
UNIVERSITÀ DEGLI STUDI DEL PIEMONTE ORIENTALE

“AMEDEO AVOGADRO”

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di laurea in Fisica



TESI DI LAUREA

TEST DELL'ELETTRONICA DI FRONT-END, BASATA SU CHIP
CMAD, PER IL RIVELATORE RICH DELL'ESPERIMENTO
COMPASS

Relatore: Prof. Daniele Panzieri

Correlatrice: Dott.ssa Michela Chiosso

Candidata: Serena Panati

ANNO ACCADEMICO 2009-2010

Indice

Sommario	9
1 Introduzione	
all'esperimento COMPASS	11
1.1 Struttura	12
1.1.1 Rivelatori di tracciamento	14
1.1.2 Muon Walls	14
1.1.3 Identificazione di particelle	15
1.2 Il rivelatore RICH	15
1.2.1 L'effetto Cherenkov	15
1.2.2 Conversione dei fotoni	17
1.2.3 Il sistema di readout	19
2 I chip MAD4 e CMAD	25
2.1 Generale	25
2.2 CMAD vs MAD4:	
caratteristiche	26
2.2.1 Guadagno	30
2.2.2 Programmazione del livello	
di baseline e di threshold	33
2.2.3 Disposizione dei canali sulle schede	38
3 Test sui chip e	
relativa analisi statistica	41

3.1	Generale	41
3.2	Sistema di test	42
3.2.1	Procedura di test	44
3.3	Analisi dei dati	47
3.3.1	Problemi rilevati	47
3.3.2	CMAD vs MAD4	56
3.3.3	Schede con problemi maggiori	60
3.4	Creazione di una macro per l'analisi dell'rms > 6 digit	61
3.5	Creazione di una macro per l'analisi dell'rms > 3 digit	64
3.6	Creazione di una macro per l'analisi dell'rms > 2 digit	67
3.7	Analisi del valor medio dell'rms del canale 7	68
3.8	Impostazione della soglia	73
3.8.1	Scelta del fattore numerico moltiplicativo per l'rms	81
3.8.2	Impostazione finale della soglia sui 144 moduli di front-end	90
	Conclusioni	93
4	Appendice	95
.1	Codici	95
.1.1	Codice di histogram_test.C	95
.1.2	Codice di graph_rms3.C	98
.1.3	Codice di lettura_dati.C	103
.1.4	Codice di lettura_dati_2.C	112
.2	File di mapping	115
.2.1	RICH NEW ELECTRONIC CMAD Bottom Saleve	115

.2.2	RICH NEW ELECTRONIC CMAD	
	Bottom Jura	116
.2.3	RICH NEW ELECTRONIC CMAD	
	Top Jura	117
.2.4	RICH NEW ELECTRONIC CMAD	
	Top Saleve	118
.3	Files di calibrazione con rms > 3	119
Bibliografia		123

Sommario

Questa tesi rappresenta il risultato delle analisi riguardanti l'elettronica di front-end montata sul rivelatore RICH dell'esperimento COMPASS del CERN. In particolare le analisi riguardano diversi test su un nuovo chip, il CMAD, in sostituzione al precedente MAD4.

I risultati ottenuti sono da riferirsi a dati raccolti all'inizio del 2009, sia direttamente sull'apparato sperimentale, sia nei laboratori del gruppo di Torino dell'esperimento.

La creazione del CMAD è data dalla necessità di migliorare le prestazioni del sistema di readout dell'esperimento: esso infatti, a differenza del MAD4, presenta caratteristiche più elevate, quali un maggiore rate di eventi, oltre la possibilità di programmare il guadagno e di impostare soglia e baseline.

In particolare durante le analisi si è cercato di evidenziare eventuali malfunzionamenti (soprattutto riguardo gli effetti di rumore elettronico) sia del chip montato in sala sperimentale che in laboratorio. Per effettuare l'acquisizione dei dati e la loro analisi sono state utilizzate macro di Root già preesistenti e si è proceduto alla creazione di nuove.

Attualmente i chip sono montati sul RICH di COMPASS al CERN e sono in funzione.

Capitolo 1

Introduzione

all'esperimento COMPASS

“Where shall I begin, please, your Majesty?”

“Begin at the beginning”, the King said, very gravely

“and go until you come in the end: then, stop.”

L. Carrol - Alice's Adventures in Wonderland

COMPASS è l'acronimo di **CO**mmon **MU**on **P**roton **A**pparatus for **S**tructure and **S**pectroscopy ed è un esperimento a bersaglio fisso che estrae il fascio dall'acceleratore SPS (*Super Proton Synchrotron*) al CERN di Ginevra.

