



WiFi na Arduino Uno

ESP8266



Zašto dodati WiFi?

- Povezivanje na mrežu je dobro za svaki računarski sistem. To otvara puno novih mogućnosti.
- Mogu se preuzimati podaci iz cijelog WWW
- Mogu se slati podaci u oblak ili bilo koji server za smještanje podataka i obradu.




Kako?

- Potreban je komad spolješnjeg hardvera, koji je u stanju konvertovati WiFi podatke u format podataka razumljiv često korištenim mikrokontrolerima (na primjer: UART, SPI, I²C, itd).
- Dodati hardver mora podržati kompletan mrežni protokol.



Varijante!

- ▶ Ima ih više.
- ▶ Skoro svaki SoC proizvođač ima svoj WiFi SoC.
- ▶ Mi ćemo upotrijebiti jedan, koji je lako za nabaviti, niske cijene, jednostavan za upotrebu, podržan od strane brojnih korisnika.

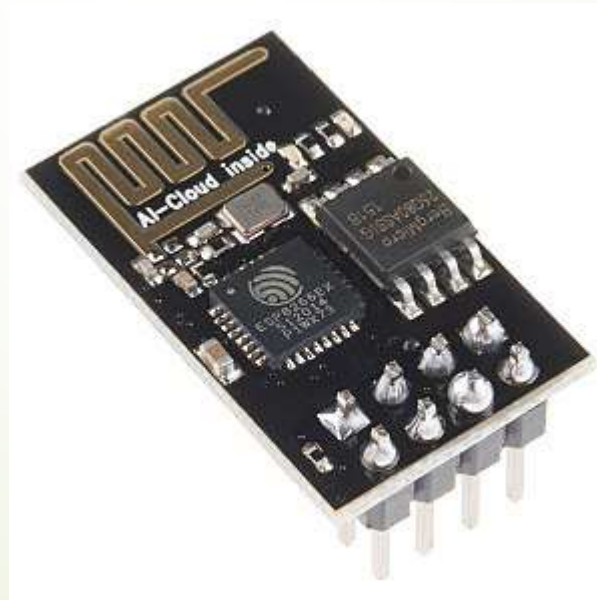


ESP8266

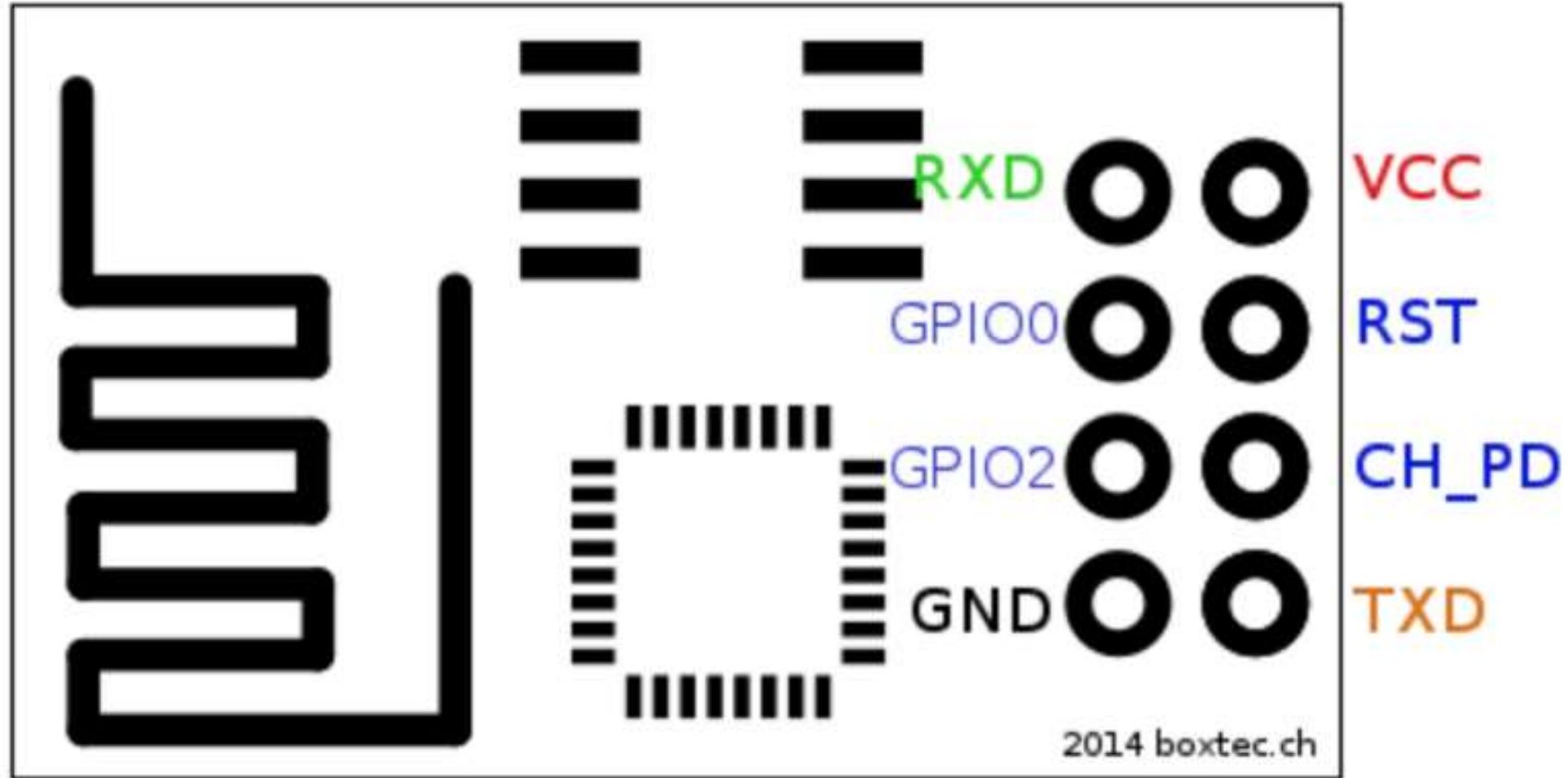
- Cijena je ne više od 5\$.
- Može se programirati kao bilo koja druga MCU.
- Posjeduje GPIO, SPI, I2C, UART, ...
- Podržava 802.11 b/g/n.
- Integrisan TCP/IP.
- Dizajniran i distribuiran od strane kineske kompanije Espressif Systems.
- Različiti modeli su raspoloživi na tržištu.

