

Laborator 10 - Programare pe Evenimente - Labview

Programare, an 1, MTR+MEC, www.viacolab.utcluj.ro, Prof. Iulian Lupea

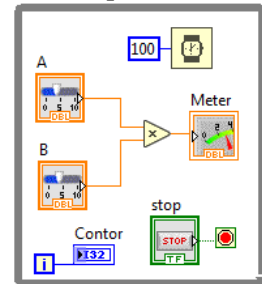
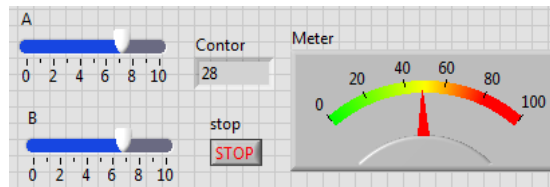
Event Structure conține cazuri multiple (asemănător cu Case structure) fiecare caz fiind executat la apariția unui eveniment sau grup de evenimente (schimbarea valorii unui control, intrare sau ieșire mouse într-o/dintr-o zona de pe PF, închiderea / schimbarea marimii unei ferestre, apăsarea unei taste, timeout etc.). Adesea se plasează într-un ciclu While.

1. PF interactiv - Ciclul While (polling) versus Event Structure

Efectuăm produsul a două controale numerice A și B cu observarea permanentă a modificării valorilor și a produsului valorilor.

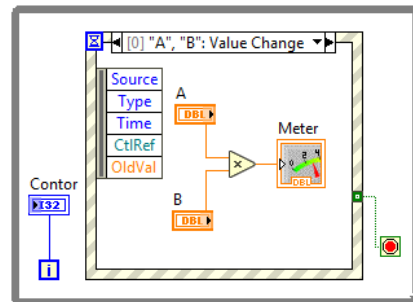
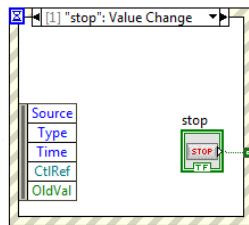
1.1. Varianta#1:

produsul este plasat în corpul ciclului While. Când nu se modifică valorile A sau B ciclarea continuă consumând resurse de calcul. Observați contorul i.



1.2. Varianta#2: programare pe evenimente - Event Structure

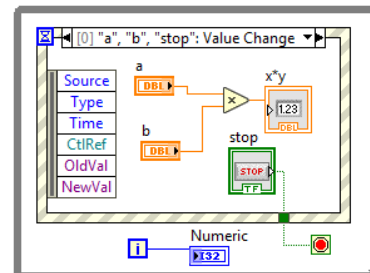
În aplicația precedentă păstrăm Ciclul while dar produsul valorilor numerice este plasat în primul caz '[0]"A","B":Value Change' al structurii Event, creat prin modificarea cazului '0 Timeout' cu comanda 'Edit Events Handled by This Case'. Se deschide o fereastră de configurare/editare evenimente și se adaugă două evenimente de tip Value Change, sursa evenimentelor fiind controalele A și B.



Oprirea ciclului și a aplicației este posibilă dacă se introduce un al doilea caz de tip Value Change în structura Event care conține terminalul butonului de STOP. Acest caz se execută la apăsarea STOP în PF iar la ieșirea din caz (după tratare Event), valoarea True a butonului oprește ciclul While.

La execuție se observă că ciclarea este suspendată (Contorul nu crește) când nu sunt înregistrate evenimente de tipul modificării valorii controalelor A sau B. Când schimbăm din PF valorile controalelor A sau B contorul ciclului crește iar înmulțirea se efectuează.

Obs. se pot trata cele trei evenimente într-un singur caz.



1.3. Cazul Timeout al structurii Event.

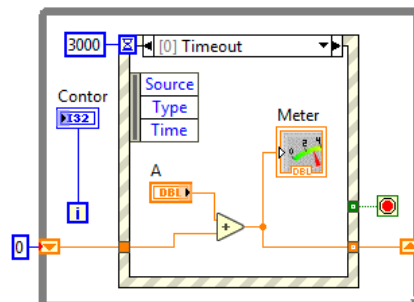
Aplicația conține două cazuri: cazul Timeout și cazul tip Value Change pentru butonul de STOP al ciclului.

