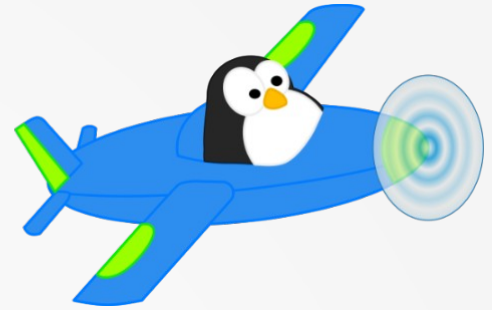


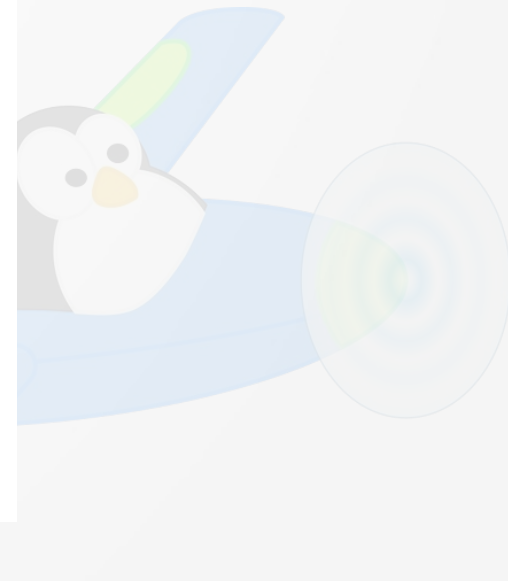
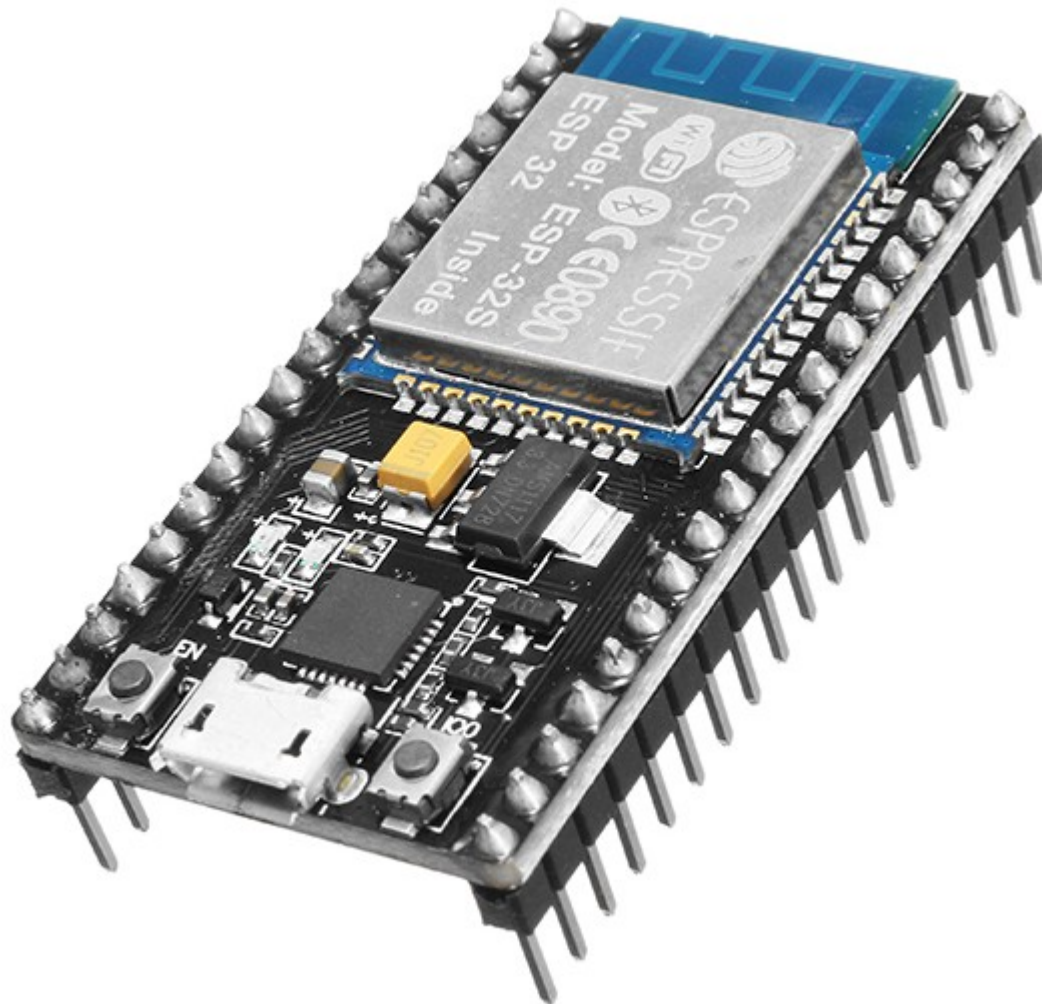
LINUX WAR



# ESP32 & IoT

Alberto

# ESP32 module



# ESP32 - inside

Processore Tensilica LX6 dual core a 240 MHz

- 520Kb di memoria SRAM
- Connettività wifi 802.11 b/g/n (supporto per WEP, WPA/WPA2 PSK/Enterprise)
- Connettività Bluetooth (classica e LE)
- 32 PIN di I/O con diverse periferiche disponibili
- accelerazione hardware per algoritmi di sicurezza (AES, SHA2, RSA-4096)

→

# Processore Tensilica LX6 dual core a 240 MHz

- Architecture RISC 32-bits Little-endian
- 24/16-bit instructions
- 64 general purpose registers with 16 visible registers
- Dual/single -core
- 240 MHz maximum frequency



# BENCHMARK RESULTS

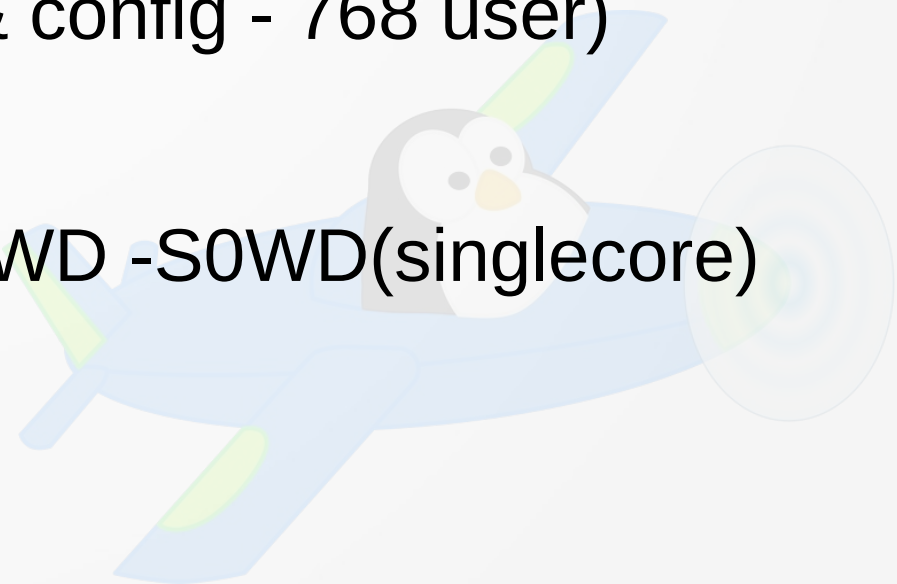
(smaller values are better)

	float [ms]	int32[ms]
Arduino uno @16MHz	3326	4446
ESP8266 @160MHz	216	28
ESP32 @240MHz	95	3



# ESP32 MEMORY

- ROM 448 KiB (boot & core )
- SRAM 520 KiB - data & program
- RTC Fast RAM 8 KiB - data
- RTC slow RAM 8 KiB - use in deep sleep mode
- eFUSE 1 KiB ( 256 bit MAC & config - 768 user)
- FLASH:
  - 0 Mib ESP32-D0WDQ6 -D0WD -S0WD(singlecore)
  - 2 MiB ESP32-D2WD
  - 4 MiB ESP32 PICO D4





