




Python for IoT

By Aldion Amirrul

 github.com/aldamr01

 linkedin.com/in/aldionamirrul



surabaya.py

Our Agenda

1. Introduction : Python programming Language
2. About Internet of Things
3. Python for Internet of Things
4. QnA Session

Introduction : Python Programming Language



It's rewind time

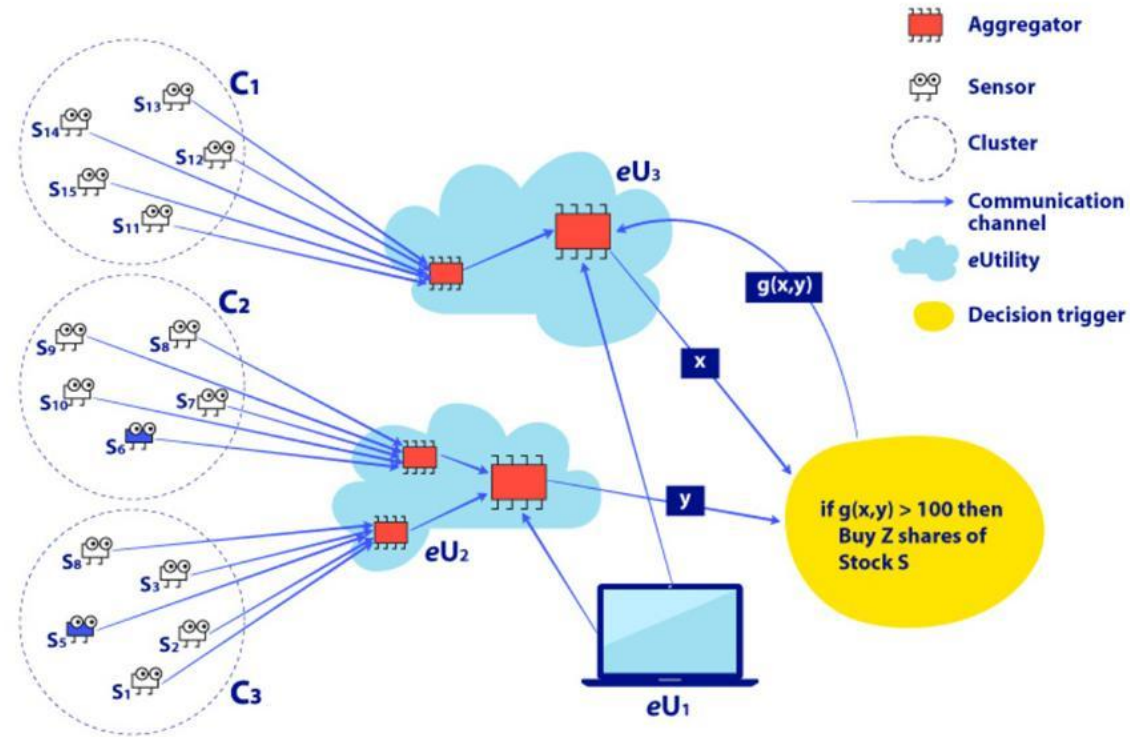
All was explained at the previous session

About Internet of Things

Internet of Things ?

The Internet of Things (IoT) describes the network of physical objects “things” that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

