



Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Jelena Gavran

Hana Novak

Petar Sušac

## **Razvoj IoT sustava za automatsko zalijevanje biljaka**

***„eKantica“***

Zagreb, 2023.

*Ovaj rad izrađen je na Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva pod vodstvom troje studenata – Jelene Gavran, Hane Novak i Petra Sušca, a pod mentorstvom prof. dr. sc. Hrvoja Džape i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2022./2023.*

# Sadržaj

1. Uvod.....	4
2. Ideja i plan projekta .....	6
3. Odabir sklopovskih komponenti sustava.....	10
3.1. Senzori za mjerenje razine vode i vlažnosti tla .....	10
3.2. Pumpa.....	13
3.3. Senzor za mjerenje razine osunčanosti.....	14
3.4. Senzor za mjerenje temperature i vlage zraka.....	15
3.5. Mikrokontroler.....	16
4. Implementacija komponenti i razvoj sustava .....	19
4.1. Mobilna aplikacija .....	19
4.2. Aplikacijsko programsko sučelje.....	26
4.3. Mreža.....	26
4.4. <i>Sleep</i> .....	27
4.5. Analogno – digitalna konverzija.....	29
4.6. VMA303 senzori .....	30
4.7. HC- SR04 senzor .....	31
4.8. DHT11 senzor .....	33
4.9. I2C protokol.....	36
4.10. SI1145 senzor .....	37
4.11. Pumpa.....	38
5. Rezultat .....	39
6. Održivost, komercijalizacija i budući razvoj .....	44
7. Zaključak .....	45
8. Zahvala.....	47
9. Popis literature .....	48
10. Sažetak i ključne riječi .....	52
11. Summary and key words.....	54

## 1. Uvod

Svijet se sve više okreće primjeni novih tehnologija u svakodnevnom životu te živimo u vremenu kada nas tehnologija okružuje više nego ikad. Internet stvari (engl. *Internet of Things* – *IoT*) koncept je koji omogućuje povezivanje mreže fizičkih uređaja i sustava u koje su ugrađeni senzori, programska potpora i druge tehnologije u svrhu povezivanja i razmjene podataka s drugim uređajima putem interneta što rezultira stvaranjem pametnijih i učinkovitijih rješenja za različita područja. Ovi uređaji variraju od običnih kućanskih predmeta do sofisticiranih industrijskih alata [1]. Neka od područja koja su vidjela značajnu primjenu IoT-a su kućanstva i agrikultura. Jedan od primjera primjene IoT-a u tim područjima automatski je sustav zalijevanja biljaka.

Tradicionalni sustavi za zalijevanje često se temelje na vremenskom rasporedu i oslanjaju se na ljudsko praćenje stanja biljaka i tla, što može dovesti do prekomjernog ili nedovoljnog zalijevanja biljaka. To može imati negativan utjecaj na rast i razvoj biljaka, kao i na potrošnju vode i drugih resursa. S druge strane, IoT automatski sustav za zalijevanje biljaka omogućuje precizno zalijevanje na temelju stvarnog stanja tla i biljaka. Senzori u tlu prate vlažnost tla te druge parametre koji utječu na rast i razvoj biljaka te na osnovu tih podataka upravljaju automatskim zalijevanjem. Takvi sustavi omogućuju uštedu vode i vremena te smanjuju potrebu za ljudskom intervencijom u procesu zalijevanja. Osim toga, omogućuju i daljinski nadzor i upravljanje putem interneta, što olakšava kontrolu i održavanje sustava. S obzirom na prethodno navedeno očito je da postoje mnoge prednosti ovakvih uređaja, ali postoje i nedostatci.

U ovom radu istraženi su principi rada IoT uređaja za automatsko zalijevanje biljaka, njihove prednosti i nedostaci te se pokušalo otkloniti neke od nedostataka postojećih uređaja razvojem novog uređaja. Najprije je opisan nastanak ideje i planiranje projekta, potom odabir sklopovskih komponenti, njihova integracija u sustav te pisanje programske potpore za njih. Uz to, opisan je i razvoj mobilne aplikacije te komunikacijskih protokola među komponentama. Također su diskutirane mogućnosti primjene ovakvog sustava u poljoprivrednoj proizvodnji, mogućnosti komercijalizacije uređaja te načini na koje se uređaj može unaprijediti kako bi se postigao još bolji učinak na rast i razvoj biljaka.

## 2. Ideja i plan projekta

Cilj projekta bio je razviti automatski sustav zalijevanja biljaka. Istražena je ponuda na tržištu te se pokazalo da postoje dvije velike grupe uređaja za automatsko navodnjavanje [2][3]. Sustavi djeluju kao sustavi navodnjavanja kap na kap, na slici 1., koji se mogu uključiti i isključiti ili se nalaze u obliku cvjetnjaka na slici 2. tj. posude u koju se posadi biljka. Kratkim razmatranjem dolazi se do zaključka da dva navedena principa nisu u potpunosti zadovoljavajuća. Primjerice, sustav u obliku cvjetnjaka ograničava veličinu biljke koja može biti posađena, a sustav kap na kap lako može prekomjerno zaliti biljku, stoga je ideja ovog projekta bila da sustav bude primjenjiv na bilo koju vrstu biljke, dakle da nije ograničen veličinom biljke, da precizno i u potpunosti ispunjava potrebe biljke, ali da je prije svega autonoman tj. da korisnik mora minimalno djelovati kako bi sustav uspješno radio. Kao što je navedeno, takav sustav mora ispuniti sve potrebe biljke, stoga je bilo potrebno utvrditi što sve utječe na njen život. Važni faktori za opstanak i napredak biljke su razina vlage te temperatura zraka u biljkinom okruženju, vlažnost zemlje u kojoj se biljka nalazi i izloženost Sunčevim zrakama [4]. S obzirom da je glavni zahtjev sustava da radi sa što manje uplitanja korisnika, potrebno je u sustavu dovesti izvor vode.



Slika 1. Sustav zalijevanja kap na kap [2]



Slika 2. Sustav zalijevanja u obliku posude [3]

S navedenim potrebama na umu započeto je planiranje projekta. Plan je bio da se u sustav integrira više senzora pomoću kojih će se očitavati potrebni podaci. Također je bilo potrebno integrirati i spremnik koji će služiti kao izvor vode koji bi bilo potrebno ponekad nadopuniti.

Cijelim sustavom upravljao bi pogodan mikrokontroler, a komunikacija s korisnikom odvijala bi se preko mobilne aplikacije.

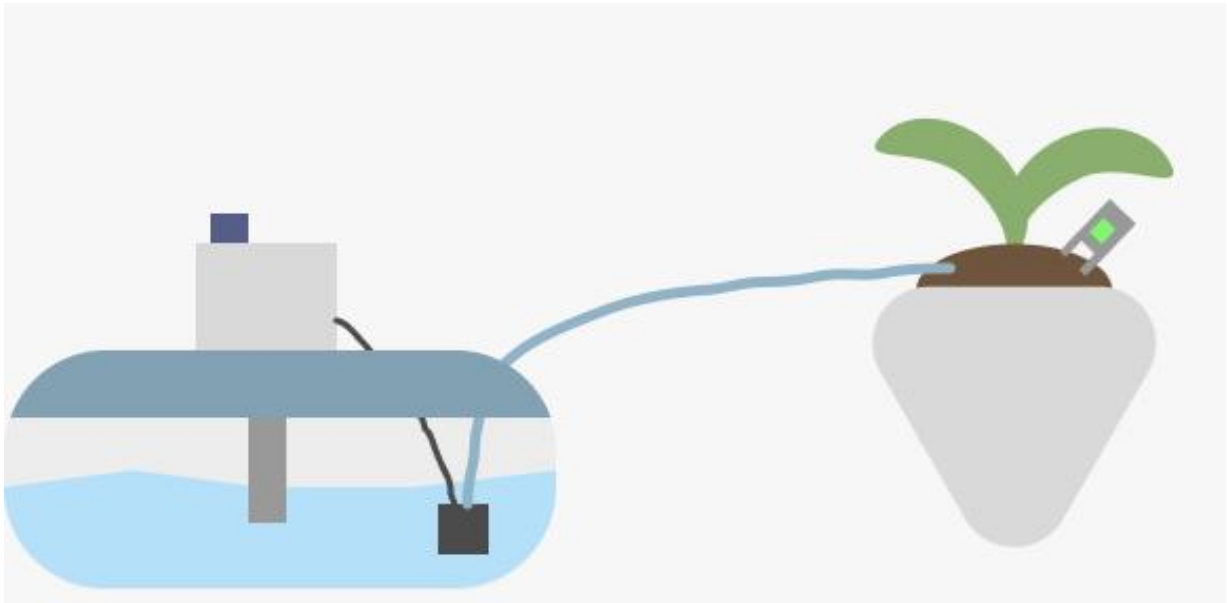
Započeta je detaljnija razrada ideje te su otpočeta istraživanja senzora, mikrokontrolera te ostalih sklopovskih i programskih tehnologija potrebnih za razvoj. Kako bi se pokrile sve potrebe biljke, odlučeno je da će se koristiti senzori za određivanje vlage zemlje, vlage i temperature zraka, razine osunčanosti te, s obzirom da je potrebno integrirati spremnik vode, senzor za određivanje razine vode u spremniku. Kako bi voda dolazila iz spremnika do biljke potrebno je u spremnik implementirati i pumpu za vodu. Početni plan bio je sve senzore povezati na jedan mikrokontroler koji bi upravljao njihovim funkcionalnostima. Podatke sa senzora koji bi se komunicirali na mikrokontroler analiziralo bi aplikacijsko programsko sučelje (engl. *Application Programming Interface - API*), dok bi mobilna aplikacija pratila zahtjeve poslane na to aplikacijsko programsko sučelje [5].

Prije početka rada korisnik bi pomoću mobilne aplikacije trebao unijeti podatke o tome kolika je optimalna vlaga zemlje, temperatura i vlaga zraka te izloženost Sunčevoj svjetlosti za biljku. Mobilna aplikacija korisniku bi davala povratne informacije o trenutnim razinama vlage, temperature i izloženosti Suncu koje izmjeri sensorima te slala upozorenja ukoliko je neki od parametara previsok ili prenizak, primjerice temperatura zraka previsoka ili izloženost Suncu preniska. Ukoliko je vlaga zemlje niska, sustav bi sam zalio biljku vodom iz spremnika koju bi pumpom i malenom cijevi doveli do biljke.

Uz već ranije navedene karakteristike sustava, bitno je i da je sustav malen tj. da ne zauzima mnogo prostora, da korisniku pruža što veći broj podataka o okruženju biljke tj. da mu daje povratne informacije o tome je li biljka u dobroj okolini te da je jednostavan za korištenje.



Nakon definiranja vrste potrebnih senzora, prešlo se na odabir specifičnih komponenti i na njihovu integraciju u prototip uređaja. Sustavu se nadjenulo ime *eKantica*, a skica budućeg sustava nalazi se na slici 3.



Slika 3. *eKantica*

### 3. Odabir sklopovskih komponenti sustava

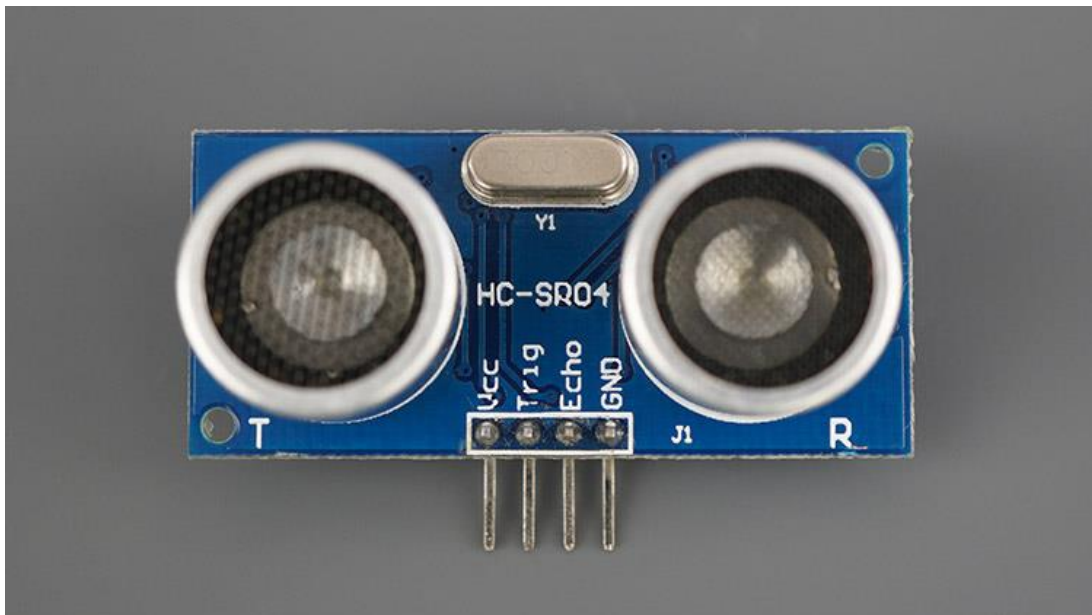
Odabir pravih komponenti za IoT sustav navodnjavanja ovisi o mnogim čimbenicima, uključujući specifičnosti okoliša, vrstu tla, potrebe biljaka, klimatske uvjete i ciljeve navodnjavanja. Senzori su jedna od najvažnijih komponenti koje se koriste u ovom sustavu jer su zaduženi za prikupljanje podataka o tlu, vlazi, temperaturi i drugim parametrima koji su bitni za navodnjavanje. Aktuatori su još jedna bitna komponenta koja se koristi u IoT sustavima navodnjavanja. Oni su zaduženi za pokretanje ventila i pumpi koje reguliraju protok vode u sustavu. Mikrokontroleri su također bitan dio ovog sustava, a oni se koriste za upravljanje aktivnostima senzora i aktuatora te za reguliranje vremena i trajanja navodnjavanja.

#### 3.1. Senzori za mjerenje razine vode i vlažnosti tla

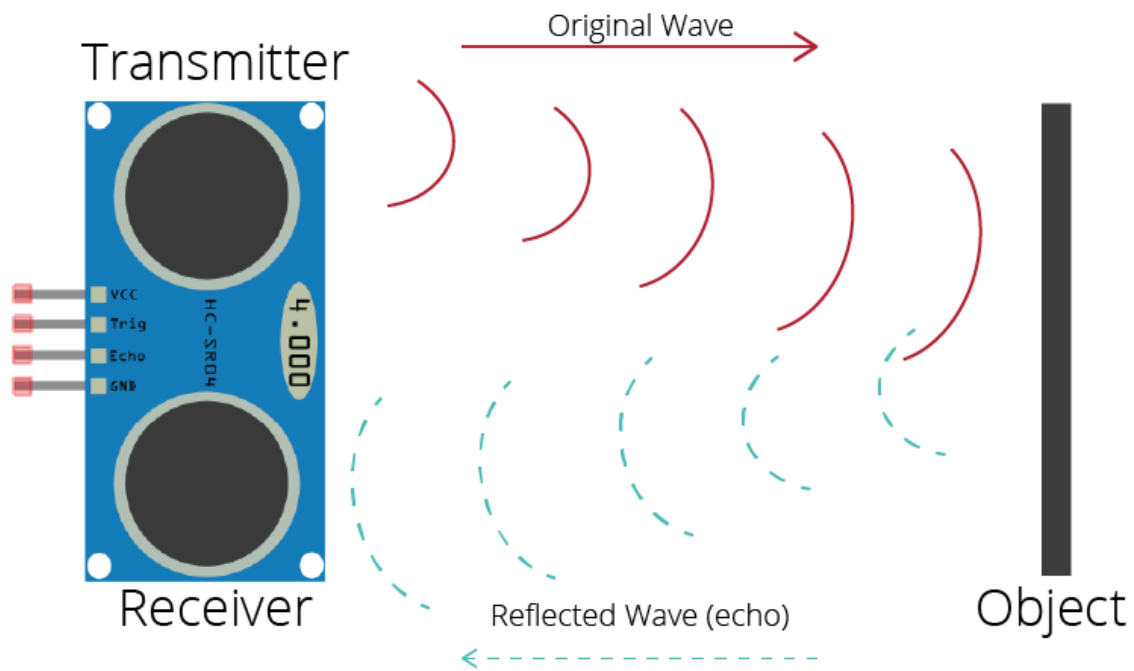
Bilo je potrebno utvrditi koji će se senzor upotrebljavati za mjerenje razine vode u spremniku. Na tržištu nema senzora koji u potpunosti odgovara potrebama ovog sustava pa su odabrana dva senzora koja pružaju različite funkcionalnosti dok se ne pojavi prikladnije rješenje u budućnosti. Radi se o senzorima HC-SR04 te WMA303.

HC-SR04 ultrazvučni je senzor udaljenosti koji se često koristi u projektima robotike i automatizacije prikazan na slici 4. Senzor koristi ultrazvučne valove za mjerenje udaljenosti objekta i omogućuje precizno mjerenje udaljenosti u rasponu od 2 cm do 400 cm. Senzor HC-SR04 ima dvije glavne komponente - odašiljač i prijemnik ultrazvuka. Odašiljač ultrazvuka šalje kratke impulse ultrazvučnih valova, a prijemnik ultrazvuka sluša odraz koji se vraća od objekta što je prikazano na slici 5. Senzor mjeri vrijeme koje je potrebno da se ultrazvuk odrazi od objekta i vrati se u senzor, a zatim izračunava udaljenost objekta koristeći brzinu zvuka. Senzor HC-SR04 ima četiri izvoda: *Vcc* (napajanje), *GND* (uzemljenje), *Trig* (odašiljač) i *Echo* (prijemnik). *Vcc* i *GND* su izvodi za napajanje senzora, dok *Trig* i *Echo* koriste za slanje i

primanje ultrazvučnih signala. Izvod *Trig* se koristi za slanje ultrazvučnog signala, dok izvod *Echo* prima odraz signala. Senzor HC-SR04 može se lako povezati s brojnim razvojnim sustavima poput Arduina, Raspberry Pi-ja, ESP32 i drugih. Napajanje senzora može biti u rasponu od 5 V do 20 V, iako se najčešće koristi napajanje od 5 V. Senzor koristi ultrazvučne signale frekvencije 40 kHz, što je izvan raspona ljudskog sluha. Senzor HC-SR04 je vrlo brz u očitavanju udaljenosti - mjerenje traje manje od 50 ms. Senzor ima kut gledanja od oko 15 stupnjeva, što znači da može precizno mjeriti udaljenost samo za objekte koji su unutar tog kuta gledanja [6]. Senzor je vrlo jeftin i dostupan pa se često koristi u raznim projektima. Važno je napomenuti da senzor HC-SR04 nije idealan za svaku situaciju. U ovom projektu korišten je za mjerenje razine vode, ali nije vodootporan na što je bilo potrebno paziti pri izradi prototipa te pisanju programske podrške. Također, senzor može imati poteškoće u mjerenju udaljenosti za vrlo malene objekte ili objekte koji su preblizu jer su ultrazvučni signali snažno reflektirani i iskrivljavaju se što znači da spremnik vode mora biti dovoljno velik kako ne bi došlo do interferencije valova te pogrešnog očitavanja.



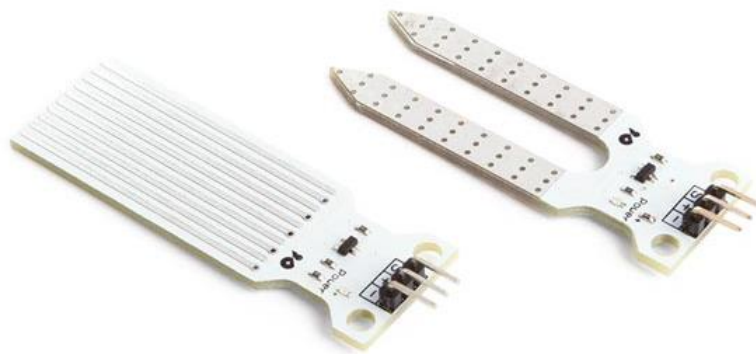
Slika 4. HC- SR04 senzor [6]



Slika 5. Princip rada HC- SR04 senzora [6]

VMA303 set je dva senzora - jednog zaduženog za praćenje razine vode, a drugog za razinu vlage tla te su prikazani na slici 6. Radi se o kapacitivnim sensorima koji se napajaju naponom od 5 V. Raspon mjerenja je od 0 do 100 posto za vlažnost tla te od 0 do 5 cm za razinu vode. Izlaz senzora analogni je napon koji svoju vrijednost povećava s dubinom uranjanja sonde u vodu ili s povećanjem vlažnosti u zemlji. Ako ploče senzora prekriva voda, analogna vrijednost bit će dostupna na izvodu *SIG*. Točnost senzora je 2% za vlagu tla te 0.5% za razinu vode. Senzori rade na temperaturnom rasponu od -10 do 60 stupnjeva celzija [7]. Senzor za vlažnost tla analogni je senzor koji se sastoji od dvije elektrode koje se ubadaju u zemlju i koje su izolirane od tla. Kada se senzor nalazi u tlu, voda u tlu djeluje kao provodnik između elektroda. Kada se napon primjeni na elektrode, struja prolazi kroz tlo, a senzor mjeri otpor između elektroda. Što je veći otpor, to je manje vlage u tlu. Na sličan princip radi i senzor za mjerenje razine vode. S obzirom da navedeni senzor za razinu vode može izmjeriti između 4 i 5 cm vode, pogodniji je za praćenje akvarija ili razine vode u posudi ljubimca nego za spremnik vode za sustav navodnjavanja koji je nešto većih dimenzija. Ipak je odlučeno implementirati ga iz

eksperimentalnih razloga jer mu je primarna svrha mjerenje razine vode u usporedbi s HC-SR04 senzorom, a trebalo ga se lako moći implementirati jer se implementirao zajedno sa VMA303 senzorom za razinu vlažnosti tla. Sensori WMA303 jednostavni su za instalaciju i održavanje, s kompaktnim dizajnom koji omogućuje fleksibilno postavljanje u tlo i vodu. Također imaju nisku potrošnju energije, što ih čini idealnim za daljinske aplikacije i uređaje na baterije.



Slika 6. VMA303 senzori za razinu vode i vlagu tla [7]

### 3.2. Pumpa

Za potrebe pumpanja vode iz spremnika korištena je mala pumpa prikazana na slici 7. Protok pumpe je 240 litara u satu, napon napajanja 12 V i nazivna struja 300 mA. Pumpa može biti izvan spremnika povezana crijevima ili može biti potopljena u vodi. Pumpu pali i gasi relej kada glavni program mikrokontrolera na temelju izmjerenih podataka izvrši naredbe za zalijevanje biljke. Releji SRD-12-VDC-SL-C također zahtjeva napajanje 12 V.

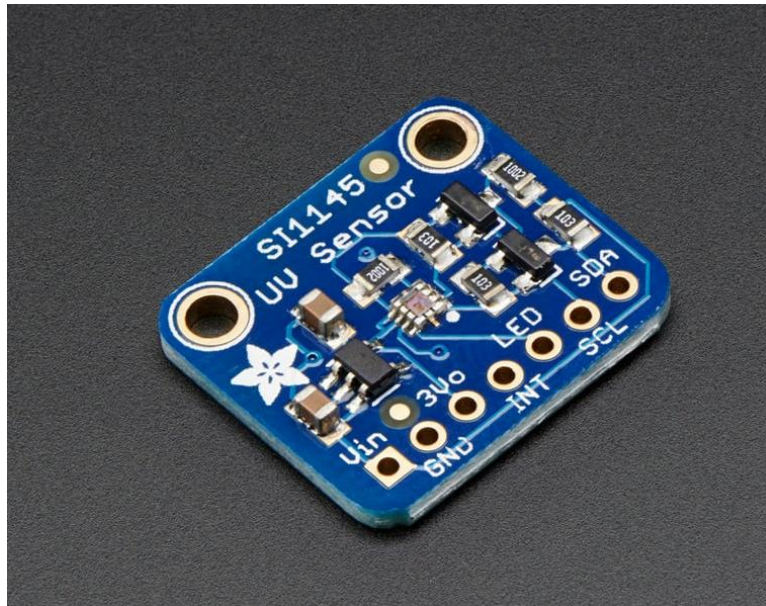


Slika 7. Pumpa za vodu

### 3.3. Senzor za mjerenje razine osunčanosti

Za dobivanje podatka o razini izloženosti Sunčevim zrakama izabran je senzor SI1145 prikazan na slici 8. To je digitalni senzor malenih dimenzija koji radi na 3.3 V, a pomoću kojeg se može dobiti precizne podatke o UV zračenju, vidljivoj i infracrvenoj svjetlosti. Koristi se za mjerenje ambijentalnog svjetla i detekciju prisutnosti objekata ili osoba. Ima ugrađen algoritam za kompenzaciju temperature, što omogućava precizna mjerenja u širokom rasponu. Senzor proizvodi tvrtka Silicon Labs, a namijenjen je za korištenje u različitim aplikacijama, uključujući pametne telefone, pametne kuće, automatsko upravljanje rasvjetom i slično zbog svoje izrazito malene potrošnje. SI1145 senzor se sastoji od dva dijela - senzora svjetla i senzora blizine. Senzor za svjetlo koristi se za mjerenje ranije navedenih vrijednosti, a senzor blizine za detekciju objekata ili osoba koji se nalaze u blizini senzora. Senzor svjetla ima visoku osjetljivost i može detektirati čak i niske razine svjetla, što ga čini idealnim za primjene u kojima je potrebno precizno mjerenje svjetla. SI1145 senzor za prijenos podataka na mikrokontroler koristi I2C (engl. *Inter-Integrated Circuit*) komunikacijski protokol, o kojemu će više riječi biti

kasnije. Osim toga, senzor ima i nekoliko programabilnih opcija, uključujući brzinu očitavanja, vremenski interval između očitavanja, različite raspon svjetlosti i još mnogo toga [8].