# WiFi

- Supports sniffer, Station, SoftAP and Wi-Fi direct mode
- Max data rate of 150 Mbps@11n HT40, 72 Mbps@11n HT20, 54 Mbps@11g, and 11 Mbps@11b
- Maximum transmit power of 19.5 dBm@11b, 16.5 dBm@11g, 15.5 dBm@11n
- Minimum receiver sensitivity of -97 dBm
- 135 Mbps UDP sustained throughput

# Varianti Espressif

Module	Chip	Flash	PSRAM	Ant.
ESP32-WROOM-32	ESP32-D0WDQ6	4MB	-	MIFA
ESP32-WROOM-32D	ESP32-D0WD	4MB	-	MIFA
ESP32-WROOM-32U	ESP32-D0WD	4MB	-	U.FL
ESP32-SOLO-1	ESP32-S0WD	4MB	-	MIFA
ESP32-WROVER	ESP32-D0WDQ6	4MB	8MB	MIFA
ESP32-WROVER-I	ESP32-D0WDQ6	4MB	8MB	U.FL
ESP32-WROVER-B	ESP32-D0WD	4MB	8MB	MIFA
ESP32-WROVER-IB	ESP32-D0WD	4MB	8MB	U.FL

- ESP32-**D**.. denotes dual core, ESP32-**S**.. denotes single core chip
- MIFA - Meandered Inverted-F Antenna
- U.FL - U.FL / IPEX antenna connector
- ESP32-WROOM-x and ESP32-WROVER-x modules are also available with custom flash sizes of 8MB or 16MB, see [Espressif Products Ordering Information \(PDF\)](#)



SPI

I2C

I2S

SDIO

UART

CAN

ETH

IR

PWM

Temperature  
sensor

Touch sensor

DAC

ADC

Bluetooth  
link  
controller

Bluetooth  
baseband

Wi-Fi MAC

Wi-Fi  
baseband

RF receive

Clock  
generator

RF  
transmit

Switch

Balun

Core and memory

2 x Xtensa® 32-bit LX6  
Microprocessors

ROM

SRAM

Cryptographic hardware  
acceleration

SHA

RSA

AES

RNG

RTC

PMU

ULP  
coprocessor

Recovery  
memory

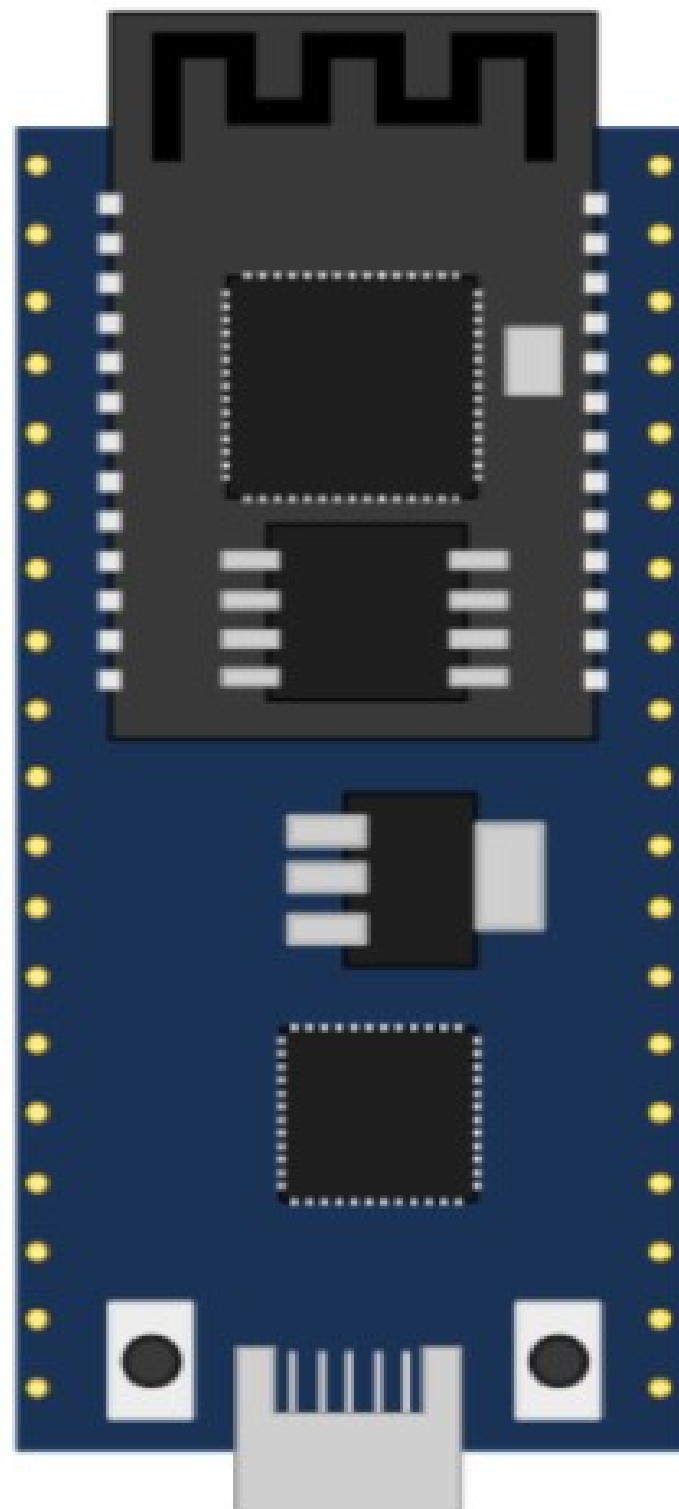
SPI

GPIO

UART

● Input  
● Only

	3.3V
	EN
● ADC1 CH0	GPIO36
● ADC1 CH3	GPIO39
● ADC1 CH6	GPIO34
● ADC1 CH7	GPIO35
ADC1 CH4	GPIO32
ADC1 CH5	GPIO33
	GPIO25
	GPIO26
	GPIO27
HSPI CLK	GPIO14
HSPI MISO	GPIO12
	GND
HSPI MOSI	GPIO13
U1RX	GPIO9
U1TX	GPIO10
U1RTS	GPIO11
	5V



GND	
GPIO23	VSP1 MOSI
GPIO22	
GPIO1	U0TX
GPIO3	U0RX
GPIO21	
GND	
GPIO19	VSP1 MISO
GPIO18	VSP1 CLK
GPIO5	VSP1 SS
GPIO17	U2TX
GPIO16	U2RX
GPIO4	
GPIO0	
GPIO2	
GPIO15	HSPI SS
GPIO8	U2CTS
GPIO7	U2RTS
GPIO6	U1CTS

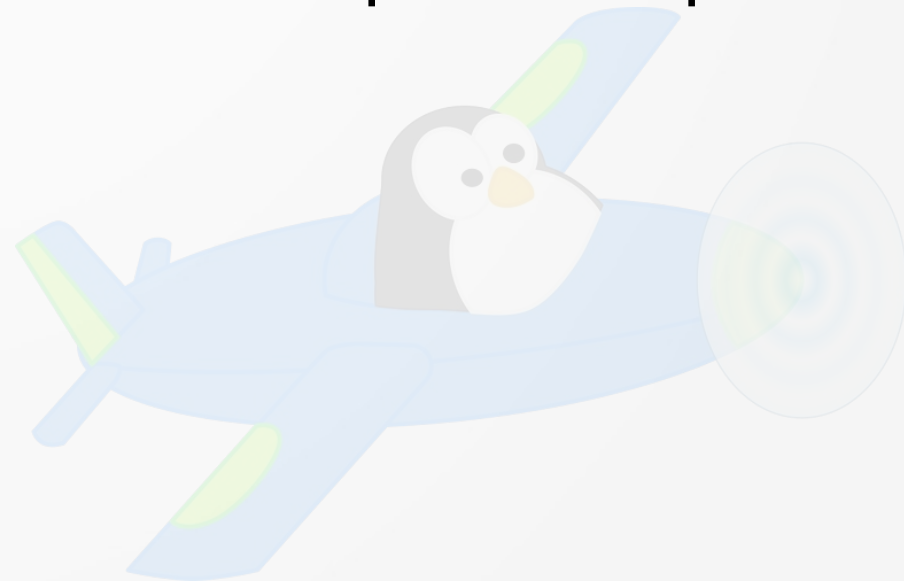
# Programmazione

- Pc ( Linux - Windows – Mac)
- Editor (eclipse)
- Toolchain Espressif ( linguaggio C )

