



Katedra za telekomunikacije

Praktikum softverski alati

LabVIEW

Prof. dr Mirjana Simić-Pejović

kabinet: 108

mira@etf.rs



Šta je LabVIEW?

- “G” programski jezik
- Suštinska razlika u odnosu na ostale programske pakete je da se u LV programi predstavljaju u vidu slika (ikona) umesto programiranja pisanjem komandi u vidu tekstualnog koda
- Proizvod kompanije *National Instruments*

www.ni.com



Zašto koristiti LabVIEW?

- Osnovna prednost LV u odnosu na druga razvojna okruženja je obimna podrška za **komunikaciju sa uređajima** (instrumentacioni hardver).
- LV predstavlja vrlo moćan programski paket, a da su pritom izbegnute sve teškoće i kompleksnosti koje ostali moćni softverski paketi zahtevaju.
- Omogućava analizu ali i formiranje složenih inženjerskih sistema, dok se istovremeno može koristiti i za ozbiljna naučna istraživanja.
- Najčešće primena je za dizajn, testiranje i implementaciju složenih sistema kao i automatizaciju procesa.



Zašto koristiti LabVIEW?

- Jednostavan za učenje
- Jednostavan za primenu
- **Nema sintakse!!!**
- Programski kod je u vidu šeme
- Jednostavan i brz razvoj programa
- Gotovi kontroleri za mnoge uređaje
- Jednostavno i brzo ispravljanje grešaka
- Ne zahteva preveliku brigu o memoriji
- Prilagodljiv
- Zabavan!!!



Mane

- **Cena** (relativno visoka), kao i to da se uopšte plaća
- Može biti težak za savlađivanje naprednih tehnika
- Kôd programa može biti nepregledan i nejasan ukoliko programer ne vodi na računa
- Kreiranje vrlo složenih korisničkih aplikacija može biti vremenski zahtevno
- **Navika!!!**



LabVIEW - karakteristike

- LV programi se zovu virtualni instrumenti jer svojim izgledom podsećaju na fizičke instrumente, kao npr. osciloskope i milimetre. Stoga je ekstenzija programa pisanih u LV

**.vi (virtual instruments)*

- Zbog prvenstvene primene (komunikacija sa uređajima), LV sadrži bogate biblioteke gotovih funkcija za kontrolu mernih instrumenata, kako za prikupljanje tako i za analizu, prezentaciju i skladištenje podataka.



LabVIEW - karakteristike

- LV programi su hijerarhijski i modularni:
 - **Hijerarhijski** su jer se mogu koristiti samostalno, dakle na najvišem nivou, ali i kao potprogrami ili potprogrami unutar potprograma
 - **Modularni** su jer se koncept rešavanja problema, naročito složenih, zasniva na podeli aplikacije na niz zadataka, koji se zatim ponovo dele sve dok se čitav problem ne svede na niz jednostavnih problema.
- Zbog prethodnog, otkrivanje eventualnih grešaka je olakšano, obzirom da se potprogrami mogu izvršavati nezavisno od programa koji ih poziva.



Sadržaj

- “G” jezik
- LV program
 - Front panel
 - Blok dijagram
- LV programiranje

“G” jezik

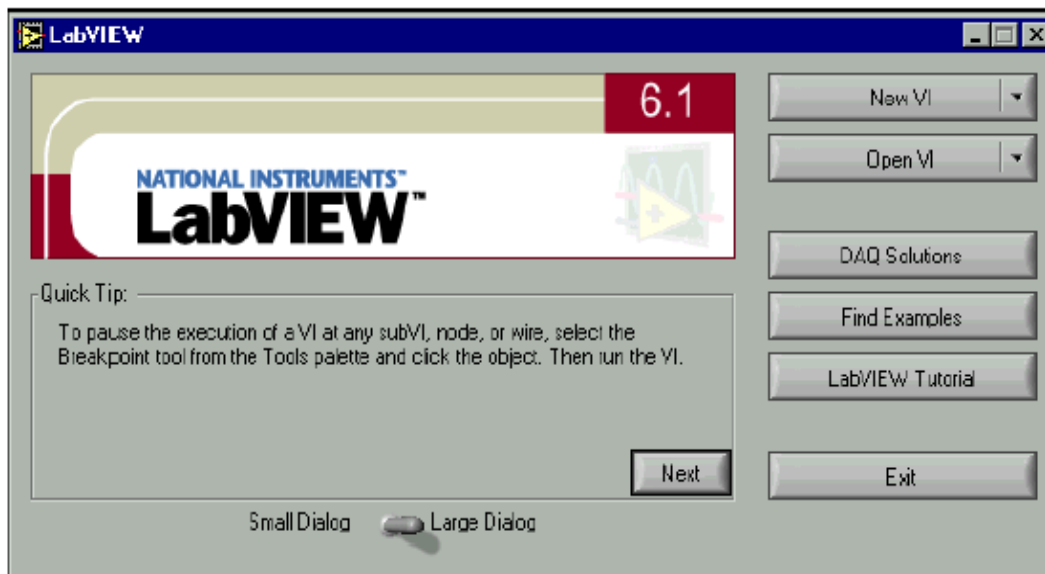
- Kompletno grafičko programiranje
- Kôd programa je **dijagram koji se sastoji od čvorova i žica (konektora)**
- Podaci “putuju” preko žica
- Proces izvršavanja programa kontroliše **protok podataka a ne kôda!**
- Pozicija na dijagramu nije bitna
- Novi način razmišljanja: **dataflow a ne codeflow!**
- **Vodeći princip:** čvor ne izvršava svoju funkciju sve dok podaci, preko žica, ne “stignu” do svih ulaza u isti



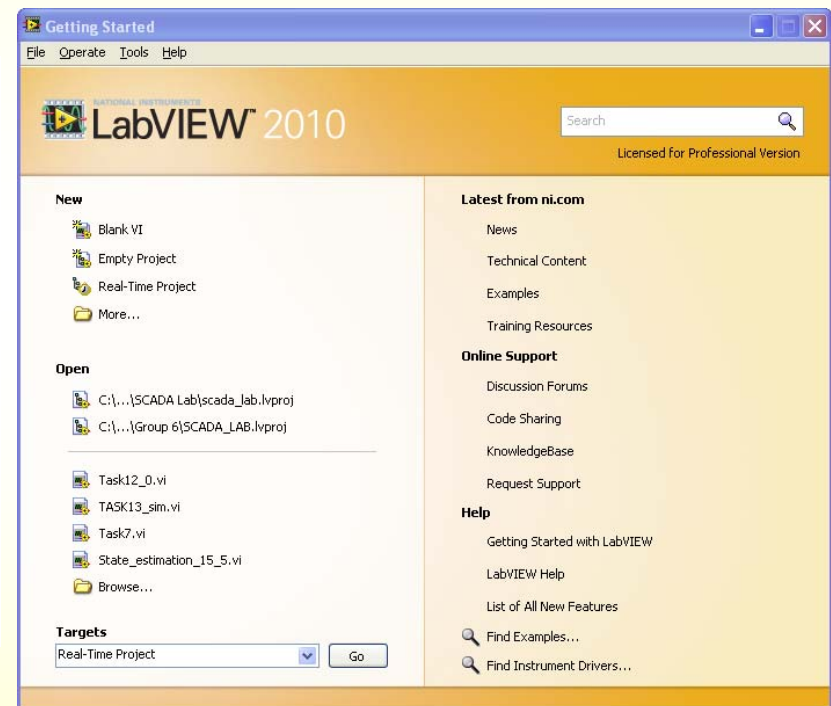
LabVIEW - Getting Started

starije verzije...

Getting Started – starije verzije



Getting Started – novije verzije



LV program

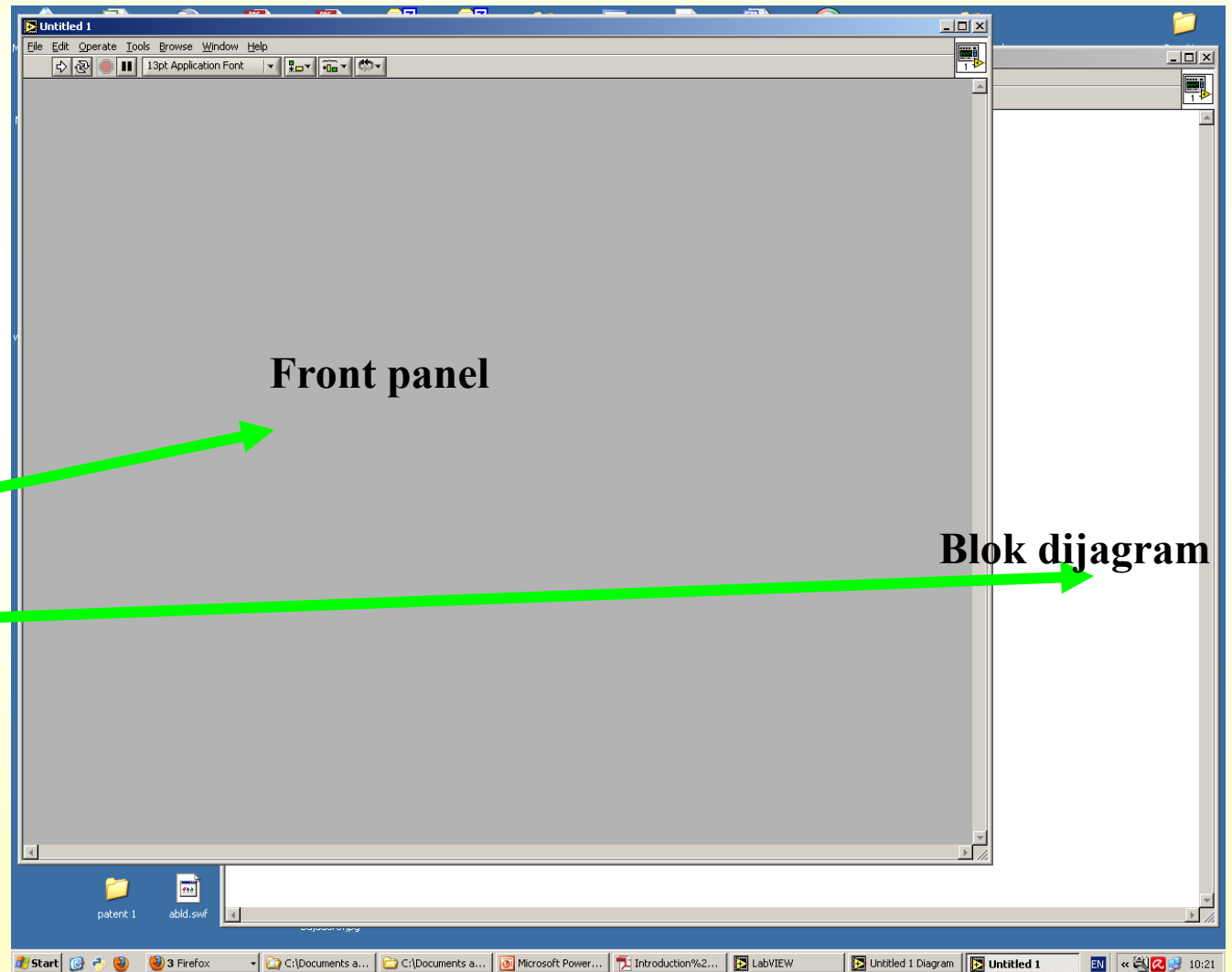
- Programi pisani u LV se zovu virtualni instrumenti i imaju ekstenziju *.vi.
- Svaki vi sastoji se iz dva dela (prozora):
 1. **front panel**
 2. **block diagram**
- Ako se planira korišćenje datog vi programa i kao potprograma, LV program dobija i treći sastavni deo:
 3. **ikone i konektori.**
- Potprogram u LV se zove subvi.



LV program

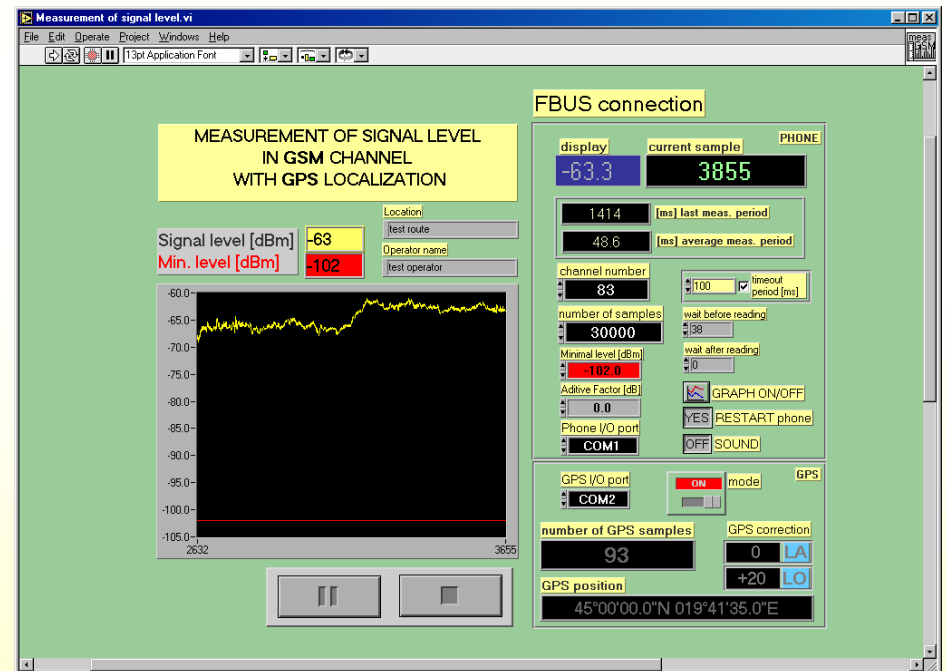
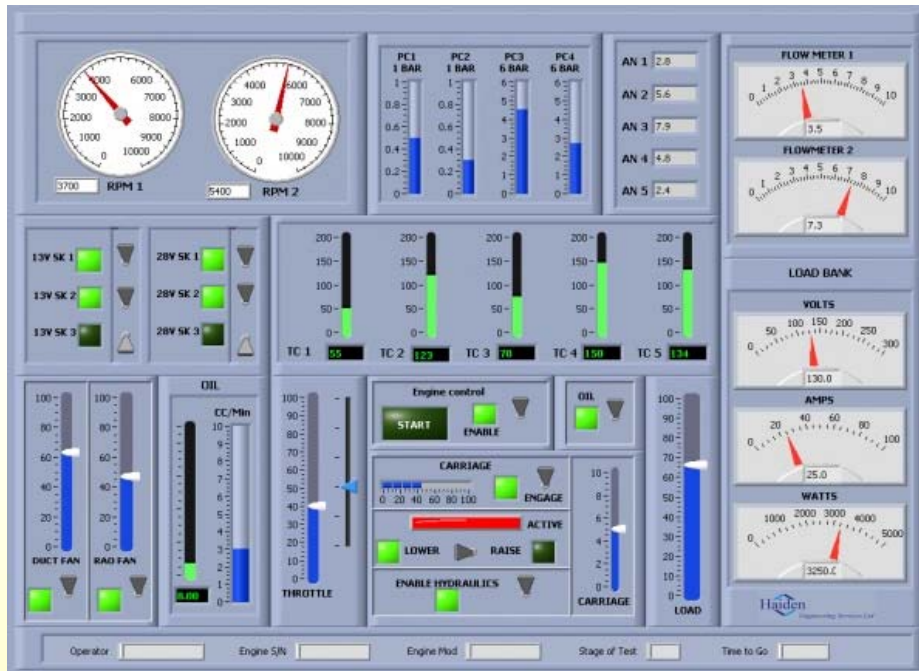
Startovanjem New vi
otvaraju se oba
prozora, pri čemu
je:

1. prvi **front panel**
(sivi)
2. a iza njega **blok
dijagram** (beli).



LV program-*front panel*

- **Front panel** predstavlja korisnički interfejs vi programa.
- Ovaj deo korisnik kreira – npr. svojim izgledom front panel može simulirati panel fizičkog instrumenta (npr. analizatora spektra), ili čitavog mernog sistema.

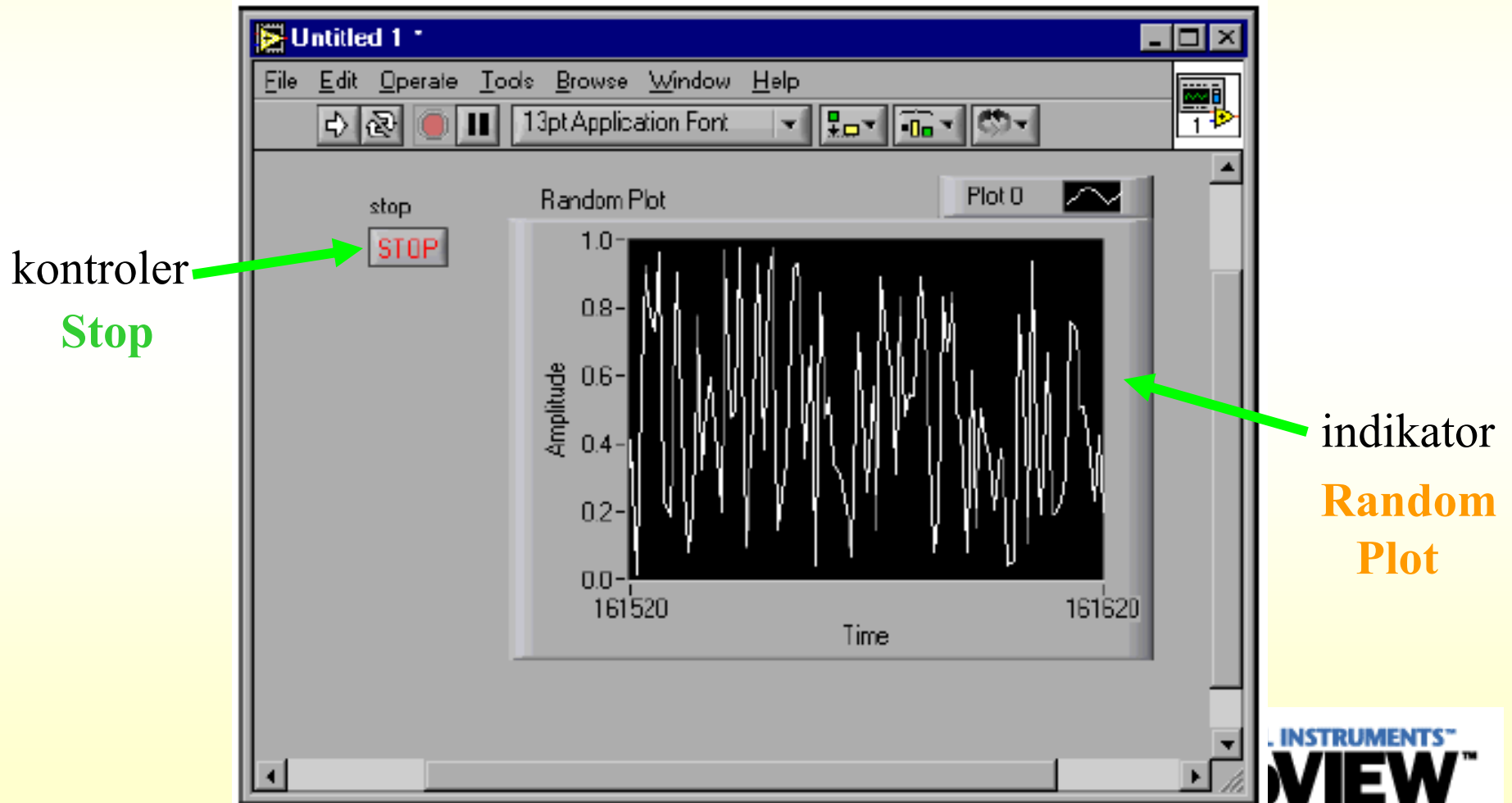


LV program-*front panel*

- **Front panel** sadrži
 - **kontrolere** (preklopnici, tasteri, ...)
 - **indikatore** (grafici, LED, skale, ...)
- Kontroleri predstavljaju **ulazne** terminale, a indikatori **izlazne**.
- Kontroleri simuliraju ulazne delove uređaja i obezbeđuju podatke za *block diagram* **vi** programa.
- Indikatori simuliraju izlazne delove uređaja, i služe za prikaz rezultata iz *block diagrama* **vi** programa.
- Svaki element front panela ima odgovarajući terminal u *block diagramu*.
- Kontroleri i indikatori dostupni su u okviru *controls palette* front panela.

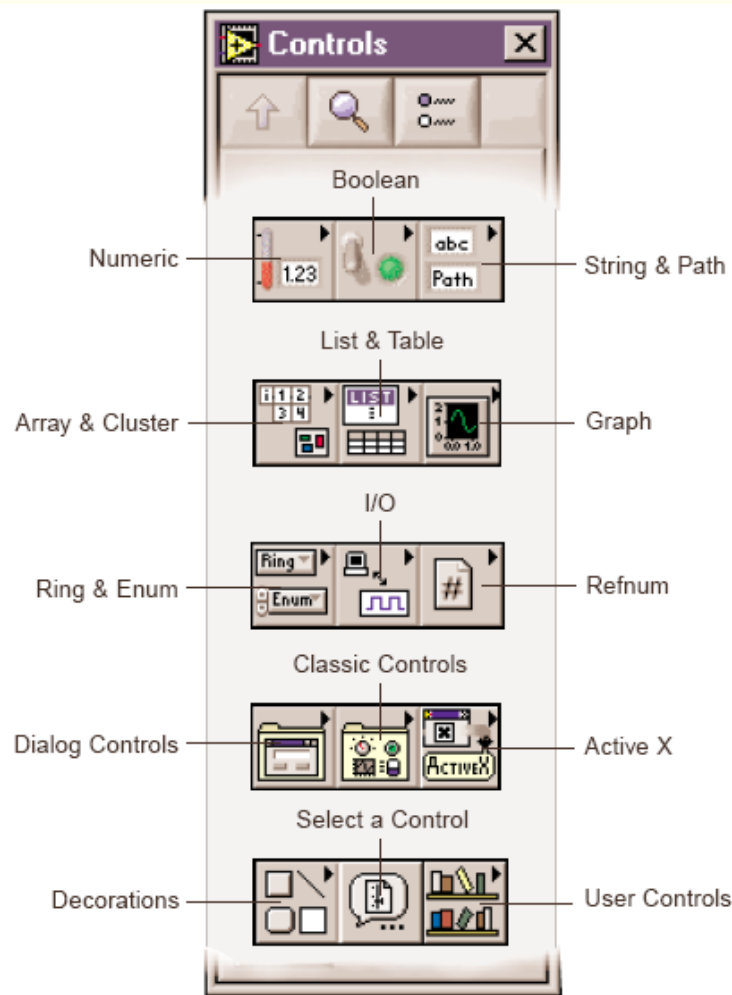
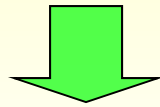


