

Curs #1, Programare sem. I - 2024

Prof.dr.ing. Iulian Lupea

PROGRAMARE GRAFICĂ

Modul de evaluare Sem. I (2020)

- 1. Verificare **Scris** cu subiecte din toată materia (curs+laborator): **50% din nota finală; NS $\geq$ 5**
- 2. Verificare **Laborator** și activitatea pe parcursul semestrului : **50% din nota finală; NL $\geq$ 5**
 - 2.1. Dosar cu 3 programe: 2 primite și 1 la alegere + teste scurte pe parcursul semestrului: ND= 40% din NL

Problema #1: dată în săpt. a 4a, Predată în săpt. a 6a (max nota 10) (Format: enunț, PF+diag+explic. detaliate)

Predată săpt. a 7a (max nota 8) altfel 2

Problema #2: dată în săpt. a 8a, Predată în săpt a 9a (max 10) (Format: enunț, PF+diag+explic. detaliate)

Predată în săpt a 10a (max 8) altfel 2

2.2. Verificare la terminal (ultimul laborator #14): **3 probleme: VT= 50% din NL**

- Sunt valabile regulamentele oficiale ale facultății privind prezența studenților la activitățile didactice
- Prezența la curs – obligatorie pt. An. 1.

Despre Absențe:

Recuperare absențelor (maxim 3 laboratoare) se face înainte de săptămâna a 14-a.

IF Nr.AbsenteLabor în săptămâna_14 >0 and Nr.AbsenteLabor < 4

Nu te poți prezenta la Test Laborator din Laborator #14

Catalog: Scris =-2 Labor =-2

Recuperarea absențelor se face în sesiunea de restante cu plată

ENDIF

Despre Nota laborator:

IF NotaTerminal $\geq$ 4.5 și NotaDosar $\geq$ 4.5

NotaLabor = 50% * NotaDosar + 50% * NotaTerminal

Te poți prezenta la examen scris

ELSE

Nu te poți prezenta la examen scris

Catalog: Scris =-2 Labor =-2

În sesiunea de restante: te prezinti din nou la TestTerminal

ENDIF

Despre Nota FINALA în cazul laboratorului promovat

IF NotaScris $\geq$ 4.5

NotaFinal = (NLaborator + NScrist)/2 % media aritm. rotunjită

CATALOG: Nota Finala $\geq$ 5

ELSE

CATALOG: NotaFinala < 5 (Nepromovat), NotaLabor $\geq$ 5 (promovat)

În sesiunea de restante: te prezinti din nou numai la Scris

ENDIF

I. BIBLIOGRAFIE selectivă

1. Bishop, R., Learning with Labview 6i, Prentice Hall, 2001.
2. Cottet, F., Ciobanu, O., Bazele programării în Labview, Editura Matrix Rom, București, 1998.
3. Hedeșiu H., Munteanu R. Jr., Introducere în programare grafică instrumentală, Ed.Mediamira, Cluj-N., 2003.
4. Johnson, G., Labview Power Programming, McGraw-Hill, NY, 1998.
5. Lupea I., Lupea, M., Limbajul C, teorie și aplicații, Casa Cărții de Știință, Cluj-Napoca, 1998.
6. Lupea I., Măsurători de vibrații și zgomote prin programare cu **LabVIEW**, Casa Cărții de Știință, Cluj-N., 2005
7. **Lupea I., LabVIEW - programare grafică, Editura Risoprint, Cluj-Napoca, 2008.**
8. **Lupea I., LabVIEW - Aplicații, Editura UTPRESS, Cluj-Napoca, 2022**
9. Figliola, R., Beasley, D., Theory and design for mechanical measurements, John Wiley & Sons, 2006.
10. WEB:
<https://www.apexwaves.com>
** <http://www.ctanm.pub.ro/academic/LabVIEW/Tutorial.htm>
** <http://www.iit.edu/~labview/Dummies.html>
** <http://www.eelab.usyd.edu.au/labview/contents.html>
** http://wiki.lavag.org/LabVIEW_tutorial
** <http://zone.ni.com/devzone/cda/tut/p/id/7466> Getting Started with NI LabVIEW Student Training
** <http://www.ni.com>
11. Setul de manuale:
** Getting Started with SCXI, National Instruments, 2012,...,2022
** Getting Started with Labview, 2012...2022
** LabVIEW for Windows, User Manual, National Instruments.
** LabVIEW Analysis VI Reference Manual, National Instruments.
** LabVIEW for Windows, Measurements Manual, National Instruments.
<https://www.ni.com/ro-ro/innovations/academic-research.html>

Arduino:

Anghel, T., Programarea plăcii Arduino, 2020

- ** <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019SSZSA2&l=en-US>
- ** <https://microcontrollerslab.com/program-arduino-labview-example/>

II. INTRODUCERE

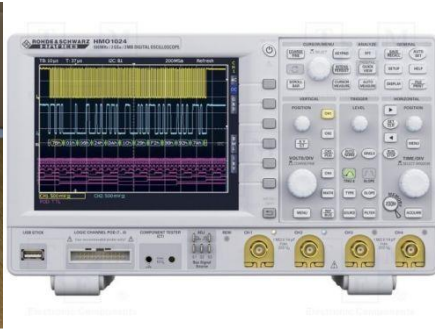
Instrumentele de măsură → Inst. specializat vs. Inst.Virtual (VI)

- evaluează mărimile ce caracterizează fenomenele fizice, chimice, industriale etc.
- oferă mijloace de observare și control în știință și tehnologie.

Echipamente specializate: osciloscopul, sonometrul, voltmetrul, tester vibrații, vibrometru, tahometru/stroboscop (turație), anemometre (viteză vânt), umidometru lemn & beton, termometre cu contact (termocuplu) sau cu fascicul laser/infraroșu, înregistratoare (data logger), multimetru, PH metre, numărătoare/cronometre, luxmetre, instrument pentru măsurare putere laser, UVmetre (radiația ultravioletă), detector rad.γ, debitmetre, analizoare spectrale, surse de tensiune, generatoare de funcții, încă intens folosite pentru supravegherea proceselor industriale, măsurători de laborator.



Sonometru



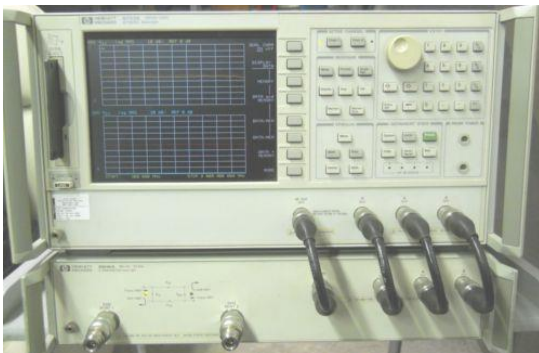
Osciloscop



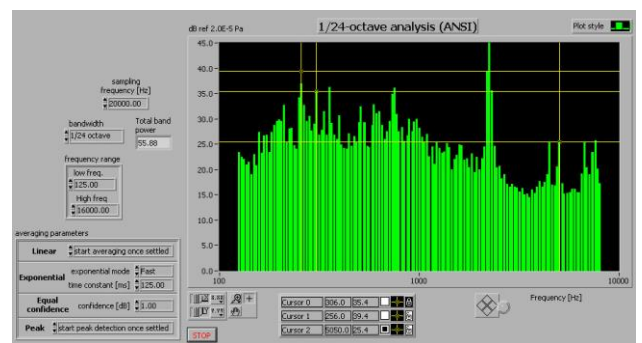
Vibrometru + tahometru



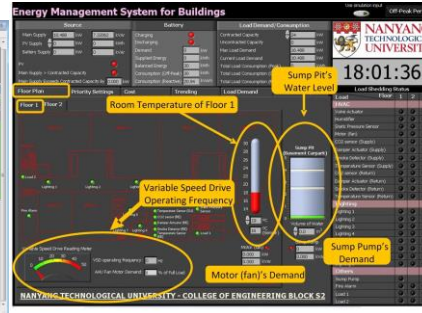
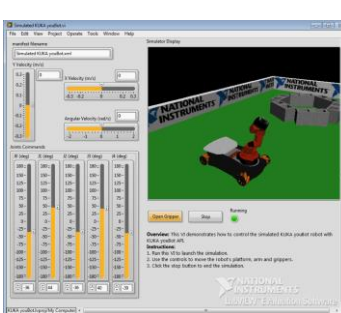
Multimetru



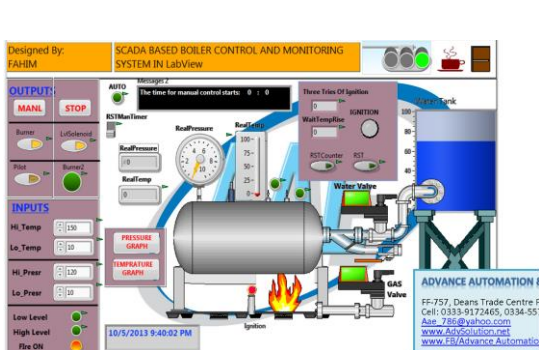
Analizor spectral dinamic



LabVIEW: Analizor spectral 1/24 octave. VI



LabVIEW: Control robot virtual. VI Sistem management energetic clădire.vi Monitorizare temperatură. VI



LabVIEW: Control Boiler. VI



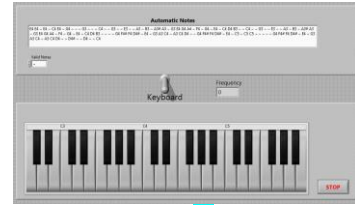
LabVIEW : Management energie solară. VI



Lego Mindstorms



LabVIEW: Control dronă.vi



LabVIEW: Pian.vi

Sistemele de măsură și control integrate prin calculator câștigă teren în laboratoarele moderne și industrie.

- se elimină erorile umane de citire a aparatelor,
- erorile de transcriere a datelor,
- mărește viteza de interpretare a fenomenelor urmărite,
- protejează operatorul uman de măsurătorile repetate rutinale.

Instrumentele de măsură sunt implementate prin intermediul calculatoarelor personale sau laptop
Folosirea sistemelor de măsură integrate prin calculator necesită cunoștințe din ingineria electrică, inginerie mecanică și programarea calculatoarelor.

Firme producătoare de sisteme de achiziție:

National Instruments- Emerson	PCB Piezotronics Inc.,	Tektronix
Bruel & Kjaer	LMS International -> Siemens	Kistler
HP	Keithley	Analog Devices (MEMS)
Data I/O	Rion,	etc.