Source : Oracle

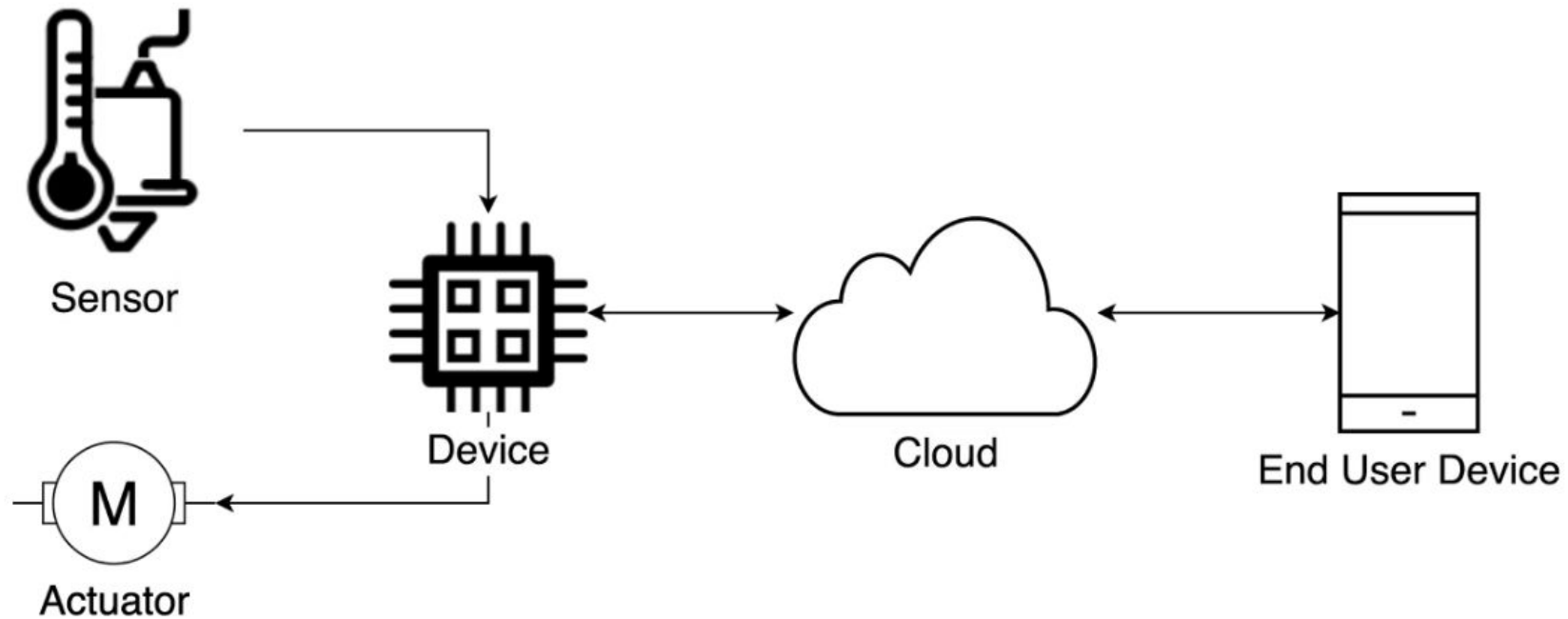


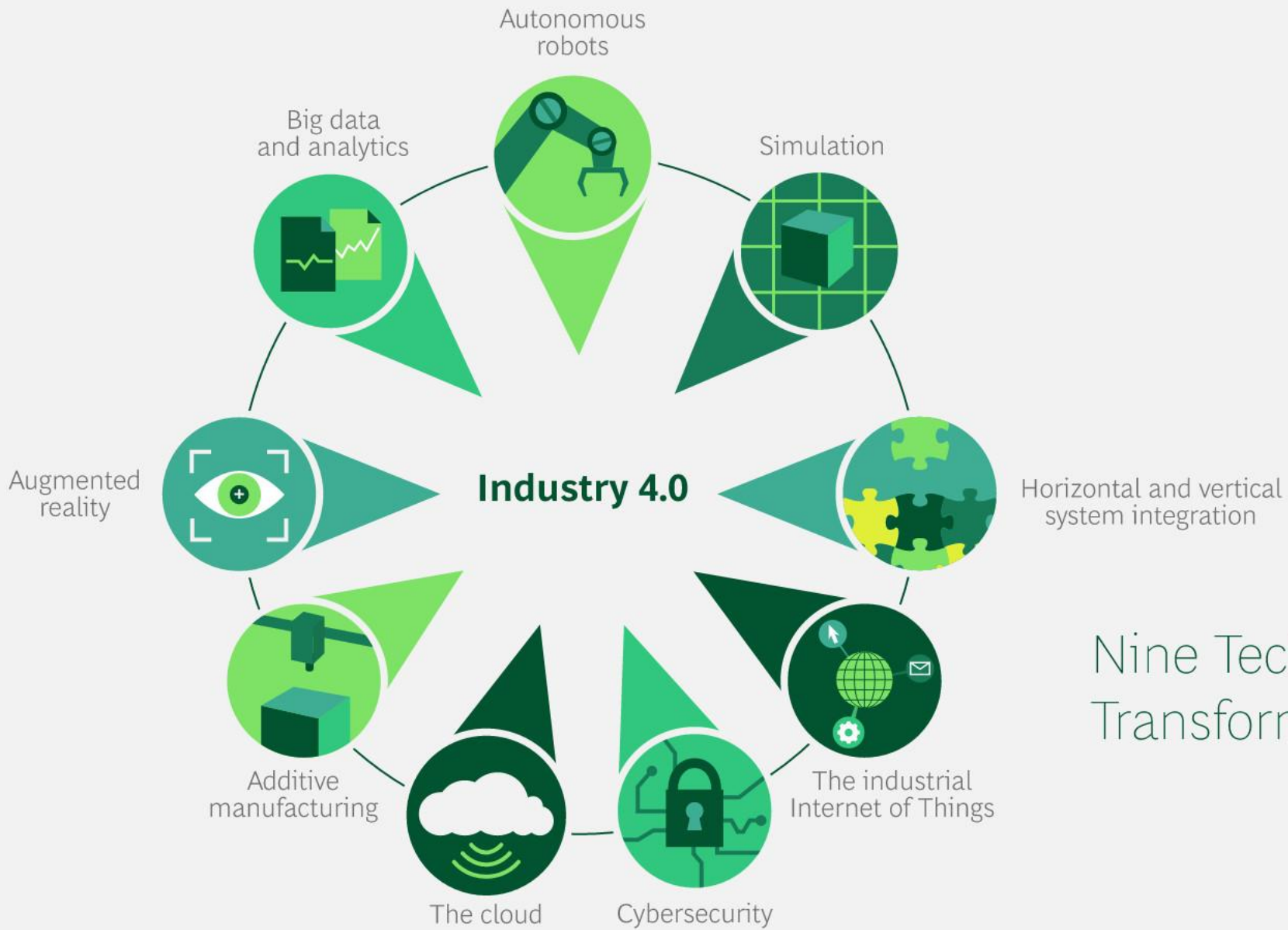
Taken from NIST-SP 800 – 183 Network if Things

