

Računalniška orodja

# Uvod v LabVIEW

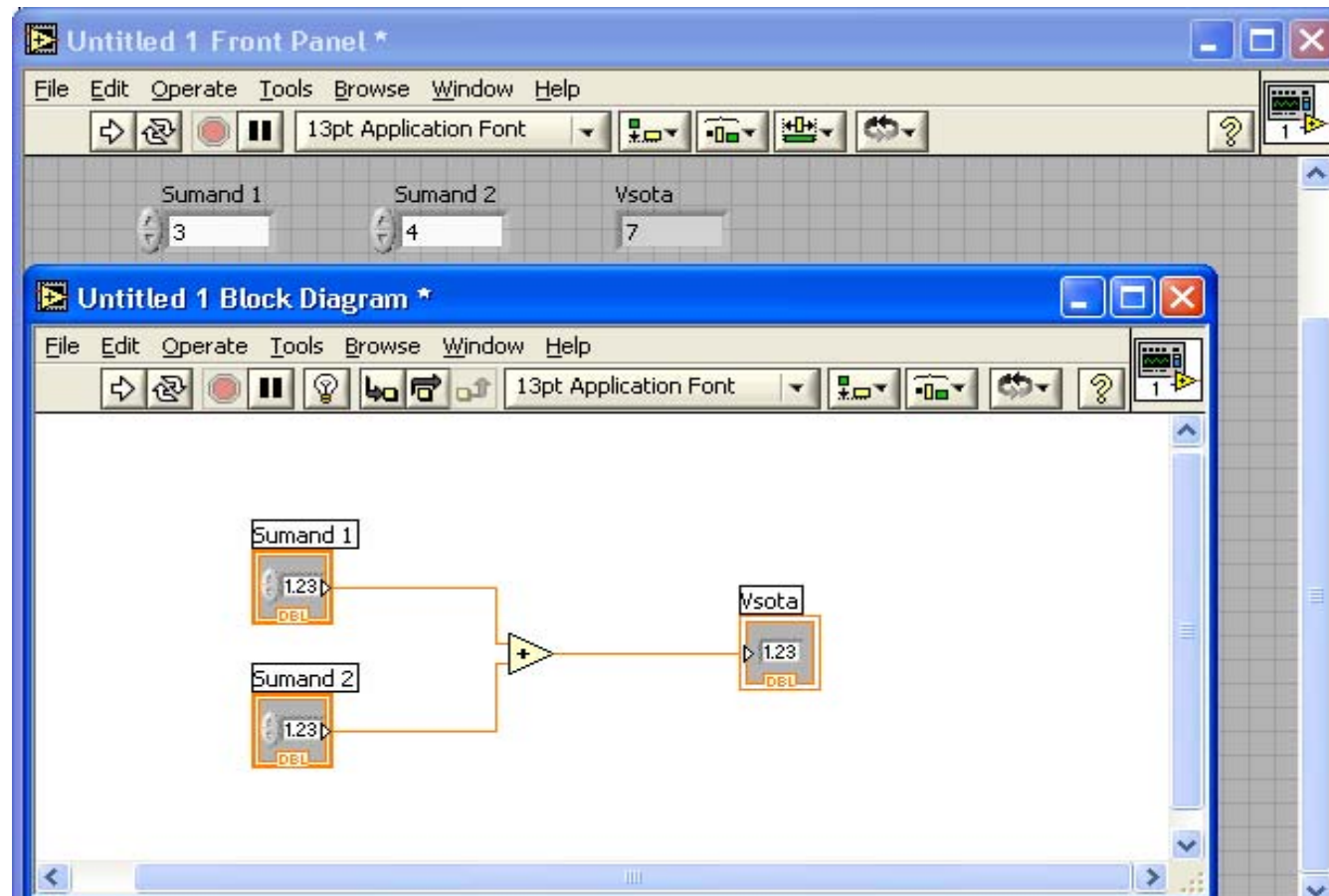
2. predavanje

Laboratorij PRAKTIKUM ZA ELEKTRONIKO  
PE soba B-304

# Blok diagram

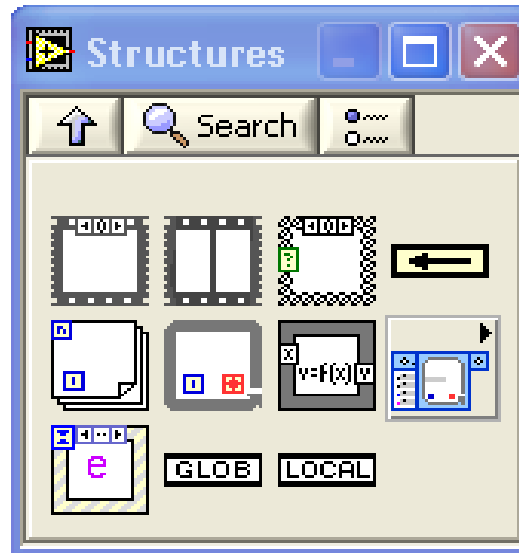
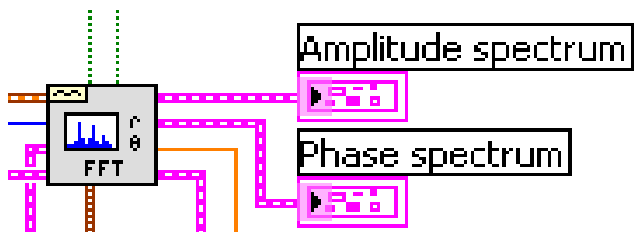
Določa potek izvajanja programa.

- 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 in analitičnih 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 

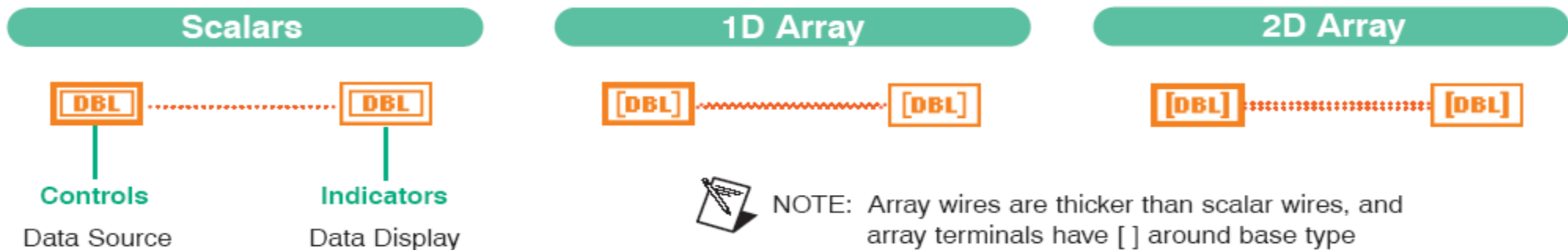
Waveform 

Polymorphic 

I/O Name Control 

# Vezi

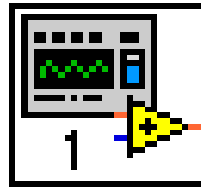
- So podatkovne poti med priključki.
- Povezujemo lahko le utripajoče priključke.



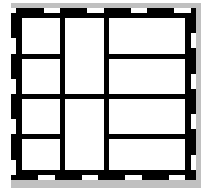
# Ikona

- Ikona predstavlja VI.
- Dva načina prikaza:

