

# Automatizirani sustav za uzgoj biljaka

---

Fiala, Ivan

Undergraduate thesis / Završni rad

2020

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://um.nsk.hr/um:nbn:hr:200:084336>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2024-09-20**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

**Stručni studij**

**AUTOMATIZIRANI SUSTAV ZA UZGOJ BILJAKA**

**Završni rad**

**Ivan Fiala**

**Osijek, 2020**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 12.09.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za završni ispit  
na preddiplomskom stručnom studiju**

<b>Ime i prezime studenta:</b>	Ivan Fiala
<b>Studij, smjer:</b>	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
<b>Mat. br. studenta, godina upisa:</b>	R3767, 23.09.2019.
<b>OIB studenta:</b>	06924935254
<b>Mentor:</b>	Izv. prof. dr. sc. Ivan Aleksi
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Prof.dr.sc. Željko Hocenski
<b>Član Povjerenstva 1:</b>	Izv. prof. dr. sc. Ivan Aleksi
<b>Član Povjerenstva 2:</b>	Izv. prof. dr. sc. Alfonzo Baumgartner
<b>Naslov završnog rada:</b>	Automatizirani sustav za uzgoj biljaka
<b>Znanstvena grana rada:</b>	<b>Arhitektura računalnih sustava (zn. polje računarstvo)</b>
<b>Zadatak završnog rada</b>	(temu rezervirao Ivan Fiala) Potrebno je primijeniti Arduino mikroupravljači sustav za prikupljanje parametara zemlje i automatizirano navodnjavanje biljaka.
<b>Prijedlog ocjene pismenog dijela ispita (završnog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	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
<b>Datum prijedloga ocjene mentora:</b>	12.09.2020.
<i>Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:</i>	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 28.09.2020.

**Ime i prezime studenta:**

Ivan Fiala

**Studij:**

Preddiplomski stručni studij Elektrotehnika, smjer Informatika

**Mat. br. studenta, godina upisa:**

R3767, 23.09.2019.

**Turnitin podudaranje [%]:**

6

Ovom izjavom izjavljujem da je rad pod nazivom: **Automatizirani sustav za uzgoj biljaka**

izrađen pod vodstvom mentora Izv. prof. 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:

# Sadržaj

1. UVOD .....	1
2. DIZAJN AUTOMATSKOG SUSTAVA NAVODNJAVANJA .....	2
2.1. Postojeće metode navodnjavanja .....	2
2.2. Odabir pogodne metode i zahtjevi sustava .....	4
3. KOMPONENTE SUSTAVA I SHEMA .....	5
3.1. Komponente sustava .....	5
3.1.1. Elektro-magnetni ventil .....	5
3.1.2. Senzori .....	6
3.1.3. Arduino MEGA 2560 upravljačka jedinica .....	8
3.1.4. Ostale elektroničke komponente .....	10
3.2. Konačna shema sklopa .....	15
4. KONAČNA REALIZACIJA .....	17
4.1. Početno spajanje komponenti .....	17
4.2. Kodiranje sklopa .....	18
4.3. Konačno testiranje .....	21
4.4. Rezultati .....	24
5. ZAKLJUČAK .....	25
LITERATURA: .....	26
SAŽETAK .....	27
SUMMARY .....	28
PRILOZI .....	29

## 1. UVOD

Za automatizirani sustav uzgoja biljaka najvažnije je da osiguramo dobar automatski sustav navodnjavanja. Navodnjavanje je postupak unosa dodatnih količina vode u tlo kako bi se održao ili pospješio rast biljke. Navodnjavanje se koristi već tisućljećima u raznim civilizacijama. Znakoviti primjer je civilizacija na području Mezopotamije u trećem tisućljeću pr.n.e; oni su gradili kanale koji su činili mrežu u obliku matrice sa poljima usjeva između njih. Koristeći taj način usmjeravali su vodu što im je omogućavalo uzgoj na područjima puno dalje od rijeka nego što je to prije bilo moguće.

Napretkom tehnologije postiglo se da se puno veće površine mogu navodnjavati, također je omogućeno da te površine budu i puno dalje od izvora vode nego u prošlosti, međutim i dalje postoje problemi prekomjernog navodnjavanja nekog područja što smanjuje učinkovitost. Prekomjerno navodnjavanje osim što povećava potrošnju vode koja je dragocjen resurs, također može uzrokovati i plodno tlo za rast korova, razvoj gljivica zbog truljenja korijena i slično.

Zadatak ovog rada je dizajnirati adekvatan sustav koji osigurava da vlažnost tla uvijek bude dovoljno optimalna za rast željene biljke, a istovremeno smanjiti prekomjernu potrošnju vode koja može imati neželjene posljedice.

## **2. DIZAJN AUTOMATSKOG SUSTAVA NAVODNJAVANJA**

### **2.1. Postojeće metode navodnjavanja**

Načini navodnjavanja koji su se razvili kroz ljudsku povijest mogu se svrstati u četiri metode[1]:

- 1) Površinsko navodnjavanje
- 2) Podzemno navodnjavanje
- 3) Navodnjavanje kišenjem
- 4) Lokalizirano navodnjavanje

Površinsko navodnjavanje je najstarija metoda navodnjavanja u svijetu, ujedno sa 60% navodnjavanih površina je i najčešća metoda. Kroz povijest najčešće je bila upotrebljavana na način da su se sezonskim poplavama navodnjavale površine koje se nalaze direktno uz rijeku. Glavna karakteristika ove metode je nanošenje vode u tankom sloju po površini. U današnje vrijeme to se najčešće dovodi gravitacijom iz povišenih spremnika, ali moguće je i dovođenje pod tlakom. Ovo je metoda koja troši najveću količinu vode, odnosno njezinom primjenom dolazi do najvećih gubitaka.

Podzemno navodnjavanje je jedna od novijih metoda navodnjavanja. Vršiti se podzemnim cijevima koje vodu dovode direktno u korijenski sustav. Također omogućava lakšu obradu tla i prolazak ljudi i strojeva. Međutim ova metoda je vrlo skupa za postavljanje jer zahtijeva složeniju opremu.

Navodnjavanje kišenjem je metoda kojom se pokušava imitirati efekt kiše. Izvršava se na način da se voda kroz cijevi sa visokim pritiskom od minimalno 7 bara dovodi do raspršivača koji zatim vodu izbacuju u obliku kapljica po poljoprivrednoj površini. Sve kulture se mogu na ovaj način zalijevati. Određeni nedostaci postoje prilikom vjetra jer dolazi do neravnomjernog zalijevanja, također pri visokim temperaturama ukoliko je velika razlika u temperaturi vode i temperaturi na površini biljke može doći do oštećenja biljke što pogoduje nastanku biljnih bolesti. Iako manje rastrošna od površinskog navodnjavanja, gubitci i dalje nisu zanemarivi zbog mogućnosti ispravanja koje se često događa kao i zbog toga što se pokrivaju dijelovi površine gdje nije uzgajana kultura. Ne smiju se zanemariti ni trošci energije potrebni za postizanje tlaka potrebnog za raspršivanje.



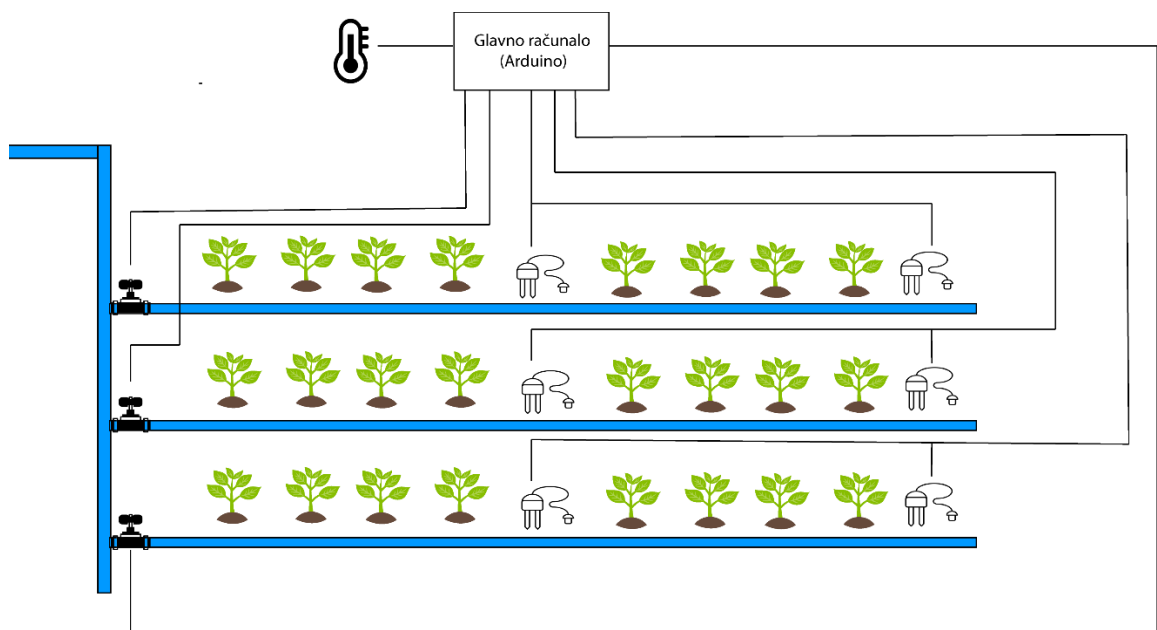
**Slika 2.1:** *Lokalizirano navodnjavanje voća.*

Lokalizirano navodnjavanje je metoda kojom se voda dovodi do korijena svake biljke, precizno i štedljivo. Ovo je najučinkovitija metoda čak i kad se uzmu u obzir troškovi postavljanja koji nisu zanemarivi, međutim velika ušteda u količini vode zbog lokalizirane primjene i mala količina energije potrebna za rad znatno smanjuju troškove kroz duži period uporabe. Ova metoda je pogodna za održavanje vlage tla u određenim intervalima. Također pogodna je za primjenu u svim klimama i svim topografijama i tipovima tla. Dijeli se na navodnjavanje kapanjem i navodnjavanje mini rasprskivačima. Kod navodnjavanja kapanjem voda se dovodi cijevima do svake biljke i vlaži vrlo mali dio zemljišta, dok mini rasprskivači djeluju slično ali umjesto kapaljki imaju male rasprskivače kod biljke koji raspršuju vodu u obliku kapljica kao i kod navodnjavanja kišenjem, ali sa tlakom do 3,5 bara i posljedično manjim dometom.



