Predgovor

Wi-Fi (ili WiFi) je tehnologija lokalnih mreža zasnovana na standardima IEEE 802.11 koji dozvoljavaju da različiti uređaji budu povezani bežično. Uređaji koji obično koriste Wi-Fi tehnologiju uključuju lične računare, pametne telefone, tablete, savremene štampače, digitalne kamere, konzole za video igre itd. Uređaji na Wi-Fi mrežama su međusobno vezani preko pristupnih tačaka (Access Point - AP). Većina kuća, radnih mesta, vladinih ustanova, železničkih stanica, restorana i drugih javnih mesta su vezana i koriste internet ili proveravaju elektronsku poštu dok su u pokretu.

Kućne AP su dometa 20 metara unutar zgrade ali se ripiteri mogu upotrebiti da prošire na željeni domet. Wi-Fi obično koristi UHF opseg učestanosti od 2.4 GHz i može se javiti interferencija sa nekim kućnim aparatima kao što su mikrotalasne peći.

Da bi se uređaj vezao na Wi-Fi potreban je procesor sa interfejsom za kontrolu mreže (NIC- Network Interface Contoroller). Laptop računari i pametni telefoni su obično opremljeni NIC i mogu se lako vezati na lokalnu Wi-Fi. Sve što je potrebno je znati naziv Wi-Fi (SSID naziv) i Wi-Fi lozinka. Vezani uređaji mogu pristupiti svim resursima interneta, uključujući slanje i primanje elektronske pošte, korišćenje Fejsbuka i Tvitera itd. Stoni računari obično nemaju ugrađen NIC ali eksterni NIC su na raspolaganju u obliku fleš uređaja preko kojih se može vezati preko USB porta pa mogu komunicirati sa lokalnim AP.

Mikrokontroleri se obično koriste za nadzor i kontrolu u domaćinstvima, trgovini i industriji. Postoji mnoštvo aplikacija gde postoji potreba da se mikrokontrolersko bazirani sistem veže na lokalnu Wi-Fi mrežu tako da pogon, deo opreme ili prekidač može biti nadziran i upravljan sa udaljenog mesta. Uglavnom je eksterni hardver u obliku NIC-a i može biti vezan na mikrokontroler tako da može komunicirati sa AP. Ranije su takvi NIC uređaji bili veliki i skupi i trošili mnogo struje, zahtevajući velika napajanja.

Odskora razvoj malog čipa zvanog ESP8266 je učinilo mogućim vezati bilo koji tip mikrokontrolera na Wi-Fi AP. ESP8266 je jeftin, sitan Wi-Fi čip, sa potpuno ugrađenim TCP/IP protokolom i 32-bitnom mikrokontrolerskom jedinicom. Ovaj čip proizvodi kineski proizvođač iz Šangaja Espressif System i kompatibilan je sa IEEE 802.11 b/g/n, a na čipu ima programsku memoriju i memoriju podataka kao i ulaze i izlaze opšte namene (GPIO) kao portove. Nekoliko proizvođača je ugradilo ESP8266 u svoje hardverske proizvode (ESP-xx, NodeMCU itd) i nude ove proizvode kao sredstva za vezu mikrokontrolerskih sistema kao što je Android, PIC mikrokontroleri i drugi na Wi-Fi. ESP8266 malo troši i košta nekoliko dolara.

Knjiga je uvod u čip ESP8266 i opisuje osobine čipa i pokazuje kako različiti firmveri i programski jezici kao što je Micropython mogu biti preneti na čip. Glavni cilj knjige je naučiti čitaoca kako upotrebiti Micropython, programski jezik, na ESP zasnovanom hardveru, naročito na NodeMCU. U knjizi je dato nekoliko zanimljivih i korisnih projekata da prikažu kako upotrebiti Micropython na uređaju NodeMCU, zasnovanom na čipu ESP8266.

Dogan Ibrahim
Ahmet Ibrahim
London 2017

Poglavlje 1 • ESP8266 procesori

1.1 Pregled

Jeftini Wi-Fi čip ESP8266 se može upotrebiti sa ili bez mikrokontrolera da bi se pristupilo Wi-Fi mreži. Čip je razvio kineski proizvođač iz Šangaja Espressif Systems i ugradio čitav TCP/IP sloj. Sam po sebi se može upotrebiti kao mali mikrokontroler ili se može povezati sa jačim razvojnim mikrokontrolerom kao što je Arduino gde se može koristiti da bi se obezbedila Wi-Fi sposobnost glavnog sistema.

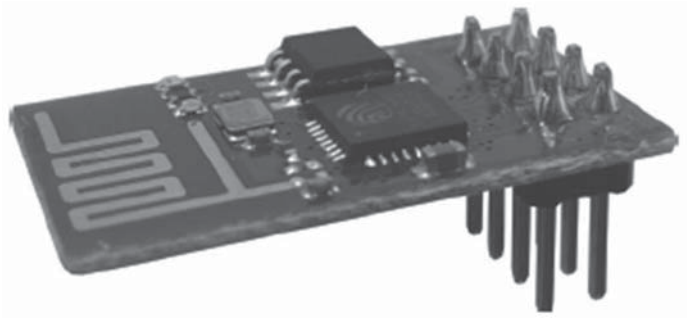
Postoji više verzija čipa ESP8266 a na raspolaganju su kao moduli na pločicama sa raznim funkcijama i različitim ulazno/izlaznim kapacitetima. Osnovna specifikacija čipa ESP8266 su navedena dole:

- 32-bitni RISC CPU: Tensilica Xtensa LX106
- Radni takt 80 - 160 MHz (uz overklok)
- Rad na 3.3 V (min +2.5 V do max 3.6 V)
- 64 Kbajta programske memorije, 96 Kbajta memorije podataka
- Eksterna QSPI fleš memorija (sve do 16Mbajta)
- Podrška za Wi-Fi po IEEE 802.11 b/g/n
- Provera i overa po WEP ili WPA/WPA2 (ili otvoren pristup mreži)
- Analogno digitalni pretvarač ADC 1 x 10 bita
- Deljeni interfejsni izvodi SDIO,SPI, I^2C, I^2S
- Integrisani sloj protokola TCP/IP
- Dodeljeni izvodi UART-a (TX samo na GPIO2 izvodu kao dodatak)
- Potrošnja u pripravnosti < 10mW

Detalji hardvera nekih popularnih ESP8266 modula su opisani u ovom poglavlju.

1.2 ESP-01

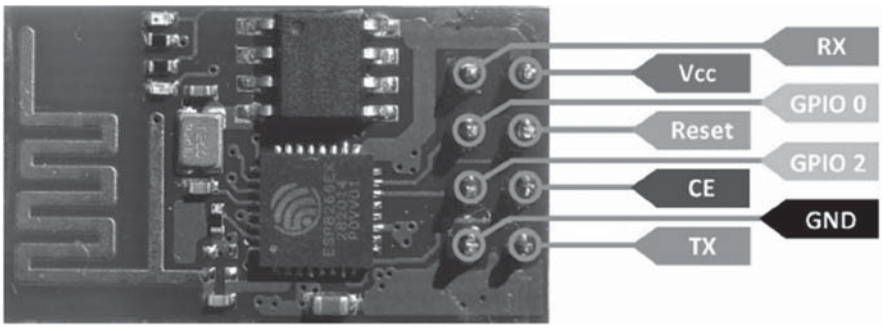
Prvu verzija čipa je napravio AI-Thinker i isporučivana je na pločici kao modul pod nazivom ESP-01. Kao što je prikazano na slici 1.1 mali modul je dimenzija 14.3 mm x 24.8 mm, sa dxDIL konektora sa po 8 izvoda (6 aktivnih pinova, napajanje i masa) i na PŠV nacrtana antena sa LE diodama koje ukazuju prisustvo napajanja i prenos podataka.



Slika 1.1 Modul ESP-01

Slika 1.2 prikazuje konfiguraciju izvoda modula ESP-01 sa sledećim značenjem

Vcc	+3.3 V napajanje
GND	izvod mase
RX	UART prijemni izvod
TX	UART predajnik
GPIO0	I/O izvod 0
GPIO2	I/O izvod 2
Reset	Izvod za reset
CE	Izvod za dozvolu rada čipa



Slika 1.2 Konfiguracija izvoda na ESP-01

Važno je zapaziti da čip radi sa eksternim napajanjem od +3.3 V i svi ulazi i izlazi su kompatibilni sa +3.3 V. Pripazite da na čip ne vežete napajanje od +5 V pošto čip oštećuje svaki napon viši od +3.3 V.

Napajanja od +5 V su u većini slučajeva na raspolaganju za većinu primena. Ukoliko napajanje od 3.3 V nije trenutno na raspolaganju moguće je upotrebiti modul regulatora sa +5 V na 3.3 V ili čip za regulaciju ili naponski delitelj koji pravi +3.3 V od +5 V ili redno vezane diode ili par AA baterija. Sve mogućnosti su opisane u daljem tekstu.

Poglavlje 2 • Softver ESP8266

2.1 Pregled

Čip ESP8266 može preuzeti softver tako da se može programirati ako se koristi jedan od sledećih programskih jezika:

- AT naredbe
- LUA (koristeći NodeMCU firmver)
- C++ (preko Arduino IDE)
- microPython

AT naredbe su razvijene za slanje komandi modemima i sličnoj komunikacionoj opremi da bi se mogla konfigurisati kao i za slanje i prijem podataka. Firmver se može preneti na ESP8266 čip tako da on može prihvatiti AT naredbe.

Lua je snažan, efikasan, lagan skript jezik koji se može ugraditi i koji podržava objektno orijentisano, proceduralno kao i programiranje preko podataka. Lua je interpreterski jezik pa je savršen za brzo učenje i pravljenje prototipova.

C++ je objektno orijentisan snažan jezik visokog nivoa koji se uglavnom koristi kod velikih aplikacija na laptopovima i desktop računarima. Programirane na C++ je izvan domena ove knjige i nalazi se u drugim knjigama, priručnicima a na internetu postoje beleške o ovoj temi. Zainteresovanim čitaocima preporučujemo da pretraže internet radi informacija.

microPython je podskup snažnog jezika visok nivoa Python (Python). Upotrebljava se na skoro svim univerzitetima po svetu na uvodnim kursevima programiranja i nalazi se na većini mikrokontrolerskih sistema.

