

DRIVER LABVIEW PENTRU OSCILOSCOP

Valentin Golea*, Marin Sărăcin**, Cristina Gabriela Sărăcin**, Mihail Dona**

* ISPCF SA Bucharest e-mail: valgo@go.ro,

** The University of Politehnics From Bucharest e-mail: sarm@electro.masuri.pub.ro

Abstract:

This paper describes the driver for the oscilloscope built using: the SCPI commands, the simplest blocks of LabVIEW and the RS232 interface which is used to communicate with the instrument. The purpose of this driver is the use of the serial bus RS-232. In order to use this interface, we started with the reset of the oscilloscope's memory so that no unuse data should appear on the display. Then the remote control chooses the destination of commands given from the computer to the instrument. The oscilloscope will receive the command of autoscaling, so that the signals can be visualised at an appropriate scale. The user can choose the timebase, the channel and the type of measurement. The trigger will be done in the conditions preset by the user.

Un driver LabVIEW reprezintă un set de VI-uri care controlează un instrument programabil. Fiecare VI corespunde unei operații a instrumentului, cum ar fi configurarea, declanșarea și citirea măsurărilor de la instrument. Driverul instrumentelor ajută utilizatorii să inițieze operarea cu instrumentul de la calculator reducând, timpul și costurile întrucât utilizatorii nu vor învăța protocolul de programare pentru fiecare instrument.

Rețeaua driverelor de instrumente ale LabVIEW-ului conține elemente pentru o gamă largă de instrumente programabile, incluzând instrumente cu interfețe GPIB, USB, TCP/IP, VXI, RS-232 și PXI. Majoritatea instrumentelor programabile au un număr mare de funcții și moduri de utilizare. Datorită acestei complexități, este necesar a se asigura un model de proiectare consistent care să ajute atât pe cel care îl realizează, dar și pe cel care îl utilizează.

Driverul de instrument LabVIEW tip plug & play conține atât o structură internă, cât și una externă. Structura externă indică interfețele driverului cu utilizatorul și alte componente software ale sistemului. Structura internă indică organizarea internă a modulului software pentru driver. Structura externă a driverului constă în VI-uri API, pe care utilizatorul le apelează pentru aplicații de nivel înalt. Aceasta indică faptul că instrumentul prezintă atât o interfață interactivă, cât și una programabilă. Interfața programabilă a aplicației (API) este un set de VI-uri ale driverului pentru utilizator, ce sunt întrebuințate în sistemele ce se termină cu utilizatorul. De exemplu, un sistem de testare al unui producător poate face ca driverul să comunice cu un multimetru sau osciloscop.

Arhitectura software pentru instrumente virtuale (VISA) prezintă o interfață I/O, ce este compusă dintr-un set de funcții LabVIEW pe care driverul le utilizează pentru a comunica cu partea hardware a instrumentului. VISA a stabilit o interfață standard pentru instrumentație care controlează magistralele GPIB, USB, serială și alte magistrale de instrument.

Structura internă a driverului de instrument definește organizarea VI-urilor pentru acesta. Funcțiile VISA realizează supravegherea dispozitivelor, intrările/ieșirile standardizate ale instrumentului și rezolvarea erorilor.

Există două tipuri de VI-uri API într-un driver de instrument: cele comune tuturor instrumentelor și cele care se referă la posibilitățile specifice ale instrumentului. VI-urile driverului de bază vor realiza operațiile comune: inițializare, închidere, resetare, auto-testare, interogare pentru revizie și interogare a erorii. Utilizatorul este cel care va defini operațiile specifice fiecărui instrument. Unele drivere pot avea definite categorii suplimentare, cum ar fi „route” și „scan” pentru instrumente de comutație.

Există trei tipuri obișnuite de drivere care realizează controlul instrumentelor în cadrul LabVIEW-ului. Diferența între ele constă în implementarea lor și nu în utilizarea lor. Cele trei tipuri de drivere pentru instrumente sunt: driverele LabVIEW Plug&Play, driverele IVI, drivere de instrument dedicate unor aplicații.

Driverele Plug&Play reprezintă un set de instrumente virtuale (VI-uri) care controlează și comunică cu un instrument programabil. Fiecare VI corespunde unei operații de programare, cum ar fi configurare, citire, scriere sau declanșare a unui instrument. Aceste drivere includ și moduri de eliminare a erorilor, panouri frontale, diagrame bloc și asistență interactivă.

Driverele IVI sunt mai sofisticate decât cele plug&play, care permit simulări și interschimbabilitate a instrumentelor. Se poate scrie un VI care poate lucra cu mai multe mărci de osciloscop, chiar dacă acestea utilizează conexiuni diferite pentru magistrală.

