JFWM2025 @ ANNECY, FRANCE 16 – 24 SEPT
JAPANESE FRENCH WEEK ON MECHATRONICS
メカトロニクスの日本フランス週間

Lecture and tutorial:
"IoT modules and its tutorial by using ~~SBC~~ MCU"

Takayuki Fujita
Professor, Deputy Director

AMERI
University of Hyogo
Himeji, JAPAN

Contents

- Introduce myself
- Overview of IoT/MEMS
- Instruction for IoT tutorial

University of Hyogo

Graduate School of Engineering

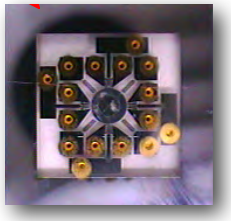
Sapporo
Aomori
Akita
Morioka
Sendai
Niigata
Fukushima
Nagano
Utsunomiya
Tokyo
Nagoya
Chiba
Shizuoka
Matsue
Tottori
Gwangju
Busan
Osaka
Hiroshima
Tsu
Himeji Kobe
Fukuoka
Matsuyama
Saga
Oita
Nagasaki
Kumamoto
Kagoshima
Miyazaki

Himeji castle

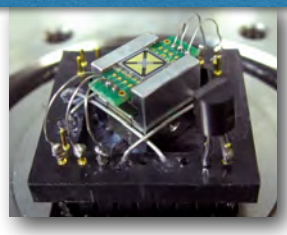
Kobe Beef

My research history

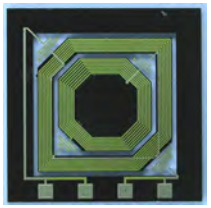
- MEMS physical sensors
  - Gyroscope
  - Accelerometers
- MEMS actuators
  - Mirror device
  - PID control system
- Environmental sensor
  - Multi sensor module for health monitoring



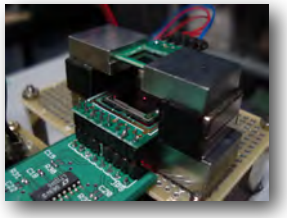
2D gyroscope



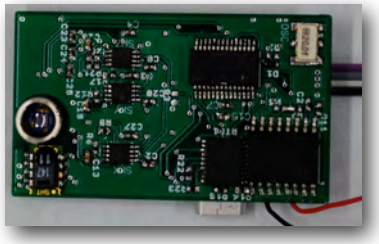
Vibratory beam accelerometer



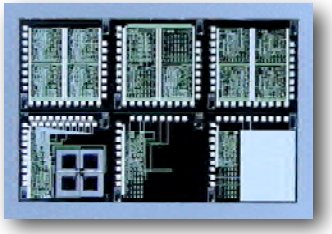
Electromagnetic MEMS mirror



Mirror control system



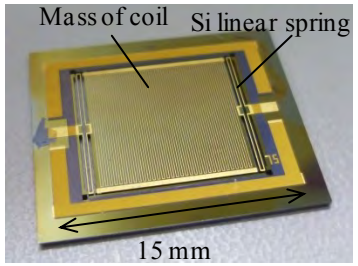
Multi-sensor module

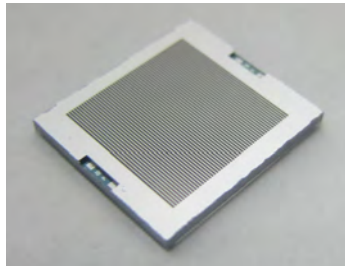


Environmental sensor

Recently, working on

- **Vibration MEMS energy harvester**
- **IoT systems for Medical applications**





Vibration energy harvester

Clean room facilities for MEMS

CleanRoom for MEMS

- Operated & managed by
  - 3 faculties
  - 1 technician
  - ~20 students

- **Full MEMS fabrication process on site**
  - Photomask fabrication, photolithography
  - Deep-RIE, dry etching, wet etching
  - Metallization, Magnetic/PZT film sputtering
  - Electroplating, Polymer-flexible electronics

AMERI, U-Hyogo

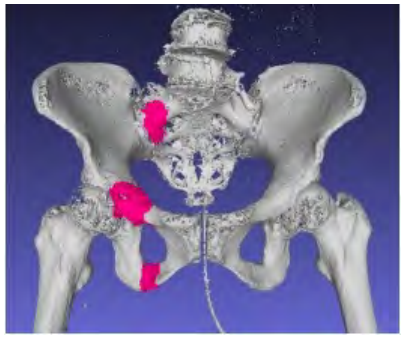
Move to New institute
open Apr. 2022

Advanced **M**edical **E**ngineering
Research **I**nstitute,
University of Hyogo

In the 2nd largest prefectural hospital in Hyogo

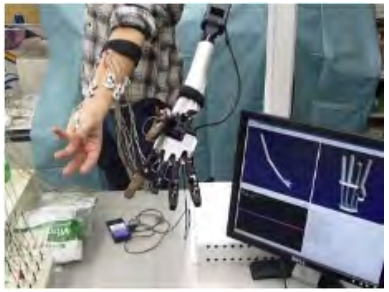
Research Topics of AMERI

Healthcare and rehabilitation Eng.

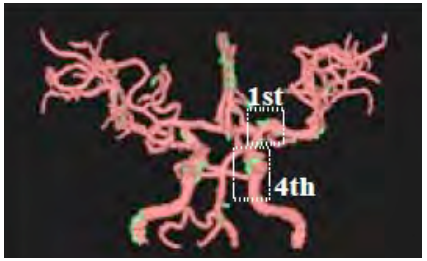


AI-based pelvic fracture location determination

Health observation using adhesive plaster-type MEMS sensors



Finger joint estimation robotic prosthetic hand



AI-based aneurysm visualization

AI related topic from AMERI will be presented on Monday's Keynote talk

Contents

- Introduce myself
- **Overview of IoT/MEMS**
- Instruction for IoT tutorial

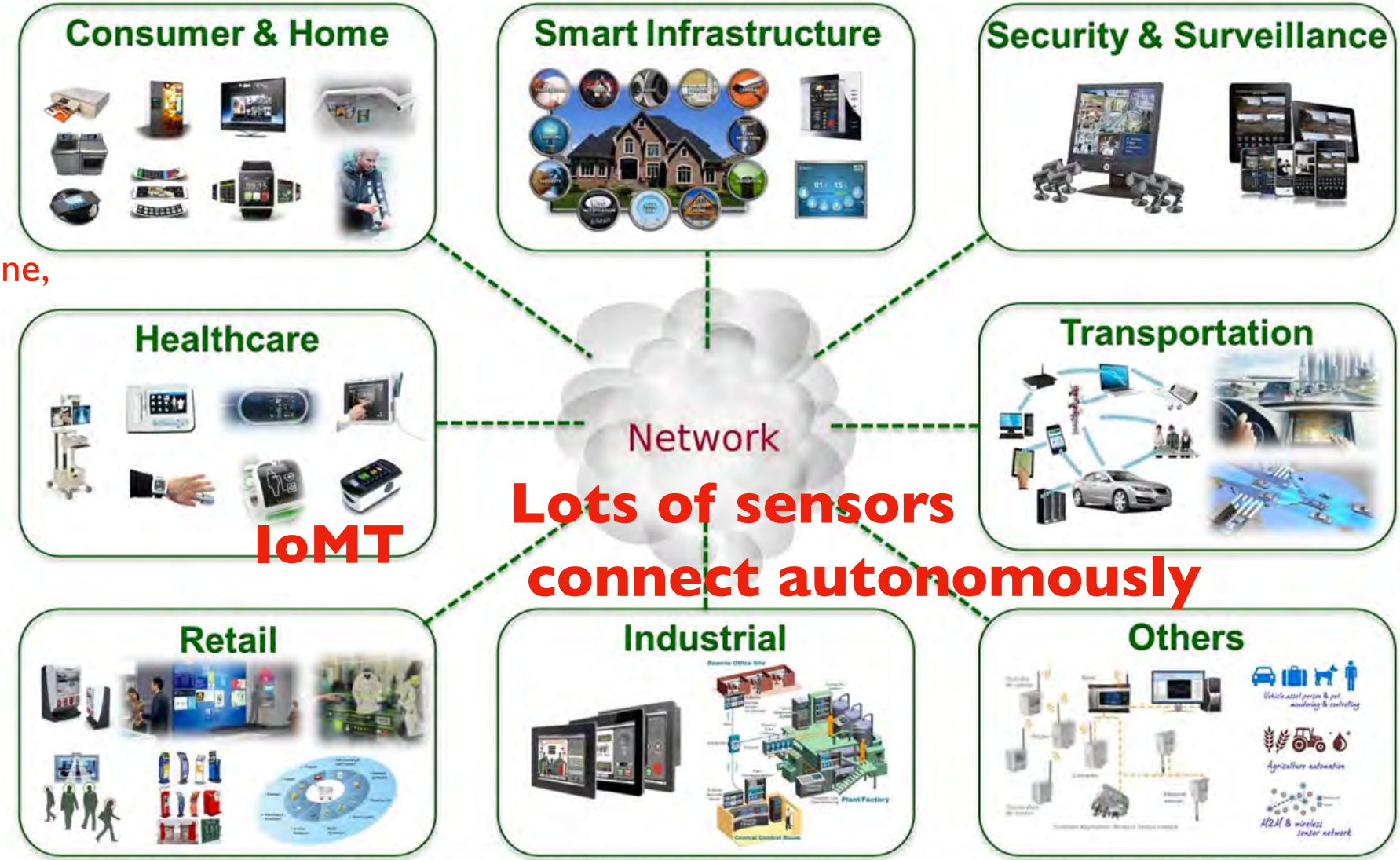
IoT (Internet of Things)

Human uses the Internet

Smartphone, Tablet, PC

IoT

Internet connection for everything not limited to Human operated device



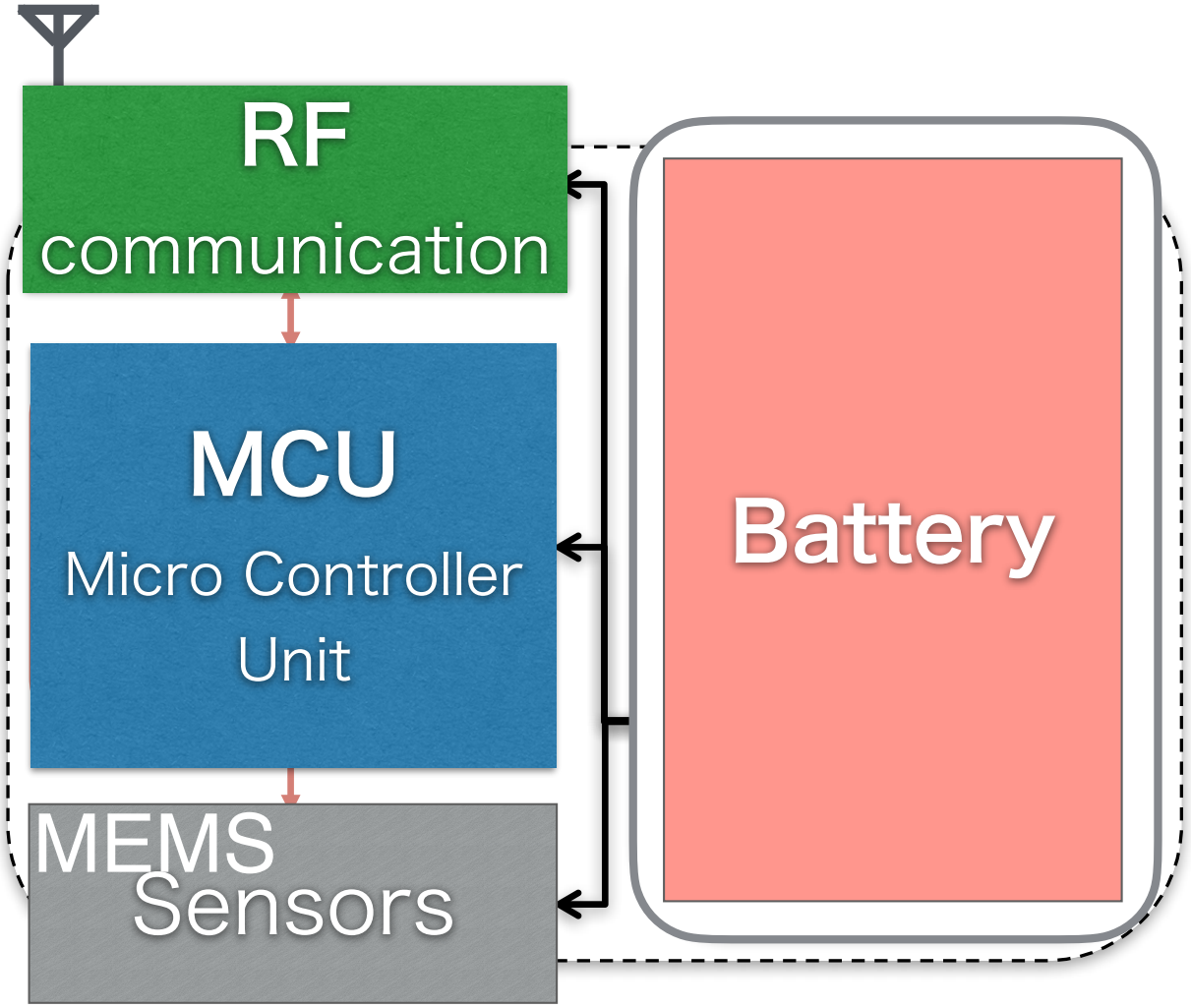
IoMT

Lots of sensors connect autonomously

Vivante and the Vivante logo are trademarks of Vivante Corporation. All other product, image or service names in this presentation are the property of their respective owners. © 2013 Vivante Corporation

IoT node architecture

Typical architecture of IoT node



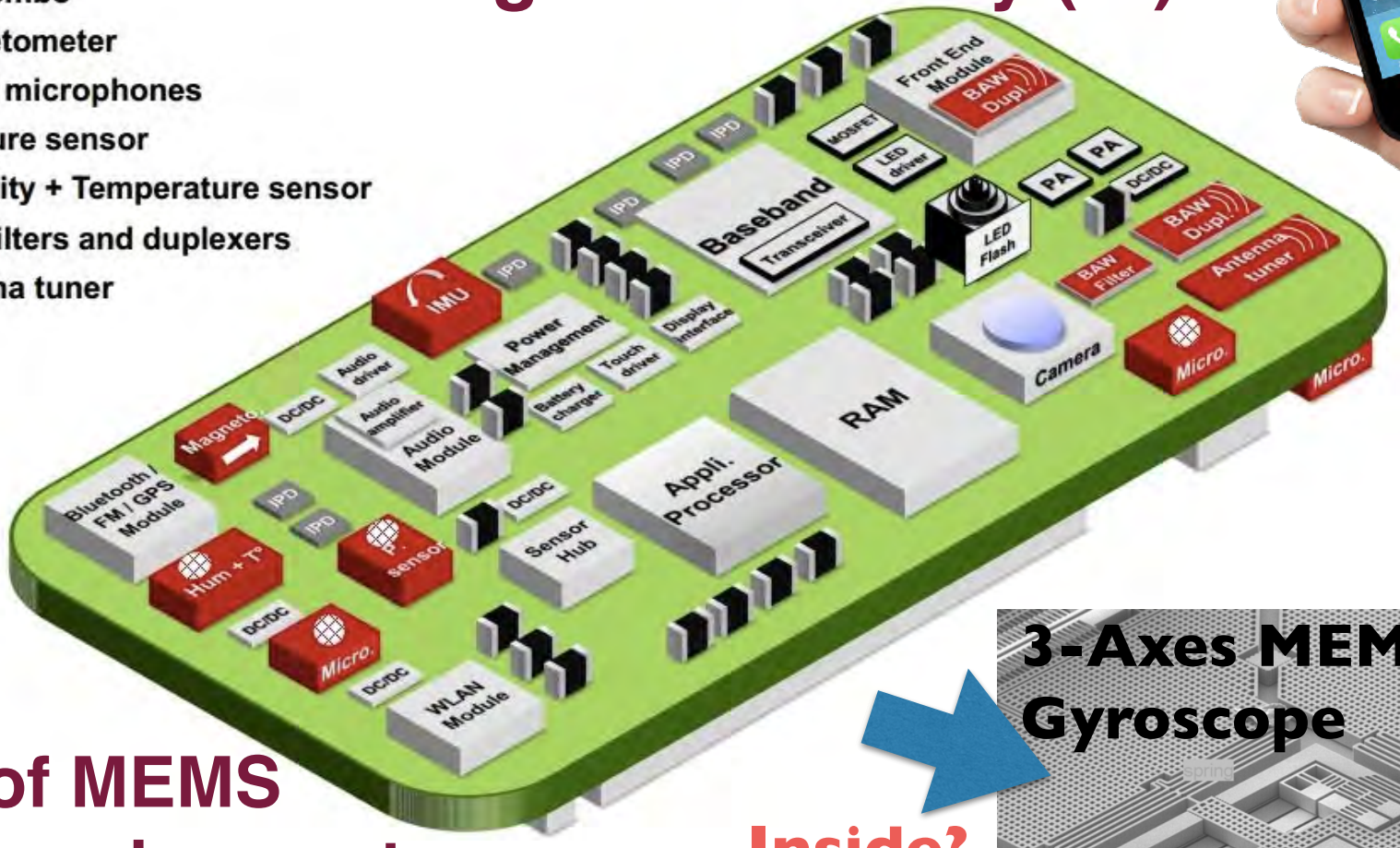
Architecture of a sensor node

IoT has developed dramatically thanks to **MEMS**

MEMS - Micro Electromechanical Systems

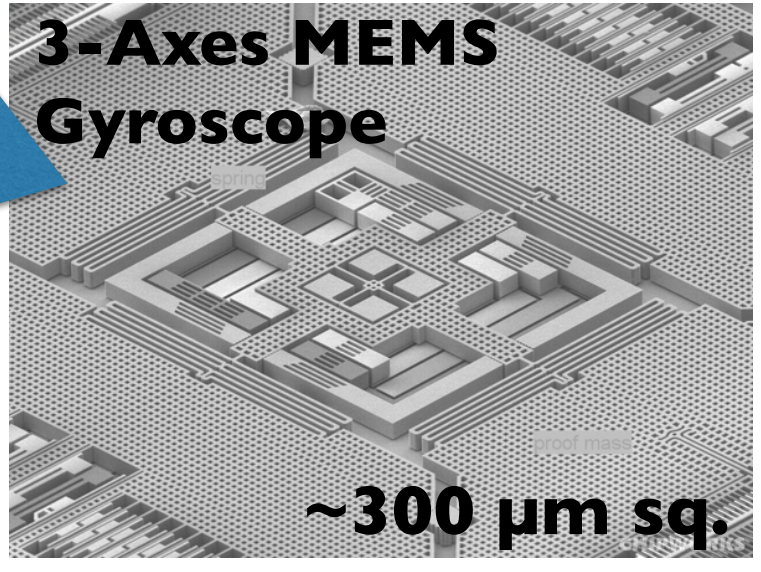
MEMS = Fusion of ultra small machine and Integrated Circuity (IC)

- IMU combo
- Magnetometer
- MEMS microphones
- Pressure sensor
- Humidity + Temperature sensor
- BAW filters and duplexers
- Antenna tuner



Full of MEMS sensors in smart phone

Inside?



at least <u>5 MEMS sensors</u>

History of MEMS application

Automotive

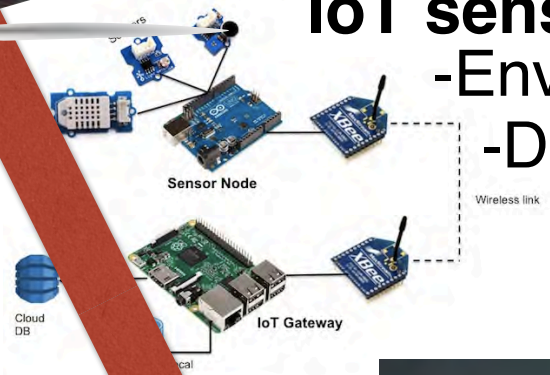
35 yrs ago

- **Airbag** control
- Electronic stability control
- Rollover detection
- Navigation (GPS support)

Mobilephone / Entertainment

- Input device
- Navigation

IoT sensor node



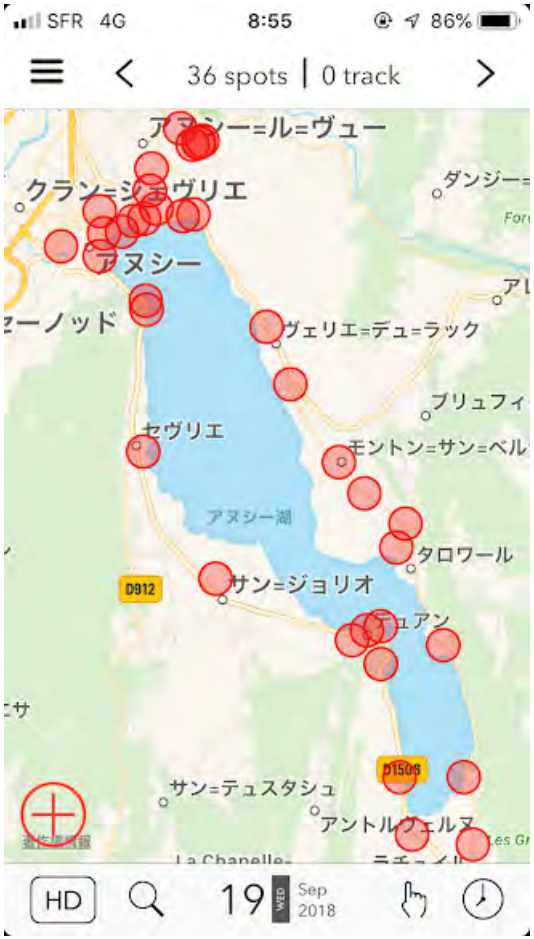
- Environmental sensing
- Drones

Health

- Medical devices

Performance & Number ↑
Size & Cost ↓

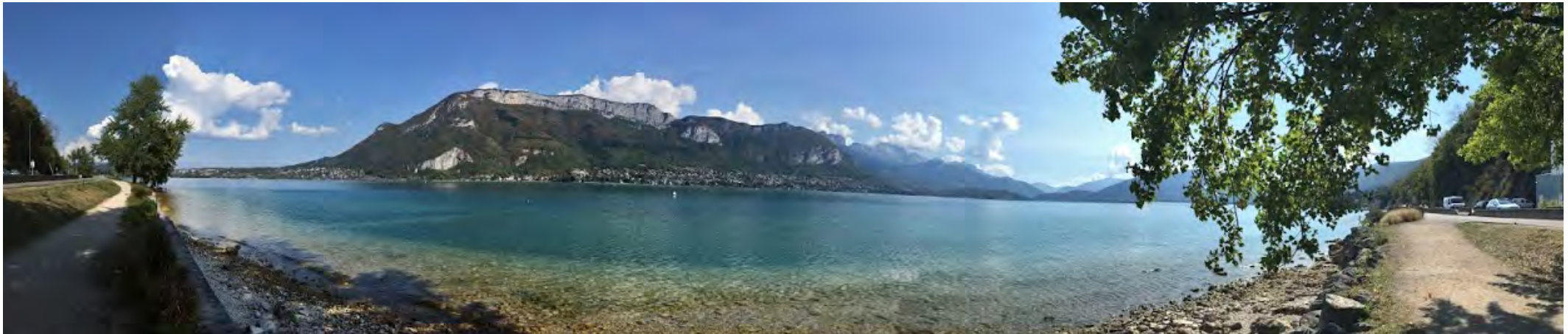
MEMS application > Navigation



Tracking: Cycling around the Lac d'Annecy

Position tracking by combined sensor info.

- **GPS** ⛔
- Celler
- WiFi
- Geomagnetic
- **Gyroscope**
  - **Angular velocity ∫ Angle**
- **Accelerometer**
  - **Acceleration ∫ Velocity ∫ Position**



AirTag (not GPS but BLE/UWB)

Thanks to the AirTag(IoT)
The package finally arrived.

15th PM

KIX Osaka Airport

16th AM

GVA Airport

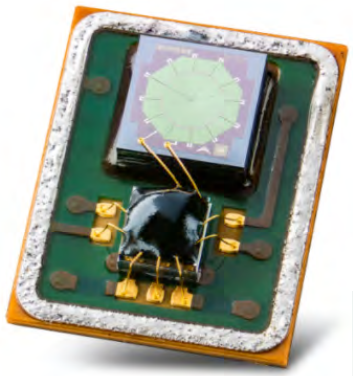
16th PM

Divonne-les-Bains

16th Evening

Annecy

MEMS device 1: Microphone



4 x 3 mm

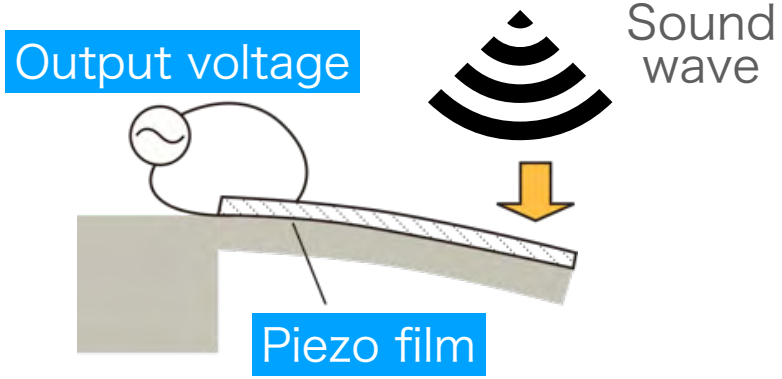
© Vesper

Rice grain size

vesper

MEMS makes compact, high-performance microphones inexpensive

Cantilever with piezoelectric material vibrate by sound



iPhone 11

© Voista Media

iPhone 11 has 4 microphones

3 4 for main mic.

Others are use for noise cancelation

MEMS device 2: Apple watch etc.

