Maketa sustava za gađanje meta u pokretu

Dudjak, Dominik

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij: Automatika

MAKETA SUSTAVA ZA GAĐANJE META U POKRETU

Završni rad

Dominik Dudjak

Osijek, 2023.

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

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

Osijek, 21.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Dominik Dudjak
Studij, smjer:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. Pristupnika, godina upisa:	A 4455, 20.07.2017.
OIB Pristupnika:	94418765936
Mentor:	izv. prof. dr. sc. Tomislav Keser
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Tomislav Galba
Član Povjerenstva 1:	izv. prof. dr. sc. Tomislav Keser
Član Povjerenstva 2:	izv. prof. dr. sc. Ivan Aleksi
Naslov završnog rada:	Maketa sustava za gađanje meta u pokretu
Znanstvena grana završnog rada:	**Automatizacija i robotika (zn. polje elektrotehnika)**
Zadatak završnog rada	Projektirati, izraditi i testirati maketu sustava za gađanje mete u pokretu. Mete mogu biti raznih veličina a bodovanje pogodaka se provodi na način da najveća meta donosi najmanje bodova, odnosno što je meta manja pogodak donosi proporcionalno više bodova. Samo proces gađanja i upravljanje gađanjem realizirati ili mobilnom platformom (Android ili iOS) ili nekakvim drugim računalnim sustavom. (Rezervirano za: Dominik Dudjak)
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	21.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 13.10.2023.

Ime i prezime studenta:	Dominik Dudjak
Studij:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. studenta, godina upisa:	A 4455, 20.07.2017.
Turnitin podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Maketa sustava za gađanje meta u pokretu**

izrađen pod vodstvom mentora izv. prof. dr. sc. Tomislav Keser

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 2
1.1. Zadatak i struktura rada 3
2. SUSTAV ZA GAĐANJE POKTRETNE METE 4
2.1. Teorijski osvrt na problematiku zadatka 6
2.2. Prijedlog sklopovskog rješenja 7
2.3. Prijedlog programskog rješenja 8
3. REALIZACIJA MAKETE SUSTAVA ZA GAĐANJE META U POKRETU 10
3.1. Korišteni alati i programska okruženja 10
3.3. Realizacija programskog okruženja 23
4. TESTIRANJE I REZULTATI 26
4.1. Metodologija testiranja 28
4.2. Testiranje 26
5. ZAKLJUČAK 30
LITERATURA 31
SAŽETAK 32
SUMMARY 33
ŽIVOTOPIS 34
PRILOZI I DODATCI 35

1. UVOD

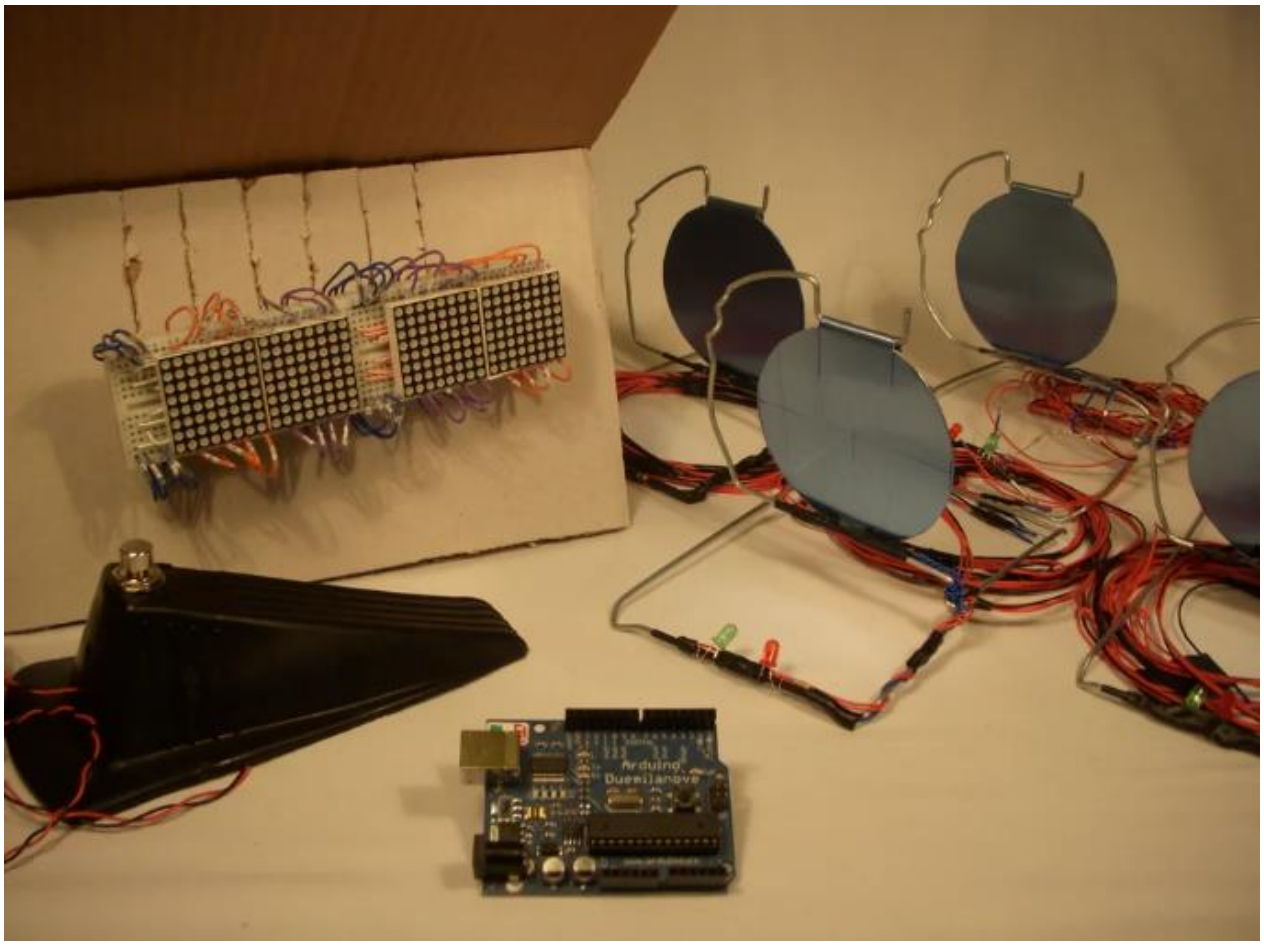
Završni rad obuhvaća segmente osmišljavanja i realizacije "Makete sustava za gađanje pokretne mete". Unutar navedenih poglavlja predstavljena je cjelokupna procedura dolaska do konačnog cilja. Kao što je već i navedeno cilj je realizacija funkcionalnog sustava za gađanje pokretne mete koje je podijeljeno na dvije različite problematike. Maketa sustava za gađanje usko je povezana i osmišljena na temelju primjera iz stvarnoga života. Paralelu je moguće povući sa sustavom za gađanje koje se koristi u vojne svrhe kao što su protu obrambeni sustavi, sustavi za napad i sustavi za detekciju objekta. U navedenim slučajevima sustavi su realizirani automatski ili uz pomoć ljudskoga faktora. Uzimajući za primjer obrambeni sustav koji je u većini slučajeva automatiziran jer on pomoću senzora za pokret ima mogućnost praćenja i detekcije objekta koji se kreće ili približava, dok je s druge strane prisutan sustav za napad kojemu je u većini situacija potreban ljudski faktor kako bih upravljao njime ručno ili uz pomoć aplikacije i programa u današnje moderno vrijeme. Primjena sustava za gađanje je također prisutna i u zabavne svrhe što se može poistovjetiti s društvenim igrama ili timskim sportovima kao što je airsoft jer kroz njega svatko može imati doticaja sa sustavima za gađanje kao što su vojni. Uz spomenuti sustav za gađanje kao drugi dio sustava postavlja se pokretna meta. Navedene pokretne mete imaju vrlo široku uporabu te okružuju ljudsku populaciju putem medija ili u svakodnevnome životu. Pokretne mete najveću uporabu imaju u sportovima kao što je streljaštvo[1] gdje je sportašima u interesu da putem hitca pogode metu te ostvare bodove i ujedno rezultat. Specifičnost navedenih stvari je u tome što se one mogu automatizirati što zapravo i postavlja temelj za osmišljavanje makete. Motiv pri rješavanju zadatka je na primjeru postojećih rješenja, dobiti maksimalno koristan i funkcionalan rad za koji postoji opcija da se koristi u različite svrhe i da što više ljudi ima pristup njemu.

1.1. Zadatak i struktura rada

Cilj same izrade je kroz unaprijedno planiranje, organizaciju i odabir traženih komponenata dobiti funkcionalnu maketu. Navedeni zadatak je podijeljen na dva djela, točnije unutar svakoga od navedenih poglavlja prikazuje se posebno izrada sustava za gađanje, a zasebno pokretne mete. Cilj prvoga djela zadatka je kreirati strukturu koja će imati mogućnost rotacije. Osim mogućnosti rotacije potrebno je osmisliti način komunikacije i povezivanja mobilne aplikacije sa sustavom za gađanje. Drugi dio zadatka prožima osmišljavanje pokretne mete. Mete moraju biti konstantno u pokretu i moraju imati mogućnost očitovanja udarca kako bi mogle ispisati vrijednost. Rad je napisan kroz poglavlja koja prikazuju osmišljavanje makete, odabir komponenata i realizaciju uz pomoć adekvatnih alata i programskih okruženja. Prvo poglavlje završnog rada predstavlja motiv i cilj rješavanja sustava, točnije navodi primjere koji su smjernice za realizaciju zadatka. Unutar drugoga poglavlja prikazan je prijedlog programskog i sklopovskog rješenja sustava. Zapravo se kroz drugo poglavlje dobiva jasnija slika u kojem smjeru će se razvijati maketa, što bi se moglo koristiti za realizaciju i daje blokovske prikaze ideja za rješavanje zadataka. Treće poglavlje uvodi u samu srž jer se navode komponente, programsko okruženja i ono što je unutar drugog poglavlja bila samo ideja se provodi u realizaciju makete i dovodi do konačnog rješenja. Unutar poglavlja prikazana je realizacija programskog i sklopovskog rješenja kroz koje je cilj prikazati način pristupanja, odabira komponenata i pretvaranja ideje u realnost na funkcionalan način. Kao zadnje poglavlje uzima se četvrto u kojemu je prikazan način testiranja makete. Testiranje je provedeno u programskom okruženju te u stvarnome životu kroz ispitivanje stavki koje utječu na funkcionalnost. Zaključak četvrtog poglavlja je uklanjanje svih greški koje se pojavljuju te koje bi eventualno mogle utjecati na rad makete.

2. MAKETA SUSTAVA ZA GAĐANJE META U POKRETU

Prikazom same strukture sustava za gađanje ili iz samoga naziva dana je slika u kojem smjeru će se on razvijati, i koje su asocijacije i poveznice s njime. Kao što je definirano zadatak je realizacija makete sustava za gađanje. Radi lakšeg pregleda postojećih stanja koja su već realizirana zadatak se može podijeliti na dva zasebna, kako bi se lakše predočila struktura samoga sustava. Sami naziv pokretne mete asocira na već nekoliko primjera koji su realizirani. Glavna i temeljna ideja koja je postojeća, a poslužila je kao ideja za razvoj ovoga rada je pokretna meta koja se koristi u sportu kao što je streljaštvo[1]. U današnje vrijeme većina stvari koja nas okružuje je automatizirana, baš iz tog razloga za ideju je uzeta pokretna meta iz različitih sportova. Krenuvši od njene strukture može se razlikovati po veličini i obliku. Navedene stavke mogu se povezati na temelju definiranja veličine mete. Kao najbitnija poveznica između makete i postojećeg rješenja pokretne mete je njezina mobilnost. Mogućnost konstantne kretnje pokretne mete daje taj osjećaj zabave i napetosti prilikom uporabe ili natjecanja. Kao zadnja poveznica navodi se najbitnija stavka, a to je ona tko je pobjednik prilikom natjecanja? Taj segment se definira i prikazuje sustavom bodovanja, koji je također u većini slučajeva digitaliziran radi lakšeg praćenja pogodaka. Sličnost između postojećeg rješenja pokretne mete i makete može se istaknuti u segmentima kao što su mobilnost mete, veličina mete i sustav bodovanja. Kao drugi dio unutar podjele makete uzima se sustav gađanja. Sustav gađanja može se realizirati na velik broj načina. Kao prvi primjer i jedna od okosnica ideje za njegovu realizaciju može se istaknuti sustav gađanja temeljen na zabavnim sportovima kao što je "Airsoft"[2]. Unutar navedenog sporta prikazuje se automatizirani sustav gađanja unutar kojega se gađanje vrši pomoću lasera ili gumenih metaka. Osim navedene opcije spominju se i prikazuju različiti primjeri istoga sadržaja kao što je maketa sustava samo što se oni temelje na različitim komponentama.



Slika 2.1. *Primjer pokretne mete[3]* .

Sustav prikazan na slici 2.1 je rad slične problematike kao i sustav koji se realizirao unutar završnoga rada. Sami koncept pokretne mete sa slike koristi se za svrhe vježbanja u sportu kao što je streljaštvo[1] ili u zabavne svrhe. Temelji se na mikroupravljaču i pokazivaču koji prilikom pogotka ispisuje informaciju o pogotku. Sličnost između dva rada je što oba imaju sustav bodovanja koji se ispisuje putem pokazivača.

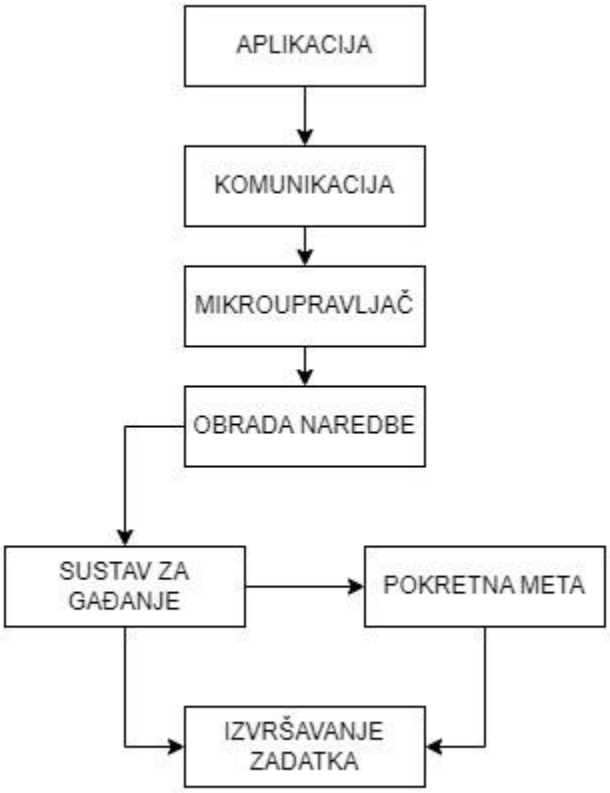
Slika 2.2. *Primjer sustava gađanja[4]* .

Na slici 2.2 prikazan je sustav gađanja vrlo sličan završnome radu. Navedeni sustav temelji se na mikroupravljaču i kameri koja prati pokrete. Sličnost između ova dva rada je ta što oba imaju mogućnost rotacije u svim smjerovima i automatizirani sustav pucanja.

2.1. Teorijski osvrt na problematiku zadatka

Teorijski osvrt prikazuje slijed procesa koje je potrebno uzeti na razmatranje, te korak po korak razrješavanje njihove problematike jer se samim time dolazi do konačnog sustava. Cilj je spajanjem hardverskih i softverskih komponenta unošenje ljudskoga faktora u sami proces. Funkcionalnost se ne može ostvariti ako nije osmišljen način ispunjavanja svakog problema koji se pojavljuje. Prvotno je potrebno omogućavanje komunikacije između mobilne aplikacije i ESP-32, nakon što je omogućen proces komunikacije između navedenih stavki pomoću Bluetooth modula, moguće je obrađivanje informacija i zadataka koji se dobivaju od strane rukovatelja. Kroz omogućen proces komunikacije i zaprimanja informacija ostvaruje se mogućnost upravljanja sustavom za gađanje i ispunjavanja zadataka kao što je ispaljivanje hitca i rotacije u zadanome smjeru. Također unutar djela sustava za gađanje povezanoga s pokretnom metom potrebno je realizirati ispisivanje vrijednosti putem pokazivača ovisno o sustavu bodovanja. Sami koncept sustava bodovanja zasniva se na veličini mete i vrijednostima koje su unutar kôda definirane za svaku metu. Prilikom

pogotka mete dobiva se putem senzora informacija koja je meta točno pogođena , te se ispisuje zadana vrijednost na pokazivač.

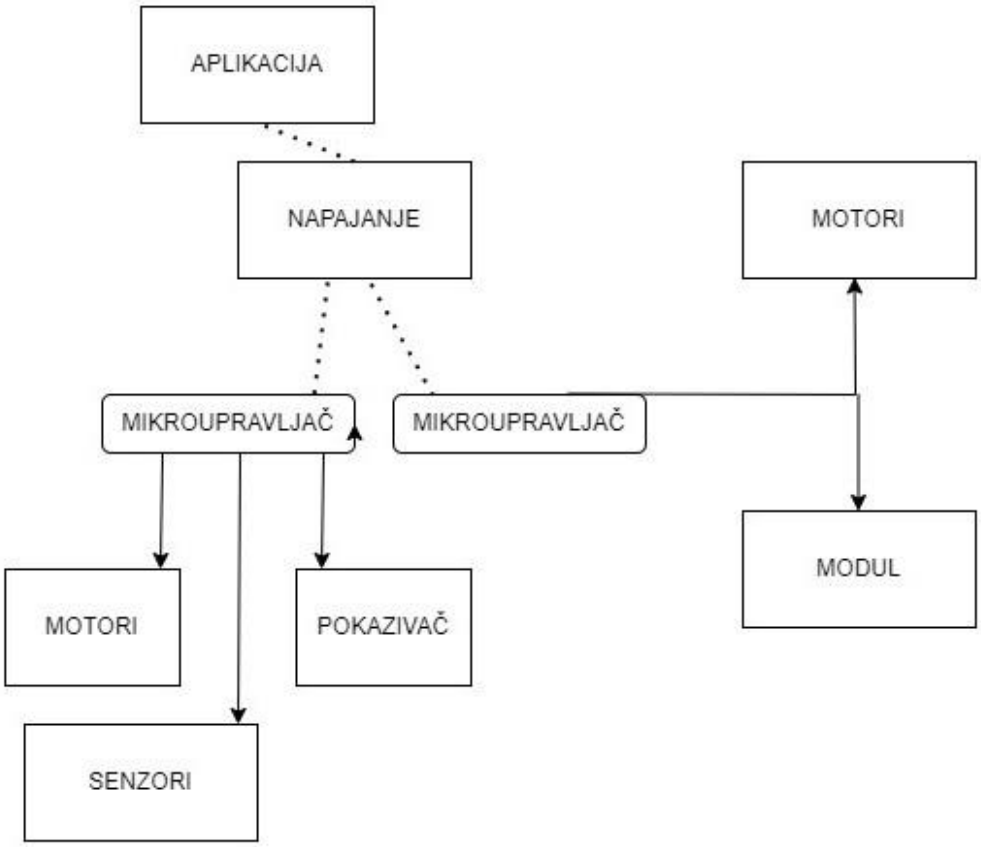


Slika 2.4. *Prikaz teorijskog osvrta na rad makete.*

2.2. Prijedlog sklopovskog rješenja

Realizaciji makete pristupa se na dva zasebna načina. Zamišljeno je da sustav gađanja ima strukturu kao vojni sustav gađanja iz stvarnoga život. Sama rotacija navedene strukture bazira se na servo motorima. Točnije određeni će služiti za rotaciju strukture, a drugi će pomicati nogicu koja ima zadatak guranja metaka iz spremnika. Kao glavni dio sustava za gađanje uzima se mikroupravljač ESP-32 za koji je zamišljeno da su na njega servo motori povezani. Također uz servo motore na izlazu strukture bit će pozicionirani istosmjerni motori koji će obavljati zadatak ispucavanja metaka. Poveznica između dva pristupa rješavanju makete je modul koji omogućava komunikaciju

između mobilne aplikacije i sustava za gađanje. Drugi dio makete također se temelji na ESP-32 na koji je povezan motor koji omogućava kretanje pokretne mete. Za zadnje stavke uzimaju se senzori koji informaciju o pogotku mete šalju na pokazivač koji ispisuje koncept bodovanja mete. Oba djela sustava napapaju se putem zajedničkog napajanja. Na slici 2.1 prikazan je prijedlog sklopovskog rješenja.

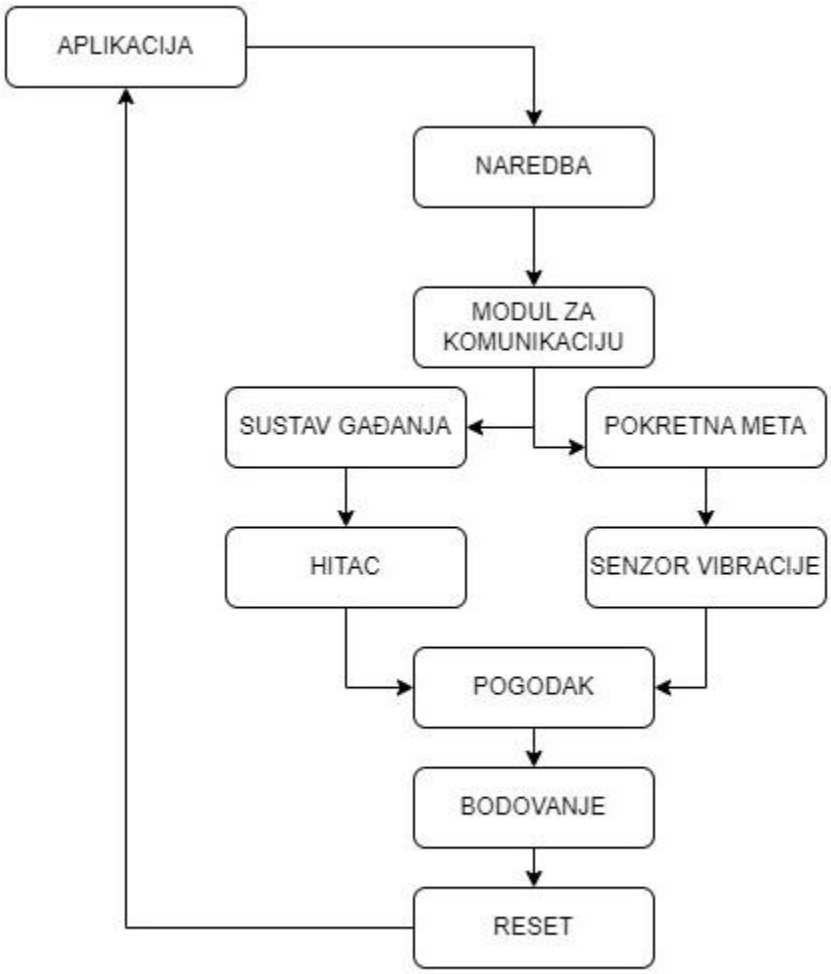


Slika 2.4. *Prijedlog sklopovskog rješenja.*

2.3. Prijedlog programskog rješenja

Upravljanje sustavom za gađanje vrši se putem HC-05 Bluetooth modula. Navedeni modul omogućava komunikaciju između mikroupravljača na koji su povezani servo motori i mobilne aplikacije. Zadatak servo motora je rotiranje strukture sustava za gađanje ili ispucavanje metaka iz spremnika ovisno o informaciji koju zadobije od strane rukovatelja putem mobilne aplikacije. Također unutar prijedloga programskog rješenja prikazana je međusobna ovisnost između senzora

vibracije i zaslona koji su povezani na ESP-32. Senzori vibracije očitavaju silinu udarca prilikom pogotka mete te navedenu vrijednost ispisuju na zaslon putem kojega se vrši bodovanje. Nakon što rukovatelj dođe do ukupnog rezultata od deset bodova rezultat se pomoću tipkala resetira.



