

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Samo Kralj

**Letalo brez posadke z
ArduPilot Mega avtopilotom**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.



Št. naloge: 01819/2012

Datum: 15.03.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SAMO KRALJ**

Naslov: **LETALO BREZ POSADKE Z ARDUPILOT MEGA AVTOPILOTOM
UNMANNED AIRCRAFT WITH THE ARDUPILOT MEGA AUTOPILOT**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Letalo brez posadke je naprava, ki jo je mogoče upravljati na daljavo ali pa jo vodi v letalo vgrajen avtopilot. Takšna letala imajo v določenih primerih pomembne prednosti pred letali s človeško posadko in so zato zanimivo področje raziskovanja. V ta namen raziščite možnosti za izdelavo preprostega letala te vrste. Kot osnovo uporabite odprtokodni projekt ArduPilot Mega in njegovo strojno platformo. Izberite ustrezne senzorske in aktuatorske komponente ter izdelajte vezje avtopilota vključno z radijsko komunikacijo. Izdelajte tudi vso potrebno programsko opremo in vgradite avtopilot v model letala. Pravilnost delovanja preverite na primeru vnaprej podane misije letenja in podajte možne načine za izboljšave.

Mentor:


prof. dr. Dušan Kodek

Dekan:


prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Samo Kralj,

z vpisno številko 63050156,

sem avtor diplomskega dela z naslovom:

Letalo brez posadke z ArduPilot Mega avtopilotom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. 04. 2012

Podpis avtorja:

Zahvala

Zahvaljujem se družini, prijateljem in dekletu za pomoč in podporo med študijem.
Hvala tudi vsem ostalim, ki ste mi stali ob strani.

Kazalo

Povzetek	1
Abstract	2
1 Zračna plovila brez posadke	3
1.1 Zgodovina	4
1.2 Kategorizacija	5
1.3 Komponente letala brez posadke	5
2 Strojna oprema	7
2.1 ArduPilot Mega	8
2.1.1 APM glavna plošča	9
2.1.2 APM IMU plošča	9
2.1.3 ArduPilot Mega 2	10
2.2 Vhodno-izhodne naprave	11
2.2.1 3-osni žiroskop in pospeškometer	11
2.2.2 Barometrični višinomer	12
2.2.3 Pitojeva cev	13
2.2.4 GPS modul	13
2.2.5 Brezžična komunikacija	16
2.2.6 Servomotorji in pogonski sklop	18
3 Programska oprema	19
3.1 Strojno-programska koda	19
3.1.1 Kontrolna zanka avtopilota	20
3.1.2 Zlivanje senzorjev	23
3.2 Režimi letenja	23
3.3 Oddaljena nadzorna postaja	25
3.4 MAVLink protokol	27

3.4.1	MAVLink sporočila	27
3.4.2	Načrtovanje misij	30
4	ArduPilot Mega v praksi	32
4.1	Vgradnja APM v letalo	33
4.2	XBee konfiguracija	34
4.3	Nalaganje programske opreme avtopilota	34
4.4	Nastavitve in letenje	36
4.5	Preizkusna misija	37
5	Zaključek	41
	Seznam tabel	42
	Seznam slik	43
	Literatura	45

Seznam uporabljenih kratic in simbolov

ADC	Analog-to-Digital converter
APM	ArduPilot Mega
DCM	Direction Cosine Matrix
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPS	Global Positioning System
IMU	Inertial Measurement Unit
NMEA	National Marine Electronics Association
PID	Proportional Integral Derivative

Povzetek

Zračna plovila brez posadke so letalne naprave, ki jih je mogoče upravljati na daljavo ali pa jih nadzoruje avtopilot. Diplomsko delo na kratko predstavi zgodovino in koncept letal brez posadke, nekoliko podrobneje pa množico različnih komponent, ki so prisotne v preprostem modelu letala z vgrajenim avtopilotom ArduPilot Mega (APM) [2] in jih, v kompleksnejši izvedbi, srečamo tudi v večjih avtopilotih.

Prvo poglavje na kratko predstavi namen ter zgodovino takšnih letal. V drugem je predstavljena strojna oprema APM avtopilota ter samega letala - nadzorni del, senzorji in aktuatorji ter brezžična komunikacija. Tretje poglavje se osredotoči na programski del. Programsko opremo, ki skrbi za krmiljenje letala, podporno programsko opremo za oddaljen nadzor in komunikacijski protokol MAVLink, razvit posebej za upravljanje in nadzor manjših avtonomnih zračnih plovil. Diplomsko delo se v zadnjem poglavju zaključuje z vgradnjo avtopilota v model letala, konfiguracijo in preizkusom delovanja avtopilota z enostavno avtonomno misijo.

Ključne besede: avtopilot, ArduPilot Mega, plovilo brez posadke, MAVLink

Abstract

Unmanned aerial vehicles are flying devices that can be controlled remotely or autonomously by an on-board autopilot. This thesis gives a brief insight into the history of unmanned aerial vehicles and a more thorough overview of components found in ArduPilot Mega (APM) [2] that can also be found in complex commercial autopilots.

The first chapter gives an insight into the short history of such vehicles. The second chapter focuses on hardware components of APM autopilot - control logic, sensors, actuators and wireless telemetry. Software side of APM is covered in the third chapter, with details of autopilot capabilities, firmware, ground control software and MAVLink communication protocol designed specifically for small aerial vehicles. The fourth chapter explains installation of autopilot into model airplane, different configuration options and analyzes a simple autonomous mission.

Key words: autopilot, ArduPilot Mega, unmanned aerial vehicle, MAVLink

Poglavje 1

Zračna plovila brez posadke

Zračno plovilo brez posadke (angleško *Unmanned Aerial Vehicle*) je naprava, ki jo je mogoče upravljati na daljavo ali pa jo nadzoruje avtopilot, pri čemer človeški operater ni fizično prisoten v samem plovilu, saj ga nadzoruje - pilotira ali zgolj spremlja delovanje - na daljavo. Takšna plovila imajo v določenih situacijah pred plovili s človeško posadko več pomembnih prednosti:

- Večja je vzdržljivost zaradi manjše utrujenosti operaterjev v nadzornem centru.
- Nižji so stroški ob morebitni izgubi plovila, tudi na račun zmanjšanja žrtev med operaterji letal in izvidniško vojsko.
- Možne so operacije v kemično, biološko, radioaktivno in eksplozivno nevarnih območjih.

Poleg omenjenih imajo manjša letala brez posadke, med katere spada tudi letalo predstavljeno v tem diplomskem delu, še eno pomembno prednost - ceno. Celotni stroški opreme - avtopilot, letalo, programska in strojna oprema (brez cene osebnega ali prenosnega računalnika) se gibljejo med 500 in 1000 evri za letalo, zgrajeno na APM platformi, pa do nekaj tisoč evrov za podobne komercialne različice mikro avtonomnih zračnih plovil.



Slika 1.1: Letalo brez posadke MQ9 Reaper

1.1 Zgodovina

Začetki prvih mehaničnih avtonomnih plovil segajo v čas antične Grčije. Arhit (428-347 pr. n. št.) naj bi že 425 let pred našim štetjem načrtoval in izdelal prvo letalno napravo z lastnim pogonom [3]. Golob, kot se je naprava imenovala, je letel s pomočjo curka pare in vode. Kasneje so bila mehanska letalna plovila z in brez posadke priljubljena v renesančnem obdobju Leonarda da Vincija, ideja o zračnih plovilih brez posadke pa se je v nekoliko modernejši obliki pojavila med I. svetovno vojno. Že leta 1915 je Nikola Tesla vizionarsko opisal floto takšnih letal [4], realnost pa je bila precej drugačna. Letala brez posadke, ki so jih v vojni uporabili leta 1917, so bila nezanesljiva in nenačnana, njihovo pravo vrednost na bojišču in vlogo v prihodnosti pa so, poleg Tesle, priznavali le redki. Resneje so letala brez posadke poletela leta 1964 na vojaških misijah v Vietnamu [5], popoln razmah pa so doživela po operaciji puščavska nevihta (*Desert Storm*) leta 1991.

Še danes vojska ostaja z naskokom največji vlagatelj v raziskave in razvoj na področju zračnih plovil brez posadke - predvsem v Združenih državah Amerike, ki bodo leta 2015 v to področje predvidoma vložile 16 milijard dolarjev (12 milijard evrov). Sledi jim Evropa s predvideno 6-krat manjšim (2 milijardi evrov) proračunom, namenjenim tovrstnim plovilom [4].

1.2 Kategorizacija

Zračna plovila brez posadke lahko kategoriziramo glede na več kriterijev - po funkciji, dometu ali specifično glede na potrebe večjih naročnikov oziroma investitorjev (vojske). Ameriška vojska takšna plovila po funkcionalnosti deli na:

- **tarče** - posnemanje sovražnih letal pri vojaških vajah
- **izvidnica** - podatki o bojišču in spremljanje dogajanja
- **bojna plovila** - napad v zelo tveganih misijah
- **logistika** - transport ali drugačna logistična podpora
- **raziskave in razvoj**
- **civilna in komercialna zračna plovila brez posadke**

Brezpilotno letalo, predstavljeno v diplomskem delu, bi lahko uvrstili v zadnji dve kategoriji.

1.3 Komponente letala brez posadke

Letalo brez posadke sestavlja množica komponent, obseg katerih je odvisen predvsem od namena uporabe. Ker se to diplomsko delo osredotoča na letalo z avtopilotom ArduPilot Mega, so v njem predstavljene le komponente, potrebne za delovanje tega avtopilota. Podobne komponente, v kompleksnejši, razširjeni in redundantni obliki, pa srečamo tudi v večjih komercialnih in vojaških brez-pilotnih letalih.

Osnova plovila je samo letalo, ki ga v tem diplomskem delu ne bomo posebej predstavljali. V letalo so vgrajeni različni senzorji, ki služijo izračunavanju orientacije, položaja in hitrosti plovila, krmilniki (avtopilot) in aktuatorji, ki poganjajo krmilne površine in pogon letala. Pomembna, a ne nujno obvezna, je tudi brezžična komunikacija, preko katere lahko operater letala z oddaljene nadzorne postaje nadzira plovilo.

Poleg strojne opreme na letalu ali tleh sta pomembni tudi programska oprema avtopilota in podporne oddaljene nadzorne postaje. Prva skrbi za zbiranje informacij s senzorjev, njihovo interpretacijo in krmiljenje aktuatorjev, slednja

pa omogoča pregled nad dogajanjem na letalu, nadzor njegovega delovanja, nastavljanje avtopilota, ustvarjanje misij in podobno.

Poglavje 2

Strojna oprema

Osnova avtonomnega plovila je polje senzorjev in aktuatorjev, ki jih povezuje krmilna logika. Krmilna logika interpretira vhodne podatke iz senzorjev in na podlagi teh prek aktuatorjev upravlja plovilo - ga usmerja po vnaprej določeni poti, stabilizira let, kroži nad določeno točko ipd.

Nabor senzorjev je odvisen od želja in zahtev, ki jih mora izpolnjevati plovilo. Med pogosto uporabljenimi, ki so uporabljeni tudi pri letalu v diplomskem delu, pa so:

- žiroskopi;
- pospeškometri;
- barometrični višinomer;
- GPS modul;
- senzor zračne hitrosti.