Activity tracking for health care

Similar technology

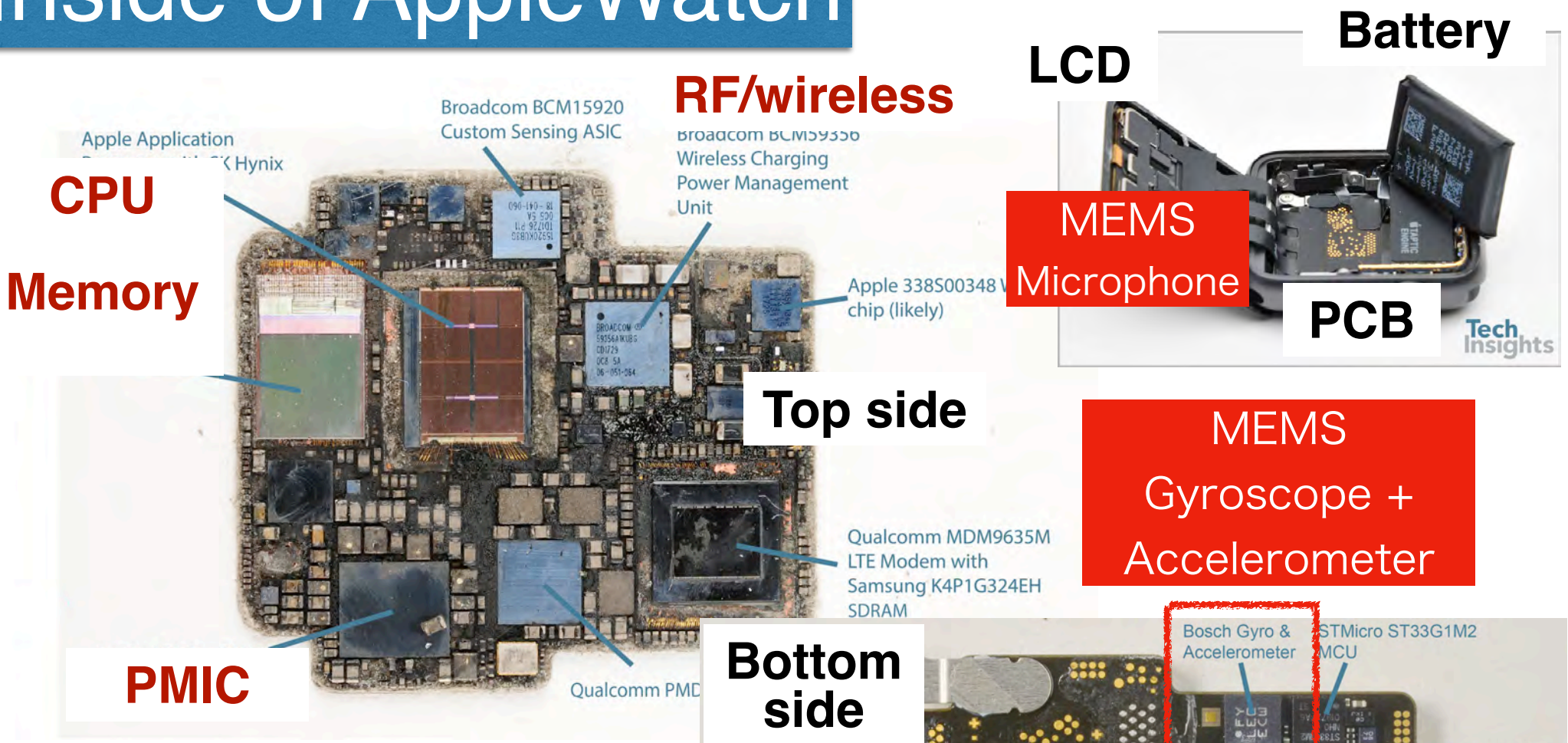
RF, MCU, battery
&
MEMS sensors
Acc. Gyro,

Optical sensor
For SPO_2, heart rate

Various devices of "healthy gadgets"

Inside of AppleWatch

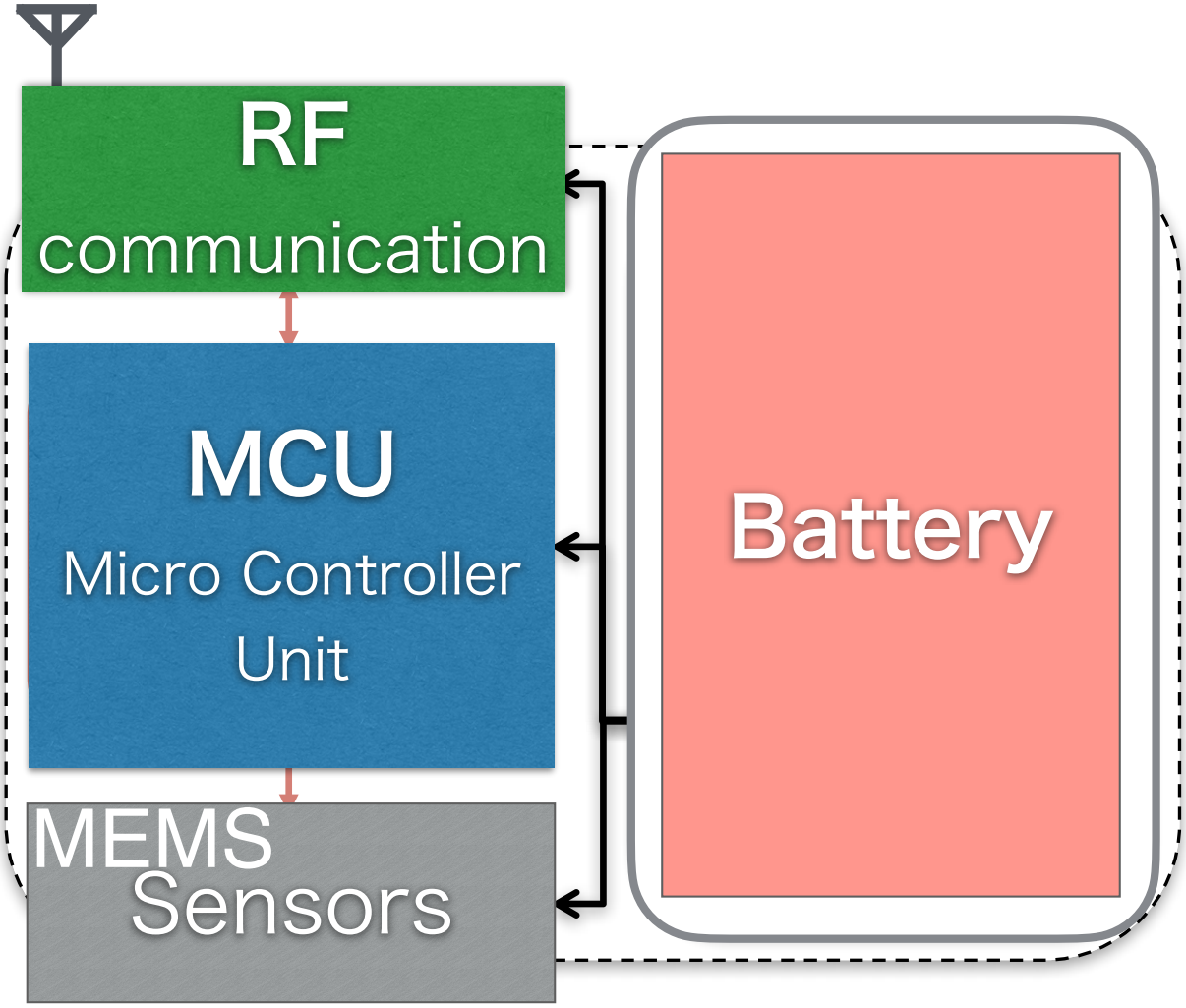
(C) TechInsights



AppleWatch 3rd gen.

IoT node architecture

Typical architecture of IoT node



Architecture of a sensor node

>NEXT

Explains the internal details of IoT applications in the market

IoT on market : IoT Body scale

¥20,000
(USD 150)
worth the price

Withings

Made in France

- **WiFi connection**
- **No switch, Easy to use**
- **Automatically data upload**

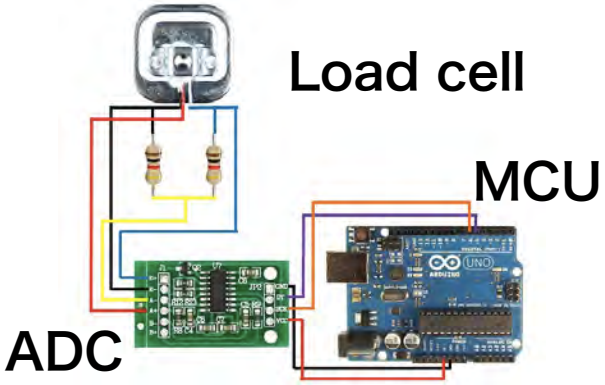
Replaced new one

Data still ongoing

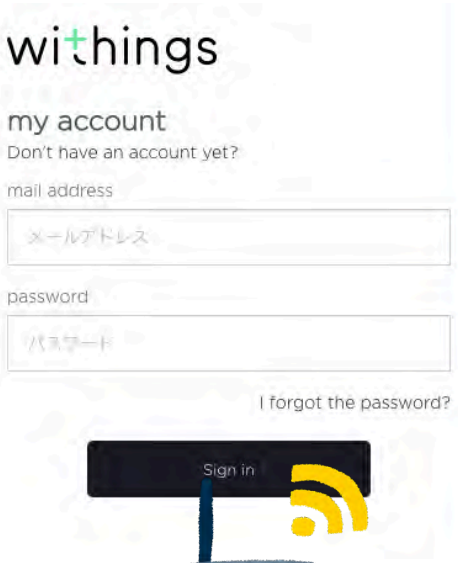
Statistics of my body weight for 14 yrs. are stored on Cloud

Inside of Withings IoT scale

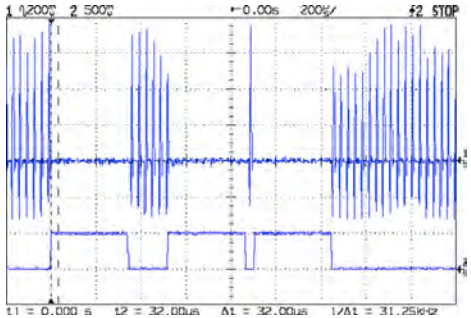
Body weight measured
by load-cell (strain sensor)



Withings server



Body fat % by impedance measurement

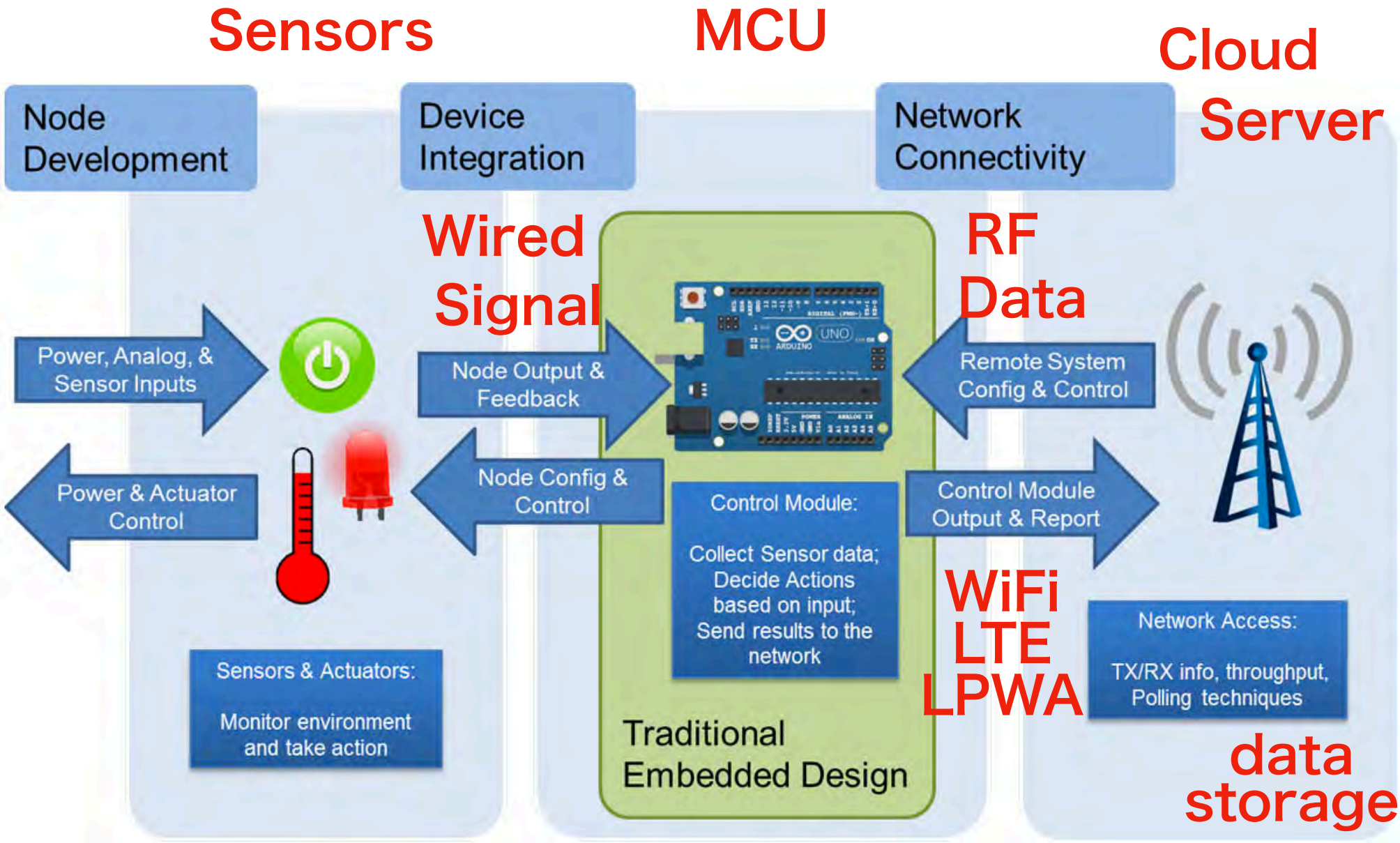


WiFi

Generate graph on cloud

Connect via Wifi (pre-configured SSID / password)
> upload data to Withings cloud server

Typical configuration IoT



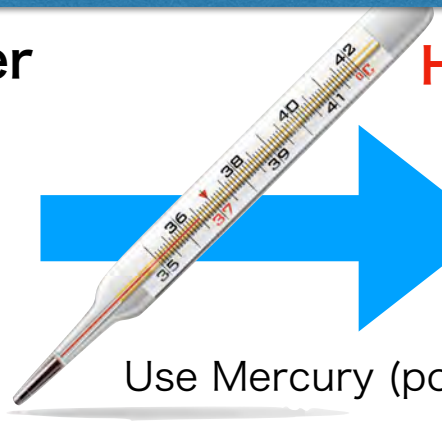
>NEXT **Introduction to the history of measurement, using temperature as an example**

Temperature measurement (primitive method)

ex. Glass tube thermometer

Have you ever seen it?

Physical change
> Temperature



Use Mercury (poison)

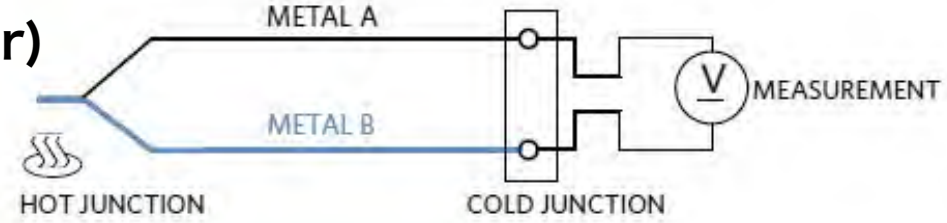
Physical change
> thermal expansion of alcohol

No electricity

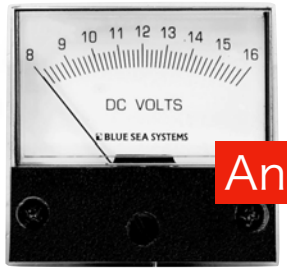
Visual check only

---

ex. Thermocouple
(temperature sensor)



Analog meter



Analog value

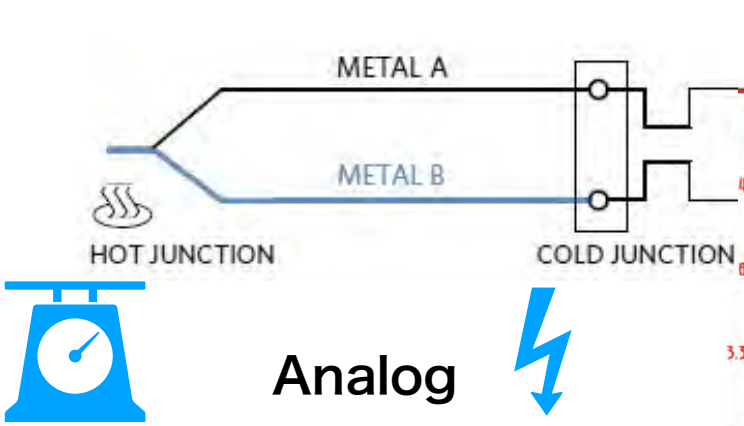
Physical change
> Temperature

Electrical change
> Electromotive force

Physical change
> meter movement

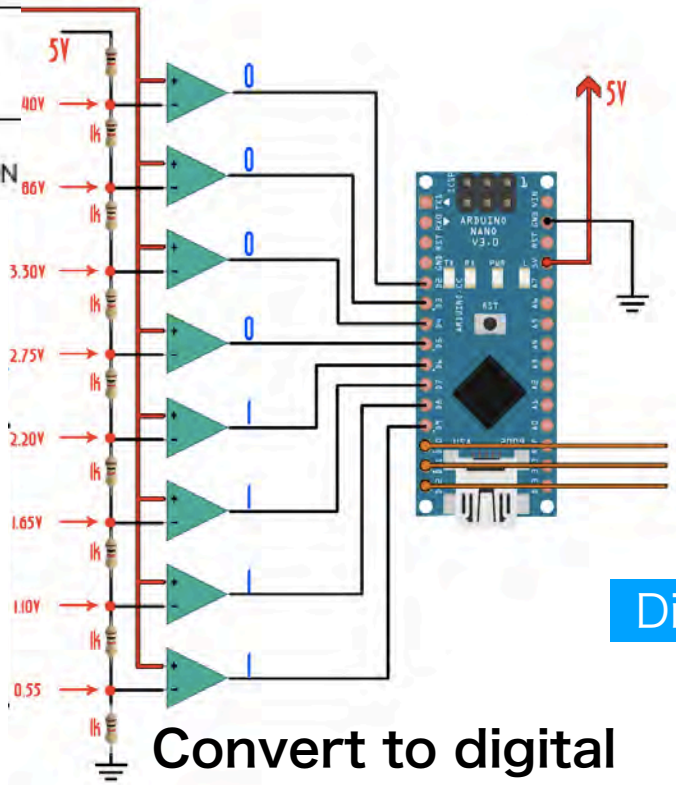
Analog values are difficult to connect MCU or SBC

Temperature measurement (w/ ADC)



Analog

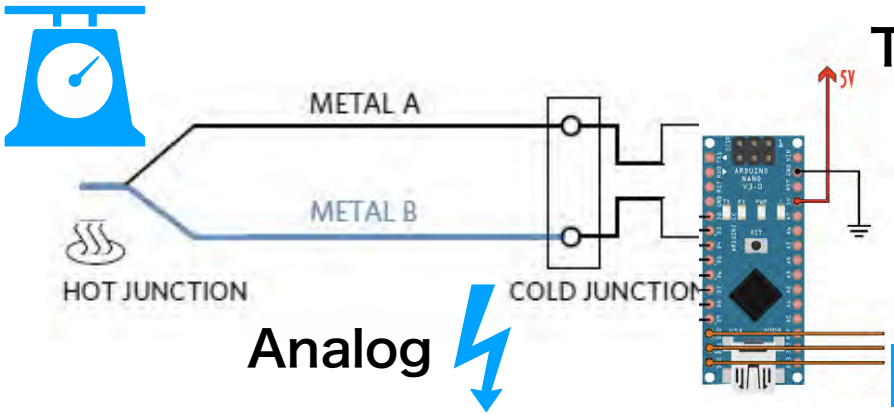
series of comparators



Resolution depends on the number of ladder resistors

Digital digit

Convert to digital (ADC)



Analog

Typical MCU (ESP32) has
Internal ADC
12 bits, 3.3V

Digital digit

Resolution: 0.8 mV

Recent digital output MEMS sensor

BOSCH
Invented for life

BME280
combo sensor

Use this on tutorial

Fully digital output with compensation parameters

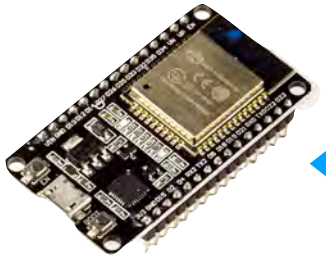
Single chip
-Temperature
-Humidity
-Pressure

Main features

Relative Humidity
Measures relative humidity with a fast response time

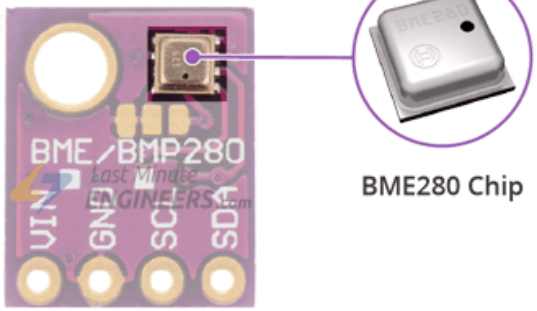
Pressure
Measures barometric pressure and altitude

Temperature
Measures ambient temperature



MCU

Digital



Sensor

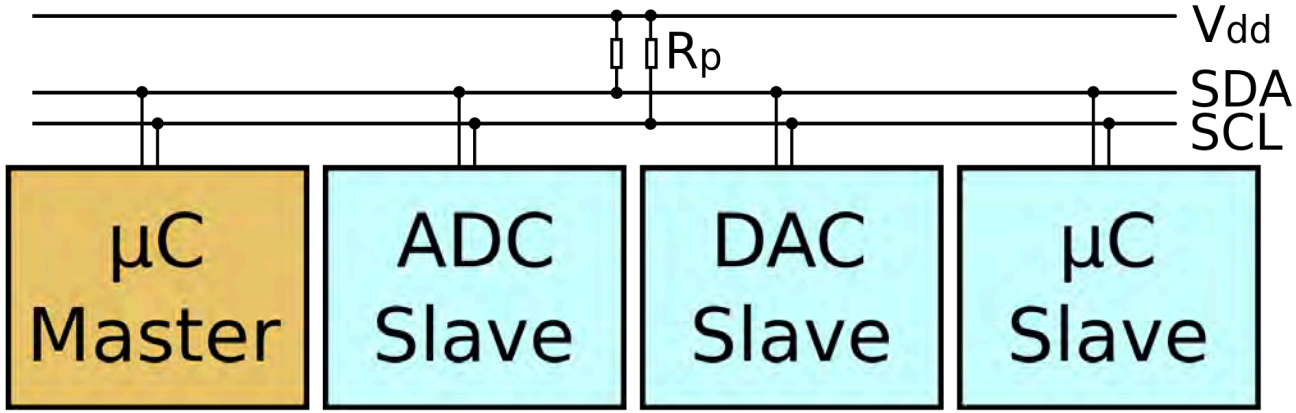
BME280 Chip

Direct connection via I2C/SPI protocol

What is interface protocol?

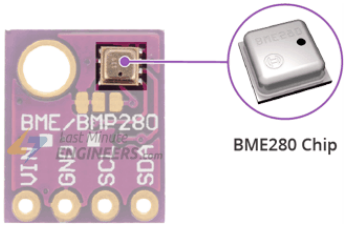
1) Inter-Integrated Circuit (I²C)

I²C (**Inter-Integrated Circuit**), pronounced *I-squared-C*, is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial computer bus. It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication. Alternatively I²C is spelled **I2C** (pronounced I-two-C) or **IIC** (pronounced I-I-C).



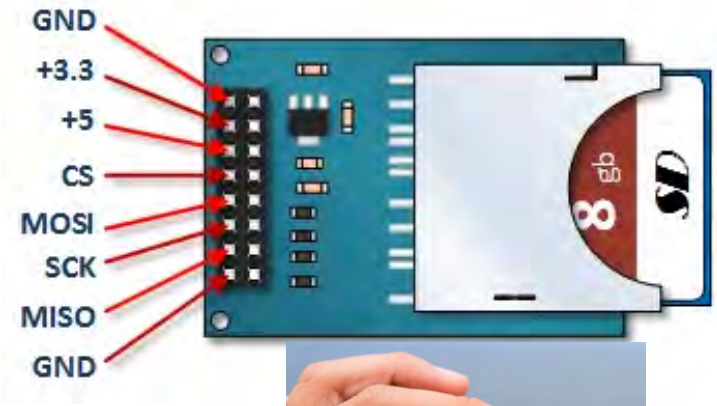
2 digital signal line + 2 power line

Selected by unique-address

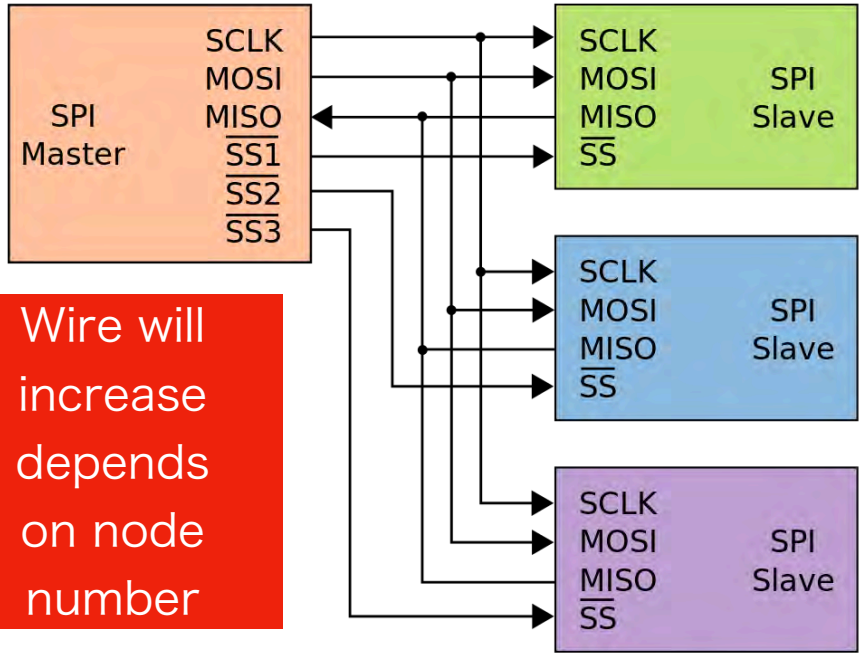


2) Serial Peripheral Interface (SPI)

The **Serial Peripheral Interface** (**SPI**) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems. Typical applications include Secure Digital cards and liquid crystal displays. SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave-devices are supported through selection with individual slave select (SS), sometimes called chip select (CS), lines.



High-speed



Wire will increase depends on node number

2 digital signal line + 2 power line

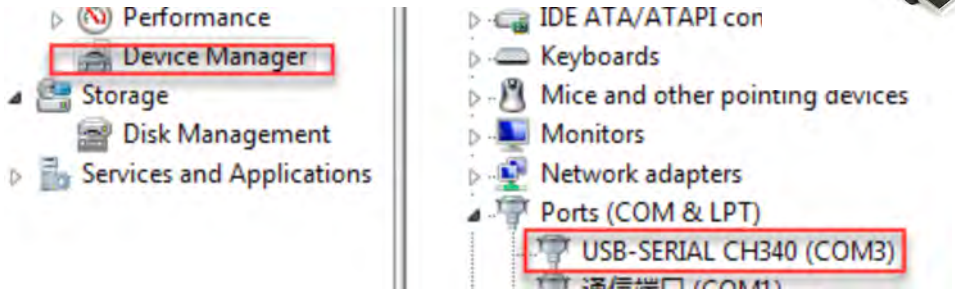
+1 CS line for chip select

3) Universal Asynchronous Receiver/Transmitter (UART)

A universal asynchronous receiver-transmitter (UART) is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable. The electric signaling levels are handled by a driver circuit external to the UART. Two common signal levels are RS-232, a 12-volt system, and RS-485, a 5-volt system.