Mediul de programare grafică LabVIEW dezvoltat de National Instruments

- ocupă un loc privilegiat în programarea interfețelor calculator pentru achiziții de date

Conținutul cursului:

bazele programării în limbajul grafic Labview

- pentru crearea de aplicații numite instrumente virtuale (VI)

Instrumentul virtual =

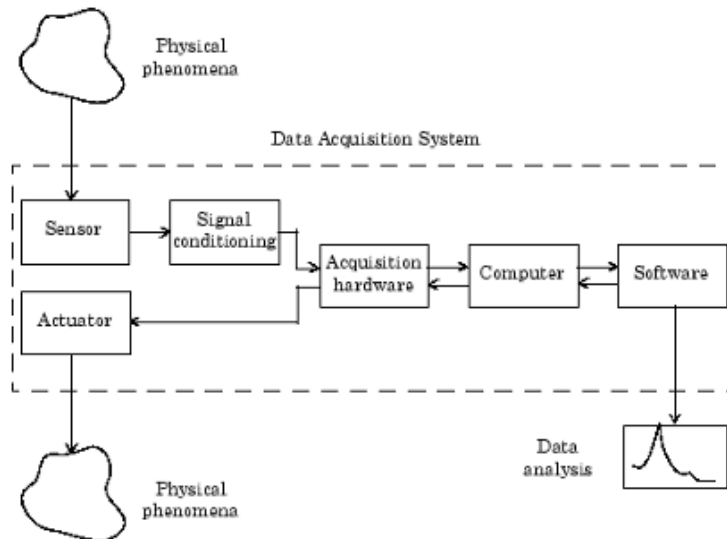
hardware de măsură și control modular +

software specializat (Labview) +

calculator standard (desktop, laptop, telefon mobil)

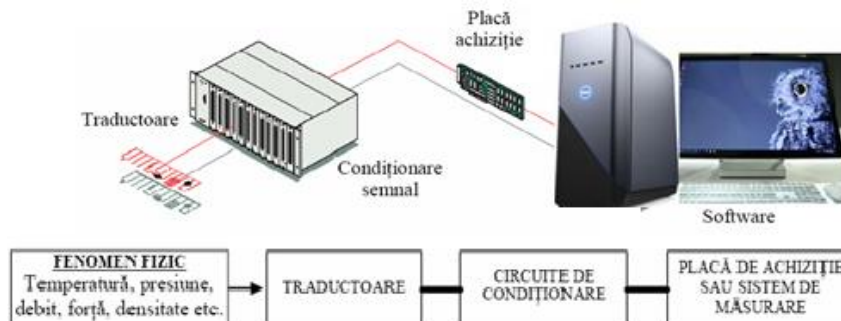
împreună îndeplinind funcțiile unui instrument sau sistem de măsură (și control) specializat.

III. Componentele unui sistem de achiziții de date și măsurători:



Etapele achiziției de date:

1. colectarea de **semnale de la senzori** sau sursele de măsurare
2. condiționarea/**pregătire semnale** și discretizarea semnalelor
3. **stocarea** semnalelor,
 analiza semnalelor
 prezentarea rezultatelor.



2.1. **Eșantionarea** (discretizarea) semnal

→ realizată de convertorul analogic-digital ADC

al plăcii de achiziție

Semnalul analogic $f(t)$ provenit de la traductori urmează să fie eșantionat la fiecare dt secunde, rezultând șirul de valori măsurate:

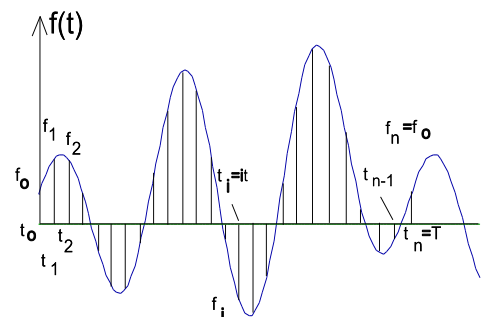
$f_0, f_1, \dots, f_{n-1}$ corespunzător valorilor de timp:

$t_0, t_1, \dots, t_{n-1}$ vezi Figura →

dt este intervalul de eșantionare,

$fs=1/dt$ este frecvența de eșantionare (eșantioane/ secundă),

$fs/2$ este frecvența Nyquist.



Eșantionare semnal de la senzor

Frecvența de eșantionare maximă (f_{s_max}), este un important parametru al unei plăci de achiziție de semnal. Odată cu creșterea frecvenței de eșantionare crește fidelitatea reprezentării semnalului analogic.

Exemplu: placa de achiziție **PCI 4451** poate eșantiona semnalul la frecvența maximă de **204.8 kS/s**. (204800 eșantioane pe secundă)

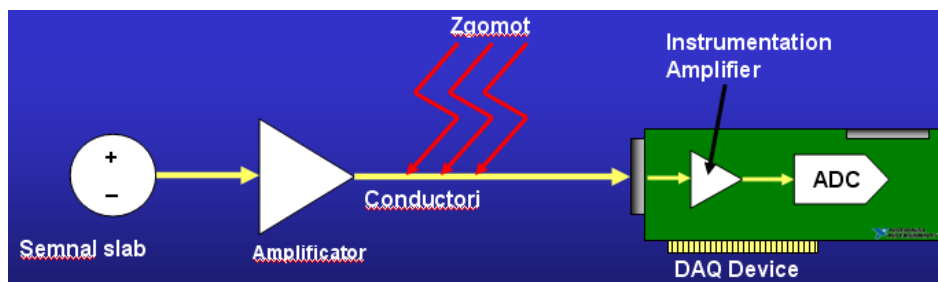


Componentele unui sistem de achiziții date:

1. **Senzorul (transducer)**= este un dispozitiv care convertește fenomenul fizic sau chimic (presiunea, umiditatea, câmpul magnetic, accelerația, forța, intensitatea sonoră, radiații ș.a) într-un semnal electric proporțional a cărui tensiune sau curent se măsoară.

Sunt interfată între instrumente și lumea reală. Fenomenul fizic este monitorizat cu ajutorul senzorilor:

Senzor	Fenomen măsurat
accelerometrul	accelerația la suprafața structurilor
microfonul, hidrofonul	presiunea undelor sonore în aer sau apă
traductor de deplasare liniară, potențiometrul	poziție, deplasare
termocuplu, termistor, RTD=Resistance Temperat. Detectors	temperatură
senzor de forță, traductor tensometric	forță, tensiune structurală, presiune
electrod pH (aciditate)	pH concentrația ionilor de hidrogen H^+ dintr-o soluție



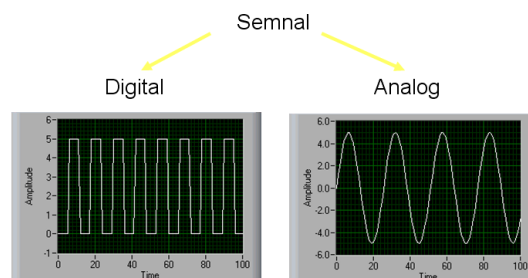
2. Semnalele generate de senzori sau semnalele de comandă, pot fi:

a) **analogice** – variază continuu în timp (caracterizate prin nivel, formă și frecvență) provenite de la accelerometru, microfon, traductor tensometric, senzor de proximitate etc. sau transmise spre excitator de vibrații etc.

b) **digitale** - în general are două nivele (TTL tranzistor-to-tranzistor logic) nivel jos (între 0V și 0.8V) și înalt (între 2V și 5V).

Semnalul este caracterizat de stare (una din cele două) și de rata de schimbare a stării (ex.:semnal de la celulă fotosensibilă sau comandă lichid de răcire, închide/deschide ușă protecție)

Clasificare Semnale



3. Condiționarea sau pregătirea semnalului

Uneori semnalul provenit de la senzor poate fi prea slab și însoțit de zgomot sau prea înalt (tensiune mare) pentru a fi direct măsurat de sistemul de achiziție. Astfel semnalele au nevoie de amplificare sau atenuare înaintea conversiei. Semnalele sunt astfel pregătite pentru a maximiza posibilitățile de măsurare a sistemului de achiziție, a garanta siguranța în exploatare etc. Pregătirea semnalului se face cu accesorii hardware corespunzătoare, care îndeplinesc funcțiuni de: filtrare, amplificare / atenuare, izolare, alimentare senzor pentru a funcționa, multiplexare.

Filtrarea este procesul prin care conținutul în frecvențe al unui semnal este modificat. De exemplu controlul bass alterează frecvențele joase ale unui semnal (piesă muzicală) difuzată de un sistem stereo. Similar frecvențele înalte sunt alterate prin controlul înaltelor (treble).

Filtrele în funcție de banda de frecvențe alterată (eliminată, atenuată) sunt:

1. **trece jos** (lasă să treacă frecvențele joase și atenuază pe cele înalte),
2. **trece sus**, 3. **trece bandă** și 4. **stop bandă**.

Observăm filtre analogice și digitale:

- **filtrele analogice anti-aliasing**: elimină înainte de digitizare deci din semnalul analogic, frecvențele semnalului care depășesc frecvența Nyquist de interes. Plăcile pentru achiziție dinamică DSA conțin astfel de filtre.

- **filtre digitale/soft**: se aplică semnalului după discretizare; sunt obținute prin algoritmi implementați sau programabili soft și fac parte din categoria uneltelor de procesare digitală a semnalelor fiind mai noi (aprox.50 ani) decât filtrele analogice; nu pot fi utilizate pentru anti-aliasing;

- constă în aplicarea transformatei Fourier semnalului discretizat, multiplicarea amplitudinii componentelor de frecvență din domeniul de interes să atingă FRF dorit și transformarea inversă a semnalului din domeniul frecvență în domeniul timp prin transformata Fourier inversă.

- filtrări digitale curente sunt eliminarea zgomotului și decimarea sau reducerea frecvenței de eșantionare.

4. **Hardul** pentru achiziții de date (**plăci de achiziție** - **DAQ board**)

Principalele funcțiuni ale unei plăci de achiziție sunt:

- este interfață între calculator și exterior
- conversia A/D sau discretizarea semnalului analogic recepționat astfel încât calculatorul să poată analiza semnalul,
- achiziția poate fi de la mai multe canale analogice:
 - multiplexare (n canale de achiziție și *un singur* ADC) și
 - eșantionare simultană (n canale și pe fiecare canal este câte un ADC)
- conversii de la semnal digital la analog D/A pentru comanda unor dispozitive
- intrări / ieșiri digitale
 - operații de numărare / temporizare (counter/timer)
- realizează diverse frecvențe de achiziție și generare de semnal.



