

Laborator 7

SIMULAREA FUNCȚIONĂRII CIRCUITULUI DE SEMNALIZARE AL AUTOVEHICULULUI CU AJUTORUL INSTRUMENTAȚIEI VIRTUALE

1. Scurt istoric al instrumentației virtuale

Instrumentația virtuală reprezintă un ansamblu de concepte și tehnologii care permite interacțiunea unui utilizator cu un mediu virtual simulat cu ajutorul unui calculator. Instrumentația virtuală este un domeniu interdisciplinar care combină: ingineria mecanică, electronică, automatică, robotică, tehnologii informaționale, fizica, medicina, și factorii umani.

Definiție:

Un instrument virtual reprezintă un ansamblu de componente software și hardware cu ajutorul cărora se achiziționează și analizează date, se monitorizează și se procesează semnale, se efectuează automatizarea și controlul industrial prin intermediul calculatorului personal.

Apariția instrumentației virtuale este strâns legată de apariția calculatoarelor personale și a condus la revoluționarea instrumentelor de măsurare. Folosind calculatoare, aplicații software, plăci de achiziție și drivere pentru acestea se pot realiza diferite sisteme de măsurare bazate pe software cu funcții definite de utilizator, care pot înlocui cu succes instrumentele clasice.

Punem calculatorul în instrument sau punem instrumentul în calculator?

Producătorii au integrat în instrumentele de sine-stătătoare calculatoare pentru a realiza diferite funcții. Construind mai multe componente pentru instrumentație, producătorii realizează instrumente în PC-

uri. Ce se folosea într-un sistem modular real-time sau cu sistem de operare propriu a devenit acum un PC sau un PC înăuntrul unei cutii. Punând PC-ul în interiorul cutiei producătorii au transferat unele beneficii ale PC-ului la utilizator. În plus, producătorii pot ușor să aducă noi caracteristici asemănătoare PC-urilor la instrumente cu un minim efort și costuri.

Alternativ, punând instrumentul în interiorul calculatorului, utilizatorul avea control deplin asupra calculatorului, aplicației soft și a instrumentelor care erau create. Pe măsură ce tehnologia a evoluat, utilizatorul a putut profita imediat de aceste avantaje mai repede decât producătorul de instrumente clasice i le oferea, cu sau fără un onorariu. Așa cum calculatoarele sunt în creștere din punct de vedere al performanțelor utilizatorii pot să cumpere un calculator de ultimă generație și să refolosească placa de măsură și software-ul pentru a crește performanțele.

Utilizând arhitectura unui sistem virtual al Sistemelor de Măsurare Digitale, utilizatorii pot să aleagă plăci cu cost mic cu achiziție brută sau cu posibilitatea de stimuli și să creeze restul prin soft. În plus, tehnologia cu soft modular ca Active X și Instrumente Virtuale Interschimbabile permit utilizatorului să construiască instrumente virtuale. Mediul de dezvoltare grafic cum ar fi LabVIEW profită din plin de avantajele soft-ului modular și permit utilizatorului să creeze instrumente la alegere (indiferent de platforma hard sau magistrala de interfață). Evoluția continuă în tehnologia convertoarelor analog-digitale, instrumentele cu computer de bază și placă de achiziție de date continuă să rivalizeze cu aparatele de sine-stătătoare pe care le poate face utilizatorul. Pentru că Sistemele de Măsură Digitale profită din plin de avantajele ultimelor tendințe în tehnologia convertoarelor analog-digitale și în tehnologia PC-urilor costurile unui asemenea sistem scad.

2. Prezentarea mediului de programare grafic LabVIEW

LabVIEW [1] [2], un limbaj grafic care provine dintr-o prescurtare din limba engleză (**L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench), este un software de proiectare care oferă atât inginerilor, cât și oamenilor de știință, instrumentele necesare pentru a crea și implementa sisteme de măsură și control. Utilizarea acestui limbaj de programare s-a remarcat în mod special în cazul instrumentației de măsurare bazată pe tehnica de calcul. Programarea unui sistem de măsurare cu ajutorul calculatorului într-un limbaj (C++, Pascal, etc.) consumă foarte mult timp. Interfețele acestor limbaje clasice sunt de multe ori neclare și greu de înțeles. Prin utilizarea limbajului grafic

LabVIEW, rapiditatea programării crește foarte mult datorită introducerii unei interfețe grafice mai intuitive.

Mediul de programare LabVIEW a fost lansat pe piață de firma National Instruments, s-a dezvoltat începând cu anul 1983 și s-a perfecționat în mod continuu.

LabVIEW se găsește printre primele limbaje de programare grafică utilizate în aplicațiile de achiziții de date cu tehnica de calcul. Cu ajutorul plăcilor de achiziții de date, fluxul de date numerice sau analogice provenite de la diverse traductoare poate fi prelucrat sau analizat. Prelucrarea datelor analogice sau numerice cu ajutorul limbajului de programare, permite simularea aparatelor de măsură și control sau permite crearea instrumentelor virtuale.

