

Sisteme înglobate și reconfigurabile programabile cu LabVIEW™

Silviu FOLEA



UTPRESS
Cluj-Napoca, 2024
ISBN 978-606-737-710-1

Silviu FOLEA

**Sisteme înglobate și reconfigurabile
programabile cu LabVIEW™**



UTPRESS

Cluj-Napoca, 2024

ISBN 978-606-737-710-1



Editura UTPRESS
Str. Observatorului nr. 34
400775 Cluj-Napoca
Tel.:0264-401.999
e-mail: utpress@biblio.utcluj.ro
<http://biblioteca.utcluj.ro/editura>

Recenzia: Conf. dr. ing. George MOIȘ
Conf. dr. ing. Roxana RUSU-BOTH

Pregătire format electronic on-line: Gabriela GROZA

Copyright © 2024 Editura UTPRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii UTPRESS.

ISBN 978-606-737-710-1

Cartea se adresează studenților care urmează studiile de master în cadrul secției de Automatică, a Facultății de Automatică și Calculatoare, Universitatea Tehnică din Cluj-Napoca. Parcurgerea lucrărilor presupune cunoașterea mediului de programare LabVIEW™ și are ca scop acumularea de cunoștințe pe baza aplicațiilor prezentate în LabVIEW™ *Real Time* și FPGA (*Field Programmable Gate Array*), implementări care rulează pe echipamentele industriale CompactRIO™ și PXI™. Cartea prezintă 46 de aplicații care să constituie un suport pentru procesul didactic.

O parte din lucrările prezentate au condus la finalizarea unor lucrări de licență, disertație, o teză de doctorat, lucrări științifice publicate și susținute la conferințe internaționale sau în reviste de prestigiu.

Prof. dr. ing. Silviu FOLEA

Cuprins

Introducere	7
1. <i>Benchmark</i> -uri pentru sisteme de calcul industriale	11
2. Testarea performanțelor aplicațiilor de timp real.....	17
3. Implementarea și testarea filtrelor digitale.....	23
4. Controlul turației unui motor de curent continuu.....	31
5. Măsurarea puterii consumate din rețeaua electrică *	39
6. Generatoare de semnale analogice *	47
7. Determinarea capacității energetice a bateriilor.....	53
8. Amortizarea vibrațiilor într-o grindă metalică **	61
9. Controlul intensității luminoase a LED-urilor	69
Bibliografie	73

* lucrarea se poate implementa într-un timp mai mare;

** lucrarea are un grad mai mare de complexitate.

Introducere

Aplicațiile prezentate în această carte au fost realizate ca suport pentru cursurile *Sisteme înglobate* și *Sisteme reconfigurabile*. Scopul principal este cunoașterea echipamentelor CompactRIO™ și PXI™, a componentelor hardware și software pe care se bazează cursurile de master și aplicațiile de laborator.

Se vor implementa programe simple (*VI-uri*) sau proiecte complexe pentru testarea performanțelor aplicațiilor pe aceste sisteme înglobate de timp real sau pe sisteme reconfigurabile hardware de tip FPGA (*Field Programmable Gate Array*). Software-ul utilizat este LabVIEW™ cu modulele RT (*Real Time*) sau FPGA. Compilarea unui program simplu pentru FPGA poate dura între 3 și 5 minute.



Figura I.1 – PXI™, CompactRIO™, TPC, standuri de laborator în 2018

Calculatoarele industriale PXI™ cu sistem de operare Windows sau de timp real Phar Lap ETS sunt cele mai performante sisteme de calcul și achiziție de semnale pentru procese industriale, produse de firma National Instruments™ (NI). În funcție de șasiul ales pot include mai multe module de extensie în aceeași unitate.

Sistemele de tip PXI™ sunt incluse în automatizări pentru procese de fabricație, testare de circuite integrate sau cablaje imprimate, sisteme radio de comunicații, sisteme de navigație, în instalații chimice și termice complexe sau în centrale electrice.

Un sistem industrial tip PXI™ / PXI™ Express are în componență:

- un **controller** - bazat pe cele mai performante procesoare, permite rularea unor sisteme de operare dedicate: Phar Lap ETS, VxWorks, Windows, NI Linux RT și aplicații complexe. Este un sistem de timp real proiectat pentru aplicații în domenii cu cerințe ridicate de performanță.
- un **șasiu** - proiectat pentru mediul industrial, include alimentarea sistemului, magistrala de comunicație între module și sistemul de ventilație. Oferă performanțe și viteze ridicate de transfer a datelor.

- **module de extensie** - există o gamă foarte largă de modele.



Figura I.2 – Controler PXI™, șasiul și magistrala de comunicație între module

Controler programabil pentru automatizare (PAC) - este un sistem reconfigurabil de achiziție de semnale și control, de dimensiuni reduse, destinat aplicațiilor care necesită performanțe ridicate de calcul și fiabilitate.

Un sistem CompactRIO™ de tip PAC are în componență:

- un **controler** - realizat ca modul separat detașabil sau integrat în carcasă, include procesoare pe 32 sau 64 de biți, cu unul sau mai multe nuclee. Are sistem de operare de timp real VxWorks, NI Linux RT; include interfețele de comunicație RS232, RS485, Ethernet.
- un **șasiu** - proiectat pentru mediul industrial, include componenta FPGA programabilă în LabVIEW™ FPGA, utilizată pentru comunicația internă între module. Oferă resurse pentru implementarea unor algoritmi de tip PID, funcții de procesare de semnale, FFT (*Fast Fourier Transform*) sau de generare de semnale direct în circuitul reconfigurabil FPGA.
- **module de extensie** - există o gamă foarte largă de modele care oferă performanțe ridicate de achiziție sau generare de semnale.



Figura I.3 – CompactRIO™, controler programabil pentru automatizare

Pentru vizualizarea locală a datelor de la sistemele de tip PAC, se poate utiliza un monitor sau un echipament de tip TPC (*Touch Panel Computer*) programabil din LabVIEW™. Comunicația între un TPC și sistemele de tip PAC se face prin interfața Ethernet.

Anumite modulele de extensie din seria C, de exemplu NI 9211, pot fi utilizate separat ca plăci de achiziție de semnale pe portul USB al unui calculator,

prin intermediul dispozitivului **NI USB-9162**, *USB C Series Single Module Carrier*, oferind astfel o maximă flexibilitate și portabilitate.

Pentru desfășurarea orelor de laborator, există următoarele echipamente:

- **3 calculatoare industriale de tip PXI™ sau PXI™ Express**, cu sistem de operare de timp real Phar Lap ETS, care pot rula și Windows 10, cu următoarele configurații:

A. **controler** - bazat pe următoarele procesoare:

- **NI PXI-8106** cu procesor Core 2 Duo, T7400 la 2.16 GHz, 2 GB DDR II la 667 MHz [1];
- **NI PXI-8105** cu procesor Dual-Core, T2500 la 2.0 Ghz, 2 GB DDR II la 667 MHz [2];
- **NI PXIe-8106** cu procesor Core 2 Duo, T7400 la 2.16 GHz, 1 GB DDR II la 667 MHz [3];

- controlerul include diferite interfețe: Ethernet 10/100/1000 BaseTX, ExpressCard/34, USB, GPIB, serial, DVI și IEEE 1284 ECP/EPP;

B. **șasiu** care include sursa de alimentare de la rețeaua de tensiune de 230 Vac:

- **NI PXI-1031**, cu 4 sloturi 3U PXI;
- **NI PXI-1042**, cu 8 sloturi 3U PXI;
- **NI PXIe-1062Q**, cu 8 sloturi 3U PXI Express;

C. **module de extensie** – pentru achiziția de semnalele analogice sau digitale:

- **NI PXI-6281**, 16 AI (*intrări analogice*), 625 kS/s, 18 biți, ± 0.1 V, ± 0.2 V, ± 0.5 V, ± 1 V, ± 2 V, ± 5 V, ± 10 V; 2 AO (*ieșiri analogice*), 2.8 MS/s, 16 biți, ± 1 V, ± 2 V, ± 5 V, ± 10 V; 24 DIO (*intrări sau ieșiri digitale*) și 2 CNT (*numărătoare*) pe 32 biți, 80 MHz [4];
- **NI PXI-6280**, 16 AI, 625 kS/s, 18 biți, similar cu NI PXI-6281; 24 DIO și 2 CNT pe 32 biți, 80 MHz [5];
- **NI PXI-6733**, 8 AO, 1.0 MS/s, 16 biți, ± 10 V; 8 linii DIO și 2 CNT pe 24 biți [6];
- **NI PXI-7811R**, 160 DIO reprogramabile cu FPGA de tip Virtex II, programabil cu LabVIEW™ FPGA [7];
- **NI PXI-4351**, pentru măsurări de temperatură; 16 AI diferențiale pe 24 biți sau 14 intrări pentru termocuple; 60 S/s, cu amplificare reglabilă și generatoare de curent integrate de 25 μ A și 1 mA; precizie ridicată: 0.42 °C pentru termocuple de tip J, 0.03 °C pentru termistoare și 0.12 °C pentru termorezistente; include reglarea automată de zero și compensarea joncțiunii reci pentru termocuple [8];
- **NI PXI-5114**, osciloscop, 2 canale cu eșantionare și conversie simultană, 250 MS/s, 8 biți, lățime de bandă 125 MHz. Domeniul de măsură este de la 40 mV la 40 V, iar memoria de stocare este de la 8 MB la 256 MB pe fiecare canal [9];

-
- modul de conectare TBX-68T și 3 module de conectare CB-68LP.

- **7 sisteme de control** de tip CompactRIO™ cu sistem de operare de timp real NI Linux RT și șasiuri cu FPGA, programabile cu LabVIEW™ RT și FPGA, cu următoarele configurații:

A. **controler** – bazat pe procesoare cu două nuclee:

- **NI cRIO-9040**, controler de timp real pe 64 de biți, procesor Intel Atom E3930 *dual core* la 1.3 GHz (1.8 GHz burst), cu 2 GB DDR3L și 4 GB memorie NAND pentru stocare [10];
- **NI cRIO-9053**, controler de timp real pe 64 de biți, procesor Intel Atom E3805 *dual core* la 1.33 GHz, cu 1 GB DDR3L și 4 GB memorie NAND pentru stocare [11];

- controlerul include diferite interfețe: RS232, RS485, 2x Ethernet, 2x USB-C host și slave, suport pentru card SDHC;

B. **șasiu** – include componenta FPGA care asigură și comunicația internă cu modulele:

- FPGA Xilinx **Kintex-7 7K70T**, asigură 4 sloturi de extensie;
- FPGA Xilinx **Artix-7 A50T**, asigură 4 sloturi de extensie;

C. **module de extensie** din seria **C** oferă o gamă diversă de modele:

- **NI 9239** – 2 module, 4 AI simultane, 50 kS/s, 24 biți, ±10 V [12];
- **NI 9381** – 4 module, 8 AI, 20.0 kS/s, 12 biți, 0-5 V; 8 AO, 20 kS/s, 12 biți, 0-5 V; 4 DIO tip LVTTTL, 1 MHz [13];
- **NI 9211** – 2 module, 4 AI, 14 S/s, 24 biți, ±80 mV pentru termocuple și compensarea joncțiunii reci [14];
- **NI 9215** – 2 module, 4 AI simultane, 100 kS/s, 16 biți, ±10 V [15];
- **NI 9201**, 8 AI, 500 kS/s, 12 biți, ±10 V [16];
- **NI 9401** – 2 module, 8 DIO, tip TTL de 5 V, 5-30 MHz [17];
- **NI 9403**, 32 DIO, tip TTL de 5 V, 3-10 MHz [18];
- **NI 9263** – 4 module, 4 AO, 100 kS/s, 16 biți, ±10 V [19];
- **NI 9230**, 3 AI, 12.8 kS/s, 24 biți, ±30 V pentru achiziția de sunet și vibrații; suport pentru TEDS (IEEE 1451) [20].

- 3 module de adaptare **NI USB-9162** pentru conectarea modulelor de extensie direct la portul USB al unui calculator [21].

- **1 kit NI ELVIS II+**, „*Instrumentation, Prototyping, and Teaching Platform for Labs*” pentru realizarea de prototipuri electronice. Include osciloscop cu 2 canale la 100 MHz, surse generatoare de semnal și de alimentare programabile etc. Permite atașarea unor plăci de extensie [22].

1. Benchmark-uri pentru sisteme de calcul industriale

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ RT și FPGA pentru testarea performanțelor de calcul [23, 24] pe echipamentele CompactRIO™ și PXI™, sisteme de timp real și reconfigurabile hardware pe baza FPGA-urilor incluse.

Se creează un proiect care include VI-urile de test (*programe în LabVIEW™*) care să permită determinarea timpului de calcul al unei expresii matematice, pentru fiecare echipament inclus în proiect: CompactRIO™, PXI™ și calculator. Pentru acest test se implementează relația următoare (*utilizată în aplicațiile de măsurare, pentru scalarea mărimilor de intrare*):

$$Y = A \cdot X + B, \text{ unde: } A = 0.5, B = 0.3 \text{ și } X = 0.4.$$

Testati ce se întâmplă pe fiecare sistem atunci când VI-ul include mai multe bucle *For* care se execută în paralel. Realizați un tabel comparativ:

100 M de execuții	1 buclă	2 bucle	3 bucle	4 bucle	5 bucle	Observații
	<i>For</i> T _{ex} [ms]	<i>For</i> T _{ex} [ms]	<i>For</i> T _{ex} [ms]	<i>For</i> T _{ex} [ms]	<i>For</i> T _{ex} [ms]	
1. Calculator, i7-11800H, 8x2.3 GHz, Windows 11						- reprezentare <i>DBL</i> - funcții de timp cu rezoluție de ms
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT						- reprezentare <i>DBL</i> - rezoluție de ms, μs
3. cRIO, FPGA 40 - 200 MHz - nu are sistem de operare						- reprez. <i>FXP, I32</i> - rezoluție de ms, μs , ceas
4. PXI™, Core 2 Duo 2.2 GHz, Phar Lap ETS						- reprezentare <i>DBL</i> - rezoluție de ms, μs

Modificați setul anterior de VI-uri de test prin înlocuirea buclei *For* cu *While*. Se vor nota diferențele între timpii de execuție (T_{minim} și T_{maxim}) ai programelor pe fiecare sistem de calcul. Realizați un tabel comparativ:

100 M de execuții	1 buclă	2 bucle	3 bucle	4 bucle	5 bucle	Observații
	<i>While</i> T _{ex} [ms]	<i>While</i> T _{ex} [ms]	<i>While</i> T _{ex} [ms]	<i>While</i> T _{ex} [ms]	<i>While</i> T _{ex} [ms]	
1. Calculator, i7-11800H, 8x2.3 GHz, Windows 11	154 157	133 159	135 139	136 144	137 158	- reprezentare <i>DBL</i> - funcții de timp cu rezoluție de ms
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	2285 2549	3420 4169	5131 5978	7068 8276	8763 10214	- reprezentare <i>DBL</i> - rezoluție de ms, μs
3. cRIO, FPGA 40 - 200 MHz - nu are sistem de operare						- reprez. <i>FXP, I32</i> - rezoluție de ms, μs , ceas
4. PXI™, Core 2 Duo 2.2 GHz, Phar Lap ETS	1066 1067	977 978	1399 1407	1913 1921	2369 2387	- reprezentare <i>DBL</i> - rezoluție de ms, μs

Aplicația are o buclă principală *For* care se execută de 100 de milioane de ori (*100 M*). Se determină timpul de execuție atunci când se lucrează cu valori numerice reale reprezentate în dublă precizie (*DBL*) sau cu valori numerice în virgulă fixă (*FXP*) în cazul dispozitivelor FPGA incluse.

O variantă de implementare cu buclă *While* este prezentată în figurile următoare. Sunt prezentate tabele cu rezultate pentru o parte din implementări, nefiind incluse toate cazurile cerute anterior.

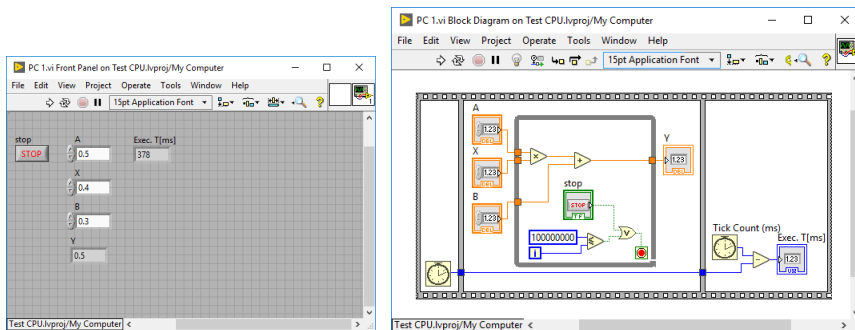


Figura 1.1 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe calculator (1 buclă *While*)

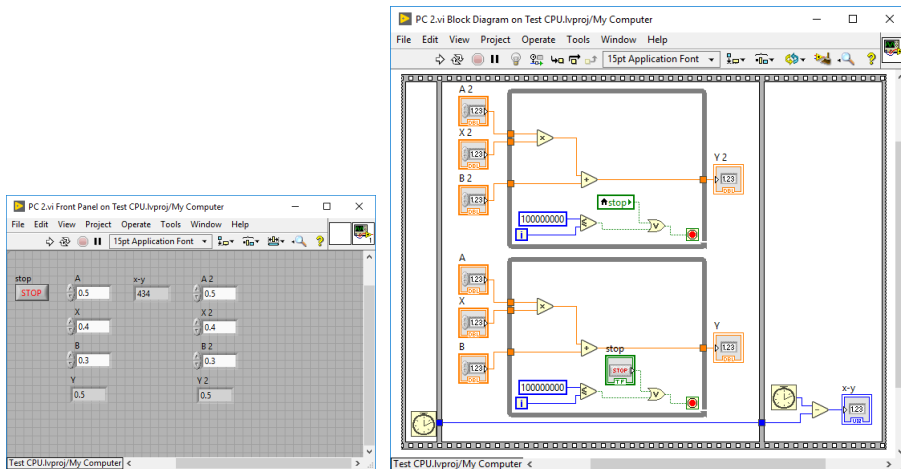


Figura 1.2 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe calculator (2 bucle *While*)

Includeți codul de test într-o buclă *For* care se execută de 5 ori și afișați datele pe un indicator numeric de tip *Array*. Exemplul prezentat este pentru varianta de implementare cu bucla *For*. Se observă că prima valoare de timp este mai mare decât următoarele. **Justificați de ce apare această diferență?**

Pentru a simplifica multiplicarea buclelor, în locul variabilelor de control pentru A, B și X se pot utiliza **constante**. Alternativ puteți configura aceste

controale accesând din meniu opțiunea care permite setarea unei valori introduse în interfață, ca și valoare implicită la inițializare (*default*): *Data Operation -> Make Current Value Default*.

Testați dacă timpul de execuție este diferit atunci când se utilizează reprezentări diferite ale datelor (*I32*) sau datele introduse au valoarea zero.

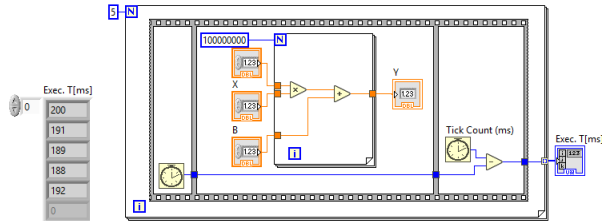


Figura 1.3 – Diagrama bloc, codul de test se execută de 5 ori succesiv

Rezultatele obținute pentru VI-urile care rulează pe calculator de 100 de milioane de ori sunt prezentate în tabelul următor:

100 M de execuții	T_{ex} [ms]	T_{ex} [ms]	T_{ex} [ms]	Observații
	1 buclă <i>While</i>	2 bucle <i>While</i>	3 bucle <i>While</i>	
1. Calculator, i7-11800H, 8x2.3 GHz, Windows 11	154	132	135	- reprezentare date <i>DBL</i> - rezoluție de ms
	160	139	139	
1. Calculator, i7-11800H, 8x2.3 GHz, Windows 11	96	92	93	- varianta fără buton de STOP în bucla <i>While</i> .
	100	95	97	

Performantele de calcul sunt mai bune dacă se elimină din buclă citirea butonului de STOP.

Rezultatele obținute pentru VI-urile care rulează pe CompactRIO™ RT de 100 de milioane de ori sunt prezentate în tabelul următor:

100 M de execuții	T_{ex} [ms]	T_{ex} [ms]	T_{ex} [ms]	Observații
	1 buclă <i>While</i>	2 bucle <i>While</i>	3 bucle <i>While</i>	
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	2195	3363	5158	- reprezentare date <i>DBL</i> - rezoluție de ms
	2396	4268	5923	
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	-	3681 4462	5578 6504	- varianta cu variabila locală pentru butonul de STOP
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	804	1567	2129	- varianta fără buton de STOP în bucla <i>While</i> .
	873	1735	2644	

Rezultatele obținute confirmă ca utilizarea variabilelor locale presupune un efort de calcul mai mare, timpul de execuție fiind cu până la 5-10% mai mare.

Rezultatele obținute pentru VI-urile care rulează pe FPGA, de 100 de milioane de ori sunt prezentate în tabelul următor. În interfața cu utilizatorul rezultatul este afișat în microsecunde.

100 M de execuții	T_{ex} [ms] 1 buclă While	T_{ex} [ms] 2 bucle While	T_{ex} [ms] 6 bucle While	Observații
3. cRIO, FPGA 40 - 200 MHz - nu are sistem de operare	10000	10000	10000	- reprezentare date <i>FXP</i> - rezoluție de μs
3. cRIO, FPGA 40 - 200 MHz - nu are sistem de operare	7500	7500	7500	- varianta care include constante în loc de controale în interfață

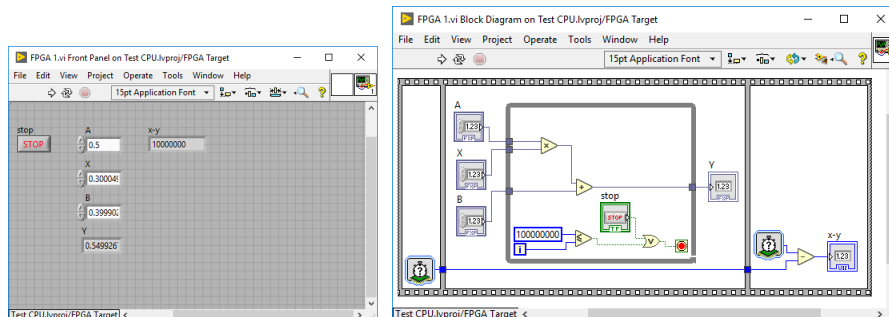


Figura 1.4 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (1 buclă *While*)

Funcțiile de înmulțire reprezentate ca un dreptunghi roz sunt special implementate și optimizate pentru performanță (*High Throughput Math*) inclusiv prin utilizarea DSP-urilor (*Digital Signal Processors*) incluse în FPGA.

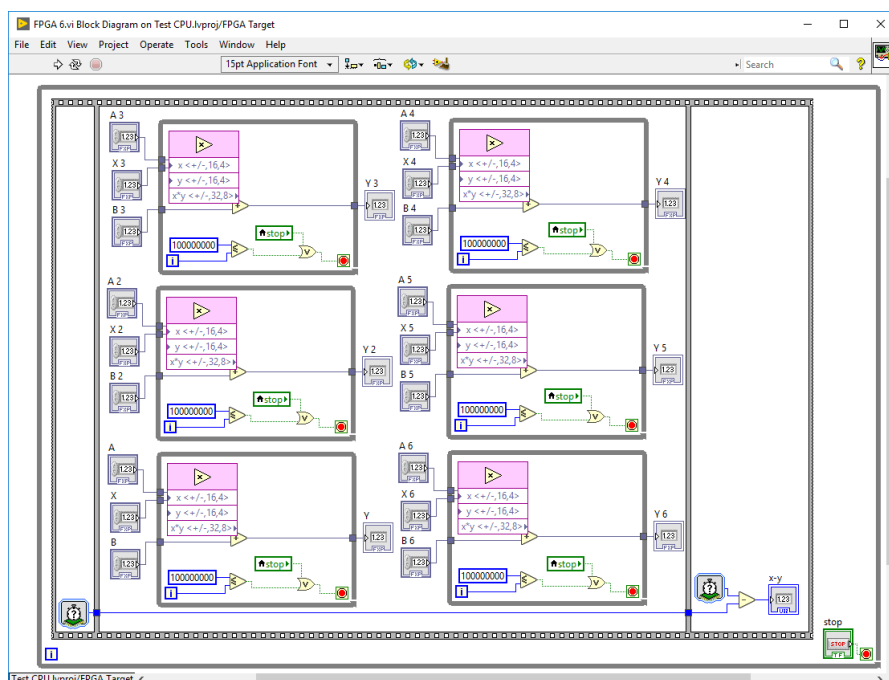


Figura 1.5 – Diagrama bloc, VI-ul care rulează pe FPGA (6 bucle *While*)

Se observă că inclusiv la 6 bucle *While* care rulează real în paralel se obține același rezultat. Numărul de bucle pe care le poate rula FPGA-ul în paralel este

limitat doar de resursele fizice existente, calculele se efectuează în virgulă fixă, datele sunt notate cu *FXP* în LabVIEW™ FPGA.