ESP8266

- ▶ Mi ćemo koristiti ESP-01 modul
- ▶ Modul omogućava povezivanje bilo kojeg mikrokontrolerskog dizajna na Internet, jednostavno upotrebom serijske komunikacije, odnosno UART interfejsa.
- ▶ Dodavanjem ovog modula na Arduino Uno otvaraju se mogućnosti za nove vrlo zanimljive projekte.



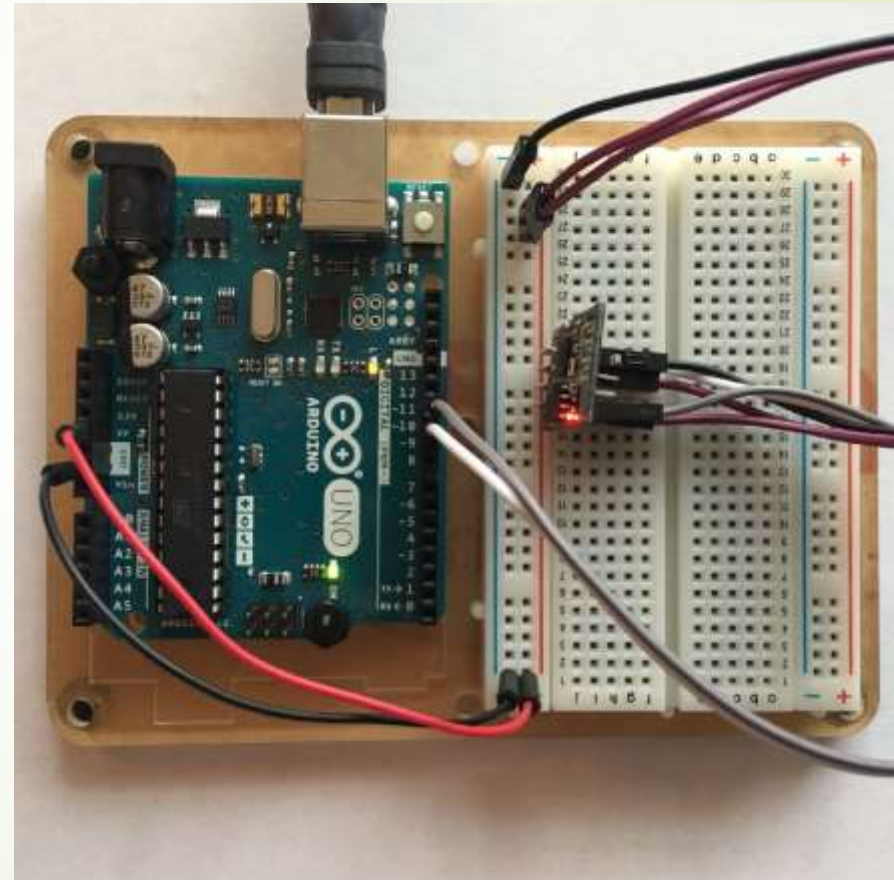
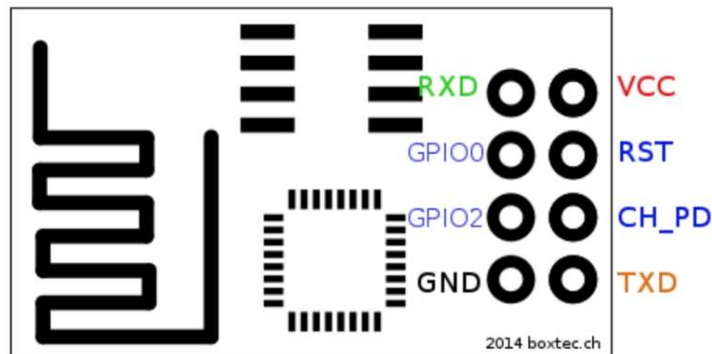
Priključci ESP-01 modula



