

Sezione di Genova

INFN/BE-81/5
22 Gennaio 1981

M. Anghinolfi e F. Masulli: SOFTWARE DI BASE PER LA
GESTIONE DI UN CAMAC DA MINICOMPUTER.

Istituto Nazionale di Fisica Nucleare
Sezione di Genova

INFN/BE-81/5
22 Gennaio 1981

SOFTWARE DI BASE PER LA GESTIONE DI UN CAMAC DA MINICOMPUTER.

M. Anghinolfi
INFN, Sezione di Genova

e
F. Masulli
Istituto di Scienze Fisiche dell'Università di Genova.

1. - INTRODUZIONE.

Tra i diversi standard di interconnessione strumentazione-elaboratore il CAMAC, proposto dai gruppi NIM ed ESONE, riflette meglio le esigenze della strumentazione nell'ambito della fisica nucleare; questa strumentazione è infatti normalmente concentrata in un unico luogo e raggiunge una elevata velocità di trasferimento dati.

Nel nostro laboratorio si sta sviluppando un sistema di acquisizione dati costituito da un minicalcolatore PDP11/34 a 16 Kword di memoria collegato mediante un controller (ORTEC DC 011) a un crate CAMAC (NUCLEAR ENTERPRISE).

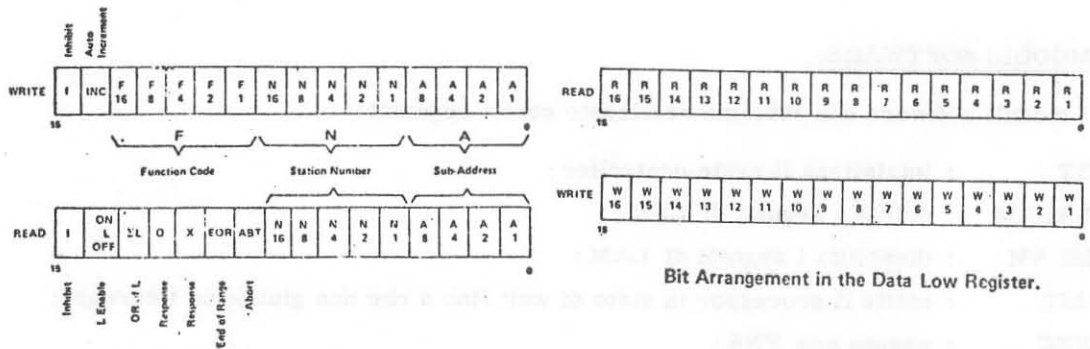
Nel presente lavoro sono presentate alcune routines in MACRO11 da noi realizzate, che costituiscono un software di base utilizzabile per sviluppare i programmi di acquisizione dati in linguaggio evoluto, FORTRAN, sotto sistema operativo RT11 v.03.

2. - REGISTRI DEL CRATE CONTROLLER.

Il crate controller è visto dalla CPU del PDP11 come un insieme di quattro registri a 16 bit di lettura e scrittura che servono a scambiare dati, comandi e informazioni sullo stato del CAMAC.

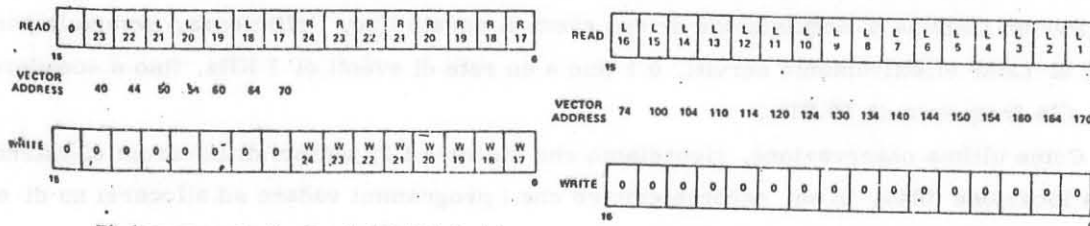
L'indirizzo di tali registri viene determinato da otto ponticelli del crate; nel nostro caso è stato scelto tra 164010_g e 164016_g. All'inizio di una operazione sul dataway CAMAC il Control

Monitor Register (all'indirizzo 164010₈) conterrà l'FNA e le informazioni riguardanti il tipo di trasferimento dati (Fig. 1). Alla fine nel Control Monitor Register si troveranno le infor-



Bit Arrangement in the Control Monitor Register.