– simbol



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

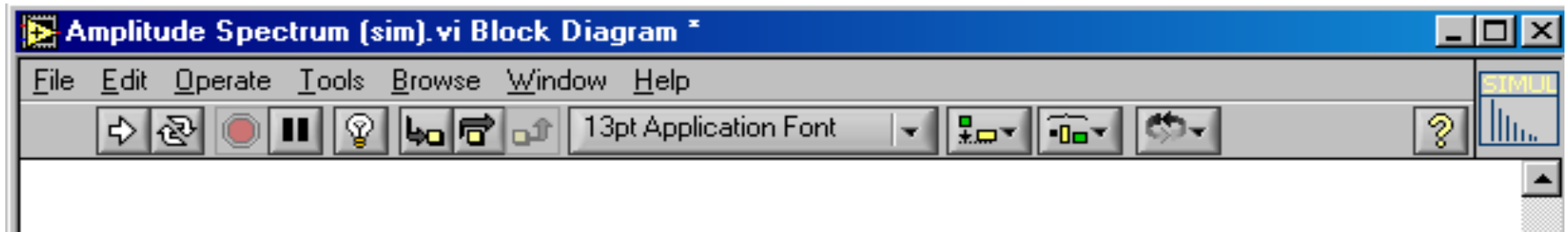


# Orodna vrstica

## Orodna vrstica blok diagrama

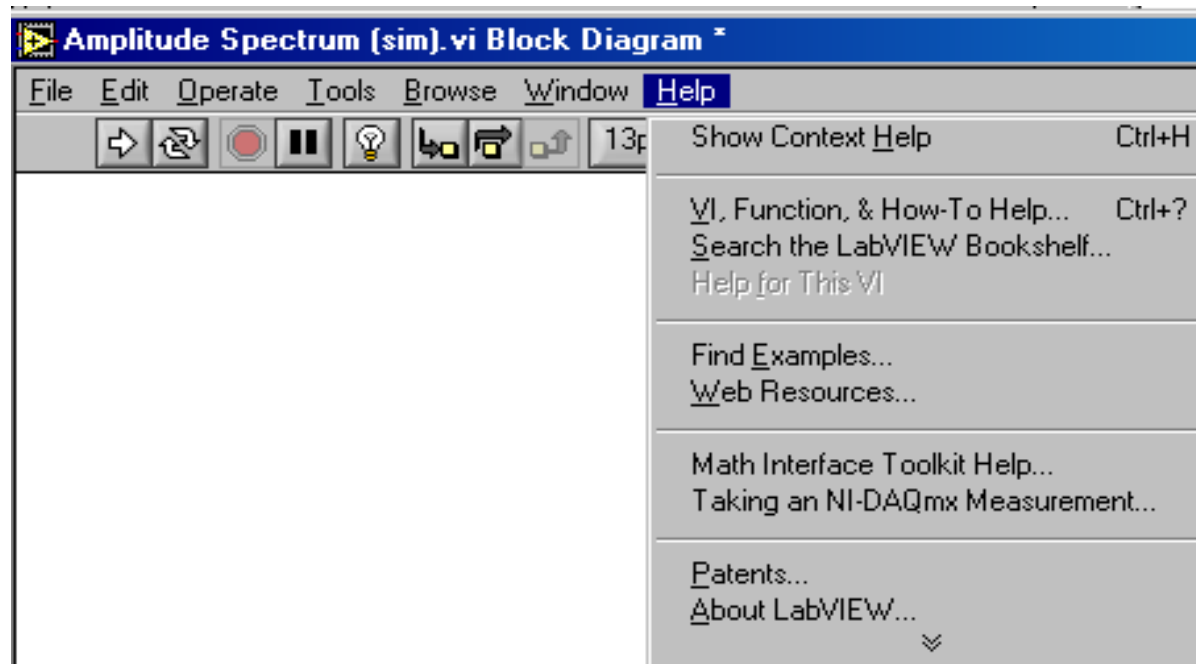
Poleg orodij čelne plošče so še orodja za iskanje in odpravljanje napak:

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



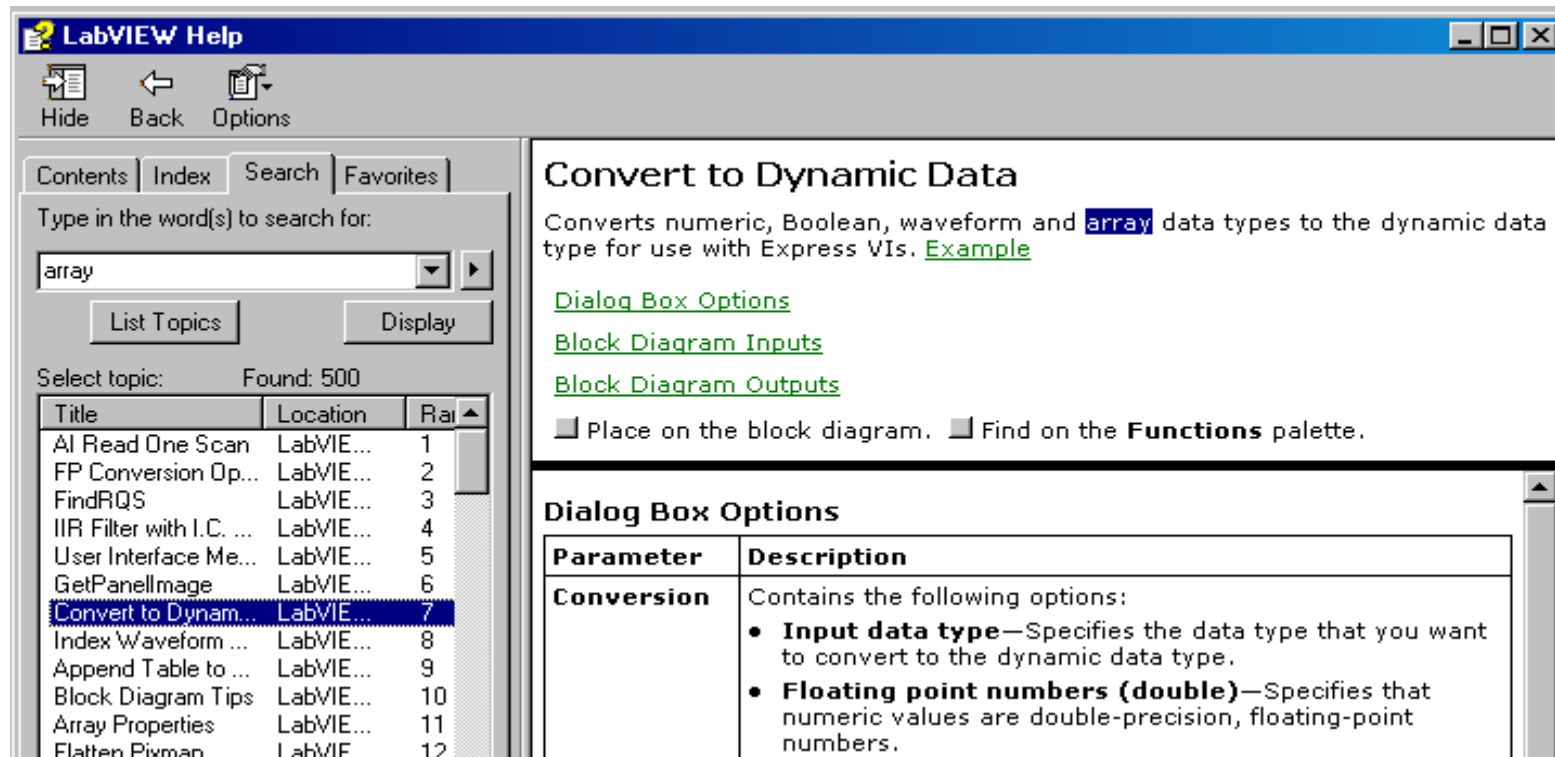
# Pomoč

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



# Pomoč LabVIEW

- iskalni del
- opisni del
- podatkovni del



# 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 polimorfnega 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

