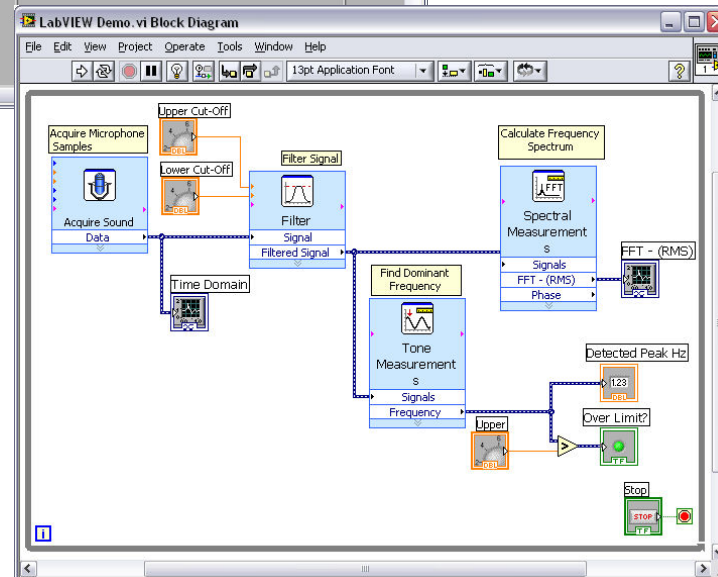
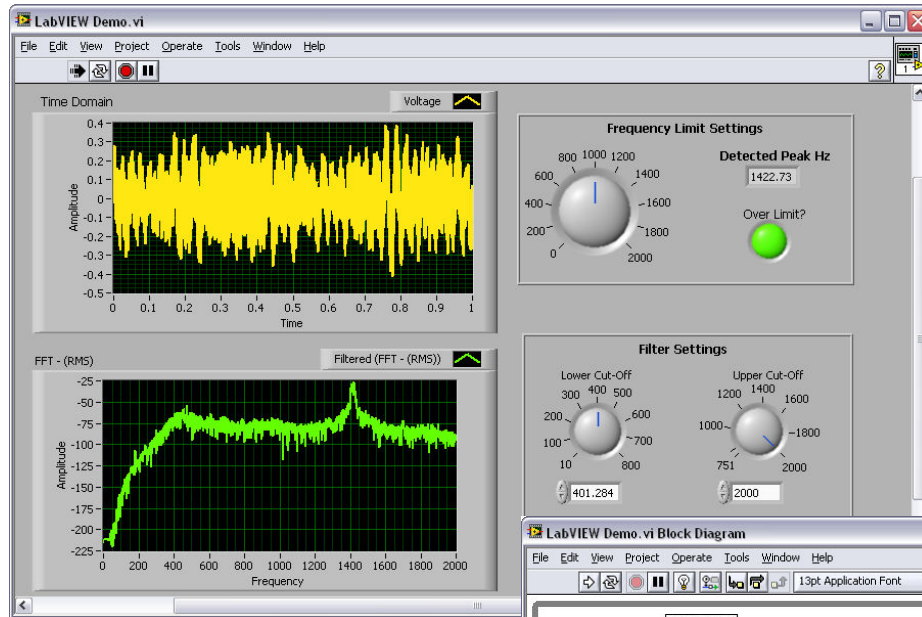




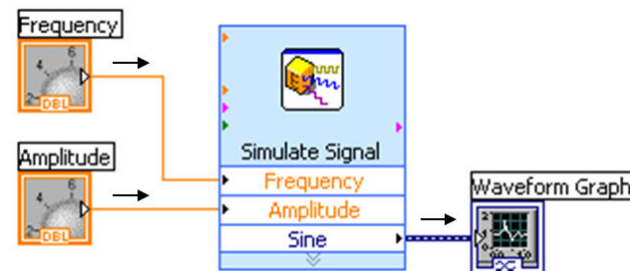
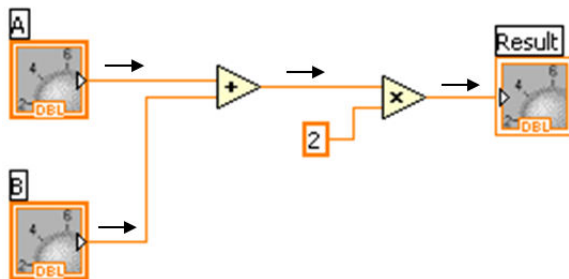
# LabVIEW™

## GRAFIČKO PROGRAMIRANJE ZA INŽENJERE I ISTRAŽIVAČE





- ❖ LabVIEW je grafički programski jezik.
- ❖ Za razliku od programskih jezika čija sintaksa je zasnovana na tekstu LabVIEW koristi blokove umesto linija teksta.
- ❖ *Kod* programa je dijagram koji se sastoji od čvorova i žica.  
Podaci “putuju” preko žica.  
**Tok izvršavanja programa zavisi od protoka podataka, a ne koda!**





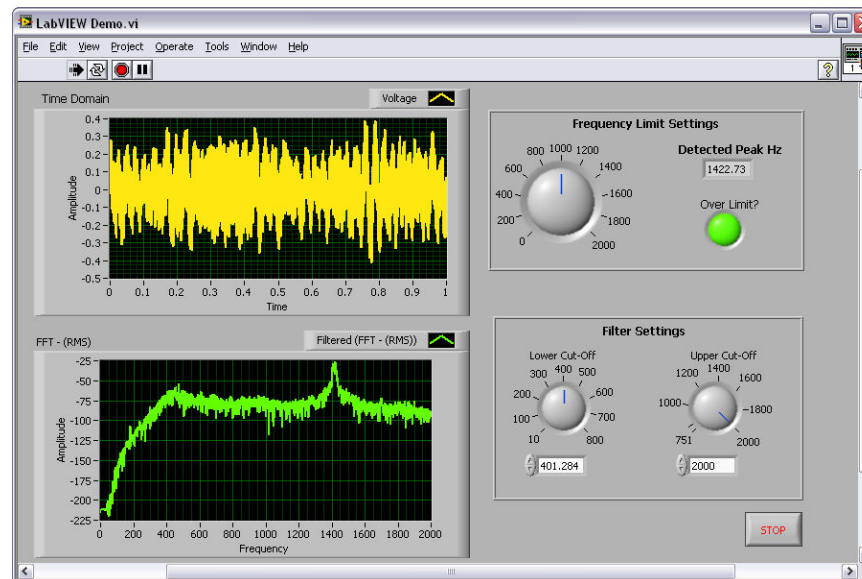
- ❖ LabVIEW je grafički programski jezik.
- ❖ Za razliku od programskih jezika čija sintaksa je zasnovana na tekstu LabVIEW koristi blokove umesto linija teksta.
- ❖ *Kod* programa je dijagram koji se sastoji od čvorova i žica.  
Podaci “putuju” preko žica.  
**Tok izvršavanja programa zavisi od protoka podataka, a ne koda!**
- ❖ Novi način razmišljanja: *data flow* umesto *code flow (chart flow)*!
- ❖ Vodeći princip: čvor ne izvršava svoju funkciju sve dok podaci, preko žica, ne “stignu” do ulaza u isti.



❖ LV programi se zovu virtualni instrumenti jer svojim izgledom podsećaju na fizičke instrumente, kao npr. osciloskope i multimetre.

❖ Ekstenzija programa pisanih u LV

\*.vi (*virtual instruments*)





- ❖ LV programi se zovu virtualni instrumenti jer svojim izgledom podsećaju na fizičke instrumente, kao npr. osciloskope i multimetre.
- ❖ Ekstenzija programa pisanih u LV
  - \*.vi (*virtual instruments*)
- ❖ Zbog prvenstvene primene kao mernih virtuelnih uređaja, LV sadrži bogate biblioteke gotovih funkcija za **kontrolu** mernih instrumenata, kao i za **prikupljanje**, **analizu**, **prezentaciju** i **skladištenje** podataka.



## Zašto koristiti LabVIEW?

- ❖ Jednostavan za učenje
- ❖ Jednostavan za primenu
- ❖ Programski kod je u vidu šeme
- ❖ Nema sintakse!!!
- ❖ Jednostavno i brzo ispravljanje grešaka
- ❖ Prilagodljiv

# Pokretanje LabVIEW programa



National Instruments LabVIEW

Pokretanjem NI LabVIEW programa pojaviće se *NI LabVIEW inicijalni prozor*

## Inicijalni prozor

### Početak:

#### 1. Iz novog dokumenta:

New»Blank VI

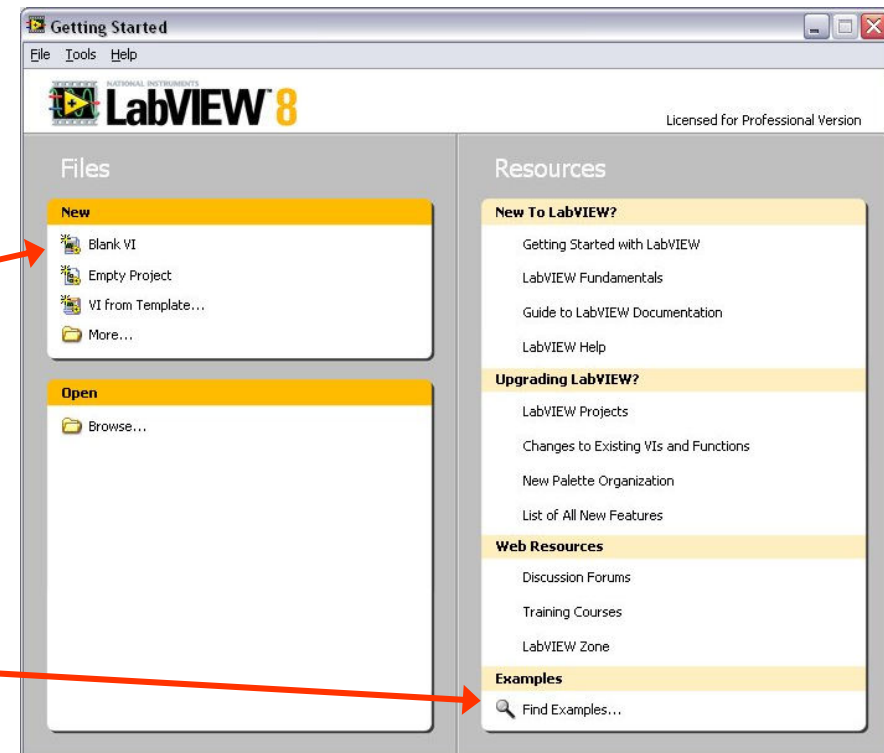
VI from Template

ili

#### 2. Iz postojećeg primera:

Examples»Find

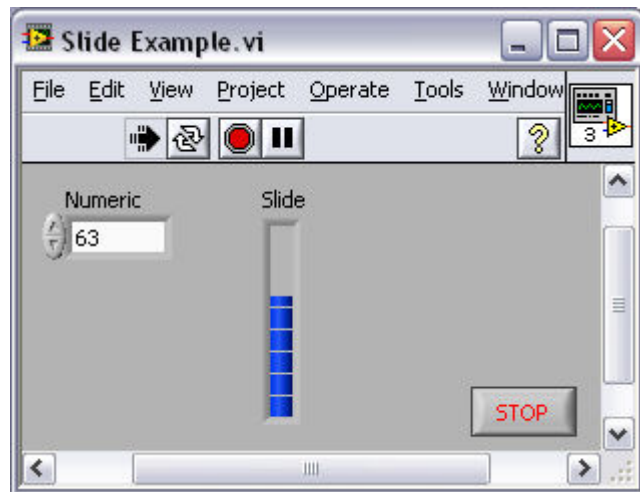
Examples...



# LabVIEW program

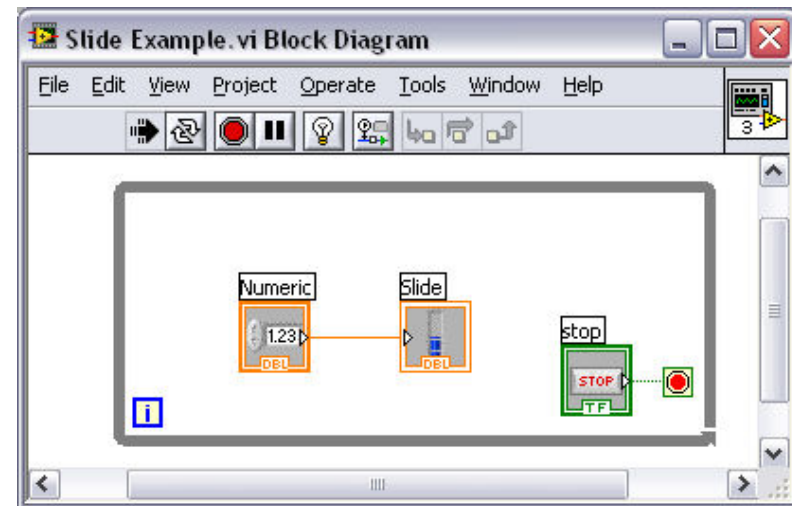
- ❖ Svaki LabVIEW program (\*.vi) se sastoji iz dva dela:
  - *front panel*
  - *block diagram*
- (\*.vi) uvek startuje sa front panelom

## Front Panel korisnički interfejs



- Kontrole = Ulazi
- Indikatori = Izlazi

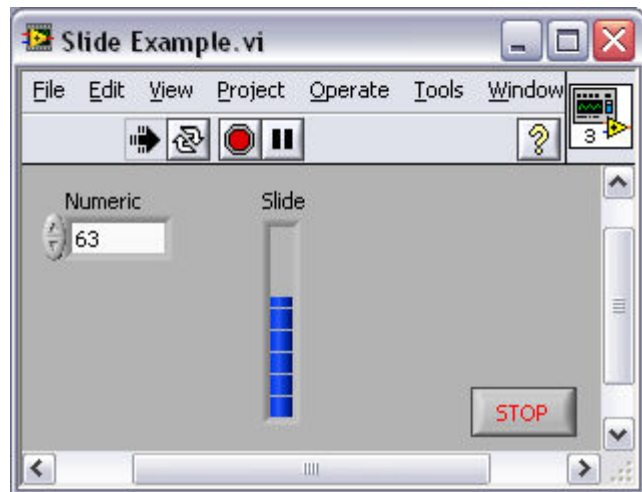
## Block Diagram Grafički kod



- Podaci “putuju” žicama od kontrola preko funkcija do indikatora.