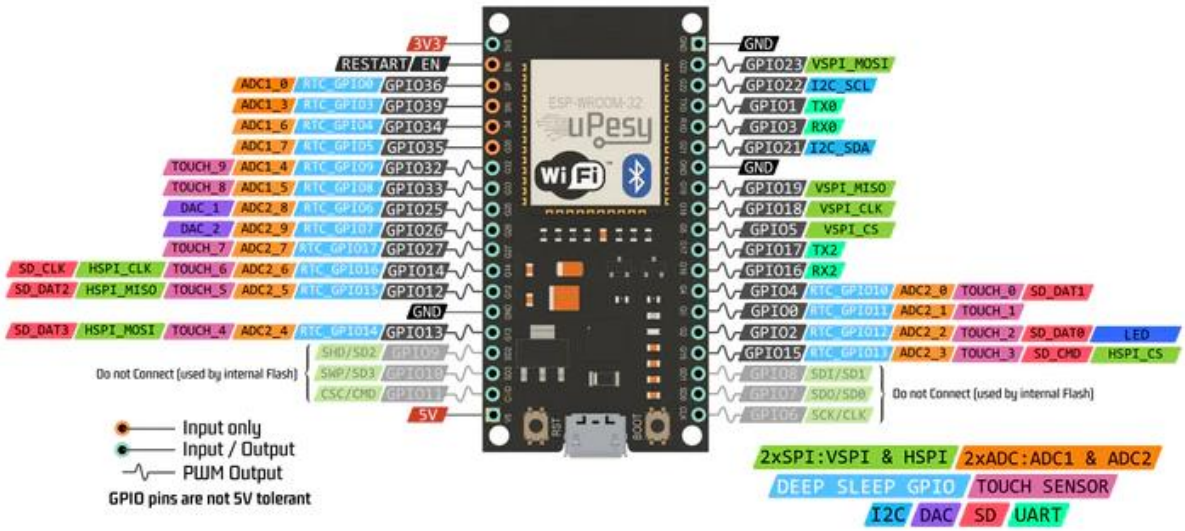
Slika 2.5. *Prijedlog programskog rješenja.*

3. REALIZACIJA MAKETE SUSTAVA ZA GAĐANJE META U POKRETU

Unutar ovoga poglavlja predstavljene su hardverske komponente i programi koji su potrebni za provođenje ove ideje u stvarnost.

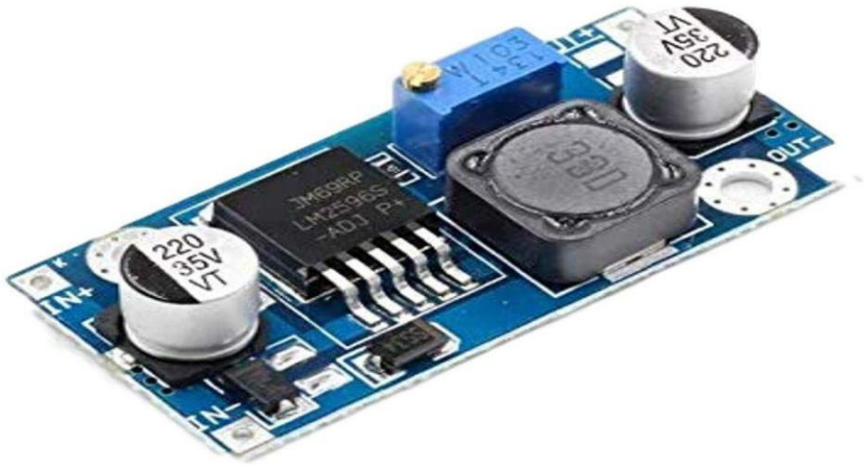
3.1. Korišteni alati i programska okruženja

Za realizaciju makete korišteni su programski jezik C i Arduino za programsko rješenje. Programski jezik C korišten je za logiku rada i pisanje kôda, a Arduino je korišten radi lakšeg implementiranja kôda na mikroupravljač.



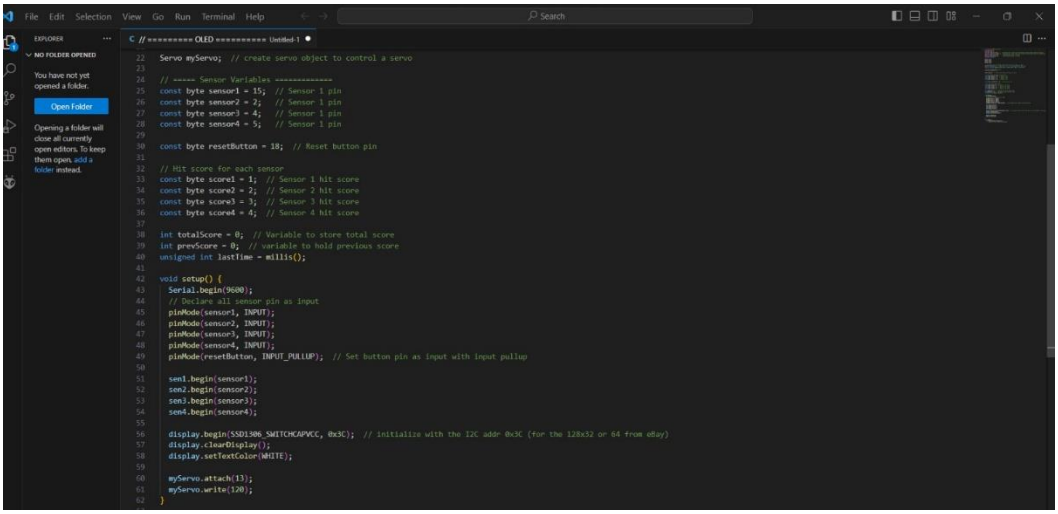
Slika 3.1. *ESP-32 mikroupravljač[5].*

ESP-32 je vrsta mikroupravljača koji dolazi s već unaprijed "ugrađenim" mogućnostima kao što su WIFI i Bluetooth. Osim što ima mogućnosti WIFI-ja i Bluetooth-a dolazi u verziji s već unaprijed "instaliranim" komponentama kao što su pojačala snage, antene, filteri... Samim time jedna od prednosti ESP-32 je što njegova upotreba može stagnirati to jest ona varira o vremenu njegove upotrebe što znači da se može regulirati vrijeme njegove upotrebe, te samim time se dolazi do opcije smanjenja energije koja se troši unutar sustava.



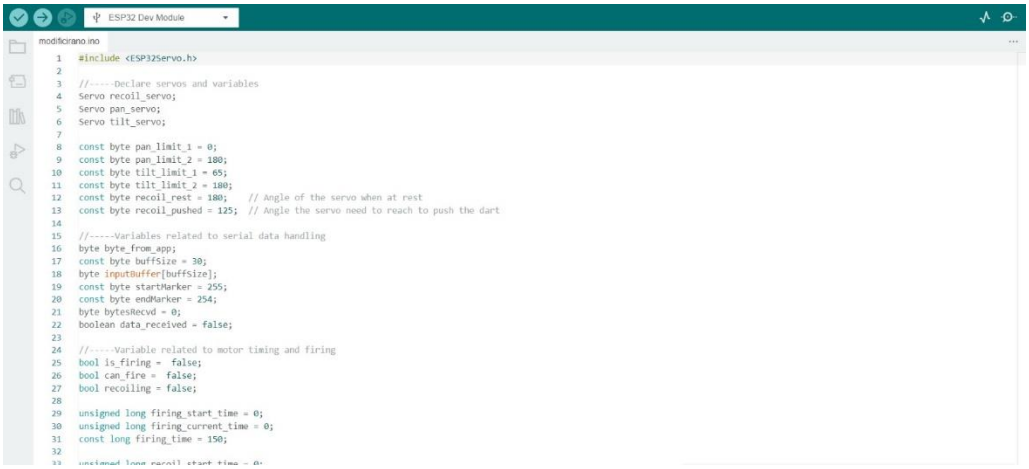
Slika 3.2. *Regulator napona[6]*

Regulator napona je jedna od bitnijih komponenti koje su korištene prilikom izrade makete. Putem nje je napajanje regulirano na traženi izlazni napon koji se regulira putem trimera na pločici.



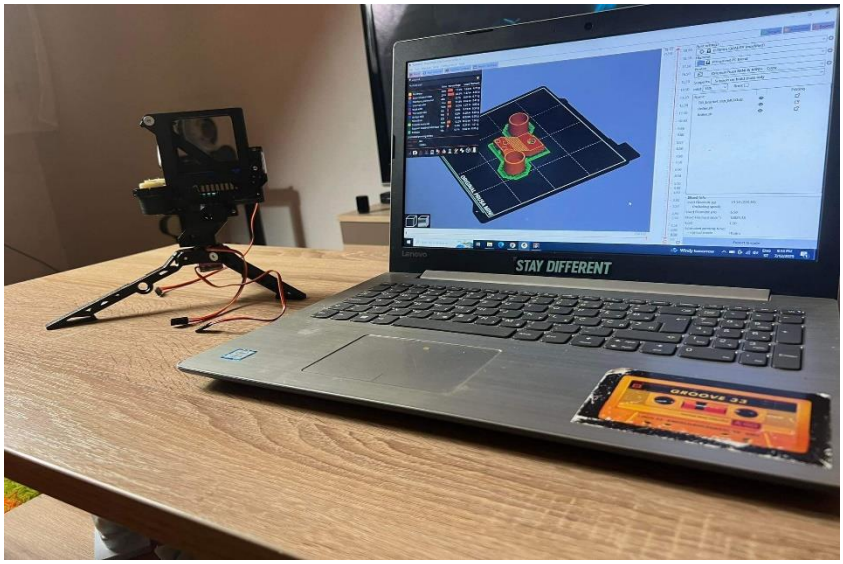
Slika 3.3. *Primjer prikaza programa Visual Code.*

Visual studio[7] služi za izradu softvera točnije aplikacija, računalnih programa i web stranica. Posjeduje sve alate pomoću kojih je moguće izraditi softver od početka do kraja. Omogućuje korisnicima da putem njega pišu i uređuju kôd, te ga da kompajliraju kako bi izradili softver. Napravljen je tako da podržava 36 programskih jezika od kojih su neki već ugrađeni poput C, C++, JavaScript...



Slika 3.4. Primjer *prikaza Arduino IDE sučelja.*

Uz Visual Code korišten je i Arduino IDE. Kôd koji je napisan u Visual Code je uz pomoć Arduino IDE[8] dodatno prepravljen, te uspješno implementiran na korišteni mikroupravljač. Arduino Ide sastoji se od više dijelova koji se nalaze na alatnoj traci a neki od njih su izbornik, konzole, pomoć i dio za odabir datoteke.