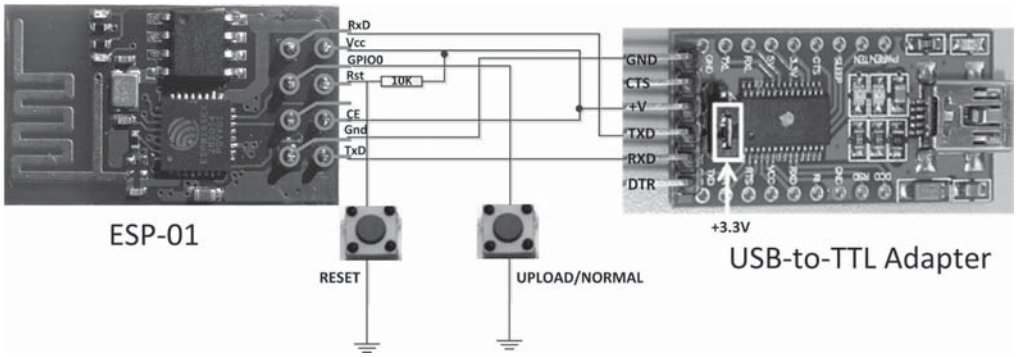
Upotreba Micropythona na ESP8266 pločicama je glavni cilj ove knjige. U ovom poglavlju ukratko pregledamo kako da preuzmemo firmver tako da našu ESP 8266 pločicu programirano uz pomoć AT naredbi, Lua i sa Micropythonom. Detalji Micropythona sa primerima će bit dati u sledećem poglavlju.

2.2 ESP8266 sa AT naredbama

ESP-01 je fabrički opremljen sa firmverom koji prepoznaje AT naredbe za rad sa Wi-Fi-jem. Firmver u vašem uređaju može biti star pa je potrebno ažuriranje. Postupak preuzimanja se normalno radi uz pomoć PC-ja. U ovom odeljku ćemo pogledati korake za preuzimanje novog firmvera za naš ESP-01.

2.2.1 Postavka hardvera

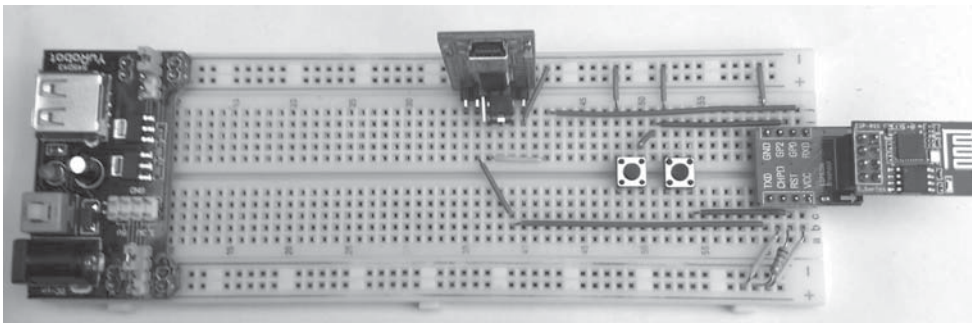
Pre upotrebe ESP-01 za normalan rad ili za preuzimanje novog firmvera potrebno je uređaj vezati na USB port našeg PC-ja uz pomoć USB-TTL adaptera kao što je prikazano u prethodnom poglavlju na slici 1.10. Slika 2.1 ponovo pokazuje potrebnu električnu šemu ali ovog puta slikovito. Taster RESET se koristi za reset uređaja. Slično taster UPLOAD/NORMAL se koristi za izbor normalnog rada ili za preuzimanje novog softvera na uređaj. Zapazite da će vam možda biti potrebno da vežete eksterno napajanje od +3.3 V (ili 2 AA baterije) da bi bilo dovoljno energije naročito ako se koristi Wi-FI.



Slika 2.1 Minimalna postavka za upotrebu ESP-01

Možda je najlakša postavka napraviti kolo na standardnoj eksperimentalnoj pločici. Zapamtite da ESP-01 nije kompatibilna sa eksperimentalnom pločicom pošto su izvodi blizu tako da će Vam biti potreban adapter da bi ESP-01 vezali na eksperimentalnu pločicu. Pogodan adapter za ESP-01 se može nabaviti na sledećem sajtu: *www.diymalls.com*

Slike 2.2 i 2.3 prikazuju pogled odozgo i sa strane na kolo napravljeno na pločici. Zapazite da je adapter USB-TTL montiran vertikalno. Kolo se napaja sa 2 x AA alkalne baterije.

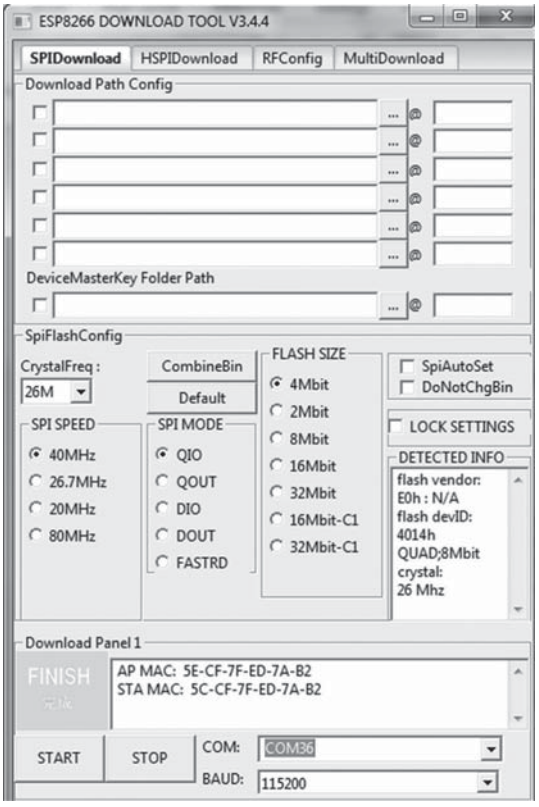


Slika 2.2 Pogled odozgo na kolo

```
# NON-BOOT MODE
## download
    eagle.flash.bin              0x00000
    eagle.irom0text.bin          0x10000
    blank.bin
        Flash size 8Mbit:        0x7e000 & 0xfe000
        Flash size 16Mbit:       0x7e000 & 0x1fe000
        Flash size 16Mbit-C1:    0xfe000 & 0x1fe000
        Flash size 32Mbit:       0x7e000 & 0x3fe000
        Flash size 32Mbit-C1:    0xfe000 & 0x3fe000
```

*Slika 2.14 Odeljak **download** readme fajla*

- Povežite vaš ESP-01 kao na slici 2.1 i pokrenite Flash Download Tool i izaberite procesor ESP8266. Postavite COM port parametre kao ranije u ovom primeru, COM36 sa brzinom u bodima od 115200.
- Pritisnite i držite taster RESET a zatim držite taster UPLOAD/NORMAL na ESP-01. Otpustite taster RESET, kliknite na taster START na ESP Flash Download Tool-u na PC-ju. Trebalo bi da vidite ekran sličan onome koji je prikazan na slici 2.15 koji detektuje i pokazuje specifikaciju pločice. Zapamtite veličinu fleš memorije navedenu u prozoru DETECTED INFO. U ovom primeru veličina fleš memorije je 8Mbit-a a brzina kristala 26MHz.



Slika 2.15 Pokretanje Flash Download Tool-a

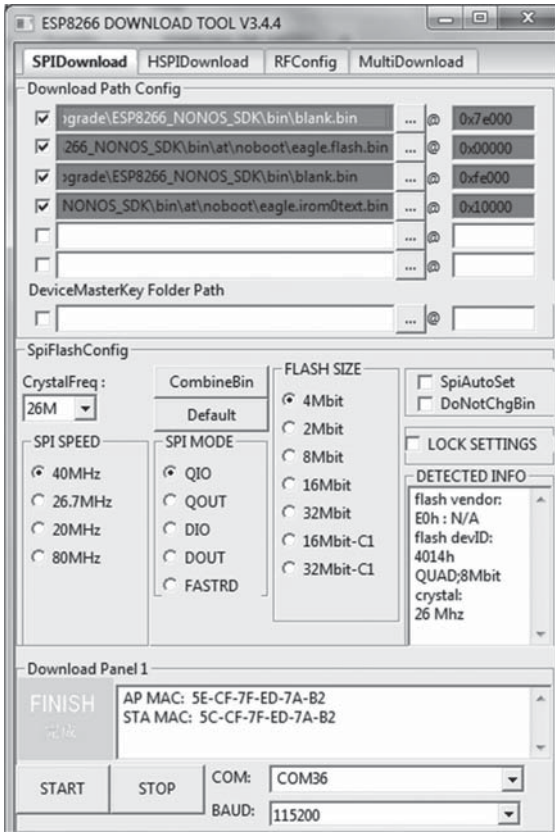
- Firmver je podeljen u nekoliko fajlova (normalno 3) a oni su neophodni za ESP Flash Download Tool zajedno sa adresama u readme fajlu. Ovi fajlovi su navedeni u NON-BOOT odeljku fajla i oni su (imena direktorijuma i pod direktorijuma mogu biti različiti na vašem PC-ju:

  eagle.flash.bin u pod direktorijumu
  ESP8266-AT-upgrade\ESP8266_NONOS_SDK\bin\at\noboot

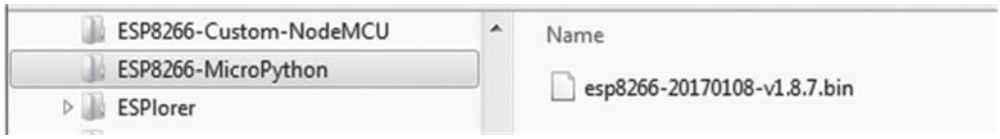
  eagle.irom0text.bin u pod direktorijumu
  ESP8266-AT-upgrade\ESP8266_NONOS_SDK\bin\at\noboot

  blank.bin u pod direktorijumu
  ESP8266-AT-upgrade\ESP8266_NONOS_SDK\bin

Ispunite pravilno putanje do fajlova i adresa (ofsete). Zapazite da se fajl **blank.bin** pojavljuje dva puta sa dva različita ofseta. Posle ispunjavanja putanja i adresa trebalo bi da imate ekran sličan onome na slici 2.16. Putanje i adrese su detaljno prikazane na slici 2.17



Slika 2.16 Formular sa ispunjenim putanjama i adresama



Slika 2.45 Poslednja verzija firmvera memorisana u pod direktorijumu

- Sada preuzimamo **esptool** koji je softverski alat korišćen za skidanje firmvera na ESP8266 procesore. Nastavite sledećim koracima:
- Dok je još otvorena komandna linija idite u osnovni ‚root' direktorijum unosom CD\
- Unesite sledeću naredbu da bi instalirali esptool na vaš PC (vidi sliku 2.46). Trebalo bi da vidite poruku **Successfully installed esptool** na kraju instalacije.