LV program-*front panel*

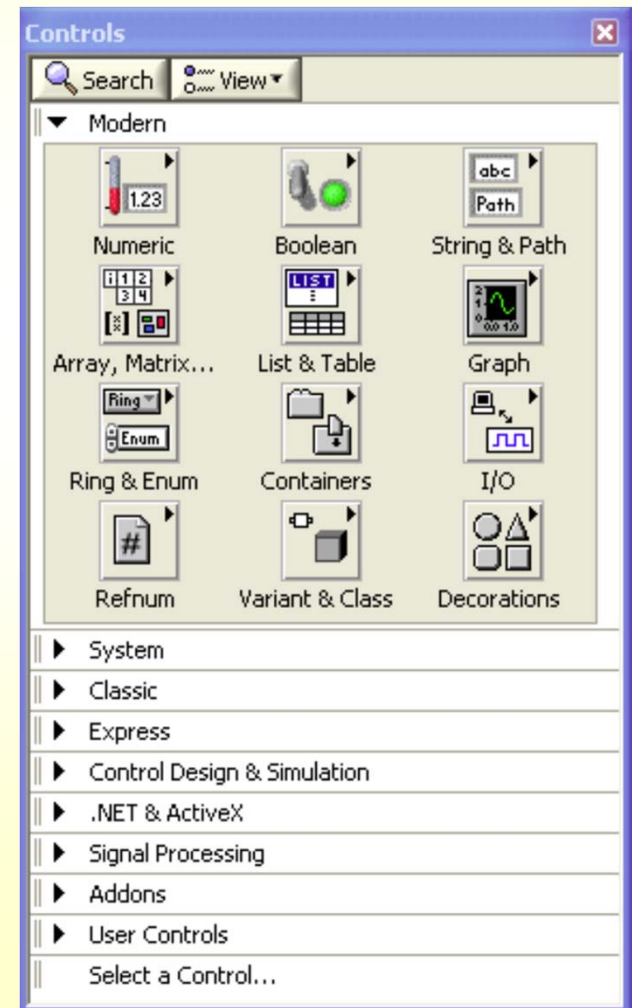
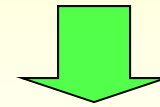


Controls palette – front panel

controls palette – starije verzije

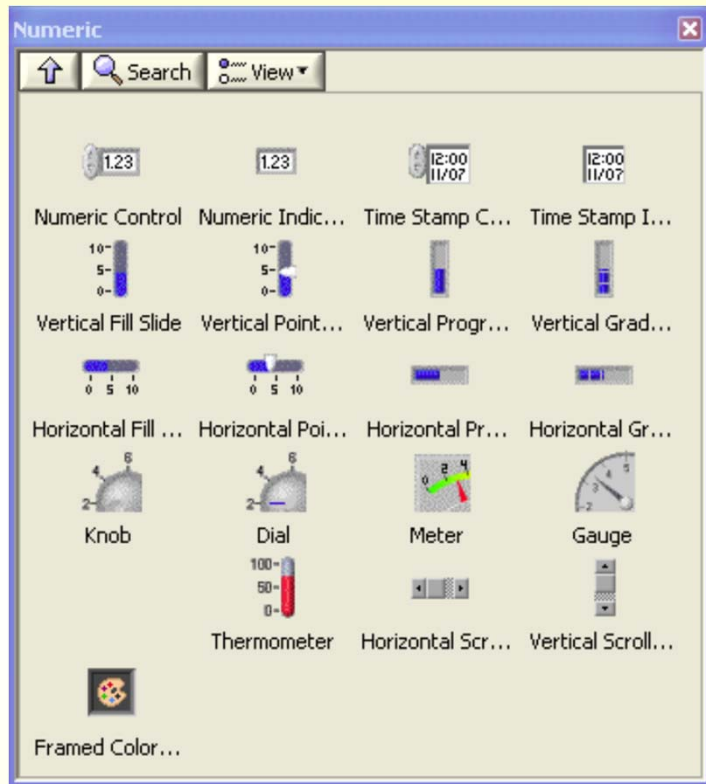


controls palette – novije verzije

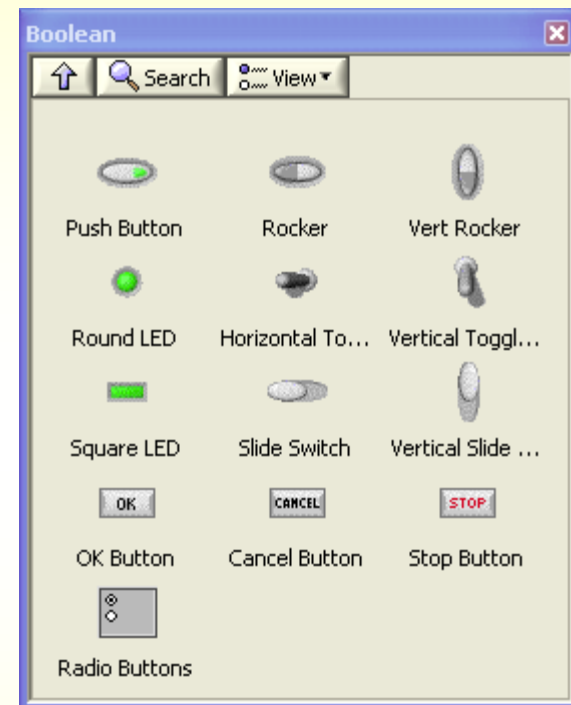


Controls palette – front panel

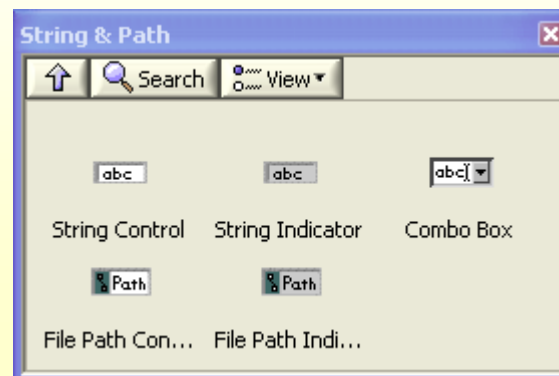
Numeric sub palette



Boolean sub palette



String sub palette



LV program-*block diagram*

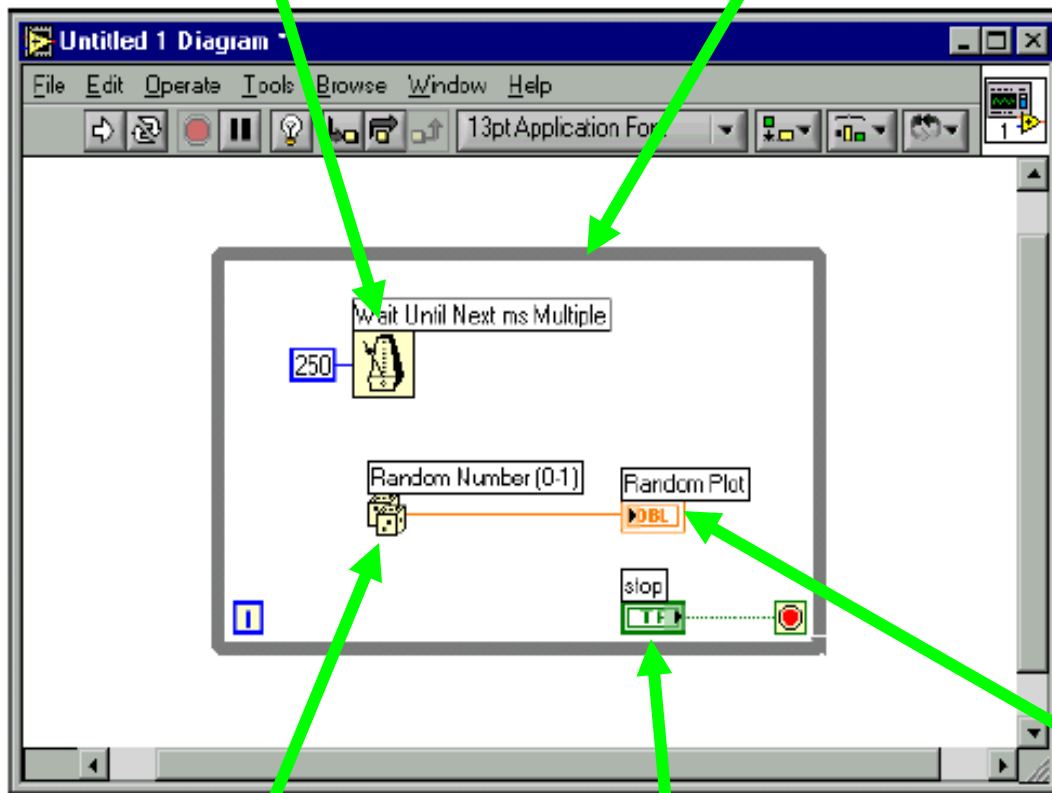
- **Blok dijagram** je grafički prikaz tela programa.
- Sadrži terminale kontrolera i indikatora iz Front panela, kao i ostale čvorove (funkcije, strukture – petlje, potprograme, ...)
- Za razliku od front panela koji je namenjen korisniku, blok dijagram je namenjen programeru.
- Blok dijagram se konstruiše u programskom jeziku “G”.
- Programiranje u blok dijagramu vrši se povezivanjem terminala i korišćenjem glavnog menija blok dijagrama – *Functions palette*.



