

UNIVERZA V LJUBLJANI

Fakulteta za strojništvo

Razvoj servomotorja in grafičnega vmesnika za nadzor

Zaključna naloga Univerzitetnega študijskega programa I. stopnje
Strojništvo - Razvojno raziskovalni program

Vid Tilen Ratajec

Ljubljana, avgust 2023

UNIVERZA V LJUBLJANI

Fakulteta za strojništvo

Razvoj servomotorja in grafičnega vmesnika za nadzor

Zaključna naloga Univerzitetnega študijskega programa I. stopnje
Strojništvo - Razvojno raziskovalni program

Vid Tilen Ratajec

Mentor: doc. dr. Rok Vrabič

Somentor: doc. dr. Gašper Škulj

Ljubljana, avgust 2023



PRIJAVA TEME ZAKLJUČNEGA DELA

Študent: Vid Tilen Ratajec

Študijski program: Univerzitetni študijski program prve stopnje Strojništvo - Razvojno raziskovalni program

Številka teme zaključnega dela: UN I/1846

Naslov teme zaključnega dela v slovenskem jeziku: Razvoj servomotorja in grafičnega vmesnika za nadzor

Naslov teme zaključnega dela v angleškem jeziku: Developing a servomotor and a graphical interface for control

Dispozicija zaključnega dela:

Servomotorji so pomemben del mnogih mehatronskih sistemov pri katerih potrebujemo zaprozančno krmiljenje. Sestavljeni so iz aktuatorja, ki je običajno elektromotor, absolutnega pozicijskega kodirnika in krmilnika. Cilj zaključne naloge je izdelati servomotor z uporabo magnetnega kodirnika in mikrokrmilnika ter razviti računalniški program za nadzor prek dvosmerne serijske povezave.

V zaključni nalogi preglejte osnove delovanja servomotorjev. Izberite komponente in opišite teoretično ozadje njihovega delovanja. Izdelajte servomotor na osnovi koračnega motorja z gonilnikom, magnetnega kodirnika in mikrokrmilnika. Za programiranje mikrokrmilnika uporabite okolje Arduino. Izdelajte računalniški program za krmiljenje v programskejem jeziku Python, ki bo z uporabniškim vmesnikom narejenim s knjižnico PyQt6, omogočal krmiljenje servomotorja prek serijske povezave, omogočene z uporabo knjižnice pySerial. Program mora omogočati inkrementalni pomik, pomik na absolutno pozicijo in zaznavo napak. Analizirano naj bo delovanje izdelanega servomotorja.

Mentor: doc. dr. Rok Vrabič

Somentor: doc. dr. Gašper Škulj

Datum odobrene teme: 28.06.2023



Opomba: Zaključno delo je treba oddati v jezikovno in terminološko pravilnem slovenskem jeziku. Rok za oddajo zaključnega dela v informacijski sistem VIS je šest mesecev od odobrene teme.

Izvleček

UDK 681.5:621.3.077.2:004.4(043.2)

Tek. štev.: UN I/1846

Razvoj servomotorja in grafičnega vmesnika za nadzor

Vid Tilen Ratajec

Ključne besede: krmiljenje
 servomotorji
 mikrokrmilniki
 koračni motorji
 kodirniki
 gonilnik motorjev

Servomotor je osnoven del mehatronskih sistemov pri katerih potrebujemo zaprto zančno krmiljenje. Sestavljen je iz aktuatorja, ki je običajno električni motor, absolutnega pozicijskega kodirnika in krmilnika. Izdelan servomotor je sestavljen iz koračnega motorja, magnetnega kodirnika, gonilnika motorja ter mikrokrmilnika. Narejen je bil program z grafičnim vmesnikom, ki preko serijske povezave nadzira vrtenje servomotorja. Testirana je bila ponovljivost rotacij, čas branja kota ter hitrost.

Abstract

UDC 681.5:621.3.077.2:004.4(043.2)

No.: UN I/1846

Developing a servomotor and a graphical interface for control

Vid Tilen Ratajec

Key words: control
 servomotors
 microcontrollers
 stepper motors
 encoders
 stepper motor drivers

The servomotor is a basic part of mechatronic systems that require closed-loop control. It consists of an actuator, which is usually an electric motor, an absolute position encoder and a controller. The constructed servomotor consists of a stepper motor, a magnetic encoder, a motor driver, and a microcontroller. A program with a graphical interface has been developed which controls the rotation of the servomotor via serial communication. Servomotor was tested for repeatability of rotations, the angle reading time and speed.

Kazalo

Kazalo slik	vii
Kazalo preglednic	viii
1 Uvod	1
1.1 Servomotor	1
1.2 Cilji naloge	1
2 Teoretične osnove in pregled literature	2
2.1 Hibridni bipolarni koračni motor	2
2.1.1 Standard NEMA	3
2.1.2 Krmiljenje koračnega motorja	3
2.1.3 Napaka koraka	5
2.2 Rotacijski kodirnik	5
2.2.1 Inkrementalni kodirnik	5
2.2.2 Absolutni kodirnik	6
2.2.3 Ločljivost absolutnega kodirnika	6
2.2.4 Diferencialni magnetni absolutni kodirnik	6
2.2.5 Napake	7
2.3 Mikrokrmilnik RP2040	9
2.3.1 UART Serijska komunikacija	9
2.3.2 I^2C	10
2.4 Zaprtozančno PID krmiljenje/vodenje	10
2.4.1 Proporcionalni del/odziv	10
2.4.2 Integralni del/odziv	11
2.4.3 Diferencialni del/odziv	11
2.4.4 Nastavljanje parametrov	11

3 Metodologija raziskave	12
3.1 Komponente	12
3.1.1 Mikrokrmlnik	12
3.1.2 Gonilnik motorjev A4988	13
3.1.3 Senzor rotacije AMS AS5600	14
3.1.4 Elektronsko vezje in shema	15
3.2 Program	16
3.2.1 Krmiljenje koračnega motorja	16
3.2.2 Komunikacija s servomotorjem	19
3.2.3 Izdelava grafičnega vmesnika	19
4 Rezultati	21
4.1 Izdelani servomotor	21
4.2 Analiza delovanja	21
4.2.1 Ponovljivost rotacije	21
4.2.2 Čas branja senzorja	22
4.2.3 Hitrost vrtenja servomotorja	22
5 Zaključki	24
Literatura	25

Kazalo slik

Slika 2.1:	Komponente motorja	2
Slika 2.2:	Prikaz polov diametrično magnetiziranega magneta	6
Slika 2.3:	Shema senzorjev v čipu [6]	7
Slika 2.4:	Nelinearnost magnetnega kodirnika [6]	7
Slika 2.5:	Neporavnanost sistema	8
Slika 2.6:	Linearno območje [6]	9
Slika 2.7:	Blokovna shema PID	10
Slika 3.1:	Sestavna risba servomotorja	12
Slika 3.2:	Stanje senzorja glede na rotacijo magneta [5]	14
Slika 3.3:	Shema vezja	15
Slika 3.4:	Prikaz spodnje in zgornje strani vezja	15
Slika 3.5:	Diagrama poteka osnovnega programa ter preverjanja delovanja	16
Slika 3.6:	Diagram poteka branja in izvedbe ukaza	17
Slika 3.7:	Diagram poteka absolutnega premika	18
Slika 3.8:	Diagram poteka inkrementalnega premika	18
Slika 3.9:	Format ukaza	19
Slika 3.10:	Grafični vmesnik za upravljanje servomotorja	19
Slika 4.1:	Servomotor (a) in prototipno vezje (b)	21
Slika 4.2:	Graf hitrosti vrtenja pri različnih številih branj	23
Slika 4.3:	Graf $t_{medkoraki}$ ter delež branja senzorja	23

Kazalo preglednic

Preglednica 2.1:	NEMA oznake in pomen črk v imenu koračnega motorja	3
Preglednica 2.2:	Stanje faz med korakom - valovna metoda	4
Preglednica 2.3:	Stanje faz med korakom - metoda dveh prižganih faz	4
Preglednica 2.4:	Stanje faz med polkorakom	4
Preglednica 2.5:	Nelinearnost glede na odmik od sredine čipa	8
Preglednica 3.1:	Logična preglednica mikrokorakov	13
Preglednica 3.2:	Specifikacije kodirnika AS5600	14
Preglednica 3.3:	Deli ukaza	19
Preglednica 4.1:	Rezultati meritev ponovljivosti	22

1 Uvod

1.1 Servomotor

Servomotor je rotacijski ali linearni aktuator s povratno zanko za krmiljenje zasuka oziroma pozicije in hitrosti. Namenjen je hitremu in natančnemu obračanju mehanskih sklopov. V primerjavi z odprto zančnimi sistemi je krmiljenje rotacije sistema bolj zahtevno, vendar pa obremenitev in motnje ne vplivajo na končno pozicijo sistema. Servomotor je sestavljen iz krmilnika, aktuatorja ter kodirnika. Krmilnik glede na povratni signal krmili motor. Povratni signal je pridobljen z branjem kodirnika, kodirnik je lahko absolutni ali inkrementalni. Aktuator je lahko linearni ali rotacijski električni motor. Pri izdelanem servmotorju bo uporabljen koračni motor, absolutni magnetni kodirnik ter mikrokrmilnik.

1.2 Cilji naloge

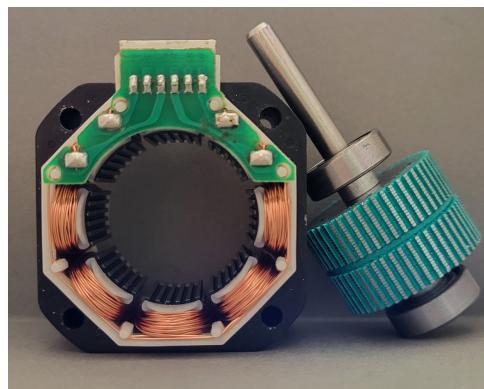
Cilj naloge je izdelati servomotor. Ta bo izdelan z uporabo magnetnega kodirnika AMS Osram AS5600, mikrokontrolerja Raspberry Pi RP2040, gonilnika koračnih motorjev Allegro Microsystems A4988 ter koračnega motorja velikosti NEMA17. Ustvariti program za upravljanje in krmiljenje motorja, ki z mikorkrmilnikom komunicira preko serijske povezave. Izdelan program za upravljanje servomotorja ima grafični vmesnik narejen v PyQt.

2 Teoretične osnove in pregled literature

2.1 Hibridni bipolarni koračni motor

Koračni motor (*ang. Stepper motor*) je brezkrtačni DC motor. Ena rotacija motorja je enakomerno razdeljena na *korake* oziroma inkremente. Motor je krmiljen z namenskim vezjem za krmiljenje, ki ustvari pulze, en pulz motor premakne za en korak. Kot rotacije se določi s štetjem in smerjo korakov, tovrstni odprto zančni sistem za določitev pozicije ne potrebuje senzorja.

Hibridni bipolarni koračni motor je sestavljen iz dveh glavnih komponent. Prikazani sta na sliki 2.1.



Slika 2.1: Komponente motorja

Rotor je sestavljen iz gredi in permanentnega aksialnega magneta. Ta je sestavljen iz permanentnega magneta in dveh zobatih obročev, ki delujeta kot severni in južni pol. Zobata obroča sta zamknjena za $\frac{1}{2}$ modula. Stator je sestavljen iz navitij in nazobčenih polov. Število zob na rotorju ne sme biti manjše ali enako številu zob na statorju. Pri motorju z 200 koraki je število zob na rotorju enako 50, število zob na statorju pa 48. Navitja žic na polih statorja, ki so povezane skupaj imenujemo faza. Število polov faz in zob na rotorju določita velikost koraka. Kot enega koraka θ_K se

izračuna po enačbi 2.1

$$\theta_K = \frac{360}{2n \cdot Z_R} \quad (2.1)$$

Simbol n predstavlja število faz, število zob pa Z_R .

Koračne motorje se krmili z namenskim vezji. Ta ustvarijo električne pulze in omejujejo tok. Koračni motorji se pri konstantno zaprtemu tokokrogu ne vrtijo, temveč obdržijo pozicijo - ustvarjajo upor oziroma navor. Ob pulzu toka skozi navitja se rotor premakne za en korak oziroma del koraka. Število korakov v rotaciji je odvisno od kota koraka (2.1). Število korakov N določimo po enačbi 2.2.

$$N = \frac{360}{\theta_K} \quad (2.2)$$

2.1.1 Standard NEMA

Velikost in način montaže koračnega motorja sledi standardu ameriške organizacije proizvajalcev električne opreme NEMA (*National Electrical Manufacturers Association*). Standard podaja dimenzije motorja pomnožene z deset (inch), te se pretvorijo v SI enote. Lastnosti motorja se poda kot:

NEMA DDMMLLL-CCCIIVVV

Pomen simbolov je podan v preglednici 2.1.

Preglednica 2.1: NEMA oznake in pomen črk v imenu koračnega motorja

Oznaka	Pomen
DD	širina motorja v inch $\times 10$
MM	Tip montaže
LLL	Dolžina v inch $\times 10$
CCC	Tok faze $\times 10$
I	Razred izolacije
VVV	Napetost faze $\times 10$
W	Tip navitja

2.1.2 Krmiljenje koračnega motorja

Korak - rotor naredi en korak, ko se spremeni katera faza je prižgana. Rotor se poravnava s prižgano fazo in tam obstane - valovana metoda (*ang. Wave form*). S prižiganjem in ugašanjem faz se rotor vrati v korakih. Za povečanje navora lahko uporabimo metodo dveh prižganih faz (*ang. 2 Phase on*). Stanja faz za obe metodi sta predstavljeni v logičnih preglednicah 2.2 in 2.3 za metodo dveh prižganih faz. Velike tiskane črtke

Preglednica 2.2: Stanje faz med korakom - valovna metoda

Faza	1	2	3	4
A	1	0	0	0
B	0	1	0	0
\bar{A}	0	0	1	0
\bar{B}	0	0	0	1

Preglednica 2.3: Stanje faz med korakom - metoda dveh prižganih faz

Faza	1	2	3	4
A	1	0	0	1
B	1	1	0	0
\bar{A}	0	1	1	0
\bar{B}	0	0	1	1

predstavljajo fazo, števke pa zaporedje.

Pol korak - rotor se zavrti za polovico koraka ($\frac{\theta_K}{2}$). S tem povečamo ločljivost motorja, zmanjšajo se vibracije in navor. Pri tem načinu delovanja sta prižgani dve fazi hkrati potem pa samo ena. Zaporedje prižiganja faz je predstavljeno v logični preglednici 2.4.

Preglednica 2.4: Stanje faz med polkorakom

Faza	1	2	3	4	5	6	7	8
A	1	1	0	0	0	0	0	1
B	0	1	1	1	0	0	0	0
\bar{A}	0	0	0	1	1	1	0	0
\bar{B}	0	0	0	0	0	1	1	1

Mikrokorak - rotor se zavrti samo za del koraka. Najmanjša velikost mikrokoraka je odvisna od gonilnika motorja. Velikost mikrokoraka je odvisna od razmerja toka v fazah, natančnost kontroliranja toka torej določi, kako majhen korak lahko naredimo. Z manjšanjem toka se manjša tudi navor motorja [2]. Z manjšanjem velikosti koraka se manjšajo tudi vibracije motorja. Navor, ki ga lahko proizvede motor pri mikrokoraku, se izračuna po enačbi:

$$M_{Mikorkorak} = M_{Korak} \cdot \sin\left(\frac{90}{N_{MK}}\right) \quad (2.3)$$

M_{Korak} predstavlja navor enega koraka, N_{MK} pa število mikrokorakov v koraku. Z številom mikrokorakov N_{MK} je določena velikost mikrokoraka. Ta se izračuna po enačbi 2.4.

$$\theta_{MK} = \frac{\theta_K}{N_{MK}} \quad (2.4)$$

Po enačbi 2.5 se izračuna kot rotacije. I_a in I_b predstavljata tok skozi fazi A in B, θ_{MK} pa velikost mikrokoraka.

$$\theta_{MK} = \arctan\left(\frac{I_b}{I_a}\right) \quad (2.5)$$

Da bi zagotovili enakomerno porazdelitev mikrokorakov v koraku, morata biti oba signala sinusne oblike z zamikom 90° . Pri uporabi mikrokorakov ne pride do poravnave zob rotorja in statorja, natančnost mikrokorakov je veliko bolj odvisna od razmerja amplitud in oblike toka skozi fazi kot od konstrukcije motorja.

2.1.3 Napaka koraka

Večina koračnih motorjev ne vsebuje kodirnika za zaznavanje rotacije zato krmilni sistem ne dobiva povratnih informacij o poziciji motorja. Koračni motor po opravljenem koraku oscilira okoli želene pozicije ali pa se pri vsakem koraku ne obrne za točno isti kot. Oscilacije so kratkotrajne, čas trajanja je odvisen od konstrukcije motorja. Napaka koraka se meri v procentih enega koraka. Končna napaka ni seštevek napak več korakov, temveč je vedno enaka napaki enega. Napaka koraka je posledica mehanskih in električnih faktorjev. Mehanske napake so posledica natančnosti izdelave rotorja in statorja, koncentričnosti in neenakosti zračne reže med njima. Kako se koračni motor premika med koraki, je podano z linearnostjo motorja. Napaka motorja je lahko večja od mikrokoraka, ki ga ta lahko opravi. Nelinearnost motorja pa povzroči, da mikrokoraki niso enakih velikosti oziroma niso enakomerno porazdeljeni. V primeru prevelike obremenitve se korak ob pulzu ne izvede oziroma preskoči. Zaznavanje preskočenih korakov se izvede z goničnikom ali s kodirnikom.

2.2 Rotacijski kodirnik

Rotacijski kodirnik je meritnik, ki meri spremembo oziroma absolutno pozicijo zasuka. Kodirniki se delijo v dve podskupini - inkrementalne in absolutne. Razlikujejo se tudi glede na tehniko načina pridobivanja signala. Kodirniki so ponavadi sestavljeni iz dveh glavnih komponent - obroča, ki je pritrjen na rotirajoče se telo, ter senzorja.

