

iotMakers

Oficina de ESP8266  
#cpbr11

# Quem somos?

## Douglas Esteves

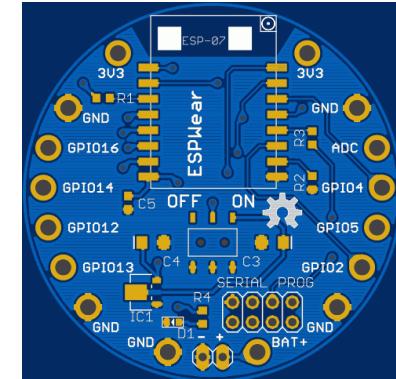
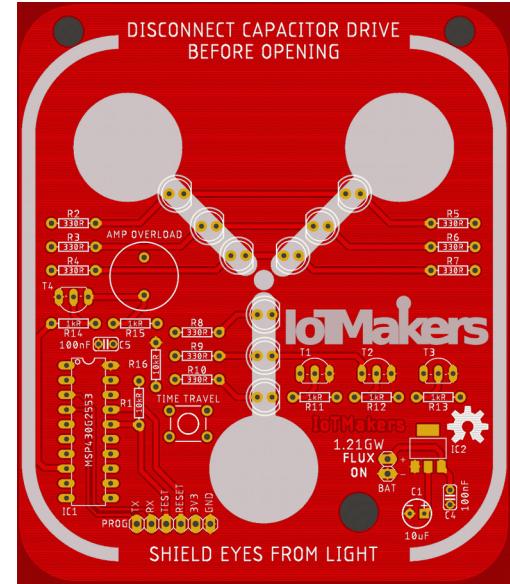
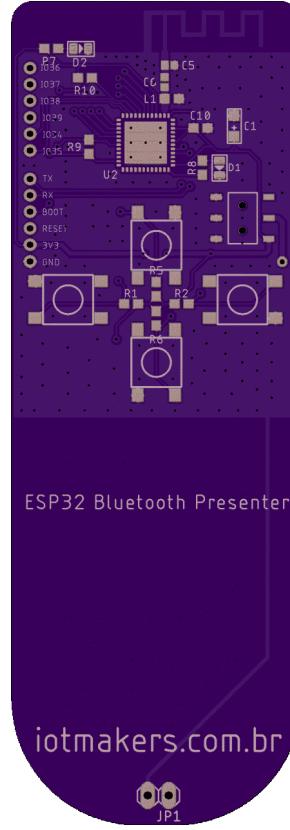


## Pedro Minatel



# IoT Makers

- Projetos de IoT
  - Hardware
    - WiFi
    - Bluetooth
    - SigFox
    - LoRa
  - Firmware
- Consultoria
- Treinamentos
- Kits de Desenvolvimento
- Hack Badges



Em breve! Cursos online de IoT



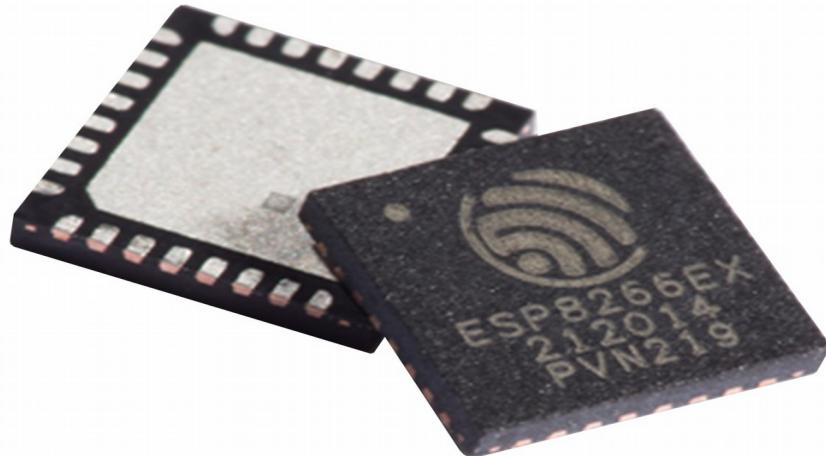
[cursosiotmakers.com.br](http://cursosiotmakers.com.br)

