



ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Genova

INFN/TC-07/8
26 Giugno 2007

**SISTEMA DI ACQUISIZIONE PER IMPIEGO IN LABORATORIO DI FISICA
NUCLEARE, BASATO SU BRIDGE USB CAEN® V1718**

M. Ivaldi, M. Battaglieri, A. Bersani, M. Brunoldi, R. De Vita, G. Ottonello, D. Piombo,
A. Rottura

*INFN-Sezione di Genova, c.o. Dipartimento di Fisica - Università degli Studi di Genova,
I-16146 Genova, Italy*

Abstract

Nella presente pubblicazione è descritto il funzionamento hardware e software di un sistema di acquisizione, basato sull'utilizzo di un Bridge CAEN® V1718, assemblato e testato nella seconda metà del 2006 presso il Gruppo III INFN Sezione di Genova.

*Published by SIS-Pubblicazioni
Laboratori Nazionali di Frascati*

1 INTRODUZIONE

Per l'impiego in fase di test e prototipaggio per misure di fisica nucleare, è stato sviluppato un sistema di acquisizione VME multi-purpose, basato su un controllore CAEN V1718.

Tale controllore appartiene alla famiglia dei cosiddetti master modules e rappresenta la parte "intelligente" di una sezione di acquisizione VME: utilizzando per l'interfaccia al PC il bus USB 2.0, capace di trasferire dati in modalità High Speed Bulk Transaction fino a 30Mbyte/s, è dotato di un buffer di 128KB che evita rallentamenti nel trasferimento nel caso il rate risulti maggiore della massima velocità del flusso del protocollo.

Il sistema è stato alloggiato in un rack che potesse ospitare la componentistica ordinata secondo il criterio mostrato in fig. 1.

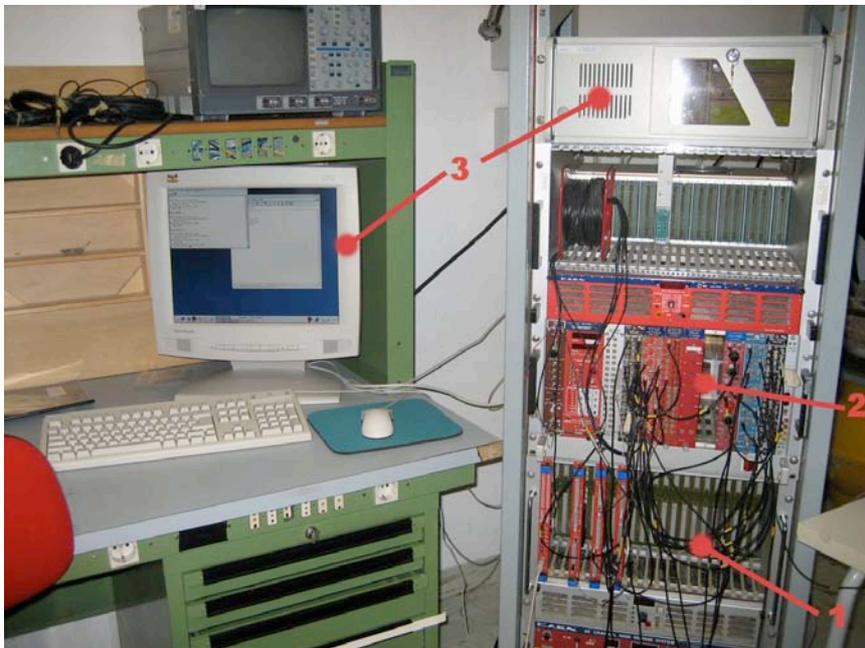


FIG. 1: (1) BackPlane VME nel quale sono presenti, da sinistra a destra nell'ordine: Bridge V1718, modulo CAEN V775 e CAEN V792 (2) BackPlane e moduli NIM (3) Monitor, tastiera, mouse e PC Host.

Una configurazione di questo genere, integrando l'elettronica VME, largamente presente in ogni laboratorio di fisica nucleare ed un programma modulare, compatibile su qualsiasi PC, permette una notevole scalabilità e si presta ad ogni sviluppo futuro.

2 DESCRIZIONE DEL SISTEMA: HARDWARE

Sul mercato esistono svariati moduli per il controllo di unità VME, basati su tecnologie diverse; esistono sia unità master dedicate al controllo diretto su rack che bridge in grado di interfacciare l'unità ad un PC Host esterno.

Tra i bridge sono presenti soluzioni sia di tipo standardizzato a larga scala (come quelle Ethernet o USB), che di tipo ad interfacciamento proprietario (come ad esempio gli strumenti della National Instruments). Il fatto che, nel nostro caso, le scelte siano ricadute sulla prima delle due soluzioni, sono date dal fatto che il protocollo USB, standard ormai universale e presente su qualsiasi PC, consente la completa compatibilità con le attrezzature disponibili nei laboratori, è relativamente economico rispetto ad altri e le periferiche possono essere facilmente programmate secondo le proprie esigenze.

Tuttavia, un'architettura di questo tipo ha dei limiti intrinseci, dipendenti dalla struttura stessa della comunicazione. Essa è infatti di tipo Master/Slave e non Peer-to-Peer; ciò implica che la direzione del segnale del buffer guidante è controllata dal Master e la sincronizzazione per il cambio della direzione è impostata nell'architettura del protocollo. Il trasferimento avviene attraverso una struttura molto simile al TCP/IP: i dati vengono suddivisi in pacchetti e successivamente inviati bidirezionalmente.

Questo insieme di caratteristiche strutturali comporta ritardi temporali nel trasferimento dei pacchetti qualunque sia la modalità operativa dell'USB utilizzata (le due tecniche base sono handshaking e token). Tali limitazioni, nelle applicazioni di un tipico laboratorio di fisica nucleare, sono comunque accettabili per il tipo di impiego che il sistema deve sostenere (basso rate e numero limitato di canali).