Povezivanje ESP8266 sa Arduino Uno u cilju pristupanja iz Arduino Uno koda

Esp8266 | Arduino

| | | |
|--------|--|-------------|
| RX | | A3 (TX) |
| TX | | A2 (RX) |
| GND | | GND |
| VCC | | 3.3v |
| CH_PD | | 3.3v |
| RST | | A1 |
| GPIO 0 | | None (same) |
| GPIO 2 | | None (same) |



AT komande

Komande podržavaju do 4 varijante.

| Varijanta | Primjer | Opis |
|--------------|------------------|---|
| Test | AT+CIPSTART=? | Ispitivanje ranga vrijednosti (Do sada samo AT+CWMODE=? Podržava ovo) |
| Upit | AT+CMD? | Vraća trenutnu vrijednost parametra. |
| Postavljanje | AT+CMD=Parameter | Postavlja korisnički definisanu vrijednost parametra I pokreće. |
| Izvršavanje | AT+CMD | Pokreće komandu bez definisanja parametara. |

Detaljnije o komandama na linku:

<https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/#AT+CWSAP>

AT komande

- Sa modulom se komunicira AT komandama

Modul podržava sljedeće AT komande

Basic

AT
AT+RST
AT+GMR
AT+GSLP
ATE

WiFi layer

AT+CWMODE
AT+CWJAP
AT+CWLAP
AT+CWQAP
AT+CWSAP
AT+CWLIF
AT+CWDHCP
AT+CIPSTAMAC
AT+CIPAPMAC
AT+CIPSTA
AT+CIPAP
AT+CIOBAUD

TCPIP Layer

AT+CIPSTATUS
AT+CIPSTART
AT+CIPSEND
AT+CIPCLOSE
AT+CIFSR
AT+CIPMUX
AT+CIPSERVER
AT+CIPMODE
AT+CIPSTO
AT+CIUPDATE
+IPD

ThingSpeak

- IoT analitička platforma koja omogućuje prikupljanje, vizuelizaciju i analizu podataka uživo.
- ThingSpeak omogućuje inženjerima i naučnicima da naprave prototip i IoT sistem bez postavljanja servera i razvijanja web softvera.
- Otvorena IoT platforma sa MATLAB analitikom
- Nalazi se na adresi: [IoT Analytics - ThingSpeak Internet of Things](#)

ThingSpeak™ Channels Apps Support Commercial Use How to Buy

ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

[Get Started For Free](#) [Learn More](#)

ThingSpeak prijavljivanje

A screenshot of the MathWorks login page. The page features the MathWorks logo at the top left. Below the logo, there is a section for 'Email' with an empty input field. Underneath the input field, the text reads 'No account? [Create one!](#)'. Below this, it says 'By signing in, you agree to our privacy policy.' At the bottom right of the form, there is a blue 'Next' button.A screenshot of the 'Create MathWorks Account' form. The form has several fields: 'Email Address' (with a red error message 'Missing required information'), 'Location' (a dropdown menu showing 'Montenegro'), 'First Name', and 'Last Name'. At the bottom of the form, there are two buttons: a blue 'Continue' button and a white 'Cancel' button.

Kreiraj račun na ThingSpeak <https://thingspeak.com/>

HTTP – za komuniciranje sa ThingsSpeak

- ▶ HTTP – Hypertext Transfer protokol
 - ▶ Dizajniran da omogući komunikaciju između servera i klijenta
 - ▶ Protokol zahtjeva i odgovora
 - ▶ Klijent šalje HTTP zahjev serveru – server klijentu uzvrća odgovor
 - ▶ Odgovor sadrži status izvršenja zahtjeva, a može sadržati i dodatne podatke.
-
- ▶ U radu sa ThingSpeak platformom Arduino uređaj će imati ulogu klijenta a ThingSpeak platforma ulogu servera.



HTTP zahtjev

HTTP zahtjev generiše klijent, prema imanovanom host-u, lociranom na serveru.

Cilj zahtjeva je pristup resursu na serveru.

Korektno sastavljen HTTP zahtjev sadrži sljedeće elemente:

- ▶ Liniju zahtjeva;
- ▶ HTTP zaglavlja;
- ▶ Tijelo poruke, ako je potrebno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon poslednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP zahtjev – Linija zahtjeva

Linija zaglavlja je prva linija u poruci zahtjeva. Sastoji se iz tri dijela:

- Metod. Metod je jedno-rječna komanda koja govori serveru što da radi sa resursom. Na primjer, server može biti upitan da pošelje resurs klijentu.
- Komponenta staze URL-a za zahtjev. Staza identifikuje resurs na serveru.
- Broj HTTP verzije, ukazuje na HTTP specifikaciju s kojom je klijent pokušao uskladiti poruku.

