



Srednja šola za kemijo,
elektrotehniko in računalništvo

IZDELAVA MULTIKOPTERJA

Raziskovalna naloga

Mentor:

Gregor Kramer, univ. dipl. inž. el.

Avtorja:

Matej Žaberl, E-3. B

Gašper Trupej, E-3. B

Kazalo vsebine

Vsebina

Kazalo vsebine	2
Kazalo slik	3
Zahvala	4
Povzetek	5
Ključne besede	5
Uvod	6
1. Hipoteze	7
2. Opis raziskovalne metode	7
2.1. Metode dela	7
3. Osrednji del naloge.....	11
3.1. Predstavitev modula Arduino.....	11
3.1.2 Čip AtMega 328	11
3.1.3 Izvedbe Arduino modulov	11
3.2 Arduino Nano	11
4. Girooskop	13
4.1. Elektronski girooskop in merilec pospeškov MPU-6050	13
4.1.1. Karakteristike giroskopa	13
4.1.2. Posebnosti lastnosti MPU-60X0 in posebnosti MPU-6050	14
4.1.3. Princip delovanja elektronskega giroskopa.....	14
4.1.4. Pogonski motor A2212/13T.....	14
4.1.5. 30A brezkrtačni ESC.....	15
4.1.6. 1045 Elise.....	15
4.2. Preprosta shema povezovanja komponent trikopterja	17
4.3. Program	18
5. Razprava	22
6. Zaključek.....	23
Seznam virov	24

Kazalo slik

Slika 1: Primer trikopterja.....	6
Slika 2: Gugalnica z DC-motorji, pogled zgoraj.....	7
Slika 3: Gugalnica z motorji, pogled od strani	8
Slika 4: Pripravljen motor na ohišju.....	8
Slika 5: Ohišje z dvema motorjema	9
Slika 6: Opazovanje periode signala iz daljinca	10
Slika 7: Testiranje motorjev, povezanih z Arduino modulom	10
Slika 8: Čip AtMega328.....	11
Slika 9: Modul Arduino Nano.....	12
Slika 10: Girooskop	13
Slika 11: MPU-6050	13
Slika 12: Princip delovanja elektronskega giroskopa	14
Slika 13 motor A2212	15
Slika 14: ESC 30A	15
Slika 15: Elise 1045	16
Slika 16: Prvi program.....	18
Slika 17: Prvi del drugega programa – branje vrednosti giroskopa.....	19
Slika 18: Drugi del drugega programa - izpisovanje na LCD zaslonu	19
Slika 19: Prvi del končnega programa	20
Slika 20: Drugi del končnega programa.....	20
Slika 21: Izpisovanje stopinj nagiba na LCD-zaslonu	21

Zahvala

Zahvaljujeva se mentorju, gospodu Gregorju Kramerju, ki je s svojimi nasveti veliko pripomogel pri doseganju najinih zelenih ciljev, Anžetu Žaberlu, ki nama je pomagal pri nabavi motorjev, profesorju Mateju Kališku, sošolcema Domnu Lipniku in Jaku Roberju ter vsem ostalim, ki so nama ali s svojim znanjem ali kako drugače pomagali pri izvedbi projekta.

Povzetek

V raziskovalni nalogi sva raziskala delovanje multikopterjev, njihovo upravljanje in programiranje. Ideja za to raziskovalno nalogo se je porodila kar tako, ob spremljanju medijev in njihovega nenehnega omenjanja dronov in razvoju le-teh. Stvar se je sprva zdela zelo preprosta, ampak ko sva se poglobila, sva ugotovila, kako široko je pravzaprav to področje. Ob delu sva si zelo razširila obzorje glede programiranja in uporabe mikrokontrolerov Arduino in obenem spoznala elemente, ki so malo manj vsakdanji, na primer giroskop.

Ključne besede

- Arduino Nano
- AtMega 328
- giroskop MPU-6050
- brezkrtačni elektromotor
- trikopter

Uvod

Multikopterji so v današnjem času zelo razširjeni. Njihove cene se gibajo od nekaj deset evrov za osnovne modele, pa vse do nekaj tisoč evrov za modele z dodatnimi senzorji, namenjenimi vojski, okoljevarstvenikom in reševalcem. Ob spremljanju razvoja teh naprav sva si zastavila vprašanje, ali bi ga mogla izdelati sama. Odločila sva se, da bova naredila trikoopter.

Ko sva raziskovala komponente, ki jih trikoopter vsebuje, sva spoznala, da jih je možno tudi posamezno kupiti v trgovini z ustrezno opremo. Preprost trikoopter, ki ga bova izdelala, bova upravljala v smeri naprej – nazaj in v obračanju levo – desno. Vseboval bo tri brezkrtačne pogonske motorje in servo-motor za obračanje v vertikalni osi. Krmiljen sistem bo Arduino Nano. Prav tako bo imel trikoopter v zraku regulacijo stabilizacije, ki jo bo upravljal z žiroskop-senzorjem MPU6050. Raziskovala bova po korakih. Najprej bova preučila delovanje Arduina in senzorja ter komunikacije med njima, nato izpisovanje podatkov na display. Potem bova preučila delovanje regulatorja vrtljajev za brezkrtačne motorje, nato pa bova napisala program za stabilizacijo v zraku in ustrezno krmilje za pogon.

Prvo brezpilotno plovilo, ki je kadarkoli obstajalo, se je imenovalo 'zračna tarča' (angl. aerial target). Izdelal ga je Archibald Montgomery Low leta 1917. Za izstrelitev je uporabil stisnjen zrak.

Trikoopter je možno voditi v smeri naprej – nazaj s tremi motorji, če se dva uravnava, tretji pa služi za nagib v želeno smer v osi naprej – nazaj. Če želimo trikoopter obračati po vertikalni osi, potrebujemo še četrti servo-motor, ki nagiba tretji motor v želeno smer.

Trikoopterji imajo bistveno prednost v svoji kategoriji, saj imajo le tri pogonske motorje, s katerimi se že lahko omejeno gibljejo po prostoru. Stabilizacija je bistveno enostavnejša kot pri drugih vrstah multikopterjev, saj se stabilizirata ali le prednja motorja ali pa tudi zadnji s sprednjima, kar je še vedno manj zapleteno kot stabilizacija na primer štirih motorjev pri štirikoopterju. Program in samo krmilje sta bistveno enostavnejša kot pri drugih multikopterjih.



Slika 1: Primer trikoopterja

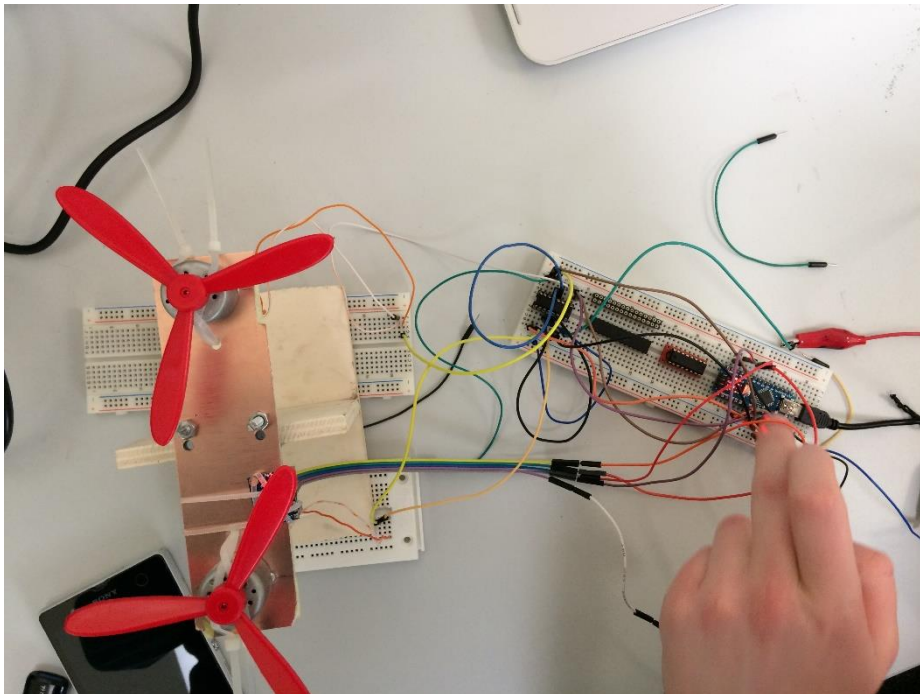
1. Hipoteze

- Dron kljub zunanjim motnjam sam drži ravnotežje v eni osi.
- Trikopter je mogoče voziti v smeri naprej – nazaj.
- Trikopter je mogoče v prostoru voditi okoli vseh treh osi.

2. Opis raziskovalne metode

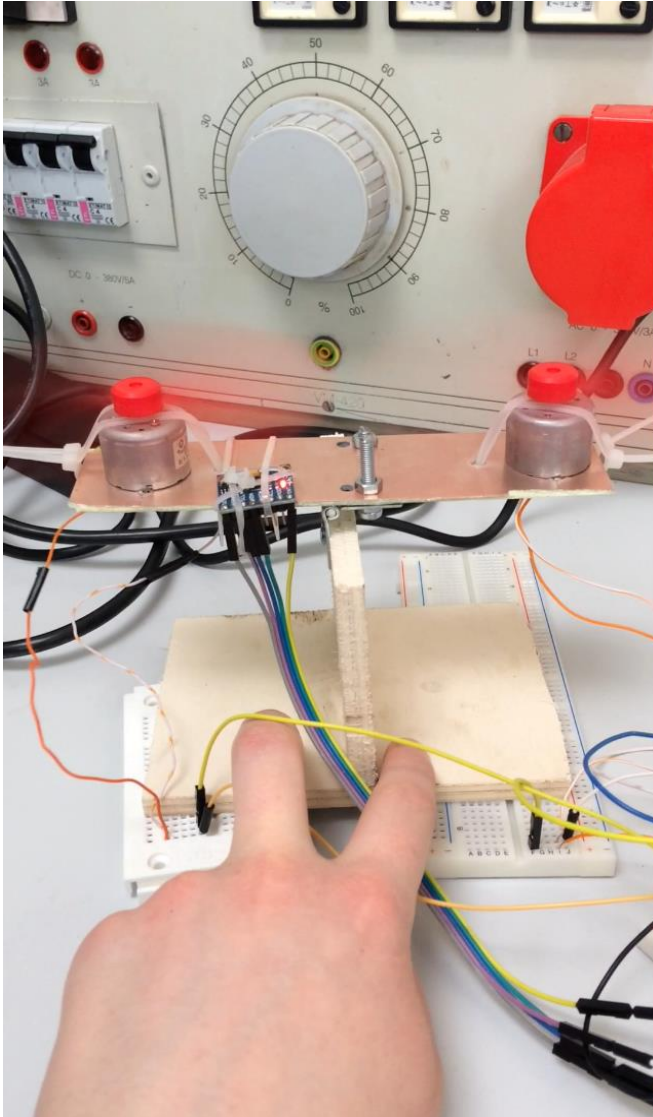
2.1. Metode dela

Preden sva začela z izdelavo samega trikopterja, sva se morala odločiti, katere komponente bova uporabila in jih tudi preučiti. Začela sva z izbiro Arduino modula. Zaradi svoje majhnosti se je za najbolj uporabnega izkazal Arduino Nano. Trikopter ne more leteti brez nenehnega uravnavanja samega sebe. Za to je seveda potreben giroskop. Izbrala sva model MPU-6050. Pred njegovo dejansko uporabo je bilo potrebno preučiti njegovo delovanje. Najprej sva na Arduino prenesla program, ki je prebral surove vrednosti iz giroskopa in jih izpisal na LCD-zaslon. Potem sva te vrednosti uporabila za prižiganje LED diod. Prižiganje je temeljilo na samem nagibu giroskopa. Surove vrednosti sva s pomočjo preprostega deljenja s konstanto spremenila v stopinje nagiba, ki sva jih izpisala na zaslonu. Nadaljevala sva z uporabo navadnih DC-motorjev, ki sva jih namestila na gugalnico, ki se je uravnavala s pomočjo giroskopa.



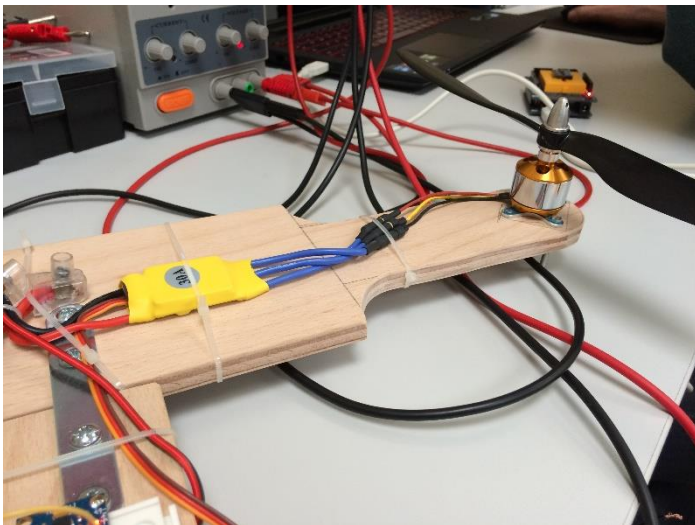
Slika 2: Gugalnica z DC-motorji, pogled zgoraj

Srednja šola za kemijo, elektrotehniko in računalništvo
IZDELAVA MULTIKOPTERJA



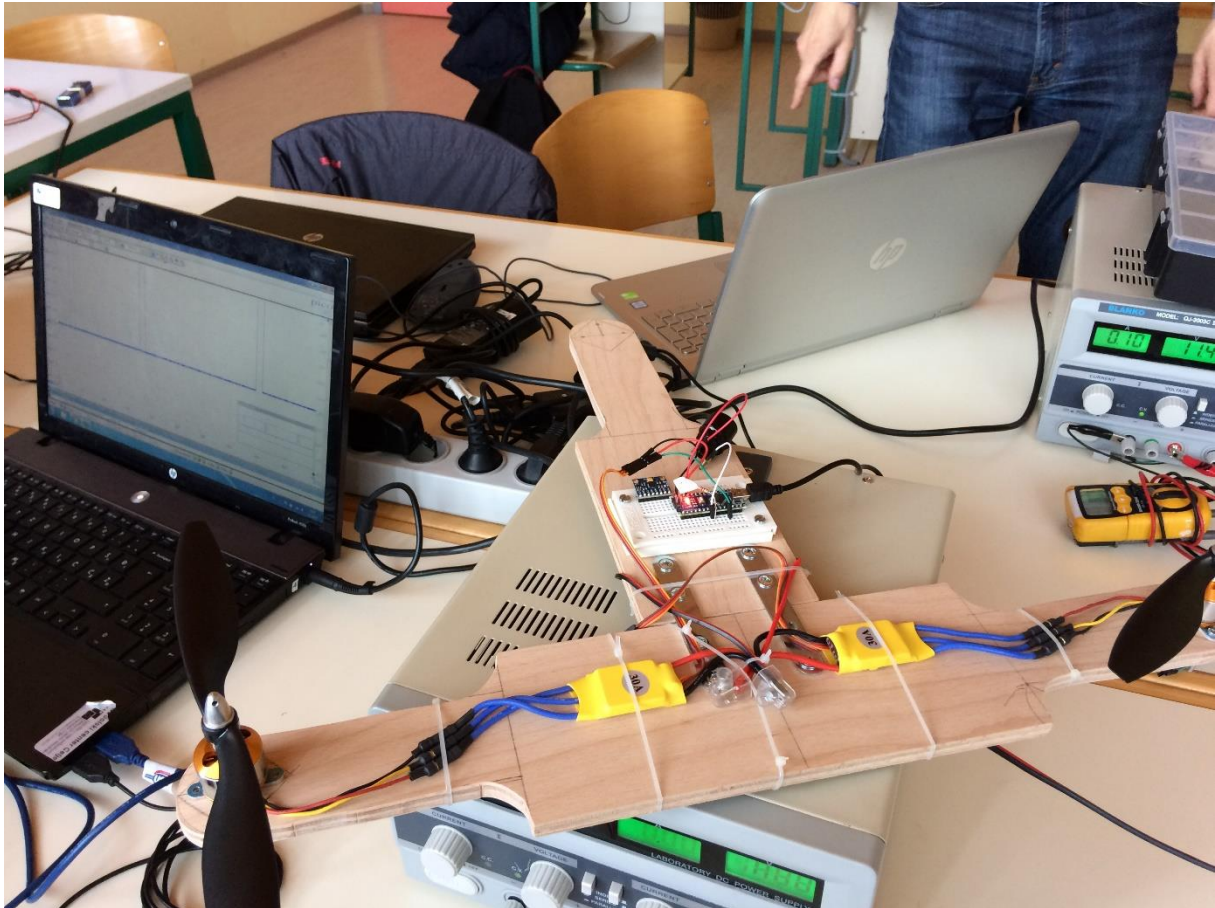
Slika 3: Gugalnica z motorji, pogled od strani

Ko so prispeli brezkrtačni motorji, ki sva jih naročila preko Ebay-ja, sva jih pripravila za uporabo (namestila elise, jim privijačila podnožja, prispajkala priključke) in preizkusila njihovo delovanje na Arduino modulu.



Slika 4: Pripravljen motor na ohišju

Sledila je izdelava ohišja, ki sva ga izdelala iz vezane plošče bukovega lesa. Zanj sva se odločila zaradi njegove trdote, ki onemogoča zvijanje ohišja. Izbrala sva debelino 1cm. Ohišje tehta približno 1 kg, kar za motorje ne predstavlja posebne težave, saj lahko eden dvigne okoli 800 g. Na ohišje sva namestila motorje, regulatorje, giroskop in sam Arduino modul.