needinfo

# ESP8266

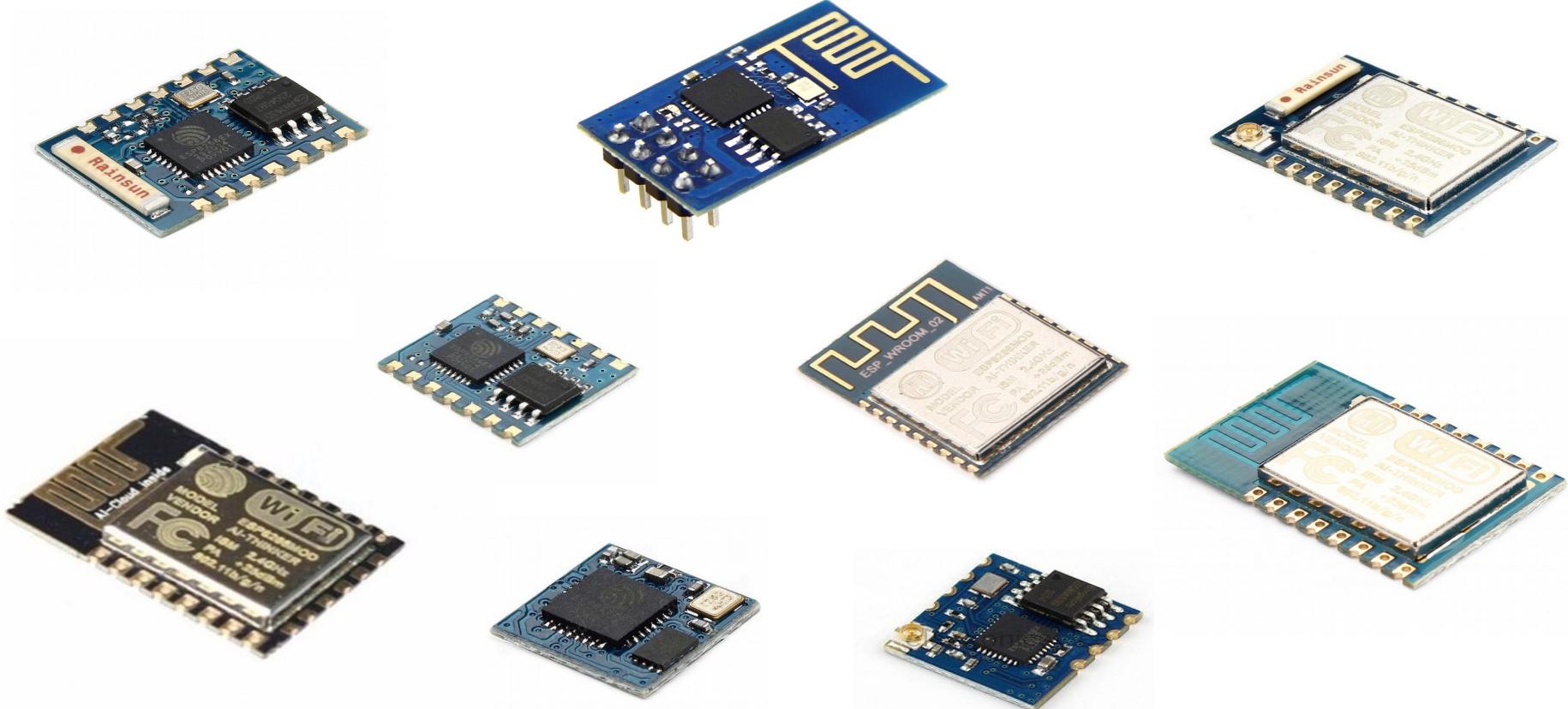
Hardware e Software

# ESP8266EX



乐鑫信息科技  
ESPRESSIF SYSTEMS

# Módulos com ESP8266



# ESpecificações Hardware

# ESpecificações

Categories	Items	Parameters
Wi-Fi	Standard	CCC / FCC / CE / TELEC / SRRC
	Protocols	802.11 b/g/n
	Frequency Range	2.4 G ~ 2.5 G (2400 M ~ 2483.5 M)
	Tx power	802.11 b: + 20 dBm
		802.11 g: + 17 dBm
		802.11 n: + 14 dBm
	Rx Sensitivity	802.11 b: - 91 dbm (11 Mbps)
		802.11 g: - 75 dbm (54 Mbps)
		802.11 n: - 72 dbm (MCS7)
	Antenna	PCB on-board, external, IPEX connector, ceramic chip
Hardware	Peripheral interface	UART / SDIO / SPI / I2C / I2S / IR Remote Control GPIO / PWM
	Operating voltage	3.0 V ~ 3.6 V
	Operating current	Average: 80mA
	Operating temperature range	-40 °C ~ 125 °C
	Storage temperature range	-40 °C ~ 125 °C

