Wi-Fi modul ESP32

Mihelčić, Branimir

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:259947

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Wi-Fi MODUL ESP32

Završni rad

Branimir Mihelčić

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 06.09.2018.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Branimir Mihelčić
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	4086, 26.09.2017.
OIB studenta:	94395637521
Mentor:	Doc.dr.sc. Ivan Aleksi
Sumentor:	
Sumentor iz tvrtke:	David Zovko
Naslov završnog rada:	Wi-Fi modul ESP32
Znanstvena grana rada:	**Procesno računarstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	06.09.2018.
Datum potvrde ocjene Odbora:	17.09.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT

IZJAVA O ORIGINALNOSTI RADA

Osijek, 17.09.2018.

Ime i prezime studenta:	Branimir Mihelčić
Studij:	Preddiplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	4086, 26.09.2017.
Ephorus podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom: **Wi-Fi modul ESP32**

izrađen pod vodstvom mentora Doc.dr.sc. Ivan Aleksi

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Branimir Mihelčić

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PRIMIJENJENE TEHNOLOGIJE I ALATI 2

2.1. Arduino razvojno okruženje 2

2.2. Arduino Uno razvojni sustav 3

2.3. ESP32 sustav na čipu 4

2.4. W25Q32JV flash memorija 5

2.5. SE5218 linearni naponski regulator 5

2.6. CP2102 most 5

3. REALIZACIJA SUSTAVA 6

3.1. Elektronička shema sustava 6

3.2. Izrada makete sustava 8

3.3. Programsko rješenje sustava 12

3.3.1. Skeniranje WiFi mreža 12

3.3.2. Senzor dodira 13

3.3.3. WiFi server 14

4. ZAKLJUČAK 16

LITERATURA 17

SAŽETAK 18

ABSTRACT 19

ŽIVOTOPIS 20

PRILOG A. Program za ESP32 razvojni sustav „WiFi Scan“ 21

PRILOG B. Programski kod za senzor dodira 22

PRILOG C. Program za ESP32 razvojni sustav „WiFi Server“ 23

1. UVOD

Svijet teži stvaranju boljih uvjeta za život i olakšanoj svakodnevici i ljudi najprije žele ostvariti to u svojoj neposrednoj blizini, odnosno domu. Svima je jedan od primarnih ciljeva uštedjeti vrijeme koje gubimo na svakodnevne poslove, a jedan od načina je povezivanje kućanskih uređaja na Internet te njihovo upravljanje pomoću pametnih telefona. Takozvani Internet stvari (engl. *Internet of Things*) svijet svakim danom sve je veći i ESP32 čip za sada je jedan od najrazvijenijih na tržištu. Razvojem sustava s jednim ovakvim čipom otvaraju se vrata novim idejama te se omogućava stvaranje učinkovitijih i naprednijih tehnoloških naprava.

Cilj ovog završnog rada je dizajnirati i proizvesti pločicu baziranu na ESP32 integriranom krugu sa svim relevantnim ulazima i izlazima izvedenim na način da se eksperimentiranje jednostavno može izvoditi na eksperimentalnoj pločici. Također, na pločici je ugrađen i punjač za litij-ionske, odnosno litij-polimerske baterije te je iste moguće koristiti kao izvor napajanja za samostalan, energetski neovisan rad.

Od konačnog sustava se očekuje normalan i ispravan rad svih njegovih funkcija te mogućnost ostvarivanja punog potencijala ESP32 mikroupravljača. To znači da cijeli dizajnerski i proizvodni rad u procesu realizacije konačnog sustava treba biti pribran, točan i precizan.

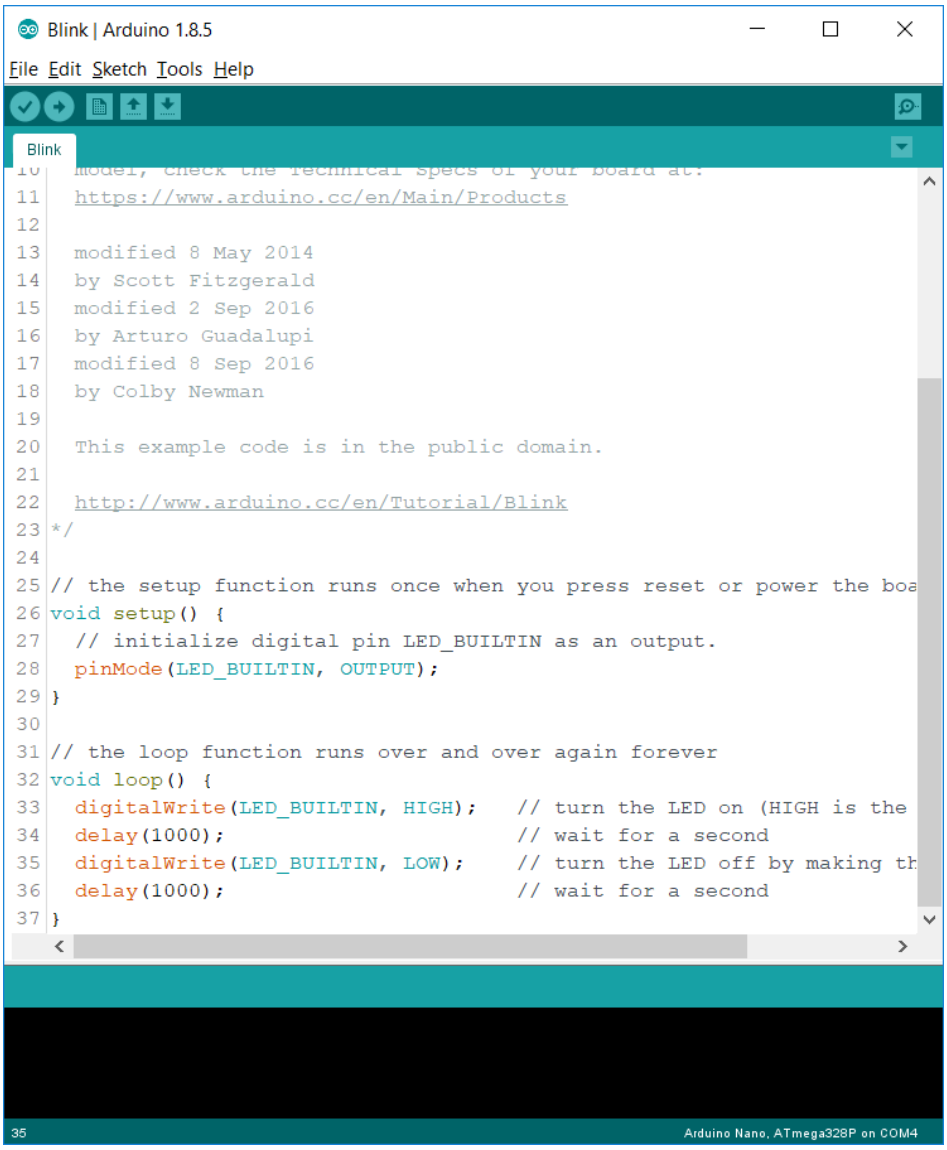
1.1. Zadatak završnog rada

U ovom radu je potrebno razviti tiskanu elektroničku pločicu za ESP32 čip koji se spaja WiFi vezom na Internet. Potrebno je razviti tiskanu elektroničku pločicu u programu Eagle. Modul je potrebno testirati povezivanjem s Arduino pločicom.

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

2.1. Arduino razvojno okruženje

Arduino integrirano razvojno okruženje (engl. *Arduino IDE*) besplatno je i javno dostupno korisničko sučelje za pisanje i implementiranje programske podrške u Arduino mikrokontrolere i ostale Arduino kompatibilne pločice. Kao što je navedeno u [1] i [2], ovaj IDE razvijen je u programskom jeziku Java te ima osnovne značajke obradnika znakova poput naglašavanja sintakse nekog programskog jezika (bojanje teksta), usklađivanje lijevih i desnih margina, preslikavanje i ljepljenje znakova, pretraga, zamjena i slično. Od ostalih značajki osim tekstualnog obradnika, arduino razvojno okruženje sadrži i prostor za poruke i obavijesti, alatnu traku s tipkama za uobičajene funkcije te niz izbornika.



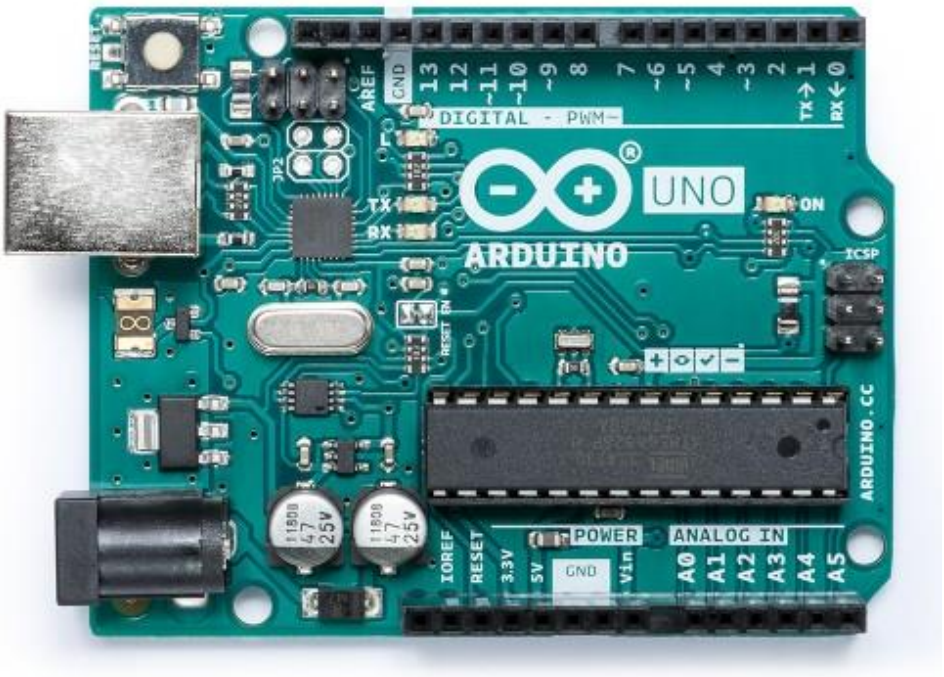
Sl. 2.1. *Arduino razvojno okruženje (IDE)*

Programi napisani u Arduino IDE-u nazivaju se skice. Zahvaljujući navedenim karakteristikama razvojnog okružja, korisnik ima mogućnost pisanja programskog koda poput naizgled jednostavnog teksta koji kasnije, klikom na tipku „*Verify*“, program prevoditelj (engl. *compiler*) pretvara iz programskog jezika u binarni strojni jezik. Skice se klikom na tipku „*Upload*“ također prevode zatim prenose na konfiguriranu pločicu uslijed čega će se u prozoru za poruke prikazati statusne obavijesti i povratne informacije ili moguće pogreške.

Budući da cjelokupna „priča“ oko Arduina pripada takozvanom konceptu „Otvorenog izvora“ (engl. *Open source*) [3] velika većina programskih rješenja dostupna je javnosti na uvid, korištenje ili eventualne izmjene tako da se u Arduino razvojnom okružju mogu naći i brojni primjeri programskih kodova.

2.2. Arduino Uno razvojni sustav

Arduino UNO ime je razvojne pločice temeljene na ATmega328P mikroupravljaču na kojoj se nalaze dodatni elementi za jednostavniju upotrebu i brže postavljanje. Na pločici se nalazi USB priključak preko kojeg se UNO spaja izravno na računalo kao i čip koji omogućava komunikaciju mikroupravljača i računala, odnosno pretvara diferencijalne USB (engl. *Universal Serial Bus*) signale u asinkrone UART (engl. *Universal Asynchronous Receiver-Transmitter*) signale i obratno. Nadalje, napajanje Arduino Uno-a moguće je upravo preko USB kabela ili vanjskim izvorom poput baterije. Ukoliko se korisnik odluči za napajanje preko vanjskog izvora trebao bi svakako pripaziti na ulazni napon napajanja. Na Arduino Uno razvojnoj pločici nalazi se linearni naponski regulator koji će ulazni napon od 6 do 20 V spustiti na 5 V potrebnih za normalan rad mikrokontrolera, međutim preporuča se da za najoptimalniji rad ulazni napon bude u rasponu od 7-12 V. [4]



Sl. 2.2. *Arduino Uno, Izvor: store.arduino.cc*

Razlog zašto je Arduino tako zanimljiv i prikladan za razne sustave automatizacije je mogućnost povezivanja mnoštva drugih senzora i aktuatora na 14 digitalnih i 6 analognih ulaza/izlaza (engl. GPIO – *general-purpose input/output*) od kojih neki imaju i posebne funkcije poput *SPI* i *I2C* protokola, pulsno širinske modulacije (*PWM*), serijske komunikacije te prekida (engl. *interrupts*).

2.3. ESP32 sustav na čipu

Prema [5], ESP32 je serija mikroupravljača razvijena od strane tvrtke Espressif Systems, s ugrađenim *WiFi* i *Bluetooth* mogućnostima. Postoje jednojezgrene i dvojezgrene varijacije s ugrađenom antenom, RF prilagodnim sklopom, pojačalom snage, filterima, upraviteljima snage. O tome koliko je ESP32 mikroupravljač visokotehnološki razvijen govore i njegove značajke navedene pod [6]. Sadrži 32 bitni procesor koji radi na 160 ili 240 MHz, što je za ovu kategoriju mikroupravljača poprilično napredno, deset puta brže od navedenog Arduino Uno-a koji radi na frekvenciji od 16 MHz. Nadalje ESP32 ima 520 KiB radne memorije, 12-bitne analogno-digitalne pretvarače, dva 8-bitna digitalno-analogna pretvarača, deset senzora dodira (kapacitivnost), senzor temperature, četiri serijska periferna sučelja (engl. *SPI*), dva *I2C* sučelja, tri *UART* sučelja, *PWM*, Hallov senzor (engl. *Hall effect sensor*) i drugo. ESP32 je dizajniran za mobilne, nosive elektroničke naprave te *Internet of Things* primjene. Ima sve značajke najsuvremenijeg tehnološkog dostignuća u području elektroničkih čipova niske potrošnje, uključujući više različitih

postavki potrošnje. ESP32 se također može „buditi“ periodički ili kada se ostvari određeni uvjet što će isto smanjiti potrošnju energije.

2.4. W25Q32JV flash memorija

Budući da neka programska rješenja zahtijevaju više memorije za pohranu, ESP32 mikroupravljaču je dodana i vanjska *flash* memorija. Za to je upotrebljen W25Q32JV čip. Prema [7], W25Q32JV čip je serijska *flash* memorija koja pruža rješenja za pohranu podataka za sustave s ograničenim resursima. 25Q serija pruža prilagodljivost i rad iznad mogućnosti običnih serijskih *flash* uređaja. Idealni su za spremanje zvukovnih i tekstualnih zapisa.

W25Q32JV niz je organiziran u 16384 programabilnih stranica, od kojih svaka ima 256 bajta prostora. U isto vrijeme može biti programirano do 256 bajta. Stranice mogu biti izbrisane u grupama od 16 (4 KB), od 128 (32 KB), od 256 (64 KB) ili cijeli čip.

Čip komunicira preko SPI protokola, dakle ima 4 signala: Radni takt (CLK), odabir čipa (/CS), serijske ulazne podatke (DI) i serijske izlazne podatke (DO). ESP32 mikroupravljač ima predodređene nožice za komunikaciju s takvim čipom te je time i njihovo povezivanje omogućeno.

2.5. SE5218 linearni naponski regulator

Prema [8], SE5218 serija linearnih naponskih regulatora sa stabilnim izlaznim naponom i malim padom napona dizajnirana je za prijenosne uređaje, uobičajeno napajane baterijom koji zahtjevaju okolinu s malim šumom, dovoljno brz odaziv te nizak pad napona regulatora. Ovi regulatori imaju referentnu naponsku razinu, PMOS tranzistor, otpornike za postavljanje izlaznog napona te zaštitne krugove koji ograničavaju struju i temperaturu.

Neke od značajki ovih regulatora su: vrlo mal pad napona od 175 mV pri 150 mA, precizan izlazni napon koji je u rasponu od 2% od nazivnog, brz odaziv 20 µs, termalna zaštita.

2.6. CP2102 most

Računalo komunicira koristeći diferencijalne USB signale, dok ESP32 mikroupravljač za komunikaciju koristi asinkrone UART signale. Kako bi komunikacija mikroupravljača i računala bila izvediva, potreban je element koji će biti „prevoditelj“ između ova dva sugovornika. Za to nam služi CP2102 USB na UART most. Pod [9] vidimo neke od značajki ovog čipa poput, integriranog oscilatora, regulator ulaznog napona, integrirani USB prijemnik i predajnik, brzine do 12 Mb/s i drugo.

3. REALIZACIJA SUSTAVA

3.1. Elektronička shema sustava

Elektronička shema ESP32 razvojne pločice u potpunosti je izrađena u programu „EAGLE“ tvrtke Autodesk Inc. Na slici 3.1. vidi se cjelokupna elektronička shema sustava sa svim najvažnijim sastavnicama realizacije sustava te dodatnim pasivnim komponentama bitnim za normalan rad. Shema je podijeljena u devet cjelina na način da svaka cjelina obavlja određenu funkciju.

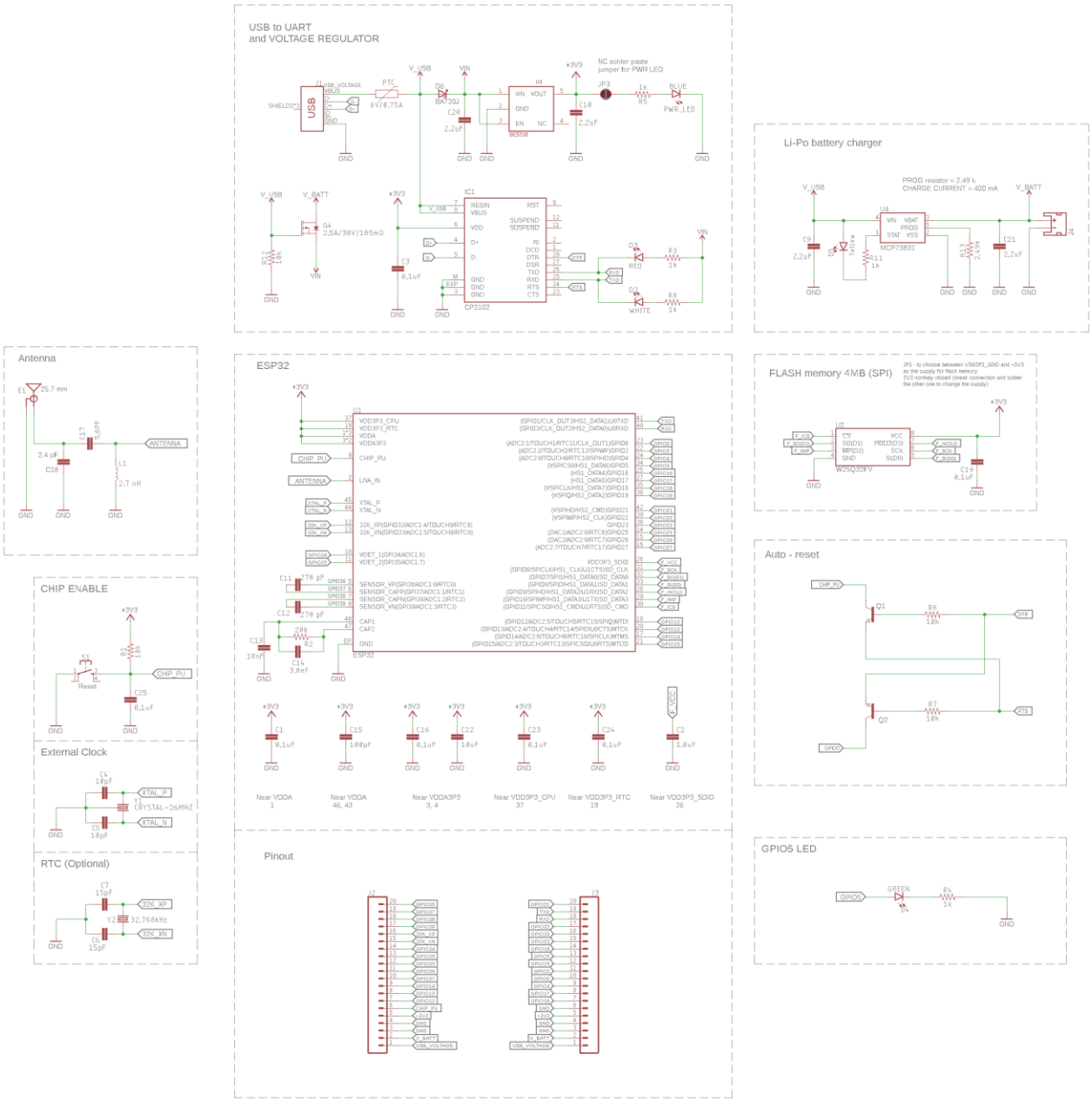
Središnji i najvažniji dio sustava čini ESP32, mikroupravljač opisan u potpoglavlju 2.3., a pored njega su smješteni neki od kondenzatora za smanjenje šuma. Iznad mikroupravljača nalazi se shema ulaznog sklopa, odnosno USB priključak, naponski regulator te USB na UART most. Zadaća naponskog regulatora je spustiti ulaznih 5V na 3.3V jer je to nazivni napon ESP32 mikrokontrolera, a naponi iznad 3.6V bi ga oštetili. Nadalje, kako USB ima svoj diferencijalni komunikacijski protokol, potreban je čip koji će pretvoriti te signale u neki drugi oblik, razumljiv za mikroupravljač ESP32, odnosno UART. Tu funkciju obavlja CP2102 most. [9]

Na lijevoj strani u shemi nalazi se antena sa prilagodnim sklopom impedancije, gumb za resetiranje mikrokontrolera te dva sklopa s oscilatorima za generiranje radnog takta.

Na drugoj strani, desno od ESP32 mikrokontrolera (na shemi) nalazi se MCP73831 punjač za litij-ionske baterije. Njegova uloga je reguliranje napona i struje za punjenje baterije što znači da ima režime konstantne struje (engl. CC – *Constant Current*) i konstantnog napona (engl. CV – *Constant Voltage*). Također može otkriti bateriju koja ima prenizak napon i potreban joj je poseban režim punjenja. Baterija se tada puni određenim postotkom od iznosa struje u CC režimu (10%). [10]

Tu se još nalaze vanjska memorija, sklop za automatsko resetiranje pri prenošenju koda sa računala te ugrađena svjetleća dioda (engl. LED – *Light Emitting Diode*) na GPIO 5.

I na kraju, shemi su dodani konektori za sve bitne ulaze/izlaze kao i naponske točke USB-a (5V), izlaza naponskog regulatora (3.3V), napona baterije te mase (engl. GND – *ground*). Ti konektori rasporedom na pločici u realnom prostoru poklapaju se sa dimenzijama eksperimentalne pločice pa je i njezino korištenje u realizaciji projekata i eksperimentiranju omogućeno.



Sl. 3.1. *Elektronička shema ESP32 razvojne pločice*

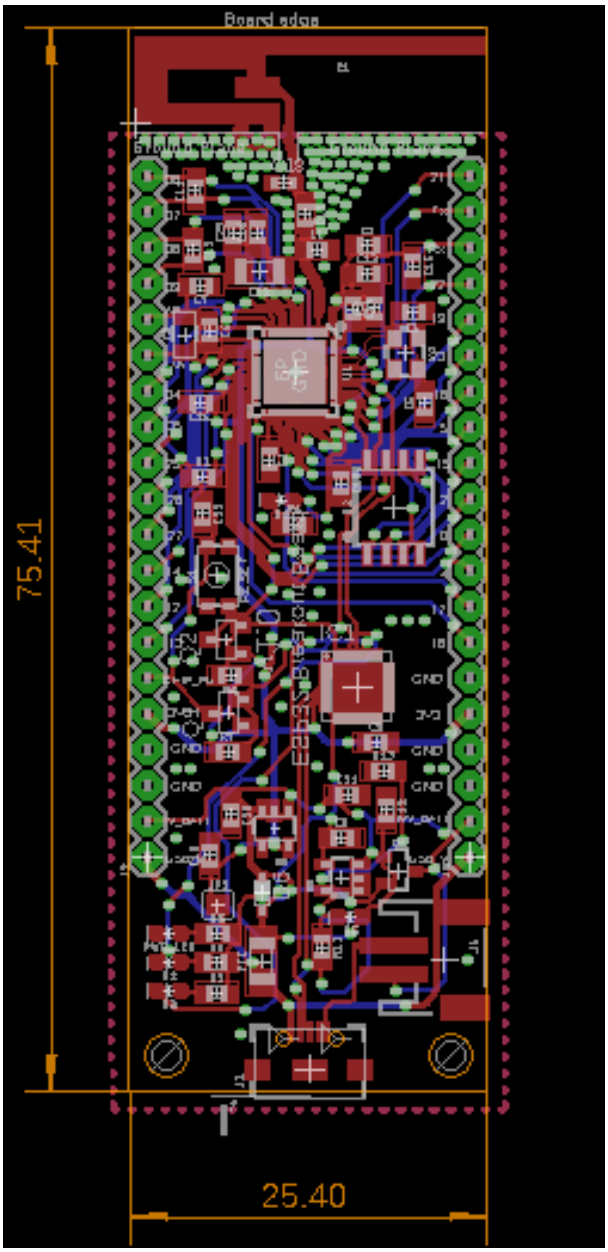
3.2. Izrada makete sustava

Nakon izrade sheme, slijedi dizajn pločice, preraspodjela komponenti, njihovo povezivanje i konačan izgled pločice u fizičkom svijetu. Nužno je bilo odrediti pravilne dimenzije pločice, kako bi konektori za ulaze i izlaze bile na pravom mjestu za korištenje na eksperimentalnoj pločici.

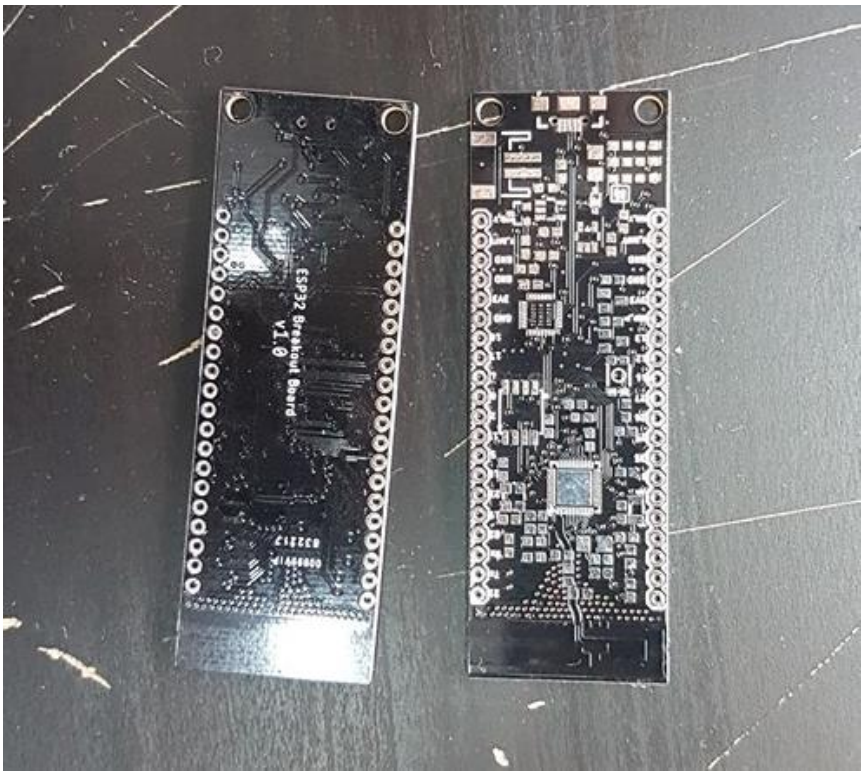
Pri dnu pločice postavljen je USB konektor na koji se povezuje ulazni naponski regulator. U tom donjem dijelu nalaze se još i punjač i konektor za li-ion bateriju, čip za komunikaciju s računalom preko UART veze, svjetleće diode za vizualni prikaz komunikacije, svjetleća dioda koja označava normalan rad pločice (*power*) i ostale jednako važne pasivne komponente.

U sredini su smješteni ESP32 čip kao centralna jedinica oko koje se sve grana, memorija, gumb za ponovno paljenje i oscilatori kao izvori takta.

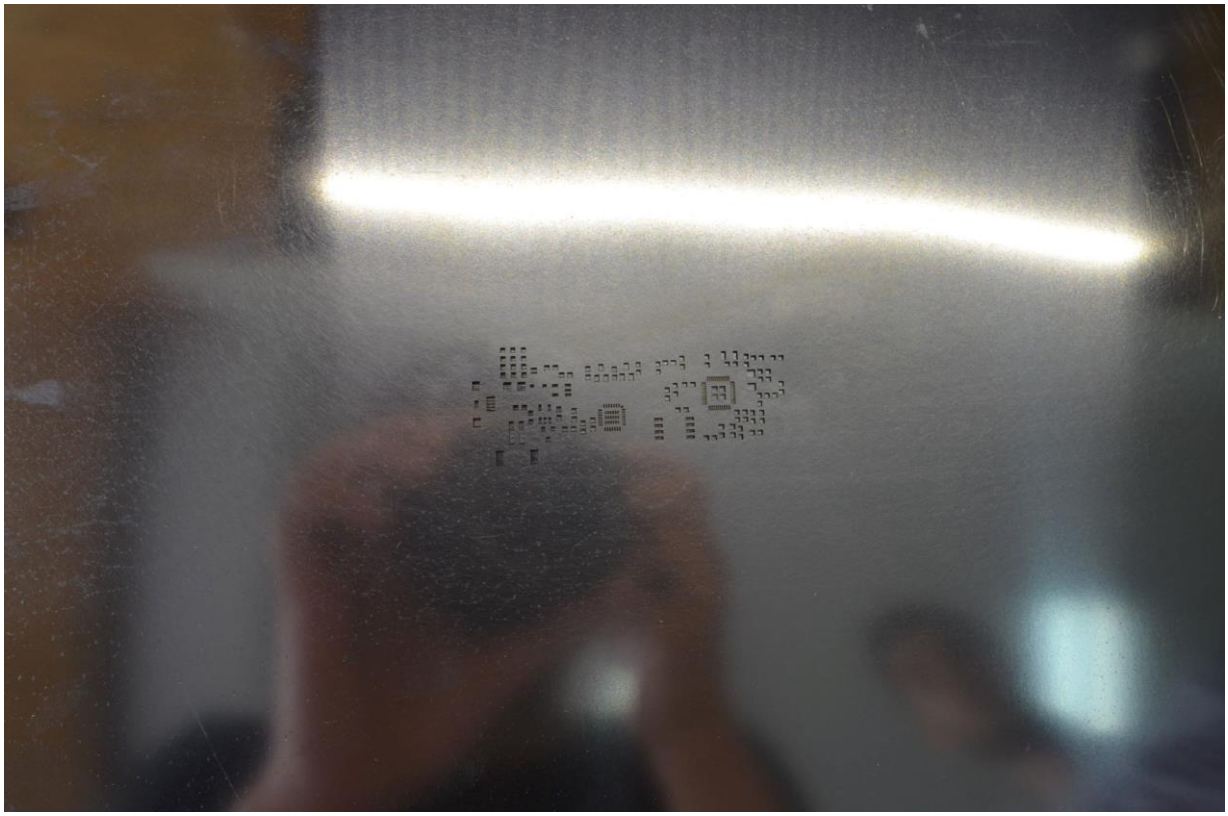
I na kraju, na samom vrhu nalazi se WiFi antena, udaljena od ostalih komponenti kako bi se smanjio utjecaj šuma.



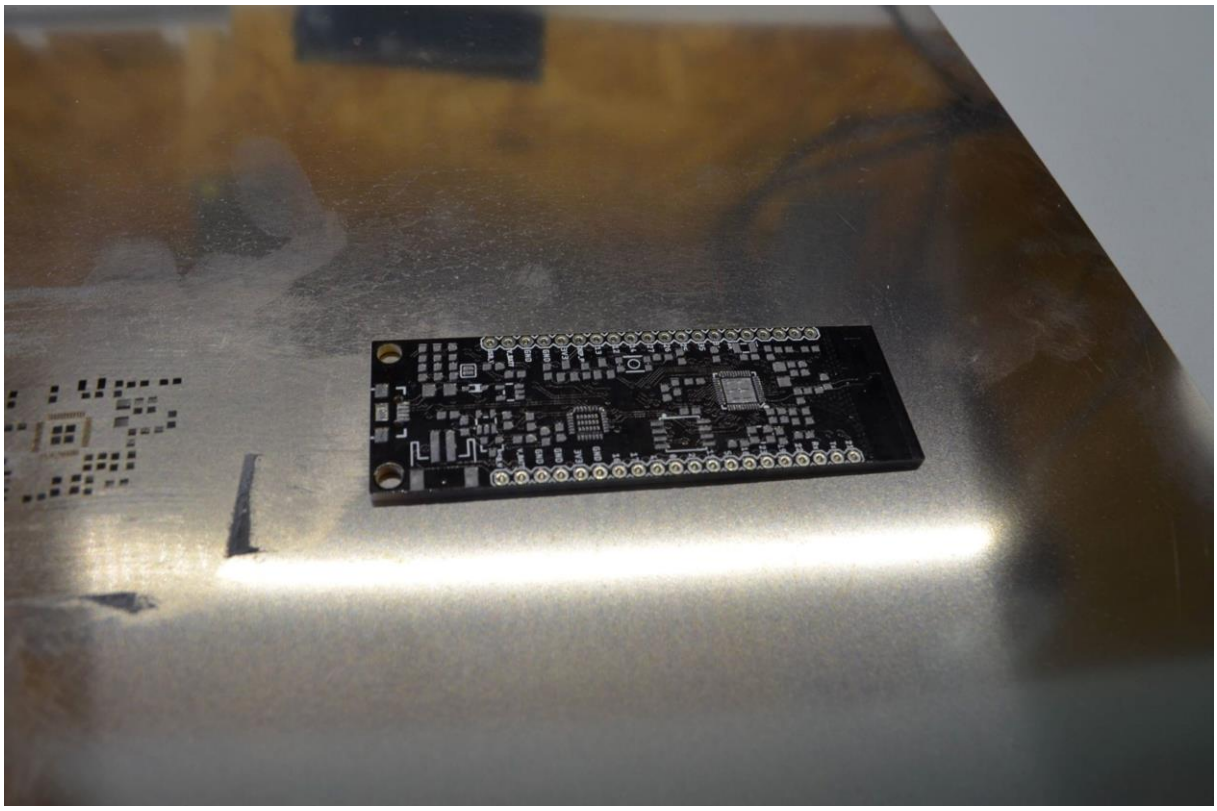
Sl. 3.2. *Dizajniranje ESP32 razvojne pločice (.brd)*



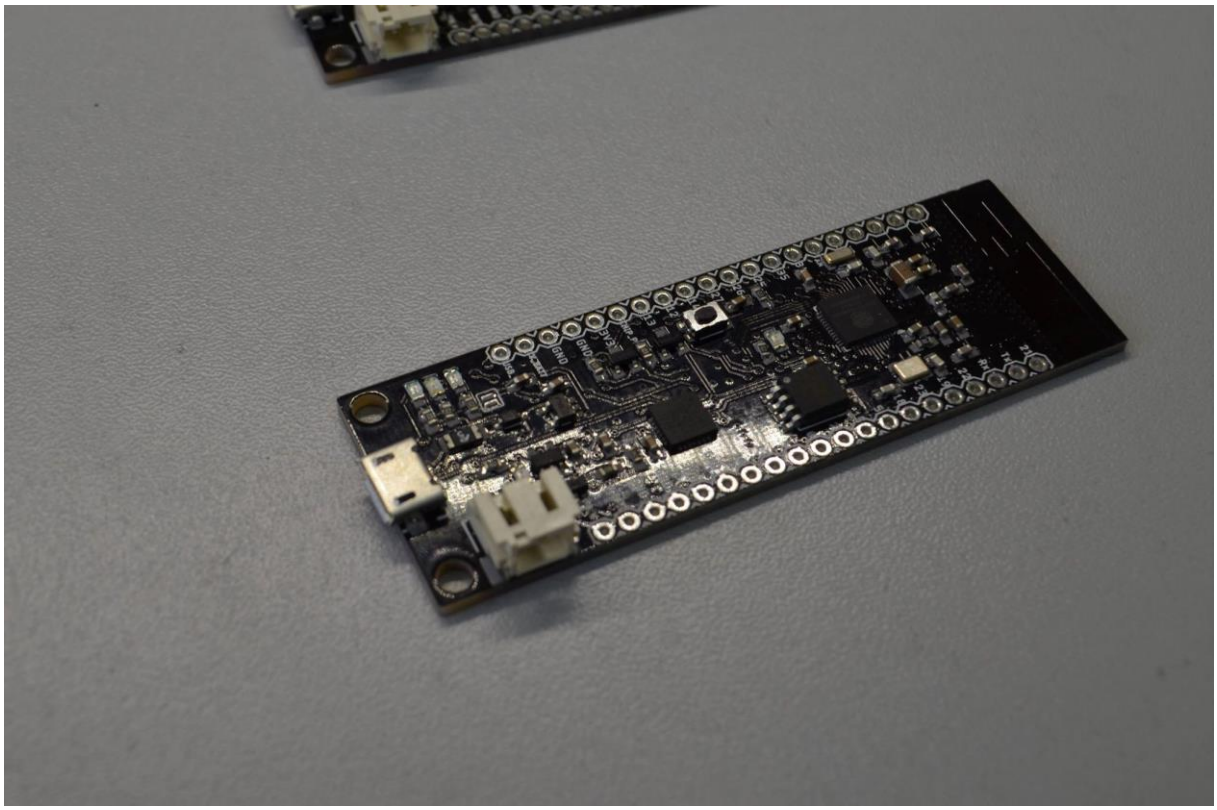
Sl. 3.3. *ESP32 razvojna pločica nakon proizvodnje (bez zalemljenih komponenti)*



Sl. 3.4. *Postavljanje paste za lemljenje*



Sl. 3.5. *ESP32 razvojna pločica s pastom za lemljenje*

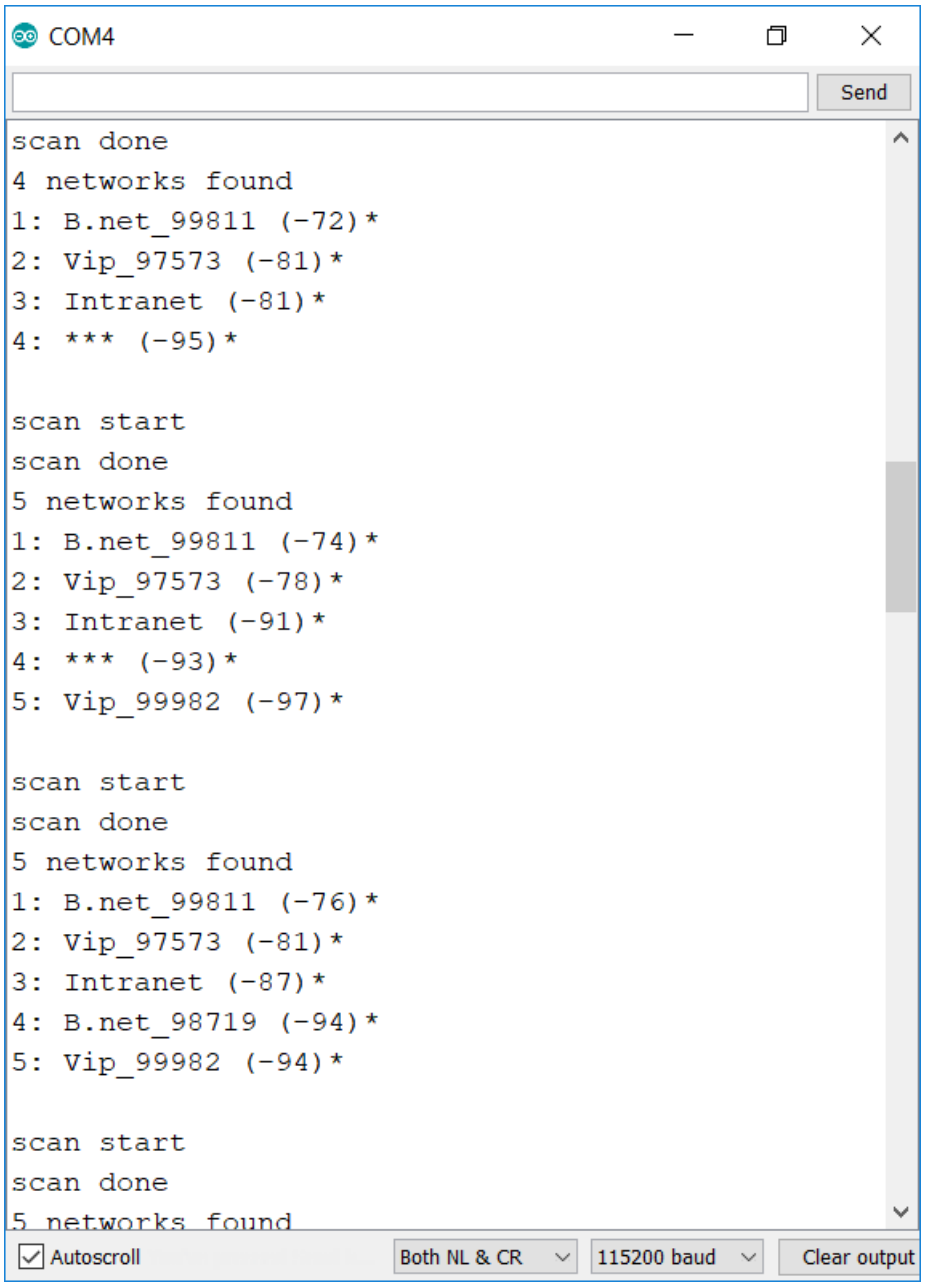


Sl. 3.6. *ESP32 razvojna pločica, v1.0*

3.3. Programsko rješenje sustava

3.3.1. Skeniranje WiFi mreža

U prvom primjeru testirana je funkcionalnost WiFi antene, odnosno ESP32 mikrokontroler je uspješno otkrio okolne bežične (engl. WLAN – Wireless Local Area Network) mreže. Nakon prenošenja programskog koda na mikroupravljač preko serijskog sučelja ESP32 je ispisao sljedeće rezultate. Programski kod je dan u prilogu A.

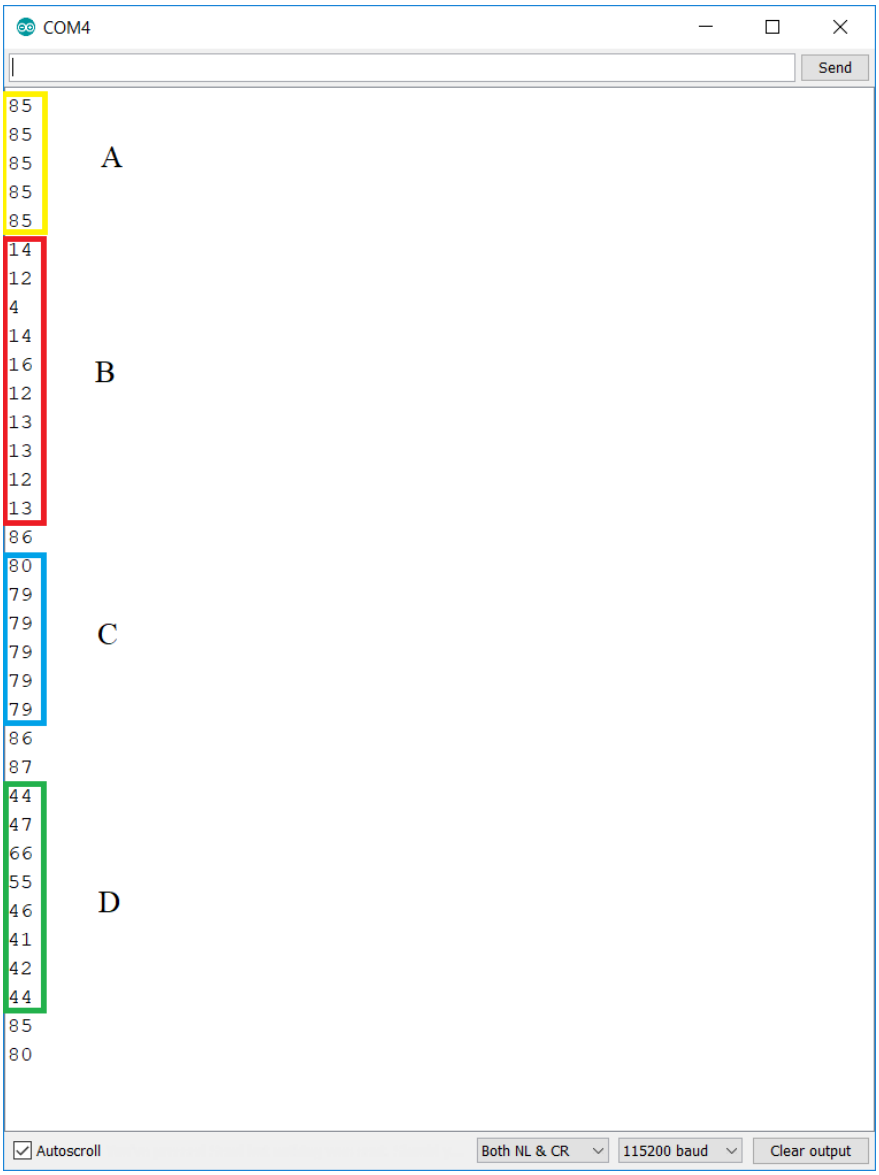


Sl. 3.7. *Ispis detektiranih WLAN mreža*

Detektirane wifi mreže ispisane su silazno po jačini signala, na što upućuju i podatci u zagradama, odnosno dobitak antene na tom mjestu izražen u dBi.

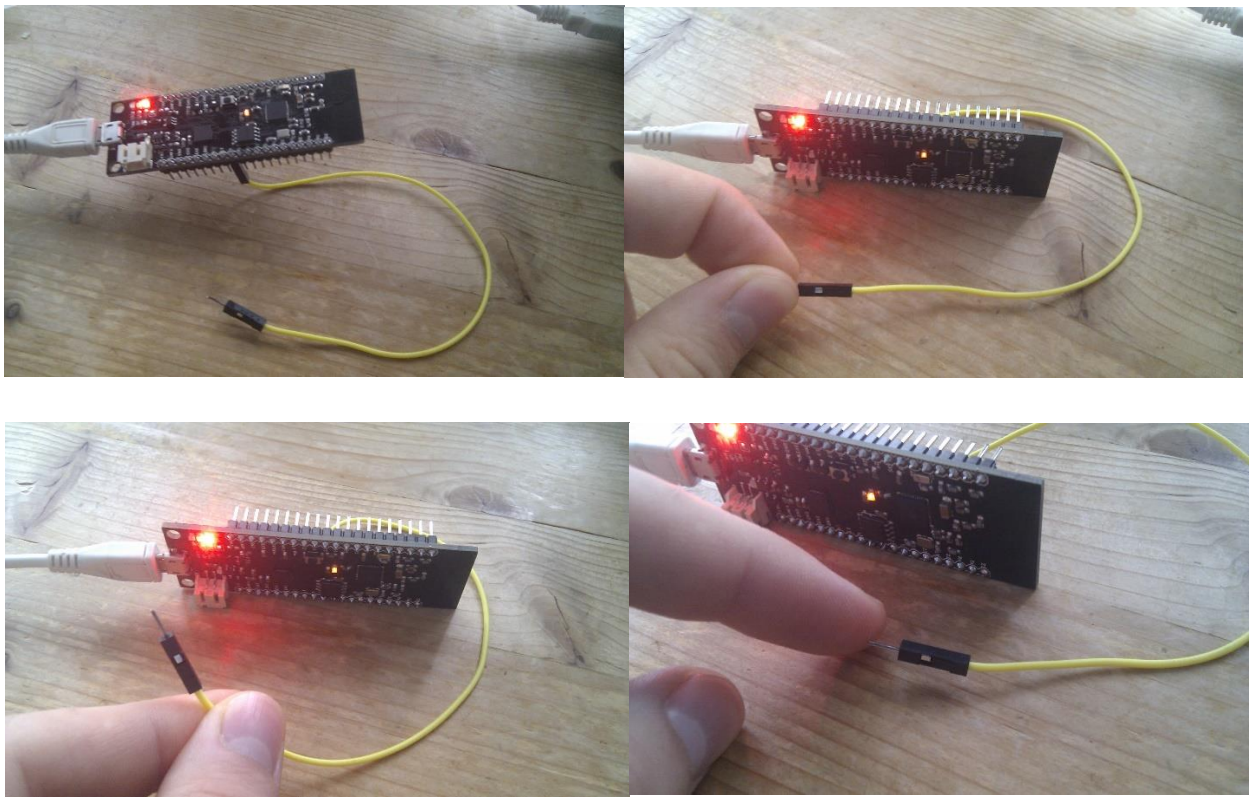
3.3.2. Senzor dodira

U drugom primjeru testirano je funkcioniranje kapacitivnog senzora, odnosno senzora dodira (engl. *Touch sensor*). Senzor dodira nalazi se na pinu 4 na koji je povezan jedan eksperimentalni kablić i pokus je proveden u četiri različite situacije. Prvo bez dodirivanja vodiča na pinu 4, zatim dodir s dva prsta, pritisak na izolirani dio kabela i zadnje lagani dodir jednim prstom na vrh metalnog dijela kabela. Dobiveni su sljedeći rezultati.



Sl. 3.8. *Ispis vrijednosti senzora dodira*

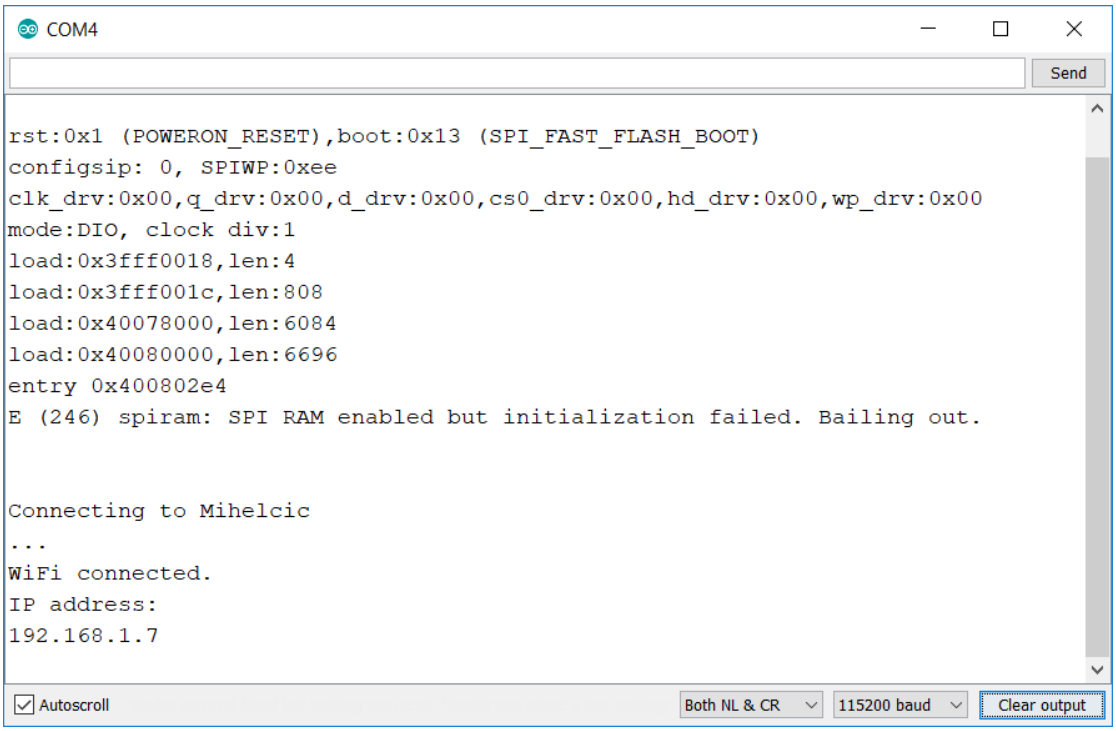
Sljedeće slike prikazuju pojedine situacije.



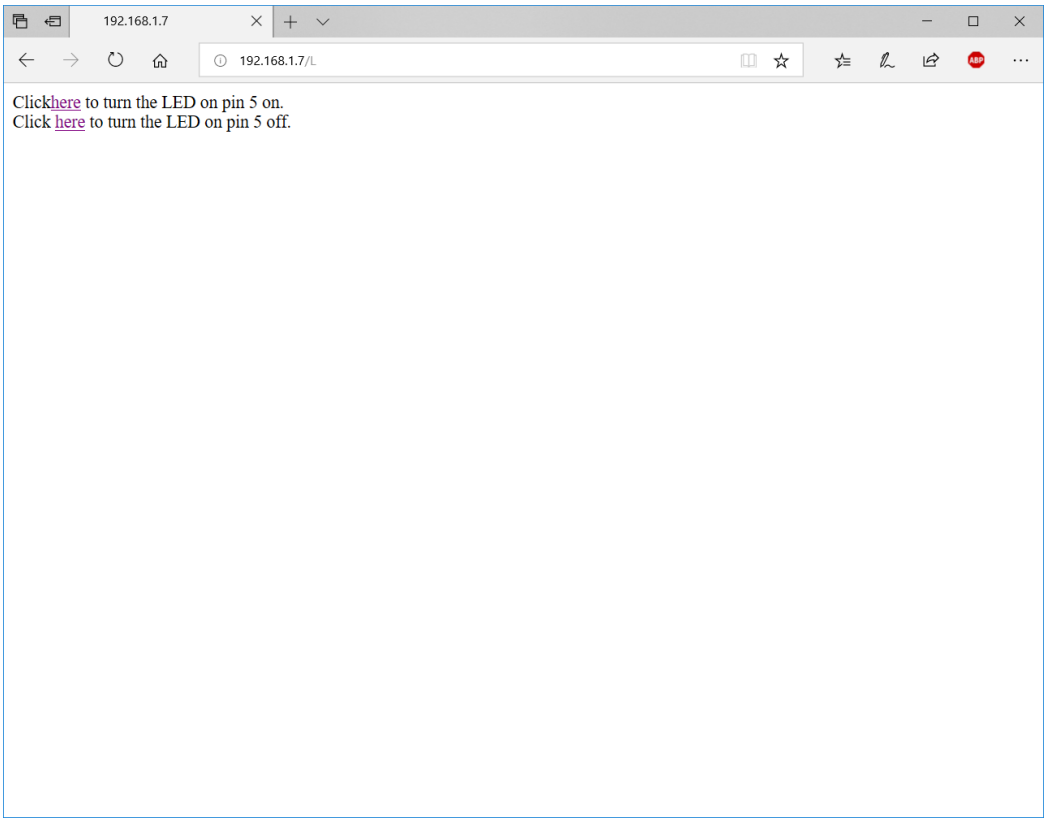
Sl. 3.9. *A (gore lijevo), B (gore desno), C (dole lijevo), D (dole desno)*

3.3.3. WiFi server

U trećem primjeru ESP32 pločica je podešena da ima ulogu servera. Nakon programiranja i izvedbe koda, na serijskom sučelju ispisuju se podatci o vezi mikroupravljača i lokalne WiFi mreže. Također ispisuje se i pridodjeljena IP (engl. *Internet Protocol*) adresa. Putem te adrese možemo pomoću računala ili pametnog telefona pristupiti mikroupravljaču preko Interneta uz uvjet da su oba uređaja spojena na istu WiFi mrežu i upravljati ugrađenom svjetlećom diodom na nožici 4 ESP32 mikroupravljača.



Sl. 3.10. *WiFi server, podatci o vezi*



Sl. 3.11. *WiFi server, pristup preko IP adrese*

4. ZAKLJUČAK

U ovom završnom radu uspješno je napravljen razvojni sustav baziran na ESP32 mikroupravljaču koji je svojim dimenzijama prilagođen za rad i testiranje pomoću eksperimentalne pločice. Sve testirane funkcije rade ispravno te je korisniku dana mogućnost realizacije nekih projekata u skladu i okviru mogućnosti ESP32 mikroupravljača.

Konačna pločica relativno je malih dimenzija pa tako može naći primjenu u širem krugu *Internet of things* svijeta. Također, brzina obrade podataka, a s druge strane, niska potrošnja čipa čine ovaj sustav jednim od logičnih izbora pri odabiru središnje kontrolne jedinice, odnosno „mozga“ operacije u nekom automatiziranom sustavu.

LITERATURA

[1] Arduino, https://hr.wikipedia.org/wiki/Arduino, lipanj 2018.

[2] Arduino Software (IDE), https://www.arduino.cc/en/guide/Environment, lipanj 2018.

[3] Otvoreni kod, https://hr.wikipedia.org/wiki/Otvoreni_kod, lipanj 2018.

[4] Arduino Uno, https://store.arduino.cc/usa/arduino-uno-rev3, lipanj 2018.

[5] ESP32 – Wikipedia, https://en.wikipedia.org/wiki/ESP32, lipanj 2018.

[6] ESP32 – tehnička dokumentacija, https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, lipanj 2018.

[7] W25Q32JV, https://www.winbond.com/resource-files/w25q32jv%20dtr%20revf%2002242017.pdf, rujan 2018.

[8] SE5218 naponski regulator, https://html.alldatasheet.com/html-pdf/292068/SEAWARD/SE5218/293/1/SE5218.html, rujan 2018.

[9] CP2102, https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf, lipanj 2018.

[10] MCP73831, https://cdn.sparkfun.com/assets/learn_tutorials/6/9/5/MCP738312.pdf, lipanj 2018.

SAŽETAK

Naslov: WiFi modul ESP32

U radu je opisan proces dizajniranja i proizvodnje razvojnog mikroupravljačkog sustava baziranog na ESP32 mikroupravljaču koji ima mogućnost povezivanja na WiFi, odnosno Internet. Bitno je naglasiti da je jedan od najbitnijih elemenata u pripremi dobro proučiti tablicu s tehničkim podatcima elemenata koje ulaze u sustav te prema preporukama proizvađača dizajnirati najprije samu shemu sustava, zatim konačan izgled i raspored elemenata. Također od velike važnosti je naučiti se koristiti programima, tj. alatima, poput programa Eagle, bez kojih rad ne bi bio moguć. Ovakav proces proizvodnje zahtjeva inženjerski pristup, ustrajnost i staloženost pri radu, drugim riječima svaki dio sustava treba se pomno pregledati više puta prije same izrade i nastavka rada. Krajnji rezultati pokazali su se vrlo uspješnim jer je svaka testirana funkcija mikroupravljača ispravno radila.

Ključne riječi: ESP32, Internet, mikroupravljač, WiFi modul

ABSTRACT

Title: ESP32 WiFi module

The paper describes the process of designing and manufacturing a development microdrive system based on an ESP32 microcontroller that has the ability to connect to WiFi or the Internet. It is important to emphasize that one of the most important elements in preparation is to study the datasheets of elements used in the system and, according to the manufacturer's recommendations, first design the system itself, then the final layout and layout of the elements. It is also of great importance to learn to use programs, i.e. tools, such as the Eagle program, without which work would not be possible. Such a production process requires engineering approach, perseverance, and job diligence, in other words, every part of the system needs to be thoroughly reviewed several times prior to its creation and continuation. The final results proved to be very successful because every tested microcontroller function was working properly.

Keywords: ESP32, Internet, microcontroller, WiFi module

ŽIVOTOPIS

Branimir Mihelčić rođen je 24. svibnja 1996. godine u Slavonskom Brodu. Osnovnu školu završava u OŠ „Ivan Meštrović“ Vrpolje, nakon čega pohađa Gimnaziju „Matija Mesić“ u Slavonskom Brodu, program Prirodoslovno-matematička gimnazija te maturira 2015. godine. Iste godine upisuje preddiplomski sveučilišni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku te se nakon prve godine opredjeljuje za smjer Komunikacije i informatika.

PRILOG A. Program za ESP32 razvojni sustav „WiFi Scan“

```
#include "WiFi.h"

void setup(){
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);
    Serial.println("Setup done");
}

void loop(){
    Serial.println("scan start");
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0){
      Serial.println("no networks found");
    }
    else{
      Serial.print(n);
      Serial.println(" networks found");
      for (int i = 0; i < n; ++i){
        Serial.print(i + 1);
        Serial.print(": ");
        Serial.print(WiFi.SSID(i));
        Serial.print(" (");
        Serial.print(WiFi.RSSI(i));
        Serial.print(")");
        Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
        delay(10);
      }
    }
    Serial.println("");
    delay(5000);
}
```

PRILOG B. Programski kod za senzor dodira

```
void setup()
{
  Serial.begin(115200);
  delay(1000);
  Serial.println("ESP32 Touch Test");
}

void loop()
{
  Serial.println(touchRead(T0));
  delay(1000);
}
```

PRILOG C. Program za ESP32 razvojni sustav „WiFi Server“

```
/*
 WiFi Web Server LED Blink

 A simple web server that lets you blink an LED via the web.
 This sketch will print the IP address of your WiFi Shield (once connected)
 to the Serial monitor. From there, you can open that address in a web browser
 to turn on and off the LED on pin 5.

 If the IP address of your shield is yourAddress:
 http://yourAddress/H turns the LED on
 http://yourAddress/L turns it off

 This example is written for a network using WPA encryption. For
 WEP or WPA, change the Wifi.begin() call accordingly.

 Circuit:
 * WiFi shield attached
 * LED attached to pin 5

 created for arduino 25 Nov 2012
 by Tom Igoe

ported for sparkfun esp32
31.01.2017 by Jan Hendrik Berlin

 */

#include <WiFi.h>

const char* ssid     = "yourssid";
const char* password = "yourpasswd";

WiFiServer server(80);

void setup()
{
    Serial.begin(115200);
    pinMode(5, OUTPUT);

    delay(10);

    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
```

```
    server.begin();

}

int value = 0;

void loop(){
 WiFiClient client = server.available();

  if (client) {
    Serial.println("New Client.");
    String currentLine = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n') {



          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();

            client.print("Click <a href=\"/H\">here</a> to turn the LED on pin 5
on.<br>");
            client.print("Click <a href=\"/L\">here</a> to turn the LED on pin 5
off.<br>");
            client.println();
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }


        if (currentLine.endsWith("GET /H")) {
          digitalWrite(5, HIGH);
        }
        if (currentLine.endsWith("GET /L")) {
          digitalWrite(5, LOW);
        }
      }
    }
    client.stop();
    Serial.println("Client Disconnected.");
  }
}
```