

Arduino a ESP8266

Představení, způsoby jejich propojení
a zamyšlení nad smyslem jejich propojování

Petr Stehlík
www.pstehlik.cz

Osnova

- žijeme v úžasné době
- mikro-představení Arduina
- trošku větší představení ESP8266
- možnosti propojení a důvody pro ně
- chytáky při propojení přes UART
- knihovny pro Arduino
- esp-link
- Lua a další skriptovací jazyky, zbude-li čas
- ESP32 jako taková pěkná tečka za tím naším případem

Žijeme v úžasné době

- Záplava jednodeskových mikropočítačů
- Každý týden další nový, rychlejší a levnější
- 32bit, 1 GHz a 1 GB nestačí? Zdvojnásobíme!
- \$35? \$25? \$19? \$9? \$5? zadarmo!
- Nenechte se svést k používání kanónů na vrabce
- Use „the right tool for the job“

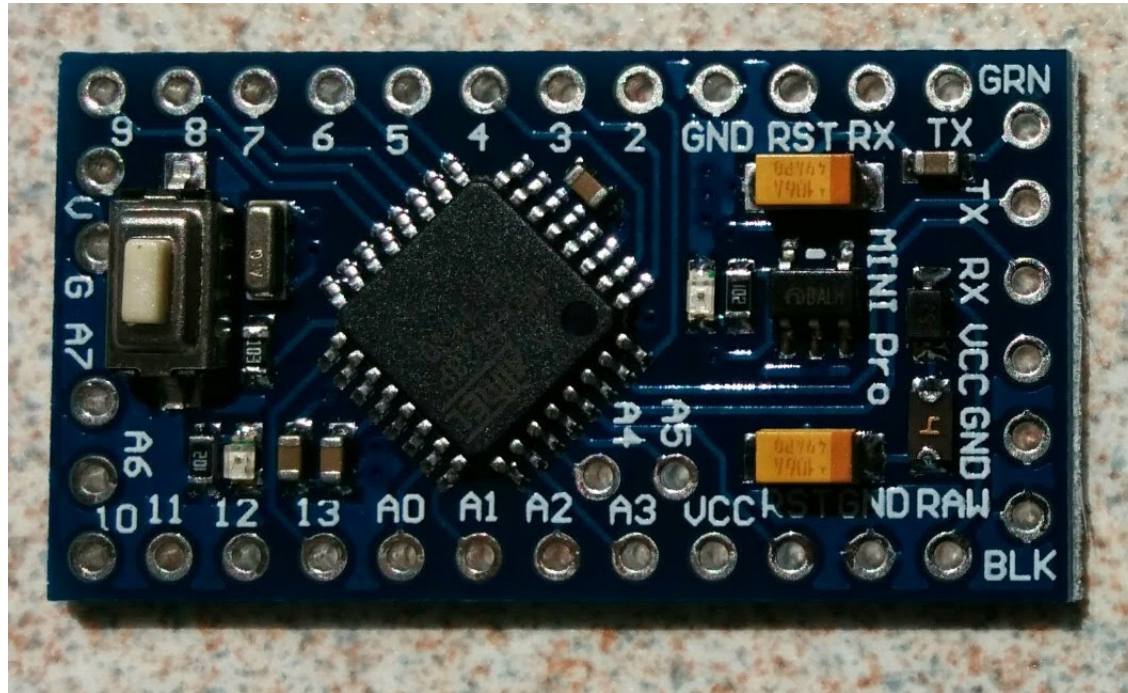
Arduino

- Arduino je otevřená (open source) elektronická platforma, založená na uživatelsky jednoduchém hardware a software
- Srdcem většiny 8bitových Arduin je ATMEGA328p
- Základní parametry: napájení 5V, frekvence CPU 16 MHz, 32 kB flash, 2 kB RAM, 22x GPIO
- Dostačuje na téměř všechny úkoly, ale chybí mu síťová konektivita

Arduino rodina



Pro Mini (Arduino kompatibilní)



ESP8266

(Gratton @ linux.conf.au 2016)

At a glance

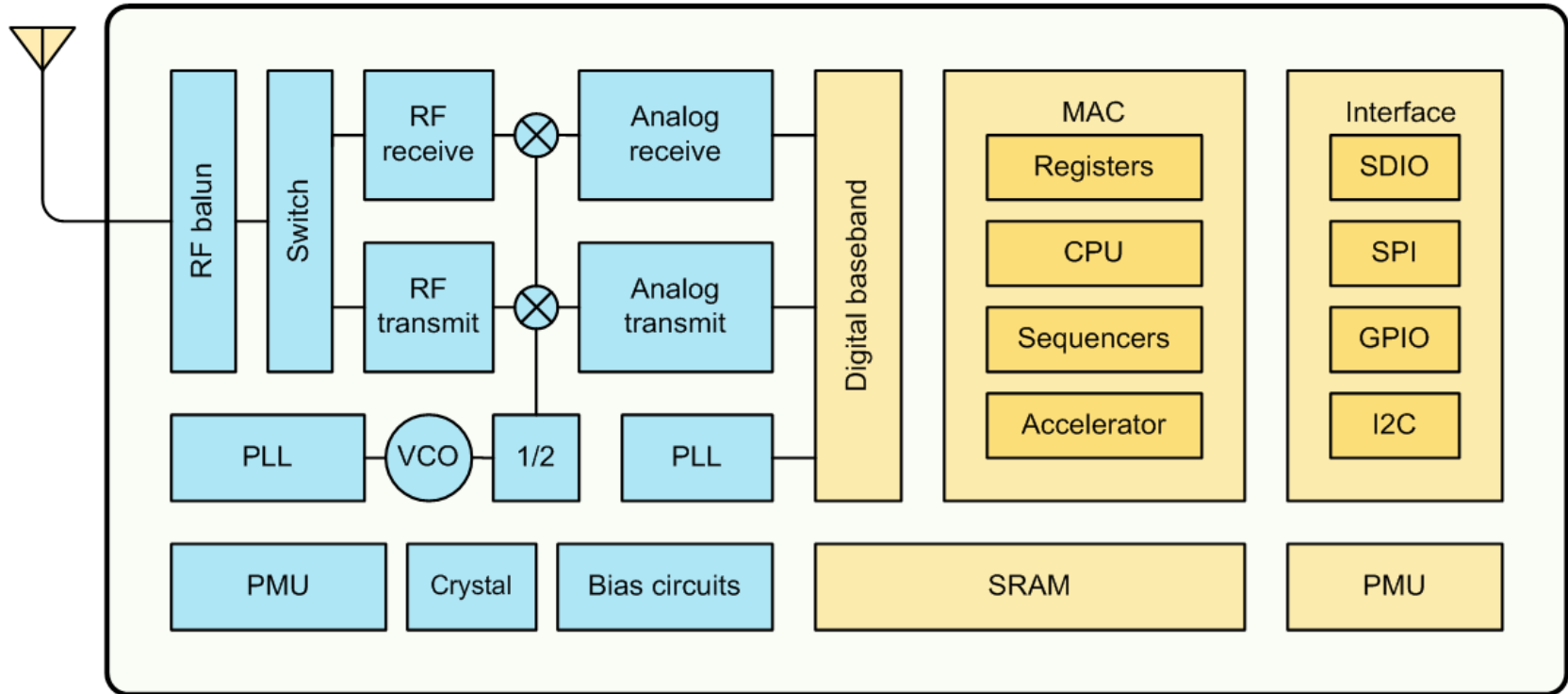
- 32-bit 80MHz Xtensa lx106 processor
- 802.11b/g/n WiFi (AP/client modes)
- 96KB data RAM
- 64KB instruction RAM
- 64KB boot ROM
- External SPI flash
- Analog to Digital Converter
- 2xSPI, 2xUART, SDIO, I2S
- Low power modes