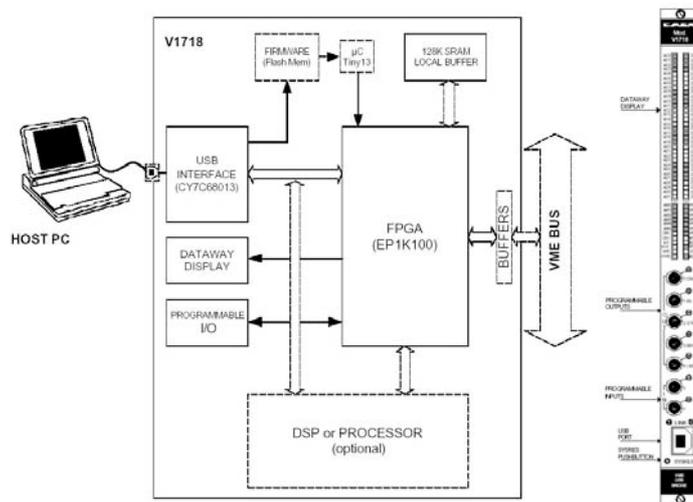


FIG. 2: Schema a blocchi dell'architettura interna del bridge e visione del pannello frontale

Nella configurazione iniziale, assemblata per lavorare sullo sviluppo e sul test del sistema, sono stati inseriti: un ADC, un TDC ed uno Scaler della CAEN di recente concezione, provvisti di una memoria bufferizzata sull'uscita, denominata MEB (Multiple Event Buffer).

Nelle schede che dispongono di tale buffer, infatti, possono essere memorizzati fino a 32 eventi organizzati secondo una logica FIFO, con puntatori in lettura/scrittura e controllo sugli stati "Status Full" e "Status Empty" dell'uscita.

Nel nostro caso specifico, vengono riportate brevemente di seguito le caratteristiche delle tre schede Slave utilizzate.

- **CAEN V775** è un'unità che incorpora 32 convertitori TDC (Time-to-Digital) a 32 canali. Il valore di fondo scala può essere impostato tra 140ns e 1,2 μ s con una risoluzione di 8 bit; può operare sia in modalità *Common Start* che in *Common Stop* e ha numerose caratteristiche flessibili, quali la soppressione programmabile dei livelli di zero e un trigger counter.
- **CAEN V792** è un'unità che incorpora 32 convertitori QDC (Charge-to-Digital) a 32 canali con integrazione degli ingressi negativi. Per ciascun canale, l'ingresso viene convertito in livelli di tensione QAC (Charge-to-Amplitude Conversion), nel range 0÷400pC. Le uscite della sezione QAC sono multiplexate e successivamente convertite da due fast ADC a 12bit.
- **CAEN V830** è un'unità Multievent Latching Scaler, con 32 canali indipendenti; ciascuno di essi ha una risoluzione a 32bit ed accetta sia ingressi ECL che LVDS. La frequenza massima di ingresso è circa 250MHz e tra le caratteristiche avanzate vi è la gestione di segnali come VETO, CLEAR ECL e TEST NIM input.

Molto importante, alla luce dell'impiego con il bridge USB, è il pieno supporto da parte di tutte le schede per i comandi avanzati di Block Transfer e per quelli di Multicast.

3 DESCRIZIONE DEL SISTEMA: SOFTWARE

L'idea alla base del software di controllo è quella di rendere il più semplice possibile l'espansibilità del sistema: si utilizza per questo un *database* di moduli slaves VME facilmente accessibile ed aggiornabile attraverso l'impiego di files di configurazione.

Da un'analisi sulle scelte più convenienti da adottare per la stesura del software, si è deciso di procedere secondo lo schema riportato di seguito.

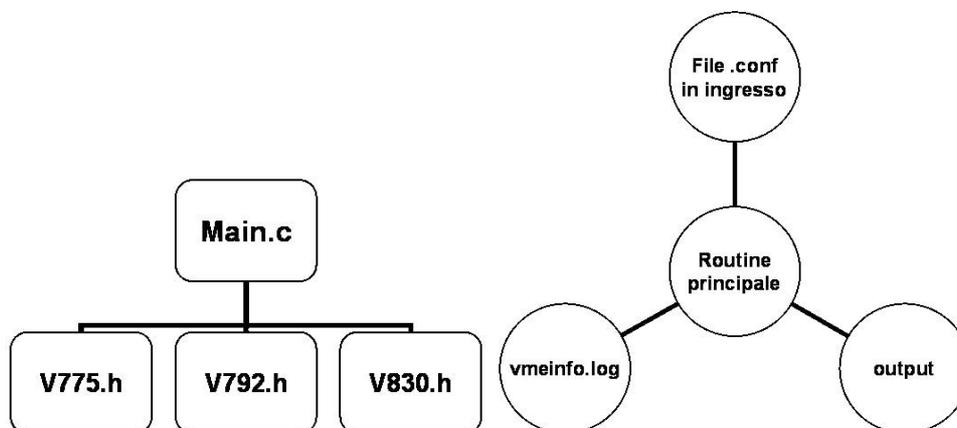


FIG. 3: Schema a blocchi della struttura software.

Il programma si compone di un insieme di files sorgenti scritti in C e funzionanti in ambiente operativo Scientific Linux kernel 2.4.21-27: sono previste la lettura di alcuni files di inizializzazione, configurazione ed essenzialmente la scrittura di due files di output.

Nella RAM del PC su cui il programma gira, si sono create delle *immagini* della struttura dei registri dei moduli slaves sulle quali operare direttamente.

La routine principale carica un certo numero di files header, in funzione del tipo di moduli slaves inseriti nel Bus e, successivamente, legge ed interpreta i files di configurazione, la cui funzione è riportata di seguito:

- **vmebus.conf** contiene un insieme minimo di dati necessari al setup dei moduli e alle impostazioni iniziali, passati alla routine principale attraverso l'impiego di parole-chiave;
- **vXXX.conf** sono i files di configurazione che contengono le parole-chiave ed i valori che vengono scritti nei registri di ogni singolo modulo, necessari all'inizializzazione. Nel caso di inserimento nel *database* di una nuova scheda, è necessario procedere alla stesura di un nuovo file di questo tipo.