Bit Arrangement in the Data Low Register.



Bit Arrangement in the Data/LAM High Register.

Bit Arrangement in the LAM Low Register.

FIG. 1

mazioni relative all'esito dell'FNA e in particolare i valori di X e Q.

Gli altri tre registri (Data Low Register, Data High Register e Lam Low Register) vengono utilizzati per accumulare i dati scambiati tra il calcolatore e il CAMAC e a segnalare le stazioni che eventualmente hanno generato il segnale di LAM (Fig. 1).

3. - LAM PROCESSING.

Il crate controller associa le 23 stazioni del crate CAMAC a un set di 23 vettori di interrupt, costituiti ciascuno da due word della memoria del PDP11.

L'indirizzo di base di questo insieme di vettori è modificabile mediante tre ponticelli che risiedono sull'interfaccia; nel nostro caso è 1040₈.

Il segnale di LAM può essere generato da un cassetto che ha completato una operazione di I/O. Quando il crate controller riceve questo segnale lancia a sua volta una Bus Request al PDP11.

Il servizio di una Bus Request nel PDP11 consiste in alcune operazioni svolte dall'hardware che provvedono a salvare nello stack i valori correnti del Program Counter e del Pro-

gram Status e a caricare al loro posto i valori programmati nel vettore di interrupt, nel nostro caso associato alla stazione da cui è partito il LAM.

4. - MODULI SOFTWARE.

I moduli software che abbiamo realizzato sono i seguenti :

- a) INIT : inizializza il crate controller ;
- b) ENALAM : abilita i segnali di LAM ;
- c) DISLAM : disabilita i segnali di LAM ;
- d) WAIT : mette il processor in stato di wait fino a che non giunge un interrupt ;
- e) EXEC : esegue una FNA ;
- f) LAMLNK : collega il LAM di una stazione a una routine di servizio.

Di seguito sono riportati i sorgenti di questi programmi autodocumentati (Figg. 2, 3 e 4). In Fig. 5 è riportato un programma FORTRAN usato per provare alcune di queste routines.

Per un sistema di acquisizione driven event di questo tipo, l'efficienza, ovvero la percentuale di LAM effettivamente serviti, è 1 fino a un rate di eventi di 1 KHz, fino a scendere al 10% alla frequenza di 10 KHz.

Come ultima osservazione, ricordiamo che siccome i 23 vettori di interrupt si estendono dalla locazione 1040₈ in su, occorre evitare che i programmi vadano ad allocarsi su di essi. Ciò è possibile per esempio con l'opzione :

```
.../BOTTOM: 1500
```

nella fase di link dei programmi.

```
; ROUTINE DI INIZIALIZZAZIONE CRATE CONTROLLER

        .TITLE INIT
        .GLOBL INIT

INIT:
        MOV     #32711,@#164010
        MOV     #32710,@#164010
        MOV     #30751,@#164010
        RETURN
        .END

; ROUTINE ABILITAZIONE LINEE LAM

        .TITLE ENALAM
        .GLOBL ENALAM

ENALAM:
        MOV     #32752,@#164010

        RETURN
        .END

; ROUTINE DISABILITAZIONE LINEE LAM

        .TITLE DISLAM
        .GLOBL DISLAM

DISLAM:
        MOV     #30752,@#164010

        RETURN
        .END

; ROUTINE DI ATTESA A UN INTERRUPT ESTERNO

        .TITLE WAIT
        .GLOBL WAIT

WAIT:
        WAIT
        RTS     PC
        .END
```

```
*****
#
#           E X E C
#
#*****
#
#VERSIONE 1.0 HW:PDF11/34 + CAMAC CON CRATE CONTROLLER ORTEC DC 011
#AUTORI:
#   M. ANGHINOLFI  -I.N.F.N. SEZ. GENOVA
#   F.  MASULLI   -I.S.F. UNIVERSITA' DI GENOVA
#
#
#ESEQUE IL CICLO CAMAC CARICANDO IL
#CONTROL MEMORY REGISTER CON LA FNA/
#
#CHIAMA IA FORTRAN:
#   INTEGER*4 DATO
#           .
#           .
#           .
#   CALL EXEC(IF,IN,IA,DATO,IQ,IX)
##
#
```