Scopo dell'esperimento è lo studio della struttura dei nucleoni e degli altri adroni; in particolare il contributo di gluoni e quark allo spin dei nucleoni, la spettroscopia adronica, la produzione e lo studio dei mesoni esotici e i mesoni charmati.

1.1 Struttura

Struttura dell'esperimento La struttura dell'esperimento COMPASS è schematizzata nelle figure 1.1 e 1.2.

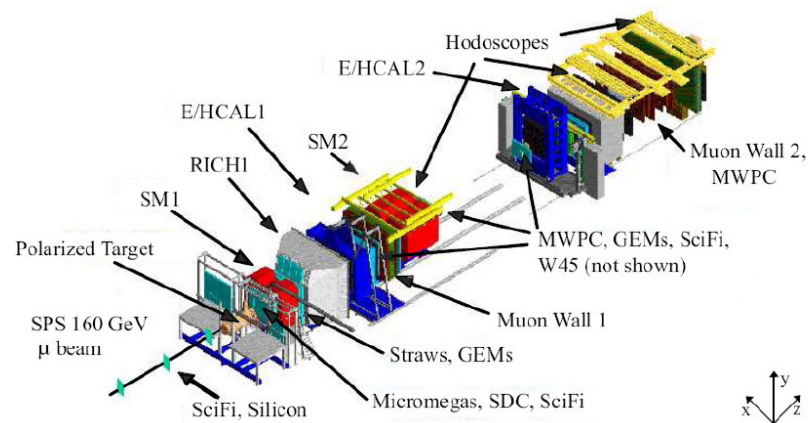


Figura 1.1: COMPASS - visuale.

Come è possibile vedere, l'esperimento consta di un complesso apparato composto da più elementi atti a monitorare i diversi aspetti dell'evento, ovvero degli effetti della collisione delle particelle del fascio sul bersaglio fisso. È caratterizzato da parametri di alta luminosità, da una considerevole capacità di gestire *rate* di dati molto elevati, da un'eccellente capacità di indentificazione delle particelle e da un'ampia accettazione angolare.

Deviazioni a piccolo e grande
angolo

E' possibile considerare l'apparato sperimentale diviso in tre parti: la prima parte riguarda i rivelatori delle particelle del fascio; dopo il bersaglio si hanno due spettrometri: il primo, nell'ordine, si occupa delle deviazioni delle particelle diffuse a grande angolo, mentre il secondo si occupa di quelle deviate a piccolo angolo.

Ciascun spettrometro è costruito attorno ad un magnete analizzatore, preceduto e seguito da rivelatori di tracciamento delle particelle e completato da

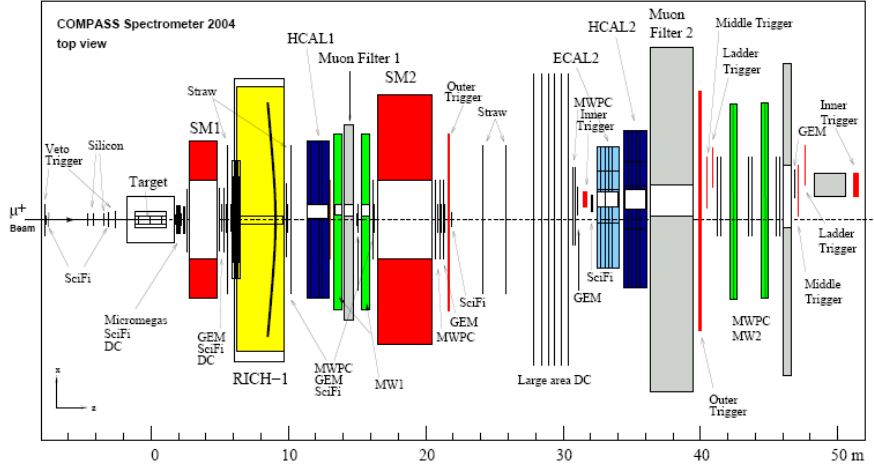


Figura 1.2: COMPASS - spaccato delle parti.

un calorimetro adronico e un filtro muonico per l'identificazione dei muoni ad alte energie.

Per motivi di flessibilità, alcune parti dell'apparato sperimentale sono posizionate su binari per favorirne lo spostamento a differenti distanze dalla collisione; qui si ha il primo calorimetro adronico, il primo filtro muonico, il secondo magnete analizzatore e i rivelatori di tracciamento.

In dettaglio si ha, nella primissima parte dello spettrometro, a monte del bersaglio, il BMS (*Beam Momentum Station*), utile alla misura della quantità di moto, definita su base evento per evento, delle particelle del fascio; questo include tre magneti di dipolo, quattro magneti di quadrupolo, diversi piani di odoscopi scintillatori e rivelatori a fibre scintillanti.

