

**SVEUČILIŠTE J. J. STROSSMAYER U OSIJEKU**

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

**Diplomski studij Automobilsko računarstvo i komunikacije**

**UGRADBENI RAČUNALNI SUSTAVI**

**izv. prof. dr. sc., Tomislav Keser**

**TEHNIČKA DOKUMENTACIJA PROJEKTA**

# **AUTOMATSKA HRANILICA ZA PSE**

**Marko Ćosić**

**Akademska godina 2022./2023.**

**Osijek, 14.2.2024.**

## Sadržaj

1. Uvod.....	1
1.1 Zadatak i struktura rada .....	1
2. Sustav doziranja hrane, slatkiša i vode na temelju uzrasta psa i ovisno o dobu dana .....	2
2.1 Teorijski osvrt na sustav doziranja hrane, slatkiša i vode na temelju uzrasta psa i ovisno o dobu dana .....	2
2.2 Prijedlog sklopovskog rješenja.....	4
2.3 Prijedlog programskog rješenja.....	5
3. Realizacija doziranja hrane, slatkiša i vode ovisno u dobu dana i uzrastu psa.....	6
3.1 Korišteni alati i programska okruženja .....	6
3.2 Realizacija konstrukcijskog i sklopovskog rješenja .....	7
3.3 Realizacija programskog rješenja.....	17
4. Testiranje i rezultati.....	19
4.1 Metodologija testiranja .....	19
4.2 Rezultati testiranja .....	19
5. Zaključak .....	23
6. Literatura .....	24
7. Prilozi i dodaci .....	25
7.1 Korištene komponente .....	25
7.2 Programski kod.....	31

# 1. Uvod

U današnje vrijeme, može se primijetiti da se način života ubrzan u usporedbi s onime kakav je bio prije 30-40 godina. Ljudi sve više rade, zauzeti su, uvijek žure i sve manje vremena provode kod kuće. Ovo se na prvi pogled ne čini kao problem, ali je problem ukoliko ti isti ljudi imaju kućnog ljubimca, točnije psa. Pas, iako je čovjekov najbolji prijatelj, predstavlja i određenu obavezu. Potrebno ga je prošetati, nahraniti, pobrinuti se da ima vode, posebice u vrućim ljetnim danima. Neke tvrtke, poput Xiaomija, nastojale su riješiti ovaj problem [1], ali riješile su ga djelomično. Xiaomi je navedeni problem riješio automatskom hranilicom koja ima mogućnost doziranja suhe hrane. Ideja ovog projektnog zadatka je ponuditi bolje rješenje od onih kakva su danas na tržištu, a to je automatska hranilica za pse koja ima mogućnost doziranja hrane, slatkiša i vode. Doziranje hrane, slatkiša i vode prvenstveno bi se radilo u ovisnosti o uzrastu psa kojeg vlasnici imaju. Moguće je odabrati jedan od tri predefinirana uzrasta, a ovisno o tome odabere li korisnik malog, srednjeg ili velikog psa, u posudici za hranu će se dozirati 20, 40 ili 60 grama hrane. Slatkiši se također doziraju u ovisnosti o odabranom uzrastu psa. Doziranje vode radi na način da pas uvijek ima punu posudicu vode. Samo doziranje hrane, slatkiša i vode radi se dva puta dnevno, točnije ujutro u 7:00 sati i navečer u 19:00 sati. Kako bi se osiguralo da pas tijekom cijelog dana ima dovoljno vode, posebice u vrućim ljetnim danima, dodana je provjera vode svakih sat vremena, te ukoliko je potrebno, dolijeva se razlika vode. Ukoliko je pas tijekom dana pojeo samo dio predviđene hrane, u večernjem doziranju hrane i slatkiša dobit će samo razliku hrane kako bi dobio predviđenu količinu za taj dan.

## 1.1 Zadatak i struktura rada

Zadatak rada je dizajnirati automatsku hranilicu za pse koja će omogućiti prethodno opisano doziranje hrane, slatkiša i vode za pse. Potrebno je osigurati da hranilica dozira hranu, slatkiše i vodu u ovisnosti o odabranom uzrastu psa. Sama hranilica treba biti jednostavna i jasna za korištenje, što se nastoji ostvariti korištenjem tri gumba i LCD zaslona za unos vremena i odabir uzrasta psa. U drugom poglavlju će biti opisana teorijska podloga i teorijski osvrt na sam projekt. Također, bit će predstavljeni prijedlozi sklopovskog i programskog rješenja. U trećem poglavlju će biti predstavljena opisna karakteristika samog sustava, točnije korištene komponente, alati i programska okruženja, realizacija konstrukcijskog i sklopovskog rješenja, te realizacija programskog rješenja. U četvrtom poglavlju će biti predstavljeno testiranje i rezultati. U petom poglavlju će biti iznesen zaključak. U šestom poglavlju će biti literatura, dok će u sedmom poglavlju biti prilozi i dodaci.

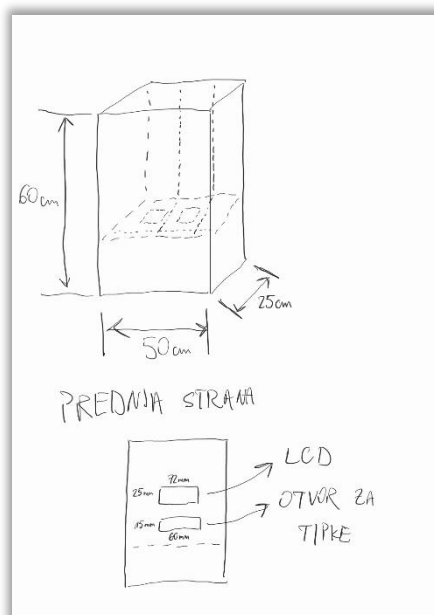
## **2. Sustav doziranja hrane, slatkiša i vode na temelju uzrasta psa i ovisno o dobu dana**

Kako je prethodno navedeno, jedna od stvari po kojima se ova automatska hranilica za pse ističe u odnosu na druge, jest to da nudi mogućnost doziranja hrane, slatkiša i vode. Ne samo to, automatska hranilica za pse radi ovisno o odabranom uzrastu psa. Uzrast psa se može odabrati između tri predefinirana uzrasta, malog, srednjeg ili velikog psa, na temelju kojih će se dozirati 20, 40 ili 60 grama hrane, te količina slatkiša koja je primjerena za odabrani uzrast. Doziranje hrane, slatkiša i vode obavljat će se dva puta dnevno, ujutro u 7:00 sati, te navečer u 19:00 sati. Dodatno, bit će implementirana opcija provjere razine vode u posudici svakih sat vremena, kako bi se osiguralo da pas u bilo kojem trenutku dana ima dovoljno vode.

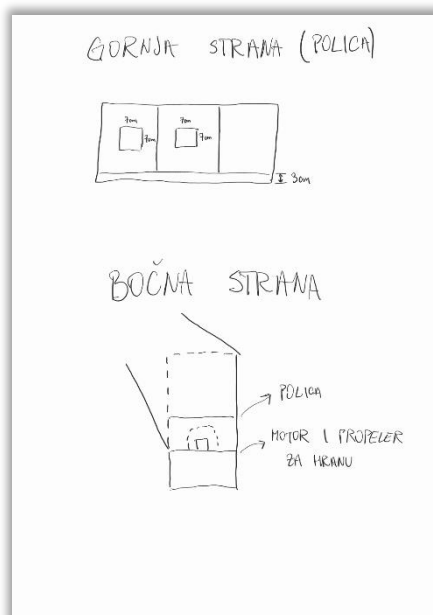
### **2.1 Teorijski osvrt na sustav doziranja hrane, slatkiša i vode na temelju uzrasta psa i ovisno o dobu dana**

Problem doziranja suhe hrane i slatkiša riješen je korištenjem zemljine sile teže. Sami spremnici za hranu i slatkiše nalaze se pri vrhu same hranilice. Otvaranjem čepova posuda, hrana i slatkiši padat će u posudice za hranu koje se nalaze pri dnu same hranilice. Ostaje problem točnog mjerenja i doziranja hrane, što je riješeno drvenim cilindrima. Drveni cilindri smješteni su točno ispod čepa posude, a u sebi imaju manje otvore. Kako se cilindri rotiraju, tako omogućuju hrani i slatkišima da padnu u manje otvore, te u konačnici padnu u posudicu za hranu. Spremnik vode također se nalazi pri vrhu hranilice. Dodatni problem koji se pojavio kod doziranja vode jest hidrostatski tlak. Zbog hidrostatskog tlaka, jednom kada voda krene kroz crijevo prema posudici za vodu, neće se zaustaviti dok se spremnik vode u potpunosti ne isprazni, ili dok kraj crijeva koji je u spremniku vode ne „povuče zrak“. Problem je riješen tako što je probušena manja rupica za zrak na dijelu crijeva koji se nalazi u spremniku vode, te čime je omogućeno kontroliranje doziranja vode. Kao što je ranije navedeno, odabir uzrasta psa omogućen je putem gumba i LCD zaslona.

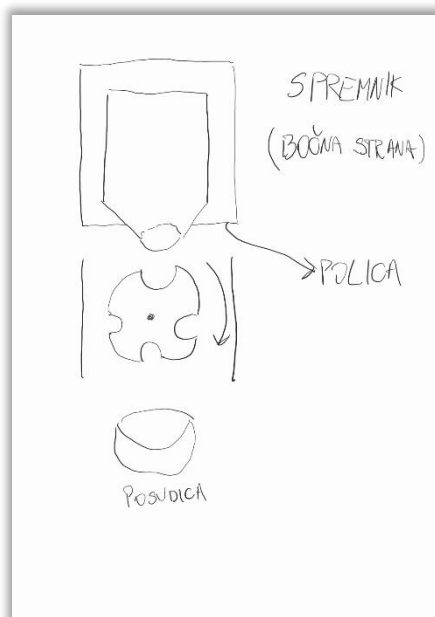
Inicijalne skice same automatske hranilice, položaja spremnika za hranu, slatkiše i vodu, posudica za hranu, slatkiše i vodu, rotirajućih cilindara te prikaz prednje strane s gumbima i LCD zaslonom prikazane su na slikama 2.1., 2.2. i 2.3. u nastavku.