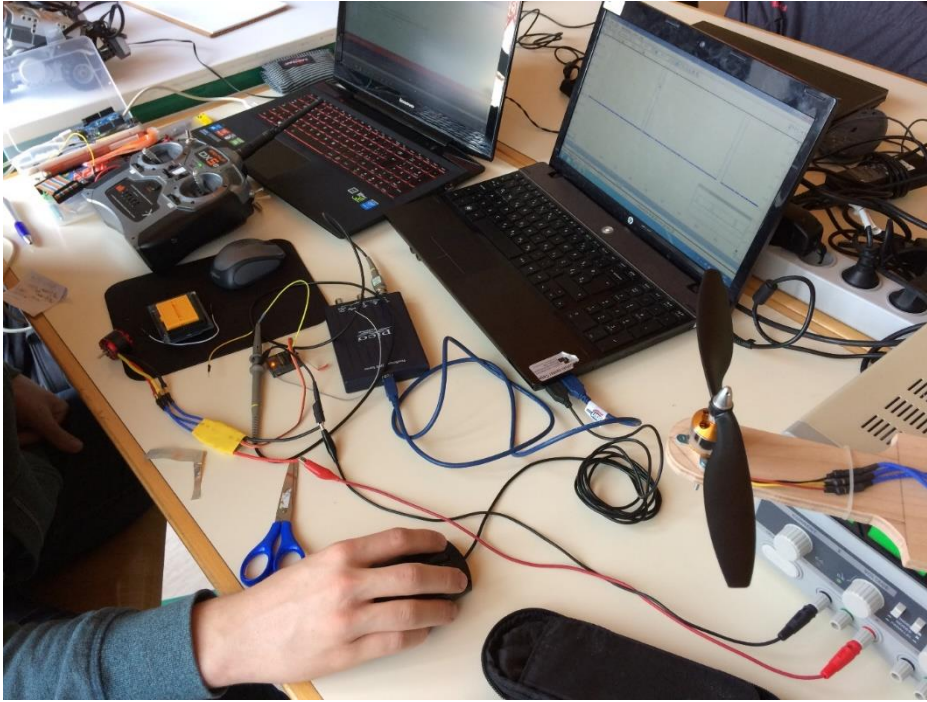


Slika 5: Ohišje z dvema motorjema

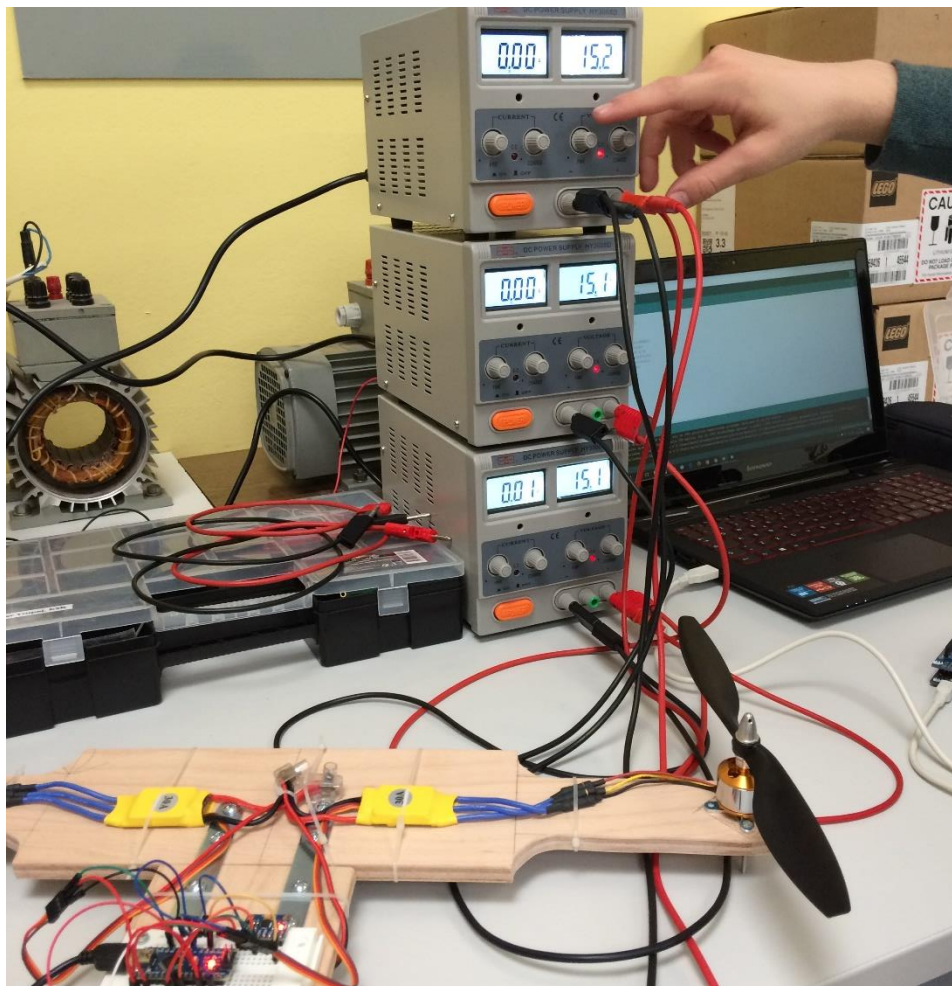
Raziskovalnega dela sva se lotila delno doma, delno v šoli. Na fotografijah je prikazanih nekaj stopenj izdelave in testiranja.

Opazovala sva signal, ki prihaja iz daljinca in tako sva lažje ugotovila, kako napisati program za krmiljenje motorjev.

Srednja šola za kemijo, elektrotehniko in računalništvo
IZDELAVA MULTIKOPTERJA



Slika 6: Opazovanje periode signala iz daljinca



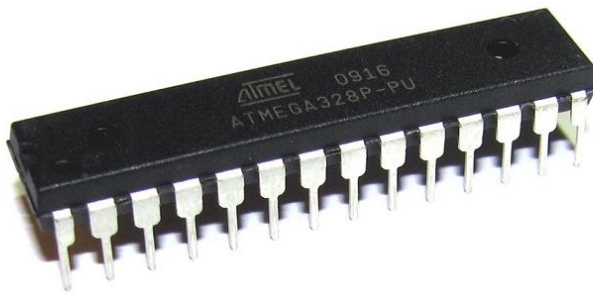
Slika 7: Testiranje motorjev, povezanih z Arduino modulom

Motorje sva povezala najprej na Arduino modul in jih pognala, nato sva v vezje vključila še giroskop za nadzorovanje vrtenja, ki je temeljil na kotu nagiba. Zaradi velike porabe toka motorjev sva morala uporabiti 3 usmernike, ki sva jih na isti napetosti povezala vzporedno, da sva lahko zagotovila optimalen rezultat delovanja motorjev.

3. Osrednji del naloge

3.1. Predstavitev modula Arduino

3.1.2 Čip AtMega 328



Slika 8: Čip AtMega328

Uporabila sva čip AtMega328, ki vsebuje 32 8-bitnih, splošno namenskih registrov. Ima 4/8/16/32 kilobytni »flash« spomin za shranjevanje prenesenega programa; 256/512/1 kilobytni EEPROM in 512/1K/2 kilobytni začasni pomnilnik SRAM. Deluje na napetosti 1.8 – 5.5 V, v temperaturnem območju

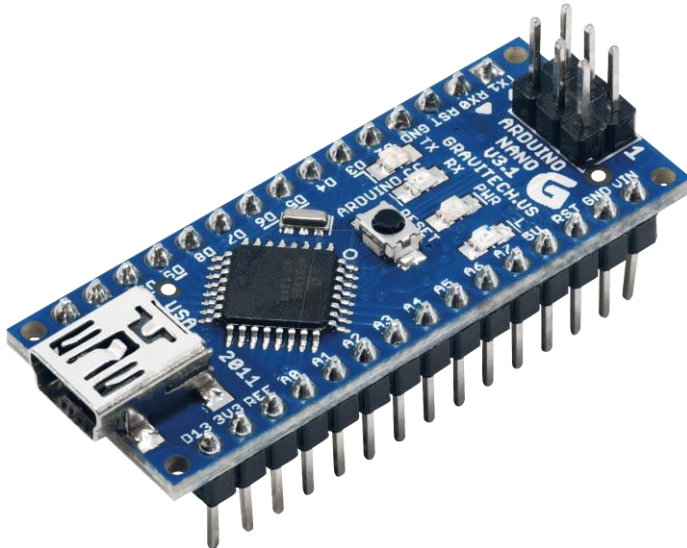
40 °C – 85 °C. Na voljo so tri izvedbe: pri prvi ima oscilator frekvenco od 0 do 4 MHz, deluje na napetosti 1,8 – 5,5 V; druga ima oscilator s frekvenco od 0 do 10 MHz, lahko ga priključimo na napetost 2,7 – 5,5 V; zadnja, tretja izvedba ima frekvenco obratovanja od 0 do 20 MHz na napetosti 4,5 – 5,5 V. Pri frekvenci 1 MHz, napetosti 1,8 V in temperaturi 25 °C ima vključen porabo 0,2 mA, v izklopljenem načinu pa 0,1 μA. (Vir: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf)

3.1.3 Izvedbe Arduino modulov

Arduino modul je lahko v več izvedbah, npr. Arduino Uno, Arduino Nano, Arduino 101, Arduino Mega, Arduino Tian in druge. Midva sva zaradi svoje majhnosti in uporabnosti v primeru trikopterja izbrala Arduino Nano. (Vir: <https://www.arduino.cc/en/Main/Products>)

3.2 Arduino Nano

Modul Arduino Nano je eden najmanjših izvedb tovrstnih modulov. Vsebuje enak procesor kot Arduino Uno, in sicer AtMega328.