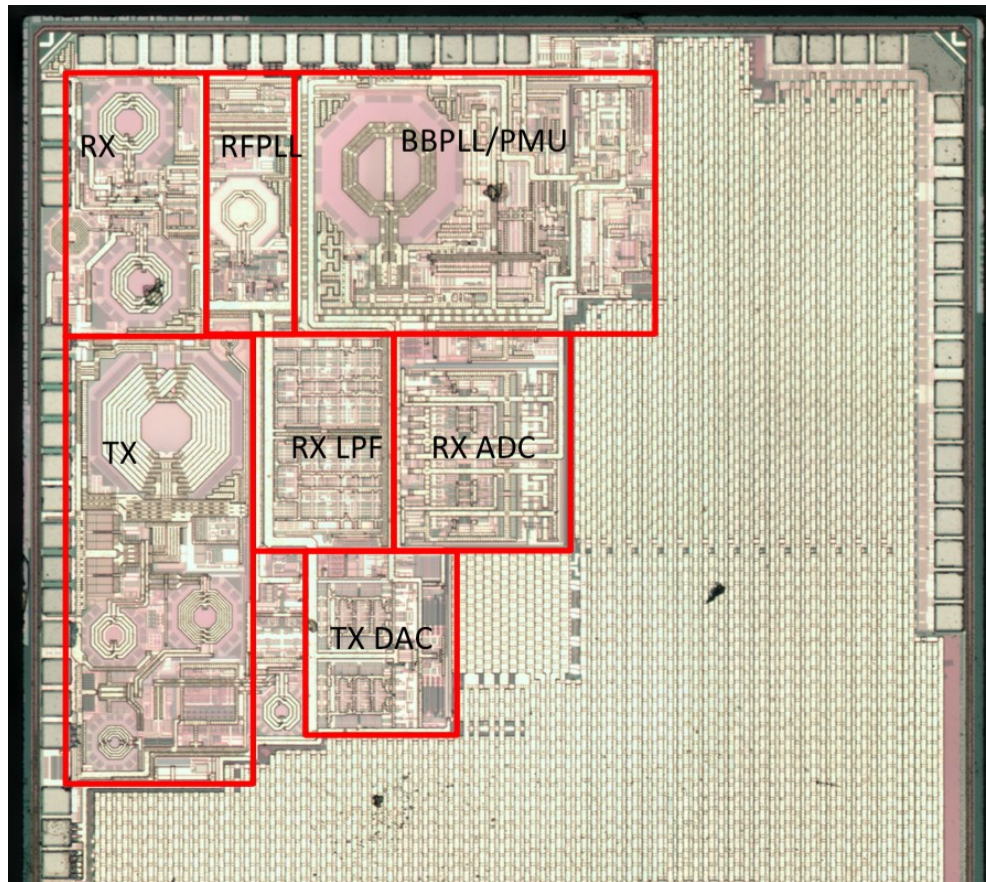
ESP8266

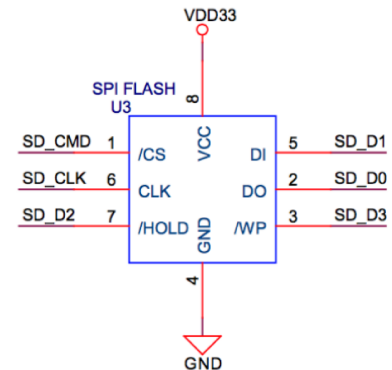
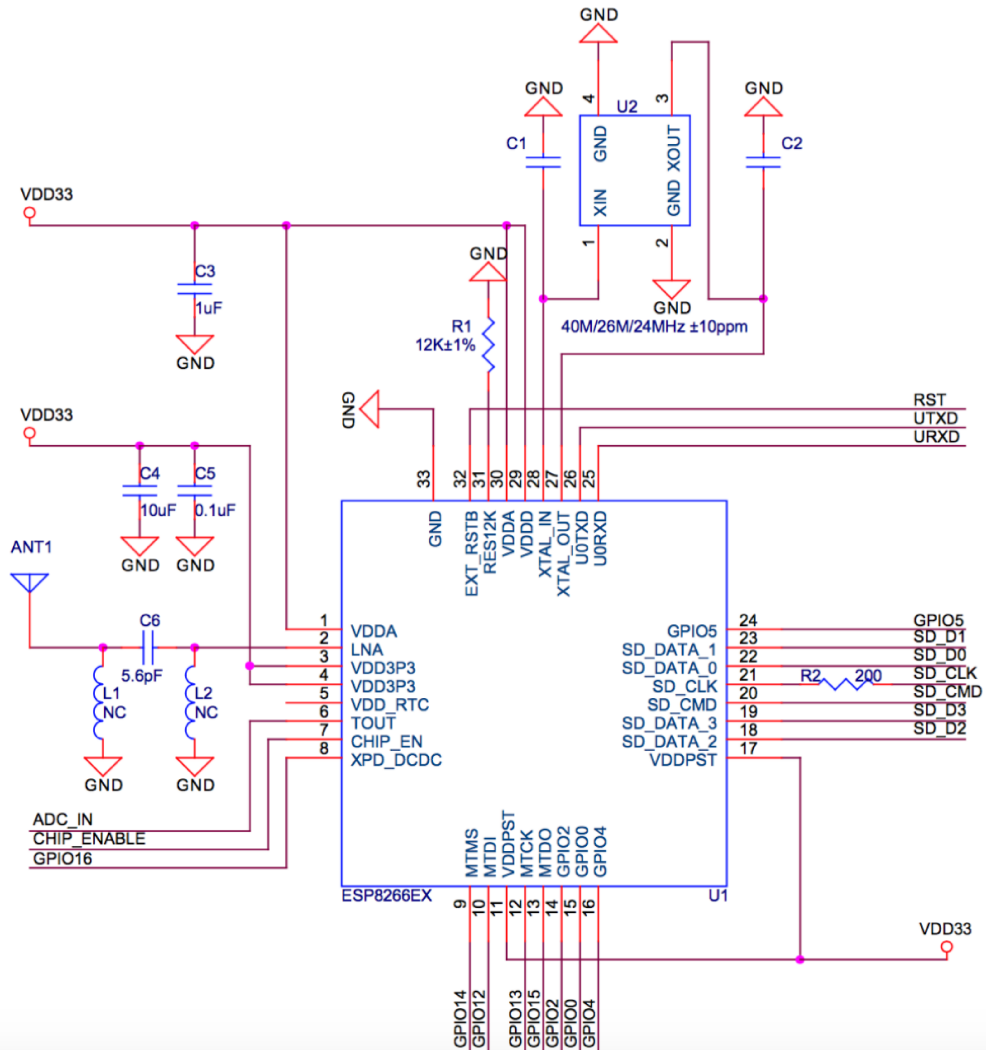
- Espressif Systems „Smart Conectivity Platform“
- kompletní WiFi SoC s minimem externích součástek
- 802.11bgn WPA2 v módu klient i AP, kompletní TCP/IP stack
- srdcem je 32bit MCU Tensilica L106 a všechno je Ultra Low Power
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, ADC
- Základní parametry: 3,3 V, 80/160 MHz, 64 kB boot ROM, 96 kB RAM (volné <36 kB), 16 (10) GPIO
- Externí flash paměť (512 kB – 16 MB) připojena přes SPI rozhraní
- Serial-WiFi konvertor (ala Bluetooth SPP) nebo samostatný počítač
- Výchozí „AT“ firmware, možnost nahrát vlastní (přes UART či OTA)
- SDK, GCC, Sming, ArduinoIDE s ESP8266 pluginem

