INSTITUTO FEDERAL
Paraíba

Sallatiel Fernandes Freire Cardoso
Bacharelando em Engenharia Elétrica

ESP e ThingSpeak

Hardware, software e Cloud Computing

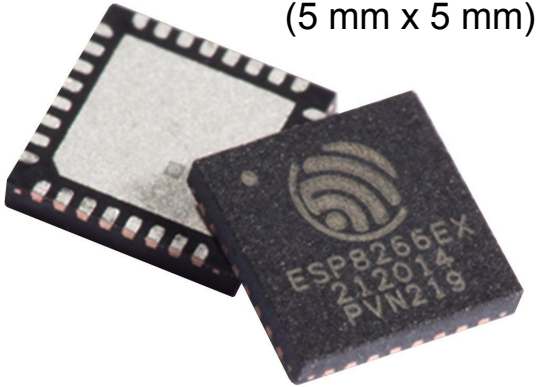
Veremos

- ESP8266
- ESP32
- Programação
- ThingSpeak

ESP8266

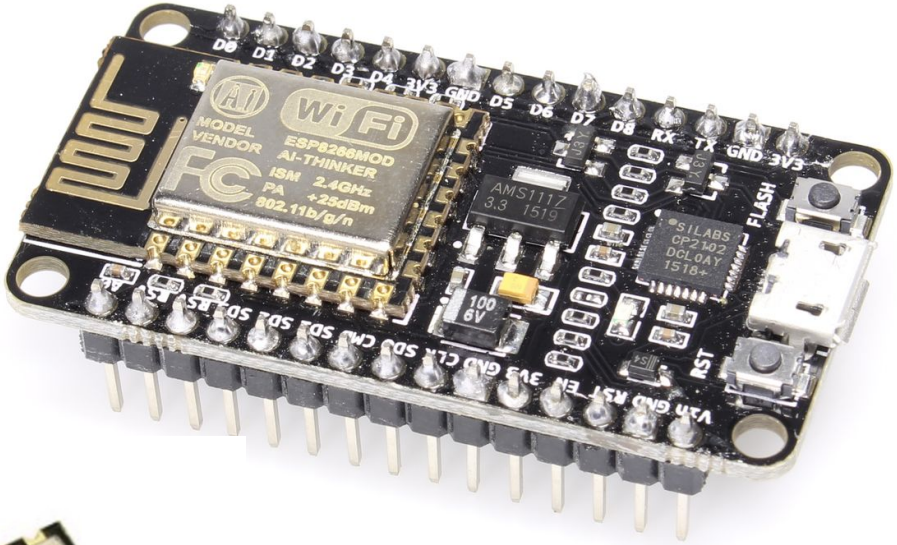
ESPRESSIF

QFN32-pin
(5 mm x 5 mm)

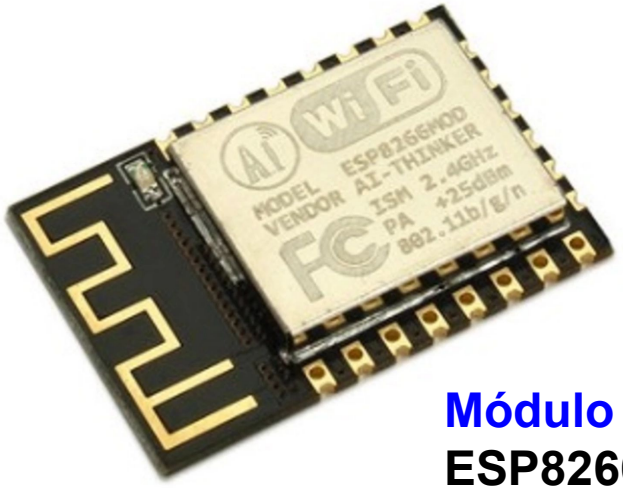


Circuito Integrado
ESP8266EX

https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf



Placa de desenvolvimento
NodeMCU



Módulo
ESP8266

Circuito Integrado
ESP8266EX

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz
- Memory:
  - 32 KiB instruction RAM
  - 32 KiB instruction cache RAM
  - 80 KiB user-data RAM
  - 16 KiB ETS system-data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
  - Integrated TR switch, balun, LNA, power amplifier and matching network
  - WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI
- I²C (software implementation)
- I²S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC - SAR ADC)

Início da Produção: 30 de dezembro de 2013

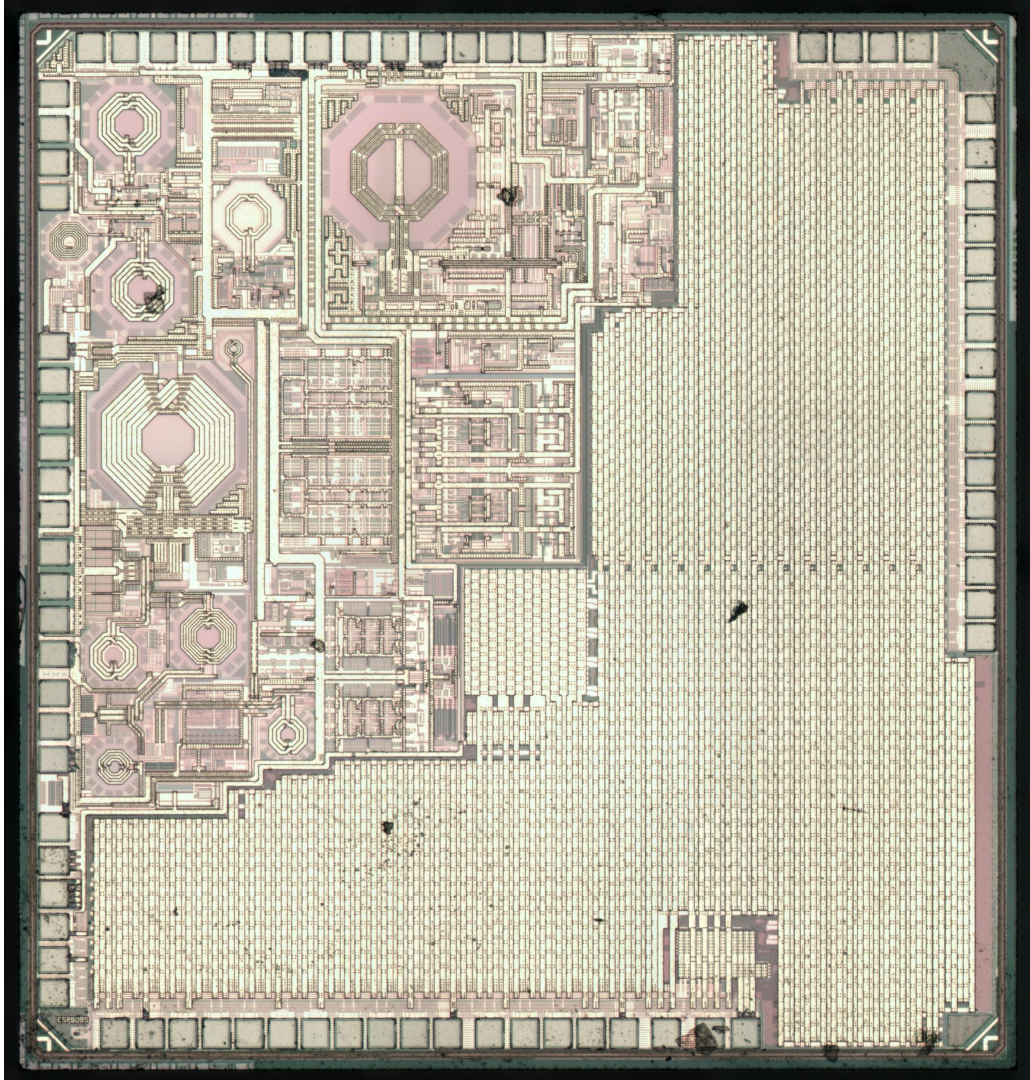
https://en.wikipedia.org/wiki/ESP8266

Espressif ESP8266 WiFi-serial interface : weekend die-shot

Since August of 2014 internet is literally blown by WiFi-serial modules on new ESP8266 chip which are currently being sold for less than 4$. Chinese company Espressif managed to cram entire WiFi, TCP/IP and HTTP stack into on-chip memory, without external DRAM. Analog front-end requires minimal external components, all filters are internal. All this allowed them to offer extremely aggressive price. Chip has marking ESP8089, which is their more advanced 40nm product. Apparently, they only differ in bonding and ROM content.

Die size 2050x2169 µm, half of which is occupied by transceiver and PA, 25% - on-chip memory (rough size estimations are ~300KiB), and the rest is Xtensa LX106 CPU core and other digital logic.

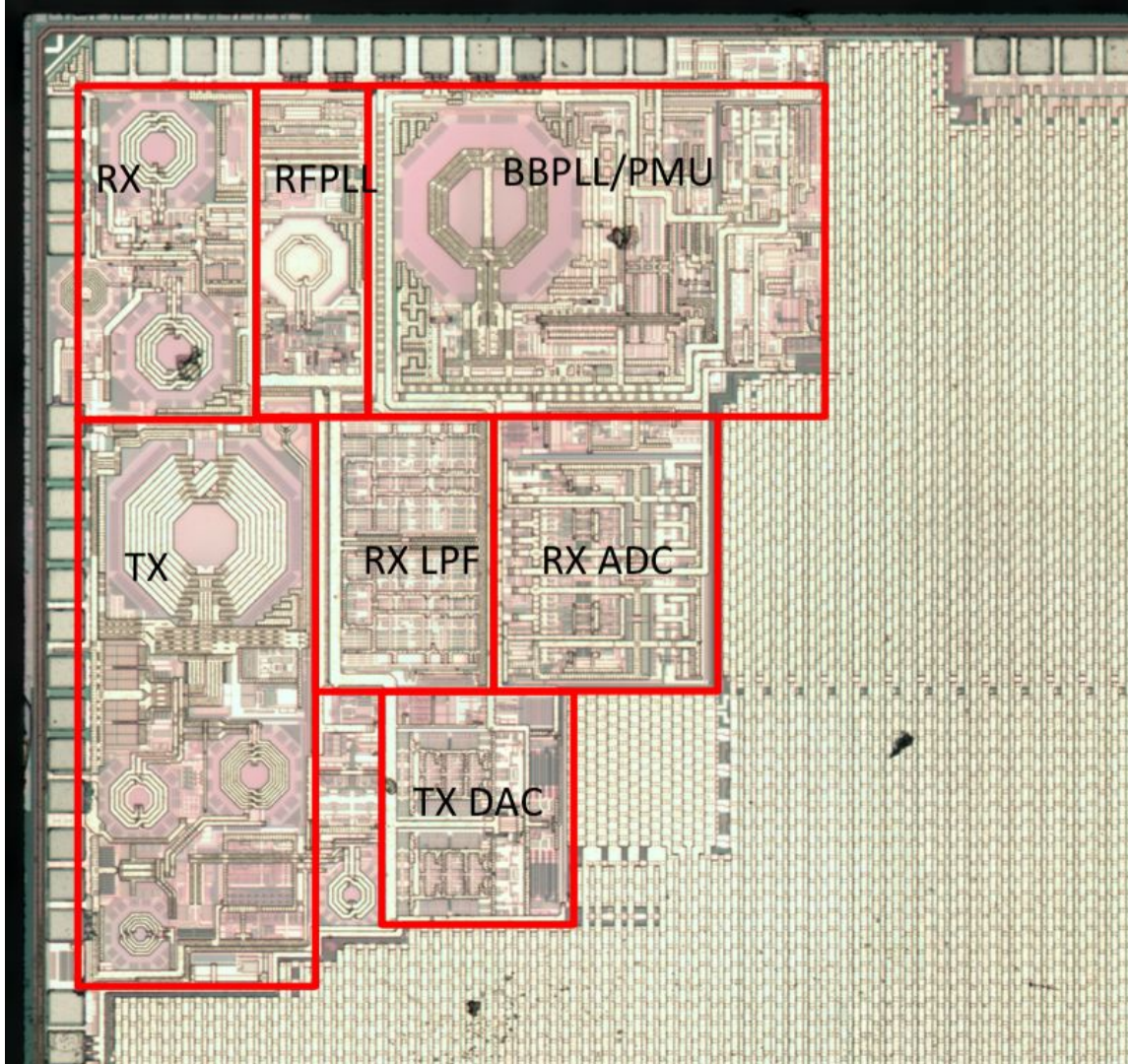
https://zeptobars.com/en/read/Espressif-ESP8266-wifi-serial-rs232-ESP8089-IoT



RX - Receptor
RF PLL - Oscilador da portadora (2.4GHz)
BB PLL - Oscilador de clock (80MHz)
PMU - Unidade de potência
TX - Transmissor
RX LPF - Filtro de baixa potência do RX
RX ADC - Conversor AD do RX
TX ADC - Conversor AD do TX

PLL - Phase-locked loop
BB - Baseband
PMU - Power Management Unit

Sobre RFPLL e BBPLL (pág.34):
https://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf



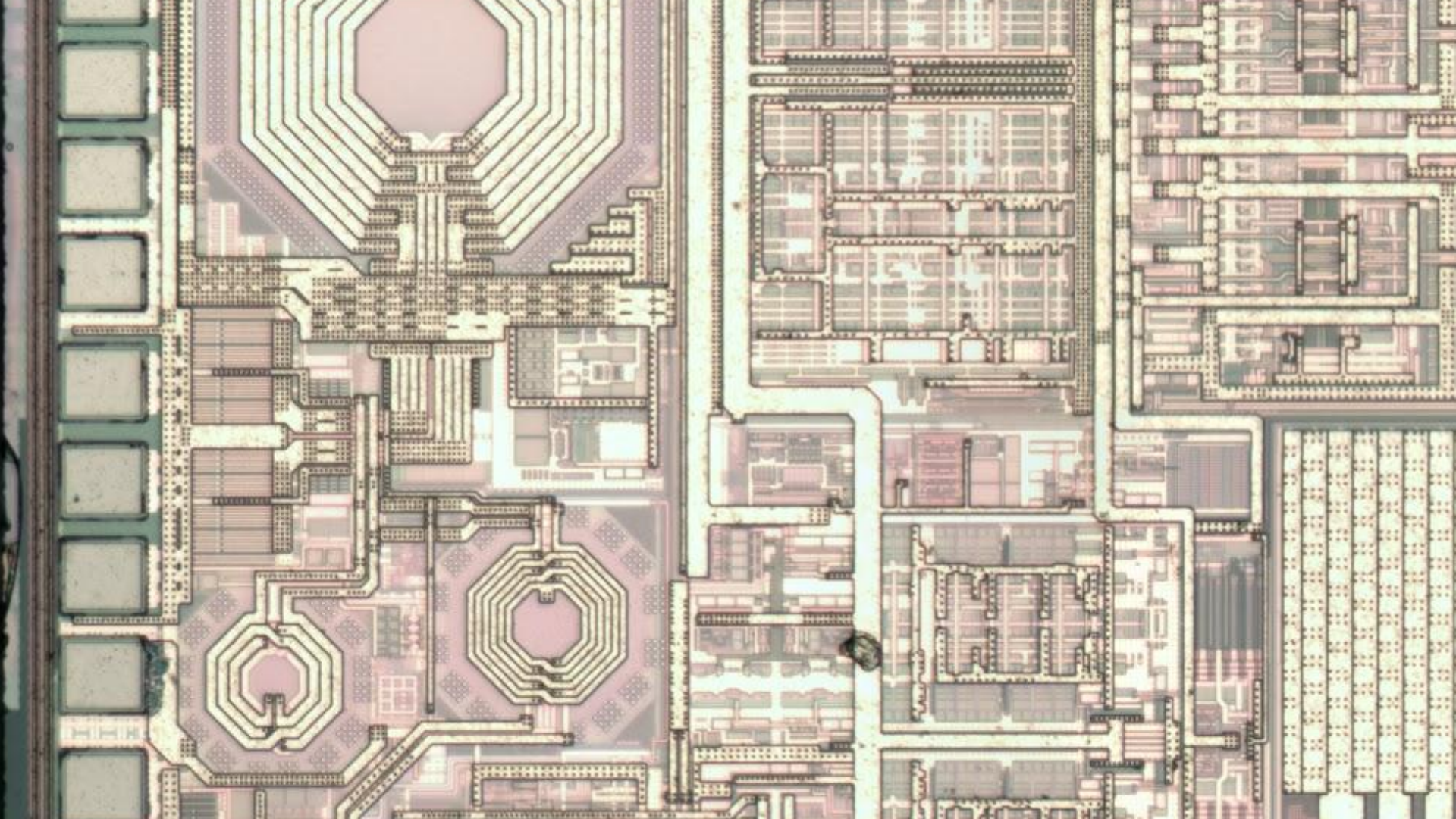


Figure 2-1 shows the pin layout for 32-pin QFN package.