La fiecare 3000ms de așteptare dacă nu apare un eveniment se execută automat cazul 'Timeout' în care se adună valoarea lui A la registrul Shift și se afișează. Dacă schimbăm valoarea controlului A din PF nu se întâmplă nimic imediat, doar după 3000ms se adună valoarea schimbată a lui A (schimbarea lui A nu este considerată ca fiind eveniment).

1.3.2. Adăugați un caz nou pentru sesizarea schimbării valorii lui A și actualizarea imediată a sumei (vezi variabile locale pentru A și Meter).

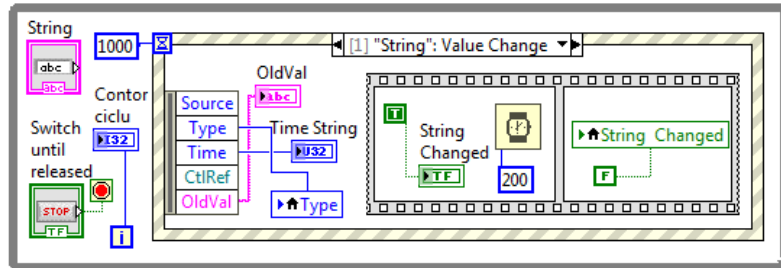
1.3.3. Includeți în cazul Timeout și evenimentul 'schimbare valoare A' (Value Chance).

1.3.4. Se va ilumina un led verde la fiecare 4 secunde și se contorizează și afișează timpul la fiecare 3 iluminări a ledului folosind Tick Count.



2. Eveniment static - schimbarea valorii unui control șir de caractere

Se include Event Structure în diagrama bloc. Implicit există cazul '[0] Timeout'. Se adaugă cu comanda 'Add Event Case' un nou caz pentru tratarea unui eveniment de tipul 'Value Change' - caz înregistrat static (Static Event Registration). Se va urmări schimbarea valorii unui control String. La apariția evenimentului se execută codul din cazul asociat vizualizat în figură.



Lansăm aplicația. Se execută repetitiv ciclul

While pe cazul '[0] Timeout' de fiecare dată după așteptare de 1000ms.

Evenimentul are loc dacă se tastează un șir caractere sau se modifică un șir existent (fără Enter) în controlul String urmat de Click pe suprafața PF.

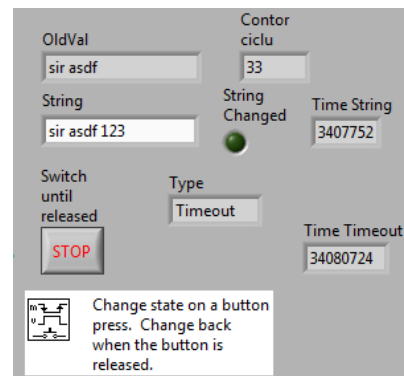
Acest eveniment declanșează execuția codului din caz [1]:

- a) se afișează în indicatorul OldVal vechiul șir care a fost modificat;
- b) se execută secvența indice '0' din structura Sequence în care LEDul indicator String Changed se luminează (T) pentru 200ms după care în secvența indice '1' LEDul este stins (F) și se iese din Event Structure.

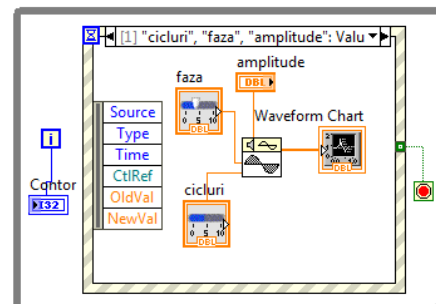
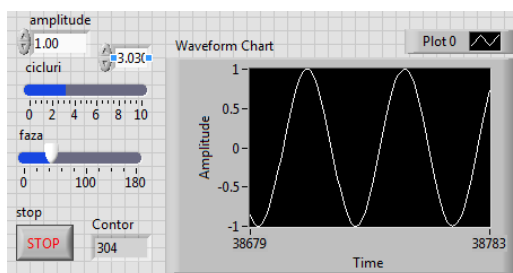
Așteptarea apariției unui eveniment (schimbare șir caractere) durează 1000ms. Dacă nu s-a schimbat șirul se execută evenimentul 'Timeout' după care se părăsește Structura event și se repetă ciclul While dacă Butonul STOP nu-i pus pe T (apăsă).

