

Aplicația II.7

SALVAREA PE MEDIUL EXTERN. CREAREA APLICAȚIILOR DE TIP STAND-ALONE (.exe)

7.1. Scopul lucrării

Versiunile recente de **LabVIEW** au fost dezvoltate și în ideea monitorizării și supervizării proceselor tehnologice, folosind echipamente de firmă pentru care există driver-e de compatibilizare cu acesta. Aceste facilități permit realizarea de sisteme **SCADA** (**Supervisory Control And Data Acquisition**), variante frecvent întâlnite în automatizările moderne.

În consecință, promotorii mediului **LabVIEW** au realizat facilități de salvare a datelor într-o manieră ușoară pe medii externe pentru a avea un istoric al evoluției unor parametri investigați, precum și posibilitatea realizării de aplicații executabile, care pot lucra fără a folosi programul **LabVIEW** propriu-zis.

Scopul acestei lucrări este, pe de o parte, de a prezenta aplicații care să demonstreze maniera de salvare automată și la cerere a datelor într-un fișier de tip Microsoft Excel, iar pe de altă parte, de a arăta modalitatea de obținere a unei aplicații stand-alone (tip **.exe**), capabilă de a rula pe alt calculator unde nu este instalat **LabVIEW**.

7.2. Salvarea automată în fișier de tip Microsoft Excel

Înainte de a exemplifica modalitatea de salvare a datelor, este necesară realizarea unui instrument virtual capabil să genereze date, într-o manieră cât mai intuitivă pentru utilizator. În acest sens s-a considerat că semnalul sinusoidal este bine cunoscut și ușor de interpretat, consecința fiind că primul pas care se realizează constă în generarea acestuia.

Așa cum s-a prezentat în **aplicația II.4**, există în **LabVIEW** – în principiu – două posibilități de generare a semnalului sinusoidal:

- prin utilizarea generatorului de semnal din paleta **Functions**, meniul **Analyze >> Waveform Generation >> Sine Waveform** introdus în diagrama aplicației; acest obiect va genera un semnal de tip **cluster**, din care vor trebui extrase datele utile într-un **array** (detalii în secțiunea **4.3.2** din **aplicația II.4**);

- prin folosirea modulelor dedicate din paleta **Functions >> Numeric >> Trigonometric**, modalitate descrisă în detaliu în secțiunea **4.4.1** din **aplicația II.4**.

În cazul concret al aplicației de față s-a preferat a doua modalitate, întrucât datele generate sunt de tip **array**, fără a mai fi necesară prelucrarea lor ulterioară.

Urmând pașii descriși în detaliu în secțiunea 4.4.1 din **aplicația II.4**, se construiește generatorul de sinusoidă având diagrama bloc din fig.7.1, cu următoarele

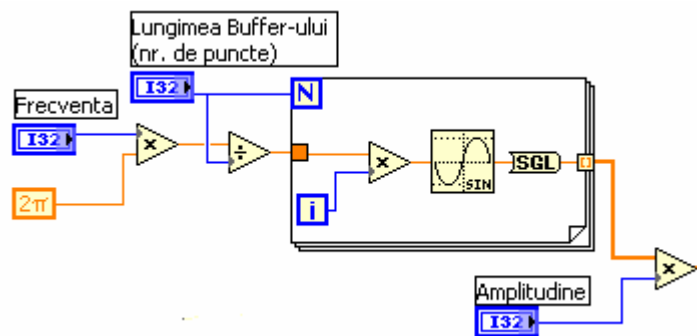


Fig.7.1. Generarea unui semnal sinusoidal

precizări:

- numărul de puncte prin care se reprezintă semnalul este variabil, fiind dat de controlul **Lungimea Buffer-ului (nr. de puncte)** prevăzut pe panoul frontal;
- atât frecvența cât și amplitudinea semnalului sunt variabile, cu posibilitatea modificării din controalele **Frecvența** și **Amplitudine** plasate – de asemenea – pe

