Memulai dengan IoT

Table of contents

01 **Review teknologi**

Review teknologi IoT

02 **IoT development**

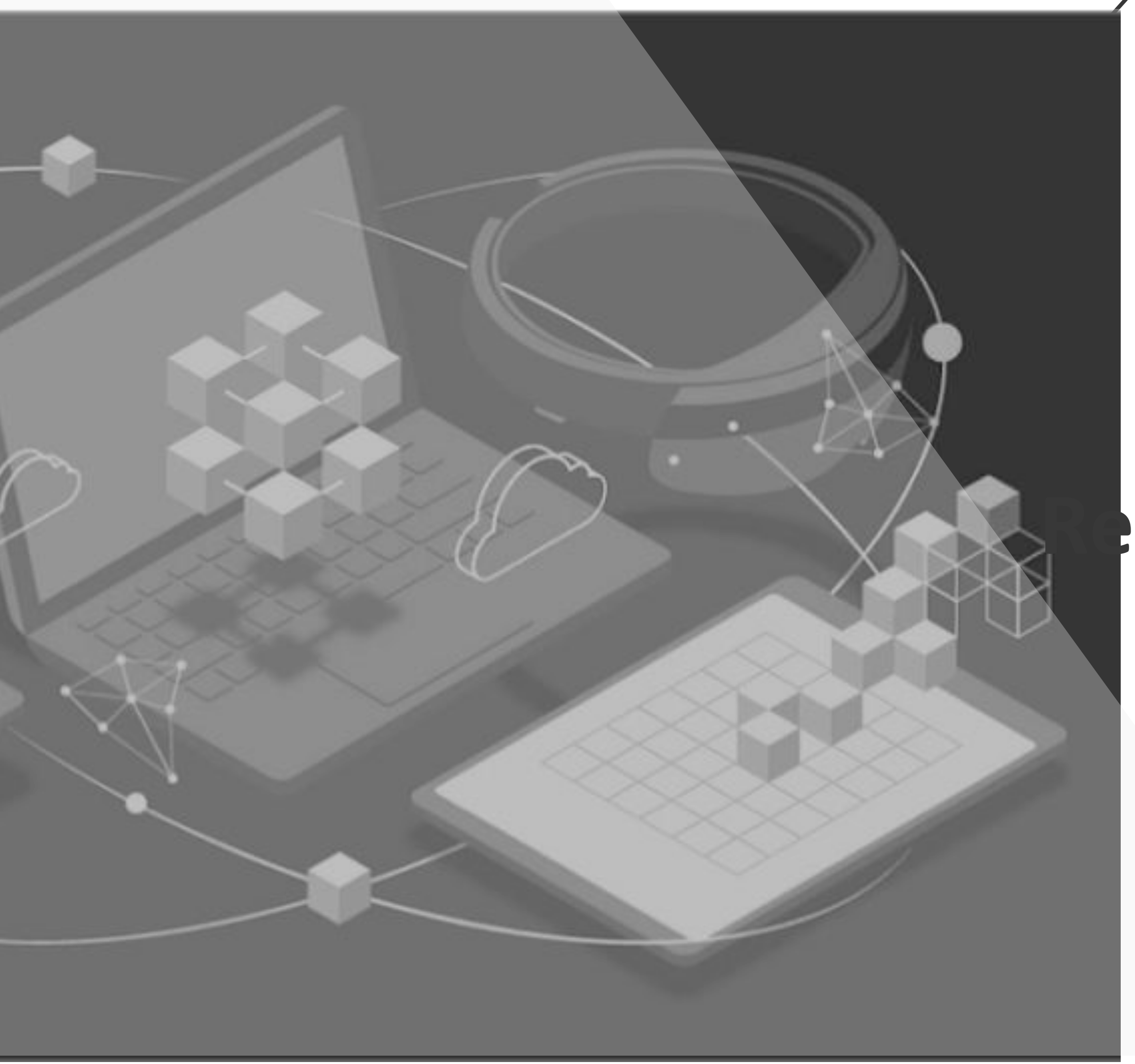
Langkah untuk memulai menguasai Aplikasi IoT

03 **ESP32**

Development solusi IoT dengan ESP32

04 **Sensor**

Sensor aplikasi



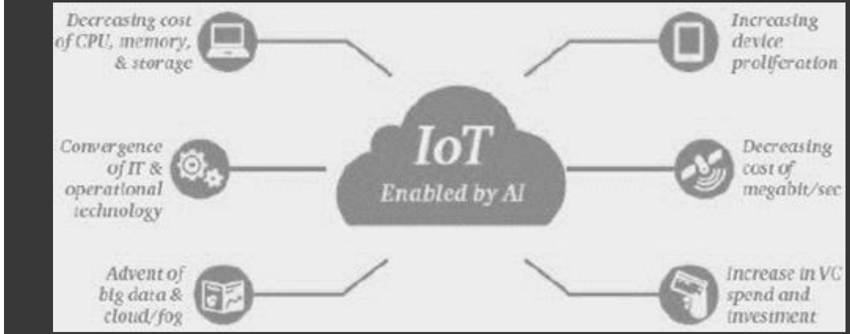
01

Review teknologi

Review teknologi IoT

IoT

Hari ini di era Industri 4.0 dan persiapan society 5.0, IoT menjadi enabling teknology


Decreasing cost of CPU, memory, & storage
Increasing device proliferation
Convergence of IT & operational technology
IoT
Enabled by AI
Decreasing cost of megabit/sec
Advent of big data & cloud/fog
Increase in VC spend and investment

IoT

Dari sensor ke cloud, integrated circuits yang mampu secara akurat mengambil, memproses dan mengirim data sensor secara pintar.

2. IoT development

Langkah untuk memulai menguasai Aplikasi IoT

Aplikasi dari IoT

Building and home automation

Automasi gedung dan rumah
Power management, AC, Deteksi gas bocor, Motion sensor, Smart Lock

Smart Cities

Pengaturan konsumsi daya seperti pada lampu jalan, CCTV, menggunakan koneksi jarak jauh (LoRa/NB-IoT), biasanya dikontrol secara centralized

Aplikasi dari IoT

Smart Manufacturing

Smart factory dan Industri 4.0, system yang membutuhkan desain security dan robust. Untuk mencapai lingkungan factory/pabrik yang smarter, safer, dan more efficient

Automotive

Teknology otomotif yang pintar, mulai dari OBC, Head unit, Telementry kontrol.

Aplikasi dari IoT

WEARABLES

Ultra low power untuk wearable device

HEALTCARE

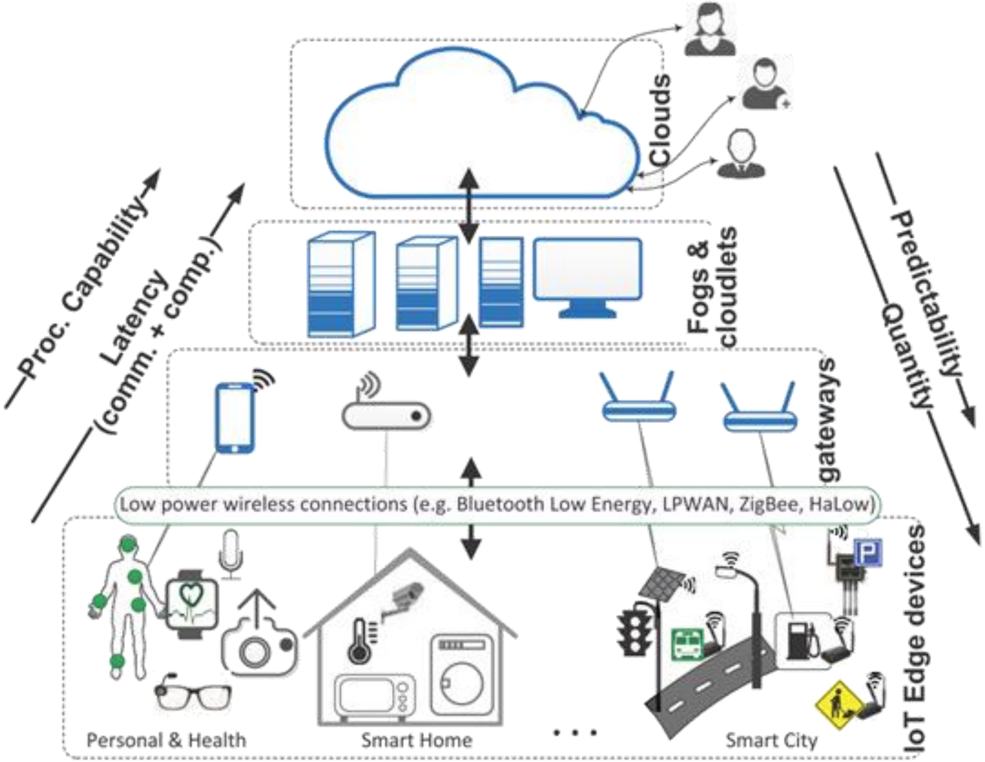
Revolusi kesehatan, monitoring pasien, telehealth system

AGRICULTURE

Mempercepat process dan efisiensi pertanian. Transport, drone/Survey, automasi

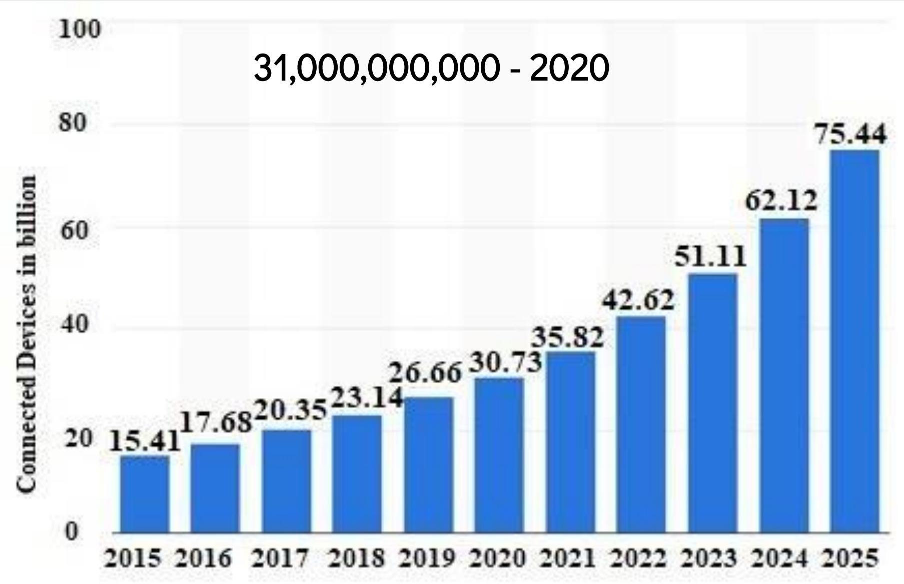
IoT Architecture

- **IoT devices** – Perangkat interkoneksi
- **Networks** – Gateway yang memungkinkan koneksi ke Cloud
- **Cloud** - Remote servers yang berada di data centre



PERTUMBUHAN IOT

Cisco merilis bahwa telah ada 31 billion connected devices di tahun 2020 dan akan menjadi 75 billion devices by 2025.