Slika 9: Modul Arduino Nano

Iz slike je razvidno, da je Arduino Nano sestavljen iz: mini USB-priključka, ki služi za napajanje in prenos programa, mikroprocesorja AtMega328, tipke RESET, 8 analognih priključkov, ki služijo le kot vhodi in merijo napetost od 0 V do 5 V. Vsi analogni pini, razen 6 in 7, so lahko uporabljeni kot digitalni. Priključka 4 (SDA) in 5 (SCL) podpirata I2C komunikacijo, pri tem pa je potrebno upoštevati, da za to potrebujemo ustrezno knjižnico. Plošča ima tudi 14 digitalnih priključkov, ki se lahko definirajo kot vhodi ali izhodi. Skozi vsakega izmed njih lahko teče tok, največ 40 mA. Vsak digitalni pin ima vgrajen notranji »pull-up« upor od 20 do 50 k Ω . Digitalna pina 0 (RX) in 1 (TX) služita lahko tudi za serijsko komunikacijo z računalnikom. Pini 3, 5, 6, 9, 10 in 11 omogočajo tudi pulzno-širinsko modulacijo.

Na plošči so štiri LED-diode. Dve prikazujeta RX- in TX-komunikacijo, ena LED-dioda je povezana na pin 13 in je vključena, ko je pin 13 na stanju »HIGH«. Seveda plošča poleg pinov z napetostjo 5 V oz. 3 V in GND vsebuje še druge pine, npr. pin »AREF«, nanj priključimo referenčno napetost za analogne vhode in druge.

(Vir: <https://www.arduino.cc/en/Main/ArduinoBoardNano>)

4 Giroskop

Giroskop ali tudi žiroskop je naprava, ki za svoje delovanje porablja načelo ohranitve vrtilne količine. Je v bistvu simetrična vrtavka, ki se giblje okrog stacionarne točke. Trenje pri vsem tem zanemarimo, prav tako zanemarimo vse zunanje sile. Vrteči se giroskop se upira vsem spremembam smeri gibanja. To napravo je leta 1852 izumil francoski fizik in astronom Jean Bernard.



Slika 10: Giroskop

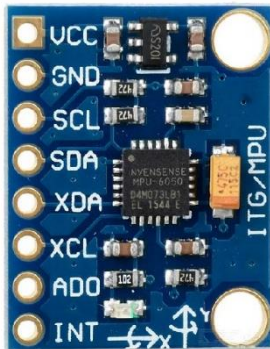
Pri girokopu se med drugim pojavi tudi precesija, to je pojav, ko os nekega vrtečega se telesa orisuje plašč stožca.

Poznamo tudi tako imenovane girokompase, ki dandanes nadomeščajo navadne, magnetne kompase v letalstvu in vesoljskih plovilih. Giroskop se uporablja še za stabilizacijo, na primer pri kolesu, železnici na enem tiru, nameritvi letalske bombe, kot zaloga vrtilne količine in za ohranjanje in prenos energije v nekaterih strojih (vztrajnik v motorju) idr.

(Vir: <https://sl.wikipedia.org/wiki/%C5%BDiroskop>)

4.1. Elektronski giroskop in merilec pospeškov MPU-6050

Giroskopi so danes nepogrešljiv del v elektronskih napravah, na primer tablicah in telefonih, saj brez njih marsikatera aplikacija (virtualna resničnost, virtualen sprehod po prostoru, 360° pogled ...) ne bi delovala.



Slika 11: MPU-6050

Tudi multikopter je potrebno orientirati in vseskozi odpravljati motnje, ki nastajajo pri letenju (veter, nesorazmerja v gradnji ohišja, izven-težiščna lega ...), zato je pri izdelavi plovil v zraku obvezno vključiti tudi giroskop. Nisva uporabila mehanskega, ampak elektronski MPU-6050.

MPU-60X0 je serija prvih sledilcev gibanja v šestih oseh na svetu, ki so giroskopi in merilci pospeškov v vseh treh smereh in obenem tudi digitalni procesorji gibanja.

Meri le 4 x 4 x 0.9 mm.

4.1.1. Karakteristike giroskopa

Giroskop ima območje merjenja ± 250 , ± 500 , ± 1000 in ± 2000 %/s. Zunanji sinhronizacijski signal, povezan na FSYNC pin, podpira sliko, video in GPS-sinhronizacijo. Poraba toka v obratovanju giroskopa je 3,6 mA, v načinu mirovanja pa 5 μ A.

4.1.2. Posebnosti lastnosti MPU-60X0 in posebnosti MPU-6050

Komunikacija z vsemi registri MPU-60X0 se izvaja z uporabo I²C na 400 kHz. Vsebuje tudi temperaturni senzor in oscilator na čipu z ± 1 % odstopanjem skozi celoten temperaturni obseg delovanja. MPU-60X0 ima 10000 g toleranco šokov in programljiv nizko-pasovni filter za giroskop in merilec pospeškov. Napajamo ga z napetostjo od 2,375 V do 3,46 V, poleg tega MPU-6050, ki sva ga uporabila, predvideva še VLOGIC referenčni pin, ki določa raven logike svojega I²C-vmesnika. VLOGIC napetost je lahko $1,8 \text{ V} \pm 5 \%$ ali napajalna napetost.

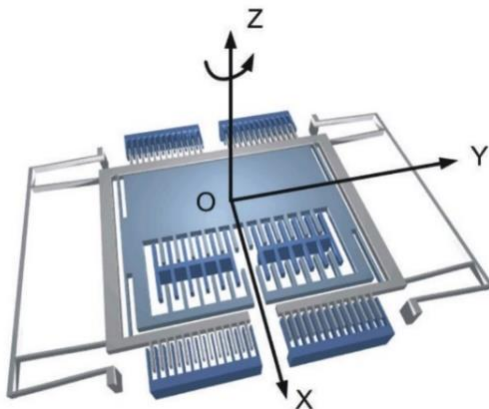
MPU-6050 podpira samo I²C-vmesnik, medtem, ko na MPU-6000 podpira tudi SPI.