C:> pip install esptool

```
C:\>
C:\>pip install esptool
Collecting esptool
  Using cached esptool-1.3.tar.gz
Requirement already satisfied: pyserial>=2.5 in c:\users\dogan\appdata\local\pro
grams\python\python36-32\lib\site-packages (from esptool)
Installing collected packages: esptool
  Running setup.py install for esptool ... done
Successfully installed esptool-1.3

C:\>
```

Slika 2.46 Instalacija esptool-a

- Preporučuje se brisanje fleš memorije ESP8266 pre upisa novog firmvera. Da bi to uradili vežite NodeMCU na PC preko mini USB kabla. Pronađite broj serijskog porta na koji je vezan NodeMCU (vidite sliku 2.4) i unesite sledeću naredbu. U ovom primeru se pretpostavlja da je NodeMCU vezan na serijski port COM2. Slika 2.47 pokazuje koja se naredba izvršava da bi se obrisala fleš memorija. Postupak brisanja bi trebalo da potraje oko 10 sekundi.

C:>esptool.py—portCOM2 erase_flash

ili

C:>python C:\pzthon27\lib\site/packages\esptool.pz//portCOM2 erase_flash

```
C:\>python c:\python27\lib\site-packages\esptool.py --port COM2 erase_flash
esptool.py v1.3
Connecting....
Running Cesanta flasher stub...
Erasing flash (this may take a while)...
Erase took 11.8 seconds

C:\>
```

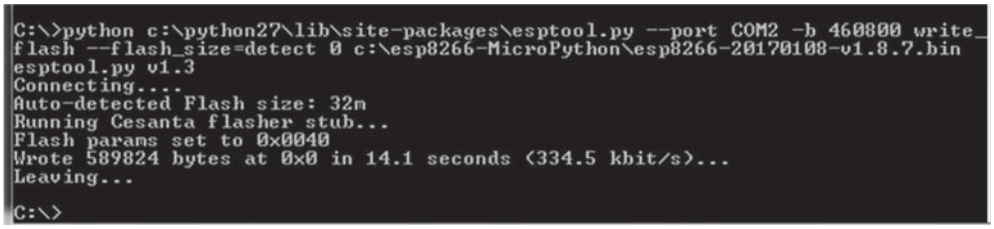
Slika 2.47 Brisanje celokupne memorije ESP8266

- Sačuvajte fajl u pod direktorijumu. U ovom primeru memorisan je u pod direktorijumu ESP8266-MicroPython a naziv fajla je ESP8266-20170108-v1.8.7.bin (vidi sliku 2.45)Sada smo spremni da prenesemo Micropythonov firmver na našu razvojnu pločicu NodeMCU. Proverite da li je NodeMCU još uvek vezan na USB port i unesite sledeću naredbu (vidite sliku 2.48) . Veličina fleš memorije će biti automatski detektovana (32Mbit _ 4MByte u ovom primeru) a Micropython firmver će se preneti na NodeMCU:
  C:>esptool.py—portCOM2 —baud 460800 write_flash —flash_size=detect 0

  C:\ESP8266/MycroPython\ESP8266-20170108-v1.8.7.bin

ili

  C:>python c:\python27\lib\site-package\esptool.py —portCOM2 —baud 460800 write_flash - flash_size =detect 0 C:\ESP8266-MycroPython\ESP8266-201701-v1.8.7.bin



Slika 2.48 Preuzimanje Micropython firmvera

Pošto se prenos završio pritisnite taster REST pa će NodeMCU sada biti spreman za aplikacije Micropythona

2.8 Upotreba REPL u Micropythonu

REPL (Read-Evaluate-Print-Loop) omogućava interaktivne naredbe (iskaze) koji se unose preko prozora emulatora terminala da bi se poslale naredbe Micropythonu. REPL je koristan naročito kad se uči Micropython dok korisnik eksperimentiše i uči Micropythonove iskaze. REPL se koristi preko prozora emulatora terminala kao što je Putty ili HyperTerm. Pre upotrebe REPL-a trebalo bi konektovati modul NodeMCU na PC i pronaći broj dodeljenog serijskog porta (vidi sliku 2.4) U ovom odelju upotrebljavamo HyperTerm kao emulator terminala. Koraci korišćenja REPL-a su dati dole:

- Povežite NodeMCU na PC i pronađite broj dodeljenog serijskog porta (COM45 u ovom primeru)
- Pokrenite HyperTerm emulator terminala i konekciji dodelite naziv (npr. MycroPython). Izaberite serijski port kao što je dato na slici 2.49

Poglavlje 3 • Upotreba Micropythona na NodeMCU

3.1 Pregled

Python je interpreter, interaktivni i objektno orijentisani programski jezik visokog ranga, koga je prvi razvio Gvido Rosum 1980 u Nacionalnom istraživačkom institutu za matematiku i računarske nauke u Holandiji. Python je izveden iz mnoštva drugih jezika, uključujući C i C++, Modula 3, SmallTalk i Juniks šela. Sada jezik održava tim ljudi na institutu. Micropython je podskup Pythona i razvijen je uglavnom za ugradne procesore kao što su PIC, Arduino, ARM i drugi. Micropython poseduje većinu osobina svog velikog brata Pythona uključujući mnoštvo ugrađenih biblioteka. Osobine opisane u ovom odeljku važe kako za Python tako i za Micropython.

Python je postao jedan od najpopularnijih jezika i koriste ga mnogi koledži i univerziteti kao prvi programski jezik pošto se lako uči i lako održava. Python je interaktivan što znači da možete upotrebiti naredbu i videti rezultat trenutno bez kompajliranja ili prenošenja naredbe na Python procesor. Ova osobina je krajnje korisna kada se uči programski jezik. Python je takođe interpreter pa ne zahteva prethodno kompajliranje pre rada sa njim.

Python podržava veliku biblioteku funkcija što je možda jedna od najvećih prednosti po čemu se razlikuje od ostalih programskih jezika. Micropython podržava mnoštvo funkcija čineći lakšim razvoj ugradbenih programa za razne uređaje .

U ovom poglavlju ćemo pogledati detaljno Micropythonov programski jezik pa bi videli kako se pišu programi u ovom jeziku. Koristićemo aktivni interfejs REPL na Micropythonu tako da trenutno vidimo rezultate različitih programskih iskaza. Koristićemo razvojnu pločicu NodeMCU koja je opremljena Micropython firmverom iz poglavlja 2. Ovo poglavlje je organizovano kao priručnik i namenjeno je početnicima u Micropython programskom jeziku. Pretpostavlja se da čitaoci imaju nekakva praktična znanja visokih programskih jezika.

trebalo bi vezati NodeMCU na vaš ŠC i pokrenuti softver za emulaciju terminala da bi komunicirali sa majkropajtonovim firmverom. HyperTerm emulator terminala (vidi poglavlje 2.8) se koristi u ovom poglavlju

3.2 Nazivi promenljivih

Kao većina programskih jezika Nazivi Micropythonovih promenljivih razlikuju velika i mala slova i mogu počinjati slovima od A do Z ili a do z ili znakom donje povlake”_” iza čega sledi slova ili brojevi od 0 do 9. Ipak nije dozvoljeno naziv promenljive početi sa brojem. Primeri važećih i nevažećih naziva promenljivih su dati dole:

Count	- važeća
Total	- važeća
_sum	- važeća
9Cnt	- nevažeća
$100	- nevažeća
%total	- nevažeća
Count1	- važeća
My_Name	- važeća
@loop	- nevažeća

Zapazite da pošto Micropython razlikuje velika i mala slova promenljive kao što su COUNT, COUNt, COUnt, Count i count su potpuno različite.

3.3 Rezervisane reči

Neki nazivi su rezervisani zato što ih koristi Micropython i ne smeju se upotrebiti kao nazivi promenljivih u našem programu. Spisak rezervisanih naziva je dat dole. Zapazite da se sva rezervisana imena sastavljena od malih slova.

And	for	raise
assert	from	return
break	global	try
class	if	while
continue	import	with
def	in	yield
del	ins	
elif	lambda	
else	not	
except	or	
exec	pass	
finally	print	

3.3 Primedbe - komentari

Linije komentara u Micropythonu počinju znakom #. Svi znaci posle znaka # se ignorišu od Micropythonovog interpretera. Komentari su ili na početku linije ili posle iskaza. Neki primeri komentara su prikazano dole:

```
# This is a Micropython comment line
cnt=10 # inicijalizacija promenljive cnt do 10
```

3.4 Nazupčavanje novih redova

Većina programskih blokova koda u programskim jezicima se označava zagradama na početku i kraju bloka ili označavanjem kraja bloka kao END, WEND ili ENDIF. U jeziku Micropython nema zagrada ili iskaza koji bi ukazivali na početak ili kraj koda. Umesto toga blokovi koda se označavaju uvlačenjem novih redova (linija). Svi izrazi u bloku moraju biti uvučeni za isti stepen. Stvarni broj razmaka koji se koristi za uvlačenje nije bitan sve dok su iskazi koji pripadaju istom bloku uvučeni za isti broj razmaka. Neki primeri su dati dole:

Poglavlje 4 • Kreiranje programa i prenos na NodeMCU

4.1 Pregled

U poslednjim poglavljima smo videli detalje Micropythonovog programskog jezika i videli kako se interaktivno koristi Micropythonov interpreter. Interaktivno programiranje je korisno kada želimo da testiramo delove programskog koda ili iskaze.

