

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

**LabVIEW FPGA zasnovan sistem upravljanja kretanjem
holonomskog robota**

ZAVRŠNI RAD
I ciklusa studija

Mentor: Red. Prof. dr. Jasmin Velagić

Kandidat: Emina Hasanović

Sarajevo, 2017

Izjava o autentičnosti rada

Ime i prezime: Emina Hasanović

Naslov rada: LabVIEW FPGA zasnovan sistem upravljanja kretanjem holonomskog robota

Vrsta rada: Završni rad 1. ciklusa studija

Broj stranica: 53

Potvrđujem:

- da sam pročitala dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijски program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjesna univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačila prisustvo citiranog ili parafraziranog materijala;
- da sam dosljedno navela korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačila svaku pomoć koju sam dobila pored pomoći mentora i akademskih tutora;

Sarajevo, 28.9.2017.

Potpis: _____

Sadržaj

Sažetak	5
Postavka rada	6
1. Opis problema	7
1.1. Motivacija za rad.....	7
1.2. Struktura rada.....	7
1.3. Pregled područja istraživanja	7
1.4. Metodologija izrade rada	8
2. Sklopovska realizacija sistema.....	9
2.1. Uvod.....	9
2.2. FPGA	9
2.2.1. Pojediniosti kartice.....	11
2.2.2. Proizvođači i softverska podrška za FPGA.....	12
2.3. NI sbRIO-9363.....	12
2.3.1. Osnovne karakteristike kartice.....	12
2.3.2. Raspored portova	14
2.4. Zaključak.....	16
3. Softverska podrška.....	17
3.1. Uvod.....	17
3.2. LabVIEW	17
3.3. Instalacija kartice	18
3.3.1. Korištenje Ethernet kabl.....	18
3.3.2. Konfigurisanje statičke IP adrese.....	18
3.4. Zaključak.....	23
4. Sistem za lokomociju	24
4.1. Uvod.....	24
4.2. Koračni motori	24
4.2.1. Osobine i podjela koračnih motora	25
4.2.2. Hibridni i bipolarni korčni motori.....	26
4.3. Pogon HOLBOS-a	27
4.3.1. "Mecanum Wheel" točkovi	27
4.3.2. "Quadstepper 4 Stepper Motor Driver Board"	29

4.4.	Zaključak.....	32
5.	Implementacija sistema za upravljanje kretanjem robota	33
5.1.	Uvod.....	33
5.2.	Konvertor	33
5.3.	Projekat u LabVIEW-u	34
5.4.	Upravljanje jednim koračnim motorom.....	38
5.5.	Program za upravljanje HOLBOS-om.....	48
5.6.	Zaključak.....	50
6.	Zaključno poglavlje	51
7.	Literatura.....	53

Sažetak

U ovom radu razvit ćemo FPGA zasnovan sistem upravljanja kretanjem holonomnim robotom HOLBOS. Upoznat ćemo se s radom jedne nove tehnologije, FPGA, Real-Time simulacijom i programskim paketom LabVIEW. Također, izučit ćemo i rad koračnih motora, prednosti, mane, vrste koračnih motora, upotrebu i načine kojima možemo da upravljamo njima. Zadatak je da upravljamo pogonskim mehanizmom robota, tj. sa svim njegovim točkovima istovremeno.

Abstract

The aim of this paper is to create a control system for holonomic robot called Holbos. We will learn about new technology, FPGA, Real-Time simulation and software package LabVIEW. Moreover, we will analyse the working principles of stepper motors, the advantages, their drawbacks, different types, their uses and ways in which we can control them. The final goal is controlling the driving mechanism of the robot, that is, controlling all of its wheels simultaneously.

Mentor: Prof. dr. Jasmin Velagić

Postavka rada

Naslov: LabVIEW FPGA zasnovan sistem upravljanja kretanjem holonomskog robota

Kurs: Mehatronika

Student: Emina Hasanović

Cilj: Implementacija sistema upravljanja kretanjem holonomskog robota putem LabVIEW i FPGA target okruženja korištenjem Spartan-3E FPGA Xilinx i RIO evaluacijskog kita.

Opis: Potrebno je izvršiti implementaciju sistema upravljanja robotom i sve simulacijski dobivene podatke demonstrirati na realnom modelu.

Očekivani rezultati: Demonstracija funkcionalnosti sistema upravljanja kretanjem robota zasnovanog na razvojnoj platformi NI kartice Spartan-3E FPGA Xilinx i RIO evaluacijskog kita i LabVIEW programskog okruženja.

1. Opis problema

Rad se bavi definisanjem problema i implementiranjem aplikacije za upravljanje kretanjem holonomskog robota. Upravljanje se obavlja pomoću LabVIEW programskog okruženja zasnovanog na FPGA metodologiji programiranja. FPGA sklopovi u današnjoj hijerarhiji predstavljaju vrhunac tehnologije, ukoliko se u vidu ima fleksibilnost sistema. Pružaju vrlo visok stepen slobode pri izradi složenih sistema, te jednostavnu ponovljivost i mogućnost brze izmjene dizajna. U nastavku će biti data definicija i postavka problema koju ovaj rad rješava. Potom će biti obrazložena metodologija izrade rada, te naposljetku data struktura rada i prokomentarisani rezultati dobiveni testiranjem.

1.1. Motivacija za rad

Rad je nastao kao posljedica stečenih znanja u toku preddiplomskog studija, te i interesovanja autora u ovom području automatike i najnovijih tehnologija. Cilj je bio ispraviti propuste koji su napravljeni u dosadašnjem obrazovanju i upotpuniti stečeno znanje. Većina rada se temeljila upravo na istraživanju do tada nepoznatih oblasti. To je omogućilo da prihvatimo i usvojimo nove načine rada i utvrdimo one stečene do sada.

1.2. Struktura rada

U prvom dijelu rada navedeni su opšti uslovi nastanka FPGA tehnologije, te razvoj i primjena iste, dalje se govori i o arhitekturi FPGA, NI Single-Board RIO-9636 FPGA baziranom uređaju, te o LabVIEW, tj. korištenom softveru. Poslije toga je opisan pogonski sistem korištenog robota, te karakteristike motora i točkova koji pogone robota. U nastavku je obrađeno razvojno okruženje National Instruments-a pod nazivom LabVIEW. Na kraju, zadnje poglavlje, govori o rezultatima urađenog i analizira se samo rješenje problema i koliko je efikasno.

1.3. Pregled područja istraživanja

Prvi programabilni logički kontroleri i uređaji (PLC i PLD) [1] zajedno sa PROM memorijom, kao preteče FPGA logičkih sklopova, razvili su se uz pomoć inženjera *General Motors-a* 1968 godine (međutim stvarna realizacija se dogodila u osamdesetim godinama i to prvobitno od *Ross Freeman-a* 1984 godine a kasnije od *Steve Casselman-a* 1992 godine) kada se pokušala pronaći alternativna zamjena za složene relejne kontrolne sisteme.

Tadašnji kontrolni sistem morao je zadovoljiti sljedeće zahtjeve:

- jednostavnije programiranje
- programske izmjene bez sistemskih intervencija
- manje, jeftinije i pouzdanije komponente kod relejnih kontrolnih sistema
- jednostavno i jeftino održavanje

Postepenim razvojem došlo se do sistema koji je omogućio jednostavnije povezivanje binarnih sistema. Uvjeti pod kojim su ovi signali trebali biti povezani, bili su određeni u kontrolnom programu, s novim sistemima bilo je po prvi put moguće očitavanje signala na ekranu, te njihovo pohranjivanje u vidu datoteka u elektronskoj memoriji.

Nasuprot pomenutim tehnologijama, PLD (Programmable logic device) sadrži mnogo manji broj logičkih kola, pa su njegove mogućnosti znatno skromnije od onih koje pružaju ASIC i ASSP. Sa druge strane, funkcija ovakvih kola se ne određuje fabrički, već se ova kola mogu reprogramirati veliki broj puta. Također, vrijeme izrade je znatno manje, pa su njihove cijene prihvatljivije. FPGA tehnologija je zadržala dobre osobine i jednog i drugog navedenog pristupa.[2]

Ovakav dizajn sadrži veliki broj logičkih kola što mu daje mogućnost realizacije veoma složenih funkcija (što bi bila glavna odlika ASIC-a), a opet, funkcija ovakvih kola se ne konfigurišu fabrički, već se skoro neograničen broj puta mogu reprogramirati, kako je to ranije rečeno „*in the led*“. Ovakav pristup značajno smanjuje vrijeme izrade, kao i troškove proizvodnje, što je učinilo FPGA tehnologiju vrlo popularnom. Kada su se na tržištu pojavili prvi FPGA dizajni, oni su bili prilično skromnih mogućnosti. Ta njihova "skromnost" nije davala neke velike mogućnosti za praktičnu primjenu, pa se FPGA koristio uglavom za realizaciju nekih jednostavnih funkcija. Međutim daljnim razvojem FPGA načina programiranja, ovaj tip je postao sve prihvatljiviji kod korisnika.

1.4. Metodologija izrade rada

Potrebno je prvo definisati sistem kojim se upravlja, način na koji se njime upravljalo, kao i bliže objasniti programsko okruženje u kojem ćemo raditi. U okviru razvoja i implementacije, za dati sistem upravljanja je prije svega potrebno definisati veličinu koju je potrebno regulisati. Nakon toga se vrši odabir mjernog sistema, s ciljem prikupljanja relativnih informacija o stvarnoj vrijednosti veličine. Povezivanje elemenata i integracija sistema sama

po sebi je kompleksan zadatak koji zahtijeva dobro poznavanje programskog okruženja u kojem se obavlja implementacija upravljačkog algoritma, kao i hardverske strukture i njenih karakteristika. Za upoznavanje s postavljenim problemom iskorištena je literatura navedena u nastavku izlaganja kao i sve prednosti koje nudi internet računarska mreža. Njen brz razvoj i širenje nam otvara mogućnosti za realizaciju mjernih sistema za nadzor i upravljanje. Kod dizajna upravljačkog sistema potrebno je prvo da se omogući istovremeno kretanje sva četiri koračna motora. Sistem mora biti sposoban da u najkraćem mogućem vremenu pošalje upravljačke signale, što se odnosi na složenost programa koji se spušta na FPGA platformu i to sve treba biti u skladu sa performansama robota. Upravljanje mora biti jednostavno i pristupačno korisniku, odnosno kreiranje grafičkog interfejsa koji je pouzdan i jednostavan za korištenje. Prvo je slijedilo upoznavanje sa radom NI Single-Board RIO-9636 FPGA uređaja, zatim sa svim ostalim stvarima kako je rad na projektu odmicao. Prije samog testiranja programa koji je razvijen u LabVIEW-u na stvarnom sistemu, robotu, izvršena je samo simulacija, što je neki vid validacije i verifikacije. Rezultati će biti prikazani u narednim poglavljima.

2. Sklopovska realizacija sistema

2.1. Uvod

U ovom poglavlju će biti definisan svaki od hardverskih dijelova koji smo koristili pri izradi rada, njihova struktura, nominalni naponi i struje i načini povezivanja sa ostatkom sistema. U ovom poglavlju bit će detaljno opisana FPGA arhitektura i struktura, te način funkcionisanja na osnovu gradivnih jedinica FPGA. Dat je poseban osvrt na korišteni uređaj pri radu.

2.2. FPGA

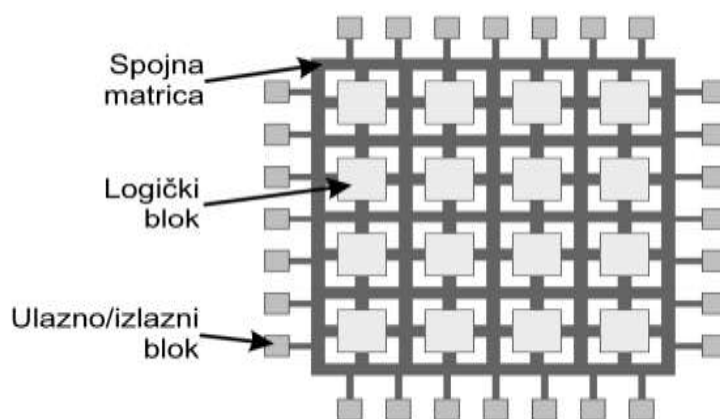
Do sada smo već spomenuli da kartica NI sb-RIO 9636 temelji na FPGA čipu (*slika 2.1.*). FPGA (*Field Programmable Gate Array*) predstavlja integrisano kolo projektovano tako da se njegova unutrašnja struktura može definisati od strane korisnika. Unutrašnju strukturu FPGA komponente korisnik može definisati pomoću HDL-a (*Hardware Description Language*) ili šematskih dijagrama. Slično gejtovskom polju, FPGA čini veliki broj identičnih logičkih ćelija koje se naknadno povezuju kako bi se ostvarila željena funkcija. Od trenutka otkrića i početka razvoja ovog čipa počela je revolucija samog načina programiranja i komunikacije između host računara i kartice. FPGA je jednostavan za upotrebu i

konfiguraciju i obzirom na to je naišao na veliko prihvatanje od strane korisnika. Njegova upotreba se posebno uočava ukoliko želimo da imamo mogućnost modifikacije i brze izvedbe prototipova.



Slika 2.1. Izgled FPGA čipa

Iako svaki proizvođač FPGA kola koristi neku svoju specifičnu FPGA arhitekturu, sve one, u osnovi, su varijacija ahitekture prikazane na slici 2.2. Arhitekturu čine: (a) konfigurabilni logički blokovi, raspoređeni u dvodimenziono polje; (b) U/I blokovi, raspoređeni po obodu kola i (c) programabilna sprežna mreža smeštena u kanalima između logičkih blokova. Logički blokovi služe za realizaciju logičkih funkcija, sprežna mreža omogućava povezivanje logičkih blokova u cilju kreiranja složenijih funkcija, dok se putem U/I blokova ostvaruje sprema internih resursa sa pinovima kola. Dodatno, FPGA može sadržati i različite specijalizovane logičke resurse, kao što su ALU jedinice, RAM memorija, dekoderi i dr. Kod FPGA koriste se SRAM i antifuse tehnologije programiranja



Slika 2.2. Arhitektura FPGA čipa

FPGA se može programirati na tri različita nivoa: (1) na nivou funkcije logičkih blokova; (2) na nivou sprege logičkih blokova; i (3) na nivou ulaza i izlaza. Sva tri nivoa se konfiguriraju

preko niza bitova koji se preko eksternog izvora pune u kolo. U zavisnosti od načina konfiguracije, razlikujemo četiri tipa FPGA kola. Konfigurabilna kola (prvi tip) se mogu konfigurirati od strane korisnika samo jedanput nakon čega dodatne promjene konfiguracije nisu izvodljive. (Iz ovog razloga ova kola se nazivaju one-time programmable, ili OTP). FPGA zasnovani na antifuse tehnologiji pripadaju ovoj kategoriji. Rekonfigurabilna FPGA kola su zasnovana na SRAM tehnologiji i mogu se konfigurirati proizvoljan broj puta. Dije se na dve klase: statička i dinamička. Kod statičkih kola (drugi tip) konfiguracioni niz se puni jedanput, na početku rada, i dok je kolo operativno ne mijenja se. Tek nakon resetovanja sistema u FPGA je moguće napuniti novi niz. Kod dinamički rekonfigurabilnih kola moguće je vršiti promene u konfiguraciji u bilo kom trenutku rada kola. Dinamički rekonfigurabilna kola se dalje mogu podijeliti na potpuno rekonfigurabilna (treći tip), pri svakom novom punjenju rekonfiguriše se cijelo kolo, i parcijalno rekonfigurabilna (četvrti tip) moguće je rekonfiguracija samo jednog dijela kola. Svi sinhroni elementi unutar FPGA, kao što su registri konfigurirani kao flip-flopovi, pobuđuju se taktnim, ili clock, signalom. Taktni signal potiče iz spoljnog svijeta, unosi se u FPGA kroz poseban ulazni pin, a zatim razvodi i povezuje sa odgovarajućim registrima.[3,4,5]

2.2.1. Pojediniosti kartice

Ovaj model kartice pored standardnih analognih i digitalnih ulaza/izlaza na svom gornjem dijelu sadrži neke jako korisne dijelove. Te dijelove može svrstati u dvije grupe i to dijelove koji služe za:

- mjerenje
- kontrolu i očitavanje

U dijelove koji su namijenjeni za mjerenje spadaju:

- sistem za mjerenje temperature
- Sistem za mjerenje napona
- Enkoderski signal za motore
- HMI dugmad

U dijelove koji su namijenjeni za kontrolu i očitavanje ubrajamo:

- LCD ekran

2.2.2. Proizvođači i softverska podrška za FPGA

Najveći proizvođači FPGA su **Xilinx** i **Altera** koji zajedno u prosjeku drže oko 70% svjetskog tržišta, mada Xilinx i dalje ima malu prednost. Pored njih manji proizvođači FPGA su i **Lattice Semiconductor**, **SiliconBlue Technologies**, **Acronics** i drugi.



Slika 2.3. Izgled kartice NI sb-RIO 9636

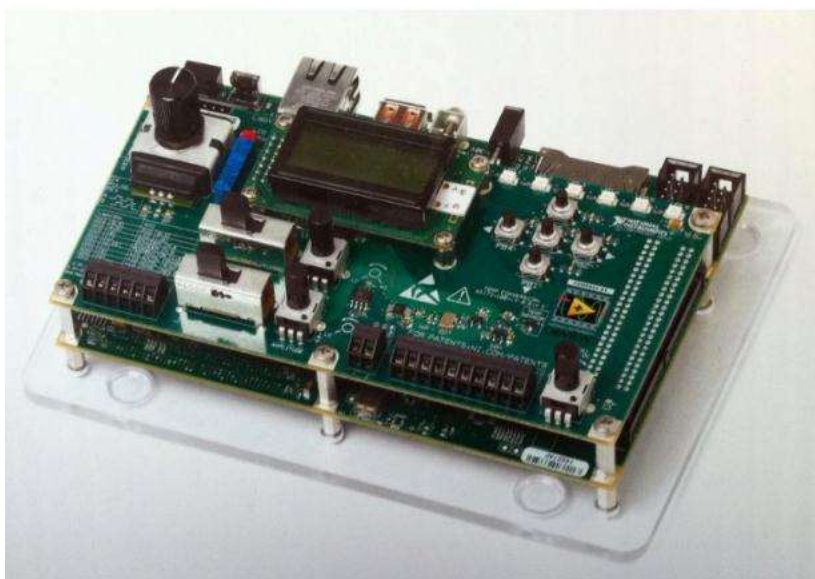
Razvojem FPGA čipova i razvojnih integrisanih pločica na kojima se čipovi nalaze, kao i njihova sve veća primjena, uslovlila je da pojedini proizvođači softvera u svoje alate dodaju i opcije za programiranje FPGA čipova. Tako naprimjer, tako jedan od najrasprostranjenijih softvera koji se koristi u inženjerskim simulacijama, *Matlab*, ima podršku za konfigurisanje šeme logičkih elemenata koju je moguće prenijeti na FPGA, direktno iz Matlab okruženja. Pored *Matlab-a*, podršku za FPGA ima i softver firme *National Instruments*, *LabVIEW*, u okviru svog dodatnog paketa (toolkit) pod imenom *FPGA Toolkit*. [6]

2.3. NI sbRIO-9363

2.3.1. Osnovne karakteristike kartice

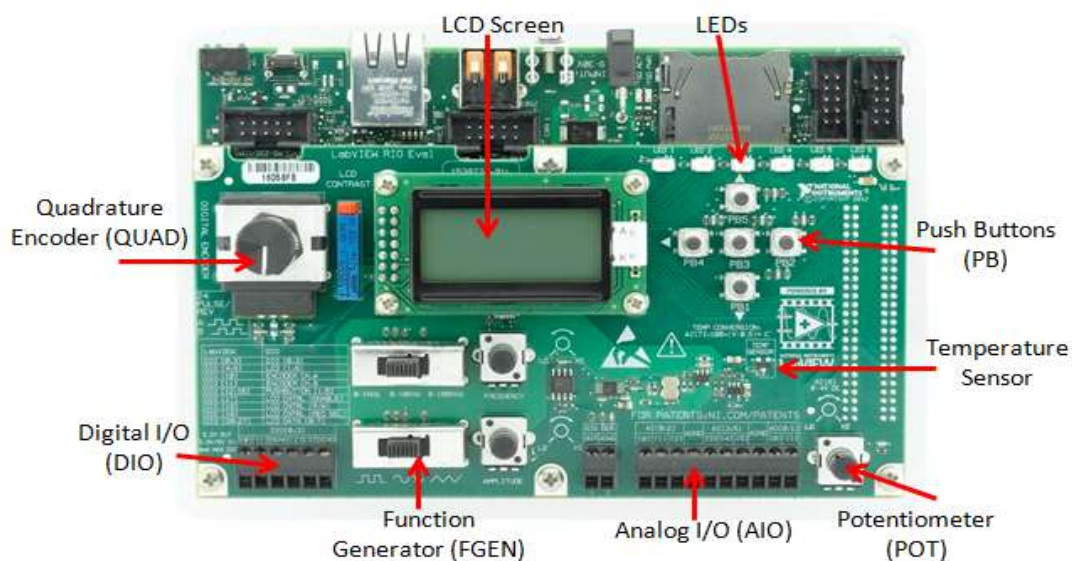
Kartica koja je korištena je NI sb-RIO 9636. (slika 2.4.) Ovaj model kartice posjeduje procesor 400 MHz, 512 MB memorije i 256 MB DRAM-a. Pored toga na njoj se nalazi 16 16-bitnih analognih ulaza koji idu do maksimalnog napona ± 10 V, 4 16-bitna analogna izlaza i 28 digitalnih ulaza/izlaza sa nominalnim naponom 3.3 V, ugrađena periferija kao što je integrirani 10/100BASE Ethernet, RS232 serijski port, RS485 serijski port, USB, CAN i

SHDC portovi, od 9 do 30V istosmjerni napon napajanja. Princip programiranja se temelji na *Real-Time* programiranju i FPGA čipu.[7]



Slika 2.4. NI sbRIO9363 sa razvojnom pločom

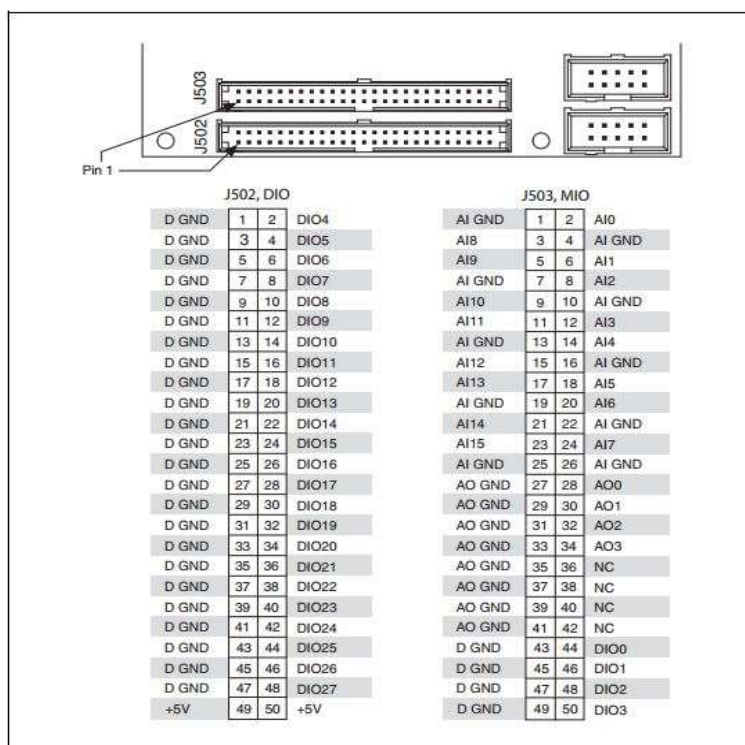
Ove kartice imaju mogućnost dodavanja specijaliziranih periferija od strane korisnika. SbRIO ima moćan floating-point procesor, rekonfigurabilnim FPGA i modularnim I/O. Sve hardverske komponente se programiraju pomoću LabVIEW-a kako bi se dala mogućnost da brzo projektuju aplikacije bez poznavanja mašinskog jezika. Pored ove kartice pored nje se nalazi još jedna i to modela RIO evaluation kit (*slika 2.5.*). Ovaj tip kartice je predviđen za programiranje u LabVIEW, i ona je sve što je potrebno kako bi se iskusilo NI grafičko okruženje i dizajnerski pristup ugrađenim sistemima. Ovaj pristup kombinuje LabVIEW softverski dizajn sistem i standardne FPGA bazirane platforme kako bi se smanjilo vrijeme potrebno za kontrolu sistema.[8,9]



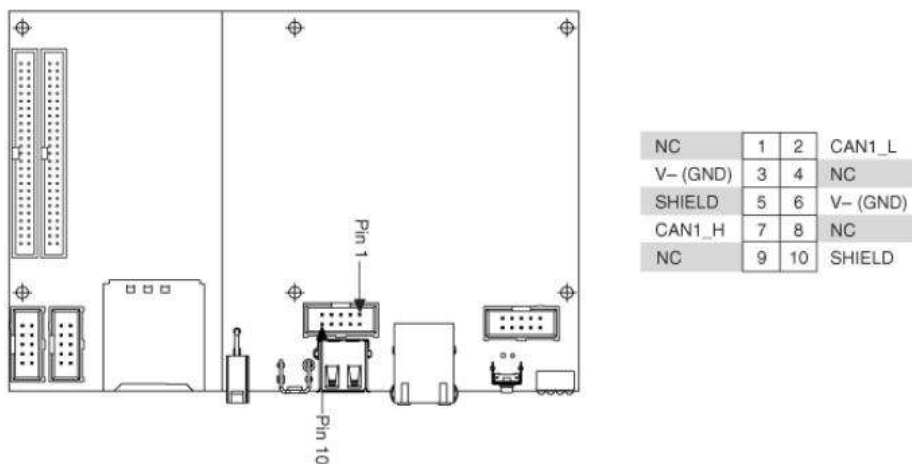
Slika 2.5. Izgled gornje kartice (RIO evaluacijski kit)

2.3.2. Raspored portova

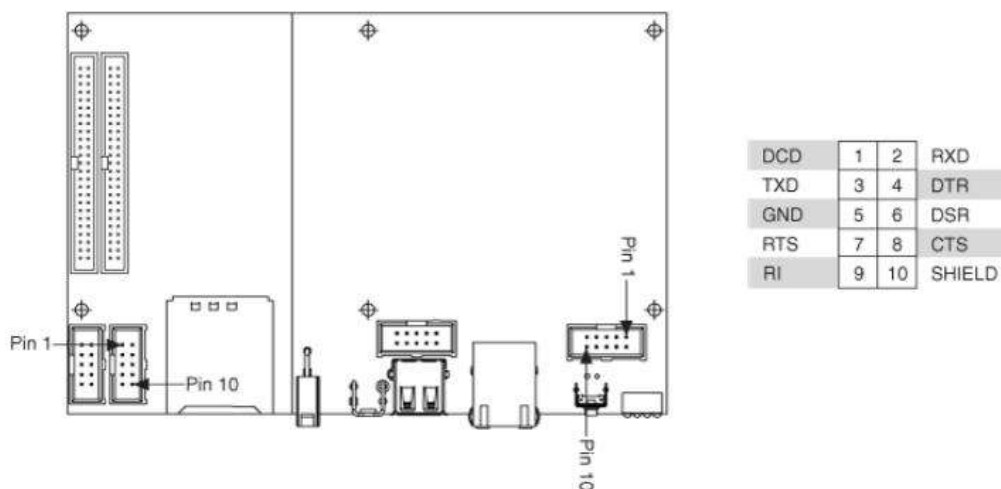
Na kartici se nalazi 16 16-bitnih analognih ulaza koji idu do maksimalnog napona ± 10 V, 4 16-bitna analogna izlaza i 28 digitalnih ulaza/izlaza sa nominalnim naponom 3.3 V, ugrađena periferija kao što je integrirani 10/100BASE Ethernet, RS232 serijski port, RS485 serijski port, USB, CAN i SHDC portovi, od 9 do 30V istosmjerni napon napajanja. Sljedeće slike pokazuju raspored portova, kao i mjesto na kartici gdje se nalaze izvodi.[10]



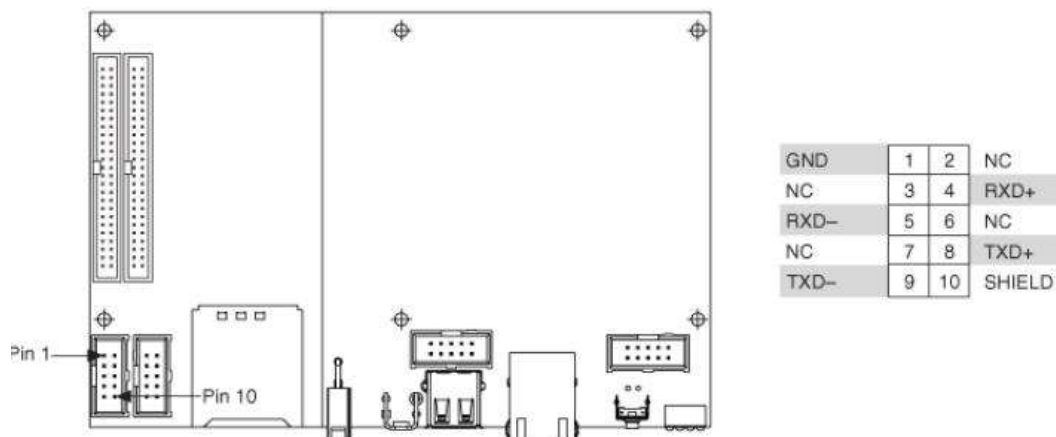
Slika 2.6. Pinout MIO i DIO konektora



Slika 2.7. Raspored pinova za konektor W500, CAN Port



Slika 2.8. Raspored pinova za konektore W501 i W503, RS232 serijski port



Slika 2.9. Raspored pinova za konektore W502, RS485 serijski port

2.4. Zaključak

Prvi korak ka rješenju postavljenog problema jeste upoznavanje s radom FPGA kartice i njenog razvojnog okruženja, što je i urađeno u ovom poglavlju broj 2. Istraživanje rada ove kartice nastavljeno je i prilikom rada na sistemu. Kartica je relativno nedovoljno istražena i nije bilo prilike da se tokom studija upoznaju karakteristike ovog moćnog alata.

3. Softverska podrška

3.1. Uvod

Budući da je rad zasnovan na kartici Spartan-3E FPGA Xilinx i RIO evaluacijskom kitu programsko okruženje koje treba da koristimo u ovom slučaju je LabVIEW. Kako kartica posjeduje FPGA čip, programiranje na njoj zahtijeva FPGA Target. Iz svega navedenog možemo zaključiti da LabVIEW nije dovoljan kako bismo ostvarili svoje ciljeve. Iz tih razloga potrebno je pored LabVIEW upotrijebiti još nekoliko određenih modula koji služe kao nadogradnja. Moduli koji se koriste u ovom slučaju su:

- *Real-Time 2014*
- *Xilinx 14.7 2014*
- *FPGA module 2014*

Prije samog razvoje aplikacije potrebno je prvo instalirati LabVIEW pa zatim već navedene module. Nakon uspješne instalacije možemo pristupiti razvoju prvog programa.

3.2. LabVIEW

LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*) (slika 3.1.) je platforma i razvojno okruženje za vizuelne programske jezike iz *National Instruments-a*. LabVIEW se obično koristi za prikupljanje podataka, kontrolu instrumenata i u industrijskom automatici na različitim platformama. Dozvoljava brzu ugradnju kompjuterskog prikupljanja podataka i sistema za njihovu analizu koji mogu biti široko prilagođeni potrebama korisnika paketa. Ovaj programski paket se znatno razlikuje od paketa s kojima se sreo u dosadašnjem radu.[11]



Slika 3.1. Početni prozor LabVIEW-a

LabVIEW je upotpunosti grafički program koji izgleda poput crteža elektronske šeme s jedne strane, i električnih mjernih uređaja s druge strane. To je Block dijagram i front panel. LabVIEW je hijerarhijski program, tj. da se bilo koji virtualni (VI) koji se programira može pretvoriti u modul koji može biti potprogram drugog VI. To je analogno konceptu programiranja kod konvencionalnih programskih jezika. Dizajniran je da se može proširivati dodavanjem odgovarajućeg modula. Proizvođač mjernog uređaja ili kartice može dodati modul u LabVIEW koji predstavlja karticu i sve njene funkcionalnosti.

3.3. Instalacija kartice

3.3.1. Korištenje Ethernet kabla

Najlakši način za spajanje RIO evaluacijskog hardvera (vašeg Real-Time Targeta) je preko lokalne mreže sa *DHCP serverom*. Ako ste povezani na internet, onda vjerovatno postoji i *DHCP server* koji omogućava automatsko pribavljanje *IP adrese* kako bi se ostvarilo samo povezivanje.

Ukoliko ovo opisuje vašu internet vezu, jednostavno iskoristite uključeni ethernet kabl kako biste povezali target sa ruterom ili prekidač na istu internet vezu kao što je povezan Windows na PC-ju. Ovo će omogućiti targetu da automatski primi IP adresu vašeg DHCP servera. Ukoliko ovo ne opisuje vašu internet vezu pogledajte sljedeću sekciju.

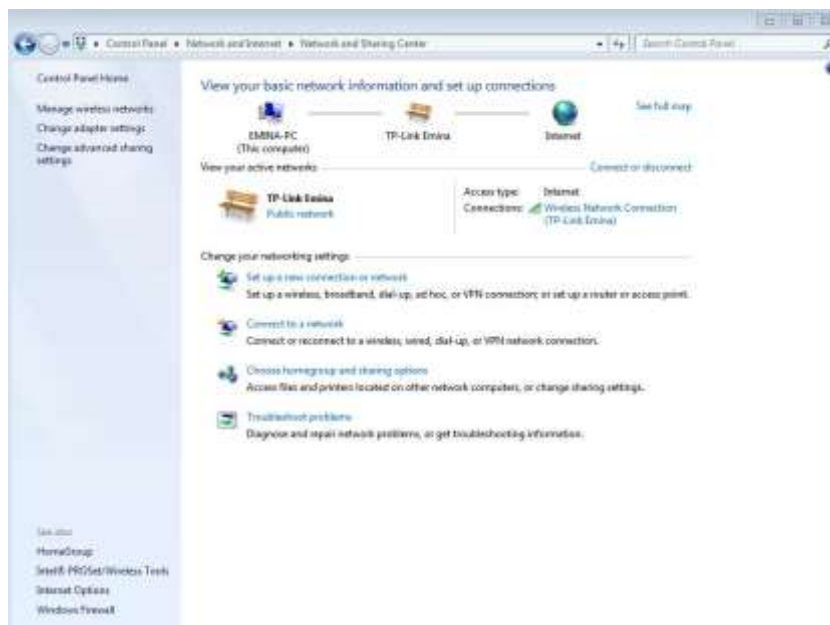
3.3.2. Konfigurisanje statičke IP adrese

Ukoliko je *DHCP konekcija*, opisana u prethodnom paragrafu, nedostupna, možete vaš *RIO evaluacijski* hardver direktno povezati sa vašim kompjuterom i pratiti korake koji će biti

opisani kako biste ručno podesili vašu IP adresu. Ukoliko vaš PC ima samo jedan ethernet kabel, ovo će zahtijevati da diskonektujete kompjuter sa bilo koje internet veze na koju je bio povezan.

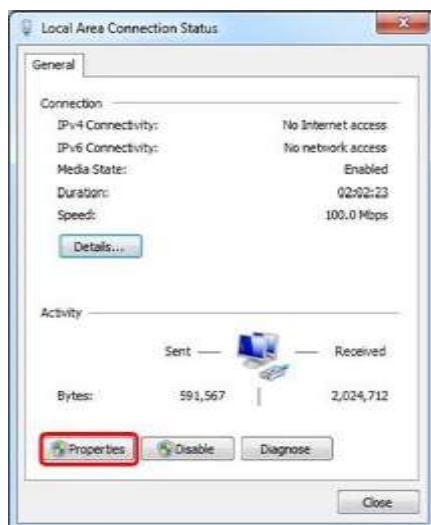
Jednom kada je RIO evaluacijski hardver spojen na PC, uključite napajanje i pratite sljedeće korake kako biste konfigurisali IP adresu:

- 1) Otvorite Control Panel
 - Windows XP/Visita/Windows 7 start meni: Start->Control Panel
- 2) Pregledajte Network Connections
 - Windows XP: odaberite Network Connections
 - Windows Vista/7: odaberite Network and Sharing Center
- 3) Otvorite Local Area Connection Properties (*slika 3.2.*)



Slika 3.2. Prozor Local Area Connection Properties

- Windows XP: Desni klik Local Area Connection i odaberite Properties
- Windows Vista/7: Local Area Connection, i onda odaberite Properties
- Windows 8: Odaberite Ethernet konekcija i odaberite Properties



Slika 3.3. Prozor Local Area Connection Status

4) Otvorite Internet Protocol (TCP/IP) Properties

- Windows XP: Odaberite Internet Protocol (TCP/IP) iz liste, i onda odaberite Properties
- Windows Vista/7/8: Odaberite Internet Protocol Version 4 (TCP/IPv4) iz liste, i onda odaberite Properties



Slika 3.4. Prozor Local Area Connection Properties

Ukoliko ovo ne uspije sa ovim odabirom, možda ćete morati konfigurirati Internet Protocol Version 6 (TCP/IPv6) umjesto ovog.

5) Konfigurirajte Internet Protocol (TCP/IP) Properties

- Windows XP/Vista/7/8: Odaberite Use following IP address: i unesite sljedeće vrijednosti za konfigurisanje (ostalo ostavite prazno):

IP address: 10.0.0.1

Subnet mask: 255 255 255 0



Slika 3.5.Prozor Internet Protocol (TCP/IP) Properties

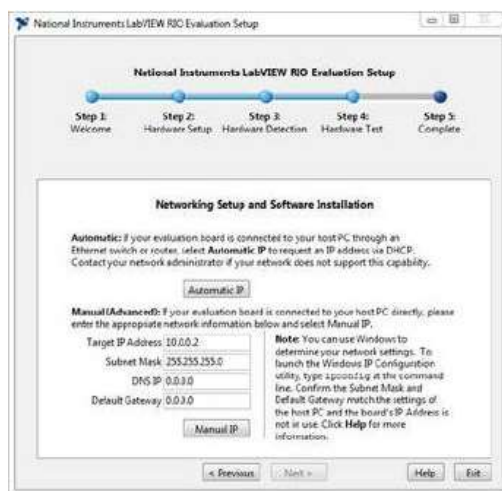
- 6) Odaberite OK i zatvorite Local Area Connection Properties (Windows 8: Ethernet Status) prozor
- 7) Nastavite instalacioni proces opisan ovdje i u printanom upustvu dobivenom uz karticu. Pokrenite LabVIEW RIO Evaluation Kit Setup Utility iz Start-> All Programs-> National Instruments-> labVIEW RIO Evaluation Kit
- 8) Dok koristite National Instruments RIO Evaluation Setup Utility na koraku 7 trebali biste biti u mogućnosti da konfigurirate postavke targeta. U Manual sekciji, unesite IP adresu targeta i Subnet Mask kao što je prikazano na slici i pritisnite Manual IP

IP address: 10.0.0.2

Subnet mask: 255 255 255 0

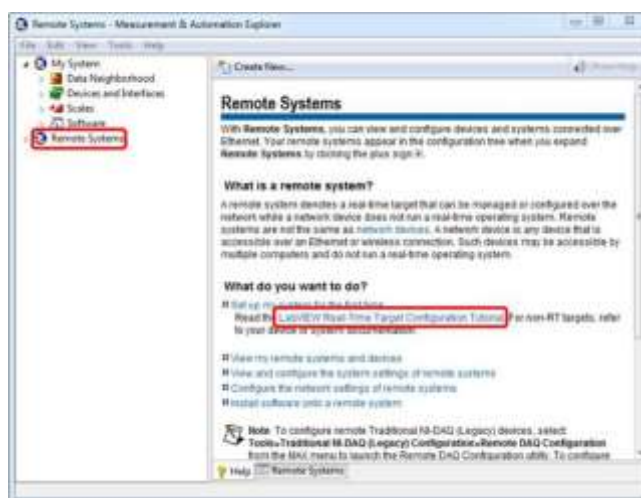
DNS IP: 0. 0. 0. 0.

Default GateWay: 0. 0. 0. 0.



Slika 3.6. Prozor National Instruments LabVIEW RIO Evaluation Setup

- 9) Target sada treba da ima prihvatljivu IP adresu, dopuštajući vam da nastavite kroz Setup Utility
- 10) Ako imate bilo kakvih problema sa Setup Utility i IP adresom nju također možete konfigurisati preko NI Max (Start->All Programs -> National Instruments->NI Max)
- Pritisnite na Remote Systems
 - U Help tabu na desnoj strani ekrana, kliknite na Set up my System for the first time i otvorite LabVIEW Real-Time target Configuration Tutorial kako biste konfigurisali postavke vaše veze



Slika 3.7. Prozor Remote System Mesaurmetn and Automation Explorer

Nakon instalacije LabVIEW i zahtijevanih modula treba restartovati kompjuter. nakon restartovanja program National Instruments LabVIEW RIO Evaluation Setup bi trebao automatski pokrenuti. Ovaj program bi vam trebao omogućiti da verifikujete da je RIO evaluation hardver postavljen ispravno i da je omogućeno da se koristi. Nakon pokretanje trebat ćete pritisnuti nekoliko puta Next dugme i proći kroz nekoliko koraka kako biste bili u mogućnosti koristiti program.

Ukoliko se program ne pojavi automatski nakon restartovanja, trebali biste biti u mogućnosti da ga pokrenete iz Start->All Programs->National Instruments->labVIEW RIO Evaluation Kit ->Setup Utility

Nakon svih ovih koraka koje ste obavili trebali biste biti u mogućnosti da koristiti LabVIEW sa svim svojim pratećim modulima i RIO Evaluacijskim hardverom.[12]

3.4. Zaključak

U ovom poglavlju je opisan način instalacije kartice na računar kako bi se pristupilo onda njenom programiranju. Instalacija je relativno jednostavna i nije bilo poteškoća u ovom dijelu zadatka, pa se moglo brzo pristupiti upozvanju kartice nakon instalacije i upoznavanju njenih modula u LabVIEW-u.