In seguito, nella seconda parte, si ha, come sopra descritto, lo spettrometro a grande angolo, il LAS (*Large Angle Spectrometer*), dotato di un angolo di accettazione di $\pm 180\text{mrad}$ per particelle con impulso di $1\text{ GeV}/c$; costruito attorno al magnete SM1, è preceduto e seguito da rivelatori di tracciamento. SM1 è seguito dal rivelatore RICH, utilizzato per identificare adroni carichi. LAS è

1.1. STRUTTURA

infine completato da un calorimetro elettromagnetico, da un ampio calorimetro
HCAL1 adronico (HCAL1) e da un filtro muonico.

SAS La terza parte riguarda il SAS (*Small Angle Spectrometer*), ossia il rivelatore
per particelle deviate a piccolo angolo (angolo di accettazione di $\pm 30\text{mrad}$) e con
SM2 impulso di 5 GeV/c o superiore. Il suo elemento centrale è il magnete SM2, ma
SAS include inoltre rivelatori di tracciamento, un calorimetro elettromagnetico,
ECAL2 un calorimetro adronico e un filtro muonico. Il calorimetro elettromagnetico
ECAL2 è usato, ad esempio, per rilevare raggi gamma e pioni neutri. Il calo-
rimetro adronico HCAL2, analogamente ad HCAL1, viene utilizzato anche per
la formazione del trigger.

1.1.1 Rivelatori di tracciamento

Allo scopo di ricostruire le traiettorie delle particelle del fascio che attraversa
tutto l'apparato, sono posizionati diversi tipi di rivelatori di tracciamento, gas-
sosi, al silicio o a materiali fortemente scintillanti; nello specifico, i rivelatori di
tracciamento possono essere raggruppati in tre principali tipologie:

- VSAT • VSAT (*Very Small Area Trackers*), i quali, di piccola misura, coniugano
la capacità di gestire grandi flussi con un'eccellente risoluzione spaziale;
- SAT • SAT (*Small Area Trackers*), rivelatori di media grandezza che permettono
di ottenere un'ottima risoluzione spaziale a fronte di una minima quantità
di materiale inserita sul fascio; vengono, in questo frangente, utilizzati
tre MICROMEGAS (*MicroMesh Gaseous Structure*) e undici GEM (*Gas
Electron Multiplier*);
- LAT • LAT (*Large Area Trackers*) che permettono la definizione della traiettorie
a grande angolo e la copertura dell'ampio spazio richiesto per l'accettazione
sperimentale.

1.1.2 Muon Walls

L'identificazione dei muoni diffusi avviene tramite due *Muon Walls* costituiti
MW1 da piani di rivelatori separati da un filtro. Il primo, MW1 (*Muon Wall 1*) è

posizionato all'estremità finale di LAS, davanti a SM2, mentre il secondo, MW2 (*Muon Wall 2*) si trova nell'ultimissima parte di SAS.

MW2

1.1.3 Identificazione di particelle

L'insieme degli apparati LAS e SAS di COMPASS comprende numerosi rivelatori per l'identificazione di particelle. Il RICH, posizionato nel primo spettrometro (RICH-1), identifica gli adroni in pioni, kaoni e protoni, fino ad un valore di impulso al massimo pari a 55 GeV/c. I due calorimetri HCAL1 e HCAL2 misurano l'energia degli adroni e forniscono un equivalente segnale di trigger; il calorimetro elettromagnetico ECAL2 determina le energie dei fotoni e degli elettroni diffusi a piccolo angolo; come già detto in precedenza, in SAS e in LAS i muoni diffusi sono rilevati in due *Muon Walls* (MW1 e MW2).

1.2 Il rivelatore RICH

RICH è l'acronimo di *Ring Imaging Cherenkov*, rivelatore basato sull'effetto Cherenkov (vedi paragrafo 1.2.1), che permette l'identificazione di adroni in un range di energia tra i 5 GeV/c e i 55 GeV/c. Le dimensioni notevoli rispecchiano la necessità di copertura dell'intera accettazione angolare di LAS, e perciò $\pm 250\text{mrad}$ sul piano orizzontale e $\pm 180\text{mrad}$ su quello verticale.

1.2.1 L'effetto Cherenkov

RICH si basa su quello che viene chiamato Effetto Cherenkov. [15] La particella carica che si muove in un mezzo dispersivo con velocità superiore a quella della luce con cui nello stesso mezzo emette onde luminose; è la cosiddetta *radiazione Cherenkov*, dal nome del fisico sovietico che la scoprì nel 1935.