Old PC has RS-232C Serial port

USB (Universal Serial Bus) Replaced it

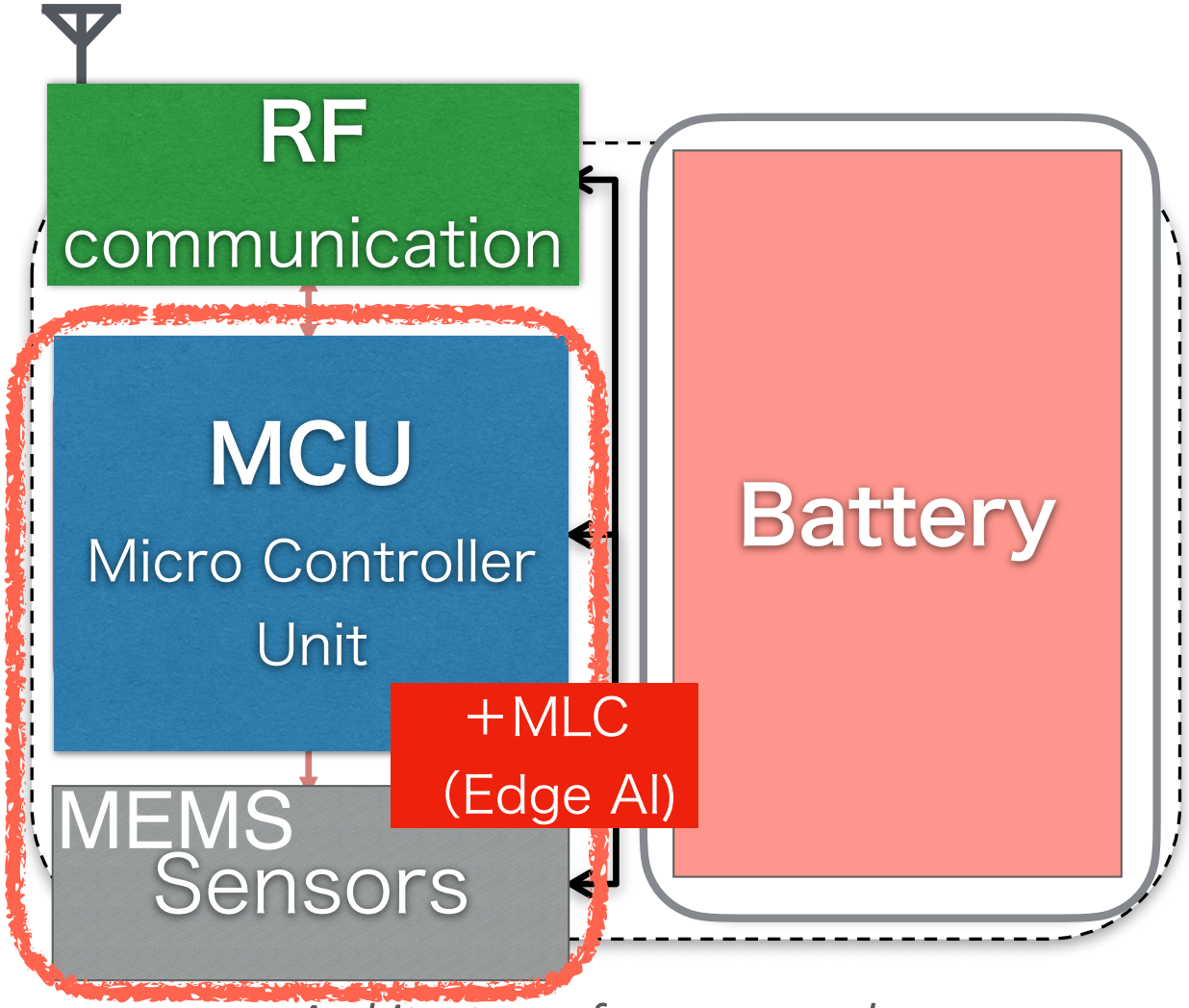


Windows recognize it as COMmunication port

Almost all MCU board includes USB-UART converter chip Ex. CH340

Cutting edge IoT node architecture

Edge AI



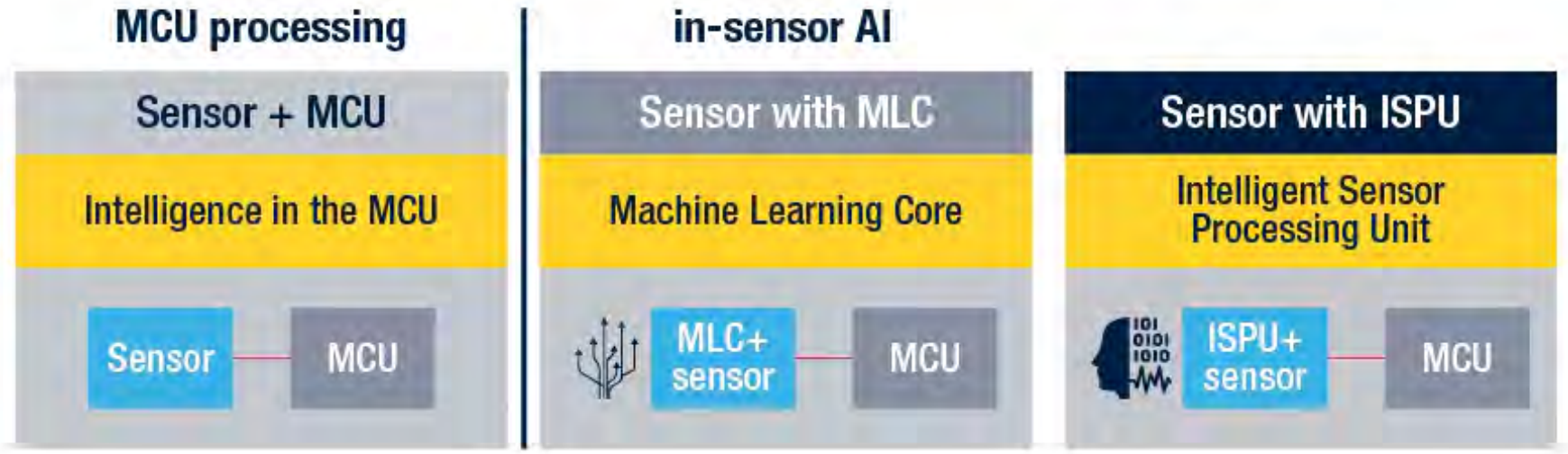
Architecture of a sensor node

Explains the internal details of IoT applications in the market

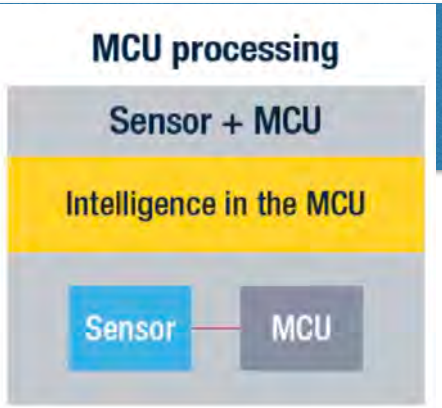
Commercial Edge AI from STMicro

A leading company of MEMS sensor & Edge AI

To ensure developers find the most effective solution in terms of computing capacity and flexibility in programming, ST offers a choice of several technologies for in-sensor processing: sensors with an embedded **machine learning core (MLC)** and sensors with an **intelligent sensor processing unit (ISPU).**



Edge AI on market



Panasonic e-bike & ST Micro

DETECTING LOW TIRE PRESSURE ON E-BIKES VIA EDGE AI

Problem: Electric-assist bicycles in Japan often suffer punctures caused by low tire pressure.

Approach (no pressure sensor needed):

- Compare rear-wheel motor RPM with front-wheel speed sensor data
- AI detects abnormal differences → indicates low pressure in either front or rear tire

Implementation:
Edge AI processes data locally and triggers a warning alarm

Generated by ChatGPT

Contents

- Introduce myself
- Overview of IoT/MEMS

- **Instruction for IoT tutorial**

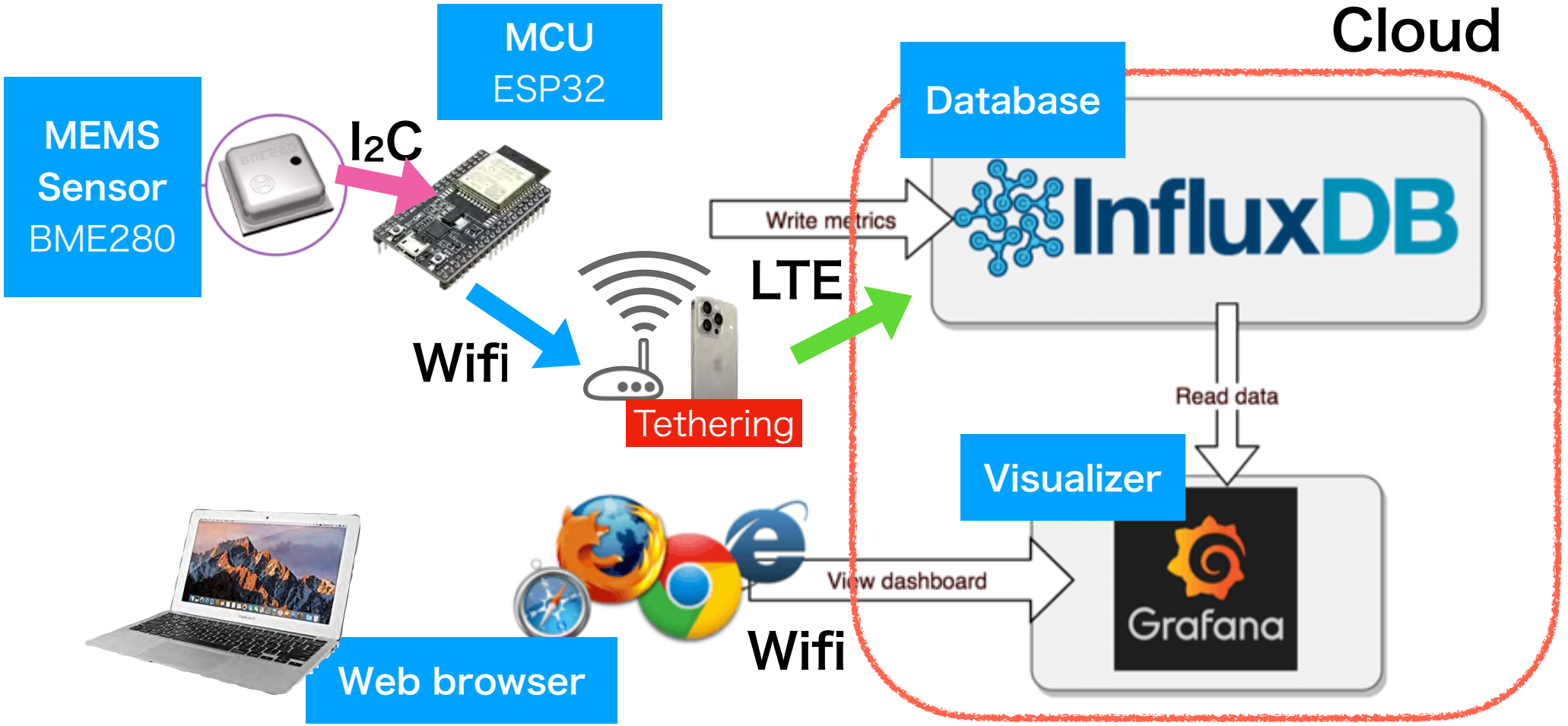
Goal of tutorial

Sensing data from MEMS environmental sensor

> gather & processing data on MCU

> transmit to InfluxDB database on cloud

> visualized by Grafana > display on web browser



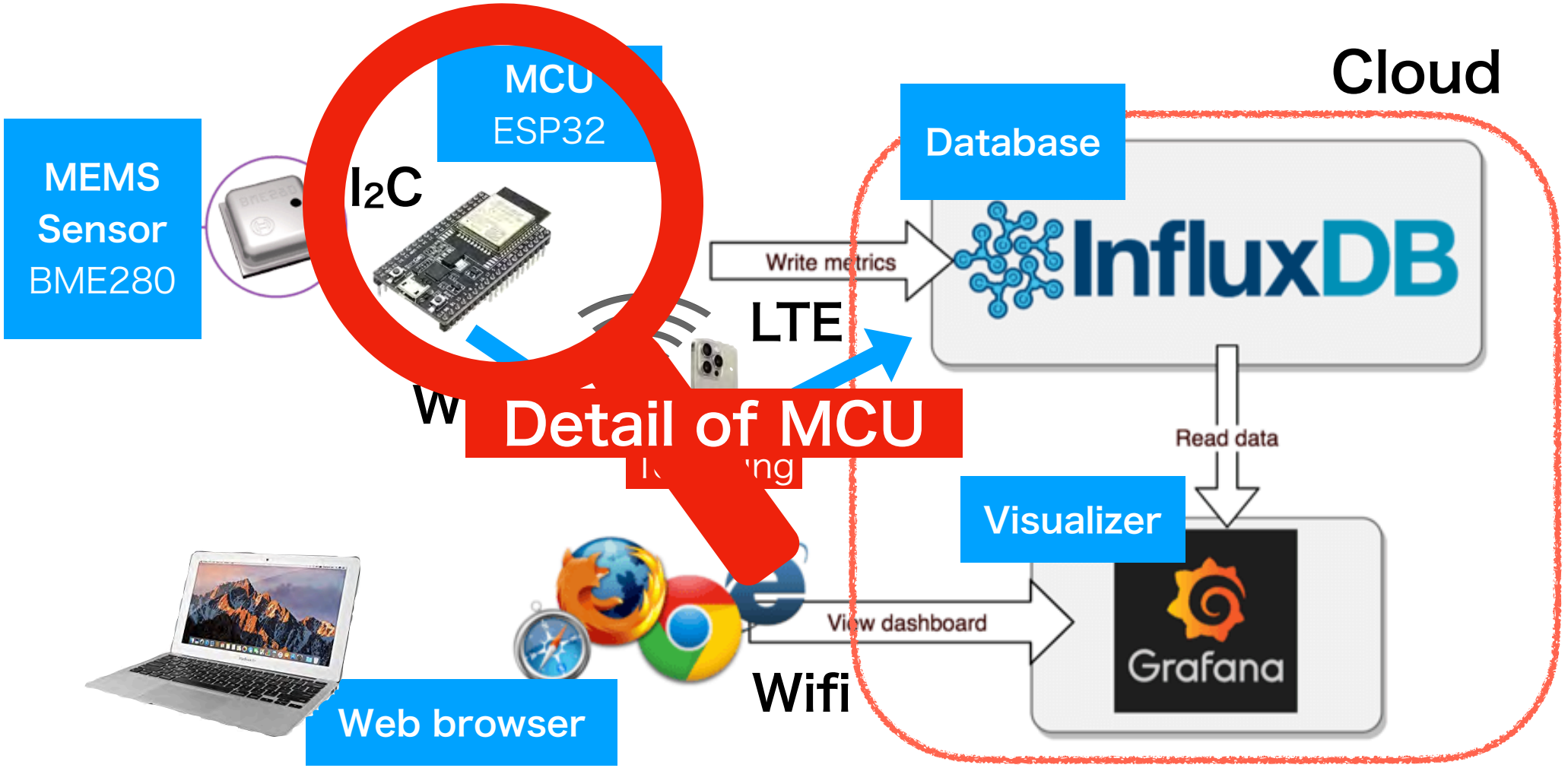
Goal of tutorial

Sensing data from MEMS environmental sensor

> gather & processing data on MCU

> transmit to InfluxDB database on cloud

> visualized by Grafana > display on web browser



Wide variety of MCU variations

ARM-based MCU is popular these days

Apple silicon also ARM based core

Raspberry Pi Pico

Wi-Fi, BLE

ARM Cortex M0+

ESP32

Use this on tutorial

Tensilica Xtensa LX6

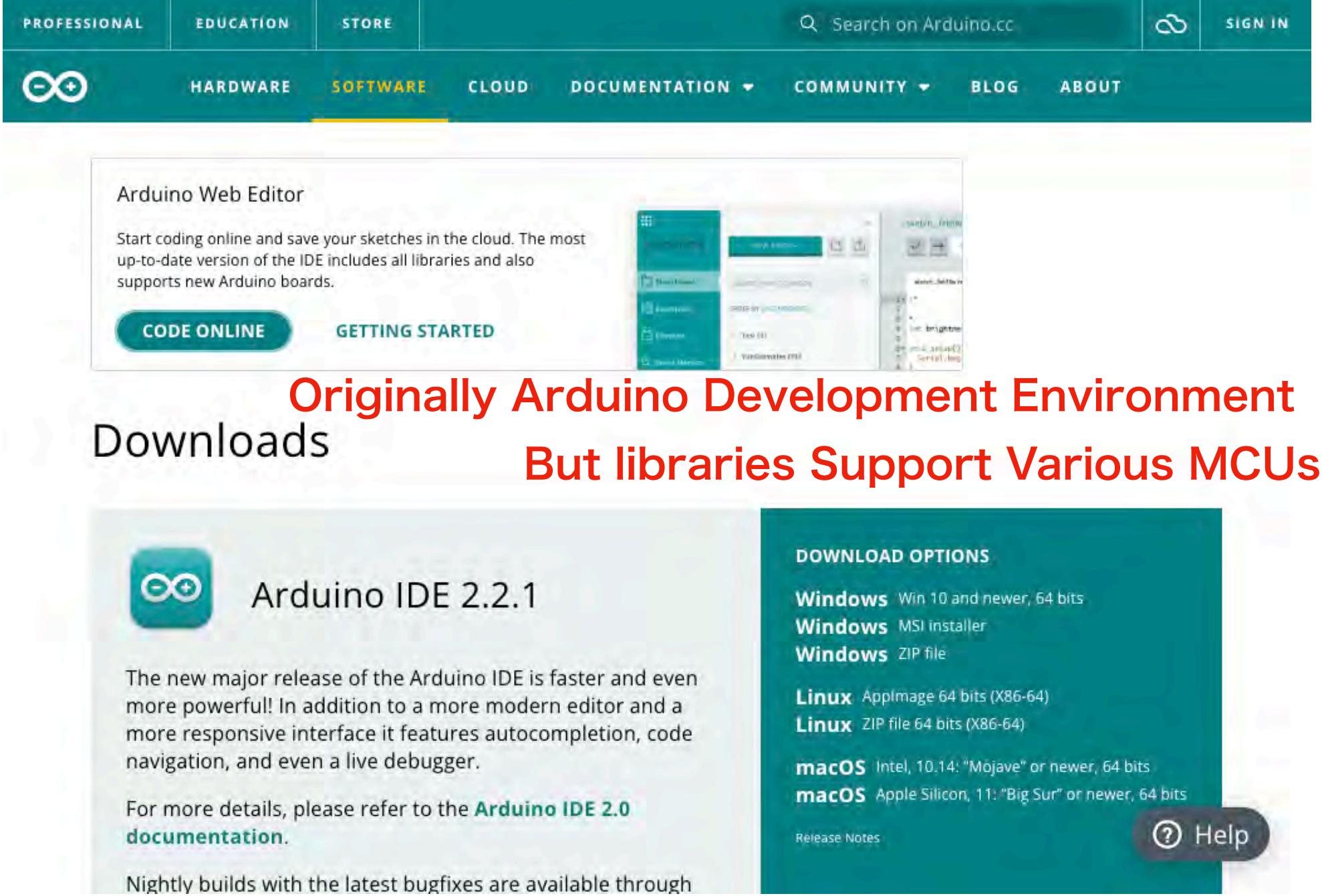
nRF52840

ARM Cortex M4

may be used in Challenge

Each company's original IDE (Integrated Development Environment)

Generic development environment is useful in terms of asset utilization


PROFESSIONAL
EDUCATION
STORE
Search on Arduino.cc
SIGN IN
HARDWARE
SOFTWARE
CLOUD
DOCUMENTATION
COMMUNITY
BLOG
ABOUT
Arduino Web Editor
Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.
CODE ONLINE
GETTING STARTED
Originally Arduino Development Environment
Downloads
But libraries Support Various MCUs
Arduino IDE 2.2.1
The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.
For more details, please refer to the Arduino IDE 2.0 documentation.
Nightly builds with the latest bugfixes are available through
DOWNLOAD OPTIONS
Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file
Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)
macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits
Release Notes
Help
Based on C/C++ language

Comparison of SBC and MCU

SBC

Single Board Computer

= Like a small PC

Raspberry Pi5

Multiple Softwares

Game | Web | Python

OS — **Linux..**

Hardware — **SBC..**

Very flexible operation

require shutdown procedure

MCU

Micro Contoroller Unit

= CPU + peripheral I/F

ESP32

No OS

Single software= Firmware

e.g. Temp. Cont.

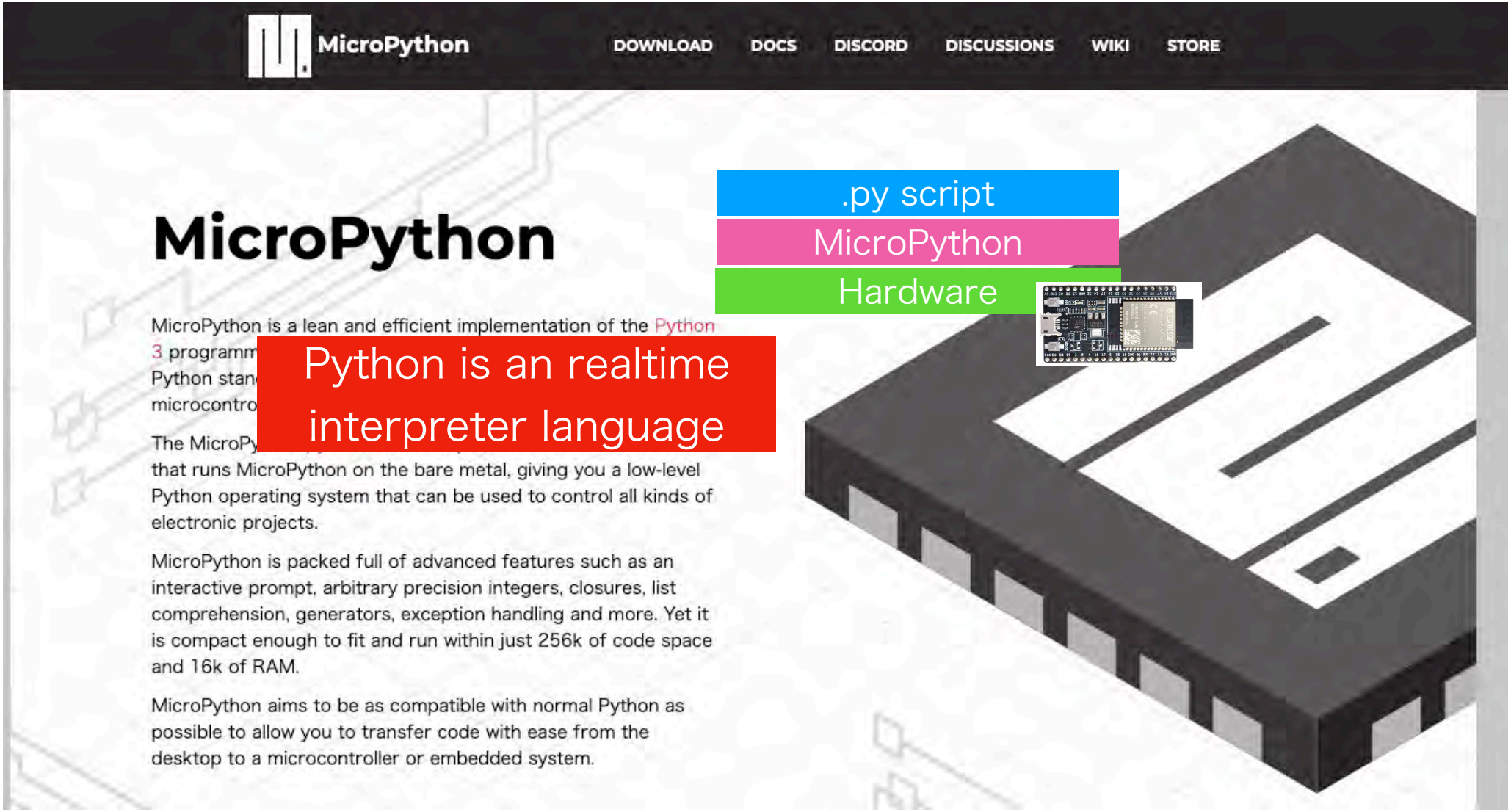
Temp. Control

Hardware

Very robust operation

just power on/off

This tutorial uses Python on the MCU as an alternative solution



Programming language for small resources compatible with Python

Run within just 256k of code space and 16k of RAM.