# Front Panel korisnički interfejs

- Kontrole = Ulazi
- Indikatori = Izlazi



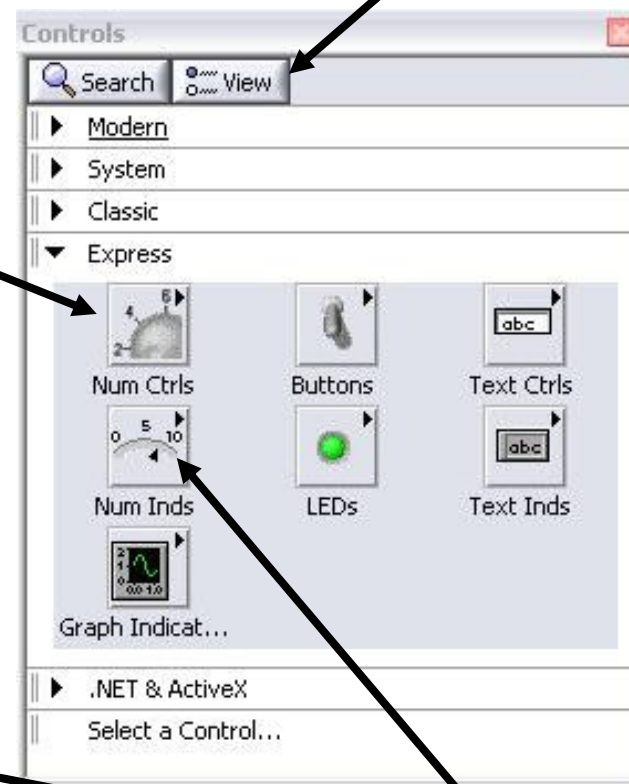
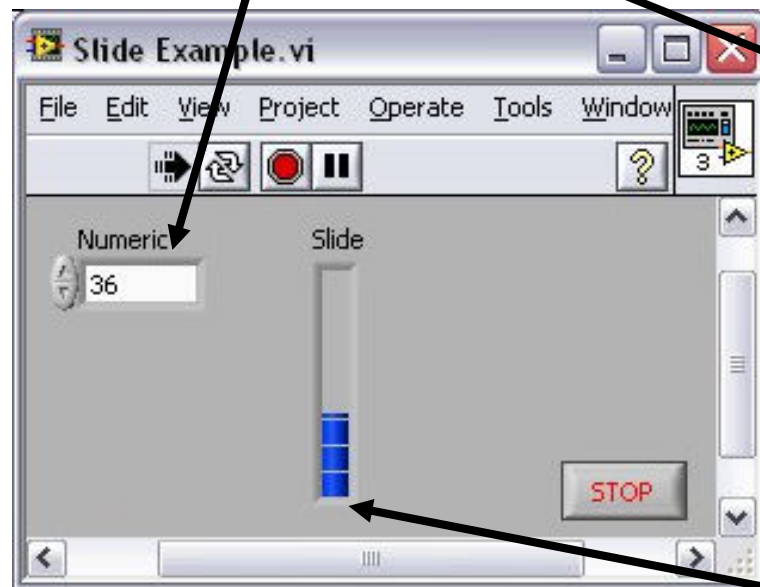
- ❖ Front panel predstavlja korisnički interfejs programa.
  - Simulira panel fizičkog instrumenta.
  - Front panel sadrži:
    - kontrole (okretna dugmad, klizači, tasteri,...)
    - indikatore (grafikoni, displeji, skale,...)
- ❖ Kontrole simuliraju ulazne delove uređaja i obezbeđuju podatke za *block diagram* vi programa.
- ❖ Indikatori simuliraju izlazne delove uređaja i služe za prikaz rezultata iz *block diagram-a* vi programa.
- ❖ Svaki element front panela ima odgovarajući *objekat* u *block diagramu*.
- ❖ Kontrole i indikatori dostupni su u okviru *Controls palette* front panela.

# Controls Palette

➤ Dostupna je samo tokom rada na front panelu.

prilagođavanje izgleda palete

Ulaz/kontrola  
*Numeric*

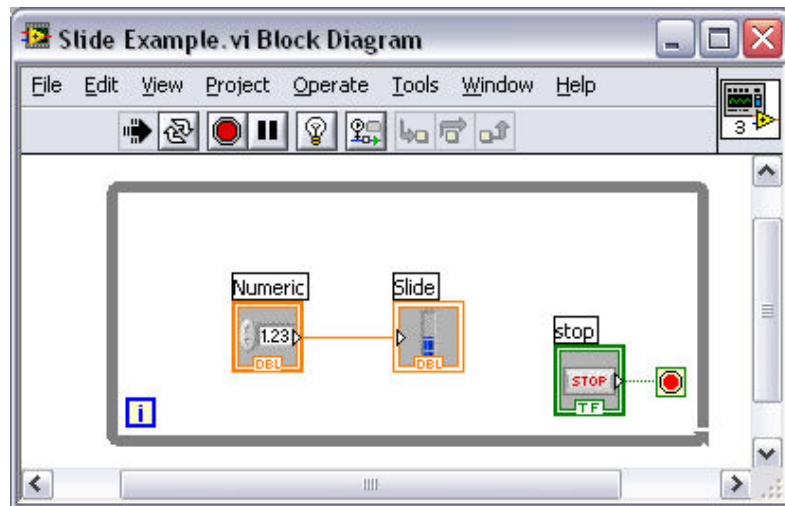


Izlaz/Indicator:  
*Numeric Slide*

# Block Diagram

## Grafički kod

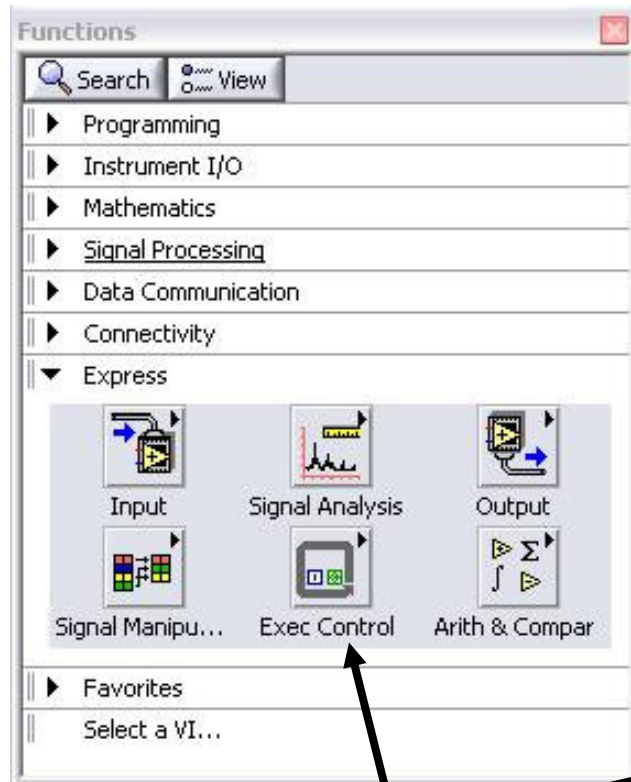
- Podaci “se prenose” žicama od kontrola preko funkcija do indikatora.



- ❖ Blok dijagram je grafički prikaz tela programa.
- ❖ Za razliku od front panela koji je namenjen korisniku, *blok dijagram je namenjen programeru.*
- ❖ Grafičko programiranje se vrši kreiranjem blok dijagrama sa već gotovim funkcijama koje se nalaze u bibliotekama LabVIEW-a, gde se funkcije prilagođavaju potrebama podešavanjem izmenljivih parametara.
- ❖ Glavni meni blok dijagrama u kome se nalaze funkcije – *Functions palette*

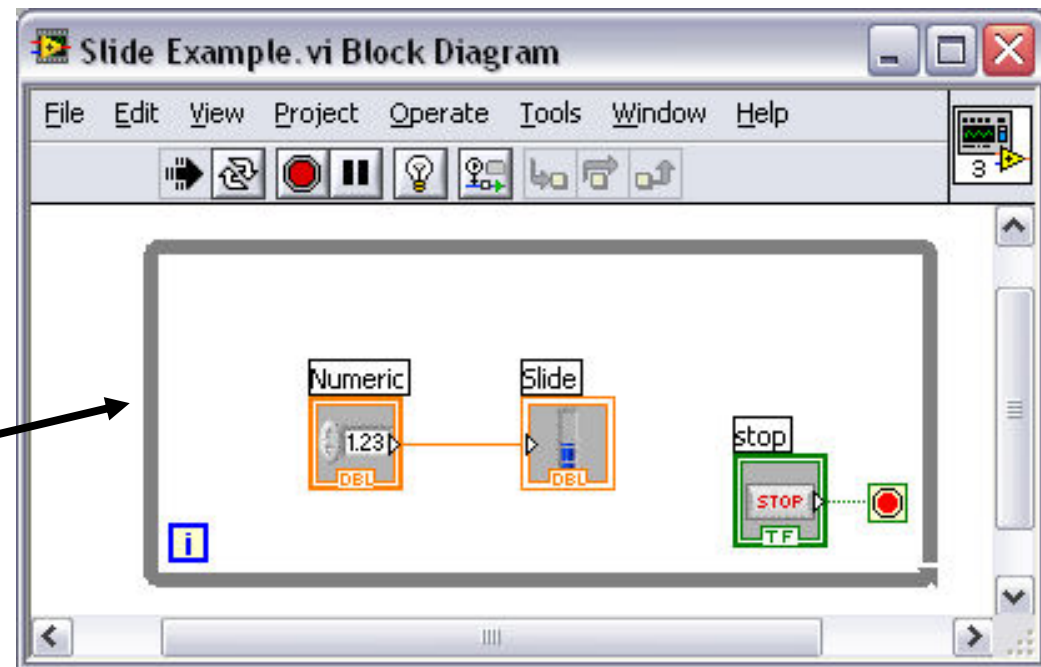
# Functions Palette

- Dostupna je samo tokom rada na Blok dijagramu.



Smeštanje elemenata  
u Block Diagram prozoru

While petlja



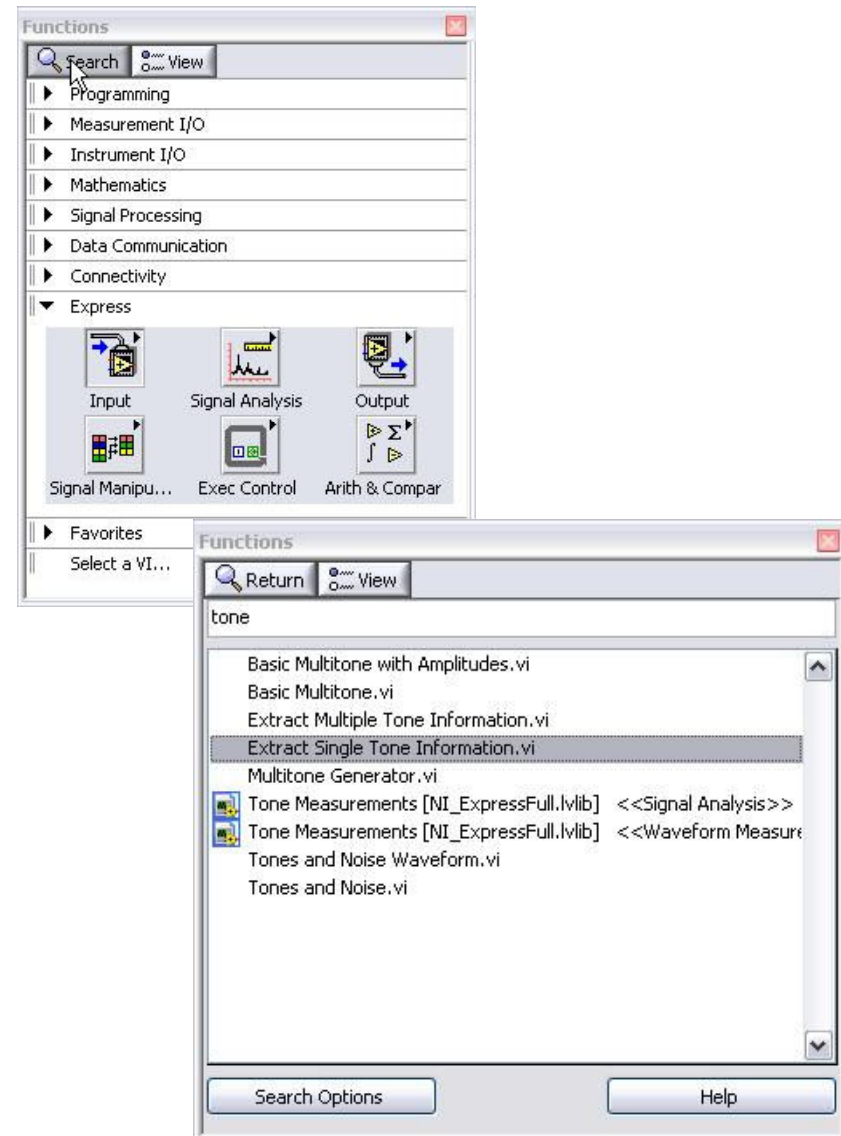
# Pretraga kontrola i funkcija

❖ Paleta su ispunjene stotinama kontrola i funkcija.

Kliknuti na taster za pretragu

❖ Pronaći željenu kontrolu ili funkciju

❖ Kliknuti i prevući objekat iz prozora za pretragu na blok dijagram/front panel.

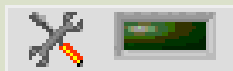


- ❖ Pri pisanju programa u LabVIEW-u se koriste sledeće tri palete:
  1. **Controls (Front Panel / View » Controls Palette)**
  2. **Functions (Block diagram / View » Functions Palette),**
  3. **Tools (View » Tools Palette).**

## Tools Palette



- PREPORUKA: Automatski izbor alatke
- Alati za rad i modifikaciju front panel i blok dijagram objekata.



Automatic Selection Tool

- LabVIEW automatski bira odgovarajući alat sa palete Tools



Operating Tool



Positioning/Resizing Tool





Labeling Tool




Wiring Tool

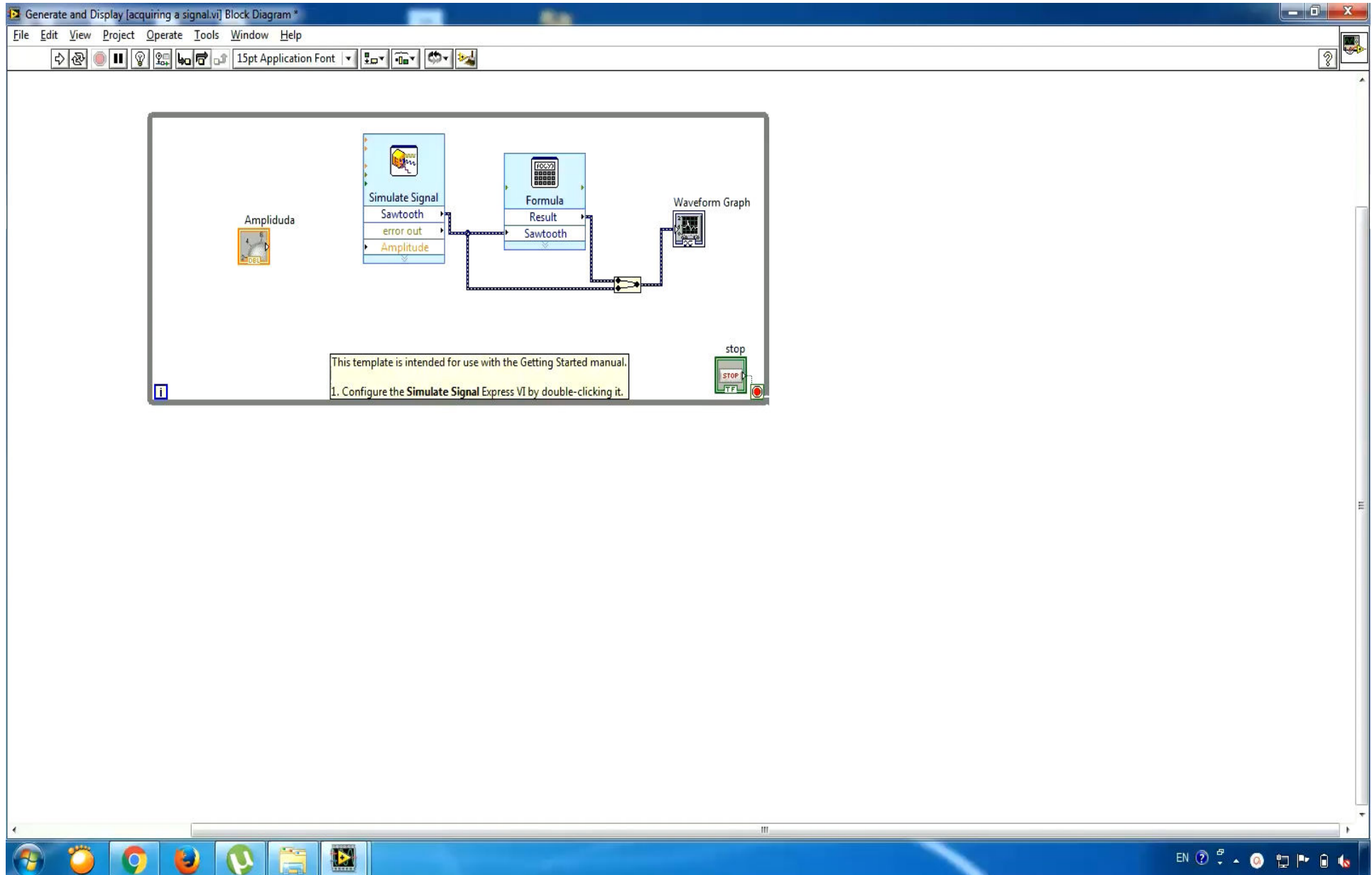
# Tools paleta

 Operating Tool

 Positioning/Resizing Tool






 Labeling Tool

 Wiring Tool



# Status Toolbar



		Run Button
		Continuous Run Button
		Abort Execution

	<b>Align Objects</b>
	<b>Distribute Objects</b>
	<b>Reorder</b>

# Tehnike debugovanja

## Prvi način



Run Button



Greška u kodu → slomljena strelica

Klikom miša na strelicu pojavljuje se spisak grešaka u kodu

## Drugi način

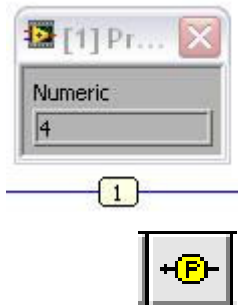


U Status Toolbar-u klikom miša na “lampicu”, a zatim Pokrenuti program i posmatrati Blok Dijagram.

Iznad “žica”, terminala, iza funkcija i potprograma se pojavljuju tekuće vrednosti.

Rad programa u ovom modu je usporen što omogućava detaljno praćenje izvršavanja koda.

## Treći način



U paleti **Tools** izborom **Probe Tool** koji se postavi ga na onu “žicu” čija vrednost se posmatra pri izvršavanju.

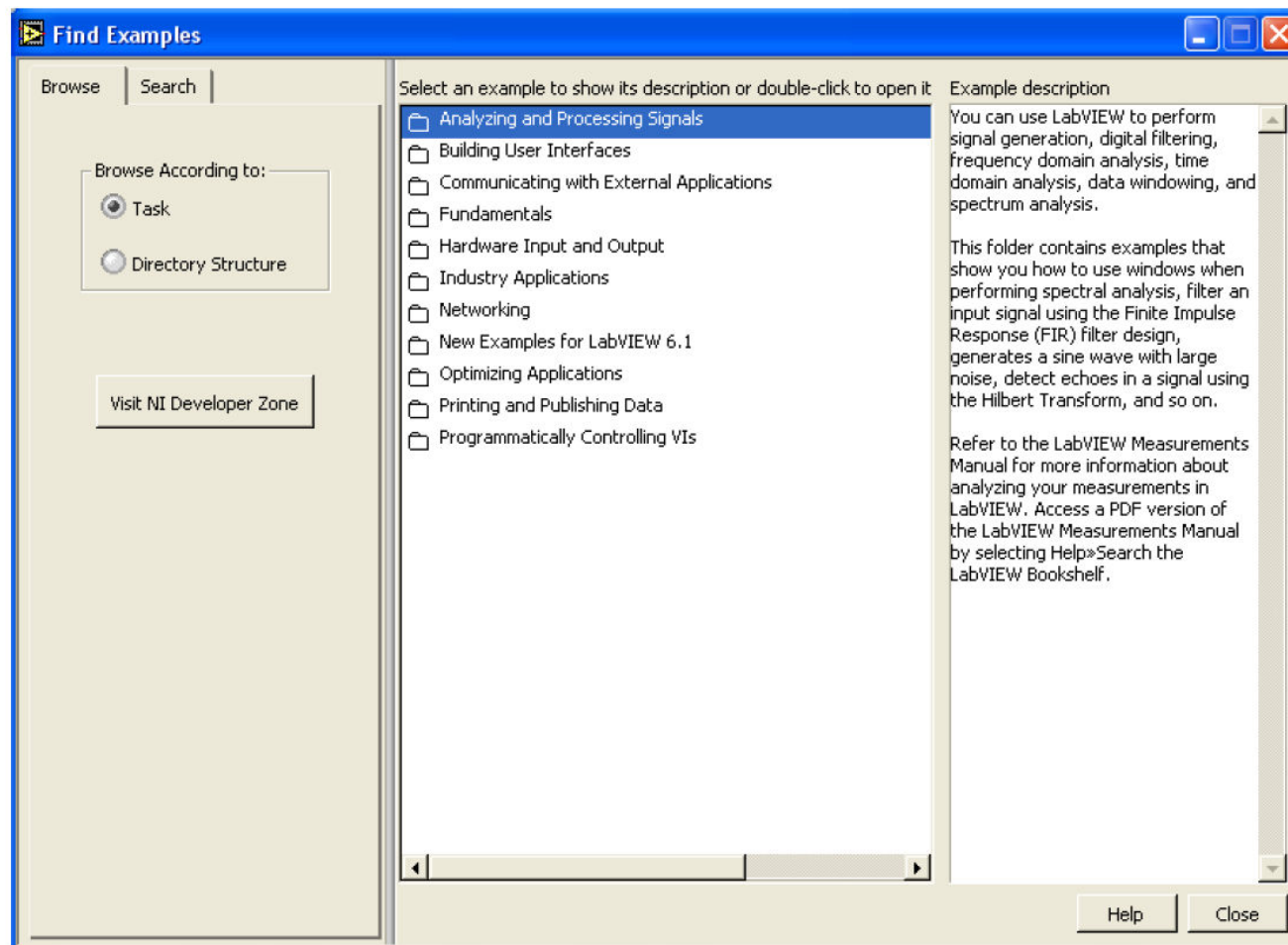
Pokrenuti program i posmatrati vrednost koja se javlja u “letećem” Probe prozoru.

# Pomoć pri radu / Help

## Prvi način

LabVIEW sadrži veliki broj gotovih primera do kojih se može doći na sledeći način: izabrati Help»Find Examples.

Pretraga je moguća po tematskim jedinicama i po ključnim rečima.



# Pomoć pri radu / Help

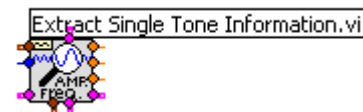
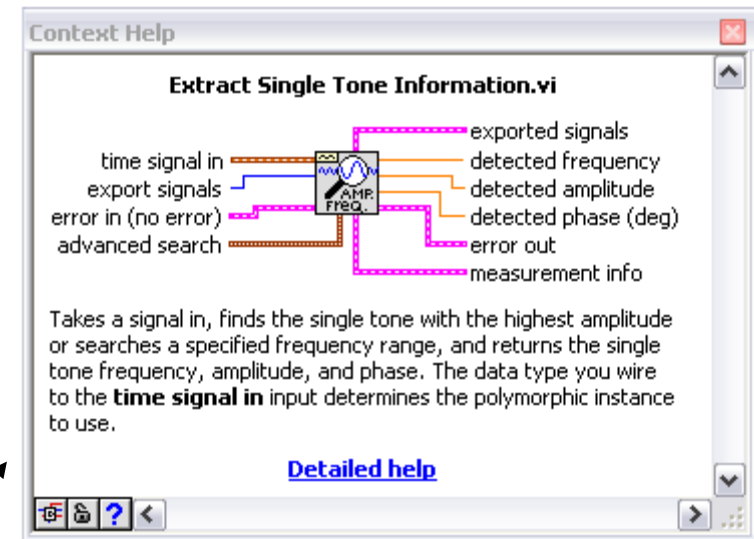
## Prvi način

LabVIEW sadrži veliki broj gotovih primera do kojih se može doći na sledeći način: izabrati Help»Find Examples.

Pretraga je moguća po tematskim jedinicama i po ključnim rečima.

## Drugi način

Stati iznad funkcije za koju želite pogledati detaljnija objašnjenja i pritisnuti CTRL+H.



## Pomoć pri radu / Help

- Prvi način** LabVIEW sadrži veliki broj gotovih primera do kojih se može doći na sledeći način: izabrati Help»Find Examples.  
Pretraga je moguća po tematskim jedinicama i po ključnim rečima.
- Drugi način** Stati iznad funkcije za koju želite pogledati detaljnija objašnjenja i pritisnuti CTRL+H.
- Treći način** Na sajtu National Instruments-a postoji link ka NI Developer zoni (zone.ni.com) gde se mogu naći gotova rešenja konkretnih problema.

# PRIMER 1

## Primer 1-a:

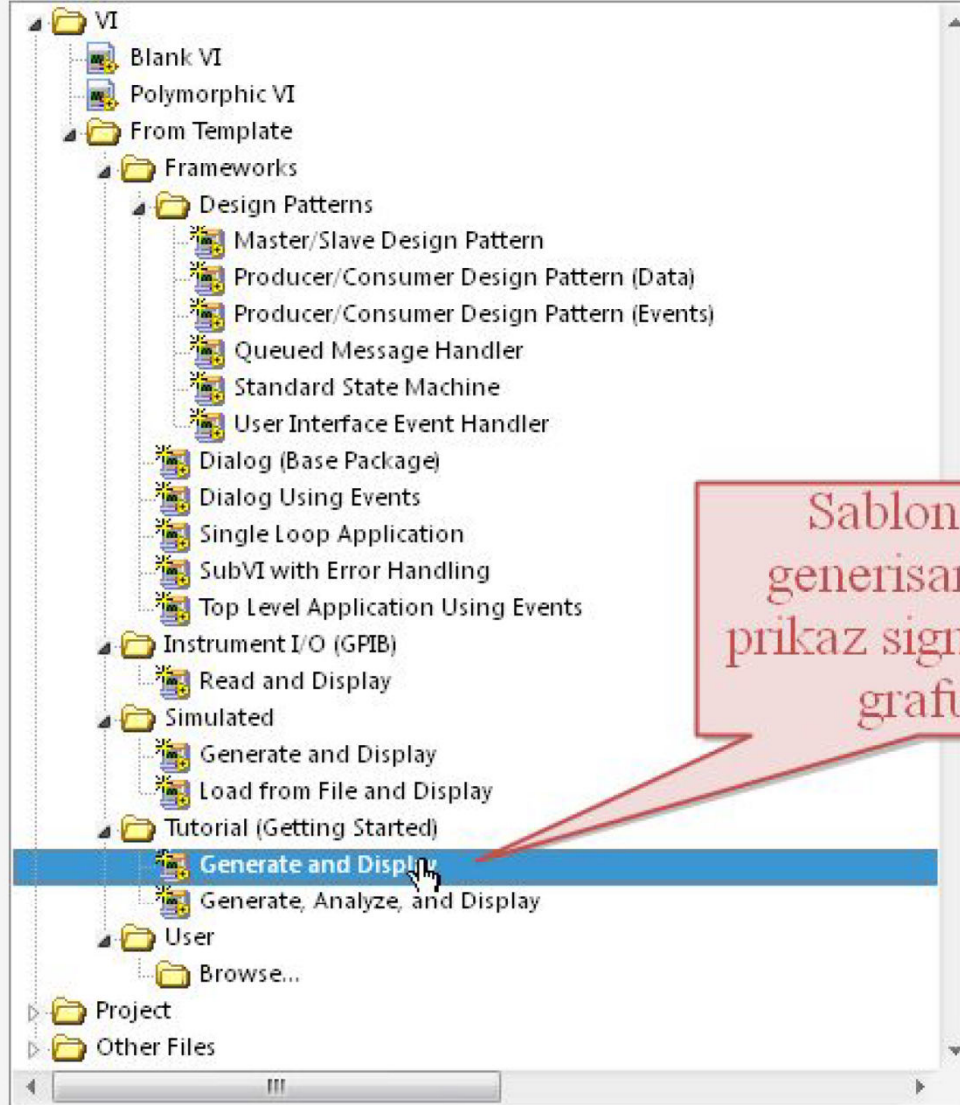
Kreirati virtuelni instrument koji:

1. Generiše testerasti signal.
2. Omogućava promenu amplitude signala iz Front panela.
3. Prikazuje signal na grafu Front panela.

➤ LabVIEW sadrži ugrađene šablone virtuelnih instrumenata (templates) koji mogu iskoristiti za većinu jednostavnih aplikacija za merenje.

➤ Za rešenje [primera 1](#) koristiće se upravo jedan ovakav šablon.

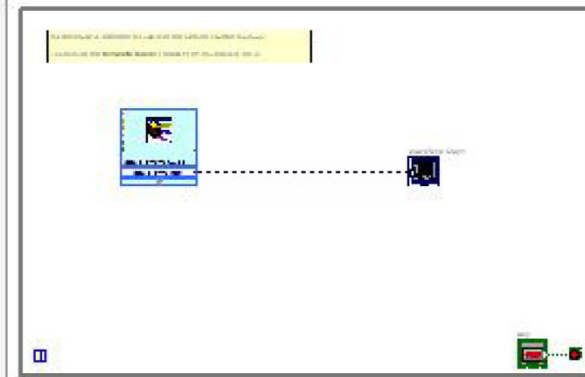
Create New



A hierarchical tree view of LabVIEW templates. The root is 'VI', which branches into 'Blank VI', 'Polymorphic VI', and 'From Template'. 'From Template' further branches into 'Frameworks', 'Instrument I/O (GPIB)', 'Simulated', 'Tutorial (Getting Started)', 'User', 'Project', and 'Other Files'. Under 'Frameworks', there are sub-folders like 'Design Patterns', 'Dialog (Base Package)', 'Dialog Using Events', 'Single Loop Application', 'SubVI with Error Handling', and 'Top Level Application Using Events'. Under 'Design Patterns', there are templates like 'Master/Slave Design Pattern', 'Producer/Consumer Design Pattern (Data)', 'Producer/Consumer Design Pattern (Events)', 'Queued Message Handler', 'Standard State Machine', and 'User Interface Event Handler'. Under 'Simulated', there are 'Generate and Display' and 'Load from File and Display'. Under 'Tutorial (Getting Started)', there are 'Generate and Display' (highlighted in blue), 'Generate, Analyze, and Display', and 'User'. Under 'User', there is a 'Browse...' folder.

Sablon za generisanje i prikaz signala na grafu

Description



Use this template to generate a signal and display the generated data in a graph. You can use this template with the exercises in the Getting Started with LabVIEW

This template enables automatic error handling by default. If you create from this template.

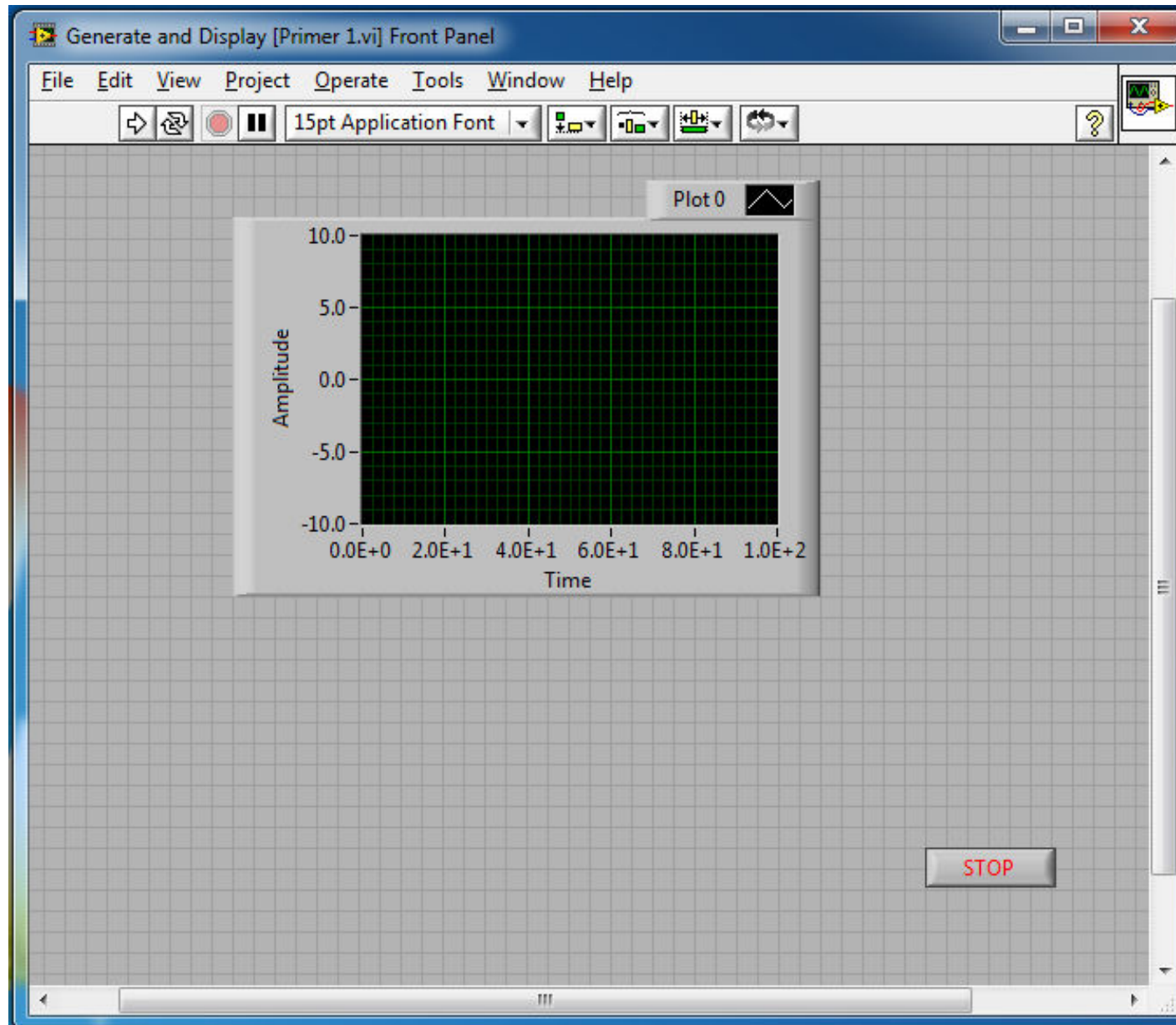
Add to project

OK

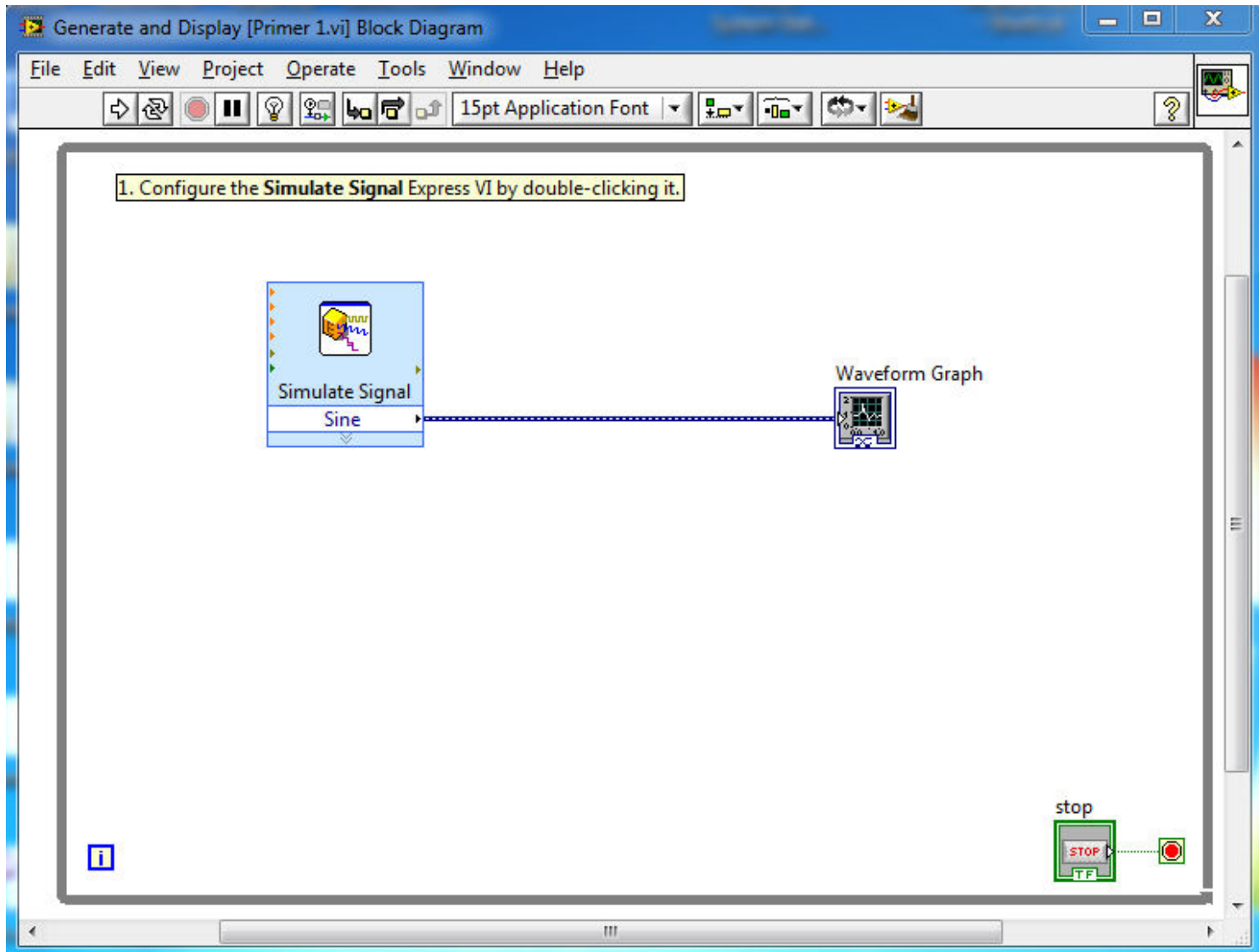
Cancel

Help

➤ Nakon odabira šablona pojavljuje se Front Panel...



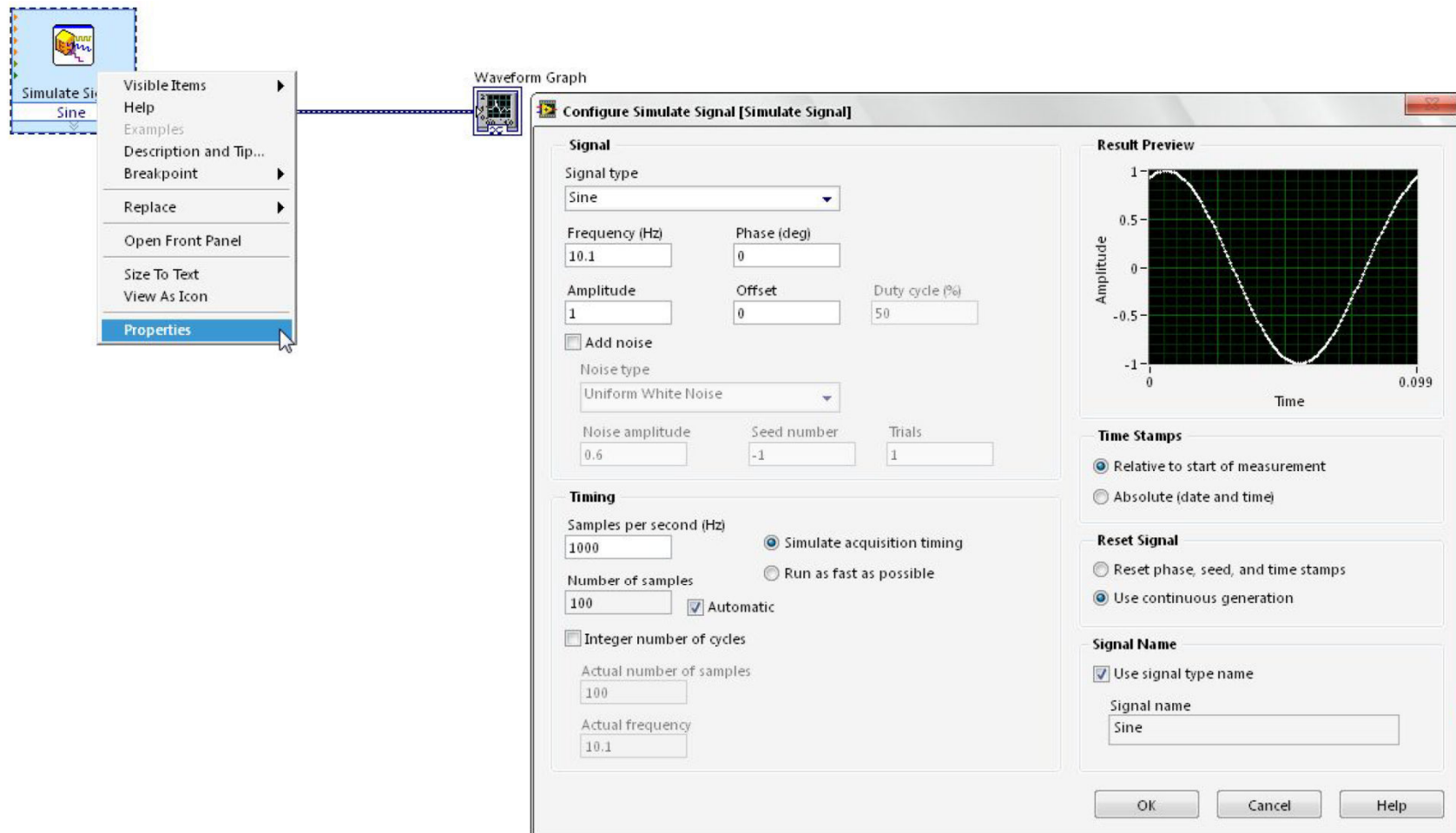
- I ekran za kreiranje blok dijagrama virtuelnog instrumenta sa blokovima za generisanje i prikaz signala.
- Ovi blokovi su u odgovarajućoj sprezi sa kontrolama/indikatorima na Front Panelu.



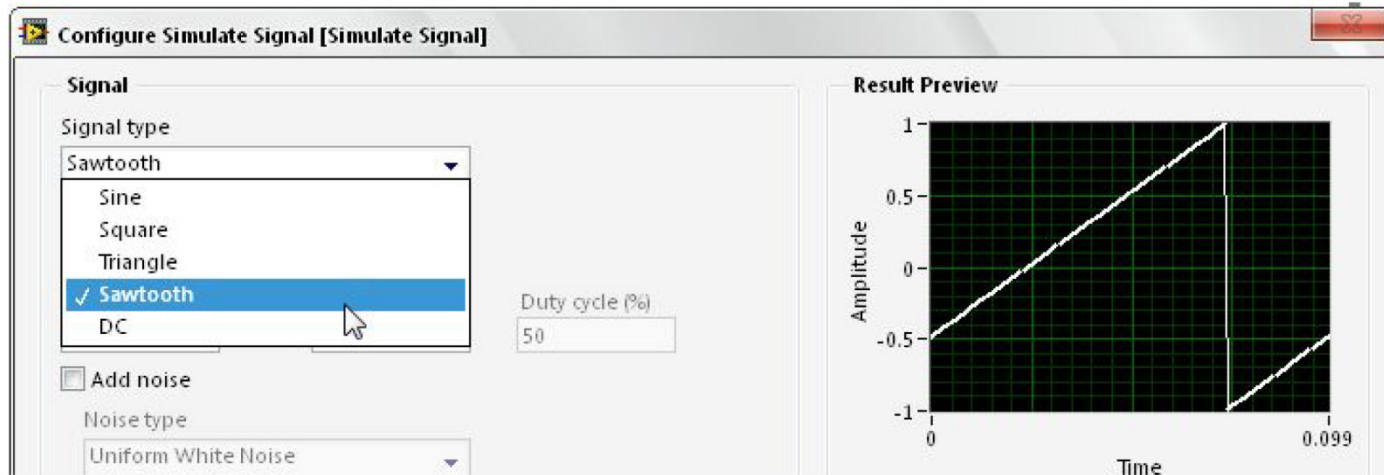
# 1. Generisati testerasti signal

➤ Blok *Simulate Signal*.

➤ Desnim klikom na simbol ovog bloka se selektuje *Properties*, gde se nakon toga pojavljuje ekran za izmenu atributa bloka.

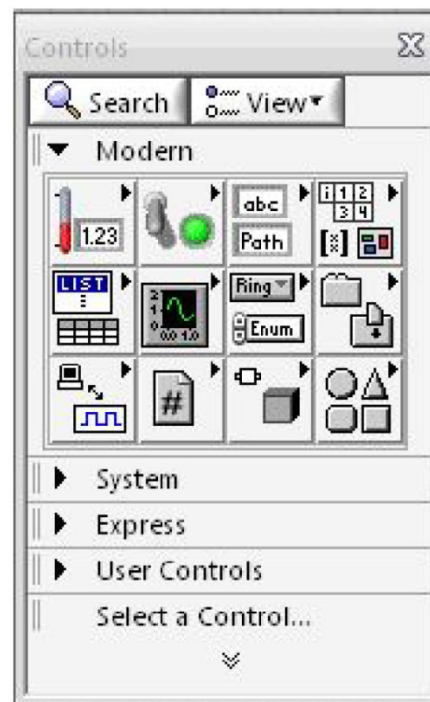
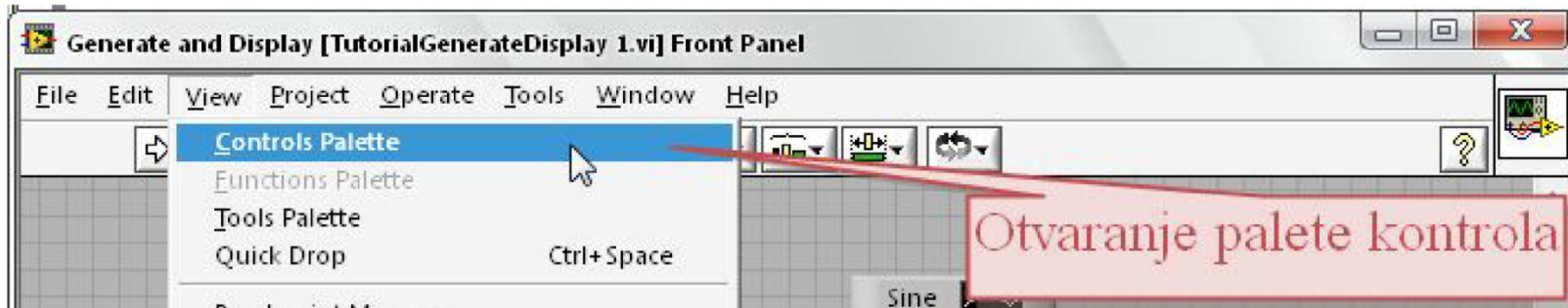


➤ U okviru liste **Signal Type** izabrati tip generisanog signala **Sawtooth** što će rezultirati generisanjem testerastog signala.

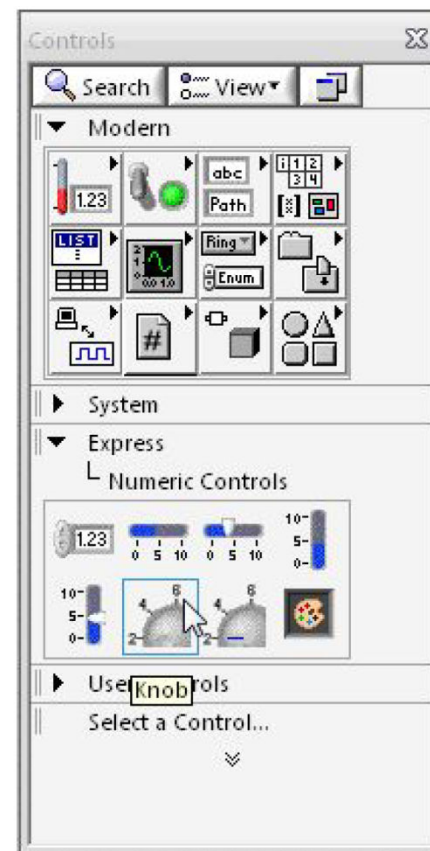
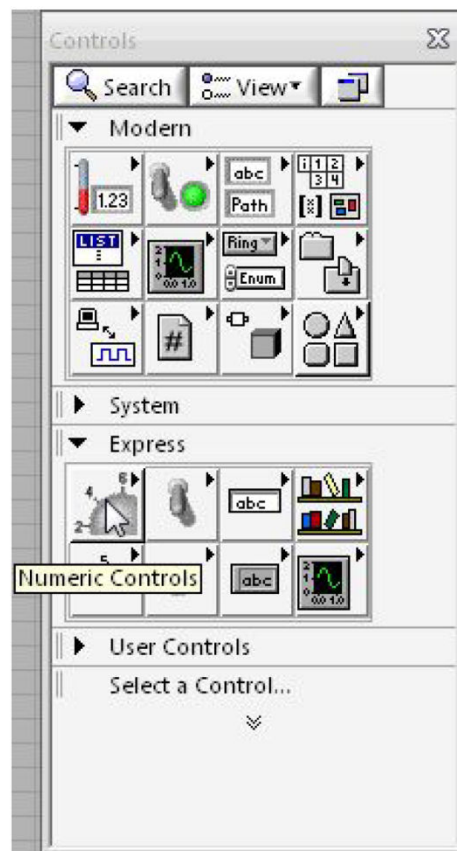


## 2. Omogućiti promenu amplitude signala iz Front panela.

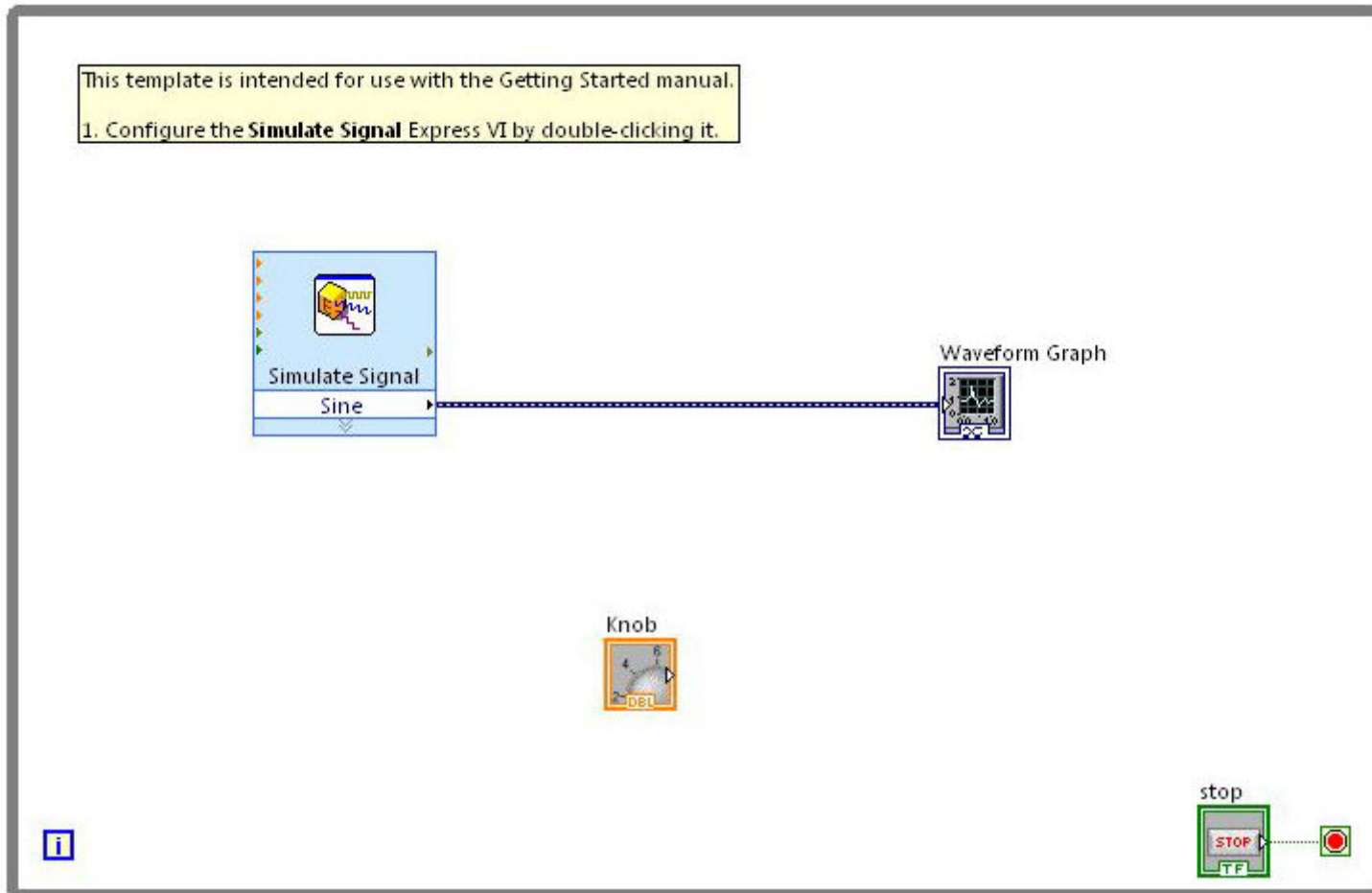
- Dodavanje novih kontrola u okviru **Front Panel**-a se postiže selektovanjem odgovarajuće iz palete kontrola.



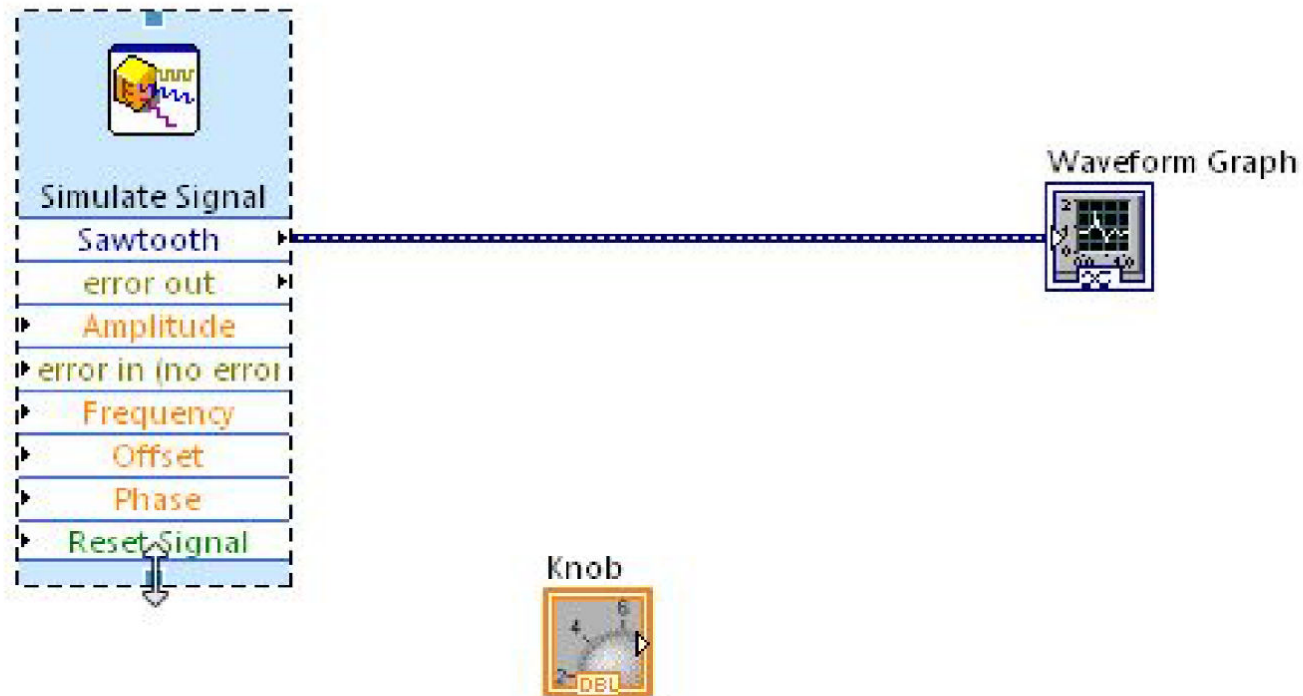
- Kontrole su grupisane po kategorijama.
- U okviru **Express** kategorije selektovati **Numeric Controls**, a zatim odabrati dugme (**Knob**) kontrolu i postaviti je na površinu panela (npr . pored grafa).



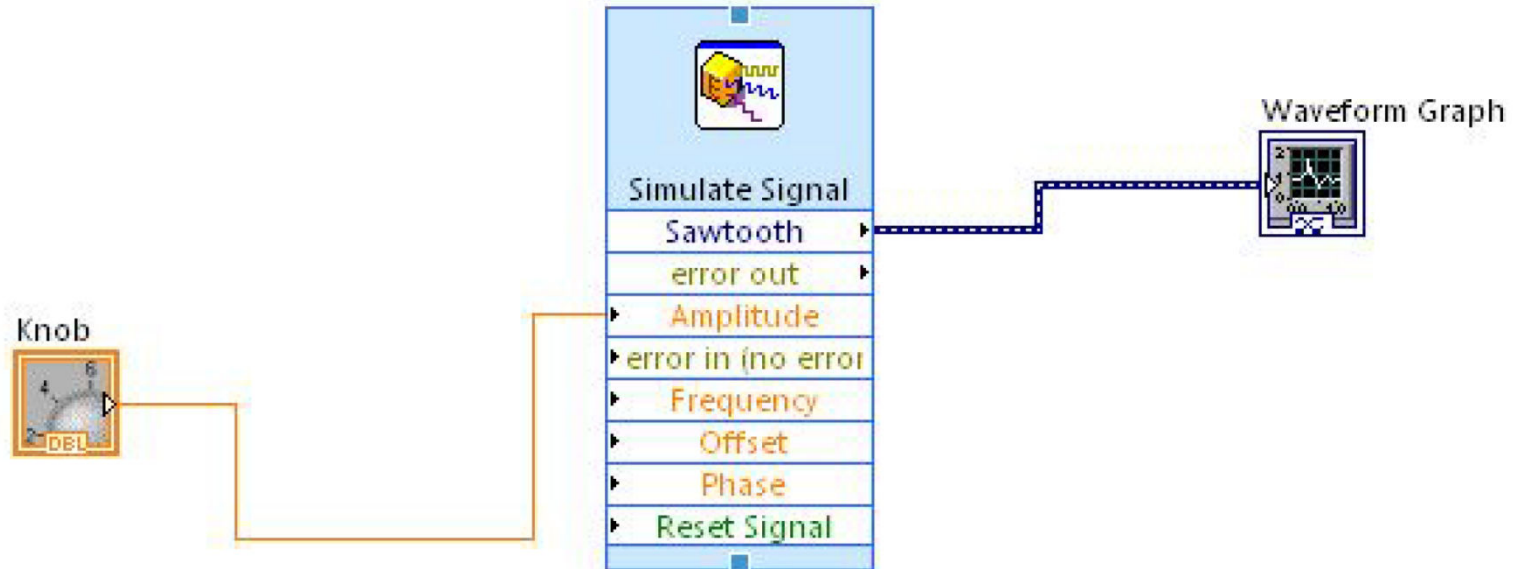
- Dugme (**Knob**) će biti iskorišćen za promenu amplitude generisanog signala.
- Na radnoj površini **Block Diagram** editora nakon svih akcija nalaze se blokovi kao na slici.



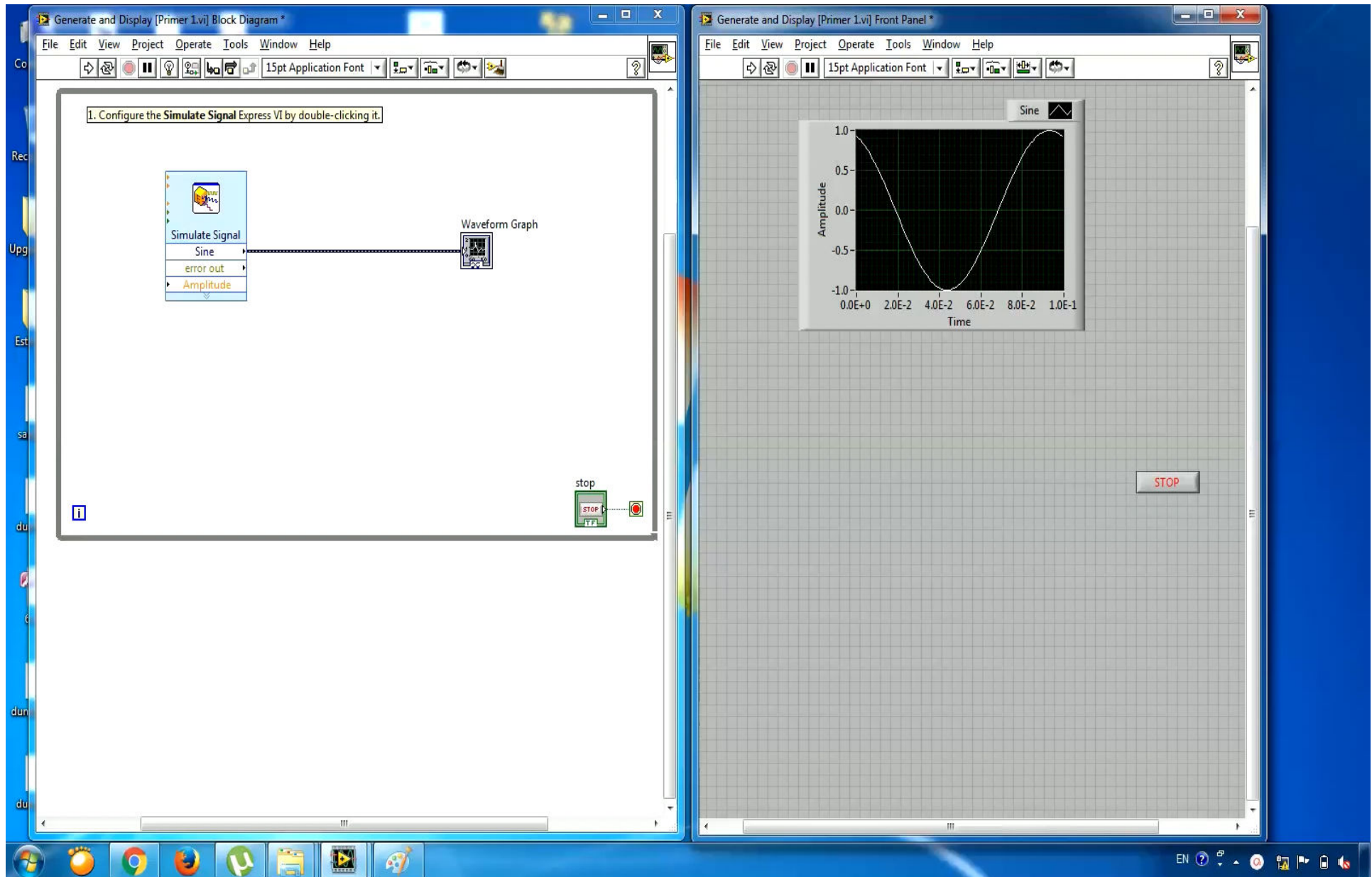
- Blok za simuliranje signala sadrži attribute koji se prikazuju nakon proširenja blok **Simulate Signal**.



➤ Povezivanje **Knob** kontrole sa **Simulate Signal Amplitude** atributom se vrši njihovim “ožičavanjem” (ustopavljanjem veze).



➤ Startovanjem programa sa **Run** se može videti uticaj **Knob** kontrole na amplitudu generisanog signala.

