MERITVE



1. Uvo	od v LabVIEW	2
1.1.	Navidezni instrument NI	2
1.2.	Funkcije v programu LabVIEW	3
1.3.	Programiranje v programu LabVIEW	4
1.4.	Delovanje in uporaba programa LabVIEW	4
1.5.	Spoznavanje LabVIEW aplikacij	5
2. Gra	dnja virtualnih instrumentov	6
2.1.	Virtualni instrumenti	6
2.2.	Kontrolna plošča	7
2.3.	Izgradnja kontrolne plošče (front panel)	7
2.4.	Paleta tools	8
2.5.	Upravljanje z objekti na kontrolni plošči	11
2.6.	Blokovni diagram	12
2.7.	Oblikovanje blokovnega diagrama za merjenje temperature	14
2.8.	Povezovanje objektov	16
2.9.	Shranjevanje VI	19
2.10.	Ustvarjanje podprograma	20
2.11.	Uporaba VI kot podprogram v poljubnem virtualnem instrumentu	21
3. Mer	jenje temperature in prikazovanje merilne vrednosti v diagramu	22
3.1.	Urejanje kontrolne plošče	
3.2.	Urejanje Diagrama	23
4. Upc	praba While zanke	24
4.1.	Kontrola hitrosti delovanja zanke	
4.2.	Zagon virtualnih instrumentov	
4.3.	Odkrivanje napak VI in spremljanje delovanja	
5. Upc	braba FOR zanke	
5.1.	Prikazovanje merilnih vrednosti v tabeli	
6. X-Y	′ graf	32
7. Raz	vejanje programa s Case strukturo	
8. For	mula Node	35
9. Ska	liranje merilnih vrednosti	
10. S	hranjevanje	39
10.1.	Shranjevanje in odpiranje merilnih rezultatov	39
10.2.	Shranjevanje merilnih rezultatov v .txt datoteko:	41
11. P	rikazovanje izmerjenih merilnih vrednosti iz datotek	
12. O	Snovna merjenja z virtualnimi instrumenti	
12.1.	Merjnje napetosti	43
12.2.	Merjenje frekvenčnega spektra in THD	44
12.3.	Zajemanje merilnih vrednosti	45



1. Uvod v LabVIEW

S pojmom merilna instrumentacija ne mislimo samo na merilne instrumente, ki so lahko analogni, digitalni ali z vgrajenimi mikroprocesorskimi enotami. Zaradi kompleksnosti meritev in merilnih postopkov se sedaj v merilno instrumentacijo uvršča tudi programska oprema. V svetu se je zelo uveljavil programski paket LabVIEW. Razvili so ga v podjetju National Instruments. Razvijati so ga začeli leta 1984. LabVIEW temelji na programskem jeziku G. Zaradi grafičnega programiranja ga je bilo težje prenesti na osebni računalnik, vendar se je s popularizacijo grafično delujočih operacijskih sistemov rešila tudi ta težava. Od leta 1992 je na voljo tudi za operacijsko okolje Windows, na katerem je doživel največji razmah.

Pri izbiri merilne instrumentacije imamo dve možnosti. Lahko si izberemo instrument, ki ima že zajeto mikrokrmilniško vodeno procesno enoto za zajemanje in obdelavo podatkov ali pa si takšen instrument sestavimo sami , na lastnem osebnem računalniku. V nobenem primeru pa se ne moremo izogniti prenosu podatkov na osebni računalnik, ki so dan danes nujno potrebni za nadaljnjo obdelavo.

1.1.Navidezni instrument NI

Osnovna programska celica programskega paketa LabVIEW se imenuje navidezni instrument, krajše NI (ang. VI-virtual instrument). Sestavljena je iz sprednje plošče (sl. 1.1) to je uporabniški vmesnik oz. tisti del programa, ki je ob izvajanju viden uporabniku in na katerem lahko izvajamo nastavitve in opazovanje merilnih vrednosti. V blok diagramu so grafično predstavljene vhodne in izhodne spremenljivke. Vhodne spremenljivke ali kontrole (controls) so izvor podatkov za program. Pri izvajanju programa se iz njih preberejo podatki in nato obdelujejo s posameznimi funkcijami. Izhodne spremenljivke ali indikatorji (indicators) so spremenljivke, v katere program izpisuje rezultate svojih funkcij oziroma računanj.



Slika 1.1: Sprednja plošča preprostega navideznega instrumenta.

Blok diagram (sl. 1.2) je okno, v katerem z usmerjanjem toka podatkov med funkcijami izdelujemo program. Praktični izgled blok-diagrama je povezava funkcij oz. blokov v smiselno celoto. Programer ima pri tem olajšano delo saj so povezave različnih tipov oz. različnih barv. Primer: oranžna povezava pomeni tip spremenljivke z decimalno vejico, modra cela števila, vijolična znakovne nize in zelena logične spremenljivke. Pri spremenljivkah je povezava tanka, pri enodimenzionalnem polju ima podvojeno, pri večdimenzionalnem polju pa početverjeno debelino. V blok-diagramu ločimo programske funkcije, ki vhodnim



spremenljivkam priredijo izhodne spremenljivke in programske strukture, ki so namenjene upravljanju in izvajanju programa.



Slika 1.2: Primer diagrama navideznega instrumenta

Tudi v programskem paketu LabVIEW lahko kreiramo lastne funkcije. Funkcije prav tako imenujemo navidezni instrumenti – NI. Kadar želimo funkcijo povezati v nov blok diagram, jo moramo prikazati kot ikono, ki ima svoje konektorje. Ikona (sl. 1.3) je grafičen element, ki je viden na blok diagramu in predstavlja našo funkcijo. Vhodne in izhodne spremenljivke so v sami funkciji povezane na konektor. Celotno priključitev vhodnih in izhodnih spremenljivk na funkcijo lahko prikažemo kot novo ikono. Izdelamo lahko poljubno globoko hirarhično zgradbo programa.



Slika 1.3 : Ikona zgrajenega navideznega instrumenta

1.2.Funkcije v programu LabVIEW

Za obdelavo podatkov in izgradnjo aplikacij v LabVIEW-ju je na voljo množica že pripravljenih funkcij – blokov. Te tvorijo zraven osnovnega paketa vgrajenih funkcij razširjen nabor uporabnih blokov. Razlika med vgrajenimi in razširjenimi funkcijami je, da so razširjene funkcije sestavljene pretežno iz vgrajenih funkcij ali pa se navezujejo na dele programa, napisanega v kakšnem drugem programskem jeziku (Pascal, C,...). Razširjene funkcije so združene v naslednje enote: funkcije, namenjene komunikaciji z instrumenti ali drugimi enotami; funkcije za obdelavo podatkov in funkcije ki so namenjene za delo z multifunkcijskimi vmesniki. LabVIEW direktno omogoča dve vrsti komunikacije z instrumenti in tri vrste komunikacij z drugimi računalniki ali enotami. Z instrumenti komuniciramo preko serijskega vmesnika (RS 232) ali preko vmesnika GPIB. Povezava preko vmesnika GPIB temelji na standardu IEEE 488.1 ali IEEE 488.2. Komunikacija z drugimi računalniki lahko poteka preko protokolov TCP/IP, UDP ali preko mehanizmov NET DDE.