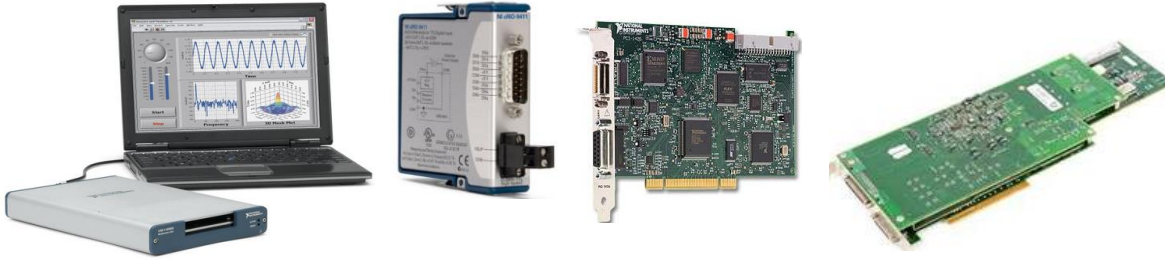
Exemplul: Achiziție de la placa **NI USB 4431**

102.4 kS/s, 4-Input/ 1-Output
Sound and Vibration Device

Calitatea semnalului discretizat depinde de:

- rezoluția de intrare sau ieșire (12 bits, **24 bits**)
- domeniul de tensiune pe canale de intrare ($\pm 5V$, **$\pm 10V$** , $\pm 40V$, ± 42 la 316mV)
- domeniul de tensiune pe canalele de ieșire ($\pm 3.5V_{pk}$)

- rata de eșantionare pe canal (51 kS/s, **102.4**, 204.8 kS/s)
- domeniul dinamic (**Dynamic Range** - DR) sau raportul dintre semnalul maxim și cel minim (noise floor) care pot fi achiziționate prin placă
 - *DR este important pentru aplicațiile DSA (Dynamic Signal Acquisition) cum sunt măsurarea presiunii acustice sau a vibrațiilor mecanice.
 - * $DR = 20 \log_{10}(V_{\max}/V_{\min})$
 - * valori comune pentru plăci DSA sunt (100 dB, 110 dB, 114 dB, 118 dB):
- modul intrării (single-ended sau diferențial)
- zgomotul prezent.



Plăcile de achiziție pot fi conectate la Bus-ul/magistrala calculatoarelor:

- sloturi **PCI** ale calculatoarelor PC sau
- conectabile la **USB**, **Firewall 1394**, **PCMCIA** sau **PCI Express** pentru sistemele portabile
- conectare **Wireless** și **Ethernet** pentru măsurare și **monitorizare la distanță** a semnalelor și proceselor în cazul sistemelor de achiziție mai noi.
- calculatoare specializate PXI și PXI Express
- plăcile sunt proiectate pentru diverse platforme hardware și sisteme de operare
- sistemele moderne prezintă TEDS (Transducer Electronic Data Sheets)= bază de date pentru memorarea caracteristicilor senzorilor folosiți (sensibilitate, domeniu etc.); datele pot fi citite/scrise prin softul de aplicație într-o memorie locală atașată senzorului.

5. CompactDAQ produse de [National Instruments](#)

5.1. Sasiu **USB NI compactDAQ™-9178**, **8-sloturi**

- integrează (includ) hardware pt. data I/O cu [LabVIEW](#) pentru achiziție date, procesare date achiziționate și analiză date de la senzori

In sloturi intră 8 plăci de achiziție --->

- sistemul este portabil, conectare prin USB
- canale I/O analogice și canale I/O digitale;
- la canalele de intrare se conectează senzori
- șasiul CompactDAQ- 9178 este la un pret mai redus decât platforma (șasiul) NI **PXI** Platform.

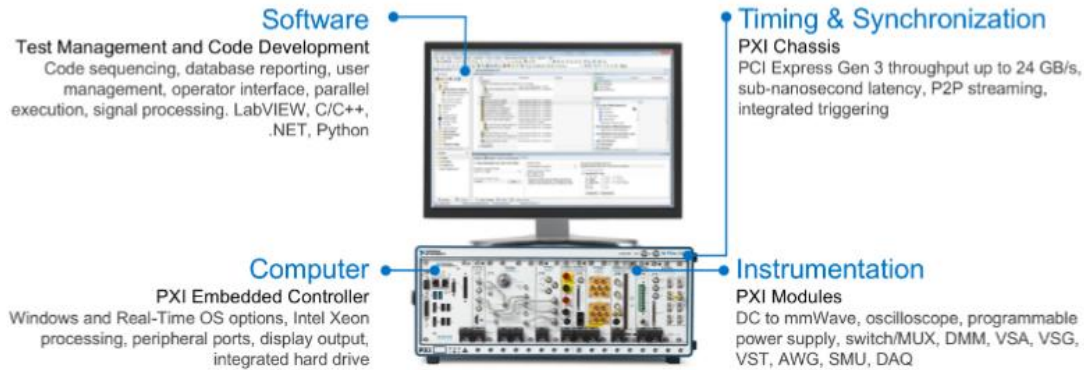


5.2. **CDAQ-9133** Controller →

cu certificări standard pentru industrie. Integrates a multi-core processor with modular I/O, standard peripheral connectivity, and display capabilities (20mii - 30mii lei).



6. NI PXI Platform (PCI eXtensions for Instrumentation) <http://www.pxisa.org/>



PXI = platforme de înaltă performanță pentru diverse aplicații:

- teste în industria manufacturieră,
- aplicații militare și aerospațiale,
- monitorizare mașini prelucrătoare, industria de automobile, diverse teste în industrie.

7. CompactRIO system = a real-time embedded industrial controller (produse de NI)

pentru sisteme de control industrial

CompactRIO = o combinație de real-time controller,

Reconfigurable **IO** Modules (RIO),

FPGA module și un

Ethernet expansion chassis.

- sunt ideale pentru **Industrial Internet of Things (IIoT)**, aplicații de monitorizare și control în industrie

- prezintă controler cu procesor și/ sau user-programable FPGA
- prezintă capacități de procesare de înaltă performanță,
- include module de condiționare de semnale I/O produse de NI sau alți furnizori
- conectare directă la senzori
- integrează software
- sunt robuste - pentru mediu industrial

[1]

Compact RIO COMPONENTS:



8. myRIO **Student** Embedded Device

The LabVIEW myRIO Toolkit is a software add-on for LabVIEW that provides the necessary driver and tools to quickly develop and deploy applications to the myRIO Student Embedded Device. the toolkit includes configuration-based Express and Advanced I/O VIs for each I/O type on the myRIO device, which helps students quickly prototype ideas and create stand-alone applications.

Stand-alone or computer-based devices either integrate with standard desktop and laptop computers or allow you to use them without the need for other modular hardware.



myRIO Student Embedded Device

Starting from RON 2.160,00

Provides reconfigurable I/O that allows you to teach and implement multiple design concepts with one device.



myRIO Controls Mechatronics and Robotics Device

Starting from RON 18.950,00

Extends the capabilities of the myRIO Student Embedded Device for teaching mechatronics and controls.



myDAQ Student Data Acquisition Device

Starting from RON 1.775,00

Combines a comprehensive set of plug-and-play computer-based lab instruments with portability for hands-on student learning in or outside the lab.

9.Arduino

- este o placă 'open source hardware' construită pe baza unui microcontroler
- include conectoare spre periferice
- microcontrolerul este programat prin calculator (mediul IDE Arduino, Labview etc.)
 - * programul este transferat prin cablu USB spre placă
- sunt multe biblioteci 'open source' pentru interfațarea controlerului cu componente externe: **senzori, LED ,motor, LCD, keypad, Bluetooth module, GSM module** etc.

Foarte cunoscute studenților din inginerie sunt:

Arduino UNO R3

Arduino MEGA



alte plăci populare: Galileo, Raspberry Pi

2. Drive-re și aplicații software

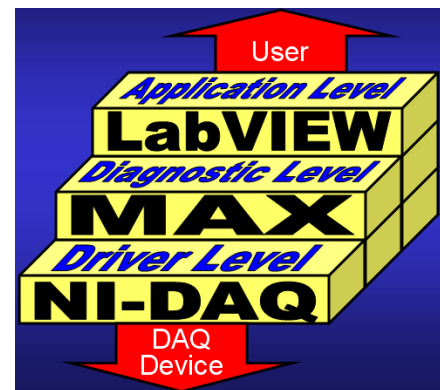
a) NI-DAQ și NI-DAQ (Legacy) = **drive-re** care rulează în calculator controlând (conducând) dispozitivele hard de achiziție, fiind punte de legătură între acestea și softul de aplicație.

* **NI-DAQmx** = **drive-re** care au crescut mult viteza de dezvoltare a aplicațiilor prin asistare și interactivitate grafică pentru configurare, testarea sistemului de achiziție.

b) **Measurement & Automation Explorer (MAX)**

- este un utilitar instalat de Labview sub Windows
- folosit **pentru configurarea plăcilor de achiziție**, a canalelor de achiziție și pentru diagnoză,
- după instalarea unei plăci de achiziție se va rula MAX.

c) **Produsele soft de aplicație** sunt medii de dezvoltare în care sunt programate aplicațiile finale cu orientare spre client. Acestea realizează:



- trimiterea de comenzi către drivere pentru achiziție/comanda
- prelucrarea datelor achiziționate și
- prezentarea rezultatelor.

NI oferă patru produse soft pentru dezvoltarea de aplicații:

LabVIEW – programare grafică,

LabVIEW NXG (2017) = integrează cod dezvoltat în alte produse soft și limbaje open-source cu third-party software interoperability tools.

LabWindows/CVI pentru programatorii în C++
Measurement Studio

DIADEM pentru calcule și raportare.

Prin introducerea softului de aplicație Labview
SignalExpress se oferă dezvoltarea de aplicații finale
pentru client fără a necesita efort de programare.

- softul de aplicație conține biblioteci de funcții de analiză și vizualizarea datelor.

Pentru a observa instalarea unei plăci de achiziție se va urmări lista componentelor hardware a calculatorului:
Start /Control Panel /System Properties /Device Manager/



Developer(s)	National Instruments
Initial release	1986; 38 years ago
Stable release	LabVIEW NXG 5.1 LabVIEW 2024 Q3 / July 2024; 3 months ago
Written in	C, C++, C#
Operating system	Cross-platform: Windows, macOS, Linux
Type	Data acquisition, instrument control, test automation, analysis and signal processing, industrial control, embedded system design
License	Proprietary
Website	www.ni.com/labview

IV. CARACTERISTICI LabVIEW

(LABoratory Virtual Instrument
Engineering Workbench)

- National Instruments a introdus conceptul de instrument virtual (VI)
- Realizează achiziție de date, analiză complexă a datelor măsurate, simulează aparate de analiză și control, este performant în prezentarea datelor și rapoarte – toate acestea într-un singur mediu de dezvoltare
- Fiind **intuitiv** prin natura grafică a programării permite concentrarea la teorie, programare și experiment și mai puțin la sintaxă, cum este în cazul programării textuale
- Timpul necesar dezvoltării de aplicații complexe este mai scurt decât în cazul programării textuale
- LabVIEW este considerat standard industrial pentru teste, măsurători, automatizare și control în SUA.
- Este dedicat inginerilor, chimiștilor, fizicienilor, oamenilor de știință, studenților, elevilor
- Prin compilare (built-in) se generează cod optimal obținând viteză de execuție comparabilă cu programele C compilate,
- LabVIEW suportă **Matlab** Mscript și HiQ (ambele folosite pentru calcule numerice și simbolice), suportă directă conectare cu Mathematica, MathCad, Excel, MultiSim from Electronics Workbench, and Texas Instruments' Code Composer Studio1,
- Se poate scrie direct **cod C** în Labview sau apela rutine cod mașină,
- Se constată succes și cerere de Labview în multe universități,
- Familiarizează studentul cu instrumentația virtuală și cu concepte de virtualitate,
- Oferă o alternativă la limbajele textuale diversificând sectorul de programare,
- Este relativ ușor de învățat,
- Există variante pentru studenți: **LabVIEW Student Edition**,
- Este introdus și în învățământul preuniversitar/ liceu,
- Folosit în cercurile de robotiști pentru control,
- În mediul industrial este mai utilizat decât limbajele textuale,
- Pe lângă calculul secvențial suportă natural calculul paralel,

- Depanarea programelor este sugestivă și vizuală,
- Lecția de programare învățată în anii mici va fi baza pentru aplicații de achiziții de date și control în anii mai mari,
- Calculele numerice prin programare în Labview se pot face rapid la fel ca în MathCad, MatLab etc.
- Pentru aplicații specializate oferă module independente/ Toolsets:

LabVIEW Toolsets

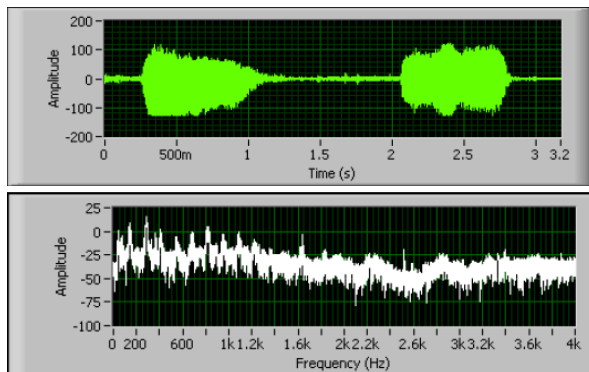
- Digital Filter Design ,
- DSP Test Integration Toolkit (for TI DSP)
- LabVIEW Math Interface Toolkit
- Simulation Interface Toolkit
- Enterprise Connectivity Toolset
- LabVIEW FPGA (add-on) design FPGA-based systems; Digital Communication Interfaces
- PID Control Toolset
- Sound and Vibration Measurement Suite
- Report Generation Toolkit for Microsoft Office
- Control Design Toolkit
- Modulation Toolkit
- LabVIEW Express VI Development Toolkit
- LabVIEW State Diagram Toolkit
- Signal Processing Toolset
- Order Analysis Toolkit
- System Identification Toolkit
- etc.

- Advanced Signal Processing Toolset conține:

Time Frequency Analysis Tools (remove noise from signal), Wavelet Analysis Tools, Time Series Analysis Tools...

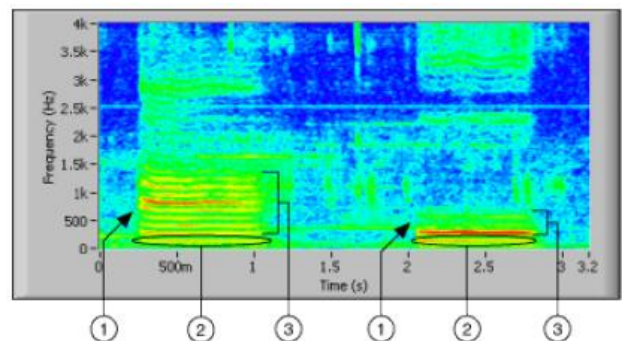
→ Time-frequency analysis on a nonstationary signals

Semnal vorbire (în timp), FFT semnal,

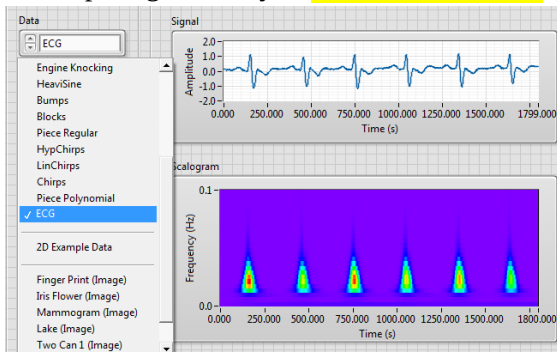


a) STFT spectrogram

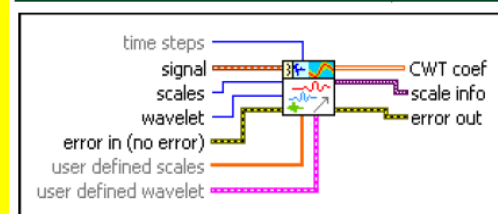
1.Onset, 2=fundamental frequency, 3=harmonic frequencies



b computing the analytic wavelet transform (WT)



WA Continuous Wavelet Transform (Waveform)



modul WFBD = Wavelet and Filter Banks Designer = folosit pentru data compression

(prin proiectare interactivă a filtrelor interactively design the filters in the filter bank and simultaneously monitor the effect they have on the signal. Because of their ability for effective feature extraction, wavelets are widely used for AI.

Arduino™-Compatible Compiler for LabVIEW



FROM € 99,00

Price may vary with addition of accessories or

Item details

Supported OS: Windows
License Size Unit of Measure: Seat(s)
License Size Value: 1
Delivery Method: Download
Language: English
License Term: Perpetual
Software Use: Development

The Arduino™-Compatible Compiler for LabVIEW helps you compile and download LabVIEW VIs to Arduino targets.

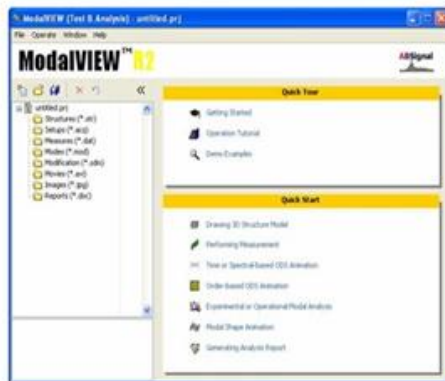
The Arduino™-Compatible Compiler for LabVIEW is a software add-on that you can use to program stand-alone embedded systems on Arduino-based hardware. The add-on uses a subset of LabVIEW functionality and supports several LabVIEW primitives and Arduino API VIs. You can also use the Arduino™-Compatible Compiler for LabVIEW to interface with Arduino I/O and shields using custom palettes.

Part Number(s): 784149-35 | 784150-35

ModalVIEW

ABSignal

FROM € 4.990,00



- ModalVIEW expands support to [NI USB-443x](#)
- [ModalVIEW R2](#) is available for download

ModalVIEW -- easily acquire multi-channel sound or vibration signals from a running machine, or the static or dynamic loading of a structure by utilizing data acquisition hardware; ■ with the set of time histories, it can animate the response of a structure and show you the structure's vibration behavior at once; ■ extract and visualize useful modal parameters information from acquired time- and frequency-domain experimental data; ■ ModalVIEW software is developed under National Instruments' LabVIEW, an open environment designed to make interfacing with any measurement hardware rapidly and simple; ■ ModalVIEW is compatible with any dynamic signal acquisition (DSA) hardware of National Instruments including plug-in boards, USB devices, and PXI systems and speed your time to perform experimental measurement; ■ ModalVIEW a low-cost modal testing and analysis tool.



LabVIEW Real-Time Module

The LabVIEW Real-Time Module helps you create reliable, stand-alone applications that execute on embedded hardware devices.

The LabVIEW Real-Time Module is add-on software for the LabVIEW programming environment that you can use to create and deploy real-time, distributed system applications for test, monitoring, and control. You can take advantage of a real-time OS to ensure reliability and precise timing in your stand-alone system. You also can save valuable development time by selecting from hundreds of prewritten LabVIEW libraries and hardware APIs, which enable deterministic data acquisition. You can choose from a development license or a debug/deployment license, which you can use to fix issues on applications that were created with a development license.

LabVIEW Internet Toolkit: provides you with the ability to incorporate Internet capabilities into VIs. You can use LabVIEW to work with XML documents, build CGI programs and URLs, and send and receive data.

File Transfer Protocol (FTP), HyperText Markup Language (HTML); HyperText Transfer Protocol (HTTP); Telnet Protocol

TCP/IP Communication in LabVIEW

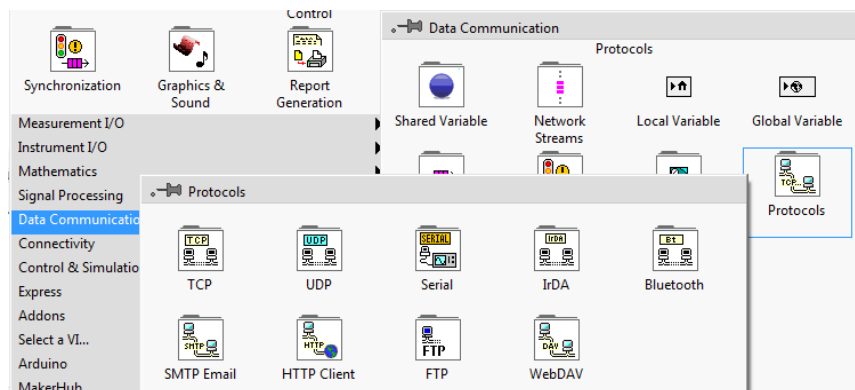
Two basic tools for network communication:

1. Transmission Control Protocol (TCP) and 2. Internet Protocol (IP). With TCP/IP you can communicate over single networks or interconnected networks (Internet).

Refer to the [Using LabVIEW with TCP/IP and UDP](#) from LabVIEW Help for more information about how TCP/IP communication works.

TCP/IP functions located on the **Functions»Data Communication»Protocols»TCP** palette for TCP communication in LabVIEW.

As with DAQ, instrument, and File I/O communication, the process involves opening the connection, reading and writing the information, and closing the connection. The programmer is responsible for developing both the client and the server.



Se poate integra/apela cod:

Python, MathWorks® MATLAB® Software, C/C++ și .NET
în fluxul grafic de date din aplicațiile LabVIEW

1. Integrate/call **Python scripts** din Diagrama Bloc LabVIEW (folosim Python Node)
2. Call **.m files** developed in MATLAB (folosim MATLAB Node în block diagram-BD)
3. Call **C/C++ DLLs** (dynamic link libraries) in LabVIEW
4. Crează/ comunică cu .NET Framework objects in LabVIEW (folosim Constructor Node în BD)

*In versiunea 8.2, s-a introdus LabVIEW Object-Oriented Programming (**LVOOP**)

Artificial Intelligence toolkits



LabVIEW Analytics and Machine Learning Toolkit

FROM € 455,00

Price may vary with addition of accessories and services.

Item details

Delivery Method: Download

Supported OS: Windows

License Size: 1 Seat(s)

The LabVIEW Analytics and Machine Learning Toolkit integrates predictive analytics and machine learning into LabVIEW.... (2018)

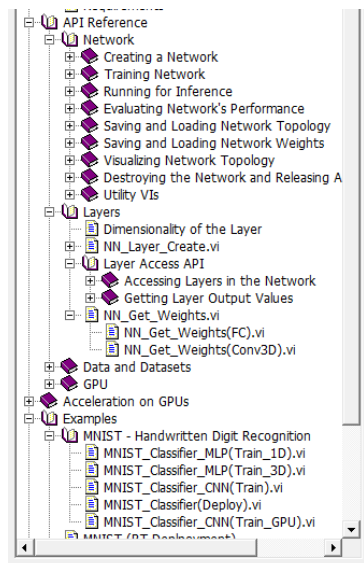
The LabVIEW Analytics and Machine Learning Toolkit is a software **add-on** for LabVIEW that provides training machine learning models. You can use these models to discover patterns in large amounts of data with anomaly detection and classification, and clustering algorithms. Additionally, these models can recognize patterns in new data on NI Linux Real-Time and Windows targets. The LabVIEW Analytics and Machine Learning Toolkit helps you work with condition monitoring, predictive maintenance applications, and more. The add-on also includes getting-started and real-world examples. You can choose from a development license or a deployment license, which you can use to distribute developed code. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. <https://forums.ni.com/t5/NI-Labs-Toolkits/LabVIEW-Machine-Learning-Toolkit/ta-p/3514074>

Deep Learning Toolkit pentru LabVIEW este un soft **add-on**:

create, configure, train, and deploy deep neural networks (DNNs).

