



ESP32 Low Power

Here is where your
presentation begins



TABLE OF CONTENTS

01

INTRODUCTION

02

ABOUT ESP32

03

ESP32 THINGSBOARD

04

ESP32 AS WEARABLE DEVICE

05

ESP32 LOW POWER



An abstract graphic design featuring a central orange circle with the white number '01'. This circle is connected by a thick orange line to a larger dark grey circle on the left, which contains a smaller orange circle. Various other organic shapes in teal, white, and olive green are scattered around the central element. The background is a light blue-grey color.

01

INTRODUCTION

LAST YEARS

Link github (<https://bit.ly/30RHsZ3>)
<https://github.com/hasbiida/TrainingEspRpi>

ESP32 – Cloud/Rpi (dash python+database)

ESP32 simple webserver



HIGHLIGHTS THIS YEARS

ESP32 Thingsboard
ESP32 as IoT node devices

01

02

**ESP32 as
Wearable device**

How to plot sensor data to
smartphone (esp as
webserver)

03

**ESP32 low
power**

How to tuning ESP for
lower energy usage

ABOUT ESP32

02

An abstract graphic design on the right side of the slide. It features several organic, teardrop-like shapes in orange, light green, and dark grey. A large orange shape in the foreground contains a white circle with the number '02' inside. Other shapes are scattered around, including a light green shape with a dark grey circle inside, and a dark grey shape with a light green circle inside. There are also several small dots in blue, green, and black scattered throughout the design.

GETTING STARTED

01

ESP32

A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications

02

SPEC

Dual core
Wi-Fi and bluetooth
32 bit programs.
Freq up to 240MHz and
it has a 512 kB RAM
30 or 36 pins