Le funzioni e le strutture raggruppate nei files header, sono invece organizzate nella maniera seguente:

- **struttura relativa alla mappa dei registri** in cui sono definite delle variabili, che fanno riferimento al nome e all'indirizzo dei registri dei vari moduli ed occupano in memoria uno spazio logico corrispondente;
- **struttura relativa alla mappatura della della ROM** in cui sono definite variabili che fanno riferimento al nome e all'indirizzo degli elementi del Firmware dei vari moduli ed occupano in memoria uno spazio logico corrispondente;
- **funzione di lettura e stampa della mappa dei registri** che consente di eseguire un ciclo di read di tutti i registri, restituendo il valore letto per ognuno di essi;
- **funzione di lettura e stampa del Firmware delle schede** che consente di eseguire un ciclo di read del firmware, analogamente al caso precedente;
- **funzione di interpretazione delle parole-chiave dei files vXXX.conf** che consente di interpretare il valore di inizializzazione relativo ai vari registri e definiti nei singoli files vXXX.conf.

I dati acquisiti e le informazioni relative all'inizializzazione delle schede, vengono salvati in memoria e, al termine dell'acquisizione, trasferiti nei files in uscita descritti di seguito:

- **vmefinfo.log** - contiene le informazioni relative alle singole schede inserite, ovvero un elenco dello stato dei registri e del firmware al momento della scrittura dei files di configurazione ed un *dump* del sistema al momento del runtime iniziale. Tutto ciò è utile sia in fase di debugging che per verificare il corretto funzionamento del sistema di acquisizione attraverso un riassunto delle variabili di stato.

- **vmeoutput.txt** - contiene i dati dell'acquisizione, organizzati secondo un formato compatibile con i principali software usati per le analisi.

4 SPECIFICHE E PRESTAZIONI DEL SISTEMA

Sono stati effettuati test di calibrazione sulle tre diverse schede al fine di verificarne le prestazioni e di valutare il comportamento dell'intero sistema.

Calibrazione dei moduli

QDC

La procedura di calibrazione effettuata ha consentito di valutare la linearità della risposta del modulo; a tale scopo, è stato impiegato il setup descritto in figura 4:



FIG. 4: Foto dell'hardware e della strumentazione impiegati per i test. Il generatore di impulsi, visibile nella parte destra, è collegato con l'elettronica NIM del rack; esso fornisce ai moduli VME le temporizzazioni e i segnali di Gate, oltre che alla forma d'onda per l'acquisizione, impiegata come riferimento.

- Per generare gli impulsi da registrare, è stato utilizzato un generatore Hewlett-Packard mod.8082A, che è stato regolato su diversi valori di riferimento in ampiezza;
- per confronto, è stato utilizzato un oscilloscopio digitale LeCroy Waverunner 6100A, con funzione di QDC, essendo questi in grado di integrare i segnali forniti in ingresso.

In particolare, i due test effettuati hanno consentito:

- 1) la verifica della relazione di linearità tra ampiezza del segnale in uscita dal generatore e tra il valore in uscita al QDC, convertito in carica;
- 2) la verifica della corrispondenza tra i valori rilevati con due diversi sistemi di acquisizione: oscilloscopio digitale e QDC.

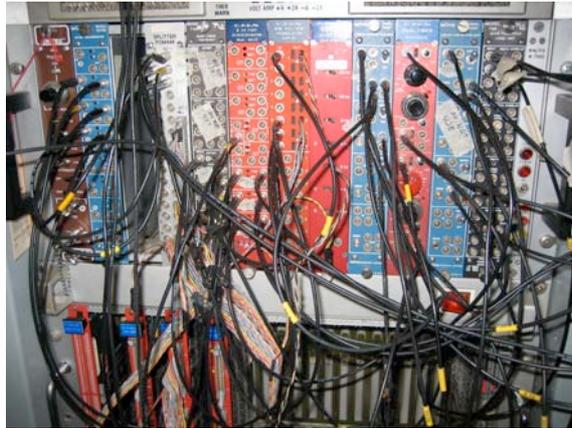


FIG. 5: Per una corretta impostazione dell'acquisizione, il segnale del generatore è stato elaborato da una catena NIM.

I successivi grafici evidenziano i risultati dei due test sopra descritti.

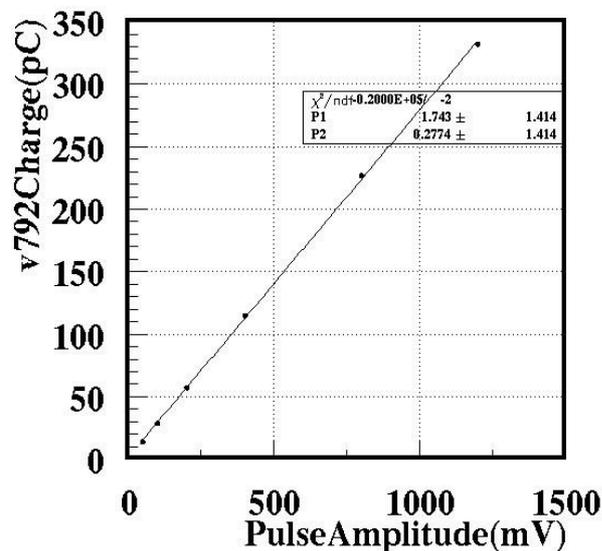


FIG. 6: relazione tra tensione di riferimento generata con il Pulse Generator 8082A e il valore di carica acquisito con il QDC, convertito nella scala dei pC. Il fit dei dati sperimentali è una retta che tiene conto dei range di errore relativi alla stabilità dell'impulso generato ed alla risoluzione del modulo V792.