U ovom poglavlju ćemo pogledati kako napisati kompletne Micropython programe i smestiti ih na našu razvojnu pločicu NodeMCU. Jednostavni programski primeri će biti dati u ovom poglavlju.

4.2 Upotreba 'ampy'

ampy (adafruit Mycropython Tool) je softverski alat koji se može koristiti da bi imali serijski interakciju sa pločicom ESP8266 uz pomoć Micropythona. 'Ampy' je jednostavan komandni alat koji se može upotrebiti za izvođenje sledećih operacija na Micropythonovoj pločici (pločici NodeMCU):

- Lista sadržaj direktorijuma na Micropythonovoj pločici (naredba: ls)
- Kreira direktorijum na Micropythonovoj pločici (naredba: mkdir)
- Dovodi fajl sa na Micropythonove ploče (naredba: get)
- Šalje fajl na Micropythonovu ploču (naredba: put)
- Uklanja direktorijum i njegov sadržaj sa Micropythonove pločice
- Izvršava skript na PC-ju i prikazuje njegov izlaz (naredba : run)
- Izvodi reset pločice (naredba: reset)

Ampy bi trebalo da bude instaliran na PC-ju pre nego počne da se koristi. Ovo zahteva prethodnu instalaciju Pythona 2.7.x ili 3.x koji bi trebalo da su instalirani na našem računaru. U ovom odeljku pretpostavljamo da Python 2.7 instaliran na PC-ju. (vidi odeljak 2.7)

Da bi se instalirao 'ampy' unesite sledeće naredbe sa komandne linije na vašem PC-ju (vidi sliku 4.1):

```
C:\> cd python27\scripts

C:\Python27\Scripts> pip install adafruit ampy
```

```
C:\>cd Python27\scripts

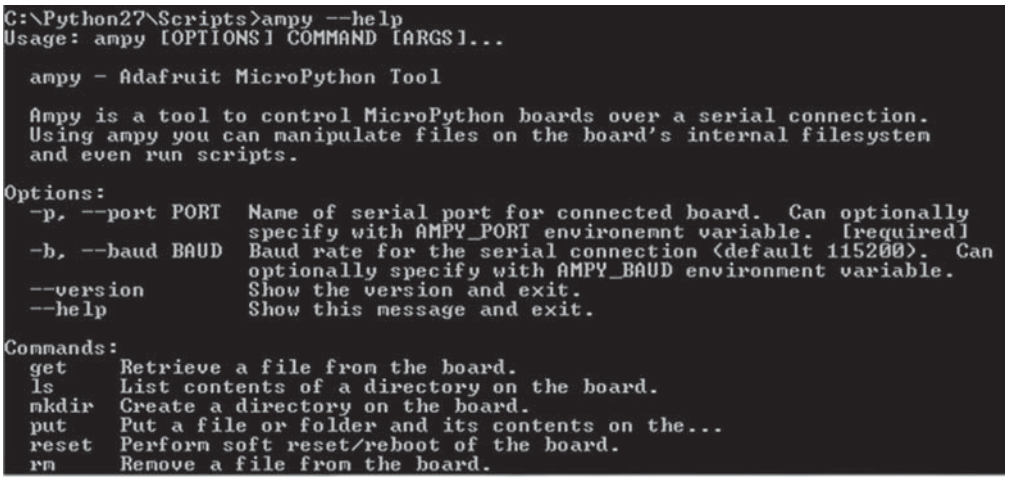
C:\Python27\Scripts>pip install adafruit-ampy
Collecting adafruit-ampy
  Downloading adafruit_ampy-1.0.1-py2.py3-none-any.whl
Collecting click (from adafruit-ampy)
  Downloading click-6.7-py2.py3-none-any.whl (71kB)
    100% |################################| 71kB 484kB/s
Requirement already satisfied: pyserial in c:\python27\lib\site-packages (from a
dafruit-ampy)
Installing collected packages: click, adafruit-ampy
Successfully installed adafruit-ampy-1.0.1 click-6.7

C:\Python27\Scripts>
```

Slika 4.1 Instalacija 'ampy'

Bilo da se 'ampy' instalirao uspešno ili ne unosom sledeće naredbe (vidi sliku 4.2) dobijate pomoć

C:\Python27\Scripts> ampy —help



Slika 4.2 Naredba pomoć u 'ampy'

4.2.1 Upotreba ampy - primer

Primer upotrebe ampy je dat dole. Pre upotrebe naredbi trebalo bi vezati vašu NodeMCU pločicu na PC i pronaći broj serijskog porta dodeljenog vašoj pločici.

Primer je dat na slici 4.3 koja lista fajlove na Micropythonovoj ploči:

C:\Python27\Scripts> ampy —port COM2 ls



Slika 4.3 Spisak fajlova na pločici

4.2.2 Kreiranje i izvršavanje programa

Upotrebom ampy-ja možemo pokrenuti Micropythonov program na našoj Micropython pločici. Program se može kreirati uz pomoć najčešće korišćenog tekst editora kao što je Notepad (ili čak ESPlorer). Primer programa kreiranog uz pomoć Notepad-a je dat na slici 4.4. Program se naziva **sinetable.py** i smešten je pod direktorijumu **C:\Python\Scripts**. Ovaj program prikazuje vrednosti trigonometrijskog sinusa za ugao od 0 do 90 stepeni u koracima od 5 stepeni.

Poglavlje 5 • Micropythonove hardverske biblioteke

5.1 Pregled

U poslednjem poglavlju smo videli kako pisati Micropython programe i kako iz izvršiti na razvojnoj pločici NodeMCU koja je snabdevena Micropython firmverom. Takođe je dato nekoliko primera Micropython programa da bi se pokazale različite osobine Micropython programa.

U ovom poglavlju ćemo pogledati module hardvera kao specifične Micropython biblioteke koje se mogu iskoristiti u Micropython projektima.

5.2 machine Library Module

Ovaj modul sadrži hardver koji se odnosi na odgovarajuće Micropython funkcije. Neke najčešće korišćene funkcije ovog modula su:

machine.reset(): Resetuje uređaj kao da je pritisnut taster RESET

machine.reset.cause(): Vraća uzrok reseta.
Na raspolaganju su sledeće konstante:

machine.PWRON.RESET
machine.HARD_RESET
machine.WDT_RESET
machine.DEEPSLEEP_RESET
machine.SOFT_RESET

machine.enable_irq(state): Ponovo dozvoljava zahtev za prekidom (interrupt). 'State' je vrednost vraćena sa poslednjeg poziva disable_irq

machine.disable_irq(): Onemogućava zahtev za prekidom

machine.freq(): Vraća učestanost CPU (u Hz)

machine .sleep(): Zaustavlja CPU i onemogućava svim periferijama izuzev WLAN-u. Izvori za buđenje moraju biti konfigurisani za ponovno buđenje.

machine.unique_id(): Vraća bajt string sa jedinstvenim identifikatorom pločice koja se koristi

5.3 machine Library Module Classes

Neke od obično korišćenih klasa koje podržava 'machine' biblioteka su:

Pin:	klasa ulazno - izlaznih izvoda
ADC:	klasa analogno digitalnih pretvarača
PWM:	izlaz širinske modulacije
I2C:	klasa protokola I2C

RTC: klasa časovnika realnog vremena
SD: klasa SD kartice
SPI: klasa SPI protokola
Timer: klasa internih tajmera
UART: klasa serijskog UART-a
WDT: klasa nadzorno vremenskog relea (watchdog)
DH11: klasa DHT DHT11 temperature i vlažnosti

5.3.1 Klase izvoda (pinova)

Objekt izvoda se upotrebljava ua upravljanje izvodima opšte namene (GPIO). Klasa pinova ima opšti oblik (samo prva dva parametra su obavezna a ostali nisu):

machine.Pin(id, mode, pull, value, drive, alt)

id: Može biti ceo broj pina ili naziv kao string

mode: Određuje način rada pina. Obično korišćeni su:

Pin.IN: izvod konfigurisan kao ulaz
Pin.OUT: izvod konfigurisan kao izlaz

pull: Određuje da li na izvodu postoji otpornik na napajanju (pull-up). Mogućnosti su:

Pin.None: nema otpornika prema napajanju ili masi
Pin.PULL_UP: dozvoljen otpornik na napajanju
Pin.PULL_DOWN: dozvoljen otpornik prema masi

value: Važi za Pin.OUT i određuje početno stanje na izvodu.

drive: Određuje izlaznu snagu (tekuću sposobnost pobudne struje) na pojedinim izvodu. Raspoložive mogućnosti su:

Pin.LOW_POWER
Pin.MED_POWER
Pin.HIGH_POWER

alt: Određuje alternativnu funkciju izvoda

Sledeći obično korišćeni metodi su:

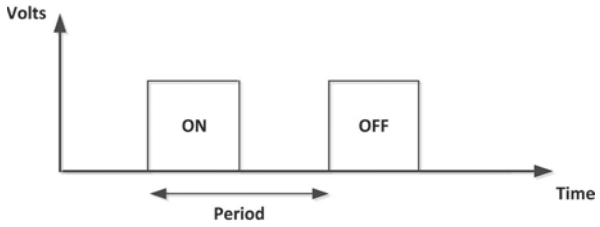
pin.value([x]): Dozvoljava zadavanje i dobijanje vrednosti pina. Ukoliko je pin konfigurisan kao ulazni pin (argument x je izostavljen) onda ovaj metod dobija digitalnu vrednost pina, vraćajući 0 ili 1. Ukoliko je pin konfigurisan kao izlaz onda vraćena vrednost nije definisana i port je postavljen na datu vrednost.