Vnitřní schéma ESP8266



Pohled dovnitř čipu ESP8266





Funkce GPIO pinů

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	SPID		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	SPICS0		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	HSPiD 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

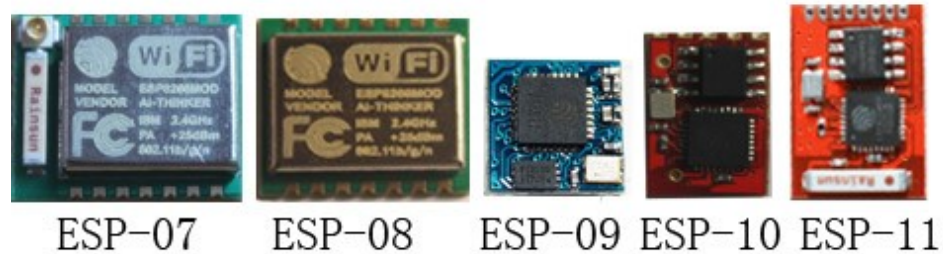
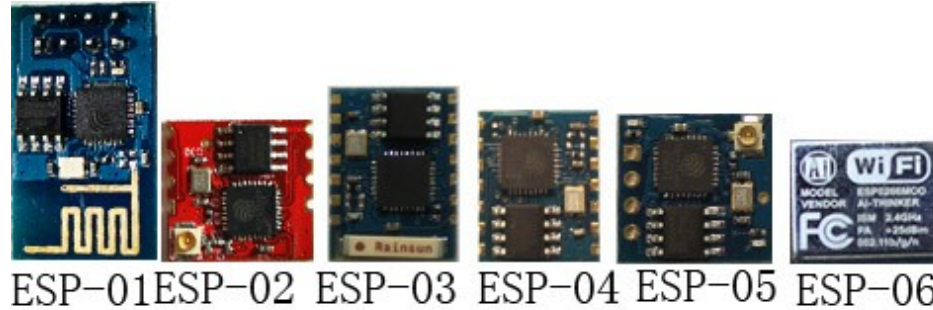
Jak rychlý je procesor v ESP8266?

- Dekódování streamovaného MP3 s výstupem na I2S – OK
- Emulování Commodore C64 s obrazovkou v prohlížeči – hmmm
- Barevné televizní vysílání v RF pásmu – WTF?
- Server pro multiplayer 3D FPS hru přes VNC – insane!

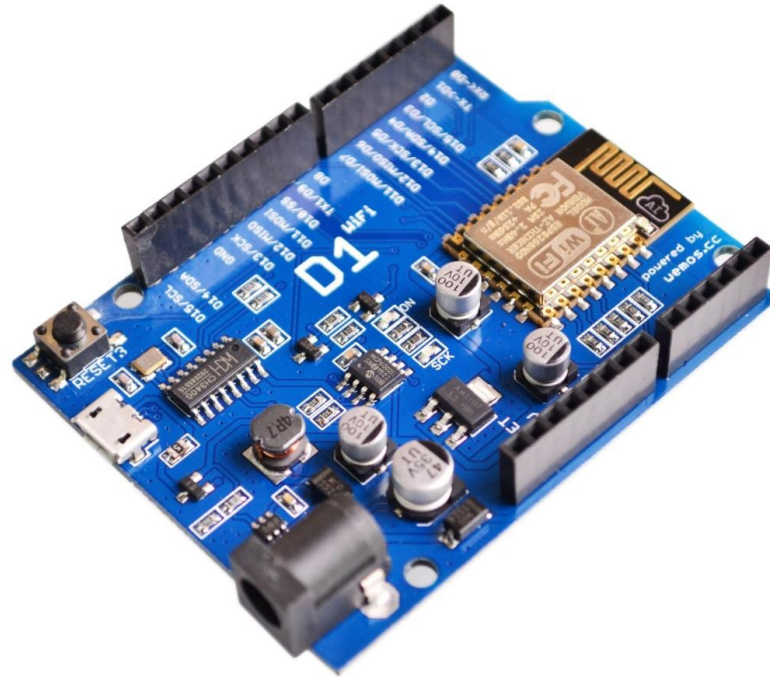
TV vysílání přes I2S na kanále 3 = 61 MHz



verze ESP8266 desek



Arduino? Téměř, ale ne úplně...