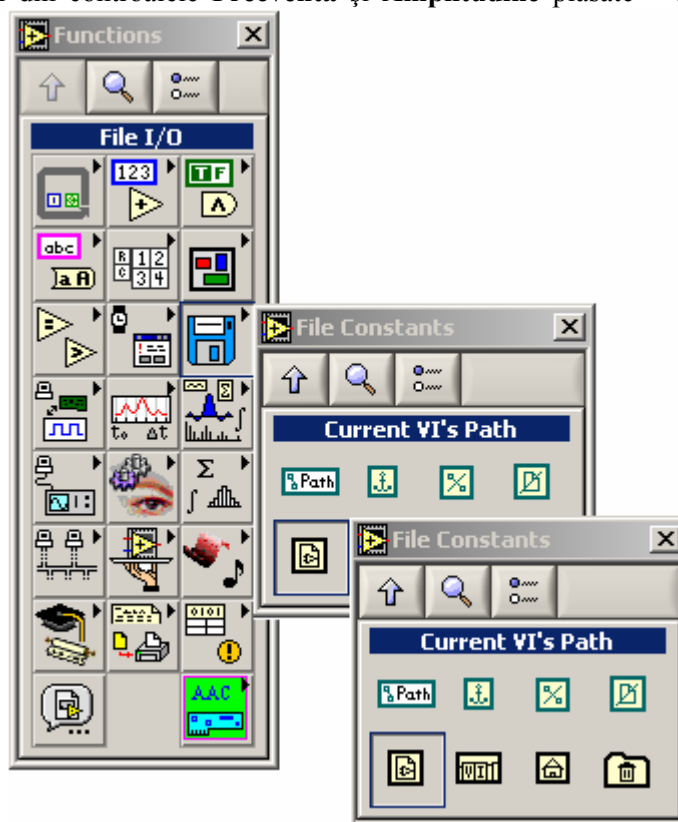


Fig.7.2. Constanta **Current VI's Path**

panoul frontal;

- rezultatul obținut la ieșirea funcției **Sine** este convertit la tipul single-precision folosind primitiva **To Single Precision Float** adusă din paleta **Functions >> Numeric >> Trigonometric >> Conversion**, iar graficul obținut va fi afișat pe un indicator grafic de tip **Waveform Chart**.

Lucrul cu fișiere în mediul de dezvoltare **LabVIEW** este destul de facil, în continuare exemplificându-se modalitatea de salvare a datelor generate de aplicația anterior dezvoltată. Se va lucra, în principal, cu module din meniul **File I/O** din paleta **Functions**.

Pentru aceasta, în primul rând trebuie să se realizeze calea către fișierul în care se dorește salvarea datelor. **LabVIEW** pune la dispoziția utilizatorului constante de tip fișier, una dintre acestea fiind **Current VI's Path**, care furnizează calea către fișierul **VI** în care se lucrează, așa cum este reprezentarea din fig.7.2.

De menționat că această cale, de exemplu de forma **C:\My Documents\STV Lucrare.vi**, conține și numele fișierului. Se dorește salvarea datelor într-un fișier de tip **.xls**, care să fie localizat tot în directorul

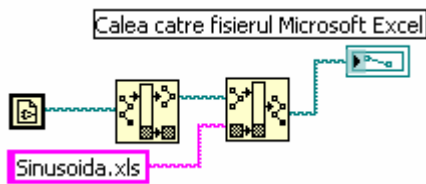


Fig.7.3. Diagrama pentru formarea path-ului către fișierul Microsoft Excel

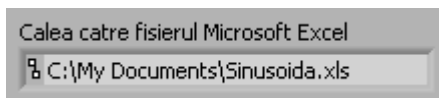


Fig.7.4. Calea către fișierul Microsoft Excel

curent. De aceea se aplică constanta sus-menționată la funcția **Strip Path**, adusă din meniul **File I/O**, care va genera doar calea către directorul curent. Rezultatului obținut i se adaugă, prin intermediul funcției **Build Path** - adus din meniul **File I/O**, numele fișierului în care se dorește salvarea datelor, astfel că se obține, la ieșirea funcției **Build Path**, calea către fișierul **.xls**, așa cum se poate observa în fig.7.3 și fig.7.4 (diagrama bloc și rezultatul în panoul frontal).

În acest moment s-a creat fișierul în care vor fi scrise datele. Se alege, din meniul **File I/O**, funcția **Write To Spreadsheet File**, reprezentată în fig.7.5, care acceptă ca date de intrare vectori de tip 1D sau 2D, calea către fișier și formatul datelor. În plus, se dispune de două switch-uri foarte importante, care vor fi folosite în continuare, și anume switch-ul "Transpose" care va insera în fișier transpusa matricii de intrare, precum și "Append" care are ca efect suprascrierea fișierului la fiecare inserare de date sau adaugarea datelor în fișiere.

Pentru partea de *salvare automată*, va trebui să se adauge fiecare buffer de date în fișier, deci se va seta switch-ul pe **True** cu ajutorul unei constante boolene. De

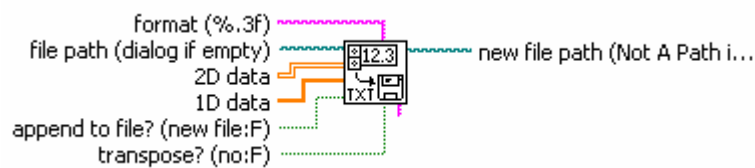


Fig.7.5. Funcția **Write To Spreadsheet File**

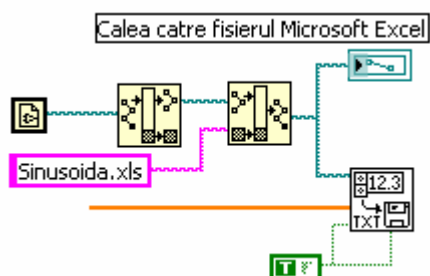


Fig.7.6. Diagrama pentru scriere în fișierul Microsoft Excel

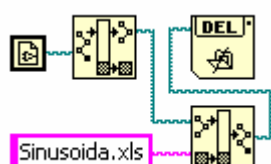


Fig.7.7. Ștergerea fișierului la intrarea în aplicație

asemenea, se dorește a avea datele pe coloane, așa încât se va scrie în fișier transpusa matricei de intrare. Schema rezultată va arăta ca în fig.7.6.

În acest moment datele generate prin diagrama sinusoidă (firul portocaliu) vor fi scrise în fișierul Microsoft Excel.

Dacă se analizează diagrama din fig.7.6, se observă că datele vor fi adăugate în fișier la nesfârșit. De asemenea, când se oprește aplicația, se iese din **LabVIEW**, apoi se reintră rulându-se aplicația, datele vor fi

adăugate la cele vechi, lucru care nu este întotdeauna de dorit. Pentru a salva doar datele rezultate în urma unei rulări a aplicației, și totodată a nu păstra istoricul lor, se recurge la următorul artificiu:

- la începutul programului se crează o diagramă pentru ștergerea fișierului vechi;
- restul diagramei (cea care generează sinusoida, cea de salvare etc) se va insera într-un ciclu **While Loop**, în acest fel garantându-se că fișierul de salvare date va fi

șters o singură dată la începutul programului.

În consecință, pentru a șterge fișierul de date, se aplică aceeași metodă de a construi calea prezentată mai sus, iar rezultatul va fi conectat la un modul de tip **Delete**, adus în diagramă din meniul **File I/O >> Advanced File Functions** (fig.7.7).

NOTĂ: Trebuie menționat că, în cazul în care nu se găsește fișierul de șters, **LabVIEW** nu generează eroare. Acest lucru permite rularea aplicației prima dată fără a crea manual un fișier Microsoft Excel, acesta fiind generat automat la prima rulare cu numele **Sinusoida.xls**.

7.3. Salvarea la cerere într-un fișier Microsoft Excel

Noțiunea de salvare la cerere presupune salvarea unui buffer al sinusoidă (specificat prin numărul de puncte fixat pe controlul de pe panoul frontal), atunci când se apasă un buton din interfața cu utilizatorul. Pentru aceasta se prevede un buton de tip **Stop** din meniul **Boolean** în interfața cu utilizatorul, care va fi denumit sugestiv **Salveaza**. După care, cu click dreapta pe acesta, se selectează **Mechanical Action** și apoi **Latch When Pressed**, așa cum se arată în fig.7.8.

Se revine în diagramă și se adaugă o structură de tip **Case**, care va fi "comandată" de către butonul **Salveaza**. Astfel, pe ramura **True**, când butonul a fost apăsat, se generează fișierul **Sinusoida_la_cerere.xls**, conform procedurii de mai sus. Acest fișier va fi șters de fiecare dată când se dă comanda **Salveaza**, întrucât nu se dorește păstrarea datelor vechi. Schema va arăta ca în fig.7.9, cu mențiunea că pe ramura **False** a structurii **Sequence** nu se prevede nici un element. De asemenea,