Driverele realizate de utilizatori sunt incluse pentru a fi utilizate în cazul unor aplicații speciale și nu pot să asigure îndeplinirea tuturor funcțiilor instrumentului specificat.

Astfel pot fi create librării care să conțină drivere pentru o gamă largă de instrumente programabile (multimetru, osciloscop, generator) interfațate serial sau paralel.

Pentru realizarea driver-ului în limbajul LabVIEW pentru osciloscop s-au urmărit etapele prezentate în continuare.

Prima etapă o reprezintă inițializarea interfeței seriale prin intermediul LabVIEW-ului, pentru care am apelat la blocul *VISA Configure Serial Port* din biblioteca *Serial*.

După cum se poate observa din figura 1, pentru interfața serială trebuie să fie setate:

VISA resource name – numele portului calculatorului la care este legată magistrala serială;

baud rate – rata de transfer a informațiilor pentru interfața serială;

data bits – numărul biților de date pentru transfer;

parity – paritatea interfeței seriale;

stop bits – biții de stop;

flow control – controlul pentru transferul datelor.

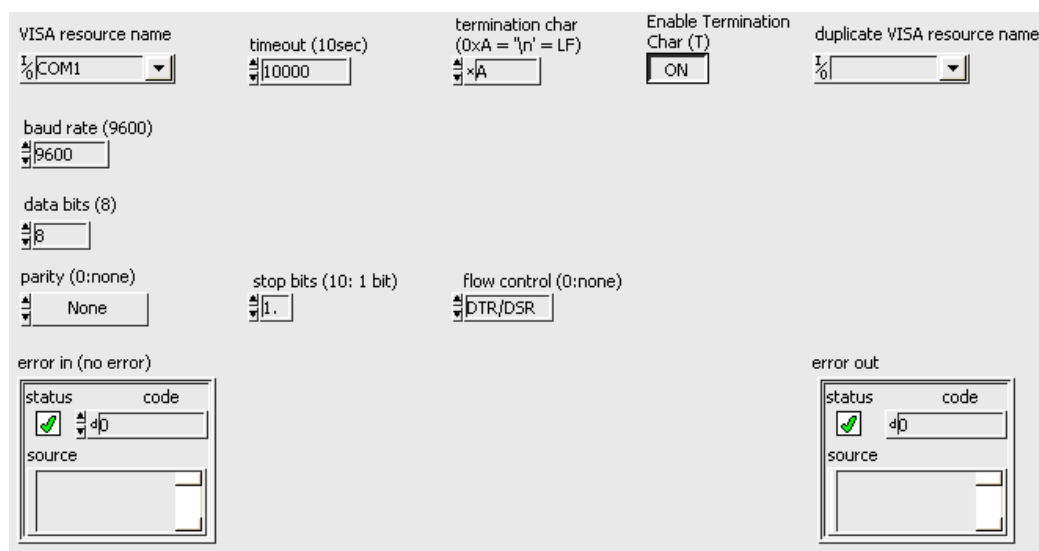


Fig. 1 Blocul Configure Serial Port

O dată setați acești parametri trebuie avut în vedere ca valorile acestora să fie identice cu cele ce se găsesc și la nivelul instrumentului fizic, evitându-se astfel apariția unor erori sau lipsa de comunicație între calculator și instrument.

Etapa a doua a constat în implementarea funcției *:AUToscale*, utilizată pentru a permite osciloscopului să se autoscaleze (ce conduce la setarea: canalului vertical, a bazei de timp și a nivelului de declanșare).

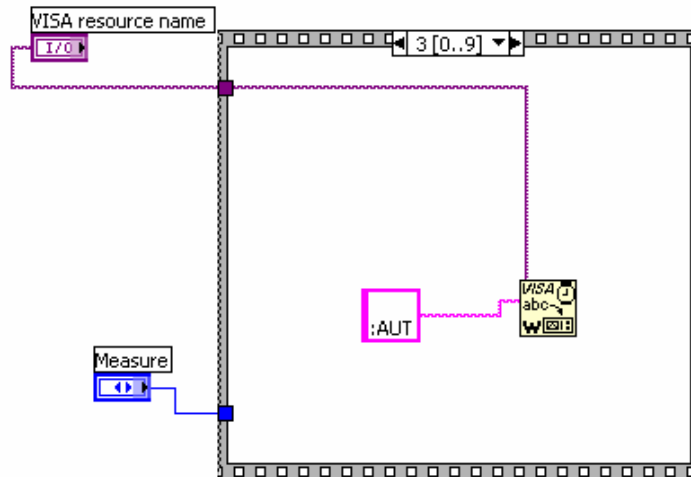


Fig. 2 Diagrama bloc pentru funcția de autoscalare a osciloscopului

În etapa a treia s-a realizat blocul pentru fiecare grup de funcții recunoscute de instrument, în continuare fiind dat ca exemplu blocul pentru măsurarea diverselor mărimi cu ajutorul osciloscopului. Pentru acest lucru s-a apelat la alegerea domeniului măsurărilor, adică tensiune sau timp. Cu ajutorul unui switch se alege domeniul în care se face măsurarea (tensiune sau timp).