☺ Modificați în proiect frecvența circuitului FPGA la una din frecvențele următoare 80 MHz, 120 MHz, 160 MHz sau 200 MHz și rulați unul din programele implementate pentru a compara timpul de execuție.

☺ Testați cât de multe bucle puteți include într-o aplicație care rulează în FPGA, până când compilatorul returnează eroare, fiind depășit numărul de componente disponibile.

Resursele consumate pentru 64 de bucle care rulează în paralel în FPGA sunt prezentate în tabelul următor:

Total Slices: 45.0% (4608 out of 10250)	Time compiling: 04:02
Slice Registers: 14.9% (12191 out of 82000)	- Generate Xilinx IP: 00:00
Slice LUTs: 27.6% (11303 out of 41000)	- Synthesize - Vivado: 01:55
Block RAMs: 2.2% (3 out of 135)	- Optimize Logic: 00:12
DSP48s: 0.0% (0 out of 240)	- Place: 00:44
	- Optimize Timing: 00:10
	- Route: 00:34
	- Generate Programming File: 00:20

Rezultatele obținute pentru VI-urile care rulează pe calculatorul PXI™ de 100 de milioane de ori sunt prezentate în tabelul următor.

100 M de execuții	T_{ex} [ms]	T_{ex} [ms]	T_{ex} [ms]	Observații
	1 buclă <i>While</i>	2 bucle <i>While</i>	3 bucle <i>While</i>	
4. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>	833	1024	1627	- <i>DBL</i> - rezoluție de μs
4. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>	834	930	1351	- varianta fără buton de STOP în bucla <i>While</i> .

Performanțele sistemelor de calcul cu procesoare din generații recente sunt mai rapide decât cele din generațiile anterioare. Pe calculatorul cu sistem de operare Windows 10 sau 11, în *Task Manager - Performance* se poate urmări creșterea frecvenței procesorului, care se modifică de la 2.3 GHz la 4.3 GHz cât timp au rulat testele.

2. Testarea performanțelor aplicațiilor de timp real

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ RT și FPGA pentru testarea performanțelor de timp real și măsurarea variațiilor timpului de execuție a codului (*jitter*), pe echipamentele CompactRIO™ și PXI™, cu sistemele de operare de timp real NI Linux RT și Phar Lap ETS, dar și aplicații LabVIEW™ care rulează pe calculatoare cu Windows 10 sau 11.

Se creează un proiect care include VI-urile de test care să permită determinarea variației timpului de execuție pentru calculul unei expresii matematice, pe fiecare sistem inclus în proiect: CompactRIO™, PXI™ și calculator. Pentru acest test se implementează relația următoare, codul testat fiind inclus într-o buclă *For* care se execută de 100.000 de ori:

$$Y = A^9 + B^8 + X^7, \text{ unde: } A = 0.5, B = 0.3 \text{ și } X = 0.4.$$

Se determină timpul de execuție (T_{ex}) al fiecărei bucle de 1000 de ori, se reprezintă grafic și se realizează o histogramă a timpului de execuție.

Determinați timpul de execuție al VI-ului pe fiecare din sistemele incluse în proiect atunci când în loc de bucla *While* care include programul principal de testare, se utilizează *For* sau *Timed Loop*. Realizați un tabel comparativ:

1000 de execuții	T_{ex} [ms] bucula <i>For</i> , <i>While</i> , <i>Timed Loop</i>	Variația T_{ex} [ms]	Observații
1. Calculator, i7-11800H, 8x2.3 GHz, <i>Windows 11</i>			- funcții de timp cu rezoluție de ms
2. cRIO, Atom E3930 2x1.3 GHz, <i>NI Linux RT</i>			- rezoluție de ms, μs
3. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>			- rezoluție de ms, μs

O variantă de implementare cu buclă *While* este prezentată în figurile următoare. Sunt prezentate tabele orientative cu rezultate pentru o parte din implementări, fără a fi testate toate cazurile cerute.

În teste se pot elimina primele 100 de valori din vectorul timp (10%), uneori aceste valori sunt afectate de alocarea resurselor și nu sunt reprezentative pentru testele de performanță.

Rezultatele obținute pentru prima variantă de VI care rulează pe calculator sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] bucula <i>While</i>	Variația T_{ex} [ms]	Observații
1. Calculator, i7-11800H, 8x2.3 GHz, <i>Windows 11</i>	14 (748)	13-16	- rezoluție de ms

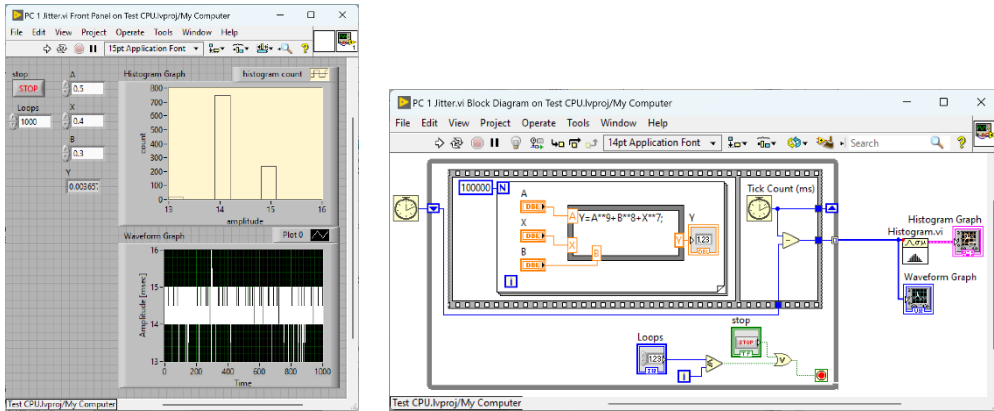


Figura 2.1 – Interfața cu utilizatorul și diagrama bloc, VI-ul rulează pe calculator (varianta 1)

Pentru că în prima variantă de implementare, în bucla *While* se utilizează variabilele de control *Loops* și *stop*, iar în bucla *For* variabilele de control A, B, X și indicatorul Y, acestea vor influența negativ timpul de execuție a VI-urilor care se rulează pe CompactRIO™ și PXI™. Echipamentele transferă datele pentru afișare prin Ethernet către *Front Panel-ul* VI-ului care se execută pe calculator.

În varianta a doua de implementare butonul de *stop* a fost eliminat din buclă.

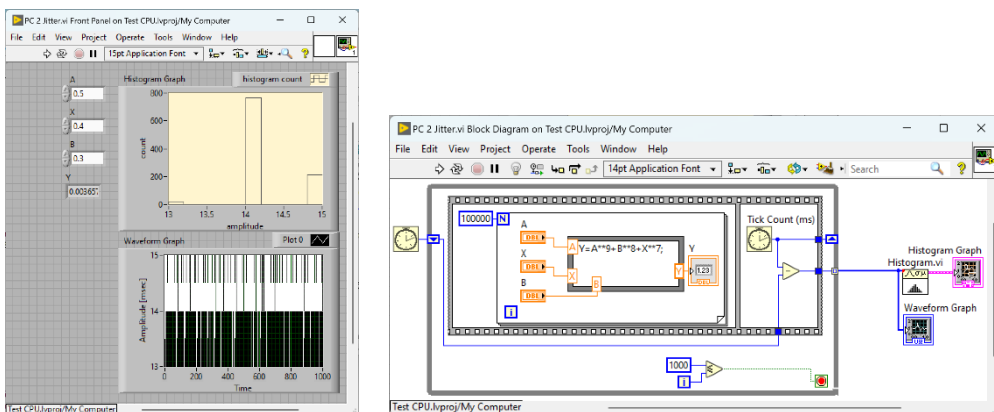


Figura 2.2 – Interfața cu utilizatorul și diagrama bloc, VI-ul rulează pe calculator (varianta 2)

În varianta a doua de implementare nu se observă o reducere semnificativă a timpului de execuție. Rezultatele obținute pentru varianta a doua de VI care rulează pe calculator sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] bucla <i>While</i>	Variația T_{ex} [ms]	Observații
1. Calculator, i7-11800H, 8x2.3 GHz, Windows 11	14 (765)	13-15	- rezoluție de ms

Pentru testarea pe CompactRIO™ RT, funcțiile de timp *Tick Count* au fost configurate cu rezoluție de μs pe 32 de biți (paleta pentru *Real Time – RT Timing*).

Rezultatele obținute pentru prima variantă de VI care rulează pe CompactRIO™ RT sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] bucla <i>While</i>	Variația T_{ex} [ms]	Observații
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	100 (825)	100-142	- funcții de timp cu rezoluție de μs

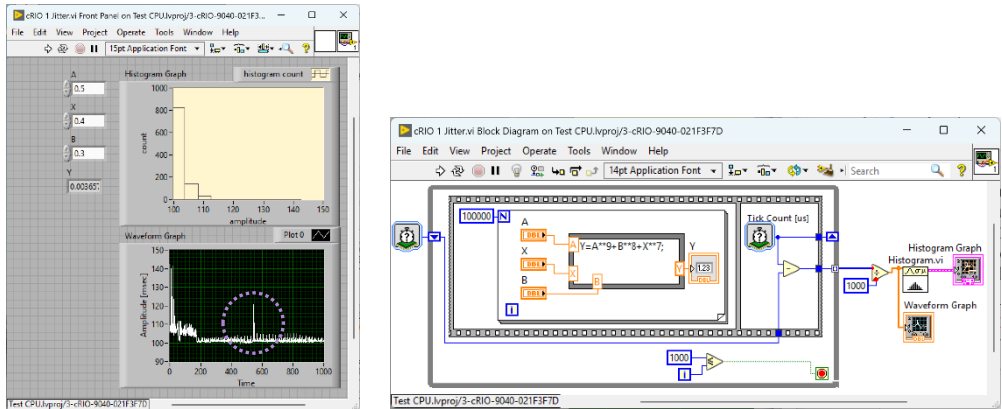


Figura 2.3 – Interfața cu utilizatorul și diagrama bloc, VI-ul rulează pe CompactRIO™ (var. 1)

În varianta a doua de implementare se înlocuiesc în buclă variabilele de control prin constante și se elimină actualizarea valorilor în indicator.

Rezultatele obținute pentru varianta a doua de VI care rulează pe CompactRIO™ sunt prezentate în tabelul următor. Se observă o reducere a timpului de execuție și un interval mai mic de variație.

1000 de execuții	T_{ex} [ms] bucla <i>While</i>	Variația T_{ex} [ms]	Observații
2. cRIO, Atom E3930 2x1.3 GHz, NI Linux RT	80 (739)	80-103	- funcții de timp cu rezoluție de μs

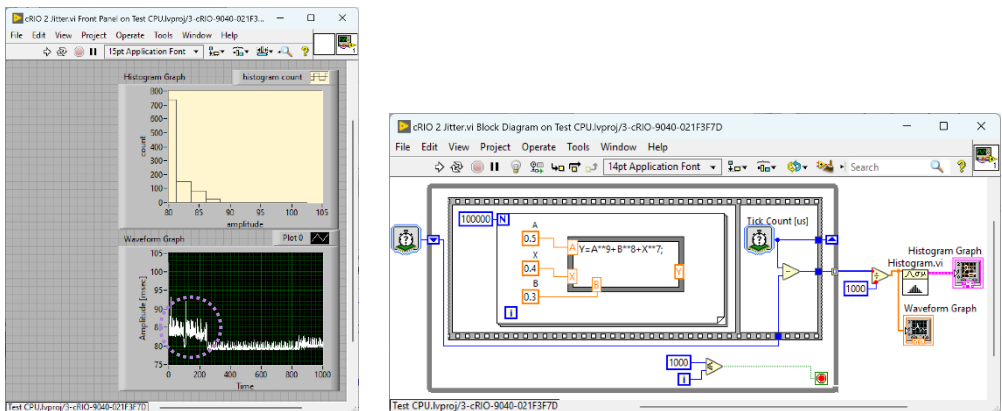


Figura 2.4 – Interfața cu utilizatorul și diagrama bloc, VI-ul rulează pe CompactRIO™ (var. 2)

Pentru testarea pe PXI™, funcțiile de timp *Tick Count* au fost configurate cu rezoluție de μs pe 32 de biți. Rezultatele obținute pentru prima variantă de VI care rulează pe PXI™ sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] bucla <i>While</i>	Variația T_{ex} [ms]	Observații
3. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>	77.6 (279)	77.55-78.25	- funcții de timp cu rezoluție de μs

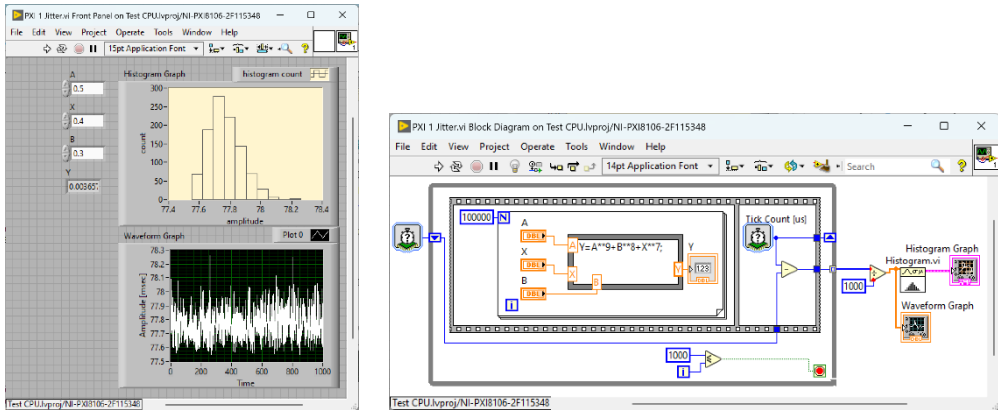


Figura 2.5 – Interfața cu utilizatorul și diagrama bloc, VI-ul rulează pe PXI™ (varianta 1)

În varianta a doua de implementare se înlocuiește bucla *While* cu o buclă temporizată *Timed Loop* care a fost configurată să se execute la 80 ms.

Rezultatele obținute pentru varianta a doua de VI care rulează pe PXI™ sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] <i>Timed Loop</i>	Variația T_{ex} [ms]	Observații
3. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>	80.0 (410)	78.6-81.5	- funcții de timp cu rezoluție de μs

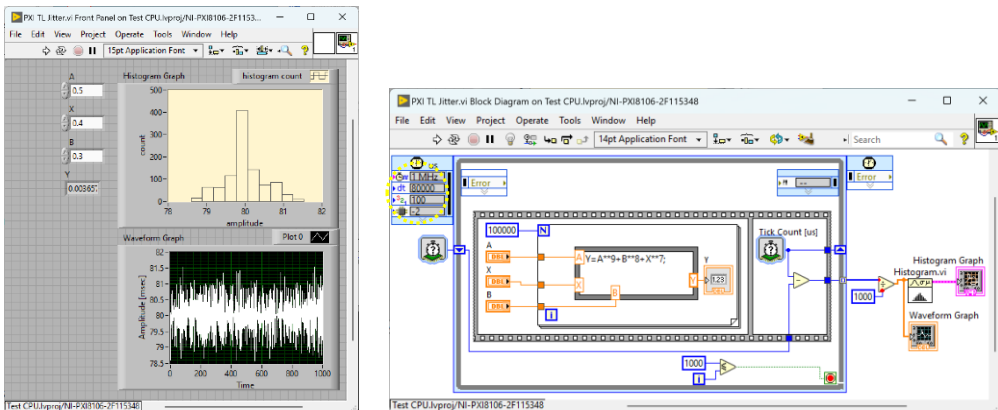


Figura 2.6 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe PXI™ (varianta 2)

Pentru structura *Timed Loop* fereastra de configurare permite alegerea unui ceas de sincronizare de 1 kHz sau 1 MHz (caz în care se pot alege perioade de execuție de până la 1 μ s). Se poate alege prioritatea task-ului și un anumit nucleu din procesor pe care să se execute VI-ul, iar distribuirea se poate face automat de către sistemul de operare sau manual de către utilizator.

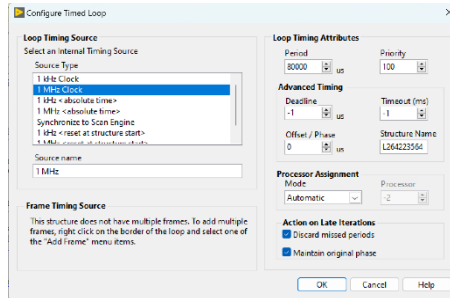


Figura 2.7 – Fereastra de configurare pentru structura *Timed Loop*

Timpul de repetare al unei bucle trebuie să fie mai mare decât timpul de execuție al codului, dacă se dorește respectarea constrângerilor de timp. Dacă se alege un timp de execuție al buclei mai mic decât timpul de rulare atunci se obține de fapt timpul minim de execuție al buclei.

În varianta a treia de implementare se înlocuiesc variabilele de control prin constante și se elimină indicatoarele din buclă.

Rezultatele obținute pentru varianta a treia de VI care rulează pe PXI™ sunt prezentate în tabelul următor. Se observă o variație mică de ~100 μ s a timpului de execuție. Cea mai mare valoare de timp apare la pornirea programului.

1000 de execuții	T_{ex} [ms] <i>Timed Loop</i>	Variația T_{ex} [ms]	Observații
3. PXI™, Core 2 Duo 2.2 GHz, Phar Lap ETS	33.6 (667)	33.6-33.7	- funcții de timp cu rezoluție de μ s

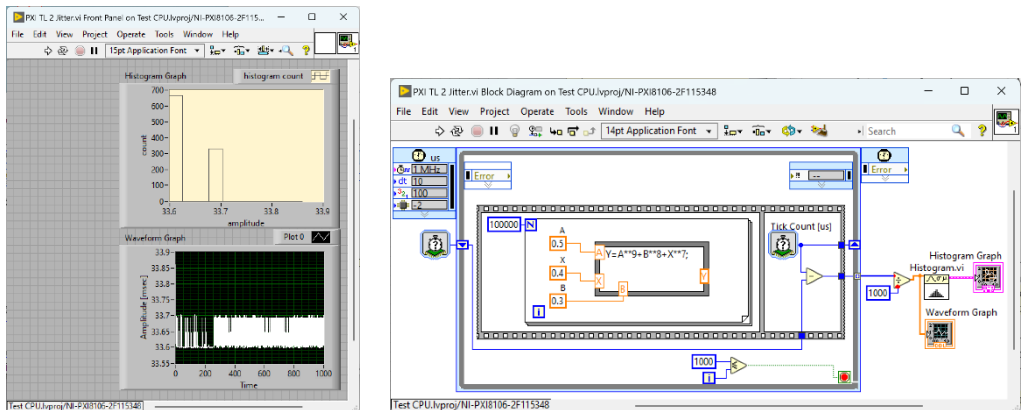


Figura 2.8 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe PXI™ (varianta 3)

Comparativ cu varianta a treia de program, în varianta a patra se implementează VI-ul cu buclă *While*.

Timpul de execuție al VI-ului cu buclă *While* comparativ cu *Timed Loop* este puțin mai mare și variația timpului de execuție este mai mare.

Rezultatele obținute pentru varianta a patra de VI care rulează pe PXI™ sunt prezentate în tabelul următor:

1000 de execuții	T_{ex} [ms] bucla <i>While</i>	Variația T_{ex} [ms]	Observații
3. PXI™, Core 2 Duo 2.2 GHz, <i>Phar Lap ETS</i>	33.8 (365)	33.77-33.92	- funcții de timp cu rezoluție de μs

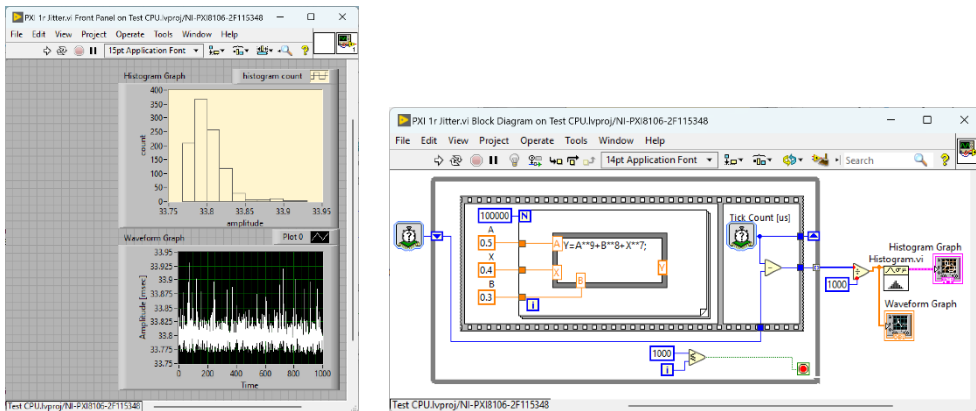


Figura 2.9 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe PXI™ (varianta 4)

Dacă analizăm rezultatele obținute pentru timpul de execuție pe platforma PXI™, se observă că timpul de execuție este mai puțin influențat de implementare, fie că se utilizează variabile de control sau constante, incluse sau nu în bucla de testare. Dacă conectați un monitor la PXI™ prin conectorul DVI-I se poate vizualiza încărcarea procesorului în interfața afișată.

Pe platforma CompactRIO™ timpul de execuție este influențat de implementare.

Calculatorul este mai rapid decât cele două platforme industriale testate, dar are variațiile cele mai mari ale timpului de execuție.

Pornirea sau oprirea unor programe pe calculatorul cu sistem de operare Windows poate duce la modificarea timpului de execuție și la obținerea unui interval mai mare de variație care se poate evidenția în timpul testelor realizate.

3. Implementarea și testarea filtrelor digitale

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ cu modulele RT și FPGA care să ruleze pe echipamentul CompactRIO™, de filtrare a semnalelor audio. Realizarea unor programe pentru testarea performanțelor de filtrare și comanda LED-urilor RGB (*Red, Green, Blue*) de putere utilizând un modul dedicat (*driver*). Implementarea unei aplicații care modulează lumina după muzică (*orgă de lumini*) și rulează integral pe CompactRIO™.

Se creează un proiect care include calculatorul și CompactRIO™, fiecare având câte un VI atașat. Cele două VI-uri vor comunica între ele prin variabile partajate de tip *Network-Published* prin rețeaua Ethernet.

- VI-ul de pe calculator va realiza achiziția de semnal de la microfonul atașat la calculator și procesarea semnalului cu trei filtre IIR în topologie *Butterworth* de ordinul 3, dar se pot face teste și cu alte opțiuni;
- VI-ul de pe CompactRIO™ RT citește din rețeaua Ethernet datele pentru cele 3 canale de sunet obținute prin filtrare și în funcție de pragurile setate de utilizator, va comanda sau nu aprinderea LED-urilor RGB;
- VI-ul de pe CompactRIO™ FPGA trimite comanda *ON / OFF* către *driver*-ul LED-urile RGB de putere.

ATENȚIE: evitați să priviți direct spre LED-uri, pentru că luminează intens!

ATENȚIE: standul funcționează cu tensiune de +5 V! Nu alimentați modulul de comandă a LED-urilor cu tensiune mai mare, nu are protecție la supratensiune.

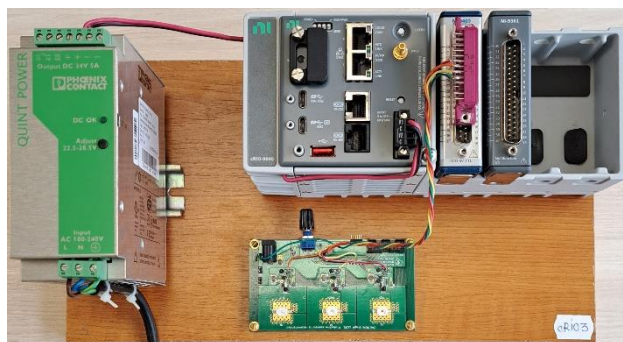


Figura 3.1 – Standul pentru comanda LED-urilor RGB

Pentru comanda LED-urilor se utilizează un kit de evaluare cu LED-uri RGB de putere, de la Texas Instruments, modelul TPS62260LED-338 [25].

O variantă de implementare a VI-urilor este prezentată în figurile următoare. Se utilizează un microfon cu *jack stereo* de 3 mm conectat la calculator, configurat pentru sensibilitate maximă (+30 dB), în scopul de a obține un semnal audio de amplitudine cât mai mare.

