

1. Osnovi rada u LabVIEW

1.1. Uvod

Glavni cilj laboratorijskih vježbi iz područja virtualne instrumentacije je upoznavanje s načinom korištenja kompjutera za prikupljanje mjernih podataka i nakim zamkama i greškama koje se mogu dogoditi tijekom tog postupka. Dodatno je važno da učenik izvrši određene praktične radnje u prikupljanju podataka te da pokuša samostalno izraditi određene aplikacije koje su mu od interesa.

Okosnica praktičnog rada na laboratorijskim vježbama je korištenje paketa za grafičko programiranje virtualnih instrumenata **LabVIEW**. Taj je programski paket posebno izrađen tako da dozvoljava brzu ugradnju kompjuterskog prikupljana podataka i sustava za njihovu analizu koji mogu biti široko prilagođeni potrebama korisnika paketa. Taj se programski paket odlikuje velikim mogućnostima ali se vjerojatno znatno razlikuje od bilo kojeg programskog paketa s kojim se neki programer mogao sresti u svom radu do sada. Zbog toga se od učenika očekuje sustavno učenje programiranja u LabVIEWu prije nego što se počne profitirati u radu s njim. Nažalost je neka drugi način rada nemoguć. Da bi sam početak savladavanja programskog paketa LabVIEW bio što lakše i brže postoji nekoliko ograničenja i pretpostavki koji se postavljaju pred učenike:

- poznavanje koncepta kompjuterskog sustava kao što su **file**, pohranjivanje **fileova**, **direktiriji**, ispis itd.
- poznavanje **GUTa** (Graphical User Interface) te korištenje miša i tipkovnice
- naglasak je dat na tehnike prikupljanja podataka a ne na " lijepom " programiranju u LabVIEWu (važno je dobra obaviti posao prikupljanja podataka a ne elegantnost postupka)

1.2. Pogled na LabVIEWu

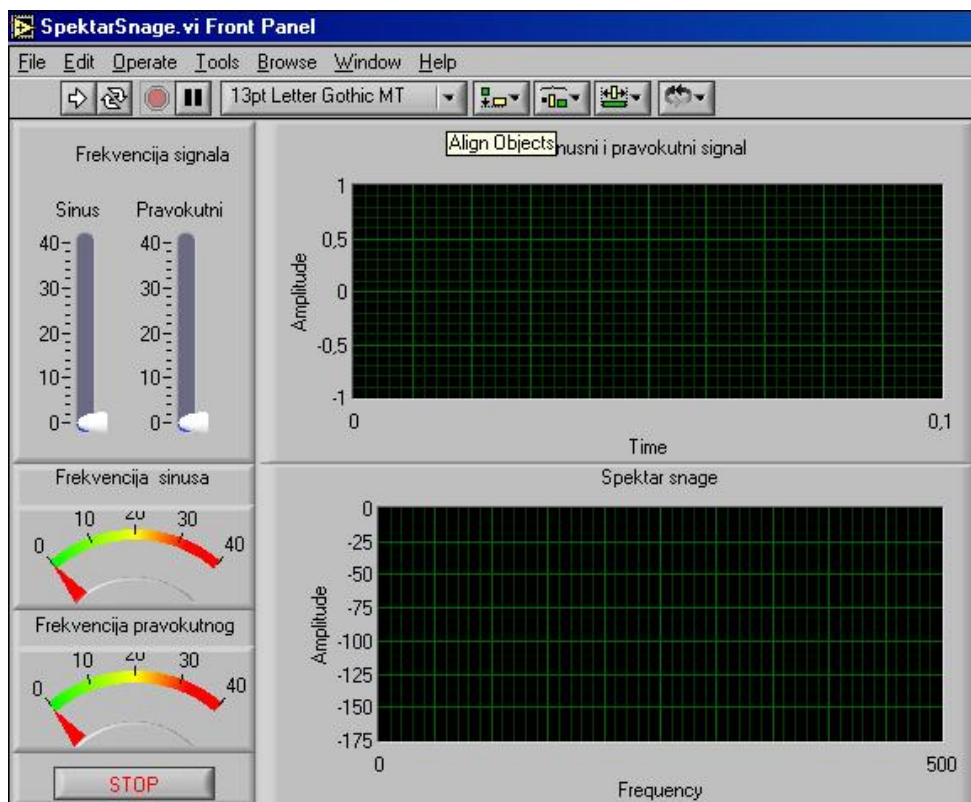
LabVIEW je u potpunosti grafički program koji izgleda poput crteža elektroničke sheme s jedne strane i stila električkih mjernih uređaja iz pedesetih godina dvadesetog stoljeća. To čini koncept **blok dijagrama (blok diagram)** i **prednje ploče (front panel)**. LabVIEW je hijerarhijski program u tom smislu da se bilo koji **virtualni instrument** (bilo koja potpuna funkcionalna cjelina naziva se **virtualni instrument** te se gotovo uvijek označava kao **VI**) koji se programira može brzo pretvoriti u modul koji može biti potprogram drugog **VI**. To je u potpunosti analogno konceptu programiranja kod konvencionalnih programskih jezika.

LabVIEW je dizajniran osim toga tako da se može proširivati. Mogu se dodavati moduli pomoću različitih sredstava. Proizvođač mjerne kartice ili nekog mjernog instrumenta može pogramirati LabVIEW driver koji se ponaša kao **VI** koji predatavlja mjernu karticu i njenu funkcionalnost u LabVIEW okruženju. Korisnik može također sam napisati LabVIEW modul i koristiti ga kao **VI** u drugim programima ili može napisati modul koji se sučeljavaju s LabVIEWom u drugim jezicima kao što su C i C++ . Oni su poznatui kap **podVI (subVI)** i ne razlikuju se od **VI** osim što je sučelje određeno za slijedeću razinu. PodVI u C ili C++ su vrlo korisni ukoliko je potrebno izvesti vrlo složeni numerički postupak na podacima koji nisu pokriveni u standardnoj LabVIEW rutini.

1.3. Osnovni koncepti LabVIEWa

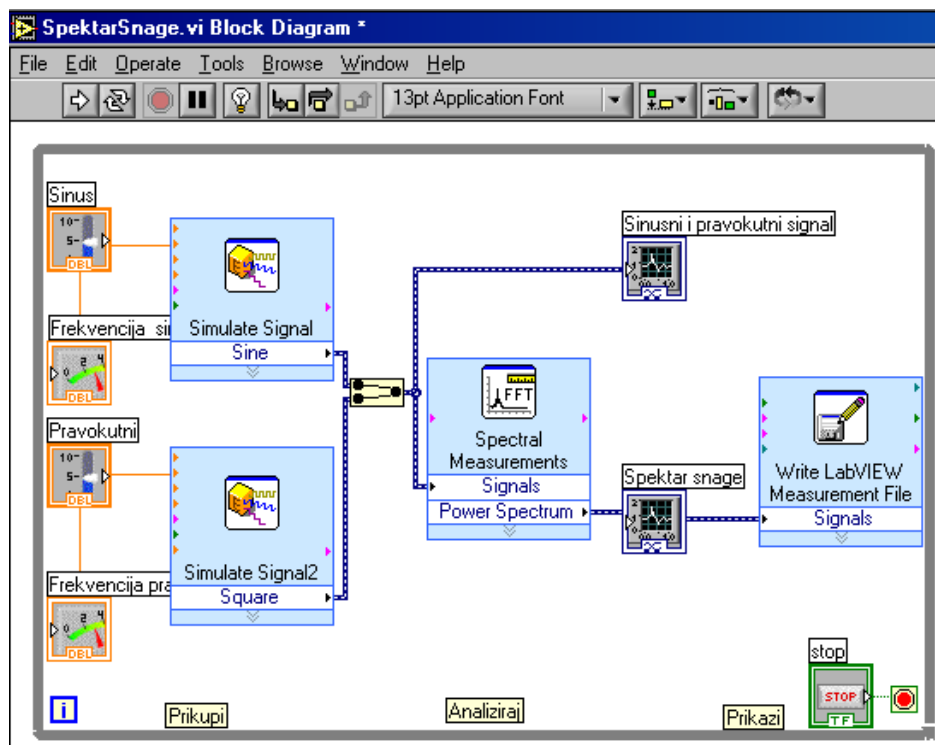
Kao što je spomenuto ranije postoje dva izgleda bilo kojeg LabVIEW virtualnog instrumenta. To su **blok dijagram (blok diagram)** i **prednja ploča (front panel)**.

Prednja ploča (sl 1-1.) je ono što korisnik vidi pri radu s mjernim sistemom. Ona sadrži **kontrolne** i **indikatore**. LabVIEW ima vrlo bogat izbornik jednih i drugih (čak se može dizajnirati i vlastita kontrola ili indikator) i to omogućava široko područje mogućnosti programeru.



Slika 1-1

Kontrolne mogu imati mnoštvo oblika. Mnogi od tih oblika su virtualne «slike» realnih kontrola koje se koriste kod stvarnih instrumenata kao rotirajući gumb (**knob**) na primjer. Ostali imaju digitalni koncept. Sve kontrolne imaju neku vrstu vizualne informacije na osnovu koje korisnik uvijek zna u kojem se stanje nalaze. To jako pomaže jer se ne mora pri programiranju eksplicitno programirati i stanje pojedine kontrolne. Drugo vrlo korisno svojstvo kontrola je to da se može odrediti njihova reakcija ukoliko je postavljena ulazna vrijednost nestabilana. U specifičnom slučaju, ukoliko ulazna kontrola može poprimiti samo cjelobrojne vrijednosti u opsegu od 0 do 10 može se specificirati što se treba dogoditi ukoliko se unese ulazna vrijednost 3,5 ili -1 ili «nula» (zero) kao tekstualna naredba. Programski riješiti tu vrstu problema zahtijeva previše programerskog vremena tako da se ovom opcijom štedi na vremenu.



Slika 1-2.

Indikatori imaju mnoštvo različitih oblika. Neki od njih su preslik stvarnih indikatora (instrumenata i sl.) a neki su dizajnirani prikladnije za zaslon računala. Koncept indikatora uključuje grafove (graphs) i prikaze (charts) što je druga važna ušteda u vremenu jer se ti pokazivački elementi ne moraju dizajnirati zasebno.

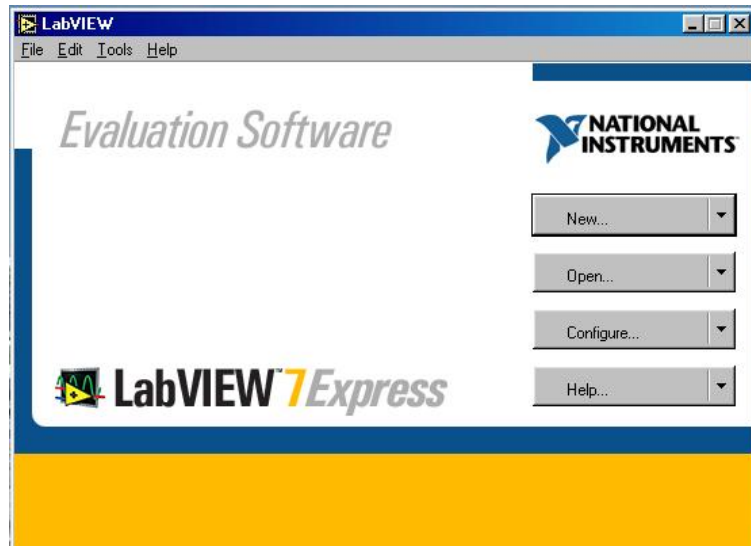
Blok dijagram (sl. 2) predstavlja pozadinu njegove prednje ploče koji prikazuje kako su sve kontrole i indikatori povezani međusobno kao i skrivene module gdje se odvija sva programska obrada podataka. Izgleda na neki način poput elektroničkog shematskog dijagrama i barem je konceptualno ožičen na isti način. Takav prikaz često izgleda vrlo složen kao u elektroničkoj shemi stvarnog elektroničkog mjernog instrumenta. Stoga je jedno od najvažnijih svojstava u LabVIEW programiranju odrediti timing i redoslijed operacija. U konvencionalnom programskom jeziku to je ostvareno redoslijedom naredbi i korištenjem različitih programskih petlji (FOR, WHILE itd.). LabVIEW radi na potpuno isti način jedino što je način na koji se određuje redoslijed izvršenja naredbi profinjeniji. Koncept korišten u LabVIEWu je dataflow – radnja se na nekom elementu izvršava kad su svi njegovi ulazi na raspolaganju. To znači paralelnost u izvršenju radnje (ili barem pseudoparalelizam). Uobičajeno izvršavanje radnji je s lijeva na desno jer su ulazi obično na lijevom dijelu elementa a izlazi na desnom dijelu. Međutim to nije zahtjev već samo stvar dogovora. Programske petlje i redoslijed kojim se odvijaju naredbe odvija se pomoću struktura koje izgledaju poput knjiga s brojevima stranica ili kao odsječci filmske trake.

1.4. Sučeljem LabVIEWa

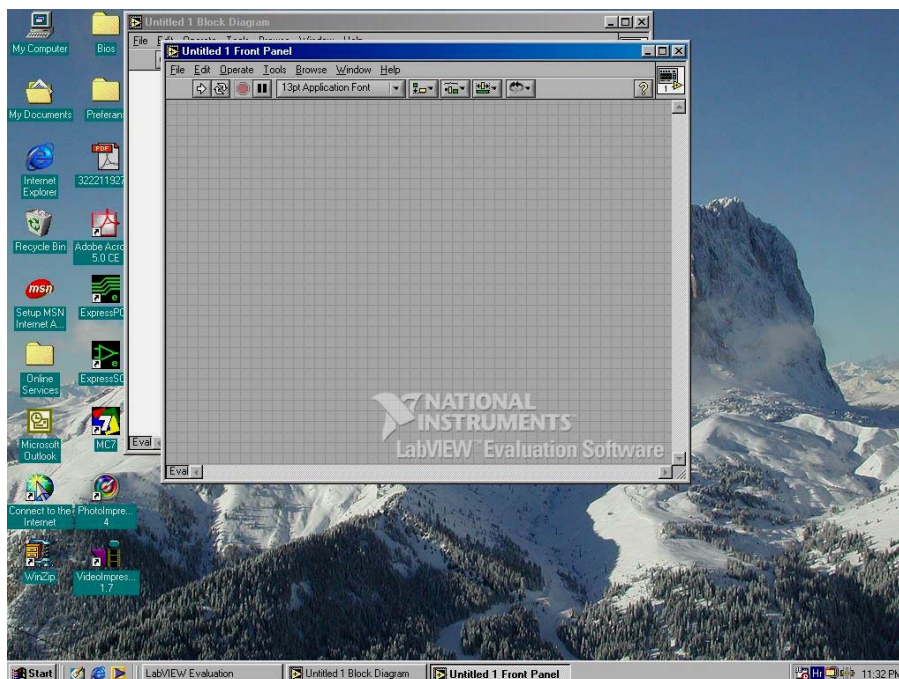
Slijedi vrlo kratki pregled sučelja LabVIEWa. Neće se prijeći svim mogućim mogućnostima već samo najčešćim. Dok se čita ovaj dio potrebno je sjesti uz računalo i praktički usvojiti i razumjeti postupak rada sa sučeljem LabVIEWa.

LabVIEW se poziva sa **Start>> Programs>>National Instruments>> LabVIEW** (ili ukoliko je ikona LabVIEWa na zaslonu kompjutera može se pokrenuti LabVIEW dvostrukim škljocanjem na tu ikonu). Nakon otvaranja programa zaslon kompjutera izgleda poput sl. 1-3

Škljocnite na padajući izbornik uz **New...** i izaberite **Blank VI...** Pojavit će se prazna prednja ploča virtualnog instrumenta (siva površina) i u pozadini prozor blok dijagrama (bijela površina)





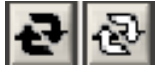


Slika 1-3.








Slika 1-4.

Obratite pažnju najprije na prozor sa sivom pozadinom – prednju ploču. Prozor je prazan jer još nije dizajnirana nikakva prednja ploča a pri vrhu se nalazi sedam ikona pri vrhu prozora i jedna velika ikona u gornjem desnom kutu. Deset ikona pri vrhu služe za dizajniranje prednje ploče dok velika ikona u gornjem desnom kutu služi za predstavljanje čitave prednje ploče kad izrađujemo ili mijenjamo podVI.

Devet je ikona podijeljeno u dvije skupine. Prve četiri služe za pokretanje VI

Ikona	Značenje	Objašnjenje
	RUN	Jednom pokreće VI. VI se neće izvršiti ukoliko im se ne da takova naredba. Gumb RUN mijenja izgled kad je VI stvarno pokrenut.
		VI se ne može pokrenuti jer e programu postoje greške Škljocom na PREKINUTI RUN pojavit će se popis grešaka s objašnjenjima
	RUN NEPREKIDNO	VI se izvršava stalno. Taj način izvršenja VI nije preporučljiv osim pri vrlo jednostavnim VI i ukoliko se ne vrši provjera VI.
	STOP	Zaustavlja VI koji je pokrenut
	PAUZA/ NASTAVI	Jednim škljocom VI prelazi u stanje pauze dok ponovni škljoc nastavlja izvršenje VI

Posljednjih pet služe za izgradnju prednje ploče

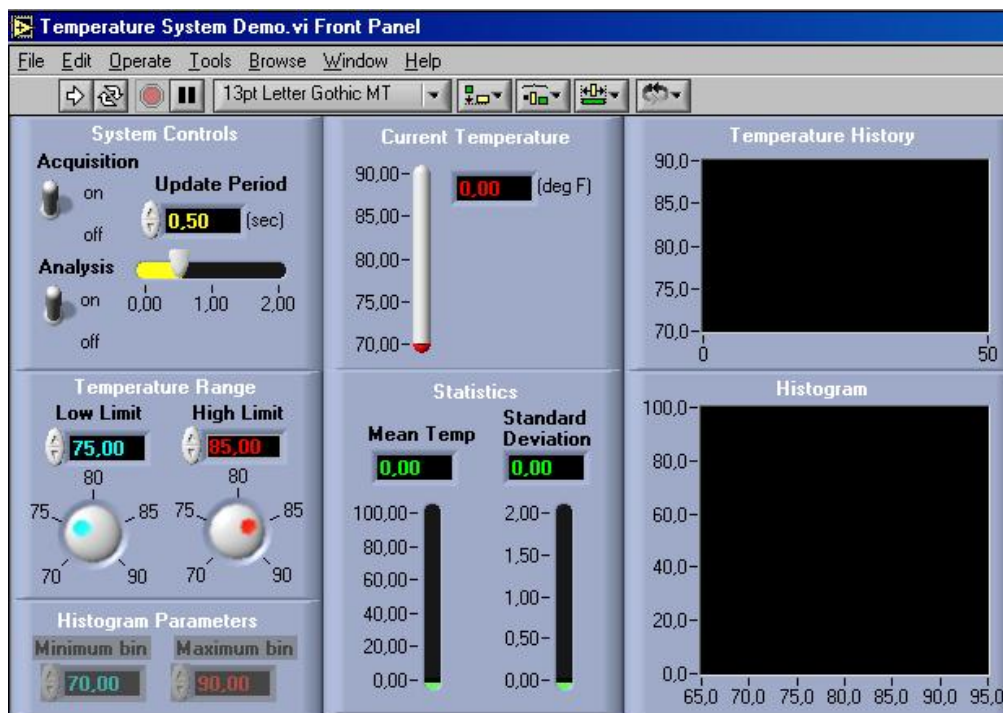
Ikona	Značenje	Objašnjenje
	FONTS	Odabir slova (veličina i oblik) koja se koriste za prednju ploču
	PORAVNJANJE	Poravnava objekte i ikone prema vodećoj ikoni ili objektu
	RAZDIOBA	Razdjeljuje objekte i ikone
	PROMIJENI VELIČINU	Mijenja veličini objekta ili ikone
	PROMIJENI REDOSLIJED	Mijenja redoslijed objekata ili ikona (ispred/iza , gore/dolje)

Iznad linije ikona nalazi se sedam menija označenih tekstem

FILE	To je izbornik koji omogućuje operacije s <i>fileom</i> kao što su otvaranje, zatvaranje, pohranjivanje, ispis. VI je <i>file</i>
EDIT	Sadrži kontrole za uređivanje elemenata
OPERATE	Sadrži operacije koje uključuju pokretanje VIa
TOOLS	Sadrži pojedinosti koje omogućuju konfiguriranje LabVIEWa, vaših projekata i vaših VIa
BROWSE	Sadrži pojedinosti koje omogućuju uvid u aspekte trenutnog VIa i njegovu hijerarhiju
WINDOWS	Odabire se koji je prozor vidljiv
HELP	Jedno od najvažnijih područja programa. Postoji nekoliko načina ostvarenja pomoćci za svaku pojedinost ili koncept i o tom će se dosta govoriti.

Cilj Potrebno Učitati i pokrenuti VI. Program je standardni primjer programa LabVIEW i simulira sustav za mjerenje temperature. Cilj je vježbe pokazati nekoliko stvari koje čine LabVIEW i usvojiti jednostavne operacije u LabVIEWu.

Odaberi **Help>>Find Examples...** zatim odaberite **Search** i upišite *temp*. Pojavit će se dvije opcije *template* i *temperature* – te dvaput škljocni na drugu opciju. U tablici će se pojaviti šest primjera koji zadovoljavaju vaš kriterij pretraživanja. Škljocnite dva puta na **Temperature System Demo.vi**.



Trebali bi dobiti ekran izgleda prikazanog na slici 1-5.

Slika 1-5.

U gornjem lijevom kutu prozora nalazi se skupina kontroli. Dva klizna preklopnika koja imaju dva stanja **on** i **off** – predstavljaju booleove kontrole. Tu je i slijedeća kontrola koja ima digitalni i analogni dio. Digitalni dio je ploča s brojem u njoj i dvije strelice za povećavanje i smanjivanje vrijednosti. Digitalne ulazne vrijednosti mogu se obično promijeniti škljocom na strelicu ili upisujući vrijednost preko već upisane vrijednosti u displej. Analogni dio je klizač koji se može vući mišem mijenjajući pri tom vrijednost.

Ispod te ploče u sredini lijevo je slijedeći par kontroli koji ima sličan digitalni dio sami što su kontrole rotirajuće a ne klizne. Među njima nema razlike samo što je teže mišem rotirati gumb nego pomicati klizač. Međutim kako su kod nevirtualnih instrumenata puno češće rotirajuće kontrole od kliznih zgodno je na taj način stvoriti veću iluziju fizičkog instrumenta.

Na sredini zaslona su dvije ploče s pokazivačima na njima. Oba pokazivača imaju analogni i digitalni dio. Digitalni dio pokazivača izgleda gotovo isto kao i digitalna kontrola samo bez strelica kojima se mijenja vrijednost. Analogni pokazivač ima izgled termometra.

U krajnje desnom dijelu nalaze se neki od najmoćnijih pokazivača u LabVIEWu – grafovi i prikazi (charts). To su grafički podaci izlaza i mogu se prikazati na mnoštvo različitih načina.

Na prednjoj se ploči nalazi još nekoliko stvari. Dolje lijevo su dvije kontrole koje su “zasivljene“ što znači da im se momentalno ne može pristupiti i promijeniti vrijednost.

Prije no što se pokrene VI potrebno je objasniti što je njegova zadaća. VI simulira očitavanje temperature brzinom koja je određena s “ update rate “ te izračunava srednju vrijednost i standardno odstupanje zadnjih 10 očitavanja. Prikazuje se povijest očitanih vrijednosti temperature s trenutnim vrijednostima gornje i donje granice te histogram očitanih vrijednosti. Konačno, VI aktivira signalizatore prijelaza gornje i donje granične vrijednosti.


Prije nego što se pokrene VI treba pokazati kako se printaju prikazi sa zaslona. Postoje dva postupka ovisno o ome što se želi printati

Printanje sadržaja VI

1. Odberi **File>>Print**
2. Pojavit će se izbornik s pitanjem što se želi printati:

Printanje prikaza cijelog ekrana

1. Maksimiziraj VI pritiskom gumba za maksimiziranje ukoliko je neophodno
2. Pritisni gumb **PrintScreen**
3. Otvori program za crtanje **Start>>Programs>>Accessories>>Paint**
4. Umetni sliku **Edit>>Paste**
5. Printaj sliku **File>>Print**

Pritisnite Ikonu RUN gore lijevo . Uoči da je ikona promijenila oblik u i da se na prednjoj ploči prikazuju vrijednosti temperature (u ovom primjeru simulirane) . Ovaj VI se neće nikad zaustaviti jer sadrži beskonačnu petlju kao što će se vidjeti kasnije.

Koristeći radne alate mogu se kontrole prednje ploče pomicati, zakretati, stiskati i pisati preko njih. Da bi se zakrenuo gumb ili sklopka postavite radne alate na kontrolu i koristite lijevu tipku miša. Da bi se promijenila vrijednost na određenoj digitalnoj kontroli postavite radni alat na odgovarajuću strjelicu i škljocnite (iznos koji kontrola mijenja se može

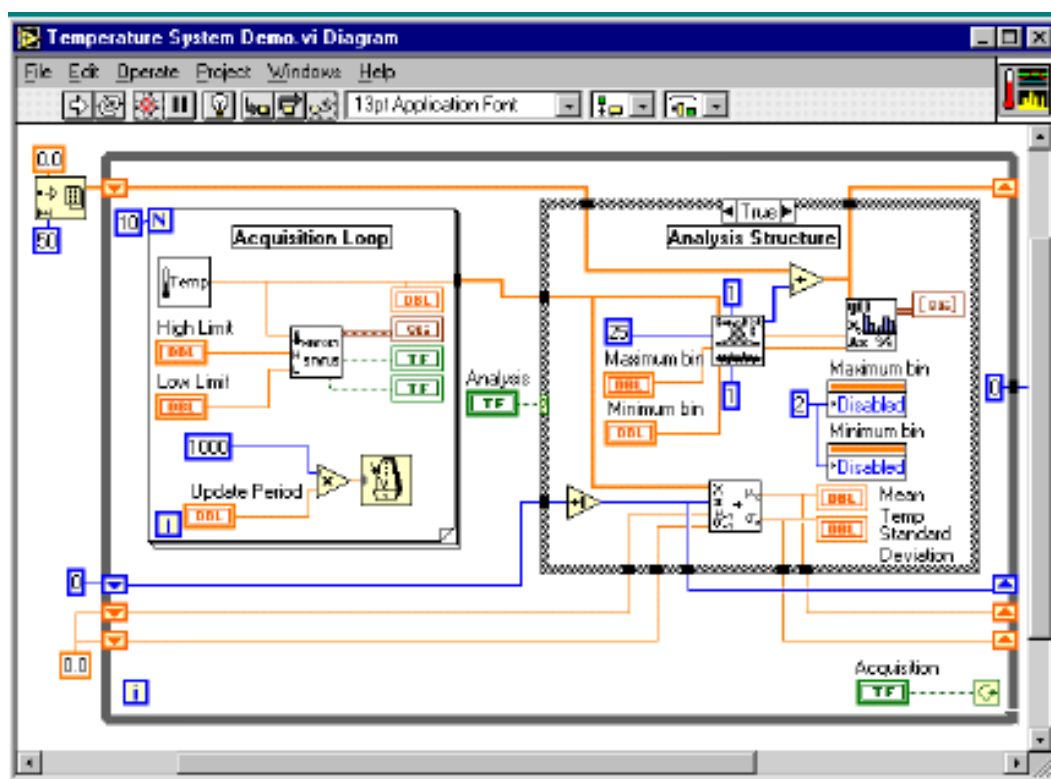
promijeniti kao što ćemo kasnije vidjeti). Da bi se napisala vrijednost preko postojeće postavite alat za pisanje preko stare vrijednosti i upišite novu.

Promijenite nekoliko kontrola i uočite efekte. Primijetite da kod analognih kontrola postoje gornja i donja granica vrijednosti dok kod digitalnih kontrola tih granica nema.

Nakon što završite s radom zaustavite VI prebacivanjem sklopke za akviziciju u položaj «off». VI se možda neće odmah zaustaviti jer treba završiti ono što je obavljao. To je mnogo bolji način zaustavljanja VI nego pritiskom na gumb **STOP** na alatnoj traci jer dozvoljava čist završetak operacija VI. Gumb **STOP** ostavlja operacije nezavršenom.





Sad ćemo pogledati u zaleđe *prednje ploče* – *blok dijagram*. Odaberi **Windows>>Show Block Diagram**. Pri tom uočite da blok dijagram nije zamijenio prednju ploču već su sad oba vidljiva u isto vrijeme što je korisno pri razvijanju operacija.

Prvo što primjećujete na blok dijagramu je to da je on složeniji od prednje ploče. Uostalom većina opreme ljepše izgleda izvana nego iznutra.



Slika 1-6.

Osim toga uočit ćete da su se na alatnoj traci blok dijagrama pojavila slijedeća četiri alata kojih na alatnoj traci prednje ploče nije bilo:

Ikona	Značenje	Objašnjenje
	Gumba za animirano izvršenje VI	Omogućava animirano promatranje izvršavanja VIa. Prikazuje se tijek podataka blok dijagramom od jednog čvora do drugog
	Gumb za ulazak u VI	Ulazak u Izvršenje VI korak po korak. Vrlo zgodno kad se želi vidjeti u kojem je čvoru VI došlo do greške.
	Gumb za prelazak preko VIa	Prelazak preko VI (podVI) pri izvršenju VI korak po korak.
	Gumb za izlazak iz VIa	Izlazak iz VI pri izvršenju VI korak po korak.

Slijedeće su tri osjenčana okvira – veliki vanjski i dva unutar njega. To su okviri petlje. Vanjski okvir je WHILE petlja koja se izvršava dok booleov uvjet ne postane «false». Unutrašnja petlja na lijevoj strani je FOR petlja koja se izvršava jednom za svaku iteraciju WHILE petlje (te su petlje su dakle ugniježdene ako što dijagram sugerira) dok pravokutnik na desnoj strani nije petlja već CASE čvor koji obavlja uvjetivanu operaciju «ako-onda» («if-else»). Ono što je prikazano momentalno na blok dijagramu predstavlja «true» slučaj. Škljocnite na strelicu uz riječ «true» pri vrhu tog pravokutnika u sredini i tako da se prikaže i «false» slučaj. Još jednim škljocom se vraća stanje na prikazano.

Primjećuje se da su «žice» koje povezuju dijelove i neke pravokutnike blok dijagrama različitih boja. LabVIEW naime koristi različite obojane «žice» da bi označio različite tipove varijabli. Zelena (točkasta) predstavlja booleovu varijablu, plava je cjelobrojna varijabla a narančasta je varijabla dvostruke preciznosti. Uočite da se narančaste žice javljaju u dvije debljine – tanja linija predstavlja skalare a deblja polja (matrice).

Budući se LabVIEW izvršava s lijeva na desno jasno je da se unutar svake iteracije vanjske WHILE petlje lijeva FOR petlja odvije kompletno nakon čega desni čvor if-else izvršava.

Veći broj malih pravokutnika oivičenih jednom obojenom linijom s brojevima unutar nje predstavljaju konstante. Pravokutnici oivičeni dvostrukom linijom (vanjska deblja – unutarnja tanja) predstavljaju kontrole – čija je pozadina odgovarajuća kontrola na prednjoj ploči. Dvaput škljocnite na komandu označenu «update period». Prikaz se prebacuje na prednju ploču i možete uočiti da je kontrola «update period» sad označena na prednjoj ploči. To su prednja i stražnja strana te kontrole. Vratite se na blok dijagram.

Pravokutnici oivičeni dvostrukom tankom linijom su indikatori. Probajte sličnu akciju na onom označenom s «DBL» gore desno u FOR petlji. To odgovara termometru i njenoj digitalnoj vrijednosti u trenutnom temperaturnom području.

Ono što je vrlo važno na blok dijagramu su kvadratični blokovi uglavnom u crno-bijelom prikazu. Oni su također VI. Pokušaj dvostruko škljocnuti na jednog od njih (onog u sredini FOR petlje). Pojavit će se nova prednja ploča. Kontrole prednje ploče kontroliraju se pomoću ulaznih žica u VI a indikatori dobivaju svoj izlaz kroz spojene žice u ostale smjerove. Uočite da postoje tri kontrole i tri indikatora. Zatvorite taj prozor i uočite da u taj

VI postoje tri ulaza (s lijeve strane) i tri izlaza (s desne strane). To je potprogram u LabVIEWu obično zvan podVI (subVI). Svaki podVI može biti izveden neovisno i povezan s ostalim podVi tako tvori složeniju cjelnu.

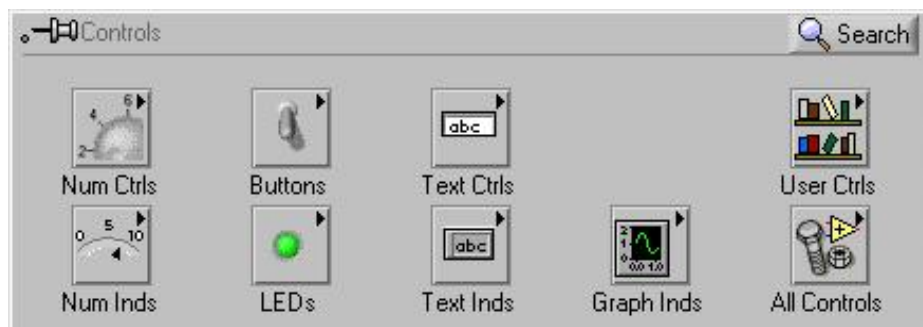
Sad zatvori prednju ploču (time će se automatski zatvoriti i blok dijagram). Ukoliko su nastale neke promjene u VI ignorirajte ih (**Save Changes>>No**).

Slijedi izgradnja vrlo jednostavnog VI. Da bi bili sigurni da ćemo početi iz iste točke zatvorite LabVIEW zatvaranjem praznog prozora prednje ploče.

2. Jednostavni primjer

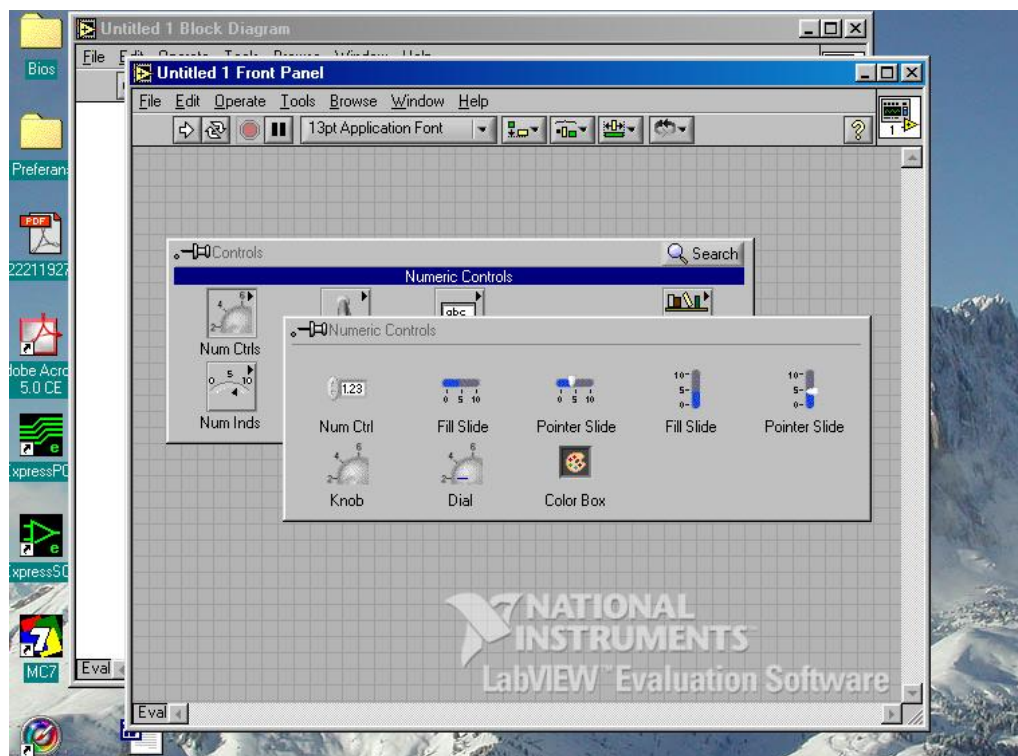
U ovom će primjeru biti pokazan način programiranja u grafičkom programskom jeziku LabVIEW na primjeru jednostavne matematičke operacije množenja.

Otvori LabVIEW i postavi prozor za formiranje prednje ploče (siva pozadina) vidljivim na ekranu kompjutera.



Slika 2-1

Da bi se realizirao ovaj primjer potrebne su dvije broјčane kontrole (za **varijablu a** i za **variablu b**). Osim toga trebat će i broјčani pokazivač za prikazivanje rezultata množenja. Da bi se pozvala neka kontrola potrebno je odabrati paletu kontrola *controls palette* (**Windows >> Show Controls Palette**) odnosno jednostavnije postaviti miš na sivu površinu prednje ploče i škljocnuti desnim mišem bilo gdje na sivu površinu prozora prednje ploče. Nakon što se pojavi prozor s paletom kontrola potrebno je miš postaviti na gornju lijevu ikonu. Pri tome će se pojaviti paleta naziva **Numeric Control** (za razliku od **Buttons**, **Text Control** itd.) te će zaslon monitora izgledati kao na slici 2.





Slika 2-2

Zatim treba škljocnuti ikonu **Num Ctrl** pri čemu nestaju sve palete i ostaje samo iscrtkani pravokutnik koji se može pomicati alatom za pomicanje po sivoj površini prednje ploče je što upravo ono što smo htjeli ostvariti. Iscrtkani pravokutnik treba postaviti na željeno mjesto na prednjoj ploči i škljocnuti jednom lijevom tipkom miša. Na prednjoj će se ploči tako smjestiti kontrole *numeric* u kojoj je upisana vrijednost 0 a zajedno s njom dolazi mali pravokutnik spreman za editiranje imena te kontrole. **NEMOJTE STISNUTI ENTER !** već preko trenutnog naziv kontrole **Numeric** (bijela slova na crnoj podlozi) upišiti novi (vaš) naziv te varijable (npr. **Var A** – upišite pomoću tipkovnice **Var A** !).

Da bi to ime bilo prihvaćeno i na prednjoj ploči i na blok dijagramu potrebno je u krajnje lijevom dijelu alatne trake pritisnuti kvačicu koja se u međuvremenu pojavila ili što je jednostavnije škljocnuti jednom lijevom tipkom miša na sivu površinu prozora prednje ploče.

Sada odaberi samo naslov kontrole i pokušaj ga pomaknuti po prozoru prednje ploče. Uočite da se ikona kontrole ne pomiče. Učinite sada isto samo s ikonom kontrole. Primjećujete da se sada pomiče ne samo ikona kontrole već i njen naslov. Vratite naslov kontrole uz komandu. Ovo je bitno znati zbog toga da bi znali kada im kako što pomičemo u programu LabVIEW.

A sad malo estetike !


Da ne bi prikaz prednje ploče virtualnog instrumenta bio sumorno sivo – crno – bijeli kao boje u poznu jesen potrebno je znati kako promijeniti boje kontrola i ostalih površina prednje ploče. Nema neke praktičnog razloga za to ali razvija maštu, pokazuje estetske sklinosti i zabavno je. Za to su vam potrebna dva alata koja su smještena na paleti alata. Odaberi **Windows>>Show Tools Palette** i paleta alata bi se trebala pojaviti (ukoliko već nije bila prisutna na ekranu !). Alat za kopiranje i lijepljene boje  nalazi se u trećem redu desno dok je alat za odabir boje i bojanje  smješten skroz na dnu alatne palete (sl. 3.)



Slika 2-3

Mogu se obojiti mnogi dijelovi kontrole ali u ovom će se primjeru obojiti samo slijedeća dva :

- Pozadinu prednje ploče
- Kontrole i indikator

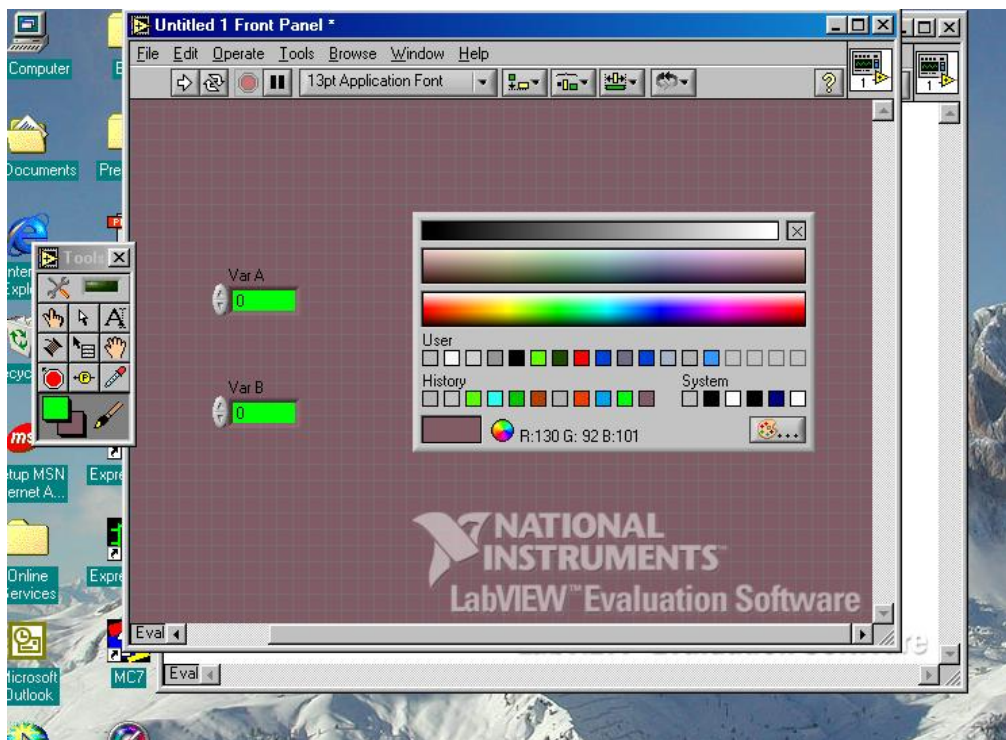
Da bi se pojavila paleta sa spektrom boja kao na slici 2-4. potrebno je škljocnuti na alat za bojanje  tako se kursor miša promijeni u oblik kista i zatim škljocnuti onu površinu koja se želi obojiti.

Treba uočiti da na paleti boja nalaze tri podpalete na kojima se mogu kontinuirano ili diskretno odabrati boja i njen ton.

Gornja je paleta monokromatska od crne boje, preko nijansi sive boje do bijele boje. Prvensrveno je namijenjena za bojanje pozadine prednje ploče.

Srednja je paleta također namijenjena za bojanje pozadine prednje ploče, boje nisu više monokromatske ali su boje i njihovi tonovi prigušeni.

Donja je paleta žarkih boja i tonova i namijenjena je tome da neku komandu na prednjoj posebno označimo.




Slika 2-4

Cilj Obojiti prednju ploču virtualnog instrumenta.

Najprije će se obojiti siva površina pozadine prednje ploče. Škljocnuti kistom na sivu površinu i zatim kist pomaknuti na srednju paletu boja. Uočiti da se boja pozadine mijenja s obzirom na položaj kista. Istovremeno se mijenja ista boja u donjem lijevom pravokutniku kao i postoji uz oznaku RGB. Kad se odabere željena boja i nijansa škljocnuti jednom lijevom tipkom miša tako da se tom bojom oboji površina prednje ploče.

Isti postupak treba ponoviti za bojanje numeričkih kontrola. Škljocnuti desnom tipkom miša dok j alat u obliku kista i s donje palete boja odabrati boju škljocom na lijevu tipku miša.

Boja se može odabrati osim da se odabere s jedne od tri palete i tako da se škljocne kistom na neku od diskretno ponuđenih boja.

Druga će se brojčana kontrole realizirati kloniranjem prve. Treba odabrati prvu komandu alatom za odabir. (srednjim u gornjem redu alatne palete ili odabrati  tako da se pri postavljanju miša na određenu komandu, tekst i ostalo vrsta alata automatski mijenja.) Treba istovremeno pritisnuti i držati lijevu tipku miša s odabranom brojčanom komandom i držati pritisnutu tipku **Ctrl** na tipkovnici. Lagano pomaknuti mišem klon numeričke komade ispod originala. Otpustiti lijevu tipku miša i tipku **Ctrl**.

Pojavio se klon prve numeričke kontrole s nazivom **Var A 2** koji će se ponašati kao druga nezavisna varijabla. Zgodnobi je bilo nazvati varijabla B. Postavite miš na iznad taksta A 2 škljocnite mišem tako da se pojavi alat za pisanje i upišite preko A 2 B tako da je naziv druge brojčane kontrole **Var B**.

Slijedeće što nam treba je *indikator* koji se nalazi na podpaleti *Numeric Indicator* palete *Controls*. Kao što se na toj je paleti ikona *Numeric Indicator* smješten ispod ikone *Numeric Control*. Štosa radi odabrat će se klizni tip indikatora generiran pomoću klizne kontrole (*Fill Slide*).


.Slijed radnji je:

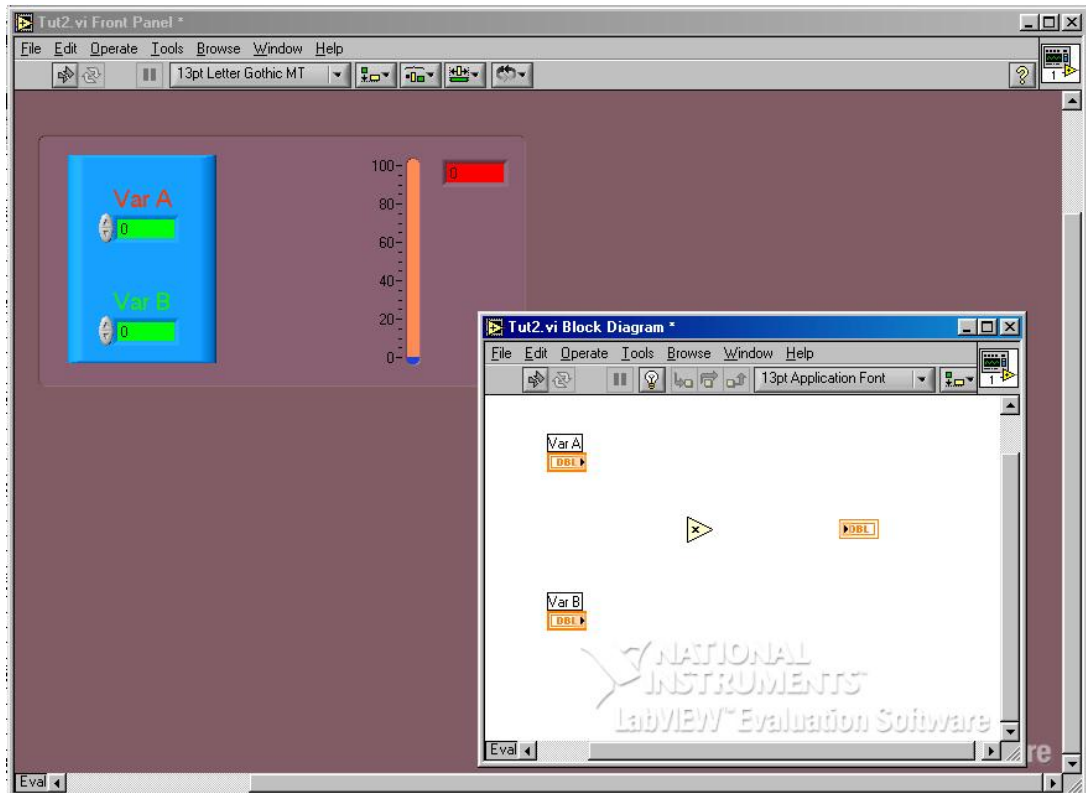
- škljoc desnim mišem na pozadinu prednje ploče
- postaviti miš na ikonu *Num Ctrls*
- škljoc na ikonu **Fill Slide** na podpaleti *Num Ctrls*
- prenijeti ikonu na prednju ploču alatom za pomicanje
- smjestiti ikonu u prozor prednje ploče škljocom na lijevu tipku miša
- odmah unijeti preko tastature naziv indikatora npr. **Umnožak**
- škljocnuti bilo gdje na površinu prozora prednje ploče
- škljocnuti desnom tipkom miša na **Fill Slide** i odabrati **Change To Indicator**
- obojiti indikator prema vlastitom ukusu

Konačno se može izgled prednje ploče dodatno urediti da ljepše izgleda. Pokušajte odabrati istovremeno obje kontrole (Koristite alat za odabir, počnite gore lijevo i pomaknite miš preko obaju kontrola do dolje desno držeći pritisnutu tipku miša) te i poravnajte okomito koristeći alat za poravnavanje iz gornjeg prozora.

Prijeđite sad na blok dijagram. Vidjet ćete zaleđe obaju kontrola i indikatora. Za početak ih možete pomaknuti u željene položaje (prisjeti se da treba pomicati komandu/indikator a ne samo njihove nazive!). Vrlo važno svojstvo programiranja u LabVIEWu je **dataflow programiranje** – naredba se izvršava kad su podaci potrebni za njeno izvršenje na raspolaganju i po konvenciji izvršavanje naredbi se odvija s lijeva na desno. Budući su u našem problemu obje ulazne varijabla na raspolaganju treba ih poravnati okomito prema lijevo tako da bi se rezultat pojavio na desnoj strani.

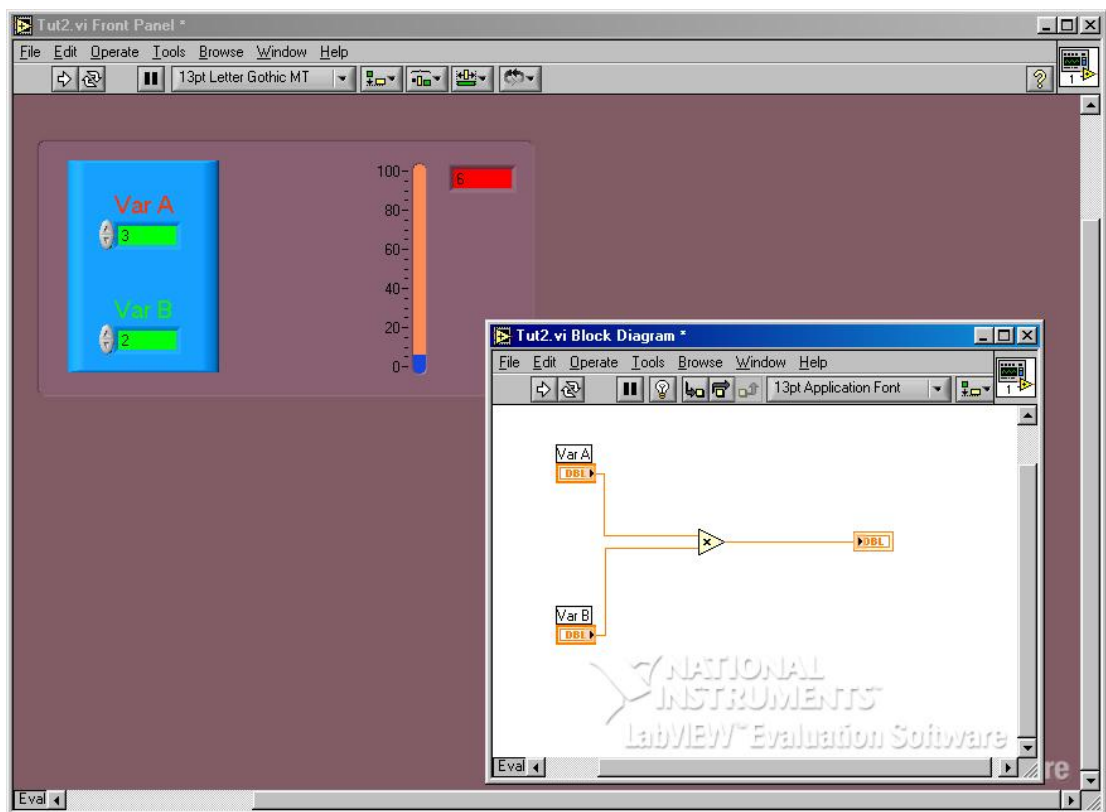
Između ulaza i izlaza potrebna je operacija množenja. Ista se može dobiti iz palete **Functions (Windows>>Show Functions Palette)** odnosno jednostavnije postaviti miš na bijelu površinu blok dijagrama i škljocnuti desnim mišem bilo gdje na nju. Nakon što se pojavi prozor s paletom kontrola potrebno je miš postaviti na donju desnu ikonu (**All Functions>> Numeric**) te izabrati operator množenja. Smjestite ga na sredinu prozora blok dijagrama između ulaza i izlaza.

Sad je potrebno povezati kontrole s ulazima operatora množenja, izlaz s indikatorom alatom za povezivanje. U LabVIEWu je to špulica žice . Ona će se automatski pojaviti ukoliko se alat postavi na izlaz kontrola, ulaz indikatora, ulaz ili izlaz operatora. Krenimo od prve ulazne kontrole. Kad postavimo alat na desni kraj kontrole pojavit će se špulica žice a ikona kontrole će početi treptati. Da bi pričvrstili žicu za izlaz kontrole treba pritisnuti lijevu tipku miša dok ikona treperi držati je stisnutu i pomicati miš dok se ne dosegne gornji lijevi ulaz operatora množenja koji pri tom također počne treperiti i pojavi si natpis **x**. Tada otpustite lijevu tipku miša. Između izlaza prve ulazne kontrole i ulaza **x** operatora pojavit će se narančasta žica. Ukoliko se pojavi crkana žica pritisnite **Ctrl-B** (briše loše spojene žice) i pokušajte ponovno. Istim postupkom spojite drugu ulaznu komandu na **y** ulaz (lijevo dolje) operatora i njegov izlaz na indikator.





Slika 2-5

Konačni izgled ekrana treba biti poput onog sa slike 2-6.



Slika 2-6

Prijeđite na prednju ploču i pokrenite VI. Buduće da su pretpostavljene vrijednosti obaju ulaza 0.0, odgovor je 0.0. Da bi se provjerila funkcionalnost VI postavite vrijednost prvog ulaza na npr. 2.0 a drugog na 3.0. Pri tom VI mora biti zaustavljen. Provjerite izgleda li indikator za pokretanje VIa ovako . Sad upotrijebite alat za provođenje operacija  i promijenite vrijednost ulaznih kontrola škljocom na strjelice uz ulazne kontrole (škljocom na strjelicu usmjerenu gore povećavate vrijednost ulazne varijable u slijedu 1.0, 2.0 ...) ili alatom za pisanje upišite željene vrijednosti preko 0.0. Pokrenite ponovno VI i rezultat bi trebao iznositi 6.0.

Da bi spremili svoj rad odaberite iz alatne trake **File>>Save As** spremite file u željeni direktorij odgovarajućim imenom. Ovo je konvencionalni način spremanja fileova. U LabVIEWu se mogu osim toga srodni fileovi mogu spremati u biblioteke (*Library*).

2.1. Ulazni podaci i način njihova prikazivanja

Sad ćemo isprobati neke varijacije u gornjem primjeru. LabVIEW naime ima tu sposobnost da nadzire način na koji je ulaznim podacima potvrđuje vjerodostojnost i kako će biti prikazani. Sve se to nadzire pomoću padajućeg menua koji se pojavi kad škljocnemo desnim mišem na pojedinu komandu, indikator i ostale elemente prednje ploče i blok dijagrama. Neki od najkorisnijih stvari koje se mogu učiniti su:

Promijeniti pretpostavljene vrijednosti	Odaberi Data Range i promijeni <i>Default Value</i>
Promijeniti ispis za koji se varijabla mijenja kad se škljocne «gore» i «dolje» strjelica	Data Range makni kvačicu u <i>Use Default Range</i> i promijeni vrijednost <i>Increment</i>
Promijeniti dizvoljeni opseg podataka	Data Range i promijeni <i>Maximum</i> i <i>Minimum</i> vrijednosti
Promijeni što se događa na i iznad granične vrijednosti	Data Range i odaberi “If value is out of range” - opcije su: <i>Ignore</i> Ne provodi provjeru granica <i>Coerce</i> Ako je vrijednost izvan granice, zaokružava se na najbližu graničnu vrijednost. Strjelicom za povećanje/smanjenje vrijednosti neće se premašiti opseg <i>Suspend</i> Ako vrijednost ode izvan opsega izvršenje VIa se prekida – može se dodati poruka greške
Promijeniti način na koji se padaci prikazuju	Format & Precision i učini odgovarajuće promjene

Probaj promijeniti nekoliko postavki i pogledaj kako djeluje na prikazane vrijednosti.

Može se također promijeniti i opseg prikazivanja termometra mijenjajući maksimalnu vrijednost na pokazivaču pomoću alata za pisanje.

2.2. Ukrašavanje prikaza prednje ploče

Moguće je još na prednju ploču dodati naslov, boje i razne ukrase.

Naslovi se dodaju alatom za pisanje a izgled teksta se može promijeniti odabirom teksta i korištenjem menua fontova na vrhu ekrana.

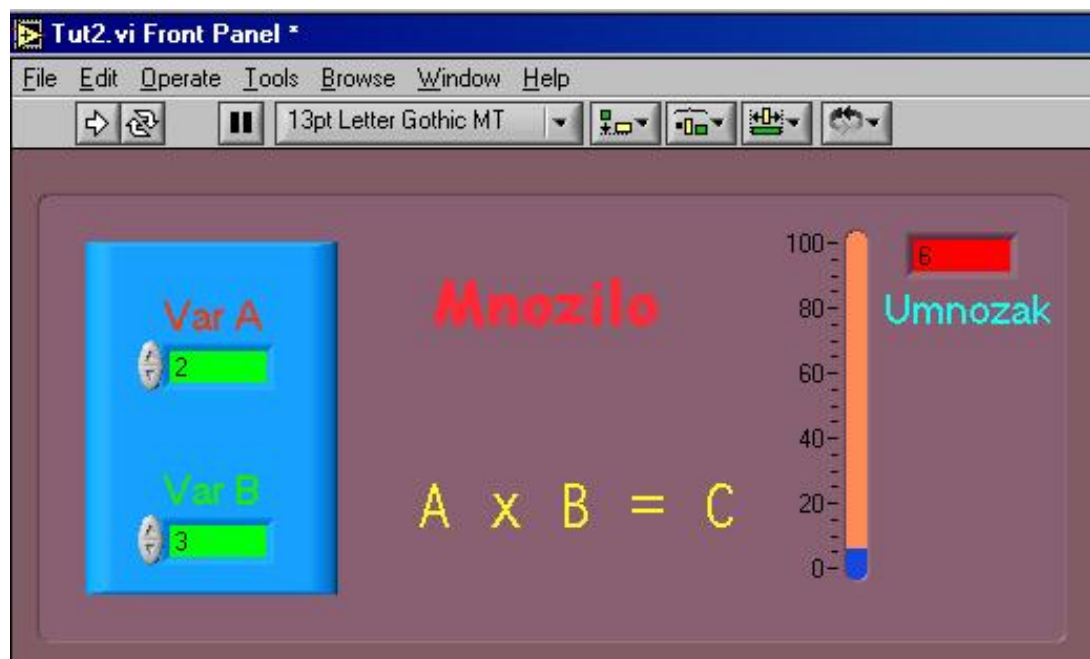
Boje se dodaju alatom za bojanje kao što je već objašnjeno.

Dekoracijama se može pristupiti na podpaleti koje se mogu pojedinačno obojiti. Veličina dekoracije se može mijenjati ukoliko postavimo miš u njihov ugao i zatim povlačenjem povećamo ili smanjimo njenu površinu.

Možete pokušati i sa slojevitim prikazom (*layer*) – jedna pojedinost stoji iznad druge – tako da je samo gornji objekt vidljiv. Dakle ukoliko se postavi dekorativna podloga iza kontrole to je upravo ono što se želi. *Layer* se kontrolira iz **Edit** menua i odabrana pojedinost se može postaviti iznad, ispod, ispred i iza layera. Većinom je dovoljno koristiti samo postavljane kontrole ispred ili iza layera.

Konačni izgled prednje ploče može izgledati kao na slici .

Pravi unikat, zar ne?



Slika 2-7

3. Napredovanje u LabVIEWu

3.1. WHILE petlja i sklopke

Nastavit ćemo s LabVIEWom učeći kako stvoriti petlju unutar koje će se program odvijati i završiti kad se ta petlja zaustavi. Takav je način zaustavljanja VI, kao što je već objašnjeno mnogo primjereniji nego zaustavljanje tipkom s alatne trake STOP. Koristit će se petlja WHILE koja se izvršava stalno iz početka tako dok je upravljačka varijabla u stanju **true**.

Može se iskoristiti već napisani VI, staviti u petlju tako da se mogu mijenjati vrijednosti ulaznih varijabli i promatrati promjene izlaza. Usvojiti će se osim toga način korištenja sklopke da bi se VI zaustavio kad to želimo.

Prijeđite na blok dijagram VI *množilo*, škljocnite desnom tipkom miša na površinu prozora i nakon što se pojavi paleta **Functions** potražite **All Functions** pa **Structures** i u njoj **While Loop**. Pojaviti će se oznaka petlje koju treba postaviti u lijevi gornji kut blok dijagrama i držeći pritisnutu lijevu tipku miša povući petlju tako da uokviri sve elemente blok dijagrama.

Uočite dva kvadratića u donjim kutovima. U lijevom kutu je varijabla **I** koja broji (od 0) koliko se puta petlja odvila. U desnom je kutu booleova (logička) upravljačka varijabla. Ukoliko je njen ulaz **true** petlja se izvršava, ukoliko je njen ulaz **false** petlja se zaustavlja na kraju trenutne iteracije.

Naučimo sad jednu LabVIEW prečicu. Škljocnimo desnim mišem na okvir petlje i odaberimo *create indicator*. Na prednjoj se ploči pojavi digitalni indikator. To je stvarno brz način kreiranja indikatora ukoliko vam treba. Indikator možete pomicati okolo i imenovati ga odabirući indikator i birajući **show>>label** s padajućeg izbornika (desna tipka miša).







Ista se stvar može učiniti s kontrolom petlje – iskoristi padajući da kreiraš kontrolu. Koristeći alat za provođenje operacija možeš postaviti komandu na položaj **ON** i pokrenuti VI. Indikator petlje će pokazivati da se petlja odvija vrlo brzo. Sad se ulazne varijable mogu mijenjati u dinamičkim uvjetima a izlaz pokazuje odgovarajuće rezultantne vrijednosti. Kad stisnete tipku **OFF** VI se zaustavlja (strjelica RUN stane a gumb STOP se zamuti). Probaj sad ponovno pokrenuti VI pritiskom na gumb RUN. Ništa se ne događa zbog toga što je gumb **ON/OFF** još uvijek u položaju **OFF**. Da bi se VI pokrenuo ponovno morat ćete postaviti gumb **ON/OFF** u položaj **ON** i pokrenuti VI.

Prilično nespretno i neuobičajeno, zar ne? Trebao bi nam pametniji gumb **ON/OFF**.

Ukoliko ste, na početku, pomislili da su booleove sklopke dosta jednostavne – one to nisu – jer postoji mnogo pitanja u svezi s njima:

- želite li da sklopka zadrži novu vrijednost ili samo reagirati kad je stisnete?
- želite li da se to dogodi kad stišćete tipku miša ili kad je otpuštate?
- želite li biti sigurni da LabVIEW učita sklopku u njenom novom stanju?


To vodi k šest različitih tipova sklopki:


Ikona	Zadržati?	Akcija uključenja	Koliko puta LabVIEW učitava položaj sklopke
	DA	STISNI	Mnogo puta
	DA	OTPUSTI	Mnogo puta
	NE	STISNI/OTPUSTI	Možda nikad (prebrzo otpuštanje) Možda mnogo puta (presporo otpuštanje)
	NE	STISNI	Jednom – vraća se na početni položaj nakon što ga LabVIEW pročita
	NE	OTPUSTI	Jednom – vraća se na početni položaj nakon što ga LabVIEW pročita
	NE	STISNI	Barem jednom, a možda više puta – vraća se na početni položaj nakon što ga LabVIEW pročita i tipka je otpuštena

Za naš je primjer najprimjerenija sklopka koja ne zadržava položaj i vraća se na početnu vrijednost nakon što je LabVIEW učita (četvrta na listi). Promijeni vrstu booleove komande u taj tip pomoću padajućeg a izabirući **Mechanical action**. Također se treba promijeniti početno stanje gumba na **ON** (ručno podesite gumb na **ON** i postavite kao pretpostavljenu vrijednost).

Sad je za pokretanje VI potrebno samo pritisnuti gumb RUN. Program se zaustavlja pritiskom na booleovu sklopku **ON/OFF**. Da bi osigurali da netko tko ne poznaje zakonitosti rada u LabVIEW ne bi zaustavio izvršavanje VI pomoću naredbe STOP iz alatne trake možete je od tamo maknuti. (**File >> VI Properties >> Category >> Window Appearance >> Custom >> Customize >> Show Abort Button >> maknuti kvačicu**)

3.2. Prikaz dinamičkih signala

Bilo bi interesantno vidjeti kako bi se mogao prikazati umjesto statičkog ulaznog signala dinamički koji bi se mijenjao 10 puta u sekundi ili brže. Da ne bi korisnik u ovom primjeru jednu od varijabli mijenjao na taj način postoji u LabVIEWu, koji je vrlo bogat različitim funkcijama, generator slučajnih brojeva (0...1) koji se mijenjaju vrlo brzo (ovisno o taktu računala). Da bi se zamijenila jedna varijabla (npr. Var B) tim generatorom najprije je potrebno odabrati tu varijablu na prednjoj ploči i pritisnuti **Delete** na tipkovnici da bi ju izbrisali (tim postupkom nije moguće izbrisati komandu s blok dijagrama). Zatim treba prijeći na blok dijagram i s **Ctrl-B** izbrisati loše vodove. Najbrži način dolaženja do generator slučajnih brojeva (koji izgleda kao par kockica za igru ) je taj da se na blok dijagramu umjesto na površinu tog prozora desnom tipkom miša škljocne na neku od ikona. Pojavit će se padajući gdje treba izabrati **Express Numeric Palette >> Random Num.**

Standardni (i sporiji) je način odabira nakon škljoca desnom tipkom miša na površinu prozora blok dijagrama **Functions >> All Functions >> Numeric >> **.

Generator slučajnih brojeva smjestite unutar WHILE petlje i povežite špulicom na operator množenja.

Ukoliko se pitate kako generator zna kad treba «generirati» novu slučajnu vrijednost odgovor je slijedeći. LabVIEW pita svaku komponentu programa da li ima vrijednost na raspolaganju. Digitalna komanda kaže «da» i proizvede vrijednost komande, generator slučajnih brojeva također kaže «da» i proizvede novi slučajni broj.

Ukoliko sada pokrenete VI «živa» u termometru će početi drhtati ukoliko ste odabrali odgovarajuće skaliranje odabirom faktora množenja Var A i mjerni opsega termometra.

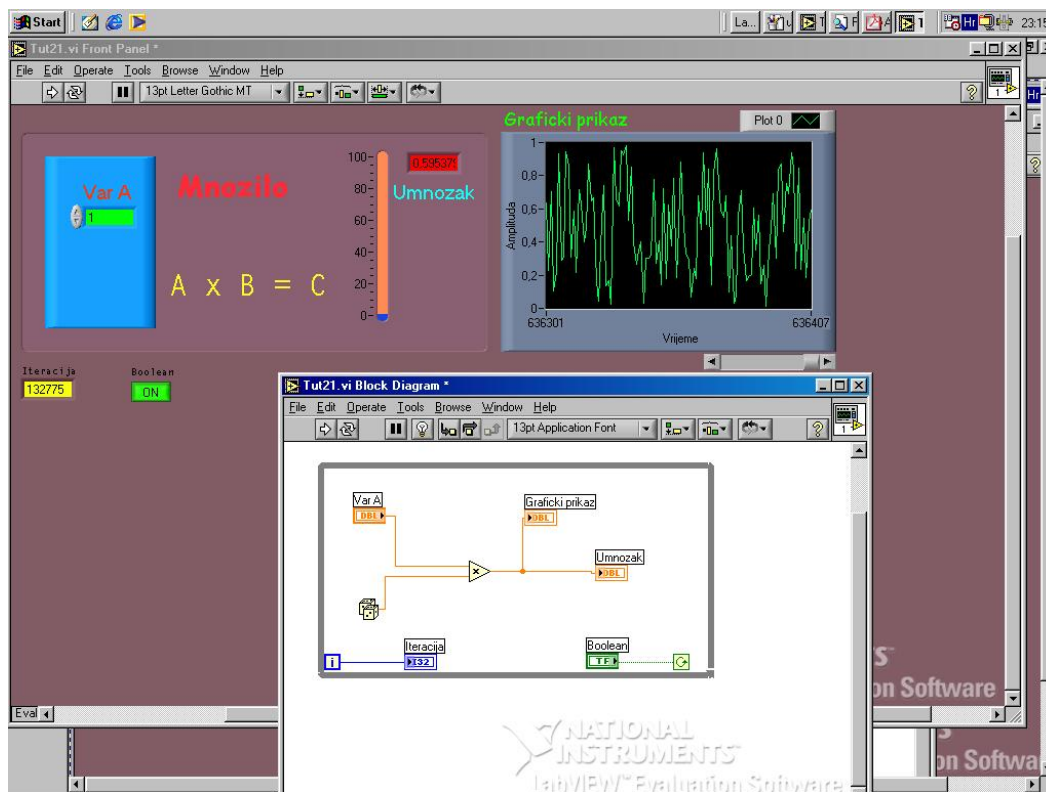
Bilo bi također vrlo lijepo ukoliko bi imali prikaz izlaza u vremenu – grafikon ili prikaz. LabVIEW razlikuje dvije vrste grafičkih prikaza, **graph** u kojem su svi podaci dostupni zajedno (kao u polju) i **chart** kad je samo jedan podatak na raspolaganju u nekom trenutku (kao prikaz na pomičnom papiru). Dodat ćemo **chart** VIu. Za to je potrebno najprije povećati površinu prozora predenje ploče (povuci desnu ivicu prozora lijevom tipkom miša) dok ne dobiješ po prilici dvostruko veću površinu prozora. Sad odaberi **controls>>graph>>waveform chart** i postavi **chart** u desni dio prednje ploče. Dodaj naslov prije škljoca mišem na prednju ploču.

Chart se na blok dijagramu može spojiti paralelno termometru tako da postavim alat mišem na žicu između operatora množenja i izlaznog indikatora (pojavit će se špulica a žica će početi treperiti). Stisnite desnu tipku miša i povucite žicu do **charta**. Budite sigurni da je indikator unutar WHILE petlje.

Nakon što pokrenete VI vidjet ćete prikaz 50 posljednjih vrijednosti jer je **chart** u stvari pomični prozor za posljednjih 50 vrijednosti. Pomoću padajućeg izbornika moguće je kontrolirati veliki broj parametara. Možete odrediti broj vidljivih vrijednosti mijenjajući vrijednost horizontalne ljestvice i kako su te vrijednosti raspodijeljene mijenjajući veličinu okvira i povlačeći ga horizontalno. Možete pogledati i povijest prošlih vrijednosti koje nisu trenutno prikazane na **chartu** korištenje mogućnosti spremnika te horizontalnu ljestvicu s kliznom trakom. Pretpostavljene vrijednost spremnika su 1024 vrijednosti i 50 vidljivih na prikazu. Aktiviranjem klizne trake pomoći padajućeg izbornika **show>>scrollbar** možete prijeći preko svih pohranjenih vrijednosti. Isto tako se može promijeniti duljina skupa pohranjenih podataka pomoću **Chart History Length**.

Možda će neki od Vas uočiti da s uključenim **chartom** VI se izvršava nešto sporije očito zbog toga što više prikaza zahtjeva više procesiranja odnosno više vremena za izvršavanje pojedine petlje a to rezultira manjim rojem izvršenih petlji u sekundi. O tome se i te kako mora voditi računa.

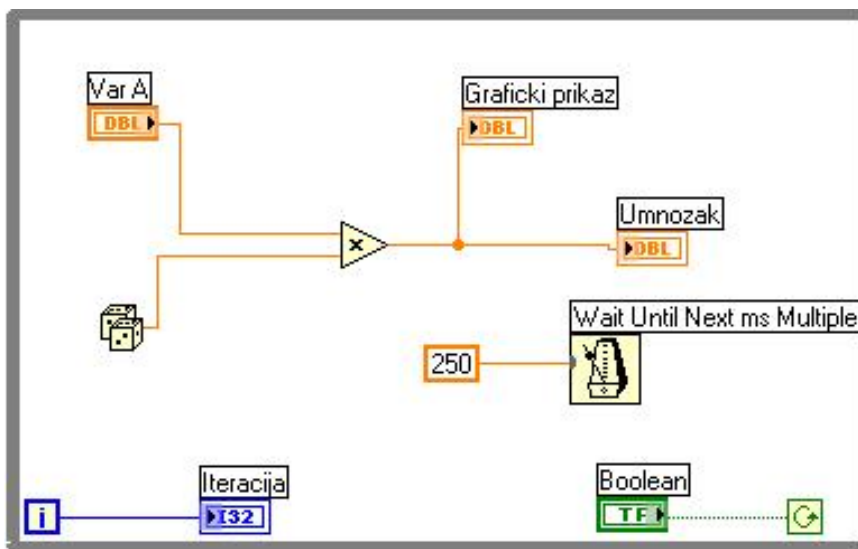
Izgled VI bi sad trebao biti poput ovog:



Slika 3-1.

3.3. Što učiniti ukoliko Vam je izvršenje VIa prebrzo !?

Sad, nakon što ste vidjeli kako se brzo VI izvršava možda ćete zaključiti da je to za Vas prebrzo. Možda bi odvijanje jedne petlje svakih 250 ms bilo sasvim lijepo. Vremensko nadziranje petlje sastoji se u tome da se u petlju uključi vremenski element (npr. Poput metronoma koji nadzire takt sviraču muzičkog instrumenta). Odaberite pomoću **All Functions>> Functions>>Time & Dialog>>Wait Until Next ms Multiple** sličicu metronoma, smjestite je unutar petlje i pomoću padajućeg izbornika **Create>>Constant** napravite na ulazu metronoma konstantu vrijednost 250 (vrijednost u ms). Sad će se petlja odvijati četiri puta u sekundi.



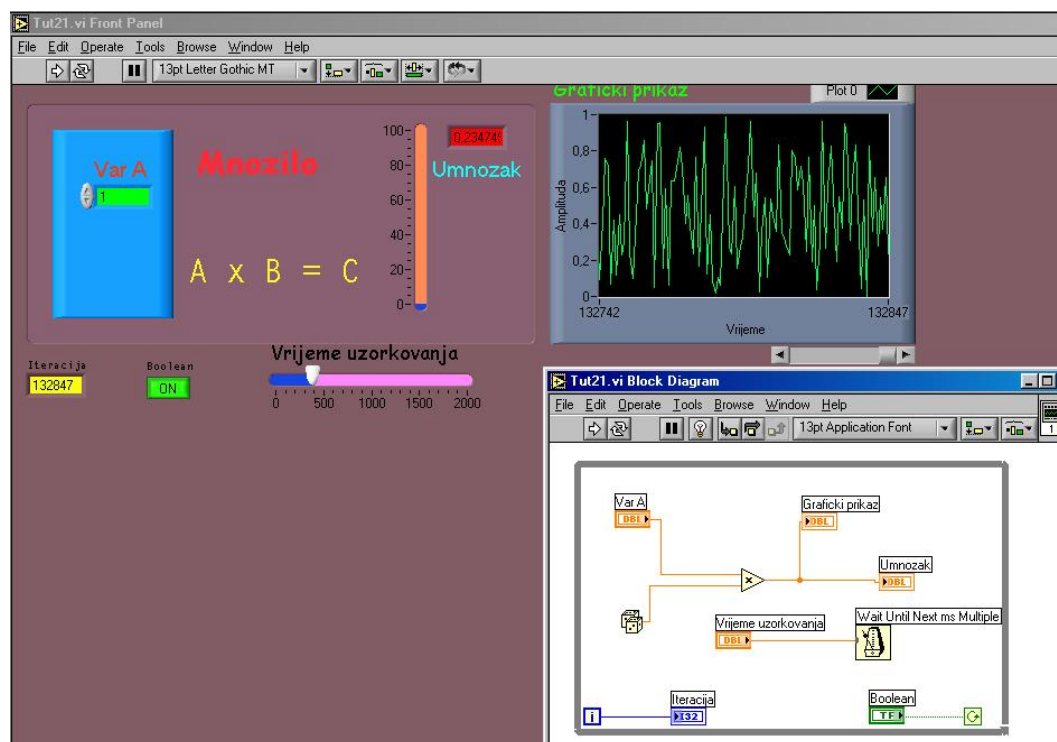
Slika 3-2.

Konačno može se i ručno upravljanje brzinom izvršavanja WHILE petlje.

Zamijeni brojčanu konstantu **250** kontrolom te na prednjoj ploči zamijeni brojčanu kontrolu s **Horizontal Pointer Slide**. Upišite preko krajne vrijednosti klizača **10** novu vrijednost **2000**.

Sada ćete moći kontinuirano mijenjati brzinu izvršavanja WHILE petlje od 2000 ms tj. od 2 sekunde do maksimalne brzine koja ovisi procesorskoj brzini računala. **Probajte !**

Konačni izgled prednje ploče i blok dijagrama je za ovaj primjer prikazan na slici



Slika 3-3.













4. Predstavljanje podataka i pomoć

4.1. Predstavljanje podataka

Na Vašem će blok dijagramu žice bit obojene trima bojama ovisno trima vrstama podataka koji se njima prenose

- Narančasta/smeđa - s DBL
- Plava - s I32
- Točkasta zelena - s TF

LabVIEW podržava veliki broj vrsta podataka i njihove pretvorbe. Glavne vrste numeričkih varijabli su:

Skraćenica	Vrsta	Duljina spremnika (Byte)	Minimalna vrijednost	Minimalna vrijednost
EXT 	Realni proširene točnosti			
DBL 	Realni dvostruke točnosti	8		
SGL 	Realni jednostruke točnosti	4		
I32 	Dugi cjelobrojni	4		
I16 	Cjelobrojna riječ	2	-32766	32767
I8 	Cjelobrojni byte	1	-126	127
U32 	Pozitivni dugi cjelobrojni	4	0	
U16 	Pozitivna cjelobrojna riječ	2	0	65535
U8 	Pozitivni cjelobrojni byte	1	0	255
CTX 	Kompleksni proširene točnosti			
CDB 	Kompleksni dvostruke točnosti			
CSG 	Kompleksni jednostruke točnosti			

Budući da se slične aritmetičke operacije mogu izvesti na svakoj od tih vrsta, aritmetičke se funkcije, komande i indikatori moraju nositi s tim. Za ostvarenje toga koriste se tri različite tehnike: promjena vrste, implicitna pretvorba vrste i eksplicitna pretvorba vrste.

Da bi se vidio primjer prve i druge upotrijebi padajući izbornik na ulaznoj komandi u operator množenja i promijeni način prikazivanja podataka. Uoči da su žice narančasto/smeđe za lebdeći a plave za neoznačeni cjelobrojni.

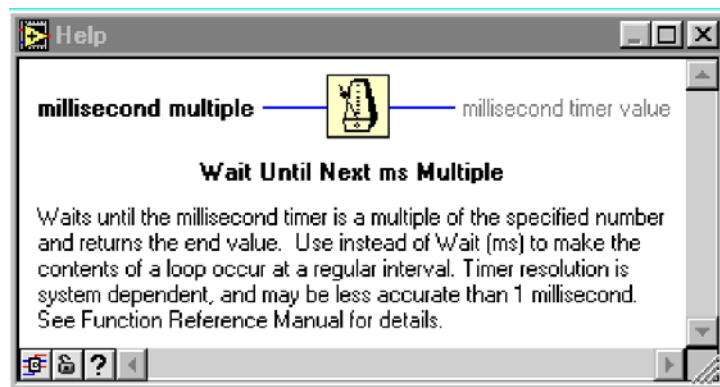
Da bi se vidio primjer primjer eksplicitne pretvorbe promijenite vrstu predstavljanja izlaza u cjelobrojnu riječ (To Word Integer). Pobrišite žice s izlaza operatora množenja do *charta* i indikatora pomoću alata za izbor i tipke **Delete** na tastaturi. Pomoću **Ctrl-B** pobrišite preostale prekinute vodove. Odaberite **All functions>>Functions>>Numeric>> Conversion >>To Word Integer** i namjestite pretvornik nasuprot izlazu operatora množenja. Kompletirajte ožičenje s operatora množenja na ulaz pretvornika a s njega na *chart* i indikator. Primjećujete da se *chart* automatski promijenio u **I16** kad ste ga spojili na konverter a indikator je ostao **DBL** i ali ipak provodi pretvorbu što je indicirano konverzijskom točkicom. Možete promijeniti vrstu indikatora odabirući u padajućem izborniku pod **Representation I16** nakon čega vrsta podataka postaje konzistentna po svim elementima i konverzijska točkica nestane.

Postoje mnogi slučajevima pri programiranju u LabVIEW gdje ćete morati bit svjesni vrste podataka i osigurati da se konverzija tipa podataka odvija korektno.

4.2. Pomoć !

U LabVIEWu postoje dva oblika pomoći – LabVIEWeva pomoć Vama, programeru, i Vaša pomoć korisniku za kojeg pišete program.

LabVIEW pruža pomoć na mnogo načina. Postoji ugrađeni sustav pomoći (pod **Help>>VI, Functions & How To Help...** ili **Ctrl ?**) koji vam pruža opširnu pomoć o LabVIEWu. Pojavit će se prozor pomoći gdje se unutar opcija **Contents**, **Index**, **Search** i **Favorites** može pronaći sve od interesa u svezi



Slika 4-1.

LabVIEWom.

Osim takovog, uobičajenog načina pomoći postoji u LabVIEWu puno specifičniji sustav pomoći koji govori o svakoj pojedinosti kad se na nju dovede mišem alat za odabiranje. Taj je sustav dostupan pod **Help>>Show Context Help**. Tako se za svaku odabranu pojedinost s prednje ploče ili blok dijagrama dobije kratki sažetak pojedinosti. Kako to izgleda za izabrani metronom prikazano je na slici.

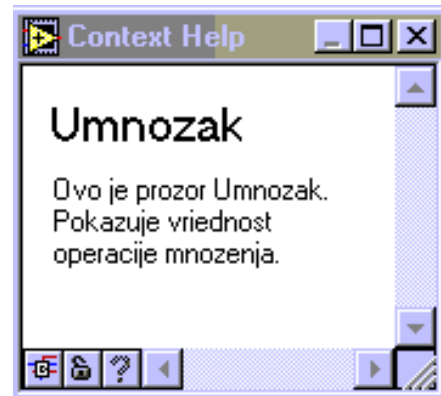
Ako ne želite da vam ta vrst pomoći bude stalno uključena već samo kad vi to želite zgodno je koristiti u tim slučajevima prečicu **Ctrl-H**.

Uočit ćete također da ukoliko stavite alat za odabiranje na komandu ili indikator da će se u prozoru pokazati njen naziv i informacija **“No description available”**. Isto se može promijeniti odabirom u padajućem izborniku **Data Operations>>Description**. Pojavit će se jednostavan prozor u koji možete upisati korisna informacij o toj pojedinosti. Nakon toga kad Vi ili netko drugi potraži pomoć za tu pojedinost pojavit će se i taj opis. Ovdje je kratka poruka koja se može upisati uz termometar **Umnožak**.(Sl. 4-2.)

Pisanje tih opisa predstavlja veliku olakšicu korisniku programa. Dokumentacija je prokletstvo i spas mnogih programa i nešto vremena potrošenog na pisanju tih stvari u ranoj fazi stvaranja VI može uštediti mnogo vremena u kasnijoj fazi.

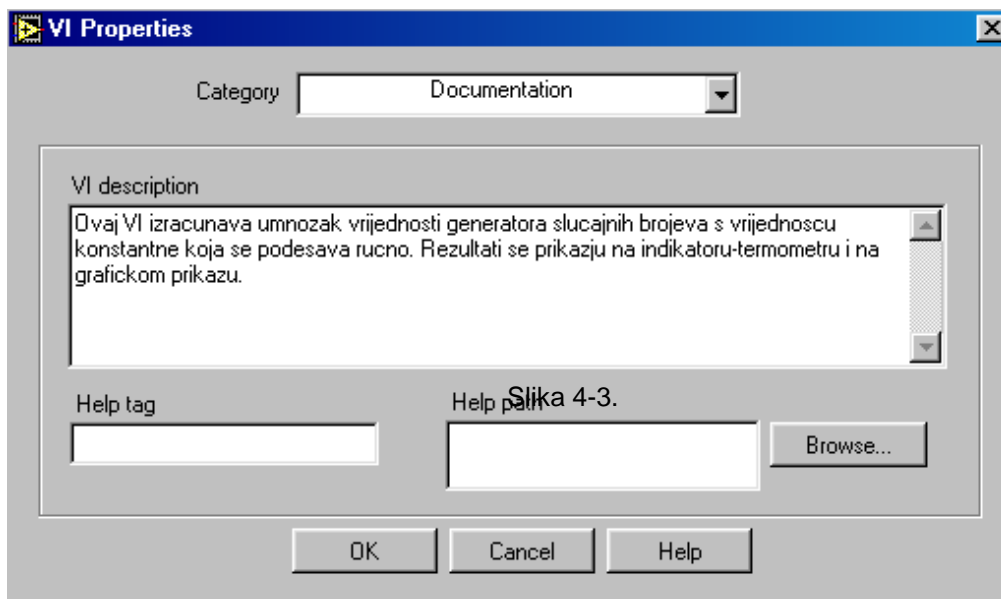
Konačno tu je i prostor za cjelokupni opis VI.

U tom se prostoru može upisati cjelokupni opis VI i način njegovog funkcioniranja. Ponovno je vrlo korisno kao nedjeljiva dokumentacija VI



to

Slika 4-2.



Slika 4-3.

Slika 4-3.

5. Pomični spremnici i FOR petlja

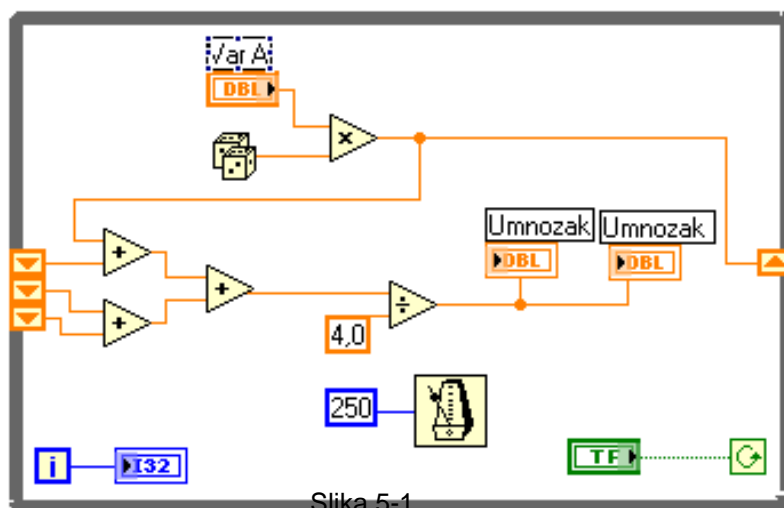
5.1. Pomični spremnici (*shift register*)

Često se pri mjerenju neke veličine želi obrađivati podaci koji su prikupljeni u nekoliko zadnjih iteracija. Npr. želi se izračunati srednja vrijednost posljednjih četiriju vrijednosti generiranih generatorom slučajnih brojeva. Za takav izračun je neophodan pomični spremnik. On uzima podatak pri jednoj iteraciji i daje ga na izlazu pri slijedećoj iteraciji. Zbog čega se takav postupak zove pomični spremnik. Zbog toga što njegova duljina može biti po volji velika a vrijednosti se pomiču za jedan element u nekom trenutku. Podtoji dakle jedan ulaz i mnoštvo izlaza.

Pretpostavimo da postoji pomični registar s četiri elementa koji sadrži vrijednosti 66, 87, 54, 39. Nova vrijednost od 82 ulazi u spremnik i sve se pomiče za jedan korak. U pomičnom j spremniku sad 82, 66, 87, 54. U slijedećoj iteraciji ulazi i u spremniku je sad 33, 82, 66, 87 itd.

Pomični se spremnik može stvoriti tako da se škljocom na desnu tipku miša na okvir WHILE petlje otvori padajući izbornik i odabere **Add shift register**. Pojavit će se dvije nove ikone na desnoj (ulaz) i lijevoj (izlaz) strani petlje. Otiđite na izlaz (lijevo) i otvorite padajući izbornik škljocom na desnu tipku miša i odabere **add element**. Ponovite isto još jednom i imat ćete tri elementa kojima će se vrijednosti pomicati a nova (četvrta) vrijednost će dolaziti uvijek na vrh spremnika.

Rearanžirajte dosadašnji VI tako da pokazuje srednju vrijednost posljednjih četiriju slučajno generiranih vrijednosti. Izgled blok dijagrama bi trebao biti kao na slici 5-1.

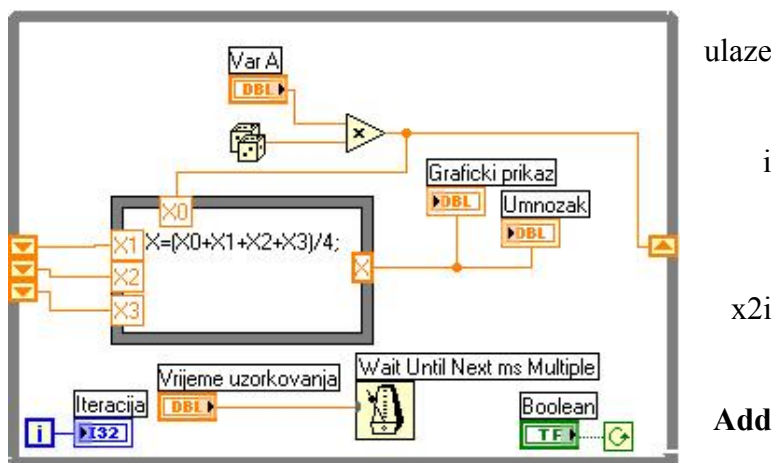


Pokrenite VI i trebali bi dobiti očekivane vrijednosti.

Ovakav način izračunavanja neke vrijednosti je pregledan samo ukoliko se radi o manjem broju varijabli i manjem broju operatora jer s slučajju većeg broja istih blok dijagram postaje nepregledan.

U izborniku **strukture** pored petlje i ostaloga postoji pojedinost zvana **formula čvor (formula node)**. Taj je čvor stvoren da bi olakšao mnoge stvari. Uklonite operatore množenja i dijeljenja za izračunavanje srednje vrijednosti i unesite formula čvor u ispražnjeni prostor.

Na okviru formula čvora treba stvoriti tunele za i izlaze koji moraju imati nazive poput varijabli u algebarskom izrazu što u stvari jesu. Formulu, zatim, treba utipkati u unutar okvira formula čvora. Ukoliko se ulazne varijable nazovu x0, x1, x3 a izlaz x kreiramo ih otvarajući padajući izbornik na okviru formula čvora i birajući **Input** i **Add Output** respektivno. U okvir upišimo formulu za srednju vrijednost četiriju posljednjih generiranih slučajnih brojeva u obliku $x=(x_0+x_1+x_2+x_3)/4$; (nemojte zaboraviti na kraju retka točku-zatez (;), važan je zbog sintakse). Pokrenite VI sad i sve bi se trebalo odvijati kao i prije.



Slika 5-2.

Iako su u ovom primjeru oba primjera jednako dobra (ili loša) kao i drugi u slučaju složenijih matematičkih izraza formula čvor bi imao prednost zbog veće preglednosti i jasnoće.

FOR petlja

Sada će se uvesti nova petlja dostupna u programu LabVIEW i time uvesti koncept polja (array). Problem se sastoji u prezentiranju srednje vrijednosti i standardne devijacije određenog broja prikupljenih podataka (vrijednosti) pomoću VI. Isto se može ostvariti kao što je objašnjeno u primjeru prije eksplicitno ali je to najčešće dosadno i zamorno pa u LabVIEWu postoji za takav slučaj bolji način rješavanja problema. Potrebno je najprije grupirati podatke u polje te ih nakon toga analizirati.

Nastavit ćemo s blok dijagramom koji sadrži pomični registar i da bi sve bilo jednostavnije obrišite pomični registar s ruba WHILE petlje i kod za izračun srednje vrijednosti u tom primjeru.

FOR petlja je tako konstruirana tako da se pokazuje vrlo zgodna za korištenje jer automatski sprema neke izlazne podatke u polje. Da bi se uvjerali u to probajte slijedeće. Otvorite **Functions >> All Functions >> Structures >> For Loop** i povucite je preko čitavog source koda VI ostavljajući brojilo petlje i sklopku za prekid programa izvan.

Kao što se vidi FOR petlja ima dva broja koja su joj pridružena:

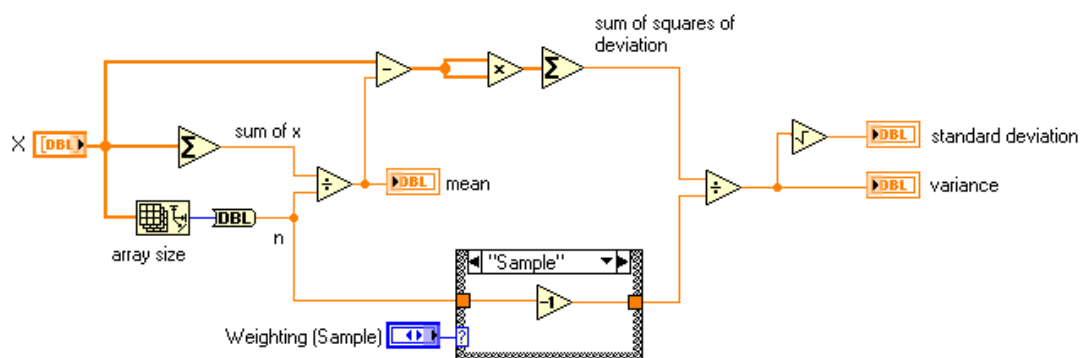
- Broj za završavanje **N**
 - u gornjem lijevom kutu
 - koji predstavlja broj koliko će se puta odvit FOR petlja prije završetka
- Broj iteracija
 - u donjem lijevom kutu

Dovedite miš do ulaza **N** škljocnite desnom tipkom miša nakon što se je pojavila špulica i dodajte kontrolu (izvan petlje).

U ovom je trenutku zgodno objasniti kako se najlakše može pronaći stvorena kontrola na prednjoj ploči. Škljocnite desnim mišem na stvorenu kontrolu i odaberite **Find Control**. Otvorit će se prednja ploča s označenim položajem numeričke kontrole za odabir broja izvršenja FOR petlje.

Da bi se mogle smjestiti ikone VI za izračun srednje vrijednosti i standardne devijacije izvan FOR petlje a unutar WHILE petlje potrebno je malo raširiti WHILE petlju.

Sad nađite **Functions>>All Functions>>Analysis>>Probability and**



$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

Slika 5-3.

Population standard deviation

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

Sample standard deviation

Statistics>>Standard Deviation. Taj VI uzima za ulaz polje podataka i daje srednju vrijednost i standardnu devijaciju tog polja na izlazu. Sigurno će Vam trebati pomoć s tim modulom stoga upotrijebite **Ctrl-H**. Ožičite ulaz s izlazom **I16** pretvornika u FOR petlji. Dok to radite učit ćete dvije stvari:

1. pojavio se crni kvadratić na granici petlje indicirajući da se nešto događa
2. žica unutar petlje postaje tanka a izvan petlje debela

Razlozi su u tome što crna točka skuplja pojedinačne izlaze svake iteracije petlje u polje čija je veličina broju izvršenja petlje - **Termination count**.

Najvažnije je postaviti broj **Termination count** \geq i tad pokrenuti VI. Pokušajte eksperimentirati s brojevima >0 .

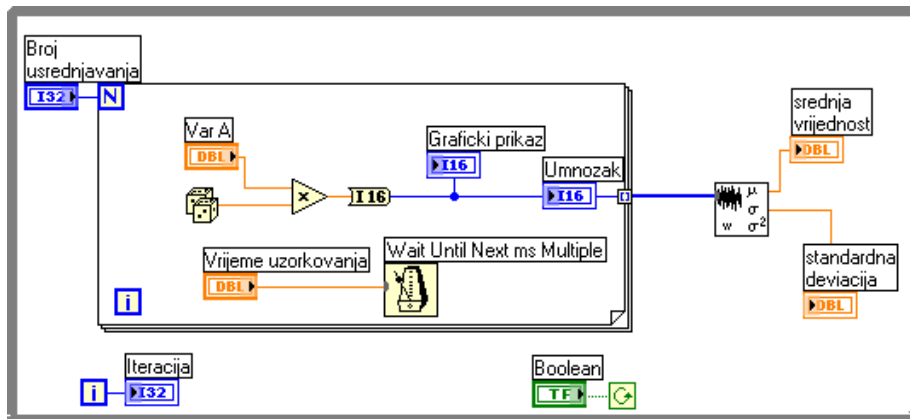
Zbog čega N ne može biti 0 ?

Pa, kakvo bi to bilo polje s 0 elemenata i kako bi se našla srednja vrijednost i standardna devijacija. Ta situacija sigurno vodi k grešci.

Tu situaciju možemo izbjeći na dva načina:

- Ograničiti opseg podataka tako da uvijek $N > 0$
- Instalirati detektor greške

Prvu opciju već možete sami izvesti a o drugoj će se još govoriti.



Slika 5-4.

6. Postupanje s greškom i CASE petlja

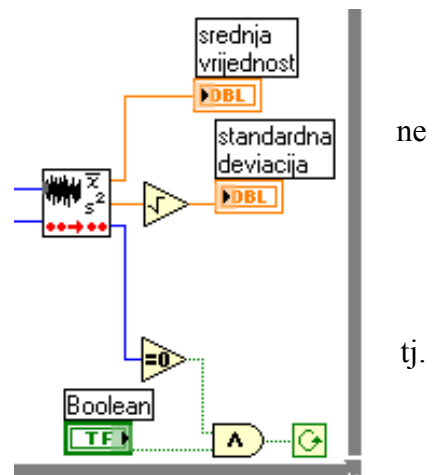
6.1. Rukovanje s greškom

Prije no što se pokaže kako se spaja indikator greške u sustav važno je shvatiti da on treba nešto učiniti s greškom da se ona ne ponovi. U ovom će se slučaju pronaći greška da je dimenzija polja $N < 1$ jer ukoliko ne se poveća dimenzija polja ili ukoliko se ne prekine izvršavanje programa greška će se ponavljati ponovno i ponovno.... Zbog toga je uvijek vrlo važno razmisliti što učiniti s greškom kao i dati obavijest o nastaloj grešci. Ukoliko želimo prekinuti program mora se osigurati da se izvješće o grešci dogodi prije nego što se program zaustavi mora se osigurati tijekom podataka kroz indikator greške.

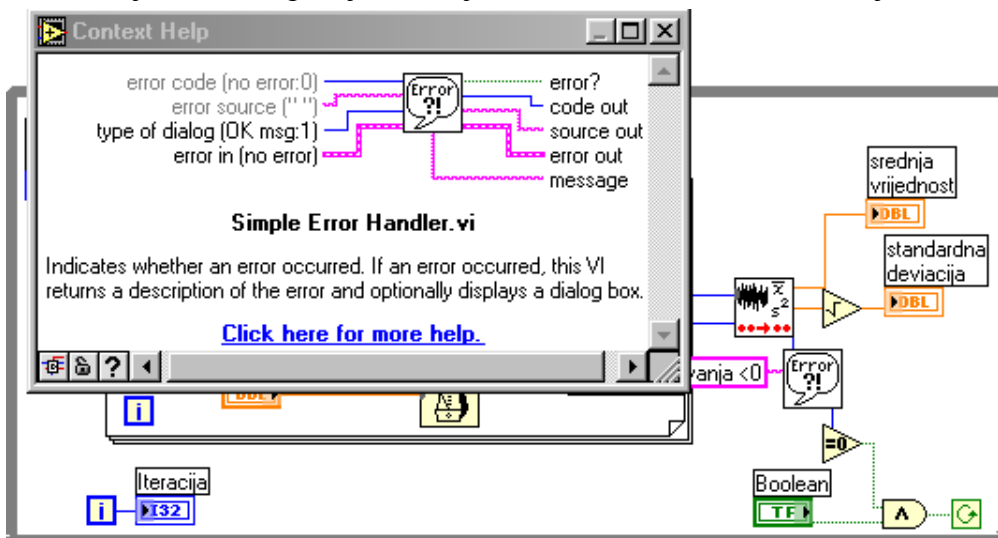
Upravo stoga na bloku **standard deviation** postoji izlaz greške (**error output**). To je cjelobrojni broj čija je vrijednost 0 dok se program odvija bez greške odnosno različita od nule ukoliko se pojavi greška. Da bi se program zaustavio ako je na izlazu greške vrijednost različita od nule potrebno je taj izlaz spojiti na terminal ta zaustavljane WHILE petlje preko logičkog operatora **Equal To 0 ?** (čiji je izlaz u stanju **true** dok mu je ulaz brojčano jednak 0 a **false** u svim ostalim slučajevima). Ili drukčije rečeno: program se odvija tako dugo dok je sklopka za pokretanje programa u položaju uključeno (true) i izlaz greške je jednak nuli.

Na taj će se način sigurno zaustaviti VI ukoliko izlaz greške postane različit od nule ali nećete znati odnosno bolje rečeno korisnik VI neće znati zbog čega se program zaustavio i o kojoj se vrsti greške radi. Ono što trebate je indikator koji će reći o kojoj se grešci radi. Ukoliko se postavi obični numerički indikator on će pokazati brojčanu oznaku greške (točno - 20003) ali taj broj nam ne govori što je greška. Želim da se greška objasni u potpunosti. To je posao jednostavnog indikator greške (**simple error handler**) koji se može naći pod **Functions>>All Functions>>Time & Dialog** (u donjem lijevom kutu). Nakon što ste ga smjestili na blok dijagram isti bi trebao izgledati kao na slici.

Ta je funkcija dosta složena što se može vidjeti kad se uključi jednostavna pomoć (**Ctrl-H**). Srećom jedina dva priključka koja nam u ovom trenutku trebaju su ulaz "kod



Slika 6-1.



Slika 6-2.

greške” (“**error code**” input) i “kod izlaz” “**code out**”). Jednostavni indikator greške treba tako postaviti u blok shemu da kog greške prođe njime prije nego što je VI zaustavljen. Da bi se osigurao pravilan tijek podataka potrebno je postaviti miš na žicu koja izlazi iz izlaza greške bloka **standard deviation** i nakon što se pojavi alat za pozicioniranje otvorite škljocom na desnu tipku miša padajući izbornik i odaberite **Insert** te umetnite **simple error handler**.

Sad nakon što ste pokrenuli VI pokušajte upotrijebiti broj 0 kao broj elemenata čiju srednju vrijednost želite odrediti i pogledajte što se je dogodilo. Pojavit će se lijep prozor greške koji će vam priopćiti “Error -20003 has occurred at an unknown location” i “Probable cause: The number of samples for analysis is less than one”.

Završni akt uređivanja VI je dodavanje tekstualne poruke koja govori gdje je nastala greška. U nekom VI mogu postojati mnogi izvori gdje neka greška može nastati i ono što želite znati je to o kojoj se grešci radi i na kojem je mjestu nastala. Stvar se može urediti tako da se doda “string” konstanta na ulazu izvor greške (**error source**) i upiše odgovarajući tekst o tome na kojem je mjestu u VI nastupila pogreška te će se taj tekst pojaviti u prozoru greške umjesto sentence . Uočite da je boja žice kojom je “string” konstanta spojena na **simple error handler** potpuno nova. Do sad se takova boja nije pojavljivala za žice.

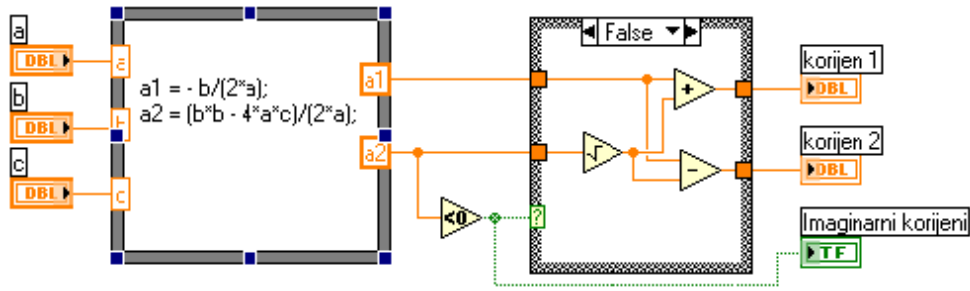
6.1.1. Stvaranje odluka u LabVIEWu (CASE petlja)

Često se u programima moraju obavljati različite operacije ovisno o odluci. Takova odluka može biti ostvarena voljom operatera (ručno) koji koristi program ili automatski ukoliko se ostvare određeni uvjeti u programu. Izbor vrste matematičke operacije (zbroy, razlika, množidba ili djelidba npr.) nad dvije varijable ili odbacivanje izvršenja operacije drugog korijena ukoliko je broj pad korijenom negativan je primjer takvog odlučivanja. U drugom bi slučaju na taj način mogli izbjeći grešku koja bi se javila ukoliko bi broj bio negativan.

Za rješavanje takovih slučajeva u LabVIEWu postoji CASE petlja. Ta se petlja sastoji od određenog broja preklopnih okvira koji svaki na određeni način korespondiraju s ulaznim varijablama. Najjednostavnija je CASE petlja **true/false** gdje se odluka može ostvariti booleovim varijablama. Kompleksnije CASE petlje zahtijevaju više stanja i upotrebu izbornika događaja.

Svaki pojedini okvir unutar CASE petlja mora imati isti broj ulaza i izlaza. Izlazi moraju biti povezani sa svakim pojedinim okvirom CASE petlje dok ulazi mogu biti povezani sa svakim okvirom iako neki ulazi pojedinog okvira mogu biti ignorirani.

Može se konstruirati jednostavna CASE petlja ukoliko se žele naći korijeni kvadratne jednadžbe s koeficijentima **a**, **b** i **c**. Osim toga treba se pojaviti obavijest u slučaju da su korijeni imaginarni. Takav se VI može realizirati na više načina. Na ovim je slikama prikazan moj prijedlog takvog VIa. Vi ga trebate sami napraviti i eksperimentirati njime za različite vrijednosti koeficijenta **a**, **b** i **c**.



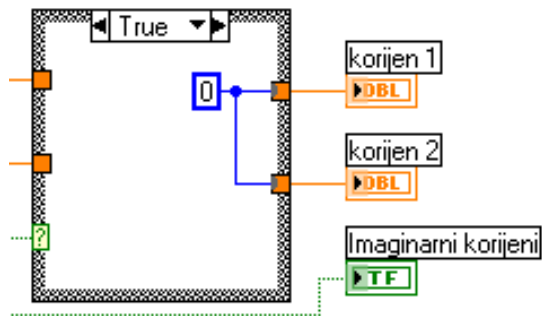
Opaska: Case petlja se nalazi u All

Functions>>Structures. Opcija **True/False** se mijenja strjelicama uz okvir. Brojčani se indikatori i buleov indikator realiziraju tako da se iz **All Functions>>Numeric** i **All Functions>>Boolean** odaberu odgovarajuće konstante i nakon toga promijene u indikatore škljocom desnim mišem na ikonu konstante i izborom **Change To Indicator**.

Osim toga na prednjoj ploči treba Booleov indikator promijeniti da ima izgled led diode koja svijetli zeleno ako su korijeni realni a promijeni biju u crveno ukoliko su korijeni imaginarni.

Neka u slučaju imaginarnih korijena kvadratne jednadžbe oba indikatora pokazuju 0. tome služi drugi okvir prikazan na slici.

Vidi se da svi postojeći izlaz s prethodnog okvira moraju biti spojeni na konstantu 0 dok ulazi mogu biti zanemareni.



Slika 6-4.

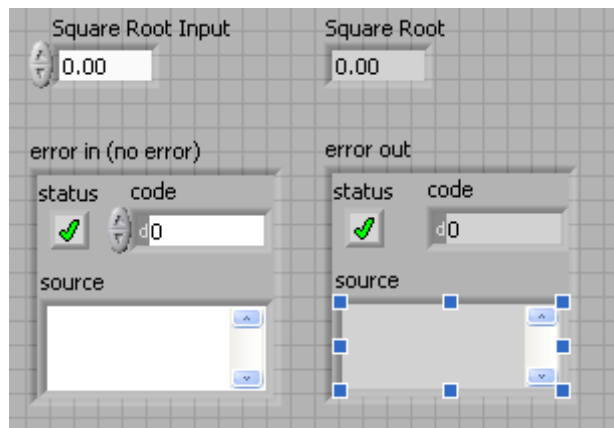
Za vježbu možete probati slijedeći jednostavno primjer koji računa kvadratni korijen broja a u slučaju broj negativog pojaviti će se kod greške

Prednja ploča

Otvorite novi VI i izradite slijedeću prednju ploču koristeći slijedeći postupak.

Napravite brojčanu kontrolu i nazovite je **Ulaz kvadratnog korijena** i napravite brojčani indikator za **Kvadratni korijen**.

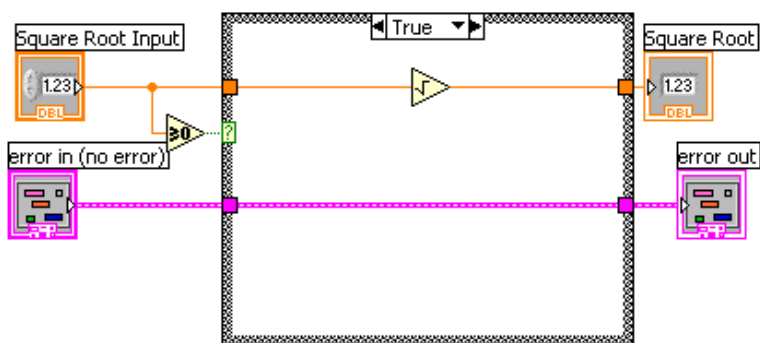
Stavite **Error In 3D.ctl** iz **Controls »All Controls»Arrays & Clusters**.



Slika 6-5.

Blok dijagram

Izradite sljedeći blok dijagram.



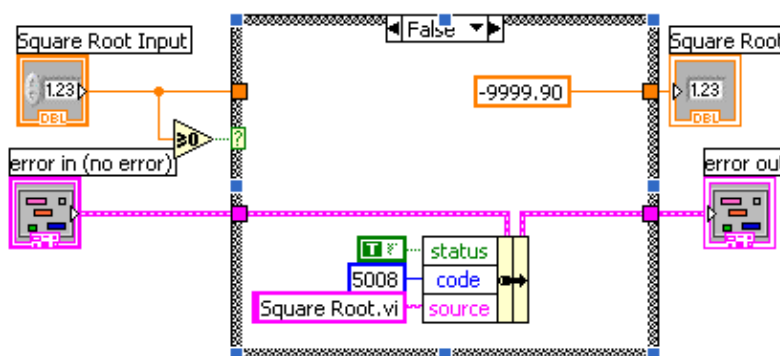
Slika 6-5.

Stavite Case strukturu iz palete **All Functions»Execution Control**.

Stavite **Greater or Equal to 0?** from the **Functions»Arithmetic and Comparison»Comparison** i ožičite je s uvjetnim priključkom CASE strukture.

Stavite u **True Case**:

- funkciju kvadratnog korijena iz palete **Functions»Arithmetic and Comparison»Numeric**.



Slika 6-5.

Stavite u **False Case**:

- stavite broječanu konstantu i upišite -9999.90.
- Stavite **Bundle By Name** iz **Functions »All Functions» Arrays & Clusters**.
- Ožičite s **Error in** do središnjeg priključka **Bundle by Name**, upišite constants. Ožičite s **Error Out** indikatora do izlaza **Bundle by Name**.

Spremite VI kao Square Root.vi.

Otvorite prednju ploču i pokrenite VI.

Spremite i zatvorite VI.

6.2. CASE petlja s višestrukim izborom

Cilj Konstruirati VI koji izvršava osnovne računске operacija (zbroy, razlika, množidba i djelidba) s dva broja. Osim kontrola za promjenu vrijednosti brojeva treba napraviti i izbornik vrste operacije.

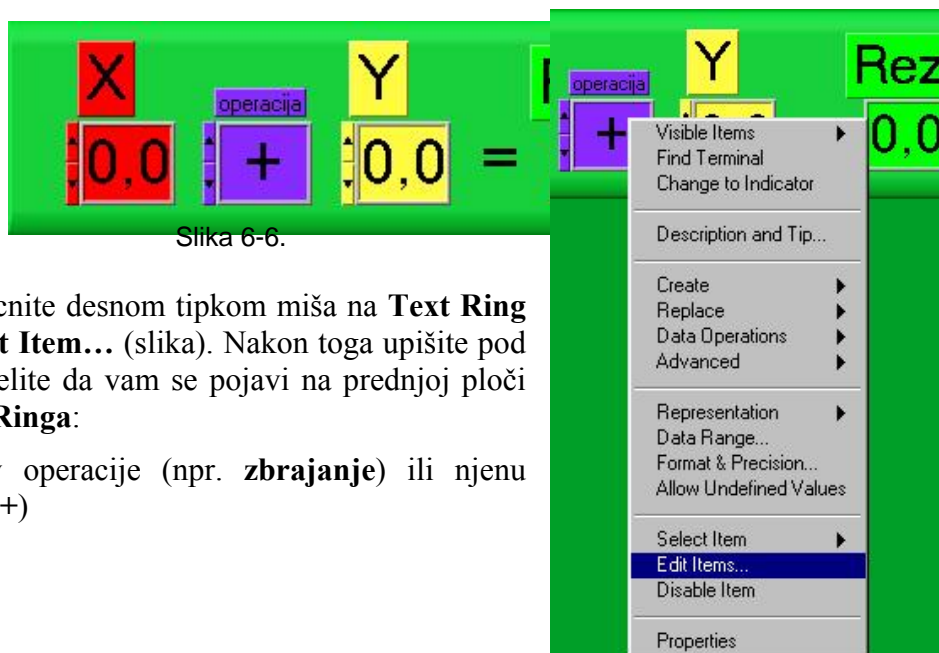
Prednja ploča

Zgodno bi bilo kreirati prednju ploču VI tako da njezin izgled odgovara uobičajenom izgledu računskih operacija tipa:

$$X \text{ [operacije (+ ; - ; * ; /)] } Y = \text{Rezultat}$$

Za realizaciju takove prednje ploče potrebne su dvije brojčane kontrole iz **Controls>>Num Ctrls>>Num Ctrl**, indikator iz **Controls>>Num Inds>>Num Ind** i kontrola za odabir vrste operacije koja može biti prikazana nazivom ili simbolom operacije. U svakom slučaju je potreban prstenasti tekstualni izbornik (**Text Ring**) iz **Controls>>Text Ctrls**. Nakon primijenjene kozmetike prednja ploča može poprimiti izgled kao na slici:

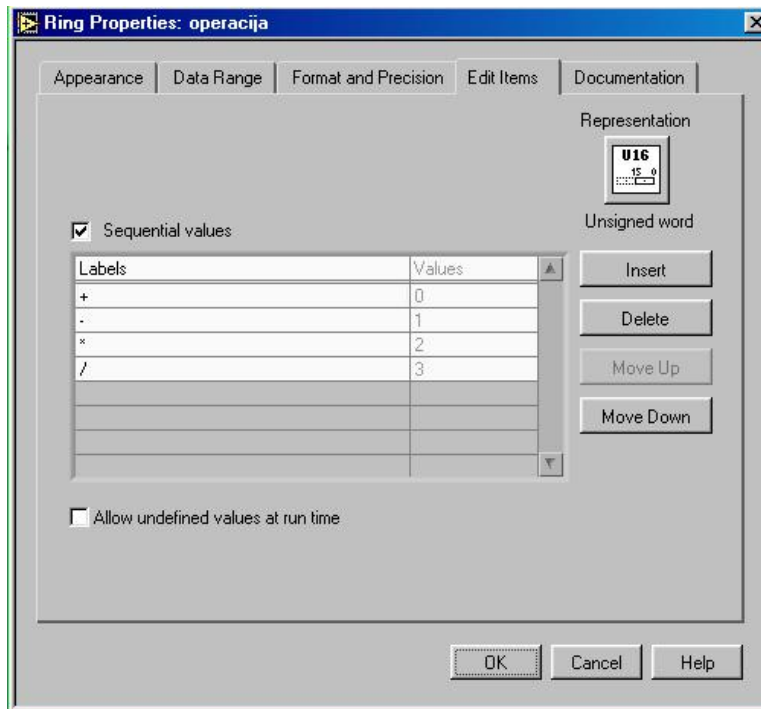
Da bi se mogle birati sve četiri osnovne matematičke operacije potrebno je te četiri opcije upisati u **Text Ring**.



Škljocnite desnom tipkom miša na **Text Ring** izaberite **Edit Item...** (slika). Nakon toga upišite pod **Labels** što želite da vam se pojavi na prednjoj ploči unutar **Text Ringa**:

Naziv operacije (npr. **zbrajanje**) ili njenu oznaku (npr. +)

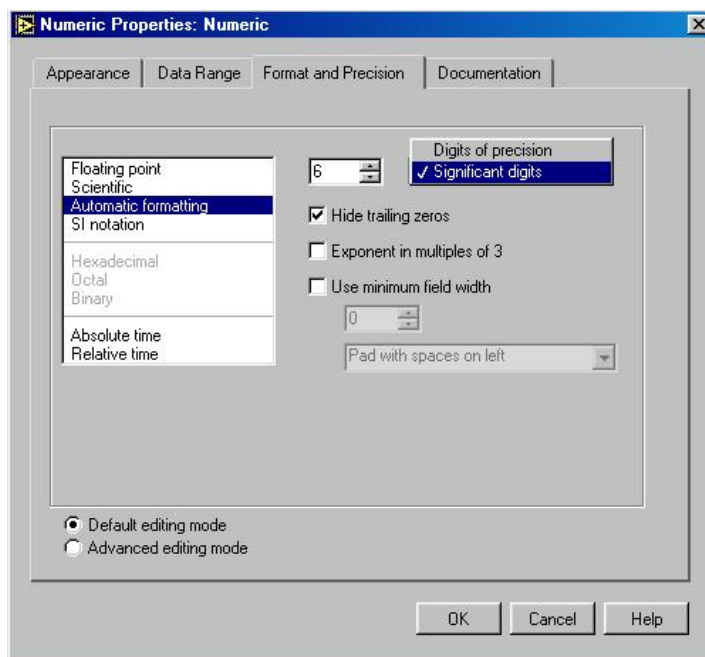
Slika 6-7.



Slika 6-8.

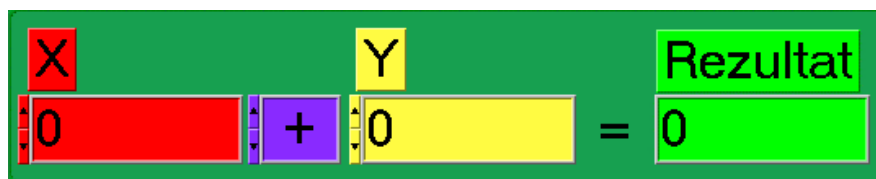
U ovom je primjeru odabrana druga opcija.

Također je zgodno odabrati način prikazivanja upisanih i izračunatih vrijednosti pomoću **Format & Precision...** (odabire se iz padajućeg izbornika nakon škljoca desnom tipkom miša na svaku pojedinačnu broјčanu kontrolu i indikator). Postavljene su vrijednosti prikazane na slici (**Automatic Formatting**, **6 Significant Digits** i **Hide trailing zeroes**) što znači da će prikaz broja biti ili decimalan ili znanstveni ovisno o pogodnosti prikaza da će se prikazati uvijek barem 6 značajnih znamenaka i da se preostale nule na kraju broja neće prikazati. Ukoliko želimo drugačiji način prikaza brojeva tako da se uvijek prikazuju decimalni brojevi s preciznošću od 6 znamenaka onda treba odabrati **Floating point** i **Digits of precision**.



Slika 6-9.

Uz takav odabir **Format & Precision...** i rearanžirane dimenzije brožanih komanda i indikatora prednja ploča poprima slijedeći izgled:



Slika 6-10.

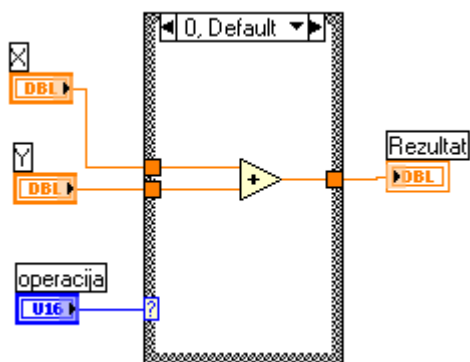
Blok dijagram

Pređite sad na blok dijagram da bi kreirali kod za provedbu operacija za traženi VI. Ono što trenutno postoji na blok dijagramu su ikone dviju ulaznih kontrola **X** i **Y**, ikona indikatora **Rezultat** te ikona **Text Ringa**.

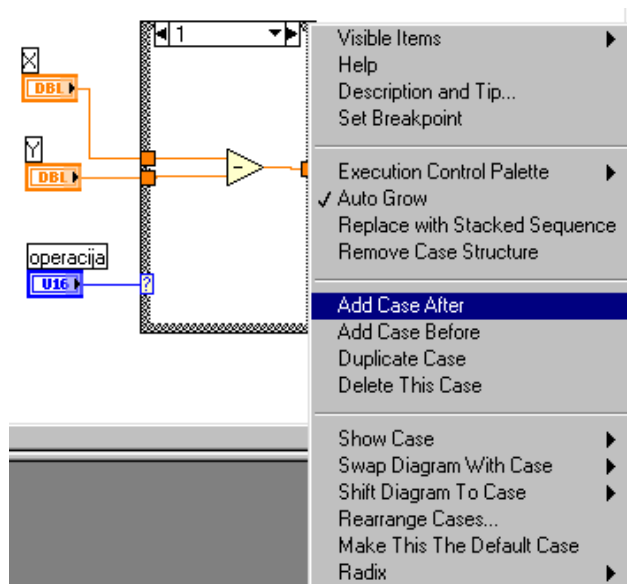
Da bi mogli birati različite osnovna matematičke operacija kao što je uneseno pod **Edit Items...** u **Text Ringu** potrebno je na blok dijagram donijeti CASE petlju koja inicijalno ima samo dva slučaja (**0, true** i **1, false**).

Budući da su u **Ring Properties** unesene matematičke operacije slijedećom redoslijedom + ; - ; * ; / uz koje su vrijednosti 0...3 potrebno je u istobrojni slučaj postaviti pripadajuću operaciju.

Dovedite operacije zbrajanja (Add) i oduzimanja (-) i smjestite ih u CASE petlju 0 i 1 respektivno. Povežite najprije ulazne kontrole s ulazima matematičkih operacija a indikator s njihovim izlazom pazeći da povezivanje teče kroz nastale tunele. Osim toga povežite Text Ring s okvirom CASE petlje tako da se dobije kod na blok dijagramu prikazan na slikama:

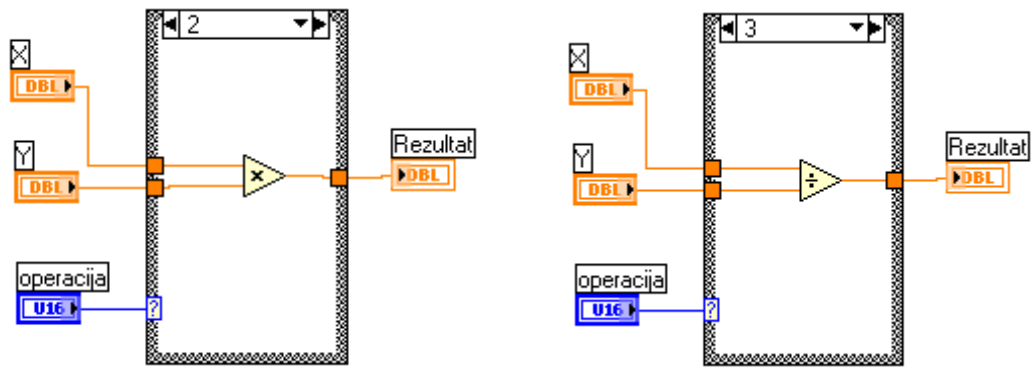


Slika 6-11.



Slika 6-12.

Da bi se otvorila još dva slučaja unutar CASE petlje za operacije množenja i djeljenja potrebno je pri otvorenom slučaju 1 (operacija dijeljenja) škljocnuti desnom tipkom miša na ivicu CASE petlje i iz padajućeg izbornika kao što je prikazano na slici odabrati **Add Case After**. Zatim unijeti operator množenja te povezati ulaze i izlaze s odgovarajućim tunnelima tako da se dobiju kodovi prikazani na slikama.



ulator,

Slika 6-13.

Vratite se na prednju ploču neprekidno izvršavanje i eksperimentirajte s broječanim kontrolama i izbornikom operacija u text ringu.

S
pre
ite
VI
pod
imen
om
Moj
Kalk

7. Kreiranje podVIa

7.1. Razumijevanje hijerarhije

Osnovno za izradu LabVIEW aplikacija je razumijevanje i korištenje hijerarhijske prirode VIa. Nakon izrade VI, dizajner ga može koristiti kao podVI u blok dijagramu VIa više razine. Zbog toga je podVI analogan podrutini u C jeziku. Isto tako kao što nema ograničenja u broju podrutina koje se mogu koristiti u C programu tako isto nema ograničenja u broju podVIa koje se mogu koristiti u LabVIEW programu. Dizajner može također pozivati podVI unutar drugog podVIa.

Kada se tvori aplikacija, počinje se na najvišoj razini i definiraju ulazi i izlazi za aplikaciju. Nakon toga, konstruiraju se podVIevi koji izvršavaju neophodne operacije nad podacima kako one teku blok dijagramom. Ako blok dijagram ima veliki broj ikona, potrebno ih je grupirati u VIe niže razine, da bi se održala jednostavnost blok dijagrama. Ovaj modularni pristup čini aplikaciju jednostavnom za provjeru programa (debug), razumjevanje i održavanje.

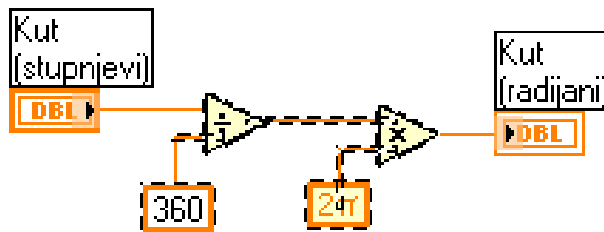
7.1.1. Tvorba podVIa

CILJ Kreirati ikonu i konektor za *St_U_Rad VI* koji pretvara kutne stupnjeve u kutne radijane te koristiti taj VI kao podVI.

Da bi se koristila VI kao podVI, mora se kreirati ikona da bi prikazivala taj VI u blok dijagramu VI kao i konektorski okvir na koji se mogu povezati ulazi i izlazi.

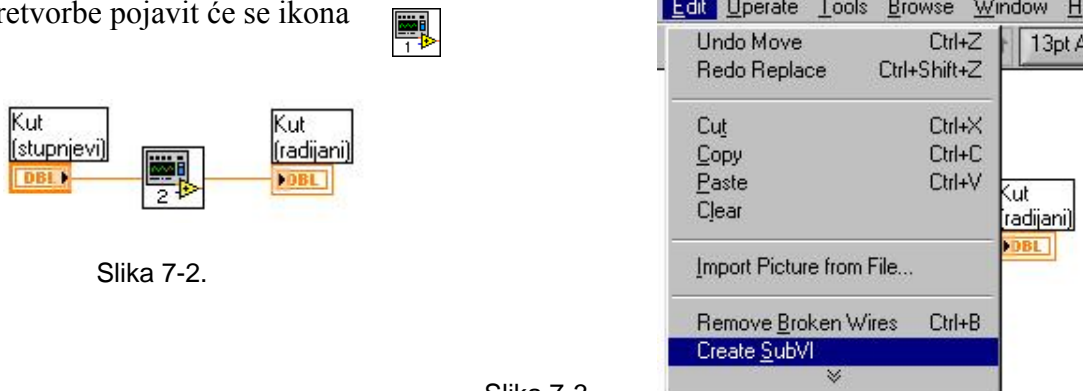
IKONA Kreirati ikonu koja predstavlja VI u blok dijagramu druge VI. Ikona može biti slikovna predstava namjene VI ili pak može biti tekstualni opis VI ili njegovih terminala.

Otvorite novi prazni VI i realizirajte blok dijagram prikazan na slici kojim se pretvaraju kutovi u stupnevima u kutove u radijanima po formuli (kodu).



Slika 7-1.

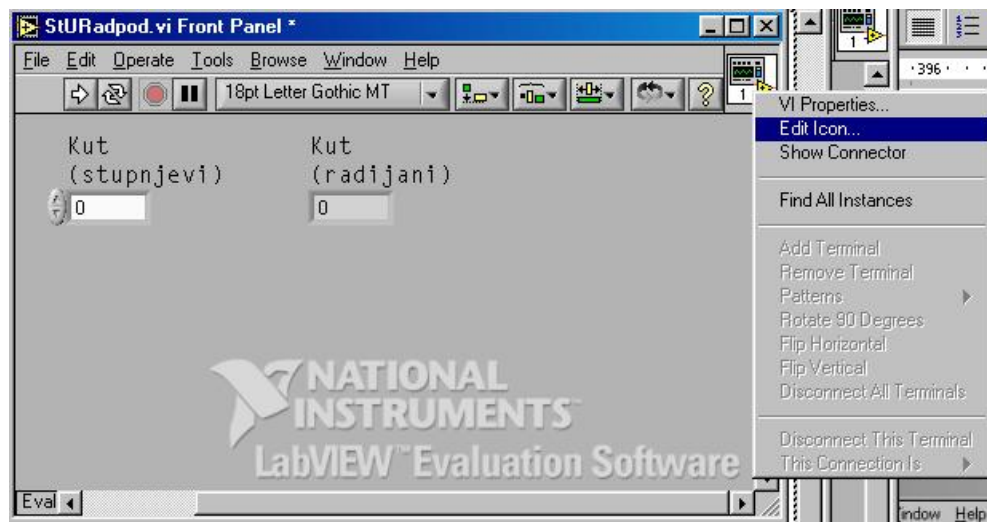
Da bi se podVI kreirao automatski odaberite kod pretvorbe (crtkani dio na slici) potražite **Edit>>Create SubVI...** . Umjesto odabranog koda pretvorbe pojaviti će se ikona



Slika 7-2.

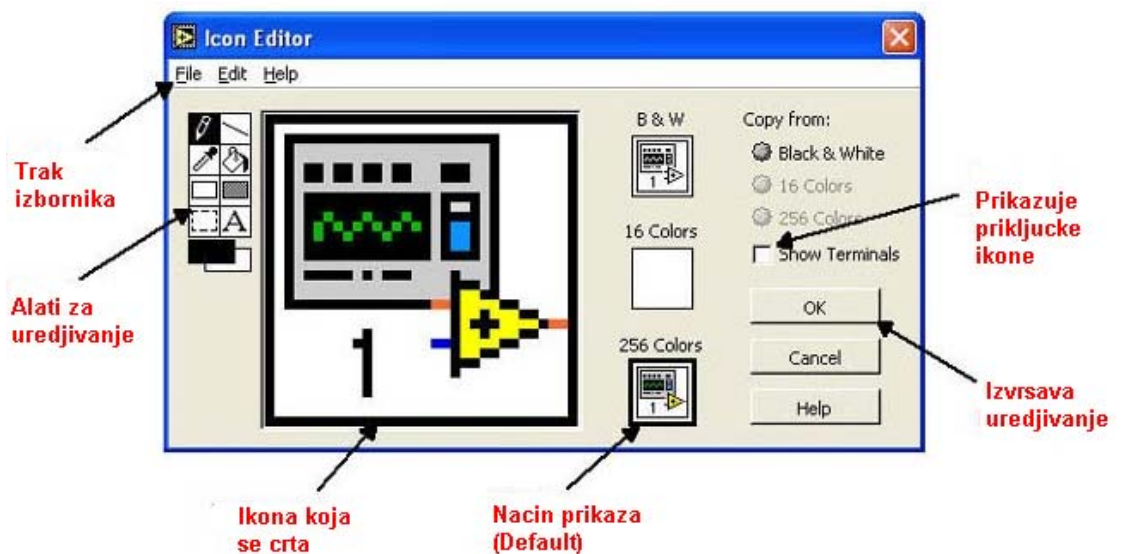
Slika 7-3.

Škljocom desnom tipkom miša na površini ikone u gornjem desnom pozvati Icon Editor uglu prednje ploče birajući Edit Icon. Prečac je moguć tako da se dvostruko škljocne na površinu ikone da bi se ona uredila.



Slika 7-4.

Alati za uređivanje ikona prikazani s lijeva na desno odozgo prema dolje:



Slika 7-5.

- Crta i briše piksel po piksel.
- Crta ravnu liniju. Pritisnuti <Shift> a zatim vucite ovaj alat da bi se nacrtala horizontalna, vertikalna i dijagonalna linija.
- Kopira boju prednjeg plana iz nekog elementa u ikonu.
- Puni označenu zonu bojom prednjeg plana.
- Crta pravokutne granice u boji prednjeg plana (pp). Dvput škljocnuti na ovaj alat da se uokviri ikona u boji prednjeg plana (pp).

- Crta kvadratični okvir obrubljen bojom pp i ispunjen sa bojom zadnjeg plana (zp). Ako se dvaputa škljocne, uokvirava se ikona u pp boju i puni sa zp bojom.
- Odabire zonu oko ikone da bi se mogla kretati, klonirati (kopirati), ili drugo.
- Unosi tekst u dizajn ikone.
- Prikazuje trenutne boje pp i zp. Škljocnuti na svaku da se dobije paleta boja iz koje se onda može izabrati nova boja ako se želi promjeniti.

Gumbi desno od ekrana za editiranje vrše slijedeće funkcije;

OK Pohranjuje crtež kao VI ikonu i vraća se na prednju ploču.

Cancel Vraća se na prednju ploču bez pohranjivanja promjena

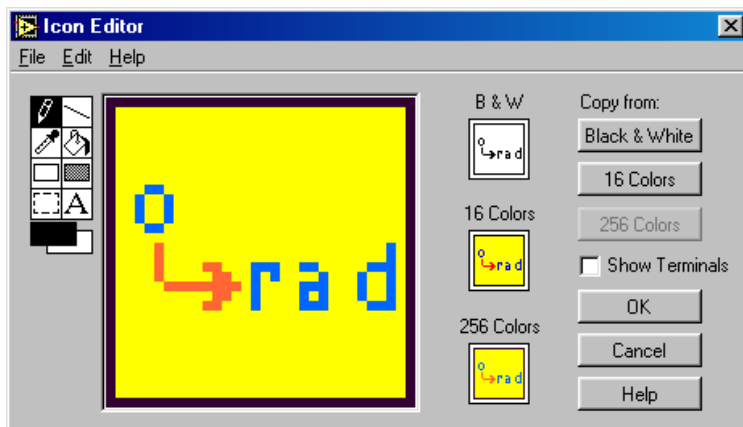
Prije no što nacrtate vlastitu ikonu potrebno je pretpostavljenu ikonu izbrisati.

- Odabirati **256 Colors** jer je jedino moguće u tom okviru kreirati novu ikonu.
- Alat za odabiranje uokviriti pretpostavljenu ikonu (probajte ne odabrati pri tom crni vanjski obrub ikone). Pritisnuti **<Delete>** na tipkovnici.

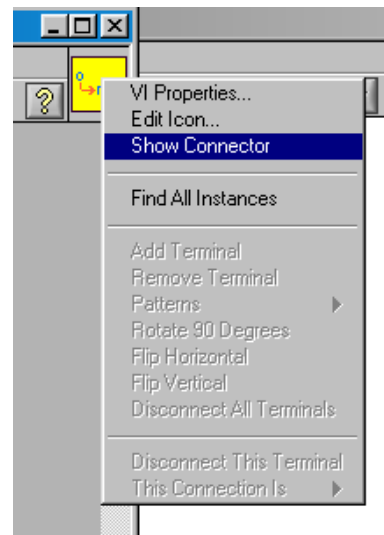
Nakon toga treba nacrtati termometar alatom olovke. Kreirati tekst sa alatom za pisanje. Da bi promijenili vrst slova dvaput škljocnite na alat za pisanje

Eksperimentirati s editorom da bi se naučili alati.

Vaša bi ikona trebala sada izgledati otprilike kao na slijedećoj slici:



Slika 7-6.



Zatvoriti editor ikone škljocnuvši na OK kada ste završili s crtanjem. Nova ikona će se pojaviti na polju ikone (pane) u gornjem desnom uglu prednje ploče.

7.1.2. Konektor

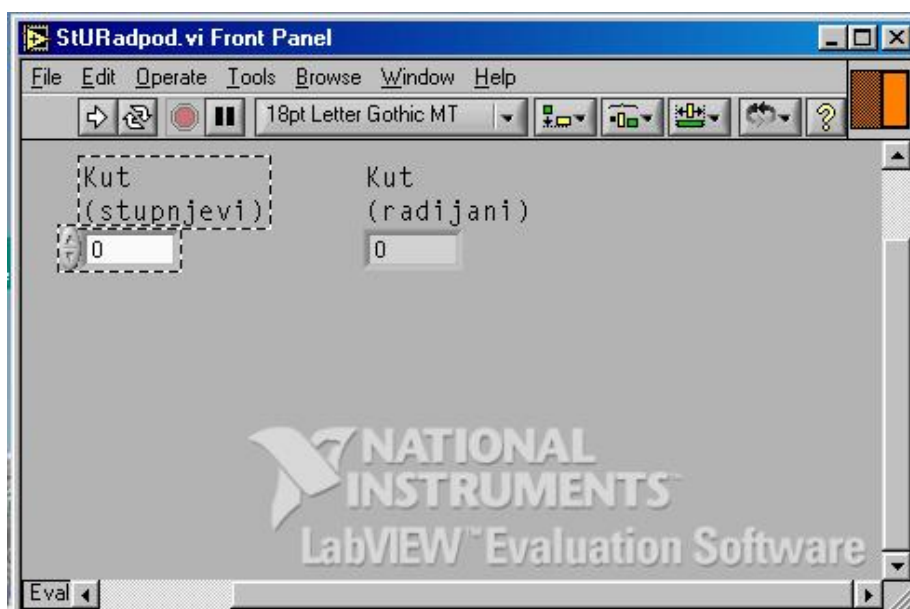
Sada se može kreirati konektor.

Definirati konektorski terminal škljocnuvši na polje ikone (pane) na prednjoj ploči birajući **Show Connector**, kako to pokazuje slijedeća slika.

Budući da LabVIEW izabire oblik terminala baziran na broju kontrolnih elemenata i indikatora na prednjoj ploči, u ovom slučaju postoji samo jedan terminal – indikator pretvornika stupnjeva u radijane.

Doznačite terminal za pretvornik kutnih stupnjeva u kutne radijane.

- Škljocnite na ulazni terminal u konektoru. Kursor se automatski mjenja u alat za ožičenje a terminal pocrni.
- Škljocnite na ulazni brojčanu kontrolu **Kut (stupnjevi)**. Pojavit će se oko te kontrole crtkana linija kako to pokazuje slijedeća slika:
- Ponovite isto i za izlazni terminal



Slika 7-7.

Ako se škljocne na otvorenu površinu prednjeg plana, crtkana linija isčezava i odabrani terminal posivi pokazujući na taj način da ste doznačili indikator terminalu. Ako je terminal bijel niste napravili spoj korektno. Ponovite prethodne korake ako je neophodno.

Pohraniti VI birajući **File>>Save**. VI je sada kompletan i spreman za korištenje kao podVI u drugim VIa. Ikona predstavlja VI u blok dijagramu pozivajući VI. konektor (sa jednim terminalom) daje na izlazu.

Opaska: Konektor specificira ulaze i izlaze u VI kada ga koristite kao podVI. Potrebno je imati na umu da kontrolni elementi prednjeg panela mogu biti koršćeni samo kao ulazi: indikatori prednjeg panela mogu biti samo kao izlazi.

Zatvoriti VI birajući **File>>Close**.

7.2. Korištenje VI kao podVI

Moguće je koristiti bilo koji VI koja ima ikonu i konektor kao podVI u drugoj VI. Izbor VIa koji će se koristiti kao podVI izvršite u blok dijagramu izabравši ih u **All Functions>Select a VI**. Izabравši ovu opciju pojavit će se file dijalog okvir iz kojeg možete izabrati bilo koji VI u sustavu. Ako otvorite VI koji nema ikonu i konektor pojavljuje se bijeli prazan kvadrat u blok dijagramu pozivajućeg VI. Nije moguće ožičiti ovakav čvor.

PodVI je analogan podprogramu. PodVI čvor (ikona/konektor) je analogan pozivu podprograma. PodVI čvor ne predstavlja sam podVI, kao što nije ni **call statement** u podprogramu (instrukcija poziva) nije sam podprogram. Blok dijagram koji sadrži nekoliko identičnih podVI čvorova poziva istu podVI nekoliko puta.

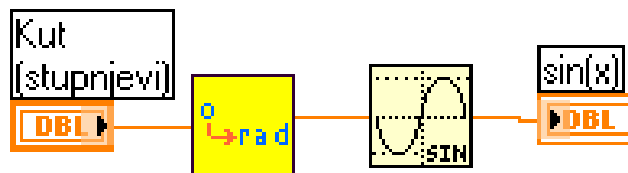
CILJ Izgraditi VI koji će koristiti kao podVI.

St_U_Rad VI koji je dizajniran daje kutne radijane kao varijablu **x** koja može biti ulazna varijabla bilo koje trigonometrijske funkcije npr. **sin (x)**.

Blok dijagram

U ovom ćemo slučaju nakon otvaranja novog VI s **File>>New** ili škljocanjem na **New VI** dijalog okviru odmah prijeći na blok dijagram i u njega odmah smjestiti **St_U_Rad** kao podVI.

Na ulaznom terminalu podVIa kreirati kontrolu čije ime odmah odgovara željenome tj. **Kut (stupnjevi)**. Iz **All Functions >>Functions >> Numeric>> Trigonometric** prenijeti funkciju **sin (x)**. Kreirati numerički indikator na izlazu funkcije **sin (x)** te ožičiti blok dijagram da izgleda kao na slici:



Slika 7-8.

Svaki put kada se kreira novi kontrolni element ili indikator, LabVIEW automatski kreira odgovarajući terminal u blok dijagramu. Simboli terminala sugeriraju tip podatka kontrolnog ili indikacijskog elementa. Na primjer DBL terminal predstavlja dvostruku preciznost, TF je Booleov terminal, I16 terminal predstavlja regularni 16-bitni integer (cijeli broj); a ABC predstavlja string (niz karaktera).

Opaska: O tipu možete konstante zaključiti po njenoj boji.

- Plave brojčane konstante su integer (cijeli brojevi)
- narandaste konstante su brojevi dvostruke preciznosti.

LabVIEW automatski pretvara brojeve u odgovarajući format kada je to potrebno.

Opaska: Podsjetite se mogućnosti da škljocnete na funkciju i izaberete **Create Constant** da se automatski kreira i ožiči odgovarajuća konstanta s funkcijom.

Opaska: Ukoliko se pojavi izlomljena crta između ikone i terminala to pokazuje da ste doznacili podVI konektorski terminal nekorektno sa indikatorom na prednjem panelu. Nakon modifikacije podVI, možda će biti potrebno izabrati **Relink to podVI** iz pop-up menija ikone. Ako je potrebno izaberite **Edit>>Remove bad Wires**.

Prednja ploča

Izabrati **Windows>>Show Front Panel** dizajnirati je po vlastitom ukusu tako da opseg kutnih stupnjeva bude od 0 do 360 a za vrst prikaza u **Format & Precision...** odabrati **Floating point** i **Digits of precision 6**.

Vratiti se na prednji panel i škljocnuti na pokrenite VI.

8. Polja , klasteri i grafovi

8.1. Polja (Arrays)

Polje se sastoji od skupa elemenata podataka koji su istog tipa. Polje ima jednu ili više dimenzija i do $2^{31} - 1$ elemenata po jednoj dimenziji (ukoliko naravno ima dovoljno memorije). Polja u LabVIEW mogu biti bilo kojeg tipa (osim tipa polje, chart ili graf). Pristupa se svakom elementu polja preko njegovog indeksa. Indeks je u opsegu 0 do n-1, gdje n je broj elemenata u polju. Slijedeće jednodimenzionalno polje brojčanih vrijednosti ilustrira ovu strukturu. Primijetimo da prvi element ima indeks 0, drugi element indeks 1, i tako dalje.

index	0	1	2	3	4	5	6	7	8	9
10-element array	1.2	3.2	8.2	8.0	4.8	5.1	6.0	1.0	2.5	1.7

Slika 8-1.

8.2. Kontrolna polja, konstante i indikatori

Kontrolna polja, konstante i indikatore se kreiraju na prednjoj ploči ili na blok dijagramu kombiniranjem školjke polja (array shell) s brojčanom, Booleovim, stringom ili klasterom (cluster - grozd, svežanj). Već smo rekli da element polja ne može biti drugo polje, chart ili graf.

8.3. Graf

Grafički indikator se sastoji od dvodimenzijskog prikaza jednog ili više polja koji se zovu plots (zapisi). LabVIEW ima tri tipa grafova:

- XY grafovi, grafovi valnog oblika (**waveform graphs**) i grafove intenziteta (**intensity graphs**).

Razlika između grafa i charta je u tome da graf iscrtava podatke kao blok dok chart iscrtava podatke točka po točka ili polje po polje.

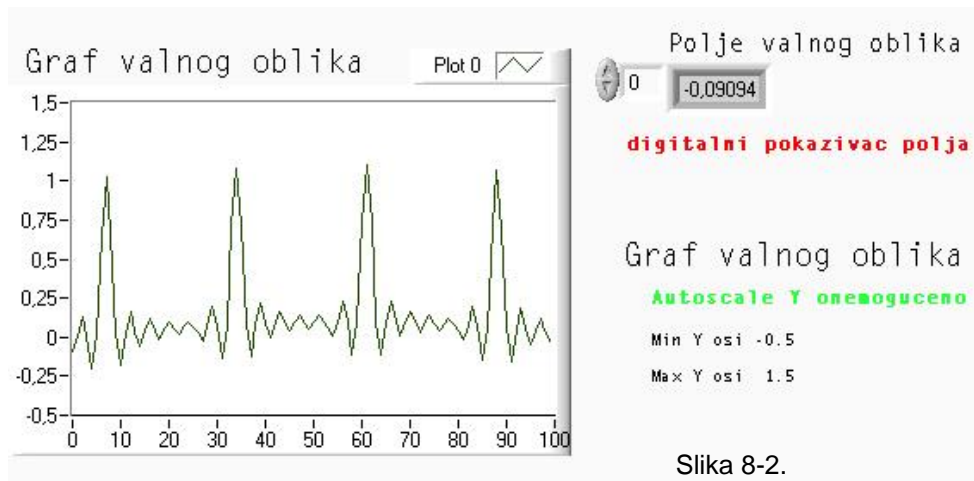
8.3.1. Kreiranje polja sa autoindeksiranjem

CILJ Kreirati polje koristeći autoindeksiranje **FOR** petlje i iscrtati polje podataka u grafu valnog oblika (waveform graph).

Izgradite VI koji generira polje koristeći **Generate Waveform.vi (Program files \National instruments\LabVIEW\activity)** i iscrtava polje u grafu valnog oblika. VI će se tako modificirati da iscrtava višestruke ispise

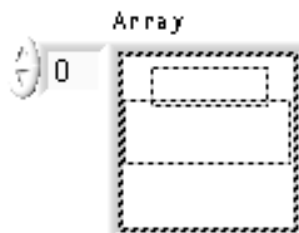
Prednja ploča

Otvorite novu prednju ploču. Postaviti školjku polja iz **Controls>>All Functions>>Array & Cluster>>Array** na prednju ploču.



Slika 8-2.

Nasloviti školjku polja kao Polje valnog oblika. Postaviti digitalni indikator iz **Controls>>Numeric** unutar pokazivača elementa školjke polja, kako to pokazuje slika . Ovaj indikator prikazuje sadržaj polja.



Slika 8-3.

Kao što je već rečeno, graf indikator je dvodimenzijски prikaz jednog ili više polja podataka koji se nazivaju zapisima (plots). LabVIEW ima tri tipa grafova: XY grafovi, grafovi valnog oblika (waveform graphs) i grafovi intenziteta (intensity graphs).

Postaviti **Graph Waveform** iz **Controls>>Graph** na prednju ploču. Naslovi **Graf valnog oblika**.

Graf valnog oblika iscrtava polja s ravnomjerno raspoređenim točkama, kao prikupljeni valni oblici koji se mijenjaju u vremenu.

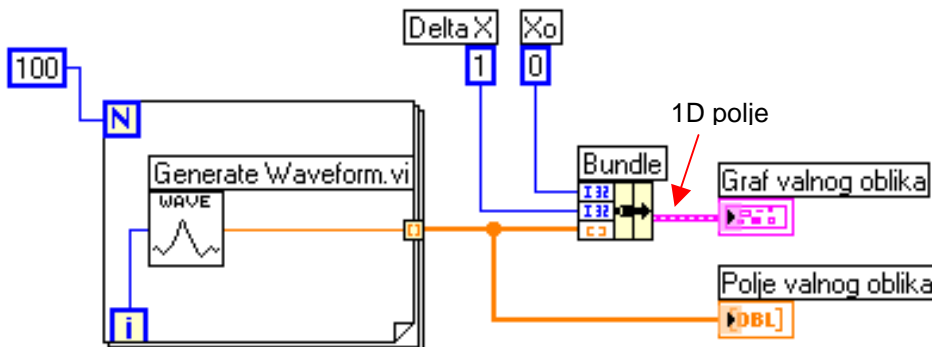
Ako želite možete povećati graf vučenjem ugla kursorom za promjenu veličine.

Pretpostavljeno je grafovi automatski podešavaju vrijednosti ljestvice (**Autoscale**). To jest, oni automatski podešavaju X i Y granične vrijednosti ljestvica da bi prikazali cjelokupan skup ulaznih podataka. Stoga treba onemogućiti automatsko podešavanje Y ljestvice škljocom na graf i isključiti kvačicu uz **Scale>>Autoscale Y**.

Granice Y osi se mogu promijeniti tako da se dvaput škljocne na granice ljestvica kad se pojavi alat za pisanje i unosu nove vrijednosti. Upisati minimum Y osi - 0.5 a maksimum 1.5.

Blok dijagram

Izgraditi blok dijagram prikazan na prethodnoj slici.



Slika 8-4.

VI koji simulira valni oblika (**Generate Waveform**) a nalazi se u **Functions>>Tutorials** davat će po jednu točku valnog oblika. VI zahtjeva skalarni indeksirani ulaz tako da je potrebno ožičiti terminal broja iteracija u konturi sa ovim ulazom. Škljocom na VI izborom **Show>>Label** prikazat će riječ **Generiraj valni oblik** u naslovu. Primijete da žica od **Generiraj valni oblik** VI postaje deblja kako se mijenja u polje na ivici petlje.

FOR petlja automatski akumulira polja na svojoj ivici. Ovo se naziva autoindeksiranje (*auto-indexing*). U ovom slučaju, numerička konstanta ožičena na numerički ulaz za brojač petlje FOR petlje, kreira 100 elementno polje (indeksirano 0 - 99).

Funkcija uvezivanja (Bundle function) (u **Functions>>Cluster**) povezuje komponente isertavanja (plot) u klaster (svežanj - grozd). Potrebno je promijeniti ikonu Bundle funkcije prije nego što ćete je ožičiti na pravilan način. Postaviti alat za pozicioniranje na donji desni ugao ikone. Alat se mijenja u kursor za izmjenu veličine (resizing), pokazan na lijevoj strani.

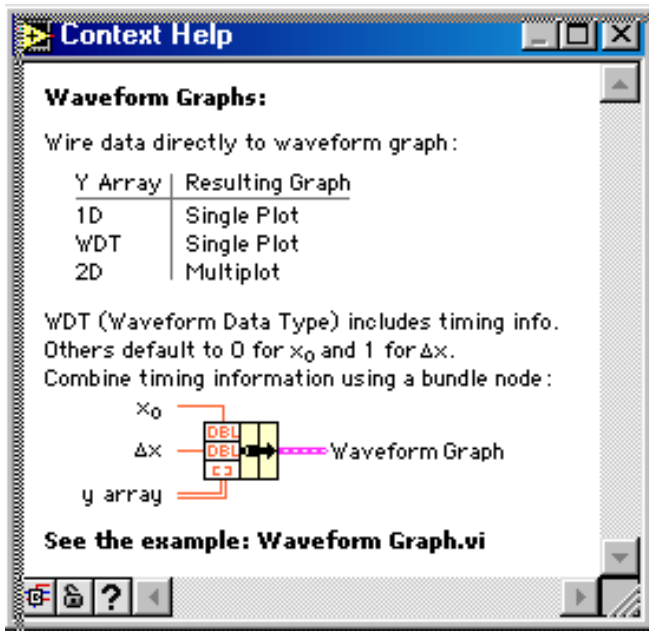
Kada se alat promjeni, škljocnuti i vuci sve dotle dok se ne pojavi i treći terminal. Sada možete nastaviti sa ožičavanjem blok dijagrama kao što je pokazano na prvoj slici ovog dijela.

Klaster može sadržavati elemente podataka različitih tipova. Klaster u blok dijagramu kojeg mi gradimo u ovom primjeru, grupira povezane elemente podataka iz različitih mjesta na dijagramu, reducirajući broj žica i njihovo umršvanje. Kada koristite klaster, podVIi zahtijevaju manje terminala za spajanje. Klaster je inače analogan *struct* u C jeziku. Klaster se može zamisliti kao svežanj žica (slično telefonskom kablju). Svaka žica u kablju bi predstavljala različit element klastera. Komponente uključuju inicijalnu vrijednost za X (0), delta (inkrementnu) vrijednost za X (1), kao i Y polje (podaci valnog oblika, koje osiguravaju brojčane konstante na blok dijagramu). U LabVIEW treba koristiti **Bundle** funkciju da se stvori klaster.

Opaska: Provjeriti da se generiraju oni tipovi podataka koje će prihvatiti grafovi i chartovi. U procesu tvorbe ovog blok dijagrama, provjeriti tipove podataka kroz slijedeće korake:

- Otvoriti prozor pomoći birajući **Help>>Show Help**
- Pomjeriti alat za ožičenje preko terminala grafa.

• Provjeriti informaciju o tipu podatka koja se pojavljuje u prozoru pomoći. Kao primjer pogledati sljedeću sliku:



Slika 8-5.

Tri brojučane konstante postavljaju sljedeće vrijednosti: broj iteracija FOR petlje, početnu X vrijednost i inkrement (delta) X vrijednost. Primijete da se može škljocnuti na terminal brojača FOR petlje, prikazan na lijevoj strani, i izabrati **Create Constant** da automatski dodamo i ožičimo broječanu konstantu za taj terminal. Svaka iteracija FOR petlje generira jednu točku u valnom obliku koju VI pohranjuje u polje za valni oblik automatski kreirano na ivici petlje. Nakon što petlja okonča svoje izvršenje, Bundle funkcija uvezuje početnu vrijednost X (XO), delta vrijednost za X i polje za crtanje grafa.

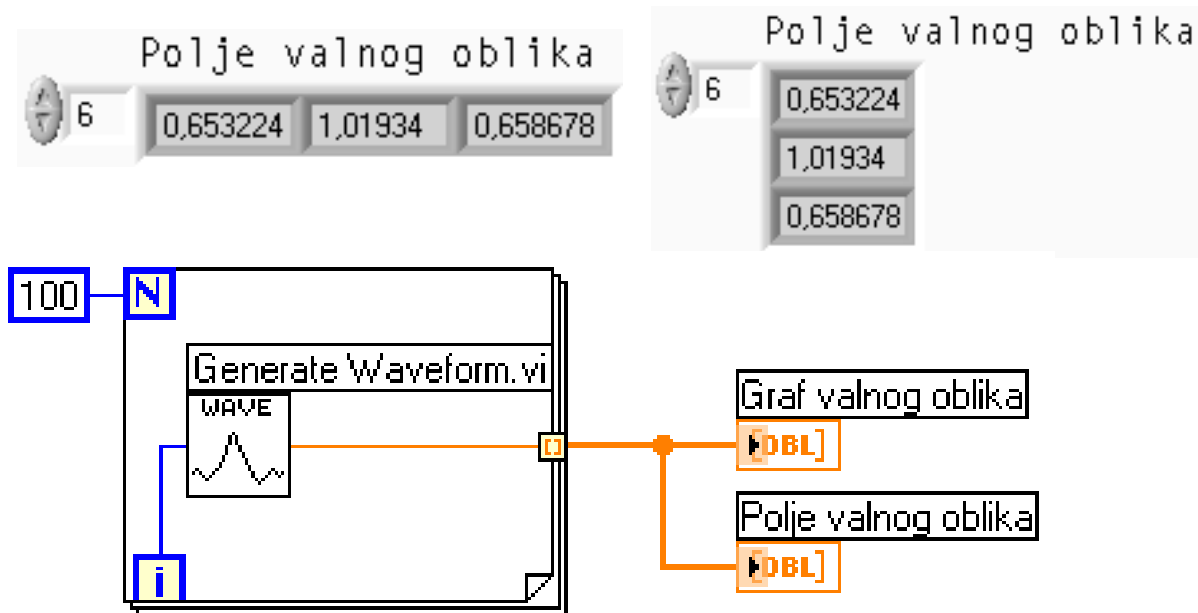
Vratite se na prednju ploču i pokrenite VI. VI iscertava autoindeksirano polje valnog oblika na grafu. Početna vrijednost X je 0 a delta X vrijednost je 1.

Promijenite delta X vrijednost na 0.5 a početnu vrijednost na 20. Ponovno pokrenite VI.

Primijetite da graf sada prikazuje istih 100 točaka podataka sa početnom vrijednošću od 20 s delta X = 0.5 za svaku točku (pogledati X os). U testu sa vremenskom kontrolom, ovaj graf će korespondirati sa 50 sekundi podataka, počevši sa 20 sekundi. Eksperimentirajte sa nekoliko kombinacija za početne i delta X vrijednosti.

Možete gledati bilo koji element u polju unoseći indeks tog elementa u indeksni pokazivač. Ako unesete broj veći nego što je veličina polja, pokazivač blijedi (dim) pokazujući na taj način da niste definirali vrijednost za taj indeks.

Ako želite da gledate više od jednog podatka, možete povećati indikator polja. Postaviti alat za pozicioniranje na donji desni ugao polja. Alat se mijenja u cursor za promjenu veličine (resizing), pokazan na lijevoj strani. Kada se alat promjeni, vucite na desno ili dolje. Polje sada prikazuje nekoliko elemenata u rastućem redoslijedu indeksa, počevši s elementom koji korespondira određenom indeksu kako to pokazuje sljedeća slika:



Slika 8-6.

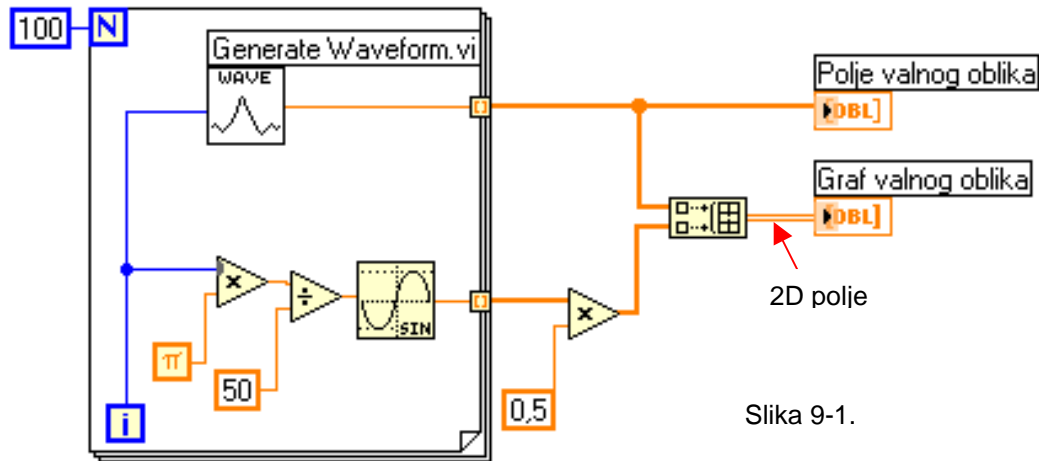
U prethodnom blok dijagramu, specificirali ste početnu vrijednost X i delta X valnog oblika. Često je pak, početna vrijednost X nula a delta X vrijednost 1. U ovim situacijama možete ožičiti polje valnog oblika direktno sa terminalom grafa valnog oblika, kako je to vidljivo na sljedećoj slici:

Vratite se ponovno na blok dijagram. Obrisati **Bundle** funkciju i brojčane konstante ožičene sa njom. Da bi se obrisala funkcija i konstante, izabrati funkciju i konstante sa alatom za pozicioniranje a onda pritisnuti <Delete>. Izabrati **Ctrl-B**. Završiti ožičenje blok dijagrama kako je to pokazano na prethodnoj slici.

Pokrenuti VI. VI sad iscrtava valni oblik sa početnom vrijednoscu $X = 0$ i delta $X = 1$.

9. Višestruki grafovi

Višestruki grafovi valnog oblika možete kreirati gradeći VI polje od tipa podataka koji se normalno šalju u jednostruki graf.



Slika 9-1.

Nastaviti graditi blok dijagram kako je pokazano na dijagramu. Funkcija sinusa nalazi se u (**Functions >> All Functions >> Numeric >> Trigonometric**). U ovom se primjeru koristi ova funkciju unutar FOR petlje da se stvori polje točaka koje predstavljaju jednu periodu sinusoide.

Funkcija gradnje polja (**Build array** u **All Functions >> Array**) koristiti se da kreirate točnu strukturu podataka da bi iscrtali dva polja na grafu valnog oblika koji je u ovom slučaju dvodimenzijnsko polje. Povećati funkciju za gradnju polja (**Build array**) tako da kreirate dva ulaza vučenjem ugla sa alatom za pozicioniranje.

Pi - đ (konstanta (**All Functions >> Numeric >> Additional Numeric Constant**)).

Podsjetimo da možete naći funkcije množenja i dijeljenja (**Multiply & Divide**) u **All Functions >> Numeric**.

Prebacite se na prednju ploču. Pokrenite VI.

Primijetite da se dva valna oblika iscrtavaju na istom grafu valnog oblika. Početna vrijednost X ju na 0 a vrijednost za delta X na 1 za oba dva skupa podataka.

*Opaska: Možete promijeniti izgled ispisa na grafu škljocom na legendu za određeni ispis. Na primjer, možete promijeniti sa crtanog grafa na stupčani graf, izabirući **Common Plots >> Bar Graph**.*

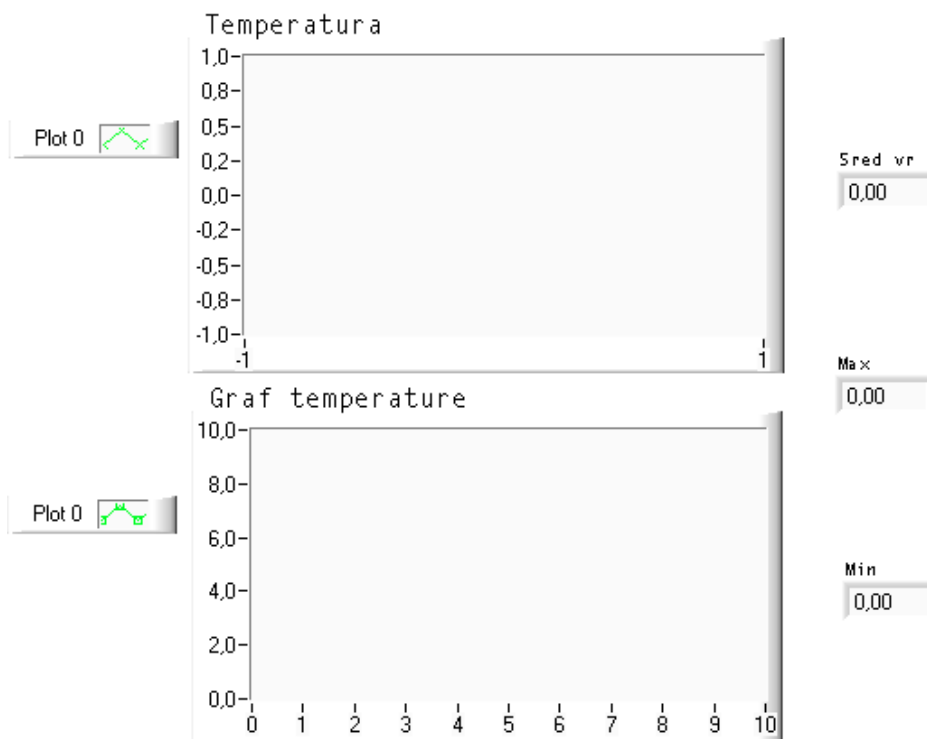
Pohranite i zatvorite VI. Nazovite je **Moj graf polja valnog oblika.vi**.

Pohranite je u direktorij (folder) **mojrad.llb**.

9.1. Korištenje VI za graf i analize

CILJ Izgradite VI koja mjeri temperaturu svakih 0.25 sekundi tijekom 10 sekundi. Za vrijeme prikupljanja podataka VI prikazuje mjerenja u realnom vremenu na strip chartu. Nakon završetka prikupljanja podataka

VI iscertava podatke na grafu i izračunava srednju vrijednost, maksimalnu i minimalnu temperaturu.



Slika 9-1.

PREDNJA PLOČA

Otvori novi prednju ploču i izgraditi sučelje kakvo je prikazano na slici. Modificirajte vrst točaka chart i graf valnog oblika, škljocom na njihove legende.

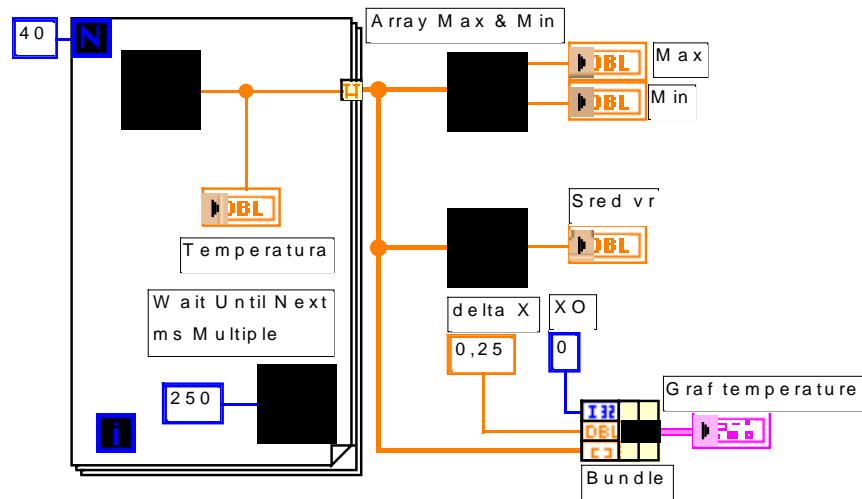
Chart valnog oblika temperature prikazuje temperaturu onako kako je prikupljena.

Nakon prikupljanja VI iscertava podatke u temp Grafu. Digitalni indikatori: Srednja vrijednost (Mean), Max i Min prikazuju srednju vrijednost, maksimalnu i minimalnu temperaturu.

BLOK DIJAGRAM

Izgraditi blok dijagram pokazan na prethodnoj slici, Koristeći slijedeće elemente:

VI termometra (**All Functions>>Select a VI...**) se nalazi u mapi **activity**. Ova funkcija nakon izvršenja daje jedno mjerenje temperature



Slika 9-2.

Čekati dok se ne izvrši funkcija *Wait until next ms multiple function* (**All Functions>>Time & Dialog**). U ovom primjeru ta funkcija osigurava da se FOR petlja izvrši svakih 0,25 sek (250 ms). Škljocnite na *Wait Until next ms Multiple function* i izabrati **Create Constant** da se automatski kreira i ožiči brojana konstanta te u nju upišite 250 .

Funkcija maksimuma i minimuma polja **Array Max&Min** (**All Functions>>Array**) daje maksimalnu i minimalnu temperaturu izmjerenu za vrijeme prikupljanja.

Funkcija srednje vrijednosti **Mean** (**Functions>>Analysis>>Probability and Statistics**) daje srednju vrijednost izmjerenih vrijednosti temperature.

Bundle funkcija (Funkcija uvezivanja iz **Functions>>Cluster**) povezuje komponente ispisa u klaster (grozd, svežanj). Komponente uključuju početnu vrijednost za $X(0)$, delta vrijednost za X (0.25), kao i polje Y vrijednosti (temperaturni podaci). Koristiti alat za pozicioniranje da se promjeni veličina funkcije vukući jedan od uglova.

FOR petlja se izvršava 40 puta. Funkcija *Wait Until Next ms Multiple* uzrokuje da svaka iteracija se dogodi poslije 250 ms. VI pohranjuje mjerenju temperaturu u polje kreirano na ivici FOR petlje (autoindeksirano). Nakon što se kompletira izvršenje FOR petlje polje prenosi podatke različitim čvorovima.

Polje Max & Min funkcije daje maksimalnu i minimalnu temperaturu. Srednja VI daje prosjek vrijednosti mjerene temperature.

Kompletan VI će povezati polje podataka sa početnom vrijednošću $X(0)$ i delta X vrijednošću (0.25). VI zahtjeva delta $X = 0.25$ tako da bi iscrtavao vrijednost temperature svakih 0.25 sekundi na grafu valnog oblika.

Vratite se na prednju ploču i pokrenite VI.

Pohranite VI u mojrad.llb kao *Moja analiza temperature.vi*.

10. Stringovi, file I/O

10.1. Stringovi

String (niz znakova) je skup ASCII znakova. Stringovi se mogu koristiti više načina a ne samo za jednostavne tekstualne poruke. Kod formiranja instrumentalnih VIa mogu se koristiti za prijenos brojčanih podatke kao niz znakova a zatim konvertirati te nizove u brojeve. Pohranjivanje brojčanih podataka na disk može također uključivati stringove. Da bi se pohranili brojevi u ASCII file, moramo prvo konvertirati brojeve u stringove prije nego što zapišemo brojeve u file na disku.

Kreiranje kontrolnih i indikacijskih elemenata stringa



Elemente za kontrolne i prikazivanje stringa mogu se naći u **Controls>>String&Table**. Upisivanje ili promijeniti tekst unutar string kontrole može se izvršiti alatom za pisanje. Povećati string kontrolne elemente i indikatore vukući uglove alatom za pozicioniranje.

Slika 10-1.

10.1.1. Stringovi i file I/O



Slika 10-2.

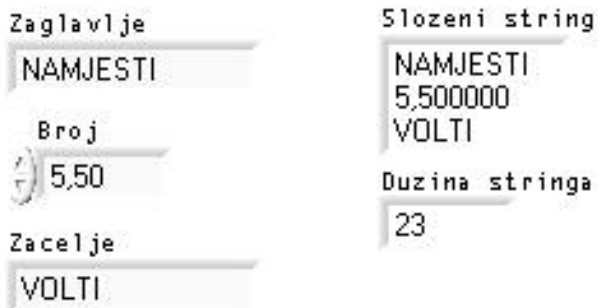
Ako želite smanjiti prostor koji zauzima kontrolni ili indikacijski element string na prednjoj ploči, izaberite **Show>> Scrollbar**. Ako je ova opcija zasjenjena tj. nedostupna morate povećati vertikalnu dimenziju prozora da je učinite raspoloživom.

10.1.2. Korištenje string funkcija

Cilj

LabVIEW ima mnogo funkcija za manipuliranje stringovima. Naći ćete ove funkcije u **Functions>>String**. Izgradit će se VI koji konvertira broj u string i spaja jedan string sa drugim (concatenate), da bi se formirao jedinstveni izlazni string. VI će također odrediti dužinu izlaznog stringa.

Prednja ploča



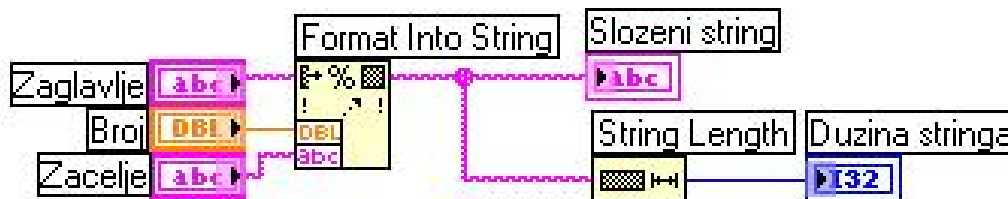
Otvoriti novu prednju ploču i izgraditi sučelje pokazano na slici. Modificirajte kontrolne i indikatorske elemente kako je to prikazano na slici. Dva kontrolna elementa za string i digitalni kontrolni element mogu biti kombinirani u jedinstveni izlazni string i prikazani u indikatoru stringa. Digitalni indikator prikazuje dužinu stringa.

Slika 10-3.

Izlaz kombiniranog stringa u ovome primjeru ima format sličan kombiniranim stringovima koji se koriste kod GPIB (IEEE 488) i serijskih (RS-232 ili RS-422) instrumenata.

Blok Dijagram

Izgraditi blok dijagram pokazan na slici 10-4.



Slika 10-4.

Funkcija **formatiraj u string (Functions>>String)** spaja (concatenate) i formatira članove stringa i stringove i jedinstveni izlazni string. Koristite kursor za promjenu velicine na ikoni da dodate tri ulaza za argumente.

Funkcija dužine stringa (**Functions>>String**) daje broj znakova u spojenom stringu.

Pokrenite VI. Uočite da funkcija Format Into String povezuje dva string kontrolna elementa u jedinstveni izlazni string.

Pohraniti VI kao **moj_string.vi**. Koristit ćete ovaj VI u slijedećem primjeru.

10.2. File I/O

LabVIEW file I/O funkcije (**Functions>>File I/O**) su moćan i fleksibilan skup alata za rad sa fileovima. Osim čitanja i pisanja podataka LabVIEW file I/O funkcije miču i mijenjaju imena fileova i direktorija (foldera), kreiraju spreadsheet (tabelarne) tipove fileova čitljivog ASCII teksta, i zapisuju podatke u binarnoj formi radi brzine i kompaktnosti.

Možete pohraniti, pronaći i učitati podatke iz fileova u ova tri različita formata:

- **ASCII byte stream** (ASCII niz byteova). Trebate pohraniti podatke u ASCII formatu kada im želite pristupiti iz drugog softwareskog paketa, kao npr. Word processora ili nekog spreadsheet programa. Da bi pohranili podatke na ovaj način, morate pretvoriti sve podatke u ASCII stringove.
- **Datalog fileovi**. Ovi fileovi su u binarnom formatu kojima samo LabVIEW može pristupiti. Datalog fileovi su slični database fileovima jer možete pohraniti nekoliko različitih tipova podataka u jedan zapis u file-u.
- **Binarni byte stream**. Ovi file-ovi su najkompaktniji i najbrža metoda pohranjivanja podataka. Morate pretvoriti podatke u binarni string format i morate znati točno koje tipove podataka koristite da pohranite i povratite podatke u i iz file-ova.

Ovo će poglavlje analizirati **ASCII byte stream** fileove jer je to najčešći file format za podatke.

10.2.1. File I/O funkcije

Najveći broj file I/O operacija uključuje tri osnovna koraka:

- otvaranje postojećeg filea ili kreiranje novog file-a;
- pisanje u ili čitanje iz filea;
- zatvaranje filea.

Zbog toga LabVIEW sadrži mnoge korisne VIe u **Functions>>File I/O**.

Ovo Poglavlje opisuje devet visokorazinskih korisničkih alata (utilities). Ove korisničke funkcije su izgrađene nad VIima srednjeg nivoa koje utjelovljuju provjeru grešaka kao i funkcije za file I/O (ulazno-izlazne manipulacije).

Možemo također postaviti razdjelnicu (delimiter) ili string razdjelnicu, kao što su tabovi (razdjelnica fiksne dužine najčešće 4 prazna polja), zarezi itd. u spreadsheet-u.

Ovo štedi napor razdjeljivanja spreadsheeta ako smo koristili razdjelnicu različitu od taba, da uspostavimo polja u spreadsheetu.

VI Upisi znakove u file (Write Characters to File), upisuje string znakova u novi byte stream file ili dodaje string već postojećem file-u. VI otvara ili kreira file, upisuje podatke i zatim zatvara file.

VI čitaj znakove iz filea (Read Characters from File) čita specificirani broj znakova iz byte stream file-a počevši od specificiranog znaka (offset). Ovaj VI otvara file ispred i zatvara ga nakon toga.

VI čitaj linije iz file-a (Read Lines From File), čita specificirani broj linija iz byte stream file-a počevši od specificiranog offseta (pomaka) . Ovaj VI otvara file prije toga i zatvara ga nakon toga.

VI Upisi u spreadsheet file (Write To Spreadsheet File), pretvara 1D (jednodimenzijaska) ili 2D (dvodimenzijaska) polja brojeva jednostruke preciznosti za string tekst i upisuje tekst u novi byte stream file ili priključuje string postojećem file-u. Može se također, kao opciju transponirati podatke (transpose). Ova VI otvara ili kreira file prije toga i zatvara ga nakon operacije. Ovaj se VI može koristiti da kreiramo tekst fileove koje mogu čitati većina spreadsheet programa.

VI Čitaj iz spreadsheet file-a (Read from Spreadsheet File), čita specificirani broj linija ili redova iz brojčanog tekst file-a, počevši od specificiranog offset znaka, i pretvarajući podatke u 2D, polje brojeva jednostruke preciznosti. Možemo opciono transponirati polje. Možemo koristiti ovu VI da čitamo spreadsheet file-ove u tekst formatu.

10.2.2. Upisivanje u Spreadsheet file

Vrlo česta aplikacija za pohranjivanje podataka u file je formatiranje tekst file tako da ga možete otvoriti u spreadsheet-u. U većini spreadsheet-ova tabovi razdjeljuju kolone a **EOL (End of line)**, kraj linije teksta) znakovi razdjeljuju redove, kako je to pokazano na narednoj slici:

0.00	0.4258	☛ = Tab
1.00	0.3073	☛ = Line Separator
2.00	0.9453	
3.00	0.9640	
4.00	0.9517	

	A	B	C
1	0	0.4258	
2	1	0.3073	
3	2	0.9453	
4	3	0.964	
5	4	0.9517	
6			

Slika 10-5.

Otvarajući file koristeći spreadsheet program daje slijedeću tablicu:

CILJ Modificirat će se postojeći VI tako da se može koristiti file I/O funkciju za pohranjivanje podatka u novi file u ASCII formatu. Kasnije, možemo pristupiti ovom file-u iz spreadsheet aplikacije.

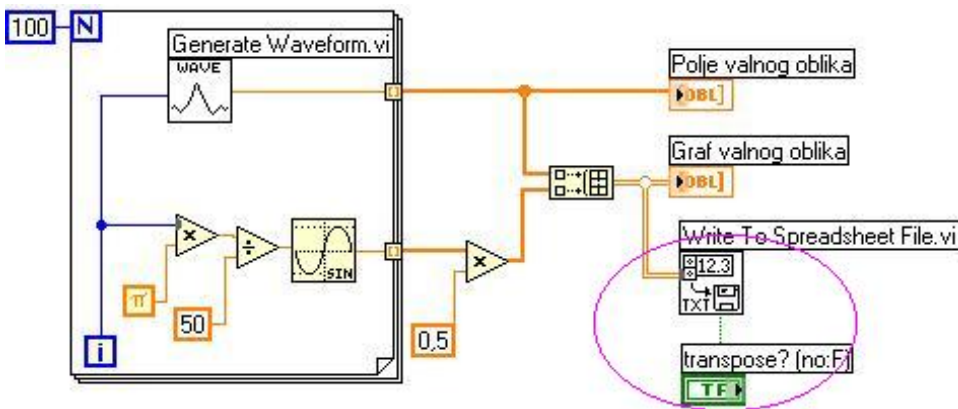


Prednja ploča

Otvorite *My Graph Waveform Arrays.vi* koji ste izgradili u ranije. Ako se sjećate tajj VI generira dva skupa podataka u poljima i iscrtava ih na grafu. Modificirat ćemo ovaj VI za upisivanje dva polja podataka u file gdje svaka kolona sadrži po jedno polje podataka.

Blok Dijagram

Otvorite blok dijagram od *My Graph Waveforms Arrays* i modificirajte VI dodajući blok dijagramu funkcije koje se nalaze unutar ovalne kružnice, kako je to prikazano na slici.



Slika 10-7.

VI upiši u spreadsheet file (Write To Spreadsheet File u All Functions>>File I/O) pretvara dvodimenzijsko polje u spreadsheet string i upisuje ga u file. Ako niste specificirali ime staze (path) gdje će se upisati file, tada će se pojaviti dijaloški okvir i zatražiti da upišete ime filea. **VI upiši u spreadsheet file** upisuje bilo 1D ili 2D polje u file. Budući je polje podataka u

ovome primjeru 2D ne mora se ožičavati u 1D ulaz. S ovim se VI može koristiti spreadsheet razdjelnica (delimiter) ili string delimiter kao što su tabovi ili zarezi u dobivenim podacima.

Booleova kontrola (**Functions>>Boolean**) provjerava transponira li ili ne LabVIEW 2D polje prije nego što ga upiše u file. Da bi se promjenila vrijednost na TRUE škljocnite na kontrolu alatom . U ovom se slušaju podaci trebaju biti transponirani jer su polja podataka specifičnih redova (tj. svaki red dvodimenzijskog polja je polje podataka). Budući da svaka kolona spreadsheet filea sadrži polje podataka 2D polje se mora najprije transponirati.

Vratite se na prednja ploču i pokrenite VI. Nakon što se polja podataka generiraju file dijalog okvir će zatražiti ime filea za novi file kojeg kreiramo. Unesite ime filea i škljocnite na **OK**.

*Opaska: Ne pokušavajte unijeti podatke u VI knjižice kao što je **mojrad.llb**. Ukoliko ovo pokušate to može rezultirati u prepisivanju preko vaše knjižice i gubitku prethodnog rada.*

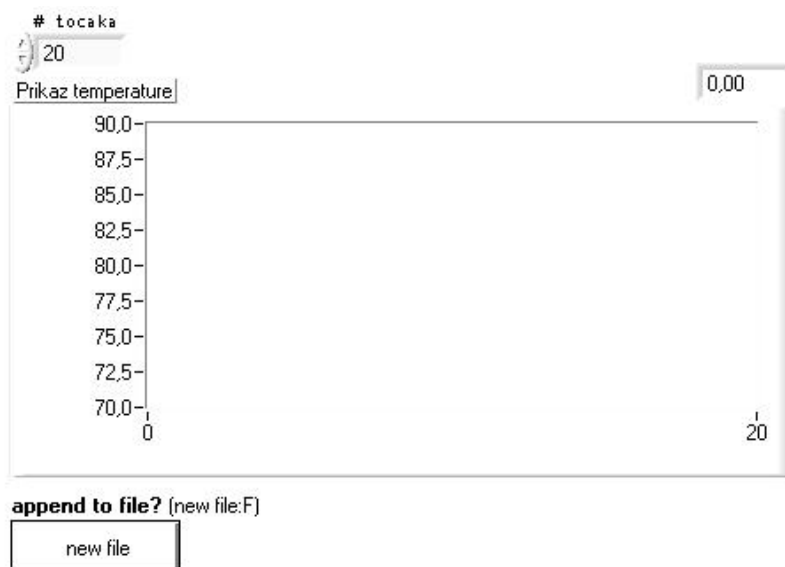
Pohranite VI, nazovite je My Waveform Arrays to File i zatvorite VI.

Sada možete koristiti spreadsheet program ili editor teksta da otvorite i pogledate u file koji ste upravo kreirali. Trebate vidjeti dvije kolone od po 100 elemenata. U ovome primjeru podaci nisu bili pretvoreni ili upisani u file sve dok sva polja podataka nisu bila prikupljena. Ako prikupljamo velike spremnike podataka ili bi htjeli upisati vrijednosti podataka na disk u trenutku kada su generirani tada moramo koristiti različite VI za **File I/O**.

10.2.3. Dodavanje podataka u file

CILJ Kreirati VI u koji će dodavati podatke temperature u file u ASCII formatu. Ova VI koristi For konturu da generira temperaturne vrijednosti i pohrani ih u file. Za vrijeme svake iteracije vršit će se pretvaranje podataka u string, dodati zarez (comma) kao rezdjelnik znakova (delimiter), i pridodati (append) string fileu.

Prednja ploča

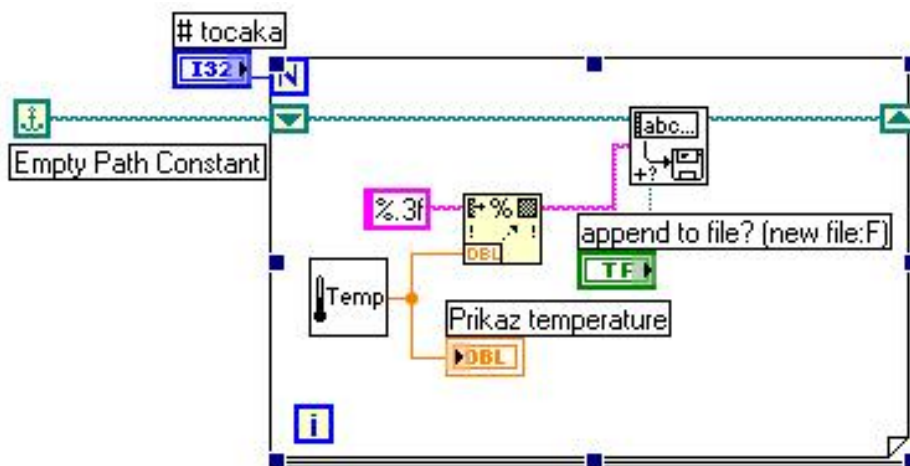


Slika 10-8.

Otvoriti novi Prednja ploča i postaviti objekte kako je to pokazano na prethodnoj slici.

Pored digitalne brojačane kontrolne na prednjoj ploči je i chart valnog oblika. Izaberite **Show>>Digital Display**. Kontrolni element # **točaka** specificira koliko vrijednosti temperaturnih treba prikupiti i upisati u file. Chart prikazuje krivulju temperature. Postaviti na ljestvici y osi charta opseg 70.0 do 90.0 na x osi opseg 0 do 20. Škljocnuti (pop-up) na # **točaka** digitalnu kontrolu i izabrati **Representation>>Long**.

Blok Dijagram



Slika 10-9.

Otvoriti blok dijagram

Dodati For petlju i uvećati je. Ova VI generira niz temperaturnih vrijednosti specificiranih sa # točaka kontrolnim elementom. Dodati pomični registar u konturu škljocom na ivicu konture. Ovaj pomični registar sadrži ime staze (path name) do filea.

Završiti ožičenje objekata.

VI Isprazni konstantu staze (Empty path constant ,iz Functions>>File I/O >>File Constants) . Ova funkcija inicira pomični registar tako da prvi put kada pokušate upisati vrijednost u file, staza je prazna. File dijalog okvir će tražiti unos imena filea.

VI Digital Thermometer (Functions>>Tutorial) daje simuliranu vrijednost mjerene temperature iz senzora temperature.

Funkcija Formatiraj u string (Functions>>String) pretvara vrijednost temperature (broj) u string i dodaje mu zarez koji ga slijedi .

String konstanta (Functions>>String). Ovaj format specificira da mi zelimo pretvoriti broj u format stringa sa frakcijom u string sa zarezom.

VI Upisi znakove u file (Write Characters To File , u (All Functions>>File I/O) upisuje string znakova u file.

Booleova konstanta (Functions>>Boolean) postavlja **append to file?** ulaz od Write Characters To File Vi -a na vrijednost TRUE tako da nove vrijednosti temperature su pridodate u izabrani file svaki put kada kontura se ponovno izvrsava (iterira). Škljocnite sa Operativnim alatom na konstantu da bi postavili njenu vrijednost na TRUE.

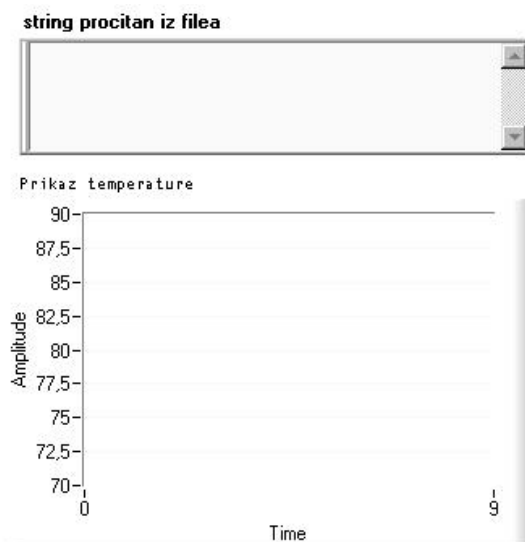
Vratite se na Prednja ploča i izvršite VI sa # točaka postavljeno na 20. File dijalog okvir će zatražiti da unesete ime filea. Kada unesete ime file-a, VI počinje upisivanje vrijednosti temperaturnih u taj file kako se generise svaka tacka.

Pohranite VI, imenujte je My Write Temperature to File.vi, i zatvorite VI. Koristite bilo koji software za procesiranje teksta kao naprimjer Write za Windows-e da otvorite taj file podataka i posmatrate njegov sadrzaj. Treba da ste dobili file koji sadrzi 20 vrijednosti podataka (sa preciznoscu od 3 mjesta nakon decimalne tacke) razdjeljenih medjusobno sa zarezima.

10.2.4. Čitanje podataka iz File-a

Cilj Kreirati VI koji čita file podataka koje smo upisali u prethodnom primjeru i prikazati te podatke na grafu valnog oblika. Moramo citati podatke u istom formatu u kojem smo ih pohranili. Zbog toga, posto smo prvobitno pohranili podatke u ASCII formatu koristeći string tip podataka, moramo ih čitati u obliku stringa podataka sa jednim od VIa za file I/O.

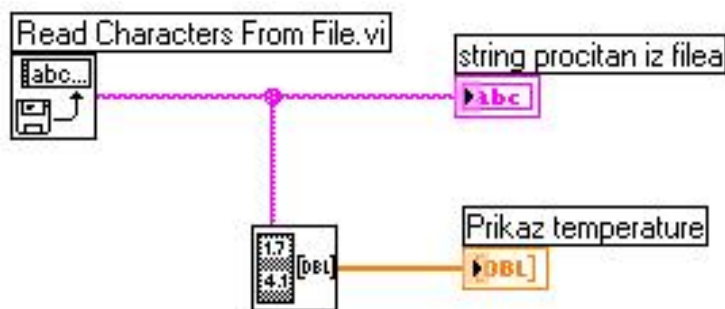
Prednja ploča



Otvoriti novi VI i izgradi prednja ploču pokazan na prethodnoj slici.

Prednja ploča sadrži string indikator i graf valnog oblika. Indikator **String Read from File** prikazuje temperaturene podatke razdjeljene sa zarezima iz file-a kojeg smo upisali u posljednjem primjeru kojeg smo uradili. Graf prikazuje krivulju temperature.

Slika 10-10.



Slika 10-11.

Blok Dijagram

Izgradite blok dijagram kakav je prikazan na prethodnoj slici.

VI **Čitaj znakove iz filea (Read Characters From File iz Functions>>File I/O)** čita podatke iz file-a daje i informacije u string. Ako ime staze (path) nije specificirano file dijalog okvir će tražiti unos imena filea. U ovome primjeru ne treba određivti broj znakova koje treba pročitati jer je broj znakova u fileu od postavljene vrijednosti (512).

Mora se znati kako su podatci pohranjeni u file da bi ih iščitali iz filea. Ako znate koliko je dugačak file možete koristiti

VI **Čitaj znakove if file-a (Read Characters From File)** određuje poznati broj znakova koje treba pročitati.

VI **Izvuci brojeve (Extract Numbers iz Functions>>Tutorials)** uzima ASCII string koji sadrži brojeve odijeljene sa zarezima LF (**line feed** nova linija) ili drugim nebrojčanim znakovima i pretvara ih u polje brojčanih vrijednosti.

Vratite se na Prednja ploča i pokrenite VI.

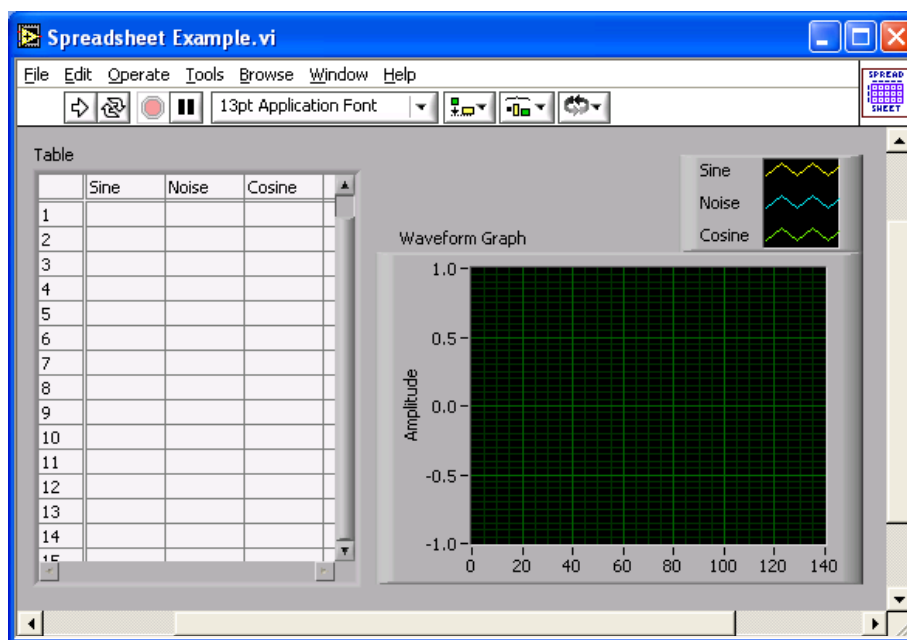
Izaberite file podataka koji ste upravo upisali na disk kada vam to zatraži file dijalog okvir. Trebate vidjeti iste vrijednosti podataka prikazane u grafu kao što ste ih vidjeli u primjeru VI My Write temperature to File .

Pohranite VI, nazovite je My Temperature from File.vi , i zatvorite VI.

10.2.5. Još spreadsheet primjera

Cilj Kreirajti VI koji sprema brojčano polje u file u obliku takvog formata kojem se može pristupiti spreadsheet programom.

Otvorite novi VI i izradite slijedeću prednju ploču VI..



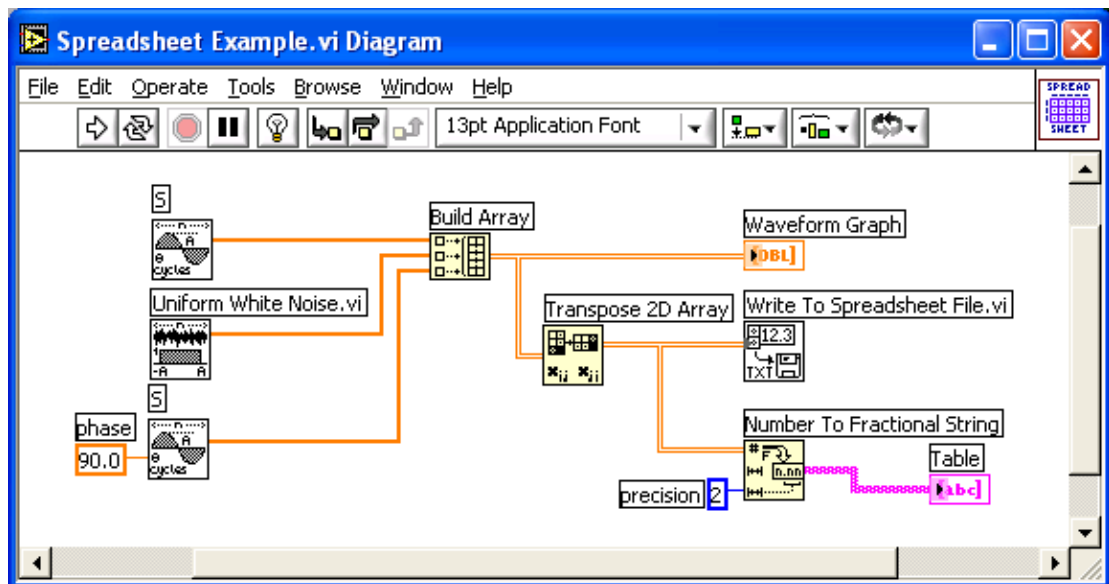
Slika 10-12.

Sine Pattern VI je smješten u **Functions»Analyze»Signal Processing »Signal Generation** daje brojčano polje od 128 elementa koji čine sinusni uzorak. Konstanta 90.0, u drugom čvoru određuje fazu sineusnog ili kosinusnog uzorka.

Uniform White Noise VI je smješten u **Functions»Analyze»Signal Processing» Signal Generation** daje brojčano polje od 128 elementa koji čine uzorak šuma

Build Array funkcija smještena u **Functions»Array** gradi 2D polje sastavljeno od polja sinusnog uzorka, polja uzorka šuma i polja kosinusnog uzorka.

Transpose 2D Array funkcija smještena je u **Functions»Array** rearanžira elemente 2D polja tako da element $[i,j]$ postaje element $[j,i]$.



Slika 10-13.

Write To Spreadsheet File VI smještena u **Functions»File I/O** formatira 2D polje u spreadsheet string i upisuje string u file.

Number To Fractional String funkcija smještena u **Functions»String» String/Number Conversion** pretvara polje brojčanih vrijednosti u polje stringa koji pokazuje tablica.

Pokrenite VI. VI generira 2D polje od 128 redova i 3 stupca. Prvi stupac sadrži podatke sinusnog valnog oblika, drugi stupac sadrži podatke valnog oblika šuma a treći stupac sadrži podatke kosinusnog valnog oblika. VI crta svaki stupac u graf i prikazuje podatke u tablici. Nakon što VI prikaže podatke pojavit će se dijalog okvir.

Utiskajte wave.txt, i škljocnite **Save** ili **OK** gumb.

Otvorite file pomoću word procesora ili spreadsheet aplikacije (Notepad ili WordPad) i pogledajte njegov sadežaj .

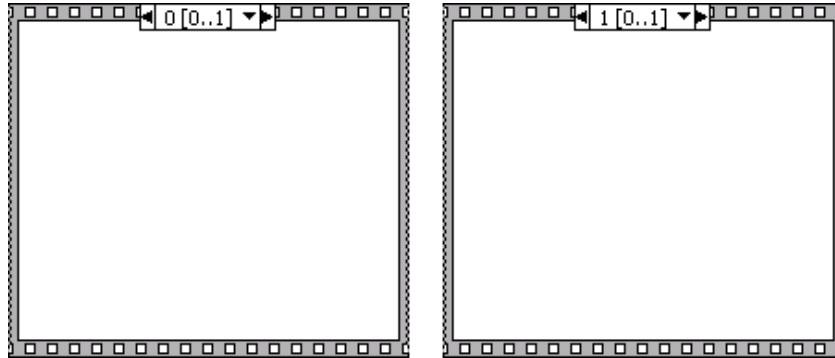
Podaci sinusnog valnog oblika pojavljuju se u prvom stupcu, podatci valnog oblika šuma pojavljuju se u drugom stupcu a podatci kosinusnog valnog oblika pojavljuju se u trećem stupcu.

Izađite iz word procesora ili spreadsheet aplikacije i vratite se u LabVIEW.

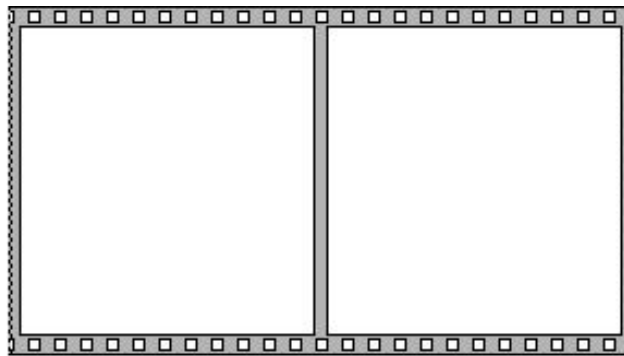
Spremite VI as **Spreadsheet Example.vi**. i zatvorite VI.

11. Struktura sekvence i čvor svojstava (Property Node)

11.1. Korištenje strukture sekvence



Sl. 11 -1. Stacked Sequence structure.



Sl.11 - 2. Flat Sequence Structure

Struktura sekvence koja izgleda kao okvir filma ima jedan ili više poddijagrama ili okvira (frames) i izvršava ih slijedno. U konvencionalnim programskim jezicima, programske naredbe se izvršavaju u redoslijedu u kojem se pojavljuju. U *data flow* programiranju čvor se izvršava kada su podatci raspoloživi na svim ulazima čvora iako je zbog toga potrebno izvršiti jedan čvor prije drugoga. LabVIEW koristi sekvencijalnu strukturu kao metodu kontrole redoslijeda kojim se čvorovi izvršava

Strukturu sekvence može se u blok dijagram postaviti kao **Stacked Sequence structure** ili **Flat Sequence Structure (Functions>>All Functions >>Structures)**.

Kod **Stacked Sequence structure** prikaza naslov okvira pri vrhu strukturu sekvence je sličan naslovu izbora slučaja CASE strukture. Naslovi okvira imaju broj okvira u sredini i strjelice za povećavanje i smanjivanja na svakoj strani. Škljocom na te strjelice mogu se pregledati svi raspoloživi okviri. Može se škljocnuti i na strjelicu koja pokazuje dalje i izabrati okvir iz padajućeg izbornika. Za razliku od naziva izbornika slučaja na mogu se upisivati vrijednosti u naslov okvira. LabVIEW sam postavlja brojeve u naslove okvira ukoliko se okviri dodaju, brišu ili rearanžiraju.

Sekvence struktura izvršava **okvir 0, okvir 1, okvir 2** sve do posljednjeg. Izvršavanje programa i davanje podataka neće završiti dok se i zadnji okvir ne izvrši.

Izbornik brojeva okvira pri vrhu sevice strukture pokazuje broj trenutnog okvira i opseg okvira u sredini između strjelice za povećavanje i smanjivanja na svakoj strani. Npr. broj **0** broj trenutnog okvir a [0..2] je opseg okvira.

Sekvence strukturu treba koristiti da bi se kontrolirao redosljed izvršavanja blok dijagrama kad prirodni slijed ovisnosti podataka ne postoji. Čvor koji prima podatke od drugog čvora ovisi o nekom prethodnom čvoru i mora se izvršiti poslije nego što su se svi prethodni čvorovi izvršili

Unutar svakog okvira sekvence strukture kao u ostatku blok dijagrama ovisnost o podacima određuje redosljed izvršavanja čvora

Tuneli sekvence strukture mogu imati samo jedan izvor podataka za razliku od CASE strukture. Izlaz može postojati iz bilo kojeg okvira ali podatci napuštaju sevice strukture samo kad svi okviru završe izvršavanje tj. ne napuštaju sevice strukture kad se pojedini okvir izvrši. Kao i kod CASE strukture podatci na ulaznim tunelima su na raspolaganju svim okvirima..

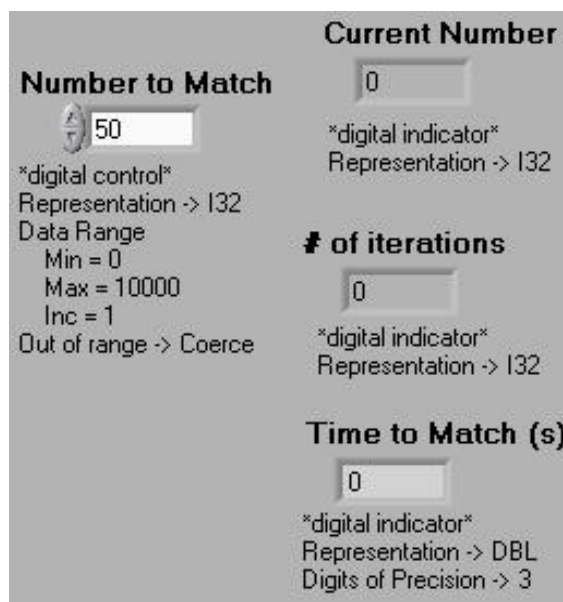
To bi se prosljedili podatci iz jednog okvira u bilo koji slijedeći okvir treba koristiti priključak lokalne sekvence (**sequence local terminal**). Pojavit će se mali kvadratični blok a strjelica koja pokazuje prema van unutar kvadrata će se automatski pojaviti kada ožičite funkciju do elementa koji predstavlja izvor podataka. Priključci na svim slijedećim okvirima imaju strjelicu koja pokazuje prema unutra. Pokazujući da je taj priključak izvor podataka za taj okvir. Ne može se koristiti priključak lokalne sekvence u okvirima prethode prvom okviru gdje je ožičena lokalna sekvenca.

Cilj Izgradit će se VI koj izračunava vrijeme potrebno da se generira slučajni broj koji se podudara sa zadanim brojem.

Prednja ploča

Otvoriti novu prednju ploča i izgraditi je kao pokazan na slijedećoj slici. Modificirate kontrolne elemente i indikatore kao što je to opisano u tekstu koji slijedi nakon slike.

U kontrolnom elementu **Number to Match** se nalazi broj s kojim se treba izvršiti uparivanje (**match**). Indikator trenutnog broja (**Current Number**) prikazuje trenutno generirani slučajni broj. Indikator sa oznakom # **iritacija** prikazuje broj iteracija prije uparivanja (**match**). **Indikator Time to Match** pokazuje koliko je sekundi trebalo da se pronađe uparujući broj.



Slika 11-3.

11.1.1. Modifikacija formata brojčane vrijednosti

Prema postavci LabVIEW prikazuje vrijednosti u brojčanim kontrolnim elementima u decimalnoj notaciji s dva decimalna mjesta (npr. 3.14). Može se koristiti **Format & Precision...** opcija u padajućem izborniku indikatora ili kontrolnog elementa za promjenu preciznosti prikaza ili za prikaz vrijednosti u znanstvenoj, inženjerskoj ili automatskoj notaciji. Može se također

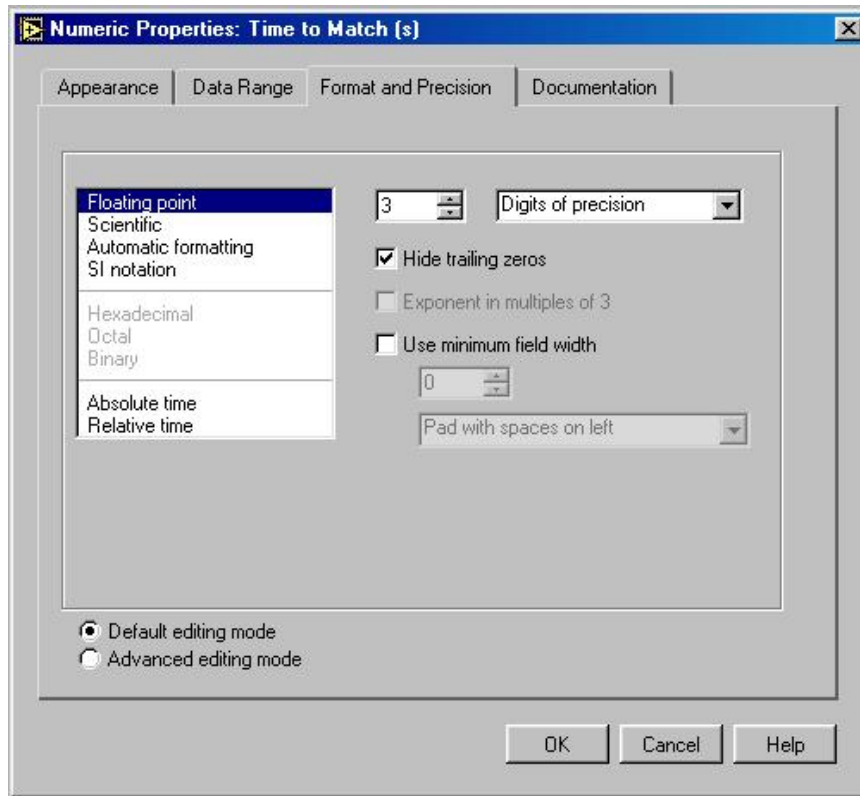
koristiti **Format & Precision...** opciju za definiranje formata vremena i nadnevka za brojčane elemente (**Absolute time** i **Relative time**).

1. Promijeniti preciznost na indikatoru **Time to Match**.

a) Škljocnuti na **Time to Match** digitalni indikator i izabrati **Format & Precision...**

➤ Mora se biti u prednjoj ploči da bi mogli pristupiti meniju.

b) Unijeti vrijednost 3 za **Digits of Precision** i škljocnuti na OK.



Slika 11-4.

2. Promijeniti prikazivanje digitalnog kontrolnog elementa i dva digitalna indikatora na **long integer**.

a) Škljocnite na digitalni kontrolni element **Number to match** i izaberite **Representation>>Long**.

Ponoviti prethodne korake za **Current Number** kao i za **# of iterations** digitalne indikatore.

Postavljanje opsega podataka (Data range)

Opcijom **Data Range...** može se spriječiti korisnika vašeg programa da postavi vrijednost kontrolnog elementa ili indikatora izvan zadanog opsega ili inkrementa promjene vrijednosti. Na raspolaganju su opcije:

- da ignorirate vrijednost izvan opsega
- prisiliti je da ostane unutar dopuštenog opsega

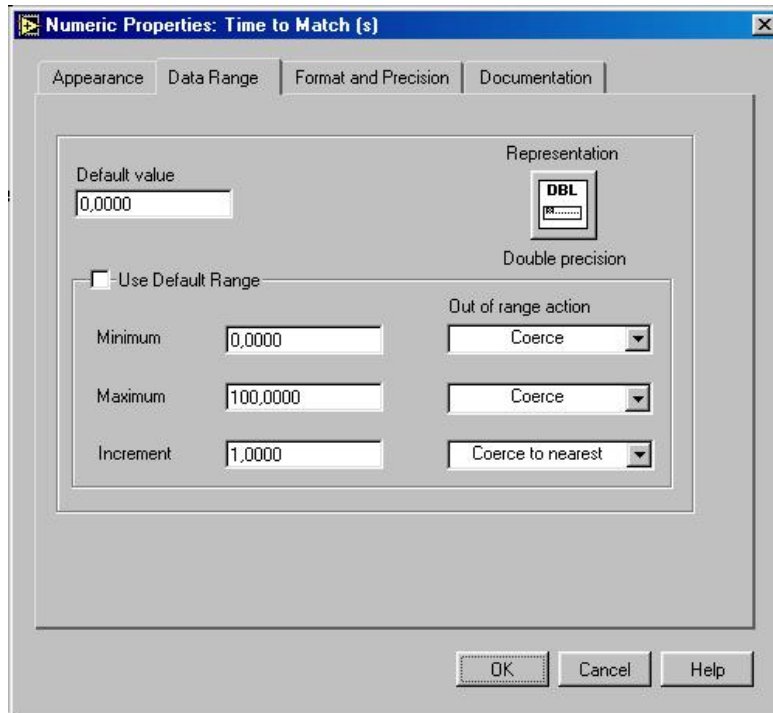
- ili da zaustavi izvršenje VI.

Simbol greške opsega pojavit će se umjesto gumba za izvršenje na alatnom traku kad greška opsega zaustavi izvršenje. Također će tamni okvir označiti kontrolni element koji je izvan opsega.

1. Postaviti opseg podataka između 0 i 100 sa inkrementom 1.

a) Škljocnuti na **Time to Match** indikator i izabrati **Data Range**.

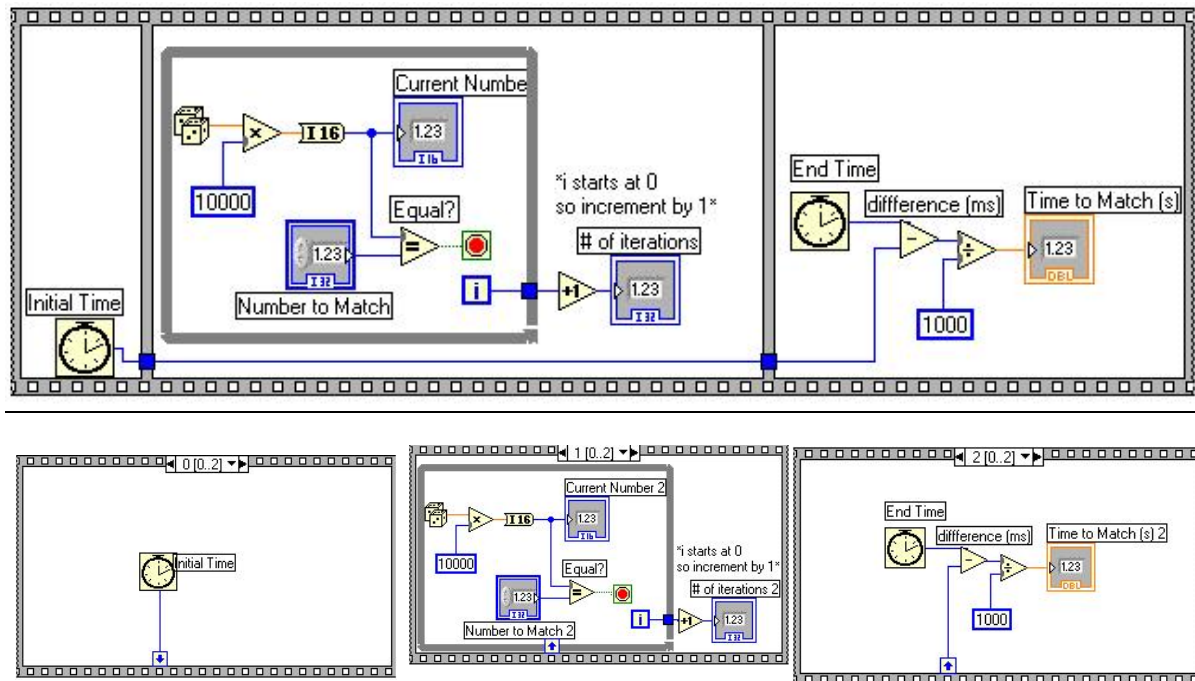
b) Unjeti tekst u dijalog prozor, kako je pokazano na narednoj ilustraciji i škljocnuti na **OK**.



Slika 11-5.

Blok dijagram

1. Otvoriti blok dijagram



Slika 11-6.

LabVIEW postavlja blok dijagram kojeg će prvog izvršiti unutar granica **okvira 0** zatim postavlja dijagram kojeg izvršava drugog unutar granica **okvira 1** itd. Slično kao u slučaju CASE strukture samo što je jedan **okvir** vidljiv u svakom trenutku vremena.

2. Povećati strukturu vučenjem jednog ugla sa kursorom za promjenu veličine.

3. Kreirati novi **okvir** (okvir) škljocom na granici okvira i izabirući **Add Okvir After**. Ponoviti ovaj korak da se kreira okvir 2. Kod **Stacked Sequence structure** treba na **okviru 0** kreirati mali blok na ivici okvira sekvencijalne strukture unutar kojeg će se pojaviti strjelica kad se priključi ura početnog vremena (Initial Time). Taj blok je lokalna varijabla sekvence koja prenosi podatke između okvira sekvencijalne strukture.

Podaci ožičeni do okvira lokalne sekvence bit će na raspolaganju u narednim okvirima. Međutim ne može se pristupiti podacima u okvirima koji prethode okviru u kojem ste kreirali lokalnu sekvencu.

4. Kreirati lokalnu sekvencu škljocom na donjoj granici **okvira 0** i birajući **Add Sequence Local**. Lokalna sekvencu se pojavljuje kao prazni kvadrat. Strjelica unutar kvadrata se automatski pojavljuje kada ožičite funkciju do lokalne sekvence.

5. Završiti blok dijagram koristeći ove funkcije

Funkcija brojača otkucaja (**Tick count (ms)**) iz **Functions>>Time & Dialog** daje broj milisekundi koje su prošle nakon uključenja napajanja. Za ovaj primjer trebat će dvije funkcije **Tick Count**.

Funkcija slučajnog broja (0 - 1) (**Functions>>Numeric**) daje slučajni broj između 0 i 1.

Funkcija množenja (**Multiply** iz **Functions>>Numeric**) množi slučajni broj sa 100 tj. funkcija daje slučajni broj između 0.0 i 100.0.

Funkcija brojčane konstante (**Functions>>Numeric**) predstavlja maksimalni broj koji može biti pomnožen.

Funkcija **Zaokruži na najbližu vrijednost** (**Round to nearest** iz **Functions>>Numeric**) zaokružuje slučajni broj između 0 i 100 na najbliži cijeli broj.

Funkcija **Nije jednako?** (**Not equal?**, iz **Functions>>Comparison**) uspoređuje slučajni broj sa brojem upisanim na prednjoj ploči i daje TRUE ako brojevi nisu jednaki. U suprotnom slučaju ova funkcija daje FALSE.

Funkcija inkrementa (**Functions>>Numeric**) povećava brojač WHILE petlje za 1.

Funkcija oduzimanja (**Functions>>Numeric**) daje vrijeme (u milisekundama) proteklo između Okvira 2 i Okvira 0.

Funkcija dijeljenja (**Functions>>Numeric**) dijeli broj milisekundi koji je protekao s 1000 da bi pretvorila broj u sekunde.

Brojčana konstanta (**Functions>>Numeric**) pretvara broj iz milisekundi u sekunde.

U **okviru 0** funkcija **Broj otkucaja** (**Tick Count**) daje trenutno vrijeme u milisekundama. Ova funkcija je ožičena sa lokalnom sekvencom pa je vrijednost na raspolaganju u narednim okvirima. U **okviru 1** VI izvršava WHILE petlju sve dok se upisani broj ne upari sa slučajnim brojem (0 – 1) koji ta funkcija daje. U **okviru 2** funkcija **Tick Count (ms)** daje novo vrijeme u milisekundama. VI oduzima staro vrijeme (preneseno iz okvira 0 kroz lokalnu sekvencu) od novog vremena da izračuna proteklo vrijeme.

6. Vratiti se na prednja ploča i unijeti broj unutar kontrolnog elementa **Number to Match** i izvršiti VI.

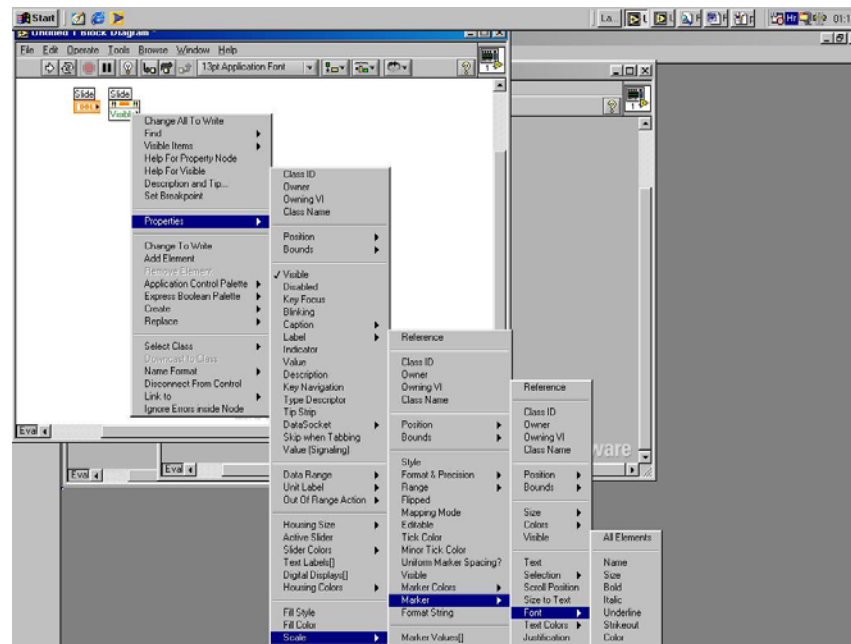
7. Pohraniti Vi i zatvoriti je. Nazvati je **My time to match.vi**.

11.2. Čvor svojstava (Property Nodes)

Čvor svojstava (**Property Node**) treba koristiti da se dobiju i postave različita svojstva na nekoj aplikaciji ili VI. Čvor svojstava se pojavljuje kad na ikonu nekog objekta na prednjoj ploči ili na blok dijagramu škljocnemo desnom tipkom mišem i odaberemo **Create >>Property Node** kao što je to prikazano na slici u primjeru horizontalnog klizača.

Nakon toga će se prednja ploča automatski pojaviti čvor svojstava na blok dijagramu. Na njemu je moguće odaberi svojstva nekog čvora desnim škljocom na tekst nekog svojstava (obično **Visible**) i odabirom **Properties**.

Na donjoj je slici prikazano otvaranje svojstava fonta oznake ljestvice u slijedu **Properties>>Scale>>Marker>>Font**.



Slika 11-6.

Možete učitati ili upisati mnoga svojstva koristeći jedan čvor. Pomoću alata za pozicioniranje može se povećati čvor svojstava tako da se pojave novi priključci. Mala strjelica s desne strane čvora svojstava znači da se to svojstvo učitavate.

Mala strjelica s lijeve strane čvora svojstava znači da se to svojstvo upisujete

Škljocnite desnom tipkom miša na svojstvo i odaberite **Change to Read** ili **Change to Write** iz izbornika prečice da bi promijenili vrst operacije svojstava.

Čvor se izvršava odozgo prema dolje. Čvor svojstava se neće izvršiti ukoliko se dogodi greška prije no što se izvrši. Stoga uvijek provjerite moguće greške. Ukoliko se greška pojavi unutar svojstava LabVIEW ignorira ostala svojstva i daje grešku. **Error out** klaster sadrži informaciju o tome koje je svojstvo izazvalo grešku.

11.2.1. Implicitno povezani čvor svojstava

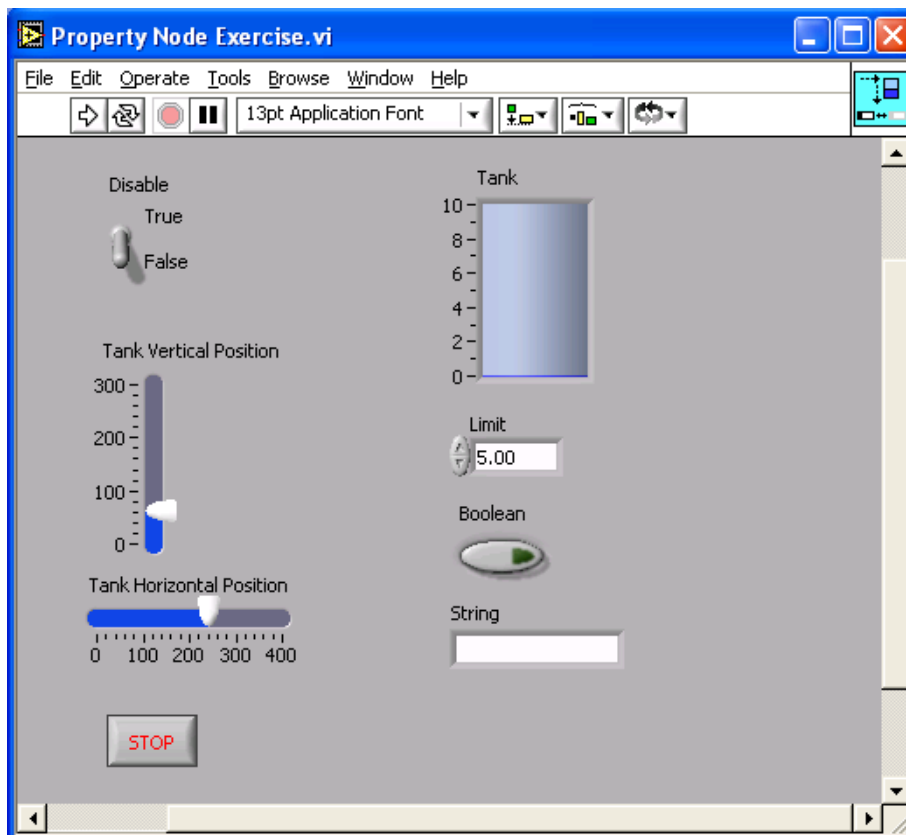
Kad se tvori čvor svojstava od objekata s prednje ploče desnim škljocomna objekt o odabirom **Create»Property Node** iz izbornika prečeca, LabVIEW stvara čvor svojstava na blok dijagramu koji je implicitno vezan za objekt prednje ploče. Zbog toga što su ovi čvorovi svojstava are implicitno vezani za objekt od kojeg je stvoren oni nemaju **refnum** ulaz pa netrebate ožičiti čvor svojstava na terminal objekta s prednje ploče ili kontrolni refnum.

Cilj Sagraditi VI koji programski mijenja položaj, onemogućuje mijenjanje svojstava prednje ploče i boji sadržaj spremnika

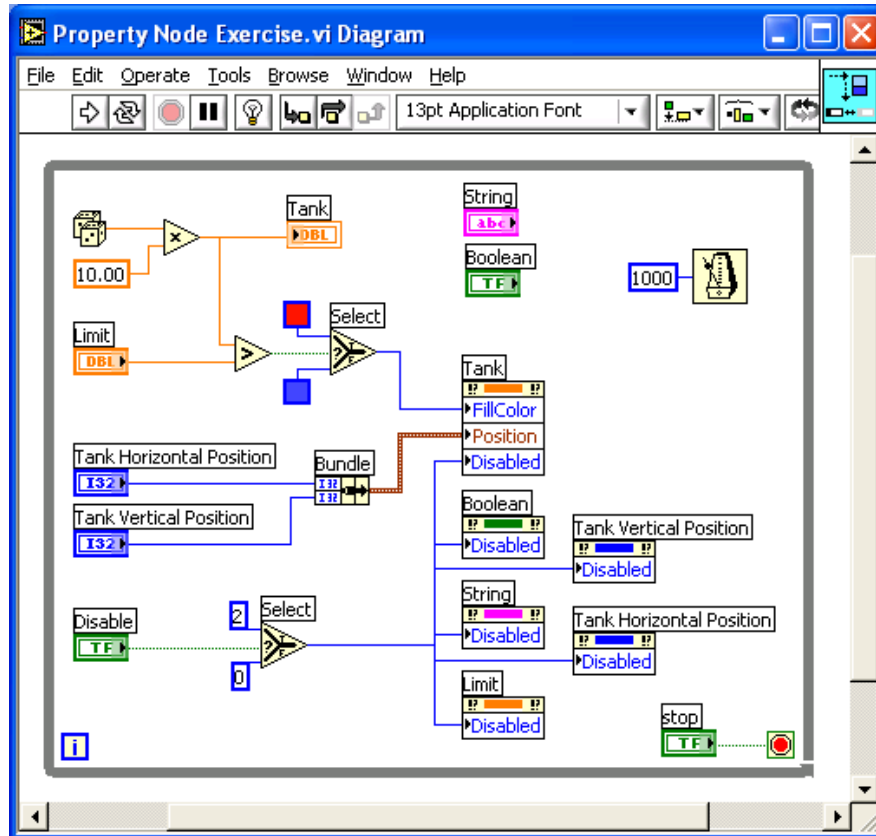
Otvorite novi VI sagradite slijedeću prednju ploču i blok dijagram

Spremi VI kao Vježba čvor svojstava.vi.

Vratite se na prednju ploču i pokrenite VI..



Slika 11-7.



Slika 11-8.

Događat će se nekoliko stvari. Prvo kako se generiraju i upisuju novi slučajni brojevi u spremnik (tank) boja ispunje je crvena ako je generirana slučajna vrijednost veća od granične vrijednosti a boja postaje plava ako je slučajna vrijednost manja od granične.

Dva klizača mijenjaju položaj spremnika na ploči. Pomakni te vrijednosti i pogledaj kako se spremnik pomiče.

Disable sklopka upravlja time mogu li se promijeniti vrijednosti ili ne. Prebaci sklopku **Disable** na **True** i svi objekti na ploči osim te sklopke i **Stop** gumba posive tako da se ne mogu mijenjati njegove vrijednosti.

Zaustavite i zatvorite VI kad završite.