Propojení ESP nejen s Arduinem

- Proč ještě vůbec používat Arduino? Protože 8xADC, funkční PWM, dokumentace, stabilita, realtime, 5V GPIO, ...
- Propojení Aduina s ESP8266 přes UART či přes SPI
- přes UART to jde s „AT“ firmwarem rovnou (pozor na úskalí)
- přes SPI vyžaduje vlastní firmware (existuje komerční řešení)
- nejen Arduino, ale i ARM nebo malé 8bit/16bit počítače chtějí WiFi konektivitu

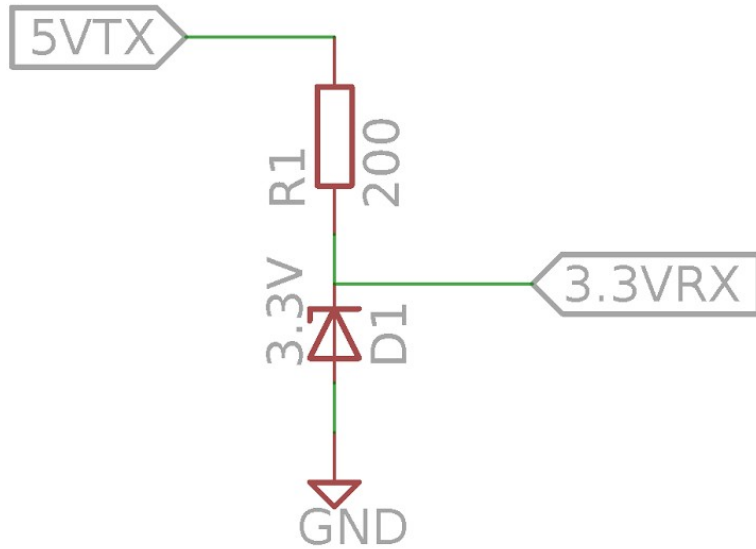
Komunikace přes UART

- Nastavit obě strany stejně: 115200 8N1
- Nastavit stejný handshaking (nic, SW Xon/Xoff, HW RTS/CTS)
- Propojit kříženě RX a TX, pozor na napětí
- Arduino speciality: HW Serial vs SoftwareSerial
- Pozor na přetečení dat

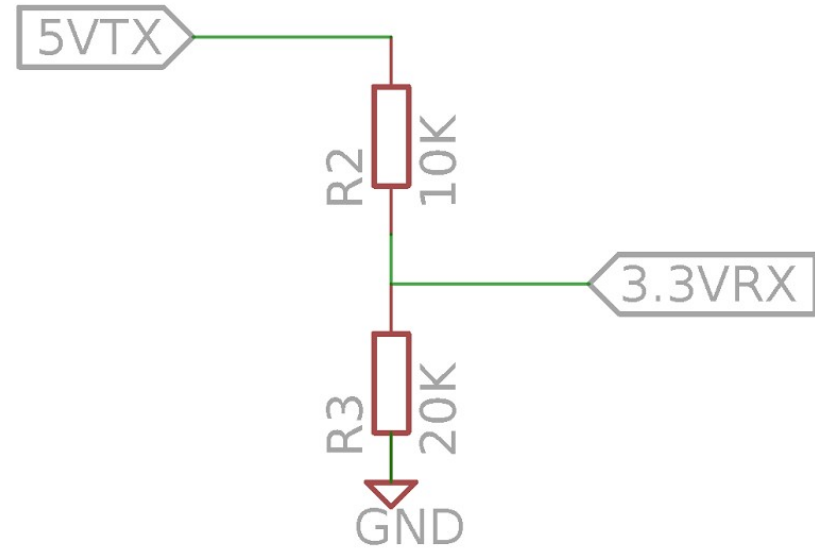
Konverze napětových úrovní

- Arduino 5V (40 mA) != ESP8266 3,3 V (12 mA)
- MOSFET
- napětový dělič ze dvou rezistorů
- Zenerova dioda
- Clamping dioda
- Dioda v závěrném směru