03

PROGRAMMING ENV

Arduino IDE
Espressif IDF
Micropython
JavaScript
LUA

EXAMPLES OF ESP32 BOARDS

DOIT DEVKIT V1



ESP32 DevKit



ESP-32S NodeMCU



ESP32 Thing



WEMOS LOLIN32



"WeMos" OLED



HUZZAH32

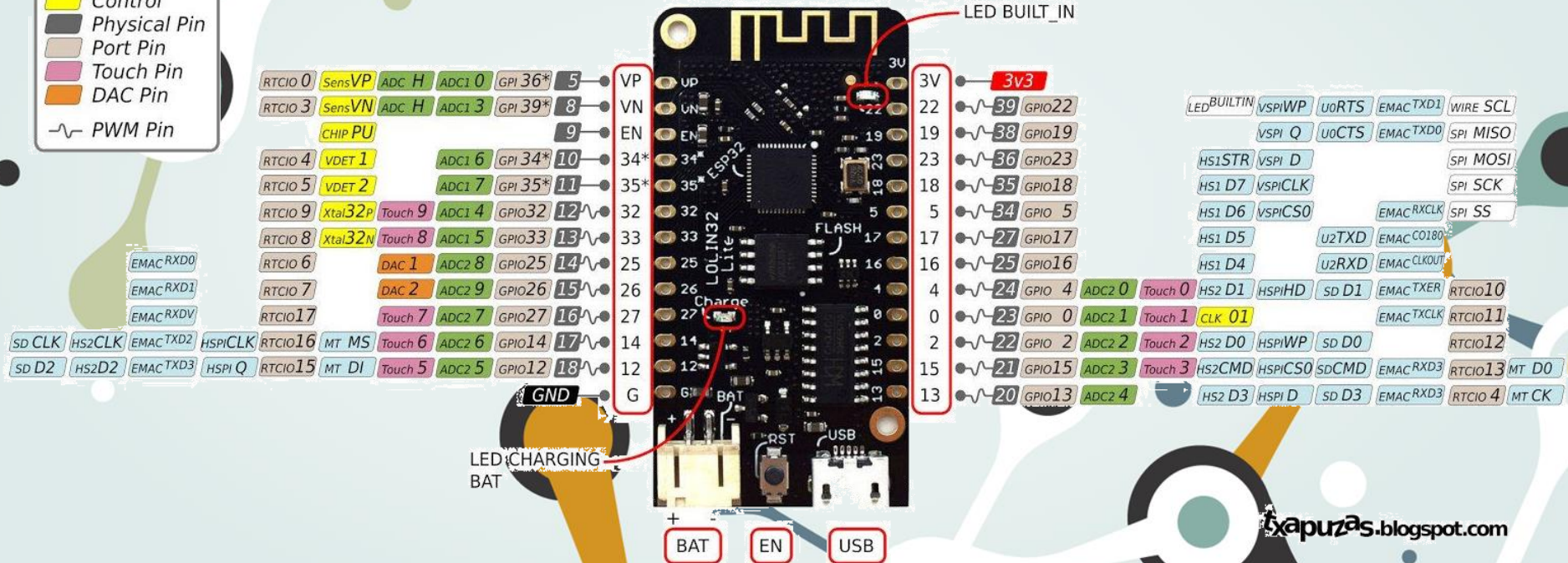


Others

(...)

Lolin32 Lite pinout

- Power
- GND
- Serial Pin
- Analog Pin
- Control
- Physical Pin
- Port Pin
- Touch Pin
- DAC Pin
- ~ PWM Pin



ARDUINO IDE



- Install IDE
- Add board manager URL
https://dl.espressif.com/dl/package_esp32_index.json
- Tools → board → board manager → esp32
- Open example blink
- change pin 22 (led built in)
- Upload

ESP32

THINGSBOARD

ESP32 as IoT node devices





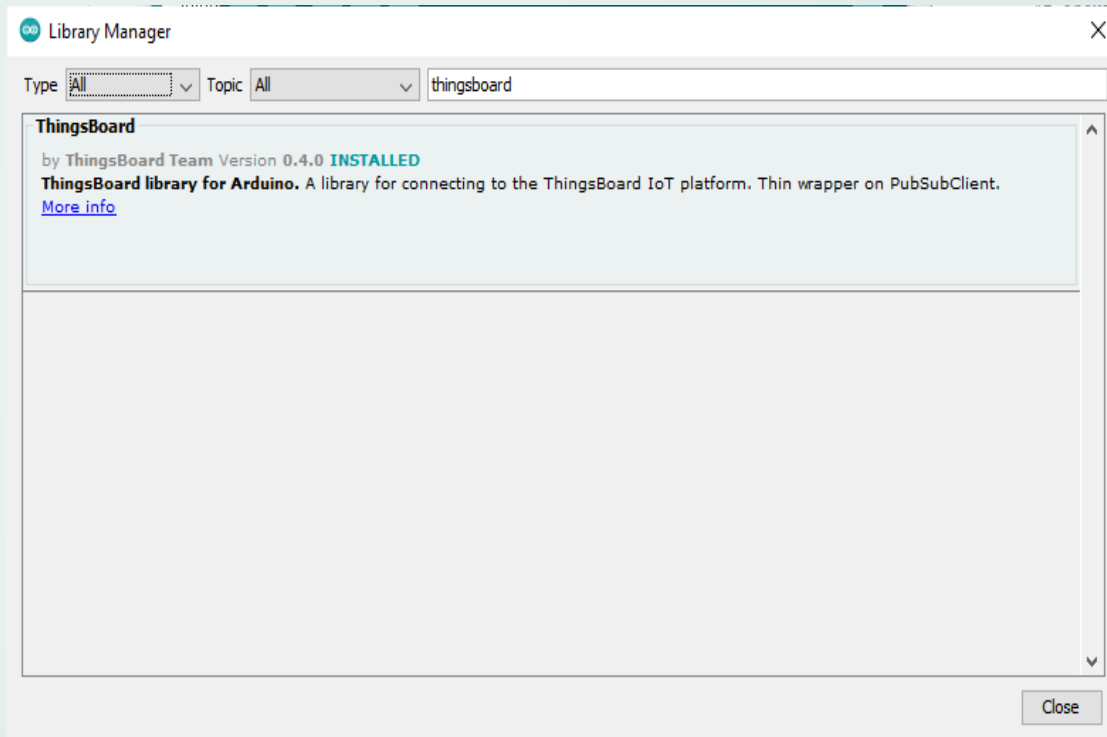
Download bahan materi

[https://github.com/hasbiestheim](https://github.com/hasbiestheim/FGD_Tel-U_2020)
[/FGD_Tel-U_2020](#)

<https://bit.ly/3iPmUGG>

INSTALL LIBRARY

- Thingsboard
- ArduinoHttpClient
- ArduinoJson
- DHT sensor library



PROGRAM NODE IOT THINGSBOARD

- Untuk sensor DHT11
 - ESP32_thingsboard_DHT11
- Untuk sensor Pulse Sensor
 - ESP32_thingsboard_PulseSensor

THINGS BOARD LOGIN

demo.thingsboard.io/login



Log in to see ThingsBoard in action.

Username (email) *

✉ hasbiida@gmail.com

Invalid email format.

Password



[FORGOT PASSWORD?](#)

LOGIN

[Do not have an account?](#)

[CREATE AN ACCOUNT](#)



THINGSBOARD DASHBOARD



ThingsBoard is an open-source server-side platform yang memungkinkan untuk monitor dan control perangkat IoT. Gratis untuk digunakan secara personal dan commercial dan dapat digunakan dimana saja



FITUR DARI THINGSBOARD



1. **Provision** perangkat, aset, dan pelanggan serta menentukan hubungan di antara mereka.
2. **Kumpulkan dan visualisasikan** data dari perangkat dan aset.
3. **Menganalisis** telemetry yang masuk dan memicu alarm dengan pemrosesan peristiwa yang kompleks.
4. **Kontrol** perangkat Anda menggunakan remote produce call(RPC).
5. Buat **alur kerja** berdasarkan life cycle perangkat, event, REST API, RPC request, dll
6. Desain **dasbor** dinamis dan responsif serta telemetry perangkat atau aset dan wawasan terkini kepada pelanggan Anda
7. Aktifkan fitur khusus kasus penggunaan menggunakan **rule-chain** yang dapat disesuaikan.
8. **Push** data perangkat ke sistem lain.

THINGSBOARD

Detail_sensor1

IoT Dashboard ▾

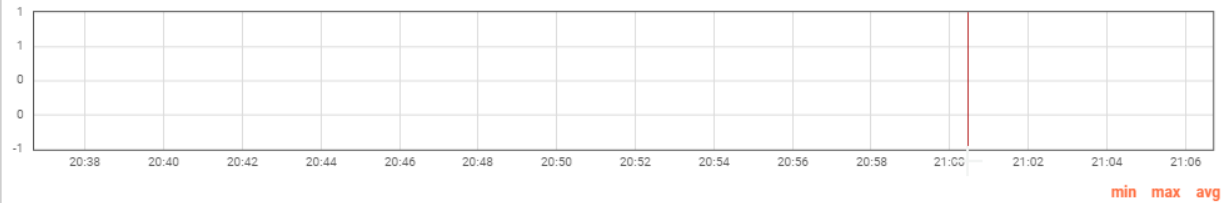
Entities

Realtime - last minute



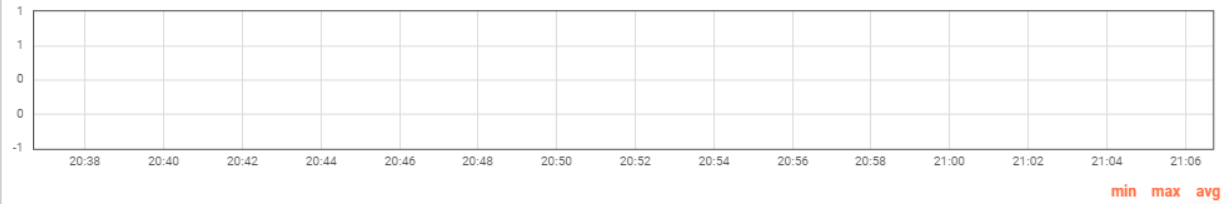
Suhu History

Realtime - last 30 minutes



Kelembapan History

Realtime - last 30 minutes



Power IoT

Powered by [Thingsboard v3.1.1](#)

KONEKSI ESP DENGAN THINGSBOARD

- Add device dan copy access token pada demo.thingsboard.io
- Sesuaikan access token dengan device node (esp32)
- Cek koneksi pada telemetry
- Buat dashboard dengan data sesuai data telemetry



INTERNET ALL THE THINGS



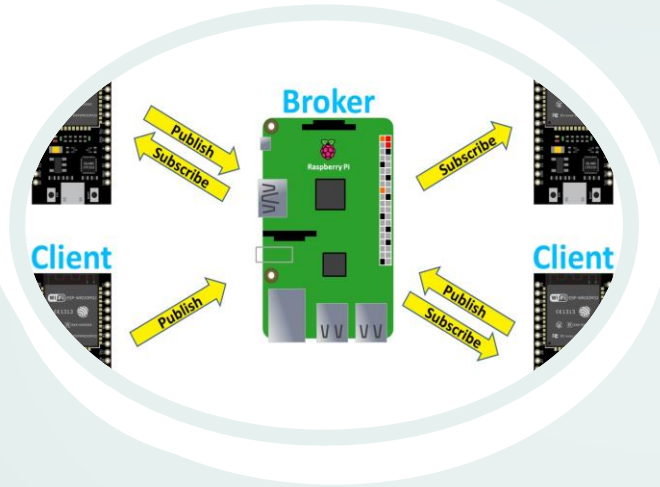
04

ESP32 AS WEARABLE DEVICE

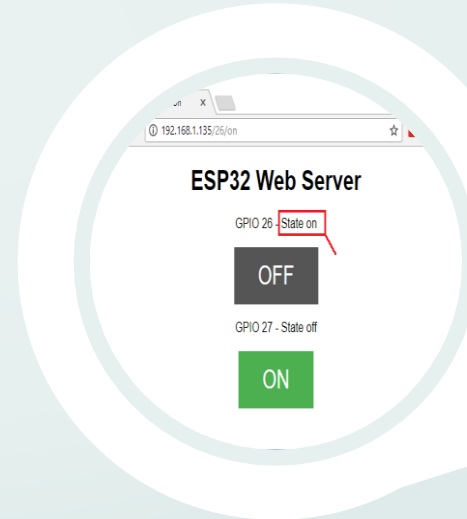
How to plot sensor data to
smartphone (esp as webserver)