Problem dalam development

Human Resource
Technology /Tools
Time
Place
Financial

Step Pendekatan aplikasi iot

1. Exploration phase

Mengidentifikasi apa yang penting untuk hari ini

2. Prototyping phase

Merubah ide menjadi protoype, experiment dengan kit sederhana seperti Raspberry pi dan arduino

Field test phase

Penggunakan solusi IoT ke dalam lingkungan bisnis sesungguhnya. Perhatian terhadap kompetisi, pemilihan teknologi dan regulasi

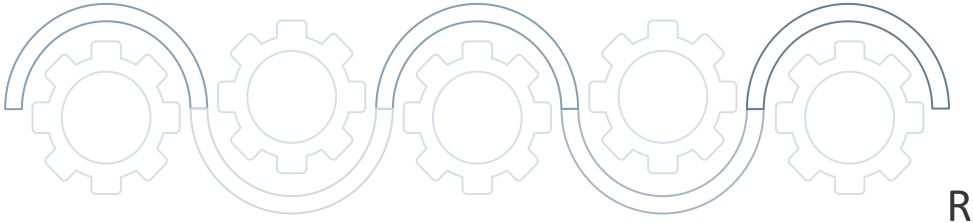
Transformation phase

Transformasi bisnis menjadi solusi Total/keseluruhan menggunakan cloud based IoT. Aspek bisnis sangat diperhatikan

Exploration phase

Bertemu dengan expert dibidangnya dan team bisnis yang mengerti, kemudian tanyakan
Apa yang paling penting hari ini?
Apa yang memerlukan koneksi?
Untuk menemukan Ide

Prototyping phase



CONNECTIVITY

Banyak standar koneksi yang biasa digunakan tergantung dari kebutuhan

POWER MANAGEMENT

Supply Daya menggunakan baterai, energy harvesting.

SECURITY

Hardware security dan protokol yang aman/secure.

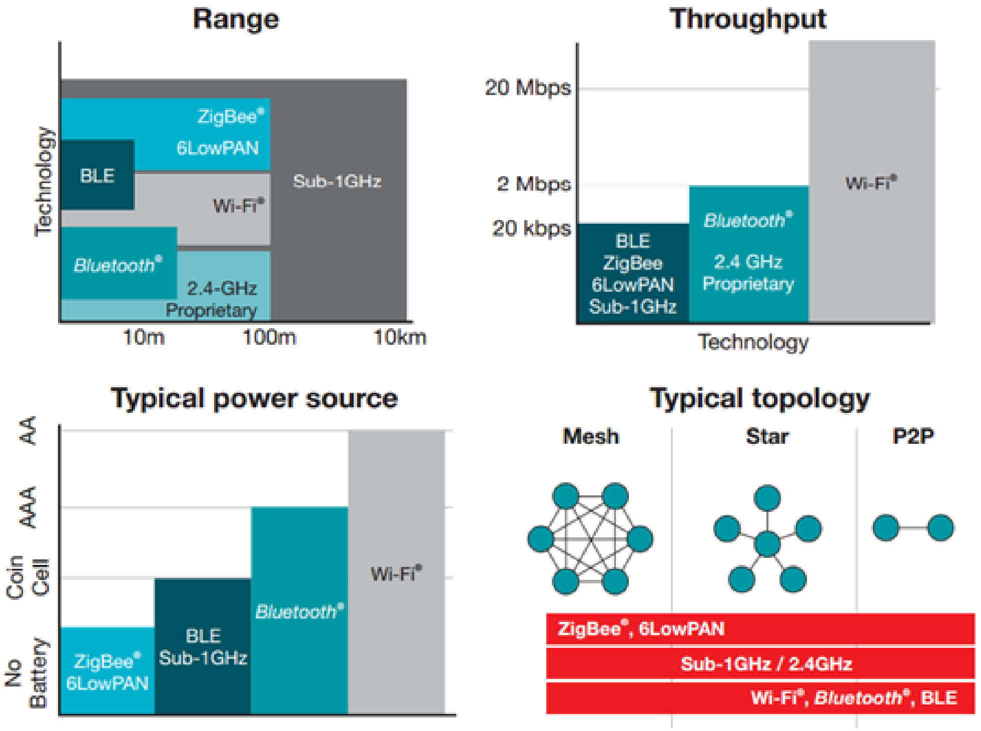
COMPLEXITY

Kemudahan desain dan development

RAPID EVOLUTION

Flexibilitas bisa digunakan di berbagai aplikasi

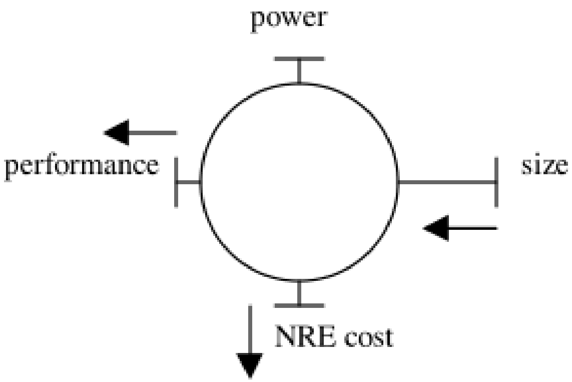
PARAMETER CONNECTIVITY



- Range
- Throughput
- Power source
- Topology

Esp32 board dev

Perbandingan development board



ESP32 **STM32** **Atmel**

Perbandingan Prosesor

Model	Clock	Flash	SRAM
ATMega328 (Arduino Nano)	16 Mhz	32 kB	2 kB
STM32F103C8T (Blue Pill)	72 Mhz	64 kB	20 kB
LPC1769 (LPCXpresso)	100 MHz	512 kB	64 kB
ESP32	240 MHz (600 MIPS)	External ~16 MB (tipikal 4 MB)	520 kB
ESP8266	80 ~ 160 MHz	External ~ 16 MiB	80 kB

ESP32 Features and Specifications

- Wireless connectivity WiFi: 150.0 Mbps data rate with HT40
- Bluetooth: BLE (Bluetooth Low Energy) and Bluetooth Classic
- Processor: Tensilica Xtensa Dual-Core 32-bit LX6 microprocessor, running at 160 or 240 MHz
- ROM: 448 KB
- SRAM: 520 KB
- Low Power: ensures that you can still use ADC conversions, for example, during deep sleep.

Peripheral Input/Output:

- Peripheral interface with DMA that includes capacitive touch
- ADCs (Analog-to-Digital Converter)
- DACs (Digital-to-Analog Converter)
- I²C (Inter-Integrated Circuit)
- UART (Universal Asynchronous Receiver/Transmitter)
- SPI (Serial Peripheral Interface)
- I²S (Integrated Interchip Sound)
- RMII (Reduced Media-Independent Interface)
- PWM (Pulse-Width Modulation).

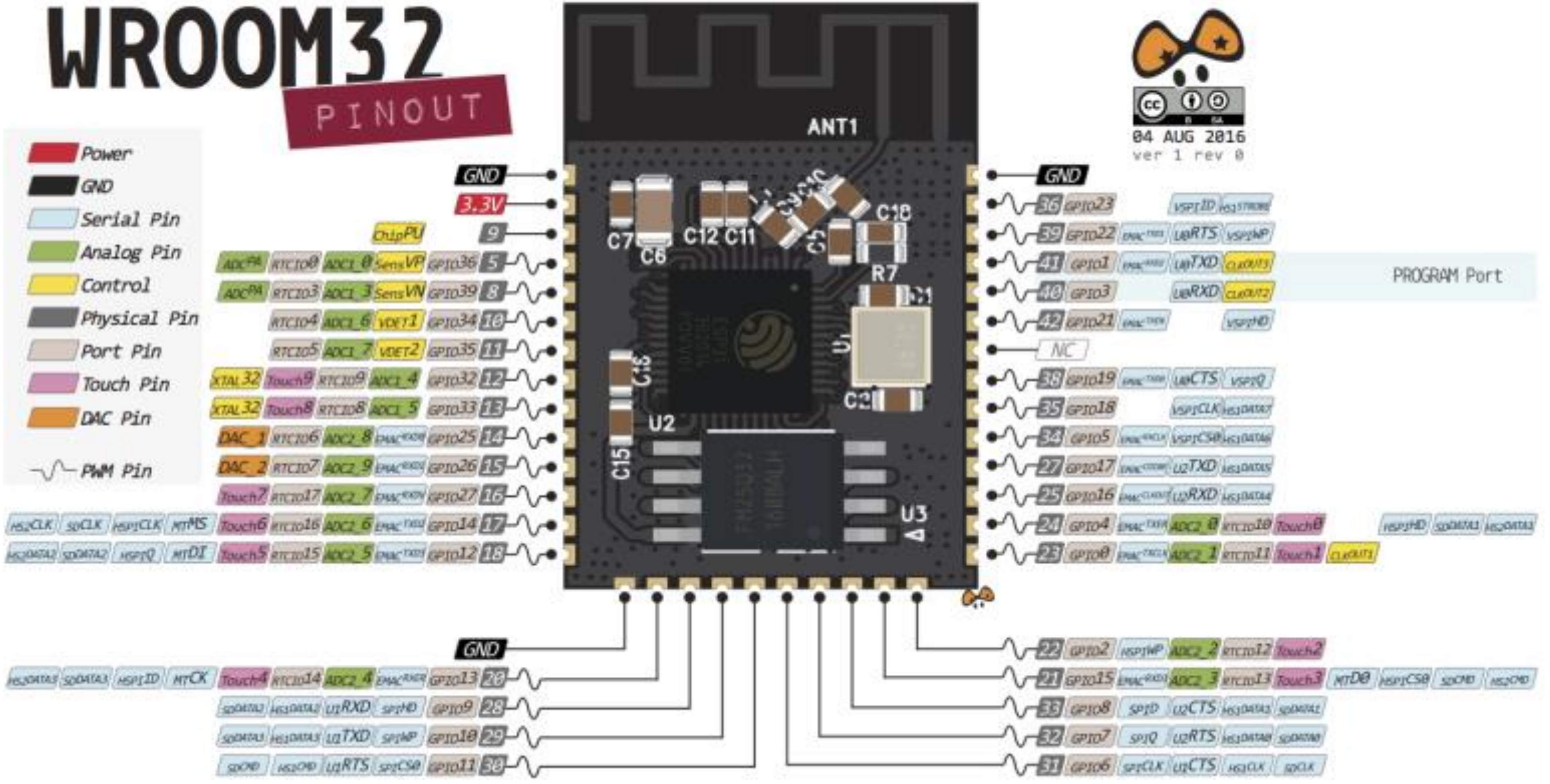
Program Env

Arduino IDE
Espressif IDF
Micropython
JavaScript
LUA

(Windows, Mac OS X and Linux)