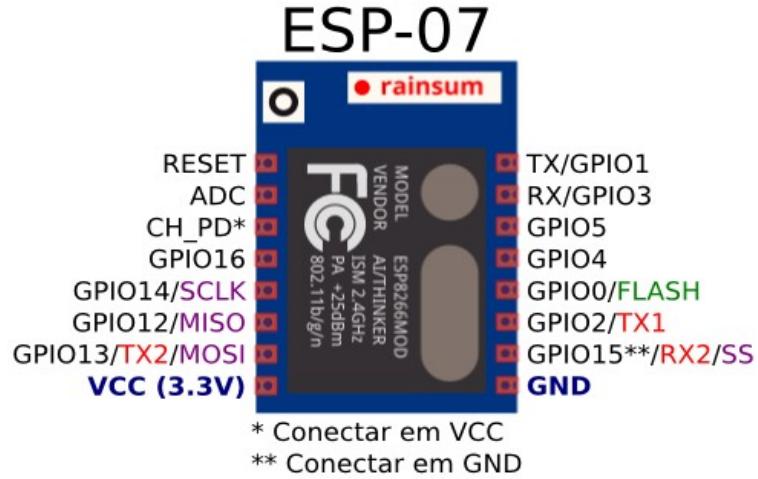
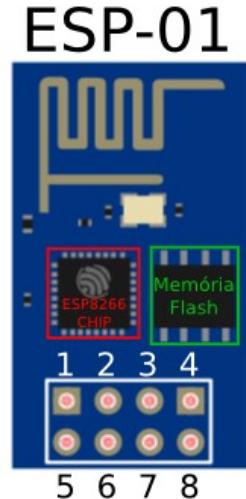
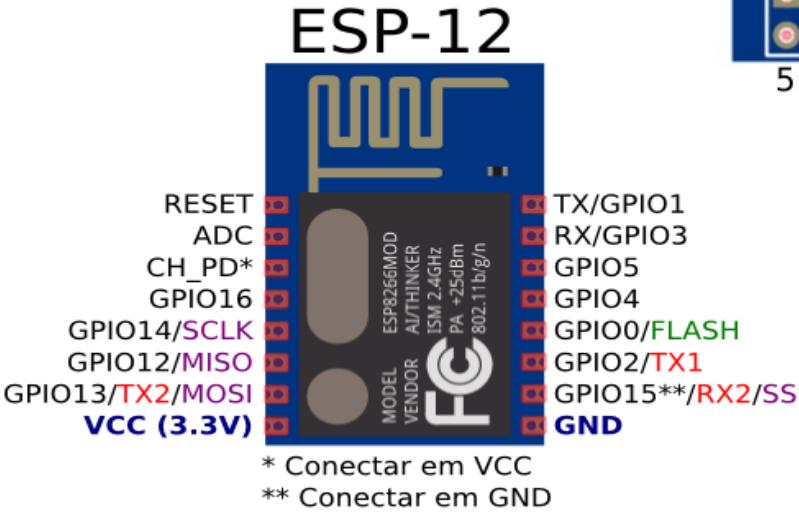
# ESpecificações

Categories	Items	Parameters
Software	Package size	QFN32-pin (5 mm x 5 mm)
	External interface	N/A
	Wi-Fi mode	station / softAP / SoftAP + station
	Security	WPA / WPA2
	Encryption	WEP / TKIP / AES
	Firmware upgrade	UART Download / OTA (via network)
	Software development	SDK for customised development / cloud server development
	Network Protocols	IPv4, TCP / UDP / HTTP / FTP
	User configuration	AT Instruction Set, Cloud Server, Android/ iOS App