## 2.2. Odabir pogodne metode i zahtjevi sustava

S obzirom da je svrha ovog rada automatizacija sustava za uzgoj sa što većom učinkovitošću vidimo kako nam je najpogodnija metoda za razvoj ona lokaliziranog navodnjavanja. Oba sustava u lokaliziranom navodnjavanju su prigodna za automatski uzgoj, međutim zbog lakše izvedbe i lakše kontrole odabiremo navodnjavanje kapanje odnosno tzv. kap po kap sustav. Tokom rada sustava bitno je da senzor vlage tla može dobro odrediti vlagu kako bi na vrijeme pokrenuo navodnjavanje i zaustavio navodnjavanje kada je zemlja dovoljno natopljena. O tome treba voditi računa prilikom izrade načelne sheme sustava, radi lakše kontrole i neometanog rasta biljke, senzor se može postaviti na zasebno mjesto dalje od same biljke, također trebalo bi koristiti više senzora i uzimati njihovu srednju vrijednost, jer nisu sva mjesta jednako izložena svim čimbenicima koji mogu dovesti do smanjivanja vlage. Također vrlo je bitno da se ne zalijeva tokom velikih vrućina tokom dana jer u tom slučaju većina vode će ispariti stoga je potreban i senzor temperature koji će služiti da onemogući navodnjavanje pri visokim temperaturama. Za propuštanje vode ćemo koristiti magnetski ventil. Zbog njega za sustav nam je dovoljno spajanje na vodovodnu mrežu, na taj način se štedi energija koja bi bila potrošena na rad pumpe ukoliko bi vodu crpili iz nekog bunara ili spremnika. Također, njegovim korištenjem možemo kontrolirati zalijevanje za svaki red posebno, to nam omogućuje dodatnu uštedu vode.



Slika 2.2: Prikaz organizacije sustava.

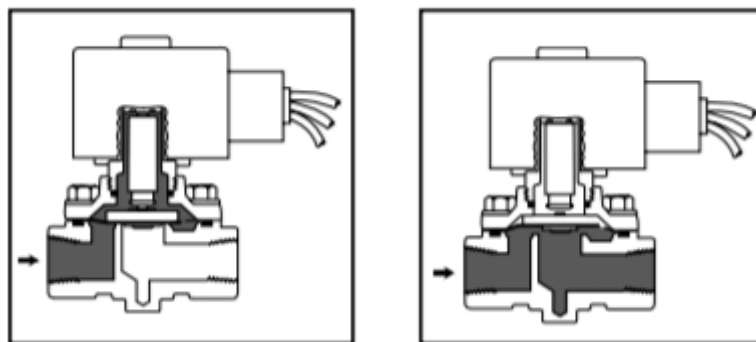
### 3. KOMPONENTE SUSTAVA I SHEMA

#### 3.1. Komponente sustava

Za izvedbu sustava ćemo koristiti elektromagnetni ventil, senzor za temperaturu, senzor za vlagu tla, mikrokontroler i dodatne elektroničke komponente potrebne za funkcioniranje sustava poput otpornika i tranzistora.

##### 3.1.1. Elektro-magnetni ventil

Elektromagnetni ventil je elektromehanička komponenta koja se koristi kao element za kontrolu protoka fluida. Razlikujemo normalno otvoren i normalno zatvoren ventil. Normalno zatvoren ventil je ventil koji sprječava protok fluida sve dok se ne dovede napon do njega, dok kod normalno otvorenog je obrnuta situacija[2]. Osim po ovoj podijeli razlikuju se i u karakteristikama električne struje koju koriste, jačini magnetskog polja koje stvaraju, mehanizmu koji koriste za regulaciju fluida te vrsti i karakteristikama fluida kojim upravljaju. Elektromagnetni ventili imaju zadan smijer protoka i na to se mora pripaziti u suprotnom ventil neće raditi i može čak doći do oštećenja



**Slika 3.1:** Dijagram rada normalno zatvorenog elektromagnetnog ventila bez i sa djelovanjem napona[3].

Za potrebe ovog projekta koristit ćemo se elektromagnetnim ventilom kojem je potreban istosmjerni napon od 12 V, on može podnijeti pritisak od 0,8 MPa što je i više nego dovoljno za vrtno crijevo na koje ćemo ga spajati (Slika 3.2). Također kako bismo smanjili potrošnju električne energije na minimum koristit ćemo normalno zatvoren ventil koji ćemo otvarati električnim signalom na kratke intervale kad za to postoje uvijeti.



**Slika 3.2:** *Elektromagnetni ventil.*

### 3.1.2. Senzori

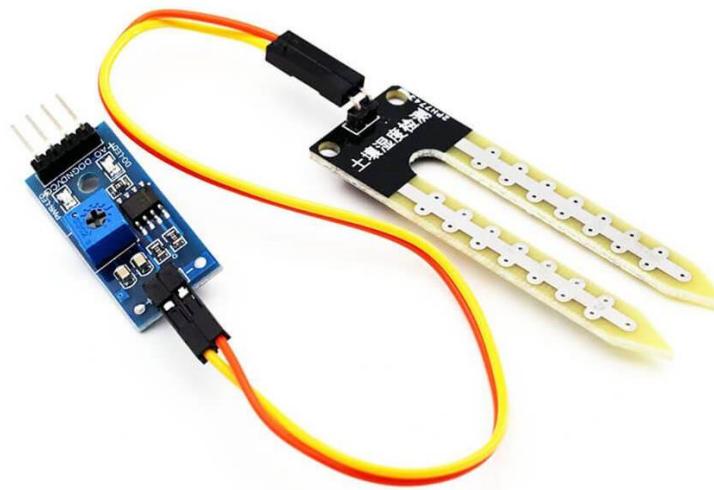
Senzor je uređaj koji mjeri neku fizikalnu veličinu i pretvara ju u signal pogodan za čitanje i daljnje upravljanje procesima. Od senzora ćemo koristiti senzore za toplinu zraka i senzor za vlažnost tla.

DHT 11 je osnovni, ultra jeftin digitalni senzor temperature i vlage. Koristi kapacitivni senzor vlage i termistor za mjerenje okolnog zraka i šalje digitalni signal na podatkovni pin[4]. Ima 4 pina i potreban je otpornik od 4,7 k $\Omega$  za pull up sa mikrokontrolera međutim često se može naći u izvedbama gdje je već spojen na plčicu sa dodanim otpornikom što olakšava spajanje(Slika 3.3).



**Slika 3.3:** *DHT 11.*

Prema slici 3.3 vidimo da imamo 3 pina za spajanje. Pin VCC možemo spajati direktno na napajanje ili na odgovarajući izlaz na mikrokontroleru. Pin GND se može spajati na masu jer GND znači uzemljenje, ali se isto tako može spojiti na odgovarajući izlaz na mikrokontroleru. Pin D<sub>out</sub> često označen i kao samo D se spaja na digitalni ulaz na mikrokontroleru i on nam služi za komunikaciju odnosno uzimanje vrijednosti sa senzora. Osim što nam daje temperaturu ovaj senzor daje i vrijednost vlage što ostavlja dodatne mogućnosti usaršenja sustava koji bi uzimao u obzir i vlagu zraka, a također daje i mogućnost dodavanja ventilatora ukoliko se radi o plastenicima koji bi onda održavali i konstantnu vlagu zraka.

