

Relazione del periodo di stage effettuato presso i Laboratori Nazionali di Frascati dell'Istituto Nazionale di Fisica Nucleare

Periodo dello stage :

11 Giugno/6 Luglio 2001

Gruppo: Da ne-Luce

Tutor: dott. Augusto Marcelli, dott.sa Elisabetta Pace

Tema: Interfacciamento di un computer per il controllo remoto di apparati sperimentali

Stagisti: Gianluca Saraga, Roberto Vinci
classe 4^e dell'ITIS Enrico Fermi di Frascati

Relazione realizzata da Saraga Gianluca

Argomenti trattati durante il periodo di stage:

- *Studio di interfaccia seriale RS232*

- *Studio del programma LabVIEW (versione 6.1) della National Instruments e del linguaggio di programmazione G*

- *Assemblaggio di connettori DB9 e DB25*

- *Creazione di un programma in LabVIEW in grado di comunicare con un interfaccia serie RS232*

- *Lettura remota dei dati forniti attraverso porta seriale RS232 di uno Stepping Motor Controller PMCD-05 della Toyama e di un Angle Encoder ND261 delle Heidenhain*

Lo scopo del nostro stage era quello di leggere sul PC i dati fornitici da un controllore di motore passo-passo PMCD-05 della Toyama e di un goniometro ND262 della Heidenhain. Per controllare i due dispositivi in modo remoto abbiamo usato il programma LabVIEW della National Instruments e abbiamo interconnesso il PC con la porta seriale RS232 di cui i due dispositivi sono forniti.

Studio di interfaccia seriale RS232.

Spiegazione del funzionamento di un interfaccia seriale.

I microprocessori operano con caratteri di 8,12 o 16 bit trasmessi in parallelo, ovvero simultaneamente. Questo vuol dire che non possono operare direttamente con i caratteri trasmessi dalle periferiche, poichè queste trasmettono dati in serie.

Per questo motivo c'è bisogno di un sistema di interfaccia serie, capace di convertire una sequenza di bit trasmessi in serie in parallelo, e viceversa, in modo sincrono o asincrono. Questo modo di comunicazione si serve di due dispositivi, uno di elaborazione dati detto DTE (Data Terminal Equipment) e un apparecchio di ritrasmissione dei dati detto DCE (Data communication Equipment).

La comunicazione può essere asincrona e sincrona. Per comunicazione asincrona e sincrona si intende il modo in cui i bit vengono trasmessi e ricevuti.

Nel modo di comunicazione asincrono i bit vengono trasmessi carattere per carattere. Un carattere, oltre a contenere i bit del dato e della parità, contiene anche un bit di partenza (start bit) e uno o più bit di arresto (stop bit).

Nel modo di comunicazione sincrono i dati vengono trasmessi come un flusso di bit, in cui non è possibile identificare l'inizio e la fine di un carattere. Per questo è necessario sincronizzare i caratteri all'inizio del blocco dati.

Nella comunicazione seriale sono importanti alcuni parametri come il baud rate, il bit parity, lo stop bit.

Lo studio dell'interfaccia seriale è servito per capire il funzionamento di un dispositivo RS232 e controllarlo con un apposito programma.

Studio del programma LabVIEW e del linguaggio G

Lo studio del programma LabVIEW è stata la parte principale del nostro lavoro al laboratorio.

LabVIEW è un sistema di sviluppo per applicazioni di acquisizioni e analisi dati per PC. Il linguaggio di programmazione usato da LabVIEW è chiamato G. Con questo linguaggio di programmazione grafica è possibile scrivere dei programmi disegnando semplicemente lo schema a blocchi.

Con questo programma l'acquisizione dei dati e il controllo della strumentazione possono essere effettuati tramite diversi tipi di interfacce, come IDE 488 (GPIB), RS232/422 oltre che con schede di acquisizione dati plug-in.

Il controllo della strumentazione viene facilitato da librerie di driver per gli strumenti più diffusi già fornite con LabVIEW stesso.

LabVIEW permette di presentare i dati in forma grafica, e modificarli per adattarli alle esigenze dell'utente.

I programmi che vengono realizzati con LabVIEW vengono chiamati strumenti virtuali (in inglese virtual instruments, VI) e sono realizzati da tre parti: il pannello frontale, il diagramma a blocchi e l'icona/connettore.

Il pannello frontale è lo strumento che permette di introdurre i valori in ingresso e vedere i risultati generati dal VI. Per analogia con il pannello frontale di uno strumento reale, gli ingressi vengono chiamati controlli e le uscite indicatori. È possibile utilizzare vari tipi di controlli e indicatori, come manopole, interruttori, bottoni, istogrammi grafici, ecc. per rendere il pannello frontale più comprensibile.

Ad ogni pannello frontale è associato un diagramma a blocchi, che può essere realizzato con il linguaggio G e può essere considerato l'equivalente del codice sorgente nella programmazione tradizionale. G come C o Basic è un linguaggio generale con librerie estese di funzioni per ogni tipo di programmazione. A differenza degli altri linguaggi di programmazione, basati su un testo, G è un linguaggio grafico.

L'ultima parte di un VI è l'icona/connettore che permette di trasformare un VI in un oggetto che può essere utilizzato come sottoprogramma (sub VI).

Assemblaggio di connettori DB9 e DB25.

Per leggere i dati forniti dai due apparecchi da interfacciare con il computer abbiamo avuto bisogno di collegare le rispettive porte seriali RS232 sia al computer sia all'apparecchio.

Una porta seriale RS232 può avere vari tipi di connettori, a 9 10 o 25 pin. Nel nostro caso abbiamo usato dei connettori DB9, a 9 piedini, e DB25, a 25 piedini. Per fare la connessione ci siamo avvalsi dei cavi, già presenti in laboratorio, ai quali sono stati collegati i connettori. All'estremità da connettere ai due dispositivi abbiamo collegato un connettore DB25, mentre all'estremità da connettere al PC abbiamo collegato un connettore DB9. I collegamenti tra i due connettori sono stati effettuati secondo la seguente tabella:

Tabella collegamenti DB9-DB25

Pin DB25	Segnale	Pin DB9
2	RXD	3
3	TXD	2
7	RTS	4
8	CTS	5
6	DSR	6
4	DTR	20

Il collegamento è stato effettuato saldando i fili dei cavi ai rispettivi pin dei connettori, utilizzando una saldatura a stagno e con l'ausilio di un termorestringente per isolare la saldatura dei pin.

Segnali usati nella comunicazione seriale.

Nella comunicazione seriale vengono utilizzati 6 segnali: RXD, TXD, RTS, CTS, DSR, DTR.

RXD: (Receive data) dati ricevuti. Linea usata dal DTE per ricevere i dati inviati dal DCE.

TXD: (Transmitted Data) dati trasmessi. Linea usata dal DTE per la trasmissione dei dati al DCE.

RTS: (Request to send) richiesta di invio. Linea attivata dal DTE per impegnare il DCE all'invio dei dati

CTS: (Clear to send) disponibilità all'invio. Linea attivata dal DCE per notificare al DTE la propria disponibilità all'invio dei dati.

DTR: (Data terminal ready) terminale dei dati pronto. Linea attivata dal DTE quando vuol segnalare al DCE il proprio stato di servizio attivo.

DSR: (Data set ready) apparecchiatura di connessione pronta. Linea attivata dal DCE, in risposta al DTR, quando vuol segnalare al DTE che la connessione con l'apparato remoto è attiva.

Creazione di un programma in LabVIEW per comunicare con l'RS232.

Per realizzare uno strumento virtuale in grado di comunicare con l'interfaccia seriale RS232 abbiamo usato il programma LabVIEW. Era necessario creare un VI che comunicasse con l'oggetto da interfacciare tramite l'interfaccia seriale RS232 presente sullo strumento. Per fare ciò ci siamo avvalsi dell'uso di una scheda PCI a 4 porte RS232 della National Instruments, che è stata installata su un PC con sistema operativo Windows 2000. Abbiamo installato la porta inserendola in uno slot libero all'interno del PC, e essendo una scheda plug and play, appena il calcolatore è stato riaccessato il sistema operativo ha riconosciuto la scheda ed ha eseguito la configurazione e l'installazione dei driver. Una volta terminato questo processo abbiamo controllato che le impostazioni della scheda fossero corrette, e abbiamo eseguito un test delle porte seriali presenti sulla scheda, che sono risultate tutte correttamente funzionanti.

A due ingressi della scheda, precisamente porta COM7 e COM8, abbiamo collegato i cavi provenienti dai due dispositivi. I due dispositivi da interfacciare erano uno Stepping Motor Controller PMCD-o5 della Toyama e un Angle Encoder ND261 della Heidenhain.

Il primo è un controllore di un motore passo-passo, il secondo è un goniometro. Le due porte RS232 del PC sono state configurate in accordo con quelle degli apparecchi da interfacciare, tramite switch presenti sul retro degli strumenti. Con questa operazione sono stati adatti i parametri delle due porte (bit parity, baud rate...).

Il programma doveva attivare le funzioni caratteristiche dello strumento reale, tramite una serie di comandi illustrati nei manuali d'istruzione di gli apparecchi sono corredati. Dovevamo far sì che il VI riuscisse a riportare su un indicatore i dati che si trovavano sul display dello strumento reale. Il programma è costituito da due subVI, uno per inizializzare la porta seriale RS232, e l'altro a inviare e ricevere i dati dalla porta.

Lettura dei dati forniti dalle porte RS232 dei due dispositivi.

Dopo aver sviluppato il programma in LabVIEW e aver collegato i cavi sia al computer sia agli apparecchi, abbiamo provato a far girare il programma. Dopo numerosi tentativi il programma in risposta ci ha dato una stringa in risposta al comando che gli abbiamo dato. Il nostro comando è stato un comando di tipo S70R<CR> <LF>, ovvero abbiamo

chiesto alla CPU presente sullo Stepping Motor Controller di inviarci un dato sul suo stato, se fosse occupata o in attesa. Questa è stata la conferma che il nostro lavoro è stato svolto bene, senza troppi problemi. I due dispositivi mandano indietro al PC dei dati soltanto su richiesta ed in un formato preciso a seconda della richiesta fatta.

Nel caso dello Stepping Motor Controller un comando S201 <CR> <LF> corrisponde una richiesta di stato della CPU presente sul controller, e questo rispondeva con un a stringa di tipo RA <2 cifre esadecimali> <CR> <LF> dove ogni bit dei numeri esadecimali ha un significato.

Oppure il computer poteva non aspettarsi nessuna risposta, come ad esempio quando viene commutato lo stato del controller da remoto a locale. Questo lavoro è stato utile per utilizzare i due strumenti che erano già presenti in laboratorio, e che dovevano essere collegati.