CMR:	.WORD	164010	#CONTROL MONITOR REGISTER
DLB:	.WORD	164012	#DATA LOW BUFFER
DHB:	.WORD	164014	#DATA HIGH BUFFER
LHB:	.WORD	164015	#LAM HIGH BUFFER
LLB:	.WORD	164016	#LAM LOW BUFFER
IF:	.WORD		#VALORE DI F
IN:	.WORD	0	#NUMERO DELLA STAZIONE
IA:	.WORD	0	#VALORE DEL SOTTOINDIRIZZO
DATO:	.BLKW	2	#BUFFER PER I DATI SCAMBIATI
BUFFER:	.WORD	0	
IQ:	.WORD	0	#STATO DEL DISPOSITIVO
IX:	.WORD	0	#COMANDO ACCETTATO

MESSAGGI DI ERRORE

```
M1: .ASCIZ/ USO DI F ERRATO /
M2: .ASCIZ/ TENTATIVO DI INDIRIZZAMENTO A N>23./
     .EVEN
```

```
.NCALL .PRINT
.TITLE EXEC
```

###PRENDE I PARAMETRI

EXEC::

```
TST   (R5)+
MOV   @(R5)+,IF
MOV   @(R5)+,IN
MOV   @(R5)+,IA
MOV   @(R5)+,DATO
MOV   @(R5),DATO+2
```

#CONSTRUISCE LA PAROLA FNA E LA CARICA NEL C.M.R.

FIG. 3a

```
MOV R3, -(SP)
MOV IF, BUFFER
MOV #5, R3
2#: ASL BUFFER
DEC R3
BNE 2#
ADD IN, BUFFER
MOV #4, R3
3#: ASL BUFFER
DEC R3
BNE 3#
ADD IA, BUFFER
MOV BUFFER, @CMR
;
; DETERMINA IL TIPO DI F
;
BIT #10, IF
BNE OUT1 ;FB=1: F NON GENERA CILO CAMAC
BIT #20, IF
BNE W ;F16=1: F USA LE W LINES
BR R ;F16=0: F USA LE R LINES
OUT1: MOV (SP)+, R3
RETURN
W: MOVE DAT0+2, @DHB ;START CICLO CAMAC DI SCRITTURA
MOV DAT0, @ILB
BR OUT2
R: MOV @DLB, DAT0 ;START CICLO CAMAC DI LETTURA
MOVE @DHB, DAT0+2
CLRB DAT0+3
OUT2: BIT #1000, @CMR ;TESTA L'ABORT BIT
BEQ 10#
.PRINT #M1 ;IL BIT E' SETTATO: STAMPA IL MESSAGGIO
10#: BIT #2000, @CMR ;TESTA L'END-OF-RANGE BIT
BEQ 20#
.PRINT #M2 ;IL BIT E' SETTATO: STAMPA IL MESSAGGIO
;
; CARICA I RISULTATI DEL CICLO
;
20#: MOV DAT0+2, @R5
MOV DAT0, @R5
MOV #1, IQ
BIT #1000, @CMR ;TESTA IQ
BNE 30# ;IQ=1
CLR IQ ;IQ=0
30#: MOV IQ, @4(R5)
MOV #1, IX
BIT #4000, @CMR ;TESTA IX
BNE 40# ;IX=1
CLR IX ;IX=0
40#: MOV IQ, @6(R5)
MOV (SP)+, R3 ;RIPRENDE R3 ED ESCE
RTS PC
.END
```

FIG. 3b

```
#####
*
*   L A M L N K
*
#####
*
*
* VERSIONE 1.0
*
* HW: PDP11/34 E CAMAC CON CRATE CONTROLLER ORTEC DC 011
*
* DATA 1-10-80
*
* AUTORI: M.ANGHINOLFI-I.N.F.N. SEZ. DI GENOVA
*         F.MASULLI   -IST.SCIENZE FISICHE UNIVERSITA' DI GENOVA
*
*
* COLLEGATORE-ESECUTORE DI ROUTINES FORTRAN DI SERVIZIO AI LAM
* DEI CASSETTI DEL CRATE CAMAC
* RENDE TRASPARENTE ALL'UTENTE LA MACCHINA DIGITAL
*
*
* CHIAMATA FORTRAN:
*   EXTERNAL ISUB1,ISUB2
*
*   .
*   .
*   .
*
*   CALL LAMLNK(ISTAZ1,ISUB1,IPRIO1)
*   CALL LAMLNK(ISTAZ2,ISUB2,IPRIO2)
*
* ISUB: ROUTINE DI SERVIZIO
* ISTAZ: NUMERO STAZIONE CUI COLLEGARE ISUB
* IPRIO: PRIORITA' CON CUI ENTRA ISUB
*
*
*
* PRIO: .WORD 0  *PRIORITA' ROUTINE DI INTERRUPT
* STAZ: .WORD 0  *N.STAZIONE CUI E' COLLEGATA LA ROUTINE DI INTERRUPT
* ISUB: .WORD 0  *INDIRIZZO DI PARTENZA DELLA ROUTINE DI INTERRUPT
* BASE: .WORD 0  *VETTORE DI INTERRUPT CORRISPONDENTE A ISUB
* VSUB: .BLKW 23, *VETTORE CHE CONTERRA' GLI START ADDRESS DELLE ROUTINES
*         *DI SERVIZIO AI LAM
*
*
** CODICE DI INIZIALIZZAZIONE
*
*   .TITLE LAMLNK
*
LAMLNK!!
*
* A) CARICA I PARAMETRI
*
*
*       TST      (R5)+
*       MOV      0,(R5)+*STAZ
*       MOV      (R5)+,J5 IN
*       MOV      0,(R5),PRIO
```

FIG. 4a


```

;
;B) COLLEGATORE

      MOV     STAZ,R4  ;COMPILA L'ELEMENTO DI VSUB
      ASL     R4       ;CORRISPONDENTE A STAZ CON
      ADD     #VSUB-2,R4 ;L'INDIRIZZO DELLA ROUTINE FORTRAN ISUB
      MOV     ISUB,(R4)

; CALCOLO INDIRIZZO ROUTINE DI INTER. (LAM23-LAM1)

      NEG     STAZ
      MOV     STAZ,R0
      ADD     #23.,R0 ;R0=23.-STAZ
      ASL     R0
      ASL     R0
      ASL     R0      ;R0=R0*8.
      ADD     $LAM23,R0
      ADD     #23.,STAZ
      ASL     STAZ
      ASL     STAZ
      ADD     #1040,STAZ
      MOV     STAZ,BASE ;BASE=(23+STAZ)*4+1040
      MOV     R0,BASE   ; IL PRIMO ELEMENTO DEL VETTORE DI INTER.
                        ; L'INDIRIZZO DEL LAM CORRISPONDENTE
;
;C)CALCOLO DELLA NUOVA PSW
;
      MOV     #5,R4
LAB1:  ASL     PRIO
      DEC     R4
      BNE    LAB1
      ADD     #2,BASE ;IL SECONDO ELEMENTO DEL VETTORE DI INTER.
      MOV     PRIO,BASE ;CONTIENE LA NUOVA PSW
      RTS     PC

;; CODICE DI GESTIONE INTERRUPT

;
; E' COSTITUITO DA 23 ROUTINE DI INTERRUPT CHE VENGONO LANCIATE AD HW
; TERMINANO CON RTI E RITORNANO AL PROGRAMMA CORRENTE
; INDIVIDUANO IN VSUB LO START ADDRESS DELLA ROUTINE DI SERVIZIO
; COLLEGATA NELLA PRIMA PARTE

LAM23: ;ROUTINE DI INTER. ASSOCIATA AL LAM GENERATO DAL 23 CASSETTO
      MOV     #44.,R3
      JMP     LAM
LAM22: MOV     #44.,R3 ; * * * * * DAL 22-ESIMO CASSETTO ECC.
      JMP     LAM
LAM21: MOV     #42.,R3
      JMP     LAM
LAM20: MOV     #40.,R3
      JMP     LAM
LAM19: MOV     #38.,R3
      JMP     LAM
LAM18: MOV     #36.,R3
      JMP     LAM
LAM17: MOV     #34.,R3
      JMP     LAM
LAM16: MOV     #32.,R3
      JMP     LAM
LAM15: MOV     #30.,R3
      JMP     LAM

```

FIG. 4b

```
LAM14: MOV    #28.,R3
        JMP    LAM
LAM13: MOV    #26.,R3
        JMP    LAM
LAM12: MOV    #24.,R3
        JMP    LAM
LAM11: MOV    #22.,R3
        JMP    LAM
LAM10: MOV    #20.,R3
        JMP    LAM
LAM9:  MOV    #18.,R3
        JMP    LAM
LAM8:  MOV    #16.,R3
        JMP    LAM
LAM7:  MOV    #14.,R3
        JMP    LAM
LAM6:  MOV    #12.,R3
        JMP    LAM
LAM5:  MOV    #10.,R3
        JMP    LAM
LAM4:  MOV    #8.,R3
        JMP    LAM
LAM3:  MOV    #6.,R3
        JMP    LAM
LAM2:  MOV    #4.,R3
        JMP    LAM
LAM1:  MOV    #2.,R3
```

```
LAM:
?  MOV    #VSUB-2,R2      ;INDIVIDUA IN VSUB LO START ADRS. DELLA
  ADD    R3,R2          ;ROUTINE DI SERVIZIO
  JSR    PC,@(R2)       ;LANCIA LA ROUTINE DI SERVIZIO IN FORTRAN
  RTI
```

.END

```
C  PROGRAMMA PER LEGGERE I DATI DA UN CASSETTO
C  RICHIEDE UN TDC / ADC NELLA STAZIONE NUMERO N
C
C
C      EXTERNAL ISUB
C      INTEGER*4 DATO
C      COMMON/NN/N,IA
C      COMMON/AA/IA,IX
C      COMMON/DD/DATO
C
C      CALL INIT      ! INIZIALIZZA IL CRATE
C      CALL ENALAM    ! ABILITA IL LAM
C
C  LEGGE I DATI
C
C      CALL PRINT(' PRIORITA ?')
C      READ(5,12) IP
12  .  FORMAT(13)
C      CALL PRINT(' STAZIONE? ')
C      READ(5,12) N
C      CALL PRINT(' SOTTOINDIRIZZO ?')
C      READ(5,12) IA
C
C  CARICA NEL VETTORE DI INTERRUPT L'INDIRIZZO DELLA
C  ROUTINE DI SERVIZIO
C      CALL LAMLNK(N,ISUB,IP)
C
C  ABILITA IL LAM DEL CASSETTO
C      CALL EXEC(26,N,IA,DATO,0,0)
C
C  ATTEDE IL LAM DEL CASSETTO
3   CONTINUE
C      CALL WAIT
C      GOTO 3
C      STOP
C      END
C
C      SUBROUTINE ISUB
C      INTEGER DATO(2)
C      COMMON/NN/N,IA
C      COMMON/AA/IA,IX
C      COMMON/DD/DATO
C
C  LEGGE UN DATO
C      CALL EXEC(2,N,IA,DATO,IA,IX)
C      RETURN
C      END
```