PERBEDAAN

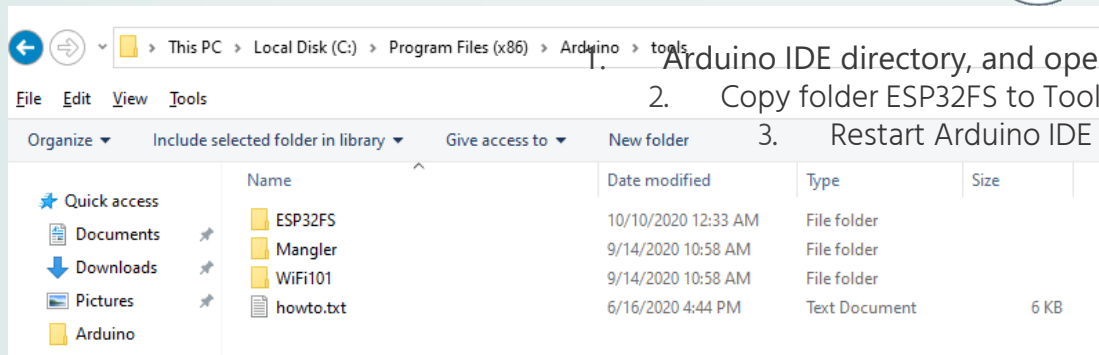
ESP AS WEBSERVER



ESP AS CLIENT



Install ESP32 Filesystem Uploader in Arduino IDE





Plot Sensor Readings in Real Time Charts – Web Server



1. Install Filesystem uploader plugin
2. Install Libraries [ESPAsyncWebServer](#) and [AsyncTCP](#)
3. Uploading Code and Files - Tools > ESP32/ESP8266 Data Sketch Upload
4. Web Server using SPIFFS (SPI Flash File System)



05

ESP32 LOW POWER

How to tuning ESP for lower
energy usage

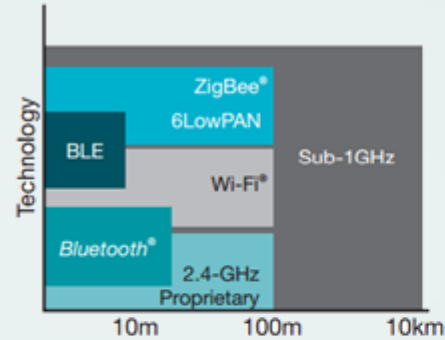
DESAIN METRIC

METRIC (1- 5)	PRIORITY	ADVANTAGES	DISADVANTAGES
CONNECTIVITY	3	Low power	Lambat dan jarak dekat
POWER MANAGEMENT	5	Small battery	Low performance
SECURITY	2	Low cost	Unsecure
COMPLEXITY	3	Mudah digunakan	Terbatas
RAPID EVOLUTION	1	Arduino support	Terbatas

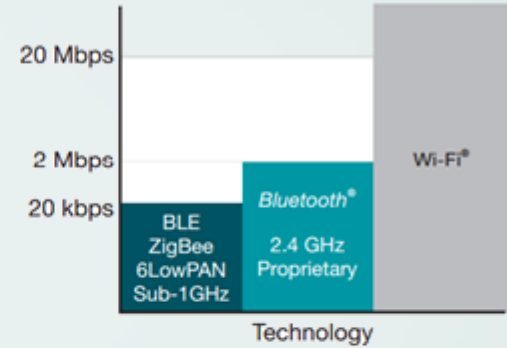
PARAMETER CONNECTIVITY

1. Range
2. Throughput
3. Power source
4. Topology

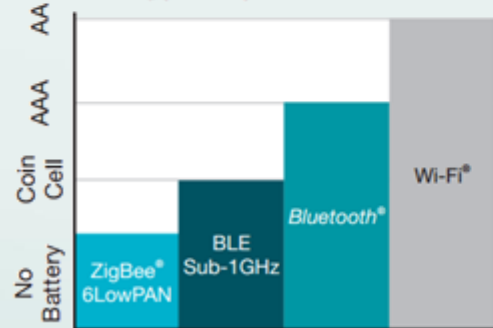
Range



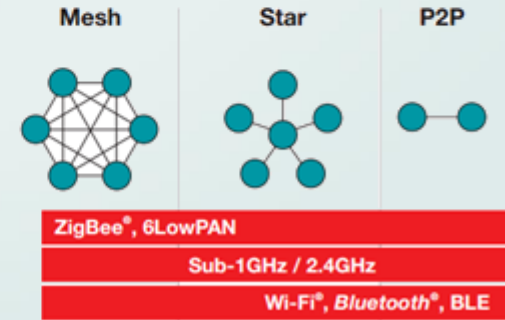
Throughput



Typical power source



Typical topology



ESP32 Power Modes



Active Mode



Active:

- WiFi
- Bluetooth
- Radio
- ESP32 Core
- ULP Coprocessor
- Peripherals
- RTC

Inactive:

Power Consumption:

160~260mA

- Active Mode
- Modem Sleep Mode
- Light Sleep Mode
- Deep Sleep Mode
- Hibernation Mode



MODEM SLEEP LIGHT SLEEP DEEP SLEEP HIBERNATION

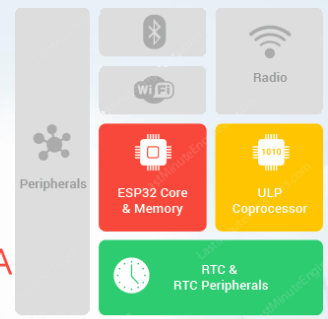
Active Mode



- Active:**
- WiFi
 - Bluetooth
 - Radio
 - ESP32 Core
 - ULP Coprocessor
 - Peripherals
 - RTC

Power Consumption:
160~260mA

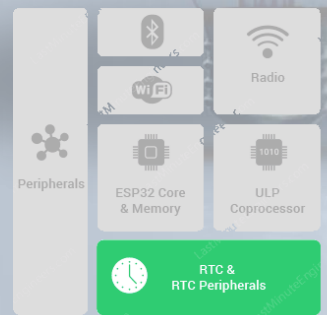
Modem Sleep



- Active:**
- ESP32 Core
 - ULP Coprocessor
 - RTC
- Inactive:**
- WiFi
 - Bluetooth
 - Radio
 - Peripherals

Power Consumption:
3~20mA

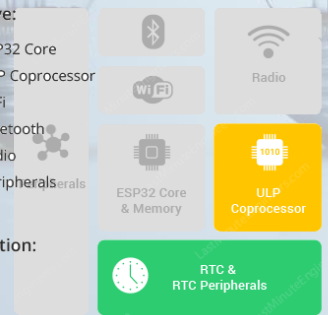
Hibernation



- Active:**
- RTC

Power Consumption:
2.5µA

Deep Sleep

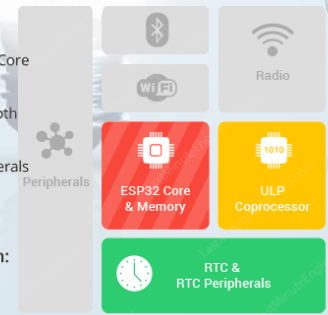


- Inactive:**
- ESP32 Core
 - ULP Coprocessor
 - WiFi
 - Bluetooth
 - Radio
 - Peripherals

- Active:**
- ULP Coprocessor
 - RTC

Power Consumption:
10µA

Light Sleep



- Inactive:**
- ESP32 Core
 - WiFi
 - Bluetooth
 - Radio
 - Peripherals

- Active:**
- ULP Coprocessor
 - RTC
- Paused:**
- ESP32 Core
- Inactive:**
- WiFi
 - Bluetooth
 - Radio
 - Peripherals

Power Consumption:
0.8mA



Code Diff (Thingsboard sleepy) using sleep timer



1. RTC store data
2. Init sleep timer (waker)
3. Function to print cause of wake up
4. Sleep command in main program

```
RTC_DATA_ATTR int bootCount = 0;
```

```
esp_sleep_enable_timer_wakeup(time_in_us)
```

```
void print_wakeup_reason()
```

```
esp_deep_sleep_start()
```



Code Diff (Chart webserver sleepy) using touch wake up



```
RTC_DATA_ATTR int bootCount = 0;
```

```
#define Threshold 40
```

```
touchAttachInterrupt(T3, callback, Threshold);
```

1. RTC store data
2. Init touchpin and threshold sensitivity (waker)
3. Function to print cause of wake up
4. Sleep command in main program

```
void print_wakeup_reason()
```

```
esp_sleep_enable_touchpad_wakeup()
```

THANKS

Do you have any questions?

hasbiida@gmail.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution