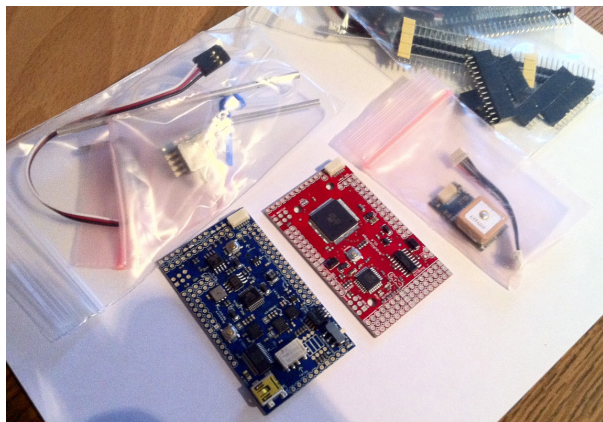
Kot aktuatorji so v tem projektu uporabljeni servomotorji za premikanje smernega in višinskega krmila ter elektromotor za pogon letala.

Komunikacija z letalom (preklop med različnimi režimi delovanja, prevzem vodenja letala ipd.) poteka s klasičnim oddajnikom za brezžično vodenje hobi modelov, ki deluje v frekvenčnem pasu okrog 2.4 GHz. Prenos telemetrije in dodatnih ukazov med letalom in nadzorno postajo na zemlji pa poteka preko brezžičnih XBee modulov.

2.1 ArduPilot Mega

ArduPilot Mega [2] je odprtokodni projekt majhnega, poceni in relativno preprostega avtopilota, ki poleg odprte programske kode vsebuje tudi odprto strojno platformo. Odprtost platforme omogoča, da si APM prilagodimo po svojih specifičnih potrebah - ne samo programski del, ampak tudi strojni del, saj so poleg programske opreme prosto dostopne tudi sheme električnega vezja avtopilota.

Avtopilot je sestavljen iz dveh večjih strojnih komponent - APM glavne plošče in APM IMU plošče, ki vsebujeta mikroprocesorje, senzorje in pomožno strojno logiko. Programski del pa obsega programsko opremo avtopilota in programsko opremo za oddaljeni nadzor (*Ground Control Station* - v nadaljevanju *nadzorna postaja*). APM platforma tako omogoča planiranje misij z vnaprej določenimi 3D točkami (GPS koordinate z višino), avtomatično vzletanje in pristajanje, stabilizacijo leta (*fly-by-wire*), kroženje okrog določene ali trenutne točke in podobno. Z vgrajeno opremo za brezžično komunikacijo pa med samim letom preko nadzorne postaje spremljamo parametre senzorjev in spreminjamo nastavitve misije.



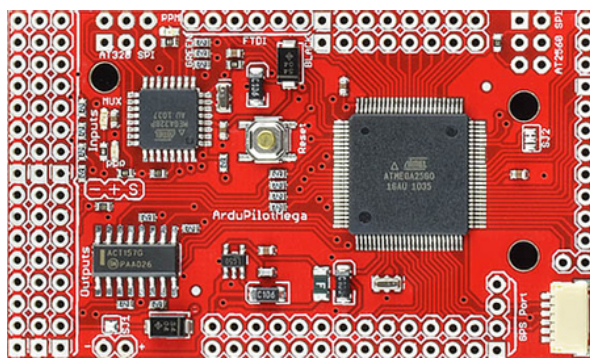
Slika 2.1: APM komplet pred spajkanjem

Strojni del APM lahko dobimo že v celoti izdelan in pripravljen za vgradnjo v model letala ali pa kot komplet, pri katerem s spajkanjem konektorjev obe plošči povežemo med seboj in z zunanjimi moduli (senzorjem zračne hitrosti, GPS modulom, brezžično telemetrijo). Komplet prav tako nima naložene programske opreme, tako da moramo poskrbeti tudi za to. Pri tem si lahko

pomagamo z nadzorno postajo APM Planner, ki na APM namesto nas naloži najnovejšo verzijo avtopilota. Lahko pa jo prevedemo in naložimo tudi sami. Izgradnja, programiranje, priprava na let in samo letenje so podrobneje opisani v poglavju 4.

2.1.1 APM glavna plošča

APM glavna plošča velikosti 40 x 69 mm je elektronsko vezje, ki skrbi za nadzor in krmiljenje vhodno-izhodnih naprav ter senzorjev.



Slika 2.2: ArduPilot Mega glavna plošča

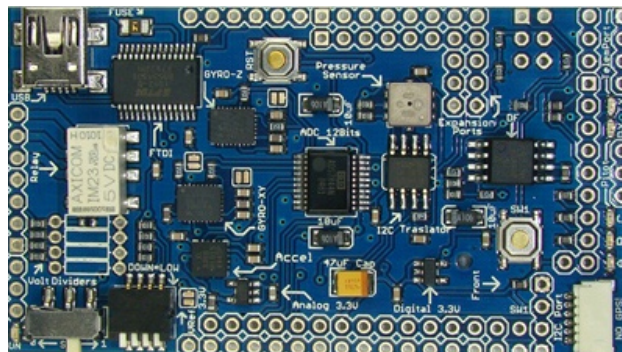
Jedro plošče je 16 MHz Atmelov ATmega1280, 8-bitni RISC mikroprocesor. Programska koda avtopilota se nahaja v vgrajenem 128 kB flash pomnilniku, na samem čipu pa je še 8 kB SRAM pomnilnika in 4 kB EEPROM pomnilnika. Na plošči je 16 analognih vhodov z ADC pretvorniki, 40 digitalnih vhodov/izhodov in štiri serijski vmesniki, med katerimi je tudi vmesnik za GPS modul.

Poleg glavnega mikroprocesorja je na plošči še pomožni ATmega328 mikroprocesor, ki služi kot PPM kodirnik (*Pulse-position modulation*) in sistem za obravnavanje napak, ki v primeru neodzivnosti avtopilota ali nedosegljivosti naprave za radijsko vodenje, nadzor prek multiplekserja avtomatično preklopi mimo avtopilota na napravo za radijsko vodenje plovila.

2.1.2 APM IMU plošča

Druga polovica avtopilota je APM IMU plošča, ki vsebuje senzorje, potrebne za izračun orientacije in gibanja letala na podlagi sprememb pospeškov.

Telesa zaradi inercije ohranjajo svoj položaj in rotacijo konstantno, dokler jih ne zmoti zunanja sila ali navor [6]. Merilnim enotam, ki na podlagi sprememb pospeškov določajo položaj in rotacijo telesa, pravimo *inercialne merilne enote* (*Inertial Measurement Unit - IMU*). Pri določanju rotacij se inercialne merilne enote opirajo na žiroskope, ki pomagajo pri fiksaciji in stabilizaciji osi, okrog katerih s pospeškomeri merimo kotne pospeške. Vsi ti senzorji se nahajajo na IMU plošči, poleg njih pa tudi barometrični senzor za določanje nadmorske višine, vodilo za priklop senzorja zračne hitrosti in vodilo za priklop magnetnega kompasa.



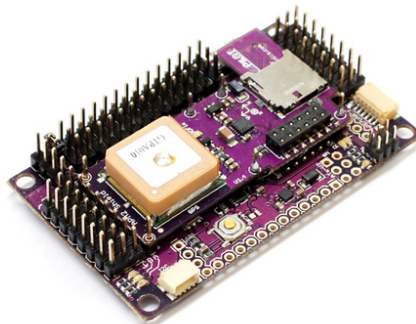
Slika 2.3: ArduPilot Mega IMU plošča

Poleg senzorjev plošča vsebuje rele, ki ga lahko uporabimo za proženje ob različnih dogodkih, 16 MB flash pomnilnika za beleženje senzorskih podatkov, priključek za XBee modul ter USB vmesnik za komunikacijo z APM - preko njega tudi poteka nalaganje programske opreme avtopilota.

2.1.3 ArduPilot Mega 2

V času pisanja diplomskega dela je tik pred izidom osvežena različica strojnega dela APM platforme z imenom ArduPilot Mega 2 (programska oprema se osvežuje ločeno in zelo aktivno). Kljub temu, da diplomsko delo temelji na starejši različici, velja omeniti spremembe, ki jih prinaša nova platforma.

Funcionalnosti novega avtopilota so skoraj identične njegovemu predhodniku, med opaznejšimi spremembami pa izstopa kompaktnejša izvedba z manjšim številom zunanjih komponent - večina senzorjev je vgrajena na glavno ploščo, na ločeni manjši plošči sta le GPS modul in kompas. Na ta način sta zagotovljeni večja zanesljivost ter manjša velikost in teža samega avtopilota.



Slika 2.4: ArduPilot Mega 2 (*Vir: diydrones.com*)

Nov model vsebuje nekaj novejših senzorjev, med katerimi velja izpostaviti InvenSenseov MPU-6000. Čip v QFN ohišju vsebuje 3-osni žiroskop, 3-osni akcelerometer in DMP procesor (*Digital Motion Processor*), ki opravlja vlogo zlivanja senzorjev ter tako nadomesti programsko zlivanje senzorjev, opisano v poglavju 3.1.2. Na ta način se razbremeni glavni procesor, ki je po novem ATmega256.

2.2 Vhodno-izhodne naprave

Nabor vhodno-izhodnih (V/I) naprav je odvisen od želja in potreb upravljalca letala. Zmogljivejša avtonomna letala poleg kompleksnejših, zanesljivejših in pogosto redundantnih različic V/I naprav, predstavljenih v tem poglavju, vsebujejo še množico drugih - radarje, različne kamere, morebitno bojno opremo in podobno. Za potrebe enostavnega, a kljub temu zmogljivega raziskovalnega letala, pa povsem zadostujejo V/I naprave, ki so prisotne na APM avtopilotu in so na kratko predstavljene v tem poglavju.

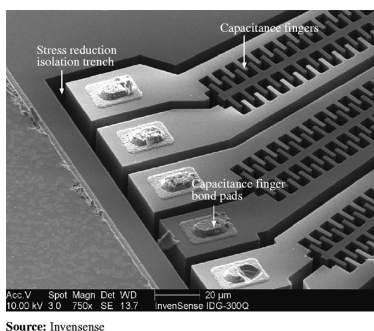
2.2.1 3-osni žiroskop in pospeškometer

Kot že omenjeno, žiroskop v povezavi s pospeškometerom omogoča določanje položaja (orientacije) telesa v prostoru.

Na IMU plošči sta dva žiroskopa (XY in Z), izdelana v MEMS tehnologiji (*Microelectromechanical systems*) [8]. Ločena čipa sta bila v prvi različici av-

topilota verjetno uporabljena zaradi lažje dostopnosti teh čipov, sedaj pa obstajajo cenovno dostopne izvedbe na enem čipu [7], ki so uporabljene na APM 2. Poleg žiroskopa je uporabljen 3-osni pospeškometer ADXL335.