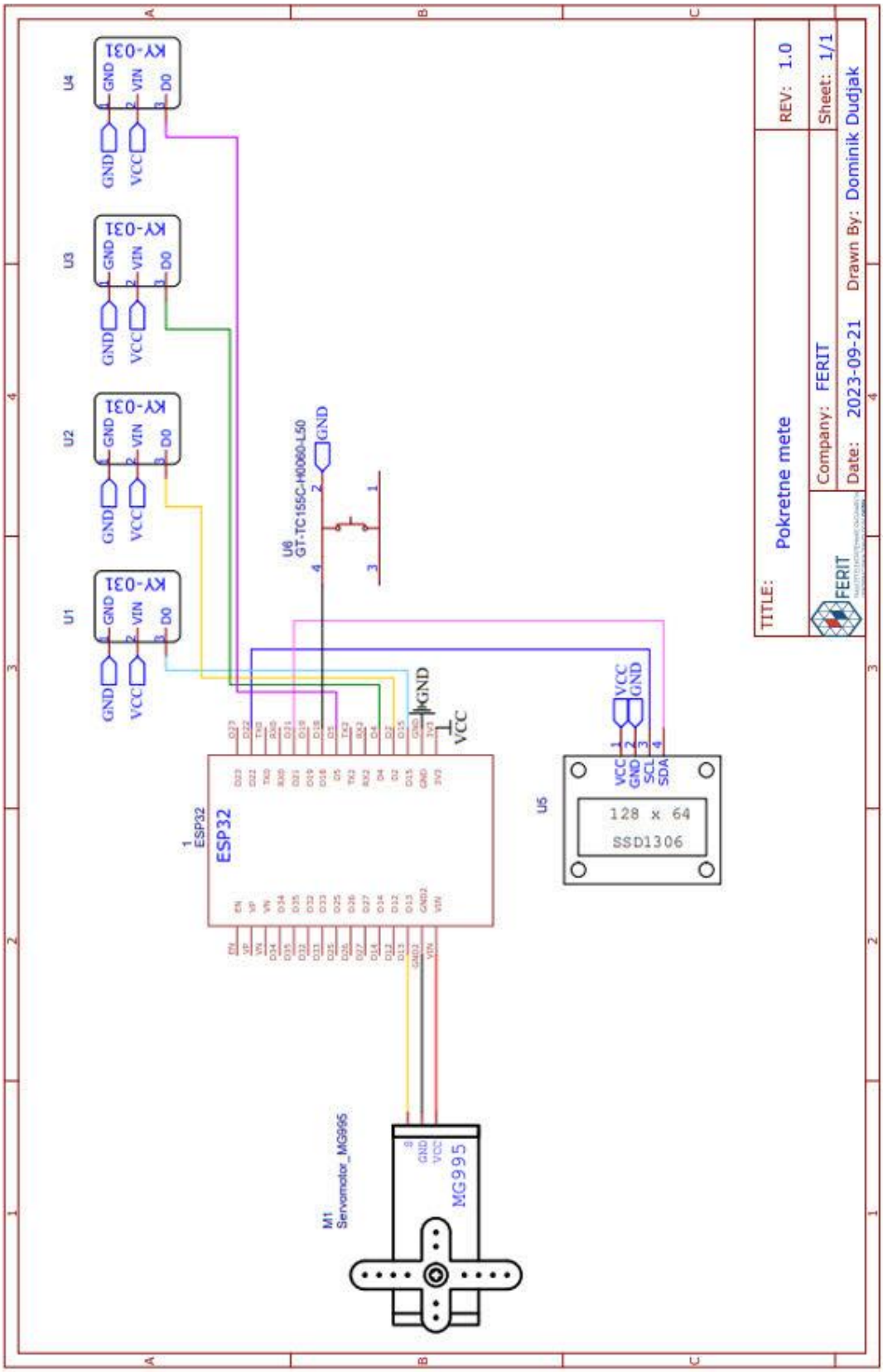
Slika 3.5. *Prikaz strukture sustava za gađanje u programu Prusa Slicer.*

Za izradu strukture sustava za gađanje korišten je program Prusa Slicer[9] za 3D printanje. Navedeni program omogućio je kreiranje modela, te odabir temperature za daljnje printanje. Nakon što je kreiran željeni model strukture sprema se u datoteku kako bi u daljnjim koracima mogao biti isprintan. Nakon kreiranja datoteke potrebno ju je odabrati te dodati unutar navedenoga programa.

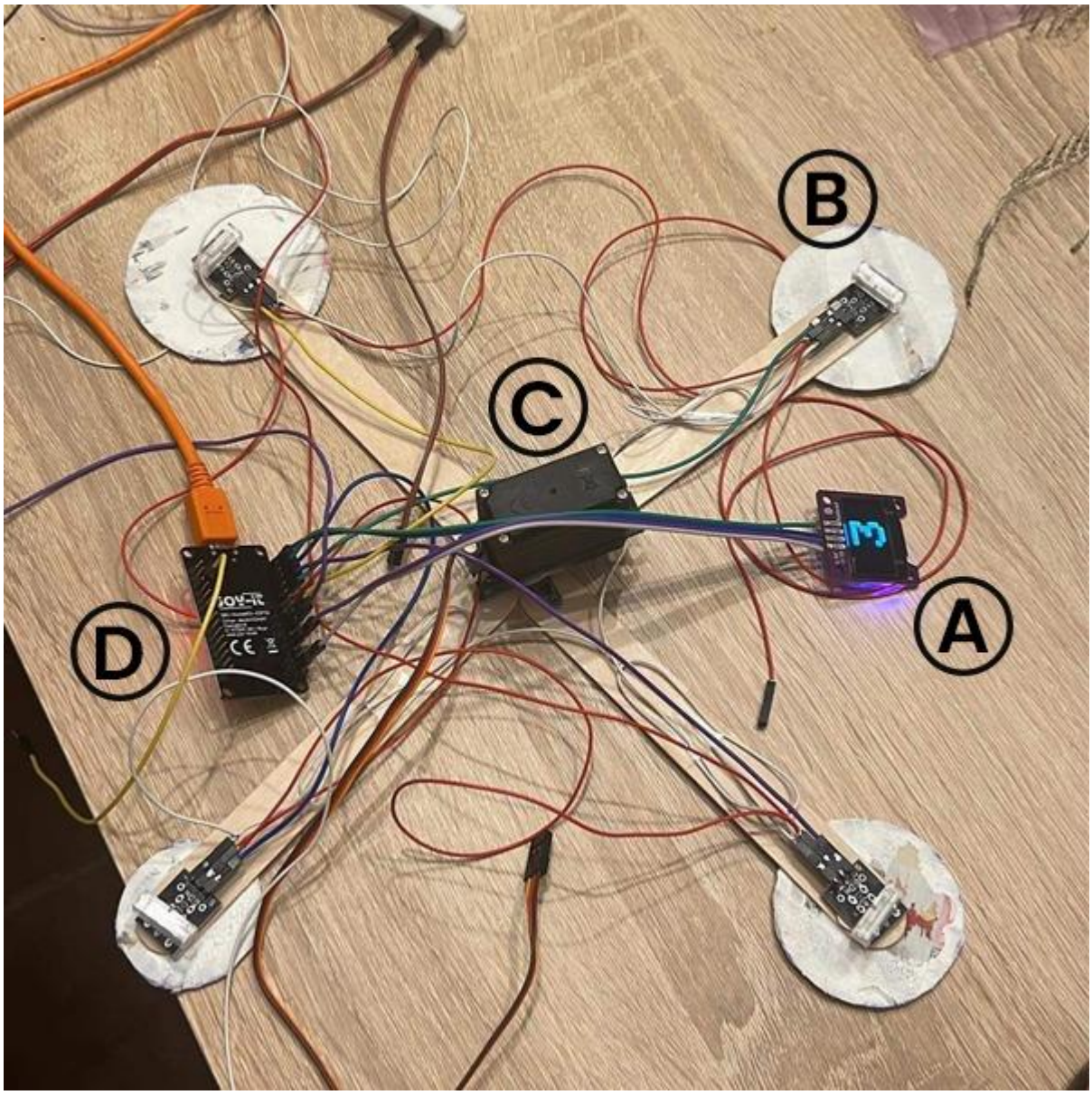
3.2. Realizacija sklopovskog rješenja

Na slici 3.6 prikazana je električna shema pokretne mete. Prilikom izrade mete bilo je potrebno omogućiti da ona bude konstantno u pokretu. Navedena opcija omogućena je tako da je za izradu odabran servo motor MG995 koji ima mogućnost rotacije za 360 stupnjeva. Servo motor MG995 povezan je na ESP-32 na pin D13. Unutar kôda definirana je opcija da navedeni servo motor ima ograničeno kretanje, točnije kako ne bi dolazilo do zapetljavanja žica MG995 se kreće određeni broj sekundi u lijevo a zatim u desno. Samim definiranjem vremena unutar kako bi bila sanirana problematika koja se pojavila, i dalje se ispunjava zadatak konstantnog kretanja pokretne mete. Na ESP-32 povezana su i četiri senzora vibracije. Svaki od četiri senzora vibracije namijenjen je jednoj od četiri mete koje su kreirane. Navedeni senzori povezani su na mikroupravljač putem pinova D15, D2, D4 i D5. Tako je zapravo omogućena komunikacija između pokazivača l2C koji obavlja funkciju ispisivanja vrijednosti prilikom pogotka jedne od kreiranih meta. Senzori vibracije KY-031 imaju na sebi tri pina. Pin na njegovoj lijevoj strani služi za obradu digitalnoga signala, te se putem tog pina svaki od senzora povezuje s mikroupravljačem na već spomenute pinove D15, D2, D4 i D5. Srednji pin na KY-031 služi za povezivanje napajanja sa senzorom vibracije, točnije kako bi na njega bio doveden napon. Napajanje oba djela makete obavlja se putem univerzalnog napajanja koji ima mogućnost odabira napona od 3V do 12V. Napon koji se isporučuje na komponente od kojih se sastoji pokretna meta regulira se pomoću regulatora napona LM2596, koji pušteni napon od 7.5V regulira na 5V putem kojih se napajaju senzori vibracije. Zadaća zadnjeg pina na senzorima vibracije je uzemljenje. Uzemljenje i napajanje je realizirano zapravo tako da je više žica povezano u jednu. Ta jedna žica povezuje sve pinove sa senzora vibracije namijenjenih za napajanje i uzemljenje s regulatorom napona, točnije uzemljenje na OUT-, a napajanje na OUT+. Kao stavka koja upotpunjuje realizaciju pokretne mete uzima se pokazivač l2C. Na

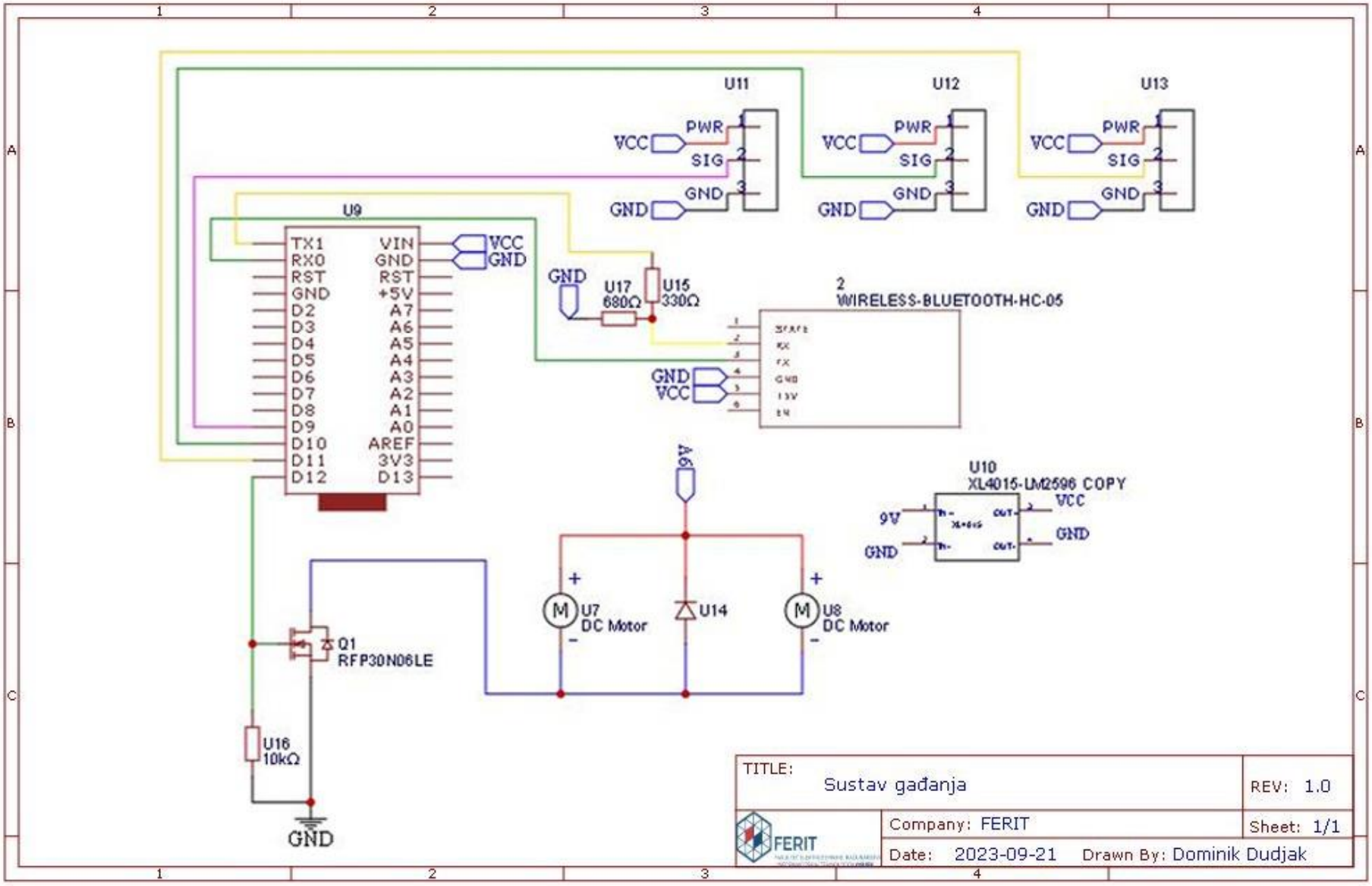
navedenom pokazivač nalaze se četiri oznake, a to su s lijeva na desno SCL, SDA, VCC i GND. Oznake SCL i SDA na komponenti su povezane putem pinova D21 i D22, a VCC i GND koje zapravo predstavljaju napajanje i uzemljenje su povezane s ESP-32 putem pinova 3V3 i GND kako bi pokazivač dobio potrebno napajanje.



