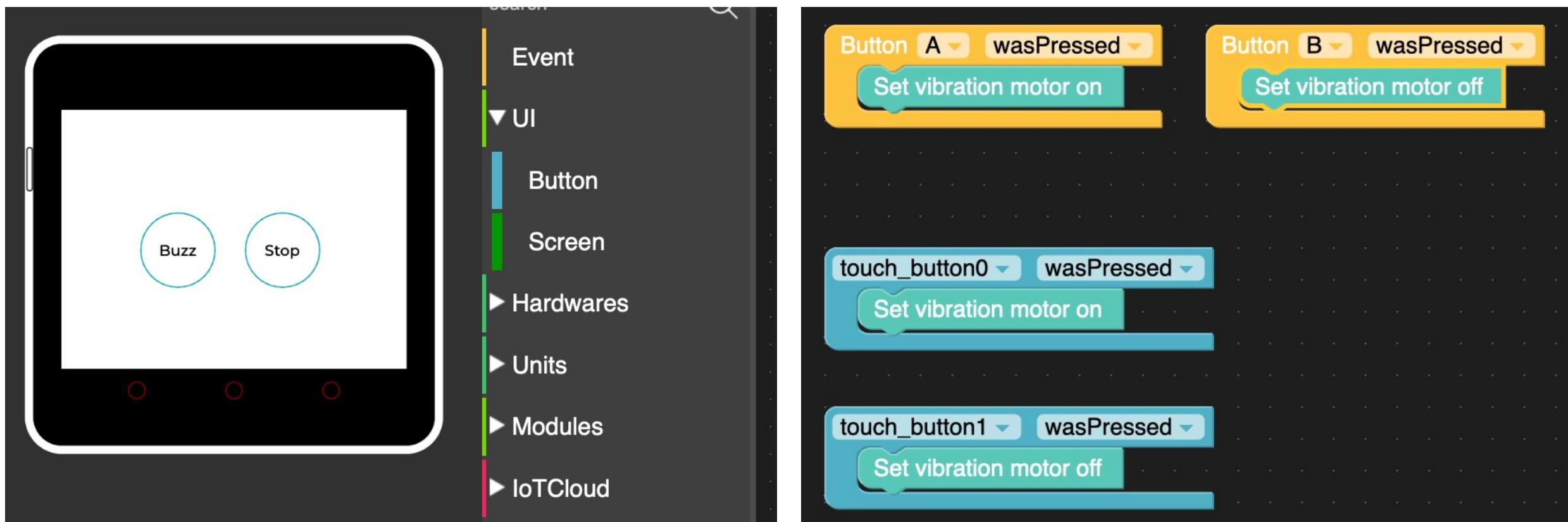
# Flow.m5stack.com



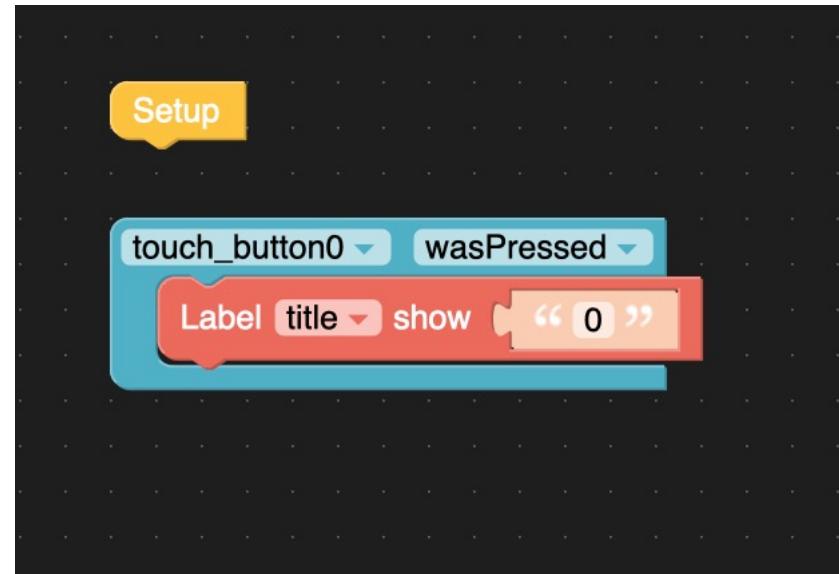
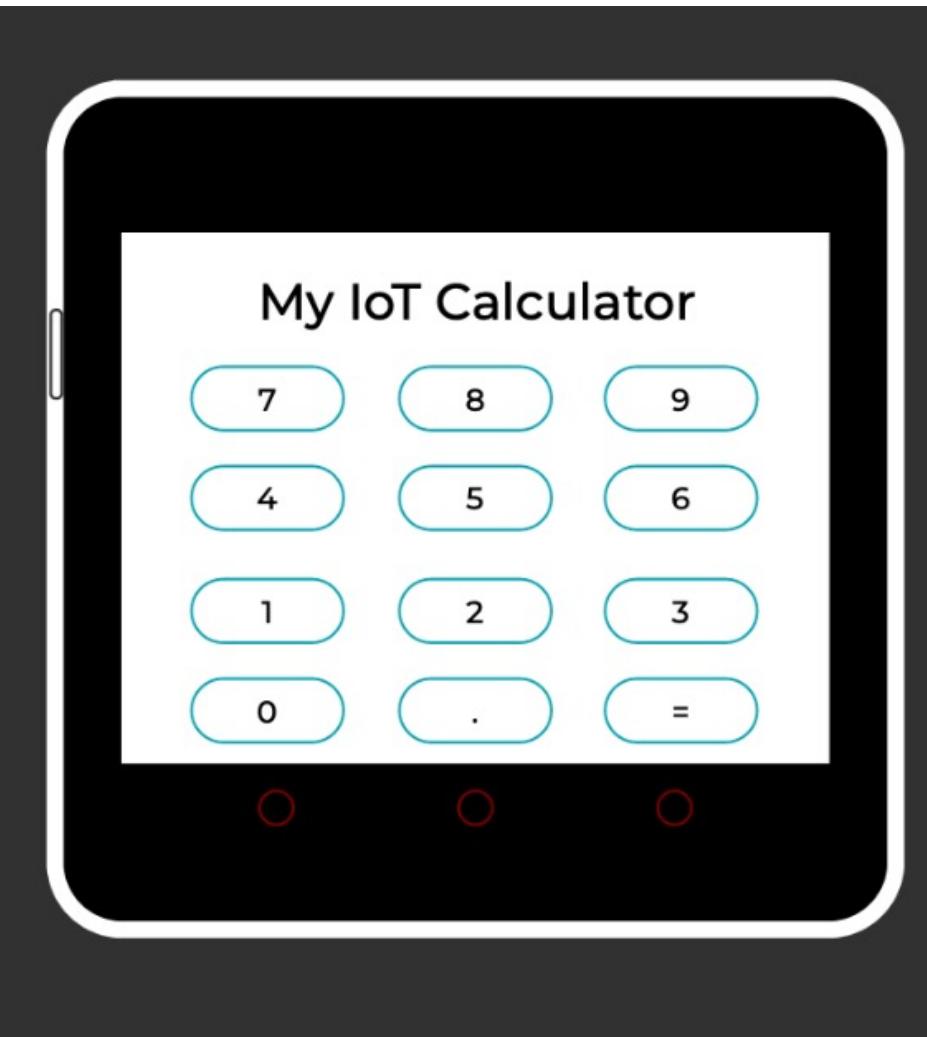
# Flow.m5stack.com