Izhodi obeh senzorjev so analogni in povezan na 8-kanalni, 12-bitni ADC pretvornik (*Analog-to-digital Converter*) ADS7844, s katerim preko serijskega vmesnika komunicira mikroprocesor.



Slika 2.5: Del MEMS žiroskopa pod elektronskim mikroskopom (*Vir [9]*)

2.2.2 Barometrični višinomernik

Za merjenje nadmorske višine je uporabljen absolutni Boschov barometrični višinomernik BMP085, ki omogoča veliko večjo natančnost merjenja višine, kot bi bila mogoča samo z uporabo podatkov GPS modula. Mikroprocesor preko I²C povezave s senzorja prebira vrednost zračnega tlaka in trenutno temperaturo, nato pa na podlagi barometrične enačbe (2.6) izračuna trenutno absolutno višino plovila.

$$\rho = \rho_b \cdot \exp \left[\frac{-g_0 \cdot M \cdot (h - h_b)}{R^* \cdot T_b} \right] \quad (2.1)$$

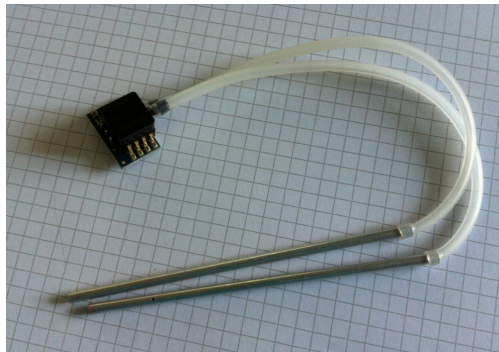
$$h = - \frac{R^* T_0 \ln \frac{\rho}{\rho_0}}{M g_0} \quad (2.2)$$

ρ	Zračni tlak
ρ_b	Statični tlak
g_0	Gravitacijski pospešek
M	Molska masa zraka ($0.0289644 \frac{kg}{mol}$)
R^*	Univerzalna plinska konstanta za zrak ($8.31432 \frac{Nm}{molK}$)
T_b	Standardna temperatura
h	Nadmorska višina
h_b	Višina na dnu plasti b

Slika 2.6: Absolutna nadmorska višina, izražena iz barometrične enačbe

2.2.3 Pitojeva cev

Pitojeva cev, ki se uporablja za določanje zračne hitrosti letala, deluje na osnovi Bernoullijeve enačbe. Naprava je sestavljena iz dveh cevi, od katerih je ena odprta v smeri leta (dinamična), druga (statična) pa je odprta pravokotno na smer zračnega toka.



Slika 2.7: Senzor zračne hitrosti na osnovi Pitojeve cevi

Iz dobljenih tlakov v obeh ceveh (totalnega p_t in statičnega p_s) in Bernoullijeve enačbe lahko nato izračunamo hitrost toka zraka oziroma gibanja plovila skozi zrak v našem primeru.

$$p_t = p_s + \left(\frac{\rho V^2}{2} \right) \quad (2.3)$$

$$V = \sqrt{\frac{2(p_t - p_s)}{\rho}} \quad (2.4)$$

V Hitrost tekočine (zraka)
 p_t, p_s Totalni in statični tlak
 ρ Gostota tekočine

Slika 2.8: Bernoullijeva enačba in hitrost zračnega toka

Obe analogni vrednosti senzorja zračne hitrosti (statični in dinamični tlak) sta povezani na dva izmed vhodov ADC pretvornika.

2.2.4 GPS modul

MediaTekov MT3329 GPS modul je potreben za natančno določanje lokacije našega plovila, kar je pomembno pri množici različnih scenarijev delovanja av-

topilota - pri sledenju točkam, vračanju k vzletni točki, kroženju okrog točke pa tudi pri delovanju same IMU enote pri računanju orientacije letala. 66 kanalni GPS sprejemnik z vgrajeno anteno izračuna položaj vsakih 100 ms, z natančnostjo 3 m, kar je za potrebe relativno počasnega modela letala povsem zadovoljivo. Komunikacija z GPS modulom poteka preko serijskega vmesnika, modul pa podatke pošilja po NMEA 1083 (*National Marine Electronics Association*) protokolu.



Slika 2.9: MediaTek 3329 GPS modul z anteno

NMEA 1083

NMEA 0183 [11] je enostaven ASCII serijski protokol, ki podatke pošilja v obliki stavkov, ki so sestavljeni po naslednjem formatu:

- Prvi znak vsakega stavka je \$.
- Naslednjih 5 znakov identificira sporočevalca in tip sporočila (2 + 3 znaki).
- Sledijo podatkovna polja, ki so ločena z vejico.
- Prvi znak, ki sledi koncu podatkov, je *.
- Sledita mu dva znaka kontrolne vsote (*ekskluzivni ali* vseh znakov med \$ in *).
- Stavek se konča z znakoma za novo vrstico (*CRLF*).

NMEA standard predpisuje 19 različnih stavkov, APM pa obdeluje le stavke GPGGA in GPVTG (prvi dve črki povesta, da je sporočevalec GPS naprava, naslednje tri pa podajo ime stavka):

- GPGGA (*Global Positioning System Fix Data*) - Vsebuje podatke, povezane s položajem in časom, ko je GPS nazadnje pridobil te podatke.
- GPVTG (*Track Made Good and Ground Speed*) - Iz tega sporočila pridobimo podatke o smeri in hitrosti premikanja sprejemnika.

Primer GPGGA stavka

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Vrednost	Opis
123519	Podatki zabeleženi ob 12:35:19 po UTC času
4807.038,N	48° 07.038' S geografske širine
01131.000,E	11° 31.000' V geografske dolžine
1	Kvaliteta podatka <ul style="list-style-type: none"> • 0 - neveljaven • 1 - GPS podatek • 2 - DGPS podatek • ...
08	Število satelitov
0.9	Relativna horizontalna natančnost podatka
545.4,M	Nadmorska višina v metrih
46.9,M	Višina geoida (v metrih) nad WSG84 elipsoidom
prazno polje	Čas (v sekundah) od zadnje DGPS posodobitve
prazno polje	ID DGPS postaje
*47	Kontrolna vsota

Tabela 2.1: Primer NMEA GPGGA stavka

2.2.5 Brezžična komunikacija

Komunikacija z letalom in avtopilotom poteka na dva načina - s klasično brezžično napravo za vodenje letalskih modelov, ki deluje v 2.4 GHz ISM frekvenčnem pasu, in brezžično XBee komunikacijo.

Sprejemnik avtopilotu posreduje ukaze, ki jih letalu pošiljamo z napravo za vodenje letala, avtopilot pa nam lahko prepusti popoln nadzor nad letenjem (ročni način), nam pri letenju pomaga s stabilizacijo ali pa vodenje v celoti prevzame. Prav tako lahko z napravo za vodenje letala prožimo določene dogodke, na primer preklapljanje med različnimi režimi delovanja avtopilota.

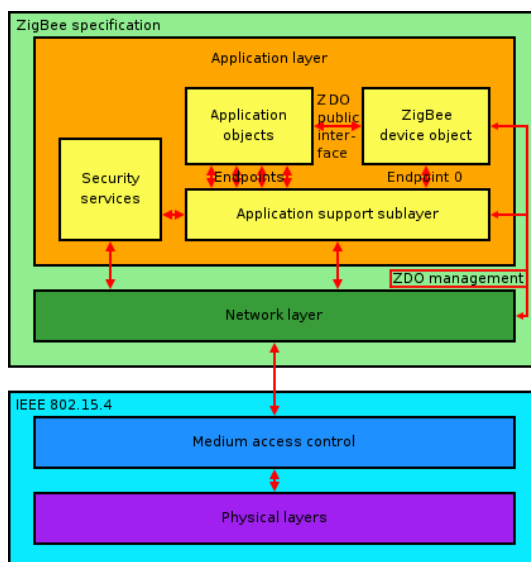


Slika 2.10: Brezžična XBee telemetrija (*Vir [2]*)

Dodatni način komunikacije je preko XBee modulov. Vgradnja teh modulov za samo delovanje avtopilota ni ključna, saj je sposobno povsem avtonomnega letenja tudi brez nje, vendar pa s to razširitvijo dobimo boljši pregled nad dogajanjem na letalu. V živo lahko spremljamo parametre leta, vrednosti senzorjev in aktuatorjev, prav tako pa lahko z nadzorne postaje letalu pošiljamo ukaze, s čimer lahko med samim letom na primer prožimo določene akcije, spreminjamo pot leta, preklapljammo med režimi delovanja in podobno. Nadzorna postaja in MAVLink protokol sporočil, ki se pošiljajo med nadzorno postajo ter avtopilotom, sta podrobneje predstavljena v poglavjih 3.3. in 3.4.

XBee, ZigBee in IEEE 802.15.4

APM za brezžično komunikacijo uporablja XBee module podjetja Digi International, ki temeljijo na ZigBee standardu [15]. V letalo sem vgradil module XBee PRO S2B [14] z oddajno močjo 63 mW, ki omogoča komunikacijo na razdalji do 3 km vidne linije. Hitrost komunikacije je, odvisno od kvalitete signala in nastavitvev omrežja, do 35 kb/s.



Slika 2.11: Shema ZigBee in IEEE 802.15.4 standardov (Vir [15])

ZigBee protokol opisuje delovanje omrežne in aplikacijske plasti (po OSI modelu), fizična in MAC (*Media Access Control*) plast sta narejeni po IEEE 802.15.4 paketnem standardu. ZigBee standard predpisuje dva različna tipa vozlišč:

- RFD (*reduced-function device*) - preprosta vozlišča z izredno majhno porabo, ki so zaradi svoje omejene funkcionalnosti sposobna komunikacije le z FFD vozlišči.
- FFD (*full-function device*) - poleg RFD funkcij opravlja še naloge koordiniranja mreže.

Topologije so lahko preproste *peer-to-peer* (*enak z enakim*), zvezdaste in mrežne (*mesh*), vendar pa vsaka mreža potrebuje vsaj eno FFD vozlišče (koordinatorja).

Ostale lastnosti standarda so še:

- izogibanje trkom s CSMA/CA (*Carrier sense multiple access with collision avoidance*) algoritmom;
- potrjevanje paketov;
- podpora varnim povezavam (128 bitna AES enkripcija);

- majhna poraba zaradi hitrega aktiviranja modula (modul lahko večino časa *spi*);
- do 65000 vozlišč v enem omrežju;
- odkrivanje prostih kanalov in oblikovanje mreže znotraj njih.

2.2.6 Servomotorji in pogonski sklop

Letalo krmilimo z dvema servomotorjema - enim za premikanje višinskega krmila, s katerim spreminjamo naklon letala (*pitch*), in enega za premikanje smernega krmila, s katerim spreminjamo smer leta (*yaw*). Servomotor za radijsko vodene modele je sestavljen iz elektromotorja, povezanega na potenciometer. Radijski sprejemnik, oziroma v našem primeru APM, servomotorju pošlje PPM signal, ki ga elektronika v servomotorju dekodira in na podlagi širine pulza spremeni položaj servomotorja. Potenciometer pri tem služi kot povratna informacija elektroniki o tem, v katerem položaju se servomotor nahaja [16].



