MEDIUL LABVIEW. CASETA FUNCŢII

1. Scopul lucrării

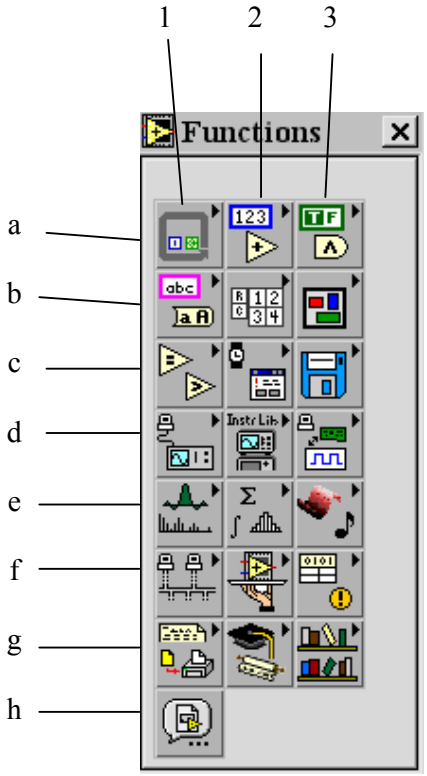
Lucrarea are drept scop prezentarea elementelor limbajului G care trebuie cunoscute pentru realizarea unui *IV* în mediul LabView.

2. Consideraţii teoretice

După realizarea panoului frontal al *IV* , trebuie implementată funcţionalitatea programului: se construieşte diagrama bloc care reprezintă codul sursă al instrumentului adică arată ***CUM*** se rezolvă problema. În acest scop se utilizează limbajul grafic G. Utilizatorul selectează şi utilizează componente grafice de execuţie definind astfel funcţionalitatea *IV*.

Elementele utilizate pentru realizarea diagramei bloc sunt clasificate în trei grupe: *noduri, terminale şi fire.*

- *Nodurile* sunt elementele de execuţie ale unui *IV*. Acestea sunt disponible prin ***caseta cu funcţii ale IV*** (fig.1).



Elementele componente ale casetei cu funcţii sunt:

a1- instrucţiuni pentru controlul execuţiei programelor, formula de calcul, variabilă locală şi globală;

a2 – funcţii aritmetice, trigonometrice, logaritmice etc.;

a3 – funcţii logice;

b1 – funcţii şi constante pentru tipul şir de caractere;

b2 – funcţii şi constante de tip tablou;

b3 – funcţii şi grup de date (cluster);

c1 – funcţii de comparare;

c2 – funcţii de timp;

c3 – funcţii şi *IV* pentru gestionarea de fişiere;

d1 – *IV* pentru comunicaţii cu instrumente GPIB, VISA sau serial;

d2 – *IV* pentru GPIB, VISA, instrumente de măsură;

d3 – *IV* pentru achiziţia datelor;

e1 – *IV* pentru analiza datelor, generare de semnal etc.;

e2 – *IV* pentru calcule în exemple de simulare;

e3 – funcţii pentru apelarea procedurilor scrise în C, pentru manevrarea datelor etc;

f1 – *IV* pentru comunicări în reţea;

f2 – caseta de controlul aplicaţiilor include funcţii Help, Menu, Print etc.;

f3 – funcţii avansate;

g1 – generare raport;

g2 – tutorial;

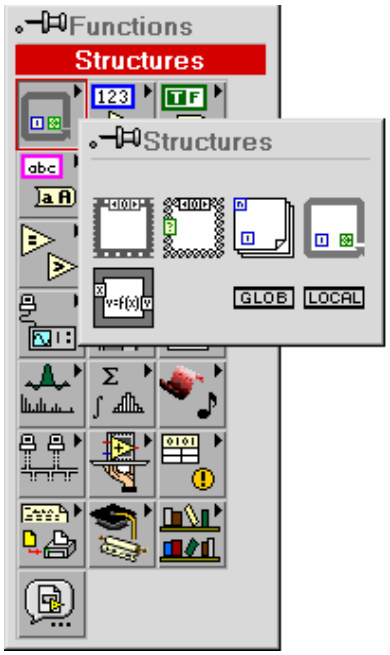


Fig. 2

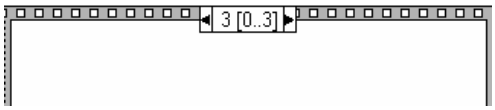


Fig.3

g3 – bibliotecă de *IV* salvate în …\ *LabView\User.Lib ;*
h1 – selectare *IV;*
Există 4 tipuri de instrucţiuni pentru controlul execuţiei programului (Fig.2):

