Značajke i mogućnosti uređaja za mjerenje i upravljanje NI CompactRIO

Špoljarić, Dražen

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:939567

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij

Značajke i mogućnosti uređaja za mjerenje i upravljanje NI CompactRIO

Završni rad

Dražen Špoljarić

Osijek, 2016.

Sadržaj

1. UVOD 1
1.1. Zadatak završnog rada 1
2. OSNOVE CompactRIO KONTROLERA 2
2.1. Tehničke karakteristike kontolera 2
2.2. Način rada CompactRIO kontrolera 3
2.2.1. Procesor 3
2.2.2. Naknadno programirljiv integrirani krug (FPGA) 4
2.3. Mehanička svojstva i povezivanje s uređajima 4
2.4. Operacijski sustav za NI CompactRIO 5
3. NAČINI RADA S CompactRIO UREĐAJEM 6
3.1. Razvoj povezanog sustava 6
3.2. Mogućnosti uređaja 6
3.3. Senzor pokreta 7
3.4. Ugrađeni softver 8
3.5. Praćenje i analiza podataka 9
3.6. Prikazivanje podataka preko zaslona i komunikacija s protokolima 10
3.7. Analogni i digitalni ulaz /izlaz (I/O) 11
4. POVEZANI SUSTAV PROGRAMA C/C++ I LabVIEW 15
4.1. Pregled 15
4.2. Uvod u NI povezanu sistemsku građu 16
4.3. Zahtjevi programiranja 18
4.3.1. Real-Time Programiranje 19
4.3.2. FPGA Programiranje 19
4.4. Odabir najpouzdanijih programa 19
4.5. Korištenje LabVIEW za FPGA programiranje 22
5. UPOREĐIVANJE SVOJSTAVA VIŠE KONTOLERA 24
5.1. Pregled 24
5.2. Izvođenje testova za usporedbu 24
5.3. Test na LabVIEW 26

5.4. Test Propusnosti podataka 27
5.5. Test na aplikacijama 29
5.6. Utjecaj povezanog sučelja 31
5.7. Osvrt na CompactRIO kontoler 33
6. PRIMJENA CompactRIO UREĐAJA 34
6.1. Elektrane- promatranje generatora pomoću dijagnostike 34
6.1.1. Sistemska konfiguracija 35
6.1.2. Praćenje rada visokotlačne crpke 35
6.2. Primjena kod zrakoplova 37
7. ZAKLJUČAK 41
LITERATURA 42
SAŽETAK 43
ŽIVOTOPIS 44

1. UVOD

Potreba da se razni pogoni što bolje parametriraju i reguliraju dovodi do razvijanja raznih tehnologija i uređaja koji su u mogućnosti pružiti željene radnje. Ručno upravljanje je izrazito sporo i neučinkovito zbog toga što se mora čim prije reagirati na neku pojavu i ispraviti potrebne podatke, a i uvijek postoje ljudske pogreške. Radi toga su usavršeni uređaji kao što je NI CompactRIO i slični kontroleri. NI CompactRIO je proizvod tvrtke National instruments i on je kombinacija pravovremenog (real-time) kontrolora, izmjenjivih ulazno-izlaznih modula, FPGA modula i Ethernet proširenog postolja. CompactRIO kontoleri se mogu programirati pomoću LabVIEW dok se regulatori mogu programirati još s C ili C++.

Prvi dio rada objašnjava općenite podatke samog uređaja CompactRIO, također i njegove mogućnosti i specifikacije kao što su: kontrola pokreta, nadgledanje, obrada i analiziranje podataka, također i vizualizaciju podataka. Drugi dio rada se odnosi na programiranje uređaja. Dani su i primjeri uporabe uređaja u pogonima kao što su: zrakoplovi i elektrane.

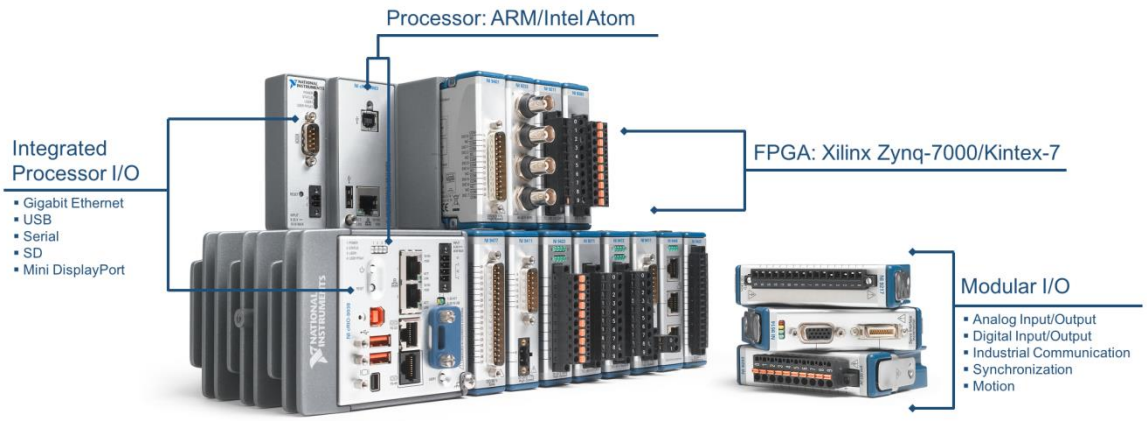
1.1. Zadatak završnog rada

Zadatak završnog rada je upoznavanje s mogućnostima NI CompactRIO platforme, upoznavanje kruga njegove primjene, te se osvrnuti na komunikaciju samog uređaja sa sustavima nadzora i upravljanja industrijskih postrojenja i pogona koristeći program LabVIEW.

2. OSNOVE CompactRIO KONTROLERA

CompactRIO kontroleri su povezani regulatori visokih performansi s industrijskim I/O modulom, dobrih mehaničkih sposobnosti, standardnih certifikata, integrirane vizije pokrete, industrijske komunikacije i ljudskih mehaničkih sposobnosti [1].

2.1. Tehničke karakteristike kontolera



Slika 2.1. CompactRIO [2].

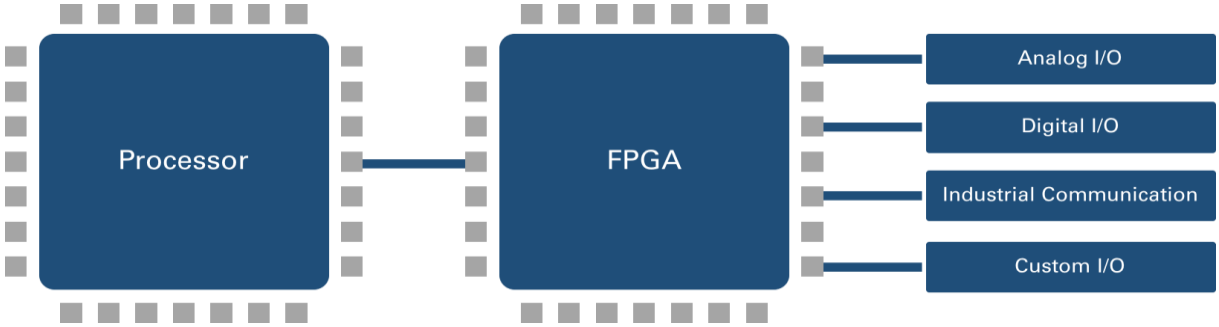
CompactRIO kontroler koristi najnovije procesore Intel i ARM sa snažnim FPGA iz Xilinx tehnologije i visoke točnosti I/O s mjerenim specifičnim signalnim uređajem u čvrstom i kompaktnom kućištu. NI Linux Real-Time OS pomaže programerima oko inovacija s raširenim softverom koje pruža Linux zajednica. Postoje i razne izvedbe I/O koje se mogu odabrati za uporabu [2].

Tablica 2.1. Podaci CompactRIO uređaja [2].

Features	Specifications
Processor	667 MHz Dual-Core ARM Cortex-A9, 1.33 GHz Dual-Core Intel Atom, 1.91 GHz Quad-Core Intel Atom
Operating System	NI Linux Real-Time
Onboard Storage	Up to 16 GB
FPGA	Zynq-7000 (Z-7020), Kintex-7 (70T, 160T, 325T)
FPGA-Accessible DRAM	Up to 128 MB available on some models
I/O	Up to 8 I/O module slots, Gigabit Ethernet, USB, RS232/RS485, Mini DisplayPort, SD, WiFi (802.11a/b/g/n)
Power	9 VDC to 30 VDC supply range, up to 46W max
Ruggedness	Up to -40° C to 70° C, up to 50 g shock and 5 g vibration
Size	Starts at 17.8 x 8.7 x 6.4 cm^3
Certifications	

2.2. Način rada CompactRIO kontrolera

CompactRIO regulatori imaju dva cilja obrade: pravovremeni (real-time) procesor za komunikaciju i obradu signala i FPGA za provedbu nadzora velike brzine, vremena i za pokretanje izravno u hardveru [2].



Slika 2.2. Shema CompactRIO uređaja [2].

2.2.1. Procesor

Postoje razni procesori koji se koriste u kontolerima. Kao što su: Intel Atom System-on-Chip (SoC) i ARM-based procesor. Intel Atom SoC ima integrirani GPU, višejezgreni 1.91 GHz procesor i SD card pohranu. Također ima mogućnost gubitaka snage od 6 W, koji zadržava kontroler mehaničkog dizajna kompaktan i dalje omogućuje prošireni raspon radne temperature. Koristi se još i Zynq SoC s 66 MHz dual-core ARM Cortex-A9 procesor za visoke performanse uz manju potrošnju energije [2].

2.2.2. Naknadno programirljiv integrirani krug (FPGA)

CompactRIO regulatori koriste Xilinx 7 Series FPGA modele koji imaju veću učinkovitost i 50 posto manju potrošnju energije po ćeliji od prethodnih generacija i koriste 28 nm tehnologiju koja omogućava više logičkih ćelija i digitalnu obradu signala (DSP) na istoj veličini čipa. Moguće je koristiti Artix-7 FPGA modele (Zynq-7000 SoC) s 85.000 logičkih ćelija i 220 DSP slojeva ili Kintex-7 FPGA s do 407,600 logičkih ćelija i 840 DSP slojeva. S dodatnim FPGA prostorom se može provesti više naprednih kontrola, obrada signala, filtriranje i mnogo drugih logičkih radnji [2].

2.3. Mehanička svojstva i povezivanje s uređajima

CompactRIO kontroler ima zavidnu kontrolu u kompaktnom, nepristupačnom paketu te ima raspon radne temperature od -40 ° C do 70 ° C (-40 ° F do 158 ° F), dodatne ulaze za napajanje, zadovoljava pravila sigurnosti i certifikate sustava, te daje dobre rezultate u teškim industrijskim okolinama.

Eliminira se potreba za odvojenim podsustavima spajanjem komponenti direktno na CompactRIO kontroler s ugrađenim procesorom I/O.

Izravno se mogu spojiti industrijske kamere i motori kako bi se stvorili napredni algoritmi za obradu podataka. CompactRIO kontroleri se koriste kao industrijski posrednici i spajaju se s različitim uređajima i infrastrukturama s izvornom podrškom za industrijske protokole kao što su: PROFINET, OPC UA i EtherCAT. projektna interakcija je prepuna GUI značajkama i spojena s lokalnim, udaljenim ili mobilnim HMIs za vizualizaciju podataka i pregledno sučelje [2].

2.4. Operacijski sustav za NI CompactRIO

CompactRIO kontroler koristi NI Linux Real-Time, real-time OS baziran na standardnoj jezgri i dizajniran posebno za pouzdan i deterministički rad u dugoročnom razvoju, s značajkama kao što su izvorna podrška za povećanje sigurnosti Linuxa kako bi se poboljšala sigurnost i pouzdanost. Razvijanje je brže s Linux sustavom aplikacija ako se za pomoć koriste primjeri, te suradnjom između zajednice korisnika i programera. Također povezivanjem s NI Alliance Partner Network zajednicom koja je raširena po čitavom svijetu, uključujući povezane kontrole, elektronički dizajn i viziju specijalnih partnera. Bilo da su u pitanju: proizvodi, sustavi, integracija, konzultacije ili obuka, partneri unutar zajednice posebno su obučeni za rješavanje nekih od najtežih inženjerskih projekata [2].