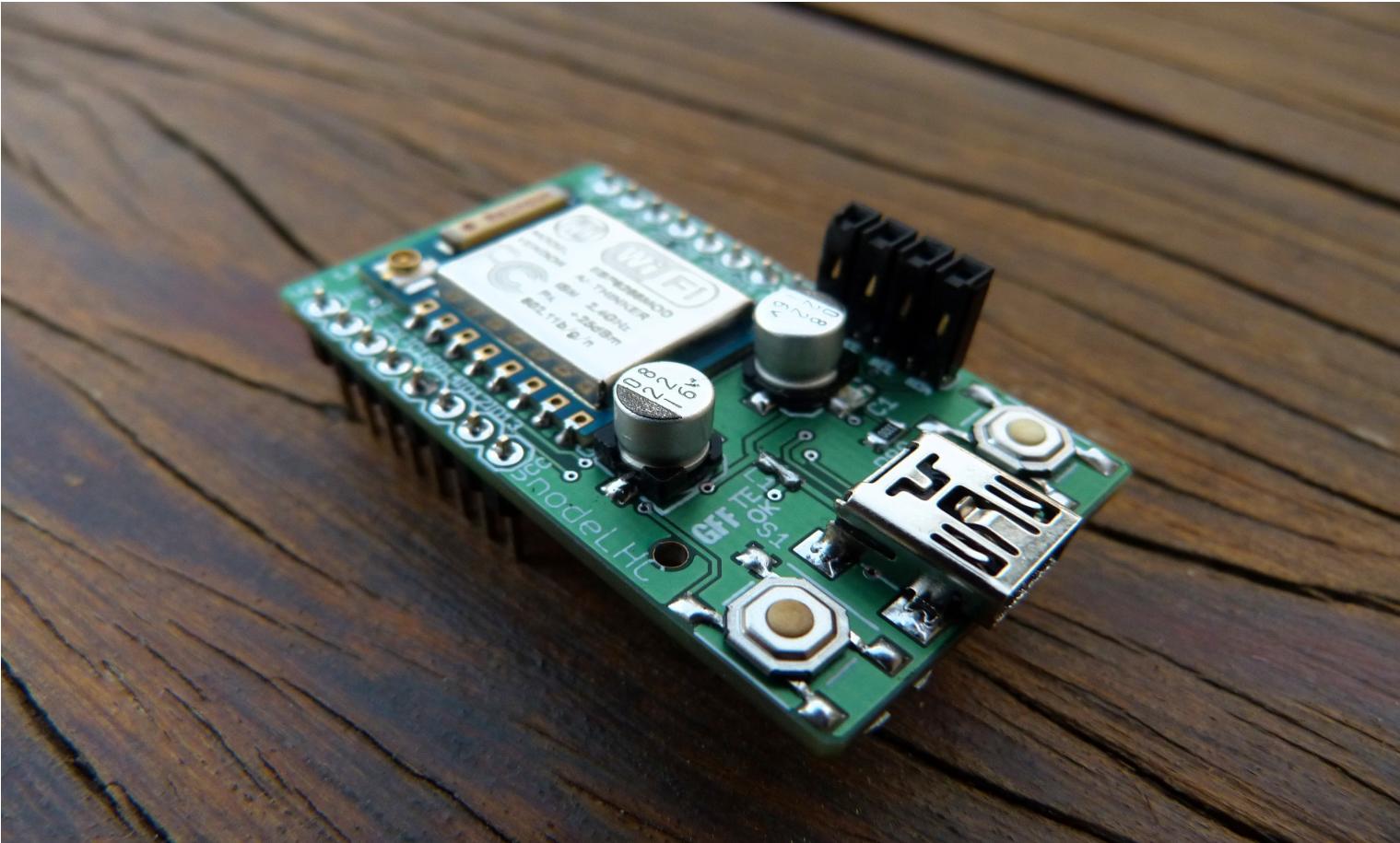
# Funcionalidades do ESP e Hardware

- O ESP é 3.3V. Em tensões maiores? Queima!
- O mesmo se aplica as GPIOs e a Serial/UART
- O ESP consome, em picos, cerca de 250mA
- Ligar no máximo 12mA em cada GPIO
- Algumas GPIOs tem funções no boot!
- O ADC tem 10bits (0-1023 & 0-1V)
- Somente 4 PWMs simultâneos (RGBW)

# Funcionalidades do ESP e Hardware



# nodeLHC



<http://pedrominateletronica.com.br/nodelhc-placa-de-desenvolvimento-para-o-esp8266/>

# Funcionalidades do ESP e Hardware

- GPIOs

GPIO	Inst Name	Function 0	Function 1	Function 2	Function 3	Function 4	At Reset	After Reset	Sleep
0	GPIO0_U	GPIO0	SPICS2			CLK_OUT	oe=0, wpu	wpu	oe=0
1	U0TXD_U	U0TXD	SPICS1		GPIO1	CLK_RTC	oe=0, wpu	wpu	oe=0
2	GPIO2_U	GPIO2	I2SO_WS	U1TXD		U0TXD	oe=0, wpu	wpu	oe=0
3	U0RXD_U	U0RXD	I2SO_DATA		GPIO3	CLK_XTAL	oe=0, wpu	wpu	oe=0
4	GPIO4_U	GPIO4	CLK_XTAL				oe=0		oe=0
5	GPIO5_U	GPIO5	CLK_RTC				oe=0		oe=0
6	SD_CLK_U	SD_CLK	SPICLK		GPIO6	U1CTS	oe=0		oe=0
7	SD_DATA0_U	SD_DATA0	SPIQ		GPIO7	U1TXD	oe=0		oe=0
8	SD_DATA1_U	SD_DATA1	SPIID		GPIO8	U1RXD	oe=0		oe=0
9	SD_DATA2_U	SD_DATA2	SPIHD		GPIO9	HSPIHD	oe=0		oe=0
10	SD_DATA3_U	SD_DATA3	SPIWP		GPIO10	HSPIWP	oe=0		oe=0
11	SD_CMD_U	SD_CMD	SPICSO		GPIO11	U1RTS	oe=0		oe=0
12	MTDI_U	MTDI	I2SI_DATA	HSPIQ_MISO	GPIO12	U0DTR	oe=0, wpu	wpu	oe=0
13	MTCK_U	MTCK	I2SI_BCK	HSPIID_MOSI	GPIO13	U0CTS	oe=0, wpu	wpu	oe=0
14	MTMS_U	MTMS	I2SI_WS	HSPICLK	GPIO14	U0DSR	oe=0, wpu	wpu	oe=0
15	MTDO_U	MTDO	I2SO_BCK	HSPICS	GPIO15	U0RTS	oe=0, wpu	wpu	oe=0
16	XPD_DCDC	XPD_DCDC	RTC_GPIO0	EXT_WAKEUP	DEEPSLEEP	BT_XTAL_EN	oe=1,wpd	oe=1,wpd	oe=1