**Sl. 2.1.** Skica automatske hranilice i prednje strane automatske hranilice



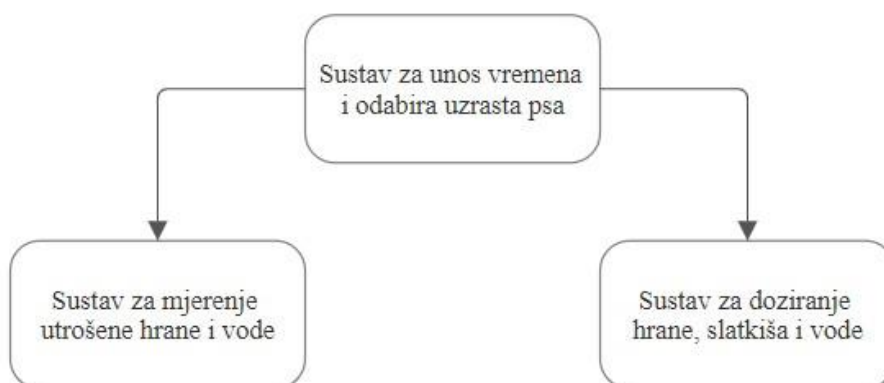
**Sl. 2.2.** Skica gornje i bočne strane automatske hranilice



**Sl. 2.3.** Skica spremnika hrane i slatkiša te rotirajućih cilindara za doziranje hrane i slatkiša

## 2.2 Prijedlog sklopovskog rješenja

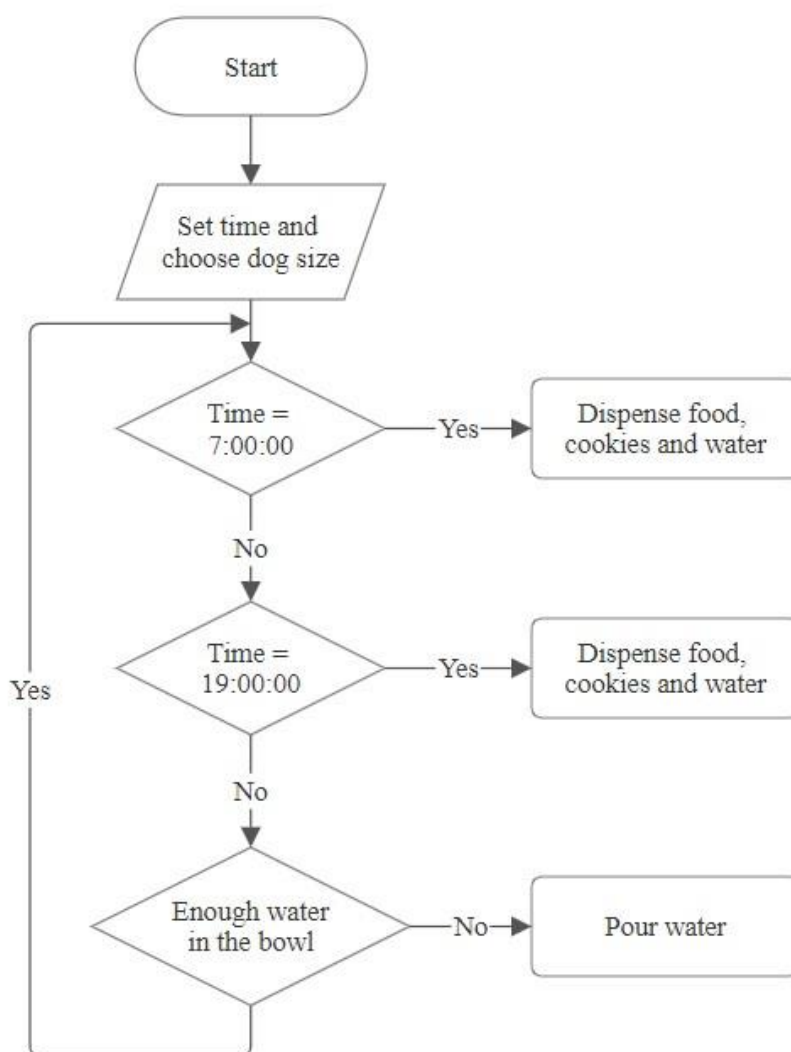
Sklopovsko rješenje sastoji se od tri sustava. Prvi je sustav, ujedno i glavni, sustav za unos vremena i odabira uzrasta psa. Na temelju ovog sustava, radit će druga dva sustava koja se koriste u sklopovskom rješenju. Prije svega, to je sustav doziranja hrane, slatkiša i vode koji će na temelju vremena, ali i odabranog uzrasta psa dozirati hranu, slatkiše i vodu. Posljednji je sustav za mjerenje utrošene hrane i vode. Ovaj sustav bi trebao ujutro u 7:00 sati, te navečer u 19:00 sati mjeriti utrošenu hranu i vodu, te na temelju toga, dozirati razliku hrane ili vode do razine koja je predviđena. Ovaj sustav također bi trebao mjeriti razinu vode u posudici za vodu svakih sat vremena. Prijedlog sklopovskog rješenja nalazi se na slici 2.4. u nastavku.



**Sl. 2.4.** Blok dijagram korištenih sustava u sklopovskom rješenju

## 2.3 Prijedlog programskog rješenja

Nakon uključivanja automatske hranilice za pse, prvo je potrebno unijeti vrijeme. Vrijeme se unosi u satima, minutama i sekundama. Nakon toga, potrebno je odabrati uzrast psa između tri predefinirana uzrasta, a to su mali, srednji i veliki pas. Nakon unosa vremena i odabira uzrasta psa, automatska hranilica za pse kreće s radom. Ujutro u 7:00 sati, te navečer u 19:00 sati dozirat će se predviđena količina hrane za odabrani uzrast psa, a uz hranu će se dozirati i slatkiši. Nakon unosa vremena i odabira uzrasta psa, napraviti će se provjera ima li dovoljno vode u posudici za vodu. Ova će se provjera vršiti svakih sat vremena kako bi se osiguralo da pas u svakom trenutku ima dovoljno vode. Prijedlog programskog rješenja nalazi se na slici 2.5. u nastavku.



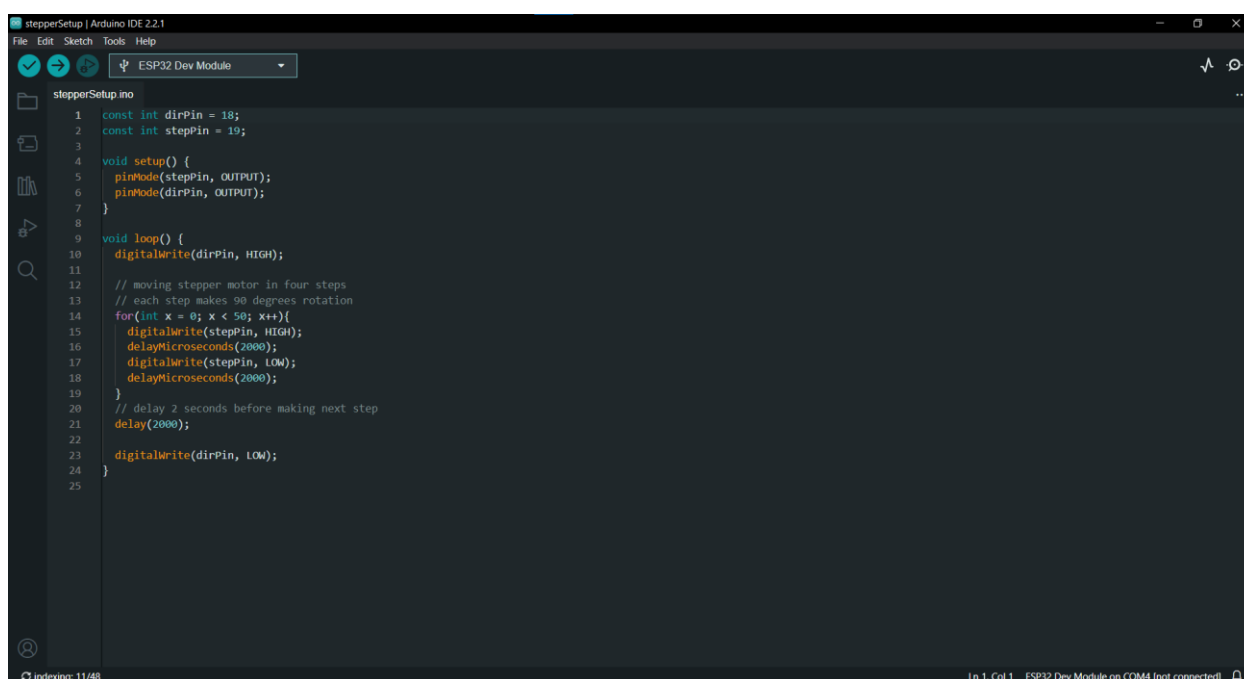
Sl. 2.5. Blok dijagram programskog rješenja

### 3. Realizacija doziranja hrane, slatkiša i vode ovisno u dobu dana i uzrastu psa

U ovom će poglavlju biti opisani korišteni alati i programska okruženja. Također, bit će predstavljena realizacija konstrukcijskog i sklopovskog rješenja, te realizacija programskog rješenja.

#### 3.1 Korišteni alati i programska okruženja

Alat korišten za razvoj programskog rješenja je Arudino IDE. Arduino IDE predstavlja razvojno okruženje koje omogućava kodiranje i učitavanje koda na mikroupravljač. Vrlo je jednostavan za korištenje, zbog čega je vrlo popularan. Ima podršku za brojne mikroupravljače i brojne biblioteke. Za realizaciju ovog projektnog zadatka korišten je mikroupravljač esp32, o kojem se može pročitati više u poglavlju 7.1 o korištenim komponentama. Kako bi se navedeni mikroupravljač koristio u Arduino IDE-u, potrebno je instalirati paket za esp32 mikroupravljač, odabrati ga u izborniku *Tools -> Board*, te odabrati port na kojem je mikroupravljač spojen s razvojnim računalom, što se radi u izborniku *Tools -> Port*. Izgled razvojnog okruženja Arudino IDE-a s primjerom proizvoljnog koda naveden je na slici 3.1. u nastavku.



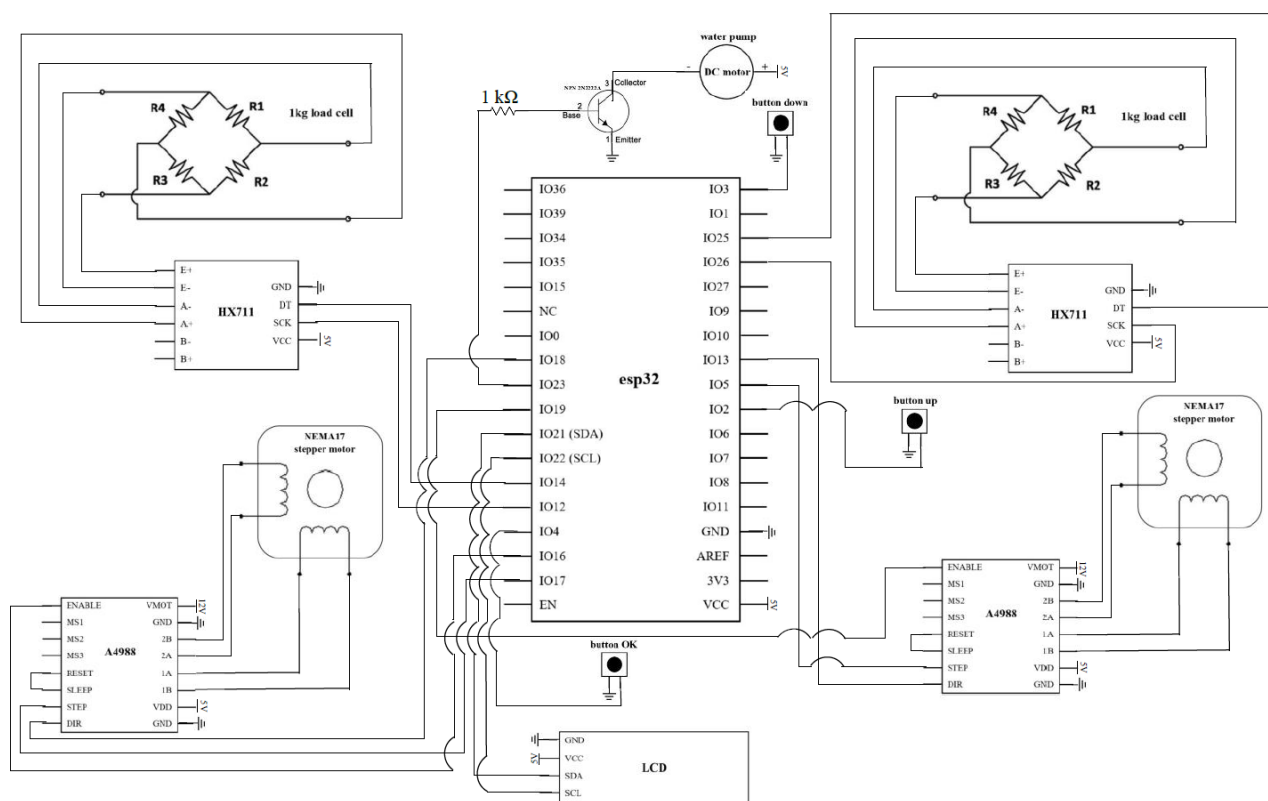
Sl. 3.1. Izgled Arduino IDE razvojnog okruženja



### 3.2 Realizacija konstrukcijskog i sklopovskog rješenja

Sklopovsko rješenje sastoji se od sljedećih komponenata: esp32 mikroupravljača, dva koračna (engl. *stepper*) motora NEMA17, dva upravljača (engl. *driver*) koračnog motora A4988, dvije vage kapaciteta od jednog kilograma, dva upravljača za vage HX711, vodene pumpe, NPN tranzistora 2N2222A, otpornika od jednog kilo oma [kΩ], tri gumba te LCD zaslona. Mikroupravljač esp32 spojen je na uzemljenje na GND pinu, te je spojen na napajanje od 5 volta na VCC pinu. Napajanje od 5 volta omogućuje rad mikroupravljača. Koračni motori NEMA17 spojeni su na pinove 1B, 1A, 2A i 2B upravljača koračnih motora. Upravljači koračnih motora spojeni su na uzemljenje na GND pinovima, na napajanje od 5 volta na VDD pinovima, te na napajanje od 12 volta na VMOT pinovima. Napajanje od 5 volta omogućuje rad samih upravljača, a napajanje od 12 volta služi za pokretanje koračnih motora. Prethodno opisani pinovi spajaju se na isti način, dok se DIR, STEP i ENABLE pinovi spajaju na drugačiji način. DIR pin prvog upravljača spojen je GPIO 13 pin esp32 mikroupravljača, a DIR pin drugog upravljača spojen je na GPIO 18 pin. Ovi pinovi služe za određivanje smjera kretanja koračnih motora. STEP pin prvog upravljača spojen je na GPIO 5 pin esp32 mikroupravljača, a STEP pin drugog upravljača spojen je na GPIO 17 pin. Ovi pinovi služe za određivanje koraka samog koračnog motora. ENABLE pin prvog upravljača spojen je na GPIO 19 pin esp32 mikroupravljača, a ENABLE pin drugog upravljača spojen je na GPIO 16 pin. Ovi pinovi služe za uključivanje i isključivanje samih koračnih motora, što se ostvaruje slanjem *HIGH* signala, odnosno logičke jedinice na njihov pin. Ovim načinom rada ostvaruje se efikasno upravljanje samim koračnim motorima što je vrlo korisno u ovako složenim projektima. Vage kapaciteta od jednog kilograma spojene su na upravljač HX711 na pinove E+, E-, A- i A+. Upravljači vaga spojeni su na uzemljenje na GND pinovima, te su spojeni na napajanje od 5 volta na VCC pinovima. Napajanje od 5 volta omogućuje rad samih vaga i upravljača. Prethodno opisani pinovi spajaju se na isti način na oba upravljača. DT pin prvog upravljača spojen je na GPIO 14 pin esp32 mikroupravljača, a DT pin drugog upravljača spojen je na GPIO 25 pin. Ovi pinovi služe za prijenos podataka s vage (engl. *DT - data*). SCK pin prvog upravljača spojen je na GPIO 12 pin esp32 mikroupravljača, a SCK pin drugog upravljača spojen je na GPIO 26 pin. Ovi pinovi služe za sinkronizaciju serijske komunikacije (engl. *SCK – Serial Clock*). Upravljanje vodenom ostvareno je pomoću NPN 2N2222A tranzistora. Vodena pumpa spojena je na napajanje od 5 volta na VCC pinu, što omogućuje njen rad. GND pin vodene pumpe spojen je na kolektor (engl. *collector*) tranzistora. Emiter (engl. *emitter*) tranzistora spojen je na uzemljenje. Baza (engl. *base*) tranzistora spojena je na GPIO 23 pin esp32 mikroupravljača preko otpornika od 1 kilo oma [kΩ]. Tri gumba, „gore“ (engl. *up*), „dolje“ (engl. *down*) i „u redu“ (engl. *OK*) koji služe za unos vremena

i odabir uzrasta psa kod pokretanja automatske hranilice na svojim su prvim pinovima spojeni na uzemljenje. Drugi pinovi gumba spojeni su na GPIO pinove esp32 mikroupravljača na slijedeći način: gumb „gore“ spojen je na GPIO 2 pin, gumb „dolje“ spojen je na GPIO 3 pin, a gumb „u redu“ spojen je na GPIO 4 pin esp32 mikroupravljača. Posljednja je komponenta LCD zaslon. Spojen je na uzemljenje na GND pinu, te na napajanje od 5 volta na VCC pinu. Napajanje od 5 volta omogućuje rad LCD zaslona. SDA pin LCD zaslona spojen je na GPIO 21 pin esp32 mikroupravljača. Ovaj pin služi za prijenos podataka (engl. *SDA – Serial Data*). SCL pin LCD zaslona spojen je na GPIO 22 pin esp32 mikroupravljača. Ovaj pin služi za sinkronizaciju prijenosa podataka (engl. *SCL – Serial Clock*). Shema prethodno opisanog sklopovskog rješenja prikazana je na slici 3.2. u nastavku.



**Sl. 3.2.** Izgled sklopovskog rješenja

Inicijalna skica automatske hranilice za pse, s odgovarajućim dimenzijama, te smještajem odgovarajućih komponenti u kućište automatske hranilice nalazi se u poglavlju 2.1 na slikama 2.1., 2.2. i 2.3. Izvedeno stanje, odnosno konstrukcijsko rješenje same automatske hranilice za pse prema prethodno navedenim skicama nalazi se na slikama u nastavku.

Na slici 3.3. u nastavku prikazan je izgled gotovog projekta, točnije izrađene automatske hranilice za pse prema inicijalnim skicama.



**Sl. 3.3.** Izgled automatske hranilice za pse

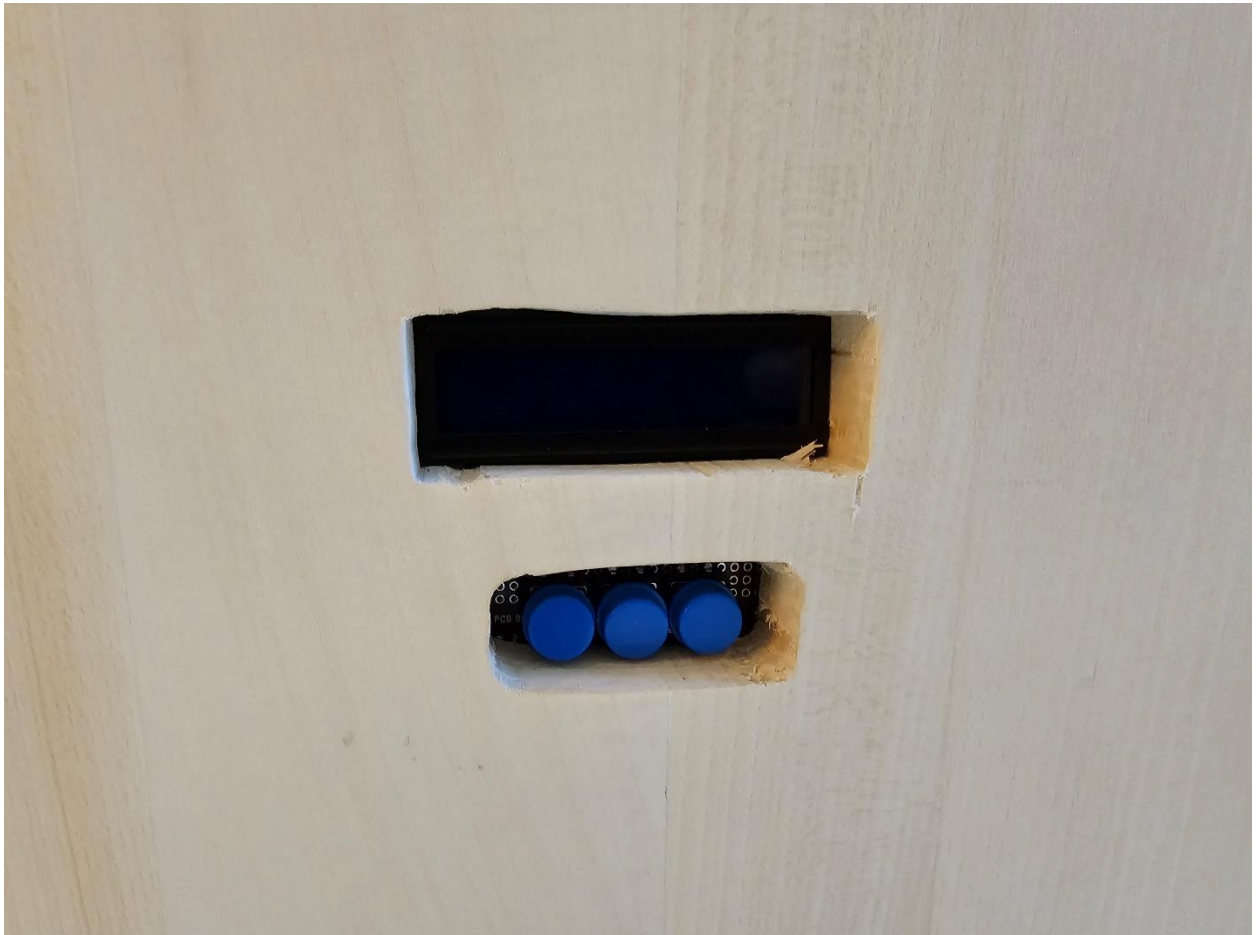
Na slici 3.4. u nastavku prikazani su spremnici u automatskoj hranilici. Spremnici s lijeva na desno: spremnik za hranu, spremnik za slatkiše, te spremnik za vodu. U spremniku za vodu može se vidjeti spojena vodena pumpa koja ulijeva vodu u posudicu za vodu.



Sl. 3.4. Izgled postavljenih spremnika u automatskoj hranilici



Na slici 3.5. u nastavku prikazan je izgled postavljenog LCD zaslona te gumba pomoću kojih se unosi vrijeme i pomoću kojih se vrši odabir uzrasta psa. Gumbi s lijeva na desno: gumb „gore“, gumb „dolje“, koji služe za odabir vremena (sati, minuta, sekunda) i uzrasta psa, te gumb „u redu“ koji služi da potvrdi unos.