INTERNET OF THINGS

Yes it is

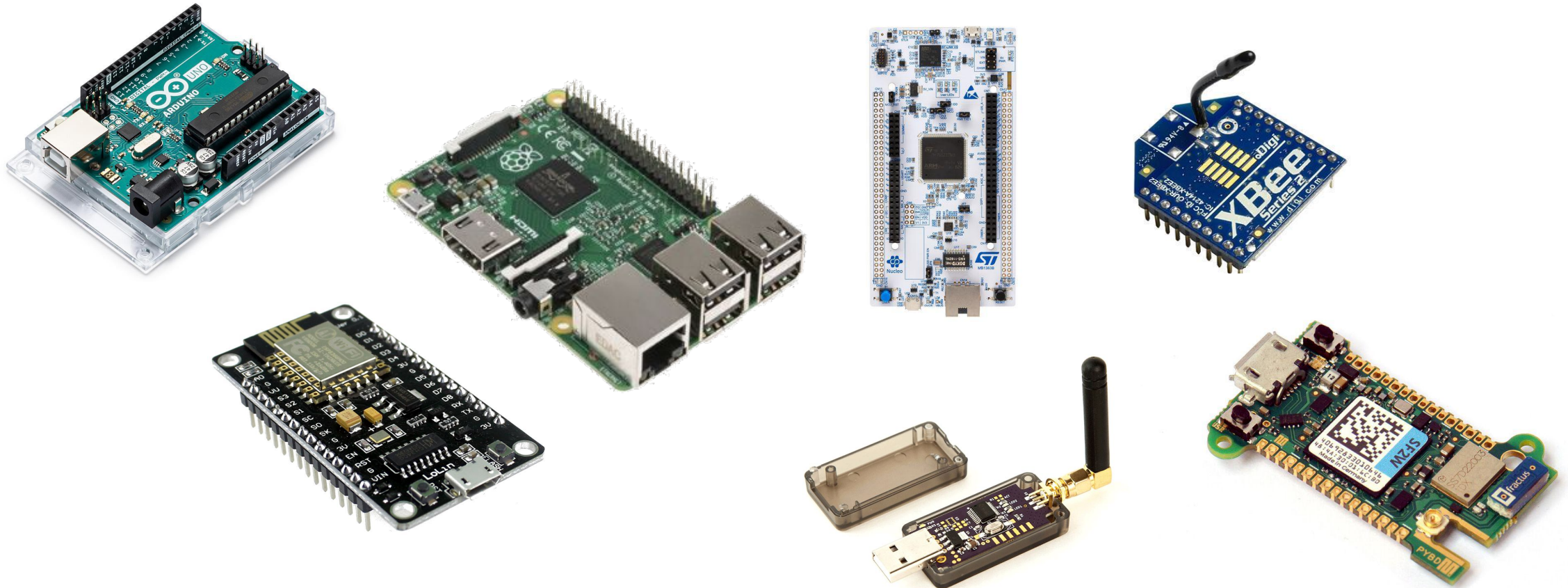
IoT at Glance - It's all about digitalization



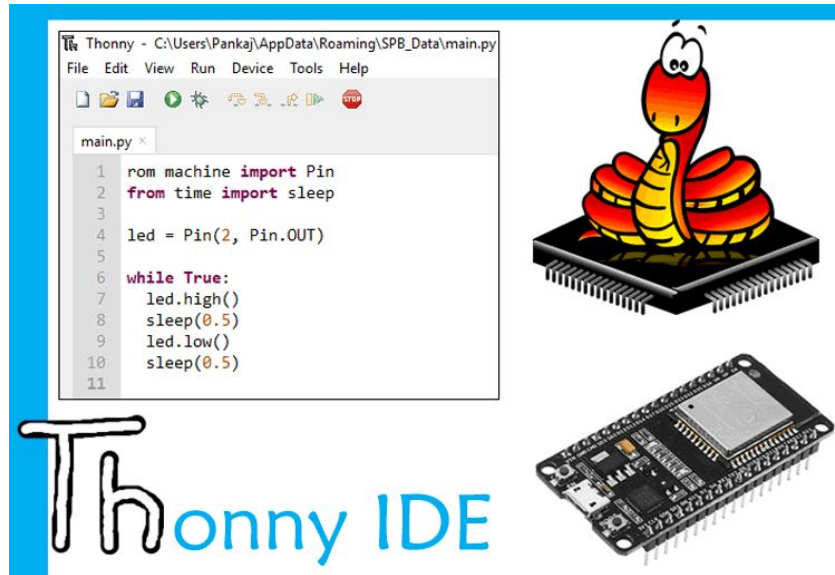


Nine Technologies Are Transforming Industrial Production

IoT Device development boards



IoT Development IDEs

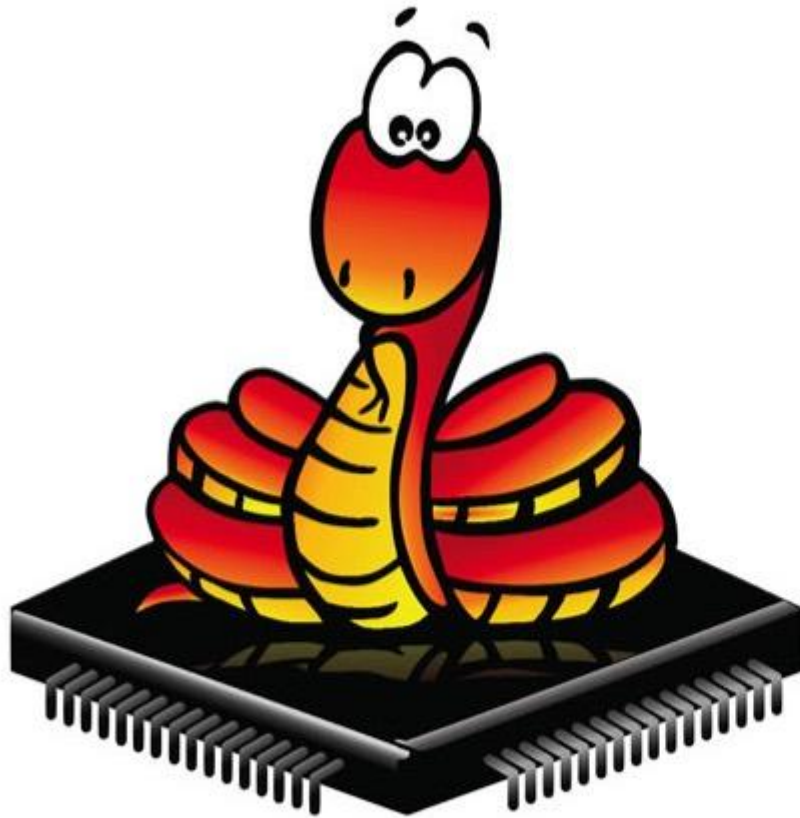


IoT Use Cases



Python for IoT

Introduction of Micropython



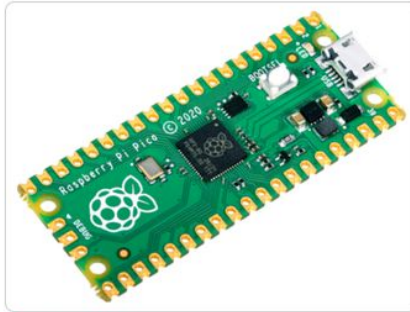
MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised to run on microcontrollers and in constrained environments.