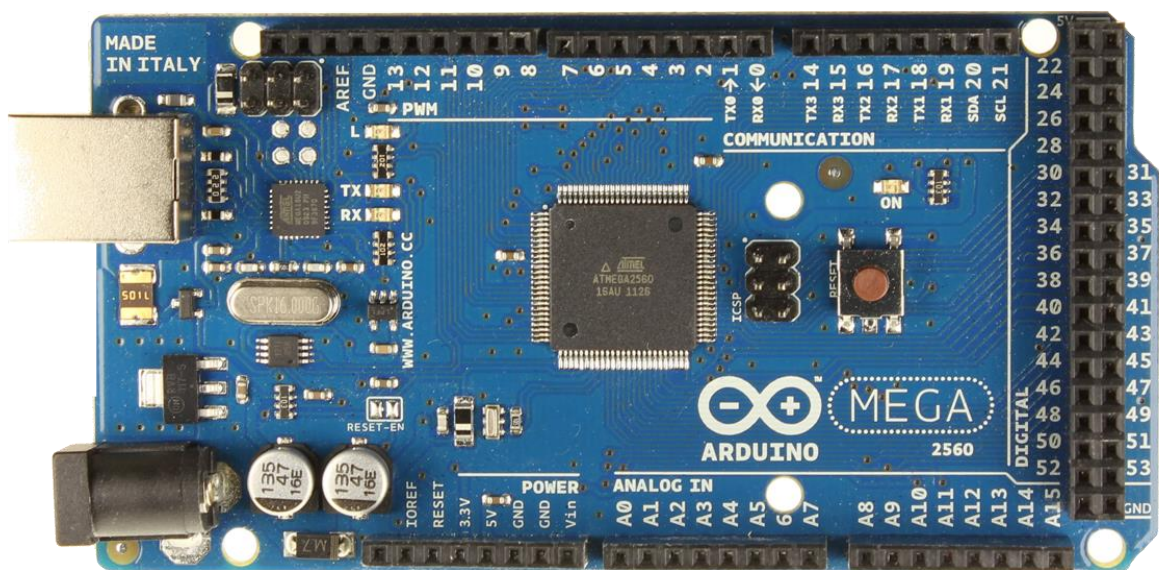


**Slika 3.4:** *Senzor vlažnosti tla FC-28.*

Senzor vlage tla upotrebljava kapacitet za mjerenje dielektrične propusnosti okolnog medija. U tlu je dielektrična propusnost funkcija sadržaja vode. Senzor stvara napon proporcionalan dielektričnoj propusnosti, a samim tim i sadržaju vode u tlu. Jedan od glavnih problema ovakvih senzora je njihov relativno kratak životni vijek zbog konstantne izloženosti vanjskim uvjetima, posebice vlazi. Zbog protoka DC struje senzor će polako gubiti bakreni sloj te na kraju neće više moći provoditi struju, odnosno mjeriti vlažnost tla. Ovaj problem se može riješiti tako da ne propuštamo konstantno struju kroz njega već da senzor se aktivira samo povremeno kada nam je potrebno njegovo mjerenje, to izvodimo spajanje VCC pina na neki drugi pin na mikrokontroleru umjesto naponskog pina. Na ovaj način iako će senzor korodirati, to neće biti tako brzo kao po normalnom principu spajanja, i to je zbog svoje jeftinoće trošak koji se može lako podnijet svakih nekoliko mjeseci, naravno ukoliko su potrebna česta mjerenja postoje i otporniji senzori koji mogu zamijeniti ovaj[5].

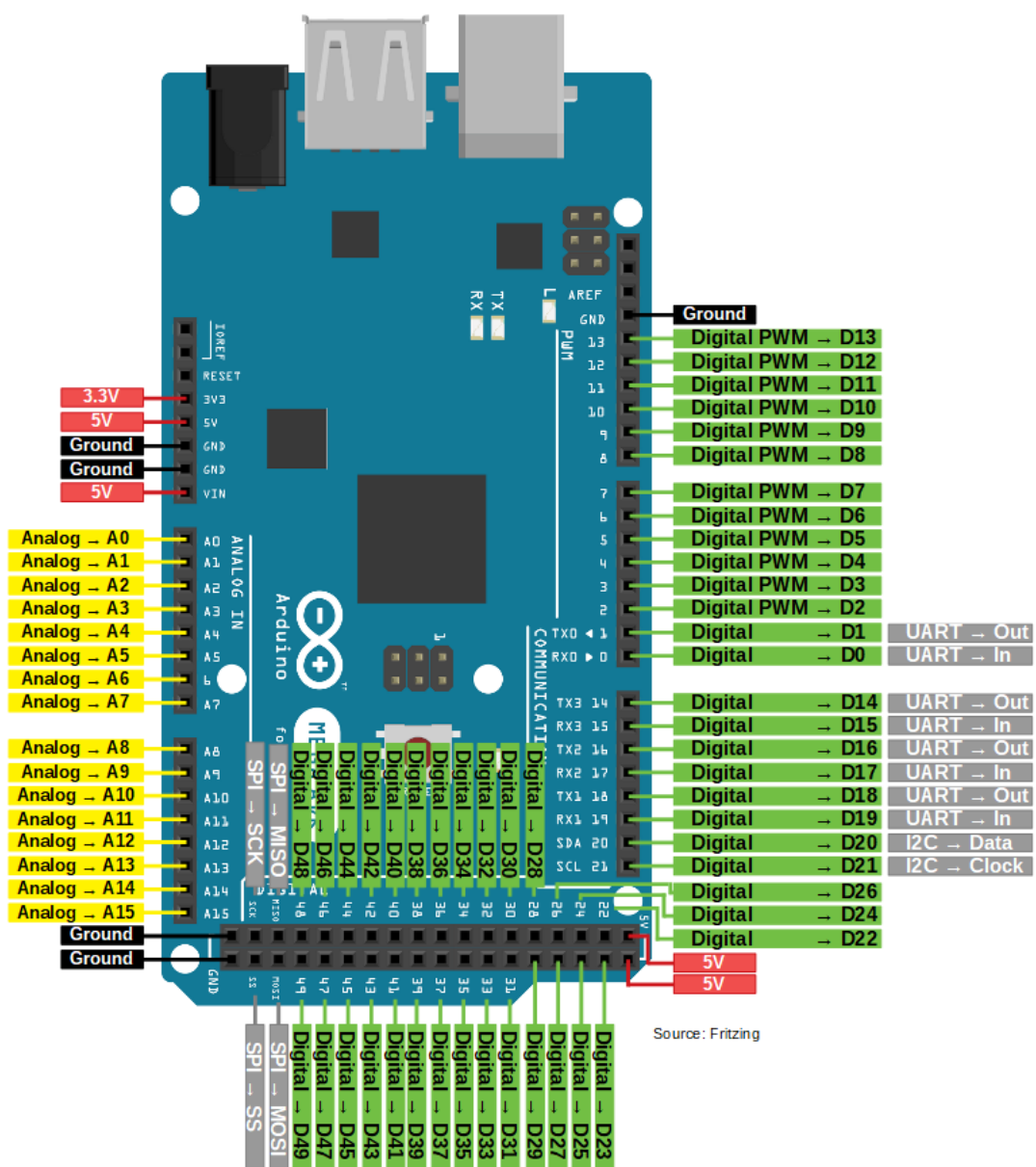
### 3.1.3. Arduino MEGA 2560 upravljačka jedinica

Arduino je hardverska i softverska platforma otvorenog tipa, ona omogućava kreatorima korištenje malih pločica koji imaju ugrađene mikrokontrolere i još neke uređaje u svrhu spajanja računala sa njihovim kreacijama, kao i samu kontrolu uređaja[6]. Različiti dizajni Arduino pločica koriste razne mikroprocesore i kontrolere. Ploče su opremljene s digitalnim i analognim ulazno/izlaznim (I/O) pinovima koji se mogu povezati s raznim proširujućim pločama ("štitnicima") ili pločama (za prototipiranje) i drugim krugovima. Ploče sadrže serijska komunikacijska sučelja, uključujući Universal Serial Bus (USB) na nekim modelima, koja se također koriste za učitavanje programa s osobnih računala. Mikrokontroleri se mogu programirati pomoću programskih jezika C i C ++, koristeći standardni API koji je također poznat kao "Arduino jezik". Osim korištenja tradicionalnih alata za sastavljanje alata, projekt Arduino također pruža i integrirano razvojno okruženje (IDE) kao i alat za pisanje arduino programa. Projekt Arduino započeo je 2005. godine kao alat za studente Instituta za dizajn interakcije Ivrea u Ivrei, Italija, sa ciljem da početnicima i profesionalcima omogući jeftin i jednostavan način stvaranja uređaja koji u interakciji s okolinom koriste senzore i aktuatorne. Uobičajeni primjeri takvih uređaja namijenjenih početnicima hobistima uključuju jednostavne robote, termostate i detektore kretanja. Ime Arduino potječe od bara u Ivrei u Italiji gdje su se sastajali neki od osnivača projekta. S obzirom da je Arduino platforma otvorenog tipa, nastale su i mnoge druge verzije, jedna od njih je i Croduino koji je izradila e-radionica.



Slika 3.5: Arduino MEGA 2560.

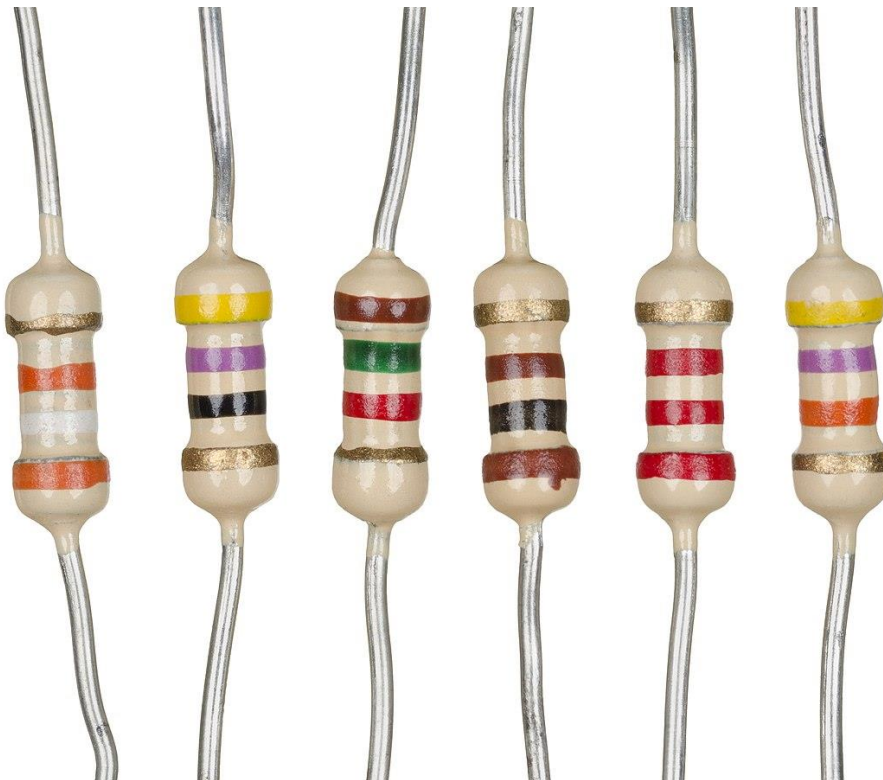
Za potrebe ovog projekta koristit ćemo Arduino MEGA 2560 Upravljač. Arduino MEGA 2560 je mikrokontroler ploča koja se temelji na ATmega2560 [7]. Ima 54 digitalna I/O pina (od kojih se 14 mogu koristiti kao PWM izlazi), 16 analognih ulaza, 4 UART (hardverski serijski portovi), kristalni oscilator od 16 MHz, USB priključak, priključak za napajanje, ICSP zaglavlje i gumb za resetiranje. Sadrži sve što je potrebno za podršku mikrokontrolera. Napajanje se može postići povezivanjem uređaja preko USB-a na računalo, ali također je moguće i povezivanje na bateriju kao i putem AC/DC adaptera na izmjeničnu mrežu. Iako bi ovaj projekt bio izvediv i putem Arduino UNO pločice, koristit ćemo Arduino MEGA kako bi ostalo dodatnog mjesta za moguća proširenja i nadogradnje sustava u budućnosti.