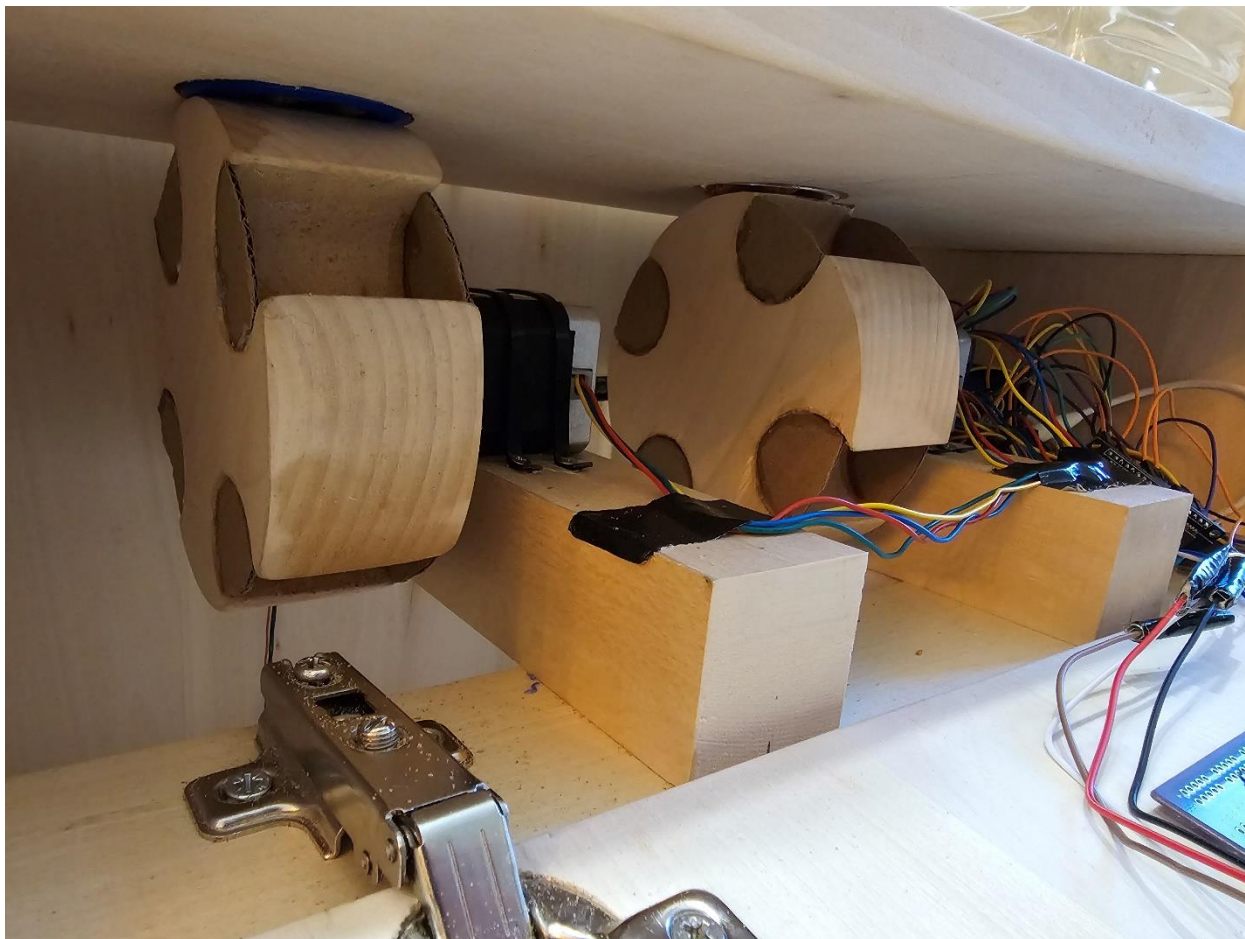
**Sl. 3.5.** Izgled postavljenih gumba i LCD zaslona

Na slici 3.6. u nastavku prikazan je izgled postavljenih posudica. Posudice s lijeva na desno: posudica za hranu, posudica za slatkiše, te posudica za vodu. Posudice za hranu i vodu ispod sebe imaju postavljenu vagu, koja služi da mjerenje utrošene hrane i vode kod doziranja.



**Sl. 3.6.** Izgled postavljenih posudica za hranu, slatkiše i vodu

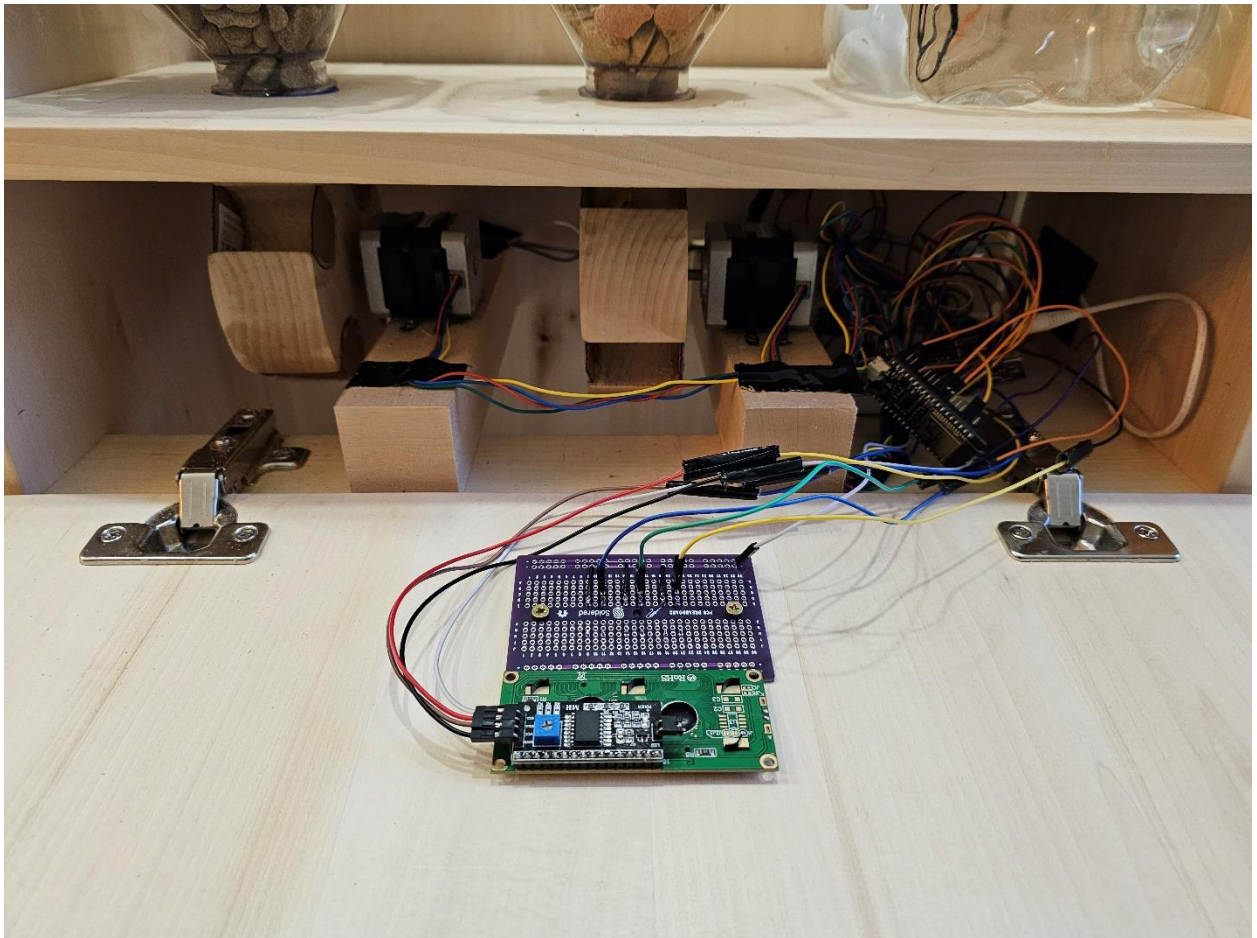
Na slici 3.7. u nastavku prikazani su drveni rotirajući cilindri koji služe za doziranje hrane i slatkiša. Svojom rotacijom zatvaraju spremnike hrane i slatkiša, ili omogućuju propadanje hrane i slatkiša u sam cilindar, ovisno u kojem se položaju nalaze. Također, na slici se mogu vidjeti NEMA17 koračni motori koji su korišteni za pokretanje samih cilindara.



**Sl. 3.7.** Izgled rotirajućih drvenih cilindara s NEMA17 koračnim motorima



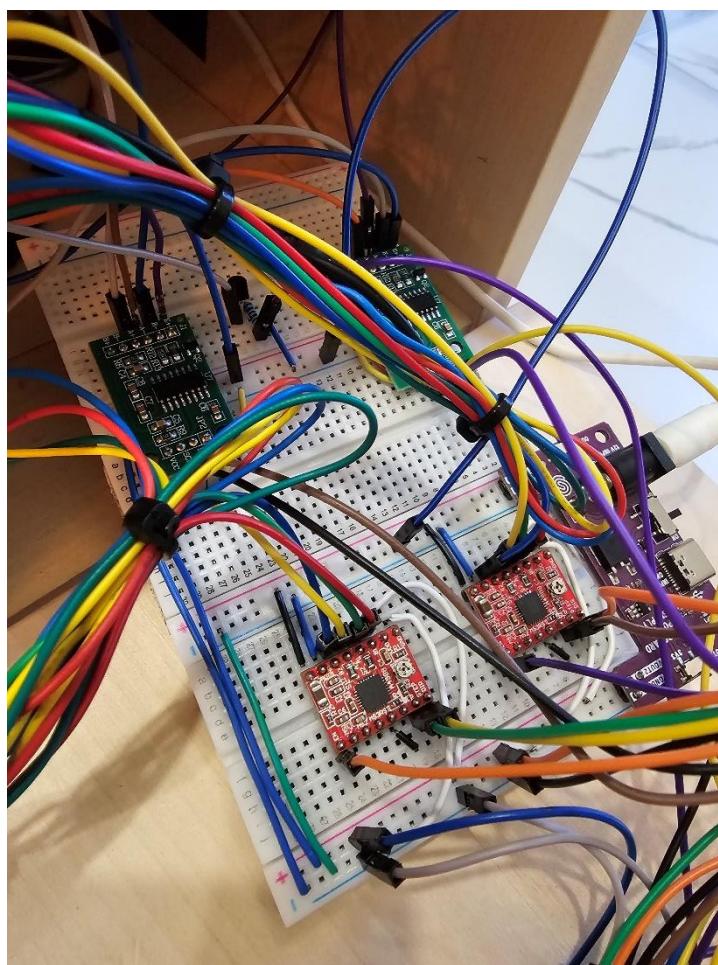
Na slici 3.8. prikazan je prednji pogled na unutrašnji dio automatske hranilice. Na slici se vide prethodno opisani rotirajući drveni cilindri s koračnim motorima, spojeni gumbi i LCD zaslon, te glavno ožičenje koje se nalazi s desne strane unutrašnjosti automatske hranilice.