2.2.1 Inkrementalni kodirnik

Inkrementanli rotacijski kodirnik je meritnik, ki meri spremembo pomika rotacije. Inkrementalni kodirnik pretvori rotacijsko gibanje v električni signal. Rotacija se lahko meri na več načinov - optično, kapacitivno, induktivno, magnetno ali kontaktno. Merilo za ločljivost inkrementalnega kodirnika je podana v številu inkrementov na en obrat. V primeru uporabe enega senzorja lahko določimo le absolutno vrednost pomika in hitrost, ne pa smeri. Za določanje smeri rotacije potrebujemo sva senzorja, ki sta zamaknjena za 1.5 inkrementa.

2.2.2 Absolutni kodirnik

Absolutni rotacijski kodirnik je merilnik, ki meri absolutno pozicijo osi. Za določitev pozicije ne uporablja štetja inkremenotv ampak absolutno vrednost merjene veličine. Pozicija osi se določi iz vrednosti izmerjene veličine in reference - vrednosti meritve pri kotu 0. V primeru izgube spomina naprave lahko pozicijo določimo iz ene meritve. Absolutni kodirniki so lahko optični ali magnetni. Oba tipa lahko za vsako pozicijo določita unikatno vrednost, iz katere se določi rotacija.

2.2.3 Ločljivost absolutnega kodirnika

Ločljivost rotacijskega kodirnika je definirana kot število stanj kodirnika v enem obratu oziroma najmanjšo spremembo pozicije, ki je še lahko zaznana. Podana je s številom bitov. Število pozicij izračunamo z enačbo 2.6

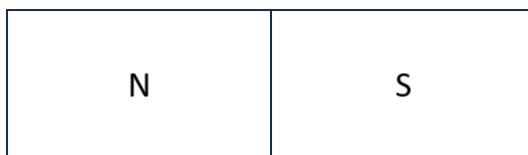
$$2^n \quad (2.6)$$

Najmanjša zaznana sprememba je podana v kotu:

$$\varphi = \frac{360}{2^n} \quad (2.7)$$

2.2.4 Diferencialni magnetni absolutni kodirnik

Absolutni magnetni kodirnik zaznava rotacijo magnetnega polja z uporabo Hallovega pojava. Hallov pojav povzroča nastanek razlike v napetosti na tokovodniku, ko je ta v magnetnem polju. Napetost inducirana v tokovodniku je linearno sorazmerna z magnetnim pretokom. Z merjenjem spremembe magnetnega pretoka lahko določimo rotacijo. Za zaznavanje rotacije mora biti magnet magnetiziran *diametrično* - v smeri premera ne višine. Pola diametrično magnetiziranega magneta sta prikazana na sliki 2.2.

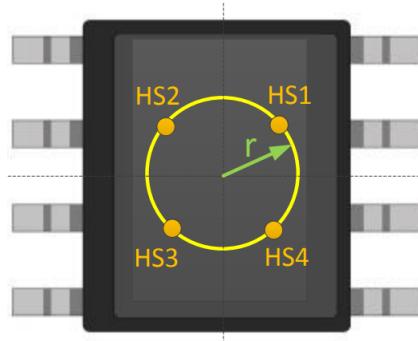


Slika 2.2: Prikaz polov diametrično magnetiziranega magneta

Diferencialni način zaznavanja rotacije, ki ga uporablja čip AMS AS5600, uporablja štiri Hallove senzorje. Ti so enakomerno razporejeni po krožnici, center mora biti čim bolj koncentričen s centrom magneta. Razporeditev senzorjev v čipu je prikazana na sliki 2.3.

Matematični model za izračun kota je popisan z enačbami 2.8 in 2.9

$$Signal_1 = U_{HS1} + U_{HS2} - U_{HS3} - U_{HS4} \quad (2.8)$$



Slika 2.3: Shema senzorjev v čipu [6]

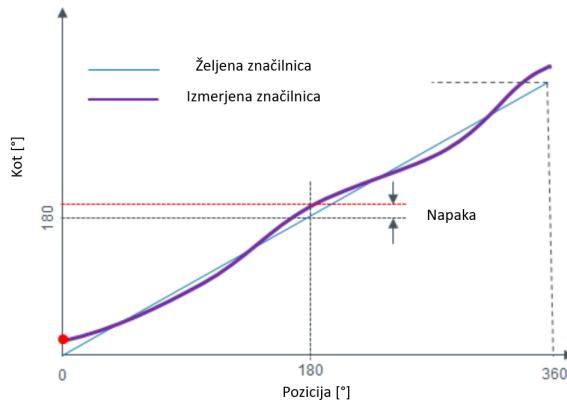
$$\text{Signal}_2 = U_{HS1} - U_{HS2} - U_{HS3} + U_{HS4} \quad (2.9)$$

Pri rotaciji magneta se na senzorjih ustvari sinusni signal. Te seštejemo oziroma odštejmo po formulah 2.8 in 2.9. Signala 1 in 2 sta zamaknjena za 90° in predstavljata *sinus* in *cosinus*. Kot rotacije α izračunamo po enačbi 2.10.

$$\alpha = \text{atan2}(\text{Signal}_1, \text{Signal}_2) \quad (2.10)$$

2.2.5 Napake

Nelinearnost je parameter pozicijskih senzorjev. Je razlika med teoretičnim in dejanskim kotom pri določeni rotaciji. Nelinearnost magnetnega absolutnega kodirnika je prikazana na sliki 2.4.



Slika 2.4: Nelinearnost magnetnega kodirnika [6]

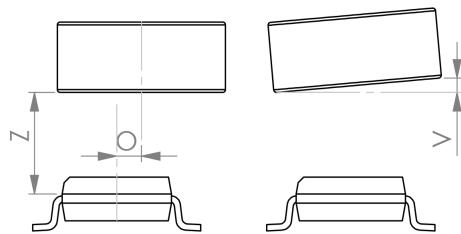
Napaka se izrazi z integralsko nelinearnostjo (enačba 2.11). Enačba je pridobljena iz vira [6]

$$NIL = \frac{\text{Odstopek}_{max} - \text{Odstopek}_{min}}{2} \quad (2.11)$$

Preglednica 2.5: Nelinearnost glede na odmik od sredine čipa

Odmik [μm]	NIL []
0	0.11
250	0.15
500	0.31
1000	0.71

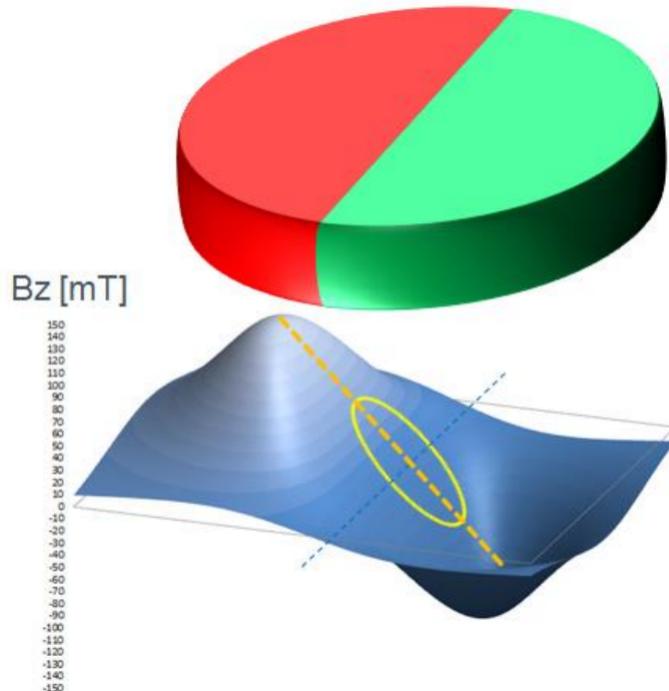
Napaka magnetnega kodirnika je odvisna predvsem od magnetnih značilnosti in poravnave sistema. Preglednica 2.5 prikazuje nelinearnost izračunano po 2.11 glede na neporavnanost sistema. Podatki so bili pridobljeni iz kataloga proizvajalca. Postavitev je prikazana na sliki 2.5. Črka O označuje odmik sredine magneta od sredine čipa, V označuje nagnjenost, Z pa višino magneta. Na napako vpliva tudi jakost magnetnega



Slika 2.5: Neporavnanost sistema

polja. Jakost polja se z večanjem razdalje med čipom in magnetom manjša, velikost signala pade in naraste šum v sistemu.

Merilni sistem je občutljiv tudi na vzporednost magneta in senzorja. Na sliki 2.6 je prikazan magneti pretok komponente polja B_Z , rumena elipsa prikazuje projekcijo kroga prikazanega na sliki 2.3 na to polje. Med lokalnima ekstremoma je območje linearно. Manjše napake v vzporednosti magneta in senzorjev tako v tem območju ne vplivajo zelo na meritev. Stopnja neporavnanosti je odvisna od velikosti magneta ter jakosti magnetnega polja.