a) instrucţia Secvenţială - Sequence;
b) instrucţia de selecţie multiplă – Case;
c) instrucţiunea repetitivă For – For Loop;
d) Instrucţiunea repetitivă While – While Loop.

Pe cea de-a doua linie din casetă se găseşte opţiunea pentru formule de calcul.

Instrucţiunea secvenţială permite să se impună ordinea de execuţie a unor subdiagrame, între care nu există dependenţa datelor. Instrucţiunea este formată din una sau mai multe subdiagrame, fiecare fiind susţinută de un cadru. Cadrele sunt suprapuse şi prin numărul pe care îl reprezintă consemnează succseiunea de execuţie. O formă generală este prezentată în figura 3. Sunt patru subdiagrame (0, 1…3). Introducerea unei subdiagrame se poate realiza relativ la una existentă: în faţă sau după.

Instrucţiunea de selecţie multiplă permite execuţia unei singure instrucţiuni, din mai multe alternative, pe baza valorii unei expresii.

Instrucţiunea repetitivă For asigură reluarea instrucţiunilor, care formează corpul ciclului, de un anumit număr de ori.

Instrucţiunea repetitivă While condiţionează execuţia instrucţiunii de valoarea logică a expresiei de oprire. Pentru valoarea logică "Adevărat" se reia execuţia iar pentru "False" execuţia se opreşte.

Ultima opţiune - formula de calcul – dă posibilitatea scrierii în mod text a unei formule de calcul.

- *Terminalele* reprezintă "porţi" (tunele) prin care se realizează transferul datelor:

a) bidirecţional între panoul frontal şi diagrama bloc;
b) unidirecţional între nodurile diagramei bloc.

Terminalele au o reprezentare grafică sugestivă şi sunt terminale sursă – pentru datele de intrare – şi respectiv terminale destinaţie (ieşire).

- *Firele* definesc şi reprezintă grafic fluxul datelor în diagrama bloc.

Fluxul datelor este de la terminalele sursă spre terminalele destinaţie. Prin culoarea şi tipul liniei, firele codifică tipul datelor transmise.

3. Mersul lucrării

Pentru exemplificare considerăm o aplicaţie de generare aleatorie a unui semnal, concomitent cu vizualizarea acestuia.

Panoul frontal conţine un buton de pornit / oprit conectat la o lampă de semnalizare şi un instrument de vizualizare (fig.4)

Diagrma bloc divide aplicaţia în două secvenţe succesive numerotate “0” (fig.5) şi “1” (fig.6).