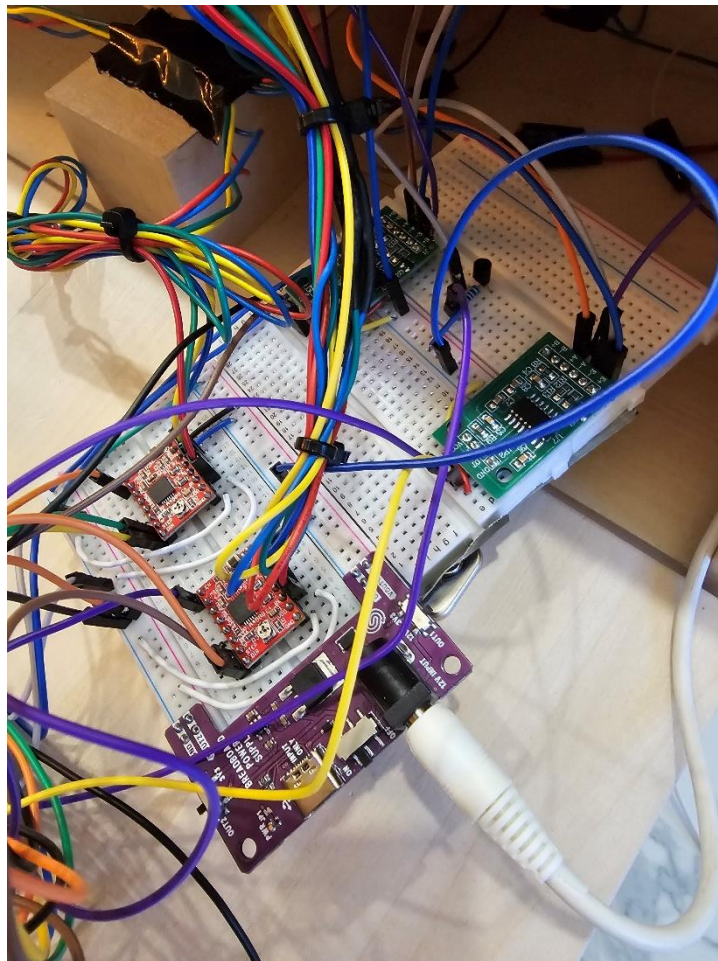
**Sl. 3.8.** Prednji pogled na unutrašnjost automatske hranilice



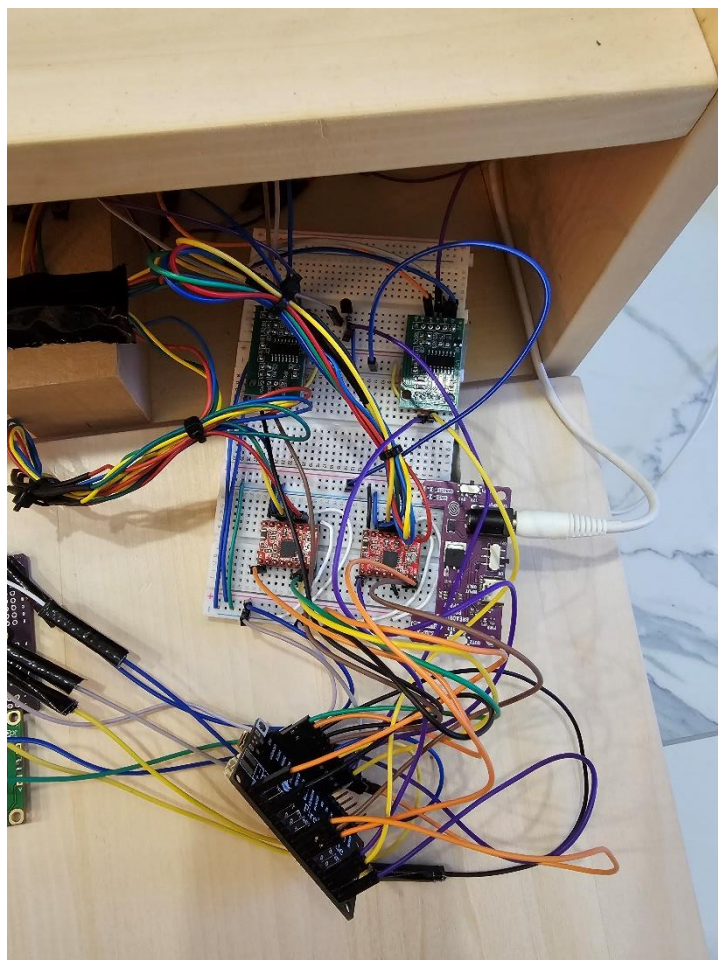
Na slikama 3.9., 3.10. i 3.11. u nastavku prikazano je glavno ožičenje projekta. Na slici se mogu vidjeti preostale komponente koje su ranije opisane, te način na koje su spojene na eksperimentalnoj pločici (engl. *breadboard*). Na slici se nalaze esp32 mikroupravljač, dva upravljača HX711 za vage (zeleni upravljači na slikama), dva upravljača A4988 za koračne motore (crveni upravljači na slikama), NPN tranzistor 2N2222A, otpornik od jednog kilo oma [k $\Omega$ ], strujni adapter (bijeli kabel na slikama) koji s 220 volta pomoću adaptera napajanja (ljubičasti adapter na slikama) pruža 12 i 5 volta na eksperimentalnoj pločici, koji su potrebni za napajanje svih prethodno opisanih komponenti.



**Sl. 3.9.** Lijevo prikaz glavnog ožičenja projekta



**Sl. 3.10.** Desni prikaz glavnog ožičenja projekta

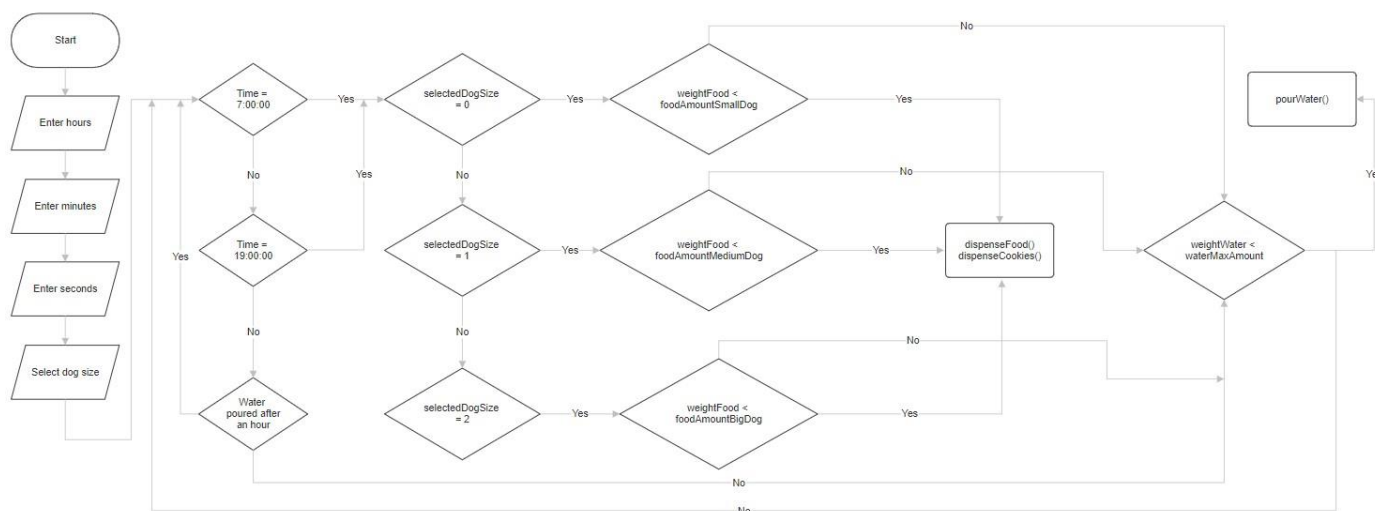


**Sl. 3.11.** Prednji prikaz glavnog ožičenja projekta

### 3.3 Realizacija programskog rješenja

Nakon uključivanja automatske hranilice, korisnik prvo mora unijeti vrijeme, a zatim vrši odabir uzrasta psa. Unos vremena i odabir uzrasta psa rade se pomoću tri gumba, a svoj odabir korisnik može vidjeti na LCD zaslonu. Korisnik prvo mora unijeti sate, pri čemu je početna vrijednost nula. Na raspolaganju su gumbi „gore“ i „dolje“ pomoću kojih korisnik može povećati ili smanjiti sate. Sati se unose u 24 satnom formatu. Nakon što je zadovoljan s unesenim satima, korisnik klikne tipku „OK“, te na isti način unosi minute i sekunde. Kao i kod sati, početna vrijednosti minuta i sekunda je nula. Nakon unesenog vremena, korisnik na isti način korištenjem tipki odabire jedan od tri predefinirana uzrasta psa, a to su mali, srednji i veliki pas. Ovisno o samom uzrastu psa koji se odabere, u posudicu za hranu će se dozirati 20, 40 ili 60 grama hrane za malog, srednjeg, odnosno velikog psa. Isto tako, slatkiši se doziraju ovisno o odabranom uzrastu psa. Kapacitet posudice za vodu je 140 mililitara, a doziranje vode je ograničeno na 120 mililitara kako bi se spriječilo prelijevanje vode iz posudice. Nakon postavljenog vremena i odabranog uzrasta psa, na LCD zaslonu prikazivat će se vrijeme i odabrani uzrast psa, te automatska hranilica može započeti s radom. Vršiti se provjera je li 7:00 sati ujutro. Ako je, sljedeća provjera koja se vrši je koji je uzrast psa odabran. *selectedDogSize* varijabla imat će vrijednost nula ukoliko je odabran mali pas, vrijednost jedan ukoliko je odabran srednji pas, te vrijednost dva ukoliko je odabran veliki pas. Na temelju *selectedDogSize* varijable, odnosno odabranog uzrasta psa, te očitavanja s vage koja se nalazi ispod posudice za hranu, vrši se provjera ima li u posudici dovoljno hrane koja je predviđena za odabrani uzrast. Tako će varijabla *foodAmountSmallDog* imati vrijednost 20, varijabla *foodAmountMediumDog* imati vrijednost 40, a varijabla *foodAmountBigDog* imati vrijednost 60. Ukoliko je očitavanje težine s vage manje od količine hrane koja je predviđena za odabrani uzrast psa, a definirana je u prethodno spomenutim varijablama, pozvat će se funkcije *dispenseFood()* i *dispenseCookies()* koje će aktivirati koračne motore kako bi se započelo doziranje hrane i slatkiša. Ove će se funkcije izvršavati dok se posudica za hranu ne napuni predviđenom vrijednošću, ili dok se ne napuni razlika utrošene hrane. Sukladno tome, dozirat će se i slatkiši za predviđeni uzrast psa. Ukoliko očitavanje s vage pokaže da je u posudici dovoljna količina hrane, izvršit će se provjera ima li dovoljno vode u posudici. Maksimalna količina vode kojom se sprječava prelijevanje vode iz posudice definirana je u varijabli *waterMaxAmount*, koja ima vrijednost 120. Ukoliko se dogodi da je u posudici količina manja od predviđene, pozvat će se funkcija *pourWater()* koja će aktivirati vodenu pumpu i doliti razliku vode. Ukoliko se u prvobitnoj provjeri pokaže da nije 7:00 sati ujutro, vrši se provjera je li 19:00 sati navečer. Ako je, vrše se iste provjere kako je prethodno opisano u slučaju da je 7:00 sati ujutro te je li potrebno dopuniti posudice za hranu, slatkiše ili

vodu. Ukoliko se u toj provjeri pokaže da nije ni 19:00 sati navečer, vršit će se provjera je li voda dolijevana u posljednjih sat vremena. Provjera se vrši pomoću *bool* varijable *waterPouredAfterAnHour*. Ukoliko je varijabla postavljena u vrijednost „istina“ (engl. *true*), program će smatrati da ne treba dolijevati vodu. Ukoliko je varijabla postavljena u vrijednost „neistina“ (engl. *false*), vršit će se provjera ima li u posudici dovoljno vode. Provjera se vrši pomoću očitavanja s vage te usporedbom s ranije objašnjenom varijablom *waterMaxAmount*. Nakon dolijevanja vode, varijabla *waterPouredAfterAnHour* postavlja se u vrijednost „istina“ (engl. *true*) te započinje odbrojavanje od sat vremena. Nakon isteka vremena, njena će se vrijednost promijeniti u „neistina“ (engl. *false*) čime će se ponovno vršiti provjera ima li u posudici dovoljno vode. Ovim mehanizmom omogućena je provjera svakih sat vremena ima li pas dovoljno vode u posudici, uz jutarnju i večernju provjeru kada se uz vodu doziraju i hrana i slatkiši. Na slici 3.12. u nastavku prikazan je detaljan blok dijagram toka programskog rješenja.



