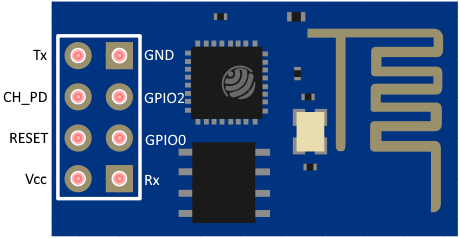
# Ako naprogramovať WI-FI čip ESP-01

Existuje veľké množstvo modulov s WI-FI čipom ESP8266, s rôznymi veľkosťami. Mňa zaujala najmenšia verzia modulu s čipom ESP8266 , a to konkrétne modul ESP-01. Tento modul je zaujímavý svojou veľkosťou a parametrami.

Opis modulu

* Veľkosť modulu približne 25x15mm
* WI-FI čip ESP8266
* veľkosť FLASH pamäte 1MB vo verzii s čiernou doskou od AI-cloud a 512kb v modrej verzii PCB, základnej manufaktúrnej verzii.
* Veľkosť sRAM pamäte 80KB
* Napájacie napätie 3,3V
* Logická úroveň pre GPIO 3,3V
* Podpora WI-FI noriem 802.11/b/g/n
* Integrovaný TCP/IP protokol stack
* Podpora WI-FI security WPA, WPA2
* 2x digitálny vstup/vystup

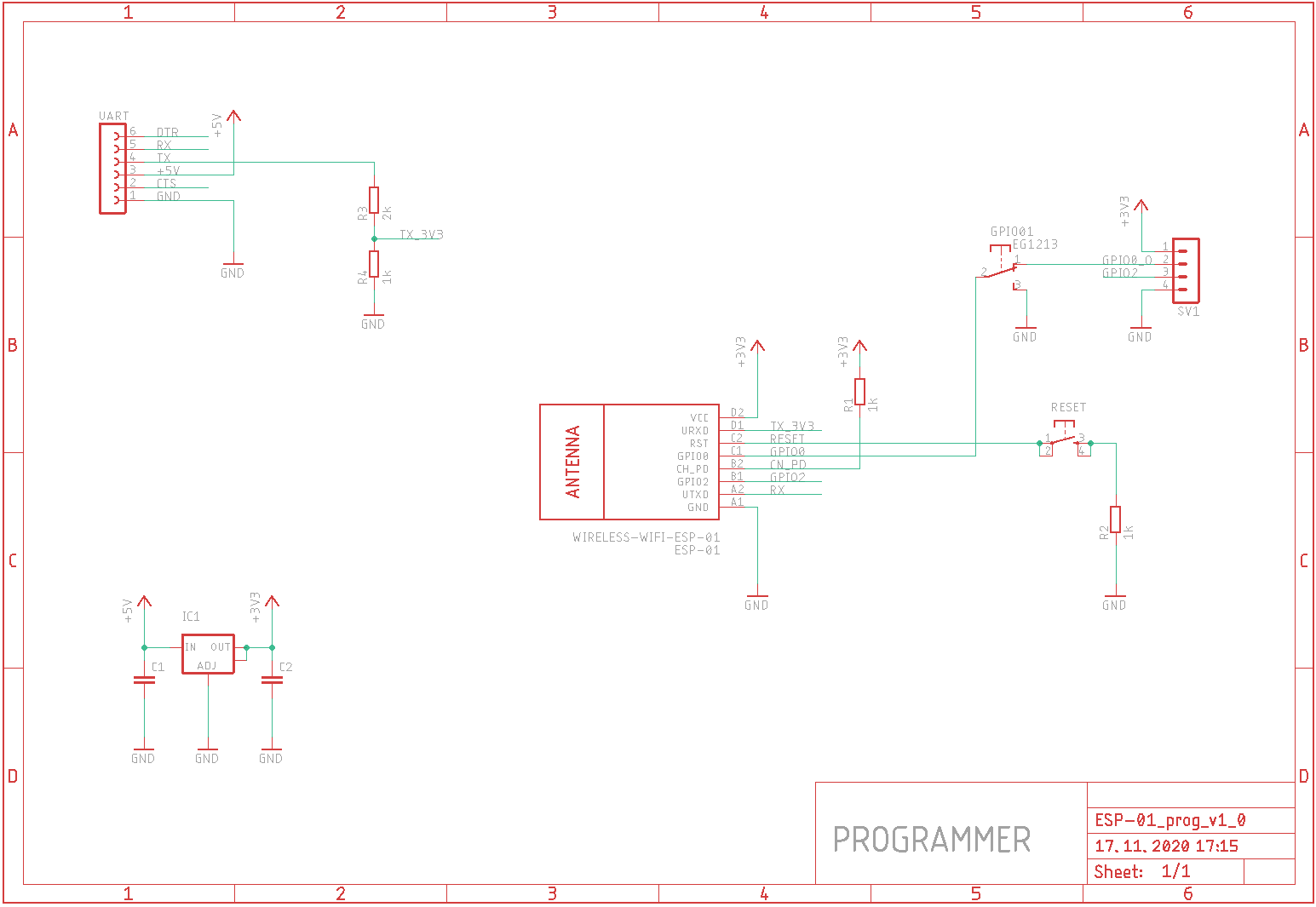


# Opis

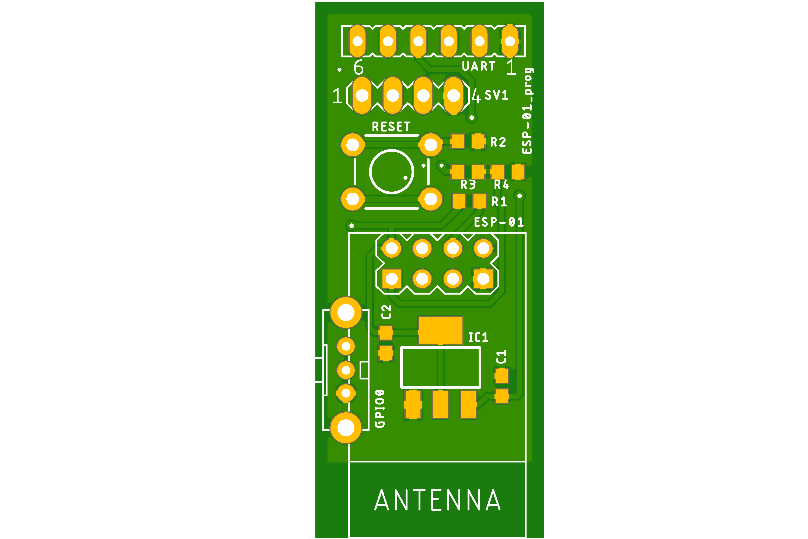
Základná verzia ESP-01 od Ai-cloud používa AT príkazy pre komunikáciu s externým MCU. V tejto verzii je už napálený firmware v ESP-01, ktorý využíva pre komunikáciu s okolitým svetom AT príkazy cez UART rozhranie. Zoznam týchto AT príkazov nájdete na internete. Tieto AT príkazy podporujú napríklad pripojenie modulu k WI-FI sieti, vytvorenie AP z WI-FI modulu, pripojenie sa na TCP-IP server, pripojenie na http server, a veľa ďalšieho. V prípade, že chcete do spomínaného **ESP-01** dostať vlastný firmware, a používať MCU **ESP-8266** ako štandardný MCU, musíte si vyrobiť vlastný programátor. Jestvuje veľa čínskych programátorov pre **ESP-01** modul, ale na tie si musíte dať pozor. Základným dôvodom je, že v čínskych e-shopoch označia za **ESP-01** programátor dosku, ktorá nie je programovacia, ale je to iba USB-UART prevodník, pre prácu s AT príkazmi. Nezabudnite si dať pozor na spotrebu prúdu pre modul. MCU ESP-8266 sa považuje za „low power WI-FI MCU“, keďže môže mať v „Deep-Sleep“ móde spotrebu okolo 10uA. No pri maximálnom vyťažení môže byť spotreba prúdu až 170mA počas komunikácie s WI-FI sieťou.

# Schéma zapojenia

Schéma zapojenia pozostáva z jedného prepínača, tlačidla, stabilizátora napätia a pár rezistorov.

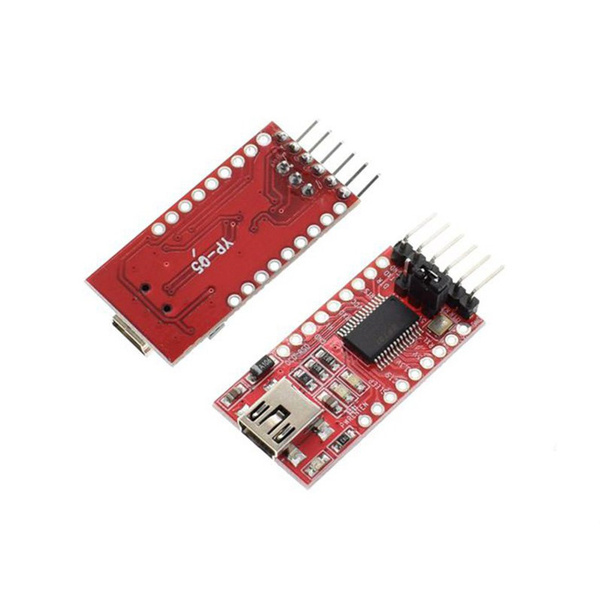


Stabilizátor napätia je potrebný, keďže **ESP-01** potrebuje napájacie napätie 3,3V. Komunikácia s **ESP-01** cez UART rozhranie tiež prebieha na napäťovej úrovni 3,3V. Pozor, štandardné Arduino komunikuje logickou úrovňou 5V, čo môže nenávratne poškodiť **ESP-01** modul. Z toho dôvodu je potrebné na RX vstup modulu **ESP-01** použiť napäťový delič, ktorý spraví z 5V približne 3,3V. Existujú špeciálne integrované obvody „**logic level convertor**“ pre upravenie napäťových úrovní, avšak v tomto prípade je postačujúci napäťový delič. Na TX výstup modulu **ESP-01** nie je potrebné pridávať nič, keďže 3,3V je stále v tolerancii. Ďalej je použitý jeden prepínač, ktorým určujeme či budeme modul programovať, alebo s ním komunikovať, a taktiež tlačidlo  **RESET**.

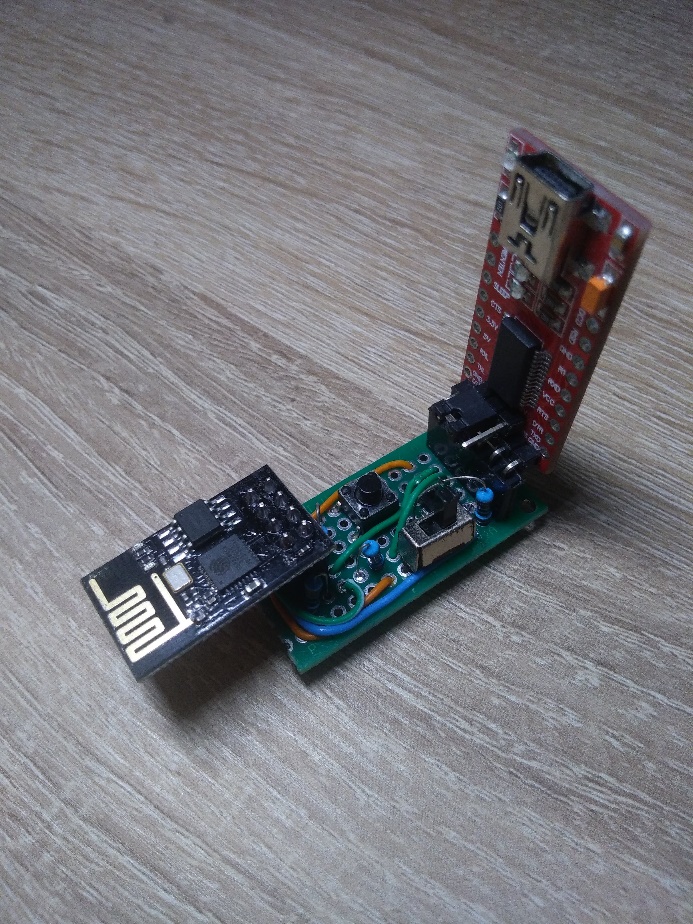


# Konštrukcia

Pre komunikáciu s programátorom, použijeme štandardný čínsky USB to UART prevodník, s možnosťou nastavenia logickej napäťovej úrovne. Možnosť nastavenia logickej úrovne v tomto prípade nevyužijeme. Tento prevodník využíva **FT232RL** čip, ale nie je vždy dodaný s originálnym čipom.



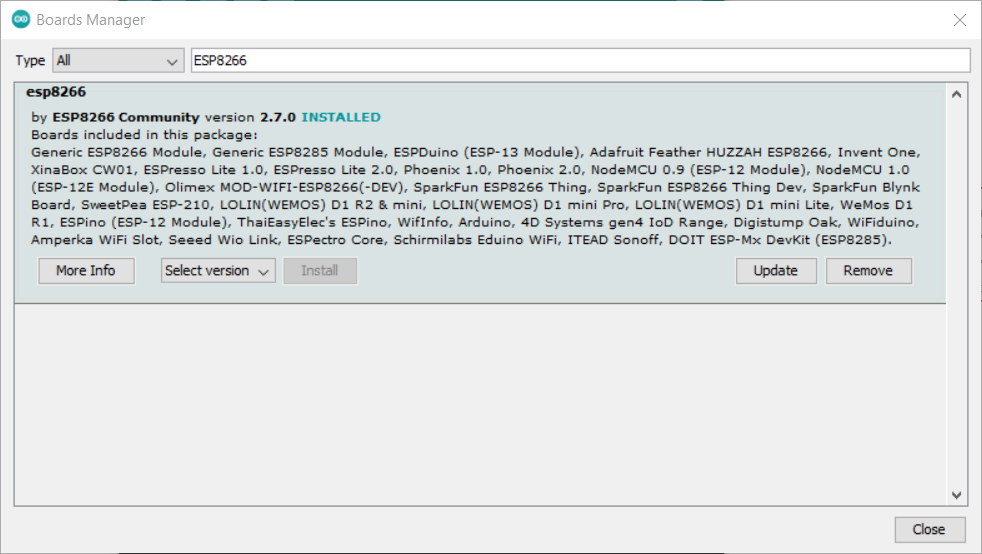
Aktuálne nemám vyrobený plošný spoj, takže som si konštrukciu osadil a otestoval na univerzálnom spájkovacom poli. Výsledný efekt so zapojeným prevodníkom a WI-FI modulom je nasledovný:



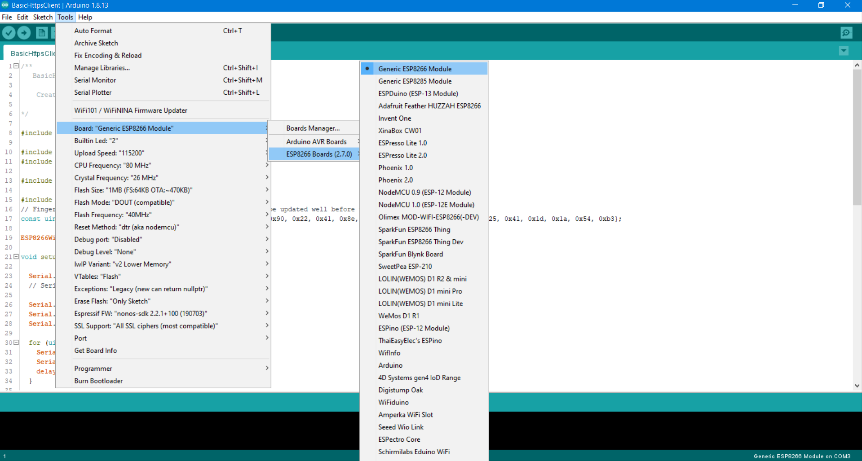
# Použitie

USB to UART prevodník nastavíme na logickú úroveň 5V. Vďaka tomu budeme mať na výstupe napájacie napätie z prevodníka 5V, a aj komunikáčnú logickú úroveň na 5V.

Ďalej si doinštalujeme podporu **ESP8266** MCU do **Arduino IDE**. Nájdeme to v časti „**tools-> Board -> Board manager“** kde do vyhľadávača napíšeme **ESP8266** a následne si nainštalujeme podporu **ESP8266** modulov.

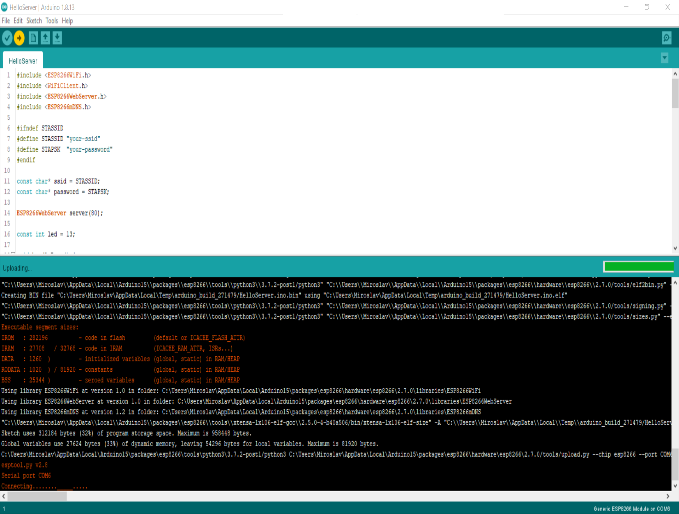


Následne si v časti **„tools->board“** vyberieme dosku „**Generic ESP8266**“

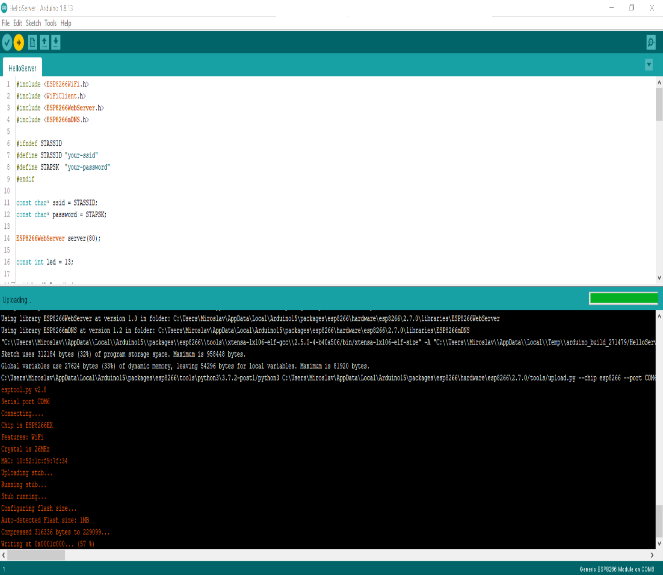


Môžeme si napríklad otvoriť example pre http WEB server, z  „**file->example->ESP8266WebServer->HelloServer“**. Ostáva už iba upraviť prihlasovacie údaje pre vašu WI-FI sieť, skompilovať zdrojový kód a nahrať ho do modulu **ESP-01**.

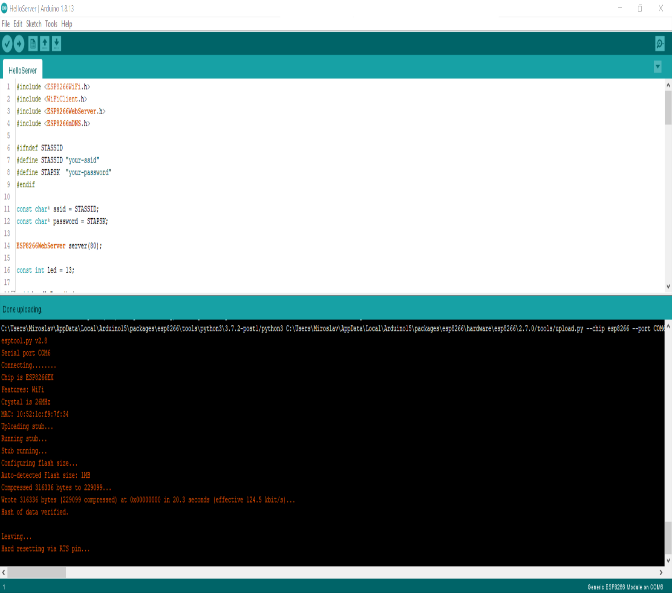
Pred tým ako spustíme nahrávanie zdrojového kódu do modulu **ESP-01**, musíme prepnúť prepínač **GPIO01** do polohy, kedy bude uzemnený **GPIO01** pin. Potom stlačíme tlačidlo v **Arduino IDE** pre začatie kompilovania a nahrávania zdrojového kódu do MCU. Po kompilácii uvidíme v konzole, ako sa **Arduino IDE** pokúša spojiť s modulom **ESP-01**.



V poslednom riadku nasledujúceho okna si môžete všimnúť text „**Connecting**“ a ako pribúdajú bodky a podtržníky. Vtedy stlačíme tlačidlo **RESET**, a mali by sme vidieť ako sa spustilo nahrávanie zdrojového kódu do modulu **ESP-01**.



Na obrázku vidíme, že prebieha nahrávanie zdrojového kódu. Po nahratí zdrojového kódu do modulu **ESP-01** uvidíme hlášku „**Hard reseting via RTS pin...**“



Potom prepneme prepínač **GPIO01** do polohy, kedy **GPIO01** pin nie je spojený so zemou a stlačíme tlačidlo **RESET**, pre reset modulu **ESP-01**. Po tejto sekvencii nabootuje modul **ESP-01** s naším zdrojovým kódom, a je možné pozerať výstup sériovej konzoly napríklad v monitore cez **Arduino IDE**.

Aktuálny programátor využíva externý USB to UART prevodník a prepínač s tlačidlom pre programovanie. Pracujem na vývoji PCB, kde bude USB to UART prevodník už na doske, kde táto doska bude používať automaticky reset MCU a nastavenie pinov pre programovanie MCU **ESP8266**.

V prípade, ak by ste mali záujem o plošné spoje programátora ma môžete kontaktovať.

# Odkazy

Zdrojový kód, potrebné knižnice, návrh PCB, schému a všetky potrebné podklady nájdete v mojom git repozitári.

# Kontakt

miroslav.pivovarsky@gmail.com

Ďakujem za prečítanie článku, dúfam že vás konštrukcia zaujala.