With this add-on, you can accelerate training and inference of DNNs on **Nvidia graphics processing unit (GPUs)**. The add-on supports LabVIEW Real-Time targets for deployment and inference. Plus, the Deep Learning Toolkit for LabVIEW helps you visualize DNN topology and metrics such as memory footprint and computational complexity. When executed on a **CPU**, the add-on does not require an external library or engine to function.

Deep Learning by Ngene - Toolkit for LabVIEW <https://www.ngene.co/>



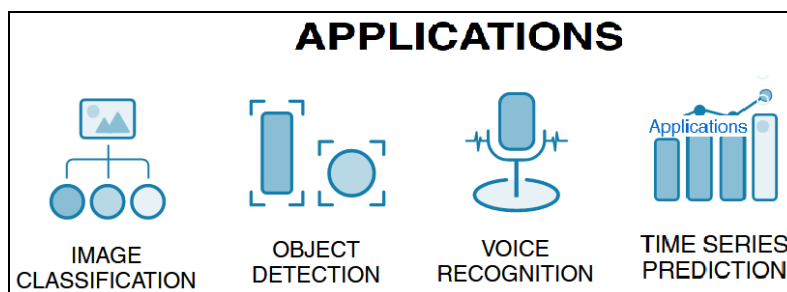
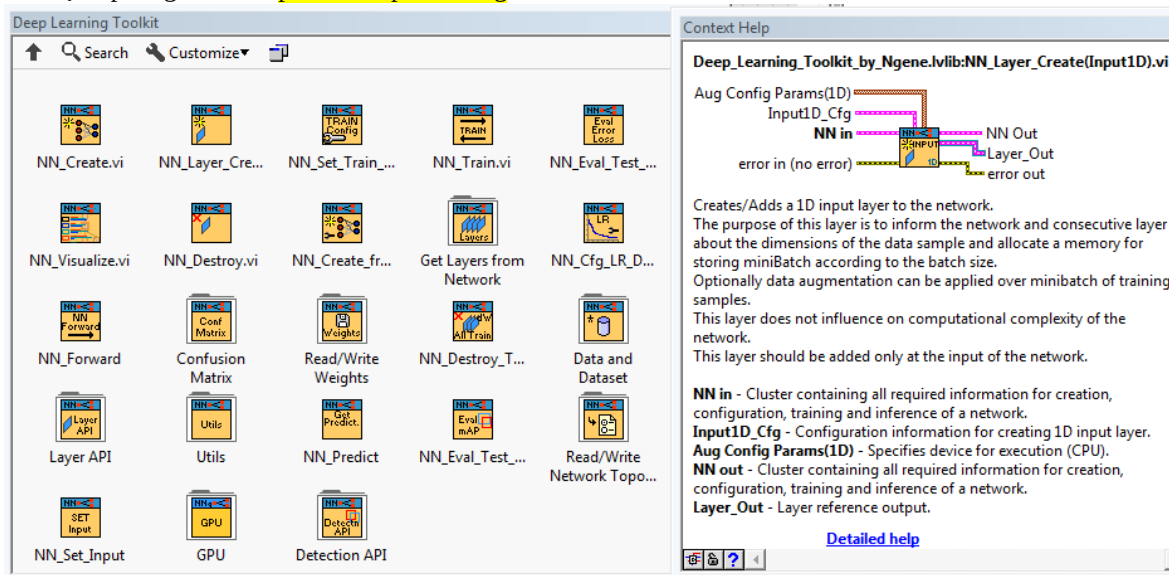
DEEP LEARNING TOOLKIT for LabVIEW

Deep Learning is a subfield of machine learning that provides state of the art performance in such fields of artificial intelligence, as object classification and detection, and is finding more applications in the fields, like autonomous vehicles, medical diagnosis, speech recognition, predictive maintenance and many other applications where explicitly programmed algorithms are not applicable or incapable to provide acceptable performance. There are many frameworks for Deep Learning (Caffe, TensorFlow, Theano, Torch) in the market, but they are mainly supporting Python and C/C++ programming language. Deep Learning toolkit, developed at Ngene, is a high abstraction level API providing possibility to build, configure, train, evaluate and deploy deep neural networks in the LabVIEW programming environment.

Deep Learning Toolkit for LabVIEW is designed to bring the possibilities of deep learning infrastructure into LabVIEW and provide access to such machine learning applications as image recognition, object detection and speech recognition to the community of LabVIEW developers. Designed in a modular way the toolkit simplifies the process of creating, building, configuring, training and visualizing of deep neural networks in LabVIEW.

Deep Learning Toolkit was completely developed in LabVIEW and when executed on CPU does not require any external library or engine for functioning, which makes seamless the process of deploying trained systems on embedded targets for inference.

funcții - pictograme din **paleta Deep Learning**

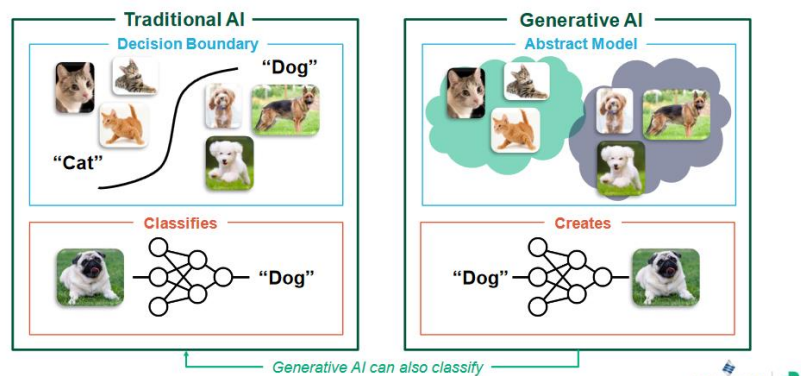


Versions: since 2017 ... → [8.0.1.252 \(Jun 13, 2024\)](#)

LabVIEW Generative AI 2024 - va fi inclus în LabVIEW Deep Learning Module

designed to facilitate the local execution of open-source large language models (LLMs).
crește eficiența automatizărilor industriale

The power of LabVIEW combined with HAIBAL's efficient architecture make the deep learning toolkit provides solutions with easy-to-learn, yet extremely powerful features to shorten product development time, reduce costs and improve quality.



NIGEL AI = AI assistant that can **help code in LabVIEW** and is capable of controlling hardware and creating graphical code

V. Mediul LABVIEW

1. PANOUL FRONTAL (PF)

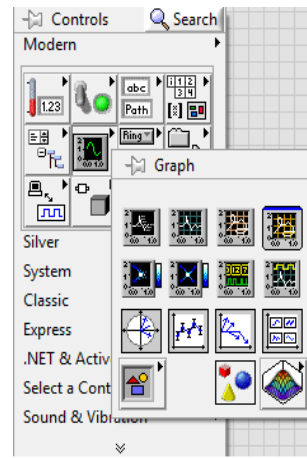
PF conține **controale** și **indicatoare**.

PF are aspect și funcționalitate asemănător panoului frontal al aparatului fizic (sonometru, osciloscop, generator de funcții/semnal, analizor spectral etc.)

*conține obiecte de **control** și reglaj (butoane, comutatoare, etc.) de tip numeric, logic prin care se introduc date în VI

*obiecte pentru **vizualizarea**/ indicarea rezultatelor de prelucrare sau măsurare

– are largi posibilități de formatare a obiectelor, valorilor numerice, de setare a culorilor, a preciziei de afișare.



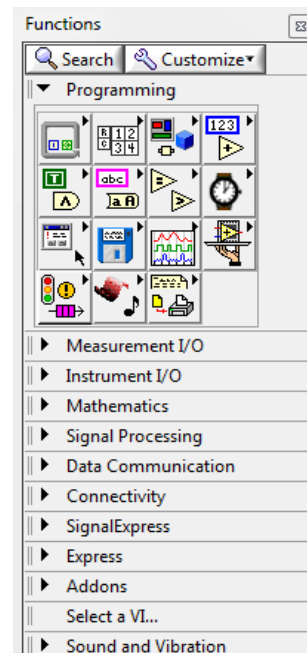
Palete în Panoul

2. DIAGRAMA Block (DB)

- conține codul sursă în format grafic, seamănă cu o schema logică de funcționare a instrumentului,

- obiectele tip control sau indicator din PF apar în DB ca terminale. Conține structuri de programare, funcții și furnizează date indicatorilor din PF. Terminalele asociate controalelor și indicatoarelor, structurile și funcțiile se numesc **noduri**.

Nodurile sunt conectate prin **fire**.



Palete în Diagrama

** prezintă meniul/paleta **Functions** →

3. tipuri de date:

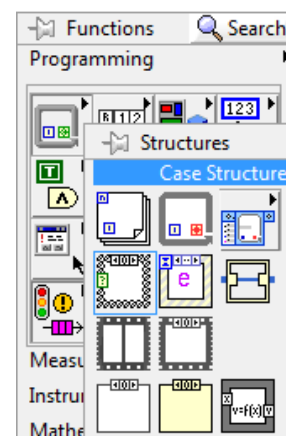
- ***întregi** cu și fără semn, enumerare,
- ***reale** (simplă și dublă precizie), **complexe**,
- *tipul **logic** (boolean),
- *tipul **caracter** și șir de caractere,
- *tipul **cale spre fișier**,
- *tipuri de date structurate de tip **tablou** și **structură**,
- *tipul **formă de undă** (waveform)

-conversii forțate între tipuri de date

- sunt disponibile: constante numerice, logice, șir de caractere.