Slika 3.6: Raspored pinova na Arduino MEGA 2560.

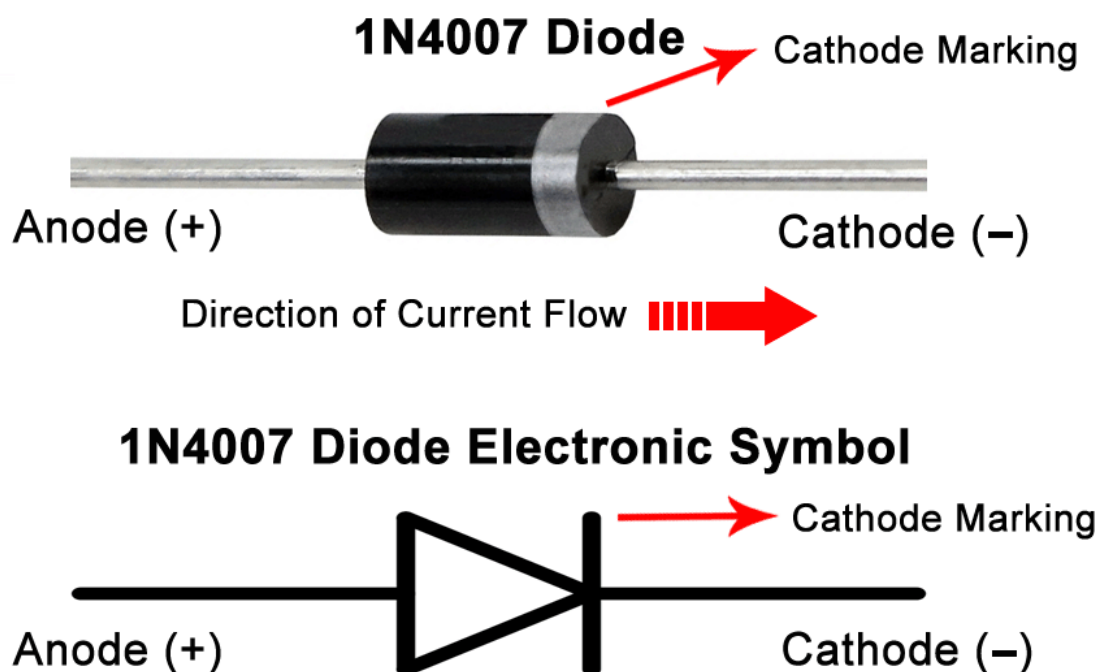
### 3.1.4. Ostale elektroničke komponente

Otpornik je pasivna električna komponenta koja implementira električni otpor kao element kruga. U elektroničkim krugovima, otpornici se koriste za smanjenje protoka struje, podešavanje razine signala, za podjelu napona i ostale upotrebe. Otpori velike snage koji mogu rasipati više Watta električne energije kao toplina, mogu se koristiti kao upravljački dijelovi u motorima, u sustavima distribucije električne energije ili kao ispitna opterećenja za generatore. Fiksni otpornici imaju otpor koji se neznatno mijenjaju samo temperaturom, vremenom ili radnim naponom. Promjenjivi otpornici mogu se koristiti za podešavanje elemenata kruga (poput regulacije glasnoće ili zatamnjenja svjetiljke) ili kao senzorski uređaji za toplinu, svjetlost, vlagu, silu ili kemijske aktivnosti. Otpornici su uobičajeni elementi električnih mreža i elektroničkih sklopova i sve su prisutni u elektroničkoj opremi. Praktični otpornici kao diskretne komponente mogu se sastojati od različitih spojeva i oblika. Otpornici se također primjenjuju unutar integriranih krugova. Električna funkcija otpornika određena je njegovim otporom. Uobičajeni komercijalni otpornici proizvode se u rasponu većem od devet reda veličine. Nominalna vrijednost otpora spada u proizvodnu toleranciju, naznačenu na komponenti. Za ovaj projekt koristit ćemo otpornike različitih vrijednosti već integrirane u strujne krugove, ali i jedan otpornik od  $4,7\text{ k}\Omega$  na bazi tranzistora koji ćemo sami postaviti.



**Slika 3.7:** *Primjer otpornika.*

Snubber se često koriste u električnim sustavima s induktivnim opterećenjem, gdje nagli prekid struje dovodi do oštrog porasta napona preko sklopnog uređaja ("induktivni udar"), u skladu s Faradayevim zakonom[8]. Taj prolazni izvor može biti izvor elektromagnetskih smetnji (EMI) u drugim krugovima. Uz to, ako napon generiran preko uređaja premašuje ono što uređaj može tolerirati, može ga oštetiti ili uništiti. Snubber pruža kratkoročni alternativni put struje oko sklopnog sklopa struje tako da se induktivni element može sigurno isprazniti. Ima puno različitih izvedbi snubbera, međutim za ovaj sklop koristit ćemo najjednostavniju izvedbu sa samo jednom diodom spojenom u paraleli sa induktivnim trošilom što je u ovom slučaju magnetni ventil. Za tu svrhu koristit ćemo najjednostavniju diodu 1N4007.



**Slika 3.8:** Dioda 1N4007.

Tranzistor je aktivni poluvodički element koji se koristi za pojačavanje elektronskih signala i električne energije u analognim sklopovima, a kod digitalnih sklopova se koristi kao sklopka[9]. Sastoji se od poluvodičkog materijala s najmanje tri terminala za spajanje na vanjski krug. Napon ili struja primijenjeni na jedan par terminala tranzistora upravlja strujom kroz drugi par terminala. Budući da kontrolirana (izlazna) snaga može biti veća od upravljačke (ulazne) snage, tranzistor može pojačati signal. Danas se neki tranzistori pakiraju pojedinačno, ali mnogo ih se više nalazi u integriranim krugovima. Tranzistori su izvršili revoluciju u području elektronike i utrošili put manjim i jeftinijim radio uređajima, kalkulatorima i računalima, između ostalog. Prvi

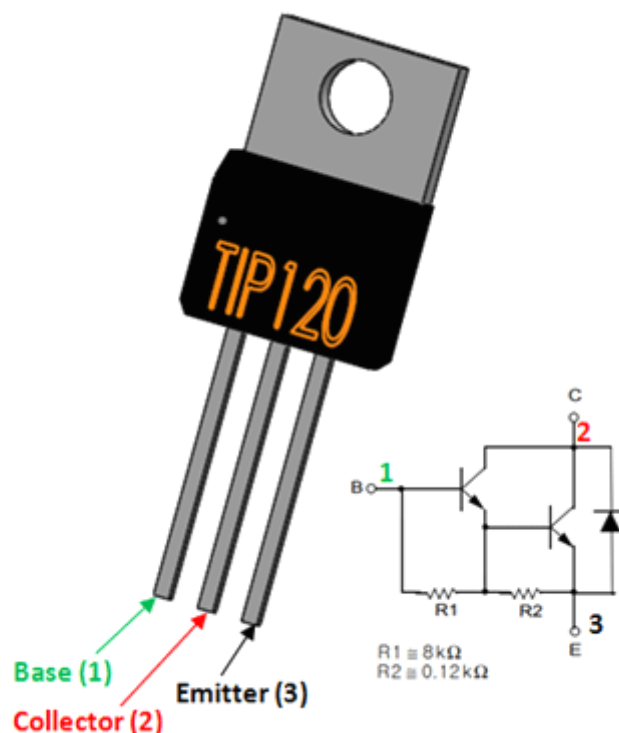


tranzistor i MOSFET nalaze se na popisu IEEE prekretnica u elektronici. Većina tranzistora izrađena je od vrlo čistog silicija, a neki od germanija, ali ponekad se koriste i neki drugi poluvodički materijali. Tranzistor može imati samo jednu vrstu nosača naboja, u terenskom tranzistoru ili može imati dvije vrste nosača naboja u uređajima s bipolarnim spojnim tranzistorima. U usporedbi s vakuumskom cijevi, tranzistori su općenito manji i za rad im je potrebno manje energije. Određene vakuumske cijevi imaju prednosti u odnosu na tranzistore pri vrlo visokim radnim frekvencijama ili visokim radnim naponima. Mnoge vrste tranzistora izrađuju se prema standardiziranim specifikacijama više proizvođača.