Slika 8. SI1145 senzor [8]

### 3.4. Senzor za mjerenje temperature i vlage zraka

DHT11 jeftin je digitalni senzor temperature i vlage zraka koji se često koristi u projektima. Senzor je razvila tvrtka Aosong, također poznata kao Adafruit Industries i osmišljen je da bude jednostavan za korištenje i pouzdan. DHT11 senzor mjeri temperaturu u rasponu od 0 do 50 stupnjeva celzija, s točnošću od 2 stupnja te relativnu vlažnost zraka u rasponu od 20% do 90% s točnošću od 5%. Senzor se napaja sa 3 do 5 V napajanja i ima jednostavan digitalni izlaz, što ga čini lako integrabilnim sa širokim spektrom mikrokontrolera. DHT11 senzor ima samo jedan izlazni izvod, koji se koristi za slanje podataka u digitalnom formatu. Podaci se šalju u vremenskom intervalu od 1 sekunde, a sastoji se od 40 bitova podataka. Prvih 16 bitova predstavlja temperaturu u binarnom formatu, sljedećih 16 bitova predstavlja relativnu vlažnost u binarnom formatu, a posljednjih 8 bitova je provjera greške [9]. S obzirom na svoju jednostavnost, DHT11 senzor je čest izbor za hobi projekte koji uključuju mjerenje temperature

i vlage zraka u zatvorenom okruženju čime je bio idealan izbor za prototip ovog uređaja, a prikazan je na slici 9.



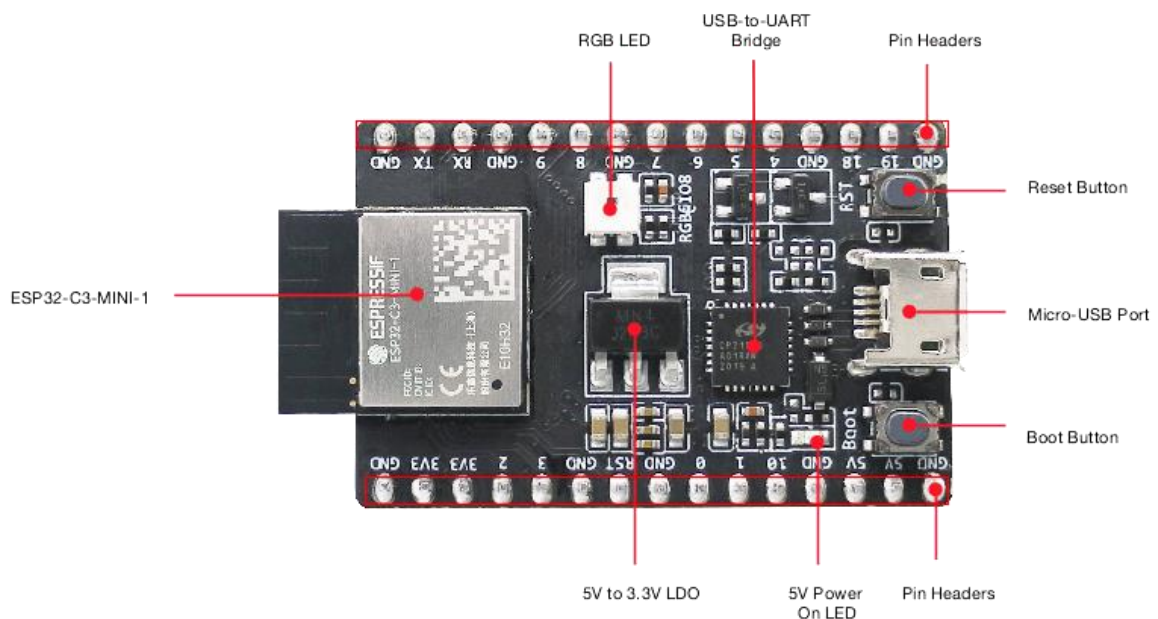
Slika 9. DHT11 senzor [17]

### 3.5. Mikrokontroler

Centralni sklopovski dio je mikrokontroler, stoga je vrlo bitno odabrati onaj karakteristika pogodnih za projekt. Za ovaj projekt bio je potreban mikrokontroler malenih dimenzija te malene potrošnje, ali dovoljno snažan da upravlja i napaja nekoliko senzora. Odabran je ESP32 tj. model ESP32-C3-DevKitM-1 prikazan na slici 10. Razvojni kit temeljen je na mikrokontroleru ESP32-C3, serije SoC (engl. *System on Chip*) platformi koje je razvila tvrtka Espressif Systems. ESP32-C3 je nova generacija ESP32 serije mikrokontrolera s jednojezgrenim 32-bitnom RISC-V CPU (engl. *Central Processing Unit* - središnja procesorska jedinica). Modul ima 400 kB SRAM (engl. *Static Random-access Memory*) memorije od kojih je 16 kB rezervirano za priručnu memoriju (engl. *cache*). Ima i 384 MB ROM memorije (engl. *Read-only memory*) te 4 MB *flash* memorije. Izabran je jer je niskobudžetan i energetski

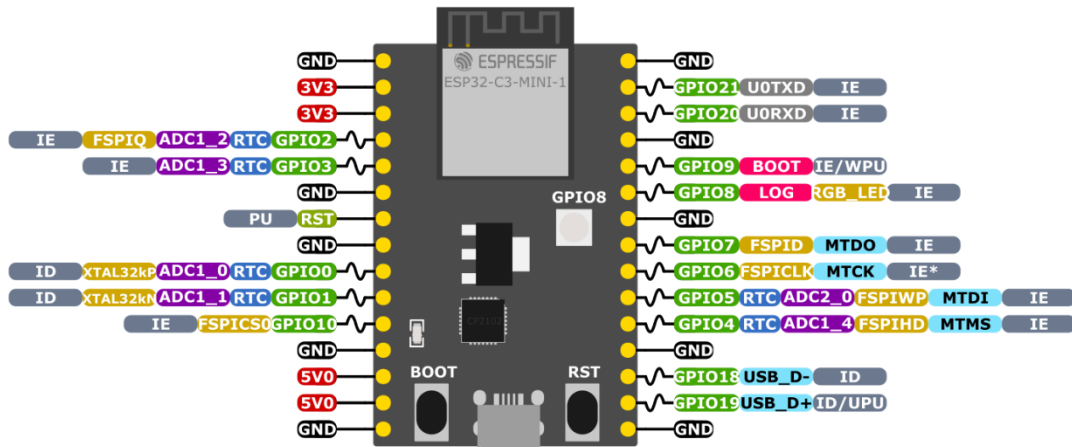


učinkovit te namijenjen za IoT aplikacije i bežične senzorske mreže. ESP32-C3-DevKitM-1 idealan je za razvoj prototipa, testiranje i različite primjene na području IoT-a, robotike i drugih područja. Ima integrirano sučelje za USB i JTAG za programiranje i komunikaciju sa računalom te digitalna sučelja UART (engl. *Universal Asynchronous Receiver/Transmitter*), SPI (engl. *Serial Peripheral Interface*), I2C i I2S (engl. *Inter-Integrated Circuit Sound*). Također ima integriran modul koji podržava WiFi i Bluetooth za bežičnu povezivost i komunikaciju, a uključuje i 22 programabilna GPIO (engl. *General Purpose Input Output*) izvoda za povezivanje senzora, aktuatora i drugih perifernih uređaja. Ispis izvoda nalazi se na slici 11. Dolazi s podrškom za FreeRTOS i MicroPython, što olakšava programiranje i razvoj aplikacija. Osim toga, dostupno je i razvojno okruženje ESP-IDF (engl. *Espressif IoT Development Framework*) za programiranje i razvoj aplikacija za ESP32-C3 [10][11].



Slika 10. Mikrokontroler ESP32-C3-DevKitM-1 [10]

# ESP32-C3-DevKitM-1



## ESP32-C3 Specs

32-bit RISC-V single-core @160MHz  
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz  
 Bluetooth LE 5  
 400 KB SRAM (16 KB for cache)  
 384 KB ROM  
 22 GPIOs, 3x SPI, 2x UART, I2C,  
 I2S, RMT, LED PWM, USB Serial/JTAG,  
 GDMA, TWAI®, 12-bit ADC

- PWM Capable Pin
- GPIO Input and Output
- JTAG for Debugging and USB
- External Flash Memory (SPI)
- Analog-to-Digital Converter
- Other Related Functions
- Serial for Debug/Programming
- Strapping Pin Functions

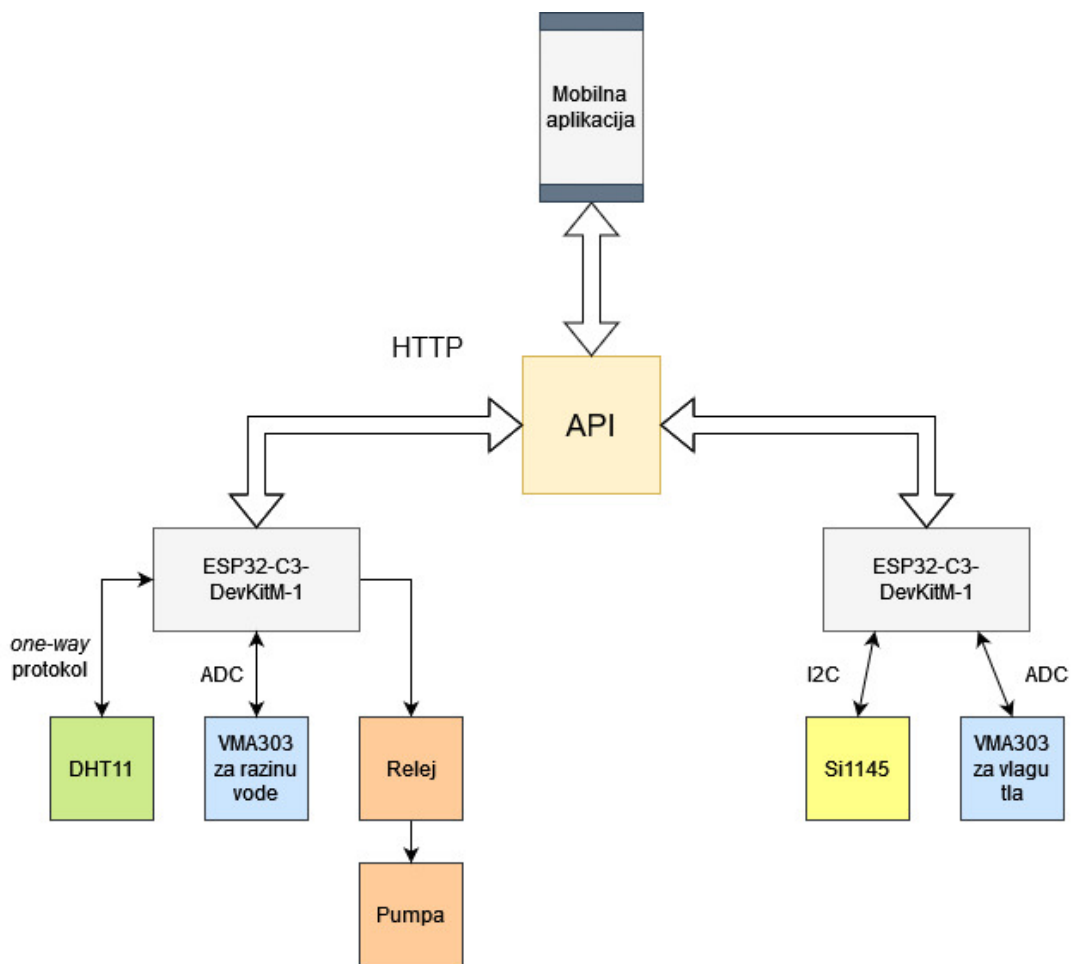
- RTC Power Domain (VDD3P3\_RTC)
- Ground
- Power Rails (3V3 and 5V)

- GPIO STATE:**
- UPU:** USB Weak Pull-up
  - WPU:** Weak Pull-up (Internal)
  - WPD:** Weak Pull-down (Internal)
  - PU:** Pull-up (External)
  - IE:** Input Enable (After Reset)
  - IE\*:** Input Enable (Depends of FUSE\_DIS\_PAD\_JTAG)
  - ID:** Input Disabled (After Reset)
  - OE:** Output Enable (After Reset)
  - OD:** Output Disabled (After Reset)

Slika 11. Popis izvoda ESP32-C3-DevKitM-1 [10]

## 4. Implementacija komponenti i razvoj sustava

Nakon odabira sklopovlja, započeta je njihova implementacija u sustav. Također je započet razvoj programske potpore za sustav - mobilne aplikacije, aplikacijskog programskog sustava te mreže koji će služiti za slanje i obradu podataka sa senzora te upravljanje sustavom. Blok shema sustava prikazana je na slici 12. Tijekom implementacije odlučeno je kako će se senzori podijeliti na dva ESP32 mikrokontrolera kako bi sustav bio jednostavniji za upotrebu.