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

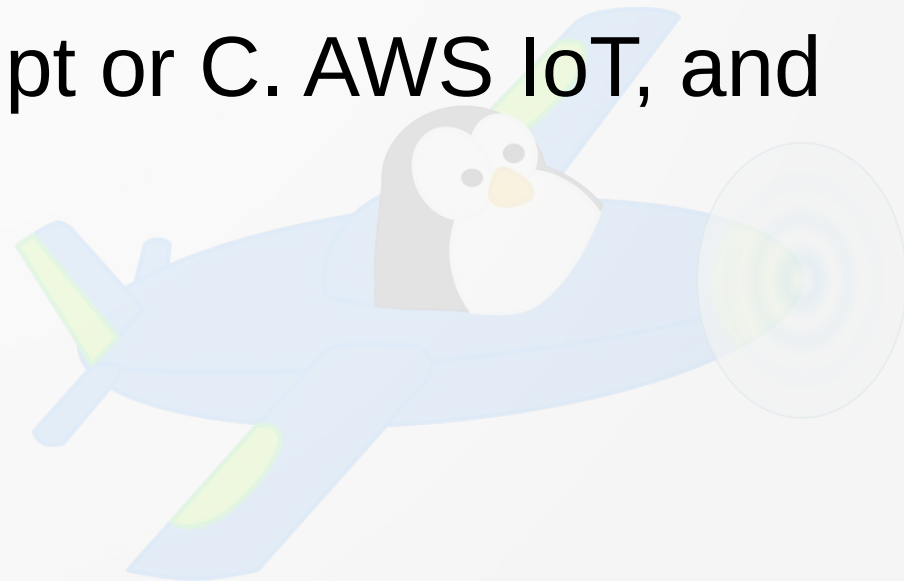
- Cavo usb

→



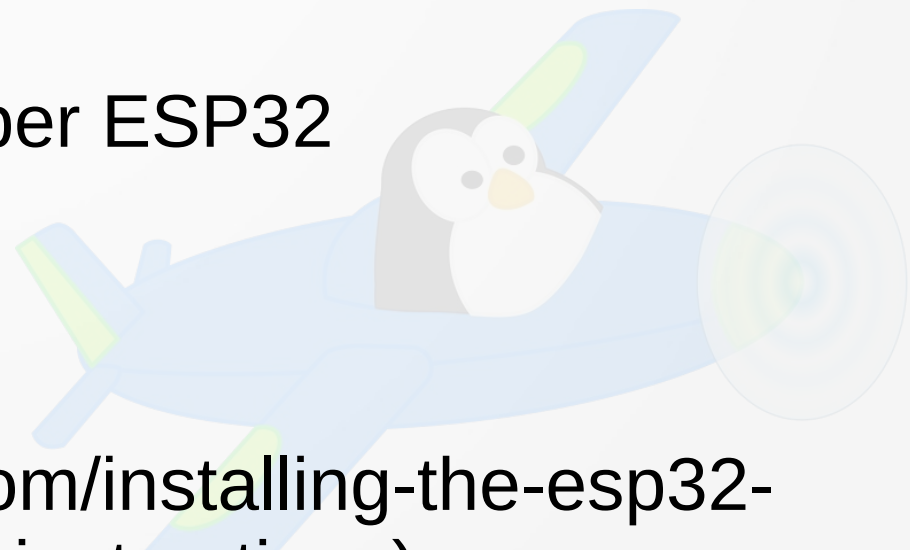
# Linguaggi e SDK

- Arduino IDE with the ESP32 Arduino Core
- ESP Easy – developed by home automation
- Espruino – JavaScript SDK emulating Node.js
- Lua RTOS for ESP32 or NodeMCU
- Mongoose OS – JavaScript or C. AWS IoT, and Google Cloud IoT.
- mruby for the ESP32
- MicroPython



# Programmazione semplificata

- Installare IDE arduino
- Installare i Board add-on  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)  
into the “Additional Board Manager URLs
- 
- Installare le eventuali librerie per ESP32
- 
- 
- (<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions>)



SketchStrumentiAiuto

Verifica / CompilaCtrl+R

CaricaCtrl+U

Carica tramite un programmatoreCtrl+Maiusc+U

Esporta sketch compilatoCtrl+Alt+S

Apri cartella dello sketchCtrl+K

#include libreria

Aggiungi file...

(LOW is the voltage level)

Contributed librerie

A6lib

AV2TelegramBot

Arduino-INA226

ArduinoJson

ArduinoOTA

BluetoothSerial

DNSServer

EEPROM

ESP32

ESP32 Async UDP

ESP32 Azure IoT Arduino

ESP32 BLE Arduino

ESPmDNS

Espalexa

FFat

FS

FlashStorage

HTTPClient

HTTPUpdate

IRremote

IRremoteESP8266

MF52C52

edStatus) {

t.sendMessage(chat\_id, "Led is ON", "");

se {

t.sendMessage(chat\_id, "Led is OFF", "");

xt == "/start") {

ng welcome = "Welcome to Flash Led Bot " + fr

ome += "/ledon : to switch the Led ON\n";

ome += "/ledoff : to switch the Led OFF\n";

ome += "/status : Returns current status of L

sendMessage(chat\_id, welcome, "Markdown");

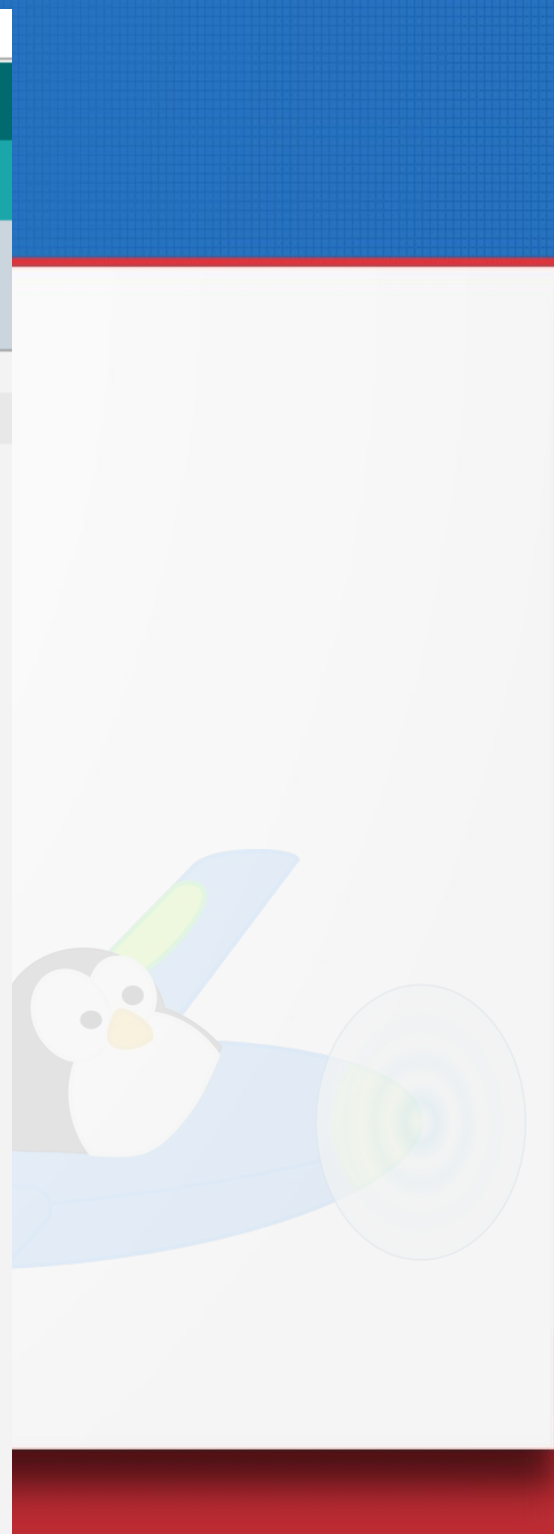
() {

egin(115200);

pt to connect to Wifi network:

rint("Connecting Wifi: ");

completato





Nuovo Ctrl+N

Apri... Ctrl+O

Apri sketch recenti >

Cartelle degli sketch >

Esempi >

Chiudi Ctrl+W

Salva Ctrl+S

Salva con nome... Ctrl+Maiusc+S

Imposta pagina... Ctrl+Maiusc+P

Stampa... Ctrl+P

Impostazioni... Ctrl+Virgola

Esci Ctrl+Q

```
String welcome = "Welcome to
welcome += "/ledon : to swi
welcome += "/ledoff : to sw
welcome += "/status : Retur
bot.sendMessage(chat_id, we
```

```
}
```

```
}
```

```
}
```

```
void setup() {
  Serial.begin(115200);
```

```
// Attempt to connect to Wifi n
Serial.print("Connecting Wifi:
```