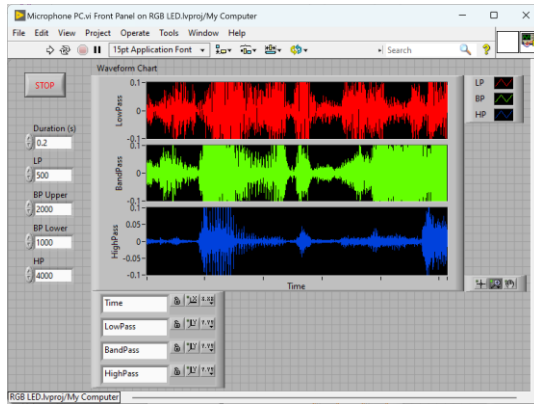


Figura 3.2 – Interfața cu utilizatorul, VI-ul care rulează pe calculator

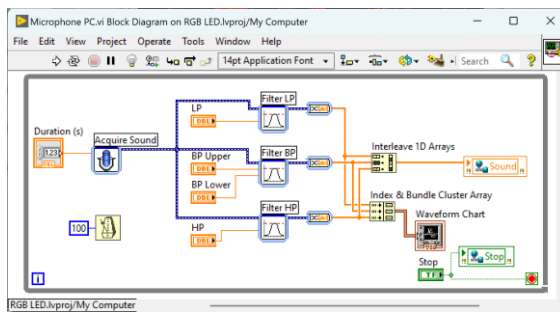


Figura 3.3 – Diagrama bloc, VI-ul care rulează pe calculator

Se utilizează trei tipuri de filtre, configurate cu următoarele valori:

- $F.LP$ – filtru trece jos, cu frecvența de tăiere de 500 Hz;
- $F.BP$ – filtru trece bandă, cu frecvențele de tăiere de 1000 Hz și 2000 Hz;
- $F.HP$ – filtru trece sus, cu frecvența de tăiere de 4000 Hz.

Express VI-ul de achiziție de sunet *Acquire Sound* apelat în VI-ul care rulează pe calculator trebuie configurat cu un canal de achiziție, având durata de 0.1 secunde și o rată de achiziție cât mai mică (în acest exemplu de 11.025 Hz).

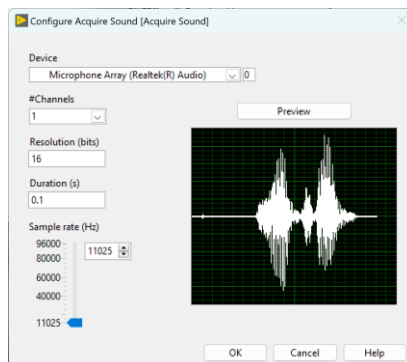


Figura 3.4 – Fereastra de configurare a *Express* VI-ului de achiziție de sunet *Acquire Sound*

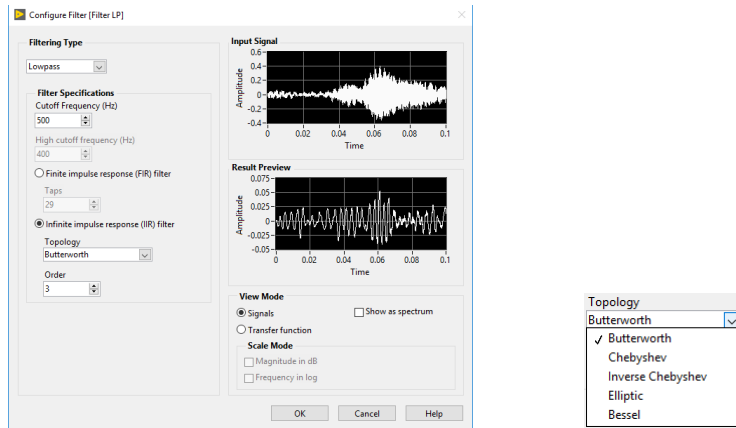


Figura 3.5 – Fereastra de configurare a *Express VI*-ului de filtrare (*F.LP*)

Fereastra de configurare a filtrelor permite diferite opțiuni, iar în partea dreaptă se afișează semnalul de intrare și rezultatul aplicării filtrului asupra semnalului de intrare, cu parametrii setați în partea stângă a ferestrei.

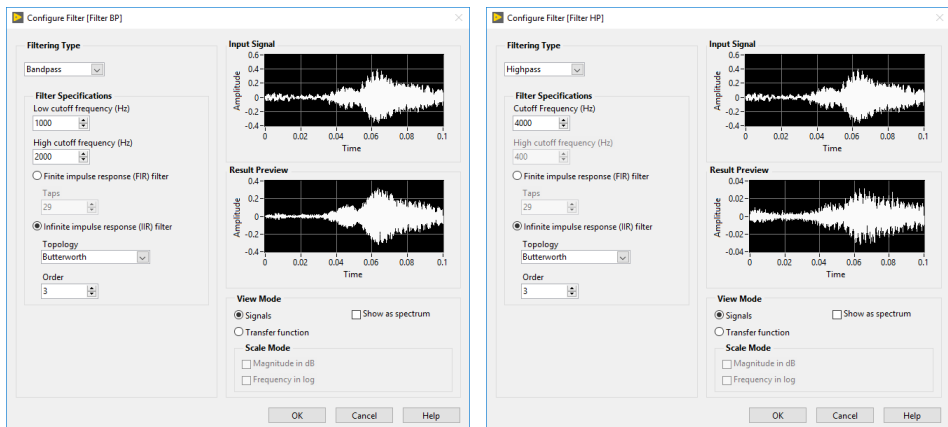


Figura 3.6 – Ferestrele de configurare a *Express VI*-urilor de filtrare (*F.BP*) și (*F.HP*)

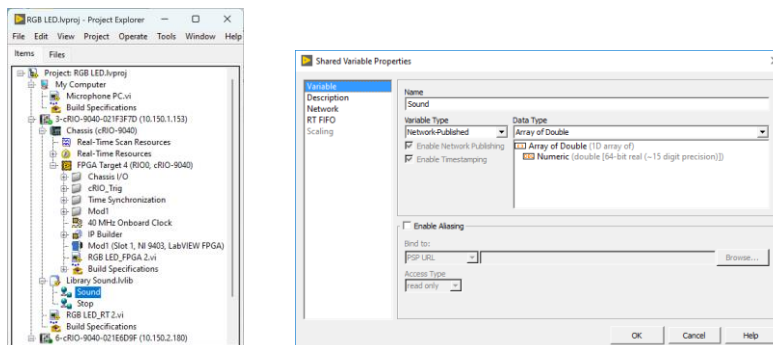


Figura 3.7 – Fereastra de proiect și de configurare a variabilei de tip *Network-Published*

Transferul datelor între cele două VI-uri care rulează pe calculator și pe CompactRIO™ RT se face prin variabila partajată *Sound* de tip *Network-Published* creată în fereastra de proiect și configurată ca un vector de numere reale (*Array of Double*), din opțiunea *Variable*. În opțiunea *Network* se alege un *buffer* de 500 valori și un singur proces de scriere.

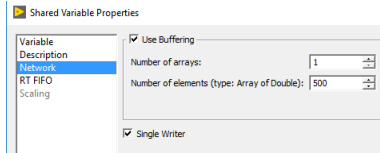


Figura 3.8 – Variabila de tip *Network-Published*, opțiunea *Network*

Înainte de compilarea VI-ului, din fereastra de proiect liniile DIO0÷3 ale modului NI 9401 trebuie configurate de la proprietăți să fie de tip *Output*.

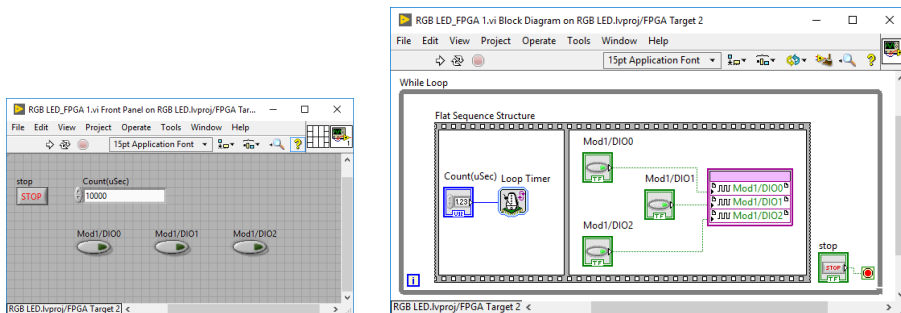


Figura 3.9 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (varianta 1)

Programul care rulează pe CompactRIO™ RT este prezentat în continuare. Valorile semnalelor de la cele trei filtre sunt extrase în trei vectori și comparate cu valorile de prag corespunzătoare celor trei culori pe care dorim să le comandăm. Pentru comanda LED-urilor se utilizează ieșirile digitale oferite de modulul NI 9401 (liniile DIO0, 1, și 2) sau NI 9403 (liniile DIO5, 6, și 7).

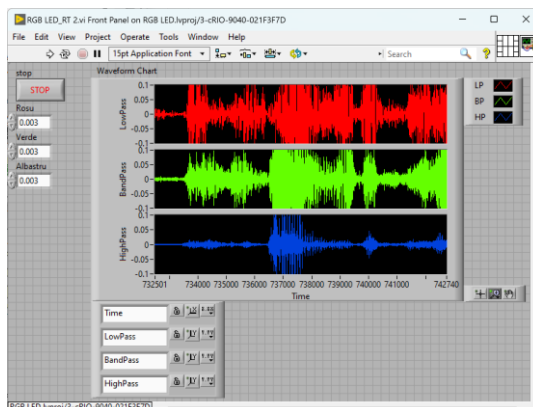


Figura 3.10 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT (varianta 1)

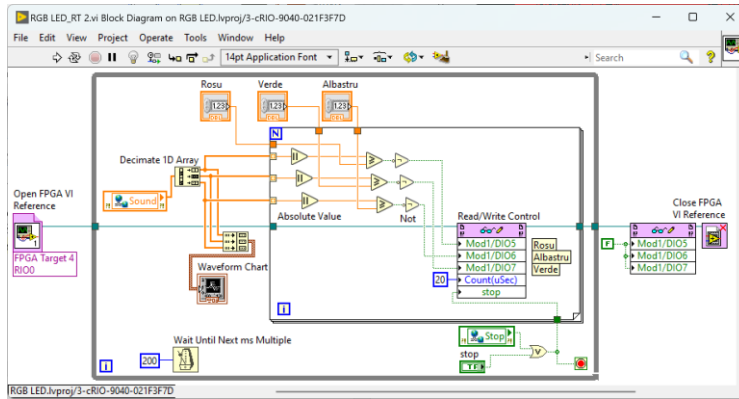


Figura 3.11 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT (varianta 1)

Varianta a doua de program citește microfonul, într-o buclă separată care rulează în paralel cu prima buclă în FPGA. Pentru achiziția de sunet este necesară conectarea unui microfon la un modul cu intrări analogice, de exemplu NI 9239, NI 9230, NI 9215, NI 9381 sau NI 9201.

Cele două bucle *While* din aplicația FPGA rulează real în paralel. În prima buclă se citește un număr de puncte *Scan* (de exemplu 2000) de pe intrarea analogică AI0 unde este conectat microfonul, după care se generează întreruperea numărul 0, configurată fără confirmare. Bucla se repetă cu perioada stabilită de funcția *Loop Timer*, cu valoarea setată în controlul numeric *Count uPh*.

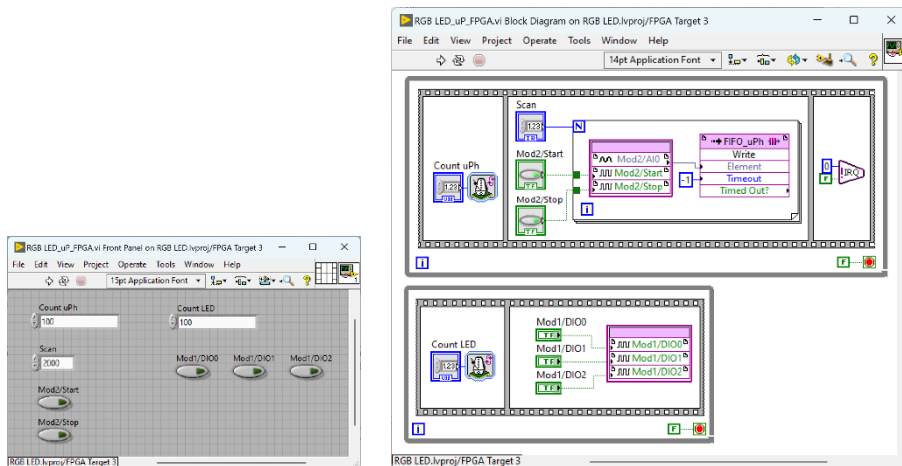


Figura 3.12 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (varianta 2)

A doua buclă comandă cele 3 LED-uri conectate pe ieșirile digitale, iar bucla se repetă cu perioada stabilită de funcția *Loop Timer*, cu valoarea setată în controlul numeric *Count LED*. Aplicația care rulează pe CompactRIO™ FPGA este configurată să facă achiziția continuu, fără să aștepte confirmarea întreruperii.

Pentru transferul datelor între FPGA-ul inclus în șasiu și aplicația RT de pe CompactRIO™ se utilizează o structură de tip *buffer FIFO (First In, First Out)*, creată și configurată pe *FPGA Target*.

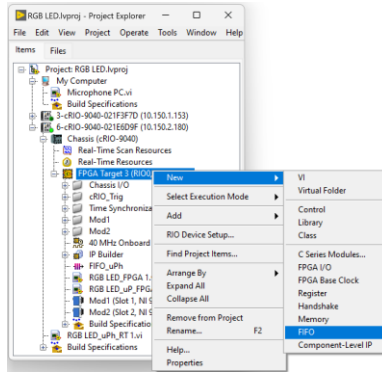


Figura 3.13 – Fereastra de proiect pentru CompactRIO™, crearea unui FIFO

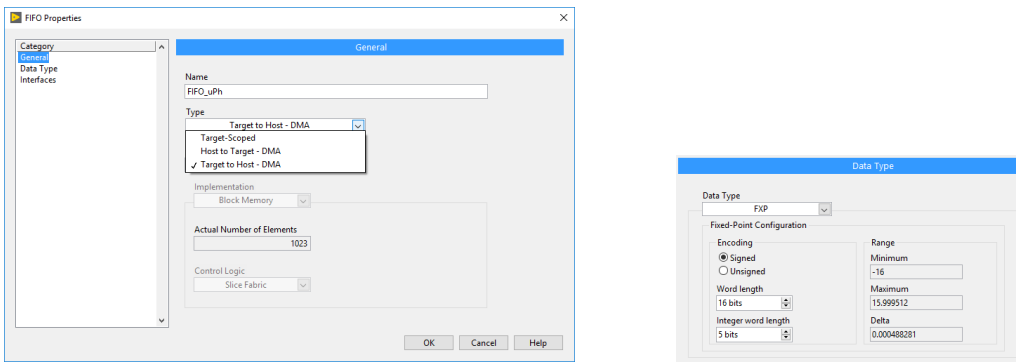


Figura 3.14 – Fereastra de configurare a FIFO-ului, tipul de date și direcția de transfer

În aplicația care rulează pe CompactRIO™ RT se poate seta numărul de valori citite de la microfon și se așteaptă primirea întreruperii cu numărul 0.

Se citește din FIFO un număr de puncte egal cu valoarea *Scan* și se confirmă sau nu întreruperea, în funcție de configurare.

Pentru fiecare filtru se setează frecvența de tăiere. Semnalul de la ieșirea filtrului este comparat cu o valoare de prag diferită pentru fiecare filtru. Ca rezultat al comparației, LED-ul va fi aprins o durată mai mare sau mai mică în funcție de amplitudinea și perioada semnalului.

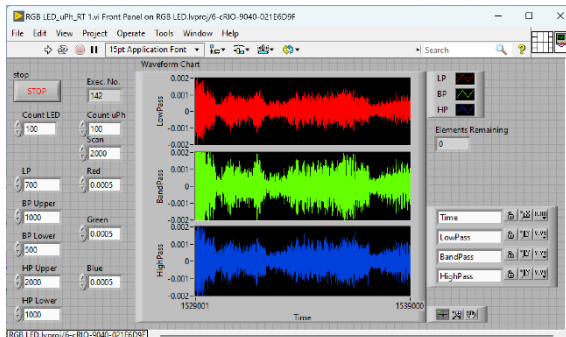


Figura 3.15 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT (varianta 2)

Afișarea semnalelor pentru vizualizare în trei ecrane distincte cu axa X comună este utilă atunci când semnalele sunt complexe sau au amplitudine diferită. În funcție de performanțele microfonului utilizat, dar și de melodia pe care o ascultăm, diferența dintre semnalele de frecvență joasă (10 mV) și cele de frecvență ridicată (1 mV) poate să fie de un ordin de mărime. Se va selecta pentru *Waveform Chart* opțiunea de afișare *Stack Plots*.

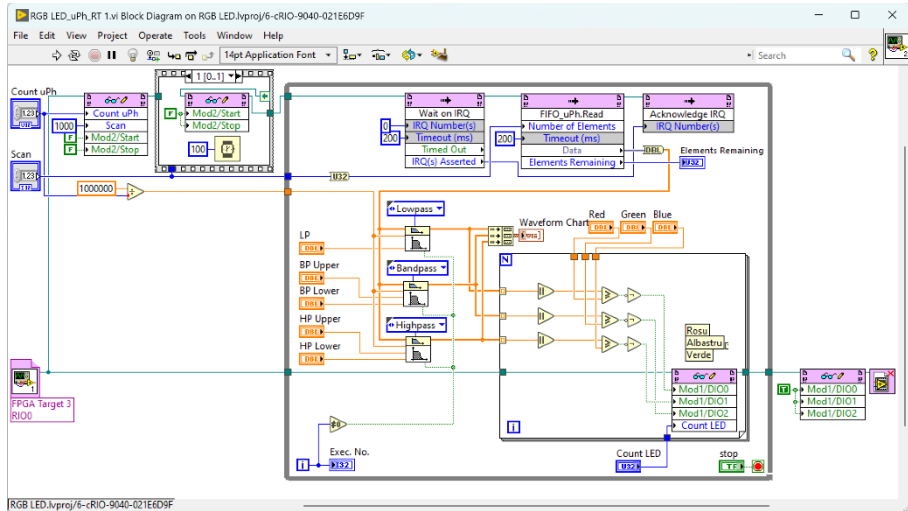


Figura 3.16 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT (varianta 2)

Aplicația dezvoltată în LabVIEW™ cu modulele RT și FPGA poate să fie compilată și salvată în memoria flash pe CompactRIO™, iar la punerea sub tensiune a *target*-ului sau după un reset, să fie pornită automat fără interacțiunea utilizatorului. Cu această setare *target*-ul poate funcționa independent și să ruleze o aplicație.

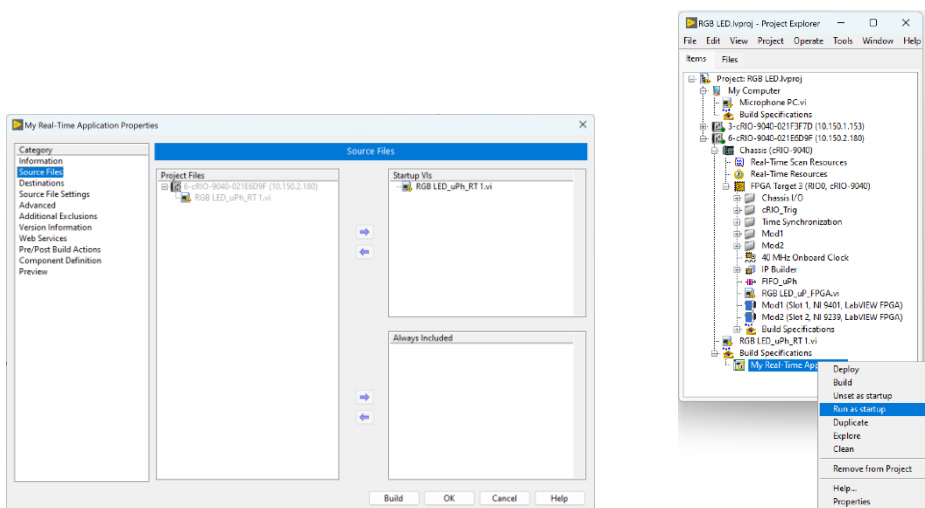


Figura 3.17 – Fereastra din care se face *Build*, *Deploy* și se poate configura VI-ul să ruleze automat la pornirea echipamentului CompactRIO™

4. Controlul turației unui motor de curent continuu

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ cu modulele RT și FPGA pentru măsurarea turației unui motor de curent continuu și generarea unor semnale de comandă de tip PWM [26, 27] pe echipamentul CompactRIO™. Implementarea unui regulator PI care permite setarea referinței și reglarea turației.

Se creează un proiect care include VI-urile pentru implementarea unui regulator de turație pentru un motor de curent continuu de mică putere:

- a. VI-ul de pe FPGA include două bucle *While*, care pot fi implementate separat, iar după ce au fost testate și funcționează sunt incluse într-un VI;
 - prima buclă este pentru măsurarea turației motorului de curent continuu pe linia digitală DIO3 și transferul datelor spre aplicația RT prin citirea valorii indicatorului *Speed*;
 - a doua buclă este pentru generarea unui semnal de tip PWM pe linia digitală DIO7, a cărui frecvență se stabilește din controlul *PWM_Count (usec)* și are factorul de umplere variabil din controlul *Duty Cycle [%]*, prin introducerea sau scrierea unor valori;
- b. VI-ul de pe CompactRIO™ RT permite citirea sau scrierea datelor din FPGA și transmiterea lor prin rețeaua Ethernet cu o variabilă partajată de tip *Network-Published* numită *SP_Spd_DC*;
- c. VI-ul de pe calculator permite afișarea grafică a celor 3 mărimi (referința, comanda și turația citită) și setarea valorii referinței de turație.

ATENȚIE: *standul funcționează cu tensiune de 5 V! Nu alimentați cu tensiune mai mare, modulul electronic nu include protecții la supratensiune.*



Figura 4.1 – Standul pentru controlul turației

Conectarea motorului de curent continuu se face cu un modul electronic dedicat care include un adaptor pentru procesarea semnalului de la traductorul de turație și *driver*-ul de comandă implementat cu circuitul integrat L293 [27].

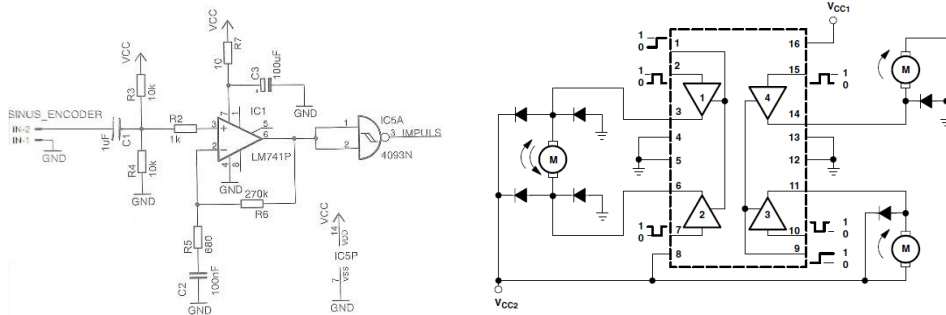


Figura 4.2 – Schema modului de adaptare a semnalului pentru măsurarea turației și *driver*-ul de comandă L293

Prima soluție de implementare este prezentată în continuare. Pentru intrările și ieșirile digitale se utilizează modulul NI 9401 (DI3 și DO7), care oferă metoda *Wait On Rising Edge* pentru detecția frontului crescător al semnalului. Înainte de compilarea VI-ului, din fereastra de proiect liniile DIO4÷7 ale modului NI 9401 trebuie configurate de la proprietăți să fie de tip *Output*.

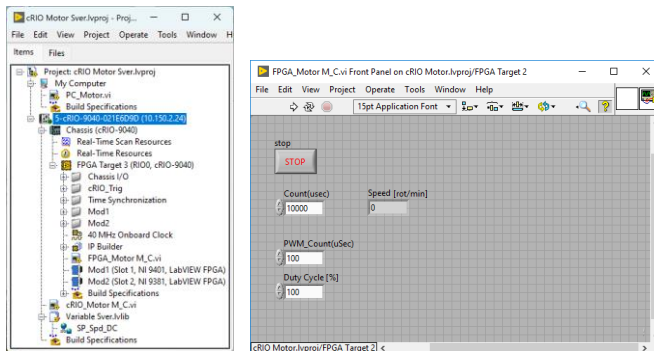


Figura 4.3 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe FPGA

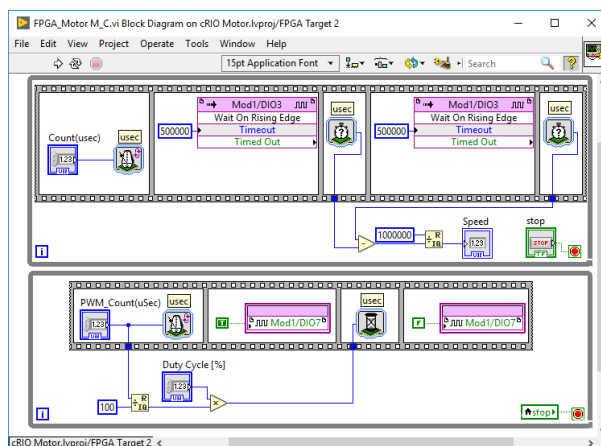


Figura 4.4 – Diagrama bloc, VI-ul care rulează pe FPGA

Se vor realiza trei experimente cu o frecvență a semnalului PWM de comandă a motorului de 100 Hz, 1 kHz și 10 kHz, pentru care factorul de umplere

este între 10% și 90%. Un exemplu cu semnale este prezentat în figura următoare. Se măsoară și se notează în tabel frecvența de la ieșirea traductorului de turație.

Se vor vizualiza toate semnalele cu osciloscopul TDS1012B, TDS1012C sau DSO1012A pentru a verifica funcționarea corectă a programelor.

Frecvența semnalului de comandă	100 Hz	1 kHz	10 kHz
Turația / frecvența măsurată			

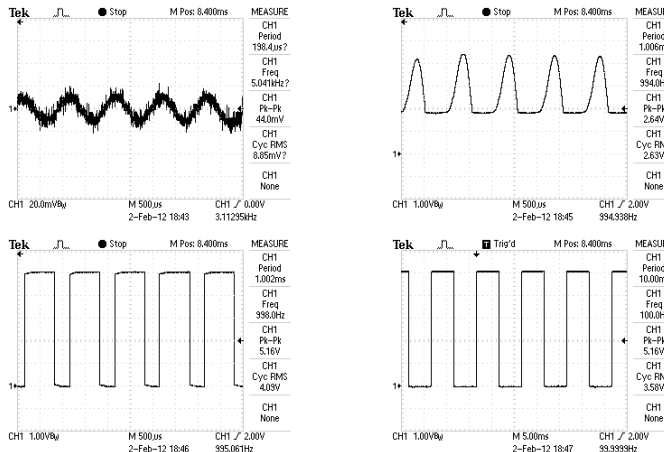
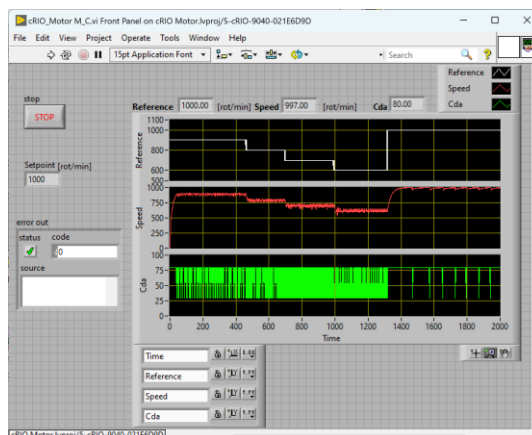


Figura 4.5 – Ieșirea traductorului de turație semnalul original, semnalul filtrat și amplificat, ieșirea din *trigger*-ul Schmitt și semnalul PWM de comandă

☺ Notați dacă în urma experimentelor realizate observați diferite fenomene legate de frecvența semnalului de comandă a motorului de curent continuu: motorul vibrează pentru frecvențe joase sau emite sunete care pot să fie deranjante pentru frecvențe medii ale semnalului de comandă.

Programul care rulează pe RT este simplu și implementează un regulator bipozițional pentru testarea motorului. Include citirea turației din indicatorul *Speed*, scrierea comenzii în controlul *Duty_Cycle [%]* și transferul datelor către calculator prin variabila partajată de tip *Network-Published* numită *SP_Spd_DC*.



Waveform Chart -> Stack Plots

Figura 4.6 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT

Variabila pentru comunicație este de tip *Array of UInt32*, configurată cu 3 *buffere* cu 50 de elemente și cu mai multe procese de scriere: **opțiunea Single Writer nu trebuie selectată**. Bucla *While* include o întârziere *Wait Until Next ms Multiple* pentru timp real, configurată cu microsecunde și reprezentare pe 32 biți.

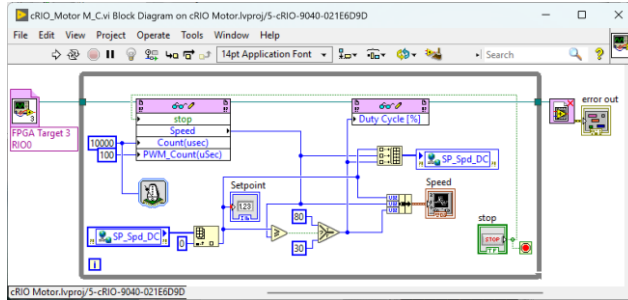


Figura 4.7 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Prima variantă de program care rulează pe calculator va afișa grafic cele 3 mărimi, iar utilizatorul poate seta valoarea de referință pentru turație: **referința de turație** este reprezentată pe grafic cu alb; **comanda** între 0 și 100% este reprezentată cu verde; **turația măsurată** este reprezentată cu roșu. Pentru vizualizarea semnalelor în trei ecrane distincte cu axa X comună se va selecta pentru *Waveform Chart* opțiunea de afișare *Stack Plots*.

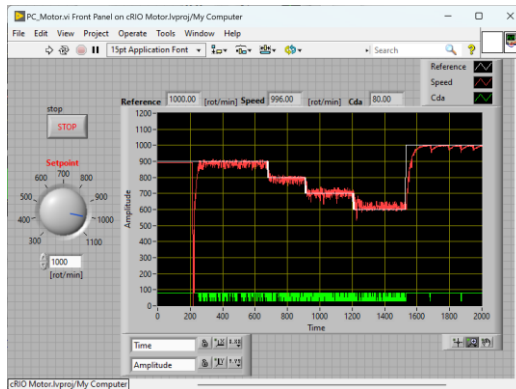


Figura 4.8 – Interfața cu utilizatorul, VI-ul care rulează pe calculator

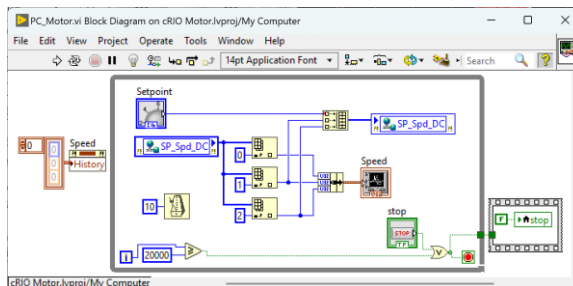


Figura 4.9 – Diagrama bloc pentru VI-ul care rulează pe calculator

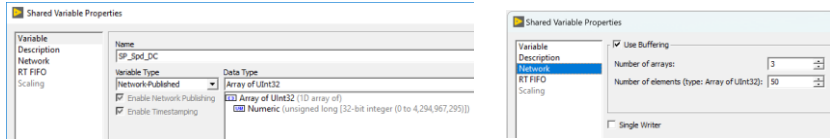


Figura 4.10 – Fereastră de configurare pentru variabila „SP_Spd_DC”

A doua soluție de implementare în LabVIEW™ prezentată în continuare este mai complexă. Componentele incluse în fereastra de proiect sunt:

- VI-ul care rulează pe FPGA și include *Express VI*-ul cu regulatorul PID;
- cele două FIFO-uri pentru transferul datelor între FPGA și RT;
- VI-ul care rulează pe RT și include variabilele partajate de tip *Network-Published* pentru transferul datelor între CompactRIO™ și calculator;
- VI-ul care rulează pe calculator și oferă interfața cu utilizatorul.

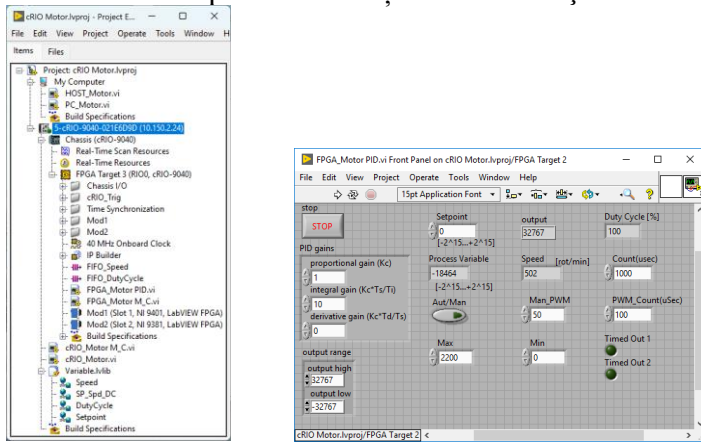


Figura 4.11 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe FPGA

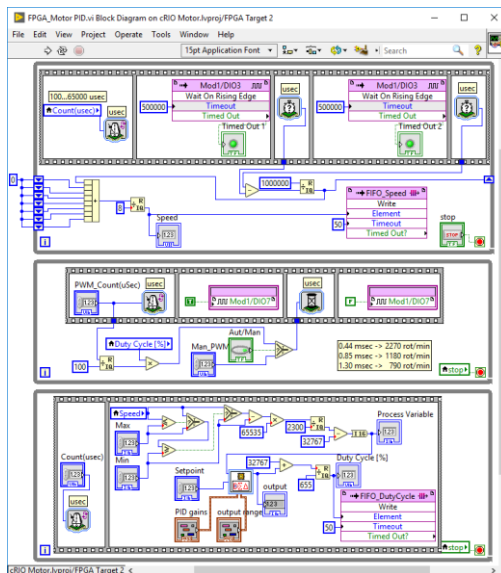


Figura 4.12 – Diagrama bloc, VI-ul care rulează pe FPGA

- variabila *DutyCycle* este de tip *Array of Uint16*, iar de la opțiunea *Network* a fost setat un *buffer* cu 100 de elemente și un singur proces *Write*.

Pentru soluția a doua de implementare, în VI-ul care rulează pe calculator se face afișarea grafică a celor 3 mărimi și se setează valoarea de referință pentru turație: **referința de turație** este reprezentată pe grafic prin culoarea albă; **comanda** este reprezentată cu verde și a fost scalată în același interval ca și turația; **turația măsurată** este reprezentată pe grafic cu roșu.

Diferența dintre cele două variante de VI-uri care rulează pe calculator este legată de numărul de variabile partajate utilizate și felul în care acestea sunt accesate.

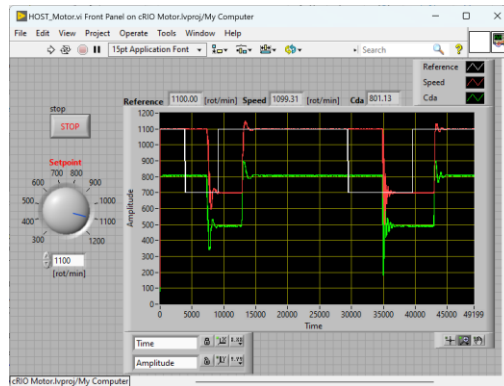


Figura 4.15 – Interfața cu utilizatorul, VI-ul care rulează pe calculator

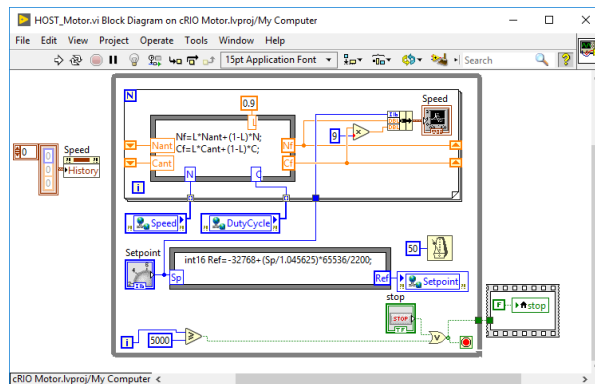


Figura 4.16 – Diagrama bloc, VI-ul care rulează pe calculator

☺ Modificați VI-ul care rulează pe RT, implementarea cu regulator bipozițional astfel încât frecvența de execuție să fie de 1 kHz sau 10 kHz. Testați dacă funcționează corect reglarea de turație în acest caz. Observați comportarea motorului pentru semnalul de comandă aplicat.

5. Măsurarea puterii consumate din rețeaua electrică *

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ pe CompactRIO™ RT și FPGA pentru citirea tensiunii de la rețeaua de distribuție și a curentului consumat de un echipament. Este prezentată o aplicație complexă care permite măsurarea defazajului între tensiune și curent, calcularea puterilor active și reactive; măsurarea frecvenței; calculul transformatei FFT [29].

Se creează un proiect care include mai multe VI-uri pe diferite *target*-uri, astfel:

- primul VI care rulează pe FPGA citește datele de la modulul NI 9215 pe canalele analogice: AI0 (*tensiune*) și AI1 (*curent*);
- al doilea VI care rulează pe CompactRIO™ RT așteaptă primirea întreruperii (0) configurată fără confirmare, pentru transferul datelor prin DMA FIFO din aplicația FPGA și procesarea datelor. VI-ul include două bucle *While* care rulează în paralel, prima buclă pentru transferul datelor de la FPGA și a doua buclă pentru procesarea și afișarea datelor.

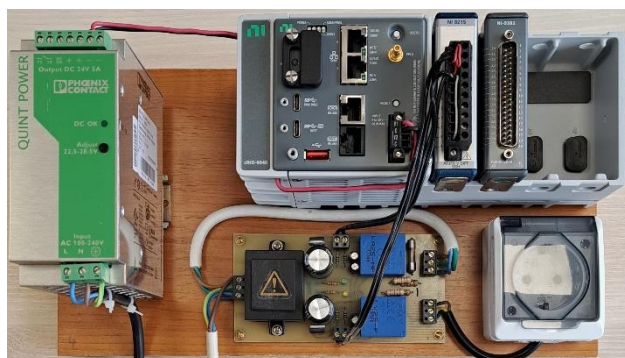


Figura 5.1 – Standul pentru măsurarea puterii, modulul cu senzorii Hall

Schema electronică implementată pentru a măsura tensiunea rețelei de distribuție și consumul de curent este prezentată în figura următoare.

Tensiunea rețelei electrice se măsoară cu traductorul LEM LV 25-P. Acest traductor are următoarele avantaje: precizie și liniaritate bună, derivă termică scăzută, timp de răspuns mic, lățime de bandă mare, imunitate la interferențe externe [30].

Măsurarea consumului de curent se face cu traductorul LEM LA 25-NP configurat pentru gama de 5A pentru a avea o rezoluție maximă de măsurare. Acest traductor are următoarele avantaje: lățimea de bandă mare, pierderi reduse și capacitate ridicată de curent la suprasarcină [31].

Traductoarele LEM au la ieșire semnale în domeniul ± 20 mA, iar conversia la tensiune pentru adaptarea la domeniul de intrare a modulului NI 9215 a fost implementată cu rezistențe de 200 ohmi.

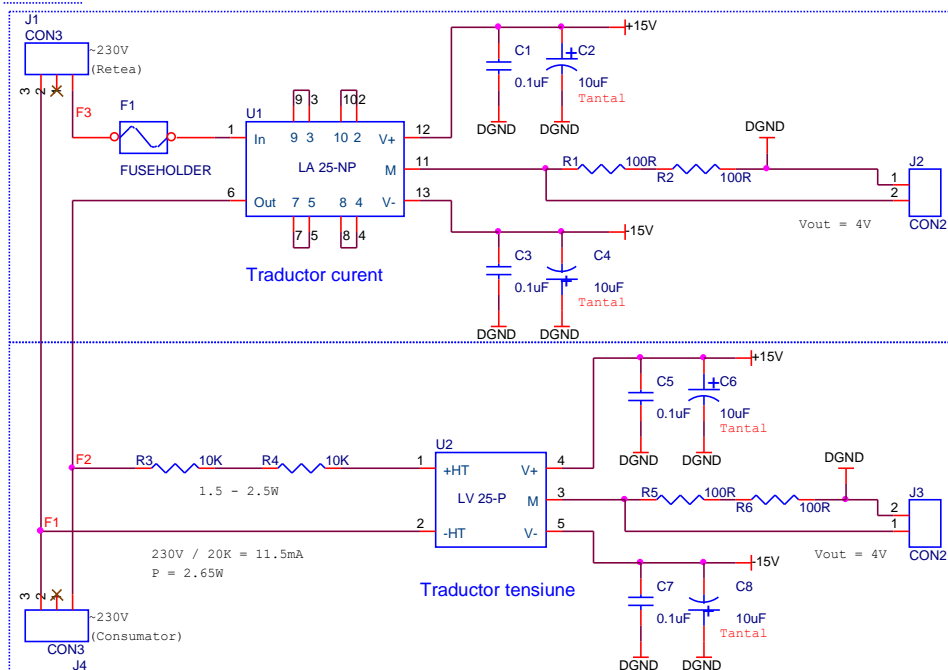


Figura 5.2 – Schema modului cu traductoare LEM de tensiune și curent

O soluție de implementare a aplicației este prezentată în continuare. Este implementată și o variantă simplificată, pentru a înțelege mai ușor funcționarea componentelor incluse în proiect.

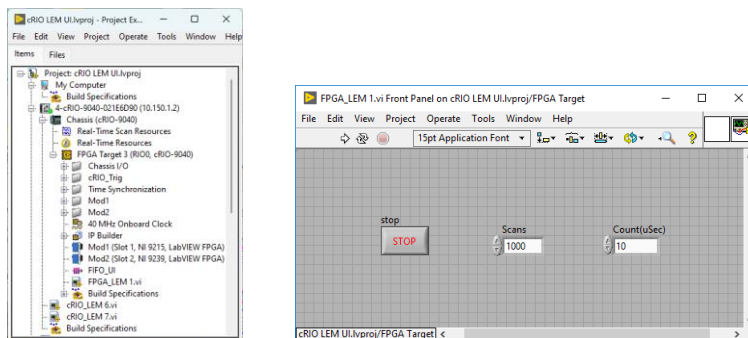


Figura 5.3 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe FPGA

În diagrama bloc a aplicației care rulează pe CompactRIO™ FPGA este implementată rata de eșantionare prin apelarea funcției *Loop Timer*. Transferul de date între cele două *target-uri* FPGA și RT se face prin DMA FIFO, iar sincronizarea transferului de date se face cu întreruperea 0, fără confirmare.

Justificarea alegerii modului NI 9215 cu eșantionare și conversie simultană pe patru canale este aceea că oferă achiziția pentru cele două mărimi măsurate U și I, în același timp. Valorile măsurate simultan pentru tensiunea rețelei și curentul consumat sunt necesare pentru a determina defazajul și a calcula

puterea activă și reactivă. Este posibilă achiziția simultană pe fiecare canal cu o rată maximă de 100 kS/s, iar pentru analiza armonicilor achiziția semnalelor de tensiune și curent se face la o rată minimă de 200 de eșantioane pe o perioadă.

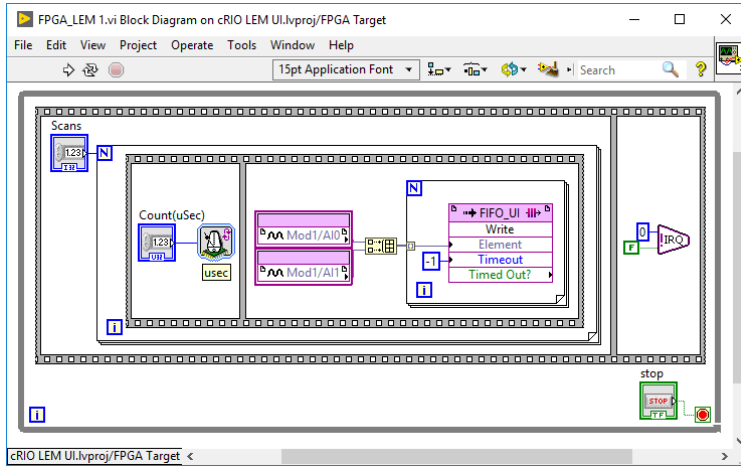


Figura 5.4 – Diagrama bloc, VI-ul care rulează pe FPGA

Interfața cu utilizatorul prezentată în figura următoare include indicatoare grafice pentru valorile măsurate de tensiune, curent, defazaj ϕ (U, I), calculul puterii active, reactive și FFT (*Fast Fourier Transform*). Această interfață este oferită de VI-ul care rulează pe CompactRIO™ RT. Sunt afișate frecvența și valorile efective pentru tensiune și curent.

Tab-ul numit *Settings* permite utilizatorului să configureze în aplicația care rulează pe FPGA rata de eșantionare a semnalului și numărul de citiri.

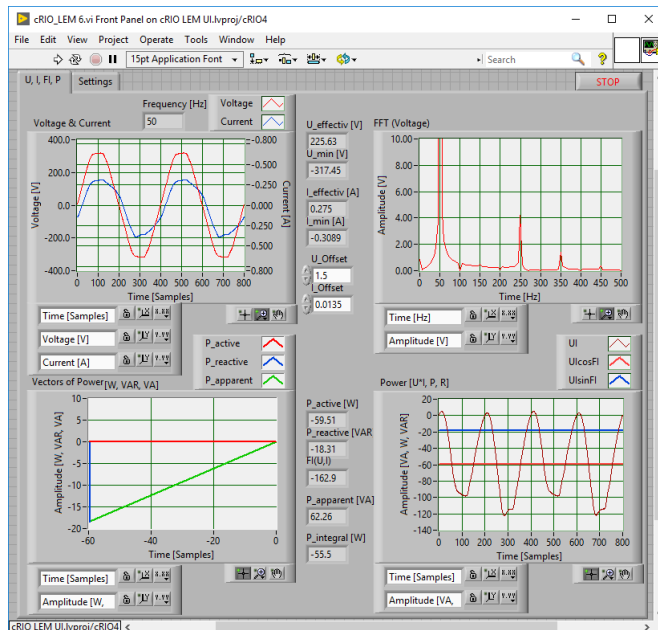


Figura 5.5 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT

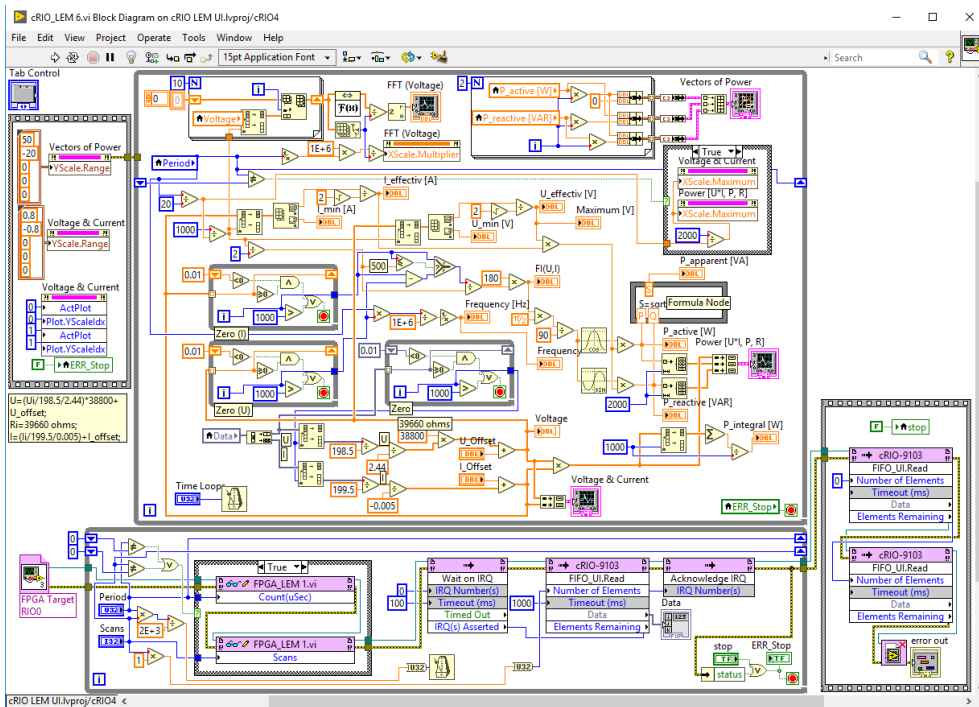


Figura 5.6 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Pentru a calcula defazajul dintre tensiune și curent, din datele reprezentate sub formă de vectori se determină indicii corespunzători valorilor de „zero”-valorii obținute la trecerea prin zero a semnalelor de tensiune și curent. Valoarea defazajului se determină ca diferență dintre doi indici:

$$\phi(U, I) = \frac{U_{zero} - I_{zero}}{N/2} \cdot 180,$$

unde $N = 200 - 5000$ este numărul de eșantioane pentru o perioadă, în funcție de setările realizate în controlul numit *Scans*.