4. structuri de control ale limbajului în meniul “**Functions**”:

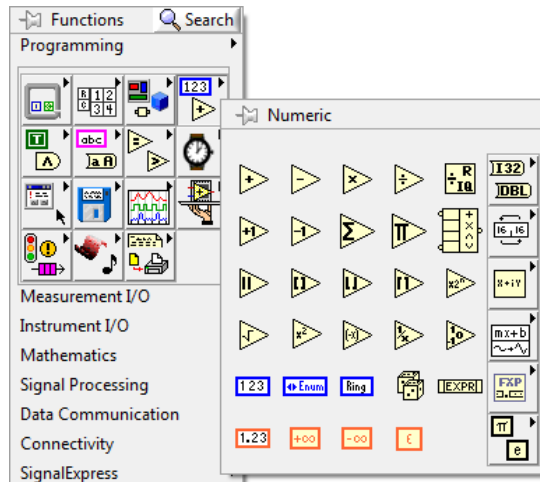
- structuri de ciclare tip **FOR**:
numărul de repetări cunoscut dinainte,
- structuri de ciclare tip **WHILE**
cu oprirea ciclării printr-o condiție,
- structuri pentru decizii multiple (tip **CASE**),
- stabilesc secvențe în derularea calculelor și acțiunilor (**Sequence**),
- programare pe evenimente **Event Structure**
- controlul timpului **Timed Structures** (timed Loop, timed Sequence)



Structuri de control

5. palette de funcții

- paletele de funcții pentru prelucrarea datelor sunt în meniul “**Functions**”
- introducerea de formule de calcul facilitează ‘Formula Node’ și ‘Expression Node’,
- operatori aritmetici pentru numere reale și complexe, operatori logici și operatori relaționali de tip unar, binar și ternar,
- funcțiilor trigonometrice, hiperbolice, exponențiale și logaritmice.
- funcții de prelucrare a tablourilor și referirea elementelor tablourilor



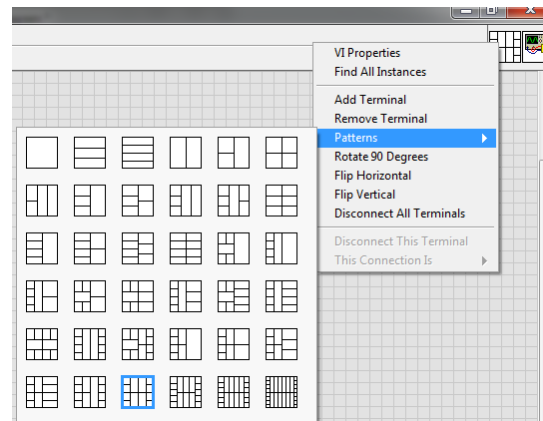
Paleta Programare/ Subpaleta **Numeric**

- pot fi generate și prelucrate diverse semnale în domeniul timp și frecvență.
- filtre digitale, funcții pentru prelucrări numerice, funcții specifice algebrei liniare, funcții pentru calcule statistice, analiză de regresie, etc.
- salvarea datelor în fișiere.
- funcții predefinite (VI) pentru achiziții de date și generare de semnale analogice și digitale.

3. PANOUL de CONECTARE (connector pane)

Connector Pane (CP) este utilizat pentru intrarea și ieșirea datelor din SubVI.

CP este disponibil în panoul frontal PF colțul sus dreapta.



6. FLUXUL DE DATE în execuția unui program LabVIEW

Programul Labview:

1. Construirea **diagramei bloc** DB (codul sursă Labview) și a
2. Panoului Frontal sau interfața cu utilizatorul (controale și indicatoare necesare)

Diagrama = rețea grafică de **NODURI** conectate prin **FIRE**

Nodurile= funcții, subprograme, operatori

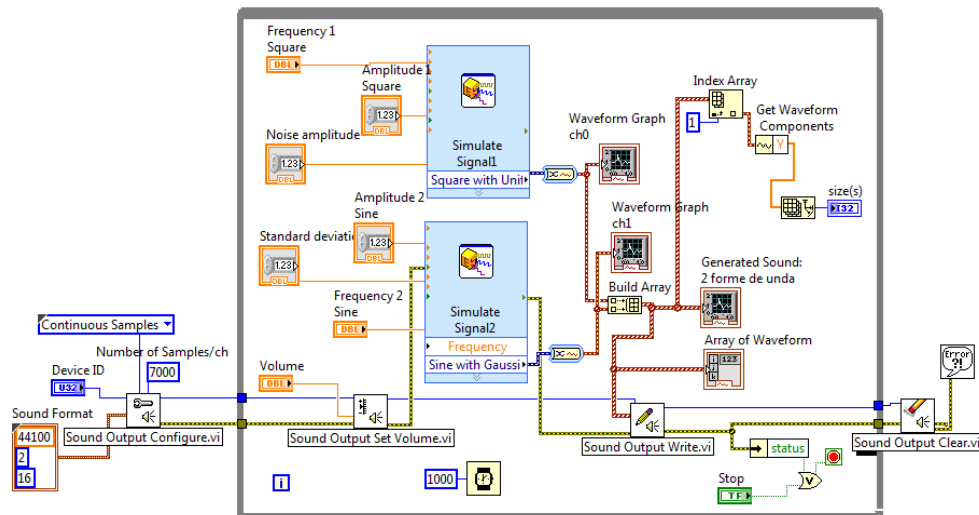
Firele = legături prin care se face transfer de date (variabile) într-un singur sens.

O FUNCȚIE se EXECUTA atunci **când are disponibile toate datele necesare** primite prin firele de intrare (de la funcții sau controale). Un nod (funcție) fără intrări se execută imediat.

Mai multe noduri pot avea simultan date necesare la intrare deci **se pot executa în paralel**

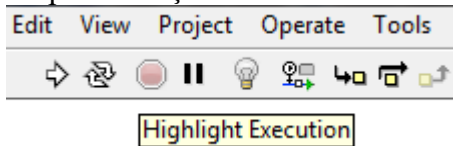
După execuția unui NOD se generează date de ieșire care se propagă prin Fire la nodurile conectate sau la indicatori în PF.

Deplasarea datelor prin noduri reprezintă execuția programului



Vizualizarea FLUXului de DATE

Se va folosi opțiunea de depanare (**debugging**) pentru vizualizarea fluxului de date în timpul execuției unui VI "Execution Highlighting" 💡.



Fire și tipuri de date - exemple

tip dată	scalar	1D Array	2D Array
Numeric			
Boolean			
String			

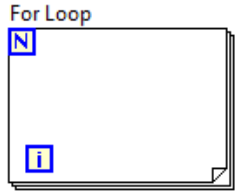
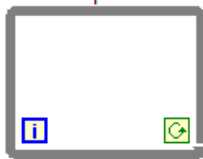
Un instrument virtual a.VI sau aplicație Labview poate fi executată ca:

1. un program independent, cu panoul frontal având rol de interfață cu utilizatorul
2. ca funcție **a.vi** cu pictograma plasată în cadrul unei aplicații mai ample **ab.vi**;
 - a.vi este nod în diagrama lui ab.vi
 - panoul frontal a lui a.vi definește intrările și ieșirile nodului cu ajutorul panoului de conectare (**connector pane**)
 - a.vi înainte de a deveni nod în ab.vi poate fi ușor testată pentru rulare corectă.

7. CONTROL și INDICATOR în PF și terminale în DB

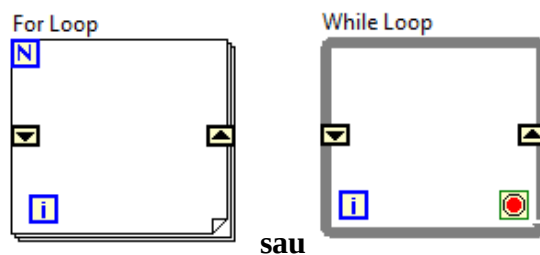
In Panoul Frontal PF		Terminale în Diagramă DB
Controale de diverse tipuri de date. Controlul are asociat în diagr. un terminal cu contur linie groasă .	Indicatorul are asociat în diagramă un terminal cu <u>contur subțire</u> <u>Indicatoare</u> de diferite tipuri de date (numeric, logic, șir caractere).	Numeric  Array  Numeric 2  Array 2  control (bold) indicator

8. INSTRUCȚIUNI DE CICLARE

FOR loop	WHILE loop
	While Loop  While Loop  sau:
Execută (repetă) codul din corpul ciclului (subdiagrama) de N ori; se conectează din exterior o valoare întreagă la terminalul N	Repetă codul din corpul ciclului cât timp o condiție are loc. Terminalul conditional funcționează în 2 variante: 1. STOP if TRUE 2. CONTINUE if TRUE Corpul se repetă cel puțin odată.
i - <u>contorul de iterații</u> furnizează valoarea iterației curente de la 0 la N-1.	

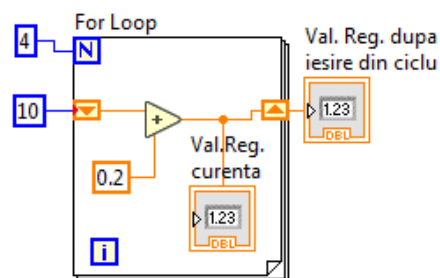
Registrul de transfer (Shift register)

1.Registrul de transfer transferă date de la o iterație a ciclului For sau While la iterația următoare. Terminalele registului sunt negre deci nu este încă stabilit ce tip de dată va fi conținut în registru.



2.Registrul inițializat

Val.Reg. curenta	Val. Reg. dupa iesire din ciclu
10.8	10.8



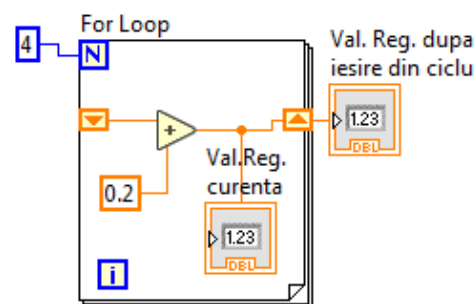
3.Registrul neinițializat

La terminalul stânga al SR nu este conectată nici o dată de input astfel registrul este neinițializat.

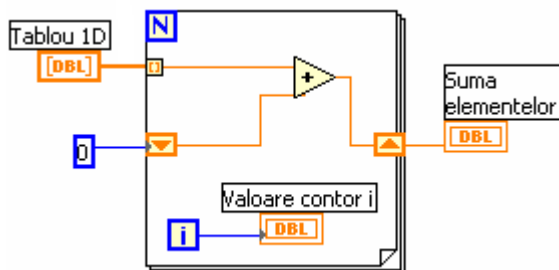
Dacă terminalul din stânga nu are date de intrare conectate la prima iterație a ciclului data din terminalul stâng va fi valoarea implicită pentru tipul de dată care este memorat în registru.

Val.Reg. curenta	Val. Reg. dupa iesire din ciclu
0.8	0.8

Val.Reg. curenta	Val. Reg. dupa iesire din ciclu
1.6	1.6



Suma elementelor din tabloul 1D / FOR loop



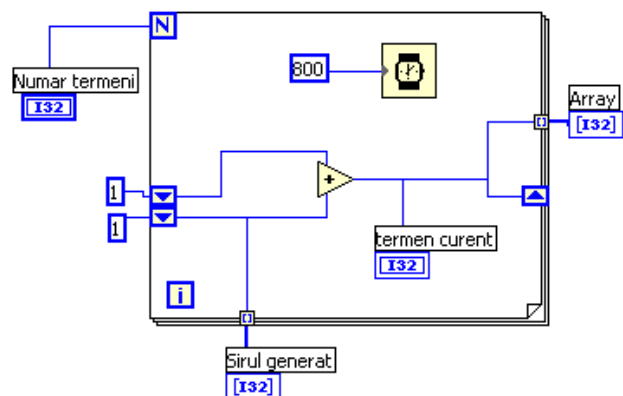
Registrul Shift **multiplu** sau stivuit

Relația de calcul a elementelor din șir:

$$f_{n+1} = f_{n-1} + f_n, \quad f_0 = f_1 = 1$$

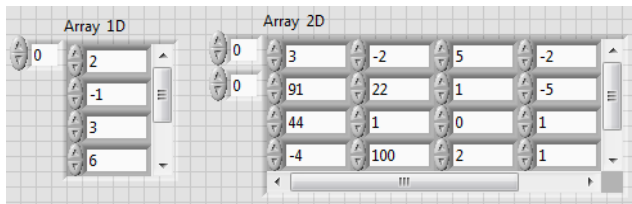
La prima ciclare cele doua 'sertare' din stanga ale unicului registru de transfer primesc valoarea 1; după adunare valoarea (2) se salvează în sertarul dreapta; la iterația următoare valoarea 2 se transferă în sertarul de sus-stanga iar valoarea din sertarul sus este salvată automat în sertarul stanga jos etc.