Micropython

PROS (+)	CONS (-)
Human-Readable Language	Slow than C / C++
Built-In Exception- and Error-Handling	Not all boards compatible with Micropython
Open Source and Free	Little bit Ruwet
Object Oriented Language	
Abstracts Out the Hardware Layer	

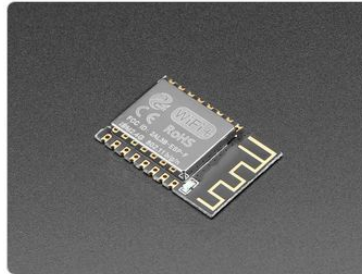
Micropython Supported Boards

Raspberry Pi RP2040 microcontroller boards



Raspberry Pi Pico

Espressif ESP-based boards



Generic ESP8266 module



Generic ESP32

STM32 boards

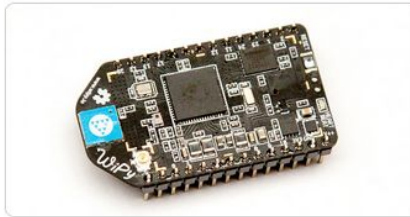


STM32 Nucleo and Discovery boards

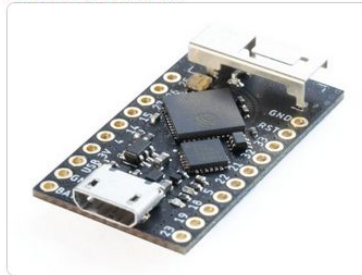


Espruino Pico

TI CC3200 boards

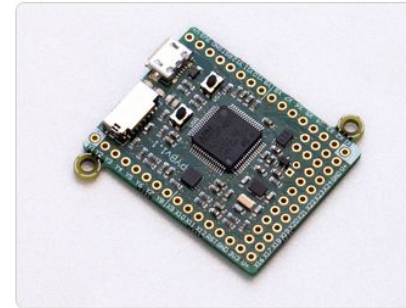


WiPy module



TinyPICO

Pyboard v1 and D-series

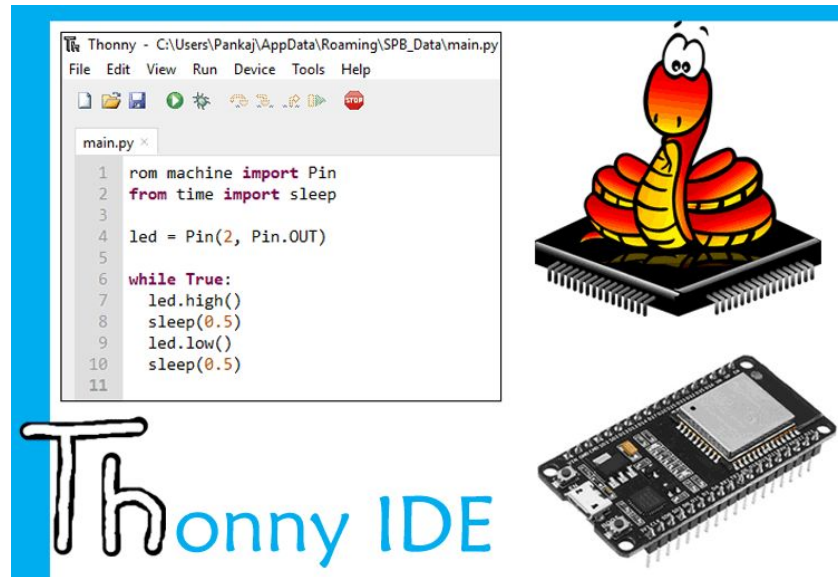


pyboard v1



pyboard D-series

Suggested IDEs



Let's Try

Requirements

1. Install Python3.

(<https://www.python.org/downloads/>)

2. Install python3 esptool.

(<https://github.com/espressif/esptool>)

3. Install Thonny.

(<https://thonny.org/>)