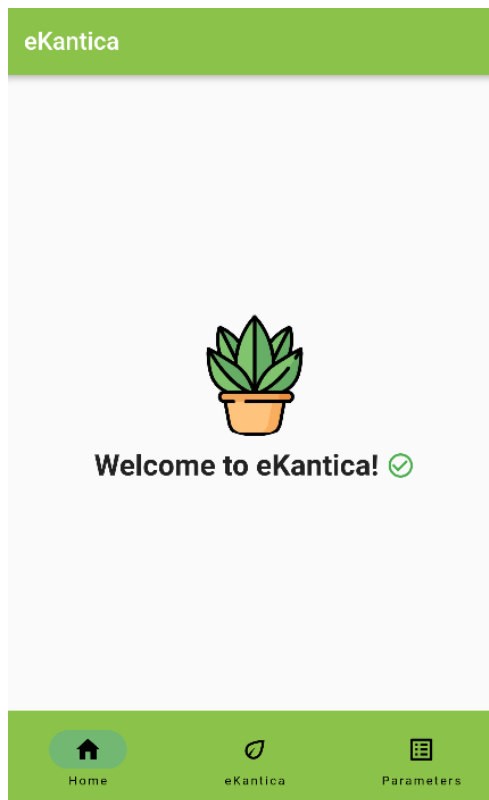
Slika 12. Blok shema *eKantice*

### 4.1. Mobilna aplikacija

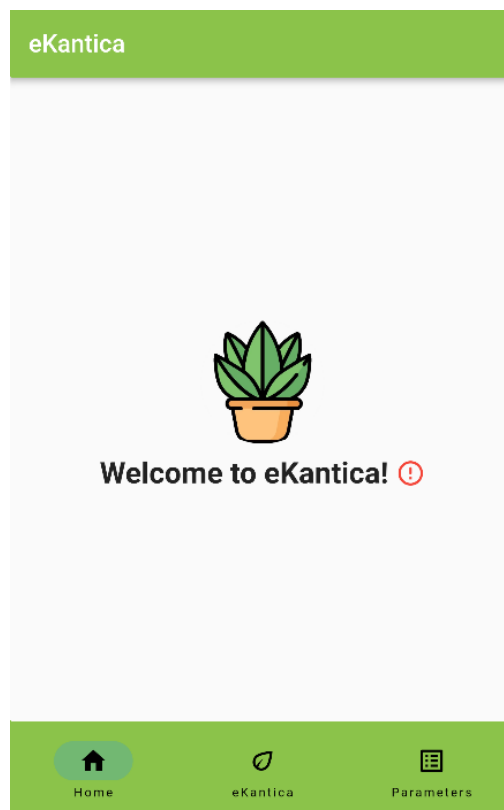
Sučelje između korisnika i sustava ostvareno je mobilnom aplikacijom. Aplikacija je izrađena pomoću razvojnog okvira Flutter, koji omogućava istovremeno kreiranje aplikacija za iOS i

Android sustave [12]. Ona korisniku omogućuje pregled trenutnog stanja biljke te promjenu dozvoljenih vrijednosti parametara.

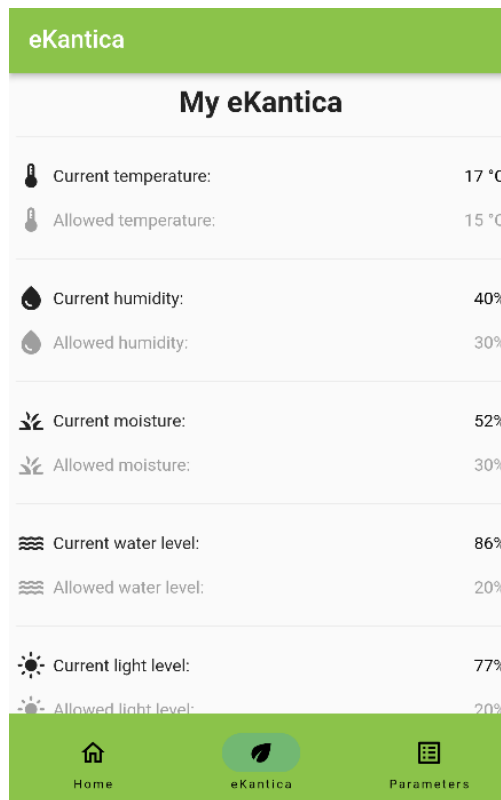
Aplikacija se sastoji od tri prikaza odnosno kartice - stranica dobrodošlice, prikaz trenutnih vrijednosti i ažuriranje dozvoljenih vrijednosti što je prikazano na slici 13. Početna stranica prikazuje logo *eKantice* te ikonu trenutnog stanja. Ako su svi uvjeti zadovoljeni, ikona je zelena, no ako nisu, ikona promijeni boju kao znak upozorenja što je prikazano na slici 14. Na stranici s trenutnim vrijednostima moguće je pregledati sve trenutne i dozvoljene vrijednosti što je prikazano na slikama 15. i 16. Aplikacija također prikladnom bojom označava parametre čije su vrijednosti ispod dozvoljene razine što je prikazano slikom 17. Ukoliko stvarne vrijednosti u okolišu biljke padnu prenisko ili se previše povise u odnosu na zadane vrijednosti, aplikacija šalje obavijest o trenutnom stanju kako je pokazano slikom 18. Posljednja stranica predviđena je za izmjenu dozvoljenih parametara i prikazana je slikom 19. Moguće je unijeti željene vrijednosti i pohraniti ih nakon čega stigne obavijest o uspješnoj promjeni vrijednosti što je prikazano slikom 20.



Slika 13. Početna stranica mobilne aplikacije



Slika 14. Početna stranica aplikacije – upozorenje

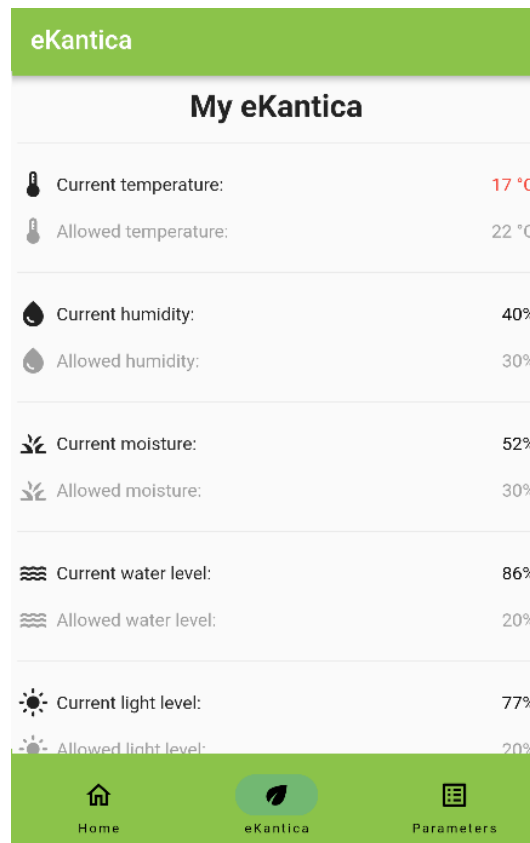


Slika 15. Stranica trenutnih vrijednosti

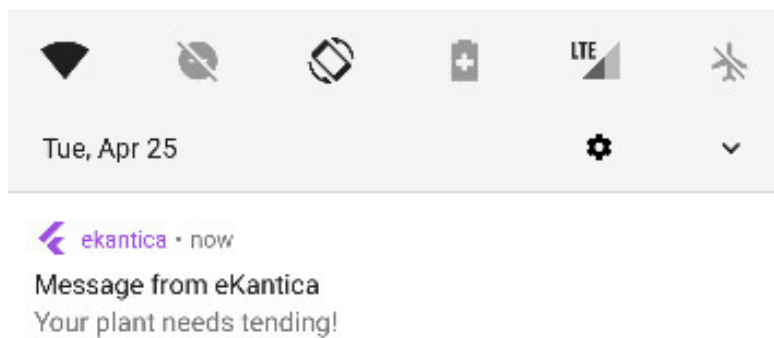
The screenshot shows the 'Parameters' page with a list of monitored parameters:

Current temperature:	17 °C
Allowed temperature:	15 °C
Current humidity:	40%
Allowed humidity:	30%
Current moisture:	52%
Allowed moisture:	30%
Current water level:	86%
Allowed water level:	20%
Current light level:	77%
Allowed light level:	20%
<b>UV</b> Current UV:	<b>31%</b>
<b>UV</b> Allowed UV:	<b>20%</b>

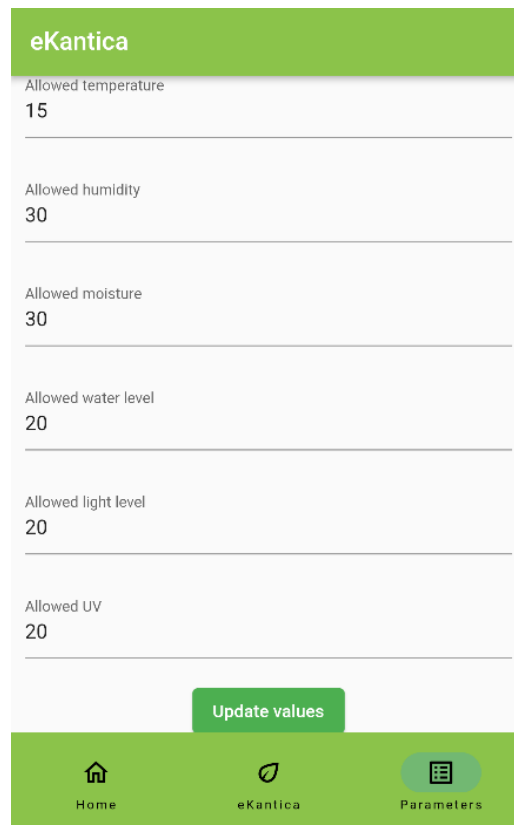
Slika 16. Popis praćenih parametara



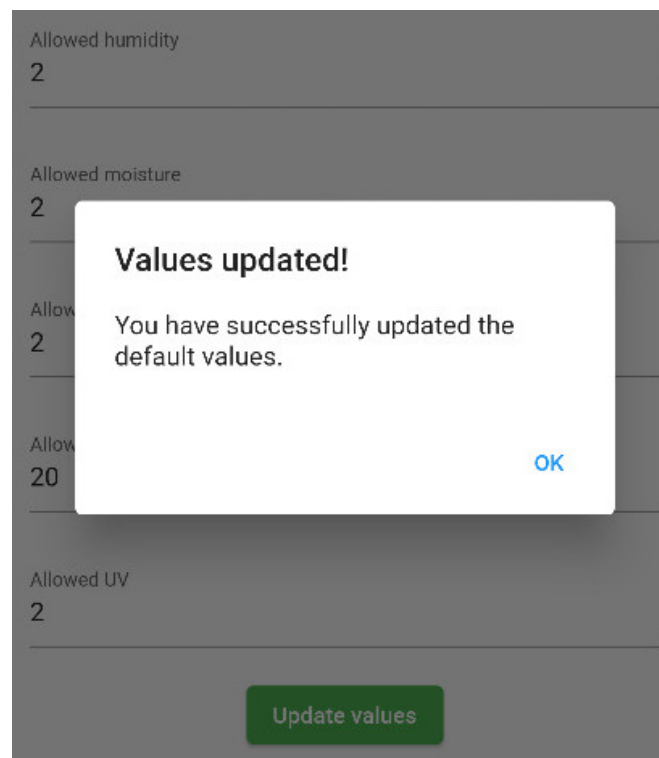
Slika 17. Parametri ispod dozvoljene vrijednosti



Slika 18. Obavijest aplikacije



Slika 19. Podešavanje željene vrijednosti parametara



Slika 20. Obavijest o uspješnom podešavanju vrijednosti



Aplikacija uz komunikaciju s korisnikom, komunicira i s API-jem putem `/set_default` i `/fetch_data` funkcija. Funkcija `fetchData` u aplikaciji koristi HTTP (engl. *Hyper Text Transfer Protocol*) paket za slanje *GET* zahtjeva na URL (engl. *Uniform Resource Locator*) "`https://ekantica-railway.up.railway.app/api/fetch_data`" i čeka odgovor servera [13]. Ako je status odgovora *200 OK*, odgovor u JSON (engl. *JavaScript Object Notation*) formatu analizira se, a relevantni podaci se izdvajaju i pohranjuju u varijable stanja pomoću metode `setState`. Ako poslužitelj ne vrati odgovor *200 OK*, izbacuje se iznimka s prikladnom porukom o neuspješnom dohvat u podataka. Ova funkcija je asinkrona, što znači da se izvršava u pozadini bez blokiranja glavne dretve aplikacije. To znači da se funkcija može izvoditi istovremeno s drugim zadacima bez čekanja na završetak izvođenja. U ovom slučaju, funkcija koristi `await` za dohvat odgovora od HTTP zahtjeva, a zatim izvršava odgovarajuće radnje ovisno o statusu odgovora. Time se ne blokira korisnikov rad u aplikaciji.

Pri mijenjanju dozvoljenih parametara korisnik pritišće gumb *Update values* koji ima postavljenu funkciju povratnog poziva `onPressed` (engl. *callback*). Prvo se koristi `_formKey.currentState!.save` da se spremi trenutno stanje svih tekstualnih polja u formi. Zatim se poziva `setParameters` funkcija s argumentima koji su izvučeni iz tekstualnih polja. Ta funkcija mijenja interne varijable mobilne aplikacije koje služe za praćenje dozvoljenih vrijednosti. Nakon toga se poziva funkcija `checkTending`, koja provjerava treba li korisnik intervenirati u brizi za biljku. Funkcija također provjerava je li potrebno zaliti biljku, i pritom postavlja parametar `needsTending` u obliku *boolean*.

Dalje se gradi URL *string* s parametrima `upita` koji sadrže dozvoljene vrijednosti postavljenih parametara, a zatim se poziva `http.post` metoda koja šalje *POST* zahtjev na API s tim parametrima. Zahtjev se šalje HTTP protokolom na `api/set_default`. Ako postoji greška, aplikacija će je uhvatiti.