Valoarea maximă a tensiunii și curentului sunt determinate din vectorii de date pentru o perioadă. Valorile efective sunt calculate cu formulele:

$$U_{efectiv} = \frac{U_{max}}{\sqrt{2}}, \quad I_{efectiv} = \frac{I_{max}}{\sqrt{2}}.$$

Într-un sistem monofazat relațiile de calcul pentru puterea activă P și puterea reactivă Q sunt definite astfel:

$$P = U \cdot I \cdot \cos \phi, \quad Q = U \cdot I \cdot \sin \phi.$$

Pentru puterea aparentă relațiile de calcul sunt:

$$S^2 = U_{ef}^2 \cdot I_{ef}^2 \quad S^2 = P^2 + Q^2$$

Unitățile de măsură utilizate pentru cele două puteri sunt [Watt] și [VAR]. Unitatea de măsură utilizată pentru consumul de energie este kilowatt oră [kWh]. Timpul de execuție al unei bucle care include transferul datelor, calcule și afișarea datelor este de aproximativ 11 ms. Se va seta o temporizare de 20 ms care este suficient de mare pentru respectarea perioadei de execuție a codului.

În continuare este prezentată soluția simplificată care nu include detecția trecerilor prin zero și calcularea defazajului, FFT, configurare, include doar afișarea formelor de undă a tensiunii și curentului.

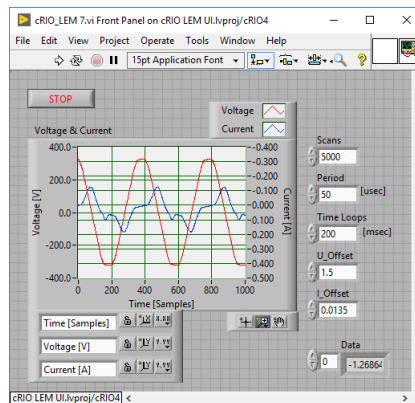


Figura 5.7 – Interfața cu utilizatorul, VI-ul simplificat care rulează pe CompactRIO™ RT

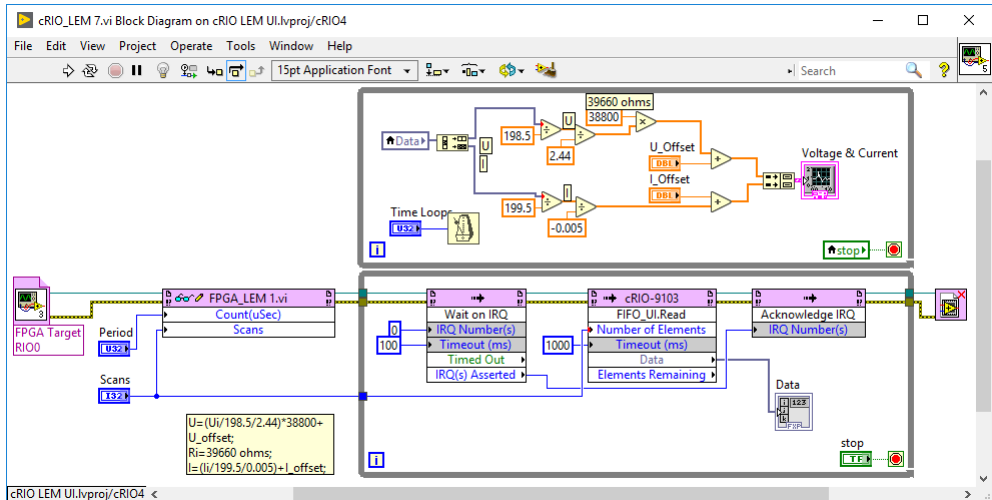


Figura 5.8 – Diagrama bloc, VI-ul simplificat care rulează pe CompactRIO™ RT

Au fost măsurate consumurile pentru trei tipuri de sarcini, pentru a verifica funcționarea sistemului cu sarcină rezistivă, inductivă și capacivă. Rezultatele obținute pentru tensiune și curent, precum și vectorii puterilor sunt prezentate în figurile următoare.

Pentru testarea standului cu o sarcină rezistivă a fost utilizat un bec de 100 W, cu filament.

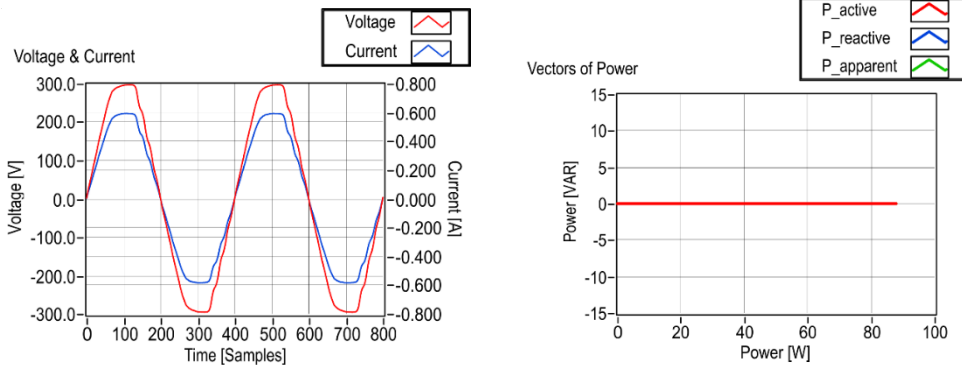


Figura 5.9 – Semnale de tensiune și curent, vectorii puterilor ($Q=0$, sarcină rezistivă)

Pentru testarea standului cu o sarcină inductivă a fost utilizată o stație de lipire cu transformator, care consumă aproximativ 48 W la încălzire.

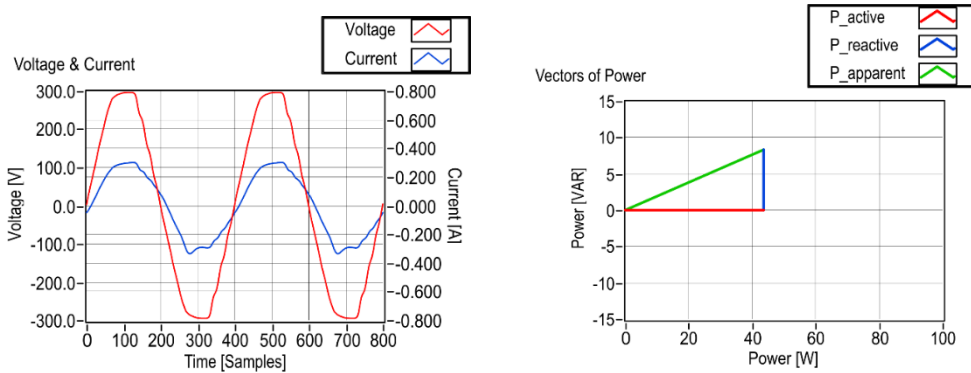


Figura 5.10 – Semnale de tensiune și curent, vectorii puterilor (sarcină inductivă)

Pentru testarea standului cu o sarcină capacitivă a fost conectat un bec de 100 W legat în serie cu un condensator de $50 \mu\text{F} \pm 5\% / 450\text{V}_{ac}$.

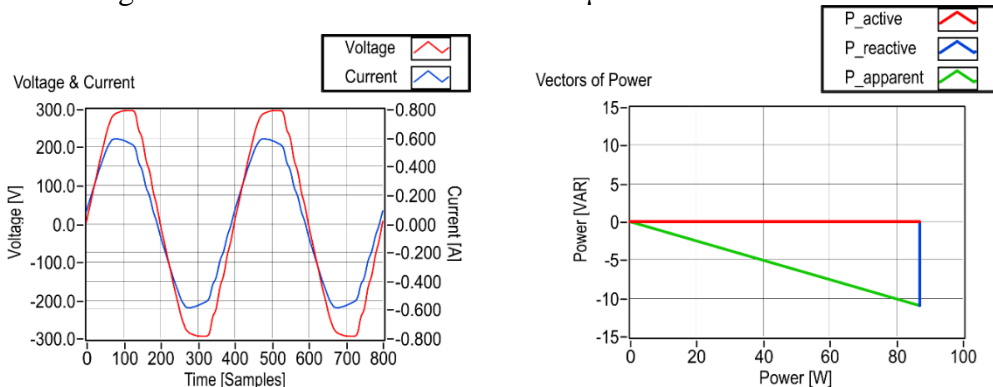


Figura 5.11 – Semnale de tensiune și curent, vectorii puterilor (sarcină capacitivă)

În următorul exemplu sunt prezentate formele de undă pentru tensiune și curent pentru o sursă în comutație inclusă într-un calculator personal.

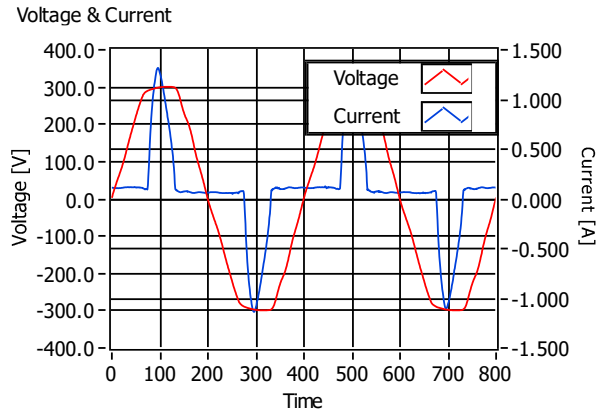


Figura 5.12 – Semnale de tensiune și curent pentru o sursă în comutație

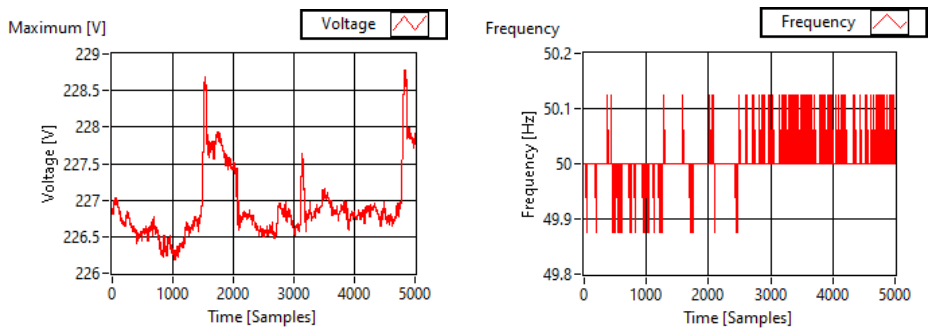


Figura 5.13 – Valori nominale ale tensiunii și frecvenței în rețeaua de distribuție

Pe echipamentul CompactRIO™ RT a fost implementat un analizor de armonici (FFT). O armonică este definită ca un multiplu întreg al frecvenței fundamentale. Frecvența fundamentală a rețelei de distribuție de energie este de 50 Hz. Armonica a doua este de două ori frecvența fundamentală sau 100 Hz.

Sursele în comutație sunt echipamente neliniare care generează armonici în rețea, așa cum se observă în figurile următoare:

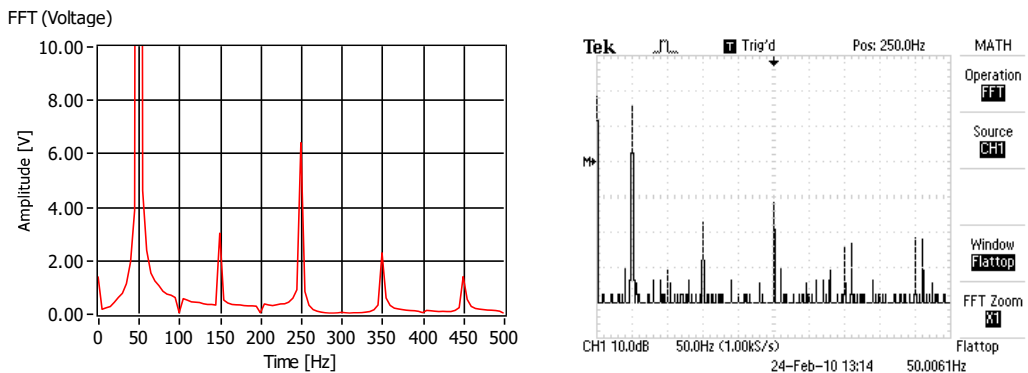


Figura 5.14 – FFT, rezultat obținut pe CompactRIO™ și pe osciloscopul TDS1012B

6. Generatoare de semnale analogice *

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ pe sistemul CompactRIO™ RT și FPGA pentru generarea de semnale analogice și digitale, de amplitudine și frecvență reglabile, sau generarea unor forme de undă cu funcții dedicate de generare.

Se creează un proiect care include VI-uri care să permită generarea de semnale sinusoidale, utilizând *Express VI*-uri sau funcții dedicate de generare de semnale și modulul de ieșire analogică NI 9263. Se verifică funcționarea corectă a programelor cu un osciloscop TDS1012B, C sau DSO1012A.

O primă soluție de implementare utilizează funcții dedicate (SIN, PWM etc.) care sunt incluse în diagrama bloc a VI-ului care rulează pe FPGA și este prezentată în continuare. Semnalul generat pe ieșirea A00 cu setările din figura următoare are amplitudine de 8.2 V vârf la vârf și frecvența de 10.0 kHz.

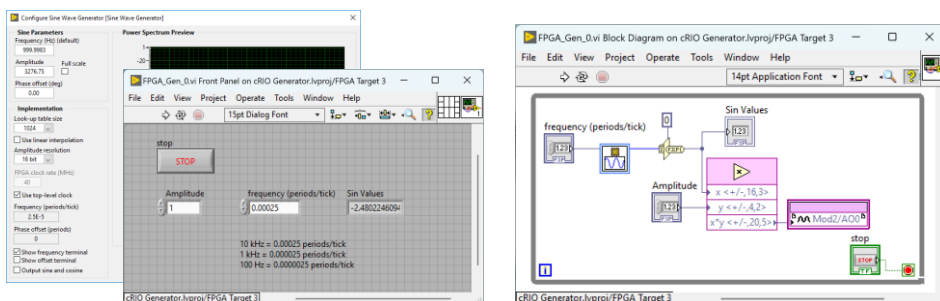


Figura 6.1 – Interfața cu utilizatorul și diagrama bloc, VI-ului care rulează pe FPGA (varianta 1)

Deși soluția este simplă la schimbarea funcției sau parametrilor, atunci când nu se poate reconfigura din *Front Panel-ul Express VI*-ului este necesară recompilarea programului, operație care durează aproximativ 3-5 minute.

Varianta a doua de program permite generarea de semnale între 0.05 Hz și 1 kHz, cu amplitudine maximă de ± 8 V.

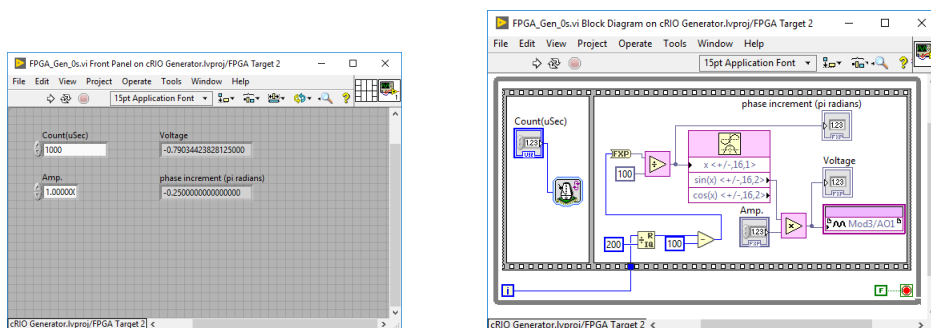


Figura 6.2 – Interfața cu utilizatorul și diagrama bloc, VI-ului care rulează pe FPGA (varianta 2)

Semnalul generat cu setările din figura anterioară are amplitudine de 2.0 V vârf la vârf și frecvența de 5 Hz.

Alte variante mai eficiente decât primele două, care să ofere un mod rapid de modificare a parametrilor semnalelor ce urmează a fi generate sunt bazate pe calcularea punctelor formei de undă pe CompactRIO™ RT și apoi transferarea lor într-o zonă de memorie sau printr-un FIFO către aplicația care rulează pe FPGA, citirea datelor și generarea semnalului pe o ieșire a modului NI 9263.

Calcularea punctelor poate să dureze un timp mai mare decât o perioadă a semnalului și va avea ca rezultat obținerea unui semnal care are câte o „pauză” după fiecare perioadă.

O soluție este calcularea punctelor în afara buclei care trimite datele prin FIFO către FPGA. Această soluție are dezavantajul că la modificarea parametrilor semnalului (amplitudine, frecvență, număr de puncte pe o perioadă) programul trebuie oprit, se modifică parametrii și apoi se repornește programul.

Semnalul generat pe ieșirea AO1 cu setările din figură are amplitudine de ± 5.0 V și frecvența de 3.03 kHz (cu valoarea $Period = 10 / 100$ se obține un semnal cu frecvența de 782 Hz / 78.8 Hz).

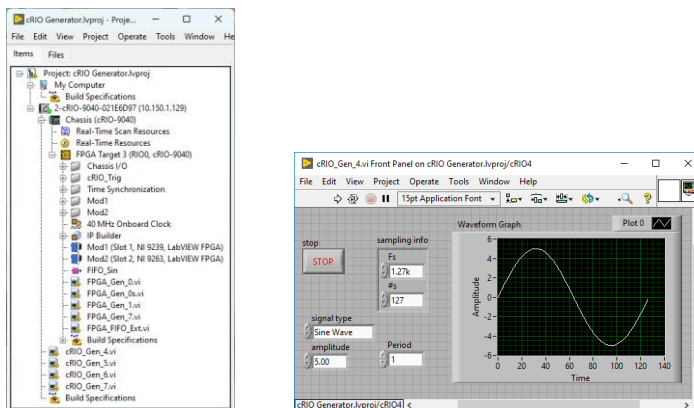


Figura 6.3 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT (varianta 1)

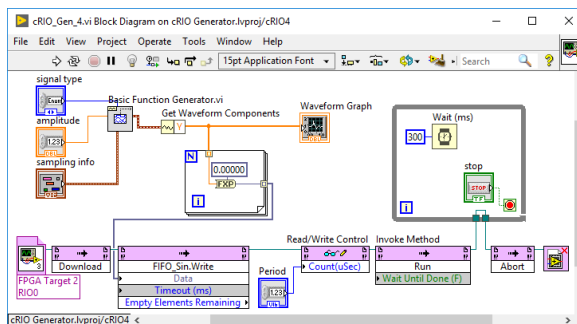


Figura 6.4 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT (varianta 1)

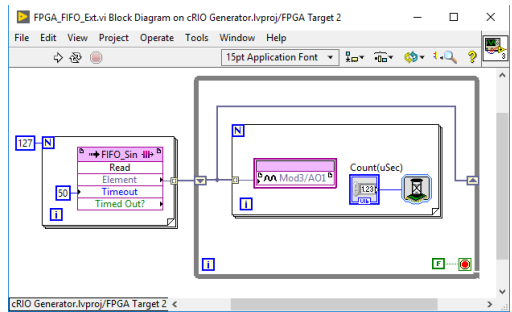
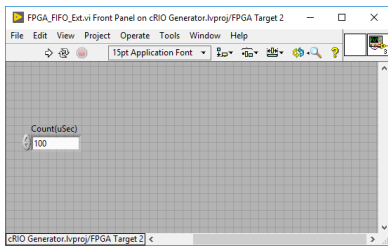


Figura 6.5 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (varianta 1)

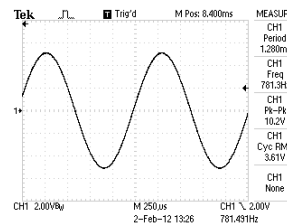
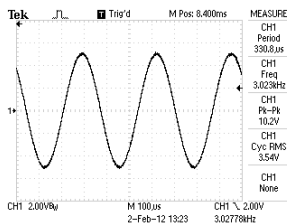


Figura 6.6 – Vizualizări cu osciloscopul, semnal sinusoidal generat cu diferite frecvențe

Cu următoarea implementare pot fi generate semnale de frecvență mai mare față de soluția precedentă de până la 30 kHz, dar semnalul este de o calitate slabă din punct de vedere al formei de undă pentru că are un număr redus de puncte pe o perioadă.

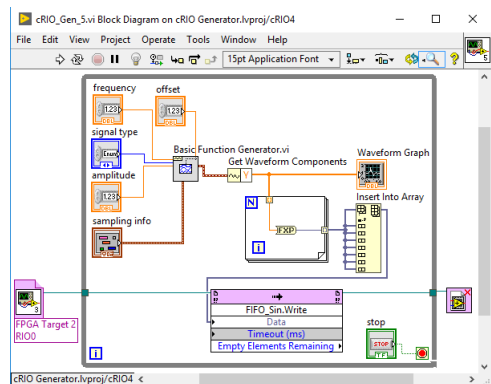
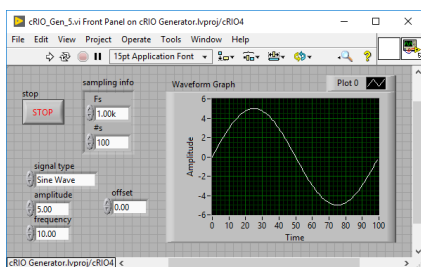


Figura 6.7 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe CompactRIO™ RT (varianta 2)

O soluție mai bună este aceea în care calcularea semnalului se face în buclă, iar vectorul punctelor pentru o perioadă se copiază de N ori multiplicând numărul de perioade. Această operație de copiere a vectorului durează puțin și se obține un vector extins care poate să aibă până la 10.000 de puncte. Funcția de generare a semnalului va avea suficient timp pentru calcularea unui set nou de puncte cu alți parametrii.

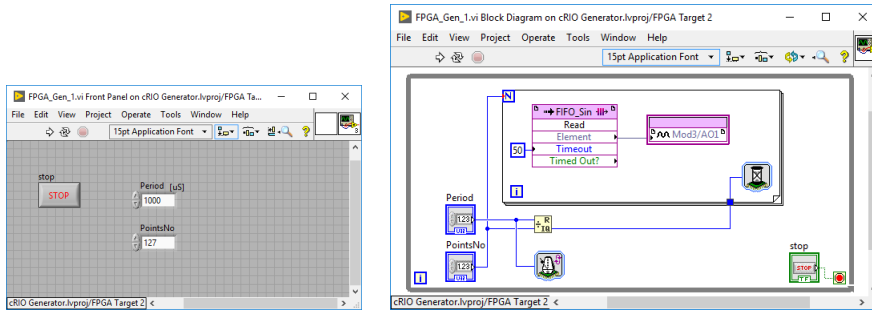


Figura 6.8 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (varianta 2)

În figurile următoare sunt vizualizate cu osciloscopul semnalele generate fără întreruperi. Se pot genera semnale cu frecvență de până la 10 kHz, dar în această situație numărul de puncte pe o perioadă este foarte mic, cu doar 10 valori.

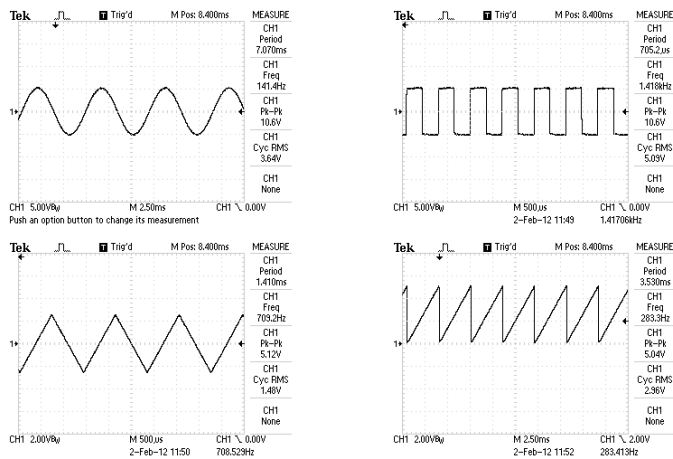


Figura 6.9 – Vizualizări cu osciloscopul ale semnalelor generate

Când timpul de calcul pentru funcția de generare este prea mare, semnalul generat este discontinuu ca în figura următoare. Interfața cu utilizatorul este aceeași și se prezintă diagrama bloc a VI-ul care rulează pe CompactRIO™ RT și o vizualizare cu osciloscopul a semnalului întrerupt periodic.

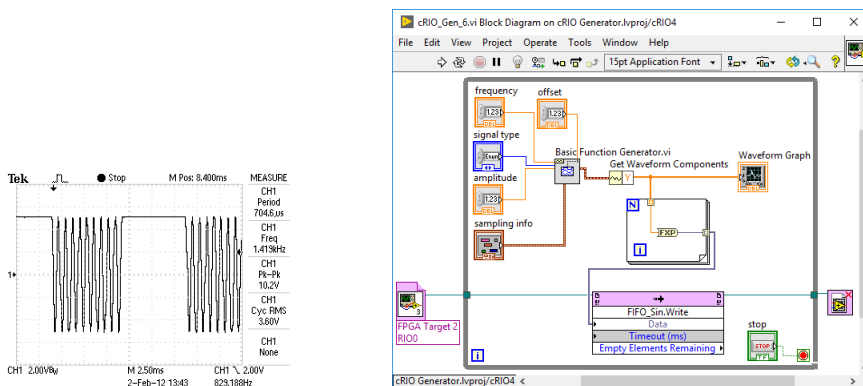


Figura 6.10 – Vizualizare cu osciloscopul a semnalului generat și diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Implementările prezentate nu sunt singurele posibile. Se poate determina simplu dacă în variabilele de control pentru configurarea funcției de calcul pentru o formă de undă a fost o schimbare, utilizatorul a apăsat un buton și doar atunci să fie apelată funcția de calcul.