Obdelava podatkov zajema vse običajne načine obdelave podatkov, kot so iskanje srednje vrednosti, računanje Fourierjeve transformacije, računanje različnih spektrov, aproksimacije krivulj in ploskev, generiranje različnih krivulj ipd. Zajem podatkov preko multifunkcijske kartice je zraven komunikacije z instrumenti in obdelave podatkov eden osnovnih razlogov za uporabo LabVIEW-ja. Funkcije, namenjene temu so zelo preproste in fleksibilne.



1.3. Programiranje v programu LabVIEW

Programiranje v programskem paketu LabVIEW je zelo podobno risanju blok diagrama električne sheme. Z definiranjem vhodnih in izhodnih podatkov med sabo povežemo funkcije. Za programiranje so na voljo različna orodja V osnovi jih delimo na orodja za preizkušanje in orodja za urejanje.

Posebnost programskega paketa LabVIEW so orodja za odkrivanje napak v programu. Ker je celoten program grafičen, je tudi orodje za odkrivanje napak grafično. Pri izvajanju programa lahko opazujemo posamezne dele programa in pri tem natančno določimo vzrok za nastalo napako. Vsak podatek je predstavljen kot kroglica, ki potuje po povezavi od ene funkcije do druge. Ob vstopu ali izstopu iz funkcije se poleg mehurčka pojavi še vrednost trenutnega podatka. Omogočeno je tudi koračno izvajanje programa, postavljanje prekinitev in vpisovanje novih vrednosti spremenljivk v realnem času delovanja.

1.4. Delovanje in uporaba programa LabVIEW

LabVIEW ima omogočeno večnitno delovanje in nam omogoča hkratno izvajanje različnih procesov. Teoretično to pomeni, da se določena funkcija izvede takoj, ko dobi vse vhodne parametre oz. podatke. Če funkciji nista povezani, se bosta izvajali paralelno. Praktično se funkciji +, ki dobita vhodne parametre, vpišeta na listo čakajočih. Tista ki ima višjo prioriteto, se bo vpisala višje in tudi prej izvedela. Za uporabnika je to vidno kot več-nitno delovanje.

Z razvojem se programski paket LabVIEW ne uporablja samo v merilni tehniki, ampak tudi v drugih aplikacijah. Najpogostejša laboratorijska uporaba programa LabVIEW, kjer zajemamo podatke iz specialnih instrumentov preko vodila GPIB ali serijskega vmesnika. Pri teh aplikacijah gre večinoma za meritve, ki so pogojene z merilnim instrumentom; LabVIEW pa skupaj z računalnikom uporabljamo za obdelavo in hranjenje podatkov ter za njihov prikaz na zaslon oz. izpis na tiskalnik.

Druga vrsta aplikacij temelji na uporabi multifunkcijskih vmesnikov v osebnih računalnikih. Za pripravo signalov uporabljamo različne sisteme, kot je primer sistem SCXI. Takšne aplikacije zasledimo večinoma v proizvodnih procesih pri testiranju in zagotavljanju kakovosti izdelkov. Multifunkcijski vmesniki imajo večje število kanalov, preko katerih zajemamo podatke na različnih senzorjih in pretvornikih.

Z omrežnimi povezavami osebnih računalnikov shranjujemo podatke v baze podatkov, iz katerih jih lahko ponovno kličemo. Lahko pa jih uporabljamo kot povratno informacijo o procesu. S tem izboljšamo kritične točke procesa (zmanjšamo število izpadlih delov in povečamo splošno kakovost).



1.5. Spoznavanje LabVIEW aplikacij

Program LabVIEW poženemo tako, da poiščemo ikono od programa na osebnem računalniku in 2X kliknemo na ikono. Prikaže se nam naslednje okno (sl. 1.5)



Slika 1.5: Zagon programskega paketa LabVIEW

Gumbe, ki so na osnovnem oknu, uporabljamo za različne namene. Če hočemo narediti nov navidezni instrument, kliknemo na Blank VI. Če kliknemo na gum kjer piše Open lahko odpremo že narejene navidezne instrumente. S klikom na puščico na gumbu Open lahko izbiramo med nazadnje odprtimi navideznimi instrumenti oz lahko odpreme primere (ang. Example). Pod gumbom Help pa se skriva pomoč za delo s programskim jezikom LabVIEW.



Slika 1.6: Okno v katerem lahko izbiramo ali odpiramo različne VI

2. Gradnja virtualnih instrumentov

V naslednjem poglavju spoznamo, kako v LabVIEW aplikaciji oblikujemo navidezen instrument. Natančneje si bomo ogledali gradnjo digitalnega termometra. Poglavje vsebuje razlago izrazoslovja, ki ga uporabljamo pri oblikovanju in zagonu LabVIEW programov. Predstavljena je tudi osnovna tehnika gradnje navideznih instrumentov.

2.1.Virtualni instrumenti

LabVIEW je sestavljen iz kontrolnega okna in blokovnega diagrama. Kontrolno okno je uporabniški vmesnik, blokovni diagram pa izvorna koda virtualnih instrumentov. Uporabimo lahko običajne komponente kot so gumbi, stikala, termometri, grafi, tabele in prikazovalniki za kontrolno okno instrumentov.





tipkovnice.



Slika 2.1: Kontrolno okno

Slika 2.2 : Blok diagram

2.2.Kontrolna plošča

Kontrolna okna uporabljamo kot uporabniški vmesnik pri oskrbi vhodov z podatki iz opazovanih izhodov instrumentalnega sistema. LabVIEW omogoča enostavno načrtovanje, ter sestavo kontrolnega okna. Nudi pa tudi veliko uporabniških kontrol in indikatorjev. Ko je virtualni instrument sestavljen, uporabimo kontrolno ploščo za kontrolo sistema. Spremembe v sistemu lahko vnašamo tudi takrat, ko je virtualen instrument aktiven in sicer s

klikom na stikalo, premikom drsnika, stiskom gumba ali z vpisanim vrednosti preko

2.3.Izgradnja kontrolne plošče (front panel)

Ko pričnemo z izgradnjo kontrolnega okna, najprej odpremo nov prazen navidezni instrument (Blank VI) kot je opisano v prvem poglavju. Odpreta se nam dve okni kontrolno okno in okno blokovnega diagrama.

Tako kontrolno okno kot okno blokovnega diagrama vsebujeta različna orodja za delo z navideznimi elementi. Zraven orodij, ki jih vsebujeta okni imamo še posebne palete. Paleto tools uporabljamo kontrolnem oknu, kot tudi v blokovnem diagramu. Paleto Controls uporabljamo samo v kontrolnem oknu, med tem ko paleto Functions v blokovnem diagramu.



Slika 2.3:Paleta največkrat uporabljenih kontrol



Slika2.4: Paleta vseh kontrol

Tukaj vidimo da imamo v bistvu dve paleti kontrol. To se pojavi z verzijo 7.0 Express v prvi paleti so kontrole ki so največkrat uporabljene kar nam olajša iskanje in nam skrajša čas izdelave čelne plošče. Enako imamo pri paleti funkcij kar se vidi na slikah 2.5 in 2.6



Functions		×
🔍 Search 🛛 👫 Vi	iew •	
💌 Express		
	Luc	Q,
Input	Signal Analysis	Output
∎≓		$ \overset{}{}{}{}{}{}{}{$
Sig Manip	Exec Control	Arith & Compar
Favorites		
Iser Libraries		
Select a VI		
	*	

🔍 Search 🛛 👫 View 🔻 Programming Measurement I/O Instrument I/O Vision and Motion Mathematics Signal Processing Data Communication Connectivity Control Design & Simulation SignalExpress Express L. **Ş**' **`**@ Input Signal Analysis Output ⊳Σ' ∫ ⊳ . ₽ Sig Manip Exec Control Arith & Conpar Addons Favorites User Libraries Select a VI... *

Slika 2.5: Paleta največkrat uporabljenih funkcij

Slika 2.6: Paleta vseh funkcij



2.4.Paleta tools

Paleta z orodji se uporablja v kontrolni okni in v blok diagramu. V oknu z orodji vidimo na vrhu zeleno lučko pred katero je narisano orodje. To je funkcija ki nam omogoča da nam program sam izbira primerno orodje in ne rabimo mi izbirati katero orodje bomo uporabljali. Orodje nam izbira glede na to kje se nahajamo z miško. Če to možnost izključimo si moramo orodje izbirat sami. To naredimo tako da z miško kliknemo na želeno orodje in sem nam miškin kazalec spremeni v simbol na katerega smo kliknili lahko pa s tipko TAB na hitro



menjavamo med orodji katero orodje trenutno uporabljamo vidimo ko pogledamo simbol miškinega kazalca.

Ustvarimo in popravimo lahko virtualne instrumente samo takrat kadar trenutno okno ni v stanju delovanja (EDIT mode). V aktivnem (RUN mode) stanju pa lahko z virtualnimi instrumenti merimo in prikazujemo merjene vrednosti . Za preklop pritisnemo miško na gumb Run ali Run continuosly (za Run mode) in Abort execution za edit mode.

EDIT mode



RUN mode





Na začetku dela, ko imamo aktivno kontrolno ploščo, je okno v urejevalnem ali EDIT mode načinu. Kurzor deluje kot delovno orodje. Uporabimo ga za nastavljanje kontrol, številskih vrednosti na prikazovalnikih, premikanju segmentov na kontrolni plošči in blokovnem diagramu.

Najprej sestavimo kontrolno ploščo s termometrom, ki prikazuje temperaturo.

MERITVE





Slika 2.6: Menu Contols

Nad prikazovalnikom termometra se pokaže majhen pravokotnik. Vanj vpišemo s tipkovnico TEMPERATURA in kliknemo z miško na prazno polje. Digitalni prikaz termometra dobimo tako da z desnim miškinim gumbom pritisnemo na termometer in kliknemo na Visible items odpre se nam podmeni kjer izberem digital display.



- Orodje za urejanje teksta

Na začetku je razpon skale na termometru od 0 do 100°C. Vrednost skale spremenimo tako da z orodjem za urejanje teksta dvakrat kliknemo na vrhu skale kjer piše 100 in vtipkamo želeno





MERITVE

maksimalno vrednost skale. S spremembo zgornje vrednosti se nam skala avtomatsko porazdeli. Spreminjamo lahko vse vrednosti ki so na skali napisane. To storimo tako da z orodjem za urejanje teksta dvakrat kliknemo na bilo katero številko in vpišemo novo vrednost. Skala se nam potem sama postavi in porazdeli glede na vrednost ki smo jo vtipkali. Spodaj je nekaj primerov različnih skal.







Slika 2.7 : Sprememba skale na termometru

2.5. Upravljanje z objekti na kontrolni plošči



Če želimo urediti ali spremeniti velikost ikon na kontrolni plošči ali blokovnem diagramu uporabimo orodje za pozicioniranje Positioning tool (sl.2.8)







Slika 2.8: Urejanje elementov

Izberemo orodje Positining tool in kliknemo na termometer. S klikom izberemo objekt, kateremu spreminjamo lastnosti. Objekt v tem primeru termometer, obdaja premikajoča prekinjena obrobna linija imenovana marquee. Objekt je tako izbran. Sedaj ga lahko brišemo, režemo, kopiramo ali premikamo. Če želimo izbrati nov objekt ali trenutnega narediti neaktivnega, kliknemo na nov element, oziroma v prazen prostor na delovni površini.

Z orodjem za pozicioniranje lahko objekte tudi povečujemo ali zmanjšujemo. Dinamični objekti (objekti ki jim lahko spreminjamo velikost) imajo ob kotih tipične kotnike (resizing handles), ki pa se pojavijo šele, ko na njih nastavimo kurzor. Takrat pritisnemo in držimo levi gumb, da s premikom miške poljubno nastavimo njihovo velikost.



Kurzor orodja nastavimo na levi zgornji kot termometra, dokler ne spremeni oblike. Takrat kliknemo in držimo levi gumb ter premaknemo polje termometra navzgor. Velikost termometra se spremni (sl. 2.9)



Slika 2.9: Sprememba velikosti termometra

2.6.Blokovni diagram

Blokovni diagram je izvorna koda virtualnega instrumenta. Gradnja poteka brez podrobnega tekstovnega programiranja. Ko končamo z vnosom v kontrolno okno, se nam v blokovnem diagramu pojavijo pripadajoči bloki (sl. 2.10)

Funkcijske bloke povežemo z linijami, ki

določajo in prenašajo izmenjavo podatkov. Bloki vsebujejo enostavne aritmetične operacije, mrežo in podatkovne I/O funkcije, katere hranijo ali posredujejo informacije v ASCII, binarnem ali celo spreadsheet formatu.



Slika 2.10 : Kontrolno okno in blok diagram

Za prikaz temperaturne vrednosti vzamemo kot vhodno veličino napetost. Odvisna je od temperature in jo v procesu pretvorimo v stopinje z ustrezno metodo, glede na vrsto senzorja. Tako lahko programiramo VI, da operacijo opravi sam, z izgradnjo merilnega diagrama. Kot primer predpostavimo, da je senzor linearen in je enostaven prikazovalni faktor zadosten.

Izberemo opcijo prikaži diagram Show Diagram iz okna Windows menu. Blokovni diagram je sedaj aktivno okno, ki že vsebuje terminal za temperaturno indikacijo in spremenljivko, ki smo mu jo vpisali.



- Blokovni diagram - končni indikator

Ikona ki je prikazana pod spremenljivko (napisom) Temperatura je terminal in ustreza indikatorju na kontrolni plošči. Z dodajanjem kontrol ali indikatorjev na kontrolno ploščo, se pojavi njihov terminal tudi na blokovnem diagramu. Diagram lahko vsebuje tudi ikone, ki predstavljajo funkcije.

Za ureditev terminalov in ostalih ikon na blokovnem diagramu uporabimo orodje za pozicioniranje. S tem , ko smo objekt izbrali, ga lahko tudi brišemo, izrežemo, kopiramo ali premikamo. V blokovnem diagramu ne moremo brisati terminalov, ki pripadajo kontrolam sprednje plošče. Za odstranitev takšnega terminala, moramo opraviti postopek brisanja na kontrolni plošči .

Izberemo orodje za pozicioniranje. Uporabimo ga za premik temperaturnega terminala na desno stran blokovnega diagrama. S klikom na levi gumb miške terminal primemo in prenesemo na ustrezno mesto, kot je prikazano na sliki (sl. 2.11). S premikom terminala smo avtomatsko premaknili tudi oznako terminala. Če bi na isti način premaknili samo oznako, bi terminal ostal na prejšnjem mestu.



Slika 2.11 : Prenos elementa

2.7. Oblikovanje blokovnega diagrama za merjenje temperature

Za delovanje blokovnega diagrama potrebujemo algoritem, ki simulira delovanje VI. Iz njega lahko dobimo napetost, katera ustreza merjeni temperaturi.



- express VI ki nam simulira različne vrste napetosti

V delovni površini blokovnega diagrama kliknemo na miškino desno tipko in se nam na zaslonu odpre paleta s funkcijami. Izberemo iz menija »Analysis« »simulate sig« odpre se nam okno express VI-ja v katerem si nastavimo napetost, obliko napetosti izberemo lahko funkcijo da nam doda šum v signal kot motnjo. Signal lahko simuliramo tudi na drug način na primer z funkcijo »random number« ,ki jo najdemo v paleti »Numeric« (sl. 2.12) .

6

- funkcija »random number«



Slika 2.12 : Paleta numeric

Če uporabimo »random number« funkcijo moramo še uporabit funkcijo za množenje«. To pa zaradi tega ker nam »random number« funkcija na izhodu da številke med 0 in 1. Funkcijo za množenje najdemo v isti paleti kot funkcijo »random number« (sl. 2.12)

- funkcija množenja

100 - Numerična konstanta

Numerično konstanto bomo uporabili za množenje številk ki jih dobimo iz funkcije »random number« tako da bomo dobili normalne vrednosti ki bojo med 0 in 100. Te vrednosti bojo nam predstavljale temperaturo. Numerično konstanto najdemo v isti paleti (numeric) kot ostali dve funkciji. Z orodjem za urejanje teksta lahko spreminjamo vrednosti konstante tako da 2X kliknemo v polje ker je napisana številka in vpišemo želeno vrednost.

Z orodjem za pozicioniranje nato postavimo objekte blokovnega diagrama v neko smiselno postavitev da se jo da lepo in razločno prebrat kaj pomeni. Postavimo jo na primer tako kot je na sliki (sl. 2.13).



Slika 2.13 : Razporeditev elementov (na vrhu z express VI-jem od spodaj s funkcijo »random number«)

2.8.Povezovanje objektov

Vsi indikatorji in kontrole na kontrolni plošči imajo v blokovnem diagrami svoj terminal. Za prenos podatkov iz terminala do ikone, ki predstavlja neko funkcijo, moramo povezati terminal in ikono s povezavami. Črte nam predstavljajo povezave, ki prenašajo podatke. Vsako vrsto podatka določa različen tip povezav. Primer: tanka cela povezava predstavlja numerično skalarne podatke (Numeric Scalar Data), debela povezava pa predstavlja numerično podatkovno polje (Numeric Array Data). Prav tako pri prepoznavi vrste podatka pomagajo tudi barve povezav. Modra povezava prenaša podatke tipa intiger (celoštevilčno), oranžna povezava prenaša števila s plavajočo vejico (Floating Point Numbers), zelena žica prenaša podatke tipa boolean (True – False) in rožnata žica prenaša podatke tipa string (besedila), modro bela žica prenaša dinamični tip podatkov. Tipe povezav imamo prikazane tudi na naslednji sliki (sl. 2.14)





 (\mathbb{N})



Za informacije o povezavah blokovnega diagrama lahko aktiviramo okno pomoč (Help) in izberemo možnost (show Context Help). Pomoč nam pokaže vhodne in izhodne funkcije, ko na želeno funkcijo postavimo kurzor. Če premikamo kurzor preko ikone VI, se terminali za povezave osvetljijo. Pri povezavah osvetljeni terminali pomagajo pri pravilni priključitvi vhodov in izhodov.



Za povezovanje blokov v blokovnem diagramu izberemo orodje za povezovanje.

Najprej bomo povezali express VI z termometrom. Postavimo kurzor na desno stran express VI-ja kjer piše Sine ali DC offset, odvisno kako smo nastavili simulacijo. Ko se nam terminal osvetli kliknemo z levim gumbom na miški. Ko gumb spustimo potegnemo z miško v desno in se pojavi črta ki je na eni strani pritrjena na blok. To črto nato potegnemo do termometra. Ko se nam vhod osvetli zopet kliknemo z levim miškinim gumbom in spustimo. Povezava se nam obarva glede na tip podatka ki ga bomo prenašali. Za povezovanje s funkcijo »random number« je postopek enak s tem da povežemo vse funkcije skupaj in nato iz funkcije za množenje potegnemo še do termometra kot je prikazano na sliki 2.15



Slika 2.15: Prikaz povezav med terminali

Če se nam pri povezovanju prikaže črtkana sivo obarvana črta pomeni da povezava ni dobra oz. da sta povezana napačna tipa podatkov. V tem primeru moramo pogledat kaj je narobe. Napako lahko odkrijemo tako da v primeru ko se nam pojavi črtkana črta in ne vemo kaj je narobe kliknemo v skrajnem zgornjem levem kotu zlomljeno puščico ki nam kaže da je nekaj narobe. Ko kliknemo na puščico se nam odpre okno kjer so napisane napake v povezavah. Tako najlažje odkrijemo kaj je narobe. Primer slabe povezave nam prikazuje naslednja slika (sl. 2.16).

 (\mathbb{N})





- znak da je napaka v blok diagramu

VI List Untitled 1		
2 errors and warnings ● Block Diagram Errors Multiply: contains unwired or bad terminal This wire connects more than one data source.	Show Warnings	Thermometer
Details One or more required inputs to this function are not wired or are wired incorr Context Help window to see what the connections to this function should be Close Show Error	rectly. Show the A	

Slika 2.16: Primer slabe povezave in prikaz napak

Slabe povezave najlažje odstranimo s kombinacijo tipk »Ctrl+B« s tem da se v tem primeru zbrišejo čisto vse slabe povezave v vezju. Lahko pa brišemo posamezne povezave tako da kliknemo na povezavo 2X da se nam izbere celotna povezava oz samo 1X da izberemo samo del povezave in pritisnemo tipko »Delete«.

Ko popravimo vse povezave se nam namesto zlomljene puščice pojavi cela puščica. Tako vemo da je v vezju vse pravilno in naj bi program deloval. To izvemo samo na en način tako da ga poženemo. S tipkami »Ctrl+E« na hitro preklopimo med oknoma in se nam pojavi čelna plošča. Nato kliknemo s puščico in na termometru se nam prikaže izmerjena vrednost. Če pritisnemo tipko za »continuosuly run« se nam bo vrednost na termometru spreminjala.

Za izdelan termometer napravimo še preklopnik za izbiro temperaturne skale in matematično pretvorbo za merjenje temperature v °C ali v °F.

Kontrolno ploščo dopolnimo s stikalom, blok diagram pa z matematičnimi operatorji odštevanje, deljenje in izbiro enote T/F (True/False). Matematične operatorje najdemo v paleti »Numeric« enoto za T/F pa najdemo v paleti Boolean.



2.9.Shranjevanje VI

Izberemo opcijo »Save As« iz menuja »File«. Predhodno pa kreiramo novo knjižnico New VI Library s svojim imenom, to shranimo v mapo, v katero želimo.

V polje za ime VI-ja vtipkamo npr. Digitalni Termometer.vi« in kliknemo OK kot je prikazano na sliki (sl. 2.17). Naslov aktivnega okna se spremeni v ime ki smo ga pravkar podali.



Slika 2.17 : Shranjevanje navideznega instrumenta

(JS

2.10. Ustvarjanje podprograma

Da lahko odpremo digitalni termometer VI iz blokovnega diagrama, ali ga uporabljamo v kateremkoli drugem programu Lab VIEWa moramo narediti ikono in konektor. Ikona je grafična podoba, konektor pa določa kontrole in indikacije VI, ter vhodne in izhodne terminale.

Kontrolno okno se mora nahajati v edit načinu če želimo izdelati ikono. Ko imamo kontrolno okno v edit načinu in imamo na kontrolni okni vse kontrole in indikacije ki bi se naj nahajale v tej ikoni kliknemo z miško v zgornjem desnem kotu na oštevilčeno ikono in izberemo »Edit Icon« (sl. 2.19)





Nato se nam pojavi okno editorja kjer lahko sami poljubno oblikujemo ikono ki želimo da se nam kasneje pojavlja v vezju (sl. 2.20)





Ko končamo z risanjem ikone kliknemo na gumb OK ikonskega editorja. Novo kreirana ikona nadomesti staro.

Podatke pošiljamo in sprejemamo od odvisnega VI skozi terminale in konektorja. Konektorje definiramo tako, da si najprej izberemo terminal konektorja, ki ga želimo za VI in nato dodelimo kontrolno ploščo ali indikator vsakemu terminalu posebej. Samo kontrole in indikatorji, ki jih bomo uporabljali pri programiranju potrebujejo terminal na konektorskem delu.



Kliknemo na ikono ki smo jo kreirali, kliknemo desni gumb da se nam pokažejo možnosti in kliknemo »Show conectors«.

LabVIEW nam ponudi osnovni terminal ki se ujema s številom vhodov in izhodov glede na to koliko imamo kontrol in indikatorjev na čelni plošči. Da določimo konektorje kontrolam (te so na levi polovici kvadratka) in indikatorjem (te so na desni polovici kvadratka) uporabimo orodje za povezovanje. Najprej kliknemo na konektor na ikoni nato pa kliknemo na kontrolo oz. indikator ki želimo da je povezan s konektorjem. Terminal na ikoni se nam obarva in tako vemo da smo določili kontroli oz. indikatorju konektor. Ko določimo vse konektorje kliknemo z desno miško na ikono, klilnemo Show Icon in natoVI shranimo.

2.11. Uporaba VI kot podprogram v poljubnem virtualnem instrumentu

Odpremo New VI iz palete BlankVI. Če želimo prejšnji VI, ki ima dodeljeno ikono in terminal uporabiti v drugem VI-ju ga izberemo tako da v spodnjem levem kotu funkcijske palete kliknemo na »select VI« (sl. 2.21)



Slika 2.21 : Select VI

Kliknemo na ikono in izberemo VI, ki smo ga prej shranili, sedaj pa ga lahko uporabimo kot podprogram v kakšnem drugem VI-ju.



3. Merjenje temperature in prikazovanje merilne vrednosti v diagramu

3.1. Urejanje kontrolne plošče

Nenehno želimo opazovati temperaturo nekega prostora oziroma naprave. Aplikacijo lahko sestavimo tako, da VI odčita temperaturo enkrat na sekundo in prikaže temperaturo na diagramu. Najprej odpremo nov VI kot je opisano na začetku gradiva.

VI potrebuje preklopnik na kontrolni plošči za začetek in konec merjenja, ter oblikovan diagram za prikaz temperature.

Izberemo vertikalni preklopnik iz palete »Bolean« kontrolnega menija. Spremenljivki stikala vpišemo ime npr. »Stikalo«. Izberemo »Chart« diagram iz palete »Graph« kontrolnega menija. Nameščen je v zgornjem levem kotu graf palete. V polje za spremenljivko vpišemo ime npr. »Diagram trenutne temperature«.

Z diagramom se pojavita dva dodatka. Paleta orodja za spremembo diagrama in legenda. S pomočjo legende lahko definiramo barvo in slog za izris diagrama. Z uporabo opcij na paleti lahko spreminjamo format diagrama in skalo osi. Prikažemo ali skrijemo lahko opcije z uporabo »Show« podmenija v roletnem meniju diagrama (sl. 3.1)



Slika 3.1 : Kontrolno okno



Spremenimo območje da dobimo temperaturne vrednosti. Z orodjem za označevanje dvokliknemo na najnižjo vrednost in vtipkamo 70. Enako spremenimo zgornjo mejo na 90. Ko pritisnemo Enter se skala avtomatsko porazdeli.

Če želimo spremeniti barvo grafa (v primeru, da nismo v črnobelem načinu) izberemo orodje za barvanje in kliknemo z desnim gumbom na graf. Pojavi se sledeča paleta barv. Če želimo operacijo preklicati, postavimo kurzor izven palete, preden spustimo desni gumb na miški.



Orodje za barvanje in izbiro barv



3.2.Urejanje Diagrama

Iz Windows menija izberemo »Show Diagram«. Na nezasedenem področju diagrama kliknemo z desnim gumbom ter iz menija »Functions« izberite VI. Pojavi se okno, kot ga prikazuje naslednja slika (sl. 3.2). Poiščemo in odpremo knjižnico kamor smo prej shranili prejšnji VI. Dvokliknemo na shranjen VI ali pa ga označimo in potem kliknemo na gumb OK.



Slika 3.2 : Odpremo VI instrument



Postavimo in povežemo ikono digitalnega termometra tako kot je prikazano na sliki (sl. 3.3)



Slika 3.3 : Povezava grafa z ikono termometra

Prekinjena povezava med digitalnim termometrom in grafom lahko pomeni da sta terminala in indikator na nadzorni plošči nepravilno povezana. Zato izberemo »Show Context Help« iz »Help« menija. Ko kliknemo na ikono digitalnega termometra se pokažejo povezave. Če ikona nima izhoda za temperaturo, si preglejmo navodila v drugem podpoglavju kako narediti konektor.

4. Uporaba While zanke

LabVIEW ponuja strukture programiranja kot je ponavljajoča ali pogojna zanka in razne zaporedne ponavljajoče in razvejane operacije. Za ponavljanje meritve je potrebno objekte postaviti znotraj zanke »While Loop«.

While Loop je časovna-pogojna zanka. Poddiagram (diagram znotraj roba) se izvaja tako dolgo, dokler pogoj ni izpolnjen. Zanka »While Loop« ima pogojni terminal, na katerega priključite logično vrednost in ponavljajoči terminal, ki hrani vrednost trenutnega ponavljanja. While Loop najdemo v funkcijski paleti pod structures (sl. 3.4)





Slika 3.4 : While Loop Pojavi se nam spodnja ikona.



- numerični izhod terminala nam pove koliko krat se je zanka izvršila. Štet začne z 0

- pogojni terminal definira, kdaj se zanka konča. Sprejema logične vrednosti T/F. Z njo vklopimo in izklopimo meritev.

Poglejmo si delovanje pogojne zanke. Zanko bomo potegnili čez prej povezan graf s termometrom. To bomo storili tako da bomo izbrali zanko ko se nam pojavi črtkana črta potegnemo kvadrat čez vse bloke ki želimo da so zajeti v tej zanki v našem primeru so to vsi bloki. Ko imamo zajete vse bloke v zanki spustimo gumb na miški in se nam izriše ikona zanke. (sl. 3.5)



Slika 3.5: While Loop



Tako lahko določimo velikost zanke in jo postavimo kamorkoli na diagram. Če želimo kasneje spremeniti velikost zanke enostavno postavimo kurzor na kateri koli kot zanke in počakamo, da se pojavi orodje za spreminjanje velikosti zanke. Nato kliknemo in raztegnemo zanko do želene velikosti.

Povežemo terminal stikala za vklop s pogojnim terminalom zanke. Črtkana povezava med njima predstavlja binarno informacijo (sl. 3.6)



Slika 3.6: Povezava gumba true/false

4.1.Kontrola hitrosti delovanja zanke

Iz palete »Time & Dialog« v functions meniju izberemo funkcijo »Wait Until Next ms Multiple«(sl. 3.7). Izberemo funkcijo »Numeric constant« iz palete »Structs & Constants« v funkcijskem meniju, ter ji spremenimo vrednost na 1000. Funkcija »Wait Until Next ms Multiple« zagotovi, da se zanka izvrši vsako sekundo (vsakih 1000ms).



Slika 3.7: Paleta Time & Dialog



Povežemo blok diagram, kot je prikazano na naslednji sliki (sl. 3.8)





V funkcijski paleti Express VI lahko izberemo ikono Time Delay, ki prav tako kot zgornja napravi zakasnitev v delovanju zanke.

-😭 Functions			🔍 Search
	Executio	n Control	
		Q .	
Input	Analysis	Output	User Libraries
	▷ Σ ¹	₽ ₽	
	-🔁 Execution (Iontrol	
Exec Ctrl		Time Delay	
			84101978 7 8 8 8 9 9 9
	While Loop	Flat Sequence	Case Structure
	Time Delay	Elapsed Time	

4.2.Zagon virtualnih instrumentov



Vrnimo se na nadzorno ploščo in sicer tako, da iz Windows menija izberemo »Show Panel« ali s pritiskom na tipki »Ctrl+E«. Z ikono delovnega orodja kliknimo na stikalo za vklop tako, da bo v zgornjem položaju vključeno). Za zagon virtualnega instrumenta kliknemo na gumb zagon.



ڬ - gumb za zagon

Na grafu se pokažejo vrednosti temperature. Vi izbere in prikaže novo vrednost vsakič, ko se zanka ponovi. Prikaze na grafu lahko poljubno izbiramo.

4.3.Odkrivanje napak VI in spremljanje delovanja

Napake pri konstrukciji VI se nam pojavijo zaradi logičnih napak ali napak sintakse. Ko poženemo VI in če se nam v Run funkciji prikaže zlomljena puščica, potem se nam izpiše lista napak. Če izberemo Find, nam najde objekt in terminal, kjer je napaka nastala, hkrati pa napiše vrsto napake.



- zlomljena puščica

Zelo uporabna metoda odkrivanja napak je tudi pregled pretoka informacij z uporabo osvetlitve poteka. Vrnimo se v blok diagram tako, da v windows meniju izberemo show diagram. Poženemo VI in kliknemo na gum za osvetljavo. Gumb se spremeni v svetlečo žarnico.



Izvršitev osvetlitve poteka

Če VI ni vključen, kliknemo na gumb za zagon. Osvetljava poteka simulira potek informacij v blok diagramu (sl. 3.9). Potujoče točke predstavljajo pretok informacij skozi VI. Pozorno je treba spremljati tudi vrednost, ki se prikaže ob točki. Vrednost predstavlja informacijo, ki se prenaša po žici.



Slika 3.9: Prikaz osvetljave poteka

Prav tako je potrebno opazovati zaporedje v katerem se funkcije izvajajo. V navadnih tekstovnih jezikih se program izvršuje po zaporedju, v katerem je napisan. V LabVIEW paketu, pa se funkcija izvrši šele takrat, ko ima na razpolago vse potrebne vhodne informacije, kar pomeni, da se program ne izvaja nujno od zgoraj navzdol ali od leve proti desni.





postopen način v neprekinjenem načinu delovanja

Dober način odpravljanja napak je izvajanje VI po korakih

Medtem, ko VI deluje kliknemo na gumb za koračni način. Desno je nov gumb, s katerim lahko preskočimo izvajanje podprogramov.



preskok podprogramov

Če opazujemo delovanje podprograma, ga zapustimo s pritiskom na naslednji gumb.



- zapustimo program

Osvetljava poteka in izvajanje po korakih sta pomoč pri odpravljanju nastalih napak. Zelo cenjeni pa sta pri razvoju VI.

Še zadnjič kliknemo na gumb za zagon, da poženemo VI pri največji hitrosti.

S pomočjo delovnega orodja izklopimo stikalo. S tem povzročimo zaustavitev zanke in VI se ustavi.

VI si shranimo pod nekim imenom v svojo mapo.

5. Uporaba FOR zanke

Kadar uporabljamo FOR zanko moramo uporabljati za prikazovanje merilnih vrednosti Graf.







FOR zanko najdemo jo v funkcijski paleti, all functions, structures:



Izbereš zanko ki se bo ponavljala in jo razširiš čez terminale.Diagram znotraj zanke se izvede tolikokrat, kolikor ponovitev določimo (števec, ki je konstanta).





5.1.Prikazovanje merilnih vrednosti v tabeli

FOR in tabela.vi Front Panel *					
<u>File Edit Operate Tools Browse Window H</u> e	lp				
🖒 🐼 🛑 🔢 13pt Application Font	- .	 ■< Ø 			
	Table				
	Št. meritve	Temperatura(C)			
	0.00000	0 597408	Wisible Items	•	/ Label
	1.000000	0.357730	Find Terminal		Cantion
	2,000000	0.215494	Change to Control		Index Display
	3,000000	0,561463			✓ Vertical Scrollbar
	4,000000	0,531448	Description and Tip		✓ Horizontal Scrollbar
					Row Headers
			Create	2	✓ Column Headers
			Data Operations	1	
			- Advanced	5	
	_			<u> </u>	
			✓ Selection Scrolling		
			Editable Headers		
			Moveable Column Separate	ors	
			Autosize Row Height		
			Multi-line Input		
			Sinooch Scrolling		
			Properties		







6. X-Y graf

To obliko grafa uporabljamo pri merjenju prenosnih karakteristik vezij. Uporabljamo jo podobno, kot bi na osciloskopu izključili časovno bazo s tipko X-Y.



(JS



Diagram:



7. Razvejanje programa s Case strukturo

Case strukturo uporabljamo tam, kjer želimo da se diagram izvaja po določenih stopnjah oziroma korakih. Case strukturo najdemo v funkcijski paleti, pod Execution control:



Za primer napravimo enostavni kalkulator, ki bo vrednosti sešteval, odšteval,množil in delil, vsako operacijo pa bo opravljal v svoji Case structuri.

MERITVE



Diagram kalkulatorja:





8. Formula Node

V čelni plošči oblikujemo kontrolo za izhodno napetost ojačevalca in vhodno napetost. Oblikujemo tudi indikator za velikost ojačanja.

Þ	Formula dB.vi Front Panel *										
Eile	Edit Operate Tools Browse Window Help										
	수 🐼 🛑 💵 13pt Application Font 🖃 🏣 🖬 🕮 🖬 🏧										
	Text Settings										
	Uout										
	÷) o										
	Ojačanje v dB										
	Uin										

V diagramu uporabimo Express VI in funkcijske palete podprogram formula:





Zapišemo enačbo:

🔀 Configure Formula [Formula]								
20*log(X1/	x2)							
Input	Label	Home	Ва	ckspace	Clear		End	
×1	X1	е	**	log	In	mod	min	
X2	X2	Pi	sqrt	log2	exp	rem	max	
X3	X3							
X4	X4	7	8	9		sin	int	
X5	_ ^5	4	5	6		tan	sign	
X6	X6	1	2	3			sign	
X7	X7	0		E	+			
X8	X8	More	e Functions		-	~		
				ОК	Cance		Help	

Oblikujemo diagram in preizkusimo enačbo s konkretnimi podatki .





9. Skaliranje merilnih vrednosti

Meriti želimo temperaturo od 0° do 100 °C. Merilni pretvornik nam daje tokovni izhod 4-20mA. Tok vodimo skozi 250 Ω upor, zato na njem nastane padec napetosti od 1-5V. To napetost skaliramo v območje temperature. Naredimo čelno ploščo VI:

Ð Skaliranje 4-20mA.vi Front Panel * File Edit Operate Tools Browse Window Help <u>*l+</u>-13pt Application Font 4 ±....▼ || •0....▼ **C** - C \checkmark П Ŧ Tokovni vhod Skalirana vrednost 0,02-40 60 80 20 0,015-100 n 0,01 0,004= STOP

V diagramu v funkcijski paleti imamo podprogram za skaliranje vrednosti:



MERITVE



Define Sigr	nat		
— Data Points	5		Defined Signal
X	Y		100-
	0 100		90- 80- 70- 60- 50- 40- 30- 20- 10- 0- 1, 1,5, 2, 2,5, 3, 3,5, 4, 4,5, 5 X Axis
Rescale New minimum 1 New maximun 5	X New minimum 0 n X New maximum 100	Y n Y	Show interpolated values
Load Dat	a Save Dal	:a	OK Cancel Help

Dvokliknemo na Scale&Map in konstruiramo tabelo:

Izdelamo program v diagramu VI:





10. Shranjevanje

10.1. Shranjevanje in odpiranje merilnih rezultatov

Merilne rezultate lahko shranjujemo v različne datoteke. Najbolj pogosto uporabljamo tekstovne datoteke (.txt) in posebne LabVIEW datoteke (.lvm).

V funkcijski paleti pod Output izberemo Write LVM:

+ Functions		🔍 Sear	:h			
Express			•			
		Ö ,	M. Outruit			
Inout	Signal Analysis	Output	¶⊒ Odipol	Write To Mea	acurement File	
∎≓∎		≥Σ ∫ ⊳				
Sig Manip	Exec Control	Arith & Compar	DAQ Assist	Instr Assist	Instr Drivers	
Favorites User Libraries			113 LEC + + 113 LEC	.		
Select a VI			Build Text	Display Msg	Play Waveform	
	*					
			Write Meas File	Report	DIAdem Report	
				\$~	Յ մաթե	Ę
				NI-FGEN (Std)	NI-FGEN (Arb)	NI-DCPower
			A'	() Մախի		
			NI-RFSG (CW)	NI-RFSG (Arb)		

Z dvoklikom izberemo, kam bomo shranili podatke:

ijoze\LabVIEW Data\test.lvm					
Action	Segment Headers				
Save to one file	 One header per segment 				
Ask user to choose file	One header only				
 Ask only once 	◯ No headers				
O Ask each iteration	- X Value Columns				
If a file already exists	⊙ One column per channel				
Ouse next available file name	One column only				
O Append to file	O Empty time column				
Overwrite file	Delimiter				
	⊙ Tab				
Save to series of files (multiple files)	◯ Comma				
File Description					

V diagramu napravimo program za shranjevanje rezultatov v datoteko in prikazovanje v tabelo





10.2. Shranjevanje merilnih rezultatov v .txt datoteko:

Poglejmo si primer, kako shranimo merilne vrednosti naključnega signala v .txt datoteko in kako iz te datoteke rezultate preberemo.

Primer shranjevanja v .txt datoteko nam kaže spodnji program. Naključne merilne vrednosti prikazujemo v grafu.





11. Prikazovanje izmerjenih merilnih vrednosti iz datotek

Če želimo prikazati merilne rezultate iz shranjenih .lvm datotek lahko napravimo sledeči VI. Vrednosti beremo iz prejšne shranjene datoteke, prikazujemo pa vrednosti v tabeli.

Þ	Read iz file v tabelo .vi Block Diagram *	
Eile	e Edit Operate Tools Browse Window Help	
	🖒 🕑 🔲 🧣 🛵 🗗 🔐 Arial Unicode MS 🔽 🚛 🖬 🥵	
	Read LabVIEW Build Table Signals Table	
	Read LabVIEW Measurement File Signals Table	

Če želimo prikazati merilne rezultate iz shranjenih .txt datotek lahko napravimo sledeči VI. Merilne vrednosti bomo prikazali v grafu.



(JS



12. Osnovna merjenja z virtualnimi instrumenti

12.1. Merjenje napetosti

Izdelajmo simulacijski VI, ki bo nam meril srednjo vrednost napetosti in RMS vrednost. Uporabimo Express VI:

Ð	Basic Amplitu	ıde Measurements.vi	Fro	nt Panel							
Eile	<u>E</u> dit <u>O</u> perate	<u>T</u> ools <u>B</u> rowse <u>W</u> indow	<u>H</u> elp								
	수 & 🔵 🛙	13pt Application Font			· 🖬 📲	≞⊸∣<	5-1				
	STOP										
	Amplitude	Frequency In	put					Sine 📈		Mean	
	10-	100-	e	,0-						Mean	
	9-	90-	2	.0-Â	Λ	Λ	ΛΛ	\wedge /		0.05	
	8-	80-				μ.				10,05	
	7-	70-	e '	,0-							
	6	60	jiji (,0-	ļ		\ } }	\		PMS	
	5-	50 -	ا ج	,0-	1	11	11			KMD	
	4	40 -			- \		- 17			3.81	
	3-	30 -	-4	,0- () V	V	V	V	VV		5,01	
	2-	20-	-6	,0-							
	1 -	10-		0,000	0,020	0,040 T	0,060 me	0,080 0,100)		
	0-	0-									

S kontrolami amplitude in frekvence spreminjamo vhodni signal.

Diagram VI:

V rumenih poljih je napisano navodilo za preizkus simulacije merjenja amplitude.



 (\mathbf{M})



12.2. Merjenje frekvenčnega spektra in THD

V čelni plošči izdelajmo nastavitev frekvence, prikazovanje oscilograma vhodne napetosti, frekvenčnega spektra in THD merilnika.



Diagram izdelan z Express VI:



 (\mathbb{M})



12.3. Zajemanje merilnih vrednosti

Merilne vrednosti zajamemo s pomočjo DAQ asistenta, ki je lociran v funkcijski paletti v ikoni Input. Dvokliknemo na ikono DAQ Assist.



(JS



Odpre sa nam okno NI-DAQ, ki nam omogoča definirati, katere signale bomo zajemali. Kliknemo na želeni vhod.



Kliknemo na veličino, ki jo želimo meriti.





Definiramo strojno opremo, s katero bomo zajemali merilne vrednosti. Ta mora biti predhodno nameščena in konfigurirana v računalniku. Namestitev in konfiguracijo opravimo s programom NI-MAX, ki nam omogoča enostavno nameščanje strojne opreme.



 (\mathbf{M})

Edit Operate Iools Browse Window He 수 준 (1) 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이	p Dialog Font		b-
Create New NI-DAQ [™]			
Select the channel(s) to add to your task. You also can add or copy existing global channels to your task. <i>Global channels</i> are channels created from MAX or your application software that are saved in MAX and can be used in any task or application. You can only add global channels that support the measurement. You also can add or copy physical channels associated with transducer electronic data sheet (TEDS) sensors to your task. A <i>TEDS</i> is a data sheet for an analog sensor. A TEDS contains the critical information needed by a device or measurement system to identify, characterize, interface, and properly use signals from an		Physical Supported Physical Channels Dev1 (PCI-6024E) ai0 ai1 ai2 ai3 ai4 ai5 ai6 ai7 ai8 ai9 ai10 ai11 ai12	

V naslednjem oknu nastavimo konfiguracijo terminala (RSE, NRSE, Differential),nastavimo Acquation mode, Sample To Read in testiramo kanal.

Eile Edit Operate Iools Browse Window Help	
수 🕸 🔘 💵 😵 👦 📅 🗊 13pt Dialog Font 🔽 🏪 🖬 🐨	9
DAQ Assistant	
Voltage Ipput Satur	🔄 Back 🔡 🖄
Volage in par secup	Moacuring
Volkage E Settings	Voltage
Lipput Ranne	Most measurement
DAQ Assistant	devices are designed
Max 5 Volts V	reading, voltage. Two
Min -5 Volts V	measurements are DC
	and AC.
	DC voltages are useful for measuring
Terminal Configuration	phenomena that
Differential 💌	time, such as
Custom Scaling	or strain.
No Scale> V	AC voltages, on the
	other hand, are waveforms that
	constantly increase,
Task Timing Task Triggering	polarity. Most
Acquisition Mode	voltage.
1 Sample (On Demand) Clock Settings	
1 Sample (HW Timed) Samples To Read 100	🖻 赵
	Torminal Configuration
Continuous Rate (Hz) 1000	specifies the grounding
Advanced Clock Settings	mode used for the channel:
Clock Type Active Edge Clock Source	Differential
Internal 💟 Rising 💟	Depending on your
	specific hardware, the positive and
	negative inputs for the channel are
imir Express Task 🌽 Connection Diagram	either upreferenced 🔍

(JS

Na DAQ izberemo kanal, na katerem želimo meriti.



Kliknemo Finish in konfiguracija zajemanja merilne vrednosti je pripravljena za prikazovanje.

נא 🛃	Jntit	ed 1 i	Block	c Dia	igram	*		
Eile	Edit	<u>O</u> pera	te <u>T</u>	ools	Brows	e <u>v</u>	Vindow	F
	₽	ন্দ্র 🤇		9	40	đ	<u>1</u>	.3pl
			۶ ۲	Ĺ	X	•		
			D	AQ A	ssistar	ht		
				da	ata 🛛	÷		

Če želimo popraviti nastavitve potem dvokliknemo na ikono in postopek ponovimo.

Ostala merjenja z VI bomo izvajali kot vaje, ki bodo objavljene na spletni strani: ftp://www.scv.si/vss/joze_lukanc/