Esempi per ESP32 Pico Kit

ArduinoOTA >

BluetoothSerial >

DNSServer >

EEPROM >

ESP32 >

ESP32 Async UDP >

ESP32 Azure IoT Arduino >

ESP32 BLE Arduino >

ESPmDNS >

FFat >

HTTPClient >

HTTPUpdate >

NetBIOS >

Preferences >

SD(esp32) >

SD\_MMC >

SimpleBLE >

SPI >

SPIFFS >

Ticker >

Update >

WebServer >

WiFi >

WiFiClientSecure >

AnalogOut >

Camera >

ChipID >

DeepSleep >

ESPNow >

FreeRTOS >

GPIO >

HallSensor >

I2S >

ResetReason >

RMT >

Time >

Timer >

Touch >

ExternalWakeUp

TimerWakeUp

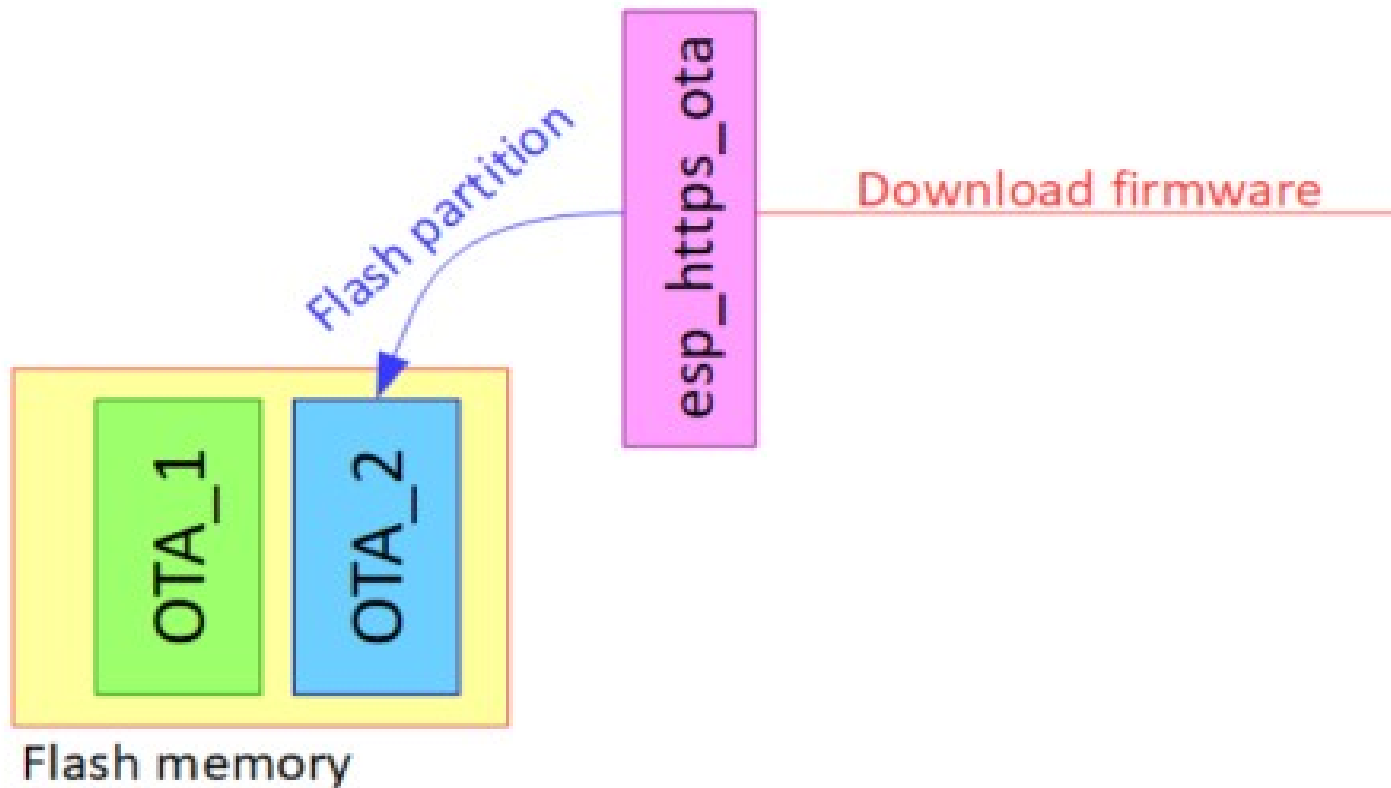
TouchWakeUp

the voltage level)

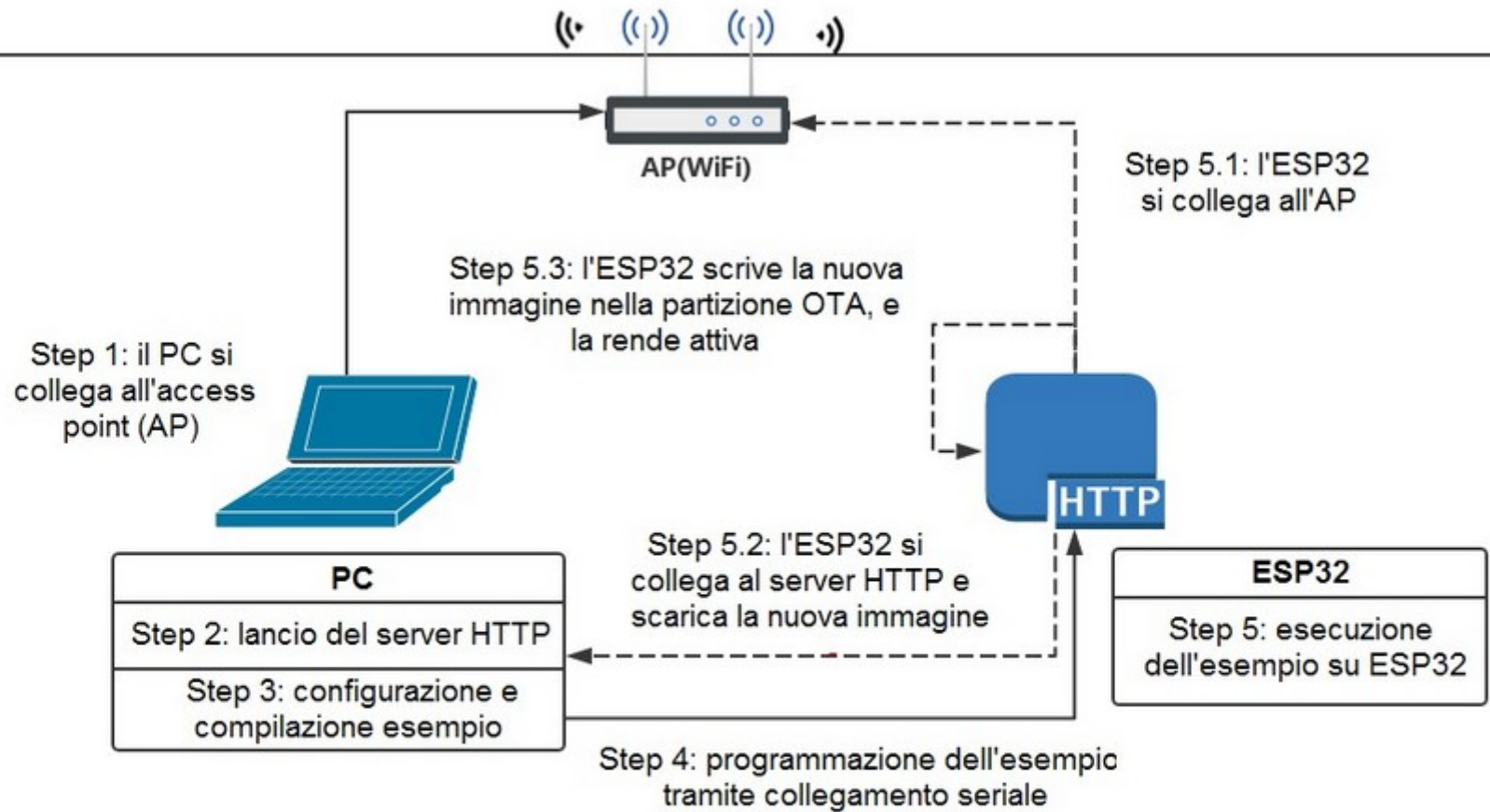
Caricamento completato

Leaving...

# Over The Air programming

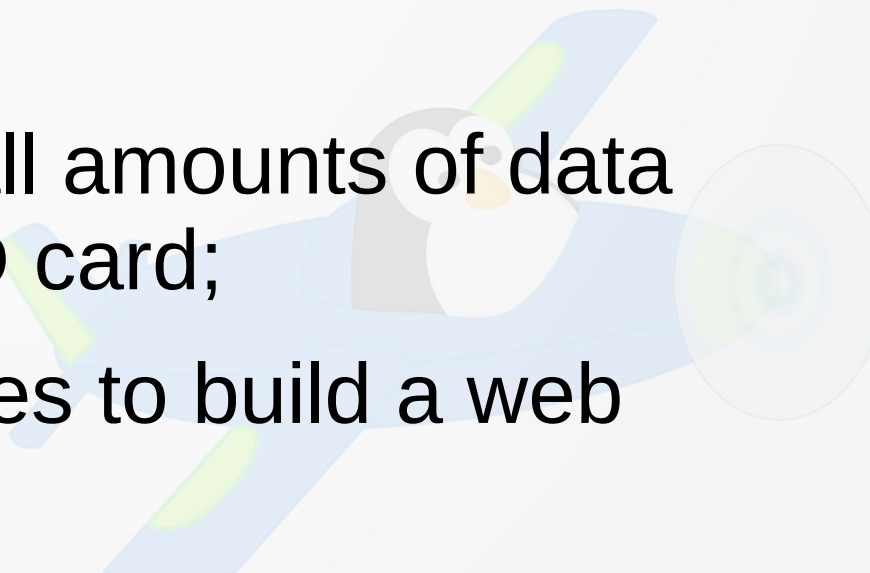


# OTA



Workflow della demo OTA

# ESP32 SPIFF

- The ESP32 contains a Serial Peripheral Interface Flash File System (SPIFFS).
  - 
  - Create configuration files with settings;
  - Save data permanently;
  - Create files to save small amounts of data instead of using a microSD card;
  - Save HTML and CSS files to build a web server;
- 

# Moduli base

Genuine

**OLD**



**NEW**



Q Mouse over to zoom in



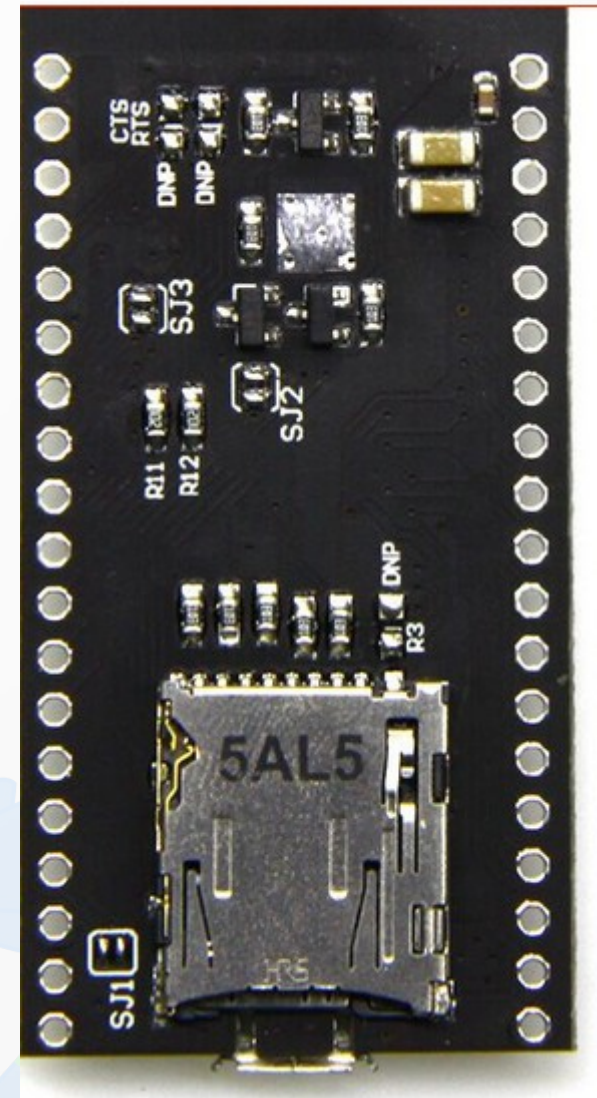


# Moduli USB

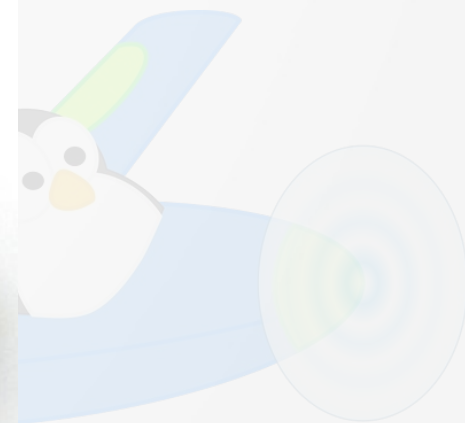
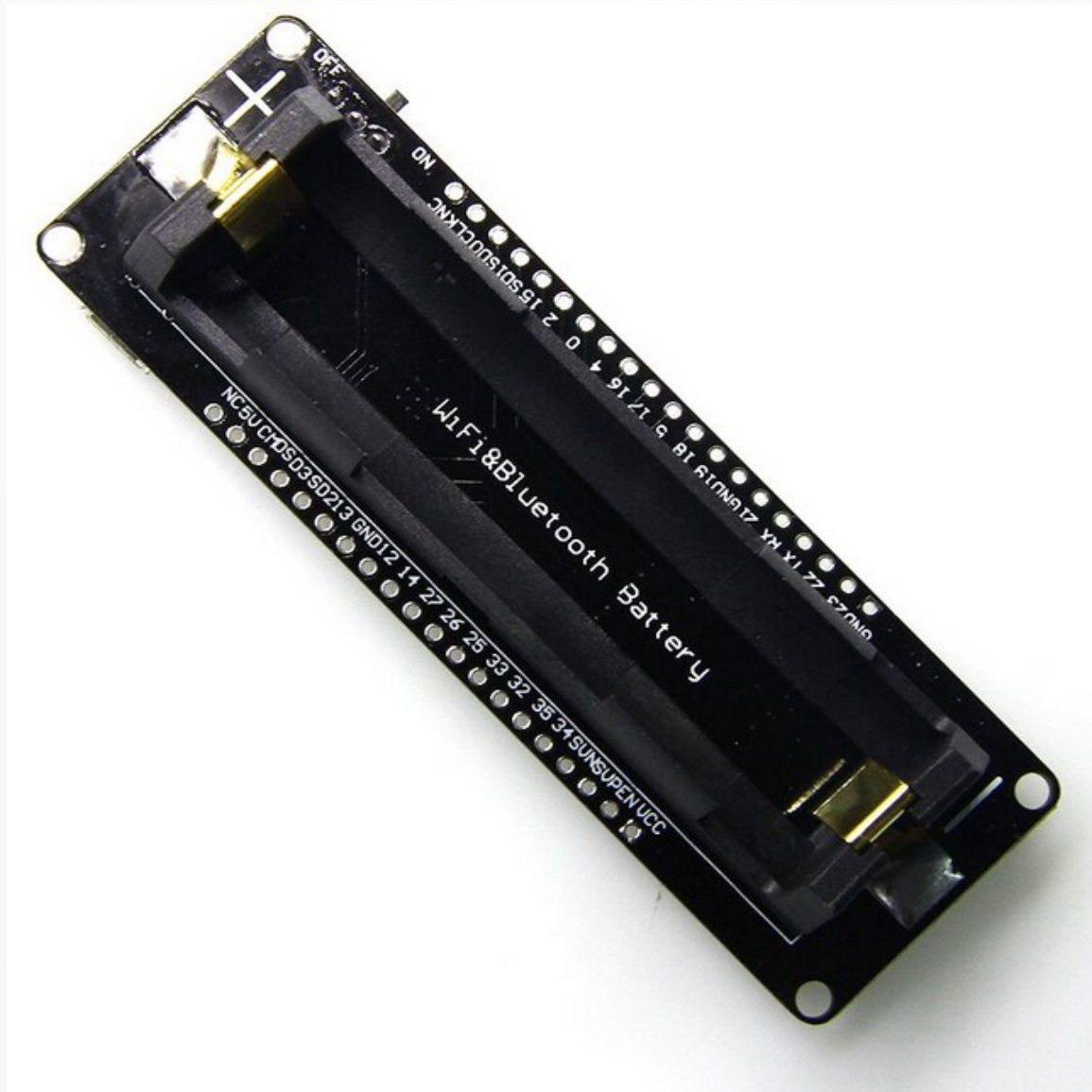