5.3.3 klasa PWM

Oblici signala širinske modulacije PWM se koriste u raznim aplikacijama upravljačkih sistema gde se zahteva promena nivoa analognog napona koji se dovodi eksternom uređaju. Slika 5.18 pokazuje tipičan oblik PWM koji je u osnovi povorka kvadratnih impulsa sa vremenima uključenja i isključenja (ON i OFF). Ova vremena u engleskom se respektivno nazivaju MARK i SPACE (kod nas respektivno IMPULS i PAUZA). Period talasnog oblika je zbir uključenog i isključenog vremena. Parametar odnosa uključeno/period (odnos signal/period - duty cycle) je važan parametar i definiše odnos vremena uključeno (ON) u odnosu na vreme perioda tj.:

odnos signal/period = uključeno (ON) / (uključeno (ON) + isključeno (OFF))

Ovaj odnos se obično navodi u procentima. Tako 0% odgovara vremenu ON a 100% odgovara vremenu OFF.



Slika 5.18 tipičan talasni oblik PWM-a

Oblik PWM-a se može dozvoliti na svim izvodima izuzev pina 16. Učestanost PWM-a može biti postavljena na bilo koju vrednost u opsegu od 1 do 1000 Hz a odnos može biti postavljen od 0 (0%) do 1023 (100%).

Sledeće metode su podržane za klasu PWM

PWM(Pin(n)): kreira PWM objekt na pinu n
pwm.freq(): daje tekuću učestanost PWM-a
pwm.freq(n): postavlja učestanost PWM na 'n'
pwm.duty(): daje tekuću vrednost PWM odnosa
pwm.duty(n): postavlja PWM odnos
pwm.deinit(): ne dozvoljava PWM

Dole dati primer pokazuje kako se PWM klasa koristi u programima

Primer 5.5
Napišite program koji generiše talasni oblik PWM na izvodu GPIO0 (D3) na pločici NodeMCU. Učestanost i odnos talasnog oblika bi trebalo da su respektivno 500Hz i 50%. Prikazati talasni oblik na osciloskopu.

Rešenje 5.5
Zahtevani program je prikazan na slici 5.19 (program:pwm.py). Program generiše PWM talasni oblik učestanosti 500Hz i sa odnosom impuls-period od 50% na izvodu GPIO0.

```
#-------------------------------------------------------------
#
#          PWM WAVEFORM
#          ============
#
# This program generates a PWM waveform on pin GPIO0 of the
# NodeMCU development board. The freuquency and duty cycle of
# the generated waveform is 500Hz and 50% respctively.
#
#
#
# Program :    PWM.py
# Date    :    April, 2017
# Programmer:  Dogan Ibrahim
#-------------------------------------------------------------
from machine import Pin, PWM

duty_cycle = 50
duty = int(duty_cycle * 1023 / 100)

pwm0 = PWM(Pin(0))
pwm0.freq(500)
pwm0.duty(duty)
```

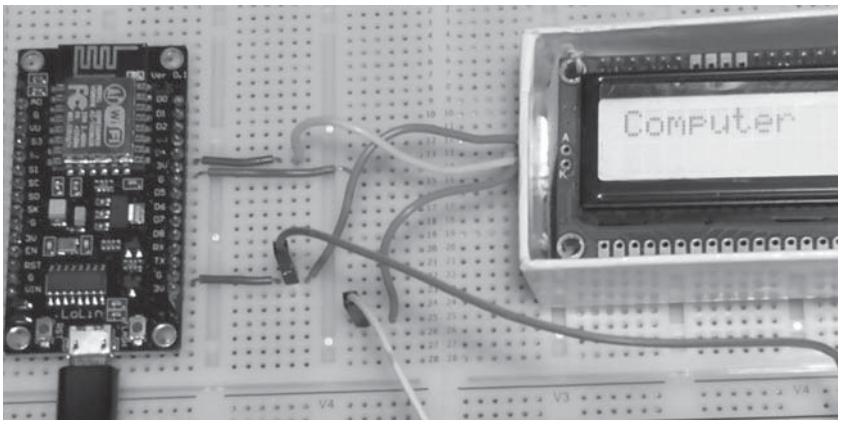
Slika 5.19 Listing programa za primer 5.5

Slika 5.20 pokazuje interaktivni rad programa uz pomoć softvera za emulaciju terminala.

```
>>> from machine import Pin, PWM
>>> duty_cycle = 50
>>> duty = int(duty_cycle * 1023 / 100)
>>> pwm0 = PWM(Pin(0))
>>> pwm0.freq(500)
>>> pwm0.duty(duty)
>>> _
```

Slika 5.20 Interaktivni rad sa programom

Generisani PWM talasni oblik je prikazan na PC Velleman osciloskopu PCS-GU250. Kao što je prikazano na slici učestanost je 500 Hz a odnos je 50%.



Slika 5.29 Kolo sa LCD na eksperimentalnoj pločici

5.3.7 DH11 klasa

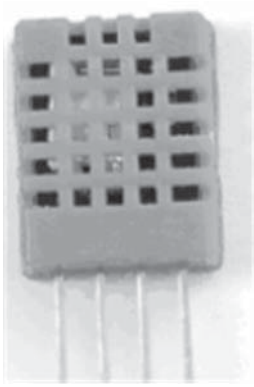
Klasa DH11 se koristi sa čipovima DHT11/22, senzorima za temperaturu i vlagu. U ovom projektu čip DHT11 se koristi da meri temperaturu i vlažnost okoline. DHT11 je manje tačan od DHT22 i radi u nižem opsegu ali je jeftiniji.

Sledeće metode su na raspolaganju sa klasom DHT11:

temperature(): vraća temperaturu u °C
humidity(): vraća vlažnost u procentima

DHT11 je čip sa digitalnim izlazom sa 4 izvoda (koriste se 3) kao na slici 5.30 sa izvodima +V, GND i podaci (data). Izvod ‘data‘ se mora vezati na napajanje (pull-up) na +V preko otpornika od 10kOhm. Čip koristi kapacitivni senzor vlage i termistor za merenje temperature okoline. Podaci na izlazu se pojavljuju svake sekunde. Osnovne osobine DHT11 su:

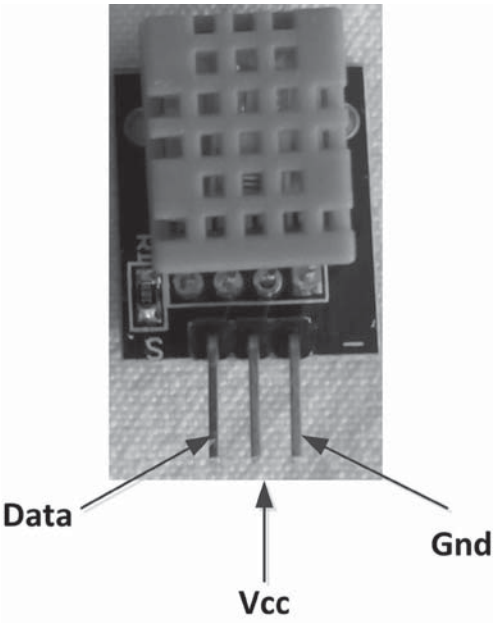
- Rad na 3 do 5 V
- Potrošnja struje je 2.5 mA (za vreme pretvaranja)
- Očitavanje je u opsegu od 0 do 50°C sa tačnosti od ±2°C
- Očitavanje vlažnosti je u opsegu od 20-80% sa 5% tačnosti
- Pločica mora biti kompatibilna sa razmakom otvora od 0.1 inč



Slika 5.30 Standardni čip DHT11

Donji primer pokazuje kako se DHT11 koristi u projektu

Primer 5.9
U ovom primeru koristimo modul DHT11 sa ugrađenim otpornikom od 10k koji smo nabavili kod Elektora. Ovo je čip sa 3 izvoda sa rasporedom pinova kao što je na slici 5.31.



Slika 5.31 Elektorov modul DHT11

Slika 5.32 prikazuje električnu šemu projekta. Ovde je izlaz podataka na DHT11 vezan na pin GPIO0 (D3= NodeMCU. Kolo na eksperimentalnoj pločici je prikazano na slici 5.33

Poglavlje 6 • Mycropython Wi-Fi programiranje na ESP8266

6.1 Pregled

U poslednjem poglavlju smo pogledali Mycropythonove funkcije koje se odnose na hardver i videli kako se koriste za kontrolu procesora baziranog na ESP8266. U ovom poglavlju ćemo pogledati kako konfigurisati razvojnu pločicu NodeMCU uz pomoć Mycropythonovog programskog jezika tako da možemo uraditi programiranje mreže i komunicirati sa ostalim čvorovima na mreži preko TCP/IP i UDP protokola.

6.2 Konekcija na Wi-Fi ruter - biblioteka mreže

Pre komunikacije sa drugim čvorovima mreže moramo uspostaviti sebe kao čvor na mreži i posedovati sopstvenu IP adresu.

Mycropythonova biblioteka se naziva network obezbeđuje funkcije koje se odnose na mrežu kao što je povezivanje na mrežu, pronalaženje IP adrese itd. Ova biblioteka mora biti uvezena u Mycropythonove programe pre korišćenja mrežnih funkcija.

Mrežna biblioteka ima klase koje se nazivaju WLAN. Neke najčešće korišćene metode ovih klasa su:

network.WLAN(network.STA_IF): kreira WLAN mrežni objekt koji se vezuje na Wi-Fi

network.WLAN(network.AP_IF): kreira WLAN mrežni objekt koji se vezuje na AP, dozvoljavajući ostalim Wi-Fi klijentima da se konektuju

active(True):	aktivira mrežni interfejs
active(False):	deaktivira mrežni interfejs
connect("ssid", "password")	konektuje se na određenu Wi-Fi mrežu
disconnect():	diskonektuje se sa mreže na koju je vezan
isconnect():	koristi se da nađe da li smo konektovani na Wi-Fi
ifconfig():	vraća adresu dodeljenu našoj konekciji

Primer na slici 6.1 u interaktivnom komandnom načinu rada koji uspostavlja konekciju preko Wi-Fi mreže određivanjem 'ssid'-ja (u ovom primeru BTIlub5-6SPN) i lozinke (u ovom primeru af4b5e6243) a zatim prikazuje dodeljenu IP adresu koja je, eto, 192.168.1.145.

```
>>> import network
>>> net = network.WLAN(network.STA_IF)
>>> net.isconnected()
False
>>> net.active(True)
>>> net.connect("BTHub5-6SPN", "af4b5e6243")
>>> net.isconnected()
True
>>> net.ifconfig()
('192.168.1.145', '255.255.255.0', '192.168.1.254', '192.168.1.254')
>>> _
```

Slika 6.1 Vezivanje na Wi-Fi mrežu

Sledeće Mycropythonove funkcije se mogu upotrebiti da bi se na Wi-Fi mrežu konektovali automatski, na primer za vreme podizanja sistema. Dodeljena IP adresa se prikazuje na kraju funkcije:

```
def Connect_WiFi():
import network
net=network.WLAN(network,STA_IF)
if not net.isconnected():
        net.active (True)
        net.connect("ssid, "Password")
print (netifconfig())
```

6.3 Korišćenje biblioteke 'Socket' - UDP i TCP/IP paketi

UDP i TCP/IP su dva najčešće korišćena protokola za slanje i prijem podatak preko mreže. TCP/IP je pouzdana protokol koji uključuje sporazumevanje (handshaking) i garantuje isporuku paketa na zahtevano odredište. UDP sa druge strane nije tako pouzdan ali je zato brži protokol.