4. Sistem za lokomociju

4.1. Uvod

Lokomocija je proces u kome se pod djelovanjem sila pomjera robot, mobilna platforma ili neko drugo vozilo. Postoje različite strategije kretanja koje se mogu primijeniti na mobilne robote, ali izbor strategije zavisi od primjene mobilnog robota i o njegovoj okolini. Mobilni robot koji je korišten kao objekat upravljanja spada u grupu holonomskih ili svesmjernih robota jer se može kretati u bilo kom pravcu bez posebno definisanog manevra. Brojne su prednosti koje ovakav model kretanja daje, a neke su: veći izbor pravaca kretanja, lakši i brži dolazak do cilja, ušteda energije dok je nedostatak cijena samih točkova zbog specifičnosti njihove mehaničke izrade. Točkove korištenog robota, HOLBOS, pokreću koračni motori, korišten je i drajver za koračne motore.

4.2. Koračni motori

Koračni motori (*slika 4.1.*) su elektromehanički pretvornici energije, koji pulsnu električnu pobudu pretvaraju u koračni mehanički pomak (rotacijski ili translacijski). Koračna brzina kod komercijalnih motora kreće se od 100 do preko 10000 koraka u sekundi. Na malim koračnim brzinama rotor se zaustavlja na svakom koračnom položaju. Na srednjim brzinama nema zaustavljanja rotora na svakom koračnom položaju, ali ugaona brzina oscilira ovisno o položaju. Na velikim koračnim brzinama ugaona brzina prelazi u kontinuirano kretanje.



Slika 4.1. Koračni motor, vanjski izgleda

Pri uzbudi namotaja po programiranom redosljedu rotor se pomakne u diskretnim koracima, odnosno pokrene se u željenom smjeru za određeni ugao. To su dakle sinhroni motori. Smjer vrtnje se može mijenjati promjenom impulsnog slijeda, brzina rotacije promjenom frekvencije impulsa, a ugao (predviđeni broj koraka) zavisi od ukupnog broja pristiglih impulsa. Koriste se za precizno upravljanje i pozicioniranje pokretnih mehanizama u praktičnim uređajima i sistemima. [13]

4.2.1. Osobine i podjela koračnih motora

Kod ovih motora ne postoje kolektor i četkice, cijena im je povoljna, jednostavnog su dizajna, velika akceleracija i snaga, te imaju veliki raspon brzine vrtnje koje je proporcionalna frekvenciji ulaznih impulsa. Nedostatak im je mala korisnost, sklonost rezonanciji, jako zagrijavanje na zahtjevanim režimima, niska izlazna snaga s obzirom na veličinu i težinu, moment znatno opada s povećanjem brzine.

Parametri koračnih motora:

- Rezolucija (broj koraka po okretanju ili dužina koraka)
- Odziv jednog koraka (pokazuje brzinu, oscilatornost i tačnost odziva)
- Tačnost koračnog motora (zavisi od preciznosti izrade motora, izražava se relativnom greškom)
- Statički moment (zavisnost uspostavljenog momenta u motoru od pomaka rotora)
- Dinamički moment (zavisnost srednje vrijednosti momenta motora od brzine vrtnje)
- Start-stopni moment (moment tereta s kojim motor može krenuti pri zadanoj frekvenciji koračnih impulsa, a da ne izgubi kontakt)

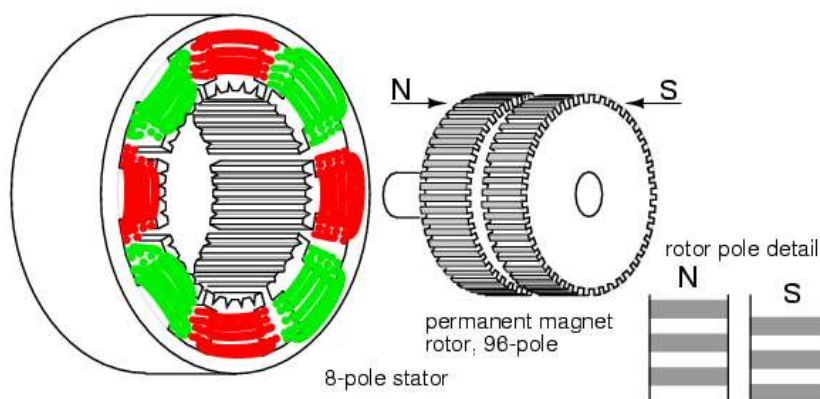
Podjela koračnih motora:

- Prema vrsti uzbude:
 - prema načinu stvaranja uzbude (elektromagnetna uzbuda, uzbuda permanentnim magnetima)
 - prema smještaju uzbude (uzbude na rotoru, uzbuda na statoru)
- Prema izvedbi rotora:
 - nazubljeni čelični rotor
 - rotor s permanentnim magnetima
 - hibridni motori (kombinacija predhodna dva)

- Prema broju faza (najčešće 2-6 faza)
- Prema broju pari polova:
 - koračni motori s permanentnim magnetima imaju na rotoru 1-4 pari polova
 - ostali serijski proizvedeni koračni motori imaju 1-90 pari polova
- Prema načinu kretanja:
 - rotacijski
 - translacijski
- Prema konstrukciji namotaja, odnosno napajanju:
 - unipolarni
 - bipolarni

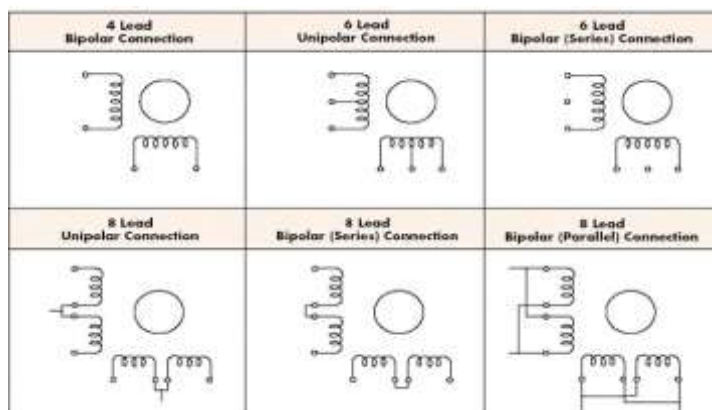
4.2.2. Hibridni i bipolarni koračni motori

Motori korišteni za pogon sva četiri točka su POWERMAX II step motori sa oznakom P22NXXD-LXX-XX-00, veličine NEMA 23, i to su hibridni bipolarni motori. Hibridni motori rade na kombinaciji permanentnomagnetnih i motora sa promjenjivom reluktansom. Stator i namotaji odgovaraju onim kakve imaju reluktantni motori. Rotor hibridnog motora se sastoji od uzdužno orijentisanog permanentnog magneta koji na svakom polu ima nazubljen disk. Jedan disk je na južnom, a drugi na sjevernom polu i međusobno su zarotirani za pola zuba čime je postignuta visoka rezolucija koračanja. Statorski namotaji su podjeljeni u najmanje dvije faze. Ovaj tip motora ima visoku preciznost i veliki moment, a može se konfigurisati i za koračne uglove od 1.8° .



Slika 4.2. Hibridni koračni motor

Priključnih žica može biti 4, 5, 6 ili 8. Statorski namotaji bipolarnog koračnog motora nemaju srednji izvod na pojedinom statorskom namotaju. Za postizanje promjene struje, a time i magnetskog polja kod bipolarnim motora je moguće to izvesti samo promjenom vrijednosti na oba kraja namotaja. Ako su 4 priključne žice, to je bipolarni koračni motor i zahtjeva bipolarni kontroler (*slika 4.3*).[14]



Slika 4.3. Konstrukcije namotaja i način njihovog spajanja

4.3. Pogon HOLBOS-a

Prije kreiranja programa bilo je potrebno upoznati pogonski sistem robota, točkove, motore i drajver koji je korišten za upravljanje s četiri koračna motora čije su opterećenje bili točkovi Holbos-a. Poznavanje lokomocije mobilnog robota je neophodno za upravljanje njegovim kretanjem, određivanjem pozicije i orijentacije mobilnog robota u radnoj okolini i planiranje kretanja ka ciljnoj konfiguraciji. Pogonski mehanizam je opisan u nastavku i sve popratne elektronske komponente koje su korištene pri tome.

4.3.1. "Mecanum Wheel" točkovi

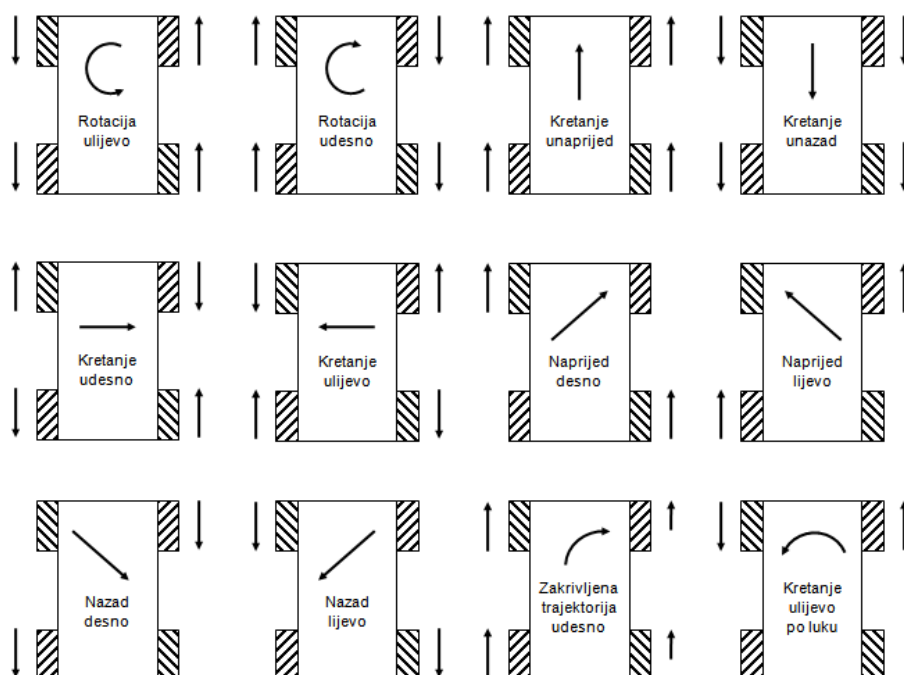
U procesu izrade svesmjernog pogona postoje dva konceptualno različita tipa, na Holbos-u je korišten tzv. Mecanum wheel (*slika 4.4.*) koji je dizajniran tako da se točkovi nalaze u vrhovima četverougla. Benght Ilon je idejni tvorac ovog dizajna, u Švedskoj 1973. godine, kada je radio za kompaniju Mecanum AB.



Slika 4.4. Mecanum wheel točovi

To je konvencionalni točak sa serijom valjaka konektovanih na noseću metalnu strukturu. Ovi valjci imaju ugao rotacije 45 stepeni u odnosu na pravu koja je referentna i za rotaciju cjelokupnog točka. Tipična konfiguracija zahtjeva četiri točka, kako bi se ispunio minimum za postizanje svih smjerova. Orijentacija treba biti takva, da su valjci postavljeni pod ravnim uglom u odnosu na suprotnu dijagonalu vrha u kojem se nalaze. Ako tako postavimo ovaj vrlo interesantni mehanizam, onda pokretanjem ili rotiranjem svih točkova u istom smjeru dobivamo kretanje unaprijed ili unazad. Isto tako, rotacijom jedne strane robota suprotno drugoj vidjet ćemo rotaciju robota desno ili lijevo. Ako pak točkovi jedne strane kreću u suprotnim smjerovima, a tako isto i druge strane, na sceni će biti kretanje u desno ili lijevo. Različitim kombinacijama možemo postići bilo koji ugao kretanja mobilne platforme i uvjeriti se u mogućnosti koje dobivamo sa ovakvim mehanizmom.

Princip rotacije zavisi od smjera pojedinih točkova, kao što je prikazano na *slici 4.5.*



Slika 4.5. Kretanje robota u zavisnosti od smjera rotacije pojedinih točkova

Postoje desno orijentirani i lijevo orijentirani točkovi koji se postavljaju na jednu osovinu. Kada se ovakva dva točka rotiraju u jednom smjeru, komponente brzina koje su paralelne osama rotacije točkova se poništavaju, pa se točkovi kreću naprijed ili nazad. Kada se smjer rotacije lijevog i desnog točka razlikuje, komponente brzina koje su paralelne osama rotacije se sabiraju, pa tako dolazi do kretanja u jednu ili drugu stranu. U ovom završnom radu implementirana su sljedeća kretanja robota: rotacija ulijevo, rotacija udesno, naprijed, nazad, lijevo i desno, tj. nisu implementirana kretanja gdje su uključena samo dva točka.

4.3.2. "Quadstepper 4 Stepper Motor Driver Board"

Pogonski mehanizam robota čine četiri bipolarna step sa osam izvoda spojenih u seriju, a za upravljanje motorima je korišten "Quadstepper 4 Stepper Motor Driver Board" (slika 4.6.) . Drajver omogućava upravljanje do četiri bipolarna step motora istovremeno koristeći nekoliko ulazno-izlaznih pinova. Svaki izlaz za motor može obezbijediti maksimalnu struju u iznosu od 2 A po fazi za vrijednost napona od 35 V. Treba imati u vidu da će ova struja izazvati pregrijavanje drajvera, pa je korišten hladnjak ili ventilator, ukoliko koristimo drajver u punom kapacitetu. Ukoliko se drajver zagrije iznad određene temperature, doći će do isključivanja motora (ploča ima ugrađen PTC senzor za mjerenje temperature). Ploča ima mogućnost upravljanja motora u nekoliko modova rada: upravljanje sa punim korakom, pola

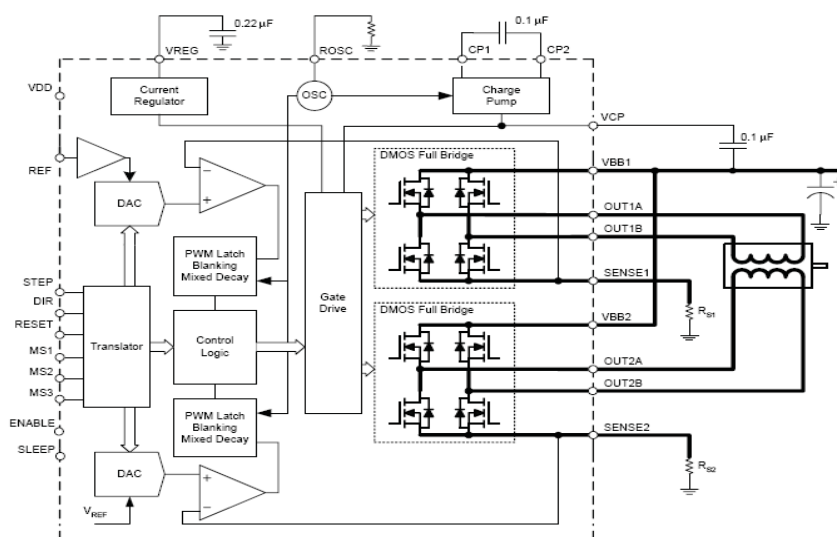
koraka, jednom četvrtinom, osminom i šesnaestinom koraka.[15] Osnovne karakteristike drajvera su:

- Može kontrolisati do 4 bipolarna motora istovremeno
- Svaki izlaz za motor može dati struju do 2 A po fazi na naponu od 35 V
- Mogućnost upravljanja motorima u full, half, quater, eight i sixteenth – step modovima
- Napajanje drajvera, treba da bude priključen na 35 V maksimalno i da može omogućiti struju koja je dovoljna za motore
- Jumper sa oznakom 5 V kontroliše napajanje logike i Vout pina, može imati vrijednosti 3.3 V ili 5 V
- Motori se mogu upravljati spajanjem na sabirnicu od 6 pinova



Slika 4.6. Quadstepper 4 Stepper Motor Driver Board

Ploča se sastoji od ulaznog kruga i četiri identična dijela za upravljanje četiri motora. Ulazni krug se sastoji od regulatora napona LM317, a glavna komponenta koja upravlja motorom je drajver za motor sa oznakom A4983 i na ploči su prisutna ta četiri identična kola. A4983 predstavlja kompletan mikrostopping drajver sa ugrađenim kolom za dešifrovanjem ulaznog signala, predstavlja ključ za laku implementaciju samog kola. Dovođenjem jednog impulsa na ulaz za STEP na drajveru će pomjeriti osovinu motora za jedan korak. Ovaj drajver posjeduje i regulator struje. Njegova blok strukutra je prikazana na sljedećoj slici (*slika 4.7.*):



Slika 4.7. Drajver A4983

Opis pojedinih pinova drajvera:

RESET ulaz – vraća tvorničke postavke na translator i isključuje sve FET izlaze za motore. Svi STEP ulazi se ignorišu dok se RESET ne vrati na logičko 1.

STEP ulaz – Prelazak s niskog nivoa na visoki, na ovom ulazu pokreće sekvencu u translatoru i pokreće motor za jedan korak.

MICROSTEP SELECT (MS1, MS2, MS3) – Pomoću ovih pinova se bira mikrostopping format (tabla 4.1.)

DIR (ulaz za smjer) – Ovim pinovima se određuje smjer motora. Ako je low onda se motor kreće u smjer kazaljke na satu, a ukoliko je high onda će motor da se rotira u suprotnom smjeru (tabela 4.2.)

ENABLE – Ovaj ulaz isključuje/uključuje sve FET izlaze. Kada je na ovom pinu logičko 1, svi izlazi su isključeni.

Tabela 4.1. Rezolucija koraka u zavisnosti od pinova MS1, MS2, MS3

MS1	MS2	MS3	Rezolucija koraka	Pobuda faze motora
L	L	L	Puni korak	2 faze
H	L	L	Pola koraka	1 – faze
L	H	L	Četvrtina koraka	W1-2 faze
H	H	L	Osmina koraka	2W1-2 faze
H	H	H	Šestina koraka	4W1-2 faze

Tabela 4.2. Pregled vrijednosti DIRx pinova za određene smjerove kretanja robota

SMJER	DIR1	DIR2	DIR3	DIR4
Rotacija ulijevo	0	1	1	1
Rotacija udesno	1	0	0	0
Naprijed	1	1	1	1
Nazad	0	0	0	0
Udesno	0	0	1	1
Ulijevo	1	1	0	0
Naprijed desno	1		1	
Naprijed lijevo	/	1	/	0
Nazad desno		1		0
Nazad lijevo	0	/	0	/
Zakrivljena trajektorija udesno	1	1	1	0
Kretanje lijevo po luku	0	1	/	/

4.4. Zaključak

Poznavanje rada komponenti koji služe za upravljanje je ključno za kvalitetno upravljanje i realizaciju samog zadatka i kreiranje softvera. Samo spajanje drajvera s motorima i FPGA karticom je bilo potrebno detaljno proučiti kako ne bi došlo do oštećenja komponenti i nepravilnog kretanja robota.

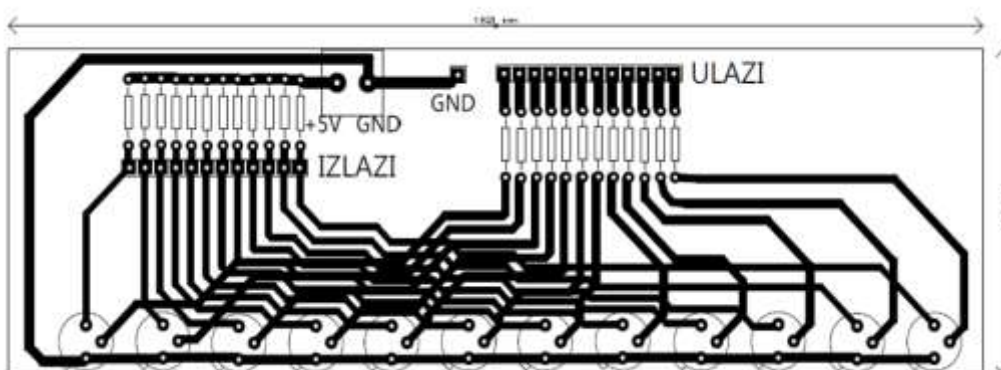
5. Implementacija sistema za upravljanje kretanjem robota

5.1. Uvod

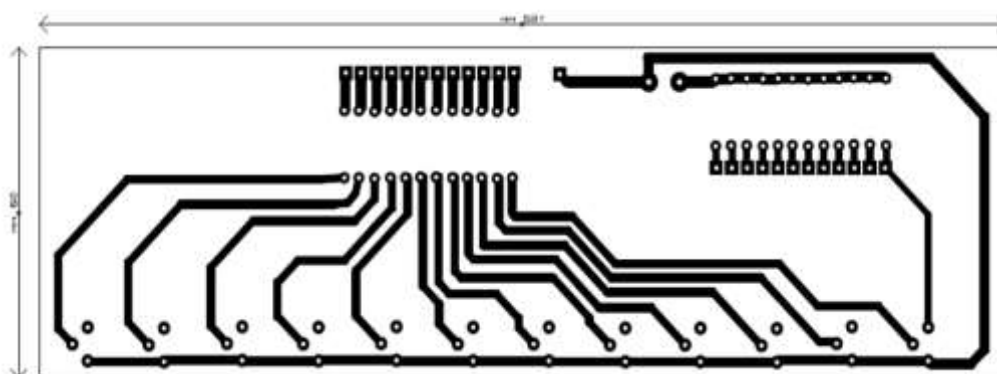
U predhodnom poglavlju je dat detaljan opis pogonskog sistema robota, na osnovu poznatih karakteristika komponenti koje služe za upravljanje realizovan je i program u LabVIEW-u. Korišteni su analogni i digitalni izlazi drajvera, te je Holbos ožičen tako da su svi pinovi drajvera preko flat kabla spojeni na odgovarajuće izlaze kartice. U nastavku se opisuje kako je povezana kartica i drajver i izvršeno prilagođavanje naponskih nivoa kartice i drajvera.

5.2. Konvertor

Visoki logički nivo drajvera je 5 V, a kartice je 3.3 V pa je potrebno izvršiti prilagođavanje napona na svim digitalnim linijama. Prilagodba napona temelji se na povezivanju NPN tranzistora koji rade u režimu zasićenja. Svaki tranzistor predstavlja vezu između izlaznog pina kartice i odgovarajućeg ulaznog pina drajvera. U bazu tranzistora dovodimo izlazni napon sa pina kartice 3.3 V, a kolektorski dio predstavlja izlaz koji vodimo do odgovarajućeg ulaznog pina motora na kojem je potreban napon od 5 V. Konvertor je prikazan na sljedećim slikama (PCB šema):



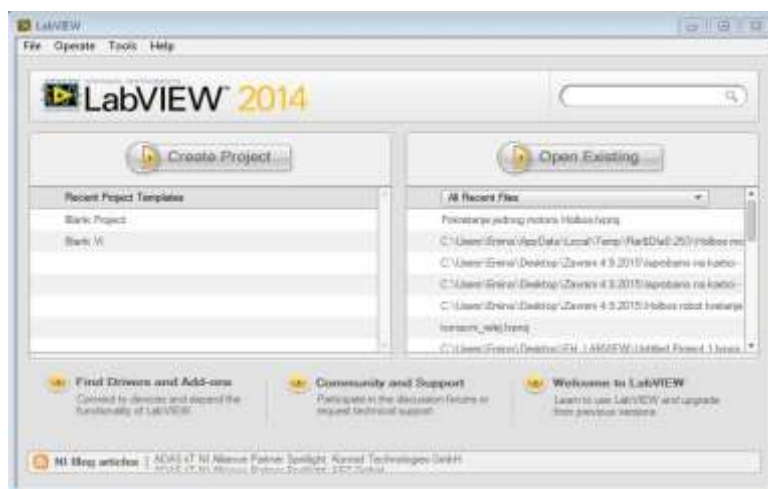
Slika 5.1. PCB šema gornje strane konvertora



Slika 5.2. PCB šema donje strane konvertora

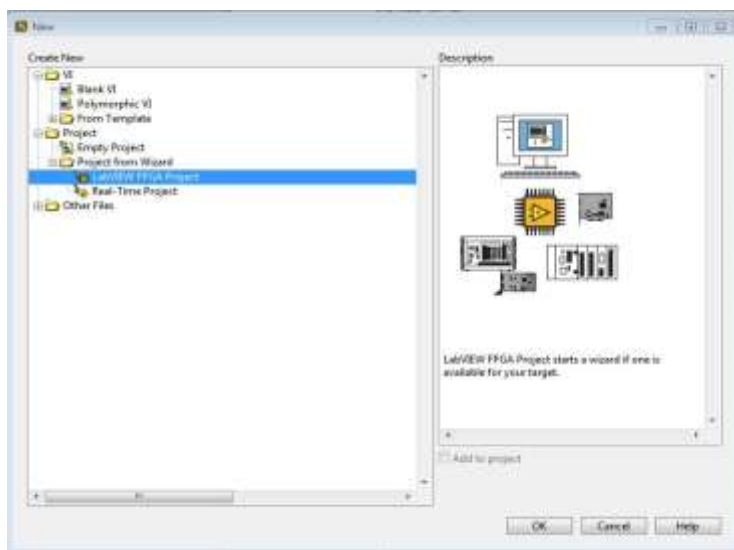
5.3. Projekat u LabVIEW-u

Nakon instaliranja kartice na računar preko ethernet kablja i konfiguracije statičke IP adrese kao što je pokazano u jednom od predhodnih poglavlja, pristupamo pisanju programa koje ćemo prebaciti na FPGA čip. Prije svega potrebno je otvoriti LabVIEW, u našem slučaju opredijelili smo se za verziju 2014. (slika 5.3.)



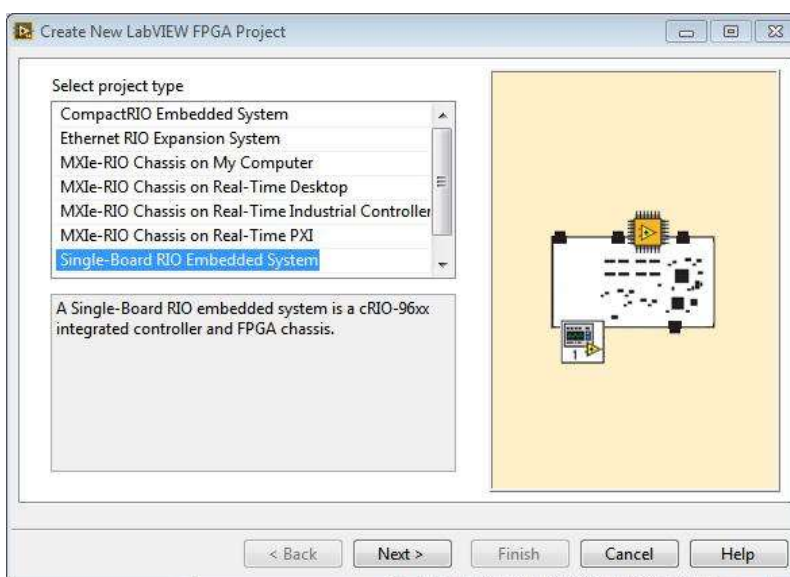
Slika 5.3. Početni prozor

Nakon toga otvorimo *File* i odaberebno *New...*, nakon čega se otvara novi prozor *New* (slika 5.4.):



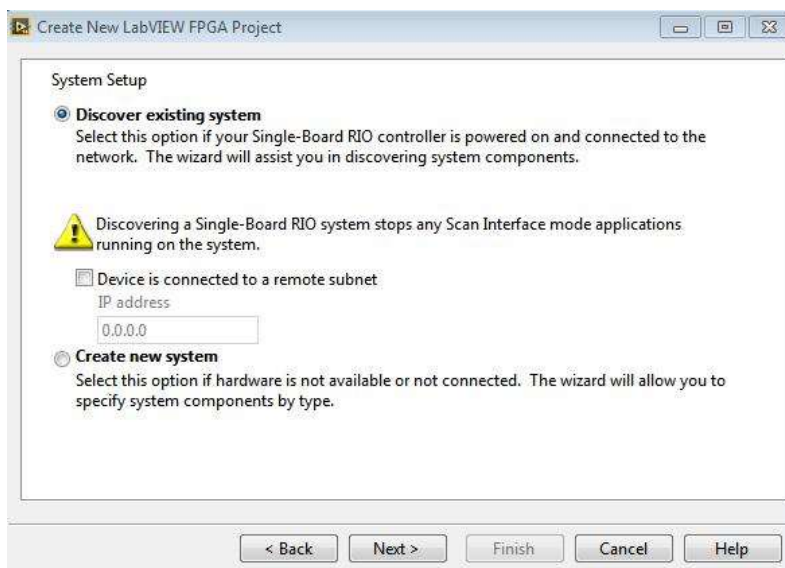
Slika 5.4. Prozor New

Odaberemo LabVIEW FPGA Project i pritisnemo OK. Otvara se novi prozor *Create New LabVIEW FPGA Project* (slika 5.5.) pod kojim odaberemo *Single-Board RIO Embedded System* i idemo na Next.



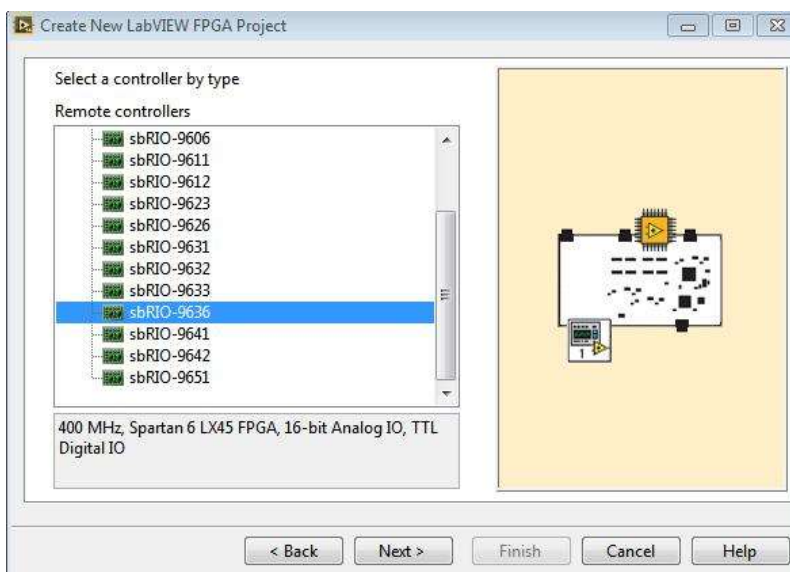
Slika 5.5. Odabir sistema

Otvora se novi prozor s mogućnosti odabira jedne od dvije opcije. Ukoliko znamo da je kartica otkrivena i da je tačna IP adresa biramo *Discover existing system* i slijedimo daljnje korake, ukoliko ne onda idemo na *Create new system* i idemo na Next. (slika 5.6.).



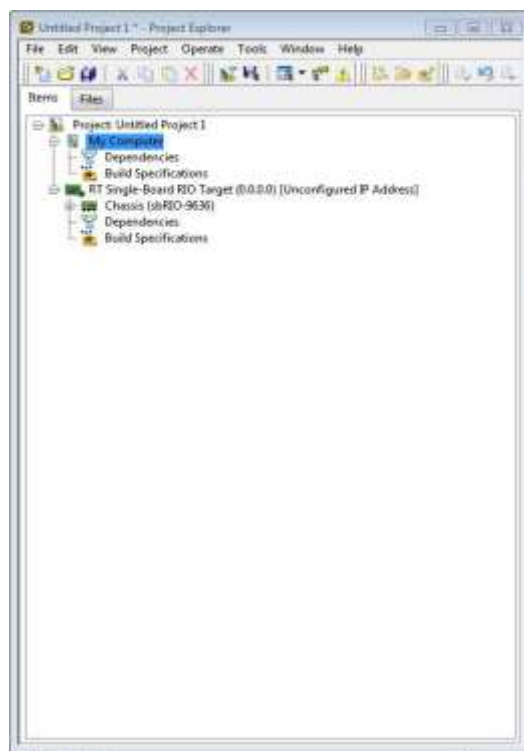
Slika 5.6. Odabir načina otkrivanja sistema

Odaberemo karticu *sbRIO-9636* i nakon toga pritisnemo *Finish*. (slika 5.7.)



Slika 5.7. Odabir vrste kartice

Dalje nam se otvara *Untitled Project* (slika 5.8.):



Slika 5.8. Izgled novog projekta

Ukoliko odabremo da kreiramo novi sistem radimo u simulaciji i ne prebacivamo projekat na nikakav FPGA, a ukoliko pronađemo svoj uređaj onda će svako kompajliranje prebacivati ponovo program na karticu ukoliko mi ne odabremo suprotno. Sada ćemo kreirati VI (Virtual Instrument) tako što kliknemo desnim klikom na FPGA target i odabremo New >> VI. Kreirali smo FPGA VI koji će se upisati na FPGA target (u ovom slučaju to je *sbRIO-9636 Chassis*) i direktno iz ovoga VI možemo upravljati kontrolama u VI. Također, ovdje dodajemo jednostavnom *drag and drop* metodom analogne i digitalne ulaze i izlaze koji se nalaze pod menijima pod nazivom *Connector0*, *Connector1* itd. Da bismo upisali program na kraticu odaberemo *Run*. Kompajliranje može trajati od nekoliko minuta do nekoliko sati, ali ovi projekti i njihovo prebacivanje traje najviše 30 min. Ono što je specifičnost FPGA bazirane kartice jeste komuniciranje sa FPGA VI preko Host VI. Host VI je program pomoću kojeg preuzimamo podatke koje smo prikupili, prikazujemo ih na grafu, usrednjavamo, aproksimiramo ili snimamo u neki document. Preporučljivo je kreirati Host VI zbog ograničenih manipulacija nad podacima u Target VI. Najpogodnije je operacije nad podacima izvršiti u Host, a samo prikupljanje podataka sa kartice i slanje na karticu u Target VI. Host

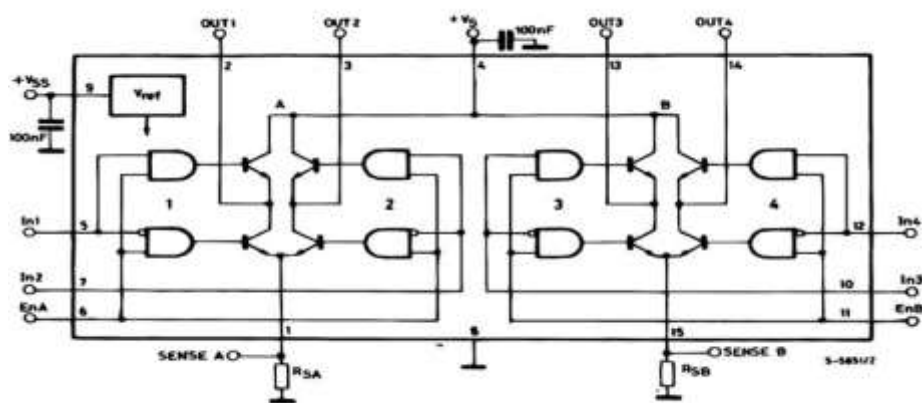
VI možemo kreirati unutar projekta desnim klikom na *Project Ime_projekta.lvproj* <<New<<VI. Sada možemo kreirati *Block Diagrame* i ispisivati ih na karticu.

5.4. Upravljanje jednim koračnim motorom

Prvo je kreiran projekat za upravljanje jednim bipolarnim, hibridnim koračnim motorom korištenjem drajvera koji je kreiran korištenjem kola L298N i L297 koja smo povezali u laboratoriji, to je vršeno s ciljem boljeg upoznavanja kretanja koračnog motora i njegovih performansi. Kolo L298N zapravo predstavlja integrirani dvojni H-most sa standardnim TTL logičkim nivoima. Blok diagram i izgled kola je prikazan na sljedećim slikama(*slika 5.9. i slika 5.10.*).



Slika 5.9. L298N izgled kola



Slika 5.10. Block diagram kola L298N

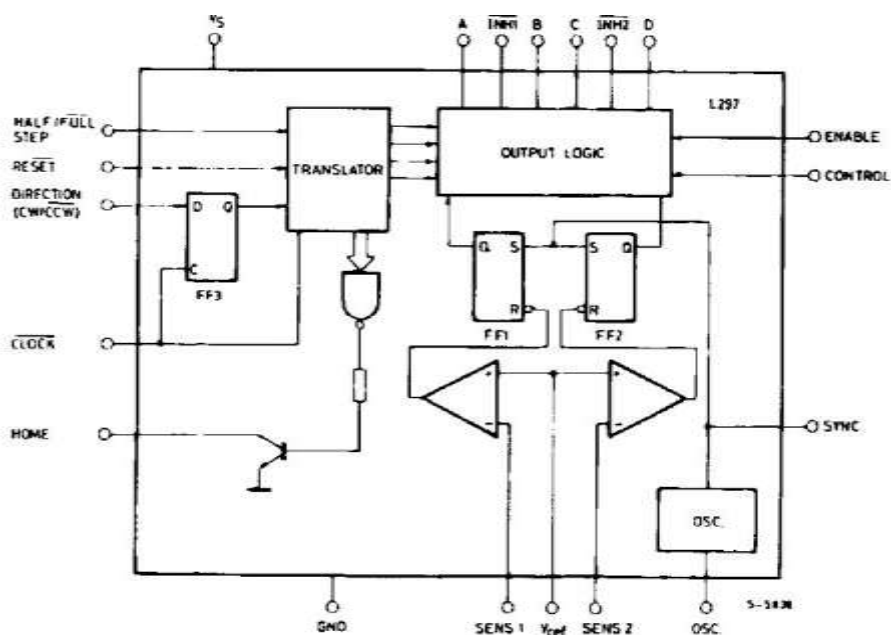
Kolo L297 (*slika 5.11. i slika 5.12.*) je kontroler za stepper motor. Motor se može pokretati u polukoračnom, koračnom modu. Prednosti upravljanja ovog kola su što zahtijeva samo clock

signal, signal za smjer i signal za mod rada pa je upravljanje mnogo lakše. Drajver formiran od ova dva kola je korišten jer je upravo sličan radu drajvera koji je korišten na robotu.



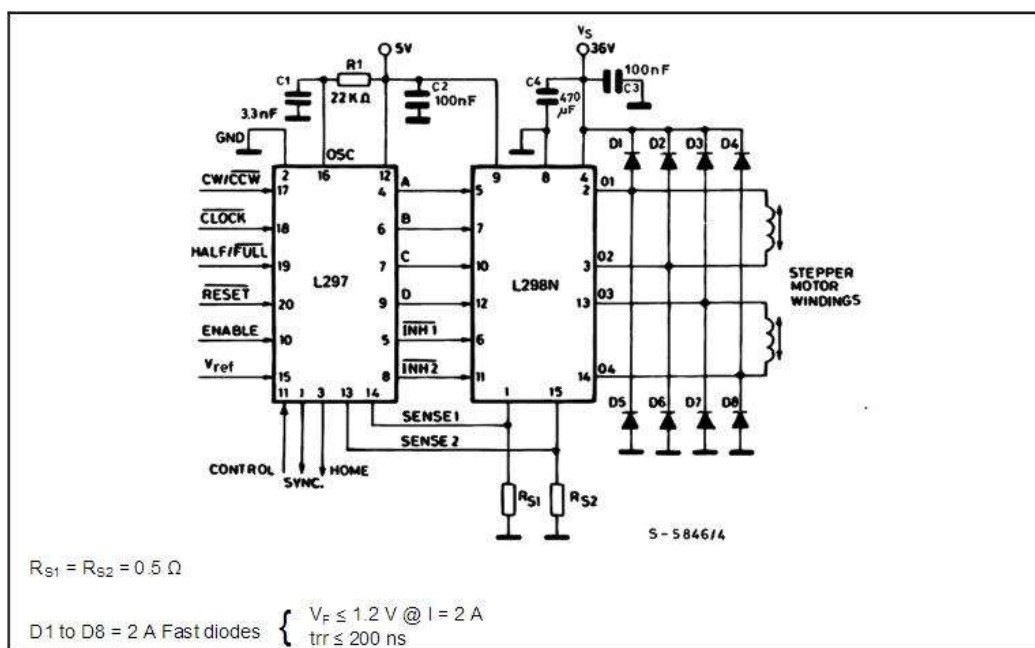
DIP20

Slika 5.11. Izgled kola L297

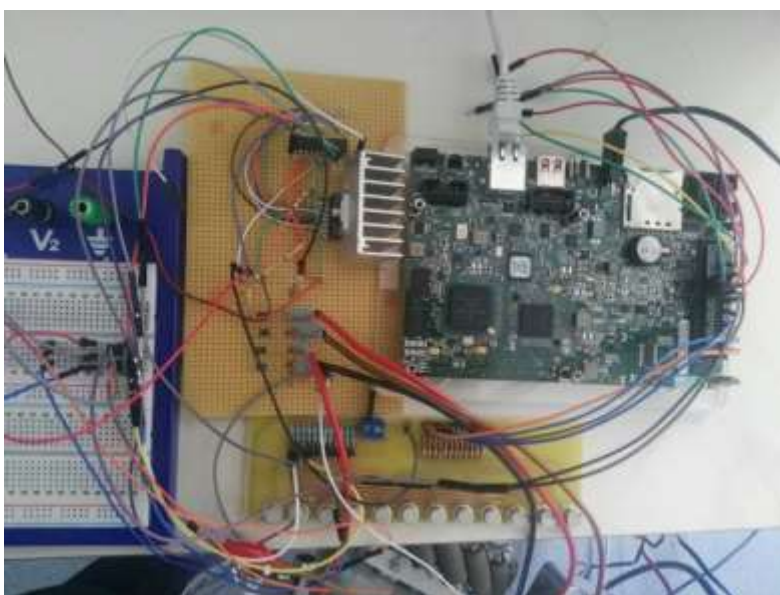


Slika 5.12. Block diagram kola L297

Spajanje ova dva kola kako bismo drajver je prikazano u nastavku (slika 5.13.i slika 5.14.).

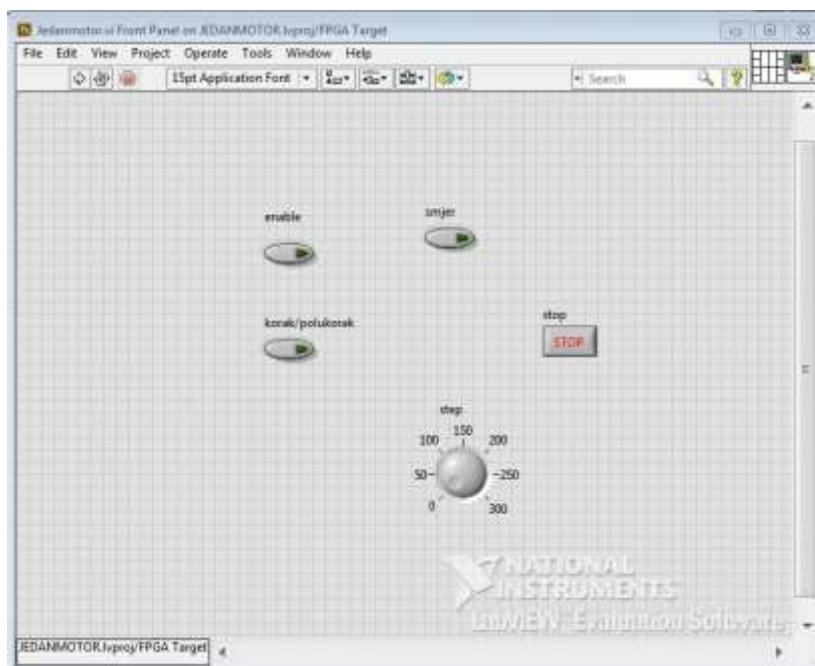


Slika 5.13. Povezivanje kola L297 i L298N s ciljem dobivanja drajvera za koračni motor

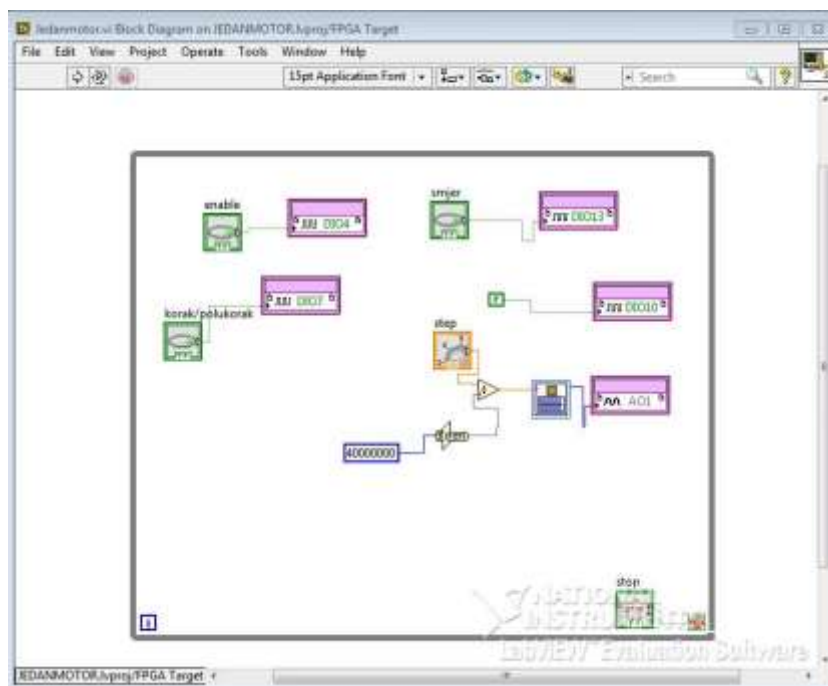


Slika 5.14. Povezivanje drajvera, kartice i konvertora

U nastavku je prikazan jednostavni program u LabVIEW-u kojim upravljamo koračnim motorom preko predhodno pomenutog, kreiranog drajvera (slika 5.14.-5.17.).



Slika 5.15. Front panel FPGA targeta

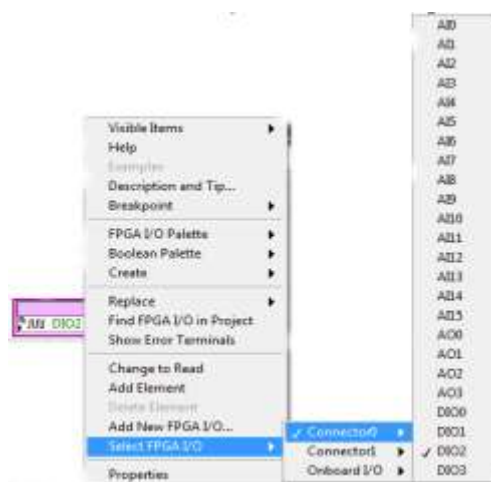


Slika 5.16. Block diagram FPGA targeta

Kako bismo bili u mogućnosti da upravljamo digitalnim i analognim ulazima i izlazima potrebno je da odaberemo sa kojim od njih želimo upravljati. Izgled jednog čvora je prikazan na slici 5.16. a način odabira kojim od digitalnim ili analognih ulaza/izlaza upravljamo na slici 5.17..



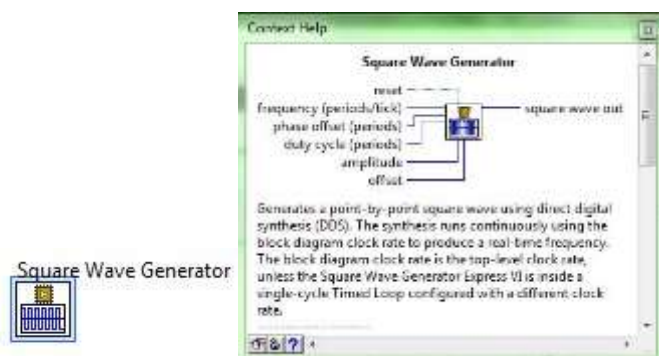
Slika 5.17. Izgled jednog analognog ili digitalnog čvora



Slika 5.18. Način odabira ulaza/izlaza

Također je poznato da ulazima/izlazima možemo pristupiti samo preko FPGA Targeta, tako da je besmisleno pokušavati dodati ih na HOST Target.

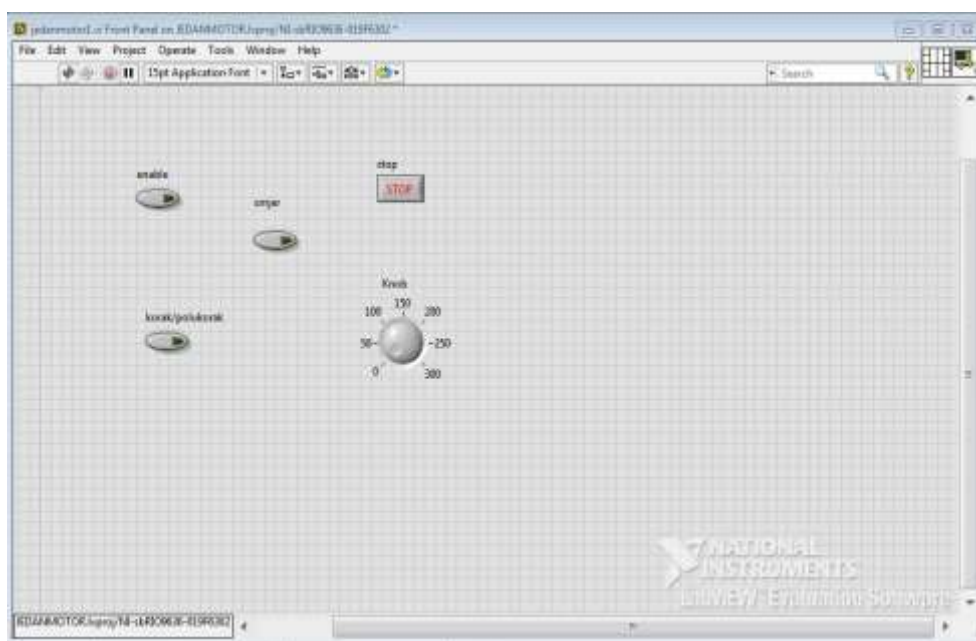
Enable ulaz sa drajvera spajamo direktno na digitalni izlaz DIO4, kao i ostale ulaze drajvera, za određivanje moda rada (*korak/polukorak*) na DIO7, za određivanje *smjera* na DIO13, na DIO10 je povezan ulaz drajvera koji određuje koji izlazi će s drajvera biti omogućeni. Na kraju, na analogni izlaz kartice AO1 šaljem četvrtke iz bloka koji generira četvrtke, *SquareWave Generator* (slika 5.18).



Slika 5.19. Blok Square Wave Generator

NI sbRIO-9636 ima *onboard-clock* od 40MHz, a ulaz u blok *SquareWave Generator* prima *FXP* (*fixed-point*) broj koji je multipl od 40MHz. Frekvenciju u Hz koju želimo primijeniti je predhodno potrebno podijeliti sa 40 000 000 Hz. Korišteni blokovi za generisanje četvrtki u ovom projektu su *Integer to Fixed-Point Cast*, *FPGA I/O Node*, *Knob*, *SquareWave Generator*, *Devide* i *Intager*. Za blok *Knob* potrebno je bilo kliknuti na njega desnim klikom i odabrati *Representation* i *SGL* da bi se moglo kompajlirati.

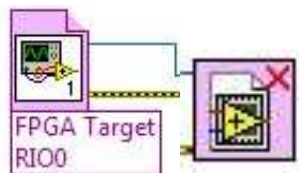
Izgled Host VI je prikazan u nastavku (*slika 5.19.* i *slika 5.20.*)



Slika 5.20. Izgled Front panel HOST-a

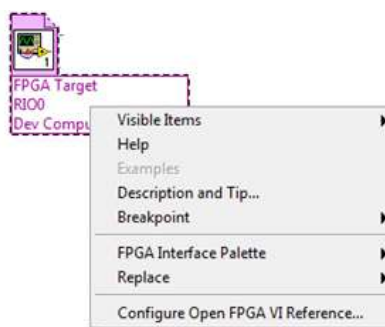
Do ovog trenutka smo se već upoznali sa principom rada LabVIEW, tako da nećemo objašnjavati značenje svakog bloka pojedinačno, samo one koji nisu uobičajeni.

Na sljedećim slikama su prikazani blokovi koji služe za pristup i zatvaranje FPGA-a Targeta, respektivno *Slika 5.22*.



Slika 5.23. Izgled blokova Open FPGA VI Reference i Close FPGA VI Reference

Kako bismo bili u mogućnosti da izvršimo pristup FPGA Targetu moramo da odaberemo kojem programu pristupamo. To radimo na način da desnim pritisnemo na *Open FPGA VI Reference* i odaberemo *Configure Open FPGA VI Reference*. (slika 5.23.)



Slika 5.24. Izgled padajućeg menija nakon pritiska desnog klika

Nakon toga se otvara novi prozor *Configure Open FPGA VI Reference* čiji je izgled prikazan na slici 5.24.



Slika 5.25. Izgled prozora *Configure Open FPGA Reference*

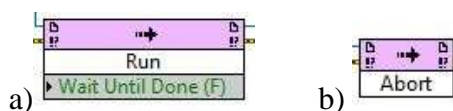
U ponuđenom prozoru odaberemo na koji FPGA Target se pozivamo i način odabira je prikazan na slici 5.25.



Slika 5.26. Izgled prozora preko kojeg selektujemo na koji FPGA Target se pozivamo

Nakon objašnjenje prethodna 2 bloka pristupit ćemo objašnjenju blokova:

- *Invoke Method* prikazanog na slici 5.26 a) i b)
- *Read/Write Control* prikazanog na slici 5.27.
- *FPGA I/O Node* prikazanog na slici 5.28.



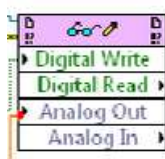
Slika 5.27. a) i b) Prikaz *Invoke Method*-a sa odabranim metodama *Run* i *Abort*

Nakon što odaberemo *Invoke Method* i prenesemo je u radni prostor preduslov za odabir metode je da prvo izvršimo njeno povezivanje sa blokovima *Open* i *Close FPGA VI Reference*, jer u suprotnom nećemo biti u mogućnosti ništa uraditi sa ovim blokom. Zatim je potrebno da odaberemo metodu koju će ovaj blok predstavljati. Taj dio ćemo odraditi tako što na metodi pritisnemo desni klik i odaberemo jednu od ponuđenih metoda iz padajućeg menija koji je prikazan na slici 5.27.



Slika 5.28. Način odabira metode koju će predstavljati blok Invoke Method

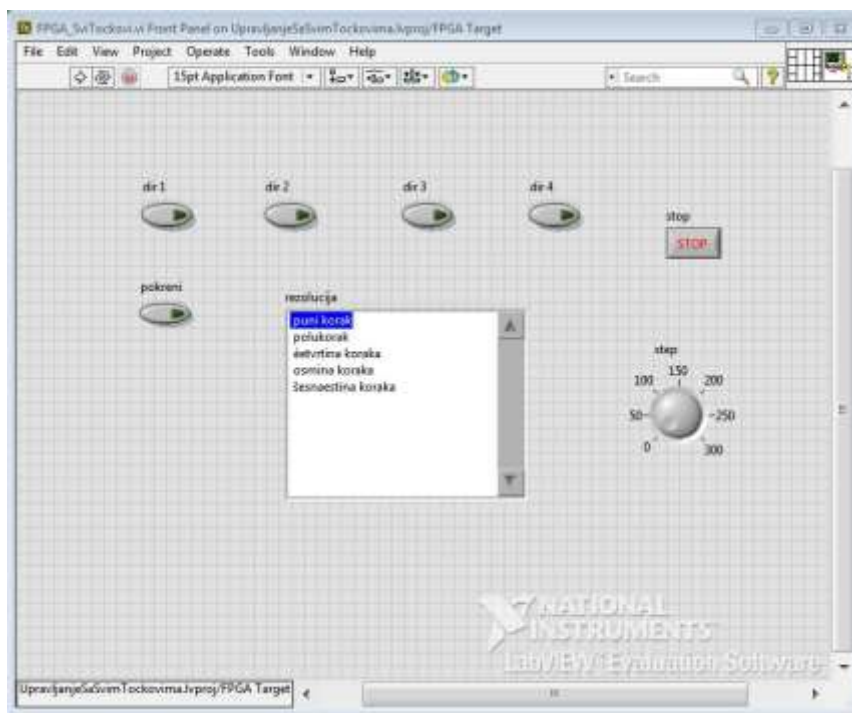
Read/Write Control prikazana na slici 5.28. nam omogućava da pristupamo kontrolama koje smo definisili na FPGA Targetu preko Host Targeta. Ukoliko mišem razvučemo kontrolu u mogućnosti smo da preko Host Targeta upravljamo svim kontrolama navedenim na FPGA Targetu. Način odabira kontrole je tako što desnim kliknemo na ponuđene kontrole i iz padajućeg menija odaberemo s kojom kontrolom želimo upravljati.



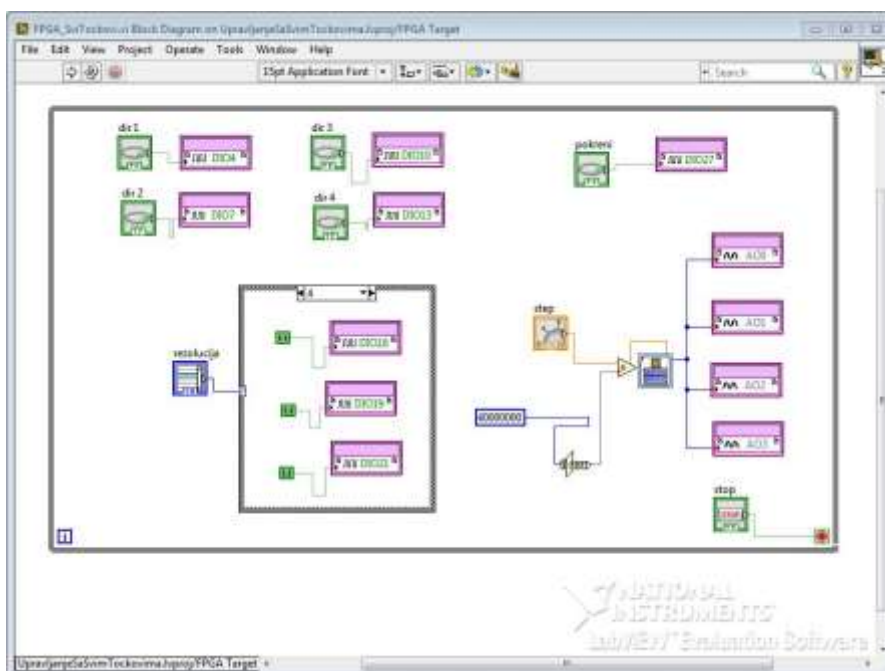
Slika 5.29. Izgled bloka Read/Write Control

5.5. Program za upravljanje HOLBOS-om

Za upravljanje HOLBOS-om potrebno je 12 linija, 8 digitalnih linija i 4 analogna izlaza. Sljedeće slike pokazuju izgled VI FPGA targeta i HOST-a.

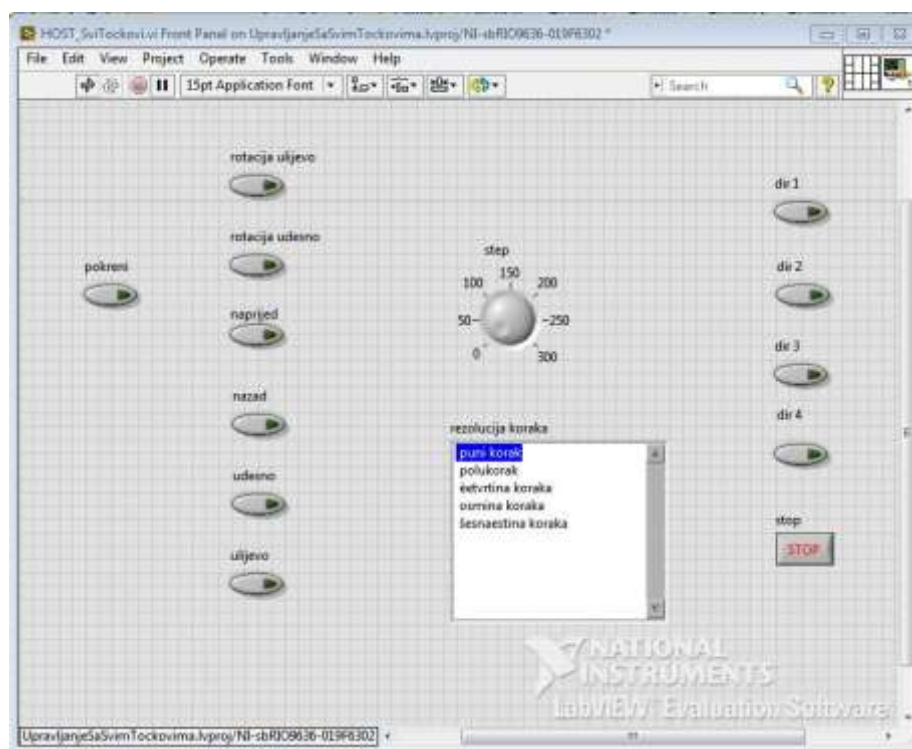


Slika 5.30. Front Panel FPGA targeta za upravljanje HOLBOS-om

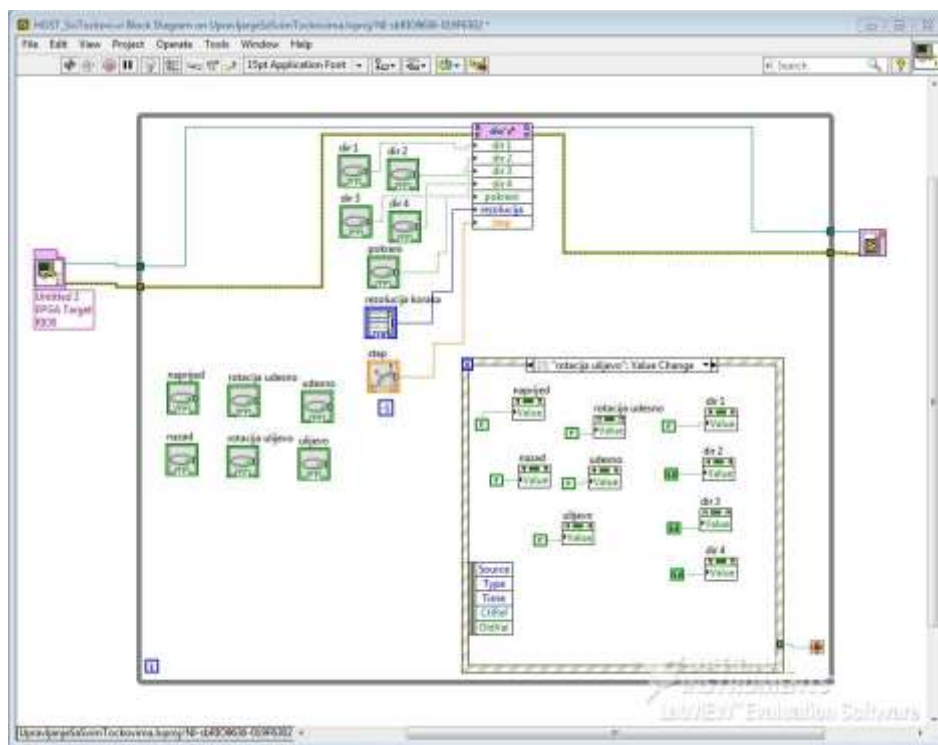


Slika 5.31. Izgled Block Diagrama FPGA targeta za upravljanje HOLBOS-om

Za pinove ENABLE koristimo jedan digitalni izlaz i u ovisnosti od toga da li želimo da omogućimo ili neomogućimo kretanje točkova šalje na taj pin logičko 0 ili 1. Za pinove koji određuju smjer, DIRx korišteno je 4 digitalna izlaza, ostale digitalne izlaze (tri) koriste se za odabir rezolucije koraka. Za odabir koraka koristimo *Listbox* i *Case Structure*, pri čemu je *Default Case* puni korak. Analogne izlaze koristimo za generisanje četvrtki na pinove STPx, pri čemu ove izlaze spajamo direktno na pinove drajvera jer oni mogu dati napon od 5 V. U HOST-u smo implementirali logiku upravljanja kontrolama *naprijed*, *nazad* itd., korišteni su *Push Buttons* pa s obzirom da ih može biti više istovremeno uključeno onda je korištena i *Event Structure* kako bi se otklonila mogućnost da ih bude više istovremeno uključeno. Izgled *Front Panela* i *Block Diagrama* HOST-a prikazani su u nastavku (slika 5.31. i 5.32.)



Slika 5.32. Izgled Front Panela HOST-a za upravljanje HOLBOS-om



Slika 5.33. Izgled Block Diagrama HOST-a za upravljanje HOLBOS-om

5.6. Zaključak

Prikazani programi su kreirani na jednostavan način, ali su funkcionalni i mogu se vrlo lahko nadograditi. Iako je sam program i njegova logika jednostavna, ipak je dosta truda uloženo na upoznavanje rada same kartice kojom smo upravljali i njenog programiranja u LabVIEW-u, svih pomenutih blokova, pinova i ožičavanja. Cijeli program je priložen uz ovaj pisani dokument u posebnom fajlu.

6. Zaključno poglavlje

Ovo poglavlje daje osvrt na rješenje zadanog problema, problema sa aspekta pristupačnosti, složenosti i mogućnosti nadogradnje. U nastavku je data i slika robota HOLBOS-a (*slika 6.1.*)



Slika 6.1. Robot HOLBOS, raspored točkova i unutrašnja struktura

Napisani programski kod je funkcionalan i *user friendly*, što je njegova najbitnija karakteristika jer korisnik programa ne mora poznavati sam pogonski mehanizam niti strukturu i osobine drajvera za motore da bi upravljao njegovim kretanjem. Također je bilo bitno da se naš program izvršava u realnom vremenu i da pri kretanju nema poteškoća zbog toga, to je sve moguće jer korištena kartica NI sbRIO-9636 posjeduje *Real-Time Modul* koji je omogućio simulaciju u realnom vremenu. Uočili smo da je brzina odziva robota i njegovog pogona jako dobra. Kako robot HOLBOS posjeduje i senzore za izbjegavanje prepreka, zatim platformu za podizanje tereta itd., napisani program je vrlo lahko moguće nadograditi i tako ostvariti još kvalitetnije i cjelokupno upravljanje robotom. Dizajn i implementacija sistema za kretanje je u principu osnova u daljem projektovanju robotskog sistema. Prezentirane su adekvatne strukture koje omogućavaju kretanje tijela mobilnog robota. To su mehanizmi za svesmjerno kretanje (eng.Omni wheel) i hardversko-softverska podrška za motore. Pažnja je data na hardver koji omogućava traženo preključivanje faza step motora kao i softver kojim definiramo modove rada i dajemo inicijalne signale za pokretanje motora. Svesmjerni, mobilni roboti su našli primjenu u svakodnevnom životu kao uređaji za brži, lakši transport ljudi i objekata, od viljuškara u industrijskoj proizvodnji do vodiča u muzejima. Opisane su

detaljno hardverske i softverske karakteristike pogonskog mehanizma robota HOLBOS. Vrlo bitan dio ovog završnog rada je i upoznavanje s radom jedne nove tehnologije, FPGA, čiji rast i razvoj ima tendenciju eksponencijalnog rasta. Primjena FPGA sklopova, ne samo u oblasti automatike, već i u oblastima medicine, obrade slike i zvuka, enkripcije podataka i sl. vrlo je opravdana imajući u vidu sve njene prednosti, u koje smo se mogli uvjeriti tokom izrade ovog rada.

7. Literatura

- [1] Xilinx. FPGA vs. ASIC - What is the difference between a FPGA and an ASIC?, www.xilinx.com/fpga/asic.htm
- [2] Xilinx. Field Programmable Gate Array (FPGA), www.xilinx.com/training/fpga/fpga-eld-programmable-gate-array.htm
- [3] Xilinx. Machine Vision Systems, www.xilinx.com/applications/industrial/machine-vision-systems.htm
- [4] Xilinx. Machine Vision, www.xilinx.com/applications/industrial/machine-vion.htm
- [5] Xilinx. What is a FPGA?, www.xilinx.com/fpga/Altera. Altera FPGAs, www.altera.com/products/fpga/overview.htm
- [6] National Instruments (2014). FPGA fundamentals, www.ni.com/white-paper/6983/en
- [7] Xilinx. What is a FPGA?, www.xilinx.com/fpga/Altera. Altera FPGAs, www.altera.com/products/fpga/overview.htm
- [8] National Instruments (2014). NI LabVIEW FPGA module, www.ni.com/labview/fpga
- [9] National Instruments (2014). sbRIO-9636 Embedded Device With Analog I/O and DIO, LX45 FPGA, sine.ni.com/nips/cds/view/p/lang/en/nid/21042
- [10] National Instruments (2014). NI Single-Board RIO FAQ, www.ni.com/white-paper/7490/en/
- [11] National Instruments (2014). NI LabVIEW RIO Evaluation Kit – Additional Exercises, www.ni.com/example/7783/en/#toc3
- [12] National Instruments (2014). NI sbRIO-9605/9606/9623/9626/9633/9636 OEM Operating Instructions and Specifications, www.ni.com/pdf/manuals/373378c.pdf
- [13] N.K. Thulasiraman, A.F. Haider and Y.S. Cheng (2010). "A Reconfigurable Wireless Stepper Motor Controller Based On FPGA Implementation," IEEE Symposium on Industrial Electronic and Applications, pp. 585-590
- [14] J. Velagić, Predavanje iz predmeta Aktuatori, Lekcija 6, people.etf.unsa.ba/jvelagic/laras/dok/Lekcija6.pdf
- [15] Allegro Microsystems. A4983 DMOS Microstepping Driver with Translator, cdn.sparkfun.com/datasheets/Widgets/A4983SETTR-T-Allegro-datasheet-123643.pdf