# ESP32 + SD card



# ESP32 + battery



# ESP32 + OLED

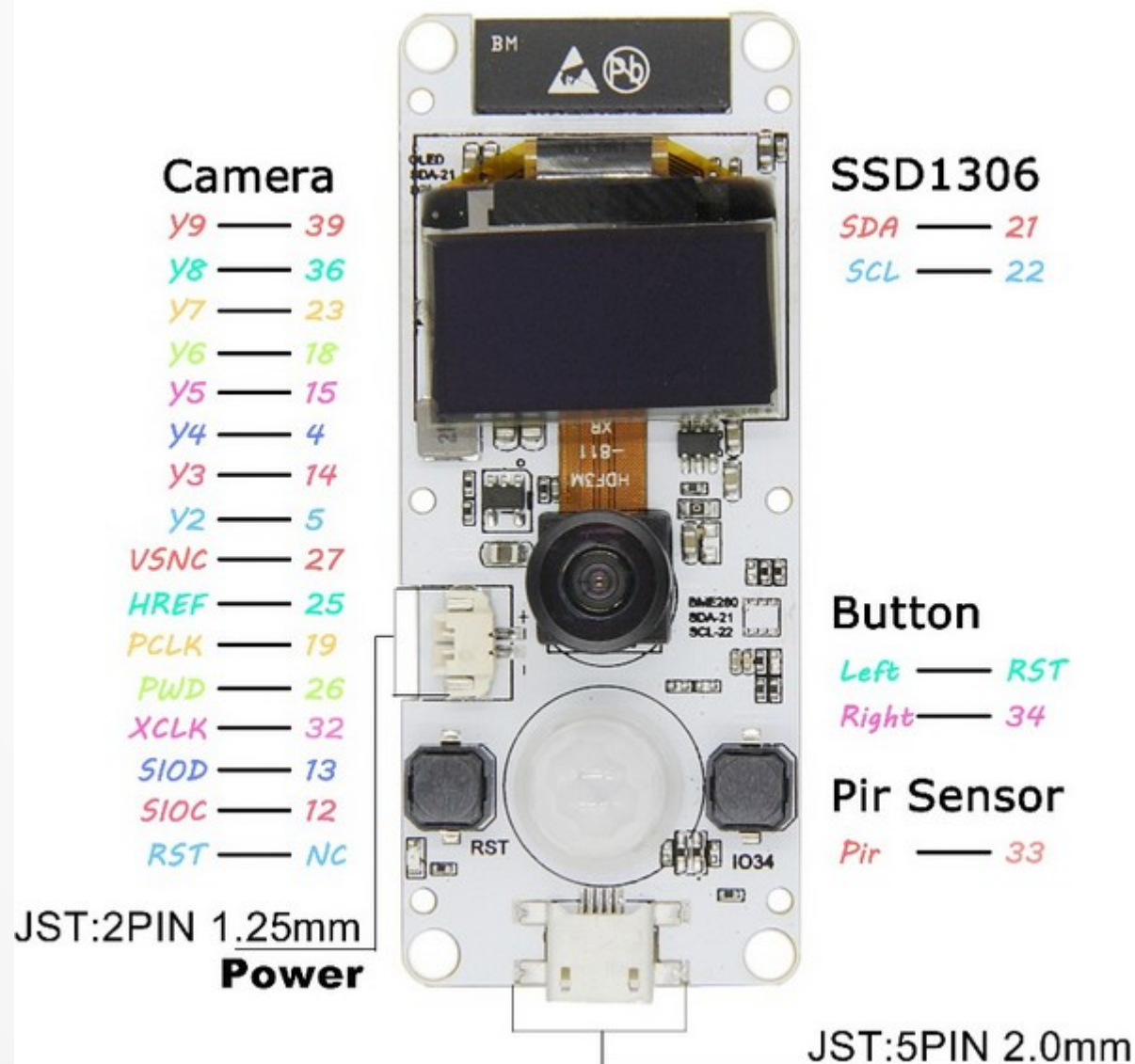




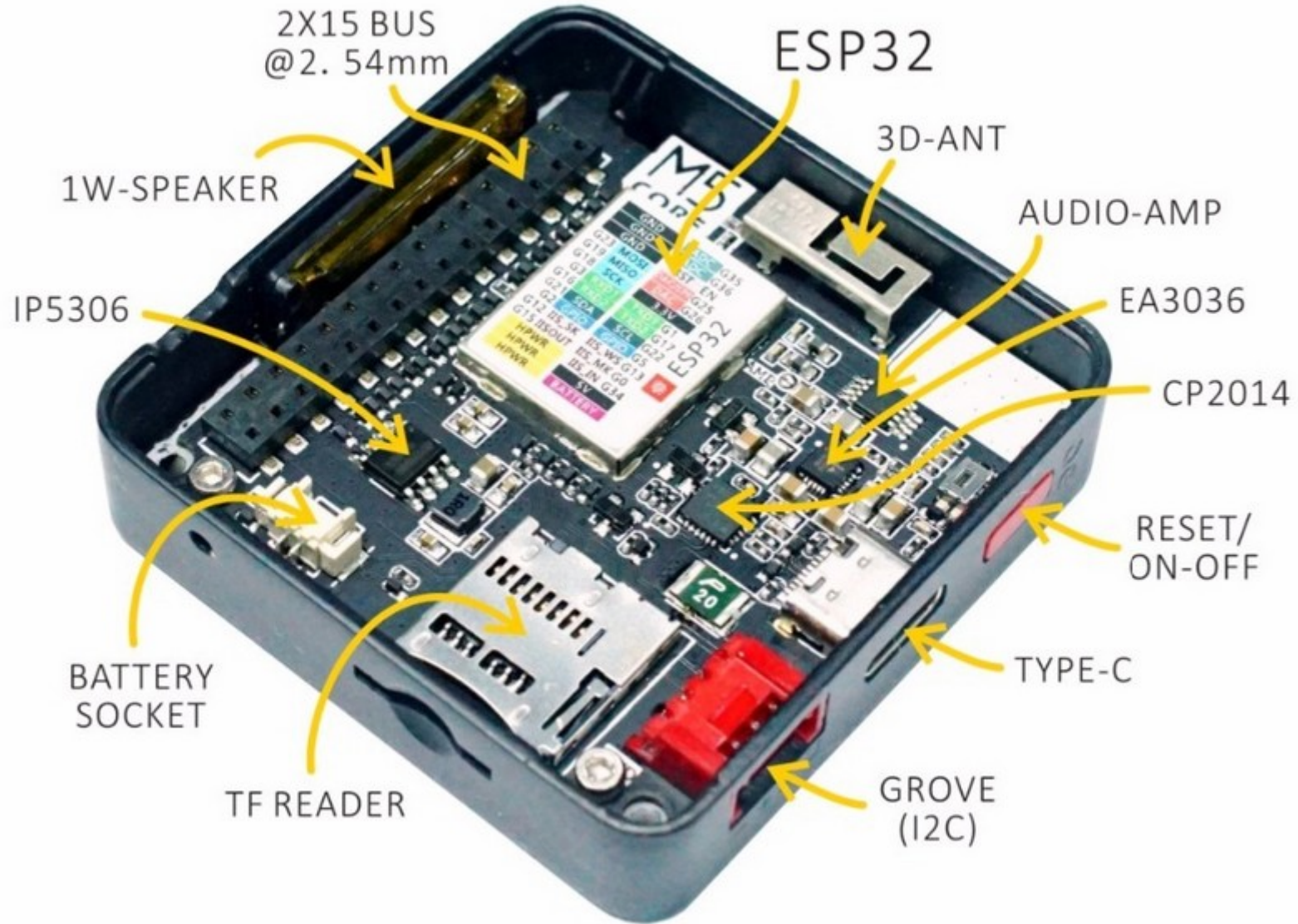
# ESP32 + LoRa + GPS



# ESP32 camera + PIR + Oled 64x128



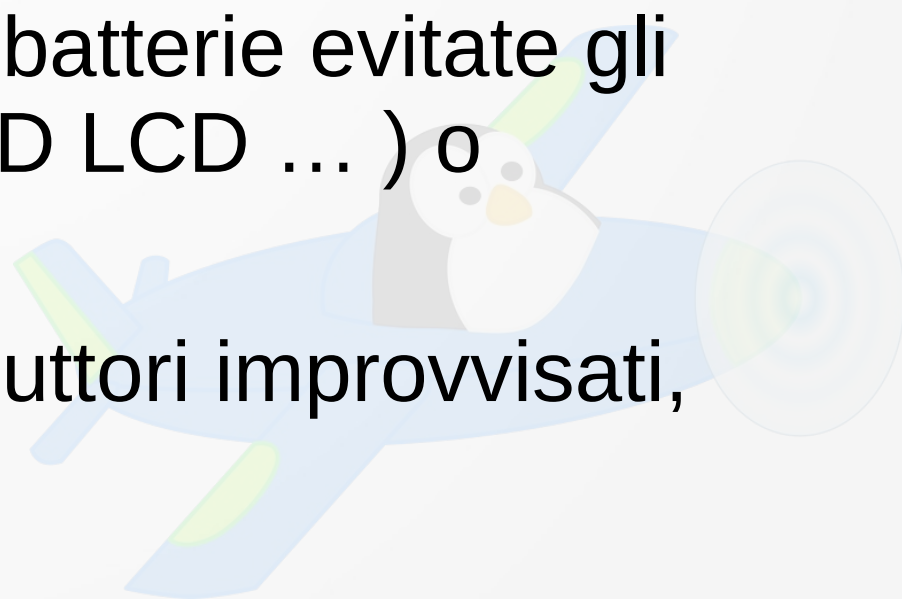
# M5 Stack





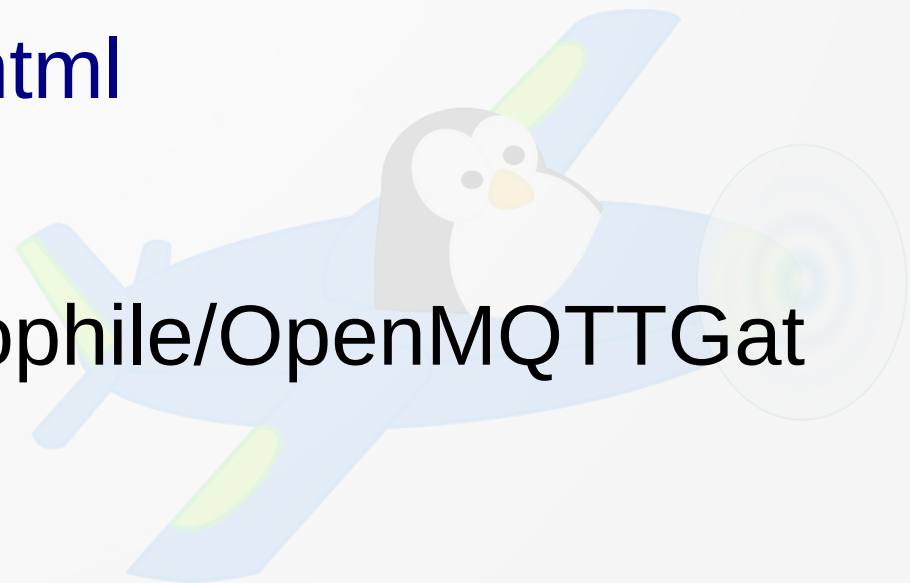
# Consigli per gli acquisti

- SE costa troppo poco .... a volte non funziona
- Ma e' dual core o single core ?
- Quanta memoria flash ha realmente ? ( non quanta ne supporta o potrebbe montarne)
- Se lo dovete alimentare a batterie evitate gli accessori inutili ( USB LED LCD ... ) o spegneteli .
- Attenti ai falsi profeti ( traduttori improvvisati, youtuber cerca-click ... )