Slika 3.6. *Električna shema pokretne mete.*

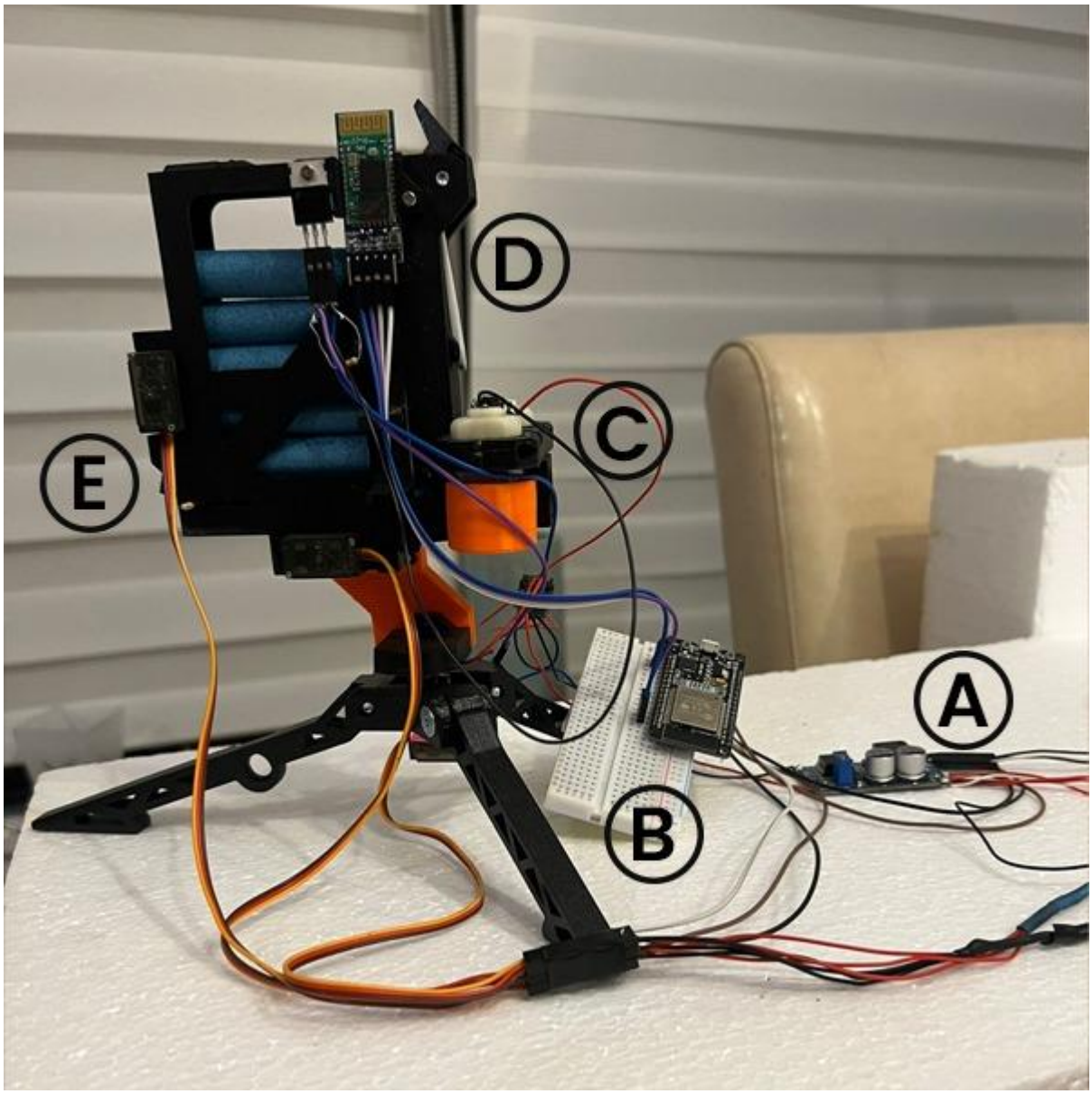


Slika 3.7. *Struktura pokretne mete.*


U11
U12
U13
PWR
SIG
GND
VCC
U9
TX1
RX0
RST
GND
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
VIN
GND
RST
+5V
A7
A6
A5
A4
A3
A2
A1
A0
AREF
3V3
D13
U17
680Ω
U15
330Ω
2
WIRELESS-BLUETOOTH-HC-05
U10
XL4015-LM2596 COPY
9V
U7
DC Motor
U14
U8
DC Motor
Q1
RFP30N06LE
U16
10kΩ
TITLE:
Sustav gađanja
REV: 1.0
FERIT
Company: FERIT
Sheet: 1/1
Date: 2023-09-21
Drawn By: Dominik Dudjak

Slika 3.8. *Električna shema sustava za gađanje mete.*

Slika 3.8 prikaz je električne sheme sustava za gađanje. Struktura sustava za gađanje realizirana je putem 3D printera unutar programa Prusa Slicer[9]. Na strukturu su vezane najbitnije komponente za realizaciju navedenog sustava za gađanje. Sustav gađanja temelji se na tri MG90S servo motora od kojih svaki od njih obavlja različitu zadaću, te su povezani s mikroupravljačem putem pinova D12, D13 i D14 . Prvi servo motor fiksiran je na podnožje strukture te obavlja funkciju rotacije sustava za gađanje u smjerovima lijevo i desno. Drugi servo motor je fiksiran na dio koji spaja spremnik za metke i podnožje strukture te on zapravo diže i spušta spremnik za metke. Treći i najbitniji servo motor obavlja funkciju pomicanja nogice koja gura metke iz spremnika prema naprijed. Napajanje navedenih servo motora izvršava se putem multifunkcionalnog napajanja putem kojeg se propušta napon od 7.5V, ali napon koji dolazi na servo motore je 5V jer se on regulira pomoću regulatora napona. Navedeni servo motori to jest njihove žice namijenjene za napajanje i uzemljenje povezane su u jedan sklop koji je povezan s regulatorom napona na oznake OUT+ i OUT-. Također na spremniku za metke, točnije njegovome izlaz nalaze se dva istosmjerna motora na koje je pušten napon od 7.5V. Njihova zadaća je metke iz spremnika koji se nogicom dovedu među njih svojom vrtnjom ispucati iz spremnika. Napajanje istosmjernih motora ide direktno s multifunkcionalnoga napajanje, te se ono ne regulira pomoću regulatora napona. Zadnja komponenta koja upotpunjuje sustav gađanja je Bluetooth modul HC-05. Navedeni modul omogućava komunikaciju između mikroupravljača i mobilne aplikacije. Točnije tako je omogućeno slanje informacija od strane rukovatelja, kako on zadaje funkcije koje servo i istosmjerni motori obavljaju. Kao što je prikazano na slici 3.8 navedeni modul povezan je s ESP-32 putem pinova RXO I TXO.

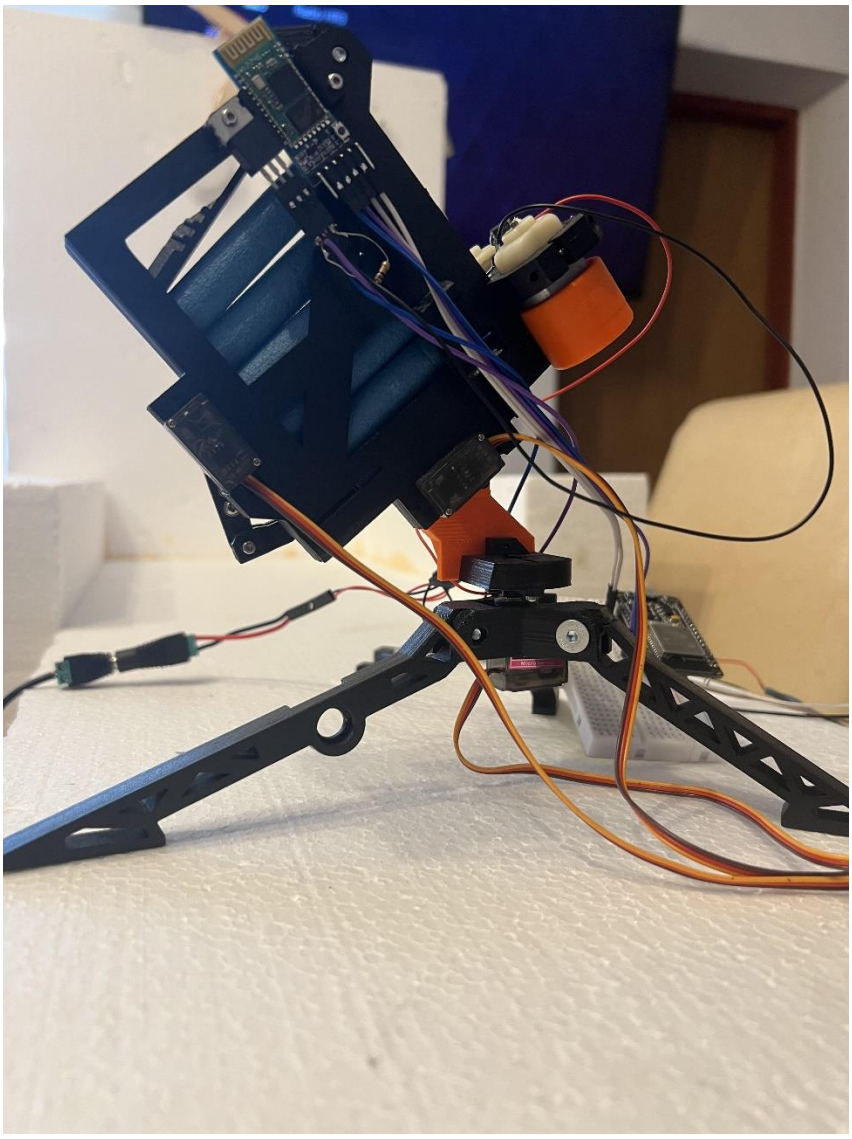


Slika 3.9. *Struktura sustava za gađanje mete.*

Na slici 3.9 prikazane su međusobne ovisnosti između upotrebljenih komponentni. Osim što je prikazana njihova međusobna povezanost vidljiva je struktura sustava za gađanje kreirana putem 3D printera. Također slika daje uvid kako su servo motori pozicionirani, te se samim time može lakše predočiti na koji način oni rotiraju strukturu i ispunjavaju svoje zadaće koje im se definiraju putem mobilne aplikacije.