C++ IDE vs. μPython for development

C++

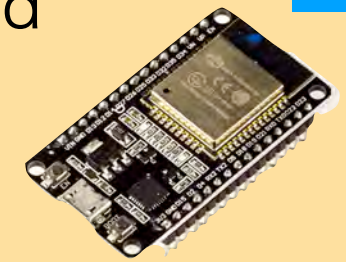
On PC

Arduino IDE

Generate binary

Upload

On MCU



Programing & Compiling

Retry

Reset, Operation check

Little complicated

MicroPython

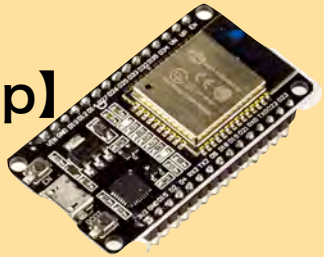
On PC

Thonny IDE

Once μPython farm is uploaded

On MCU

REPL 【Read-Evaluation-Print Loop】



Revise

Operation check

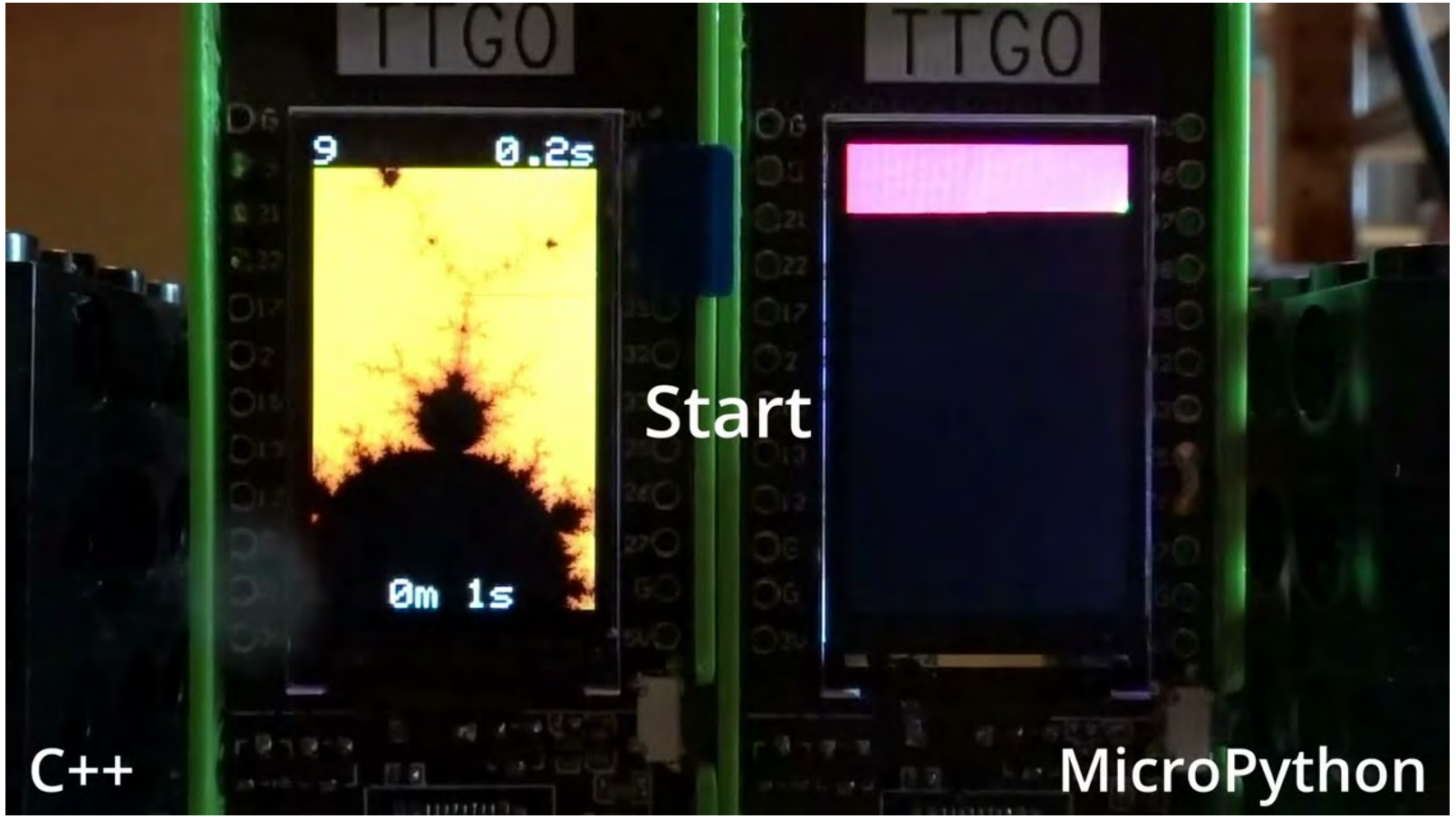
Very easy for prototyping

C++ IDE vs. μPython (performance)

r/esp32 • 2 yr. ago
TomFlatterhand

How fast is Python - MicroPython versus C++

Mandelbrot with ESP32+LED



©YouTube

C++ is 400 times faster than μPython @ ESP32

> μPython is for prototyping

> C++ is for production

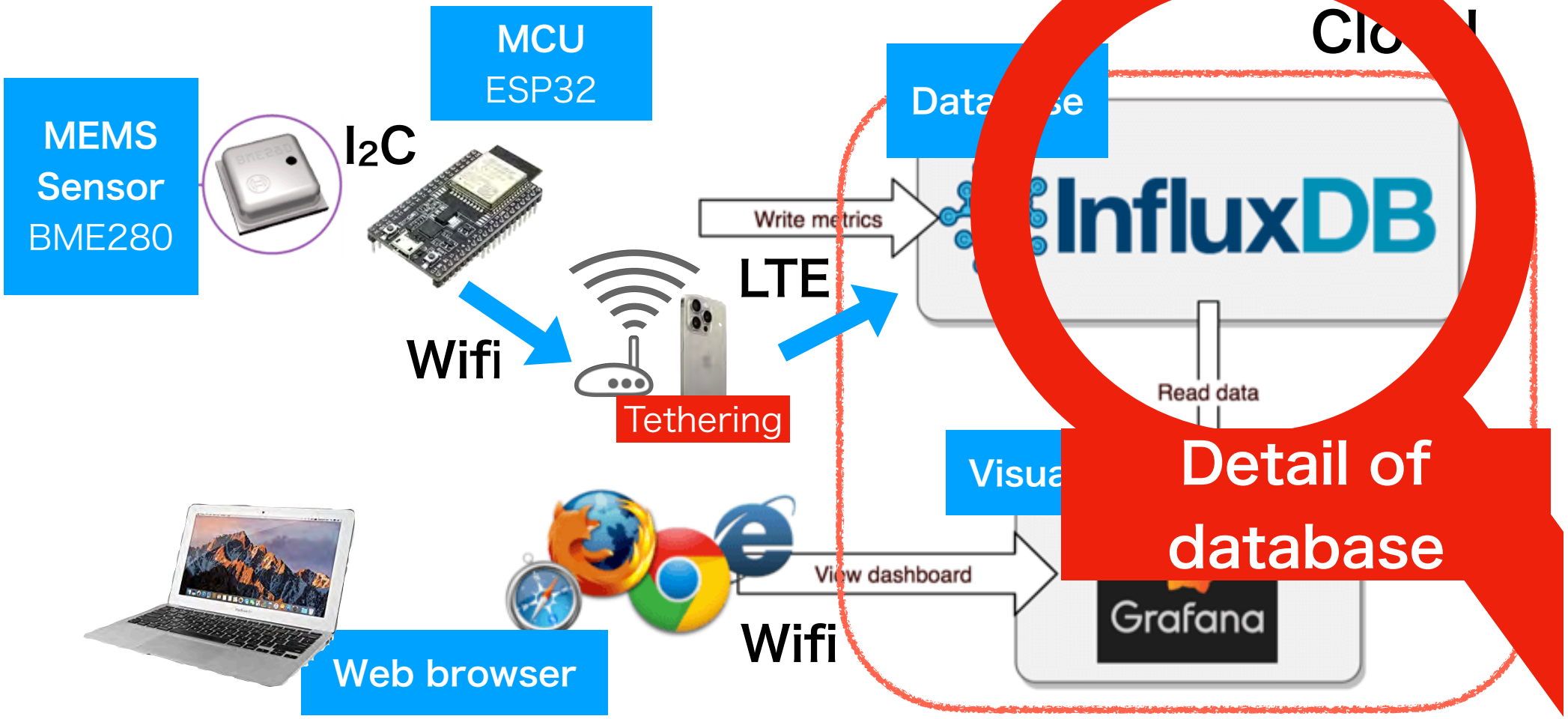
Goal of tutorial

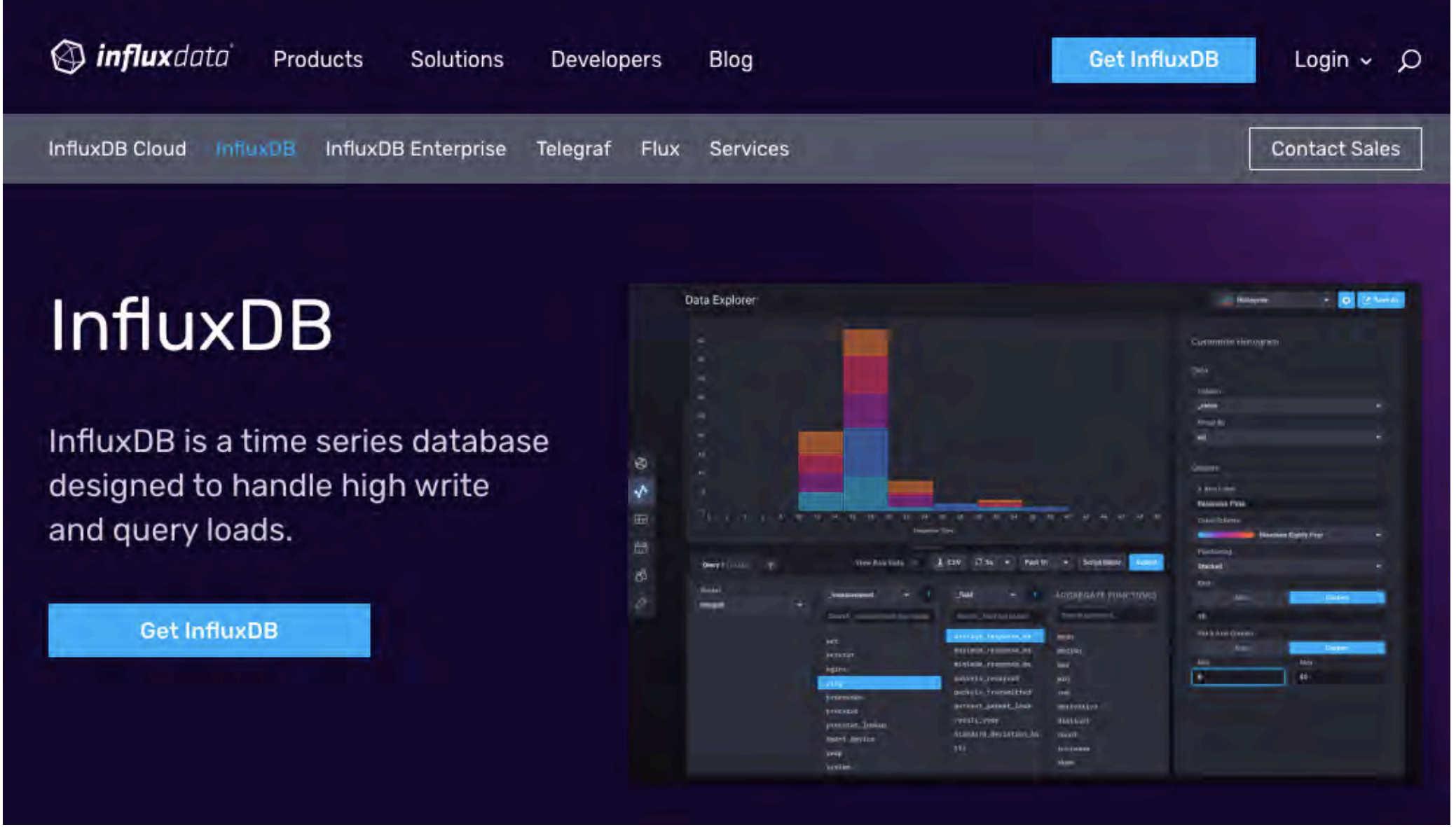
Sensing data from MEMS environmental sensor

> gather & processing data on MCU

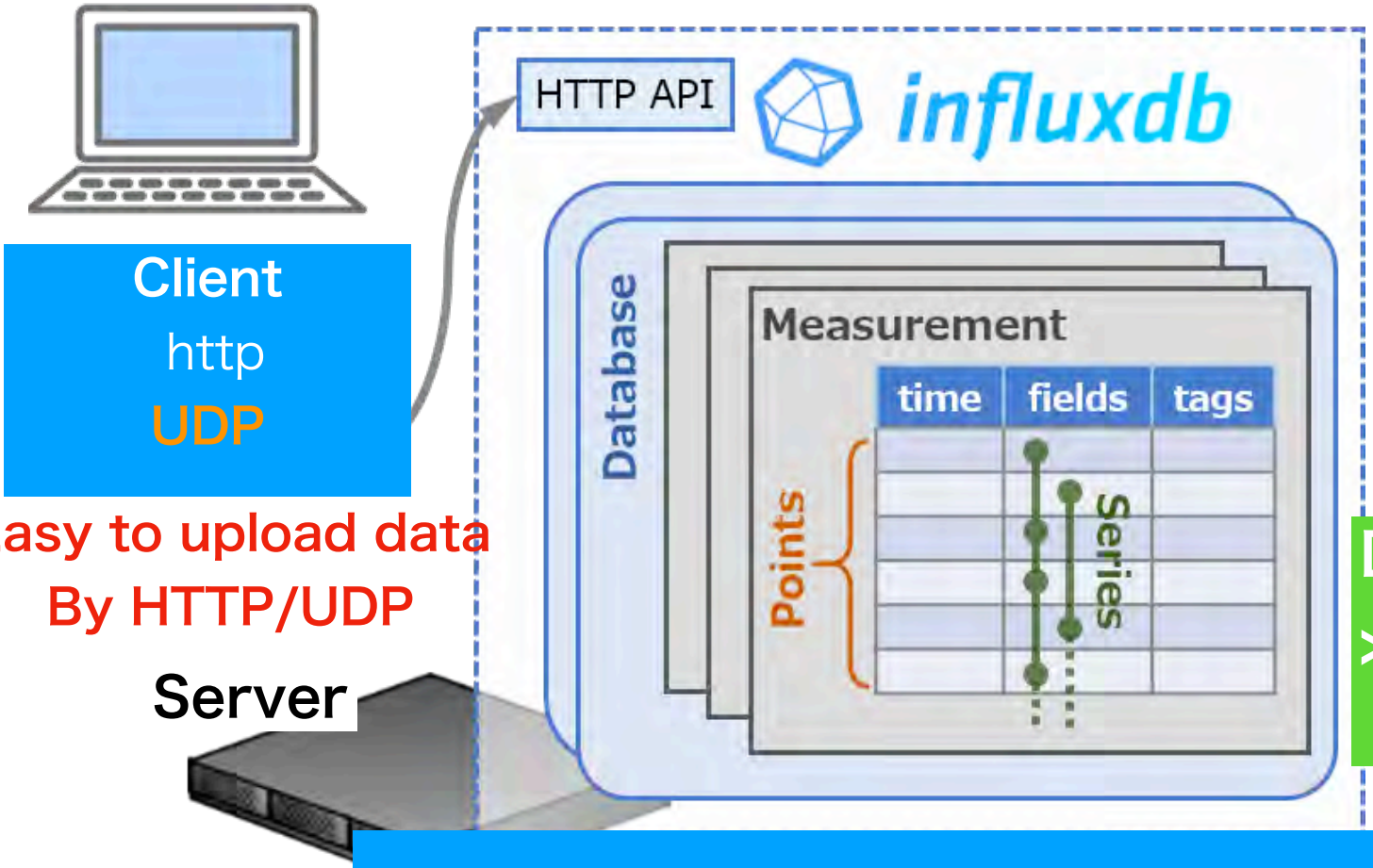
> transmit to InfluxDB database on cloud

> visualized by Grafana > display on web browser





InfluxDB is a high-performance data store written specifically for time series data. It allows for high throughput ingest, compression and real-time querying. InfluxDB is written entirely in Go and compiles into a single binary with no external dependencies. It provides write and query capabilities with a command-line interface, a built-in HTTP API.



Structure of InfluxDB database

Database

Measurement

Values

	Timestamp	Location	Temperature	Humidity	Barometer
Point	1597746439000000000	bedroom	23.4	67.8	1023.4
	1597746499000000000	bedroom	23.5	67.9	1023.6
	1597746559000000000	bedroom	23.5	68.1	1023.7

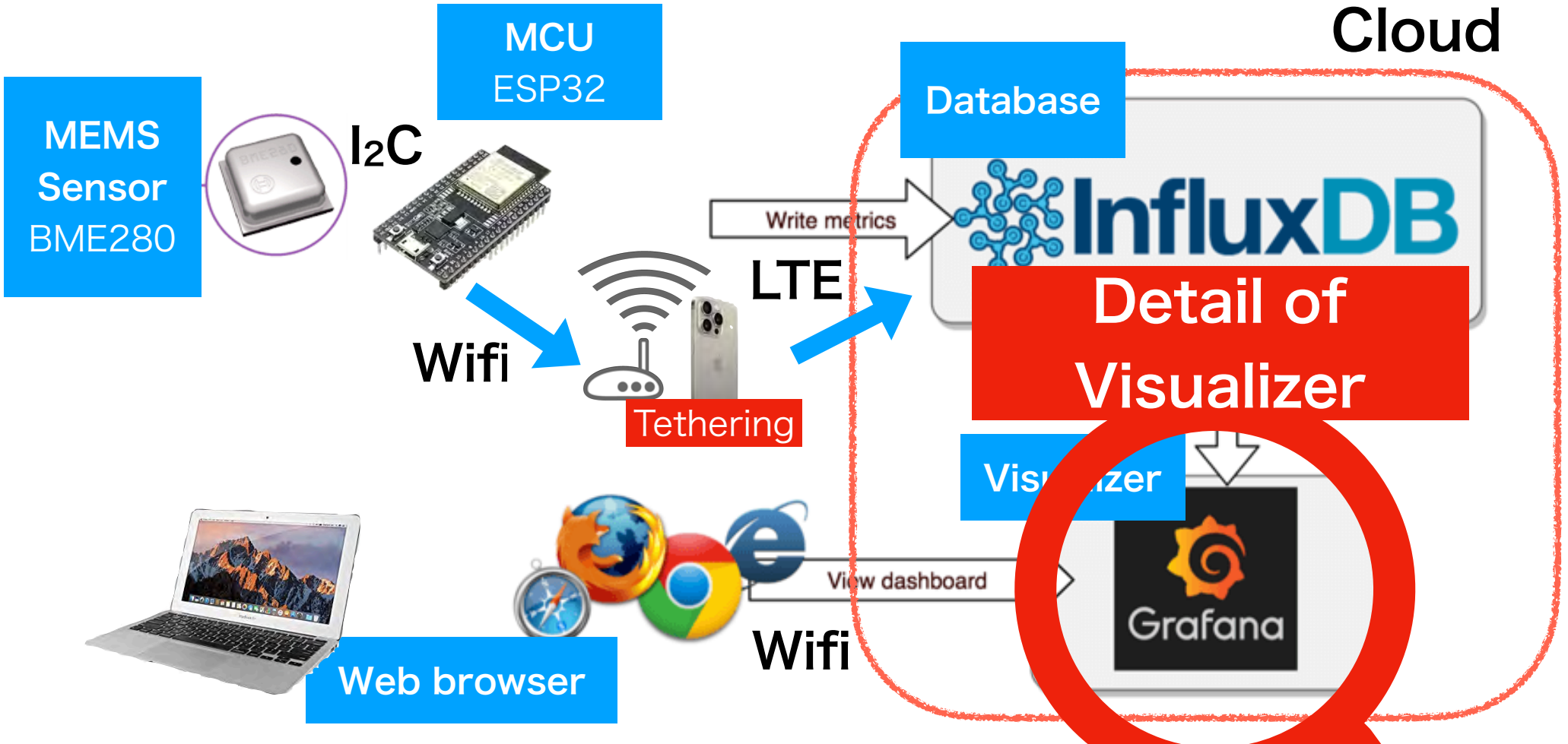
Goal of tutorial

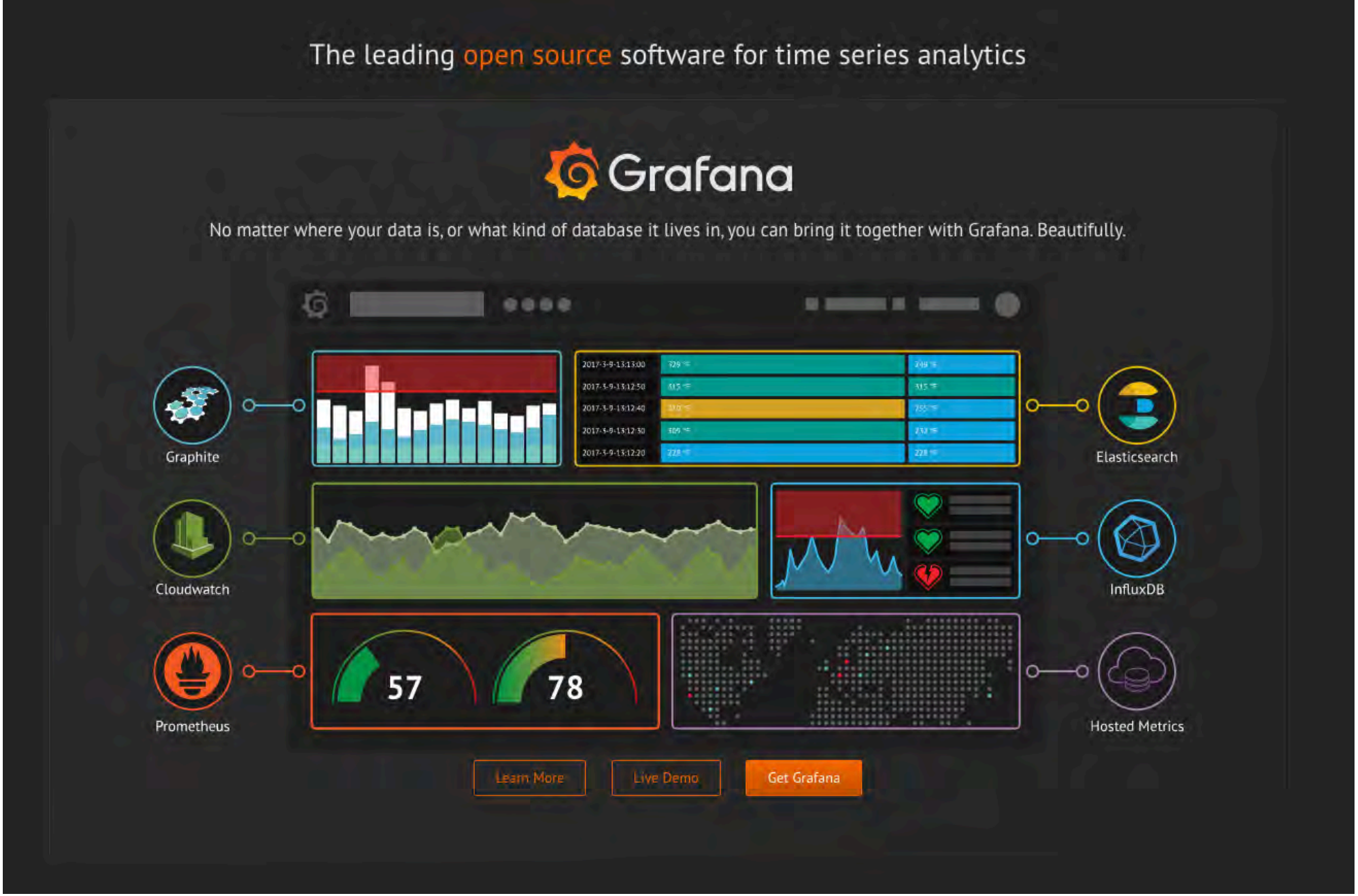
Sensing data from MEMS environmental sensor

> gather & processing data on MCU

> transmit to InfluxDB database on cloud

> visualized by Grafana > display on web browser





Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share dashboards with your team and foster a data driven culture.

Example of Grafana (my home env.) on RPi

Tutorial list

Instruction & sample program

http://iotwork.org/smb/

Preparation

0. MicroPython install on ESP32C3 by Thonny IDE

Experiments

1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP ~~from ESP32~~
   Revised > Data send by UART to PC > UDP upload
5. Visualization on Grafana

(Extra) BLE Beacon RSSI monitoring on Grafana

Today

90' Lecture
10:00 - 11:30
T. Fujita, University of Hyogo, JP
IoT modules & its tutorial by using SBC
BXXX

Lunch break
11:30-13:15

This afternoon

Challenge
13:15 - 14:45
Group : C

90' Tutorial - 13:15 - 14:45
H. Sawada
Image processing for Mechatronics
CXXX - Group : B

90' Practical work
13:15 - 14:45
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : A
A

Split to 2 small groups ex.
Group A1 → A1a, A1b (mixed universities)

Thursday Sept 18

45' Keynote
9:00 - 9:45
H. Sawada
AI

Coffe Break

enge: Start
15 - 11:45
oups : B

90' Practical work
10:15 - 11:45
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : C
C

90' Tutorial - 10:15 - 11:45
H. Sawada
Image processing for Mechatronics
CXXX - Group : A

Friday Sept 19

90' Lecture
8:15 - 9:45
Y. Suzuki, University of Tokyo
Design of broadband vibration energy harvesting device
BXXX

Coffe Break

Challenge
10:00 - 11:30
Group : C

90' Practical work
10:00 - 11:30
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : B
B

90' Practica
10:00 - 1
Y. Suzu
Design of bro
vibration energy
devic
CXXX - Gro

Before the tutorial

Split to 2 small teams

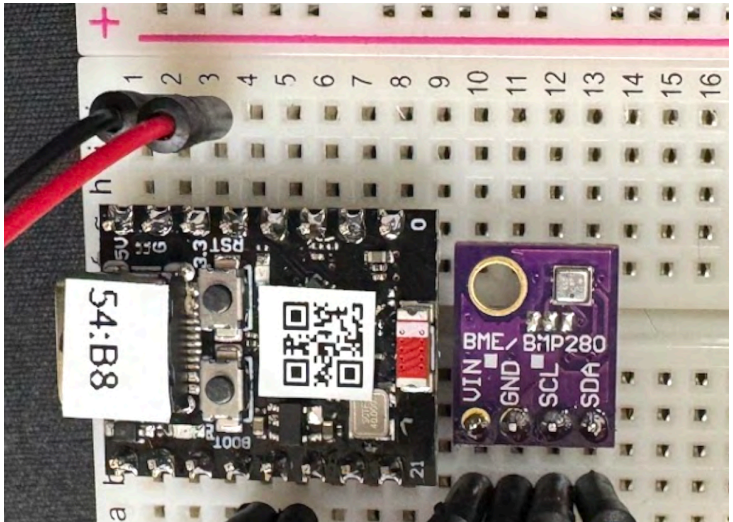
ex. Team A1 → A1a A1b

(mixed universities)

GROUP A				TEAM
PHILIPPE	Nicolas	MAM2	FRA	A1a
ENATSU	Koichi	Tokyo	JPN	
TAKAGI	Yukihiro	Waseda	JPN	A1
CRIADO	Romain	FISA5	FRA	
GHERBI	Choayb	FISA5	FRA	A1b
VALLIER	Lucillien	FISA5	FRA	
REY	Dimitri	FISA5	FRA	
DJERABE	Gauthier	MAM2	FRA	
WANG	Koichi	Tokyo	CHN	A2a
TODA	Ittetsu	Waseda	JPN	
SHIOKAWA	Yuma	Waseda	JPN	
GUIA	Nicolas	FISA5	FRA	A2
ANANI	Nawel	FISA5	FRA	A2b
BERNIER	Théo	FISA5	FRA	
MIRET	Thibault	FISA5	FRA	

GROUP B				TEAM
[illegible]	Julien	MAM2	FRA	B1a
RAJERISON	Lorry	FISA5	FRA	
AMANO	Sota	Waseda	JPN	
NAGAMINE	Yuma	Waseda	JPN	B1
UMEKI	Shusuke	Tokyo	JPN	
PESSÉ	Victor	FISA5	FRA	B1b
BAZIN	Khalil	FISA5	FRA	
DOIX	Paul	FISA5	FRA	
PALLIER	Tristan	MAM2	FRA	
SATAKE	Rito	Waseda	JPN	B2a
KAWAHARA	Yuto	Waseda	JPN	
LEMOS	Hervé	FISA5	JPN	
BEAUFORT	Nicolas	FISA5	FRA	B2
FONSECA SOUSA	Lucas	FISA5	FRA	B2b
KHATRI	Khaoula	FISA5	FRA	
MOUHAMMAD	Hamza	MAM2	FRA	

GROUP C				TEAM
THIAW	Sidy	MAM2	FRA	C1a
SALAMEH	Ahmad	MAM2	FRA	
OKAWA	Yutaro	Waseda	JPN	
TSUDA	Kaoru	Waseda	JPN	C1
CARRERA	Alexis	FISA5	FRA	
BEAUTIER	Antoine	FISA5	FRA	C1b
LO	Baye Djily	FISA5	FRA	
FRANCHINI	Antoine-Hugo	FISA5	FRA	
LATRECHE	Zakaria	MAM2	LBN	
CHIKUSA	Yuto	Waseda	JPN	C2a
YUZAWA	Koki	Waseda	JPN	
SHIMURA	Yuto	Tokyo	JPN	
ROQUES	Alexandre	FISA5	FRA	C2
GAILLARD	Adrien	FISA5	FRA	C2b
HEBERT	Tanguy	FISA5	FRA	
SAUNIER	Jérémie	FISA5	FRA	



Each group uses ESP32C3 + BME280 sensor

+you own PC

End of my talk

See you in Tutorial

Tutorial instruction

Today

90' Lecture
10:00 - 11:30
T. Fujita, University of Hyogo, JP
IoT modules & its tutorial by using SBC
BXXX

Lunch break
11:30-13:15

This afternoon

Challenge
13:15 - 14:45
Group : C

90' Tutorial - 13:15 - 14:45
H. Sawada
Image processing for Mechatronics
CXXX - Group : B

90' Practical work
13:15 - 14:45
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : A
A

Split to 2 small teams

ex.