'Contor ciclu' incrementează la fiecare secundă (1000ms) dacă nu apare evenimentul.

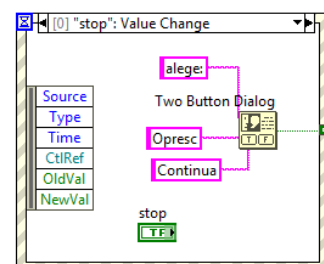
La execuția fiecărui caz se afișează în indicatorul unic(local variable) Type mesajul Timeout sau Value Change. Indicatoarele Time Timeout și Time String indica momentul execuției cazului respectiv. Butonul de STOP nu este tratat ca un eveniment motiv pentru care oprirea ciclului While se realizează prin apăsarea mai de durată a butonului de STOP.



3. Eveniment: schimbare parametri Sine Patern.vi + Grafic sinus + Buton 2 decizii



La schimbarea unui parametru SinePatern.vi se actualizează repetat graficul sinus. Dacă se apasă Stop, acest eveniment declanșează cazul [0]. Stop nu este laget la Terminal Ciclu While. Se alege între Opresc (T) și Continuă (F) asociat funcției Two Button Dialog.



4. Evenimente Mouse Enter, Mouse Leave și Value Change

Aplicația generează 'Nr val random' valori aleatoare, trasează graficul, trece graficul în mod Blink și revine, afișează (și apoi șterge) în indicatorul MESAJE etichetele unor indicatoare și controale când se trece cu mouse-ul peste zona alocată lor din PF.

Lista evenimente:

[0]: La apăsare START (schimbare T→F sau F→T) se generează Nr val random=200 și afișează grafic valori random;

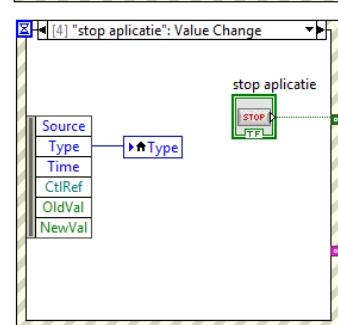
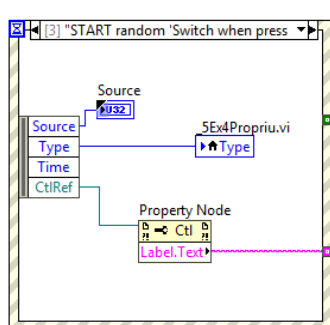
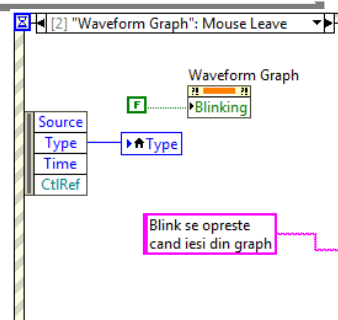
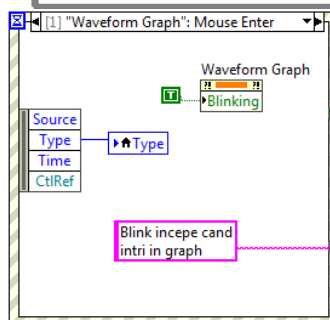
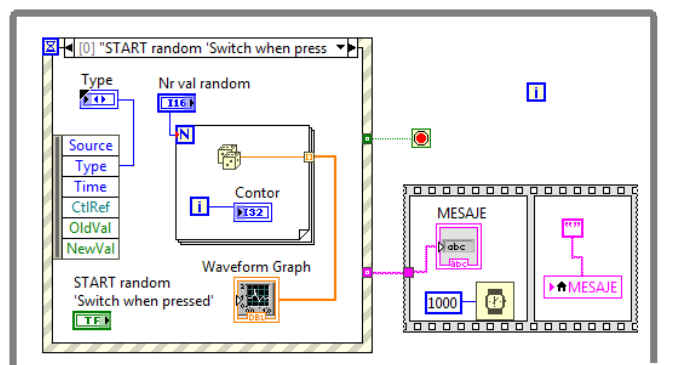
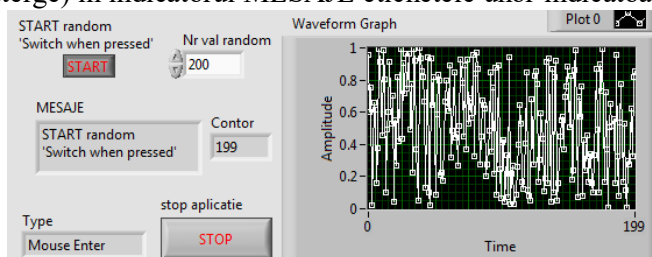
[1]: când Mouse intră în zona indicatorului grafic 'Waveform Graph', graficul începe 'Blink'; o constantă șir caractere trece în indicatorul MESAJE