# Warm up: The “Hello World” of IoT



# Warm up: Use UIFlow to create and upload a simple interface, eg a calculator on the device



```
18 touch_button0 = M5Btn(text='0', x=125, y=50, w=70, h=30, bg_c=0xFFFFFFFF, text_c=0x000000,
19 touch_button9 = M5Btn(text='9', x=218, y=60, w=70, h=30, bg_c=0xFFFFF, text_c=0x000000,
20 touch_button = M5Btn(text='0', x=31, y=200, w=70, h=30, bg_c=0xFFFFF, text_c=0x000000,
21 touch_buttonComma = M5Btn(text='.', x=125, y=200, w=70, h=30, bg_c=0xFFFFF, text_c=0x000000,
22 touch_buttonEQ = M5Btn(text='=', x=218, y=200, w=70, h=30, bg_c=0xFFFFF, text_c=0x000000
23
24 def touch_button0_pressed():
25     # global params
26     title.set_text('0')
27     pass
28 touch_button0.pressed(touch_button0_pressed)
29
30
```

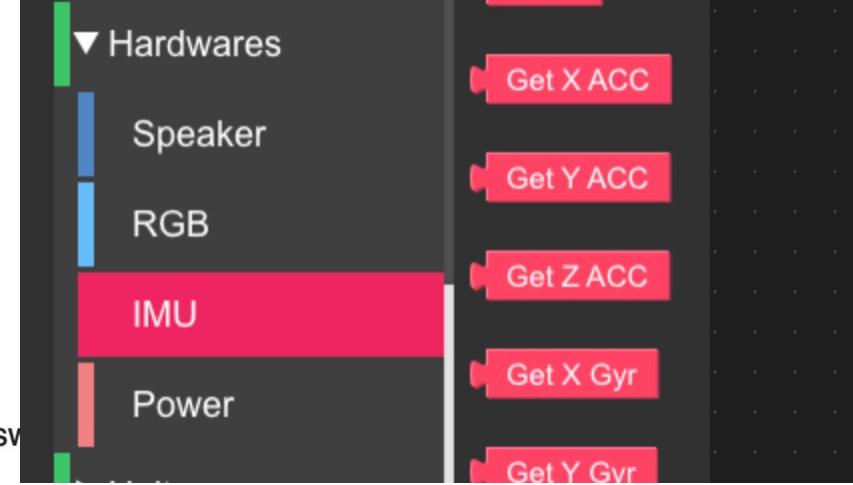
# Warm up: Magic 8 Ball



**Use the accelerometer of the device to create a magic 8 ball**

A standard Magic 8 Ball has 20 possible answers, including 10 affirmative answers (●), 5 non-committal answers (○), and 5 negative answers (●). The answers are:

|                       |                       |                              |                        |
|-----------------------|-----------------------|------------------------------|------------------------|
| ● It is certain.      | ● As I see it, yes.   | ○ Reply hazy, try again.     | ● Don't count on it.   |
| ● It is decidedly so. | ● Most likely.        | ○ Ask again later.           | ● My reply is no.      |
| ● Without a doubt.    | ● Outlook good.       | ○ Better not tell you now.   | ● My sources say no.   |
| ● Yes definitely.     | ● Yes.                | ○ Cannot predict now.        | ● Outlook not so good. |
| ● You may rely on it. | ● Signs point to yes. | ○ Concentrate and ask again. | ● Very doubtful.       |



