

Kratak opis softvera

LabVIEW



Šta je LabVIEW?

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

www.ni.com



Zašto koristiti LabVIEW?

- LV predstavlja vrlo moćan programski paket, a da su pritom izbjegnute sve teškoće i kompleksnosti koje ostali moćni softverski paketi zahtijevaju
- Omogućava analizu ali i formiranje složenih inženjerskih sistema, dok se istovremeno može koristiti i za ozbiljna naučna istraživanja
- Najčešće primjena je za dizajn, testiranje i implementaciju složenih sistema, sa procijenjenim smanjenjem vremena razvoja a povećanjem produktivnosti za faktor 4 do 10



Zašto koristiti LabVIEW?

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



Mane

- Cijena (relativno visoka)
- Može biti težak za savlađivanje naprednih tehnika
- Kod programa može biti nepregledan i nejasan ukoliko programer ne vodi računa
- Kreiranje vrlo složenih korisničkih aplikacija može biti vremenski zahtjevno
- Navika!!!



LabVIEW - karakteristike

- LV programi se zovu virtualni instrumenti jer svojim izgledom podsjećaju na fizičke instrumente, kao npr. osciloskope i multimetre. Stoga je ekstenzija programa pisanih u LV
**.vi (virtual instruments)*
- Zbog prvenstvene primjene (mjerni uređaji), LV sadrži bogate biblioteke gotovih funkcija za kontrolu mjernih 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 rješavanja problema, naročito složenih, zasniva na podjeli aplikacije na niz zadataka, koji se zatim ponovo dijele 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

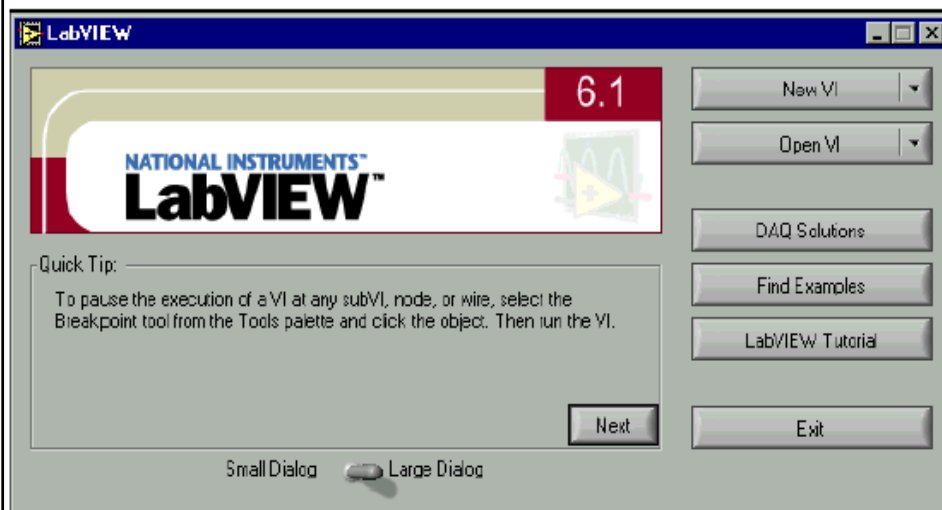


“G” jezik

- Kompletno grafičko programiranje
- Kod programa je dijagram koji se sastoji od čvorova i žica
- Podaci “putuju” preko žica
- Proces izvršavanja programa kontroliše protok podataka a ne koda!
- 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-start



LV program

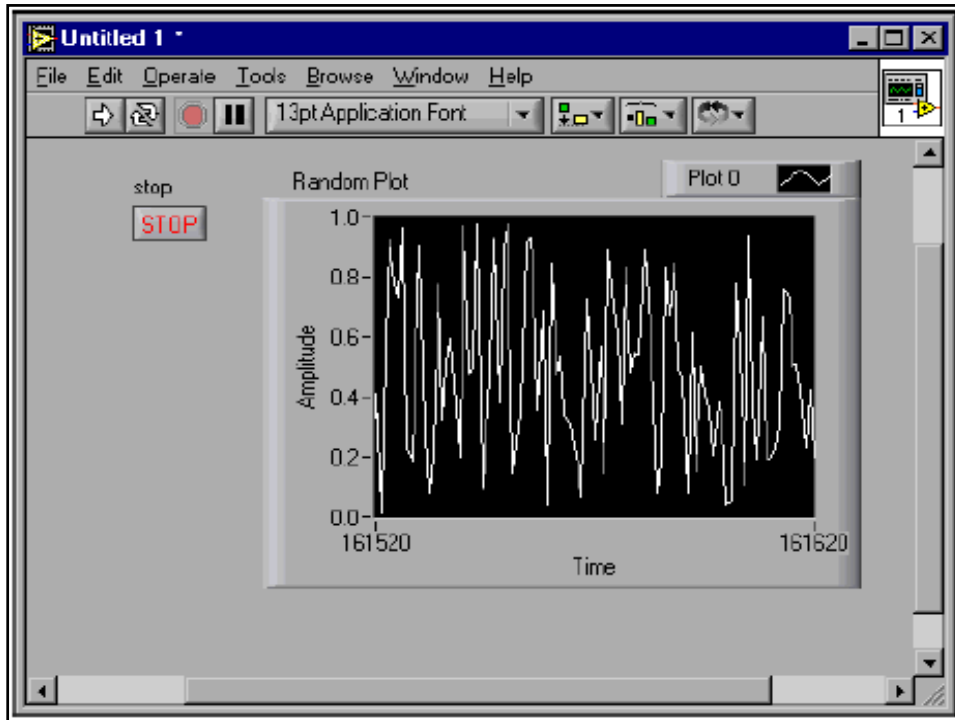
- Programi pisani u LV se zovu virtualni instrumenti i imaju ekstenziju *.vi
- Svaki vi sastoji se iz dva dela:
 - *front panel*
 - *block diagram*
- Ako se planira korišćenje datog vi programa i kao potprograma, LV program dobija i treći sastavni deo: ikone i konektori. Potprogram u LV se zove **subvi**.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



LV program-*front panel*

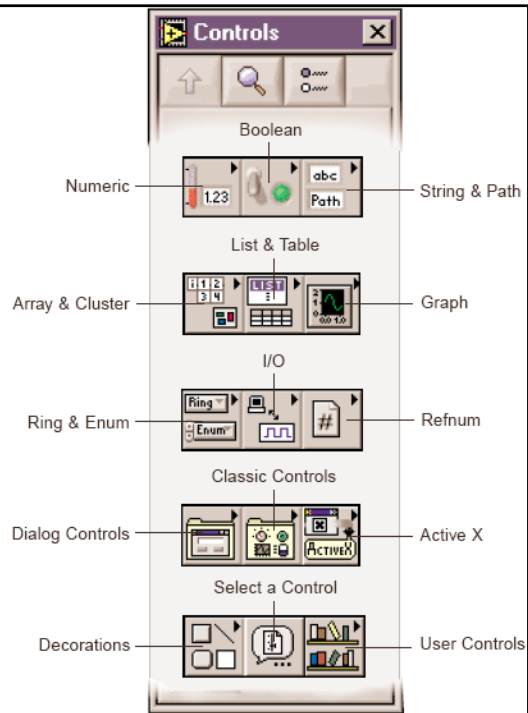
- Front panel predstavlja korisnički interfejs programa
- Ovaj dio simulira panel fizičkog instrumenta
- Front panel sadrži
 - kontrolere (preklopnici, tasteri, ...)
 - indikatore (grafici, LED, skale, ...)
- Kontroleri predstavljaju ulazne terminale, a indikatori izlazne
- Kontroleri simuliraju ulazne 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