# ESpecificações Software

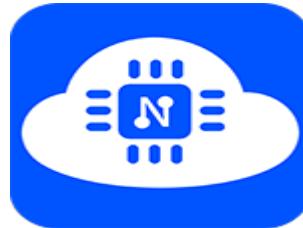
# Software

Programando o ESP8266:

- Programável em C nativo
- Programável com a plataforma Arduino
- Comandos AT (Firmware original)
- Programável em LUA (nodeMCU)
- MicroPython
- Basic

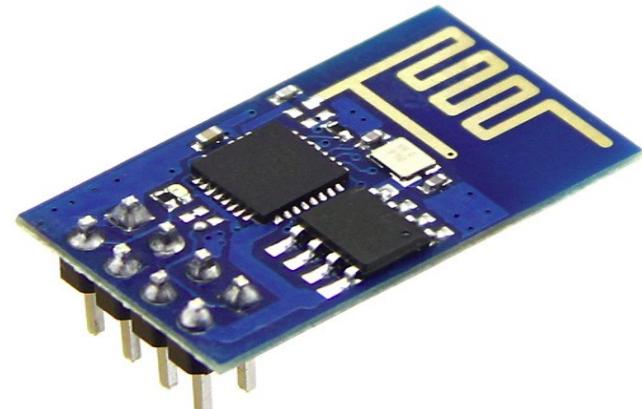
# Programação com nodeMCU

- Programação em LUA script
- Curta curva de aprendizado
- Se faz um código com poucas linhas de código
- [http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html)



# Programação com Arduino IDE

- Programação similar ao Arduino
- Reuso de códigos já desenvolvidos no Arduino
- Reaproveitamento de conhecimento



# Programação com Arduino IDE

- IDE 1.8.0 ou superior
  - Windows, Linux ou Mac
- Pacote ESP8266
  - Versão 2.3.0
  - [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
  - <http://bit.ly/1QeAVwh>
- Driver CH340 para alguns casos no Windows
- GNU/Linux
  - Manjaro, Ubuntu, Debian, Mint, ElementaryOS...



# PlatformIO

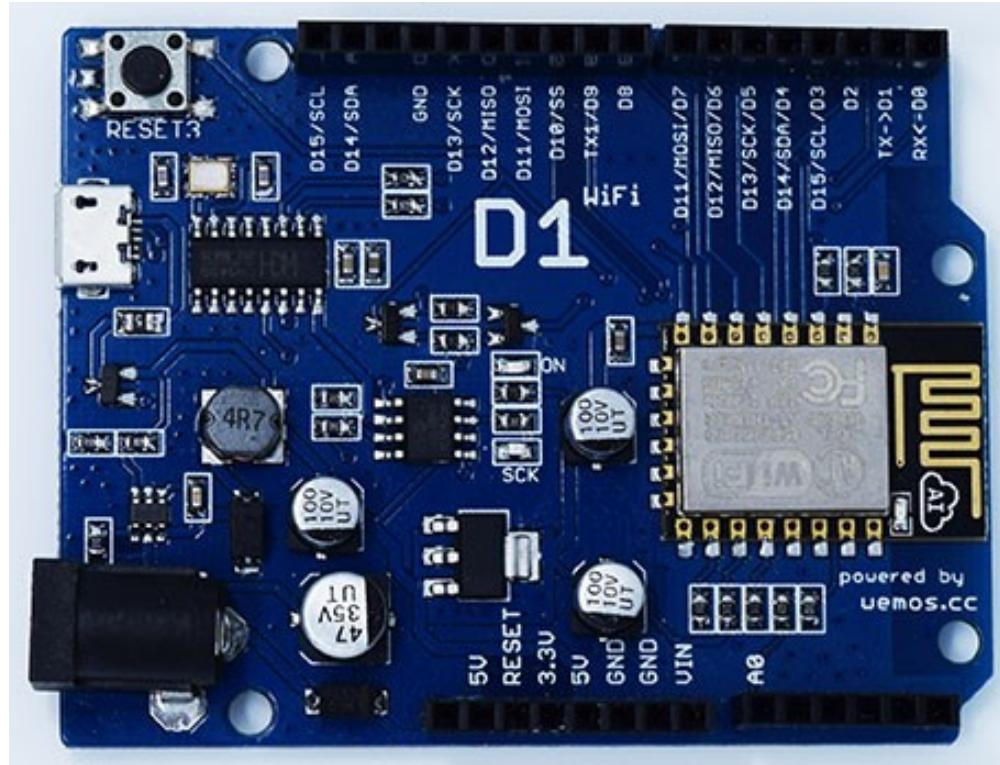
**PlatformIO is an open source ecosystem for IoT development**

