ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

AVTONOMEN MODEL LETALA

Tematsko področje: ELEKTROTEHNIKA, ELEKTRONIKA IN ROBOTIKA

Avtorja:
Simon Strmšnik, 4. letnik - elektrotehnika
Kamil Kosi, 4. letnik - mehatronika

Mentor:
Jože Lukanc, univ. dipl. inž.

Velenje, 2019

Raziskovalna naloga je bila opravljena na Šolskem centru Velenje, na Elektro in računalniški šoli.

Mentor: Jože Lukanc, univ. dipl. inž.

Datum predstavitve: 5. 3. 2019

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD ŠC Velenje, šolsko leto 2018/2019

KG Letalstvo / Elektrotehnika / Strojništvo

AV STRMŠNIK, Simon, KOSI, Kamil

SA LUKANC, Jože

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2019

LI 2019

IN **AVTONOMEN MODEL LETALA**

TD Raziskovalna naloga

OP VI, 48 str., 5 tab., 21 sl., 3 graf., 8 pril., 13 vir.

IJ SL

JI sl/en

AI Danes si v svetu letalstva skorajda ne znamo predstavljati upravljanja letal brez pomoči avtopilota. Avtopilot nam omogoča preprostejše, varnejše in manj utrujajoče upravljanje letala. Kljub dobro razvitemu sistemu ga večina lahkih letal še vedno v celoti ne uporablja. Težava je, da je izdelava zanesljivega krmilja, ki bi bil zmožen brezhibno opraviti vse faze leta zelo zahtevna in tudi finančno velika. Zatorej lahka in tudi športna letala večinoma ne uporabljajo tega sistema. V raziskovalni nalogi sva si zadala cilj, da bi izdelala poseben model letala, ki bi ga opremila z lastno izdelanim sistemom krmiljenja, ki bi opravljal funkcijo avtopilota. Krmilno vezje je sestavljeno iz več mikrokrmilniških ploščic Arduino, ki med seboj komunicirajo preko serijske komunikacije. Podatke med letenjem pridobivava z več senzorji. Žiroskop nama daje podatke o naklonu, nagibu in smeri letala, GPS modul daje podatke o položaju ter višini in ultrazvočni senzor, ki služi za merjenje višine pri pristanku. Letalo lahko leti v samodejnem ali pa ročnem načinu, kjer letalo vodimo z daljincem. Krmiljenje letala je izvedeno s servo motorji, glavni pogon pa predstavlja brezkrtačni enosmerni motor. Namen raziskovalne naloge je bil, da bi s cenovno dostopnimi komponentami izdelala zanesljiv sistem brezpilotnega letenja, ki bi se zaradi dostopnosti lahko uporabljal v letalskem modelarstvu, pa tudi pri lahkih letalih predvsem za pomoč pri vzdrževanju smeri in izvajanje preprostih manevrov v zraku.

KEY WORD DOCUMENTATION

ND Šolski center Velenje, šolsko leto 2018/2019
CX Aviation / Electrical Engineering / Mechanical Engineering
AU STRMŠNIK, Simon, KOSI, Kamil
AA LUKANC, Jože
PP 3320 Velenje, SLO, Trg mladosti 3
PB Šolski center Velenje,
PY 2019
TI **AUTONOMOUS AIRCRAFT MODEL**
DT Research work
NO VI, 48 p., 5 tab., 21 fig., 3 graf., 8 ann., 13 ref.
LA SL
AL sl / en
AB Today, in the aviation world, we almost cannot imagine the usage of the plane without the help of the autopilot. Autopilot makes piloting simpler, safer and less tiring. Despite the well-developed autopiloting systems, most light aircraft still do not fully use it. The problem is that the reliable control system that automates all phases of the flight, is very hard to make and it is also financially tiring. Therefore, lightweight and sporty aeroplanes do not contain this type of system. In the research project, we set ourselves the goal of creating a special model equipped with the controlling-system that can perform autopilot functions. Our control circuit consists of several Arduino microcontroller boards that communicate with each other via serial communication. Data is obtained during flying with multiple sensors. The gyroscope gives us the slope, the inclination and the plane's deviation data, the GPS module provides us with the position and height information and the ultrasonic sensor is used for measuring the height at landing phase. The plane can fly in automatic or manual mode. Aeroplane control is carried out with servo motors, and the main drive is a brushless DC motor. The research work is intended to make a reliable system of autopilot that can be used in scaled models and even real light aeroplanes, especially for maintaining and controlling the straight flight and for automation of simple manoeuvres in the air.

KAZALO VSEBINE

KAZALO TABEL

KAZALO SLIK

KAZALO GRAFOV

KAZALO PRILOG

1 UVOD

Živimo v času, ko postaja letalstvo vedno bolj pomemben dejavnik, saj govorimo o prevažanju ljudi. Klasična prevozna sredstva kot so avtomobili, imajo dve slabosti; da lahko z njimi potujemo sorazmerno počasi in da onesnažujejo okolje. Problem se pojavlja predvsem v zelo velikih mestih, kjer zaradi gostote prometa potrebujete veliko časa, da prevozite neko razdaljo. Vizionarji vidijo edino rešitev v tem, da človek ne bi več potoval samo po tleh, ampak da bi ljudje začeli potovati po zraku, kjer je prostora bistveno več kot samo na tleh. Vodilna podjetja že razvijajo tako imenovane letalnike, ki bodo služili temu namenu.

Ta tehnologija se bo za razliko od današnjega potniškega letalstva, kjer letalo prevaža naenkrat tudi več sto ljudi, osredotočila na posameznika. Ravno zaradi tega je nesmiselno, da bi ta moderna vozila oz. plovila imela pilota, saj bo teh vozil "malo morje" hkrati pa je tudi nesmiselno, da bi bila krmiljena ročno, saj bi bilo posameznika težko naučiti upravljati takšno plovilo. Zato se inženirji nagibajo k temu, da bodo plovila delovala popolnoma avtonomno, kar bo uporabnikom prijazno in hkrati tudi zelo varno.

Prej opisana vizija prihodnosti ima prav v Sloveniji največji potencial, saj na tem dela podjetje Pipistrel. Kot ljubitelja letalstva sva bila zelo navdušena nad idejo avtonomnega letenja.

Z izkušnjami v letalskem modelarstvu v kombinaciji znanja s področja elektrotehnike in strojništva, sva tudi midva želela "zagristi znanost" ter narediti korak naprej. Avtonomno letenje je pojem, ki sva ga čimbolj želela dodelati. Kot modelarja sva se odločila, da bova izdelala poseben model letala, ki bi čimbolje upravljal funkcijo brezpilotnega letenja, katerega uporabnost bi v začetni fazi lahko bila v športnem letalstvu in letalskem modelarstvu.

Raziskovalna naloga obsega raziskovanje čimbolj optimalnih rešitev, načrtovanje, pa tudi samo izdelavo letala ter seveda izdelavo krmilnega vezja s funkcijo.

1.1 Hipoteze

Hipoteze, ki sva si jih postavili, so naslednje:

1. Z žiroskopskim modulom MPU 6050 je možno vzdrževati nagib, naklon in smer letala.
2. Avtopilot lahko med letom samodejno izvaja krog čakanja pred pristankom (angl. holding).
3. Izdelava avtopilota, ki regulira smer letenja je izvedljiva s cenovno ugodnimi komponentami.

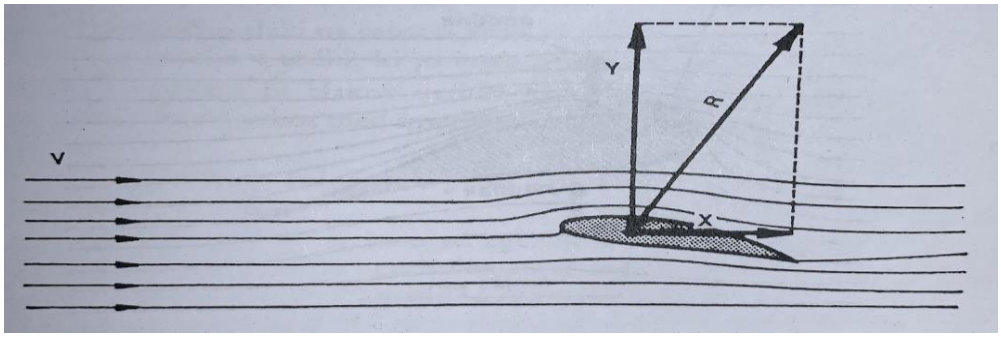
2 PREGLED OBJAV

2.1 Osnove aerodinamike pri letalu

Letalo je težje od zraka. Za letenje potrebuje silo, ki je nasprotna sili zemeljske težnosti. Ta sila je vzgon, ki se med premikanjem skozi zrak, ustvarja na nosilnih površinah letala (običajno krilih).

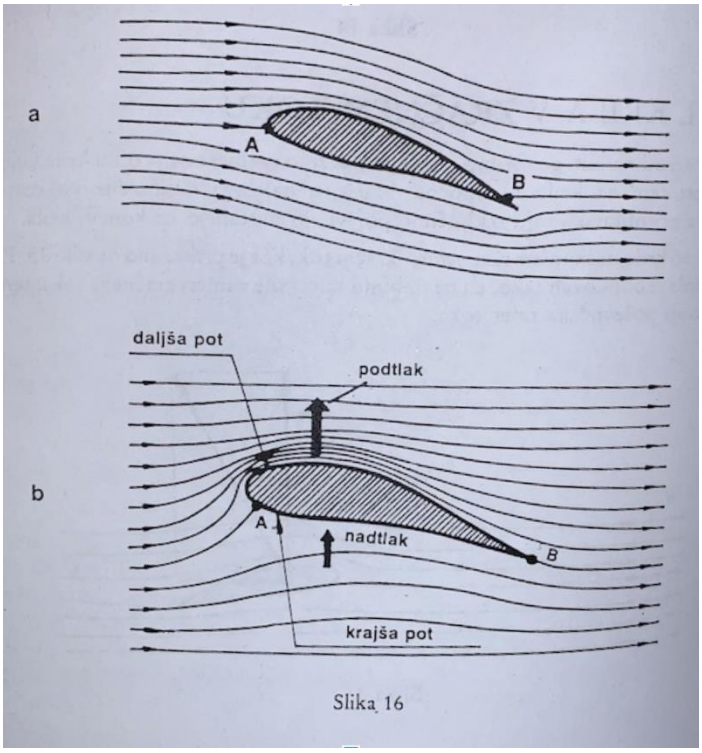
Profil krila je oblikovan tako, da pri premikanju skozi zračno maso ne povzroči samo upora, temveč tudi silo, ki deluje pravokotno na smer zračnega toka. Rezultanto teh dveh sil imenujemo rezultirajoča aerodinamična sila. Komponenta v smeri zračnega toka je zračni upor profila X, pravokotno na tok fluida pa se pojavi sila vzgona Y. Upor profila X je vsota tlačnega upora X_p in upora zaradi trenja X_r.

Na sliki 1 je s tokovnicami prikazan tok zraka okoli profila. Profil sam imenujemo ničelna tokovnica, saj skozenj zračni tok ne teče. Na krilu imamo točko A in B. Po zakonu o ohranitvi mase, mora zrak, ki se je razcepil v točki A, istočasno priti na zadnji rob profila v točko B po zgornji in spodnji konturi. Ker je pot od točke A do točke B po zgornji konturi veliko daljša kot po spodnji, mora imeti zrak po zgornji konturi profila veliko večjo hitrost.



Slika 1: Delovanje sil na krilo letala (Brezar, 1995)

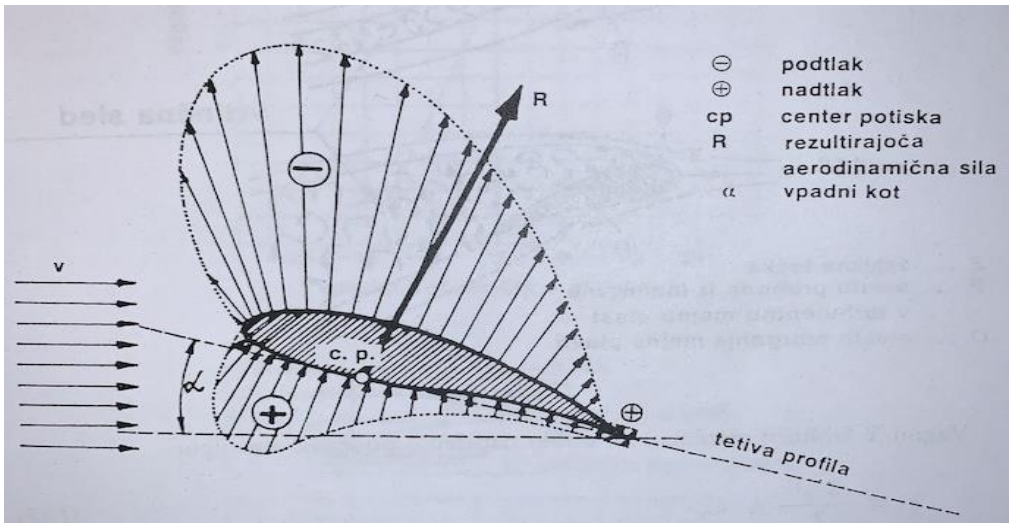
Po zakonu o ohranitvi energije se vsota kinetične in tlačne energije na enoto mase vzdolž iste tokovnice ne spreminja. Iz tega sledi, da mora biti na zgornji konturi, kjer je večja hitrost zraka, manjši tlak kot na spodnji konturi profila. Vzdolž zgornje in spodnje konture profila se hitrost zraka spreminja in temu ustrezno se spreminja tlak, kot je prikazano na sliki 2.



Slika 2: Prikaz zračnega toka čez profil krila (Brezar, 1995)

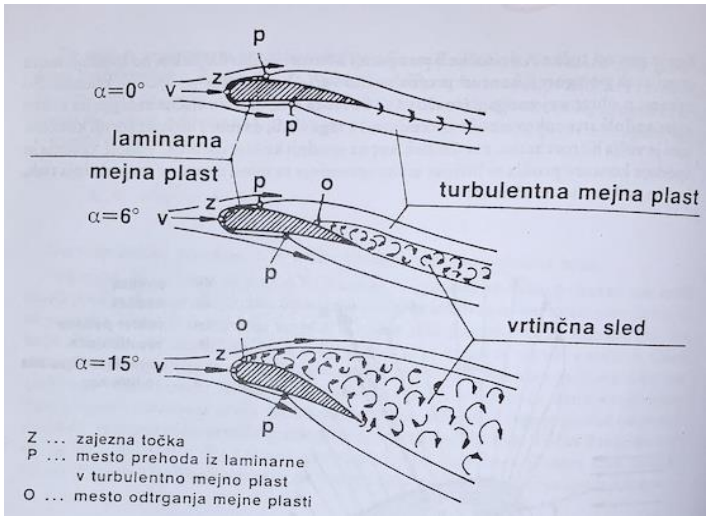
Zaradi omenjenega pojava, se na spodnji strani krila pojavi nadtlak, to je višji tlak, kot je atmosferski tlak v okolici. Na zgornji konturi profila pa se pojavi velik podtlak, ki je manjši tlak, kot je tlak okolice.

Slika 3 prikazuje silo, ki jo povzročijo tlaki na celotni konturi profila. To je rezultirajoča aerodinamična sila R, ki deluje na točko, ki jo imenujemo center potiska (c.p.).



Slika 3: Nastanek vzgonske sile (Brezar, 1995)

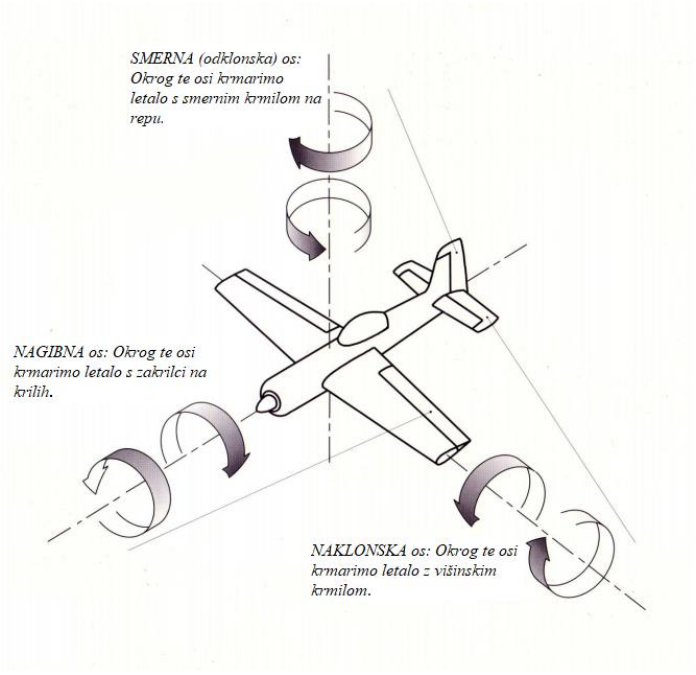
Porazdelitev hitrosti in tlakov vzdolž konture profila se spreminja z vpadnim kotom α. Vpadni kot α je kot med smerjo zračnega toka pred profilom in tetivo profila. Dogajanje ob določenem profilu pri različnih vpadnih kotih je prikazano na sliki 4. Pri vpadnem kotu $\alpha = 0°$ imamo na zgornji in spodnji konturi najprej laminarno mejno plast, ki v točki P preide v turbulentno in se ohrani do zadnjega roba profila. Ko vpadni kot povečamo na $\alpha = 6°$, pride na zgornji konturi v točki 0 do odtrganja mejne plasti. Pri vpadnem kotu $\alpha = 15°$ pa se mejna plast na zgornji konturi odtrga že čisto spredaj, kar povzroči porušitev vzgona (angl. stall).



Slika 4: Tipi zračnega toka (Brezar, 1995)

Vzgon je odvisen od kinetičnega tlaka, tlorisne površine krila A in brezdimenzijskega vzgonskega količnika C_y. Z meritvami se določa vzgonski količnik C_y in količnik upora C_x, za različne vpadne kote. Večji kot je vpadni kot, tem večja je sila vzgona pod pogojem, da kritičnega vpadnega kota ne prekoračimo (Brezar, 1995).

Položaj letala opisujejo tri osi. Iz slike 5 razberemo, da te osi imenujemo smerna os, nagibna os in naklonska os. Ta poimenovanja bova uporabila v nadaljevanju naloge.



Slika 5: Prikaz gibanja letala po X, Y, Z osi (http://dk.mors.si/Dokument.php?id=878, 13. 1. 2019)

2.2 Avtopilot

Avtopilot je sistem naprav za nadziranje in popravljanje trajektorije leta letala brez človekovega poseganja. Avtopilot v letalu ne nadomešča človeka, ampak mu zgolj pomaga pri upravljanju letala pri dolgih letih. To pilotu omogoča, da se posveti drugim, za let pomembnim nalogam, kot so preverjanje trajektorije leta, vremena, komunikacijo s kontrolo letenja in delovanjem letala.

Danes v večjih letalih velikokrat zasledimo polega avtopilota tudi avtomatski nadzor potiska letala (angl. autothrottle), ki nadzira moč letalskih motorjev.

V preteklosti so morali piloti ves čas nadzirati smer letala za varen let. Tekom let, se je doseg letal drastično podaljšal, s čimer pa se je tudi dolžina leta podaljšala, kar je pospešilo razvoj avtopilota.

Prvi enostaven žiroskopski avtopilot je izdelalo podjetje Sperry Corporation leta 1912. Krmiljen je bil s pomočjo mehanskega žiroskopa ter višinomera, ki sta letalo stabilizirala po dveh oseh. Zakrilca za naklon in smer so bila krmiljena s pomočjo hidravlike in so popravljala naklon in smer letala. Kontrola nagiba je bila rešena mehansko in sicer s pomočjo V-loma.

Danes vsa letala še vedno nimajo avtopilota. V splošnem letalstvu je avtopilot še vedno zelo redek in tudi manjša potniška letala za kratke razdalje še vedno ne uporabljajo avtopilota.

Avtopilote lahko delimo v 3 skupine:

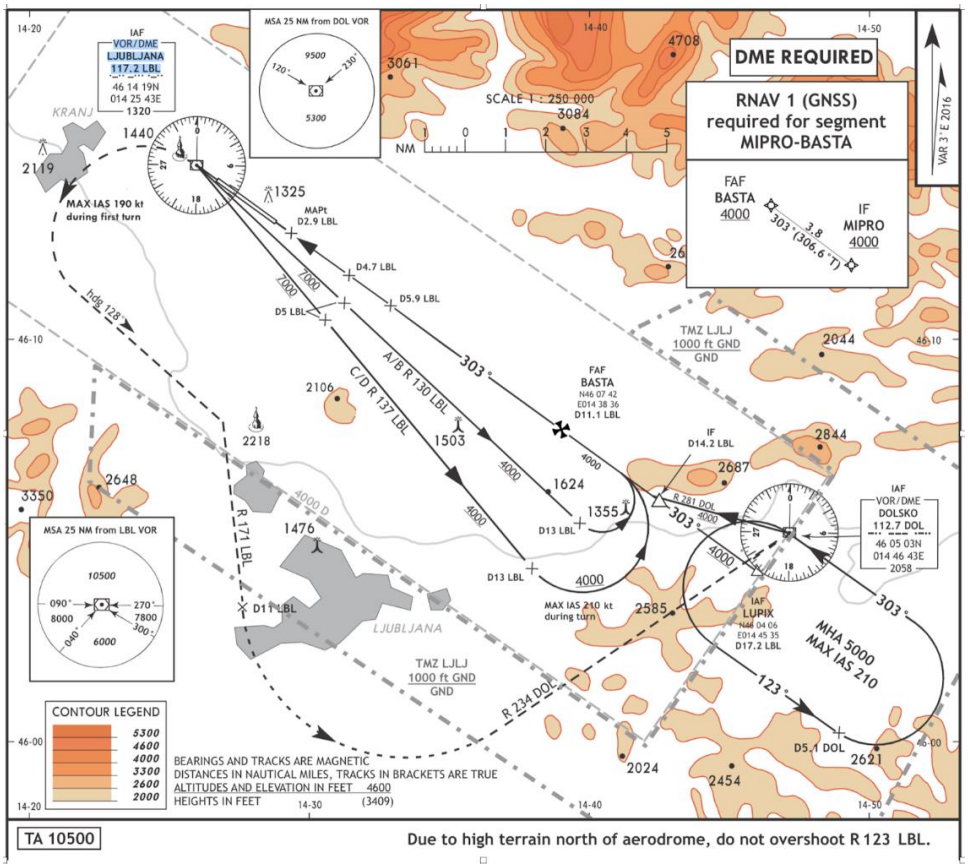
Prva skupina zajema avtopilota, ki nadzira samo eno os letenja in sicer nagib (angl. wing leveller). Druga skupina zajema dve osi, ki sta naklon in nagib. Takšen avtopilot ponavadi najdemo v manjših potniških letalih. 3-osni avtopilot nadzira tudi smer letala in na ta način popravlja tudi smer leta ali smer.

Moderni avtopiloti so ponavadi 3-osni in ponavadi razdelijo let na vožnjo po tleh (angl. taxi), vzlet (angl. takeoff), vzpon (angl. climb), vodoravni let (angl. cruise), spust (angl. descent), pristop k pristanku (angl. approach) ter pristanek (angl. landing phase). Danes obstajajo sistemi, ki avtomatizirajo vse te postopke letenja razen vožnje po tleh in vzletanja. Sistem za pristajanje poznamo pod imenom Autoland ali CAT, ki je prisoten na mnogih letališčih in je zelo uporaben za pristajanje v slabem vremenu.

Kljub temu inercialni sistem za nadzor (INS) po določenem času nabere napake v merjenju. Napake se pojavijo, zaradi napak v merjenju žiroskopa za merjenje Z-osi. To napako imenujemo napaka po z osi (angl. Z-axis drift). Daljši kot je let, tem večja je napaka tega senzorja. V letalstvu si zato pomagamo s kompenzacijo te osi s pomočjo GPS signala ter digitalnim kompenziranjem signala s pomočjo Kalmanovega filtra.
Pri implementiranju sistema za avtopilota je vedno na prvem mestu redundanca. Na primer Rockwell Collins AFDS-770 Autopilot, ki ga uporabljajo na letalu Boeing 777 uporablja potrojeni procesor tipa FCP-2002, ki ga mnogokrat preverijo in je izdelan v procesu, ki je odporen na radiacijo. (https://en.wikipedia.org/wiki/Autopilot)

2.3 Zadrževanje v letalstvu

V letalstvu je zadrževalni (angl. holding) manever, s katerim zadržujemo letalo v zraku, medtem pa vzdržujemo točno določen obseg kroženja. Pri vizualnem letenju VFR (angl. visual flight rules) ponavadi letimo po navidezni poti, ki se sklicuje na vidne objekte na zemlji kot so mostovi, jezera ali avtocestni odcepi. Pri instrumentalnem letenju IFR (angl. instrumental flight rules) je ponavadi začetna točka kroženja radijski oddajnik.
Kroženje ponavadi traja 4 min. Letalo leti 1min v določeni smeri, nato kroži 1min tako, da se obrne za 180 stopinj in to ponavlja. Primer kroga čakanja na letališču Jožeta Pučnika prikazuje slika 6. Glavni namen tega manevra je zadrževanje letala v zraku na točno določeni poziciji, ki je že prispelo do letališča, pa še ne more pristati bodisi zaradi gostega prometa, slabega vremena ali zaradi zasedenosti steze. Letala, ki čakajo na pristanek čakajo eno nad drugim z višinsko razliko 1000 feet. Letalo, ki je najnižje je naslednje, ki bo pristalo (https://en.wikipedia.org/wiki/Holding_(aeronautics), 3. 12. 2018).



Slika 6: Prikaz kroga čakanja (ang. Holding) v TMA Ljubljana (https://www.sloveniacontrol.si, 15. 1. 2019)

3 MATERIALI IN METODE DELA

V pregledu objav sva navedla teoretično podlago, ki je služila kot osnova za raziskovanje, načrtovanje in izdelavo brezpilotnega letala. Naloga se prepleta z raziskovanjem in projektiranjem, zato sva se odločila, da bosta v nalogi oba dela predstavljena hkrati, kar je za razumevanje najlažje. Kot raziskovalca, sva želela čim bolje zasnovati konstrukcijo letala, katero sva opremila s tehničnimi argumenti in izračuni. Opisan je tudi sam postopek izgradnje samega letala, kot tudi izdelava krmilnega vezja. Glavne rezultate sva dobila šele, ko je bil izdelek dokončan in sva brezpilotno letalo lahko preizkusila, zato sva se odločila, da rezultatov ne bova navajala sproti, ampak na koncu v poglavju Rezultati.

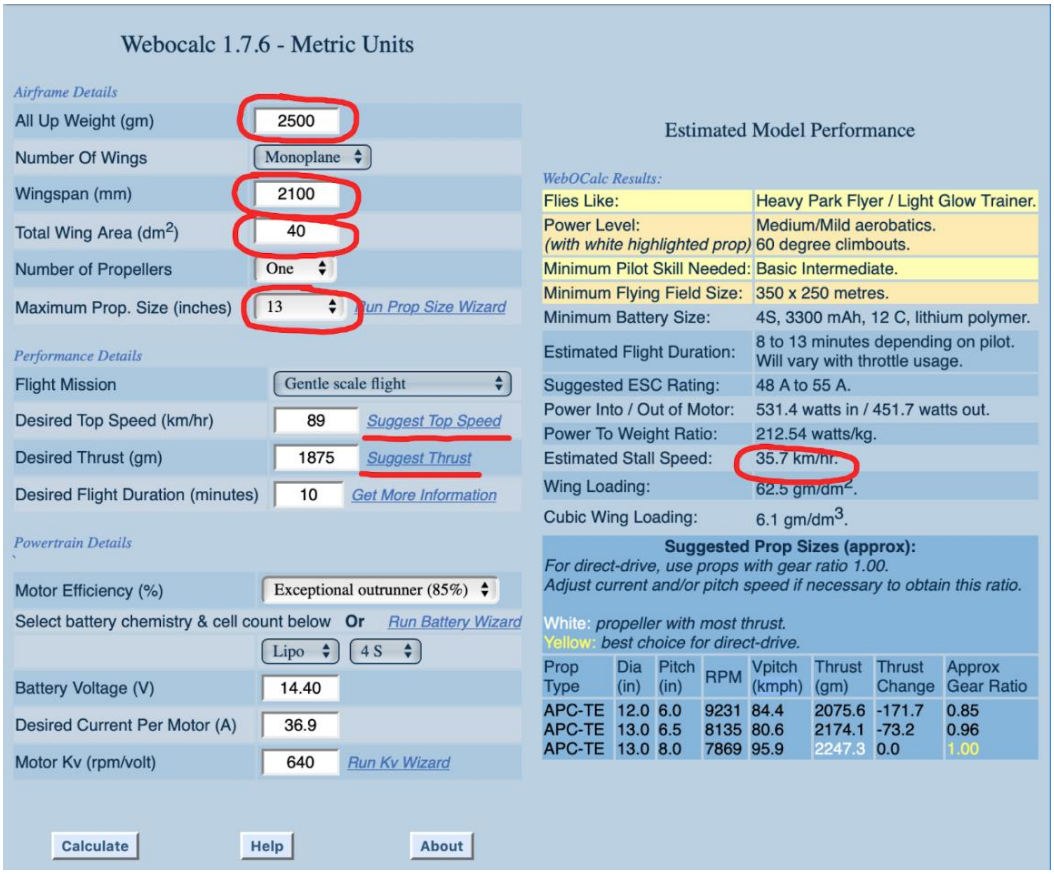
3.1 Raziskovanje in zasnova strojnega dela projekta

3.1.1 Zasnova letala

Zasnova letala je temeljila na štirih temeljnih točkah:

1. Za lažje testiranje sva želela, da letalo vzleti in pristane pri čim manjši hitrosti.
2. Druga pomembna lastnost letala je, da ima velik prostor za krmilnik.
3. Letalo mora biti med letom čim bolj stabilno.
4. Letalo mora imeti podvozje, s katerim lahko pristane na grobem ternu.

Z izdelavo letala sva začela tako, da sva predvidela težo celotnega letala in določila vrsto letala. Glede na težo napajalne baterije, pogonskega sistema, servo mehanizmov in krmilnika sva predpostavila, da bo teža letala znašala približno 2.5 kg.
Načrtovanje letala sva začela s krili za letalo. S pomočjo programa Webocalc 1.7.6 sva ugotovila, da optimalen premer kril znaša 2 m s skupno površino 40 dm^2. Rezultati iz programa so vidni na sliki 7. Podatek, ki nam pove osnovno karakteristiko letenja v zraku se imenuje WCL (angl. wing cubic loading). WCL nam pove, razmerje med površino krila in težo letala. Večji kot je ta parameter, tem hitreje se bo moralo letalo premikati skozi zrak, da bo letelo. Več o kubični obremenitvi krila je zapisano v poglavju 3.1.6. Pri velikosti teh krilih nama je omenjeni program, izračunal minimalno hitrost (angl. stall speed) v višini 35.7 km/h, kar je sprejemljivo za najino aplikacijo.



Slika 7: Program Webocalc 1.7.6 (vir lasten)

3.1.2 Izbira pogonskega motorja

Za pogonski motor sva se odločila, da bova izbrala električni brezkrtačni (BLDC) motor, saj ti motorji ponujajo izjemno razmerje med izhodno močjo in težo pogonskega sistema. Za letenje letala, je pomembno razmerje med potiskom motorja in težo letala. V tabeli 1 vidimo omenjeno razmerje pri različnih letalih.

Tip letala	Potisk / teža
Airbus A380	0.227
Concorde	0.372
Mig - 29	1.09
Space Shuttle	3

Tabela 1: Prikaz razmerja med potiskom in težo različnih letal (https://en.wikipedia.org/wiki/Thrust-to-weight_ratio, 10. 1. 2019)

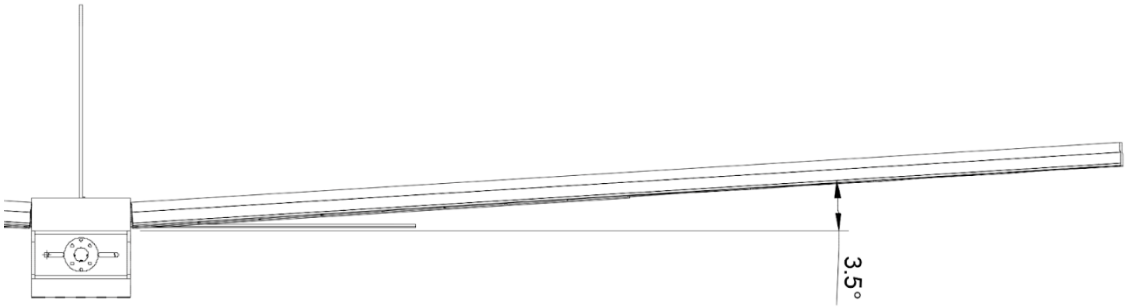
Pri modelih letal imamo raje, da so ta razmerja podvojena, saj je ponavadi prostor za vzletanje in pristajanje omejen, stil letenja pa je veliko bolj agresiven kot pri pravih različicah. Za najin model sva se odločila, da bova izbrala razmerje 1:1, kar pomeni, da sva izbrala motor, ki lahko zagotovi 25 N potiska.

Kupila sva motor z nazivno maksimalno močjo 800 W in hitrostjo 600 kv (600 obratov na volt na minuto). S propelerjem 13 inchov x 6 inchov proizvede omenjeni motor željeni potisk.

Propeler ima premer 33 cm in naklon, pri katerem prepotuje letalo v enem obratu 15cm v smeri letenja. Glede na naklon propelerja in glede na baterijo, ki sva jo uporabila, znaša idealna, maksimalna hitrost 22 m/s oz. 82 km/h ((15 V x 600 kv x 0.15 m) / 60 s).

3.1.3 V-lom pri letalskem krilu

V-lom (angl. dihedral) pomeni v letalstvu kot med vodoravno črto in krilom letala. V-lom je lahko pri letalih pozitiven, kar pomeni, da so letalska krila pod kotom navzgor, lahko pa je tudi negativen. Kot V-loma ima zelo velik vpliv na istoimenski efekt (angl. dihedral effect). Ta efekt se kaže kot nagibni moment letala, ki se pojavi ko to ni v uravnoteženi, vodoravni legi. Ta lastnost je ključna za stabilnost letala po osi, ki poteka v smer premikanja letala. Slika 8 nam prikazuje kot V-loma pri letalu.



Slika 8: Prikaz V-loma na lastnem modelu (vir lasten)

Na modelu, ki sva ga izdelala sva ta efekt izkoristila z namenom, da bi bilo letalo čim bolj stabilno tudi brez vključenega avtopilota. Odločila sva se, da bova krila na najinem letalu postavila pod kotom 3.5°.

3.1.4 Izračun vzgona na letalskem krilu

Za izračun teoretičnega vpadnega kota letala moramo razumeti formulo za izračun vzgona na letalskem krilu. Spodaj navedeno formulu sva pridobila z Nasine strani, kar je razlog, da so vse nadaljnje enote v formuli imperialne. Preprosta formula za izračun vzgona letalskega krila je sledeča:

$$L = \frac{1}{2} \cdot d \cdot v^2 \cdot s \cdot CL$$

Pri čemer upoštevamo:

- L … Vzgon (Lift), ki mora biti enak masi letala [lb]
- d … Specifična gostota zraka, ki se spreminja glede na nadmorsko višino. Te vrednosti lahko dobimo v I.C.A.O. Standard Atmosphere Table.
- v … Zračna hitrost letala [$\frac{\text{ft}}{\text{s}}$]
- s … Površina letalskih kril [ft^2]
- CL … Koeficient vzgona, ki je odvisen od profila krila ter od vpadnega kota

Vpadni kot letala in CL sta povezana in ju lahko razberemo z uporabo CL grafa, ki se nahaja v prilogi.

Zgoraj omenjeno formulo sva uporabila za izračun vpadnega kota, ki je potreben za letenje pri določenih hitrosti. Zračno hitrost in talno hitrost sva izenačila, saj z najinim modelom nikoli nisva letela v vetrovnem vremenu.

Najprej sva iz formule izrazila zahtevano količino. Vpadni kot, ki ga želimo izračunati je povezan s koeficientom vzgona (CL). Izražena formula, vstavljeni podatki letala:

$$CL = \frac{L}{0.5 \cdot d \cdot v^2 \cdot s} = \frac{5{,}44\text{lb}}{0{,}00241 \cdot (36.45)^2 \frac{\text{ft}}{\text{s}} \cdot 4{,}31\text{ft}^2}$$

L = 2.466 kg = 5.44 lb

d = 0.00241 (Razbrana iz tabele 2 - *I.C.A.O.* - tabela standardnih atmosferskih tlakov, če upoštevamo, da leži letališče Šoštanj na 380 m (1247 ft) nadmorske višine)

v = 40 $\frac{\text{km}}{\text{h}}$ = 36.45 $\frac{\text{ft}}{\text{s}}$ (Predpostavljena zračna hitrost letenja)

s = 43.7 dm^2 = 4.31ft^2

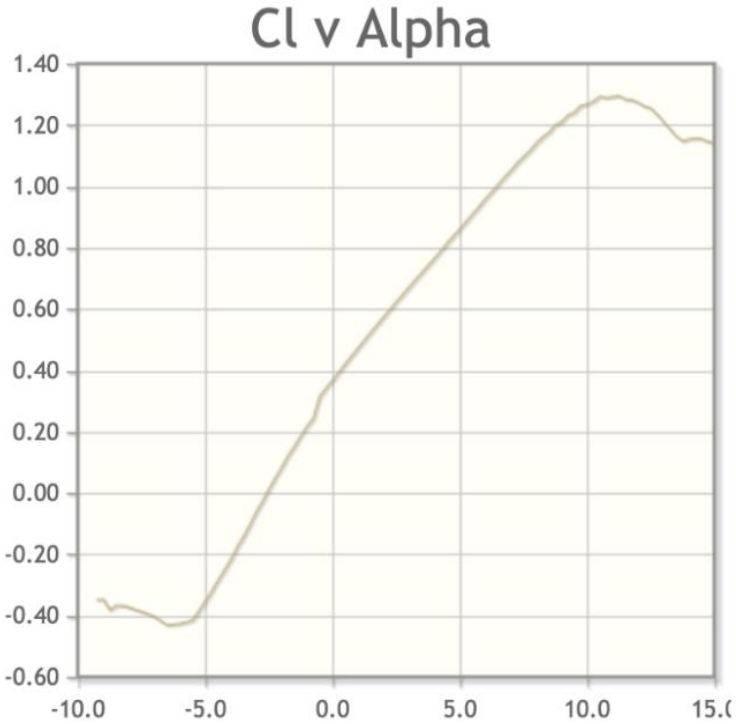
Po zgornji formuli sva izračunala koeficient vzgona, ki znaša 0.85. Če pogledamo v CL tabelo za najin profil krila, pri minimalnem Reynoldsovem številu ugotovimo, da znaša potreben vpadni kot za naše krilo pri hitrosti 40 km/h približno 5 stopinj. Vpadni kot prikazuje slika 9. Reynoldsovo število je koeficient, ki nam pove stopnjo turbulentnega ali laminarnega toka fluida. V letalstvu je število odvisno od hitrosti fluida, ki teče ob krilu in od višine profila krila. Ta koeficient vpliva na vrednosti v *CL* grafu prikazano v grafu 1. Graf, ki sva ga upoštevala pri izračunu, ima minimalen Reynoldsov koeficient, saj je debelina profila pri najinem letalu zanemarljivo majhna v primerjavi s krilom pravega letala.
(https://www.grc.nasa.gov/www/k-12/WindTunnel/Activities/lift_formula.html, 8. 11. 2018)

Chart A
I.C.A.O. Standard Atmosphere Table

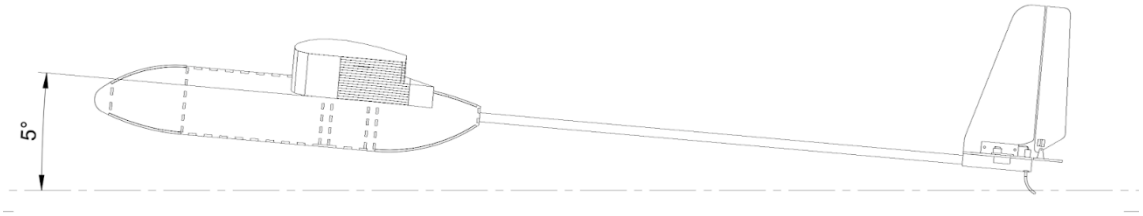
Altitude (Feet)	Density (d)	Speed of Sound (Knots)
0	.002377	661.7
1,000	.002308	659.5
2,000	.002241	657.2
3,000	.002175	654.9
4,000	.002111	652.6
5,000	.002048	650.3
6,000	.001987	647.9
7,000	.001927	645.6
8,000	.001868	643.3
9,000	.001811	640.9
10,000	.001755	638.6
15,000	.001496	626.7
20,000	.001266	614.6
25,000	.001065	602.2
30,000	.000889	589.5
35,000	.000737	576.6
36,089*	.000706	573.8
40,000	.000585	573.8
45,000	.000460	573.8
50,000	.000362	573.8
55,000	.000285	573.8

* Geopotential of Tropopause

Tabela 2: I.C.A.O. - tabela standardnih atmosferskih tlakov (https://www.grc.nasa.gov/www/k-12/WindTunnel/Activities/lift_formula.html, 8. 11. 2018)



Graf 1: Odvisnost koeficienta vzgona od vpadnega kota(http://airfoiltools.com/airfoil/details?airfoil=goe285-il#polars, 3.1.2019)



Slika 9: Stranski ris lastnega, modela letala pri vpadnem kotu 5° (vir lasten)

3.1.5 Preverjanje težišča na fizičnem modelu

Po končani izgradnji letala sva nadaljevala s preverjanjem težiščne točke, ki je za stabilno letenje izjemnega pomena. Klasično letalo ima ponavadi težiščno točko na prvi tretini profila krila. V grobem sva to točko preverila s tem, da sva celotno letalo podprla na omenjenem mestu na obeh krilih in letalo se je uravnovesilo.
Težišče sva preverila tudi z računanjem. Mentor nama je predstavil tehniko določanja te točke na pravem letalu, kar sva na najinem modelu preizkusila tudi sama.
Najprej sva izmerila težo na vsakem kolesu posebej, nato pa sva dolžino od sprednjega roba krila do najine točke izračunala po spodnji enačbi.
Izmerjeni podatki:

Fg_d = 11,5 N
Fg_l = 11,5 N
Fg_z = 1,27 N
d_d = 0,008 m
d_l = 0,008 m
d_z = 1,115 m

Fg_d … Sila teže izmerjena na desnem kolesu [N]
Fg_l … Sila teže izmerjena na levem kolesu [N]
Fg_z … Sila teže izmerjena na zadnjem kolesu [N]

d_d … Dolžina med sprednjim robom krila in desnim kolesom [m]

d_l … Dolžina med sprednjim robom krila in levim kolesom [m]

d_z … Dolžina med sprednjim robom krila in zadnjim kolesom [m]

M_d … Moment na desnem kolesu [Nm]

M_l … Moment na levem kolesu [Nm]

M_z … Moment na zadnjem kolesu [Nm]

CG … Dolžina med sprednjim robom krila in težiščem [m]

$$M_d = Fg_d \cdot d_d = 11{,}5N \cdot 0{,}008m = 0{,}092\,\text{Nm}$$

$$M_l = Fg_l \cdot d_l = 11{,}5N \cdot 0{,}008m = 0{,}092\,\text{Nm}$$

$$M_z = Fg_z \cdot d_z = 1{,}27N \cdot 1{,}115m = 1{,}416\,\text{Nm}$$

$$CG\ [\text{m}] = \frac{\Sigma M\ [Nm]}{\Sigma Fg\ [N]}$$

$$CG = \frac{Md\ +\ Ml\ +\ Mz}{Fgd\ +\ Fgl\ +\ Fgz}$$

$$CG\ = \frac{0{,}092\ Nm + 0{,}092\ Nm + 1{,}416\ Nm}{11{,}5\ N + 11{,}5\ N + 1{,}27\ N} = 0{,}066\ \text{m}$$

$$CG\ = 0{,}066\ \text{m}\ = 6{,}6\ \text{cm}$$

Če želimo, da je letalo dobro vodljivo mora biti CG pred centrom vzgona. Pri modelih letal je nezapisano pravilo, da je to ponavadi na prvi tretjini širine krila. Premer najinega profila krila znaša približno 20 cm, kar pomeni, da je izračunani CG ustrezen.

3.1.6 Kubična obremenitev krila

Kubična obremenitev krila oz. WCL (angl. wing cubic loading) je indikator za sortiranje radijsko vodenih maket po njihovih karakteristikah letenja. WCL enako kot površinska obremenitev krila ne upoštevata aerodinamike, ki je potrebna, da letalo sploh leti. Idealno bi moral biti WCL modela enak WCL parametru pravega letala, če želimo posneti njegove letalske karakteristike. V tabeli 3 je zbranih nekaj primerov letal s pripadajočo kubično obremenitvijo.

$$WCL = \frac{teza\ (oz.)}{površina\ kril\ (sq.\ ft.)^{1.5}}$$

Tip letala	WCL [oz./sq. ft.]
L - 13 Blanik	6
Najin model	9.9
Cessna 182	21.6
Pipistrel Alpha trainer	19

Tabela 3: Izračunan *WCL* različnih pravih letal (vir lasten)

WCL za prave modele letal je izračunan glede na maksimalno vzletno težo.

Za presek krila sva vzela prerez Gottingen 285 (priloga 3), ker ima lastnosti počasnega vadbenega letala. Profil sva nato preko SVG datoteke prenesla v CAD programu Fusion 360 in narisala celotno konstrukcijo krila. Za bolj trpežno konstrukcijo smo se skupaj z mentorjem odločili, da bomo za krilo uporabili dve nosilni palici iz smrekovega lesa, ki je primeren za to aplikacijo, ker je relativno lahek in prožen.

Vsak profil posebej sva nato izvozila v DXF format tako, da sva jih lahko potem izrezala iz balza lesa na laserskemu CNC rezalniku. Po končani izdelavi delov, sva krilo zlepila z epoxy lepilom (priloga 4) in ga na koncu prevlekla s posebno folijo za prekrivanje letalskih kril. Krilo v tej fazi prikazuje slika 10. Folija je samolepilna in ima dve strani. Na eni strani je lepilo, ki se aktivira, ko ga segrejemo, hkrati pa se ob segrevanju folija napne in ustvari površino krila.

Po končani izdelavi kril sva se lotila izdelave trupa (priloga 5). Glavna naloga trupa je, da je uporaben prostor v kabini dovolj velik, da lahko vanj zloživa vsa potrebna elektronska vezja, hkrati pa mora biti dovolj trpežen, da zdrži trša pristajanja in seveda, da je enostaven za izgradnjo in popravila. Trup sva pravtako izrezala na CNC laserskem rezalniku iz lipove vezane plošče debeline 3mm. Izrez je prikazan v prilogi 6.

Slika 10: Izdelava kril modela (vir lasten)

Za rep letala sva izbrala lahko palico iz aluminjeve zlitine, saj sva na ta način privarčevala na skupni teži letala ter hkrati zagotovila, da se rep ne bo zlomil pri tršem pristajanju. Zadnji stabilizator sva izdelala iz lahkega balza lesa, saj sva želela, da bi bil rep čim lažji. Specifična teža balza lesa znaša približno 140 kg/m^3, medtem ko znaša specifična teža lipovega lesa približno 530 kg/m^3. Teža repa pomembno vpliva na težišče letala, ki mora biti pred točko vzgona. V nasprotnem primeru letalo ni stabilno. V programu Fusion360 sva v ta namen uporabila funkcijo težišča (angl. center of mass), ki avtomatsko izračuna težišče objekta glede na specifično težo, ki je opredeljen v programu. Skupna teža letala, ki nama jo je izračunal program znaša 2165.53 g, pri čemer je že upoštevana teža pogonskega sklopa. Kontrolne površine sva pravtako izdelala iz balza lesa in so krmiljene s pomočjo malih servo mehanizmov tipa 9 g. Podvozje letala sva izdelala iz jeklene vzmetne žice premera 3 mm, ki je izpostavljena zvijanju navzven in na ta način deluje

kot vzmetenje letala pri pristajanju. Končnano konstrukcijo letala na pisti letališča prikazuje slika 11.

Slika 11: Letalo pripravljeno za vzlet (vir lasten)

3.1.7 Tabela osnovnih mehanskih lasnosti najinega modela

Za lažjo predstavo sva glavne specifikacije o letalu zapisala v tabeli 4.

Tabela specifikacij letala	
Teža letala pripravljenega za vzlet	2466 g
Premer letala čez krila	2000 mm
Dolžina kril	2 x 950 mm
Površina kril	2 x 21.85 dm^2
Višina profila krila	25 mm
V-lom	3.5 °
Širina horizontalnega stabilizatorja	675 mm
Višina navpičnega stabilizatorja	200 mm
Dolžina letala	1430 mm
Premer kabine	100 mm
Dolžina kabine	520 mm
Tip propelerja	13 inch x 6 inch
Premer pristajalnih koles	90 mm

Tabela 4: Osnovne mehanske lastnosti letala (vir lasten)

3.2 Krmilno vezje

”Možgani” najinega krmilnega vezja so nedvomno mikrokrmilniške ploščice Arduino. Ker je najin projekt dokaj kompleksen, sva se odločila, da bova za samo obdelavo podatkov uporabila več teh mikrokrmilnikov, ki med sabo komunicirajo, saj bi bila izvedba s samo eno mikrokrmilniško ploščico zelo težko izvedljiva. Izvedla sva komunikacijo gospodar-suženj (angl. master-slave), kar pomeni, da podatke s senzorjev bereva na več slave Arduino mikrokrmilnikih ter jih pošiljava preko Serial komunikacije (Tx, Rx) na glavni (angl. master) mikrokrmilnik Arduino, ki glede na prejete podatke krmili servo motorje in glavni pogonski motor.

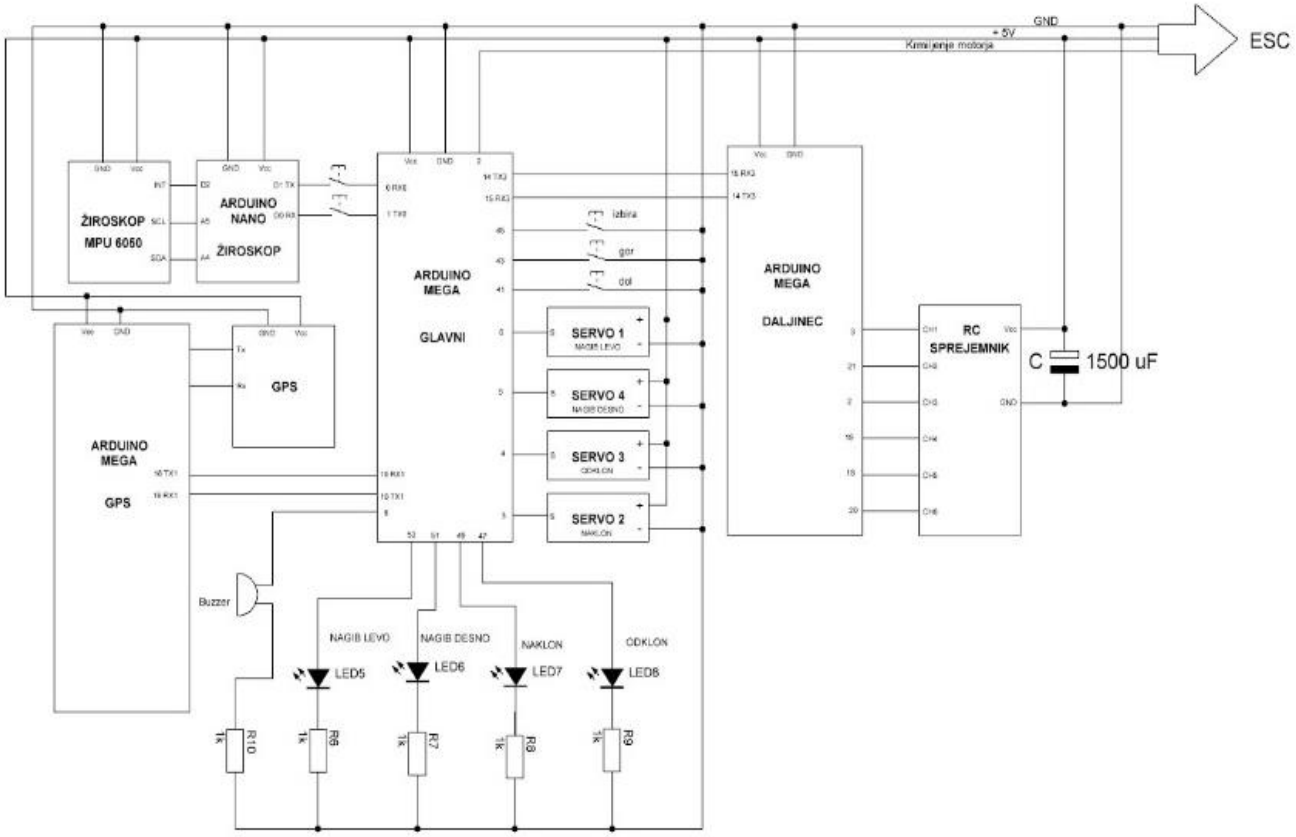
Na začetku sva sistem želela opremiti s kar nekaj senzorji, kot so žiroskop, GPS modul in ultrazvočni senzor. Zaradi kompleksnosti izvedbe in pomanjkanja časa, sva se odločila, da se bova osredotočila na sprejem podatkov iz dveh območij. Uporabila sva žiroskopski senzor, ki nama daje podatke o nagibu in naklonu letala in ker sva letalo želela upravljati tudi ročno tj. z uporabo daljinca, sva v krmilno vezje vključila radijski oz. RC sprejemnik. Uporabo ostalih senzorjev sva si zadala v bodočnosti, zaenkrat sva želela izvesti čimboljše krmilje z uporabo žiroskopa in radijskega sprejemnika.

Tako sva se dogovorila, da bova za glavni Arduino izbrala Arduino Mega, za podrejeni oz. suženj Arduino za branje podatkov žiroskopa, pa sva uporabila Arduino Nano in še en slave, natančneje Arduino Mega za branje podatkov radijskega sprejemnika. Razlog, da sva za branje podatkov sprejemnika izbrala mikrokrmilnik Arduino Mega je v tem, da edini vsebuje 6 prekinitvenih priključkov (angl. interrupt pins), ki so bolje opisani v poglavju 3.2.2.

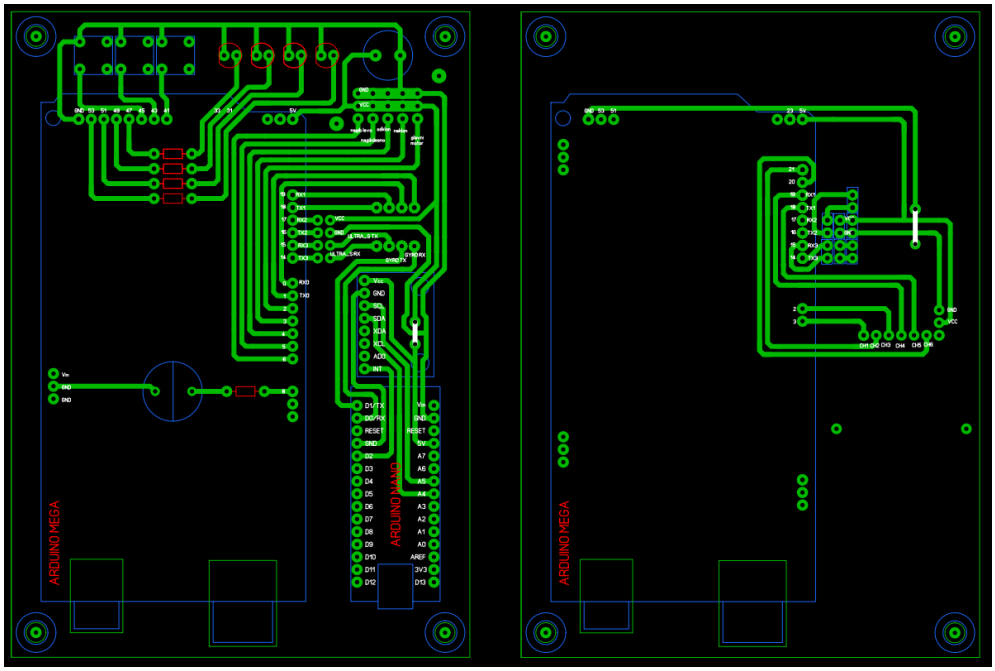
Naloga glavnega Arduina je ta, da prebere podatke na svojih Serial priključkih ter nato preračuna podatke. Neposredno iz tega Arduina krmiliva servo motorje, ki so zasluženi za premik zakrilc letala ter tudi glavni pogonski brezkrtačni enosmerni motor.

Elektroniko in motor skupaj napajava s 4 celično Lipo baterijo kapaciteto 3800 mAh, ki nama nudi dovolj energije, da je letalo zmožno leteti nekje 20 minut. Ker sva model letala izdelala že preden sva fizično izdelala tudi krmilno vezje, je bilo treba vezje zasnovati tako, da ga je bilo možno vstaviti in pritrditi v letalo. Vezje sva najprej sestavila na testni ploščici ali bread boardu ter opravila testiranja, potem pa je bilo potrebno izdelati resno vezje. Sama sva izdelala vezje, ki sva ga narisala v programu Sprint Layout ter ga s

pomočjo CNC rezkalnika tudi naredila. Vezje, prikazano na sliki 13, je zaradi več komponent razdeljeno v dve nadstropji, saj sva za izdelavo uporabila plošče, ki imajo sloj bakra samo na eni strani. V načrtu, ki ga prikazuje slika 12, je vključen tudi GPS modul, kot sva že povedala, ga zaenkrat nisva uporabila. V prilogi 1 je prikazano tudi vezje vstavljeno v trup letala.



Slika 12: Shema vezja (vir lasten)



Slika 13: Vezje v programu Sprint Lay-Out (vir lasten)

3.2.1 Krmilnik Arduino

Arduino je mikrokrmilnik, zasnovan tako, da bi bil postopek z uporabo elektronike v multidisciplinarnih projektih čimbolj dostopen. Strojno opremo krmilnika sestavljajo: odprtokodna oblika plošče in 8-bitni mikrokontroler Atmel AVR ali 32-bitni Atmel ARM. Programska oprema je sestavljena iz standardnega programskega jezika, prevajalnika in zagonskega nalagalnika, ki se izvaja na mikrokrmilniku. Krmilnik predstavlja računalnik v malem, saj ko nanj naložimo program, deluje povsem samostojno in ni nujno povezan z računalnikom. Preko USB lahko nalagamo lastno programsko opremo, poleg tega pa se preko USB vrat tudi napaja, tako da ne potrebujemo dodatnih kablov. (https://zmaga.com/content.php?id=4394, 13. 2. 2019)

Za Arduino sva se odločila, ker ga je zaradi prijaznega razvojnega okolja in jezika C++ sorazmerno enostavno uporabljati in programirati.

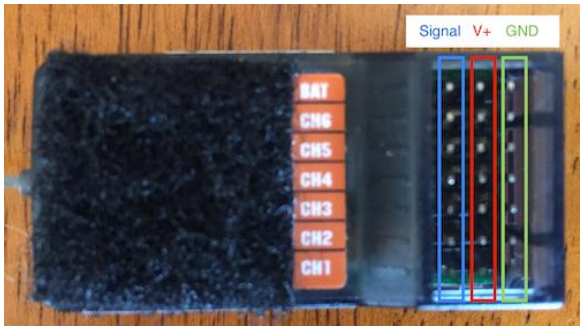
3.2.2 Prekinitveni priključki Arduina

Vsak mikrokrmilnik Arduino ima tako imenovane prekinitvene priključke (angl. interrupt pins). Arduino Uno in Arduino Nano imata oba po dva omenjena priključka (angl. pin), saj temeljita na procesorju ATmega 168/328, ki ima dva takšna priključka INT0 in INT1, ta pa sta povezana na 2. in 3. pin na krmilniku. Te pine lahko nastavimo, da sprožijo

določen odsek programa, ko signal na pinu pade iz logičnega stanja 1 na 0 ali pa zraste iz 0 na 1 (angl. trigger on RISING /FALLING signal edges). Ti priključki so opredeljeni s strani strojne opreme in zato reagirajo veliko hitreje od napisanega programa. Prekinitveni priključki so pri krmilniku Arduino Mega: pin 2, 3, 18, 19, 20 in pin 21. Omenjeni priključki so zelo uporabni, saj so zelo hitri in lahko rešijo časovno omejene zahteve v našem programu. Dober primer njihove uporabe je pri branju rotacijskega dekoderja. Če želimo, da program nikoli ne zgreši impulza iz dekoderja uporabimo zgoraj omenjen priključek. Na ta način lahko krmilnik opravlja druge naloge in hkrati preverja pulse na teh priključkih. Če zazna spremembo stanja na definiranem pinu, se ne glede na to kje je program ostal, začne izvajati, program za prekinitvene priključke. (http://playground.arduino.cc/code/interrupts). To funkcionalnost sva v najini aplikaciji izkoristila za branje dolžine impulzov, ki prihajajo iz radijskega sprejemnika. Na ta način ne izpuščava nobenega izmed podatkov, hkrati pa lahko v vmestnem času pridobljene podatke pošiljava naprej po vodilu.

3.2.3 Modelarski radijski sprejemnik

Komunikacijo med uporabnikom in letalom sva izvedla s pomočjo standardnega, radijskega, modelarskega oddajnika in sprejemnika, ki ga prikazuje slika 14. Vsak oddajnik ima svoj sprejemnik, ki je izdelan tako, da lahko nanj povežemo servo motorje.



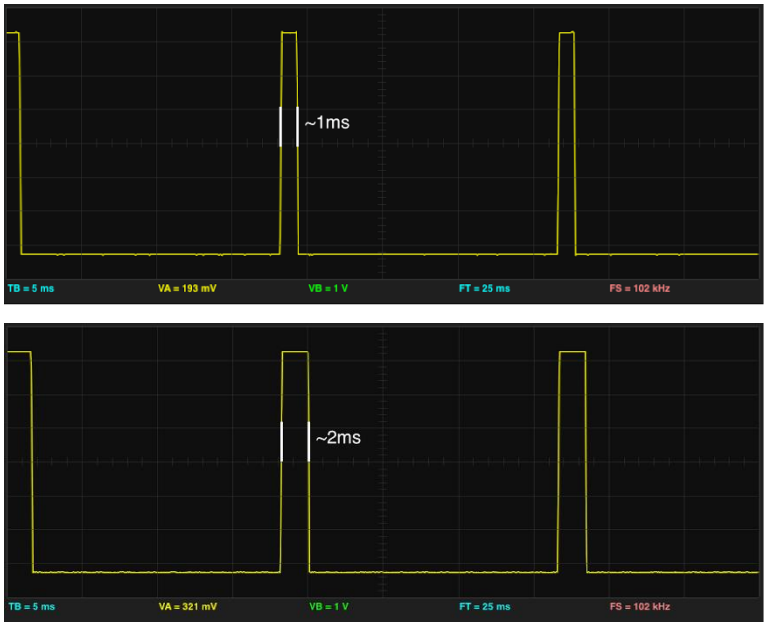
Slika 14: Radijski sprejemnik (https://ryanboland.com/blog/reading-rc-receiver-values/, 15. 1. 2019)

Vsak modelarski, servo motor ima tri vodnike. Prvi je beli ali rumeni vodnik in je signalni vodnik. Signal predstavlja pulzno pozicijsko moduliran signal oz. PPM (angl. pulse

position modulation). Naslednji rdeči vodnik je napajanje (DC 5V), zadnji, črn vodnik, pa predstavlja GND (0V).

Sprejemnik oddaja pridobljene podatke v obliki PPM signala. Na vsake 20 milisekund se postavi izhodni signal v stanje 1 ali HIGH. To stanje lahko nato traja od 1 ms vse do 2 ms. Puls z dolžino 1ms pomeni minimum, 1.5 ms pomeni sredino, 2 ms pa pomenijo maksimum. Dolžina pulza nam indicira položaj kontrolne palčke na daljinskem upravljalniku.

Oddajnik, ki sva ga uporabila ima 6 kanalov, kar pomeni, da sprejemnik oddaja 6 podatkov v obliki PPM pulza. Delovanje PPM signala najlažje razumemo s pomočjo slike 15.



Slika 15: Potek PPM signala (https://ryanboland.com/blog/reading-rc-receiver-values/, 15. 1. 2019)

3.2.4 Napajanje

Za napajanje sva izbrala 3 celično LiPo baterijo, katere nazivna kapaciteta znaša 3800 mAh pri napetosti 11.1 V. V modelarstvu se tap baterije uporablja, saj imajo odlično razmerje med izhodno močjo in težo. Takšno baterijo sva izbrala, saj potrebuje motor za polno moč približno 12 V. S to baterijo napajava tako glavni, pogonski motor, kot tudi krmilnik in krmilne servo mehanizme.

3.2.5 Način regulacije kontrolnih površin

Preden sva se lotila samega koncepta programiranja, sva raziskala tudi, kakšna vrsta regulacije se bi v najinem sistemu najbolje izkazala.
Regulacija je proces, s katerim skušamo čim bolj slediti in se približati željeni vrednosti. Regulacija ima, za razliko od krmilja, povratno vezavo, s katero meri dejansko vrednost na izhodu in jo primerja z željeno vrednostjo. Izvedemo jo lahko na različne načine.
V najinem primeru uporabljava regulacijo za sledenje določene smeri letenja. Krmilje izvaja na letalu funkcijo preprostega proporcionalnega regulatorja. Regulator deluje tako, da sprejema realne vrednosti o naklonu, nagibu in smeri iz žiroskopskega senzorja in jih primerja z željenimi vrednostmi, ki jih dobi od krmilnika. Željene vrednosti zapiše v avtomatskem načinu krmilnik sam.
(https://ucilnica.fri.uni-lj.si/pluginfile.php/21024/mod_resource/content/1/
Regulacije.pdf, 5. 12. 2018)

Program sva kasneje zasnovala tako, da izračuna razliko med vrednostima, jo po potrebi množi s stalnim številom in jo nato pošlje do servo motorjev. Večja kot je razlika med vrednostima, večja je izhodna korekcijska vrednost.
Slabost takšne regulacije je, da se realna vrednost nikoli čisto ne pokrije z željeno veličino. Težava je v tem, da manjša kot je razlika med željeno in realno vrednostjo, manjša je vrednost na izhodu.
Težava takšnega regulatorja je tudi odzivnost. Vse te težave rešuje pulzni, integralni in diferencialni regulator oz. PID regulator. Kljub ugodnim lastnostim PID regulatorja se z njim v začetni stopnji nisva ukvarjala, saj so bili rezultati testiranj s proporcionalnim regulatorjem zadovoljivi.

3.2.6 Programski del

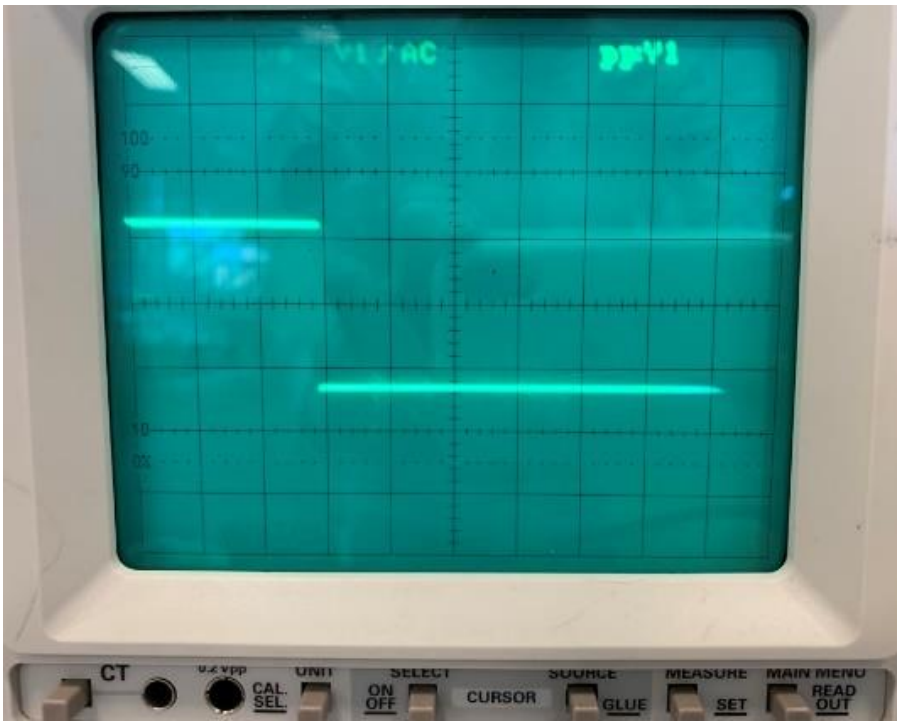
Program sva pisala vzporedno z izdelavo testnega vezja na testni plošči (angl. breadboard). Pisala sva ga v več delih, saj je najin sistem zaenkrat sestavljen iz treh krmilnikov Arduino. V sistemu je en glavni krmilnik, ki se ukvarja s sprejemanjem obdelanih signalov iz različnih senzorjev ter s krmiljenjem izhodnih servo mehanizmov. Prvi pomožni krmilnik bere dolžino impulzov iz radijskega sprejemnika in jih nato pošilja

po serijski povezavi do glavnega krmilnika. Drugi pomožni krmilnik je namenjen branju podatkov iz žiroskopskega modula MPU 6050. Program je razdeljen med tri krmilnike, saj so nekatere naloge, kot so branje signala iz radijskega sprejemnika časovno kritične in so zato veliko breme za relativno počasen krmilnik Arduino. Program je dodan k prilogam.

1. Del programa (Glavni krmilnik Arduino Mega).
   Program, ki sva ga zapisala za glavni krmilnik je precej preprost. Glavna funkcija tega dela programa je, da hitro in zanesljivo bere podatke, ki mu jih pošiljajo drugi krmilniki. S povezavo sva na začetku imela precej težav, zato sva se odločila, da bova za pošiljanje informacij uporabila knjižnico Easy Transfer. Knjižnica je zelo enostavna za uporabo in omogoča prenos podatkov po serijski povezavi med krmilniki. Omenjen program je napisan tako, da ima dva načina upravljanja letala. Prvi način je ročni način, pri katerem lahko letalo vodimo s pomočjo daljinskega upravljalnika. Avtomatski način je način, kamor zapišemo kodo, ki jo želimo testirati. Kodo za avtomatski način sva oblikovala tako, da lahko v programu definiramo željeni kot, po katerem se nato najino letalo ravna. V avtomatskem režimu dela, pogram najprej primerja željeni naklon in nagib z realnimi podatki, ki jih dobi iz pomožnih krmilnikov. Nato izračuna razliko med temi podatki in obdelano razliko pošlje na izhodne servo mehanizme.

2. Del programa se izvaja na krmilniku Arduino Nano. Branje podatkov s pomočjo modula MPU 6050 je s pomočjo različnih Arduino knjižnic v mirnem okolju precej enostavno, če pa izpostavimo takšen senzor vibracijam, ki delujejo na letalu, pa se pojavijo težave. Testirala sva veliko različnih knjižnic in načinov za branje podatkov s tega modula. Na koncu sva izbrala tako imenovano knjižnico Jeff Rowlberg. Kljub temu da je program, ki ga je napisal zelo kompleksen, sva ga izbrala, saj uporablja za branje kotov tako žiroskopski senzor kot tudi merilec pospeškov, oba senzorja pa se nahajata v omenjenem modulu. Ugotovila, sva, da lahko nepravilnosti pri merjenju, ki sicer nastanejo zaradi vibracij, skoraj popolnoma odpraviva z uporabo teh dveh senzorjev. Pridobljene podatke nato pošiljava do centralnega krmilnika

3. Del programa se izvaja na pomožnem krmilniku Arduino Mega. Ta tip krmilnika sva izbrala, saj ima Arduino Mega šest prekinitvenih priključkov za razliko od krmilnika Arduino Nano, ki ima le en omenjeni pin. Program na tem krmilniku je odgovoren za komunikacijo med radijskim sprejemnikom in sistemom avtopilota. To je še posebej pomembno za testiranje, saj mora imeti upravljalec v vsakem trenutku možnost pilotiranja letala v ročnem načinu v primeru napake na sistemu. Podatki iz radijskega sprejemnika prihajajo v obliki PPM signala, o katerem lahko več preberete v poglavju Pregled objav. Program je napisan tako, da ves čas spremlja interupt pine na krmilniku. Če eden izmed njih spremeni stanje iz 0 na 1 začne šteti in konča, ko signal na tem pinu spet pade v logično stanje 0. Čas, ki ga merimo znaša od 1ms do 2ms in je proporcionalen s položajem kontrolnih ročk na radijskem oddajniku (daljincu). Razpon tega podatka nato obdelava in ga takšnega pošljeva po serijski povezavi do centralnega krmilnika.



Slika 16: Prikaz PPM signala na osciloskopu (vir lasten)

Na sliki 16 je prikazan izhodni signal za krmiljenje servo mehanizmov, katerega sva izmerila z osciloskopom. Spreminjanje dolžine impulza sva spremljala s pomočjo

osciloskopa. Po končanem programiranju sva na ta način preverila, kako se v avtonomnem načinu letenja, signal spreminja v odvisnosti od spreminjanja naklona letala. Na zgornji sliki je letalo v nevtralnem položaju, zato dolžina impulza znaša 1,5 ms.

3.3 Rezultati

Najino raziskovanje avtopilota smo skupaj z mentorjem začeli z izdelavo radijsko vodenega modela letala. Več o izgradnji in načrtovanju letala lahko preberete v poglavju Mehanika letala. Naše prvo testiranje je bilo namenjeno preizkusu mehanske plati letala. Naš prvi uspeh nas je razveselil, ko je naše letalo s pomočjo daljinskega upravljalnika prvič zapustilo trdna tla. Ugotovili smo, da smo letalo zelo dobro narisali, saj se je letalo v zraku obnašalo po pričakovanjih. Zaradi V-loma, nekoliko daljšega repa in nekoliko večjega naklonskega stabilizatorja je bilo letalo zelo stabilno v zraku. Pravtako se je podvozje letala izkazalo kot dobro in je zelo dobro ublažilo silo pri pristajanju. Kljub temu smo na prvem poletu ugotovili, da ima naš model nekaj šibkih točk, ki pa smo jih do naslednjega testiranja popravili. Glavna slabost letala je bila šibka podpora aluminijaste palice, ki je začela kazati vidne razpoke že pred poletom. Seveda smo napako hitro odpravili. V programu sva slabe lastnosti omenjenega dela dopolnila in nato del natisnila na 3D-tiskalniku iz ABS plastike. Pri prvem poletu sva letalo upravljala izključno preko radijskega daljinca. To pomeni, da je sistem deloval v ročnem načinu, saj sva avtomatski način letenja želela postopoma dodajati kasneje, po uspešnem testiranju letala v zraku. Servo motorji so bili povezani direktno v modelarski radijski sprejemnik, ta pa jih je vodil glede na sprejete podatke s tal. Krstni polet sva izvedla na letališču Šoštanj. Polet je bil zelo uspešen in nam je dal dodatno motivacijo za nadaljnje delo.

Slika 17: Prvi uspešen polet (vir lasten)

Pri drugem poletu sva se osredotočila predvsem na delovanje žiroskopskega senzorja. Najin cilj drugega poleta je bil, da bi letalo po preklopu v avtomatski način samodejno vzdrževalo trenutno smer leta. Po uspešni izdelavi krmilnika sva morala določene parametre v programu kalibrirati glede na realni, mehanski del letala. S pomočjo testnega programa na krmilniku sva nastavila kalibracijski faktor za izhodiščna stanja vsakega zakrilca posebej. Po končani kalibraciji sva napisala program za samodejno regulacijo nagiba letenja.

Najin krmilnik sva ponovno preizkusila na letališču Šoštanj. Z letalom sva vzletela, tako kot prejšnjič, s pomočjo radijskega oddajnika. Nato sva letalo ročno vodila na varno višino in prvič vklopila avtomatski način. Po kratkem avtonomnem letu sva ugotovila, da se letalo pravilno odziva na zunanje dejavnike. Tudi to testiranje je bilo uspešno, saj sva ugotovila, da krmilnik zelo dobro regulira nagib letala. Pri regulaciji naklona letala pa sva naletela na težavo.

Krmilnik na letalu je sicer pravilno vzdrževal nagib letala, vendar je letalo pri tem nagibu letenja izgubljalo višino. Težava je v vpadnem kotu leta (angl. angle of attack). Letalo potrebuje za letenje določen vzgon. Vzgon je pri vzdrževanju konstantne višine ves čas enak teži letala, je odvisen od hitrosti zračnega toka, ki teče ob krilu, od vpadnega kota pod katerim se krilo pomika skozi zrak in od mnogih drugih spremenljivk.

Če upoštevamo prejšnjo poved, lahko ugotovimo, da se kot, pri katerem letalo leti, spreminja v odvisnosti od hitrosti zračnega toka mimo kril.

Težavo bi lahko rešila z uporabo variometra, s katerim bi lahko zaznavala morebitno spreminjanje tlaka. Sprememba zračnega tlaka je v odvisnosti od spremembe višine. Na ta način bi lahko krmilnik samodejno kompenziral izgubo višine s tem, da bi povečal vpadni kot letala.

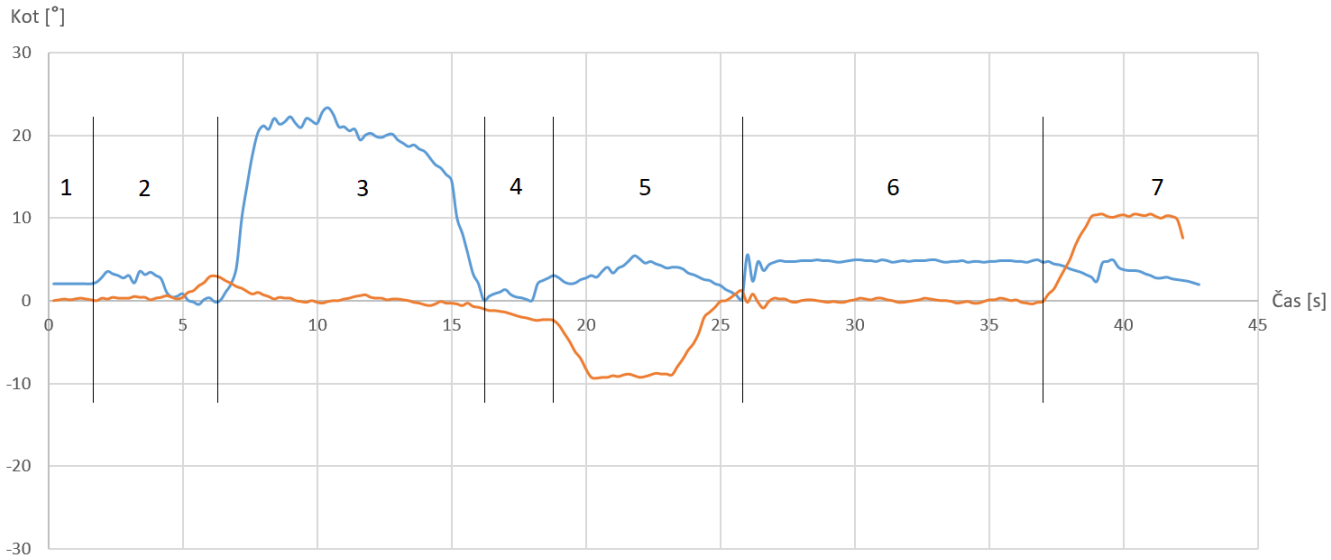
Odstopanje pri vzdrževanju ustrezne višine sva začasno odpravila s tem, da sva vpadni kot letala za določeno hitrost izračunala s pomočjo formule za izračun vzgona na letalskem krilu.

S pomočjo formule za izračun vzgona sva ugotovila, da mora letalo leteti pod vpadnim kotom približno 5 stopinj, da lahko leti na konstantni višini pri hitrosti 40km/h.

Pred naslednjim testiranjem sva teh 5 stopinj, pod katerimi mora letalo leteti, kompenzirala v programu.

Pri tretjem testiranju sva se ponovno osredotočila na vzdrževanje naklona in nagiba letala. Za lažje razumevanje dogajanja na letalu sva dobila sledečo idejo. Podatke sva med letom merila s pomočjo telefona, na katerega sva naložila aplikacijo Gyro. Telefon sva pred poletom namestila na trup letala. Na sliki 18 je prikazan model med tem poletom. Telefon je pritrjen na vrhu trupa. Z aplikacijo sva zajemala podatke o nagibu in naklonu letala, aplikacija pa je te parametre zapisala v obliki tabele. Podatke sva nato prenesla v program Exel. Rezultate sva preučila ter narisala graf 2 in graf 3.

Slika 18: Zbiranje podatkov o nagibu in naklonu s pomočjo telefona (vir lasten)



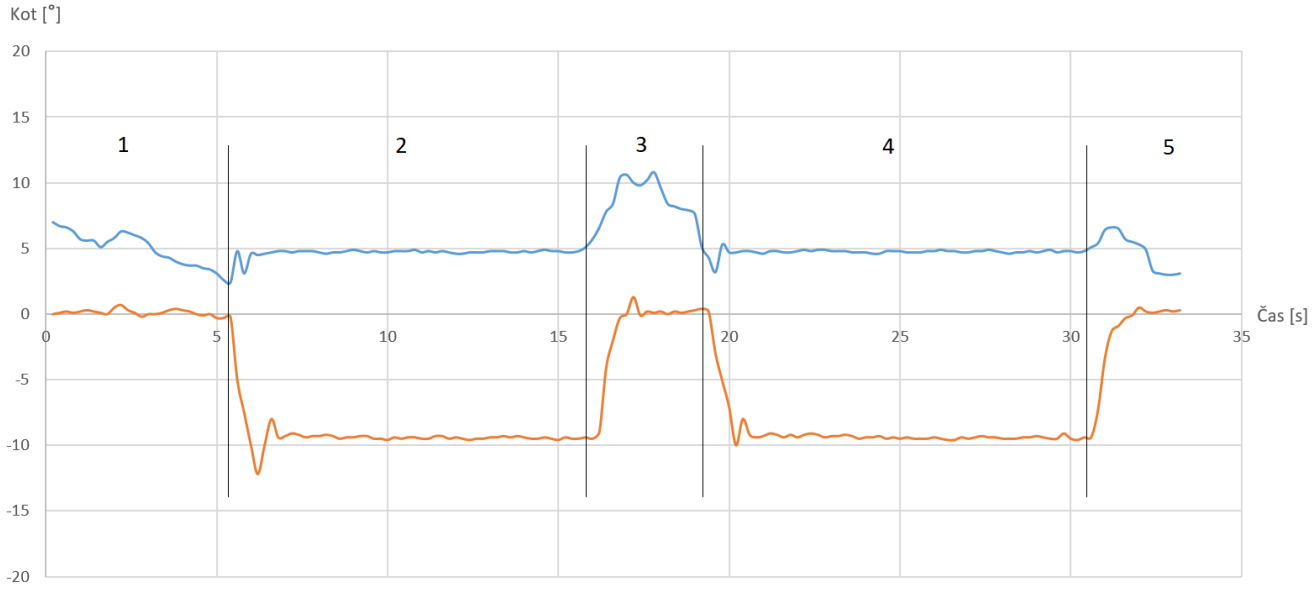
Graf 2: Zajeti podatki pri tretjem testiranju (vir lasten)

Graf 2 sva razdelila na več delov. Vsak izmed njih prikazuje določeno stopnjo med poletom. Zgornji graf sva razdelila na sedem delov in vsakega izmed njih označila s številko.

1. Odsek predstavlja letalo v mirovanju na tleh. Iz grafa je razvidno, da znaša nagib letala 0°, naklon letala pa znaša približno 3°. Odstopanje naklona od ničle na začetku je normalno, zaradi same konstrukcije podvozja. Letalo je na ravni podlagi pod kotom, zaradi lažjega vzletanja.
2. Odsek prikazuje letalo, ki pospešuje na stezi pred vzletom. Med pospeševanjem letalo razvije določeno hitrost ter se poravna, kar je razvidno na grafu pri 5. sekundi. Nato se rep letala spet spusti, vzgon, ki pa nastane, pa povzroči, da se letalo začne dvigovati.
3. Odsek na grafu prikazuje dviganje letala. Kot, pod katerim letalo leti, se poveča, s čimer se poveča tudi vpadni kot letalskega krila, ki posledično poveča vzgon.
4. Odsek prikazuje letalo, ki se je po osi naklona poravnalo s tlemi in leti naravnost.
5. Odsek prikazuje letalo v zavoju. Zavoj pri letalih izvedemo tako, da nagnemo letalo na poljubno stran in z višinskim krmilom povzročimo, da začne letalo zavijati.

6. Odsek prikazuje delovanje avtopilota. Avtopilot je zasnovan tako, da ga vklopimo na varni višini. Od vklopa avtopilota naprej upravljalec ne more več nadzirati letala, saj to leti v avtonomnem načinu. Iz grafa je razvidno, da ostajata nagib in naklon letala v avtomatskem načinu konstantna. Nagib letala sledi kotu 0°, naklon letala pa znaša približno 5°. Naklon 5° je izračunan naklon, ki zagotavlja, da leti letalo na konstantni višini pri zračni hitrosti 40km/h.

Najin četrti polet je bil namenjen testiranju zadrževalnega manevra (angl. holding).



Graf 3: Zajeti podatki pri četrtem testiranju (vir lasten)

Graf 3 sva podobno kot prejšnjega razdelila na več delov, ki prikazujejo posamezne faze poleta. Za razliko od prejšnjega preizkusa, kjer sva pokazala tudi vožnjo po tleh ter sam vzlet, se tukaj s tem nisva ukvarjala, saj naju je bolj zanimalo obnašanje pri kroženju.

1. Odsek prikazuje prosto vožnjo letala s pomočjo daljinca. Letalo sva predhodno spravila na neko višino ter mu potem postopoma manjšala naklon, da se je spustil malo nižje. Pri tem je letalo letelo sorazmerno naravnost, saj je nagib letala zajemal vrednosti okoli 0°.
2. Odsek se začne s trenutkom vklopa avtopilota, ki ga prikazuje tudi slika 19. Pogoj, da letalo kroži je ta, da se nagib letala spremeni na določeno vrednost hkrati pa

mora naklon letala ostati konstanten. Iz grafa je razvidno, da je avtopilot lepo zadrževal naklonski kot 5°, kot nagiba pa je znašal 10°. Kot nagiba sva naključno izbrala. Pri kroženju sva naletela na težavo pri vzdrževanju višine leta, kar sicer ni razvidno iz grafa. Težava je v spremembi parametrov med krožnim letom. Med izvajanjem tega manevra, se letalo nagne, pri tem pa se zmanjša sila vzgona, ki deluje pravokotno na podlago. Za vzdrževanje višine pri tem manevru, je potreben večji vzgon, ki je povezan z večjim vpadnim kotom. S pomočjo variometra bi to težavo odpravila, saj bi glede na prebrane podatke lahko vplivala na vpadni kot leta ali na hitrost leta. S tem bi se povečal vzgon letala. Krog kroženja je prikazan na sliki 19.

3. Odsek grafa prikazuje vmesten manever, ki je bil izveden s pomočjo daljinskega upravljalnika. V tem delu, sva letalo usmerila proti sebi, da ne bi ušlo iz vidnega polja.
4. Odsek prikazuje ponovni vklop avtopilota za izvajanje zadrževanja v zraku.
5. Odsek prikazuje ponoven preklop krmilnika v ročni način.

Slika 19: Prikaz kroženja letala od zgoraj (vir lasten)

3.3.1 Analiza hipotez

Pred začetkom raziskovanja sva si zastavila tri hipoteze.

Prvo hipotezo sva delno potrdila, saj sva ugotovila, da združuje modul MPU6050 žiroskopski senzor in senzor za merjenje pospeška. S tema dvema senzorjema lahko zelo natančno merimo nagib in naklon letala, težave pa se pojavijo pri merjenju smeri. Omenjeni senzor pri merjenju smeri nima referenčne točke, tako kot jo ima pri merjenju nagiba in naklona v obliki gravitacijskega pospeška. Zaradi te lastnosti se pri merjenju smeri sčasoma pojavi odstopanje, ki mu pravimo (angl. Z axis drift). Težavo bova rešila z uporabo kompasa ali pa z uporabo GPS modula. Med testiranjem nama je uspelo, da je letalo uspešno sledilo izbrani smeri.

Drugo hipotezo sva ponovno delno potrdila. Uspelo nama je, da je letalo samodejno krožilo pod konstantnim nagibom in konstantnim naklon letala, vendar je pri tem izgubljalo višino. To napako sva kompenzirala z krmiljenjem višinskega krmila, podoben učinek pa bi lahko dosegla tudi z dodajanjem moči pogonskega motorja.

Tretjo hipotezo sva v celoti potrdila, saj sva kljub mnogim težavam ugotovila, da je izdelava preprostega avtopilota z uporabo cenovno ugodnih komponent vsekakor izvedljiva. Seveda najin prvi model krmilnika še ni tako izpopolnjen, da bi lahko sam prevzel zahtevnejše naloge, vsekakor pa je že v sedanji stopnji zelo uporaben za začetnike, ki se želijo naučiti pilotiranja radijsko vodenih modelov letal.

Za začetnike je značilno, da pogosto izgubijo občutek za orientacijo letala, kar je razlog, da se začne letalo nekontrolirano spuščati. V takem primeru pride najin krmilnik do izraza, saj se letalo samodejno poravna, če sprožimo avtonomen način letenja.

Hipotezo lahko podpreva še z dejstvom, da cena najinega krmilnika v trenutnem stanju znaša 19.96 EUR, kar je razvidno iz tabele 5.

Komponenta krmilnika	Cena komponente v EUR (Aliexpres)
MPU 6050	0.68
Arduino Mega (2x)	6.38
Arduino Nano	1.52
Izdelava vezja in drobne komponente	5
Skpupna cena	19.96

Tabela 5: Cena posameznik komponent krmilnika (vir lasten)

4 RAZPRAVA

Veliko težav, ki so nama prišle nasproti sva rešila. Kljub temu pa nama je zaradi pomanjkanja časa ostalo še nekaj izzivov in izboljšav. Ob začetku leta sva bila osredotočena predvsem na mehanski del letala. Najin prvi uspeh je bil, ko je letalo poletelo, čeprav je to bil šele začetek naloge. Začela sva z izdelavo krmilnega sistema, ki se je postopoma razvijal skozi tri različice preden je delal po pričakovanjih. Kar nekaj težav sva imela z zanesljivostjo sistema, vendar sva jih uspešno odpravila. Najin naslednji velik uspeh je bil, ko je najino letalo prvič avtonomno preletelo 70 m v zaželjeni smeri. Po vseh težavah in zapletih, ki sva jih uspešno rešila, so se nama takrat odprla vrata za mnoge izboljšave in, ki pa jih bova nedvomno realizirala v prihodnosti.

Prva izboljšava je **uporaba variometra (VSI)** za korekcijo vpadnega kota letala oz. za vzdrževanje višine leta. Med izdelavo projekta sva ugotovila, da ima letalsko krilo veliko zunanjih parametrov, ki v veliki meri vplivajo na letenje letala, na katere na začetku nisva bila pozorna. S pomočjo variometra bi lahko na krajših razdaljah popravljala vpadni kot letalskega krila in na ta način vzdrževala višino leta. Začasno sva to težavo rešila z izračunom omenjenega kota letenja, ki sva ga vnesla v program. Ta rešitev seveda ni idealna, saj mora letalo v ta namen leteti s konstanto hitrostjo, napake pri vzdrževanju višine pa se pojavijo predvsem pri kroženju, kjer se letalo nagne, zaradi česar se posledično vzgon zmanjša.

Druga izboljšava je **uporaba kompasa**. Modul MPU-6050 je senzor, ki na enem čipu združuje pospeškometer in žiroskopski senzor. Ugotovila sva, da lahko s takšnim senzorjem zelo natančno merimo nagib ter naklon letala. Želela sva ga uporabiti tudi za spremljanje smeri, vendar sva ugotovila, da imajo takšni senzorji napako pri spremljanju Z-osi, ki ji pravimo (angl. Z-axis drift). Testiranja, ki sva jih izvedla so pokazala, da je napaka v okolju, kot je letalo, ki je polno vibracij, zelo velika in znaša približno 5 stopinj na minuto, kar je za najino aplikacijo neuporabno.

Krmilnik bi lahko reguliral tudi smer letenja po odklonski oz. smerni osi z uporabo kompasa. Modul, ki vsebuje takšen senzor je MPU-9255. To vezje vsebuje žiroskopski senzor, pospeškometer in kompas. Senzor sva že kupila in ga bova v prihodnje zamenjala s starim modulom MPU-6050.

V uvodu sva zapisala, da je eden izmed najinih ciljev tudi ta, da bi letalo avtonomno letelo do določene kordinate. V prihodnje želiva to tudi izvesti s pomočjo **GPS modula**. GPS modul sva že preizkusila, vendar nama ga zaradi pomanjkanja časa še ni uspelo implementirati.

Pravtako sva v najin sistem želela vključiti tudi **ultrazvočni senzor**, ki bi nam pomagal pri pristajanju, ki je prikazano na sliki 20. Tudi to je ostal izziv za v prihodnje.

Slika 20: Letalo tik pred pristankom (vir lasten)

5 ZAKLJUČEK

Raziskovalna naloga, ki sva si jo zastavila, nedvomno zajema poznavanje veliko različnih področij. Najino delo je bilo ves čas prepleteno s poznavanjem veliko različnih področij. Oplemenitila sva svoje znanje na področju elektrotehnike, računalništva ter strojništva strojništva, prav tako pa sva morala dobro poznati področja same konstrukcije letal, aerodinamike letal in teorije letenja letal v instrumentalnih pogojih. Elementi kot so programiranje mikrokrmilnika Arduino, izbira motorjev, izbira ustrezne baterije, načrtovanje in izdelava vezja itd., ne bi bili dovolj, če se projektu ne bi dovolj dobro posvetila tudi z vsemi prej omenjenimi področji, ki zadevajo letalstvo. Ta so zajemala optimalno konstrukcijo letala, izdelavo podvozja, pravilno razmerje med razponom kril in dolžino letala, mehansko povezavo s servo motorji ipd. Svoje znanje sva ves čas dopolnjevala na podlagi reševanja težav, na katere sva med delom naletela. Naučila sva se, kako poiskati čimbolj optimalne rešitve za rešitev problemov, kar je danes v znanosti

in tehnologiji precej zaželjeno. Pri delu sva bila zelo dosledna, saj bi vsaka manjša napaka ogrozila varno letenje modela. Samo letalo je mehansko zelo dodelano, kar gre pripisati temu, da sva pri delu uporabljala tudi sodobne stroje kot so 3D printer, laserski rezalnik in CNC rezkalnik. Rezultat tega je bil, da je sama konstrukcija bila dovolj dobra in uspešno prestala vse preizkusne polete. Povdariva lahko, da je letalo skozi preizkusne polete ostalo nepoškodovano. Preizkusa vzdrževanja smeri ter izvajanje zadrževalnega manevra, sta pokazala, da je najino krmilno vezje zanesljivo in dobro za opravljanje teh funkcij, čeprav vsebuje še kar nekaj pomanjkljivosti in možnosti izboljšanja. Kljub zaključeni raziskovalni nalogi, to še zdaleč ne pomeni, da je najino delo zaključeno, ampak to predstavlja dobro popotnico za nadaljne delo.

6 POVZETEK

Najin sistem avtopilota bi bil uporaben na športnih letalih kot tudi na modelarskih letalih. S tem bi se bistveno povečala natančost pri vzdrževanju smeri kot tudi pri izvajanju zadrževalnega manevra. Velikokrat se zgodi, da pilot zaradi zasedenosti pristajalne steze ne more nemudoma pristati. Takrat pilot začne izvajati manever holding, kar pomeni da kroži nad letališčem tako dolgo, da dobi dovoljenje za pristanek. Z uporabo najinega sistema bi v takšni situaciji pilot s pritiskom na gumb vključil izvajanje avtopilota ter letalo bi začelo krožiti. V veliko pomoč bi pilotu bil tudi avtopilot pri vdrževanju smeri letenja. Pilot bi bil tako manj obremenjen in bi se lahko bolje posvetil spremljanju parametrov med poletom ter komunikaciji s kontrolorji. Pri daljših poletih, bi to bilo še toliko bolj koristno za pilota. Pri modelarskih letalih pa bi bilo to zelo koristno pri modelarskih začetnikih, saj bi preklop v avtomatski način letenja pri nekontrolirani vožnji v zraku, zopet poravnal letalo in tako preprečil nezgodo. Poleg tega je možno program za avtomatsko letenje spremeniti, kar bi vsakomur omogočilo, da bi lahko napisal program, ki bi v zraku izvajal različne figure in akrobacije. Z manjšo predelavo letala in prilagoditvijo najinega krmilnega vezja, bi bilo možno sistem uvesti na različnih letalih.

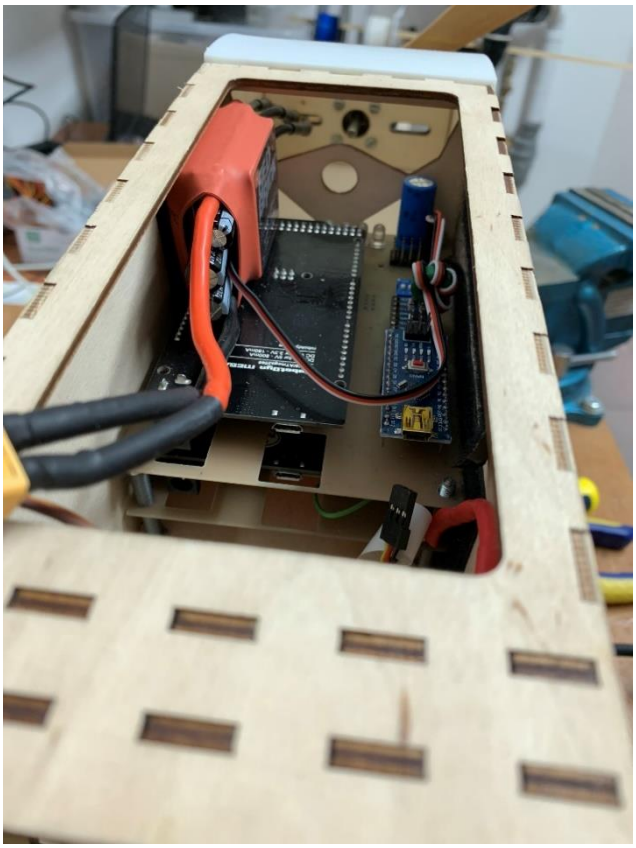
7 ZAHVALA

Za pomoč pri izdelavi raziskovalne naloge se zahvaljujeva:

- Mentorju Jožetu Lukancu, ki nama je pomagal ob vsaki težavi in bil v podporo. Tako smo vedno poiskali rešitve za vse izzive, ki so prišli naproti.
- Jožetu Hrovatu, ki nama je pomagal pri izdelavi mehanskega dela projekta.
- Učiteljici Nataši Meh Peer za lektoriranje.
- Učiteljici Jolandi Melanšek za lektoriranje angleške različice povzetka.
- Najinim prijateljem in sorodnikom za podporo pri nastajanju naloge.
- ŠC Velenje – Elektro in računalniški šoli, ki nama je nudila odlične pogoje za izdelavo raziskovalne naloge.

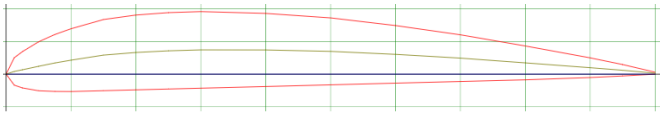
8 PRILOGE

Priloga 1: Krmilnik za izvajanje avtopilota (vir lasten)

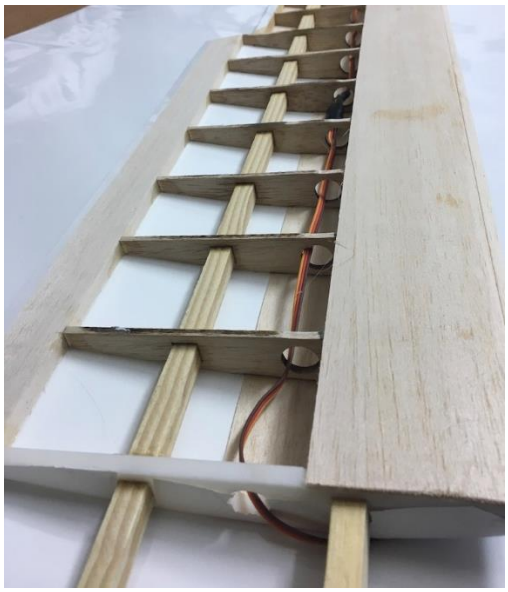


Priloga 2: Priprava na vzlet (vir lasten)

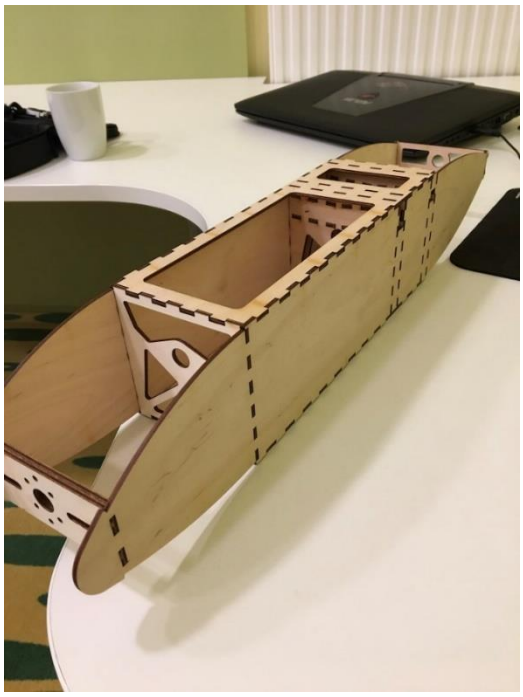
Priloga 3: Profil krila "Gottingen 285" (http://airfoiltools.com/airfoil/details?airfoil=goe285-il, 13. 2. 2019)



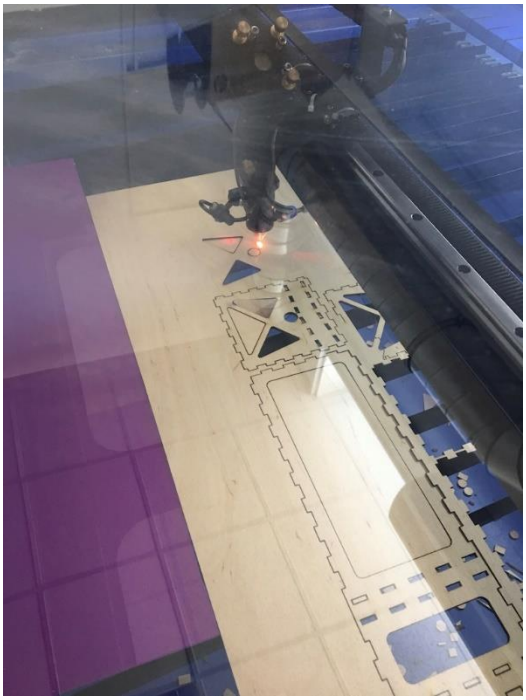
Priloga 4: Konstrukcija krila (vir lasten)



Priloga 5: Načrtovanje in izdelava trupa (vir lasten)



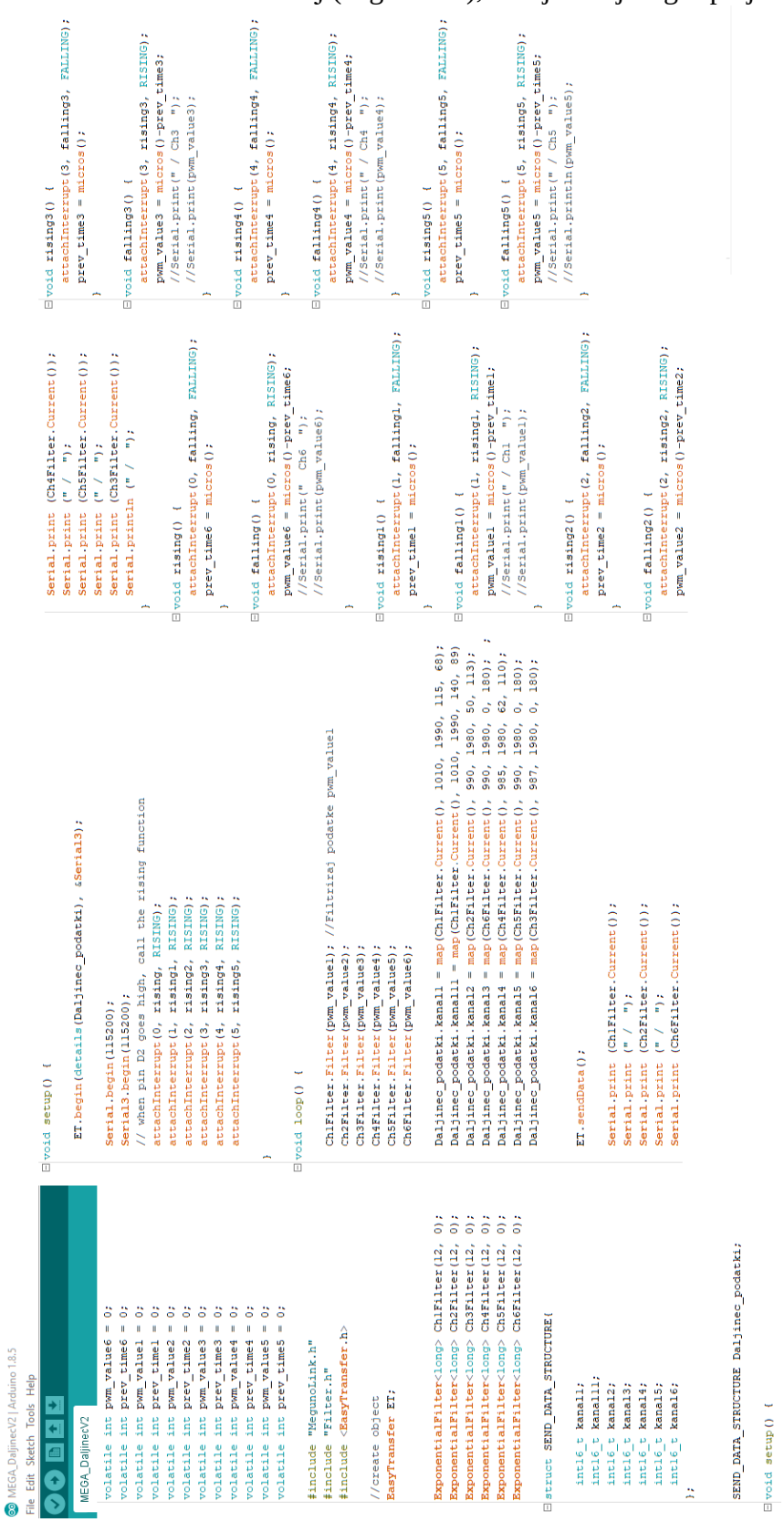
Priloga 6: Razrez vezane plošče z CNC laserjem (vir lasten)



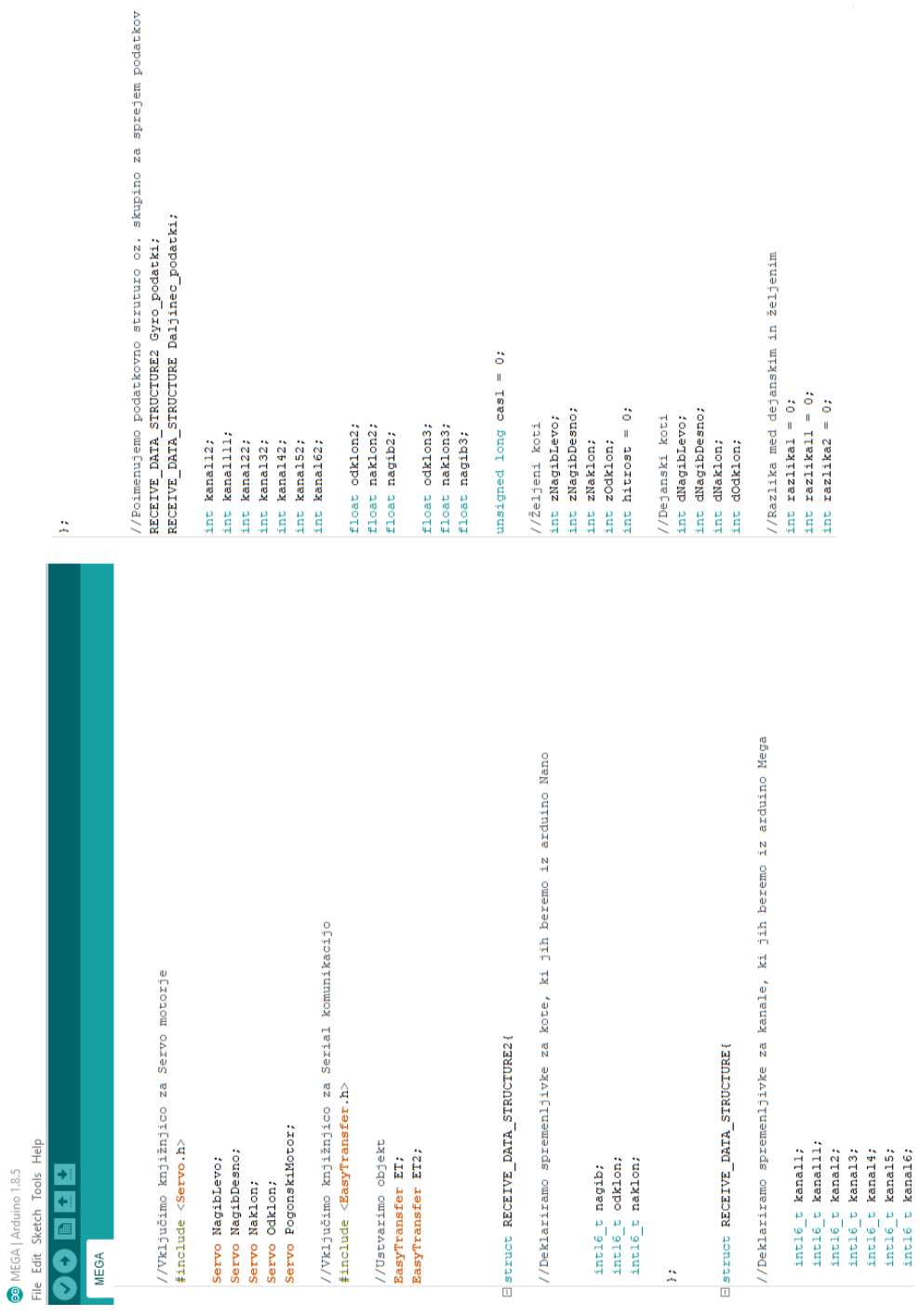
Priloga 7: Letalo med poletom (vir lasten)

Priloga 8: Program

Program za Arduino Nano suženj (angl. slave), branje radijskega sprejemnika:



Program za glavno (angl. master) mikrokrmilniško ploščičo Arduino.



```
//Poimenujemo podatkovno strukturo oz. skupino za sprejem podatkov
RECEIVE_DATA_STRUCTURE2 Gyro_podatki;
RECEIVE_DATA_STRUCTURE Daljinec_podatki;

  int kanal12;
  int kanal111;
  int kanal22;
  int kanal32;
  int kanal42;
  int kanal52;
  int kanal62;

  float odklon2;
  float naklon2;
  float nagib2;

  float odklon3;
  float naklon3;
  float nagib3;

  unsigned long cas1 = 0;

  //Željeni koti
  int zNagibLevo;
  int zNagibDesno;
  int zNaklon;
  int zOdklon;
  int hitrost = 0;

  //Dejanski koti
  int dNagibLevo;
  int dNagibDesno;
  int dNaklon;
  int dOdklon;

  //Razlika med dejanskim in željenim
  int razlika1 = 0;
  int razlika11 = 0;
  int razlika2 = 0;
};
```

```
MEGA | Arduino 1.8.5
File Edit Sketch Tools Help
MEGA

//Vključimo knjižnjico za Servo motorje
#include <Servo.h>

Servo NagibLevo;
Servo NagibDesno;
Servo Naklon;
Servo Odklon;
Servo PogonskiMotor;

//Vključimo knjižnjico za Serial komunikacijo
#include <EasyTransfer.h>

//Ustvarimo objekt
EasyTransfer ET;
EasyTransfer ET2;

struct RECEIVE_DATA_STRUCTURE2{
//Deklariramo spremenljivke za kote, ki jih beremo iz arduino Nano

  int16_t nagib;
  int16_t odklon;
  int16_t naklon;

};

struct RECEIVE_DATA_STRUCTURE{
//Deklariramo spremenljivke za kanale, ki jih beremo iz arduino Mega

  int16_t kanal1;
  int16_t kanal11;
  int16_t kanal2;
  int16_t kanal3;
  int16_t kanal4;
  int16_t kanal5;
  int16_t kanal6;
```

Program se nadaljuje na naslednji strani.

```
int razlika3 = 0;

int korekcijaNagibLevo = 0;
int korekcijaNagibDesno = 0;
int korekcijaNaklon = 0;
int korekcijaOdklon = 0;

int gumbIzbira = 45;
int gumbGor = 43;
int gumbDol = 41;

int ledNagibLevo = 47;
int ledNaklon = 51;
int ledOdklon = 53;
int ledNagibDesno = 49;

int stopnja = 0;
int tipka = 1;

int prejsnje = 0;
int vzorec = 0;
int stanje = 0;
int pritisk = 1;

unsigned long cas = 0;
unsigned long cass = 0;
float simon = 0;
float lolek = 0;
float prejsnjiOdklon = 0;
float razlikaOdklon = 0;
int zapisNagibLevo = 0;
int zapisNagibDesno = 0;
int zapisOdklon = 0;
int zapisNaklon = 0;

float zeljenNaklon = 0;
float razlikaNaklon = 0;

float zeljenNagib = 0;
float razlikaNagib = 0;
```

```
  Serial.print(" ");
  Serial.print(kanal22);
  Serial.print(" ");
  Serial.print(kanal32);
  Serial.print(" ");
  Serial.print(kanal42);
  Serial.print(" ");
  Serial.print(kanal52);
  Serial.print(" ");
  Serial.print(kanal62);
  Serial.print("   /   ");
  Serial.print(odklon3);
  Serial.print(" ");
  Serial.print(naklon3);
  Serial.print(" ");
  Serial.print(nagib3);
  Serial.print("   /   ");
  Serial.print(zapisNagibLevo);
  Serial.print(" ");
  Serial.print(zapisNagibDesno);
  Serial.print(" ");
  Serial.print(zapisNaklon);
Serial.print(" ");
  Serial.println(zapisOdklon);

*/

  //Podprogrami
  IzbiraStopnjeKorekcije();
  Korekcija_Zakrilc();

  Branje_Podatkov();

if(kanal52 < 90)
{
  RocniNacin();
}
else
{
  AvtomatskiNacin();
}

  Krmiljenje_Servo_Motorjev();
}

void IzbiraStopnjeKorekcije()
{
  tipka = digitalRead(45);

  if(prejsnje ==! tipka && vzorec == 0)
  {
    cas = millis();
    stanje = tipka;
    vzorec = 1;
  }
  if(((millis() - cas) > 75) && vzorec == 1)
  {
    vzorec = 0;

    if((stanje == tipka))
    {
    pritisk = pritisk + 1;
    }

    if((stanje == tipka) && (pritisk == 1 || pritisk == 3))
    {

    if(pritisk == 3)
    {
      stanje = 0;
    }

    stopnja = stopnja + 1;

    if(stopnja == 5)
    {
      stopnja = 0;
    }

    tone(8, 1800, 50);
```

Program se nadaljuje na naslednji strani.

Program se nadaljuje na naslednji strani.

```
}
void Branje_Podatkov() {

    //Preverimo, če so podatki prispeli
    if(ET.receiveData()){

      //Spremenljivke za posamezne kote enačimo z prebranimi podatki
      //Pomnožimo jih z 0.01, da dobimo decimalna števila

        Kanal12 = Daljinec_podatki.kanal1;
        Kanal1111 = Daljinec_podatki.kanal111;
        Kanal22 = Daljinec_podatki.kanal2;
        Kanal32 = Daljinec_podatki.kanal3;
        Kanal42 = Daljinec_podatki.kanal4;
        Kanal52 = Daljinec_podatki.kanal5;
        Kanal62 = Daljinec_podatki.kanal6;
        /*
        Serial.print(kanal12);
        Serial.print(" ");
        Serial.print(kanal22);
        Serial.print(" ");
        Serial.print(kanal32);
        Serial.print(" ");
        Serial.print(kanal42);
        Serial.print(" ");
        Serial.print(kanal52);
        Serial.print(" ");
        Serial.println(kanal62);
        */
    }

    if(ET2.receiveData()){

      odklon2 = 0;  //Gyro_podatki.odklon*0.01;
      naklon2 =  0;  //Gyro_podatki.naklon*0.01;
      nagib2 =   0; //Gyro_podatki.nagib*0.01;

      odklon3 = Gyro_podatki.odklon*0.01;
      naklon3 = Gyro_podatki.naklon*0.01;
      nagib3 = Gyro_podatki.nagib*0.01;

    }
}

// ================================================== // R O Č N I   N A Č I N //

void RocniNacin() {

//Pretvorimo decimalne vrednosti podatkov Gyra v cela števila

  dNagibLevo = (int)nagib2;
  dNagibDesno = (int)nagib2;
  dNaklon = (int)naklon2;
  dOdklon = (int)odklon2;

  //željen kot
  zNagibLevo = kanal12;
  zNagibDesno = kanal1111;
  zNaklon = kanal22;
  zOdklon = kanal42;


//Mapiramo kote za servote

  dNagibLevo = map(dNagibLevo, -90, 90, 0, 180);

  dNagibDesno = map(dNagibDesno, -90, 90, 0, 180);
```

Program se nadaljuje na naslednji strani.

```
//Zapis Servo motorje

        zapisNagibLevo = 94 + razlika1 + korekcijaNagibLevo;

        if(zapisNagibLevo < 68)
        {
          zapisNagibLevo = 68;
        }

        if(zapisNagibLevo > 115)
        {
          zapisNagibLevo = 115;
        }

        zapisNagibDesno = 112 + razlika11 + korekcijaNagibDesno +25; // +razlika1

        if(zapisNagibDesno < 89)
        {
          zapisNagibDesno = 89;
        }

        if(zapisNagibDesno > 140)
        {
          zapisNagibDesno = 140;
        }

        zapisNaklon = 65 + razlika2 + korekcijaNaklon; // +razlika2

        if(zNaklon < 50)
        {
          zapisNaklon = 50;
        }

        if(zapisNaklon > 113)
        {
          zapisNaklon = 113;
        }

dNaklon = map(dNaklon, -90, 90, 0, 180);

dOdklon = map(dOdklon, -90, 90, 0, 180);

//Izračunamo razliko med dejanskim in željenim kotom

    if (dNagibLevo != zNagibLevo)
    {
    razlika1 = dNagibLevo - zNagibLevo;
    /*
      Serial.print(zNagibLevo);
      Serial.print(" ");
      Serial.print(razlika1);
      Serial.print(" ");
      Serial.println(zapisNagibLevo);
      */
    }
//ROČNI NAČIN

    if (dNagibDesno != zNagibDesno)
    {
    razlika11 = dNagibDesno - zNagibDesno;
    }

    if (dNaklon != zNaklon)
    {
    razlika2 = dNaklon - zNaklon;
    }

    if (dOdklon != zOdklon)
    {
    razlika3 = dOdklon - zOdklon;
    }
```

Program se nadaljuje na naslednji strani.

```
zapisNaklon = map(razlikaNaklon, -10, 10, 50, 113);
zapisNagibLevo = map(razlikaNagib, -10, 10, 115, 68);
zapisNagibDesno = map(razlikaNagib, -10, 10, 140, 89);

Serial.print(" ");
Serial.println(zapisNaklon);

}

void Krmiljenje_Servo_Motorjev(){

  if(zapisNagibLevo < 68)
  {
   zapisNagibLevo = 68;
  }

  if(zapisNagibLevo > 115)
  {
   zapisNagibLevo = 115;
  }


  if(zapisNagibDesno < 89)
  {
   zapisNagibDesno = 89;
  }

  if(zapisNagibDesno > 140)
  {
   zapisNagibDesno = 140;
  }

}

// ================================================== // A V T O M A T S K I   N A Č I N //

void AvtomatskiNacin(){

zeljenNaklon = 2;
zeljenNagib = 10;

razlikaNaklon = 0 - zeljenNaklon - naklon3;
razlikaNagib = zeljenNagib - nagib3;

Serial.print(razlikaNaklon);

zapisOdklon = 77 + razlika3 + korekcijaOdklon; //razlika3

  if(zapisOdklon < 62)
  {
   zapisOdklon = 62;
  }

  if(zapisOdklon > 110)
  {
   zapisOdklon = 110;
  }

/*
   NagibLevo.write(91); // nula = 91 , min = 115, max = 68
   NagibDesno.write(113);// nula = 112, max = 140, min = 89;
   Naklon.write(77);// nula = 69, max = 113, min = 50
   Odklon.write(82);// nula = 78, max = 110, min = 62
   PogonskiMotor.write(hitrost);
   */

}

if(zapisNaklon > 113)
{
 zapisNaklon = 113;
}

if(zapisOdklon < 62)
{
 zapisOdklon = 62;
}

if(zapisOdklon > 110)
{
 zapisOdklon = 110;
}

   NagibLevo.write(zapisNagibLevo);
   NagibDesno.write(zapisNagibDesno);
   Naklon.write(zapisNaklon);
   Odklon.write(zapisOdklon);
   //PogonskiMotor.write(hitrost);
```

Program se nadaljuje na naslednji strani.

Program za Arduino Nano (angl. slave), branje podatkov žiroskopa:

Program se nadaljuje na naslednji strani.

9 VIRI IN LITERATURA

1. Brezar, J., 1995. Jadralno letalstvo. Letalska zveza Slovenije, Ljubljana.
2. http://dk.mors.si/Dokument.php?id=878, (13. 1. 2019).
3. https://en.wikipedia.org/wiki/Autopilot, (3.12.2018).
4. https://en.wikipedia.org/wiki/Holding_(aeronautics), (3.12.2018).
5. https://www.sloveniacontrol.si, (15. 1. 2019).
6. https://en.wikipedia.org/wiki/Thrust-to-weight_ratio, (10. 1. 2019).
7. https://www.grc.nasa.gov/www/k-12/WindTunnel/Activities/lift_formula.html, (8.11.2018).
8. http://airfoiltools.com/airfoil/details?airfoil=goe285-il#polars, (3.1.2019).
9. https://zmaga.com/content.php?id=4394, (13.2.2019).
10. http://playground.arduino.cc/code/interrupts, (5.2.2019).
11. https://ryanboland.com/blog/reading-rc-receiver-values/, (15. 1. 2019).
12. https://ucilnica.fri.uni-lj.si/pluginfile.php/21024/mod_resource/content/1/Regulacije.pdf, (5.12.2018).
13. http://airfoiltools.com/airfoil/details?airfoil=goe285-il, (13. 2. 2019).