Se l'indice di rifrazione del mezzo è n e la velocità della particella è v , la condizione affinché abbia luogo l'effetto Cherenkov è

$$v > \frac{c}{n} \quad (1.1)$$

1.2. IL RIVELATORE RICH

Indicando con E e $|\mathbf{p}|$ l'energia e il momento della particella (di massa m), e usando:

$$v = \frac{c^2|\mathbf{p}|}{E} = \frac{c\sqrt{E^2 - m^2c^4}}{E}, \quad (1.2)$$

la (1.1) si traduce nella condizione:

$$E > \frac{n}{\sqrt{n^2 - 1}} mc^2 \quad (1.3)$$

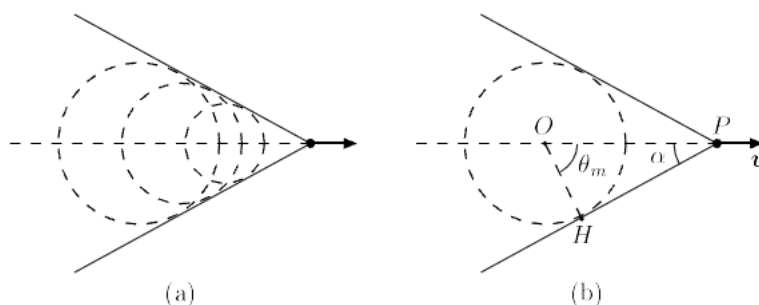


Figura 1.3: L'effetto Cherenkov.

Come si vede in Fig. 1.3 (a), la superficie involuante delle onde emesse è un cono il cui vertice è la posizione della particella (si noti che il mezzo è perturbato all'indietro). Supponiamo che all'istante $t = 0$ la particella si trovi in O ed emetta un fotone (Fig. 1.3 (b)). Al tempo t , essa si sarà spostata in P , con $\overline{OP} = vt$, e il fronte d'onda avrà raggio $(c/n)t$. Dalla geometria del triangolo rettangolo OHP si ricava il valore del semiangolo α al vertice del cono di radiazione,

$$\sin \alpha = \frac{\overline{OH}}{\overline{OP}} = \frac{c}{nv} \quad (1.4)$$

La misura di α permette di determinare la velocità e, nota la massa, l'energia della particella carica che attraversa il mezzo.

1.2.2 Conversione dei fotoni

Il *vessel* (Fig. 1.5), che occupa gran parte del volume di RICH-1, è riempito con il *vessel* perfluoruro di butano (C_4F_{10}), gas radiatore con indice di rifrazione uguale a $n = 1.0015$. I fotoni Cherenkov emessi nel gas vengono riflessi su due specchi sferici. I fotoni sono convertiti in elettroni grazie ai fotocatodi allo Ioduro di Cesio (CsI) delle otto MWPCs (*MultiWire Proportional Chambers*), che amplificano il singolo fotoelettrone e lo rilevano.

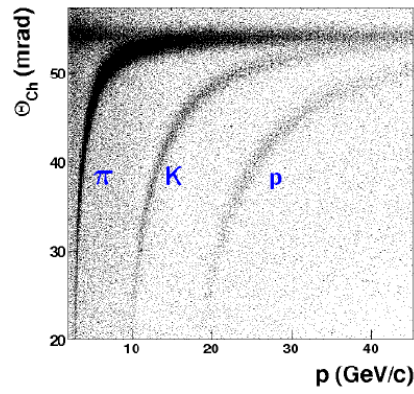


Figura 1.4: Identificazione delle particelle; nello specifico e nell'ordine, da sinistra: pioni (π), kaoni (K), protoni (p).

L'impulso dei muoni oltre il quale non è possibile distinguere due particelle di massa m_1 e m_2 è dato dalla (1.5):

Impulso della particella
incidente

$$p = \frac{1}{\sqrt{n_\sigma}} \sqrt{\frac{(m_2^2 - m_1^2) \sqrt{N_{ph}}}{2 \tan \theta \cdot \sigma_\theta^{ph}}} \quad (1.5)$$

Dove m è la massa della particella, θ è l'angolo Cherenkov, N_{ph} è il numero di fotoelettroni, σ_θ^{ph} è la risoluzione angolare del singolo fotone.

Il rivelatore di fotoni è posizionato lontano dalla linea del fascio e fuori dall'accettanza dello spettrometro. Il corrispondente sistema di specchi consiste in due superfici riflettenti posizionate al di fuori dell'asse del fascio, in modo tale che le immagini degli anelli Cherenkov siano focalizzate fuori dall'accettanza del LAS.