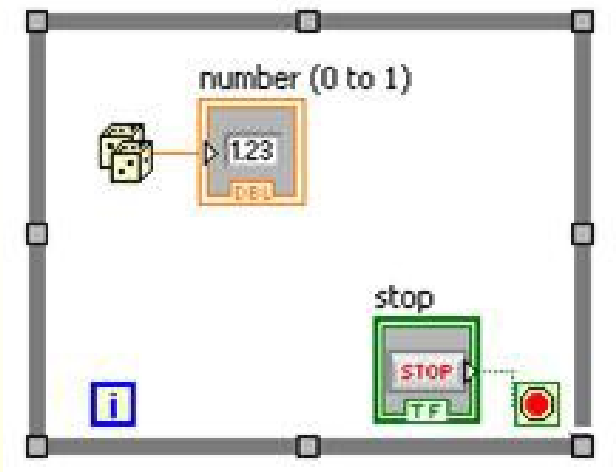
LV program-*block diagram*

vremenska funkcija

While petlja



drugačiji prikaz ikona...



funkcija za generisanje slučajnih brojeva

terminal kontrolera

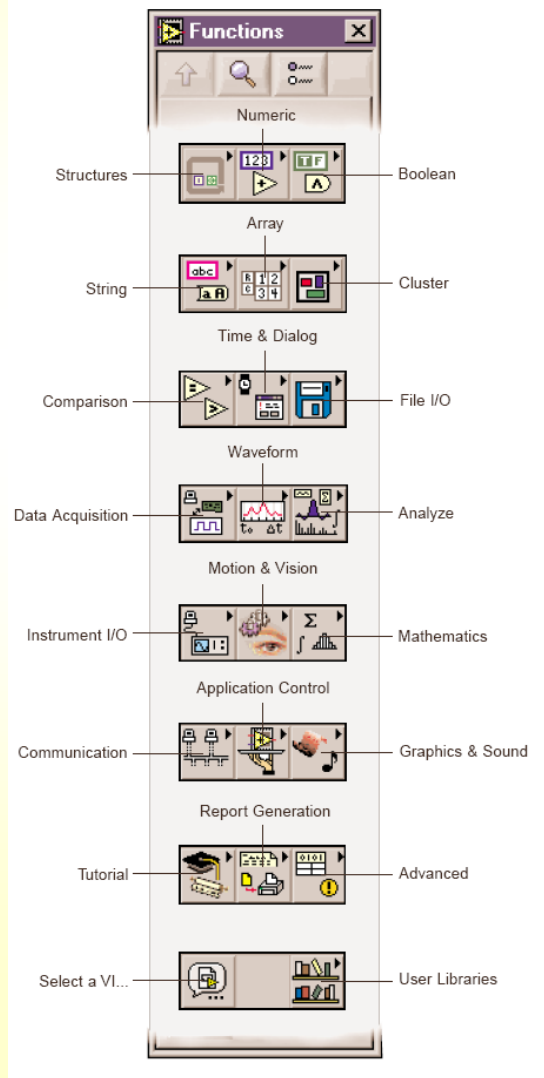
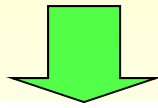
Stop

terminal indikatora

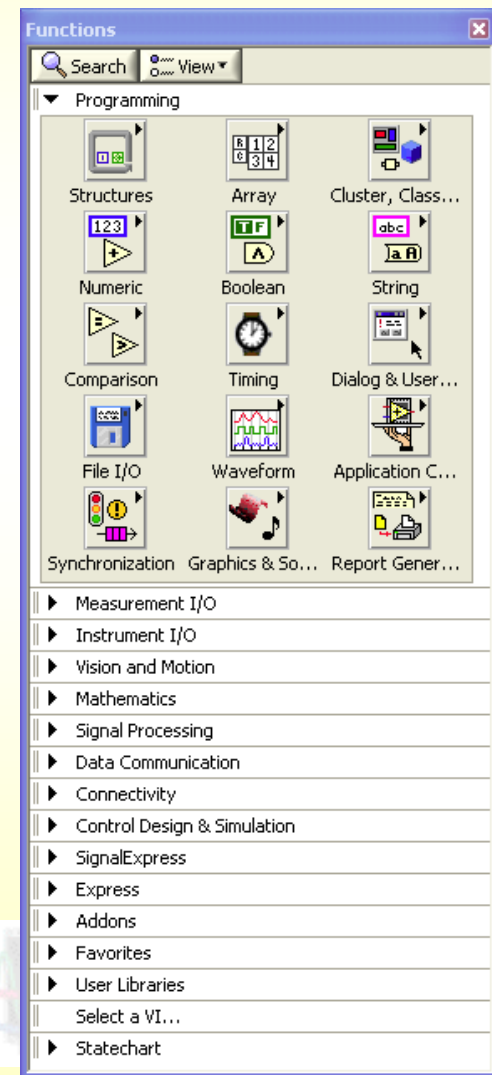
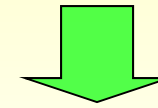
Random Plot

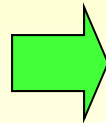
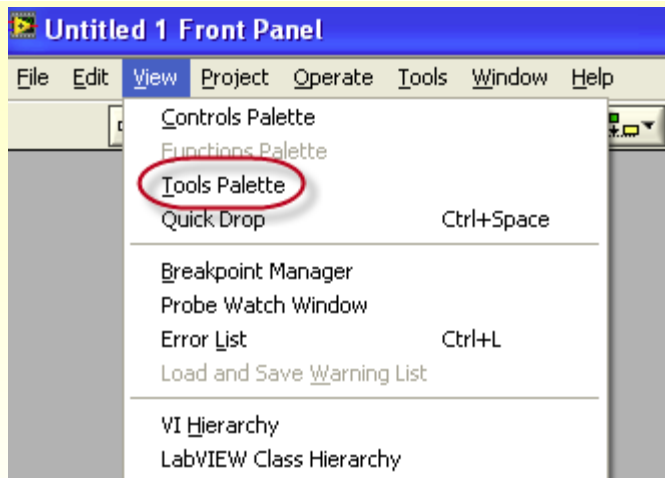
Functions palette - block diagram

functions palette – starije verzije



functions palette – novije verzije





Operating Tool—Changes the value of a control or selects the text within a control.



Positioning Tool—Positions, resizes, and selects objects.



Labeling Tool—Edits text and creates free labels.



Wiring Tool—Wires objects together on the block diagram and connects control to connector pane.



Object Shortcut Menu Tool—Opens the shortcut menu of an object.



Scroll Tool—Scrolls the window without using the scroll bars.



Breakpoint Tool—Sets breakpoints on VIs, functions, wires, loops, sequences, and cases.



Probe Tool—Creates probes on wires.



Color Copy Tool—Copies colors for pasting with the Color Tool.



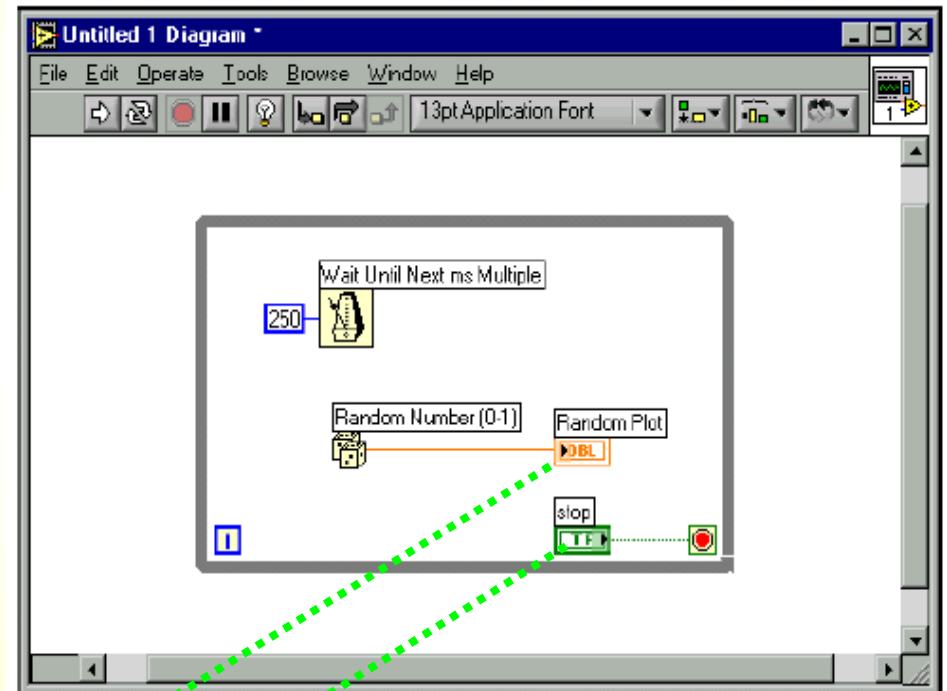
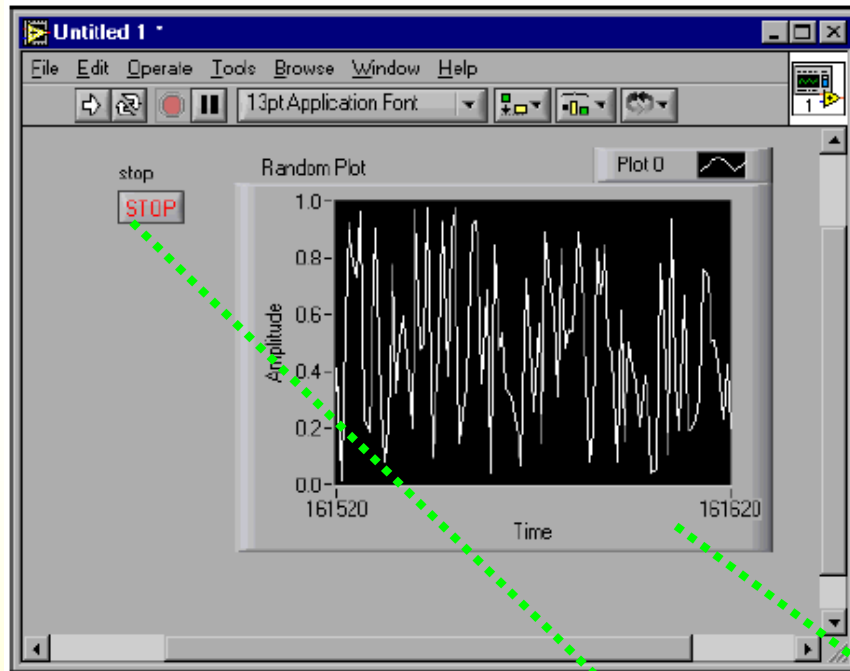
Color Tool—Sets the foreground and background colors.

Tools palette *front panel i block diagram*

Front panel i block diagram

Front panel

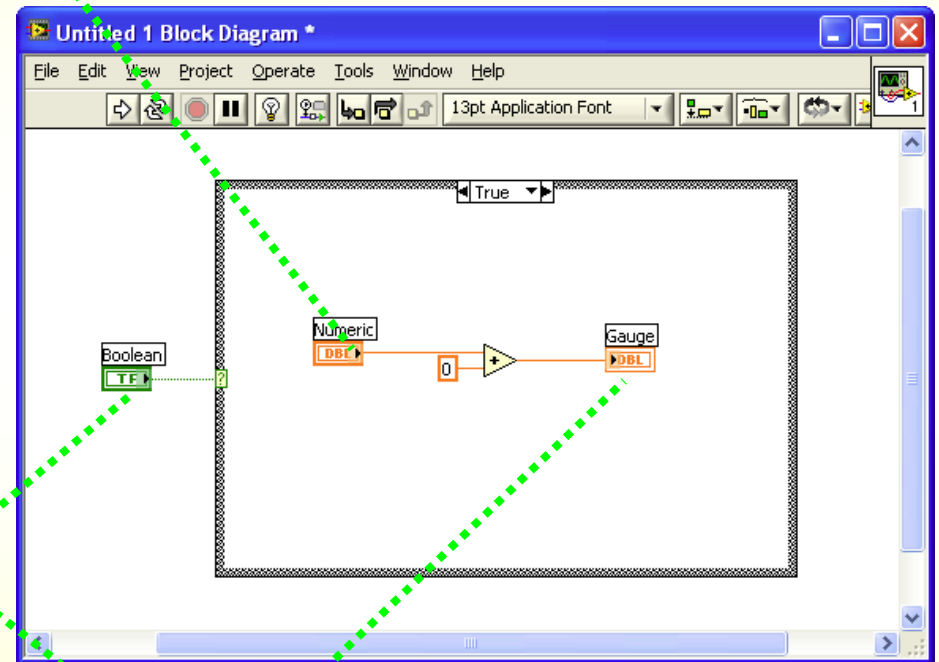
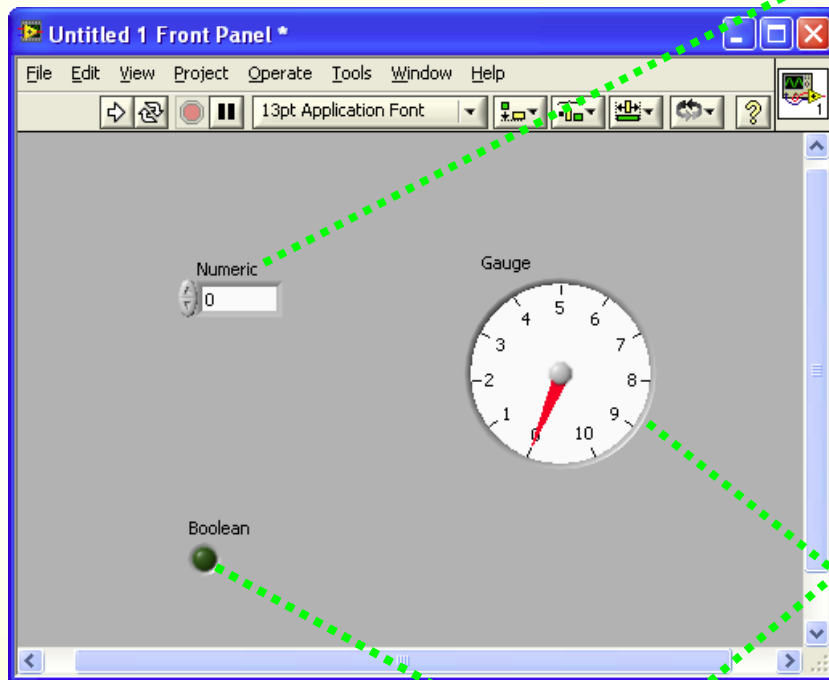
Block Diagram



Front panel i block diagram

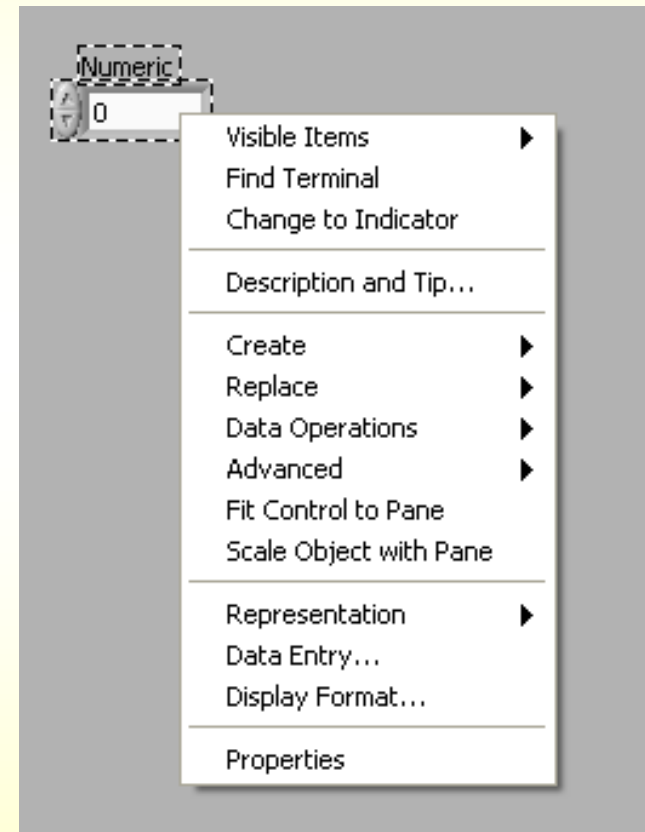
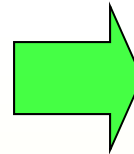
Front panel

Block Diagram




The Objects short-cut menu

- Svi objekti na front panelu (indikatori i kontroleri) i blok dijagramu (terminali, funkcije i ostali čvorovi), kao i sam prazni deo front panela (siva zona) i blok dijagrama (bela zona) mogu se podešavati – menjati im se izgled ili način funkcionisanja.
- To se postiže korišćenjem short-cut menija objekata, koji je dostupan desnim klikom na željeni objekat.

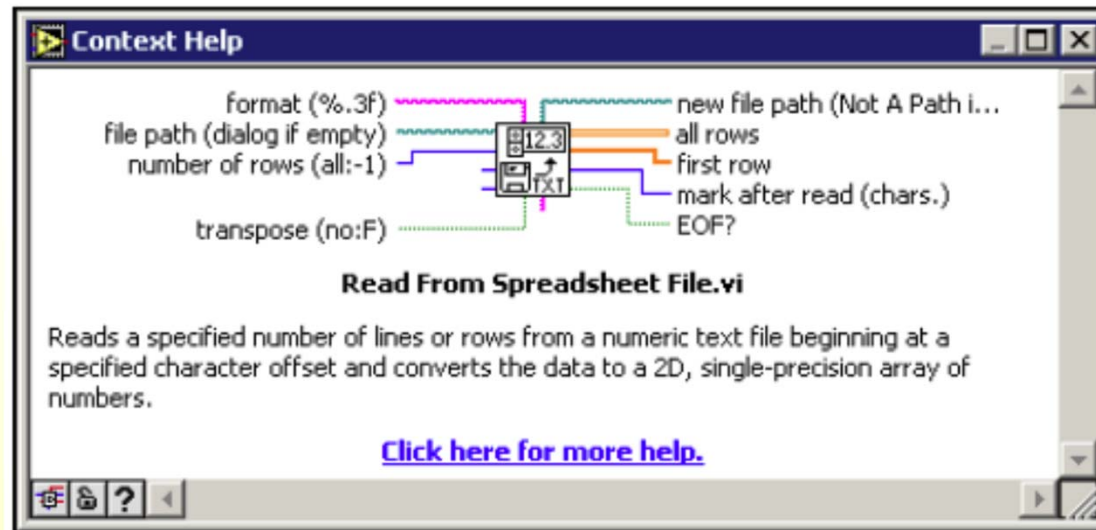


NATIONAL INSTRUMENTS™
LabVIEW™

Context Help

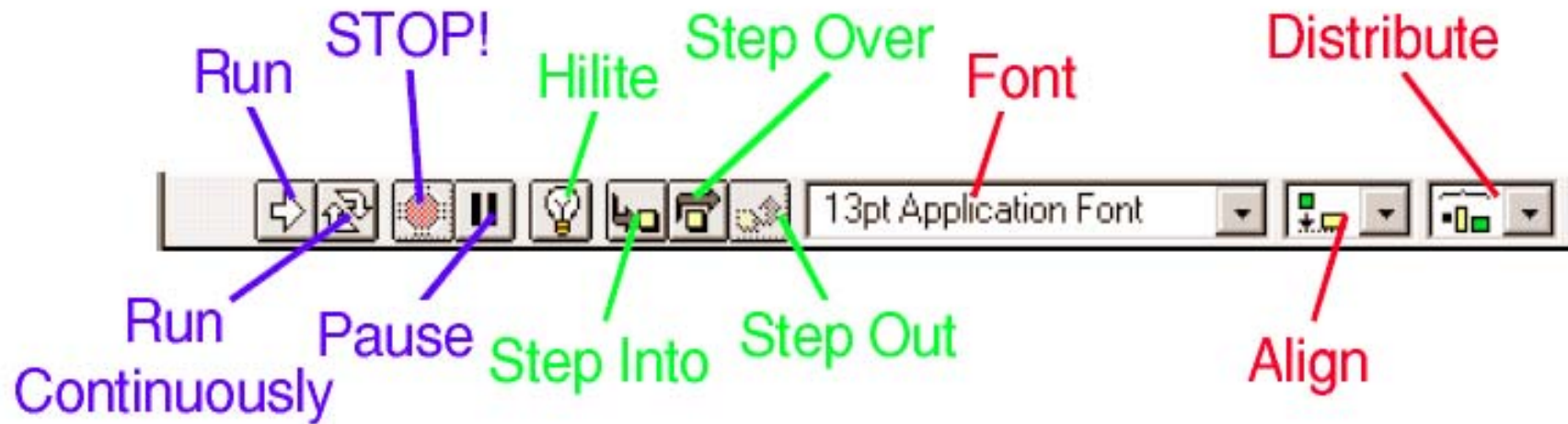
 The Context Help window (Ctrl +H)

- **Context Help** prozor prikazuje osnovne informacije o objektu na koji se pozicioniramo na front panelu ili blok dijagramu.
- Vrlo je koristan jer oslobađa programera pamćenja velikog broja različitih ikona tj. objekata.



Help-Show Context Help, ili Ctrl-H

Run mod LV programa

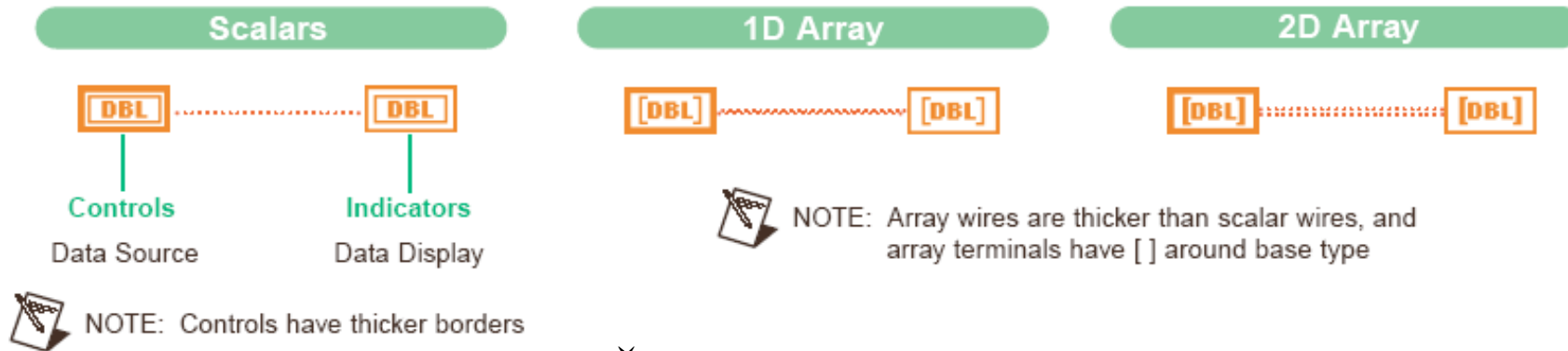


Dobar VI!



Loš VI!

Tipovi žica



Žice se razlikuju po:

- boji (jedinствena za određeni tip podataka)
- debljini



Numerički podaci

Logički podaci

String podaci

Tipovi terminala

Signed Integers

8-bit 

16-bit 

32-bit 

Unsigned Integers

8-bit 


16-bit 

32-bit 

Real Floating-Point

Single 

Double 

Extended 

Complex Floating-Point

Single 

Double 

Extended 


Boolean 


String 

Path 

Variant 

Refnum 

Cluster of numerics 

Cluster of mixed data type 

Waveform 

Polymorphic 

I/O Name Control 

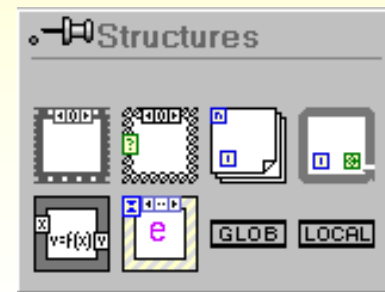
Terminali za kontrolere i indikatore su isti samo što kontroleri imaju deblji okvir terminala



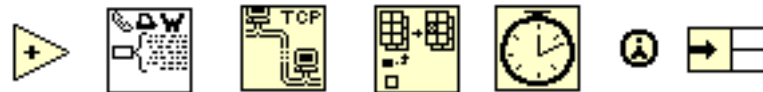
NATIONAL INSTRUMENTS™
LabVIEW™

Tipovi čvorova

- Strukture



- Funkcije

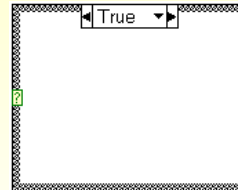


- *User VIs*



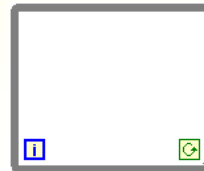
Struktura

- *Case* struktura

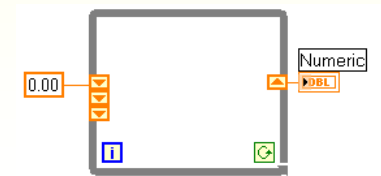


if...then...else

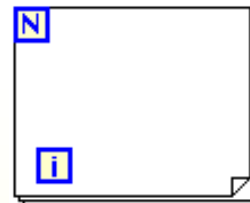
- *While* petlja



While sa shift registrima

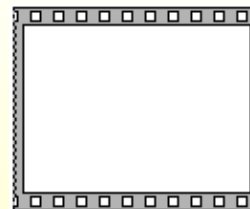


- *For* petlja

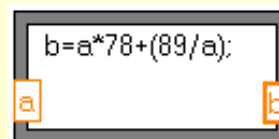


$i = 0$ to $N-1$

- Sekvenca



- *Formula Node*



Funkcije

- Niskog nivoa
 - Aritmetičke
 - Bulove
 - Funkcije poređenja
- Srednjeg nivoa
 - Funkcije za manipulaciju nizovima
 - Funkcije za manipulaciju stringovima
 - Vremenske funkcije

Funkcije

- Visokog nivoa
 - *File* I/O
 - Funkcije za komunikaciju sa uređajima (GPIB, serial, ...)
 - Funkcije za komunikaciju sa mrežom (TCP...)
 - Funkcije za analizu i obradu signala (FFT, filtri, ...)
 - Funkcije za zvuk i grafiku
 - Funkcije kontrole programa
 - *Advanced* funkcije



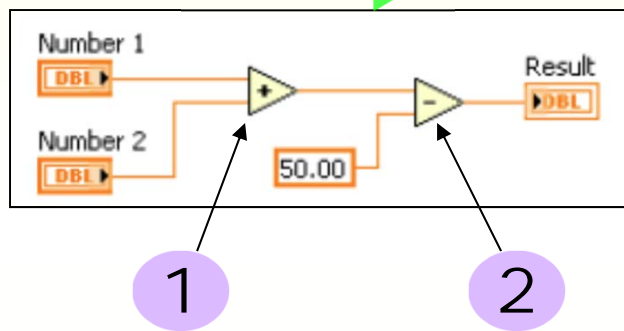
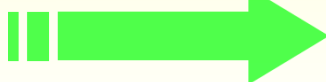
Dataflow programming

- Kao što je već rečeno, LV koristi princip *dataflow* (protok podataka) umesto *codeflow* programiranja (protok kôda).
- *Dataflow* programiranje zasniva se na principu da se čvor izvršava onda kada su mu dostupni svi ulazni podaci.
- Kada taj čvor izvrši svoju funkciju, podaci na njegovom izlazu snabdevaju ulaze sledećeg čvora, itd...
- Kod programskih jezika koji se baziraju na *codeflow* principu (C, C++, Java, ...) izvršenje programa kontrolisano je sekvencijalnim redosledom izvršavanja delova programa.



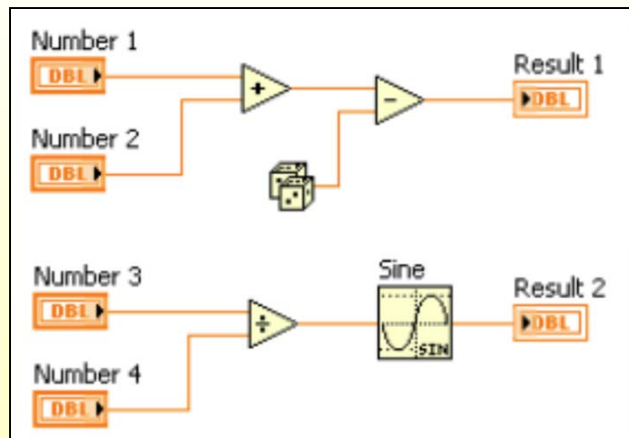
Dataflow programming

Dataflow



```
C:\matlab\work\sigproc\noisemaker.m
[none time use
[ini,fsl,nbitsl] = wavread('gamma591.wav');
right = inl(:,1)';
nsamples = length(right);
pause;
right = morsegen(1000,1,1); %code go wings
nsamples = length(right);
code2 = morsegen_code2(3000,.25,.25); %mmm bop interference
ncode2samples = length(code2);
if (ncode2samples < nsamples),
    code2 = cat(2,code2,zeros(1,nsamples-ncode2samples));
else
    right = cat(2,right,zeros(1,ncode2samples-nsamples));
end
wnoise = 0.2*randn(size(right));
ttt = 0:1/fsl:nsamples/fsl-1/fsl;
sixty = 1.0*cos(2*pi*60*ttt);
rn = .5*right + wnoise + 2.0*sixty + code2;
```

Codeflow



Šta se ovde prvo izvršava???



Application Builder

- Softver koji se primenjuje na gotov LabVIEW program.
- U starijim verzijama AppB se kupovao i instalirao nezavisno od LV, ali u novijim verzijama (6,7...) ugrađen je u instalaciju LV.
- Formira izvršnu aplikaciju (.exe) koja se može pokrenuti i na računarima na kojima nije instaliran LabVIEW.
- Poslednja priprema gotovog programa pre nego što se uruči korisniku.
- Omogućava korišćenje LV programa samo kao izvršne aplikacije bez mogućnosti uvida u blok dijagram i bez mogućnosti izmena i dorada gotovog programa od strane korisnika.



Hvala na pažnji!