Group A1 → A1a, A1b
(mixed universities)

Thursday Sept 18

45' Keynote
9:00 - 9:45
H. Sawada
AI

Coffe Break

enge: Start
15 - 11:45
oups : B

90' Practical work
10:15 - 11:45
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : C
C

90' Tutorial - 10:15 - 11:45
H. Sawada
Image processing for Mechatronics
CXXX - Group : A

Friday Sept 19

90' Lecture
8:15 - 9:45
Y. Suzuki, University of Tokyo
Design of broadband vibration energy harvesting device
BXXX

Coffe Break

Challenge
10:00 - 11:30
Group : C

90' Practical work
10:00 - 11:30
T. Fujita
IoT modules tutorial using SBC
CXXX - Group : B
B

90' Practica
10:00 - 1
Y. Suzu
Design of bro
vibration energy
devic
CXXX - Gro

Split to 2 small teams

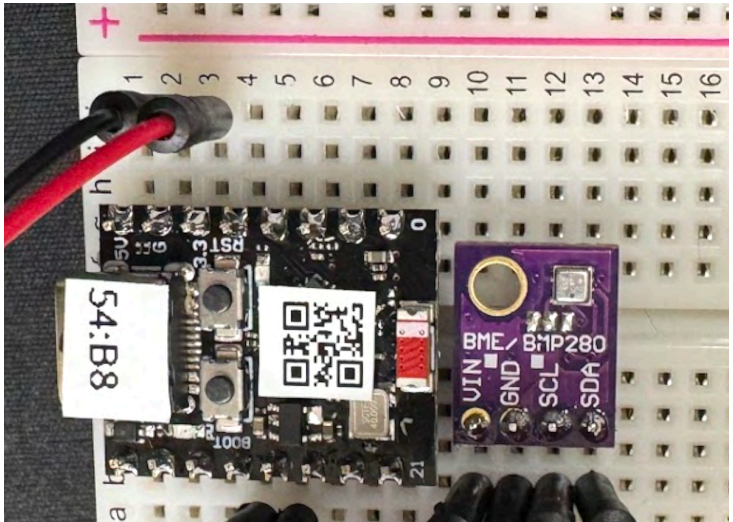
ex. Team A1 → A1a A1b

(mixed universities)

GROUP A				TEAM
PHILIPPE	Nicolas	MAM2	FRA	A1a
ENATSU	Koichi	Tokyo	JPN	
TAKAGI	Yukihiro	Waseda	JPN	A1
CRIADO	Romain	FISA5	FRA	
GHERBI	Choayb	FISA5	FRA	A1b
VALLIER	Lucillien	FISA5	FRA	
REY	Dimitri	FISA5	FRA	
DJERABE	Gauthier	MAM2	FRA	
WANG	Koichi	Tokyo	CHN	A2a
TODA	Ittetsu	Waseda	JPN	
SHIOKAWA	Yuma	Waseda	JPN	
GUIA	Nicolas	FISA5	FRA	A2
ANANI	Nawel	FISA5	FRA	A2b
BERNIER	Théo	FISA5	FRA	
MIRET	Thibault	FISA5	FRA	

GROUP B				TEAM
NOIREAUX-FATTAZ	Julien	MAM2	FRA	B1a
RAJERISON	Lorry	FISA5	FRA	
AMANO	Sota	Waseda	JPN	
NAGAMINE	Yuma	Waseda	JPN	B1
UMEKI	Shusuke	Tokyo	JPN	
PESSÉ	Victor	FISA5	FRA	B1b
BAZIN	Khalil	FISA5	FRA	
DOIX	Paul	FISA5	FRA	
PALLIER	Tristan	MAM2	FRA	
SATAKE	Rito	Waseda	JPN	B2a
KAWAHARA	Yuto	Waseda	JPN	
LEMOS	Hervé	FISA5	JPN	
BEAUFORT	Nicolas	FISA5	FRA	B2
FONSECA SOUSA	Lucas	FISA5	FRA	B2b
KHATRI	Khaoula	FISA5	FRA	
MOUHAMMAD	Hamza	MAM2	FRA	

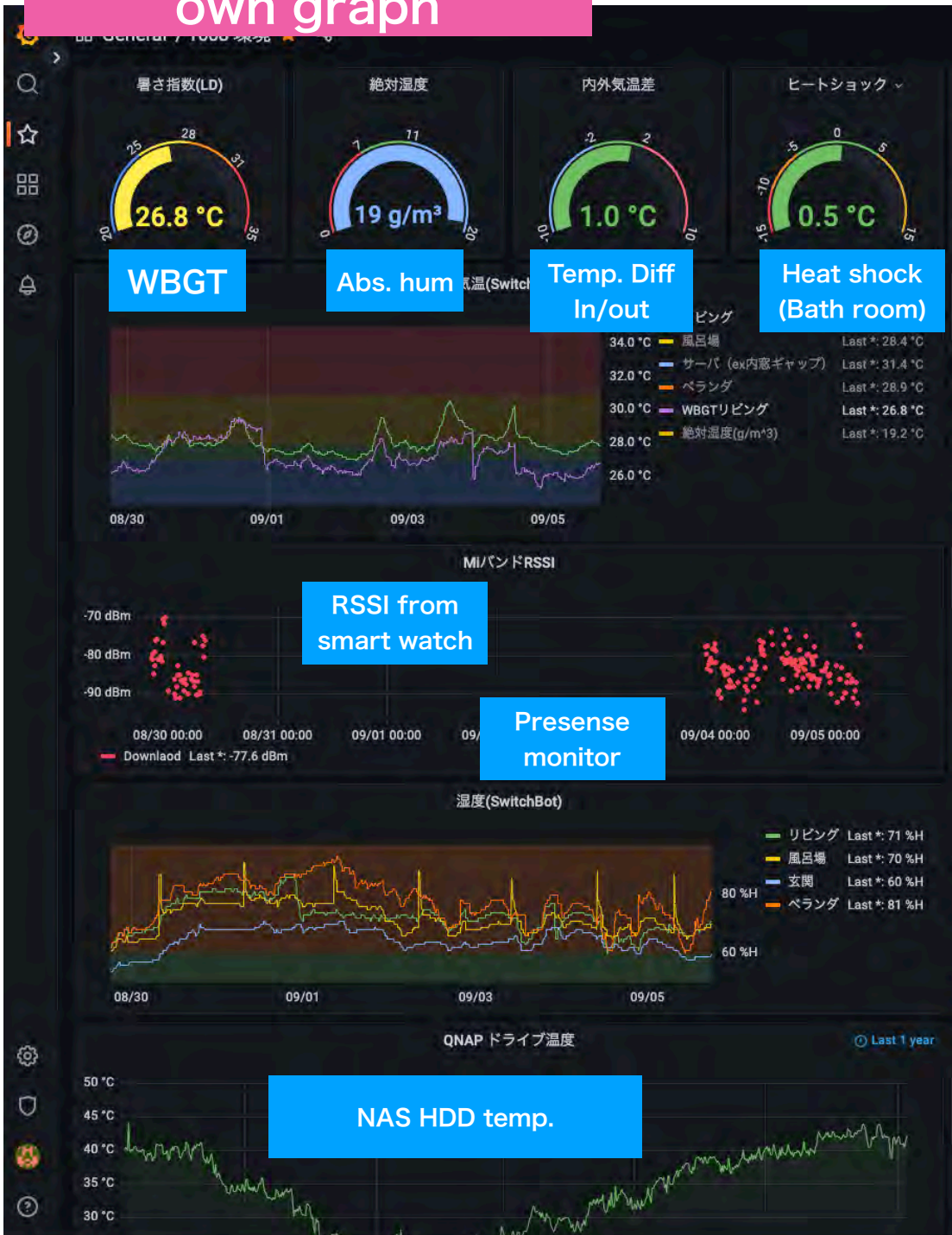
GROUP C				TEAM
THIAW	Sidy	MAM2	FRA	C1a
SALAMEH	Ahmad	MAM2	FRA	
OKAWA	Yutaro	Waseda	JPN	
TSUDA	Kaoru	Waseda	JPN	C1
CARRERA	Alexis	FISA5	FRA	
BEAUTIER	Antoine	FISA5	FRA	C1b
LO	Baye Djily	FISA5	FRA	
FRANCHINI	Antoine-Hugo	FISA5	FRA	
LATRECHE	Zakaria	MAM2	LBN	
CHIKUSA	Yuto	Waseda	JPN	C2a
YUZAWA	Koki	Waseda	JPN	
SHIMURA	Yuto	Tokyo	JPN	
ROQUES	Alexandre	FISA5	FRA	C2
GAILLARD	Adrien	FISA5	FRA	C2b
HEBERT	Tanguy	FISA5	FRA	
SAUNIER	Jérémie	FISA5	FRA	



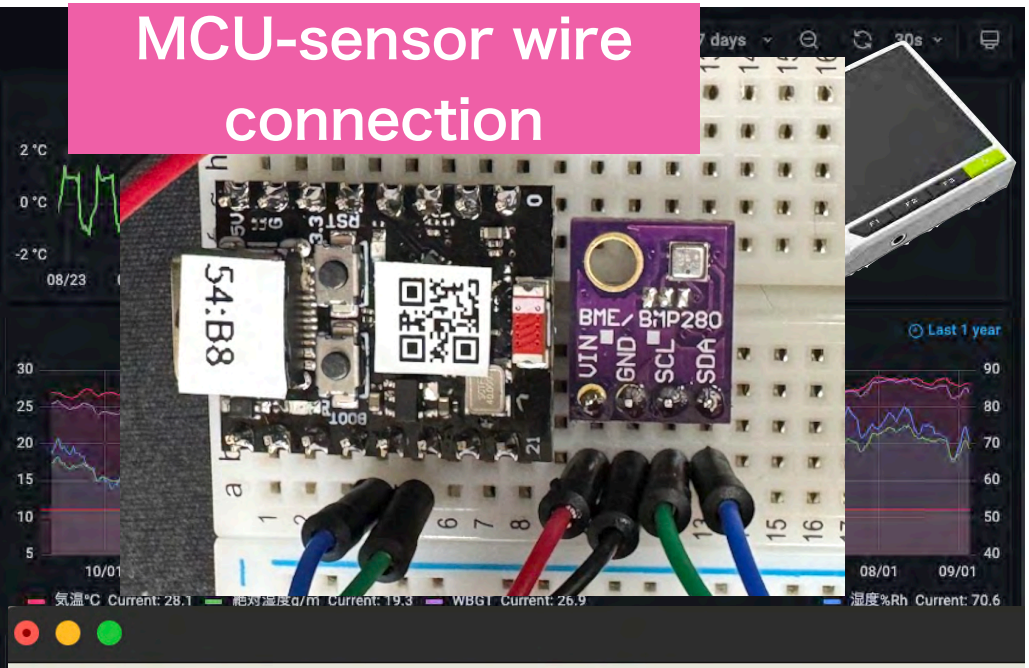
Each group uses ESP32C3 + BME280 sensor

+you own PC

Final goal : Make your own graph



MCU-sensor wire connection



<untitled> * [BME280.py]

```
1  from machine import Pin, SoftI2C, unique_id
2  from time import sleep
3  import BME280
4  import socket
5  import ubinascii
6  import network
7  import time
8
9  # BME280 setup
10 SCL = 22
11 SDA = 21
12 i2c = SoftI2C(s
13 bme = BME280.BM
14
15 # Wi-Fi credent
16 SSID = "JFWM-WS
17 PASSWORD = "goodlife
18
19 print(f"Connecting to Wi-Fi network: {SSID}")
```

µPython Programming (Copy & paste) on ESP32

Tutorial list

Instruction & sample program

http://iotwork.org/smb/

Preparation

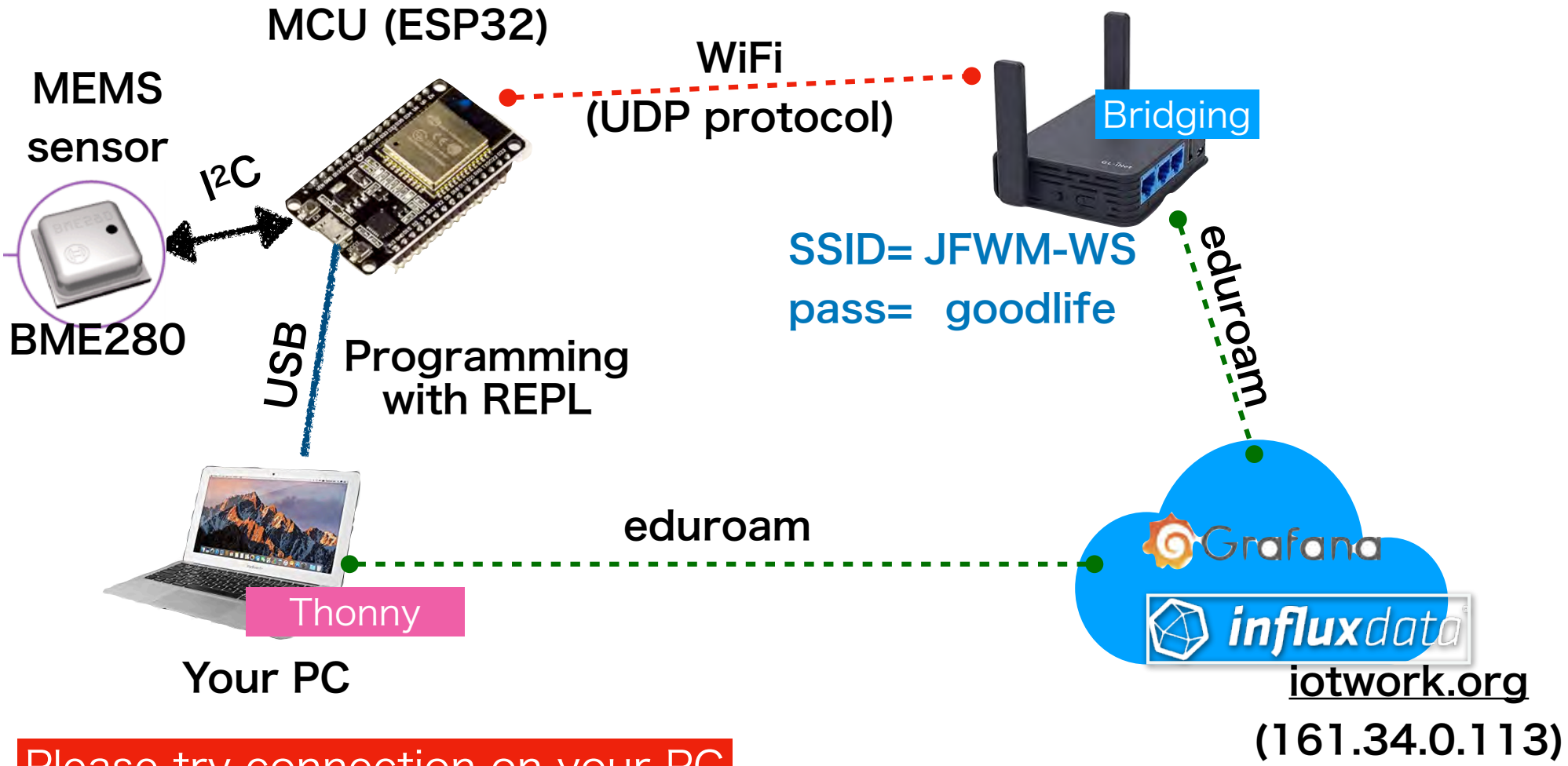
0. MicroPython install on ESP32C3 by Thonny IDE

Experiments

1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP ~~from ESP32~~
   Revised > Data send by UART to PC > UDP upload
5. Visualization on Grafana

(Extra) BLE Beacon RSSI monitoring on Grafana

Overview of connections



Please try connection on your PC

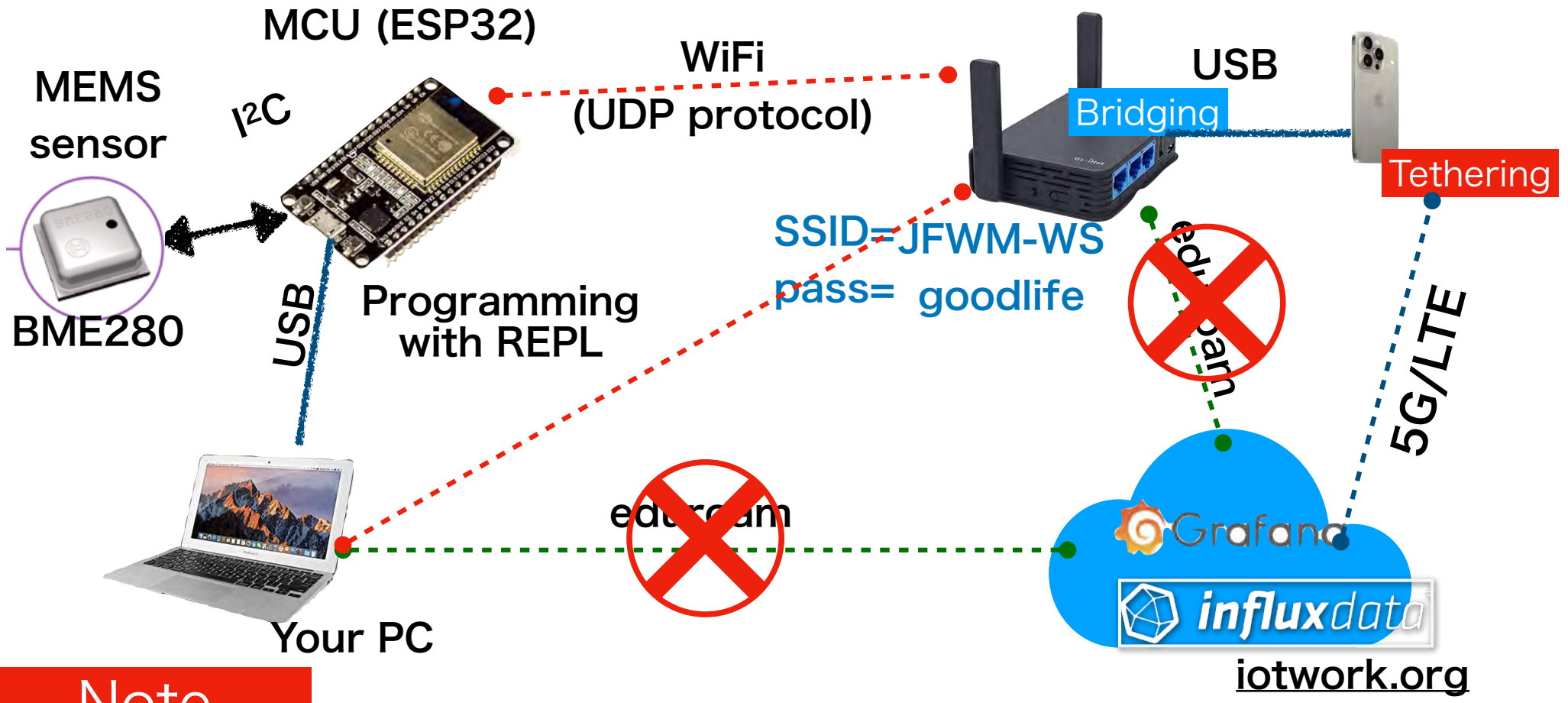
1) Can you to connect http://iotwork.org/smb/ for instructions ?

Yes

2) Can you to connect http://iotwork.org:3000 for Grafana ?

Might be No

Overview of connections (revised)

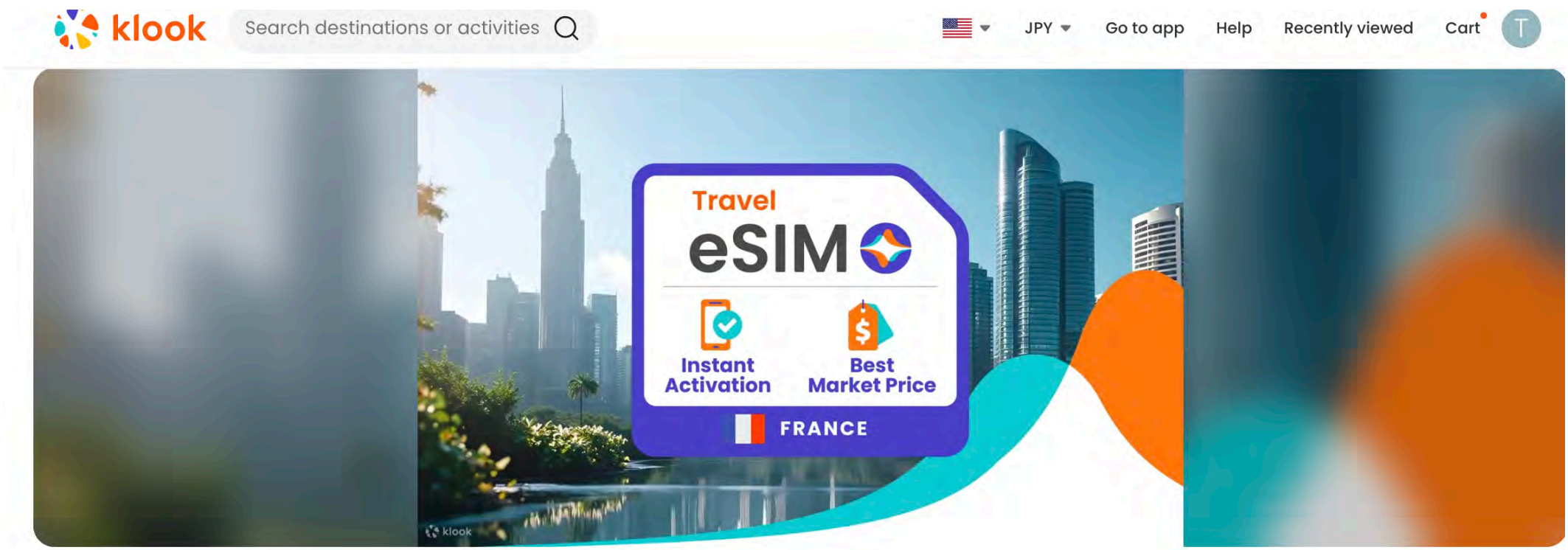


Note

eduroam@USMB's security policy is restricting the Grafana communication

Please connect to 'JFWM-WS' during the tutorial to connect to Grafana.

I provided eSim for 3 days unlimited in France



Klook Travel

5G France eSIM | Orange SFR

★ 4.8 (29 reviews) · 4K+ booked

♡ Save to wishlist

¥ 1,018

See selected options

Let's connect ESP32 to PC

Connect ESP32 board to Windows PC /Mac w/μ-USB cable

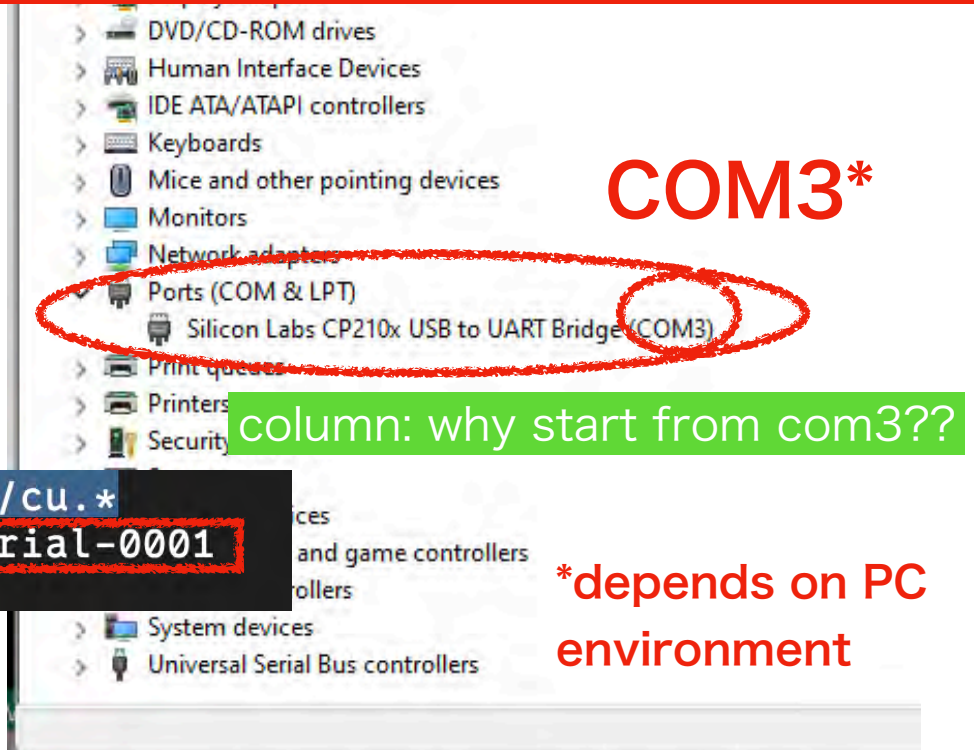
Confirm COM port number on Device Manager

Trouble??

Google

cp210x driver windows

Download the driver first, I will help you.



COM3*

column: why start from com3??