**Sl. 3.12.** Izgled programskog rješenja



## 4. Testiranje i rezultati

### 4.1 Metodologija testiranja

Testiranje automatske hranilice za pse vršit će se za sva tri predefinirana uzrasta psa, na način da će se provjeravati količina hrane koja je dozirana u samu posudicu. Ukoliko je odabran mali uzrast psa, u posudicu treba dozirati 20 grama hrane. Ukoliko je odabran srednji uzrast psa, u posudicu treba dozirati 40 grama hrane. Ukoliko je odabran veliki uzrast psa, u posudicu treba dozirati 60 grama hrane. Osim prethodno navedene tri provjere, vršit će se i provjera točnosti doziranja vode u posudicu. Kapacitet posudice za vodu je 140 mililitara, a samo doziranje je ograničeno na 120 mililitara kako bi se spriječilo prelijevanje vode iz posudice. U prvom testiranju provjeravat će se točnost doziranja hrane i vode za malog psa u deset iteracija, u drugom testiranju točnost doziranja hrane i vode za srednjeg psa u deset iteracija, u trećem testiranju točnost doziranja hrane i vode za velikog psa, također u deset iteracija. Rezultati testiranja bit će prikazani u obliku tablice koja će imati informacije o iteraciji koja se promatra u trenutnom testiranju, količini hrane koja je dozirana u posudicu za hranu, količinu vode koja je dozirana u posudicu za vodu te referentne vrijednosti kako bi se mogla vidjeti točnost same automatske hranilice. Dodatno, bit će prikazani i dijagrami kako bi se točnost doziranja mogla vidjeti i grafički.

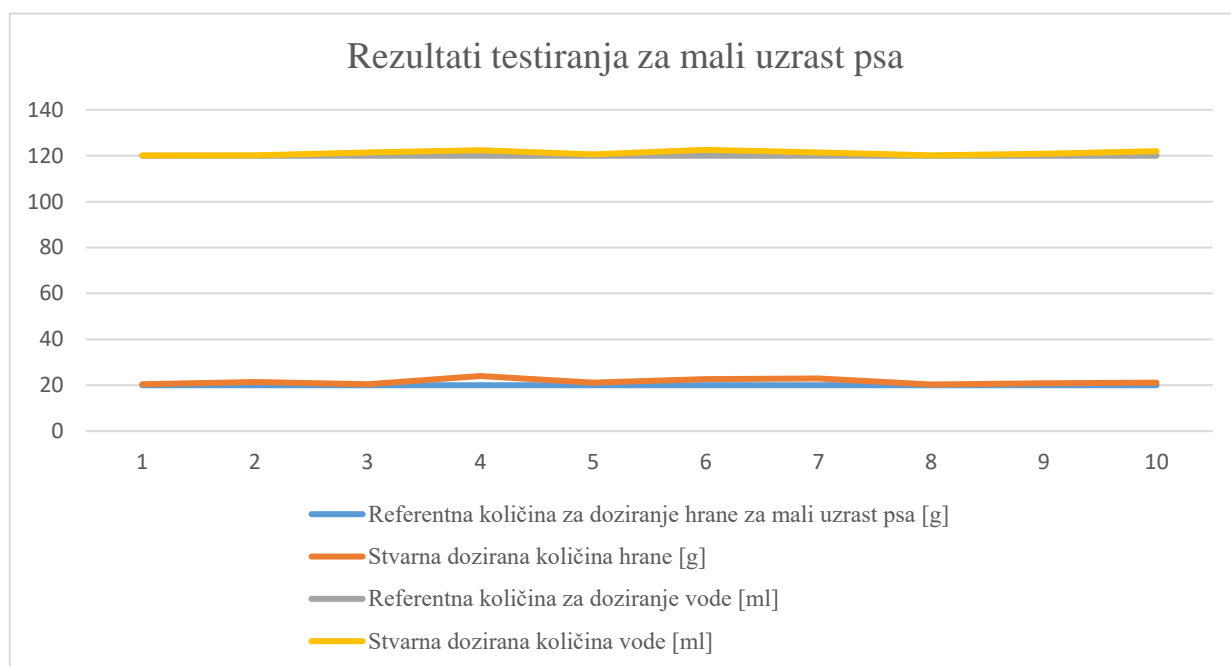
### 4.2 Rezultati testiranja

Rezultati testiranja za odabrani mali uzrast psa koji su se provodili u deset iteracija prikazani su u nastavku u tablici 4.1.

**Tab. 4.1.** Rezultati testiranja za mali uzrast psa

Iteracija	Referentna količina za doziranje hrane za mali uzrast psa (u gramima)	Stvarna dozirana količina hrane (u gramima)	Referentna količina za doziranje vode (u mililitrima)	Stvarna dozirana količina vode (u mililitrima)
1	20	20,40	120	120,12
2	20	21,37	120	120,21
3	20	20,41	120	121,36
4	20	24,02	120	122,36
5	20	21,13	120	120,58
6	20	22,65	120	122,59
7	20	22,98	120	121,48
8	20	20,32	120	120,23
9	20	20,89	120	120,89
10	20	21,08	120	122,02

Grafički prikaz rezultata testiranja za mali uzrast psa nalaze se na slici 4.1. u nastavku. Na x osi nalazi se broj iteracija, a na y osi su referentna količina hrane, količina dozirane hrane, referentna količina vode i količina dozirane vode.



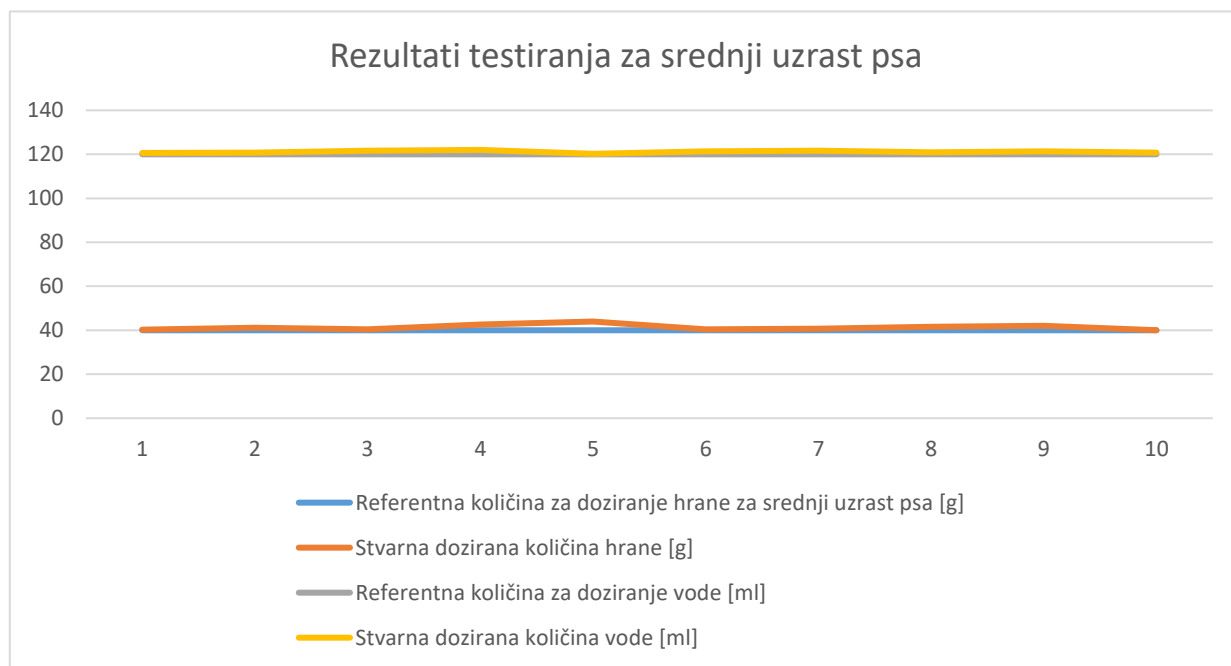
**Sl. 4.1.** Grafički prikaz rezultata testiranja za mali uzrast psa

Rezultati testiranja za odabrani srednji uzrast psa koji su se provodili u deset iteracija prikazani su u nastavku u tablici 4.2.

**Tab. 4.2.** Rezultati testiranja za srednji uzrast psa

Iteracija	Referentna količina za doziranje hrane za srednji uzrast psa (u gramima)	Stvarna dozirana količina hrane (u gramima)	Referentna količina za doziranje vode (u mililitrima)	Stvarna dozirana količina vode (u mililitrima)
1	40	40,22	120	120,56
2	40	41,12	120	120,84
3	40	40,36	120	121,63
4	40	42,56	120	122,01
5	40	43,98	120	120,24
6	40	40,48	120	121,32
7	40	40,77	120	121,68
8	40	41,65	120	120,87
9	40	42,09	120	121,32
10	40	40,02	120	120,74

Grafički prikaz rezultata testiranja za srednji uzrast psa nalaze se na slici 4.2. u nastavku. Na x osi nalazi se broj iteracija, a na y osi su referentna količina hrane, količina dozirane hrane, referentna količina vode i količina dozirane vode.



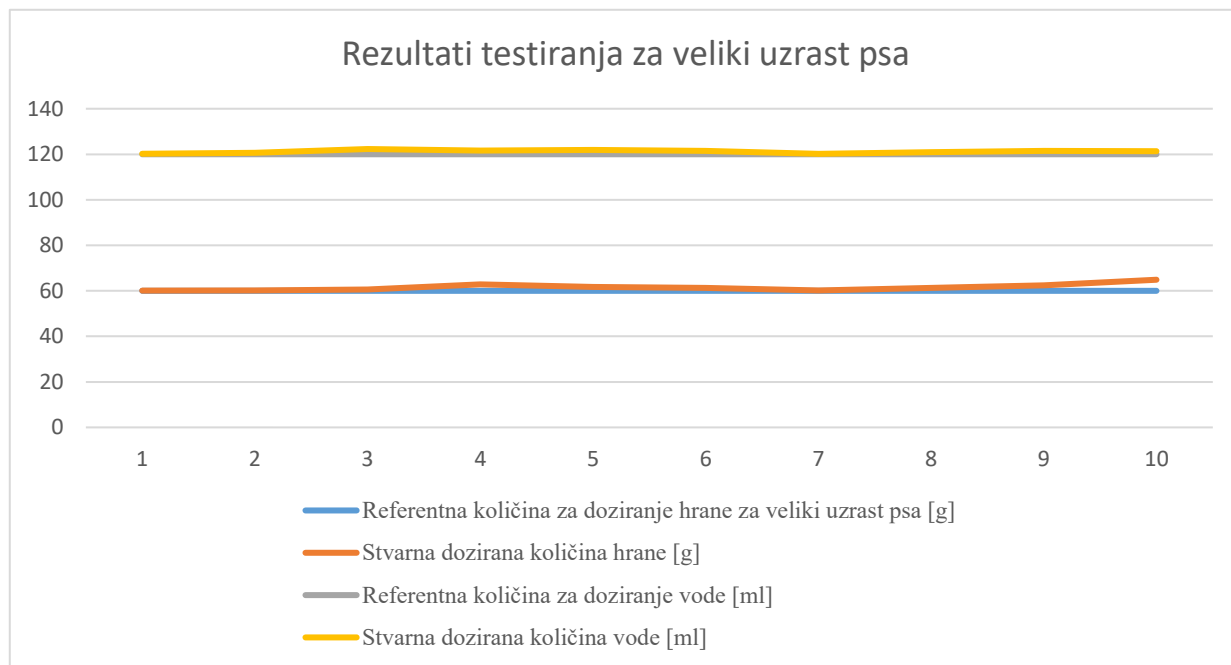
Sl. 4.2. Grafički prikaz rezultata testiranja za srednji uzrast psa

Rezultati testiranja za odabrani veliki uzrast psa koji su se provodili u deset iteracija prikazani su u nastavku u tablici 4.3.

Tab. 4.3. Rezultati testiranja za veliki uzrast psa

Iteracija	Referentna količina za doziranje hrane za veliki uzrast psa (u gramima)	Stvarna dozirana količina hrane (u gramima)	Referentna količina za doziranje vode (u mililitrima)	Stvarna dozirana količina vode (u mililitrima)
1	60	60,10	120	120,21
2	60	60,23	120	120,58
3	60	60,55	120	122,31
4	60	62,88	120	121,59
5	60	61,76	120	121,84
6	60	61,23	120	121,52
7	60	60,15	120	120,19
8	60	61,28	120	120,89
9	60	62,35	120	121,55
10	60	64,88	120	121,32

Grafički prikaz rezultata testiranja za veliki uzrast psa nalaze se na slici 4.3. u nastavku. Na x osi nalazi se broj iteracija, a na y osi su referentna količina hrane, količina dozirane hrane, referentna količina vode i količina dozirane vode.



**Sl. 4.3.** Grafički prikaz rezultata testiranja za veliki uzrast psa

Kao što se može primijetiti, u svim doziranjima postoje mala odstupanja u odnosu na referentne količine. Vage su vrlo precizne te imaju odstupanje od pola grama u odnosu na stvarnu težinu. Razlog odstupanja je u dizajnu rotirajućih drvenih cilindara za doziranje hrane. Kada hrana upada iz spremnika u cilindar, ne upadne uvijek ista količina hrane u jedan utor cilindra. Zbog toga, nekada se dogodi da je doziranje hrane u posudici vrlo precizno, a nekad, ukoliko upadne više hrane od očekivanog, se dogodi da se u posudici dozira prosječno dva do tri grama hrane više od očekivanog. Vodena pumpa radi nešto preciznije, a radi na način da kada je potrebno doliti razliku vode u posudici, vodena pumpa se uključi na jednu sekundu i odmah se isključi. Na ovaj način se postigla visoka preciznost dolijevanja vode, ali i spriječilo prelijevanje vode iz posudice, što je vrlo bitno.



## **5. Zaključak**

Uspješnom realizacijom ovog projektnog zadatka napravljena je automatska hranilica za pse. Riječ je o vrlo korisnom proizvodu, ali i prilagodljivom s obzirom da korisnici mogu odabrati uzrast psa kojeg imaju. Korištenjem automatske hranilice korisnici ne moraju brinuti ukoliko duži vremenski period moraju izbivati od kuće. Njihovi kućni ljubimci imat će osiguranu hranu, slatkiše i vodu cijeli dan. Postupkom testiranja utvrđeno je da automatska hranilica ima nedostatak, a to je manja preciznost kod doziranja hrane i vode. Razlog tome je dizajn rotirajućih drvenih cilindara koji će nekada izbaciti manje, a nekada više hrane kod doziranja u posudicu. Potencijalni problemi koji su se pojavili prilikom izrade automatske hranilice, kao što su problem napajanja svih komponenti, problem spajanja glavnog ožičenja, riješeni su pomoću ranije opisanog adaptera napajanja kojim su sve komponentne dobile potrebno napajanje. Isto tako, glavno ožičenje skriveno je unutar same automatske hranilice i na siguran način spojeno, kako bi korisnik prema van vidio samo gumbe i LCD zaslon.

## 6. Literatura

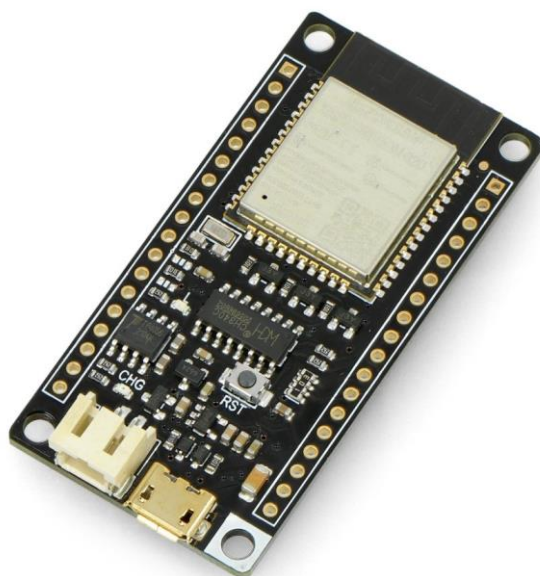
- [1] *Xiaomi Smart Pet Food Feeder*, <https://mi.hr/xiaomi-smart-pet-food-feeder.html>, pristup 14.2.2024.
- [2] *esp32 mikroupravljač*, <https://botland.store/arduino-compatible-boards-dfrobot/9153-dfrobot-firebeetle-esp32-iot-wi-fi-bluetooth-6959420912155.html>, pristup 16.2.2024.
- [3] *NEMA17 koračni motor*, <https://botland.store/stepper-motors/3615-stepper-motor-jk42hs48-2504-200-steps-31v25a047nm-5904422359379.html>, pristup 16.2.2024.
- [4] *A4988 upravljač koračnih motora*, <https://botland.store/stepper-motor-drivers/2660-stepper-motor-driver-a4988-reprap-red-5904422359201.html>, pristup 16.2.2024.
- [5] *Vaga kapaciteta od jednog kilograma*, <https://soldered.com/hr/proizvod/load-cell-senzor-1kg/>, pristup 16.2.2024.
- [6] *HX711 upravljač vage*, <https://soldered.com/hr/proizvod/ploca-s-load-cell-pojacalom-hx711/>, pristup 16.2.2024.
- [7] *Mini pumpa za vodu*, <https://soldered.com/hr/proizvod/mini-pumpa-za-vodu/>, pristup 16.2.2024.
- [8] *LCD zaslon*, <https://botland.store/alphanumeric-and-graphic-displays/19732-lcd-display-2x16-characters-blue-with-connectors-justpi-5903351243131.html>, pristup 16.2.2024.
- [9] *Gumb*, <https://soldered.com/hr/proizvod/tipkalo-pushbutton-12mm/>, pristup 16.2.2024.
- [10] *NPN 2N2222A tranzistor*, <https://soldered.com/hr/proizvod/npn-tranzistor/>, pristup 16.2.2024.

## 7. Prilozi i dodaci

### 7.1 Korištene komponente

#### 1. esp32 mikroupravljač

esp32 je mali, ali moćni mikroupravljač [2]. Integrira dvojezgreni ESP-WROOM-32 modul koji podržava komunikaciju u mikroupravljaču, te WiFi i Bluetooth modove. Namijenjen je za razne projekte internata objekata (engl. *IoT – Internet of Things*), a može biti napajan putem USB-a ili putem baterije. Izgled opisane komponente nalazi se na slici 7.1. u nastavku.



**Sl. 7.1.** esp32 mikroupravljač

#### 2. NEMA17 koračni motor

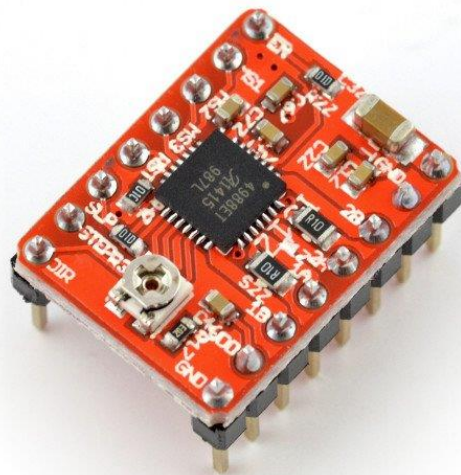
NEMA17 je vrlo pristupačni bipolarni motor, koji ima jednostavan način spajanja na četiri žice [3]. Okretna rezolucija mu je 200 koraka, a zbog svojeg velikog okretnog momenta, ima vrlo raširenu upotrebu na tržištu. Izgled opisane komponente nalazi se na slici 7.2. u nastavku.



**Sl. 7.2.** NEMA17 koračni motor

### 3. A4988 upravljač koračnog motora

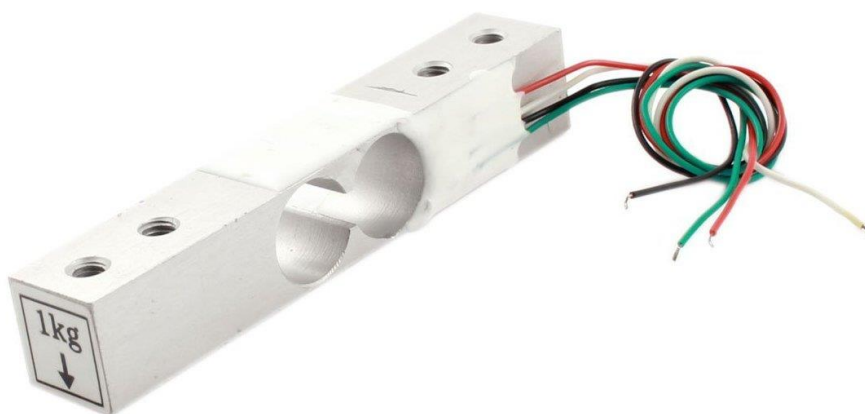
Upravljač koračnih motora A4988 [4] vrlo je praktičan i jednostavan za korištenje. Nudi mogućnost jednostavnog i efikasnog upravljanja koračnim motorom, kao što je NEMA17 [2]. Izgled opisane komponente nalazi se na slici 7.3. u nastavku.



**Sl. 7.3.** A4988 upravljač koračnih motora

#### 4. Vaga kapaciteta od jednog kilograma

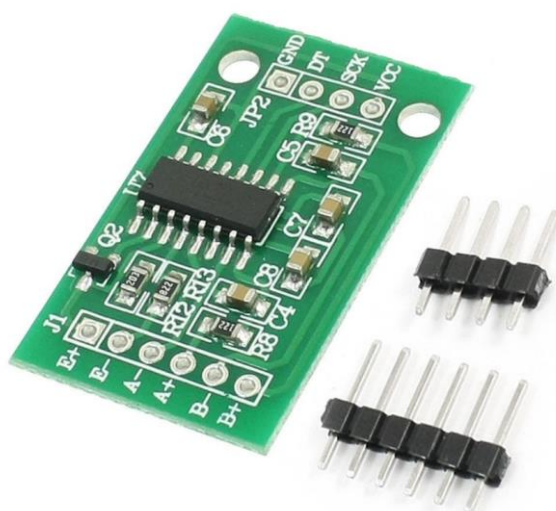
Vaga koja na sebi ima navoje za četiri različita šarafa [5], što pojednostavljuje način montiranja posude za mjerenje, odnosno posudice za hranu i vodu. U sebi ima četiri različita otpornika, te je vrlo precizna kod mjerenja – ima odstupanje od pola grama u odnosu na stvarnu težinu. Izgled opisane komponente nalazi se na slici 7.4. u nastavku.



**Sl. 7.4.** Vaga kapaciteta od jednog kilograma

#### 5. HX711 upravljač vage

Upravljač vage [6] koji nudi visoku preciznost kod očitavanja mjerenja s vage [5]. Očitavanje s vage prenosi digitalnim putem na mikroupravljač, te pretvara očitanu vrijednost što olakšava korištenje. Vrlo je jednostavno spaja, a s vagom se spaja putem četiri žice. Izgled opisane komponente nalazi se na slici 7.5. u nastavku.



**Sl. 7.5.** HX711 upravljač vage

#### 6. Mini pumpa za vodu

Pumpa koja je vrlo jednostavna za korištenje [7]. Na izlaz pumpe se spaja crijevo. Radi na naponu od 3 do 6 volta, a iako je vrlo mala, ima protok vode od 120 litara na sat. Izgled opisane komponente nalazi se na slici 7.6. u nastavku.



**Sl. 7.6.** Mini pumpa za vodu

## 7. LCD zaslon

Alfanumerički zaslon [8] koji prezentira vrijednosti u dva retka, po 16 znakova u svakom retku. Radi na naponu od 5 volta. Pozadina je plave boje, a zbog jednostavnosti korištenja i lakog postavljanja, vrlo je široke upotrebe. Izgled opisane komponente nalazi se na slici 7.7. u nastavku.



**Sl. 7.7.** LCD zaslon

## 8. Gumb

Gumb [9] malih dimenzija koji ima plastični poklopac (proširenje) što omogućuje njegovu jednostavniju upotrebu. Izgled opisane komponente nalazi se na slici 7.8. u nastavku.



**Sl. 7.8.** Gumb s plastičnim poklopcem

## 9. NPN 2N2222A tranzistor

Sveprisutan i vrlo koristan tranzistor [10] koji se u ovom projektu koristio kao sklopka za upravljanje vodenom pumpom. Osim toga, može biti korišten i za pojačavanje struje. Izgled opisane komponente nalazi se na slici 7.9. u nastavku.



**Sl. 7.9.** NPN 2N2222A tranzistor



## 7.2 Programski kod

```
#include <Wire.h>

#include <TimeLib.h>

#include <LiquidCrystal_I2C.h>

#include "HX711.h"


// DOUT, SCK scale for food -> 14, 12
HX711 scaleFood(14, 12);

// DOUT, SCK scale for water -> 25, 26
HX711 scaleWater(25, 26);


// LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);


// calibration factor for 1kg scale for food is -1024, -896
float calibrationFactorScaleFood = -1208;
float weightFood;
float ouncesFood;

// calibration factor for 1kg scale for water is -1208
float calibrationFactorScaleWater = -1209;
float weightWater;
float ouncesWater;


// switch for the water pump
const int waterPumpSwitch = 23;


// food stepper motor
const int stepperFoodEnablePin = 19;
const int stepperFoodDirPin = 13;
const int stepperFoodStepPin = 5;
```

```

// cookies stepper motor
const int stepperCookiesEnablePin = 16;
const int stepperCookiesDirPin = 18;
const int stepperCookiesStepPin = 17;

// buttons
const int buttonUpPin = 2;
const int buttonDownPin = 3;
const int buttonOKPin = 4;

// variables for setting time and dog size
int setParametersState = 0; // 0: Hours, 1: Minutes, 2: Seconds, 3: Dog Size, 4: Done
int enteredHours = 0;
int enteredMinutes = 0;
int enteredSeconds = 0;
int selectedDogSize = 0; // 0: Small Dog, 1: Medium Dog, 2: Big Dog

// food amounts for dog sizes (in grams)
int foodAmountSmallDog = 20;
int foodAmountMediumDog = 40;
int foodAmountBigDog = 60;

// water amount for the bowl (bowl has capacity of 140ml)
int waterMaxAmount = 120;

unsigned long startTime;
bool waterPouredAfterAnHour = false;

void setup() {

```

```
Serial.begin(9600);
```

```
scaleFood.set_scale(calibrationFactorScaleFood);
```

```
scaleFood.tare();
```

```
scaleWater.set_scale(calibrationFactorScaleWater);
```

```
scaleWater.tare();
```

```
Serial.println("Readings: ");
```

```
lcd.begin();
```

```
lcd.print("Automatic Dog");
```

```
lcd.setCursor(0, 1);
```

```
lcd.print(" Feeder");
```

```
delay(5000);
```

```
lcd.clear();
```

```
lcd.print("Enter time");
```

```
delay(3000);
```

```
// turn off both drivers for stepper motors and water pump when starting
```

```
digitalWrite(stepperFoodEnablePin, HIGH);
```

```
digitalWrite(stepperCookiesEnablePin, HIGH);
```

```
digitalWrite(waterPumpSwitch, LOW);
```

```
pinMode(waterPumpSwitch, OUTPUT);
```

```
pinMode(stepperFoodEnablePin, OUTPUT);
```

```
pinMode(stepperFoodDirPin, OUTPUT);
```

```
pinMode(stepperFoodStepPin, OUTPUT);
```

```
pinMode(stepperCookiesEnablePin, OUTPUT);
```

```
pinMode(stepperCookiesDirPin, OUTPUT);
```

```

pinMode(stepperCookiesStepPin, OUTPUT);
pinMode(buttonUpPin, INPUT_PULLUP);
pinMode(buttonDownPin, INPUT_PULLUP);
pinMode(buttonOKPin, INPUT_PULLUP);
}

void loop() {
    // turn off both drivers for stepper motors and water pump when starting
    digitalWrite(stepperFoodEnablePin, HIGH);
    digitalWrite(stepperCookiesEnablePin, HIGH);
    digitalWrite(waterPumpSwitch, LOW);

    handleButtons();
    printOnLCD();
    delay(500);

    if (setParametersState == 4) {
        updateTime();
    }

    // scale for food
    Serial.println("Reading: ");
    weightFood = scaleFood.get_units(), 10;
    if (weightFood < 0) {
        weightFood = 0.00;
    }
    ouncesFood = weightFood * 0.035274;
    Serial.print(weightFood);
    Serial.println(" g of food");
}

```

```

// scale for water
Serial.println("Reading: ");
weightWater = scaleWater.get_units(), 10;
if (weightWater < 0) {
    weightWater = 0.00;
}
ouncesWater = weightWater * 0.035274;
Serial.print(weightWater);
Serial.println(" ml of water");

// dispensing food and cookies, and pouring water in morning
if(hour() == 7 && minute() <= 2 && second() <= 59) {
    if(selectedDogSize == 0 && (weightFood < foodAmountSmallDog)) { // small dog
        dispenseFood();
        dispenseCookies();
    }
    if(selectedDogSize == 1 && (weightFood < foodAmountMediumDog)) { // medium dog
        dispenseFood();
        dispenseCookies();
    }
    if(selectedDogSize == 2 && (weightFood < foodAmountBigDog)) { // big dog
        dispenseFood();
        dispenseCookies();
    }
    if(weightWater < waterMaxAmount) {
        pourWater();
    }
}

// dispensing food and cookies, and pouring water in the evening

```

```

if(hour() == 19 && minute() <= 2 && second() <= 59) {
    if(selectedDogSize == 0 && (weightFood < foodAmountSmallDog)) { // small dog
        dispenseFood();
        dispenseCookies();
    }
    if(selectedDogSize == 1 && (weightFood < foodAmountMediumDog)) { // medium dog
        dispenseFood();
        dispenseCookies();
    }
    if(selectedDogSize == 2 && (weightFood < foodAmountBigDog)) { // big dog
        dispenseFood();
        dispenseCookies();
    }
    if(weightWater < waterMaxAmount) {
        pourWater();
    }
}

// checking every hour if there is enough water in the bowl
if (!waterPouredAfterAnHour) {
    if(weightWater < waterMaxAmount) {
        pourWater();
        waterPouredAfterAnHour = true;
        startTime = millis();
    }
}

// 3600000 milliseconds equals to 1 hour
// check every hour if there is enough water
if (waterPouredAfterAnHour && millis() - startTime >= 3600000) {

```

```

    waterPouredAfterAnHour = false;
}
}

void dispenseFood() {
    digitalWrite(stepperFoodEnablePin, LOW);
    digitalWrite(stepperFoodDirPin, HIGH);

    // moving stepper motor in four steps
    // each step makes 90 degrees rotation
    for(int x = 0; x < 50; x++){
        digitalWrite(stepperFoodStepPin, HIGH);
        delayMicroseconds(12000);
        digitalWrite(stepperFoodStepPin, LOW);
        delayMicroseconds(12000);
    }
    delay(2000);

    digitalWrite(stepperFoodDirPin, LOW);
    digitalWrite(stepperFoodEnablePin, HIGH);
}

void dispenseCookies() {
    digitalWrite(stepperCookiesEnablePin, LOW);
    digitalWrite(stepperCookiesDirPin, HIGH);

    for(int x = 0; x < 50; x++){
        digitalWrite(stepperCookiesStepPin, HIGH);
        delayMicroseconds(12000);
        digitalWrite(stepperCookiesStepPin, LOW);
    }
}

```

```

    delayMicroseconds(12000);
}
delay(2000);

digitalWrite(stepperCookiesDirPin, LOW);
digitalWrite(stepperCookiesEnablePin, HIGH);
}

void pourWater() {
    digitalWrite(waterPumpSwitch, HIGH);
    delay(1000);
    digitalWrite(waterPumpSwitch, LOW);
    delay(1000);
}

void handleButtons() {
    if (digitalRead(buttonUpPin) == LOW) {
        switch (setParametersState) {
            case 0:
                if (enteredHours < 23) enteredHours++;
                else enteredHours = 0;
                break;
            case 1:
                if (enteredMinutes < 59) enteredMinutes++;
                else enteredMinutes = 0;
                break;
            case 2:
                if (enteredSeconds < 59) enteredSeconds++;
                else enteredSeconds = 0;
                break;

```



```

case 3:
    if (selectedDogSize < 2) selectedDogSize++;
    else selectedDogSize = 0;
    break;
}
}

```

```

if (digitalRead(buttonDownPin) == LOW) {
    switch (setParametersState) {
        case 0:
            if (enteredHours > 0) enteredHours--;
            else enteredHours = 23;
            break;
        case 1:
            if (enteredMinutes > 0) enteredMinutes--;
            else enteredMinutes = 59;
            break;
        case 2:
            if (enteredSeconds > 0) enteredSeconds--;
            else enteredSeconds = 59;
            break;
        case 3:
            if (selectedDogSize > 0) selectedDogSize--;
            else selectedDogSize = 2;
            break;
    }
}

```

```

if (digitalRead(buttonOKPin) == LOW) {
    updateState();
}

```

```
}  
}
```

```
void updateState() {  
    setParametersState++;  
  
    if (setParametersState > 3) {  
        setParametersState = 4; // Set to 4 to indicate that time setting and dog size selection are done  
        setTime(enteredHours, enteredMinutes, enteredSeconds, 1, 1, 2024);  
    }  
}
```

```
void printOnLCD() {  
    lcd.clear();  
  
    if (setParametersState <= 2) {  
        lcd.print("Set ");  
        if (setParametersState == 0) {  
            lcd.print("hours: ");  
            lcd.print(enteredHours);  
        } else if (setParametersState == 1) {  
            lcd.print("minutes: ");  
            lcd.print(enteredMinutes);  
        } else if (setParametersState == 2) {  
            lcd.print("seconds: ");  
            lcd.print(enteredSeconds);  
        }  
  
    } else if (setParametersState == 3) {  
        lcd.print("Select dog size:");  
    }
```

```

lcd.setCursor(0, 1);
switch (selectedDogSize) {
  case 0:
    lcd.print("Small (1-5kg)");
    break;
  case 1:
    lcd.print("Medium (5-15kg)");
    break;
  case 2:
    lcd.print("Big (15-30kg)");
    break;
}

} else {
  lcd.print(hour());
  printDigits(minute());
  printDigits(second());
  lcd.setCursor(0, 1);
  switch (selectedDogSize) {
    case 0:
      lcd.print("Small dog");
      break;
    case 1:
      lcd.print("Medium dog");
      break;
    case 2:
      lcd.print("Big dog");
      break;
  }
}
}

```

```
}
```

```
void printDigits(int digits) {
```

```
    lcd.print(":");
```

```
    if (digits < 10)
```

```
        lcd.print('0');
```

```
    lcd.print(digits);
```

```
}
```

```
void updateTime() {
```

```
    adjustTime(0.5);
```

```
}
```