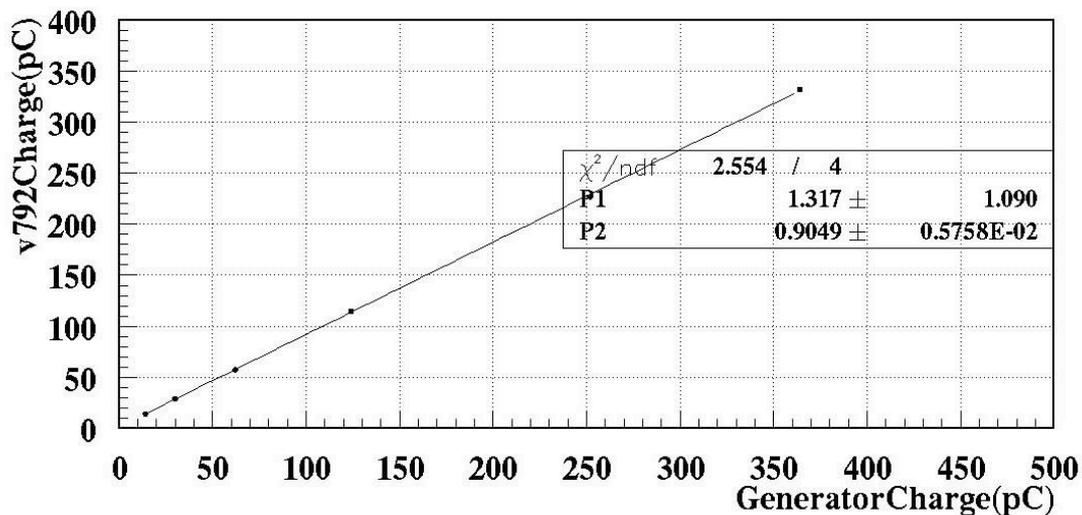
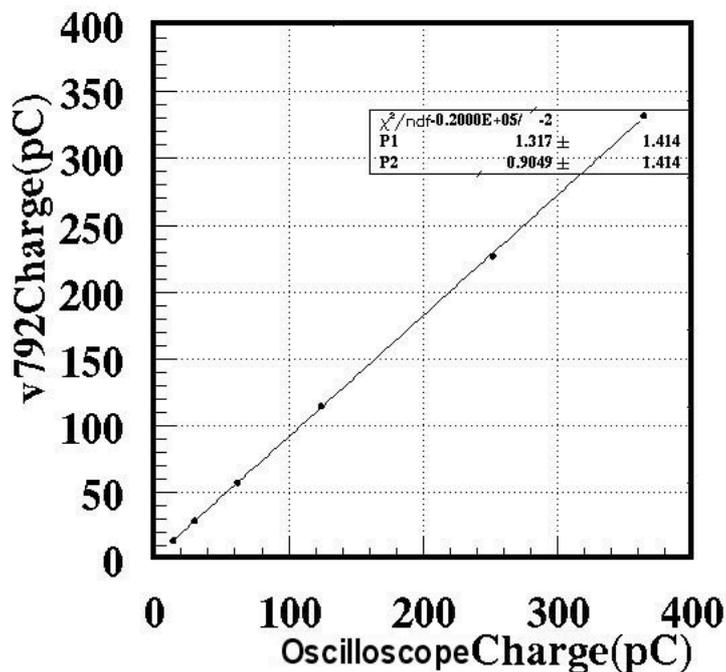


FIG. 7: i due grafici fanno rispettivamente riferimento a rilevamenti effettuati con corrente di piedistallo misurata a vuoto e fuori tempo rispetto alla finestra di Gate (Out of Time). La carica generata, misurata con l'oscilloscopio è rappresentata in ascissa; in ordinata vi è quella misurata dal QDC: è attesa una relazione lineare del tipo $y = x$; dal fit dei dati sperimentali emerge che i parametri della retta sono compatibili con i valori attesi entro un certo range di errore dato dalla risoluzione del QDC.

TDC

La procedura operativa seguita per calibrare il TDC V775 è simile a quella utilizzata nel caso del QDC.

Dovendo valutare dei tempi, il generatore di impulsi è stato regolato su grandezze fissate e, per misurare l'attività del modulo tra i segnali di START e STOP, il segnale di ingresso al canale è stato temporizzato inserendo dei ritardi noti.

Il test ha consentito, analogamente a quanto fatto con il QDC, di valutare la relazione di linearità tra la temporizzazione del segnale in uscita dal generatore ed il valore in uscita al TDC.