Primjer linije zahtjeva:

GET /software/htp/cic/indeks.html HTTP/1.1

Linija zahtjeva može sadržati i dodatne podatke.

HTTP zahtjev – Zaglavlje (Header)

- ▶ Pruža prijemnoj strani informacije o poruci, pošiljaocu i načinu na koji pošiljaoc želi da komunicira sa primaocem.
- ▶ Svako HTTP zaglavlje se sastoji od imena i vrijednosti.
- ▶ HTTP protokol definiše standardni set HTTP zaglavlja i opisuje kako ih koristiti korektno.
- ▶ HTTP zaglavlje zahtjeva klijenta sadrži informacije koje server može upotrijebiti u odlučivanju kako da odgovori na zahtjev. To može biti da klijent čita zahtijevani dokument na francuskom ili njemačkom jeziku i da dokument treba biti poslat jedino ako je mijenjan od naznačenog datuma.

```
Accept-Language: fr, de  
If-Modified-Since: Fri, 10 Dec 2004 11:22:13 GMT
```




HTTP zahtjev – Tijelo poruke

- ▶ Može se nazvati i tijelom zahtjeva
- ▶ Aktuelni sadržaj poruke.
- ▶ Tijelo poruke može biti u originalnom obliku ili može biti kodirano.
- ▶ Može se nazvati i tijelom zahtjeva
- ▶ Prikladno je za neke metode zahtjeva, dok za druge nije.
- ▶ Na primjer, zahtjev sa POST metodom, koji šalje ulazne podatke serveru, ima tijelo poruke, koje sadrži te podatke.
- ▶ Zahtjev sa GET metodom, koji od servera traži da pošalje resurs, ne sadrži tijelo poruke.



HTTP odgovor

- ▶ HTTP odgovor generiše server i šalje klijentu.
- ▶ Cilj odgovora je da obezbijedi klijentu treženi resurs ili da ga informiše o izvršenju zahtjeva ili da dojavu da je došlo do greške.
- ▶ HTTP odgovor se sastoji iz:
 - ▶ Statusne linije;
 - ▶ Zaglavlja;
 - ▶ Tijela poruke, koje je obično neophodno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon posljednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP odgovor - Statusna linija

- Statusna linija je prva linija u odgovoru. Sasloji se iz tri segmenta:
 - Broj HTTP verzije, koji ukazuje na HTTP specifikaciju po kojoj je server pokušao da usladi odgovor.
 - Statusni kod, koji je trocifarski broj i ukazuje na rezultat izvršenja zahtjeva.
 - Fraza razloga, poznata i kao tekst statusa, koji je čitljiv čovjeku i sažima značenje statusnog koda.


Primjer statusne linije:

```
HTTP/1.1 200 OK
```

HTTP odgovor – Zaglavlja (Headers)

- Sadrži informacije koje klijent koristi da pronađe više podataka o odgovoru, kao i da pronađe podatke o serveru koji je poslao poruku.
- Ove informacije mogu pomoći klijentu u prezentaciji odgovora korisniku.
- Na primjer, prikazana zaglavlja govore klijentu kada je odgovor poslat, od strane kojeg servera je poslat, kao i da je to JPEG slika.

```
Date: Thu, 09 Dec 2004 12:07:48 GMT  
Server: IBM_CICS_Transaction_Server/3.1.0(zOS)  
Content-type: image/jpeg
```




HTTP odgovor – Tijelo poruke

- ▶ Naziva se i tijelom odgovora.
- ▶ Većina odgovora sadrže tijelo poruke. Izuzeci su kada server odgovara na zahtjev klijenta, koji je koristio HEAD metod (koji koristi zaglavlja ali ne i tijelo odgovora) i gdje server koristi određene statusne kodove.
- ▶ U odgovoru na uspješno izvršen zahtjev, tijelo poruke sadrži resurs koji je klijent zahtijevao ili neke informacije o statusu radnje koji je klijent zahtijevao.
- ▶ U odgovoru na neuspješno izvršen zahtjev, tijelo poruke može da pruži dodatne informacije o razlozima greške ili o nekoj radnji koju klijent treba da preduzma da bi se zahtjev uspješno izvršio.



HTTP zahtjev - metode

- ▶ HTTP definiše set metoda (načina) da indicira akciju koja će biti izvršena na datom resursu.
- ▶ Mada mogu biti i imenice, metode zahtjeva se često označavaju kao HTTP glagoli.
- ▶ Svaki metod koristi različitu semantiku.



HTTP zahtjev – vrste metoda

- **GET** Get metod često zahtijeva reprezentaciju navedenog resursa i samo vraća podatke.
- **HEAD** HEAD metod očekuje odgovor identičan GET zahtjevu, ali bez tijela odgovora.
- **POST** POST metod šalje entiteti specificiranom resursu, često izazivajući promjene stanja ili druge efekte na serveru.
- **PUT** PUT metod sve tekuće prikaze ciljnog resursa sa sadržajem zahtjeva.
- **DELETE** DELETE metod briše specificirani resurs.
- **CONNECT** Uspostavlja vezu sa serverom, onosno, identificiranim ciljanim resursom.
- **OPTIONS** OPTIONS metod opisuje komunikacione opcije za ciljani resurs.
- **TRACE** TRACE metod vrši praćenje komunikacionog linka do ciljanog resursa.
- **PATCH** PATCH metod obavlja parcijalnu modifikaciju resursa

HTTP zahtjev – **GET** metod

- ▶ GET se koristi za traženje podataka iz specificiranog izvora
- ▶ Treba imati na umu da se upitni string (par ime/vrijednost) šalje u URL-u GET zahtjeva.

`/test/demo_form.php?name1=value1&name2=value2`

- ▶ Nekoliko napomena u vezi GET zahtjeva:
 - ▶ GET zahtjevi se mogu keširati (spremiti u predmemoriju)
 - ▶ GET zahtjevi ostaju u historiji pregledača
 - ▶ GET zahtjevi se mogu objeležiti
 - ▶ GET zahtjevi se nikada ne bi trebali koristiti kada se radi o osjetljivim podacima
 - ▶ GET zahtjevi imaju ograničenje dužine
 - ▶ GET zahtjevi se koriste samo za traženje podataka (ne promjenu)

HTTP zahtjev – **POST** metod

- ▶ POST metod se koristi za slanje podataka serveru za kreiranje/ažuriranje resursa
- ▶ Podaci poslani serveru POST metodom smješteni su u tijelu HTTP zahtjeva.

```
POST /test/demo_form.php HTTP/1.1  
Host: w3schools.com
```

```
name1=value1&name2=value2
```

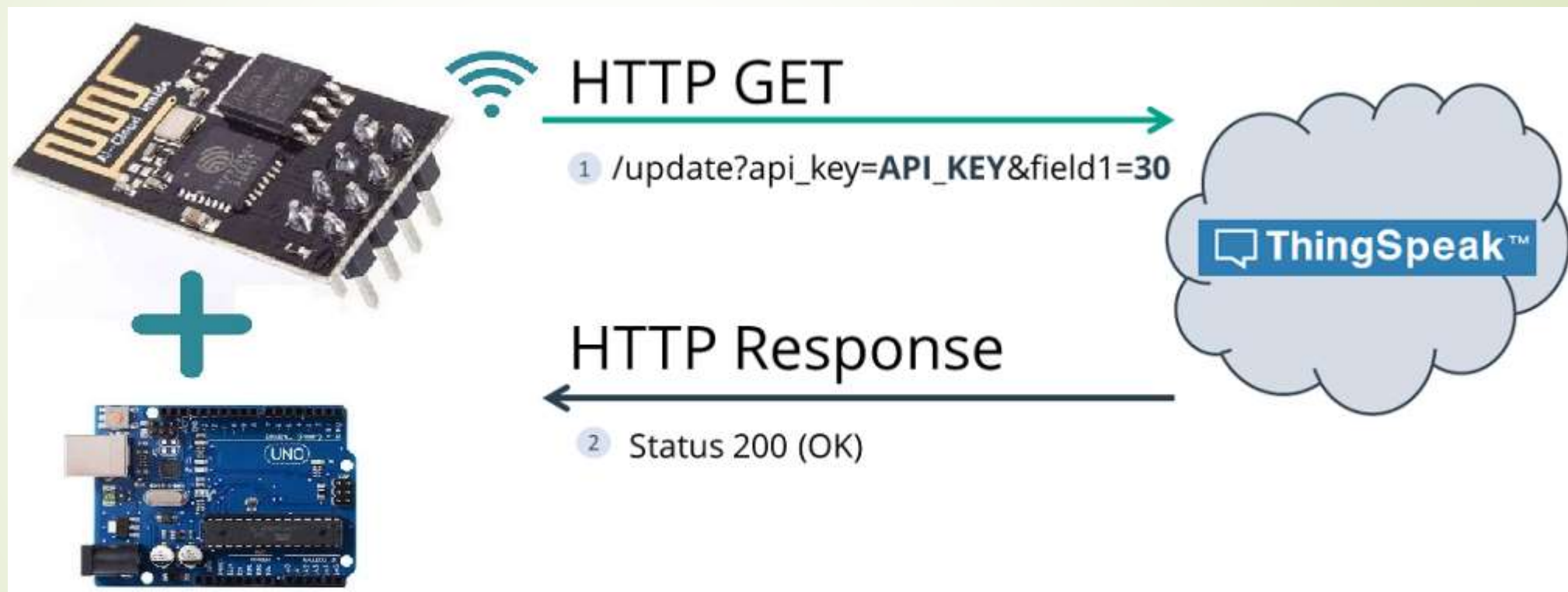
- ▶ Nekoliko napomena u vezi POST zahtjeva:
 - ▶ POST zahtevi se nikada ne kešuju
 - ▶ POST zahtevi ne ostaju u historiji pregledača
 - ▶ POST zahtevi se ne mogu označiti
 - ▶ POST zahtevi nemaju ograničenja u pogledu dužine podataka