Převodník napět'ových úrovní

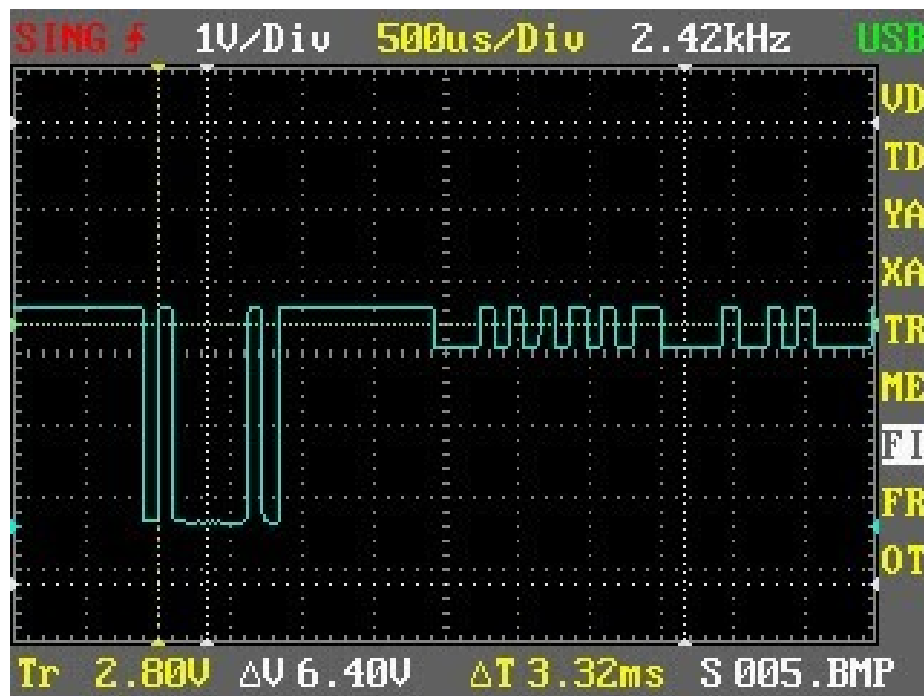


With a Zener diode

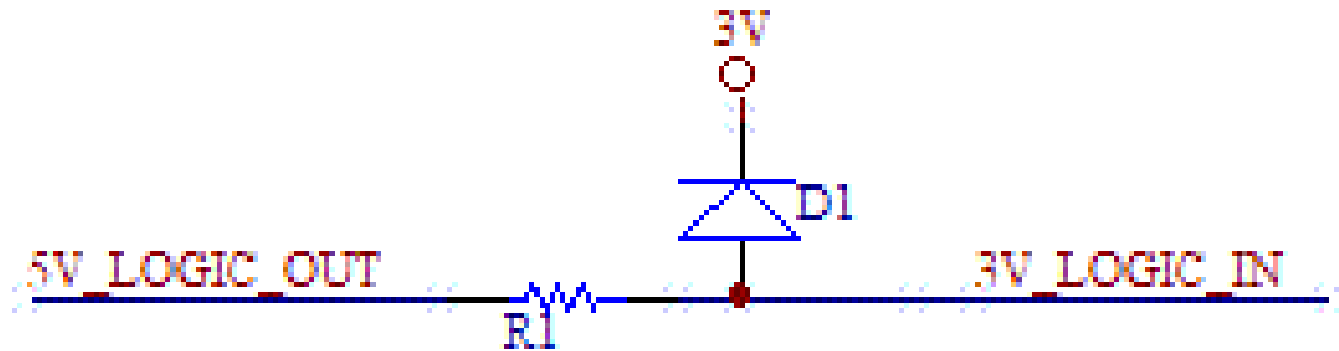


With a voltage divider

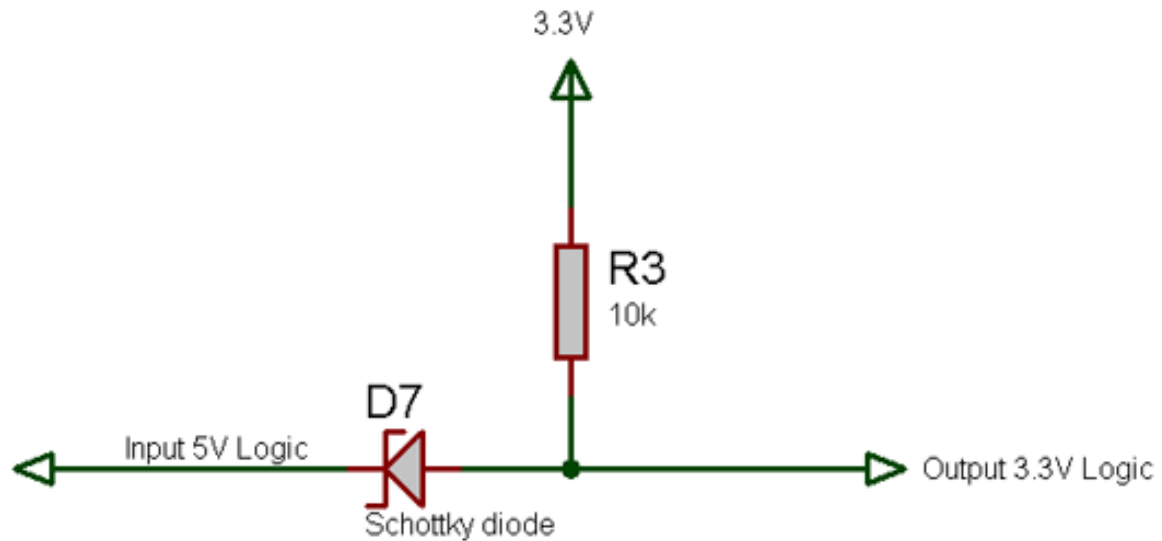
„AT“ s rezistorovým děličem



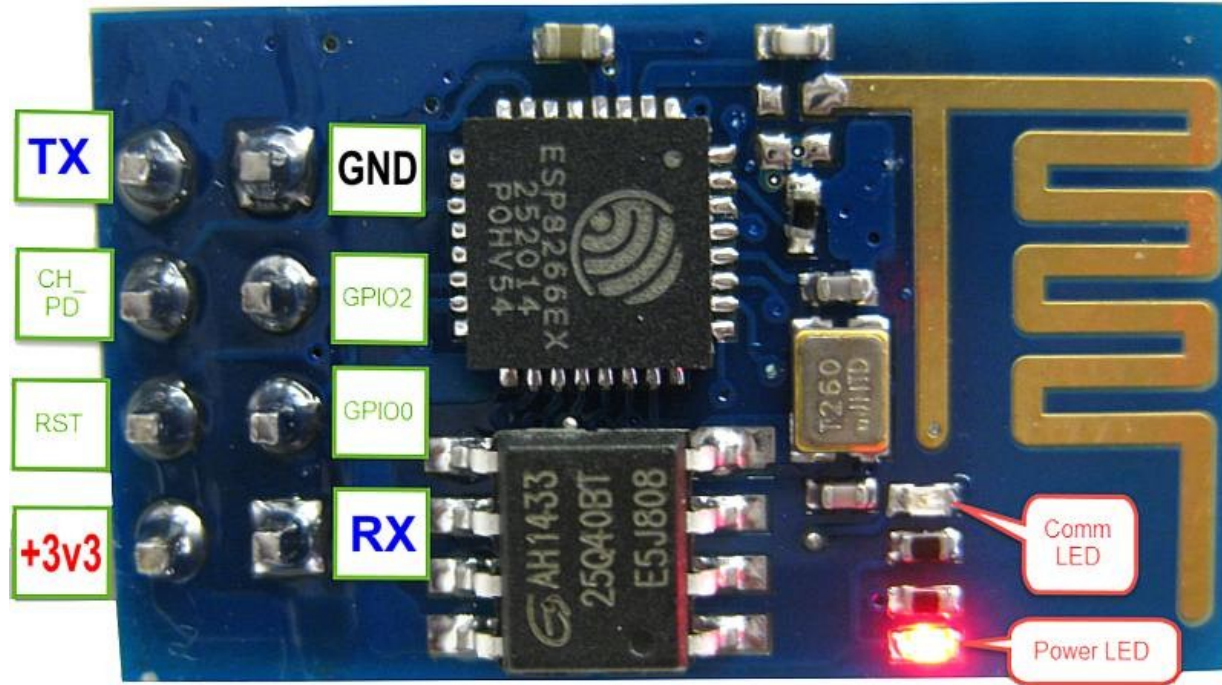
Clamping dioda