Într-o variantă simplificată de implementare, prezentată în continuare, se testează doar tipul semnalului. Se utilizează un comparator și structura *Case* și se verifică dacă valoarea curentă a controlului *signal type* este aceeași sau diferă de valoarea anterioară.

Valoarea anterioară este memorată între execuțiile buclei cu ajutorul elementului *Shift Register* amplasat pe marginea buclei *While*.

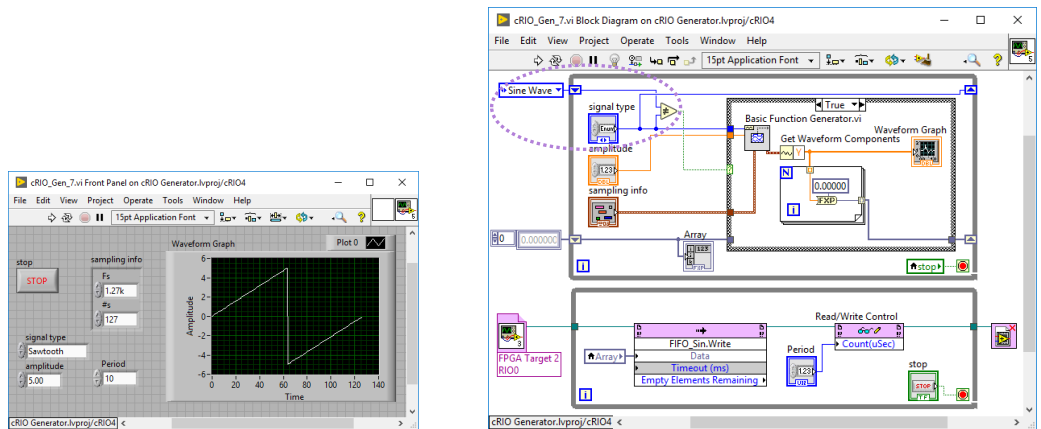


Figura 6.11 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe CompactRIO™ RT (varianta 3)

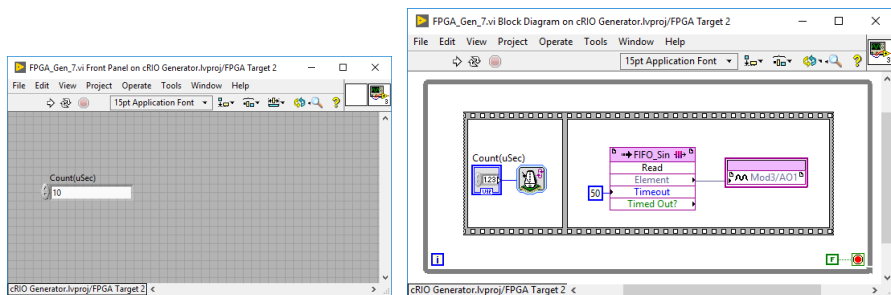


Figura 6.12 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe FPGA (varianta 3)

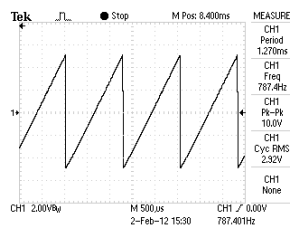


Figura 6.13 – Vizualizare cu osciloscopul a semnalului generat cu varianta a 3-a de program

☺ Realizați un VI pe *target*-ul FPGA care să permită generarea unui semnal dreptunghiular cu factor de umplere variabil între 0 și 100% pe o linie digitală de la unul din modulele NI 9381 (permite generarea de semnale cu frecvența de 1 MHz) sau 9401 (permite generarea de semnale cu frecvența între 5 MHz și 20 MHz).

Determinați cu osciloscopul frecvența maximă pe care o puteți genera. Pentru ca VI-ul să ruleze mai repede puteți modificați frecvența FPGA-ului la una din frecvențele următoare 80 MHz, 120 MHz, 160 MHz sau 200 MHz.

☺ Realizați un VI care să permită generarea unui semnal prin includerea unui set de 16 *slide*-uri așezate vertical la distanță mică într-un vector din interfața cu utilizatorul, astfel încât să creeze impresia unui ecran care afișează un grafic cu bare verticale.

☺ Realizați un VI care să permită generarea unui semnal pe baza unei ecuații matematice introduse de utilizator.

7. Determinarea capacității energetice a bateriilor

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ pe CompactRIO™ RT și FPGA pentru generarea de semnale cu amplitudine, frecvență și factor de umplere selectabile, citirea tensiunii pe bateriile testate și a curentului consumat de o rezistență de sarcină comandată; calcularea puterii consumate; afișarea grafică a tensiunii și curentului [32]. Măsurarea temperaturii bateriilor în timpul testelor. Vizualizarea semnalelor generate cu osciloscopul.

Se creează un proiect care include VI-urile de test astfel:

- primul VI rulează pe FPGA și citește de la modulul NI 9381 pe canalele analogice de intrare AI1, AI3 (tensiunile pe baterii) și AI0, AI2 (curenții consumați de rezistențele de sarcină comandate) și generează cu același modul pe canalele analogice de ieșire AO0, AO1 semnale de comandă de amplitudine, frecvență și factor de umplere selectabile;
- al doilea VI care rulează pe CompactRIO™ RT citește datele prin DMA FIFO de la aplicația din FPGA și trimite datele spre calculator pentru procesare și afișare prin variabile partajate de tipul *Network-Published*;
- al treilea VI care rulează pe calculator citește datele din variabilele partajate de tipul *Network-Published* și afișează datele pe indicatoare numerice sau grafice. Acest VI permite setarea valorilor pentru amplitudinea semnalului, frecvență și factor de umplere. Poate rula pe un TPC cu Windows fără să fie necesare modificări ale codului.



Figura 7.1 – Standul pentru măsurătorile de capacitate pentru două baterii

Durata experimentului depinde de capacitatea bateriilor, valoarea curentului consumat și factorul de umplere al semnalului generat. Capacitatea unei baterii alcaline este între 1500 mAh și 2500 mAh, dar depinde de curentul consumat și temperatura de funcționare sau de depozitare.

Condițiile în care au fost testate bateriile sunt prezentate în continuare:

- descărcare continuă la un curent de 500 mA;

- descărcare cu impulsuri de 500 mA cu lățime de 10% *ON* și 90% *OFF* generate cu un semnal de tip PWM cu amplitudine selectabilă.

O comparație între valorile de catalog *de la producător (afișate cu italic)* [33] și valorile măsurate pe stand este prezentată în tabelul următor:

	Tip baterii	500 mA continuu	10% 500 mA impulsuri	Creșterea de temperatură la descărcare continuă cu 500mA
1	Energizer L91 Litiu	3100 mAh <i>(3100 mAh)</i>	3136 mAh	28 °C 35 °C
2	Energizer E91 Alcaline	1243 mAh <i>(1400 mAh)</i>	2277 mAh	25 °C 33 °C

Un consum de curent de valoare mare are influență diferită pentru bateriile cu litiu și cele alcaline: capacitatea bateriilor cu litiu rămâne aproximativ aceeași la valoarea de 3100 mAh, în timp ce capacitatea bateriilor alcaline se reduce semnificativ.

Pentru bateriile alcaline capacitatea a scăzut de la 2277 mAh la 1243 mAh, diferența reprezentând aproximativ 55%, iar această valoare este semnificativă. La bateriile cu litiu capacitatea pentru un consum cu impulsuri de 10% *ON* din 500 mA este aproape aceeași cu descărcarea continuă. În acest experiment curentul măsurat în realitate a fost de 580 mA în loc de 500 mA.

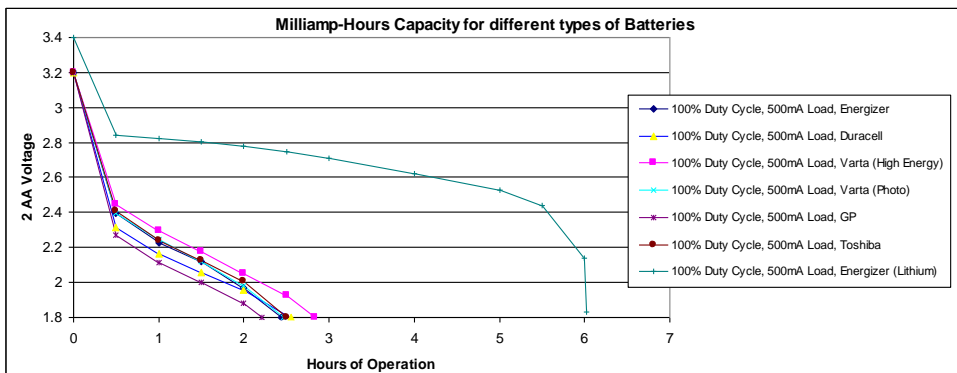


Figura 7.2 – Profilurile de descărcare pentru diferite tipuri de baterii

În funcție de materialele din care sunt construite bateriile de la diferiți producători pentru aceeași capsulă AA, au următoarele capacități:

- 7.5 ÷ 8.6 Wh pentru bateriile *Lithium Thionyl Chloride*, la 3.6V și curent nominal de descărcare sub 2.0 mA. Pentru impulsuri de curent de valoare mare se pot conecta condensatoare de capacitate mare în paralel;
- 4.6 Wh pentru bateriile cu litiu, la 1.5 V și curent de descărcare de 500 mA;
- 3.8 Wh pentru baterii OxyRide, la 1.5 V și curent de descărcare de 15 mA;
- 2.1 ÷ 2.7 Wh pentru baterii Alcaline, la 1.5 V și curent de descărcare de 500 mA.

În continuare sunt prezentate date de catalog pentru bateriile cu *litium* și *alcaline*. Bateriile cu litium au un domeniu mai larg de temperaturi la care pot funcționa și un ciclu mai mare de viață, după 10 ani pot avea disponibilă 90% din capacitate, în condițiile în care au fost stocate într-un mediu dificil. Acest aspect este un avantaj în cazul echipamentelor portabile amplasate în exterior.

Figura următoare prezintă capacitatea bateriilor *Energizer* la diferiți curenți de sarcină. Graficul este preluat din documentul de specificații.

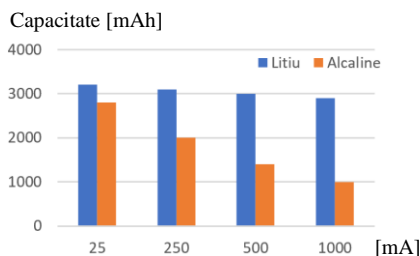


Figura 7.3 – Capacitatea energetică a bateriilor AA la temperatura camerei [33]

Rezultă din figură o capacitate energetică mare pentru bateriile cu litium în comparație cu cele alcaline pentru curenți de sarcină ridicați de 250 ÷ 1000 mA.

Selecția bateriilor se face pe baza unui ghid dat de producător, în funcție de destinație. Sunt baterii a căror destinație este dată de curentul de sarcină pe care îl suportă – curent de sarcină ridicat sau un timp îndelungat de exploatare.

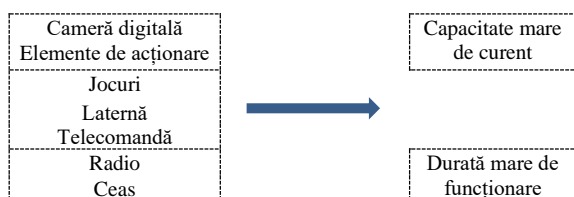


Figura 7.4 – Selecția bateriilor în funcție de destinație

Din graficele următoare rezultă o capacitate ridicată la temperaturi scăzute apropiate de 0 °C pentru bateriile cu litium. În cazul bateriilor alcaline se observă o scădere accentuată a capacității la temperaturi apropiate de 0 °C.

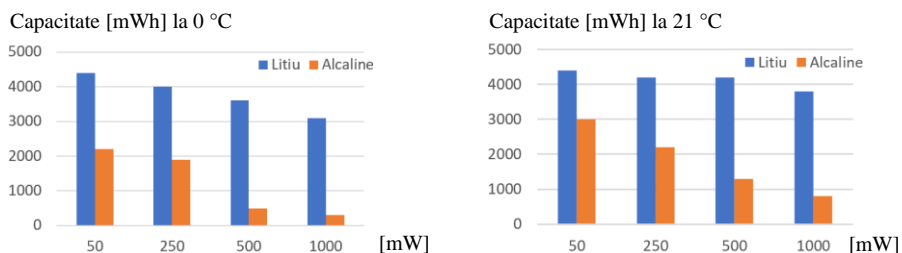


Figura 7.5 – Capacitatea bateriilor în mWh la temperatura de 0 °C și la 21 °C [33]

Capacitatea bateriilor cu litiu față de cele alcaline scade mai puțin, cu 15% până la 30% în funcție de valoarea sarcinii. Condițiile de testare au fost la o putere constantă de descărcare, până la valoarea de 0.8 V, atât la 0 °C cât și la 21 °C.

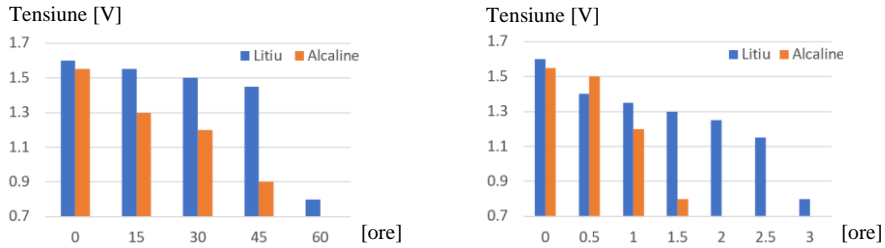


Figura 7.6 – Curbele de descărcare la curent constant de 50 mA și 1000 mA, bateriile cu litiu și alcaline la temperatura de 21 °C [33]

Bateriile alcaline au o curbă de descărcare aproape oblică, tensiunea la bornele bateriei scade mai repede și rezultă o creștere a valorii curentului consumat pentru o putere constantă, în timp ce bateriile cu litiu au o curbă de descărcare relativ plată și o densitate de energie mai mare, menținând valoarea tensiunii la borne.

Din curbele tipice de descărcare rezultă următoarele concluzii:

- bateriile cu litiu pot avea un timp de funcționare de 3 ori mai mare decât cele alcaline, la curenți de sarcină ridicați. Aceasta se datorează rezistenței interne mai mici de $90 \div 150 \text{ m}\Omega$ la bateriile cu litiu față de $150 \div 300 \text{ m}\Omega$ la bateriile alcaline;
- oferă un avantaj în cazul dispozitivelor care funcționează discontinuu, au o perioadă scurtă de timp ($< 1\%$) în care consumă mai mult și o perioadă lungă de timp ($> 99.0\%$) în care consumă foarte puțin: senzorii *wireless*.

Schema electronică pentru sarcina comandată în curent prezentată în figura următoare, include un amplificator operational LM358 și un tranzistor MOSFET IRF540N. Măsurarea precisă a curentului se poate face cu un *șunt de curent* implementat cu circuitul specializat INA138 [34].

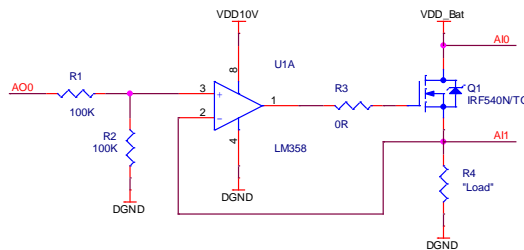


Figura 7.7 - Circuitul utilizat pentru testare, sarcina comandată

Fereastra pentru proiect include dispozitivul CompactRIO™ RT și FPGA, definirea FIFO-urilor și variabilelor partajate de tip *Network-Published*.

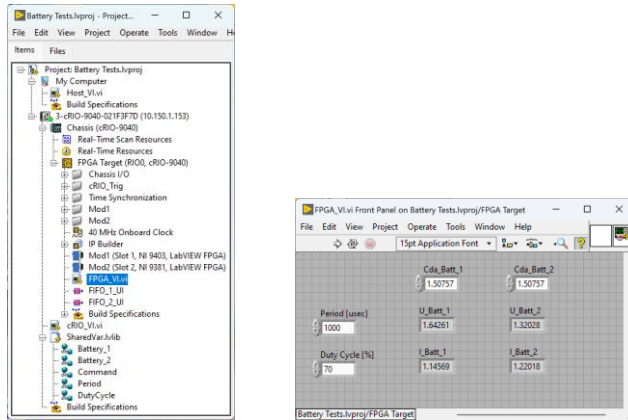


Figura 7.8 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe FPGA

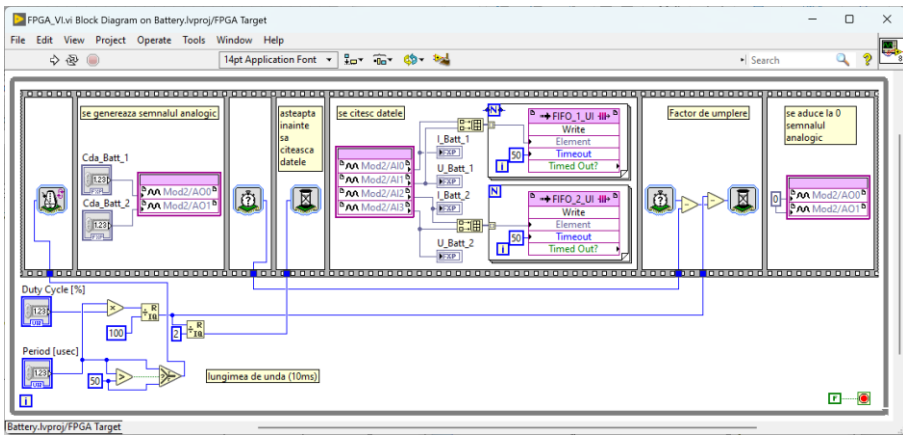


Figura 7.9 – Diagrama bloc, VI-ul care rulează pe FPGA

VI-ul care rulează pe CompactRIO™ RT asigură transferul datelor între *target*-ul RT și calculator, dar poate implementa afișarea grafică și salvarea locală a datelor, oferind un instrument de testare complet autonom.

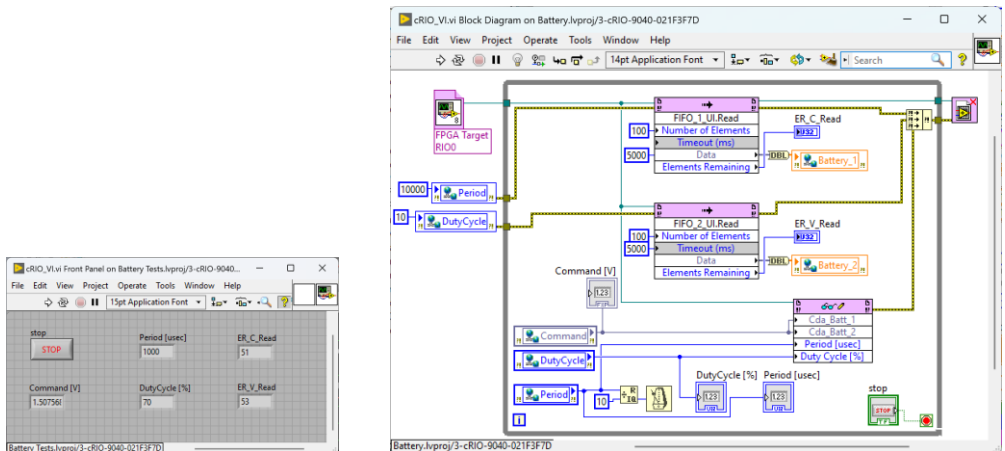


Figura 7.10 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Sunt prezentate două seturi de date pentru experimente: în primul caz frecvența semnalului PWM de comandă este de 100 Hz, factorul de umplere este de 10%, iar amplitudinea semnalului este de 0.5 V; în al doilea caz frecvența semnalului PWM de comandă este de 1 kHz, factorul de umplere este de 20%, iar amplitudinea semnalului este de 2 V. În figurile din dreapta se observă tensiunea totală de 2.6V la bornele a două baterii conectate în serie, care sunt descărcate.

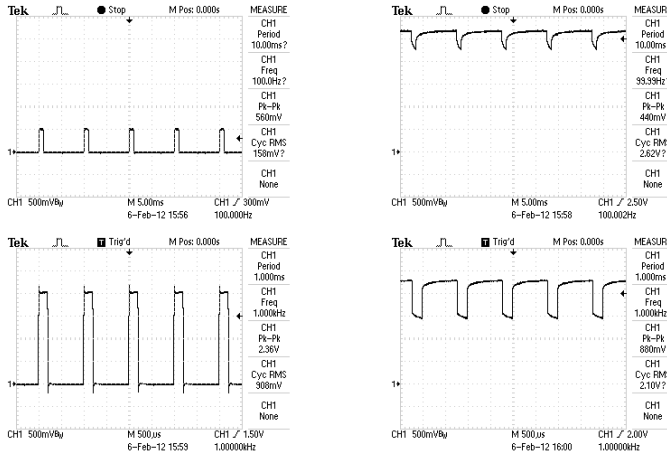


Figura 7.11 – Vizualizarea semnalului PWM de comandă și tensiunea pe bateriile testate

Din interfața cu utilizatorul *Front Panel* se poate configura: frecvența semnalului PWM de comandă din controlul *Period [us]*, factorul de umplere din controlul *Duty Cycle [%]* și amplitudinea din controlul *Amplitude* cu valori cuprinse între 0 mA și 50 mA (sau 0 V și 2.8 V).

Cele două grafice din figura următoare indică valoarea curentului mediu de descărcare [mA] și valoarea tensiunii [V] pe o baterie.

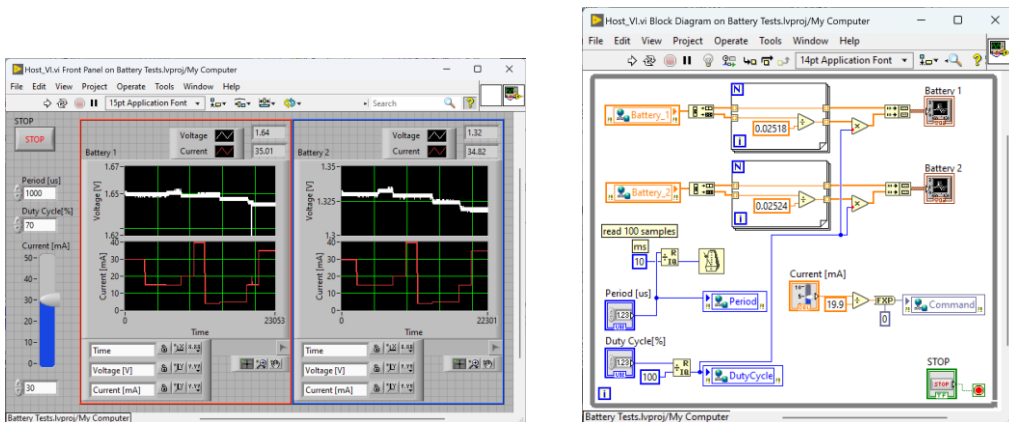


Figura 7.12 – Interfața cu utilizatorul și diagrama bloc, VI-ul care rulează pe calculator

Schimbul de date dintre CompactRIO™ RT și calculator se face prin rețeaua Ethernet, prin variabile partajate de tip *Network-Published*. Cele cinci variabile de tip *Network-Published* au fost configurate astfel:

- variabilele *Battery_1* și *Battery_2* sunt de tip *Array of Double*, iar de la opțiunea *Network* a fost setat un *buffer* cu 200 de elemente și un singur proces *Write*;
- variabila *Command* este de tip *FXP (3.16)*, configurată cu un singur proces *Write*;
- variabilele *Period* și *DutyCycle* sunt de tip *UInt32*.

Cele două FIFO-uri au fost configurate pentru date numerice reprezentate în virgulă fixă (*FXP*).