Tabela 6.1 upoređuje UDP i TCP/IP tip komunikacije

TCP/IP	**UDP**
Protokol okrenut konekciji	Protokol bez konekcije
Spor	Brz
Veoma pouzdan prenos podataka	nije tako pouzdan
Paketi su poređani po redu	Nema reda paketa
20 bajta je veličina zaglavlja	Zaglavlje od 8 bajta
Provera grešaka i ponovljeni prenos	Nema provere grešaka
Potvrda prijema podataka	Bez potvrde prijema
Koristi se kod HTTP, HTTPS,FTP itd.	Korist se kod DNS, DHCP, TFTP itd.

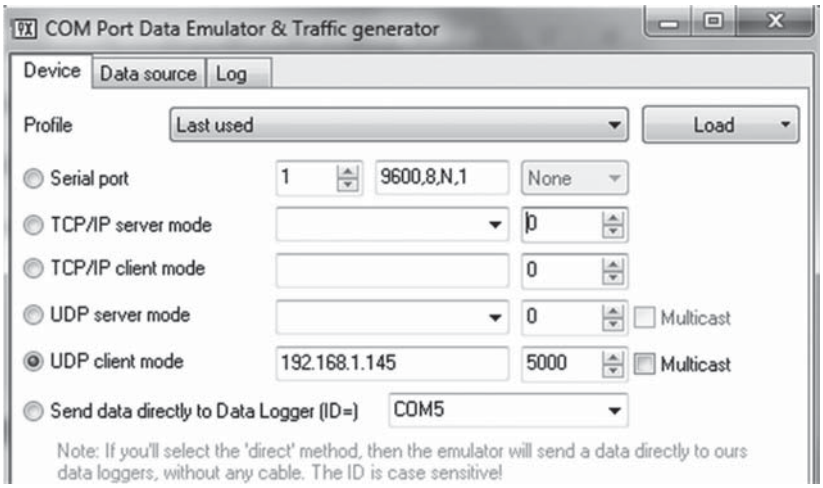
Tabela 6.1 Poređenje UDP i TCP/IP

Programi bazirani na UDP i TCP/IP protokolima su u osnovi server -klijent gde jedan čvor šalje podatke a drugi čvor prima i obratno. Podaci se prenose preko portova gde server i klijenti moraju da imaju iste brojeve porta.

Sada ćemo pogledati kako programi bazirani na UDP i TCP/IP protokolima mogu biti napisani uz pomoć Mycropythonovog programskog jezika. Biblioteka '**socket**' se koristi za razvoj programa baziranih na UDP i TCP/IP protokolima.

6.3.1 UDP programi

UDP je protokol bez konekcije i zato nema potrebe praviti konekciju prema odredišnom čvoru pre nego što se paket pošalje. Komunikacija između servera i klijenta je sledeća:



Slika 6.6 Ekran AGG softvera

6.3.2 TCP/IP programi

TCP/IP konekcija je sa pouzdanim protokolom. Primer dat u ovom odeljku prikazuje kako protokol može biti upotrebljen za razmenu podataka preko mreže.

Primer 6.2

Napišite TCP/IP program za prijem poruke sa TCP/IP klijenta i prikažite poruku. Zatim klijentu pošaljite poruku Hello From the Server. Pretpostavimo sledeće adrese i brojeve portova

Server(NodeMCU) IP adresa:	192.168.145	port:2000
Destination(PC) IP adresa:	192.168.1.132	port: 2000

Neka vaša razvojna pločica NodeMCU bude serverski čvor a vaš PC čvor klijenta. Prikažite primljenu poruku na ekranu.

Rešenje 6.2

Postavka sistema za ispitivanje je kao na slici 6.2. Ovde se opet koristi AGG softver ali ovog puta kao TCP/IP klijent.

TCP/IP program za NodeMCU je dat na slici 6.7 (program.tcpip.py) Na početku program se uvozi biblioteka 'socket', za TCP/IP soketa definiše (SOCK_STREAM), adresa servera je definisana funkcija **'bind'** se koristi da poveže adresu i port. Zatim se koristi funkcija **'listen'** da se čuje konekcija od strane klijenta. Konekcija se prihvata funkcijom **'accept'**. Funkcija **'recv'** prima podatke klijenta. Podaci se prikazuju i šalje poruka nazad klijentu pomoću funkcije **'sendall'**.

```
#-----------------------------------------------------------------------
#            TCP/IP PROGRAM
#            ==============
#
# This is a MicroPython TCP/IP program. The program sends the message
# “Hello From the Server” to a client. The IP addresses and ports
# of the server and client are:
#
#    Server IP address: 192.168.1.145  POrt: 2000
#    Client IP address: 192.168.1.132  Port: 2000
#
# The cleint sends the message “Hello From Client”
#
#
# Program :    tcpip.py
# Date    :    April, 2017
# Programmer:  Dogan Ibrahim
#-----------------------------------------------------------------------
import socket

Message = “Hello From the Server”

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = (“192.168.1.145”, 10000)
sock.bind(server_address)
sock.listen(1)
conn,cient_address = sock.accept()
#
# Receive data from the client and display it
#
data = conn.recv(1024)
print (data)

#
# Send data to the client
#
conn.sendall(Message)
#
# Close the socket
#
conn.close()
```

Slika 6.7 Listing programa za primer 6.2

Primljeni i prikazani podaci od strane klijenta (AGG softver) su prikazani na slici 6.8

Poglavlje 7 • Projekt - Šta danas da obučem?

7.1 Pregled

Svi želimo da znamo kakvo će vreme biti pre nego što se ujutro obučemo. Ukoliko će biti kiše možda će nam biti potreban kišni mantil i trebalo bi poneti kišobran, ukoliko je hladno trebalo bi obući topliju odeću, a ako će biti topao ili vreo dan možda bi obukli laku letnju odeću. Obično prvo proverimo prognozu ili pogledamo u nebo i zatim odlučimo šta ćemo obući.

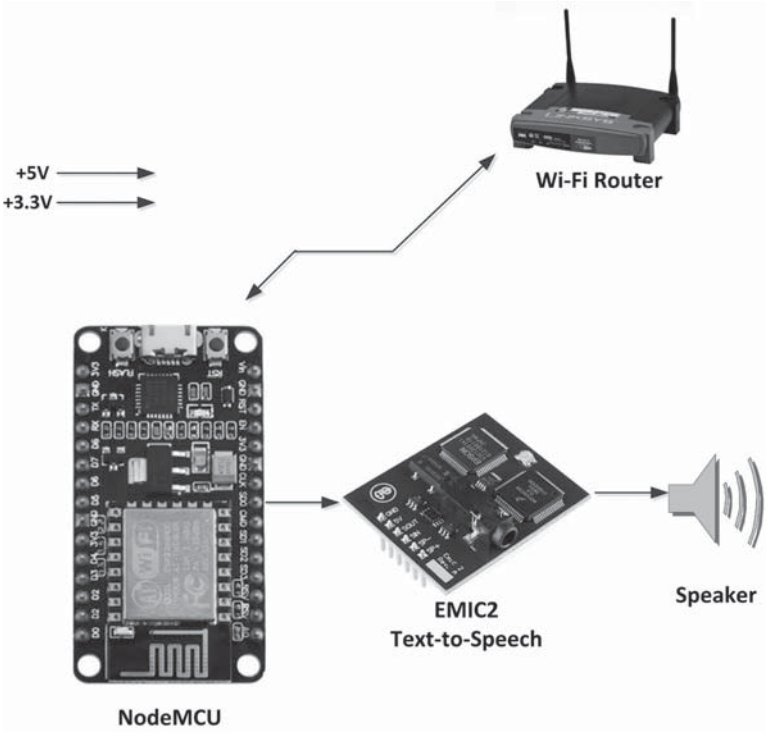
U ovom projektu ćemo razvijati projekt na NodeMCU razvojnoj pločici sa procesorskim modulom za pretvaranje teksta u govor (Text-to-Speach). Pretpostavićemo da će se sistem aktivirati automatski čim otvorimo vrata garderobe pre nego što odlučimo šta bi obukli tog dana. Realni rad sistema bi išao ovako:

Kada se otvori garderoba prekidač se uključuje i napaja sistem. Kada se aktivira sistem će dobiti lokalnu prognozu uz pomoć HTTP upita preko našeg kućnog rutera. Zatim na osnovu vremenske prognoze sistem će izgovoriti preko zvučnika razne vremenske parametre kao što su temperatura, vlažnost, vetar, pritisak i opštu vremensku prognozu. Sistem će takođe odlučiti i reći vam preporuku šta obući tog dana prema lokalnim vremenskim prilikama.