Pentru domeniul tensiunilor, s-au utilizat următoarele comenzi, după modelul SCPI: *:MEASure:VPP?*, pentru măsurarea valorii de tensiune vârf-vârf; *:MEASure:VRMS?*, pentru măsurarea tensiunii medii; *:MEASure:VMAX?*, pentru măsurarea valorii maxime de tensiune; *:MEASure:VMIN?*, pentru măsurarea valorii minime de tensiune.

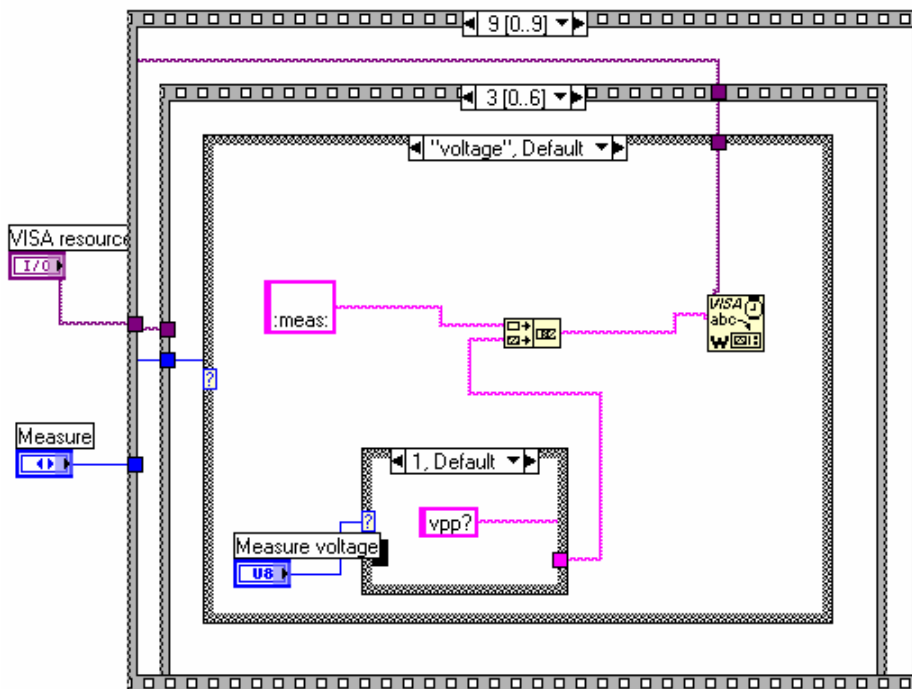


Fig. 3 Diagrama bloc pentru măsurarea tensiunii vârf-vârf a semnalului

Pentru domeniul timp: *:MEASure:FREQuency?*, pentru măsurarea frecvenței; *:MEASure:PERiod?*, pentru măsurarea perioadei semnalului; *:MEASure:DUTYcycle?*, pentru măsurarea factorului de umplere; *:MEASure:PWIDth?*, pentru măsurarea duratei pulsului pozitiv; *:MEASure:NWIDth?*, pentru măsurarea duratei pulsului negativ; *:MEASure:FALLtime?*, pentru măsurarea timpului de cădere a semnalului; *:MEASure:RISEtime?*, pentru măsurarea timpului de creștere a semnalului.

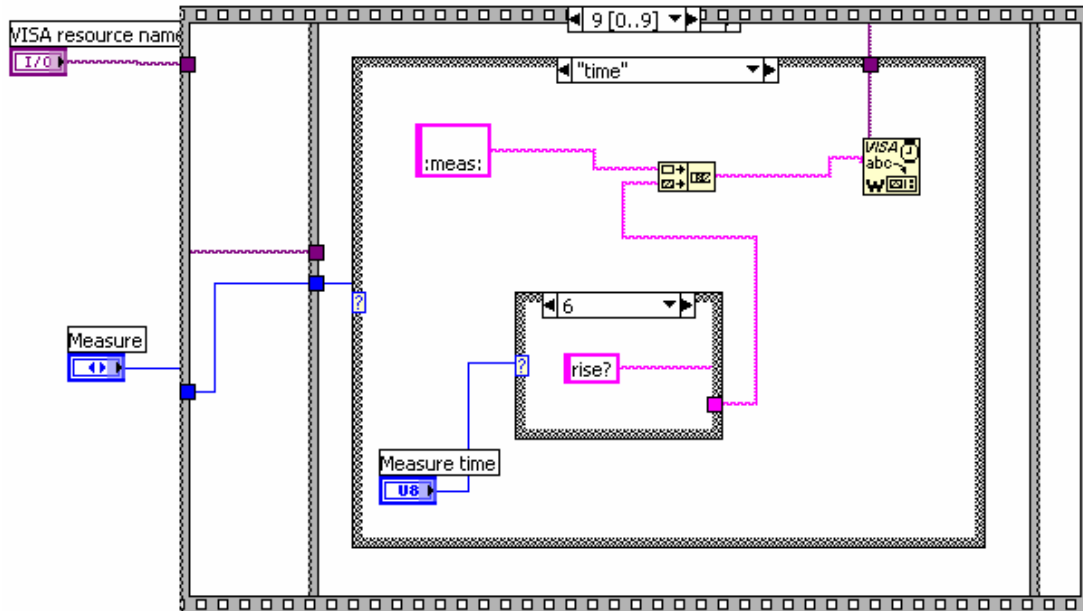


Fig. 4 Diagrama bloc pentru măsurarea timpului de creștere a semnalului

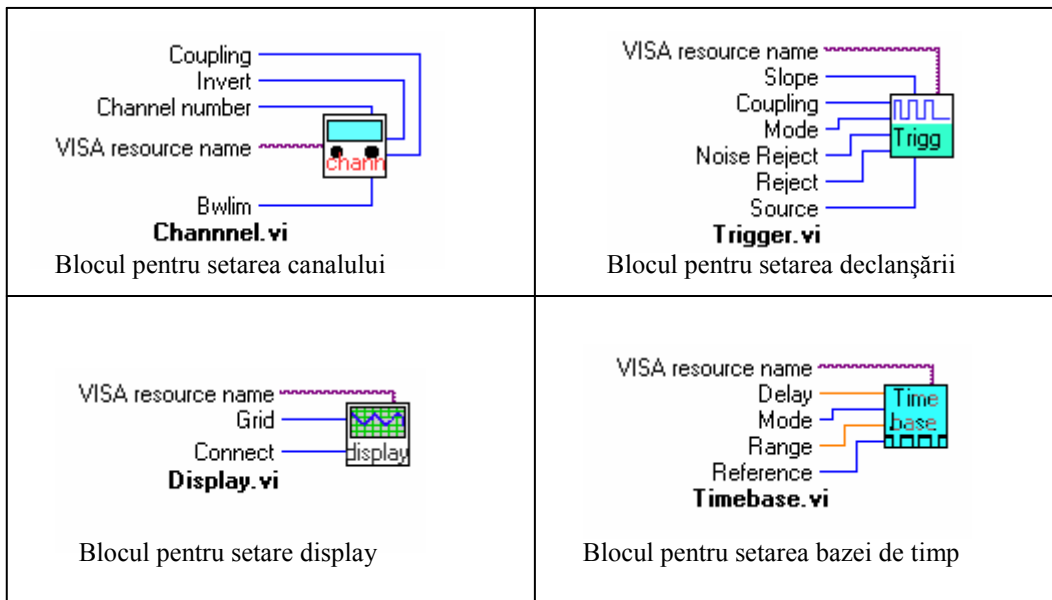


Fig. 5 Vi-urile instrumentului de achiziție

Întrucât un driver de instrument se reprezintă prin intermediul icon-urilor ce sunt destinate fiecărui bloc din componența instrumentului, s-au realizat VI-uri separate pentru funcțiile osciloscopului, care grupate constituie un instrument achiziție de semnale. Aceste icon-uri sunt prezentate în figura 5.

Panoul frontal al osciloscopului virtual realizat în LabVIEW este prezentat în figura 6. Acest driver pentru osciloscop prezintă ca VI-uri:

- blocul pentru interfațarea serială;
- pentru fiecare canal se pot alege: numărul, modul de cuplare, limita benzii de frecvență;
- pentru baza de timp se poate alege: întârzierea la declanșare, modul de declanșare, domeniul bazei de timp;
- pentru declanșare se poate alege: modul de cuplare, rejecția zgomotelor, canalul de declanșare, panta semnalului de declanșare;
- pentru măsurări se poate alege parametrul ce va fi măsurat, adică tensiune sau timp;
- pentru tensiune pot fi măsurate printre altele valoarea vârf-vârf, valoarea efectivă și cea medie;
- pentru timp se va măsura perioada, timpul de creștere și cădere.

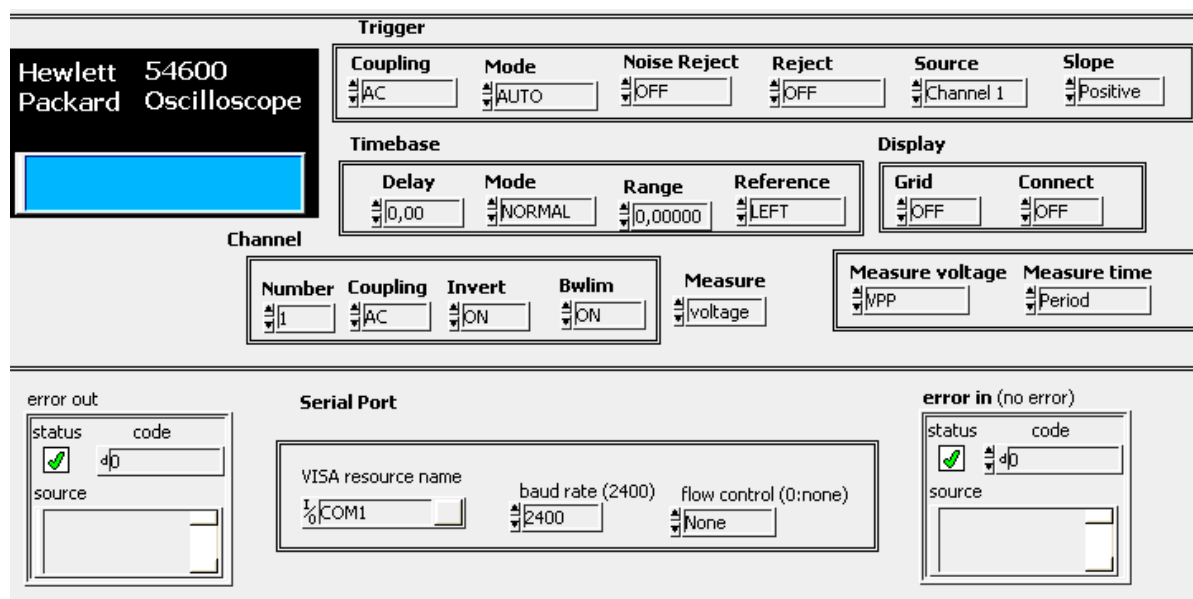


Fig. 6 Panoul frontal al osciloscopului

CONCLUZII:

Lucrarea de față prezintă realizarea unui driver pentru osciloscop utilizând limbajul SCPI cu ajutorul celor mai simple blocuri ale ale LabVIEW-ului, comunicația cu instrumentul realizându-se prin intermediul interfeței seriale RS232.

De aceea el prezintă următoarele avantaje:

- Instrumentul virtual ce reprezintă interfața hardware și software, transformă orice calculator standard într-un instrument mai ieftin decât cel tradițional, oferind utilizatorului posibilitatea de a-și implementa funcțiile de care are nevoie, când are nevoie;
- Monitorul calculatorului personal poate fi mai bun decât tubul catodic al unui osciloscop, microprocesorul calculatorului PC este mai rapid decât controller-ul instrumentului tradițional;
- Prețul unui astfel de instrument este mult mai mic decât al unui instrument tradițional;
- Un alt avantaj al instrumentului virtual îl reprezintă faptul că printr-o conectare la internet se pot primi în timp real rezultatele unui experiment ce are loc în alt continent, sau se poate controla instrumentul virtual oriunde în lume.

ATEE 2004

- LabVIEW integrează toate mijloacele necesare pentru programarea inginerescă științifică într-o singură metodologie, dând posibilitatea utilizatorilor să vizualizeze și să construiască, complet automat, sisteme de măsurare.
- Prin utilizarea acestui limbaj, rapiditatea programării crește foarte mult datorită introducerii unei interfețe grafice intuitive.

Bibliografie

Sărăcin, M., Sărăcin, C.G., Golea, V.V. *Măsurări electronice și instrumentație virtuală*, Editura Economică, București, 2002.

****Tektronix, Test and Measurement, 1991-2002.

****Measurement Journal of the International Measurement Confederation, 1993-2002.

****Analog Devices, Technical Library

****HP54600A, User Guide

****National Instruments, Technical Library

****Oscilloscopes HP 54600A et HP 54601A, Guide d'utilisation et de maintenance, 1991.

****Review of Scientific Instruments, 1992-2002.