Zbog manjka resursa aplikacija je dostupna samo na Android uređajima. Međutim, u idućim iteracijama omogućilo bi se korištenje aplikacije i na iOS uređajima.

## 4.2. Aplikacijsko programsko sučelje

Most između programskog dijela i sklopovlja sustava ostvaren je pomoću aplikacijskog programskog sučelja. Web aplikacija koja izlaže API postavljena je na javnom poslužitelju pomoću besplatnog servisa za objavu aplikacija. Za izradu API-ja korišten je radni okvir za razvoj web aplikacija Django, u programskom jeziku Pythonu.

API u obliku internih varijabli pohranjuje vrijednosti primljene s ESP32 mikrokontrolera o trenutnim vrijednostima očitanih sa senzora, kao i donje dozvoljene vrijednosti parametara.

API nudi nekoliko krajnjih točaka (engl. *endpoints*):

- `api/fetch_data` – služi za dohvat trenutnih i dozvoljenih vrijednosti parametara,
- `api/set_default` – služi za postavljanje graničnih vrijednosti,
- `/dana` – služi za postavljanje trenutnih vrijednosti parametara.

Mikrokontroler podatke očitane sa senzora šalje *GET* zahtjevom putem HTTP protokola na `/data`, a API ih uspoređuje sa zapisanim dozvoljenim vrijednostima, i na temelju usporedbe šalje mikrokontroleru prikladan odgovor. Trenutna implementacija podržava zalijevanje biljke isključivo na temelju vlažnosti zemlje - ako je trenutna vrijednost očitana sa senzora manja od dozvoljene, API mikrokontroleru šalje HTTP odgovor da je biljku potrebno zaliti. API šalje odgovor u obliku stringa, odnosno *True* ako je zalijevanje potrebno i *False* ako nije.

## 4.3. Mreža

Podaci dobiveni sa senzora šalju se na poslužitelj korištenjem WiFi mreže. ESP – IDF pruža gotova sučelja za WiFi komunikaciju. ESP se postavlja u način rada *WiFi Station* tj. klijent i spaja na predefiniranu WiFi mrežu. Aplikacijski protokol je HTTPS (engl. *Hypertext Transfer*

*Protocol Secure*), a transportni TLS (engl. *Transport Layer Security*) tj. SSL (engl. *Secure Sockets Layer*), što omogućuje sigurnu enkriptiranu vezu. U trenutnoj verziji ne provjerava se certifikat poslužitelja što je moguća nadogradnja za budućnost. Očitavanja senzora oblikuju se u *query string* koji se šalje na poslužitelj kao dio HTTP *GET* zahtjeva. U HTTP odgovoru poslužitelj šalje informaciju je li potrebno uključiti pumpu za zalijevanje na temelju analize podataka sa senzora.

#### **4.4. Sleep**

ESP32 podržava nekoliko načina rada niske potrošnje: *Modem Sleep*, *Light Sleep*, *Deep Sleep* i *Hibernation*. ESP32 ima četiri glavna načina rada s energijom: *Modem Sleep*, *Light Sleep*, *Deep Sleep* i *Hibernation*.

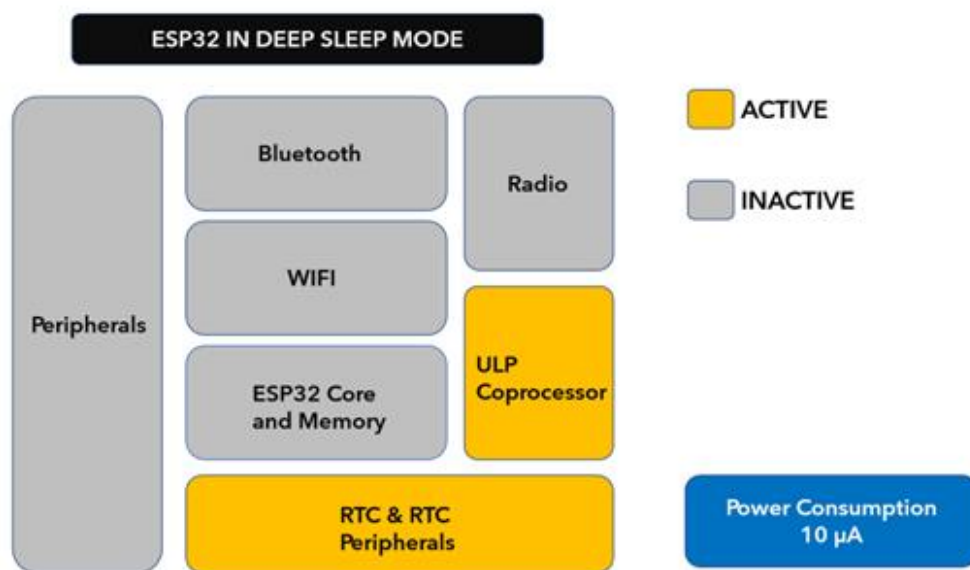
*Modem Sleep* je oblik potrošnje energije u kojem samo Wi-Fi ili Bluetooth podsustavi rade, a glavni CPU je ugašen. U ovom načinu rada, ESP32 može primiti bežične signale i održavati vezu bez trošenja puno energije.

*Light Sleep* je način rada u kojem se CPU i većina perifernih uređaja isključuju, ali RAM ostaje aktivan. U ovom načinu rada, ESP32 može brzo povratiti svoje stanje ako primi zahtjev za izvršavanje.

*Deep Sleep* je oblik potrošnje energije u kojem se CPU, većina perifernih uređaja i RAM isključuju. U ovom načinu rada, ESP32 troši vrlo malo energije i može se koristiti u projektima koji zahtijevaju dugotrajnu bateriju. Za vraćanje u normalni način rada, ESP32 se mora resetirati.

*Hibernation* je najdublji način rada u kojem se sve funkcije ESP32 isključuju i stanje sustava se sprema u nevladajuću memoriju RTC. U ovom načinu rada, ESP32 ne troši energiju, ali ne može primiti signale. Za povratak u normalni način rada, ESP32 se mora resetirati [14].

U sklopu *eKantica* koristi se način rada *Deep Sleep* prikazan slikom 21. *Deep Sleep* podrazumijeva gašenje napajanja svim jedinicama mikrokontrolera osim ULP (engl. *ultra low power*) koprocesora i RTC (engl. *Real Time Clock*) periferije. To znači da se gube svi podaci u radnoj memoriji i stanje glavnog procesora. No, to ne predstavlja problem jer je *eKantica* projektirana tako da ne mora pamtit stanje između dva očitavanja senzora što je izvedeno na serveru. Korištenjem ovog načina rada može se postići potrošnja reda veličine 10 uA za vrijeme spavanja u usporedbi s 200 mA tijekom normalnog rada. To predstavlja znatnu uštedu energije jer *eKantica* očitava stanje senzora i komunicira s poslužiteljem razmjerno rijetko. Za buđenje iz *Deep Sleep* načina rada brine se RTC periferija, koja budi procesor nakon isteka zadanog vremenskog intervala.



Slika 21. Blok shema ESP32 u *Deep Sleep* modu [14]

#### **4.5. Analogno – digitalna konverzija**

ADC (engl. *Analog- to-digital Conversion*) konverzija postupak je pretvaranja analogne ulazne vrijednosti u digitalni format. Ovaj proces sastoji se od dva glavna koraka: uzorkovanja i kvantizacije.

Uzorkovanje se odnosi na proces mjerenja ulaznog signala u redovitim vremenskim intervalima. Ovaj proces se obično provodi uz pomoć uzorkovatelja, koji čita ulazni signal i pretvara ga u niz digitalnih vrijednosti. Frekvencija je ključni čimbenik u određivanju kvalitete konverzije, jer viša frekvencija omogućava preciznije uzorkovanje i bolju reprodukciju originalnog signala.

Kvantizacija je postupak pretvaranja analognog signala u digitalni format. Ovaj proces uključuje klasificiranje svake vrijednosti signala u jedan od mogućih diskretnih vrijednosti, koje su obično određene brojem bitova u digitalnom formatu. Na primjer, ako koristimo 8-bitni ADC, svaka vrijednost signala će biti kvantizirana u jednu od 256 mogućih vrijednosti.

Ovaj proces kvantizacije može dovesti do grešaka u konverziji, poznatih kao kvantizacijske pogreške. Ove greške mogu biti manje ili veće, ovisno o broju bitova u digitalnom formatu i opsegu vrijednosti koje se pretvaraju.

U konačnici, ADC konverzija omogućuje pretvaranje analognih signala u digitalni format koji se može dalje obrađivati i pohraniti u digitalnim uređajima. Kvaliteta konverzije ovisi o brojnim čimbenicima, uključujući stopu uzorkovanja, broj bitova u digitalnom formatu i kvalitetu uzorkovatelja i kvantizatora [15].



Slika 22. ADC [15]

Funkcije za osnovnu inicijalizaciju ADC-a koja je potrebna ostalim komponentama koje koriste analogno-digitalnu pretvorbu implementirane su u zaglavlju `ADC.h`. Koristi se 12-bitna rezolucija za što preciznije mjerenje, a nivo smanjenja napona podešen je na 6 dB. Funkcija `adc1_config_width(width)` postavlja opisanu konfiguraciju.

#### 4.6. VMA303 senzori

Senzor za vlažnost tla je analogni senzor koji se koristi za mjerenje količine vlage u tlu. On se sastoji od dvije elektrode koje se stavljaju u zemlju i koje su izolirane od tla. Kada se senzor nalazi u tlu, voda u tlu djeluje kao vodič između elektroda. Kada se primijeni napon na elektrode, struja prolazi kroz tlo, a senzor mjeri otpor između elektroda. Što je veći otpor, to je manje vlage u tlu. Budući da su senzori za vlažnost tla i zemlju analogni, vraćaju analogni signal koji je analogno-digitalnom pretvorbom potrebno pretvoriti u digitalni signal.

Senzor za vlažnost tla spojen je na ADC kanal 2, te se pri pokretanju sustava konfigurira nivo smanjenja napona tog kanala. Budući da se mjerenja obavljaju rijetko, analogno-digitalna pretvorba se ne odvija kontinuirano, već se pokreće i zaustavlja programski. Funkcija `read_moisture` pokreće čitanje *raw* vrijednosti sa senzora i vrši kalibraciju senzora. Vlažnost vode vraća u obliku cijelog broja koji predstavlja postotak.

Kod kalibracije senzora uzete su dvije karakteristične vrijednosti ADC-a, a to su vrijednost ADC-a kod suhe zemlje i vrijednost kod mokre zemlje. Granične su vrijednosti dobivene

eksperimentalnim putem - senzor je najprije umetnut u potpuno suhu zemlju i dobivena vrijednost odabrana je kao donja granica vlažnosti zemlje. Iduće mjerenje dobiveno je umetanjem senzora u zemlju jako natopljenom vodom. Također, izmjerena je vrijednost koju senzor vraća pri potpunom uranjanju u vodu. Dobivena vrijednost jednaka je vrijednosti dobivenoj pri izrazito vlažnoj zemlji, stoga je ta vrijednost uzeta kao gornja granica vlažnosti zemlje. Vrijednost kod suhe zemlje odgovara 0%, a vrijednost kod mokre zemlje odgovara 100%, dok se sve između računa prema jednadžbi pravca.

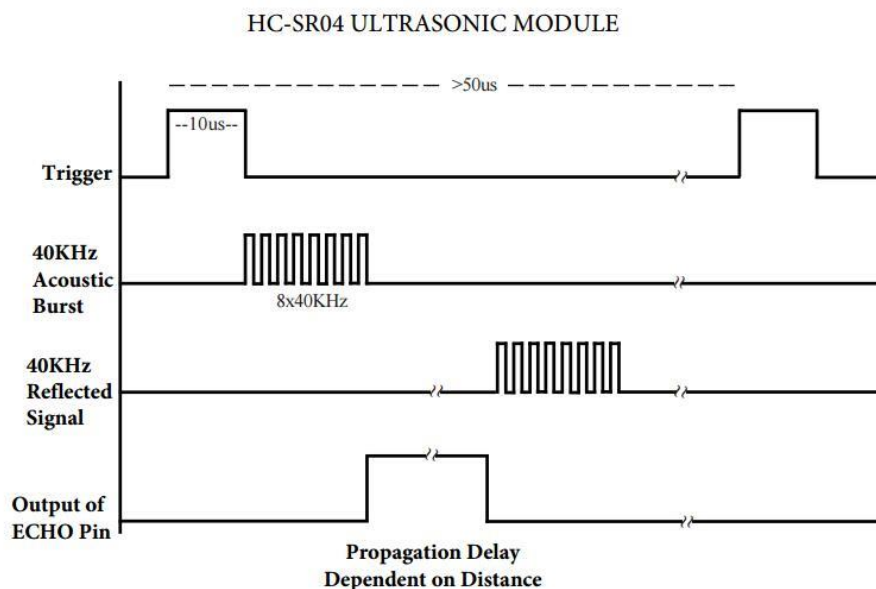
Senzor za razinu vode u spremniku spojen je na ADC kanal 3 koji se konfigurira na jednak način kao i senzor za vlažnost tla. Čitanje i kalibriranje senzora obavlja funkcija `read_water_level`, a čitanje i kalibracija senzora vrše se na ranije opisan način. Granične vrijednosti su dobivene potpunim uranjanjem senzora u vodu, te čitanjem vrijednosti sa senzora na zraku. Međutim, potpuno suh senzor ne vraća jednaku vrijednost kao i senzor koji je bio uronjen u vodu i zatim izvađen na zrak - kapljice vode ostanu na senzoru, stoga vraća vrijednosti više nego što daje potpuno suh senzor. Uzevši to u obzir, za donju granicu odabrana je vrijednost vlažnog senzora umjesto potpuno suhog, jer senzor nikada neće biti u potpunosti suh iako nije uronjen u vodu.