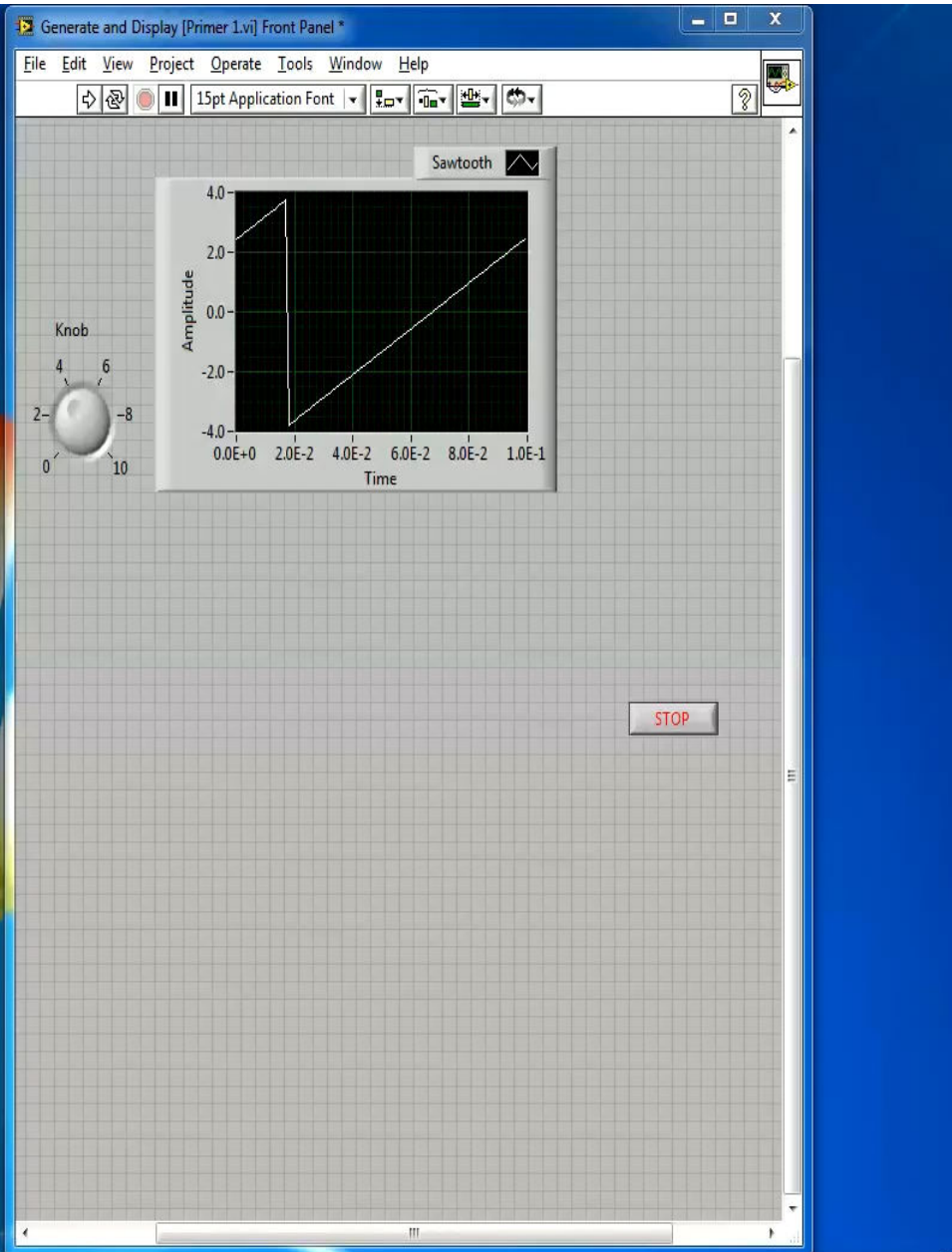
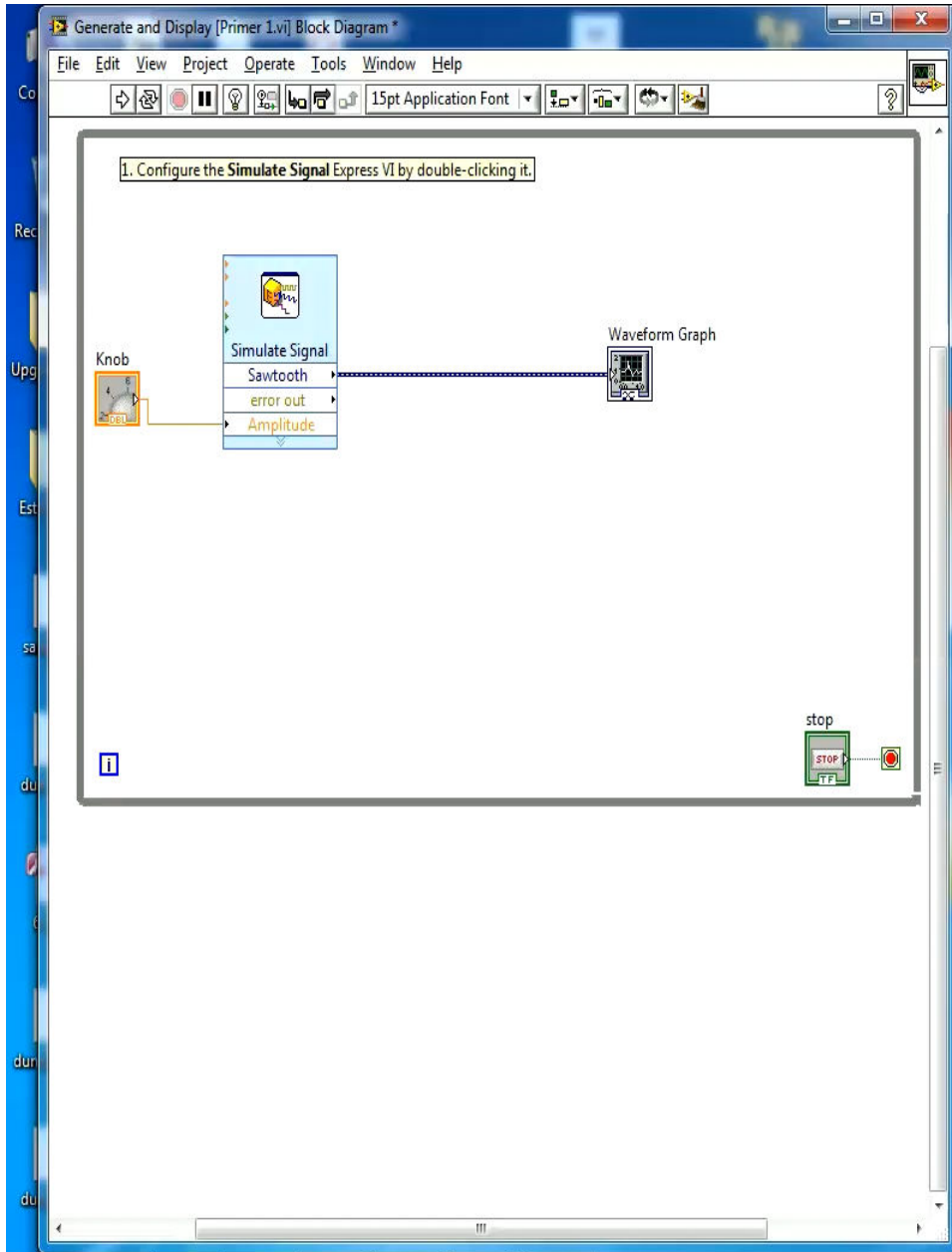


## Primer 1-b:

Na grafu Front panela prikazati:

(1) generisan trouglasti signal,

(2) kao i signal čija je amplituda 10 puta veća od signala (1).



## ► Uređivanje Block diagrama i prilagođavanje kontrola Front panela

The image displays two side-by-side windows from the LabVIEW software interface, illustrating the configuration of a signal simulation and its visualization.

**Left Window: Generate and Display [Primer 1.vi] Block Diagram**

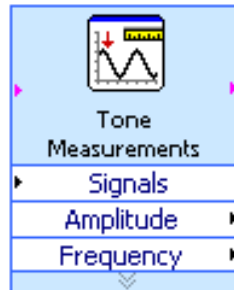
This window shows the internal logic of the program. A **Knob** control is connected to the **Amplitude** input of a **Simulate Signal** block. The **Simulate Signal** block is configured with **Sawtooth** as the signal type. Its **error out** terminal is connected to the **Formula** block. The **Formula** block is set to **Sawtooth** and outputs the **Result**. The output of the **Formula** block is connected to the **Waveform Graph** block. A **stop** button is also present in the diagram.

**Right Window: Generate and Display [Primer 1.vi] Front Panel**

This window shows the user interface. It features a **Knob** control on the left, which is currently set to a value of approximately 4. To the right of the knob is a **Waveform Graph**. The graph has **Amplitude** on the vertical axis (ranging from -30.0 to 30.0) and **Time** on the horizontal axis (ranging from 0.0E+0 to 1.0E-1). Two waveforms are displayed: a black **Sawtooth** wave and a red **Sawtooth (Formula Result)** wave. The red wave is a scaled and shifted version of the black wave, reflecting the knob's position. A **STOP** button is located at the bottom right of the front panel.

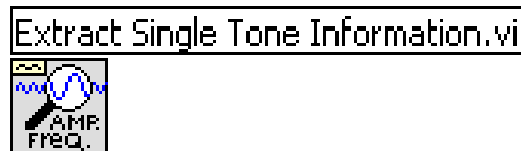
## Tri vrste funkcija (Functions Palette)

Express VI: (plavi okvir)



To su interaktivni VI koji imaju svoj dijalog prozor iz koga je korisniku omogućeno da konfiguriše Express VI .

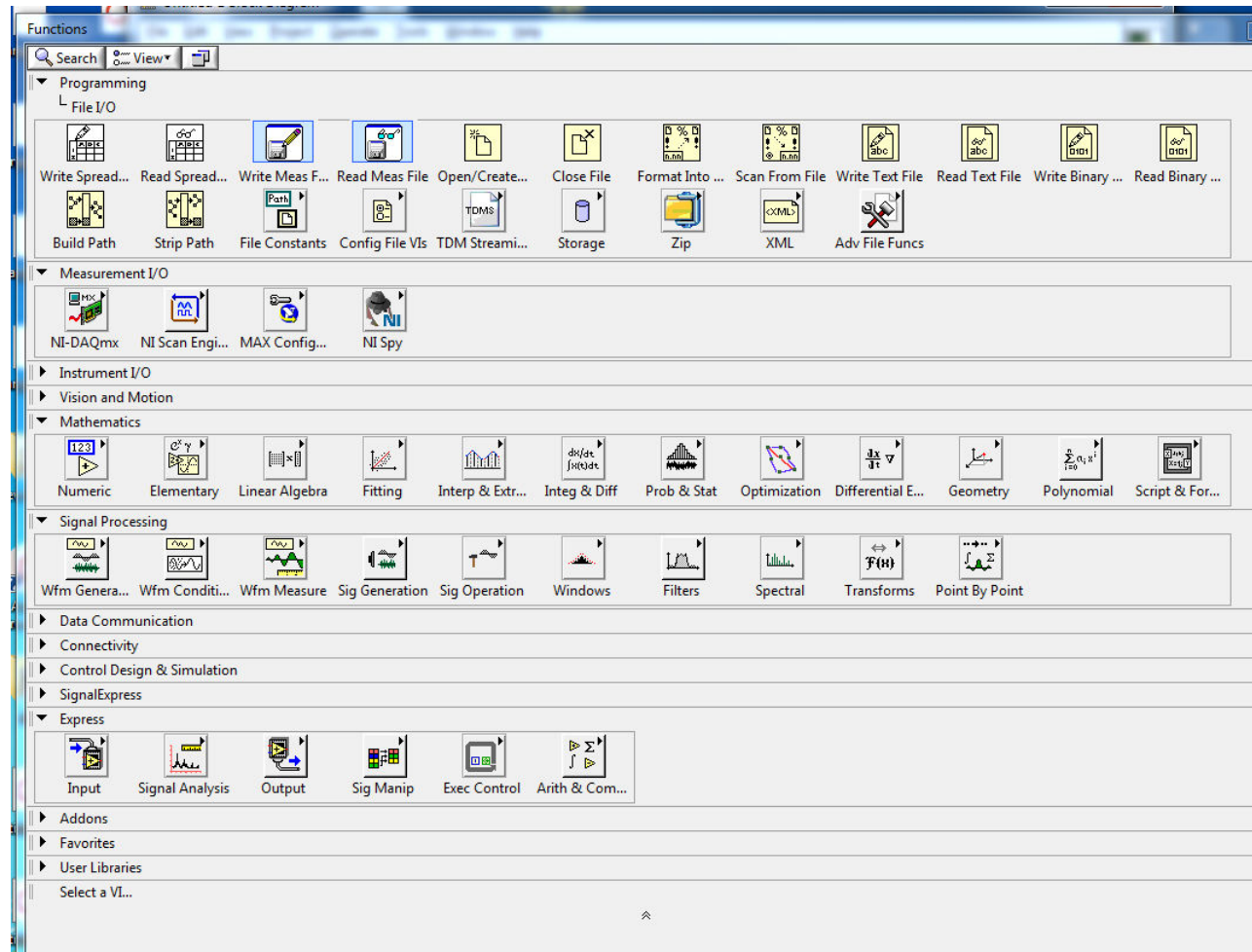
Standardni VI:



**Funkcije:** (žute boje) Osnovni elementi pomoću kojih su izgrađeni svi blokovi VI. Funkcije nemaju front panel ili blok dijagram.



# Koje funkcije su dostupne?



# Koje funkcije su dostupne?

## ❖ Ulaz/Izlaz

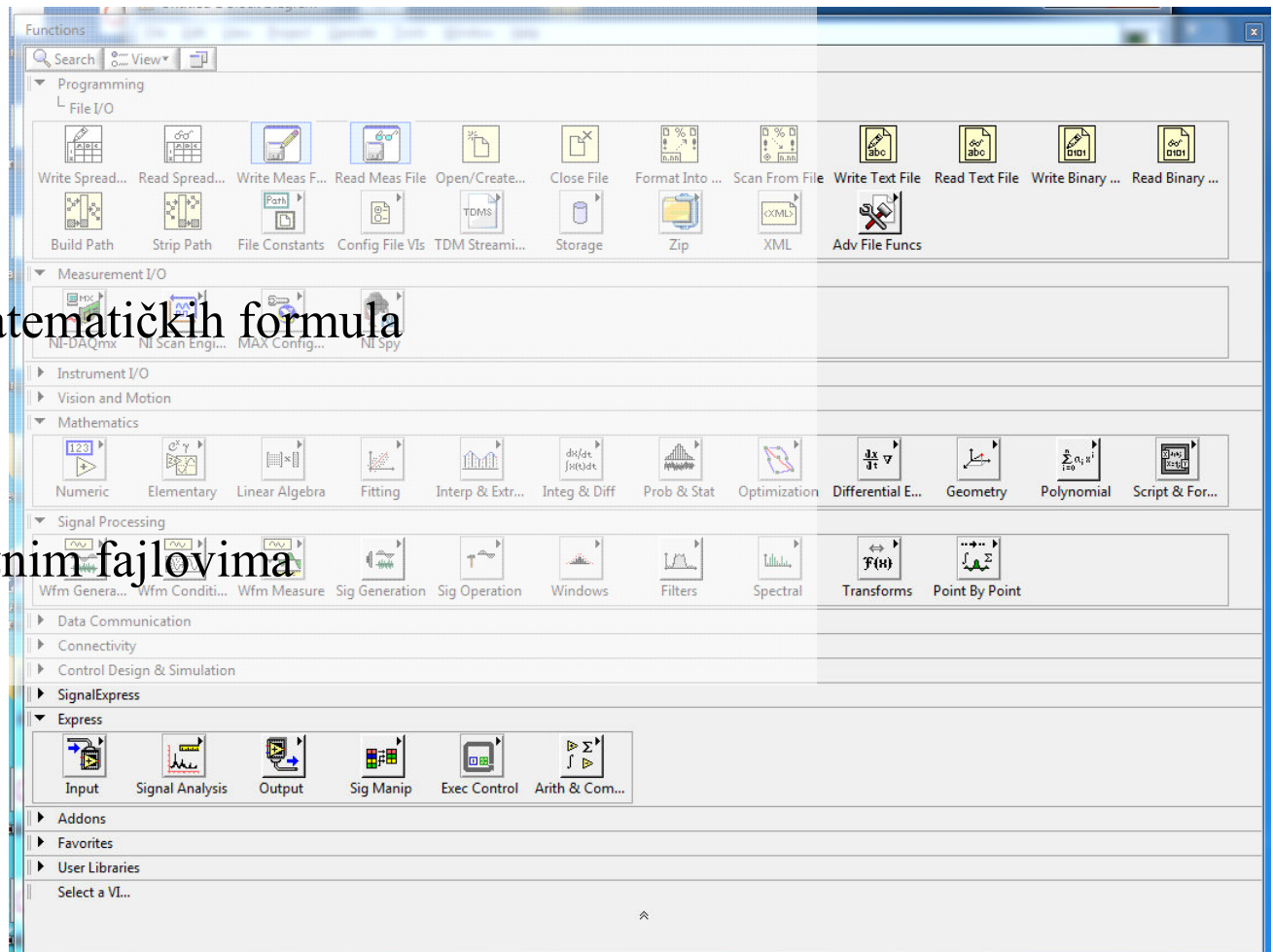
- Simulacija signala i podataka
- Akvizicija i generisanje signala korišćenjem uređaja za akviziciju podataka (DAQ)

## ❖ Analiza

- Obrada signala
- Statistika
- Primena složenih matematičkih formula

## ❖ Skladištenje

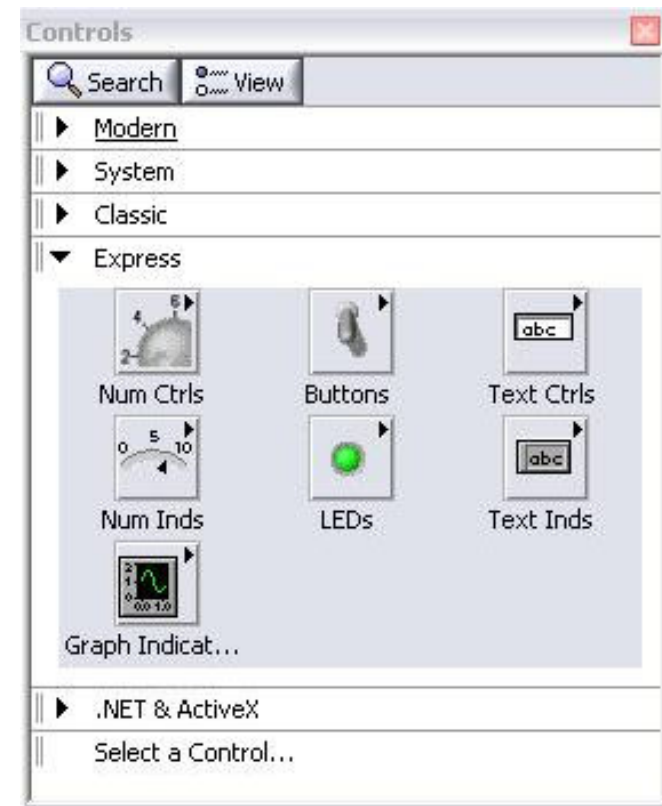
- Rad sa izlazno / ulaznim fajlovima



## Koje kontrole i indikatori su dostupni?

- **Numerički podaci**
  - Numerički ulaz i izlaz
- **Podaci Bulove logike**
  - Dugme i lampice
- **Predstavljanje podataka**
  - numerički displej
  - 2D grafikon
  - 3D grafikon
- **Ukrašavanje**
  - strelice
  - tabulatori
- **Ostalo**
  - String and tekst boksovi

*Express Controls Palette*



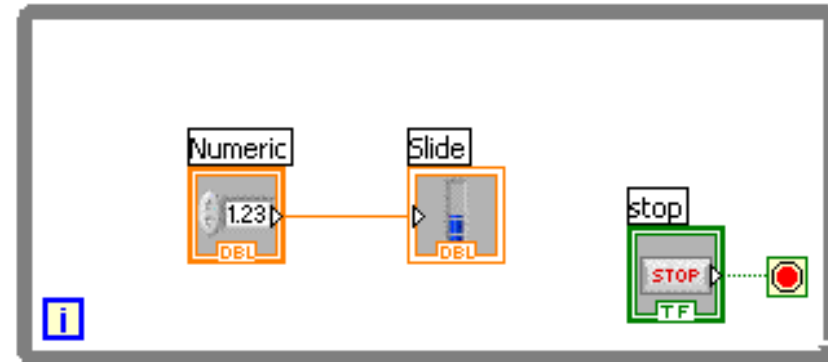
# Petlje

- **While petlje**

- **i** terminal broji iteracije
- Uvek se izvršava najmanje jednom
- Izvršava se dok uslov za zaustavljanje nije ispunjen



While petlja

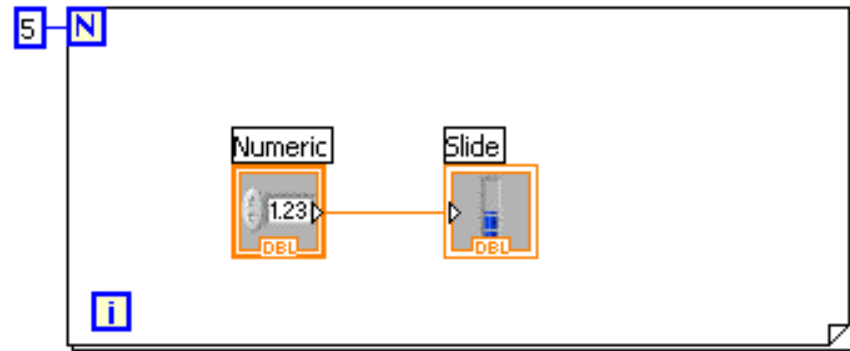


- **For petlje**

- **i** terminal broji iteracije
- Izvršava se N - puta u skladu sa ulaznim terminalom

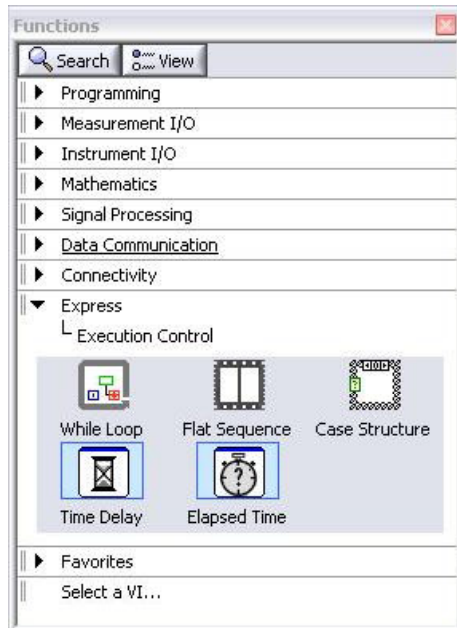


For petlja

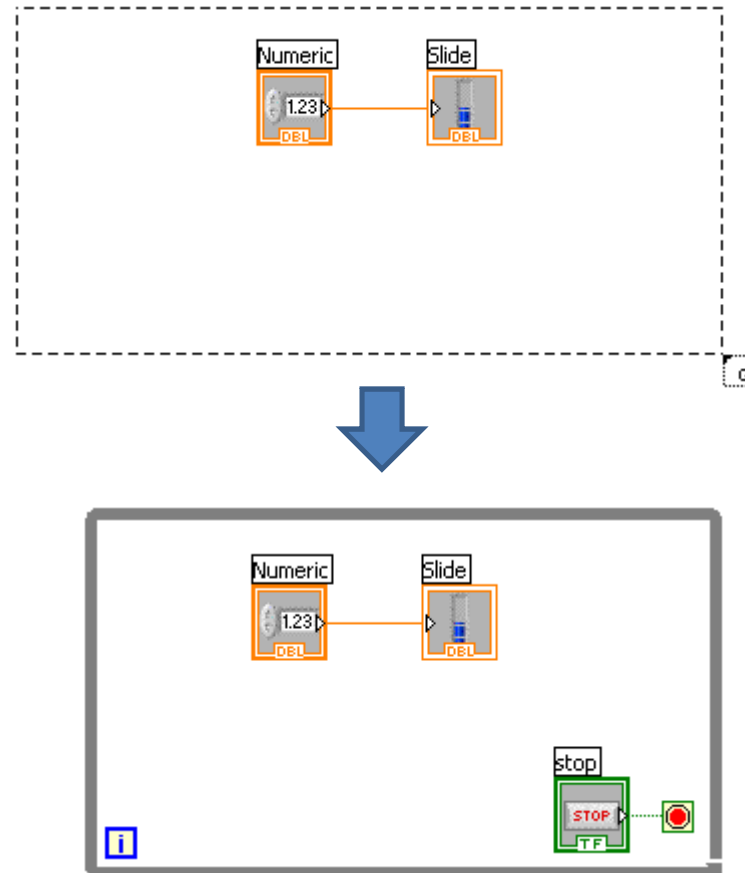


# Unošenje/crtanje petlje u blok dijagram

## 1. Izbor iz Functions palette



## 2. Ograditi kod koji se ponavlja



# Primer upotrebe While petlje

The image displays two side-by-side screenshots of the LabVIEW software interface, illustrating the development of a simple signal processing application.

**Left Screenshot: Block Diagram**  
The window title is "Generate and Display [Primer 1.vi] Block Diagram". The diagram includes:

- A **Knob** control connected to the **Amplitude** input of a **Simulate Signal** Express VI.
- The **Simulate Signal** VI has its **Sawtooth** and **error out** outputs connected to the **Formula** block.
- The **Formula** block contains a **Sawtooth** function and a **Result** output.
- The **Result** output is connected to a **Waveform Graph**.

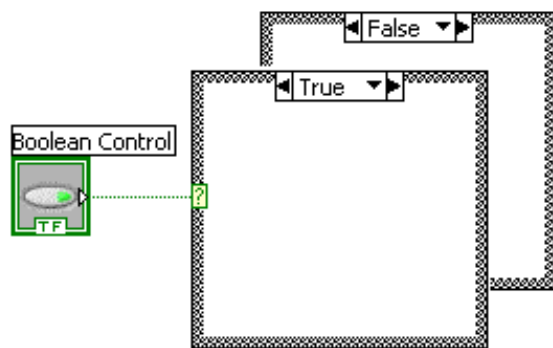
A text box at the bottom of the diagram area contains the instruction: "1. Configure the **Simulate Signal** Express VI by double-clicking it."

**Right Screenshot: Front Panel**  
The window title is "Generate and Display [Primer 1.vi] Front Panel". The front panel features:

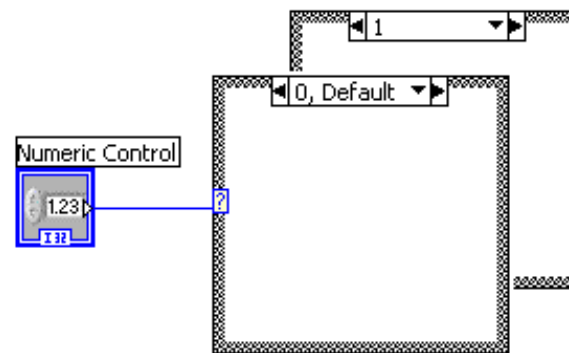
- A **Knob** control with a scale from 0 to 10.
- A **Waveform Graph** with the following axes:
  - Y-axis:** Amplitude, ranging from -1.0 to 1.0.
  - X-axis:** Time, ranging from 0.0E+0 to 1.0E-1.
- A legend at the top right of the graph area showing two data series:
  - Sawtooth** (represented by a downward-pointing triangle icon)
  - Sawtooth (Formula Result)** (represented by an upward-pointing triangle icon)

# Donošenje odluka u LabVIEW-u

## 1. Case Structura

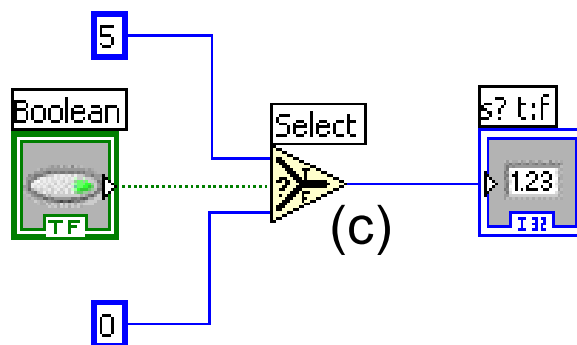


(a)



(b)

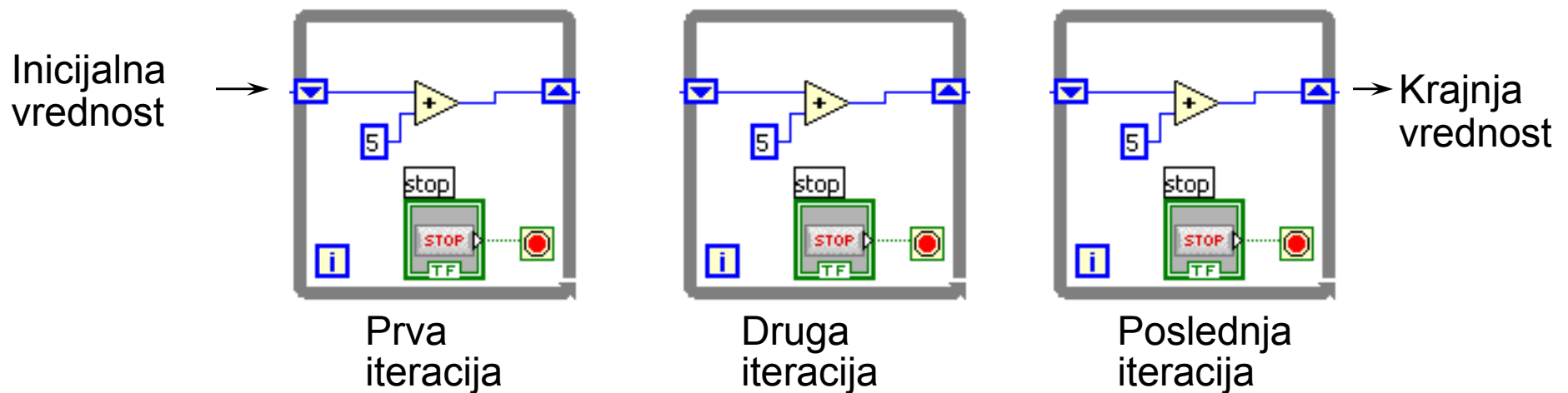
## 2. Select



(c)