La geometria del RICH-1 si risolve in una superficie di fotorivelazione di Geometria

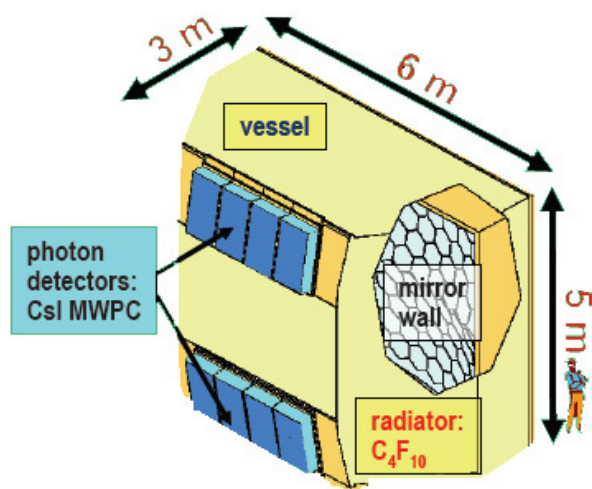
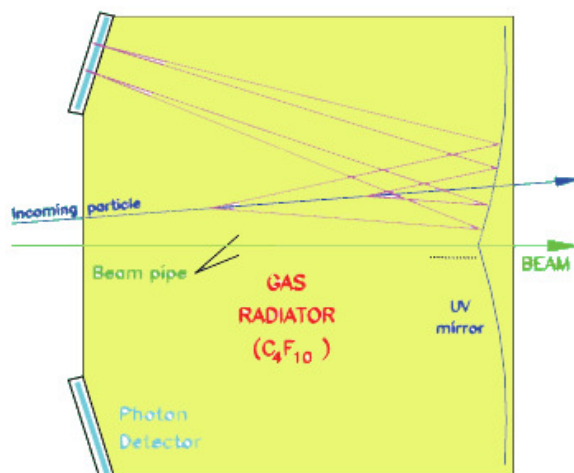


Figura 1.5: Spaccato della struttura del vessel. E' illustrato il cammino del fascio di particelle, la posizione dei vari componenti (quali il rivelatore di fotoni), e le notevoli dimensioni (paragonate alla statura di una persona).

5.6 m². Nella zona perimetrale (che occupa il 75% della superficie attiva) la rivelazione dei fotoni avviene tramite fotocatodo a Ioduro di Cesio di otto camere proporzionali MWPCs (*MultiWire Proportional Chambers*), mentre nella parte centrale (25% della superficie attiva) tramite MAPMT (*Multi Anode PhotoMultiplier Tubes*).

1.2.3 Il sistema di readout

Il sistema di readout nel corso del tempo ha subito alcuni upgrade, che hanno riguardato in particolar modo l'elettronica di lettura dei MAPMTs, in origine basata sul chip preamplificatore-discriminatore MAD4, in seguito sostituito dal più performante CMAD.

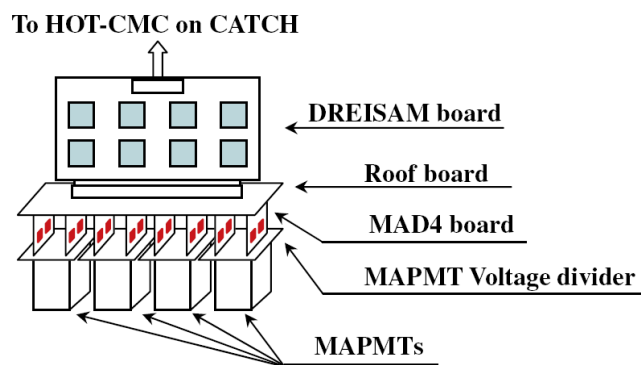


Figura 1.6: Overview del sistema di readout: sono presenti ancora i MAD4, poi sostituiti coi più performanti CMAD.

In totale i 576 MAPMTs che si hanno a disposizione, associati a telescopi con lenti di silice fusa e sensibili ad un ampio intervallo di lunghezza d'onda, posseggono un'ottima risoluzione temporale ($< 1\text{ns}$) e sostengono grandi rate di dati. Tuttavia, nella necessità di coprire grandi superfici, possono risultare parecchio dispendiosi in termini economici.

Ogni scheda di front-end, su cui è montato un chip CMAD, è collegata direttamente, tramite due connettori, al circuito stampato con il circuito partitore

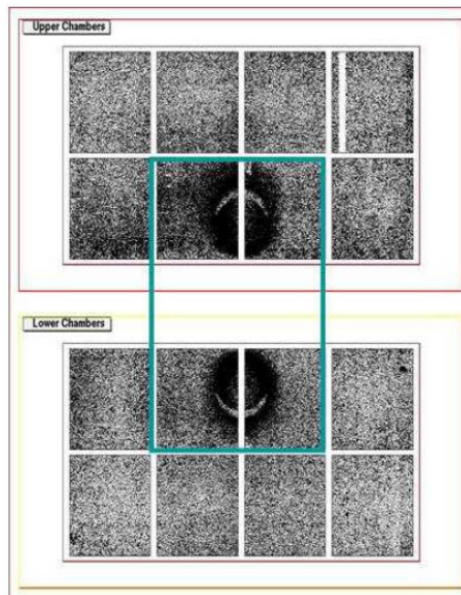


Figura 1.7: Area sensibile del RICH-1. In azzurro sono evidenziate le aree nelle quali sono state rimosse le MWPCs ed installati i MAPMTs.