#### **4.7. HC- SR04 senzor**

HC-SR04 senzor ultrazvučni je senzor za mjerenje udaljenosti. U ovom projektu koristi se za mjerenje razine vode u spremniku. U kodu se koriste dva GPIO izvoda za komunikaciju sa senzorom. *TRIG\_PIN* se koristi za slanje ultrazvučnog signala, a *ECHO\_PIN* za primanje signala odbijenog od prepreke. Definiramo funkciju `hcsr04_task()` koja se izvodi u beskonačnoj petlji, a pokreće se u glavnoj funkciji.

Prvo su inicijalizirani GPIO izvodi koji se koriste za senzor. `gpio_pad_select_gpio` je korišten za odabir GPIO izvoda, a `gpio_set_direction` postavlja smjer izvoda u ulazni ili izlazni.

Slanje ultrazvučnog signala prema prepreci izvodi se pomoću PWM-a (engl. *pulse-width modulation*). Prvo se postavlja `TRIG_PIN` na nisku razinu (0) i čeka se 2  $\mu$ s. Zatim se `TRIG_PIN` postavlja na visoku razinu (1) i čeka se 10  $\mu$ s prije nego što se ponovno postavi na nisku razinu (0). Prethodni postupak prikazan je slikom 23.



Slika 23. Vremenski dijagram HC-SR04 [6]

Nakon što je poslan ultrazvučni signal, očekuje se odbijanje signala od prepreke i vraćanje natrag do senzora. Trajanje signala koji se vraća mjeri se pomoću `ECHO_PIN` izvoda i funkcije `gpio_get_level`. Ovo vrijeme govori koliko dugo je signal putovao od senzora do prepreke i natrag.

Korištenjem funkcije `esp_timer_get_time` mjerenje trajanja signala izvodi se na mikrosekundnoj razini. U prvoj petlji čeka se dok `ECHO_PIN` ne postane visok, odnosno dok



se signal ne vrati natrag do senzora. Ako se dogodi da je čekanje predugo, prekida se petlja i ispisuje se poruka u serijsku konzolu.

U drugoj petlji mjeri se trajanje signala dok je *ECHO\_PIN* visok. Opet se provjerava trajanje signala i ako petlja predugo traje, prekida se. U slučaju da je sve prošlo kako treba, izmjereno trajanje signala sprema se u varijablu *duration*.

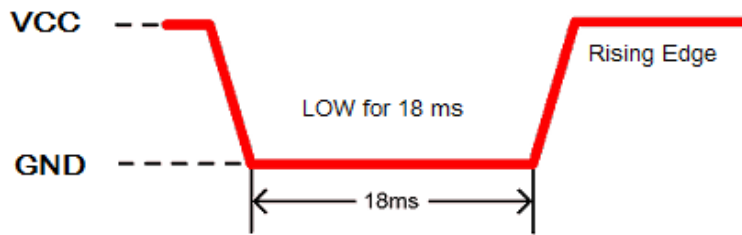
Formula koja se upotrebljava za izračun udaljenosti je  $distance = duration * 0.034 / 2$  te ona daje vrijednost udaljenosti prepreke u centimetrima. Dalje se jednostavnom formulom dobije postotak napunjenosti spremnika.

#### **4.8. DHT11 senzor**

DHT11 osnovni je digitalni senzor temperature i vlažnosti. Senzor dolazi unaprijed kalibriran i ne zahtjeva vanjski krug za mjerenje temperature ili vlažnosti. Za implementaciju korištena je službena biblioteka komponenti ESP-IDF *frameworka* [16].

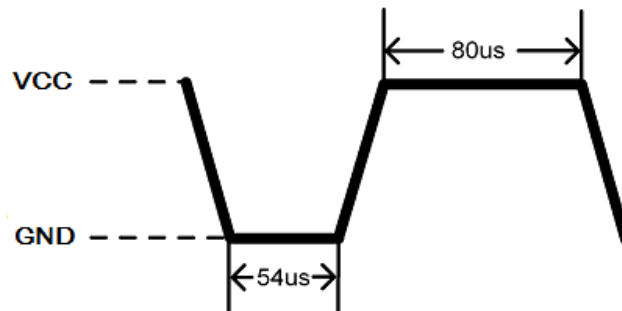
U glavnom programu implementacije senzora poziva se funkcija `dht_read_float_data`, koja provjerava mjere li se i temperatura i vlažnost te vraća izmjerene vrijednosti u postotnom obliku. Ova je funkcija primarno omotač oko funkcije `dht_read_data` koja dohvaća podatke u kritičnom odsječku. Dohvaćeni su podaci polje bajtova, gdje svaki par bajtova predstavlja jednu vrijednost očitane sa senzora. Nad tim se podacima zatim poziva funkcija `dht_convert_data` koja po dva bajta pretvara u jednu 16-bitnu vrijednost.

Dohvat podataka u kritičnom odsječku obavlja se funkcijom `dht_fetch_data`. Ovaj se proces sastoji od četiri faze. Prva je početni signal prikazan slikom 24. Za početak komunikacije s DHT11 senzorom, mikrokontroler mora poslati startni signal na *DATA* izvod DHT11 senzora. *DATA* izvod tada prema zadanim postavkama prelazi u *high*. Početni signal je *low* tijekom 18 ms, nakon čega slijedi prijelaz s *low* - *high* tj. uzlazni rub.



Slika 24. Početni signal DHT11 [17]

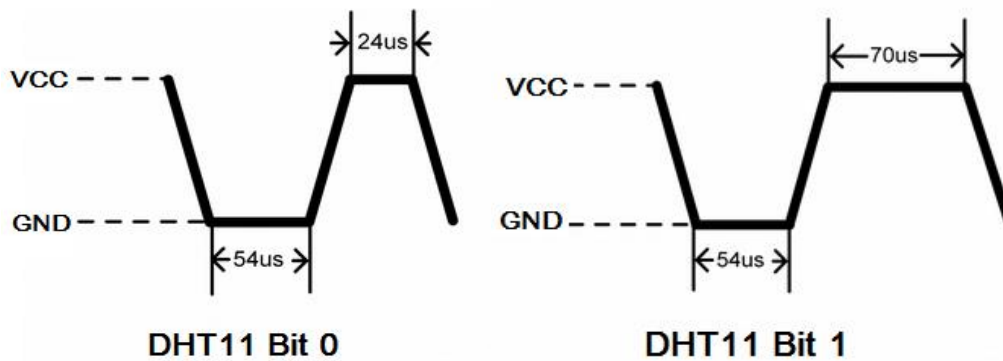
Druga je faza odgovor prikazana slikom 25. Nakon primitka početnog signala od mikrokontrolera, DHT11 šalje signal odgovora koji označava da je spreman za prijenos podataka senzora. Impuls odgovora je *low* 54  $\mu$ s, a zatim *high* tijekom 80  $\mu$ s.



Slika 25. Odgovor DHT11 [17]

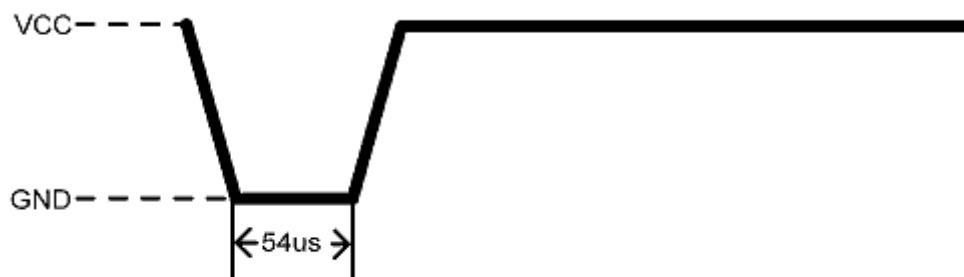
Tijekom treće faze stižu podatci. Nakon slanja impulsa odgovora, DHT11 će početi slati podatke senzora koji sadrže vrijednosti vlažnosti i temperature zraka te bajt kontrolne sume. Paket podataka sastoji se od 40 bitova.

Bitovi se prenose kao vremenski signali, gdje širina impulsa digitalnog signala određuje je li to bit 1 ili bit 0. Bit 0 počinje s *low* signalom (GND) u trajanju od 54  $\mu$ s, nakon čega slijedi *high* signal (VCC) trajanja 24  $\mu$ s.



Slika 26. Signal DHT11 logičke nule i jedinice [17]

Posljednja faza je kraj paketa. Nakon prijena 40-bitnog paketa podataka, senzor šalje *low* na 54 µs, a zatim će preći u *high* na podatkovnom pinu što je prikazano na slici 27. Nakon toga senzor prelazi u stanje mirovanja tj. niske potrošnje energije [17].



Slika 27. Kraj DHT11 podatkovnog paketa [17]

Senzor temperature i vlažnosti DHT11 ne koristi ADC jer šalje digitalni signal. Senzor DHT11 komunicira s mikrokontrolerom ili kontrolnom pločom pomoću *one-wire* digitalnog protokola, koji ne zahtjeva ADC. Digitalni izlaz senzora je *high* ili *low*, što predstavlja očitavanja temperature i vlažnosti zraka. Mikrokontroler zatim može koristiti ovaj digitalni signal za izračunavanje stvarne vrijednosti temperature i vlažnosti.

## 4.9. I2C protokol

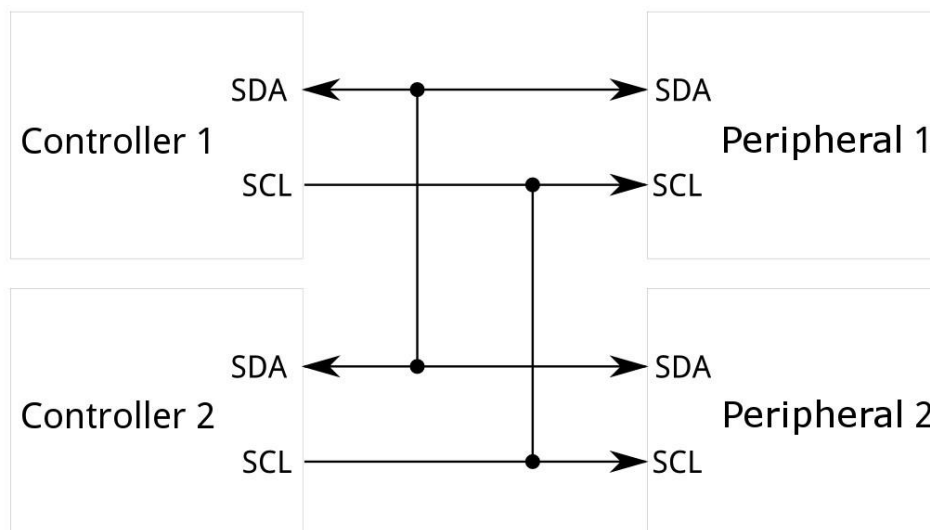
I2C je serijski komunikacijski protokol koji se koristi za komunikaciju između mikrokontrolera, senzora, aktuatora i drugih digitalnih uređaja. I2C je protokol koji je razvila tvrtka NXP Semiconductors i danas se koristi u širokom rasponu aplikacija.

I2C koristi dvije žice za komunikaciju: SDA (engl. *Serial Data*) i SCL (engl. *Serial Clock*). SDA linija prenosi podatke, dok SCL linija kontrolira vrijeme u kojem se podaci prenose. Ove dvije linije zajedno omogućuju sinkronu serijsku komunikaciju između uređaja.

I2C koristi tzv. *master - slave* arhitekturu, gdje je jedan uređaj, *master*, odgovoran za kontrolu komunikacije i inicijaciju prijenosa podataka, dok su ostali uređaji, *slave*, pasivni sudionici u komunikaciji. *Master* uređaj kontrolira prijenos podataka kroz SCL liniju i koristi SDA liniju za slanje ili primanje podataka određenog *slave* uređaja.

I2C uređaji koriste adresiranje kako bi se osiguralo da se podaci razmjenjuju samo između određenog *master* i *slave* uređaja. Svaki *slave* uređaj ima jedinstvenu adresu, što omogućuje *master* uređaju da prepozna kojem uređaju je namijenjen prijenos podataka. Osim toga, I2C protokol podržava i više *master* uređaja, što omogućuje fleksibilnost u dizajnu mreže.

I2C je relativno sporiji protokol u usporedbi s drugim serijskim protokolima kao što su SPI i UART, ali ga karakterizira manja potrošnja energije i jednostavnija implementacija u odnosu na druge protokole [18]. Zbog svojih karakteristika, I2C se često koristi u aplikacijama s ograničenom potrošnjom energije ili ograničenim resursima, kao što su senzorske mreže, IoT uređaji i slično.



