LabVIEW

Dr Marko Dimitrijević

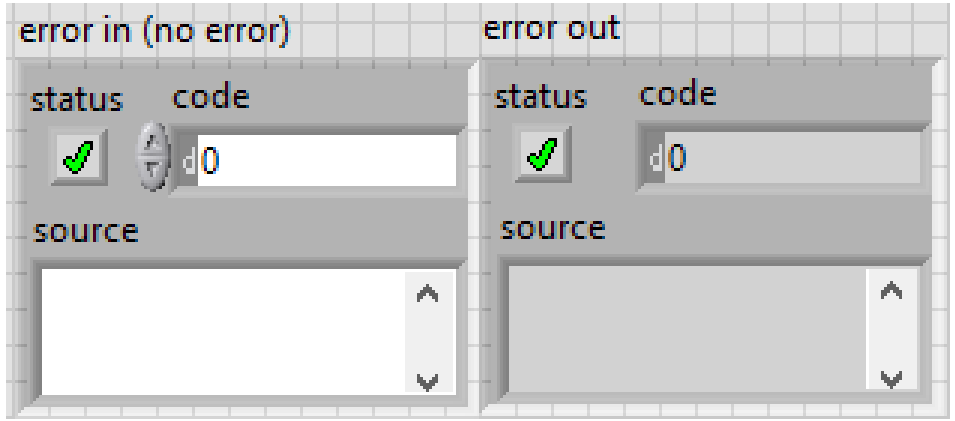
Napredne tehnike programiranja

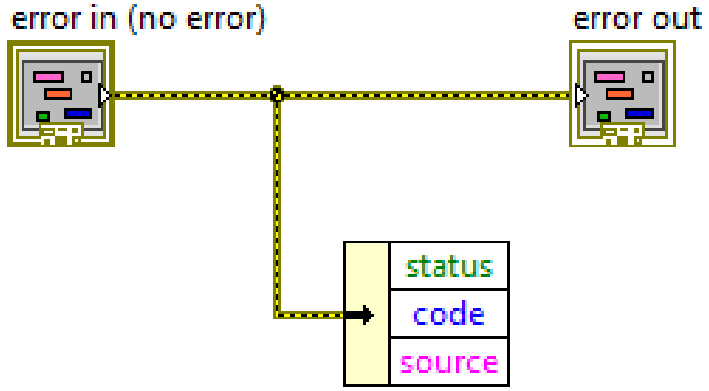
Napredne tehnike programiranja

- Kontrola grešaka.
- Korisnički interfejs i događaji

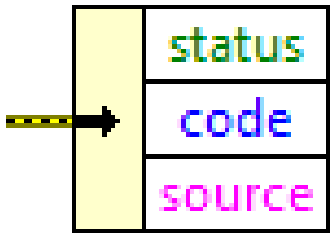
Klaster za kontrolu grešaka

- error in i error out klasteri se mogu koristiti za kontrolu grešaka u virtuelnom instrumentu.
- Klasteri za kontrolu grešaka se nalaze u **Controls»Array, Matrix & Cluster** paleti





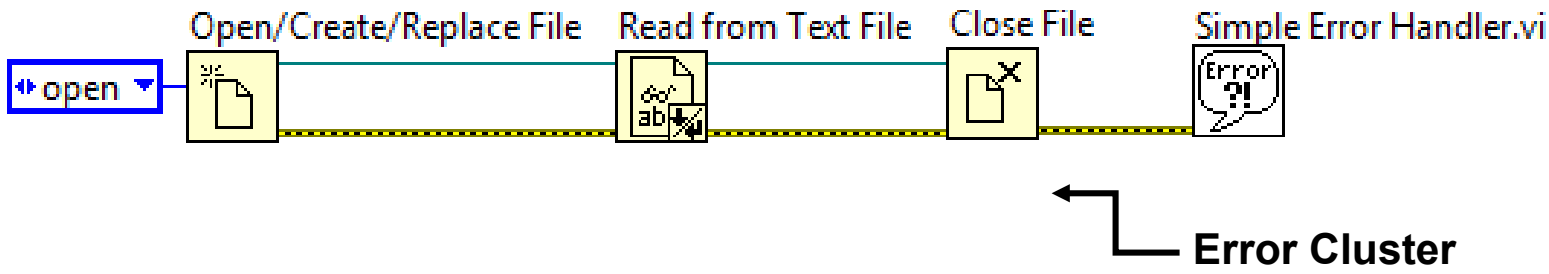
Komponente klastera za kontrolu greške



- **Status** je logička promenljiva jednaka vrednosti TRUE ukoliko se dogodila greška. Veliki broj virtuelnih instrumenata, funkcija i struktura koje imaju logički ulaz, prepoznaju ovaj parametar.
- **Code** je označeni 32-bitni celi broj koji numerički identifikuje grešku. Ukoliko je različit od nule, a pri tome Status ima vrednost FALSE, ukazuje na upozorenje (warning).
- **Source** je string koji ukazuje na to gde je greška nastala.

Kontrola grešaka pomoću klastera

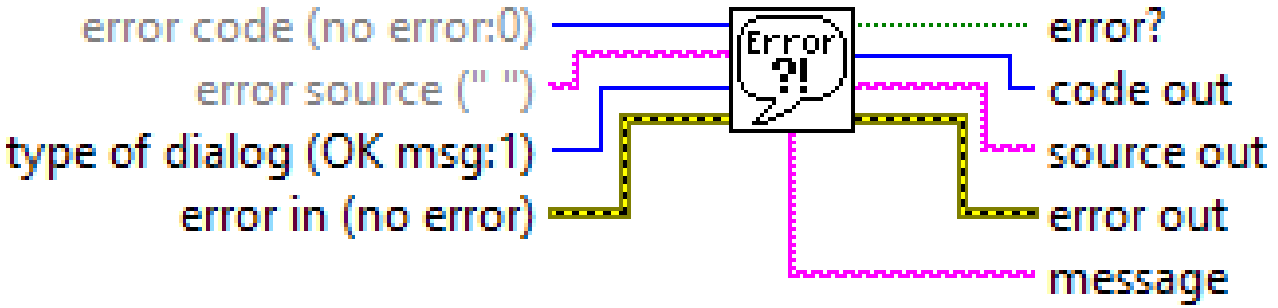
- LabVIEW ne kontroliše greške automatski. Kontorla grešaka se može ostvriti funkcijama na blok dijagramu virtuelnog instrumenta.
- Kontrola grešaka prati data-flow paradigmu. Kao što se podaci prenose kroz virtuelni instrumet, prenosi se i informacija o greškama.
- Povežite tok greške od početka (prve funkcije) do kraja virtuelnog instrumenta.



Simple Error Handler funkcija

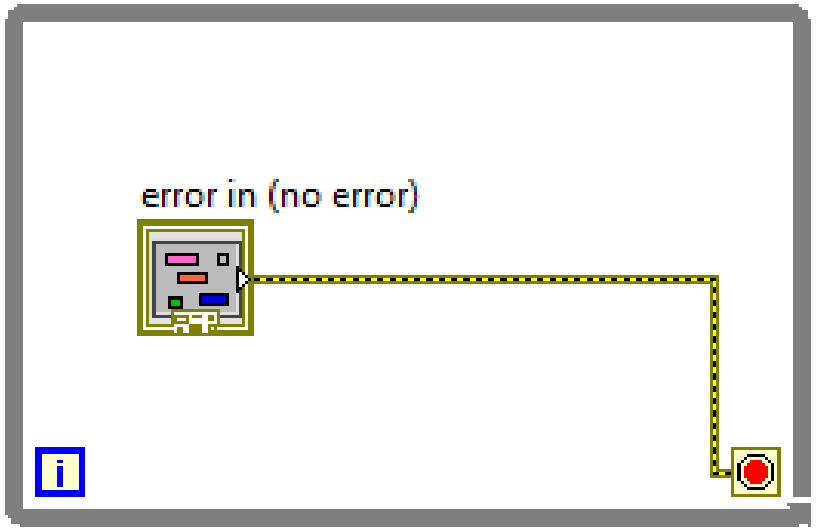
- **Simple Error Handler** funkcija služi za kontrolu greške na kraju izvršavanja virtuelnog instrumenta.

- Simple Error Handler se nalazi u located on the **Functions»Programming » Dialog & User Interface** paleti.



Kontrola greške u While petlji

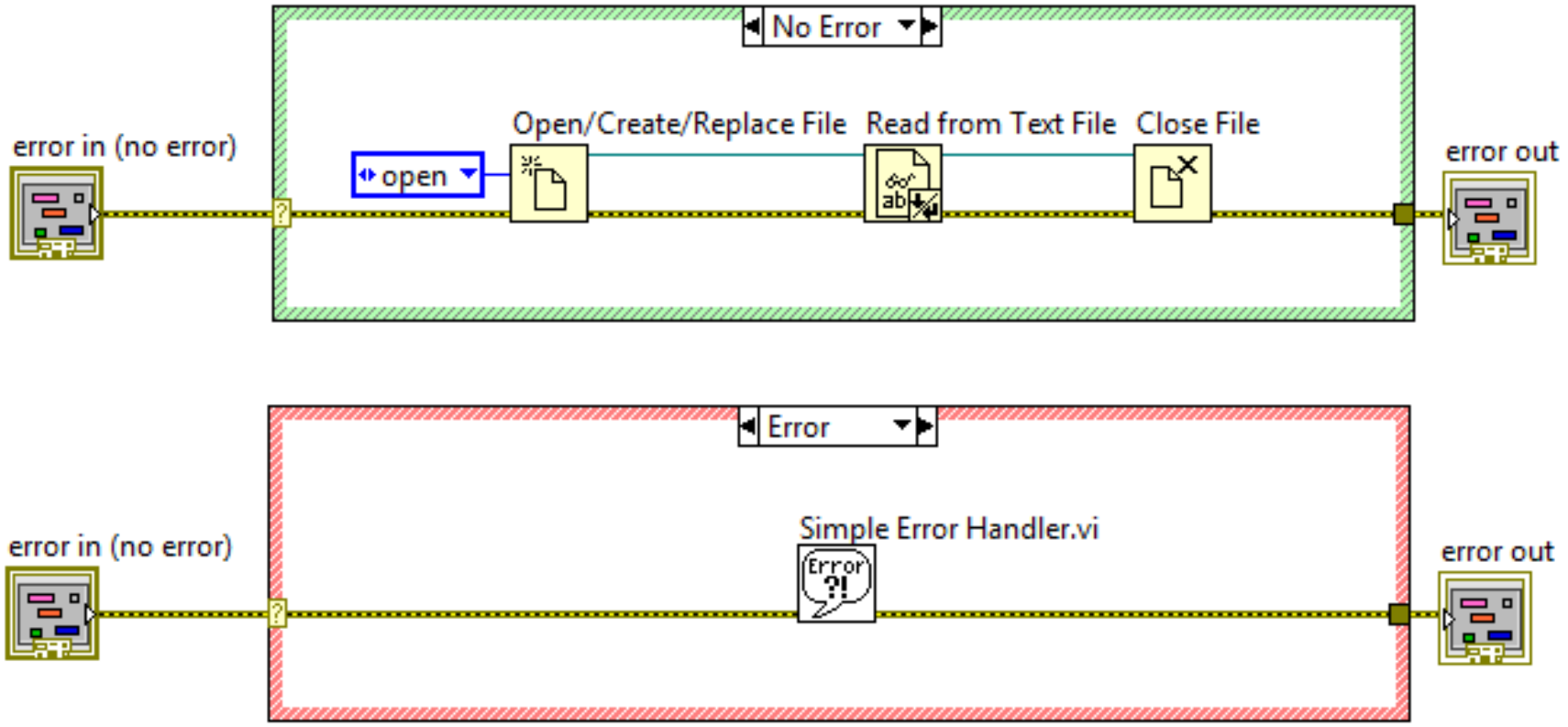
- Klaster za kontrolu greške se može vezati za Conditional terminal While petlje, čime se zaustavlja petlja u slučaju greške.
- U ovom slučaju, samo Status element klastera koji ima TRUE ili FALSE vrednost se prosleđuje terminalu.



Kontrola greške pomoću Case strukture

- Klaster za kontrolu greške se može vezati za selektorski terminal Case strukture.
- U ovom slučaju, samo Status element klastera koji ima TRUE ili FALSE vrednost se prosleđuje terminalu.
- Case struktura u ovom slučaju ima dva subdijagrama: Error i No Error.
- Subdijagrami se izvršavaju u zavisnosti da li se pojavila greška

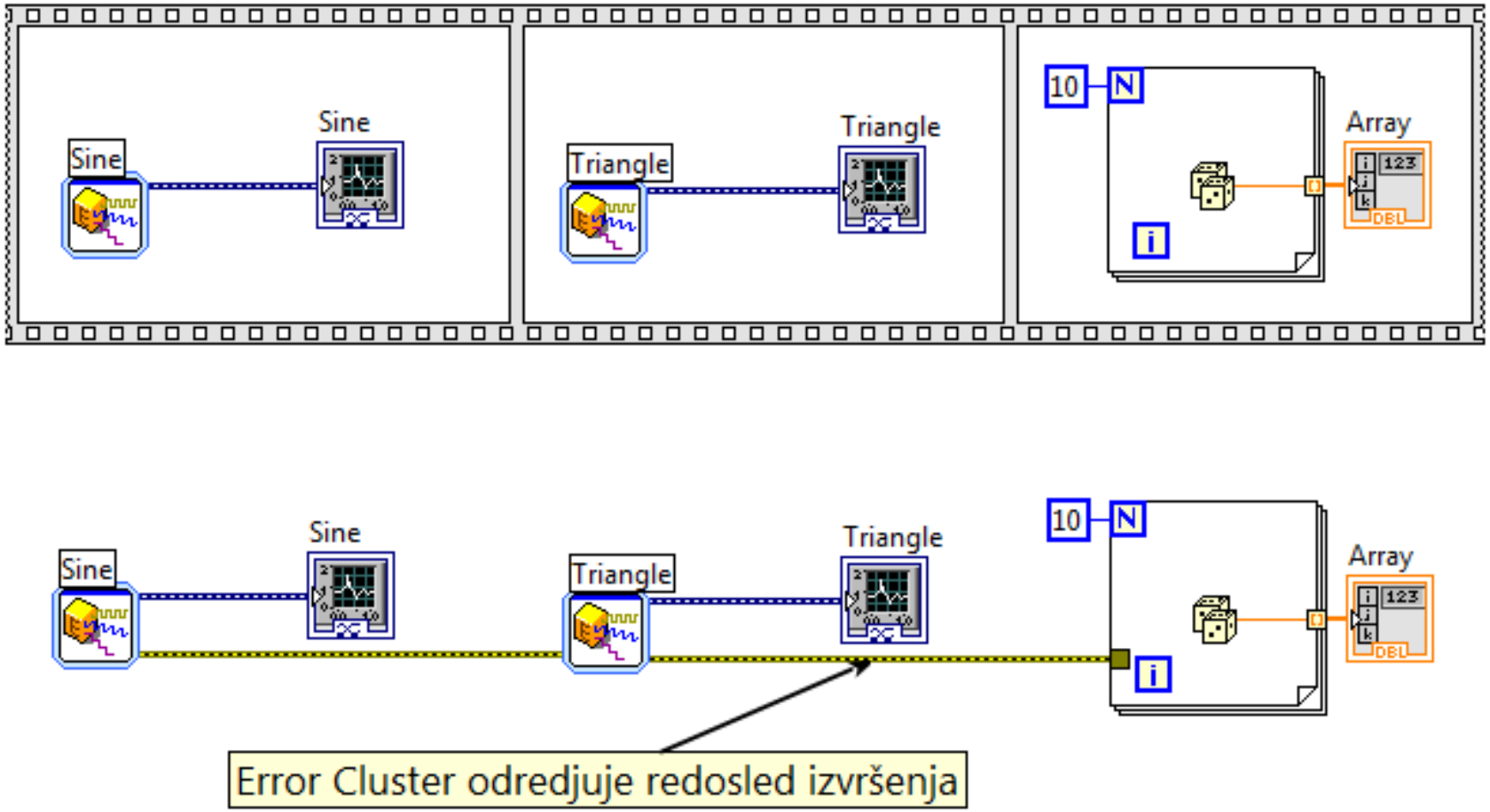
Kontrola greške pomoću Case strukture



Redosled izvršavanja

- Redosled izvršavanja kôda se reguliše strukturama **Flat Sequence** ili **Stacked Sequence**.
- Redosled se može kontrolisati data-flow konceptom, iako ne postoji eksplicitna zavisnost izvršavanja koncepta
- Error Cluster data-flow se može iskoristiti za određivanje redosleda izvršavanja

Redosled izvršavanja

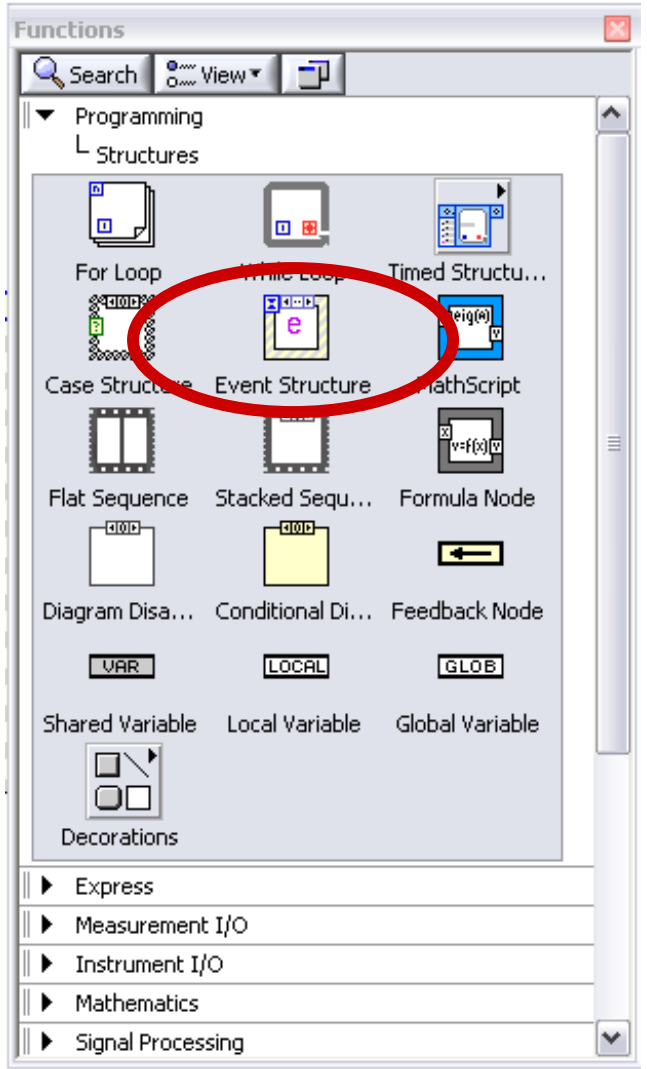


Pregled

- Kontrola grešaka pruža uvid u greške nastale u toku izvršavanja virtuelnog instrumenta.
- Klaster za kontrolu greške sadrži informaciju o statusu, tipu greške i lokaciji nastanka greške.
- error in i error out klasteri se koriste za ulaz/izlaz informacije o greškama u subVI.
- Terminali Case strukture i While petlje mogu se vezati za error klaster, pri čemu terminali prepoznaju status greške.
- Error Cluster se može iskoristiti za određivanje redsoleda izvršavanja čvorova

Korisnički interfejs i događaji

- Ukoliko je virtuelni instrument aplikacija sa složenim korisničkim interfejsom, implementacija događaja (events) može značajno poboljšati performanse
- Dobijeni kôd je pregledniji i jednostavniji
- Najčešće se koriste dva metoda kontrole korisničkog interfejsa: **polling** i **event-driven** metod.



Efikasan korisnički interfejs

- Prikazuje informacije intuitivno i jasno
  - Jasne labele koje opisuju kontrole i indikatore
  - Ne preterivati sa detaljima, fontovima (maks. 3), i dekoracijama
- Pruža laku kontrolu korisnika nad virtuelnim instrumentom
  - Unošenje vrednosti
  - Menjanje parametara
  - Drag-n-drop
- National Instruments Style Guide

Korisnički interfejs realizovan polling metodom

- While petlja proverava stanje kontrola
- Ukoliko je došlo do promene stanja, registruje se događaj
- U zavisnosti od događaja, izvršava se neki deo koda
- Nedostaci polling metode
  - Opterećuje CPU
    - Vrednost kontrola se proverava svaku iteraciju petlje
    - Aplikacija može biti spora ili zamrznuta
  - Može da ispusti korisnikovu akciju
    - Reakcija zavisi od toka podataka

Event-driven metod

- Efikasniji metod od polling metoda
- Ubrzava odziv aplikacije
- Kôd je pregledniji i jednostavniji
- Najčešće korišćen metod u LabVIEW, podržan Event Structure strukturom

Događaji (events)

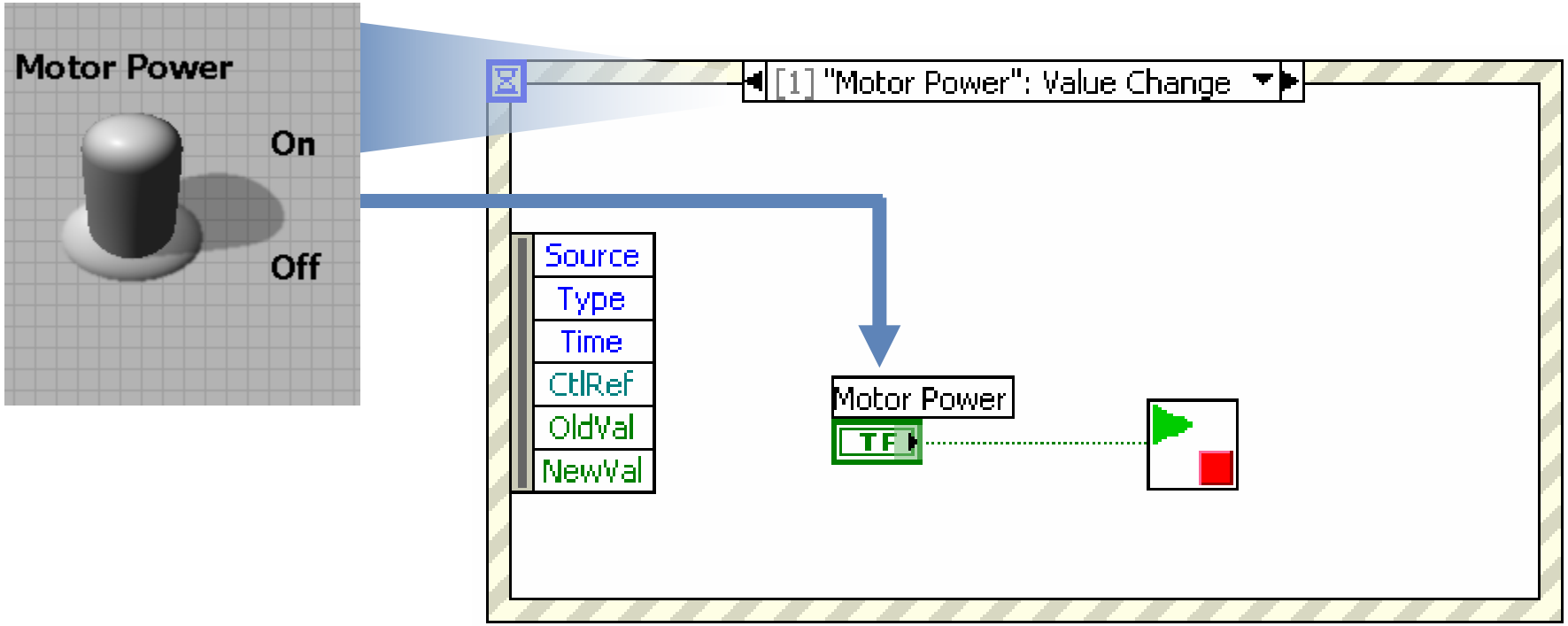
- Događaj je akcija koja je inicirana promenom stanja
  - Korisničkom intervenciom
    - Pritisak na dugme interfejsa
    - Klik miša
  - Iniciran od strane operativnog sistema
    - timeout
  - Iniciran od strane softvera (aplikacije)
    - Poruka druge aplikacije ili procesa
    - Varijabla ima određenu vrednost

Događaji elemenata korisničkog interfejsa

- Pritisci (klikovi) na dugmad
- Unos podataka u kontrole
- Promena vrednosti kontrole/indikatora
- Drag and drop operacije
- Kontekstni meniji (desni klik)

Promena vrednosti kontrole

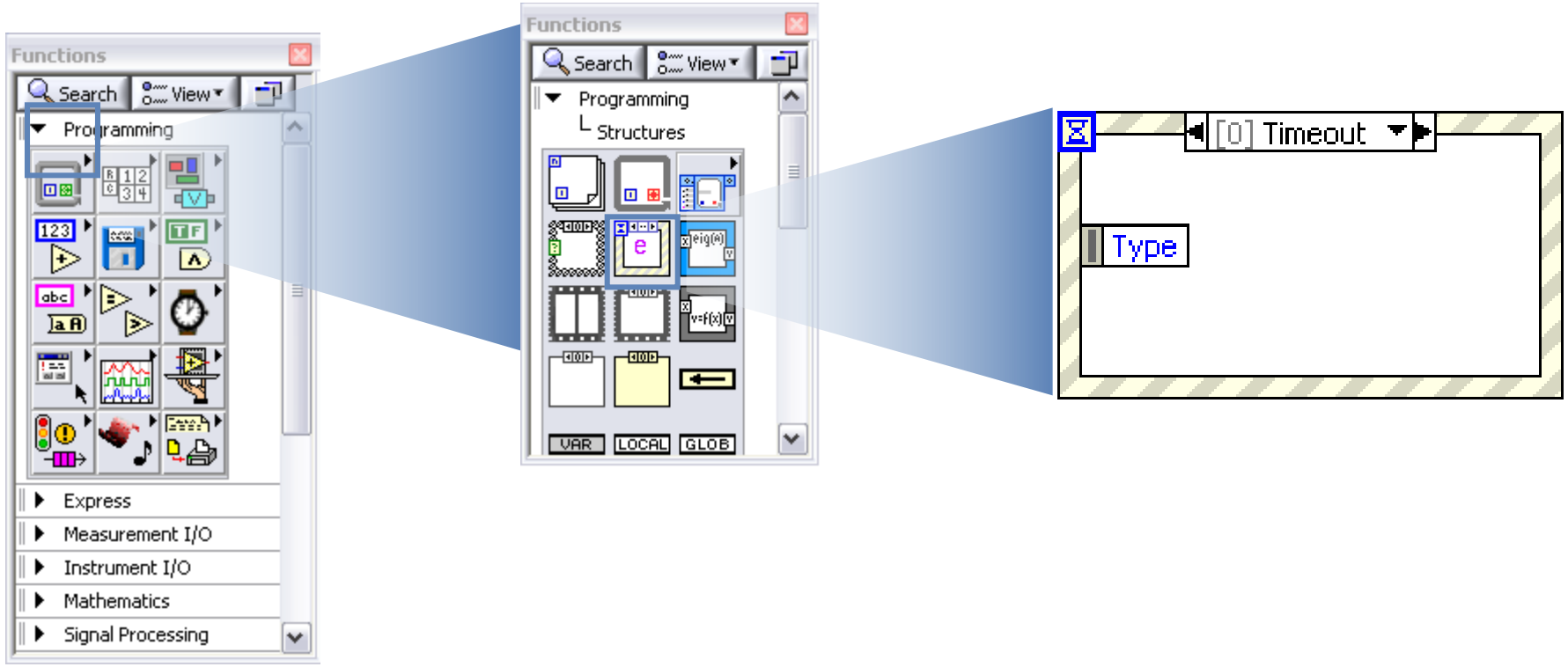
- Primer događaja je promena vrednosti logičke promenljive - kontrole **Motor Power**



Event Handler

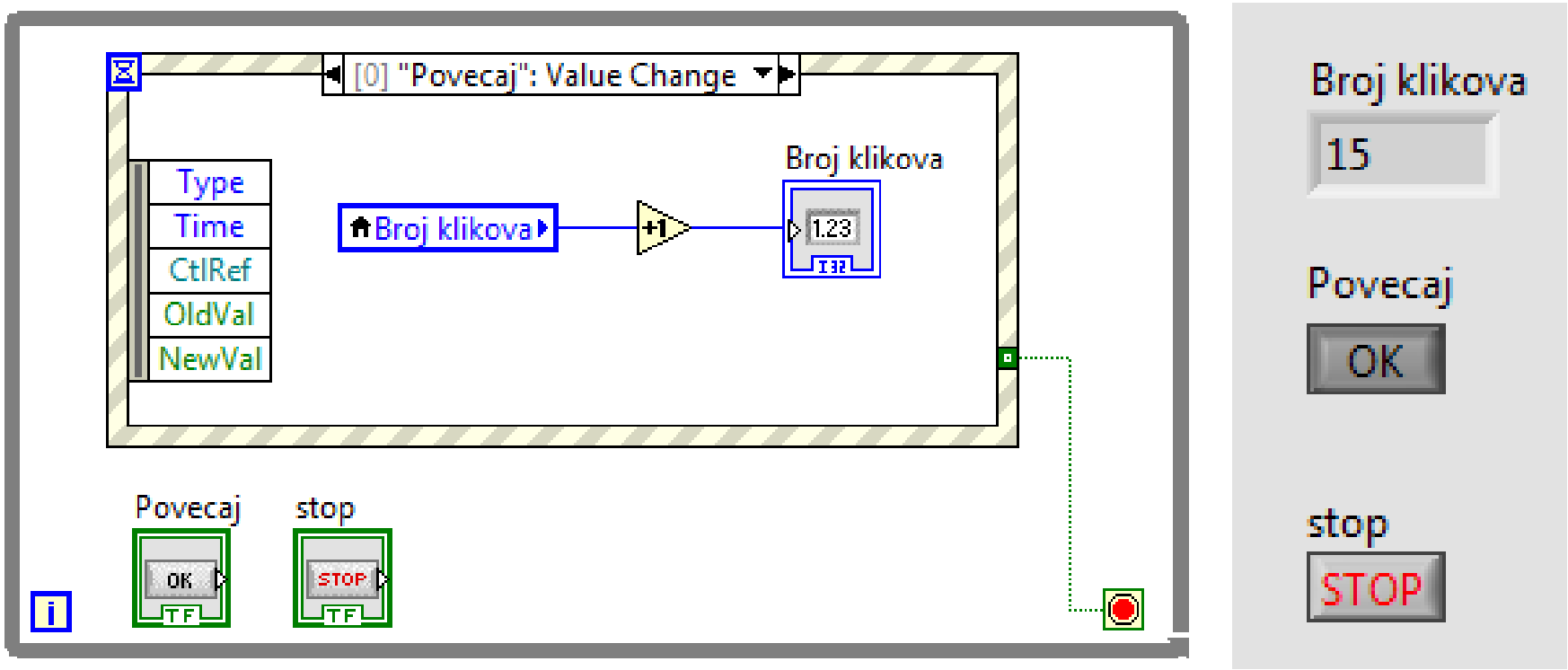
- Mehanizam komunikacije procesa i operativnog sistema
- Interpretira informacije o svakom događaju na niskom nivou
- Asinhrona callback rutina

LabVIEW Event Structure



LabVIEW Event Structure se može postaviti iz palete **Programming»Structures»Event Structure**

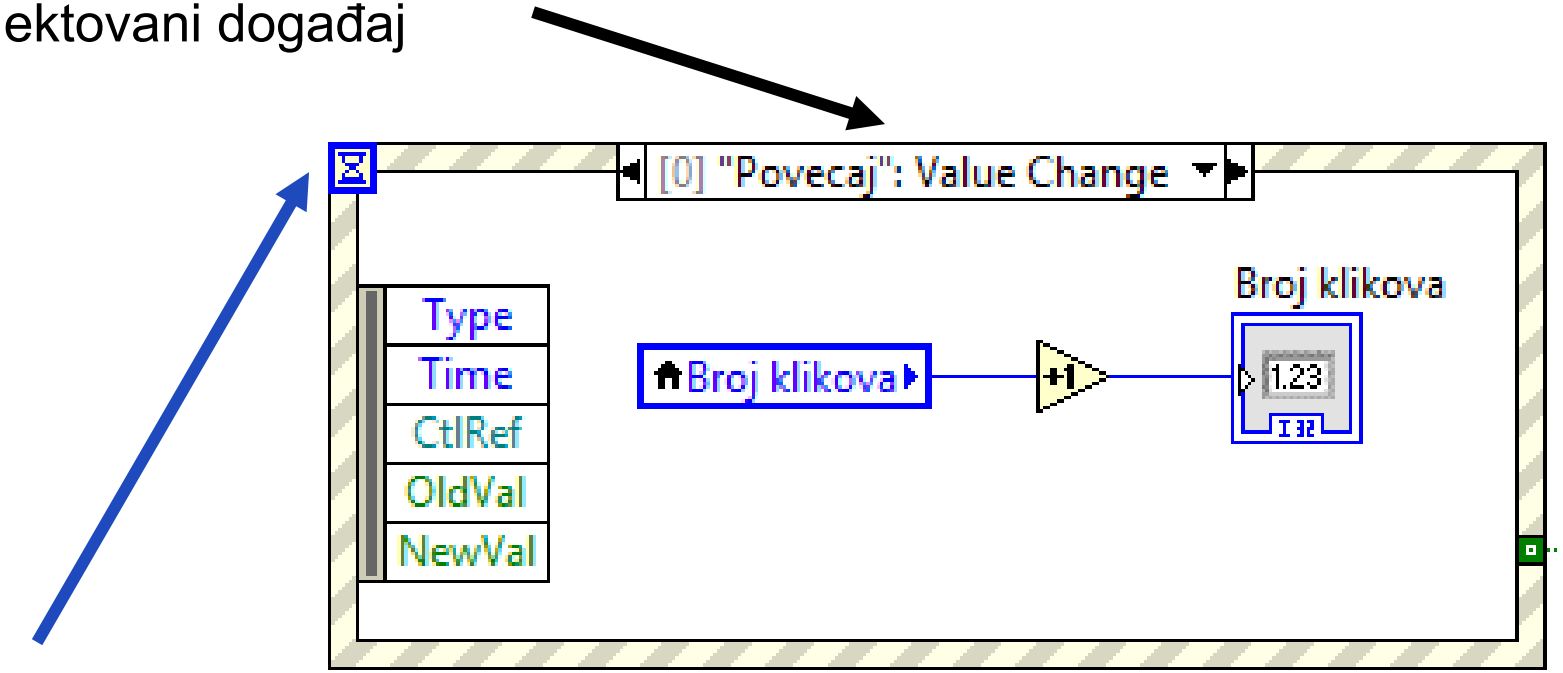
Primer Event Structure



- Klikom na kontrolu **Povecaj** aktiviraće se događaj „Povecaj“: Value Change, **Broj klikova** se inkrementira

Elementi Event Structure

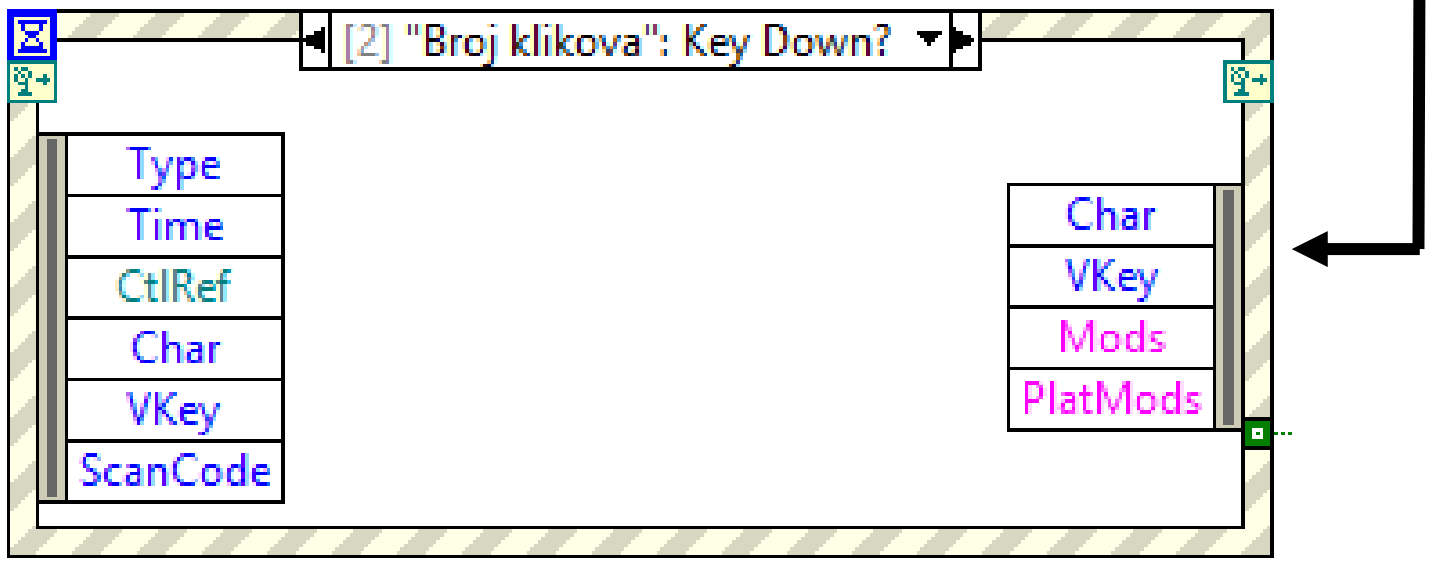
Event Selector Label – prikazuje kôd za selektovani događaj



Timeout – vreme u milisekundama koje se čeka na događaj podrazumevana vrednost je –1 (beskonačno)

Parts of an Event Structure

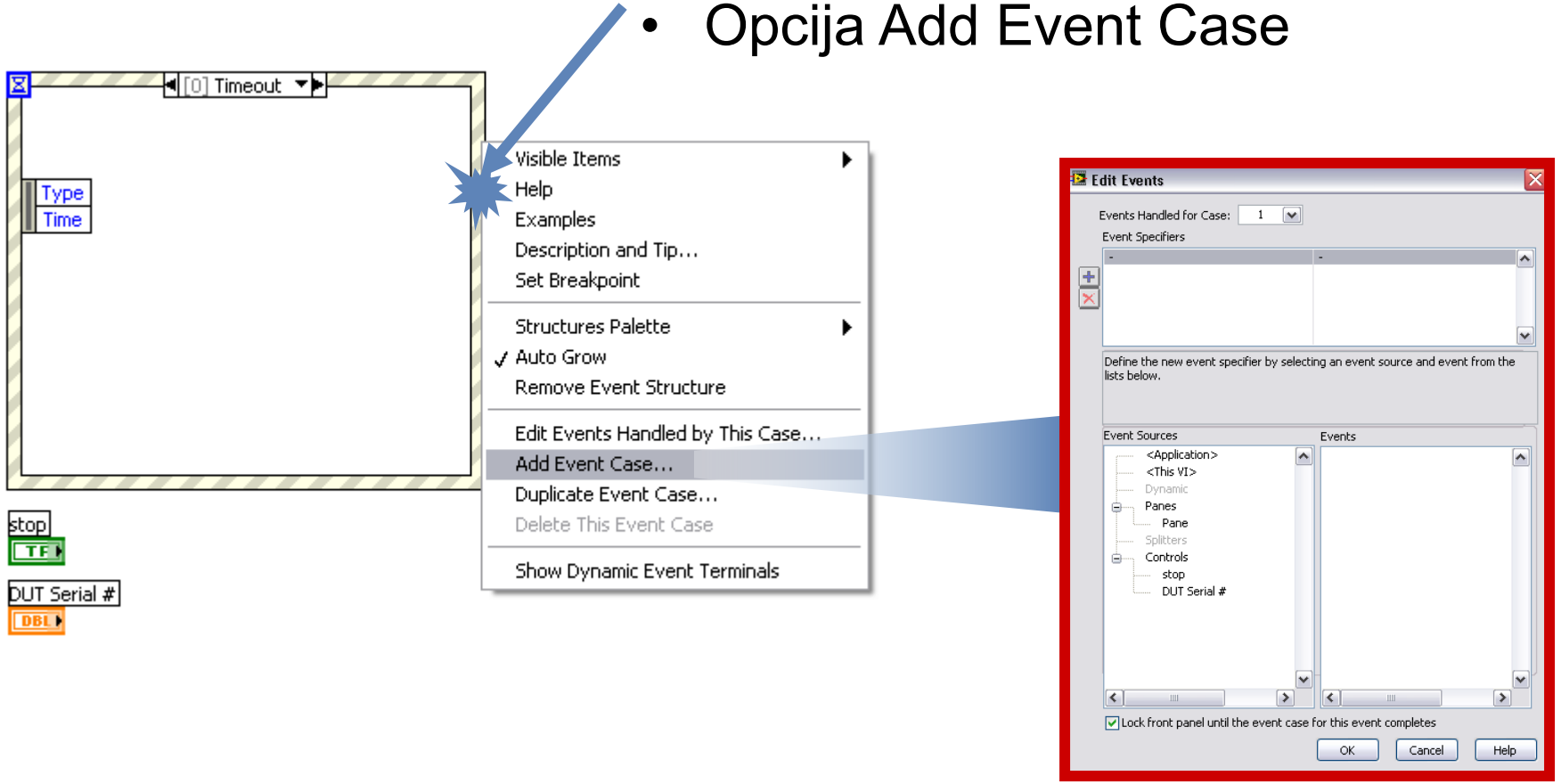
Event Filter Node – omogućava menjanje ili odbacivanje podataka o događaju pre nego što je događaj pocesiran



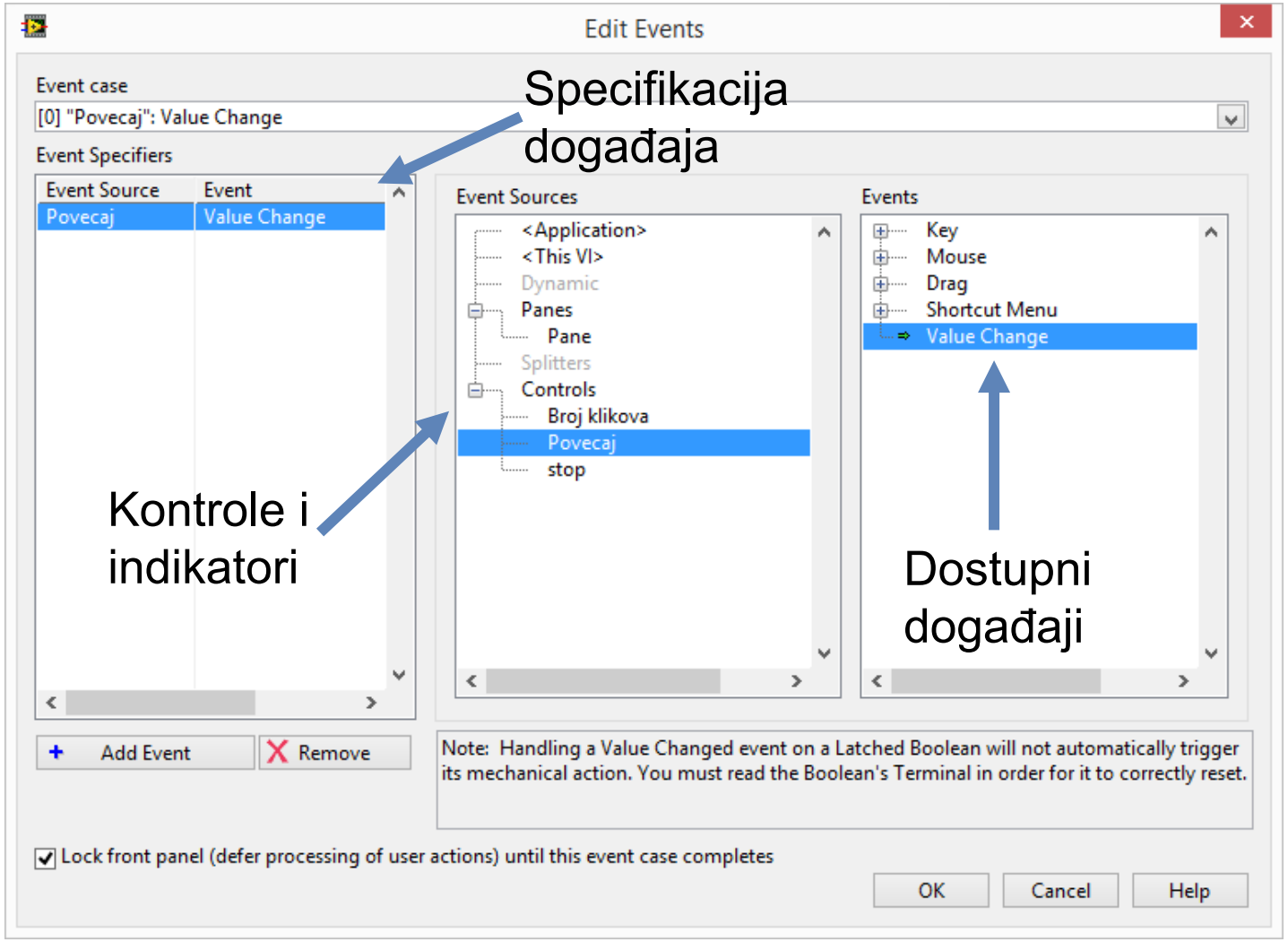
Event Data Node – podaci jedinstveni za svaki događaj

Dodavanje subdijagrama u Event Case

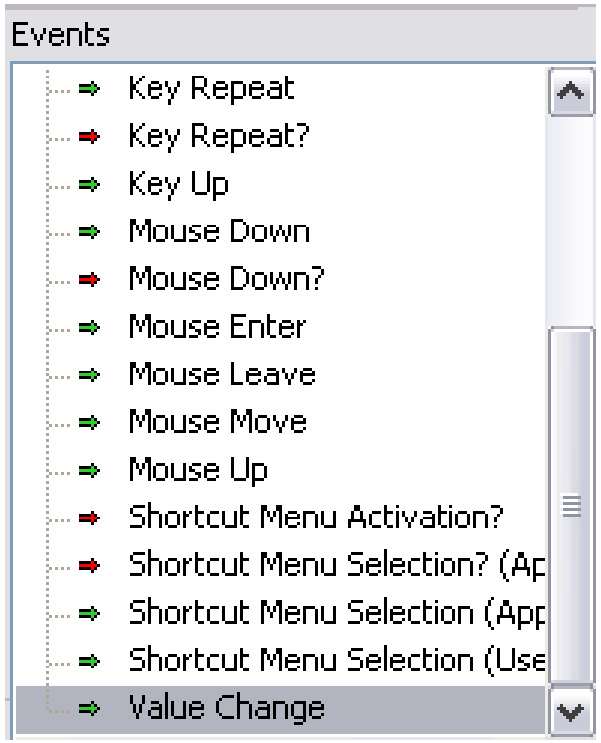
- Desni klik na strukturu
- Opcija Add Event Case



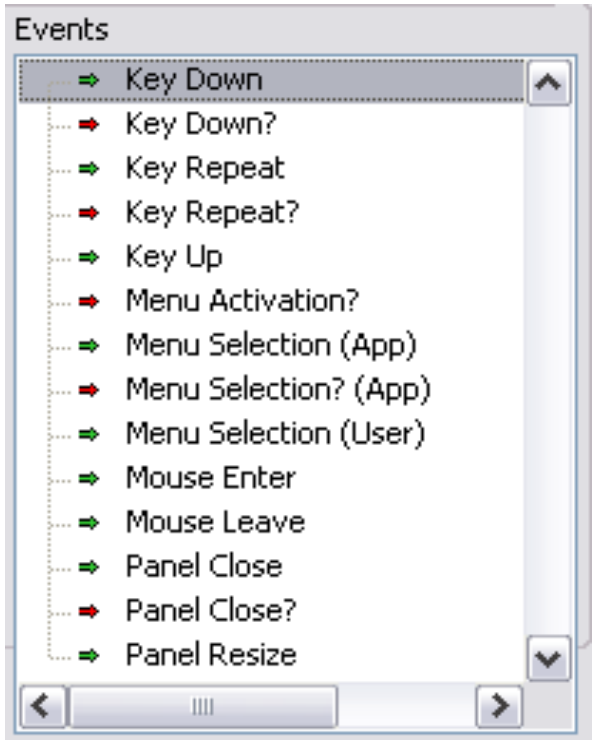
Dijalog za definisanje događaja



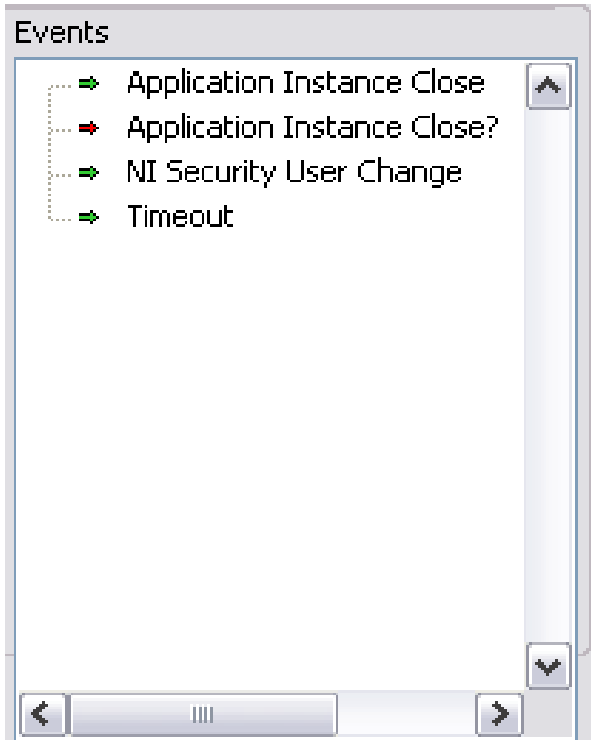
Tipovi događaja



Controls

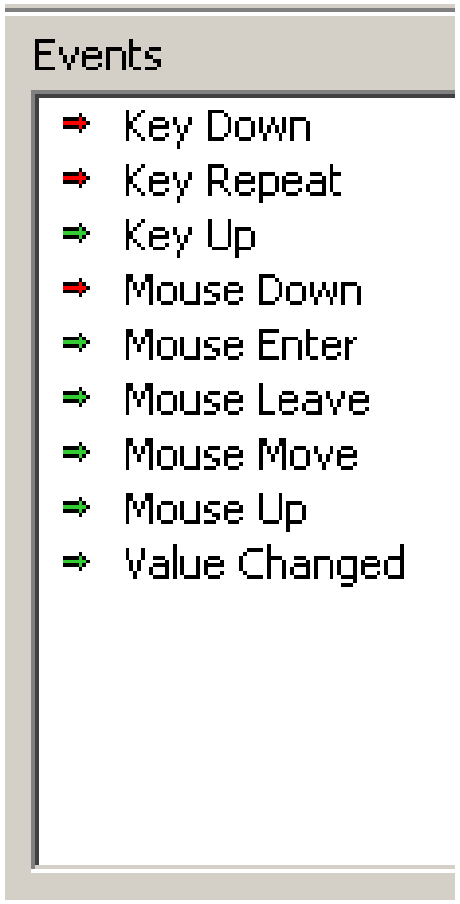


This VI



Application

Notify i Filter događaji



Notify Events

Obaveštavaju da se dogodila interakcija sa korisnikom već dogodila; dostupan je samo Event Data Node.

Filter Events

Proverava i modifikuju podatke o događaju pre nego što ih korisnički interfejs procesira, sprečavajući da dođe do promene u VI; dostupni su Event Filter Node i Event Data Node.

Ostali događaji

- Kontekstni meniji
  - Registruje događaje na osnovu aktiviranja i izbora opcije iz kontekstnih menija.
  - Funkcioniše sa korisnički definisanim menijima ili menijem aplikacije.
- Drag and Drop
  - Drag and drop podataka između kontrola i indikatora
  - Alternativa copy/paste

Saveti

- Event Structure postavite u While petlju
- Logičke kontrole (latched) postavite u subdijagrame koji obrađuju događaje vezane za te kontrole
- Koristite Stop događaj za zaustavljanje petlje
- Nikada ne postavljajte jednu Event strukturu u drugu

Pregled

- Struktura **Event Structure** je neaktivna dok se ne registruje događaj
- Kada je događaj registrovan, izvršava se odgovarajući subdijagram u strukturi
- Struktura se izvršava samo jednom, prilikom registrovanja prvog događaja.
- Svi događaji se stavljaju u red čekanja
- Izvršava subdijagrame asocirane događajima prema redosledu registrovanih događaja.

Vežba 10 – Kalkulator sa Event strukturom

Jednostavan kalkulator

- Realizujte VI koji ima funkciju kalkulatora
- Kalkulator ima funkcije sabiranja, oduzimanja, množenja, deljenja i kvadratnog korena.
- Operacije se izvršavaju klikom na odgovarajući taster.
- VI se izvršava kontinualno, može se prekinuti klikom na taster **Stop**

Povezivanje instrumenata preko serijskog interfejsa

Povezivanje instrumenata preko serijskog interfejsa

- **V**irtual **I**nstrument **S**oftware **A**rchitecture (VISA)
- Serijska komunikacija

VISA

- VISA (**V**irtual **I**nstrument **S**oftware **A**rchitecture) je standard za konfigurisanje i programiranje instrumenata i mernih sistema povezanih preko GPIB, VXI, PXI, Serial, Ethernet, i USB interfejsa.
- VISA se sastoji od sledećih komponenti:
  - Softverskih biblioteka
  - Interaktivnih alata
  - Konfiguracionih alata.

Serijska komunikacija

- Serijski (RS-232) port

Konektori

Kabl

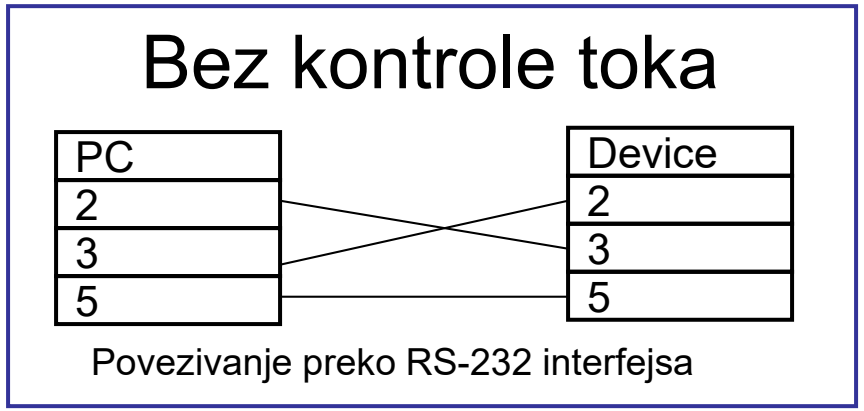
4 x RS-232 to PCI

RS-232 to USB

RS232 interfejs

- Serijski interfejs koristi jednu liniju za prenos jednog bita podatka od predajnika ka prijemniku.

number	name	description	level	Dir.	Etc.
1	DCD	Data Carrier Detect	RS232	Input	Mandatory
2	RXD	Receive Data	RS232	Input	Mandatory
3	TXD	Transmit Data	RS232	Output	Mandatory
4	DTR	Data Terminal Ready	RS232	Output	Optional
5	GND	Ground	Ground	-	Mandatory
6	DSR	Data Set Ready	RS232	Input	Optional
7	RTS	Request To Send	RS232	Output	Optional
8	CTS	Clear To Send	RS232	Input	Optional
9	RI	Ring Indicator	RS232	Input	Optional

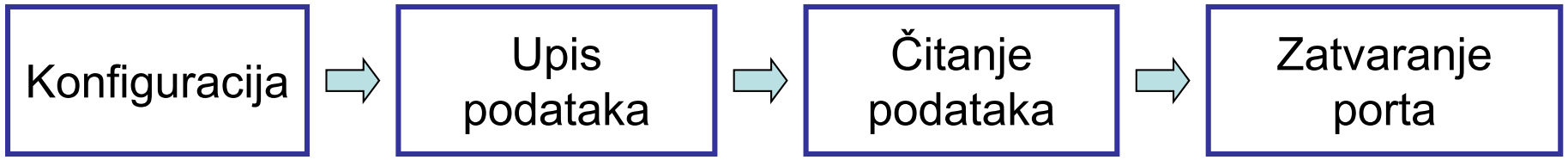


Bez hardverske kontrole toka (RTS/CTS)

Pinovi konektora

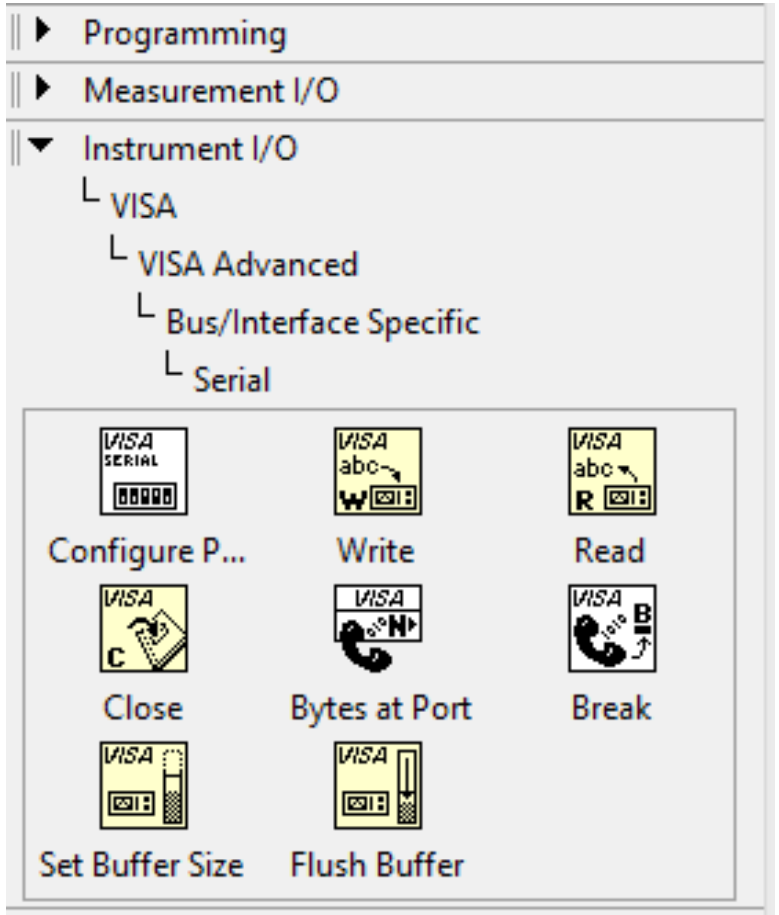
Serijska komunikacija

Model jednostavne komunikacije

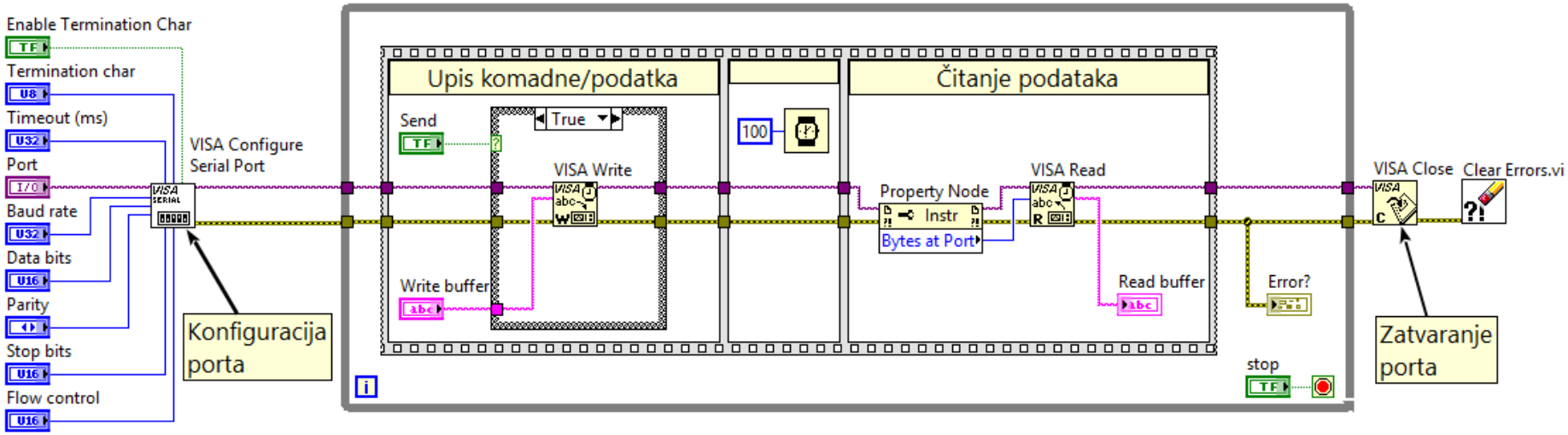


VISA paleta

- U VISA paleti se nalaze funkcije za konfiguraciju porta, čitanje podataka, upis podataka, konfigurisanje i brisanje bafera i zatvaranje porta.

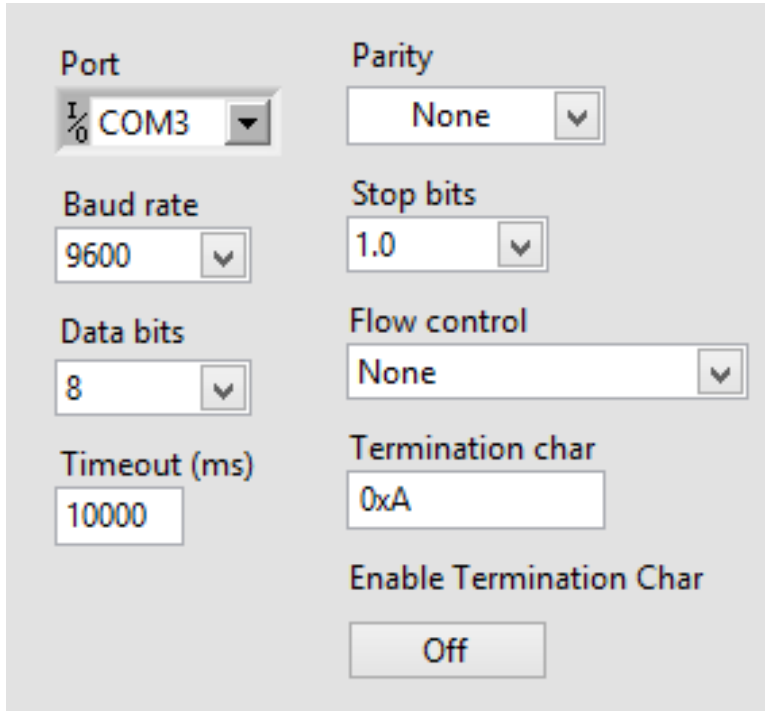


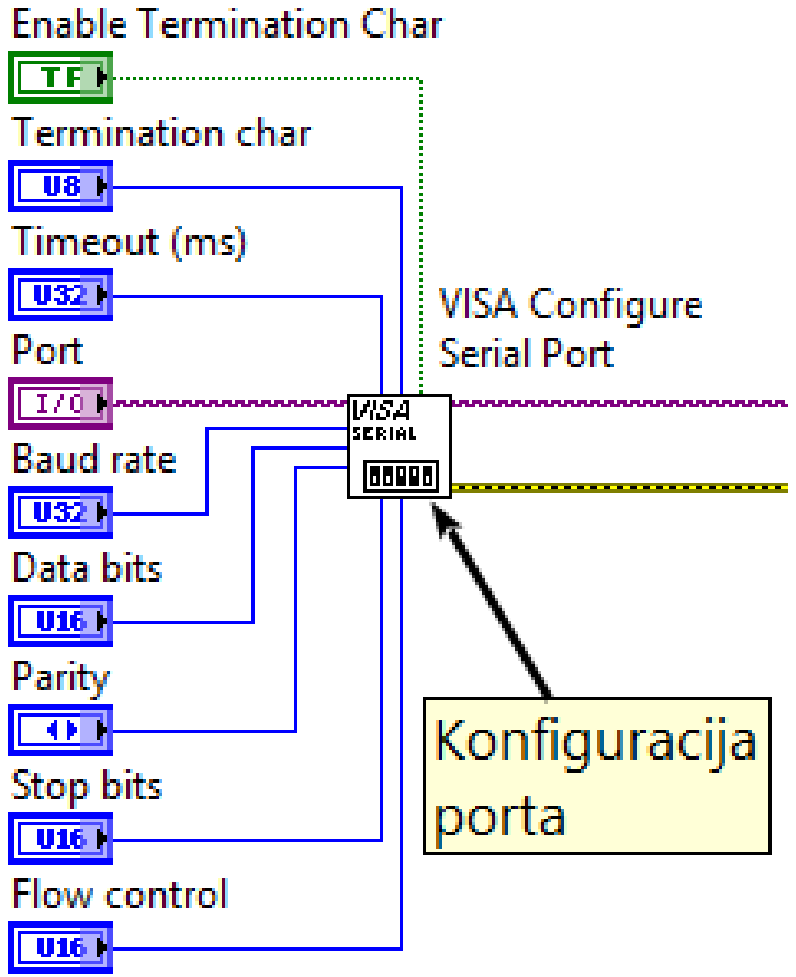
Primer VI za komunikaciju



Konfiguracija porta

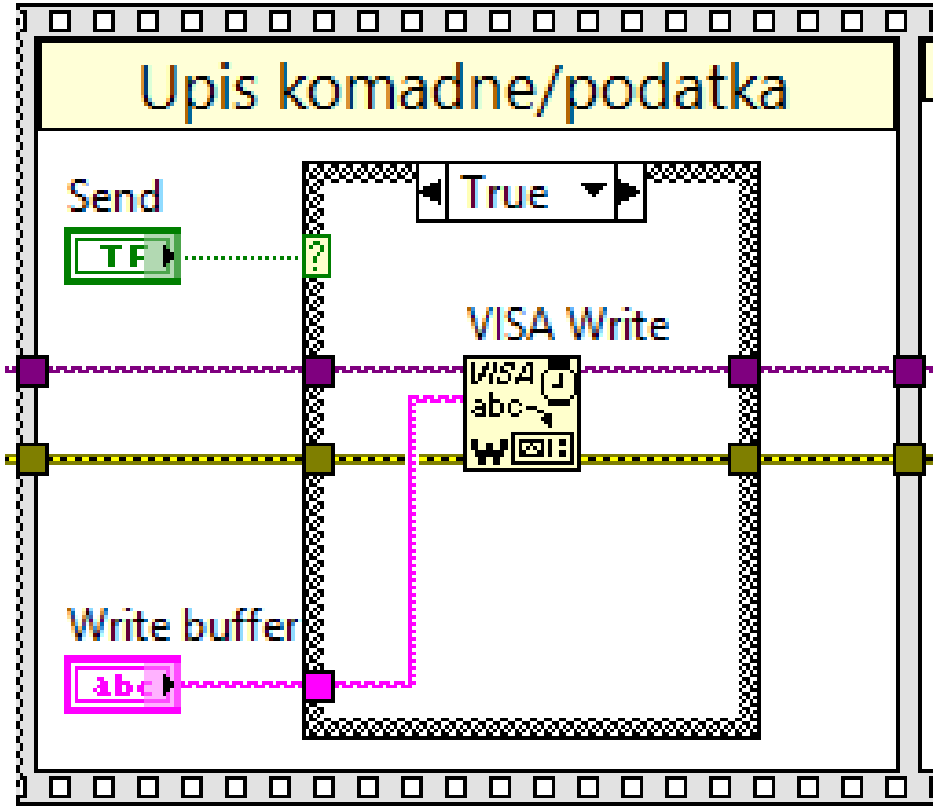
- Konfiguracija porta (baud rate, data bits, parity, stop bits and flow control…)





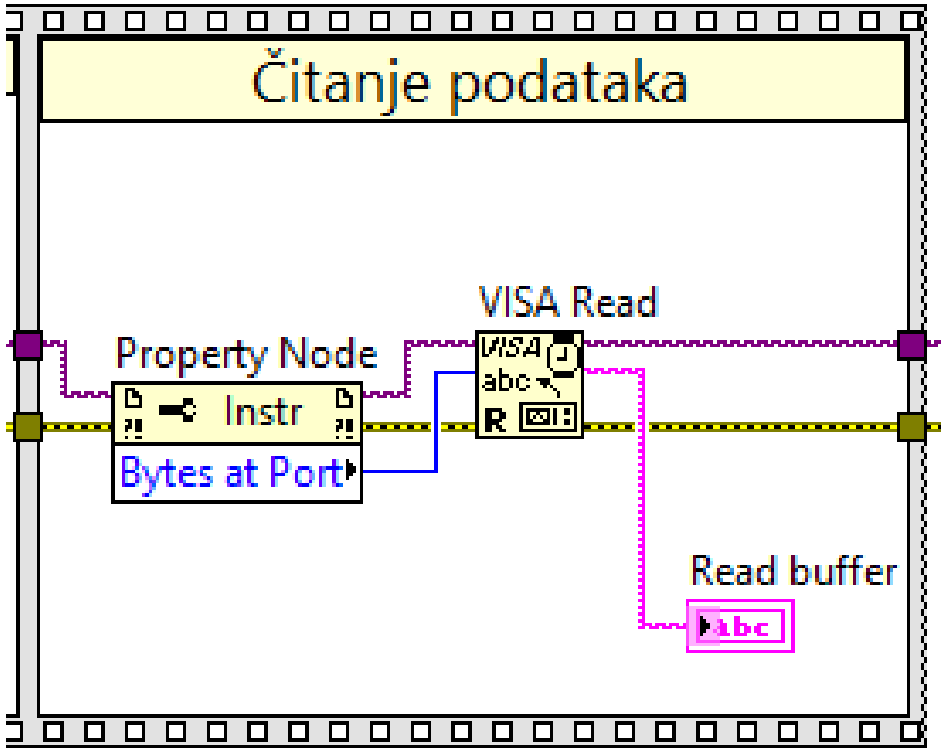
Upis podataka

- Upis podataka preko serijskog porta ostvaruje se **VISA Write** funkciojom. Ulazni podatak **Write Buffer** je tipa string.
- Između funkcija upisa i čitanja je potrebna vremenska pauza



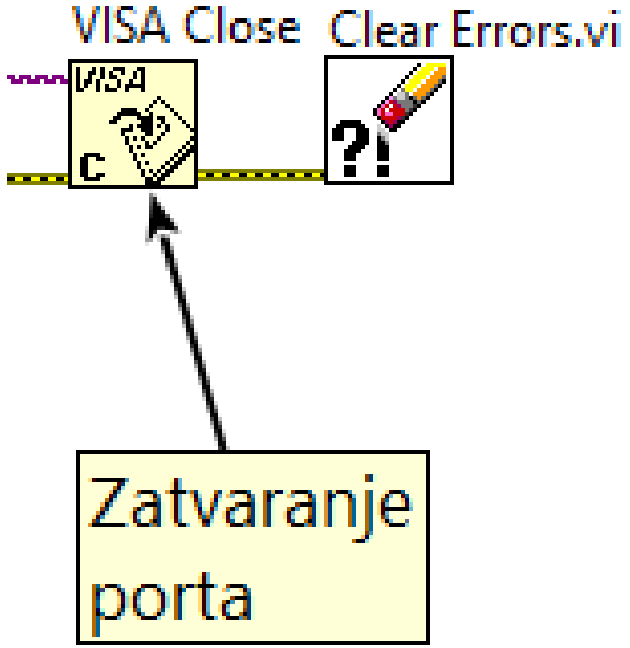
Čitanje podataka

- Čitanje podataka preko serijskog porta ostvaruje se **VISA Read** funkcijom. Ulazni podatak je **Bytes to Read**. Izlazni podatak **Read Buffer** je tipa string.
- Da bi se pročitala informacija, prvo je potrebno odrediti broj bajtova u ulaznom baferu. Broj bajtova se može dobiti **Bytes at Port** funkcijom.



Čitanje podataka

- Na kraju izvršavanja je potrebno zatvoriti port, kako bi bio dostupan drugim aplikacijama



Pregled

- LabVIEW poseduje veliki broj biblioteka za kontrolu instrumenata drugih proizvođača
- Mogu se povezati uređaji preko GPIB, VXI, PXI, Serial, Ethernet, i USB interfejsa.
- Preko serijskog interfejsa (RS232/455/485) je moguće kontrolisati instrumente i uređaje jednostavnim VISA modelom.