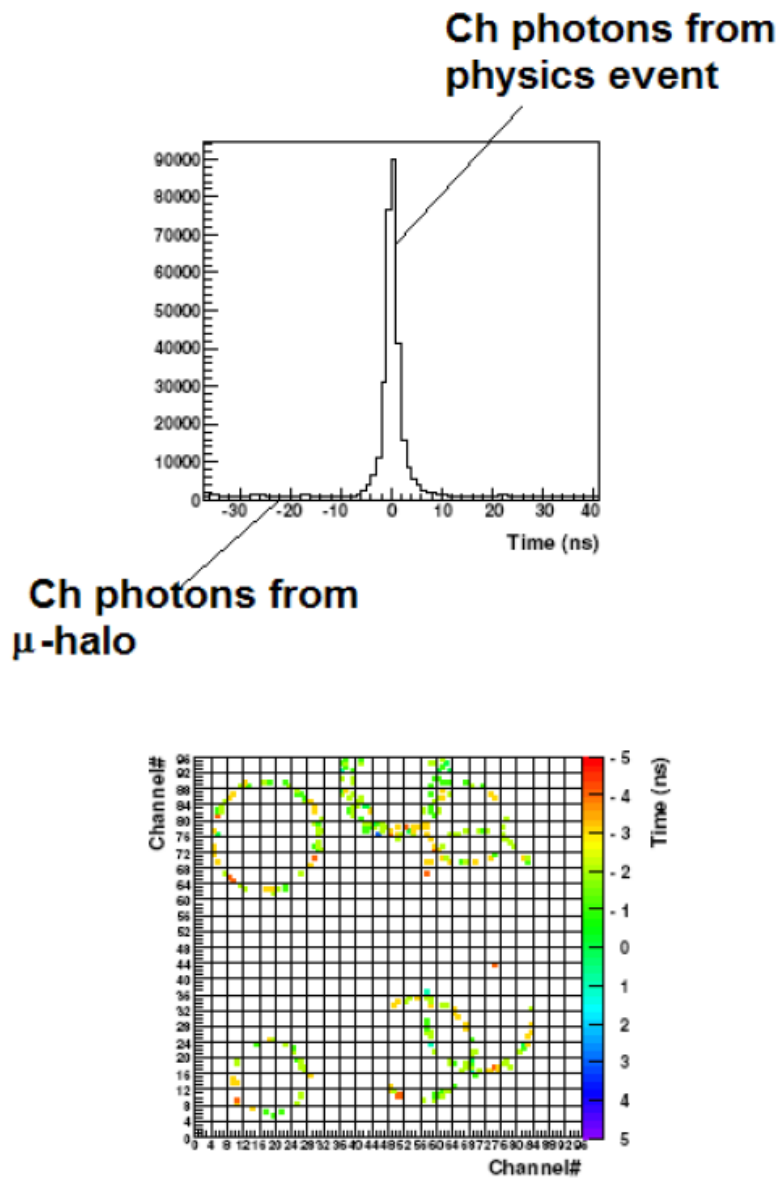


Figura 1.8: In alto: lo spettro temporale del segnale fisico dei fotoni Cherenkov in ambiente sperimentale CERN. In basso: display online di un evento rilevato nel 2006 anelli Cherenkov generati da adroni.

1.2. IL RIVELATORE RICH

di tensione, che serve a distribuire l'alta tensione agli elettrodi del MAPMT.

I segnali di output degli otto chip sono trasferiti, tramite una scheda Roof intermedia, alla scheda digitale di readout, la *Digital RICH ElectronIc SAMpling* DREISAM (DREISAM), basata sul F1 TDC [4].

Scheda Roof La scheda Roof possiede una doppia mansione: quella di fissaggio meccanico delle schede di readout, e quella di distribuire l'alimentazione a bassa tensione (LV, *Low Voltage power*) alle schede CMAD e gestire il trasferimento dei dati di input/output tra schede CMAD e DREISAM.

I dati digitali provenienti dalla DREISAM sono trasferiti mediante fibre ottiche alla scheda di acquisizione, l'HOT-CMC (*HOT-fibre CATCH Mezzanine Card*), montata su modulo VME, che si trova a qualche metro di distanza dal rivelatore.