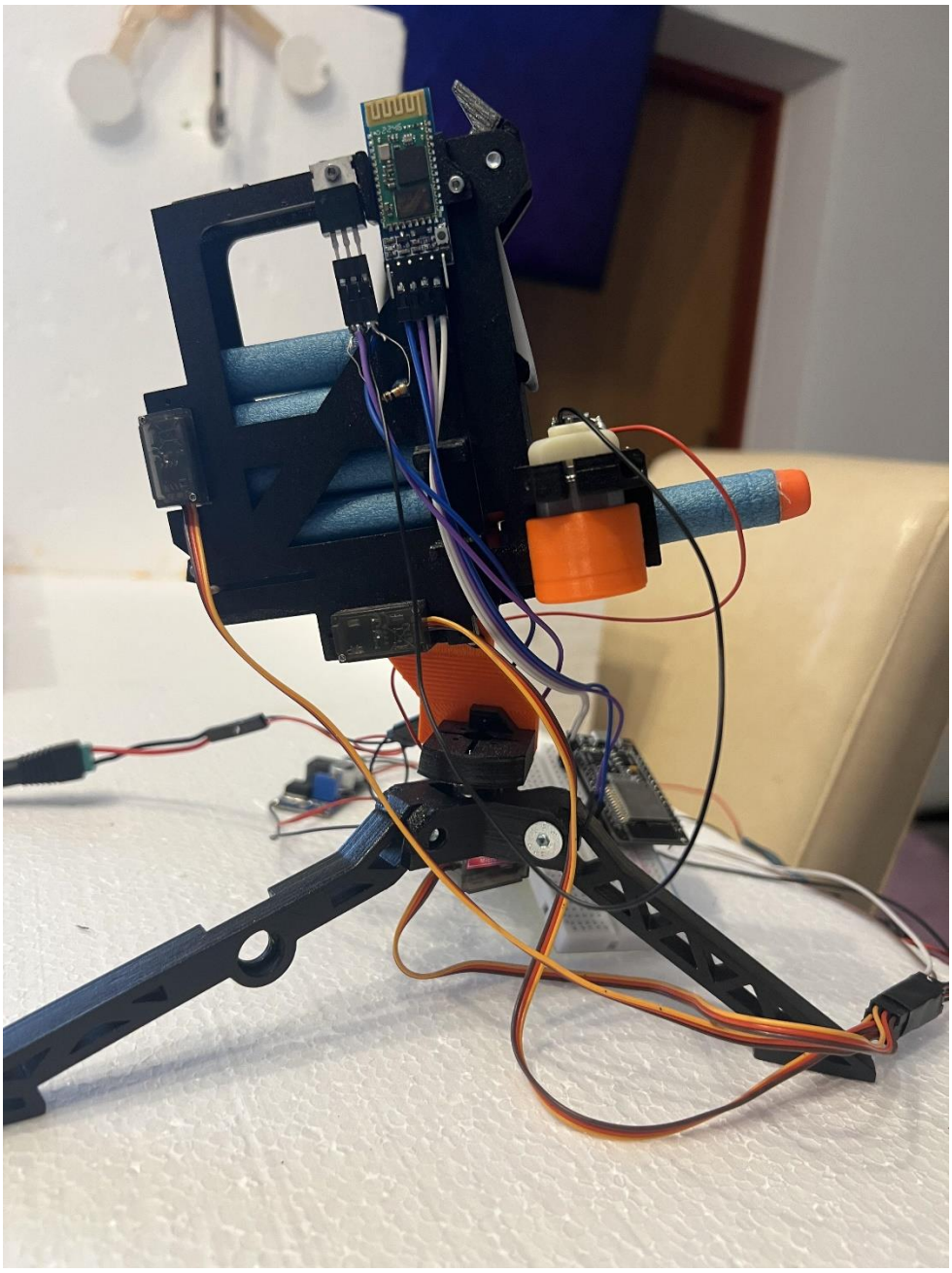
Slika 3.10. *Prikaz nogice povezane sa servo motorom.*

Na slici 3.10 prikazan je servo motor koji upravlja radom nogice. Prikazana nogica pomoću servo motora zadavanjem opcije za pucanje unutar sučelja mobilne aplikacije gura metke iz spremnika među istosmjerne motore koji svojom rotacijom ispaljuju metke.



Slika 3.11. *Prikaz pomicanja strukture prema gore pomoću servo motora.*

Slika 3.11 prikaz je strukture sustava za gađanje. Između nogica i na dnu spremnika za metke fiksirana su dva servo motora. Servo motor fiksiran između nogica omogućava rotaciju strukture u smjerovima lijevo i desno. Servo motor koji je fiksiran na dnu spremnika za metke pomiče spremnik prema gore i dolje. Kao što je prikazano na slici spremnik je po želji rukovatelja pozicioniran prema gore.

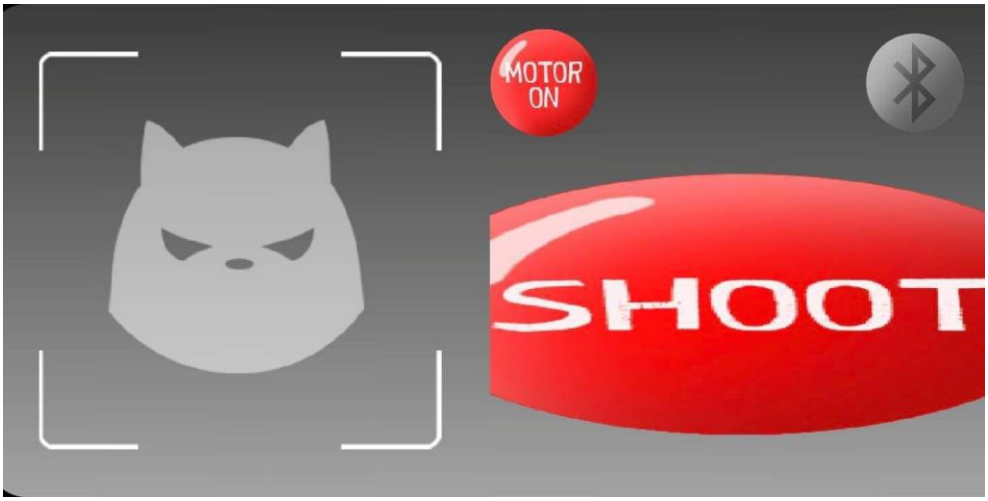


Slika 3.12. *Prikaz ispucavanja metka pomoću dva istosmjerna motora.*

Na slici 3.12 prikazan je trenutak u kojemu je metak pomoću nogice poguran iz spremnika za metke između dva istosmjerna motora. Motori se pale pomoću sučelja unutar mobilne aplikacije, te u tome trenutku svojom vrtnjom ispucavaju metak.

3.3. Realizacija programskog okruženja

Programsko rješenje kreće od toga da je potrebno omogućiti komunikaciju između mobilne aplikacije i mikroupravljača radi zaprimanja informacija i obavljanja zadataka zadanih od strane ljudskoga faktora. Komunikacija između dvije navedene stavke omogućena je bluetooth modulom. Kao što je prikazano na slici 3.8 Bluetooth modul je spojen na pinove RXO i TXO, te instaliranjem aplikacije na mobilni uređaj i opcijom uključivanja bluetooth-a na osobnom uređaju odabire se HC-05 modul. Nakon što je modul povezan s mobilnim uređajem putem aplikacije odabiru se željene radnje poput opcije pucanja. Putem programskog kôda realizirano je ispisivanje vrijednosti prilikom očitovanja udarca na pokazivač. Vrijednost je definirana unutar kôda, točnije na principu da ispisivanja vrijednosti ovisi o veličini mete. Unutar kôda definirano je da manja meta nosi više bodova, a veća meta manje bodova kao što je vidljiva vrijednost na slici 3.7 pod oznakom A prilikom pogotka jedne od meta. Aplikacija je koncipirana tako da sadrži sve bitne opcije za upravljanje sustavom za gađanje.

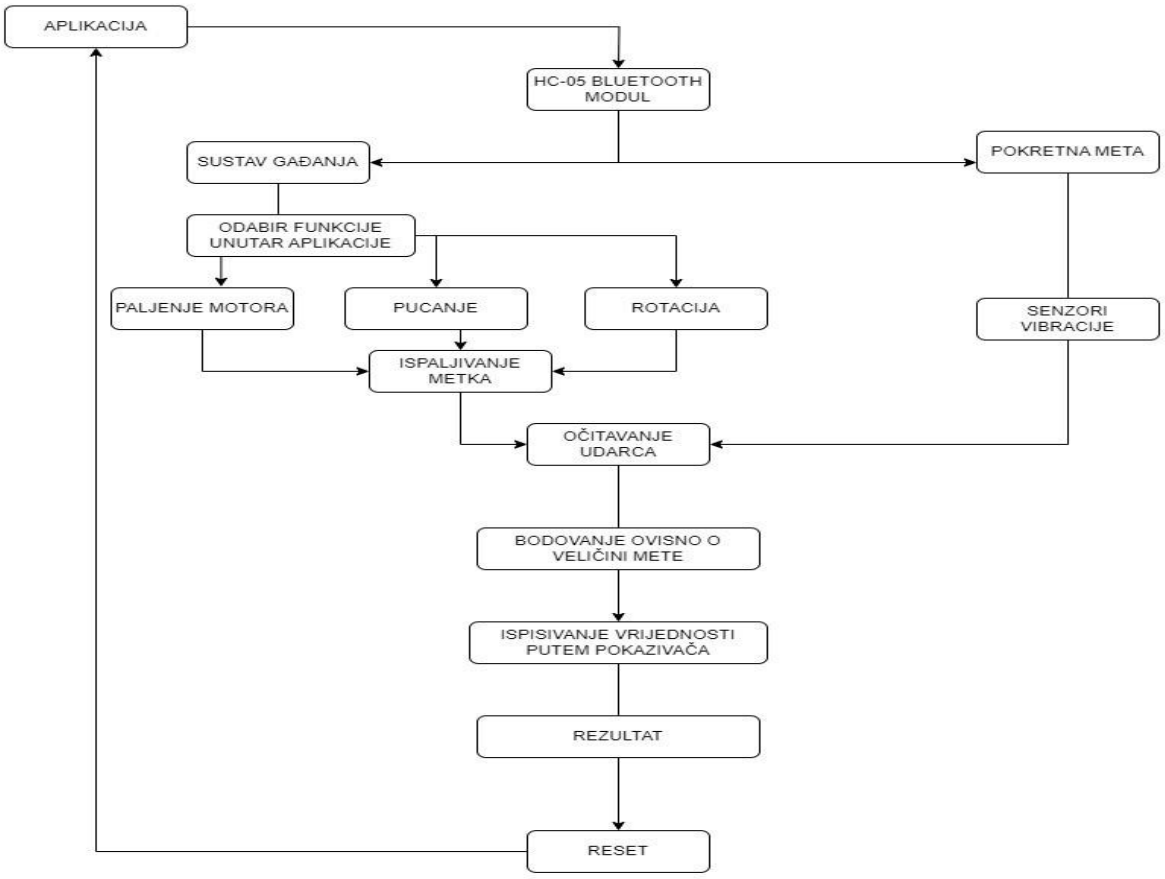


Slika 3.13. *Sučelje aplikacije za upravljanje sustavom za gađanje.*

Slika 3.13 prikazuje sučelje aplikacije za upravljanje sustavom za gađanje. Područje unutar kvadrata je namijenjeno za rotaciju sustava prstom ovisno o odluci rukovatelja. Putem opcije za paljenje motora pokreću se istosmjerni motori koji ispaljuju metke iz sustava za gađanje pritiskom tipke za pucanje. Putem funkcije paljenja motora unutar mobilne aplikacije određuje se rad motora, točnije njegovo gašenje i paljenje što može predstavljati prekid u radu sustava za gađanje.

Slika 3.14. *Povezivanje putem bluetooth modula.*

Na slici 3.14 vidljivo je sučelje unutar kojega se omogućuje povezivanje HC-05 modula s mobilnim uređajem putem kojega će se vršiti upravljanje sustavom za gađanje.



Slika 3.15. *Programsko rješenje makete sustava za gađanje i pokretne mete.*

Slika 3.15 prikazuje logički slijed operacija koje se moraju izvršiti kako bi se obavio proces. Proces započinje povezivanjem mobilne aplikacije putem HC-05 modula s ESP-32. Rukovatelj pomoću mobilne aplikacije zadaje naredbe kao što su pucanje, paljenje motora i rotacija u željenom smjeru. Nakon što se odabere jedna od naredbi ili ti opcija šalje se informacija o odabranoj naredbi prema mikroupravljaču unutar sustava za gađanje. Nakon što se izvrše naredbe dođe do ispucavanja metka iz sustava za gađanje. Kada metak pogodi jednu od meta senzor vibracije šalje informaciju o pogotku mete, te se zatim ispisuje vrijednost na pokazivač. Slika 3.7 pod oznakom A prikaz je ispisivanja vrijednosti putem pokazivača.