Slika 2.12: Servomotor za radijsko vodene modele

Za pogone letala služi brezkrtačni elektromotor z BEC krmilnikom (*Battery Eliminator Circuit* - akumulator, ki napaja pogonski motor, hkrati dovaja elektriko tudi krmilni logiki in servomotorjem). Nadzor krmilnika je enak kot pri servomotorjih - s PPM signalom.

Poglavje 3

Programska oprema

Tudi programska oprema ArduPilot platforme je pisana pod odprtokodno licenco, zato sta spreminjanje in vpogled v kodo omogočena vsakomur, ki ima dovolj znanja za poglobljanje v kodo avtopilota in spremljajoče programske opreme. V tem poglavju sta podrobneje predstavljena programska oprema avtopilota in njegove funkcije ter programska oprema oddaljene nadzorne postaje.

3.1 Strojno-programaska koda

APM temelji na Arduino platformi, ki je bila zasnovana na *Wiring* in *Processing* platformah. Vsem trem je skupen en cilj - poenostavitev procesa uporabe elektronike v multidisciplinarnih projektih in jih tako približati širši množici. Programska oprema Arduino je sestavljena iz C/C++ prevajalnika, zagonskega nalagalnika, ki teče na Arduino plošči (v našem primeru na APM), in množice različnih knjižnic, namenjenih komunikaciji (serijski, XBee, Ethernet), prikazovanju (LED diode, segmentni prikazovalniki), senzorjem (ultrazvočni), shranjevanju podatkov (EEPROM, SD Card), krmiljenju aktuatorjev (servomotorji, koračni motorji ...) in podobno.

Vsak Arduino program vsebuje dve glavni funkciji - *void setup()*, ki se zažene, ko zagonski nalagalnik opravi svoje delo (po priklopu ploščice na napajanje ali resetiranju ploščice), in *void loop()*, ki predstavlja telo neskončne zanke, ki se izvaja, dokler je prisotno napajanje ploščice. Enostaven Arduino program, v

katerem se servomotor, priklopljen na digitalni izhod 9, premika levo in desno (od 0-180 stopinj in nazaj), je prikazan v spodnjem primeru.

```
#include <Servo.h>

Servo myservo;
int pos = 0;

void setup() {
  // servo signal na izhoda 9
  myservo.attach(9);
}

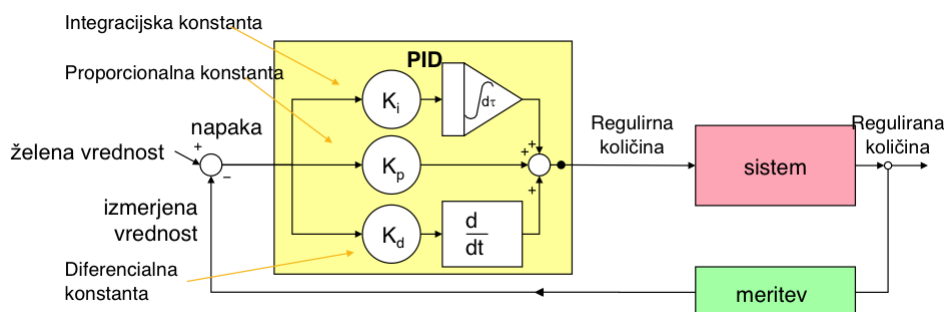
void loop() {
  // 0 do 180 stopinj in nazaj
  for(pos = 0; pos < 180; pos++) {
    myservo.write(pos);
  }
  for(pos = 180; pos >= 1; pos--) {
    myservo.write(pos);
  }
}
```

Listing 3.1: Primer enostavne kode za nadzor servomotorja

Strojno-programaska koda ArduPilot Mega standardnemu naboru Arduino knjižnic doda še knjižnice za komunikacijo z ostalimi V/I napravami, prisotnimi na APM, ter kodo samega avtopilota, ki podatke s senzorjev ustrezno interpretira in se nanje odzove.

3.1.1 Kontrolna zanka avtopilota

Kot že omenjeno, se glavno procesiranje na Arduini ploščah dogaja v neskončni zanki. Zato je tako zgrajen tudi avtopilot - v neskončni zanki prebira vrednosti senzorjev, jih ustrezno združi in glede na njihove vrednosti poskrbi za pravilno postavitev servomotorjev in moči pogonskega motorja. Pri tem so izhodne vrednosti servomotorjev in pogonskega motorja regulirane z zaporedno (kaskadno) vezanima PID regulatorjema (*Proportional Integral Derivative*). Prvi regulator izračunava želen nagib in naklon letala glede na informacijo o trenutni smeri in želeni smeri, drugi pa določa potrebne spremembe vrednosti servo motorjev za doseg želenega nagiba oziroma naklona.



Slika 3.1: Shema PID regulatorja (Vir slike: Lotrič, U.[17])

PID regulator sestavljajo tri konstante - proporcionalna, diferencialna in integracijska. Vsaka izmed teh določa, kolikšen je proporcionalen, diferencialen ali integracijski prispevek vhodnih parametrov in napak pri določanju izhodnih vrednosti - na primer odklona smernega ali višinskega krmila in moči motorja. Delovanje samega regulatorja je močno odvisno od *pravilnih* vrednosti PID konstant, napačno nastavljene vrednosti pa lahko med drugim povzročijo prepočasno odzivanje regulatorja ali pa preveliko nihanje od zelene vrednosti. Vrednosti PID konstant lahko spreminjamo preko nadzorne postaje, med testiranjem pa se je izkazalo, da so prednastavljene vrednosti v mojem primeru delovale zadovoljivo.

Delovanje takšnega kaskadnega PID krmilnika in samega avtopilota lahko opišemo s preprosto psevdokodo.

```
void loop() {
    // preberi vrednosti senzorjev
    read_sensors();

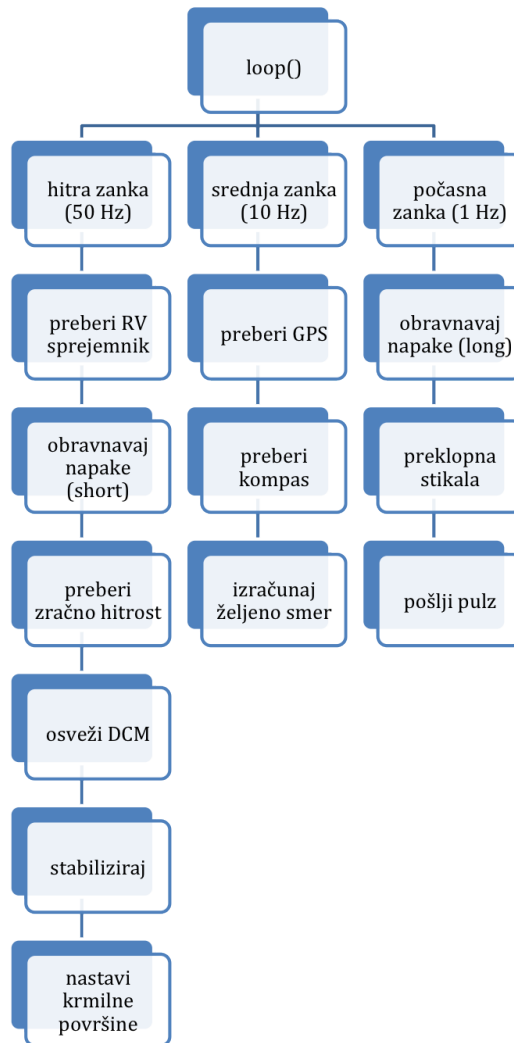
    // izracunaj zeljen nagib letala (reg. 1)
    navigate();

    // spremeni vrednosti servomotorjev za
    // dosego zeljenega nagiba (reg. 2)
    stabilize();
}
```

Listing 3.2: Psevdokoda zanke avtopilota

Avtopilot poleg branja vrednosti senzorjev in reguliranja servomotorjev skrbi še za vrsto drugih opravil - preverjanje izbranega režima delovanja, beleženje

podatkov in telemetrijo, zlivanje vrednosti senzorjev in drugo. Ker vse naloge niso enako pomembne, ali pa so nekatere računsko zahtevnejše od drugih, bi bilo nesmotrno vse izvajati v enakih intervalih. APM zato naloge razdeli v tri skupine, ki jih izvaja v 50 Hz, 10 Hz in 1 Hz zankah (dejanska frekvenca izvajanja lahko nekoliko odstopa, prav tako pa so naloge v nekaterih skupinah razdeljene naprej v podskupine in se izvajajo še redkeje). Večji del nalog avtopilota je predstavljen v spodnjem diagramu.

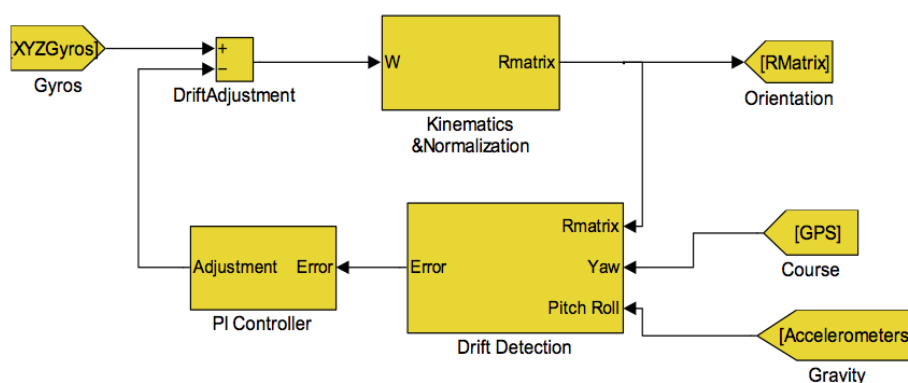


Slika 3.2: Diagram delovanja APM avtopilota

3.1.2 Zlivanje senzorjev

Med računsko zahtevnejšimi in pomembnejšimi nalogami avtopilota (do strojne verzije 2, ko to nalogo nadomesti specializiran čip) spada zlivanje oziroma povezovanje senzorjev (*Sensor Fusion*) [6]. Zlivanje je povezovanje vrednosti različnih senzorjev v neko boljšo celoto - boljšo v smislu natančnejše predstavitve z večjo dodano vrednostjo. Zlivanje senzorjev se na primer uporablja v IMU enoti pri računanju orientacije letala na podlagi podatkov žiroskopov in pospeškometerov. V tem primeru so za nas vektorji orientacije, hitrosti in položaja veliko pomembnejši kot posamezne surove vrednosti žiroskopov, akcelerometrov in GPS sprejemnika.

Senzorje lahko zlivamo na več različnih načinov - na primer s Kalmanovim filtrom, opisanim v [6], Bayesovimi mrežami ali DCM metodo (*Direction Cosine Matrix*), opisano v [18] in uporabljeno v APM. DCM se v primerjavi s Kalmanovim filtrom izkaže za skoraj enako natančnega pri mnogo manjši računski zahtevnosti. Pomankljivost DCM metode je le nekoliko večja občutljivost na *dobre* začetne podatke - IMU enota mora biti med začetno kalibracijo čim bolj vodoravna na površino Zemlje in mora mirovati.



Slika 3.3: Shema DCM algoritma (*Vir slike: [18]*)

3.2 Režimi letenja

APM lahko deluje v več različnih načinih delovanja, od katerih je eden namenjen inicializaciji naprave, eden povsem ročnemu načinu letenja, ostali pa

avtonomnemu letenju ali pomoči pri upravljanju letala.

V nobenem izmed režimov nismo povsem odrezani od nadzora nad letalom, saj lahko kadarkoli spremenimo režim delovanja, pri večini pa lahko med samim letenjem nekoliko vplivamo na smer letenja (letalo *dregnemo* iz njegove predvidene poti).

Ročni način

V ročnem načinu avtopilot le spremlja senzorje, beleži njihove vrednosti in skrbi za telemetrijo, nadzor nad močjo motorja in servomotorji pa je prepuščen pilotu. Ročni način lahko nadgradimo s stabilizacijo, ki (v primeru, da izpustimo krmilo) letalo avtomatično poravnava.

Vračanje k vzletni točki

Ob preklopu v ta režim avtopilot letalo vrne na koordinate, ki jih je shranil pred vzletom med inicializacijo. Ob povratku na *domače* koordinate letalo kroži, dokler ne prevzamemo nadzora nad letalom.

Avtonomni način

Med zanimivejšimi režimi delovanja je avtonomni način s sledenjem točkam. Za povsem avtonomno letenje pred poletom v pomnilnik avtopilota vnesemo misijo, ki jo mora letalo izvesti. Misije so sestavljene iz različnih ukazov, od preprostih 3D GPS koordinat, ki jih mora letalo preleteti, do različnih pogojev (dosežena višina, hitrost, oddaljenost). Misijo določimo pred poletom prek nadzorne postaje, če pa je letalo opremljeno s kompletom brezžične telemetrije, jih lahko urejamo med letom.

Fly-by-wire

V *fly-by-wire* načinu vrednosti krmilnih palic na našem oddajniku dejansko nadomestijo izhode prvega PID regulatorja (omenjen v podpoglavju *Kontrolna zanka avtopilota*), torej predstavljajo vrednosti želenega nagiba, naklona in hitrosti.

Obstajata dva različna fly-by-wire načina delovanja. Pri prvem (FBW A) avtopilot spremlja naklon in nagib letala, ki ju določa pilot s krmilnimi palicami, in jima ne dovoli izven določenega območja, moč motorja pa je povsem prepuščena pilotu. Pri drugem (FBW B) avtopilot potrebuje tudi vrednosti zračnega senzorja hitrosti. Avtopilot poskuša ohranjati nagib letala, določen s krmilnimi palicami (podobno kot pri prvem načinu), s kontrolnimi palicami za naklon pa spreminjamo želeno spremembo višine leta (napako višine), avtopilot pa to napako poskuša popraviti s spreminjanjem naklona in nagiba letala. V primeru, da senzorja hitrosti nimamo, se avtomatično preklopi na FBW A.

Avtomatizirano pristajanje in vzletanje

Pristajanje in vzletanje lahko določimo le v nadzorni postaji med načrtovanjem misije. Pri vzletanju moč motorja določa pilot, avtopilot le spremlja, da moč ne gre čez vnaprej določeno maksimalno vrednost. Ko letalo preseže določeno vzletno hitrost, fiksira smer in se vzpenja, dokler ne doseže zelene višine.

Podobno deluje pristajanje, le da avtopilot nadzira tudi moč motorja. Ko se letalo približa pristajalnemu mestu, določenem v misiji, fiksira smer in postopoma zmanjša moč motorja. V načrtu misije lahko določimo tudi različne dogodke, ki se prožijo med pristajanjem (proženje zračne zavore, spuščanje podvozja in podobno).

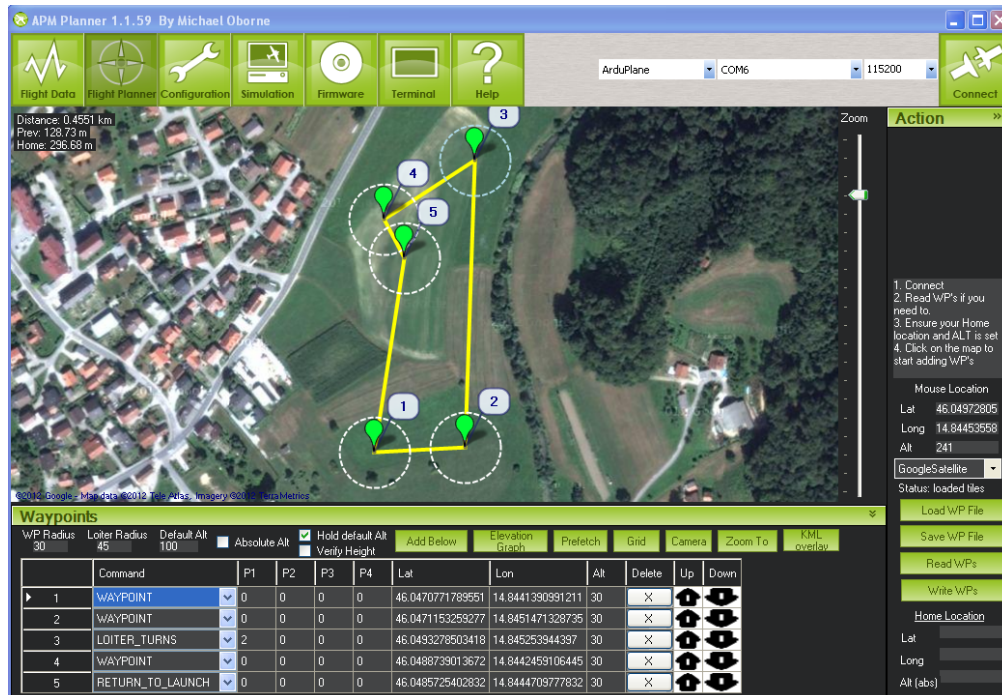
Obravnavanje napak

V primeru izgube signala z RV naprave za več kot 1.5 s (*short failsafe*) se APM preklopi v varnostni način. Glede na trenutni režim delovanja se preklopi na *stabilizacijo leta* (če smo imeli v ročnem načinu) ali *vračanje domov* (v vseh ostalih primerih). V primeru, da signal po izgubi ponovno pridobimo prej kot v 20 sekundah, se režim preklopi nazaj na prvotnega, če pa signal izgubimo za več kot 20 sekund (*long failsafe*), moramo režim preklopiti ročno.

3.3 Oddaljena nadzorna postaja

Oddaljena nadzorna postaja (*Ground Control Station*) je oprema, s katero človeški operater spremlja in nadzira delovanje plovila brez posadke - na grafičnih

prikazovalnikih in števcih spremlja vrednosti senzorjev ter plovilu pošilja ukaze, nalaga misije in podobno.



Slika 3.4: APM Planner nadzorna postaja med načrtovanjem misije poleta

APM lahko komunicira z nadzornimi postajami, ki podpirajo MAVLink protokol, nadzorne postaje, preizkušene s tem avtopilotom, pa so:

- APM Planner**
 Izdelana je posebej za APM. Deluje le na Windows platformi, vključuje pa tudi možnosti nalaganja programske opreme avtopilota, konfiguriranja APM in načrtovanje misij.
- HappyKillmore GCS**
 Tudi ta deluje le na Windows platformi, konfiguriranje APM pa je možno le z ročnim nastavljanjem s pisanjem ukazov v ukazno vrstico.
- QGroundControl**
 Izmed trojice je QGC edini prenosljiv na različne platforme (Windows, Linux, Mac) in podpira več različnih avtopilotov, ki za komunikacijo uporabljajo MAVLink protokol.

V diplomskem delu bom uporabil APM Planner, saj je napisan posebej za APM in poleg osnovnih funkcij nadzorne postaje (spremljanje parametrov leta) omogoča še nastavljanje parametrov avtopilota (PID konstant avtopilota, maksimalne in minimalne vrednosti aktuatorjev, hitrosti leta, hitrosti spreminjanja smeri in podobno), nalaganje programske opreme, branje in analiziranje shranjenih vrednosti senzorjev in podobno.

3.4 MAVLink protokol

MAVLink (*Micro Air Vehicle Communication Protocol*) [19] je protokol, namenjen komunikaciji majhnih zračnih plovil. MAVLink protokol se lahko uporablja tako pri komunikaciji med letalom in nadzorno postajo kot tudi med posameznimi komponentami avtopilota (na primer med IMU in krmilno enoto) ali celo med samimi procesi. Pri APM pa se MAVLink uporablja le za komunikacijo med APM in nadzorno postajo.

Protokol podpira množica avtopilotov - poleg ArduPilot Mega še:

- pxIMU Autopilot
- SLUGS Autopilot
- FLEXIPILOT
- SenseSoar

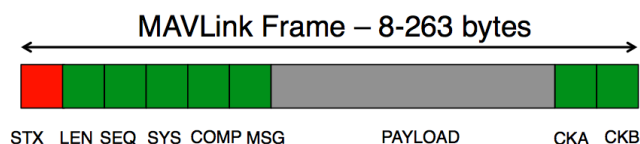
Med nadzornimi postajami pa je MAVLink podprt v:

- APM Planner (Windows)
- QGroundControl (Windows / Mac / Linux)
- HK Ground Control Station (Windows)
- Copter GCS (Android)

3.4.1 MAVLink sporočila

Bistvo protokola so MAVLink sporočila, ki vsebujejo podatke, ki jih pošiljamo, ena izmed njihovih glavnih prednosti pa je njihova majhnost, saj ima posamezen poslan paket le 8 bajtov pribitka. Najmanjša dolžina praznega sporočila

je torej 8 bajtov, največja pa je omejena na 263 bajtov (255 bajtov podatkov + 8 bajtov pribitka).



Slika 3.5: Anatomija MAVLink paketa

Indeks	Vsebina	Vrednost	Razlaga
0	začetek paketa	0xFE	Naznani začetek novega paketa.
1	dolžina podatkov	0-255	Pove dolžino podatkov v paketu.
2	številka paketa	0-255	Zaporedna številka paketa (omogoča zaznavanje izgubljenih paketov).
3	ID sistema	0-255	ID pošiljateljevega sistema (omogoča uporabo več različnih plovil z eno nadzorno ploščo).
4	ID komponente	0-255	ID komponente, ki pošilja sporočilo (omogoča razločevanje med več komponentami znotraj plovila).
5	ID sporočila	0-255	ID sporočila, ki pove, kaj podatki v sporočilu predstavljajo.
6 do (n+6)	Podatki	od 0 do 255 bajtov podatkov	Podatki sporočila.
(n+7) do (n+8)	kontrolna vsota	2 bajta kontrolne vsote	Zgoščena vrednost bajtov 1 do (n+6), izračunana z ITU X.25/SAE AS-4 zgoščevalno funkcijo.

Tabela 3.1: MAVLink paket

Definiranje sporočil

MAVLink sporočila definiramo v XML datotekah, ki jih nato z generatorjem prevedemo v C zaglavne (.h) datoteke. Te datoteke vsebujejo vso potrebno kodo za uporabo protokola in sporočil - torej ni potrebe po povezovanju našega programa z zunanji knjižnicami ali klicanja zunanjih knjižnic, kar vpliva na hitrejšo izvajanje. Poleg generatorja C kode obstajajo tudi generatorji za C++, C# in Python.

V spodnjem primeru *Ping* sporočila lahko vidimo, da je definicija sporočila sestavljena iz ID-ja, njegovega imena in opisa, podani pa so tudi atributi, ki to sporočilo sestavljajo, skupaj z njihovimi tipi in opisi.

```
<message id="4" name="PING">
  <description>A ping message either requesting or responding to a
    ping. This allows to measure the system latencies , including
    serial port , radio modem and UDP connections.</description>
  <field type="uint64_t" name="time_usec">Unix timestamp in
    microseconds</field>
  <field type="uint32_t" name="seq">PING sequence</field>
  <field type="uint8_t" name="target_system">0: request ping from
    all receiving systems , if greater than 0: message is a ping
    response and number is the system id of the requesting system
  </field>
  <field type="uint8_t" name="target_component">0: request ping
    from all receiving components , if greater than 0: message is
    a ping response and number is the system id of the requesting
    system</field>
</message>
```

Listing 3.3: XML definicija Ping sporočila

Generator nato zgornjo XML definicijo preslika v kodo, ki vsebuje strukture sporočila (C-jevski *struct*) in funkcije za serializacijo (*pack*), deserializacijo (*unpack*), kodiranje, dekodiranje in pošiljanje sporočila. Generirana koda Ping sporočila je razvidna v spodnjem izvlečku (telesa funkcij, argumenti in komentarji, ki jih generator ustvari avtomatično na podlagi XML definicije, so bili odstranjeni zaradi boljše preglednosti).

```
typedef struct __mavlink_ping_t {
    uint64_t time_usec;
    uint32_t seq;
    uint8_t target_system;
    uint8_t target_component;
```

```

} mavlink_ping_t;

static inline uint16_t mavlink_msg_ping_pack (...);
static inline void mavlink_msg_ping_send (...);
static inline uint16_t mavlink_msg_ping_encode (...);
static inline void mavlink_msg_ping_decode (...);

```

Listing 3.4: Definicija strukture sporočila in podpornih funkcij v jeziku C

APM poleg približno 70 splošnih MAVLink sporočil (sporočila za pošiljanje različnih ukazov, branje parametrov, senzorjev, spreminjanje vrednosti parametrov in podobno) vsebuje še 15 lastnih, povezanih predvsem z *ograjevanjem* avtopilota, ki območje leta omeji na določeno področje. Med zanimivejšimi pa so sporočila, povezana z načrtovanjem misij.

3.4.2 Načrtovanje misij

Preko nadzorne postaje lahko na avtopilot naložimo poljubne misije, ki jih nato letalo avtonomno izvede. Enostavnejše misije vsebujejo le letenje od točke do točke, kompleksnejše pa na določenih točkah oziroma pod določenimi pogoji prožijo različne akcije - na primer prožijo vgrajen rele, obrnejo letalo, spremenijo višino leta in podobno. Misije lahko načrtujemo na dva načina - s pošiljanjem ukaznih sporočil ali z *Waypoint* protokolom.

V prvem primeru misijo načrtujemo s pošiljanjem več sporočil *COMMAND_LONG*, s katerimi plovilu sporočamo seznam akcij, ki se morajo izvajati oziroma morajo biti dosežene. Avtopilot akcije s seznama prebira eno za drugo - ko zaključi s prvo, prebere naslednjo in odreagira skladno z njenim pomenom in parametri.

Vsako *COMMAND_LONG* sporočilo vsebujejo ID ukaza akcije in 7 parametrov, katerih pomen je odvisen od samega ukaza. V primeru ukaza *MAV_CMD_NAV_WAYPOINT*, ki letalu sporoči točko, ki jo mora plovilo obiskati, so ti parametri:

- Čas zadrževanja plovila na tej točki - uporablja se pri helikopterjih, pri letalih se ignorira.
- Tolerančni polmer, znotraj katerega se šteje, da je bila točka dosežena.
- Nadzor tirnice leta - ponavadi 0 (letalo leti skozi točko).
- Zelena smer plovila v točki (se ignorira pri letalih).

- Zemljepisna širina, dolžina in nadmorska višina točke.

Vseh ukazov, ki jih lahko določimo posamezni akciji v misiji, je 32, v spodnji tabeli pa je predstavljenih 10 pogosteje uporabljenih.

Ime ukaza	Opis
NAV_WAYPOINT	Navigacijska točka, ki jo želimo obiskati.
NAV_LOITER_TURNS	Naredi n krogov okrog točke.
NAV_RETURN_TO_LAUNCH	Vrni se k začetni točki.
NAV_LAND	Pristani na določeni točki.
NAV_TAKEOFF	Vzleti s tal/z roke.
CONDITION_DELAY	Zakasni izvajanje misije za n sekund.
CONDITION_DISTANCE	Zakasni izvajanje misije, dokler nismo oddaljeni n metrov od naslednje NAV točke.
CONDITION_CHANGE_ALT	Zakasni izvajanje misije, dokler ni dosežena določena nadmorska višina.
DO_SET_RELAY	Nastavi rele na 0 ali 1.
DO_SET_MODE	Nastavi režim delovanja avtopilota.

Tabela 3.2: Pogosteje uporabljeni MAV_CMD ukazi

Poglavje 4

ArduPilot Mega v praksi

Delovanje avtopilota sem v praksi preizkusil z modelom letala EasyStar. Model, izdelan iz trde, stiroporu podobne pene, sem izbral zaradi nizke cene, enostavnosti gradnje in morebitnih popravil ter zaradi dobre stabilnosti v zraku, zaradi česar dopušča nekoliko več napak pri eksperimentiranju z nastavitvami avtopilota.

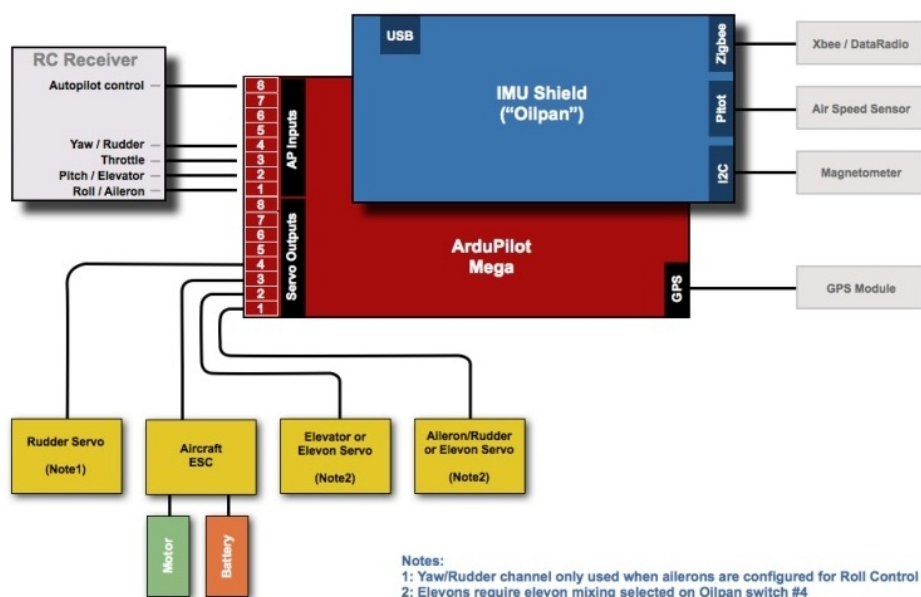
Glavni cilj testiranja je bil letalo usposobiti za povsem avtonomni let po vnaprej določeni poti. V tem poglavju so predstavljeni koraki, potrebni za vgradnjo avtopilota v letalo, nalaganje programske opreme, nastavitve in rezultati avtonomnega leta.



Slika 4.1: Letalo Easystar in pilotska kabina z avtopilotom, GPS modulom in vidno statično in dinamično Pitojevo cevjo.

4.1 Vgradnja APM v letalo

APM komplet je najprej potrebno sestaviti - prispajkati nekaj konektorjev, s katerimi povežemo glavno in IMU ploščo. Na glavno ploščo nato priklopimo GPS modul in senzor zračne hitrosti ter ustrezne izhode sprejemne naprave povežemo na vhode APM plošče, izhode pa na servomotorje in krmilnik elektromotorja.



Slika 4.2: Shema povezav (Vir slike: APM Wiki)

Povezan APM nato vstavimo v pilotsko kabino letala, pri čemer moramo biti pozorni na to, da leži čim bolj vodoravno in da je pravilno usmerjen (povezave servomotorjev in sprejemnika so usmerjene proti repu letala). GPS modul namestimo na izbrano odkrito mesto, da lahko vidimo, kdaj je GPS pridobil lokacijo in je letalo primerno za let - v mojem primeru na rob pilotske kabine. Da se APM in GPS med letenjem ne premikata ali padeta iz letala, ju v letalo čvrsto pritrdimo.

4.2 XBee konfiguracija

Preden uporabimo XBee modula, ju moramo ustrezno nastaviti in določiti parametre mreže, v kateri delujeta. Za nastavljanje modul preko USB vmesnika priklopimo na računalnik in uporabimo orodje X-CTU, ki poleg spreminjanja parametrov modula omogoča še nalaganje programske opreme, testiranje povezav ter branje nastavitvev na modulu.

V našem primeru moramo vzpostaviti povezavo med dvema vozliščema, od katerih mora biti eno obvezno koordinator (FFD), drugo pa preprosto končno vozlišče (RFD). Za naše potrebe lahko večino nastavitvev pustimo na privzetih vrednostih - na primer ID omrežja (PAN ID), način skeniranja kanalov, varnostne nastavitve in podobno, nastaviti moramo le lastna naslova obeh vozlišč in naslova vozlišča, s katerim se pogovarjata. Lasten naslov vozlišču določimo s parametroma SH (*Serial Number High*) in SL (*Serial Number Low*), naslov ciljnega vozlišča pa z DH (*Destination Address High*) in DL (*Destination Address Low*).

	Vrednost (FFD)	Vrednost (RFD)
SH	13A200	13A200
SL	406F4C75	406F4C55
DH	13A200	13A200
DL	406FC55	406F4C75

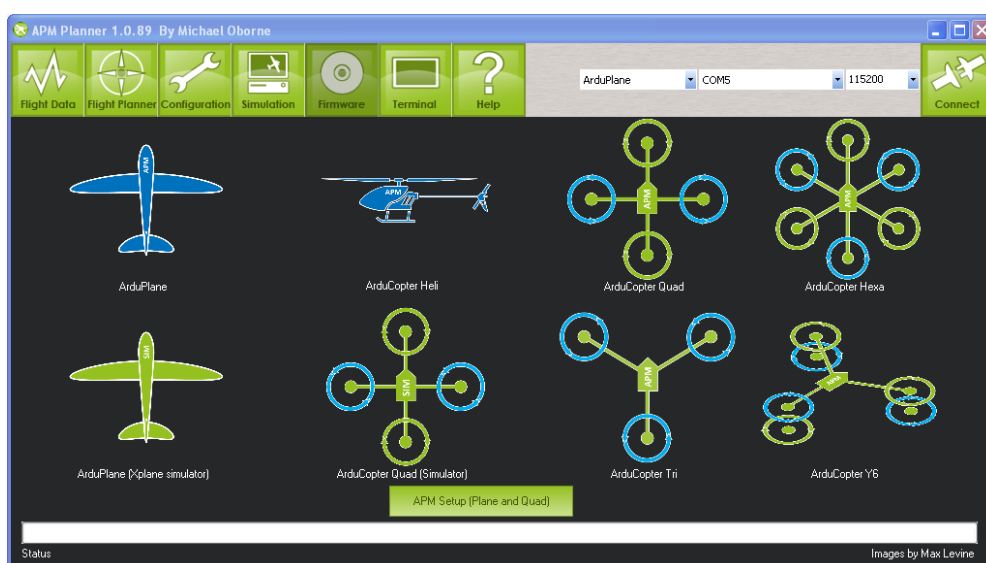
Tabela 4.1: Naslavljanje XBee modulov

Poleg naslavljanja spremenimo še hitrost komunikacije BD (*Baud Rate*) na 57600 baud/s. Ko sta oba modula ustrezno nastavljena, enega izmed njiju priključimo na letalo, drugega pa pustimo povezanega na računalnik, saj bo služil kot sprejemnik/oddajnik za nadzorno postajo.

4.3 Nalaganje programske opreme avtopilota

Za nalaganje ustrezne programske opreme APM z USB kablom povežemo na računalnik, ki priklopljen APM prepozna kot nov serijski vhod (COM port).

Program najlažje naložimo s pomočjo APM Planner nadzorne postaje, ki na spletu avtomatično poišče zadnjo stabilno verzijo programske opreme za naš model avtopilota in jo naloži. To storimo tako, da na zavihku *Firmware* kliknemo na tip naprave – v našem primeru ArduPlane – in zadnja verzija programske opreme se bo avtomatično naložila na naš avtopilot. Na tem zavihku vidimo, da platforma podpira več različnih naprav - letala, helikopterje in helikopterje z več fiksnimi rotorji - vse naprave so podprte z istim APM avtopilotom, a nekoliko različno programsko opremo.



Slika 4.3: APM Planner nadzorna postaja - izbira tipa letala za avtomatično nalaganje najnovejše verzije programske opreme

Ročno prevajanje kode

Program avtopilota je mogoče naložiti tudi ročno, z razvojnim okoljem Arduino. Na ta način lahko delovanje avtopilota prilagodimo svojim željam, kar sem izkoristil tudi sam. Atmega1280 mikrokontroler, vgrajen v avtopilot, ima namreč premalo pomnilnika za celoten program avtopilota, zato so določene funkcionalnosti v uradni programski opremi, ki se naloži z nadzorne postaje, izključene - med drugim je izključeno shranjevanje telemetrije na vgrajen flash pomnilnik. Ker sem poleg osnovne telemetrije želel tudi podatke iz žiroskopov in akcelerometrov, ki se zaradi večje količine ne pošiljajo brezžično, sem moral

kodo nekoliko prilagoditi (izklopiti določene neuporabljene funkcije in vklopiti beleženje podatkov), da sem v 128 kB notranjega pomnilnika namestil želene funkcionalnosti. Prilagajanje je bilo relativno enostavno, saj lahko precej funkcij omogočimo ali onemogočimo s pravilno definicijo C-jevskih makrojev.

Za zmanjšanje velikosti avtopilota na 128 kB sem moral izklopiti del logike za komunikacijo prek terminala - menije za pomoč, testiranje in nastavitve, saj jih ne potrebujem, ker je do njih mogoče dostopati prek nadzorne postaje. Prav tako sem izklopil avtomatično detekcijo GPS modula in naložil le knjižnice, potrebne za MediaTekov GPS, ki ga uporabljam. Na koncu sem vključil še beleženje podatkov in določil, kateri parametri naj se beležijo (GPS in RAW - podatki akcelorometrov in žiroskopov, ki se zajamejo in shranijo 50-krat v sekundi). Večino teh nastavitvev se lahko opravi v *config.pde* datoteki, z definicijo ustreznih makrojev.

Kodo avtopilota ročno naložimo tako, da na spletni strani projekta APM poiščemo zadnjo verzijo APM kode in v mapo z Arduino projekti kopiramo mapi *libraries*, kjer se nahajajo knjižnice senzorjev, IMU plošče in podobno, ter *ArduPlane*, v kateri se nahaja koda avtopilota v .pde in .h datotekah. Z Arduino razvojnim okoljem odpremo datoteko *ArduPilotMega.pde*, kar avtomatično odpre tudi vse ostale izvorne datoteke tega projekta. V meniju *Tools/Board* izberemo *Arduino Mega (ATmega1280)* in v *Tools/Serial Port* vrata, na katera je priklopljen APM. Nato projekt prevedemo in naložimo s klikom na gumb *Upload*. APM lahko ročno nastavljamo preko vgrajenega terminala s klikom na *Serial Monitor* ali pa preko nadzorne postaje.

4.4 Nastavitve in letenje

Ko na novo naložimo programsko opremo ali pobrišemo nastavitve iz EEPROMa, moramo ponovno nastaviti parametre avtopilota - omogočiti opsijske zunanje naprave (senzor zračne hitrosti, magnetni kompas, sonar) in kalibrirati razpon vrednosti, ki jih avtopilot prejme iz RV naprave. Pri tem si lahko pomagamo z APM Planner orodjem (zavihkom *Configuration*).

Poleg osnovnih nastavitvev lahko spremenimo tudi množico drugih parametrov, ki določajo obnašanje avtopilota:

- parametre PID regulatorjev,
- maksimalno, minimalno in potovalno zračno hitrost letenja,

- maksimalne in minimalne dovoljene naklone oziroma nagibe letala,
- razpon moči motorja,
- druge konstante, povezane z delovanjem APM platforme.

Pred poletom določimo, katere režime lahko izbiramo z RV napravo. Kot je razvidno iz diagrama slike 4.2, je en kanal sprejemnika povezan na osmi vhod avtopilota, ta pa je namenjen nadzoru nad avtopilotom - preklapljanju med režimi delovanja. Med koliko različnimi režimi lahko preklapljam, je odvisno od RV naprave - z dvopozicijskim preklopnim stikalom lahko na primer izbiramo le med dvema, od katerih je eden obvezno ročni.

Ko imamo APM in RV napravo ustrezno nastavljena, lahko letalo tudi dejansko preizkusimo. Pred priklopom pogonske baterije, ki napaja tudi avtopilota, vklopimo RV napravo in prekllopimo stikalo za izbiro režima v ročni način, nato priklopimo napajanje in med utripanjem LED diod ABC (kalibracija senzorjev) letalo pustimo vodoravno. Ko diode prenehajo z utripanjem in GPS modul dobi signal (modra LED dioda na modulu sveti), je letalo pripravljeno na polet. Vzletimo lahko v ročnem načinu ali pa prekllopimo na sekundarni režim delovanja, če je primeren za vzlet (na primer *Fly-by-wire* ali *Stabilize* način). V zraku lahko med ročnim in sekundarnim načinom poljubno preklapljam. Če med preizkusom RV napravo ugasnemo in s tem simuliramo izgubo signala, bo letalo nemudoma preklopilo v varnostni način.

Dogajanje na letalu lahko z uporabo brezžične telemetrije v živo spremljamo na nadzorni postaji - smer, položaj, hitrost ter še nekatere druge podatke s senzorjev.

4.5 Preizkusna misija

Za test delovanja avtopilota sem izdelal enostavno misijo s poligonom v obliki pravokotnika, ki naj bi ga letalo preletelo brez mojega posredovanja (razen vzleta in pristanka). Testna misija vsebuje 5 akcij, od tega 4 točke, ki jih mora letalo obiskati, ob koncu pa se mora vrniti k vzletni točki in nad njo krožiti. Testiranje je potekalo v brezveterju, cilj pa je bil ugotoviti, kako uspešno avtopilot preleti zadano pot. Pri načrtovanju misije moramo biti pozorni na višine točk, ki jih vnašamo, in morebitnih ovir med njimi (drevesa, zgradbe), paziti pa moramo tudi na velikost polmera kroženja pri morebitnem kroženju okrog točk.

	Ukaz	P1	P2	P3	P4	Lat	Lon	h
1	WAYPOINT	0	0	0	0	46.0470771	14.8441390	20
2	WAYPOINT	0	0	0	0	46.0471153	14.8451471	20
3	LOITER_TURNS	2	0	0	0	46.0493278	14.8452539	20
4	WAYPOINT	0	0	0	0	46.0488739	14.8442459	20
5	RETURN_TO_LAUNCH	0	0	0	0	46.0485725	14.8444709	20

Tabela 4.2: Misija poleta

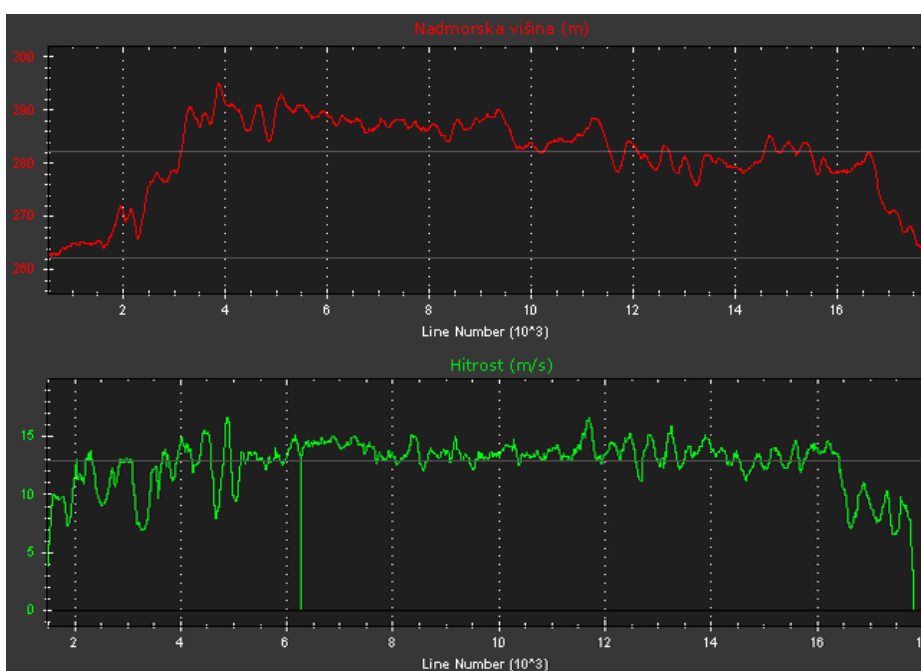


Slika 4.4: Pot leta

Z letalom sem v ročnem načinu vzletel s točke 0 in poletel proti vzhodu ter naredil nekaj krogov, da je letalo pridobilo višino (zeleno črta). Nato sem vključil avtopilot (prehod med zeleno in modro črto) in letalo se je obrnilo proti zahodu k točki 1, ki jo je doseglo na njeni južni strani, nad njo zakrožilo

in nadaljevalo proti točki 2 in dalje k točki 3. V tej točki so bila navodila avtopilotu, naj nad njo naredi 2 kroga s polmerom 45 m in nato nadaljuje pot proti 4. točki. Ko jo je dosegel, mu je naslednja akcija velela preklopiti v način vračanja k vzletni točki (*return-to-launch*), zato se je letalo vrnilo nad vzletno točko in nad njo krožilo s polmerom 45 m, nakar sem ponovno prevzel nadzor nad letalom in pristal.

Kot je razvidno z označene poti na sliki 4.4, je letalo približno 3700 metrov načrtovane poti (2900 metrov avtonomno) obletelo dobro, brez pretiranega vijuganja. Na večjem poligonu, kjer bi bile točke med seboj še bolj oddaljene, bi se preleti posameznih točk verjetno pokazali izraziteje. Na zgornjem poligonu, kjer so točke oddaljene od 50-120 m, pa ima letalo le nekaj sekund, da se mirno usmeri k naslednji točki, zaradi česar naredi ovinek več, kot bi od njega pričakovali. Prav tako bi avtopilot pot verjetno izbral drugače, če bi bral več kot samo eno (trenutno) akcijo v prihodnost.



Slika 4.5: Grafa nadmorske višine in hitrosti letala

Vse točke so imele nadmorsko višino določeno 20 m nad višino vzleta. Kot je razvidno z grafa na sliki 4.5, je letalo to višino (približno 282 metrov) vzdrževalo s približno ± 5 metrov odstopanja - na začetku nekoliko več, saj

sem v ročnem načinu poletel nekoliko previsoko, nato pa se je letalo postopoma spustilo nižje. Na grafu je razvidna tudi hitrost letala, ki je bila približno 13 m/s, kot je določeno v nastavitvah avtopilota.

Poglavje 5

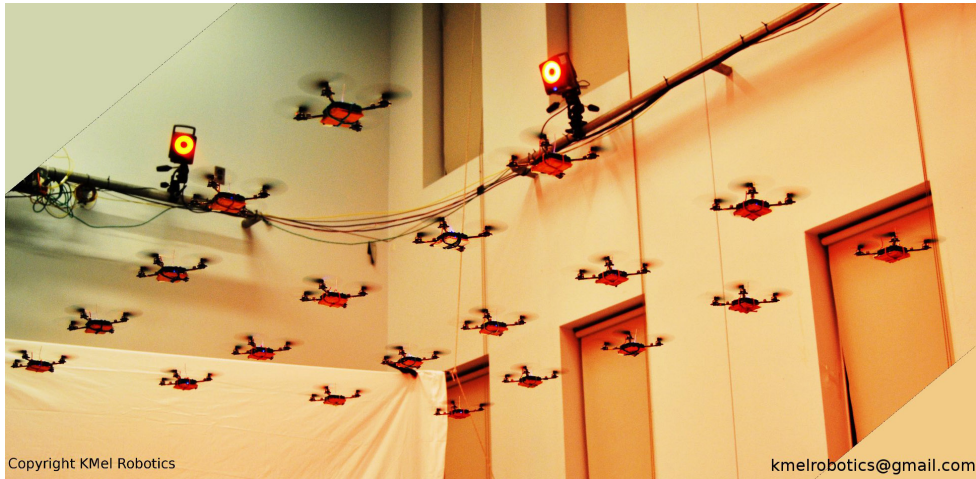
Zaključek

Področij uporabe (mikro) letal brez posadke je mnogo - nadzor, kartografija, raziskave, vojska in ne nazadnje zabava. V tem pregledu ArduPilot Mega avtopilota in podporne programske opreme ter preizkusu na terenu se je avtopilot izkazal kot stabilna naprava, ob podpori APM Plannerja pa sta bila nastavljanje in kalibracija enostavna in intuitivna. Ročno letenje ob podpori stabilizacije se je izkazalo za neprimerno lažje, saj pilotu ne dovoli izvajati manevrov, ki bi lahko vodili k nestabilnosti in strmoglavljenju. Pozitivno presenečen pa sem bil tudi nad povsem avtonomnim letenjem po vnaprej načrtani misiji, saj je letalo že ob prvem poizkusu takšnega leta brez kakršnihkoli težav izvedlo načrtovano misijo.

Ob težavah pri ročnem nalaganju celotne programske opreme avtopilota, ki je bila prevelika za Atmelov ATmega1280 mikroprocesor, vgrajen v to verzijo avtopilota, pa se je pokazala še ena izmed dobrih strani platforme - koda avtopilota je berljiva in enostavna za prilagajanje lastnim željam in potrebam. Poleg enostavnosti uporabe in zrelosti programske opreme, ki z oddaljeno nadzorno postajo omogoča ustvarjanje relativno kompleksnih misij, pa je velik plus APM platforme tudi nizka cena - avtopilot stane približno 200 evrov, celoten komplet z letalom, RV napravo in baterijami pa je mogoče sestaviti že za približno 500 evrov.

Poleg vgradnje v letalo lahko APM vgradimo tudi v model helikopterja ali helikopterja z več fiksnimi rotorji (*Quad-copter*, *Hex-copter*), kar odpira vrsto drugih možnosti uporabe, saj je ta plovila mogoče upravljati tudi v zaprtih prostorih, z njimi lebdeti nad določeno točko in, če jih opremimo z video kamero, opazovati okolico. Tudi na tem področju potekajo različne raziskave.

Med zanimivejše, ki sem jih zasledil, nedvomno sodijo raziskave z Univerze Pennsylvania pod vodstvom Daniela Mellingerja [21]. Mellinger se s sodelavci svojega laboratorija ukvarja predvse z nadzorom *nano quad-copterjev* in njihovim natančnim vodenjem v 3D prostoru.



Slika 5.1: Roj nano quad-copterjev

Tabele

2.1	Primer NMEA GPGGA stavka	15
3.1	MAVLink paket	28
3.2	Pogosteje uporabljeni MAV_CMD ukazi	31
4.1	Naslavljanje XBee modulov	34
4.2	Misija poleta	38

Slike

1.1	Letalo brez posadke MQ9 Reaper	4
2.1	APM komplet pred spajkanjem	8
2.2	ArduPilot Mega glavna plošča	9
2.3	ArduPilot Mega IMU plošča	10
2.4	ArduPilot Mega 2	11
2.5	MEMS žiroskop	12
2.6	Barometrična enačba	12
2.7	Senzor zračne hitrosti na osnovi Pitojeve cevi	13
2.8	Bernoullijeva enačba in hitrost zračnega toka	13
2.9	MediaTek 3329 GPS modul z anteno	14
2.10	Brezžična XBee telemetrija	16
2.11	Plasti ZigBee in IEEE 802.15.4 standardov	17
2.12	Servomotor za radijsko vodene modele	18
3.1	Shema PID regulatorja	21
3.2	Diagram delovanja APM avtopilota	22
3.3	Shema DCM algoritma	23
3.4	APM Planner nadzorna postaja	26
3.5	Anatomija MAVLink paketa	28
4.1	Pilotska kabina z avtopilotom	32
4.2	Shema povezav	33
4.3	APM Planner	35
4.4	Pot leta	38
4.5	Grafa nadmorske višine in hitrosti letala	39
5.1	Roj nano quad-copterjev	42

Literatura

- [1] *Arduino domača stran*
<http://arduino.cc/>
- [2] *ArduPilot Mega*
<http://code.google.com/p/ardupilot-mega/>
- [3] (1982) Valavanis K., *Advances in Unmand Aerial Vehicles*
- [4] (2010) Dempsey, Martin E., *U.S. Army Roadmap for Unmanned Aircraft Systems 2010–2035*
<http://www-rucker.army.mil/usaace/uas/US%20Army%20UAS%20RoadMap%202010%202035.pdf>
- [5] (1982) Wagner W., *Lightning Bugs and other Reconnaissance Drones*
- [6] (2008) Ranfl U., *Uporaba Kalmanovega filtra pri povezavi različnih senzorjev za določanje položaja v cestnem mobilnem kartirnem sistemu*
- [7] *Triple Axis (X/Y/Z) MEMS Gyroscopes*
<http://invensense.com/mems/gyro/tripleaxis.html>
- [8] *IDG-500 Datasheet*
http://www.sparkfun.com/datasheets/Components/SMD/Datasheet_IDG500.pdf
- [9] Bogue R., *The fast-moving world of MEMS technology*
- [10] Barometrična enačba
http://en.wikipedia.org/wiki/Barometric_formula#Density_equations
- [11] NMEA 1083 protokol
http://en.wikipedia.org/wiki/NMEA_0183
- [12] *GPS - NMEA sentence information*
<http://aprs.gids.nl/nmea/>

- [13] *IEEE 802.15.4 Task Group*
<http://www.ieee802.org/15/pub/TG4.html>
- [14] *XBee/XBee-PRO ZB RF Modules*
ftp://ftp1.digi.com/support/documentation/90000976_F.pdf
- [15] *Wikipedia - ZigBee*
<http://en.wikipedia.org/wiki/ZigBee>
- [16] *Wikipedia - RC Servomechanism*
http://en.wikipedia.org/wiki/Servomechanism#RC_servos
- [17] (2010) Lotrič U., *Skripta predavanj Procesna Informatika, Fakulteta za računalništvo in informatiko*
- [18] (2009) Premerlani W., Bizard P., *Direction Cosine Matrix IMU Theory*
<http://gentlenav.googlecode.com/files/DCMDraft2.pdf>
- [19] *MAVLink Micro Air Vehicle Communication Protocol*
<http://qgroundcontrol.org/mavlink/start>
- [20] *ArduCopter Project Page*
<http://code.google.com/p/arducopter/wiki/ArduCopter>
- [21] Daniel Mellinger et al,
<https://fling.seas.upenn.edu/~dmel/wiki/index.php?n=Main.Publications>

Št. naloge: 01819/2012

Datum: 15.03.2012



Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SAMO KRALJ**

Naslov: **LETALO BREZ POSADKE Z ARDUPILOT MEGA AVTOPILOTOM
UNMANNED AIRCRAFT WITH THE ARDUPILOT MEGA AUTOPILOT**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Letalo brez posadke je naprava, ki jo je mogoče upravljati na daljavo ali pa jo vodi v letalo vgrajen avtopilot. Takšna letala imajo v določenih primerih pomembne prednosti pred letali s človeško posadko in so zato zanimivo področje raziskovanja. V ta namen raziščite možnosti za izdelavo preprostega letala te vrste. Kot osnovo uporabite odprtokodni projekt ArduPilot Mega in njegovo strojno platformo. Izberite ustrezne senzorske in aktuatorske komponente ter izdelajte vezje avtopilota vključno z radijsko komunikacijo. Izdelajte tudi vso potrebno programsko opremo in vgradite avtopilot v model letala. Pravilnost delovanja preverite na primeru vnaprej podane misije letenja in podajte možne načine za izboljšave.

Mentor:


prof. dr. Dušan Kodek

Dekan:


prof. dr. Nikolaj Zimic