HTTP zahtjev – GET vs. POST metod

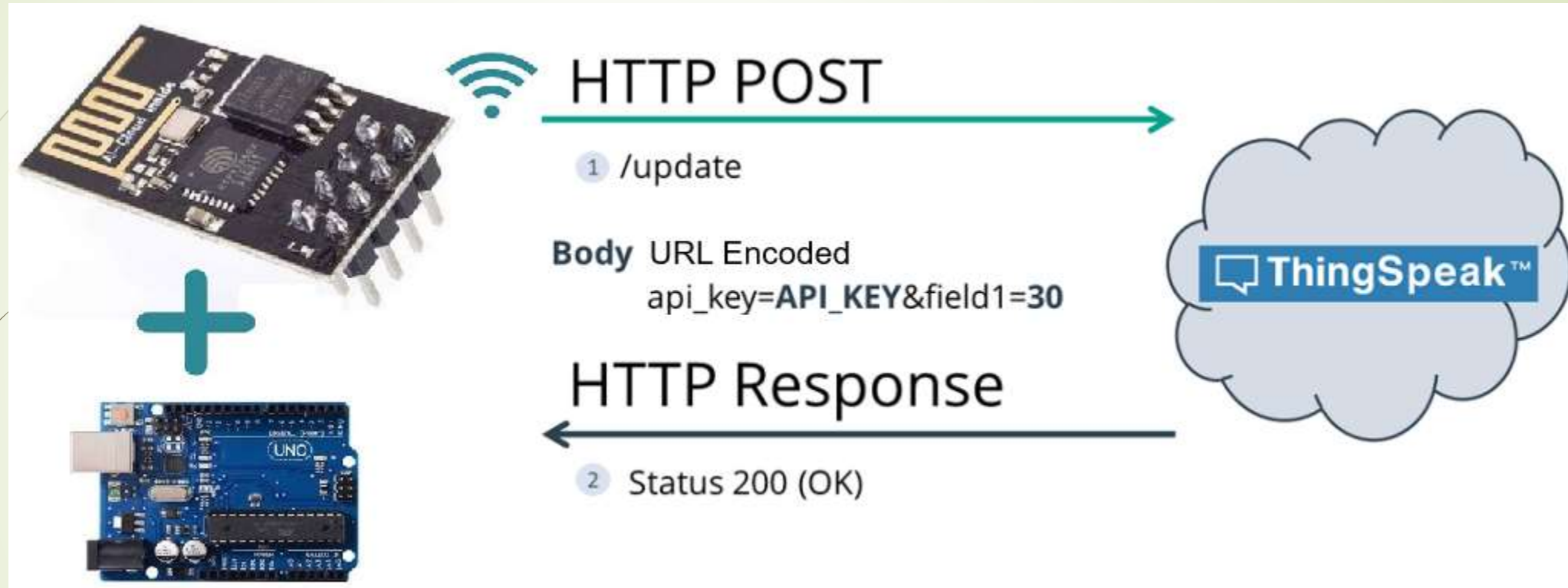
| | GET | POST |
|-------------------------------|--|--|
| BACK dugme/Ponovo učitaj | Neškodljivo | Podaci će biti ponovo poslani |
| Označenost | Može se označiti | Ne može se označiti |
| Keširanje | Može se keširati | Ne može se keširati |
| Tip kodiranja | application/x-www-form-urlencoded | application/x-www-form-urlencoded ili multipart/form-data. Upotreba višedjelnog kodiranja za binarne podatke |
| Istorija | Parametri ostaju u istoriji pregledača | Parametri ne ostaju u istoriji pregledača |
| Ograničenja u dužini podataka | Da, kod slanja podataka, GET method dodaje podatke na URL; dužina URL-a je ograničena (maximalna URL dužina je 2048 karaktera) | Bez ograničenja |
| Ograničenja u tipu podataka | Samo ASCII karakteri dozvoljeni | Bez restrikcija. Binarni podaci su također dozvoljeni |
| Bezbjednost | GET manje siguran u poređenju s POST, jer su podaci dio URL-a Ne koristiti GET kada se šalje ložinka ili druge osjetljive informacije | POST je malo sigurniji od GET jer parametri nijesu smješteni u istoriji pregledača ili u web server logu |
| Vidljivost | Podaci su vidljivi svima u URL-u | Podaci nijesu prikazani u URL-u |

Arduino UNO+ESP8266 – GET preko URL-a

URL (Uniform Resource Identifier) je u stvari web adresa oblika http ili https.