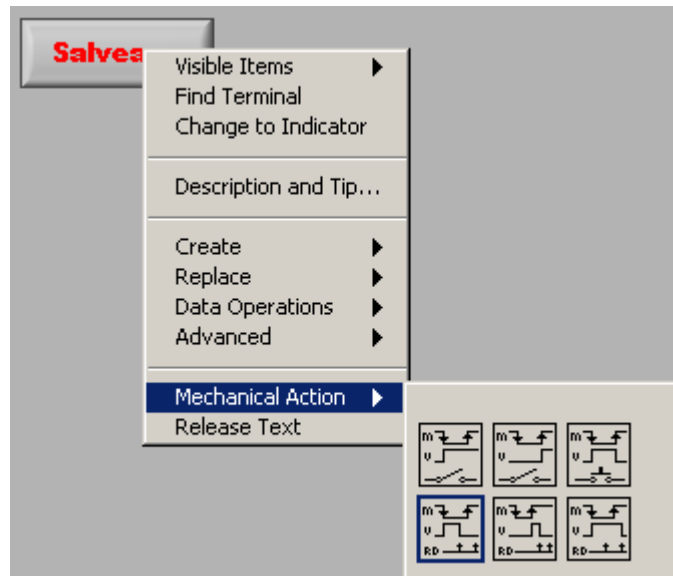


Fig.7.8. Schimbarea acțiunii la apăsarea butonului

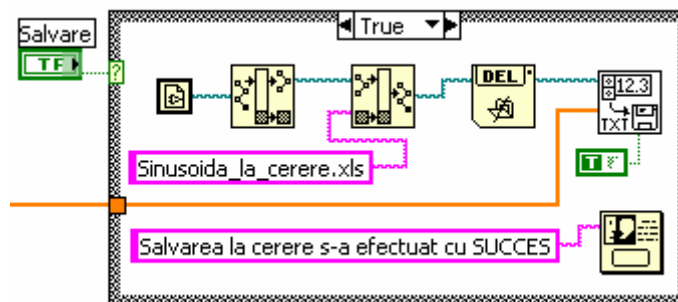


Fig.7.9. Salvarea la cerere

pentru a informa utilizatorul că operația a fost reușită, se introduce o fereastră **One Button Dialog** din meniul **Time&Dialog** din paleta **Functions**, la care se aduce un **String Constant** cu mesajul **Salvarea la cerere s-a efectuat cu SUCCES**.

Configurația finală a aplicației, care reunește atât *salvarea automată* cât și *salvarea la cerere*, este prezentată în fig.7.10 (panoul frontal), respectiv fig.7.11 (diagrama bloc). Se remarcă, în diagrama bloc, prezența unui element **Wait Until Next ms Multiple**, comandat de un **Vertical Pointer Slide** plasat pe panoul frontal, elemente necesare pentru a putea reface corect sinusoida folosind mediul Excel.

7.4. Crearea unei aplicații de tip stand-alone (.exe)

Un aspect important în crearea pachetelor de aplicații, la fel ca și în programarea textuală, este reprezentată de distribuția software. În acest sens, **LabVIEW** pune la dispoziția utilizatorilor o modalitate prin care se pot crea aplicații

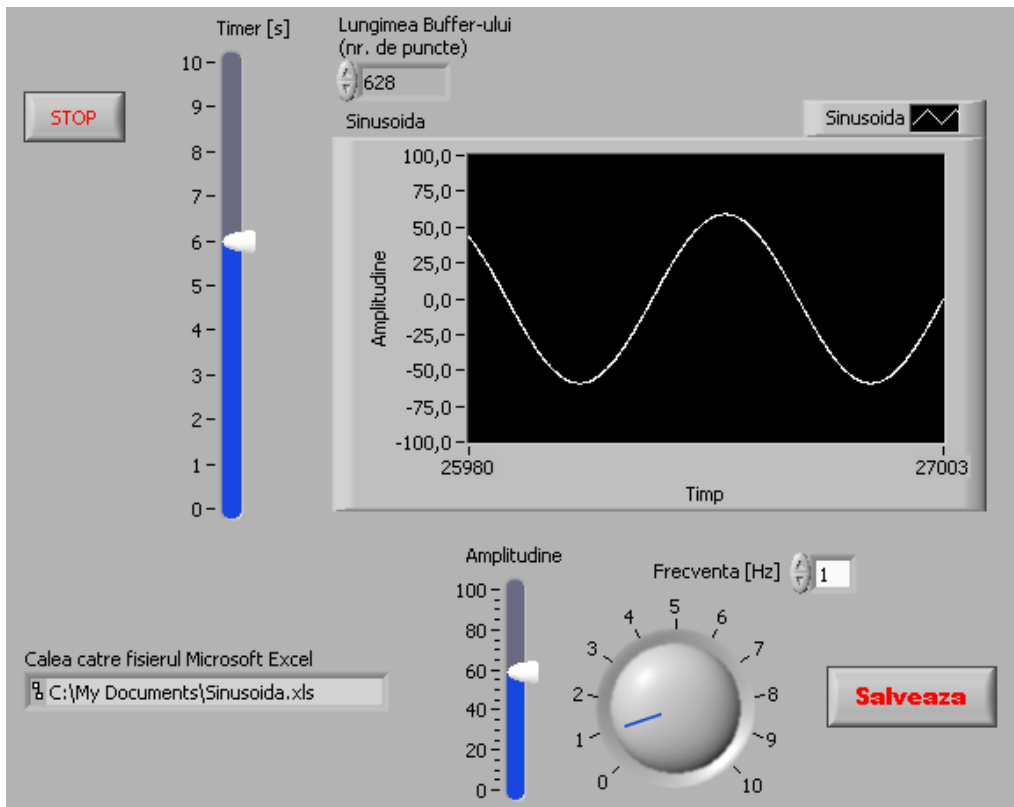


Fig.7.10. Panoul frontal final al aplicației

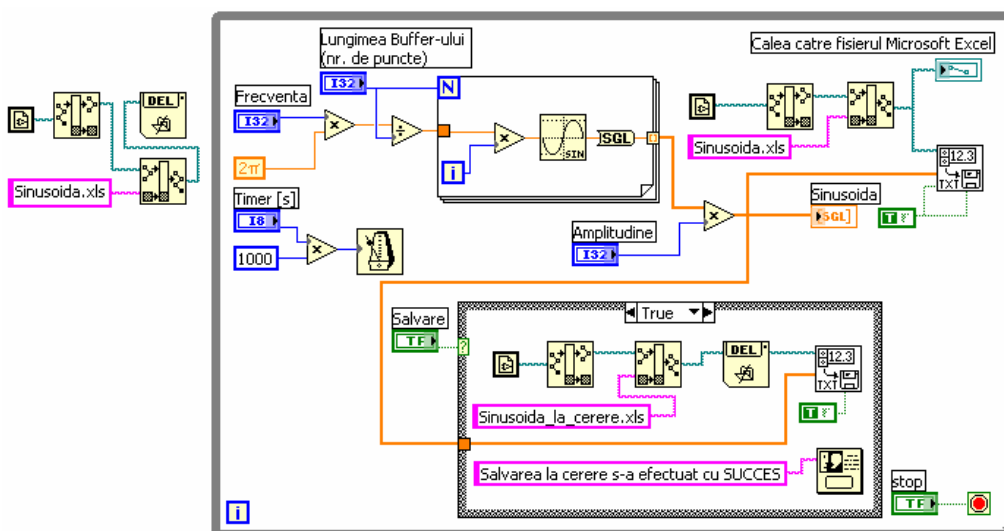


Fig.7.11. Diagrama finală a aplicației

stand-alone, de tip **.exe**, care însă au nevoie de **LabVIEW Runtime Environment** pentru a rula (similar cu **JRE- Java Runtime Environment**) Acest **LabVIEW**

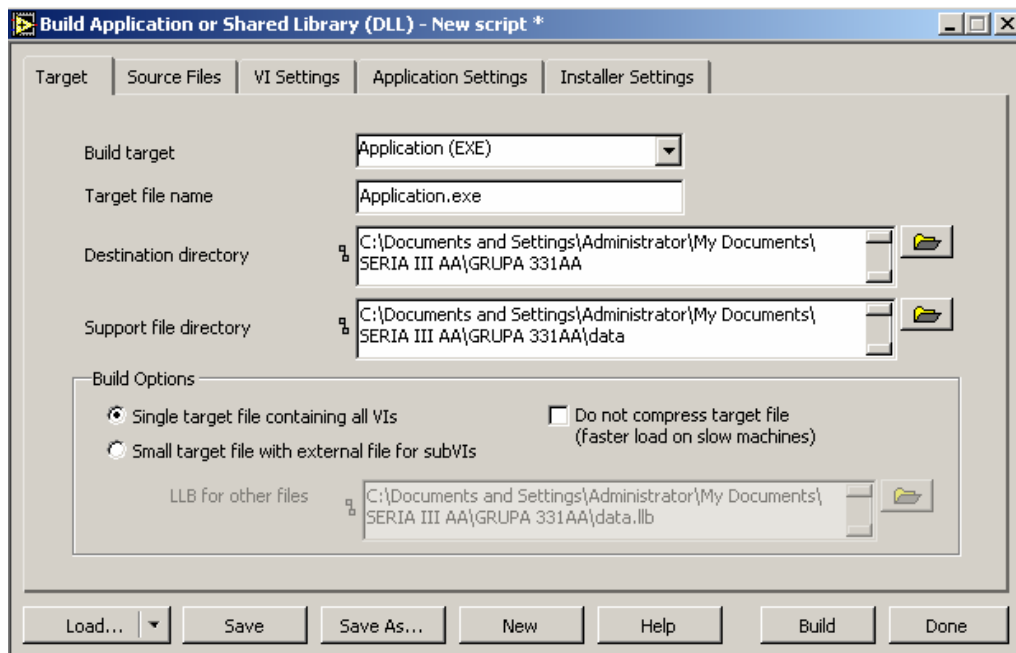


Fig.7.12. Fereastra **Build Application**

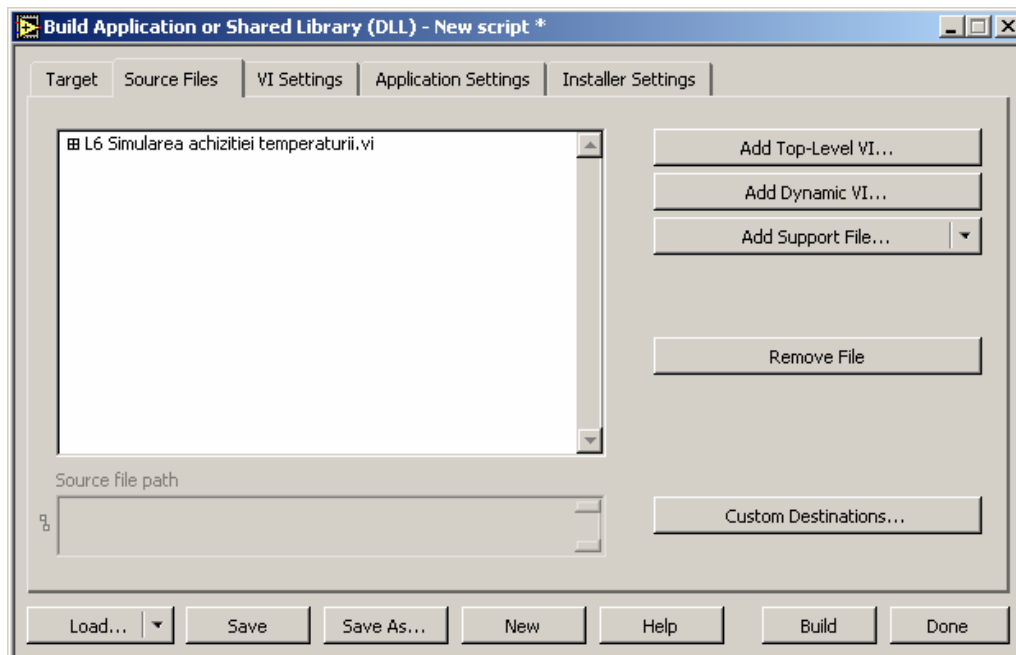


Fig.7.13. Selecția **VI**-ului ce va deveni stand-alone (**.exe**)

Runtime Environment este oferit gratuit de către National Instruments, deci un client nu are nevoie de licență pentru **LabVIEW**.

Pentru a crea o aplicație executabilă, se lansează în execuție **LabVIEW 6.1, New VI**, se accesează, din meniul **Tools** al **LabVIEW**-ului, opțiunea **Build Application or Shared Library (DLL)...** și se va deschide o fereastră similară cu cea din fig.7.12.

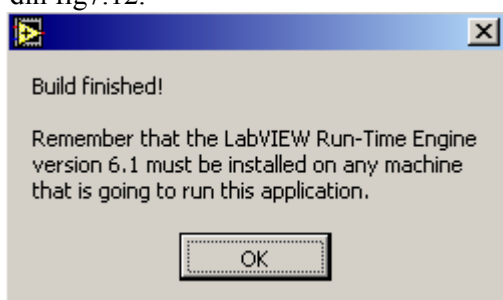


Fig.7.14. Fereastra de finalizare a operației de compilare

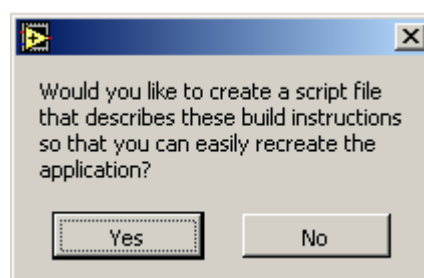


Fig.7.15. Fereastra de creare a unui fișier script (.BLD) pentru aplicația executabilă

Se completează câmpurile referitoare la numele aplicației, directorul unde se dorește salvarea acesteia, eventual alt director pentru **SubVI**-uri dacă este cazul.

În continuare se selectează tab-ul **Source Files** și, în acesta, butonul **Add Top-Level VI...** Se va deschide o fereastră de dialog de unde se selectează **VI**-ul care se dorește a fi făcut executabil. După selecție, numele **VI**-ului va apărea în fereastră, așa cum este exemplificat în fig.7.13. În acest moment se dă click pe butonul **Build** și va începe procesul de compilare.

La sfârșit, utilizatorul va fi informat că are nevoie de **LabVIEW Run-time Engine**, ca în fereastra de dialog reprezentată în fig.7.14.

După comanda **OK** în această fereastră, se va regăsi aplicația executabilă în directorul selectat pentru crearea acesteia. La părăsirea procesului de realizare a aplicației executabile, prin acționarea butonului **Done**, va apare fereastra de dialog exemplificată în fig.7.15, prin care utilizatorul va putea construi un fișier **.BLD** folosit pentru descrierea aplicației sau pentru recreerea ușoară a acesteia.

În cadrul exemplului considerat, s-a construit în directorul **C:\...\My Documents\SERIA III AA\GRUPA 331AA** fișierul executabil aferent **Aplicației II.6** cu numele **Simularea achizitiei temperaturii**. La lansarea – prin dublu click – a acestei aplicații executabile, va apare panoul frontal din fig.7.16, care se deosebește de panoul exemplificat în fig.6.1 – **Aplicația II.6**, în sensul că prima bară conține meniuri cu opțiuni diferite de cele din **LabVIEW** descrise în **aplicația II.1** – secțiunea **1.3**, iar a doua bară conține numai comenzile **Run** și **Run Continuously**. De asemenea, nu este posibilă modificarea aplicației și nici afișarea diagramei bloc a acesteia.

NOTĂ. Rularea aplicației presupune a avea pe calculatorul unde se poartă (care nu dispune de mediul **LabVIEW**) versiunea necesară – conform ferestrei de dialog din fig.7.14 – de **LabVIEW Run-Time Engine**, în consecință se procedează la

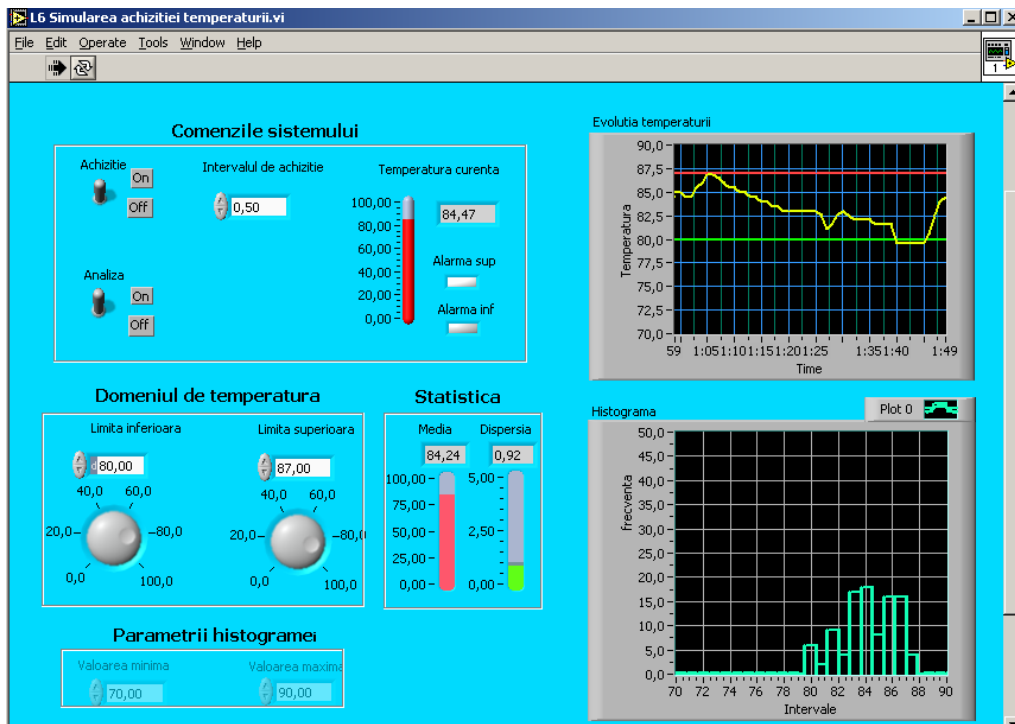


Fig.7.16. Rularea aplicației executabile **Simularea achizitiei temperaturii**

descărcarea acestui program, fie de pe site-ul www.ni.com, fie de pe CD-ul cu kit-ul **LabVIEW 6.1**, și instalarea acestuia pe calculatorul unde se dorește rularea aplicației executabile. Dacă apar probleme de rulare a aplicației executabile, acestea pot fi rezolvate prin consultarea FAQ-urilor de pe site-ul www.ni.com.

7.5. Chestiuni de studiat

1. Studiarea elementelor componente și a funcțiilor pe care acestea le realizează, pentru ca instrumentul virtual **Salvare_mediu extern.vi** să îndeplinească cerințele impuse potrivit scopului enunțat.

2. Rularea programului și observarea funcționării VI-ului în ansamblu, a execuției subdiagramelor structurilor de generare, de salvare și de salvare la cerere, precum și a modului de afișare și de înregistrare a rezultatelor în fișierele create.

3. Deducerea relației între lungimea buffer-ului, frecvență și numărul de perioade afișate, respectiv salvate, ale sinusoidei generate. Verificarea experimentală a relației stabilite, reaspectiv cum trebuie ales raportul f/N pentru ca între valorile salvate să se afle și punctele de trecere prin **0** ale sinusoidei.

4. Observarea rolului controalelor situate în panoul frontal și a constantelor cuprinse în diagrama bloc și optimizarea acestora în raport de cerințele de salvare, ținând cont de relația de la punctul 3.

5. Studiarea rolului și funcționării subdiagramei situate în exteriorul buclei **While Loop** pentru ștergerea inițială a fișierului de salvare, precum și a subdiagramei inclusă în bucla **Case** pentru ștergerea fișierului destinat salvării la cerere.

6. Crearea unui **VI** similar și aplicarea acestuia și pentru alte semnale generate în cadrul aplicațiilor precedente, de exemplu pentru variația temperaturii din **Aplicația II.6**.

7. Referitor la crearea unei aplicații de tip **stand-alone (.exe)**, se va studia procedura de realizare a unei astfel de aplicații executabile așa cum este descrisă în secțiunea 7.4.

8. Se vor face observații și comentarii cu privire la lansarea și rularea unei aplicații executabile, crearea fișierului executabil, condițiile necesare pentru rularea pe un calculator pe care nu este instalat mediul **LabVIEW**, care sunt opțiunile pentru rulare și pentru eventuale modificări ale acesteia.

7.6. Modul de lucru și prezentarea rezultatelor

Pentru punctele 1 și 2 de la **Chestiuni de studiat** se va deschide și se va experimenta instrumentul virtual **Salvare mediu extern.vi**, identificându-se obiectele dispuse în panoul frontal și în diagrama bloc, interconectările dintre ele și funcțiile pe care le realizează, în conformitate cu precizările din secțiunile prezentei lucrări. Se va rula programul de funcționare a **VI**-ului, folosindu-se, într-o primă etapă, setările din figurile 7.10 și 7.11.

Pentru realizarea obiectivelor de la punctele 3 și 4 se va proceda la modificarea valorilor atribuite controalelor din panoul frontal referitoare la procesele de generare a sinusoidei și de salvare, după cum urmează:

- pentru controlul **lungimea buffer-ului**, nr. de puncte **N**, se vor lua valorile 120; 240 ; 300; 1000;

- pentru **frecvența f**, valorile vor fi 10; 20; 40 ;55;

Se vor determina numărul de perioade afișate și salvate, precum și intervalul în radiani (exprimat în fracțiuni din π), între două puncte. Calculând raportul **f/N** și urmărind detaliile din **Aplicația II.4**, secțiunea 4.4.1, se va putea deduce relația cerută la punctul 3.

Pentru punctul 5, într-unul dintre regimurile de funcționare experimentate, se va acționa butonul **Highlight Execution**, se va urmări în diagrama bloc fluxul de date observând subdiagrama din exteriorul buclei **While Loop** și apoi se vor studia înregistrările pentru mai multe salvări succesive. La fel se va proceda și pentru salvarea la cerere.

În ceea ce privește punctul 6, folosind elementele din **aplicația II.6** și având modelul din prezenta aplicație, se va crea **VI**-ul solicitat.

Pentru rezolvarea cerințelor de la punctele 7 și 8, urmând indicațiile din secțiunea 7.4, se va realiza aplicația executabilă pentru exemplul considerat, **Simularea achiziției temperaturii din aplicația II.6**. Observând pașii parcurși pentru lansarea, crearea și rularea aplicației, inspectând panoul frontal, comparându-l cu unul corespunzător unei aplicații **LabVIEW** uzuale și încercând intervenții asupra obiectelor dispuse pe acesta sau în diagrama bloc, se vor găsi răspunsurile solicitate.

În ceea ce privește prezentarea rezultatelor, aceasta va cuprinde enunțarea, pe scurt, a operațiilor efectuate, însoțite de observații, comentarii, soluții, concluzii succinte, în raport de punctul din **Chestiuni de studiat** la care se referă. În anumite cazuri, de exemplu la punctul 6, se pot prezenta panourile frontale și diagrama bloc care să concretizeze soluțiile preconizate, iar la punctul 3 se vor exprima concluziile sub forma unei expresii generalizatoare. Rezultatele, indiferent de forma de prezentare, se vor salva în directorul aferent grupei/studentului, în care au fost salvate și rezultatele obținute la lucrările precedente.