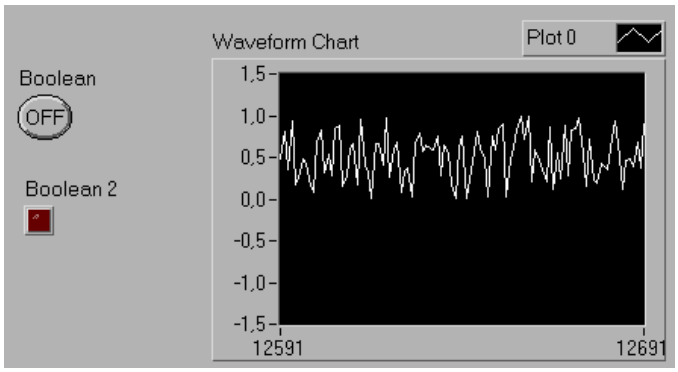


Fig.4

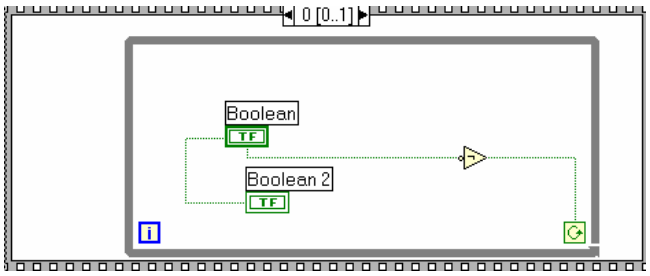


Fig.5

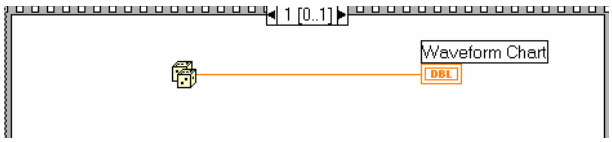


Fig.6

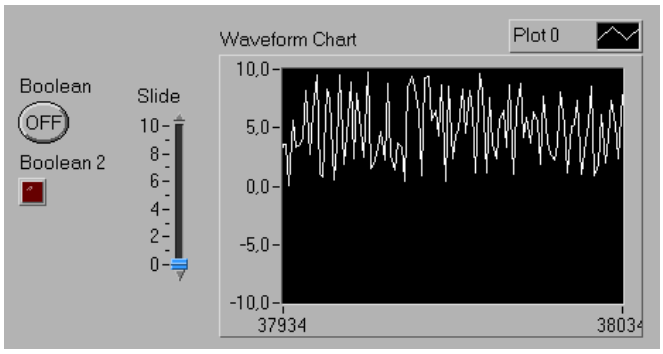


Fig.7

Realizaţi aplicaţia şi consemnaţi concluziile privind funcţionarea. Modificaţi etichetele şi indicaţi astfel rolul fiecărei componente. Personalizaţi indicaţia vizuală.

Reluaţi aplicaţia prin introducerea unui buton de reglaj vertical (fig.7). Se compară valoarea acestuia cu constanta “5” şi în funcţie de valoarea logică – adevărat sau fals – se generează semnalul aleatoriu în intervalul 0-1 sau 0-10. Diagrama bloc - în două secvenţe – este prezentată pentru secvenţa “1” în figura 8

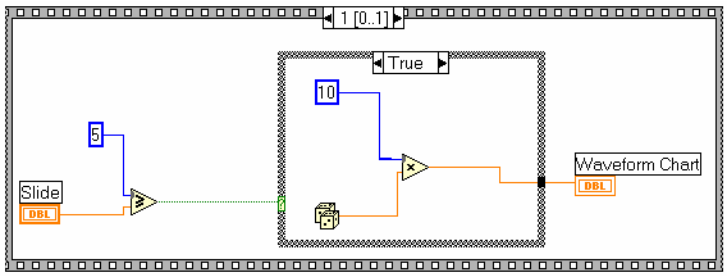


Fig.8

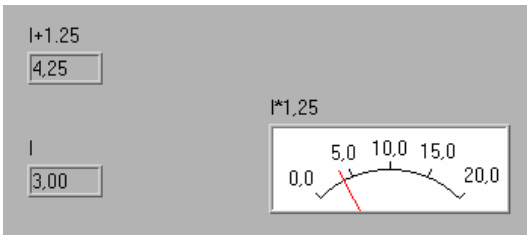


Fig.9

Consemnaţi care dintre instruţiuni pentru controlul programului se folosesc şi modul de lucru. Modificaţi valorile şi stabiliţi noul mod de lucru

Considerăm o aplicaţie într-un ciclu repetitiv FOR de adunare şi înmulţire:

$numrul = 1.25 + i$

$numrul = 1.25 \cdot i$

În cadrul aplicaţiei se doreşte vizualizarea pas cu pas a operaţiilor, valoarea lui "i" şi a rezultatului. Panoul frontal şi diagrama bloc sunt reprezentate în figura 9 şi 10.

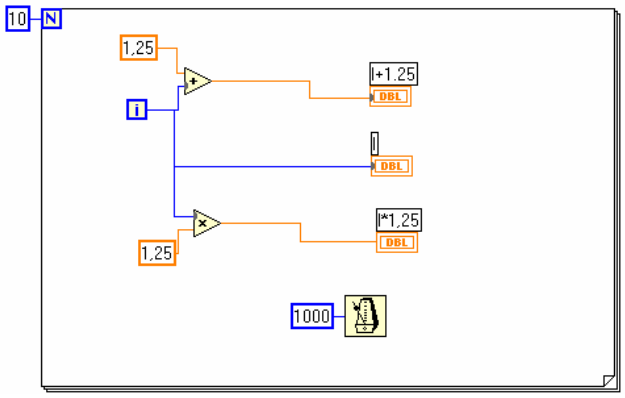


Fig.10

Consemnaţi observaţiile şi imaginaţi aplicaţii asemănătoare. Consemnaţile în referatul laboratorului

Reluaţi aplicaţia de generare a unui semnal (fig.11). realizaţi aplicaţia prin utilizarea instrucţiunii While-Loop într-o singură secvenţă (fig.12).

Consemnaţi concluziile şi observaţiile, privind instrucţiunea de control, în referatul lucrării.

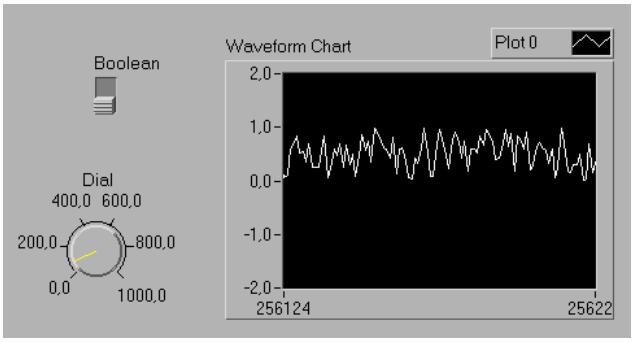


Fig.11

Realizaţi o paralelă între cele patru instrucţiuni de control al unui program şi consemnaţi obsevaţiile finale în referat.

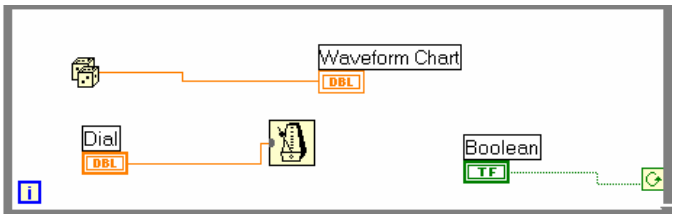


Fig.12

Enunţaţi câte o aplicaţie pentru cazurile anterioare de control a unui program, realizaţi panoul frontal şi diagramele corespunzătoare. Consemnaţi în referat concluziile corespunzătoare.

Structurile repetitive permit executarea în ciclu a uneia sau mai multor linii de cod (instrucţiuni). Distingem în Visual Basic:

- Do ……Loop
- For ……Next

```
Dim Nr1, Nr2 As Integer
Do
        Do
                Nr2=Nr2+1
                Print "Nr2=";Nr2
        Loop While (Nr2<10)
        Nr1=Nr2+1
        Print "Nr1=";Nr1
Loop Until (Nr1<10)
```

```
For I = 1 to 10 step 2
        For j = 1 to 5 step 3
                C = I+J
        Next j
Next I
```

While indică faptul că bucla este executată când condiţia este adevărată iar *Loop* până când condiţia este adevărată

Structurile de decizie testează condiţia instrucţiunii şi, în funcţie de rezultatul testului, execută diferite operaţii. Structuri de decizie disponibile în Visual Basic:

- If….Then
- If ….Then….Else
- Select Case

```
If Nr2 >9  Then
Form1.Print "Popescu este  admis"
End If
```

```
If  Nr2 < 5 Then
Form1.Print "Popescu este respins"
ElseIf Nr2 > 5 Then Form1.Print "Popescu
este admis"
ElseIf Nr2 > 8 Then Form1.Print "Popescu
este bursier"
End If
```

Realizaţi în VisualBasic programe asemănătoare, scrieţi şi rulaţi acelaşi program în LabView. Consemnaţi concluziile în referat.