Slika 28. I2C blok shema [18]

#### 4.10. SI1145 senzor

Senzor SI1145 digitalni je senzor koji mjeri razinu osunčanosti, a s mikrokontrolerom komunicira I2C protokolom.

Funkcija `si1145_init` inicijalizira SI1145 senzor. Prvo se šalje naredba da se postavi u način rada kontinuiranih očitavanja, što znači da će senzor automatski mjeriti vrijednosti i pohranjivati ih u registre koji se mogu kasnije čitati.

Funkcija `si1145_read` čita podatke iz senzora i sprema ih u varijable `vis` za vidljivu svjetlost i `uv` za UV zračenje, koje su pokazivači na 16-bitne cjelobrojne vrijednosti. Senzor može mjeriti i infracrvenu svjetlost, ali za ovaj projekt nije korištena ta mogućnost. Prvo se šalje naredba da se počne s očitavanjem podataka s adrese `SI1145_REG_VISDATAL` u SI1145 senzoru. Zatim se čita 8 bajtova podataka, koji se sastoje od vrijednosti vidljivog svjetla i UV indeksa. Ti se podaci zatim pohranjuju u polje bajtova `buf`. Nakon toga se očitane vrijednosti dekodiraju i sprema se svaka vrijednost u odgovarajuću varijablu.

Funkcija `main` glavna je funkcija programa gdje se prvo inicijalizira I2C sabirnica s izvodima SDA i SCL, a zatim se inicijalizira SI1145 senzor pozivom funkcije `si1145_init`. Nakon toga, program ulazi u beskonačnu petlju koja čita podatke s senzora pozivom funkcije `si1145_read`, ispisuje ih na serijski izlaz i čeka 1 s prije nego što ponovno pročita podatke.

#### **4.11. Pumpa**

Zalijevanje biljke odvija se pomoću pumpe koju kontrolira relej. Relej se kontrolira preko GPIO izvoda sa mikrokontrolera. Izrađena je pločica na koju se konektorima povežu pumpa, napajanje od 12 V i signal sa mikrokontrolera koji je galvanski odvojen optoizolatorom od ostatka kruga.

Relej se pali i gasi upravljačkim GPIO signalom sa mikrokontrolera. Taj signal ulazi u optoizolator DP817C kako bi napajanja mikrokontrolera i releja s pumpom bila potpuno odvojena. Na GPIO signalu nalazi se otpornik od 220  $\Omega$  koji ograničava struju na približno 20 mA što je dovoljno za rad LED diode u optoizolatoru.

Relej SRD-12-VDC-SL-C zahtjeva napajanje 12 V. Paralelno zavojnici releja nalazi se dioda 1N4007. Ona služi zaštititi pri gašenju zavojnice, kako bi inducirani napon potjerao struju kroz diodu. Spojena je tako da ne vodi dok je relej upaljen. Zavojnica releja spojena je jednim krajem na 12 V a drugim na kolektor NPN tranzistora 2n222 čijim paljenjem se drugi kraj zavojnice spaja približno na masu. Emiter je povezan na masu. Baza tranzistora preko otpornika od 1k  $\Omega$  spojena je na optoizolator te kad optoizolator primi signal tranzistor provede, time poteče struja kroz zavojnicu i relej se uključuje.

Pumpa je negativnim polom spojena na negativni pol napajanja, a pozitivni pol pumpe spojen je na NC izlaz releja, kako bi pumpa bila upaljena kada je upravljački signal u visokoj razini.

## 5. Rezultat

Rezultat projekta je sustav za automatsko zalijevanje biljaka koji prati većinu bitnih parametara biljkinog okoliša vezanih za njen rast i razvoj prikazan na slikama 29. - 34. Uređaj se sastoji od dva ESP32-C3-DevKitM-1 mikrokontrolera. Na jedan su povezani senzor razine osunčanosti SI1145 koji s mikrokontrolerom komunicira I2C serijskim komunikacijskim protokolom te VMA303 senzor za mjerenje razine vlage tla u zemlji koji vraća analogni signal koji se AD konverzijom pretvara u digitalni.

Drugi mikrokontroler upravlja pumpom preko releja te su na njega spojeni senzor za mjerenje temperature i tlaka zraka DHT11 koji s mikrokontrolerom komunicira *one-wire* protokolom te VMA303 senzor za mjerenje razine vode u spremniku čiji signal također prolazi AD konverziju. Prvi mikrokontroler stavlja se u biljku, a drugi je ostavljen na poklopac spremnika vode. Sustav se napaja prijenosnim baterijama. Popis izvoda nalazi se u tablici 1.

Dva ESP32 mikrokontrolera šalju podatke na API. Mobilna aplikacija prati zahtjeve poslane na API koji analizira podatke dobivene sa ESP32 mikrokontrolera. Mobilna aplikacija također služi za komunikaciju s korisnikom. Korisnik u nju upisuje idealne vrijednosti za svoj biljku te preko aplikacije dobiva obavijesti i upozorenja o stanju biljke i sustava.

Ovakav sustav razlikuje se od ostalih na tržištu jer nije ograničen na veličinu biljke, napajan je baterijama te ima mobilnu aplikaciju preko koje korisnik u svakom trenutku u nekoliko dodira mogu zadatai parametre te provjeriti trenutno stanje biljke i sustava.

Trenutno radi na prijenosnu bateriju ali postoji mogućnost daljnjeg razvoja i stavljanja coin baterije. Uređaj radi stabilno te je potpuno funkcionalan, stoga se može zaključiti da je cilj projekta uspješno ostvaren.

Tablica 1. Popis izvoda

ESP32-C3-DevKitM-1	DHT11
3.3V	Vcc (+)
GND	GND (-)
GPIO 10	S
ESP32-C3-DevKitM-1	VMA303 zemlja
5V	Vcc (+)
GND	GND (-)
GPIO 2	S
ESP32-C3-DevKitM-1	VMA303 voda
5V	Vcc (+)
GND	GND (-)
GPIO 3	S
ESP32-C3-DevKitM-1	Si1145
3.3V	Vin
GND	GND
GPIO 6	SCL
GPIO 7	SDA
ESP32-C3-DevKitM-1	Relej



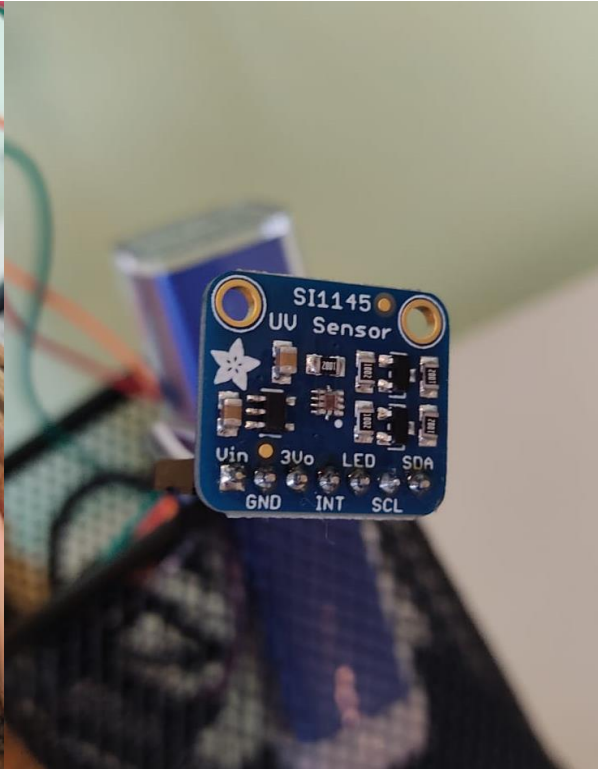
GPIO 5	PWR
GND	GND



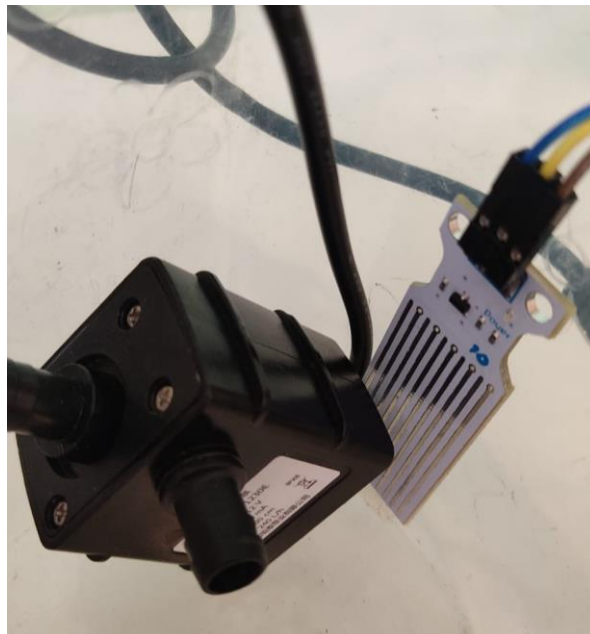
Slika 29. Rezultat projekta



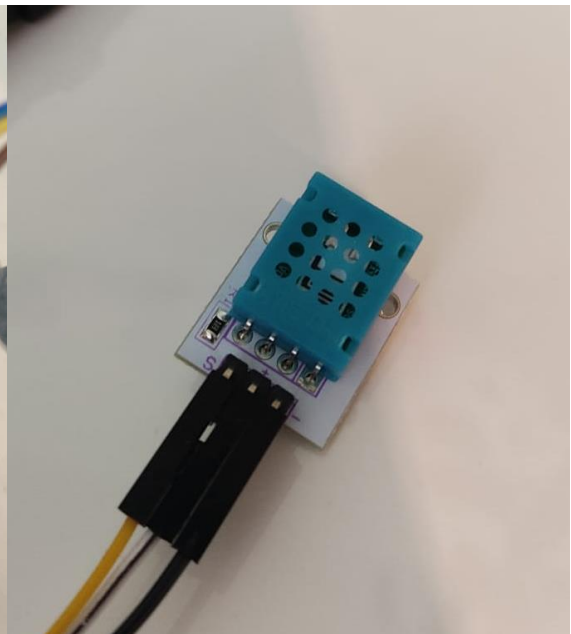
Slika 30. VMA303 senzor vlažnosti tla



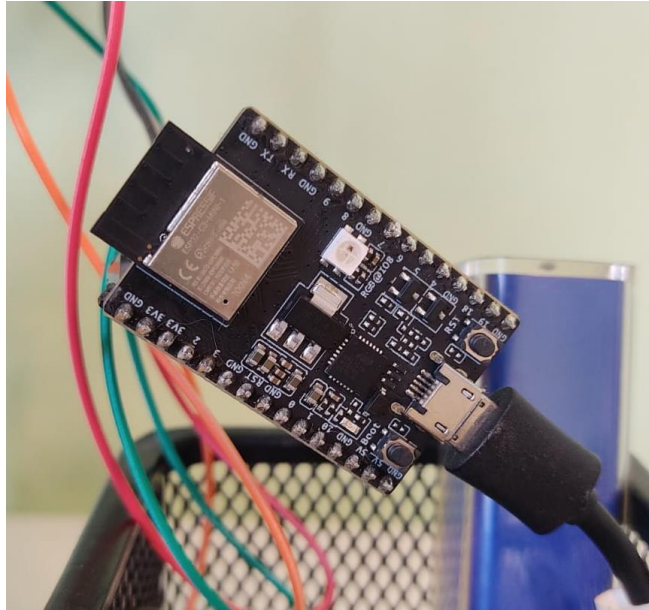
Slika 31. SI1145 senzor osunčanosti



Slika 32. Pumpa za vodu i VMA303 senzor razine vode



Slika 33. DHT11 senzor temperature i vlažnosti zraka



Slika 34. ESP32-C3-DevKitM-1

## 6. Održivost, komercijalizacija i budući razvoj

Završetkom ovog projekta tj. prototipa uređaja za automatsko zalijevanje biljaka koji prati i druge parametre u okolišu biljke otvorile su se nove mogućnosti i ideje. Prije svega, potrebno je ukloniti nedostatke trenutnog sustava - naći senzore koji više odgovaraju sustavu, smanjiti količinu žica, dizajnirati prikladno kućište, postaviti umjesto prijenosnih baterija drugi izvor napajanja itd. Uređaj je trenutno potpuno upotrebljiv i stabilno radi, ali ukoliko bi se još neko vrijeme razvijao, mogao bi postati uređaj za komercijalne svrhe.

Neke od lako izvedivih ideja za budući razvoj primjenjivih na „*eKantica 2.0*“ su da bude potpuno samoodrživa tj. da radi na male solarne panele koji bi zamijenili baterije. Također, već tijekom provedbe projekta započeta je realizacija ideje da svaki senzor bude odvojen te komunicira s jednim glavnim mikrokontrolerom pa bi bilo korisno implementirati i mogućnost spajanja više „*eKantica*“ na isti mikrokontroler koji svima upravlja, što bi bilo korisno za upotrebu na većim ekosustavima. Uz to, poželjno bi bilo prijeći na BLE (engl. *Bluetooth Low Energy*) tehnologiju jer WiFi IoT sustavi koriste do 10 puta više električne energije od BLE IoT sustava [19]. Na kraju, moguće je napraviti bazu podataka u koju će se spremati podatci sa senzora te će biti moguće korisniku preko aplikacije pružiti statistike o vrijednostima u nekom vremenskom razdoblju. Također, moguće je napraviti i bazu biljki za koje bi parametri bili unaprijed uneseni pa korisnik ne bi sam morao upisivati vrijednosti već samo odabrati biljku o kojoj će sustav brinuti.