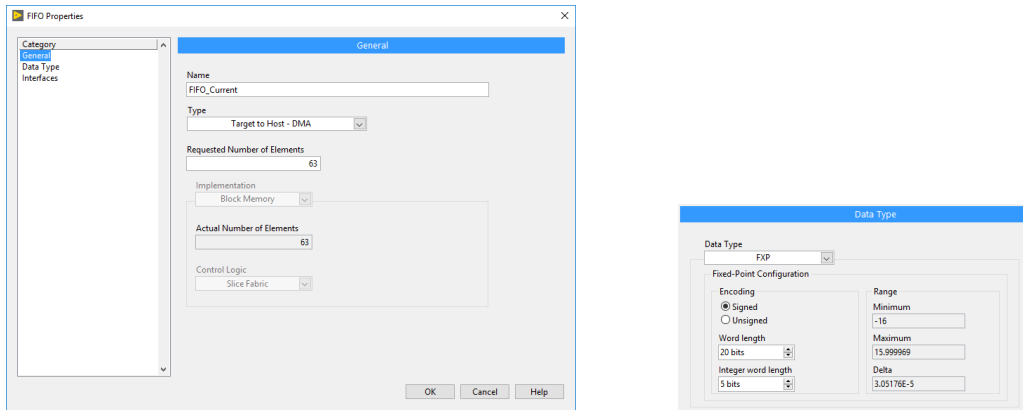


Figura 7.13 – Fereastre de configurare pentru *FIFO_Current* și *FIFO_Voltage*

Există mai multe surse posibile de erori pentru măsurătorile realizate, de care trebuie să ținem cont atunci când analizăm comparativ diferite baterii:

- curbele de referință date de producători în specificații sunt obținute pentru baterii la cel mult 30 de zile de la data fabricației;
- bateriile comparate au date de fabricație diferite sau au fost stocate în condiții diferite;
- valoarea curentului setat nu este exactă. De exemplu valoarea de 500mA setată a avut valori între 497mA și 512mA (-0.6%, +2.4%);
- nu au fost luate în calcul:
 - toleranța rezistențelor utilizate pentru a implementa sarcina comandată și deriva termică;
 - temperatura în încăperea nu este constantă pe toată perioada de testare;
 - rezistența de contact dintre baterii și suportul de baterii când se testează la un curent mare de 500mA.

Aceleași teste nu au fost repetate de mai multe ori, pe același tip de baterii pentru a crește precizia rezultatelor obținute, pentru că experimentele durează de la câteva ore la zeci de ore.

Măsurarea temperaturii bateriilor cu litiu în timpul experimentelor arată o creștere semnificativă a temperaturii bateriilor când curentului de descărcare este

mare, atunci cand tensiunea pe baterie este foarte scăzută (de exemplu 0.8 V pentru un element de 1.5 V):

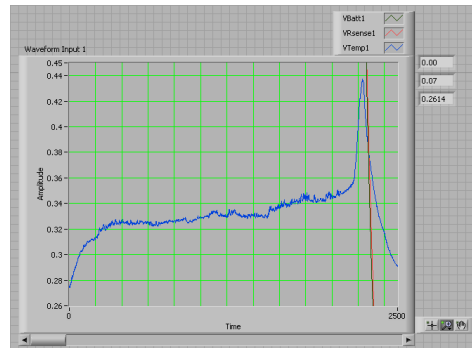


Figura 7.14 – Interfața cu utilizatorul, măsurarea temperaturii bateriilor cu litiu în timpul experimentelor

Măsurarea temperaturii se poate face cu traductorul LM35 care generează la ieșire 10 mV/°C și are o caracteristică liniară [35]. Pentru îmbunătățirea contactului dintre senzorul de temperatură și baterie trebuie utilizată pastă siliconică termoconductoare.

Pentru aplicația de laborator în scopul evitării descărcării rapide a bateriilor testate, curentul a fost limitat la 70 mA. La acest curent de descărcare nu se observă încălzirea bateriilor. Creșterea semnificativă a temperaturii apare la un curent de descărcare mai mare, de exemplu la 500 mA.

La conectarea unor condensatoare de capacitate mare cu rol de *buffer* de curent trebuie să se țină cont că acestea pot avea un curent de descărcare propriu mare. Acest curent poate lua valori de la 1 μ A la 10 μ A și trebuie luat în calcul atunci când se estimează durata de funcționare a echipamentului și se dorește ca timpul de funcționare să fie de ordinul anilor.

8. Amortizarea vibrațiilor într-o grindă metalică **

Scopul acestei lucrări este implementarea unor aplicații pe sistemul CompactRIO™ în LabVIEW™ RT și FPGA pentru măsurarea vibrațiilor unei grinzi metalice (*smart beam*), generarea de semnale sinusoidale pentru determinarea frecvenței de rezonanță și implementarea unui control activ pentru amortizarea vibrațiilor [36].

Se creează un proiect care include CompactRIO™ RT și FPGA, fiecare având câte un VI atașat. Cele două VI-uri vor comunica între ele printr-un *buffer* FIFO sau prin metoda de scriere - citire a valorilor din indicatoare sau controale:

- VI-ul care rulează pe FPGA citește valorile oscilațiilor pe canalul analogic AI3 de la modulul de intrări analogice NI 9239 și scrie valorile de comandă pe canalele AO0 + AO1 și AO2 + AO3 ale modulului de ieșiri analogice NI 9263. Include două bucle *While* care pot avea viteze diferite de execuție, fiecare având o temporizare cu altă perioadă.
- VI-ul care rulează pe RT poate include filtre trece jos, grafice pentru afișarea semnalelor, regulatoare și funcții de temporizare.

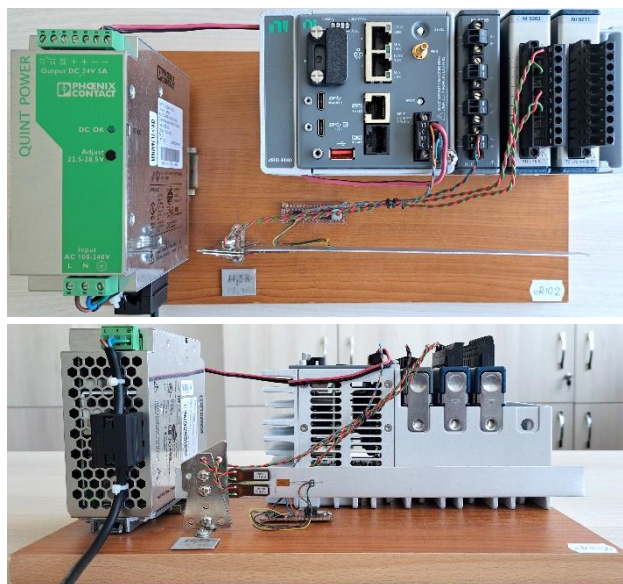


Figura 8.1 – Standul de testare include grinda, elementele de măsură și de control

Pentru măsurarea deplasării sau a vibrațiilor generate de grinda de aluminiu din stand se utilizează două timbre tensiometrice, iar pentru amortizarea vibrațiilor se utilizează două elemente de execuție de tip piezo.

Schema bloc de măsurare și conectare include doi senzori rezistivi de $120\ \Omega$ de tip 1-LY11-3/120 montați într-o punte completă cu 4 rezistențe, conform figurii următoare [37], iar pentru control include două elemente PZT (*Lead*

Zirconate Titanate) de tip P-878.A1 de la firma PI. Elementele PZT au tensiunea de funcționare de la -20 V la +120 V [38].

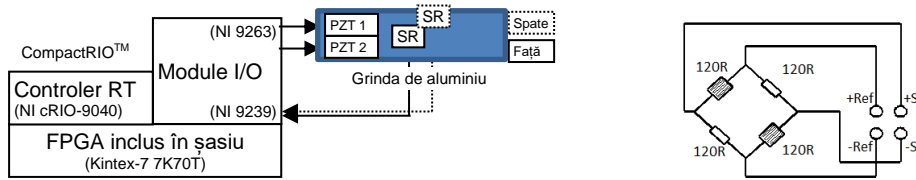


Figura 8.2 – Schema bloc de conectare, puntea de măsurare

Se vor vizualiza toate semnalele cu un osciloscop TDS1012B, C sau DSO1012A pentru a verifica funcționarea corectă a programelor. Osciloscopul este utilizat pentru determinarea frecvenței de rezonanță a grinzii. Frecvența de rezonanță depinde de mai mulți factori.

Grinda este realizată din aluminiu, iar dimensiunile ei sunt: 23 cm lungime, 2 cm lățime și 1.1 mm grosime. În timpul testelor grinda de aluminiu se încovoiaie cu mâna de capătul liber, **maxim 5-10 mm** și apoi se eliberează în timp cât mai scurt. Este de preferat ca arcuirea să fie efectuată de fiecare dată în altă direcție pentru că încovoierea repetată cu 5-10 mm în aceeași direcție determină deformarea permanentă a grinzii de aluminiu în direcția respectivă.

Prima aplicație implementată este simplă și permite generarea unui semnal sinusoidal cu frecvență reglabilă. Se măsoară amplitudinea oscilațiilor grinzii și se determină frecvența de rezonanță.

Frecvența de rezonanță se citește de pe ecranul osciloscopului și este valoarea la care amplitudinea de oscilație a capătului liber al grinzii este maximă. Orientativ această frecvență este în jurul valorii de 17.3 Hz. Se notează valoarea obținută în urma experimentului realizat.

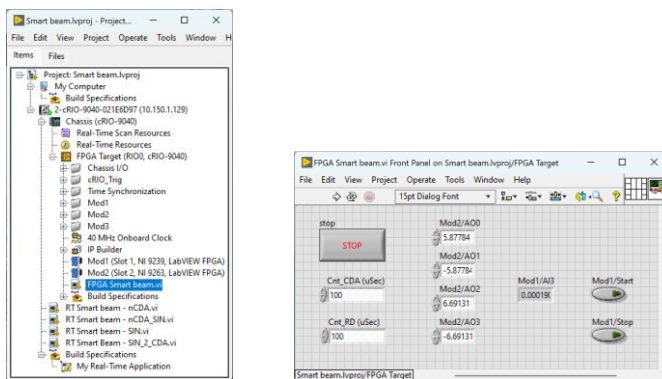


Figura 8.3 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe FPGA

În diagrama bloc a aplicației care rulează pe CompactRIO™ FPGA sunt incluse două bucle *While*. Fiecare buclă are propria perioadă de eşantionare prin apelarea funcției *Loop Timer*. Prima buclă este utilizată pentru comanda

elementelor PZT cu patru canale analogice de ieșire ale modului NI 9263: AO0-AO1 pentru comanda elementului PZT1; AO2-AO3 pentru comanda elementului PZT2. Cele două canale diferențiale de ieșire generează semnale de aceeași amplitudine, dar de semn diferit, pentru care rezultă o diferență de tensiune cât mai mare aplicată pe elementele PZT.

A doua buclă este utilizată pentru citirea semnalului diferențial pe canal analogic AI3 de la modulul NI 9239. Modulele de intrare analogică NI 9230 sau NI 9239 necesită un semnal de *trigger* extern pentru a porni achiziția, iar acesta se generează prin apăsarea butonului *Mod1/Start* din interfață: butonul trece în starea *ON* pentru un timp scurt de sute de milisecunde, apoi este comutat în starea *OFF*.

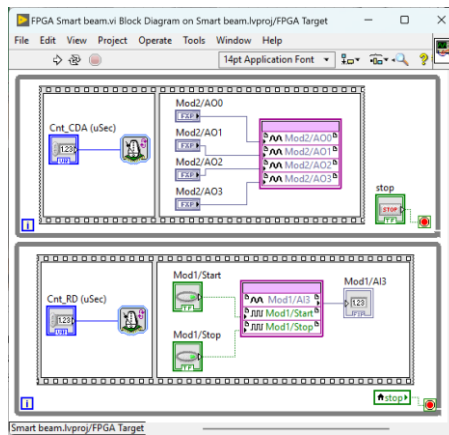


Figura 8.4 – Diagrama bloc, VI-ul care rulează pe FPGA

Pentru determinarea frecvenței de rezonanță se modifică valoarea funcției de întârziere, controlul *Count (usec)* și se măsoară amplitudinea de oscilație la capătul liber al grinzii de aluminiu sau alternativ se citește de pe indicatorul grafic *Waveform (Strain gauge)* amplitudinea semnalului cu cea mai mare valoare.

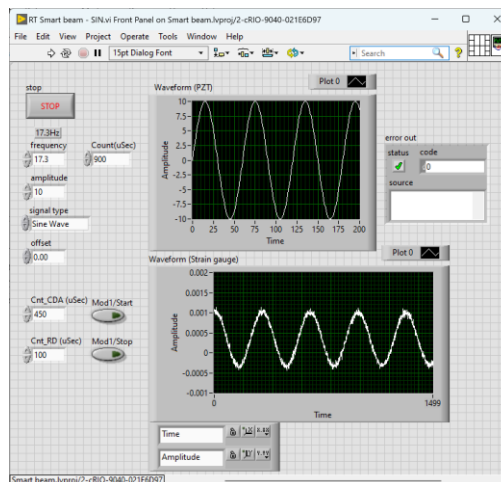


Figura 8.5 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT

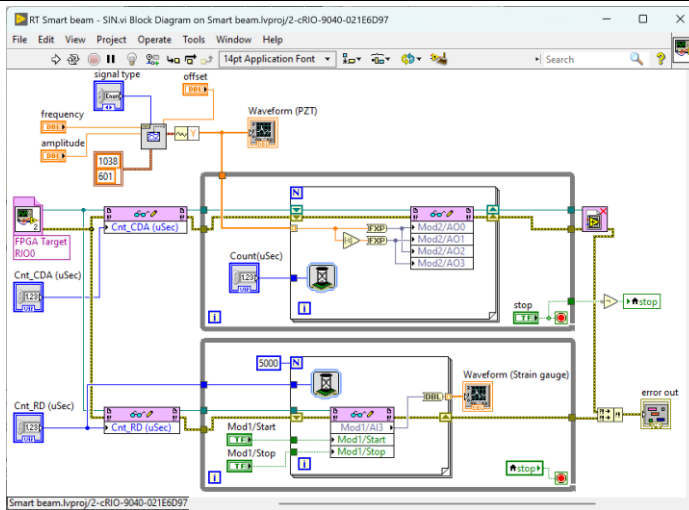


Figura 8.6 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Aplicația următoare permite generarea a două semnale sinusoidale care pot avea aceeași fază sau pot fi defazate de la 0° la 180° . Se poate urmări modul în care efectele celor două sinusuri de comandă se însumează și amplitudinea oscilațiilor este maximă când defazajul este 0° sau efectele celor două sinusuri de comandă se scad și amplitudinea oscilațiilor este minimă când defazajul este 180° .

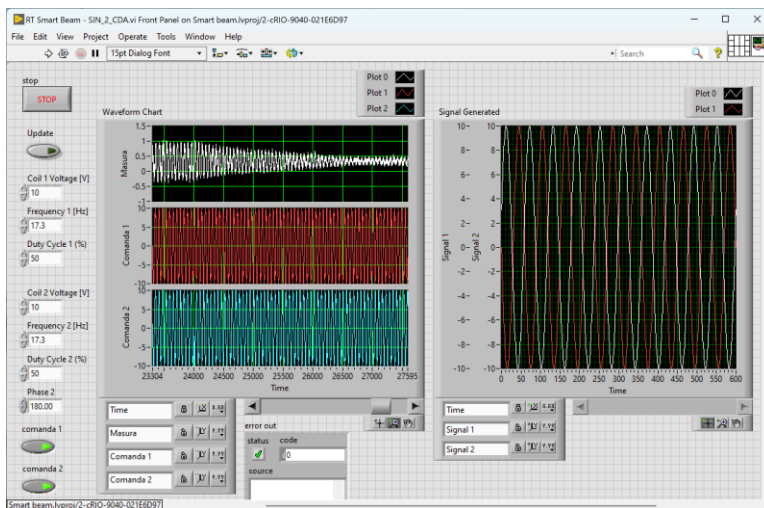


Figura 8.7 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT

VI-ul include funcțiile de generare de semnale sinusoidale în bucla principală *While*. Această implementare spre deosebire de cea anterioară, are dezavantajul ca întrerupe la un moment dat generarea continuă a semnalului în momentul apelării funcției de calcul a semnalelor generate.

Pentru a evita această întrerupere se includ elemente de memorare ale vectorului de valori de la o iterație la alta, numite *Shift Register*. Acestea elemente de memorare sunt plasate pe marginile verticale ale buclei principale *While*.

Utilizarea structurii *Case* permite ca funcțiile de generare de semnale să fie apelate doar atunci când utilizatorul apasă butonul *Update* din interfață.

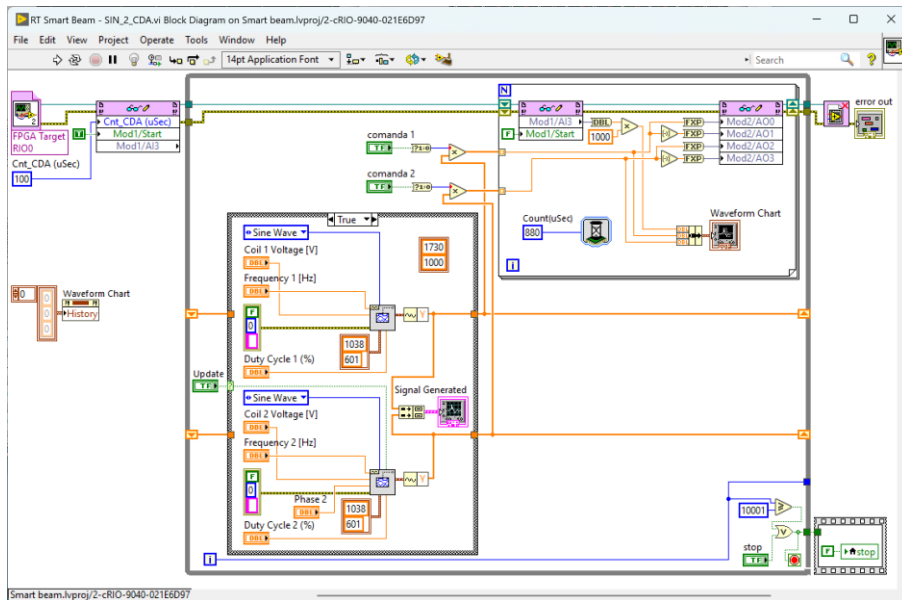


Figura 8.8 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

Aplicația următoare permite implementarea mai multor regulatoare, selectarea și testarea lor. Se măsoară amplitudinea oscilațiilor și timpul în care se face amortizarea liberă a oscilațiilor sau când este activat un regulator – butonul *comanda* este *ON*. În figura din dreapta se observă semnalul de perturbație.

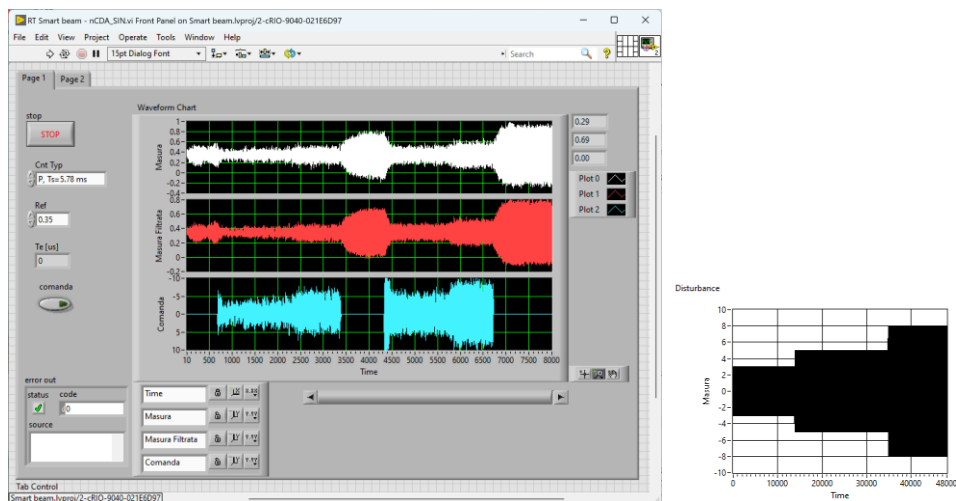


Figura 8.9 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT și perturbația

Din selectorul pentru tipul de regulator (*Cnt Typ*) se poate alege o anumită strategie de reglare. În primele două grafice este afișată amplitudinea oscilațiilor (*Măsura* și *Măsura Filtrată*), iar în graficul al treilea comanda regulatorului de tip

P cu perioada de eşantionare $T_e = 5.78$ ms. Semnalul de perturbație are oscilație continuă cu amplitudinea de ± 3.0 V, ± 5.0 V și ± 8.0 V, vizibilă în *tab-ul Page 2*.

În prima buclă este implementată măsurarea oscilațiilor și algoritmul de reglare, iar în bucla a doua generarea continuă a semnalului de perturbație.

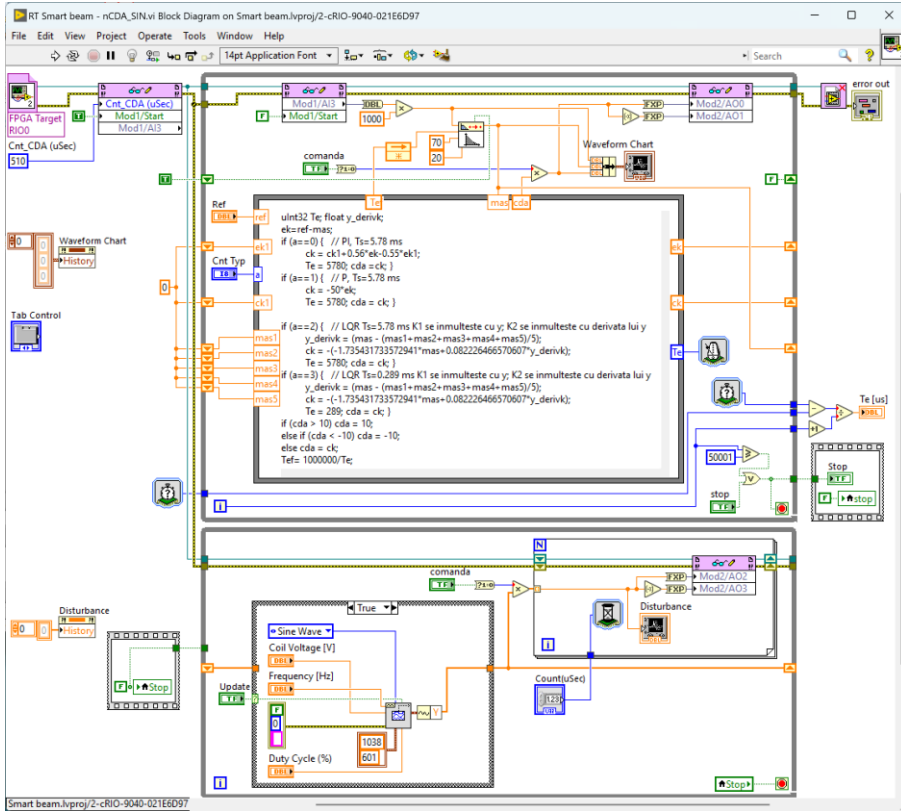


Figura 8.10 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

În aplicația următoarea sunt comandate ambele elemente PZT pentru amortizarea oscilațiilor generate prin deplasarea scurtă și apoi eliberarea grinzii.