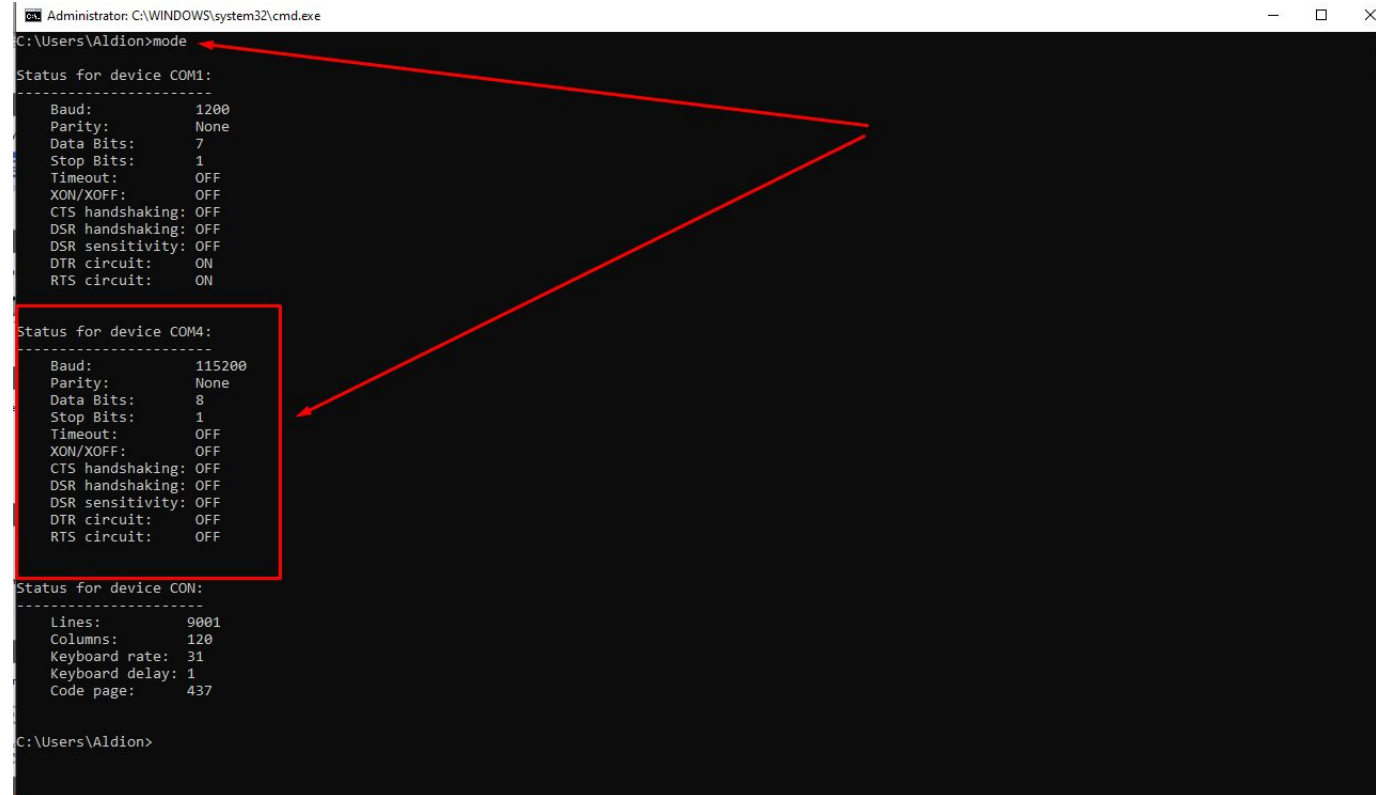
4. Install driver CH340 / CP210 drivers.

5. Download your board firmware.

(<https://micropython.org/download/>)

Installing Board's Firmware

1. Check your device port address :



```
Administrator: C:\WINDOWS\system32\cmd.exe
C:\Users\Aldion>mode

Status for device COM1:
-----
Baud:          1200
Parity:        None
Data Bits:     7
Stop Bits:     1
Timeout:       OFF
XON/XOFF:      OFF
CTS handshaking: OFF
DSR handshaking: OFF
DSR sensitivity: OFF
DTR circuit:   ON
RTS circuit:   ON

Status for device COM4:
-----
Baud:          115200
Parity:        None
Data Bits:     8
Stop Bits:     1
Timeout:       OFF
XON/XOFF:      OFF
CTS handshaking: OFF
DSR handshaking: OFF
DSR sensitivity: OFF
DTR circuit:   OFF
RTS circuit:   OFF

Status for device CON:
-----
Lines:         9001
Columns:       120
Keyboard rate:  31
Keyboard delay: 1
Code page:     437

C:\Users\Aldion>
```

Installing Board's Firmware

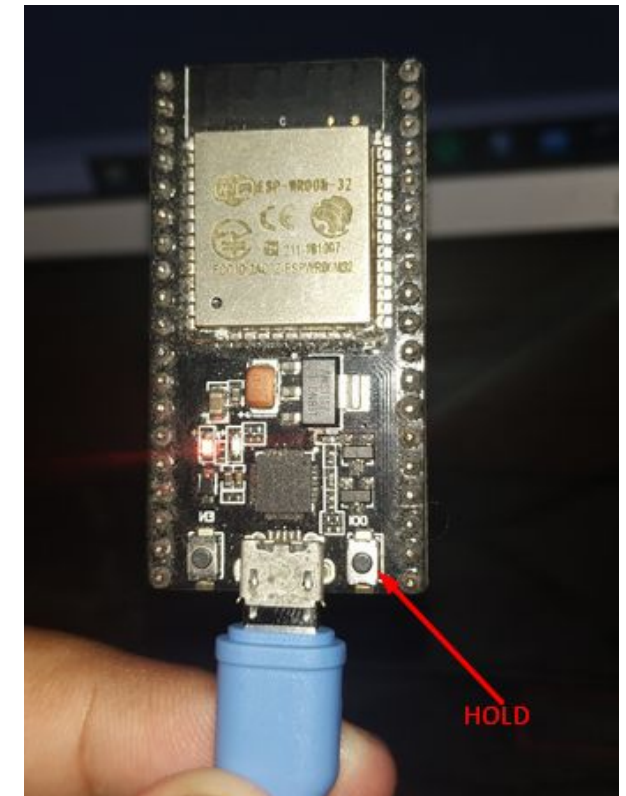
2. Erase device flash memory :

```
Administrator: C:\WINDOWS\system32\cmd.exe

C:\Users\Aldion>python -m esptool --chip esp32 --port COM4 erase_flash
esptool.py v3.0
Serial port COM4
Connecting.....
Chip is ESP32-D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 3c:71:bf:88:a1:a4
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 7.9s
Hard resetting via RTS pin...

C:\Users\Aldion>
```