Magic 8-Ball

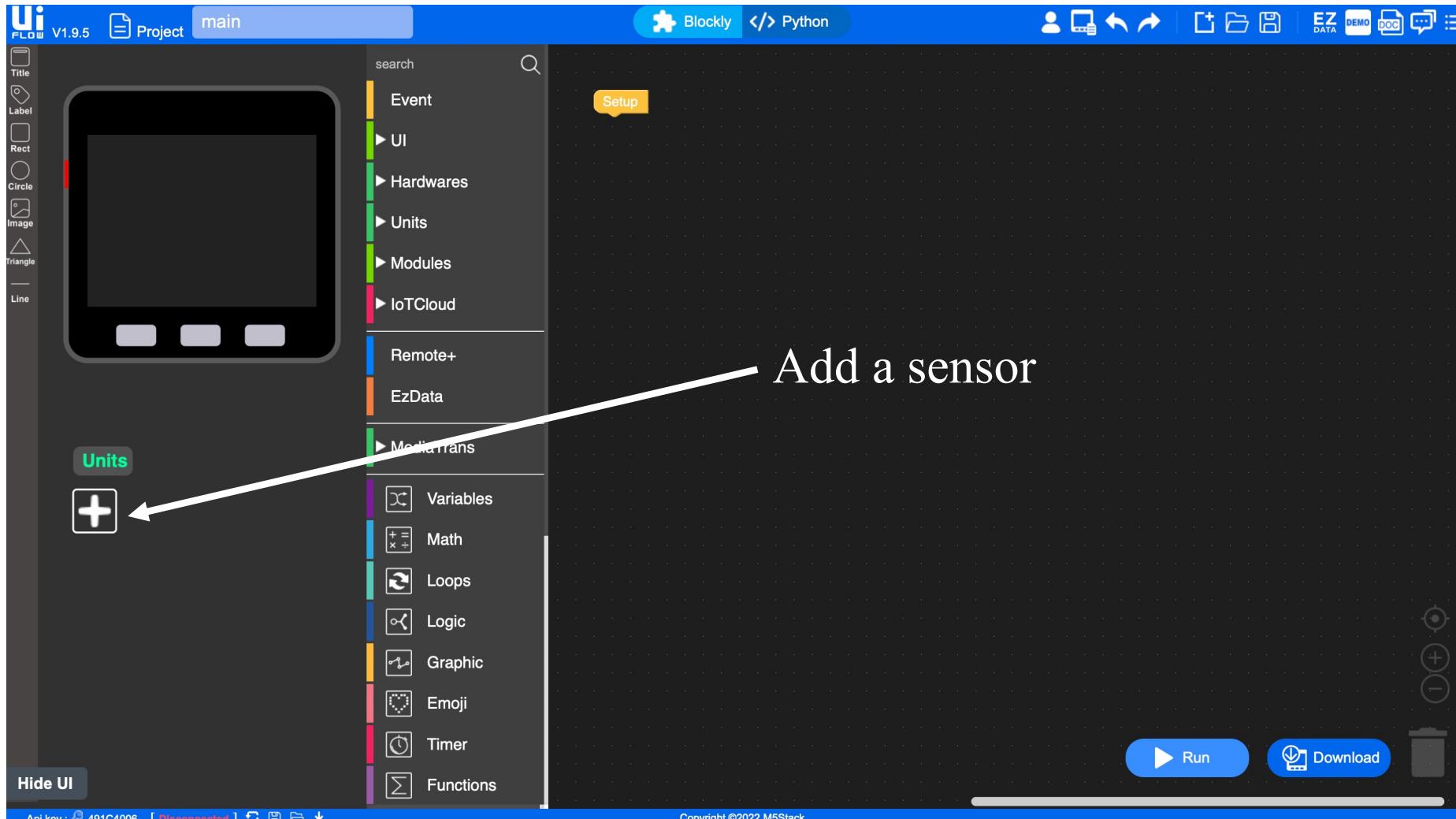


# Project 1: Read from Internet

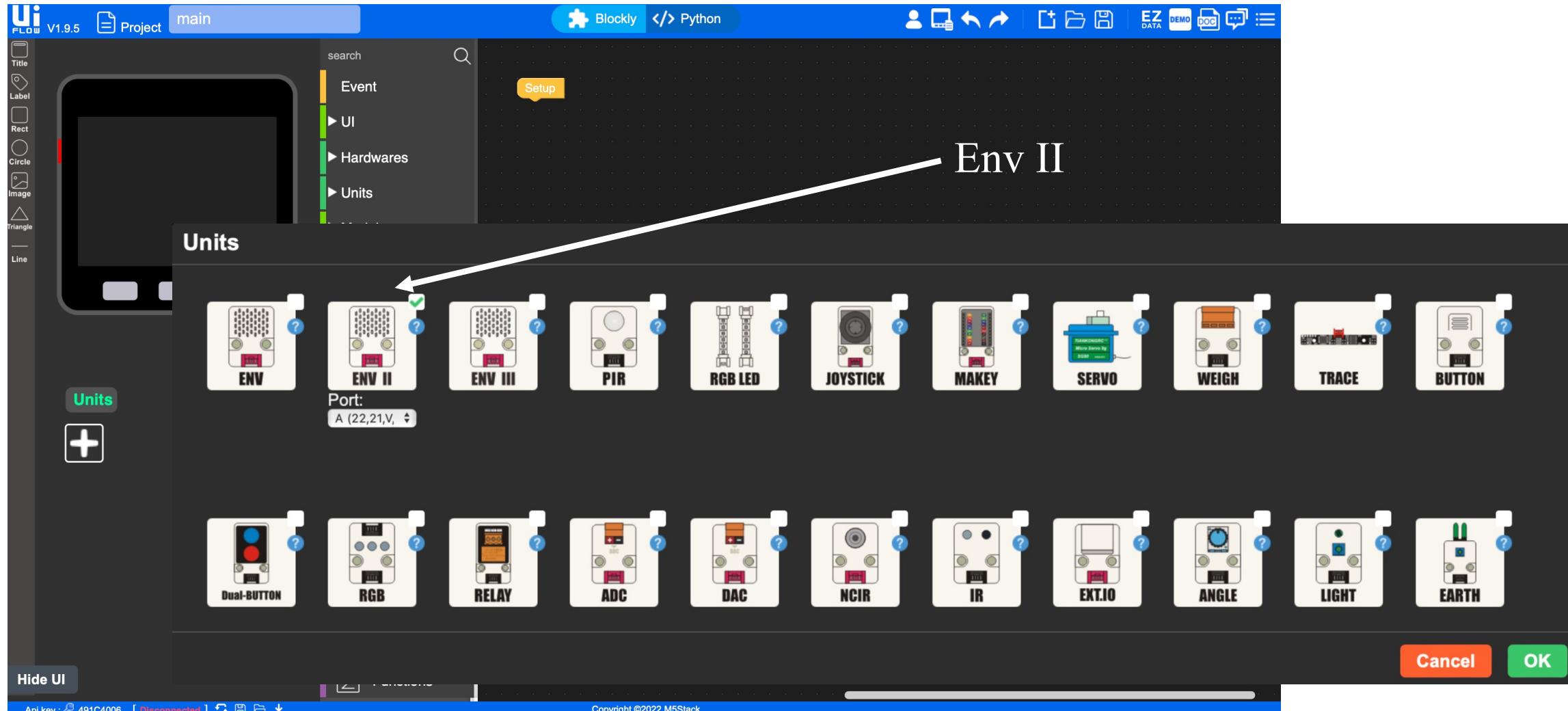
Read the weather in “Lausanne, CH” from an internet service and show it on the screen of the device