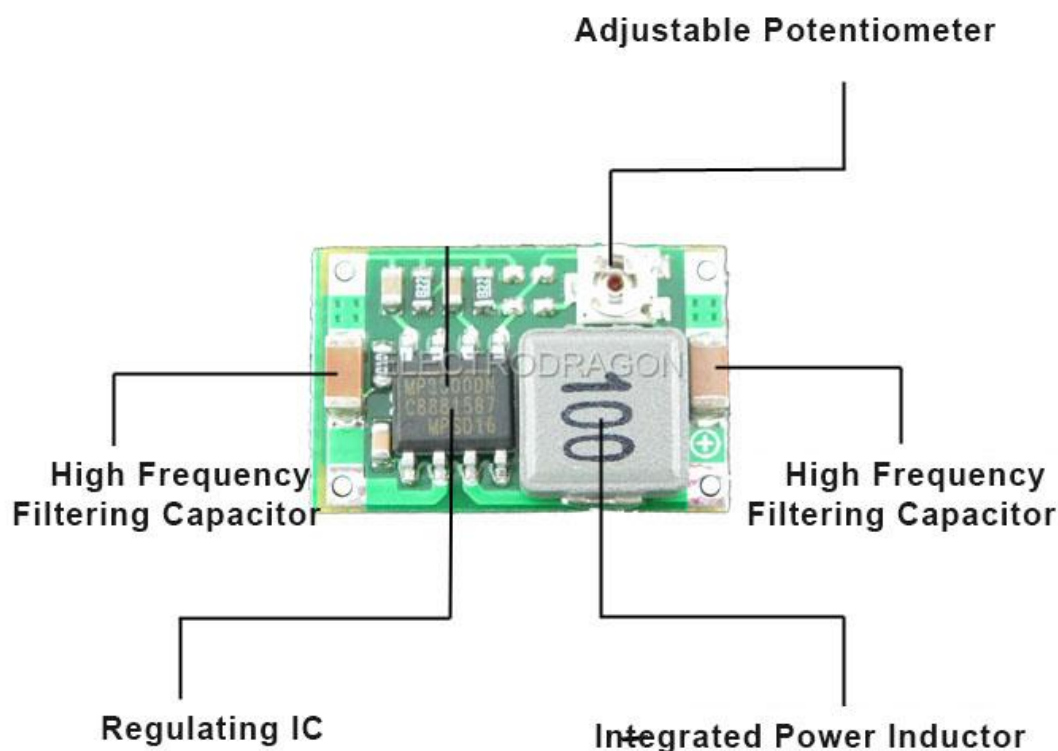
**Slika 3.9:** *TIP 120.*

U ovom projektu koristit ćemo TIP 120 model tranzistora. TIP120 je NPN energetski tranzistor u Darlingtonovu spoju[10]. Može prebaciti opterećenja do 60 V s maksimalnom strujom od 8A i kontinuiranom strujom od 5A. To ga čini pogodnim za elektroniku srednje i velike snage poput upravljačkih motora, solenoida ili LED-ova velike snage.



**Slika 3.10:** Shema TIP 120 tranzistora sa označenim izvodima[11].

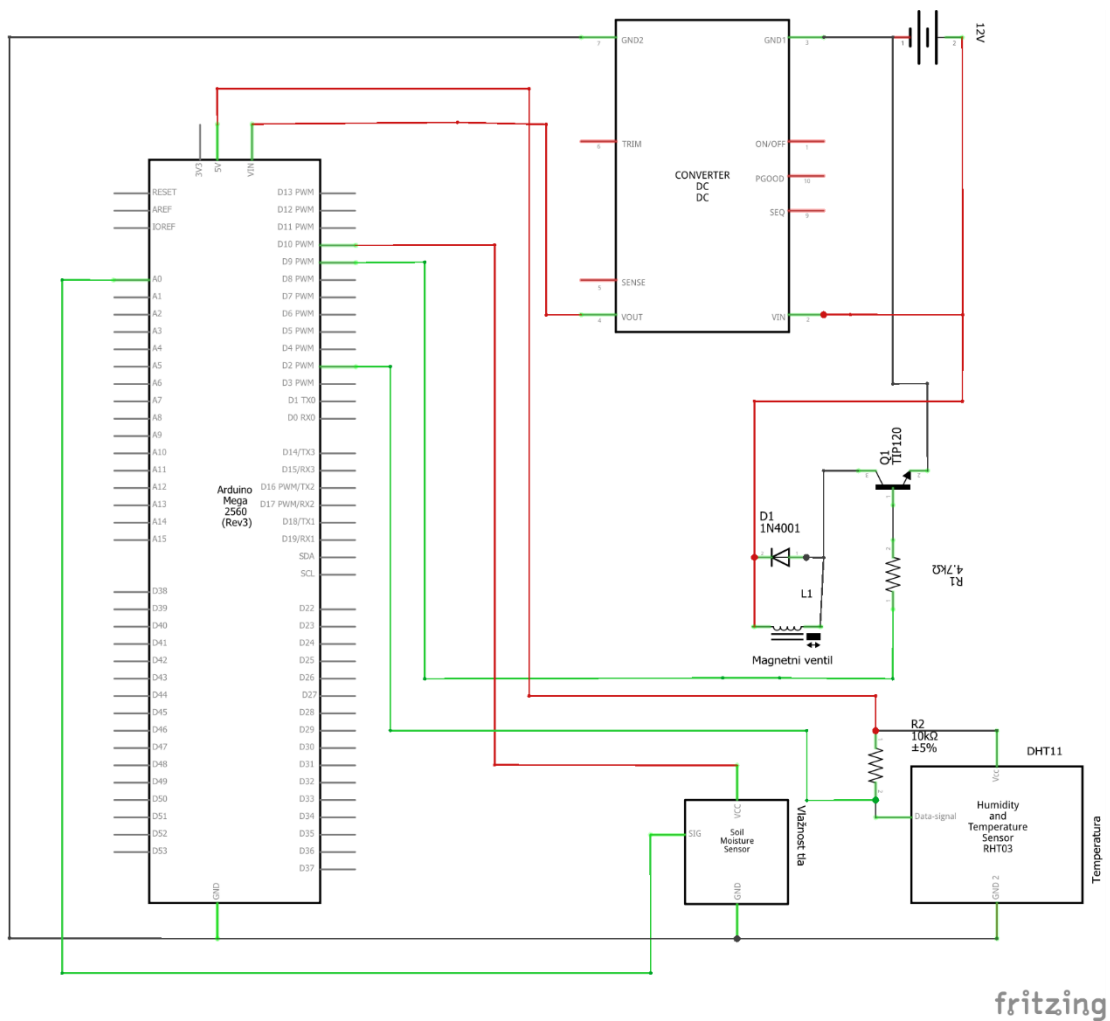
*Buck* pretvarač (*Step-down modul*) je DC/DC pretvarač koji smanjuje napon (istovremeno povećavajući struju) od svog ulaza (napajanja) do izlaza (opterećenja)[11]. To je klasa napajanja s preklapljenim načinom rada (SMPS) koji obično sadrži najmanje dva poluvodiča (dioda i tranzistor, iako moderni pretvarači često zamjenjuju diodu drugim tranzistorima koji se koriste za sinkrono ispravljanje) i barem jedan element za pohranu energije, kondenzator, induktor ili njih dvoje u kombinaciji. Kako bi se smanjilo punjenje napona, takvim se izlazima pretvarača (filtrar na strani opterećenja) obično dodaju filtri napravljeni od kondenzatora (ponekad u kombinaciji s induktorima). Preklopni pretvarači (poput pretvarača s buck-om) pružaju mnogo veću efikasnost napajanja kao pretvarači istosmjernog i istosmjernog napona u odnosu na linearne regulatore, što su jednostavniji krugovi koji snižavaju napon raspršivanjem snage kao topline, ali ne povećavaju izlaznu struju. Buck pretvarači mogu biti vrlo učinkoviti (često veći od 90%), što ih čini korisnim za zadatke kao što su pretvaranje glavnog (skupno) naponskog napajanja (često 12 V) u niži napon koji su potrebni za USB, DRAM i CPU (1,8 V ili manje).