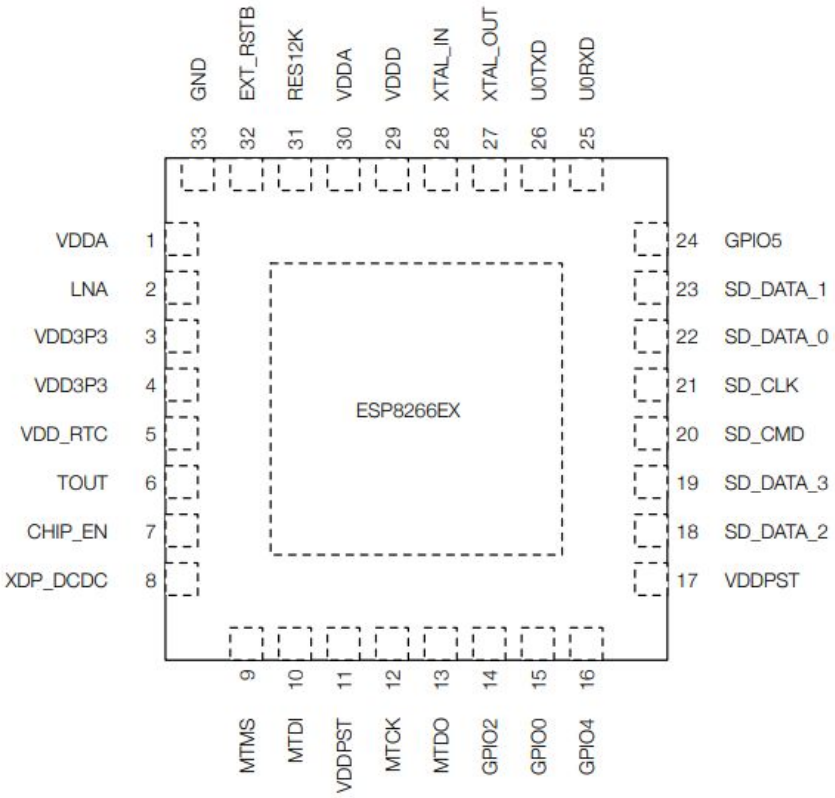
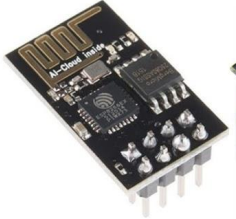


Figure 2-1. Pin Layout (Top View)

https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf



ESP-01

ESP-01S

ESP-01M

ESP-02

ESP-03

ESP-04

ESP-05

ESP-6

ESP-07

ESP-07S

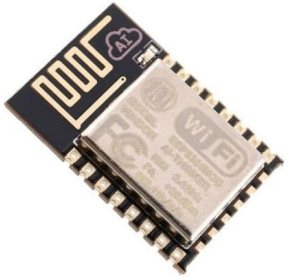
ESP-08

ESP-09

ESP-10

ESP-11

ESP-12

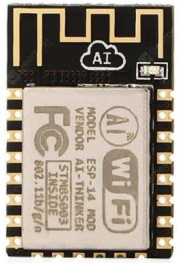


ESP-12E

ESP-12F

ESP-12S

ESP-13



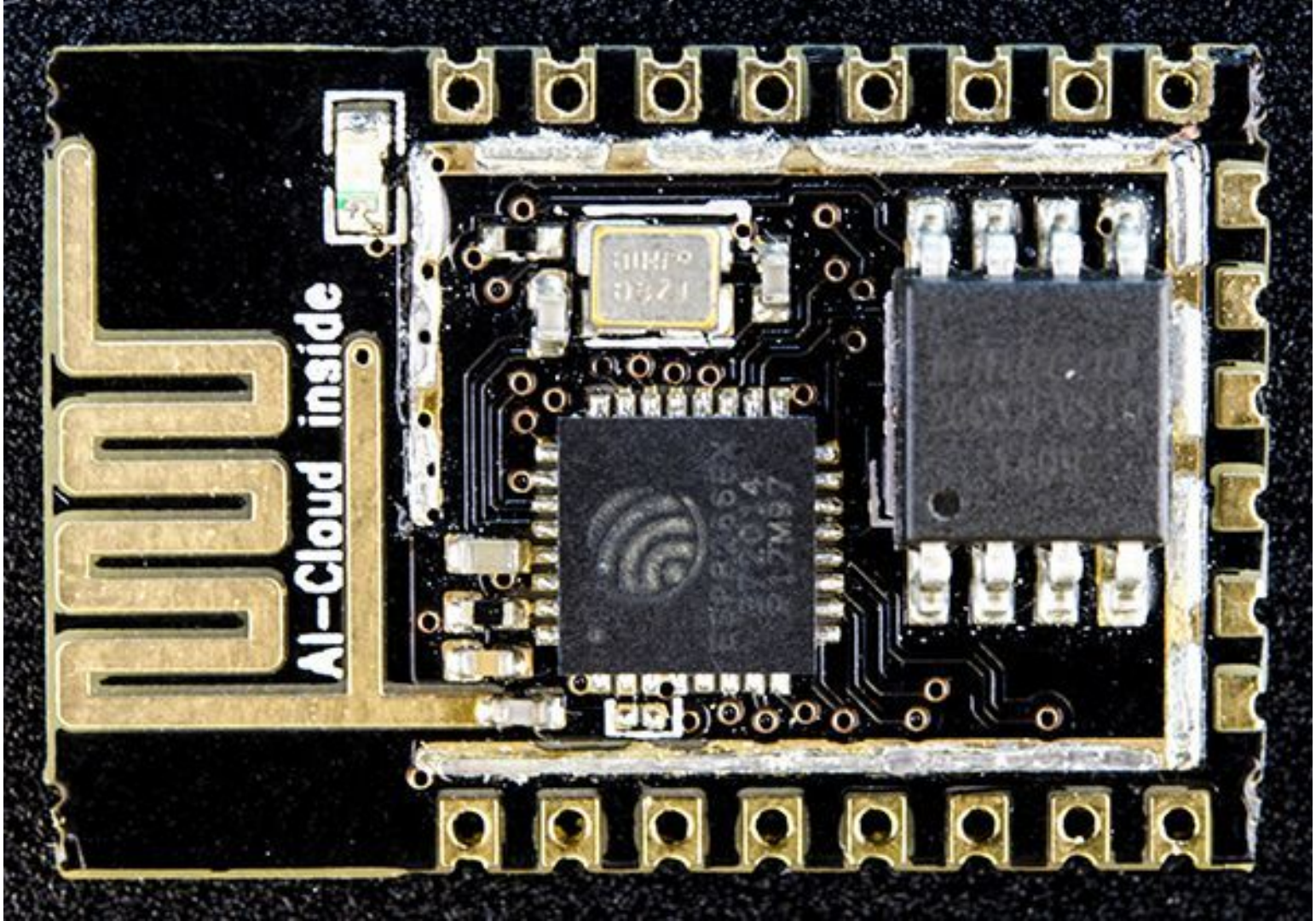
ESP-14

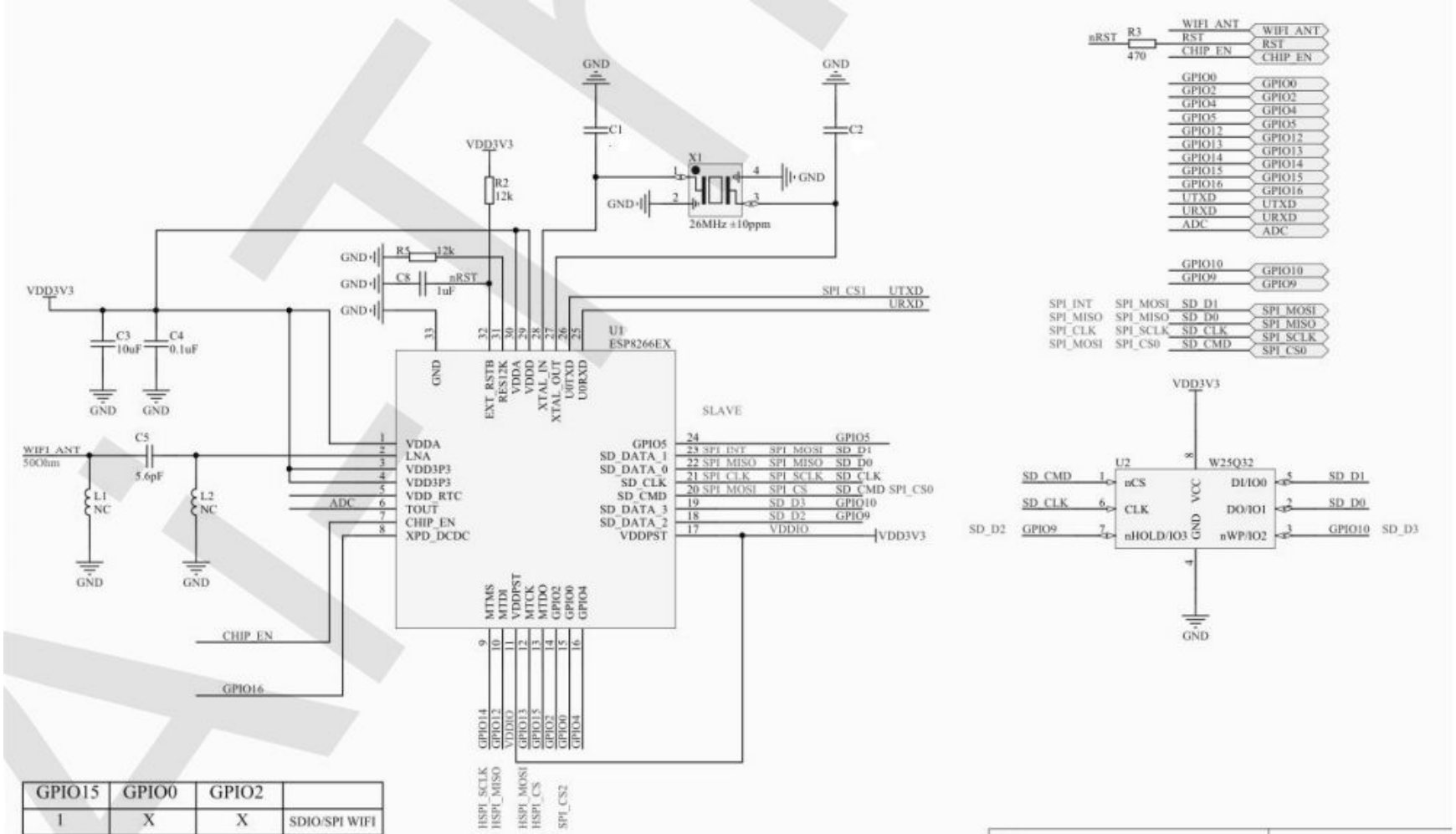
Name	Active pins	Pitch	Form factor	LEDs	Antenna	Shielded	Dimensions (mm)	Notes
ESP-01	6	0.1 in	2×4 DIL	Yes	PCB trace	No	14.3 × 24.8	512 KiB Flash and blue PCB from a generic manufacturer. 1 MiB Flash, AI-Cloud and black PCB from AI-Thinker.
ESP-01S	6	0.1 in	2×4 DIL	Yes	PCB trace	No	14.4 × 24.7	1 MiB Flash
ESP-01M	16	1.6 mm	2×9 edge connector	No	PCB trace	Yes	18.0 × 18.0	Uses ESP8285 (1 MiB built-in flash).
ESP-02	6	0.1 in	2×4 castellated	No	U.FL socket	No	14.2 × 14.2	
ESP-03	10	2 mm	2×7 castellated	No	Ceramic	No	17.3 × 12.1	
ESP-04	10	2 mm	2×4 castellated	No	None	No	14.7 × 12.1	
ESP-05	3	0.1 in	1×5 SIL	No	U.FL socket	No	14.2 × 14.2	

ESP-06	11	various	4×3 dice	No	None	Yes	14.2 × 14.7	Not FCC approved.
ESP-07	14	2 mm	2×8 pinhole	Yes	Ceramic + U.FL socket	Yes	20.0 × 16.0	Not FCC approved.
ESP-07S	14	2 mm	2×8 pinhole	No	U.FL socket	Yes	17.0 × 16.0	FCC and CE approved.
ESP-08	10	2 mm	2×7 castellated	No	None	Yes	17.0 × 16.0	Not FCC approved.
ESP-09	10	various	4×3 dice	No	None	No	10.0 × 10.0	
ESP-10	3	2 mm	1×5 castellated	No	None	No	14.2 × 10.0	
ESP-11	6	1.27 mm	1×8 pinhole	No	Ceramic	No	17.3 × 12.1	
ESP-12	14	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	FCC and CE approved.[19]
ESP-12E	20	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	4 MiB flash.

ESP-12F	20	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	FCC and CE approved. Improved antenna performance.
ESP-12S	14	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	FCC approved.[20]
ESP-13	16	1.5 mm	2×9 castellated	No	PCB trace	Yes	W18.0 × L20.0	Marked as "FCC". Shielded module is placed sideways, as compared to the ESP-12 modules.
ESP-14	22	2 mm	2×8 castellated +6	No	PCB trace	Yes	24.3 × 16.2	

https://en.wikipedia.org/wiki/ESP8266


AI-Cloud inside



GPIO15	GPIO0	GPIO2	
1	X	X	SDIO/SPI WIFI
0	0	1	Uart Download
0	1	1	Flash Boot

https://wiki.ai-thinker.com/_media/esp8266/a014ps01.pdf

Title: ESP-12F WIFI Module

Size: A4

Revision:2.0

Drawn By:

AI Thinker

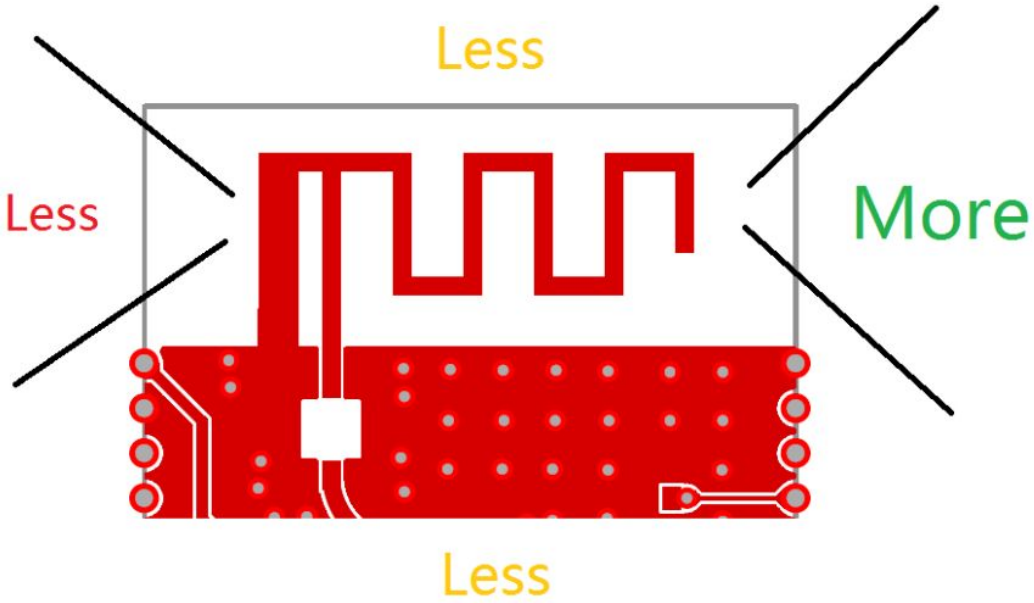
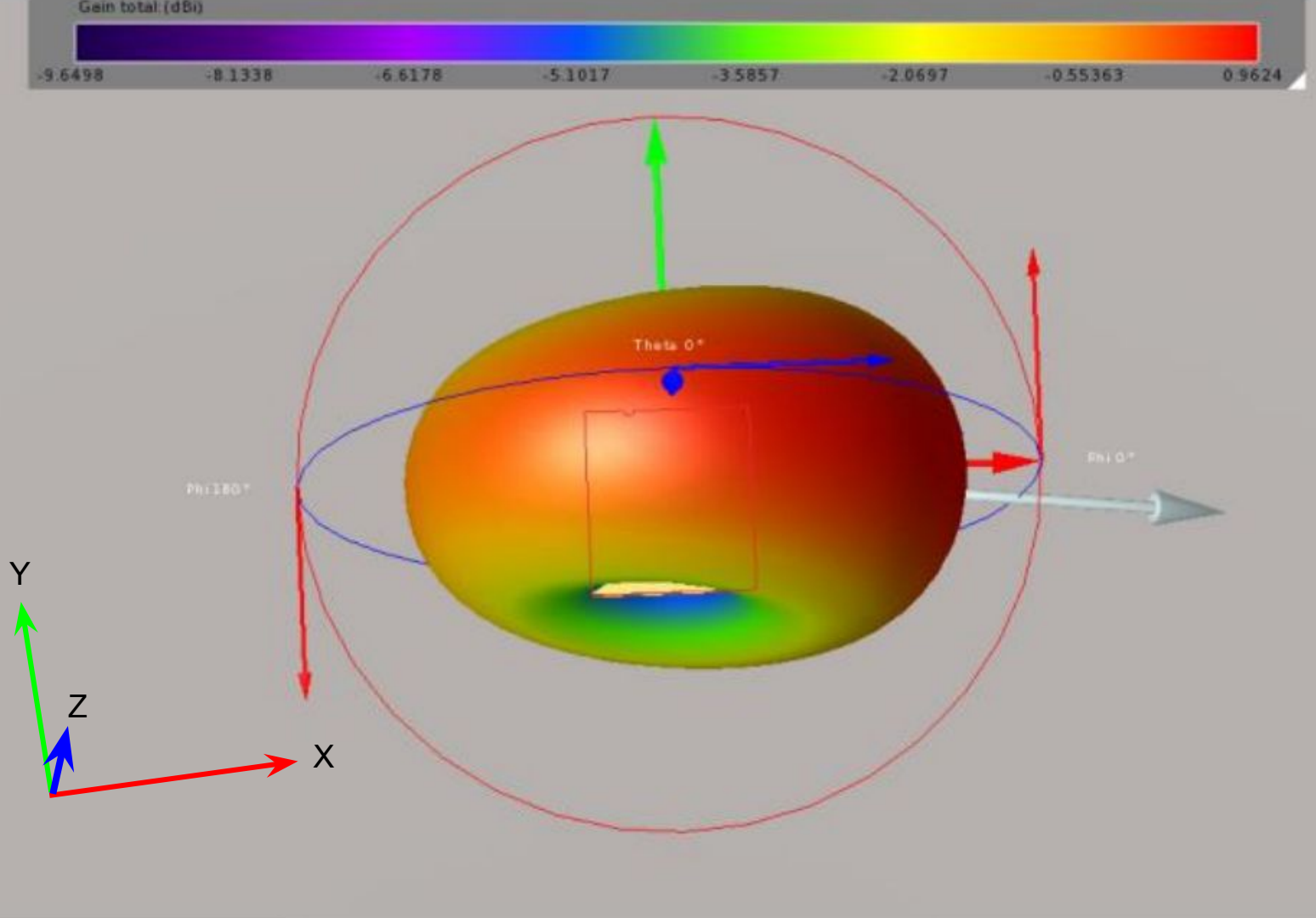


