

RAČUNALNIŠKA ORODJA

Uvodno predavanje

Laboratorij PRAKTIKUM ZA ELEKTRONIKO
PE soba B-304

Računalniška orodja - uvod

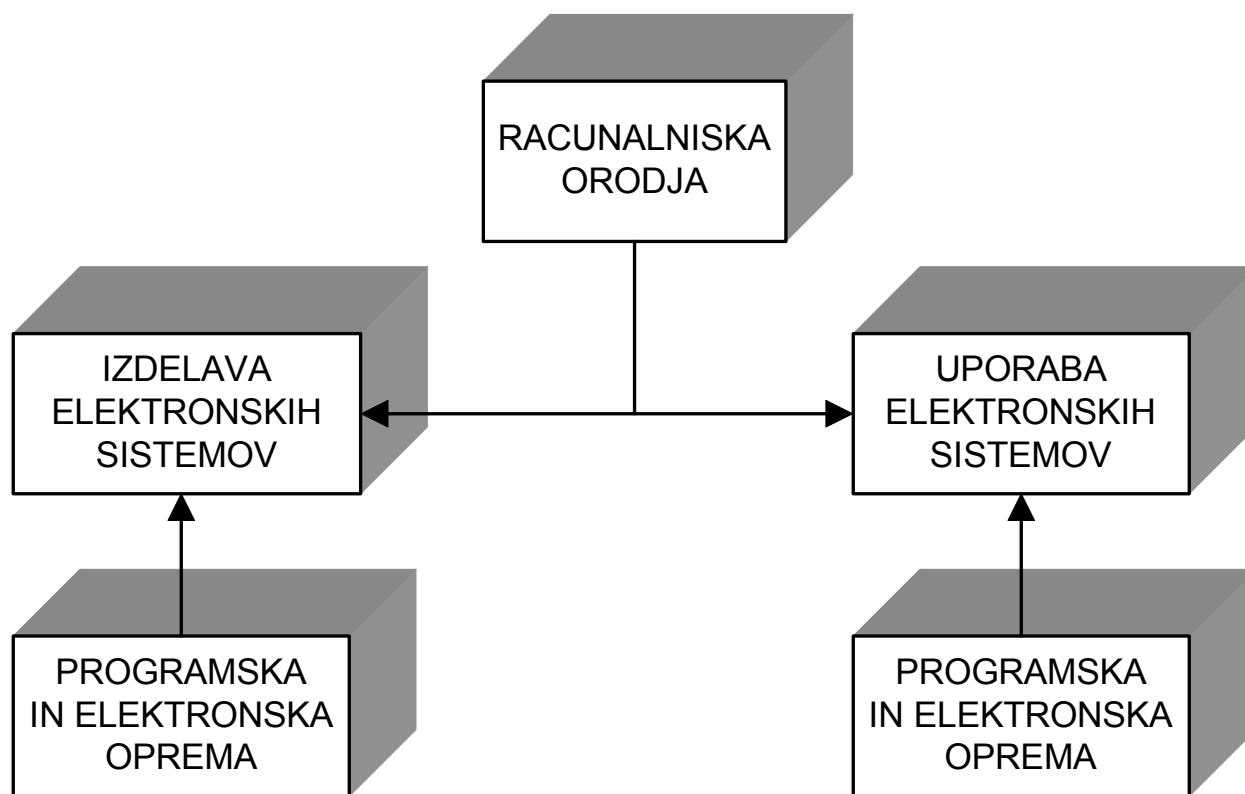
- **Predavanja:** ponedeljek, od 12 – 14 ure, PR06
- **Lab. vaje:** četrtek, od 12 – 16 ure
- Lab. vaje so obvezne, na koncu ocena iz vaj!
- Izpit bo 19., 30. 6. ter 2. 9. in 21. 9. 2009 ob 8. uri v lab. PE
- **Literatura:**
- Gorup Ž., *Računalniška orodja*, zapiski predavanj, Fakulteta za elektrotehniko, Ljubljana, 2008.
- Gorup Ž., *Računalniška orodja*, slikovno gradivo za predavanja, Fakulteta za elektrotehniko, Ljubljana, 2009.
- Gorup Ž., Jankovec M., Pirc M., *Računalniška orodja - Laboratorijske vaje*, Ljubljana, 2009.
- Gorup Ž., *LabVIEW*, Fakulteta za elektrotehniko, Ljubljana, 2007.
- Dodatna literatura v slovenskem jeziku:
- Peršič B., *Računalniška orodja*, Založba FE, Ljubljana, 1998.

Računalniška orodja - uvod

- Orodja so le tako dobra, kot so ljudje, ki jih uporabljajo.
- Vsako orodje je učinkovito le v rokah dobrega mojstra!
- Računalniška orodja sodobnega strokovnjaka elektronike.
- Namen: do cilja hitro, zanesljivo in donosno.
- Osredotočili se bomo na računalniška orodja, ki služijo za izdelavo ali pa uporabo elektronskih vezij in sistemov.
- Obstaja težko pregledna množica elektronske in programske opreme
- Sistematsko bomo razdelili računalniška orodja na način, ki je elektroniku logičen in sprejemljiv.
- Računalniška orodja za elektronike - dve skupini:
 - orodja za izdelavo elektronskih vezij in sistemov in
 - orodja za uporabo elektronskih vezij in sistemov.

Računalniška orodja - uvod

Osnovna razdelitev računalniških orodij

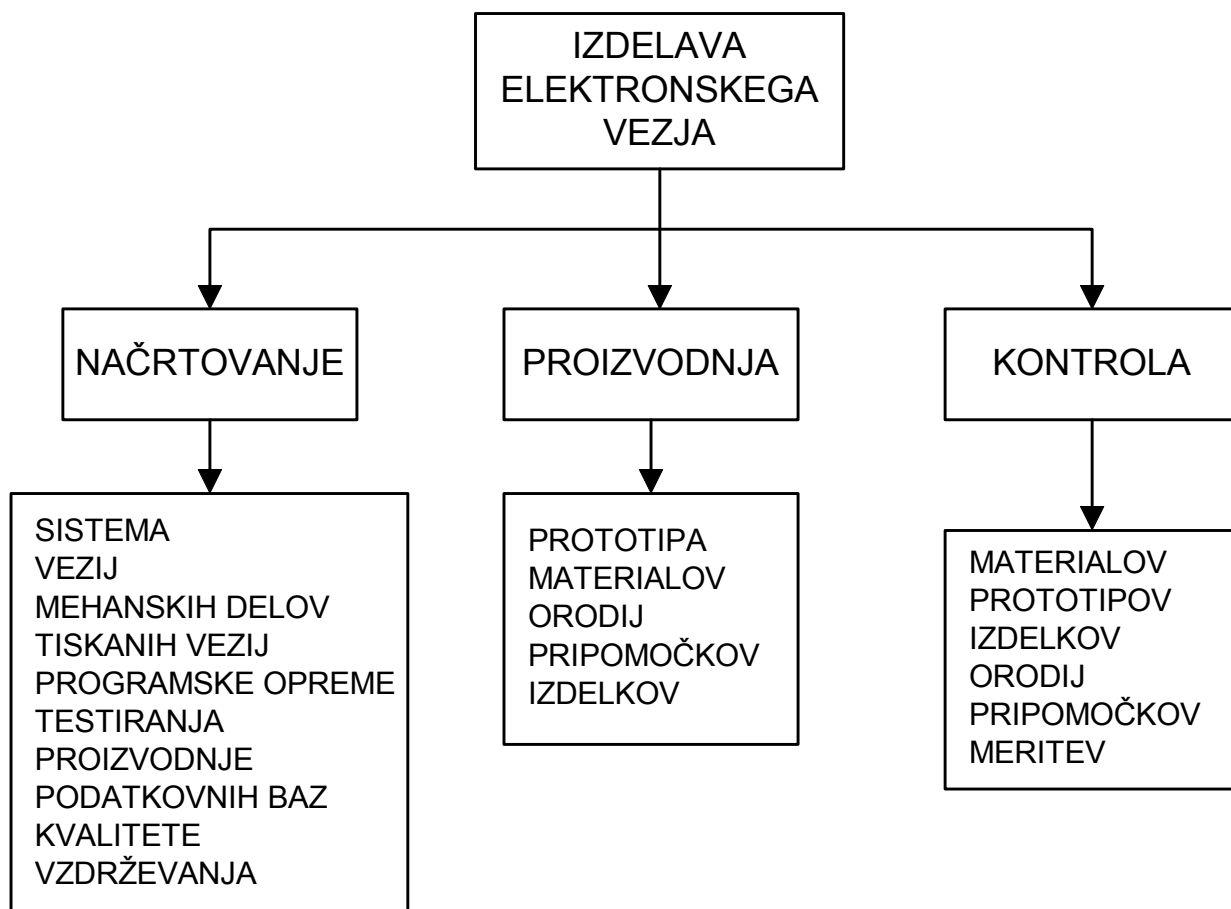


Računalniška orodja - uvod

- Elektronski sistemi so sestavljeni iz elektronskih vezij, komponent, mehanskih in drugih delov.
- Za elektronika so najpomembnejša elektronska vezja - največ pozornosti orodjem, ki služijo za gradnjo in uporabo elektronskih vezij.
- V procesu izdelave vezij so postopki, ki se izvajajo v specializiranih podjetjih. Te faze dela ne bomo podrobno obravnavali, ker so interdisciplinarnega značaja.
- Pozornost bomo posvetili le tistim fazam v izdelavi elektronskega vezja, ki zajemajo orodja za načrtovanje elektronskih vezij, tiskanih vezij in programske opreme.