Cross-platform build system. Continuous and IDE integration. Arduino and ARM mbed compatible

Primeiros passos com  
**ESP8266**

# Conhecendo a Wemos D1

# WeMos D1 (primeira versão)



<http://pedrominatel.com.br/esp32/wemos-d1-o-esp8266-com-cara-de-arduino/>

# WeMos D1

- 11 pinos de IO
- 1 ADC (max 3.2V)
- Micro USB (CH340)
- Entrada DC 9-24V
- ESP12
- 3V3
- 4MB Flash
- Clock 80/160MHz
- Formato Arduino

# Pinagem WeMos D1

Pin	Function	ESP8266 Pin
D0	RX	GPIO3
D1	TX	GPIO1
D2	IO	GPIO16
D3 (D15)	IO, SCL	GPIO5
D4 (D14)	IO, SDA	GPIO4
D5 (D13)	IO, SCK	GPIO14
D6 (D12)	IO, MISO	GPIO12
D7 (D11)	IO, MOSI	GPIO13
D8	IO, Pull-up	GPIO0
D9	IO, Pull-up, BUILTIN_LED	GPIO2
D10	IO, Pull-down, SS	GPIO15
A0	Analog Input	A0

# GPIO's

General Purpose Input Output

# O que é uma GPIO?

“**General Purpose Input/Output** (GPIO) são portas programáveis de entrada e saída de dados que são utilizadas para prover uma interface entre os periféricos e os microcontroladores/microprocessadores.”

[https://pt.wikipedia.org/wiki/General\\_Purpose\\_Input/Output](https://pt.wikipedia.org/wiki/General_Purpose_Input/Output)

# Exercício - Blink

Implementar um blink (*Hello world*) utilizando o LED.

**Tempo:** 5 minutos

```
#define LED 15  
  
void setup() {  
    pinMode(LED, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED, HIGH);  
    delay(1000);  
    digitalWrite(LED, LOW);  
    delay(1000);  
}
```

# **Exercício – Push button**

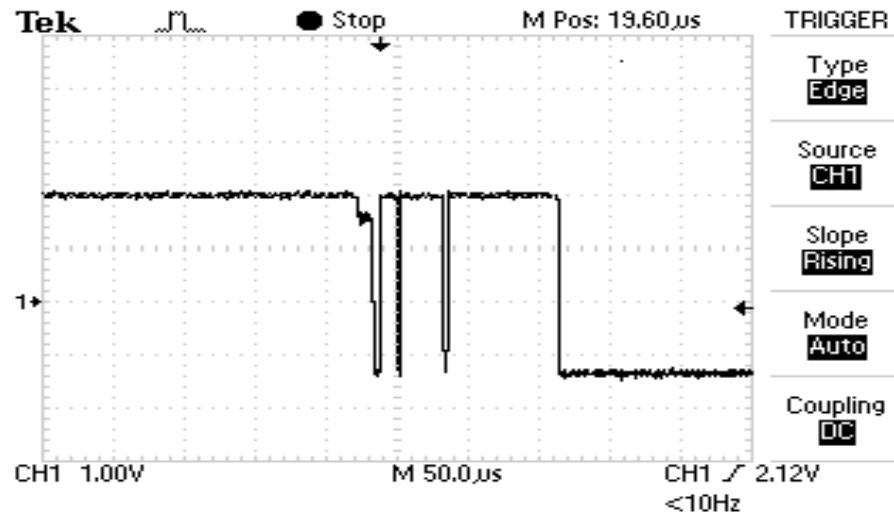
Implementar leitura do botão tact e acionar o LED.

**Tempo:** 5 minutos

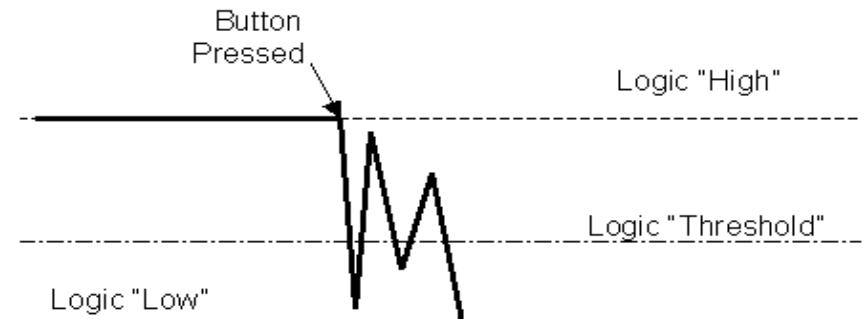
**Extra:** Implementar contato de selo.

```
#define BOTAO D13
#define LED  D10
int botao_status = 0;
void setup (void){
    pinMode(BOTAO, INPUT);
    pinMode(LED, OUTPUT);
}
void loop(void){
    botao_status = digitalRead(BOTAO);
    digitalWrite(LED,botao_status);
}
```