- Create a free account at [openweathermap](https://openweathermap.org). Get the key.

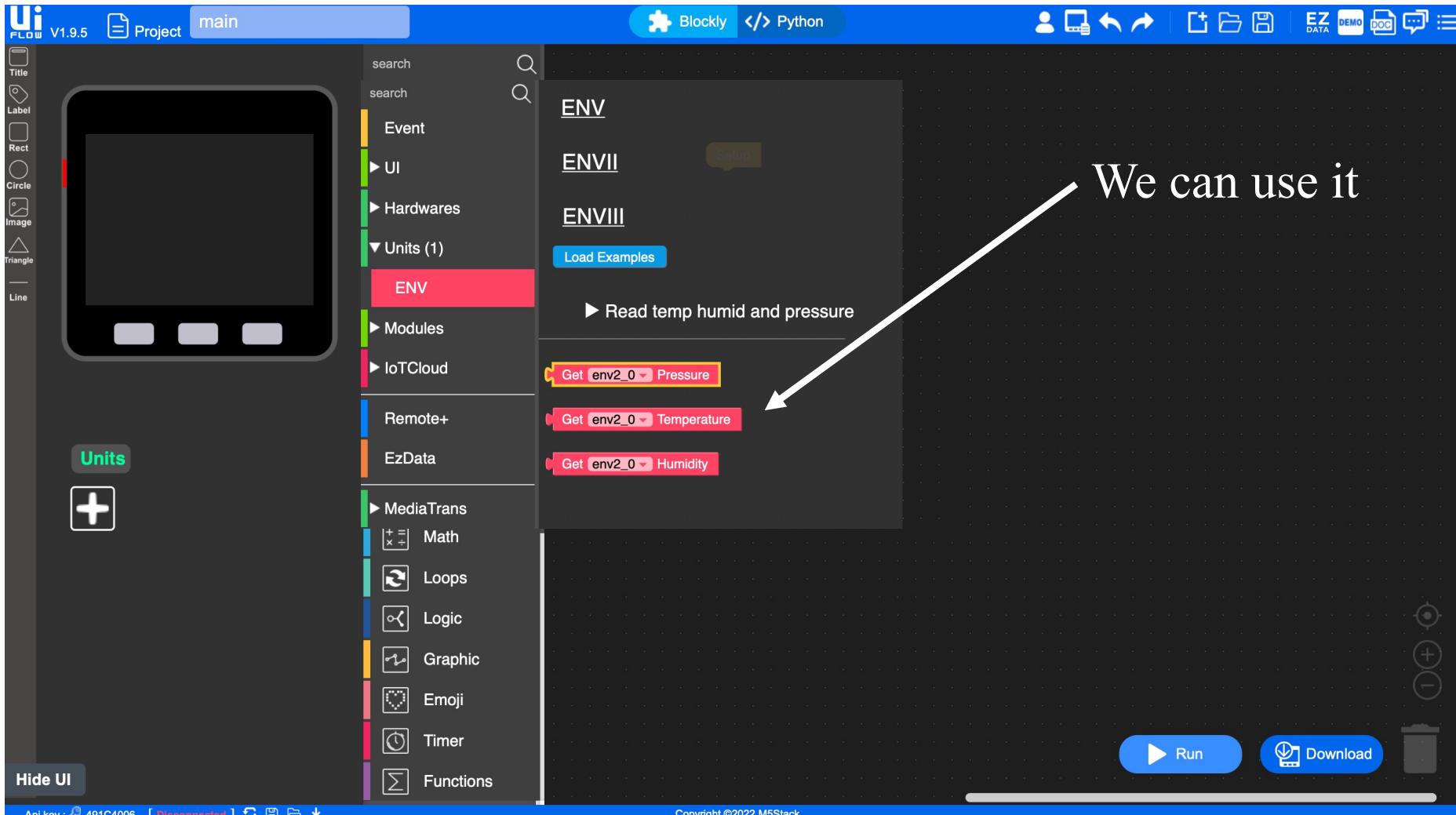
# Project 2: Read from Sensors



# Project 2: Read from Sensors



# Project 2: Read from Sensors



# Project 3: Write data from the device to the Internet

There are many ways to push data to the internet:

- Push it to Google Cloud (eg BigQuery)
- [Wolfram Data Drop](#) (universal data accumulator)
- [Blynk](#)
- Adafruit.io

# Exercise

Now we will save those data also on the Internet.  
For this we will setup an MQTT pub/sub platform.