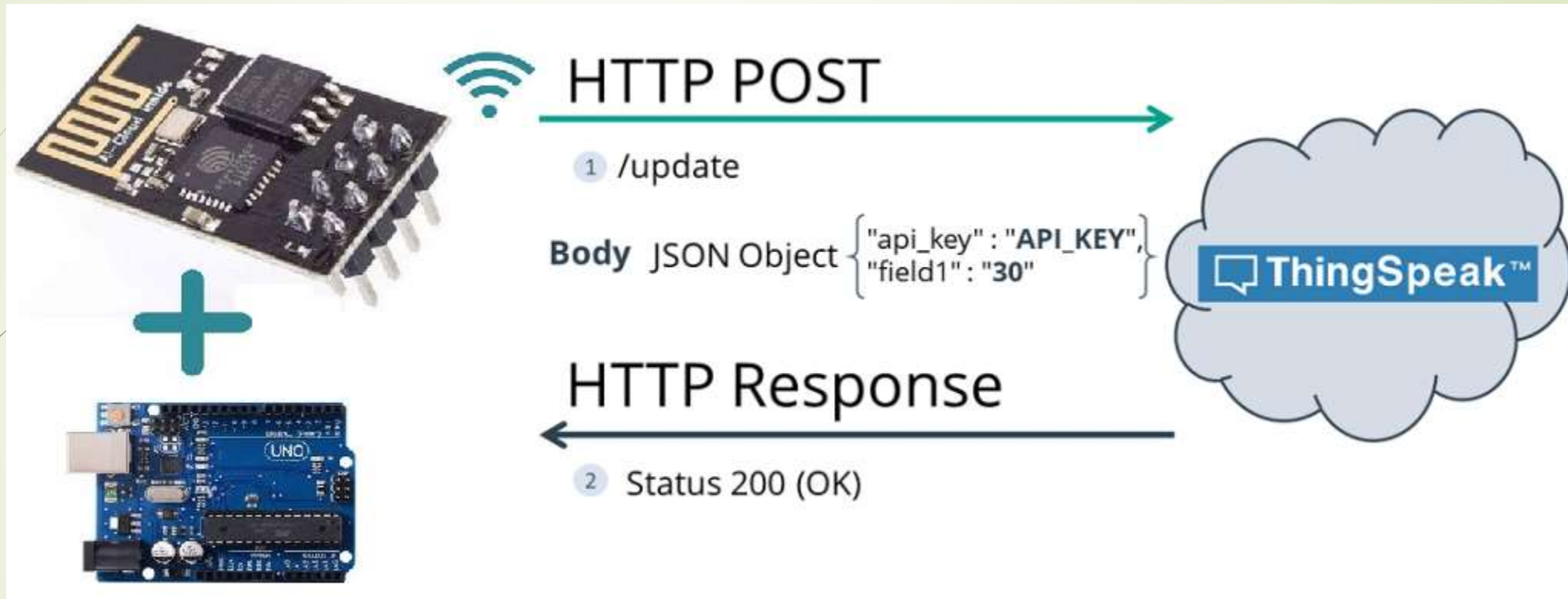


Arduino UNO+ESP8266 – HTTP POST - URL enkodiran



```
POST /update HTTP/1.1
Host: example.com
api_key=api&field1=value1
Content-Type: application/x-www-form-urlencoded
```

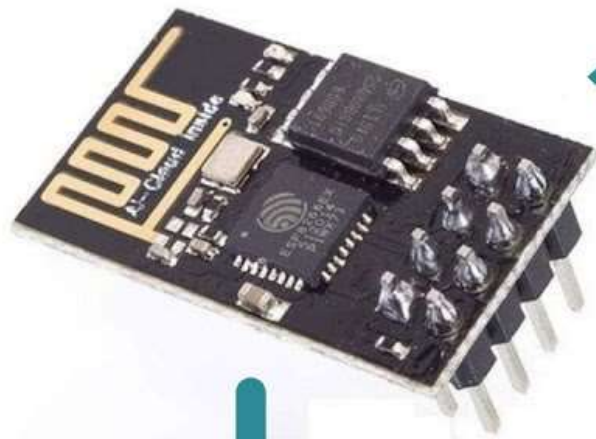
Arduino UNO+ESP8266 – HTTP POST - JSON object



```
POST /update HTTP/1.1
Host: example.com
{api_key: "api", field1: value1}
Content-Type: application/json
```

