

Skriptování ESP8266

Programování WiFi mikročipu ve skriptovacím jazyce?
Důvody pro a proti, a ukázky jak na to.

ESP moduly od Ai-Thinker



ESP-01



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12

ESP8266

- WiFi SoC
- relativně výkonný 32bit CPU @ 80/160 MHz
- relativně dost paměti: 96 kB / 50 kB
- SDK je v C/C++
- výkon nazbyt svádí...

Jak programovat ESP8266

- (AT příkazy)
- oficiální SDK – verze Non-OS nebo FreeRTOS
- ESP8266 Arduino Core - „plugin“ do Arduino IDE
- skriptovací jazyky (díky výkonu CPU a velikosti RAM)

Důvody pro skriptování

- rychlost vývoje programu
- pohodlnost vývoje programu
- přenositelnost programu na jiné platformy
- snadnější ladění

Důvody proti skriptování

- větší vzdálenost od „železa“
- další vrstva s potenciálními vlastními chybami
- program bude zřejmě běžet pomaleji

Skriptovací jazyky a ESP8266

- Lua – www.nodemcu.com
- MicroPython – www.micropython.org
- Javascript – www.espruino.com
- Basic – www.esp8266basic.com

Lua

- Lua je mocný, rychlý, lehký skriptovací jazyk
- kombinuje jednoduchou procedurální syntaxi s mocným popisem dat pomocí asociativních polí a rozšiřitelné syntaxe
- je dynamicky typovaný, za běhu interpretuje bajtkód ve virtuální mašině, má automatickou správu paměti
- ideální pro konfigurace, skriptování a rychlý vývoj
- <http://www.luafaq.org/>

Jak se naučit programovat v Lue?

- František Fuka

<http://www.fffilm.name/2013/11/lua-krasa-v-jednoduchosti-video-kniha.html>

- Pavel Tišnovský

<https://www.root.cz/serialy/programovaci-jazyk-lua/>

- další zdroje:

<http://www.luafaq.org/gotchas.html>

<http://lua-users.org/wiki/LearningLua>

<https://learnxinyminutes.com/docs/lua/>

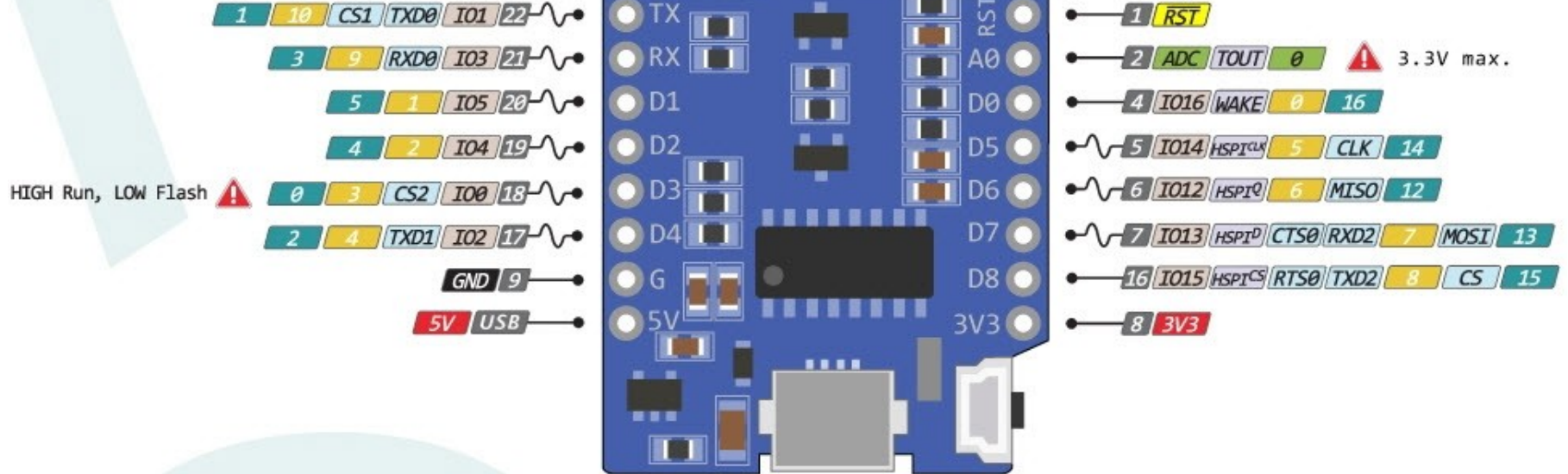
Kde se vzala Lua na ESP8266

- NodeMCU = open source vývojová platforma pro IoT
- kombinuje hardware a Lua firmware (eLua port)
- v základu je spiffs, souborový filesystem v paměti
- od léta 2015 vývoj pokračuje dobrovolnický
- za rok už NodeMCU obsahoval více než 40 modulů:
http, mqtt, onewire, u8g, ...

