

**»Mladi za napredek Maribora 2017«**  
**34. srečanje**

**Miza na dveh kolesih**

Raziskovalno področje ELEKTROTEHNIKA, ELEKTRONIKA

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor: ALJAŽ ROŽIČ  
Mentor: MILAN IVIČ  
Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

**Maribor, januar 2017**

## KAZALO

1	Povzetek.....	4
2	Uvod .....	4
3	Vsebinski del .....	5
3.1	MyRio .....	5
3.1.1	Konektor A in konektor B .....	6
3.1.2	Konektor C .....	7
3.2	Giroskop .....	8
3.3	Enosmerni motorčki s kodirnikom.....	9
3.4	Krmilnik enosmernih motorčkov.....	10
4	Programiranje in opis programske kode .....	11
4.1	Programsko okolje LabVIEW .....	11
4.2	Načrtovanje .....	19
4.3	Izdelava 3D modela.....	19
4.4	Povezovanje in preizkus.....	22
5	Zaključek .....	23
6	Družbena odgovornost .....	23
7	Viri .....	24

## Kazalo slik

Slika 1: Krmilni modul myRIO (vir: National Instruments). ....	5
Slika 2: Konektorja A in B (vir: National Instruments). ....	6
Slika 3: Konektor C (vir: National Instruments). ....	7
Slika 4: Pmod giroskop (vir: <a href="http://cdn6.bigcommerce.com">http://cdn6.bigcommerce.com</a> ). ....	8
Slika 5: Enosmerni motorček s prenosom 1:19 (vir: Avtor naloge). ....	9
Slika 6: Delovanje enkoderja (vir: Avtor naloge). ....	9
Slika 7: Krmilnik motorjev (vir: <a href="https://reference.digilentinc.com">https://reference.digilentinc.com</a> ). ....	10
Slika 8: Čelna plošča (vir: Avtor naloge). ....	12
Slika 9: Blok diagram (vir: Avtor naloge). ....	12
Slika 10: Paleta kontrol in indikatorjev (vir: Avtor naloge). ....	13
Slika 11: Paleta funkcij (vir: Avtor naloge). ....	14
Slika 12: Instalacija registrov (vir: Avtor naloge). ....	16
Slika 13: Shared Variable (vir: Avtor naloge). ....	16
Slika 14: Funkcijski blok za PID regulacijo (vir: Avtor naloge). ....	17
Slika 15: Programska koda za enkoder (vir: Avtor naloge). ....	18
Slika 16: Grafični vmesnik oziroma čelna plošča (vir: Avtor naloge). ....	18
Slika 17: Skica sil (vir: Avtor naloge). ....	19
Slika 18: Ustvarjanje novega okna. (vir: Avtor naloge). ....	20
Slika 19: Izбира. (vir: Avtor naloge). ....	20
Slika 20: Risanje nosilcev za motorje (vir: Avtor naloge). ....	21
Slika 21: 3D model, pripravljen za tisk (vir: Avtor naloge). ....	22

## 1 POVZETEK

Naloga prikazuje stabilizacijsko mizo oziroma robota na dveh kolesih, ki nam lahko pripelje tudi kakšno bremen na primer čaj ali kavo in pri tem vzdržuje ravnotežje. Za nadzor mize smo uporabili krmilni modul myRIO1900 proizvajalca National Instruments. Za stabilizacijo oziroma prepoznavanje spremembe položaja in naklona mizice skrbi girooskop. Za ustrezno spreminjanje hitrosti naprave smo uporabili posebna enosmerna motorčka z vgrajenima enkorderjema. Služita kot senzorja hitrosti vrtljajev, hkrati pa lahko krmilimo precizne nastavitve pri gibanju naprave. Mizico oziroma robot lahko vodimo z računalnikom, pametnim telefonom ali s tabličnim računalnikom.

## 2 UVOD

Ko smo izdelovali različne projekte s krmilnim modulom myRIO in uporabo programskega okolja LabVIEW, smo ob pogledu na *sagway* prišli na zamisel, da bi izdelali robot na dveh kolesih. Odločili smo se, da bo glavni del robota krmilna kartica NI myRIO1900. Robot smo poimenovali mizica, saj bo njena naloga prevažati ustrezne tovore, lahko zdravila, čaj oziroma druge predmete. Tako bo lahko koristno uporabljena za pomoč osebam z zmanjšano telesno gibljivostjo, bolnikom, ...

Uporabili bi jo lahko tudi za popestritev mladim družinam, kot didaktične igrače za njihove otroke.

Za realizacijo zastavljene naloge smo morali spoznati programsko okolje LabVIEW in krmilni modul oziroma kartico myRIO. Proučiti smo morali tudi načine regulacij, posebej še PID regulacijo, ki je neobhodno potrebna za kvalitetno delovanje robota.

## 3 VSEBINSKI DEL

### 3.1 MyRio



Slika 1: Krmilni modul myRIO (vir: National Instruments).

Krmilni modul myRIO proizvajalca National Instruments poganja dvojedrni procesor ARM Cortex A9 s čipom Xilinx FPGA (Field Programmable Gate Array), kar nam omogoča branje in obdelovanje večje količine podatkov v realnem času. Krmilni modul myRIO vsebuje tri konektorje, konektor A, konektor B in konektor C. Na vseh treh lahko koristimo 10 analognih vhodov, 6 analognih izhodov in 40 digitalnih vhodov oziroma izhodov. Pozorni moramo biti, da ne preobremenimo posameznih izhodov.

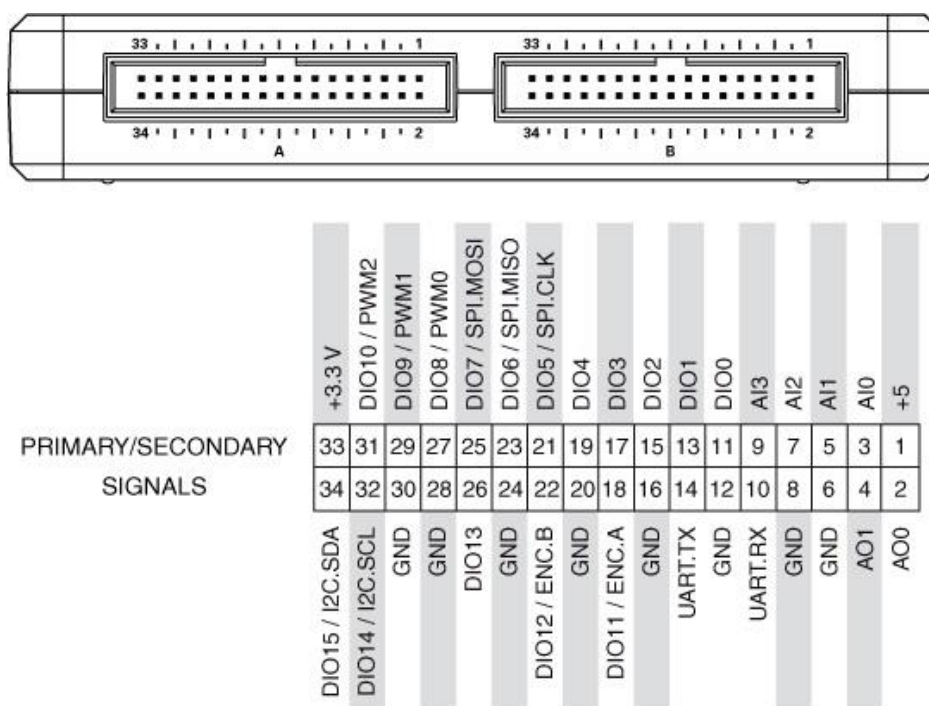
Modul myRIO najprej priključimo na napetostno napajanje, nato pa ga prek USB vrat povežemo z računalnikom. Odpre se okno *myRIO USB Monitor*, kjer je podatek s serijsko številko modula in IP naslovom.

Posebnost tega krmilnega modula sta USB konektor, audio vhod in audio izhod. Modul lahko z računalnikom komunicira na dva načina, prek USB in brezžične Wi-Fi povezave.

Za napetostno napajanje modula myRIO potrebujemo napajalnik moči 14 W z napetostjo od 6 V do 16 V.

### 3.1.1 Konektor A in konektor B

Konektorja vsebujeta 16 vhodno/izhodnih digitalnih priključkov, ki imajo oznake od DIO0 do DIO15 in jih lahko uporabljamo kot digitalne vhode ali izhode. Priključki od DIO0 do DIO13 so povezani s *pull-up* upori vrednosti 40 kΩ na napetost 3,3 V, kar se razlikuje od digitalnih priključkov mikrokontrolerja. Zato moramo paziti, kako priključujemo posamezne elemente, da bo delovanje pravilno. Priključka DIO14 in DIO15 pa sta povezana s pull-up uporoma vrednosti 2,2 kΩ na napetost 3,3 V. Digitalne izhode na konektorjih A in B lahko obremenimo s tokom največ 4 mA. Nekateri priključki pa imajo že v naprej določene naloge oziroma funkcije, kot so na primer SPI in I2C vodilo ter pulzno širinsko modulacijo (PWM).



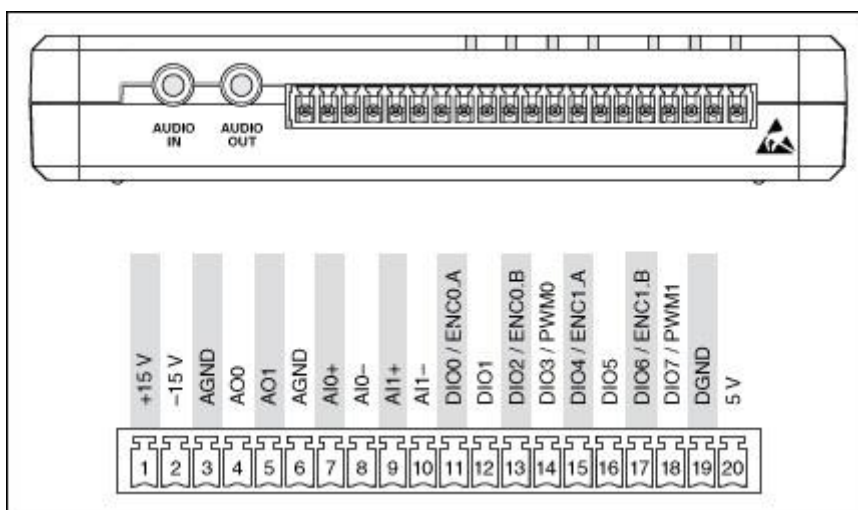
Slika 2: Konektorja A in B (vir: National Instruments).

### 3.1.2 Konektor C

Konektor C vsebuje 8 vhodno/izhodnih digitalnih priključkov z oznakami od DIO0 do DIO7. Priključki se od digitalnih priključkov konektorja A in B razlikujejo po tem, da so prek pull-down uporov vrednosti 40 k $\Omega$  povezani z maso in ne na napetost 3,3 V. Digitalne priključke na tem konektorju lahko obremenimo s tokom največ 4 mA.

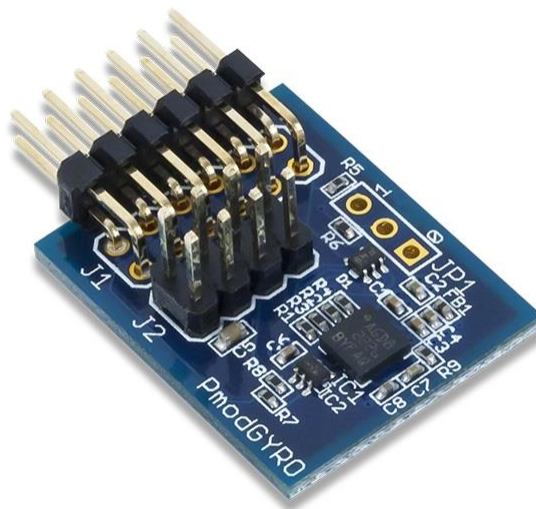
Analogni vhodi imajo 12-bitno ločljivost. Nanje lahko priključimo napetost od 0 V do 5 V. Na diferencialna analogna vhoda AI0 in AI1 konektorja C lahko priključimo napetost o območju od -10 V do +10 V.

Analogni izhodi vsebujejo 12-bitne digitalno analogne pretvornike. Maksimalna napetost na analognem izhodu znaša 5 V. Posamezni analogni izhod lahko obremenimo s tokom 3 mA. Maksimalna napetost na analognih izhodih AO0 in AO1 konektorja C znaša od -10 V do +10 V.



Slika 3: Konektor C (vir: National Instruments).

## 3.2 Girooskop



Slika 4: Pmod girooskop (vir: <http://cdn6.bigcommerce.com>).

Za zaznavanje položaja in kota smo uporabili girooskop L3G4200D podjetja Digilent. Senzor. Ima odlično ločljivost in je zato primeren za to raziskovalno nalogo. Z modulom NI myRio komunicira po I2C protokolu, ki zmanjša število potrebnih žic oziroma povezav s konektorji.

Vsa integrirana vezja ki so priključena v I2C mrežo si delijo vodilo, ki je sestavljeno iz dveh aktivnih dvosmernih linij (SCL in SDA) in skupne mase. Linija SDA služi za prenos podatkov in ukazov. Linija SCL je taktni signal, ki sinhronizira prenos in hkrati določa hitrost komunikacije. Vsako integrirano vezje, ki je priključeno na I2C vodilo mora imeti svoj naslov, ki se razlikuje od drugih in lahko deluje kot sprejemnik in/ali oddajnik.

Komunikacijo upravlja nadzorna enota *master*, v tem primeru je to krmili modul myRIO, ki generira takt in nastavlja podrejene enote *slave*, ter jim pošilja razne ukaze in podatke. Tudi podrejena enota lahko pošlje podatke nadzorni enoti, vendar le v primeru, ko jih le ta zahteva.

Dodatne specifikacije so :

- Podpira I2C standard 100 kHz in 400 kHz.
- Nastavljiva ločljivost 250/500/2000 dps.
- Način mirovanja.
- 12-pinski Pmod konektor s SPI vmesnikom.



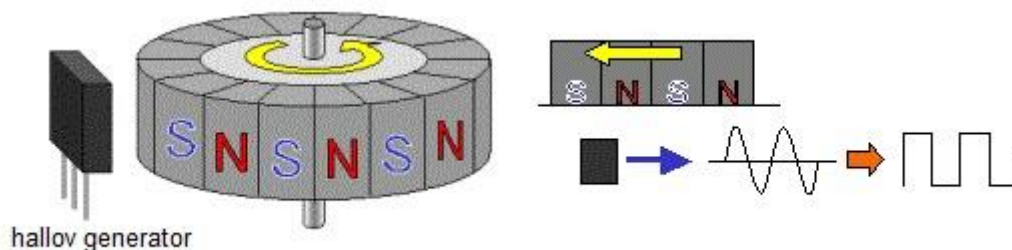
- Majhna velikost (2 cm × 2,5 cm).

### 3.3 Enosmerni motorčki s kodirnikom



Slika 5: Enosmerni motorček s prenosom 1:19 (vir: Avtor naloge).

Za poganjanjem mize smo izbrali enosmerna krtačna motorčka s prenosom 1:19. Motorčka imata vgrajen enkorder, ki nam omogoča branje položaja motorčka oziroma hitrost vrtenja.



Slika 6: Delovanje enkoderja (vir: Avtor naloge).

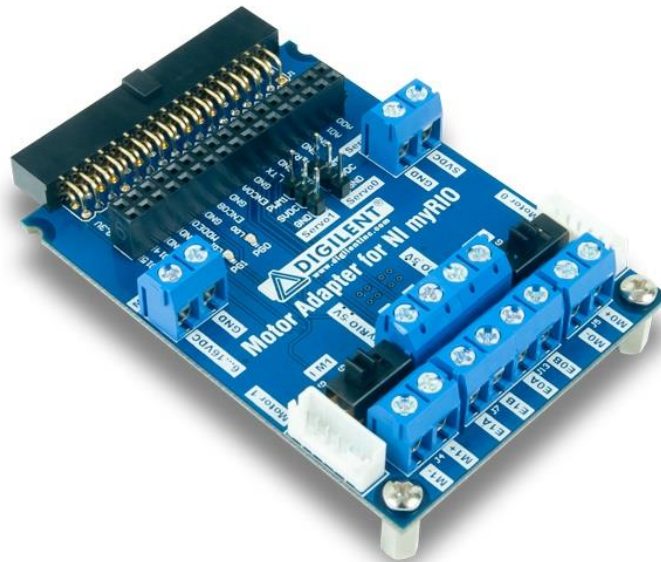
Enkorder je sestavljen iz magneta z več poli in hallovih generatorjev, ki se v tem primeru uporabljajo kot senzorji. Ob prehodu magnetnega pola preko hallovega generatorja se na njem pojavi magnetni fluks, ki povzroči sinusno napetost. Sinusno napetost pretvorimo v pravokotne signale. Z njihovo pomočjo programsko krmilimo spremembe položajev zasuka enosmernih motorčkov.

Dodatne specifikacije:

- Delovna napetost 12 V.
- Maksimalno število vrtljajev 1800 rpm.

- Maksimalen delovni tok 1 A.
- Teža 32 g.

### 3.4 Krmilnik enosmernih motorčkov



Slika 7: Krmilnik motorjev (vir: <https://reference.digilentinc.com>).

Za nadzor motorjev smo izbrali krmilno ploščico, ki predstavlja gonilnik za enosmerna motorčka. Proizvajalec krmilne ploščice je podjetje Diligent. Omogoča enostavno in široko uporabo, saj jo lahko uporabljamo kot krmilnik koračnih motorjev, enosmernih krtačnih, enosmernih brezkrtačnih motorjev ter servomotorjev. Krmilnik uporablja za kontroliranje integrirano vezje *full-bridge allegro A4973*. To omogoča kontroliranje dveh enosmernih motorčkov v obeh smereh vrtenja. Krmilna plošča je z modulom myRIO povezana z MXP konektorjem.

Dodatne specifikacije:

- Lahko kontroliramo 2 DC motorčka, 1 koračni motor ali 2 servomotorja.
- Zaznava trenutne napajalne napetosti.
- Vsebuje kratkostično zaščito.
- Vhodna napajalna napetost 6 V – 16 V.
- Vgrajen ima gonilnik enkorderja.

- Vsebuje LED indikator delovanja.
- Vsebuje 2 priključka za servomotor.

## 4 PROGRAMIRANJE IN OPIS PROGRAMSKE KODE

### 4.1 Programsko okolje LabVIEW

Programsko okolje LabVIEW (Laboratory Virtual Instrument Engineering Workbench) proizvajalca National Instruments (NI) smo uporabili za izdelavo programske kode nadzora robota oziroma krmilnega modula NI myRIO. Programsko okolje je v celoti zasnovano grafično in nam omogoča lažji razvoj kompleksnih testnih, merilnih in drugih podobnih aplikacij. Vsak program oziroma virtualni instrument je sestavljen čelne plošče in blok diagrama.

LabVIEW je blokovno programsko okolje oziroma orodje, kar pomeni, da se program v tem programskem jeziku kodira blokovno oziroma je sestavljen iz funkcij v obliki blokov, ki so med seboj povezani s povezavami. Orodje vsebuje tudi *Application Builder*, ki omogoča izdelavo izvršne (exe) verzije, tako se lahko izvaja na računalniku brez nameščenega orodja LabVIEW. To nam je koristilo pri kontroliranju robota na daljavo. Orodje je omrežno zasnovano, kar pomeni, da lahko instrument kreiramo po sklopih na več računalnikih v omrežju in ga združimo v celoto. Ta koncept se uporablja v raznih podjetjih, v katerih si razdelijo naloge in vsak opravi svoj del.

Vsak virtualni instrument oziroma merilna aplikacija je sestavljena iz treh segmentov:

1. Zajemanje podatkov.
2. Analiza podatkov.
3. Prikaz podatkov.

Ob zagonu programa LabVIEW se pojavi okno, kjer izberemo LabVIEW for myRIO. Lahko izbiramo med kreiranjem novega projekta ali odpiranjem obstoječih, shranjenih projektov. V kreiranem novem projektu, ki ga poimenujemo, izberemo novi virtualni instrument (VI). Ob tem se pojavita dve okni, Čelna plošča in Blok diagram. Čelna plošča ima sivo ozadje. Namenjeno je kreiranju vizualnega vmesnika. Blok diagram pa je namenjen kreiranju programske kode virtualnega instrumenta.

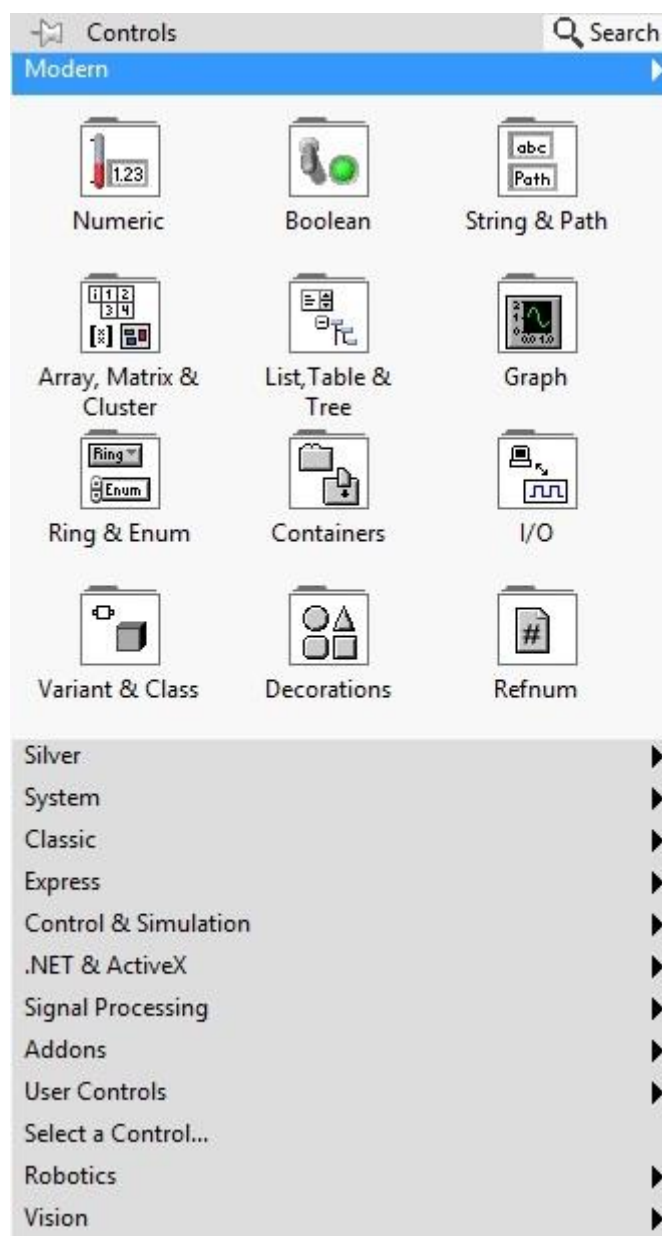


Slika 8: Čelna plošča (vir: Avtor naloge).



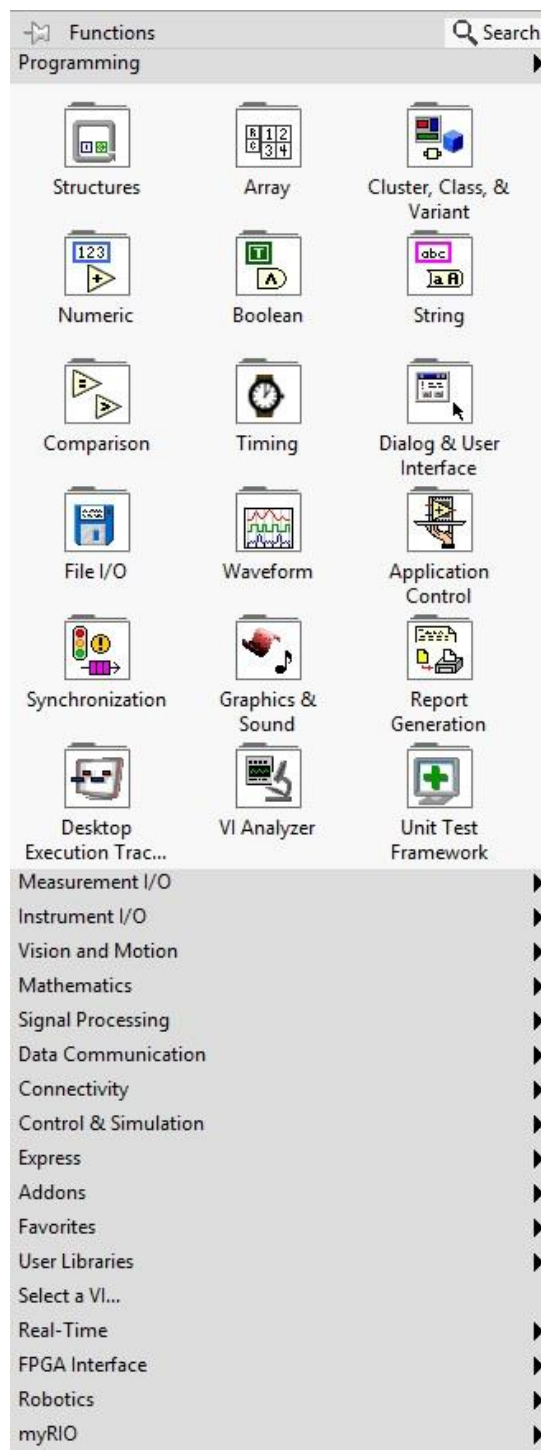
Slika 9: Blok diagram (vir: Avtor naloge).

Čelna plošča (Front Panel) predstavlja grafični vmesnik virtualnega inštrumenta. Sestavljen je iz raznih indikatorjev in kontrol, ki jih najdemo na paleti kontrol in indikatorjev. Do palete kontrol in indikatorjev dostopamo tako, da kjerkoli v praznem prostoru kliknemo na desno tipko miške ali preko izbire menija *View > Controls Palette*. Paleta kontrol in indikatorjev je sestavljena iz več podpalet v katerih lahko najdemo različne kontrole in indikatorje. Objekt (kontrolo ali indikator) namestimo na čelo ploščo tako, da ga izberemo in povlečemo na polje čelne plošče. Vhodni objekti, ki jim pravimo kontrole (*Controls*) predstavljajo vhodne podatke virtualnega instrumenta in jih lahko med delovanjem spreminjamo in s tem spremenimo oziroma vplivamo na delovanje virtualnega inštrumenta. Izhodni objekti, pravimo jim indikatorji (*Indicators*), služijo za prikaz podatkov. Med izvajanjem virtualnega instrumenta se izračunani podatki posredujejo indikatorjem, ki jih nato prikažejo v tekstovni, numerični ali grafični obliki. Objektom čelne plošče lahko preko menija objekta spreminjati njegove lastnosti. Do priročnega menija objekta dostopamo tako, da se s kurzorjem miške pomaknemo na objekt in pritisnemo na desno tipko miške.



Slika 10: Paleta kontrol in indikatorjev (vir: Avtor naloge).

Blok diagram je namenjen kodiranju programa virtualnega instrumenta (VI). Program oziroma algoritem zgradimo iz objektov, ki jih najdemo na paleti funkcij in jih v celoto povežemo s povezavami. Do palete funkcij dostopamo podobno kot do palete in kontrol indikatorjev. Funkcijo v blok diagram vstavimo tako, da jo izberemo in jo povlečemo v polje blok diagrama.



Slika 11: Paleta funkcij (vir: Avtor naloge).

Funkcijo namestimo v Blok diagram tako, da jo na paleti funkcij izberemo in povlečemo v Blok diagram.

V Blok diagramu virtualnega instrumenta se lahko nahajajo naslednji objekti:

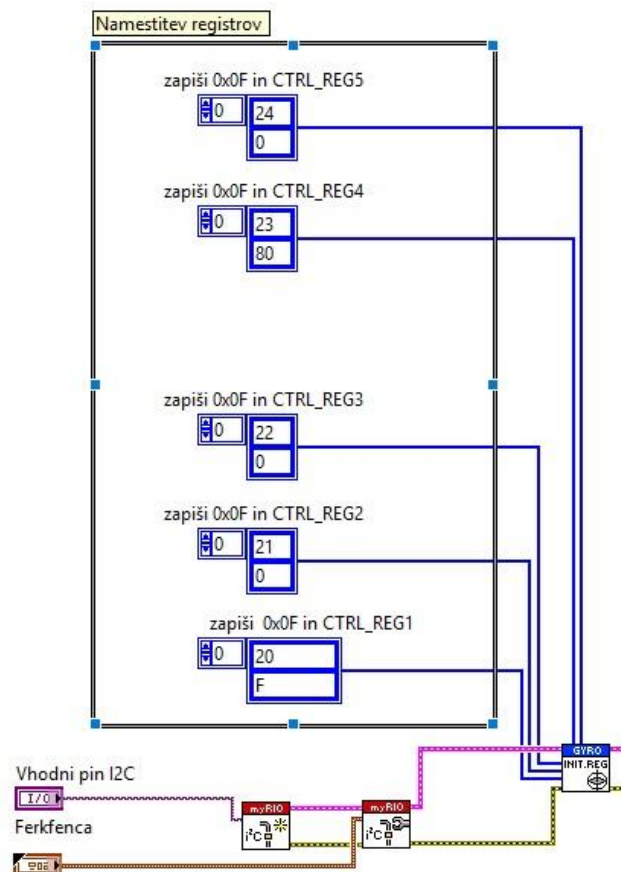
- Priključki (Terminals), to so priključni elementi objektov Čelne plošče.

- Podprogrami (SubVIs), to so dejansko virtualni instrumenti, ki prav tako vsebujejo Čelno ploščo in Blok diagram.
- Funkcije (Functions).
- Konstante (Constants), to so objekti v Blok diagramu, katerim se vrednost med izvajanjem virtualnega instrumenta ne spreminja.
- Programske strukture (Structures), to so objekti, kot so zanka While, zanka For, pogojni stavki (Case) zaporedje (Flat Sequence) in drugi objekti.
- Povezave (Wires), ki povezujejo posamezne objekte Blok diagrama in služijo prenosu podatkov.

Objekte v Blok diagramu, ki imajo vhode in izhode imenujemo Vozlišča (Nodes).

Po spoznavanju osnov programskega okolja LabVIEW smo se lotili pisanja programa za to raziskovalno nalogo. Najprej smo morali narediti program za branje giroskopa saj ga v LabVIEW ni in je s krmilnim modulom NI myRIO povezan po I2C povezavi. Definirali smo kanale na izbranem konektorju s katerimi je povezan giroskop in s katero frekvenco deluje. Nato smo v program namestili registre, ki so potrebni za branje oziroma delovanje giroskopa.

Ko smo narediti program za branje giroskopa, smo izdelali programsko kodo, ki nam podatke pretvarja v dejanske vrednosti kotov. Funkcijski blok oziroma Sub.vi *GYRO INIT.REG*, ki smo ga naredili, nam omogoča vnašanje oziroma vnos registra v program. Vrednost registrov smo našli v podatkovnem listu senzorja L3G4200D.



Slika 12: Instalacija registrov (vir: Avtor naloge).

V program smo dodali tudi tako imenovano *Shared Variable*, ki nam omogoča branje oziroma kontroliranje vrednosti v programu preko interneta, spletne strani ali mobilnega telefona.

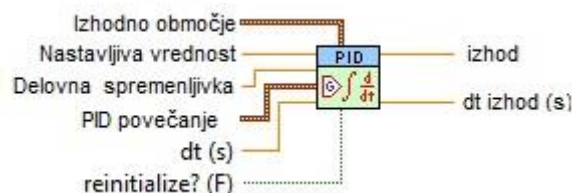


Slika 13: Shared Variable (vir: Avtor naloge).

Glavni del programa pa je bil program za kontroliranje enosmernih motorčkov s PID regulacijo. Ta program je v strukturi *time loop*, ki se izvaja nenehno pri vnaprej določeni frekvenci.



Najprej smo v program vstavili funkcijo PID, ki jo najdemo v *Functions > Control & Indicator > PID > PID.vi*, nato pa smo določimo nastavitveno točko *setpoint* in druge vrednosti, ki jih preberemo iz senzorjev.



Slika 14: Funkcijski blok za PID regulacijo (vir: Avtor naloge).

PID regulacija je zaprtozančno vodenje procesa, ki na osnovi primerjave dejanske in želene regulirane veličine na proces vpliva tako, da je napaka čim manjša. Pri sledilnem načinu delovanja mora regulacijski sistem zagotoviti, da regulirana veličina dobro sledi referenci, spremembi nastavitvene točke (npr. primerno odzivanje DC motorčka ob spremembah nastavitvene točke).

Regulacija je izvedena kot vsota proporcionalnega (P), integralnega (I) in diferencialnega (D) člena. Člen P je neposredno odvisen od trenutne velikosti napake, člen I je odvisen od integrala preteklih napak, člen D pa od hitrosti spreminjanja napake, s katerim lahko deloma predvidimo prihodnje delovanje sistema.

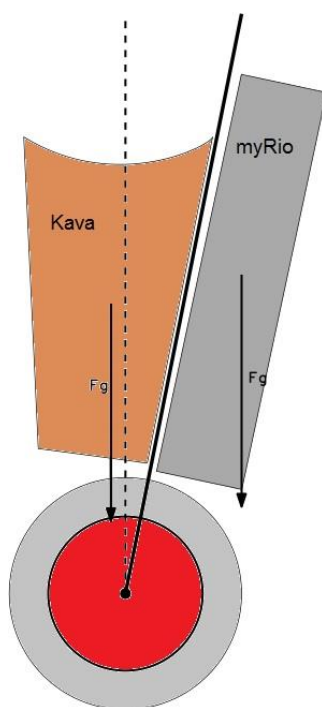
Regulator PID ima tri nastavljive parametre: ojačanje  $K_P$ , integralno časovno konstanto  $T_I$  in konstanto diferenciranja  $T_D$ . Referenčno veličino predstavlja nastavitvena vrednost (Set point), to je vrednost, ki jo želimo v našem procesu.

Za odčitavanje hitrosti vrtenja enosmernih motorčkov smo morali izdelati program za branje enkorderjev. Najprej smo vstavili funkcijo Encorder, ki smo jo našli v *Functions > myRio > Encorder.vi*. Dodali smo indikator, ki nam prikazuje število obratov in smer vrtenja motorčka. Za oboje smo uporabili *Shared Variable*, da lahko spremljamo podatke na računalniku ali tablici.



## 4.2 Načrtovanje

S programom Ipe smo narisali skico sil teže, ki vplivajo na stabilizacijo robota oziroma mizice. Naredili smo več skic iz katerih smo razbrali, da se nahaja težišče celotnega robota nižje, če je predmet (kava) nameščena vzporedno z modulom myRIO, kot pa če bi bila mizica s predmetom nameščena nad modulom myRIO. Odločili smo se, da bo mizica predstavljala nekakšen nahrbtnik. Pregledali smo tudi kote, ki se pojavljajo in jih potrebujemo za stabilizacijo robota. Najpomembnejši je naklon od vodoravne osi. Po vsem matematičnem in fizikalnem premisleku smo začeli na papir izdelovati razne skice, kako naj bi robot izgledal. Po več skicah smo se odločili, katera je prava in po ocenah najboljša.



