



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



公立大学法人
兵庫県立大学
UNIVERSITY OF HYOGO

ホーム English アクセスマップ お問い合わせ

文字サイズ 標準 拡大

受験生の方へ 在学生・教職員の方へ 卒業生の方へ 社会人へ

大学案内 学部・大学院・研究所 入試情報 教育・学生生活 國際交流・留学 研究シーズ・産学連携 地域連携

» 学部・大学院・研究所 > 工学部

Graduate School of Engineering

工学部 school of engineering

工学部公式サイトは[こちら](#)



兵庫県立大学 工学部
柔軟な発想と個性豊かな独創力を
発揮できる人材を養成します

Hyogo prefecture



Himeji castle

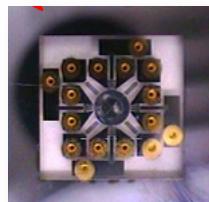


Kobe Beef

%

My research history

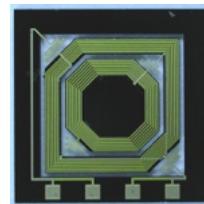
- MEMS physical sensors
 - Gyroscope
 - Accelerometers
- MEMS actuators
 - Mirror device
 - PID control system
- Environmental sensor
 - Multi sensor module for health monitoring



2D gyroscope



Vibratory beam accelerometer



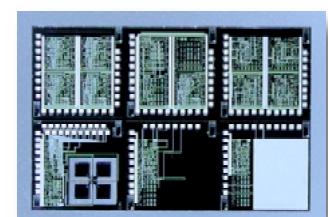
Electromagnetic MEMS mirror



Mirror control system



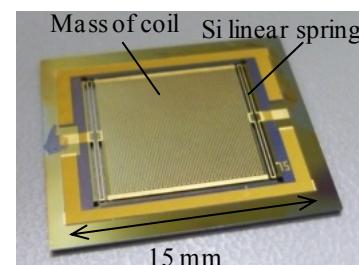
Multi-sensor module



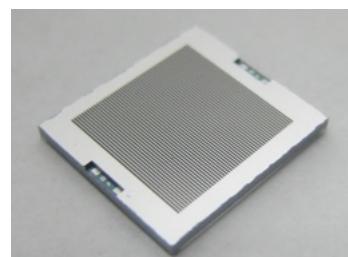
Environmental sensor

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 Medical Engineering
Research Institute,
University of Hyogo**



In the 2nd largest prefectural hospital in Hyogo



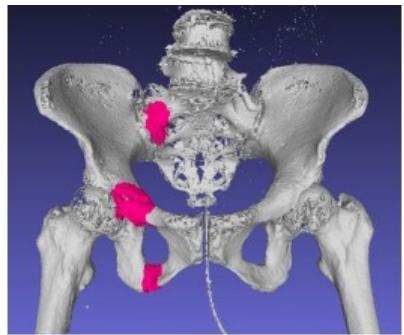
AMERI

Joint research facility

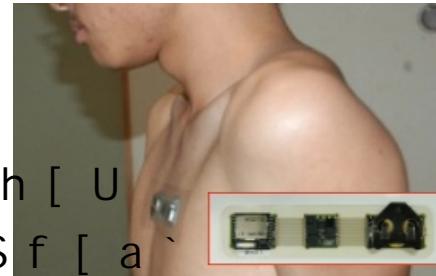
Hospital



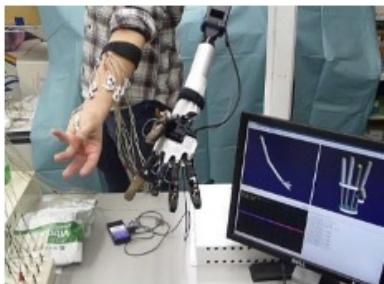
Research Topics of AMERI



Healthcare and rehabilitation Eng.



3 ; Ž T S e WV b W^ h [U
X d S U f g d W ^ a U S f [a`
V Wf Wd _ [` S f [a` : WS ^ f Z a T e Wd h S f [a
S V Z We [h W b ^ S e f Wd ?
? 7 ? E e W` e a d e



8 [` Y Wd \ a [` f
We f [_ S f [a ` da T a f [1st
b d a e f Z Wf [U z S ` v 4th



3 ; Ž T S e WV S ` Wg d k
h [e g S ^ [I S f [a `

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)

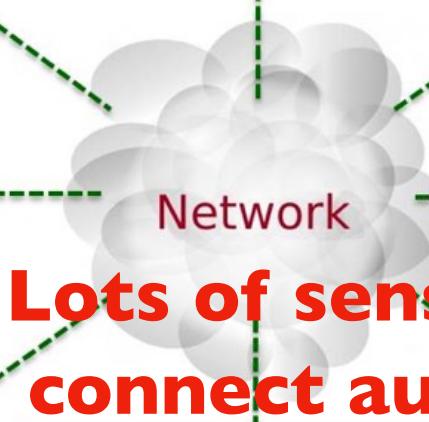


; a F

Internet connection for everything not limited to Human operated device

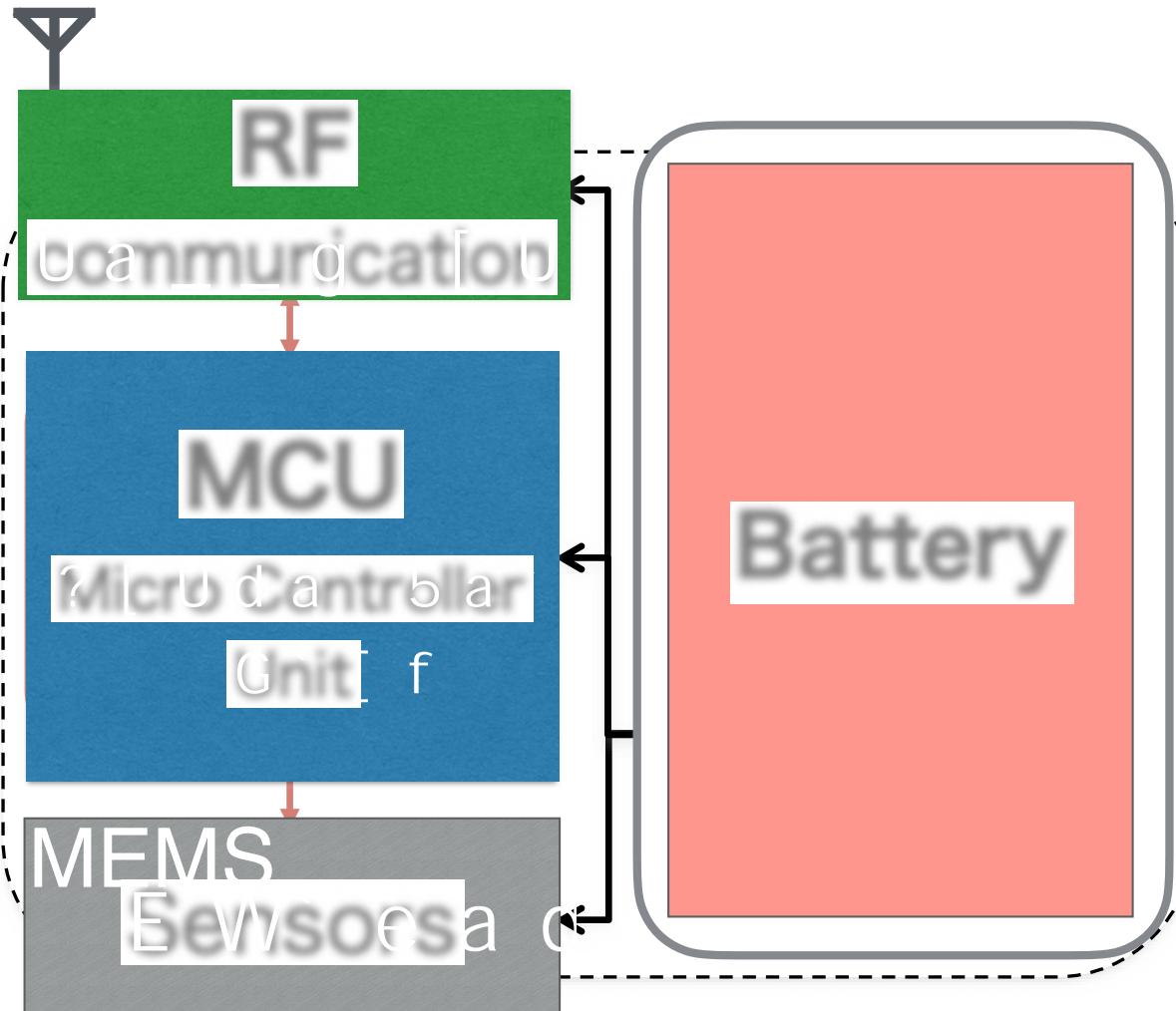
Human uses the Internet

Smartphone,
Tablet,
PC



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 Circuitry (IC)**

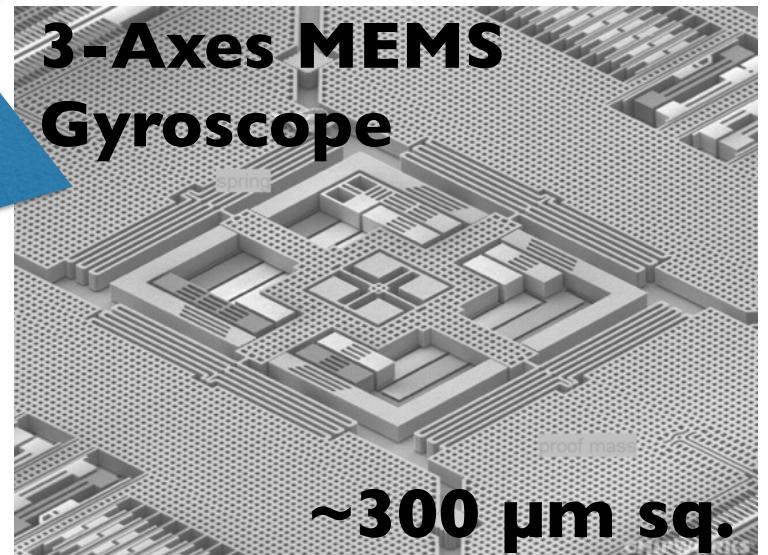
- IMU combo
- Magnetometer
- MEMS microphones
- Pressure sensor
- Humidity + Temperature sensor
- BAW filters and duplexers
- Antenna tuner



**Full of MEMS
sensors in smart
phone**

at least 5 MEMS sensors

Inside?



History of MEMS application



Automotive

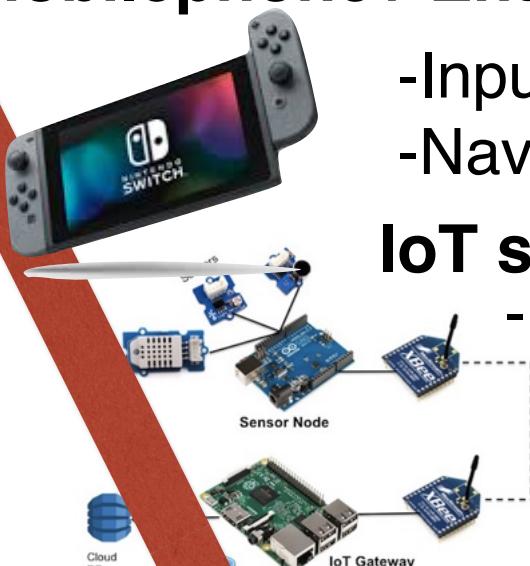
%' kde



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

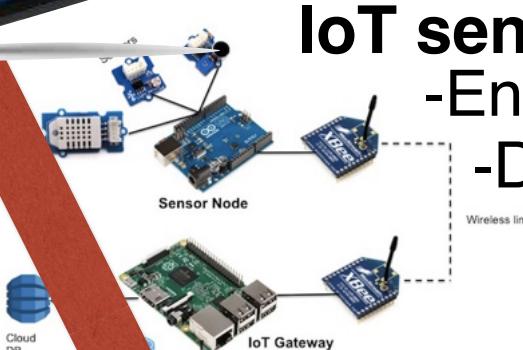


Mobilephone / Entertainment



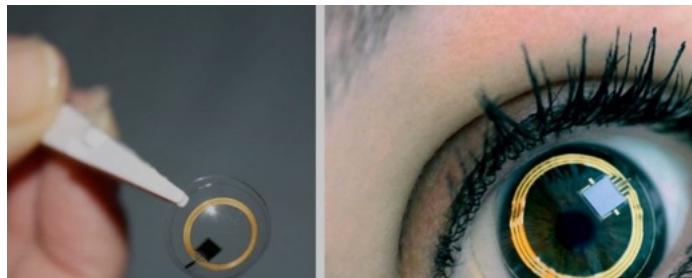
IoT sensor node

- Environmental sensing
- Drones



Health

- Medical devices



Performance
Size & Cost
↓ Number ↑

MEMS application > Navigation



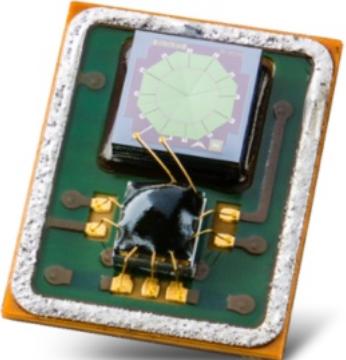
Tracking: Cycling around the Lac d'Annecy

Position tracking by combined sensor info.

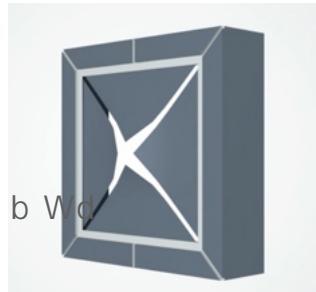
- **GPS** 
- Cell
- WiFi
- Geomagnetic
- **Gyroscope**
 - **Angular velocity ∫ Angle**
- **Accelerometer**
 - **Acceleration ∫ Velocity ∫ Position**



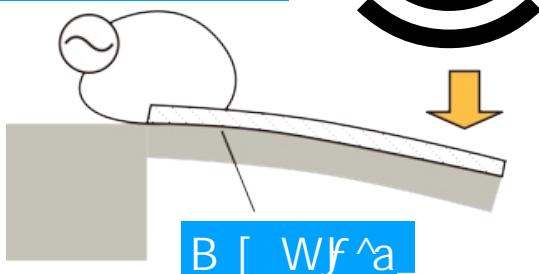
MEMS device 1: Microphone



D [U W Y d S [` e [I W
5 S ` f [^ Wh Wd i [f Z
_ S f Wd [S ^ h [T d S f



A g f b g f h



B [Wf ^a_

? 7 ? E _ S] We U a _ b S U f t Z [
b Wd X a d _ S ` U W _ [U d a b Z a `
[` Wj b W` e [h W

iPhone 11



% &

2 Ha [e f S ? W'

E a g ` v
i S h W

[B Z a ` W # # Z S e & _
% & X a d _ S [` _ [U ž
A f Z Wd e S d W
` a [e W U S ` U W ^

MEMS device 2: Apple watch etc.

Activity tracking for health care



Various devices of
“healthy gadgets”



Similar technology

RF, MCU, battery
&

MEMS sensors
Acc. Gyro,

Optical sensor
For SPO₂, heart rate

- Heart rate sensor
- Accelerometer
- Gyroscope
- Bluetooth 4.0 LE



Inside of AppleWatch

/ 5 fi F W U Z ; ` e [Y Z f e

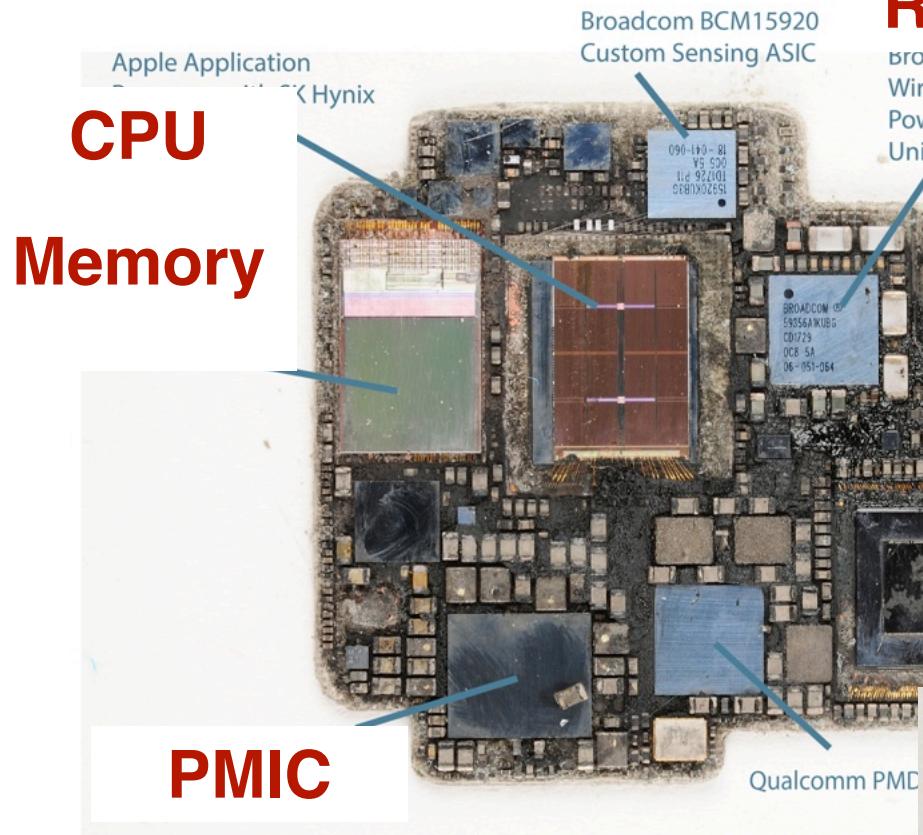
Battery

LCD



PCB

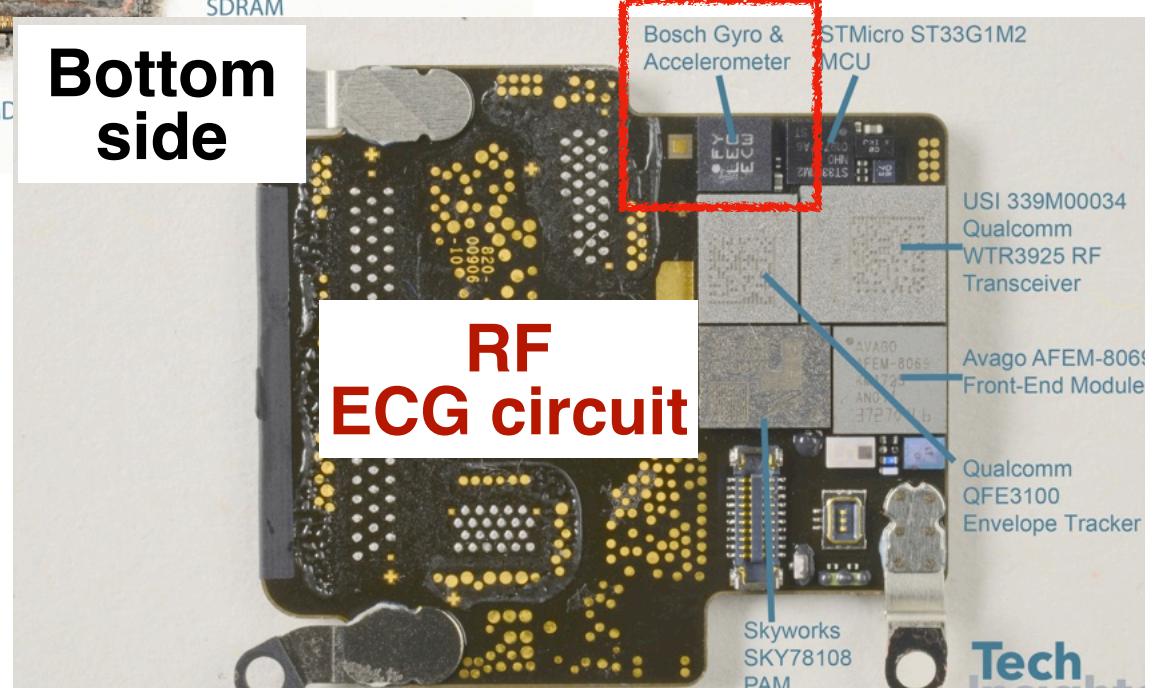
Tech Insights



Top side

Bottom side

AppleWatch 3rd gen.