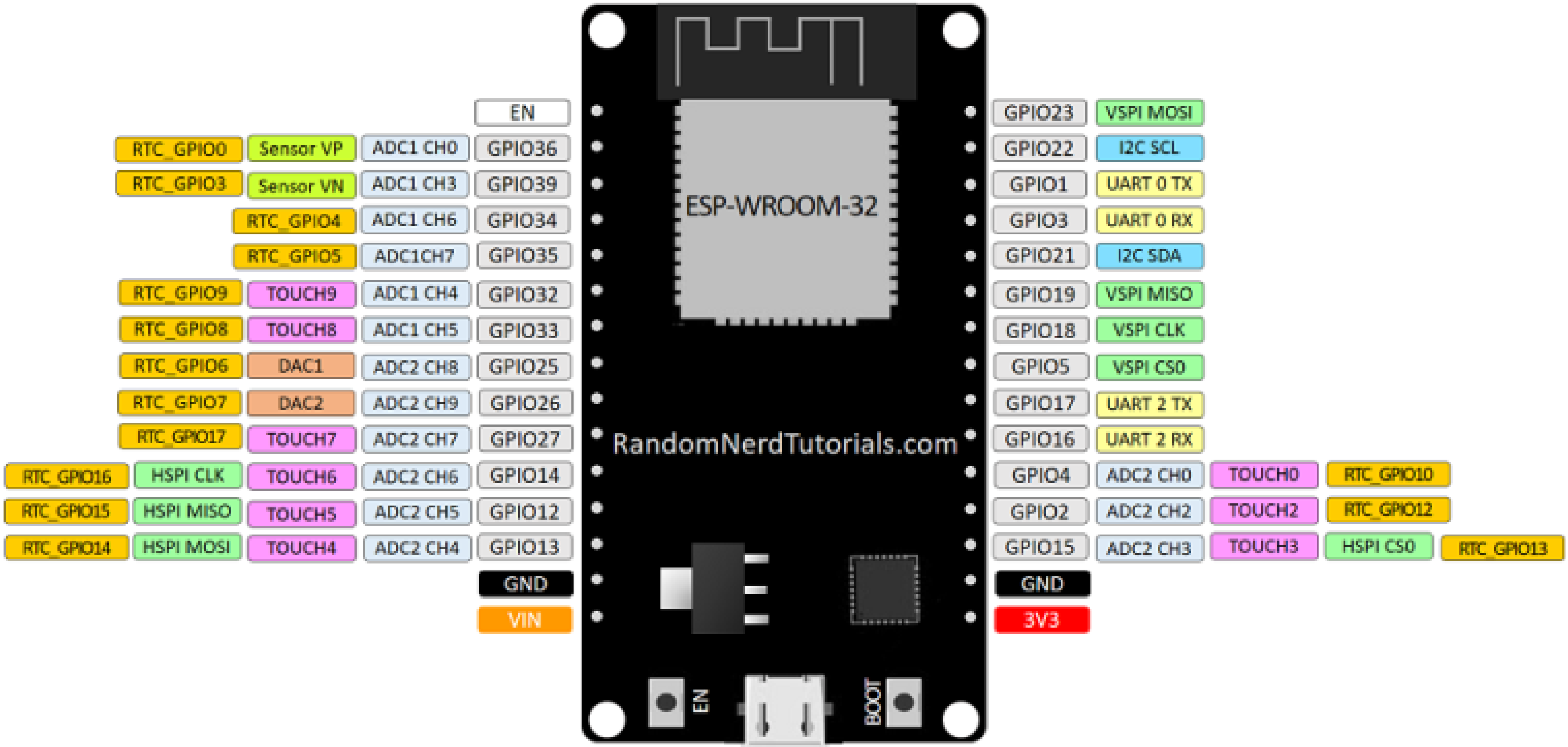
Development board

WROOM32 PINOUT


Power
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin
PWM Pin
ANT1
C7 C6 C12 C11 C5 C18 R7 C1 U1 C2 C16 C15 U2 U3
04 AUG 2016
ver 1 rev 0
GND
3.3V
ChipPU 9
ADC PA RTCIO0 ADC1_0 SensVP GPIO36 5
ADC PA RTCIO3 ADC1_3 SensVN GPIO39 8
RTCIO4 ADC1_6 VDET1 GPIO34 10
RTCIO5 ADC1_7 VDET2 GPIO35 11
XTAL32 Touch9 RTCIO9 ADC1_4 GPIO32 12
XTAL32 Touch8 RTCIO8 ADC1_5 GPIO33 13
DAC_1 RTCIO6 ADC2_8 EMAC RXD0 GPIO25 14
DAC_2 RTCIO7 ADC2_9 EMAC RXD1 GPIO26 15
Touch7 RTCIO17 ADC2_7 EMAC RXDV GPIO27 16
HS2CLK SDCLK HSPICLK MTMS Touch6 RTCIO16 ADC2_6 EMAC TXD2 GPIO14 17
HS2DATA2 SDDATA2 HSPIQ MTDI Touch5 RTCIO15 ADC2_5 EMAC TXD3 GPIO12 18
GND
36 GPIO23 VSPID HS1STROBE
39 GPIO22 EMAC TXD1 U0RTS VSPIWP
41 GPIO1 EMAC RXD2 U0TXD CLKOUT3
40 GPIO3 U0RXD CLKOUT2
PROGRAM Port
42 GPIO21 EMAC TXEN VSPIHD
NC
38 GPIO19 EMAC TXD0 U0CTS VSPIQ
35 GPIO18 VSPICLK HS1DATA7
34 GPIO5 EMAC RXCLK VSPICS0 HS1DATA6
27 GPIO17 EMAC CLK180 U2TXD HS1DATA5
25 GPIO16 EMAC CLKOUT U2RXD HS1DATA4
24 GPIO4 EMAC TXER ADC2_0 RTCIO10 Touch0 HSPIHD SDDATA1 HS2DATA1
23 GPIO0 EMAC TXCLK ADC2_1 RTCIO11 Touch1 CLKOUT1
GND
HS2DATA3 SDDATA3 HSPIID MTCK Touch4 RTCIO14 ADC2_4 EMAC RXER GPIO13 20
SDDATA2 HS1DATA2 U1RXD SPIHD GPIO9 28
SDDATA3 HS1DATA3 U1TXD SPIWP GPIO10 29
SDCMD HS1CMD U1RTS SPICS0 GPIO11 30
22 GPIO2 HSPIWP ADC2_2 RTCIO12 Touch2
21 GPIO15 EMAC RXD3 ADC2_3 RTCIO13 Touch3 MTDO HSPICS0 SDCMD HS2CMD
33 GPIO8 SPID U2CTS HS1DATA1 SDDATA1
32 GPIO7 SPIQ U2RTS HS1DATA0 SDDATA0
31 GPIO6 SPICLK U1CTS HS1CLK SDCLK

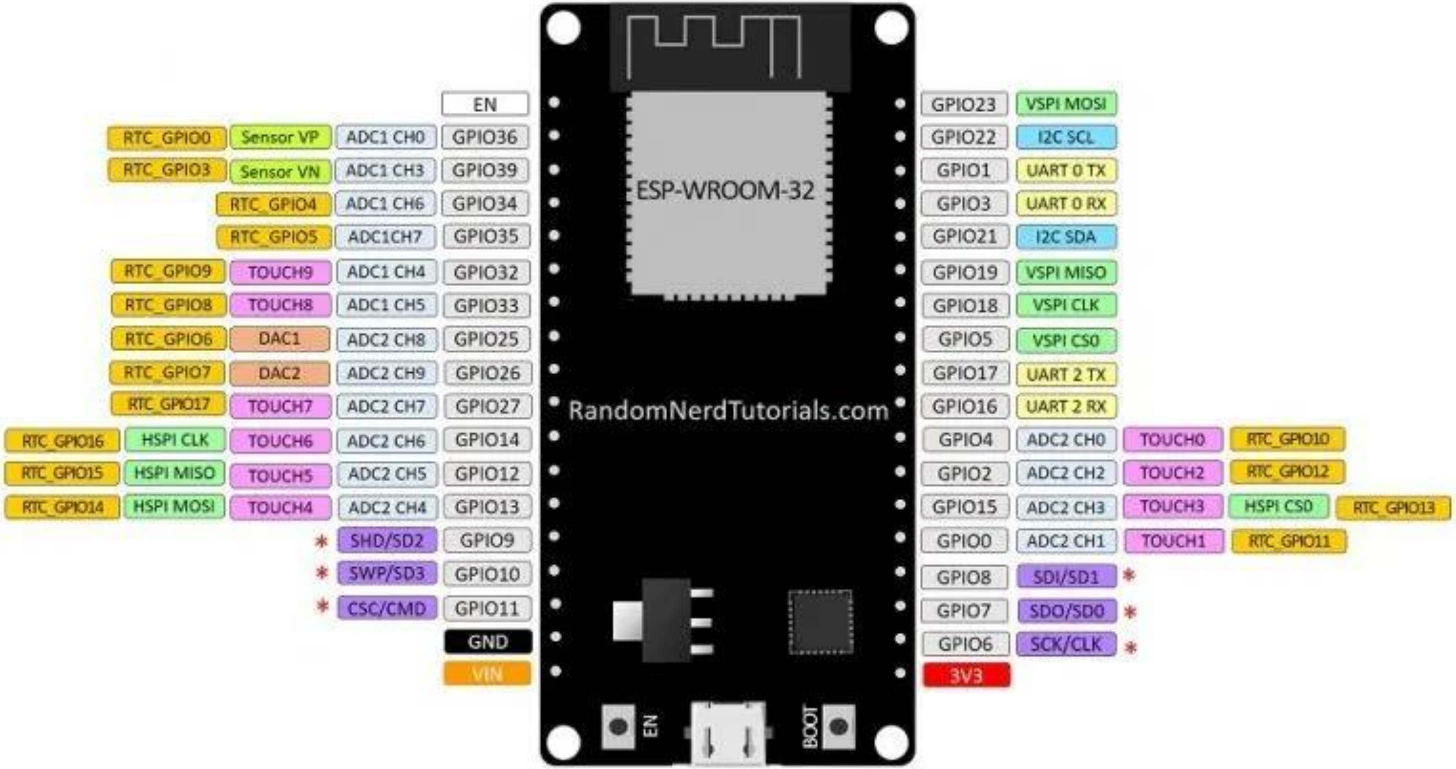
ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

ESP32 Pinout Reference

Input only pins

- GPIO 34
- GPIO 35
- GPIO 36
- GPIO 39

ESP32 has 18 x 12 bits ADC input channels (while the ESP8266 only has 1x 10 bits ADC).