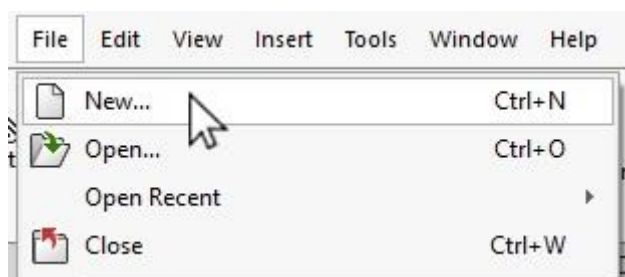
Slika 17: Skica sil (vir: Avtor naloge).

## 4.3 Izdelava 3D modela

Po končani skici ohišja smo v programu SolidWorks začeli izdelovati 3D model ohišja. SolidWorks je programski paket, ki nam omogoča izdelavo inženirskih analiz in 3D objektov. Glavni del programa je 3D modelirnik, v katerem lahko rišemo najrazličnejše

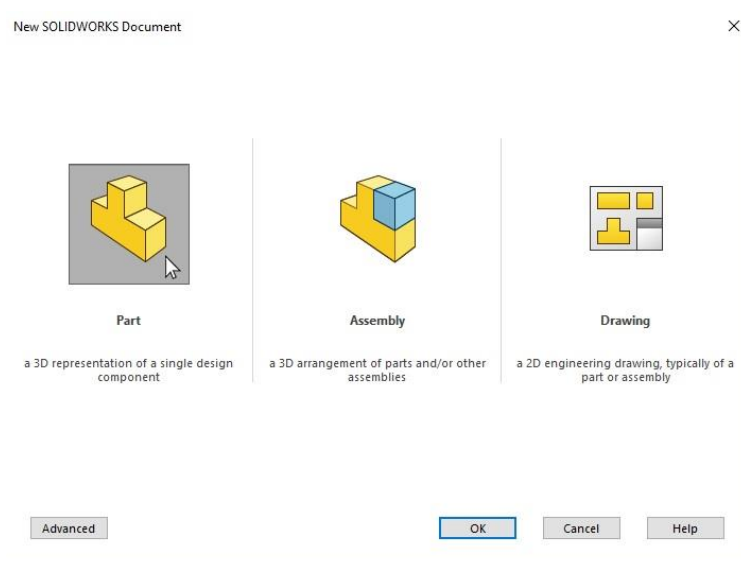
objekte, ima pa tudi druge module, kot so modul za sestavljanje, modul za izdelavo tehniške dokumentacije, modul za izdelavo električnih vezji in modul za simulacijo. Njegova uporaba je zelo priročna, zato se je uveljavil na več tehničnih področjih kot so lesarstvo, strojništvo, elektrotehnika itd. V programu je veliko funkcij, ki nam olajšajo delo.

Za izdelovanje novega modela moramo najprej ustvariti okno, v katerem bomo ustvarjali. To storimo tako, da kliknemo na ikono *File > New*.



Slika 18: Ustvarjanje novega okna. (vir: Avtor naloge).

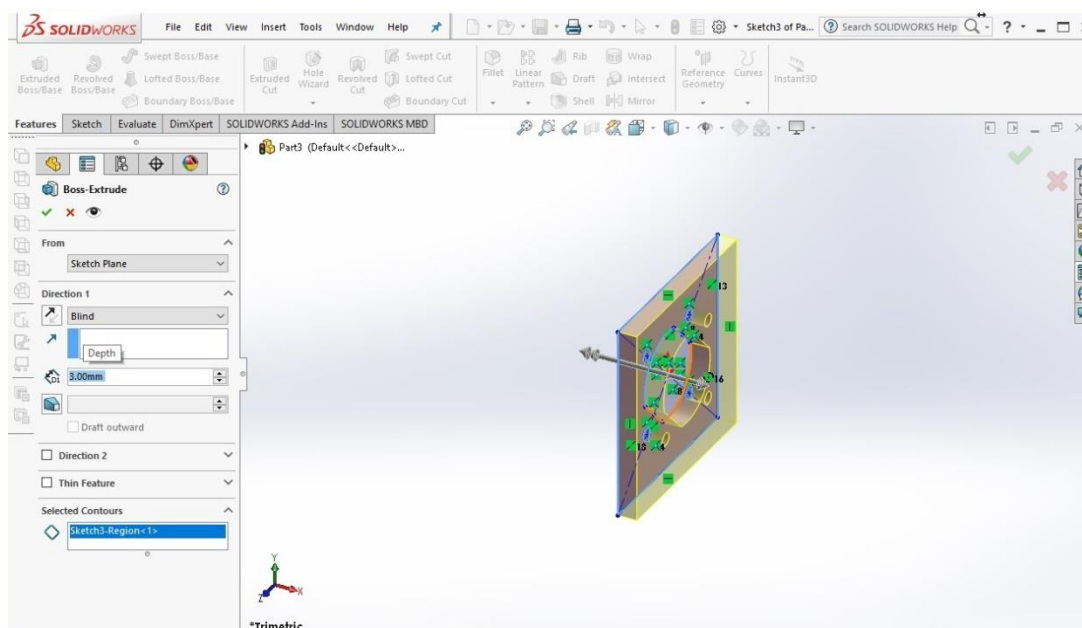
Odpre se okno, v katerem se odločimo ali bomo izdelovali novi model, sestavljali več modelov skupaj ali naredili 2D načrt modela. To storimo s preprostim klikom na ikono *Part (del)*.



Slika 19: Izbira. (vir: Avtor naloge).

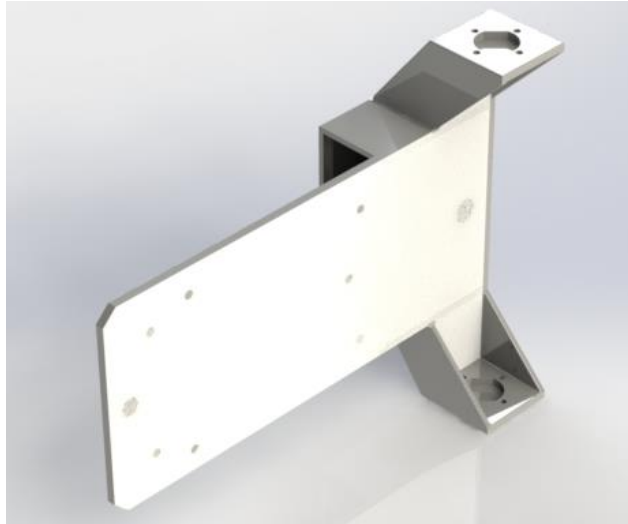
Za ustvarjanje modela si moramo najprej izbrati ploskev na kateri bomo začeli izdelovati model. Nato začnemo risati skico v 2D obliki, ki ji nato s pomočjo raznih orodji za izrezovanje, izobčevanje in ločevanje spremenimo v 3D obliko.

Tako smo s pomočjo teh navodil začeli načrtovati nosilce motorjev. Pomagali smo si s podatkovnim listom motorčka, v katerem so bile podane njegove mere. Ko smo končali risanje nosilcev, smo začeli izdelovati platformo, na katero so pritrjeni myRIO in podstavek (nahrbtnik) za prevažanje predmeta. Pri tem smo upoštevali omejitve tiskalnika, ki ne more natisniti izbočenih delov. Po končanem risanju smo model izvozil v format stl, ki ga podpira program 3D tiskalnika.



Slika 20: Risanje nosilcev za motorje (vir: Avtor naloge).

Za tiskanje smo uporabili šolski tiskalnik znamke Makerbot. Za pripravo na tiskanje moramo najprej v program Makerbot print vnesti model. Nato nastavimo debelino sloja, napolnjenost predelov z materialom, temperaturo šob itd. Po opravljenih nastavitvah si lahko ogledamo predogled tiskanja ali pa s pritiskom na gumb Print začnemo tiskati. Po tiskanju model lepo nežno odstranimo in model je končan.



*Slika 21: 3D model, pripravljen za tisk (vir: Avtor naloge).*

Ohišje smo načrtovali takšne oblike, da je modul myRIO pritrjen na sprednji strani ohišja, saj ima večjo površino in pripomore k izenačitvi bremena in krmilnika motorjev, ki je pritrjeno na hrbtne strani ohišja.

Pred tiskom smo v predogledu opazili, da tiskalnik 3D modela zaradi njegove velikosti ne more natisniti v celoti, zato smo morali skrajšati dolžino ohišja, na katerega je pritrjen modul myRIO in krmilnik motorja.

O programu SolidWorks, 3D risanju in tiskanju smo se naučil veliko novega, kar nam bo koristilo za naslednje raziskovalne naloge, hkrati bodo 3D modeliranja hitrejša in natančnejša.

#### **4.4 Povezovanje in preizkus**

Po končanem programiranju, risanju in izdelovanju ohišja je sledilo povezovanje vseh modulov v celoto. Pri povezavi smo si pomagali s programom *Multisim*, ki nam nudi risanje in virtualno preizkušanje vezij.

Multisim je interaktivni simulator, ki dovoljuje spremembe v načrtu tokokroga v realnem času. Ko simulacija teče, lahko opazujemo takojšnje spremembe prikazanih rezultatov. Tako smo vse skupaj povezali v celoto ter začeli preizkušati in dopolnjevati program, ko se je kje zataknilo in ni delovalo po predvidevanjih.

## 5 ZAKLJUČEK

V raziskovalni nalogi smo se naučili veliko novega o programiranju v okolju LabVIEW. Najprej smo začeli izdelovati preproste programe, s katerimi smo spoznavali delovanje posameznih senzorjev, nato pa smo prešli v zahtevnejše programe oziroma naloge. Ko smo izdelali že veliko različnih primerov, se nam je ob pogovoru o giroskopu utrnila ideja za izdelavo mizice, ki bi se lahko premikala na dveh kolesih. Pogledali smo si posnetke na spletu in začeli raziskovati, kako bi lahko takšno mizico stabilizirali. Prišli smo do spoznanja, da potrebujem giroskop in enosmerne motorčke, za katere moramo vsak trenutek poznati njihov položaj. Ko smo začeli izdelovati programsko kodo, nam je največ težav povzročala stabilizacija, saj integralnih izračunov nismo poznali. Z raziskovanjem smo našli rešitev, ki nam je ustrezala. Kljub velikim težavam se nismo predali in smo raziskovali naprej. Za nalogo smo potrebovali veliko časa, a nam ga ni žal, saj smo pridobili nova znanja, ki mi bo koristilo v prihodnjosti.

## 6 DRUŽBENA ODGOVORNOST

Naša raziskovalna naloga je lahko uporabna za zabavo. Lahko pa bi bila koristna ljudem, ki ne morejo hoditi in bi si lahko pripeljali čaj, kavo, zdravila ali druge potrebne predmete. Ljudem bi lahko tudi pomagala nositi razne predmete. Zaradi svoje ozke oblike na dveh kolesih pa bi prihranili prostor za shranjevanje. Nalogo bi lahko izboljšali do te mere, da bi mizica pomagala ljudem učiti hoditi ali pa bi jim nudila oporo pri premikanju.

## 7 VIRI

MyRIO Balancing Robot: <http://forums.ni.com/t5/myRIO-Balancing-Robot/gp-p/8523> (5. 12 2016).

NI myRio . <http://www.ni.com/myrio/> (3. 1 2017)

Pmod GYRO Reference: <https://reference.digilentinc.com> (20. 12 2016)

Podatkovni list I3g4200d :

<http://www.st.com/content/ccc/resource/technical/document/datasheet/04/46/d6/00/be/d9/46/ae/CD00265057.pdf/files/CD00265057.pdf/jcr:content/translations/en.CD00265057.pdf> (30. 12 2016).

Šola elektronike: <https://sites.google.com/site/solaelektronikesers/home/krmiljenje-led-diode-s-tipko>

Šola elektronike: <https://sites.google.com/site/solaelektronikesers/home/pid-regulacija> (10. 12 2016).