Upisivanje podataka u ThingSpeak kanal

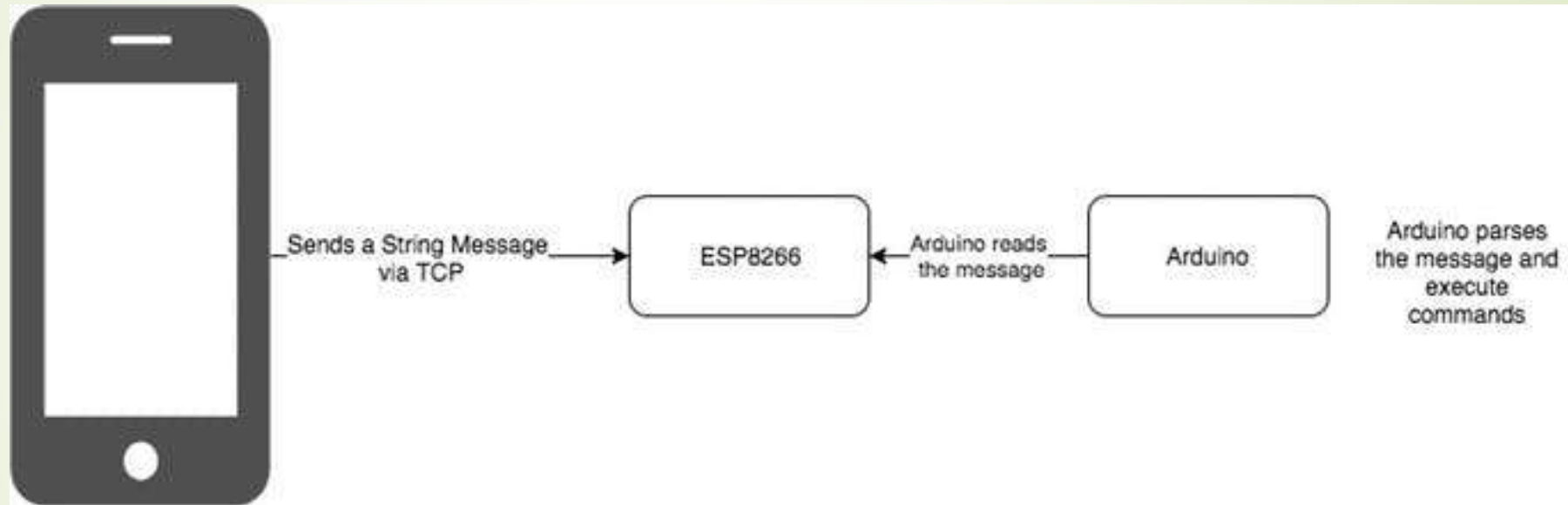
ESP8266 SEND DATA To WEBSITE



**ThingSpeak
WEBSITE**



Komuniciranje sa Arduino Uno bezžično upotrebom pametnog mobilnog telefona

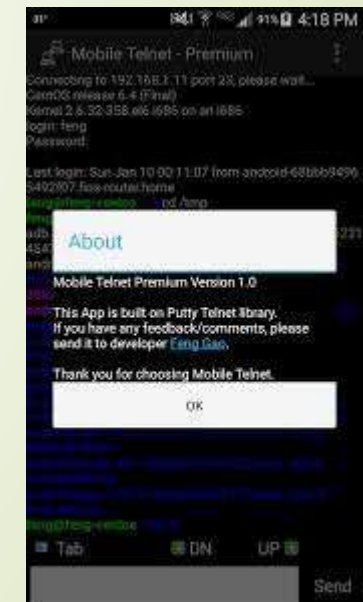
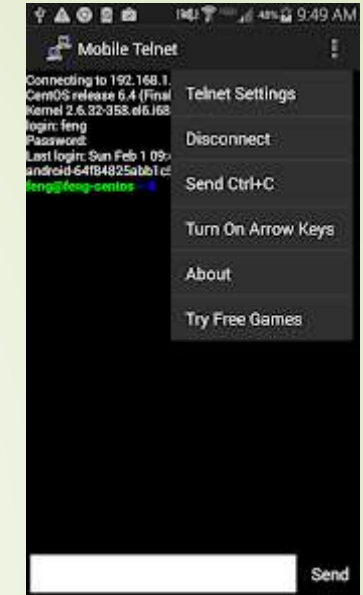


- ESP8266 će raditi u AP (Access Point) modu. Što znači da će drugim uređajima omogućavati pristup na njegovu WiFi mrežu.
- Proces je jednostavan. Upotrijebite telefon da pošaljete bilo koju komandu Arduino. Uz pomoć ESP8266 sve će raditi bezžično.

Potrebni elementi

Upišite **esp8266asAP23.ino** skeč u Arduino

- ▶ Podešavanje telefona:
 - ▶ Instalirajte TCP klinta za Android
 - ▶ Možete instalirati bilo koji TCP klijent koji je raspoloživ na Play Store, a možete i baš
Mobile Telnet
 - ▶ Povežite vaš telefon na ESP8266 WiFi
 - ▶ Nakon povezivanja otvorite Mobile Telnet aplikaciju
 - ▶ Kreirajte konekciju klikom na „Telnet Settings“ opciju menija i dodajte IP (Host Name) i Port
 - ▶ Povežite se klikom na “Connect” opciju menija





Već podržane poruke

- Kažite: HELLO
- Kažite: Kako ide?
- Uključite LED komandom LEDON
- Ili isključite LED komandom LEDOFF
- Možete mijenjati i dalje nadograđivati.



Za vježbu

1. Upotrijebiti sensor za temperaturu i vlagu. Vrijednosti dobijene sa senzora slati na ThingsSpeak i prikazivati u dva odvojena dijagrama istog kanala. (2-1)
2. Upotrebom TCP klijenta na pametnom mobilnom telefonu, podešavati poziciju servo motora i podešavati intezitet sjaja 3 LED. (3-2)
3. Upotrebom TCP klijenta na mobilnom telefonu, na LCD displeju ispisivati poruke, poslate pomoću TCP klijenta, kao i upravljati radom koračnog motora (mijenjati brzinu okretanja, mijenjati smjer okretanja i zaustavljati i pokretati koračni motor). (4-2)