**Slika 3.11:** MP2307 Step-Down modul sa prikazima glavnih elemenata.

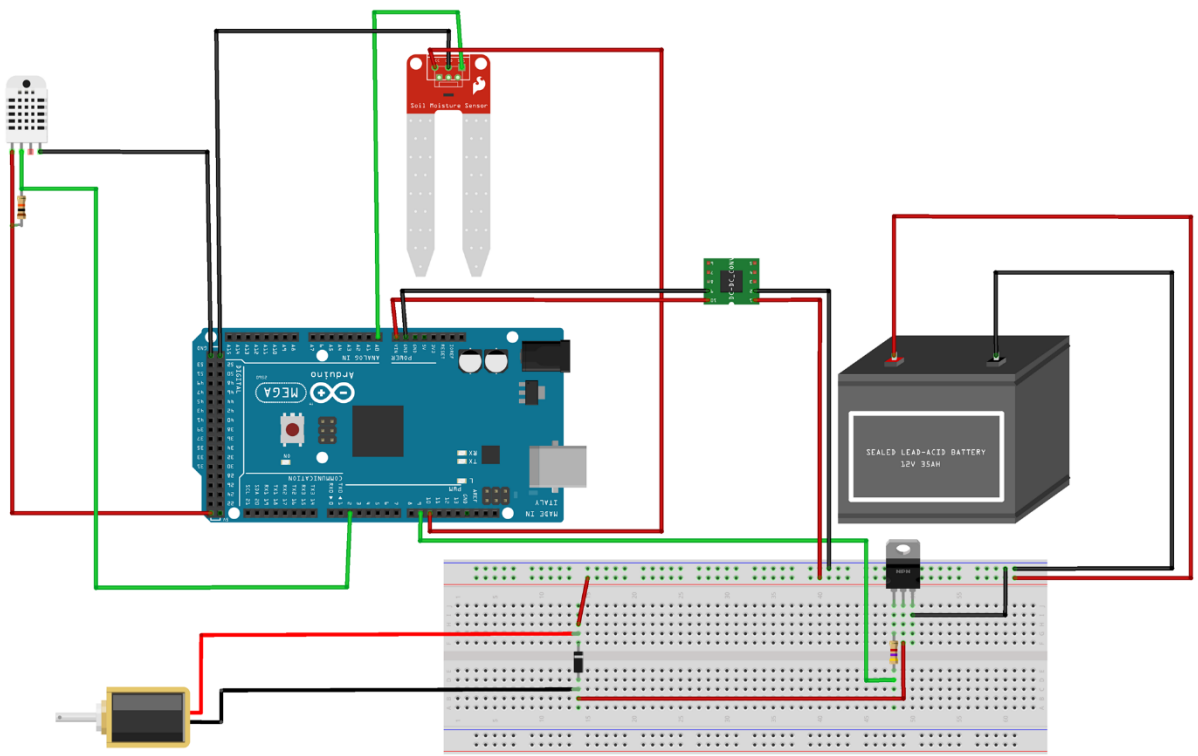
S obzirom da ćemo koristiti olovnu bateriju od 12 V što je na samoj granici prihvatljivosti rada Arduino MEGA 2560 pločice, a kako nam za sve uređaje osim magnetnog ventila treba napajanje od 5V koristit ćemo MP2307 *step-down* modul kako bismo spustili napajanje na 5V za sve uređaje osim magnetnog ventila kako bismo uštedili na energiji ali i zaštitili ploču od pregrijavanja i ostalih mogućih nepredvidivih situacija koje bi mogle oštetiti pločicu. Ovaj step-down modul ulazni napon snižava na točno određeni izlazni napon koji se podešava putem trimera na pločici. Vrlo je praktičan jer promjenom ulaznog napona, izlazni ostaje uvijek isti. Napone pretvara s visokom efikasnošću, značajno višom od standardnih regulatora napona što nam je vrlo korisno jer zbog toga manje energije prelazi u toplinu, a vrlo bitno nam je da što više spriječimo pregrijavanje uređaja kako bismo ih mogli postaviti na otvoreno tokom ljetnih mjeseci kada će i imati najveću primjenu.

### 3.2. Konačna shema sklopa



Slika 3.12: Konačna shema sklopa.

Konačna shema sklopa je izrađena u alatu Fritzing. To je programski alat koji sadrži mnoštvo komponenti i omogućava ubacivanje dodatnih komponenti kreiranih od korisnika koji pojednostavljaju izradu sheme. Osim što se može izraditi shema, također se može izraditi i ilustracija temeljena na shemi kao i nacrt PCB sklopa za daljnju izradu.



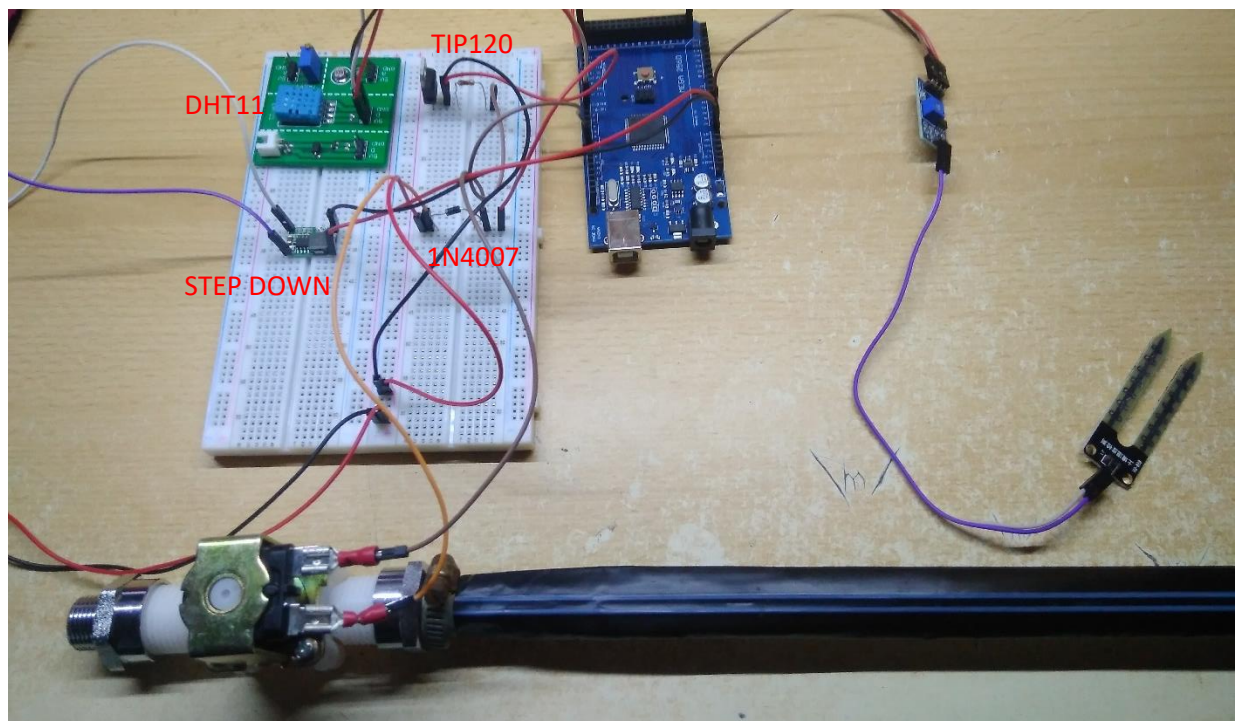
fritzing

Slika 3.13: *Ilustracija konačnog sklopa.*

## 4. KONAČNA REALIZACIJA

### 4.1. Početno spajanje komponenti

Za spajanje komponenti ćemo koristiti *breadboard* ploču. Breadboard je konstrukcijska baza za prototipiranje elektronike[12]. Riječ se izvorno odnosila na doslovnu ploču za kruh, polirani komad drva koji se koristio za rezanje kruha. 1970-ih je postala dostupna ploča bez lemljenja, a danas se za njih koristi termin "*breadboard*". Budući da ploča bez lemljenja ne zahtijeva lemljenje, može se ponovo upotrijebiti. To je čini jednostavnom za upotrebu privremenih prototipa i eksperimentiranje s dizajnom krugova. Zbog toga su ploče bez lemljenja popularne i kod učenika i u tehnološkom obrazovanju.



Slika 4.1: Početni spoj komponenti.

## 4.2. Kodiranje sklopa

Prije nego što počnemo pisati kod za funkcioniranje sustava potrebno je prije svega pripremiti sve varijable preko kojih ćemo pokretati i upravljati senzorima i magnetnim ventilom

```
//Početno postavljanje pinova i dodavanje potrebnih librarya
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
int sensor_pin = A0;
int soil_moisture ;
#define DHTPIN 2
#define DHTTYPE DHT11
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
int solenoidPin = 9;
int SoilVCCpin = 10;
```

**Slika 4.2:** Postavljanje početnih varijabli.

Nakon što smo postavili početne varijable u *void setup()* postavljamo određene pinove uz *pinMode()* u stanje *OUTPUT* kako bismo naznačili da ćemo kroz njih puštati napon, također pokrećemo i *Serial Monitor* uz *Serial.begin()* naredbu kako bismo na računalu mogli pratiti tijek programa i rezultate koje ispisuje. Osim gore navedenog uzimamo podatke o DHT senzoru iz *librarya* kako bismo na dobar način uzimali podatke o temperaturi i vlazi zraka.

```
void setup() {

    Serial.begin(9600);
    pinMode(SoilVCCpin, OUTPUT);
    dht.begin();
    pinMode(solenoidPin, OUTPUT);

    //Dohvaćanje podataka o senzoru iz librarya
    sensor_t sensor;
    dht.temperature().getSensor(&sensor);
    dht.humidity().getSensor(&sensor);

}
```

**Slika 4.3:** Postavljanje *OUTPUT* pinova i dohvaćanje podataka o senzoru.

Na početku u *void loop()* ćemo dohvatiti vrijednosti temperature i vlage zraka uz pomoć naredbi koje nam omogućuje DHT library, te ćemo ih ispisati na *serial monitor* uz pomoć *Serial.print()* naredbe

```
void loop() {

    //Dohvaćanje temperature sa senzora i ispisivanje na ekran
    sensors_event_t event;
    dht.temperature().getEvent(&event);
    Serial.print(F("Temperatura: "));
    Serial.print(event.temperature);
    Serial.println(F("°C"));
    float temperature = event.temperature;

    //Dohvaćanje vlage zraka sa senzora i ispisivanje na ekran
    dht.humidity().getEvent(&event);
    Serial.print(F("Vlaga zraka: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
}
```