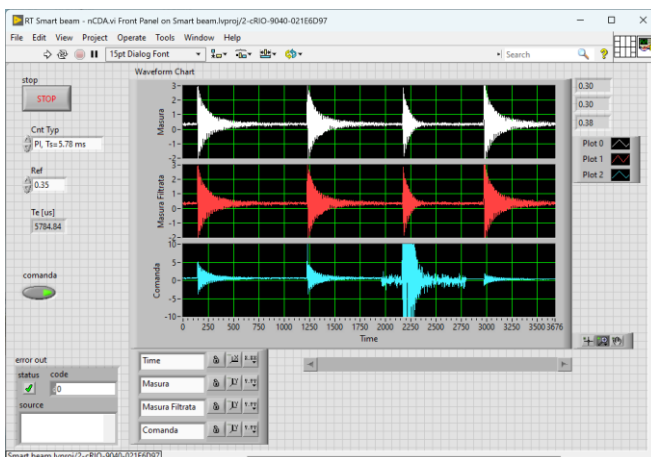


Figura 8.11 – Interfața cu utilizatorul, VI-ul care rulează pe CompactRIO™ RT

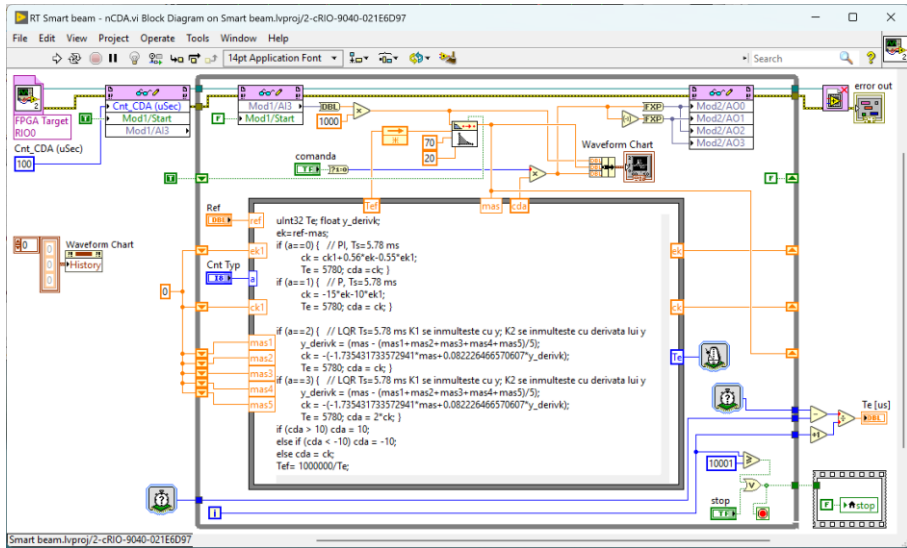
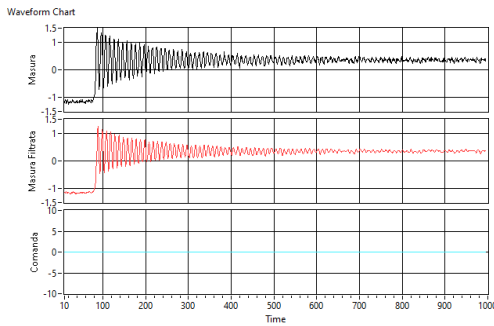
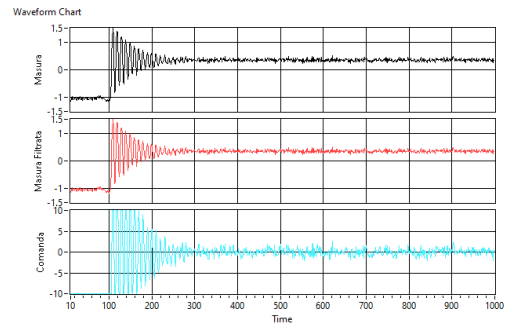


Figura 8.12 – Diagrama bloc, VI-ul care rulează pe CompactRIO™ RT

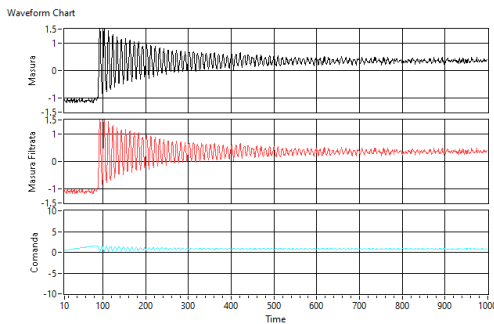
Amplitudinea oscilațiilor grinzii de aluminiu este maximă când comanda este zero. Regulatele calculate sunt pentru aceeași perioadă de eșantionare de 5.78 ms. Acordarea regulateleor pentru ca amortizarea să fie eficientă nu este simplă. Răspunsul sistemului în diferite scenarii de testare este prezentat în figurile următoare:



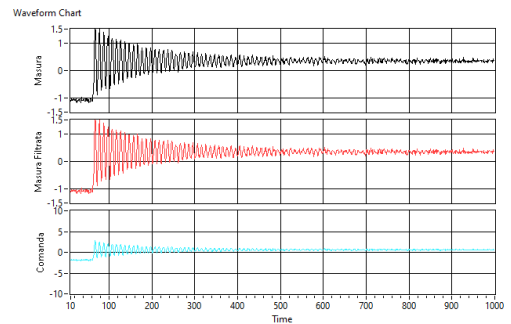
Răspuns liber, $T_e = 5.78$ ms



Regulator P, $T_e = 5.78$ ms



Regulator PI, $T_e = 5.78$ ms



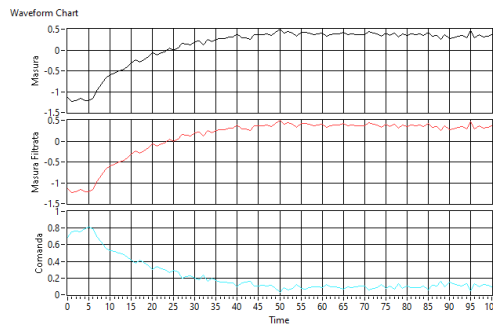
Regulator LQR, $T_e = 5.78$ ms

Figura 8.13 – Răspunsuri ale procesului cu diferite regulatele

Este nevoie de o filtrare cât mai bună a semnalului măsurat, acesta având amplitudine foarte mică și zgomot de valoare mare. În aplicație a fost implementat un filtru trece-jos de ordinul doi.

Utilizarea unui element PZT pentru generarea automată a perturbației este justificată de diferențele care apar în amplitudinea de oscilație la arcuirea grinzii cu mâna.

☺ În figura următoare, sistemul funcționează, grinda de aluminiu oscilează la fel ca și în cazurile anterioare, iar perioada de eșantionare are valoarea de 57.8 ms. Explicați de ce pe graficul din aplicație nu sunt vizibile oscilațiile? Care este valoarea maximă a perioadei de eșantionare pe care o putem alege?



Regulator PI, $T_e = 57.8$ ms

Figura 8.14 – Răspunsul procesului, oscilațiile nu sunt afișate de aplicație

După oprirea programului de la butonul *stop* în indicatorul T_e [μ s] este afișată valoarea medie a perioadei de eșantionare în microsecunde, realizată de sistem. Această valoare este determinată de performanțele implementării, de sistemul de operare și de echipamentul utilizat. De exemplu pentru o valoare impusă de 5.78 ms valoarea medie obținută pentru 10.000 de execuții ale buclei principale a fost de 5784.84 μ s, adică aproximativ 5.785 ms.

☺ Ce factori credeți că influențează modificarea perioadei de eșantionare?

Se pot determina și alte frecvențe la care grinda rezonază și care sunt multiplii ale primei frecvențe, însă amplitudinea de oscilație este semnificativ mai mică.

☺ Ce factori credeți că influențează frecvența de rezonanță?

9. Controlul intensității luminoase a LED-urilor

Scopul acestei lucrări este implementarea unor aplicații în LabVIEW™ RT pe calculatorul industrial PXI™ cu sistem de operare de timp real Phar Lap ETS, pentru testarea funcționării LED-urilor [39] și controlul intensității luminoase. Se utilizează un modul dedicat de comandă pentru LED-uri albe de putere. Aplicațiile implementate rulează pe o platformă cu resurse semnificative de calcul, care permite implementarea controlului automat a intensității luminoase pentru o clădire cu zeci de săli.

Se creează un proiect care include un calculator industrial NI PXI-8106, NI PXI-8105 sau NI PXIe-8106 având unul sau mai multe VI-uri atașate. Primul VI prezentat în continuare include două bucle *While* care au următoarele funcții:

- prima buclă *While* utilizează linia digitală DO0 a portului P0 din modulul de extensie NI PXI-6280 sau NI PXI-6281 pentru generarea semnalului PWM de comandă pentru reglarea intensității luminoase a LED-urilor;
- în a doua buclă *While* se citesc intrările analogice AI0 corespunzând sensorului de lumină și AI1 pentru sensorul de temperatură LM35 [35]. Sensorul de temperatură are caracteristică liniară și generează la ieșire 10 mV/°C. Linia digitală DO1 a portului P0 generează tensiune în starea logică „1” pentru alimentarea sensorului de temperatură. Setările pentru canalele de achiziție au fost realizate în *NI MAX*.

Pentru comanda LED-urilor se utilizează un kit de evaluare cu LED-uri albe de putere, de la Texas Instruments, modelul TL4242EVM [40].

ATENȚIE: *standul funcționează cu tensiune de 12 V! Nu alimentați cu tensiune mai mare.*

ATENȚIE: *evitați să priviți direct spre LED-uri pentru că luminează intens!*



Figura 9.1 – Standul de testare, calculatorul PXI™ și modulul cu LED-uri albe de putere

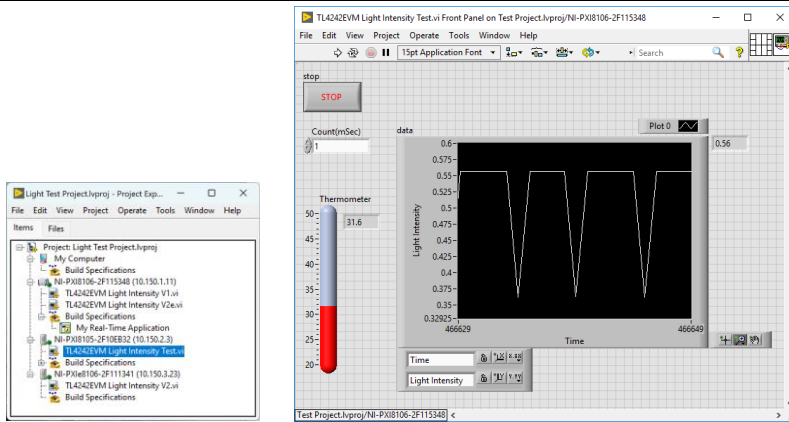


Figura 9.2 – Fereastra de proiect și interfața cu utilizatorul, VI-ul care rulează pe PXI™ (var. 1)

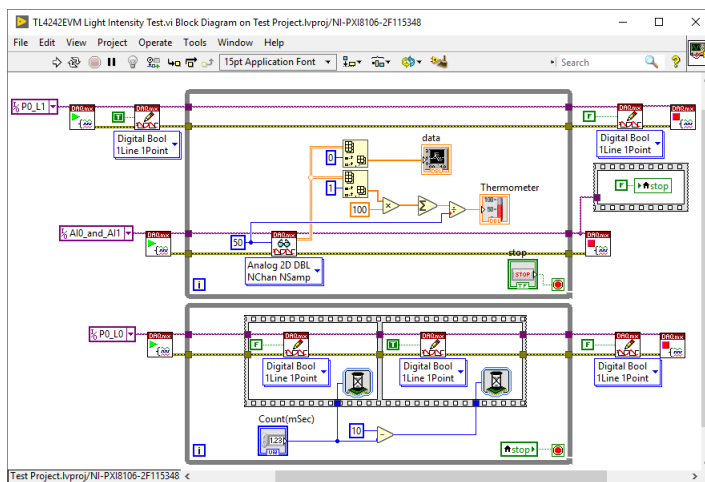


Figura 9.3 – Diagrama bloc, VI-ul care rulează pe PXI™ (varianta 1)

Pentru îmbunătățirea contactului dintre senzorul de temperatură și unul din LED-uri sau radiatorul acestuia, este recomandată utilizarea de pastă siliconică termoconductoare. În figura următoare este prezentată caracteristica *temperatură - intensitate luminoasă* atunci când comanda este maximă:

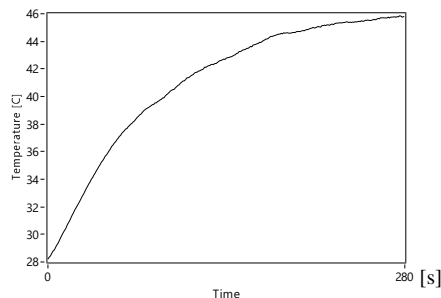


Figura 9.4 – Caracteristica *temperatură - intensitate luminoasă* pentru comandă maximă

Ridicați două caracteristici de temperatură, pe durata a 5 minute, în funcție de intensitatea luminoasă. Orientativ temperatura LED-urilor poate să ajungă la 50 °C atunci când intensitatea luminoasă este maximă.

Figurile următoare prezintă modurile în care trebuie configurate task-urile analogice de intrare corespunzătoare pentru *AIO_and_AII* în *NI MAX*. Task-urile digitale pentru *P0_LO/LI* sunt foarte simplu de configurat și nu sunt prezentate.

Linia digitală L1 (portul P0) este utilizată pentru alimentarea senzorului de temperatură cu o tensiune de la +3.0 V la +5.0 V. Modulele NI PXI-6280 / 6281 nu includ pini de alimentare cu tensiune a dispozitivelor externe. Acest mod de alimentare nu este recomandat, deși senzorul are un consum tipic foarte redus de 60 μ A și este acceptabil doar pentru un experiment de laborator.

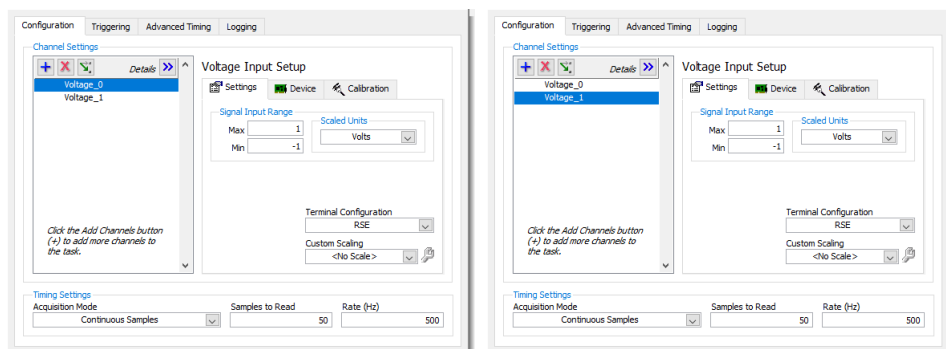


Figura 9.5 – Configurarea task-urilor analogice în *NI MAX*

Pentru controlul intensității luminoase varianta următoare de VI include direct în aplicație setările implementate anterior în *NI MAX*. În acest fel rularea programului pe diferite modele de calculatoare industriale PXITM este mult mai ușor de realizat.

Funcțiile de configurare pentru canalele fizice de achiziție de semnale și control sunt plasate în exteriorul buclelor *While* și se apelează o singură dată la pornirea programului.

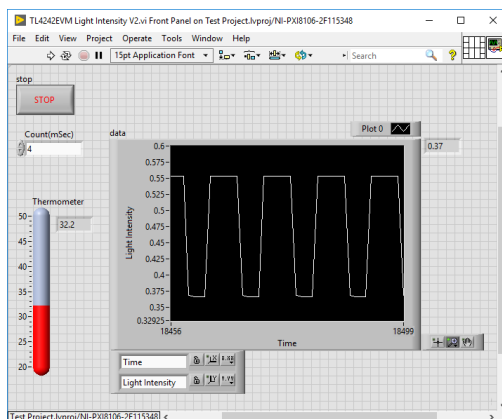


Figura 9.6 – Interfața cu utilizatorul, VI-ul care rulează pe PXITM (varianta 2)

Atenție: nu pot fi apelate în paralel două task-uri care utilizează aceeași resursă fizică: de exemplu nu se pot inițializa separat două task-uri analogice, pentru că modulele de extensie NI PXI-6280 sau NI PXI-6281 includ un singur convertor analog-numeric și un multiplexor cu 16 intrări.

În exteriorul primei bucle *While* se inițializează liniile analogice astfel: AI0 cu domeniul de intrare de ± 1 V pentru citirea intensității luminoase a unui LED cu o celulă foto și AI1 cu domeniul de intrare de ± 1 V pentru citirea temperaturii unui LED cu senzorul de temperatură LM35. În interiorul buclei sunt citite și afișate valorile pentru intensitatea luminoasă și temperatură.

În a doua buclă *While* este implementat un generator de semnal de tip PWM pentru comanda intensității luminoase a LED-urilor pe linia digitală L0 (portul P0). Așa cum este implementată bucla, se generează un semnal PWM cu frecvența de 100 Hz și care oferă 10 valori de intensitate care pot să fie selectate din controlul *Count (mSec)*. Funcțiile de întârziere *Wait* pentru timp real sunt configurate cu milisecunde și reprezentare pe 32 biți.

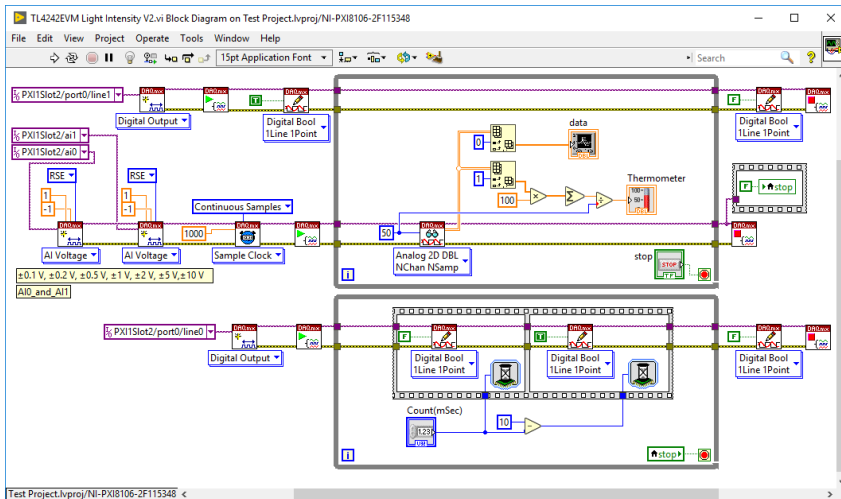


Figura 9.7 – Diagrama bloc, VI-ul care rulează pe PXI™ (varianta 2)

😊 Completați aplicația astfel încât să realizeze un control automat al intensității luminoase în funcție de o valoare stabilă de utilizator din interfață și de valoarea măsurată cu senzorul de lumină. Testați aplicația cu LED-ul de la telefonul mobil.

Bibliografie

1. NI PXI-8106 *User Manual*, December 2008, 372096B-01, <https://www.ni.com/docs/en-US/bundle/372096b/resource/372096b.pdf>
2. NI PXI-8105 *User Manual*, March 2007, 371755C-01, <https://www.ni.com/docs/en-US/bundle/371755c/resource/371755c.pdf>
3. NI PXIe-8106 *User Manual*, January 2007, 372095A-01, <https://www.ni.com/docs/en-US/bundle/372095a/resource/372095a.pdf>
4. PXI-6281 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/pci-pxi-usb-6281-specs/page/specs.html>
5. PXI-6281 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/pci-pxi-6280-specs/page/specs.html>
6. PXI-6733 *Analog Output Series User Manual*, January 2017, 370735F-01, <https://www.ni.com/docs/en-US/bundle/ni-6711-6713-6722-6723-6731-6733-features/resource/370735f.pdf>
7. PXI-7811 *Multifunction Reconfigurable I/O User Manual*, October 2023, 370489H-01, <https://www.ni.com/docs/en-US/bundle/multifunction-rio-user-manual/resource/multifunction-rio-user-manual.pdf>
8. PXI-4351, *High-Precision Temperature and Voltage Meters*, December 1998, 321566C-01, https://www.artisanng.com/info/National_Instruments_PC_4350_User_Manual.pdf
9. PXI-5114 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/pxi-5114-specs/page/specs.html>
10. cRIO-9040 *Specifications*, Updated 2023-09-28, <https://www.ni.com/docs/en-US/bundle/crio-9040-specs/page/specs.html>
11. cRIO-9053 *Specifications*, Updated 2022-11-30, <https://www.ni.com/docs/en-US/bundle/crio-9053-specs/page/specs.html>
12. NI-9239 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/ni-9239-specs/page/specs.html>
13. NI-9381 *Specifications*, Updated 2023-07-10, <https://www.ni.com/docs/en-US/bundle/ni-9381-specs/page/specs.html>
14. NI-9211 *Specifications*, Updated 2023-06-14, <https://www.ni.com/docs/en-US/bundle/ni-9211-specs/page/specs.html>
15. NI-9215 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/ni-9215-specs/page/specs.html>
16. NI-9201 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/ni-9201-specs/page/specs.html>
17. NI-9401 *Specifications*, Updated 2023-07-18, <https://www.ni.com/docs/en-US/bundle/ni-9401-specs/page/specs.html>
18. NI-9403 *Specifications*, Updated 2023-07-18, <https://www.ni.com/docs/en-US/bundle/ni-9403-specs/page/specs.html>
19. NI-9263 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/ni-9263-specs/page/specs.html>
20. NI-9230 *Specifications*, Updated 2023-08-14, <https://www.ni.com/docs/en-US/bundle/ni-9230-specs/page/specs.html>

-
21. NI-9162, *CompactDAQ Chassis*, <https://www.ni.com/ro-ro/support/model.usb-9162.html>
 22. NI ELVIS II+, *Engineering Laboratory Virtual Instrumentation Suite*, <https://www.ni.com/ro-ro/support/model.ni-elvis-ii-.html>
 23. S. C. Folea, G. Moiş, C. I. Muresan, L. Miclea, R. De Keyser and M. N. Cirstea, "A Portable Implementation on Industrial Devices of a Predictive Controller Using Graphical Programming," in *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 736-744, April 2016, doi: 10.1109/TII.2016.2532118.
 24. C. I. Muresan, S. Folea and G. Moiş, "Optimal implementation of advanced control methods on FPGA targets," *The 2013 RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, Hanoi, Vietnam, 2013, pp. 209-214, doi: 10.1109/RIVF.2013.6719895.
 25. TPS62260LED-338, *Three-Color LED Driver Evaluation Module (EVM)*, User's Guide, Literature Number: SLVU240A, May 2008–Revised October 2008
 26. C. I. Muresan, S. Folea, G. Moiş, E. H. Dulf, "Development and implementation of an FPGA based fractional order controller for a DC motor," *Mechatronics*, Volume 23, Issue 7, 2013, Pages 798-804, ISSN 0957-4158, <https://doi.org/10.1016/j.mechatronics.2013.04.001>.
 27. C. I. Muresan, G. Moiş, S. Folea and C. Ionescu, "Alternative implementations of a fractional order control algorithm on FPGAs," *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, 2013, pp. 1-6, doi: 10.1109/ReConFig.2013.6732269.
 28. Texas Instruments, L293, *Quadruple Half-H Drivers*, Slrs008c – September 1986 – Revised November 2004
 29. S. Folea, G. Moiş and L. Miclea, "Power quality measurement system using FPGAs," *2012 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, Brasov, Romania, 2012, pp. 1280-1285, doi: 10.1109/OPTIM.2012.6231855.
 30. LEM, *Voltage Transducer LV 25-P*, Technical Specification Datasheet, 080729/17.
 31. LEM, *Current Transducer LA 25-NP*, Technical Specification Datasheet, 980909/8.
 32. S. Folea, G. Moiş, L. Miclea and D. Ursutiu, "Battery lifetime testing using LabVIEW™," *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, Bilbao, Spain, 2012, pp. 1-6, doi: 10.1109/REV.2012.6293122.
 33. Energiser LR6 / E91, Specifications, EBC - 1102OALKPOW2, <https://data.energizer.com/pdfs/alk-power-aa.pdf>
 34. INA138, *High-Side Measurement Current Shunt Monitor*, SBOS122E–DECEMBER1999–REVISEDDECEMBER2017, <https://www.ti.com/lit/gpn/INA138>
 35. LM35, *Precision Centigrade Temperature Sensors*, SNIS159H–AUGUST 1999–REVISED DECEMBER 2017, <https://www.ti.com/lit/ds/symlink/lm35.pdf>
 36. C. I. Muresan, S. Folea, I. R. Birs and C. M. Ionescu, "Fractional order modeling and control of a smart beam," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Maui, HI, USA, 2017, pp. 1517-1523, doi: 10.1109/CCTA.2017.8062672.
 37. Strain Gauges, *Specifications*, <https://spectromas.ro/wp-content/uploads/2018/10/S01265.pdf>
 38. PI, *DuraAct Power Patch Transducer*, 11.05.2023, https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/datasheets/P-878-Datasheet.pdf
 39. S. Folea, S. Enyedi, L. Miclea and H. Hedesiu, "Reconfigurable test platform for modular embedded systems in manufacturing processes," *2015 10th International Design & Test Symposium (IDT)*, Amman, Jordan, 2015, pp. 72-77, doi: 10.1109/IDT.2015.7396739.
 40. TL4242EVM-543, *User's Guide*, SLVU340–October2009, <https://www.ti.com/lit/pdf/slvu340>
-



La început, pasiunea pentru electronică a însemnat demontarea unor echipamente electronice. Componentele obținute astfel le-am sortat și depozitat într-un dulap special construit din cutii de chibrituri lipite între ele.

În timpul liceului am realizat diferite proiecte în domeniul electronicii de amator: sirene multiton, lumină dinamică, interfon, radioul fără alimentare, semnalizare pentru bicicletă, alimentatoare stabilizate, ceas programabil, amplificatoare de putere, semafor, sonde de test, orgă de lumini analogică și digitală.

La terminarea facultății, pentru realizarea proiectului de diplomă am proiectat o placă de achiziții de semnale sub formă modulară. După susținerea proiectului am fost angajat la universitate ca asistent universitar și mai târziu la National Instruments România ca inginer de cercetare.

În cadrul companiei am proiectat mai multe sisteme de achiziții de semnale, dar și un minicalculator modular, sub forma unui cub cu latura de 4 cm, implementare brevetată în SUA.

Până în prezent am proiectat și realizat peste o sută treizeci de dispozitive electronice. O parte dintre acestea au fost incluse în această colecție de aplicații.