Fig 1. MIFA antenna radiation direction

Meandered Inverted F Antenna (MIFA)

https://www.nxp.com/docs/en/nxp/application-notes/AN11994.pdf

https://www.cypress.com/file/136236/download


Gain total (dBi)
-9.6498
-8.1338
-6.6178
-5.1017
-3.5857
-2.0697
-0.55363
0.9624
Theta 0°
Phi 0°
Phi 180°
Y
Z
X

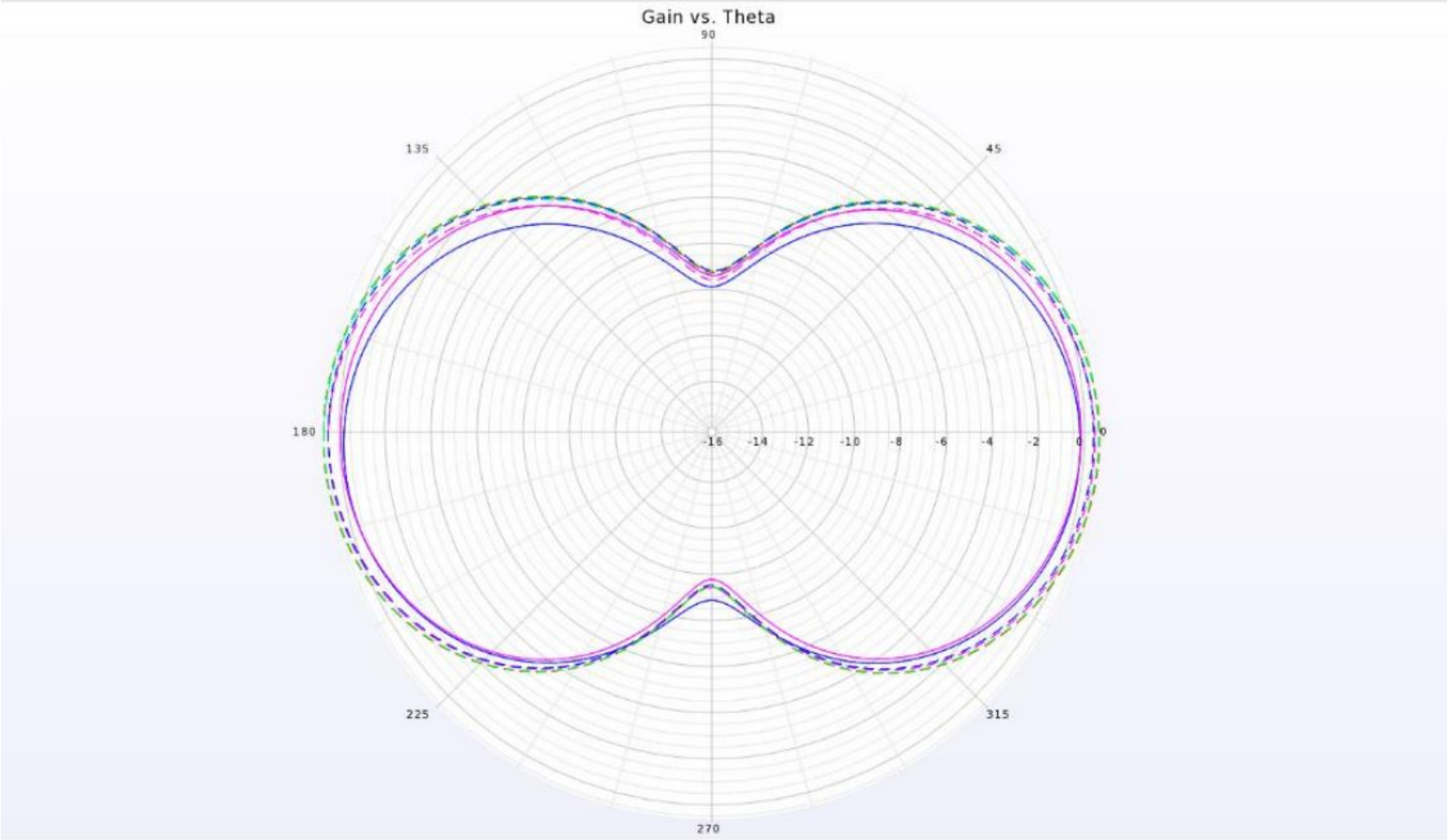


Fig 12. Antenna gain pattern @ phi = 90°

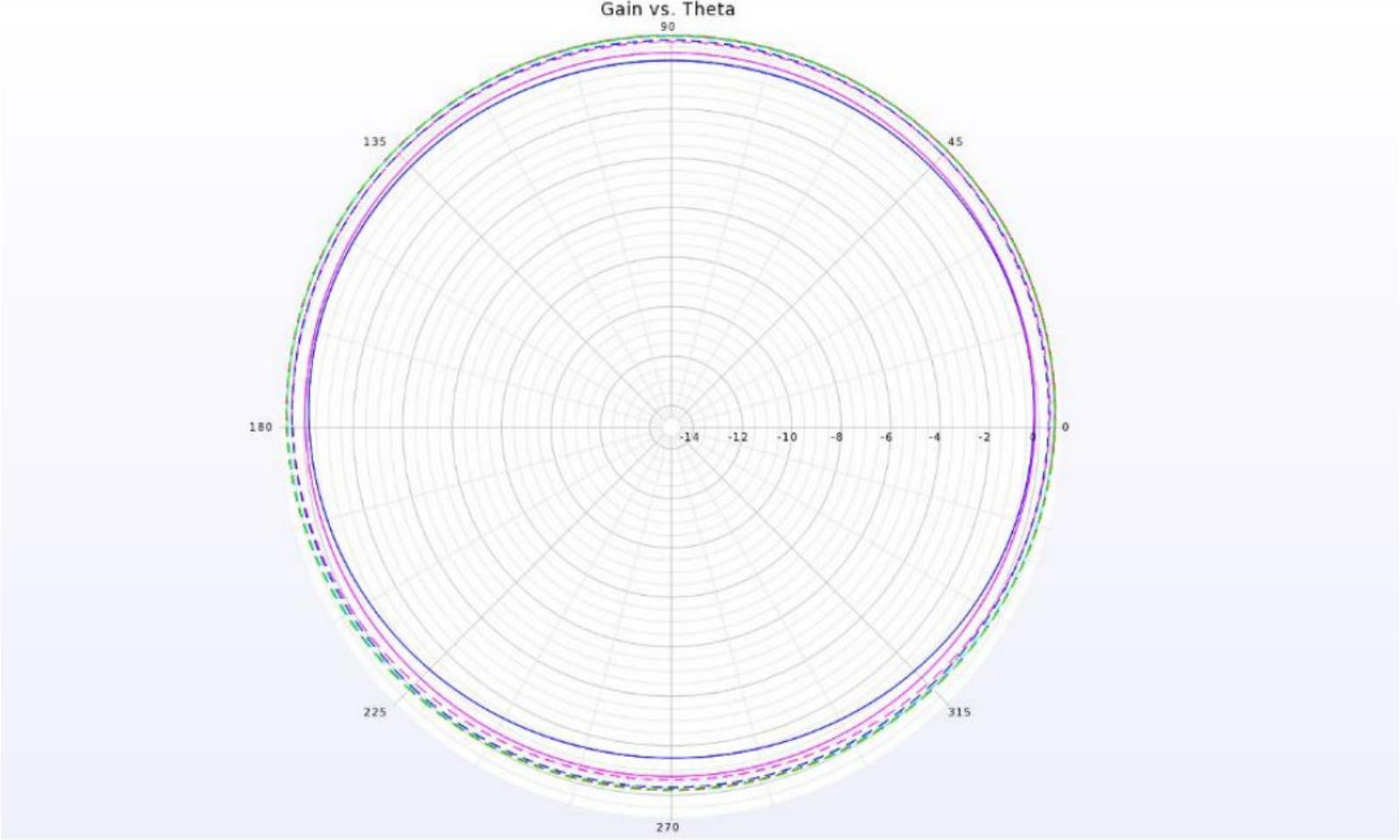
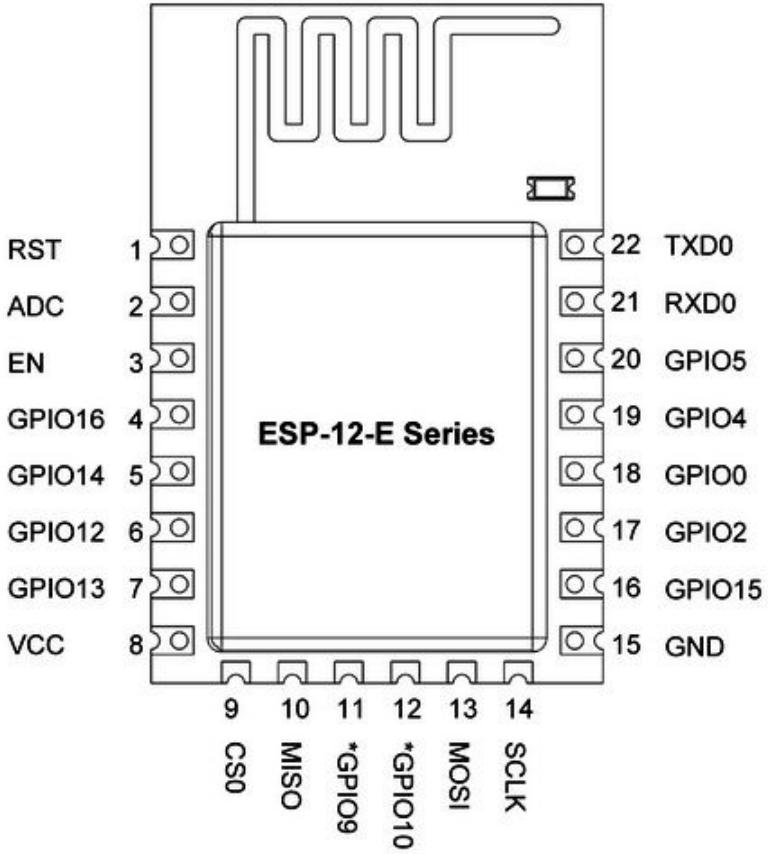
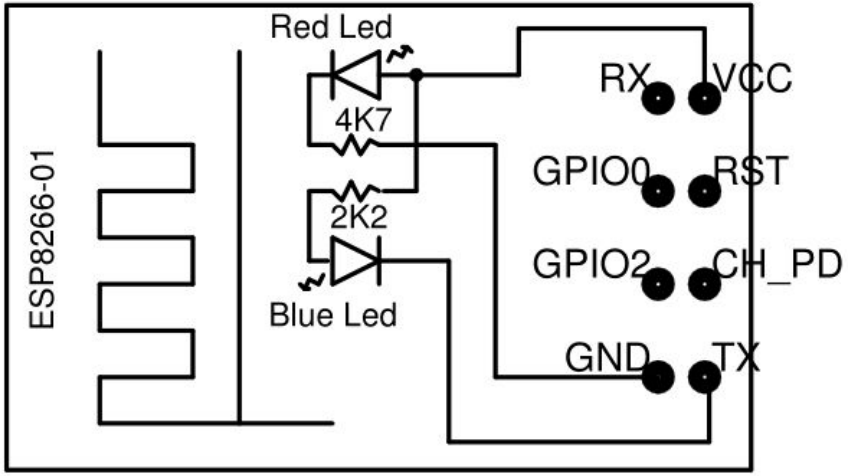


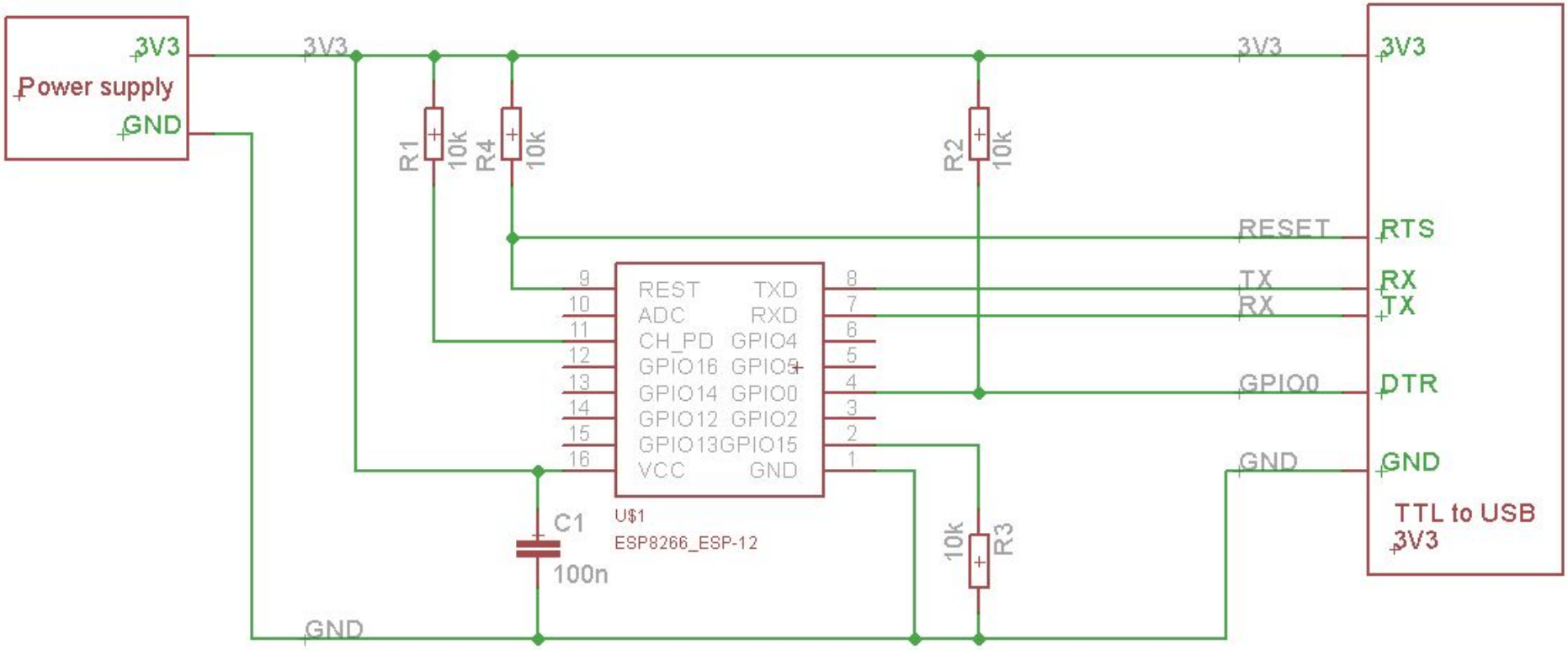
Fig 13. Antenna gain pattern @ phi = 0°

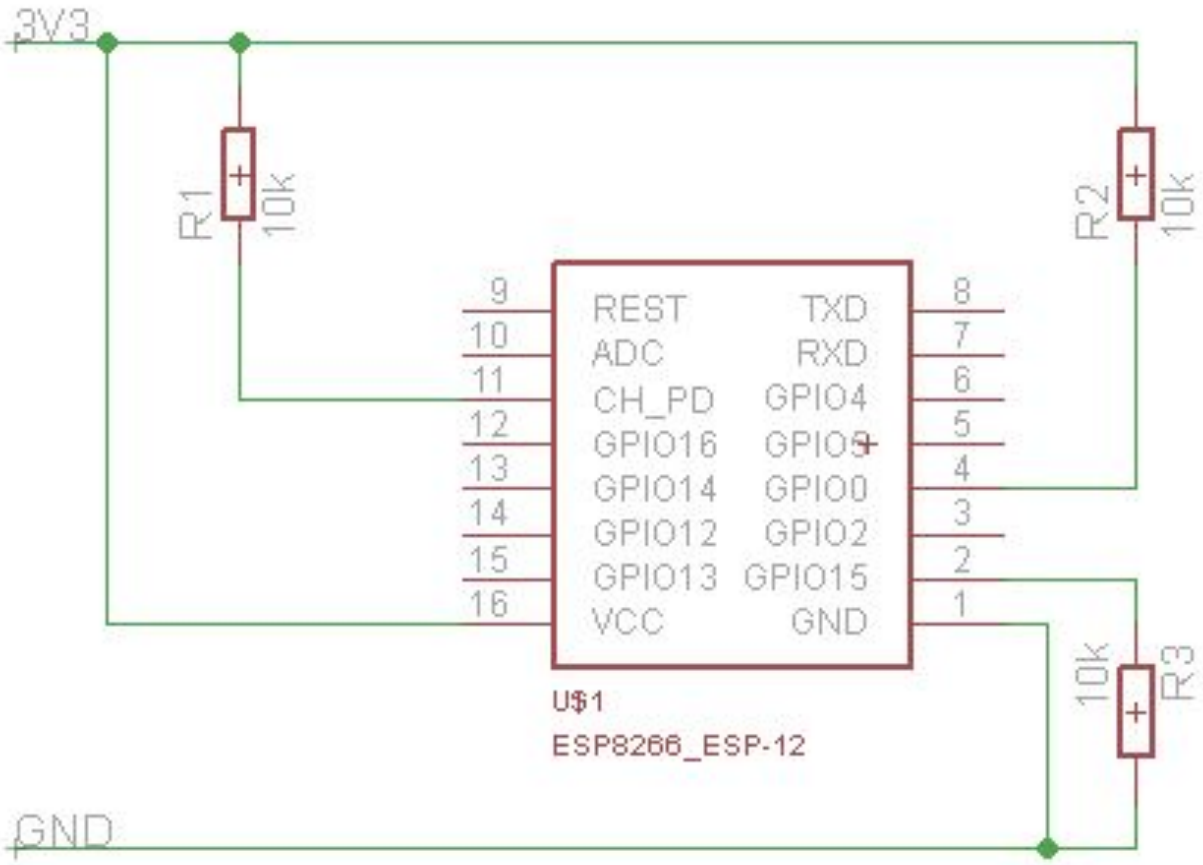

ESP-12-E Series
RST 1
ADC 2
EN 3
GPIO16 4
GPIO14 5
GPIO12 6
GPIO13 7
VCC 8
22 TXD0
21 RXD0
20 GPIO5
19 GPIO4
18 GPIO0
17 GPIO2
16 GPIO15
15 GND
9 CS0
10 MISO
11 *GPIO9
12 *GPIO10
13 MOSI
14 SCLK

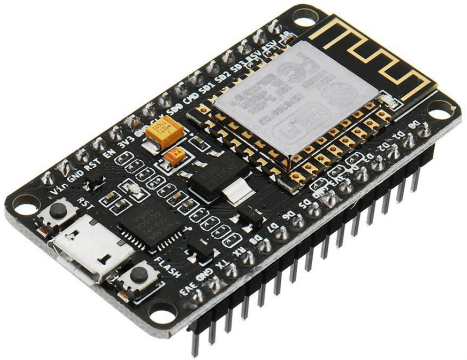

ESP8266-01
Red Led
4K7
2K2
Blue Led
RX
VCC
GPIO0
RST
GPIO2
CH_PD
GND
TX

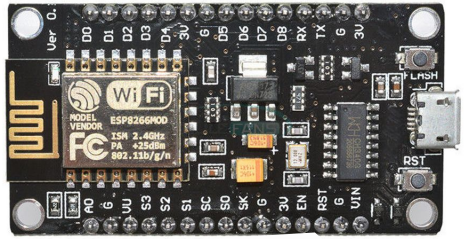
ESP8266 Boot Options

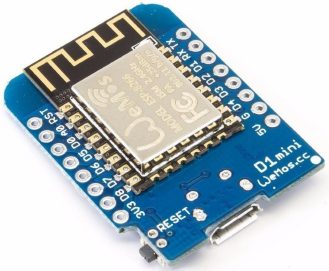
GPIO15 (MTDO)	GPIO0	GPIO2	Mode	Comments
L	H	H	Flash	Boot from SPI Flash (Normal running)
L	L	H	UART	Program via UART (TX/RX)
H	x (not care)	x (not care)	SDIO	Boot from SD-card

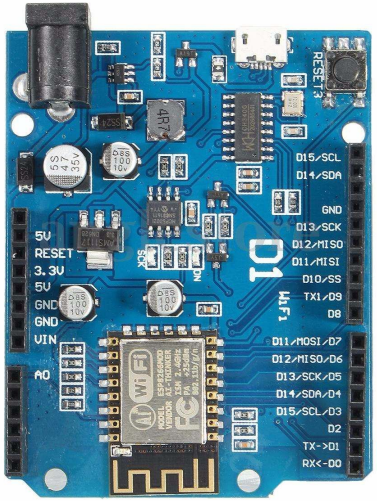

3V3
Power supply
GND
3V3
R1
10k
R4
10k
R2
10k
3V3
3V3
RESET
RTS
TX
RX
RX
TX
GPIO0
DTR
GND
GND
TTL to USB
3V3
9
10
11
12
13
14
15
16
REST
ADC
CH_PD
GPIO16
GPIO14
GPIO12
GPIO13
VCC
TXD
RXD
GPIO4
GPIO5
GPIO0
GPIO2
GPIO15
GND
8
7
6
5
4
3
2
1
U$1
ESP8266_ESP-12
C1
100n
10k
R3
GND


3V3
R1
10k
R2
10k
9
10
11
12
13
14
15
16
REST
ADC
CH_PD
GPIO16
GPIO14
GPIO12
GPIO13
VCC
TXD
RXD
GPIO4
GPIO5
GPIO0
GPIO2
GPIO15
GND
8
7
6
5
4
3
2
1
U$1
ESP8266_ESP-12
10k
R3
GND

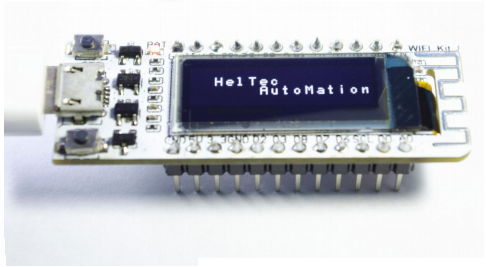


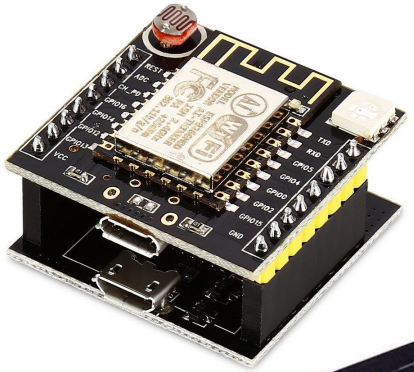

Wi Fi
MODEL ESP8266MOD
VENDOR
ISM 2.4GHz
PA +25dBm
802.11b/g/n
FLASH
RST

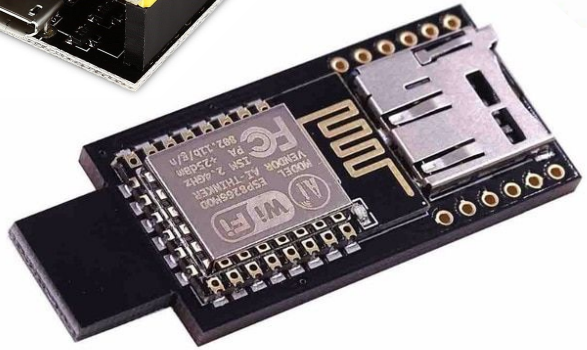

D1mini
Lenbas.cc
RESET

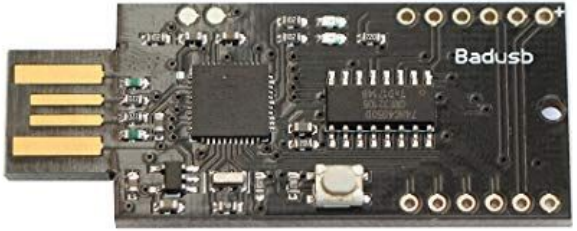

RESET
D15/SCL
D14/SDA
GND
D13/SCK
D12/MISO
D11/MISI
D10/SS
TX1/D9
D8
D11/MOSI/D7
D12/MISO/D6
D13/SCK/D5
D14/SDA/D4
D15/SCL/D3
D2
TX->D1
RX<-D0
5V
RESET
3.3V
5V
GND
GND
VIN
A0
D1
Wi Fi

SUN HOLD
RAS-0515
ESP8266-EVB


HelTec
AutoMation






Badusb

Name	Active pins	Pitch	Form factor	LEDs	Antenna	Shielded	Dimensions (mm)	Notes
Bolt IoT	14	0.1 in	2×14 DIL	Yes	PCB trace	Yes	30 × 40	Comes with an onboard SD card and features like Lib-Discovery and Fail Safe Mode. Has its own cloud for IoT.
Olimex MOD-WIFI-ESP8266[21]	2	0.1 in	UEXT module	Yes	PCB trace	No	?	Only RX/TX are connected to UEXT connector.
Olimex MOD-WIFI-ESP8266-DEV[22]	20	0.1 in	2×11 DIL + castellated	Yes	PCB trace	No	33 × 23	All available GPIO pins are connected, also has pads for soldering UEXT connector (with RX/TX and SDA/SCL signals).
NodeMCU DEVKIT	14	0.1 in	2×15 DIL	Yes	PCB trace	Yes	49 × 24.5	Uses the ESP-12 module; includes USB to serial interface.
Adafruit Huzzah ESP8266 breakout[23]	14	0.1 in	2×10 DIL	Yes	PCB trace	Yes	25 × 38	Uses the ESP-12 module.
SparkFun ESP8266 Thing[24] WRL-13231	12	0.1 in	2×10 DIL	Yes	PCB trace + U.FL socket	No	58 × 26	FTDI serial header, Micro-USB socket for power, includes Li-ion battery charger.

KNEWRON Technologies smartWIFI[25]	12	0.1 in	2×20 DIL	Yes 1 RGB	PCB trace	Yes	25.4 × 50.8	CP2102 USB bridge, includes battery charger, micro-USB socket for power and battery charging, 1 RGB LED and USER / Reflash button.
ArduCAM ESP8266 UNO[26]	12+	0.1 in	Arduino Uno	Yes	PCB trace	Yes	53.4 × 68.6	Uses the AI Thinker's ESP8266MOD module and features Micro-USB port, Battery pins, Camera pins and uSD card all on the same board. Fully compatible with Arduino Uno shields.
DoIT ESPduino[27]	12	0.1 in	Arduino Uno	Yes	PCB trace	Yes	53.4 × 68.6	Uses the ESP-WROOM-02 (ESP-13) module and Mini-USB port. Fully compatible with Arduino Uno shields.
WeatherPlus - SwitchDoc Labs[28]	26+Grove	0.1 in	Custom	Yes	PCB trace	Yes	86.0 × 50.0	Uses the AI Thinker Model ESP8266MOD (ESP-13) module and FTDI for Programming and Mini-USB port for power. Fully compatible with Adafruit Huzzah software. Includes BMP280 Barometer, ADS1115 and Grove I2C connectors. Plugs for Anemometer/Wind Vane/Rain Bucket.
WeMos[29] D1[30]	12	0.1 in	Arduino Uno	Yes	PCB trace	Yes	53.4 × 68.6	Uses the ESP-12F module and Micro-USB socket. Discontinued in favor of WeMos D1 R2.
WeMos[29] D1 R2[31]	12	0.1 in	Arduino Uno	Yes	PCB trace	Yes	53.4 × 68.6	Uses ESP-12F module and has Micro-USB socket.

WeMos[29] D1 mini[32]	12	0.1 in	2×8 DIL	Yes	PCB trace	Yes	25.6 × 34.2	Uses ESP-12S module and has Micro-USB socket.
WeMos[29] D1 mini Lite[33]	12	0.1 in	2×8 DIL	Yes	PCB trace	Yes	25.6 × 34.2	Based on the ESP8285, an ESP8266 with 1 MiB flash built-in; has Micro-USB socket.
WeMos[29] D1 mini Pro[34]	12	0.1 in	2×8 DIL	Yes	Ceramic and U.FL socket	Yes	25.6 × 34.2	Uses ESP8266EX chip; has Micro-USB socket, U.FL antenna connector, and 16 MiB flash.
ESPert ESPresso Lite[35]	16	0.1 in	2×8 DIL	Yes	PCB trace	Yes	26.5 × 57.6	Uses the ESP-WROOM-02 module. Produced in limited quantity as beta version.
ESPert ESPresso Lite V2.0[36]	24	0.1 in	2×10 DIL	Yes	PCB trace	Yes	28 × 61	Improved version of ESPresso Lite.
In-Circuit ESP-ADC[37]	18	0.1 in	2×9 DIL	No	U.FL socket	Yes	22.9 × 14.9	Uses ESP8266EX chip.

In-Circuit ESP-ADC[37]	18	0.1 in	2×9 DIL	No	U.FL socket	Yes	22.9 × 14.9	Uses ESP8266EX chip.
Watterott ESP-WROOM02-Breakout[38]	14	0.1 in	2×10 DIL	Yes	PCB trace	Yes	40.64 × 27.94	Uses the Espressif ESP-WROOM-02 module.
Geek Wave Solution IOT WROOM-02 Dev. Board[39]	20	0.1 in	?	Yes	PCB trace	Yes	93.80 × 80.02	Development board with Espressif ESP-WROOM-02 module and four relays.
Witty[40]	2 x 8	dunno	?	Yes	PCB trace	Yes	Dunno	Development board with Espressif ESP8266MOD with separable USB interface; voltage regulator on back, AMS1117. Single tri-color LED, RGB on pins GP15, GP13, GP12, respectively. LDR on-board connected to Vcc 3.3V **directly** to SoC and the other pin to A0. Vcc on the **development board** requires 5V.

https://en.wikipedia.org/wiki/ESP8266

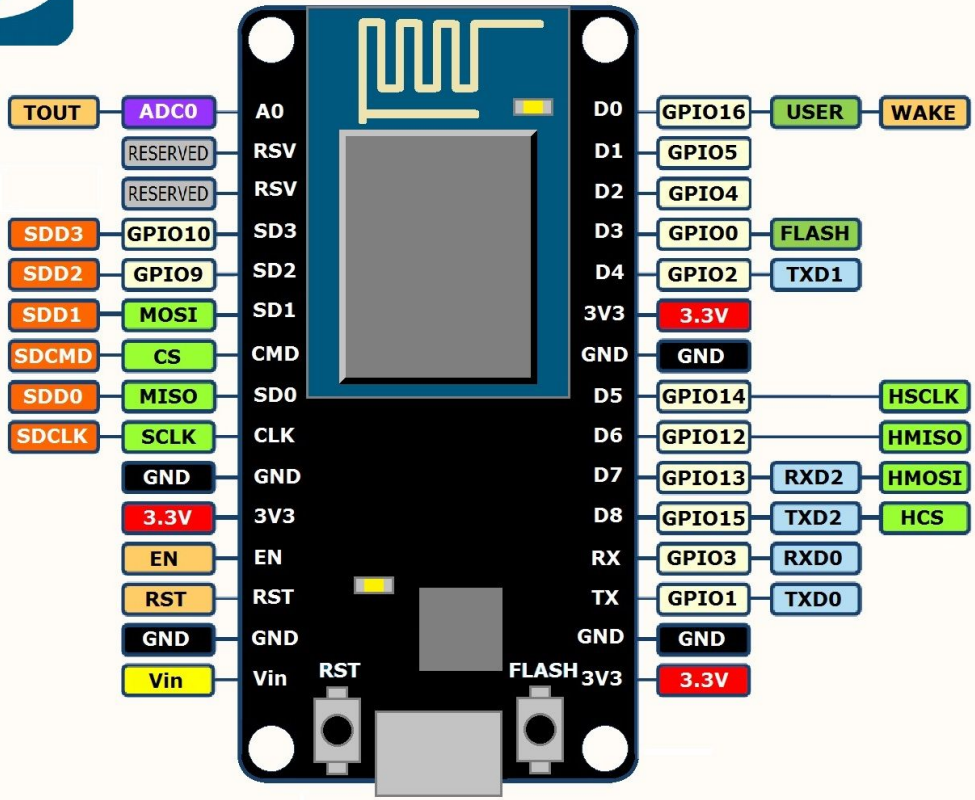
NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core,[citation needed] widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub.Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9.

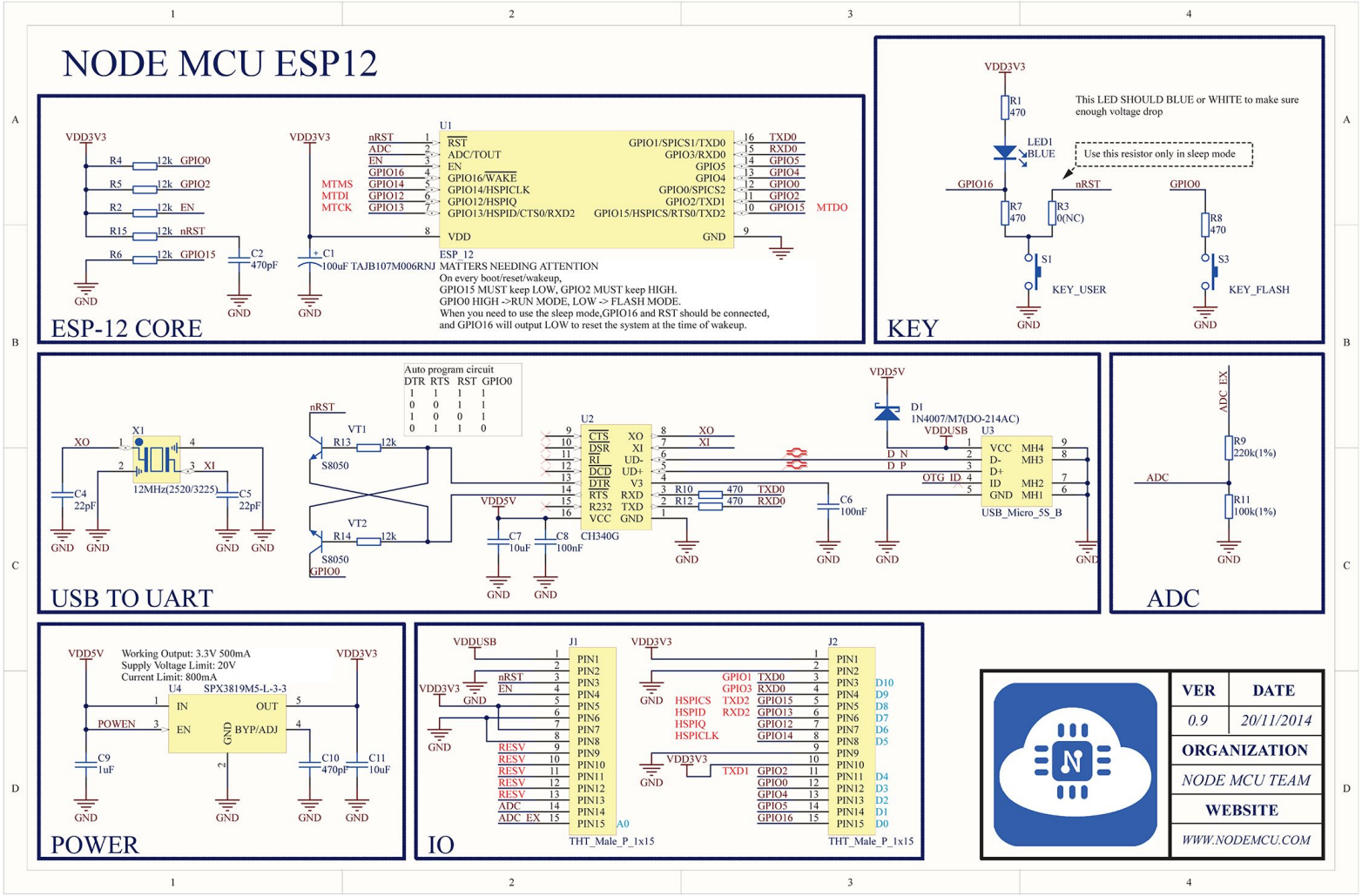
NodeMCU ESP-12 development kit V1.0

PIN DEFINITION



Arduining.com

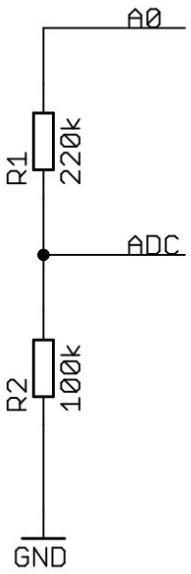
NODE MCU ESP12



https://github.com/nodemcu/nodemcu-devkit https://www.maxlinear.com/ds/spx3819.pdf

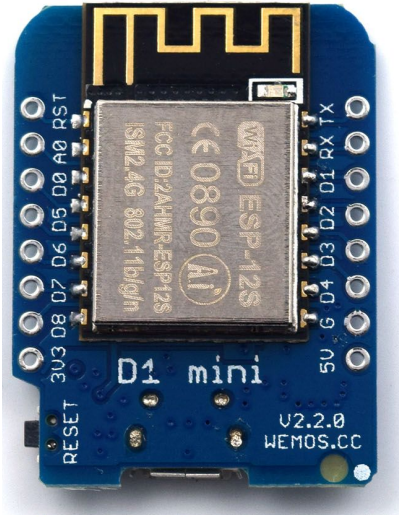
Wemos D1 mini

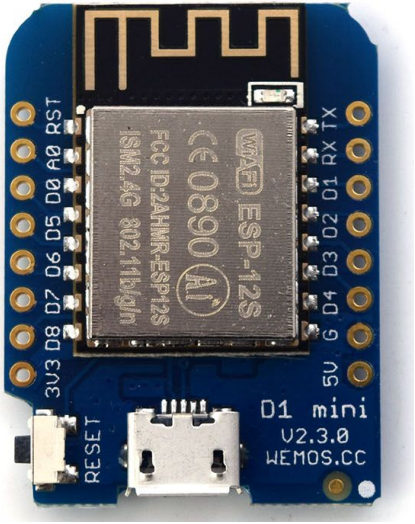
- 1 digital input/output pins, all pins have interrupt/pwm/I2C/one-wire supported(except D0)
- 1 analog input(3.2V max input)
- a Micro USB connection
- Compatible with MicroPython, Arduino, nodemcu

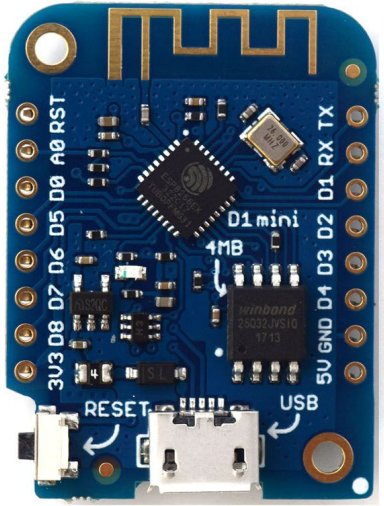


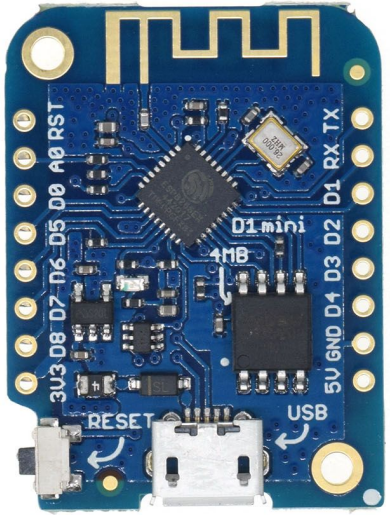
https://wiki.wemos.cc/products:d1:d1_mini

Esquemático:
https://wiki.wemos.cc/_media/products:d1:sch_d1_mini_v3.0.0.pdf

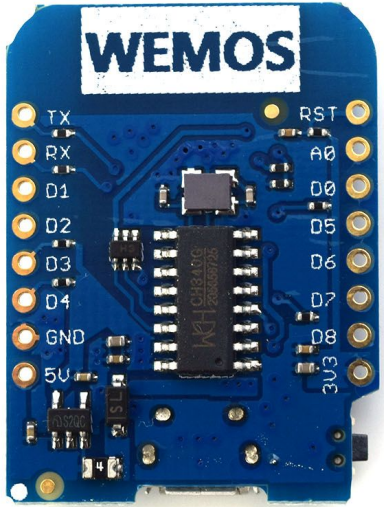

RST A0 D0 D5 D6 D7 D8 3V3
TX RX D1 D2 D3 D4 G 5V
ESP-12S
D1 mini
RESET
V2.2.0
WEMOS.CC

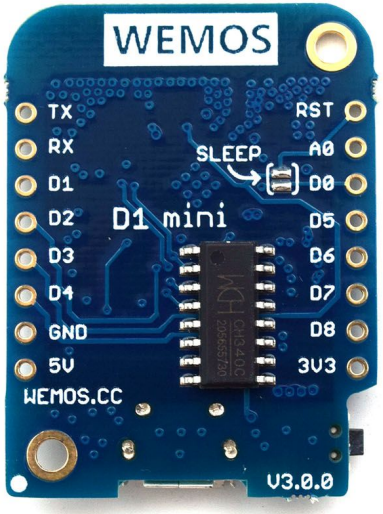

RST A0 D0 D5 D6 D7 D8 3V3
TX RX D1 D2 D3 D4 G 5V
ESP-12S
D1 mini
RESET
V2.3.0
WEMOS.CC

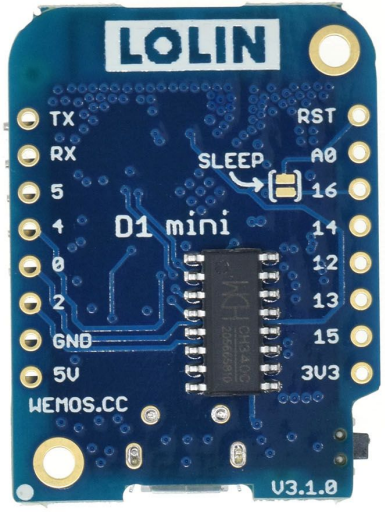

RST A0 D0 D5 D6 D7 D8 3V3
TX RX D1 D2 D3 D4 GND 5V
D1 mini
4MB
RESET
USB

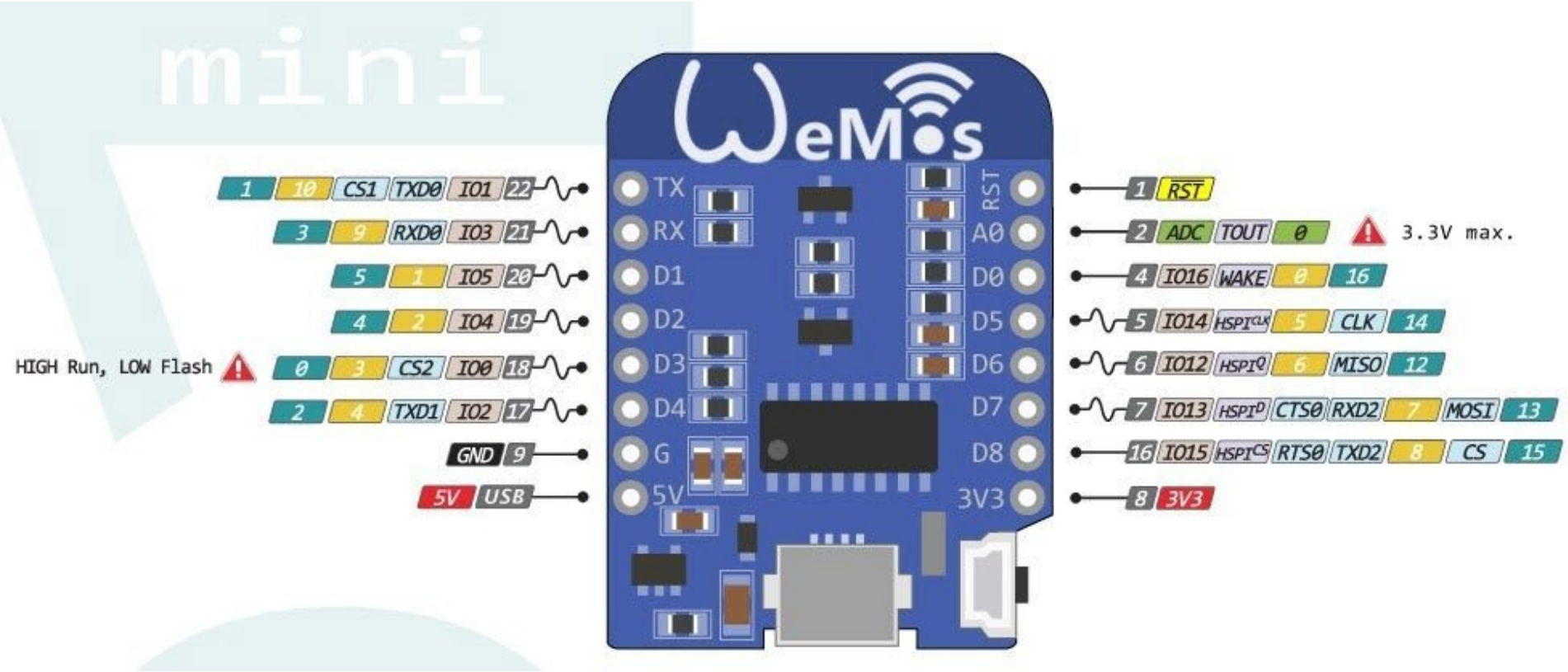

RST A0 D0 D5 D6 D7 D8 3V3
TX RX D1 D2 D3 D4 GND 5V
D1 mini
4MB
RESET
USB

WEMOS
TX RX D1 D2 D3 D4 GND 5V
RST A0 D0 D5 D6 D7 D8 3V3

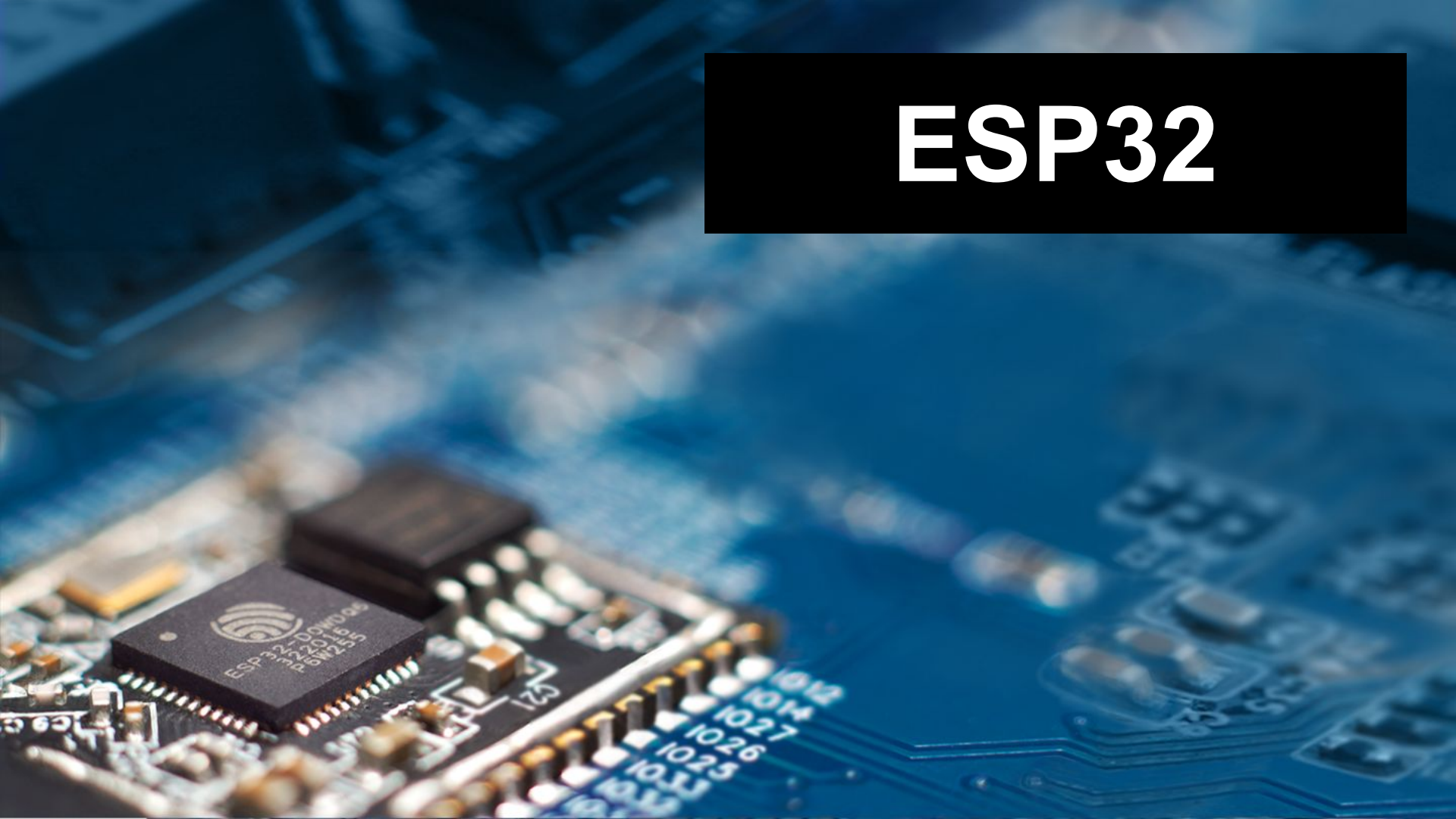

WEMOS
TX RX D1 D2 D3 D4 GND 5V
RST A0 D0 D5 D6 D7 D8 3V3

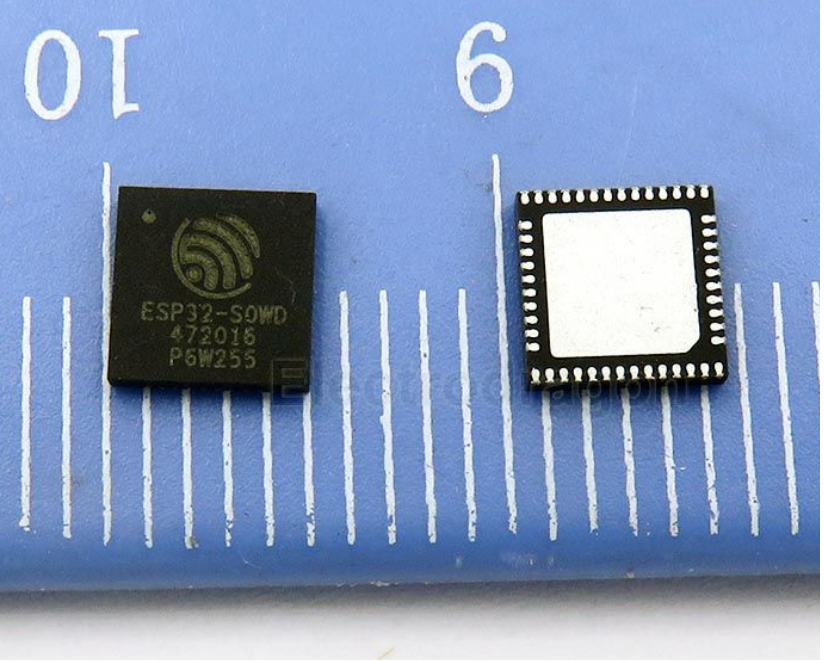

WEMOS
TX RX D1 D2 D3 D4 GND 5V
SLEEP
D1 mini
RST A0 D0 D5 D6 D7 D8 3V3
WEMOS.CC
V3.0.0


LOLIN
TX RX 5 4 0 2 GND 5V
SLEEP
D1 mini
RST A0 16 14 12 13 15 3V3
WEMOS.CC
V3.1.0


mini
WeMos
TX
RX
D1
D2
D3
D4
G
5V
RST
A0
D0
D5
D6
D7
D8
3V3
1 10 CS1 TXD0 IO1 22
3 9 RXD0 IO3 21
5 1 IO5 20
4 2 IO4 19
HIGH Run, LOW Flash
0 3 CS2 IO0 18
2 4 TXD1 IO2 17
GND 9
5V USB
1 RST
2 ADC TOUT 0
3.3V max.
4 IO16 WAKE 0 16
5 IO14 HSPICLK 5 CLK 14
6 IO12 HSPIQ 6 MISO 12
7 IO13 HSPID CTS0 RXD2 7 MOSI 13
16 IO15 HSPICS RTS0 TXD2 8 CS 15
8 3V3

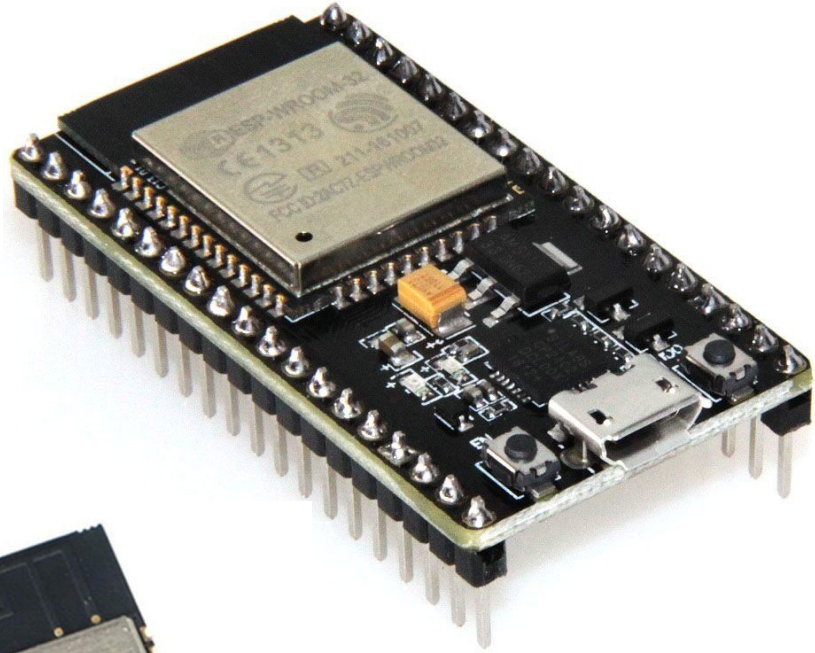
ESP32





QFN48 (6x6 mm)

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf



Circuito Integrado
ESP32

As Características do ESP32 são as seguintes:

- Processadores:
  - CPU: Xtensa dual-core (ou single-core) de 32 bits LX6 microprocessador, operando em 160 ou 240 MHz e realizar até 600 DMIPS
  - Ultra baixa potência (ULP) co-processador
- Memória: 520 KB SRAM
- Conectividade sem fio:
  - Wi-Fi: 802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR e BLE
- Interfaces para periféricos:
  - 12-bit SAR ADC até 18 canais
  - 2 × 8-bit DACs
  - 10 × Sensores de toque (GPIOs de detecção capacitiva)
  - Sensor de temperatura
  - 4 × SPI
  - 2 × Interfaces I²S
  - 2 × Interfaces I²C
  - 3 × UART
  - Controladores de host SD/SDIO/CE-ATA/MMC/eMMC

https://pt.wikipedia.org/wiki/ESP32

    - Controlador escravo SDIO/SPI
    - Interface Ethernet MAC da com DMA dedicado e Suporte Protocolo IEEE 1588 com Tempo Preciso
    - Barramento CAN 2.0
    - Controlador remoto infravermelho (TX/RX, até 8 canais)
    - Motor PWM
    - LED PWM (até 16 canais)
    - Sensor de efeito Hall
    - Pré-amplificador analógico de ultra baixa potência
- Segurança:
    - Padrão IEEE 802.11 recursos de segurança para todas as edições, incluindo WFA, WPA/WPA2 e WAPI
    - Boot seguro
    - Criptografia de Flash
    - 1024-bit OTP, até 768 bits para os clientes
    - Criptografia de hardware de aceleração: AES, SHA-2, RSA, criptografia de curva elíptica (ECC), gerador de número aleatório (RNG)
- Gerenciamento de energia:
    - Regulador de baixa eliminação interno
    - Domínio Individual de energia para o RTC
    - 5uA de consumo no sono profundo.
    - Acordar através de interrupções no GPIO, timer, atraves de medições no ADC ou pelo sensor de interrupção de toque capacitivo.

Início da Produção: 6 de dezembro de 2016

https://pt.wikipedia.org/wiki/ESP32

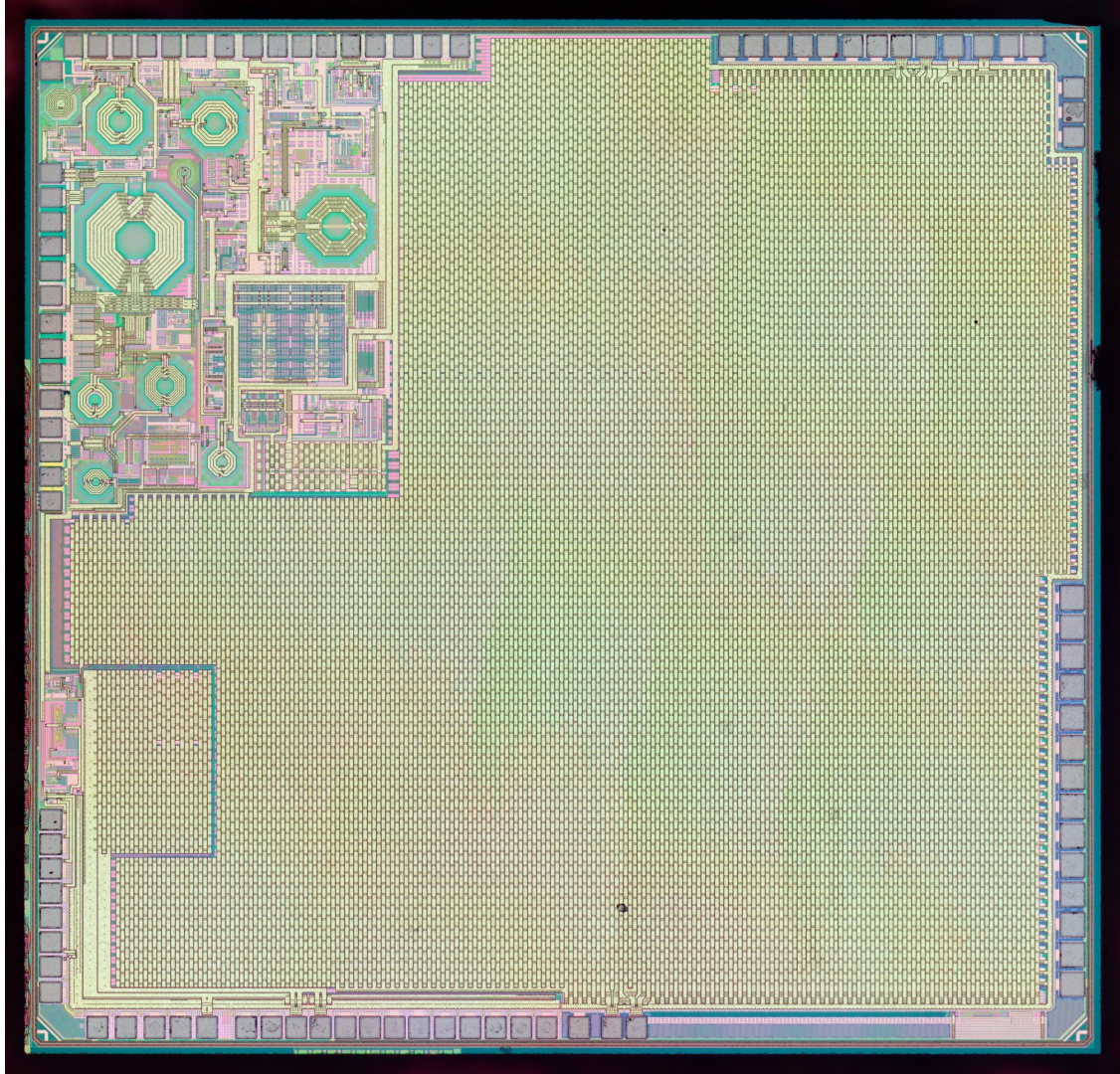
Espressif ESP32 - integrated Wi-Fi transceiver: weekend die-shot

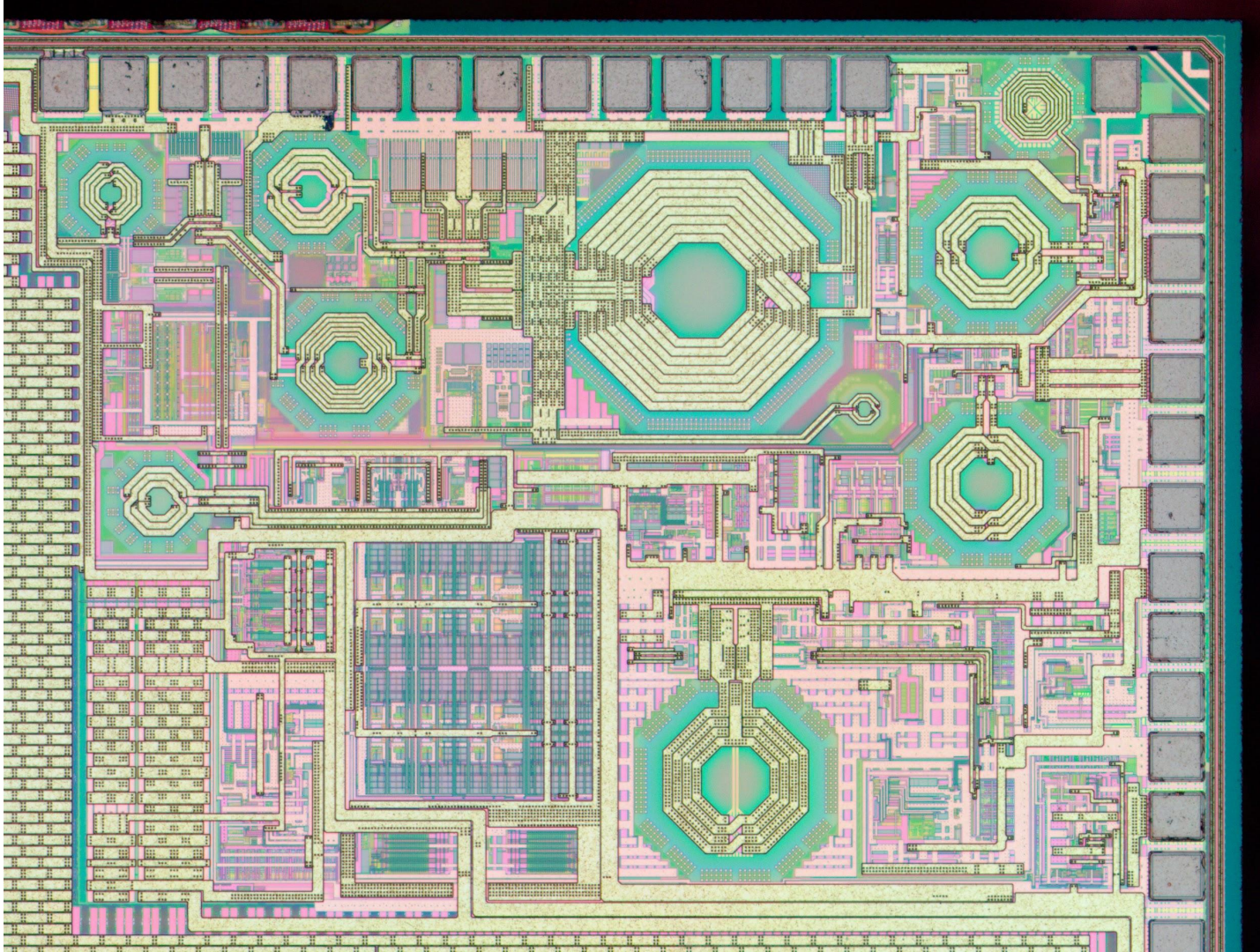
Espressif ESP32 sets the new level for widely available WiFi/Bluetooth transceivers. Now with 2x LX108 CPU core and 520kB RAM. Surprisingly - includes integrated hall sensor. Manufactured on TSMC 40nm ULP technology. Huge upgrade over their previous highly successful ESP8266.

Hopefully next time we'll see their take on 5.8GHz ISM band.

Die size 2960x2850 µm (~double the area of ESP8266).

https://zeptobars.com/en/read/Espressif-ESP32-Wi-Fi-Bluetooth-2.4Ghz-ISM





Programação

SDKs

- **Annex WiFI RDS** – Free rapid development environment . Similar to ESP8266 Basic but much faster and better stability. Self-hosting browser-based development environment. Good peripheral support.
- **Arduino** – A C++-based firmware. With this core, the ESP8266 CPU and its Wi-Fi components can be programmed like any other Arduino device. The ESP8266 Arduino Core is available through GitHub.
- **ESP8266 BASIC** – An open-source basic interpreter specifically tailored for the internet of things. Self-hosting browser-based development environment.
- **ESP Easy** – Developed by home automation enthusiasts.
- **ESP-Open-RTOS** – Open-source FreeRTOS-based ESP8266 software framework.
- **ESP-Open-SDK** – Free and open (as much as possible) integrated SDK for ESP8266/ESP8285 chips.
- **Espruino** – An actively maintained JavaScript SDK and firmware, closely emulating Node.js. Supports a few MCUs, including the ESP8266.
- **MicroPython** – A port of MicroPython (an implementation of Python for embedded devices) to the ESP8266 platform.
- **Mongoose OS** – An open-source operating system for connected products. Supports ESP8266 and ESP32. Develop in C or JavaScript.[13]
- **NodeMCU** – A Lua-based firmware.
- **PlatformIO** (https://platformio.org/platforms/espressif8266) – A cross-platform IDE and unified debugger, which sits on top of Arduino code and libraries.
- **Smick** - Smart brick, based on ESP8266, used to make IoT devices easily using K language.
- **Sming** – An actively developed asynchronous C/C++ framework with superb performance and multiple network features.
- **uLisp** - a version of the Lisp programming language specifically designed to run on processors with a limited amount of RAM.
- **ZBasic for ESP8266** (http://www.zbasic.net) – A subset of Microsoft's widely used Visual Basic 6, which has been adapted as a control language for the ZX microcontroller family and the ESP8266.
- **Zerynth** – IoT framework for programming ESP8266[14] and other microcontrollers in Python.

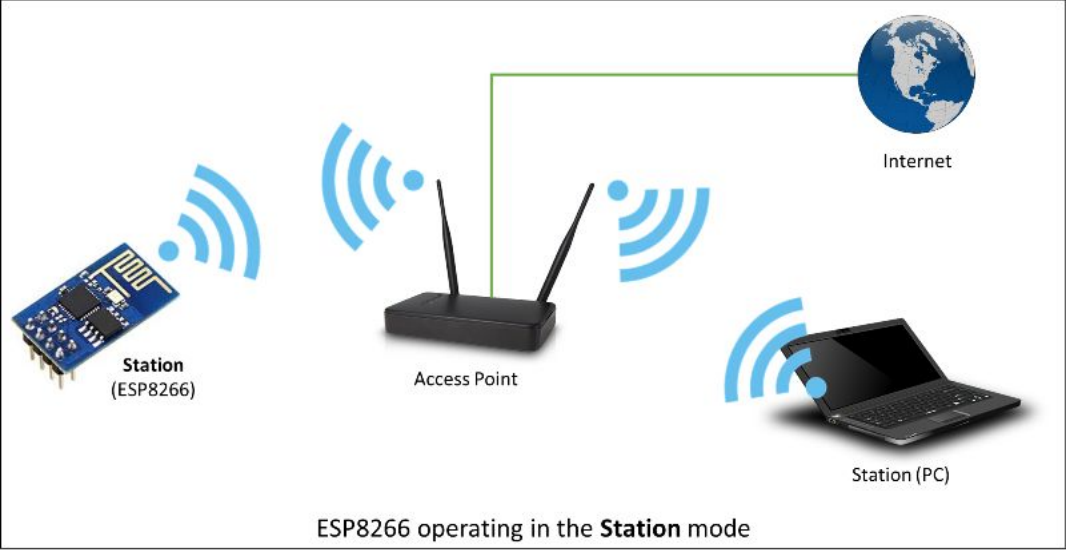
Documentação ESP8266 Arduino Core

https://arduino-esp8266.readthedocs.io/en/latest/index.html

Modos WiFi

- Station
- Soft Acess Point
- Client
- Server

https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html



ESP8266 operating in the **Station** mode

ESP8266 operating in the **Soft Access Point** mode

Station (STA)

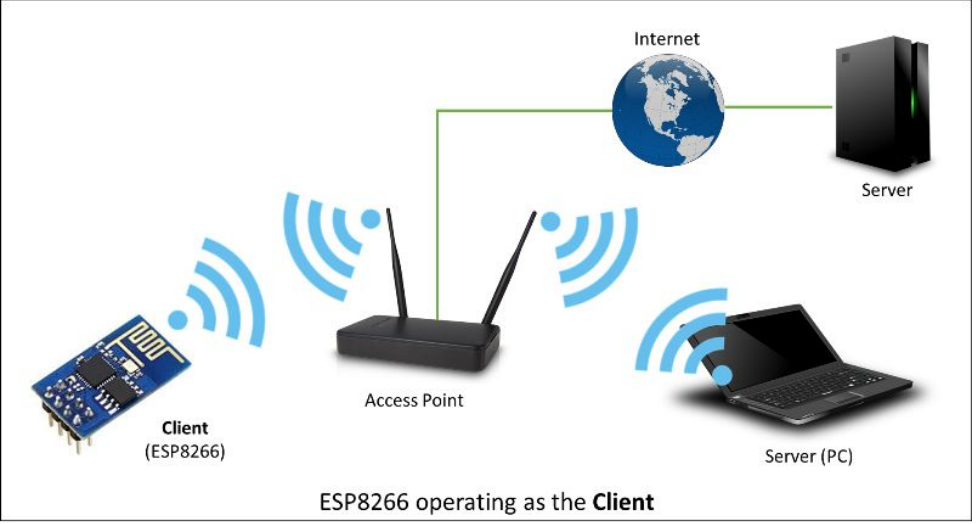
O modo **Station** (STA) é usado para conectar o módulo ESP a uma rede Wi-Fi estabelecida por um **access point** (AP).

Exemplos:
https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/station-examples.html

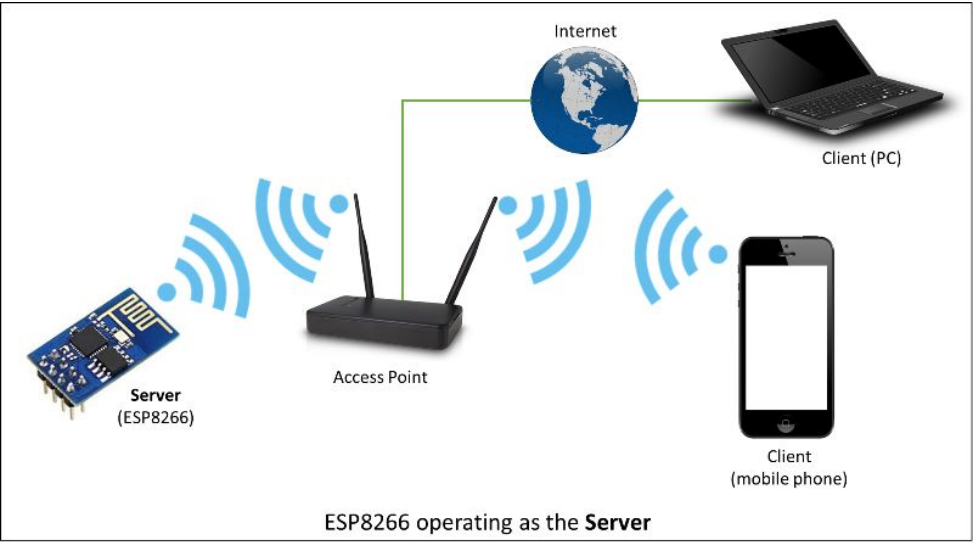
Soft Access Point (soft-AP)

Um **access point** (AP) é um dispositivo que fornece acesso à rede Wi-Fi para outros dispositivos (**stations**) e os conecta mais a uma rede com fio. O ESP8266 pode fornecer funcionalidade semelhante, exceto por não ter interface com uma rede com fio. Esse modo de operação é chamado de ponto de acesso flexível (**soft-AP**). O número máximo de estações que podem ser conectadas simultaneamente ao soft-AP pode ser definido de 0 a 8, mas o padrão é 4.

Exemplos:
https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/soft-access-point-examples.html



ESP8266 operating as the **Client**



ESP8266 operating as the **Server**

Client

A classe Client cria clientes que podem acessar serviços fornecidos por servidores para enviar, receber e processar dados.

Exemplos:
https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/client-examples.html

Server

A classe de servidor cria servidores que fornecem funcionalidade para outros programas ou dispositivos, chamados clientes.

Exemplos:
https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/server-examples.html

Rede Mesh

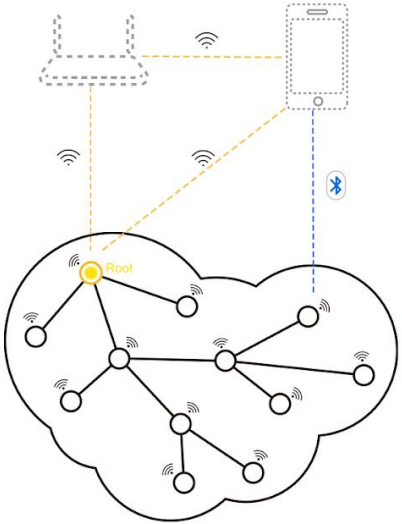
No need for expensive gateways — **Low-cost network**

Existing nodes connect to new ones automatically — **Mesh automatically formed**

Bluetooth connection between smartphone and mesh — **Wi-Fi connection**

Up to **10 Mbps** of data transfer rate — **Transfer music and videos**

Low-power sensors can connect to Wi-Fi mesh — **Low power consumption**



Every node works like a Router — Functional nodes without a Wi-Fi router

More nodes — Up to **1000 nodes** per mesh network

Long distance — Up to **200 meters** for a hop between two nodes

Wider coverage — "Hotspot" expands to a whole area

Easy implementation — Easy configuration of mesh without changing the network infrastructure

https://www.espressif.com/en/products/software/esp-mesh/overview

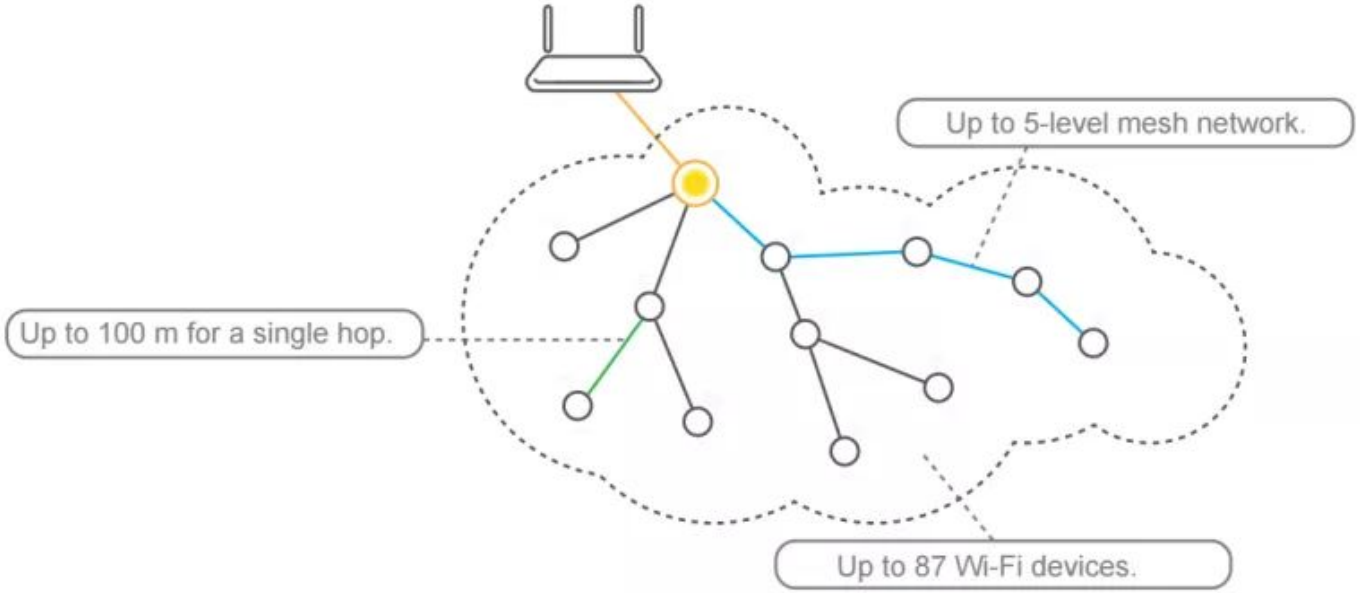


Figure 1-1. ESP-Mesh Network

https://www.dobitaobyte.com.br/como-fazer-rede-mesh-com-esp8266-parte-i/

Código

Duas formas:

1. Função

sending_data(String field, String data) -> #include <ESP8266HTTPClient.h>

https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266HTTPClient
https://www.filipeflop.com/blog/planta-iot-com-esp8266-nodemcu/

2. Biblioteca

#include <ThingSpeak.h>

https://github.com/mathworks/thingspeak-arduino
https://www.factoryforward.com/upload-sensor-data-thingspeak-using-nodemcu/
https://pplware.sapo.pt/tutoriais/nodemcu-registo-do-sinal-do-wi-fi-na-plataforma-thingspeak/

1. Função

```
// ThinkSpeak
void sending_data(String field, String data) { // Para field, mandar apenas o numero, em formato de string.
  String site = "http://api.thingspeak.com/update.json?api_key=";
  site.concat(writeAPIKey);
  site.concat("&field");
  site.concat(field);
  site.concat("=");
  site.concat(data);

  Serial.println(site);

  if (WiFi.status() == WL_CONNECTED) { //Check WiFi connection status
    //    Serial.print("Endereço IP: ");
    //    Serial.println(WiFi.localIP());
    HTTPClient http;  //Declare an object of class HTTPClient
    http.begin(site);
    int httpCode = http.GET();
    if (httpCode > 0) { //Check the returning code
      String payload = http.getString();   //Get the request response payload
      Serial.println(payload);                     //Print the response payload
    }
    http.end();   //Close connection
  }
}
```

2. Biblioteca

```
37
38  ThingSpeak.writeField(myChannelNumber, 1,val, myWriteAPIKey); //Update in ThingSpeak
39
```

Documentação

https://github.com/mathworks/thingspeak-arduino

ThingSpeak

O que é?

De acordo com seus desenvolvedores, "*o ThingSpeak é um aplicativo e API de Internet of Things (IoT) de código aberto para armazenar e recuperar dados de coisas usando o protocolo HTTP pela Internet ou por meio de uma rede local. O ThingSpeak permite a criação de aplicativos de registro de sensores , aplicativos de rastreamento de localização e uma rede social de coisas com atualizações de status*".

O ThingSpeak foi originalmente lançado pela ioBridge em 2010 como um serviço de suporte a aplicativos IoT.

O ThingSpeak integrou o suporte do software de computação numérica MATLAB da MathWorks , permitindo que os usuários do ThingSpeak analisassem e visualizassem os dados enviados usando o Matlab sem a necessidade da compra de uma licença Matlab da Mathworks.

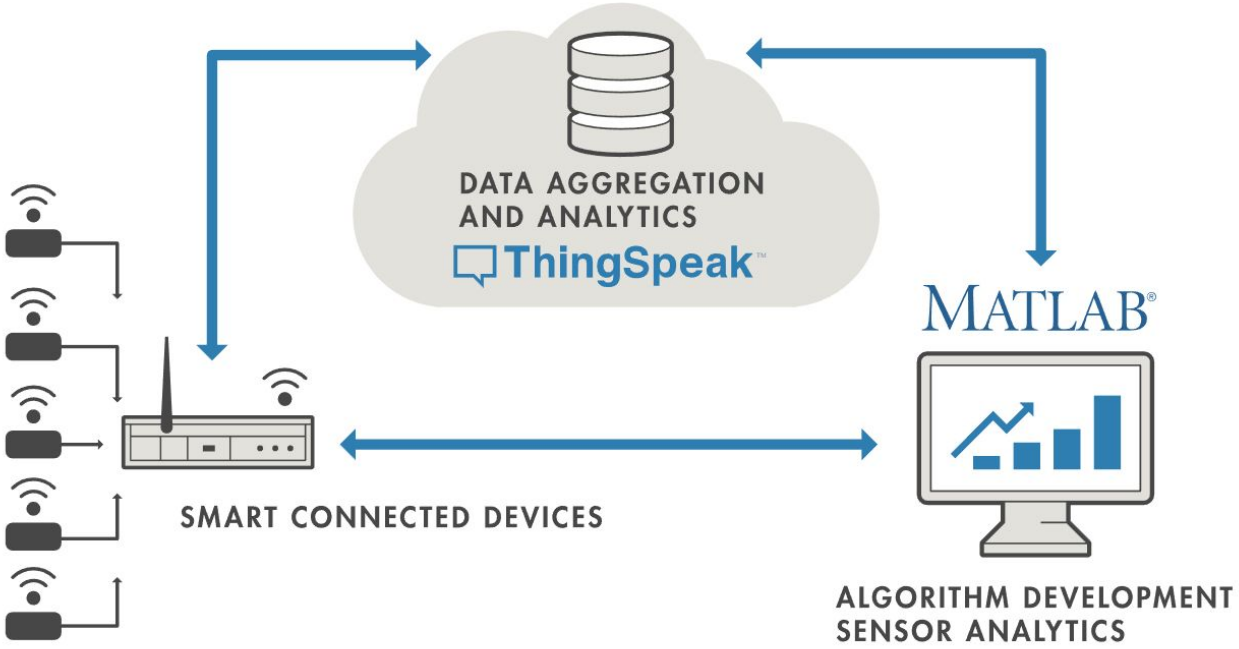
https://en.wikipedia.org/wiki/ThingSpeak

O que é?

O ThingSpeak tem um relacionamento estreito com a Mathworks , Inc. Na verdade, toda a documentação do ThingSpeak é incorporada ao site de documentação Matworks do Matworks e até mesmo permite que as contas de usuário registradas do Mathworks sejam credenciais de login válidas no site do ThingSpeak. Os termos de serviço e a política de privacidade do ThingSpeak.com estão entre o usuário concordante e a Mathworks, Inc.

O ThingSpeak tem sido objeto de artigos em sites especializados de " Maker ", como o Instructables , Codeproject e Channel 9 .

https://en.wikipedia.org/wiki/ThingSpeak
https://www.mathworks.com/help/thingspeak/



Collect

Send sensor data privately to the cloud.

Analyze

Analyze and visualize your data with MATLAB.

Act

Trigger a reaction.

https://thingspeak.com/pages/how_to

Como funciona?

O ThingSpeak ™ é uma plataforma de IoT que usa canais para armazenar dados enviados de aplicativos ou dispositivos. Você pode gravar dados em seus canais ThingSpeak de várias maneiras: Chamadas HTTP através de API **REST**, método Publicação **MQTT** e a **thingSpeakWritefunção** do MATLAB.

Escolha entre a API REST e a API MQTT

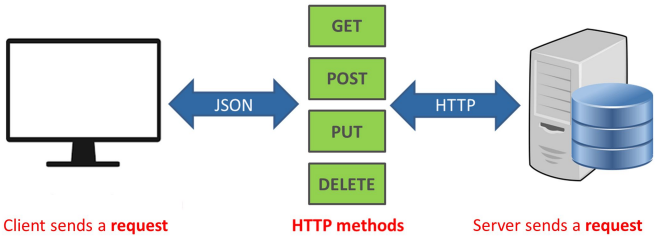
- **REST** é um estilo de arquitetura de transferência de estado representacional projetado como um modelo de solicitação / resposta que se comunica por **HTTP**.
- O **MQTT** é um modelo de publicação / assinatura executado por soquetes **TCP/IP** ou **WebSockets**. O MQTT sobre WebSockets pode ser protegido com SSL.

Você pode optar por atualizar dados para um canal ThingSpeak ™ usando uma solicitação **REST** POST ou usando o método **MQTT** Publish.

Você recupera dados do canal usando uma solicitação **REST** GET ou **MQTT** Subscribe.

https://www.mathworks.com/help/thingspeak/write-data.html
https://www.mathworks.com/help/thingspeak/choose-between-rest-and-mqtt.html

Como funciona?



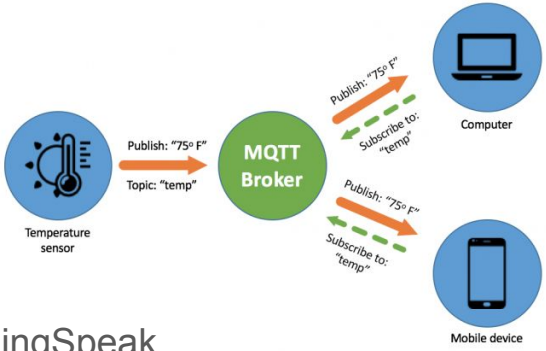
Quando usar o **REST**?

Nesses cenários, é útil usar chamadas **REST** para atualizar ou recuperar dados de um canal ThingSpeak.

- Você precisa recuperar dados históricos, como dados dentro de um intervalo de tempo definido.
- Você quer criar um canal, ler as configurações , limpar canal, deletar canal, ou criar gráficos.
- Você precisa de uma resposta para sua solicitação GET ou POST.

https://www.mathworks.com/help/thingspeak/choose-between-rest-and-mqtt.html
https://www.mathworks.com/help/thingspeak/rest-api.html

Como funciona?



Quando usar o **MQTT**?

Nesses cenários, é útil usar o **MQTT** para atualizar dados para um canal ThingSpeak.

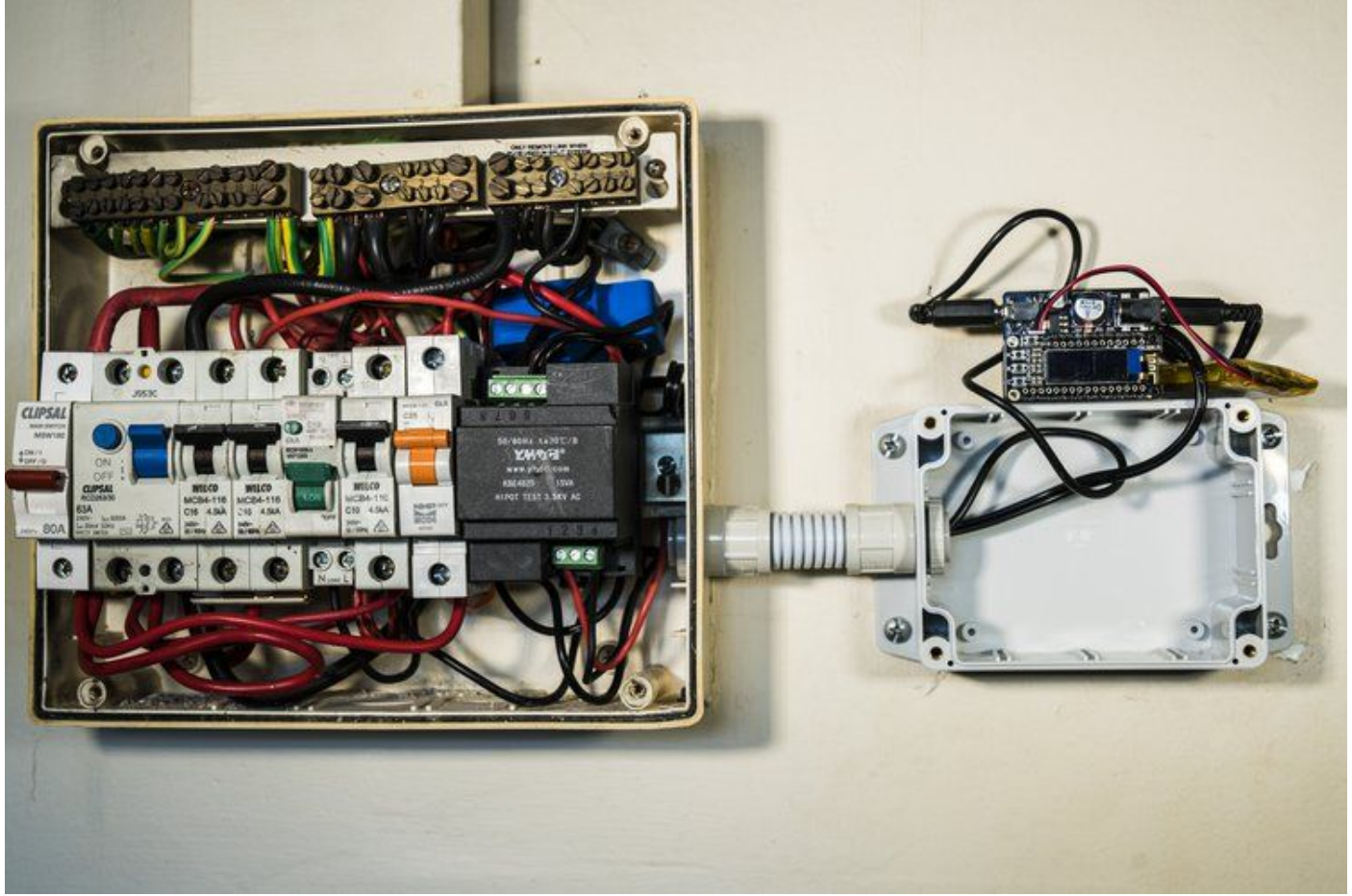
- Seu dispositivo tem restrições de energia e você deseja um consumo menor de bateria para enviar dados para o ThingSpeak. Além disso, uma operação **MQTT** PUBLISH normalmente é mais rápida nesse cenário.
- A conectividade do seu dispositivo é intermitente e você tem uso limitado de largura de banda.
- Você quer atualizações imediatas de dados postados em um canal.
- Você quer que as mensagens sejam enviadas para você em vez de precisar pesquisar no servidor novas mensagens.

https://www.mathworks.com/help/thingspeak/choose-between-rest-and-mqtt.html

Agora vamos ...

1. Criar canal
2. Gráfico de linha
3. Gauge
4. Gráfico de área e de linha com MATLAB
5. Aviso por email
6. Manipular dados de um canal para outro

https://toemat.com/light-dimming-robot/

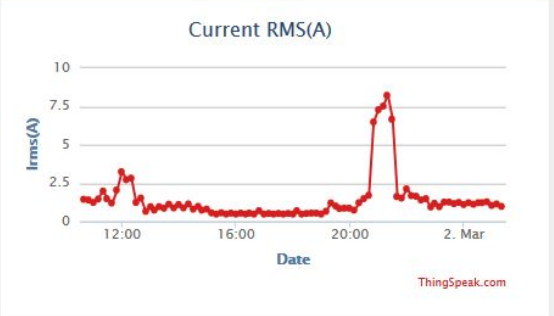


https://www.crowdsupply.com/whatnick/atm90e26-energy-monitor-kits
https://www.hackster.io/whatnick/esp8266-iot-energy-monitor-b199ed

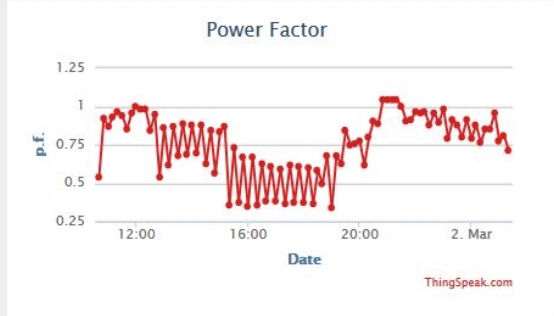
ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

Created: about a year ago
Updated: about a minute ago
Last entry: about a minute ago
Entries: 1383625

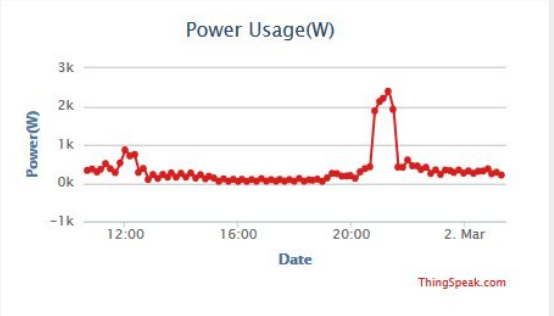
Current



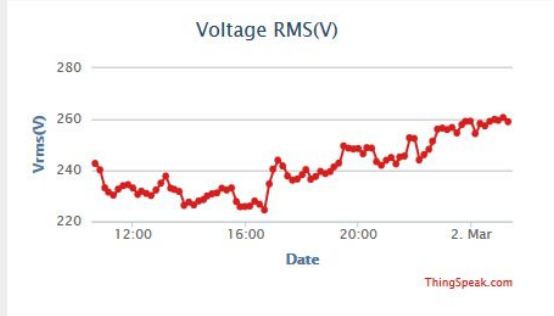
Power Factor



Power

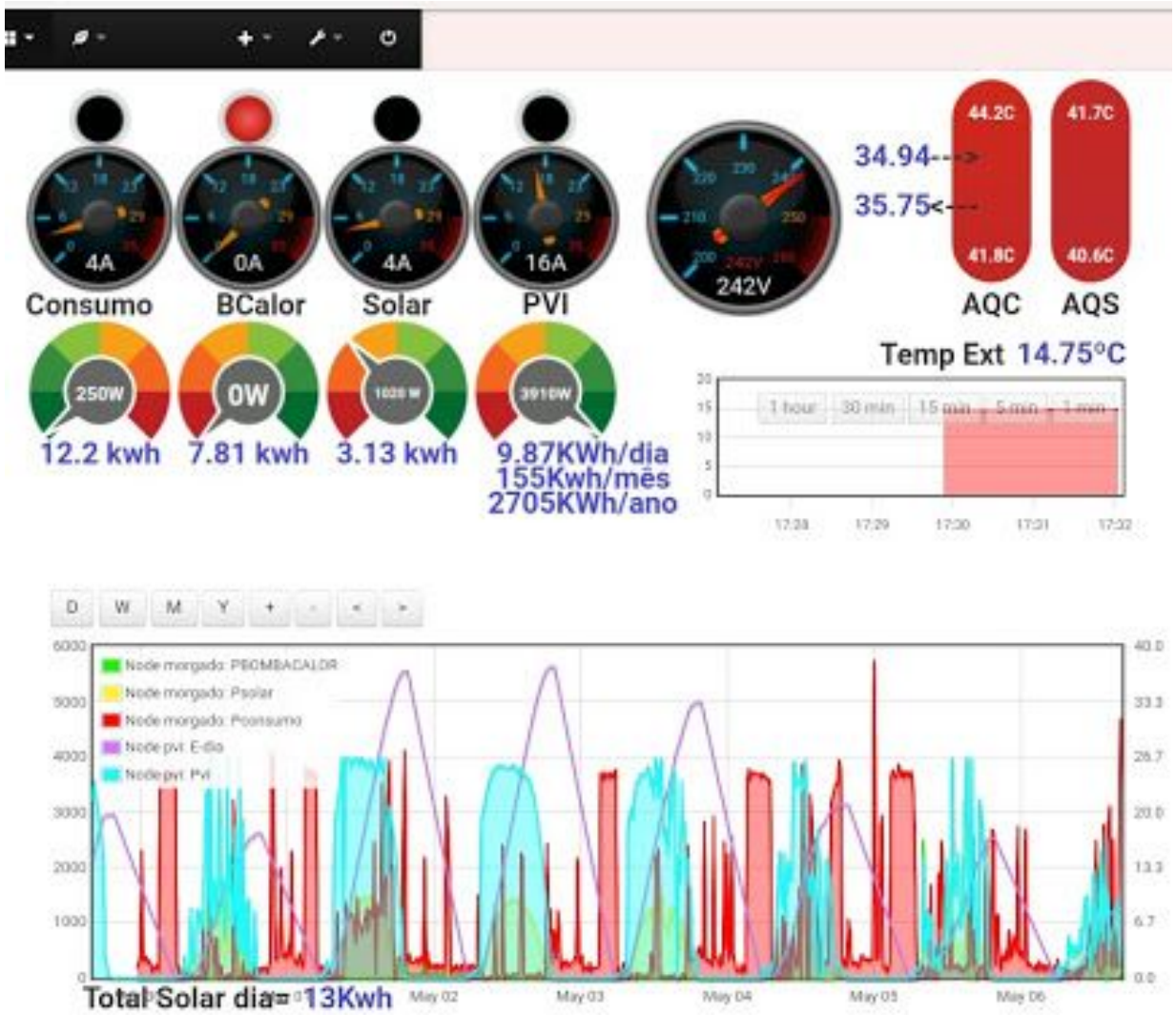


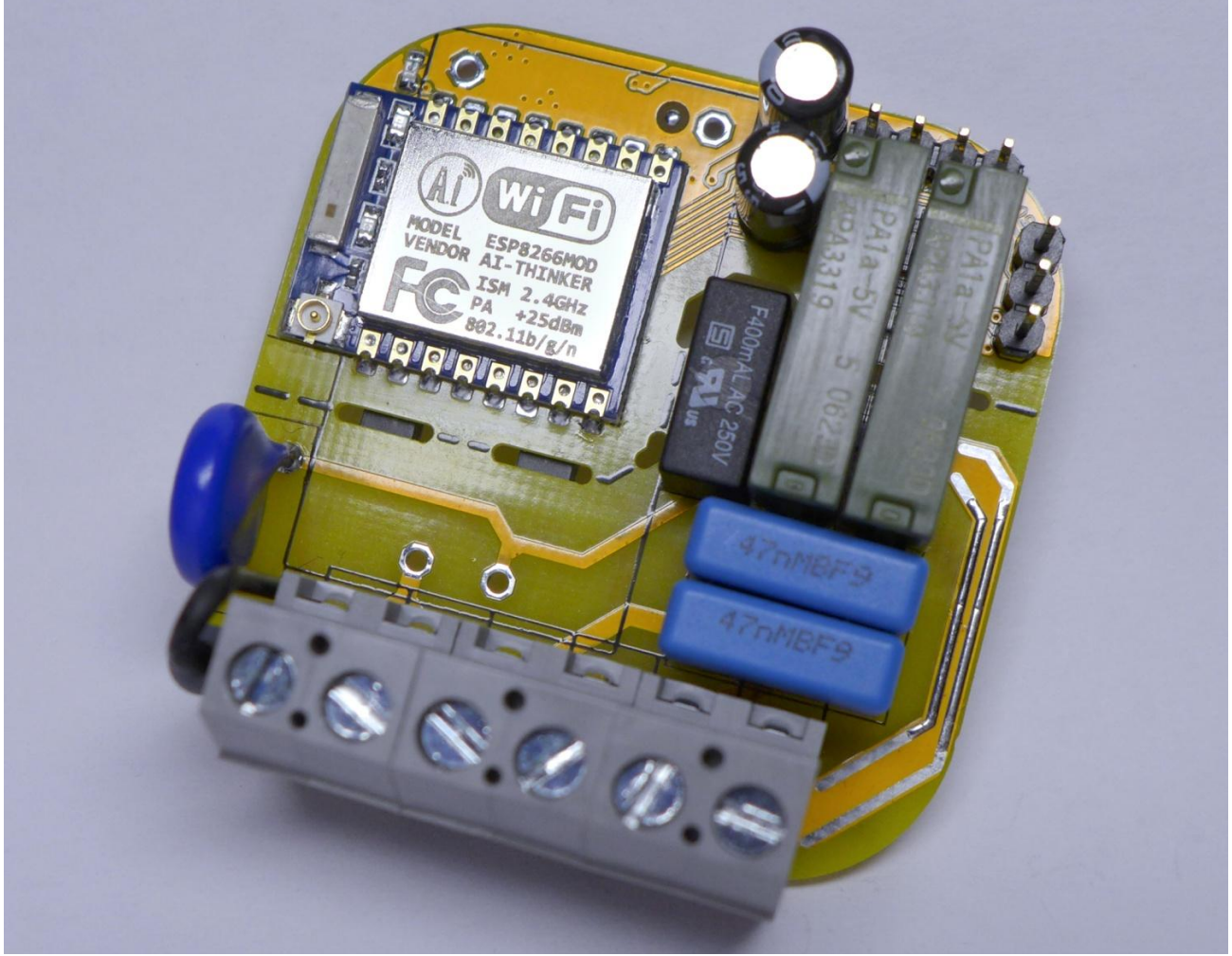
Voltage



http://tinkerman.cat/ailight-hackable-rgbw-light-bulb/#lightbox-gallery-hoZRx6A3/12/

https://github.com/apreb/eNode/blob/master/README.md


4A
0A
4A
16A
242V
Consumo
BCalor
Solar
PVI
34.94
35.75
44.2C
41.8C
41.7C
40.6C
AQC
AQS
250W
0W
12.2 kwh
7.81 kwh
3.13 kwh
9.87KWh/dia
155Kwh/mês
2705KWh/ano
Temp Ext 14.75°C
1 hour
30 min
15 min
5 min
1 min
D
W
M
Y
Node morgado: PBOMBACALOR
Node morgado: Psolar
Node morgado: Pconsumo
Node pvi: E-dia
Node pvi: Pvi
May 02
May 03
May 04
May 05
May 06
Total Solar dia = 13Kwh



https://luani.de/blog/esp8266-230v-io-modul/

https://www.briandorey.com/post/esp8266-mains-energy-monitor

Meter Readings

Gas: 16967538

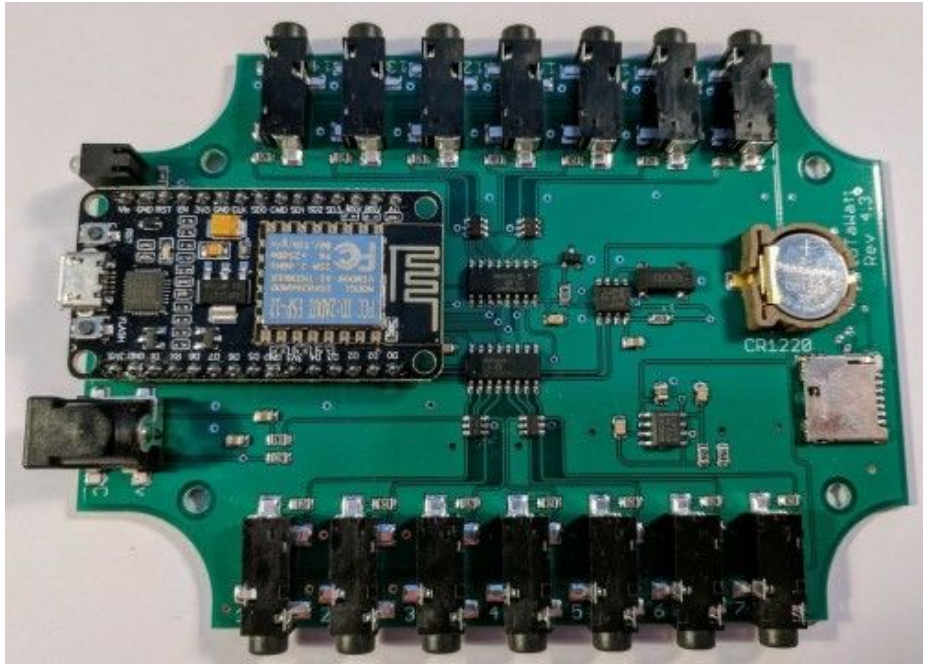
Electricity: 59653755

Mains Power

Mains Current: 5.165

Average Current over last Minute: 3.148

Max Current over last Minute: 5.270



http://iotawatt.com

Dashboards
Apps
Extra
Setup
Daily Usage:
16.12 Grid kWh 3.55 $
0.57 Solar kWh
16.69 Home kWh
590w
Total
421w
Grid
168w
Solar
121.2v
Voltage
0w
Air Con
0w
Well
67w
Fridge
138w
Barn
12w
Furnace
0w
Dryer
8w
Range
124w
Freezer
13w
Garage
228w
Misc.
Grid Import Realtime
1 hour
30 min
15 min
5 min
1 min
13:40
13:50
14:00
14:10
14:20
14:30
Grid kWh by Day
Dec 15
Dec 23
Dec 31
Jan 07
Grid Cost Per Day
Dec 15
Dec 23
Dec 31
Jan 07
30.9F
25.3 Low
31.6 High
Max Daily
6.8
3.7
Watts/m2
15
Light Lux: 1616
Light Percent: 2%
Humidity: 87.00 Low 54.00 High 87.00
Dew Point: 27.51 12.19 27.94
Pressure: 29.61 29.61 30.06
Rain Fall: 0.39
Temp. Humid. Low High
Sunroom 40.6 76 39 43
Barn 33.6 61 31 34
Barn Loft 30.0 65 25 30
Indoors 69.1 30 69 70
Lightning Strikes: 37
Storm Distance: 22

Referências

http://electrotropism17.rssing.com/chan-9545356/all_p3.html

https://zeptobars.com/en/read/Espressif-ESP32-Wi-Fi-Bluetooth-2.4Ghz-ISM

https://zeptobars.com/en/read/Espressif-ESP8266-wifi-serial-rs232-ESP8089-IoT

https://www.cnx-software.com/2017/02/06/oh-look-espressif-esp32-die-shot-thats-pretty/

https://www.esp8266.com/viewtopic.php?f=13&t=12469&start=4

https://www.forward.com.au/pfod/ESP8266/GPIOpins/ESP8266_01_pin_magic.html

Sallatiel Fernandes Freire Cardoso

sallatiel.freire@academico.ifpb.edu.br

sallatielfc@gmail.com

https://www.fb.com/sallatielfc

https://twitter.com/sallatielfc

https://github.com/Sallatiel

"Eu não tenho talentos especiais. Eu sou apenas apaixonadamente curioso."

- Albert Einstein