**Slika 4.4:** Dohvaćanje vrijednosti temperature i vlage zraka.

Nakon što smo uzeli vrijednosti temperature i vlage zraka možemo ići na glavni dio zadatka, kako bismo produžili rad senzora što je više moguće aktivirat ćemo senzor samo onda kada želimo pokrenuti navodnjavanje ukoliko je to potrebno. Kako bi se postigle što veće uštede zalijevat će se samo u periodima kada nije jako sunce, to ćemo postići tako da se omogući ispitivanje vlage tla samo kada temperatura padne ispod neke vrijednosti, u svrhu testiranja ovog programa za početak ćemo tu vrijednost postaviti na 30°C. Kada temperatura padne ispod zadane vrijednosti pustit ćemo struju kroz pin koji smo odredili kao naponski za senzor te uzeti očitavanje vlage tla. Testiranjem senzora otkrili smo da je vrijednost suhog tla oko 550, a natopljenog tla blizu 0, koristeći funkciju *map()* pretvaramo te vrijednosti u postotke. Funkcija *map()* pretvara jedan raspon brojeva u drugi i to po principu *map(varijabla, niska vrijednost početne varijable, visoka vrijednost početne varijable, krajnja niska vrijednost, krajnja visoka vrijednost)*, stoga ako napišemo *map(soil\_moisture, 550, 0, 0, 100)* možemo lako dobiti postotak vlažnosti tla. Nakon što smo dobili vrijednosti i ispisali je na serial monitoru, gasimo dovod napon do senzora koristeći *digitalWrite(SoilVCCpin, LOW)*.



```
if (temperature < 30.00){
  digitalWrite(SoilVCCpin, HIGH);
  soil_moisture= analogRead(sensor_pin);
  soil_moisture = map(soil_moisture,550,0,0,100);
  Serial.print("Vlaga tla : ");
  Serial.print(soil_moisture);
  Serial.println("%");
  digitalWrite(SoilVCCpin, LOW);
}
```

**Slika 4.5:** Dohvaćanje vlažnosti tla i zapisivanje u obliku postotka.

Nakon što smo dobili vrijednost vage u tlu možemo kreirati kod koji pokreće navodnjavanje ako je vlaga ispod granice potrebne za normalno funkcioniranje biljke, zapotrebe testiranja koda i sklopa ovu ćemo vrijednost postaviti na samu donju granicu, u stvarnosti ta bi se vrijednost postavila već negdje na sredinu samog raspona koji je potreban za normalno funkcioniranje biljke, dok bi se donja granica koristila isključivo kao okidač koji bi pokretao navodnjavanje čak i ako je i dalje vrlo visoka temperatura, odnosno jako sunce kako bi se spriječilo padanje biljke u šok i otežavanje razvoja ploda. Kao gornju vrijednost koja ćemo postaviti 50% vlažnosti tla kako bismo uspješno mogli testirati hoće li se zaustaviti, u stvarnosti ta vrijednost bi isto tako bila veća, na otprilike 70%, a u nekim situacijama i viša ovisno o biljci i u kojoj se fazi rasta nalazi. Kako bismo osigurali da se kod izvodi sve dok ne postignemo željenu vrijednost koristit ćemo while petlju. Također zalijevanje ćemo vršiti u intervalima, nakon kojih ćemo provjeravati jesmo li dostigli željenu vrijednost za prekidanje while petlje. Za potrebe ispitivanja ovdje će intervali biti dugački 10 sekundi, u stvarnosti bi to bili duži intervali, također primjenit ćemo odgodu nakon što završimo navodnjavanje od jedne sekunde prije nego uzmemo vrijednost vlage u tlu.

```

if (soil_moisture < 40){
  while(soil_moisture<50){
    digitalWrite(solenoidPin, HIGH);
    Serial.println("otvoreno");
    delay(10000);
    digitalWrite(solenoidPin, LOW);
    Serial.println("zatvoreno");
    delay(1000);
    digitalWrite(SoilVCCpin, HIGH);
    soil_moisture= analogRead(sensor_pin);
    soil_moisture = map(soil_moisture,550,0,0,100);
    Serial.print("Vlaga tla : ");
    Serial.print(soil_moisture);
    Serial.println("%");
    digitalWrite(SoilVCCpin, LOW);
  }
}

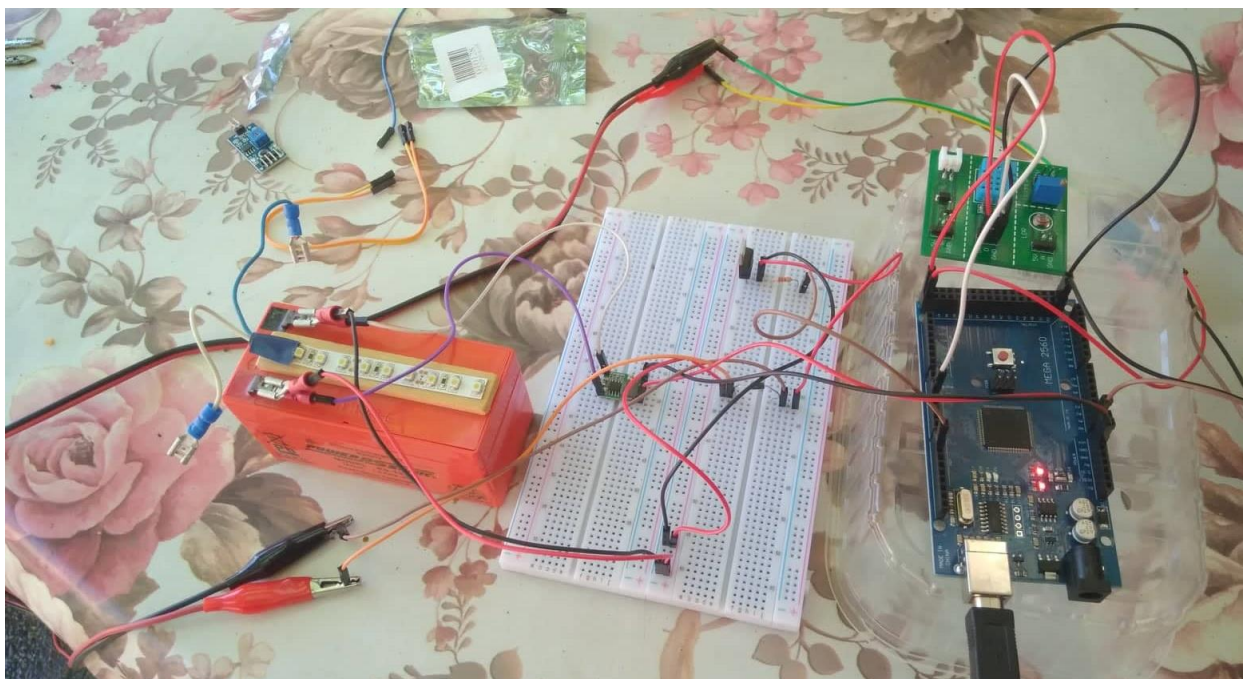
```

**Slika 4.6:** Navodnjavanje i uvijeti potrebni za njegov prekid.

Na kraju void loop funkcije dodajemo *delay()* u kojem upisujemo koliko često želimo da se ovaj cijeli kod ponavlja, to vrijeme upisujemo u milisekundama, za potrebe testiranja tu ćemo vrijednost postaviti na 10 000 milisekundi odnosno 10 sekundi, u stvarnosti ta vrijednost bi se postavila na neki duži interval, primjerice jedan sat.

### 4.3. Konačno testiranje

Konačno testiranje ćemo provesti samo na jednoj biljci umjesto na redu biljaka zbog jednsotavnosti izvedbe, u stvarnosti ovaj sustav se može skalirati na veće redove biljki. U stvarnosti sklop bi funkcionirao neovisno od računala jednom kad prenesemo kod na njega, međutim za potrebe testiranja i ispisivanje rezultata, sklop ćemo držati spojenim sa računalom kako bismo vidjeli kako funkcionira i donijeli neka unaprijeđenja, korigirali greške i slično.



**Slika 4.7:** *Konačni sklop spojen na napajanje i računalo za prikaz stanja.*

Magnetni ventil ćemo spojiti na obično vrtno crijevo što će nam stvarati dovod vode, dok će nam odvod sa ventila biti spojen na crijevo za kap po kap navodnjavanje.



**Slika 4.8:** *Magnetni ventil spojen na vrtno crijevo i crijevo za kap po kap navodnjavanje.*

Biljka na kojoj ćemo testirati sustav nalazi se u kaliću, kako bismo ograničili djelovanje vanjskih utjecaja na naše rezultate, također koristit ćemo i klasični vlagomjer koji će nam služiti za kontrolu da se uvjerimo da dobijamo točne vrijednosti sa našim sklopom.



**Slika 4.9:** Biljka nad kojom se vrši testiranje nakon uspješnog zaustavljanja navodnjavanja na zahtijevanoj vrijednosti.

#### **4.4. Rezultati**

Testiranje je pokazalo da sklop radi, pokreće se kad treba i nema problema prilikom zaustavljanja. Pokazalo se da postoje određeni problemi sa kratkim intervalima zalijevanja, trebalo ih je dosta čak i za malo povećanje od 10%, stoga bi intervale sigurno trebalo povećati, osim toga čekanje od jedne sekunde je prekratko da prikaže stvarnu vrijednost natopljenosti tla jer je to prekratko vrijeme potrebno da tlo upije vodu. Testiranje sa strane je pokazalo da je potrebno oko pet sekundi da se upije 1% vlage, stoga je potrebno produljiti čekanje nakon prestanka zalijevanja kako bi se dobila stvarna vrijednost vlage u tlu. Korigiranjem tih vrijednosti i dodavanjem dodatnog okidača koji pokreće zalijevanje kada vrijednost vlage u tlu padne ispod 40% dobijamo konačni kod koji se nalazi u prilougu P4.1

## 5. ZAKLJUČAK

Poljoprivreda je osnovna potreba današnjeg modernog društva, kako bi ona bila što učinkovitija postoji svakodnevna potreba za navodnjavanjem. Navodnjavanje se može dodatno unaprijediti automatizacijom koja u točno određenim uvjetima može pokrenuti sam proces, na taj način izbjegava se ljudska procijena potrebna za pokretanje procesa koja često može biti pogrešna. Izbacivanjem ljudskog faktora sustav postaje puno učinkovitiji, osim toga na ovaj način sustav se može pokrenuti i u situacijama kada nema ljudi u blizini ako je to potrebno.

Korištenjem Arduino MEGA 2560 pločice uspješno se može izvesti dizajn sustava da zadovoljava se potrebe uz još mjesta za moguće buduće nadogradnje i dodatno unaprijeđenje radi povećavanja učinkovitosti. Vrlo jednostavno je i prilagoditi sustav za različite biljke, dovoljno je samo spojiti pločicu na računalo i ubaciti izmjenjeni kod. Postoje određeni problemi sa korozijom senzora za mjerenje vlage tla, pokušali smo što je manje moguće koristiti ga i na taj način smanjiti koroziju, međutim ona i dalje postoji i zahtijeva zamjenu svakih nekoliko mjeseci. To se može izbjeći nadogradnjom na bolje senzore ukoliko je potrebna češća uporaba.

Naravno postoje biljke kod kojih je potrebna vlažnost puno manja, međutim to su najčešće biljke koje su navikle na puno teže uvijete poput kaktusa i dizajniranje sustava za automatsko navodnjavanje je puno kompliciranije, također postoje i biljke koje rastu u kamenovitim tlima te je i tamo nemoguća primjena klasični senzora za mjeru vlage.

## LITERATURA:

[1]. Josipović M, Kovačević V, Rastija D, Tadić L, Šoštarić J, Plavšić H, i ostali. *Priručnik o navodnjavanju*. Josipović M, urednik. Poljoprivredni institut Osijek; 2013

Dostupno na:

[https://bib.irb.hr/datoteka/655132.2013\\_11\\_18\\_3867505\\_prirucnik\\_o\\_navodnjavanju.pdf](https://bib.irb.hr/datoteka/655132.2013_11_18_3867505_prirucnik_o_navodnjavanju.pdf)

Pristup: 30.8.2020.

[2]. Solenoid Valve, [https://en.wikipedia.org/wiki/Solenoid\\_valve](https://en.wikipedia.org/wiki/Solenoid_valve) , Pristup: 30.8.2020

[3]. 2-way valves,

<https://web.archive.org/web/20131029201311/http://www.controlandpower.com/catalog/PDFs/ASCO/ASCO%2035-1%20General%20Service%202-Way%20Valves.pdf> Pristup: 30.8.2020

[4]. DHT 11, <https://www.adafruit.com/product/386> Pristup: 30.8.2020

[5]. What is a Soil Moisture Sensor and how does it work?,

<https://www.seeedstudio.com/blog/2020/01/10/what-is-soil-moisture-sensor-and-simple-arduino-tutorial-to-get-started/> Pristup: 30.8.2020

[6]. Arduino, <https://en.wikipedia.org/wiki/Arduino> Pristup: 30.8.2020

[7]. Arduino MEGA 2560, <https://store.arduino.cc/arduino-mega-2560-rev3> Pristup: 30.8.2020

[8]. Snubber, <https://en.wikipedia.org/wiki/Snubber> Pristup: 30.8.2020

[9]. Transistor, <https://en.wikipedia.org/wiki/Transistor> Pristup: 30.8.2020

[10]. TIP120-Darlington NPN transistor, <https://components101.com/transistors/tip120-pinout-datasheet-equivalent> Pristup: 30.8.2020

[11]. Buck converter, [https://en.wikipedia.org/wiki/Buck\\_converter](https://en.wikipedia.org/wiki/Buck_converter) Pristup: 30.8.2020

[12]. Breadboard, <https://en.wikipedia.org/wiki/Breadboard> Pristup: 30.8.2020

## SAŽETAK

Uspješna poljoprivreda je ključna za održavanje moderne civilizacije kakvu danas poznajemo, kao i za njezin napredak. S obzirom na rast populacije i konstantno smanjenje obradivih površina zbog urbanizacije, vrlo je bitno da poljoprivreda bude što učinkovitija. Kako bi poljoprivreda bila što učinkovitija i davala što više plodova potrebno je obratiti pozornost na količinu vode koju koristimo. Vremenske promijene uzrokovane globalnim zatopljenjem sa sve većim sušnim razdobljima, kao i konstantno smanjenje pitke vode koja je potrebna i za navodnjavanje biljaka nas tjeraju da što učinkovitije koristimo vodu. Kako bi se to postiglo razvijene su razne metode zalijevanja, od kojih je najučinkovitije odnosno najštedljivije direktno dovođenje vode do korijena poljoprivredne kulture, primjerice kap po kap. Cilj automatizirano sustava za uzgajanje biljaka je što učinkovitije dovesti i iskoristiti vodu do same biljke, bez prevelikog utjecaja čovjeka jer to unosi ljudski faktor koji često može pogriješiti. Ovaj sustav prikuplja podatke o temperaturi i vlažnosti tla te odlučuje kada je najbolje za zalijevati s obzirom na sve poznate uvijete koji ovise o samoj biljci koju se zalijeva. Sustav je kreiran tako i da štedi električnu energiju uz mjesta za daljnja proširenja i unaprijeđenja.

**Ključne riječi:** Arduino, automatizacija, navodnjavanje, senzor vlage.



## SUMMARY

Successful agriculture is key to maintaining the modern civilization as we know it today, as well as to its progress. Given the population growth and the constant reduction of arable land due to urbanization, it is very important that agriculture is as efficient as possible. In order for agriculture to be as efficient as possible and to produce as much yield as possible, it is necessary to pay attention to the amount of water we use. Weather changes caused by global warming with increasing droughts, as well as the constant reduction of drinking water needed for plant irrigation are forcing us to use water as efficiently as possible. In order to achieve this, various irrigation methods have been developed, of which the most efficient or the most economical is the direct supply of water to the roots of agricultural crops, for example drop by drop. The goal of an automated plant growing system is to bring and use water to the plant itself as efficiently as possible, without too much human influence as this is introduced by a human factor that can often go wrong. This system collects data on soil temperature and humidity and decides when it is best to water given all known conditions that depend on the plant being watered. The system is designed to save electricity in addition to space for further expansion and upgrades.

**Keywords:** Arduino, automation, irrigation, humidity sensor.

## PRILOZI

### P4.1 Konačan kod za cijeli sustav.

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
int sensor_pin = A0;
int soil_moisture ;
#define DHTPIN 2
#define DHTTYPE  DHT11
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
int solenoidPin = 9;
int SoilVCCpin = 10;

void setup() {
  Serial.begin(9600);
  pinMode(SoilVCCpin, OUTPUT);
  dht.begin();
  pinMode(solenoidPin, OUTPUT);
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  dht.humidity().getSensor(&sensor);
}

void loop() {
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  Serial.print(F("Temperatura: "));
  Serial.print(event.temperature);
  Serial.println(F("°C"));
```

```

float temperature = event.temperature;
dht.humidity().getEvent(&event);
Serial.print(F("Vlaga zraka: "));
Serial.print(event.relative_humidity);
Serial.println(F("%"));
digitalWrite(SoilVCCpin, HIGH);
soil_moisture= analogRead(sensor_pin);
soil_moisture = map(soil_moisture,550,0,0,100);
Serial.print("Vlaga tla : ");
Serial.print(soil_moisture);
Serial.println("%");
digitalWrite(SoilVCCpin, LOW);
if (soil_moisture < 40){
while(soil_moisture<70){
    digitalWrite(solenoidPin, HIGH);
    Serial.println("otvoreno");
    delay(900000);

    digitalWrite(solenoidPin, LOW);
    Serial.println("zatvoreno");
    delay(300000);
    digitalWrite(SoilVCCpin, HIGH);
    soil_moisture= analogRead(sensor_pin);
    soil_moisture = map(soil_moisture,550,0,0,100);
    Serial.print("Vlaga tla : ");
    Serial.print(soil_moisture);
    Serial.println("%");
    digitalWrite(SoilVCCpin, LOW);
}
}else if (temperature < 30.00){

```

```

digitalWrite(SoilVCCpin, HIGH);
soil_moisture= analogRead(sensor_pin);
soil_moisture = map(soil_moisture,550,0,0,100);
Serial.print("Vlaga tla : ");
Serial.print(soil_moisture);
Serial.println("%");
digitalWrite(SoilVCCpin, LOW);
if (soil_moisture < 50){
  while(soil_moisture<70){
    digitalWrite(solenoidPin, HIGH);
    Serial.println("otvoreno");
    delay(900000);

    digitalWrite(solenoidPin, LOW);
    Serial.println("zatvoreno");
    delay(300000);
    digitalWrite(SoilVCCpin, HIGH);
    soil_moisture= analogRead(sensor_pin);
    soil_moisture = map(soil_moisture,550,0,0,100);
    Serial.print("Vlaga tla : ");
    Serial.print(soil_moisture);
    Serial.println("%");
    digitalWrite(SoilVCCpin, LOW);

  }
}
}
delay(3600000);
}

```