ADC1_CH0 (GPIO 36) ADC1_CH1 (GPIO 37)
ADC1_CH2 (GPIO 38) ADC1_CH3 (GPIO 39)
ADC1_CH4 (GPIO 32) ADC1_CH5 (GPIO 33)
ADC1_CH6 (GPIO 34) ADC1_CH7 (GPIO 35)
ADC2_CH0 (GPIO 4) ADC2_CH1 (GPIO 0)
ADC2_CH2 (GPIO 2) ADC2_CH3 (GPIO 15)
ADC2_CH4 (GPIO 13) ADC2_CH5 (GPIO 12)
ADC2_CH6 (GPIO 14) ADC2_CH7 (GPIO 27)
ADC2_CH8 (GPIO 25) ADC2_CH9 (GPIO 26)

There are 2 x 8 bits DAC channels on the ESP32 to convert digital signals into analog voltage signal outputs. These are the DAC channels:

- DAC1 (GPIO25)
- DAC2 (GPIO26)

ESP32 PINOUT REFERENCE

Strapping Pins

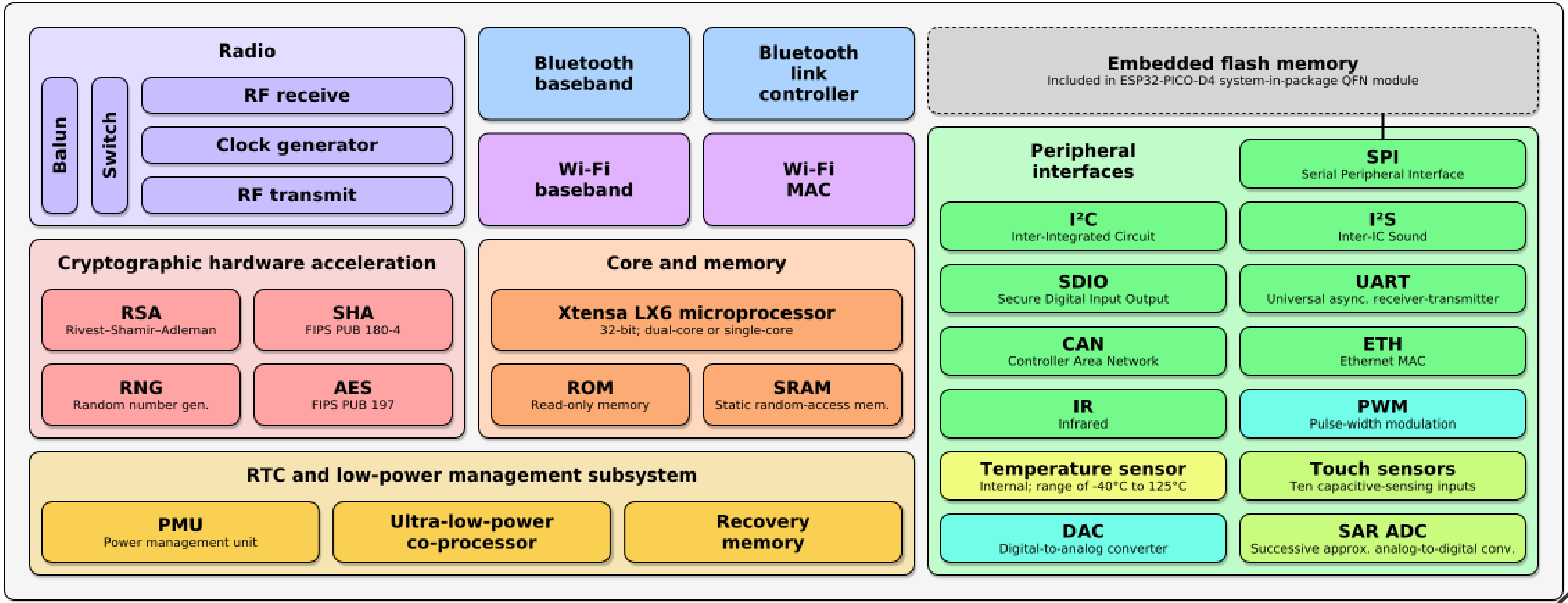
The ESP32 chip has the following strapping pins:

- GPIO 0
- GPIO 2
- GPIO 4
- GPIO 5 (must be HIGH during boot)
- GPIO 12 (must be LOW during boot)
- GPIO 15 (must be HIGH during boot)

Pins HIGH at Boot

- GPIO 1
- GPIO 3
- GPIO 5
- GPIO 6 to GPIO 11 (connected to the ESP32 integrated SPI flash memory – not recommended to use).
- GPIO 14
- GPIO 15

Espressif ESP32 Wi-Fi & Bluetooth Microcontroller — Function Block Diagram



Radio

Balun

Switch

RF receive

Clock generator

RF transmit

Bluetooth baseband

Bluetooth link controller

Wi-Fi baseband

Wi-Fi MAC

Embedded flash memory
Included in ESP32-PICO-D4 system-in-package QFN module

Peripheral interfaces

SPI
Serial Peripheral Interface

I²C
Inter-Integrated Circuit

I²S
Inter-IC Sound

SDIO
Secure Digital Input Output

UART
Universal async. receiver-transmitter

CAN
Controller Area Network

ETH
Ethernet MAC

IR
Infrared

PWM
Pulse-width modulation

Temperature sensor
Internal; range of -40°C to 125°C

Touch sensors
Ten capacitive-sensing inputs

DAC
Digital-to-analog converter

SAR ADC
Successive approx. analog-to-digital conv.

Cryptographic hardware acceleration

RSA
Rivest-Shamir-Adleman

SHA
FIPS PUB 180-4

RNG
Random number gen.

AES
FIPS PUB 197

Core and memory

Xtensa LX6 microprocessor
32-bit; dual-core or single-core

ROM
Read-only memory

SRAM
Static random-access mem.

RTC and low-power management subsystem

PMU
Power management unit

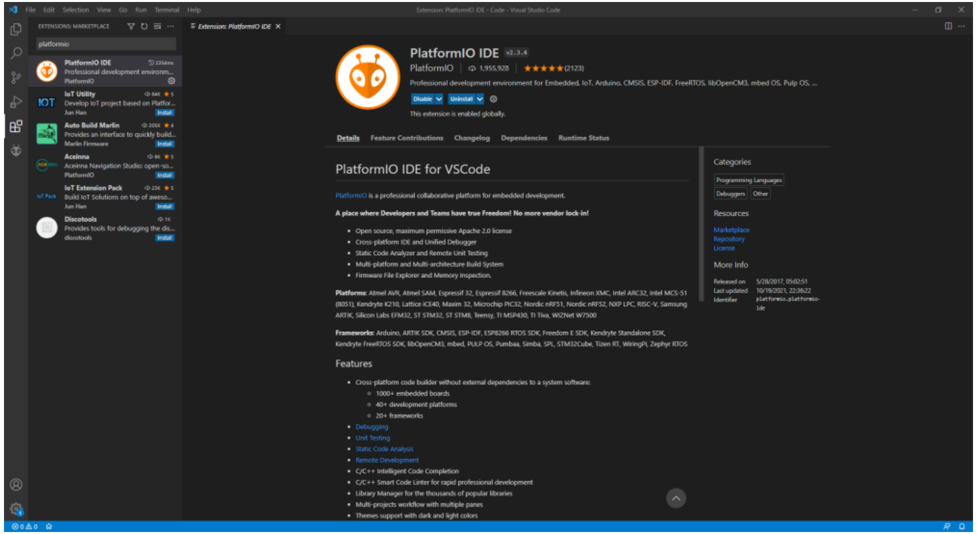
Ultra-low-power co-processor

Recovery memory

4. ESP 32 Sensor read

Platform IO IDE

PlatformIO IDE berjalan diatas VSCode sebagai official extentions
Pada menu Extention Manager pada sidebar IDE VScode– search platformIO – pilih install



PlatformIO IDE 2354ms
Professional development environm...
PlatformIO

IoT Utility 64K ★ 5
Develop IoT project based on Platfor...
Jun Han

Auto Build Marlin 205K ★ 4
Provides an interface to quickly build...
Marlin Firmware

Aceinna 6K ★ 5
Aceinna Navigation Studio: open-so...
PlatformIO

IoT Extension Pack 23K ★ 5
Build IoT Solutions on top of aweso...
Jun Han

Discotools 1K
Provides tools for debugging the dis...
discotools

PlatformIO IDE v2.3.4
PlatformIO | 1,955,928 | ★★★★★ (2123)
Professional development environment for Embedded, IoT, Arduino, CMSIS, ESP-IDF, FreeRTOS, libOpenCM3, mbed OS, Pulp OS, ...

This extension is enabled globally.

PlatformIO IDE for VSCode

PlatformIO is a professional collaborative platform for embedded development.

A place where Developers and Teams have true Freedom! No more vendor lock-in!

- Open source, maximum permissive Apache 2.0 license
- Cross-platform IDE and Unified Debugger
- Static Code Analyzer and Remote Unit Testing
- Multi-platform and Multi-architecture Build System
- Firmware File Explorer and Memory Inspection.

Platforms: Atmel AVR, Atmel SAM, Espressif 32, Espressif 8266, Freescale Kinetis, Infineon XMC, Intel ARC32, Intel MCS-51 (8051), Kendryte K210, Lattice iCE40, Maxim 32, Microchip PIC32, Nordic nRF51, Nordic nRF52, NXP LPC, RISC-V, Samsung ARTIK, Silicon Labs EFM32, ST STM32, ST STM8, Teensy, TI MSP430, TI Tiva, WIZNet W7500

Frameworks: Arduino, ARTIK SDK, CMSIS, ESP-IDF, ESP8266 RTOS SDK, Freedom E SDK, Kendryte Standalone SDK, Kendryte FreeRTOS SDK, libOpenCM3, mbed, PULP OS, Pumbaa, Simba, SPL, STM32Cube, Tizen RT, WiringPi, Zephyr RTOS

Features

- Cross-platform code builder without external dependencies to a system software:
  - 1000+ embedded boards
  - 40+ development platforms
  - 20+ frameworks
- Debugging
- Unit Testing
- Static Code Analysis
- Remote Development
- C/C++ Intelligent Code Completion
- C/C++ Smart Code Linter for rapid professional development
- Library Manager for the thousands of popular libraries
- Multi-projects workflow with multiple panes
- Themes support with dark and light colors

Categories

Programming Languages
Debuggers
Other

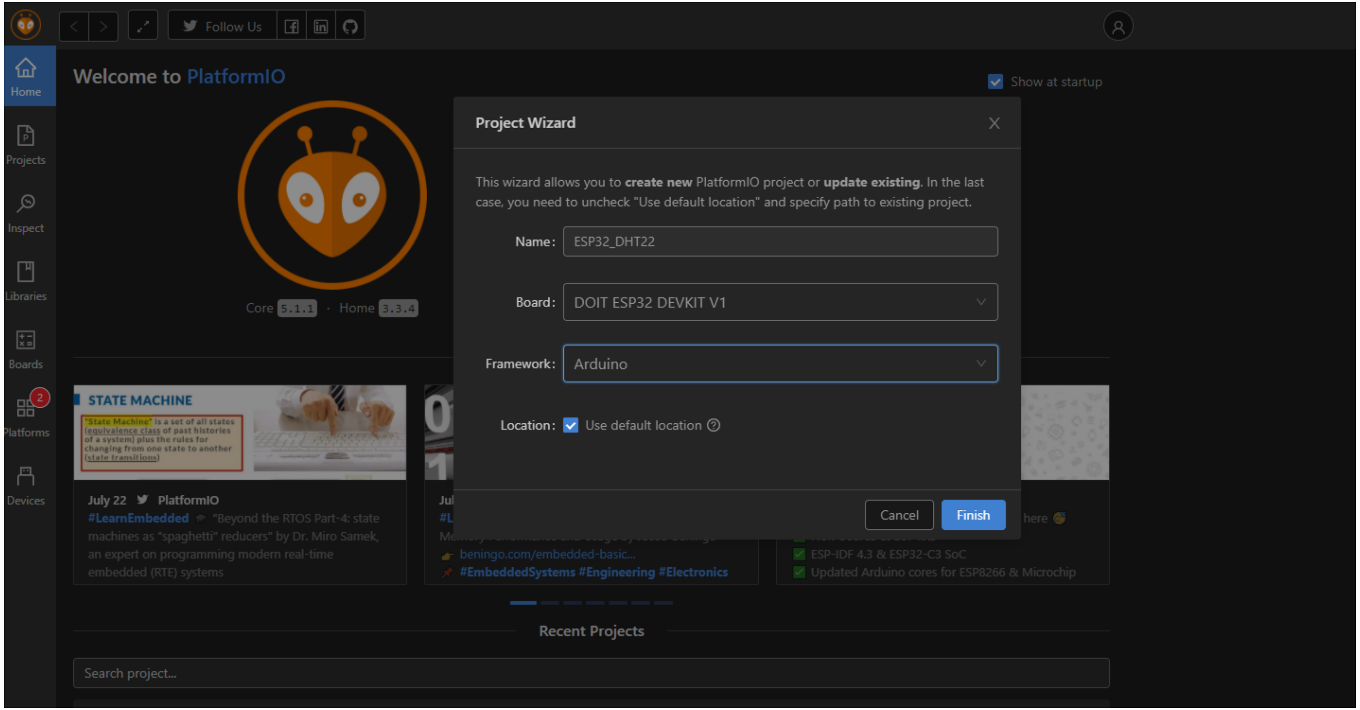
Resources

Marketplace
Repository
License

More Info

Released on	5/28/2017, 05:02:51
Last updated	10/19/2021, 22:36:22
Identifier	platformio.platformio-ide

Membuat project baru



Project Wizard

This wizard allows you to **create new** PlatformIO project or **update existing.** In the last case, you need to uncheck "Use default location" and specify path to existing project.

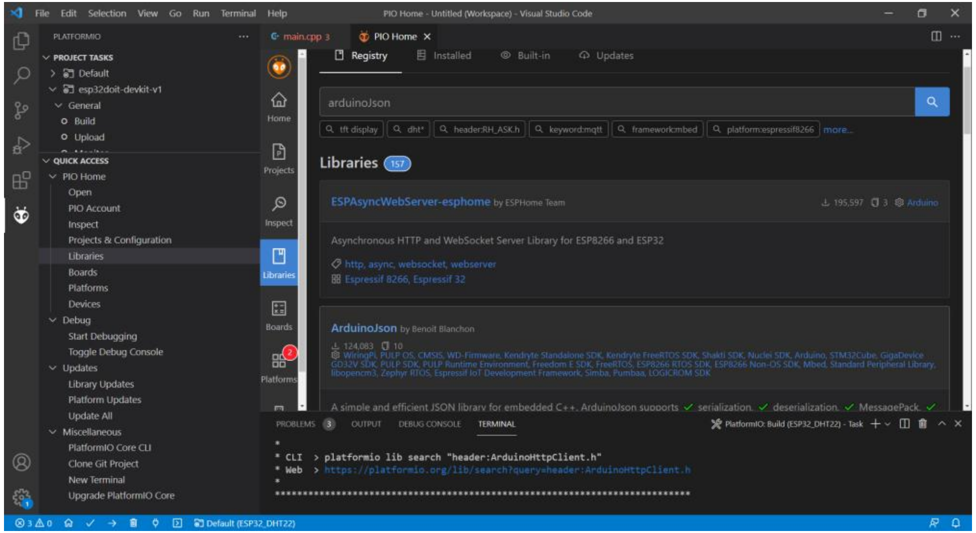
Name: ESP32_DHT22

Board: DOIT ESP32 DEVKIT V1

Framework: Arduino

Location: ☑ Use default location

Install library untuk sensor



arduinoJson

Libraries 157

ESPAsyncWebServer-esphome by ESPHome Team — 195,597 · 3 · Arduino

Asynchronous HTTP and WebSocket Server Library for ESP8266 and ESP32

http, async, websocket, webserver

Espressif 8266, Espressif 32

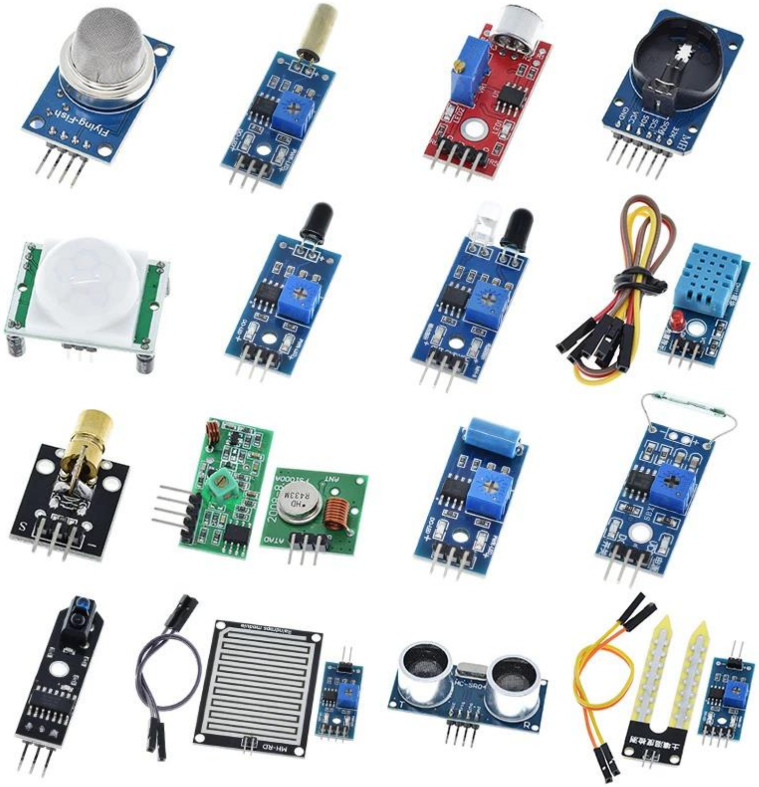
ArduinoJson by Benoit Blanchon — 124,083 · 10

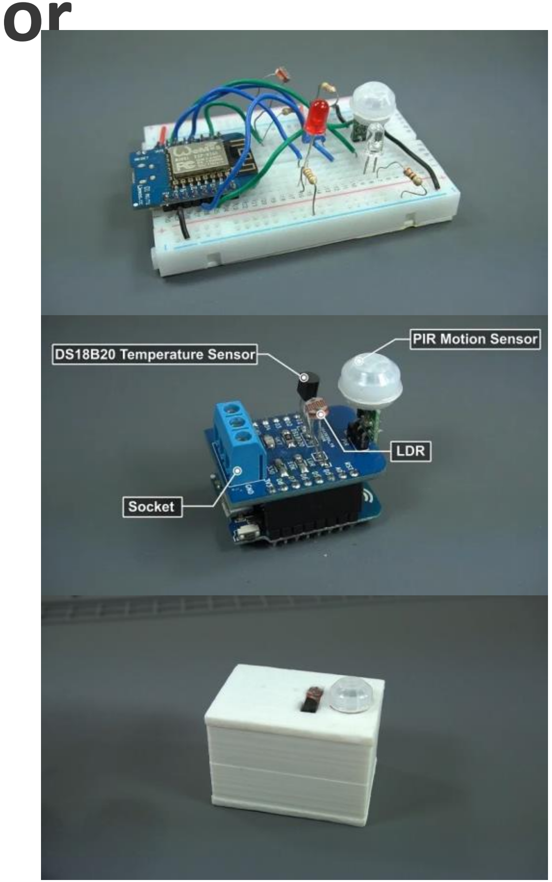
WiringPi, PULP OS, CMSIS, WD-Firmware, Kendryte Standalone SDK, Kendryte FreeRTOS SDK, Shakti SDK, Nuclei SDK, Arduino, STM32Cube, GigaDevice GD32V SDK, PULP SDK, PULP Runtime Environment, Freedom E SDK, FreeRTOS, ESP8266 RTOS SDK, ESP8266 Non-OS SDK, Mbed, Standard Peripheral Library, libopencm3, Zephyr RTOS, Espressif IoT Development Framework, Simba, Pumbaa, LOGICROM SDK

A simple and efficient JSON library for embedded C++. ArduinoJson supports ✓ serialization, ✓ deserialization, ✓ MessagePack, ✓

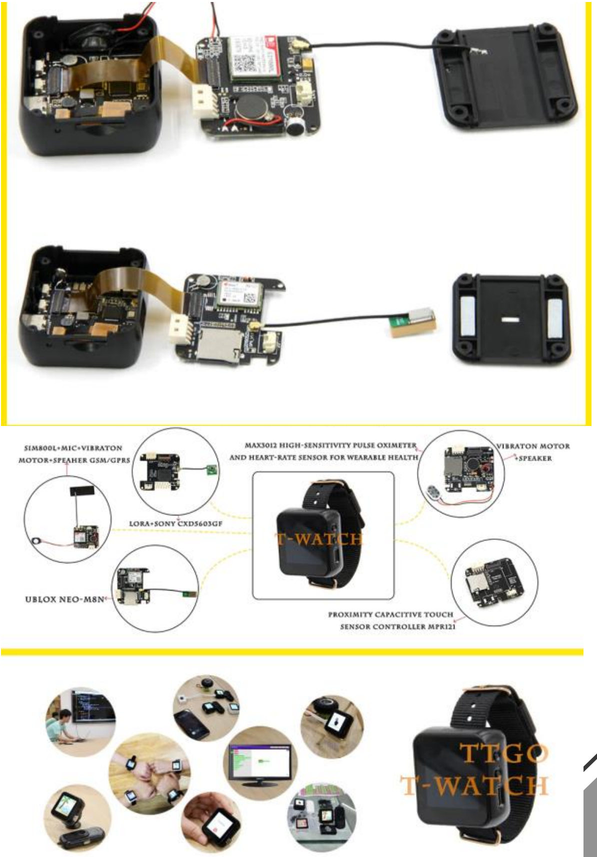
```
*
* CLI  > platformio lib search "header:ArduinoHttpClient.h"
* Web  > https://platformio.org/lib/search?query=header:ArduinoHttpClient.h
*
********************************************************************************
```

Sensor





Shield or board custom



SENSOR

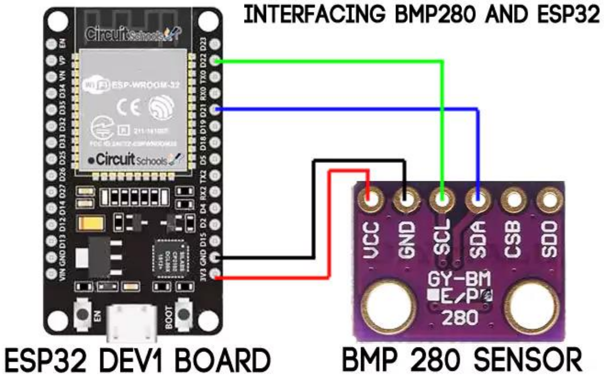
```
typedef struct
{
bool (Init)(const SensorConfig_t * const Config);
bool (Read)(const SensorObj_t * const, SensorData_t * const SensorData);
bool (*Write)(const SensorObj_t * const, SensorData_t * const
SensorData);
} Sensor_t;
```

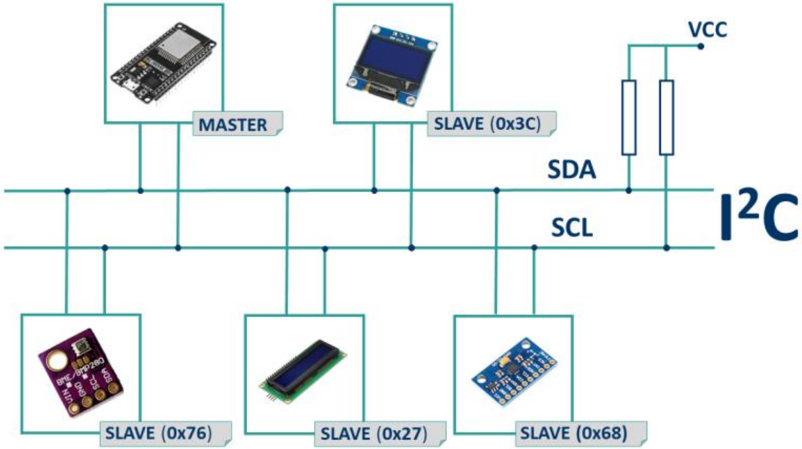
```
const Sensor_t Analog =
{
Adc_Init,
Adc_Read,
Adc_Write
};
```

```
const Sensor_t Gyro =
{
Gyro_Init,
Gyro_Read,
Gyro_Write
};
```

```
Analog.Init(AdcConfig);
Gryo.Init(GyroConfig);
```

Sensor temperature, humidity and pressure.

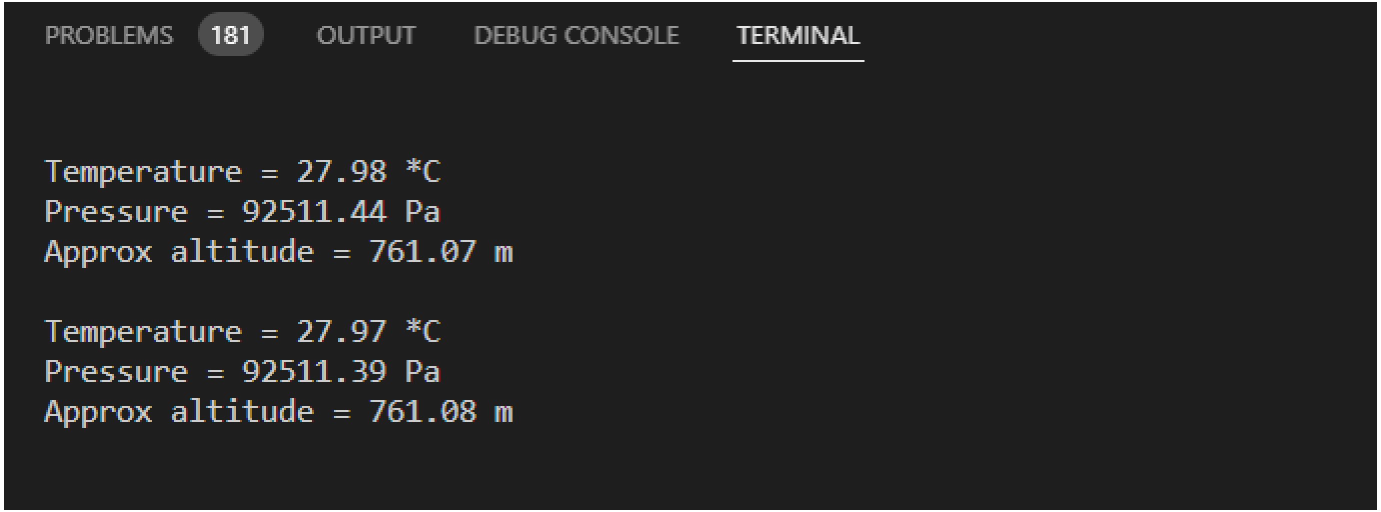




SDA	SDA (default is GPIO 21)
SCL	SCL (default is GPIO 22)

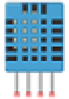
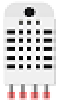
```
#for using different wire
Wire.begin(I2C_SDA, I2C_SCL);
```

BMP280 Test

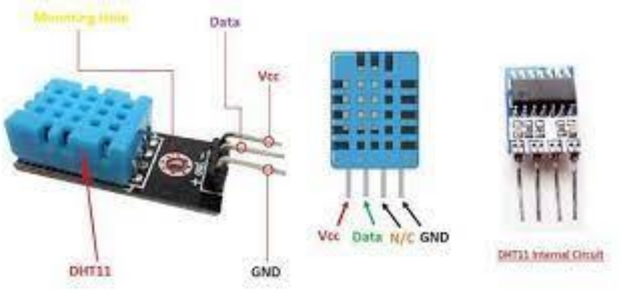


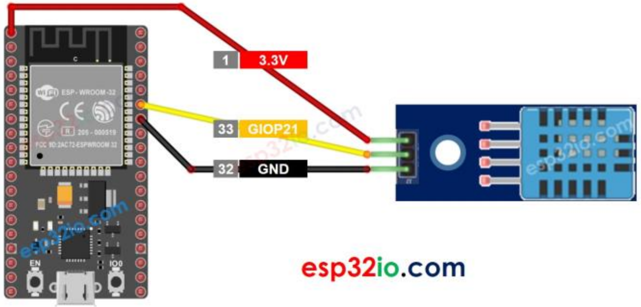
```
Temperature = 27.98 *C
Pressure = 92511.44 Pa
Approx altitude = 761.07 m

Temperature = 27.97 *C
Pressure = 92511.39 Pa
Approx altitude = 761.08 m
```

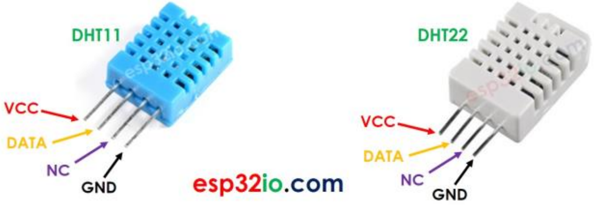
DHT11/DHT22 Temperature and Humidity

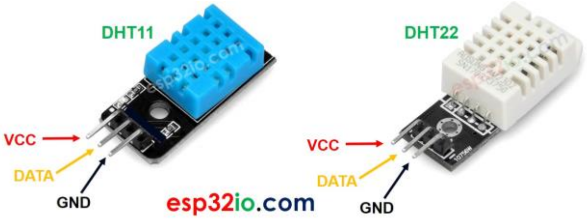
	DHT11 	DHT22 
Temperature range	0 to 50 ºC $^{+/-2\ \text{ºC}}$	-40 to 80 ºC $^{+/-0.5\text{ºC}}$
Humidity range	20 to 90% $^{+/-5\%}$	0 to 100% $^{+/-2\%}$
Resolution	Humidity: 1% Temperature: 1ºC	Humidity: 0.1% Temperature: 0.1ºC
Operating voltage	3 – 5.5 V DC	3 – 6 V DC
Current supply	0.5 – 2.5 mA	1 – 1.5 mA
Sampling period	1 second	2 seconds
Price	$1 to $5	$4 to $10

DHT 11

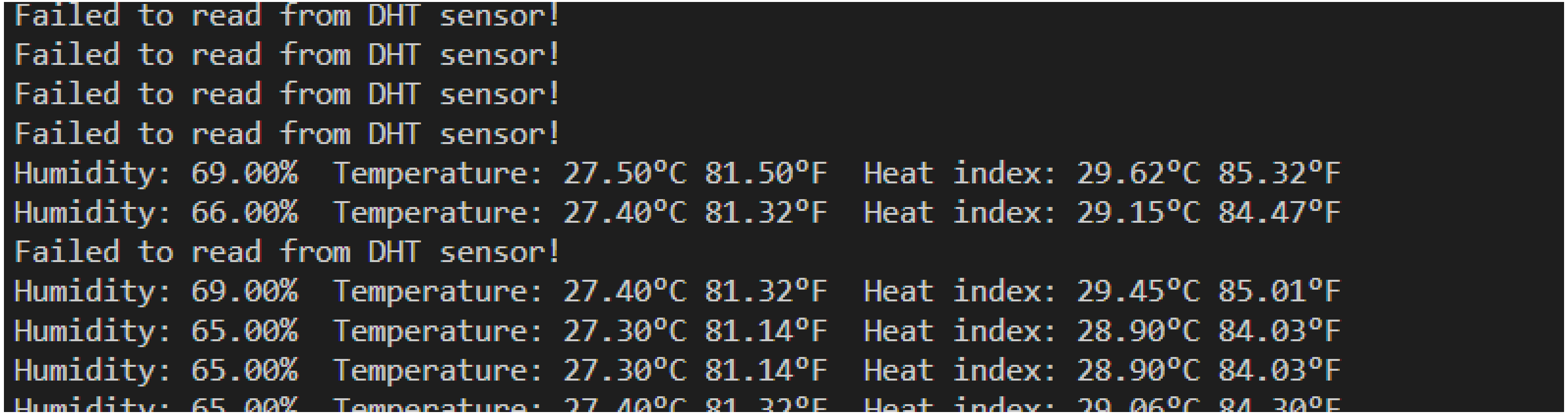

Mounting Hole
Data
Vcc
DHT11
GND
Vcc
Data
N/C
GND
DHT11 Internal Circuit


1
3.3V
33
GIOP21
32
GND
EN
IO0
esp32io.com


DHT11
VCC
DATA
NC
GND
esp32io.com
DHT22
VCC
DATA
NC
GND

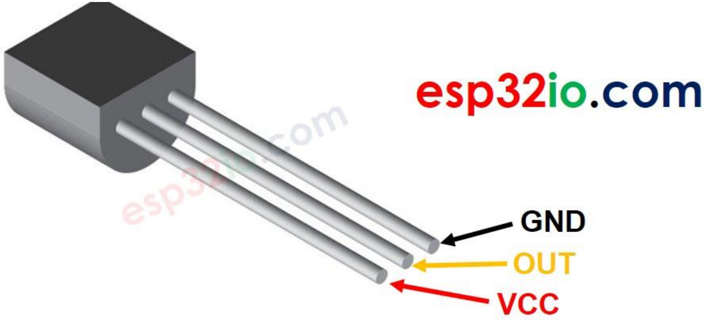

DHT11
VCC
DATA
GND
esp32io.com
DHT22
VCC
DATA
GND

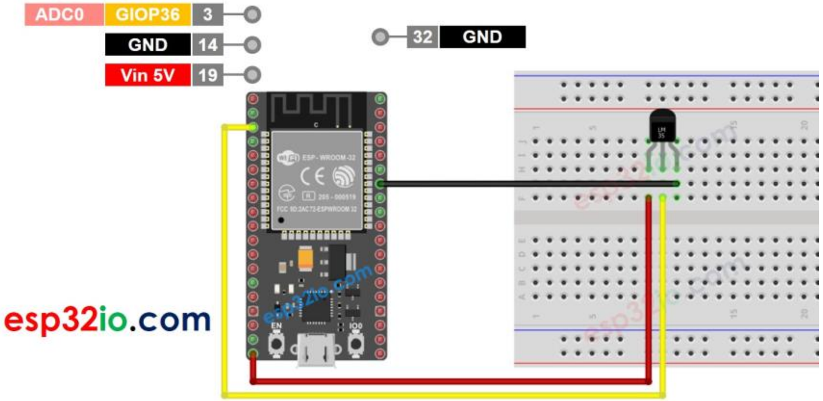
DHT 11 Test



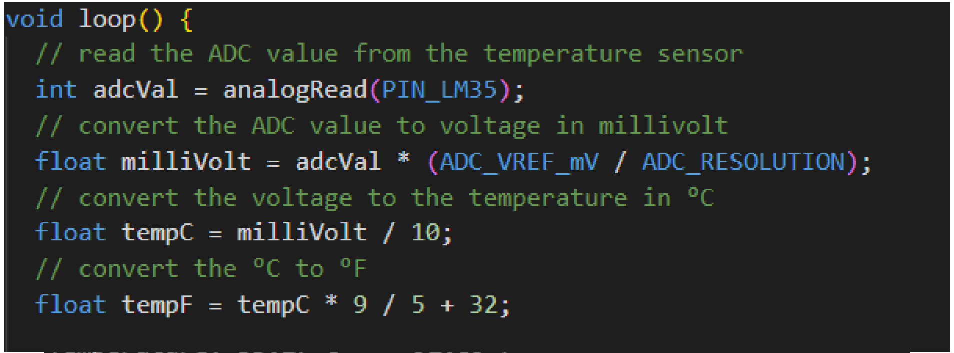
```
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Humidity: 69.00%  Temperature: 27.50°C 81.50°F  Heat index: 29.62°C 85.32°F
Humidity: 66.00%  Temperature: 27.40°C 81.32°F  Heat index: 29.15°C 84.47°F
Failed to read from DHT sensor!
Humidity: 69.00%  Temperature: 27.40°C 81.32°F  Heat index: 29.45°C 85.01°F
Humidity: 65.00%  Temperature: 27.30°C 81.14°F  Heat index: 28.90°C 84.03°F
Humidity: 65.00%  Temperature: 27.30°C 81.14°F  Heat index: 28.90°C 84.03°F
Humidity: 65.00%  Temperature: 27.40°C 81.32°F  Heat index: 29.06°C 84.30°F
```

LM35 Sensor Temperature

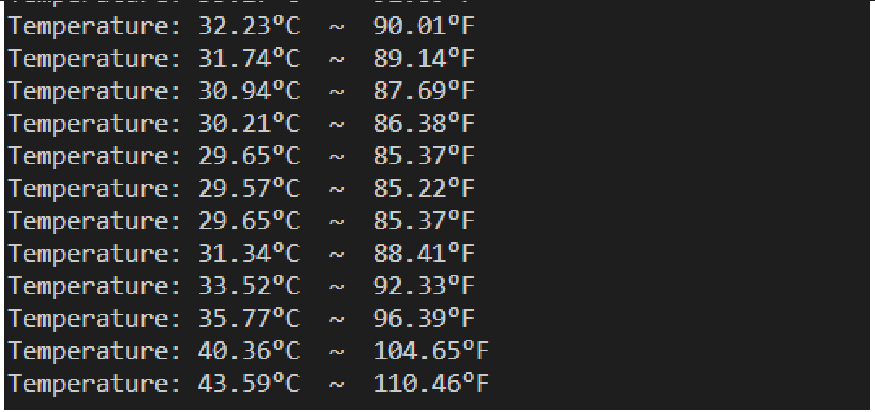




TEST LM35



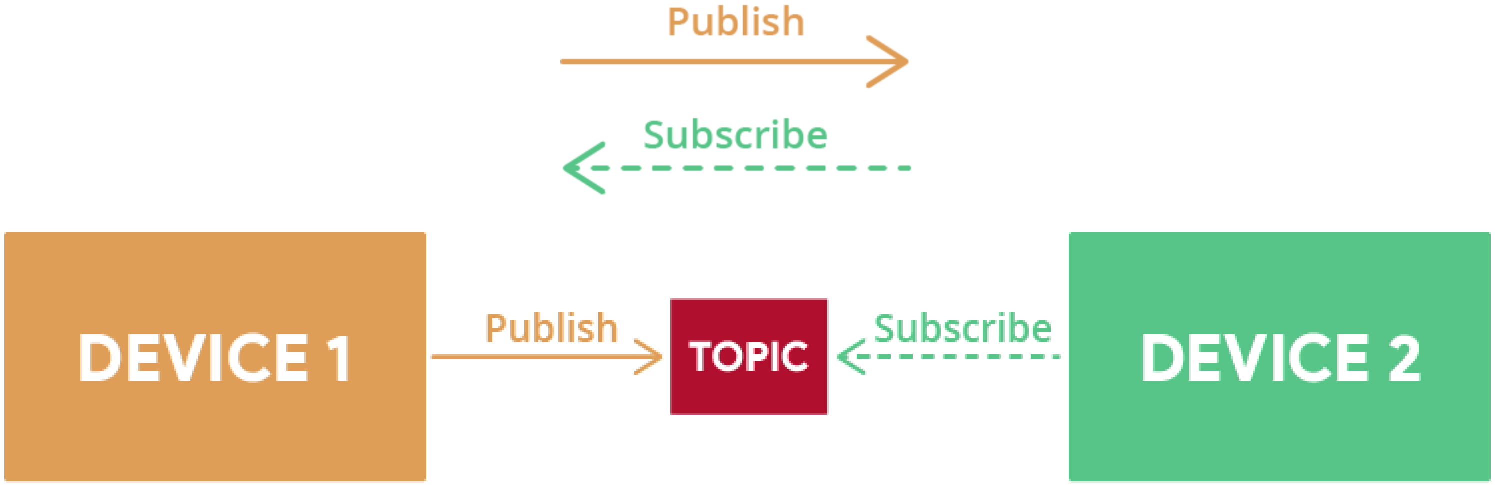
```
void loop() {
  // read the ADC value from the temperature sensor
  int adcVal = analogRead(PIN_LM35);
  // convert the ADC value to voltage in millivolt
  float milliVolt = adcVal * (ADC_VREF_mV / ADC_RESOLUTION);
  // convert the voltage to the temperature in °C
  float tempC = milliVolt / 10;
  // convert the °C to °F
  float tempF = tempC * 9 / 5 + 32;
```



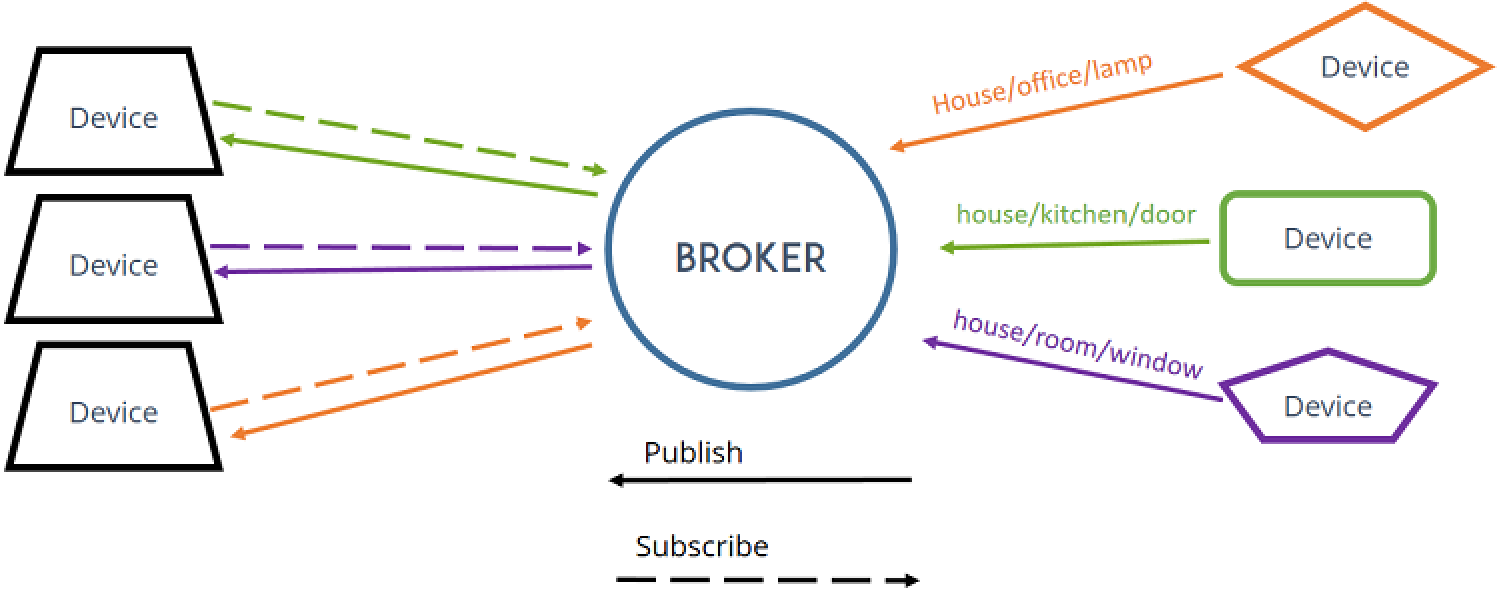
```
Temperature: 32.23°C  ~  90.01°F
Temperature: 31.74°C  ~  89.14°F
Temperature: 30.94°C  ~  87.69°F
Temperature: 30.21°C  ~  86.38°F
Temperature: 29.65°C  ~  85.37°F
Temperature: 29.57°C  ~  85.22°F
Temperature: 29.65°C  ~  85.37°F
Temperature: 31.34°C  ~  88.41°F
Temperature: 33.52°C  ~  92.33°F
Temperature: 35.77°C  ~  96.39°F
Temperature: 40.36°C  ~  104.65°F
Temperature: 43.59°C  ~  110.46°F
```