2. Calculați și afișați șirul Fibonacci folosind **doi regiștri shift**.



9. Tablouri 1D și 2D (Array 1D și 2D) numeric, boolean(logic) și Sir de caractere

Control Array / tablou 1D sau 2D de numere **reale** în PF și terminale asociate în diagrama



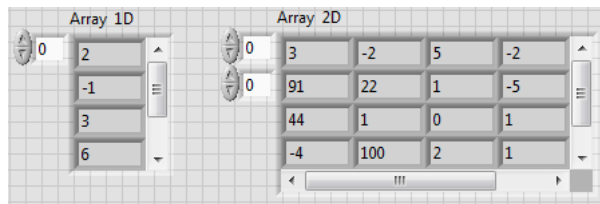
Array 1D
Real DBL



Array 2D
Real DBL



Indicator Array 1D, 2D valori **reale** și terminale asociate (DBL)



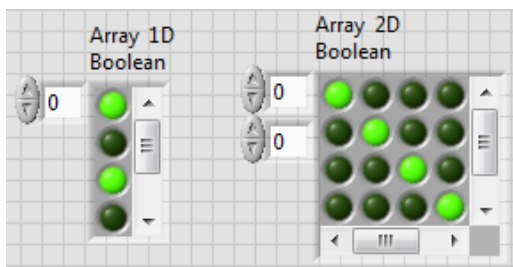
Array 1D
Real DBL



Array 2D
Real DBL



Control tablou 1D, 2D valori **logice** și terminalele asociate (TF)



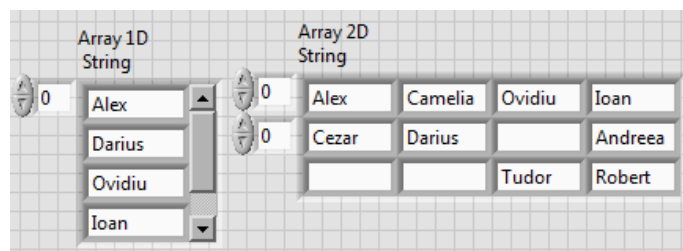
Array 1D
Boolean



Array 2D
Boolean



Control tablou 1D, 2D de **siruri de caractere** și terminalele asociate (abc)



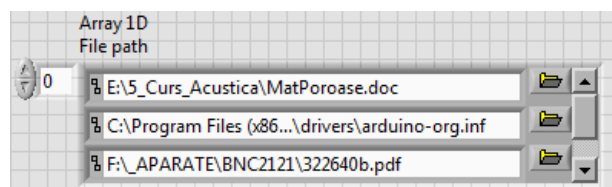
Array 1D
String



Array 2D
String



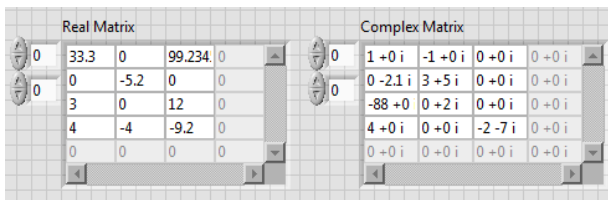
Control tablou 1D **Cale spre fișier**



Array 1D
File path



Control **matrice reale** (DBL) și matrice cu **valori complexe** (CDB):



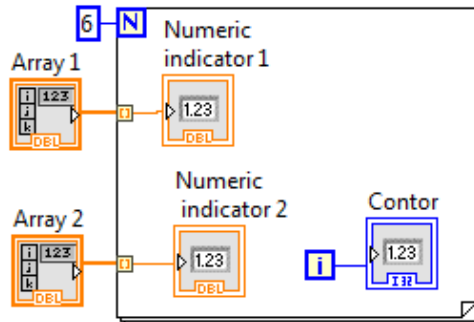
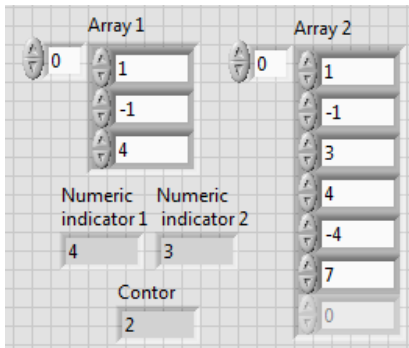
Real Matrix



Complex Matrix



Ciclul FOR, Nr. iterații = **minim(N, nr_elem_tablouri_indexate)**



10. APLICAȚII la CURS #1:

Interactiv sunt realizate următoarele **aplicații**:

Suma/media valori șir cu **FOR Loop** + registrul Shift (transfer sau deplasare),

Suma șir cu **WHILE Loop** (continue if True, stop if True),

Registrul Shift multiplu, Generare șir **Fibonacci**: $f(i+1)=f(i)+f(i-1)$, $i=0$
 $\rightarrow 0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$

Suma elemente tablou 2D cu **FOR Loop imbricat** + regiștrii Shift.

Suma elemente tablou 2D cu **FOR Loop (pentru linii) + While Loop** + regiștrii Shift.

11. GENERARE de ȘIRURI NUMERICE - tip de dată real

1.Semnale periodice de tip: ...**Pattern.vi**

(linia #3, paleta Signal Generation)

> generează un tablou/ array 1D

Sine Pattern.vi

- generează un tablou de **n valori/ samples**
 calculate cu formula:

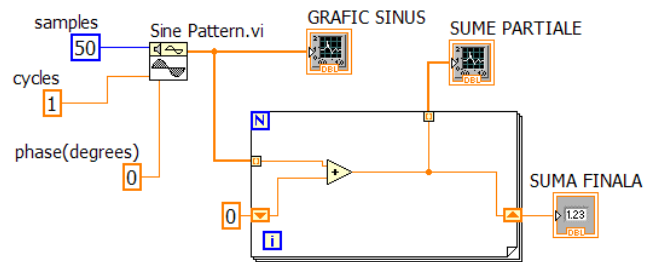
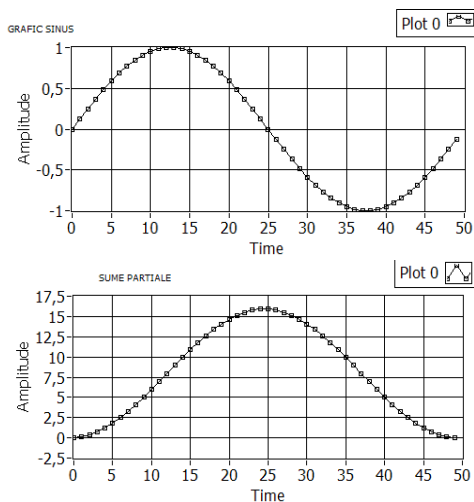
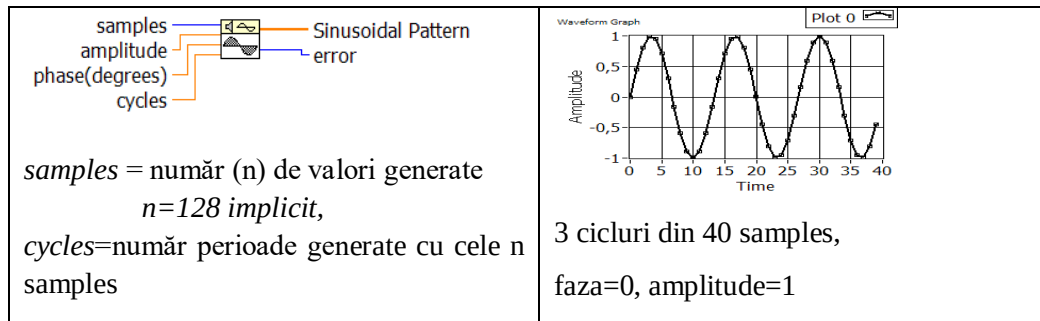
$$y_i = a \sin(x_i), \text{ unde: } i = 0, 1, 2, \dots, n-1$$

$$x_i = \frac{2\pi k}{n} i + \frac{\pi}{180} \varphi_0, \text{ unde } k = \text{cycles}, \varphi_0 = \text{phase [grade]}$$



PROF.dr.ing.Iulian Lupea

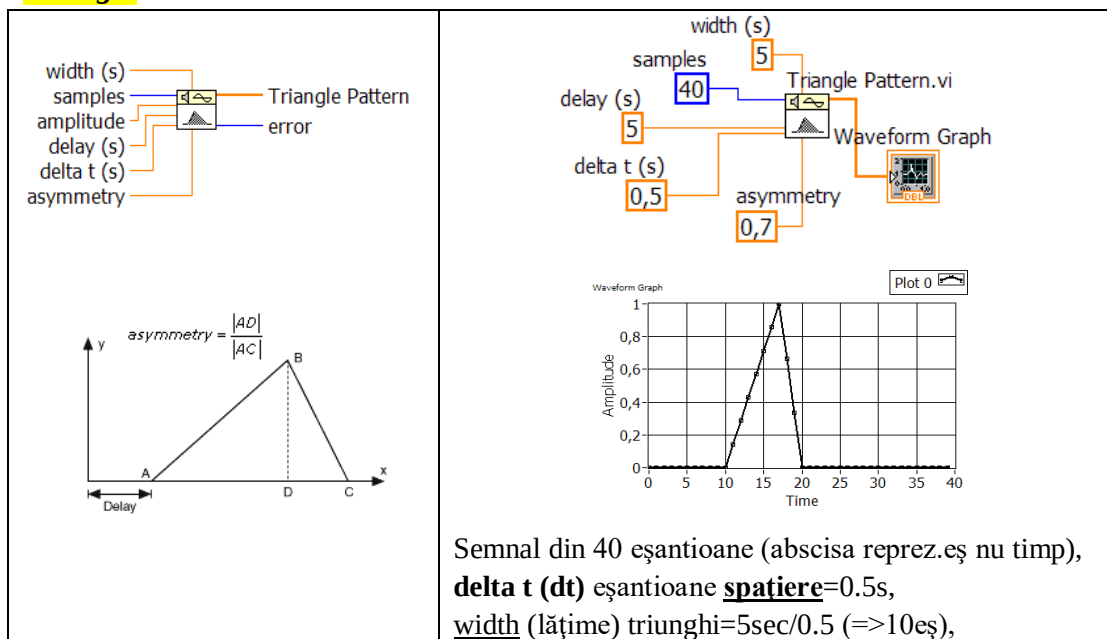
Patterns.vi și Waves.vi



Ciclul FOR, calculează sumele parțiale ale elem. din șirul sinusoidal (Indexare intrare, ieșire)

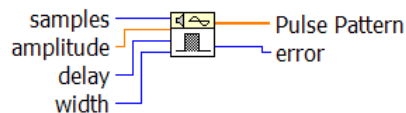
Sumele parțiale cresc cât timp sinusul este pozitiv apoi descresc la 0.