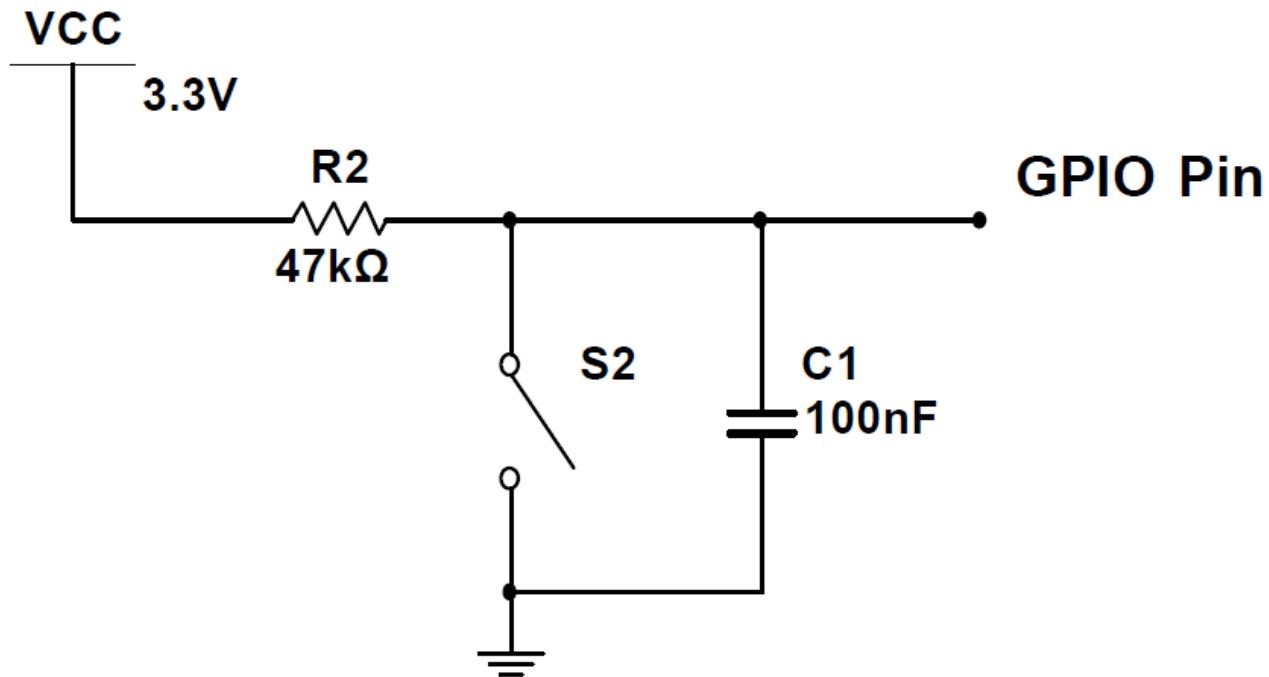
# Informação Extra - Debouncing



## Button “Bounce”



# Informação Extra - Debouncing



# PWM

Pulse Width Modulation

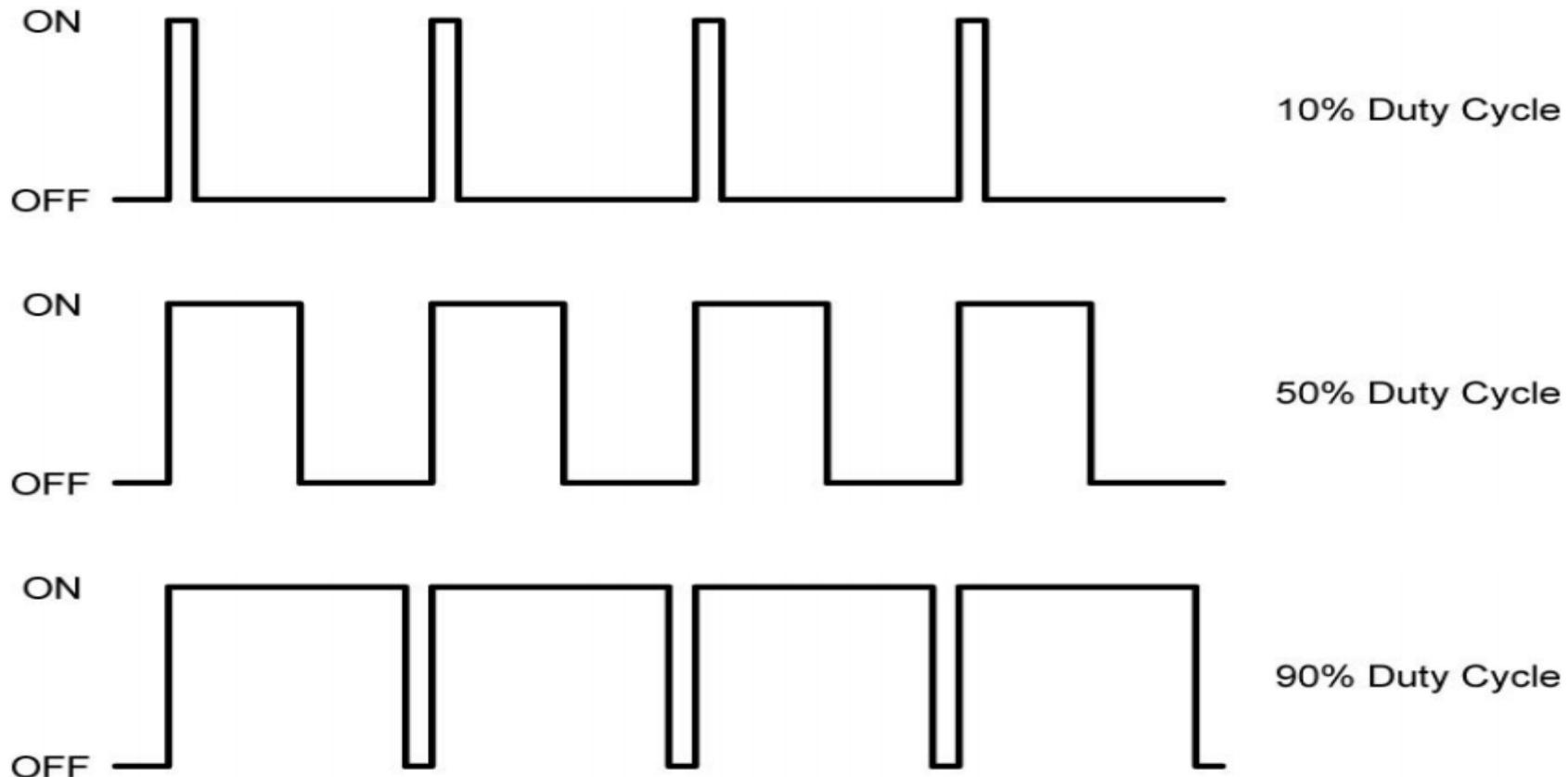
# Pulse Width Modulation

**PWM** significa "**Pulse Width Modulation**" ou Modulação de Largura de Pulso, ou seja, através da largura do pulso de uma onda quadrada é possível o controle de potência ou velocidade.



PW - Pulse Width  
(Largura do Pulso)

# Pulse Width Modulation



# ADC

## Entrada analógica

# ADC - Analog-to-digital converter

O conversor analógico-digital (frequentemente abreviado por conversor **A/D** ou **ADC**) é um dispositivo eletrônico capaz de gerar uma representação digital a partir de uma grandeza analógica, normalmente um sinal representado por um nível de tensão ou intensidade de corrente elétrica.

# **Exercício - ADC Read**

Implementar a leitura do potenciometro e controlar a luminosidade do LED por PWM.

**Tempo:** 10 minutos

```
int valor_l dr = 0;
```

```
#define LED D10
```

```
void setup(void){  
    Serial.begin(115200);  
    pinMode(LED, OUTPUT);  
}
```

```
void read_l dr(void){  
    valor_l dr = analogRead(A0);  
    Serial.print("Valor LDR: ");  
    Serial.println(valor_l dr);  
    analogWrite(LED, valor_l dr );  
}
```

```
void loop(void){  
    read_l dr();  
    delay(50);  
}
```

WiFi

# **Modos do WiFi**

- **STATION**
  - Utilizado para a conexão entre o ESP e um Access Point
- **SOFTAP**
  - Utilizado para o ESP ser um Access Point

Bibliotecas: **ESP8266WiFi.h & ESP8266WiFiMulti.h**

Enviando e recebendo dados da  
nuvem

# Serviços de IoT



# Blynk



<http://www.blynk.cc/>

# Obrigado!

## Redes Sociais

[instagram.com/iotmakers](https://instagram.com/iotmakers)

[twitter.com/iotmakers](https://twitter.com/iotmakers)

[facebook.com/IoTMakers/](https://facebook.com/IoTMakers/)

Projetos Open Hardware  
[hackaday.io/pedrominate](https://hackaday.io/pedrominate)