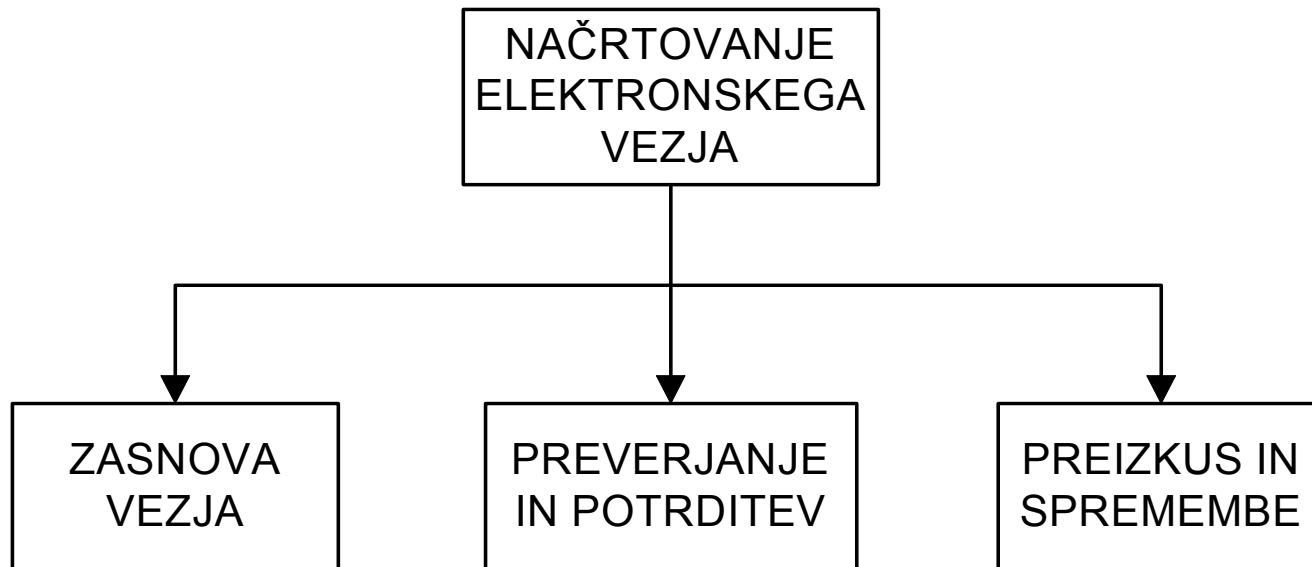
Računalniška orodja - uvod

Faze v izdelavi elektronskega vezja



Računalniška orodja - uvod

Faze v načrtovanju elektronskega vezja



Računalniška orodja - uvod

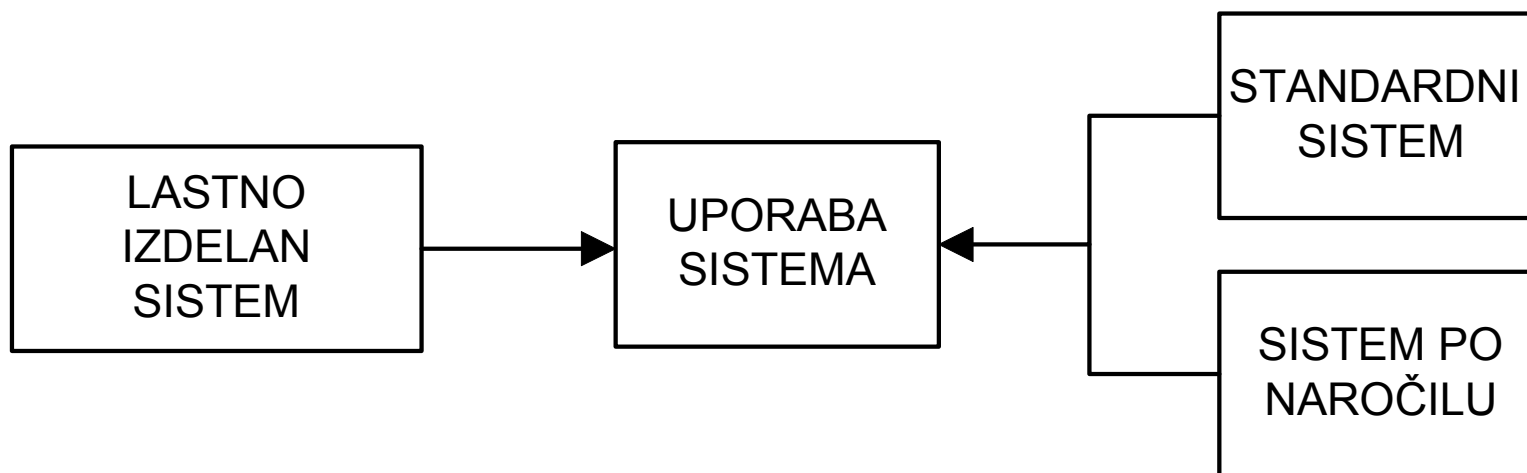
- V prvem delu bomo obravnavali virtualen sistem, ki bo temeljil na grafičnem programiranju.
- Ta pristop se v sodobni elektroniki veliko uporablja in ima napram klasičnemu pristopu nekatere bistvene prednosti:
 - programsko okolje je kompletno in omogoča upravljanje različne elektronske opreme,
 - v programskem okolju lahko zgradimo končne aplikacije,
 - za funkcionalno testiranje ni potrebno realno okolje,
 - za povezavo s fizikalnim svetom uporabimo standardno vezje, ki je izdelano in optimizirano,
 - elektronska in programska oprema lahko opravlja različne funkcije,
 - ima tudi bogato medprogramsko podporo,
 - elektronska oprema je mikroprocesorsko zasnovana in kompatibilna,
 - elektronska oprema je atestirana glede EMC,
 - celoten sistem ima vgrajena standardna vodila in komunikacijske protokole.

Računalniška orodja - uvod

- Elektronski sistemi so strukture, ki delujejo kot funkcionalne celote in opravljajo kompleksne naloge.
- Sestavljeni so iz elektronskih vezij ter električnih, mehanskih in drugih elementov.
- S stališča uporabnika lahko delimo elektronske sisteme na tiste, ki jih konstruiramo in izdelamo sami in tiste, ki so prisotni na tržišču in jih v vmesni ali pa dokončni obliki uporabimo v svojih aplikacijah.
- Pri slednjih razlikujemo še sisteme, ki so zgrajeni po naročilu (custom design) in standardne sisteme, ki so dovolj univerzalni, da jih lahko dobimo na tržišču brez posebnega dogovora s proizvajalcem.

Računalniška orodja - uvod

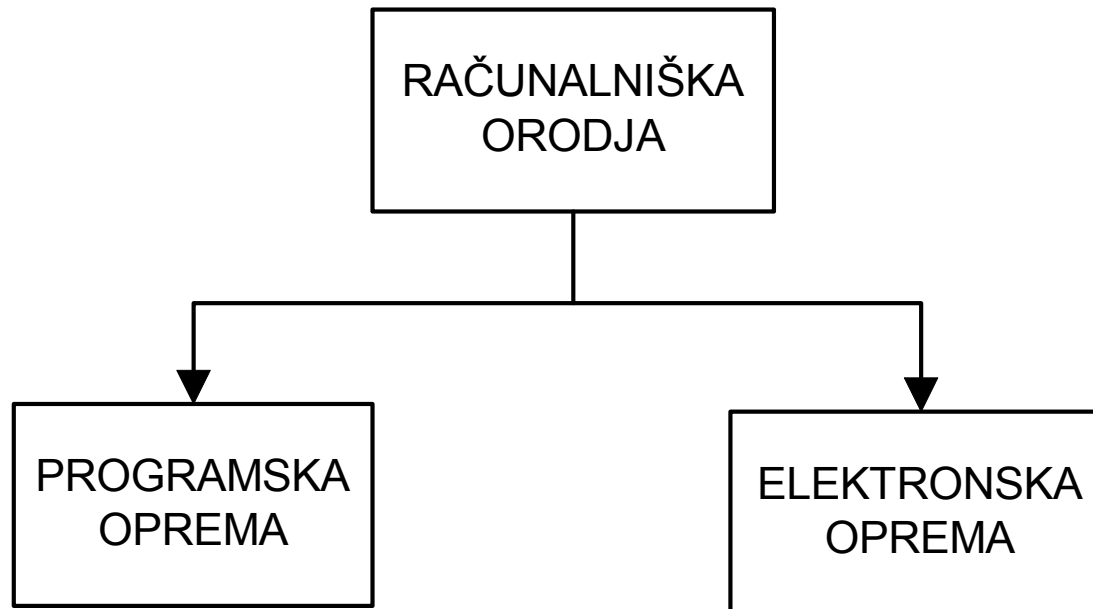
Uporaba elektronskega sistema glede na njegov izvor



Računalniška orodja - uvod

Računalniška orodja v elektroniki

- Računalniška orodja, ki jih pri svojem delu lahko uporabi elektronik, so lahko programska oprema ali pa specializirani sistemi.



Računalniška orodja - uvod

Računalniška orodja za uporabo elektronskih vezij

- računalniška orodja za uporabo standardnega elektronskega sistema: elektronska in programska oprema.
- Računalniško zasnovane meritve, krmiljenja in nadzori - uporabljamo standardne sisteme za zajemanje podatkov in generiranje signalov:
- krmiljenje zajemanja vhodnih signalov
- krmiljenje standardnih merilnih instrumentov
- kreiranje in uporaba virtualnih instrumentov
- analiza signalov in matematične funkcije
- grafični vmesnik za prikaz signalov in rezultatov
- komunikacije z merilnim okoljem (IEEE 488)
- podpora senzorjem in senzorskim sistemom (kalibracija, testiranje delovanja, samotestiranje)
- generiranje izhodnih signalov
- realizacija regulacij

Računalniška orodja - uvod

- Prikazali bomo splošne rešitve, primere pa bomo omejili na eno vrsto orodij, sicer bi bilo delo nepregledno in preveč raznoliko.
- Vaje bodo prirejene za okolje Windows, za katero obstaja tako programska, kot tudi elektronska oprema.
- Računalniška orodja se vsakodnevno spreminjajo in posodablajo.
- Ocenjujmo primernost orodja po nekaterih osnovnih kriterijih:
 - primernost za naše delovno okolje in stroške v zvezi z uvajanjem
 - primernost za ljudi, ki ga bodo uporabljali in njihov nivo znanja
 - čas, ki je potreben, da se uporabnik nauči uporabe orodja
 - čas, ko orodje povrne stroške lastne investicije
 - kompatibilnost z drugo programsko in elektronsko opremo, ki jo uporabljamo
 - cena vzdrževanja in nadgradnje obstoječe opreme (elektronske in programske).

Računalniška orodja - uvod

- Vsak izdelek je namenjen opravljanju dela.
- Sodobni izdelki morajo prinašati dobiček.
- Čas, ki je potreben od ideje do kupca, se krajša.
- Izdelki, ki tega ne upoštevajo, kljub dobri zasnovi propadejo.
- Upoštevati moramo kriterije razvoja izdelkov, ki v razvitem svetu veljajo danes in se pripraviti na načine dela, ki bodo veljali jutri.
- Odločitev za nakup orodja ni lahka in enostavna. Ne odločajmo se le na podlagi reklamnih opisov.
- Sestavimo svoj testni primer, ki zajema za naše delo kritične elemente in postopke in se dogovorimo za izdelavo preizkusnega vezja.
- Orodja ne rešujejo vseh problemov, morda pa jih vsaj opazimo.
- Pogovorimo se z ljudmi, ki orodje že dalj časa uporabljajo.

Računalniška orodja

Uvod v LabVIEW

1. predavanje

Laboratorij PRAKTIKUM ZA ELEKTRONIKO
PE soba B-304

Uvod v LabVIEW

Grafično programiranje

- Grafično programiranje je razmeroma mlado in manj uporabljano.
- Prisotna je tradicija in miselnost, ki vlada med uporabniki programske opreme.
- Spremenjen odnos do grafičnega programiranja so prispevali sodobni operacijski sistemi, ki delujejo prek zmogljivih grafičnih vmesnikov.
- Grafično programiranje je metoda, ki za zapis programa uporablja grafične objekte namesto tekstovnega opisa.
- Te grafične objekte v skladu s programskimi pravili nato ustrezno medsebojno spojimo.
- Klasični, tekstovni programski jeziki zahtevajo za zapis programa natančno določeno sintakso teksta, ki sestavlja program.
- Vsaka napaka onemogoči delovanje programa.
- Grafični program je sestavljen iz grafičnih simbolov ali ikon in povezav.
- Grafično programiranje uporabljajo različne stroke: od strojnikov, gradbenikov in arhitektov prek elektrotehnikov do oblikovalcev in umetnikov.
- Za področje elektronike so pomembni grafični programi, ki nudijo uporabniku možnosti za načrtovanje vezij in sistemov, njihovo modeliranje in simulacijo ter upravljanje vezij in sistemov v raziskovalnih in proizvodnih procesih.
- Precej sočasno so na tem področju delale predvsem tri firme: Hewlett-Packard s programom VEE, Burr-Brown (Intelligent Instrumentation), ki je razvil Visual Designer in National Instruments s programom LabVIEW.

Uvod v LabVIEW

- Na Fakulteti za elektrotehniko Univerze v Ljubljani smo se odločili za grafično programiranje v okolju LabVIEW.
- V tekstovnih programskih jezikih je nadzor nad računalniško vodeno elektronsko opremo dokaj zapleten.
- Uporabnika ozadje delovanja programa ne zanima. Pomembno je, da lahko tudi nestrokovnjak za programiranje gradi lastne aplikacije ali pa spreminja obstoječe.
- Grafični program, ki ga potrebujemo v elektroniki za računalniško podprto delo ni en sam.
- Najprej so grafične simbole dobili elektronski elementi, ki so osnova za risanje električnih načrtov. Sledili so jim programi za računalniško načrtovanje tiskanih vezij. Potem so se pojavili programi za upravljanje elektronskih sistemov z grafičnim vmesnikom.
- Program LabVIEW je sestavljen zelo kompleksno:
 - sistem za zajemanje podatkov,
 - analizo in sintezo signalov ter
 - avtomatizacijo procesov.
- Grafični program ima vgrajeno lastno dokumentiranje. Slika grafičnega programa je podobna diagramu poteka, s katerim v tekstovnih programskih jezikih opisujemo izvajanje programa. Grafični program ima enotne simbole.
- Grafični jezik ima vgrajene gonilnike za vhodno-izhodne enote.
- Tudi uporabnikov grafičnega programiranja je vsak dan več:

Uvod v LabVIEW

Sestava grafičnega programa

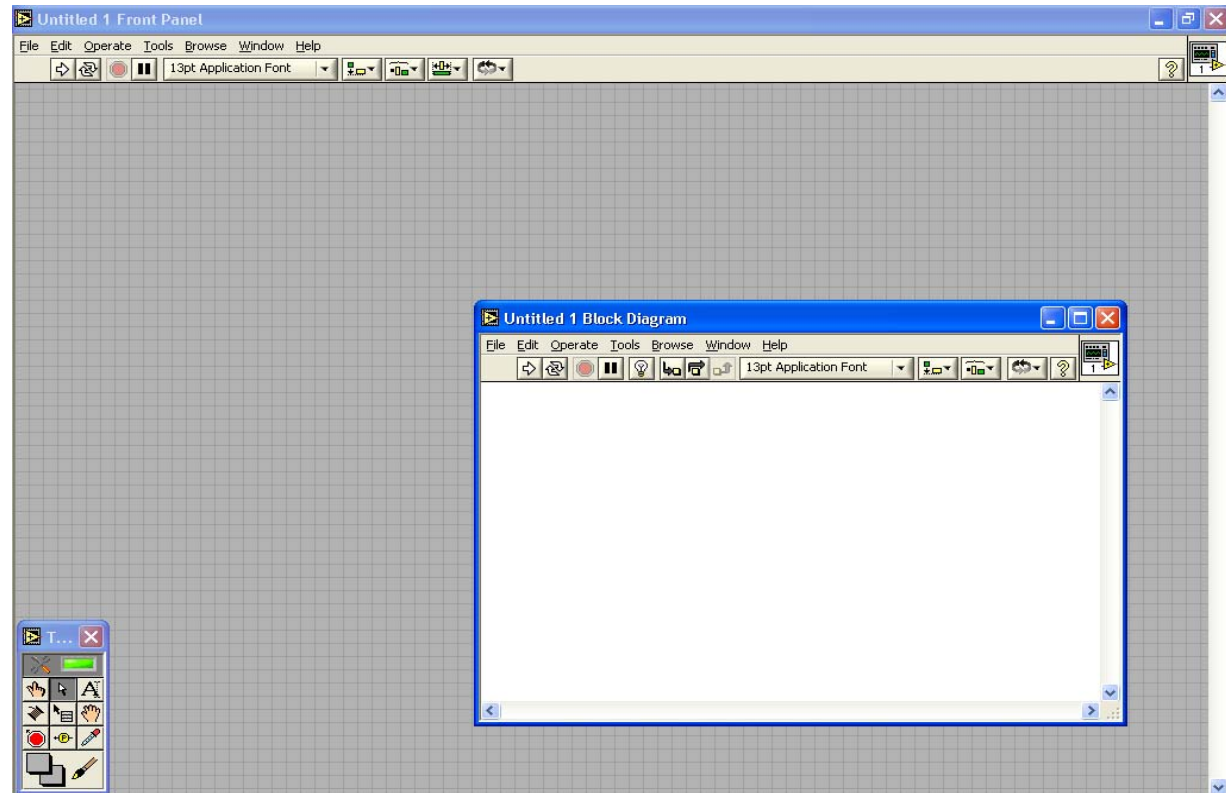
- Program deluje v dveh oknih: okno **čelne plošče** in okno **blok diagrama**.
- Vsak element čelne plošče ima pripadajoč element na blok diagramu LabVIEW, obratno pa seveda ne velja.
- Začnemo s postavitvijo osnovnih elementov čelne plošče.

Virtualni instrument

- Virtualni ali navidezni instrument (VI) je program, ki uporabniku nadomešča fizikalno okolje. Virtualni instrument sestavljajo tri enote:
 - čelna plošča,
 - blok diagram in
 - ikona.
- Čelna plošča in blok diagram delujeta vsak v svojem programskem oknu, ikona pa je za obe enoti skupna in enaka (ista).
- Čelna plošča je grafični vmesnik, ki omogoča uporabniku posredovanje pri izvajanju programa in prikaz izhodnih veličin.
- Blok diagram je sininim za kodo, ki je običajen uporabnik med izvajanjem programa ne opazuje.

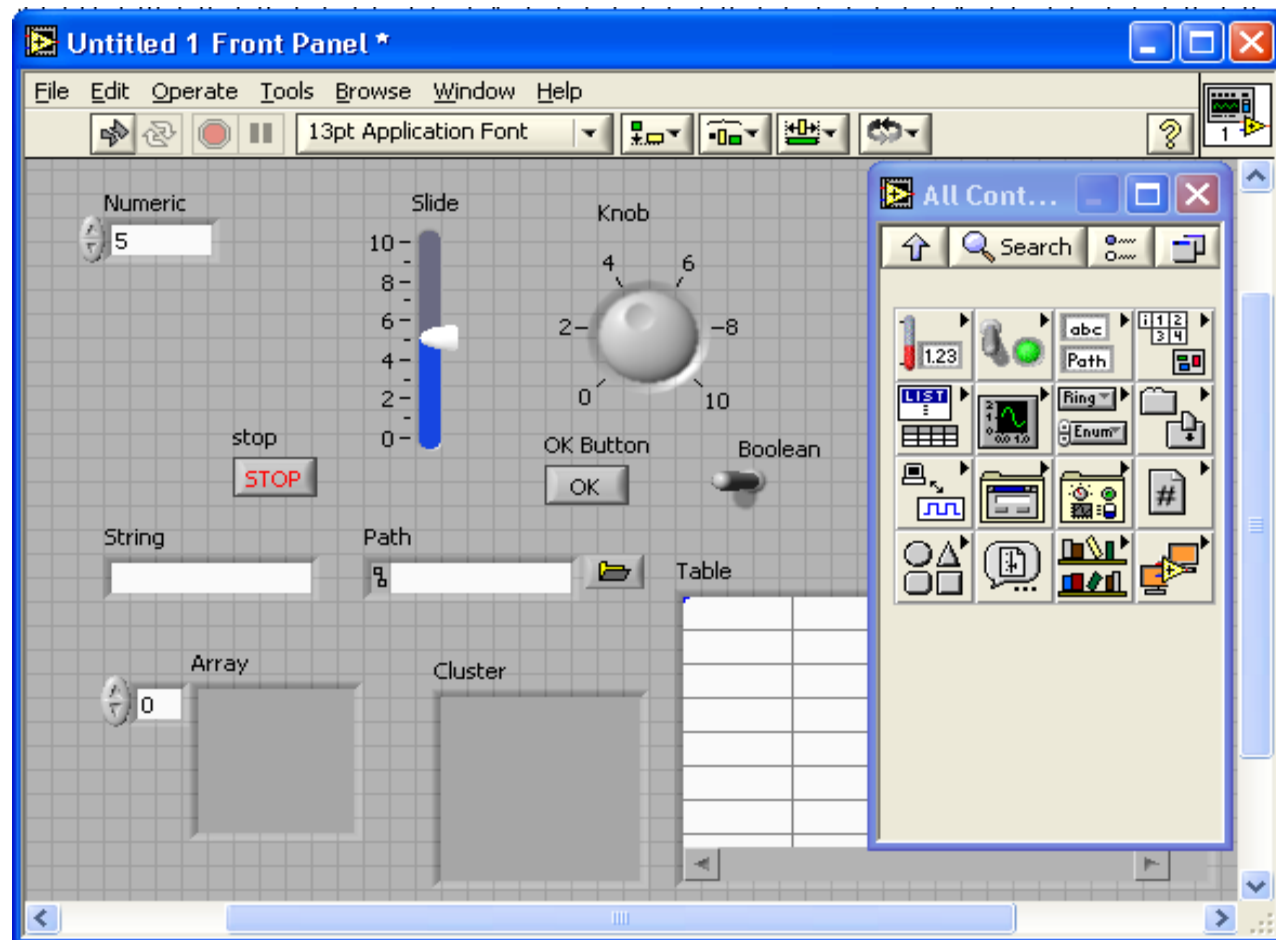
Virtualni instrument

- čelna plošča
- blok diagram
- ikona



Čelna plošča

- Je vhodno-izhodni vmesnik za uporabnika
- Elementi:
 - kontrole



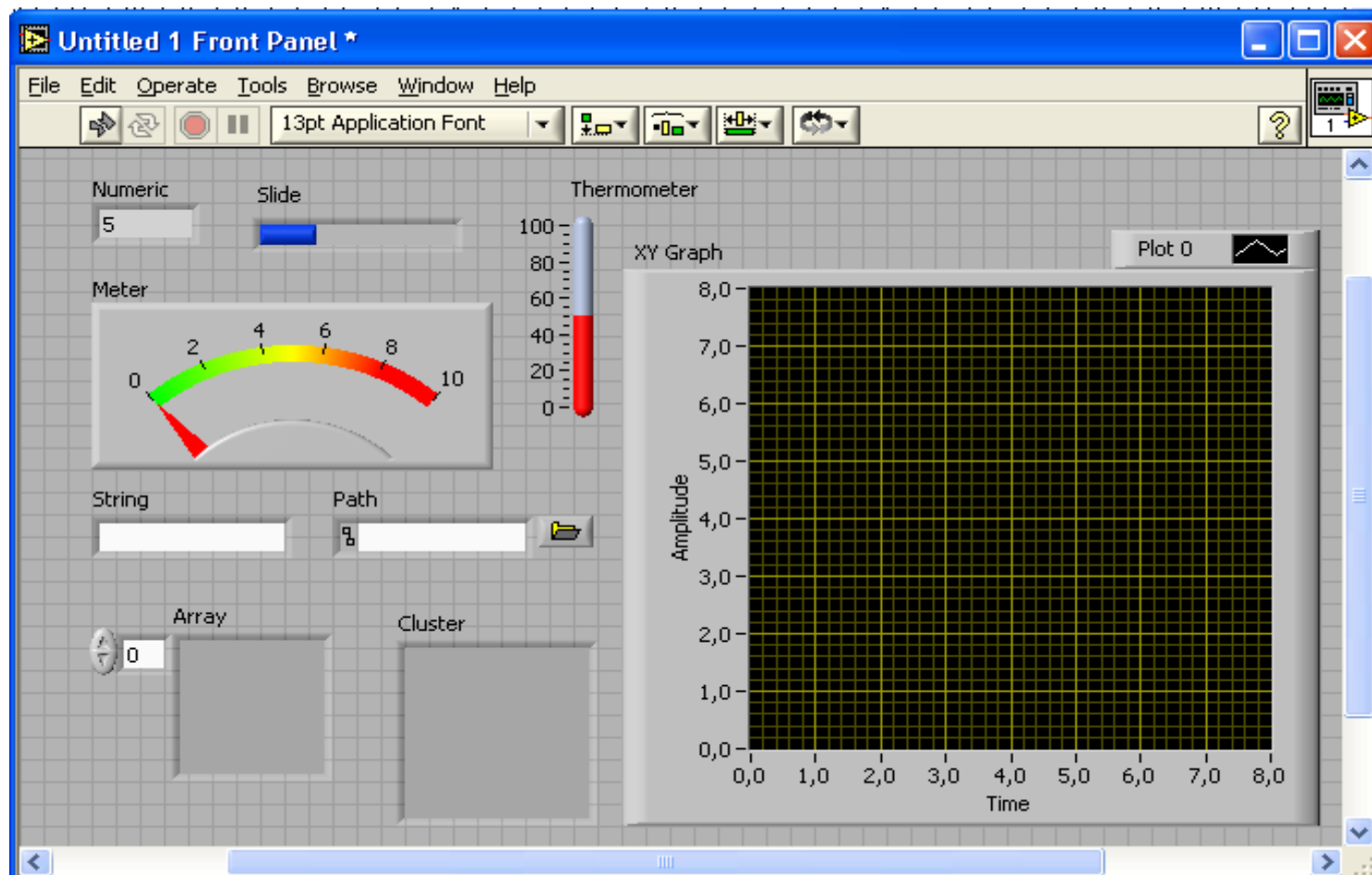
Čelna plošča

Kontrole so gradniki, ki predstavljajo vhode. Vhodi na čelni plošči so elementi, ki omogočajo kakršenkoli vnos podatkov, parametrov ali ukazov, na katere se program odziva:

- numerična (Numeric)
- logična (Boolean)
- niz (String & Path)
- polje ali grozd (Array & Cluster)
- seznam ali tabela (List & Table)
- izbira možnosti (Ring & Enumerator)
- shrambe (Containers, ActiveX, iNET)
- referenčne kode za I/O procese
- pogovorne kontrole (Dialog Controls)
- referenčne kode (Reference Numbers)
- dekoracije (Decorations)
- uporabniške kontrole (User Controls).

Čelna plošča

- Je vhodno-izhodni vmesnik za uporabnika
- Elementi:
 - indikatorji



Čelna plošča

Indikatorji so elementi čelne plošče, ki služijo za prikaz rezultatov, odzivov ali stanj.

Programsko gledano so izhodi, ki uporabniku posredujejo zahtevano informacijo.

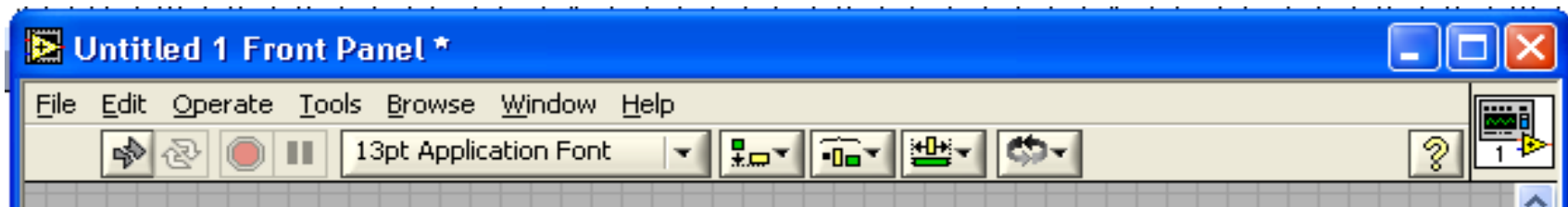
Med indikatorji lahko izbiramo širok nabor elementov:

- numerični (Numeric Indicator)
- logični (Boolean)
- grafični (Graph)
- Elementi na čelni plošči niso neposredno povezani, zato jih lahko poljubno razvrstimo na delovno površino okna.
- Ugodna za postavitev elementov čelne plošče je mreža (Grid), ki jo lahko uporabimo za približno uravnavanje objektov glede na njihovo okolico.

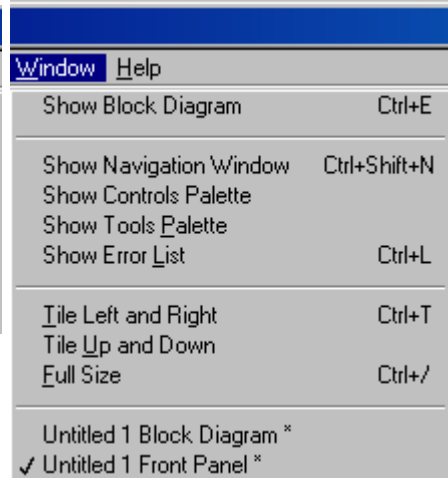
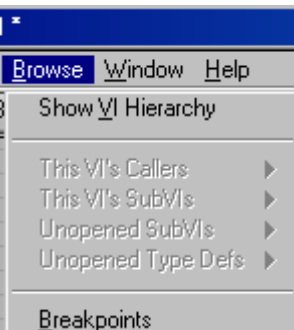
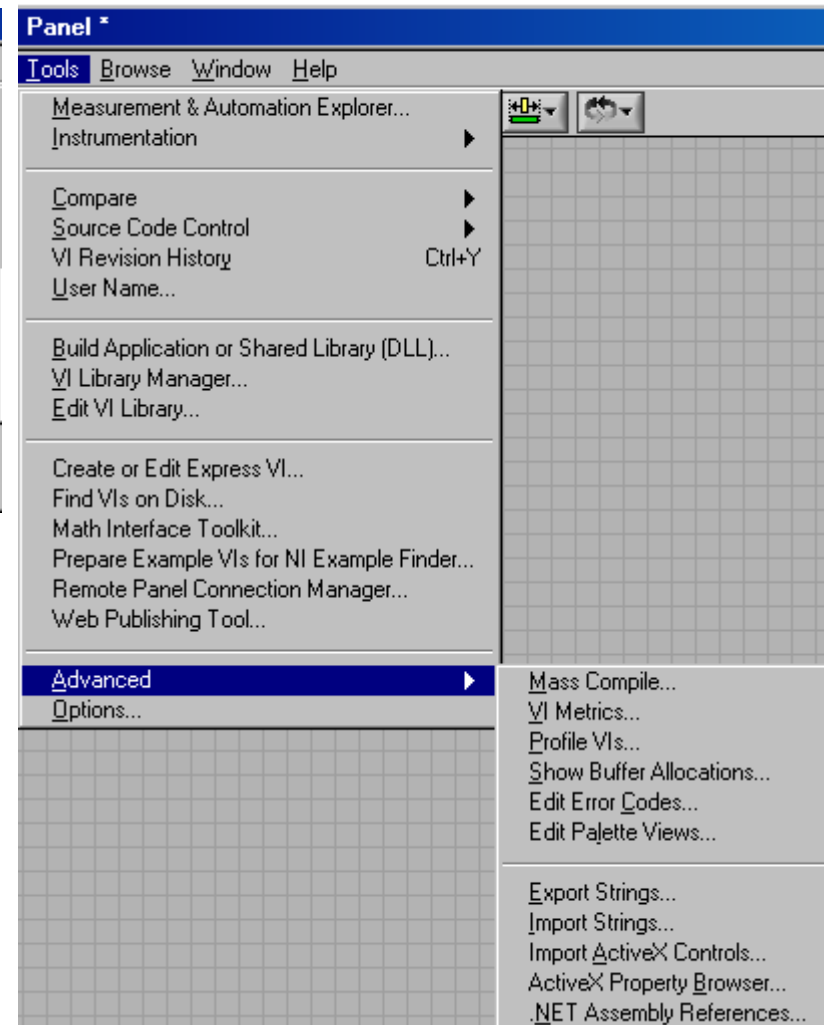
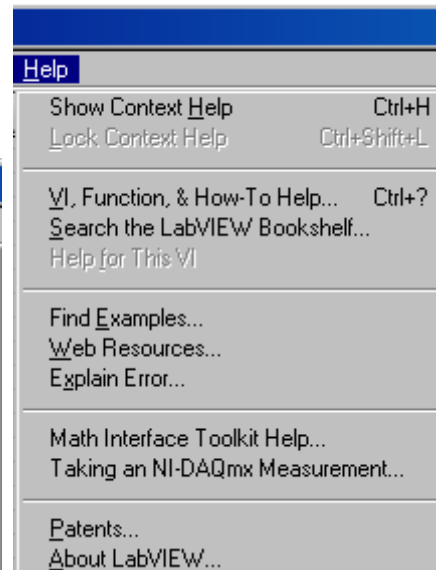
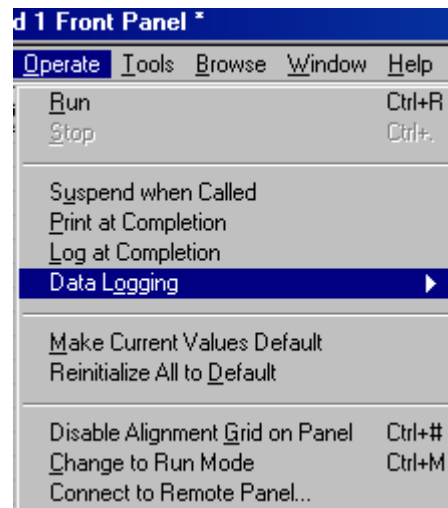
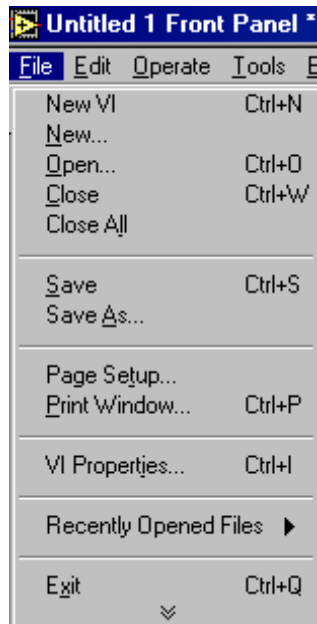
Menuji

Potezni menuji: datoteka, urejanje, delovanje, orodja, pregledovanje, okno in pomoč.

Statusna orodna vrstica: zagon programa, kontinuirano delovanje, status izvajanja programa, pavza, izbira znakov, poravnavanje, razmaki, premik elementa (nad-pod), pomoč.



Menuji



Orodja

Paleta orodij (za obe okni):

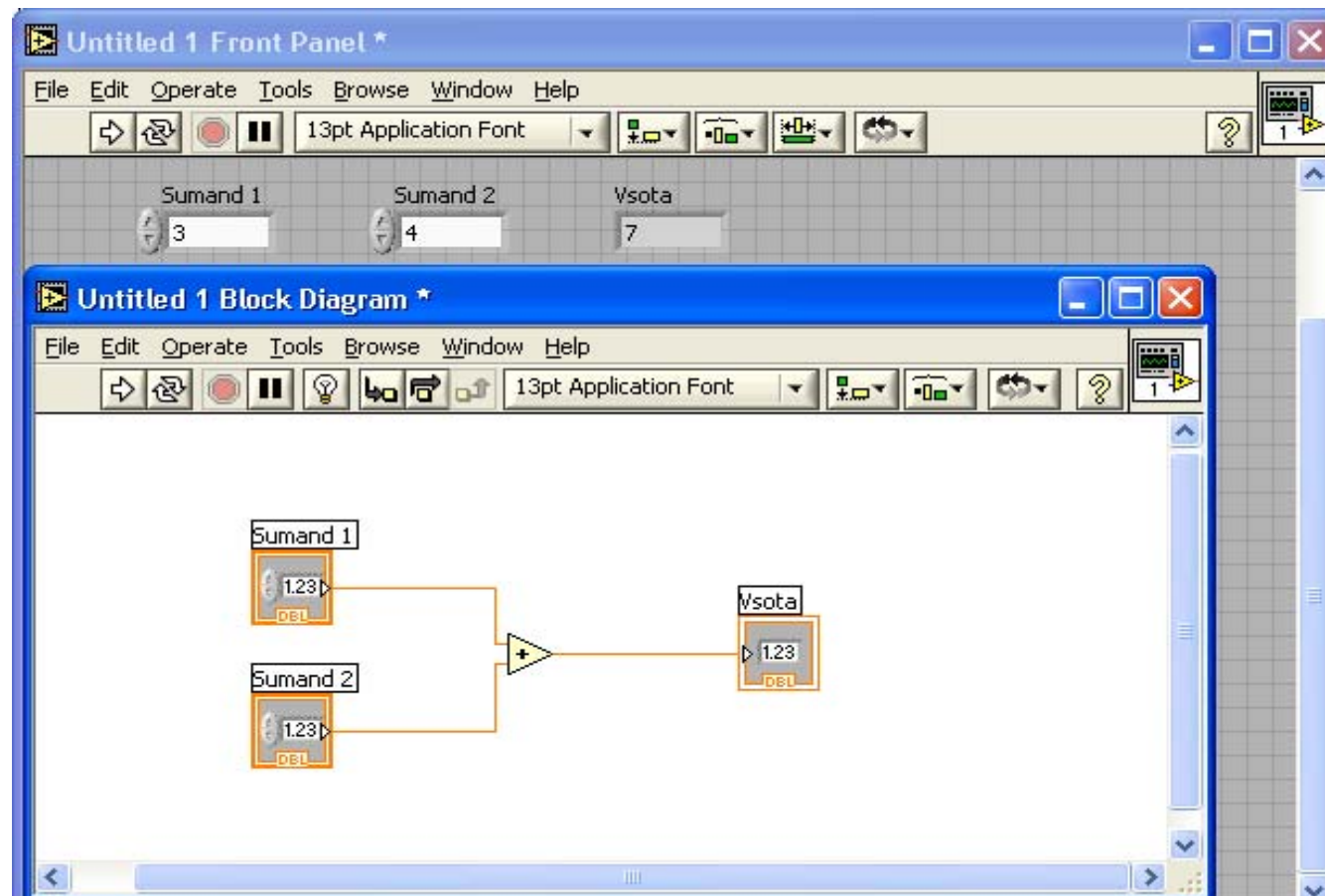
- Elementi:
 - uporabniški kazalec (vnos vrednosti)
 - spreminjanje objekta
 - urejanje teksta
 - povezovanje
 - bližnjica do objekta
 - pomik okna
 - zaustavljanje programa
 - podatki testne točke
 - vzorčenje barv
 - barvanje objekta



Blok diagram

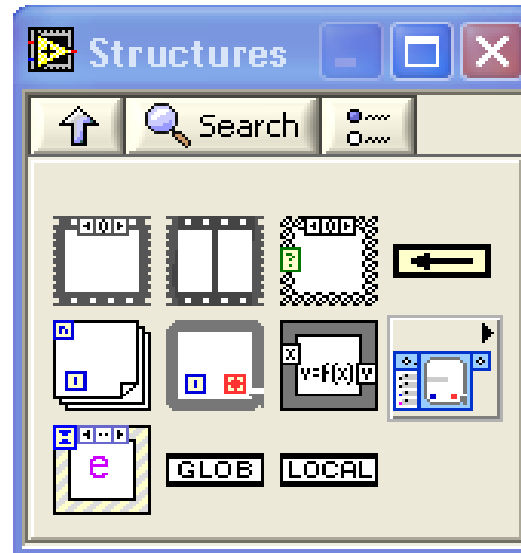
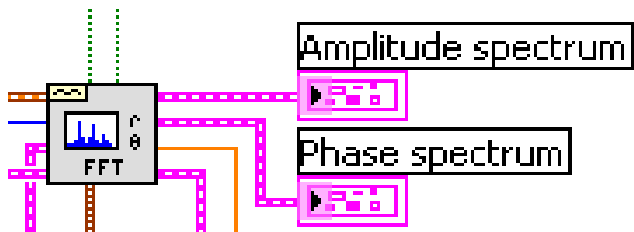
Določa potek izvajanja programa.

- Elementi:
 - vozlišča (Nodes)
 - priključki (Terminals)
 - vezi (Wires)



Vozlišča

- funkcije (Functions)
- podinstrumentna vozlišča (subVI nodes)
- strukture (Structures)



Funkcije

Lastnosti:

- Funkcije so osnovni operativni elementi programa LabVIEW.
- Ikone funkcij se nahajajo v paleti **Functions**.
- Funkcije nimajo čelne plošče, niti blok diagrama, imajo pa konektor.
- Funkcij ne moremo odpreti niti jih urejati.

Funkcije

Programski VI in Funkcije so osnovni gradniki VI.

- **Merilni I/O VI in Funkcije** se uporabljajo za povezavo tradicionalnih NI-DAQ (Legacy), NI-DAQmx in drugih elementov za zajemanje podatkov.
- **Instrumentni I/O VI in Funkcije** se uporabljajo za povezavo z GPIB, serijskimi, modularnimi, PXI in drugimi vrstami instrumentov.
- **Matematični VI** izvajajo različne matematične analize. Lahko jih povežemo z realnimi meritvami za procesiranje podatkov.
- **VI za procesiranje signalov** so namenjeni generiranju signalov, digitalnemu filtriranju, okenskim funkcijam in spektralni analizi.
- **VI in Funkcije za podatkovne komunikacije** uporabljamo za prenos podatkov med različnimi aplikacijami.
- **Povezovalni VI in Funkcije** se uporabljajo za delo z .NET objekti, ActiveX-omogočenimi aplikacijami, vhodnimi elementi, naslavljanji registrov, pri nadzoru virov in "Windows registry keys".
- **Ekspres VI in Funkcije** se uporabljajo za gradnjo splošnih merilnih opravil.

Podinstrumentna vozlišča

Glavne lastnosti:

- To so elementi, ki izvajajo podinstrumentne funkcije.
- V blok diagramu se pojavljajo kot ikone podinstrumentov.
- Lahko jih uporabimo kjerkoli pri gradnji VI.
- V istem VI se lahko pojavljajo večkrat.
- Imajo lastno ikono in konektor.

Strukture

Grafične predstavitve zank tekstovnih programskih jezikov:

- FOR
- WHILE
- CASE

Strukture za nadzor poteka izvajanja programa:

- SEQUENCE
- EVENT

Priključki

- Povezava med čelno ploščo in blok diagramom
- Vsak element čelne plošče ima v blok diagramu grafičen element
- kontrole – debelejši okvir + puščica ven
- indikatorji – tanjši okvir + puščica noter



Tipi podatkov

- Določajo, katere vhode in izhode lahko povežemo med seboj.
- Tip podatka je označen z barvo priključka.
- Barve priključkov morajo biti skladne z barvo vezi.

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 

Vezi

- So podatkovne poti med priključki.
- Povezujemo lahko le utripajoče priključke.

Scalars



1D Array



2D Array

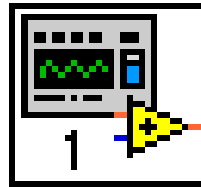


NOTE: Array wires are thicker than scalar wires, and array terminals have [] around base type

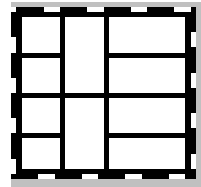
Ikona

- Ikona predstavlja VI.
- Dva načina prikaza:

– simbol



– konektor: vsaki kontroli in indikatorju pripada eno polje konektorja.

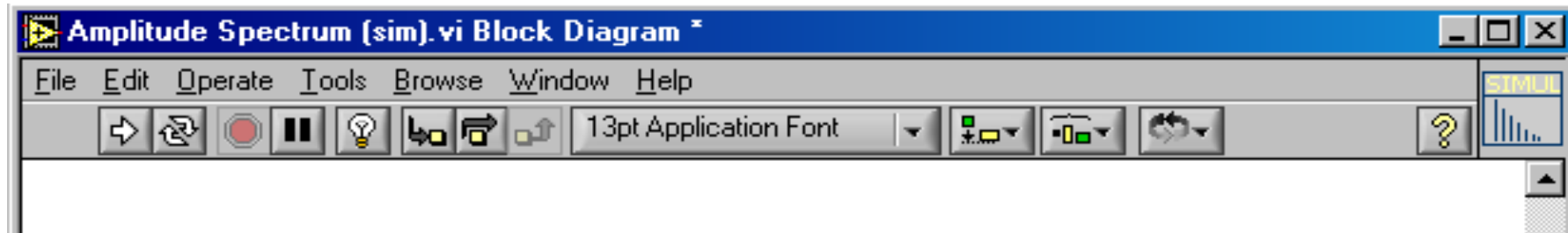


Orodna vrstica

Orodna vrstica blok diagrama

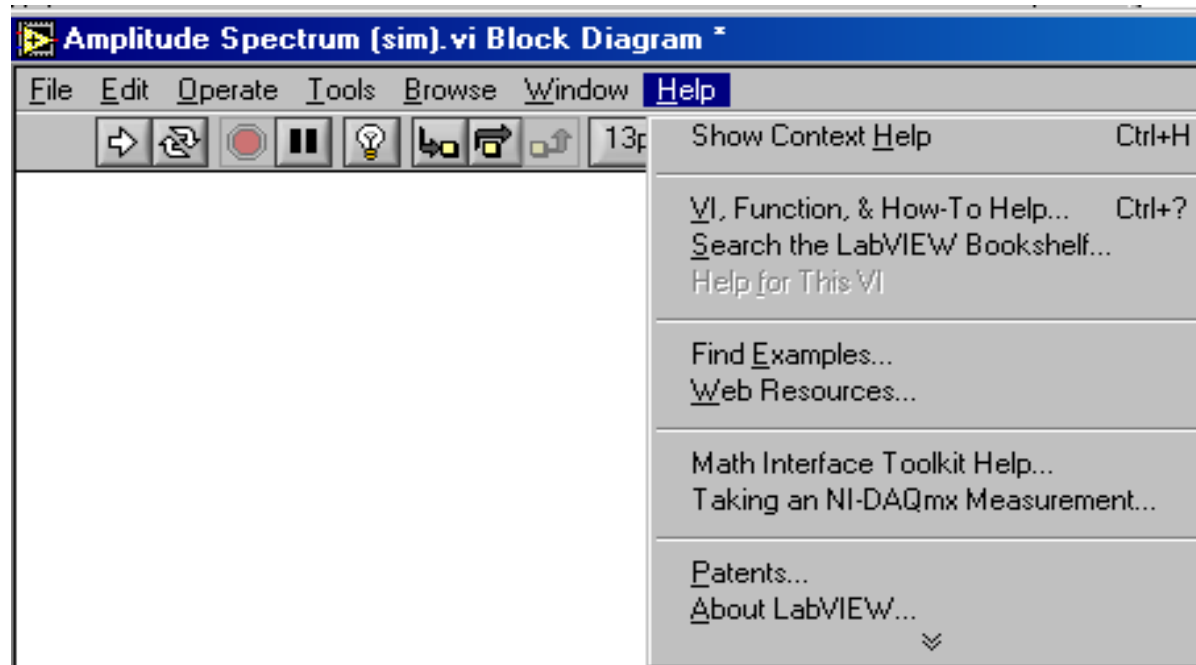
Poleg orodij čelne plošče so še orodja za iskanje in odpravljanje napak:

- prikaz izvajanja programa,
- začetek izvajanja po korakih,
- preskok zanke, bloka ali podinstrumenta,
- izhod iz zanke, bloka ali podinstrumenta.



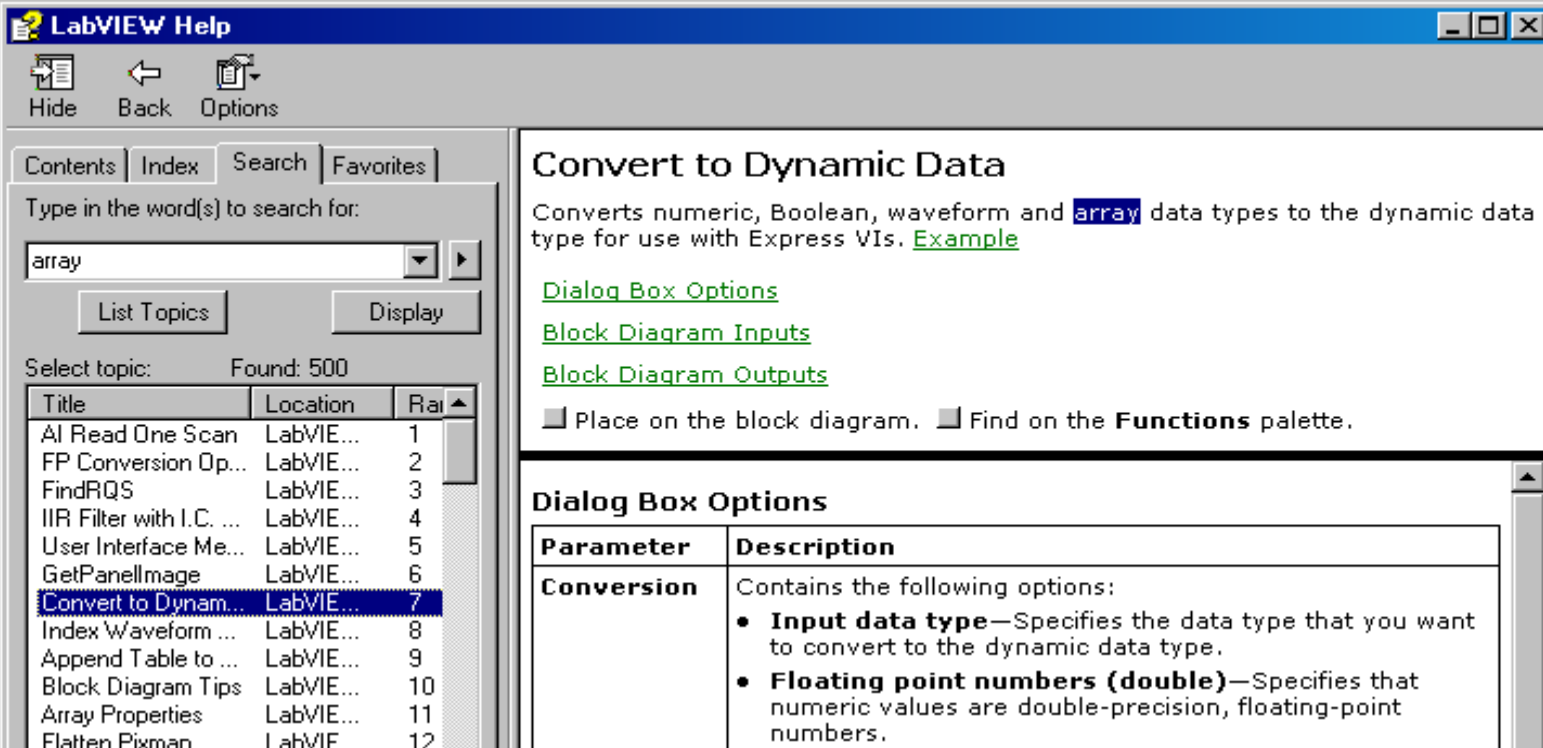
Pomoč

- pomoč s povezavo (Context Help)
- pomoč LabVIEW
- pomoč s primeri (Example Finder)



Pomoč LabVIEW

- iskalni del
- opisni del
- podatkovni del



The screenshot shows the LabVIEW Help application window. The title bar reads "LabVIEW Help". Below the title bar are icons for "Hide", "Back", and "Options". The main content area is divided into two panes. The left pane contains a search interface with tabs for "Contents", "Index", "Search", and "Favorites". The "Search" tab is active, showing a search box with the text "array" and a "Display" button. Below the search box is a table of search results with columns "Title", "Location", and "Rank". The "Convert to Dynam..." entry is highlighted. The right pane displays the "Convert to Dynamic Data" topic, which includes a description, links to "Dialog Box Options", "Block Diagram Inputs", and "Block Diagram Outputs", and a checkbox for "Place on the block diagram." Below this is a "Dialog Box Options" section with a table of parameters.

Convert to Dynamic Data

Converts numeric, Boolean, waveform and **array** data types to the dynamic data type for use with Express VIs. [Example](#)

[Dialog Box Options](#)

[Block Diagram Inputs](#)

[Block Diagram Outputs](#)

Place on the block diagram. Find on the **Functions** palette.

Dialog Box Options

Parameter	Description
Conversion	Contains the following options: <ul style="list-style-type: none">• Input data type—Specifies the data type that you want to convert to the dynamic data type.• Floating point numbers (double)—Specifies that numeric values are double-precision, floating-point numbers.

Polimorfni VI

Polimorfni VI je tisti, ki lahko sprejme različne tipe podatkov (tudi večfunkcijski).

Različni tipi podatkov se nanašajo na en vhod ali izhod.

Polimorfni VI je skupina VI s skupnim konektorjem. Vsak izmed VI skupine je del polimorfne VI.

Polimorfni VI

Gradnja polimorfnega VI

- Zgradimo dva ali več VI s soležnimi konektorji.
- Izberemo **File»New** da se pokaže [New](#) pogovorno okno.
- Na seznamu **Create New**, izberemo **VI»Polymorphic VI**.
- Izberemo **OK** tipko, da se pokaže okno [Polymorphic VI](#).
- Pritisnemo **Add** tipko, da se pokaže pogovorno okno datotek.
- Izberemo VI, ki bo sestavljal polimorfni VI in pritisnemo tipko **Open**. VI se pojavi na vrhu **Instance VIs** seznama. To je tudi privzeti VI in konektor.
- Postopek ponovimo za vse sestavne dele polimorfnega VI. Če nek VI ne ustreza kriterijem, postane ikona pretrgana.
- V urejevalniku lahko postavimo zaporedje izvajanja.
- Pritisnemo **Edit Icon** tipko za kreiranje ikone [create an icon](#) polimorfnega VI.
- Opis delovanja [Create a description](#) lahko vključimo z izbiro **File»VI Properties** in nato **Documentation** v **Category** poteznem menuju.
- Izberemo **File»Save** za shranjevanje VI.
- Izberemo **File»Close** za zaprtje okna polimorfnega VI

Iskanje in odpravljanje napak

Nekateri prijemi za zaznavanje napak:

- Prekinitev povezav
- Detekcija napake na povezavah “error in” – “error out”
- Lista napak (Error List)
- Preverjanje povezave (osvetlitev)
- Iskanje nepovezanega VI (hierarhija)
- Osvetljeno izvajanje
- Koračno izvajanje
- Izpis podatkov (Probe)
- Točke zaustavljanja programa (Breakpoints)
- Prisotnost skritih subVI (hierarhija)

Iskanje in odpravljanje napak

Programski pripomočki za odkrivanje napak:

- sonda ali testna konica (Probe)
- osvetlitev izvajanja programa (Highlighting)
- koračno izvajanje programa (Single-Step)
- zaustavitev programa (Pause)
- vstavitev točk zaustavljanja (Breakpoints)

Iskanje in odpravljanje napak



Koračno izvajanje programa (Single-Step)

- **Step Into** – Odpre vozlišče in čaka. Po ponovnem pritisku gumba se izvede prvi korak.
- **Step Over** - Izvede vozlišče in čaka na naslednjem vozlišču.
- **Step Out** - Konča izvajanje tekočega vozlišča in čaka. Ko je končano izvajanje VI, postane gumb siv.

Virtualni podinstrument - subVI

Samostojna programska enota - subVI je VI, ki je vključen v drug VI.

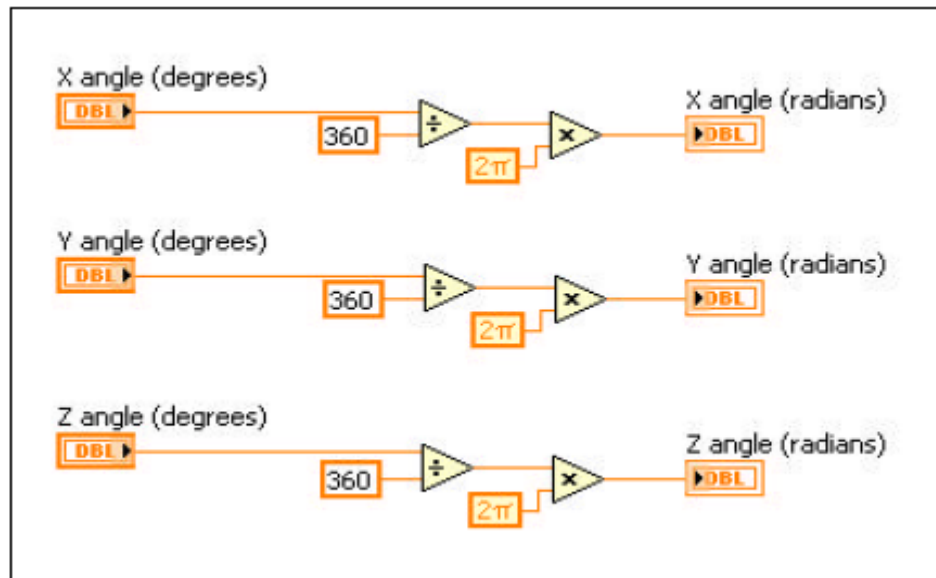
subVI omogoča:

- ponovitev programske kode,
- predstavitev kompleksne strukture z enostavnim simbolom,
- povezavo kode z uporabniškim vmesnikom,
- kreiranje kompleksnih VI, ki so razumljivi in preprosti za vzdrževanje.

Virtualni podinstrument - subVI

Primer:

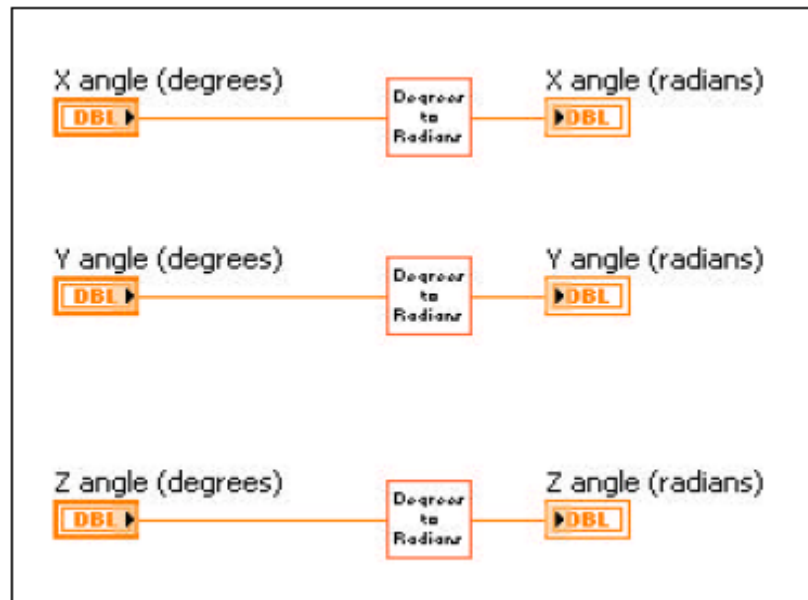
- stopinje v radiane



Virtualni podinstrument - subVI

Primer:

- izvedba s subVI “Degrees to Radians”



Virtualni podinstrument - subVI

Urejanje konektorja podinstrumenta