[2]: când Mouse iese din zona indicatorului Waveform Graph se oprește grafic blink și o constantă șir caractere este trece în indicatorul MESAJE.

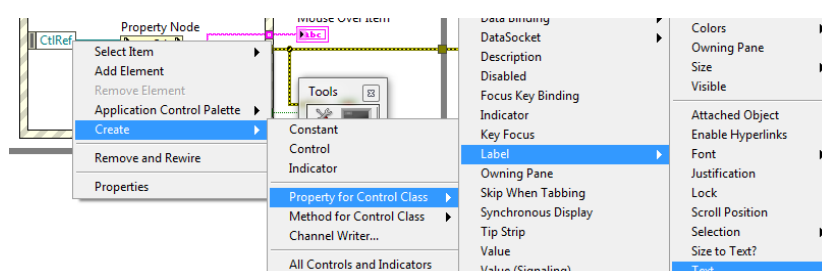
[3]: când Mouse intră în zona controlului 'Start random Switch when pressed' sau în zona controlului 'stop aplicatie' sau 'Nr val random' se afișează eticheta (șir caractere) acestor controale (CtlRef→Label.Text) în indicatorul șir de caractere MESAJE pentru o secundă. Evenimentul returnează CtlRef iar PropertyNode citește o proprietate

a unei referințe care poate fi obiect, clasă, metodă, VI (aici Property for Control Class);

[4]: la apăsare buton STOP se oprește ciclul While și aplicația curentă. Secvența indice 0 (din structura Sequence) afișează mesajul pentru 1 secundă după care în secvența următoare (indice 1) mesajul este șters prin afișare șir vid.



- [0] "START random 'Switch when pressed'": Value Change
- [1] "Waveform Graph": Mouse Enter
- ✓ [2] "Waveform Graph": Mouse Leave
- [3] "START random 'Switch when pressed'", "stop aplicatie", "Nr val random": Mouse Enter
- [4] "stop aplicatie": Value Change