+



Installing Board's Firmware


3. Install device's firmware :

```
Administrator: C:\WINDOWS\system32\cmd.exe

C:\Users\Aldion>python -m esptool --chip esp32 --port COM4 write_flash -z 0x1000 esp32-20210306-unstable-v1.14-83-g680ce4532.bin
esptool.py v3.0
Serial port COM4
Connecting.....
Chip is ESP32-D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 3c:71:bf:88:a1:a4
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Compressed 1446144 bytes to 941180...
Wrote 1446144 bytes (941180 compressed) at 0x00001000 in 85.1 seconds (effective 136.0 kbit/s)...
Hash of data verified.

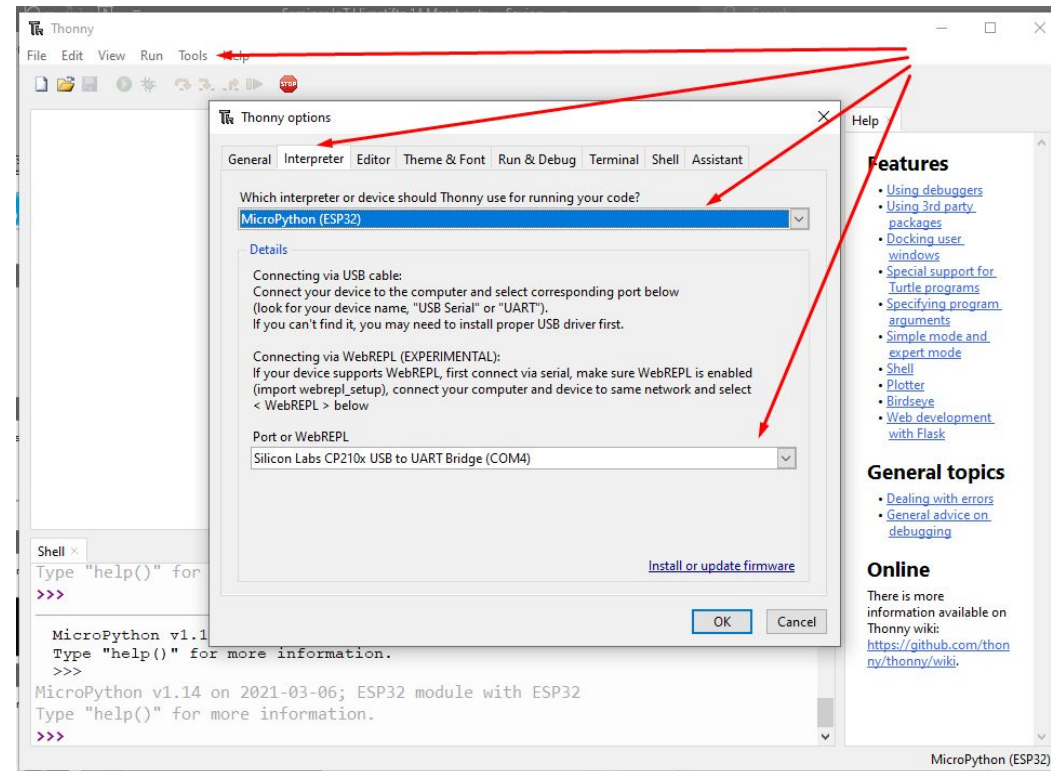
Leaving...
Hard resetting via RTS pin...

C:\Users\Aldion>
```