Slika 2.6: Linearno območje [6]

2.3 Mikrokrmlnik RP2040

Mikrokrmlnik (*ang. microcontroller*) je majhen računalnik, ki vsebuje mikroprocesor, ram, spomin in I/O enote. Vse te komponente so združene v en čip. Mikrokrmlnike se integrira v vezja. Pomembne lastnosti mikrokrmlnika so odvisne od izbranega čipa. Mikrokrmlnik Raspberry PI RP2040 je ima dvo-jedrni mikroprocesor, 264 kB SRAM in 2MB flash spomina. Glavne specifikacije mikrokrmlnika RP2040:

- Dvo-jedrni ARM Cortex-M0+ procesor
- 264kB SRAM
- do 16MB flash spomina
- 30 GPIO pinov

2.3.1 UART Serijska komunikacija

Serijska komunikacija je proces prenašanja podatkov z biti, te se med napravami prenaša preko vodila (*ang. bus*). Mikrokrmlnik Raspberry Pi RP2040 za serijsko komunikacijo uporablja UART (*universal asynchronous receiver-transmitter*). To je naprava za asinhrono serijsko komunikacijo, pri kateri se lahko nastavi hitrost in format. Obe napravi morata imeti enako hitrost sprejemanja oziroma oddajanja (*baud rate*) in enak format zapisa podatkov. Okvir oziroma paket (*ang. frame*) je sestavljen iz začetnega in končnega bita, bitov, ki predstavlja podake ter paritetnega bita. (*ang. parity bit*), ki služi za preprosto zaznavo napak. Pogost format je 8N1, ki ima 8 bitov za prenos podatkov in en stop bit.

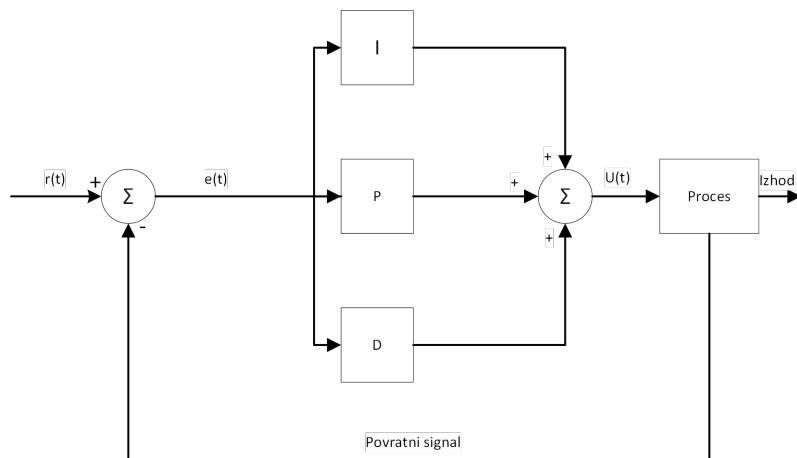
2.3.2 I^2C

I^2C (lahko zapišemo tudi kot: *IIC, I2C*) je komunikacijsko vodilo za komunikacijo med integriranimi vezji. Elektronske naprave so pogosto narejene iz mikrokrmlilnika in perifernih čipov oziroma integriranih vezij, za prenos podatkov med njimi se pogosto uporablja I^2C . Vsaka komponenta v sistemu ima status gospodarja (ang. *Master*) ali sužnja (ang. *Slave*) ter unikaten naslov. Komponenta s statusom gospodarja od drugih naprav zahteva podatke ali pa v njihov spomin piše. Več naprav je lahko povezanih ne eno I^2C vodilo, podatke lahko pošilja samo ena naprava naenkrat.

I^2C vodilo je sestavljeno iz vodil oziroma žic in dveh uporov. Eno vodilo prenaša signal ure (SCL), drugo pa serijsko podatke (SDA). Vse komponente I^2C vodila so povezane na ti dve vodili.

2.4 Zaprtovančno PID krmiljenje/vodenje

Zaprtovančno vodenje oziroma krmiljenje je proces, pri katerem se regulirana veličina primerja z želeno vrednostjo veličine. Glede na razliko med njima se ustvari signal, ki zmanjša razliko med želeno in trenutno veličino. Sistem sestavlja referenca, napaka, motnja, krmilnik, aktuator in merilnik. Proporcionalni-integralni-diferencialni (ang. *Proportional-Integral-Derivative*) krmilnik, krajše PID, je najpogosteji krmilni algoritem [8]. PID je sestavljen iz treh koeficientov, katerim dodamo uteži, končni krmilni signal pa je seštevek koeficientov pomnoženih z utežmi. Blokovna shema PID krmiljenja je prikazana na sliki 2.7.



Slika 2.7: Blokovna shema PID

2.4.1 Proporcionalni del/odziv

Proporcionalna komponenta odziva je enaka razlike trenutne vrednosti procesne spremenljivke ter želeni vrednosti. Imenujemo jo napaka (ang. *error*). Proporcionalni del množimo s parametrom K_P , ki določa hitrost sprememb sistema glede na napako -

večji kot je K_P hitreje se sistem odziva. Prevelik koeficient K_P povzroča oscilacije okoli želene vrednosti. Oscilacije povzročajo nestabilnost sistema. Napaka, ki ostane v sistemu kjer se uporablja samo proporcionalni del se imenuje napaka stacionarnega stanja, želeni vrednosti se približamo nikoli pa je ne doseženo.

2.4.2 Integralni del/odziv

Integralna komponenta odziva je enaka seštevku napake v času. Seštevek oziroma integral napak se bo konstantno večal, razen če je napaka enaka nič, v tem primeru se odpravi napaka stacionarnega stanja. Integralno nabiranje (*ang. Integral windup*) lahko povzroči nasičenost PI krmiljenja, nasičenost povzroči, da se napaka nikoli ne vrne na nič. Integralna komponenta je posledično ponavadi omejena. Integralna komponento množimo z parametrom K_I . Integralna komponenta upočasni spremembe krmilnega signala in lahko povzroča oscilacije.

2.4.3 Diferencialni del/odziv

Diferencialna komponenta povzroči, da se izhod zmanjša, če se procesna spremenljivka hitro spreminja. Diferencialni del je odvod procesne spremenljivke po času, pomnožen s parametrom K_D . Z večanjem časovnega koraka med meritvami se bo krmilni signal spremenjal hitreje. Diferencialni del zmanjšuje osciliranje sistema, v primeru dolgih časovnih korakov ali šuma v sistemu pa vnaša nestabilnosti.

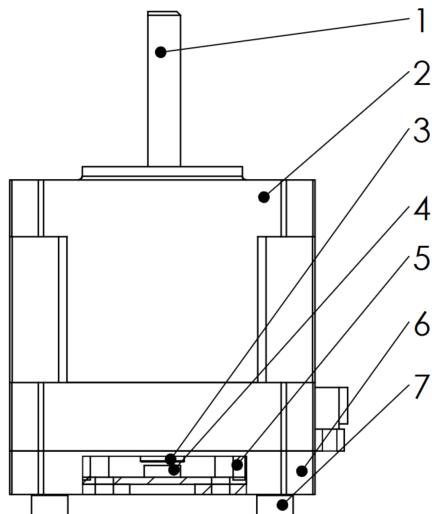
2.4.4 Nastavljanje parametrov

Za nastavljanje parametrov (*ang. Tuning*) K_P , K_I , K_D obstaja več metod. Pri uporabi metode **Ziegler-Nichols** se parametre nastavlja po sklopih, najprej samo K_P ($K_I, K_D = 0$), ko je določena najvišja vrednost parametra K_P - vrednost pri kateri sistem ne oscilira, se nastavi še parameter K_I ($K_D = 0$). Ko je sistem PI stabilen - pri izbranih parametrih K_P, K_I ne oscilira - se na enak način določi še parameter K_D .

3 Metodologija raziskave

3.1 Komponente

Servomotor je sestavljen iz koračnega motorja (1 in 2), magneta (3), kodirnika AS5600 (4), držala kodirnika (5), nosilca kodirnika (6) ter štirih vijakov M3 (7). Sestav je prikazan na spodnji sliki 3.1.



Slika 3.1: Sestavna risba servomotorja

Mikrokrmilnik ter gonilnik motorja na sliki nista prikazana.

3.1.1 Mikrokrmilnik

Za krmiljenje servomotorja je uporabljena razvojna ploščica Waveshare RP2040-Zero. Na shemi električnih povezav je prikazana kot RP2040. Vrtenje servomotorja nadzorujemo preko računalnika, ki z mikrokrmilnikom komunicira preko serijske povezave. Mikrokrmilnik je povezan s senzorjem rotacije AS5600, ki komunicira preko protokola I^2C in gonilnika koračnega motorja A4988.

3.1.2 Gonilnik motorjev A4988

Za poganjanje koračnega motorja je uporabljen krmilnik A4988 proizvajalca Allegro Microsystems. Ta omogoča uporabo mikrokorakov velikosti $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ in $\frac{1}{16}$. Krmiljen je z uporabo GPIO izhodov mikrokrmilnika. Smer vrtenja, impulz za korak ter velikost mikrokorakov določajo stanja vhodov na gonilniku motorja. Povezava med gonilnikom in mikrokrmilnikom je prikazana na shemi povezav (Slika 3.3).

Velikost mikrokoraka je določena glede na stanja vhodov MS1, MS2 in MS3. Stanja in velikost koraka so prikazana v preglednici 3.1.

Preglednica 3.1: Logična preglednica mikrokorakov

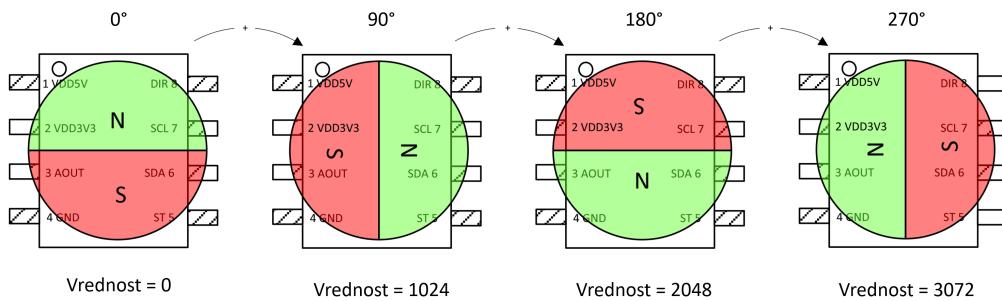
MS1	MS2	MS3	Velikost mikrokoraka
0	0	0	1
1	0	0	$\frac{1}{2}$
0	1	0	$\frac{1}{4}$
1	1	0	$\frac{1}{8}$
1	1	1	$\frac{1}{16}$

Smer rotacije določa stanje vhoda DIR, korak pa sproži impulz na vhodu STEP. Hitrost rotacije določa čas med impulzi, ne pa trajanje impulza. Iz hitrosti, ki je ponavadi podana v $\frac{\text{obrati}}{\text{minuto}}$ se po enačbi 3.1 izračuna čas med impulzi. Ta je odvisen od korakov v celotni rotaciji motorja (*KORAKI*), stopnje mikrokorakov (μ) in hitrosti podane v $\frac{\text{obrati}}{\text{minuto}}$ (*RPM*).

$$t_{medkoraki} = \frac{60}{RPM \cdot \mu \cdot KORAKI} [s] \quad (3.1)$$

3.1.3 Senzor rotacije AMS AS5600

Za zaznavanje rotacije gredi koračnega motorja je uporabljen magnetni diferencialni kodirnik AMS AS5600. Kodirnik z mikrokrmlnikom komunicira preko protokola I^2C . Povezava je prikazana na shemi vezja, slika 3.3. Preko I^2C protokola na mikrokrmlniku se pridobi kot kodirnika, to je odvisno od rotacije magneta. Stanja so prikazana na sliki 3.2.



Slika 3.2: Stanje senzorja glede na rotacijo magneta [5]

Specifikacije in lastnosti so prikazane v preglednici 3.2.

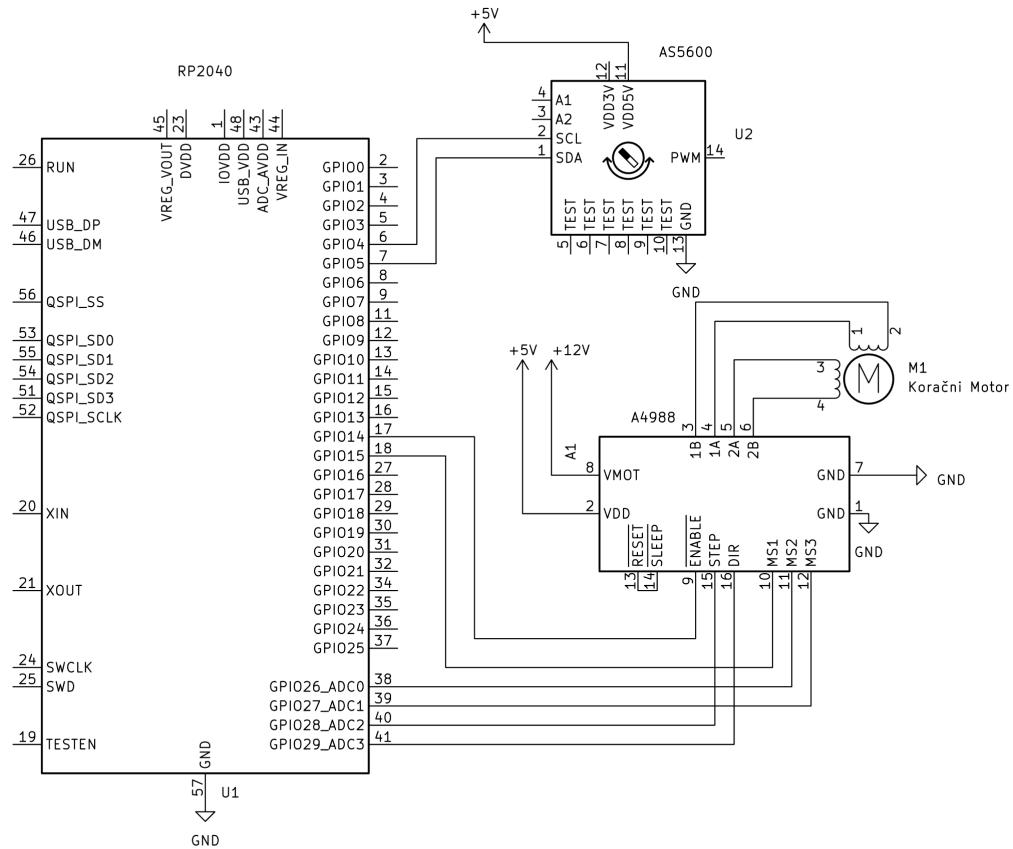
Preglednica 3.2: Specifikacije kodirnika AS5600

Specifikacija	Vrednost
Ločljivost	12bit
Število razdelkov (enačba: 2.6)	4096
Velikost razdelka (enačba: 2.7)	0.08789°
Teoretični čas branja senzorja	450 µs
Izmerjen čas branja	580µs

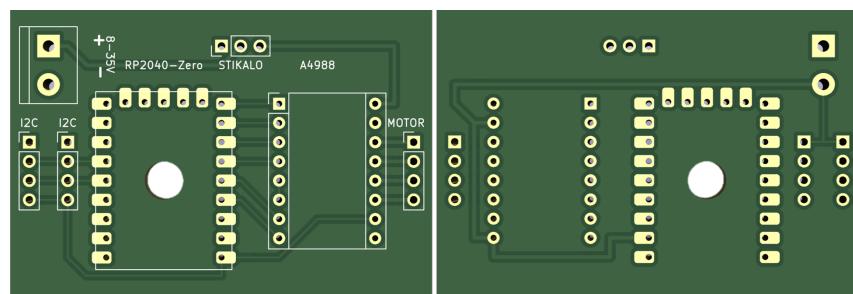
Pri izdelavi servomotorja sta najpomembnejši lastnosti ločljivost in čas branja. Ločljivost kodirnika določi natančnost premika, čas branja pa najhitrejše vrtenje. Pri izračunu časa med koraki (enačba 3.1) čas branja ni upoštevan, vendar pa vpliva na hitrost vrtenja pri manjših $t_{medkoraki}$.

3.1.4 Elektronsko vezje in shema

Na sliki 3.3 je prikazana shema komponent in povezav med njimi. Služi kot osnova za izdelavo prototipa in tiskanega vezja. Za izdelavo sheme in tiskanega vezja je bil uporabljen program KiCad 7.0. Načrt tiskanega vezja je prikazan na sliki 3.4.



Slika 3.3: Shema vezja



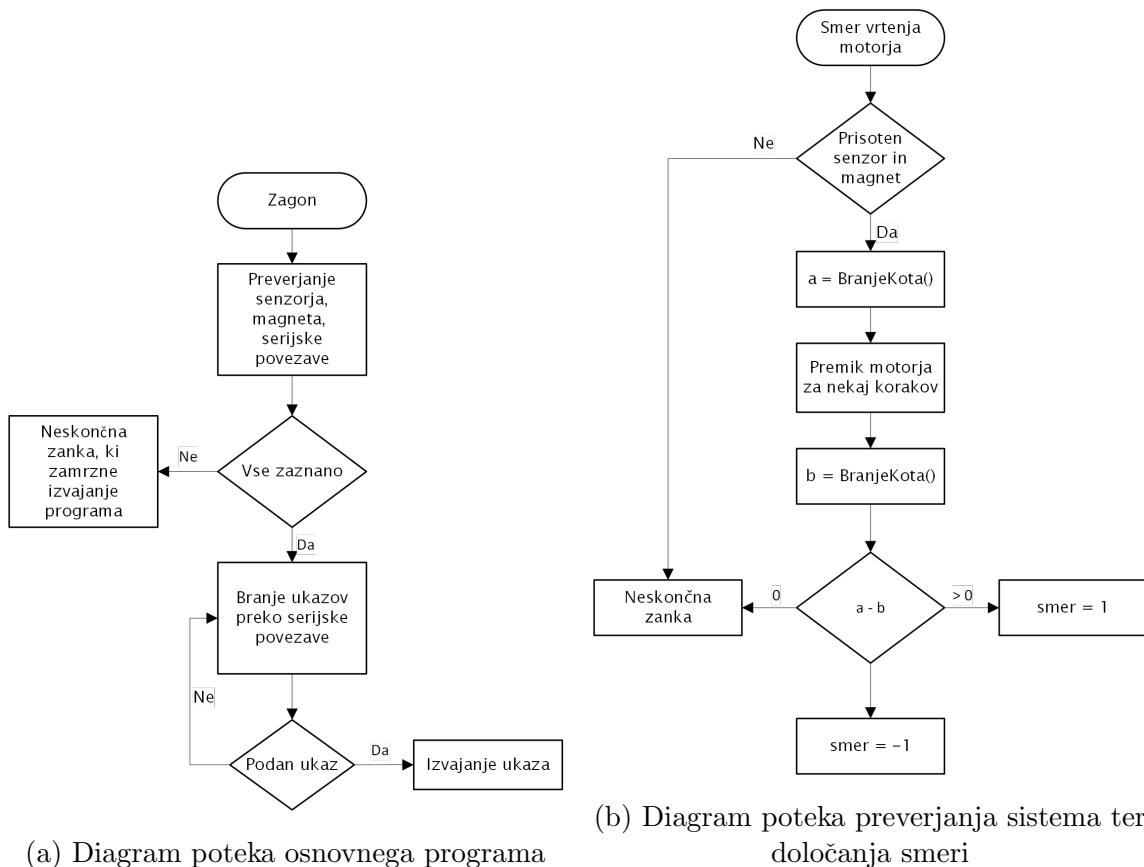
Slika 3.4: Prikaz spodnje in zgornje strani vezja

3.2 Program

3.2.1 Krmiljenje koračnega motorja

Program mikrokrmlnika je narejen v programskem okolju Arduino, predstavljen je z diagrami poteka (*ang. flowchart*). Program je razdeljen na dva glavna dela - prvi preveri prisotnost Serijske povezave, senzorja, magneta ter začetno nastavitev servomotorja, drugi pa sprejema ukaze preko serijske komunikacije in jih izvaja. Uporabljene so bil knjižnice Serial, StepperDriver ter AS5600. Na sliki 3.5a je prikazan diagram poteka osnovnega programa.

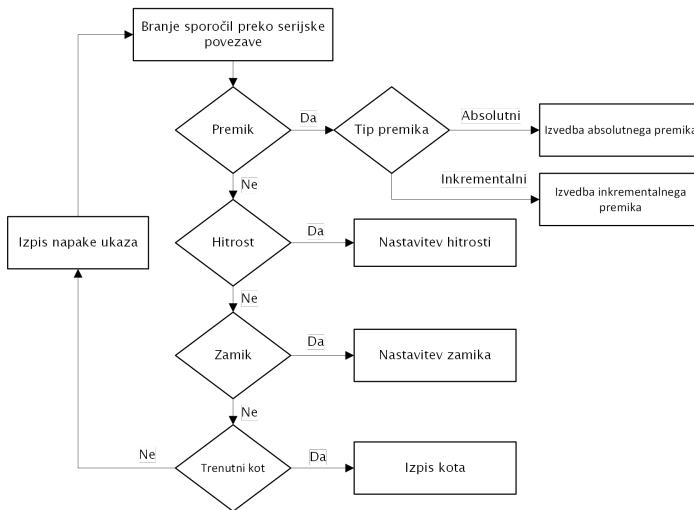
Na začetku izvajanja programa se vzpostavi serijska komunikacija z računalnikom, uporabi se vgrajena knjižnica *Serial* ter inicializira delovanje gonilnika motorja. Knjižnica *AS5600* omogoča preverjanje povezave senzorja ter magnitude magneta. V primeru da senzor ali magnet nista zaznana oziroma je magnituda magneta prevelika ali premajhna, program preko serijske komunikacije izpiše sporočilo napake ter začne izvajati neskončno zanko, ki preprečuje sprejemanje in izvajanje ukazov. V začetnem delu programa se izvede tudi večkratno branje senzorja, pri čemer se določi povprečen čas branja - ta se lahko odšteje pri izračunu $t_{medkoraki}$. Preverjanje delovanja in določanje smeri vrtnega motorja je prikazano na sliki 3.5b.



Slika 3.5: Diagrama poteka osnovnega programa ter preverjanja delovanja

Če začetni del program ne izvaja neskončne zanke, začne brati sporočila preko serijske

povezave. Če se prejeto sporočilo ujema z ukazom, ga izvede. Tip ukaza je določen iz prve črke in dolžine ukaza. Diagram poteka je prikazan na sliki 3.6. Ukaza za nastavitev zamika in hitrosti uporabita vgrajene funkcije arduino knjižnic. Nastavljena hitrost je uporabljena pri računanju časa med koraki, zamik kodirnika pa za nastavljanje novega kota. Ob zagonu mikrokrnilnika se nastavita na predpisane začetne vrednosti, funkcija branja kota pa izpiše trenutni kot.



Slika 3.6: Diagram poteka branja in izvedbe ukaza

Servomotor ima dva ukaza za premik - absolutni in inkrementalni. Absolutni pomik završi gred na želeno pozicijo, inkrementalni pa za željen kot. Funkcijam pomika je potrebno podati tudi smer vrtenja. Funkciji se delita na dva dela - izračun korakov in zamika $t_{medkoraki}$ ter izvajanje in preverjanje rotacije. Algoritem za izračun korakov je pri obeh premikih enak. V izračun števila korakov je podana želena rotacija ter lastnost koračnega motorja. Željen kot se celoštevilsko deli z velikostjo koraka, napako izračunamo kot razliko med želenim kotom ter zmnožkom korakov ter njihovo velikostjo. Število korakov se zmnoži z napako, tako se dobi koeficient napake. Izberemo tisto velikost koraka ter število, ki tvorita najmanjši koeficient napake. Večji kot je korak manj na hitrost vpliva čas branja senzorja.

Izračun števila korakov in velikosti mikrokorakov je prikazan v izseku kode 3.1

```

int stevilo_korakov = 0;
int mikrokoraki = 0;
float min_koeficient = 1000.0; // Neka zelo velika stevilka
for(int i=0; i <= 4; i++){
    float velikost_koraka = 1.8 / pow(2.0, (float) i); //1.8 -> lastnost motorja
    int koraki = zeljeni_kot / velikost_koraka;
    float napaka = zeljeni_kot - koraki * velikost_koraka; //Napaka, ki jo bomo dobili
    float koef_napake = napaka * koraki;
    if(koef_napake < min_koeficient){
        stevilo_korakov = koraki;
        mikrokoraki = (int) pow(2.0, (float) i);
        min_koeficient = koef_napake;
    }
}
  
```

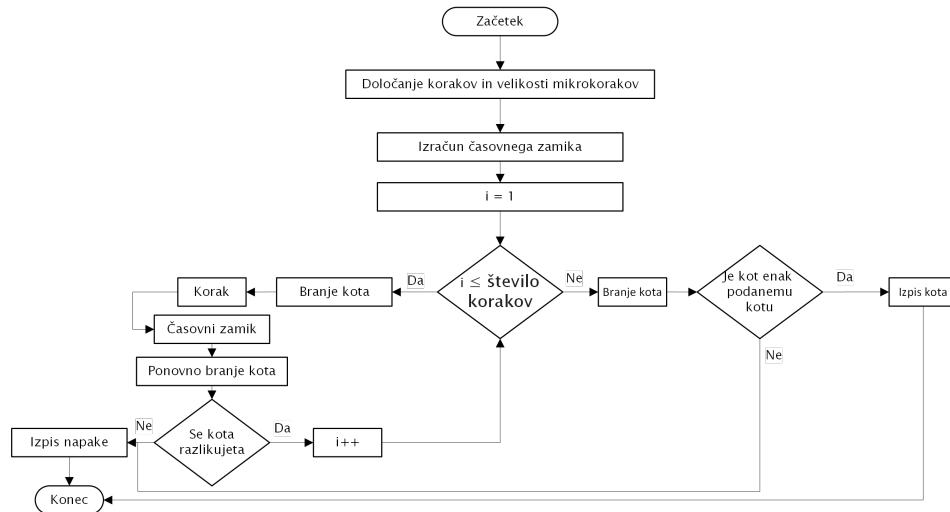
```

    }
}

```

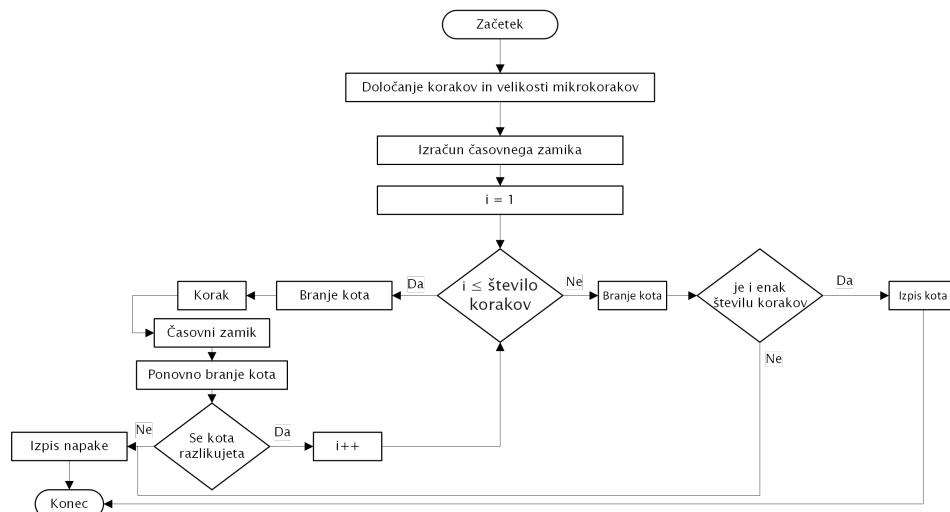
Izsek kode 3.1: izračun korakov ter velikosti mikrokorakov

Diagram poteka absolutnega premika je prikazan na sliki 3.7. Na sliki 3.8 je prikazan



Slika 3.7: Diagram poteka absolutnega premika

diagram poteka inkrementalnega premika. Razlika med funkcijama inkrementalnega

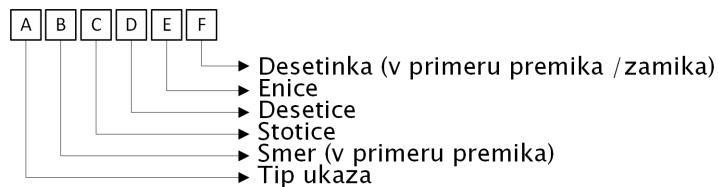


Slika 3.8: Diagram poteka inkrementalnega premika

in absolutnega pomika je v preverjanju opravljenega premika. Pri absolutnem pomiku se željen kot primerja s trenutnim. V primeru da je končni kot v okolici želenega kota se premik upošteva kot uspešen. Pri intkrementalnemu pomiku pa se preverja število opravljenih korakov, za opravljen korak se upošteva vsak pomik, pri katerem je zaznana sprememba kota. V primeru, da bi program sešteval spremembe kota namesto korakov, bi morali upoštevati tudi prehod preko 360° oziroma 0° ter seštevanje napake meritev.

3.2.2 Komunikacija s servomotorjem

Pošiljanje in prejemanje informacij poteka preko serijske povezave. Vsak ukaz vsebuje eno črko in števke, ukaz za premik ima na drugem mestu oznako za smer premika. Format ukaza na sliki 3.9.



Slika 3.9: Format ukaza

V preglednici 3.3 so podane posamične komponente ukaza.

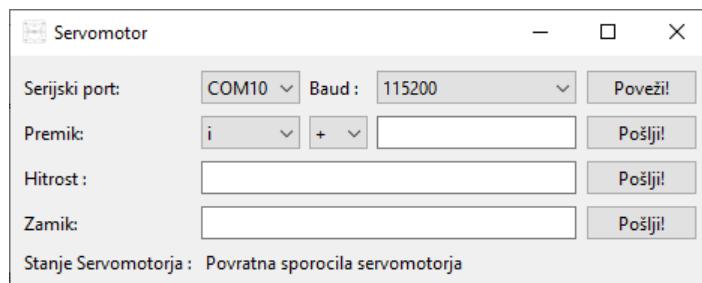
Preglednica 3.3: Deli ukaza

Ukaz	A	B	C	D	E	F
Premik	a/i	+/-/0	Števka	Števka	Števka	Števka
Zamik	n	Števka	Števka	Števka	Števka	-
Hitrost	r	Števka	Števka	Števka	-	-
Trenutni kot	t	k	-	-	-	-

Ukaz za premik je sestavljen iz tipa premika - absolutni oziroma inkrementalni, smeri - smer urinega kazalca (+) in nasproti urinega kazalca (-) oziroma najmanjšega pomika (0) v primeru absolutnega premika. Ukaz za inkrementalni premik v smeri urinega kazalca za 123.4° je $i+1234$. Ukaz za nastavitev zamika je sestavljen iz črke n ter kota (npr.: n1234 nastavi zamik na 123.4°). Ukaz za hitrost je sestavljen iz črke r ter celega števila (npr.: r50 nastavi hitrost na $50 \frac{\text{obrat}}{\text{min}}$). Z ukazom tk pa izpišemo trenutni kot.

3.2.3 Izdelava grafičnega vmesnika

Za lažjo upravljanje servomotorja je bil izdelan grafični vmesnik v okolju Python z uporabo PyQt6. Grafični vmesnik je prikazan na sliki 3.10. Grafični vmesnik



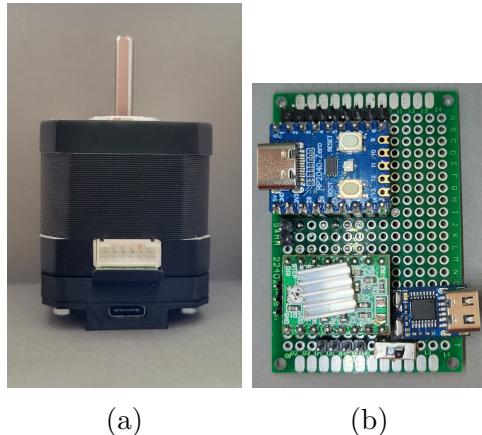
Slika 3.10: Grafični vmesnik za upravljanje servomotorja

omogoča izbiro serijskega porta ter hitrost serijske komunikacije. S klikom na gumb "Poveži!" vzpostavimo serijsko povezavo, preko katere posiljamo ukaze ter prejemamo povratne informacije. V vsako vhodno polje vpišemo samo številke z gumbom "Pošlji!" pa se formira in pošlje ukaz. Povratno sporočilo se prikaže v spodnjem delu programa pod polja Štanje Servomotorja :".

4 Rezultati

4.1 Izdelani servomotor

Izdelan servomotor in prototipno vezje sta prikazana na sliki 4.1. Uporabljene so



Slika 4.1: Servomotor (a) in prototipno vezje (b)

komponente predstavljene v poglavju 3.1. Nosilec za senzor in držalo sta 3D tiskana. Za napajanje vezja je uporabljen napajalnik, ki podpira standard PD ter USB-C vhod s čipom za izbiro napetosti. Senzor je bil zaradi lažje uporabe z mikrokrmlnikom povezan preko USB-C kabla in razširitvenih ploščic z USB-C vhodi, protokol povezave je I^2C . Magnet je na gred rotorja prilepljen s hitrim lepilom.

4.2 Analiza delovanja

4.2.1 Ponovljivost rotacije

Ponovljivost servomotorja je bila izmerjena z izvajanjem enako velikih rotacij v obe smeri pri različnih hitrostih. Velikost kota rotacije je bila med 4.5° in 90° , večala se je s korakom 4.5° . Računali smo povprečje razlike kotov pred in po rotaciji v obe smeri, to je bilo izračunano za različne hitrosti. Rezultati meritev so podani v preglednici 4.1.

Preglednica 4.1: Rezultati meritev ponovljivosti

Hitrost [$\frac{\text{obr}}{\text{min}}$]	1	2	3	Povprečna vrednost
10	-0,01	0,00	-0,04	-0,017
20	-0,03	-0,01	0,04	0,020
30	-0,4	-0,38	-0,40	-0,393
40	0,09	0	0,06	0,060
50	0,21	0,23	-0,04	0,133
60	-1,79	-3,55	-3,44	-2,927
70	-0,31	-0,19	-0,37	-0,290
80	-0,02	0,00	0,03	-0,003
90	0,13	1,17	0,13	0,477
100	0,22	0,23	0,22	0,223

4.2.2 Čas branja senzorja

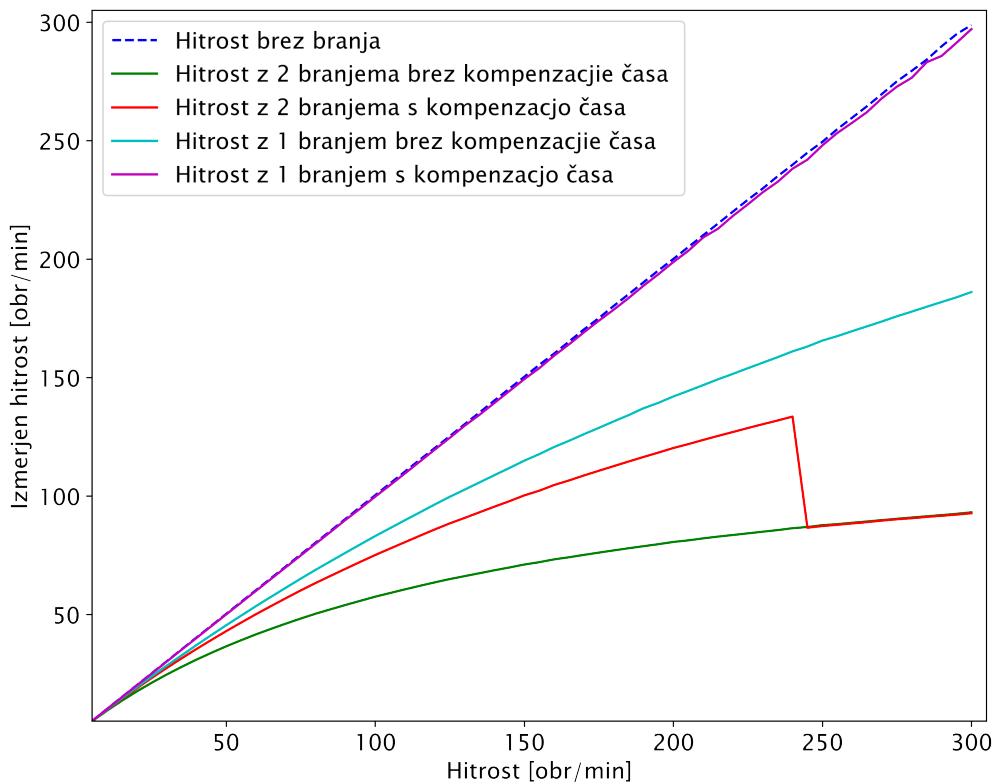
Branje kota vpliva na hitrost vrtenja, saj v program vnaša dodaten zamik. Pri določanju se čas branja lahko odšteje od $t_{medkoraki}$. Za kompenzacijo zamika je potrebno izmeriti povprečen čas branja, določimo ga z večkratnim branjem po prižigu mikrokrmilnika. Povprečni čas 20 branj znaša 559.92 μs , standardni odklon meritev pa 10.64 μs .

4.2.3 Hitrost vrtenja servomotorja

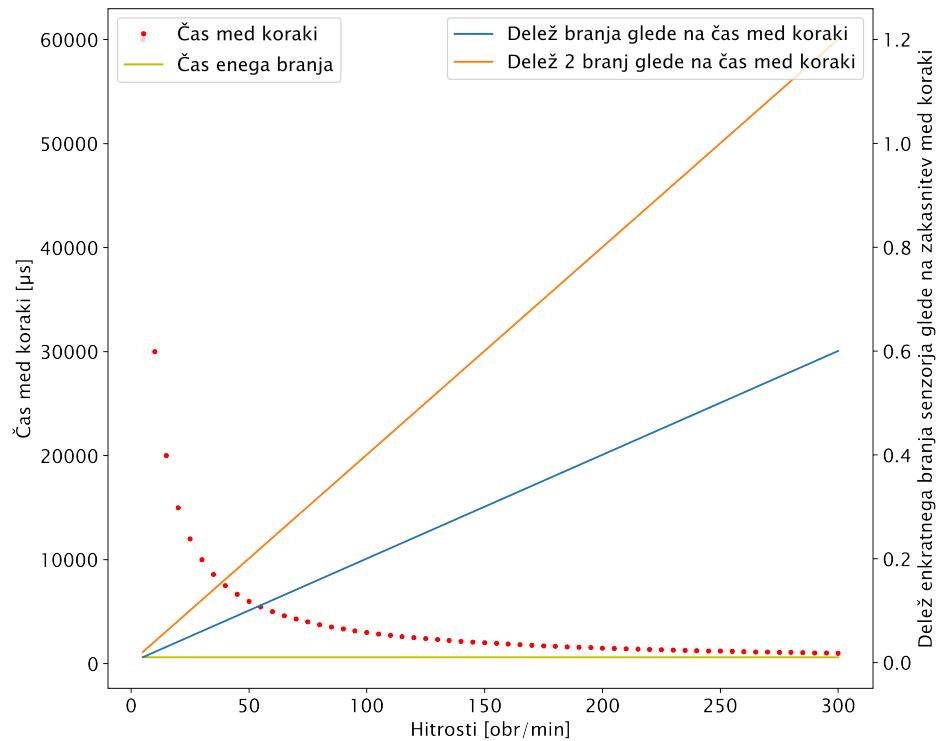
Hitrost vrtenja koračnega motorja se določi s zamiki med koraki. Za oziroma pred korakom sistem prebere kot trenutne pozicije gredi. Čas porabljen za branje torej vpliva na hitrost vrtenja. Za določitev vpliva branja in posledične kompenzacije časa servomotor po prižigu večkrat zaporedno bere iz senzorja rotacije ter izračuna povprečni čas branja. Ta je lahko naknadno uporabljen za popravo zamika med koraki. Vplivi branja so prikazani na spodnjem grafu (slika 4.2). Razpon meritev hitrosti je od 5 do 300 $\frac{\text{obrati}}{\text{min}}$, s korakom 5 $\frac{\text{obrati}}{\text{min}}$.

Značilnica imenovana *Hitrost brez branja* predstavlja izmerjeno hitrost pri premiku koračnega motorja za 200 korakov (1 obrat) z uporabo knjižnice StepperDriver. Ostale značilnice predstavljajo premik motorja za 360° oziroma 200 korakov z uporabo funkcije za inkrementalni premik. Razlika med njimi je v številu branj senzorja ter kompenzaciji časa branja. V primeru dveh branj je ta enaka času dveh branj senzorja, dokler je razlika med zamikom $t_{medkoraki}$ večja od dvokranika povprečnega časa branja. Časovni zamik med koraki je izračunan po enačbi 3.1, z upoštevanjem da je pienak 1. Na spodnjem grafu (slika 4.3) je so prikazani izračunani zamiki med koraki $t_{medkoraki}$ ter delež časa branja senzorja glede na $t_{medkoraki}$.

Razvidno je, da delež branja senzorja v primerjavi s $t_{medkoraki}$ narašča. Ko delež doseže 1, hitrost znaša le $\frac{1}{2}$ želene hitrosti, kompenzacija več ni mogoča. To dosežemo pri hitrosti 245 $\frac{\text{obrati}}{\text{min}}$. Zakasnitev zaradi branja senzorja se lahko odpravi z uporabo hitrejšega senzorja ali pa z uporabo vzporednega izvajanja meritev s uporabo večih procesorskih jeder oziroma niti.



Slika 4.2: Graf hitrosti vrtenja pri različnih številih branj

Slika 4.3: Graf $t_{medkoraki}$ ter delež branja senzorja

5 Zaključki

V zaključni nalogi je bila opisana teorija delovanja koračnih motorjev, absolutnih magnetnih kodirnikov, gonilnikov motorjev, PID krmiljenja in nastavljanje parametrov ter mikrormilnik. Predstavljeni so bili algoritmi za krmiljenje ter rotacijo servomotorja.

1. Izdelan je bil servomotor, vezje za povezavo mikrokrmilnika z gonilnikom motorja ter senzorjem rotacije, program mikrokrmilnika
2. Izdelan je bil grafični vmesnik za nadzor motorja
3. Izmerjen je bil čas branja kota, ta znaša pribljižno 560 µs.
4. Izmerjena je bil ponovljivost rotacije motorja, povprečna napaka znaša $-0,2717^\circ$.
5. Izmerjena je bila najvičja hitrost, ki jo servomotoru doseže, ta znaša $300 \frac{obr}{min}$

Predlogi za nadaljnje delo in izboljšave

Za izboljšavo servomotorja bi bilo potrebno uporabiti bolj natančen in hitrejši kodirnik, gonilnik motorja, ki podpira manjše velikosti mikrokorakov. Program, ki izvaja premike, bi izboljšali z uporabo več niti (*ang. threads*), kar bi omogočilo odpravo odstopanja hitrosti. Izvajanje več korakov brez zakasnitve bi omogočilo uporabo pospeševanja.

Literatura

- [1] Austin Hughes, *Electric Motors and Drives:Fundamentals, Types and Applications*. London: Elsevier, 2006
- [2] Dipankar Mitra, *How to Improve Motion Smoothness and Accuracy of Stepper Motors*, Texas Instruments, oktober 2021. dostopno na: <https://www.ti.com/lit/an/sloa293a/sloa293a.pdf>
- [3] Michael Schultis, *Stepper Motor Driving With Peripheral Drivers (Driver ICs)*, december 2016. dostopno na: <https://www.ti.com/lit/an/slva767a/slva767a.pdf>
- [4] *High Resolution Microstepping Driver*, september 2021. dostopno na: <https://www.ti.com/lit/an/slva416a/slva416a.pdf>
- [5] amsDatasheet, *Product Documnet - AS5600*, junij 2018. dostopno na: https://ams.com/documents/20143/36005/AS5600_DS000365_5-00.pdf
- [6] ams Application Note, *AN5000 Rotary Magnetic Position Sensors Magnet Selection Guide*, september 2016. dostopno na: <https://look.ams-osram.com/m/1f9fec31c21d8f4d/original/AnglePositionOnAxis-AN000271.pdf>
- [7] Paul Acarnley, *STEPPING MOTORS a guide to theory and practice, 4th edition*. London: IEE, 2002
- [8] National instruments, PID Theory Explained, marec 2011. dostopno na: http://experimentationlab.berkeley.edu/sites/default/files/PID_Theory_Explained.pdf
- [9] NXP Semiconductors, I2C-bus specification and user manual Rev.7, oktober 2021. dostopno na: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [10] Universal asynchronous receiver-transmitter, dostopno na: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter [ogled: 7.7.2023].
- [11] Allegro Mictosystems, A4988 Datasheet. Dostopno na <https://www.allegromicro.com/~/media/Files/Datasheets/A4988-Datasheet.ashx>