Na jakém HW programovat?

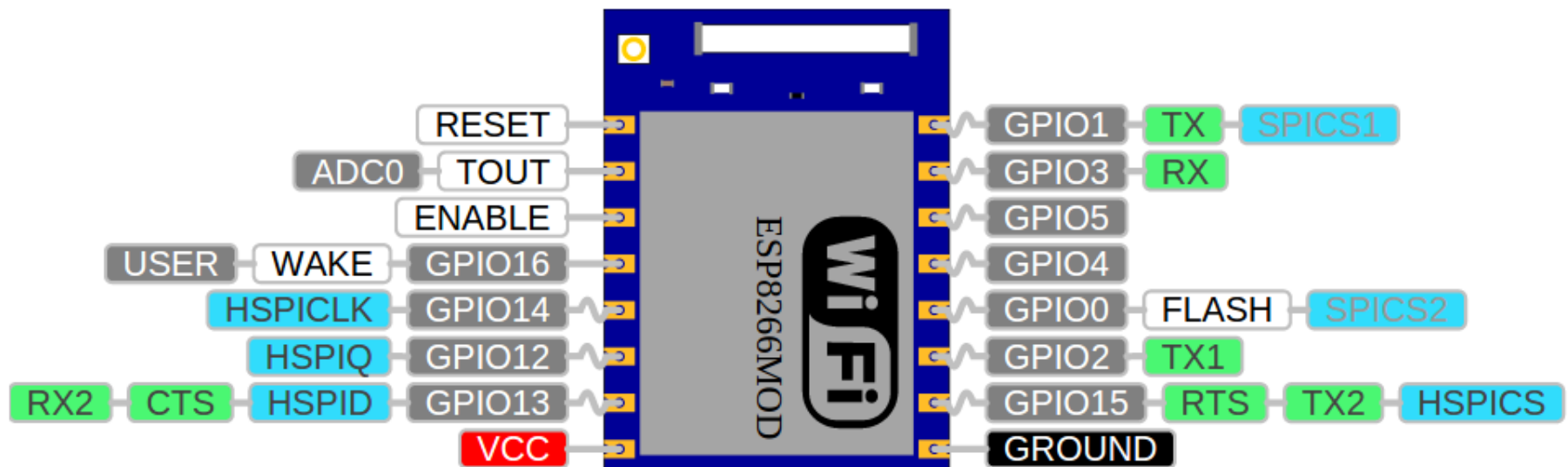
- originál NodeMCU developer kity
- holé moduly ESP v01-12 (aspoň 1 MB flash)
- Wemos a Wemos D1 Mini
- Itead WiFi Sonoff relé či Smart žárovky
- a samozřejmě vlastní HW (smart watch, ...)

mini



Power	Physical Pin	PWM/I2C/1-Wire
GND	Port Pin	Serial Pin
Control	Pin function	Analog Pin
	NodeMCU	Arduino

ESP v07/12 – popis vývodů



Mapování NodeMCU ↔ ESP8266

IO index	ESP8266 pin	IO index	ESP8266 pin
0[*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		

Omezení Luy na ESP8266

- <http://www.eluaproject.net/>
- Lua verze 5.1, chybí moduly debug, math a os
- <http://nodemcu.readthedocs.io/en/dev/en/lua-developer-faq/>
- integer a floating point verze firmware
- událostmi řízené programování (jako Node.js)
- kód by neměl běžet déle než 10 ms v kuse
- omezení daná velikostí volné RAM

Kde vzít Luu pro ESP8266

- vývoj pod OpenSource licencí na GitHubu
- <https://github.com/nodemcu/nodemcu-firmware>
- Docker image s build environmentem
- on-line buildovací služba www.nodemcu-build.com

Verze Luy pro ESP8266

- vývojové větve master a dev
- master = stabilní, dev = aktuální vývoj
- založeno na Espressif Non-OS SDK
- pozor na kompatibilitu při větším upgrade

<http://nodemcu.readthedocs.io/en/dev/en/flash/#upgrading-firmware>

„Flashování“ firmware

- esptool.py – github.com/themadinventor/esptool
- <https://nodemcu.readthedocs.io/en/master/en/flash/>

NodeMCU příkazová řádka

- terminálový program – UART na 115200 bps 8N1
- DTR pulz resetuje ESP8266
- po startu se automaticky spouští init.lua
- programy nahráváme pomocí nodemcu-uploader.py
<http://nodemcu.readthedocs.io/en/dev/en/upload/>

Esplorer

- ruský pokus o NodeMCU/Lua IDE v Javě
- žije na <http://esp8266.ru/esplorer/>
- obsahuje editor, uploader, terminál
- verze 0.2.0 odpovídá kvalitě programu
- první pokusy o podporu MicroPythonu

NodeMCU Web IDE

- Javascript i Basic měly web IDE, proč ne Lua?
- <https://github.com/joysfera/nodemcu-web-ide>
- Primitivní, ale překvapivě pohodlné
- Plán: lepší editor se syntax highlighting

Příklad v Lue: rozsvítit LED

On-board LED je připojena mezi TX1 = GPIO2 a VCC

```
gpio.mode(4, gpio.OUTPUT)
```

```
gpio.write(4, gpio.LOW)
```

Příklad: blikání LED bez čekání

```
LED_PIN = 4
```

```
gpio.mode(LED_PIN, gpio.OUTPUT)
```

```
value = true
```

```
tmr.alarm(0, 500, tmr.ALARM_AUTO, function ()
```

```
    gpio.write(LED_PIN, value and gpio.HIGH or gpio.LOW)
```

```
    value = not value
```

```
end)
```

Příklad: připojení k AP

```
wifi.setmode(wifi.STATION)
wifi.sta.config("openalt","openalt")
tmr.alarm(1, 1000, 1, function()
  if wifi.sta.getip() == nil then
    print("Připojuji...")
  else
    tmr.stop(1)
    print("Připojeno, IP je " .. wifi.sta.getip())
  end
end)
```


Příklad: HTTP GET

```
http.get("http://httpbin.org/ip", nil, function(code,
data)
    if (code < 0) then
        print("HTTP požadavek selhal")
    else
        print(code, data)
    end
end)
end)
```

Příklad: web server

```
srv = net.createServer(net.TCP)
srv:listen(80, function(conn)
  conn:on("receive", function(sck, payload)
    print(payload)
    sck:send("HTTP/1.0 200 OK\r\nContent-Type:
text/html\r\n\r\n<h1>Ahoj z NodeMCU.</h1>")
  end)
  conn:on("sent", function(sck) sck:close() end)
end)
```

Příklad: WS2812 LED pásek

```
ws2812.init()
```

```
-- první dvě RGB LED budou zelené
```

```
ws2812.write(string.char(255, 0, 0, 255, 0, 0))
```

Příklad: WS2812 LED efekt

```
ws2812.init()
local i, buffer = 0, ws2812.newBuffer(10, 3)
buffer:fill(0, 0, 0)
tmr.alarm(0, 50, 1, function()
    i=i+1
    buffer:fade(2)
    buffer:set(i%buffer:size()+1, string.char(0, 0, 255))
    ws2812.write(buffer)
end)
```

ESP32

Stručné představení horké novinky fy Espressif

ESP8266

- původně WiFi převodník k MCU bez konektivity
- 32-bit CPU na 80/160 MHz, 80 % volné kapacity
- 96 kB datové paměti
- 16 GPIO pinů
- obvyklá rozhraní: UART, SPI, I2S, ADC, PWM
- spící režimy
- drtivě nízká cena (< 40 Kč)

ESP8266 v praxi

- uzavřené SDK, verze NonOS a OS (FreeRTOS)
- slabá dokumentace, původně pouze v čínštině
- blob pro WiFi stack, asynchronní volání „userspace“
- CPU nesmíme zdržet déle než 10 ms
- polovina RAM obsazena WiFi stackem
- z 16 GPIO použitelných tak 10
- pouze jeden ADC (a jen 10bitový)
- I2C jen softwarově

ESP32 = řešení všech neduhů

- komplet open source vývoj, od začátku na GitHubu
- pouze FreeRTOS verze (žádný Non-OS)
- dvě CPU jádra Tensilica LX108 na 160 (240?) MHz!
- jedno CPU aplikační, druhé pro WiFi? Nově i plný SMP
- 512 kB RAM! Volných pro uživatele cca 170 kB
- 36 GPIO, víc a přesnější ADC, nově DAC, HW I2C, touch senzory
- důraz na bezpečnost, HW akcelerace šifrování
- důraz na nízkou spotřebu, sleep mody, lepší výrobní technologie
- Bluetooth LE a rychlejší WiFi (150 Mbps)
- RTC, ULP koprocessor, Ethernet, CAN, IR, ..., ..., ...

ESP32 modul

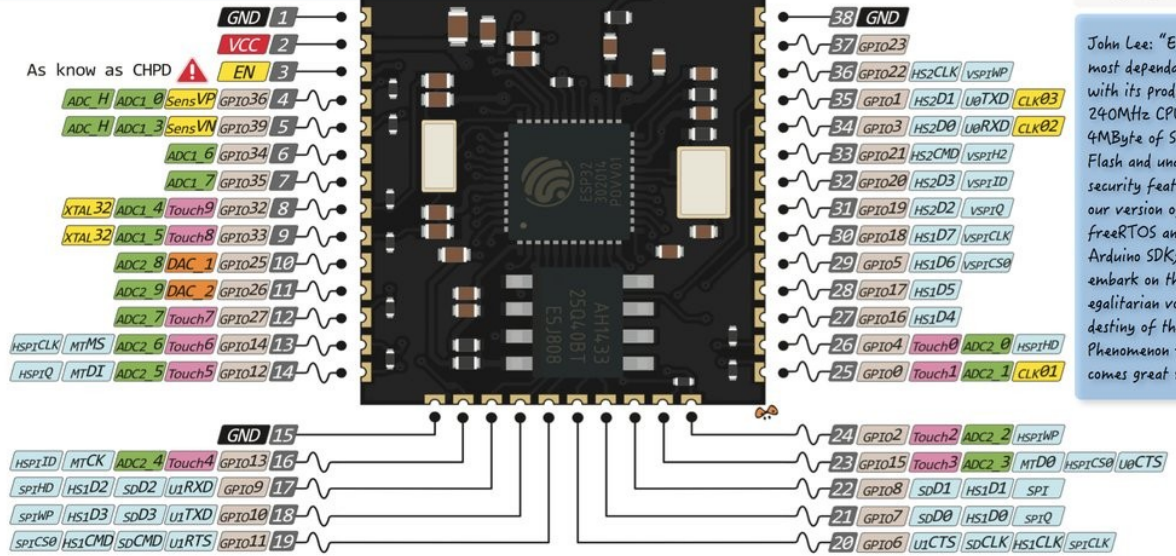
- 802.11BGN HT40 • 150Mbps • Bluetooth • Bluetooth LE • Ultra long range
- Dual core CPU • Dual FPU • Iterative 32x32 multiplier • 32-bit integer divider
- 40-bit MAC (multiply and accumulate) • 240MHz CPU clock • PWM • data cache
- 512 kByte SRAM • 8 MByte Flash • Timers • 32 x GPIO • 4uA interactive low power sleep
- Low power retention memory • Memory protection unit • 4096-bit RSA Accelerator
- ECC accelerator • AES-256 • Flash encryption • Secure boot • 1024-bit OTP memory
- 2x I2S • 3x SPI • 3x I2C • SD Card interface LCD interface • PWM motor drivers
- Capacitive touch sensor • Hall sensor • Low noise amplifier • Slow clock • Secure boot
- 12-channel dual ADC • -40°C to 125°C ambient temperature operation
- Dual 10-bit DAC • Calibrated 8MHz clock • 160kHz low power RTC
- Secure boot • 32kHz low power oscillator • 12 years longevity commitment
- 2.0V to 3.6V wide supply voltage range operation

ESP32 News!

Scheduled
Release: 1st
September 2016

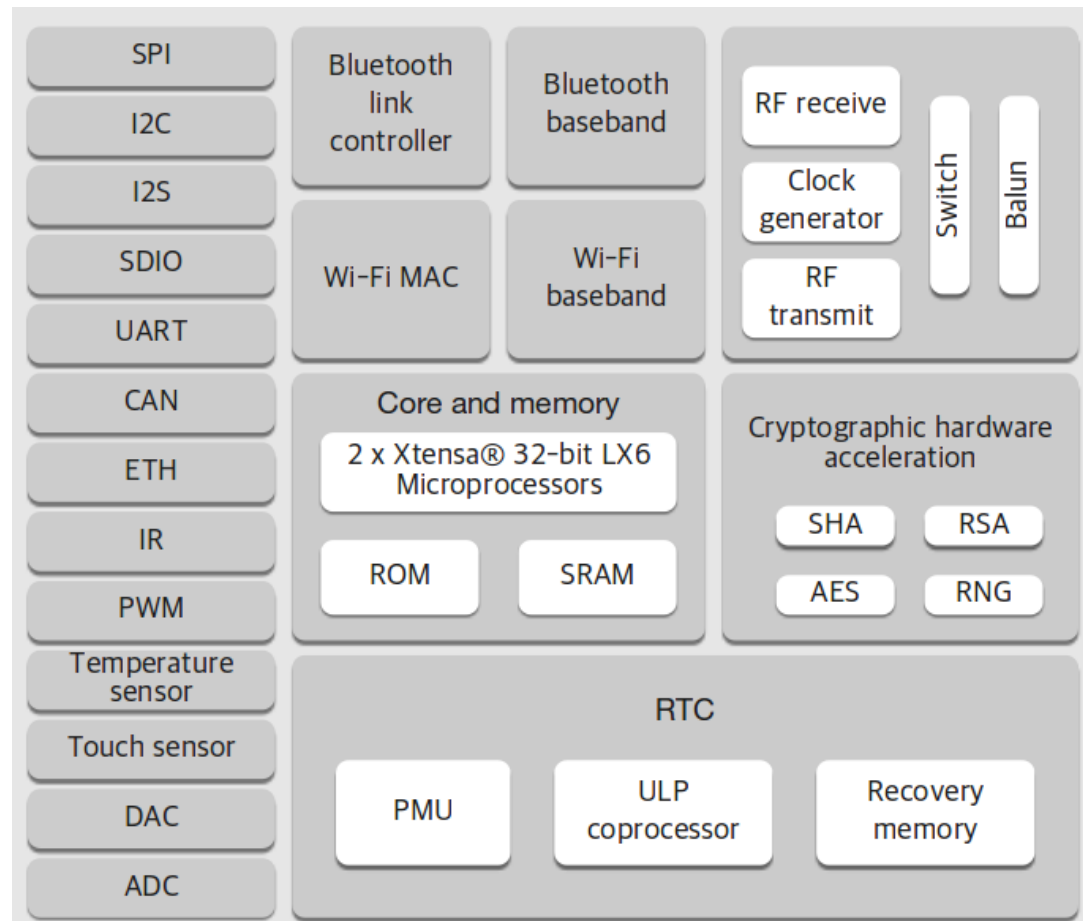
■ Power
■ GND
■ Serial Pin
■ Analog Pin
■ Control
■ Physical Pin
■ Port Pin
■ Touch Pin
■ DAC Pin
 PWM Pin

John Lee: "ESP32-wROVER is my most dependable trusty companion with its prodigious dual core 240MHz CPU+FPU, capacious 4MByte of SRAM, 16MByte of Flash and unassailable bulletproof security features. It runs with our version of multi-core freertos and the Espressif Arduino SDK; now we shall embark on this serendipitous egalitarian voyage to find the destiny of the Internet of Things Phenomenon — with great power comes great responsibility."




<http://esp32.com/>

ESP32 blokový diagram



Vývojový modul ESP32

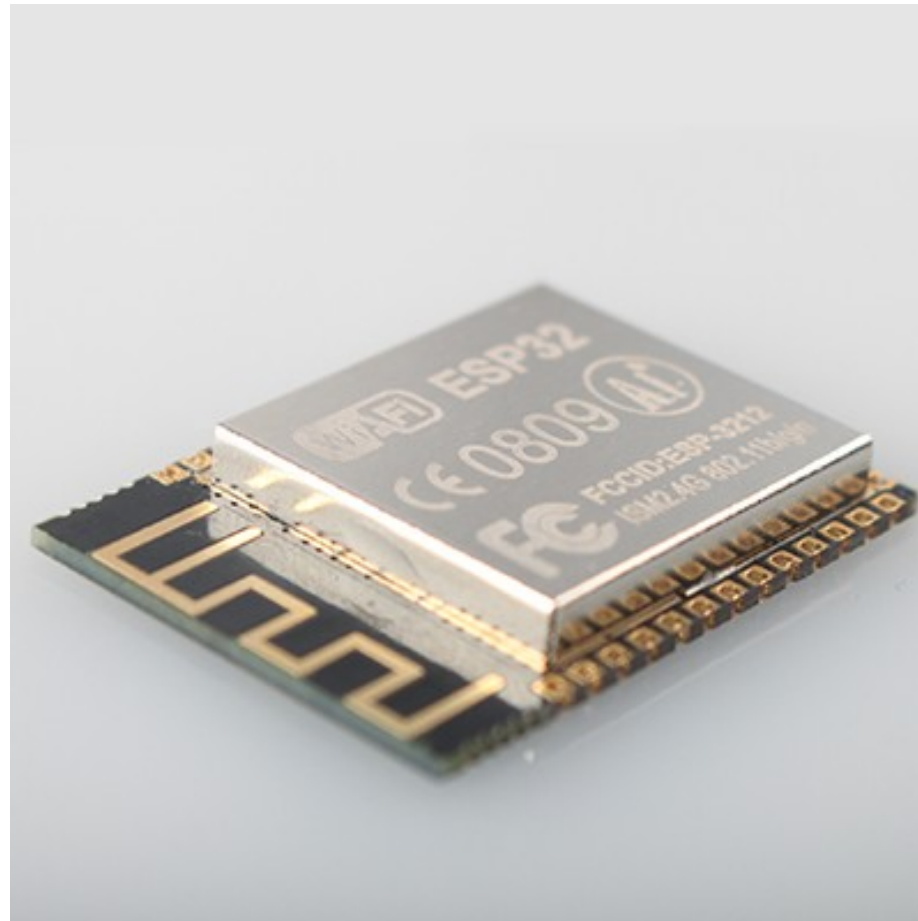
ESP32 Dev Board PINMAP

					3.3V																		
	(pu)			RESET	EN		GND																
SVP			ADC0		GPIO36		GPIO23	VSPI MOSI														SPI MOSI	
SVN			ADC3		GPIO39		GPIO22																Wire SCL
			ADC6		GPIO34		GPIO1	TX0															Serial TX
			ADC7		GPIO35		GPIO3	RX0															Serial RX
		TOUCH9	ADC4		GPIO32		GPIO21																Wire SDA
		TOUCH8	ADC5		GPIO33		GND																
DAC1			ADC18		GPIO25		GPIO19	VSPI MISO															SPI MISO
DAC2			ADC19		GPIO26		GPIO18	VSPI SCK															SPI SCK
		TOUCH7	ADC17		GPIO27		GPIO5	VSPI SS													(pu)		SPI SS
	TMS	TOUCH6	ADC16	HSPI SCK	GPIO14		GPIO17																
(pd)	TDI	TOUCH5	ADC15	HSPI MISO	GPIO12		GPIO16																
					GND		GPIO4			ADC10	TOUCH0											(pd)	
							GPIO0	BOOT		ADC11	TOUCH1												(pu)
							GPIO2			ADC12	TOUCH2												(pd)
	TCK	TOUCH4	ADC14	HSPI MOSI	GPIO13		GPIO15	HSPI SS		ADC13	TOUCH3	TDO										(pu)	
				FLASH D2	GPIO9		GPIO8	FLASH D1															
				FLASH D3	GPIO10		GPIO7	FLASH D0															
				FLASH CMD	GPIO11		GPIO6	FLASH SCK															
					5V																		

Jak ESP32 programovat?

- Primárně skrz ESP32 IoT Development Platform
- IDF (1.října verze 0.9) – GitHub espressif/esp-idf
- přehled vydání: github.com/espressif/esp-idf/releases
- Arduino core (7.října v0.0) – espressif/arduino-esp32
- MicroPython (?)
- NodeMCU Lua – esp32 branch nodemcu

~~ESP3212~~ ESP32S od Ai-Thinker



Prodejci

- Espressif prodává samotné čipy
- Ai-Thinker a další vyrábějí moduly či přímo desky
- Adafruit, AliExpress, Seeed Studio atd. je prodávají
- přehled například na www.esp32.net

ESP32 informační zdroje

- [espressif.com/products/hardware/esp32/resources](https://www.espressif.com/products/hardware/esp32/resources)
- github.com/espressif/esp-idf/tree/master/docs
- www.esp32.com fórum

Díky za pozornost

... a těším se na otázky

Petr Stehlík

www.pstehlik.cz

<https://plus.google.com/+PetrStehlik>

petr@pstehlik.cz