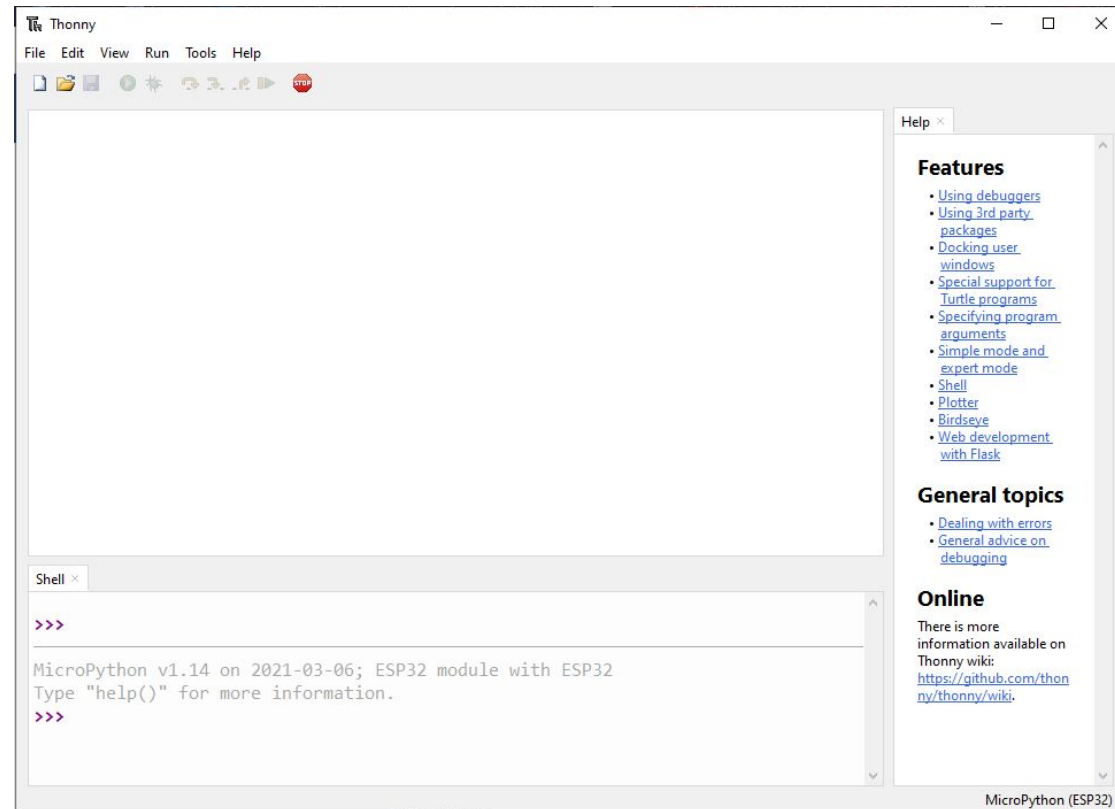
Code ESP32 with python

1. Open Thonny and set the configuration for ESP32 Device :



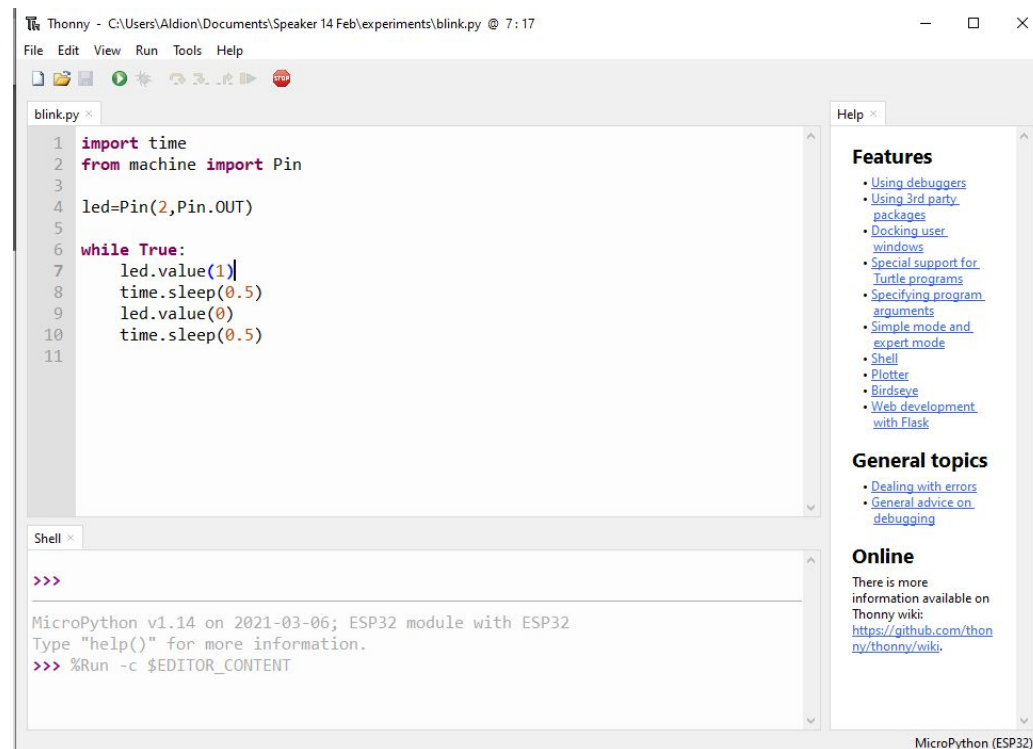
Code ESP32 with python

2. Here is your development environment :



Code ESP32 with python

3. Try make Blinking LED program and Run it :



```
Thonny - C:\Users\Aldion\Documents\Speaker 14 Feb\experiments\blink.py @ 7:17
File Edit View Run Tools Help

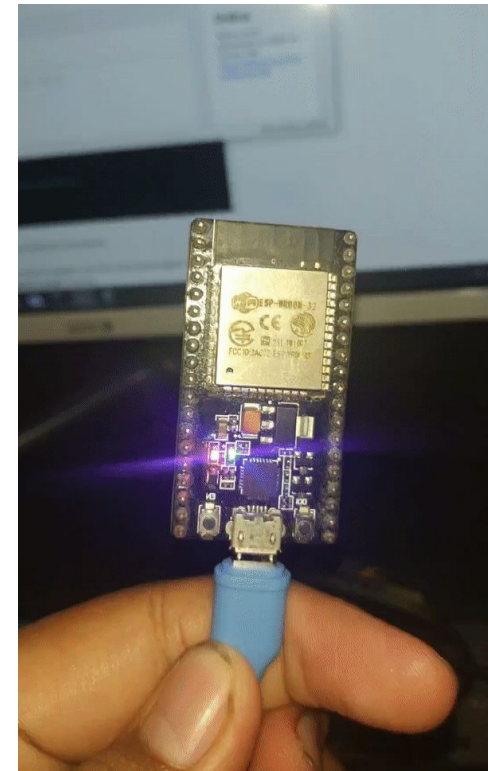
blink.py
1 import time
2 from machine import Pin
3
4 led=Pin(2,Pin.OUT)
5
6 while True:
7     led.value(1)
8     time.sleep(0.5)
9     led.value(0)
10    time.sleep(0.5)
11

Shell
>>>

MicroPython v1.14 on 2021-03-06; ESP32 module with ESP32
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

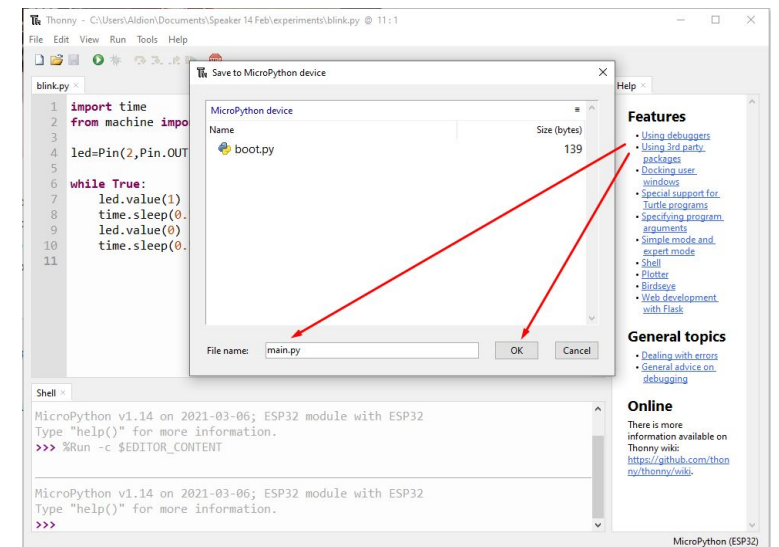
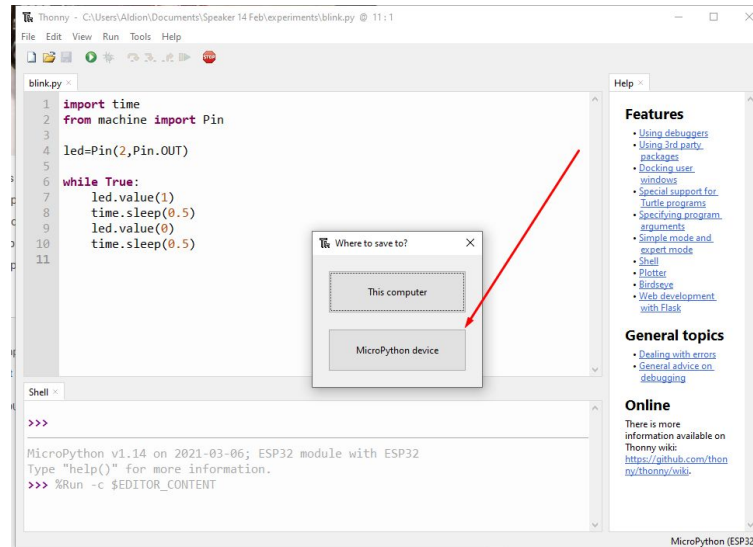
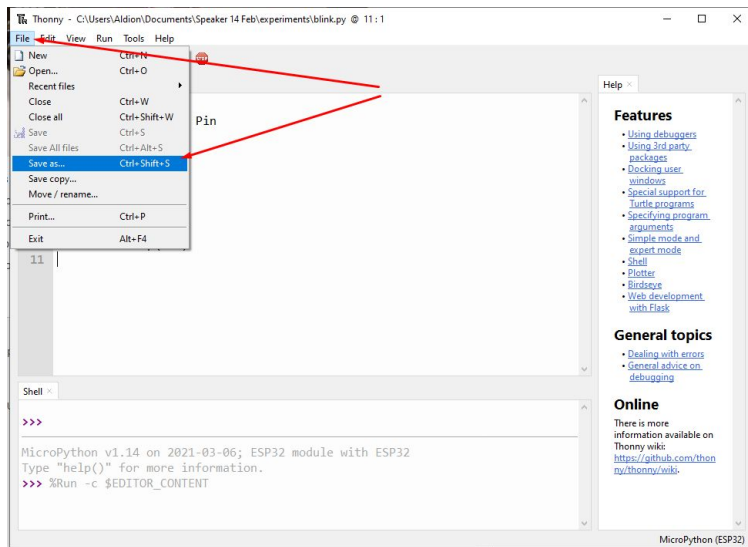
Help
Features
• Using debuggers
• Using 3rd party packages
• Docking user windows
• Special support for Turtle programs
• Specifying program arguments
• Simple mode and expert mode
• Shell
• Plotter
• Birdseye
• Web development with Flask
General topics
• Dealing with errors
• General advice on debugging
Online
There is more information available on Thonny wiki:
https://github.com/thonny/thonny/wiki.
```

