

# Razvoj drona upravljano od strane Robotskog operacijskog sustava

---

Križanac, Karla

Master's thesis / Diplomski rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:949420>

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

*Download date / Datum preuzimanja:* **2025-03-10**

*Repository / Repozitorij:*

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



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

**Sveučilišni diplomski studij Računarstvo**

**Razvoj drona upravljano od strane Robotskog operacijskog sustava**

**Diplomski rad**

**Karla Križanac**

**Osijek, 2024.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMATIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju****Ocjena diplomskog rada na sveučilišnom diplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Karla Križanac
<b>Studij, smjer:</b>	Sveučilišni diplomski studij Računarstvo
<b>Mat. br. pristupnika, god.</b>	D-1134R, 22.10.2020.
<b>JMBAG:</b>	0165075028
<b>Mentor:</b>	izv. prof. dr. sc. Damir Filko
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	prof. dr. sc. Robert Cupec
<b>Član Povjerenstva 1:</b>	izv. prof. dr. sc. Damir Filko
<b>Član Povjerenstva 2:</b>	izv. prof. dr. sc. Emmanuel Karlo Nyarko
<b>Naslov diplomskog rada:</b>	Razvoj drona upravljano od strane Robotskog operacijskog sustava
<b>Znanstvena grana diplomskog rada:</b>	<b>Automatizacija i robotika (zn. polje elektrotehnika)</b>
<b>Zadatak diplomskog rada:</b>	Potrebno je razviti drona s kamerom koji će biti upravljan od strane Robotskog operacijskog sustava. REZERVIRANA TEMA!
<b>Datum ocjene pismenog dijela diplomskog rada od strane mentora:</b>	25.09.2024.
<b>Ocjena pismenog dijela diplomskog rada od strane mentora:</b>	Dobar (3)
<b>Datum obrane diplomskog rada:</b>	14.10.2024.
<b>Ocjena usmenog dijela diplomskog rada (obrane):</b>	Vrlo dobar (4)
<b>Ukupna ocjena diplomskog rada:</b>	Vrlo dobar (4)
<b>Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:</b>	18.10.2024.



**FERIT**

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

## IZJAVA O IZVORNOSTI RADA

Osijek, 18.10.2024.

**Ime i prezime Pristupnika:**

Karla Križanac

**Studij:**

Sveučilišni diplomski studij Računarstvo

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

D-1134R, 22.10.2020.

**Turnitin podudaranje [%]:**

1

Ovom izjavom izjavljujem da je rad pod nazivom: **Razvoj drona upravljano od strane Robotskog operacijskog sustava**

izrađen pod vodstvom mentora izv. prof. dr. sc. Damir Filko

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 pristupnika:

# Sadržaj

1. UVOD .....	1
1.1. Zadatak diplomskog rada.....	2
2. OPĆENITO O DRONOVIMA .....	3
2.1. Povijest dronova .....	3
2.2. Fizikalni principi rada kvadrokoptera.....	5
2.3. Sile i momenti koji djeluju na kvadrokopter .....	7
2.4. Primjena dronova.....	10
3. ROBOTSKI OPERACIJSKI SUSTAV I ARDUPILOT .....	12
3.1. Robotski operacijski sustav (ROS).....	12
3.1.1. Arhitektura robotskog operacijskog sustava.....	13
3.2. Ardupilot.....	15
3.3. Ardupilot i ROS integracija.....	16
4. OPIS HARDVERSKOG SKLOPOVLJA.....	18
4.1. Arhitektura hardvera.....	20
4.2. Okvir drona.....	21
4.3. Motori .....	22
4.4. Propeleri .....	23
4.5. Elektronički regulator brzine (ESC).....	24
4.6. Kontroler leta.....	25
4.7. Radio odašiljač .....	26
4.8. Telemetrijski radio.....	27
4.9. Radio prijemnik .....	28
4.10. Baterija .....	29
4.11. Raspberry Pi kamera.....	30
4.12. Raspberry Pi 3 Model B .....	31
4.13. Globalni sustav pozicioniranja (GPS) .....	32
4.14. DC-DC step down konverter (buck konverter) .....	34
4.15. Radio kontroler .....	35
5. HARDVERSKA KONFIGURACIJA DRONA .....	37
5.1. Kalibracija akcelerometra i kompasa .....	37
5.2. Radio kalibracija.....	39
5.3. Test motora i propelera, definiranje način letenja (manualno upravljanje).....	40
5.4. Postavljanje PID parametara za stabilan let .....	43
6. SOFTVERSKI SKLOP DRONA.....	46

6.1. Detaljan opis softverskog sklopa.....	46
6.2. Implementacija ROS-a na Raspberry Pi 3.....	47
6.3. Autonomno upravljanje dronom.....	48
6.3.1. Navigacija i planiranje putanje.....	50
6.3.2. Stabilizacija i kontrola leta.....	51
7. TESTIRANJE I ANALIZA .....	52
8. ZAKLJUČAK .....	67
LITERATURA.....	68
SAŽETAK.....	71
ABSTRACT .....	72
ŽIVOTOPIS .....	73
PRILOZI.....	74

# 1. UVOD

U današnjem svijetu, primjena bespilotnih letjelica, odnosno dronova je sveprisutna u različitim područjima, uključujući nadzor, dostavu, zabavu i istraživanje. Razvoj dronova je jedan od ključnih tehnoloških dostignuća 21. stoljeća te njihov razvoj postaje ključan izazov u istraživačkom i industrijskom sektoru, gdje se teži postizanju autonomnih sustava koji su sposobni za različite zadatke. Njihova sposobnost snimanja iz zraka, mogućnost praćenja objekata te izvršavanja zadatka ih čini neprocjenjivim alatom za razne namjene.

U ovome radu, integracija robotskog operacijskog sustava (ROS) postaje ključan korak prema razvoju naprednih autonomnih dronova.

Ovaj rad bavi se razvojem drona opremljenog kamerom te sa integracijom ROS-a omogućuje napredne mogućnosti upravljanja dronom, uključujući autonomno letenje, analizu podataka dobivenih sa senzora te vizualnu percepciju okoline putem kamere. Razvoj drona, uključujući njegovu izradu, implementaciju sustava kao i samo testiranje u stvarnim uvjetima predstavlja vrlo izazovan zadatak koji objedinjuje znanje iz elektronike, računarstva te automatizacije.

Detaljno će se razmotriti svaka faza razvoja ovog projekta, uključujući analizu potreba, dizajn sustava, implementaciju te rezultate evaluacije. Krajnji cilj je stvoriti funkcionalan i pouzdan autonomni dron koji može poslužiti kao platforma za daljnja istraživanja i primjene u području bespilotnih letjelica.

## **1.1. Zadatak diplomskog rada**

Zadatak diplomskoga rada obuhvaća cjeloviti razvoj drona od osnovnih dijelova do završne funkcionalnosti. Dron je izrađen vlastoručno pomoću različitih komponenti, uključujući elektroničke module, vijke, pri čemu su žice precizno zalemljene kako bi se osigurala stabilna i sigurna povezanost svih dijelova.

Nakon mehaničke izrade, dron mora biti sposoban za dvije vrste upravljanja: manualno i autonomno. U svrhu autonomnog upravljanja, dron je opremljen GPS-om i telemetrijskim radijom, što omogućuje precizno praćenje lokacije i komunikaciju na daljinu. Pored toga, dron koristi Raspberry Pi 3 računalo i Raspberry Pi kameru koji u kombinaciji s ROS-om omogućuju izvođenje naprednijih zadataka kao što su autonomni let na definiranu visinu i udaljenost.

Cilj je da dron može letjeti, snimati i prikupljati podatke iz okoline, odnosno fotografirati ili snimati videozapise. Implementacijom koda na Raspberry Pi, upravljanje dronom postaje moguće što omogućuje potpuno autonomno izvršavanje zadatka.



## 2. OPĆENITO O DRONOVIMA

Dronovi, poznatiji kao bespilotne letjelice, su letjelice kojima se upravlja daljinski ili autonomno bez prisutnosti pilota unutar letjelice. Riječ „dron” obuhvaća veliki spektar letjelica koje se mogu kretati na zemlji, po zraku ili na vodi. Dronovi su opremljeni različitim sensorima, kamerama, GPS modulima i drugim tehnologijama koje im omogućuju izvođenje složenijih zadataka bilo u rekreativne svrhe ili u industrijskim primjenama.

Razvoj dronova je rezultat kombinacije tehnoloških pomaka, uključujući napredak elektroničkih komponenti, naprednijih algoritama za autonomno upravljanje i razvoj baterija. S obzirom na to da su dronovi bespilotne letjelice, imaju mogućnost letenja u uvjetima opasnim za ljude, mogu izdržati više u zraku, što ih upravo iz tih razloga čini neprocjenjivim alatom u područjima kao vojni nadzor, istraživanje okoliša ili područjima koji su opasni na ljudski život i primjena zračne fotografije.

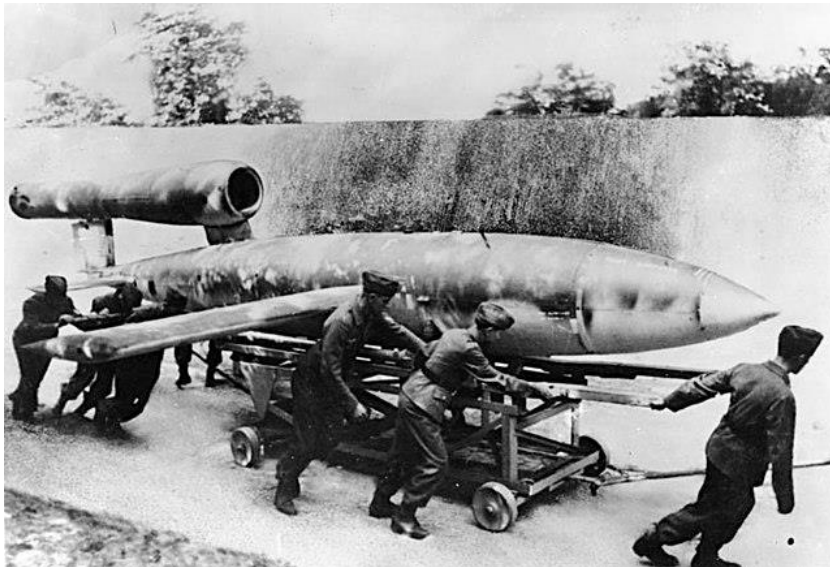
### 2.1. Povijest dronova

Prvi koncept bespilotnih letjelica potiče početkom 20. stoljeća tijekom Prvog svjetskog rata, gdje su razvijene prve verzije bespilotnih letjelica, kao što su „Kettering Bug”, bespilotna bomba vođena automatskim sustavom koja je predstavljala preteču modernih krstarećih projektila. Međutim, ovaj pokušaj izgradnje letjelice nije na kraju završio uspješno zbog tehnoloških ograničenja toga vremena. [1]



**Slika 2.1.** Prva verzije bespilotne letjelice „Kettering Bug”

Tijekom Drugog svjetskog rata, tehnologija je napredovala te je Njemačka razvila „V-1” leteću bombu koja je bila autonomna raketa vođena inercijskim navigacijskim sustavom. Nakon rata, dronovi i dalje evoluiraju, posebno u Sjedinjenim Američkim Državama i SSSR-u kada su počeli služiti kao mete za testiranje borbenih aviona i raketa. [2]



**Slika 2.2.** Leteća bomba „V-1”

Tijekom 1960.-tih i 1970.-tih, vojni dronovi su se počeli koristiti kao nadzor i izviđanje posebno tijekom Vijetnamskog rata. Vojni dron „RQ-2 Pioneer“ razvijen 1980., bio je jedan od prvih uspješno napravljenih operativnih nadzornih dronova, a njegova uporaba u Zaljevskom ratu 1991. godine dodatno je potaknula razvoj same tehnologije. [3]



**Slika 2.3.** Nadzorni dron „RQ-2 Pioneer”

S početkom 21. stoljeća, dronovi postaju znatno napredniji, posebno uz razvoj manjih i lakših elektroničkih sustava. Dronovi poput „MQ-1 Predator” postali su sinonim za vojnu primjenu, te osim nadzora, omogućavali su i precizne napade u vojnim operacijama. [4]



**Slika 2.4.** Dron „MQ-1 Predator”

Posljednjih desetljeća, dronovi su proširili svoju namjenu izvan vojne sfere. Danas, njihova primjena obuhvaća civilne i komercijalne sektore, gdje se koriste za zračne snimke, poljoprivredu, dostavu te znanstvena istraživanja. Njihov napredak tehnologije omogućio je razvoj pristupačnih i jednostavnih dronova koji su dostupni široj populaciji, što je dovelo do njihove popularnosti u rekreativnoj upotrebi.

## **2.2. Fizikalni principi rada kvadrokoptera**

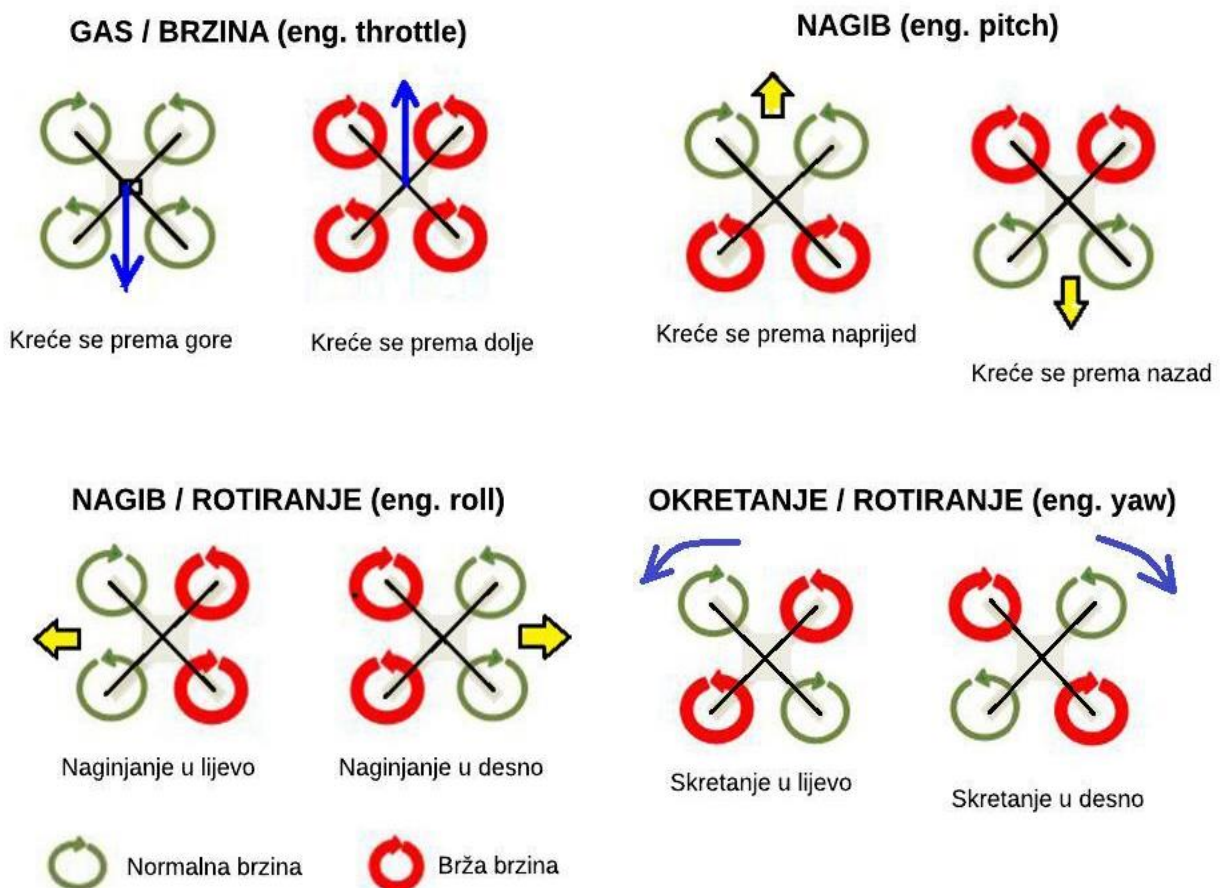
Kvadrokopteri su dronovi s četiri propelera, gdje svaki propeler pruža uzgon koji omogućava let drona. Prednost kvadrokoptera je u njihovoj sposobnosti održavanja stabilnosti i precizne kontrole gibanja zahvaljujući balansu potiska propelera.

Osnovni fizikalni princip rada kvadrokoptera se temelji na Newtonovom trećem zakonu, zakonu akcije i reakcije. Propeleri se rotiraju u parovima u suprotnim smjerovima, gdje dva propelera se rotiraju u smjeru kazaljke na satu, dok druga dva suprotno. Ova konfiguracija stvara stabilnost te eliminira potrebu za repnim rotorom, kao što je slučaj kod helikoptera. [5]

Kretanje dronova kontrolira se promjenom brzine rotacije pojedinih propelera.

Glavne kontrolne akcije:

- UZGON: Dron se uzdiže povećanjem brzine propelera, stvarajući veći uzgon ili smanjenjem brzine, stvarajući manji uzgon te spuštanje drona na tlo.
- NAGIB: Dron se nagine naprijed ili nazad promjenom brzine prednjih ili stražnjih propelera. Kada se prednji propeleri rotiraju brže, stražnji sporije, dron se nagine prema naprijed i kreće u tom smjeru.
- NAGIB (eng. *roll*): Dron se nagine lijevo ili desno kontroliranjem brzine bočnih propelera. Ako se lijevi propeleri rotiraju brže, dron se nagine na desnu stranu i obratno.
- OKRETANJE: Rotacija drona oko svoje osi kontrolira se promjenom relativne brzine rotacije suprotnih parova propelera. Ako se propeleri u jednom paru ubrzavaju, a u drugom usporavaju, dron će se rotirati u suprotnom smjeru. [6]



Slika 2.5. Prikaz rada motora na dronu prilikom upravljanja

### 2.3. Sile i momenti koji djeluju na kvadrokopter

Kada se dron kreće zrakom, na njega djeluju različite sile.

Glavne sile koje utječu na drona:

TEŽINA:

Sila mase tijela uvijek djeluje u smjeru gravitacije. Ova sila ovisi o masi drona i gravitacijskom ubrzanju što je prikazano formulom(2-1), gdje je  $g$  gravitacijsko ubrzanje i iznosi  $9.81 \text{ m/s}^2$ . Težina djeluje protiv uzgona te što je veća težina drona, potrebno je više energije za postizanje uzgona kako bi se dron podigao s tla.

$$\text{Težina drona} = \text{masa drona} \times g \quad (2-1)$$

UZGON:

Uzgon je sila koja djeluje okomito prema gore i suprotstavlja se težini drona. Ova sila omogućuje dronu da se podiže u zrak. Uzgon se stvara kada se propeleri okreću, što stvara razliku u tlaku između gornjeg i donjeg propelera. Ključni faktori koji utječu na uzgon su: brzina rotacije propelera jer što se brže propeleri okreću, veća je sila uzgona te veličina i oblik propelera jer veći propeleri stvaraju veći uzgon.

POTISAK:

Potisna sila je sila koja djeluje na dron u smjeru kretanja, bilo naprijed, nazad ili u stranu. Tijekom lebdenja, potisak je usmjeren vertikalno kako bi dron bio stabilan na mjestu. Tijekom kretanja prema naprijed/nazad potisak se naginje pod kutom prema dolje ili gore kako bi se dron pomaknuo naprijed ili nazad.

OTPOR:

Sila otpora je sila koja djeluje na dron u suprotnom smjeru od kretanja drona zbog otpora zraka. Otpor ovisi o nekoliko faktora: što se dron brže kreće, otpor je veći. Zrak ima određeni otpor prema tijelima koja se kroz njega kreću, a dron mora savladati tu silu bilo da je u pitanje održavanje brzine ili povećavanje brzine.

Uz glavne sile, također postoji i moment sile, koji nastaje zbog rotacije propelera. Propeleri koji se vrte u smjeru kazaljke na satu stvaraju moment sile koji nastoji rotirati drona u suprotnom smjeru. Zato da bi se taj efekt kompenzirao, postoji drugi par propelera koji se vrte u suprotnom smjeru od kazaljke na satu te neutraliziraju moment sile i omogućuje stabilan let dronu.

Kinematika kvadrokoptera odnosi se na proučavanje gibanja kvadrokoptera bez obzira na sile koje uzrokuju to gibanje. Uključuje opis položaja, brzine i ubrzanja kvadrokoptera u prostoru.

### 1. POTISAK PROPELERA

Potisak proizvode propeleri te je uvijek okrenut na ravninu rotacija propelera i proporcionalan je kvadratu kutne brzine propelera.

$$F_i = k_f \times w_i^2 \quad (2-2)$$

gdje je  $F_i$  potisak propelera  $i$ ,  $k_f$  konstanta proporcionalnosti te  $w_i$  kutna brzina rotacije propelera  $i$ . To znači da se povećanjem brzina rotacije propelera povećava potisak kvadratno.

### 2. MOMENT OKO X I Y OSI

Ako je  $L$  udaljenosti između dva motora ili propelera duž dijagonale drona, momenti reakcije oko X-osi i Y-osi izračunavaju se kao razlika sila koje djeluju na dijagonalne propelere.

Moment oko X-osi ( $M_x$ ):

$$M_x = (F_3 - F_4) \times L \quad (2-3)$$

gdje su  $F_3$  i  $F_4$  sile koje stvaraju propeleri 3 i 4.

Moment oko Y-osi ( $M_y$ ):

$$M_y = (F_1 - F_2) \times L \quad (2-4)$$

gdje su  $F_1$  i  $F_2$  sile koje stvaraju propeleri 1 i 2.

### 3. II. NEWTONOV ZAKON (linearno i rotacijsko gibanje)

Prema II. Newtonovom zakonu, sile i momenti koji djeluju na dron definiraju njegovo kretanje:

Za linearno gibanje:

$$F = m \times a \quad (2-5)$$

gdje je  $F$  ukupna sila koja djeluje na tijelo kvadrokoptera,  $m$  je masa kvadrokoptera te  $a$  je ubrzanje kvadrokoptera.

Za rotacijsko gibanje:

$$t = I \times a \quad (2-6)$$

gdje je  $\tau$  ukupni moment sile koji djeluje na kvadrokopter,  $I$  moment inercije kvadrokoptera oko osi rotacije, te  $\alpha$  kutno ubrzanje kvadrokoptera.

Ove jednadžbe koriste se za definiranje ponašanja drona pri različitim načinima gibanja, poput lebdenja, uzdizanja, okretanja ili naginjanja.

#### 4. UVJETI ZA LEBDENJE

Za stabilno lebdenje dron mora biti u ravnoteži. To znači da ukupna sila uzgona koja dolazi od propelera mora biti jednaka težini drona, dok su svi momenti jednaki nuli:

$$T = F_1 + F_2 + F_3 + F_4 = m \times g \quad (2-7)$$

gdje je  $T$  ukupna sila uzgona,  $F_1, F_2, F_3$  i  $F_4$  sile koje proizvode propeleri,  $m$  je masa kvadrokoptera te  $g$  je gravitacijsko ubrzanje.

Krajnja jednadžba gibanja za lebdenje:

$$T = F_1 + F_2 + F_3 + F_4 - m \times g = 0 \quad (2-8)$$

to znači da se dron neće uzdizati niti spuštati jer su sile u ravnoteži.