- Create an account at Adafruit.io
- Create a dashboard and a feed.
- **Publish** your data on a **feed** of that platform.

[Profile](#)[Feeds](#)[Dashboards](#)[WipperSnapper](#)[Actions](#)[Services](#)[My Key](#)

first\_conchs\_0 > Dashboards

[+ New Dashboard](#)

## Dashboards

 **Name****Key****Created At**

Loaded in 0.19 seconds.

boards > TemperatureSensor



### Dashboard Settings

Edit Layout

Create New Block

View Fullscreen

Dark Mode ?



Block Borders ?



Dashboard Privacy ?



Delete Dashboard

Learn  
IO Plus  
News

## YOUR ADAFRUIT IO KEY

Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.



**Username**

**Active Key**

**REGENERATE KEY**

[Hide Code Samples](#)

**Arduino**



# Project 4: Control the device from the Internet

Now we do the reverse of what we did before.

- Our device **subscribes** to a feed. The feed get the input from a dashboard at adafruit.io which contains a togglebutton.
- On the device the togglebutton controls something, eg buzzing, light, prints a text, etc.
- [See video](#)

# On Adafruit.io

# The Dashboard

first\_conchs\_0 > Dashboards

+ New Dashboard

Dashboards

Name

Key

controlM5Stack

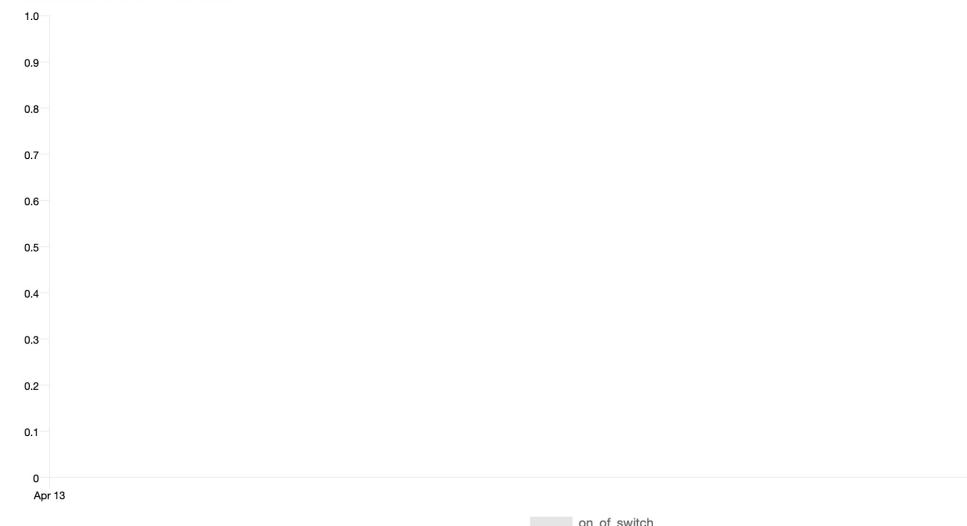
controlm5stack

first\_conchs\_0 > Dashboards > controlM5Stack



# The Feed

first\_conchs\_0 > Feeds > on\_of\_switch



+ Add Data

Download All Data

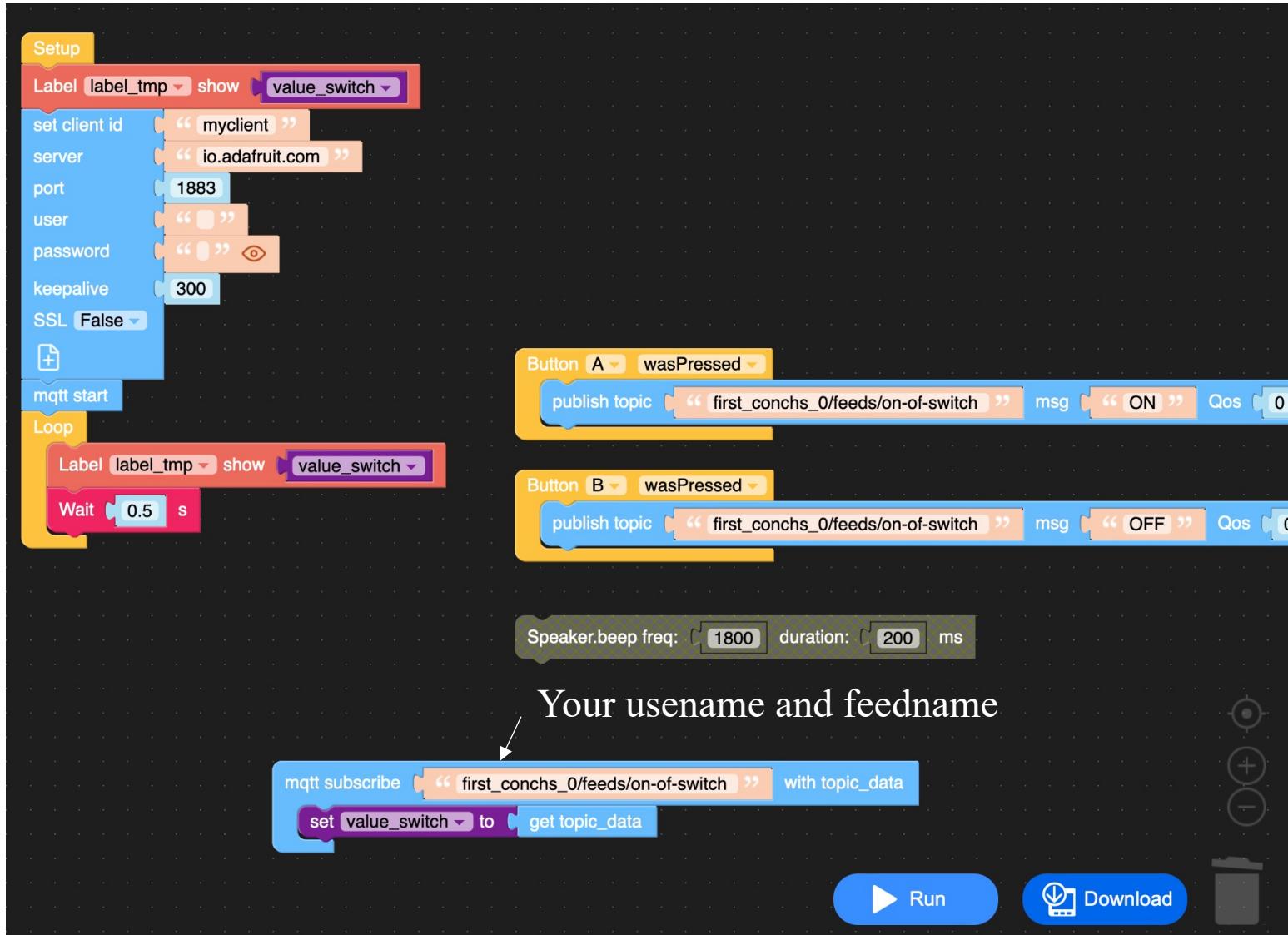
Filter

< Prev

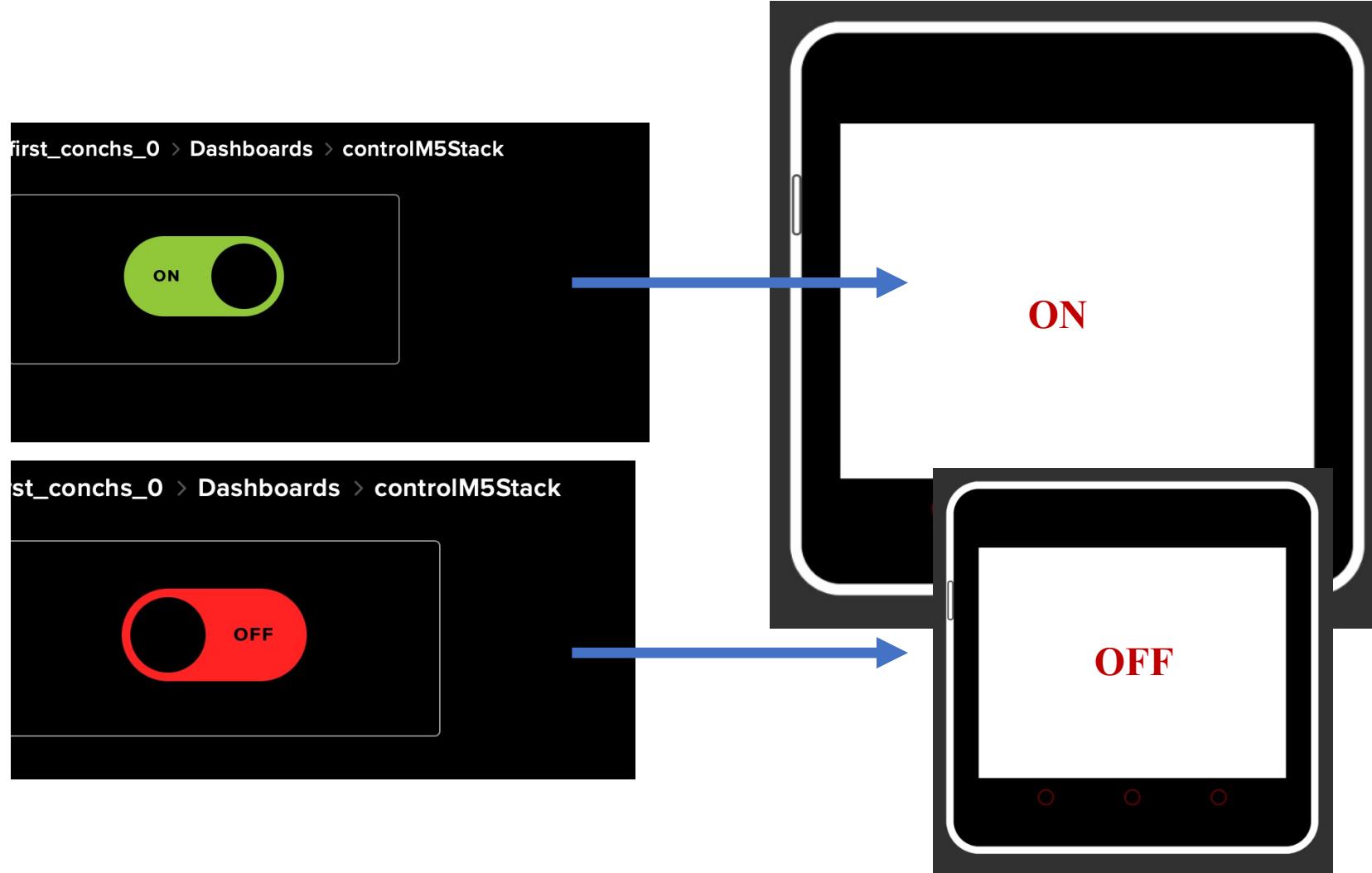
First

page 1 of 1

# On your device



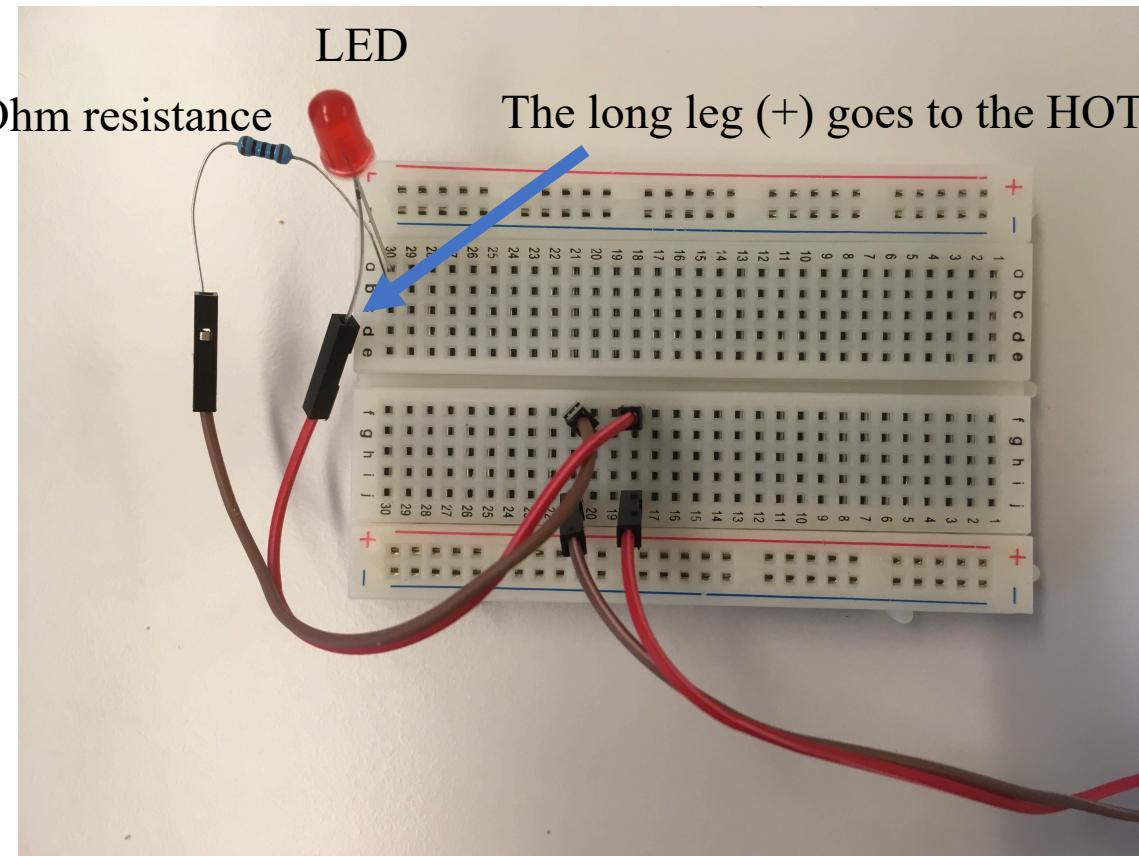
# If everything worked ok...



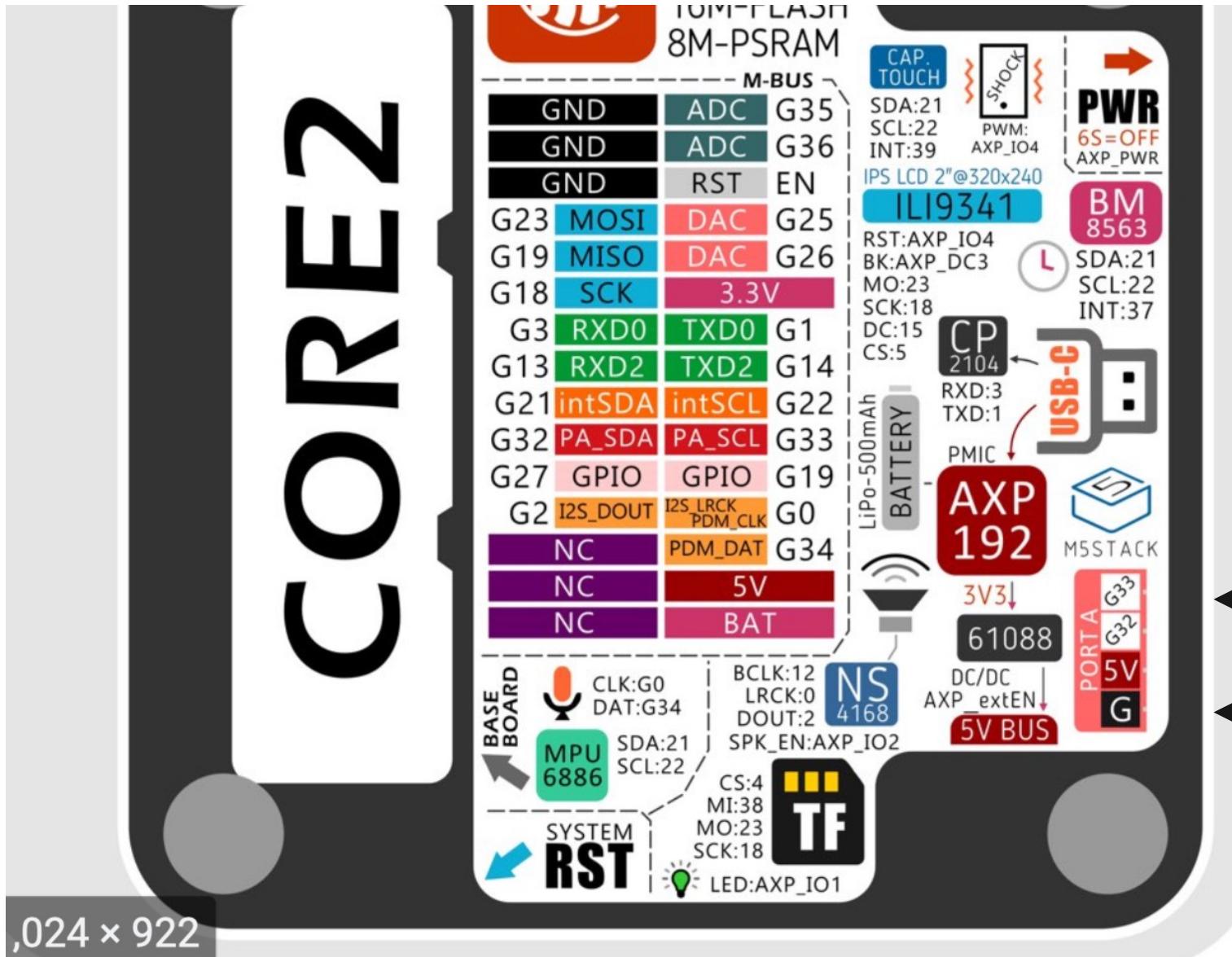
For those that like to tinker things...let's control  
a LED