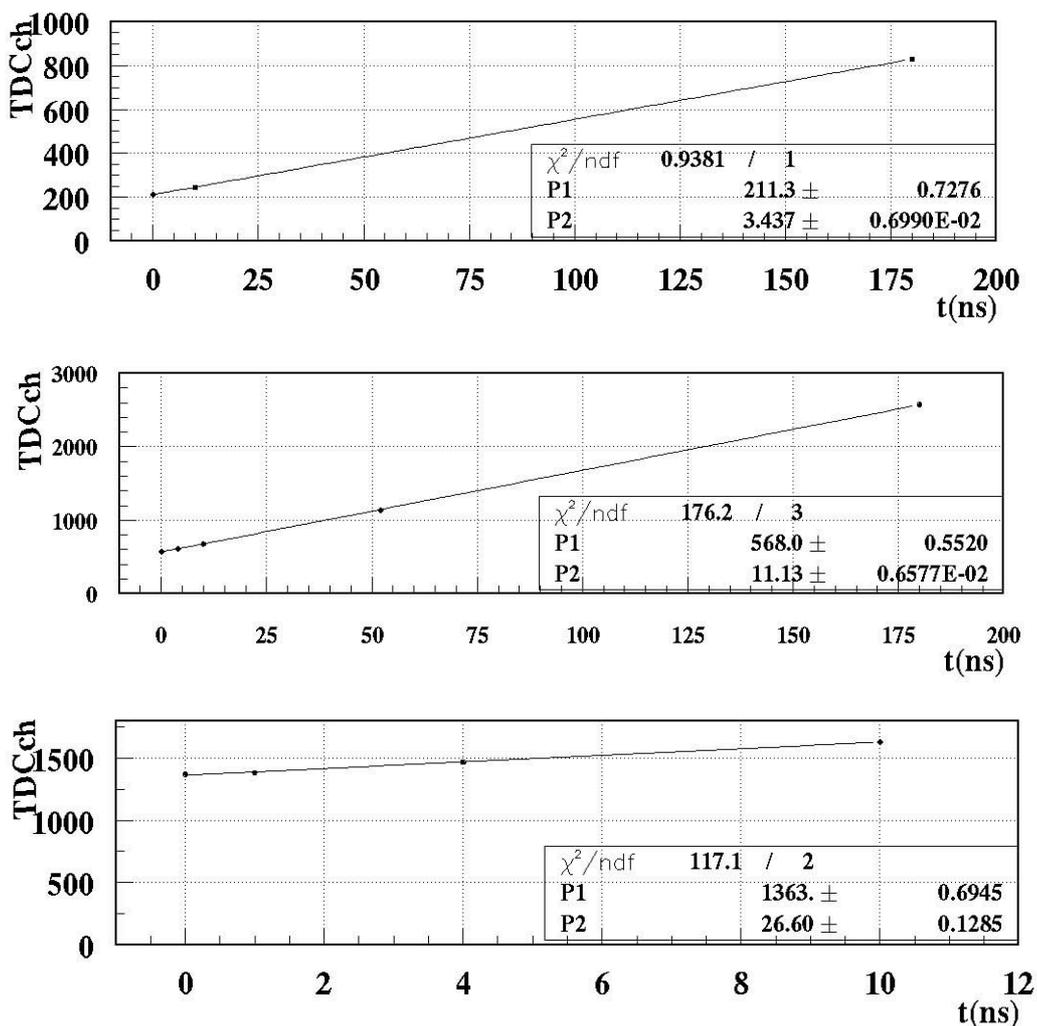


FIG. 8: Sono stati eseguiti rilevamenti per tempi fissati con tre diversi valori di fondoscala FSR (parametro impostato sul TDC). Anche qui, dal fit dei dati, considerando l'errore sull'asse delle ordinate relativo alla risoluzione del V775, la relazione risulta essere lineare.

I grafici si riferiscono a diversi valori di Full Scale Range, rispettivamente $FSR = 30$, $FSR = 100$ e $FSR = 255$, ovvero risoluzioni temporali rispettivamente di 0.33ns , 100ps e 39ps .

Scaler

L'analisi del comportamento dello Scaler consente di fare considerazioni globali sul funzionamento dell'intero sistema. Il modulo V830 conta il numero di eventi in un tempo definito; questo dato, se confrontato con il numero di eventi attesi, permette di fare valutazioni sul numero di trigger persi in funzione del rate.

Nel cablaggio NIM il segnale, che nei casi precedenti era stato inviato ai Gate degli ADC, viene discriminato e convogliato ad uno dei 32 canali dello Scaler. Il modulo è quindi impostato in modo da fornire i conteggi in un tempo $\Delta t = 1$ secondo; incrementando il rate di ingresso i conteggi ne seguono l'andamento sino ad un certo valore, per poi saturare.

I test eseguiti aumentando di volta in volta il numero di moduli inseriti nel backplane, hanno consentito di individuare proprio nell'impiego simultaneo di più moduli una delle principali cause che limitano le prestazioni.

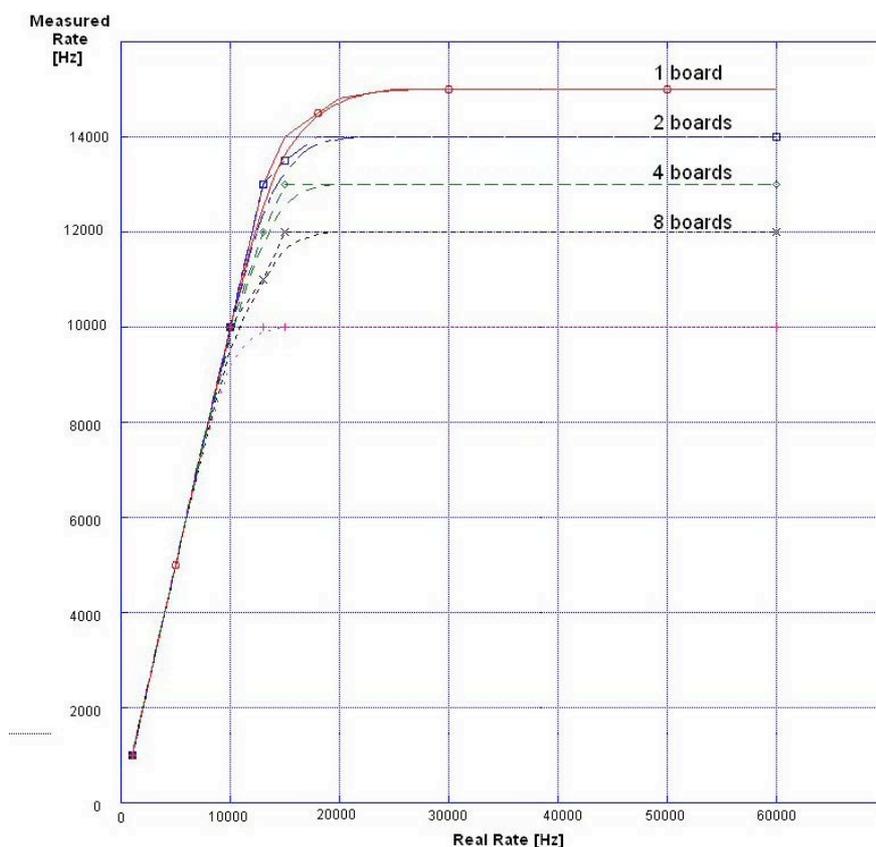


FIG. 9: Il segnale in ingresso è stato impostato con diversi valori di frequenza. I risultati forniti dallo Scaler hanno consentito di determinare la relazione tra frequenza in ingresso e quella misurata dallo scaler. Si è visto che, aumentando le schede slaves inserite, la relazione tende a saturare in corrispondenza di rate più bassi.

Un'ulteriore considerazione si può fare dall'analisi del protocollo USB.

In esso, le transizioni avvengono attraverso pacchetti, secondo la logica del polling: ogni volta che l'host vuole ricevere dati da una periferica, le invia un token; se la periferica ha dati da inviare, li invia e l'host risponde con un pacchetto di handshake.

Ogni transizione è composta dalle seguenti fasi:

- Token phase, in cui l'host inizializza il pacchetto indicando il tipo di transizione;
- Data phase, che rappresenta la fase di invio dei dati tramite pacchetti. La direzione dei dati è la medesima indicata nella token phase;
- Handshake phase, nella quale viene indicato il successo o il fallimento della transizione.

La modalità utilizzata dal Bridge V1718 (Bulk Transfer), è quella che consente l'invio massiccio dei dati; la larghezza di banda allocata in ogni transizione varia con la disponibilità del bus al momento del trasferimento.

Sulla banda utile per l'acquisizione, che è rappresentata esclusivamente dalla fase attiva di invio dati nella transizione USB, vanno quindi ad incidere i tempi morti relativi allo scambio del token e dell'handshake.

Nel sistema in esame, in base al grafico precedente, si evince che tali tempi morti non incidono fino ad un rate pari a circa 10KHz: fino a quel valore la risposta è infatti lineare.

Test di acquisizione

Al fine di verificare il comportamento del sistema nel caso di un tipico utilizzo in laboratorio, è stata implementata una catena di acquisizione, impiegando delle sorgenti radioattive Cs137 e Co60 e registrando il segnale di un fototubo Hamamatsu mod.R1652 accoppiato ad un cristallo di NaI.

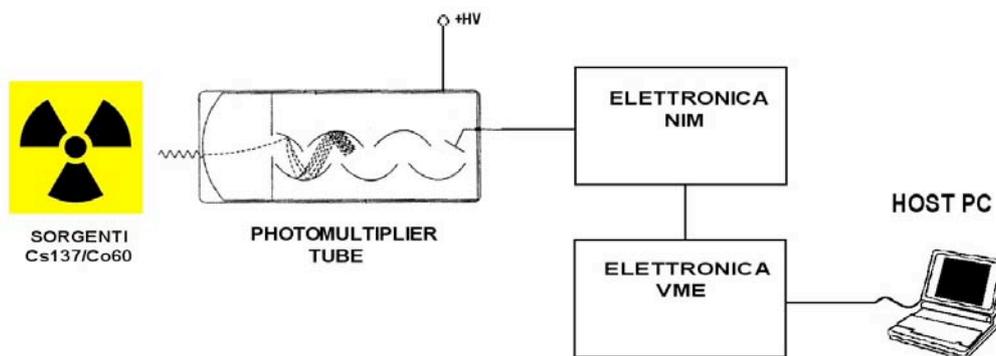


FIG. 10: E' stato possibile determinare lo spettro di emissione delle due sorgenti attraverso le misure del QDC, che fornisce una risposta canale/conteggio. Il test è stato separato in due fasi: nella prima è stato rilevato il picco per il Cs137, nella seconda sono stati rilevati i due picchi caratteristici del Co60.

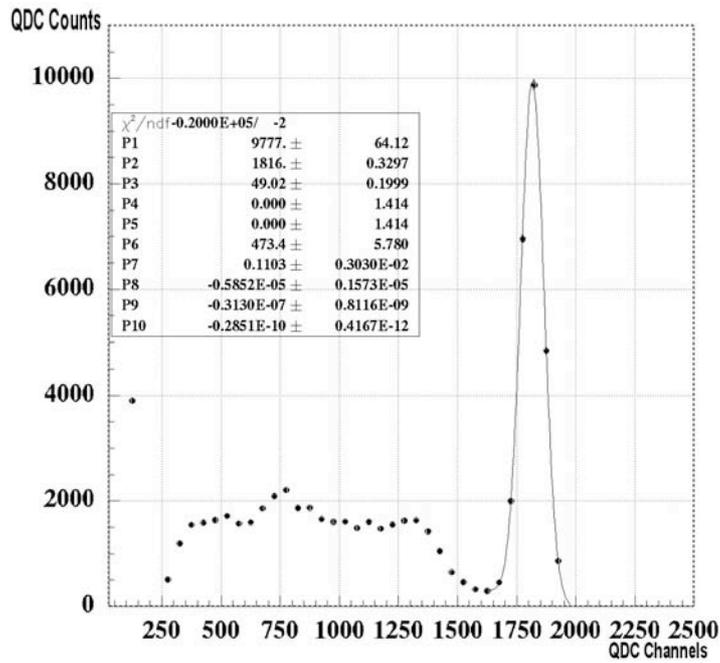


FIG. 11: Nel grafico che si ottiene, relativamente alla prima fase, il picco è in corrispondenza del canale 1816, valore in riferimento al Cs137 (662KeV).

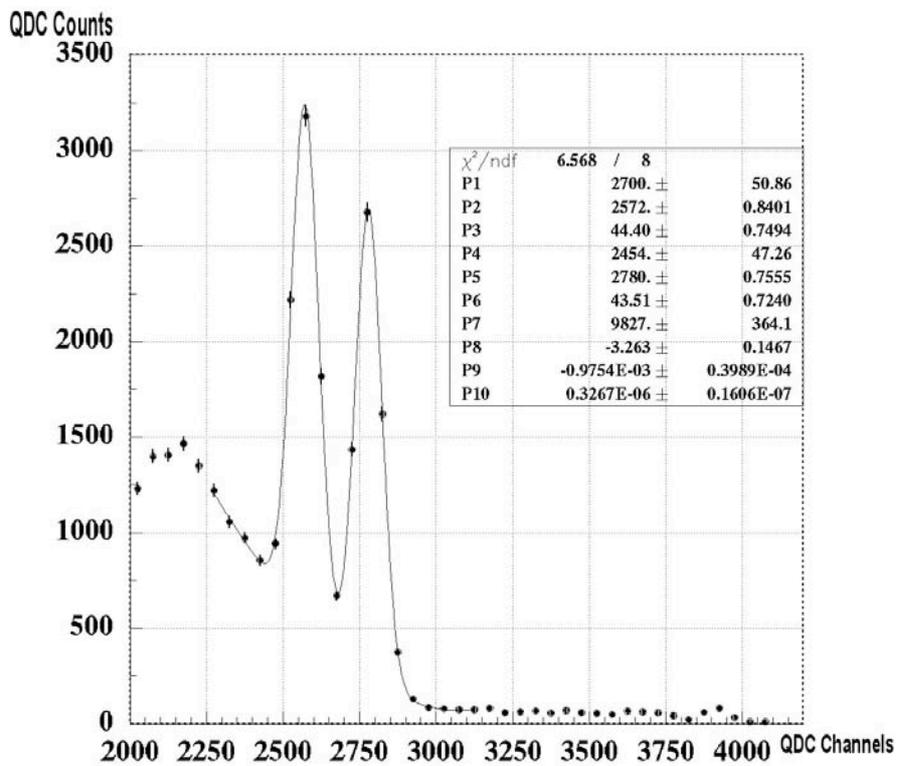


FIG. 12: In questo caso, i due picchi del Co60 vengono rilevati in corrispondenza dei canali 2572 e 2780, pari ad energie di 1173KeV e 1332 KeV.

Essendo note le energie dei fotoni emessi dal decadimento dei 2 elementi (Cs137 e Co60), è stato possibile verificare la linearità della risposta del sistema.

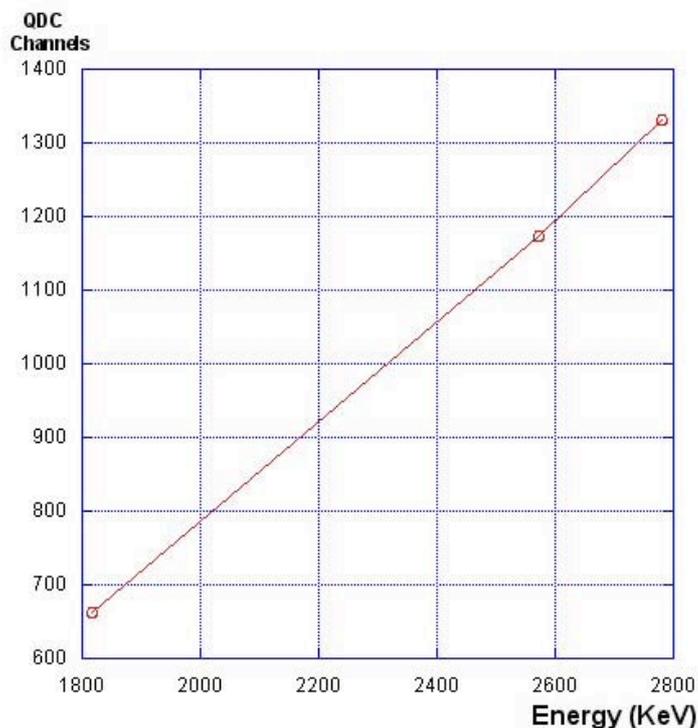


FIG. 13: Grafico Energia/Canali QDC, che evidenzia la relazione di linearità tra i picchi rilevati con le due sorgenti Cesio e Cobalto.

5 CONCLUSIONI

In conclusione il sistema sviluppato, sfruttando le risorse e le attrezzature che possono essere presenti in un laboratorio di fisica nucleare, consente di gestire un sistema di acquisizione semplice ma completo.

Lo standard USB, ampiamente diffuso sui PC lo rende interfacciabile in maniera universale, anche qualora si vogliano realizzare soluzioni portatili, ad uso didattico o per test più complessi; la possibilità di espandere il database dei moduli a seconda delle necessità, rende il sistema capace di rispondere alle singole esigenze degli utenti.

Inoltre, il sistema sviluppato presenta la caratteristica fondamentale di potersi facilmente adattare alle esigenze dell'utente, che ne può gestire la configurazione in maniera flessibile.

APPENDICE

In questa sezione sono riportati i sorgenti principali e, a titolo di esempio, alcuni files di configurazione standard.

1) Porzione di codice relativa al file header v775.h, che crea una struttura relativa al TDC CAEN V775 e alle proprie mappe di memoria:

```
typedef struct V775_Map (*1)
{
    ulong OutBuff [1024]; // REGISTER CONTENT
    ushort FirmRev[1]; // Output Buffer
    ushort GeoAddr[1]; // Firmware Revision
    ushort MCSTAddr [1]; // Geo Address
    ushort BitSet1 [1]; // MCST/CBLT Address
    // Bit Set 1
    ...
    ...
    ushort Thresh [32]; // Thresholds
} V775_Map_t;

typedef struct V775_ROMMemory (*2)
{
    ushort unused1 [19]; //
    ushort OuiMSB [1]; // OUI MSB
    ushort unused2 [1]; //
    ...
    ...
    ushort unused11 [14460]; //
} V775_ROMMemory_t;

typedef struct V775elem (*3)
{
    V775_Map_t *Map;
    V775_ROMMemory_t *Rom;
    struct V775elem *next;
} V775elem_t;

(*4)
void v775_Reg_Init (FILE *cfgfile,long Handle,unsigned int address,V775_Map_t * Map)
{
    // Mappatura dei registri
    CAENVME_ReadCycle(Handle,address+0x1000,((board->Map)->FirmRev),cvA32_U_DATA,
cvD16);
    CAENVME_ReadCycle(Handle,address+0x1002,((board->Map)->GeoAddr),cvA32_U_DATA,
cvD16);
    ...
    ...
    CAENVME_ReadCycle(Handle,address+0x1080,((board->Map)->Thresh),cvA32_U_DATA,
cvD16);
}

(*5)
void v775_Rom_Init (long Handle,unsigned int address,V775_Rom_t * Rom)
{
    // Mappatura della ROM
```

```
CAENVME_ReadCycle(Handle,address+0x8026,((board->Rom)->OuiMSB),cvA32_U_DATA,
cvD16);
...
...
CAENVME_ReadCycle(Handle,address+0x8F06,((board->Rom)->SerialLSB),cvA32_U_DATA,
cvD16);
}
(*6)
void v775_Out_Init (long Handle,unsigned long address,V775elem_t *board)
{
// Mappatura dell'Output Buffer
CAENVME_ReadCycle (Handle,address,((board >Map)->OutBuff),cvA32_U_DATA, cvD32);
}
```

- (*1) Viene creata una struttura, chiamata `V775Map_t`, contenente delle variabili array corrispondenti ai nomi dei registri e dimensionati in relazione alla grandezza degli stessi.
- (*2) La struttura della ROM viene mappata attraverso `V775ROMMemory_t`, con un criterio analogo a quello precedentemente descritto.
- (*3) `V775elem_t` è una struttura contenente i puntatori alle precedenti due strutture, definiti `*Map` e `*Rom` e un puntatore a sé stessa di nome `*next`.
- (*4) Viene dichiarata una funzione che consente di inizializzare tutte le variabili delle struct precedenti agli effettivi valori presenti sulla scheda, letti attraverso un ciclo VME con il comando di libreria `CAENVME_ReadCycle`.
- (*5) Analogamente al caso sopra, viene effettuata una lettura della ROM e inserita nella funzione `v775_Rom_Init`.
- (*6) Il MEB del V775 viene ulteriormente separato dagli elementi precedenti, accedendovi con un data width a 32bit.

2) Esempio strutturato di `vmebus.conf`:

```
# riga commentata
# 1-Board V775 initialization ""
# ""
numboard : 3 definisce il numero di moduli installati
address : 0x00610000 il Base Address del primo modulo è 0x0061
address : 0xcc620000 il Base Address del secondo modulo è 0xcc62
address : 0x11420000 il Base Address del terzo modulo è 0x1142
maxevents : 10 definisce il numero massimo di eventi
```

numboard: segue numero intero decimale che definisce il numero di schede (escluso il master V1718), installate all'interno del backplane VME.

address: segue valore esadecimale, con formato `0XXXXXXXX`, nel quale è definito il "Base Address" del modulo + l'indirizzo iniziale dal quale parte la mappatura dei registri. Nei casi predefiniti impiegati dal nostro sistema, l'indirizzo è `0x00610000` per il V775 e `0xCC620000` per il V792.

maxevent: segue valore decimale, nel quale è inserito il numero degli eventi che il sistema deve acquisire.

3) File di uscita `vmeinfo.log`; nell'esempio sotto riportato, la rappresentazione è limitata ad

una sola scheda (il V775):

Board number:	0
Board Identificative:	0x 7
CAEN V775 ROM memory map:	
ROM Memory OUI MSB value:	0x 0
ROM Memory OUI value:	0x40
ROM Memory OUI LSB value:	0xE6
ROM Memory Version value:	0x13
ROM Memory BOARD ID MSB value:	0x0
ROM Memory BOARD ID value:	0x3
ROM Memory BOARD ID LSB value:	0x7
ROM Memory Revision value:	0x4
ROM Memory Serial MSB value:	0x1
ROM Memory Serial LSB value:	0x5D
CAEN V775 board initialization :	
Data 0x 0 written on Geo Address register at address	0x 611002
Data 0x 0 written on MCST address register at address	0x 611004
Data 0x 0 written on Bit Set 1 register at address	0x 611006
Data 0x 0 written on Bit Clear 1 register at address	0x 611008
Data 0x 0 written on Interrupt Level register at address	0x 61100A
Data 0x 0 written on Interrupt Status register at address	0x 61100C
Data 0x 0 written on Control Register 1 at address	0x 611010
Data 0x 0 written on Address Decoder High register at address	0x 611012
Data 0x 0 written on Address Decoder Low register at address	0x 611014
Data 0x 0 written on Single Shot register at address	0x 611016
Data 0x 0 written on MCST/CBLT Control Register at address	0x 61101A
Data 0x 0 written on Event Trigger at address	0x 611020
Data 0x 0 written on Increment Event at address	0x 611028
Data 0x 0 written on Increment Offset at address	0x 61102A
Data 0x 0 written on Fast Clear Window at address	0x 61102E
Data 0x4804 written on Bit Set 2 at address	0x 611032
Data 0x4004 written on Bit Clear 2 at address	0x 611034
Data 0x 0 written on W Memory Test Address at address	0x 611036
Data 0x 0 written on Test Word High at address	0x 611038
Data 0x 0 written on Test Word Low at address	0x 61103A
Data 0x 0 written on Crate Select at address	0x 61103C
Data 0x 0 written on Test Event Write at address	0x 61103E
Data 0x 804 written on Event Conter Reset at address	0x 611040
Data 0x 64 written on Full Scale Range at address	0x 611060
Data 0x 0 written on R Memory Test Address at address	0x 611064
Data 0x 0 written on SW Comm at address	0x 611068
Data 0x 0 written on Slide Constant at address	0x 61106A
Data 0x 0 written on Threshold 0 at address	0x 611080

Registers map dump

Firmware Revision:	1001
Geo Address:	11100000
MCST/CBLT Address:	10101010
BitSet1:	0
BitClear1:	0
Interrupt Level:	1111000
Interrupt Vector register:	0
Status Register 1:	10011

Control Register 1:	10000011
ADER High:	0
ADER Low:	0
MCST/CBLT Ctrl:	11111100
Event Trigger Register:	11100000
Status Register 2:	1000
Event Counter Low:	0
Event Counter High:	0
Load Test Register:	11111111
FCLR Window:	0
Bit Set 2:	100010000000
Crate Select:	0
Full Scale Range:	1100100
Slide Constant:	0
AAD Register:	10
BAD Register:	10
Thresholds 0:	1000000000
Thresholds 1:	0
....	
...	
...	
Thresholds 31:	1000000000

BIBLIOGRAFIA

- (1) Mod.V1718 VX1718 VME – USB 2.0 Bridge CAEN®Technical Information Manual, Revision n°6, 28/02/2005.
- (2) Mod.V775 series Mod.V775 N series - 32/16 Channel Multievent TDCs.
- (3) CAEN®Technical Information Manual, Revision n°11, 21/02/2005.
- (4) Mod.V792 series Mod.V792 N series - 32/16 CH QDCs CAEN®Technical Information Manual, Revision n°11, 08/03/2005.
- (5) Mod.V820 series Mod.V830 series 32 Channel Latching Scalers CAEN®Technical Information Manual, Revision n°3, 05/09/2002.
- (6) Linguaggio C, Brian W.Kernighan, Dennis M.Ritchie – Editore da Jackson Libri.