În modul clasic de lucru pe un sistem de calcul cu mouse, lansarea mediului de lucru se realizează fie prin alegerea pictogramei corespunzătoare mediului de lucru (figura 1) fie prin alegerea succesivă a opțiunilor: Start / NI LabVIEW 2016 (32 - bit) ceea ce are ca efect deschiderea mediului de lucru.



Fig. 1. Pictograma mediului de programare grafic LabVIEW.

După lansarea programului va fi afișată fereastra din figura 2 din care utilizatorul trebuie să selecteze dacă dorește să deschidă un proiect creat anterior și salvat într-o locație pe computer sau utilizatorul dorește să creeze un proiect nou.

Dacă s-a selectat crearea unui proiect nou se va deschide o nouă fereastră ca în figura 3 de unde se va selecta „Blank Project” dacă vor fi incluse mai multe fișiere în cadrul proiectului sau se va selecta „Blank VI”, caz în care tot programul va fi inclus într-un singur fișier după care se apasă butonul „Finish”.

Chiar dacă va fi un singur fișier utilizatorul poate să își creeze în cadrul lui sub VI-uri care pot fi afișate ca pictograme particularizate (blocuri).

Dacă se va selecta „Sample Projects” utilizatorul poate deschide câteva exemple incluse în mediul de programare LabVIEW, unde poate vedea

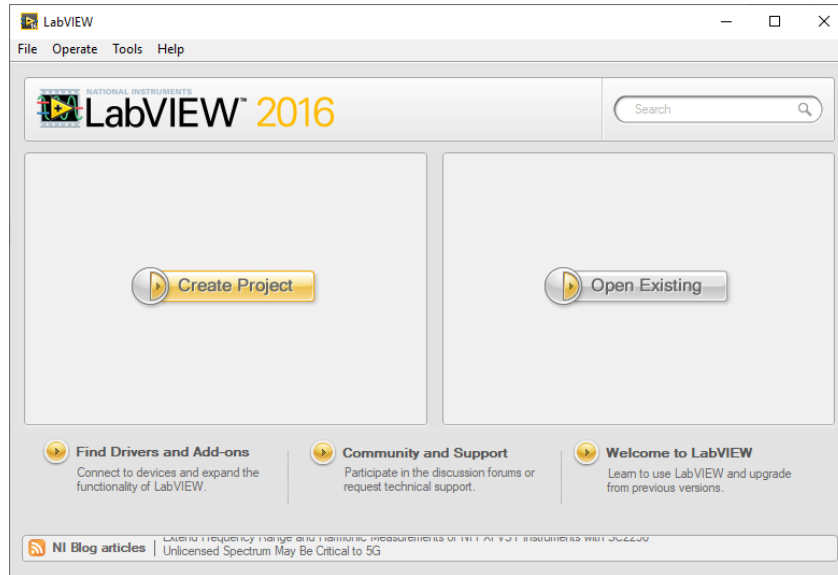


Fig. 2. Fereastra din care se va alege un proiect deja creat sau să se creeze unul nou.

setările necesare diverselor instrumente predefinite din mediul de programare. De exemplu pentru portul USB (care este o emulare a portului serial RS 232) se poate seta numărul portului, viteza de transfer a datelor, dacă există biți de paritate sau nu, etc.

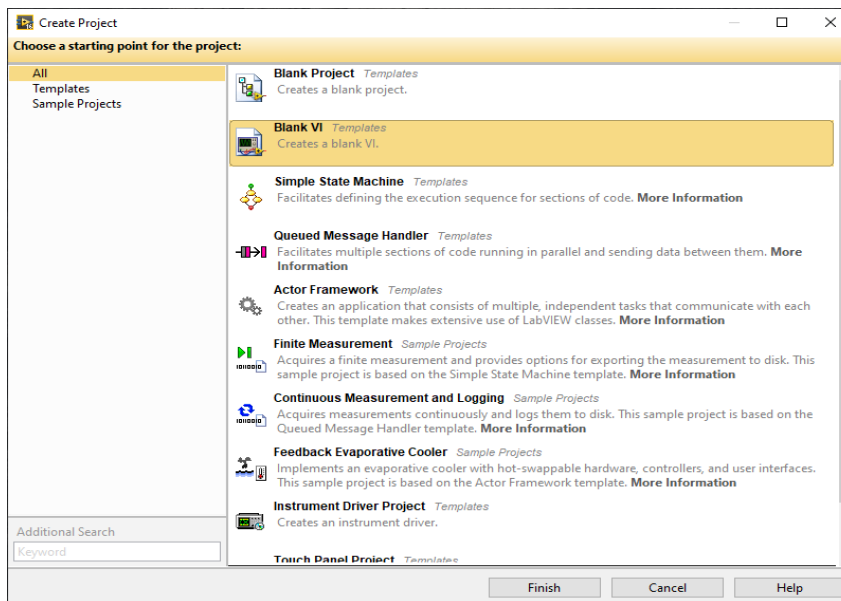


Fig. 3. Fereastra pentru selecția tipului de aplicație, proiect sau simplu instrument virtual.

Vom selecta „Blank VI”. Se vor deschide trei ferestre ca în figura 4.