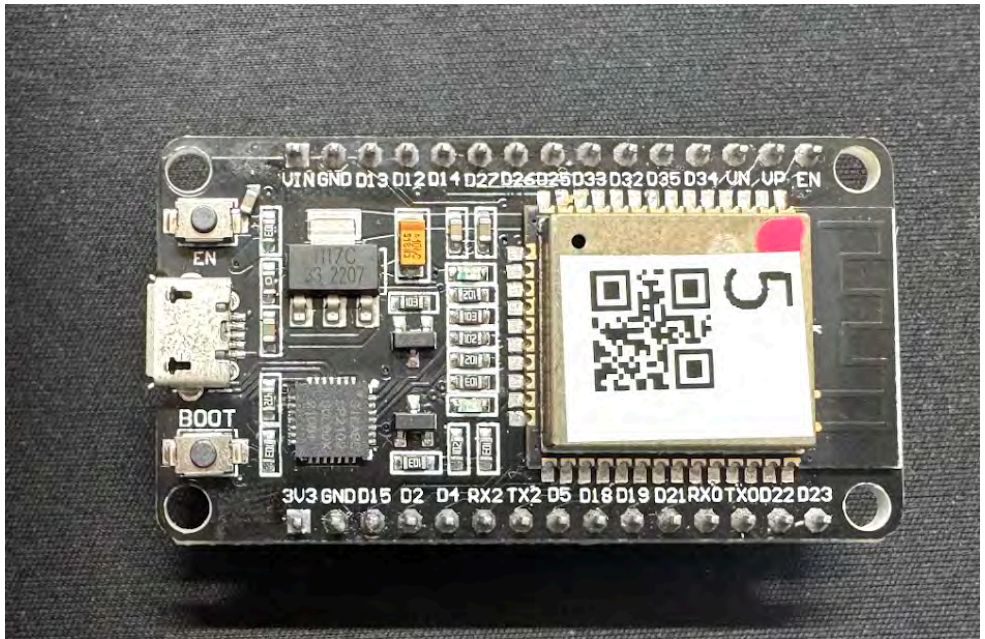
*depends on PC environment

On MacOS Terminal App. just input

ls /dev/cu.*

```
(base) takayukifujita@MacBook-Air ~ % ls /dev/cu.*
/dev/cu.BNO085-BT                 /dev/cu.usbserial-0001
/dev/cu.Bluetooth-Incoming-Port
```

ESP32 -> ESP32C3 super mini

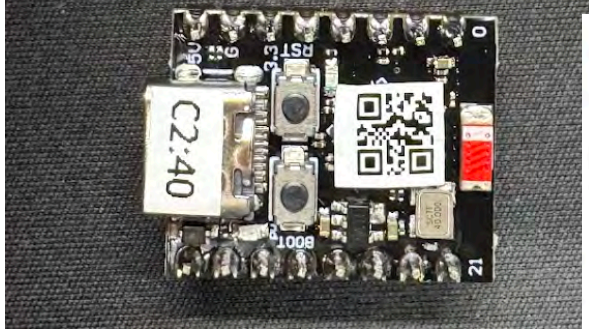


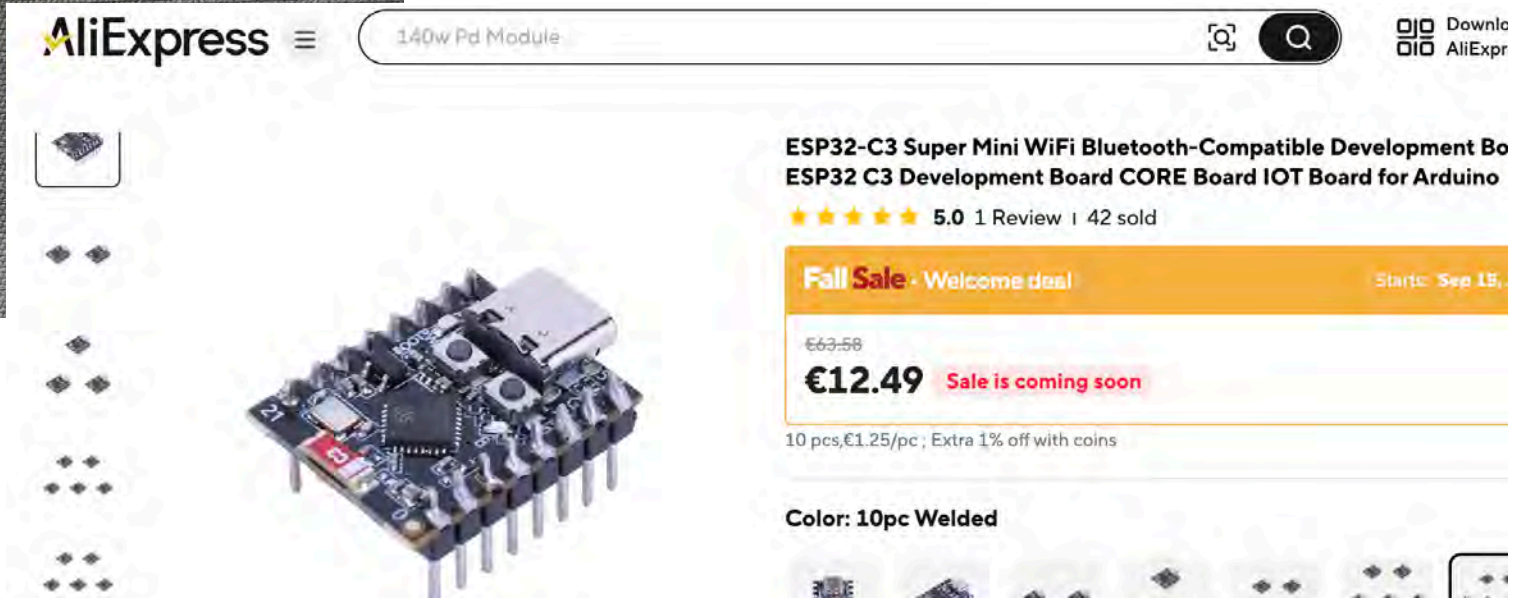
C3 super mini in JFWM2025

very cheap and small form factor

1.25 EUROPE/chip

BUT… WiFi connection unstable





Tutorial list

Instruction & sample program

http://iotwork.org/smb/

Preparation

0. MicroPython install on ESP32C3 by Thonny IDE

Install and Start

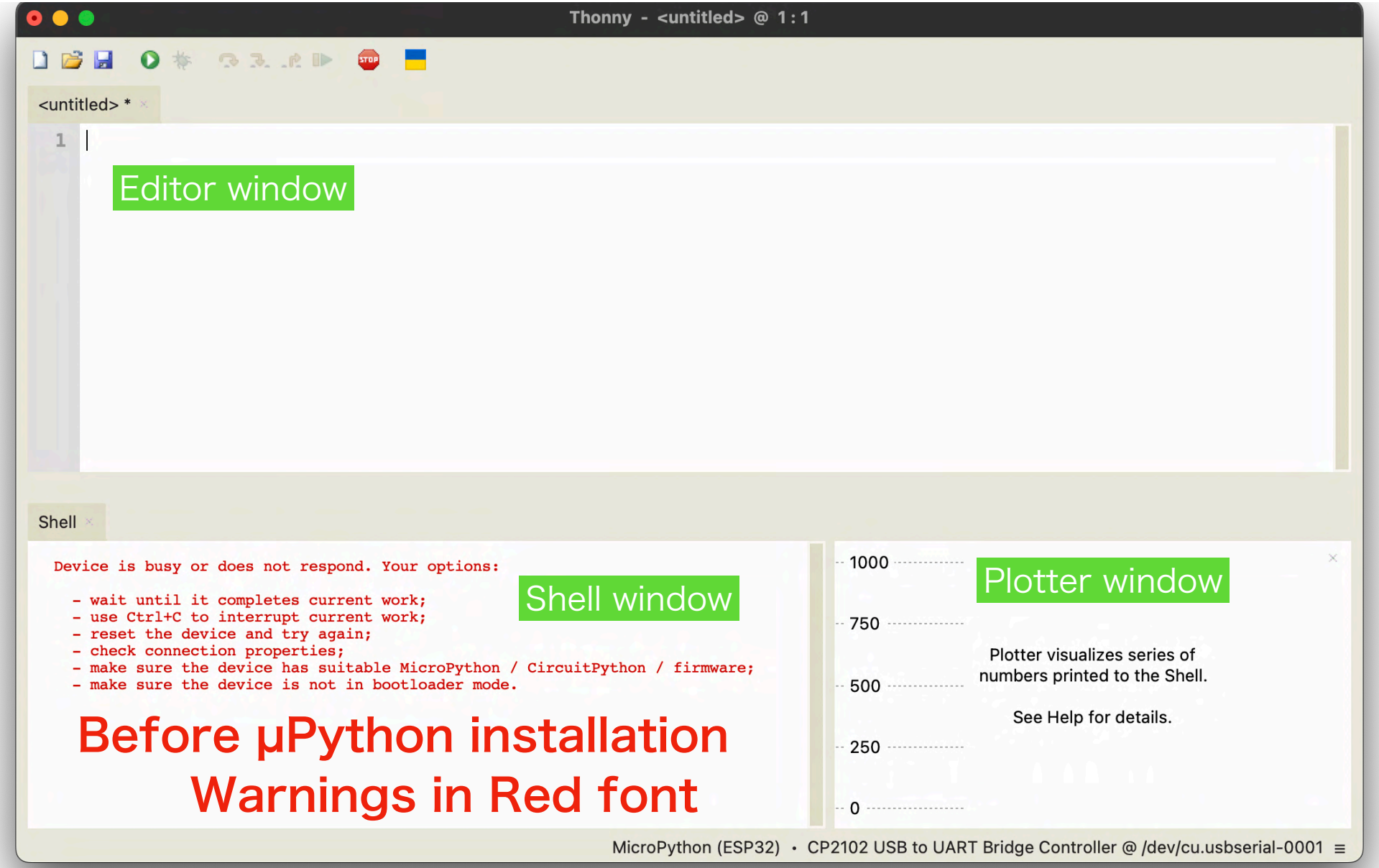
Experiments

1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP ~~from ESP32~~
   Revised > Data send by UART to PC > UDP upload
5. Visualization on Grafana

(Extra) BLE Beacon RSSI monitoring on Grafana

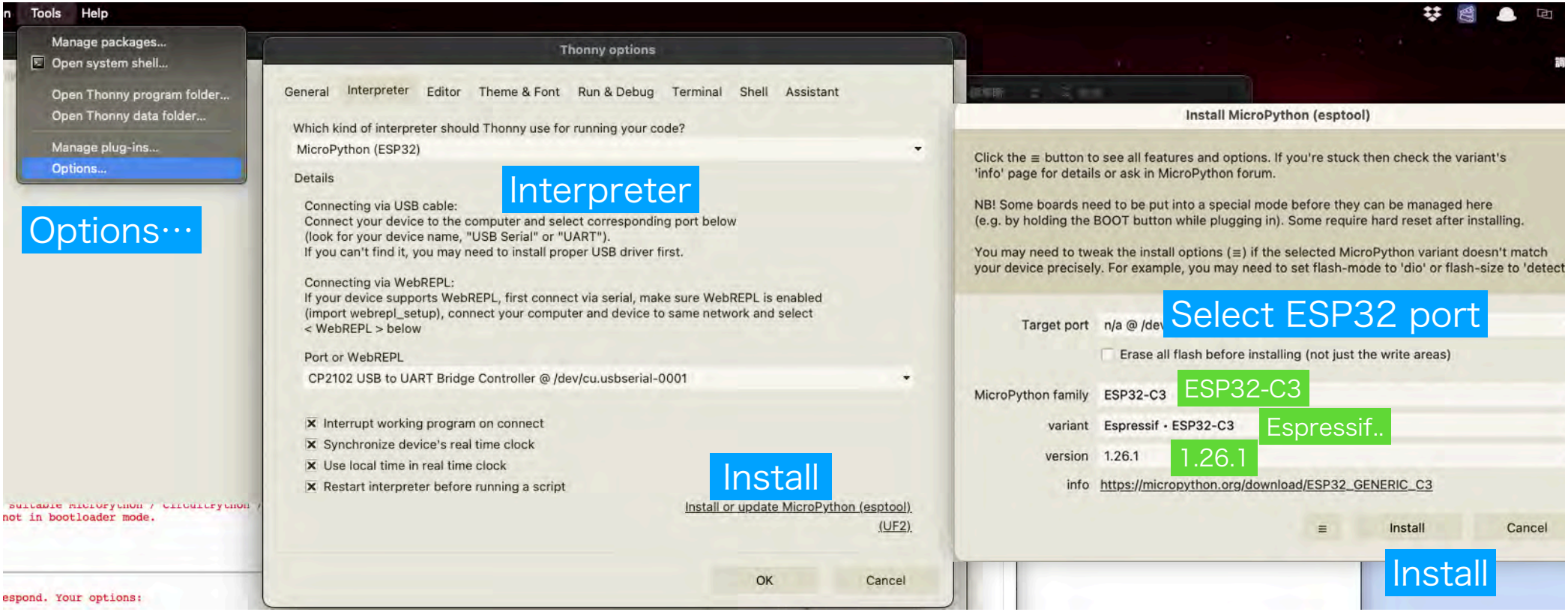
Install μPython via Thonny app

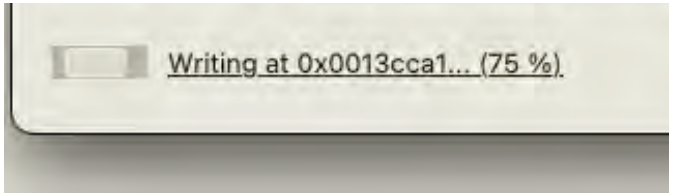
Sample of Thonny's window



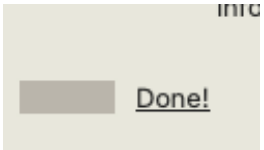
Install μPython via Thonny app

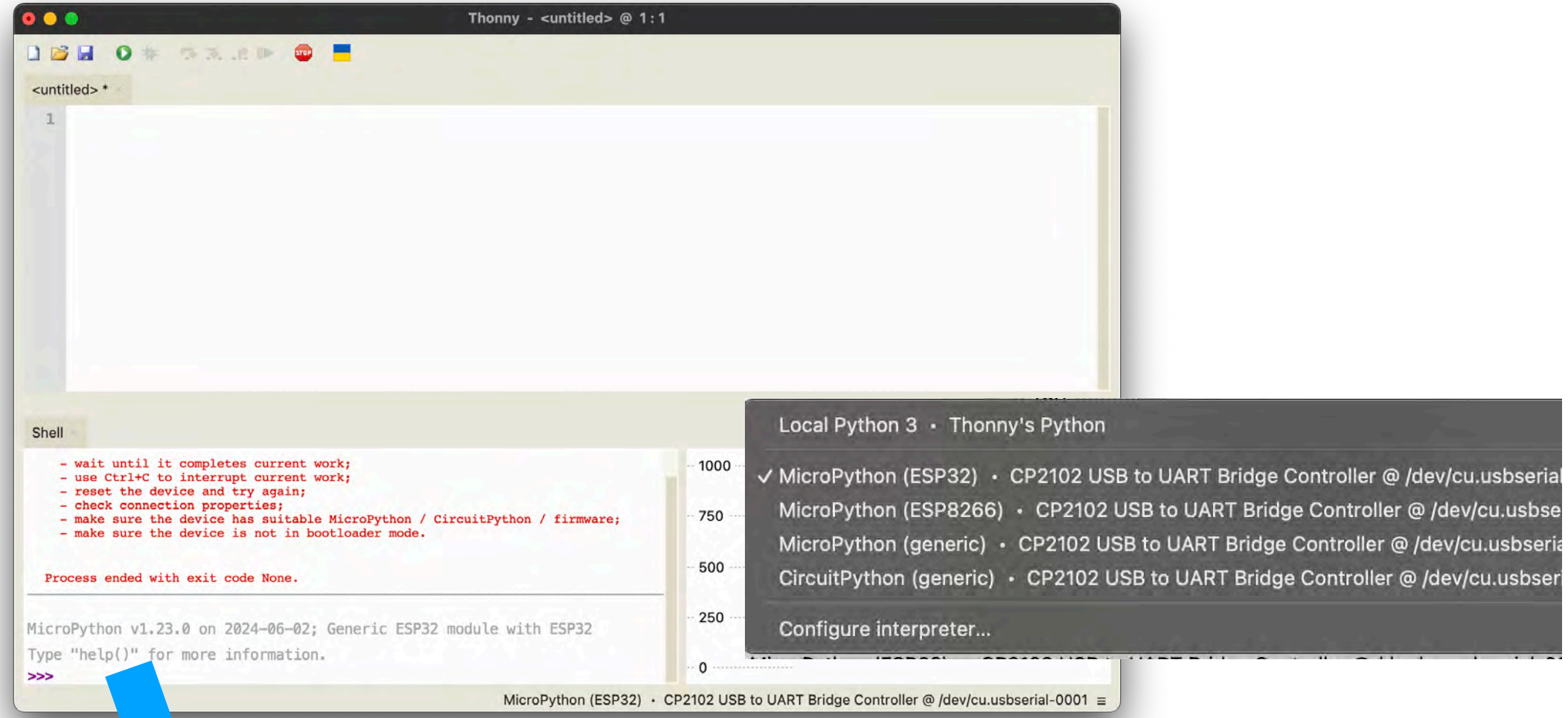
Download Thonny and install, Run





Wait for writing until “Done”





Click the right bottom to choose interpreter
(ESP32 on serial port)

Now you can use µPython on ESP32

```
MicroPython v1.23.0 on 2024-06-02; Generic ESP32 module with ESP32
Type "help()" for more information.
>>>
```

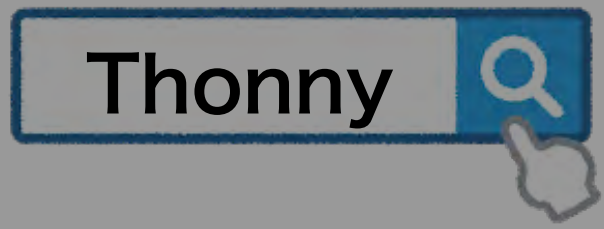
Tutorial list

Instruction & sample program

http://iotwork.org/smb/

Preparation

0. MicroPython install on ESP32C3 by Thonny IDE



Experiments

1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP ~~from ESP32~~
   Revised > Data send by UART to PC > UDP upload
5. Visualization on Grafana

(Extra) BLE Beacon RSSI monitoring on Grafana

Ex1) Blink LED

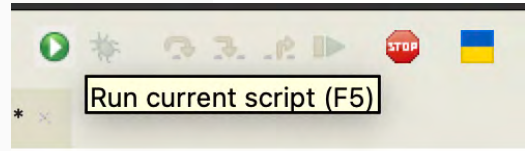
Copy & paste Python code

Thonny - <untitled> @ 12 : 39

<untitled> *

```
1  from machine import Pin
2  import time
3
4  # Initialize pin 2 for the LED as an output
5  led = Pin(2, Pin.OUT)
6
7  # Main loop
8  while True:
9      led.on()  # Turn the LED on
10     time.sleep(1)  # Wait for 1 second
11     led.off()  # Turn the LED off
12     time.sleep(1)  # Wait for 1 second
```

Then push Run



Blue LED(pin 2) blinking

Try to change blink intervals

Just change time.sleep and Run

Tutorial list

Instruction & sample program

http://iotwork.org/smb/

Preparation

0. MicroPython install on ESP32C3 by Thonny IDE

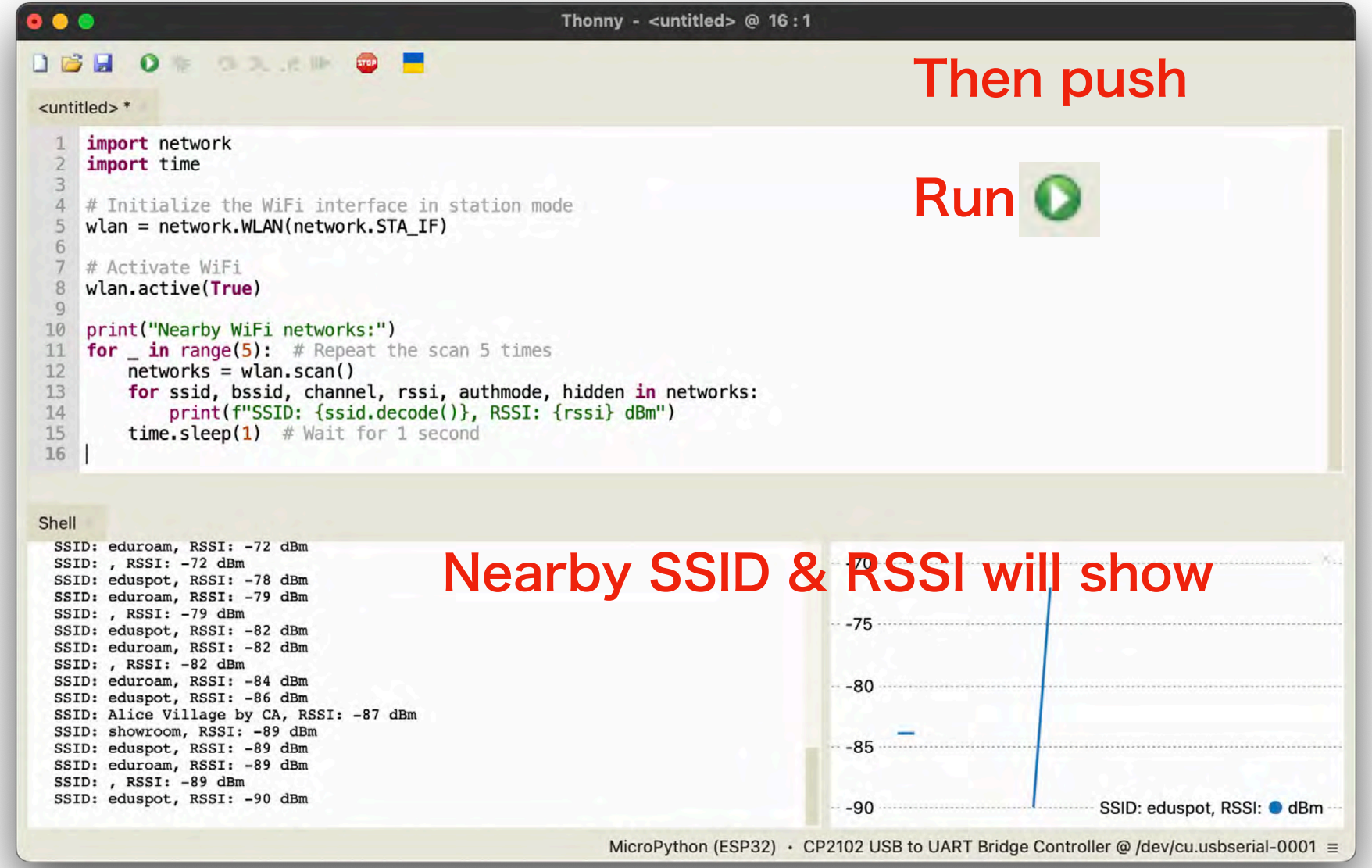
Experiments

1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP ~~from ESP32~~
   Revised > Data send by UART to PC > UDP upload
5. Visualization on Grafana

(Extra) BLE Beacon RSSI monitoring on Grafana

Ex2) Wi-Fi finder nearby

Copy & paste
Python code



Finding SSID “JFWM-WS”

Tutorial list

Instruction & sample program

http://161.34.0.113/

Preparation

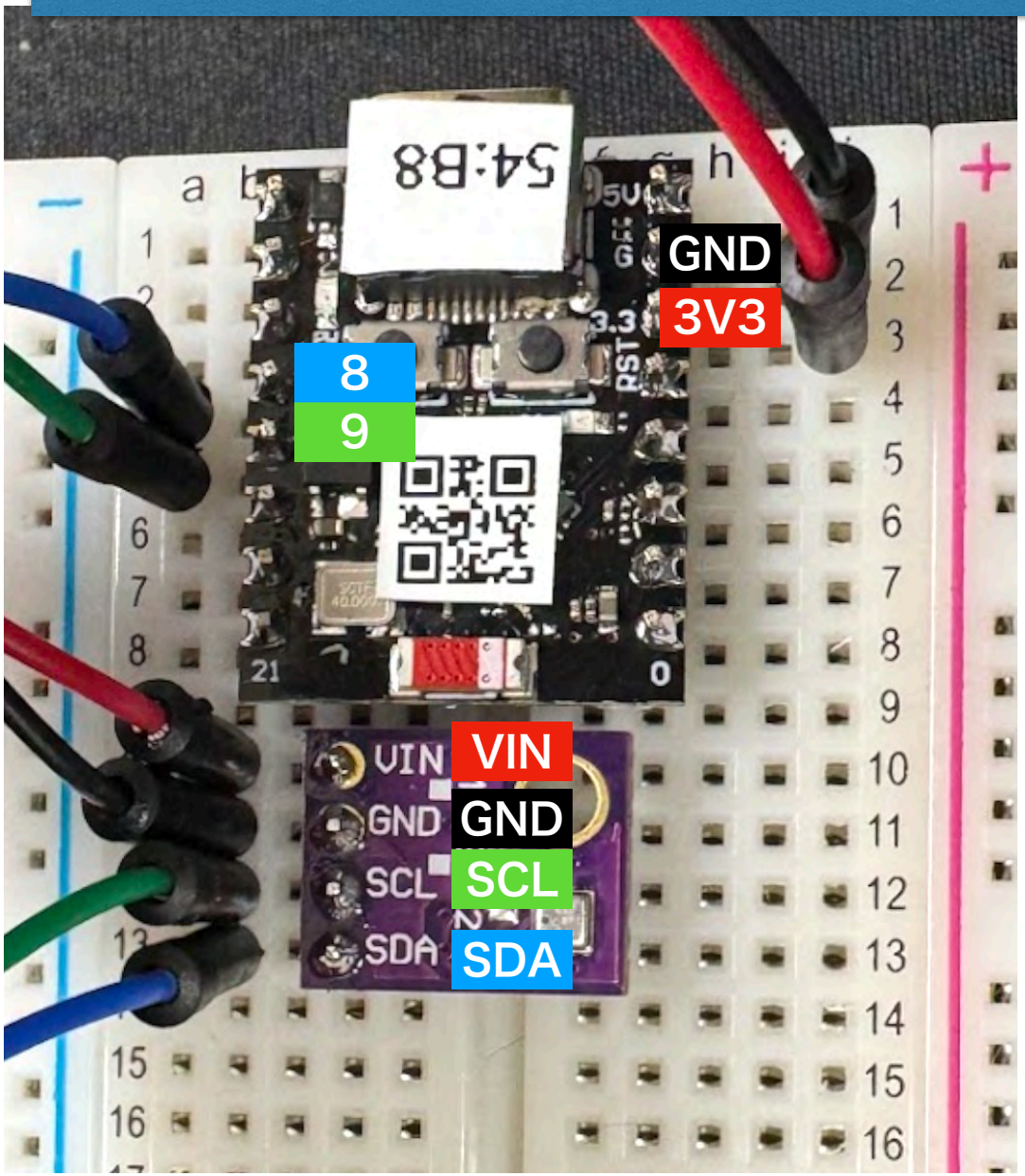
MicroPython install on ESP32 by Thonny IDE

Experiments

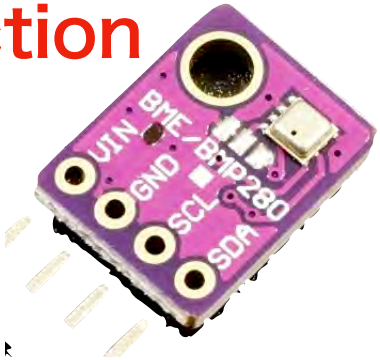
1. Blink LED on ESP32
2. Checking the WiFi function of ESP32
3. Wire BME280 to get data
   (BME280.py library installation)
4. Upload BME280 data to cloud via UDP
5. Visualization on Grafana

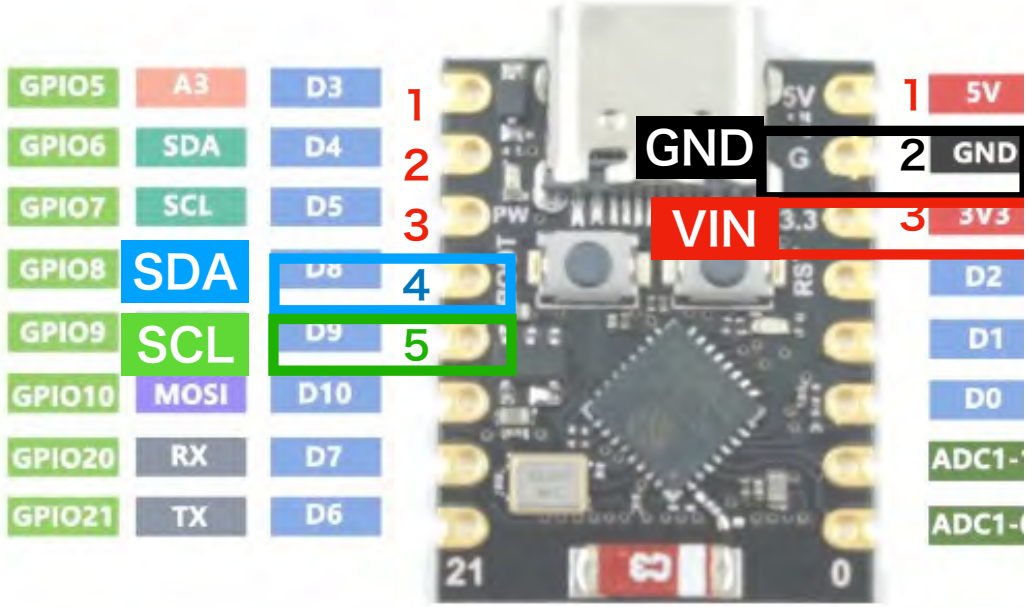
(Extra) BLE Beacon RSSI monitoring on Grafana

Wire connection on bredboard



4 wires for I2C connection





Ex3) BME280 data get

Copy & paste Python code

Thonny - <untitled> @ 17 : 13

<untitled> *

```
2  from time import sleep
3  import BME280
4
5  # BME280 setup
6  SCL = 22
7  SDA = 21
8  i2c = SoftI2C(scl=Pin(SCL), sda=Pin(SDA))
9  bme = BME280.BME280(i2c=i2c)
10
11 while True:
12     temp = bme.temperature
13     hum = bme.humidity
14     press = bme.pressure
15     print(f"Temperature: {temp}°C, Humidity: {hum}%, Pressure: {press}hPa")
16
17     sleep(1)
```

Then push

Run

Shell

```
>>> %Run -c $EDITOR_CONTENT

  MPY: soft reboot
  Traceback (most recent call last):
    File "<stdin>", line 3, in <module>
  ImportError: no module named 'BME280'

>>>
```

You will get error message

no module named 'BME280'

-75

-80

-90

Plotter visualizes series of numbers printed to the Shell.