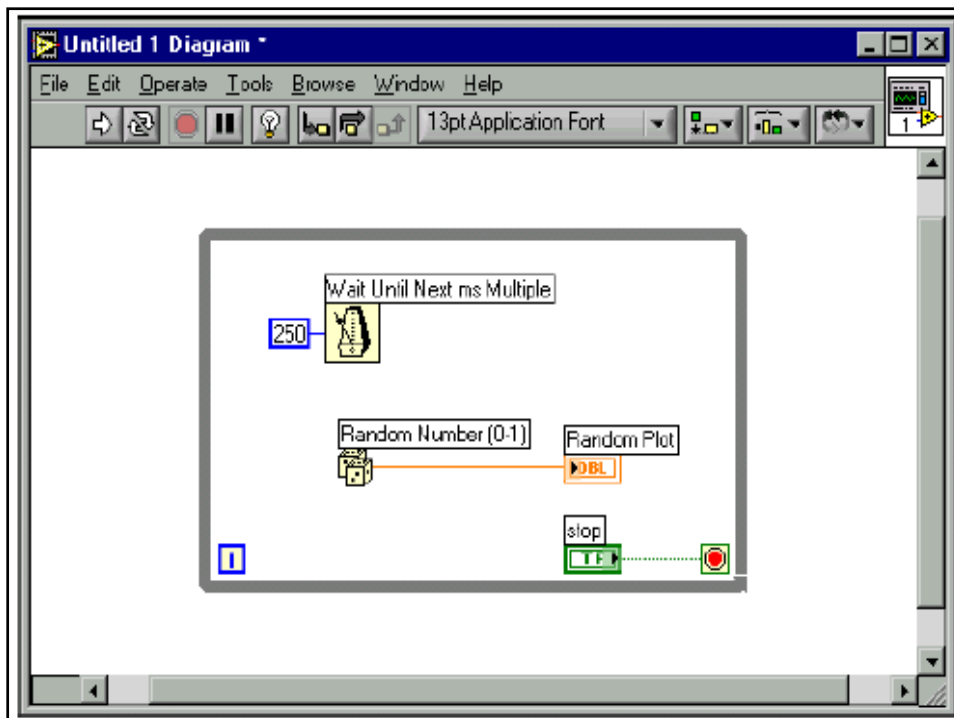
LV program
front panel
 tipovi kontrolera i
 indikatora

controls palette



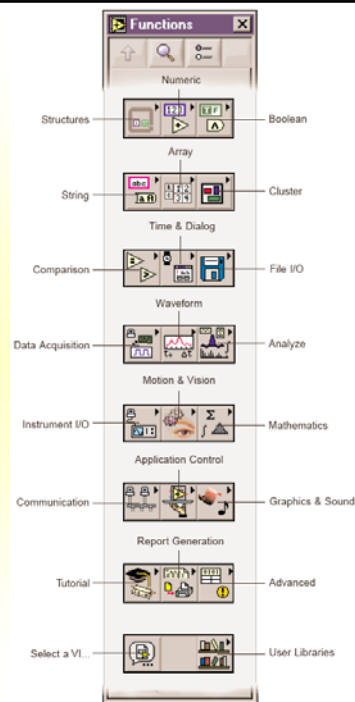
LV program-*block diagram*

- Blok dijagram je grafički prikaz tijela programa
- Za razliku od front panela koji je namijenjen korisniku, blok dijagram je namijenjen 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











functions palette



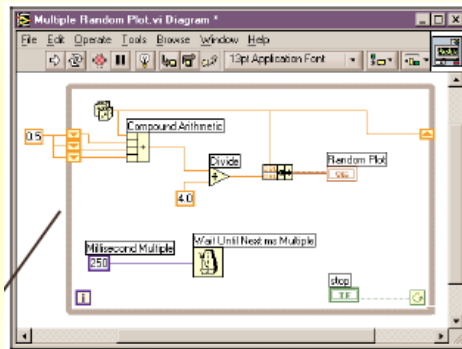
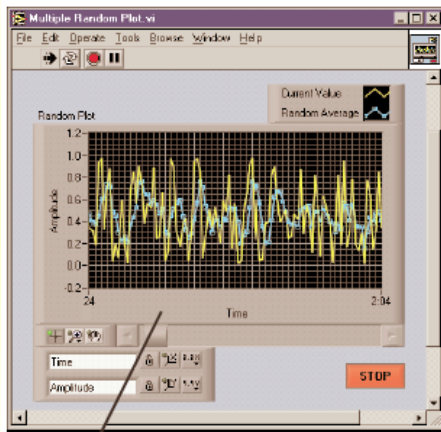
LV program
front panel i block diagram

tools palette

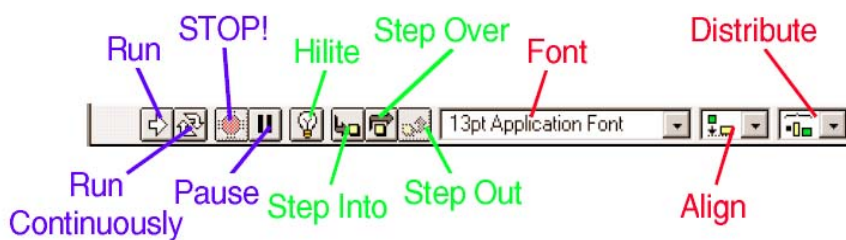



-  **Operating Tool**—Changes the value of a control or selects the text within a control.
-  **Positioning Tool**—Positions, resizes, and selects objects.
-  **Labeling Tool**—Edits text and creates free labels.
-  **Wiring Tool**—Wires objects together on the block diagram and connects control to connector 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.


Front panel i block diagram



Run mod LV programa

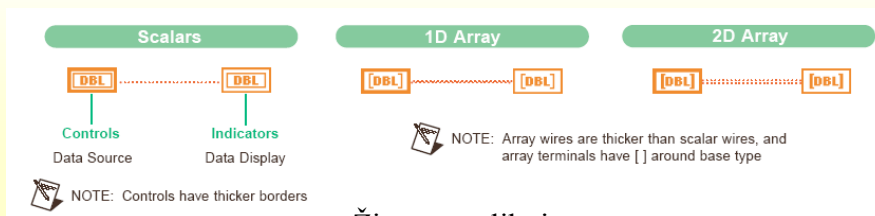


 Dobar VI!

 Loš VI!

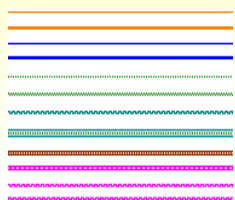


Tipovi žica



Žice se razlikuju po:

- boji (jedinstvena za određeni tip podataka)
- debljini



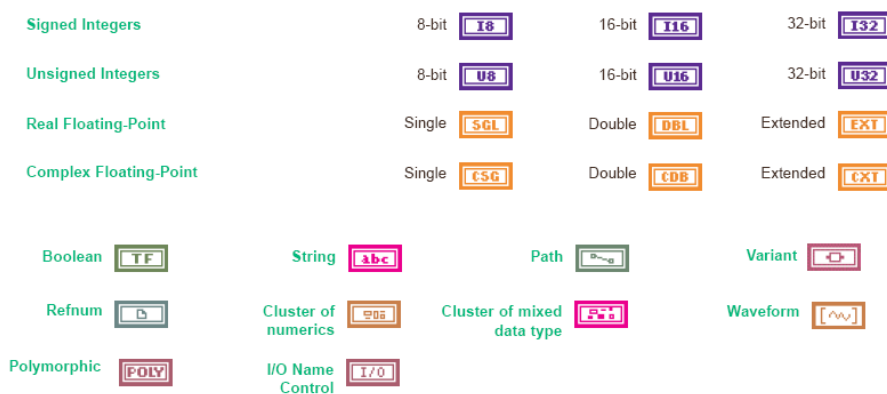
Numerički podaci

Logički podaci

String podaci



Tipovi terminala

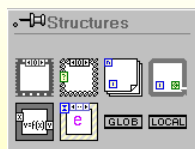


Terminali za kontrolere i indikatore su isti samo što kontroleri imaju deblji okvir terminala



Tipovi čvorova

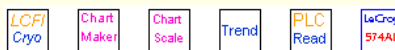
- Strukture



- Funkcije



- *User VIs*



Strukture

- *Case* struktura



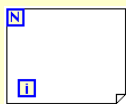
- *While* petlja



While sa shift registrima



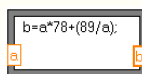
- *For* petlja



- 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



Application Builder

- Softver koji se primjenjuje 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
- Posljednja priprema gotovog programa prije 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