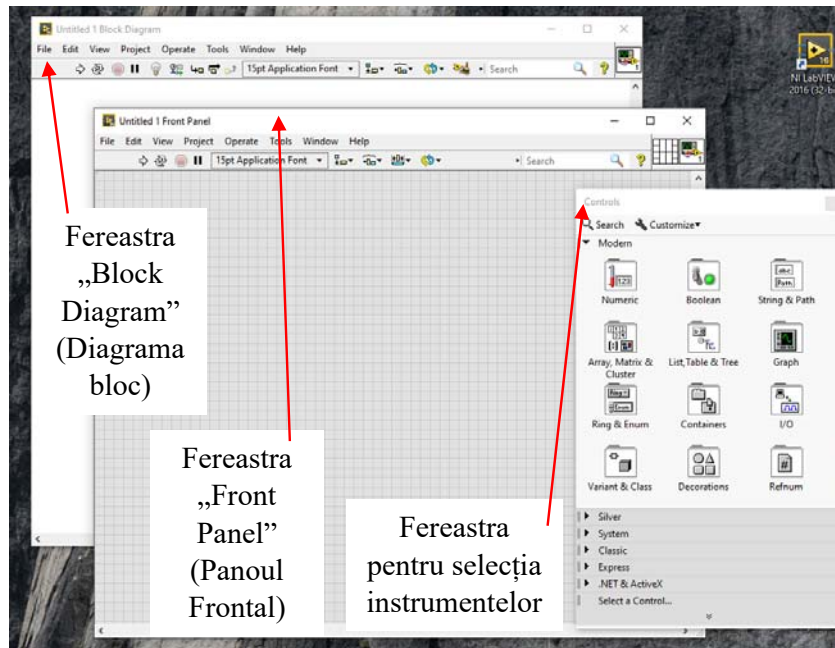


Fig. 4. Prezentarea ferestrelor unui instrument virtual.

Cele două ferestre se numesc „Front Panel” și „Block Diagram” și o fereastră din care se selectează instrumentele predefinite necesare utilizatorilor.

3. Simularea funcționării sistemului de semnalizare.

Vom începe prin simularea pornirii motorului cu ajutorul cheii de contact. Pentru aceasta se va insera în fereastra “Block Diagram” o structură “CASE” ca în figura 5. Această structură conține una sau mai multe subdiagramme. În funcție de valoarea selectorului la un moment dat se va executa doar una din subdiagramme. În figura 6 se prezintă o structură “CASE” în care sunt două subdiagramme ce se vor executa în funcție de valoarea selectorului. Acest selector poate fi un controler de tip “întreg” sau poate fi de tip “Boolean”. În figura 6 sunt prezentate două cazuri, când butonul e apăsat și se execută diagrama “True” (figura 6. a) se face adunarea celor două numere și al doilea caz când butonul nu este apăsat și se execută digrama “False” (figura 6.b) caz în care se execută scăderea celor două numere.

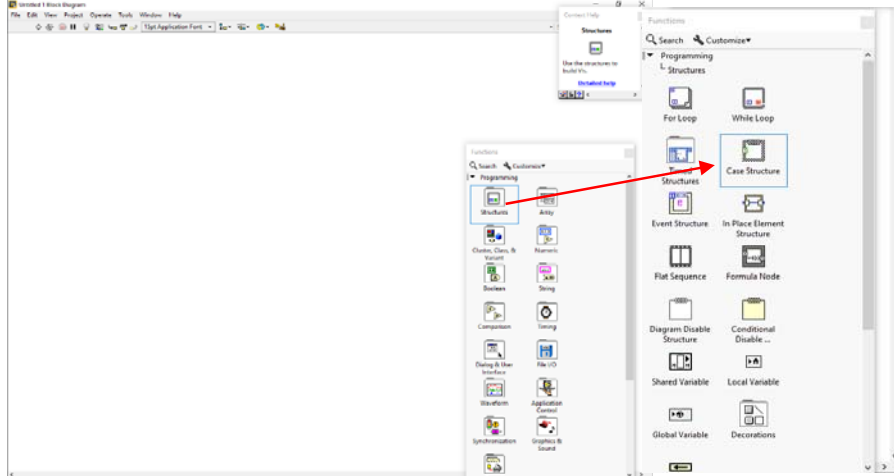


Fig. 5. Prezentarea modului de inserare a unei structuri „Case”.

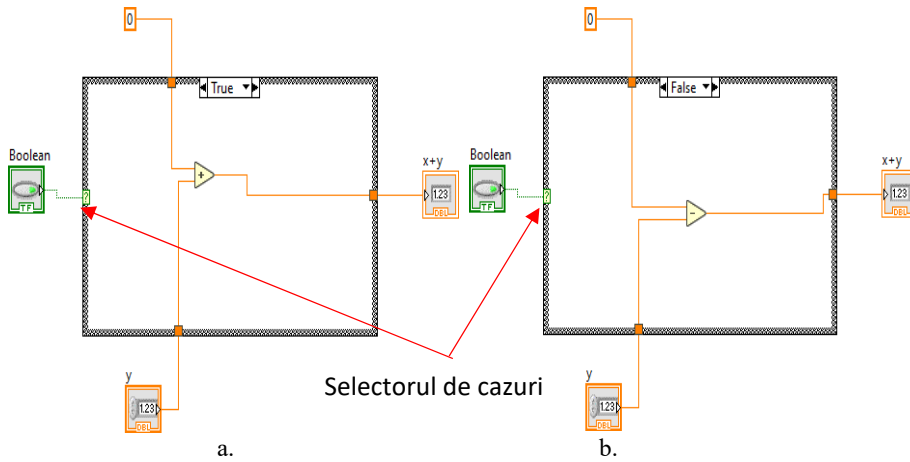


Fig. 6. Prezentarea modului de selecție a subdiagramei dorite.

Vom folosi un selector de tip ”Boolean” care va reprezenta cheia de contact. Când butonul este apăsat va avea valoarea “Adevărat” (A) și se va considera că s-a pornit motorul.

Dacă butonul de selecție nu este apăsat valoarea lui va fi “False” (F) și atunci motorul va fi oprit.

a. Panoul Frontal al instrumentului virtual

În figura 7 se prezintă panoul frontal pentru instrumentul virtual ce simulează funcționarea sistemului de semnalizare a schimbării direcției de mers a unui autovehicul.

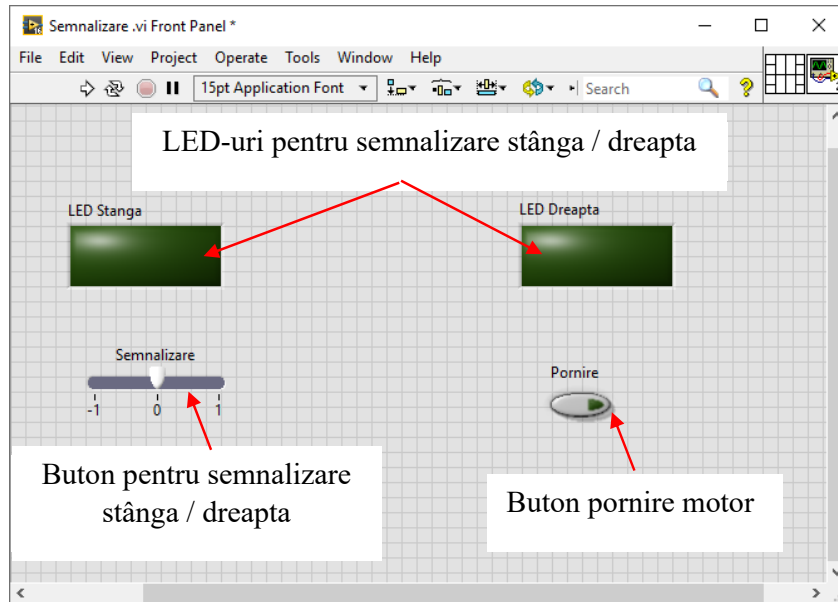


Fig. 7. Panoul frontal al instrumentului virtual ce simulează funcționarea sistemului de semnalizare a schimbării direcției de mers a unui autovehicul.

În figura 7 apar elementele:

- butonul de pornire a motorului (de tip Boolean)
- butonul de semnalizare (de tip Întreg cu cele trei valori (-1, 0 ,1), unde -1 semnifică semnalizare la stânga, 1 semnalizare la dreapta, iar 0 reprezintă poziția de repaus.
- cele două becuri pentru semnalizare stânga / dreapta.

Pentru a adăuga aceste elemente în panoul frontal dăm clic dreapta pe fereastra panoului frontal, iar din fereastra nou apărută vom selecta:

- pentru butonul de pornire vom da clic dreapta, vom selecta Boolean, din care vom selecta „Push Button”, ca în figura 8.

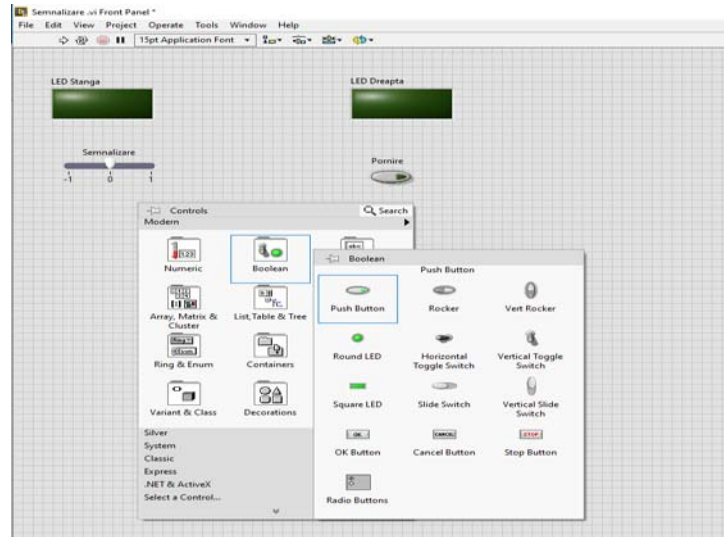


Fig. 8. Prezentarea modului de inserare a butonului de pornire „Push Button”.

- pentru butonul de semnalizare se va merge în “Numeric” și se va selecta „Vertical Pointer Slide”, ca în figura 9. După ce a fost plasat elementul în panoul frontal se vor modifica valorile implicite în valorile următoare: -1, 0, 1.

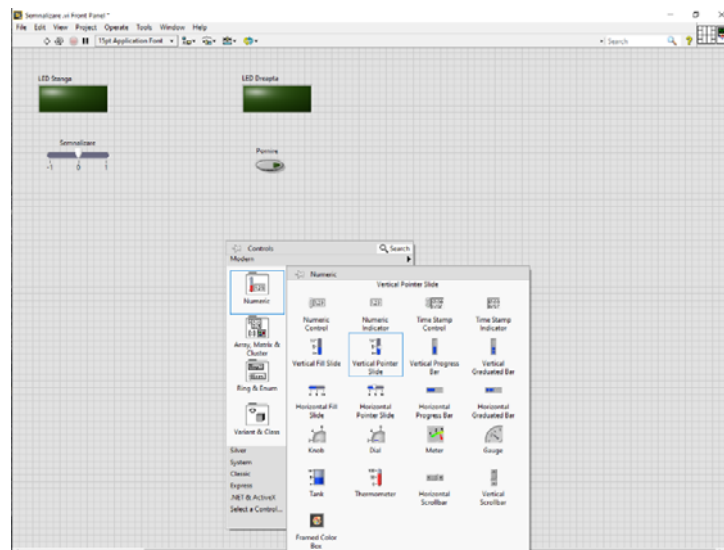


Fig. 9. Prezentarea modului de inserare a controlerului pentru semnalizarea schimbării direcției „Vertical Pointer Slide”

- pentru cele două LED-uri ce vor simula becurile se va da clic dreapta pe panoul frontal și din „Boolean” vom selecta „Square LED”, ca în figura 10.

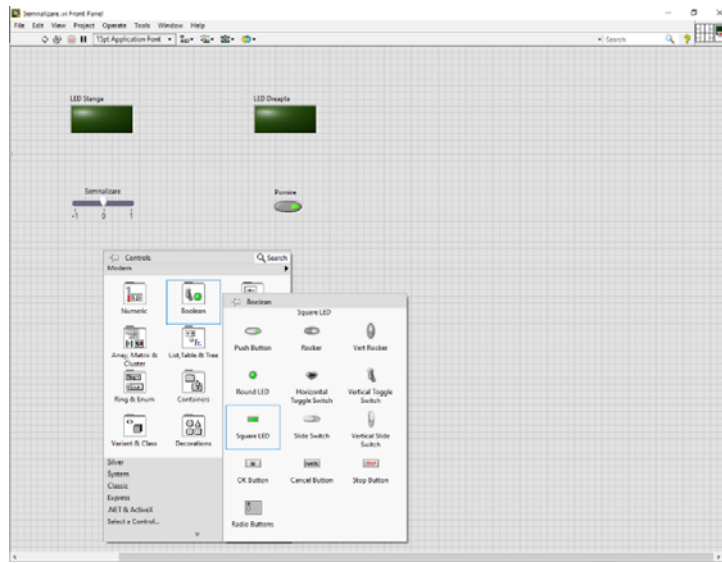


Fig. 10. Prezentarea modului de inserare a unui indicator de tip LED.

b. Diagrama bloc a instrumentului virtual

Fereastra pentru diagrama bloc este prezentată în figura 11. Se poate vedea bucla “Case” principală care simulează pornirea / oprirea motorului în momentul în care se apasă butonul de selecție a diagramei ce va fi executată denumit “Pornire”.

Pentru a porni motorul trebuie să apăsăm butonul “Pornire” și atunci se va executa diagrama “ True” din figura 11.

În cadrul acestei structuri sunt executate alte două structuri „Case” corespunzătoare celor două direcții stânga sau dreapta.

Aceste două structuri sunt identice și sunt dezvoltate astfel:

Se inserează o buclă “While”. Pe această buclă vom da clic dreapta și vom adăuga “Add Shift Register”. Acești regiștrii au două componente, cel din dreapta va memora valoarea la iterația curentă iar la registru din stânga va fi disponibilă valoarea de la iterația anterioară. Această buclă se va executa la infinit atât timp cât nu intervine o întrerupere care să oprească executarea buclei.

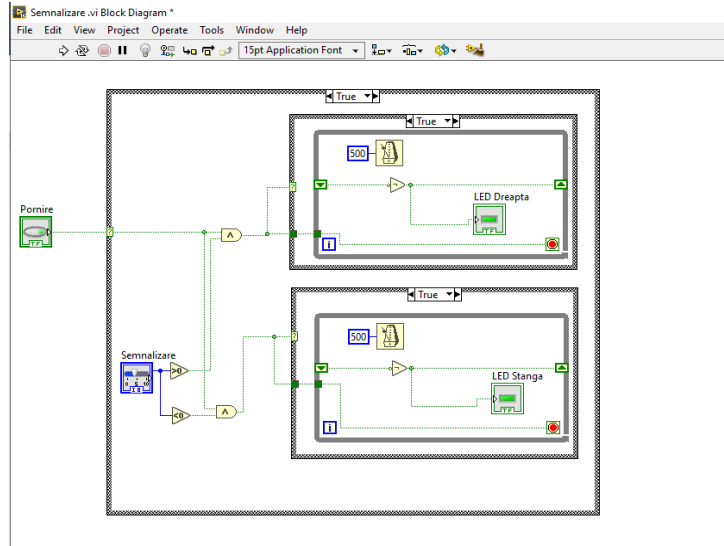


Fig. 11. Diagrama bloc a instrumentului virtual ce simulează circuitul de semnalizare a schimbării direcției de mers.

În cadrul buclei „While” se va adăuga o întârziere de 500 milisecunde și reprezintă timpul după care se va relua executarea buclei. Instrumentul pentru temporizare se găsește în meniul “Timing”, de unde selectăm instrumental “Wait Until Next ms Multiple” ca în figura 12.

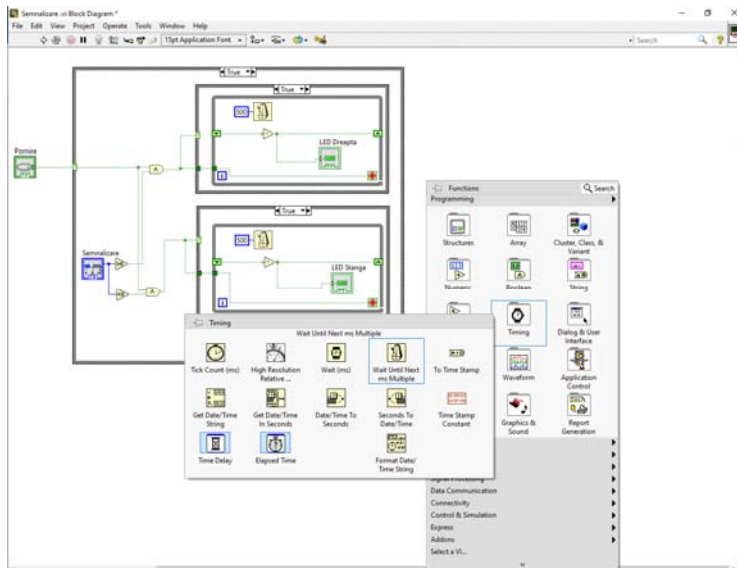


Fig. 12. Prezentarea modului de inserare a instrumentului de temporizare.

Pentru adăugarea timpului (constantei din fața instrumentului) se va da clic dreapta pe partea din stânga a instrumentului și din meniul pop-up ce apare se va selecta “Create Constant” caz în care valoarea va fi o constantă egală ca valoare cu cea scrisă, sau putem selecta ”Create Control” caz în care putem modifica valoarea de întârziere în timpul rulării simulării.

Tot în cadrul acestei bucle vom insera un instrument de tip “Not” din meniul “Boolean” ca în figura 13.

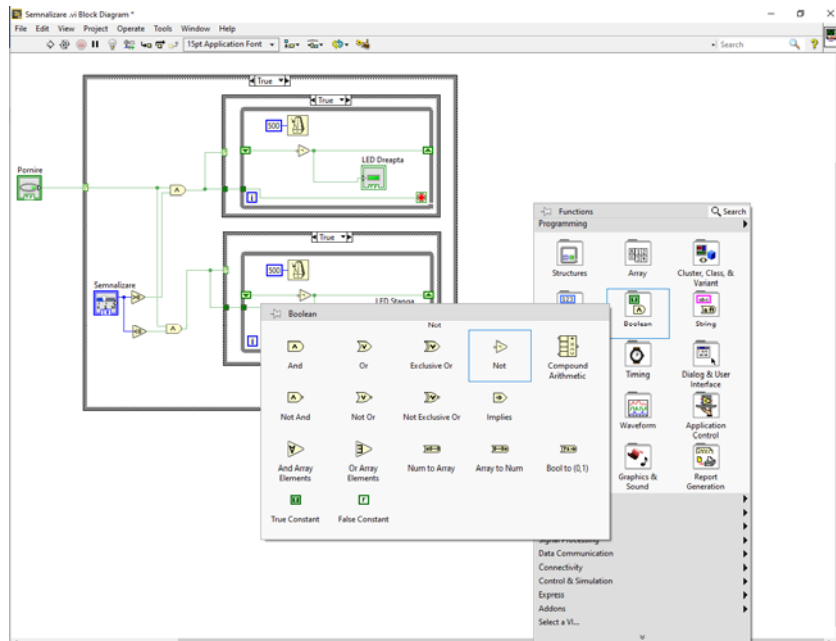


Fig. 13. Prezentarea modului de inserare a instrumentului de inversare a stării „Not”.

Vom conecta componenta din stânga a “Shift Register” la intrarea instrumentului “Not” iar ieșirea lui o vom conecta la componenta din dreapta. Astfel la pornirea simulării valoarea inițială va fi 0 (la componenta din stânga), după prima iterație la componenta din dreapta prin negarea instrumentului “Not” vom avea 1. După care la următoarea iterație a buclei valoarea 1 din dreapta va fi transferată la componenta din stânga iar prin instrumentul “Not” se va modifica în 0 și va fi transferat componentei din dreapta. Procesul va fi reluat la infinit cu o întârziere stabilă de utilizator. Astfel conectând LED-ul la ieșirea instrumentului “Not” având alternanța între 0 și 1 acesta se va aprinde și se va stinge cu o întârzierea stabilă.

Pentru selecția unei din cele două structuri „Case”, avem selectorul din figura 9. Acesta are trei valori și anume: -1, 0 și 1. De aceea vom folosi

două comparatoare, unul pentru valori mai mici decât zero "Less Than 0? Function" și unul pentru valori mai mari decât zero "Greater Than 0? Function". Acestea se regăsesc în „Comparision” ca în figura 14.

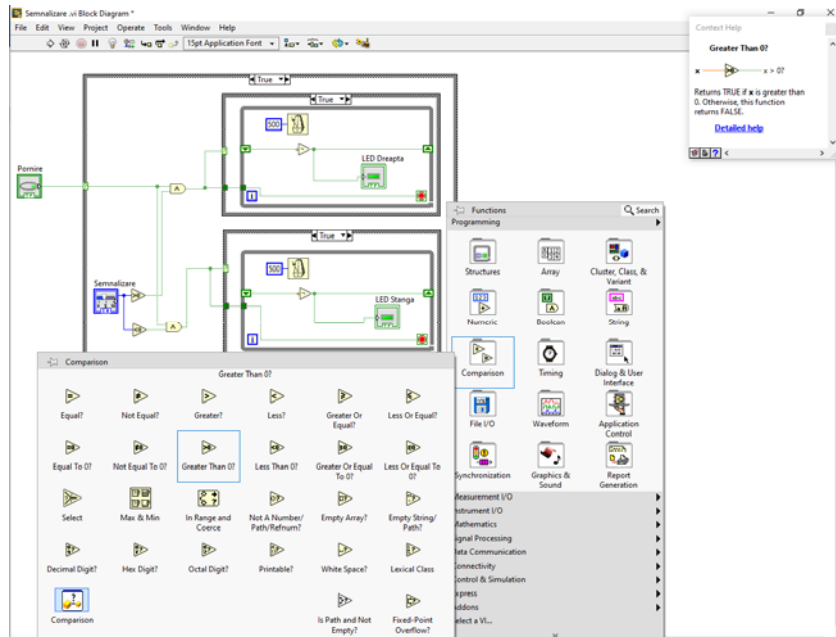


Fig. 14. Prezentarea modului de inserare a unui comparator de tipul „mai mic” sau „mai mare”

Apoi ieșirile acestor comparatoare sunt conectate la intrările unor porți „AND”, iar ieșirea acestei porți va comanda structurile „Case”. La funcționare normală aceste structuri vor fi în cazul “True”. Când nu sunt selectate aceste structuri vor fi în cazul “False”, ca în figura 15.

În cazul “False” LED-ul corespunzător trebuie să fie stins. Pentru aceasta în cadrul structurii “True” trebuie să face câte o variabilă locală pentru fiecare din cele două direcții, ca în figura 16.

Aceste variabile locale se vor copia în dublu exemplar, un exemplar va fi inserat în structura stânga sau dreapta și se vor lega la o constantă de tip „Boolean” ce va avea valoarea „False”, ceea ce înseamnă că LED-ul va fi stins.

Totodată o altă copie a acestor variabile locale vor fi copiate în structura principală „Case” ca în figura 17.

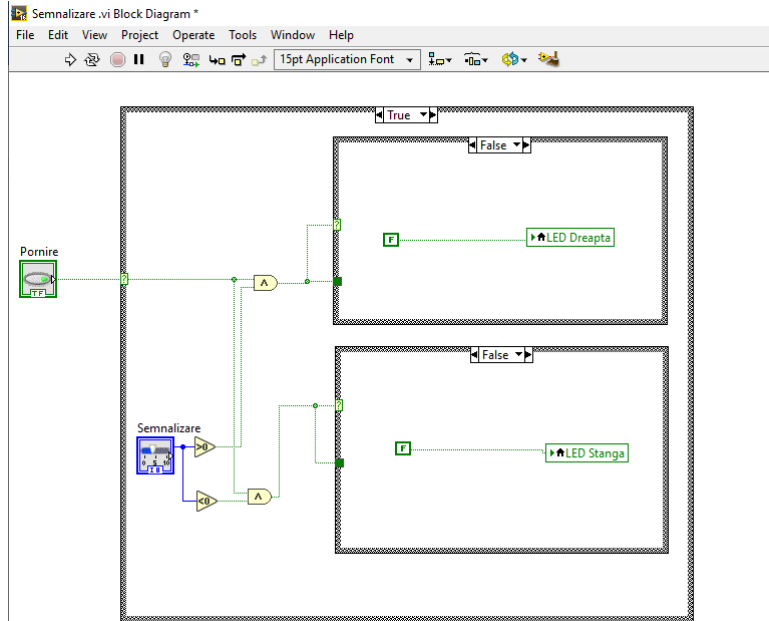


Fig. 15. Diagrama bloc a instrumentului virtual pentru cazul în care se execută subdiagrama „False”

În figura 17 se prezintă cazul în care simularea este pornită însă nu este selectat niciuna din cele două direcții de semnalizare, deci selectorul este pe poziția 0, iar structura principală case (care simulează pornirea motorului) este în diagrama „False”

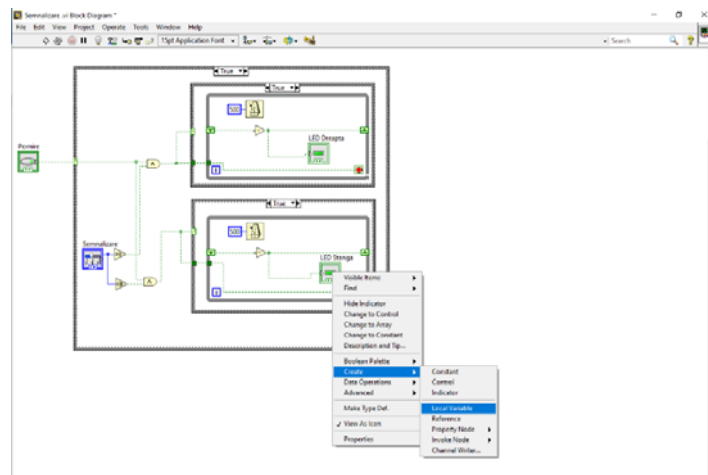


Fig. 16. Prezentarea modului de realizare a variabilei locale „Local Variable” pentru LED-uri

De asemenea în acest caz se vor conecta cele două variabile locale la o constantă ce va avea valoarea „False”

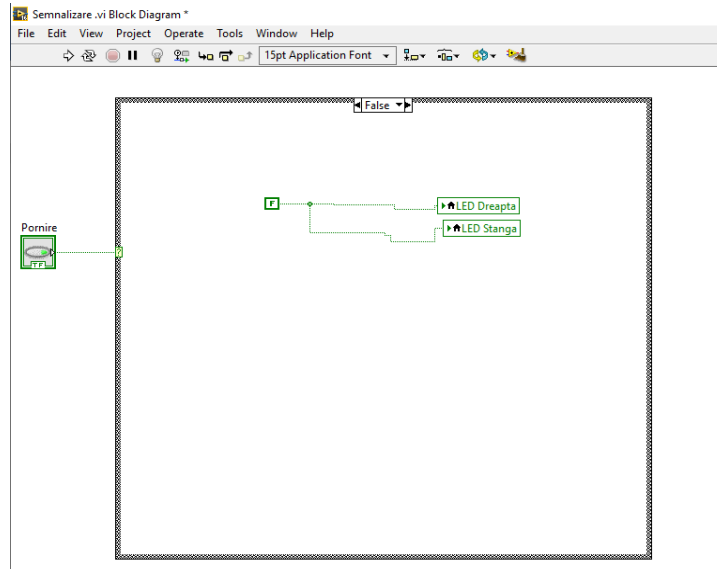


Fig. 17. Digrama bloc pentru subdiagrama „False” caz în care LED-urile vor fi stinse.

După ce s-a realizat toată schema se va porni simularea. Sunt două moduri de rulare a simulării.

1. rularea o singură dată a buclei
2. rularea încontinuu a buclei.

Pentru cele două moduri se prezintă în figura 18 butoanele aferente.

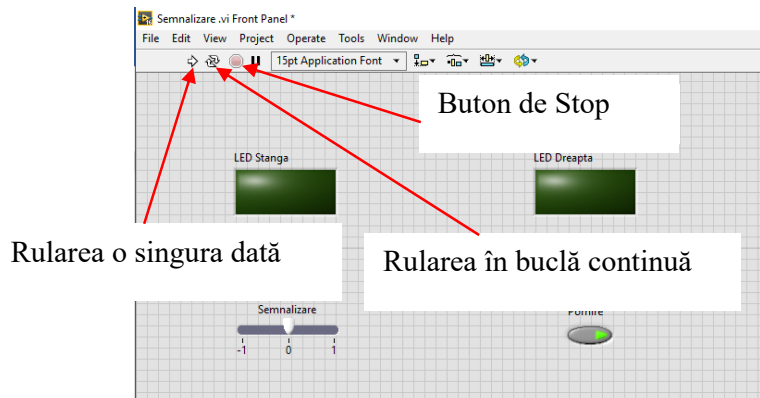


Fig. 18. Panoul frontal al instrumentului, în care se prezintă butoanele.

4. Lucrarea practică

1. Se va studia și se va realiza programul în mediul de programare grafic LabVIEW.
2. Se vor simula funcționarea sistemului de semnalizare prin aprinderea unuia din cele două LED-uri, în funcție de poziția selectorului stânga (-1) sau dreapta (1) când butonul ce simulează cheia de contact nu este apăsat.
3. Se vor simula funcționarea sistemului de semnalizare prin aprinderea unuia din cele două LED-uri, în funcție de poziția selectorului stânga (-1) sau dreapta (1) când butonul ce simulează cheia de contact este apăsat.

Bibliografie

- [1] National Instruments -tutorial
- [2] LabVIEW™ Help