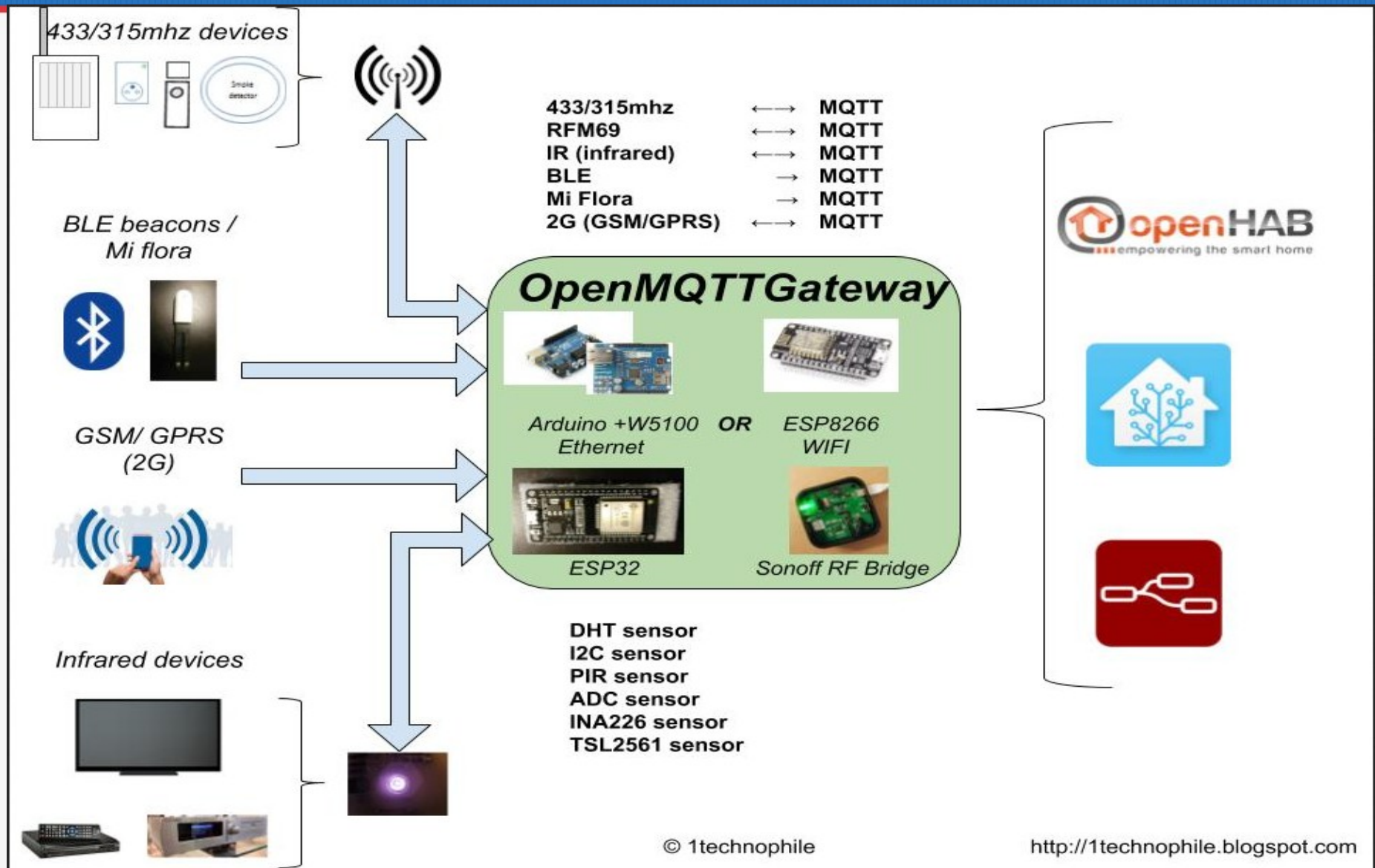


# Firmware self service

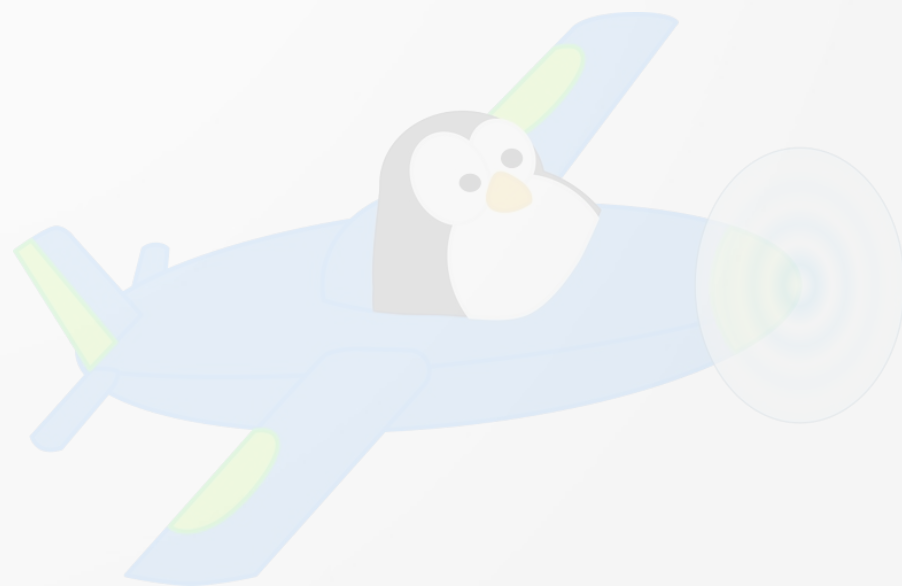
- Cosa trovate gia' pronto :
- 
- <https://github.com/arendst/Sonoff-Tasmota/wiki>
- 
- <https://esphome.io/index.html>
- 
- <https://github.com/1technophile/OpenMQTTGateway>



# IOT Home automation .... next

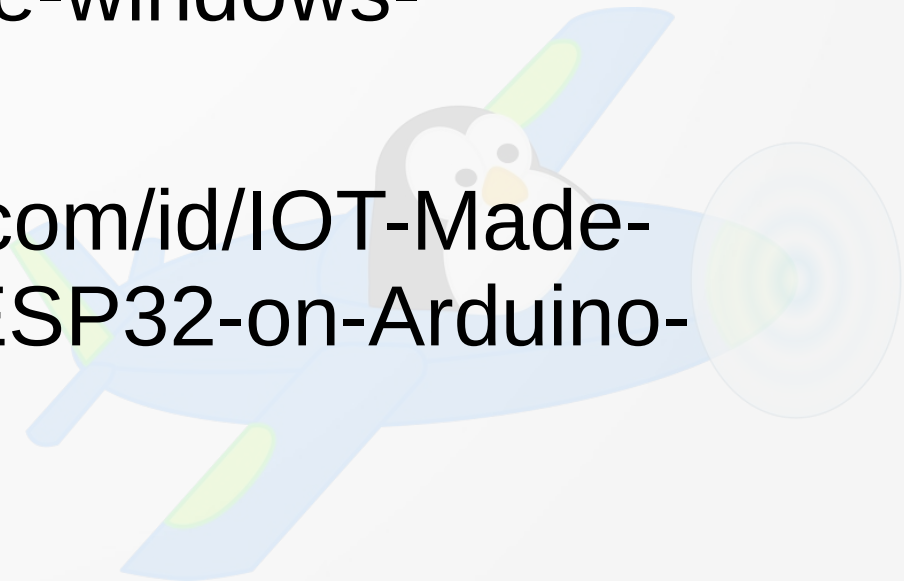


# DEMO ...



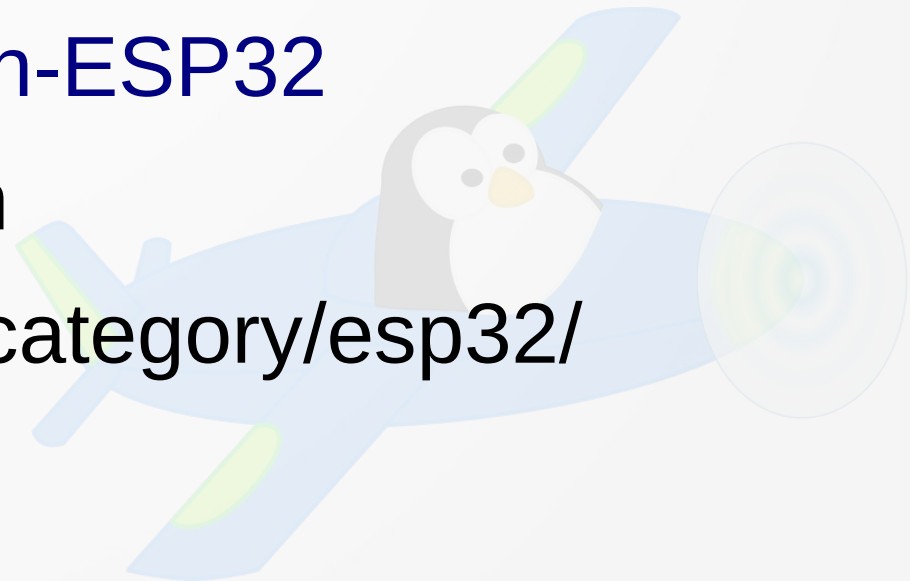
# Link utili – SETUP

- ➔ <https://forum.arduino.cc/index.php?topic=358374.0>
- ➔ <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
- ➔ <https://www.instructables.com/id/IOT-Made-Simple-Playing-With-the-ESP32-on-Arduino->



# Sitografia

- <https://en.wikipedia.org/wiki/ESP32>
- <http://esp32.net/>  
<http://iot-bits.com/category/esp32-articles/>
- <https://www.esp32.com> (english forum)
- <https://leanpub.com/kolban-ESP32>
- <https://github.com/nkolban>
- <http://www.lucadentella.it/category/esp32/>





# Tutorial (inglese)

→ Andreas Spiess

[https://www.youtube.com/channel/UCu7\\_D0o48KbfhpEohoP7YSQ](https://www.youtube.com/channel/UCu7_D0o48KbfhpEohoP7YSQ)

→

→ Peter Scargill's Tech Blog

→ <https://tech.scargill.net/>

→

→ Florian 1technophile

→ <https://1technophile.blogspot.com/>