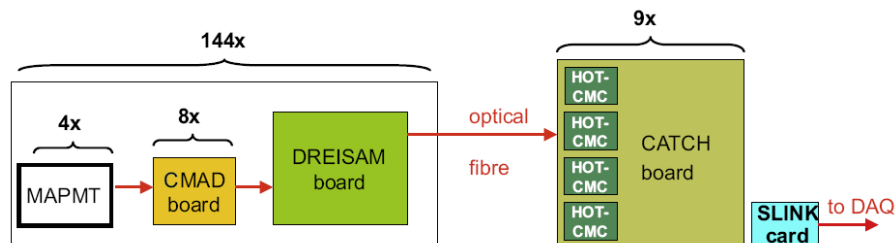


Figura 1.9: Il collegamento al catch.

Da lì, i dati sono trasferiti via S-LINKS al sistema di elaborazione dati dell'esperimento COMPASS.

Il sistema front-end, in definitiva, prevede 144 Roof board dotate, ciascuna, di 8 schede CMAD, per un totale di 9216 canali.

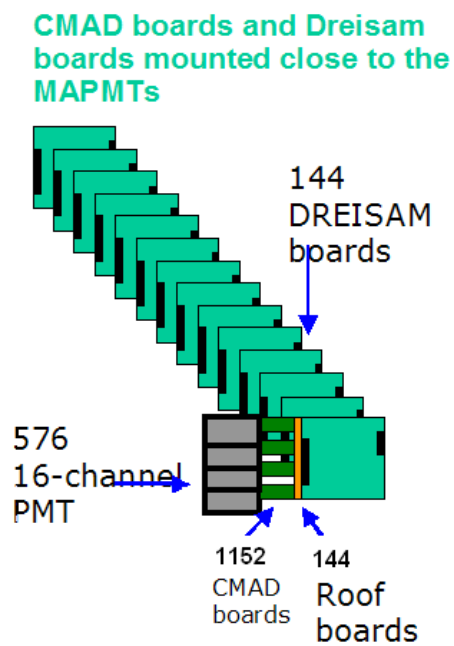


Figura 1.10: Elettronica di readout delle MAPMT.

Capitolo 2

I chip MAD4 e CMAD

Era nuovo. Era singolare. Era semplice. Doveva avere successo!

H. Nelson

2.1 Generale

Allo scopo di migliorare le prestazioni del sistema di readout del RICH si è il CMAD proceduto alla creazione e implementazione di un nuovo chip, chiamato CMAD, in sostituzione del precedente MAD4, secondo le seguenti linee guida:

- conservare la compatibilità col precedente sistema di readout;
- sostenere, a piena efficienza, *rate* di eventi fino a 5 MHz/canale;
- ottimizzare il guadagno (*gain*) per il readout delle MAPMT;
- avere 8 canali per ogni chip;
- rendere possibile la programmazione del guadagno (*gain*) canale per canale;

- avere la possibilità di regolare soglia (*threshold*) e baseline canale per canale.

2.2 CMAD vs MAD4: caratteristiche

Confronto CMAD e MAD4 In Tabella 2.2 sono schematizzate le principali differenze col MAD4:

CMAD	MAD4
8 canali	4 canali
guadagno programmabile canale per canale, da 0.4 mV/fC a 1.2 mV/fC a passi di 0.08; guadagno addizionale 4x	guadagno fissato 3.5 mV/fC
rate sostenuto di eventi > 5MHz/ch;	massimo rate sostenuto di eventi: 1MHz/ch
DAC a 10 bit integrato sul chip	DAC a 8 bit non integrato sul chip
baseline programmabile a 10 bit per ogni singolo canale	baseline non programmabile
soglia a 10 bit programmabile canale per canale	soglia programmabile a 8 bit, comune ogni 4 canali
circuito di bias incorporato nel chip	circuito di bias non incorporato sul chip

In Figura 2.1 è rappresentato uno schema di un canale del CMAD.

Caratterizzazione del chip

Come possiamo vedere dalle tabelle, il CMAD è più flessibile e performante riguardo diversi parametri (guadagno, programmazione baseline ecc.) rispetto al MAD4; in particolare possiamo prendere atto delle migliori osservando alcuni parametri chiave, riassunti nello schema in Figura 2.2, e quindi andandone a osservare alcuni plot relativi ai test (Figure dalla 2.3 alla 2.6).