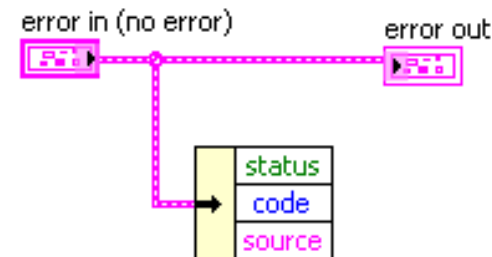
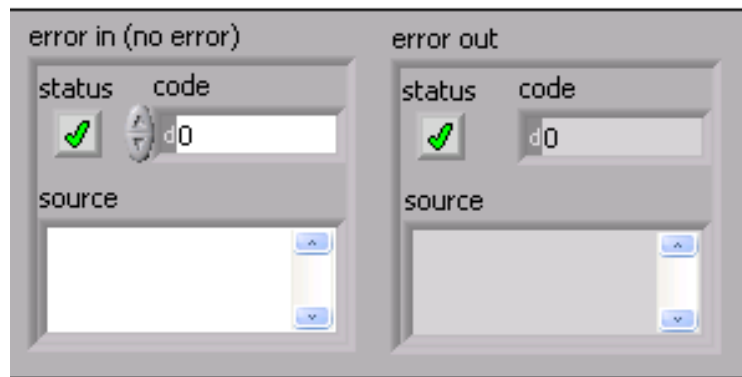


# Upravljanje napak

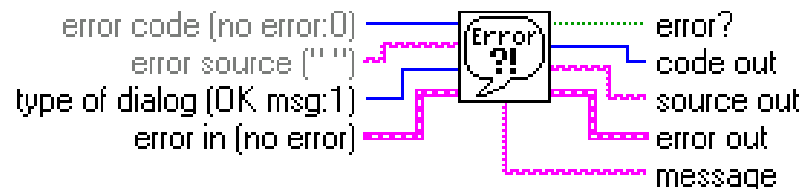
- LabVIEW po privzetku avtomatično upravlja napake delujočega VI s prekinitvijo izvajanja, osvetlitvijo subVI ali funkcije, kjer je prišlo do napake, in prikazom pogovornega okna napak.
- Snopi (grozdi) napak so strukture, ki omogočajo spremljanje napak posameznih VI.
- Error in, error out vsebujejo sledeče:
  - **status** je Boolova spremenljivka, ki da vrednost TRUE, če pride do napake.
  - **koda** je 32-bitno predznačeno število, ki z vrednostjo opredeli napako. Poljubna vrednost v kombinaciji s **statusom** FALSE pomeni opozorilo!
  - **izvor** je niz, ki pove, kje je prišlo do napake.

# Upravljanje napak

- Snopi napak (Error Clusters)

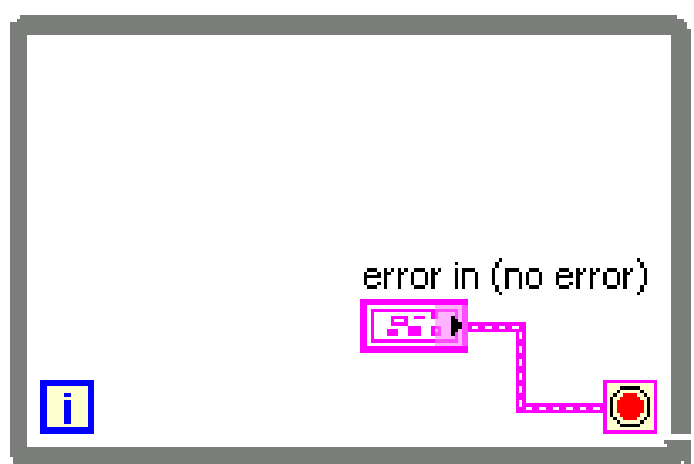


- VI za upravljanje napak (Error Handler VI)



# Upravljanje napak

- Uporaba WHILE zank pri upravljanju napak (While Loops for Error Handling)



- Ko pride do napake, se WHILE zanka ustavi.
- Če vežemo snop napak na pogojni priključek, se spremeni pogoj “Stop if True” in “Continue if True” v “Stop on Error” in “Continue while Error”.

# 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.

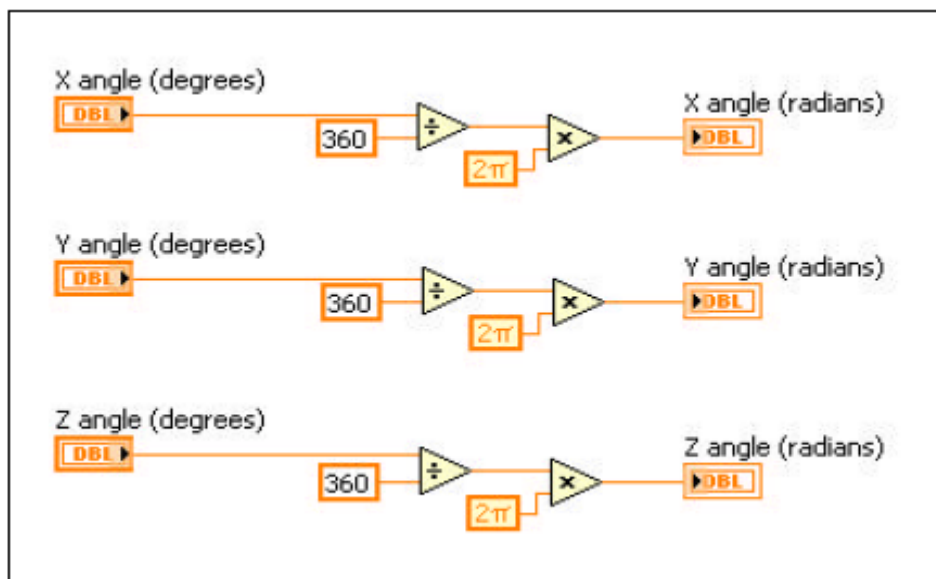
subVI omogoča:

- 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

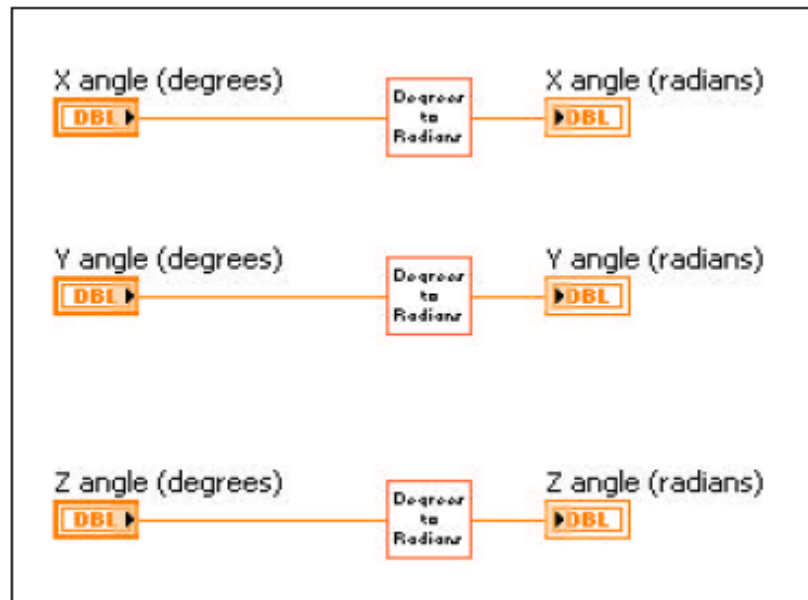




# Virtualni podinstrument - subVI

Primer:

- izvedba s subVI “Degrees to Radians”



# Virtualni podinstrument - subVI

## Urejanje konektorja podinstrumenta