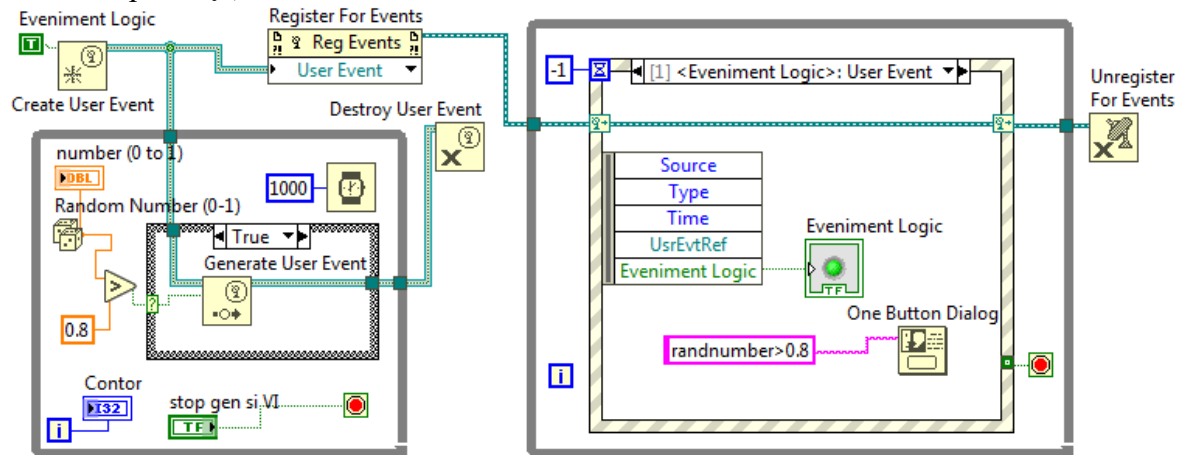
5. Eveniment *dinamic* declanșat prin program din Diagramă (nu din PF) fără intervenție User în PF

În aplicație sunt generate repetitiv numere aleatoare între 0 și 1. Dacă numărul aleator curent este mai mare decât 0.8 se declanșează un eveniment pentru care se afișează mesajul '*randnumber>0.8*' în PF. La generarea evenimentului și tratarea lui coparticipă funcțiile (paleta Dialog&User Interface/ Events):

Create User Event care cu ieșirea *user event out* se leagă la funcțiile

Generate User Event și Register For Events.

Evenimentul așteptat trebuie să aibă **tip** și **nume**. Prin intrarea **user event data type** a funcției Create User Event se stabilește Numele evenimentului (*Eveniment Logic* dat de utilizator) și Tipul de dată al evenimentului (*Logic* în acest caz) (valoarea True primită, nu are importanță).



Funcția **Create User Event** (paleta Events) returnează o referință la un User Event.

Funcția **Register For Events** înregistrează evenimentul în cadrul structurii Event (care are în exemplul curent 3 cazuri).

Funcția **Generate User Event** trimite evenimentul la toate structurile de tip Event Structure care au fost registrate pentru acel eveniment sau care conțin **un caz** pentru tratarea acestui eveniment.

Se creează în diagramă o structură Event cu 3 cazuri:

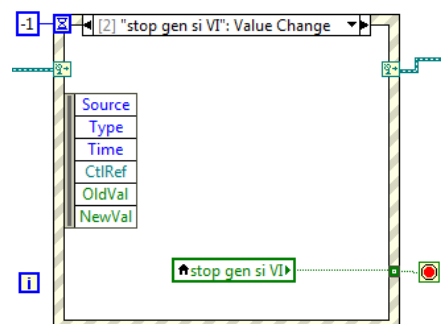
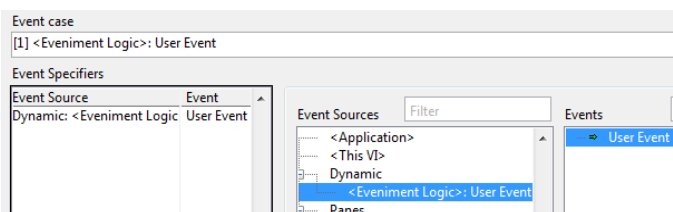
1. '[0] **Timeout**' există implicit; legarea la -1 = așteptare fără sfârșit. Se activează/bifează Show Dynamic Event Terminal la care se leagă ieșirea 1 a funcției Register For Events astfel încât la crearea unui nou caz să apară în fereastra de editare posibilitatea de a crea un Eveniment dinamic.

2. Cu comanda Add Event Case... se creează al doilea caz:

'[1] <Eveniment Logic>: User Event'. În fereastra deschisă (Event case, vezi Fig.) se selectează Event Sources: Dynamic/... Apoi se tratează cazul dynamic: se creează indicatorul Eveniment Logic și se creează mesajul *randnumber>0.8* care alimentează funcția One Button Dialog.

3. Cu comanda Add Event Case... se creează al treilea caz (caz înregistrat static) numit

'[2] **'stop gen si VI': Value Chance**'. Cazul urmărește evenimentul Value Chance a butonului STOP (cu eticheta 'stop gen si VI' din PF) schimbare care apare la apăsarea butonului. Apăsarea butonului (generează val. True) are acțiune dublă:

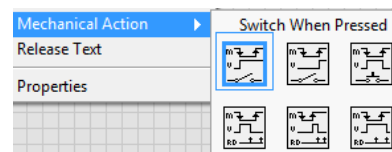


a) oprește ciclul While de generare numere aleatoare + generare eveniment dynamic și b) se execută cazul [3] din Event Case pentru oprirea ciclului While care conține Event Structure cu 3 cazuri; oprindu-se astfel întreaga aplicație sau VI.

Event Structure este pusă în ciclul WHILE deoarece după apariția unui eveniment structura Event Structure se consumă (termină) și trebuie din nou executată pentru a trata un următor eveniment.

Producerea evenimentului dynamic: când se generează un număr aleator mai mare decât 0.80 se selectează cazul True (Instr. Case) în care are loc apelul funcției **Generate User Event**. Generarea de numere aleatoare are loc repetitiv cu temporizare de 1000 ms. Se rămâne în ciclul While până la apăsare Buton 'stop gen si VI' situație în care se oprește și aplicația.

Generarea periodică de numere aleatoare se întrerupe numai cu buton STOP din Panoul frontal a cărui Mechanical Action este pusă pe poziția (1,1) în matricea de acțiuni posibile (Fig.). Numai pe acest mod se poate crea variabilă locală pe butonul de STOP și pune variabila locală în cazul al 3-lea pentru oprire ciclul While pe Stop if true când se apasă STOP. Se apasă STOP care generează True după care se apasă din nou STOP pentru revenire la buton neapăsător pentru a genera False și a nu se închide aplicația la un nou Run.



6. Filtru evenimente (Filter Event). Anumite evenimente pot fi anulate automat sau consultat operatorul înaintea efectuării acțiunii.

Panel Close? și Menu Activation?

A apărut un eveniment și anume Click pentru a închide panou frontal (PF). Acțiunea de închidere efectivă a ferestrei poate fi anulată (dacă rulează o aplicație de exemplu) prin tratarea evenimentului [0] **Panel Close?**. Acțiunile pot fi anulate sau modificate.

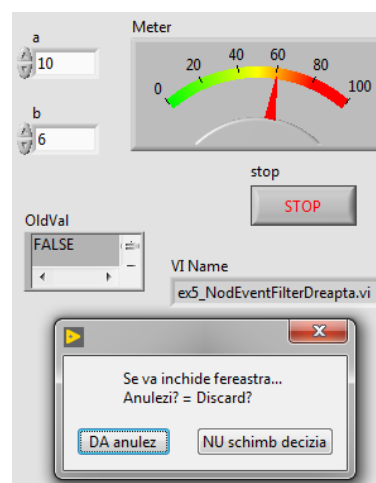
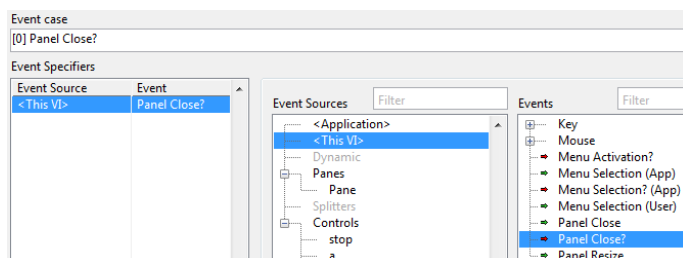
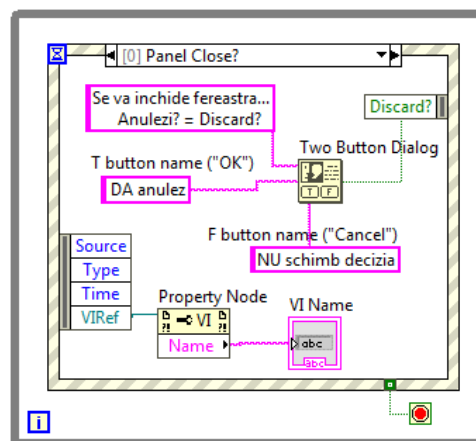
[0] **Panel Close?** -> filtrare evenimente

1. Cu mouse se acționează pentru închiderea ferestrei PF a aplicației *nume.vi*.

2. Prin **VIREf** din Nodul Event Data se extrage numele aplicației curente *nume.vi*. Numele aplicației (a cărui PF a fost acționat pentru închidere) se afișează în indicatorul șir caractere VI Name.

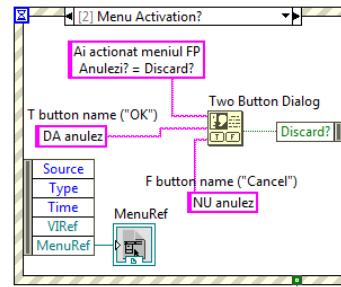
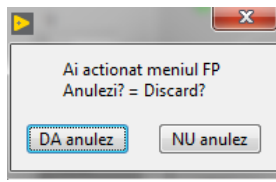
3. Prin câmpul **Discard?** al nodului Event Filter se poate anula comanda de închidere a ferestrei dacă se alege 'Da anulez' sau se acceptă decizia de închidere aplicație dacă se alege 'NU schimb decizia'. Câmpul **Discard?** apare când se selectează evenimentul **Panel Close?** în fereastra de configurare.

Tratarea acestui Event case se observă în fereastra de tratare unde observăm că sursa evenimentului este **<This VI>** iar tipul evenimentului este **Panel Close?**.



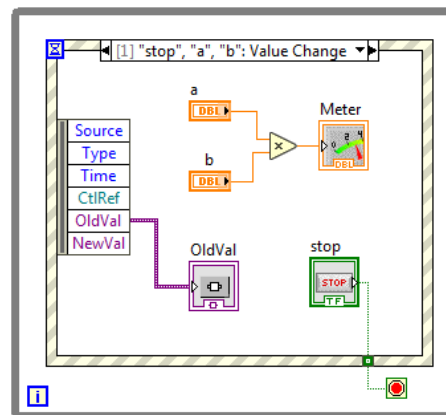
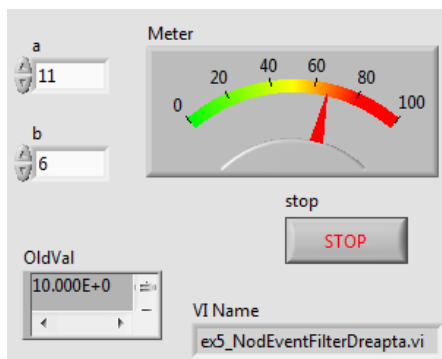
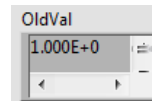
[2] Filter Event: Menu Activation?

În mod similar se crează cazul de tratare a activării meniului panoului frontal PF pentru a selecta o comandă în timpul rulării programului. Se poate anula sau continua acțiunea demarată răspunzând la fereastra dialog care apare.



[1] "stop", "a", "b": Value Change -> eveniment de notificare

1. Se modifică valoarea controlului a sau b sau stop. Prin articolul **OldVal** din nodul Event Data se afișează valoarea veche a controlului modificat. Indicatorul OldVal este de tip variant pentru că va afișa atât valori numerice (a,b) cât și logice (False).



7. Folosiți Event Structure pentru urmărire a unui eveniment de tip mișcare mouse în PF, conform cu figura alăturată.

Sunt returnate (în case) [0] "Pane": Mouse Move coordonatele mouse-ului în timp ce-l deplasăm. După oprirea ciclului While să se traseze mișcarea avută de mouse. Pe un alt caz (case) să se anuleze automat încercarea de a închide sau redimensiona panoul frontal.