->



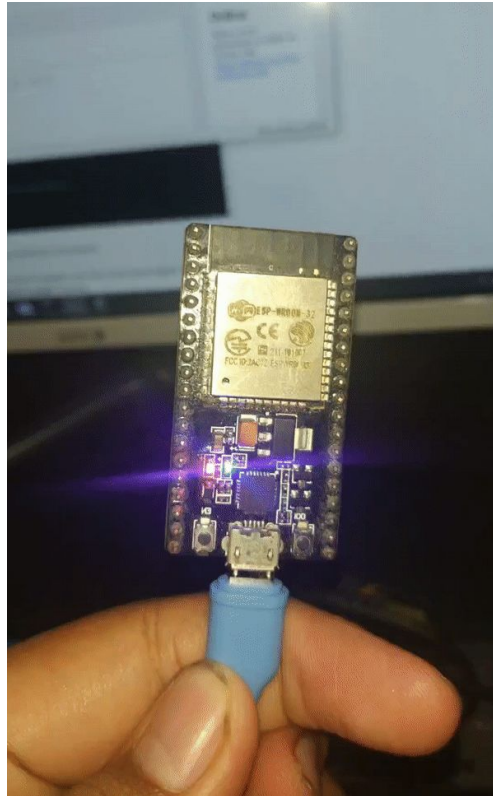
Code ESP32 with python

3b. Try make Blinking LED program and upload it :



Code ESP32 with python

4. And this is it !



Interesting right ?

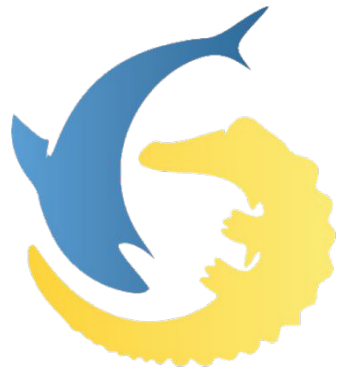
Yep, come, let's try it !



But don't forget to learn the basics first !



Let's Discuss



surabaya.py

Telegram Group : <https://t.me/surabayadotpy>

You also can find me in group with mention @Squidward_Tenpoles

Big Thanks !

References

<https://www.oracle.com/internet-of-things/what-is-iot/>

<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-183.pdf>

<https://lucidworks.com/post/how-are-iot-and-industry-4-related/>

<https://www.slideshare.net/tegarimansyah/lessons-learned-for-internet-of-things-for-students>

<https://lucidworks.com/post/how-are-iot-and-industry-4-related/>

<https://micropython.org/>

<https://github.com/espressif/esptool>

<https://forum.micropython.org/viewtopic.php?t=7140>

<https://randomnerdtutorials.com/getting-started-thonny-micropython-python-ide-esp32-esp8266/>