MQTT

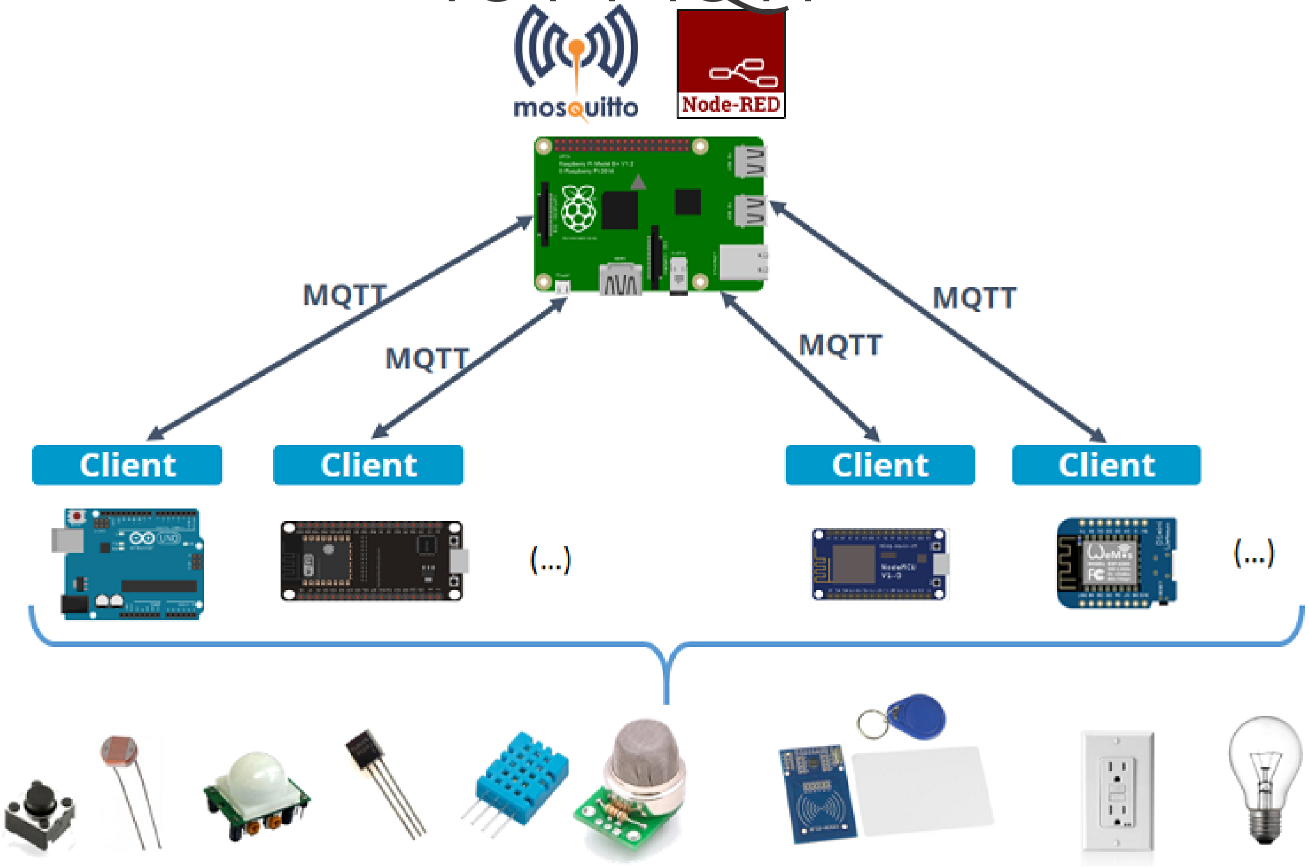
MQTT stands for Message Queuing Telemetry Transport.



MQTT Broker


Device
Device
Device
BROKER
House/office/lamp
Device
house/kitchen/door
Device
house/room/window
Device
Publish
Subscribe

IOT MQTT



mosquitto

Node-RED

MQTT

MQTT

MQTT

MQTT

Client

Client

Client

Client

(...)

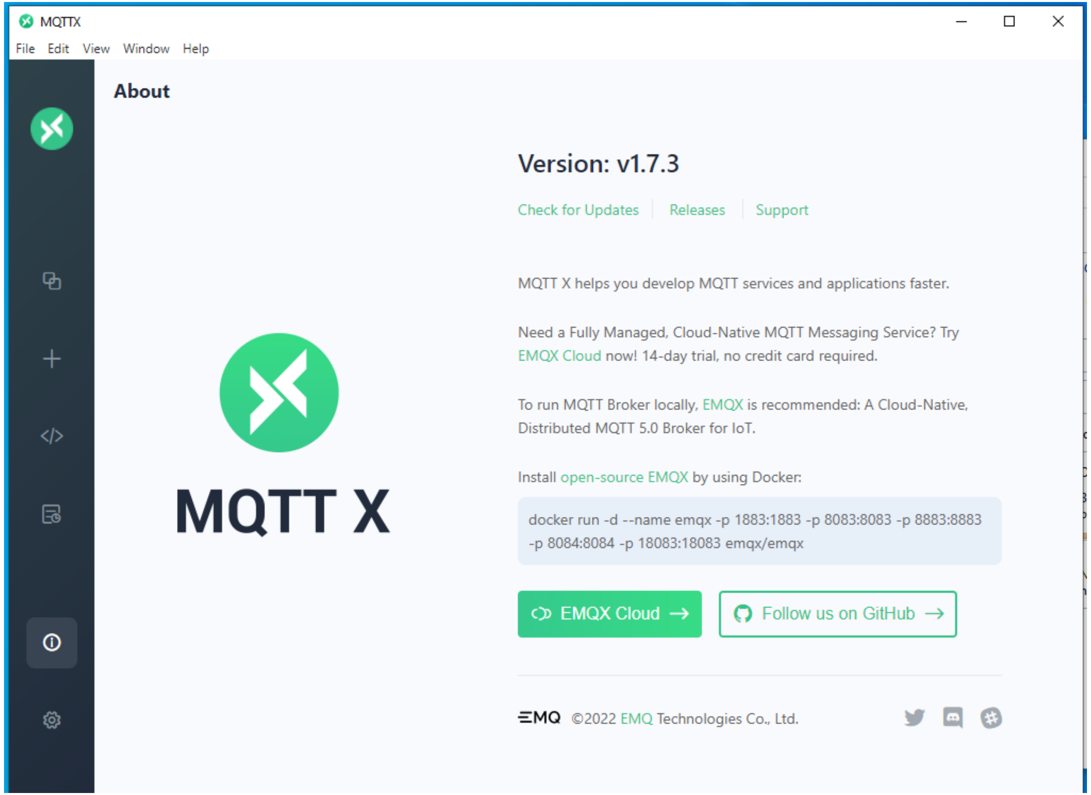
(...)

ESP MQTT

```
// WiFi
const char *ssid = "TPLINK"; // Enter your WiFi name
const char *password = "TPLINK32";  // Enter WiFi password

// MQTT Broker
const char *mqtt_broker = "broker.emqx.io";
const char *topic = "esp32/sensor/";
const char *mqtt_username = "emqx";
const char *mqtt_password = "public";
const int mqtt_port = 1883;
```

```
lib_deps =
    ;adafruit/DHT sensor library@^1.4.3
    ;adafruit/Adafruit Unified Sensor@^1.1.5
    knolleary/PubSubClient@^2.8
```

MQTT Client



About

Version: v1.7.3

Check for Updates | Releases | Support

MQTT X helps you develop MQTT services and applications faster.

Need a Fully Managed, Cloud-Native MQTT Messaging Service? Try EMQX Cloud now! 14-day trial, no credit card required.

To run MQTT Broker locally, EMQX is recommended: A Cloud-Native, Distributed MQTT 5.0 Broker for IoT.

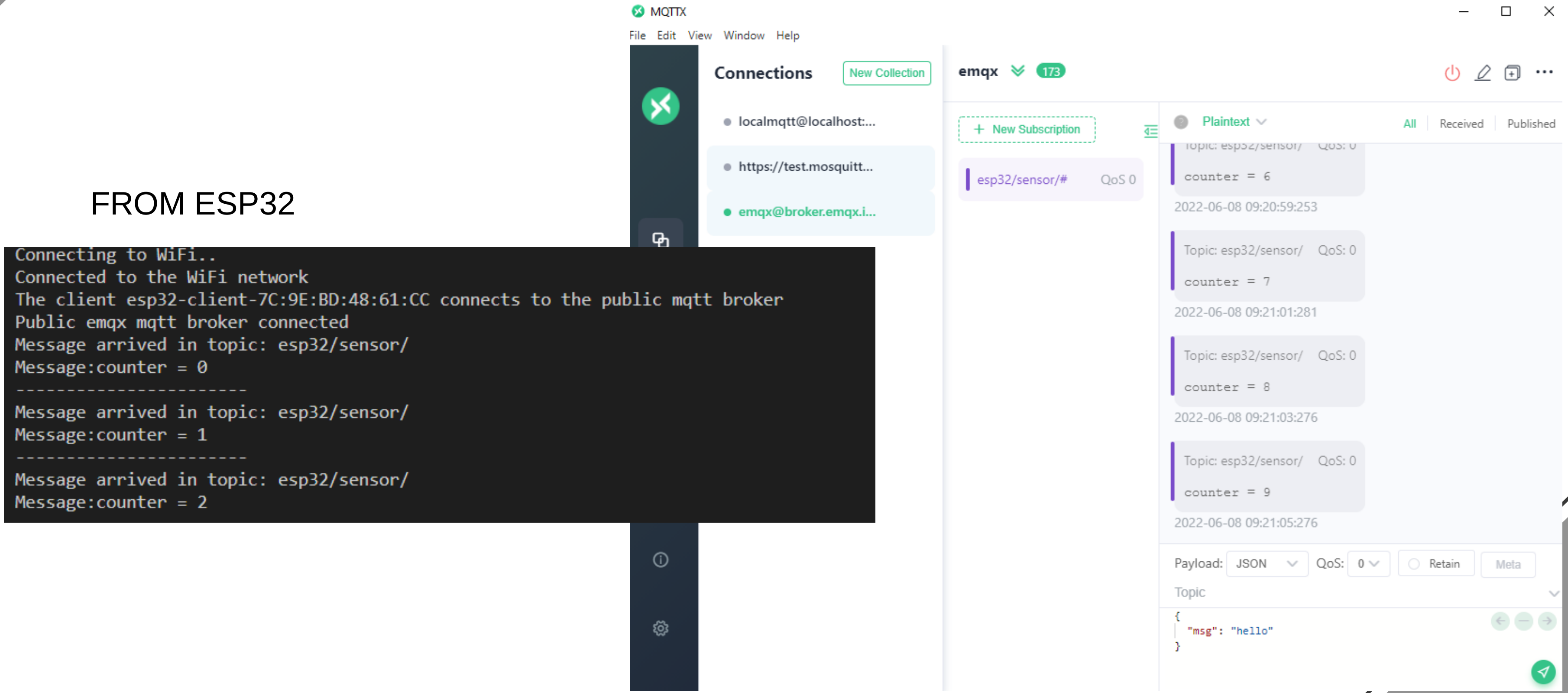
Install open-source EMQX by using Docker:

```
docker run -d --name emqx -p 1883:1883 -p 8083:8083 -p 8883:8883
-p 8084:8084 -p 18083:18083 emqx/emqx
```

ΞMQ ©2022 EMQ Technologies Co., Ltd.

TEST MQTT

FROM ESP32



```
Connecting to WiFi..
Connected to the WiFi network
The client esp32-client-7C:9E:BD:48:61:CC connects to the public mqtt broker
Public emqx mqtt broker connected
Message arrived in topic: esp32/sensor/
Message:counter = 0
-----------------------
Message arrived in topic: esp32/sensor/
Message:counter = 1
-----------------------
Message arrived in topic: esp32/sensor/
Message:counter = 2
```

Subscription: esp32/sensor/# QoS 0

```
Topic: esp32/sensor/    QoS: 0
counter = 6
2022-06-08 09:20:59:253

Topic: esp32/sensor/    QoS: 0
counter = 7
2022-06-08 09:21:01:281

Topic: esp32/sensor/    QoS: 0
counter = 8
2022-06-08 09:21:03:276

Topic: esp32/sensor/    QoS: 0
counter = 9
2022-06-08 09:21:05:276
```

Payload: JSON QoS: 0 ○ Retain

```
{
  "msg": "hello"
}
```

THANKS

Do you have any question?

hasbiida@gmail.com

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