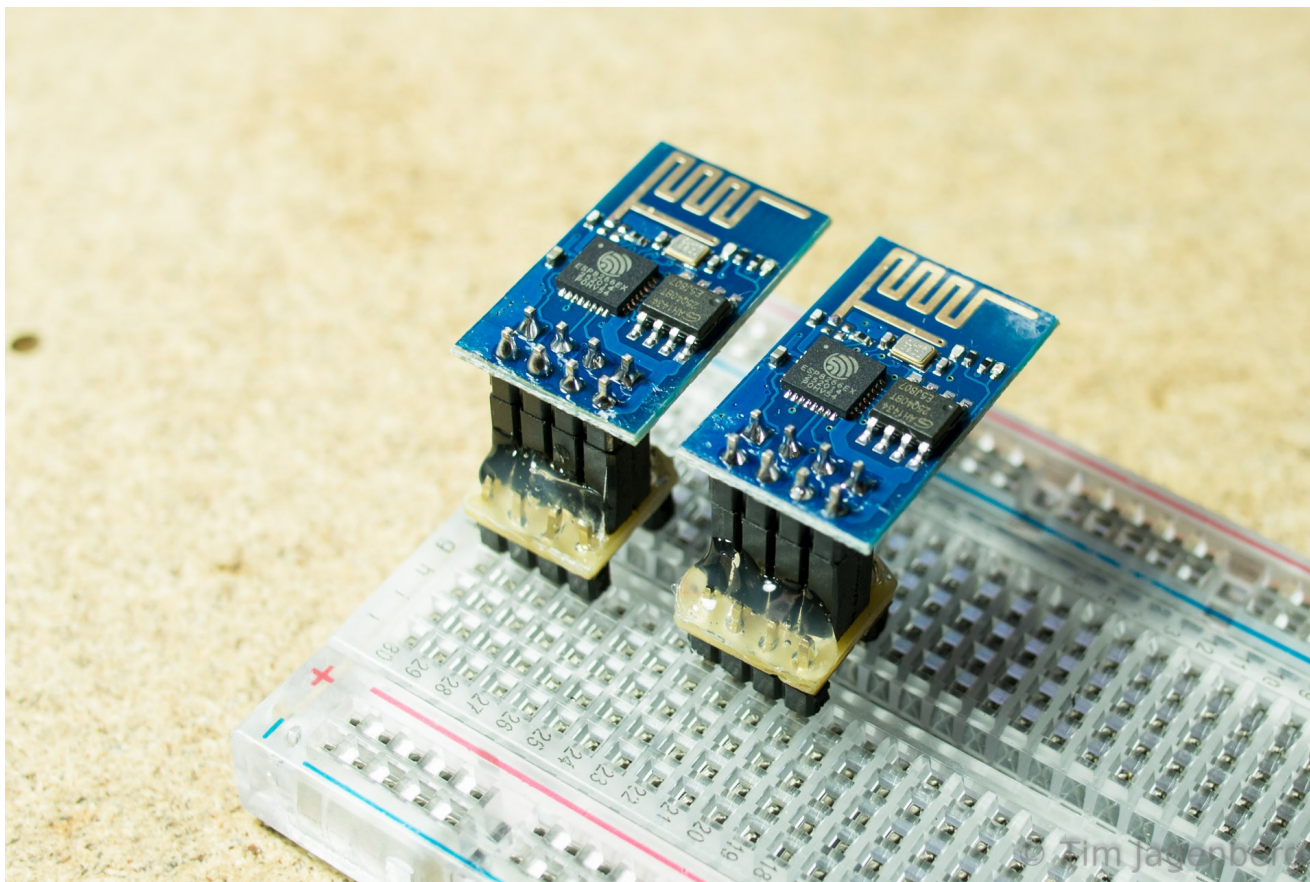
Dioda v závěrném směru



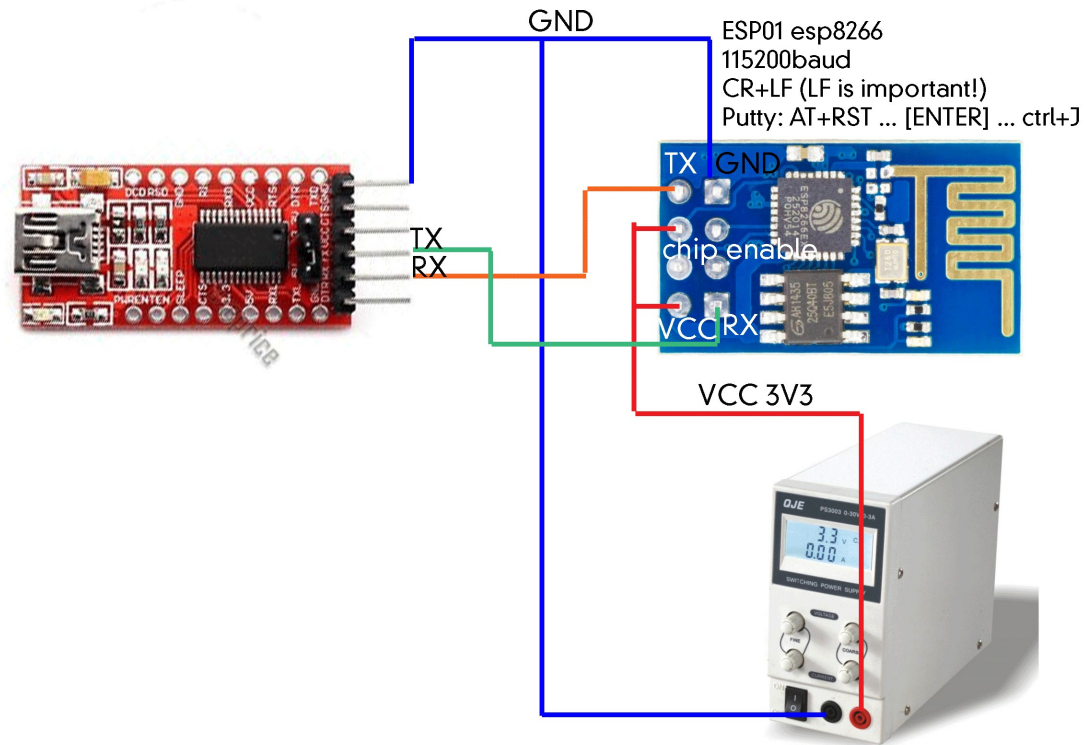
ESP8266-01



Elegantní převodník do breadboardu



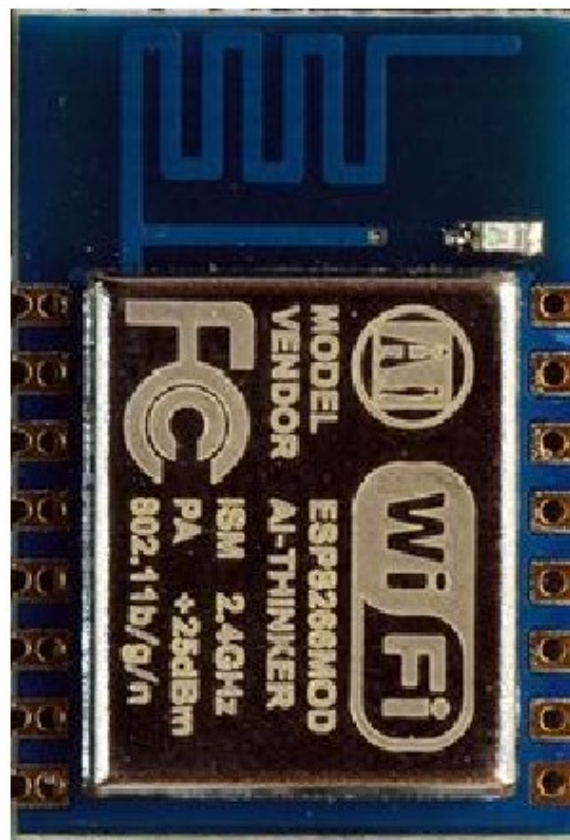
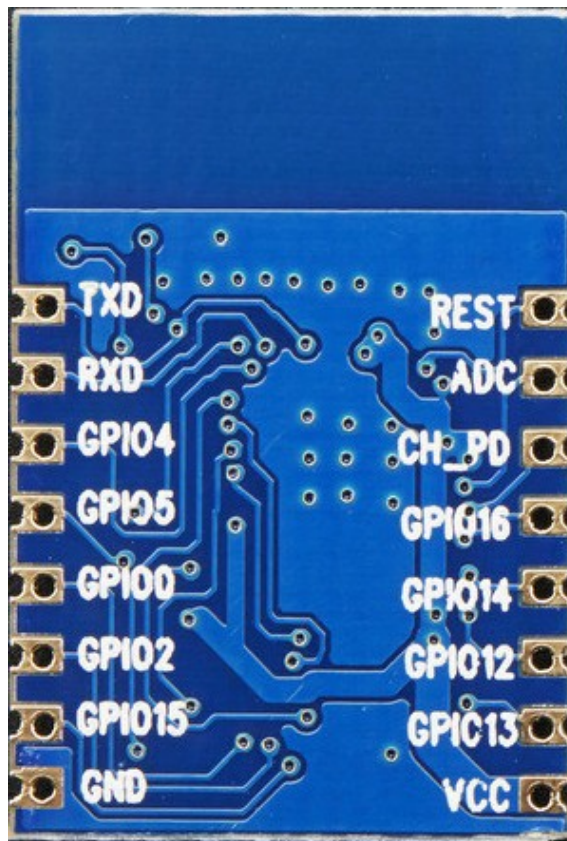
„Schéma“ propojení ESP8266 s FTDI



Jak se připojit k ESP přes UART

- VCC na stabilní zdroj 3,3 V
- CH_PD připojit k VCC
- GPIO0 a GPIO2 NEpřipojit k zemi (GPIO15 naopak uzemnit)
- RX a TX do kříže jako null-modem
- Parametry portu 115200 (nebo 9600) 8N1
- AT příkazy ukončovat CR+LF (CTRL+M CTRL+J)

ESP8266-12



ESP8266 bootovací přepínače

- IF GPIO15 není LOW THEN boot SDIO
- ELSE IF GPIO0 ani GPIO2 nejsou LOW THEN boot flash
- ELSE IF GPIO2 není LOW THEN upload UART data to flash
- V praxi chceme vždy GPIO2 NEuzemněné a GPIO15 LOW
- Vedlejší efekt: GPIO 0, 2 a 15 nejdou použít jako vstupy
- Neuzemňovat piny natvrdo (nebezpečí jejich zničení)

Spánek a probouzení

- 4 úrovně běhu: On, WakeUp, Sleep, DeepSleep, Off
- DeepSleep prý bere pouhých 10 μA (naměřeno 60+ μA)
- DeepSleep = běží pouze RTC, po timeoutu zamává GPIO16
- propojením GPIO16 s RST probudíme z DeepSleep
- nepropojovat natvrdo (nebezpečí zničení pinů)

Jak získat nový firmware

- SDK release (aktuální 1.5.2, umí SSL!)
- „flashovací“ program – esptool.py
- Flash firmware: přizemnit GPIO0 a resetnout ESP8266
- Chyták: velikost flash (minimálně 1 MB, -fs 8M)
- Chyták: NOBOOT vs BOOT firmware, SSL test ukáže

Komunikace přes AT příkazy

- Ručně – `Serial.println("AT+RST");`
- Parsování odpovědí pomocí regex knihovny
- Knihovny pro Arduino – WiFiEsp a arduino-ESP8266, úmyslně kompatibilní API s Ethernet a WiFi knihovnamí
- Chyták: obsazení RAM a flash, kompatibilita s aktuálním FW, sledování změn v SDK (různé revize AT interpretru)
- Pozor na SoftwareSerial, ztrácí či ničí data, zpomaluje, nepoužívat raději vůbec!

Pozor na přetečení dat

- Arduino HW Serial – bufer 2x64 bajtů
- ESP8266 seriál buffer – 256 bajtů?
- Příchozí komunikaci neumíme přibrzdit
- Pozor na velikost HTTP hlaviček
- Možnost přidat HW handshaking přes GPIO13 (CTS)

Příklad HTTP GET hlavičky – 411 bajtů

```
GET / HTTP/1.1
Host: 172.17.208.25:8888
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5 Build/MMB29Q)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.95 Mobile Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: cs-CZ,cs;q=0.8,en;q=0.6,sk;q=0.4
```

ESP-LINK

- Wifi-Serial Bridge s REST a MQTT
- Možnost OTA programování Arduin (či jiných přes UART)
- Vzdálené ladění – sériová konzole přímo v prohlížeči
- Arduino může volat MQTT a další fce
- Možnost i Arduino jako slave

Webové rozhraní esp-link

The screenshot displays the web interface for the esp-link firmware. It features a dark sidebar with navigation links: Home, Wifi, µC Console, and Debug log. The main content area has a yellow header with the JeeLabs logo and the text 'esp-link 0.9.5-12-gec34d9f - 2015-06-19 12:29:17 - development'. Below the header, there is a descriptive paragraph about the firmware's capabilities, a code block for programming an Arduino/AVR, and a section for configuration options. The configuration section includes a 'Wifi summary' table and a 'Pin assignment' section with radio buttons for selecting hardware configurations.

esp-link

Home
Wifi
µC Console
Debug log

JL JEE LABS
esp-link
esp-link 0.9.5-12-gec34d9f - 2015-06-19 12:29:17 - development

The JeeLabs esp-link firmware bridges the ESP8266 serial port to Wifi and can program microcontrollers over the serial port, in particular Arduinos, AVRs, and NXP's LPC800 and other ARM processors.

Program an Arduino/AVR using avrdude using a command line similar to:

```
/home/arduino-1.0.5/hardware/tools/avrdude \  
-DV -patmega328p -Pnet:esp-link.local:23 -carduino -b115200 -U \  
-C /home/arduino-1.0.5/hardware/tools/avrdude.conf flash:w:my_sketch.hex:i
```

where -Pnet:esp-link.local:23 tells avrdude to connect to port 23 of esp-link. You can substitute the IP address of your esp-link for esp-link.local if necessary.

Please refer to [the online README](#) for up-to-date help and to the forthcoming [JeeLabs blog](#) for an intro to the codebase.

Wifi summary

WiFi mode	STA
Configured network	tve-home

Pin assignment

Select one of the following signal/pin assignments to match your hardware

esp-bridge: reset:gpio12 isp:gpio13
conn led:gpio0 ser led:gpio14

Webová sériová konzole esp-link

The screenshot shows the 'esp-link' web interface. On the left is a dark sidebar with navigation links: Home, Wifi, μ C Console, and Debug log. The main content area has a yellow header with the title 'Microcontroller Console'. Below the header, a text block explains that the console shows the last 1024 characters received from UART0. A 'Reset μ C' button is followed by a 'Baud:' label and four buttons for baud rates: 57600, 115200, 230400, and 460800. The main console area is a dark brown rectangle displaying green text. It shows memory addresses 10186500 and 10187000, followed by '[hello]' and a list of memory addresses from 500 to 7000 in increments of 500.

esp-link

Home
Wifi
 μ C Console
Debug log

Microcontroller Console

The Microcontroller console shows the last 1024 characters received from UART0, to which a microcontroller is typically attached.

Reset μ C Baud: 57600 115200 230400 460800

```
10186500
10187000

[hello]
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
```

Skriptovací/interpretované jazyky

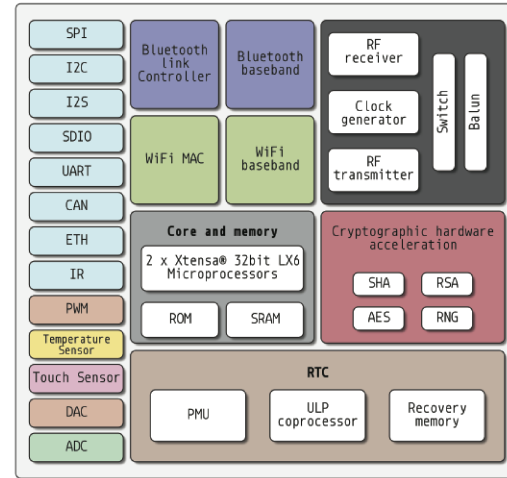
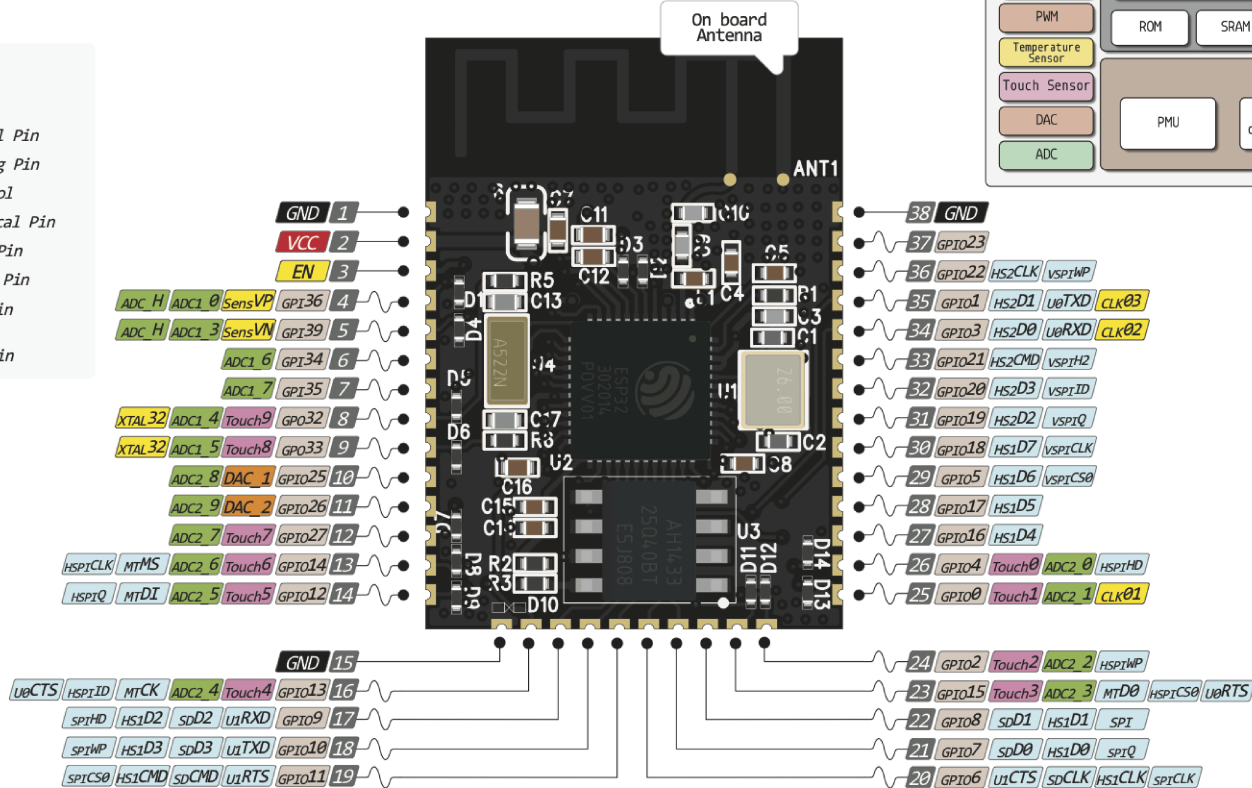
- NodeMCU Lua – dříve komerce, dnes komunitní
- MicroPython @ Kickstarter – 28 tisíc liber na vývoj SW
- BASIC - <http://www.esp8266basic.com/>
- a další

ESP32

- nový bájný nástupce ESP8266, již v dubnu (beta od prosince)
- napravuje všechny problémy/neduhy ESP8266
- single core → dual core
- 96 kB RAM → 512 kB RAM
- 10 GPIO → 33 GPIO
- Bluetooth LE 4.2 NE → ANO

ESP32 PINOUT

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



<http://esp32.com/>

Zdroje informací

- SDK/DOC/ s desítkami PDF
- bbs.espressif.com – oficiální stránka, SDK a DOC download
- www.esp8266.com - komunitní fóra
- Neilova kniha: <http://neilkolban.com/tech/esp8266/>

Kolban's
Book on
ESP8266

JANUARY 2016

Neil Kolban

Odkazy

- TV vysílání: <http://hackaday.com/2016/01/31/tv-transmitter-uses-esp8266/>
<http://hackaday.com/2016/03/01/color-tv-broadcasts-are-esp8266s-newest-trick/>
- VNC hra pylotron: <https://www.youtube.com/watch?v=oQcHhw4oTLo>
<https://github.com/cnlohr/pylotron>
- Australský reverze-engineering: <https://www.youtube.com/watch?v=AMMOrwqSPwY>
- WiFiEsp knihovna: <https://github.com/bportaluri/WiFiEsp>
- „moje“ ESP8266 knihovna: <https://github.com/joysfera/arduino-ESP8266>
- ESP-LINK: <https://github.com/jeelabs/esp-link>

Děkuji za pozornost

... a rád odpovím na vaše dotazy

www.pstehlik.cz

petr@pstehlik.cz [@joysfera](https://www.instagram.com/joysfera)