See Help for details.

Need to install 'BME280.py' libraly

MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ /dev/cu.usb

iotwork.org/smb/bme280

⚠ 保護されていない通信

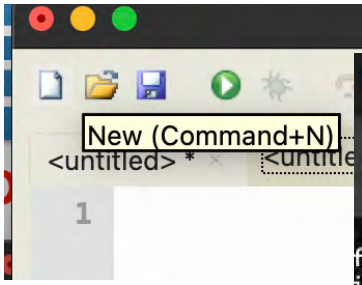
```
from machine import I2C
import time

# BME280 default address.
BME280_I2CADDR = 0x76

# Operating Modes
BME280_OSAMPLE_1 = 1
BME280_OSAMPLE_2 = 2
BME280_OSAMPLE_4 = 3
BME280_OSAMPLE_8 = 4
BME280_OSAMPLE_16 = 5

# BME280 Registers

BME280_REGISTER_DIG_T1 = 0x88  # Trimmir
BME280_REGISTER_DIG_T2 = 0x8A
BME280_REGISTER_DIG_T3 = 0x8C

BME280_REGISTER_DIG_P1 = 0x8E
BME280_REGISTER_DIG_P2 = 0x90
BME280_REGISTER_DIG_P3 = 0x92
BME280_REGISTER_DIG_P4 = 0x94
BME280_REGISTER_DIG_P5 = 0x96
BME280_REGISTER_DIG_P6 = 0x98
BME280_REGISTER_DIG_P7 = 0x9A
BME280_REGISTER_DIG_P8 = 0x9C
```

http://iotwork.org/smb/bme280.txt

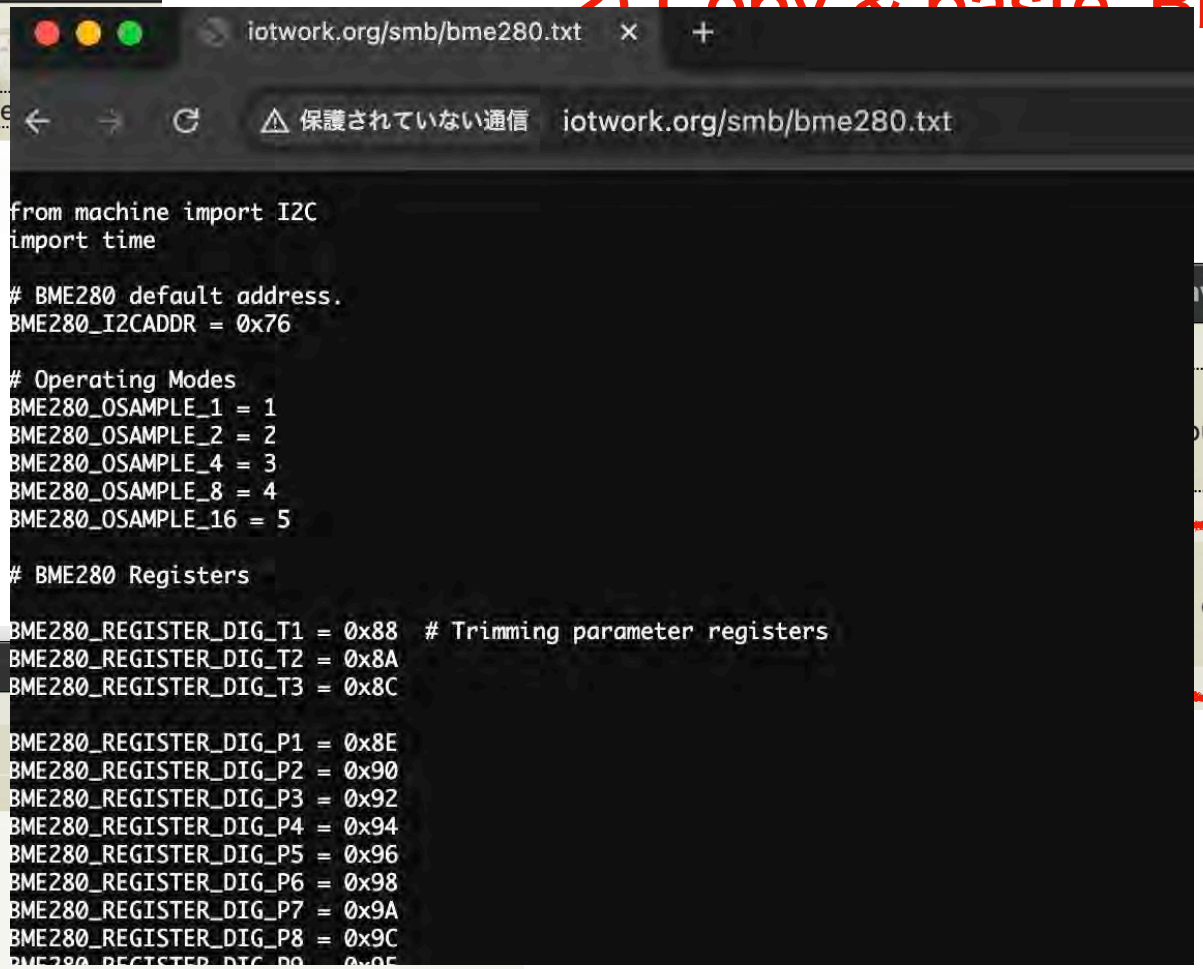
Copy on clip-board

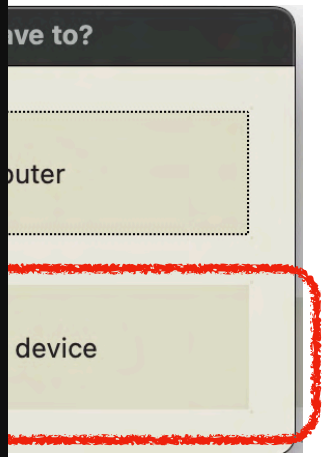
Ex3) BME280 data get

1) Create new editor window

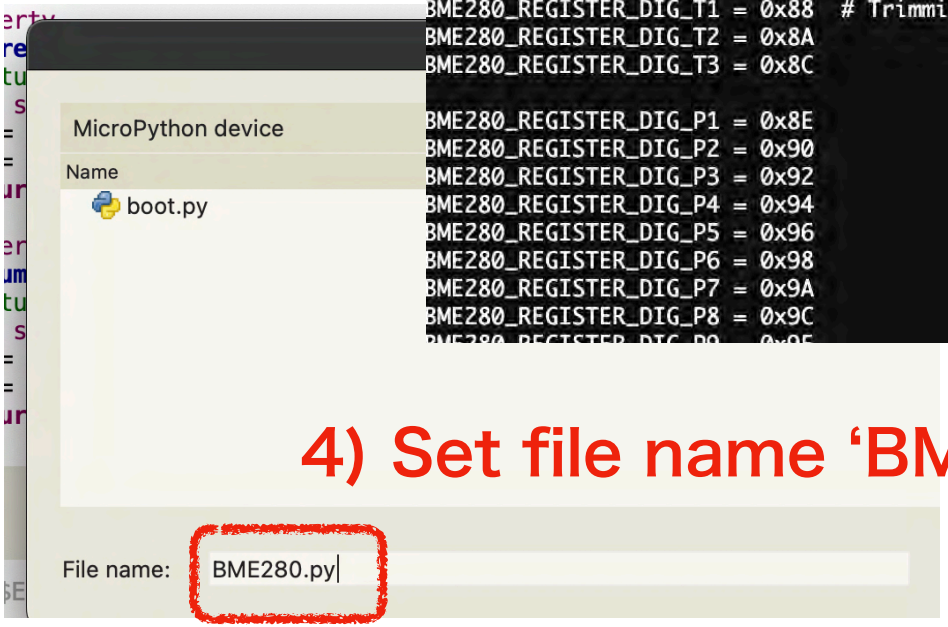


2) Copy & paste 'BME280.py'





it to ESP32

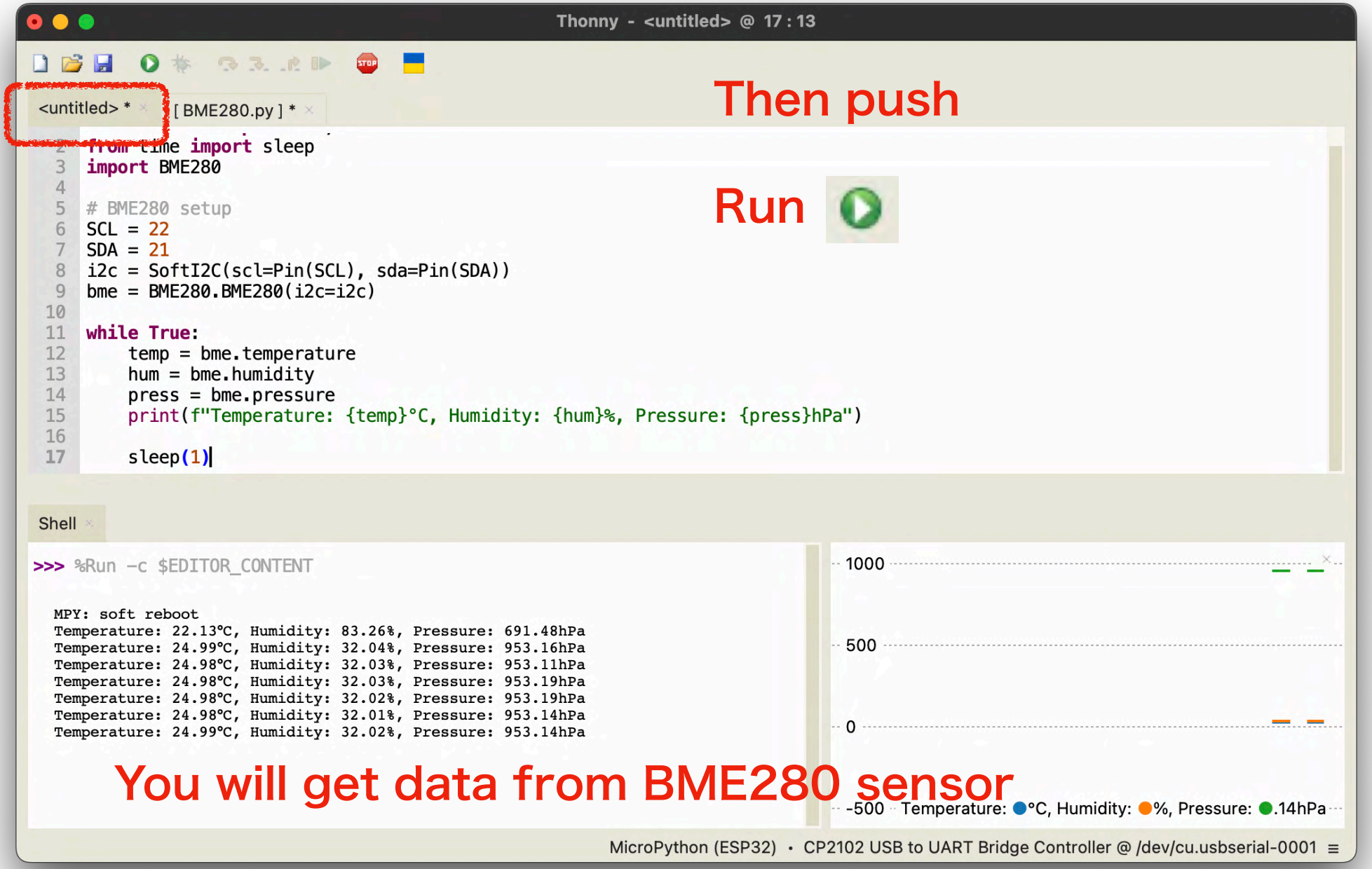


4) Set file name 'BME280.py'

Save > [OK]

Ex3) Get BME280 data

Press initial tab (Untitled..)



Ex4) BME280 data upload to InfluxDB

Copy & paste

Then push

Run

Thonny - <untitled> @ 42 : 19

<untitled> * [BME280.py] *

```
38 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
39
40 def send_to_influxdb(temp, hum, press):
41     measurement = f"ble_rssi_{mac_address}"
42     data = f"{measurement} temp={temp},hum={hum},press={press}"
43     sock.sendto(data.encode(), (INFLUXDB_IP, INFLUXDB_PORT))
44
45 while True:
46     temp = bme.temperature
47     hum = bme.humidity
48     press = bme.pressure
49     print(f"Temperature: {temp}°C, Humidity: {hum}%, Pressure: {press}hPa")
50
51     send_to_influxdb(temp, hum, press)
52
53     sleep(1)
```

Temperature, Humidity, and Pressure

Data will be uploaded to InfluxDB on Cloud (iotwork.org)

Shell

```
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
Waiting for Wi-Fi connection...
```

Via Wifi-LTE(Tethering) connection

SSID: JFWM-WS, Pass: goodlife

1000

750

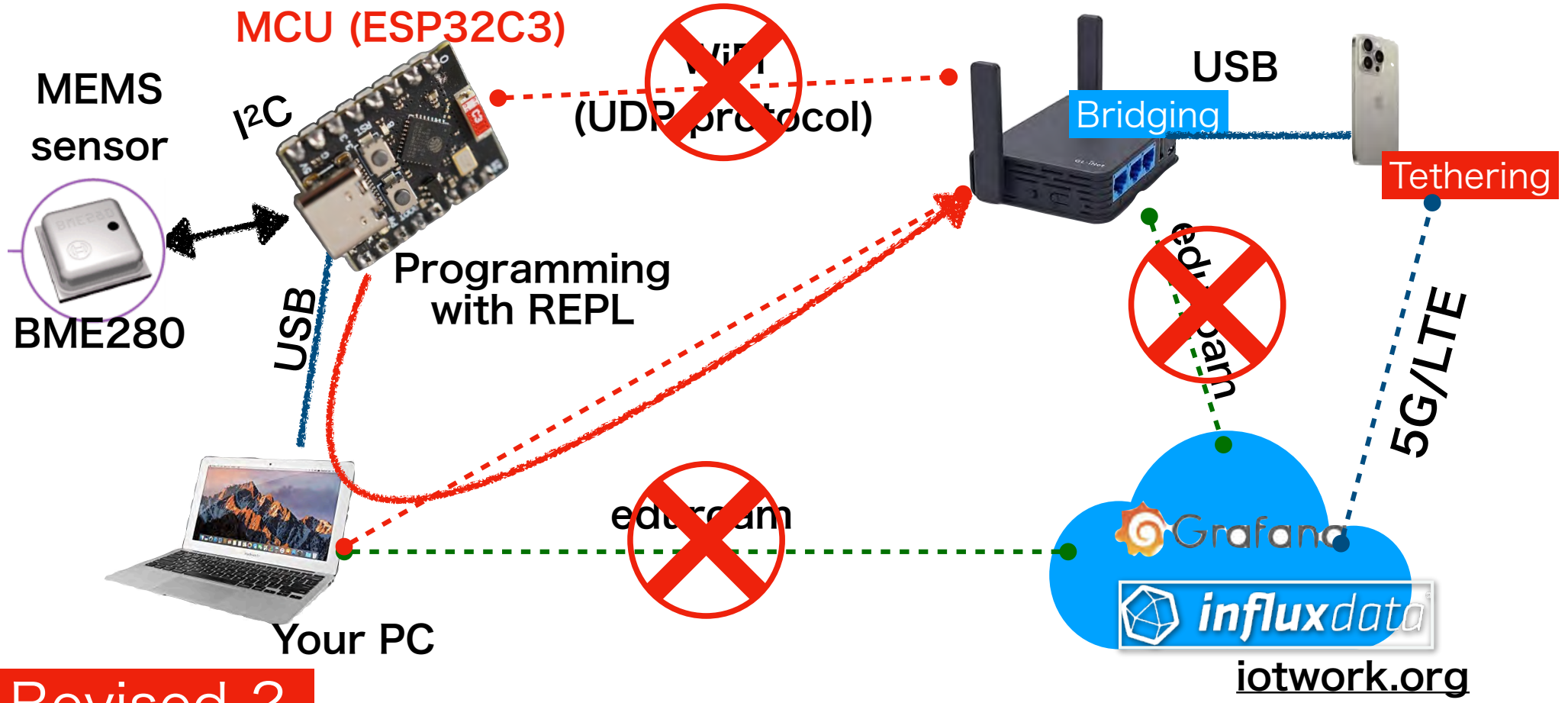
500

Plotter visualizer: Select one or more numbers printed to the Shell.

See Help for details.

Cannot work because of unstable WiFi

Overview of connections (revised)



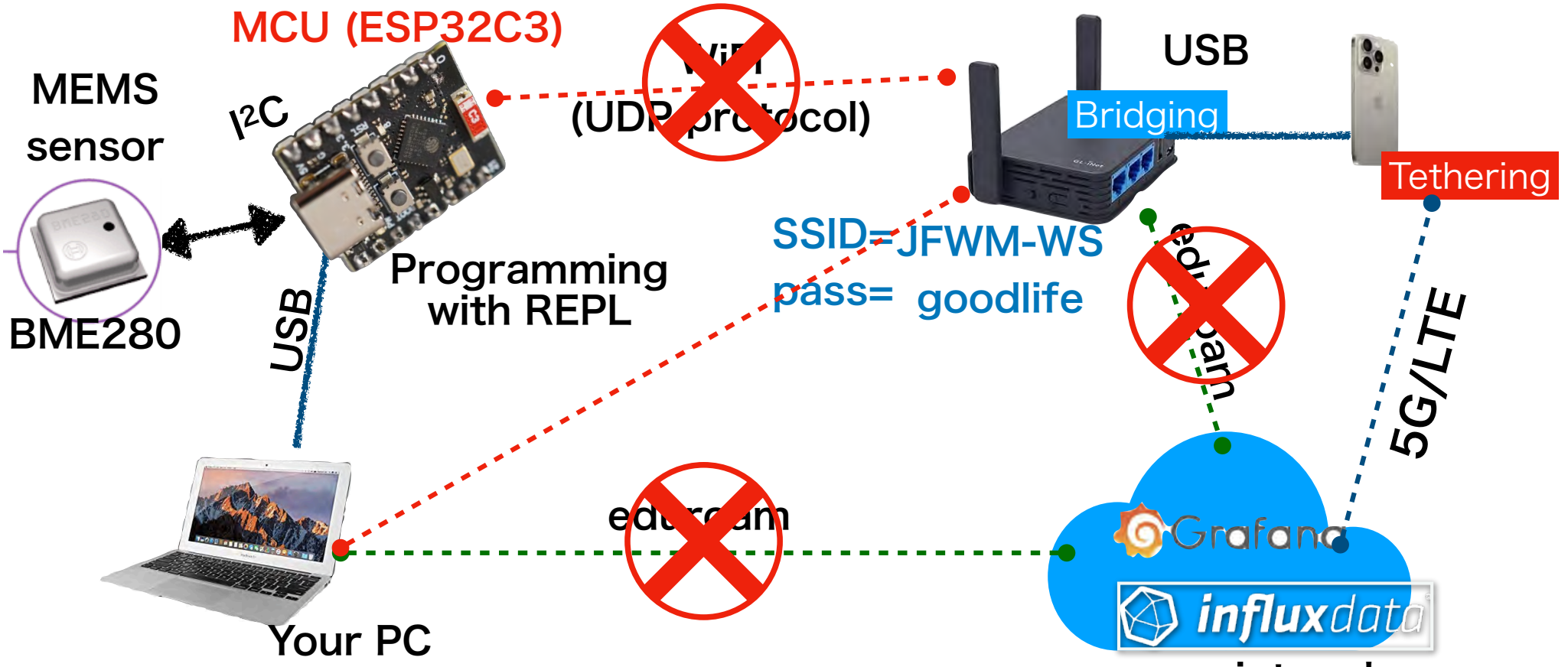
Revised-2

Because of the low quality WiFi of ESP32C3 super mini, connection unstable

ESP32C3 get data from BME280> send to PC via USB

ESP32C3 operate with standalone program [main.py]

Overview of connections (revised)



Revised-2

Because of the low quality WiFi of ESP32C3 super mini, connection unstable

ESP32C3 get data from BME280> send to PC via USB

ESP32C3 operate with standalone program [main.py]

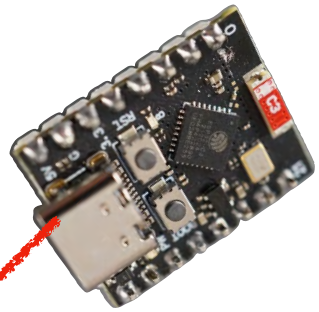
Two python script saved in ESP32

main.py (Boot and run automatic)

main_py-BME280-Serial-ESP32C3.txt

BME280.py

BME280_py-ESP32C3.txt



USB

Serial

UDP send script
on Thonny PC

Wi-Fi

Tethering

5G/LTE

USB

Cloud

Serial-UDP-up-Thonny-local.txt

[main.py] × [BME280.py] × <untitled> * × <untitled> * ×

```
1  # forward_serial_to_udp_simple.py
2  import socket, sys, time
3  import serial
4
5  # ==== Adjust these settings to your environment ====
6  SERIAL_PORT = "/dev/tty.usbmodem1101"       # Windows → "COM7"
7  #SERIAL_PORT = "/dev/ttyACM0"      # Linux
8  #SERIAL_PORT = "/dev/tty.usbmodem11101"  # macOS
9
10 BAUD = 115200
11 UDP_HOST = "iotwork.org"   # InfluxDB server host
```

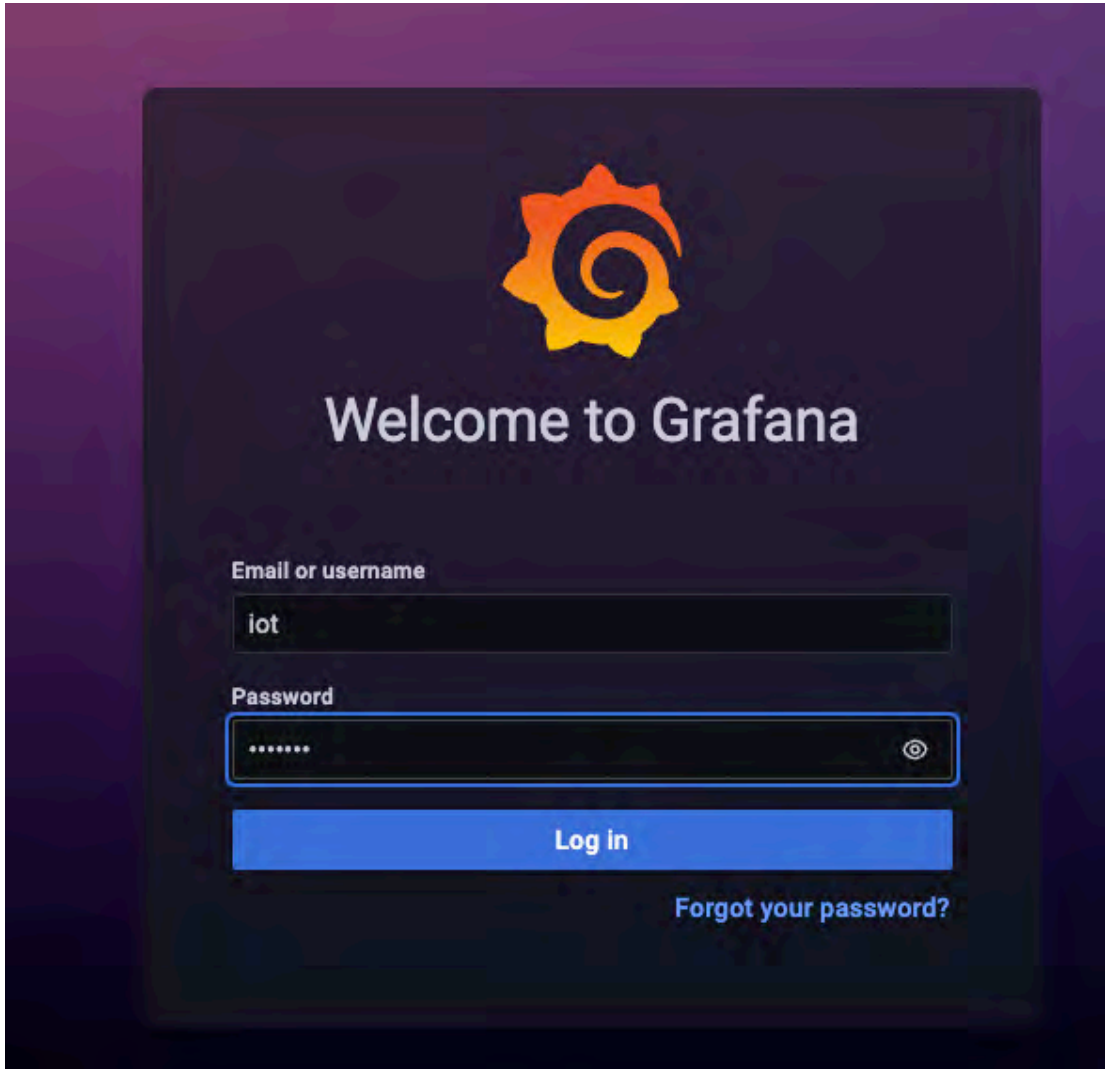
Change Serial port on your PC

influxdata

Data visualization on Grafana

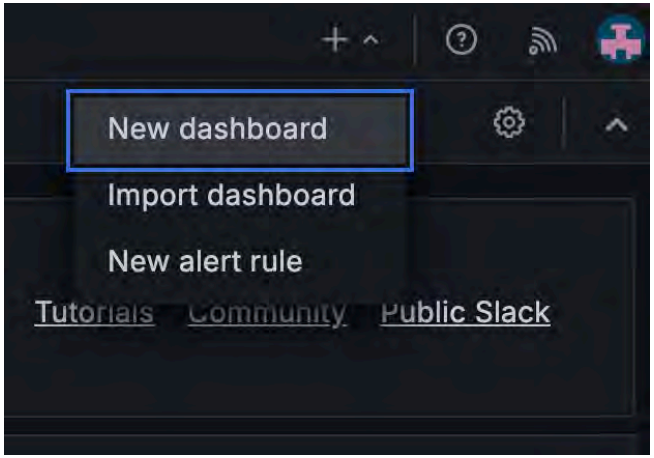
Access grafana server on cloud via AP
(SSID: JFWM-WS, pass: goodlife)

http://iotwork.org:3000/

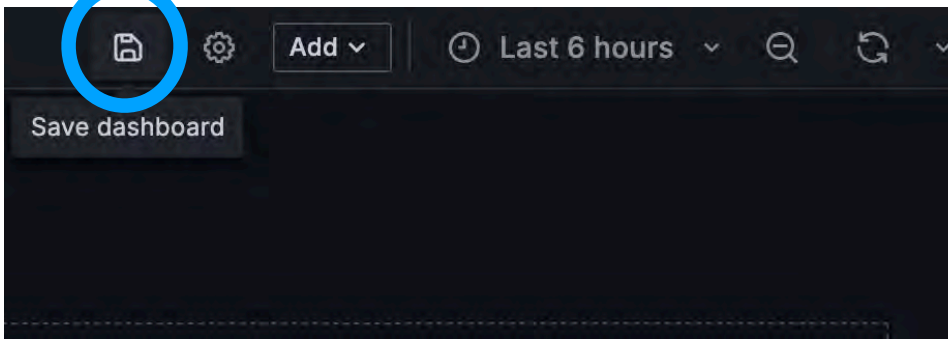


user: iomt
pass: iomt2023

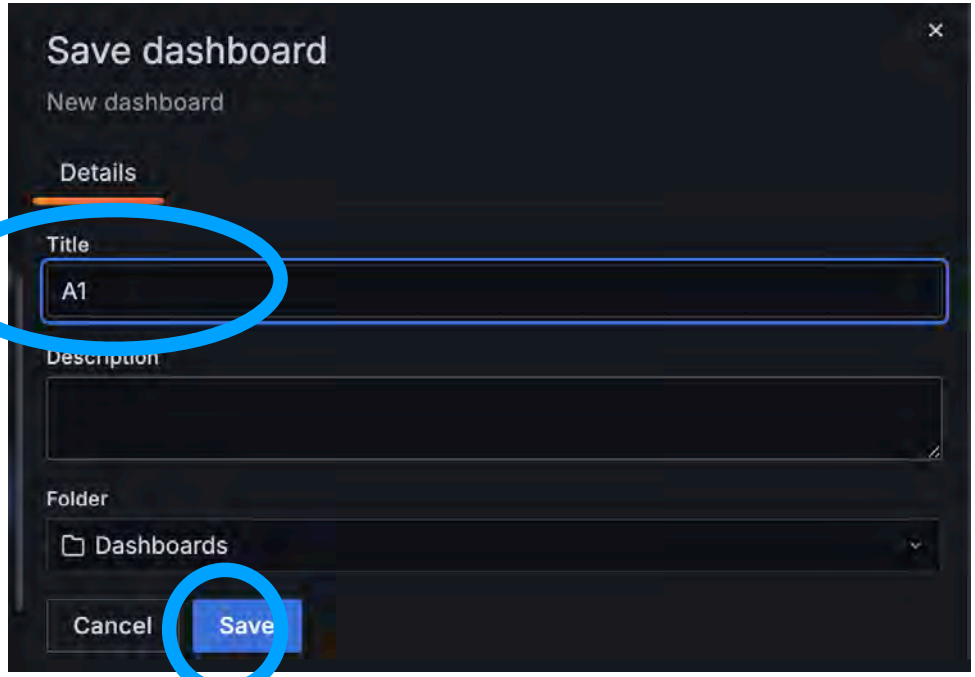
(1) create “Dashboard” from +



(2) Save…

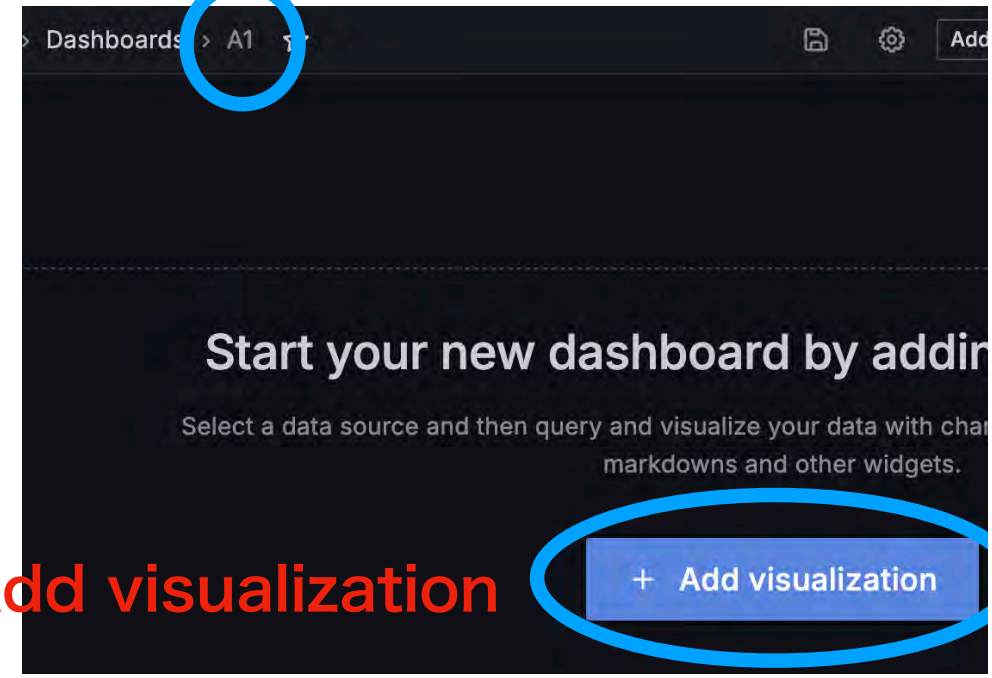


(3) Input title as your group A1a, A1b…C3



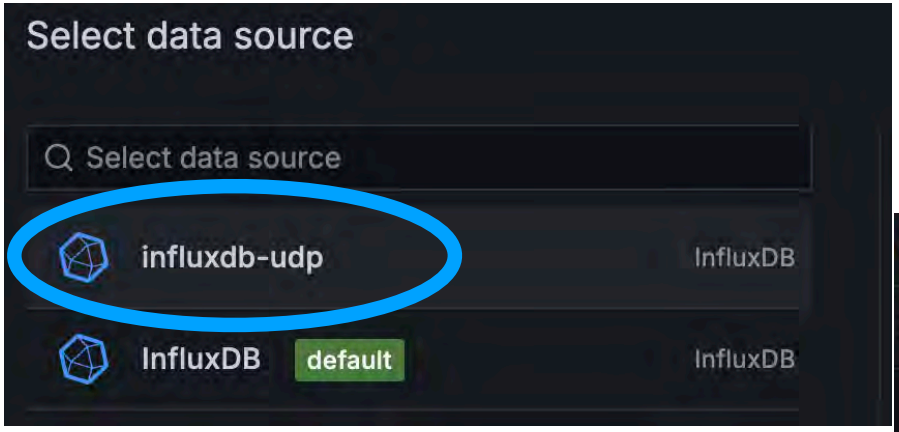
(4) Save

Confirm your group name



(5) Add visualization

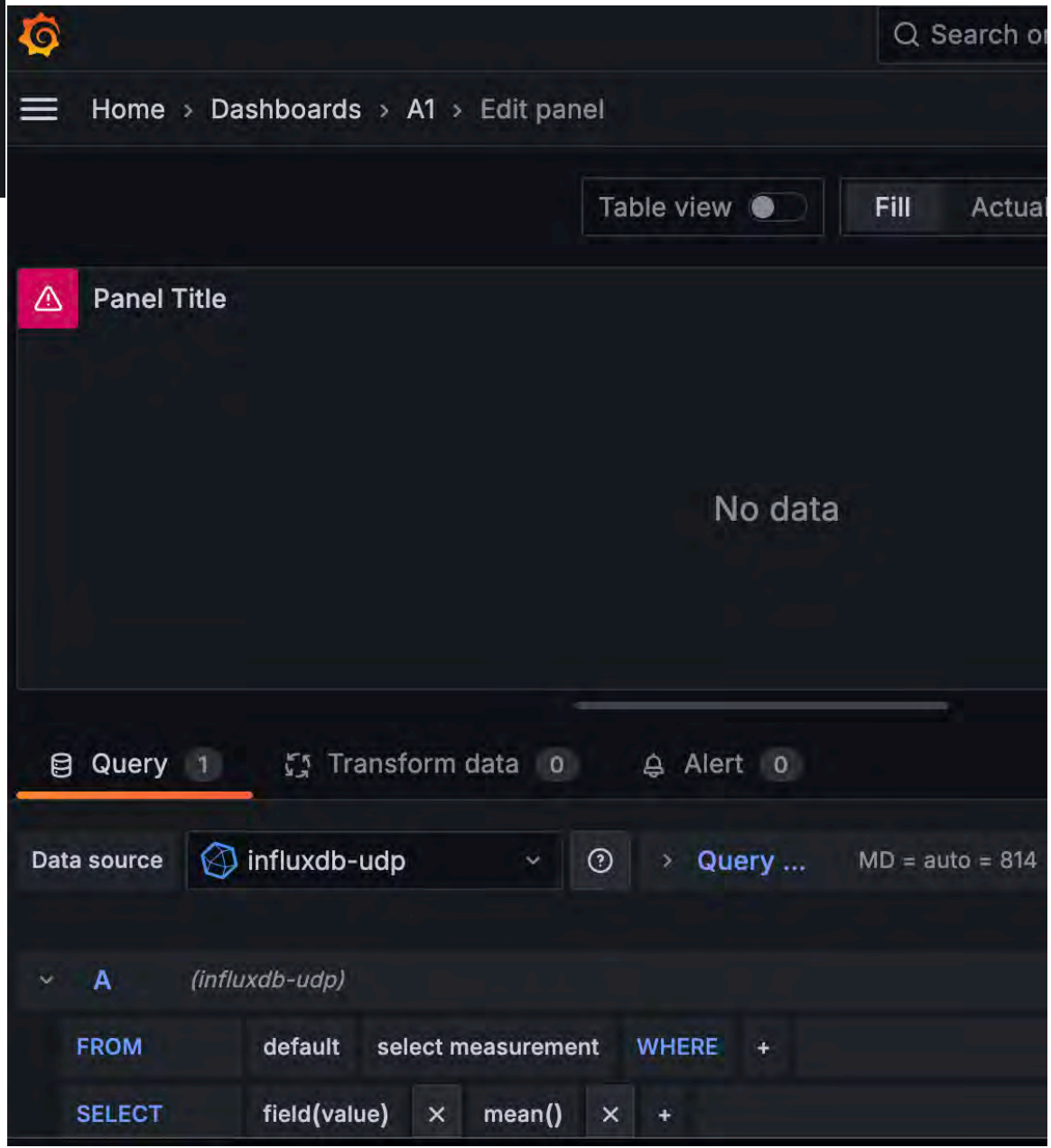
(6) Select data source

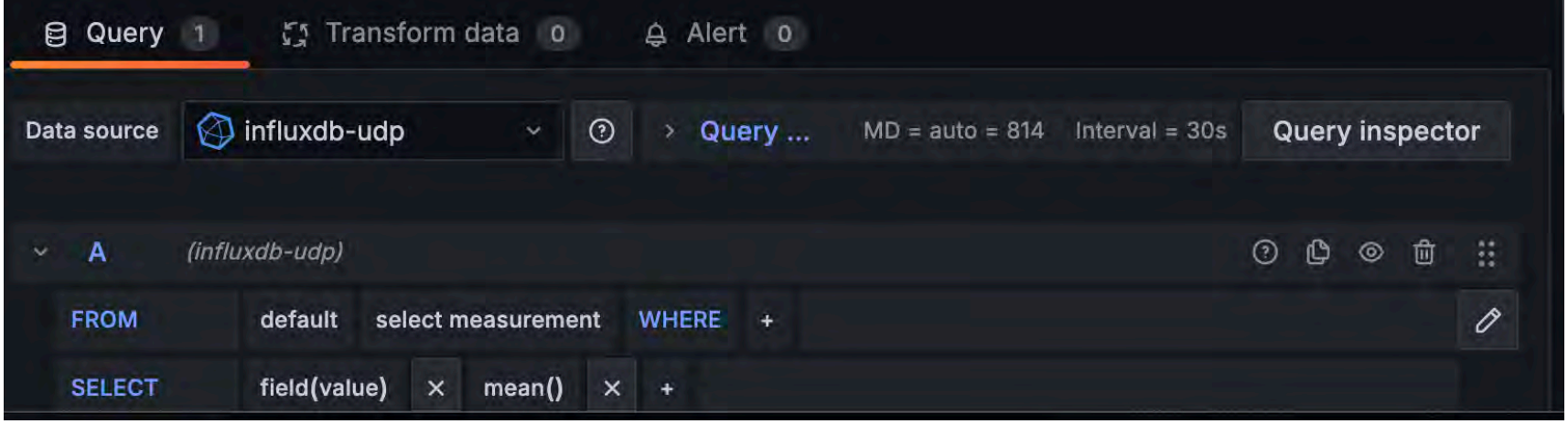


(7) Press ‘influxdb-udp’

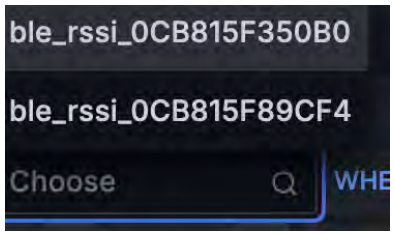
> Edit panel appears

Data from ~~ESP32~~ Thonny@PC is stored in influxdb-udp





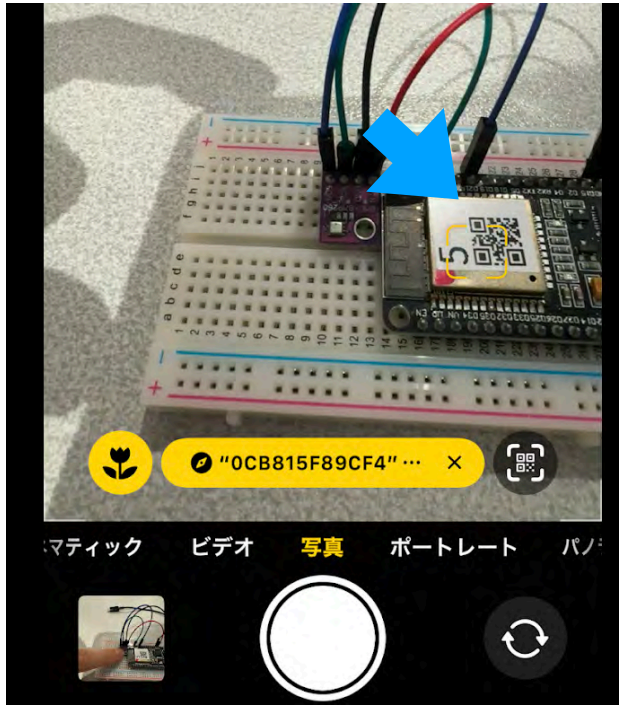
(8) Select measurement



Finding ble_rssi_????????

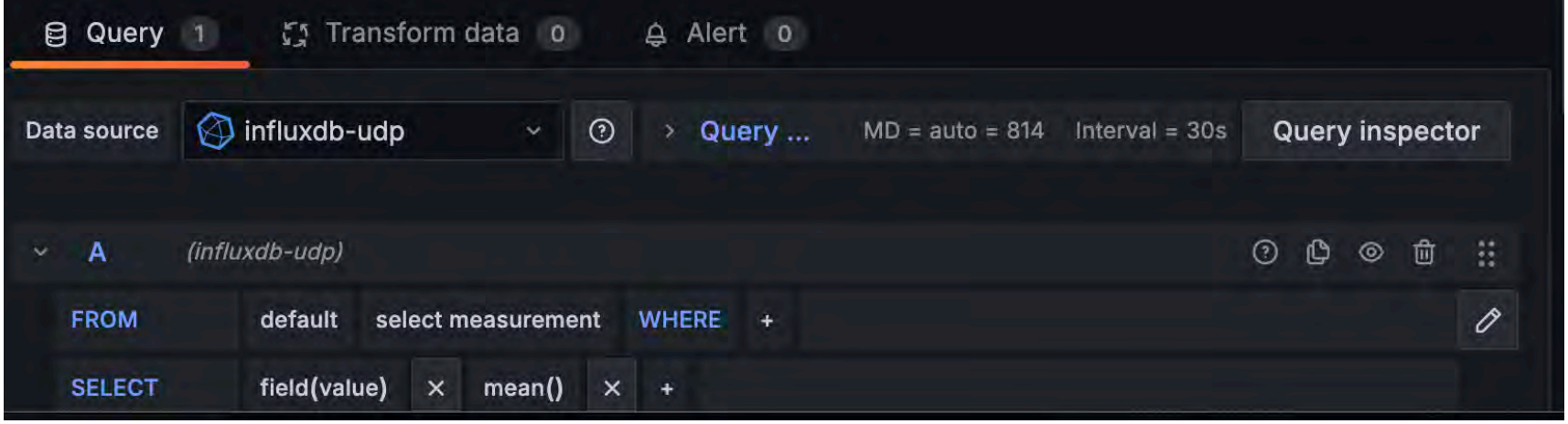
???????? is MAC address of your ESP32

You can get address by QR code reader

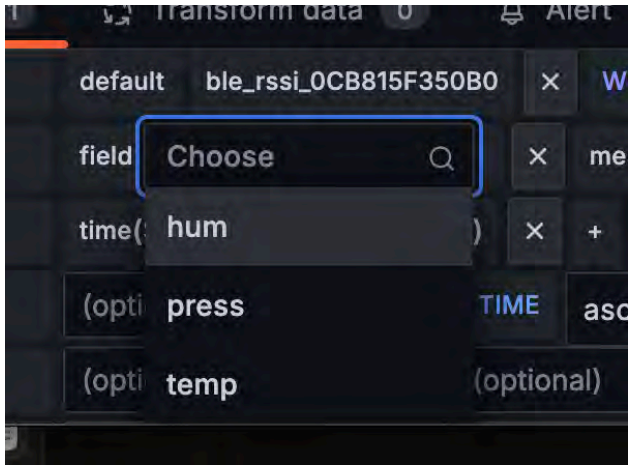


Or labelled number on ESP32 and address

Select your group's address one



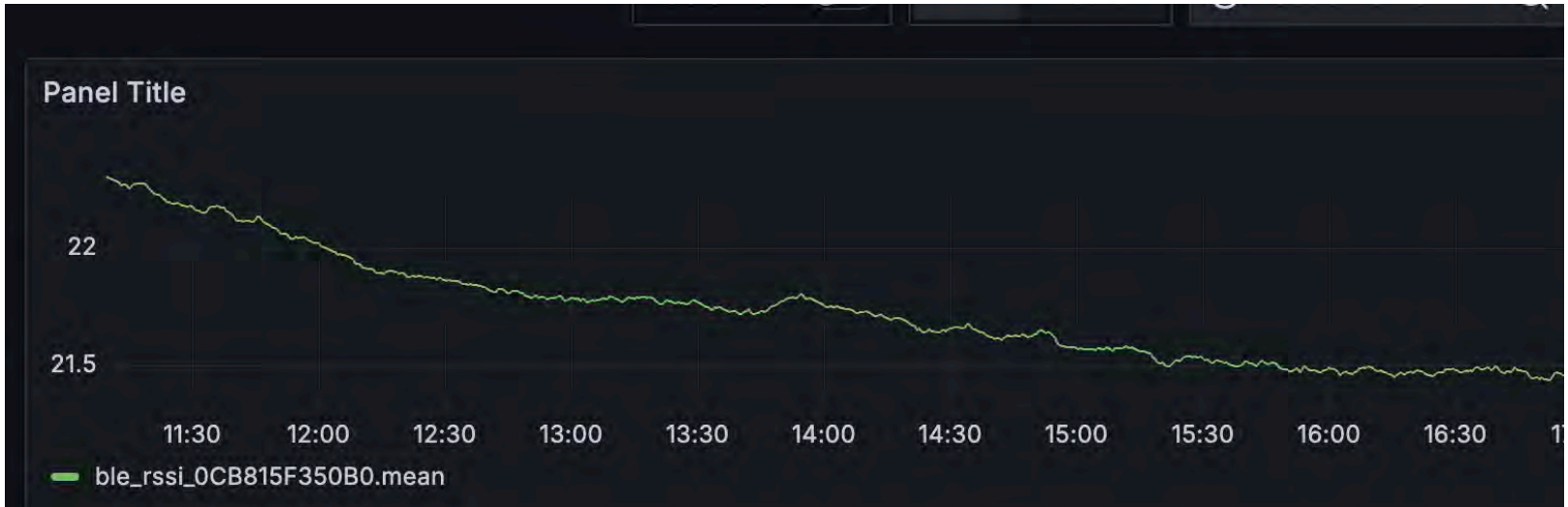
(9) Select field (value)



You can choose
hum: Humidity
press: Pressure
temp: Temperature

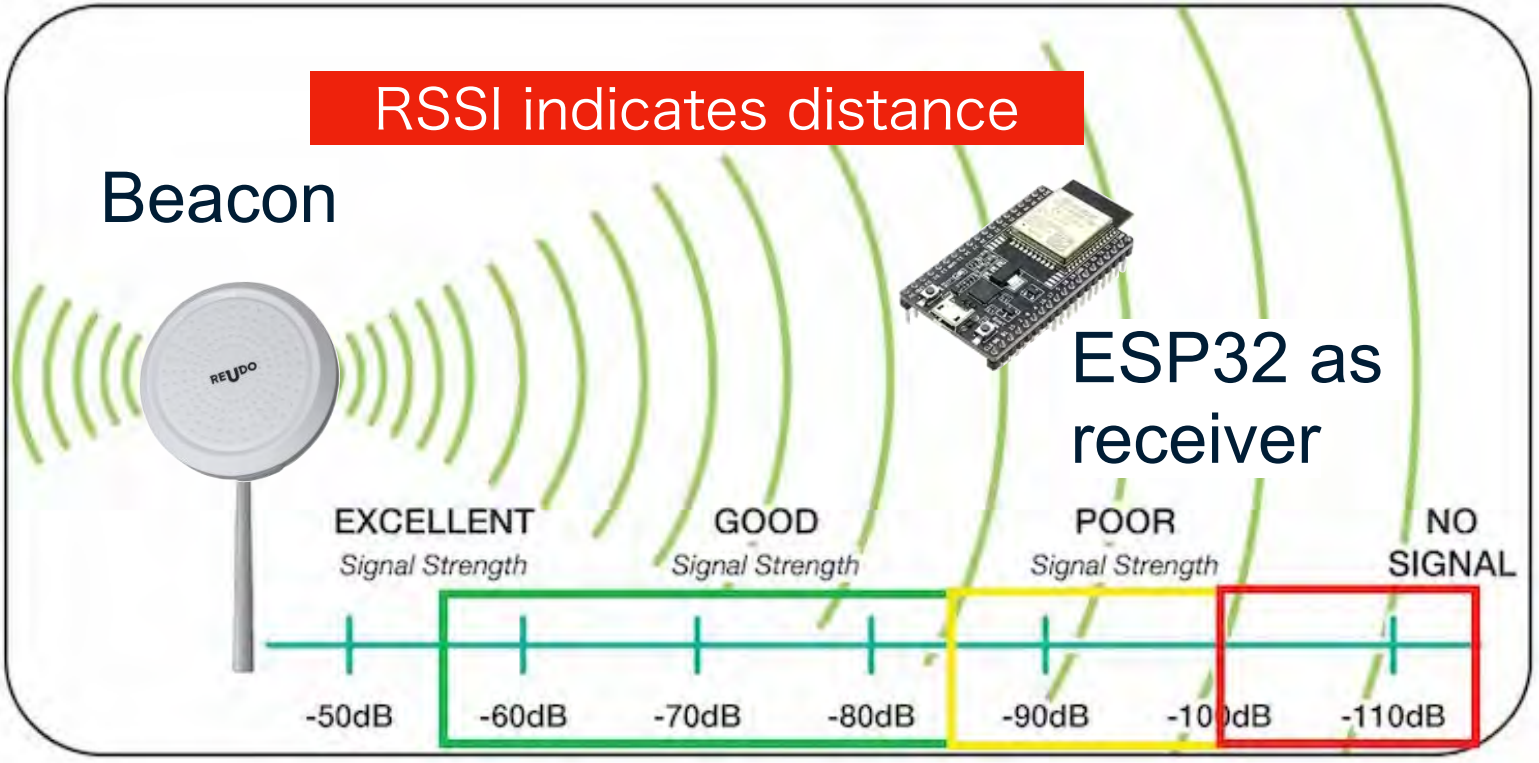
Details will be explained on site

(10)
Finally, get a graph for selection



Opt) Beacon RSSI upload InfluxDB

Bluetooth Low Energy (BLE) beacons are small, battery-powered devices that transmit data using Bluetooth Low Energy.



Before programing

Power on Beacon

Press and hold the button

LED flashes and SW ON

Opt) Beacon RSSI upload InfluxDB

Copy & paste

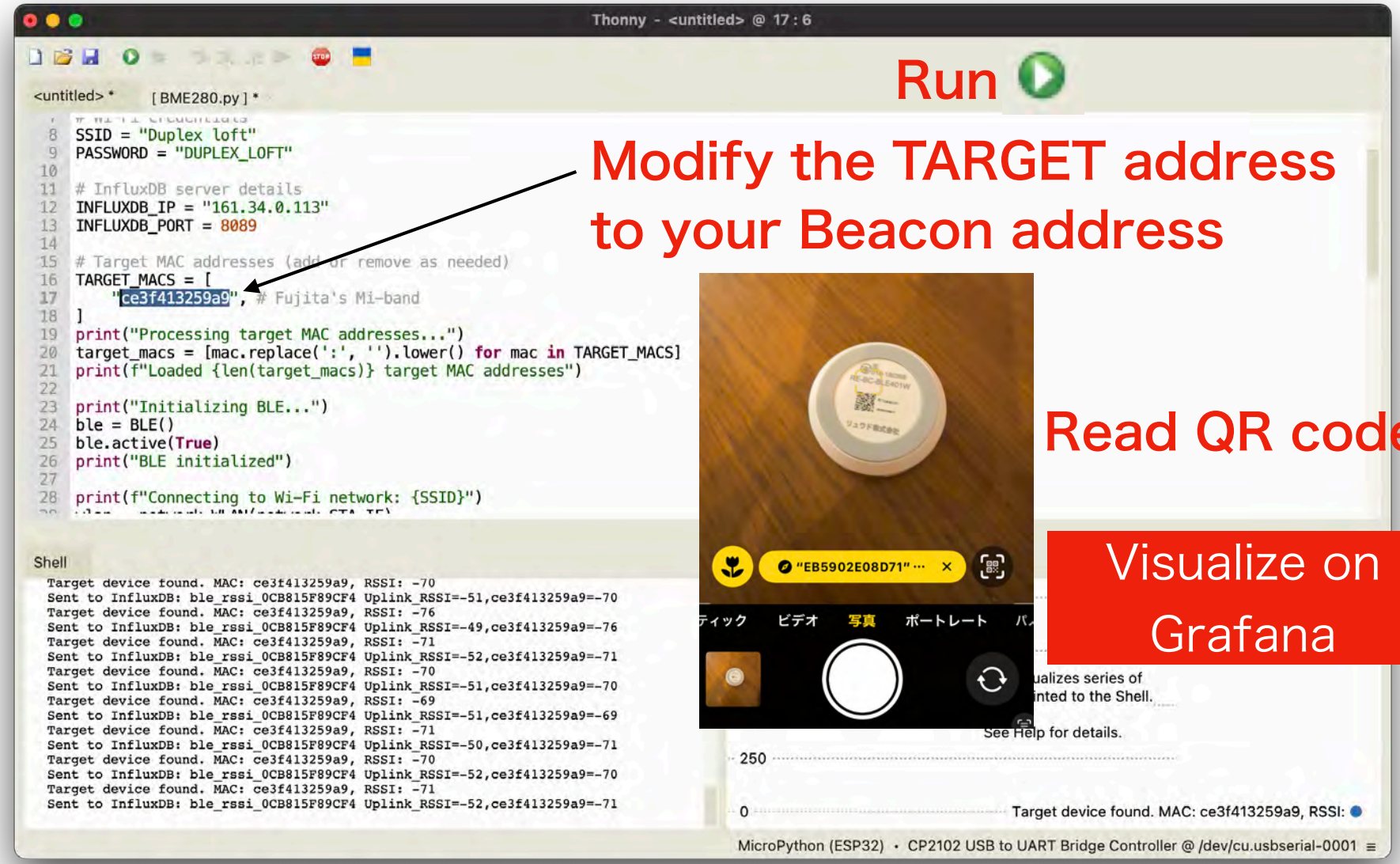
Then push

Run

Modify the TARGET address to your Beacon address

Read QR code

Visualize on Grafana



RSSI values for specific Beacon address

and uplink to AP will be uploaded