#### 5. UZDIZANJE / SPUŠTANJE DRONA

Za uzdizanje drona, uzgon mora biti veći od sile gravitacije:

$$m \times g < F_1 + F_2 + F_3 + F_4 \quad (2-9)$$

Za spuštanje drona, sila gravitacije mora biti veća od ukupnog uzgona:

$$m \times g > F_1 + F_2 + F_3 + F_4 \quad (2-10)$$

U uvjetima lebdenja, ukupna sila ostaje ista:

$$m \times g = F_1 + F_2 + F_3 + F_4 \quad (2-11)$$

#### 6. ROTACIJA OKO Z-OSI

Za gibanje oko Z-osi, rotacija se kontrolira povećavanjem ili smanjenjem brzina suprotnih propelera, stvarajući moment rotacija. Na primjer, ako se poveća brzina lijevih propelera, dron će se okretati udesno. Rotacija oko Z-osi može se opisati koristeći II. Newtonov zakon za rotaciju:

$$I_{zz} \cdot a_z = M_{net} \quad (2-12)$$

gdje je  $I_{zz}$  trenutak inercije drona oko Z-osi,  $a_z$  kutna akceleracija oko Z-osi te  $M_{net}$  koji je neto moment koji djeluje na dron oko Z-osi.

Neto moment  $M_{net}$  može se izračunati:

$$M_{net} = M_1 + M_2 + M_3 + M_4 \quad (2-13)$$

gdje su  $M_1$ ,  $M_2$ ,  $M_3$  i  $M_4$  momenti koje stvaraju pojedini propeleri. Ti momenti ovise o razlikama u brzini rotacije propelera. Na primjer, ako su propeleri 1 i 2 brži od 3 i 4, dron će se kretati udesno.

Kvadropteri su iznimno stabilni, jednostavni za kontrolu što ih skupa s njihovom fleksibilnošću čini idealnim za široku primjeru od komercijalnih dronova do profesionalnih sustava za nadzor i istraživanje. [6]

## 2.4. Primjena dronova

Pored vojne primjene koja je opisana ranije, najčešće primjene drona su:

### 1. POLJOPRIVREDA

Dronovi omogućuju nadzor poljoprivrednih površina, detekciju zdravlja usjeva, praćenje vlažnosti tla te preciznu primjenu gnojiva i pesticida tako da dronovi u poljoprivredi znatno povećavaju učinkovitost same proizvodnje. [7]

### 2. GEODETSKA ISTRAŽIVANJA

Dronovi se koriste za izradu detaljnijih zračnih karata i 3D modela terena. Dronovi opremljeni kamerom i LIDAR sensorima (eng. *light detection and ranging*) omogućuju precizno mapiranje čak i u nepristupačnim uvjetima i područjima. [8]

### 3. NADZOR I SIGURNOST

Dronovi se koriste za nadzor velikih područja, granica i kritičnih infrastruktura. U kombinaciji s naprednim sustavima za prepoznavanje objekata, pomažu u sprječavanju sigurnosnih prijetnji. [9]

### 4. HITNA DOSTAVA I MEDICINSKA PRIMJENA

Dronovi se koriste u hitnim slučajevima za dostavu lijekova, medicinske opreme i hitnih potrepština na teško dostupna područja.



## 5. ZRAČNA FOTOGRAFIJA I FILM

U komercijalnim i umjetničkim projektima, dronovi omogućuju snimanje spektakularnih zračnih snimaka, što je vrlo korisno u industriji filma, marketinga i nekretnina.

## 6. ISTRAŽIVANJE I ZNANOST

Dronovi pomažu u prikupljanju podataka o okolišu, istraživanju vulkana, ledenih polja i praćenju divljih životinja. U područjima poput ekologije i meteorologije, dronovi omogućuju istraživačima prikupljanje vrijednih podataka u stvarnom vremenu.

Zahvaljujući njihovoj svestranosti, dronovi će nastaviti evoluirati i pronalaziti nove primjene, što ih čini jednim od najvažnijih alata u razvoju novih tehnologija i poboljšanju kvalitete života.

### **3. ROBOTSKI OPERACIJSKI SUSTAV I ARDUPILOT**

Kombinacija ROS-a i ArduPilota pruža snažan alat za razvoj autonomnih sustava, omogućujući inženjerima i istraživačima da iskoriste prednosti oba sustava. Razvoj robota i dronova postaje učinkovitiji i fleksibilniji, otvarajući vrata za inovacije u različitim područjima, od istraživanja i razvoja do komercijalnih aplikacija.

#### **3.1. Robotski operacijski sustav (ROS)**

Robotski operacijski sustav (ROS) je otvorena platforma za razvoj softvera u robotici, koja omogućuje jednostavnu integraciju različitih sustava, senzora i aktuatora potrebnih za upravljanje robotom, uključujući dronove. Iako ROS nije klasičan operacijski sustav u smislu Windowsa ili Linuxa, on pruža infrastrukturu za komunikaciju između različitih programskih modula unutar robota te time olakšava razvoj složenih robotskih aplikacija.

ROS je razvijen kako bi omogućio modularni pristup razvoju robotskih sustava, što znači da se različite komponente poput senzora, kontrolera i algoritama mogu razvijati, testirati i nadograđivati neovisno jedni o drugima. Ovaj pristup čini ROS fleksibilnijim i prilagodljivijim sustavom koji se može koristiti u različitim robotskim aplikacijama, uključujući naravno i dronove. [10]

Glavne značajke ROS-a:

##### **1. DISTRIBUIRANI SUSTAV**

ROS koristi mrežni sustav komunikacije putem objavitelj/pretpatnik mehanizma gdje različiti čvorovi (npr. senzori, kamere) mogu komunicirati jedni s drugima. To omogućava učinkovitu razmjenu podataka između dijelova sustava.

##### **2. FLEKSIBILNOST I KOMPATIBILNOST**

ROS podržava raznovrsne senzore i kontrolere, čineći ga prikladnim za primjenu u područjima robotike. ROS se može koristiti na različitim hardverskim platformama, uključujući Raspberry Pi što je važno kada su u pitanju razvoji malih sustava poput dronova.

##### **3. ALGORITMI**

ROS dolazi s velikim brojem gotovih algoritama za navigaciju, mapiranje, percepciju i kontrolu gibanja, što omogućuje razvoj naprednih funkcionalnosti bez potrebe za pisanjem svih algoritama od nule.

#### 4. SIMULACIJA I TESTIRANJE

ROS se često koristi s alatima za simulaciju poput Gazebo- a (simulator robotskih sustava) te omogućuje testiranje robotskih sustava u virtualnom okruženju prije njihove primjene u stvarnom svijetu.

##### 3.1.1. Arhitektura robotskog operacijskog sustava

Arhitektura je organizirana kao distribuirani sustav što znači da su sve komponente podijeljene u čvorove, a ti čvorovi međusobno komuniciraju putem objavitelj/pretplatnik mehanizma ili usluga. Čvor je odgovoran za specifične funkcionalnosti, kao što su senzorske obrade, upravljanje motorima ili planiranja kretanja. Njegova arhitektura omogućuje fleksibilnost, jer se čvorovi mogu razvijati, testirati i zamjenjivati neovisno jedni o drugima. [11]

Glavne komponente ROS-a:

1. ČVOROVI (eng. *nodes*): čvorovi su osnovne jedinice programa koje izvršavaju određene funkcije, poput senzorskih podataka ili upravljanja aktuatorima.
2. PORUKE (eng. *messages*): čvorovi komuniciraju jedni s drugima slanjem ili primanjem poruka. Poruke prenose informacije o senzorskim podacima ili naredbama za kretanje.
3. TEME (eng. *topics*): Teme služe kao kanali za razmjenu poruka između čvorova. Čvor može objaviti poruku na temu („objava”) dok drugi čvor može slušati (prijem) na temu („pretplata”).
4. SERVISI (eng. *services*): Za razliku od teme, servisi omogućuju dvosmjernu komunikaciju između čvorova, gdje jedan čvor šalje zahtjev, a drugi čvor vraća odgovor.

ROS akcije:

ROS akcije su mehanizam komunikacije koji omogućuje čvorovima da izvršavaju dugotrajne zadatke. ROS je prisutan često kada su u pitanju dronovi zbog njihove sposobnosti da upravljaju dugotrajnim i kompleksnim zadacima.

Komponente ROS akcije su:

1. POSLUŽITELJ: Čvor koji izvršava dugotrajni zadatak. Prima ciljeve od klijenata i obrađuje ih.
2. KLIJENT: Čvor koji šalje ciljeve poslužitelju. Također može primiti povratne informacije i rezultate od poslužitelja.

3. PORUKE: To su strukturirane poruke koje definiraju komunikaciju između klijenta i poslužitelja.

Svaka akcija se sastoji od:

1. CILJ: Željeni ishod koji je poslužitelj nastojao postići.
2. REZULTAT: Konačni ishod nakon što je cilj završen.
3. POVROTNA INFORMACIJA: Međusobna ažuriranja od poslužitelja klijentu tijekom izvršenja akcije (npr. udaljenost do cilja, brzina, itd.). [12]

Utjecaj ROS akcije na dronove:

Korištenjem ROS akcija, dron može primiti cilj i izvršavati misiju korak po korak, s mogućnošću slanja povratnih informacija ili otkazivanja misije ako je potrebno. Dronovi mogu koristiti akcije za praćenje objekata ili izbjegavanje prepreka. U slučajevima kada više dronova surađuje, akcije se mogu koristiti za koordinaciju njihovih kretanja. Na primjer, jedan dron može zatražiti od drugih da se pomaknu u skladu s novim ciljem, a svi dronovi mogu slati povratne informacije o svom statusu i poziciji.

Dronovi mogu koristiti ROS akcije za navigaciju do određenih točaka u prostoru. Klijent može poslati ciljnu točku dronu, a dron će se zatim kretati prema toj točki dok šalje informacije o svome statusu. Dronovi mogu koristiti akcije za praćenje objekata ili izbjegavanje prepreka. Poslužitelj može primiti ciljeve za praćenje i neprekidno ažurirati klijenta o statusu i eventualnim preprekama na putu.

Jezici korišteni u ROS-u:

ROS podržava više programskih jezika što ga čini pristupačnim za širu zajednicu programera.

Najčešći korišteni jezici su:

- PYTHON

ROS koristi Python kroz paket nazvan „rospy”. Python se često koristi zbog svoje jednostavnosti, brzine razvoja i široke uporabe u robotskim aplikacijama. Zbog svoje jednostavnosti, Python je vrlo popularan među istraživačima i studentima, posebno za testiranje novih ideja i algoritama. [13]

- C++

C++ je još ključan jezik u ROS-u, poznat pod nazivom „roscpp”. C++ se koristi u slučajevima kada su potrebne visoke performanse, poput upravljanja motorima ili obrada složenih senzorskih podataka. Zbog svoje brzine i učinkovitosti, C++ je vrlo često izbor za implementaciju dijelova sustava koji zahtijevaju stvarno vrijeme ili gotovo stvarne performanse. [14]

- JAVA I LISP

Iako su Python i C++ najčešći korišteni jezici, postoje implementacije za druge jezike, uključujući Java i Lisp, koji se mogu koristiti neovisno o specifičnim potrebama aplikacije.

ROS koristi fleksibilan pristup pri razvoju aplikacija, pa se moduli mogu razvijati u bilo kojem od podržanih jezika, pa zatim povezati u jedan sustav putem standardiziranih ROS komunikacijskih protokola. Ovo omogućuje razvojnim timovima da mogu kombinirati različite jezike i alate prema potrebi.

## 3.2. Ardupilot

Ardupilot je otvoreni sustav za autopilotiranje koji podržava širok raspon bespilotnih letjelica, uključujući kvadrokoptere, avione, helikoptere i druge robote. Ardupilot je razvijen kako bi pružio precizne kontrole letjelicama, bilo da se s njima upravlja manualno pomoću daljinskog upravljača ili autonomno, koristeći GPS i ostale senzore.

Poznat je po svojoj fleksibilnosti, robusnosti i što pruža podršku za široki spektar hardverskih platformi. Također pruža podršku za autonomne letove, navigaciju prema unaprijed definiranim točkama.

Ključne značajke Ardupilot-a:

### 1. AUTONOMNA NAVIGACIJA:

Ardupilot omogućuje autonomne letove prema unaprijed definiranim putanjama koji se programiraju putem softvera kako što su Mission Planner ili QgroundControl. Dron može automatski slijediti zadane točke leteći s izuzetnom preciznošću zahvaljujući integraciji s GPS-om i IMU sensorima.

## 2. SENZORSKA INTEGRACIJA:

Ardupilot podržava mnoge senzora, poput GPS-a, akcelerometre, žiroskope, barometre i magnetometre, koji pomažu u stabilizaciji leta, određivanju položaja i kontroliranju visine. Senzori omogućuju dronu da održava stabilan let čak i uvjetima jakog vjetra i drugih vanjskih čimbenika.

## 3. MODOVI LETENJA:

Ardupilot pruža različite načine letenja, uključujući ručno upravljanje, stabilizirani let, automatski let, loiter mod letenja(zadržavanje na određenoj poziciji u zraku), RTL (eng. *return to launch*) mod koji omogućuje dronu automatski povratak na početku točku.

## 4. TELEMETRIJA I PRAĆENJE:

Ardupilot omogućuje praćenje leta u stvarnom vremenu putem telemetrijskih sustava. Podaci uključuju položaj drona, brzinu, visinu, status baterije i druge parametre, što je ključno za upravljanje dronom. [15]

### 3.3. Ardupilot i ROS integracija

Jedna od ključnih prednosti Ardupilot-a je njegova kompatibilnost s ROS-om. Ova kombinacija omogućuje razvoj autonomnih sustava koji mogu obavljati kompleksne zadatke, kao što su dinamičko planiranje putanje, izbjegavanje prepreka i obrada podataka sa senzora u stvarnom vremenu. Kompatibilnost između Ardupilot-a i ROS-a omogućuje integraciju robusnog sustava za upravljanje dronovima s naprednim funkcionalnostima.

Ardupilot i ROS koriste različite protokole za komunikaciju: ArduPilot koristi MAVLink, protokol za komunikaciju između dronova i kontrolnih sustava, dok ROS koristi svoj vlastiti sustav poruka. Međutim, postoji ROS paket nazvan „mavros”, koji djeluje kao most između ROS-a i MAVLink-a. Ovaj paket omogućuje ROS čvorovima da komuniciraju s Ardupilot-om u različitim aspektima, uključujući:

- **Kretanje:** upravljanje promjenama smjera leta drona.
- **Visina:** podešavanje visine leta drona.
- **Brzina:** mijenjanje brzine leta drona.

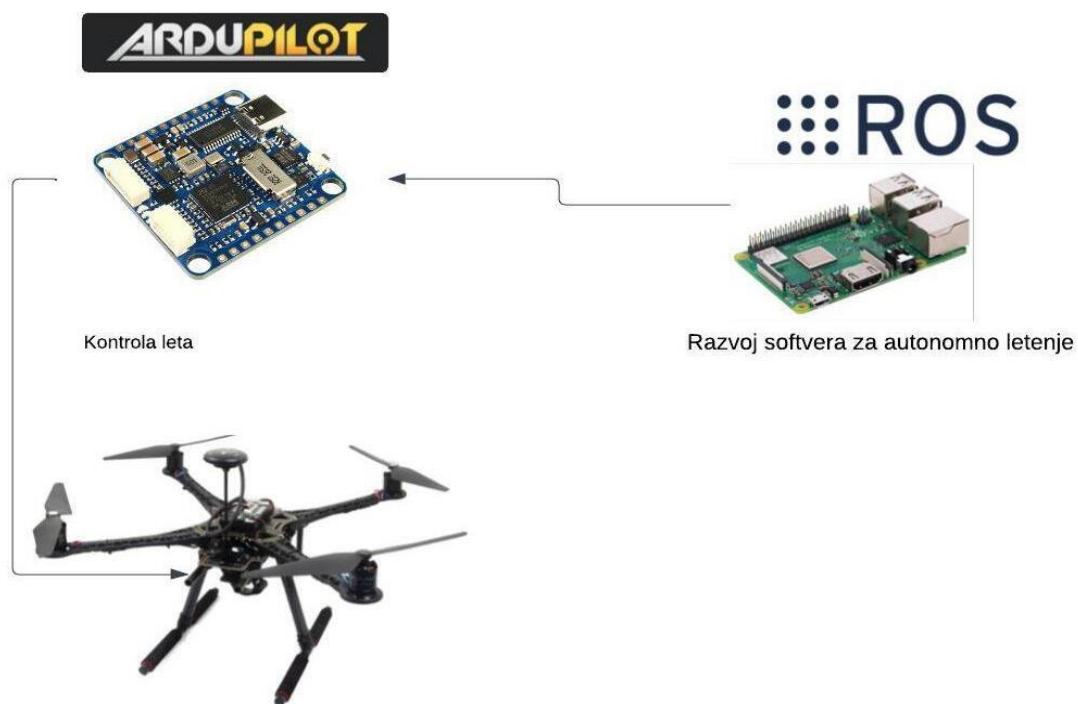
Korištenjem Raspberry Pi-a za ovu integraciju, moguće je izvršavanje ROS čvorova koji komuniciraju s Ardupilot-om putem MAVROS paketa. Ova se komunikacija ostvaruje

povezivanjem Raspberry Pi-a s kontrolerom leta putem USB veze, čime se omogućava brza i učinkovita razmjena podataka. Raspberry Pi također može upravljati sensorima poput kamera i LIDAR-a, koristeći ROS za kreiranje mapa okoline, prepoznavanje objekata te praćenje stanja drona u stvarnom vremenu.

Integracija Ardupilot-a i ROS-a donosi brojne mogućnosti. Na prvom mjestu, omogućuje integraciju različitih senzora, što doprinosi naprednijem upravljanju letom i percepciji okoline. Osim toga, ROS omogućuje implementaciju složenih algoritama kontrole i planiranja putanja, kao i simulaciju u alatu poput Gazebo. Ovo smanjuje rizik prilikom testiranja, omogućavajući korisnicima da razviju i testiraju svoje aplikacije bez potrebe za fizičkim dronom.

Međutim, ova integracija dolazi s određenim izazovima. Prije svega, složenost postavki može biti visoka, posebno za nove korisnike koji se moraju upoznati s oba sustava. Također, može doći do latencije u komunikaciji između ROS-a i Ardupilot-a, što može utjecati na pravovremenost komandi i povratnih informacija. Uz to, ne svi Ardupilot hardveri podržavaju sve mogućnosti ROS-a, što može ograničiti performanse.

U konačnici, Ardupilot može slati povratne informacije natrag ROS sustavu putem MAVLink-a, uključujući podatke o položaju, brzini, visini i stanju baterije. Ova integracija omogućuje implementaciju algoritama za autonomno donošenje odluka, čime se dodatno proširuju mogućnosti dronova.



Slika 3.1. Prikaz rada drona pri radu s ROS-om

## 4. OPIS HARDVERSKOG SKLOPOVLJA

Dron se sastoji od niza ključnih hardverskih komponenti koje zajedno omogućuju njegovo funkcioniranje, upravljanje te izvršavanje autonomnih i manualnih zadataka. Svaka komponenta ima svoju ulogu u sustavu od pogonskih do senzorskih jedinica koje omogućuju orijentaciju u prostoru i prikupljanje podataka iz okoline. [16]

HARDVERSKE KOMPONENTE:

### 1. OKVIR

Osnovna struktura je okvir drona na kojoj su montirane sve komponente. Izrađen je od karbonskih vlakana ili laganih aluminijskih legura zbog svoje visoke čvrstoće i male težine što je ključno za održavanje leta. Okvir pruža stabilnost i čvrstoću prilikom leta, a ujedno osigurava pravilno pozicioniranje svih dijelova poput motora, kontrolera leta i senzora.

### 2. MOTORI I PROPELERSKI SUSTAV

Dron se sastoji od četiri električna motora (jedan za svaki propeler) koji zajedno tvore kvadrokopter konfiguraciju. Motori bez četkica (eng. *brushless motors*) su izabrani zbog njihove učinkovitosti i dugog vijeka trajanja. Svaki motor je povezan s propelerom te uz pomoć elektroničkog regulatora brzine (eng. *ESC*) omogućuje preciznu kontrolu brzine i smjera rotacije, što je ključno za stabilan let. Korištenjem razlike u brzini rotacije pojedinih propelera dron se može kretati u različitim smjerovima i stabilizirati u letu.

### 3. KONTROLER LETA

Kontroler leta je glavni dio drona te se koristi Matek H743 Slim V3 pločica. To je napredni kontroler dizajniran za upravljanje motorima, obradu senzorskih podataka i za izvršavanje autonomnih funkcija. Matek koristi Ardupilot koji omogućava napredno planiranje leta, stabilizaciju, kontrolu visine i navigaciju. Povezan je s globalnim sustavom pozicioniranja (GPS) i sensorima osiguravajući pouzdano autonomno letenje.

### 4. RASPBERRY PI 3

Dron koristi Raspberry Pi 3 kao ključnu procesorsku jedinicu koja omogućava rad robotskog operacijskog sustava (ROS). Raspberry Pi služi za obradu podataka s različitih senzora, komunikaciju s kontrolerom leta te upravljanje kamerom i ostalim perifernim uređajima. Kroz



Raspberry Pi, dron može izvršavati autonomne zadatke, obrađivati slike i videozapise s kamere te komunicirati s vanjskim sustavima putem bežičnih mreža ili telemetrije.

## 5. RASPBERRY PI KAMERA

Na dronu se nalazi i kamera koja omogućuje snimanje slika i videozapisa. Kamera je integrirana u sustav kako bi dron mogao obavljati zadatke kao što su snimanje fotografija ili videozapisa. ROS omogućuje kontrolu kamere putem programskih modula, čime je omogućen autonomni nadzor i analiza okoline u realnom vremenu.

## 6. GLOBALNI SUSTAV POZICIONIRANJA (GPS)

Za preciznu navigaciju koristi se globalni sustav pozicioniranja koji omogućuje određivanje pozicije u prostoru. Podaci se koriste za autonomno letenje prema definiranim koordinatama te za praćenje rute leta. Uz pomoć ROS-a i GPS-a, dron može letjeti na unaprijed definirane visine i udaljenosti te se vratiti na početnu točku (eng. *Return to launch -RTL*) u slučaju potrebe.

## 7. TELEMETRIJSKI RADIO

Omogućuje dvosmjernu komunikaciju između drona i kontrolnog centra na Zemlji. Putem modula, može se pratiti stanje drona u stvarnom vremenu, uključujući podatke o poziciji, brzini, stanju baterija i slično. Također, telemetrija omogućuje slanje naredbi tijekom leta te promjenu postavki leta.

## 8. PRIJEMNIK

Dron je opremljen prijemnikom koji omogućava komunikaciju između daljinskog upravljača i drona. Prijemnik prima signale i šalje ih kontroleru leta, koja potom upravlja motorima i ostalim komponentama na temelju primljenih naredbi. Ova je komponenta je ključna za manualno upravljanje dronom te omogućuje brzu reakciju na promjene u okruženju i prilagođavanje leta.