## 7. Zaključak

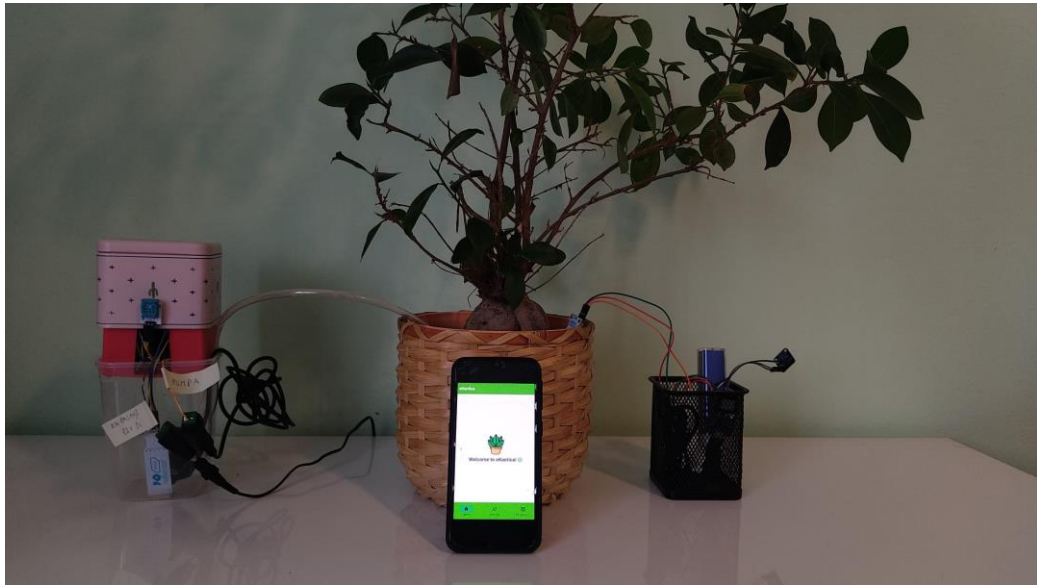
U zaključku može se istaknuti da je IoT automatski sustav za zalijevanje biljaka izrazito koristan alat u domovima i agrikulturi. Tradicionalni sustavi za zalijevanje biljaka oslanjaju se na vremenski raspored i ljudsko praćenje stanja biljaka i tla, što može dovesti do prekomjernog ili nedovoljnog zalijevanja biljaka, te potrošnje velikih količina vode i drugih resursa. Automatski sustav za zalijevanje biljaka temelji se na sensorima u tlu koji prate vlažnost tla, temperaturu i druge parametre te automatski upravljaju procesom zalijevanja na temelju tih podataka. Također, ovaj sustav omogućuje daljinski nadzor i upravljanje putem interneta, što olakšava kontrolu i održavanje sustava.

Prednosti ovakvog sustava su mnogobrojne. Precizno zalijevanje na temelju stvarnog stanja pomaže u održavanju optimalne razine vlage u tlu, što može dovesti do boljeg rasta i razvoja biljaka, povećane produktivnosti te smanjenja potrošnje vode i drugih resursa. Automatizirani sustav za zalijevanje također može smanjiti potrebu za ljudskom intervencijom u procesu zalijevanja, što štedi vrijeme i smanjuje troškove radne snage u poljoprivredi ili je jednostavno koristan uređaj za navodnjavanje cvijeća u kući.

Međutim, važno je imati na umu da trenutni sustav ima nedostatke. Neke od mana sustava su senzori koji ne odgovaraju sustavu u potpunosti, količina žica koje spajaju komponente te rasipaju energiju i unose smetnje, čimbenici poput pH vrijednosti koji utječu na biljku, a njihovo motrenje nije obuhvaćeno ovim sustavom te sam sustav zahtjeva redovito održavanje kako bi se osigurala točnost podataka. Ipak važno je napomenuti da je ovo prvi prototip uređaja koji se svakako može dalje razvijati.

Unatoč navedenim nedostacima, IoT automatski sustav za zalijevanje biljaka ima veliki potencijal te može pridonijeti povećanju produktivnosti. Stoga će ovaj sustav više istraživati i razvijati kako bi se poboljšale njegove funkcionalnosti i učinkovitost u budućnosti te povećala

moćnost uspješne komercijalizacije. Daljnji razvoj mogao bi uključivati integraciju drugih senzora i tehnologija koje bi omogućile još preciznije i učinkovitije zalijevanje biljaka, kao i veću automatizaciju i daljinsko upravljanje.



Slika 35. *eKantica*

## **8. Zahvala**

Zahvaljujemo mentoru prof. dr. sc. Hrvoju Džapi na pomoći i savjetima tijekom izrade projekta te pisanja ovog rada kao i asistentici Karli Salamun, mag. ing.

## 9. Popis literature

[1] What is IoT?, Oracle

<https://www.oracle.com/internet-of-things/what-is-iot/>

Zadnji pristup 21.4.2023.

[2] 10 Best Automatic Watering Systems, Electronics Hub

<https://www.electronicshub.org/best-automatic-watering-system-for-indoor-plants/>

Zadnji pristup 21.4.2023.

[3] Best Self-watering Planters and Plant Pots, Gardeners World

<https://www.gardenersworld.com/product-guides/growing/best-self-watering-planters/>

Zadnji pristup 21.4.2023.

[4] 4 Factors That Affect Plant Growth, Eco Gardener

<https://ecogardener.com/blogs/news/4-factors-that-affect-plant-growth>

Zadnji pristup 21.4.2023.

[5] What Is An API?, Amazon

<https://aws.amazon.com/what-is/api/>

Zadnji pristup 21.4.2023.

[6] HC-SR04 Datasheet, Sparkfun

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

Zadnji pristup 21.4.2023.



[7] WPSE303 Datasheet, Velleman

<https://www.velleman.eu/downloads/25/wpse303a4v01.pdf>

Zadnji pristup 21.4.2023.

[8] SI1145 Datasheet, Adafruit

<https://cdn-shop.adafruit.com/datasheets/Si1145-46-47.pdf>

Zadnji pristup 21.4.2023.

[9] DHT11 Datasheet, Mouser

<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

Zadnji pristup 21.4.2023.

[10] ESP32-C3-DevKitM-1 User Guide, Espressif

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/hw-reference/esp32c3/user-guide-devkitm-1.html>

Zadnji pristup 21.4.2023.

[11] ESP32-C3 Datasheet, Espressif

[https://www.espressif.com/sites/default/files/documentation/esp32-c3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf)

Zadnji pristup 21.4.2023.

[12] Flutter Documentation, Flutter

<https://docs.flutter.dev/>

Zadnji pristup 21.4.2023.

[13] HTTP, Mozilla

<https://developer.mozilla.org/en-US/docs/Web/HTTP>

Zadnji pristup 21.4.2023.

[14] ESP32 Sleep Modes & Their Power Consumption, Linux Hint

<https://linuxhint.com/esp32-sleep-modes-power-consumption/>

Zadnji pristup 21.4.2023.

[15] ADC, Tech Target

<https://www.techtarget.com/whatis/definition/analog-to-digital-conversion-ADC>

Zadnji pristup 21.4.2023.

[16] ESP-IDF Components library, ESP – IDF

<https://esp-idf-lib.readthedocs.io/en/latest/>

Zadnji pristup 21.4.2023.

[17] Arduino compatible coding 15: Reading sensor data from DHT-11 without using a library, Engineers Garage

<https://www.engineersgarage.com/articles-arduino-dht11-humidity-temperature-sensor-interfacing/>

Zadnji pristup 21.4.2023.

[18] I2C, Sparkfun

<https://learn.sparkfun.com/tutorials/i2c/all>

Zadnji pristup 21.4.2023.

[19] Bluetooth vs WiFi Comparison For the IoT Solutions, Netguru

<https://www.netguru.com/blog/bluetooth-vs-wifi-comparison-for-the-iot-solutions>

Zadnji pristup 21.4.2023.

## 10. Sažetak i ključne riječi

**Razvoj IoT sustava za automatsko zalijevanje biljaka „eKantica“,**

**Jelena Gavran, Hana Novak, Petar Sušac**

### **Sažetak**

Rad opisuje provedbu razvoja IoT sustava za automatsko zalijevanje biljaka. Prije planiranja projekta provedeno je istraživanje o ponudi sličnih uređaja na tržištu te njihovih prednosti i nedostataka. Uređaj koji se razvijao tijekom ovog projekta pokušao je otkloniti nedostatke drugih. Prototip je uspješno razvijen te je u potpunosti funkcionalan. Pri razvoju rodilo se još mnogo ideja za poboljšanjem samog prototipa.

Odlučeno je da sustav mora pratiti većinu faktora koji utječu na rast i razvoj bilo koje biljke, a to su razina vlažnosti tla, temperatura zraka te izloženost Sunčevim zrakama. Sustav bi trebao biti što autonomniji te djelovati sa što manjim uplitanjem čovjeka, stoga je bilo potrebno integrirati i spremnik koji će opskrbljivati sustav vodom. U tom slučaju, bilo je potrebno implementirati praćenje razine vode u spremniku te pumpa koja će slati vodu iz spremnika u sustav.

Prije početka integracije sustava, obavljeno je istraživanje o komponentama. Izabrani su senzori pogodni potrebama sustava te cjenovno prihvatljivi. Odabrani su senzor osunčanosti SII145, senzor za temperaturu i vlagu zraka DHT11, paket od dva VMA303 senzora od kojih je jedan za vlažnost tla, a drugi za razinu vode u spremniku. S obzirom da senzor VMA303 za vodu nema širok raspon mjerenja, kao priručno rješenje uzet je HC-SR04 senzor koji mjeri udaljenost te je integriran u spremnik. Centralni dio sklopovlja je mikrokontroler. Odabran je ESP32-C3-DevKitM-1 jer je malih dimenzija i malene potrošnje te ima mnoštvo funkcionalnosti koje su bile potrebne za ovaj projekt.

Sustav je posložen tako da su na jedan ESP32 mikrokontroler spojeni SI1145 senzor te VMA303 senzor za razinu vlage tla, a na drugi ESP32 mikrokontroler spojeni su VMA303 senzor za razinu vode (ili HC-SR04 senzor), DHT11 senzor te pumpa za vodu.

Most između programskog dijela i sklopovlja je aplikacijsko programsko sučelje. Podatke sa senzora koji se šalju na mikrokontroler analizira API, dok mobilna aplikacija prati zahtjeve poslani na to aplikacijsko programsko sučelje. Podatci sa senzora se na poslužitelj šalju WiFi-jem. Komunikacija između sustava i korisnika obavlja se preko aplikacije. Korisnik u aplikaciju upisuje idealne podatke za biljku, a aplikacija ga upozorava i šalje obavijesti kada stvarni uvjeti nisu u skladu s traženim vrijednostima.

Rezultat istraživanja i provedbe ovog projekta je IoT sustav koji se bez daljnjeg razvoja može koristiti za zalijevanje biljaka te radi potpuno stabilno.

### **Ključne riječi**

IoT, ESP32, mreža senzora, zalijevanje biljaka, pametne kuće, automatizacija

## 11. Summary and key words

### Development of IoT System for Automatic Plant Watering "*eKantica*",

#### Summary

The paper describes implementation and development of an IoT system for automatic plant watering. Before planning the project, market research was conducted on the offer of similar devices and their advantages and disadvantages. The device developed during this project tried to eliminate the shortcomings of others. The prototype was successfully developed and is fully functional. During development, many more ideas were born to improve the prototype itself.

It was decided that the system must monitor most of the factors that affect the growth and development of any plant, namely soil moisture level, air temperature and exposure to sunlight. The system should be as autonomous as possible and operate with as little human intervention as possible, therefore it was necessary to integrate the tank that will supply the system with water. In that case, it was necessary to implement water level monitoring in the tank and a pump that will send water from the tank to the system.

Before starting the integration of the system, component research was done. The selected sensors are suitable for the needs of the system and are affordable. The SI1145 insolation sensor, the DHT11 air temperature and moisture sensor, a package of two VMA303 sensors, one for soil moisture and the other for the water level in the tank, were selected. Given that the VMA303 water sensor does not have a wide measurement range, the HC-SR04 sensor that measures the distance and is integrated into the tank was taken as a temporary solution. The central part of the circuit is the microcontroller. The ESP32-C3-DevKitM-1 was chosen because it has small dimensions and low power consumption and has many functionalities that were needed for this project.

The system is arranged so that SII145 sensor and VMA303 soil moisture level sensor are connected to one ESP32 microcontroller, and VMA303 water level sensor (or HC-SR04 sensor), DHT11 sensor and water pump are connected to another ESP32 microcontroller.

The bridge between the software part and the hardware is the application programming interface. The sensor data sent to the microcontroller is analyzed by the API, while the mobile application monitors the requests sent to that API. Data from the sensor is sent to the server via WiFi. Communication between the system and the user is done through the mobile application. The user enters the ideal data for the plant into the application, and the application warns him and sends notifications when the actual conditions do not correspond to the required values.

The result of the research and development of this project is an IoT system that can be used for watering plants without further development and works completely stably.

### **Key words**

IoT, ESP32, sensor network, plant watering, smart homes, automatization