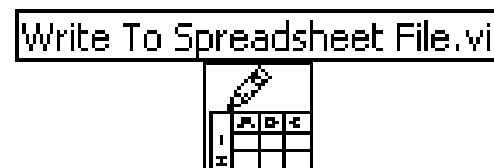
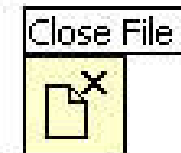
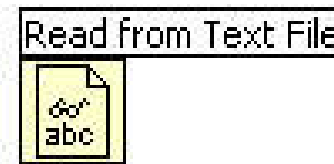
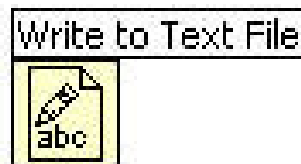
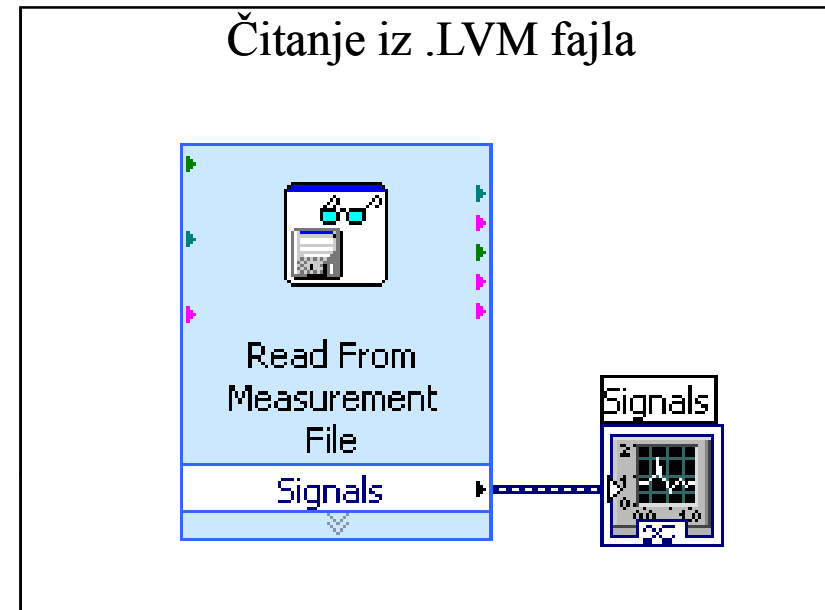
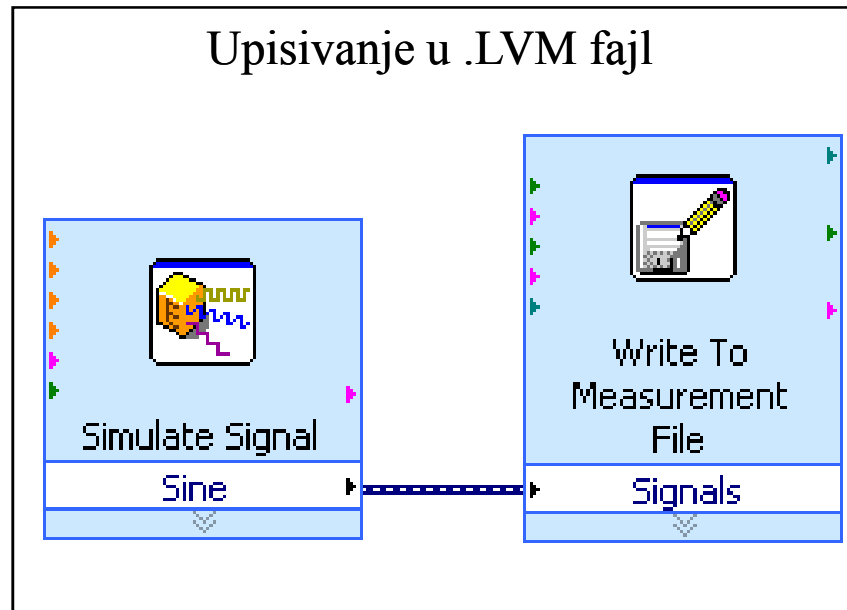
# Pristup podacima iz prethodne petlje Shift Registrar

- ❖ Unosi se desnim klikom miša na levi ili desni okvir petlje i odabirom opcije *Add Shift Register*.
- ❖ Desni terminal čuva podatke nakon završetka iteracije.
- ❖ Levi terminal obezbeđuje sačuvan podatak na početku naredne iteracije.



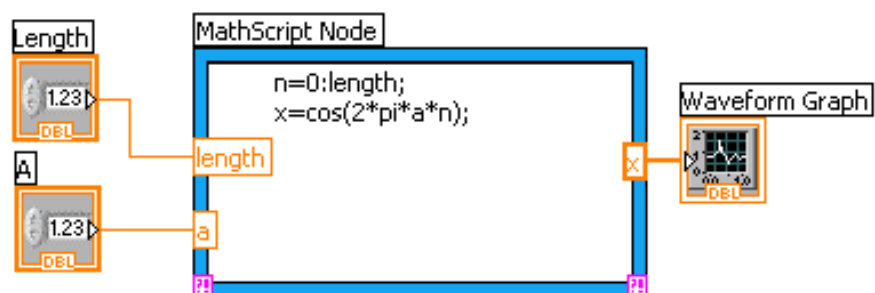
# Ulazni i izlazni fajlovi

- ❖ Omogućeno je snimanje podataka u fajl ili čitanje podataka iz fajla.

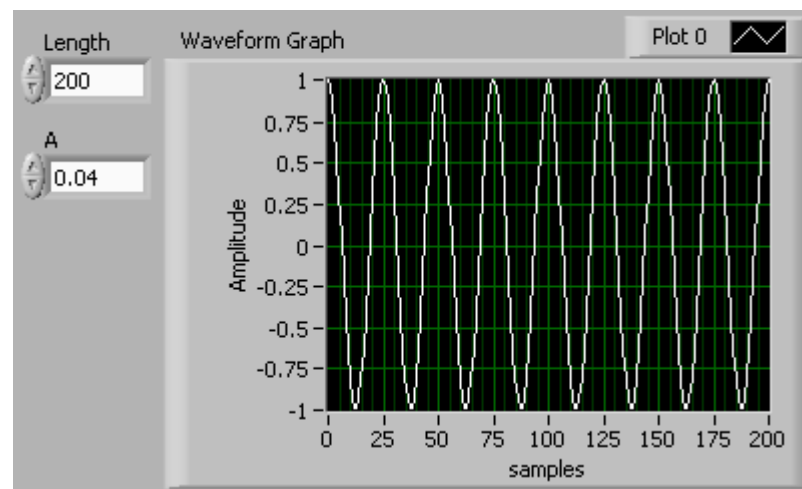


# Math Script čvor

- ❖ Mogu se implementirati tekstualne jednačine i algoritmi.
- ❖ Ulazne i izlazne promenljive se kreiraju na levoj i desnoj granici ” čvora”.
- ❖ Kompatibilan je sa m-file script jezikom.



(Functions»Programming»  
Structures»MathScript)



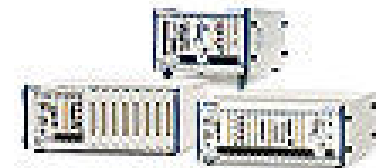
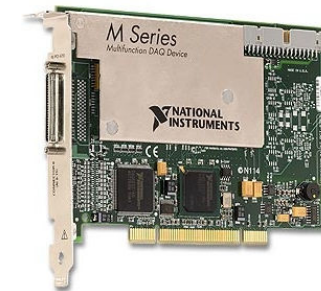
# Akvizicija podataka hardver

Hardver za akviziciju je komponenta koja može da obavlja sledeće funkcije:

- Konverziju ulaznih analognih signala u digitalni format,
- Konverziju digitalnih signala u analogni format (izlaz),
- Generisanje digitalnih izlaznih signala.

- ❖ Uređaji za akviziciju podataka  
(Data Acquisition Device - DAQ)
  - USB, PCI ili PXI uređaji
  - Konfigurišu se i testiraju u MAX-u  
(Measurement & Automation Explorer)

- ❖ Zvučna kartica
  - Ugrađena u većinu računara



# Koji uređaj izabrati?



	Sound Card	NI USB DAQ	NI PCI DAQ
AI Bandwidth	8–44 KS/s	10–200 KS/s	250 K–1.2 Ms/s
Accuracy	12–16 bit	12–16 bit	14–18 bit
Portable	x	x	—
AI Channels	2	8–16	16–80
AO Channels	2	1–2	2–4



## NI USB-6009 SPECIFIKACIJA:

- 8 (14-bit) analognih ulaza
- 12 digitalnih I/O linija
- 2 analogna izlaza

# Akvizicija podataka u LabVIEW

❖ Labview funkcije za akviziciju podataka (Data Acquisition VIs) se mogu podeliti na:

- Express VIs (DAQ Assistant) i
- Standard VIs (DAQmx VIs).

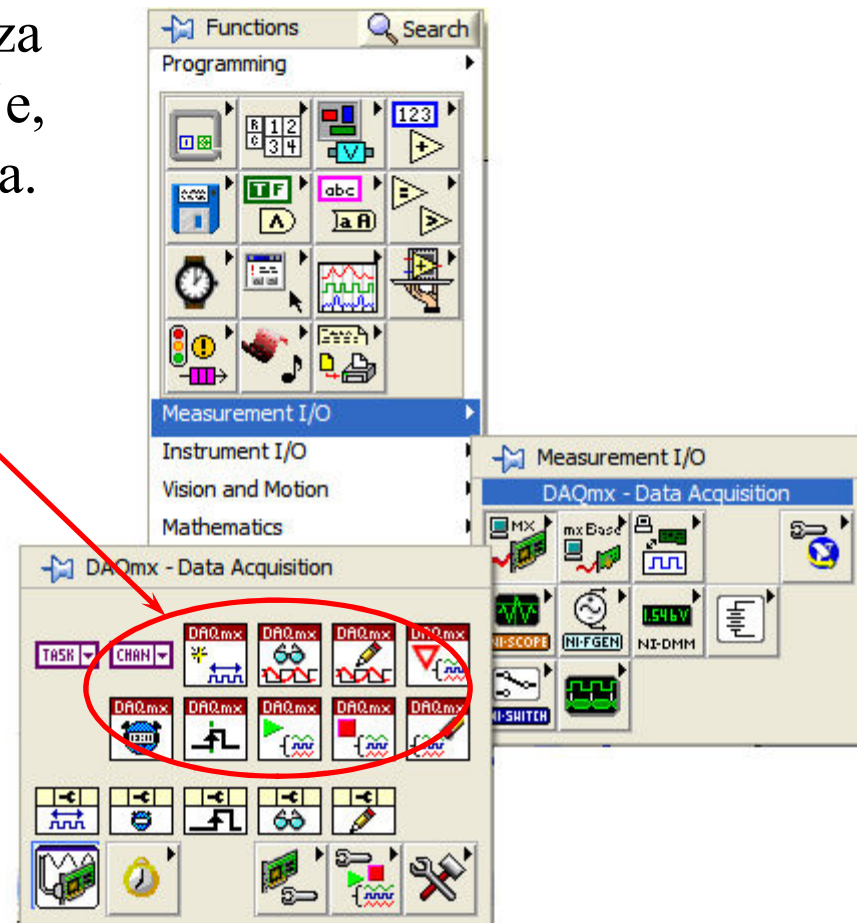
## Express VIs (DAQ Assistant)

- DAQ Assistant konfiguriše DAQ task korak po korak u nekoliko konfiguracionih prozora.



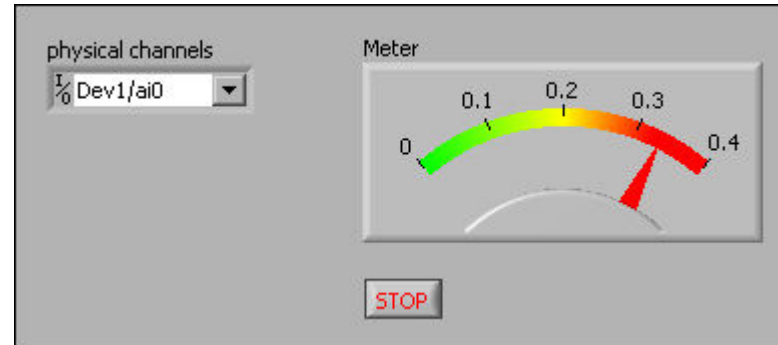
## DAQmx VIs

- Kada je brzina izvršavanja kritična, preporučuje se korišćenje **DAQmx VIs** za akviziciju podataka.
- DAQmx VIs obuhvataju funkcije za **počinjanje** i **zaustavljanje** akvizicije, **čitanje** i **upis** akvizicionih podataka.

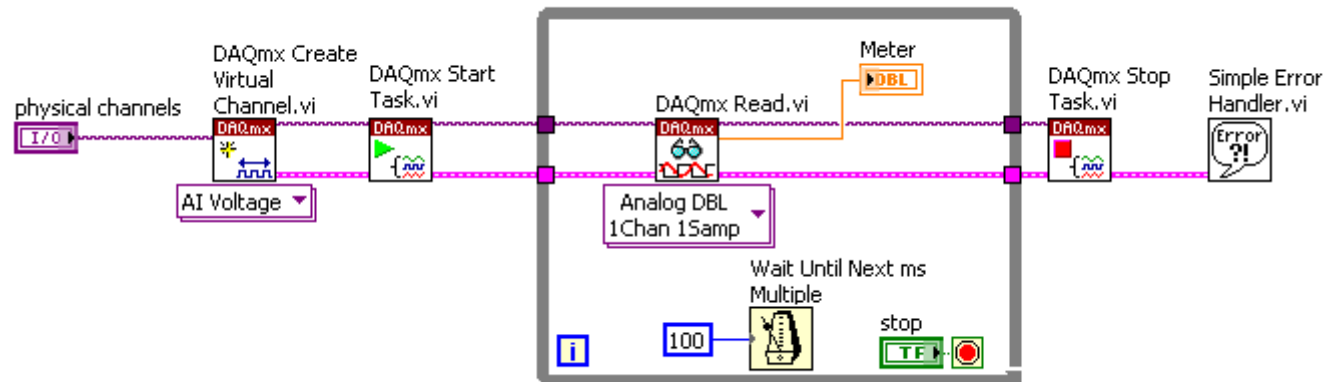


# Kreiranje virtuelnog voltmetra

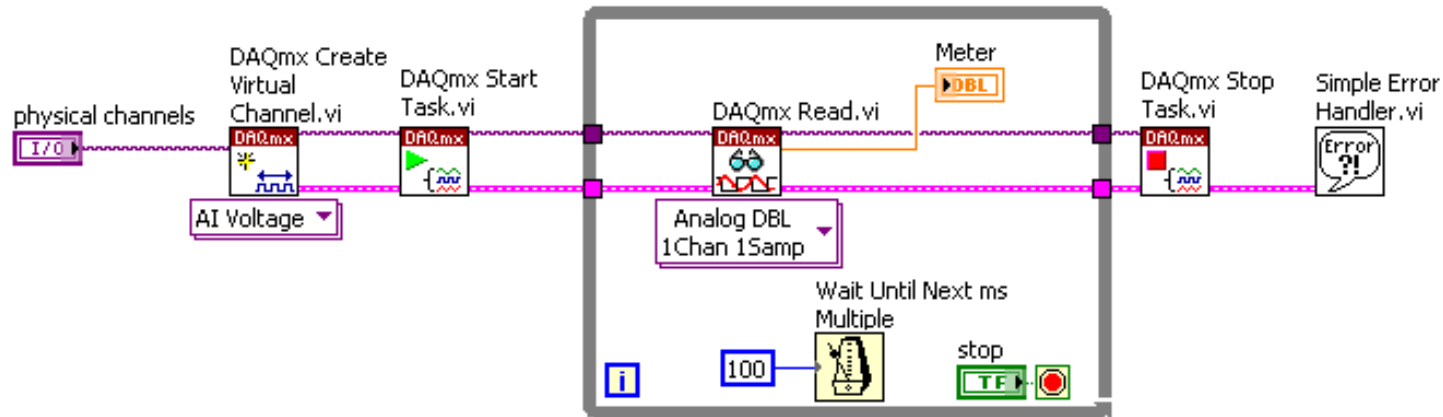
Front panel



Block diagram



# Block diagram



Formira virtuelni kanal (AI Voltage)



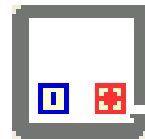
Započinje merenje



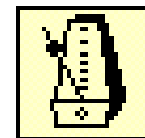
Očitavanje podataka



Zaustavlja merenje



Ponavlja podprogram – While petlja



Wait until next ms multiple



Ukoliko dođe do greške, pojavljuje se dijalog boks sa informacijom o grešci koja se dogodila.



## Zašto koristiti LabVIEW?

- ❖ Jednostavan za učenje
- ❖ Jednostavan za primenu
- ❖ Programski kod je u vidu šeme
- ❖ Nema sintakse!!!
- ❖ Jednostavno i brzo ispravljanje grešaka
- ❖ Prilagodljiv