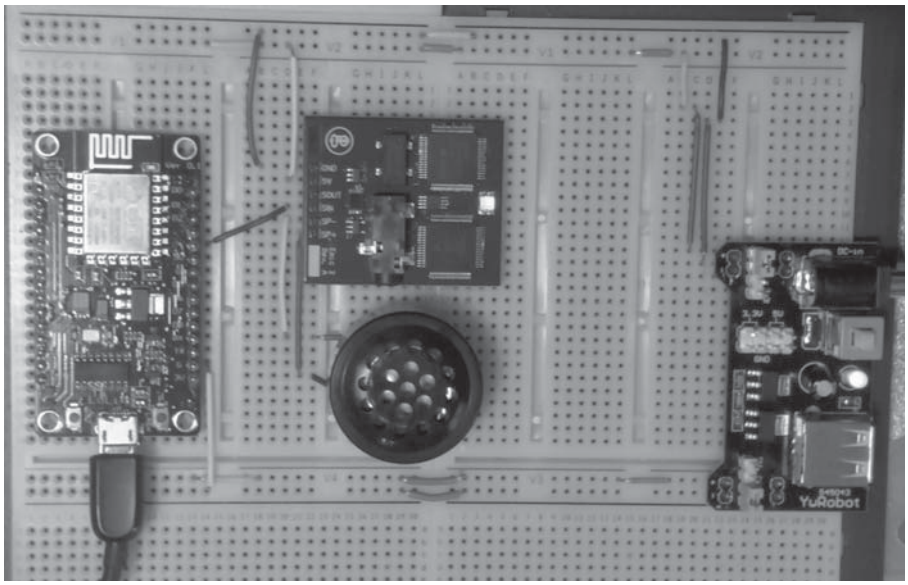
7.2 Blok dijagram

Blok dijagram razvijenog sistema je prikazan na slici 7.1. Razvijeni sistem je napravljen na eksperimentalnoj pločici i prikazan na slici 7.2 Sistem se u osnovi sastoji od:

- Razvojne pločice NodeMCU
- Govornog procesora EMIC2
- Zvučnika
- Napajanja od +5 V i 3.3 V



Slika 7.1 Blok dijagram razvojnog sistema



Slika 7.2 Napravljeni sistem na eksperimentalnoj pločici

Poglavlje 8 • Projekat - Temperatura i vlažnost smešteni na oblak (cloud)

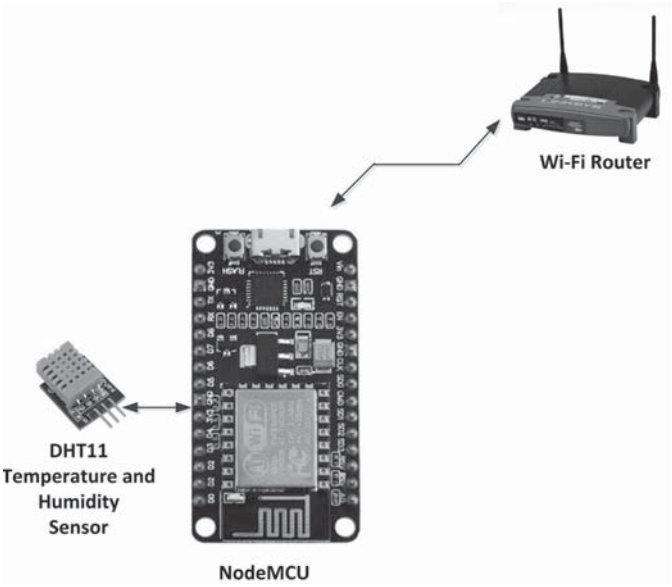
8.1 Pregled

U poslednjem poglavlju videli smo dizajn i razvoj zanimljivog MicroPython bazirane aplikacije uz pomoć razvojne pločice NodeMCU sa modulom za govor. Sistem se automatski aktivirao kad bi otvorili vrata garderobe.

U ovom poglavlju ćemo razviti drugu zanimljivu aplikaciju. Ovde će sistem dobiti temperaturu i vlažnost uz pomoć senzora i smestiti podatke na oblak tako da budu dostupni sa bilo kog mesta.

8.2 Blok dijagram

Blok dijagram sistema je dat na slici 8.1. U ovom projektu se upotrebljava čip senzora temperature i vlage DHT11 (vidi poglavlje 5). Ovaj čip normalno zahteva otpornik na napajanju ali kao što je prikazano na slici 5.31 Elektor prodaje čipove sa ugrađenim otpornikom od 10k.



Slika 8.1 Blok dijagram projekta

Električna šema projekta je data na slici 5.32. Izlaz podataka sa DHT11 se vezuje na izvod GPIO0(D3) NodeMCU. Kolo je napravljeno na eksperimentalnoj pločici kao na slici 5.33.

8.3 Čitanje temperature i vlažnosti

Programski kod koji čita temperaturu i vlagu je napisan kao funkcija i prikazan na slici 8.2. Funkcija se naziva **TempHum** i vraća temperaturu i vlažnost u glavni program.

```
def TempHun():
    import dht
    from machine import Pin
    d = dht.DHT11(Pin(0))
    d.measure()
    t = d.temperature()
    h = d.humidity()
    return t, h
```

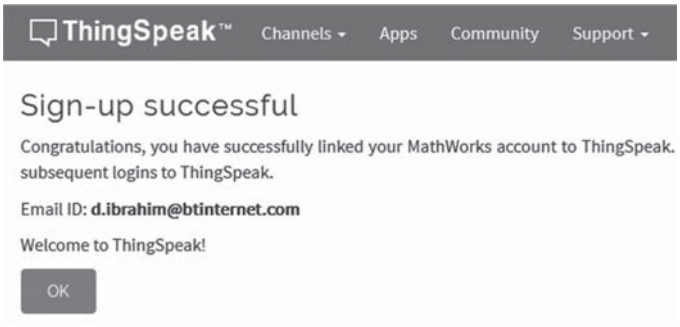
Slika 8.2 Funkcija čita i vraća temperaturu i vlažnost

8.4 Oblak podataka

Postoji nekoliko besplatnih uslužnih oblaka podataka koji se mogu iskoristiti za smeštaj podataka (na primer **Spark-Fun, Thingspeak** itd). U ovom projektu se koristi **Thingspeak**. Ovo je besplatan oblak gde se podaci sa senzora mogu smestiti i preuzeti uz pomoć jednostavnog HTTP upita. Pre upotrebe Thingspeak-a potrebno je napraviti nalog sa njihovog veb sajta i zatim se prijaviti na taj nalog. Novi nalog napravite davanjem adrese elektronske pošte i izborom lozinke otvaranjem sledećeg linka:

https://thingspeak.com/users/sign_up

Trebalo bi da dobijte e-mail sa potvrdom da možete da aktivirate vaš nalog. Posle toga kliknite Continue i trebalo bi da dobijete poruku o uspešnom upisu kao na slici 8.3 pa bi trebalo da se složite sa postavljenim uslovima (agree to the conditions).



Slika 8.3 Uspešan upis u Thingspeak

Zatim bi trebalo kreirati novi kanal klikom na **New Channel.** Ispunite formular kao što je na slici 8.4. Aplikaciji dajte naziv **MyWeather** i kreirajte dva kanala pod nazivima **Temperature** i **Humidity**, označite da kanali budu javni.

Poglavlje 9 • Projekat Daljinsko upravljane preko veba

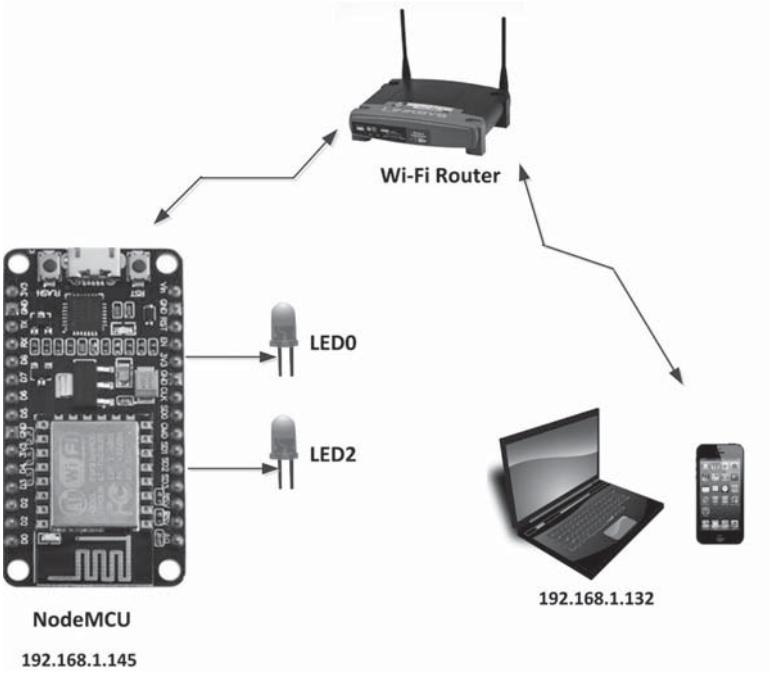
9.1 Pregled

U poslednjem poglavlju smo videli dizajn i razvoj zanimljive MicroPython Wi-Fi aplikacije koja uzima temperaturu i vlažnost okoline i smešta ih na oblak.

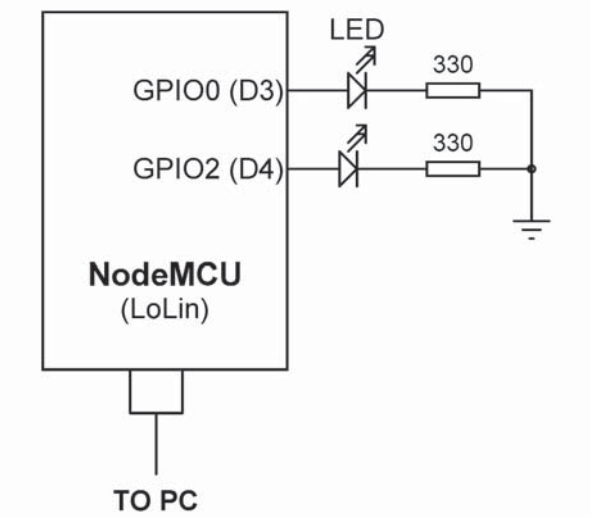
U ovom poglavlju ćemo razviti drugu zanimljivu aplikaciju. Ovde će sistem daljinski upravljati sa dve LED vezane na razvojnu pločicu NodeMCU uz pomoć HTTP veb server aplikacije. Iako se u ovom primeru koriste dve LED u opštem slučaju mogu biti vezana relea na NodeMCU ili bilo koji tip elektroopreme kojom se želi upravljati sa udaljenog mesta.