3. NAČINI RADA S CompactRIO UREĐAJEM

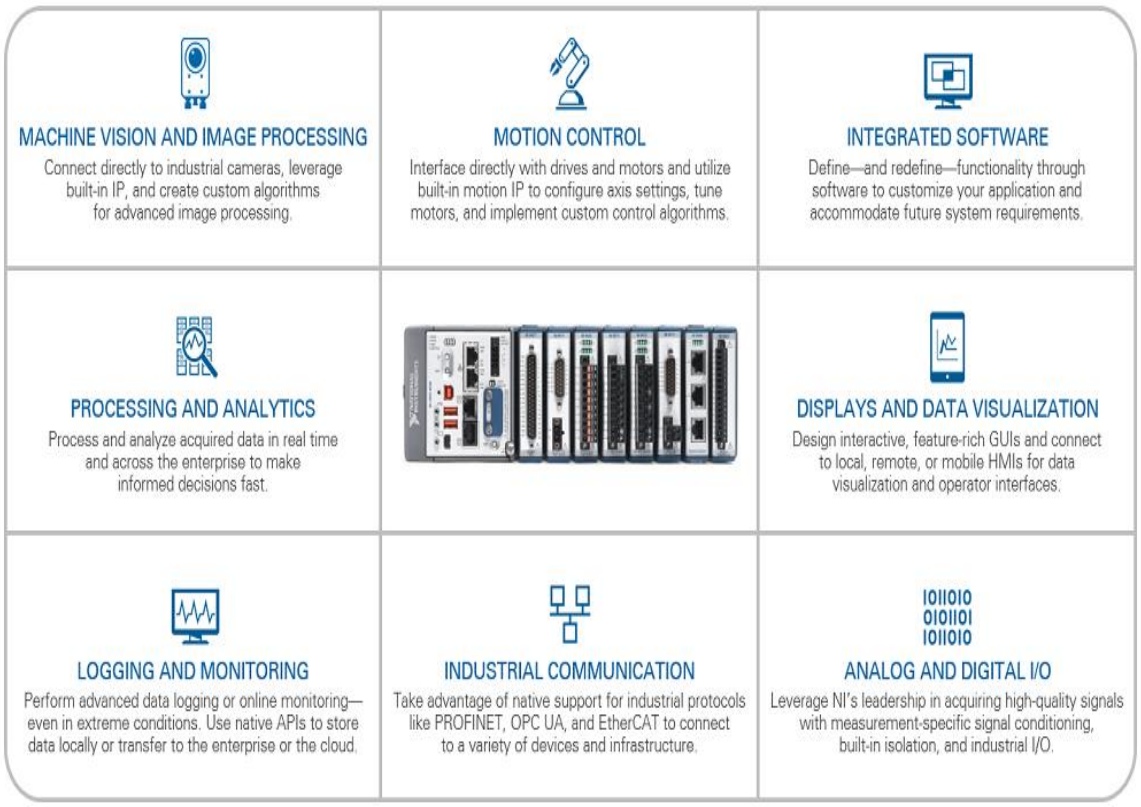
3.1. Razvoj povezanog sustava

Projektiranje kontrola I/O sustava za industrijske strojeve i naprednu opremu je zahtjevan zadatak. Projektni timovi moraju isporučiti isplative i profitabilne proizvode, a istodobno ispunjavati stroge tehničke uvjete i visoka očekivanja. Izvedba ugrađenih projekata po narudžbi je velik izazov za male timove i nosi značajne troškove u pogledu vremena, resursa i stručnosti. Duljina i složenost tih projekata također sprečava napredovanje u obradi i raznolikim računalnim elementima, jer napredak u silicijskoj industriji nadmašuje trenutne projektne cikluse. Proces se dodatno komplicira kada se moraju uključiti brojni hardverski podsustavi, obično je to od više proizvođača i programi tih podsustava s različitim softverskim alatima.

Postoje bolji načini za projektiranje kontrola I/O sustava za napredne strojeve i opremu. Način kojim se pojednostavljuje kompleksnost, poboljšava integracija i smanjuje rizik. S izdržljivom, pouzdanom i industrijski dokazanom povezanom projektnom platformom koja je u potpunosti prilagodljiva putem softvera, timovi mogu ići na višu razinu integracije u njihovom projektiranju i provesti manje vremena na integraciji sustava i projektiranju, te više vremena posvetiti vrijednostima dodatnih značajki i konkurentnim diferencijacijama [3].

3.2. Mogućnosti uređaja

CompactRIO platforma kombinira izmjenjiv hardver, prilagodljiv softver i otvoren IP podrške. Kod spajanja raznih uređaja kao što su: kamere, motori, pogona, zasloni, i senzori izravno na CompactRIO nisu mu potrebni odvojeni podsustavi ili kontroleri za projektiranje. U mogućnosti je iskoristiti kompletan i integriran niz softverskih alata za prilagodbu funkcionalnost i upotrebu snage putnog FPGA za naprednu obradu i složene kontrole, čak i bez poznavanja hardverskih jezika kao što VHDL ili Verilog [3].



Slika 3.1. Razni moduli za spajanje na CompactRIO platformu [3].

Slika 3.1. nam prikazuje koje sve uređaje možemo spojiti na kontoler (industrijske kamere, senzore pokreta), te koje mogućnosti kontoler posjeduje (obradu i analizu podataka,integrirani softver, analogno/digitalne ulaze i izlaze, praćenje, prikazivanje i bilježenje podataka).

3.3. Senzor pokreta

Također se CompactRIO može koristiti kao senzor pokreta visokih performansi. Uz izvornu podršku za NI i druge motore i pogone, CompactRIO pruža mogućnost spajanja direktno na prenosiv hardver i koristiti povezani pokretni IP za simulaciju i konfiguraciju postavki osovine, podešavanje motora, te provoditi prilagođene algoritme za upravljanje bez potrebe za odvojenim senzorom pokreta. S mogućnosti korisničkog programiranja FPGA na kontroleru CompactRIO mogu se projektirati visoko determinističke i jako složeni algoritmi kontrole pokreta bez znanja tradicionalnih opisa FPGA hardvera i jezika kao što VHDL ili Verilog.

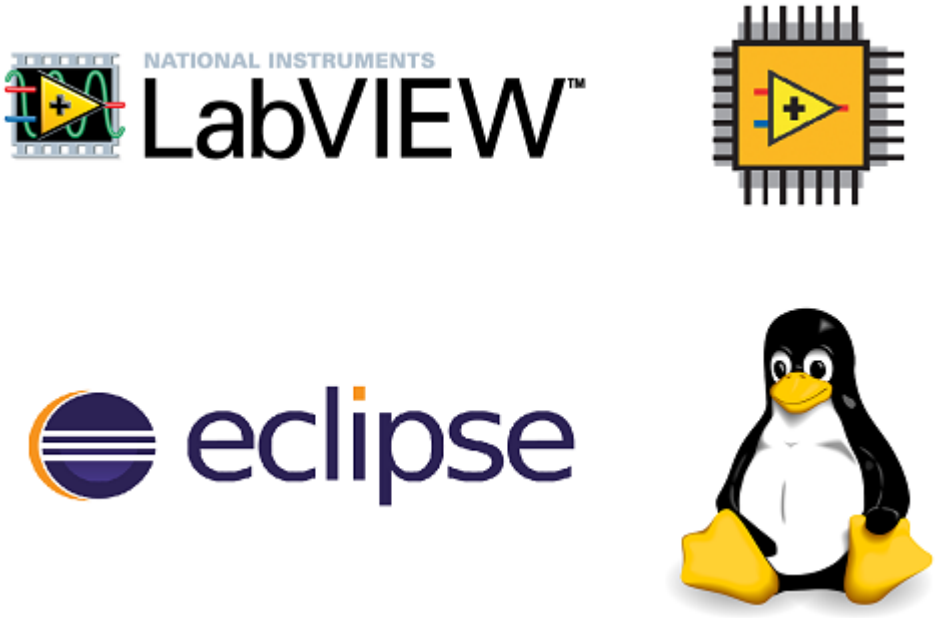
CompactRIO kao kontroler pokreta može prilagoditi funkcionalnost pokreta na najnižu razinu kontrole IP kako bi zadovoljio specifične potrebe za programe i integriranjem svojih algoritama s mjerenjem I/O i stjecanjem vizija povratne sile i viziju vođena zahtjeva pokreta [3].



Slika 3.2. Direktno povezivanje ulazno/izlaznih modula na CompactRIO [3].

3.4. Ugrađeni softver

Moguće je definirati i objasniti funkcionalnost CompactRIO sustava s intuitivnim softverom koristeći jedan, visoko optimiziran niz alata za svaku fazu projektnog ciklusa od modeliranja i simulacija, do izrade prototipa, ovjeravanja i razvoja. NI softver smanjuje rizik i povećava produktivnost pomoću eliminiranja potrebe za stvaranjem i održavanjem I/O upravljačkih programa, struktura i dr. CompactRIO kontrolori koriste testiran NI Linux Real-Time, Real-Time OS na temelju standardnih Linux tehnologija koje kombiniraju pristupačnost radne površine Linuxa sa sigurnošću i pravovremenim performansama potrebnim za industrijske primjene. A s otvorenošću NI Linux Real-Time moguće je programirati procesor s LabVIEW ili C/C++ programima kroz izvornu podršku za Eclipse programsko okruženje. Korištenjem grafičkog LabVIEW okruženja za programiranje FPGA i iskorištenje odlične snage FPGA čak i bez znanja opisa hardverskog jezika kao što su VHDL ili Verilog [3].



Slika 3.3. Logo i ikone programa koji se koriste za programiranje CompactRIO platforme [3].

3.5. Praćenje i analiza podataka

CompactRIO može pomoći pri obavljanju naprednog bilježenja podataka ili digitalnog nadzora čak i u najekstremnijim uvjetima. Platforma nudi vrlo pouzdana, lagano primjenjiva rješenja za kontinuirano praćenje najpotrebnijih uređaja. Uz CompactRIO se može provesti predviđeni plan održavanja kako bi se povećao vijek trajanja uređaja. CompactRIO platforma nudi izvorni API za pohranu podataka, spremnik podataka ili virtualni oblak za daljnju analizu i skladištenje.

Arhitektura CompactRIO platforme omogućuje obradu i analizu prikupljenih podataka u stvarnom vremenu kako bi se donijele ispravne odluke brzo. U mogućnosti je iskoristiti prednosti stotine izgrađenih procesa za obradu signala, analiza, kontrola i matematičkih funkcija koje ubrzavaju razvoj raznih problema.

Bilo da je zadatak napredne analitike, obrade signala ili složene matematike, CompactRIO se može nositi i s zadacima koji njegovo rad dovode do samog ruba iskorištenja, što čini CompactRIO najfleksibilnijim inteligentnim uređajem za IIoT [3].

3.6. Prikazivanje podataka preko zaslona i komunikacija s protokolima

CompactRIO je u mogućnosti vizualizirati svoje podatke s kombinacijom lokalnog, udaljenog ili mobilnog zaslona. Bilo da se koristite jednostavne zaslonske komponente kontrolirane preko RS232, visoko rezolucijskih kontrolnih uređaja priključenih izravno na kontrolor, na udaljeni uređaj ili mobilni internetski servis, može se projektirati interaktivan, značajkama bogat GUI te ih se može spojiti izravno na CompactRIO za ujednačenu vizualizaciju podataka i operatorsko sučelje, bez potrebe za dodatnom HMI kontrolom [3].



Slika 3.4. Vizualizacija podataka izravnim spajanjem na uređaj bez potrebe za dodatnim posrednicima [3].

Uz izvornu podršku za zajedničke industrijske protokole kao što su PROFINET, OPC UA, EtherCAT, CompactRIO se može povezati s različitim uređajima, opremom i infrastrukturom. CompactRIO je idealni IIoT uređaj koji može djelovati kao prevoditelj između bilo kojih od gotovo 20 podržanih protokola, pa čak i prilagođene komunikacijske protokole kroz FPGA [3].



Slika 3.5. Protokoli koje podržava CompactRIO [3].

3.7. Analogni i digitalni ulaz /izlaz (I/O)

Uz CompactRIO platformu se već duže razdoblje stječu kvalitetni signali pomoću mjerenja specifičnih stanja signala, ugrađenom izolacijom i industrijskim I/O spojenim na široku lepezu senzora i signala. Može se odabrati između više od 150 NI i drugih I/O modula za mjerenje, kontrolu i komunikaciju koji služe za povezivanje aplikacije na bilo koji senzor ili sabirnicu [3].



Slika 3.6. Moduli [3].

Primjeri modula koji se mogu koristiti s NI CompactRIO:

- NI cRIO-9951- Služi za obrazovne svrhe, ne uključuje podršku niti certifikate



Slika 3.7. NI cRIO-9951 [3].

- NI cRIO-9953- Sadrži 10 ćelija za spajanje i 15-pin D-Sub za spajanje na CompactRIO



Slika 3.8. Ni cRIO-9953 [3].

- NI cRIO-9954- Sadrži mjesta za 25-pin D-Sub konektore, PCB desni 15-pin D-Sub konektor za spajanje na CompactRIO



Slika 3.9. Ni cRIO-9954 [3].

- NI cRIO-9955- Uključuje mjesta za 37 pozicijski D-Sub konektor s vijcima, 15 pozicijskih D-Sub konektor za spoj s CompactRIO i 10 ćelija
- NI cRIO-9956- Posjeduje 10 ćelija i mjesta za BNC konektore i PCB 15 pozicijske desne D-Sub konektore [3].

Uz dodatno ugrađeni signalni uređaj, ti moduli imaju brojne mogućnosti za povezivanje senzora, uključujući screw-terminal, BNC i D-sub spojeve. C Series moduli su dizajnirani da rade u najtežim uvjetima s proširenim temperaturnim rasponom od -40 ° C do 70 ° C (-40 ° F do 158 ° F), pružajući potrebnu sigurnost, elektromagnetsku kompatibilnost (EMC), zadovoljavaju ekološki certifikat i zahtjeve. Ovi moduli se mogu koristiti za povezivanje senzora na CompactRIO koji je siguran za zadovoljavanje traženih potreba. Tablica daje potpuni popis I/O mogućnosti za CompactRIO platformu [3].

Tablica 3.1. I/O mogućnosti za CompactRIO platformu [3]

Signal Type	Channels	Measurement Types	Max Rate	Special Features
Analog Input*				
Voltage	4, 8, 16, 32	±200 mV, ±1 V, ±5 V, ±10 V, ±60 V, 300 V_{rms}, 400 V_{rms}, 800 V_{rms}	1 MS/s/ch	Ch-ch isolation, high-voltage bank isolation, anti-aliasing filters
Current	4, 8, 16	±20 mA, 5 A_{rms}	200 kS/s	Anti-aliasing filters, ch-ch isolation, built-in shunt resistors
Universal	2, 4	V, mA, TC, RTD, strain, Ω, IEPE	51.2 kS/s/ch	Excitation, bridge completion, anti-aliasing filters, ch-ch isolation, built-in shunt resistors, amplification
Thermocouple	4, 16	J, K, T, E, N, B, R, and S types	75 S/s/ch	CJC, bank isolation, amplification, filtering
RTD	4	100 Ω	400 S/s	50/60 Hz filtering
Strain/Bridge Based	4, 8	¼, ½, full bridge (120 or 350 Ω)	50 kS/s/ch	Excitation, bridge completion, anti-aliasing filters
Acceleration and Sound	3, 4	±5 V, ±30 V	102.4 kS/s/ch	IEPE, anti-aliasing filters
Analog Output**				
Voltage	4, 16	±10 V	100 kS/s/ch	Ch-ch isolation, bank isolation
Current	4	0 mA–20 mA	100 kS/s/ch	Open-loop detection
Digital /IO				
Input	4, 8, 16, 32	LVTTL, 5 VTTL, 12 V, 24 V, 30 V, 250 VDC/VAC;	55 ns	Ch-ch isolation, bank isolation, sinking and sourcing
Output	4, 8, 16, 32	LVTTL, 5 VTTL, 12 V, 24 V, 60 V, 0 V–50 V programmable;	55 ns	Ch-ch isolation, bank isolation, sinking and sourcing
Input/Output	4, 8, 16, 32	LVTTL, 5 VTTL, 12 V, 24 V	55 ns	Ch-ch isolation, bank isolation
Relays	4, 8	30 VDC, 60 VDC, 250 VAC; SPST and SS	1 op/sec	Ch-ch isolation, bank isolation
Specialty				
Motion	1, 1	step/dir, CW/CCW, analog PWM	—	Stepper and servo drive signals, incremental encoder feedback
Synchronization	1,3	Cabled, GPS	—	Pulse per second (PPS) accuracy of ±100 ns for multichassis synchronization

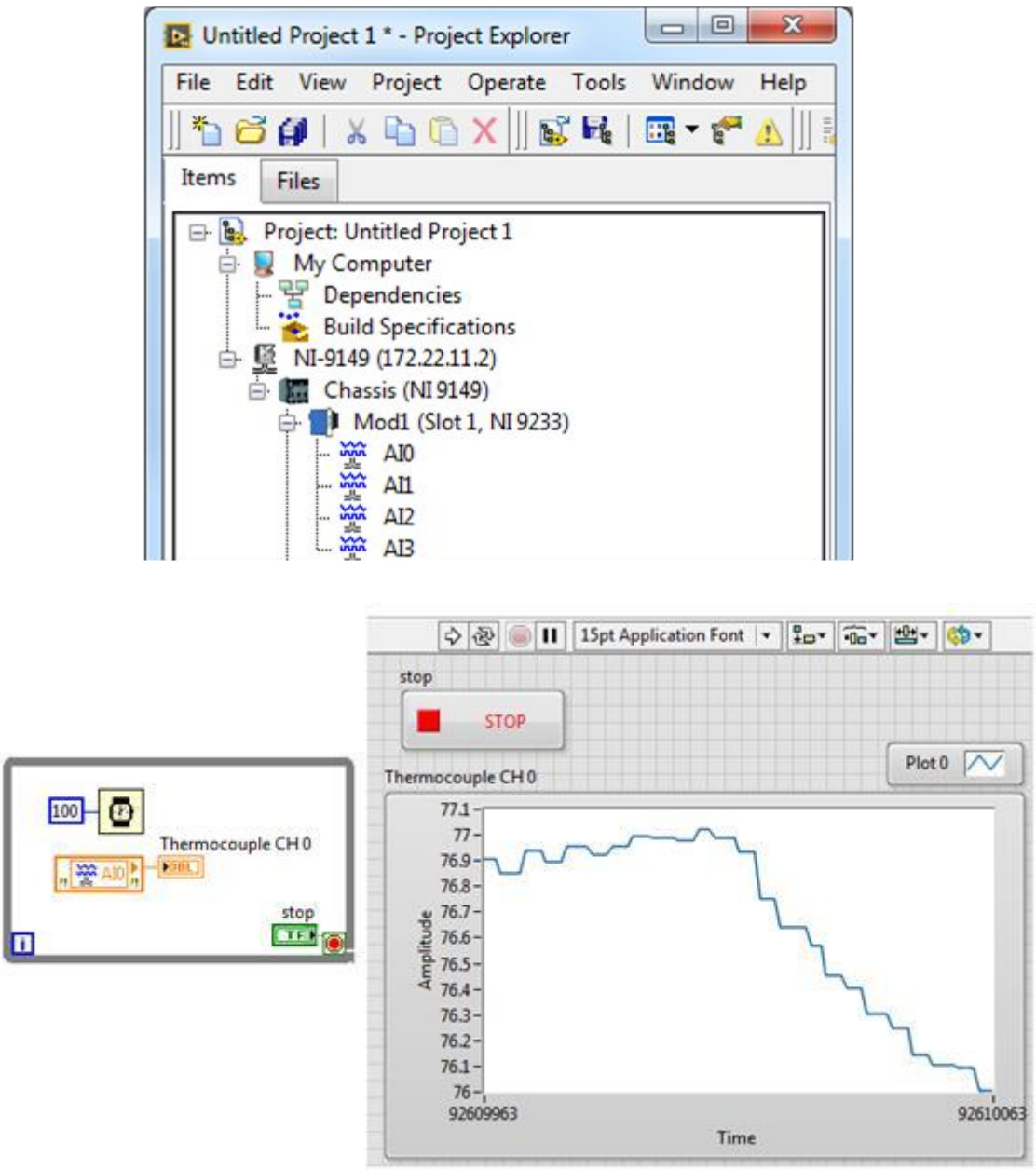
* Up to 24 bit resolution
** Up to 16 bit resolution

4. POVEZANI SUSTAV PROGRAMA C/C++ I LabVIEW

4.1. Pregled

Postoje različite mogućnosti odabira NI s mogućnosti izmijene I/O (RIO) arhitekture za napredne i kritične aplikacije, jer NI ugrađeni uređaji kombiniraju dva vrlo pouzdana cilja obrade: real-time OS (RTOS) i programabilni hardver, sa softverskim alatima potrebnim da bi ovu arhitekturu napravili pravim izborom za povezani sustav projekata.

NI Platforma je otvorena i fleksibilna, tako da se može birati između raznih programskih jezika, alata i projektnih obrazaca za provedbu sustava i najbolje iskorištenje vještina. Unaprijed testiran povezani OS temelji se na Linuxu i izgrađen je sa standardnim OpenEmbedded tehnologijama, tako da se mogu pisati programi koristeći LabVIEW, software, C/C++, textual math ili nekom njihovom kombinacijom. Mogu se jednostavno iskoristiti algoritmi iz prošlih projekata kako bi smanjili nepotrebne razvojne napore, pa čak i integrirati LabVIEW s bibliotekama i aplikacijama pisanih u bilo kojem jeziku. LabVIEW FPGA modul pruža visoku razinu programiranja koji koristi Xilinx kompajler za izgradnju optimiziranih FPGA programa. Programeri se tada na razini aplikacije mogu povezati s I/O i odmah pisati vrlo snažne algoritama u grafičkom jeziku bez VHDL iskustva u programiranju. Ugrađeni RIO programi upravljaju komunikacijom između procesora i FPGA, koja oslobađa vrijeme kako bi se više usredotočili na logiku aplikacije, a ne na manje bitne programe softvera [4].

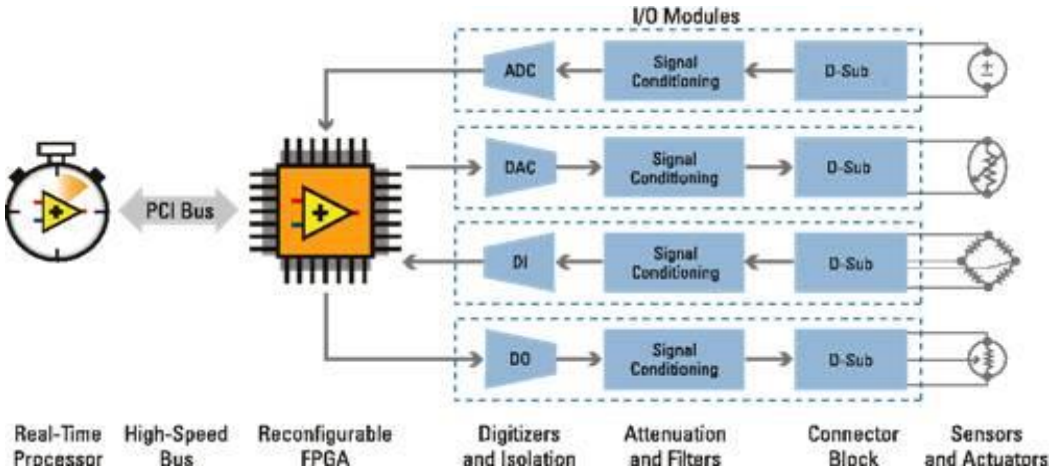


Slika 4.1. Prepoznavanje i programiranje I/O modula u LabVIEW

4.2. Uvod u NI povezanu sistemsku građu

NI RIO arhitektura kombinira rad procesora RTOS s korisnički programiranim FPGA u raznim hardverskim čimbenicima koji uključuju CompactRIO i Single-board RIO sustav na modulu (SOM). Može se koristiti njegova arhitektura kako bi se optimizirao dizajn softvera za izvedbu i determinizam, jer kod može biti podijeljen preko tih raznovrsnih procesa obrade koji se temelje na kritičnosti pojedinih zadataka. Na primjer, vršna stanja stroja, mreže i periferne komunikacije mogu raditi na pravovremenoj strani. FPGA može podržavati I/O, nadzore brzina, sigurnosnu

logiku, obradu signalu, i prilagođene protokole. DMA sabirnica velike brzine između procesora i FPGA podnosi niske latencije prijenosa podataka, prekida i naredbi [4].



Slika 4.2. NI RIO arhitektura kombinira procesor, FPGA i I/O s visoko razvijenim programskim alatima [4].

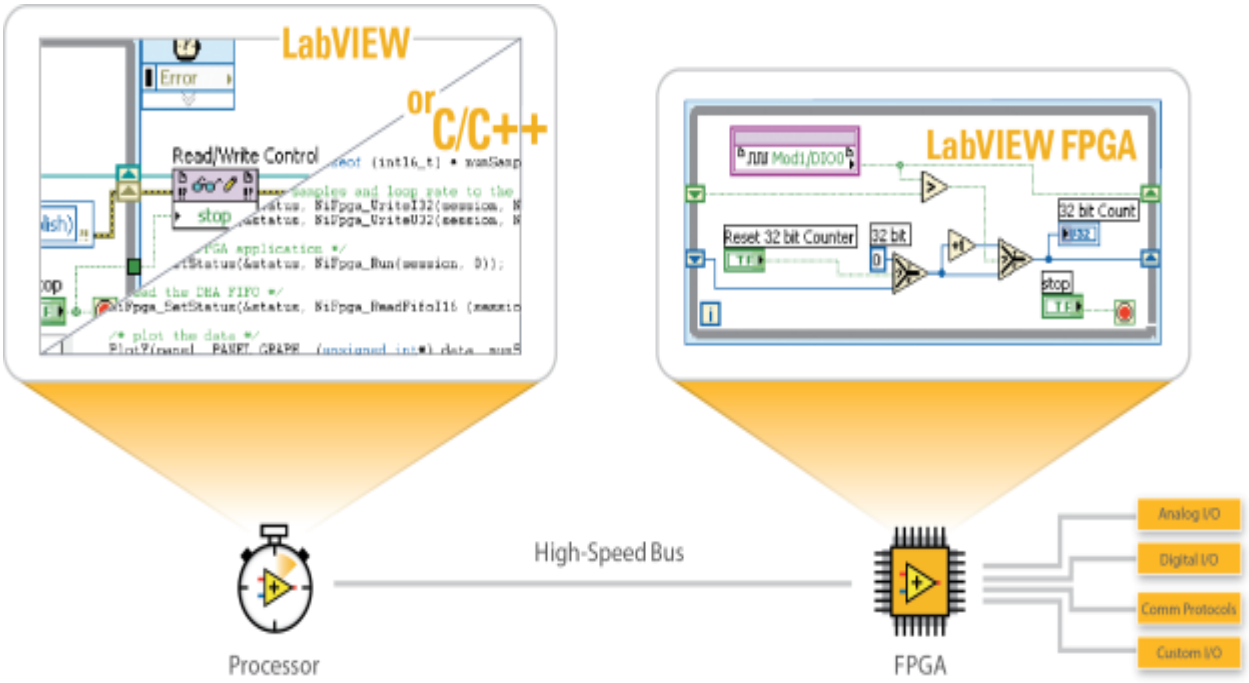
Povezani procesor s RTOS**:** NI RIO arhitektura temelji se na najvišoj kvaliteti, industrijskom standardu ARM ili Intel procesorima sa sustavom NI Linux Real-Time. NI Linux Real-Time je Linux distribucija temeljena na standardnim OpenEmbedded tehnologijama s povezanim, industrijski kvalitetnim, pravovremenim sposobnostima kroz PREEMPT_RT patch set. NI Linux Real-Time je dizajniran posebno za pouzdane i determinističke operacije u dugoročnim implementacijama, a istovremeno nudi sigurnosna poboljšanja i povećanja otpornosti na rušenje aplikacija [4].

FPGA: Korisničko programirani Xilinx FPGA, reprogramirljiv silikonski čip, dodaje peformanse i pouzdanost potrebnu za velike brzine kontrole, urednu obradu podataka ili složeni proračun vremena i okidanja. Za razliku od procesora u osobnim računalima, programiranje FPGA zahtjeva čip koji provedi svoju funkcionalnost prije nego li se pokrene softverska aplikacija. Svaka zasebna obrada zadataka je dodijeljena posebnom dijelu čip i može funkcionirati samostalno, bez utjecaja drugih logičkih blokova. Kao rezultat toga, svojstva jednog dijela zahtjeva ne utječu kada se dodaje više procesa [4].

4.3. Zahtjevi programiranja

Dok su raznovrsne arhitekture koje kombiniraju procesor i FPGA od iznimne važnosti za napredne i kritične aplikacije, programiranje takvih sustava je zahtjevan. Problemi postoje u svakoj fazi i uključuje odabir ili izgradnju operacijskog sustava, razvoj upravljačkih programa za komunikaciju s I/O i komuniciranje između procesora i FPGA, kao i uporabu vještina potrebnih za pisanje FPGA programa i zadržavanje razvojnih ciklusa ciljeva u sinkroniziranju tijekom trajanja projekta.

NI RIO arhitektura rješava probleme raznovrsnih softverskih projekata isporukom prethodno ovjerenih softvera i razvijenih programskih alata koji se mogu nadograditi, a da pritom zadrže otvorenost i prenosivost za buduće platforme. Arhitektura osigurava potpuno testirane programe i Linux OS tako da se može brzo spojiti aplikacijski kod na NI platformu. I dalje se mogu nastaviti koristiti preferirani programski alati za razvoj i iskoristiti prednosti LabVIEW Real-Time i FPGA programiranja. Prava vrijednost LabVIEW FPGA programiranja je ta da ovaj zamišljen pristup digitalnom dizajnu osnažuje aplikacijski nivo C/C++ programeru za provedbu najkritičnijih dijelova njihovih zahtjeva u hardveru, koje možda nikada nisu učinili na neki drugi način bez da su imali VHDL stručnjaka [4].



Slika 4.3. Programiranje s pomoću LabVIEW ili C/C++ i korištenje NI RIO programa za komuniciranje s LabVIEW FPGA [4].

4.3.1. Real-Time Programiranje

Dizajneri povezanih sustava koji rade s NI hardverom mogu birati između raznih programskih jezika i alata kada rade svoje sustave. Programeri mogu programirati otvoren povezani procesor s LabVIEW, C/C++ ili s textual math, te mogu ponovno iskoristiti izvorni kod iz prošlih projekata kako bi uštedjeli vrijeme razvoja. Programeri mogu ponovno koristiti C/C++ biblioteke iz prošlih projekata pozivajući ih iz LabVIEW Real-Time aplikacija, što olakšava zadatke kao što je pravovremeno planiranje. Također mogu razviti, ispraviti i unaprijediti aplikacije na povezanom procesoru koji je u potpunosti napisan u C ili C++ koristeći Eclipse ili neku drugu preferiranu integriranu razvojnu okolinu (integrated development environment) [4].

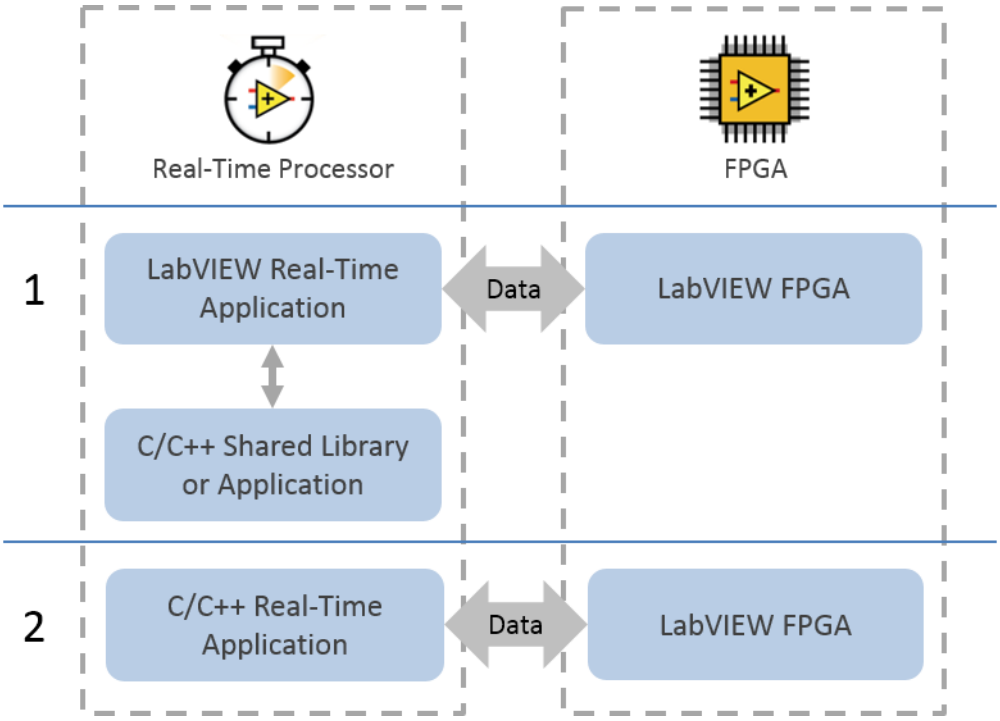
4.3.2. FPGA Programiranje

Uvođenje FPGA u povezani dizajn sustava omogućuje stvaranje digitalne logike visokih performansi, bez potrebe za prilagođenim ASIC dizajnom. Tekstualni opis hardverskih jezika kao što VHDL ili Verilog napisanih od programskih inženjera korišteni su za FPGA razvoj. Zbog izazovnih jezičnih semantika ti opisi hardverskih jezika otežavaju iskoristiti sve mogućnosti FPGA dizajna. O tome svjedoče povezani industrijski istraživači za pronalaženje veće zamišljene razine za FPGA dizajn, kao što su C-to-gates alati ili grafički programski jezici kao što je G u LabVIEW FPGA.

LabVIEW grafičko programiranje oduzima složenost FPGA hardvera i donosi programiranje poznate konstrukcije softverskih inženjera, tako da korisnici mogu dizajnirati ugrađeni sustav velikih performansi puno brže [4].

4.4. Odabir najpouzdanijih programa

Budući da se NI RIO arhitektura temelji na standardnom Linux OS, mogu se napraviti povezane aplikacije koje koriste bilo koji Linux kompatibilni alat. To daje dodatnu fleksibilnost za ponovnom uporabom IP, znanja i razvojnih alata, ali mora se poznavati gdje i kako se oni mogu integrirati s LabVIEW Real-Time i LabVIEW FPGA [4].



Slika 4.4. Ponovna uporaba već napisanih algoritama [4].

Mora se koristiti LabVIEW FPGA za programiranje Xilinx FPGA, ali također može se koristiti C/C++ ili LabVIEW Real-Time za komuniciranje s LabVIEW FPGA programom. Tu su i C API i LabVIEW API za komunikacijski dio između procesorskih aplikacija i LabVIEW FPGA programa. Najvažniji korak u projektiranju NI Linux real-time sustava je odabir najrazvijenijih softverskih programa.

Odabir LabVIEW za programiranje pravovremene (real-time) aplikacije ima mnoge prednosti, u tome LabVIEW real time uključuje: povezane jednadžbe za višestruke nizove, real-time pravilno planiranje, i mnoge druge značajke koje su specifične za izgradnju snažnog, determinističkog koda. Osim toga LabVIEW može prikazati sučelje s podijeljenim bibliotekama i aplikacijama pisanim u drugim jezicima, tako da se ova vrijedna IP može ponovno koristiti u LabVIEW real-time dizajnu uzorka.

U slučajevima kada napredne aplikacije moraju biti napisane isključivo u C/C++, može se koristiti LabVIEW samo za programiranje I/O u FPGA, i procijeniti koji kritični zadaci trebaju biti premješteni za izvršavanje u hardver, umjesto u procesor [4].

Sredstva za početak rada:

1. Poznavanje mogućih uzoraka softvera za vaš NI povezan sustav
2. Postavljanje algoritma za izgradnju, implementaciju i ispravljanje pogrešaka za C/C++ zajedničku biblioteku ili koristeći naprednu aplikaciju
3. Pozivanje C/C++ dijeljene biblioteke iz LabVIEW Real-Time aplikacije
4. Preuzimanje i sastavljanje primjera da se vidi kako komunicirati s LabVIEW FPGA programom iz C++ u real-time aplikaciji koristeći FPGA sučelje C API
5. Posjećivanje aktivnih razvojnih foruma za odgovore na pitanja u pogledu naprednog C/C++ i Linux programiranja

Za rad su potrebni sljedeći softverski alati:

1. Za programiranje NI Linux real time aplikacije potreban je barem jedan od sljedećih alata:

    1. labVIEW Real-Time module, ako se namjerava izgraditi napredna real-time aplikacija pomoću LabVIEW.

    2. Eclipse C/C++ razvojni alati, ako se namjeravaju izgraditi zajedničke biblioteke ili napredni program koji koristi Eclipse IDE. Ovaj program sadrži GNU/Linux niz kompajlera za oba, ARM i x64 Intel arhitekturu procesora.

    3. Samo GNU/Linux niz kompajlera ako se namjeravaju koristiti vlastiti C/C++ razvojni alati za izgradnju zajedničke biblioteke ili napredne aplikacije
        - GNU C i C ++ Compiler za ARMv7 Linux (Linux)
        - GNU C i C ++ Compiler za ARMv7 Linux (Windows)
        - GNU C i C ++ Compiler za x64 Linux (Linux)
        - GNU C i C ++ Compiler za x64 Linux (Windows)

2. FPGA sučelje C API ako se namjerava graditi napredna real-time aplikacija u C/C++. FPGA sučelje C API generira C zaglavlje datoteke na temelju svojih LabVIEW FPGA datoteka koje izlaže pod nazivom referentne varijable svojih FPGA registara, DMA FIFO i prekida. Također uključuje API funkcije koje se koriste za čitanje i pisanje varijabli i rukovanjem operacija FPGA koda.

3. FPGA razvoj zahtijeva sve od sljedećeg:
    1. LabVIEW
    2. LabVIEW FPGA module
    3. NI-RIO Driver [4].

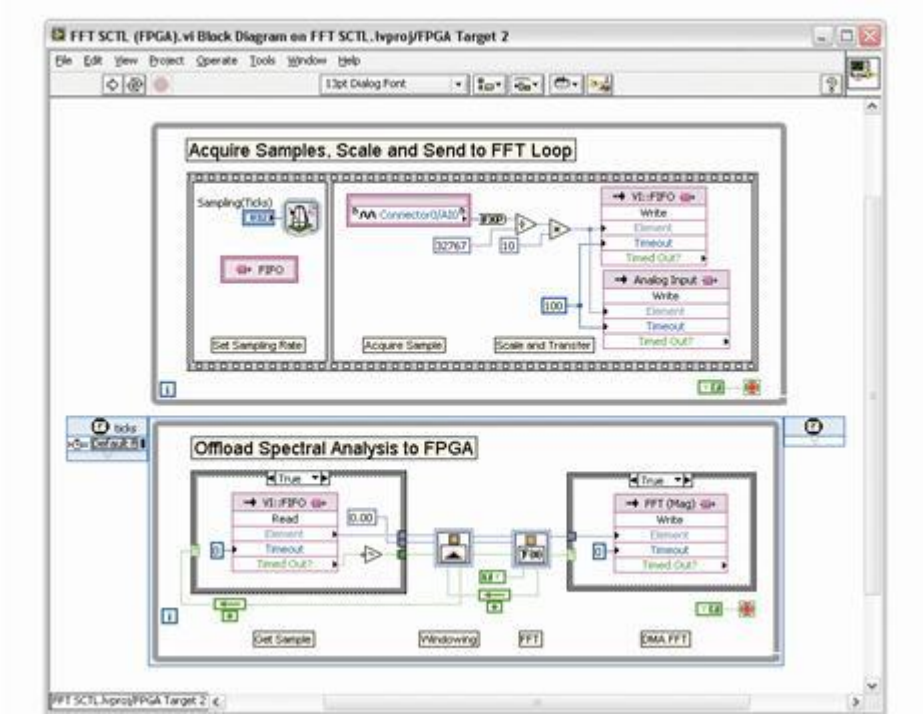
4.5. Korištenje LabVIEW za FPGA programiranje

LabVIEW je izrazito pogodan za FPGA programiranje, jer jasno predstavlja paralelizam i tok podataka. S LabVIEW FPGA inženjeri koji poznaju C jezik, mogu iskoristiti prednosti FPGA u okviru svojih projekata bez tereta učenja hardverskog dizajna. FPGA sučelje C API pruža komunikacijski sloj između FPGA i real-time aplikacije za I/O kontrolu i snimanje podataka. S ovim API-jem, mogu se koristiti LabVIEW grafički alati za programiranje FPGA i odabrati neki od LabVIEW ili C/C++ alata za programiranje procesora unutar sustava.

LabVIEW FPGA pomaže da se stvore prilagođena mjerenja i kontrola hardvera bez nisko razinskih opisa hardverskih jezika. Može se koristiti prilagođeni hardver za: jedinstveno podešavanje vremena i provođenje rutine, ultra brzinsku kontrolu, sučelje za digitalne protokole, digitalne obrade signala, i mnoge druge primjene koje zahtijevaju brz i pouzdan hardver [4].

Sredstva za početak rada:

1. Poznavanje FPGA tehnologije i njezinih mogućnosti upotrebe u aplikaciji.
2. Proučiti razne materijale koji pokazuju kako započeti raditi s LabVIEW FPGA.
3. Pronalaženje LabVIEW FPGA modula [4].



Slika 4.5. Zadaci koji se izvode na namjenskim sekcijama FPGA čipa [4].

5. UPOREĐIVANJE SVOJSTAVA VIŠE KONTOLERA

5.1. Pregled

CompactRIO performans kontroler, na temelju LabVIEW RIO arhitekture s najnovijom tehnologijom kao što su snažan 64-bitni Intel Atom E3800 System-on-Chip (SoC) i Xilinx Kintex 7 FPGA pruža mnoge mogućnosti. Intel Atom SoC pruža dobre performansi i funkcionalnosti, uključujući integrirani GPU i više jezgreni procesor. Kintex-7 325T FPGA ima skoro tri puta više složenih logičkih blokova i preko trinaest puta više različitih DSP slojeva nego kod FPGA u drugim CompactRIO kontrolerima. FPGA i procesor komuniciraju putem PCI Express sabirnice, koja poboljšava propusnost i daje pristup prema 16 DMA kanalima.

LabVIEW RIO arhitektura na CompactRIO performans kontroleru pruža fleksibilnost softvera za konsolidaciju HMI i kontrolne zadaće za isti cilj, ali na jednom operativnom sustavu. To je moguće zbog obilježja više jezgrenih Intel Atom SoC i NI Linux Real-Time OS. Ovaj OS otkriva prioritet zadataka programeru tako da dodijeli veći prioritet kontrolnim zadacima prije nego programskim zadaćama HMI sustava u istoj aplikaciji.

Kombinacija tih tehnologija dramatično poboljšava propusnost sustava i smanjuje latenciju za zatvorene primjene kontrolnih petlji. CompactRIO performans kontroler opremljuje dizajnere s fleksibilnim, razvijenim hardverom, dok smanjuje kompleksnosti sustava i troškove.

Kako bi se dokazale mogućnosti CompactRIO performans kontrolera radi se niz kontrolnih aplikacija i aplikacija za nadgledanje standardiziranih testova. Ovi testovi su jednako primjenjivi na sve dual-core i quad-core varijante kontrolora CompactRIO performansi (NI cRIO-903x) [5].

5.2. Izvođenje testova za usporedbu

Usporedni testovi su dizajnirani da predstavljaju zajedničku kontrolu i praćenje programa, kao i glavne zadaće koje se nalaze u tim programima. U tablici ispod prikazan je pregled testova provedenih u ovoj studiji [5].

Tablica 5.1. Ispitivanja provedena na NI CompactRIO kontrolorima za mjerenje programskih svojstava i propusnost podataka [5].

TEST TYPE	APPLICATION	METRIC
Test1 *Performance*	LabVIEW FPGA Control on CompactRIO Sample Project	CPU Usage vs. Control Loop Rate
Test 2 *Throughput*	Monitoring Application	CPU Usage vs. I/O Data Streaming
Test 3 *Performance*	Complex Real-World Application	CPU Usage vs. Control Loop Rate

Istraživanje je provedeno s obzirom na sljedeće hardvere:

- NI cRIO-9025 | 800 MHz, PowerPC
- NI cRIO-9068 | 667 MHz, ARM Cortex-A9 dual-core
- NI cRIO-903x | 1,33 GHz, Intel Atom dual-core
- NI cRIO-903x | 1,91 GHz, Intel Atom quad-core
- NI cRIO-9082 | 1,33 GHz, Intel Core i7 dual-core

Iako se ovo istraživanje fokusira na CompactRIO performans kontroleru i kako ga uspoređuje s postojećim kontrolorima u istoj grupi kao što je cRIO-9025 u slučaju dual-core varijanti i cRIO-9082 u slučaju quad-core varijanti, predstavnici hardvera iz drugih CompactRIO grupa poput cRIO-9068 također su uključeni kao referentne točke za prikazivanje potpunije slike o rasponu mogućnosti koje CompactRIO sustavi nude [5].

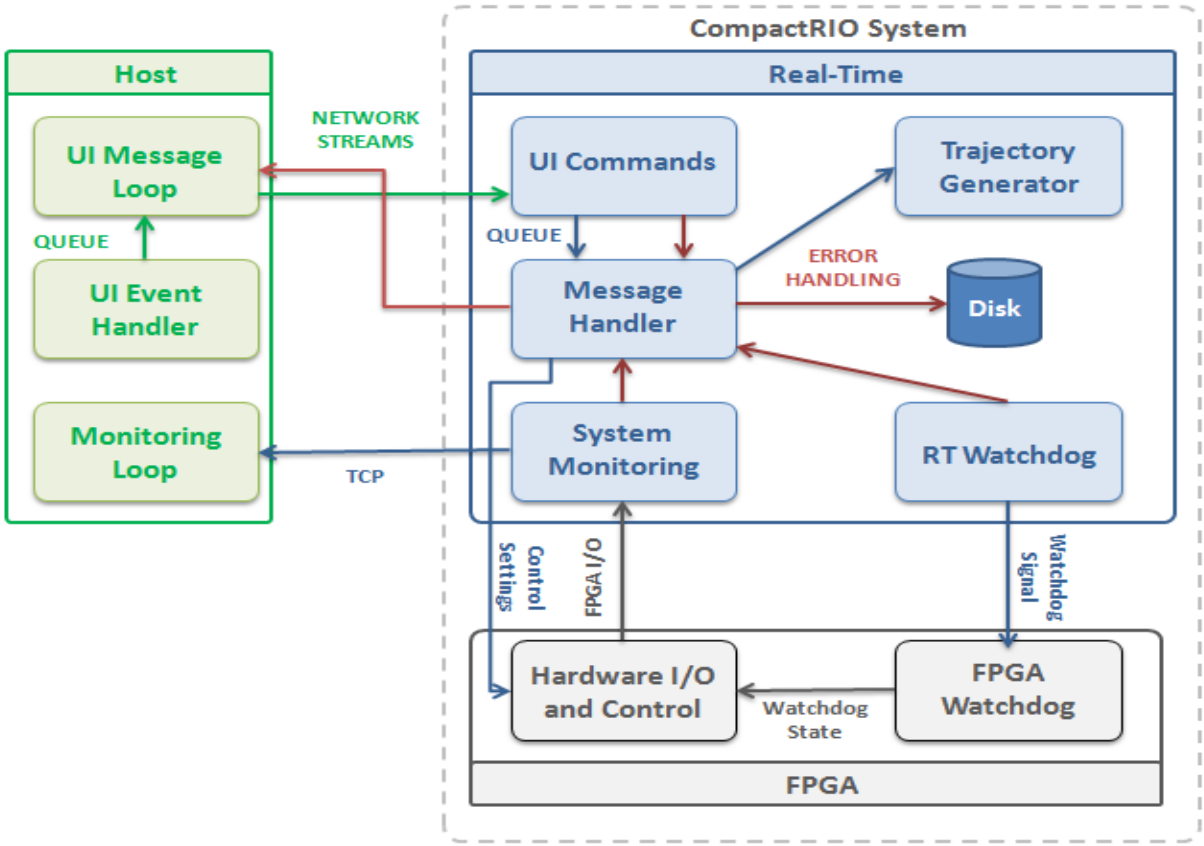
Korištenje CPU koristi kao zajednički mjerni podatak za usporedbu relativnih izvedbi tih CompactRIO kontrolera. Za svaki od ovih testova iskorištenost procesora se mjeri u stabilnom stanju, dok broj kanala za snimanje podataka ili brzina regulacijsko kruga variraju. Ove veličine nam daju ideju o raspoloživim resursima za: dodatni kod, veće brzine petlje ili više kanala za snimanje [5].

Dodatna razmatranja:

- Za CompactRIO sustav koji sadrži višejezgreni procesor, rezultati svake pojedine jezgre se izjednačuju (uzima se prosjek).
- Za CompactRIO performans kontroler, testovi su provedeni s opcijom ugrađenog UI koji mogu i omogućiti i onemogućiti da mjerenje ovih značajki ima utjecaj na ukupnu učinkovitost sustava [5].

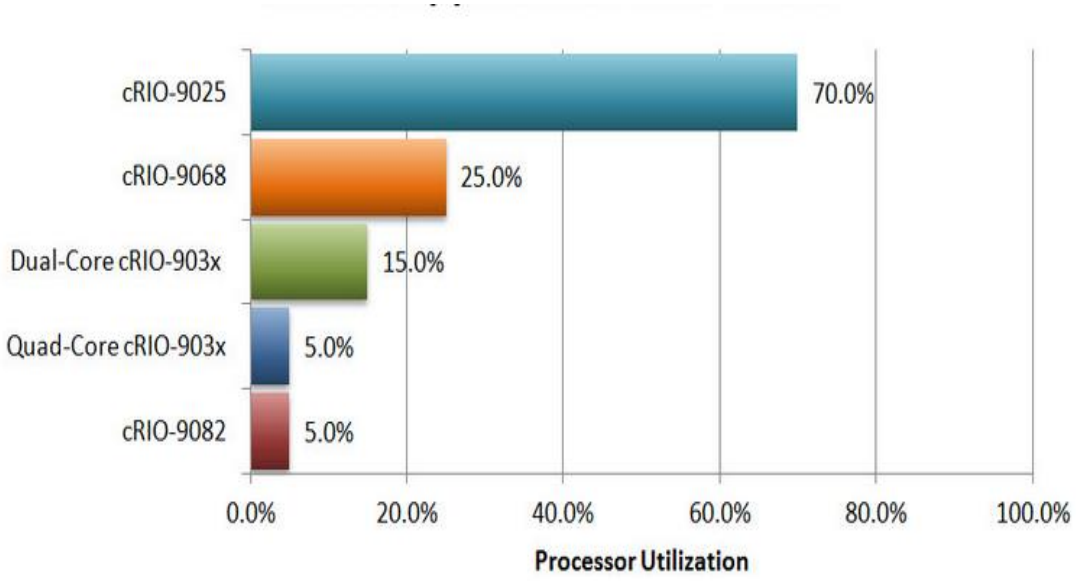
5.3. Test na LabVIEW

Kombinacija tehnologija u novim izvedbama CompactRIO sustava rezultira s poboljšanjem performansi za složene, stvarne aplikacije. Za testirati kontrolni zadatak odabran je LabVIEW FPGA Control on CompactRIO Sample Project. Primjerni projekt je proširen s osam kanala "cubic spline trajectory "generacije algoritma na "Real-Time" kontroleru za daljnje uključenje CPU [5].



Slika 5.1. Dijagram LabVIEW FPGA kontrolera za CompactRIO projekt [5].

Na slici ispod prikazuje se uporaba procesora prilikom pokretanja regulacijskog kruga kod 1,5 kHz. U ovom primjeru, dual-core izvršni kontroler koristi otprilike 15% resursa procesora. To je značajan napredak u odnosu na cRIO-9025 kontroler koji je zahtijevao 70% procesorskog resursa. Isto tako, quad-core izvršni kontroler koristi 5% resursa procesora, što je vrlo slično promatranom cRIO-9082 kontroleru [5].

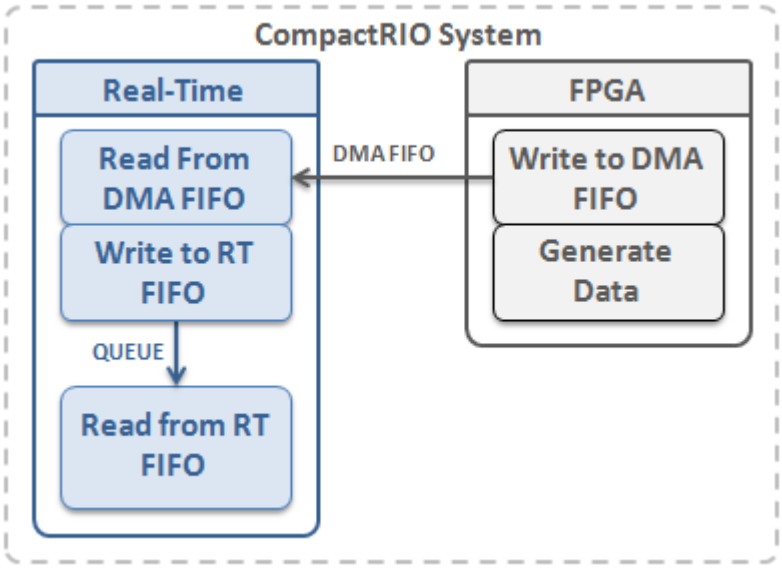


Slika 5.2. Korištenje procesora potrebno za pokretanje kontrolne aplikacije kod 1,5 kHz [5]

Test je ponovljen s ugrađenom UI opcijom dozvoljenom za CompactRIO performans controler, međutim dok god ovaj test ne uključuje HMI dio LabVIEW FPGA kontrole na CompactRIO projektu nema značajnih promjena zabilježenih u CPU korištenju [5].

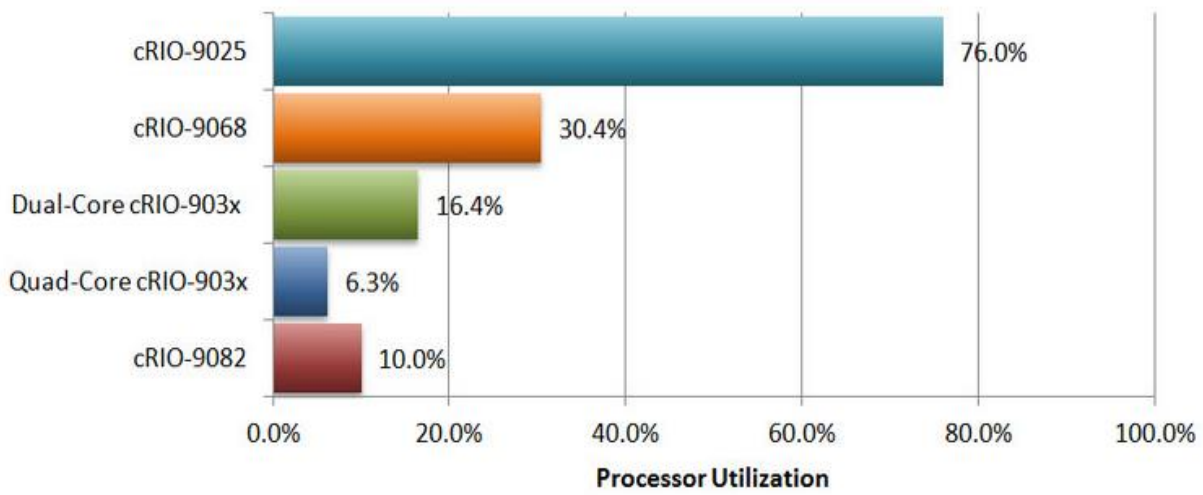
5.4. Test Propusnosti podataka

Zajednički zadatak u praćenju programa je premjestiti podatke s I/O kanala u pravovremene (Real-time) procesore za daljnju obradu ili vizualizaciju. Jednostavan test je stvoren za reprodukciju ovog scenarija i mjerenje korištenja CPU dok varira broj kanala snimljenih podataka. Svaki kanal nosi 16-bitni uzorak prenesen na 100 kHz [5].



Slika 5.3. Dijagram pojednostavljenog prijenosa podataka za primjenu praćenja [5].

Slika u nastavku prikazuje rezultat za prijenos 100 kanala na 100 kHz. U ovom testu su moguće potencijalne nepravilnosti koje proizlaze iz malih razlika u hardverskoj arhitekturi. Na primjer, dok oba cRIO-9068 i CompactRIO performans kontroler imaju 16 DMA kanale ispunjene s različitim tehnologijama sabirnica, oni i dalje predstavljaju značajne, teoretske maksimalne brzine propusnosti: 320MB/s za cRIO-9068 kontroler i 250MB/s za CompactRIO performans kontoler. Međutim, CompactRIO performans kontoler može izdržati veći broj potoka na nižem iskorištenju CPU, jer mu procesor predstavlja bolje performanse [5].



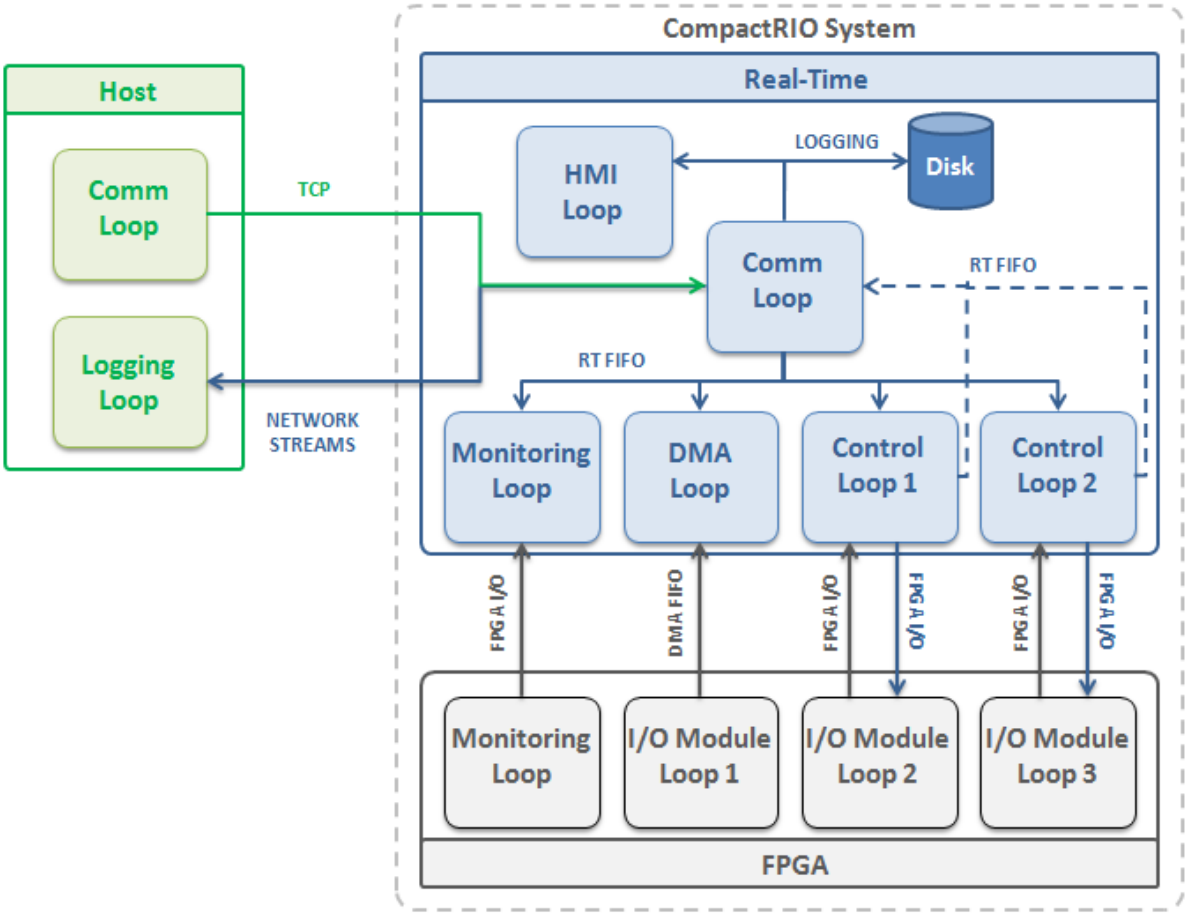
Slika 5.4. Korištenje procesora potrebno za pokretanje kontrolne aplikacije kod 100 kHz [5].

Slično kao i kod prvog testa, učinak od ugrađenih UI opcija na CompactRIO performans kontroleru nije relevantan sve dok test ne uključuje HMI.

5.5. Test na aplikacijama

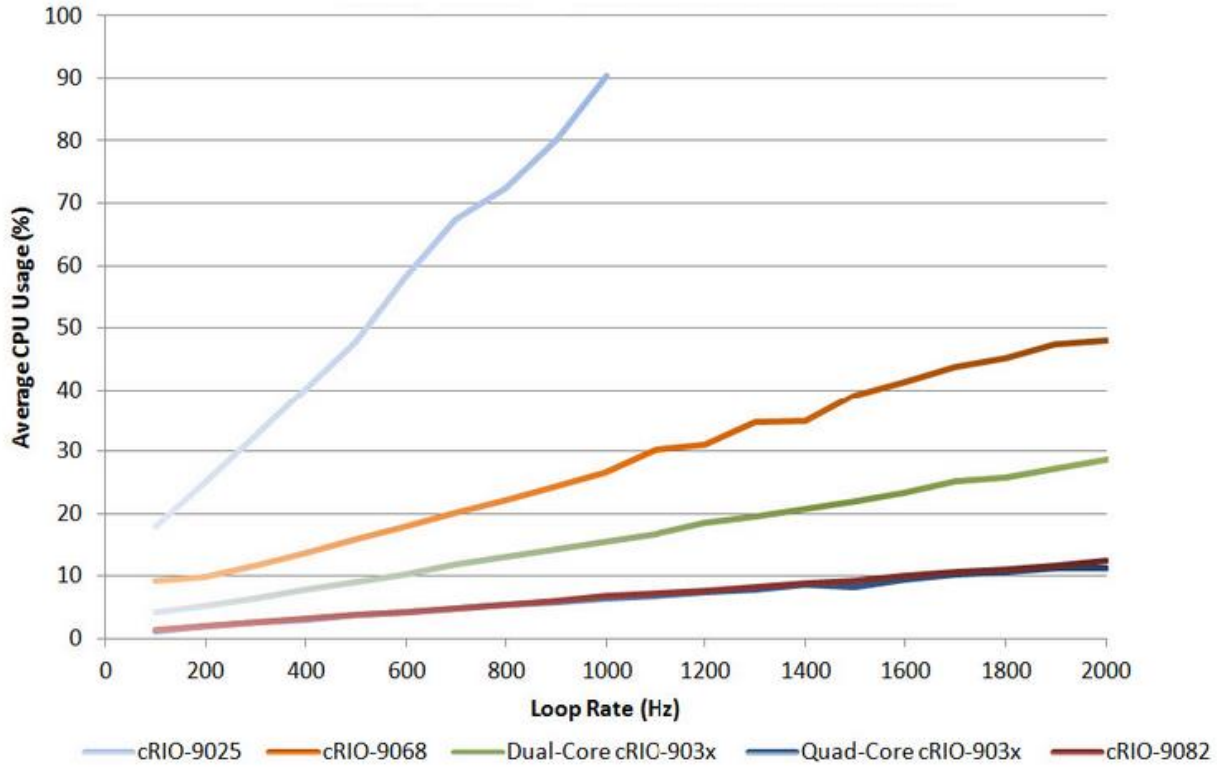
Najbolji način da se u potpunosti ostvare značajke cRIO performans kontolera je kroz test s kompleksnim, stvarnim aplikacijama. Veće aplikacije obično kombiniraju različite vrste zajedničkog nadzora i kontrole zadataka, uključujući i višestruke obrade krugova s više brzinskom kontrolom, obradu podataka, prijenos podataka iz I/O kanala, prijenos na disk, komuniciranje podataka preko mreže na udaljenom HMI, svevremensko izvođenje i praćenje stanja zadataka.

Kompleksna aplikacija koja sadrži zadatke opisane ranije je provedena pomoću zajedničkih LabVIEW izvedbi poput RT FIFOs, vremenskih petlji i mrežnih tokova za koordinaciju i komunikaciju između različitih programskih komponenti. Tu je i zadatak namijenjen za objavljivanje podataka na pokazivač koji se nalazi na prednjoj ploči i koji je na visokoj razini naglasio povezane UI značajke u cRIO performans kontoleru [5].



Slika 5.5. Dijagram složenih kontrola i praćenje primjene [5].

Koristeći ovu aplikaciju NI cRIO-9025 je u mogućnosti postignuti maksimalnu razinu regulacijskog kruga, otprilike 950Hz. U ovom primjeru, iskorištenost procesora je 91%, dok je dvojezgreni kontroler iskoristio tek 15% raspoloživih resursa procesora kako bi dostigao istu razinu. To ostavlja značajnu količinu raspoloživih resursa procesora za dodavanje dodatnih aplikacijskih zadatka, odnosno razina kruga može biti povećan na brzine i preko 2 kHz za ovu složenu kontrolu i praćenje primjene. Višejezgrene varijante pokazuju vrlo slične performanse kod cRIO-9082 na svim promatranim razinama [5].



Slika 5.6. Korištenje CPU [5].

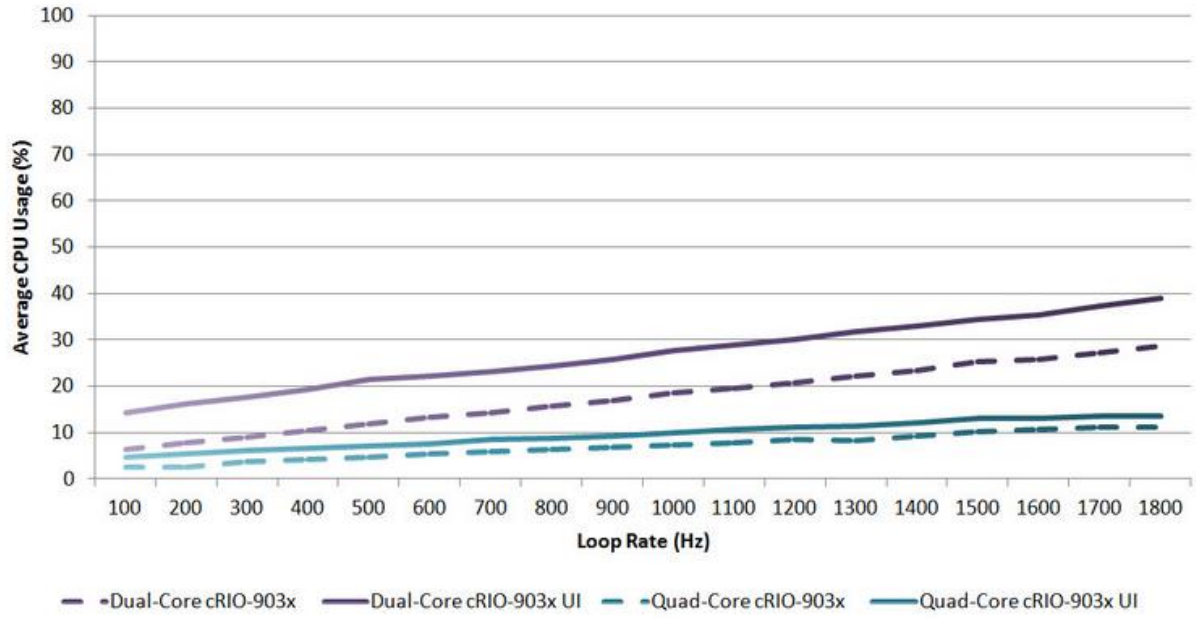
5.6. Utjecaj povezanog sučelja

CompactRIO performans kontoler smanjuje troškove sustava i složenost provedbe lokalnog HMI s povezanom UI podrškom . To je moguće jer novi kontroler uključuje najnoviji Intel Atom procesor s pripadajućom grafičkom podrškom koja pruža temelj za izgradnju logičkog sustava za kontrolu i dodirnu tehnologiju korisničkog sučelja s NI LabVIEW. Da bi se omogućila ta konsolidacija, LabVIEW razvojno okruženje i NI Linux Real-Time OS otkrivaju prioritetni zadatak programeru kako bi dodijelio veći prioritet kontrolnim zadacima nad programskim zadaćama HMI sustavom u istoj aplikaciji.

U tom pristupu resursi se koriste i za nadzor sustava i omogućavanje grafike korisničkog sučelja. To smanjuje troškove i održavanje hardverskog sustava, pojednostavljuje složenost razvoja softvera, ali ne troše dodatna regulatorska sredstva u odnosu na druge opcije prikaza [5].

Za mjerenje utjecaja ugrađenog UI nad procesorskim resursima, Test 3 je izmijenjen stalnom uporabom HMI ažuriranjem njegovih dijelova na najvišu razinu.

Slika u nastavku prikazuje učinak korištenja ugrađenih UI značajka s aktivnim korisničkim sučeljem koji je promatran u testu 3.



Slika 5.7. Konstantan porast uporabe procesora od otprilike 3-10% promatra se kada su značajke ugrađenog UI omogućene [5].

Kada je u potpunosti pod opterećenjem, korištenje povezanog sučelja ima za posljedicu povećanje CPU iskorištenosti oko 10% za dvojezgreni perfomans kontoler i 3% za višejezgreni performans kontroler. Kao i kod testova 1 i 2, učinak na povezanim UI značajkama ostaje konstantan bez obzira na varijacije u broju kanala protjecanja.

Za podršku grafičke sposobnosti povezani UI u Intel Atom SoC koristi se na GPU za proširenje CPU. GPU komunicira s procesorom putem frekventnih procesora prekidanjem signala koji uzima vrijeme obrade od LabVIEW Real-Time aplikacije. Zbog visokih performansi Intel Atom CPU-a to povećanje varijabilnog vremena obrade za ažuriranje GPU je održivo s razinama regulacijskog kruga ispod 6 kHz za primjenu usporedba u testu 3. Za dobivanje veće razine petlje, a još uvijek se koriste povezano sučelje ili GPU može biti onemogućeno, odnosno kontrolni LabVIEW kod može biti usmjeren na FPGA [5].

Za aplikacije gdje bi povezani UI potencijalno mogao ugroziti izvođenje sustava, treba pripaziti na sljedeće:

- Unaprijediti izolaciju zadataka je moguće preko procesorske jezgre raspodjelom izvršavanja zadataka u LabVIEW.
- Promijeniti kontrolni LabVIEW kod hardverske logike u FPGA, ako je potrebna daljnja izolacija i hardverska pouzdanost.
- Onemogući GPU u Intel Atom SoC za smanjenje vibracije na štetu CPU [5].

5.7. Osvrt na CompactRIO kontoler

Povezane aplikacije obavljaju veći broj funkcija kao što su: kontrola pokreta, bilježenje podataka, vizualizacija. NI CompactRIO performans kontroler ima dobre sposobnosti za te primjene s obzirom na kombinaciju tehnologija i jedinstvenih značajki koje ubrzavaju vrijeme razvoja i smanjuju složenost sustava i troškova.

CompactRIO performans kontroler predstavlja značajno poboljšanje performansi u odnosu na druge grupe kontrolera i širi raspon ciljeva kompatibilnih s ključnim tehnologijama kao što je NI Linux Real-Time. NI CompactRIO performans kontroler pruža fleksibilne, snažne hardverske i softverske tehnologije temeljene na LabVIEW RIO arhitekturi za široku paletu povezanih kontrola i praćenje njihove primjene.

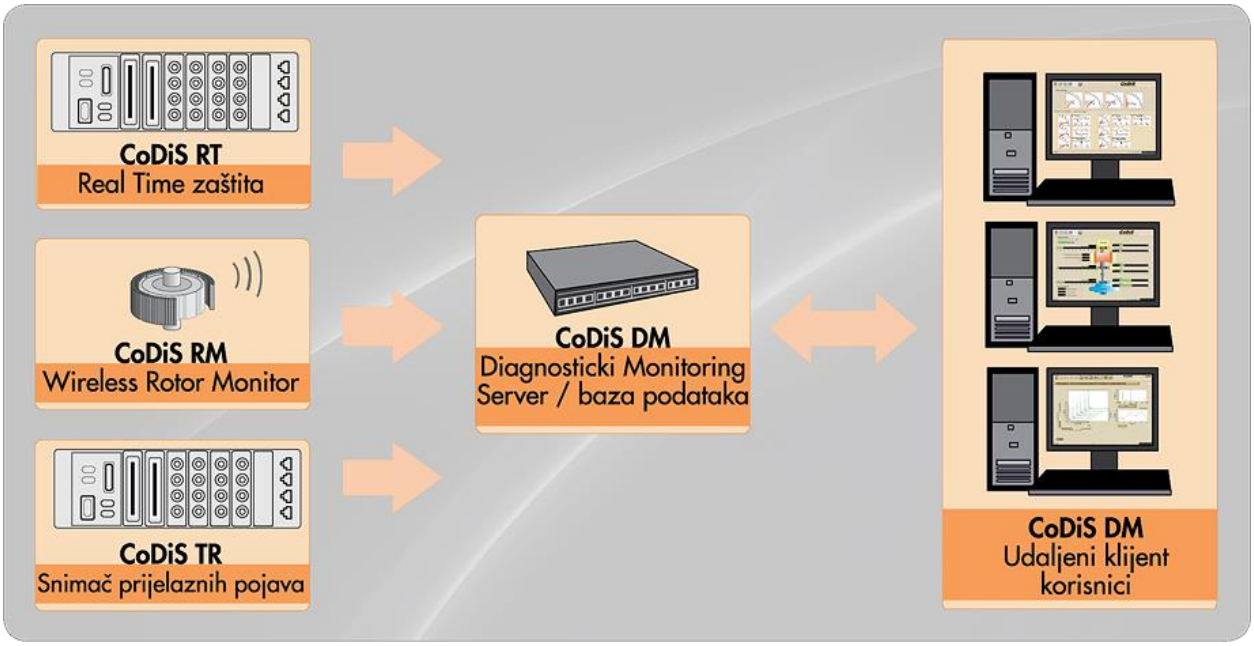
Kroz evo primjere uspoređivanja, CompactRIO performans kontroler pokazuje da ima oko 4-8 puta bolje performanse od postojećih ponuda u usporedivim grupama CompactRIO kontrolera poput cRIO-9025. Osim toga, višejezgrena varijanta ima vrlo sličan učinak na cRIO-908x [5].

6. PRIMJENA CompactRIO UREĐAJA

6.1. Elektrane- promatranje generatora pomoću dijagnostike

Praćenje sustava, je neophodno u proizvodnji električne energije u elektranama gdje se koristi za predviđanje kvarova i utvrđivanje nepravilnosti u izvedbi sustava. Ovi sustavi također se mogu koristiti za pojednostavljenje rutinskog održavanja, te spriječiti dugoročne i skupe kvarove.

Koristi se CoDiS (sustav dijagnostike) na više od 30 generatora u raznim područjima europe. Moderni sustavi industrijskog praćenja obično zahtijevaju značajke koje prikupljaju podatke o vibracijama, temperaturi, tlaku i protoku, a istovremeno održavaju ispravnom bazu podataka za upravljanje. Dizajniran je CoDiS da prati vibracije rotora, statora, uvjete zračnog raspora, električne procese, kvalitetu električne energije i hidrauliku velikih rotacijskih strojeva. CoDiS je modularan i podesiv, a može se osmisliti i prilagoditi prema potrebama. Cijela aplikacija se temelji na LabVIEW, a stvorene su i prilagođene sve analize i postupci baze podataka za prilagođeno praćenje i prikupljanje podataka [6].



Slika 6.1. Nadzor strojeva [8].

Sustav ovisi o CompactRIO za istovremene dijagnostičke funkcije i zaštite, a sastoji se od:

- CoDiS trajnog dijagnostičkog nadzora sustava (DM)
- CoDiS konvencionalnog pravovremenog praćenja za sustave signala i zaštite (RT)
- CoDiS trajnog dijagnostičkog praćenja rotora (RM)

CoDiS-DM	CoDiS-TR	CoDiS-RT	CoDiS-RM
Diagnostic Monitoring	**Transient Monitoring**	**Conventional Monitoring**	**Rotor Monitoring**
Continuous acquisition, analysis, and database recording	Continuous acquisition and trigger analysis recording according to trigger	Continuous acquisition, analysis, and D/A conversion; protective functions	Continuous acquisition and analysis of signals on rotor; wireless transfer of all signals from rotor to stator
Implementation	**Implementation**	**Implementation**	**Implementation**
Detection and measurement of permanent condition change	Transient recording and detection	Application in signaling and protection systems	Application in signaling and protection systems

Slika 6.2. Dijagnostika za različite dijelove sustava [6].

6.1.1. Sistemska konfiguracija

S CompactRIO i NI FieldPoint, napravljen je vrlo podesiv sustav i postignute visoke performanse i pouzdanost. CompactRIO se pokreće pomoću izmjenjive I/O (RIO) FPGA tehnologije i funkcija, istovremeno kao praćenje stanja i zaštite jedinice.

Ovaj sustav praćenja sastoji se od: mjernih komponenti koje uključuju sve senzore i hardver koji se koristi kako bi pridobili podatke, signalnih komponenti,CompactRIO procesnih jedinica s osobnim digitalnim asistentom (PDA) lokalne upravljačke jedinice koja se koristi za prikupljanje i analizu podataka, i središnjeg računala koje sadrži stalnu bazu podataka i radi analize za udaljene LAN korisnike. Sustav također uključuje komunikaciju s kontrolnim nadzorom i sustavima prikupljanja podatka koji koriste analogne izlaze.

CoDiS uključuje stalni nadzor i dijagnostiku relativne vibracija osovine, apsolutne vibracije ležaja, aksijalnog položaja rotora, zračnog raspora minimalne vrijednosti, magnetski tok rotora, električne parametre kao što su struja i napona, fazni pomak, kut snage, ukupna harmonijska izobličenje (THD), temperature i procesne parametre, parametre hidraulike (učinkovitost η), te djelatnu i jalovu snagu [6].

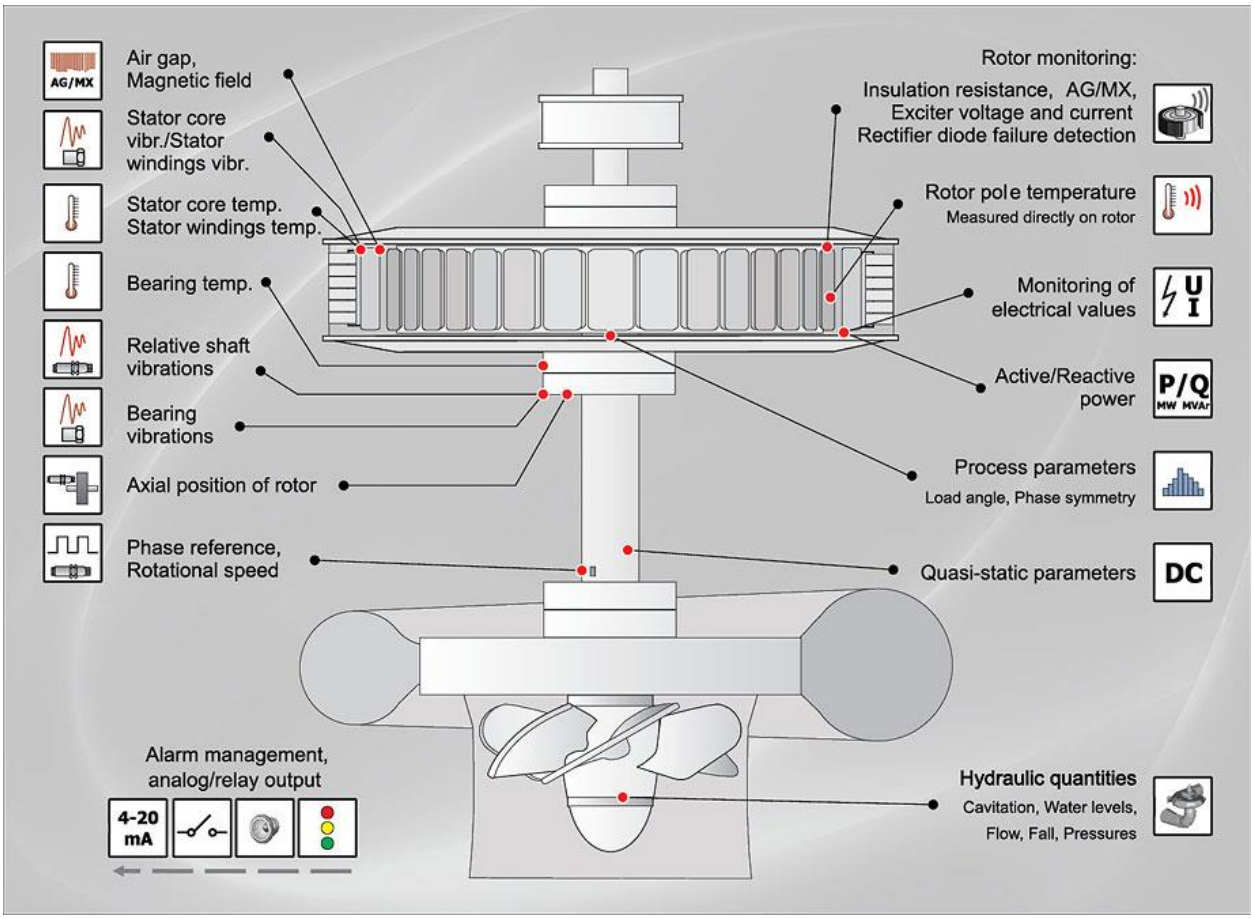
6.1.2. Praćenje rada visokotlačne crpke

Konfiguriran je CompactRIO sustav za praćenje relativne vibracije osovine i brzinu vrtnje koji pruža zaštitu od prevelike brzine. Sustav prati dva hidrogeneratora u kojima svaki generator koristi CompactRIO kao procesne jedinice.

Svaki set praćenja sastoji se od šest senzora pokreta koji se koriste za relativnu vibracija osovine, dva senzora pokreta se koriste za brzinu vrtnje, jedan NI 9201 modul za prikupljanje podataka, dva NI 9265 modula s analognim izlazom za prijenos izmjerenih signala u SCADA, jedan NI 9481 modul kao izvršni dio zaštite, i NI cRIO-9002 s NI cRIO-9101 povezanom, izmjenjivom šasijom.

CompactRIO sustav prati šest analognih ulaznih signala u voltima (tri ležaja u dvije četkice X i Y), dva analogna ulazna signala u voltima (za zaštitu od prevelike brzine), šest analognih strujnih izlaznih signala i četiri prijenosna digitalna izlazna signala.

Svi relevantni podaci, uključujući valne oblike i rezultate analize mogu se dobiti kao zajedničke varijable preko zajedničke mreže [6].



Slika 6.3. Nadzor veličina hidrogeneratora [8].

6.2. Primjena kod zrakoplova

Izgradnja jednog zrakoplova je zahtjevan posao. Uključuje desetke tisuća koraka koji se moraju napraviti i sastaviti. I najmanja pogreška kod sastavljanja može donijeti velike troškove. Zato inženjeri moraju raditi pažljivo ostavljajući jako malo mjesta za pogrešku. Dodavanjem pametnih uređaja kao što je CompactRIO kako bi se pojednostavio proces proizvodnje i uvelo sigurnije upravljanje smanjujemo rizik od pogrešaka [7].



Slika 6.4. Mjerenje sile pritiska na papučice u zrakoplovu

Današnje zrakoplovne tvornice nisu užurbani i bučni pogoni kao što je to bio primjer u prošlosti. Najmodernija tehnika, dizajn i oprema čine modernu proizvodnju učinkovitom, organiziranom i strukturiranom. Ali sve češće pitanje je kako će to sve izgledati u budućnosti. Budućnost u tvornici zrakoplova se temelji na istraživanju i unapređivanju tehnoloških projekata koji su usmjereni usavršavanju nove tehnologije za poboljšanje konkurentnosti zrakoplovnih proizvodnih procesa u kojima ručne operacije dominiraju i dalje.

Budućnost također podrazumijeva opsežnu uporabu modularne platforme s visokom razinom mogućnosti na temelju komercijalnih modula. Jedna od ključnih komponenti za poboljšanje učinkovitosti u tvornici su pametni alati. Ovi pametni uređaji su dizajnirani za komunikaciju s glavnim računalom ili općenito s operatorima ili s drugim alatima, ali samo kada je potrebno

osigurati informacije o stanju i da donesu pravovremenu odluku temeljenu na lokalnoj i distributivnoj inteligenciji u sustavu.

U slučaju proizvodnih pogona, pametni alati mogu pomoći pojednostavljenju procesa proizvodnje i poboljšati učinkovitost rješavanjem problema ili mijenjajući postavke. Inženjeri se moraju usredotočiti na njihove primarne zadatke, te za njihovo rješavanje uvijek moraju biti spremni koristiti odgovarajuće alate. Razvijanje aviona podrazumijeva nekoliko desetaka tisuća koraka koje operateri moraju slijediti sa sigurnošću na mjestu stvaranja kako bi se osigurala sama kvaliteta. Dodavanjem algoritama u sustav, pametni alati razumiju radnje koje operator mora obaviti i automatski prilagođuju alate za odgovarajuće postavke koje pojednostavljuju zadatak za operatera. Jednom kada je radnja završena, pametni alati također mogu pratiti i prijaviti rezultate neke radnje, što poboljšava učinkovitost procesa proizvodnje. Na primjer, s obzirom da podsklop aviona ima otprilike 400.000 komponenata koje treba zategnuti, što zahtijeva više od 1.100 osnovnih alata za pričvršćivanje u tom procesu proizvodnje. Operater mora pažljivo pratiti popis koraka i osigurati odgovarajuće postavke za svaku lokaciju koristeći ispravne alate [7].

Zbog ručnog procesa proizvodnje, ljudske pogreške stvaraju puno rizika. To je značajno jer ako samo jedan element nije odgovarajuće pritegnut moglo bi doći do pogreške koja će koštati stotine tisuća dolara u daljnjim primjenama. Alati za pametno zatezanje razumiju koje zadatke operater mora izvesti pomoću podataka za obradu njegove okoline i automatski postaviti moment. Uređaj može pratiti ishod zadatka u središnjoj bazi podataka kako bi se osiguralo lokaciju za pravilno postavljena.

Sa središnjim izvršnim proizvodnim sustavom baze podataka i distribuirane inteligencije uređaja, proizvodni menadžeri precizno mogu odrediti postupke i procese koji trebaju biti pregledani tijekom kontrole kvalitete i certificiranja. Zrakoplovstvo pokreće razvoj tri pametne grupe alata koje obavljaju različite proizvodne procese: bušenje, mjerenje i kvalitetu zapisivanja podataka te zatezanje [7].

Bušenje

- Procesna okolica s pripadajućim algoritmima
- Provjera materijala za bušenje
- Ispitivanje stanja za rezanje za svaki sloj materijala

- Pratiti dubinu bušenja
- Prijava rezultata bušenja na trenutnoj lokaciji
- Praćenje stanja sustava
- Izvedba automatske provjere/kalibracije

Mjerni alati

- Procesna okolica s pripadajućim algoritmima
- Preuzeti prihvatljive mjerne vrijednosti iz baze podataka
- Provjera je li mjerenje unutar parametara
- Prijava rezultata i osiguravanje potrebne aktivnosti
- Izvedba automatske provjere/kalibracije

Kvaliteta alata za provjeru valjanosti (na temelju ljudskih odluka)

- Procesna okolica s pripadajućim algoritmima
- Izvođenje ljudskih radni prepoznavanja (prepoznavanje prsta ili oka, glasovno upravljanje)
- Prijava rezultata i osigurati potrebne aktivnosti

Zatezanje

- Procesna okolica s pripadajućim algoritmima
- Postavljaju se odgovarajući momenti/brzini/kutovi
- Prate se momentne sile na zakovica
- Prijava momenta na središnjoj bazi podataka
- Izvode se automatske provjere/kalibracije [7].

Testirana je NI SOM kao temeljna platforma za sve ove pametne alate, jer su sveprisutna arhitektura i okvir koji pruža ubrzani proces razvoja od dizajna do prototipa za implementaciju. Prije razvoja na NI SOM, mogao se stvoriti prototip koji se temelji na NI CompactRIO kontroler (cRIO-9068) koji dozvoljava koristiti integrirani IP iz postojeće zrakoplovne biblioteke i algoritme za potvrdu svoje ideje jako brzo. Fleksibilnost korištenja grafičkog i tekstualnog programiranja pored ponovnog korištenja treće strane prenesene na vrhu Xilinx Zynq i NI Linux

Real-time OS nudi savršenu razinu apstrakcije za razvoj tih alata. Sada se može ponovno koristiti kod koji je razvijen na NI SOM kao bolje razvojno rješenje nego da se cijeli proces pokrene ispočetka.

Procijenjeno je nekoliko SOMs i povezanih single-board računala (SBC), i nema usporedbe s platformama temeljenim na dizajnu pristupa i integraciji hardvera ili softvera koje nudi NI. Procjenjuje se da je potrebno vrijeme rješavanja uz NI SOM desetina od vremena koristeći alternativne pristupe zbog povećanja produktivnosti Ni pristupa dizajnu sustava, osobito s NI Linux Real-time i LabVIEW FPGA Module. Uz softver unaprijed osiguran od strane NI SOM, moguće je usredotočiti se više na ključne značajke nekog sustava, kao što su obrada na FPGA.

Budućnost kod zrakoplova donosi stalno dugoročno istraživanje i tehnološki projekt koji je najvažniji za konkurentnosti u proizvodnim procesima. Brzi razvoj je kritičan za implementirajući pristup novim tehnologijama, od početnog isprobanog razvoja ideja do realnog predmeta širenja [7].

7. ZAKLJUČAK

Cilj ovog završnog rada je bio upoznati CompactRIO. Dati njegove tehničke podatke i tehnički opis uređaja, te primjenu njegov rada. Prva i najbitnija stvar je odabrati pravilne i zadovoljavajuće procesore i FPGA. Opisani su načini povezivanja s raznim modulima te načini programiranja uređaja. Navedene su mogućnosti i primjena uređaja u svrhu mjerenja različitih veličina. CompactRIO kontoler može obrađivati razne veličine kao što su: obrada slike, kontrola pokreta, nadgledanje, obrada i analiza.

CompactRIO kontoler pokreće linux, operacijski sistem koji je siguran i pouzdan za korištenje ovog uređaja. Razvoj ovog operacijskog sistema je jako bitan za daljnje usavršavanje samog kontolera tako da na njemu rade razni timovi i stručnjaci koji su povezani na razne načine. CompactRIO platforma je dovoljno razvijena da ne treba dodatne podsustave za projektiranje, tako da se može direktno spojiti s raznim kamerama, motorima, zaslonima.

NI platforma je fleksibilna po pitanju programskih jezika i alata tako da možete koristiti one koji vama odgovaraju. Kao što je već navedeno operacijski sistem se temelji na linux-u tako da se mogu koristiti programi kao što su: LabVIEW, software, C/C++, textual math i dr. Ali za samo programiranje se najčešće koriste LabVIEW i C/C++.

Nove generacije CompactRIO kontolera su izrazito razvijene, tako da imaju ugrađeni GPU i višejezgrene procesore koji znatno ubrzavaju obradu. Za dokazivanje mogućnosti uređaja sastavljaju se testovi koji uspoređivanjem daju određene zaključke. Za kontrolu i nadgledanje testova rade se aplikacije koje prate uspoređivanja.

LITERATURA

[1] National Instruments, Platforma, http://www.ni.com/compactrio/ (pristupljeno 10.5.2016.)

[2] National Instruments, Povezani sustav, http://www.ni.com/white-paper/52251/en/ (pristupljeno 10.5.2016.)

[3] National Instruments, Mogućnosti CompactRIO, http://www.ni.com/white-paper/52987/en/ (pristupljeno 16.5.2016.)

[4] National Instruments, C,C++, http://www.ni.com/white-paper/14623/en/ (pristupljeno 22.5.2016.)

[5] National Instruments, Korištenje CPU, http://www.ni.com/white-paper/52250/en/ (pristupljeno 23.5.2016.)

[6] National Instruments, Uporaba u elektranama, http://sine.ni.com/cs/app/doc/p/id/cs-10016# (pristupljeno 14.6.2016.)

[7] National Instruments, Uporaba u zrakoplovima, http://sine.ni.com/cs/app/doc/p/id/cs-16246 (pristupljeno 14.6.2016.)

[8] VESKI, CoDiS, http://www.veski.hr/hr/products/codis (pristupljeno 16.6.2016.)

[9] Z. Valter: Procesna mjerenja, Elektrotehnički fakultet Osijek, Osijek, 2008.

[10] Thomas Bress, "Effective LabVIEW Programming", NTS Press 2013

SAŽETAK

Prvi dio rada opisuje izgled, upravljanje i rad CompactRIO uređaja. U drugom dijelu rada opisane su mogućnosti uređaja. Također, opisana je obrada podataka i mogućnosti spajanja na uređaj. Treći dio rada opisuje programe u kojima se programira uređaj. Četvrti dio rada uspoređuje podatke dobivene testiranjem opterećenja procesora. Peti dio rada opisuje primjenu CompactRIO uređaja u pogonima.

Ključne riječi: CPU (procesor), FPGA (field-programmable gate array), Linux OS (operacijski sustav)

ABSTRACT

Title: Features and capabilities of CompanctRIO device for measuring and management

First part describes look, management and work of CompactRIO device. In the second part of the paper are described capabilities of the device. Also there is described data processing and capabilities of connecting to the device. Third part of the paper describes programs in which CompactRIO is programmed. Fourth part compares data obtained from processor load tests. Fifth part of the paper describes the use of CompactRIO devices in industrial facilities.

Keywords: CPU (central processing unit), FPGA (field-programmable gate array), Linux OS (operating system)

ŽIVOTOPIS

Dražen Špoljarić rođen je 27. srpnja 1994. godine u Zagrebu. Osnovnu školu završio je u Kutini, a srednju, Tehnička škola Kutina, u Kutini. Godine 2013. upisao je stručni studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku, smjer elektroenergetika.