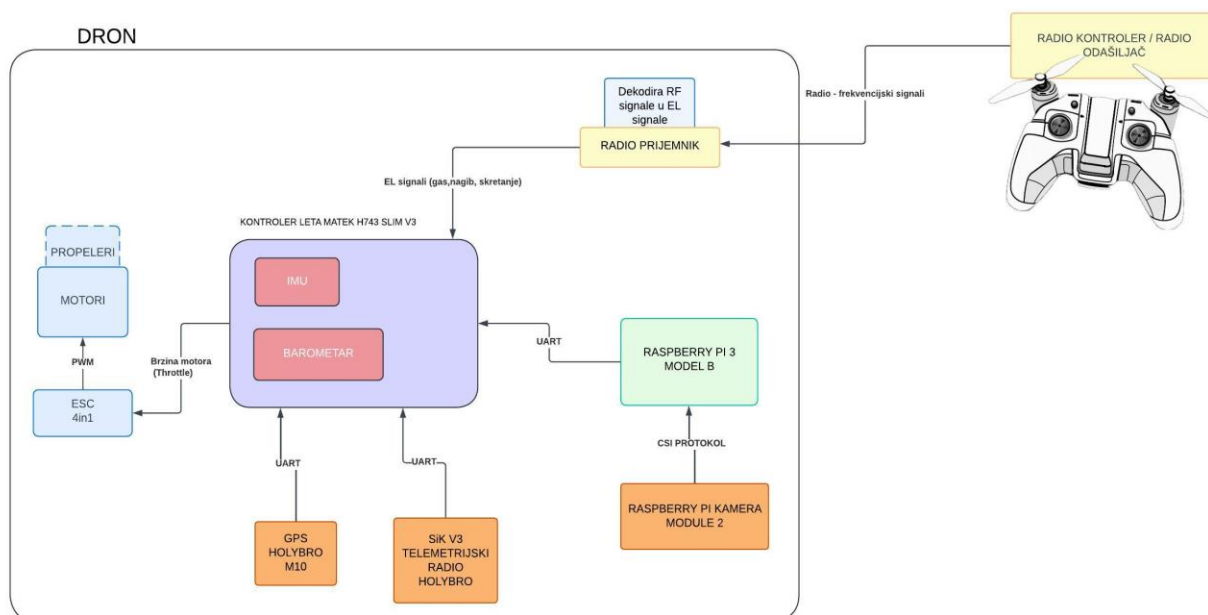
## 9. SENZORI

Dron je opremljen različitim sensorima poput barometra, akcelerometra i žiroskopa koji pružaju važne informacije o njegovom položaju, orijentaciji i visini. Ovi senzori omogućuju stabilizaciju leta i osiguravaju da dron održava zadanu visinu te ispravlja položaj u slučaju poremećaja kao što su vjetar ili turbulencije.

## 10. BATERIJSKI SUSTAV

Za napajanje svih komponenti, dron koristi litij-polimer bateriju koja osigurava visoku energetska gustoću i malu težinu, što je ključno za održavanje dugog trajanja leta. Baterija je povezana s kontrolerom napajanja, koji nadgleda potrošnju energije i omogućava siguran rad drona. [16]

### 4.1. Arhitektura hardvera



**Slika 4.1.** Grafički prikaz arhitekture hardvera

Slika 4.1. prikazuje hardversku arhitekturu komunikacije i povezivanja komponenti drona. Preko upravljača (kontrolac) operater šalje RF (radio-frekvencijske) signale koji sadrže komande za upravljanje dronom. Radio prijemnik prima RF signale i dekodira ih u EL (elektroničke signale) koji sadrže RC (eng. *remote control*) vrijednosti kanala. Dekodirane RC vrijednosti kanala šalju se na kontroler leta. Kontroler leta ima u sebi ugrađen IMU (Inercijska mjerna jedinica) koji mjeri ubrzanje, brzinu i orijentaciju te barometar koji mjeri visinu. Kontroler leta obrađuje podatke iz prijemnika i senzora te šalje PWM (*Pulse Width Modulation*) signale za kontrolu brzine motora preko elektroničkog regulatora brzine (ESC). ESC kontrolira brzinu motora koji pogone propeleri. Brzina motora se podešava prema PWM signalima kako bi dron mogao letjeti, manevrirati i održavati stabilnost.

Raspberry Pi kao računalo prima podatke od kontrolera leta putem UART veze. Također, Raspberry Pi povezan je s Raspberry Pi kamerom preko CSI protokola putem fleksibilnog kabela koja može snimati i prenositi slike i videozapise.

Raspberry Pi može koristiti WiFi i Bluetooth za bežičnu komunikaciju i prijenos podataka u realnom vremenu.

GPS modul (Holybro M10) koristi UART vezu za slanje podataka o lokaciji kontroleru leta. SIK radio (Holybro V3) također koristi UART vezu za bežičnu komunikaciju, što može omogućiti daljinsko upravljanje i nadzor drona.

Ukratko, dron funkcionira tako da operater koristi upravljač za slanje komandi koje prijemnik dekodira i šalje kontroleru leta. Kontroler leta obrađuje podatke zajedno s podacima iz IMU-a i barometra te šalje PWM signale motorima kako bi kontrolirao let. Raspberry Pi kao računalo prima dodatne podatke i može slati i primati informacije putem Wi-Fi i Bluetooth-a, dok GPS i SIK radio moduli omogućuju precizno pozicioniranje i dodatne komunikacijske mogućnosti.

## 4.2. Okvir drona

Okvir drona predstavlja temeljni strukturni element te ima ključnu ulogu u stabilnosti, aerodinamičnosti i nosivosti. Za potrebe ovog projekta odabran je okvir dimenzija 280x242x34 mm, težine 170 grama, koji zadovoljava potrebne zahtjeve u pogledu čvrstoće, lakoće i kompaktnosti.

Izrađen od čvrstog, ali laganog karbonskog materijala koji pruža optimalan omjer čvrstoće i težine, osiguravajući da dron bude stabilan i izdržljiv u različitim uvjetima letenja. Konstrukcija okvira je pažljivo dizajnirana kako bi se osigurala optimalna aerodinamičnost i minimalni otpor zraka tijekom leta.

Posebna pažnja posvećuje se rasporedu komponenti radi postizanja ravnoteže, stabilnosti i aerodinamičnosti same letjelice.



**Slika 4.2.** Okvir drona koji se koristi u radu

### 4.3. Motori

Motori su ključna komponenta u pogonskom sustavu dronova, jer omogućuju pokretanje propelera i osiguranje stabilnog leta. Postoje dva glavna tipa motora koji se koriste u dronovima: motori s četkicama (eng. *brushed*) i motori bez četkica (eng. *brushless*).

Motori s četkicama su starija tehnologija koja koristi četkice za prijenos struje na rotor. Međutim, zbog trošenja četkica, ovi motori imaju kraći vijek trajanja i manju efikasnost. S druge strane, motori bez četkica ne koriste četkice, što rezultira većom efikasnošću, dužim vijekom trajanja i boljim performansama.

Motori koji se koriste u radu su model motora bez četkica, Immersion RC 2004 1700kV, te se koriste kao pogonski sustav. Ovi motori su dizajnirani s fokusom na visoke performanse i pouzdanost, a njihova tehnička svojstva čine ih idealnim izborom za razne aplikacije.

Ključne karakteristike:

1. Dimenzije: Oznaka „2004“ označava promjer motora od 20 milimetara, dok visina iznosi 20,4 milimetara. Ove kompaktnosti omogućavaju jednostavnu integraciju u okvir drona.
2. Težina: Težina svakog motora iznosi 16,6 grama, dok ukupna težina za četiri motora iznosi 66,4 grama, što doprinosi smanjenju ukupne težine drona i povećanju efikasnosti.
3. Brzina: Oznaka „1700kV“ označava broj okretaja po voltu (RPM/V). Ovaj motor će postići 1700 okretaja u minuti za svaki volt napona, što rezultira visokim brzinama potrebnim za agilno letenje.



Slika 4.3. Motori 2004 1700kV

## 4.4. Propeleri

Propeleri su ključna komponenta autonomnih dronova jer omogućuju generiranje potisne sile potrebne za letenje. U ovom slučaju, koriste se Hurricane Props 1.5mm 2L2R 4023, visoko kvalitetne propelere dizajnirane posebno za bespilotne letjelice.

Ključne karakteristike:

1. Dimenzije: Svaki propeler ima promjer od 127 mm i debljinu od oko 1.5 mm. Ove dimenzije osiguravaju optimalan balans između snage i efikasnosti. Oznaka „4023” označava da je propeler dugačak 4 inča i (10.16 cm) širok 23 mm.
2. Konfiguracija: Ovi propeleri su dizajnirani s 2 listića (2L2R), što znači da koriste dva propelera za potisak i dva za povrat. Ova konfiguracija poboljšava stabilnost leta i manevarabilnost drona.
3. Materijal: Izrađeni su od visokokvalitetne plastike, polikarbonata. Ovaj materijal pruža optimalnu ravnotežu između čvrstoće, fleksibilnosti i lagane težine, što ga čini idealnim za upotrebu u bespilotnim letjelicama.
4. Otpornost na udarce: Plastika korištena za izradu ovih propelera otporna je na udarce i habanje, što povećava njihovu dugotrajnost i pouzdanost tijekom leta.
5. Lagana težina: S obzirom na to da su ovi propeleri lagani, smanjuju ukupnu težinu drona, što može poboljšati efikasnost i trajanje leta.
6. Aerodinamični dizajn: Hurricane Props 1.5mm 2L2R imaju aerodinamičan oblik koji smanjuje otpor zraka, čime se poboljšava ukupna performansa drona.
7. Jednostavna instalacija: Ovi propeleri lako se montiraju i demontiraju, što omogućava brzo održavanje i zamjenu.



**Slika 4.4.** Hurricane propeleri

## 4.5. Elektronički regulator brzine (ESC)

Elektronički regulator brzine (ESC) je ključna komponenta u bespilotnim letjelicama koja kontrolira brzinu rotacije motora. Njegova osnovna funkcija je primanje signala s kontrolne ploče (eng. *flight controller*) i prevođenje tih signala u odgovarajuće električne impulse koji reguliraju brzinu rotacije motora.

ESC djeluje kao posrednik između kontrolne ploče i motora, pretvarajući električne signale u brzinu rotacije motora. ESC omogućuje precizno upravljanje brzinom rotacije motora što omogućuje fino podešavanje letjelice tijekom leta.

ESC može sadržavati zaštitne mehanizme koji sprječavaju preopterećenje motora ili pregrijavanje, što osigurava sigurnost i dugovječnost komponenti.

ESC koji je korišten u samom radu je proizveden od strane hrvatske firme ORQA koji su dizajnirali 4-in-1 ESC, odnosno kombinirali su četiri ESC-a u jednu kompaktnu jedinicu. Kontinuirana struja je od 70A (po motoru), 80A tijekom prvih 10 sekundi. S visokom kontinuiranom strujom od 70A po motoru i impulsnom strujom od 80A tijekom prvih 10 sekundi, ovaj ESC pruža dovoljno snage za zahtjevne letove.

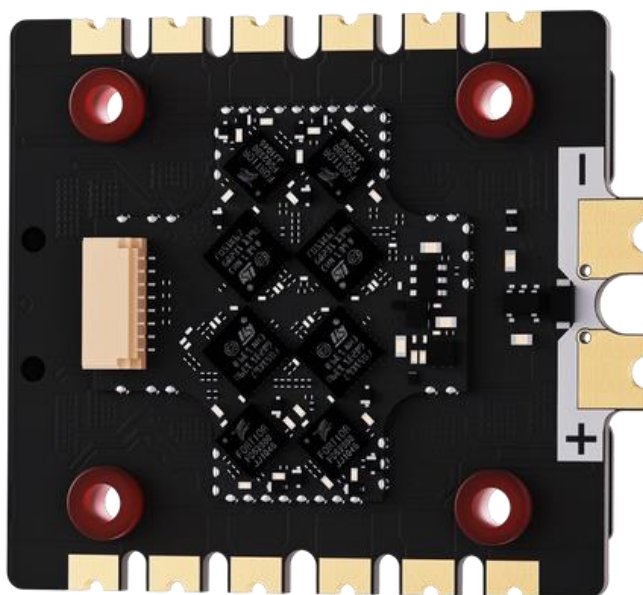
Ovaj tip ESC-a omogućuje upravljanje četiri motora s jednom pločom, što ga čini idealnim izborom za multikoptere ili kvadrokoptere poput drona s četiri motora.

Kompatibilan je s baterijskim paketima od 3S do 6S te nudi fleksibilnost u odabiru izvora napajanja za drona.

Opremljen je s BL- Heli32 firmver-om te tako ESC osigurava glatku i pouzdanu kontrolu motora za optimalne performanse.

Dizajniran je s standardnim montažnim rupama od 30 x 30 mm, težina mu je 28 gr što ga čini vrlo lakim, ali izdržljivim te minimizira dodatnu težinu samoga drona dok održava robusnost i pouzdanost.

ESC također na sebi ima četiri amortizirajuće silikonske gumice za smanjenje vibracija i osiguranje glatkog rada. [17]



**Slika 4.5.** Elektronički regulator brzine

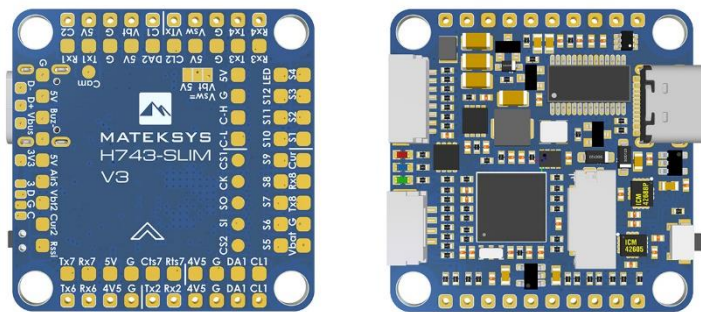
## 4.6. Kontroler leta

Kontroler leta je ključna komponenta u bespilotnim letjelicama koja omogućuje precizno upravljanje i stabilnost tijekom leta. Napredni kontroler leta Matek H743-SLIM V3 dizajniran je za bespilotne letjelice koje zahtijevaju visoke performanse i pouzdanost, pružajući pilota s alatima potrebnim za sigurno i kontrolirano letenje.

Ključne karakteristike:

1. Moćna procesorska jedinica: Opremljen je ARM Cortex-M7 procesorom koji omogućuje visoku brzinu obrade podataka i pouzdanu izvedbu.
2. Senzori: Kontroler uključuje napredne senzore kao što su inercijalna mjeračka jedinica (IMU), akcelerometar, žiroskop, barometar i kompas, što doprinosi stabilnosti i preciznosti upravljanja.
3. Dimenzije i težina: S dimenzijama od 37 x 37 mm i težinom od 11 grama, Matek H743-SLIM V3 je kompaktan, što ga čini idealnim za razne vrste letjelica, uključujući multikoptere i letjelice s jednim rotorom.
4. Telemetrija: Omogućuje dvosmjernu telemetriju putem radio veza, što omogućava praćenje i kontrolu letjelice u stvarnom vremenu.

5. Kompatibilnost: Kontroler podržava različite komunikacijske protokole poput PWM, SBUS i CAN, što omogućuje povezivanje s raznim motorima, sensorima, kamerama i drugim perifernim uređajima.
6. Softverska platforma: Matek H743-SLIM V3 je kompatibilan s različitim softverskim platformama za kontrolu dronova, uključujući BetaFlight, ArduPilot i CleanFlight. Ova fleksibilnost omogućava korisnicima prilagodbu i optimizaciju performansi drona prema vlastitim potrebama i preferencijama.[18]



**Slika 4.6.** Kontroler leta Matek H743 Slim V3

#### 4.7. Radio odašiljač

Radio odašiljači nalaze se u sustavima daljinskog upravljanja za bespilotne letjelice. Zadaća mu je prenošenje upravljačkih signala s ručnog daljinskog upravljača (odašiljač) na bespilotnu letjelicu (prijemnik).

Radio šalje naredbe poput povećanja/smanjenja brzine (gas), nagiba, skretanja i okretanja. Omogućuje korisniku sigurnosno kontroliranje drona. Radio odašiljač šalje kontrolne signale koristeći radio-frekvencijske valove (RF). Signale prima prijamnik drona, tumači naredbe i u skladu s tim upravlja mehanizmom leta.

U ovome projektu koristio se ImmersionRC Ghost UberLite Tx modul koji ima sustav radijskog upravljanja od 2,4GHz te nudi pouzdanu kontrolu na velikim udaljenostima i iznimno nisku latenciju, posjeduje raspon frekvencija od 2406 do 2479 MHz i koristi modulaciju Chirp Spread Spectrum + Adaptive FHSS za optimalnu stabilnost signala. Jedna od važnijih karakteristika je dvosmjerni protokol komunikacije koji osigurava pouzdanu vezu između prijemnika i odašiljača.

Napajanje odašiljača Ghost-a je 5V 1.75W kada je prijenosna snaga postavljena na 350mW, što osigurava pouzdan rad i trajanje baterije tijekom leta.



Uz svoje impresivne tehničke specifikacije i napredne značajke, Ghost se ističe kao vrhunski izbor za radio odašiljače za dronove, pružajući pilotima vrhunsku kontrolu i performanse. [19]



**Slika 4.7.** Radio odašiljač za RC kontroler

#### **4.8. Telemetrijski radio**

U ovome projektu se koristi SiK Telemetry Radio v3 koji je popularni modul za bežičnu telemetriju koji se koristi u dronovima i drugim bespilotnim letjelicama za komunikaciju između drona i zemaljske stanice koja će kroz rad biti detaljnije opisana.

Izabrani telemetrijski radio u ISM opsegu, na frekvencijama između 433 MHz ili 915 MHz, ovisno o regiji i zakonskim ograničenjima. Može podržavati brzine prijenosa podataka do 250 kilo bita po sekundi (kbps). U idealnim uvjetima, domet može biti nekoliko kilometara, ali stvarni domet ovisi o okolišu i postavkama snage.

Komunicira putem serijskog UART sučelja, što ga čini vrlo kompatibilnim s drugim kontrolerima leta i računalima kao što su Pixhawk i Ardupilot. Obično radi na naponu od 3.3V ili 5V.

Glavna funkcija SiK radija je prijenos telemetrijskih podataka između drona i zemaljske stanice, a to je Mission Planner. Podaci koji se dobivaju mogu uključivati informacije o statusu letjelice, u ovome slučaju drona, kao što su pozicija (GPS koordinate), visinu, brzinu, stanje baterije i drugi parametri.

Može omogućiti dvosmjernu komunikaciju, što znači da operater ne samo da prima podatke s drona, već može slati komande nazad na letjelicu. Ovo je korisno za daljinsku kontrolu i podešavanje parametara u realnom vremenu.

Omogućuje daljinsku nadogradnju firmvera na letjelici, dronu, što olakšava održavanje i poboljšanje performansi bez potrebe za fizičkim povezivanjem s dronom.

U samome povezivanju, jedan radio modul je povezan s računalom ili prijenosnim koji se koristi kao zemaljska stanica. Koristi software kao što su Mission Planner ili QGroundControl za primanje i prikaz telemetrijskih podataka.

Drugi modul je povezan s kontrolerom leta na dronu putem UART sučelja. Modul prima podatke od kontrolera leta i šalje ih zemaljskoj stanici i obratno.

SiK telemetrijski radio moduli su vrlo jednostavni za postavljanje i konfiguriranje, kompatibilni su s raznim software platformama za bespilotne letjelice te pruža veliku fleksibilnost u samome korištenju.

Uloga telemetrijskog radija u sustavu drona je ključna za osiguravanje prijenosa podataka između operatera i drona, što daje sigurnu i učinkovitu kontrolu letjelice, kao i praćenje njenog statusa u realnom vremenu. [20]



**Slika 4.8.** Holybro SiK V3 telemetrijski radio

## 4.9. Radio prijemnik

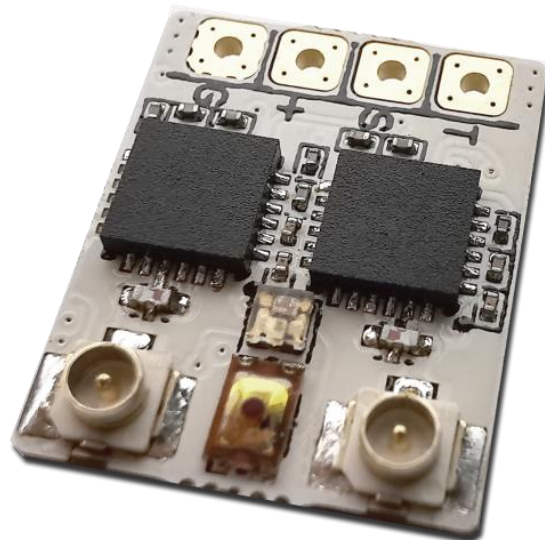
Radio prijemnik je uređaj koji hvata i dekodira radio signale koje emitira radio odašiljač. Prijemnik hvata radio signale, tumači ih i šalje kontroleru leta drona. Oni rade na principu prijema radio signala preko antene, nakon čega se signali obrađuju i dekodiraju kako bi se izvukle relevantne

informacije. Ova komunikacija omogućuje pilotima da kontroliraju pokrete drona, kao što su nagib, okretanje, ubrzavanje, skretanje itd.

U projektu koristio se ImmersionRC Ghost Atto Duo koji je inovativni prijemnik te nudi vrhunske performanse i pouzdanost. Ključna karakteristike Atto Duo prijemnika je njegova sposobnost da minimizira izgubljene kontrolne pokrete čak i pri izuzetno visokim brzinama i frekvencijama.

Prijemnik Atto Duo kombinira dva prijemna kanala i dvostruke antene što omogućava pouzdanu i konzistentnu vezu s odašiljačem.

Prijemnik Atto Duo koristi 2,4 GHz opseg za komunikaciju s odašiljačem što omogućuje brzu i preciznu kontrolu nad dronom. S end-to-end latencijom od samo 4ms, prijemnik pruža vrhunsku reakciju i odaziv što je vrlo važno za trkaće dronove gdje je svaka milisekunda presudna. [21]



**Slika 4.9.** Immersion Ghost Duo Atto radio prijemnik

## 4.10. Baterija

Baterija je dio svakoga drona, koja mu je zadaća pružanje potrebne energije za let. Postoji nekoliko vrsta baterija koje dronovi mogu koristiti, uključujući litij-polimer (LiPo), litij-ionske (Li-ion), nikal-kadmij (NiCd) i nikal-metal hidrid (NiMH). Najčešće se koriste litij-polimer (LiPo) baterije zbog visoke energijske gustoće, male težine i vrhunskih performansi.

Litij-polimer baterije se označavaju po ključnim karakteristikama, kao što su napon, kapacitet, C-rating i broj ćelija. U ovom projektu koristi se baterija FunFly 4S, kapaciteta 1300 mAh i 100C. Ova oznaka ukazuje na to da baterija ima četiri ćelije, kapacitet od 1300 mAh i C-rating od 100.

C-rating predstavlja maksimalnu brzinu pražnjenja baterije u odnosu na njen kapacitet. Ovaj broj označava koliko brzo baterija može isprazniti energiju bez oštećenja. Formula za izračunavanje maksimalne struje pražnjenja temelji se na C-ratingu i kapacitetu baterije.

Na primjer, za bateriju kapaciteta 1300 mAh i C-ratinga 100C, maksimalna struja pražnjenja iznosi 130 A (100 x 1,3 A), što znači da baterija može kontinuirano pružati tu količinu struje tijekom pražnjenja. Veći C-rating omogućava brže pražnjenje, što je korisno u aplikacijama gdje je potrebna veća snaga, poput dronova.

Kada je riječ o izračunima trajanja baterije, postoji nekoliko faktora koji utječu na to, poput kapaciteta baterije, C rating-a, opterećivanje motora i stil letenja. Jedan od najčešćih načina izračunavanja trajanja baterije prikazan je formulom:

$$\text{Trajanje baterije (min)} = \frac{\text{Kapacitet baterije (mAh)}}{\text{Potrošnja struje (mA)}} \times 60 \quad (4-1)$$

Na primjer, ako imamo bateriju od 1300mAh i letimo s dronom koji troši 20A struje, izračun bi bio:

$$\text{Trajanje baterije (min)} = \frac{1300 \text{ mAh}}{2000 \text{ mA}} \times 60 = 3.9 \text{ minuta} \quad (4-2)$$

Međutim, izračun trajanja baterije nikad ne mora biti točan jer stvarno trajanje baterije može varirati o mnogim čimbenicima poput uvjeta leta, temperature i efikasnosti samog pogonskog sustava.

## 4.11. Raspberry Pi kamera

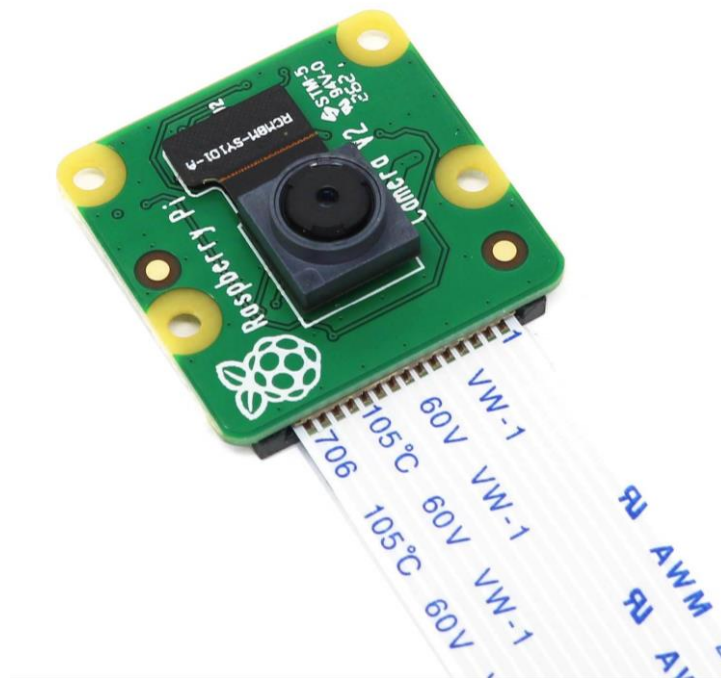
Raspberry Pi kamera module V2 od 8 MP je visokokvalitetna kamera koja je posebno dizajnirana za Raspberry Pi. Ova kamera omogućuje korisnicima da snimaju visokokvalitetne slike i videozapise direktno s Raspberry Pi platforme.

Spajanje između Raspberry Pi-ja i same kamere događa se pomoću fleksibilnog plosnatog kabela koji se sastoji od tankih vodiča i izolacijskih slojeva.

Za korištenje same kamere, omogućen je softverski paket koji omogućuje korisnicima da snimaju slike i videozapise, primjenjuju efekte na snimke te prilagođavaju postavke kamere.

Uporaba same kamere na dronu omogućava snimanje videozapisa ili slika iz zraka što korisnicima daje uvid u situaciju iz perspektive drona, prate okolinu ili snimaju prizore. Informacije slika i videozapisa koji se snimaju mogu se prenositi s Raspberry Pi-ja putem bežičnih komunikacijskih

tehnologija poput Wi-Fi-ja ili Bluetooth-a, što omogućuje korisnicima da prate u stvarnom vremenu na udaljenom uređaju poput pametnog telefona ili prijenosnog računala. [22]



**Slika 4.11.** Raspberry Pi V2 kamera module

## 4.12. Raspberry Pi 3 Model B

Raspberry Pi 3 je snažna platforma koje se koristi u raznim projektima čak i u robotskim primjenama. Njegova veličina varira o modelu, no niske je potrošnje energije i cjenovno pristupačan tako da je iznimno popularan izbor za razne projekte kao što je robotike, automatizacija, IoT (Internet of Things) itd.

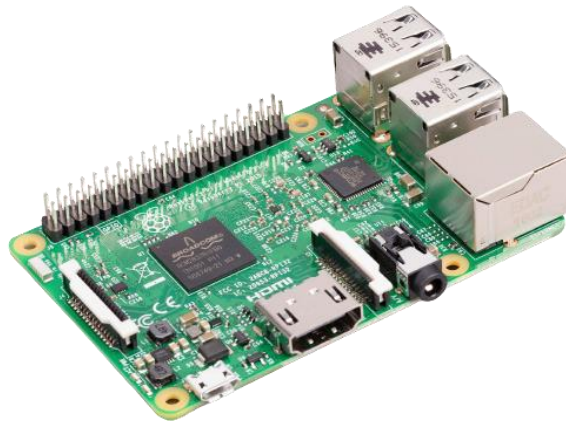
U ovom projektu Raspberry Pi ima ključnu ulogu u razvoju drona koji koristi ROS (Robot Operating System), posebno kada je u pitanju integracija kamere Raspberry Pi 8MP.

Ključne uloge:

1. Centralna jedinica upravljanja: Raspberry Pi djelovat će kao centralni mozak upravljanja dronom. Primit će podatke s senzora (GPS, žiroskop, kamera itd.) i obrađivat će ih koristeći ROS za navigaciju, stabilizaciju leta.
2. Obrada podataka: Raspberry Pi omogućava obradu podataka s kamere pomoću ROS-a te analizira okolinu poput prepoznavanje objekata ili smjerova kretanja te praćenje ili snimanje videozapisa/slika iz zraka.

3. **Komunikacija:** Raspberry Pi omogućuje komunikaciju s dijelovima drona poput kontrolera leta, aktuatora (motori, servo motori) i vanjskih uređaja. Putem ROS-a, Raspberry Pi može razmjenjivati podatke s komponentama i koordinirati njihovo djelovanje.
4. **Integracija s GPS-om:** Raspberry Pi prima podatke o lokaciji s GPS modula i koristi ih za navigaciju i kontrolu položaja drona.
5. **Kontrola kamere:** Raspberry Pi kontrolira kameru pomoću ROS-a, omogućavajući snimanje slike i videozapisa iz zraka. To uključuje kontrolu postavki kamere i upravljanje snimanjem.

Raspberry Pi igra ključnu ulogu u razvoju drona na ROS-u s kamerom jer pruža obradu podataka, kontrolu leta, komunikaciju s drugim komponentama drona te upravljanje i kontrolu same kamere. Kombinacija Raspberry Pi-a i ROS-a omogućava razvoj kompleksnih funkcionalnosti i algoritama za autonomno upravljanje dronom te pruža fleksibilnost za integraciju novih značajki. [23]



**Slika 4.12.** Razvojna pločica Raspberry Pi 3

### **4.13. Globalni sustav pozicioniranja (GPS)**

Globalni sustav pozicioniranja ima važnu ulogu u razvoju dronova, omogućujući dronovima preciznu navigaciju, autonomiju i stabilnost.

#### **1. Navigacija i pozicioniranje**

GPS omogućuje dronovima da samostalno lete prema zadanim koordinatama bez upotreba čovjeka. Ključnu ulogu ima kada su u pitanju misije poput mapiranja, dostave paketa i slično.

#### **2. Stabilnost i kontrola**

GPS je koristan ukoliko dron treba održati svoju poziciju u zraku, što je itekako važno za snimke i precizne operacije.

### 3. Praćenje i nadzor

GPS omogućuje operaterima da prate rutu drona u stvarnom vremenu. Svaki podatak dobiven od GPS koristi se za analizu letova, što pomaže u poboljšanju performansi i planiranju budućih misija.

### 4. Primjene

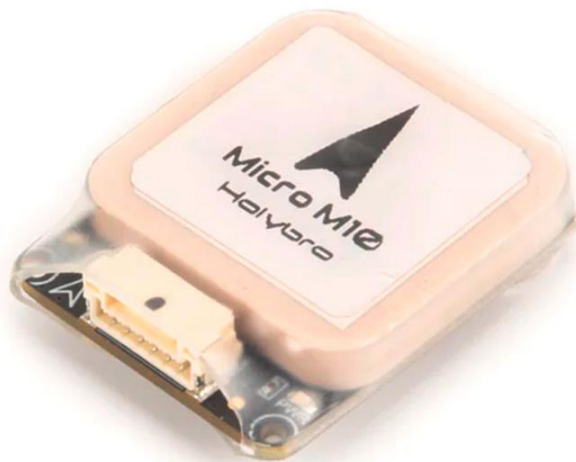
GPS se koristi za preciznu navigaciju za prikupljanje podataka o usjevima, tako da koristan za primjenu u poljoprivredi. Dronovi koriste GPS za navigaciju do određenih točaka u građevini, dalekovodima te omogućuje preciznu dostavu paketa na točnu adresu.

U ovome projektu model GPS koji se koristio je Holybro Micro M10 GPS koji je napredni GPS modula dizajniran za uporabu u dronovima i drugim autonomnim sustavima. U sebi ima IST8310 kompas, koji zajedno pružaju pouzdane i precizne podatke o poziciji i orijentaciji.

Navedeni GPS modul pruža brzo vrijeme inicijalizacije, što znači da se vrlo brzo može povezati s satelitima i pružati precizne podatke o lokaciji.

Ova vrsta modela GPS modula je idealna jer predstavlja odličan izbor za razvoj drona na ROS-u zbog lake integracije, nudeći precizne i pouzdane podatke o poziciji i orijentaciji, što omogućuje jednostavnu komunikaciju između GPS modula i drugih komponenti drona.

ROS podržava razne pakete za navigaciju i kontrolu, što olakšava implementaciju autonomnih funkcija pomoću GPS podataka. [24]



**Slika 4.13.** Holybro Micro M10 GPS

#### 4.14. DC-DC step down konverter (buck konverter)

DC-DC step down konverter poznatiji kao buck konverter, elektronički uređaj koji smanjuje napon ulaznog napajanja na niži, stabilan izlazni napon. To je vrsta preklopnog pretvarača koji koristi pulsno-širinsku modulaciju (PWM) za regulaciju izlaznog napona.

Buck konverter koristi prekidač, obično tranzistor, diodu, induktor i kondenzator. Kada je prekidač uključen, energija iz ulaznog izvora napaja induktor. Induktor pohranjuje energiju u obliku magnetskog polja. Kada se prekidač isključi, induktor oslobađa pohranjenu energiju, dioda se uključuje kako bi osigurala da struja teče prema izlazu. Kondenzator na izlazu filtrira napon, osiguravajući stabilan izlazni napon.

U ovome projektu koristio se DC-DC step down konverter čije su iduće specifikacije:

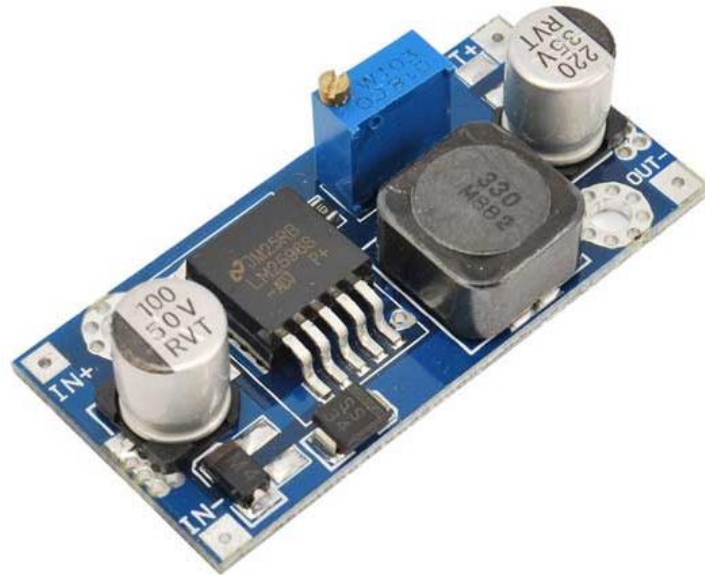
1. Ulazni napon (eng. *Input Voltage*): 3.2V – 40V
2. Izlani napon (eng. *Output Voltage*): 1.25V – 35V
3. Izlazna struja (eng. *Output Current*): max 3A
4. Učinkovitost pretvorbe (eng. *Conversion Efficiency*): do 92%
5. Izlazna valovitost (eng. *Output Ripple*): manje od 30mV
6. Frekvencija prebacivanje (eng. *Switching Frequency*): 65KHz
7. Radna temperatura (eng. *Working Temperature*): -45°C do + 85°C

Primjenu step down konverter je bilo omogućiti napajanje Raspberry Pi-u, koji kao mikroracunalo, zahtijeva stabilno napajanje od 5V. Korištenje buck konvertera omogućuje pretvaranje višeg napona baterije koji je u ovom slučaju 16.8V na stabilnih 5V koje je potrebno za napajanje. Ovakva konverzija je neophodna jer većina baterija poput Li-Po ili Li-ion imaju napon koji je između 3.7V i 4.2V po ćeliji.

Buck konverteri su učinkovitiji od linearnih regulatora, posebice kada se razlika između ulaznog i izlaznog napona povećava. Linearni regulator višak energija pretvara u toplinu, što je vrlo neefikasno i zahtijeva veće hladnjake za odvođenje topline. Buck moduli koriste preklopne metode koje su energetske učinkovitiji, prelazeći učinkovitost od 90%.



Korištenje buck konvertera omogućuje stabilno napajanje za Raspberry Pi, te ga čini idealnim za rješavanje mnogih projekta gdje je potrebno smanjenje napona iz baterije ili drugih izvora napajanja. [25]



Slika 4.14. DC-DC konverter

## 4.15. Radio kontroler

Radio kontroleri (RC) su ključna komponenta u upravljanju dronovima, omogućujući pilotima da komuniciraju s dronom i upravljaju njegovim letom na daljinu. RC sustavi za dronove obično koriste različite frekvencije i tehnologije za prijenos podataka između daljinskog upravljača i prijemnika na dronu.

Većina modernih RC sustava koristi frekvencije od 2.4 GHz što omogućuje stabilan i pouzdan prijenos podataka s minimalnim smetnjama. Maksimalna izlazna snaga od 350mW omogućuje veći domet i jači signal, što je ključno za kontrolu na većim udaljenostima.

Za ovaj projekt koristi se radio kontroler dizajniran od strane hrvatske firme s sjedištem u gradu Osijeku, Orqa, čiji radio kontroler koristi frekvenciju od 2.4GHz što znači da modul može komunicirati na velikim udaljenostima, pokrivajući raspon frekvencija od 2.4GHz. Koristi LoRa (eng. *Long Range*) tehnologiju koja omogućuje prijenos podataka na velikim udaljenostima s niskom potrošnjom energije. FLRC (eng. *Fast Long Range Communication*) je varijanta koja kombinira dug domet s bržom komunikacijom.

Ovaj RC ima frekvenciju osvježavanja signala od 500Hz, više frekvencija osvježavanja znači bržu i točniju reakciju drona na upravljačke komande. Antena je ugrađena u modul, što smanjuje potrebu za vanjskim komponentama i olakšava korištenje.

Maksimalna izlazna snaga je 350mW što ukazuje na jačinu signala koju modul može emitirati. Snaga od 350mW je relativno visoka snaga za RC module, što omogućuje veći domet i bolju penetraciju signala kroz prepreke.

Ovaj RC modul nudi kombinaciju dometa, brzine, praktičnosti i snage, što ga čini izuzetno efikasnim za kontrolu dronova u raznim uvjetima. [26]



**Slika 4.15.** Radio kontroler dizajniran od strane hrvatske firme Orqa

## 5. HARDVERSKA KONFIGURACIJA DRONA

Hardverska konfiguracija drona konfigurira se u Mission Planner-u. Prilikom konfiguracije potrebno je sve potrebne komponente spojiti i zalemiti na kontroler leta. Pristup kontroleru leta se radi putem USB-C kabla koji se direktno spaja na računalo.

Sama konfiguracija u Mission Planner-u ima nekoliko pododjeljka. Pododjeljci su tamo kako bismo postavili i konfigurirali svoj kontroler leta za daljnja testiranja.

Ukoliko već nije postavljen firmver, prvotno trebamo instalirati firmver Ardupilot na samu pločicu. Najnovija verzija ArduPilota se može pronaći na njihovim stranicama.

Prije upravljanja samim dronom potrebno je postaviti i odrediti sljedeće parametre:

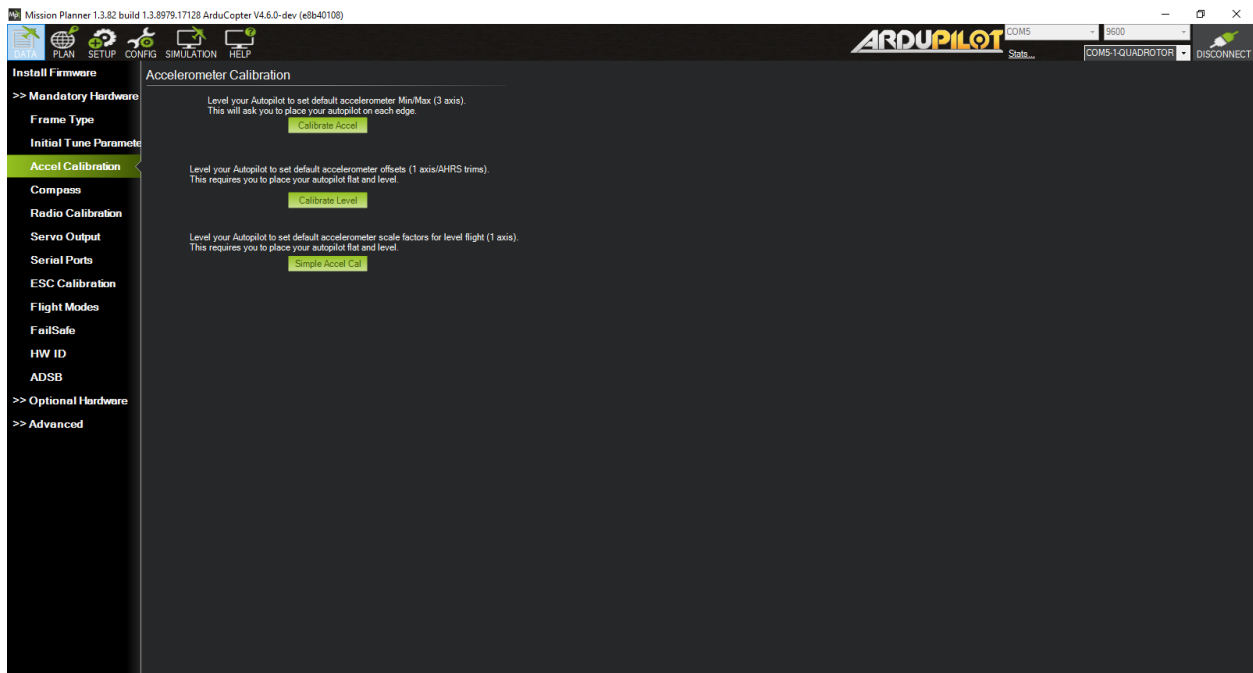
- Kalibracija akcelerometra i kompasa
- Radio kalibracija
- Test motora i rotacije propelera
- Definiranje načina letenja (manualno upravljanje)
- Postavljanje PID parametara za stabilan let

### 5.1. Kalibracija akcelerometra i kompasa

Akcelerometri se moraju kalibrirati kako bi se ispravili odstupni pomaci u sve tri osi, kao i bilo kakve varijacije izvan osi. Kalibracija akcelerometra je obavezna u ArduPilot-u i provodi se tako da se kontroler leta poveže putem USB-C na Mission Planner, nakon čega se izvršava kalibracija.

Pošto je pločica kontrolera leta u radu u ne-standardnoj orijentaciji (tj. okrenuta za 180 stupnjeva), bilo je potrebno da se ispravno postavi parametar AHRS\_ORIENTATION koji je ključan za stabilizaciju kako se pravilno očitati podaci o položaju i smjeru kontrolera leta, prije nego što se izvrši kalibracija akcelerometra.

Mission Planner zahtjeva prilikom kalibracije da se dron postavi na svaku os tijekom kalibracije. Kalibracijski položaji su horizontalno, desna strana, lijeva strana, nos prema dolje (eng. *nosedown*), nos prema gore (eng. *nosup*) i na leđima (eng. *backward*) te je vrlo važno da se prilikom kalibracije dron bude u mirnoj poziciji kako bi se uspješno provela kalibracija.

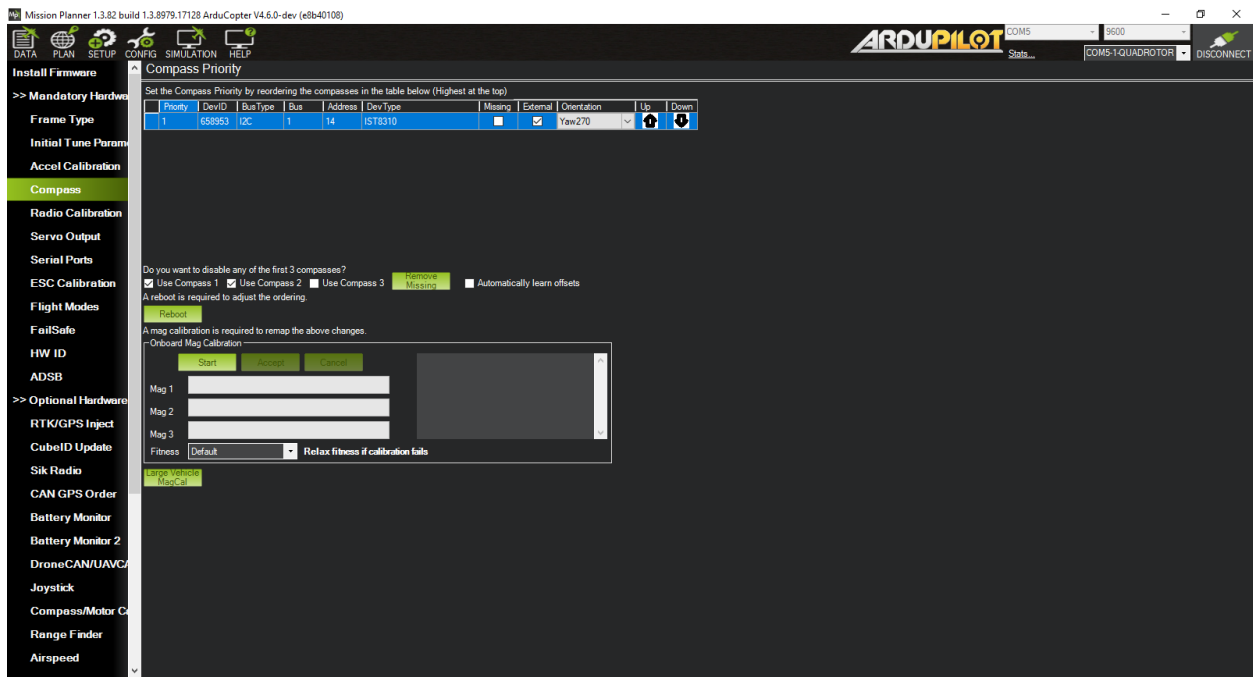


**Slika 5.1.** Prikaz kalibracije akcelerometra u Mission Planner-u

Prilikom kalibracija kompasu, potrebno je da prvotno bude dobro postavljen, te za bolju kalibraciju poželjno je na otvorenom vani izvršiti kalibraciju kompasu radi boljeg signala.

Prilikom kalibracije kompasu potrebno je osigurati prostor u kojemu u blizini nema metalnih objekata poput mobilnih telefona, metalnih stolova itd., jer će u suprotnome doći do netočne kalibracije.

Prilikom kalibracije kompasu potrebno je dron držati u zraku i rotirati ga tako da svaka strana (prednja, stražnja, lijeva, desna, gornja i donja) nekoliko sekundi pokazuje prema zemlji naizmjenično. Također je potrebno napraviti puni okret od 360 stupnjeva s svakim okretanjem usmjerenim prema različitim smjerovima i tako će se rezultirati šest punih okretaja kako bi provela i potvrdila sama kalibracija.



Slika 5.2. Prikaz kalibracije kompasa u Mission Planner-u

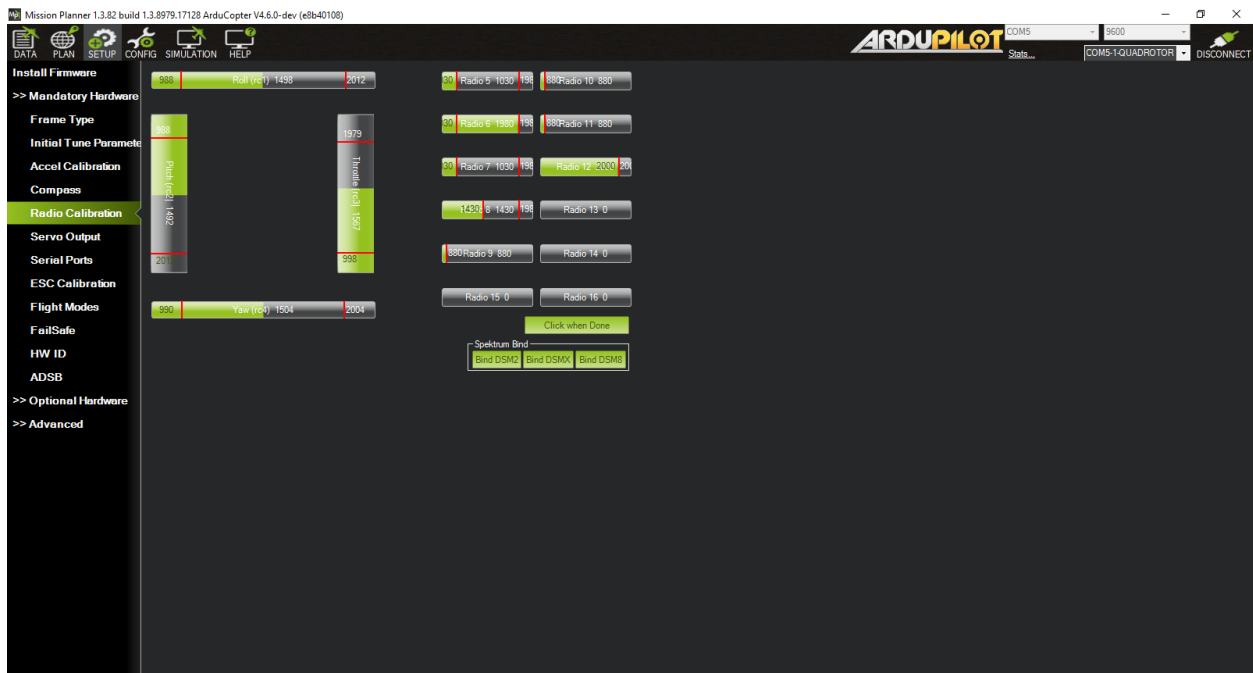
## 5.2. Radio kalibracija

RC odašiljači omogućuju korisniku da postavi način leta, kontrolira letenje i orijentaciju drona. Kako bi se svela kalibracija RC sigurnosno i točno, prvotno se treba osigurati da dron nije spojen na bateriju i spojiti kontroler leta na računalo koristeći USB-C kabel.

Kalibracija radio kontrolera u Mission Planner-u za Ardupilot je ključan korak u postavljanju i osiguravanju ispravnog rada dronova. Ovaj postupak omogućuje da kontroler ispravno interpretira ulaze od korisnika, što je važno za stabilnost i preciznost tijekom leta.

Priprema za postupak kalibracije:

- Dron treba biti isključen, ali povezan putem USB-a na računalo
- Prekidači da kontroleru moraju biti postavljeni u neutralne pozicije
- U kalibracijskom sučelju pojavljuju se stupci s prikazom trenutnih vrijednosti za svaki kanal
- Svaki kanal je u rasponu od -1000 do +1000 te se svaki kanal se pomiče lijevo - desno (eng. *yaw*), naprijed – nazad (eng. *pitch*), i tako za gas/brzinu (eng. *throttle*) te skretanje.
- Nakon pohranjenih podataka, potrebno je testirati dron na sigurnoj udaljenosti kako bi se utvrdila ispravnost postavljenih kontrola



Slika 5.3. Prikaz kalibracije radio kontrolera u Mission Planner-u

### 5.3. Test motora i propelera, definiranje način letenja (manualno upravljanje)

Prilikom testiranja motora utvrđuje se rad motora i njegove rotacije. Rotacija motora treba biti jednaka kao što je prikazano slikom jer je to standardizirano načelo postavljanja rada motora i njegove rotacije. Kao što je prikazano slikom, vidi se da dva motora se rotiraju u smjeru kazaljke na satu (eng. *clockwise*) i druga dva suprotno od smjera kazaljke na satu (eng. *counterclockwise*). Prilikom spajanja hardverskog sklopovlja, motori se spajaju na ESC pravilnim redoslijedom, tako da prva žica od motora ide na prvi pin on ESC-a , druga žica na drugi i tako dalje. Prilikom testiranja motora, odnosno rotacije, ukoliko je potrebno da motori se rotiraju u smjeru suprotnom od kazaljke na satu, tada je potrebno zamijeniti dvije žice od motora i zalemiti na ESC.



**Slika 5.4.** Prikaz testiranja rada motora u Mission Planner-u

Kopteri imaju 25 načina leta, od njih samo 10 se najviše koristi. Postoji modovi koji podržavaju različite tipove stabilizacije leta te se kontroliraju putem radija (putem prekidača na odašiljaču) putem naredbi ili korištenjem naredbi s zemaljske stanice (Mission Planner, QGround Control) ili pratećeg računala.

Tijekom manualnog testiranja drona, testirali su se tri načina letenja: stabilize, altitude hold i loiter načina.

Stabilize način letenja je jedan od osnovnih načina letenja u Ardupilot sustavu. U tom modu, dron omogućuje ručno upravljanje dok istovremeno automatski stabilizira svoju orijentaciju. To znači da će dron održavati horizontalni položaj i automatski se ispravljati ako ga nagnemo, ali visina i položaj se ne održavaju automatski - za to je potrebno ručno upravljanje.

Ovaj način je odličan za početnike jer omogućuje lakše ručno upravljanje, ali i za iskusne pilote koji žele preciznu kontrolu nad dronom. Ako pilot pusti sve kontrole, dron će se stabilizirati u horizontalnom položaju, ali visina i položaj neće biti održavani bez intervencije.

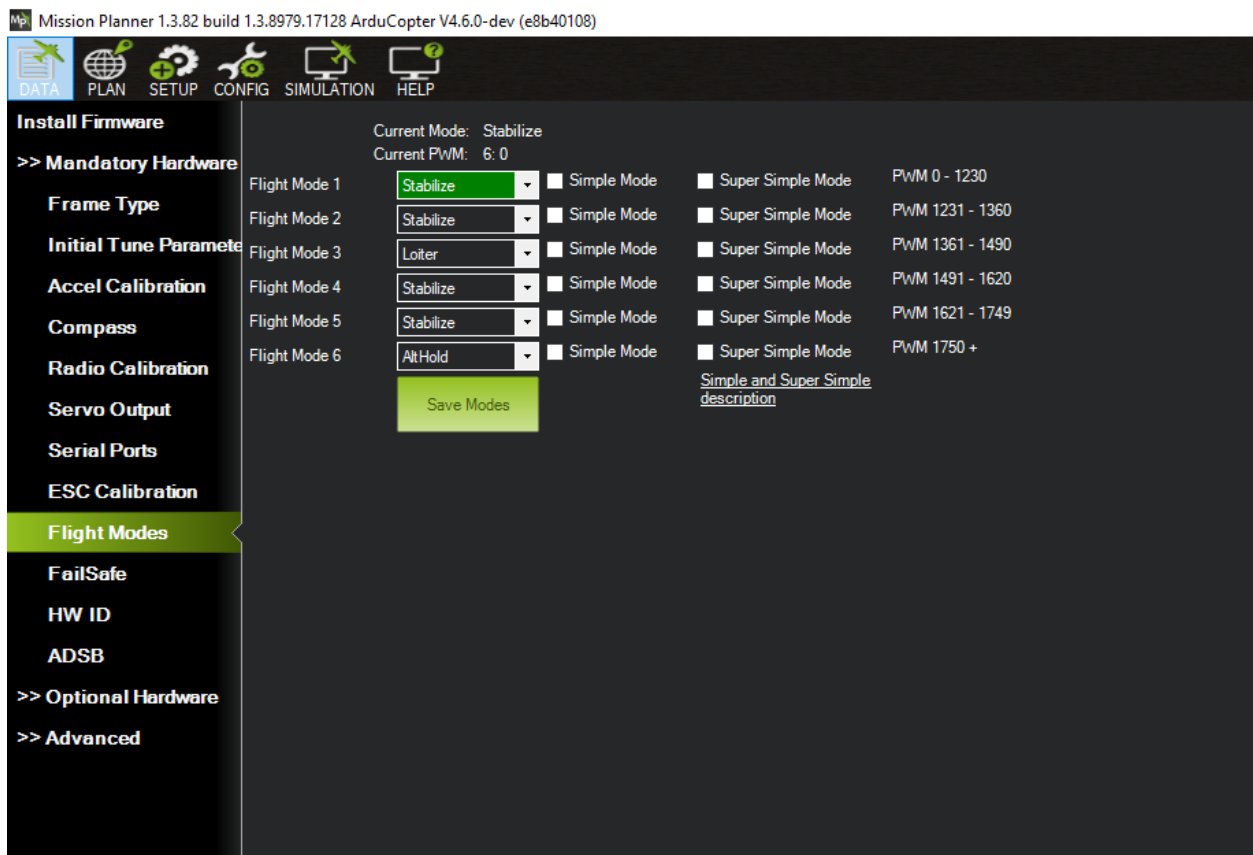
Altitude način letenja drona je u kojem dron automatski održava svoju trenutnu visinu, dok pilot i dalje ima ručnu kontrolu nad horizontalnom kretanjem i rotacijom oko vertikalne osi. U ovom modu, dron koristi barometar ili drugi senzor za visinu kako bi održao fiksnu visinu. Pilot može ručno promijeniti visinu pomoću gasa (eng. *throttle*), a kada gas ostane u srednjem položaju, dron će automatski zadržati trenutnu visinu.

Altitude hold je prijelazni način između potpuno ručnog letenja i naprednijih načina koji koriste GPS za potpuno automatsko pozicioniranje.

Loiter način letenja je napredni način letenja drona u kojem dron automatski održava svoju poziciju i visinu koristeći GPS, barometar i druge senzore. Ovaj način omogućava stabilno lebdenje na mjestu dok pilot zadržava osnovnu kontrolu nad dronom.

Dron automatski održava svoju GPS poziciju i visinu bez potrebe za stalnom ručnom kontrolom. Ako se pilot ne pomiče s kontrolama, dron će ostati točno na mjestu, lebdjet će u zraku. Ukoliko dron se pomakne zbog vjetera ili drugih vanjskih čimbenika, on će se automatski vratiti na svoju prethodnu GPS poziciju.

Loiter način letenja je vrlo koristan za fotografiju i videozapis iz zraka, jer omogućuje dronu da ostane stabilan dok pilot se može fokusirati na kameru i samo snimanje te je jedan od najsigurnijih i najstabilnijih načina letenja zbog korištenja vanjskih senzora poput GPS-a i barometra.



Slika 5.5. Prikaz postavljanja različitih načina leta u Mission Planner-u



## 5.4. Postavljanje PID parametara za stabilan let

PID (Proporcionalni, Integralni, Derivativni) regulatori koriste se za kontrolu različitih aspekata leta drona, poput stabilizacije, održavanja željene visine i preciznog manevriranja.

Funkcioniranje PID regulatora na drona:

1. Proporcionalni (P): Dio koji reagira na trenutnu pogrešku. Ukoliko je dron nagnut ili skreće s željene putanje, proporcionalni dio automatski prilagođava izlaznu snagu kako bi se pogreška umanjila.
2. Integralni (I): Dio koji se bavi akumuliranom pogreškom tijekom vremena. Ukoliko dron konstanto odstupa od željene vrijednosti (npr. stalno se naginje), integralni dio postupno povećava izlaznu snagu kako bi eliminirao trajno pogrešku.
3. Derivativni (D): Dio koji predviđa buduće odstupanje temeljem trenutne brzine promjene pogreške. Derivativna komponenta pomaže smanjiti prevelike reakcije koje bi mogle destabilizirati drona.

PID regulatori su korisni kako bi se osigurala stabilnost tijekom leta, brzo reagiranje na promjene i smanjenje odstupanja od željene pozicije ili putanje, čime zapravo osiguravaju preciznije upravljanje.



Slika 5.6. Prikaz parametara za PID regulator u Mission Planner-u



Slika 5.7. Poblži prikaz svih vrijednosti PID parametara

Prema slici 5.7. može se vidjeti kako dio za „stabilize roll” i „stabilize pitch” postavljeni na vrijednost 18 što predstavlja proporcionalni dio PID regulatora. Ove vrijednosti predstavljaju koliko brzo i snažno dron reagira na naginjanje (eng. *roll*) i nagib (eng. *pitch*). Veće *P* vrijednosti znače brži odgovor na pogrešku, ali također mogu povećati rizik od prekomjerne oscilacije ako su prevelike. Kod „stabilize yaw” vrijednost je postavljena na 10 što znači da će dron sporije reagirati na promjene u „yaw” osi u usporedbi s „roll” i „pitch”. Uobičajeno je da vrijednost „yaw” osi manja, jer nije toliko kritična za stabilnost leta.

Vrijednosti na „ACCEL MA ( Acceleration Maximum)” ograničavaju maksimalnu akceleraciju drona za zadane osi, pomažu pri održavanju stabilnosti i sprječavanju naglih pokreta.

Vrijednosti kod „rate roll” su 0.06 za *P* i *I* parametre te omogućavaju stabilnu kontrolu nad nagibom (eng. *roll*). Vrijednost *D* parametara iznosi 0.002 što rezultira sporijim smirivanjem, ali dobru reakciju na promjene.

Vrijednosti kod „rate pitch” su 0.08 za  $P$  i  $I$  parametre te su viši nego kod „rate roll” što pomaže u bržem odzivu na promjene u „pitch”. Vrijednost  $D$  parametara iznosi 0.0025. Vrijednost  $P$  kod „rate yaw” iznosi 0.100 što omogućava brže reakcije na promjene u smjeru kretanja. Vrijednost  $I$  je najniža i iznosi 0.010 te pomaže pri izbjegavanju grešaka te vrijednost  $D$  je iznosa nula, što znači da ne postoji korekcija na temelju brzine promjene „yaw”.

Vrijednosti „Throttle Accel” predstavljaju vertikalnu akceleraciju drona. Pomažu u kontroli polijetanja/slijetanja.  $P$  vrijednost je postavljena na 0.2 što znači da dron umjereno reagira na promjene u visini. Ovi parametri kontroliraju vertikalno kretanje drona.  $I$  vrijednost je postavljena na 0.250 te omogućuje dronu da održava željenu visinu čak i u prisutnosti vjetra.  $D$  vrijednost postavljena je na 0.000 što znači da derivativna komponenta nije uključena. Postavljena je na nula kako bi što brži odziv drona bio na promjene u visini bez oscilacija. odzivom drona na promjene u visini bez smanjenja oscilacija kroz derivativnu kontrolu.

Iz ovih parametara može se zaključiti da je dron postavljen na umjerenu stabilnost i kontrolu. Roll, pitch, i yaw imaju niske vrijednosti  $P$ ,  $I$ , i  $D$  komponenti, što ukazuje na to da je dron podešen za stabilan, ali ne pretjerano brz odziv. Ovi parametri daju dronu glatke pokrete s manje oscilacija, što je korisno za stabilno snimanje ili let u mirnim uvjetima.

## 6. SOFTVERSKI SKLOP DRONA

Softverski sklop drona oslanja se na naprednu arhitekturu koja omogućuje učinkovito upravljanje i komunikaciju između različitih komponenti sustava. Kontroler leta Matek H743 Slim V3 povezan je s Raspberry Pi-jem putem USB kabela, dok Raspberry Pi koristi ROS (Robot Operating System).

Pristup Raspberry Pi-u ostvaruje se putem SSH veze, a oba uređaja su spojena na istu Wi-Fi mrežu. ROS omogućuje razvoj složenih aplikacija i dvosmjernu razmjenu podataka između kontrolera leta i Raspberry Pi-a putem MAVLink protokola. Ova konfiguracija pruža temelje za autonomno upravljanje dronom, omogućavajući visoku razinu fleksibilnosti i funkcionalnosti tijekom izvođenja misija.

### 6.1. Detaljan opis softverskog sklopa

U razvoju autonomnih dronova, ključni aspekt je njihova softverska konfiguracija, koja omogućuje učinkovit rad i komunikaciju između različitih komponenti sustava. U ovom slučaju, dron se oslanja na kontroler leta Matek H743 Slim V3, koji je povezan s Raspberry Pi-jem putem USB kabela. Ovaj kontroler opremljen je naprednim sensorima, uključujući IMU, barometar i GPS, čime se osigurava precizno upravljanje i stabilizacija drona.

Raspberry Pi koristi SD karticu kapaciteta 32 GB na kojoj je instaliran operativni sustav Ubuntu 20.04.5 LTS (64-bit). Na Raspberry Pi je također instaliran ROS (Robot Operating System), što omogućuje razvoj i implementaciju složenih softverskih aplikacija za upravljanje dronom. ROS pruža široku paletu alata i biblioteka koje olakšavaju komunikaciju između različitih modula, što je posebno korisno za autonomne misije.

Pristup Raspberry Pi-u ostvaruje se putem SSH veze, koristeći terminal na računalu, gdje se unosi komanda „ssh pi@ip\_adresa“. Time se osigurava daljinska kontrola i administracija uređaja, a računalo i Raspberry Pi moraju biti spojeni na istu Wi-Fi mrežu kako bi se omogućila uspješna komunikacija. Ova konfiguracija omogućava korisnicima da instaliraju potrebne softverske pakete i izvršavaju skripte direktno s računala.

U okviru softverske konfiguracije, ROS radni prostor je postavljen za razvoj i pisanje koda u Python jeziku. Radni prostor omogućuje organizaciju projekata i upravljanje različitim ROS paketima koji su potrebni za funkcioniranje drona. Također su instalirani ključni ROS paketi, uključujući „mavros“, koji djeluje kao most između ROS-a i ArduPilot-a. Mavros omogućuje

slanje i primanje MAVLink poruka, što je vitalno za razmjenu podataka između kontrolera leta i Raspberry Pi-a.

U okviru ROS-a, podaci iz senzora, kao što su GPS i kamere, mogu se objavljivati kao ROS teme. Ostali moduli, poput algoritama za navigaciju ili stabilizaciju, mogu preplatiti na ove teme i donositi odluke temeljem primljenih informacija. Ova arhitektura komunikacije omogućuje jednostavnu integraciju dodatnih senzora i aktuatora, čime se unapređuje funkcionalnost drona.

Povezivanje Raspberry Pi kamere s Raspberry Pi-jem vrši se putem GPIO pinova, a upravljanje kamerom se može implementirati putem skripti, omogućavajući snimanje slika i videozapisa. Snimljeni materijali se pohranjuju na SD kartici, dok se istovremeno mogu slati u stvarnom vremenu na odredište, što je posebno korisno za analizu i autonomne misije.

Sveukupno, ova softverska konfiguracija omogućuje dronu učinkovito upravljanje, praćenje i analizu okoline u stvarnom vremenu. Integracija ROS-a s ArduPilot-om i povezivanje s različitim sensorima pruža temelje za razvoj naprednih autonomnih sustava, osiguravajući visoku razinu fleksibilnosti i funkcionalnosti tijekom izvođenja misija.

## 6.2. Implementacija ROS-a na Raspberry Pi 3

Implementacija ROS-a na Raspberry Pi uključuje nekoliko ključnih koraka, počevši od instalacije sustava i ROS-a pa sve do kreiranja čvorova i upravljanja cijelim sustavom.

Prvi korak je instalacija optimalnog operativnog sustava za rad s ROS-om na Raspberry Pi. Korištenjem alata Raspberry Pi Imager, na SD karticu od 32 GB se instalira Ubuntu Server 20.04.5 LTS (64-bit). Nakon odabira sustava i konfiguracije postavki, omogućeno je spajanje putem SSH-a. SD kartica se zatim vraća u Raspberry Pi, gdje se prijavljujemo u sustav i pomoću naredbe „*hostname -I*” dohvaćamo IP adresu uređaja za daljnji pristup preko računala.

Sljedeći korak je instalacija ROS-a. Preporučuje se verzija ROS Noetic zbog kompatibilnosti s Ubuntu 20.04. Nakon instalacije potrebno je postaviti ROS okruženje, instalirati dodatne alate i kreirati ROS radni prostor.

Nakon toga dolazi do stvaranja ROS paketa, pri čemu se koriste biblioteke kao što su „*std\_msgs*” za razmjenu podataka i „*rospy*” za Python implementaciju ROS čvorova.

Za potrebe autonomnog letenja koriste se paketi poput „*mavros*” za komunikaciju s ArduPilotom te „*nav\_msgs*” i „*sensor\_msgs*” za obradu podataka senzora.

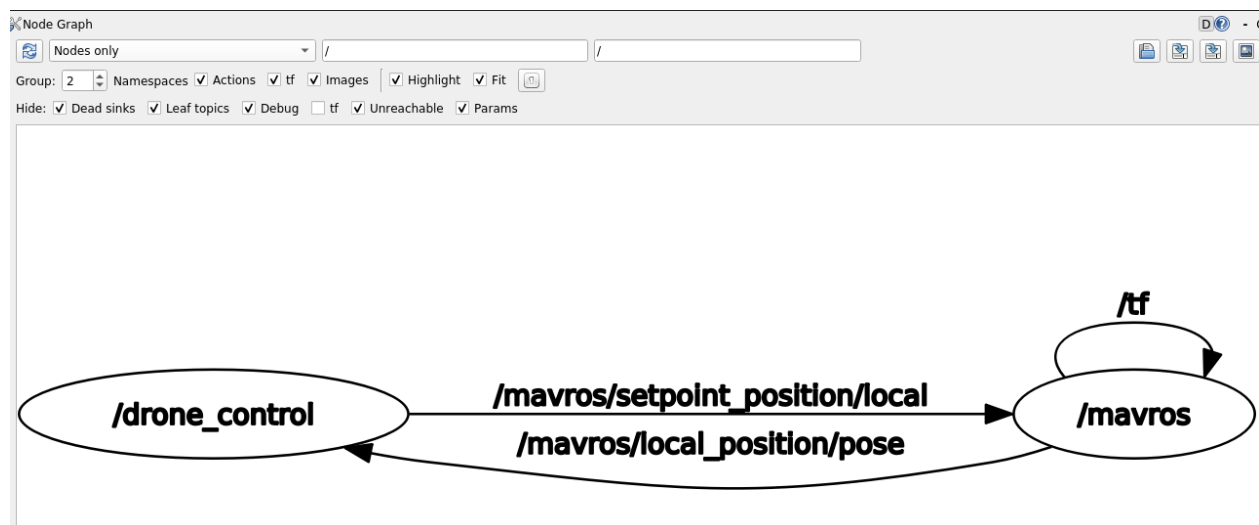
Kada je sustav u potpunosti postavljen, potrebno je provesti kalibraciju senzora, uključujući akcelerometar, žiroskop i kompas, te prilagoditi PID parametre kako bi se osigurao stabilan let.

### 6.3. Autonomno upravljanje dronom

Autonomno upravljanje dronom predstavlja složen sustav senzora, kontrolera leta, algoritama za obradu podataka i komunikacijskih protokola, koji omogućuju dronu da izvršava unaprijed definirane zadatke bez intervencije korisnika. Autonomija drona obuhvaća niz aspekata, uključujući planiranje putanje, stabilizaciju leta, izbjegavanje prepreka i analizu senzorskih podataka u stvarnom vremenu.

Autonomni sustav drona temelji se na višeslojnoj arhitekturi koja uključuje hardverske i softverske komponente:

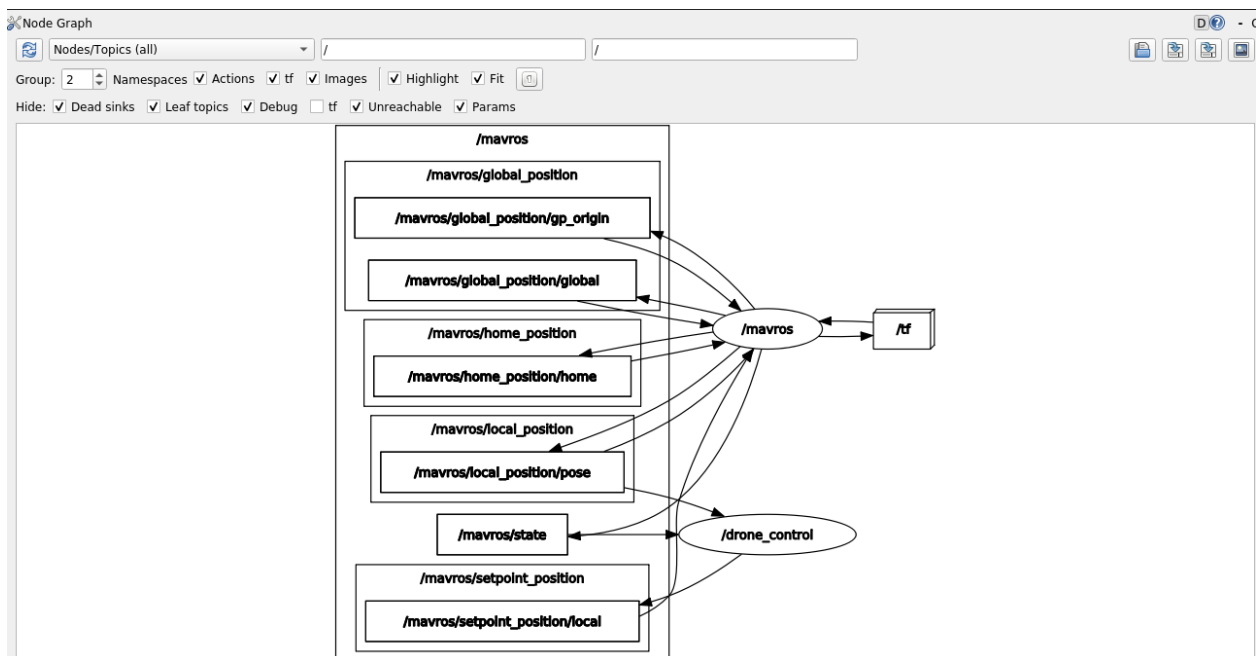
- **HARDVERSKI SLOJ:** uključuje senzore IMU, GPS, kamera, kontroler leta Matek H743 te prijemnik za manualno upravljanje. Svi podaci sa senzora šalju putem komunikacijskih protokola prema računalnoj jedinici (Raspberry Pi) ili kontroleru leta, ovisno o načinu rada.
- **SOFTVERSKI SLOJ:** uključuje ROS koji služi kao glavni komunikacijski okvir za autonomno upravljanje. Kroz ROS se obrađuju podaci sa senzora i kamera, a ovi se podaci koriste za donošenje odluka u stvarnom vremenu, planiranje putanje i kontrolu parametara.
- **KONTROLNI SLOJ:** obuhvaća kontroler leta i računalnu jedinicu (Raspberry Pi) koji provode autonomne naredbe na temelju ulaznih podataka. Kontroler leta je zadužen za osnovne funkcije letenja (visina, brzina, stabilizacija) dok ROS preuzima višu razinu autonomije.



Slika 6.1. Prikaz rqt grafa ROS čvorova

Na slici 6.1. nalazi se rqt graf koji prikazuje komunikaciju između dva ROS čvora: „drone\_control” i „mavros”. Čvor „drone\_control” odgovoran je za kontrolu drona te je zaslužan za pozicioniranje drona koristeći „mavros” za slanje tih komandi. Čvor „mavros” omogućava komunikaciju između ROS-a i drona putem MAVLink protokola. Tema „mavros/setpoint\_position/local” šalje poruke za određivanje željene pozicije drona. Mavros koristi tu temu kako bi postavio ciljnu poziciju drona u koordinatnom sustavu. Tema „mavros/local\_position/pose” koristi se za objavljivanje trenutne pozicije drona. Čvor „drone\_control” može se pretplatiti na tu temu kako bi dobio informacije o trenutnoj poziciji drona.

Tema „tf” se koristi za objavljivanje transformacija između koordinatnih sistema, povezana je s mavros-om što znači da mavros može koristiti transformacije između koordinatnih sustava za orijentaciju i pozicioniranje drona.



**Slika 6.2.** Prikaz rqt grafa svih ROS čvorova/tema

Na slici 6.2. prikazan je graf ROS čvorova i tema vezanih za kontrolu drona putem mavros-a. Glavni čvor „mavros” omogućuje komunikaciju između ROS-a i drona putem MAVLink protokola. Komunicira s različitim temam vezanim za poziciju, stanje i ciljne pozicije drona. Čvor „drone\_control” kontrolira drona i komunicira s „mavros” čvorom kako bi poslao ili primio podatke o poziciji i stanju drona.

Čvor „mavros” koristi više tema za komunikaciju:

1. „/mavros/global\_position” : tema koja obuhvaća informacije o globalnoj poziciji drona, koja može uključivati GPS podatke
2. „/mavros/global\_position/gp\_origin” i „/mavros/global\_position/global” : pod-teme koje pružaju detaljnije informacije o početnoj poziciji i trenutnoj globalnoj poziciji drona
3. „/mavros/home\_position”: pruža informacije o „home”, poziciji ili početnoj točki drona, na koju se može vratiti
4. „/mavros/home\_position/home”: pod-tema koja sadrži detalje o poziciji „home”
5. „/mavros/local\_position” : pruža informacije o trenutnoj lokalnoj poziciji drona
6. „/mavros/local\_position/pose” : koristi se za objavljivanje trenutne pozicije i orijentacije drona u koordinatnom sustavu
7. „/mavros/setpoint\_position”: tema koja se koristi za slanje ciljne pozicije drona
8. „/mavros/setpoint\_position/local”: pod-tema koja omogućava postavljanje lokalne ciljne pozicije
9. „/mavros/state” : tema koja objavljuje trenutni status drona i može uključivati informacije o letu, povezivanju
10. „/tf” : koriste se za razmjenu podataka između koordinatnih sustava

### **6.3.1. Navigacija i planiranje putanje**

1. **PREDEFINIRANE MISIJE:** podrazumijeva unaprijed određene putanje drona koje se definiraju prije leta, koristeći Mission Planner. Unutar Mission Planner-a, definiraju se točke, svaka točka je definirana koordinatama (geografska širina i dužina) te ima specifične naredbe, poput uzlijetanja, promjene visine, promjene brzine ili slijetanja. Dron slijedi točke koristeći GPS podatke. Kada su točke definirane, misija se prenosi na dron putem MAVLink protokola, dron preuzima naredbe i započinje misiju. Dron koristi kontroler leta i GPS za preciznu navigaciju između točaka. U ovom načinu rada, dron koristi svoje interne algoritme za održavanje stabilnosti leta i navigaciju.
2. **DINAMIČKO PLANIRANJE:** implementira ROS čvorove koji nadziru GPS podatke i upravljaju kretanjem drona prema unaprijed definiranim točkama. Kamera je integrirana kao ROS čvor za snimanje fotografija ili videozapisa tijekom leta. Iako s koristi samo GPS i kamera,



korištenje ROS-a omogućuje proširivanja sustava integracijom dodatnih senzora ili funkcionalnosti (npr. vizualno prepoznavanje objekata, praćenje objekata). Za pristup kameri, postavlja se ROS čvor koji prati GPS poziciju i određuje kada treba započeti ili zaustaviti snimanje.

Mission Planner je izvrstan alat za provođenje jednostavnih zadataka, poput osnovnog planiranja misija i upravljanja dronom. No, korištenjem ROS-a i nadogradnjom drona dodatnim sensorima, u budućnosti je moguće ostvariti naprednije funkcionalnosti poput izbjegavanja prepreka, prepoznavanja objekata i dinamičkog prilagođavanja putanje u stvarnom vremenu. Ovaj rad postavlja temelj za razvoj autonomnog drona, a integracija ROS-a pruža fleksibilnost i modularnost koja je ključna za daljnje istraživanje i napredak. Uz daljnju nadogradnju senzora i algoritama, može se stvoriti potpuno autonomni dron sposoban za složene zadatke u promjenjivim uvjetima.

### 6.3.2. Stabilizacija i kontrola leta

Stabilizacija drona tijekom autonomnog leta postiže se putem ugrađenih senzora, poput akcelerometra, žiroskopa i barometra koji omogućuju kontroleru leta da održava stabilnost u svim uvjetima.

**Akcelerometar:** mjeri ubrzanje u tri osi (x,y,z) i koristi se za detekciju promjena u brzini ili smjeru. Koristi se za određivanje orijentacije (npr. nagiba drona). Pomaže dronu u održavanju stabilnosti i određivanju smjera kretanja. Korištenjem podataka o ubrzanju, dron može prilagoditi svoje motore kako bi održao ravnotežu.

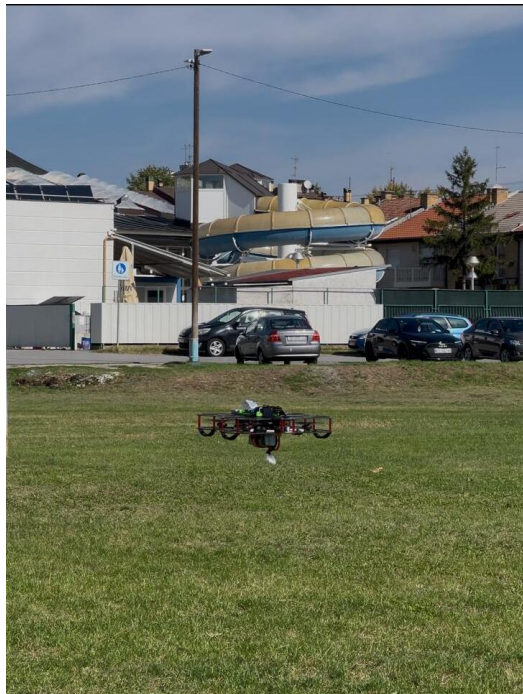
**Žiroskop:** mjeri kutnu brzinu ili rotaciju oko tri osi (eng. *roll, pitch, yaw*). Žiroskop pomaže u praćenju i kontroliranju rotacijskih pokreta drona. Pomaže dronu da održava stabilnost i precizno prati promjene u smjeru tijekom leta. Omogućuje precizne korekcije tijekom promjena položaja kako bi let bio stabilan.

**Barometar:** mjeri atmosferski tlak, a na temelju tih mjerenja može odrediti visinu iznad razine mora. Kako se tlak smanjuje s visinom, dron može koristiti barometar za održavanje određene nadmorske visine. Koristi se za kontrolu visine kako bi dron mogao precizno održavati željenu visinu tijekom leta, čak i ako se okolni uvjeti promjene.

## 7. TESTIRANJE I ANALIZA

Tijekom faze testiranja razmatrani su manualni i autonomni načini upravljanja dronom. Kod manualnog upravljanja, posebna pažnja posvećena je ispravnom povezivanju i lemljenju komponenti te hardverskoj konfiguraciji u Mission Planner-u. Ovaj postupak uključuje kalibraciju kompasa, akcelerometra, radio signala, te provjeru motora kako bi se osiguralo pravilno rotiranje.

Nakon uspješnog hardverskog postavljanja, dron je testiran u tri načina letenja: stabilan, alt hold i loiter način leta putem manualnog upravljanja. Tijekom manualnog testiranja, poseban je fokus da se ustanovi dali je kalibracija radio kontrolera i akcelerometra uspješno napravljena, te najvažnije kompasa, jer kompas ima glavnu ulogu u alt hold načinu leta i loiter načinu leta.



**Slika 7.1.** Prikaz testiranja drona na otvorenom manualnim upravljanjem

Nakon što se utvrdilo da su kalibracija akcelerometra, radio kontrolera i kompasa uspješno napravljene, dron se testira u autonomnom letu. Prilikom autonomnog testiranja, testiran je kod polijetanje/slijetanje kako bi se vidjelo ponašanje drona u zraku bez intervencije korisnika. Testirana su još dva koda, jedan u kojemu dron izvodi kvadrat 5x5 na visini od 10 metara te drugi u kojemu dron mijenja svoju brzinu i visinu. Kroz poglavlje prikazan je svaki kod te opisan.

Za autonomno testiranje drona putem ROS-a napravljena je .launch datoteka koja je prikazana na slici 7.2.

```

<launch>
  <!-- Launch MAVROS -->
  <include file="$(find mavros)/launch/apm.launch">
    <arg name="fcu_url" value="/dev/ttyACM0:921600"/>
  </include>

  <!-- Launch Drone Control Script -->
  <node pkg="drone_control" type="drone_control.py" name="drone_control" output="screen" />
</launch>

```

**Slika 7.2.** Prikaz .launch datoteke

Launch datoteka se koristi za pokretanje i konfiguriranje različitih čvorova u jednom potezu. Ona uključuje pokretanje „mavrosa” koji je most između ROS-a i Ardupilota te se koristi apm.launch što je unaprijed definira datoteka koja se koristi za povezivanje s kontrolerom leta. Postavljanjem „fcu\_url” argumenta se definira putanja do serijskog porta na koju je kontroler leta povezan, a „ttyACM0” je serijski uređaj. Postavljanjem brzina prijenosa podataka (eng. *baudrate*) na 921600 predstavlja određivanja koliko bita podataka se može prenijeti u sekundi.

Što je veća brzina prijenosa podataka, to će podaci brže putovati između uređaja. U ovom slučaju, 921600 bita u sekundi znači da se može prenijeti do 921600 bita podataka svake sekunde.

Drugi dio .launch datoteke pokreće Python skriptu drone\_control.py koja sadrži logiku za upravljanjem dronom. Paket u kojem se nalazi skripta je pod nazivom „drone\_control“ i njemu se nalazi „drone\_control.py“ Python skripta te je omogućeno da se ispisi iz programa ispisuju direktno na terminal.

Sva tri koda imaju zajedničku strukturu i koriste iste osnovne biblioteke. Razlikuju se samo u funkcionalnosti, s obzirom na to da su prilagođeni specifičnim zadacima koje dron treba izvršiti. Svaki kod proširuje osnovne funkcije kako bi se zadovoljili specifični zahtjevi misije, omogućavajući dronu da reagira na različite uvjete i izvršava različite zadaće učinkovito.

Kako bi se smanjio broj ponavljanja i izbjeglo nepotrebno objašnjavanje osnovnog koda više puta, dodatni dijelovi koji se razlikuju između kodova bit će posebno prikazani. Ova strategija omogućuje čitljivost i jasnoću.

Prvi kod predstavlja polijetanje/slijetanje drona na visinu od 10 metara, u kojemu dron nakon što dosegne zadanu visinu, prelazi u način slijetanja i vraća na početnu točku odakle je i poletio.

```
#!/usr/bin/env python3

import rospy
import subprocess
from mavros_msgs.msg import State
from mavros_msgs.srv import CommandTOL, SetMode, CommandBool
from geometry_msgs.msg import PoseStamped
import time

# Globalna varijabla za praćenje stanja drona
current_state = None

# Callback funkcija za ažuriranje trenutnog stanja drona
def state_cb(msg):
    global current_state
    current_state = msg

def main():
    # Inicijalizacija ROS čvora
    rospy.init_node('drone_control')

    # ROS pretplatnici i uslužni proxyji
    state_sub = rospy.Subscriber('/mavros/state', State, state_cb)
    set_mode_srv = rospy.ServiceProxy('/mavros/set_mode', SetMode)
    arming_srv = rospy.ServiceProxy('/mavros/cmd/arming', CommandBool)
    takeoff_srv = rospy.ServiceProxy('/mavros/cmd/takeoff', CommandTOL)
    land_srv = rospy.ServiceProxy('/mavros/cmd/land', CommandTOL)
```

**Slika 7.3.** Prikaz dijela koda za polijetanje/slijetanje

Na slici 7.3. prikazana je logika i povezivanje s ROS-om. Dodavanjem biblioteka i modula uspostavljamo komunikaciju s dronom. Dodavanjem „rospy” modula potrebno je radi korištenja ROS funkcionalnosti u Python-u. Dodavanjem „State” poruke iz „mavros\_msgs” paketa sadrži informacije o stanju drona, te dodavanjem „CommandTOL”, „SetMode” i „CommandBool” servisa može koristiti dron za polijetanje/slijetanje, promjenu načina rada drona. Callback funkcija „state\_cb” prima poruku *msg* tipa „State” te postavlja globalnu varijablu „current\_state” kako bi ažurirala trenutno stanje drona.

Prema slici možemo glavnu funkciju gdje se inicijalizira ROS čvor nazivom *drone\_control*. Kreira se pretplatnik:

- „state\_sub”: koristi „State” tip poruke, pretplatnik na /mavros/state i poziva „state\_cb” funkciju za obradu stanja drona

Kreira se i „ServiceProxy” za komuniciranje s različitim „mavros” servisima

- „set\_mode\_srv” : postavlja način rada drona
- „arming\_srv”: za pokretanje/naoružavanje (eng. *arming*) i gašenje/razoružavanje drona (eng. *disarming*)
- „takeoff\_srv” : izvođenje naredbe za polijetanje
- „land\_srv” : izvršavanje naredbe za slijetanje

```
# Osiguraj da su usluge dostupne s vremenskim ograničenjem
rospy.wait_for_service('/mavros/cmd/arming', timeout=10)
rospy.wait_for_service('/mavros/set_mode', timeout=10)
rospy.wait_for_service('/mavros/cmd/takeoff', timeout=10)
rospy.wait_for_service('/mavros/cmd/land', timeout=10)

rate = rospy.Rate(20) # 20hz

# Čekaj dok se stanje ne učita
while not rospy.is_shutdown() and current_state is None:
    rospy.loginfo("Čekanje na informacije o stanju...")
    rate.sleep()

# Čekaj na povezanost s FCU (Jedinica za kontrolu leta)
while not rospy.is_shutdown() and not current_state.connected:
    rospy.loginfo("Čekanje na povezanost s FCU...")
    rate.sleep()

# Armiraj dron
rospy.loginfo("Armiranje drona...")
if not arming_srv(True).success:
    rospy.logerr("Neuspješno armiranje drona!")
    return

# Postavi način rada na GUIDED
rospy.loginfo("Postavljanje načina rada na GUIDED")
if not set_mode_srv(custom_mode="GUIDED").mode_sent:
    rospy.logerr("Neuspješno postavljanje GUIDED načina rada!")
```

**Slika 7.4.** Prikaz dijela koda za polijetanje/slijetanje

Na slici 7.4. prikazan je kod u kojemu prije izvođenja ostatka koda, čekaju se svi servisi da postanu aktivni, odnosno dostupni te nakon toga čekamo povratno informaciju da se vidi u kojem stanju se trenutno nalazi dron te se uspostavlja komunikacija s FCU, s kontrolerom leta. Nakon uspješnog povezivanja, motori se aktiviraju i dron prelazi u „guided” način rada.

```

# Polijetanje na visinu od 10 metara
rospy.loginfo("Polijetanje na visinu od 10m")
if not takeoff_srv(altitude=10, latitude=0, longitude=0, min_pitch=0, yaw=0).success:
    rospy.logerr("Neuspješno polijetanje!")
    return

# Slijetanje drona
rospy.loginfo("Slijetanje")
if not land_srv(altitude=0, latitude=0, longitude=0, min_pitch=0, yaw=0).success:
    rospy.logerr("Neuspješno slijetanje!")
    return

# Isključi arming nakon slijetanja
rospy.loginfo("Deaktivacija drona nakon slijetanja.")
arming_srv(False)

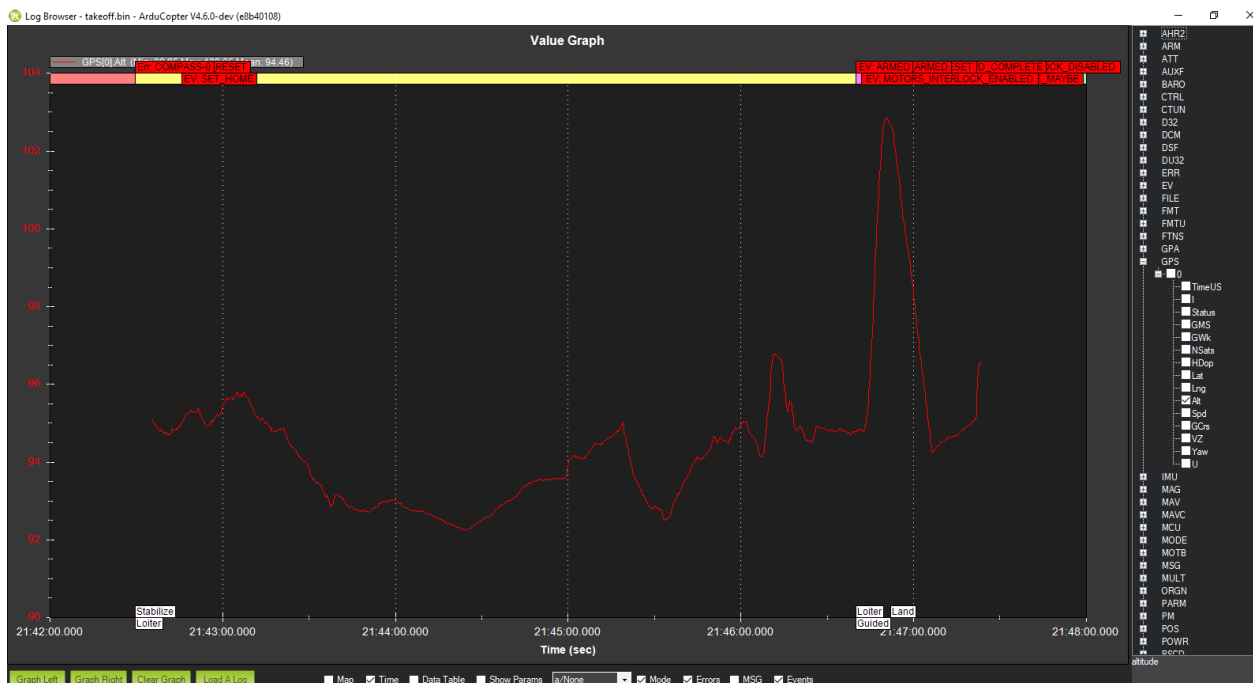
rospy.spin()

if __name__ == '__main__':
    try:
        main()
    except rospy.ROSInterruptException:
        pass

```

Slika 7.5. Prikaz dijela koda za polijetanje/slijetanje

Nakon što su se motori pokrenuli, dron polijeće na visinu od 10 metara te nakon toga slijeće nazad na tlo. Nakon slijetanja, dron deaktivira svoje motore.



Slika 7.6. Prikaz grafa za polijetanje/slijetanje drona

Na slici 7.6. prikaz je graf ponašanja drona prilikom testiranja prvoga koda za polijetanje/slijetanje drona na visini od 10 metara. Graf prikazuje stanja drona prije samog pokretanja koda. Na dnu grafa

napisana su stanja drona u pojedinom trenutku te se može vidjeti kako je početku dron u stabilizacijskom načinu rada, nakon toga prebačen u loiter način rada kako bi se provjerilo dali je GPS prikupio dovoljno satelita. Titranja na početku grafa se uzrokovana time što se dron premještao na poziciju na kojoj će započeti testiranje prvoga koda. Na grafu uočavamo kada i u kojem trenutku je dron prešao u „guided” način rada, te nakon toga poletio na visinu od 10 metara, kao što je i na grafu prikazano i sletio nazad na tlo i deaktiviramo motore, te samim ovim grafom je prikazana uspješnost prvog testiranog koda.

Drugi dio testiranja predstavlja polijetanje drona na visinu od 10 metara, prateći kvadratnu putanju dimenzija 5x5 metara. Tijekom leta, dron će koristiti kameru za snimanje i prijenos video zapisa u stvarnom vremenu na računalo, omogućujući korisniku da prati okolinu iz perspektive drona. Putanja će uključivati četiri glavne točke, a dron će se kretati kroz svaku od njih, održavajući visinu i stabilnost tijekom cijelog izvođenja zadatka.

```
#!/usr/bin/env python3

import rospy
from mavros_msgs.msg import State
from mavros_msgs.srv import CommandTOL, SetMode, CommandBool
from geometry_msgs.msg import PoseStamped
import tf
import math
import cv2
import threading
```

**Slika 7.7.** Prikaz dodanih osnovnih i dodatnih biblioteka

Za izvođenje drugog testa na osnovni dio koda, dodale su se biblioteke „import cv2” i „import threading”, „import math” te „import tf” za pristup kameri i pristup rada s višekratnim izvođenjem kao što je prikazano slikom 7.7.

```
# Callback funkcija za praćenje trenutne pozicije
def position_cb(msg):
    global current_position
    current_position = msg
```

**Slika 7.8.** Prikaz funkcije za praćenje pozicije

Slika 7.8. predstavlja dodanu funkciju u kodu koja ažurira globalnu varijablu „current\_position” Callback funkcija „position\_cb” prima poruku *msg* tipa „PoseStamped” te postavlja globalnu varijablu „current\_position” kako bi ažurirala trenutnu poziciju drona.

```

# Funkcija za slanje setpoint pozicija s yaw orijentacijom
def set_position_with_yaw(local_position_pub, x, y, z, yaw_degrees):
    goal_pose = PoseStamped()
    goal_pose.pose.position.x = x
    goal_pose.pose.position.y = y
    goal_pose.pose.position.z = z

    # Pretvaranje yaw iz stupnjeva u radijane i zatim u kvaternion
    yaw_radians = math.radians(yaw_degrees)
    quaternion = tf.transformations.quaternion_from_euler(0, 0, yaw_radians)
    goal_pose.pose.orientation.x = quaternion[0]
    goal_pose.pose.orientation.y = quaternion[1]
    goal_pose.pose.orientation.z = quaternion[2]
    goal_pose.pose.orientation.w = quaternion[3]

    local_position_pub.publish(goal_pose)
    rospy.sleep(3) # Čekaj na rotaciju i pomak na novu točku

```

**Slika 7.9.** Prikaz funkcije za slanje pozicije s orijentacijom

Slika 7.9. prikazuje funkciju koja služi za slanje pozicijskih i orijentacijskih naredbi dronu. Pomoću „local\_position\_pub” stvaramo objekt koji šalje naredbe za poziciju drona. Ciljane koordinate su x,y i z u metrima. Ciljni kut „yaw\_degrees” predstavlja horizontalnu rotaciju u stupnjevima, koji određuje smjer u kojem dron treba biti okrenut. Pošto ROS koristi radijane, morali smo konvertirati zadani „yaw” iz stupnjeva u radijane.

```

def camera_stream():
    cap = cv2.VideoCapture(0) # Pristup zadanoj kameri
    if not cap.isOpened():
        rospy.logerr("Ne mogu otvoriti kameru")
        return

    # Postavi željenu rezoluciju (npr.320x240)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)

    while not rospy.is_shutdown():
        ret, frame = cap.read()
        if not ret:
            rospy.logerr("Nije uspjelo dohvatiti sliku s kamere")
            break

        flipped_frame = cv2.flip(frame, 0)
        cv2.imshow('Stream kamere', flipped_frame)
        cv2.waitKey(1) # Potrebno za ažuriranje prikaza

    cap.release()
    cv2.destroyAllWindows()

```

**Slika 7.9.** Prikaz funkcije za prikazivanje video stream-a



Na slici je prikazana funkcija „camera\_stream” koja služi za prikazivanje video stream-a s kamere u stvarnom vremenu. Koristi se VideoCapture() funkcija za pristup kameri te provjerava je li kamera ispravna. Format, odnosno rezolucija snimljenog prozora je iznosa 320x240 piksela.

```
# Pokreni prikaz s kamere u zasebnoj dijelu
camera_thread = threading.Thread(target=camera_stream)
camera_thread.start()

# Postavljanje parametara za snimanje videa
fourcc = cv2.VideoWriter_fourcc(*'XVID') # Odabir kodeka, XVID je široko podržan
out = cv2.VideoWriter('square_video.avi', fourcc, 20.0, (640, 480))
```

**Slika 7.10.** Prikaz postavljanja parametara za snimanje videa

```
# 1. Pomak na prvu točku (5, 0, 10) s yaw kutom 0 stupnjeva
rospy.loginfo("Pomicanje na (5, 0, 10)")
set_position_with_yaw(local_position_pub, 5, 0, 10, 0)
rospy.sleep(3)

# 3. Rotacija na 90 stupnjeva i pomak naprijed za 5 metara
rospy.loginfo("Rotacija na 90 stupnjeva i pomak naprijed")
set_position_with_yaw(local_position_pub, 5, 5, 10, 90)
rospy.sleep(3)

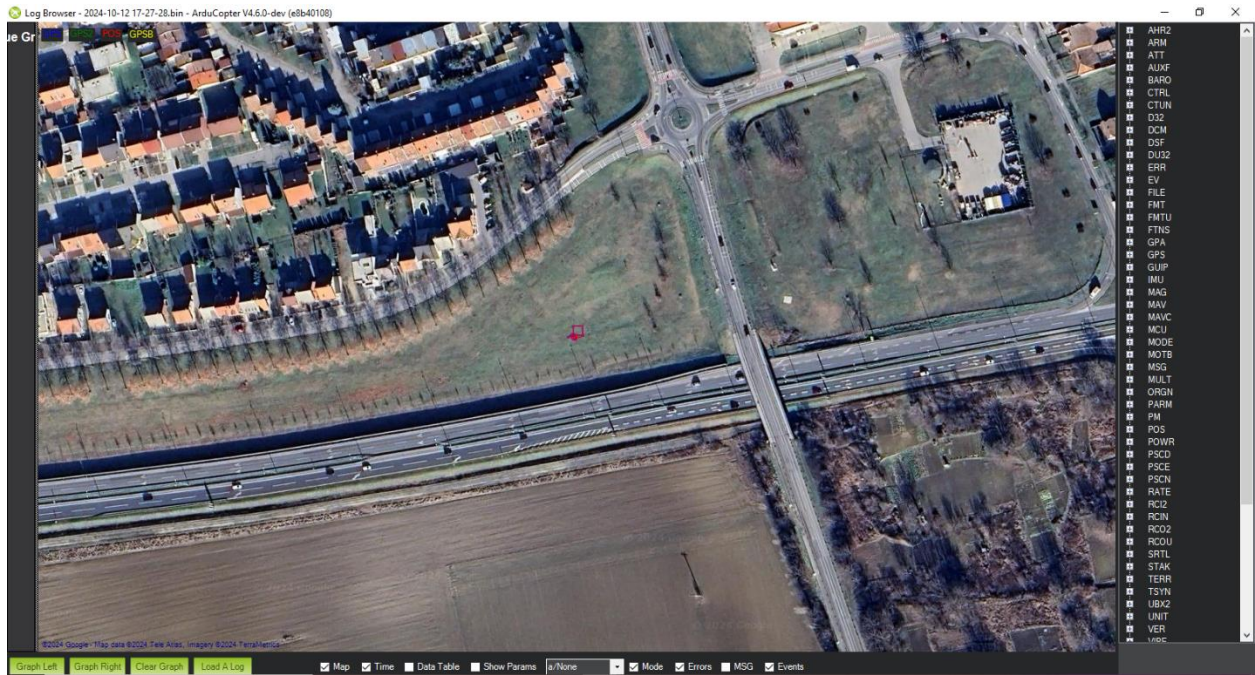
# 5. Rotacija na 180 stupnjeva i pomak naprijed za 5 metara
rospy.loginfo("Rotacija na 180 stupnjeva i pomak naprijed")
set_position_with_yaw(local_position_pub, 0, 5, 10, 180)
rospy.sleep(3)

# 7. Rotacija na 270 stupnjeva i pomak naprijed za 5 metara, vraćajući se na početnu točku
rospy.loginfo("Rotacija na 270 stupnjeva i pomak naprijed")
set_position_with_yaw(local_position_pub, 0, 0, 10, 270)
rospy.sleep(3)

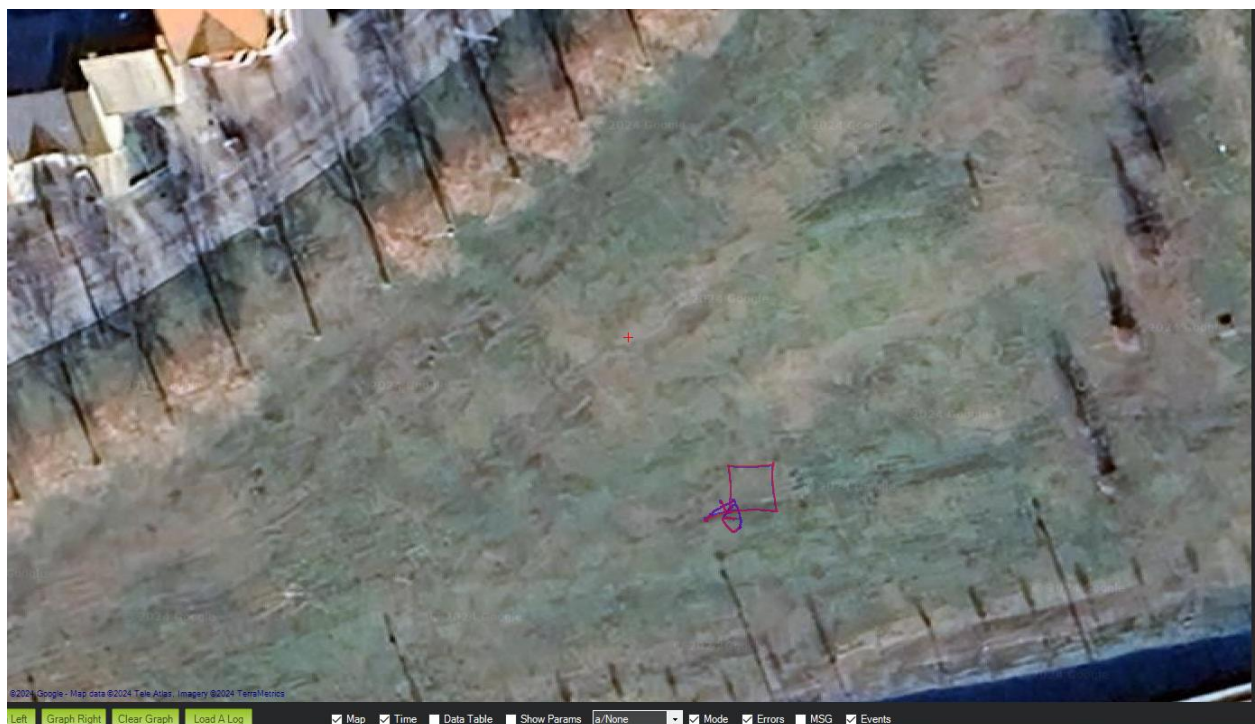
# Slijetanje drona
rospy.loginfo("Slijetanje drona")
land_srv(altitude=0, latitude=0, longitude=0, min_pitch=0, yaw=0)
```

**Slika 7.11.** Prikaz dijela koda za izradu kvadrata 5x5

Slika 7.11. prikazuje kod u kojemu dron leti i ima putanju izrade kvadrata dimenzija 5x5. Dron aktivira svoje motore i prelazi u „guided” način rada te uzlijeće na visinu od 10 metara, nakon toga dron se pomiče u četiri smjera kako bi formirao kvadrat. Prvo leti na koordinatu (5, 0, 10) s yaw kutem 0°. Zatim se rotira na 90° yaw i leti na koordinatu (5, 5, 10). Potom se rotira na 180° yaw i leti na koordinatu (0, 5, 10). Konačno, rotira se na 270° yaw i vraća se na početnu koordinatu (0, 0, 10). Na svakoj točki dron se rotira i leti prema zadanoj koordinati, čime ostvaruje točke kvadrata u zraku. Kada završi cijelu putanju, dron se spušta natrag na tlo koristeći servis za slijetanje.

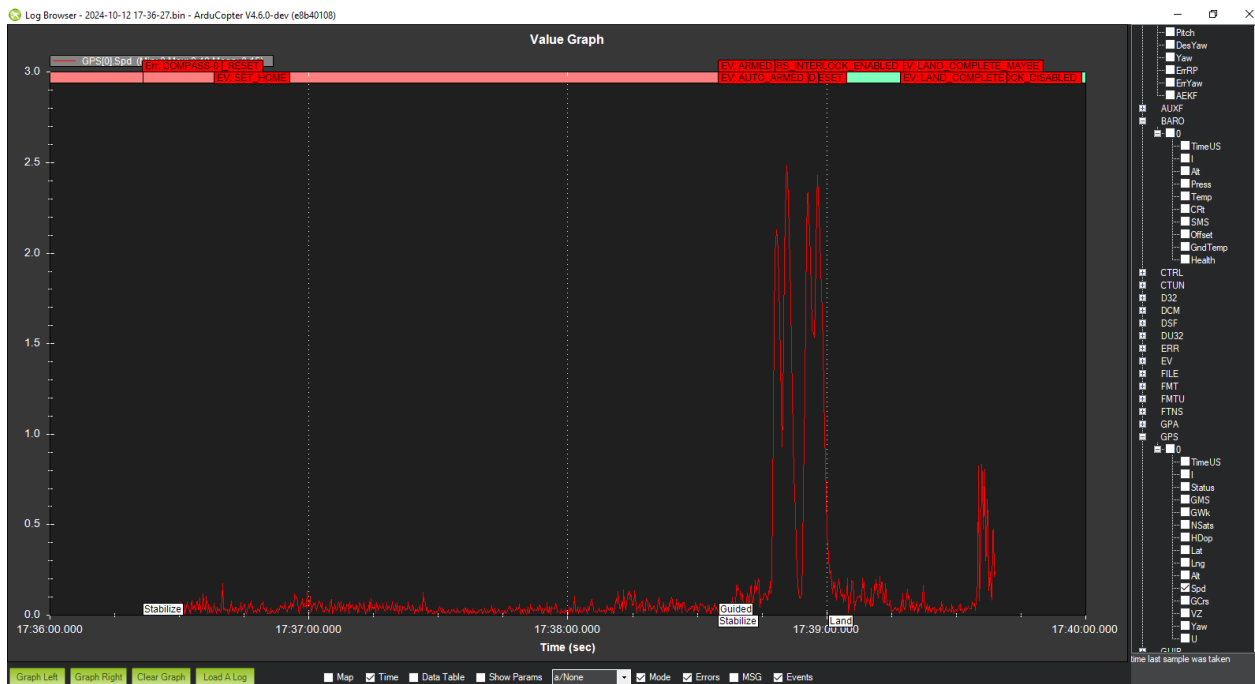


**Slika 7.12.** Prikaz izrade kvadrata 5x5



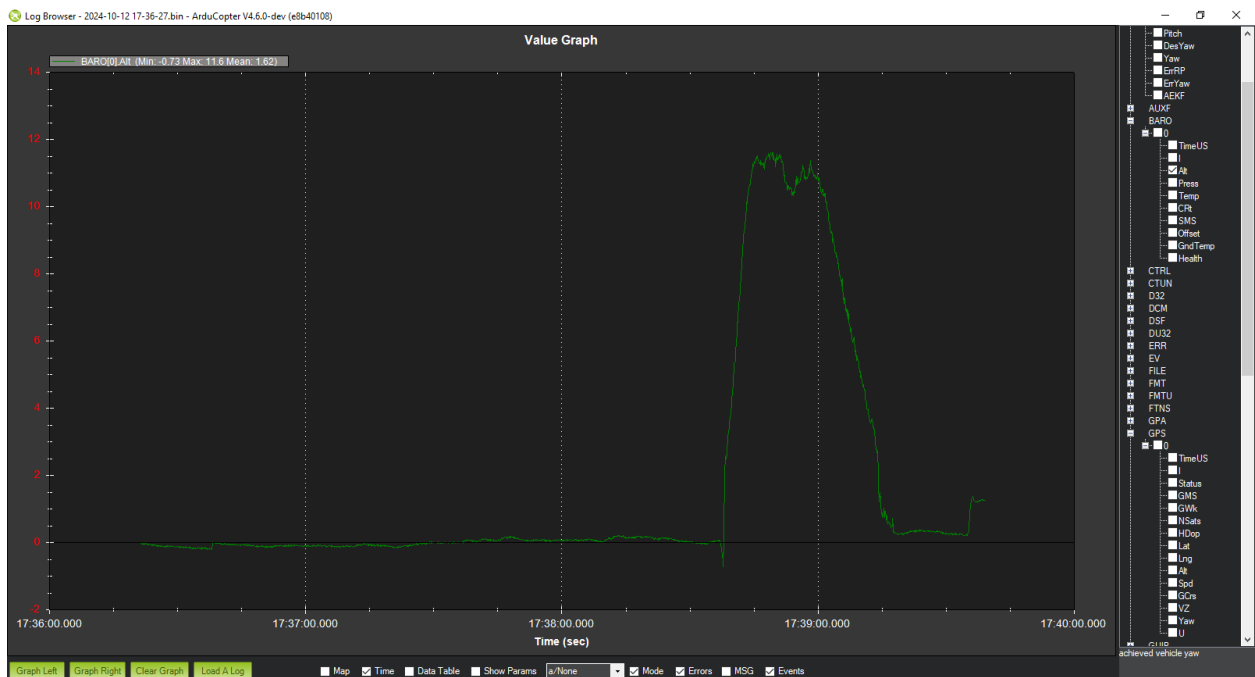
**Slika 7.13.** Povećani prikaz izrađenog kvadrata 5x5

Na slikama 7.12. i 7.13. može se vidjeti uspješnost izvođenja drugoga koda, u kojem se može uočiti na kojoj lokaciji je dron izvodio testiranje, te naravno uspješnost izvođenja pravilnog kvadrata dimenzija 5x5.



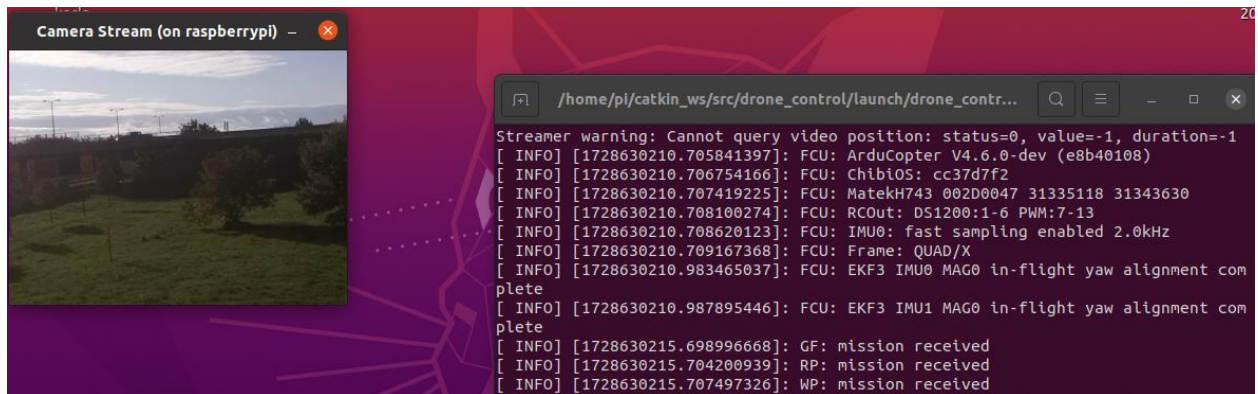
Slika 7.14. Graf prikaz brzine drona tokom leta

Na slici 7.14. prikazana je brzina drona tokom izvođenja koda. Kao što se može uočiti, brzina drona je bila otprilike 2.5 m/s. Na grafu brzine je prikazano kada i u kojem trenutku je dron poletio, odnosno kada je prešao u „guided” način rada i aktivirao motore. Na grafu se može uočiti i lebđenje u zraku u trajanju od 3 sekunde, a to je pad brzine nakon svake točke na koju dron dođe.



Slika 7.15. Graf prikaz visine drona tijekom leta

Na slici 7.15. je prikazan graf visine koju je dron imao tokom testiranja, izvođenja kvadrata u zraku. Iz grafa se može zaključiti kako je dron održavao visinu od 10 metara, uz male oscilacije koje su zanemarive jer se tijekom testiranja mora uzeti i vremenski uvjeti, poput mogućnost vjetera u tom trenutku.



**Slika 7.16.** Prikaz video stream-a s kamere

Na slici je prikaz video stream-a u stvarnom vremenu od strane drona tijekom samog testiranja. Tijekom testiranja koda, u svakom trenutku korisnik ima u uvid događanja i situacije koje se nalaze ispred drona zbog prijenosa videa. Također praćenje stanja drona su vidljivi na samom terminalu koji je također prikazan.

Treći dio testiranja drona je bio izvođenje kvadrata u dimenziji 5x5 uz promjenu visine i brzine prilikom leta.

```
# Funkcija za postavljanje ciljanih pozicija s yaw kutom i zadržavanje brzine
def set_position_with_yaw(local_position_pub, x, y, z, yaw_degrees, velocity):
    goal_pose = PoseStamped()
    goal_pose.pose.position.x = x
    goal_pose.pose.position.y = y
    goal_pose.pose.position.z = z
```

**Slika 7.17.** Prikaz funkcija za postavljanje pozicija i brzine

Slika 7.17. prikazuje funkciju „set\_position\_with\_yaw koja služi za postavljanje ciljanih koordinata (x, y, z) drona s određenim kutom yaw i brzinom. Funkcija koristi ROS poruku „PoseStamped” kako bi definirala ciljnu poziciju drona u prostoru. Funkcija prima šest parametara, „local\_position\_pub” koja se koristi za ciljne pozicije, x, y, z koordinate, „yaw\_degrees” za kut rotacije oko z-osi (yaw) izražen u stupnjevima, „velocity” brzina kojom dron treba doći do cilja.

```

# Izračunavanje udaljenosti do cilja
current_position_x = current_position.pose.position.x
current_position_y = current_position.pose.position.y
current_position_z = current_position.pose.position.z

distance = math.sqrt((goal_pose.pose.position.x - current_position_x) ** 2 +
                    (goal_pose.pose.position.y - current_position_y) ** 2 +
                    (goal_pose.pose.position.z - current_position_z) ** 2)

# Izračunavanje vremena potrebnog za dolazak do cilja pri zadanoj brzini
time_to_reach = distance / velocity if velocity > 0 else 0

local_position_pub.publish(goal_pose)
rospy.sleep(time_to_reach) # Čekanje na izračunato vrijeme za doseganje točke

```

**Slika 7.18.** Prikaz dijela koda za izračunavanje udaljenosti i vremena

Slika 7.18. prikazuje funkciju koja izračunava udaljenost između trenutne i ciljne pozicije drona te određuje vrijeme potrebno za dolazak do cilja pri zadanoj brzini. Prvo se uzimaju trenutne koordinate drona, a zatim se računa udaljenost koristeći Euclidovu formulu. Vrijeme putovanja izračunava se dijeljenjem udaljenosti s brzinom, pod uvjetom da je brzina veća od nule. Nakon toga, ciljane pozicija se objavljuje na odgovarajućem ROS kanalu kako bi se dron mogao kretati prema cilju, a program zatim čeka izračunato vrijeme da bi omogućio dronu dolazak do ciljne točke prije nastavka daljnjih operacija.

```

# Polijetanje na 10 metara
rospy.loginfo("Polijetanje na visinu od 10 metara")
takeoff_srv(altitude=10, latitude=0, longitude=0, min_pitch=0, yaw=0)
rospy.sleep(10) # Stabilizacija na 10m

# 1. Pomak na prvu točku (5, 0, 10) pri 2 m/s
rospy.loginfo("Pomicanje na (5, 0, 10) pri 2 m/s")
set_position_with_yaw(local_position_pub, 5, 0, 10, 0, 2)
rospy.sleep(3)

# 2. Rotacija na 90 stupnjeva i pomak naprijed pri 1.5 m/s
rospy.loginfo("Rotacija na 90 stupnjeva i pomak naprijed pri 1.5 m/s")
set_position_with_yaw(local_position_pub, 5, 5, 5, 90, 1.5)
rospy.sleep(3)

# 3. Rotacija na 180 stupnjeva i pomak naprijed pri 2 m/s
rospy.loginfo("Rotacija na 180 stupnjeva i pomak naprijed pri 2 m/s")
set_position_with_yaw(local_position_pub, 0, 5, 10, 180, 2)
rospy.sleep(3)

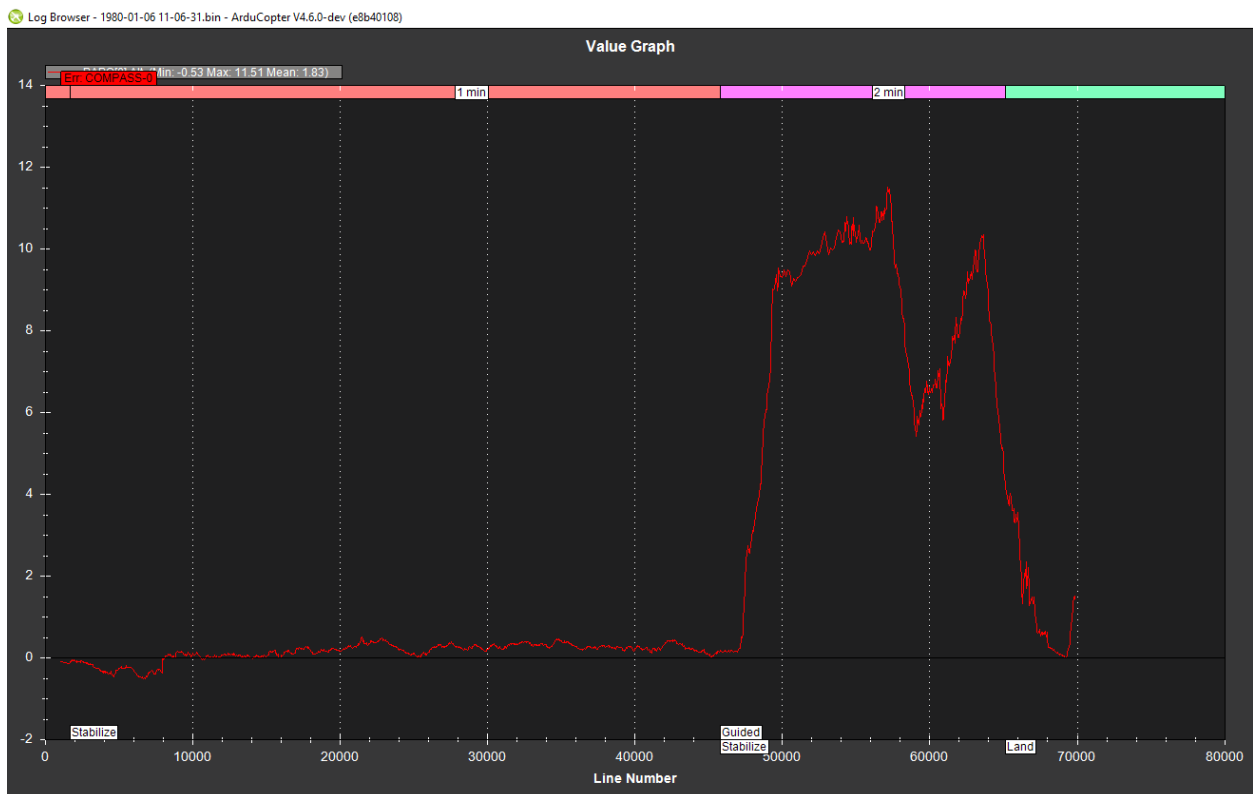
# 4. Rotacija na 270 stupnjeva i pomak naprijed, vraćajući se na početnu točku pri 1.5 m/s
rospy.loginfo("Rotacija na 270 stupnjeva i pomak naprijed pri 1.5 m/s")
set_position_with_yaw(local_position_pub, 0, 0, 5, 270, 1.5)
rospy.sleep(3)

# Slijetanje drona
rospy.loginfo("Slijetanje drona")
land_srv(altitude=0, latitude=0, longitude=0, min_pitch=0, yaw=0)

```

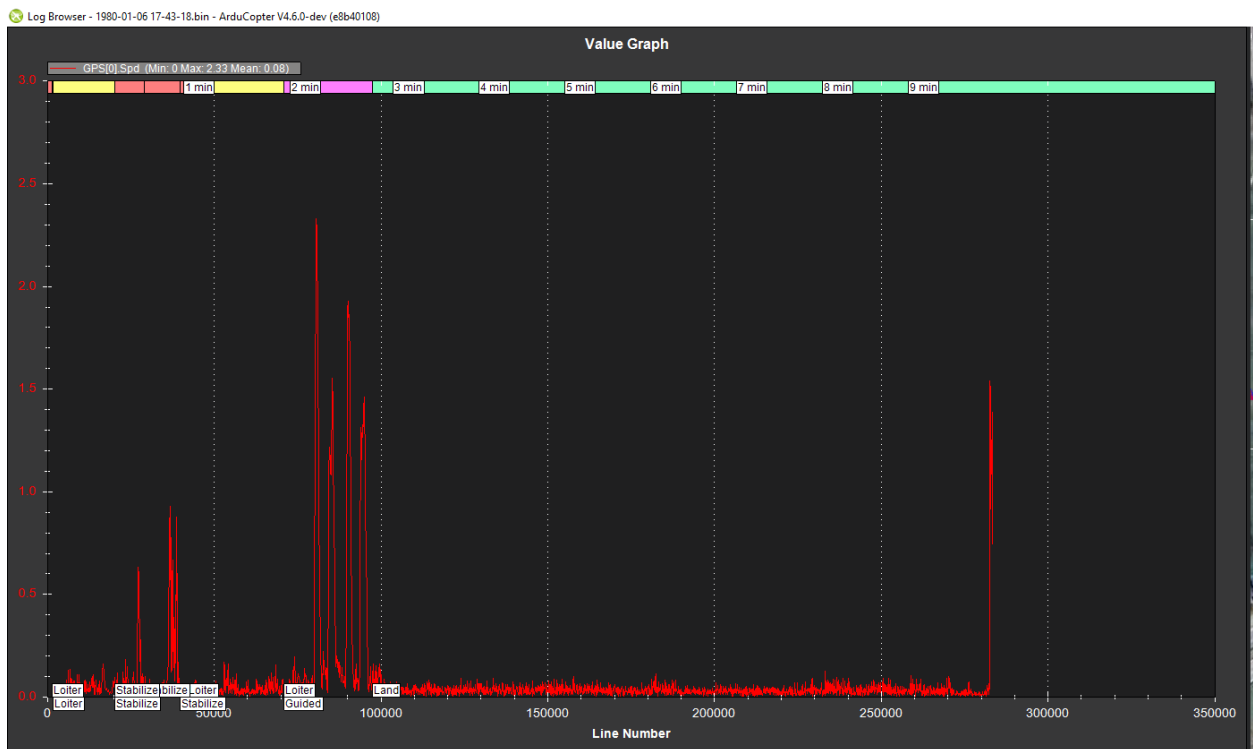
**Slika 7.19.** Prikaz dijela koda mijenjanja visine i brzine pri izradi kvadrata 5x5

Slika 7.19. prikazuje drona koji uzlijeće na visinu od 10 metara pozivom funkcije „takeoff\_srv”, koja se izvršava s parametrima za visinu, geografske koordinate i kut. Zatim, dron se pomiče na točku (5, 0, 10) pri brzini od 2m/s koristeći funkciju „set\_position\_with\_yaw”. Stabilizacija na toj točki traje tri sekunde. U sljedećem koraku, dron rotira za 90 stupnjeva, a zatim se pomiče na novu točku (5, 5, 10) pri brzini od 1.5 m/s. Ova sekvenca omogućava dronu da promijeni svoj smjer i nastavi s pomicanjem, uz stabilizaciju od tri sekunde. Nakon toga, dron se rotira za 180 stupnjeva i kreće se na točku (0, 5, 10) pri brzini od 2 m/s, također s stabilizacijom od tri sekunde. U četvrtom koraku, dron se rotira na 270 stupnjeva i vraća se na početnu točku (0, 0, 10) pri brzini od 1.5 m/s, uz stabilizaciju od tri sekunde. Na kraju, poziva se funkcija „land\_srv”, koja omogućava dronu da sigurno sleti na tlo, uz odgovarajuće parametre za slijetanje.



**Slika 7.20.** Prikaz grafa mijenjane visine

Na slici 7.20. može se uočiti mijenjanje visine tijekom izvođenja koda. Dron kroz testiranje mijenja visinu s 10 metara na 5 metara tijekom izvođenja putanje. Može se vidjeti u kojem trenutku je dron poletio, odnosno prešao u „guided” način rada. Prvo poleti na visinu od 10 metara te do prve točke drži visinu od 10 metara, nakon toga ju smanjuje na 5 metara do iduće zadane točke, te nakon toga dron povećava svoju visinu nazad na 10 metara do iduće zadane točke te nakon toga slijeće na tlo i deaktivira svoje motore. Na grafu se mogu primijetiti i manje oscilacije, jer svakako prilikom izvođenja koda moram uzeti u obzir i vremenske uvjeti poput vjetra.



**Slika 7.21.** Prikaz grafa drona prilikom mijenjanja brzine

Na slici 7.21. vidi se u kojem trenutku je dron prešao u „guided” način rada. Dron je mijenjao brzinu s 2 m/s na 1.5 m/s, te se upravo to može vidjeti na grafu. Dron je imao minimalne oscilacije po pitanju brzine, no svakako je uspješno mijenjao brzinu koja mu je bila zadana.



**Slika 7.22.** Prikaz izrade kvadrata u dimenziji 5x5

Zaključno, testiranja manualnog i autonomnog upravljanja bila su uspješna. Dron je u konačnici uspješno prošao sva testiranja koja su potkrijepljena i grafičkim prikazom. Grafičkim prikazima

se utvrdilo uspješnost svih kalibracija koje su bili potrebni za postizanje samih rezultata. Iz grafova se može zaključiti kako je dron održavao visinu, mijenjao visinu i brzinu po potrebi testiranja čime je dron spreman za daljnja unapređenja, poput integracije analize fotografija i složenijih zadataka u budućnosti.



## 8. ZAKLJUČAK

Ovaj diplomski rad detaljno prikazuje postupak razvoja drona koji omogućuje manualno upravljanje i autonomno upravljanje pomoću Robotskog Operacijskog Sustava (ROS). Odabir i povezivanje hardverskih komponenti, poput Raspberry Pi, kamere, senzora i kontrolera leta, predstavljaju ključne korake u izradi funkcionalnog sustava.

ROS, u kombinaciji s Raspberry Pi-jem, ima ključnu ulogu u postizanju autonomnog letenja drona. Raspberry Pi se koristi kao pomoćno računalo koje omogućuje obradu podataka u stvarnom vremenu, dok ROS djeluje kao posrednik koji upravlja komunikacijom između različitih dijelova sustava. Implementacija ROS paketa, poput MAVROS-a, omogućuje komunikaciju s Ardupilotom, što dronu omogućava razmjenu podataka o stanju sustava, senzorskim podacima te zadanim putanjama leta. MAVROS osigurava preciznu kontrolu i stabilnost drona tijekom autonomnog letenja, omogućujući interakciju s GPS-om, IMU-om i ostalim sensorima putem ROS poruka.

Rad je pokazao važnost preciznog postavljanja softverskog okruženja, kao i kalibraciju senzora. Uvođenjem kamere kao dijela sustava omogućeno je dodatno prikupljanje podataka u vidu fotografija i videozapisa, što pruža dodatne mogućnosti nadzora i praćenja autonomnih misija.

Cijeli proces izrade ovog drona, od inicijalnog razvoja hardverskih komponentata do konačne implementacije softvera, pokazao je složenost i interdisciplinarnu potrebu za kreiranje ovakvog sustava. Kroz integraciju ROS-a omogućeno je uspješno autonomno letenje drona, dok je istovremeno zadržana opcija za manualnu kontrolu. Ovaj rad tako pruža cjelovit prikaz procesa izrade drona koji kombinira napredne hardverske i softverske tehnologije, uz naglasak na njegovu praktičnu primjenu i fleksibilnost.

## LITERATURA

- [1] Wikipedia, Kettering Bug [online], Wikipedia Foundation Inc., 2024. dostupno na: [https://en.wikipedia.org/wiki/Kettering\\_Bug](https://en.wikipedia.org/wiki/Kettering_Bug)
- [2] Wikipedia, V-1 flying bomb [online], Wikipedia Foundation Inc., 2024., dostupno na: [https://en.wikipedia.org/wiki/V-1\\_flying\\_bomb](https://en.wikipedia.org/wiki/V-1_flying_bomb)
- [3] Wikipedia, AAI RQ-2 Pioneer [online], Wikipedia Foundation Inc., 2024., dostupno na: [https://en.wikipedia.org/wiki/AAI\\_RQ-2\\_Pioneer](https://en.wikipedia.org/wiki/AAI_RQ-2_Pioneer)
- [4] Wikipedia, General Atomics MQ-1 Predator [online], Wikipedia Foundation Inc., 2024., dostupno na: [https://en.wikipedia.org/wiki/General\\_Atomics\\_MQ-1\\_Predator](https://en.wikipedia.org/wiki/General_Atomics_MQ-1_Predator)
- [5] Wikipedia, Quadcopter [online], Wikipedia Foundation Inc., 2024., dostupno na: <https://en.wikipedia.org/wiki/Quadcopter>
- [6] CFD Flow Engineering, Working Principle and Components of Drone, Quadcopter Dynamics, Forces and Moments Acting on a Drone [online], 2024., CFD Flow Engineering, dostupno na: <https://cfdflowengineering.com/working-principle-and-components-of-drone/>
- [7] Uloga dronova u poljoprivrednoj proizvodnji [online], Digital Agro, dostupno na: <https://digitalagro.eu/uloga-dronova-u-poljoprivrednoj-proizvodnji/>
- [8] Stas Organizer, Dronovi: Revolucionarni alati za modernu građevinsku industriju [online], stasorganizer.com, 2024., dostupno na: <https://stasorganizer.com/hr/blog/1/dronovi-revolucionarni-alati-za-modernu-gradevinsku-industriju/>
- [9] Primjerna dronova (autonomni robot) u nadzoru prometa [online], Sveučilišni računski centar Srce, 2024., dostupno na: <https://www.croris.hr/crosbi/publikacija/ocjenski-rad/458482>
- [10] ROS::Home, ROS – Robot Operating System [online], video, Open Robotics, 2021., dostupno na: <https://www.ros.org/>
- [11] Robotski operacijski sustav, diplomski rad [online], Martina Šarić, 2018., dostupno na: <https://www.mathos.unios.hr/~mdjumic/uploads/diplomski/%C5%A0AR26.pdf>
- [12] Foxglove, Creating ROS 1 Actions [online], 2024., Foxglove Technologies, Inc., dostupno na: <https://foxglove.dev/blog/creating-ros1-actions>

- [13] ROS.org, Rospy, Package Summary [online], Open Robotics, 2017., dostupno na: <http://wiki.ros.org/rospy>
- [14] ROS.org, Roscpp, Package Summary [online], Open Robotics, 2015., dostupno na: <http://wiki.ros.org/roscpp>
- [15] ArduPilot Versatile, Trusted, Open, ArduPilot documentation [online], ArduPilot Dev Team, 2024., dostupno na: <https://ardupilot.org/ardupilot/>
- [16] Oscar Liang, Quadcopter Hardware Overview – Every Component Explained [online], 2018., dostupno na: <https://oscarliang.com/quadcopter-hardware-overview/>
- [17] Orqa shop, 4in1 30x30 ESC [online], Orqa Shop – High end FPV Equipment, 2024., dostupno na: <https://shop.orqafpv.com/products/orqa-orqa-electronic-speed-controller>
- [18] MatekSys, Flight Controller H743-Slim V3 [online], Matek Systems, dostupno na: <https://www.mateksys.com/?portfolio=h743-slim>
- [19] Orqa shop, ImmersionRC Ghost UberLite Tx Module [online], Orqa Shop – High end FPV Equipment, 2024., dostupno na: <https://shop.orqafpv.com/products/immersionrc-immersionrc-ghost-uberlite-tx-module>
- [20] HolyBro, SiK Telemetry Radio V3 [online], 500 mV 915mHz, HolyBro store, 2024., dostupno na: <https://holybro.com/products/sik-telemetry-radio-v3?variant=42801818239165>
- [21] Orqa shop, ImmersionRC Ghost Atto Duo Receiver [online], Orqa shop – High end FPV Equipment 2024., dostupno na: <https://shop.orqafpv.com/products/immersionrc-immersionrc-ghost-atto-duo-receiver>
- [22] Raspberry Pi, Raspberry Pi Camera Module 2 [online], dostupno na: <https://www.raspberrypi.com/products/camera-module-v2/>
- [23] What are pros and cons of using Raspberry Pi for ROS Development? [online], LinkedIn, 2024., dostupno na: <https://www.linkedin.com/advice/3/what-pros-cons-using-raspberry-pi-ros-development-skills-ros>
- [24] HolyBro, M10 GPS [online], HolyBro Store, 2024., dostupno na: [https://holybro.com/products/m10-gps?srltid=AfmBOorWz\\_tFeM5Y36V-2ejdUD3gFeGimoKl\\_vzoTM1EUHF5 RRiz6C6A](https://holybro.com/products/m10-gps?srltid=AfmBOorWz_tFeM5Y36V-2ejdUD3gFeGimoKl_vzoTM1EUHF5 RRiz6C6A)

[25] Circuit Rocks, Dc-Dc Adjustable Buck Converter Module LM2596S [online], circuitrocks, 2024., dostupno na: <https://circuit.rocks/products/dc-buck-booster-step-up-down-dc-dc-module-adjustable-lm2596s>

[26] Orqa shop, Orqa FPV.Ctrl Radio Controller [online], Orqa shop – High end FPV Equipment, 2024., dostupno na: <https://shop.orqafpv.com/products/orqa-orqa-fpv-ctrl-radio-controller>.

## SAŽETAK

Ovaj diplomski rad prikazuje dizajn, implementaciju i razvoj drona opremljenog kamerom, sposobnog za manualno i autonomno upravljanje putem Robotskog Operativnog Sustava (ROS). Cilj je dokumentirati cijeli proces izrade drona, od integracije hardverskih komponenti do kreiranja robusne softverske arhitekture koja omogućava stabilan let, prikupljanje podataka i izvršavanje autonomnih misija.

Početna faza uključuje konfiguraciju drona za ručno upravljanje pomoću Mission Planner softvera za zemaljsku kontrolu dizajniranog za konfiguraciju i operativno upravljanje dronovima baziranim na Ardupilotu. Nakon toga, sustav se proširuje kako bi omogućio autonomni let, pri čemu Raspberry Pi služi kao pomoćno računalo koje komunicira s kontrolerom leta. S ugrađenom kamerom, dron može snimati fotografije i videozapise tijekom autonomnih zadataka, što ga čini idealnim za primjene poput nadzora.

Poseban naglasak stavljen je na neometanu integraciju softverskih komponenti potrebnih za autonomnost, pri čemu ROS upravlja komunikacijom između različitih podsustava kako bi osigurao preciznu kontrolu visine, položaja i putanje leta drona. U radu su također detaljno opisani ključni koraci poput kalibracije senzora, podešavanja sustava i evaluacije performansi.

U konačnici, ovaj rad pruža sveobuhvatan vodič za razvoj i programiranje drona, s posebnim naglaskom na postizanje autonomne funkcionalnosti putem ROS-a te integraciju kamere za snimanje medijskih sadržaja.

**Ključne riječi:** dron, Ardupilot, ROS (Robotski Operativni sustav), autonomno letenje, Raspberry Pi, kamera.

# ABSTRACT

## Development of a drone operated by the ROS

This thesis presents the design, implementation, and development of a drone equipped with a camera, capable of both manual and autonomous operation via the Robot Operating System (ROS). The objective is to document the entire process of constructing the drone, from hardware integration to the creation of a robust software architecture that facilitates stable flight, data collection, and the execution of autonomous missions. The initial phase involves configuring the drone for manual control using Mission Planner, a ground control software designed for configuring and operating ArduPilot-based drones. Following this, the system is extended to enable autonomous flight, where a Raspberry Pi serves as a companion computer, interfacing with the flight controller. With its onboard camera, the drone can capture photos and videos during autonomous tasks, making it ideal for surveillance applications. A key focus is placed on the seamless integration of software components required for autonomy, with ROS managing communication between the various subsystems to ensure precise control over the drone's altitude, position, and flight trajectory. The thesis also details critical steps such as sensor calibration, system tuning, and performance evaluation. Ultimately, this work provides a comprehensive guide to the development and programming of a drone, with a particular emphasis on achieving autonomous functionality through ROS and integrating a camera for media capture.

**Keywords:** drone, ArduPilot, ROS(Robot Operating System), autonomous flight, Raspberry Pi, camera.

## ŽIVOTOPIS

Karla Križanac rođena 21. listopada 1998. godine u gradu Požega. Osnovno obrazovanje započeto 2005. godine upisom u Osnovnu školu Antuna Kanižlića. Tokom osnovnoškolskog obrazovanja postignute razna postignuća na županijskim natjecanjima iz područja tehničke kulture, povijesti i geografije. Nakon završetka osnovnog školovanja, godine 2013. upisuje Opću Gimnaziju Požega, te godine 2017. upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Preddiplomski studij završava 2020. godine te upisuje diplomski studij računarstva na istom fakultetu, izborni blok Robotika i umjetna inteligencija.

---

Karla Križanac

## **PRILOZI**

1. „Razvoj drona upravljano od strane Robotskog operacijskog sustava” u .docx formatu
2. „Razvoj drona upravljano od strane Robotskog operacijskog sustava” u .pdf formatu
3. Izvorni kod (<https://github.com/KarlaKrizanac/diplomski>)