4. TESTIRANJE I REZULTATI

4.1. Metodologija testiranja

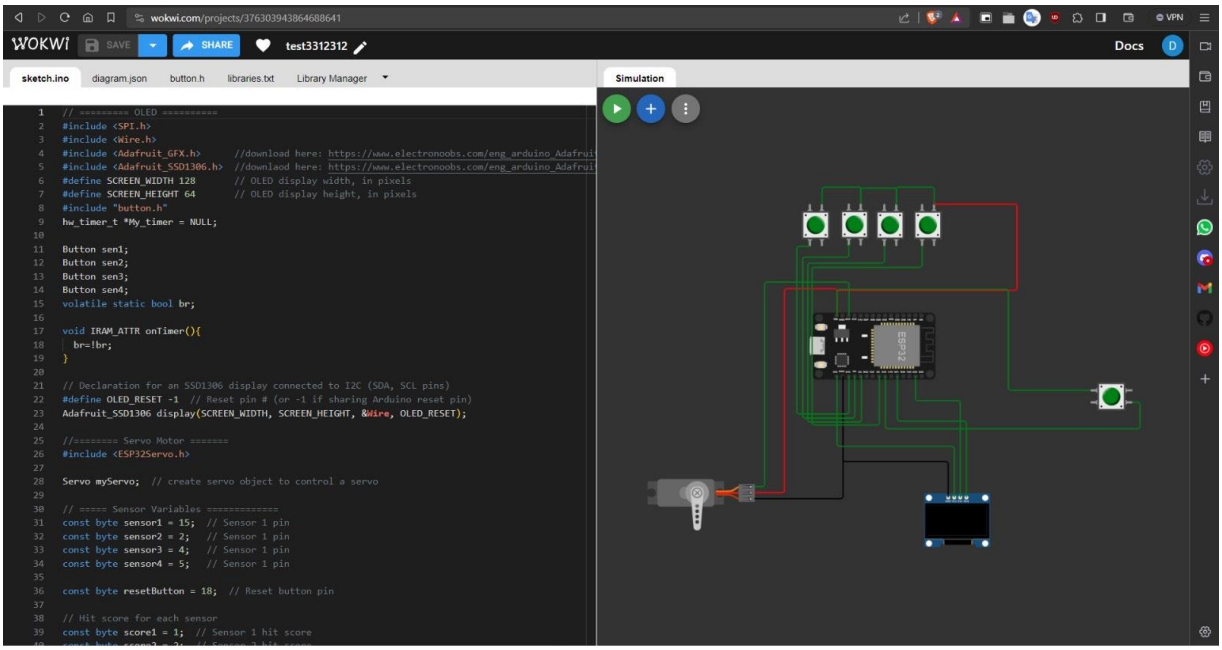
Prilikom odabira komponenti odlučeno je koristiti napajanje koje ima mogućnost propusnosti napona do 12V. Također pojavio se velik broj komponenti koje zatražuju napajanje. Od tih komponenti odabran je HC-05, tri servo motora te dva istosmjerna motora. Uvidom u specifikacije navedenih komponenti utvrđeno je da je potrebno navedene komponente napajati putem već spomenutoga napajanja te regulirati pušteni napon putem regulatora napona. Samim time pušteni napon od 7.5V se putem regulatora napona regulira na 5V, te se tako dovodi napajanje na navedene komponente i samim time omogućuje njihovo pokretanje. Nakon što je dovedeno napajanje na servo motore bilo je potrebno testirati njihovu funkcionalnost. Povezivanjem pucača putem HC-05 na aplikaciju za upravljanje pucačem utvrđena je sitna poteškoća da se servo motori ne rotiraju ispravno. Navedena problematika otklonjena je izmjenom kôda gdje su promijenjeni kutevi za sva tri servo motora. Prilikom samoga testiranja pucača utvrđeno je da istosmjerni motori pri napajanju od 9V proizvode preveliku silu za ispucavanje metka te je iz sigurnosnih razloga spušteno na 7.5V kako se ne bi proizvodila prevelika sila. Zbog utvrđivanja koliki je napon potreban za rad istosmjernih motora testirana je brzina ispaljivanja metaka iz spremnika. Na naponu manjem od 7.5V utvrđeno je da servo motor ne gura dovoljno brzo metke te oni zastaju u spremniku i ne dolaze između istosmjernih motora koji bi ih trebali ispaliti. U tablici su vidljivi rezultati to jest vrijeme koje je potrebno između dva hitca da oni iz spremnika dođu do krajnje točke točnije pokretne mete. Udaljenost na kojoj je vršeno testiranje je jedan metar od sustava za gađanje do pokretne mete. Temeljeno na rezultatima može se zaključiti da je period između ispaljivanja hitaca minimalan, i da sustav gađanja ispravno radi na 7.5V.

Tablica 4.1. *Prikaz mjerenja vremena potrebnog za ispaljivanje metka do pokretne mete.*

Hitac	Vrijeme
Hitac 1	00:00:83
Hitac 2	00:00:66
Hitac 3	00:00:76
Hitac 4	00:00:66
Hitac 5	00:00:71
Hitac 6	00:00:61
Hitac 7	00:00:63
Hitac 8	00:00:68
Hitac 9	00:00:72
Hitac 10	00:00:69
Hitac 11	00:00:63
Hitac 12	00:00:68
Hitac 13	00:00:62
Hitac 14	00:00:65
Hitac 15	00:00:69
Hitac 16	00:00:71
Hitac 17	00:00:69
Hitac 18	00:00:64
Hitac 19	00:00:67
Hitac 20	00:00:71

Prilikom testiranja utvrđeno je da na 9V kada se stisne opcija paljena motora unutar aplikacije dolazi do prevelikog zagrijavanja komponenata, te je tim putem utvrđeno da je napon potrebno regulirati na 7.5V kako ne bi dolazilo do zagrijavanja. U definiranju teme odmah je bio postavljen određeni zahtjev, a to je da se mete moraju konstantno kretati. Nakon povezivanja svih komponenti

koje su bile potrebne za realizaciju ovoga djela sustava za gađanje te implementiranja kôda bilo je potrebno testirati strukturu mete. Testiranje strukture obavljeno je tako da su sve komponente povezane, te spojene točnije zalijepljene na strukturu kako bi bila testirana konstantna rotacija putem servo motora. Testiranje rada pokretne mete također je provedeno i unutar programa Wokwi kao što je prikazano na slici 4.2 kako bi se utvrdio redoslijed izvršavanja funkcija unutar kôda prije implementiranja na ESP-32. Nakon što je ostvarena funkcionalnost pucača uz njegovu pomoć testirani su senzori vibracije. Unutar strukture pokretne mete kreirane su četiri mete na koje su povezani senzori vibracije. Svaki od četiri navedena senzora vibracije prilikom udarca metka iz pucača proizvodi vibraciju koja je potrebna za očitovanje siline udarca te omogućava ispisivanje zadane vrijednosti na zaslon. Krajnje testiranje je vršeno putem glavnog djela pokretne mete a to je servo motor koji mora omogućiti konstantno kretanje mete. Tu se pojavio problemom da je motor prilikom svoje rotacije zapetljavao žice te je bilo potrebno ograničiti njegovo kretanje to jest putem kôda definirati promjenu njegovoga smjera nakon zadanog perioda.



Slika 4.2. *Prikaz testiranja u Wokwi.*

4.2. Testiranje

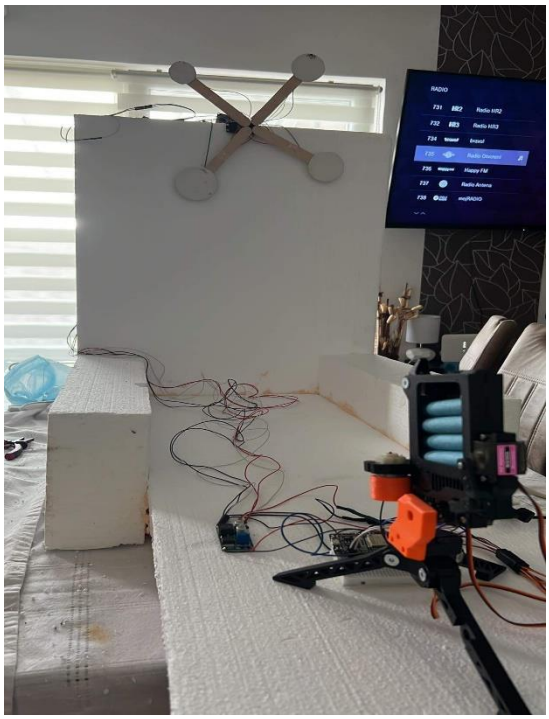
Testiranje je podijeljeno na dva zasebna testiranja. Prvotno je utvrđeno jesu li komponente ispravno povezane, te dolazi li na njih dovoljno napajanja. Nakon što je utvrđena ispravnost između komponenti započeto je zasebno testiranje sustava za gađanje a zatim i pokretne mete.

Testiranje sustava za gađanje podijeljeno je na nekoliko koraka:

- napajanje komponenata
- rotacija strukture točnije servo motora
- povezivanje putem HC-05

Testiranje mete također se dijeli na nekoliko koraka:

- funkcionalnost strukture
- testiranje senzora vibracije i ispisivanja vrijednosti na zaslon
- reguliranje rotacije servo motora



Slika 4.1. *Maketa sustava za gađanje i pokretne mete.*

Prikaz na slici 4.1 je gotova maketa na kojemu je vršeno testiranje i minimalizacija svih greški koje su bile prisutne do krajnjeg testiranja.

5. ZAKLJUČAK

Prilikom realizacije sustava koji je bio postavljen pred mene susreo sam se s dosta problema. Problemi su započeli već prilikom odabira komponenti jer neke od njih nisu bile adekvatne za provedbu istoga. Također problematika se nastavila prilikom same realizacije sustava za gađanje jer prilikom kreiranja pokretne mete servo motor je zapetljavao žice od napajanja i žice koje povezuju senzore vibracije s mikroupravljačem. Ta situacija je razriješena tako da se unutar kôda definiralo vrijeme kretanja u jednu stranu, a zatim njegova promjena u drugu stranu. Pojavljivanjem ova dva problema pojavila se prva mana ovoga rada a to su veliki financijski troškovi. Loša organizacija u početku i greške koje su se pojavljivale zatražile su sve veće preinake, samim time i kupovinu adekvatnijih komponenata koje će omogućiti realizaciju. Uz već spomenute financijske troškove kao drugu manu mogu istaknuti veličinu makete sustava za gađanje pokretne mete. Smatram da je maketa prevelika, te samim time se smanjuje njegova dostupnost i mobilnost jer nije lako prenosiv. To bi mogao biti jedan od razloga zašto on u budućnosti ako bi se plasirao na tržište ne bi bio dostupan svima. Smatram da je namijenjen svim uzrastima i kada bi ljudi imali pristup njemu da bi provodili dosta vremena zabavljajući se njime. Sama funkcionalnost bi se mogla poboljšati ako bi se postojeće napajanje zamijenilo da svaki dio makete ima zasebno napajanje putem baterije koja je dostupna svima. Zasebnim napajanjem bi se omogućio lakši transport te plasiranje na tržište ako bi se odlučio za taj korak. Također sama struktura sustava za gađanje točnije njegovo postolje bi se trebalo smanjiti kako bi se metak lakše doveo iz spremnika do istosmjernih motora jer bi time bio potreban manji napon, a time bismo smanjili mogućnost ozljede. Nadam se da će ovaj rad nekome u budućnosti biti orijentacija za izradu istoga ili sličnoga jer je kroz navedena poglavlja prikazana izrada makete, te sheme spajanja i odabir komponenata koje su potrebne za izradu.

LITERATURA

[1] Hrvatski streljački savez, https://hss-csf.hr/, pristup: 09.10.2023.

[2] Rayhaber, https://hr.rayhaber.com/2023/05/airsoft-silah-nedir-airsoft-silah-cesitleri-nelerdir/, pristup: 09.10.2023.

[3] Instructables, https://www.instructables.com/Arduino-Target-Practice/, pristup: 09.10.2023.

[4] Arduino Project Hub, https://projecthub.arduino.cc/adambeedle/full-auto-nerf-gun-that-shoots-you-in-face-using-opencv-6a372d, pristup: 09.10.2023.

[5] ESP32 mikroupravljač, slika preuzeta s: https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide, pristup: 05.10.2023.

[6] Regulator napona, slika preuzeta s : https://soldered.com/hr/proizvod/step-down-modul-s-lm2596s/, pristup: 05.10.2023.

[7] Visual Code Studio, https://dir.hr/sto-je-visual-studio/, pristup: 05.10.2023.

[8] Arduino IDE, https://www.arduino.cc/en/software, pristup: 05.10.2023.

[9] Prusa Slicer, https://stem.mik.hr/3d/uvod-u-prusa-slicer-alat-za-pripremu-modela-za-3d-ispis/, pristup: 05.10.2023.

SAŽETAK

Rad obuhvaća različite segmente planiranja i izrade sustava. Od početka do kraja je prikazan cjelokupni način izrade. Kao što je vidljivo unutar uvodnih poglavlja prikazuje programsko i sklopovsko planiranje. Samim time dovodi do prikaza koliko je bitno planirati kako bi se smanjili troškovi izrade. Rad prikazuje povezanost između odabira komponenata koje su potrebne za izradu samoga sustava i suvremenih tehnologija, što se može primijetiti u povezivanju komponenata kao što su ESP-32 i HC-05 Bluetooth modul. Nakon prikazivanja vodi nas kroz komponente koje su korištene, te nam na vrlo jednostavan i kratak način predočava tehničke mogućnosti korištenih komponenata, te koliko je bitno prvotno se upoznati s njima i njihovim karakteristikama. U završnom koraku nakon što su prikazana programska i sklopovska rješenja dolazi do testiranja navedenih rješenja, te uklanjanja greški do kojih su nas testiranja dovela. Testiranja su vršena kroz različite programe i simulatore baš kako bi se greške uklonile i kako bismo dobili funkcionalan sustav za gađanje.

Ključne riječi: mikroupravljač, sustav gađanja, pokretna meta, komunikacija

SUMMARY

Title: Model of the moving target shooting system

The work includes different segments of project planning and development. From start to finish, it shows the entire production method. As we can see in the introduction, it shows the programmatic and structural planning of the project. It emphasizes how important it is to plan in advance in order to reduce the costs of creating a project. The work shows the connection between the selections components that are necessary for the creation of the project itself and modern technologies, which can be noticed in the connection of components such as ESP-32 microcontroller and HC-05 Bluetooth module. After the presentation, it guides us through the components that were used. In a very simple and short way it presents the technical possibilities of the used components and how important it is to familiarize yourself with them and their characteristics first. In the final step after display of program and circuit solutions, it brings us to testing the named solutions and elimination of errors that the tests led us to. Testing was done through various programs and simulators so that errors would be removed and so that the project would be functional.

Key words: microcontroller, shooting system, moving target, communication

ŽIVOTOPIS

Dominik Dudjak rođen je 5. studenoga 1998. Godine u Osijeku. Pohađao je I. Gimnaziju u Osijeku, te je stekao status srednje stručne spreme. Nakon završetka srednjoškolskoga obrazovanja upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, Preddiplomski stručni studij Elektrotehnika, smjer automatika te paraleno k tome upisuje Ekonomski fakultet također u Osijeku. Za vrijeme svoga akademskoga obrazovanja okušao se u različitim studentskim poslovima, a trenutno je zaposlen u tvrtci za telekomunikacijske usluge kao agent za tehničku podršku.

PRILOZI I DODATCI

```
#include <ESP32Servo.h>

//deklariranje varijabli i servo motora
Servo recoil_servo;
Servo pan_servo;
Servo tilt_servo;

const byte pan_limit_1 = 0;
const byte pan_limit_2 = 180;
const byte tilt_limit_1 = 65;
const byte tilt_limit_2 = 180;
const byte recoil_rest = 180;    // Angle of the servo when at rest
const byte recoil_pushed = 125;  // Angle the servo need to reach to push the
dart

//-----Variables related to serial data handling
byte byte_from_app;
const byte buffSize = 30;
byte inputBuffer[buffSize];
const byte startMarker = 255;
const byte endMarker = 254;
byte bytesRecvd = 0;
boolean data_received = false;

//deklariranje varijabli za pucanje
bool is_firing =  false;
bool can_fire =  false;
bool recoiling = false;

unsigned long firing_start_time = 0;
unsigned long firing_current_time = 0;
const long firing_time = 150;

unsigned long recoil_start_time = 0;
unsigned long recoil_current_time = 0;
const long recoil_time = 2 * firing_time;

const byte motor_pin = 26 ;
boolean motors_ON = false;

//8==========================D

void setup()
{
  //-----define motor pin mode
  pinMode(motor_pin, OUTPUT);
  digitalWrite(motor_pin, LOW);

  //povezivanje servo motora s odgovarajućim pinovima
```

```
  ESP32PWM::allocateTimer(0);
  ESP32PWM::allocateTimer(1);
  ESP32PWM::allocateTimer(2);
  ESP32PWM::allocateTimer(3);
  recoil_servo.setPeriodHertz(50);           // standard 50 hz servo
  recoil_servo.attach(14, 1000, 2000);  // attaches the servo on pin 13 to the
servo object
  pan_servo.setPeriodHertz(50);           // standard 50 hz servo
  pan_servo.attach(13, 1000, 2000);
  tilt_servo.setPeriodHertz(50);           // standard 50 hz servo
  tilt_servo.attach(12, 1000, 2000);

  //-----starting sequence
  recoil_servo.write(recoil_rest);
  pan_servo.write(90);
  //tilt_servo.write(tilt_limit_2);
  delay(1000);
  //tilt_servo.write(tilt_limit_2 + abs((tilt_limit_2 - tilt_limit_1)/2));
  tilt_servo.write(105);


  Serial.begin(9600);  // begin serial communication
}

//8==========================D

void loop()
{
  getDataFromPC();
  set_motor();
  if (data_received) {
    move_servo();
    set_recoil();
    set_motor();
  }
  fire();
}

//8==========================D

void getDataFromPC() {

  //expected structure of data [start byte, pan amount, tilt amount, motor on,
firing button pressed, end byte]
  //start byte = 255
  //pan amount = byte between 0 and 253
  //tilt amount = byte between 0 and 253
  //motor on = 0 for off - 1 on
  //firing button pressed = 0 for not pressed - 1 for pressed
  //end byte = 254
```

```
  if (Serial.available()) {  // If data available in serial

    byte_from_app = Serial.read();   //read the next character available

    if (byte_from_app == 255) {     // look for start byte, if found:
      bytesRecvd = 0;                   //reset byte received to 0(to start
populating inputBuffer from start)
      data_received = false;
    }

    else if (byte_from_app == 254) {    // look for end byte, if found:
      data_received = true;               // set data_received to true so the
data can be used
    }

    else {                              // add received bytes to buffer
      inputBuffer[bytesRecvd] = byte_from_app;     //add character to input
buffer
      bytesRecvd++;                                // increment byte received
(this act as an index)
      if (bytesRecvd == buffSize) {    // just a security in case the inputBuffer
fills up (shouldn't happen)
        bytesRecvd = buffSize - 1;    // if bytesReceived > buffer size set
bytesReceived smaller than buffer size
      }
    }
  }
}

//8==========================D

void move_servo() {

  byte pan_servo_position = map(inputBuffer[0], 0, 253, pan_limit_2,
pan_limit_1);//convert inputbuffer value to servo position value
  pan_servo.write(pan_servo_position); //set pan servo position
  byte tilt_servo_position = map(inputBuffer[1], 0 , 253, tilt_limit_2,
tilt_limit_1); //convert inputbuffer value to servo position value
  tilt_servo.write(tilt_servo_position); //set pan servo position
}

//8==========================D

void set_recoil() {

  if (inputBuffer[3] == 1) {        //if fire button pressed
    if (!is_firing && !recoiling) { //and not already firing or recoiling
      can_fire = true;              //set can fire to true (see effect in void
fire())
    }
  }
  else {                  // if fire button not pressed
```

```
    can_fire = false;     //set can fire to false (see effect in void fire())
  }
}

//povezivanje istosmjernog motora s aplikacijom i njegovo paljenje

void set_motor() {


  if (inputBuffer[2] == 1) {                //if screen touched
    digitalWrite(motor_pin, HIGH);          //turn motor ON
    motors_ON = true;
  }
  else {                                   //if screen not touched
    digitalWrite(motor_pin, LOW);          //turn motor OFF
    motors_ON = false;

  }
}

//8==========================D

void fire() { //if motor byte on, turn motor on and check for time it has been on

  if (can_fire && !is_firing && motors_ON) {
    //if (can_fire && !is_firing) {
    firing_start_time = millis();
    recoil_start_time = millis();
    is_firing = true;
  }

  firing_current_time = millis();
  recoil_current_time = millis();

  if (is_firing && firing_current_time - firing_start_time < firing_time) {
    recoil_servo.write(recoil_pushed);
  }
  else if (is_firing && recoil_current_time - recoil_start_time < recoil_time) {
    recoil_servo.write(recoil_rest);
  }
  else if (is_firing && recoil_current_time - recoil_start_time > recoil_time) {
    is_firing = false;
  }

}




// inicijalizacija biblioteka

#include <SPI.h>
```

```
#include <Wire.h>

#include        <Adafruit_GFX.h>                                //download        here:
https://www.electronoobs.com/eng_arduino_Adafruit_GFX.php

#include             <Adafruit_SSD1306.h>              //downlaod             here:
https://www.electronoobs.com/eng_arduino_Adafruit_SSD1306.php

#define SCREEN_WIDTH 128       // OLED display width, in pixels

#define SCREEN_HEIGHT 64       // OLED display height, in pixels

#include "button.h"

hw_timer_t *My_timer = NULL;


Button sen1;

Button sen2;

Button sen3;

Button sen4;

volatile static bool br;


void IRAM_ATTR onTimer(){

  br=!br;

}


// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

#define OLED_RESET -1  // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


//======== Servo Motor =======
```

```
#include <ESP32Servo.h>

Servo myServo;  // create servo object to control a servo

// definiranje pinova na koje su povezani senzori vibracije
const byte sensor1 = 15;  // Sensor 1 pin
const byte sensor2 = 2;   // Sensor 1 pin
const byte sensor3 = 4;   // Sensor 1 pin
const byte sensor4 = 5;   // Sensor 1 pin

const byte resetButton = 18;  // Reset button pin

// ispisivanje vrijednosti za svaku metu
const byte score1 = 1;  // Sensor 1 hit score
const byte score2 = 2;  // Sensor 2 hit score
const byte score3 = 3;  // Sensor 3 hit score
const byte score4 = 4;  // Sensor 4 hit score

int totalScore = 0;  // Variable to store total score
int prevScore = 0;  // variable to hold previous score
unsigned int lastTime = millis();

void setup() {
  Serial.begin(9600);
```

```
  My_timer = timerBegin(0, 80, true);
  timerAttachInterrupt(My_timer, &onTimer, true);
  timerAlarmWrite(My_timer, 1000000, true);
  timerAlarmEnable(My_timer); //Just Enable

  // Declare all sensor pin as input
  pinMode(sensor1, INPUT_PULLUP);
  pinMode(sensor2, INPUT_PULLUP);
  pinMode(sensor3, INPUT_PULLUP);
  pinMode(sensor4, INPUT_PULLUP);
  pinMode(resetButton, INPUT_PULLUP);  // Set button pin as input with input pullup

  sen1.begin(sensor1);
  sen2.begin(sensor2);
  sen3.begin(sensor3);
  sen4.begin(sensor4);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  // initialize with the I2C addr 0x3C
(for the 128x32 or 64 from eBay)
  display.clearDisplay();

  myServo.attach(13);
  myServo.write(120);
}
```

```
void loop() {
  if (!digitalRead(resetButton)) {
    // resetiranje ukupnog rezultata
    totalScore = 0;  // Reset total score
    prevScore = -1;  // set previous score as -1
    delay(200);      // wait for 200 milliseconds
      display.clearDisplay();

  }
    //display.clearDisplay();

  if (sen1.debounce()) {
    totalScore += score1;
  } else if (sen2.debounce()) {
    totalScore += score2;
  } else if (sen3.debounce()) {
    totalScore += score3;
  } else if (sen4.debounce()) {
    totalScore += score4;

  }
  if (prevScore != totalScore) {
    display.setTextSize(9);
    display.setCursor(45, 0);
```

```
    display.print(totalScore);
    display.display();
  }
  servo_rotate();
}

void servo_rotate(){
  if (br){
    myServo.write(0);

  }
  else{
    myServo.write(120);}
```