Triangle Patern.vi

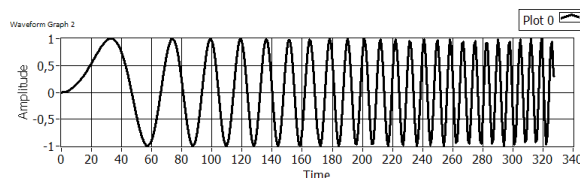


	<u>delay</u> (întârziere) = 5sec/0.5 (10eș); <u>amplitudine</u> 1 implicit.
--	--

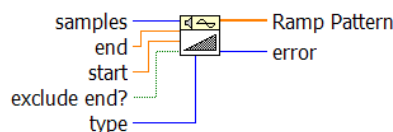
Pulse Pattern.vi



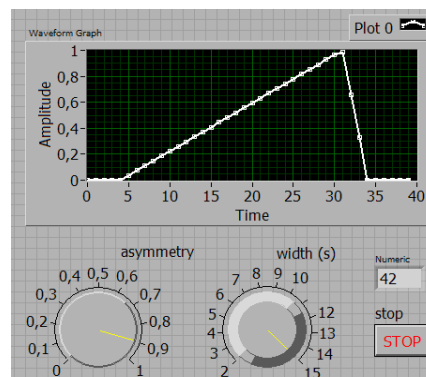
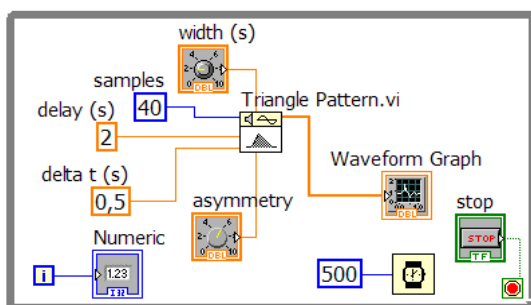
Chirp Pattern.vi (ciripit ... semnal de frecvență variabilă)



Ramp Pattern.vi (semnal tip rampă)



***Aplicație #1** - generarea unui semnal triunghiular la fiecare ciclare



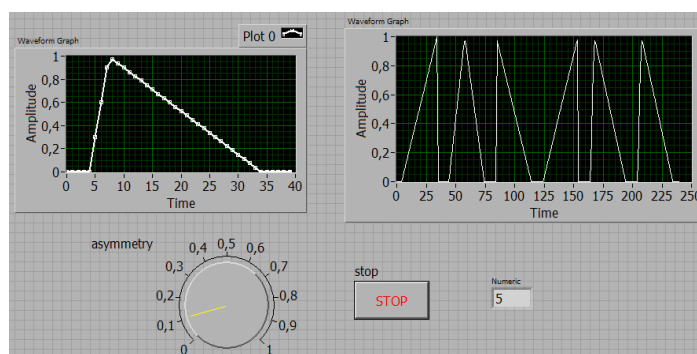
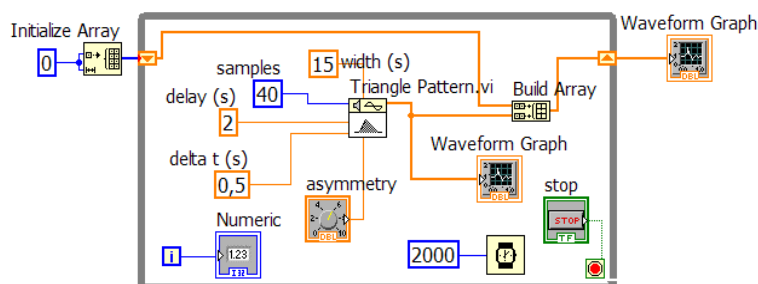
C.While; temporizare; modificare interactivă asimetrie și baza semnal triunghiular

***Aplicație #2:** concatenare de triunghiuri (tablouri reale 1D) în ciclul While

Registrul SHIFT : transferă tablou 1D de valori reale de la o iterație la următoarea

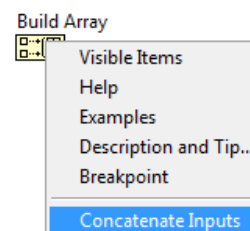
Waveform Graph: 1)afișează triunghiul curent

2)afișează listă triunghiuri după oprirea ciclului While.



Build Array: în varianta concatenare adaugă la tabloul 1D de triunghiuri (n=40 valori) tabloul curent (un triunghi de 40 valori).

Initialize Array: inițializează un tablou 1D de 0 elemente iar tabloul inițializează registrul SHIFT.



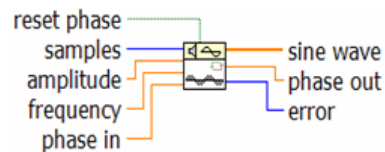
2. Funcții de tip undă: ... wave.vi (linia #4, paleta Signal Generation)

<i>Sine wave.vi</i>	<i>Square wave.vi</i>
<i>Generare (tablou) undă sinusoidală</i>	<i>Gen. (tablou) undă dreptunghiulară</i>
<i>Triangle wave.vi</i>	

-**samples**: impune numărul de eşantioane din întreg semnalul generat la un apel al funcției

Intrarea frequency permite specificarea frecvenței în unități normalizate de tipul **cicluri per eşantion** care ușor poate transformat în: un ciclu / din câte eşantioane; de exemplu pentru:

frequency=0.1 $\Rightarrow$ 1/10 sau un ciclu din 10 eşantioane sau
frequency=0.02 $\Rightarrow$ 2/100 sau 1/50 (un ciclu din 50 eşantioane).



Gestiunea fazei semnalului se realizează din 2 intrări și 1 ieșire.

-intrarea **phase in** impune faza (în grade) a primului eşantion al formei de undă dacă **reset phase=True**

-ieșirea **phase out** indică faza eşantionului următor formei de undă generate (primul eş. din setul următor)

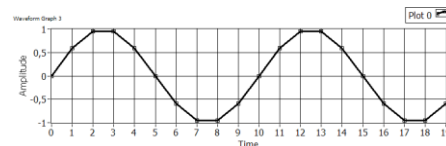
-intrarea **reset phase=False** impune ca faza primului eşantion din apelul următor al funcției să fie faza dată de **phase out** la apelul curent al funcției (**se va realiza un semnal în continuarea semnalului precedent** la apelare repetitivă)

Exemplu: **un apel Sine wave.vi** $\rightarrow \rightarrow$

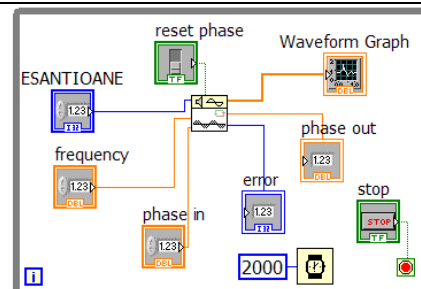
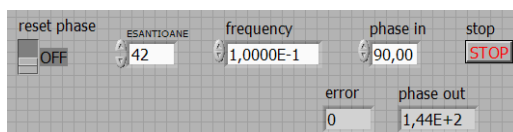
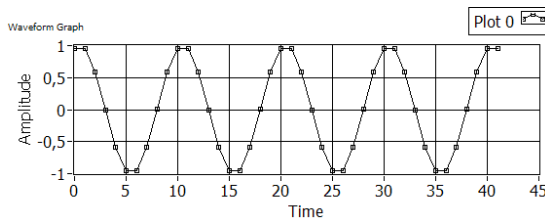
frecvența = 1/10 [cycles/sample] sau: 1 ciclu din 10 samples

Samples=20 $\Rightarrow$ se vor trasa 2=20/10 cicluri

Phase in=0 $\Rightarrow$ sinusul începe cu valoarea zero



Apeluri succesive Sine wave.vi



Fiecare apel generează un semnal care continuă semnalul precedent

Următorul set de **42 eşantioane** va fi în continuarea primului set (după cele două eş. de valoare maximă).

0,1 cicluri/ eşantion *10 $\Rightarrow$ 1 ciclu / 10 eş.,

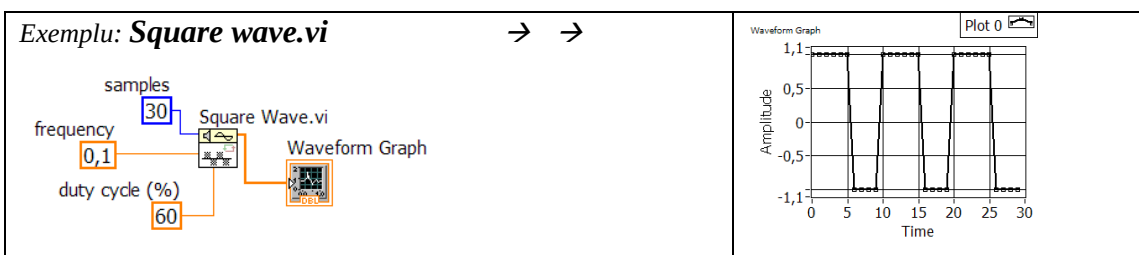
Faza primului set va fi 90 grade (deci începe cu cadranul II) dar la seturile următoare:

* se ține cont de **faza de sfârșit a setului precedent** deoarece reset phase=OFF.

Se folosește waveform graph:

*pentru **reset Phase=OFF** se observă la fiecare afișare eşantionul de început și de sfârșit;

*pentru reset phase=ON fiecare afișare este la fel.



Probleme propuse:

1. Să se pună semnalele generate de 3 apeluri succesive ale funcției *Sine wave.vi* într-un singur tablou și să se afișeze graficul prezentând semnal sinus continuu.

2. Să se apeleze *Sine pattern.vi* și în paralel să se programeze relația de calcul asociată

$y_i = a \sin(x_i)$ unde $x_i = \frac{2\pi k}{n}i + \frac{\pi}{180}\varphi_0$ demonstrând generarea aceluiași semnal (același grafic).

3. Se va calcula suma elementelor din matrice 2D folosind două cicluri While.

4. Se adaugă cod la programele de calcul a sumei (șir/ matrice) pentru calculul mediei aritmetice.

5. Calcul sume/medii pe linii in matrice și afișare șir de sume:

6. Suma elementelor din șir cu While și Stop if true.

7. Generați șir Fibonacci (n=1170, Pisa, Italia) cu 2 registri de transfer. Calculați și afișați rapoartele dintre fiecare număr din șir și numărul precedent (raportul tinde la secțiunea de aur).

8. Se va programa relația de calcul asociată *Chirp Pattern.vi*