MPU-60X0 I²C omogoča tudi branje podatkov iz zunanjih senzorjev, na primer magnetometra.

(Vir: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>)

4.1.3. Princip delovanja elektronskega giroskopa

Princip delovanja elektronskega giroskopa je zelo enostaven: v samem čipu je na vzmeteh ploskev, ki spreminja svoj položaj glede na zunanje tresljaje ali gibanje. S tem se spreminja kapacitivnost med samo ploskvijo in statičnim delom, na katerega so pritrjene vzmeti. Seveda se premična ploskev giblje v vse smeri (X, Y, Z). Ta način delovanja je lepo viden na fotografiji levo.



Slika 12: Princip delovanja elektronskega giroskopa

4.1.4. Pogonski motor A2212/13T

Elektromotor je zaradi prisotnosti magnetov v statorju in tuljav v rotorju ter elektronskega krmilja tuljav rotorja zelo koristen in učinkovit, saj ni iskrenja ter pretirane obrabe, lahko je v manjših dimenzijah enaka moč kot pri krtačnih motorjih, ki so večji. Je pa zaradi krmilja nekoliko dražji od krtačnih motorjev.

Št. celic: 2–3 Li-Poly, 6–10 NiCd/NiMH

Kv: 1000 vrt/min/V

Max. izkoristek: 80 %

Max. efektivni tok: 4–10 A (>75 %)

Brez obremenitve: 0.5 A pri 10 V

Upornost: 0.090 Ω

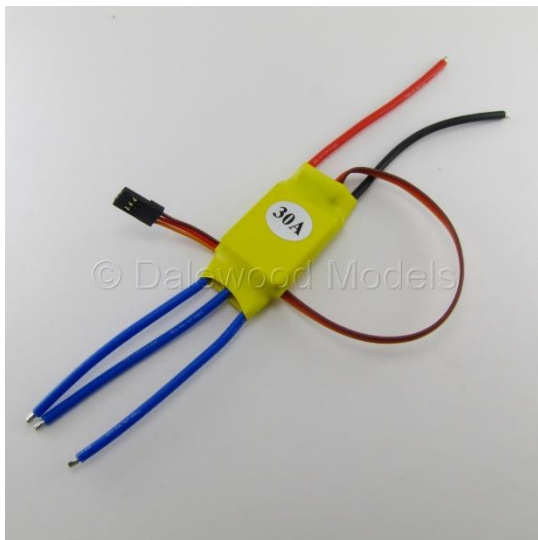
Max. tok: 13 A za 60 s
Max. moč: 150 W za 60 s
Teža: 52.7 g/1.86 oz
Velikost: 28 mm dia x 28 mm bell length
Premer gredi: 3.2 mm
Število polov: 14



Slika 13 motor A2212

4.1.5. 30A brezkrtačni ESC

Izhod: neprekinjen tok 30 A, največji tok 40 A do 10 s;
vhodna napetost: 2–4 celic li-poly ali 5–12 celic NiCd/NIMh.



BEC: 2 A/5 V (linearni način),
max. hitrost: 210000 vrt/min za 2 polni BLM,
70000 za 6 polnih BLM vrt/min, 35000 vrt/min za
12 polnih BLM (BLM: brezkrtačni motor).

ESC potrebuje za svoje delovanje signal, ki ga prejema v obliki pulzno-širinske modulacije. BEC priključek je napajanje (+5 V/2 A), zraven napajanja pa je tudi žica, po kateri potuje signal v regulator hitrosti iz sprejemnika, v najinem primeru iz Arduina. Signal je velikosti 3 V, perioda je pri minimalnem signalu, kjer se motor ne vrti, enaka 14,82 ms, širina pulza je enaka 1,1 ms.

je širina pulza 1,9 ms.

Pri največji hitrosti, kjer je perioda dolga 22,1 ms,

Slika 14: ESC 30A

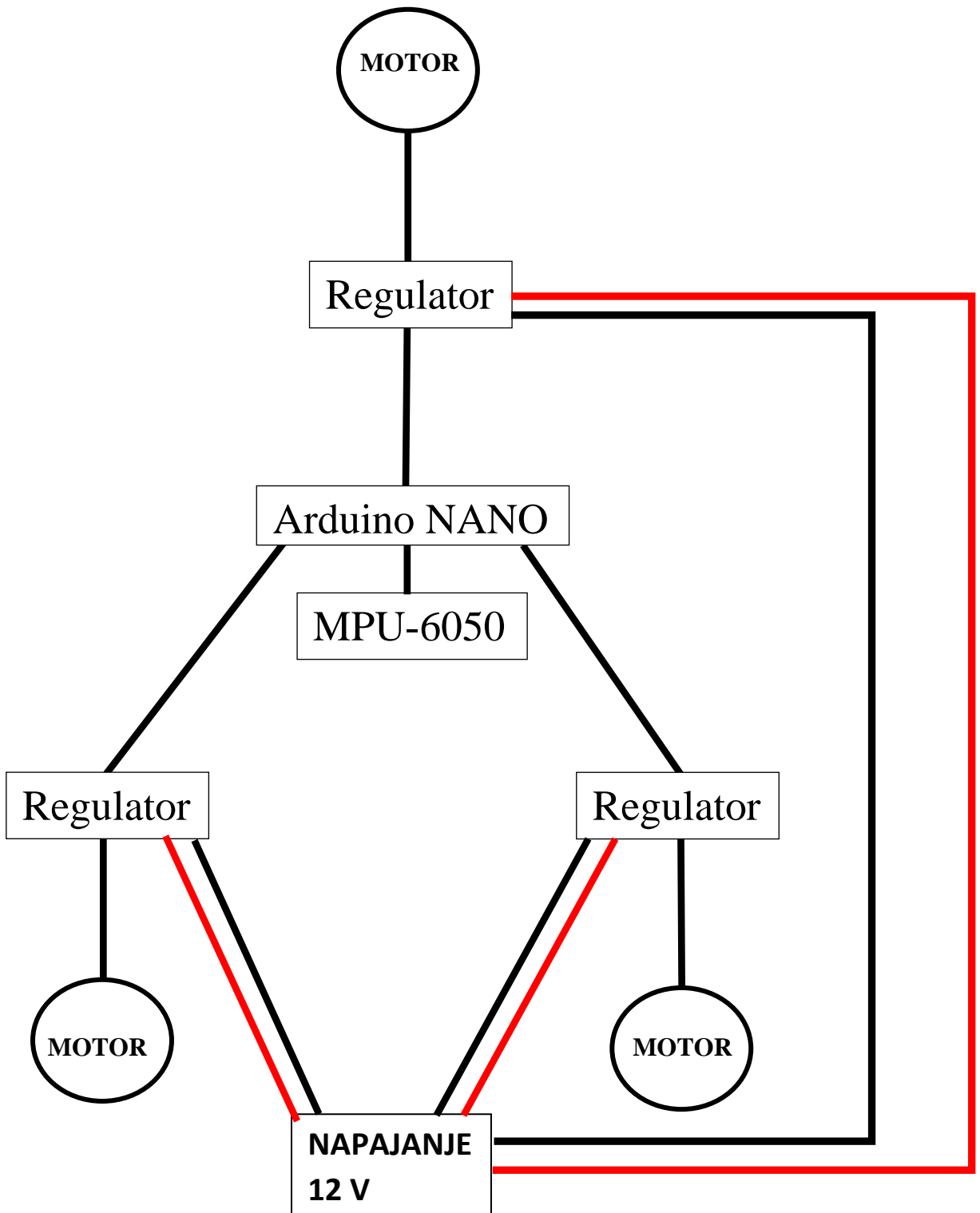
4.1.6. 1045 Elise

Premer gredi: 6,0 mm
Nastavki različnih dimenzij lukenj (3 mm, 3.2 mm, 4 mm, 5 mm, 6 mm, 6.35 mm, 7.95 mm)
Debelina centra: 9.7 mm
Teža: 15 g/par
Priporočena medosna razdalja: 550 mm–700 mm



Slika 15: Elise 1045

4.2. Preprosta shema povezovanja komponent trikopterja



4.3. Program

Za programiranje sva uporabila program Arduino IDE. Programski jezik je C++.

Najprej sva preučila program za izpisovanje surovih vrednosti na LCD-zaslonu. Ta je že narejen in ga lahko najdemo med programi v IDE-razvojnem okolju. Program je zelo pomemben, saj je osnova za uporabo surovih vrednosti in nadaljnje programiranje.

Preučila sva dva programa, in sicer vsak enega. Program sva delno prenesla iz interneta, preučila njegovo delovanje in ga dopolnila po svoje za potrebe izpisovanja na LCD.

```
171 void write_LCD() {  
172     //To get a 250Hz program loop (4us) it's only possible to write one ch  
173     //Writing multiple characters is taking to much time  
174     if(lcd_loop_counter == 14) lcd_loop_counter = 0;  
175     lcd_loop_counter ++;  
176     if(lcd_loop_counter == 1) {  
177         angle_pitch_buffer = angle_pitch_output * 10;  
178         lcd.setCursor(6,0);  
179     }  
180     if(lcd_loop_counter == 2) {  
181         if(angle_pitch_buffer < 0) lcd.print("-");  
182         else lcd.print("+");  
183     }  
184     if(lcd_loop_counter == 3) lcd.print(abs(angle_pitch_buffer)/1000);  
185     if(lcd_loop_counter == 4) lcd.print((abs(angle_pitch_buffer)/100)%10);  
186     if(lcd_loop_counter == 5) lcd.print((abs(angle_pitch_buffer)/10)%10);  
187     if(lcd_loop_counter == 6) lcd.print(".");  
188     if(lcd_loop_counter == 7) lcd.print(abs(angle_pitch_buffer)%10);  
189  
190     if(lcd_loop_counter == 8) {  
191         angle_roll_buffer = angle_roll_output * 10;  
192         lcd.setCursor(6,1);  
193     }
```

Slika 16: Prvi program

Srednja šola za kemijo, elektrotehniko in računalništvo IZDELAVA MULTIKOPTERJA

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

//MPU6050 accelgyro;

#include<Wire.h>
const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t Tmp,GyX,GyY,GyZ;
void setup(){
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Dober dan!");
  delay(5000);
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  Serial.begin(9600);
  lcd.clear();
  //accelgyro.initialize();
}
void loop(){
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
  /*Serial.print("AcX = "); Serial.print(AcX);
  Serial.print(" | AcY = "); Serial.print(AcY);
  Serial.print(" | AcZ = "); Serial.print(AcZ);*/
  Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53); //equation for temperature in degrees C from datasheet
  Serial.print(" | GyX = "); Serial.print(GyX);
  Serial.print(" | GyY = "); Serial.print(GyY);
  Serial.print(" | GyZ = "); Serial.println(GyZ);
```

Slika 17: Prvi del drugega programa – branje vrednosti giroskopa

```
lcd.setCursor(0,0);
lcd.print("X= ");
lcd.setCursor(2,0);
lcd.print(GyX);
lcd.print(" ");
delay(50);

  lcd.setCursor(8,0);
  lcd.print("Y= ");
  lcd.setCursor(11,0);
  lcd.print(GyY);
  delay(50);
  lcd.print(" ");

  lcd.setCursor(0,1);
  lcd.print("Z= ");
  lcd.setCursor(3,1);
  lcd.print(GyZ);
  delay(50);
  lcd.print(" ");

delay(333);
```

Slika 18: Drugi del drugega programa - izpisovanje na LCD zaslonu

Srednja šola za kemijo, elektrotehniko in računalništvo IZDELAVA MULTIKOPTERJA

```
int UPRAVLJANJE= analogRead(A2);
int HITROST=730;

HITROST=HITROST+UPRAVLJANJE;

float VREDNOST1 = map(GyX, -32000, 32000, 0.0, 500.0);
float VREDNOST2 = map(GyX, -32000, 32000, 500.0, 0.0);

#define STEVILO_MERJENJ 1000
//myservo2.writeMicroseconds(1000);
float ACCX = 0;
for(int i = 0; i < STEVILO_MERJENJ; i++){
  ACCX = ACCX + AcX;
}
float POVPRECJE = ACCX/STEVILO_MERJENJ;
//Serial.print(" POVPRECJE: "); Serial.println(POVPRECJE);

#define STEVILO_MERJENJ2 1000
float ACCZ = 0;
for(int i = 0; i < STEVILO_MERJENJ2; i++){
  ACCZ = ACCZ + AcZ;
}
float POVPRECJEZ = ACCZ/STEVILO_MERJENJ2;
//Serial.print("          ");
//Serial.print(" POVPRECJE-Z: "); //Serial.print("          ");
//Serial.println(POVPRECJEZ);

#define STEVILO_MERJENJ3 1000
float ACCY = 0;
for(int i = 0; i < STEVILO_MERJENJ3; i++){
  ACCY = ACCY + AcY;
```

V končnem programu sva najprej izvedla branje izhodov giroskopa in pospeškometra. Potem sva v program uvrstila spremenljivko »UPRAVLJANJE«, kamor sva shranjevala vrednost, prebrano iz potenciometra na vhodu A2, ki služi za upravljanje hitrosti motorjev, kateri se potem prišteje še izbrana vrednost osnovne hitrosti.

»VREDNOST 1« in »VREDNOST 2« sta spremenljivki, ki sva jih »mapirala«, saj surove vrednosti iz giroskopa in pospeškometra znašajo od –32000 do 32000 in so zelo nestabilne, zato niso najbolj uporabne za nadaljnjo uporabo, z »mapiranjem« pa se težava reši. Omogoča namreč, da zmanjšamo obseg števil, s katerimi upravljamo, na najbolj uporabnega.

Za stabilizacijo hitrih nagibov sva uporabila giroskop, za stabilizacijo počasnih odstopanj od ravnovesne osi pa pospeškometer. Izmed 1000 vrednosti pri vsakem sva določila povprečno, zato da bi odpravila motnje, ki nastajajo zaradi tresenja ohišja trikoterja.

```
}
float POVPRECJEY = ACCY/STEVILO_MERJENJ3;
//Serial.print("          ");
//Serial.print(" POVPRECJE-Y: "); //Serial.print("          ");
//Serial.println(POVPRECJEY);
```

```
float POVPRECJEAcY = map(AcY, -16500.0, 16500.0, -500.0, 500.0);
if(POVPRECJEAcY>4) {
myservo1.writeMicroseconds(HITROST+POVPRECJEAcY*0.5+VREDNOST1);
}
```

Slika 19: Prvi del končnega programa

```
}
...
}
```

Slika 20: Drugi del končnega programa

Nato sva »mapirala« še povprečje pospeškometra in dobljeno vrednost uporabila za krmiljenje motorja 1 v enačbi: »HITROST« (vrednost iz potenciometra in vnaprej izbrane vrednosti hitrosti) + »POVPREČJEAcY« (povprečna vrednost pospeškometra) *0,5 (faktor za določanje natančnosti) + »VREDNOST 1« (»mapirana« vrednost giroskopa). Enačba za

drugi motor je zelo podobna, le da smo od hitrosti odšteli vse ostalo.

Srednja šola za kemijo, elektrotehniko in računalništvo
IZDELAVA MULTIKOPTERJA



Slika 21: Izpisovanje stopinj nagiba na LCD-zaslону

5. Razprava

Na začetku sva si zadala hipoteze:

- Dron sam drži ravnotežje v eni osi kljub zunanjim motnjam.

To hipotezo lahko potrdiva, saj sva to dosegla z zgornjim programom. Motorja se lepo uravnavata v eni osi, kljub temu ali so motnje od zunaj sunkovite ali pa gre le za počasen odmik od osi 0.

- Trikopter je mogoče voziti v smeri naprej – nazaj.

To hipotezo lahko delno potrdiva saj predvidevava, da je to, na podlagi dosedanjih dognanj, zelo mogoče. Najina raziskava ni prišla tako daleč, da bi to hipotezo lahko povsem potrdila.

- Trikopter je mogoče v prostoru voditi okoli vseh treh osi.

Te hipoteze ne moreva ne potrditi ne ovreči, saj je nadzor trikopterja v vseh treh oseh zelo težko doseči in tudi najina raziskava ni prišla do te stopnje.

6. Zaključek

Z delovanjem stabilizacije motorjev sva zadovoljna. Trikopter bova zagotovo nadgrajevala še naprej z dodajanjem tretjega motorja, ki bo služil za premikanje v smeri naprej – nazaj.

Pri izdelavi seminarske naloge sva se še bolje spoznala z delovanjem modula Arduino in njegovim programiranjem ter z delovanjem in vesplošno uporabnostjo MPU – 6050. Razvila sva tudi večine timskega dela in nadgradila svoje splošno znanje.

Seznam virov

<http://www.ebay.com/itm/A2212-1000KV-Brushless-Motor-With-30A-ESC-1045-Propeller-for-DJI-F450-550-NEW-/261802462189?hash=item3cf4a48bed:g:UYkAAOSwxH1UI~Mb> (22.2016)

<https://oscarliang.com/quadcopter-motor-propeller/> (20.10.2016)

<http://playground.arduino.cc/Main/MPU-6050?action=sourceblock&num=1> (13.10.2016)

<http://electronics.stackexchange.com/questions/142037/calculating-angles-from-mpu6050> (11.1.2017)

<http://electronics.stackexchange.com/questions/39024/how-do-i-get-gyro-sensor-data-13g4200d-into-degrees-sec> (11.1.2017)

<https://sl.wikipedia.org/wiki/Arduino> (2.2.2017)

<http://www.nkcelectronics.com/assets/images/atmega328p.jpg> (2.2.2017)

http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf (2.2.2017)

<https://www.arduino.cc/en/Main/Products> (2.2.2017)

http://cdn-reichert.de/bilder/web/xxl_ws/A300/ARDUINO_NANO_03.png (9.2.2017)

<https://www.arduino.cc/en/Main/ArduinoBoardNano> (9.2.2017)

<https://sl.wikipedia.org/wiki/%C5%BDiroskop> (26.2.2017)

<https://upload.wikimedia.org/wikipedia/commons/6/63/Giroskop.png> (26.2.2017)

<https://www.makerlab-electronics.com/wp-content/uploads/2015/05/mpu-6050-1.jpg> (26.2.2017)

<http://cas1.elis.ugent.be/cas/pic/GyroPrinciple.jpg> (26.2.2017)

<https://govori.se/zanimivosti/droni-brezpilotne-naprave-ki-spreminjajo-zivljenja/> (2.3.2017)

https://www.google.si/search?q=A2212&source=lnms&tbn=isch&sa=X&ved=0ahUKEwiHqvWPv8DSAhWCFZoKHT-LAzIQ_AUICCGb&biw=1536&bih=710#imgdii=srOOIBbDG28wPM:&imgrc=MxnN5jPliUVFcM: (2.3.2017)

https://www.google.si/search?q=30A+ESC&source=lnms&tbn=isch&sa=X&ved=0ahUKEwi7lvEv8DSAhVkJpoKHTDOATgQ_AUICCGb&biw=1536&bih=710#imgrc=R112k_WOG7HeyM: (2.3.2017)

https://www.google.si/search?q=1045+Propeller&source=lnms&tbn=isch&sa=X&ved=0ahUKewjel-HQv8DSAhWhCJoKHx67CGQQ_AUICCGb&biw=1536&bih=710#imgrc=EdP3ti8tKC7gZM: (2.3.2017)

Srednja šola za kemijo, elektrotehniko in računalništvo
IZDELAVA MULTIKOPTERJA

<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
(28.2.2017)

http://x-inferno.com/wp-content/uploads/2014/01/IMG_8963-685x456.jpg (11.3.2017)

IZJAVA*

Mentor (-ica) GREGOR KRAMEB, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom IZDELAVA MULTIKOPTERJA, katere avtorji (-ice) so MATEJ ŽABERL, GASPER TRUPEJ :

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo (-ičino) dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 13.3.2017



Podpis mentorja(-ice)

Podpis odgovorne osebe

*
POJASNILO

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.