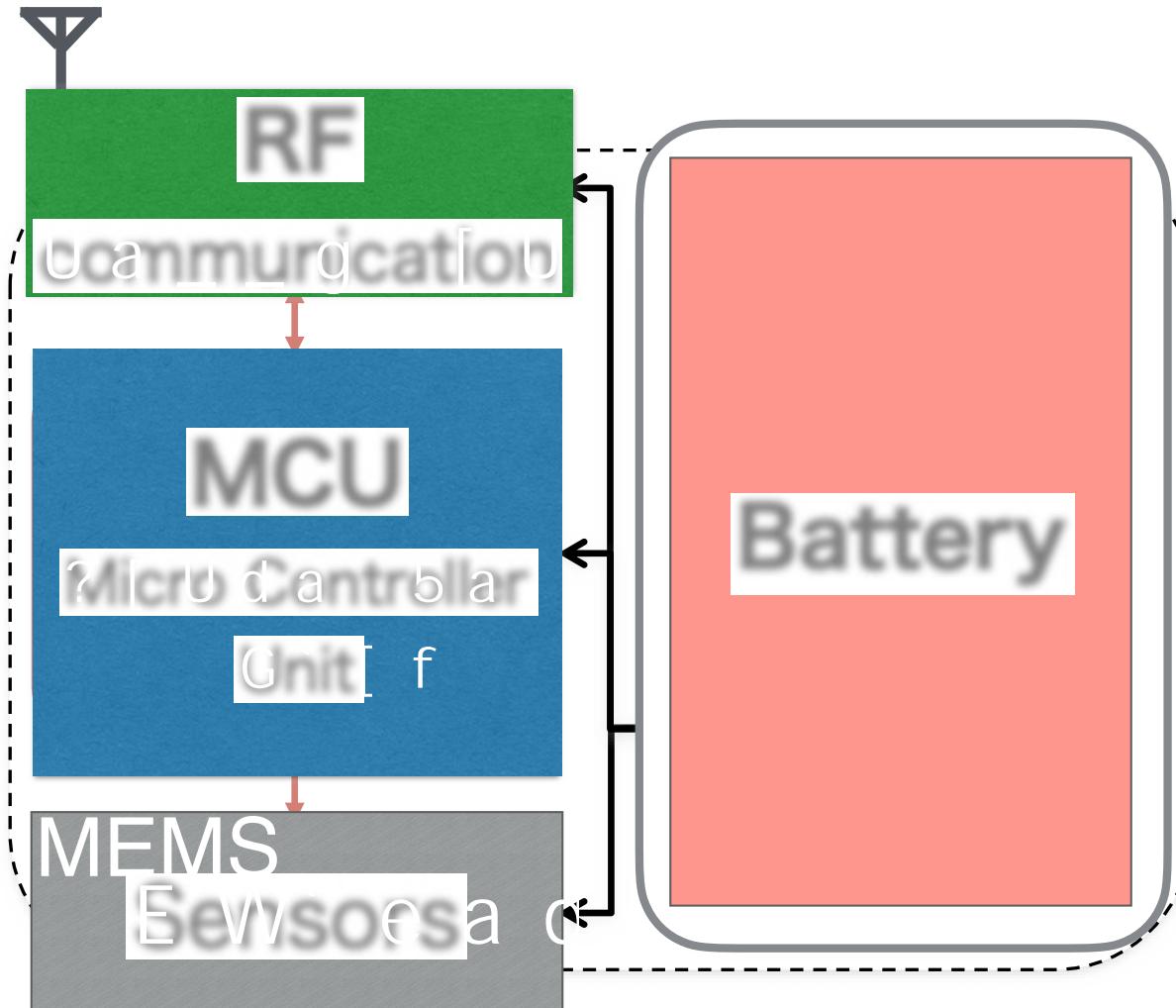


RF
ECG circuit

Tech

IoT node architecture

Typical architecture of IoT node



Architecture of a sensor node

NEXT

Explains the internal details of IoT applications in the market

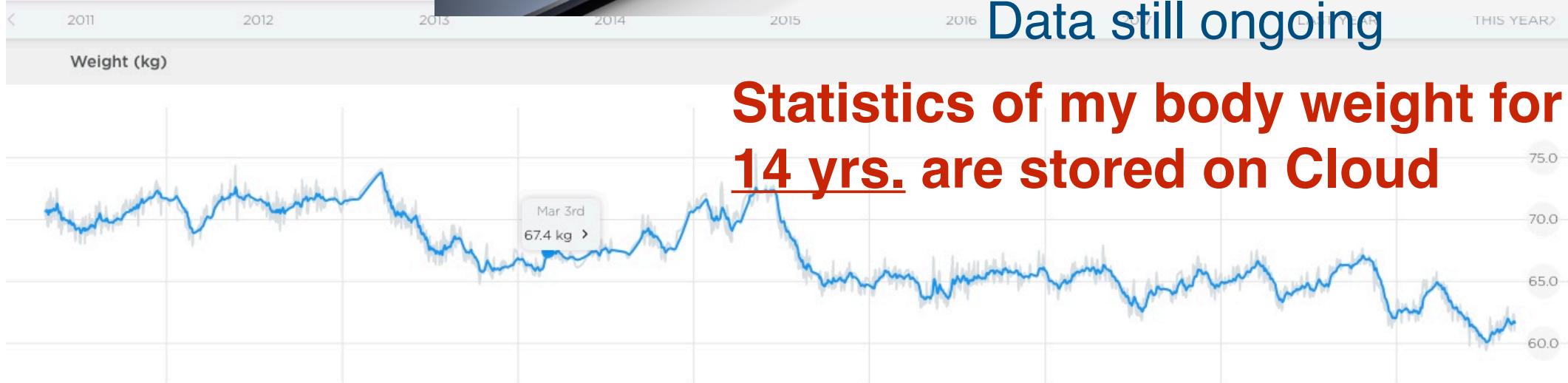
IoT on market : IoT Body scale

¥20,000
(USD 150)
worth the price

Withings

Made in France

← Weight



Smart Body Analyzer

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

US\$ 149.95



Pinpoint your weight and body composition

Get high-accuracy weight and body fat measurement as well as your Body Mass Index. Pick the right body type to optimize the fat mass measurement – if you exercise vigorously on a regular basis, switch to "Athlete mode".



Weight



BMI



Fat Mass

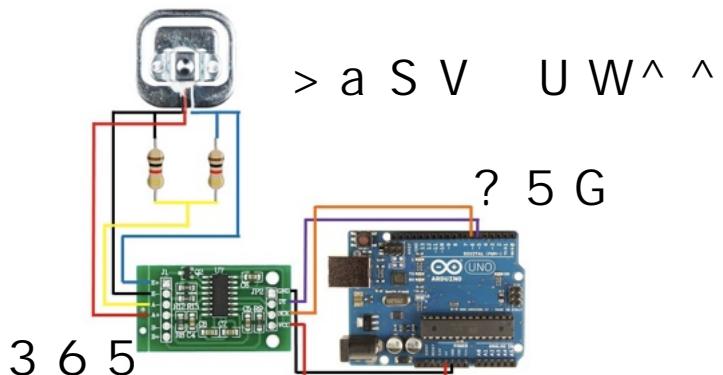


Heart rate

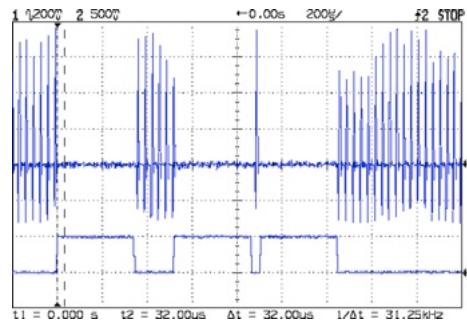
Inside of Withings IoT scale



4 a V k i W_ YWSfe g d WV[f Z [` Ye e Wd h Wd
T k ^ a S V Ž U W^ ^ / e twi+things



4 a V k X S T k . [_ b WV S ` U W _ WS e d



my account

Don't have an account yet?

mail address

メールアドレス

password

パスワード

I forgot the password?



W` f
← Weight

I [8 [9 W` Wd S f W
Y d S b Z

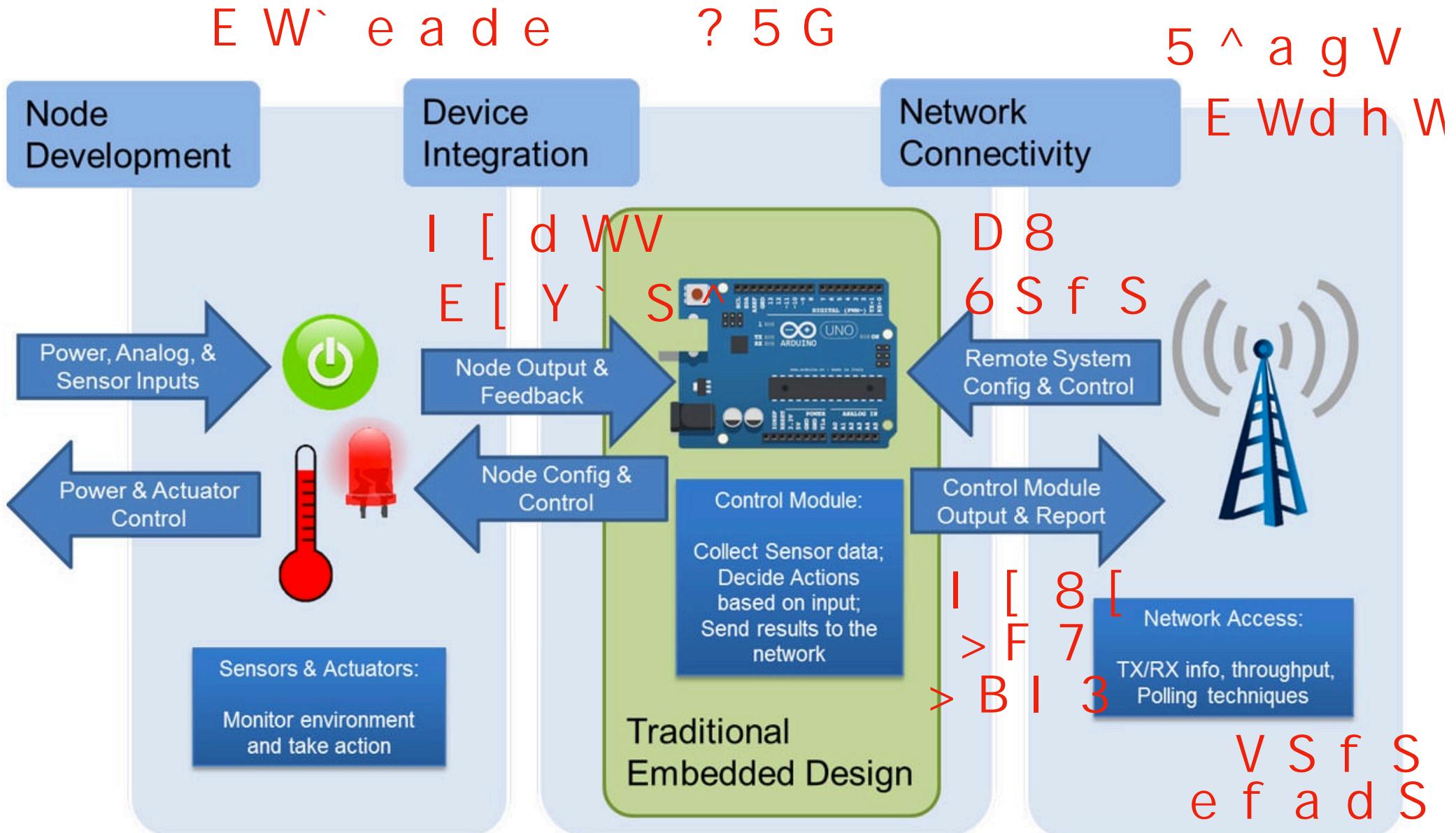
a ` U ^ a g V



5 a ` ` WU f k [A S b d W Ž Y u g d WV E E ; 6 ! b S e
0 g b ^ a S V V S f S f a I [f Z [` Ye U

Fat mass (%)

Typical configuration IoT



NEXT

; ` f d a V g U f [a ` f a f z w z [e f a d k
g e [` Y f W_ b Wd S f g d W s e s ` Wj S

Temperature measurement (primitive method)

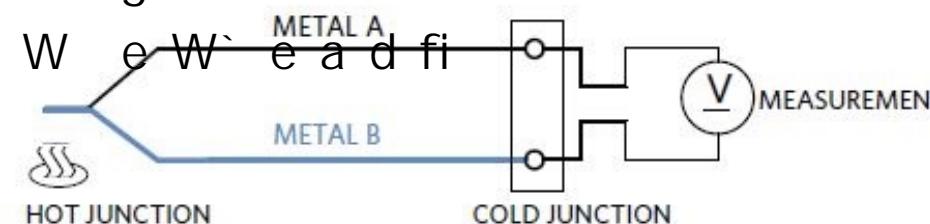
Wj ž 9 ^ S e e f g T W f Z Wd _ a _ Wf S Wd W k a g Wh Wd e W W

B Z k e [U S ^ U Z S ` Y W
O F W_ b Wd S f g d W



B Z k e [U S ^ U Z S ` Y
O f Z Wd W S b S ` e a X ` S ^ U
@ a W ^ W U f d [U [f k
H [e g S ^ U Z W U] a ` ^ k

Wj ž F Z Wd _ a U a g b ^ W
/ f W_ b Wd S f g d W e W e a d fi



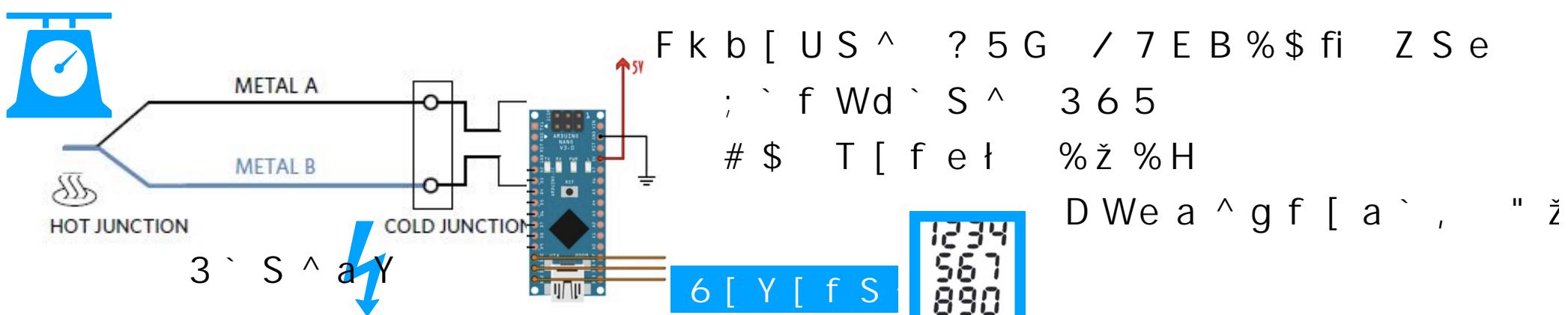
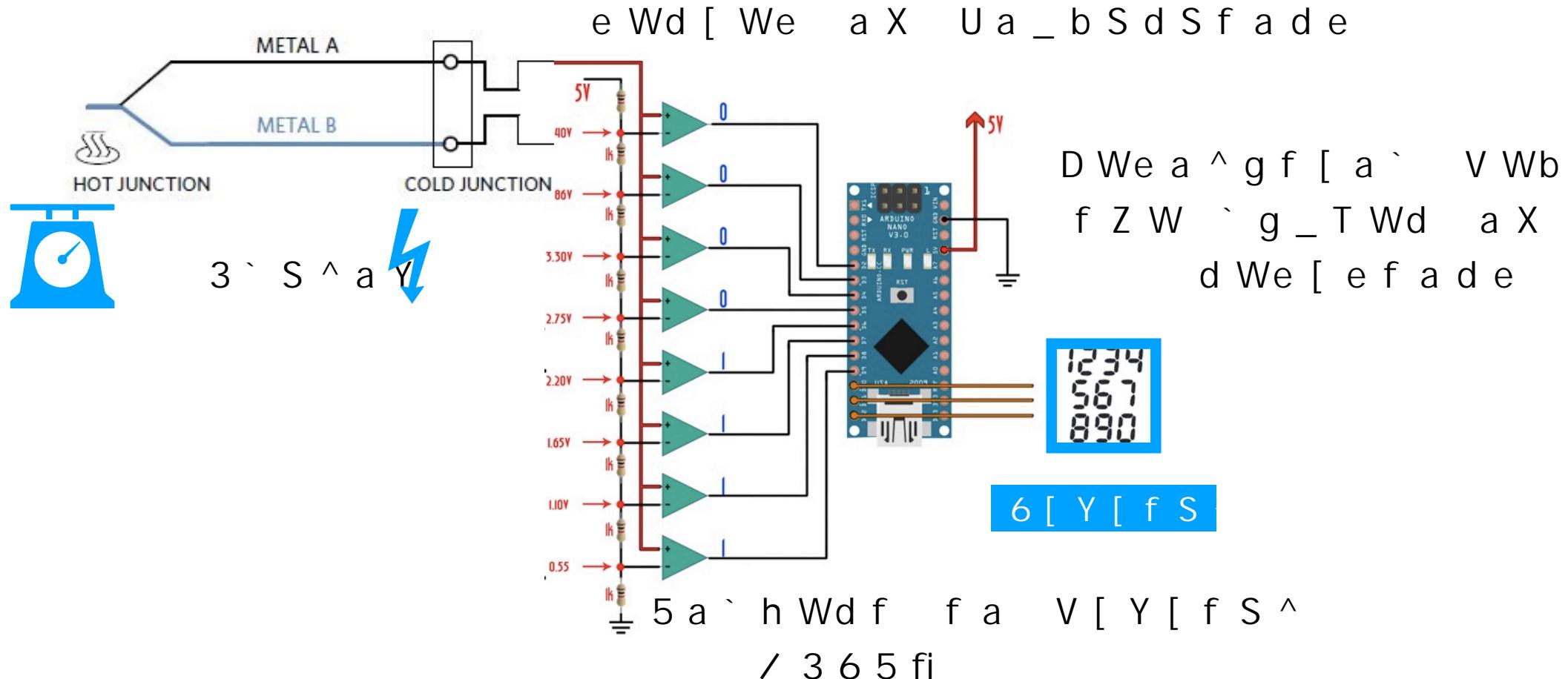
3 ` S ^ a Y

B Z k e [U S ^ U Z S ` Y W 7 ^ W U f d [U S ^
O F W_ b Wd S f g d W 0 7 ^ W U f d a _ a f [h W



3 ` S ^ a Y h S ^ g W e f U S g d ^ W f [a X U a ` ` W U f ?

Temperature measurement (w/ ADC)



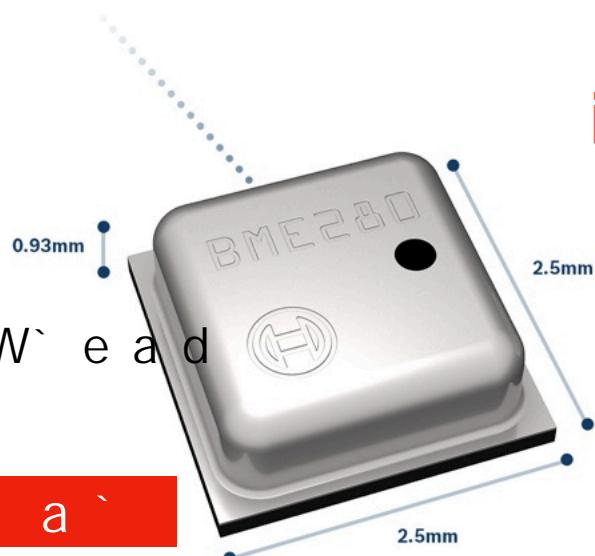
Recent digital output MEMS sensor



BOSCH

Invented for life

4 ? 7 \$ * "
U a _ T a e W` e a d



G e W f Z [e a `

Main features



Relative Humidity
Measures relative humidity
with a fast response time



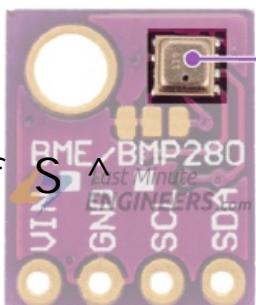
Pressure
Measures barometric
pressure and altitude



Temperature
Measures ambient
temperature



6 [Y [f



BME280 Chip

? 5 G

E W` e a d

8 g ^ ^ k V [Y [f S ^
i [f Z U a _ b W` e S
b S d S _ Wf Wd e
E [` Y ^ W U Z [b
Ž F W_ b Wd S f g d '
Ž : g _ [V [f k
Ž B d We e g d W

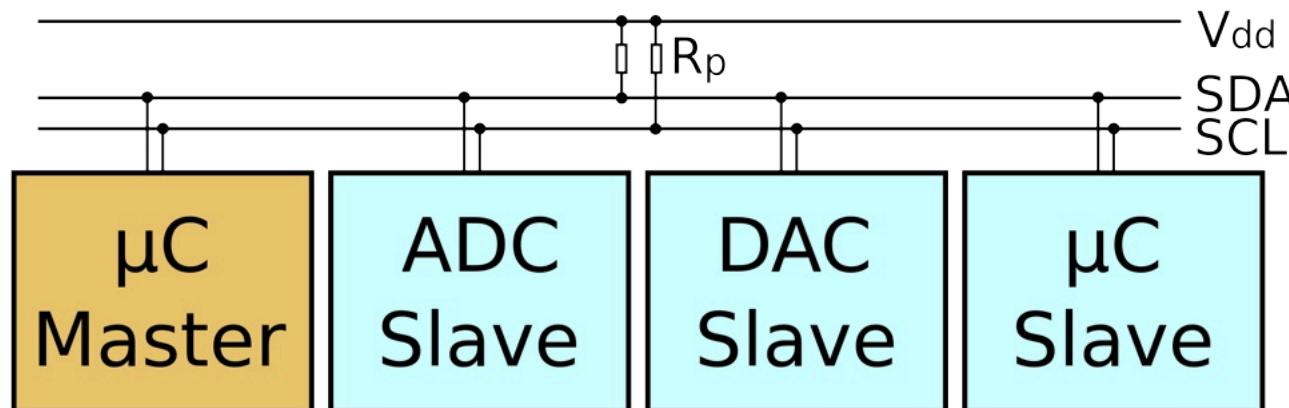
6 [d WU f U a ` ` WL
h [\$5 !; E B ; b d a f ;

I Z S f [e [` f W
b d a f a U a ^ 1

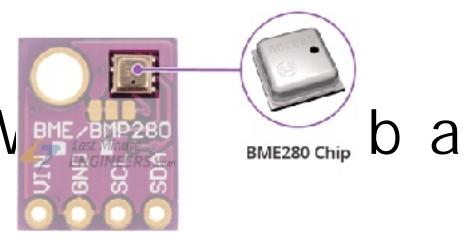
1) Inter-Integrated Circuit (I²C)



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).

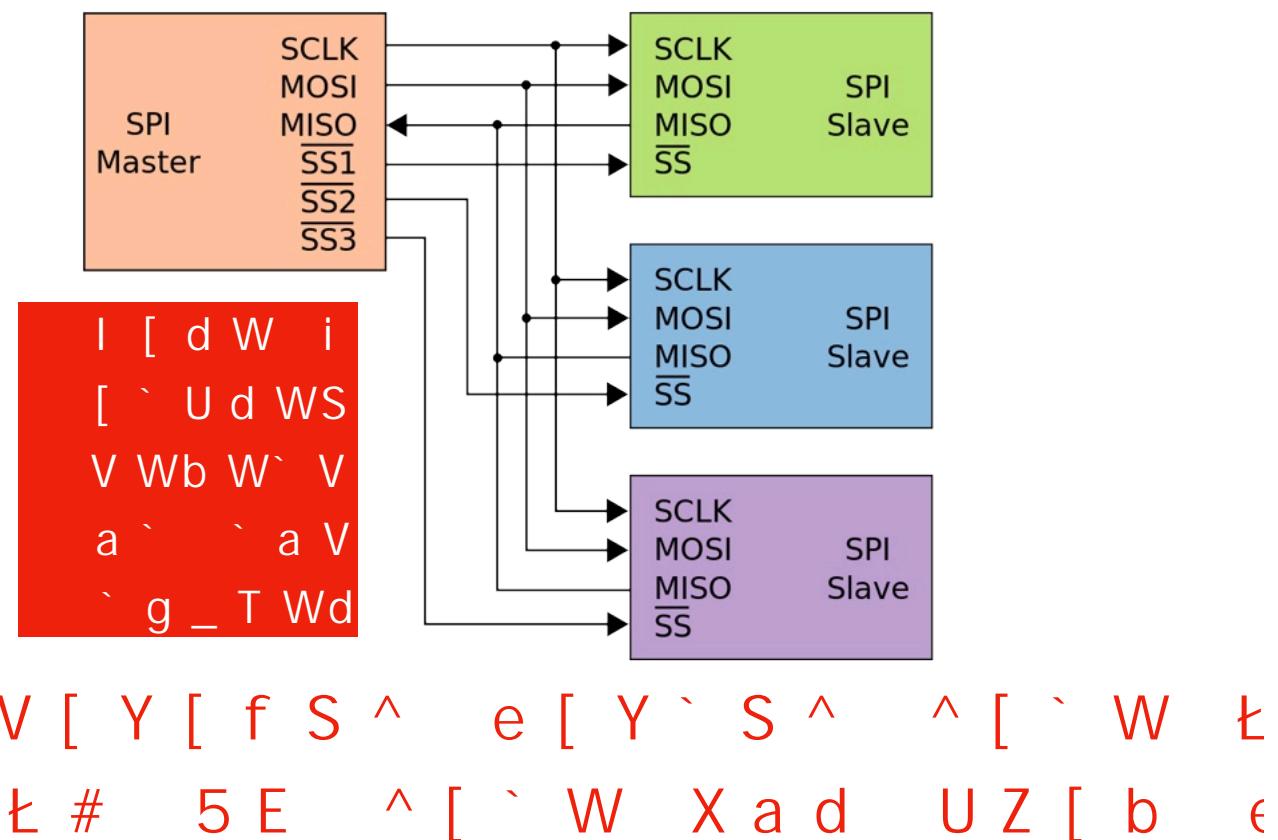
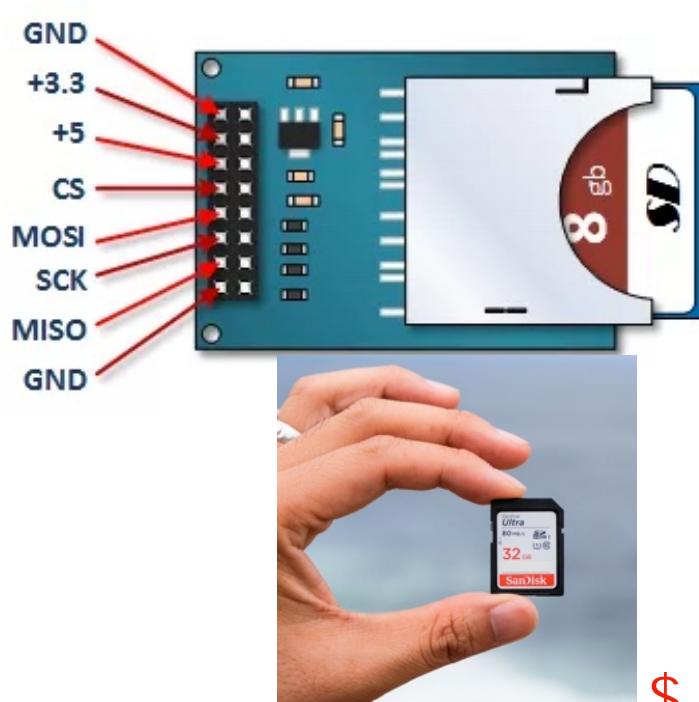


\$ V [Y [f S ^ e [Y ` S ^ ^ [` V
E W^ WU f WV T k S VgV d[WegeWŽ



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.



3) Universal Asynchronous Receiver/Transmitter (UART)

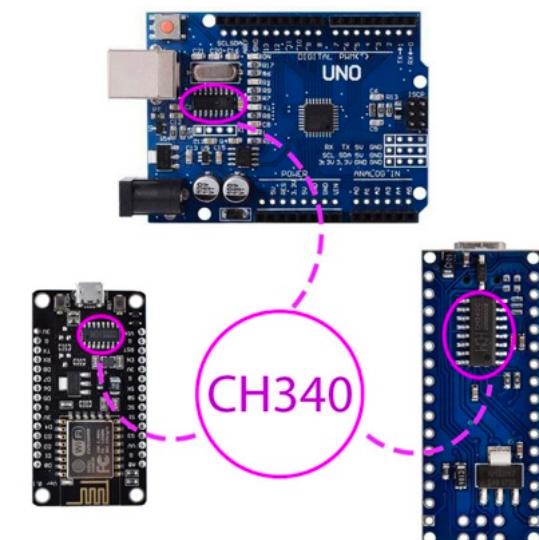
3 g ` [h Wd e S ^ S e k ` U Z d a ` a g e d W U W [h Wd Ž f d
U a _ b g f Wd Z S d V i S d W S e k h U Z d a X a g e e Wd [S ^ U a
[` i Z [U Z f Z W V S f S X a d _ S f S ` V f d S f Y e g _ d [S e T e ^ [V
F Z W W ^ W U f d [U e [Y ` S ^ [` Y ^ Wh W ^ e S d W Z S ` V ^
f a f Z W G 3 D F ž F i a U a _ _ a ` e D E Ž \$ % \$ \$ Wh # W \$ ^ ž e h
e k e f W _ l D E Ž & ! ' S ' ž h a ^ f e k e f W _ ž



A ^ V B 5 D E Ž \$ % \$ 5
E Wd [S ^ b a d f



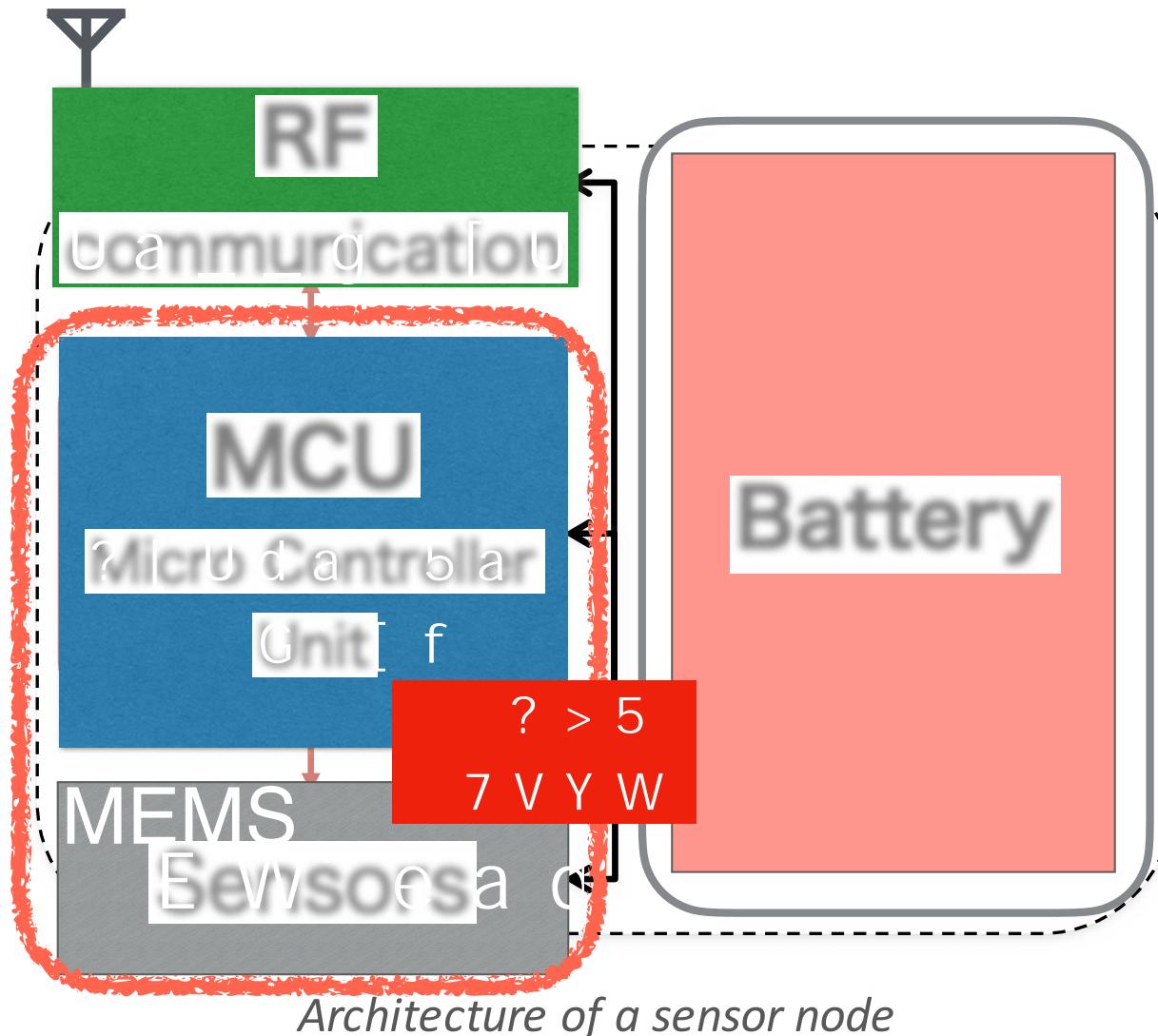
G E 4 / G ` [h Wd e S ^ E Wd ^ [s a e f 4 g e f i ^ ? 5 (
D Wb ^ S U W V [f [` U ^ g V W e G E 4 ž
U a ` h Wd f Wd U ;
7 j ž 5 : % & "



I [` V a i e d W U a Y ` 5 [A I ? W g [f U S e [a ` b a d f

Cutting edge IoT node architecture

Edge AI



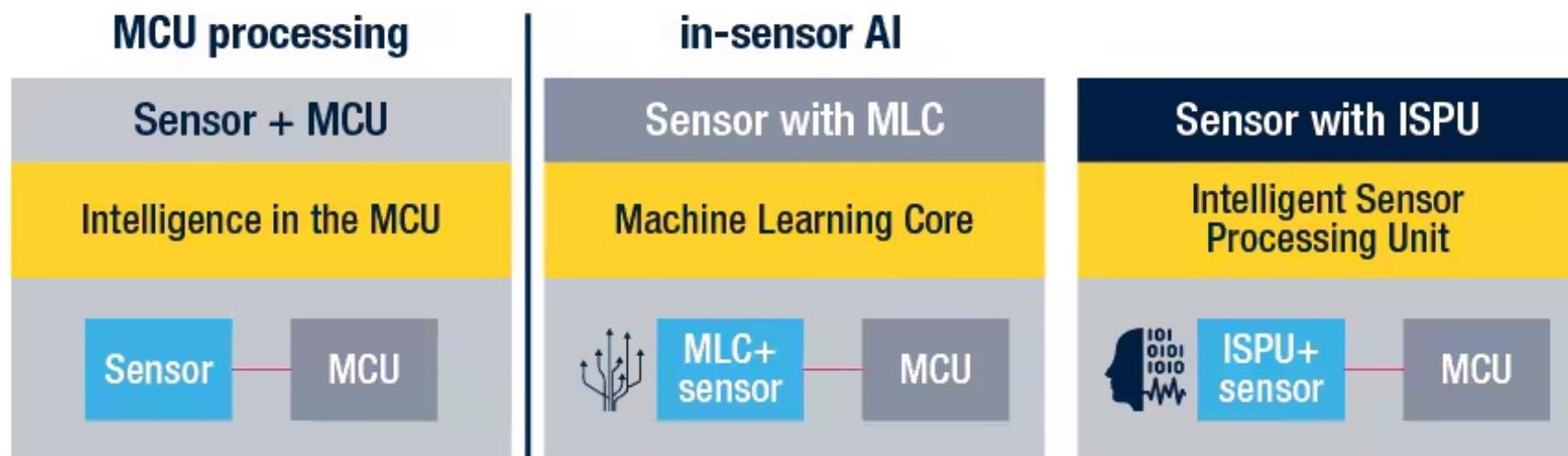
Explains the internal details of IoT applications in the market

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

B S ` S e a ` [U WŽ T [] V

MCU processing

Sensor + MCU

Intelligence in the MCU

Sensor —————— MCU

DETECTING LOW TIRE PRESSURE ON E-BIKES VIA EDGE AI

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



Implementation:

Edge AI processes data locally and triggers a warning alarm



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



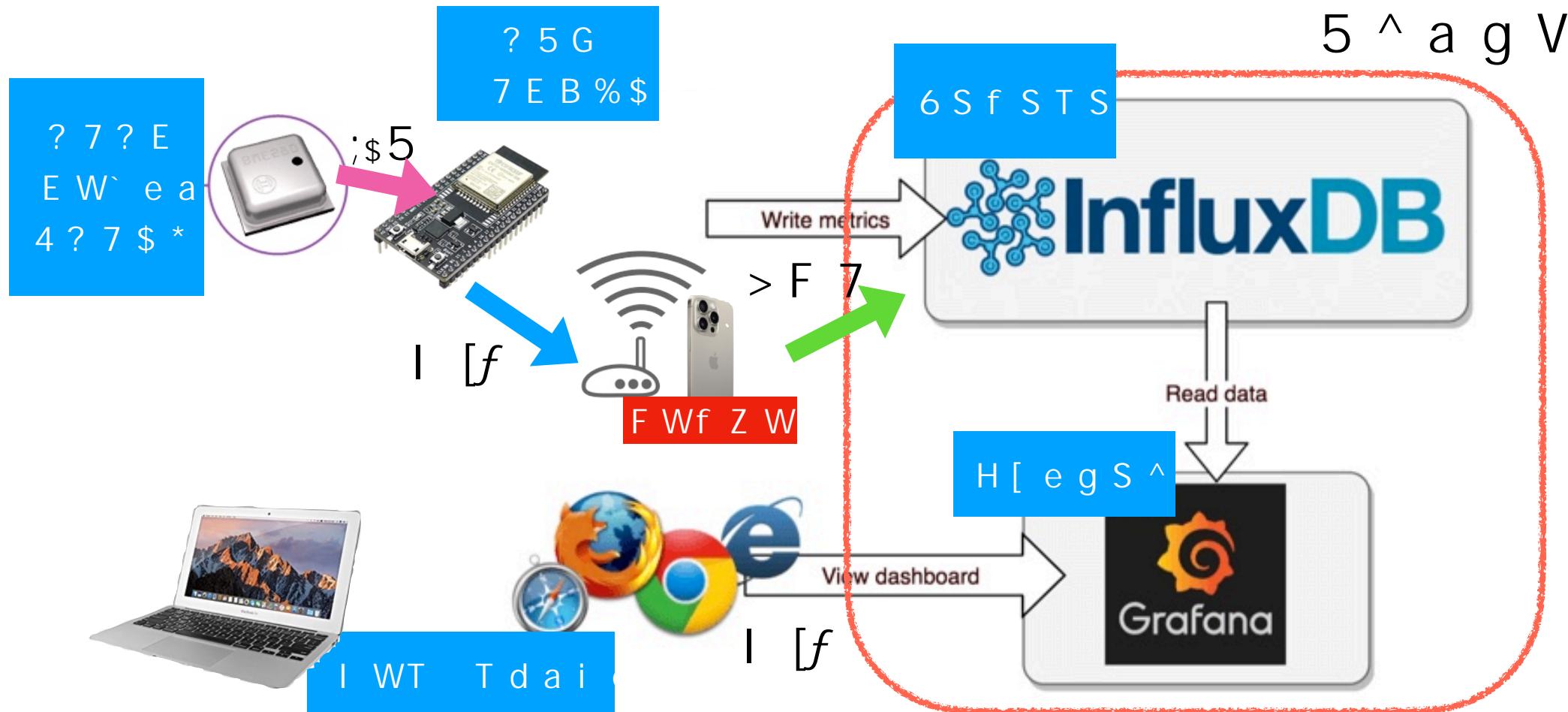
9 W` Wd S f WV T k

Contents

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

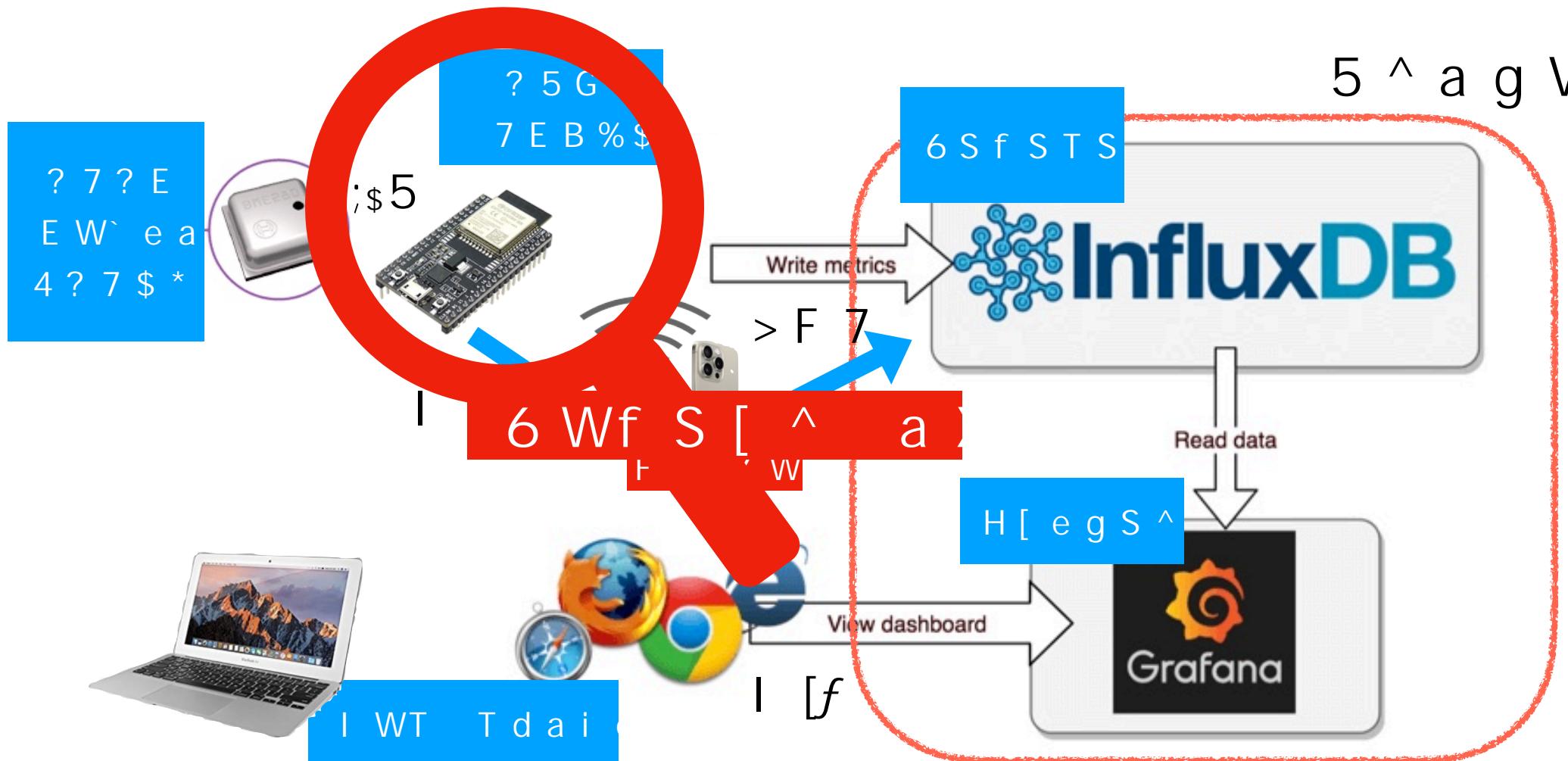
Goal of tutorial

E W` e [` Y V S f? S7 ? X d W` _h [d a ` _ W` f S ^ e W` e
0 Y S f Z Wd ` b d a U We? e 5 G Y V S f S a`
0 f d S ` e 2 g j 6 4 V S f S T S e W a` U ^ a g V
0 h [e g S ^ 9 d W X S T O S V [e b ^ S k a` i WT



Goal of tutorial

E W` e [` Y V S f? S7 ? X d W` _h [d a ` _ W` f S ^ e W` e
0 Y S f Z Wd ` b d a U We? e 5 G Y V S f S a`
0 f d S ` e 2 g j 6 4 V S f S T S e W a` U ^ a g V
0 h [e g S ^ 9 d W X S T O S V [e b ^ S k a` i WT



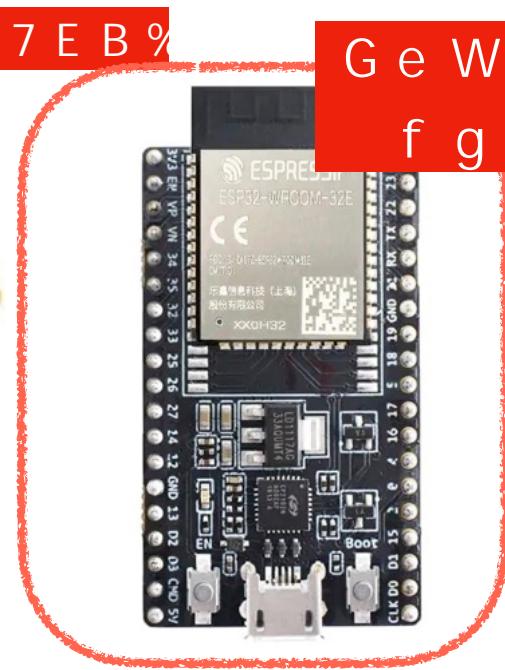
Wide variety of MCU variations

3 D ? Ž T S e WV ? 5 G [e b a b g ^ Š bdb ^ fwZ We WV t a ^
V S k e 3 D ? T S e WV U a d W

D S e b T Wd d k

7 E B % G e W f Z [f g f a d [

' D 8 ' \$ *



7 S U Z U a _ b S ` k i e a d [Y [` S ^ ; Šk7 T W g e WV [/ ; ` f W Y d S f WV 6 Wh W ^ a b _ W ` f 7 ` h [d a ` _ W ` f fi

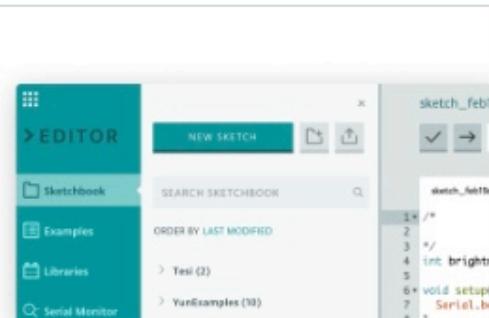
9 W ` Wd [U V Wh W ^ a b _ W ` f W ` [c
g e W X g ^ [` f Wd _ e a X S e e f





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](#)

Downloads

A d [Y [` S ^ ^ k 3 d V g [` a 6 Wh W^ a b _ \
4 g f ^ [T d S d [We E g b b a d f



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 autocomplete, 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

4 S e WV a ` 5 ! 5 Ł Ł ^ S ` Y g S Y W

Comparison of SBC and MCU

E 4 5

? 5 G

E [` Y ^ W 4 a S d V 5 a _ b g f? Wd a 5 a ` f a d a ^ '
/ > [] W S e _ S ^ ^ B 5 / 5 B G Ł b Wd [b Z

D S e b T Wd c



7 E B %



? g ^ f [b ^ W
E a X f i S d We
9 S _ V I WT B k f Z
A E > [` g j ž ž
: S d V i S d W E 4 5 ž ž

@ a A E
E [` Y ^ W e a X 8 f [i d S _ d W / d W
W ž Y ž F W _ b ž

F W _ b ž 5 a ` f
: S d V i S d W

H Wd k Wj [T ^ W a b Wd S f [a H Wd k d a T g e f a l
d Wc g [d W e Z g f V a i ` b d a U W V g \ g e f b a i Wd :

This tutorial uses Python on the MCU as an alternative solution

MicroPython

DOWNLOAD DOCS DISCORD DISCUSSIONS WIKI STORE

MicroPython

MicroPython is a lean and efficient implementation of the Python 3 programming language, based on the CPython interpreter and the Python standard library. It is optimised for a wide range of embedded microcontrollers and other constrained devices.

The MicroPython team has created a Python operating system that can be used to control all kinds of electronic projects.

MicroPython is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more. Yet it is compact enough to fit and run within just 256k of code space and 16k of RAM.

MicroPython aims to be as compatible with normal Python as possible to allow you to transfer code with ease from the desktop to a microcontroller or embedded system.

Programming language for small resources compatible with Python

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

C++ IDE vs. μ Python for development

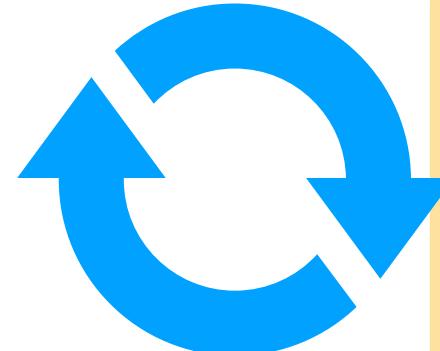
5 Ł Ł



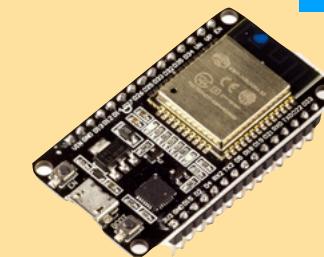
A ` B



Arduino IDE



A ` ? 5

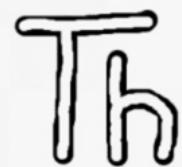


> [f f ^ W U a _

? [U d a B k f



A ` B

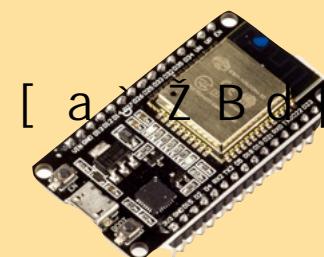
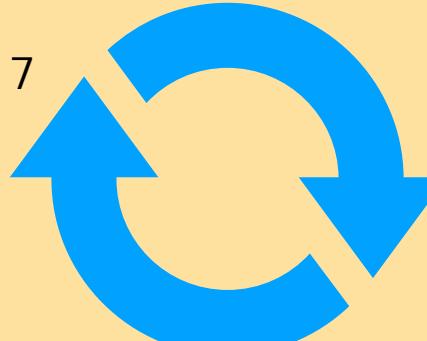


D 7 B >

F Z a `` k ; 6 7

A ` ? 5

D W S V Ź 7 h S ^ g S f [a] Ź B d R ` f > a a k

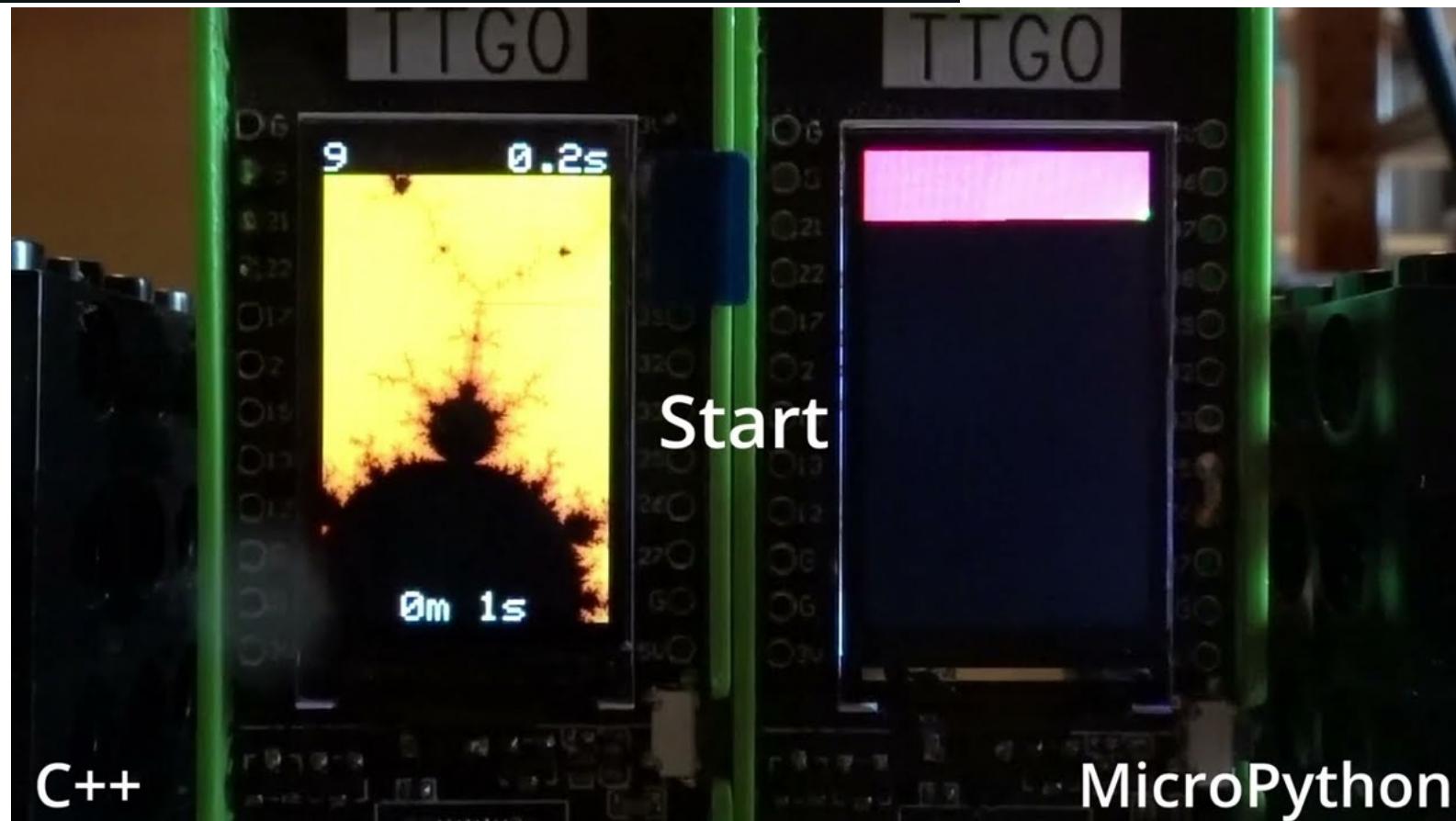


H W d k W S e k X
b d a f a f k b [` Y

C++ IDE vs. μPython (performance)

r/esp32 • 2 yr. ago
TomFlatterhand

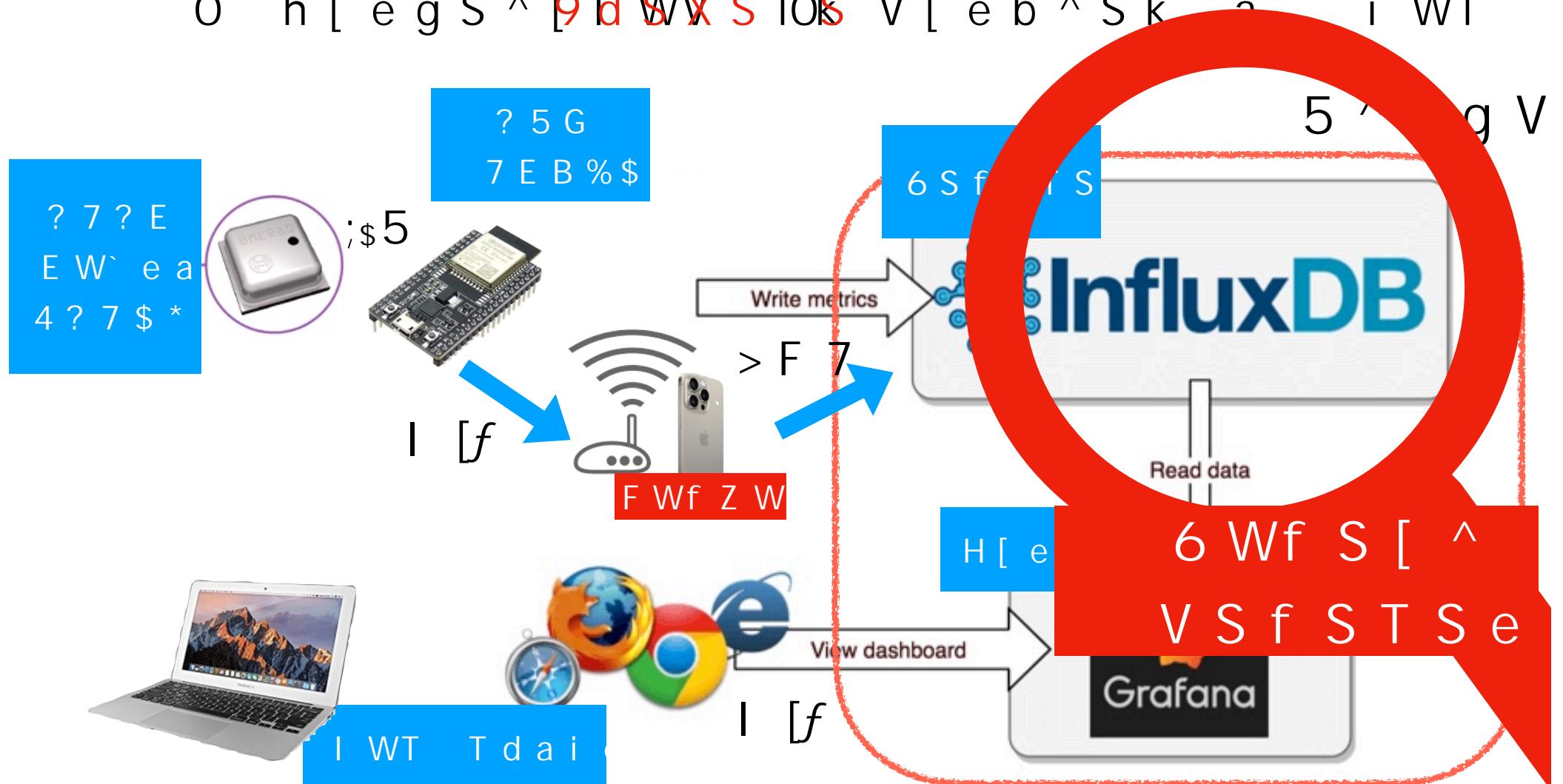
How fast is Python - MicroPython
versus C++



5 Ł Ł [e & " " f [_ We X S e f Wd f Z S ` 1 B k
0 1 B k f Z a ` [e X a d b d a f a f k Ł
0 5 Ł Ł [e X a d b d a V g U f [a `

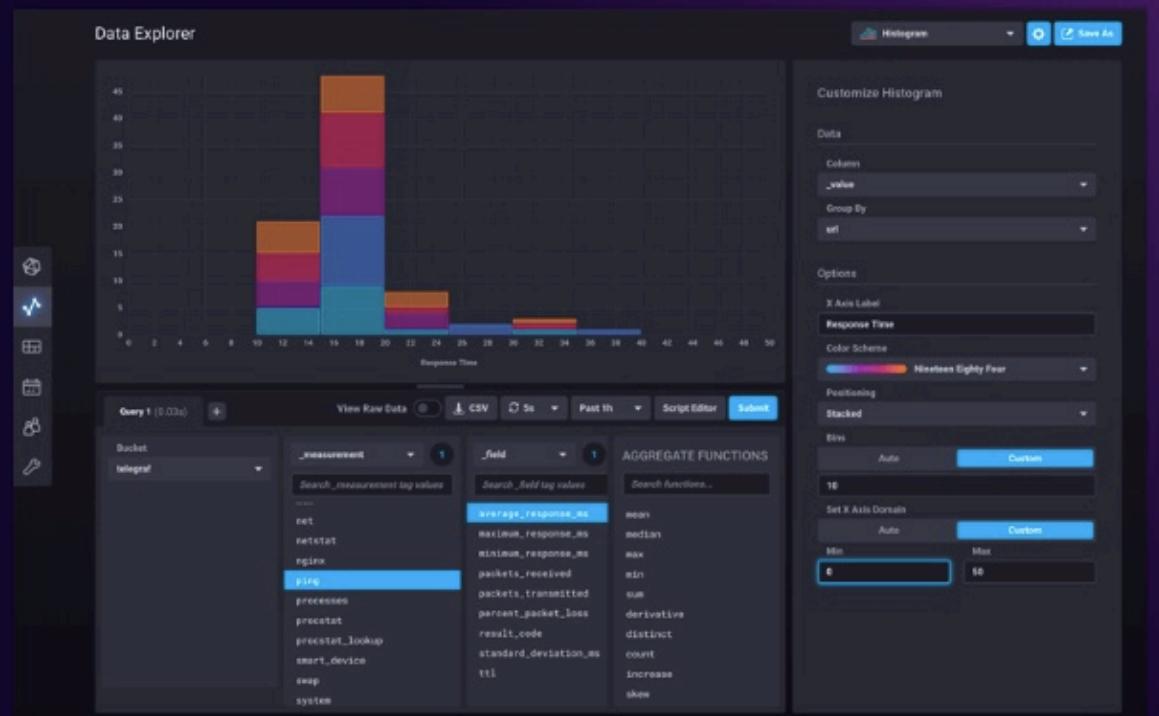
Goal of tutorial

E W` e [` Y V S f? S7 ? X d W` _h [d a ` _ W` f S ^ e W` e
0 Y S f Z Wd ` b d a U We? e 5 G Y V S f S a`
0 f d S ` e 2 g j 6 4 V S f S T S e W a` U ^ a g V
0 h [e g S ^ 9 d W X S T O S V [e b ^ S k a` i W T

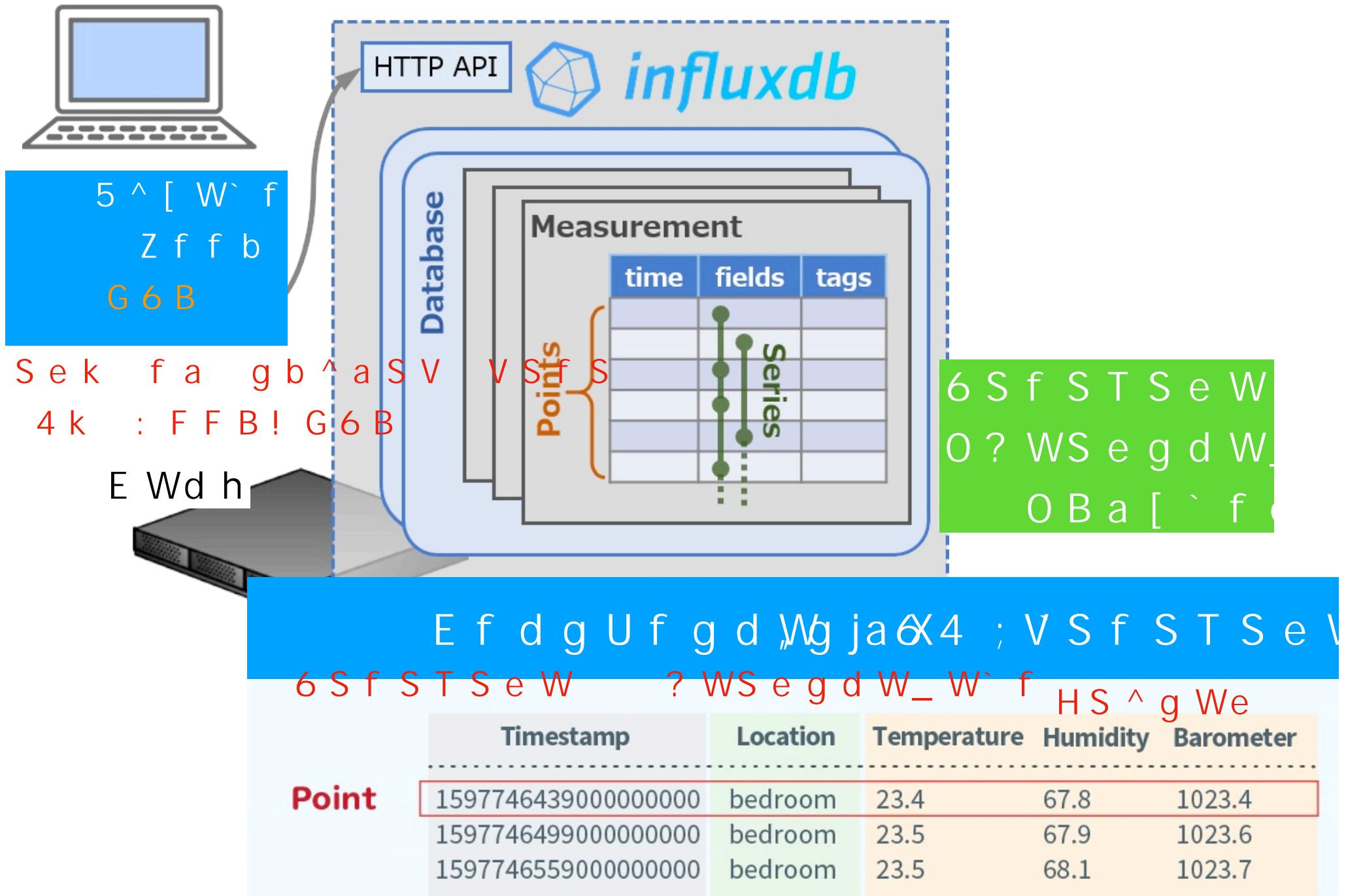


InfluxDB

InfluxDB is a time series database designed to handle high write and query loads.

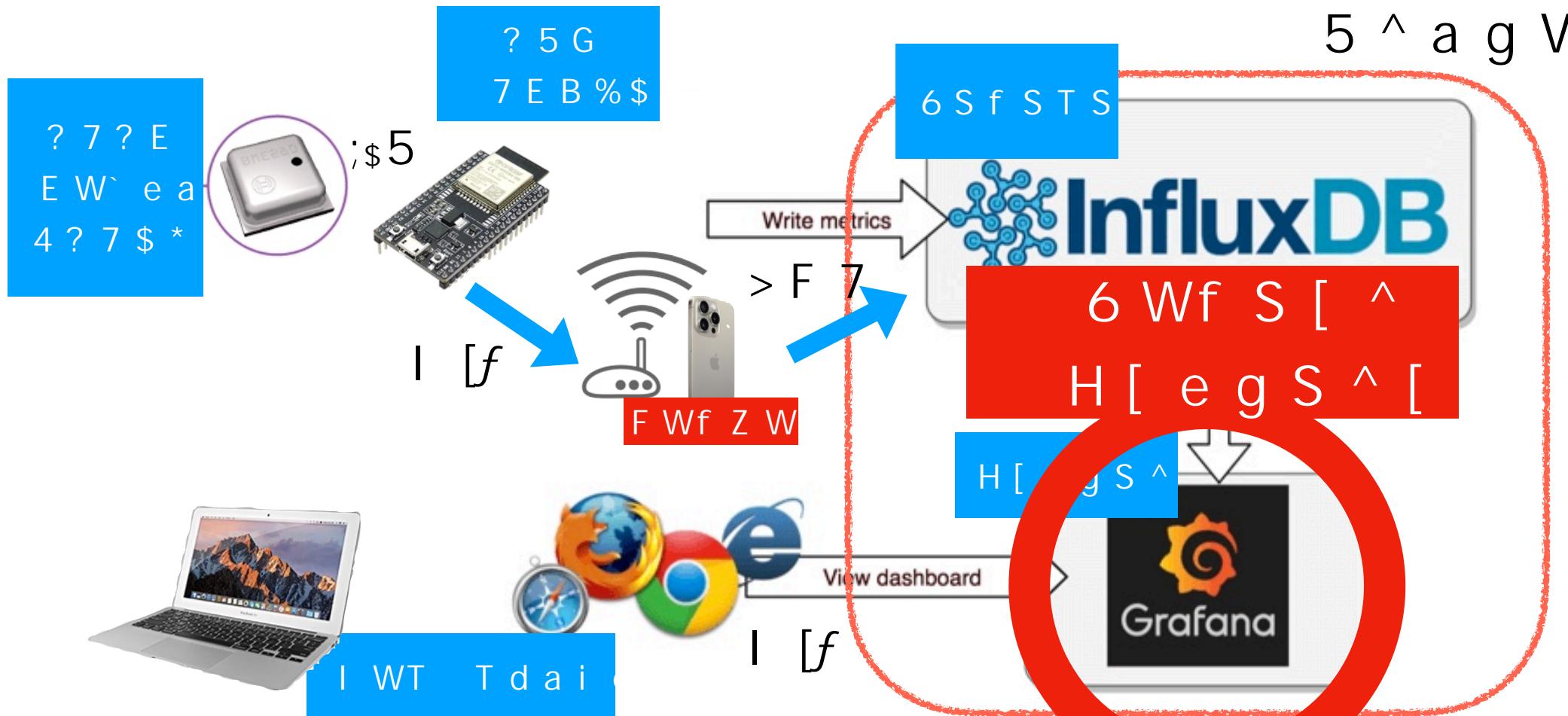
[Get InfluxDB](#)

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**.



Goal of tutorial

E W` e [` Y V S f? S7 ? X d W` _h [d a ` _ W` f S ^ e W` e
0 Y S f Z Wd ` b d a U We? e 5 G Y V S f S a`
0 f d S ` e 2 g j 6 4 V S f S T S e W a` U ^ a g V
0 h [e g S ^ 9 d W X S T O S V [e b ^ S k a` i WT



The leading **open source** software for time series analytics

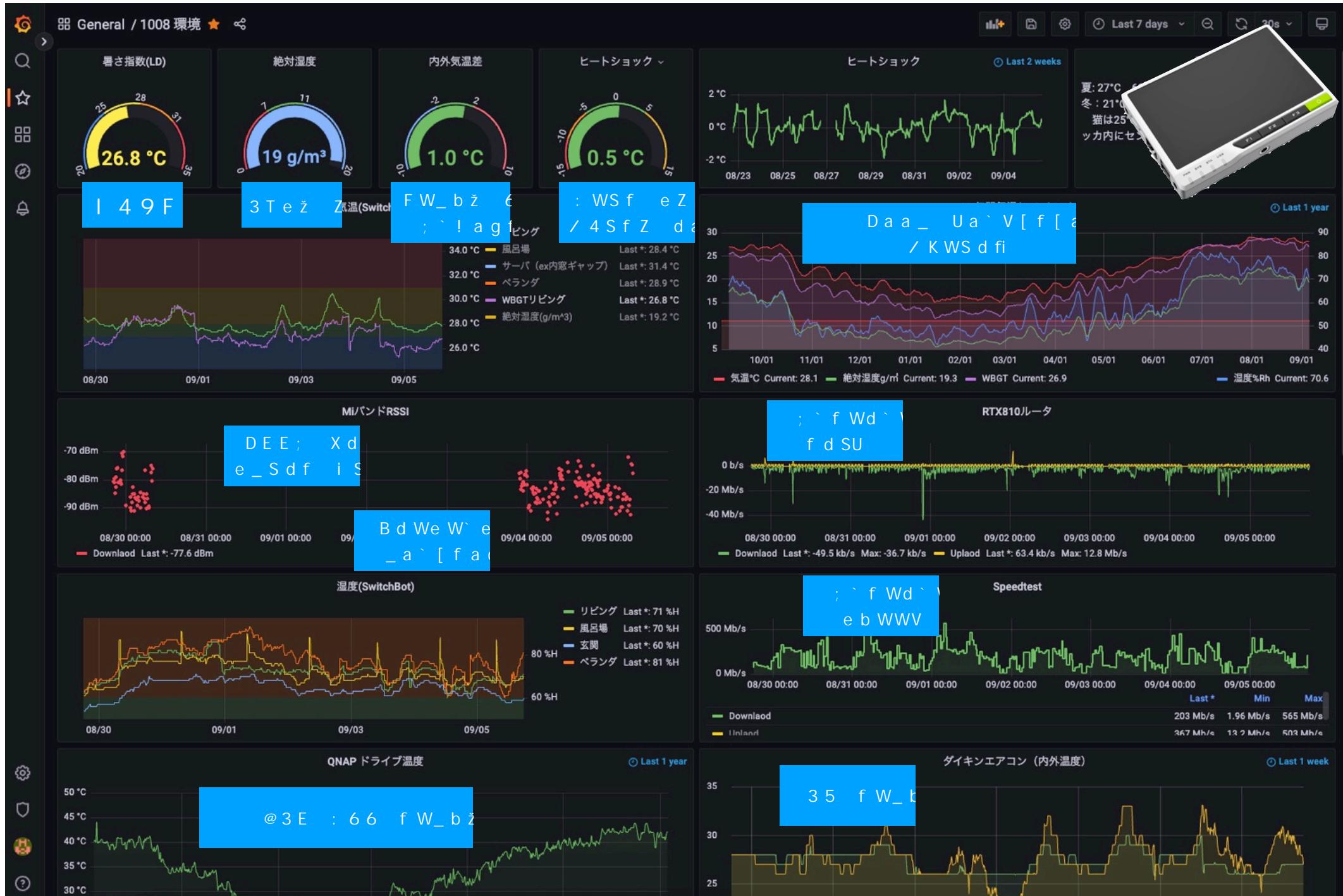
The Grafana homepage features a dark-themed dashboard with several data visualization components. At the top center is the Grafana logo. Below it is a subtitle: "No matter where your data is, or what kind of database it lives in, you can bring it together with Grafana. Beautifully." The dashboard displays five main data sources and their corresponding visualizations:

- Graphite**: A bar chart showing data over time.
- Elasticsearch**: A table showing historical data points with dates and values.
- Cloudwatch**: A line chart showing a fluctuating metric over time.
- InfluxDB**: A line chart with heart icons indicating status levels.
- Prometheus**: Two gauge charts showing numerical values (57 and 78).
- Hosted Metrics**: A world map with colored dots representing data distribution.

At the bottom of the dashboard are three calls-to-action: "Learn More", "Live Demo", and "Get Grafana".

9 d S X S ` S S ^ ^ a i e k h a [g e g f S t ^ [S l g W W b k f l a ` S ` V g `
_ W f d [U e ` a _ S f f W d i Z W d W f Z W k S d W e f a d V
V S e Z T a S d V[e f Z k a g d f W S _ S ` V X a e f W d S V S

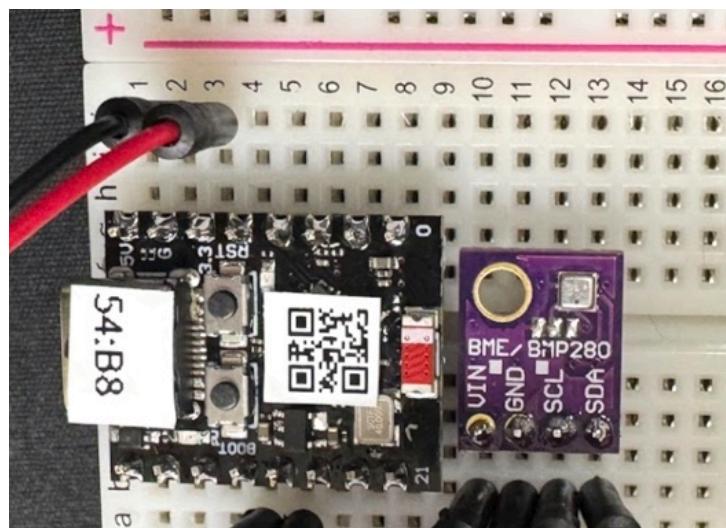
Example of Grafana (my home env.) on RPi



4 WX a d W f Z W

E b ^ [f f a \$ e _ S W ė f W S S e 3 # { 3 #] 3 # S t 3 #
/ _ [j WV g ` [h Wd e [f [We f

GROUP A				TEAM	GROUP B				TEAM	GROUP C				TEAM
PHILIPPE	Nicolas	MAM2	FRA	3 # \$	NOMREAU FATAZ	Julien	MAM2	FRA	4 # \$	THIAW	Sidy	MAM2	FRA	5 # \$
ENATSU	Koichi	Tokyo	JPN	A1	RAJERISON	Lorry	FISA5	FRA	B1	SALAMEH	Ahmad	MAM2	FRA	C1
TAKAGI	Yukihiro	Waseda	JPN		AMANO	Sota	Waseda	JPN		OKAWA	Yutaro	Waseda	JPN	
CRIADO	Romain	FISA5	FRA		NAGAMINE	Yuma	Waseda	JPN		TSUDA	Kaoru	Waseda	JPN	
GHERBI	Choayb	FISA5	FRA	3 #]	UMEKI	Shusuke	Tokyo	JPN		CARRERA	Alexis	FISA5	FRA	
VALLIER	Lucillien	FISA5	FRA		PESSÉ	Victor	FISA5	FRA	4 #]	BEAUTIER	Antoine	FISA5	FRA	5 #]
REY	Dimitri	FISA5	FRA		BAZIN	Khalil	FISA5	FRA		LO	Baye Djily	FISA5	FRA	
					DOIX	Paul	FISA5	FRA		FRANCHINI	Antoine-Hugo	FISA5	FRA	
DJERABE	Gauthier	MAM2	FRA		PALLIER	Tristan	MAM2	FRA		LATRECHE	Zakaria	MAM2	LBN	
WANG	Koichi	Tokyo	CHN	3 \$ \$	SATAKE	Rito	Waseda	JPN		CHIKUSA	Yuto	Waseda	JPN	5 \$ \$
TODA	Ittetsu	Waseda	JPN	A2	KAWAHARA	Yuto	Waseda	JPN		YUZAWA	Koki	Waseda	JPN	
SHIOKAWA	Yuma	Waseda	JPN		LEMOS	Hervé	FISA5	JPN		SHIMURA	Yuto	Tokyo	JPN	
GUIA	Nicolas	FISA5	FRA		BEAUFORT	Nicolas	FISA5	FRA	B2	ROQUES	Alexandre	FISA5	FRA	C2
ANANI	Nawel	FISA5	FRA	3 \$]	FONSECA	Lucas	FISA5	FRA		GAILLARD	Adrien	FISA5	FRA	
BERNIER	Théo	FISA5	FRA		SOUSA	Khaoula	FISA5	FRA	4 \$]	HEBERT	Tanguy	FISA5	FRA	5 \$]
MIRET	Thibault	FISA5	FRA		KHATRI	Hamza	MAM2	FRA		SAUNIER	Jérémie	FISA5	FRA	



7 S U Z Y d a g b g e We 7 E B
4 ? 7 \$ * " e W` e a d

ł k a g a i



F a V S k

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

F Z [e S X f Wc

E b ^ [f f a \$ e _
Wj ž

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

3

9 d a g b 3 # 3
/ _ [j WV g ` [h V

Thursday Sept 18

Friday Sept 19

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

Coffe Break

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

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

enge: Start
15 - 11:45
roups : B

5

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

90' Practica
10:00 - 11:
Y. Suzu
Design of bro
vibration ene
device
CXXX - Gro

4

Tutorial instruction



F a V S k

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

F Z [e S X f Wc

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

3

E b ^ [f f a \$ e _
Wj ž
9 d a g b 3 # 3
/ _ [j WV g ` [h \

Thursday Sept 18

Friday Sept 19

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

Coffe Break

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

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

enge: Start
15 - 11:45
roups : B

5

Challenge
10:00 - 11:30
Group : C

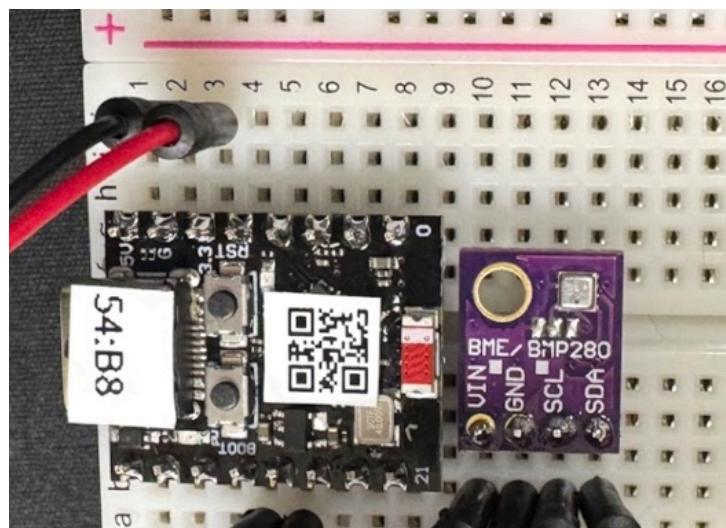
90' Practical work
10:00 - 11:30
T. Fujita
IoT modules tutorial using
SBC
CXXX - Group : B

90' Practica
10:00 - 11:
Y. Suzu
Design of bro
vibration ene
device
CXXX - Gro

4

E b ^ [f f a \$ e _ S W ė f A W S _ e 3 # 3 # 3 # S t 3 #
 / _ [j WV g ` [h Wd e [f [We f

GROUP A				TEAM	GROUP B				TEAM	GROUP C				TEAM
PHILIPPE	Nicolas	MAM2	FRA	3 # 3	NOMREAU FATAZ	Julien	MAM2	FRA	4 # 3	THIAW	Sidy	MAM2	FRA	5 # 3
ENATSU	Koichi	Tokyo	JPN	A1	RAJERISON	Lorry	FISA5	FRA	B1	SALAMEH	Ahmad	MAM2	FRA	C1
TAKAGI	Yukihiro	Waseda	JPN		AMANO	Sota	Waseda	JPN		OKAWA	Yutaro	Waseda	JPN	
CRIADO	Romain	FISA5	FRA		NAGAMINE	Yuma	Waseda	JPN		TSUDA	Kaoru	Waseda	JPN	
GHERBI	Choayb	FISA5	FRA	3 # 1	UMEKI	Shusuke	Tokyo	JPN		CARRERA	Alexis	FISA5	FRA	
VALLIER	Lucillien	FISA5	FRA		PESSÉ	Victor	FISA5	FRA	4 #	BEAUTIER	Antoine	FISA5	FRA	5 # 1
REY	Dimitri	FISA5	FRA		BAZIN	Khalil	FISA5	FRA		LO	Baye Djily	FISA5	FRA	
					DOIX	Paul	FISA5	FRA		FRANCHINI	Antoine-Hugo	FISA5	FRA	
DJERABE	Gauthier	MAM2	FRA		PALLIER	Tristan	MAM2	FRA		LATRECHE	Zakaria	MAM2	LBN	
WANG	Koichi	Tokyo	CHN	3 \$ 3	SATAKE	Rito	Waseda	JPN		CHIKUSA	Yuto	Waseda	JPN	5 \$ 3
TODA	Ittetsu	Waseda	JPN	A2	KAWAHARA	Yuto	Waseda	JPN		YUZAWA	Koki	Waseda	JPN	
SHIOKAWA	Yuma	Waseda	JPN		LEMOS	Hervé	FISA5	JPN		SHIMURA	Yuto	Tokyo	JPN	
GUIA	Nicolas	FISA5	FRA		BEAUFORT	Nicolas	FISA5	FRA	B2	ROQUES	Alexandre	FISA5	FRA	C2
ANANI	Nawel	FISA5	FRA	3 \$ 1	FONSECA	Lucas	FISA5	FRA		GAILLARD	Adrien	FISA5	FRA	
BERNIER	Théo	FISA5	FRA		SOUSA	Khaoula	FISA5	FRA	4 \$	HEBERT	Tanguy	FISA5	FRA	5 \$ 1
MIRET	Thibault	FISA5	FRA		KHATRI	Hamza	MAM2	FRA		SAUNIER	Jérémie	FISA5	FRA	



7 S U Z Y d a g b g e We 7 E B
 4 ? 7 \$ * " e W` e a d

ł k a g a i



8[` S ^ Y a S ^ ?
ai` Y d S b Z



? 5 G Ž e W` e a d
U a ` WU f [a



```
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(scl=Pin(SCL), sda=Pin(SDA))
13 bme = BME280.BME280(i2c)
14
15 # Wi-Fi credentials
16 SSID = "JFWM-WS"
17 PASSWORD = "goodtime"
18
19 print(f"Connecting to Wi-Fi network: {SSID}")
```

Tutorial list

; ` e f d g U f [a ` ^ e S _ b ^ W b d a Y d
Z f f b , ! ! [a f i a d] ž a d '

B d Wb S d S f [a `

" ž ? [U d a B k f Z a ` [` e f S ^ ^ a ` 7 E B

7 j b Wd [_ W` f e

ž 4 ^ [`] > 7 6 a ` 7 E B % \$

\$ ž 5 Z WU] [` Y f Z W I [8 [X g ` U f [a ` a X 7 E B

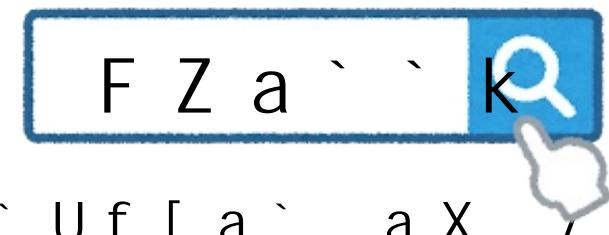
% ž I [d W 4 ? 7 \$ * " f a Y Wf V S f S

/ 4 ? 7 \$ * " ž b k ^ [T d S d k [` e f S ^ ^ S f [a

& ž G b ^ a S V 4 ? 7 \$ * " V S f S f a ~~U ^ a g V~~ h [S G

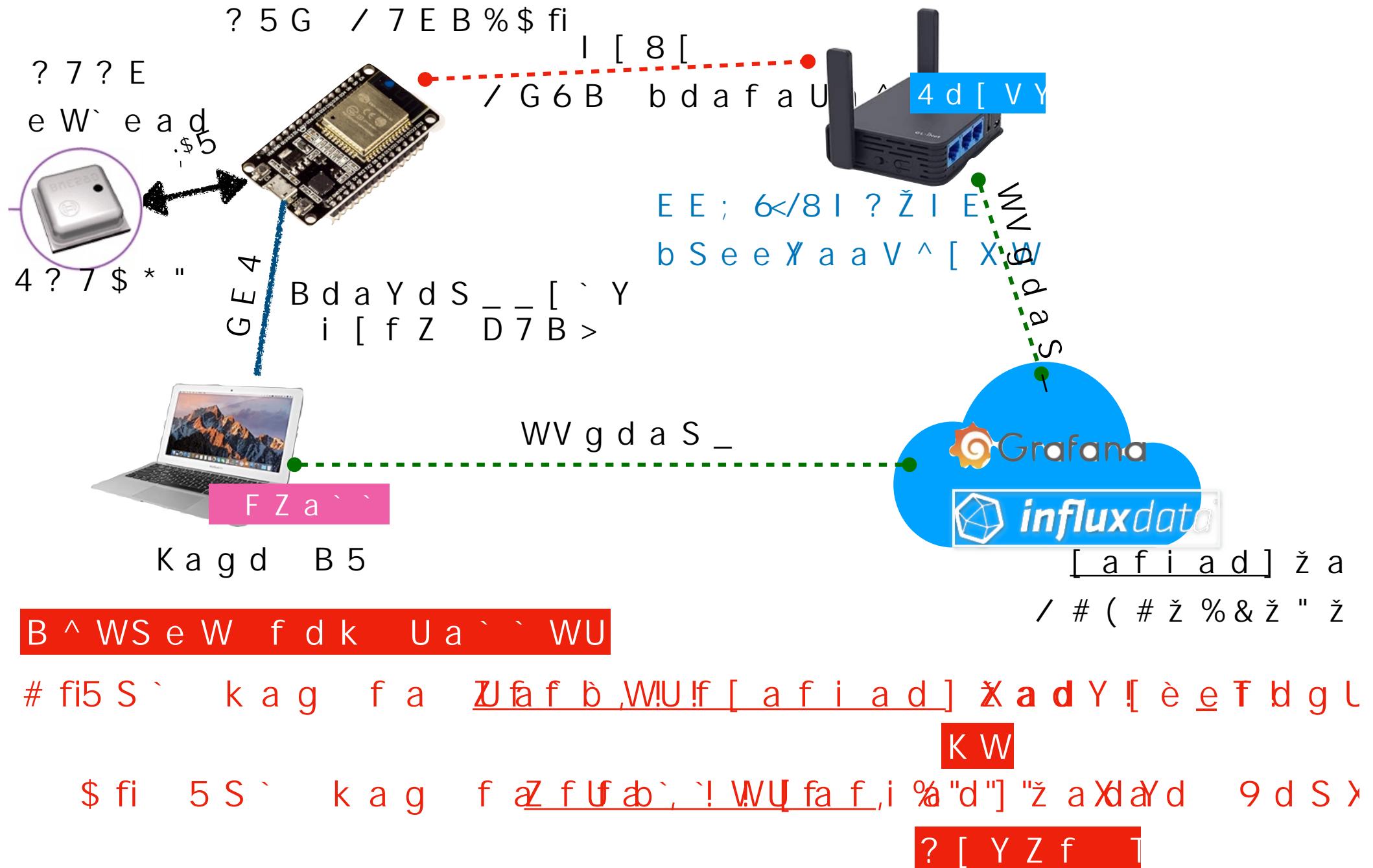
D Wh [e W V O 6 S f S e W` V T k G 3 D F f a B

' ž H [e g S ^ [I S f [a ` a ` 9 d S X S ` S

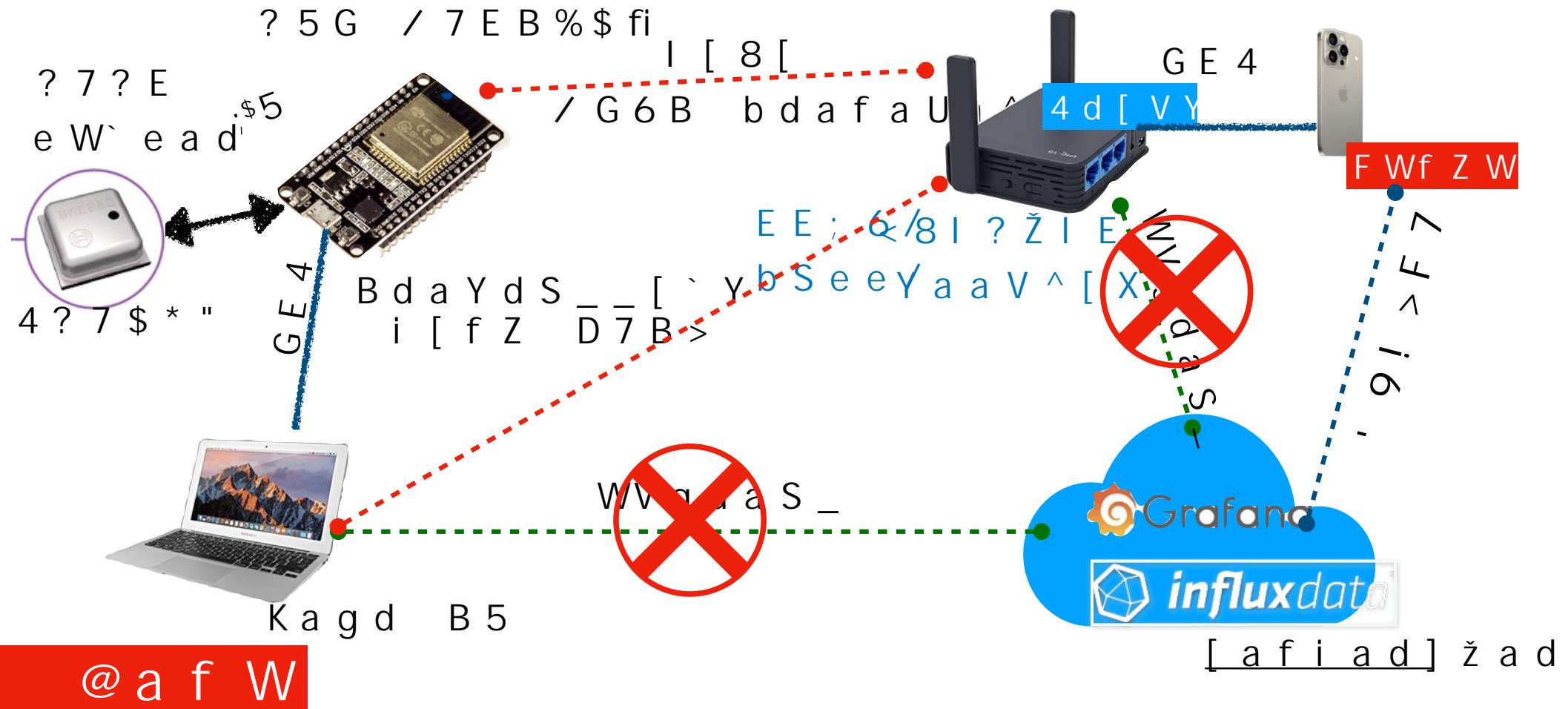


(Extra) BLE Beacon RSSI monitoring on Grafana

Overview of connections



Overview of connections (revised)



WV g d a S _ 2 G E ? 4 i e e W U g d d[W e k f d b [a U f f Z [W k Y 9 [d & X S ` S U a _ .
B ^ W S e W U a ^ < W U f ? Ž f i a E M g d [` Y f Z W f g f a d [S ^ f a U

; b d a h [V WV WE [_ X a d % V S k e g ` '



Search destinations or activities



JPY

Go to app

Help

Recently viewed

Cart



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



5 a fd _ 5 A ? b a d f ` g a T Wd
6 Wh [U W ? S ` S Y Wd

F d a g T ^ W1 1

Google

cp210x driver windows

6 a i ^ a S V f Z V
fd e f t ; i [^]



5 a ` ` WU f 7 E B % \$
f a I [` V a i e B 5 !
i ! 1 Ž G E 4 U S T ^ W

A ` ? S U A E F Wd _ [` S ^ 3 b b ž \ g e f
^ e ! V Wh

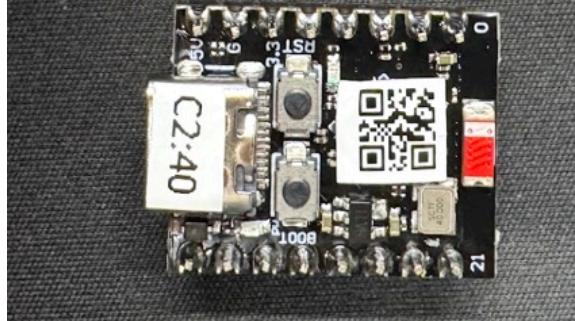
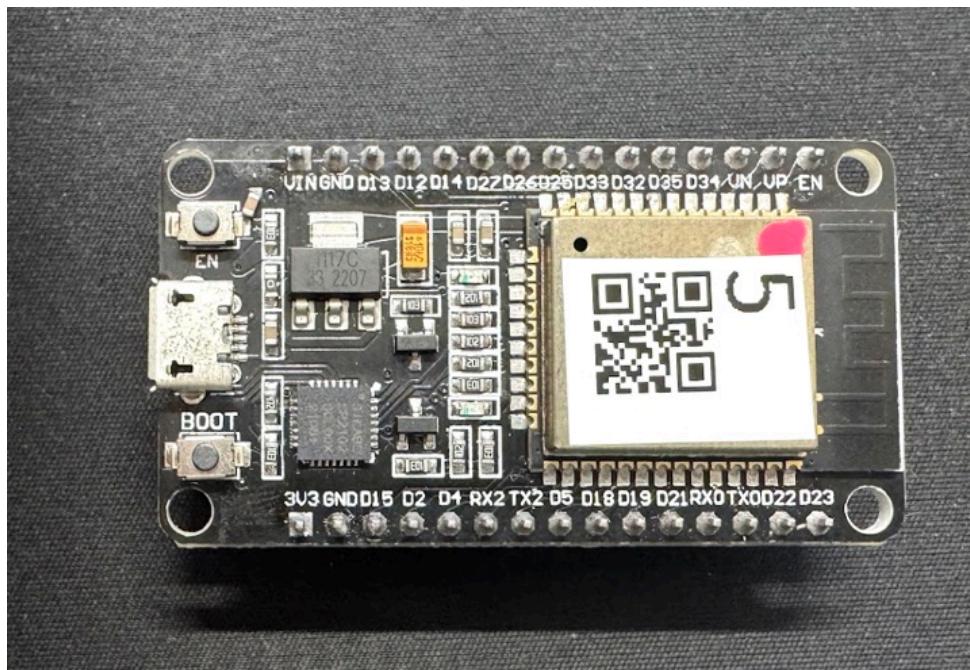
```
(base) takayukifujita@MacBook-Air ~ % ls /dev/cu.*  
/dev/cu.BN0085-BT  
/dev/cu.Bluetooth-Incoming-Port /dev/cu.usbserial-0001
```

a S d V
DVI/CDROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM3)
Printers
Printers
Security

5 A ? % fl

U a ^ g _ , i Z k e f S d
fl V Wb W` V e a
W` h [d a ^ _ W`

ESP32 -> ESP32C3 super mini



AliExpress ≡ 140w Pd Module

Download AliExpress

ESP32-C3 Super Mini WiFi Bluetooth-Compatible Development Board
ESP32 C3 Development Board CORE Board IOT Board for Arduino

★★★★★ 5.0 1 Review | 42 sold

Fall Sale · Welcome deal Starts: Sep 15,

€63.58

€12.49 Sale is coming soon

10 pcs, €1.25/pc; Extra 1% off with coins

Color: 10pc Welded

Tutorial list

; ` e f d g U f [a ` ^ e S _ b ^ W b d a Y d
Z f f b , ! ! [a f i a d] ž a d '

B d Wb S d S f [a `

" ž ? [U d a B k f Z a ` [` e f S ^ ^ a ` 7 E B

7 j b Wd [_ W` f e

ž 4 ^ [`] > 7 6 a ` 7 E B % \$



\$ ž 5 Z WU] [` Y f Z W I [8 [X g ` U f [a d a X , E B

% ž I [d W 4 ? 7 \$ * " f a Y Wf V S f S

/ 4 ? 7 \$ * " ž b k ^ [T d S d k [` e f S ^ ^ S f [a

& ž G b ^ a S V 4 ? 7 \$ * " V S f S f a ~~U ^ a g V~~ h [S G

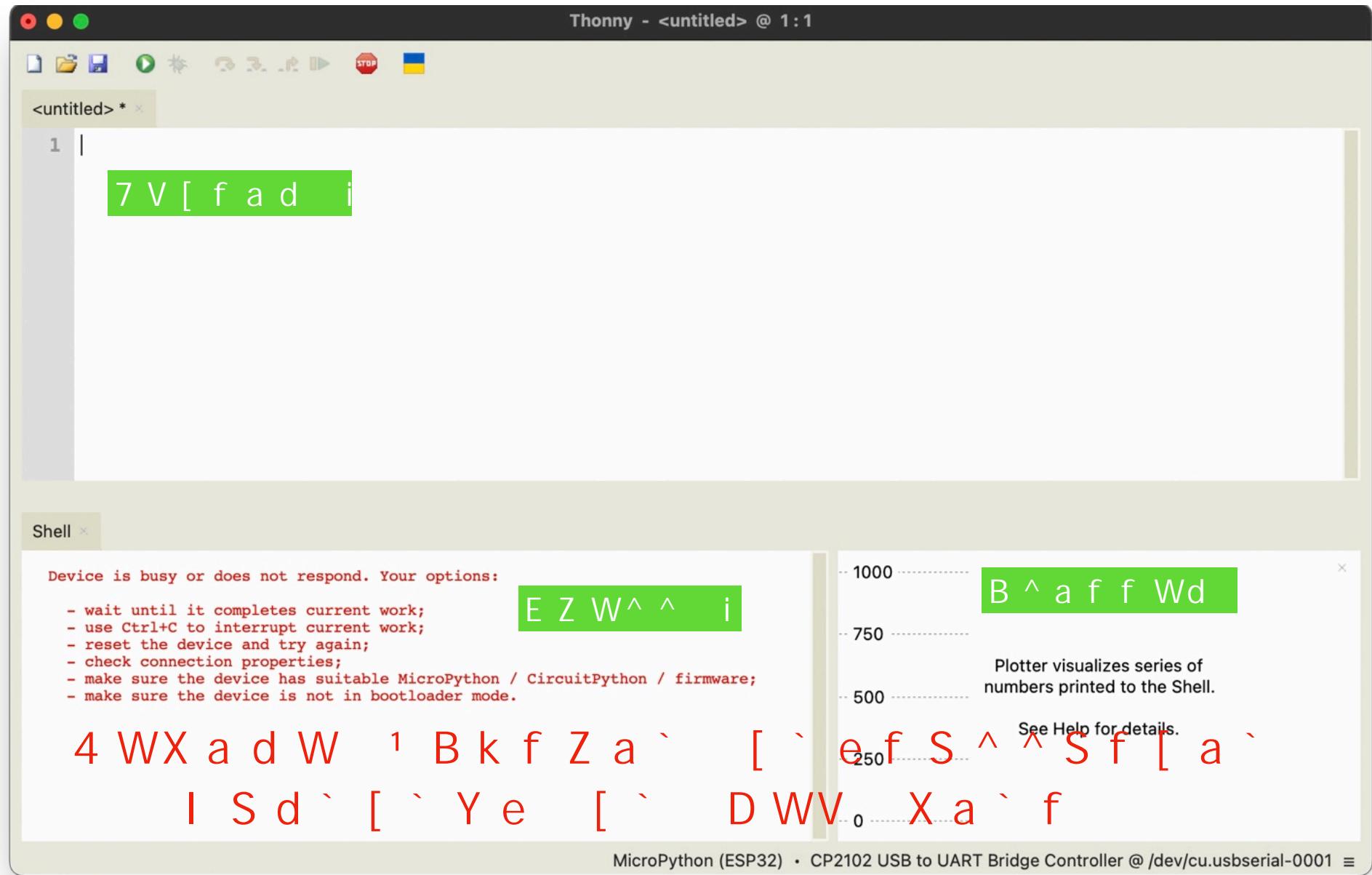
D Wh [e W V O 6 S f S e W` V T k G 3 D F f a B

' ž H [e g S ^ [I S f [a ` a ` 9 d S X S ` S

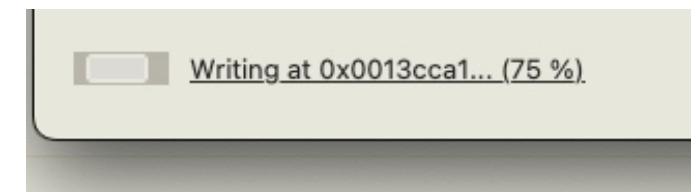
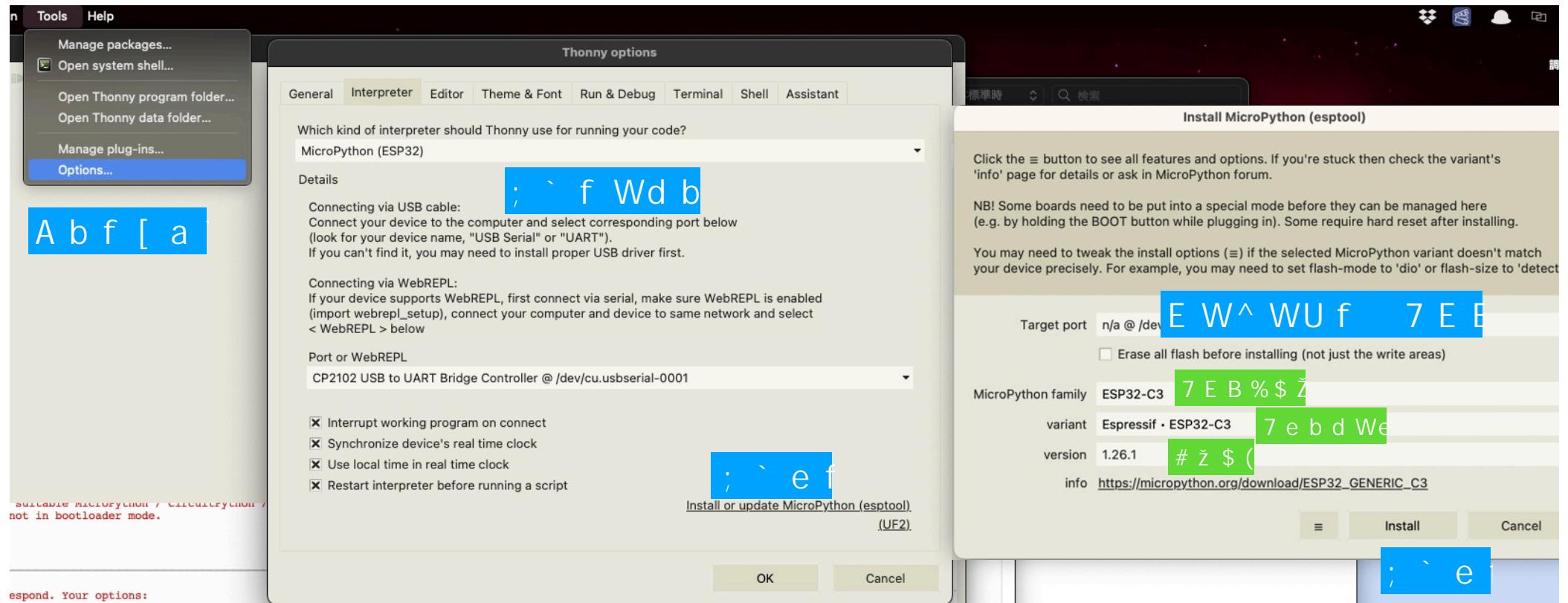
(Extra) BLE Beacon RSSI monitoring on Grafana

Install μ Python via Thonny app

E S _ b ^ W a X F Z a ` ` k q e i [` V a i



Install μPython via Thonny app



Thonny - <untitled> @ 1:1

<untitled> * x

1

Shell x

```
- wait until it completes current work;
- use Ctrl+C to interrupt current work;
- reset the device and try again;
- check connection properties;
- make sure the device has suitable MicroPython / CircuitPython / firmware;
- make sure the device is not in bootloader mode.
```

Process ended with exit code None.

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

Local Python 3 • Thonny's Python

- ✓ MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbserial
- MicroPython (ESP8266) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbse
- MicroPython (generic) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbser
- CircuitPython (generic) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbser

Configure interpreter...

MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbserial-0001 ≡

5 ^ [U] f Z W d [Y Z f T a f f a _
/ 7 E B % \$ a ` e `

@ a i k a g U S ` g e W 1 B k f Z a ` a `

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



Tutorial list

; ` e f d g U f [a ` ^ e S _ b ^ W b d a Y d
Z f f b , ! ! [a f i a d] ž a d '

B d Wb S d S f [a `

" ž ? [U d a B k f Z a ` [` e f S ^ ^ a ` 7 E B

7 j b Wd [_ W` f e

ž 4 ^ [`] > 7 6 a ` 7 E B % \$



\$ ž 5 Z WU] [` Y f Z W I [8 [X g ` U f [a ` a X E B

% ž I [d W 4 ? 7 \$ * " f a Y Wf V S f S

/ 4 ? 7 \$ * " ž b k ^ [T d S d k [` e f S ^ ^ S f [a

& ž G b ^ a S V 4 ? 7 \$ * " V S f S f a ~~U ^ a g V~~ h [S G

D Wh [e W V O 6 S f S e W` V T k G 3 D F f a B

' ž H [e g S ^ [I S f [a ` a ` 9 d S X S ` S

(Extra) BLE Beacon RSSI monitoring on Grafana

Ex1) Blink LED

5 a b k ^ b S e f W
B k f Z a ^ U a V W

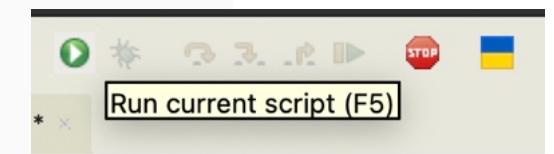
Thonny - <untitled> @ 12 : 39

```
from machine import Pin
import time

# Initialize pin 2 for the LED as an output
led = Pin(2, Pin.OUT)

# Main loop
while True:
    led.on() # Turn the LED on
    time.sleep(1) # Wait for 1 second
    led.off() # Turn the LED off
    time.sleep(1) # Wait for 1 second
```

F Z W` b g e Z
D g ^



4 ^ g W > 7 6 / b [` \$ fi T ^

F d k f a U Z S ^ Y W T ^ [`] [` f Wd h S ^ e
< g e f U Z S ^ Y W f [_ Wž e Ww b S ^



Tutorial list

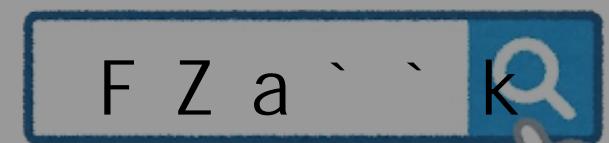
; ` e f d g U f [a ` ^ e S _ b ^ W b d a Y d
Z f f b , ! ! [a f i a d] ž a d '

B d Wb S d S f [a `

" ž ? [U d a B k f Z a ` [` e f S ^ ^ a ` 7 E B

7 j b Wd [_ W` f e

ž 4 ^ [`] > 7 6 a ` 7 E B % \$



\$ ž 5 Z WU] [` Y f Z W I [8 [X g ` U f [a ` a X E B

% ž I [d W 4 ? 7 \$ * " f a Y Wf V S f S

/ 4 ? 7 \$ * " ž b k ^ [T d S d k [` e f S ^ ^ S f [a

& ž G b ^ a S V 4 ? 7 \$ * " V S f S f a ~~U ^ a g V~~ h [S G

D Wh [e W V O 6 S f S e W` V T k G 3 D F f a B

' ž H [e g S ^ [I S f [a ` a ` 9 d S X S ` S

(Extra) BLE Beacon RSSI monitoring on Grafana

Ex2) Wi-Fi finder nearby

5 a b k ^ b S e f W
B k f Z a ^ U a V W

Thonny - <untitled> @ 16:1

```
import network
import time

# Initialize the WiFi interface in station mode
wlan = network.WLAN(network.STA_IF)

# Activate WiFi
wlan.active(True)

print("Nearby WiFi networks:")
for _ in range(5): # Repeat the scan 5 times
    networks = wlan.scan()
    for ssid, bssid, channel, rssi, authmode, hidden in networks:
        print(f"SSID: {ssid.decode()}, RSSI: {rssi} dBm")
    time.sleep(1) # Wait for 1 second
```

F Z W ^ b g e Z

D g ^

Shell

```
SSID: eduroam, RSSI: -72 dBm
SSID: , RSSI: -72 dBm
SSID: eduspot, RSSI: -78 dBm
SSID: eduroam, RSSI: -79 dBm
SSID: , RSSI: -79 dBm
SSID: eduspot, RSSI: -82 dBm
SSID: eduroam, RSSI: -82 dBm
SSID: , RSSI: -82 dBm
SSID: eduroam, RSSI: -84 dBm
SSID: eduspot, RSSI: -86 dBm
SSID: Alice Village by CA, RSSI: -87 dBm
SSID: showroom, RSSI: -89 dBm
SSID: eduspot, RSSI: -89 dBm
SSID: eduroam, RSSI: -89 dBm
SSID: , RSSI: -89 dBm
SSID: eduspot, RSSI: -90 dBm
```

@ WS d T k E E ; 6 D E E ; i [^ ^ e

MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbserial-0001 ≡

8 [` V [` Y E E ; 6 } < 8 I ? Ž I E

Tutorial list

; ` e f d g U f [a ` ^ e S _ b ^ W ' d
Z f f b , ! ! # (# ž % ž -



B d Wb S d S f [a `

? [U d a B k f Z a ` [` e f S ^ ^ a ` 7 E B %

7 j b Wd [_ W` f e

ž 4 ^ [`] > 7 6 a ` 7 E B % \$

\$ ž 5 Z WU] [` Y f Z W I [8 [X g ` U f [a ` a X

% ž I [d W 4 ? 7 \$ * " f a Y Wf V S f S

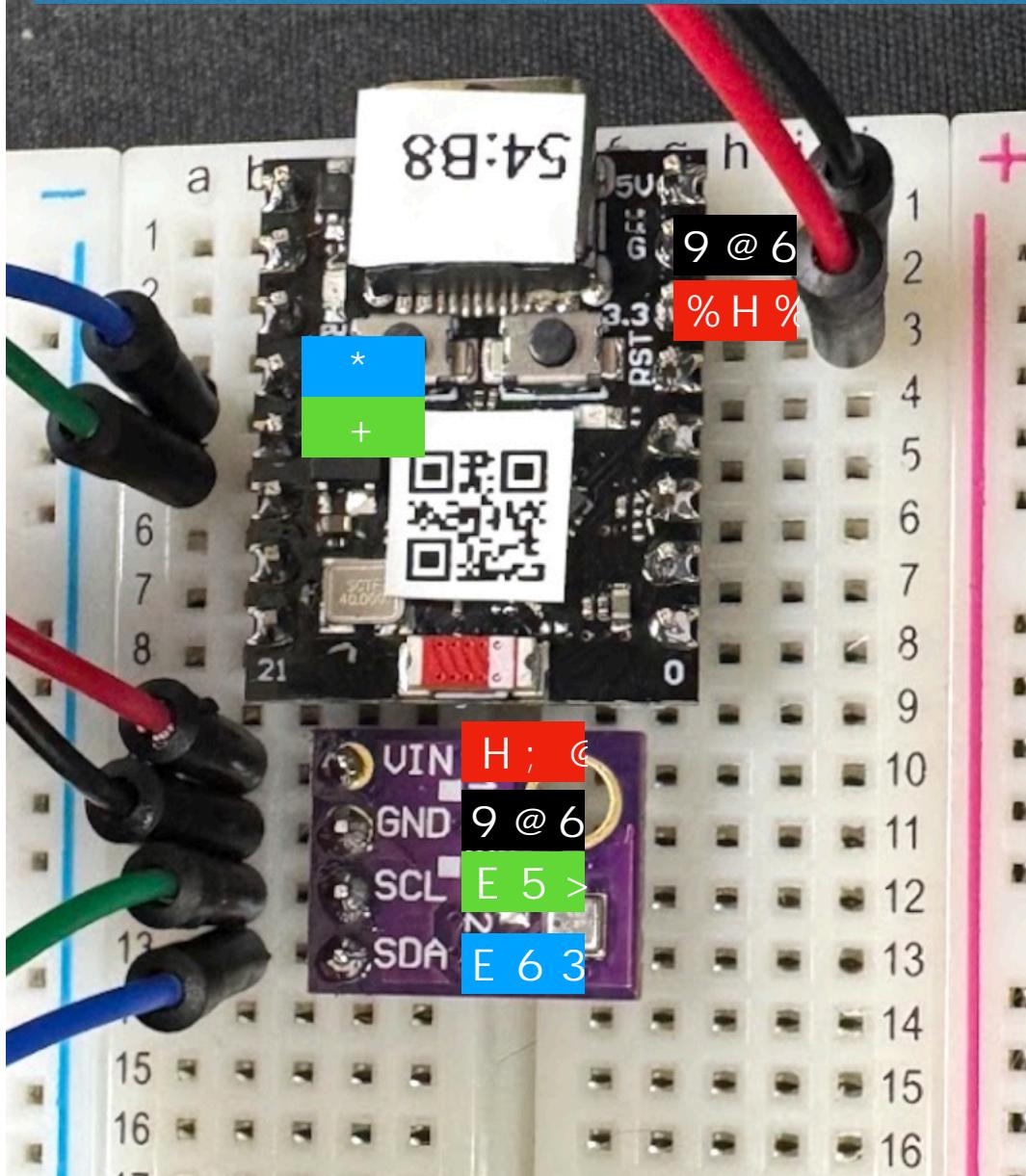
/ 4 ? 7 \$ * " ž b k ^ [T d S d k [` e f S ^ /

& ž G b ^ a S V 4 ? 7 \$ * " V S f S f a U ^ a g V h

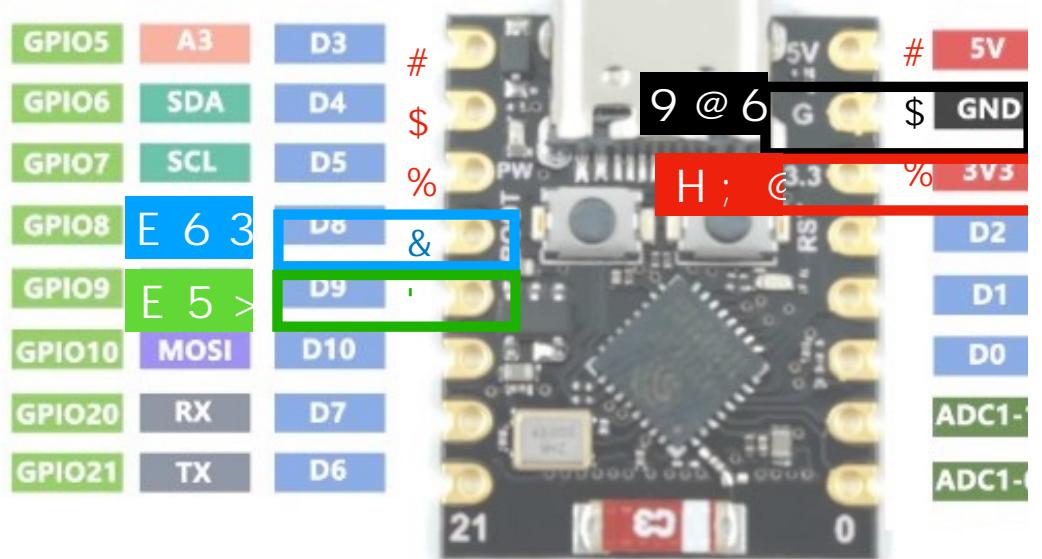
' ž H [e g S ^ [I S f [a ` a ` 9 d S X S ` S

(Extra) BLE Beacon RSSI monitoring on Grafana

Wire connection on breadboard



& i [d We X a d ; \$
U a ` ` W U f a `



Ex3) BME280 data get

5 a b k ^ b S e f W
B k f Z a ^ U a V W

Thonny - <untitled> @ 17 : 13

```
from time import sleep
import BME280

# BME280 setup
SCL = 22
SDA = 21
i2c = SoftI2C(scl=Pin(SCL), sda=Pin(SDA))
bme = BME280.BME280(i2c=i2c)

while True:
    temp = bme.temperature
    hum = bme.humidity
    press = bme.pressure
    print(f"Temperature: {temp}°C, Humidity: {hum}%, Pressure: {press}hPa")
    sleep(1)
```

F Z W^ b g e Z

D g ^

Shell

```
>>> %Run -c $EDITOR_CONTENT
K a g i [ ^ ^ Y Wf Wd d a d
` a _ a V g ^ W ` S _ WV
@ WWV f a [ ` e f S ^ ^ s
Z f f b , ! ! [ a f i a d ] ž a d Y ! e _ U ^ [ b ž T a S d V
```

Plotter visualizes series of numbers printed to the Shell.

See Help for details.

MPY: soft reboot

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

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 # Trimming
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

Ex3) BME280 data get

fi 5 d WS f W ` Wi WV [f a d i [` V a i

\$ fi 5 a b k ^ b S e f W s 4

The screenshot shows a Mac OS X desktop with two terminal windows and a file browser.

The left terminal window shows a Python script for the BME280 sensor:

```
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 # Trimming parameter registers
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
BME280_REGISTER_DIG_P9 = 0x9E
```

The right terminal window shows a file selection dialog with "device" highlighted.

E S h W [f

& fi E f W f W ` S _ W s 4 ? 7 \$ * " ž b k q

E S h W O MA = O

File name: BME280.py

Ex3) Get BME280 data

B d We e [` [f [S ^ f S T / G ` f [f ^ WV ž ž fi

F Z W` b g e Z

D g `

Shell ×

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

MPY: soft reboot

Temperature: 22.13°C, Humidity: 83.26%, Pressure: 691.48hPa

Temperature: 24.99°C, Humidity: 32.04%, Pressure: 953.16hPa

Temperature: 24.98°C, Humidity: 32.03%, Pressure: 953.11hPa

Temperature: 24.98°C, Humidity: 32.03%, Pressure: 953.19hPa

Temperature: 24.98°C, Humidity: 32.02%, Pressure: 953.19hPa

Temperature: 24.98°C, Humidity: 32.01%, Pressure: 953.14hPa

Temperature: 24.99°C, Humidity: 32.02%, Pressure: 953.14hPa

X da 4 ? 7 \$ * " e

-500 -- Temperature: ●°C, Humidity: ●%, Pressure: ●.14hPa -- 0 -- 500 -- 1000 --

K a g i [^ ^ Y Wf V S f S

MicroPython (ESP32) · CP2102 USB to UART Bridge Controller @ /dev/cu.usbserial-0001 ≡

Ex4) BME280 data upload to InfluxDB

5 a b k ^ b S e f W

F Z W ^ b g e Z

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)
```

Shell

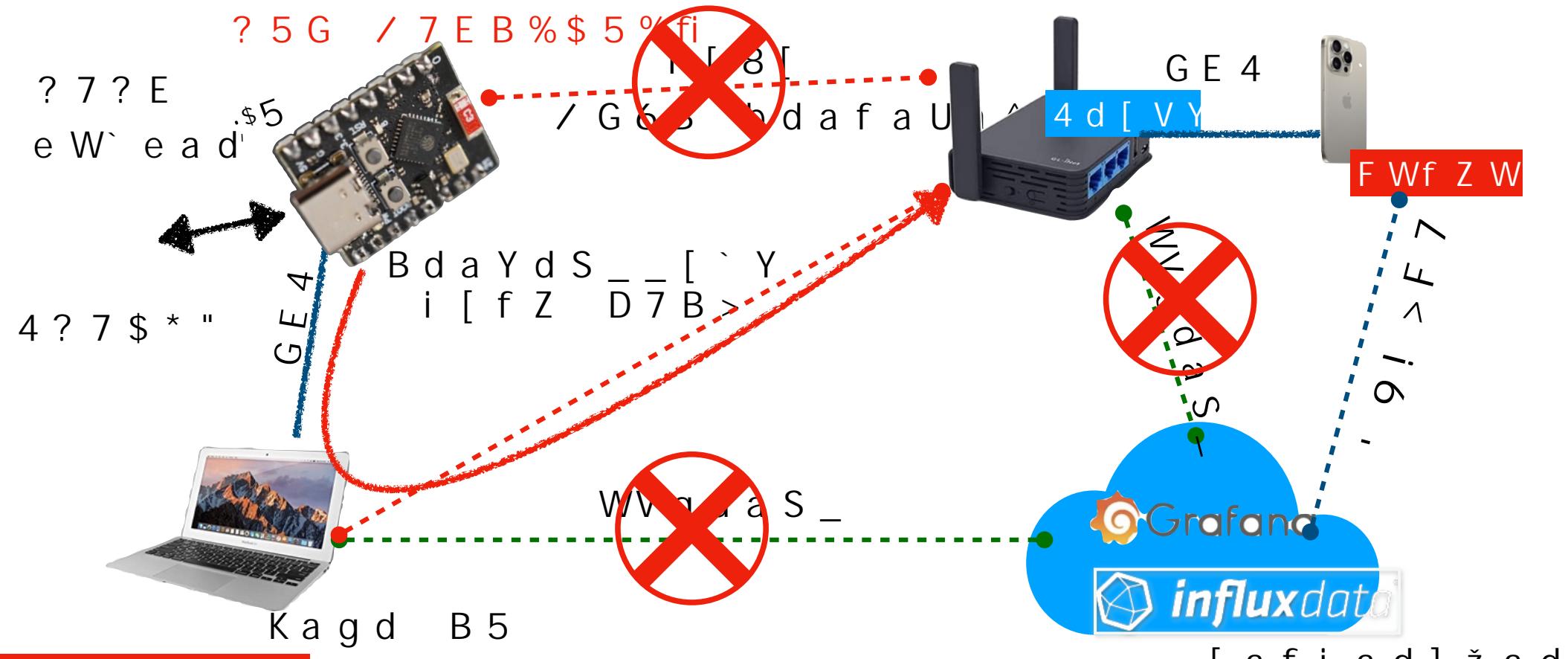
```
Waiting for Wi-Fi connection...
```

D g ^

H [S f Č F 7 / F W f Z W d [^ Y fi U a
E E ; 6 , < 8 I ? Č I E + B S e e ,
Plotter visualizes series of numbers printed to the Shell.
See Help for details.

5 S ^ a f i a d] T W
g ^ e f S T ^ W I [

Overview of connections (revised)



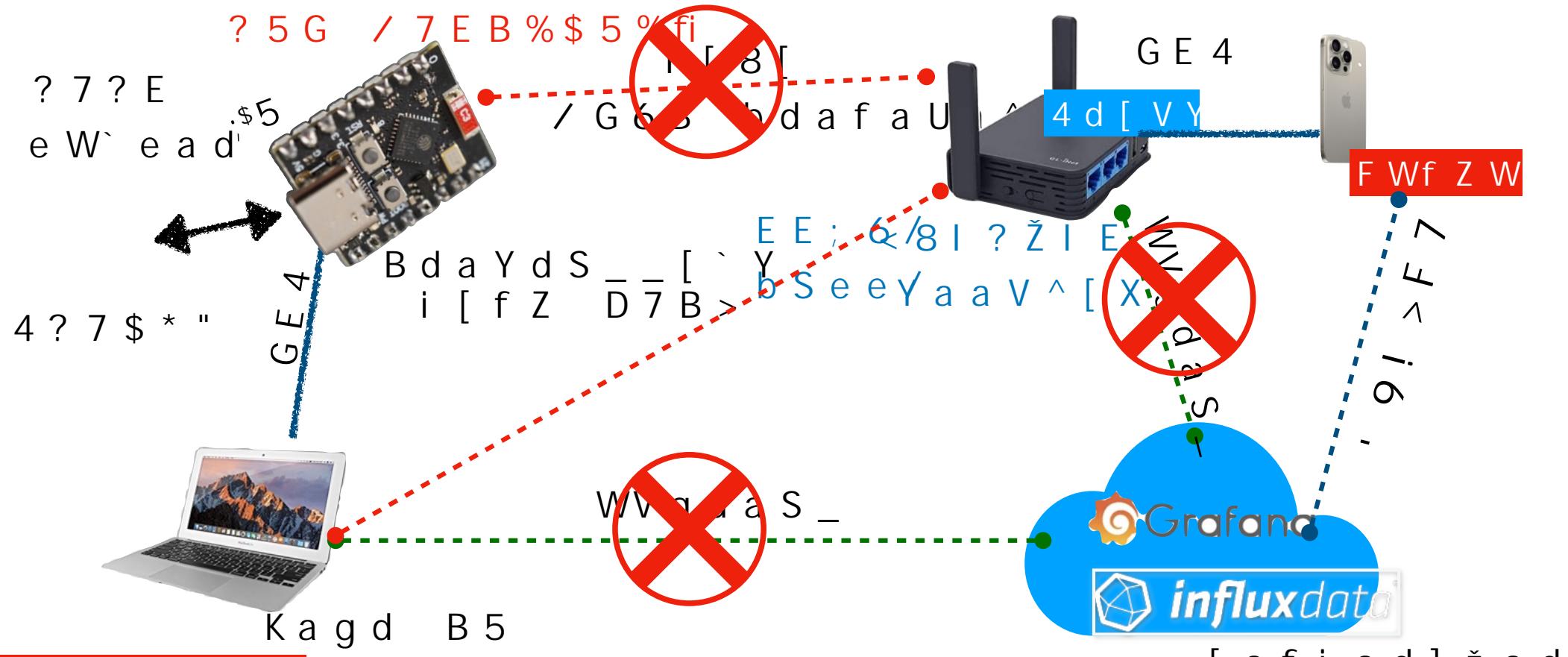
D Wh [e V

4 WU S g e W a X f Z W ^ a i c g S ^ [f k I [8 [a
U a ` ` WU f [a ` g ` e f S T ^ W

7 E B % \$ 5 % Y Wf V S f S X d a _ 4 ? 7 \$ * " 0 e W

7 E B % \$ 5 % a b Wd S f W i [f Z e f S ` V S ^ a ` \

Overview of connections (revised)



D Wh [e V

4 W U S g e W a X f Z W ^ a i c g S ^ [f k I [8 [a
U a ` ` W U f [a ` g ` e f S T ^ W

7 E B % \$ 5 % Y Wf V S f S X d a _ 4 ? 7 \$ * " O e W

7 E B % \$ 5 % a b Wd S f W i [f z e f S ` v S ^ a ` \

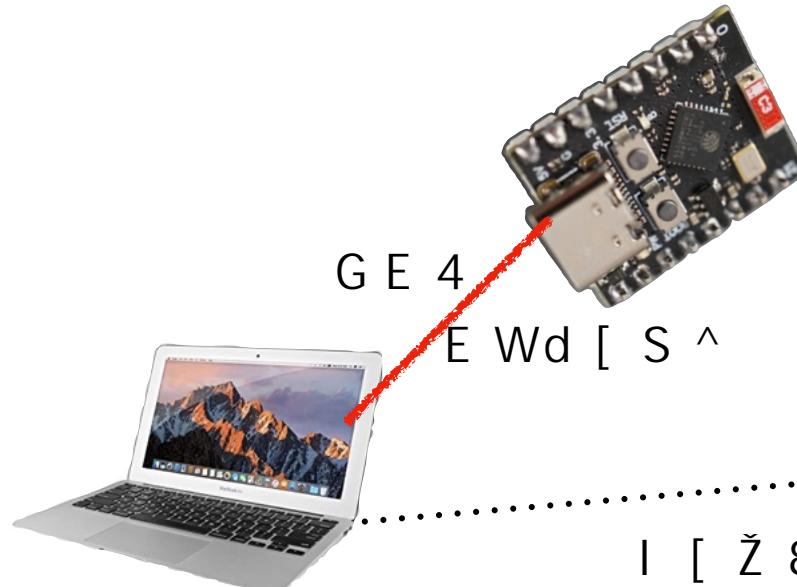
F i a b k f Z a ` e U d [b f e S h W V [

_ S [` ž b k / 4 a a f S ` V d g `

_ S [` Q b k Ž 4 ? 7 \$ * " Ž E Wd [

4 ? 7 \$ * " ž b k

4 ? 7 \$ * " Q b k Ž 7 E B



G 6 B e W` V e U d [b f
a ` F Z a ` ` k B 5

I [Ž 8]

E Wd [S ^ Ź G 6 B Ź g b Ź F Ź

A black, compact dual-band WiFi router with two external antennas and three Ethernet ports.

F Wf Z W

9! > F₇

GE 4

A close-up view of the triple-camera system on the back of an iPhone 14 Pro Max, showing three lenses and a flash.

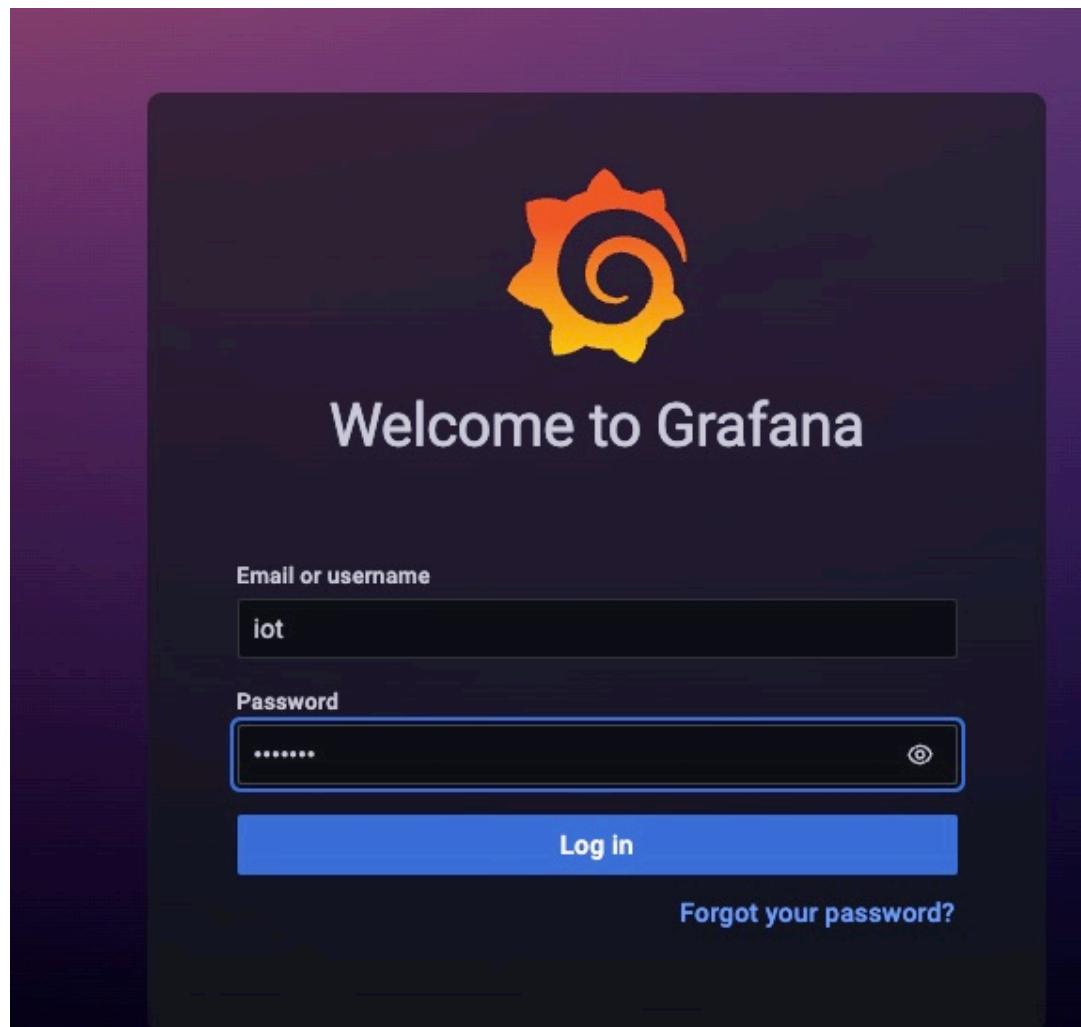
9! > F 7
5.....^ a g |

```
[ main.py ] [ BME280.py ] <untitled> * <untitled> * E Wd [ S ^ Z G 6 B Z g b Z F Z

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
```

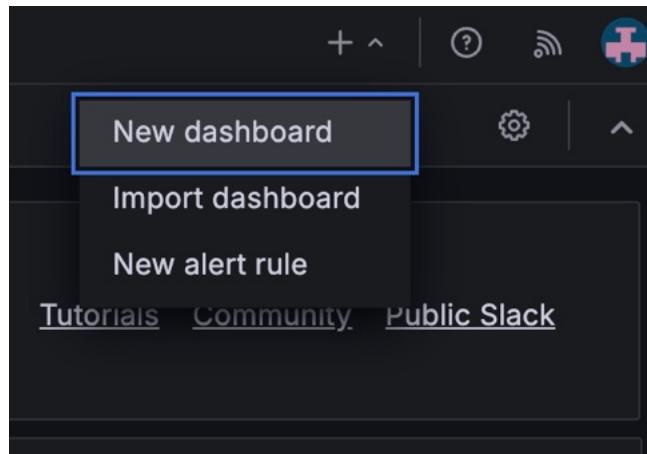
Data visualization on Grafana

3 U U We e Y d S X S ` S e Wd h Wd a ` U ^ a g V h [S
/ E E ; 6 , < 8 I ? Ź I E Ł b S e e , Y a a V ^ [X W f i
Z f f b , ! ! [a f i a d] ž a d Y , % " " " !

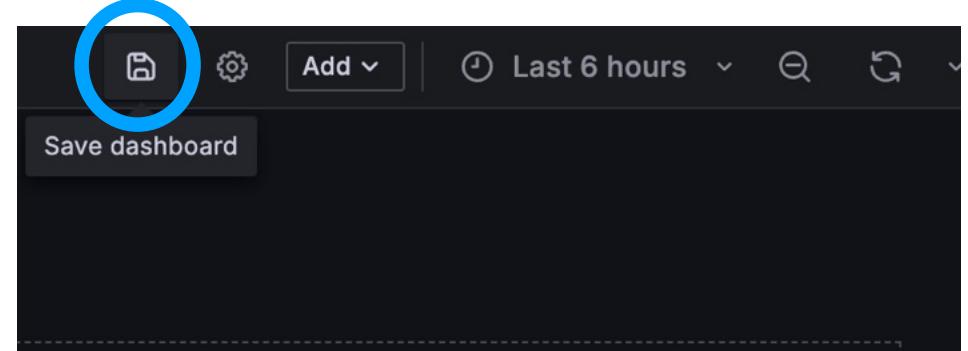


g e Wd , [a _
b S e € a _ f \$ "

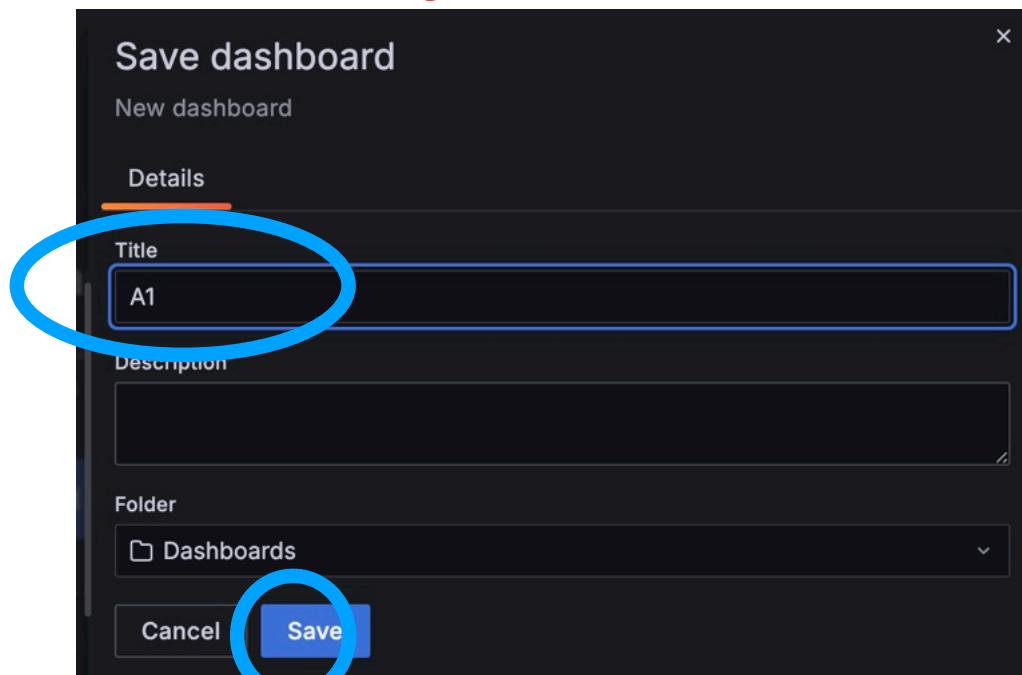
/ # fi U d WS f W } 6 S e Z T a S d V ' X d a _ Ł



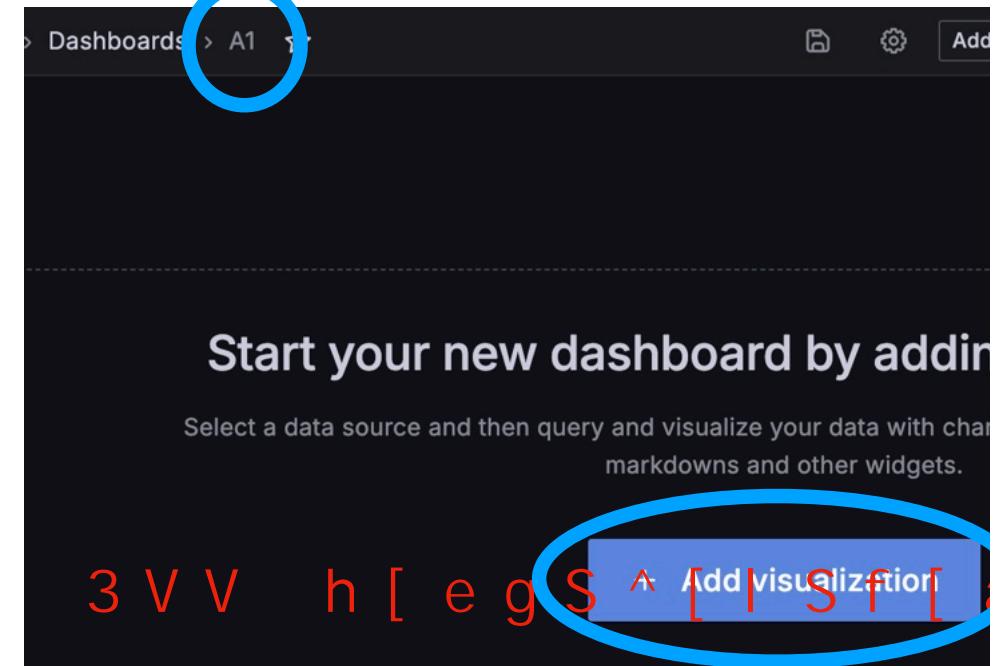
/ \$ fi E S h W



/ % fi ; ` b g f f [f ^ W S e k a g d Y d a g b 3 # S t



5 a `fd_ k a g d Y d a g b



/ & fi E S h W

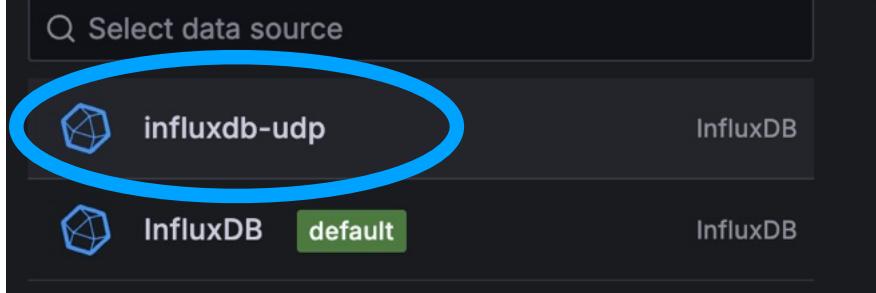
/ ' fi

3 V V h [e g S ^ [I S F [

/ (fi E W^ WU f V S f S

e a g d U W
6 S f S X da

7 E B % \$ F Z a
e f a d WV „ g J V T Ž g V b



Q Search or

Home > Dashboards > A1 > Edit panel

Table view Fill Actual

⚠ Panel Title

No data

Query 1 Transform data 0 Alert 0

Data source influxdb-udp Query ... MD = auto = 814

A (influxdb-udp)

FROM default select measurement WHERE +

SELECT field(value) × mean() × +

Query 1

Transform data 0

Alert 0

Data source influxdb-udp

Query ... MD = auto = 814 Interval = 30s Query inspector

A (influxdb-udp)

FROM default select measurement WHERE +

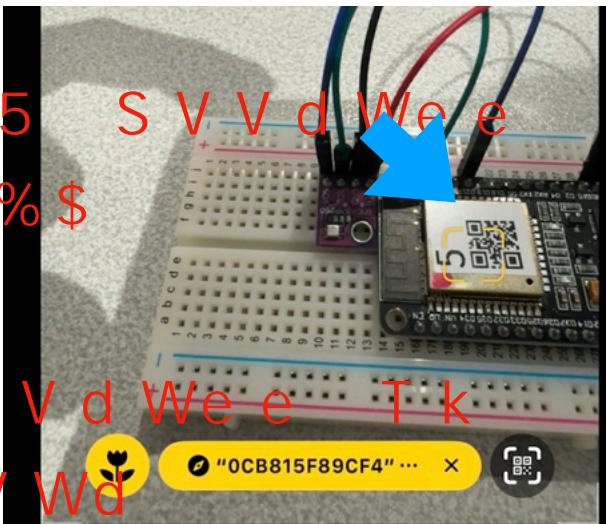
SELECT field(value) × mean() × +

/ * fi E W^ WU f _ WS e g d W_ W` f

ble_rssi_0CB815F350B0
ble_rssi_0CB815F89CF4
Choose WHE

8 [` V [` Y T ^ WQ d e e [Q 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 [e ? 3 5 S V V d W e e
a X k a g d 7 E B % \$



K a g U S ` Y W f S V V d W e e T k
C D U a V W d W S V W d

A d ^ S T W ^ ^ W V ` (7 E B % \$ S ` V S V V

E W^ WU f k a g d Y d a g b q e S V V d W e e

Query 1

Transform data 0

Alert 0

Data source influxdb-udp

Query ... MD = auto = 814 Interval = 30s Query inspector

A (influxdb-udp)

FROM default select measurement WHERE +

SELECT field(value) × mean() × +

/ + fi E WfWWWWf / h S ^ g Wfi

Transform data 0 Alert 0

default ble_rssi_0CB815F350B0 × w

field Choose × me

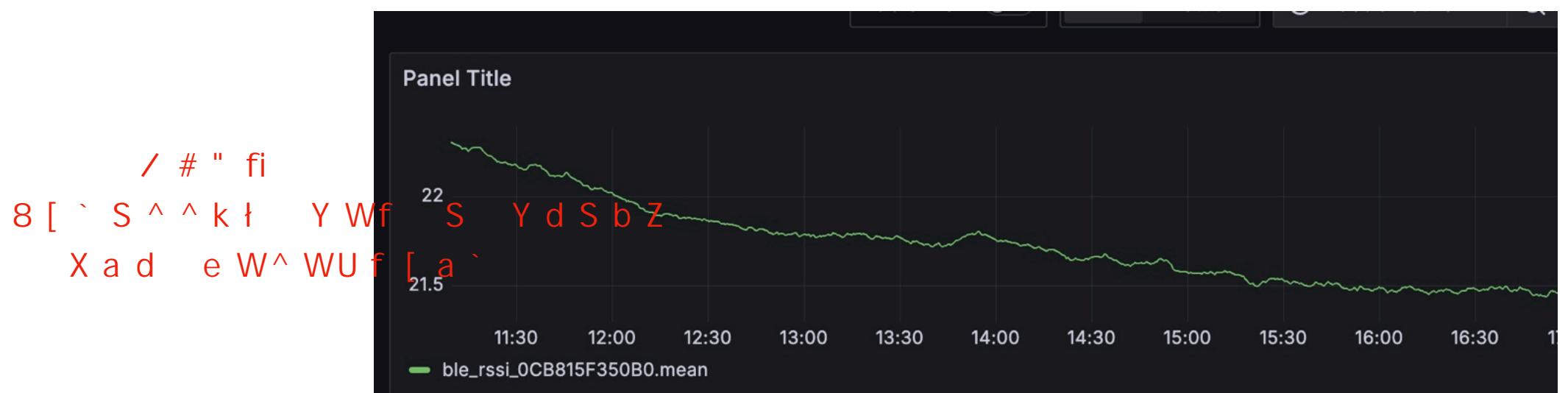
time(hum) × +

(optional) press TIME asc

(optional) temp (optional)

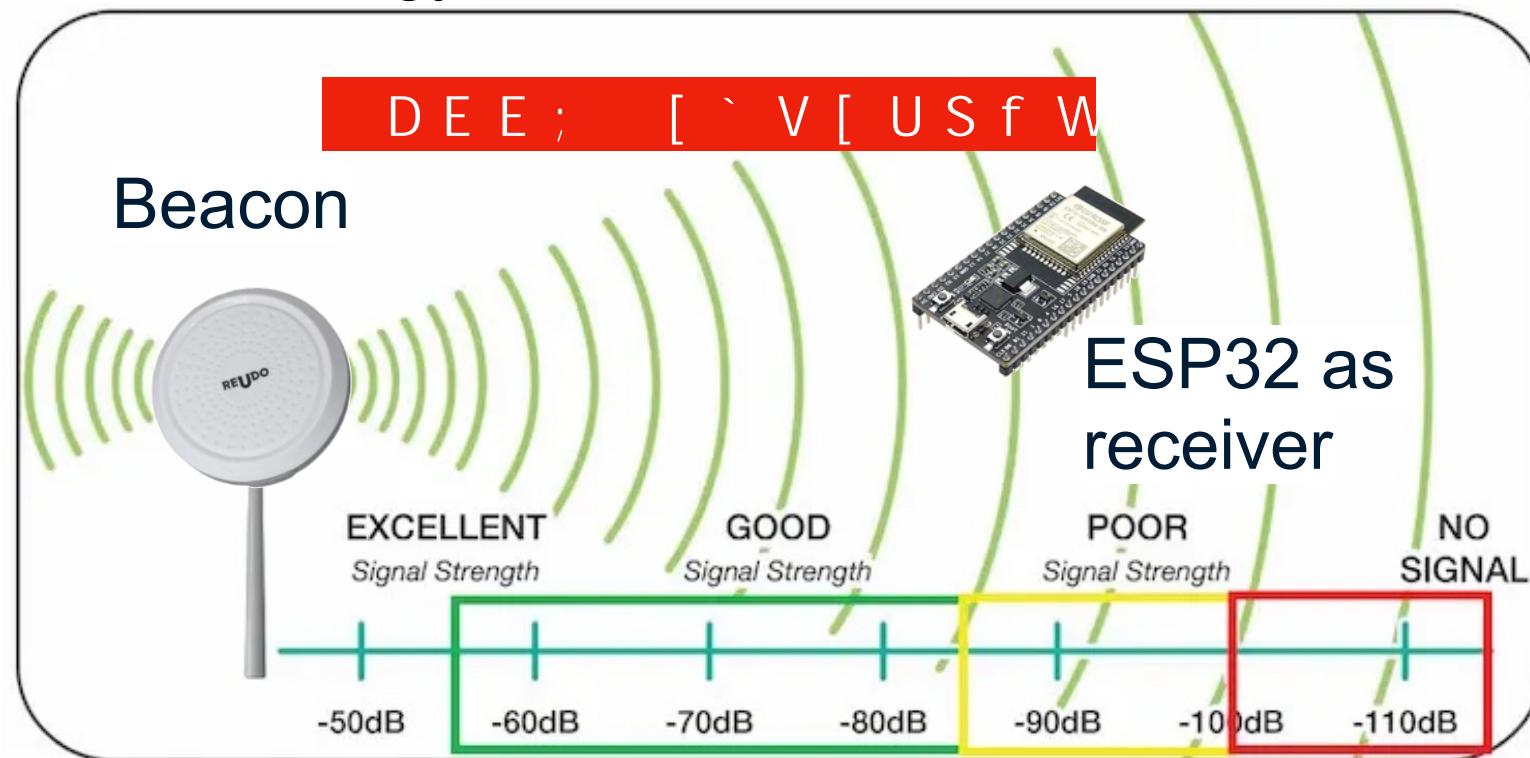
K a g U S ^ U Z a a
Z g _ , : g _ [V [f
b d We e , B d We e
f W_ b , F W_ b Wd S f g d W

6 Wf S [^
T W Wj b ^
a ` e [



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



B d We e S ` V Z a ^ V f z W T g f f a S ` V E I > 7 6, S e Z W

Opt) Beacon RSSI upload InfluxDB

5 a b k ^ b S e f W

F Z W ^ b g e Z

Thonny - <untitled> @ 17 : 6

```
<untitled> * [ BME280.py ] *
```

```
7
8 SSID = "Duplex loft"
9 PASSWORD = "DUPLEX_LOFT"
10
11 # InfluxDB server details
12 INFLUXDB_IP = "161.34.0.113"
13 INFLUXDB_PORT = 8089
14
15 # Target MAC addresses (add or remove as needed)
16 TARGET_MACS = [
17     "ce3f413259a9", # Fujita's Mi-band
18 ]
19 print("Processing target MAC addresses...")
20 target_macs = [mac.replace(':', '').lower() for mac in TARGET_MACS]
21 print(f"Loaded {len(target_macs)} target MAC addresses")
22
23 print("Initializing BLE...")
24 ble = BLE()
25 ble.active(True)
26 print("BLE initialized")
27
28 print(f"Connecting to Wi-Fi network: {SSID}")
```

Shell

```
Target device found. MAC: ce3f413259a9, RSSI: -70
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-51,ce3f413259a9=-70
Target device found. MAC: ce3f413259a9, RSSI: -76
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-49,ce3f413259a9=-76
Target device found. MAC: ce3f413259a9, RSSI: -71
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-52,ce3f413259a9=-71
Target device found. MAC: ce3f413259a9, RSSI: -70
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-51,ce3f413259a9=-70
Target device found. MAC: ce3f413259a9, RSSI: -69
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-51,ce3f413259a9=-69
Target device found. MAC: ce3f413259a9, RSSI: -71
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-50,ce3f413259a9=-71
Target device found. MAC: ce3f413259a9, RSSI: -70
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-52,ce3f413259a9=-70
Target device found. MAC: ce3f413259a9, RSSI: -71
Sent to InfluxDB: ble_rssi_0CB815F89CF4 Uplink_RSSI=-52,ce3f413259a9=-71
```



MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ /dev/cu.usbserial-0001

D E E ; h S ^ g We f X a d W S b W U [S V V d We e

S ^ V g b ^ [^] f a 3 B i [^ ^ T W g b