Matija Nestić
0036478072
Mato Lovrenović
0036468828

SEMINARSKI RAD - SPVP

Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradbu informacija
Sveučilište u Zagrebu

Sigurna i upravljiva utičnica

- Tekst je namijenjen svima zainteresiranima za pametne kuće
- Potrebna predznanja: Elektronika, računalne mreže
- Prototip pametne utičnice

14. lipnja 2018

creative commons (cc) BY-NC-SA

Sažetak

Pametne utičnice omogućuju korisnicima da uključuju i isključuju trošila putem Interneta. Sigurna, štedljiva, upravljiva utičnica nudi korisnicima veću sigurnost korištenjem senzora za detekciju trošila. Senzor omogućuje da je napon u utičnici samo kada je detektirano spojeno trošilo te se time minimizira mogućnost strujnog udara.
Utičnica se sastoji od senzora blizine, releja, tipkala za ručno uključivanje trošila i LED indikatora. Prilikom uključivanja utičnica se spaja na lokalnu WiFi mrežu. Za povezivanje sa lokalnom mrežom i upravljanje utičnicom koristi se mikrokontroler ESP8266 za kojeg je razvijena programska podrška korištenjem Arduino biblioteka.
Prednosti sigurne, štedljive i upravljive utičnice su mogućnost povezivanja sa Internetom, kontrola utičnice iz bilo kojeg kutka svijeta i sigurnost koju nam pruža senzor blizine objekata.
Projekt je koristan svima koji žele upravljati trošilima sa veće udaljenosti ili im je bitno da iz sigurnosnih razloga utičnica nije pod naponom.

Sadržaj

UVOD 3
DIJELOVI SUSTAVA (MATIJA NESTIĆ) 4
1.1. Power board 5
1.2. Logic board 7
1.2.1. Regulator napona 3,3 V 8
1.2.2. ESP8266 9
1.2.3. Komponente za interakciju sa korisnikom 11
1.2.4. Značenje i prikaz statusa LED indikatorom 12
1.2.5. Senzor za detekciju spojenog trošila (QRE1113) 13
KUČIŠTE (MATIJA NESTIĆ) 14
PROGRAMSKA PODRŠKA (MATO LOVRENOVIĆ) 17
1.2.6. Incijalizacija 18
1.2.7. Spajanje na lokalnu WiFi mrežu 19
1.2.8. Detekcija i klasifikacija pritisnute tipke 20
1.2.9. Agent za promjenu stanja 21
1.2.10. Posluživanje HTTP zahtjeva 22
ZAKLJUČAK 24
LITERATURA 25

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

Uvod

Pametne utičnice povezane sa Internetom, utikači, letve za napajanje i prekidači izvrstan su početak izgradnje pametne kuće. Dodaju praktičnost bilo kojem uređaju i mogu uvelike smanjiti račun za struju isključivanjem električnih uređaja koji se trenutno ne koriste i troše struju dok čekaju u pripravnosti (*stand-by*). Osim smanjenje potrošnje, povezanost sa Internetom omogućuje upravljanje utičnicom iz bilo kojeg kutka Zemlje.

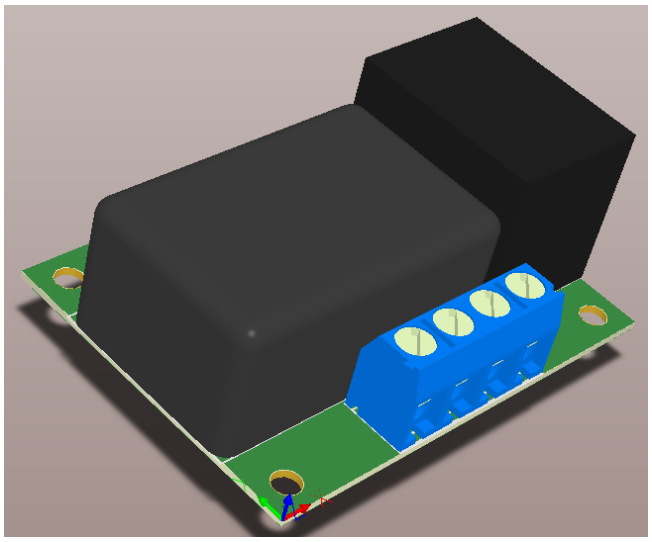
Štedljiva, sigurna i upravljiva utičnica polazi od koncepta pametne utičnice, ali mu se dodaje sigurnosni aspekt. Cilj je povećati sigurnost utičnice odnosno smanjiti broj nesreća uzrokovanih strujnim udarom. Sigurnost utičnica može biti osobito bitna obiteljima sa malo djecom koja mogu nastradati ako gurnu prst ili neki metalni predmet u utičnicu. Rješenje tog probleme je utičnica koja je pod naponom samo kada je priključeno trošilo.

Uz sigurnosni aspekt utičnica je i upravljiva preko Interneta. Uz upravljanje utičnicom to nam omogućava i slanje informacija o korištenju trošila. Takve podatke možemo iskoristiti za analizu potrošnje i korištenja ili za pokretanje nekog drugog djela pametne kuće.

Sklopovlje pametne utičnice (Matija Nestić)

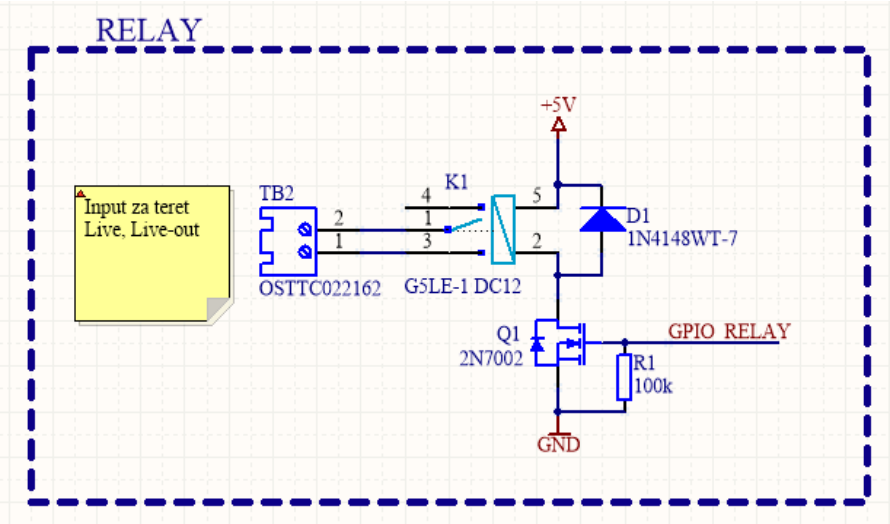
Sigurna, štedljiva i upravljiva utičnica sastoji se od kućišta, ugradbene elektronike i programske podrške. Unutar kućišta nalaze se dvije tiskane pločice. Tiskane pločice podijeljene su po funkcionalnosti, prva pločica sadrži komponente potrebne za napajanja sklopova i relej za uključivanje odnosno isključivanje trošila. Druga tiskana pločica na sebi sadrži komponente za komunikaciju, interakciju sa korisnikom (tipkalo, LED indikator) i senzor prisutnosti utikača u utičnici. Tiskane pločice su povezane kablom za upravljanje relejom te napajanje logičke tiskane pločice.

1.1. Power board

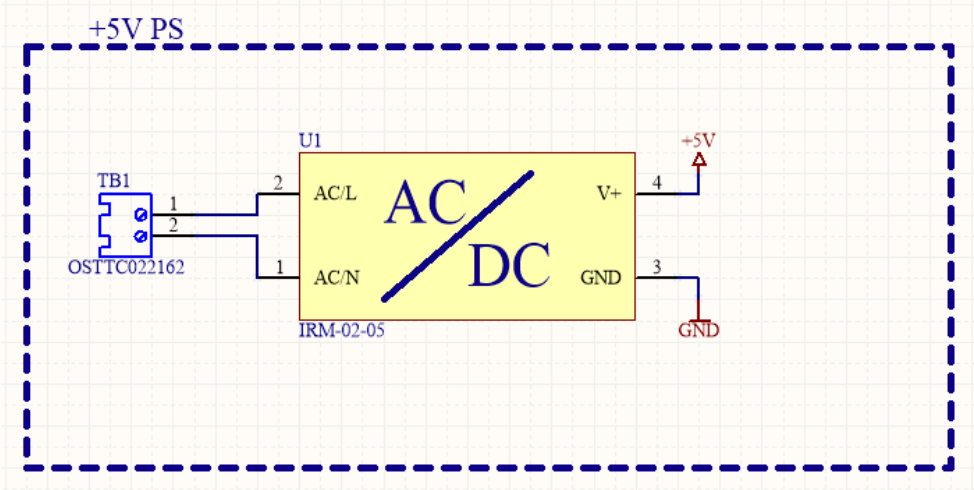


Slika 1 3D prikaz *power board* tiskane pločice

Power board je tiskana pločica na kojoj se nalaze komponente za napajanje sklopova i relej koji uključuje odnosno isključuje trošilo. Na ulazne konektore tiskane pločice dovode se obje faze gradske mreže. Relej se koristi kako bi kontrolirao uključivanje i isključivanje trošila prekidanjem jedne od faza gradske mreže. Kontaktom releja mogu teći struji do 10 A, uz prekidanje napona do 250 VAC. Špula releja uklapa kontakte sa naponima od 5 V. Za kontrolu releja koristi se MOSFET 2N7002. Pošto je špula induktivni teret koji uključujemo i isključujemo relativno velikom brzinom potrebno je zaštiti tranzistor diodom D1 (1N4148) od prevelikog induciranog napona.

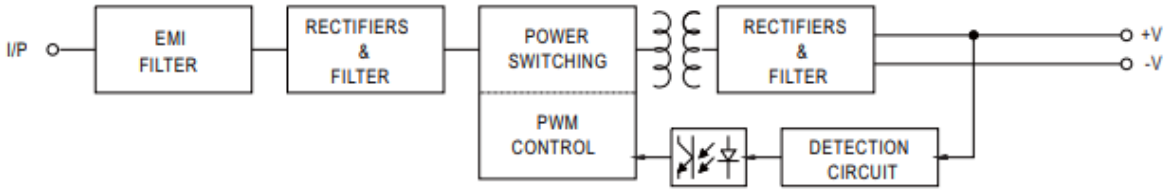


Slika 2 Shema sklopa za upravljanje i zaštitu releja



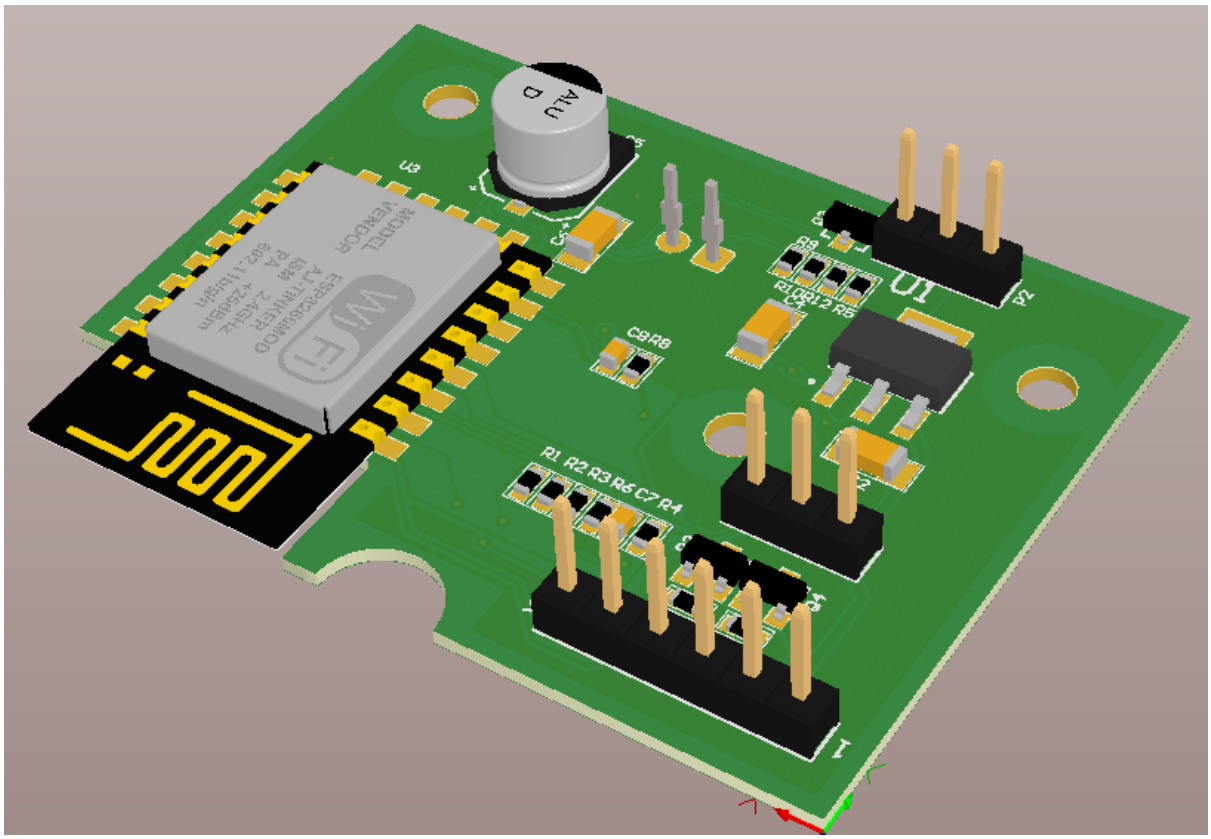
Slika 3 Shema napajanja +5 V

Za dobivanje napajanja od 5 V koristi se Meanwellovo AC/DC prekidačko napajanje. Prekidačko napajanje IRM-02-05 dolazi kao gotov modul i ne zahtjeva dodatne pasivne komponente. Snaga prekidačkog napajanja je 2 W, a izlazni napon je stabiliziranih 5 V.



Slika 4 Unutarnja blok shema IRM-02-05

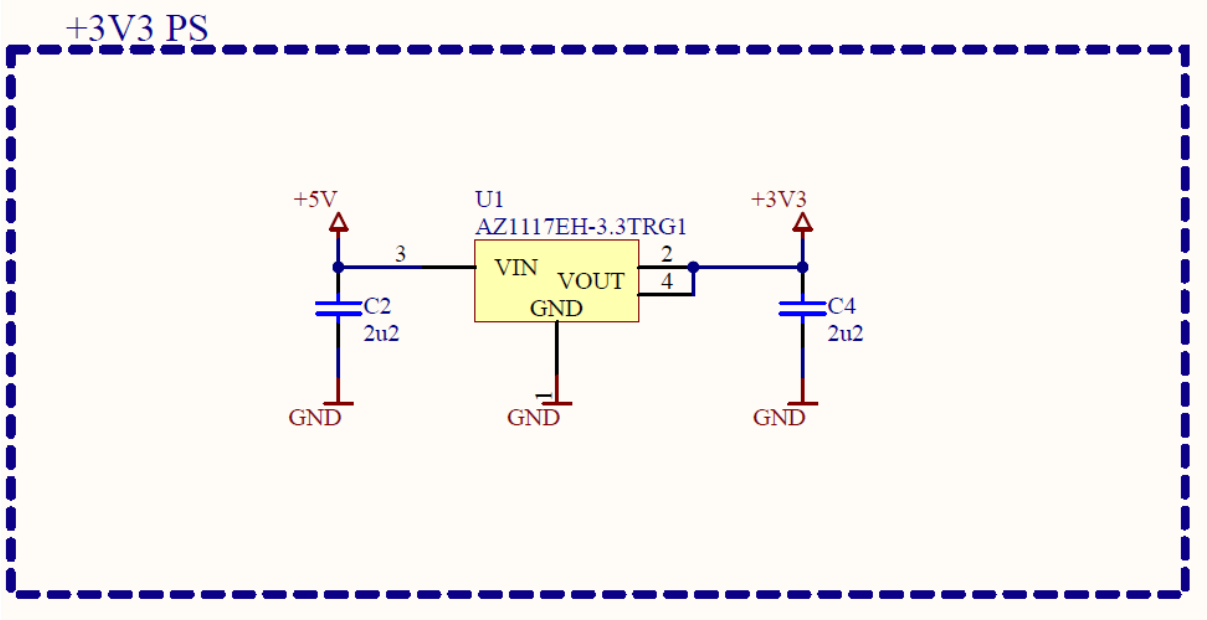
1.2. Logic board



Slika 5 3D prikaz *logic board* tiskane pločice

Logic board je tiskana pločica na kojoj se nalaze komponente za komunikaciju, upravljanje cijelim sustavom, interakciju sa korisnikom i senzor prisutnosti utikača u utičnici. Za komunikaciju se koristi ESP8266 koji se spaja na lokalnu WiFi mrežu i stvara mali web server za upravljanje utičnicom. ESP8266 koristi se i za upravljanje cijelim sustavom. Korisnik može upravljati utičnicom preko web servera ili preko tipkala na kućištu. Povratnu informaciju o stanju utičnice korisnik može dobiti preko web servera ili putem LED indikatora.

1.2.1. Regulator napona 3,3 V



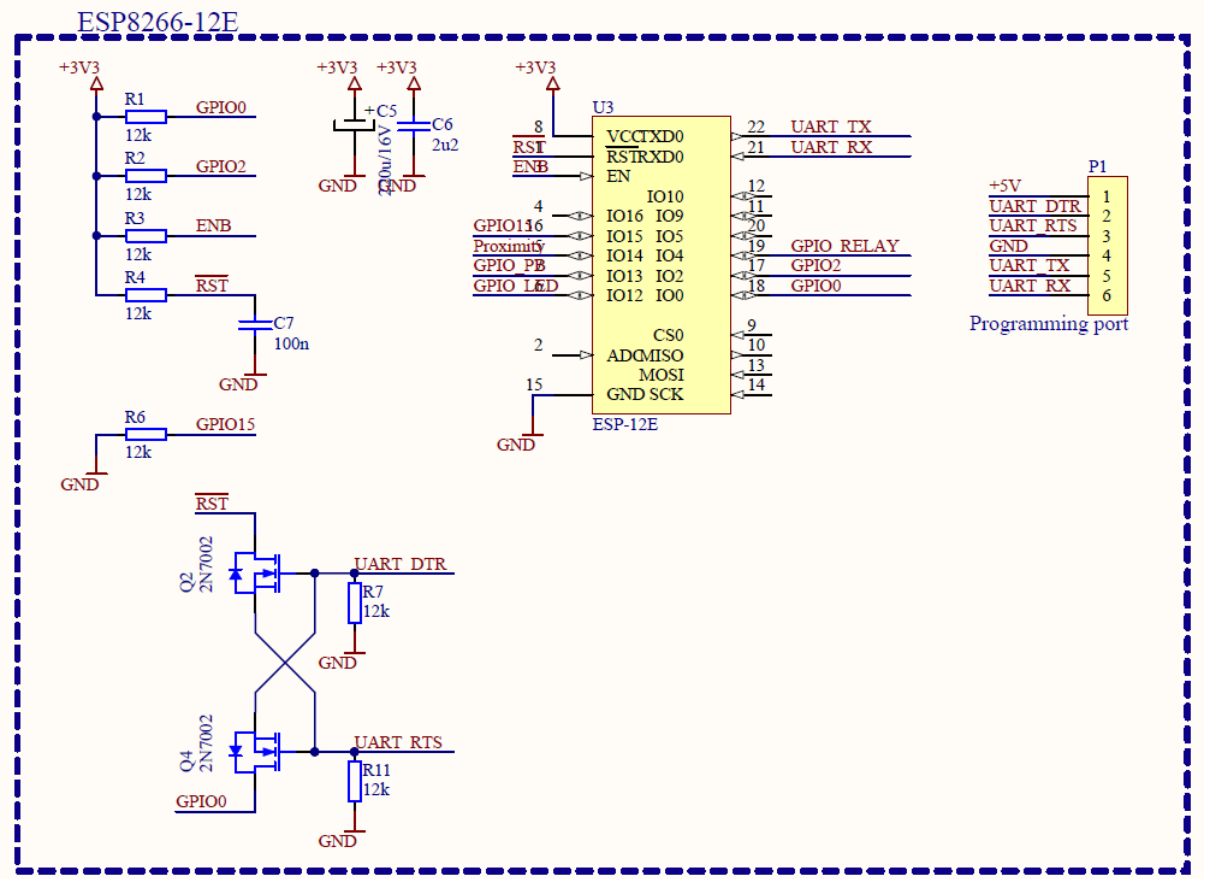
Slika 6 Shema regulatora 3,3 V

Za dobivanje napona od 3,3 V koristi se linearni regulator AZ1117EH-3.3 koji se nalazi na *logic board* PCB-u. Linearni regulator je odabran jer je ulazni napon relativno mali (5 V) pa nemamo preveliku disipaciju snage. To nam omogućuje jednostavniji dizajn u odnosu na prekidački regulator. Uz regulator imamo dva keramička kondenzatora u 1206 kućištu kapaciteta 2.2 uF. Koriste za filtriranje i stabilizaciju regulatora.

1.2.2. ESP8266

ESP8266 je jeftini mikrokontroler sa podrškom za WiFi. Dolazi sa integriranim TCP/IP *stackom*. Proizvodi ga Espressif Systems iz Kine. U sklopu ovog projekta korišten je gotov modul ESP8266-12E. Na tiskanoj pločici modula se uz mikrokontroler nalazi i *flash* memorija, antena i par pasivnih komponenti potrebnih za rad sklopa. Gotov modul omogućuje brzi razvoj i lagano povezivanje na lokalnu mrežu odnosno vezu sa Internetom. Mikrokontroler uz TCP/IP *stack* ima 17 GPIO priključaka te jedinice za komunikaciju pomoću I2C, 1-wire, SPI, UART protokola što nam omogućuje kontrolu nad sustavom (nije nam potreban još jedan mikrokontroler). U ovome projektu ESP8266 se koristi za povezivanje sa lokalnom mrežom WiFi mrežom, stvaranje web servera na mreži te za očitavanje senzora i upravljanje relejom i LED indikatorom.

Slika 7

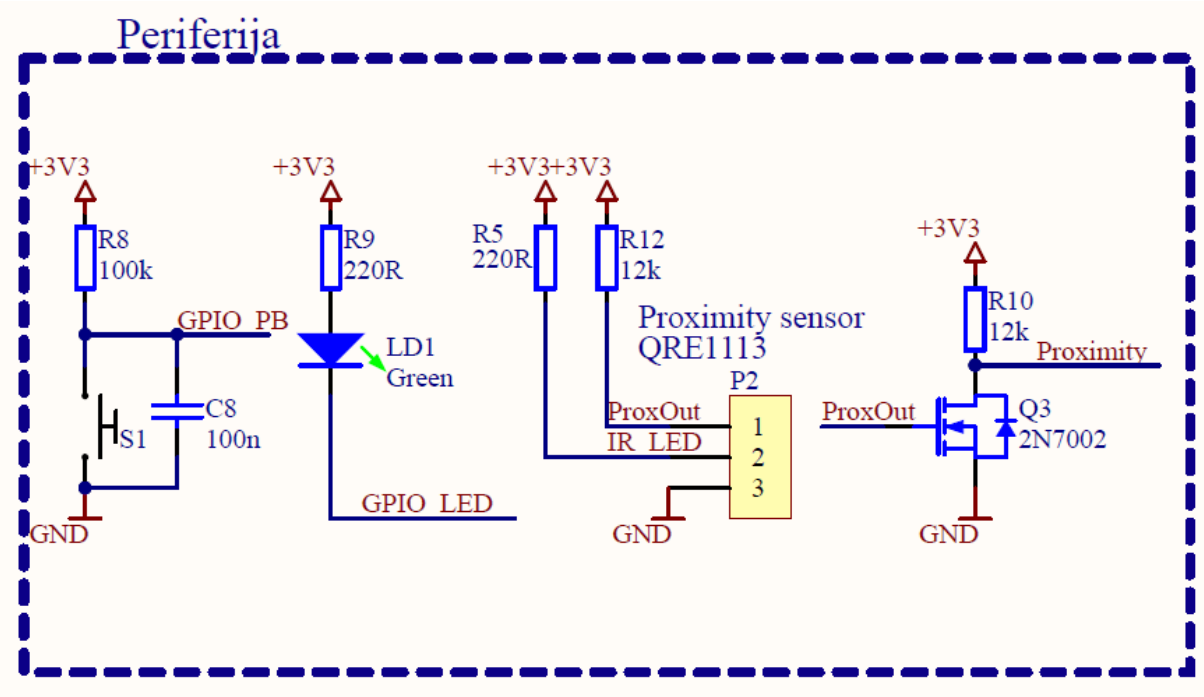


Slika 8 Shema spajanja modula ESP8266-12E sa ostatkom komponenti

Modul se napaja napajanjem od 3,3 V. Tijekom odašiljanja potrebno mu je oko 150-200 mA struje (vršna potrošnja je oko 300 mA) pa je potrebno dodati kondenzatore C5 i C6 koji moraju biti blizu modula kako bi mogli predati dovoljno energije odašiljaču. Za programiranje se koristi USB-UART *bridge* sa TX, RX i DTR, RTS linijama. TX i RX linije se koriste za prijenos podataka, a DTR i RTS linije se koriste za konfiguraciju sklopa pri programiranju (pomoću tranzistora Q4 i Q2). Pritezni otpornici R1-R6 se također koriste za konfiguraciju modula. Osim WiFi-a i UART-a koriste se ulazni GPIO priključci za prikupljanje podataka o stanju tipke i senzora blizine te izlazni GPIO priključci upravljanje relejom i LED indikatorom.

1.2.3. Komponente za interakciju sa korisnikom

Kako je već navedeno korisnik može upravljati sa utičnicom (i pregledati trenutno stanje) preko web servera koji je utičnica stvorila. Drugi način na koji korisnik može upravljati utičnicom je pomoću tipkala odnosno može saznati trenutno stanje utičnice pomoću LED indikatora na kućištu.



Slika 9 Shema periferije

Tipkalo je spojeno preko zateznog otpornika od 100 kOhma na napajanje od 3,3 V. Dok korisnik ne stisne tipkalo, mikrokontroler će očitavati visoko stanje na svom GPIO ulazu. Kada korisnik stisne tipkalo, ulaz GPIO_PB biti će spojen na masu. Kondenzator C8 koristi se za *debouncing* tipkala odnosno za sprječavanje krivog očitanja tipkala usred titranja kontakta. LED indikator spojen je na napajanje od 3,3 V preko otpornika od 220 Ohma. Katoda LED indikatora spojena je na izlazni GPIO priključak mikrokontrolera. Odabrana je ovakva konfiguracija zato jer izlazni stupanj GPIO-a ima veću *sink* od *source* struje. To nam omogućuje da po potrebi možemo podesiti otpornik R9 kako bi postigli željeno osvjetljene za specifičan LED indikator.

1.2.4. Značenje i prikaz statusa LED indikatorom

Prilikom ukopčavanja utičnice u gradsku mrežu mikrokontroler se mora inicijalizirati. Tijekom toga procesa LED indikator će cijelo vrijeme biti uključen.
Nakon toga utičnica ulazi u konfiguracijski mod. U tom stanju utičnica će stvoriti vlastitu pristupnu točku (*Access point*) pomoću koje je moguće upisati podatke potrebne za spajanje na željenu mrežu. Tijekom toga stanja LED indikator treperiti će malom frekvencijom (1 Hz).
Kada je veza ostvarena sa lokalnom mrežom LED indikator će brzo treperiti tijekom jedne sekunde te se nakon toga ugasiti.
Tijekom rada utičnice LED indikator prikazuje trenutno stanje releja utičnice. Ako je trošilo spojeno i uključeno LED indikator će svijetliti odnosno neće ako trošilo nije spojeno ili nije uključeno.

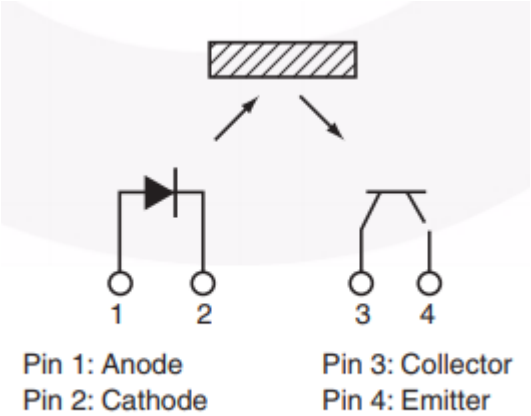
Tipkalo radi nakon konfiguracije i spajanja utičnice na lokalnu mrežu. Ako je trošilo spojeno u utičnicu, ali trenutno nije uključeno, utičnicu možemo uključiti kratkim pritiskom na tipkalo. Ako je trošilo prisutno i trenutno uključeno možemo ga isključiti dugim pritiskom na tipkalo (trajanje pritiska 2..10 sekundi).

1.2.5. Senzor za detekciju spojenog trošila (QRE1113)

Za detekciju spojenog trošila u utičnici koristio se senzor blizine objekta QRE1113. Senzor je pozicioniran u središtu utičnice. Dolazi u malom 2,7 x 3,4 mm kućištu sa četiri nožice. Unutar kućišta nalazi se IR LED dioda i fototranzistor. Detekcija objekta se vrši na principu reflektirane svjetlosti IR LED diode u fototranzistor.

Slika 10 Kućište QRE1113

Dioda je spojena na konstantni izvor napajanja od 3,3 V preko otpornika R5 od 220 Ohma (Id = 10 mA). Fototranzistor je spojen u emiterskom spoju pojačala odnosno kolektor mu je spojen na napajanje preko otpornika R12 otpora 12 kOhma. Pomoću otpora R5 i R6 moguće je mijenjati osjetljivost fototranzistora. Izlaz senzora je kolektor fototranzistora koji se nakon toga spaja na ulaz tranzistora Q3. Tranzistor 2N7002 ima napon praga od 1,2 V i koristi se kako bi osigurali da signal na ulazu GPIO priključka nikad ne dođe u nedefinirano područje rada.

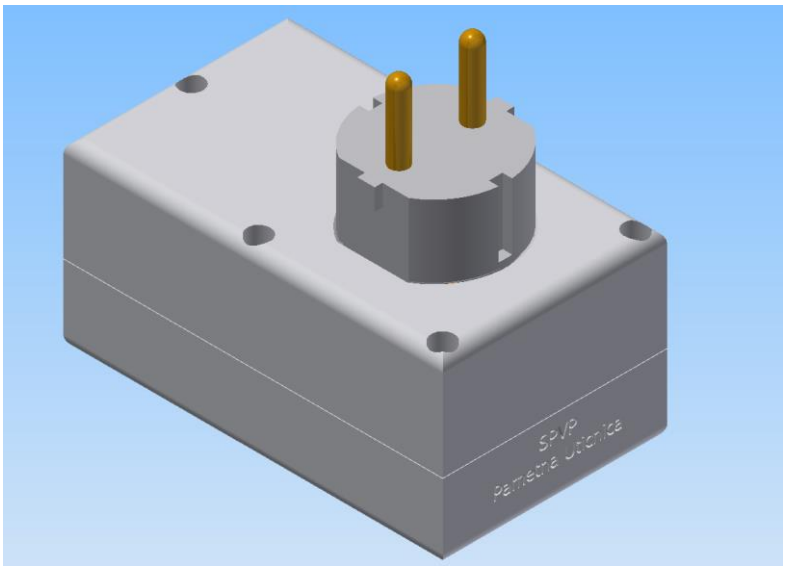


Slika 11 Princip rada senzora blizine QRE1113

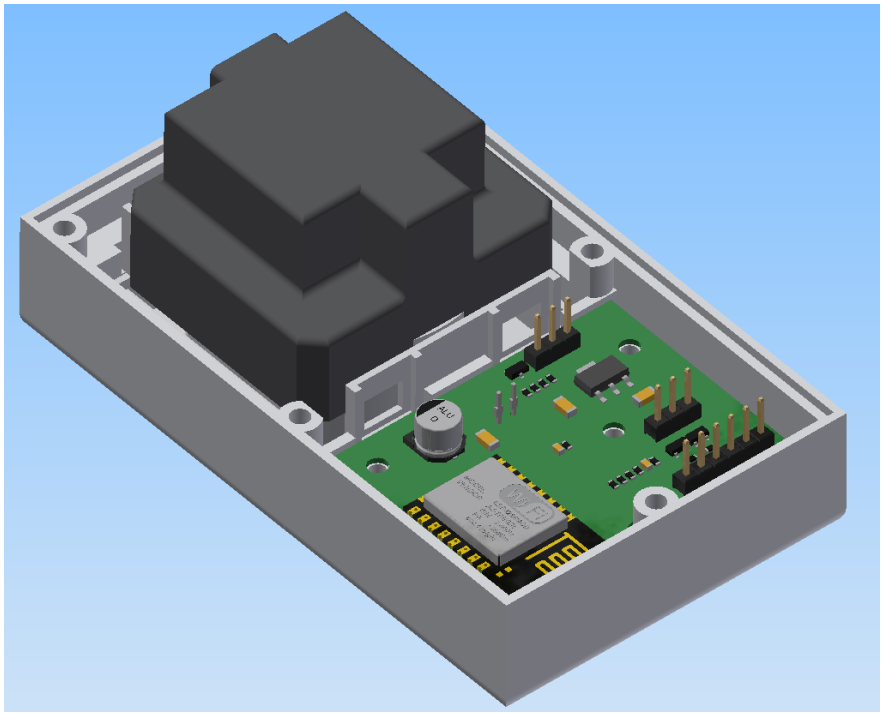
Kućište (Matija Nestić)

Za potrebe projekta dizajnirano je kućište "po mjeri" koje sadrži sve komponente koje se koriste u projektu. Kućište je dizajnirano oko utičnice, utikača i elektroničkih komponenti, a dizajnirano je pomoću Autodesk Inventora. Autodesk Inventor je moćan softverski alat za parametarsko 3D modeliranje kompleksnih sklopova, izradu prateće tehničke dokumentacije te renderiranje i mehaničke simulacije. Veći dio kućišta je napravljen pomoću 3D printera koji se koristio za izradu prednjeg i stražnjeg dijela kućišta te za izradu *power* tipkice. Stražnji dio kućišta sastoji se od stražnje maske izrađene na 3D printeru i gotovog utikača koji je kupljen i zalijepljen u kućište. Prednja strana kućišta sastoji se od gotove utičnice te prednje maske i tipkice. Utičnica je kao i utikač lijepljena na prednju masku. Kućište je zatvoreno vijcima za plastiku.

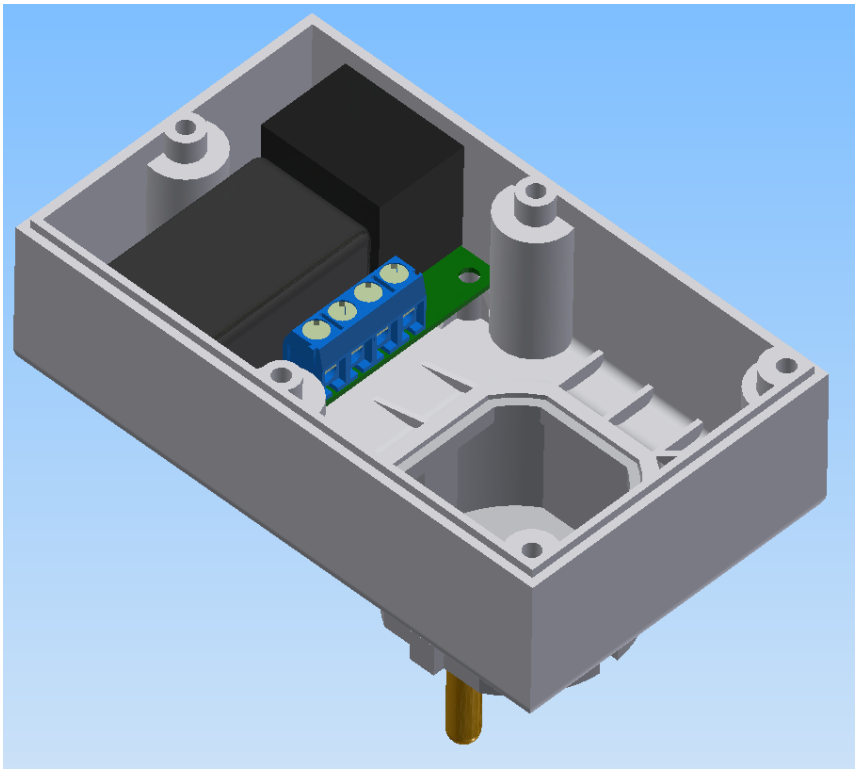
Slika 12 3D prikaz gotov kućišta



Slika 13 3D prikaz gotov kućišta



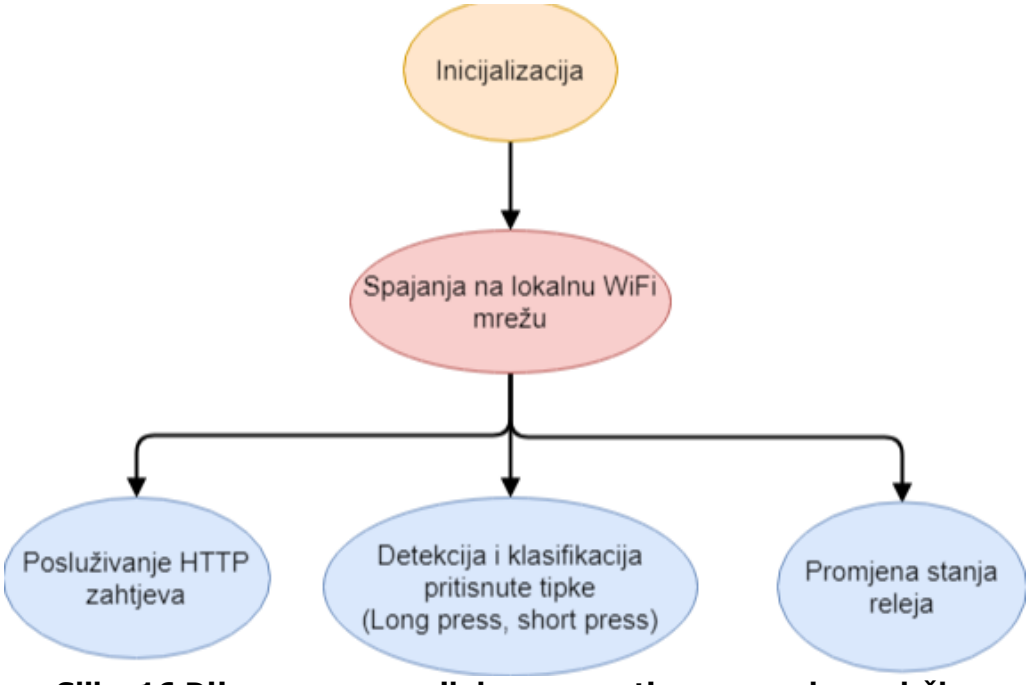
Slika 14 3D prikaz unutrašnjosti kućišta



Slika 15 3D prikaz unutrašnjosti kućišta

Programska podrška (Mato Lovrenović)

Programska podrška napisana je u Arduino IDE okruženju korištenjem biblioteka razvijenih za Arduino platformu. Kod se sastoji od nekoliko komponenti. To su inicijalizacija mikrokontrolera, spajanja na dostupnu WiFi mrežu, posluživanje http zahtjeva, detekcija vrste pritiska tipkala i agenta za uključivanje i isključivanje releja odnosno utičnice. Tijekom inicijalizacije mikrokontroler inicijalizira varijable i postavlja GPIO priključke na početne vrijednosti. Nakon toga započinje spajanje na WiFi mrežu. Ukoliko mikokontroler pronađe WiFi mrežu na koju se već spajao pokušat će se spojiti sa podacima koje ima spremljene u memoriji. Ako tijekom traženja dostupnih mreža ne postoji poznata mreža, stvara se AP sa nazivom "S2U2_" + zadnja 3 bytea mac adrese. Novostvorena mreža omogućava unos podataka za spajanje na neku od dostupnih WiFi mreža. Nakon unosa podataka i uspješnog spajanja na WiFi mrežu, mikrokontroler ulazi u beskonačnu petlju u kojoj poslužuje HTTP zahtjeve, provjerava stanje tipkala i upravlja se relejom i LED indikatorom.



Slika 16 Dijagram osnovnih komponenti programske podrške

1.2.6. Incijalizacija

Tijekom inicijalizacije mikrokontroler postavlja GPIO priključke u potrebna stanja. Inicijalizira dva izlaza za relej i LED indikator te dva ulazna priključka za očitavanje stanje tipkala i senzora blizine. Na kraju funkcije za inicijalizaciju periferije uključuje LED indikator kao signalizaciju da je uređaj uključen i da je završena faza inicijalizacije periferije.

Funkcija za inicijalizaciju periferije:

```
void initPerip (){
  //Init GPIOs direction
    pinMode(RELAY, OUTPUT);
    pinMode(LED, OUTPUT);
    pinMode(BTN, INPUT);
    pinMode(SENSE_PROXY, INPUT);

    //initial periph. states
    LedOn();
    RelayOff();
}
```

1.2.7. Spajanje na lokalnu WiFi mrežu

Spajanje na lokalnu WiFi mrežu vrši se pomoću biblioteke WiFiManager koju je razvio GitHub korisnik tzapu. Biblioteka stvara WiFi pristupnu točku sa konfiguracijskim portalom za unos podataka o WiFi mreži. Stvorena pristupna točka preusmjerava sve uređaje na adresu konfiguracijskog portala. Ako WiFiManager prilikom pokretanja pronađe WiFi mrežu koja se već koristila i podaci o njoj su spremljeni u memoriju, započinje sa automatskim spajanjem na tu WiFi mrežu. Biblioteka omogućuje dodavanje prilagođenih polja uz postojeća SSID i *password* polja. U ovom projektu je to iskorišteno za mogućnost promjene *hostnamea* mDNS usluge. Nakon spajanja na WiFi mrežu pokreće se web server i mDNS usluga.

Kod za konfiguraciju WiFi-a:

```
    //Wifi setup
    WiFiManager wifiManager;
    wifiManager.setAPCallback(wifiConfigModeCallback);

    byte mac [6];
    WiFi.macAddress(mac);
    friendlyName = "S2U2_" + String(mac[4], HEX) + String(mac[5], HEX) +
String(mac[6], HEX);
    WiFiManagerParameter hostName("server", "hostname",
friendlyName.c_str(), 40);
    wifiManager.addParameter(&hostName);


    for (int i=0; i<6; i++) Serial.println(mac[i]);
    String ssid =  friendlyName;
    char ssid_c [50] = {0};
    ssid.toCharArray(ssid_c, 50);
    if (!wifiManager.autoConnect(ssid_c, "12345678")){
      Serial.println("Connection timeout, RESET");
      ESP.reset();
    }
    //signal succesful wifi connection
    friendlyName = String(hostName.getValue());

    ledTicker.attach(0.2, LedTick);
    delay(1000);
    ledTicker.detach();
    LedOff();
    // print ip address for debuging
    Serial.println("ip addr");
    Serial.println(WiFi.localIP());
    //Wifi setup complete
```

1.2.8.

Detekcija i klasifikacija pritisnute tipke

Tipkalo se koristi kako bi korisnik mogao upravljati utičnicom bez spajanja na web server. Kako bi se spriječilo slučajno uključivanje i isključivanje trošila, uveden je dugi i kratki pritisak tipke. Dugi pritisak tipke isključuje uređaj ako uređaj trenutno radi. Kratki pritisak uključuje uređaj ako uređaj trenutno ne radi. Stanje tipkala se očitava svakih 100 ms (PB_POOL_RATE u ms). Kratki pritisak se definira kao pritisak tipke između 0,2 i 1,5 sekunde. Dugi pritisak je definiran od 1,5 do 10 sekundi. Klasifikacija se svodi na brojanje pozitivnih očitanja i vidljiva je u kodu.

Kod za detekciju i klasifikaciju stanja tipke

```
    if (readBTNpres){
      //btn is pressed
      cnt++;
    }else{
      if (cnt){
        if (cnt > PB_SHORT_PRESS_TIME * 1000/PB_POOL_RATE){
          if (cnt <= PB_LONG_PRESS_TIME * 1000/PB_POOL_RATE){
            //long press occured
            pbState = PB_LONG_PRESS;
          }else{
            //error timeout occured
            pbState = PB_NO_PRESS;
          }
        }else{
          //short press occured
          pbState = PB_SHORT_PRESS;

        }
      cnt = 0;
      }
    }
    delay(100);
  }
  private:
    uint8_t cnt;
    uint8_t pbState;
   public:
     unsigned int getPBstate(void){
        unsigned int temp = (unsigned int) pbState;
        pbState = PB_NO_PRESS;
        return temp;
   }
```

1.2.9. Agent za promjenu stanja

Utičnica može dobiti zahtjev za promjenu stanja iz raznih izvora. Zahtjev za promjenu stanja može doći putem weba, preko tipke ili zato jer trošilo više nije prisutno. Kako bi se izbjegla komplicirana struktura upravljanja releja i LED indikatora koristi se agent koji se poziva svaki put kada netko želi promijeniti stanje utičnice.

```
void socketAgent (){

  if (digitalRead(SENSE_PROXY)){
    if (socketState == STATE_ON){
      if (pbTask.getPBstate() == PB_LONG_PRESS){
        socketState = STATE_OFF;
      }
    }else{
      if (pbTask.getPBstate() == PB_SHORT_PRESS){
        socketState = STATE_ON;
      }
    }
  }else{
    //reset to default state
    if (socketState != STATE_OFF){
      socketState = STATE_OFF;
    }
  }

  if (socketState == STATE_ON){
    RelayOn();
    LedOn();
  }else{
    RelayOff();
    LedOff();
  }
}
```

1.2.10. Posluživanje HTTP zahtjeva

Jedan od načina upravljanja utičnicom je putem web preglednika. Korisnik se koristeći mDNS usluge spaja na web stranicu utičnice i od tamo može vidjeti trenutno stanje utičnice odnosno može vidjeti je li trošilo spojeno u utičnicu je li uključeno ili isključeno. Uz trenutno stanje može i upravljati utičnicom to jest slati zahtjev za uključivanje i isključivanjem utičnice. Web stranica napisana je u html kodu koji se jednim dijelom mijenja i generira ovisno o stanju utičnice. Zahtjevi za promjenu stanja prenose se GET metodom koristeći json tehnologiju.

Kod za obradu HTTP zahtjeva:

```
void httpRequest(){
  LedOn();
  int n = server.args();
  if (n) {
      String _argName = server.argName(0);
      String _argVal =  server.arg(0);
      if (_argName == "json") {
          StaticJsonBuffer<200> jsonBuffer;
          JsonObject& root = jsonBuffer.parseObject(_argVal);
          String state = root["state"];
          if (state == "on"){
            setState(STATE_ON);
          }else if (state == "off"){
            setState(STATE_OFF);
          }else{
            //error, reload page
          }
      }else{
          //error, reload page
      }
  }
  // Display the HTML web page
 String htmlMsg = "<!DOCTYPE html><html>\n";
        htmlMsg += "<head><meta name=\"viewport\" content=\"width=device-width,
initial-scale=1\">\n";
        htmlMsg += "<link rel=\"icon\" href=\"data:,\">\n";
        // CSS to style the on/off buttons
        // Feel free to change the background-color and font-size attributes to fit
your preferences
        htmlMsg += "<style>html { font-family: Helvetica; display: inline-block;
margin: 0px auto; text-align: center;}\n";
        htmlMsg += ".button { background-color: green; border: none; color: white;
padding: 16px 40px;\n";
        htmlMsg += "text-decoration: none; font-size: 60px; margin: 2px; cursor:
pointer;}\n";
        htmlMsg += ".button2 {background-color: red;}</style></head>\n";

        // Web Page Heading
        htmlMsg += "<body><h1>Pametna Uticnica</h1>\n";
        htmlMsg += "<body><h1>Ime: " + friendlyName + "</h1>\n";

        if (getState() == STATE_ON && digitalRead(SENSE_PROXY)){
          // Display current state, and ON/OFF buttons for GPIO 5
          htmlMsg += "<p> Utikac: prisutan </p>\n";
          htmlMsg += "<p> Stanje uticnice: Ukljuceno </p>\n";
          htmlMsg += "<p><a href=\"/?json={%22state%22:%22off%22}\"><button
class=\"button button2\">OFF</button></a></p>\n";
        }else{
          //current state = off
          htmlMsg += digitalRead(SENSE_PROXY)==HIGH ?  "<p> Utikac: prisutan
</p>\n" : "<p> Utikac : nije prisutan </p>\n" ;
          htmlMsg += "<p> Stanje uticnice: Iskljuceno</p>\n";
          htmlMsg += "<p><a href=\"/?json={%22state%22:%22on%22}\"><button
class=\"button\">ON</button></a></p>\n";
        }
        htmlMsg += "</body></html>\n";
        // The HTTP response ends with another blank line
        htmlMsg +="\n";
        server.send(200, "text/html", htmlMsg);
  LedOff();
}
```

Zaključak

Rezultat ovog projekta je prototip štedljive, sigurne i upravljive utičnice. Utičnica omogućava spajanje sa lokalnom WiFi mrežom, upravljanje trošilom preko Interneta te povećanu sigurnost korisnika.

Prednost S2U2 utičnice je povećana sigurnost i zaštita od strujnog udara. Koristi od utičnice će imati svi ljudi koji imaju potrebu upravljati sa trošilima sa većih udaljenosti te ljudi kojima je potrebna povećana sigurnost utičnica, kao npr. roditelji sa malom djecom.

Sljedeći koraci su stvaranje servera koji će moći upravljati sa svim S2U2 utičnicama na mreži sa jednog mjesta te skupljati podatke sa svih utičnica.

Literatura

[1] Arduino ticker, https://github.com/esp8266/Arduino/tree/master/libraries/Ticker

[2] ESP mDNS (13.6.2018.), https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266mDNS

[3] ESP8266 wiki (13.6.2018.), https://www.esp8266.com/wiki/doku.php

[4] IRM-02 datasheet (13.6.2018.), http://www.meanwell.com/Upload/PDF/IRM-02/IRM-02-SPEC.PDF

[5] NodeMCU wiki (13.6.2018.), https://en.wikipedia.org/wiki/NodeMCU

[6] SRD-05VDC Relej datasheet, http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf

[7] WiFi Manager (13.6.2018.), https://github.com/tzapu/WiFiManager

Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Arduino IDE	IDE za razvoj Arduino programske podrške	https://www.arduino.cc/en/Main/Software?
ESP8266 12E	Gotov WiFi modul sa antenom i mikrokontrolerom i TCP/IP stackom	https://en.wikipedia.org/wiki/ESP8266
WiFi…	Bežična lokalna mreža	https://en.wikipedia.org/wiki/Wi-Fi
mDNS	Multicast DNS	https://en.wikipedia.org/wiki/Multicast_DNS
HTTP	Hypertext Transfer Protocol	https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol