

Elektrotehnički fakultet – Univerzitet u Beogradu

KATEDRA ZA ELEKTRONIKU



Vežbe

# Praktikum iz Virtuelne instrumentacije

1. Čas

*Školska 2019/20*

Asistent: Haris Turkmanović (*haris@etf.rs*)

# 1. Zadatak

Potrebno je napraviti VI koji predstavlja generator prostoperiodičnog signala. Ovaj VI treba da ima sledeće funkcionalnosti:

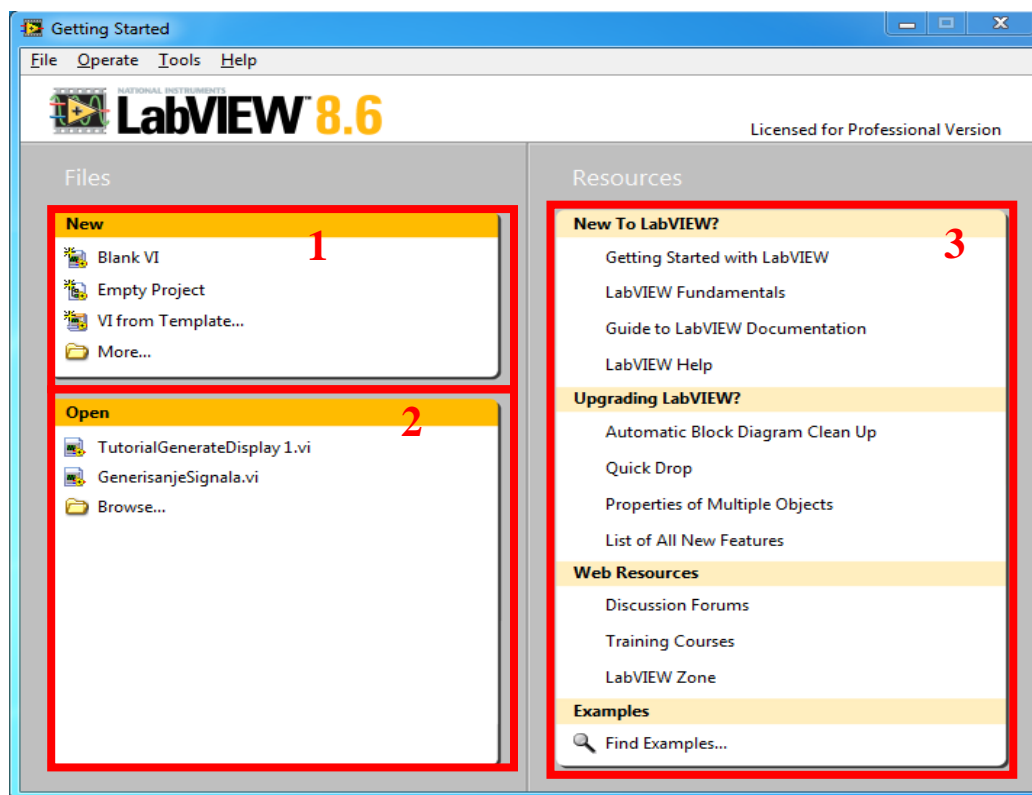
- Podešavanje karakteristika prostoperiodičnog signala (amplituda, perioda, ofset)
- Odabir više oblika prostoperiodičnog signala
- Matematičke transformacije odabranog signala (apsolutna vrednost, srednja kvadratna vrednost, kvadriranje, korenovanje, ...)

Zahtevani instrument realizovati primenom LabVIEW razvojnog okruženja

Rešenje:

Kao što je poznato, LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) je posebno softversko razvojno okruženje, kreirano od strane *National Instruments*, koje omogućava kreiranje programa na osnovu grafički dostupnih blokova koji realizuju određene funkcionalnosti. Drugim rečima, programiranje je svedeno na povezivanje određenih blokova u cilju kreiranja blok šeme sistema koja će funkcionalno ilustrovati funkcionalnost programa koji se realizuje. Svi programi kreirani posredstvom LabVIEW softverskog razvojnog okruženja nazivaju se virtuelni instrumenti i imaju ekstenziju *.vi*. Ovaj zadatak ima za cilj da uvede neke osnovne gradivne blokove koje ćemo koristiti kroz nastavak svih laboratorijskih vežbi ali isto ako da objasni neke osnovne delove LabVIEW softverskog razvojnog okruženja.

Na ovim laboratorijskim vežbama iz praktikuma iz virtuelne instrumentacije korišćemo LabVIEW 8.6 softverski paket. Nakon pokretanja ovog programskog paketa dobijamo izgled prozora kao na slici 1.1.

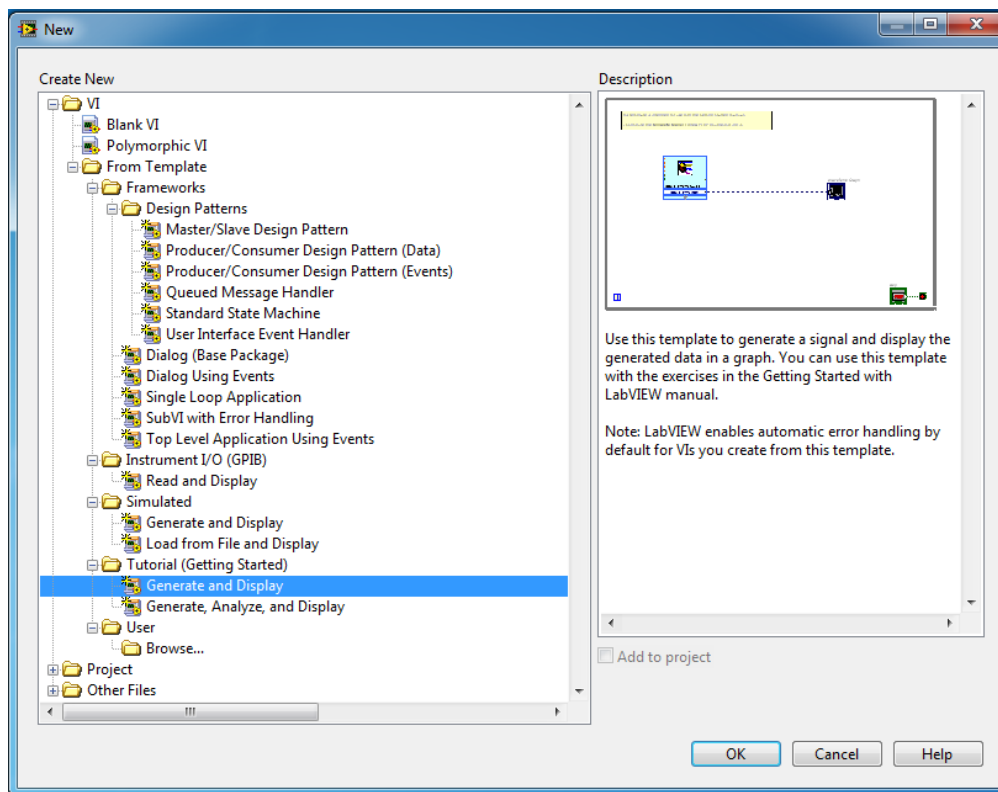


1.1 - Izgled početnog ekrana LABView softverskog paketa

U okviru ovog prozora nude se tri opcije:

1. Kreiranje novog projekta (*New*)
2. Otvaranje postojećeg projekta (*Open*)
3. Upoznavanje sa raznim aspektima programa posredstvom Help sistema (*New To LabVIEW, Upgrading LabVIEW...*)

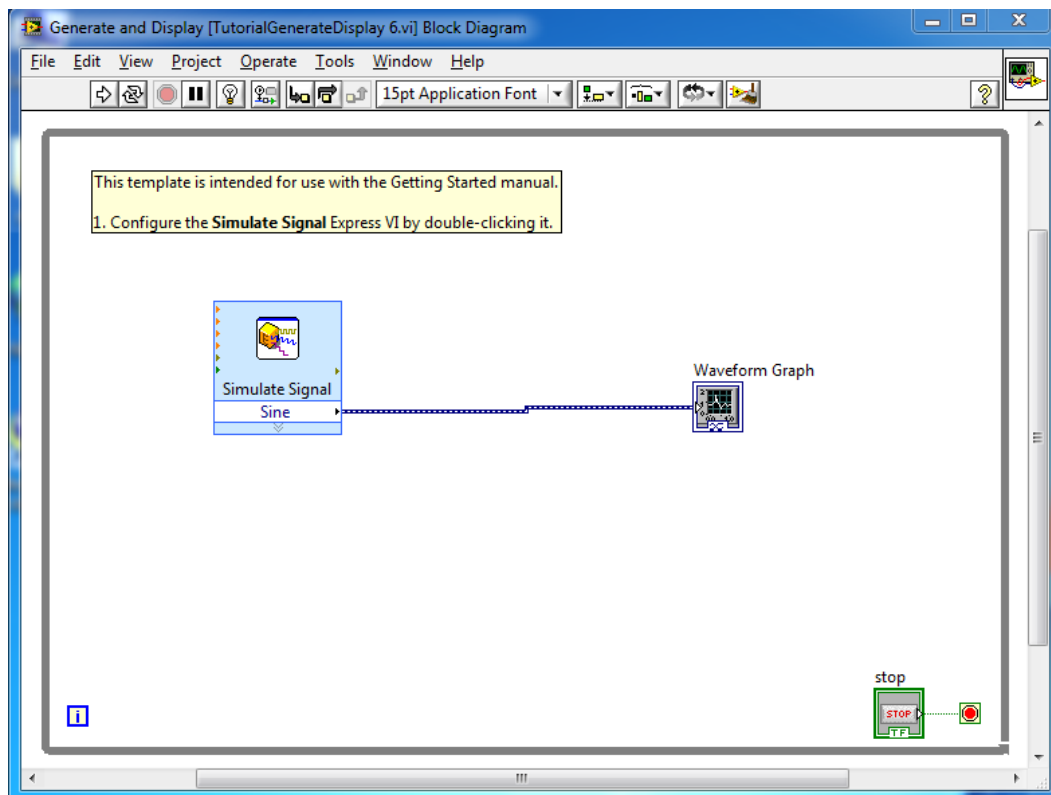
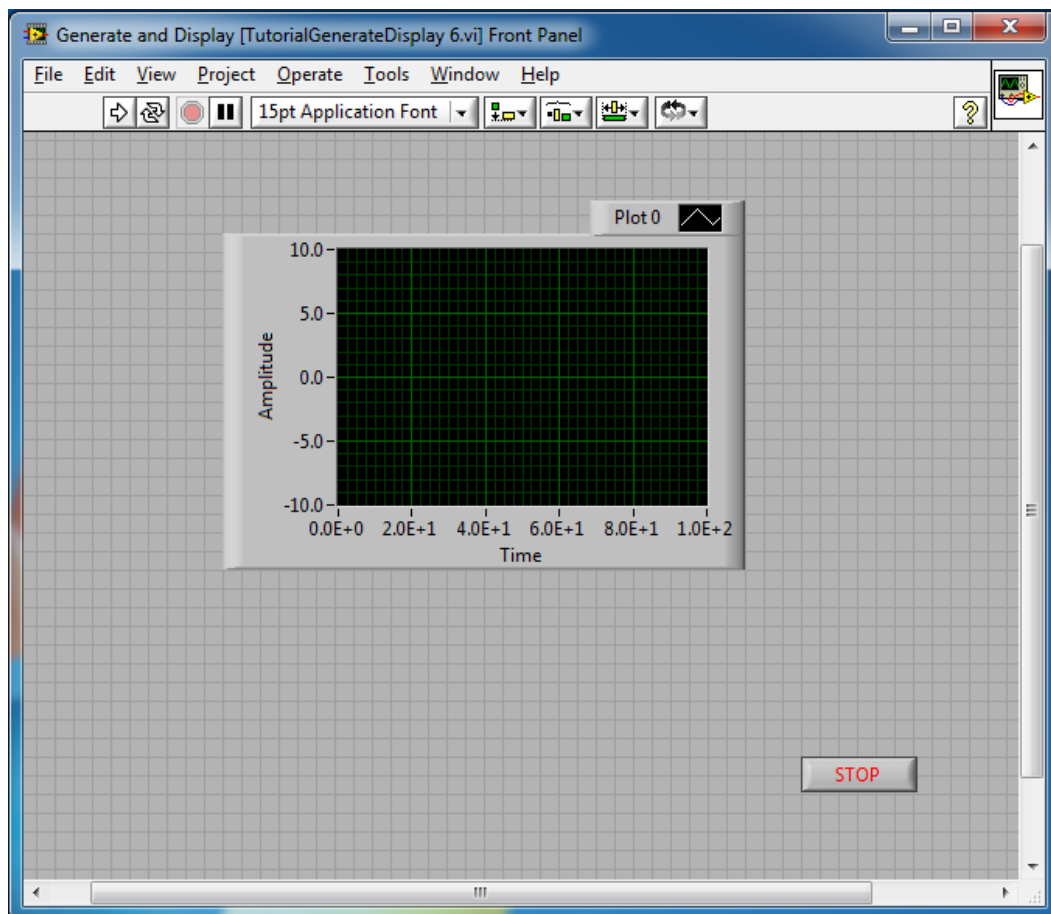
U okviru ovog primera krenućemo počevši od postojećeg projekta (*VI From Template*) dok ćemo u svim narednim primerima uvek kretati od praznog projekta (*Empty Project*). Dakle, u okviru opcije Kreiranje novog projekta (*New*) biramo podopciju kreiranje projekta na osnovu postojećeg projekta *VI From Template* nakon čega dobijamo izgled prozora kao na slici. U ovom prozoru je najpre potrebno odabrati *Generate and Display* projekat a zatim potvrditi izbor klikom na dugme *OK*



1.2 - Prozor u okviru koga se kreiraju novi VI iz generičkih šablona

Nakon odabira projekta, nad kojim želimo da kreiramo naš virtuelni instrument, generišu se dva prozora kao što je prikazano na slici 1.3. Svaki VI kreiran u LabVIEW softverskom razvojnom okruženju sadrži ova dva prozora. Prozor označen kao *Front Panel* zapravo predstavlja interfejs instrumenta prema korisniku, dok drugi prozor, označen sa *Block Diagram*, služi za programiranje virtuelnog instrumenta, tj. kreiranje blok šeme koja implementira funkcionalnost virtuelnog instrumenta. Ukoliko maksimizujemo ova dva prozora tako da ispunjavaju čitavu površinu ekrana ili ako tokom rada slučajno zatvorimo jedan od ova dva prozora, uvek se možemo vratiti na onaj drugi tako što:

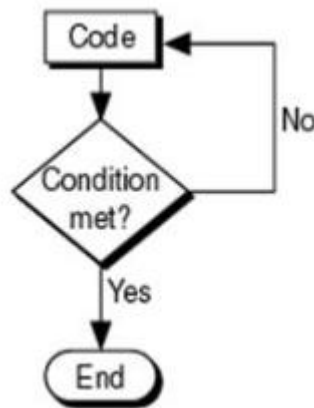
1. Ako se nalazimo u prozoru *Front Panel*, kliknemo na opciju *Window* koja se nalazi u traci sa alatima, a zatim na opciju *Show Block Diagram*
2. Ako se nalazimo u prozoru *Block Diagram*, kliknemo na opciju *Window* koja se nalazi u traci sa alatima, a zatim na opciju *Show Front Panel*



1.3 - Front Panel VI (gore) i Block Diagram VI (dole)

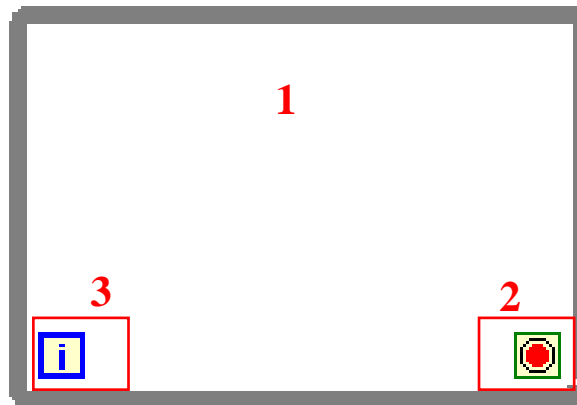
Dakle, nakon što smo otvorili već postojeći projekat, dobili smo dva prozora. U okviru prozora *Front Panel* nalazi se deo pod nazivom **Waveform Graph** koji služi za iscrtavanje signala. Ispod njega, u donjem desnom delu prozora, nalazi se dugme označeno sa STOP koji služi za zaustavljanje programa.

Da bi videli koja funkcionalnost je implementirana u okviru ovog VI potrebno je analizirati blok dijagram ovog instrumenta koji je ilustrovan u okviru prozora *Block Diagram*. U okviru te analize možemo uočiti da se svi blokovi instrumenta nalaze u okviru jedne *while* programske strukture. Iz ranijeg programskog iskustva poznato je da *while* programska struktura realizuje deo programa koji se izvršava sve dok je neki uslov ispunjen. Često ova programska struktura predstavlja glavni program u kome se kôd nekog programa izvršava sve dok korisnik ne prekine izvršavanje tog programa.



1.4 - Programski tok while petlje

Kada govorimo o implementaciji ove programske strukture u LabVIEW programskom paketu, ova struktura ima sledeći izgled:



1.5 - Izgled While loop programske strukture u LABView softverskom paketu

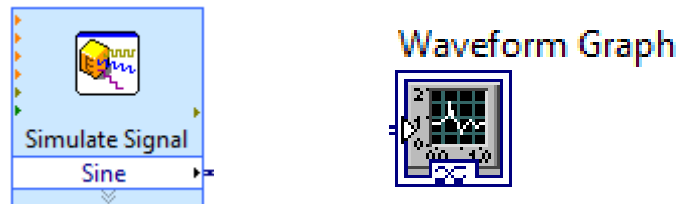
1. Deo u okviru koga se isplementira funkcionalnost koja treba da se izvršava u *while* petlji
2. Uslovni terminal na koji se dovodi uslov koji definiše do kada će se izvršavati funkcionalnosti smeštene u okviru *while* petlje. Drugim rečima, sve dok je logički uslov koji je doveden na ovaj terminal ispunjen, izvršavaće se funkcionalnost u *while* petlji. Tip signala koji se dovodi na ovaj terminal mora biti **boolean**
3. Iteracioni terminal čiji izlaz sadrži vrednost koja predstavlja broj trenutno izvršenih iteracija u petlji. Tip ove vrednosti je **decimal**

Ako uporedimo opštu strukturu *while* petlje sa konkretnom implementacijom ove petlje u prethodno pokrenutom projektu, videćemo da se uslovljenost izvršavanja *while* petlje definiše tasterom STOP. Dakle, nakon što se startuje izvršavanje programa, zaustavljanje izvršavanja vrši se pritiskom na taster STOP. Taster STOP, zaporavo realizuje promenljivu tipa **boolean**. Taster je u okviru *Front Panela* apstrahovan jednim grafičkim elementom prikazanim na slici 1.6 desno, dok se u okviru *Block Diagram* taster apstrahuje blokom prikazanim na slici 1.6 levo.



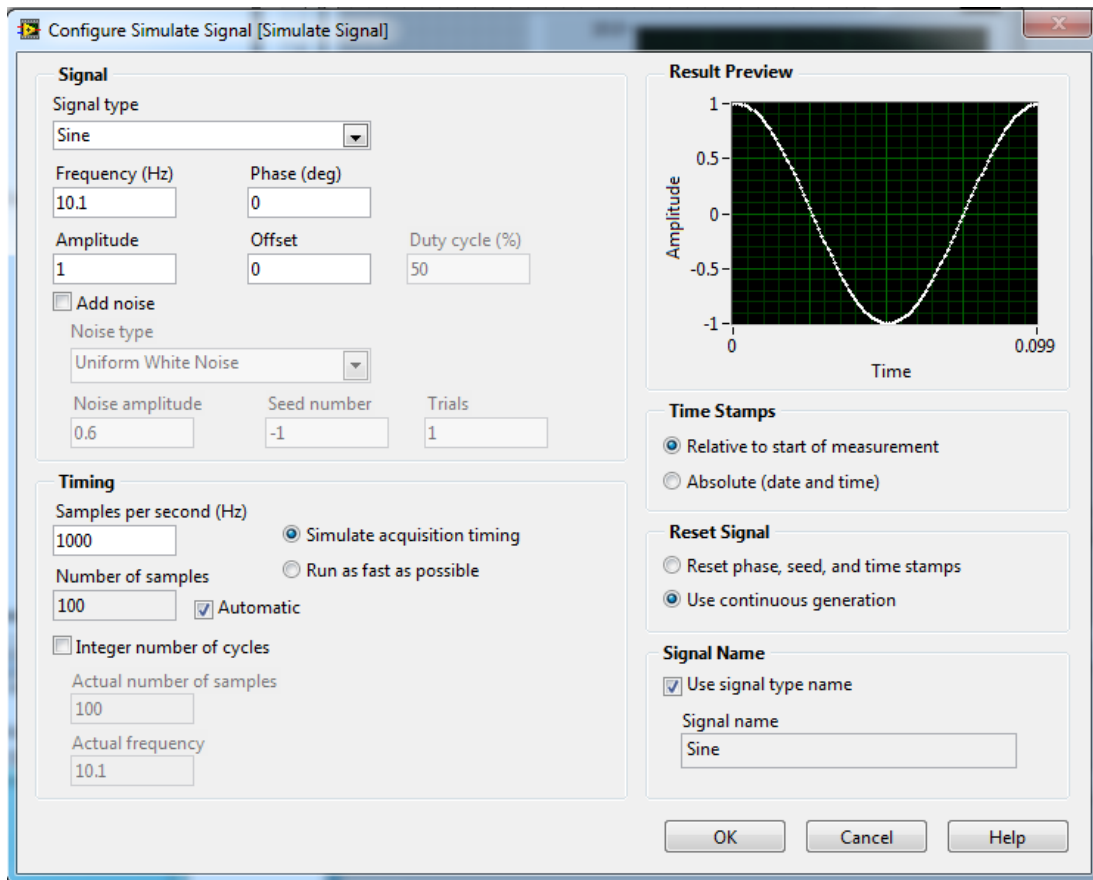
1.6 - STOP kontrola

Iako smo do sada objasnili kako se zaustavlja izvršavanja programa, logično bi bilo objasniti kako se program pokreće. Međutim, pre nego što dođemo do tog koraka potrebno je objasniti ulogu dva bloka koji se nalaze u okviru *While* petlje a koji su prikazani na slici 1.7.



1.7 - Simulate Signal blok (levo) i Waveform graph (desno)

**Simulate signal** blok implementira funkcionalnost koja generiše prostoperiodične signale različitih oblika, amplituda, ofseta i frekvencija. Izlaz ovog bloka predstavlja prostoperiodičan signal sa prethodno definisanim parametrima. Jedan od pristupa za definisanje ovih parametara predstavlja da unošenje ovih vrednosti posredstvom **Properties** prozora. Ovaj prozor se generiše nakon desnog klika na *Simulate signal* blok i ima izgled prikazan na slici 1.8.

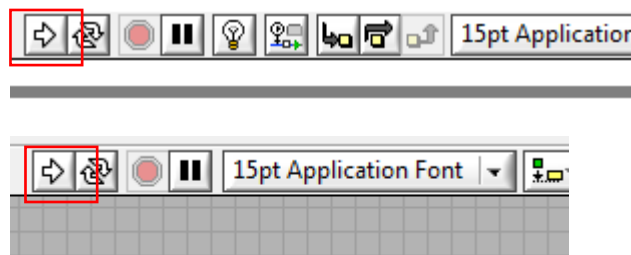


1.8 - Simulate signal blok podešavanja

U okviru ovog prozora mogu se podesiti razni parametri signala ali za nas su bitni sledeći parametri:

1. **Signal Type** – služi za odabir oblika prostoperiodičnog signala
2. **Frequency** – podešavanje frekvencije
3. **Amplitude** – podešavanje amplitude prostoperiodičnog signala
4. **Phase** – podešavanje faznog pomeraja signala
5. **Offset** – podešavanje srednje vrednosti generisanog signala

Nakon podešavanja ovih parametara potrebno je iscrtaati prostoperiodičan signal sa definisanim parametrima. U tu svrhu se koristi blok pod nazivom **Waveform Graph**, prikazan na slici 1.7, koji implementira funkcionalnost iscrtavanja signala koji se dovodi na njegov ulaz. Više o samim mogućnostima konfigurisanja koje ovaj blok nudi biće rečeno u nastavku ovog primera.

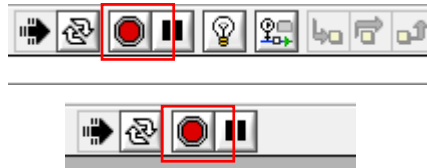


1.9 - Strelice za pokretanje LABView programa

Nakon što smo ukratko objasnili i predstavili sve delove sistema, kao i sitsem u celini, potrebno je startovati izvršavanje programa kreiranog u LabVIEW, tačnije „uključiti“ naš virtuelni instrument. To se

postigne klikom na strelicu okrenutu na desno koja se nalazi izvan blok dijagrama ili na strelicu okrenutu na desno koja se nalazi izvan front panela instrumenta, kao što je prikazano na slici 1.9.

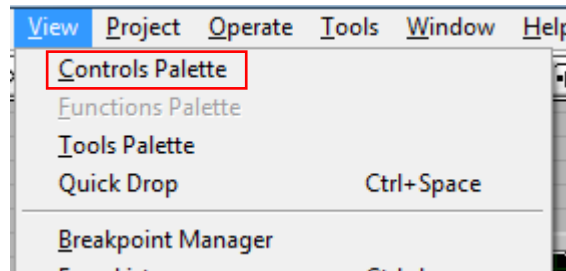
Nakon startovanja programa, uvidom u ponašanje indikatora koji se nalaze u prozoru *Front Panel* možemo zaključiti da ovaj virtuelni instrument iscrtava prostoperiodičan signal koji smo prethodno definisali u okviru *Simulate Signal* bloka. Nakon što smo uvideli implementiranu funkcionalnost, sada se program se može zaustaviti klikom na dugme „STOP“ ili klikom na dugme koje je označeno na slici 1.10.



1.10 - Izgled tastera za zaustavljanje LABView programa

Sada smo došli do dela kada je potrebno da proširimo implementiranu funkcionalnost tako da krenemo ka realizaciji zahtevanog VI koji će imati funkcionalnost generatora prostoperiodičnog signala. Da bi došli do tog cilja, a da bi bolje razumeli pojedine delove sistema, najpre ćemo realizovati funkcionalnost koja omogućava podešavanje parametara prostoperiodičnog signala i matematičke transformacije takvog signala, a zatim ćemo sve to proširiti dodavanjem mogućnosti odabira proizvoljnog oblika signala.

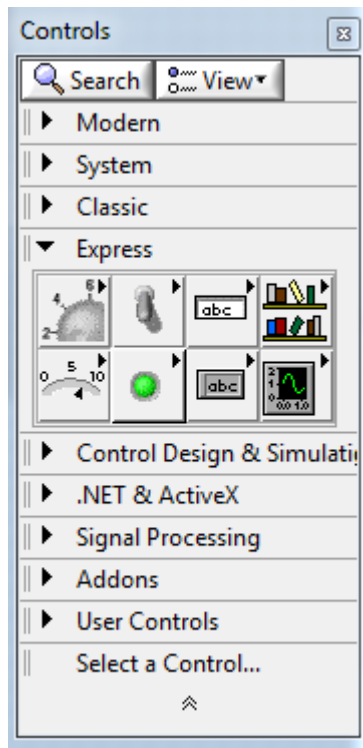
Dakle, najpre implementiramo funkcionalnost koja korisniku pruža mogućnost definisanja porizvoljnih vrednosti parametara prostoperiodičnog signala kao što su amplituda, ofset i frekvencija. Dakle, naš instrument treba da omogući korisniku da unese ove vrednosti. U tu svrhu se koriste numeričke kontrole. Generalno, kada radimo sa Front Panelom, obično taj rad podrazumeva postavljanje nekih kontrola (eng *Controls*). Sve kontrole koje su dostupne u okviru ovog programskog paketa mogu se videti tako što u okviru trake sa menijima, koja se nalazi u okviru *Front Panel* prozora, odaberemo opciju *View* a zatim odaberemo podopciju *Controls Palette* kao što je prikazano na slici 1.11.



1.11 - Meni za izbor kontrola u okviru Front Panel prozora VI

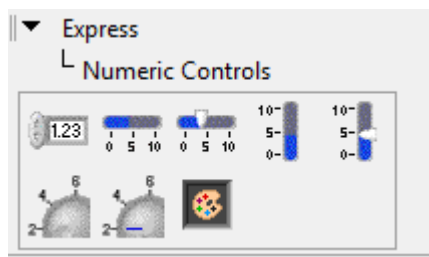


Nakon toga se otvara paleta sa svim kotrolama dostupnim u okviru ovog programskog paketa. Ta paleta je prikazana na slici 1.12.



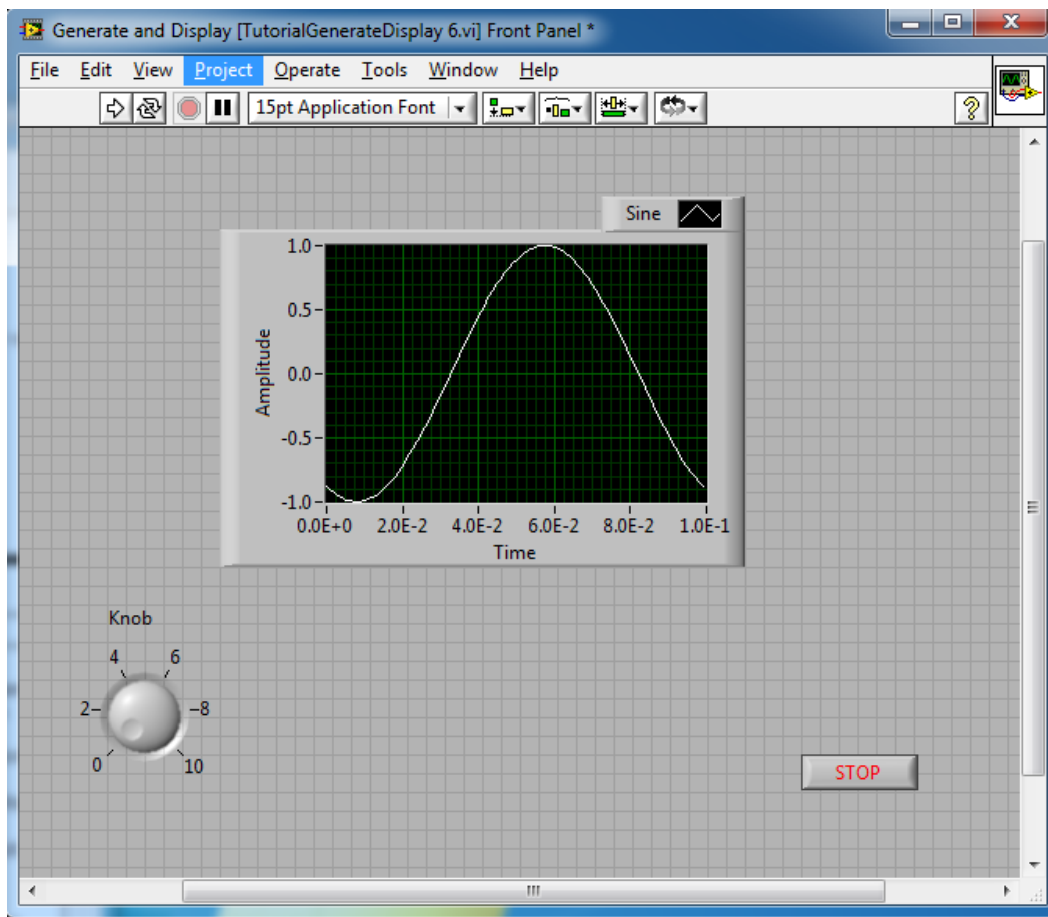
1.12 - Paleta kontrola

Sve kontrole su podeljene u okviru odgovarajućih grupa kontrola. Grupa *Express* obuhvata one kontrole koje se najviše koriste prilikom kreiranja instrumenata. U okviru ove grupe, treba izabrati podgrupu *Numerical Controls* a u okviru nje izabrati kontrolu *Knob*.



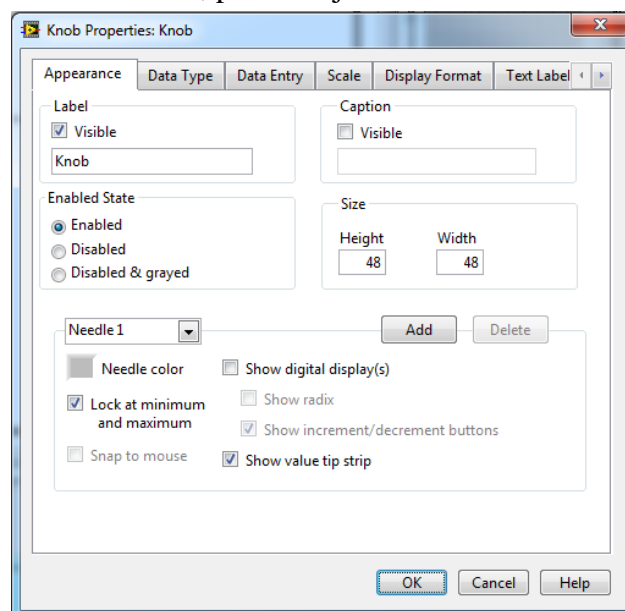
1.13 - Kontrole dostupne u okviru podgrupe Numerical Controls

*Knob* je kontrola koja je grafički predstavljena kao kružni potenciometar. Funkcionalnost koju implementira ova kontrola kao izlaznu veličinu generiše vrednost srazmernu položaju potenciometra. Nakon ubacivanja ove kontrole u *Front Panel* instrumenta dobijamo izgled prednje strane instrumenta kao što je prikazano na slici 1.15.



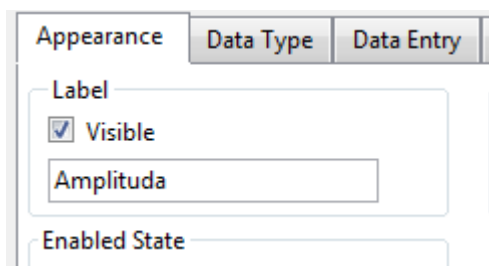
1.15 - Izgled prednje strane VI nakon dodavanja Knob kontrole

Nakon dodavanja kontrole u okviru interfejsa instrumenta, sada je potrebno konfigurirati neke od osobina ove kontrole tako da se što bolje ostvari tražena funkcionalnost. Osobine se konfiguriraju u okviru opcije **Properties** koja je dostupna nakon desnog klika na ovu kontrolu. Sve moguće osobine, kao i izgled prozora za konfiguraciju osobina kontrole, prikazan je na slici 1.14.



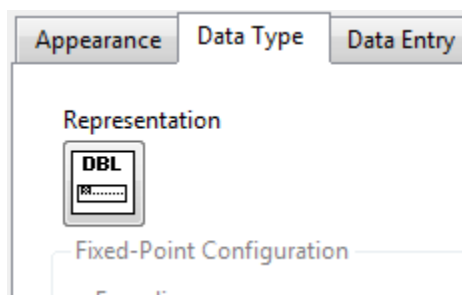
1.14 - Prozor za podešavanje osobina kontrole Knob

Nakon otvaranja opcije **Properties** u okviru kartice **Appearance** najpre dodeljujemo naziv ovom potenciometru. Ova osobina se podešava u okviru dela **Label**. Pošto je ovaj potenciometar namenjen za kontrolu amplitude signala potrebno mu je dodeliti naziv „Amplituda“. Nakon podešavanja, ova kartice izgleda kao na slici 1.16.



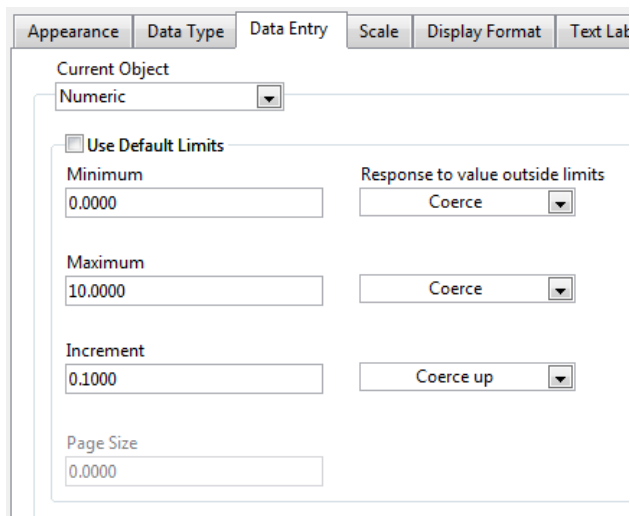
1.16 - Podešavanje naziva potenciometra

U okviru kartice **Data Type** treba odabrati tip podataka koji će se generisati na izlazu bloka koji implementira funkcionalnost potenciometra. Pošto je amplituda kontinualna veličina, tip podataka će biti **Double**. Nakon podešavanja, ova kartica izgleda kao na slici 1.17.



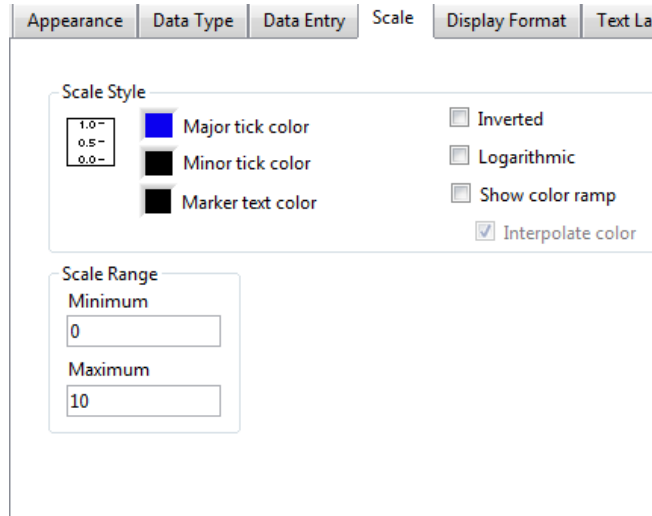
1.17 - Podešavanje tipa podatka na izlazu potenciometra

U okviru kartice **Data Entry** potrebno je podesiti rezoluciju potenciometra kao i njegov opseg. U ovom primeru uzećemo da je opseg amplituda od 0 do 10V gde jedan korak potenciometra iznosi 0.1. Nakon podešavanja odgovarajućih parametara, izgled ove kartice je kao na slici 1.18.



1.18 - Podešavanje opsega ulaznih podataka potenciometra

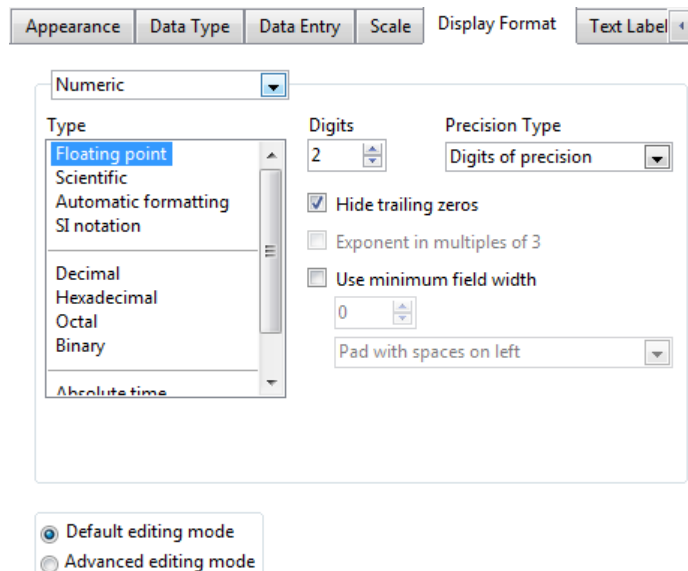
Sada je potrebno podesiti skalu koja se prikazuje pored potenciometra. Ona treba da bude u saglasnosti sa podešenim ulaznim opsegom potenciometra što znači da će skala pored potenciometra prikazivati opseg koji odgovara opsegu ulaznih vrednosti. Nakon podešavanja ova kartica izgleda kao na slici 1.19.



1.19 - Podešavanje opsega potenciometra

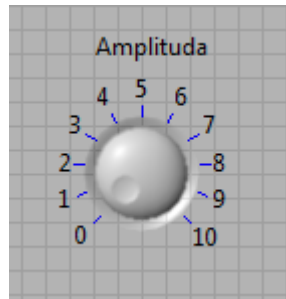
U okviru kartice *Display Format* podešavamo format prikaza brojeva koji se nalaze na skali pored potenciometra. Podesićemo da format brojeva bude *float* i da tačnost prikazivanja bude zaokružena na dve decimale. Nakon navedenih podešavanja, kartica izgleda kao na slici 1.20.

Ovim smo završili podešavanje svih parametara potenciometra koji su nam od interesa u ovom zadatku. Potrebno je potvrditi podešavanja klikom na dugme OK. Nakon toga potenciometar, u okviru Front Panel prozora, izgleda kao na slici 1.20.



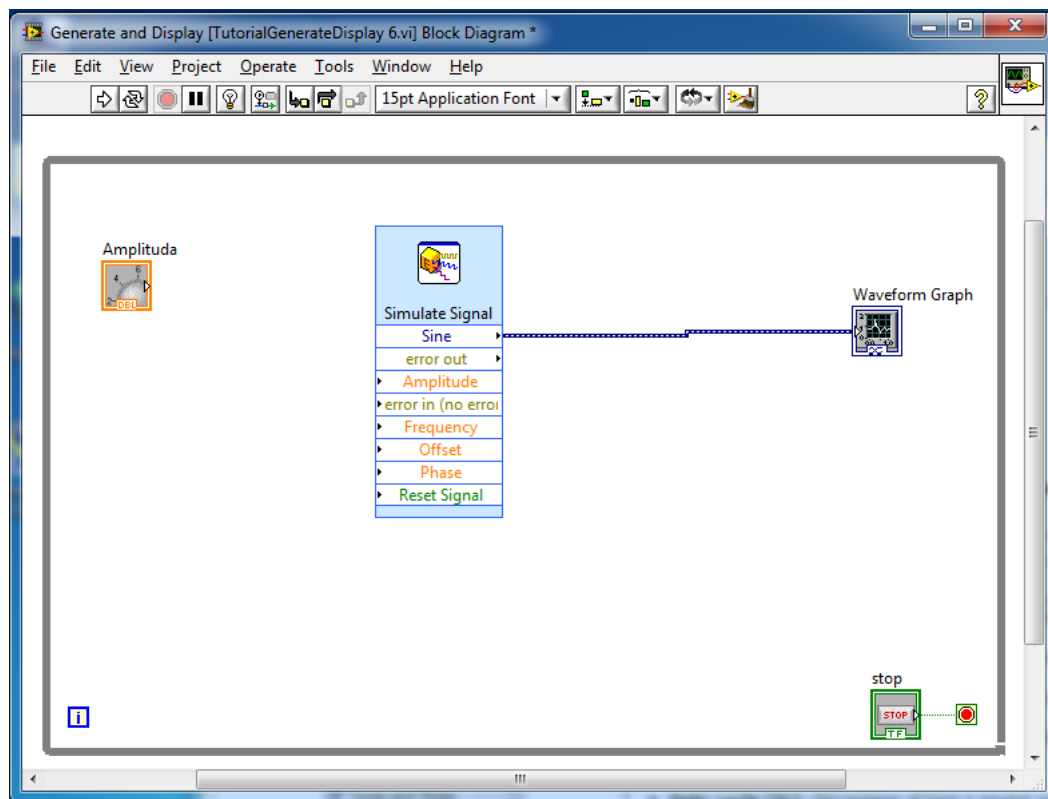
1.20 - Izgled kartice za podešavanje tipa prikazivanja brojeva na skali pored potenciometra

Da sada smo konfigurisali kružni potenciometar kako bi ostvarili željene karakteristike. Sada je potrebno povezati potenciometar kako bi on realizovao željenu funkcionalnost, a to je da se koristi kao potenciometar za podešavanje amplitude signala. Zbog toga se sada vraćamo na prozor *Block Diagram*.



1.21 - Izgled potenciometra u okviru prednjeg panela instrumenta nakon navedenih podešavanja

Nakon povratka u *Block Diagram* prozor i ekspanovanja *Simulate Signal* bloka dobijamo izgled kao na slici



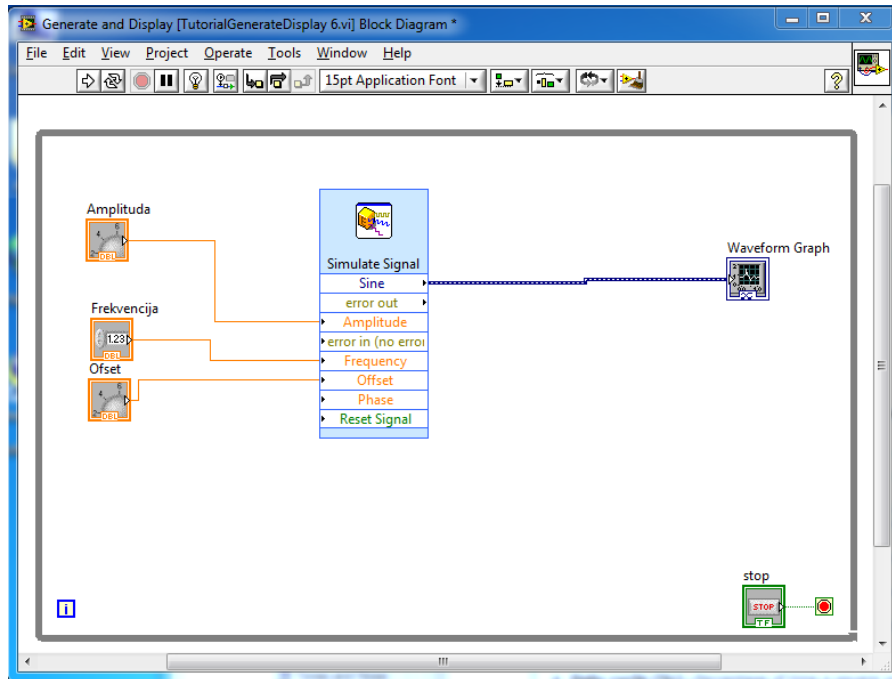
1.22 - Blok dijagram VI nakon dodavanja kružnog potenciometra

Vidimo da se nakon ekspanovanja *Simulate Signal* bloka pojavljuju ulazi koji pre toga nisu bili vidljivi. Ti ulazi omogućavaju da se osobine prostoperiodičnog signala, koje smo ranije definisali u okviru opcije **Properties**, sada definišemo pomoću kontrola koje su korisniku dostupne u okviru prozora *Front Panel*. Da bi to bilo moguće neophodno je odgovarajuću kontrolu povezati na odgovarajući ulaz *Simulate Signal* bloka. U ovom konkretnom slučaju, kontrola „Amplituda“ se povezuje na ulaz *Amplitude*.

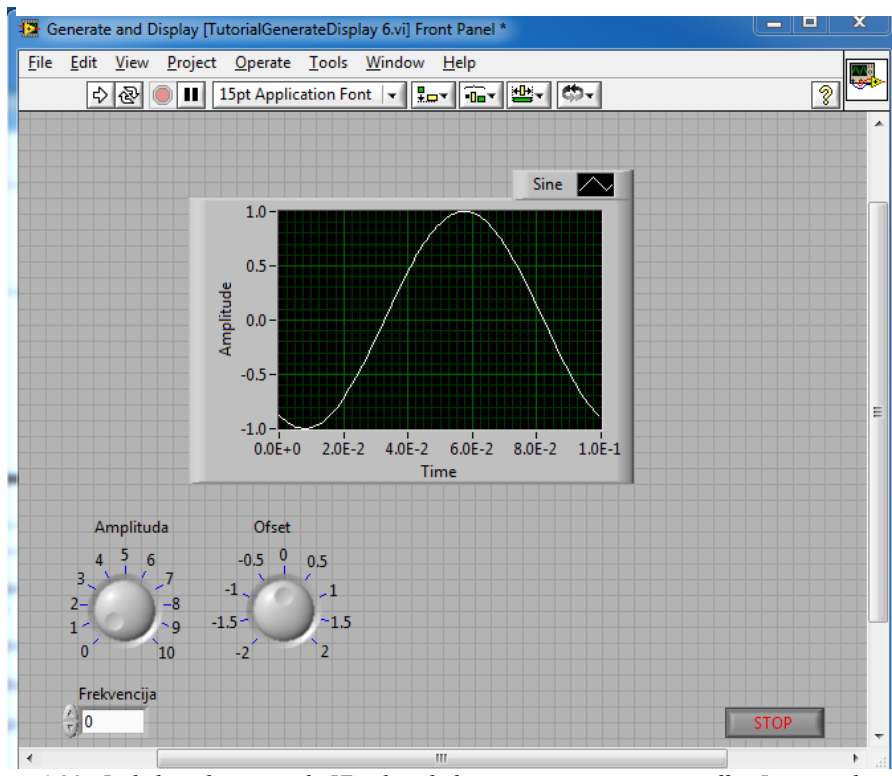
(Potrebno je da student, analogno potenciometru „Amplituda“ kreira i potenciometar „Ofset“ koji će služiti za definisanje srednje vrednosti signala. Ofset treba da se kreće u granicama od -2 do 2 sa korakom 0.1. Pored toga, korisnik treba da kreira i numeričku kontrolu tipa **Numeric Control** koja će služiti za

podešavanje frekvencije signala. Vrednosti koje se mogu podesiti u okviru ove notrole treba da budu u opsegu od 0 do 1kHz sa korakom 0.1Hz.)

Nakon svih traženih zahteva izgled blok dijagrama kao i izgled korisničkog interfejsa virtuelnog instrumenta prikazani su na slikama 1.24 i 1.23

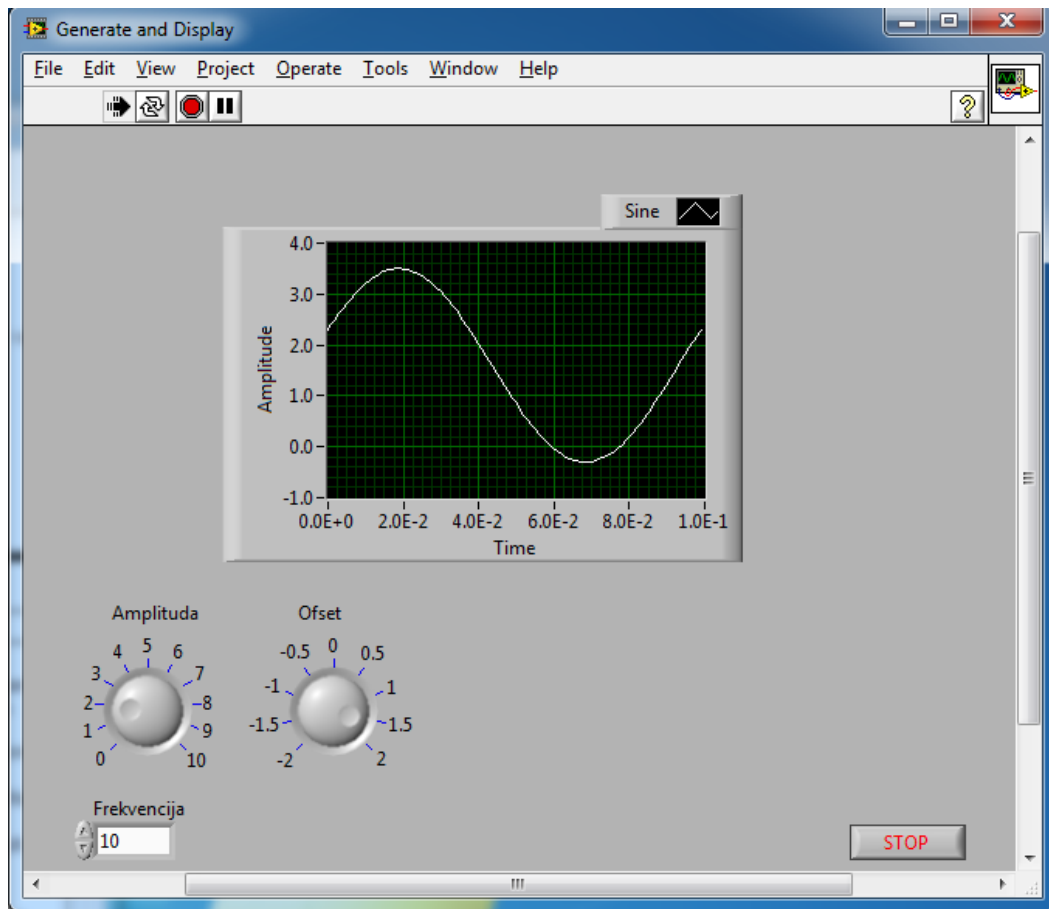


1.24 - Izgled blok dijagrama VI nakon dodavanja potenciometra za offset i numeričke kontrole za frekvenciju



1.23 - Izgled prednjeg panela VI nakon dodavanja potenciometra za offset i numeričke kontrole za frekvenciju

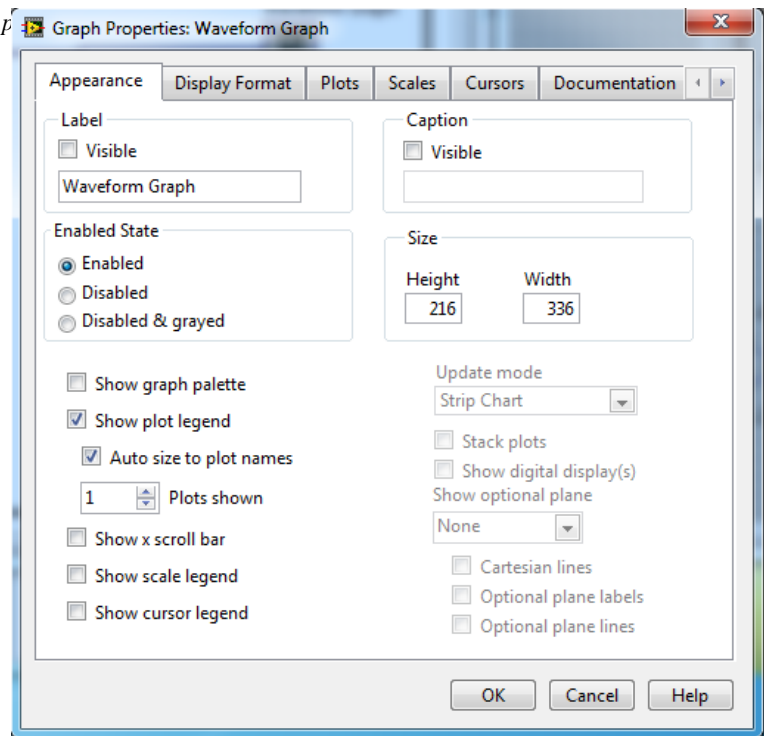
Sada je potrebno pustiti program da se izvršava i testirati implementiranu funkcionalnost. Ukoliko je sve dobro kreirano, dobija se izgled interfejsa kao na slici 1.25.



1.25 - Prednji p

Ako se nakon svih ovih koraka, za položaje potencijometara i unetu numeričku vrednost prikazanu na slici 1.25 dobija izgled i položaj signala kao na istoj toj slici, možemo smatrati da su svi koraci uspešno završeni. Sada je potrebno sačuvati sve što je do sada urađeno u okviru fajla Cas1.vi. To se postiže odabirom opcije **Save All** u okviru menija **File**.

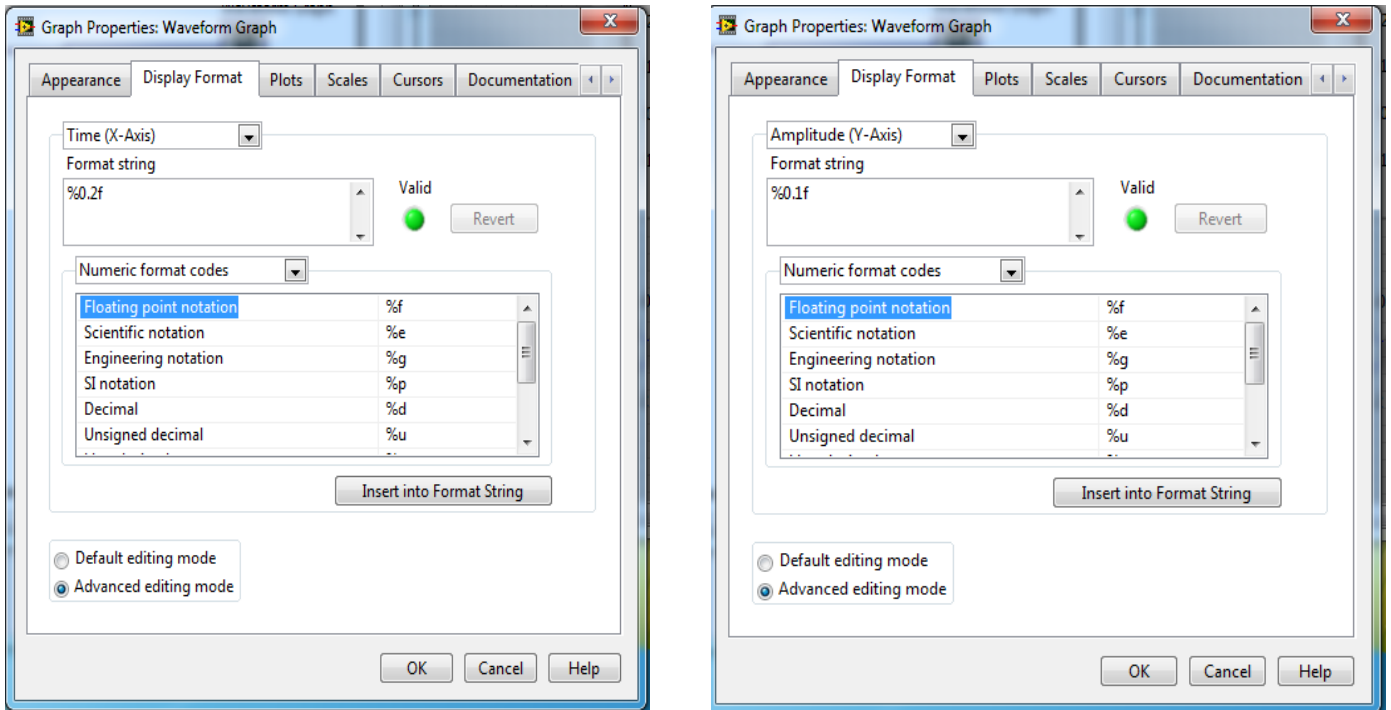
Ako detaljnije pogledamo izgled grafika sa slike 1.25, primetićemo da je veoma teško odrediti karakteristike signala na osnovu tog grafika. To je posledica male veličine grafika ali i loše konfigurisanog opsega duž osa koordinatnog sistema u kome se grafik iscrtava. Dakle, sada je na redu podešavanje osobina vezanih za grafik.



1.26 - Osobine grafika u okviru LABView softverskog paketa

Sve osobine grafika dostupne su nakon desnog klika na grafik, u okviru prozora *Front Panel*, a zatim odabira opcije **Properties**. Nakon toga otvara se prozor kao na slici 1.26.

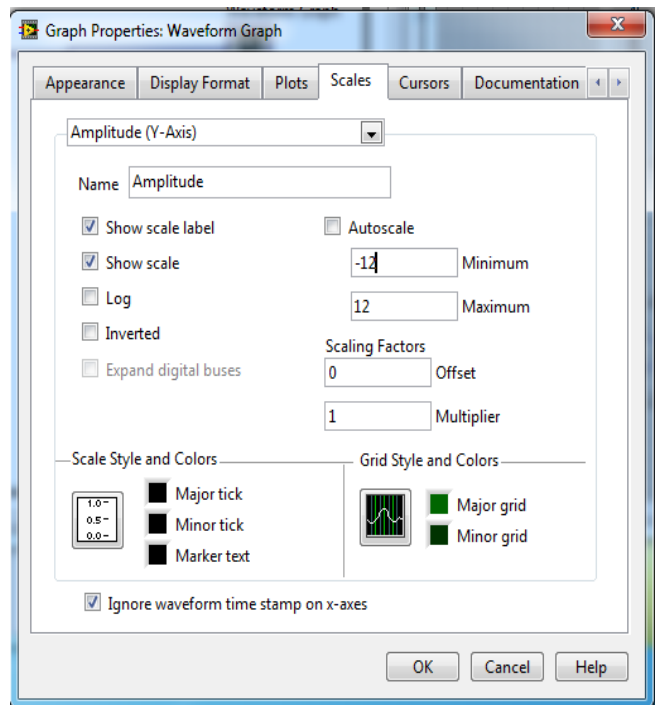
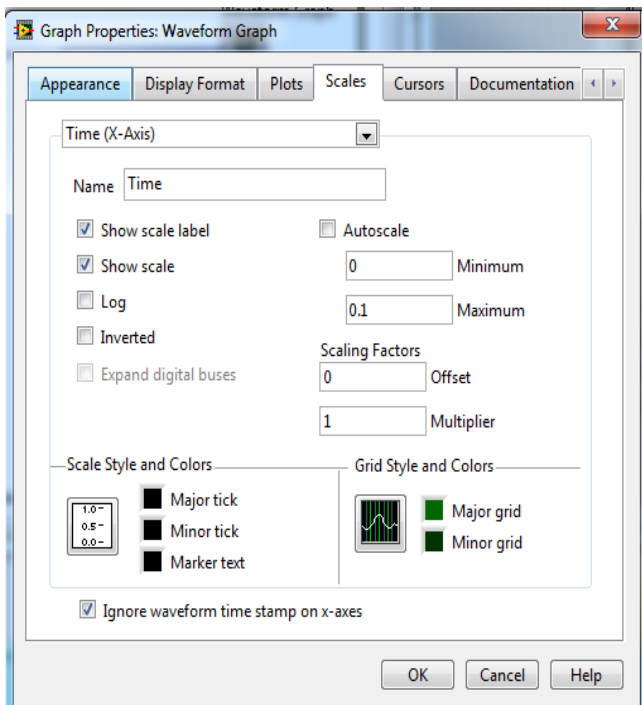
Da bi podesili prikaz brojeva na osama grafika, biramo karticu **Display Format**. Nakon toga najpre odaberemo *X-Axis*, a zatim selektujemo notaciju koju želimo da primenimo na formatiranje brojeva koji se nalaze na X osi. U našem slučaju, želimo da brojevi budu u okviru **Floating Point** notacije sa tačnošću na dve decimale. Nakon ovih podešavanja trebalo bi dobiti izgled kartice kao na slici 1.27 levo. Nakon primene istih podešavanja na Y osu, dobijamo podešavanja u okviru ove kartice kao na slici 1.27 desno.



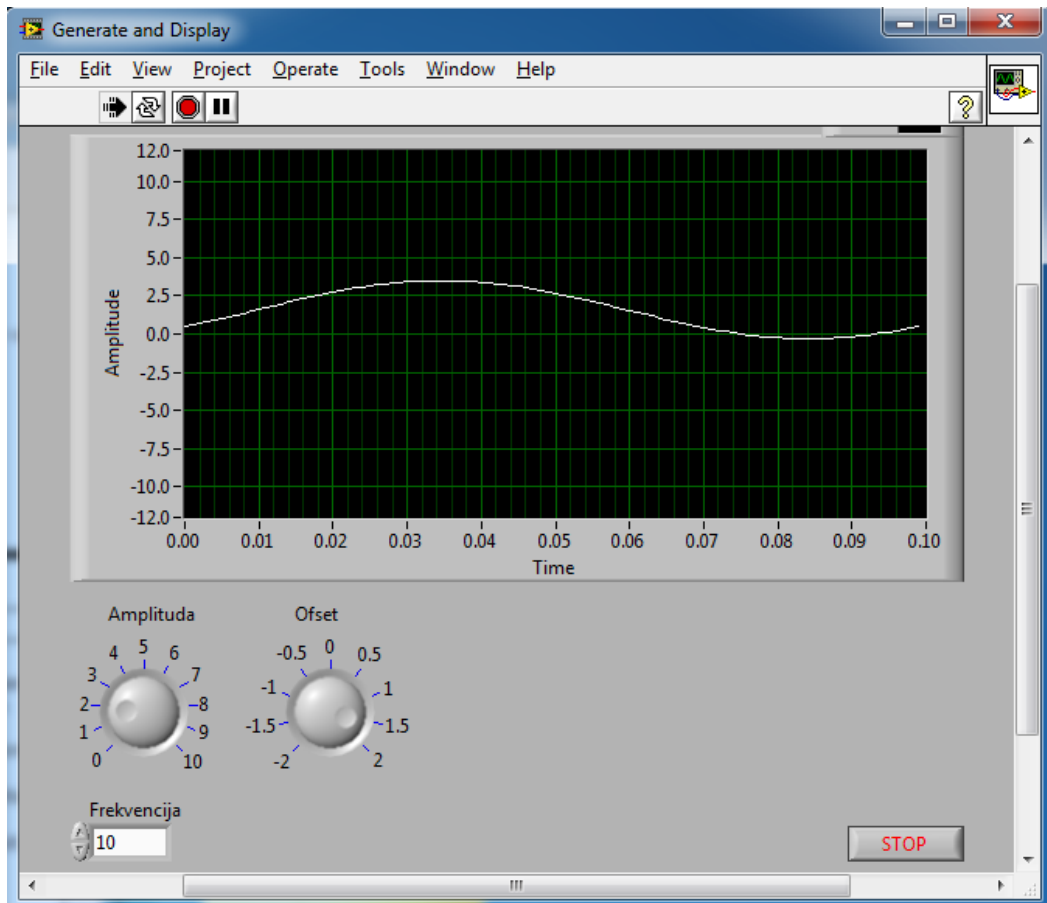
1.27 - Formatiranje prikaza brojeva na X (levo) i Y (desno) osi

Pored podešavanja formata prikaza brojeva na osama, potrebno je podesiti i opsege prikaza na obe ose. To se postiže u okviru kartice **Scales**. U okviru ove kartice najpre treba isključiti *Autoscale* opciju za X osu a zatim i za Y osu. Za X osu treba staviti da je opseg prikazivanja od 0 do 0.1 dok je opseg prikazivanja za y osu od -12 do 12. Nakon primene ovih zahteva dve varijante ove kartice izgledaju kao na slikama





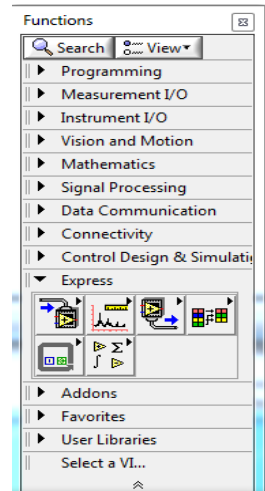
Nakon primene ovih podešavanja i nakon proširivanja veličine grafika, izgled interfejsa VI izgleda kao na slici 1.28.



1.28 - Prednji panel VI nakon podešavanja osa grafika

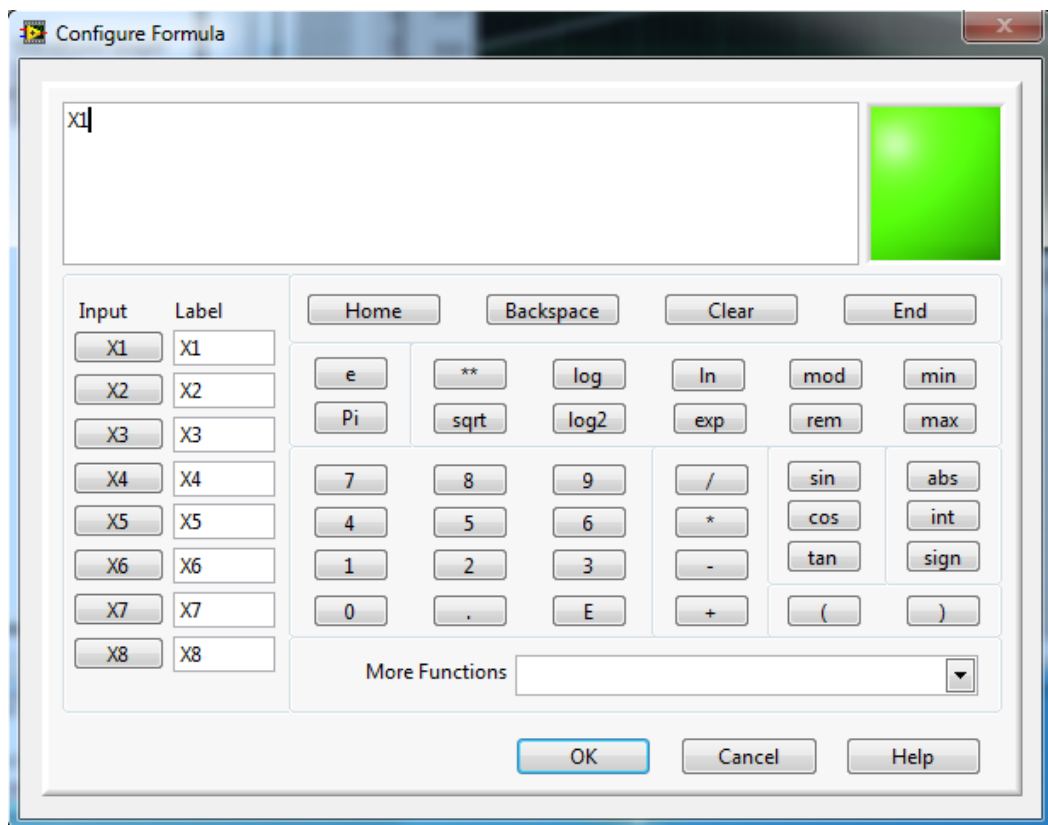
Da bi kompletirali prvu celinu zadatka potrebno je implementirati funkcionalnost koja realizuje određene matematičke transformacije nad signalom. Konkretno, u ovom primeru najpre ćemo implementirati funkcionalnost koja određuje apsolutnu vrednost odabranog signala i takav signal prikazuje na grafiku.

Kao blok koji će nam pomoći pri realizaciji ove funkcionalnosti, koristimo blok **Formula** koji je dostupan u okviru palete blokova koja se nalazi u *Block Diagram* prozoru. Da bi prikazali ovu paletu, potrebno je u okviru menija **View** odabrati opciju **Functions Palette**. Nakon toga, pojavljuje se prozor kao na slici 1.29.



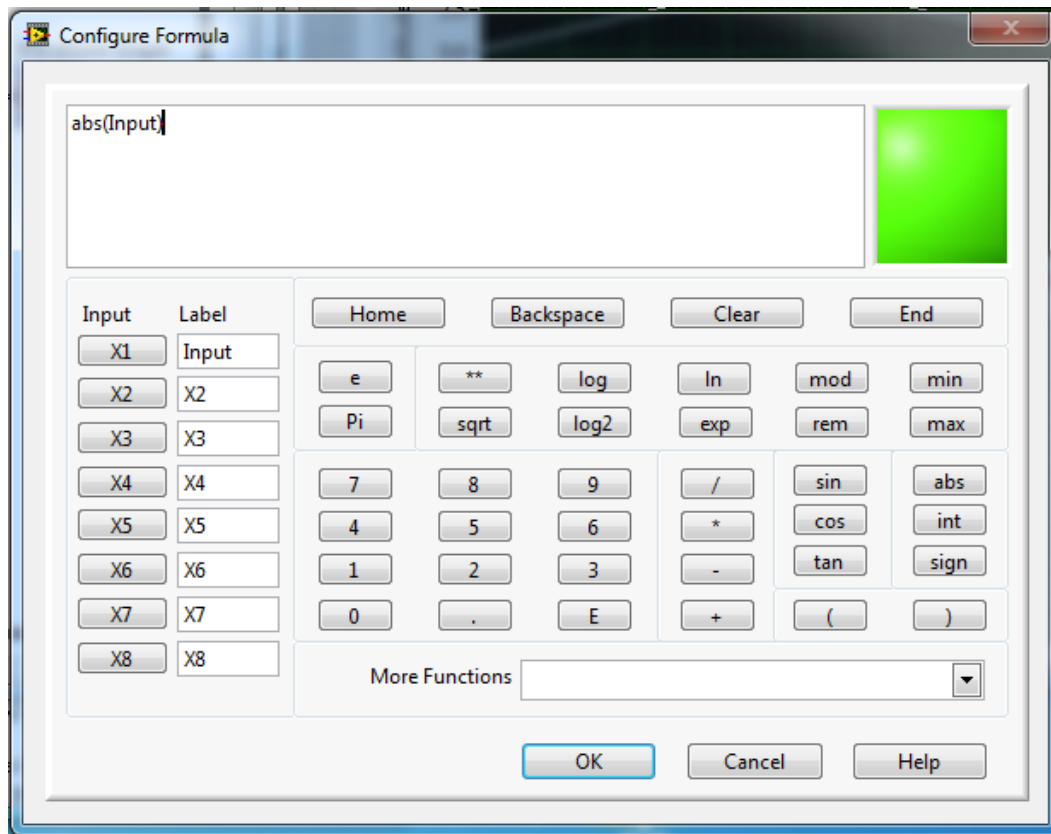
1.29 - Paleta funkcija

Sada je u okviru kategorije **Matematsics** neophodno odabrati podkategoriju **Scripts & Formulas** a u okviru te podkategorije odabrati blok pod nazivom **Formula**. Nakon odabira ovog bloka, potrebno ga je prevuci u okvir *while* petlje u prozoru *Block Diagram* našeg VI. Nakon toga otvara se prozor *Configure Formula* u okviru koga je potrebno napisati formulu koja će se primeniti nad ulaznim signalom definisanim kao promenljiva X1. Izgled ovog prozora prikazan je na slici . (Isti ovaj prozor može se otvoriti tako što najpre kliknemo desnim klikom miša na blok **Formula** a zatim odaberemo opciju **Properties**)



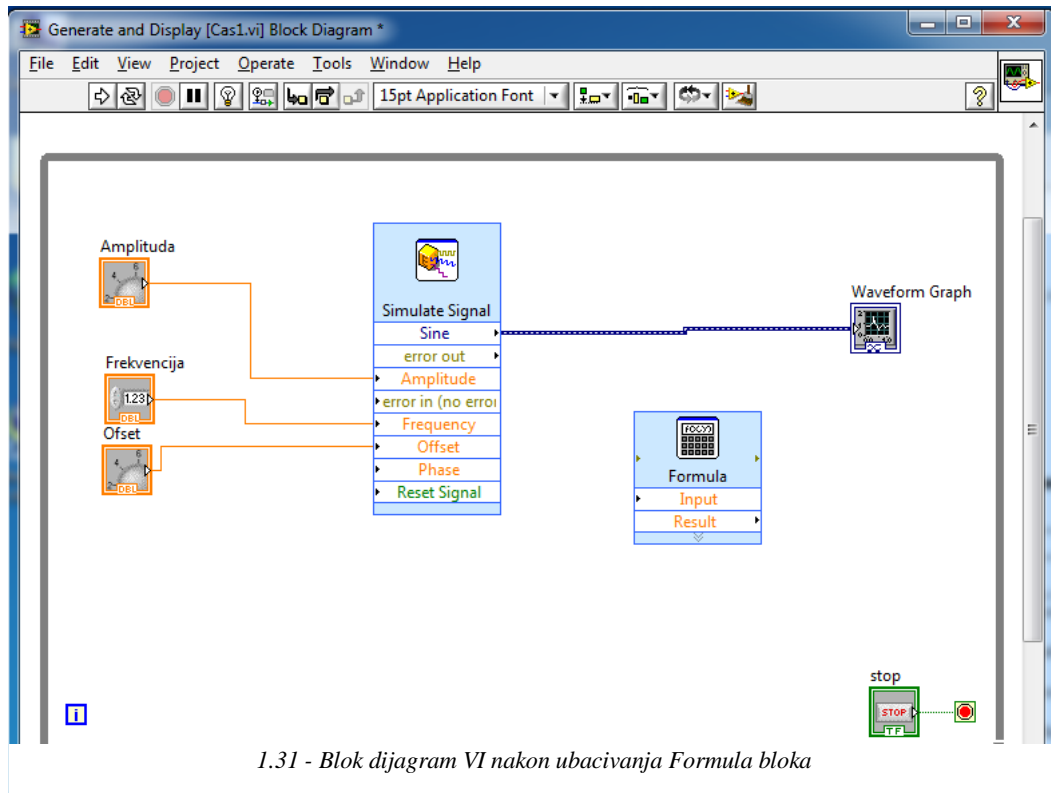
1.30 - Izgled prozora u okviru koga se unosi formula koja će se primeniti nad signalom koji se dovodi na ulaz X1

U okviru ovog prozora najpre ćemo promeniti naziv signala iz X1 u *Input* što se postiže promenom u okviru odgovarajućeg **Label** polja a nakon toga ćemo napisati fomulu koju treba primeniti nad ulaznim signalom. Nakon ovih promena, stanje u okviru prozora **Configure Formula** je kao na slici 1.32.



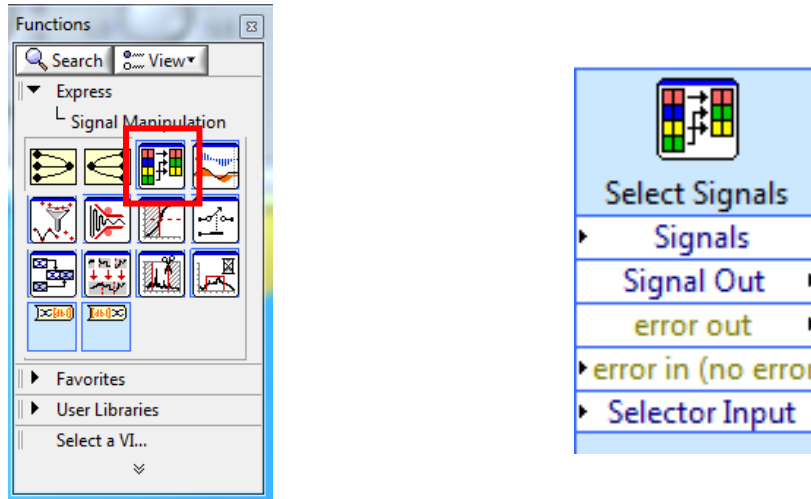
1.32 - Podešavanje Formula bloka da na svom izlazu generiše apsolutnu vrednost signala na ulazu

Klikom na dugme OK potvrđujemo da smo završili unos formule koju treba primeniti nad signalom kojeg dovodimo kao ulazni signal u blok **Formula**. Nakon potvrde na dugme OK dobijamo izgled blok dijagrama kao na slici 1.31.



1.31 - Blok dijagram VI nakon ubacivanja Formula bloka

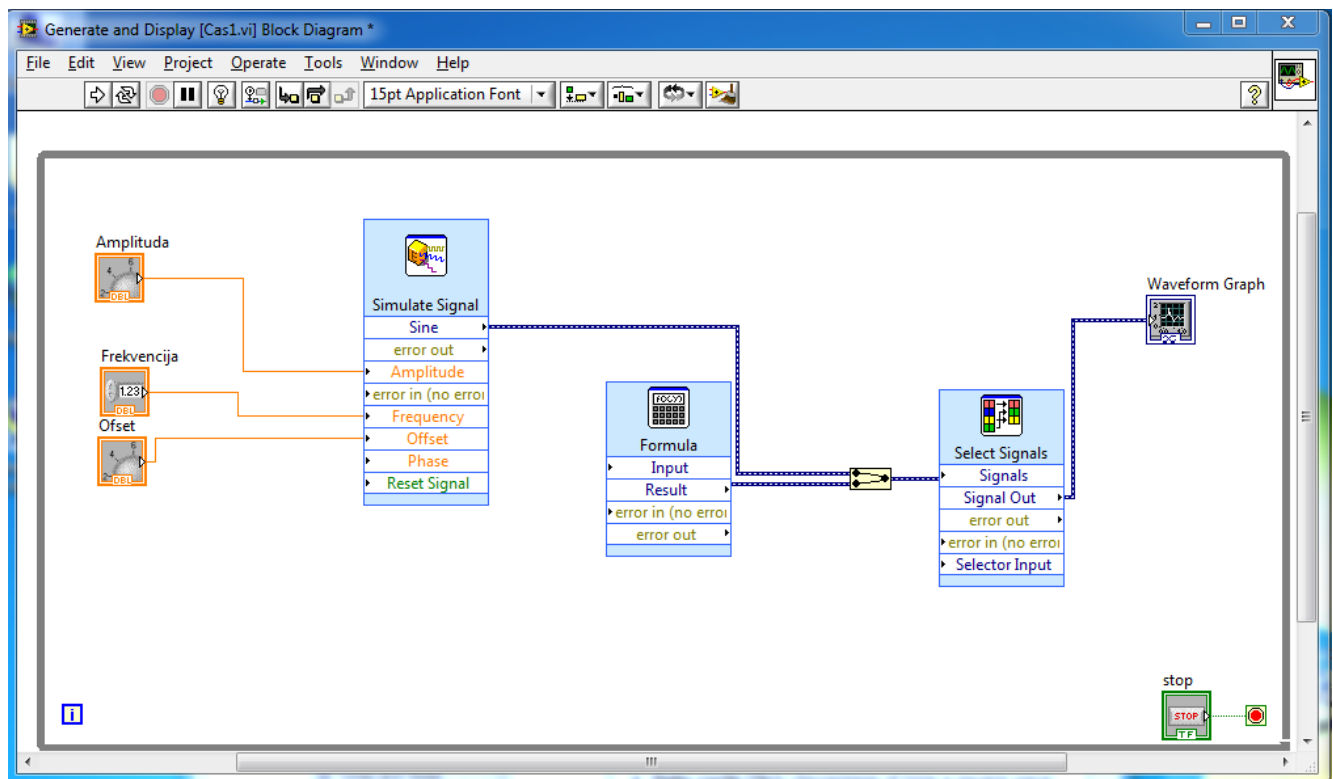
Sada je potrebno osmisliti logiku koja korisniku nudi mogućnost da bira da li će mu se prikazivati originalni signal ili signal nad kojim je primenjena neka matematička operacija (u ovom slučaju apsolutna vrednost signala). U tu svrhu najpre je, u okviru blok dijagrama sistema, potrebno uvesti blok koji se naziva **Select Signal**. Ovaj blok se nalazi u okviru kategorije **Express**, podkategorije **Signal Manipulation**. Pozicija ovog bloka u okviru palete funkcija, kao i izgled bloka nakon dovlačenja u blok šemu sistema, prikazani su na slici 1.33.



1.33 - Izgled bloka za selekciju signala u okviru palete funkcija (levo) i u okviru blok dijagrama VI (desno)

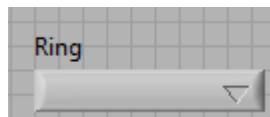
Blok **Select Signals** se ponaša kao multiplekser gde se na ulaz **Signals** dovodi više signala a na izlazu **Signal Out** se propušta samo onaj signal koji je definisan vrednošću na ulazu **Selector Input**. Ono što je karakteristično za ovaj blok je to da se na ulaz **Selector Input** dovodi binarni niz gde vrednost **True** na odgovarajućoj poziciji u nizu definiše koji signal sa ulaza će biti vidljiv na izlazu. Drugim rečima, svakom signalu na ulazu se dodeljuje odgovarajući redni broj, a ako želimo da signal sa odgovarajućim rednim brojem prosledimo na izlaz potrebno je formirati binarni niz koji na poziciji indeksa koji odgovara rednom broju signala, ima vrednost **true**.

Dakle, na ulaz bloka **Select Signals** dovodimo sve signale koje želimo da multipleksiramo. U ovom slučaju to su originalni i matematički modifikovan signal. Sada blok šema sistema, nakon dovođenja oba signala na ulaz bloka **Select Signals**, izgleda nao na slici 1.35.



1.35 - Blok šema VI nakon dovođenja oba signala na Select Signals blok

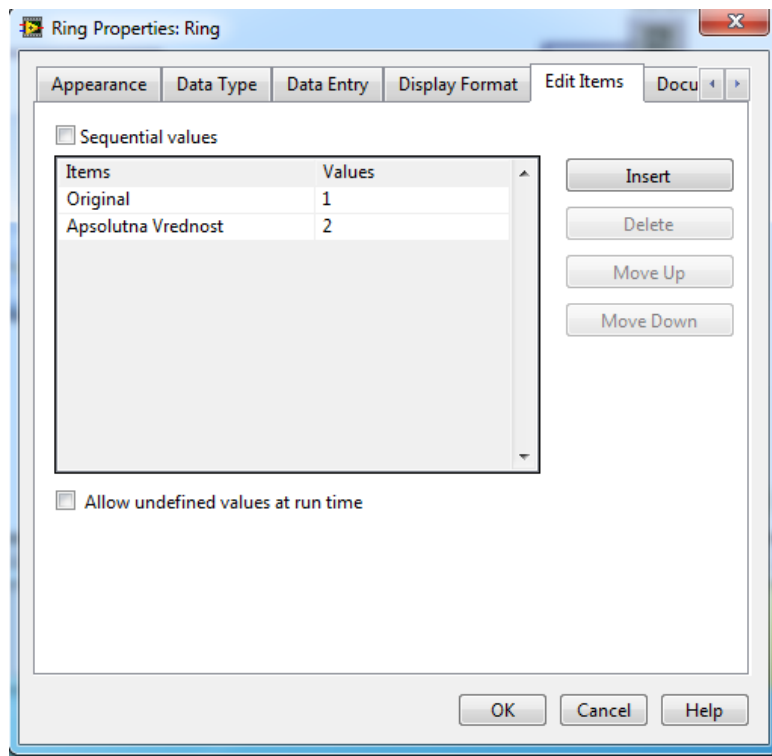
Sada je potrebno realizovati selektor koji će se koristiti za izbor između prikaza originalne i modifikovane vrednosti signala. Da bi realizovali ovaj selektor najpre je neophodno u okviru prozora **Front Panel** dodati kontrolu **Menu Ring** koja se nalazi u okviru kategorije **Express** a podkategorije **Text Controls**. Ova kontrola nakon dodavanja u okviru **Front Panel** Prozora izgleda kao na slici



1.34 - Ring Control u okviru Front Panel-a VI

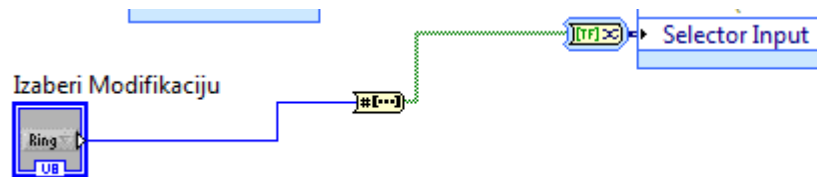
Potrebno je modifikovati podešavanja ove kontrole kako bi ostvarili traženu funkcionalnost. Na osnovu modifikacija prethodnih kontrola, treba uraditi sledeće modifikacije u okviru podešavanja ove kontrole:

1. U odgovarajućoj kartici promeniti ime kontrole u „Izaberi Modifikaciju“
2. U odgovarajućoj kartici promeniti da tip podatka na izlazu iz bloka koji odgovara ovoj kontroli bude **Unsigned Int 8** (U8)
3. U odgovarajućoj kartici promeniti opseg da bude u granicama od 1 do 2 sa korakom 1
4. U odgovarajućoj kartici podesiti da format prikaza brojeva treba da bude **Decimal**
5. U okviru kartice **Edit Items** kreirati dve tekstualne vrednosti „Apsolutna vrednost“ i „Original“. Potrebno je realizovati da tekstualnoj vrednosti „Apsolutna vrednost“ bude dodeljen broj 2 a tekstualnoj vrednosti „Original“ bude dodeljen broj 1. Nakon ovih izmena sadržaj u okviru kartice **Edit Items** izgleda kao na slici 1.37. Na ovaj način smo realizovali funkcionalnost koja će na izlazu bloka **Menu Ring** generisati broj 1 kada korisnik selektuje „Original“ dok će u slučaju da korisnik selektuje „Apsolutna vrednost“ na izlazu ovog bloka biti generisan broj 2.



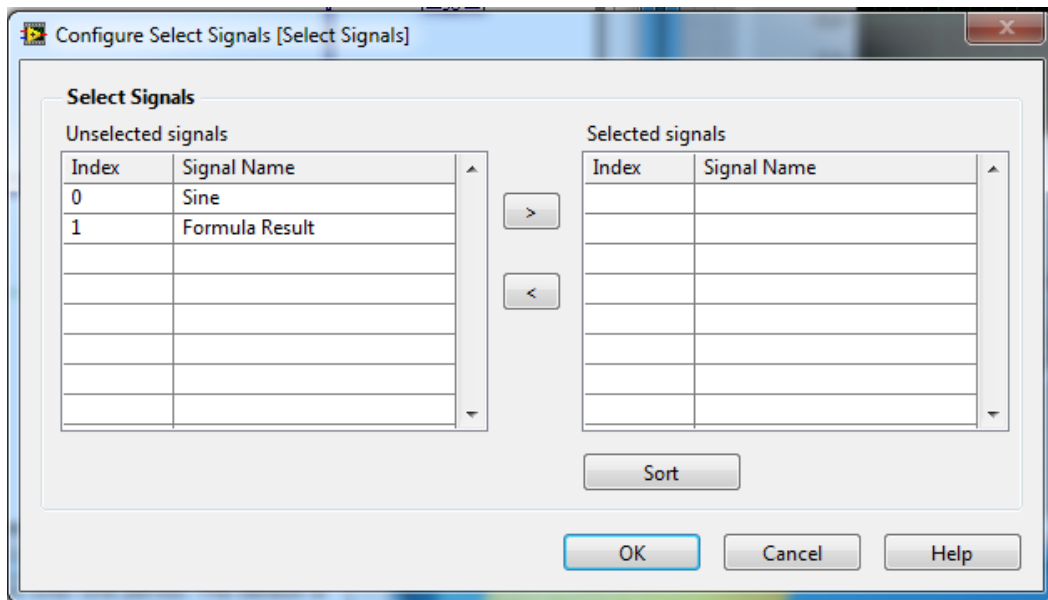
1.37 - Dodeljivanje brojeva odgovarajućim tekstualnim vrednostima

Sada je potrebno blok **Menu Ring** povezati na ulaz **Selector Input** bloka **Select Signals**. Međutim, zbog toga što je izlaz **Menu Ring** bloka decimalna vrednost, a ulaz **Selector Input** prima niz binarnih vrednosti, potrebno je izvršiti konverziju signala pre povezivanja ova dva bloka. Konverzija se postiže pomoću bloka **Number to Boolean Array** koji se nalazi u podkategoriji **Boolean** kategorije **Programming**. Nakon povezivanja, ovaj deo blok dijagrama izgleda kao na slici 1.36.



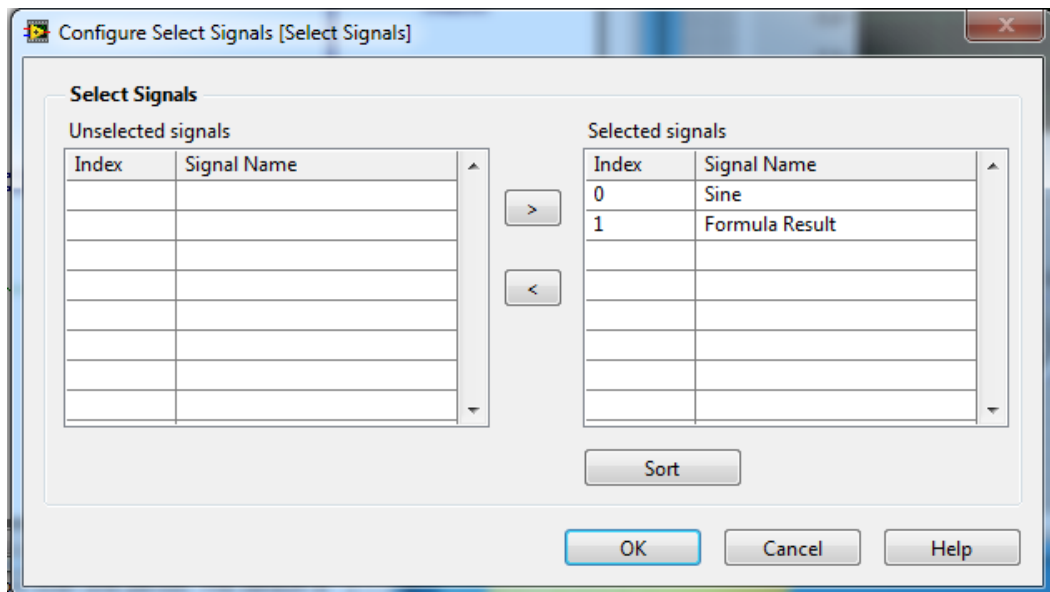
1.36 - Deo blok dijagrama VI koji ilustruje način povezivanja kontrole Ring sa ulazom Selector Input bloka Select Signals

Međutim, nije dovoljno samo povezati **Menu Ring** sa **Select Signals** blokom da bi se ostvarila funkcionalnost selektovanja odgovarajućeg signala. Pored ostvarene veze potrebno je i definisati redne brojeve koji signalu, koji se dovode na ovaj blok, imaju na ulazu. Ti redni brojevi omogućavaju da se za odgovarajuću binarnu vrednost na ulazu **Selector Input** na izlazu **Select Signals** bloka propusti odgovarajući signal, tj. signal čiji redni broj odgovara vrednosti na ulazu **Selector Input**. Da bi definisali ove redne brojeve potrebno je otvoriti **Properties** opciju koja se pojavljuje nakon desnog klika na blok **Select Signals**. Prozor opcije **Properties** ovog bloka izgleda kao na slici 1.39.



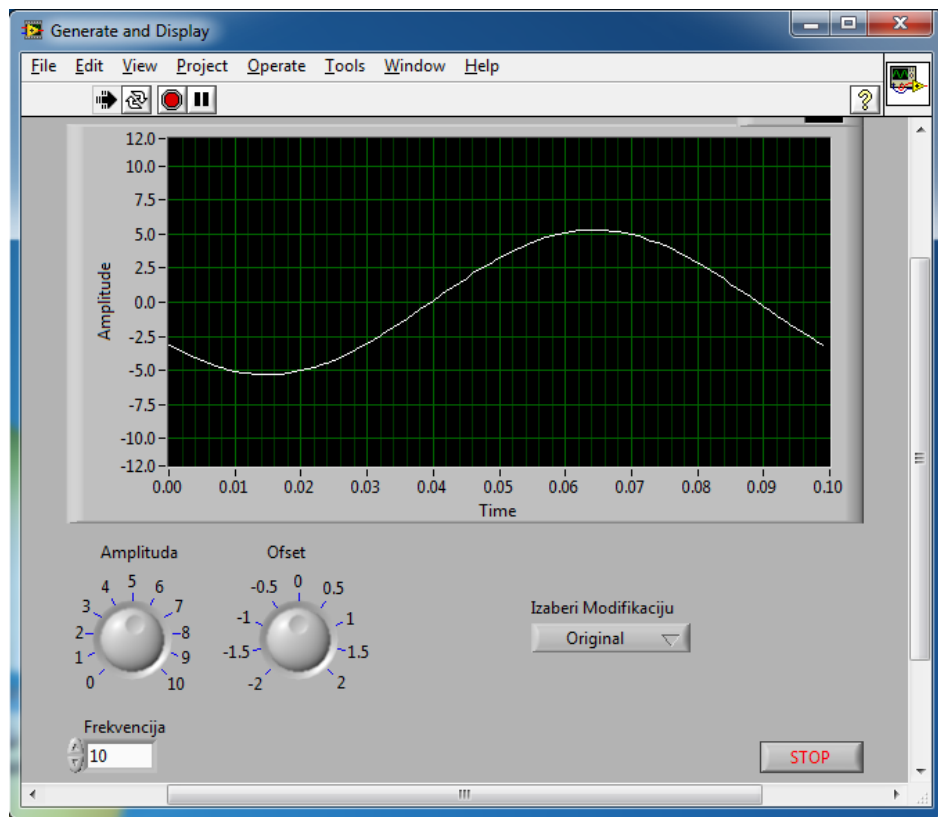
1.39 – Podešavanje Select Signals bloka kojim se signalima na ulazu ovog bloka dodjeljuje odgovarajući redni broj

Sada je potrebno u okviru tabele označene sa **Unselected signals**, kliknuti na signal Sine a zatim na dugme „>“ nakon čega se signal Sine pojavljuje u okviru tabele **Selected signals**. Istu akciju treba ponoviti i za signal **Formula Result**. Nakon toga se dobija izgled ovog prozora kao na slici 1.38.

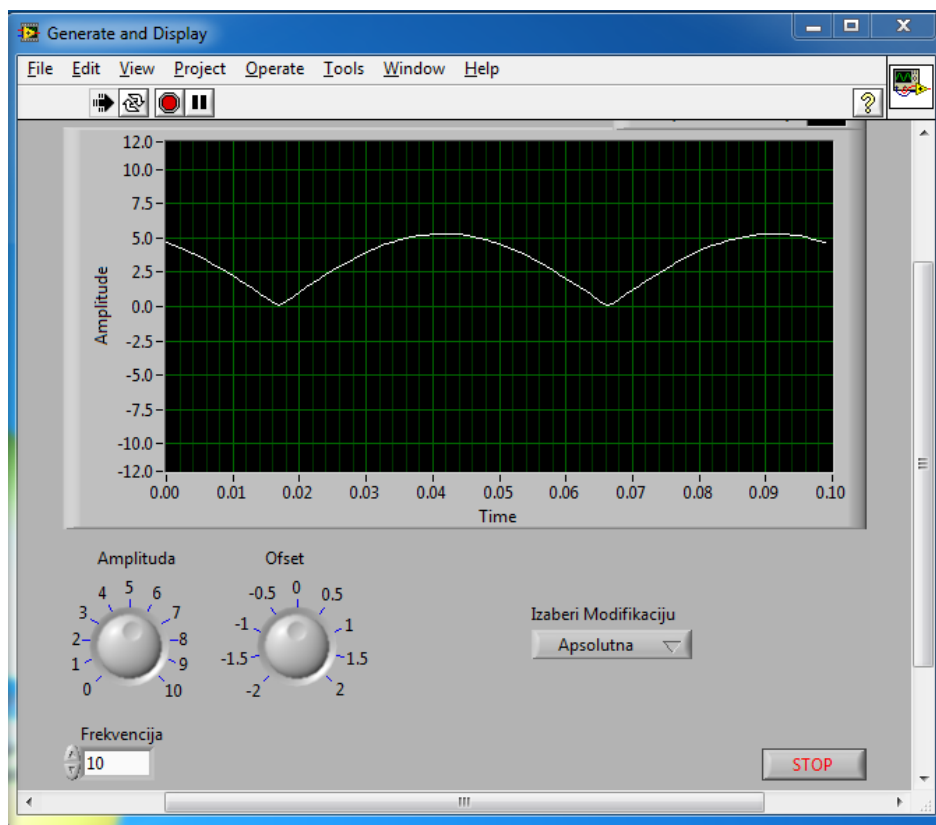


1.38 - Dodeljivanje rednog broja signalima na ulazu Select Signals bloka

Kolona **Index** označava redni broj elementa u binarnom nizu koji se dovodi na ulaz **Selector Input** koji je zadužen za kontrolu signala koji se nalazi pored tog indeksa. Sada je potrebno potvrditi selekciju klikom na dugme OK i startovati izvršavanje programa. Ukoliko je sve dobro povezano, u okviru Front Panel prozora biće omogućeno generisanje dva tipa signala kao što je prikazano na slikama



1.41 - Stanje prednjeg panela VI nakon odabira prikaza originalnog signala



1.40 - Stanje prednjeg panela VI nakon odabira prikaza signala nad kojim je primenjena odgovarajuća matematička modifikacija



Zadaci za samostalni rad:

- 2.1. Potrebno je modifikovati prethodno kreirani VI tako da se pored odabira sinusnog oblika signala omogući generisanje i prikaz periodičnih signala oblika rampe, testere i pravougaonog impulsa. Nad signalima ne treba primenjivati nikakve matematičke funkcije.
- 2.2. Potrebno je modifikovati zadatak 2.1. tako da se pored funkcionalnosti koja nudi izbor različitih oblika prostoperiodičnih signala korisniku omogući i izbor matematičkih funkcija koje treba primeniti na signale.