9.2 Blok dijagram

Blok dijagram sistema je prikazan na slici 9.1. Slika 9.2 prikazuje električnu šemu projekta. LED0 i LED2 su respektivno vezane na portove NodeMCU i to pin0 (označen sa D3) i 2 (oznaka D4) preko otpornika za ograničenje struje od 330 Ohm. LED se uključuju kad je izlaz porta na logičkoj 1 (high). Isključuju se kad je izlaz porta na logičkoj 0 (low).



Slika 9.1 Blok dijagram sistema



Slika 9.2 Električna šema sistema

9.3 HTTP veb server/klijent

Veb server je program koji koristi HTTP (Hyper Text Transfer Protocol) da služi veb stranice korisnicima kao odgovor na njihov upit. Ovi upiti se dalje šalju HTTP klijentima. Korišćenjem HTTP para server/klijent možemo kontrolisati bilo koji uređaj vezan na procesor veb servera preko veba.

Slika 9.3 pokazuje strukturu postavke veb server/klijent. na ovoj slici je NodeMCU veb server a PC (ili laptop, tablet ili mobilni telefon) je veb klijent. Uređaj koji se kontroliše je vezan na razvojnu pločicu NodeMCU. Rad sistema se obavlja ovako:

- Veb server je u režimu slušanja, očekujući upite sa veb klijenta
- Veb klijent pravi upit veb serveru šaljući HTTP upit
- Kao odgovor veb server šalje HTTP kod veb klijentu koji se aktivira u veb brauzeru od strane korisnika na veb klijentu i pokazuje kao formular na ekranu veb klijenta.
- Korisnik šalje naredbu (na primer dodiruje taster u formularu veb klijenta da bi uključio LED) i zatim šalje kod veb serveru tako da veb server može izvršiti zahtevanu operaciju.

Prilog A • Sabirnica SPI

A.1 Pregled

SPI sabirnica se obično upotrebljava u projektima baziranim na mikrokontrolerima. U ovom prilogu ćemo pogledati upotrebu ove sabirnice sa jednostavnim primerima koristeći MicroPython. Cilj je čitaoca upoznati sa funkcijama SPI biblioteke MicroPython-a i pokazati kako se može koristiti u stvarnim projektima.

A.2 SPI sabirnica

SPI (Serial Peripheral Interface) sabirnica je jedna od najčešće korišćenih mikrokontrolerskih protokola za komunikaciju sa eksternim uređajima kao što su senzori i aktuatori. SPI sabirnica je tipa master slejv gde postoji jedan master a može biti više slejvova.

SPI se sastoji od 4 provodnika:

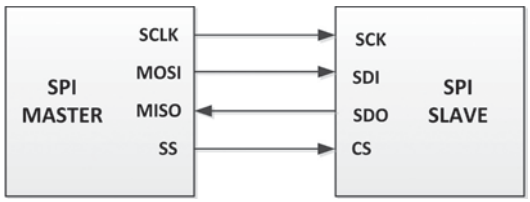
Master:

- **MISO** Master In Slave Out
- **MOSI** Master Out Slave In
- **SCLK** Serijski takt (clock)
- **SS** Izbor slejva (select)

Slejv:

- **SDI** - Slave Data In (ili MOSI)
- **SDO** - Slave Data Out (ili MISO)
- **SCK** - Serijski takt (clock) (ili SCLK)
- **CS** - Izbor čipa (chip Select)

Slika A.1 prikazuje vezu mastera i slejva na SPI sabirnici. Komunikaciju kontroliše master takt koji je obično nekoliko MHz. Master bira slejv uređaj i šalje mu podatke u serijskom obliku gde se bit šalje sa svakim signalom takta. Master šalje bit sa njegovog MOSI linije i slejv čita taj bit. Takođe slejv šalje bit na svoju MISO liniju a master čita. Prenos podataka na SPI sabirnici je urađen uz pomoć dva pomeračka registra, obično širine 8 bita, po jedan u masteru i jedan u slejvu. Podaci se pomeraju tako da izlazi prvo najznačajniji bit (MSB). Posle pomeranja (izlaza) svih bitova dolazi do razmene podataka. Ukoliko je potrebno poslati još podataka napuni se pomerački registar predajnika i podaci se pomeraju na izlazu. Razmena podataka se zaustavlja kada master ukloni takt.



Slika A.1 Veza mastera i slejva preko SPI sabirnice

Prilog B • sabirnica I2C

8.1 Pregled

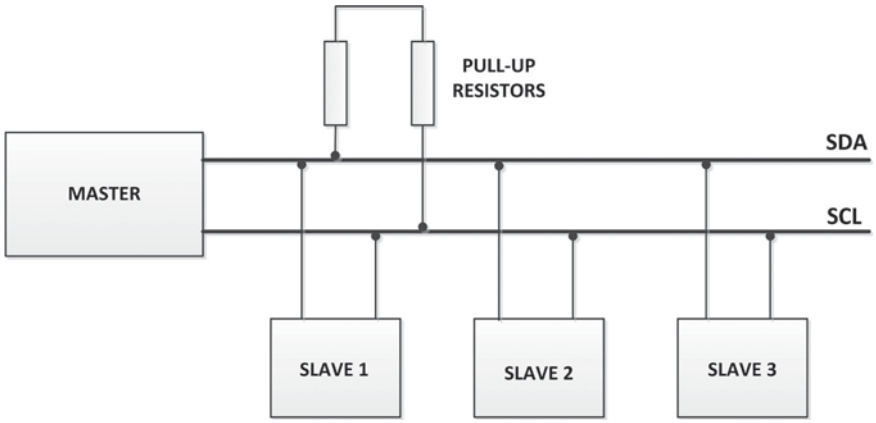
Sabirnica I2C (ili I^2C) se najčešće upotrebljava u projektima zasnovanim na mikrokontrolerima. U ovom prilogu ćemo pogledati upotrebu ove sabirnice na jednostavnim primerima uz pomoć MicroPython-a. Cilj je čitaoca naučiti funkcije MicroPython biblioteke I2C sabirnice i pokazati kako se koristi u stvarnim projektima.

8.2 Sabirnica I2C

Sabirnica I2C je jedna od najčešće korišćenih mikrokontrolerskih protokola za komunikaciju sa eksternim uređajima kao što su senzori i aktuatori. Sabirnica I2C je sa jednim masterom i više slejvova i može raditi do 50 MHz. Sabirnica se sastoji od dva voda sa otvorenim 'drain'-om i 'pull-up' otpornicima:

SDA: vod podataka
SCL: vod takta

Slika B.1 prikazuje strukturu sabirnice I2C sa jednim masterom i tri slejva



Slika B.1 Sabirnica I2C sa jednim masterom i tri slejva

Komunikaciju započinje (inicira) master koji šalje START bit. Ovim se upozoravaju svi slejvovi da neki podaci dolaze na sabirnicu i svi slejvovi osluškuju sabirnicu. Posle početnog bita šalju se 7 bita jedinstvene adrese slejva. Svaki slejv uređaj na sabirnici ima sopstvenu adresu i ovim se obezbeđuje da samo adresirani slejv komunicirana na sabirnici u to vreme da bi se izbegla kolizija. Poslednji poslati bit je 'read/write' bit koji ukoliko je 0 znači da master želi da piše po sabirnici (na registar slejva) a ukoliko je 1 znači da master želi da čita sa sabirnice (sa registra slejva). Podaci se šalju na sabirnicu prvo sa MSB (najznačajnijim bajtom).

8.3 Naredbe MicroPython-a na I2C sabirnici

Funkcije za MicroPython-ov ESP8266 jezik su implementirane u softver i češće se koriste sledeće funkcije:

Prilog C • Node-RED

Ovaj prilog obezbeđuje veoma kratak uvod u Node-RED. Detaljna upotreba i opis Node-RED su izvan područja ove knjige i ovde je dat samo uvod da bi čitaoce upozorio na ovu alatku. Zainteresovani čitaoci mogu pronaći detaljne informacije o Node-RED na sledećem veb sajtu i mnogim drugim sajtovima na internetu:

noderedguide.com

Node-RED je programska alatka sa vizuelnim prikazom gde razvojni inženjeri ubacuju predefinisane blokove koda (ili čvorove- nodes) da bi se obavio dati zadatak. Vezani 'nodovi' čine tok a tok se može sastojati od ulaznih nodova, procesnih nodova i izlaznih nodova. Ideja sa Node-RED je da se različit hardver vezuje bez programiranja. Node-RED je u početku razvio IBM za pravljenje aplikacije interneta stvari (IoT) sa idejom pojednostavljenja zadatka povezivanja hardvera da bi se obavio zadatak. Iako je napravljen za IoT aplikacije danas Node-RED se široko koristi za zadatke upravljanja i nadzora.

U aplikaciji Node-RED postoje u osnovi tri tipa 'nodova': ulazni, izlazni i procesni nodovi. Ulaznim nodovima ubacujemo podatke u tok. Ovi nodovi imaju najmanje jedan izlaz predstavljen malim sivim kvadratom na desnoj strani. Ulazni nodovi se mogu upotrebiti za dobijanje podataka sa drugih servisa kao što je Google, Twitter, TCP čvorova itd. Neki ulazni nodovi imaju tastere kojima možemo aktivirati nod. Neki tipični nodovi su:

- inject
- mqtt
- http
- serial
- tcp
- udp

Izlazni nodovi nam omogućuju da šaljemo podatke sa toka Node-RED. Ovi nodovi imaju jedan ulaz sa leve strane i mogu slati podatke drugim servisima kao što su Google, Twitter itd. Neki izlazni nodovi imaju tastere kojima možemo uključiti ili isključiti nod. Tipični izlazni nodovi su:

- debug
- http response
- serial
- tcp
- udp

Procesni nodovi se koriste za procesiranje podataka. Oni imaju jedan ulaz i jedan ili više izlaznih tačaka. Procesni nod može transformisati ulazne podatke i pretvoriti ih u drugi format, mogu aktivirati poruku itd. Neki tipični procesni nodovi su: