Aparat za pripravu napitaka

Kovačić, Ivan

Master's thesis / Diplomski 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:200:133640

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Aparat za pripravu napitaka

Diplomski rad

Ivan Kovačić

Osijek, 2018.

SADRŽAJ:

1. UVOD 1
1.1 Zadatak diplomskog rada 1
2. KORIŠTENE TEHNOLOGIJE 2
2.1 Arduino IDE 2
2.1.1 Izbornik 3
2.1.2 Brzi izbornik 3
2.1.3 Tekstualni editor 3
2.1.4 Konzola 4
2.2 OpenSCAD 5
2.3 Blynk aplikacija 6
3. KORIŠTENE KOMPONENTE 7
3.1 Node MCU 7
3.2 Istosmjerni (DC) motor 7
3.3 Vodena pumpa 8
3.4 Relej sklop 9
3.5 Uređaj za doziranje 10
3.6 Hall senzor 11
3.7 LMS2596S Step-down modul 12
4. IZRADA PROJEKTNOG ZADATKA 13
4.1 Montažna shema i popis komponenti 13
4.2 Sklopovski dio sustava 14
4.3 Programski dio sustava 17
5. ZAKLJUČAK 19
LITERATURA 20
SAŽETAK 21
ABSTRACT 22
ŽIVOTOPIS 23
PRILOZI 24
PRILOG A: Program za NodeMCU razvojnu maketu 24
PRILOG B: Skripta za kreiranje 3D objekta spirale 26
PRILOG C: Skripta za kreiranje 3D objekta dozera 28

1. UVOD

Zadatak ovog diplomskog rada je napraviti programsko i sklopovsko rješenje za temu “Udaljeno upravljani aparat za pripravu napitaka”. Cilj je napraviti maketu aparata koji će uz pomoć malog elektromotora i vodene pumpe miješati savršenu kombinaciju praha i vode za sočan napitak, te će se moći udaljeno upravljati pomoću mobilnog uređaja. Prikazat će se ponašanje određenih modula za što bolju komunikaciju uređaja s korisnikom.

1.1 Zadatak diplomskog rada

U ovom diplomskom radu je potrebno napraviti maketu aparata za pripravljanje napitaka. Potrebno je omogućiti udaljeno upravljanje aparatom. Potrebno je razmotriti moderne sustave za naplatu bankovnim transakcijama i kriptovalutama.

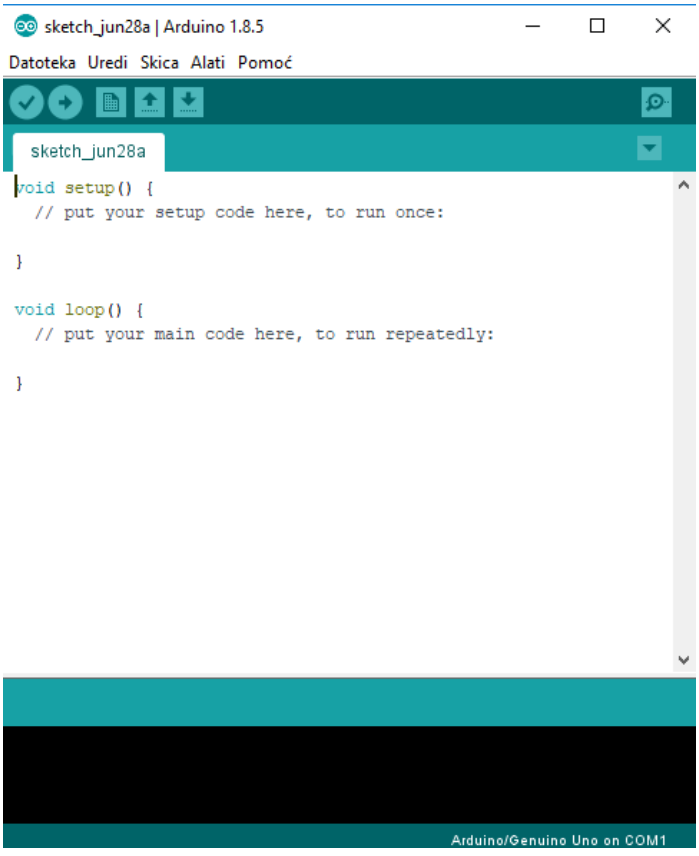
2. KORIŠTENE TEHNOLOGIJE

U procesu izrade projektnog zadatka korišteni su Arduino IDE programsko okruženje te alat za projektiranje potpomognuto računalom AutoCAD(Auto Computer Aided Design).

2.1 Aruino IDE

Arduino integrirano razvojno okruženje počiva na načelima otvorenog koda(engl. *open source)* što predstavlja softver čiji je izvorni kod dostupan svim korisnicima te im daje mogućnost izmjenjivanja, poboljšavanja i prepravljanja istog.[1] Omogućava pisanje programa u jeziku sličnom C-u i uvelike olakšava pisanje i izvršavanje koda na Arduino ili neku drugu pločicu. Sastoji se od nekoliko dijelova:[2]

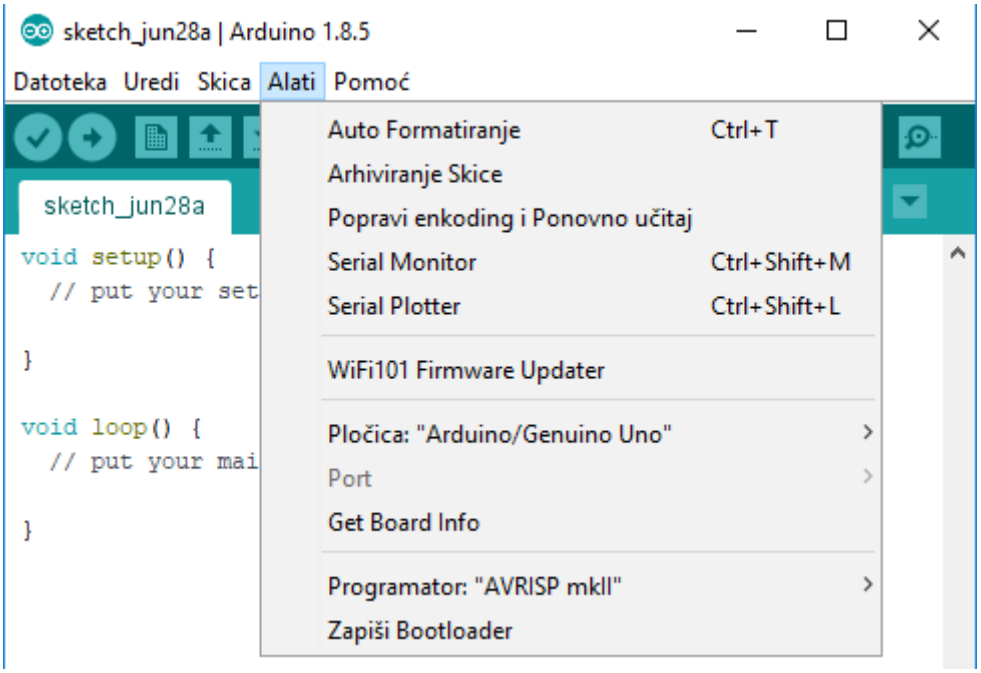
1. Izbornik
2. Traka s brzim pristupom
3. Tekstualni editor
4. Konzola



Slika 2.1. *Dijelovi Arduino IDE programskog okruženja*

2.1.1 Izbornik

Izbornik u Arduino IDE nalazi se na samom vrhu prozora i sadrži standardne funkcije za spremanje i dodatnu obradu programskog koda. Na početku korištenja sve postavke u izborniku su postavljene na *default* vrijednosti te se korigiraju sukladno s potrebama korisnika. Ovisno o pločici koja se koristi, u padajućem izborniku *tools* mijenjamo tip pločice, procesora na njima i broj COM porta koji je operacijski sustav dodijelio pločici[9].



Slika 2.2. *Prikaz padajućeg izbornika alati*

2.1.2 Brzi izbornik

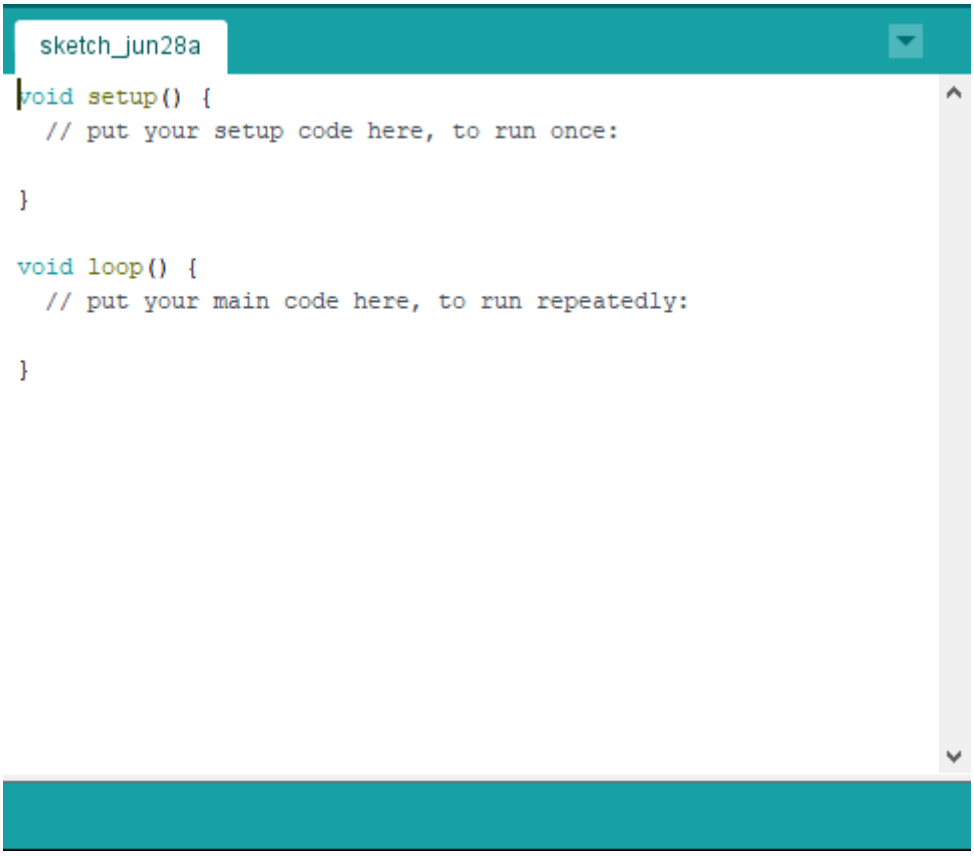
Traka brzog izbornika sadrži gumbe koji predstavljaju opcije koje su najčešće korištene u pisanju programskog koda: provjeri, prenesi, novo,otvori i pohrani. Ona omogućuje provjeru napisanoga koda, njegovo prenošenje na izabranu pločicu, kreiranje nove skicu za pisanje programskog rješenja, otvaranje postojećih te spremanje.

Slika 2.3. *Prikaz brzog izbornika*

2.1.3 Tekstualni editor

Glavni dio Arduino IDE razvojnog okruženja je tekstualni editor. Njegova glavna funkcija je omogućavanje upisa koda u posebni prozor. Napisani kod također se naziva i skica(engl. *sketch)* koja predstavlja skup instrukcija potreban za rješavanje određenog zadatka, a njeno ime se nalazi

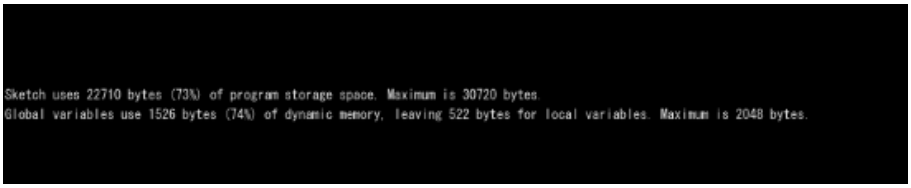
u gornjem lijevom dijelu tekstualnog editora. Za neko programsko rješenje moguće je imati veći broj skica, jednostavnijih i kompleksnijih, a uobičajena praksa je koristiti po jednu skicu za svaku posebnu zadaću koju program treba izvršiti.



Slika 2.4. *Prikaz tekstualnog editora*

2.1.4 Konzola

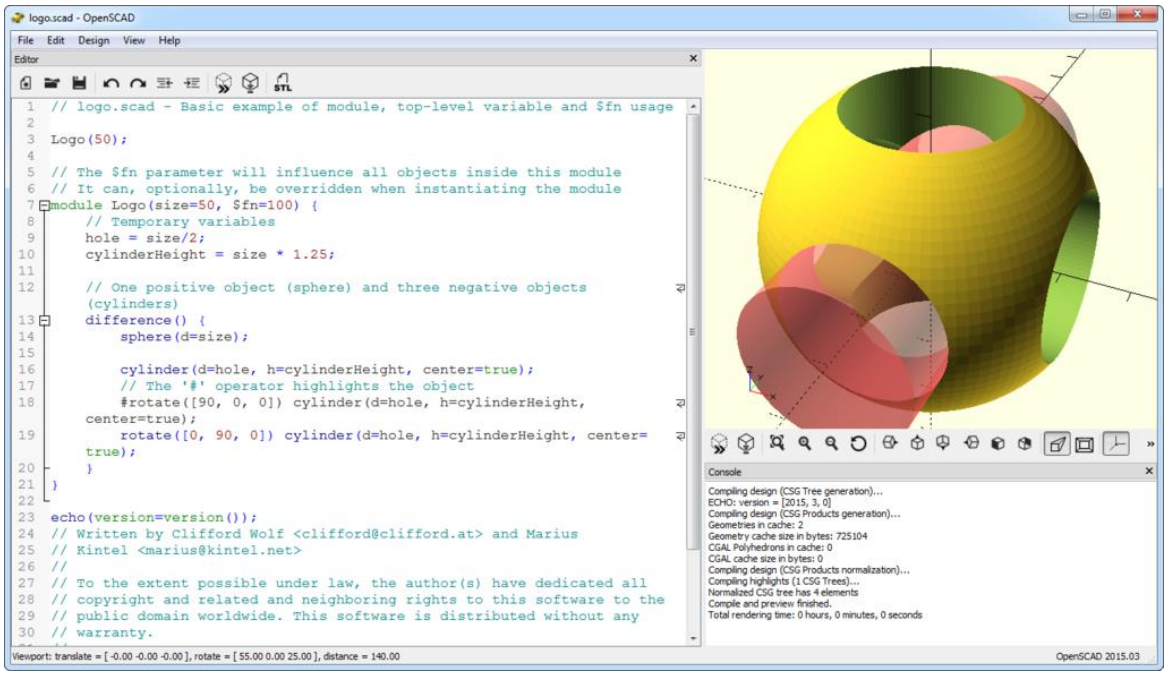
Nakon provjere napisanog koda izborom gumba *provjeri* u brzom izborniku, eventualne greške pokazat će se u konzoli. Tamo se nalaze i neke osnovne informacije o napisanom programskom kodu kao što je zauzimanje dinamički alocirane memorije na pločici, preostalog memorijskog prostora i slično. Nakon pritiska gumba *prenesi* na konzoli će se ispisati poruka kada je sav kod prenesen na pločicu.



Slika 2.5. *Prikaz ispisa na konzolu*

2.2 OpenSCAD

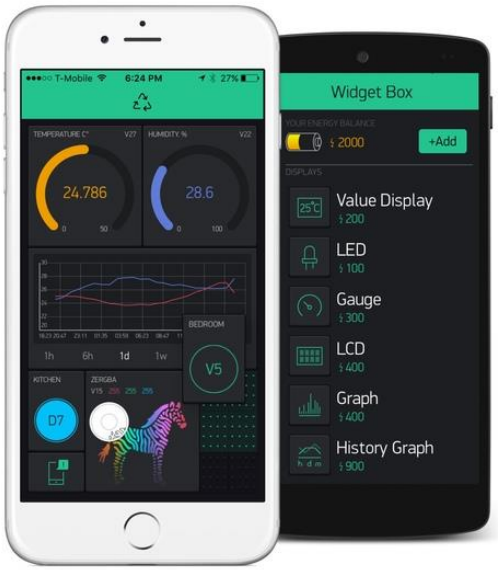
OpenSCAD je beslpatna programska aplikacija za kreiranje čvrstih 3D objekata. U suštini je skriptni modeler koji koristi svoj skriptni jezik.[3] Dijelovi mogu biti pregledani, ali se ne mogu interaktivno označiti i mijenjati pomoću miša u 3D pogledu. OpenSCAD skripte specificiraju geometrijske oblike (kao što su sfere, prizme, cilindri i slično) i definiraju kako se oni mijenjaju i kombiniraju kako bi se napravio 3D model. Kao takav, program radi konstruktivnu geometriju čvrstih tvari. U kontekstu ovog diplomskog rada korišten je u izradi 3D printanog dozera korištenog za precizno doziranje praha.



Slika 2.6. *Prikaz OpenSCAD korisničkog sučelja*

2.3 Blynk aplikacija

Blynk je platforma s Android i iOS aplikacijama za kontrolu Arduina, Raspberry Pi-a i još širokog spektra mikroupravljačkih sustava.[4] Blynk je zapravo digitalna upravljačka ploča (*engl. dashboard)* gdje se može praviti grafičko sučelje za projekt jednostavnim povlačenjem potrebnih stavki s izbornika na ekran. Vrlo je jednostavan za postavljanje i korištenje. Sve što je potrebno je preuzeti Blynk aplikaciju na mobilni uređaj gdje se kreira sučelje, kasnije se to sučelje poveže s maketom na način da se skine potrebna bibiloteka za rad s Blynk aplikacijom, te se upiše autentifikacijski kod koji se kreira kroz aplikaciju za specifično sučelje i spreman je za rad.



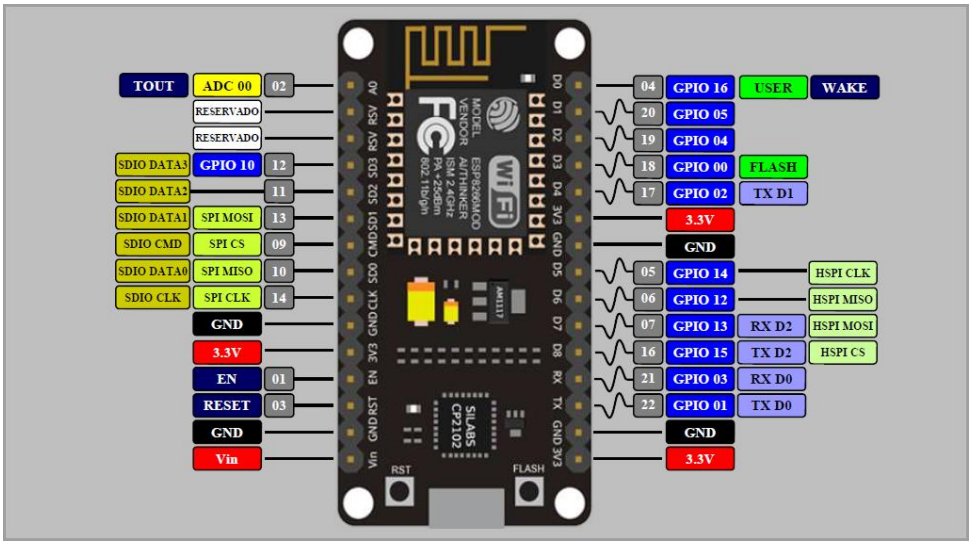
Slika 2.7. *Blynk aplikacija*

3. KORIŠTENE KOMPONENTE

U realizaciji rješenja projektnog zadatka korištene su mnogobrojne komponente kompatibilne sa Arduino IDE programskim sučeljem.

3.1 Node MCU

Node MCU je mikrokontroler baziran na ESP-12 modulu. [5] [6] Zbog svih stavki koje sadrži, a koji su ekvivalentni onima na računalu pripada kategoriji uređaja SoC(engl. System on Chip). Pruža pristup ulazno izlaznim priključnicama(engl. *GPIO-General Purpose Input/Output)* i sadrži WiFi modul.



Slika 3.1. *Konfiguracija Node MCU priključnica*

3.2 Istosmjerni (DC) motor

Istosmjerni motor pretvara istosmjernu električnu energiju u mehaničku energiju, odnosno kružno gibanje [7]. Najčešći tipovi oslanjaju se na silu proizvedenu magnetskim poljem. Gotovo svi istosmjerni motori imaju neki unutarnji mehanizam, bilo elektronički ili elektromehanički, koji povremeno mjenjaju smjer strujnog toka u dijelu motora. Koriste se u velikom broju automatiziranih sklopova, igračkama i sličnim sklopovima. U ovom projektu se koristi DC motor koji radi na naponu od 5V. Sastoji se od pozitivne i negativne priključnice, te osovine koja prenosi kružno gibanje.

Specifikacije istosmjernog motora:

• Napon: 3-6V
• Struja: 80mA (na 6V)
• Stall struja: 800mA (na 6V)
• Promjer osovine ~ 2mm
• Dužina osovine ~ 38mm

Slika 3.2. *Istosmjerni motor*

3.3 Vodena pumpa

Vodena pumpa je uređaj koji pomiče fluide (tekućine ili plinove) mehaničkom silom. Pumpa radi na rotirajućem mehanizmu, te troši energiju kako bi izvela mehanički rad potreban za pomjeranje tekućine. Mehaničke pumpe mogu biti potopljena u tekućine koju pumpaju ili stavljena izvan tekućine. Pumpa na slici je zapravo istosmjerni motor kombiniran s rotornim mehanizmom za pumpanje vode. Sastoji se od pozitivne i negativne priključnice za struju, te ulazne i izlazne cijevi za vodu.

Slika 3.3. *Vodena pumpa*

Specifikacije vodene pumpe:

- Napon: 12V DC
- Radni napon: 6-16V (niži napon, pritisak i protok se smanjuju)
- Struja: 1.2A
- Maksimalna visina pumpanja: 3.5 metra
- Protok: 3L/minuti
- Težina: 100 grama

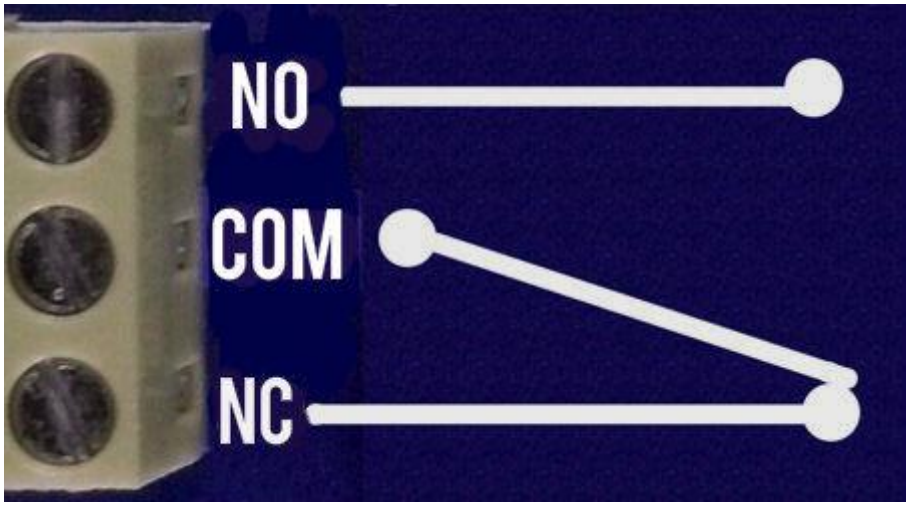
3.4 Relej sklop

Relej je električki upravljana sklopa koja je sposobna upravljati visokim naponima. [8] Može također i raditi na manjim naponima ukoliko je to potrebno. Na slici 3.4 vidljivi su VCC i GND priključci koji služe za napajanje, a priključak IN1 služi za upravljanje rada sklopke.

Slika 3.4. *Relej sklop*

Kanal ima 3 izlaza:

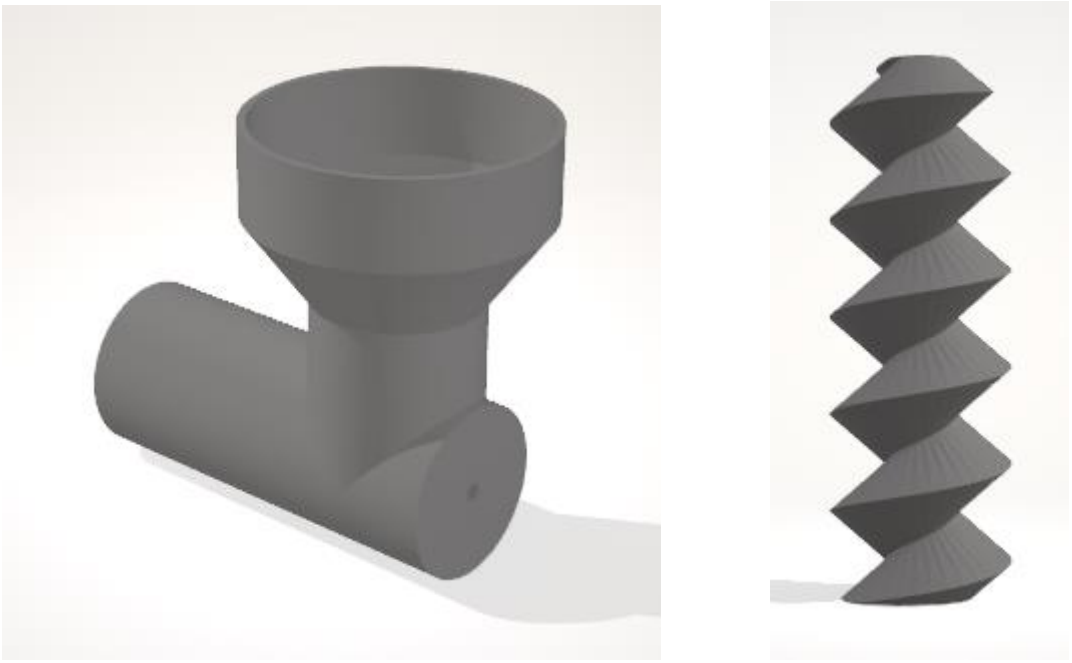
1. **NO**(eng. *normally open)* - normalno otvoren; ponaša se kao prekidač, i pošto je otvoren nema kontakta s COM. Kada se sklop aktivira, elektromagnetnom silom unutar releja se povežu NO i COM, se te zatvori strujni krug.
2. **COM**(eng. *common connection)* - zajednička konekcija
3. **NC**(eng. *normally closed)* - normalno zatvoren; uvijek je spojen s COM, čak i kada uređaj ne radi. Kada se sklopu dovede napon, prekine se veza između COM i NC, te se otvori strujni krug.



Slika 3.5. *Prikaz izlaza kanala*

3.5 Uređaj za doziranje

Kako bi omjer praha i vode bio precizno određen u stvaranju napitka, osmišljen je uređaj za doziranje. Model za 3D printanje istog izrađeni su u OpenSCADu i prikazani na slici 3.6. Model se sastoji od 2 dijela, spirale i njezinog spremnika kroz koji prah pada na spiralu, te ga se uz pomoć elektromotora izbacuje van u posudu.



Slika 3.6. *Model uređaja za doziranje u OpenSCAD-u*

3.6 Hall senzor

Za lakše korištenje uređaja osmišljen je sustav mjerenja količine vode u spremniku koji će javljati korisniku kada su zalihe vode pri kraju. Sastoji se od Hall senzora koji je postavljen na dnu spremnika, te plutajućeg magneta koji će prilikom manjka vode pasti na dno, te će se Hallov senzor oglasiti kako bi upozorio na nedostatak vode. Osim što nam to služi za obaviještavanje krajnjeg korisnika, služi i za očuvanje rada pumpe kako ona ne bi radila na "prazno".

Slika 3.7. *Prikaz plutajućeg magneta*

Hallov senzor služi kako bi očitao prisutnost magneta, te poslao električni signal. Sastoji se od 3 priključnice, napajanja, uzemljenja i izlaznom priključnicom koja šalje električni signal kada je magnet blizu sklopa.[8]

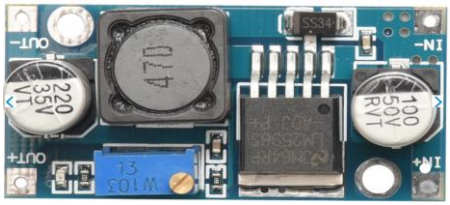
Slika 3.8. *Hallov sensor*

Specifikacije:

- Radni napon: 4.5V do 28V (uobičajeno 5V)
- Izlazni napon: jednaka ulaznom naponu
- Izlazna struja: 25mA
- Može uočiti oba pola magneta

3.7 LMS2596S Step-down modul

Ovaj modul ulazni napon snižava na točno određeni izlazni napon kojega podešavate putem trimera na pločici. Vrlo je prilagodiv jer promjenom ulaznog napona, izlazni ostaje uvijek isti, ali ipak ulazni napon mora barem za 0.7V biti veći od izlaznog. Nema zaštitnu diodu u slučaju pogrešnog polariteta, paziti pri spajanju! [9]



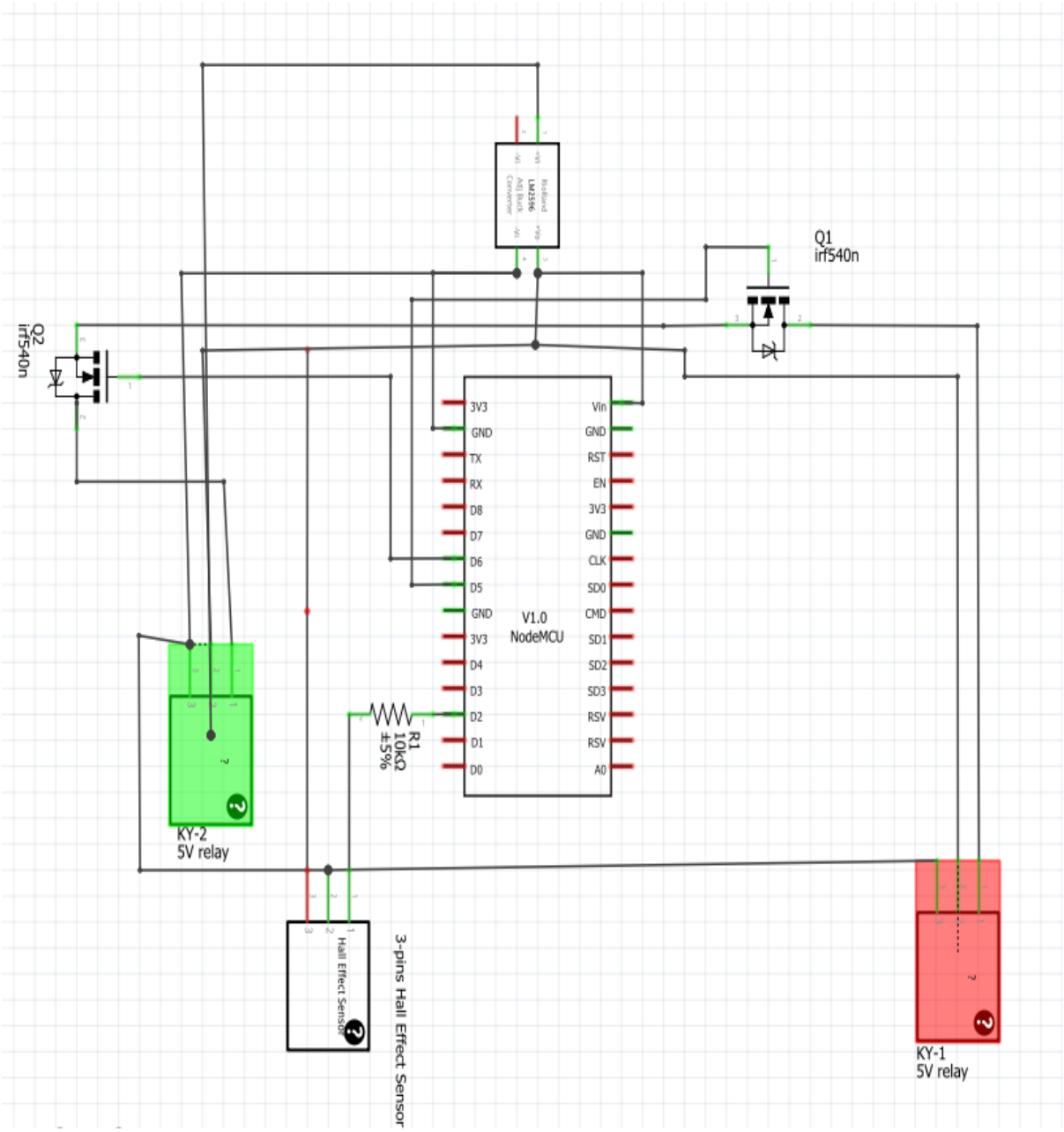
Slika 3.9 *Step-down modul*

Specifikacije:

• Ulaz: 3V - 40V
• Izlaz: 1.5V - 35V
• Max. struja na izlazu: 3A

4. IZRADA PROJEKTNOG ZADATKA

4.1 Montažna shema i popis komponenti



Slika 4.1. *Montažna shema spoja*

NodeMcu kao mikroupravljač je središnji dio ovog sklopa, kao što je i vidljivo na shemi. Za napajanje NodeMCU-a koristimo LM2596S Step-Down modul s pomoću kojega se snižava

početnih 12V na 5V potrebnih za rad mikroupravljača te releja. Priključci D5 i D6 iz mikroupravljača povezani su s relejima koji upravljaju radom vodene pumpe odnosno elektromotora za doziranje praha. Između releja i mikroupravljača postavljen je IRF540N transistor za lakše upravljanje. Na samom kraju nalazi se senzor Hallovog efekta koji služi za mjerenje razine vode u spremniku.

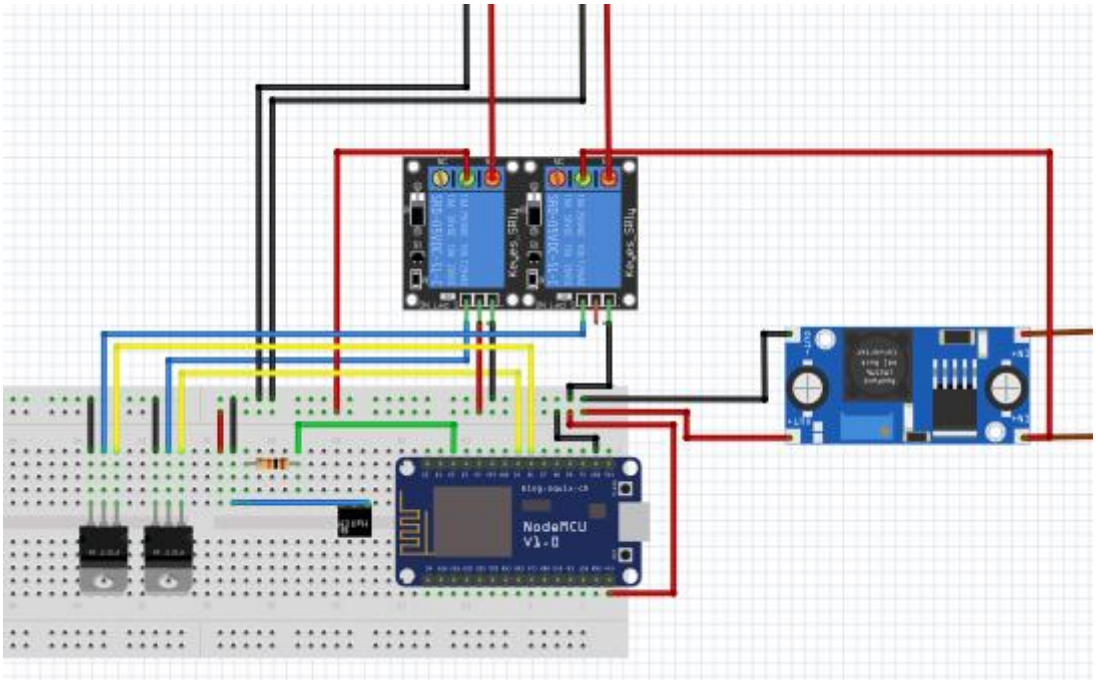
Popis komponenti:

- NodeMCU 1.0 (ESP-12E Modul)
- LM2596S Step-down modul
- IRF540n transistor
- 5V relej
- Hall efekt senzor
- DC motor
- Vodena pumpa
- 3D model dozera

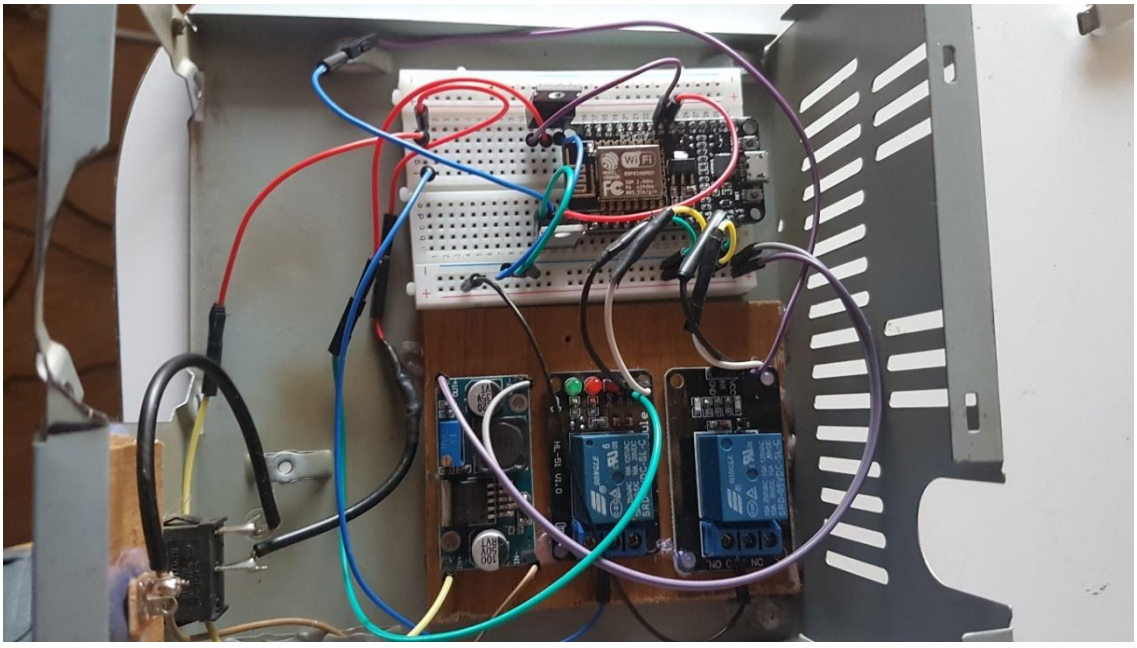
4.2 Sklopovski dio sustava

Sustav je spojen tako da je prvo doveden napon od 12V na *Step-down* modul koji spušta napon na 5V, te se taj napon grana na Vin priključak NodeMCU-a, Vcc priključke releja, te na COMM priključak jednog od releja. Na NC priključak tog istog releja je spojena anoda DC motora, dok je katoda spojena na GND. Na COMM priključak drugog releja dovodimo 12V, te NC priključak spajamo na anodu vodene pumpe, katodu također spajamo na GND. Između releja i mikroupravljača stoji IRF540n tranzistor koji omogućuje paljenje odnosno gašenje releja iz razloga što GPIO priključci na mikroupravljaču daju 0.2V čak i kada su postavljeni u stanje logičke "0" koje tranzistor poništava. GATE priključak od tranzistora spojen je u D5 odnosno D6 priključak za drugi relej, SOURCE priključak je spojen na GND, dok je DRAIN priključak spojen na IN kontrolni pin releja. Za kraj je ostao Hall senzor kojemu je Vcc priključak povezan na 3.3V, GND na GND, a SIGNAL na D2 priključak na mikroupravljaču.

Prilikom paljenja makete, pokreću se sve komponente, te mikroupravljač čeka naredbu od mobilne aplikacije kako bi krenuo s pravljenjem napitka, ukoliko nema vode u spremniku za vodu, Hallov senzor blokira izradu napitka kako se ne bi oštetila vodena pumpa.



Slika 4.2. *Model montažnog spoja u programu Fritzing*



Slika 4.3. *Unutrašnjost makete*

Slika 4.4. *Fotografija makete*

Slika 4.5. *Konačni proizvod*

4.3 Programski dio sustava

Programska logika ovog sustava je nakon inicijalizacije uređaja, mikroupravljač čeka na naredbu s Blynk mobilne aplikacije, dokle god se ne pritisne tipka za pravljenje napitka maketa će ostati u stacionarnom stanju. Maketa prima vrijednost s aplikacije pomoću sljedeće funkcije, te na isti način preuzima podešenu količinu praha koji će se ubaciti u napitak.

```
BLYNK_WRITE(V1){
butValue = param.asInt();
}
```

U programu je postavljena interrupt rutina koja se pokreće kada se detektira magnetsko polje na hall senzoru, odnosno kada magnetski plovak padne do hall senzora. Interupt rutina se postavlja na sljedeći način:

```
attachInterrupt(digitalPinToInterrupt(HALL_PIN), hallInterrupt, FALLING);
```

Nakon što je rutina pozvana, podiže se zastavica koja kasnije služi za onemogućivanje pravljenja napitka. U glavnom dijelu programa provjerava se je li pritisnuta tipka za pravljenje napitka, te također i ranije spomenuta zastavica, te ukoliko su uvjeti zadovoljeni poziva se funkcija za pravljenje napitka kojoj se kao parametar predaje količina praha.

```
if(butValue  == 1 && bInterrupt ==  0 ){     //napravi napitak
    makeSoda(sliderValue);
    butValue = 0;
  }
```

Prilikom ulaska u funkciju za pravljenje napitka provjerava se predana varijabla za količinu praha u napitku, te se sukladno tome postavlja vrijednost vremena koliko će raditi elektromotor, odnosno koliko će praha prenijeti u čašu. Nakon toga pali se elektromotor i pumpa za vodu, zatim se gasi motor za doziranje praha, te se oduzme njegovo vrijeme rada od najvećeg mogućeg vremena koji pumpa može raditi bez da voda prelije čašu, te se nakon toga gasi i pumpa za vodu. Prilikom završetka cijelog procesa na mobilni uređaj stiže poruka da je napitak završen.

```
void makeSoda(int powderAmount){
  long timeMs;
  switch(powderAmount){  //kolicina praha
      case 1: timeMs = 2000; break;     //2sec
      case 2: timeMs = 3000; break;     //3sec
      case 3: timeMs = 4000; break;     //4sec
    }
    digitalWrite(POWDER_PIN, HIGH);
    digitalWrite(WATER_PIN, HIGH);
    delay(timeMs);
    digitalWrite(POWDER_PIN, LOW);
    timeMs = MAX_WATER_TIME_MS - timeMs;
    delay(timeMs);
    digitalWrite(WATER_PIN, LOW);
    Blynk.notify("Cedevita je gotova");
}
```

5. ZAKLJUČAK

U ovom radu napravljena je maketa aparata za pripravu napitaka uz pomoć NodeMCu platforme koja je upravljanja putem Blynk mobilne aplikacije. Za izradu makete su se koristili NodeMCU mikroupravljački sklop, dva 1-kanalna releja, *step-down* modul, vodena pumpa, istosmjerni motor, te modelirani dozer. Prilikom izrade makete nastao je problema kod upravljanja relejima iz razloga što NodeMCU na svojim priključcima prilikom logičke nule ipak daje neki napon što je otežavalo rad s relejima, ali to se uspjelo riješiti spajanjem tranzistora između releja i mikroupravljača kako bi se pobrinio za taj problem.[10] Potrebno je napomenuti kako prilikom izrade makete treba biti oprezan iz razloga što komponente rade na različitim naponima i treba se pobrinuti da ne bi došlo do pregaranja istih.

LITERATURA

[1] **https://www.arduino.cc/**

Osnove o Arduino razvojnom okruženju (srpanj, 2018.)

[2] https://e-radionica.com/hr/blog/2015/08/19/arduino-ide-i-prvi-kod/

Rad s Arduino razvojnim okruženjem (srpanj, 2018.)

[3] **http://www.openscad.org/**

Upute za rad u openscadu (srpanj, 2018.)

[4] **https://www.blynk.cc/**

Rad s Blynk aplikacijom (rujan, 2018.)

[5] **http://www.nodemcu.com/index_en.html**

Općenito o NodeMCU (kolovoz, 2018.)

[6] **http://www.handsontec.com/pdf_learn/esp8266-V10.pdf**

Općenito o esp8266 chipu na NodeMCU (kolovoz, 2018.)

[7] **https://e-radionica.com/hr/istosmjerni-dc-motor.html**

Osnove rada istosmjernog motora (kolovoz, 2018.)

[8] **https://roboindia.com/tutorials/motor-relay-nodemcu**

Spajanje releja s NodeMcu (kolovoz, 2018.)

[9] **https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html**

Općenito o elektromagnetizmu i hall senzoru (kolovoz, 2018.)

[10] **https://e-radionica.com/hr/dc-dc-podesivo-napajanje-s-lm2596s.html**

Spuštanje napona (srpanj, 2018.)

[11] **https://www.infineon.com/cms/en/product/power/mosfet/20v-300v-n-channel-power-mosfet/80v-100v-n-channel-power-mosfet/irf540n/**

Općenito o mosfet tranzistoru n tipa (rujan, 2018.)

SAŽETAK

Naslov: Aparat za pripravu napitaka

U završnom radu obrađuje se izrada sklopovskog i programskog dijela za aparat za pripremu napitaka kojom se udaljeno upravlja pomoću mobilnog uređaja. Izrada se temelji na NodeMCU razvojnom sustavu koji služi kao upravljačka jedinica, dok nam ugrađeni Wifi modul omogućuje pristup maketi, a samim time i upravljanje istom.

Ključne riječi: NodeMCU, Wifi, vodena pumpa, relej, istosmjerni motor, dozer

ABSTRACT

Title: Juice maker

This paper elaborates on development of software and hardware of model remotely operated from a web page. The design is based on NodeMCU system which serves as a control unit, while embedded WiFi module allows us to access the model and control the entire system.

Key words: NodeMCU, Wifi, water pump, relay, DC motor, dosing system

ŽIVOTOPIS

Rođen 9. kolovoza 1994. godine u Osijeku od oca Željka i majke Janje. Živi s obitelji u Tenju, gdje pohađa i uspješno završava „OŠ Tenja“. Srednju školu upisuje III. Gimnaziju Osijek koju završava 2012. godine. Iste godine upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija, preddiplomski studij računarstva koji završava 2016. godine. Na istom fakultetu upisuje diplomski studij računalnog inženjerstva. Trenutno je zaposlenik Atos IT Soultion and Services u Zagrebu.

Ivan Kovačić

PRILOZI

PRILOG A: Program za NodeMCU razvojnu maketu

```
#define HALL_PIN   4  //D2
#define WATER_PIN  12 //D6
#define POWDER_PIN 14 //D5

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "75e69f261734410a8e67d647d1734ab4";
char ssid[] = "Aparat za napitke";
char pass[] = "Napitak123";

const long MAX_WATER_TIME_MS = 7000;
volatile bool bInterrupt;
int butValue;
int sliderValue = 2;

void hallInterrupt() {
  bInterrupt = true;
}

void setup() {
  pinMode(WATER_PIN, OUTPUT);
  pinMode(POWDER_PIN, OUTPUT);
  pinMode(HALL_PIN, INPUT);
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  attachInterrupt(digitalPinToInterrupt(HALL_PIN), hallInterrupt, FALLING);
}

void loop() {
  Blynk.run();
  if(butValue  == 1 && bInterrupt ==  0 ){    //napravi napitak
    makeSoda(sliderValue);
    butValue = 0;
  }
  if (bInterrupt) {
    bInterrupt = false;
```

```
  }
}

void makeSoda(int powderAmount){
  long timeMs;
  switch(powderAmount){  //kolicina praha
      case 1: timeMs = 2000; break;     //2sec
      case 2: timeMs = 3000; break;     //3sec
      case 3: timeMs = 4000; break;     //4sec
    }
    digitalWrite(POWDER_PIN, HIGH);
    digitalWrite(WATER_PIN, HIGH);
    delay(timeMs);
    digitalWrite(POWDER_PIN, LOW);
    timeMs = MAX_WATER_TIME_MS - timeMs;
    delay(timeMs);
    digitalWrite(WATER_PIN, LOW);
    Blynk.notify("Napitak je gotov");
}

BLYNK_WRITE(V0){
  sliderValue = param.asInt();
}

BLYNK_WRITE(V1){
  butValue = param.asInt();
}
```

PRILOG B: Skripta za kreiranje 3D objekta spirale

```
PI=3.141592;

module screw_thread(od,st,lf0,lt,rs,cs){
    or=od/2;
    ir=or-st/2*cos(lf0)/sin(lf0);
    pf=2*PI*or;
    sn=floor(pf/rs);
    lfxy=360/sn;
    ttn=round(lt/st+1);
    zt=st/sn;

    intersection(){
        if (cs >= -1){
           thread_shape(cs,lt,or,ir,sn,st);
        }
        full_thread(ttn,st,sn,zt,lfxy,or,ir);
    }
}

module thread_shape(cs,lt,or,ir,sn,st){
    if ( cs == 0 ){
        cylinder(h=lt, r=or, $fn=sn, center=false);
    }
    else{
        union(){
            translate([0,0,st/2])
              cylinder(h=lt-st+0.005, r=or, $fn=sn, center=false);

            if ( cs == -1 || cs == 2 ){
                cylinder(h=st/2, r1=ir, r2=or, $fn=sn, center=false);
            }
            else{
                cylinder(h=st/2, r=or, $fn=sn, center=false);
            }

            translate([0,0,lt-st/2])
            if ( cs == 1 || cs == 2 ){
                  cylinder(h=st/2, r1=or, r2=ir, $fn=sn, center=false);
            }
            else{
                cylinder(h=st/2, r=or, $fn=sn, center=false);
            }
        }
    }
}

module full_thread(ttn,st,sn,zt,lfxy,or,ir){
  if(ir >= 0.2){
    for(i=[0:ttn-1]){
        for(j=[0:sn-1])
                assign( pt = [    [0,                   0,
i*st-st            ],
                      [ir*cos(j*lfxy),     ir*sin(j*lfxy),     i*st+j*zt-st
],
                      [ir*cos((j+1)*lfxy), ir*sin((j+1)*lfxy),
i*st+(j+1)*zt-st   ],
                                           [0,0,i*st],
```

```
                        [or*cos(j*lfxy),     or*sin(j*lfxy),     i*st+j*zt-
st/2     ],
                        [or*cos((j+1)*lfxy), or*sin((j+1)*lfxy),
i*st+(j+1)*zt-st/2 ],
                        [ir*cos(j*lfxy),     ir*sin(j*lfxy),     i*st+j*zt
],
                        [ir*cos((j+1)*lfxy), ir*sin((j+1)*lfxy),
i*st+(j+1)*zt      ],
                        [0,                  0,                  i*st+st
]     ]){
            polyhedron(points=pt,
                          triangles=[     [1,0,3],[1,3,6],[6,3,8],[1,6,4],

      [0,1,2],[1,4,2],[2,4,5],[5,4,6],[5,6,7],[7,6,8],

      [7,8,3],[0,2,3],[3,2,7],[7,2,5]     ]);
        }
    }
  }
}

difference(){
screw_thread(17,9,40,50,PI/2,1);
cylinder(h=20, d=2, $fn=100);
}
```

PRILOG C: Skripta za kreiranje 3D objekta dozera

```
difference(){
    translate(v=[0,0,30]) cylinder(d=35, h=10, $fn=1000);
    translate(v=[0,0,30]) cylinder(d=33, h=11, $fn=100);
}
difference(){
    translate(v=[0,0,20]) cylinder(h=10, d1=20, d2=35, $fn=1000);
    translate(v=[0,0,20]) cylinder(h=10, d1=18, d2=33, $fn=100);
}

difference(){
    difference(){
        cylinder(h=20, d=20, $fn=1000);
        cylinder(h=20, d=18, $fn=1000);
    }
    translate(v=[0,11,0]) rotate(a=[90,0,0]) cylinder(h=50, d=18, $fn=1000);
}

difference(){
    difference() {
        difference() {
            translate(v=[0,11,0]) rotate(a=[90,0,0]) cylinder(h=50, d=20,
$fn=1000);
            translate(v=[0,9,0]) rotate(a=[90,0,0]) cylinder(h=50, d=18,
$fn=1000);
        }
        translate(v=[0,12,0]) rotate(a=[90,0,0]) cylinder(h=50, d=2,
$fn=1000);
    }
    cylinder(h=20,d=18, $fn=1000);
}
```