# GPIO – Controlling the physical world

- We will control an LED. Turn it ON and OFF.

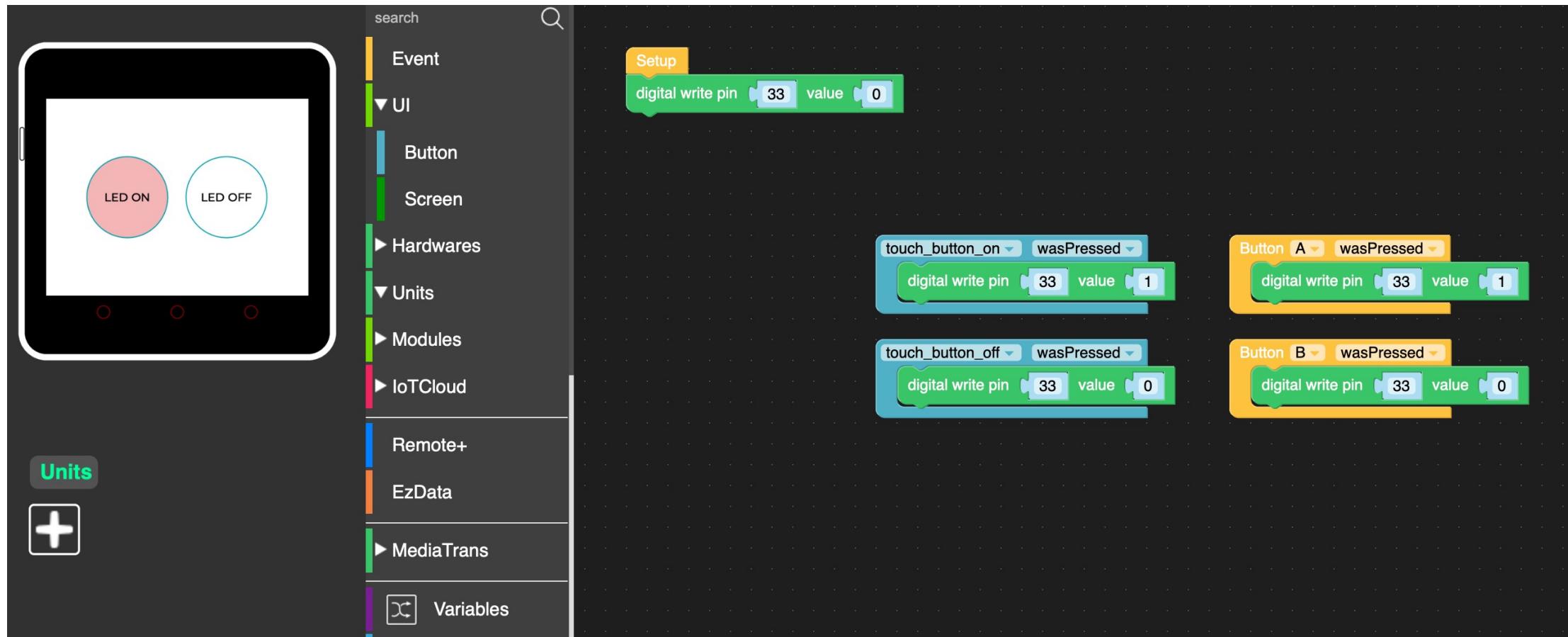


# (just for reference)



# GPIO – Controlling the physical world

- The program is super-simple. When button is pressed turn the pin 33 to ON/OFF respectively (i.e., send 5V or not)



# Tutorials on YouTube

# Selected Lessons on YouTube

- Lesson 6: [A plant monitor](#) (soil moisture unit)
- Lesson 7: [Using the temperature/humidity sensor](#)
- Lesson 8: [A digital Watch](#)
- Lesson 9: [Using the Accelerometer](#)
- Lesson 10: [Using the Potentiometer](#)
- [Weather Station with urequests and json](#)
- Lesson 12: [Remote Control a Logo RC device](#)
- Lesson 14: [Intruder Alarm with Uiflow and IFTTT](#)
- Lesson 16: [A music player](#)
- Lesson 19: [Building a game](#)

# Terminology

# Terminology

- IoT = Internet of Things
- ESP32
- Arduino
- Raspberry Pi
- MQTT (Message Queuing Telemetry Transport)
- pub-sub