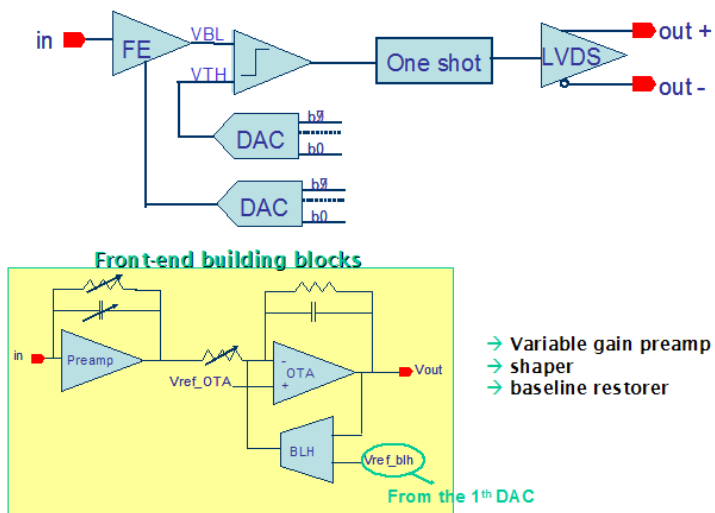


Figura 2.1: CMAD: overview del circuito.

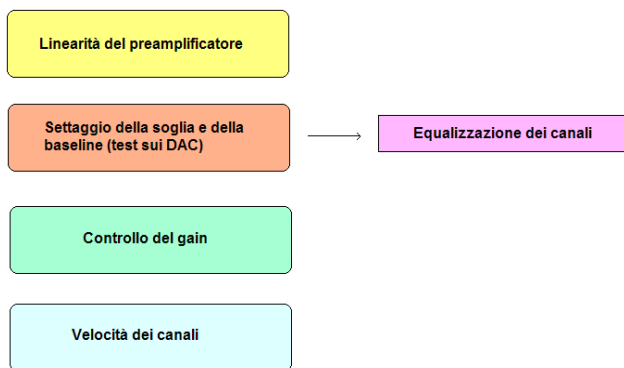


Figura 2.2: Caratterizzazione del chip - generale.

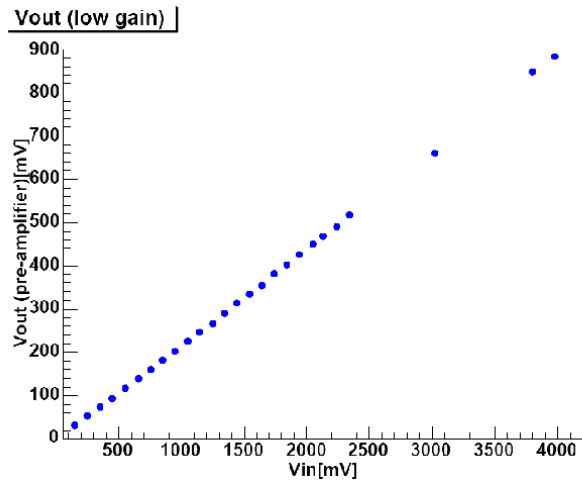


Figura 2.3: Linearità del preamplificatore.

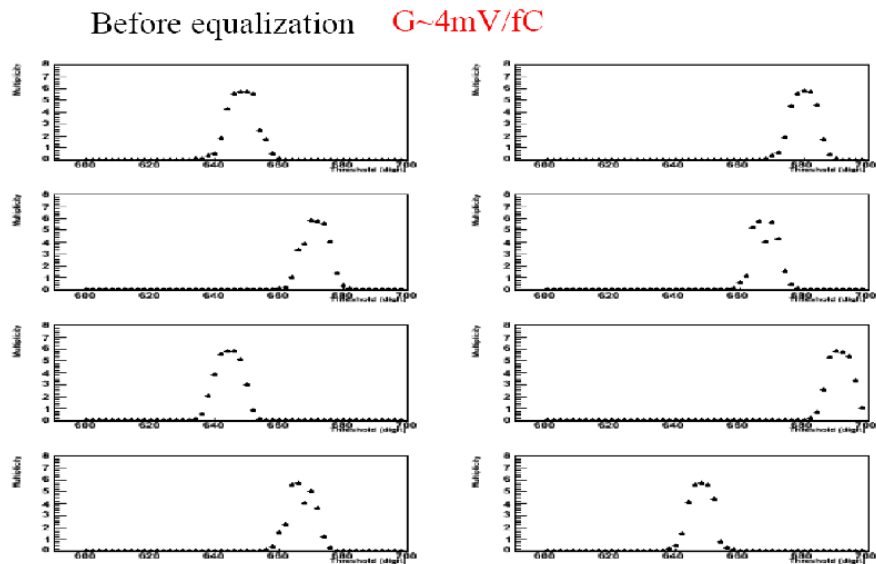


Figura 2.4: Soglia e della baseline prima dell'equalizzazione.

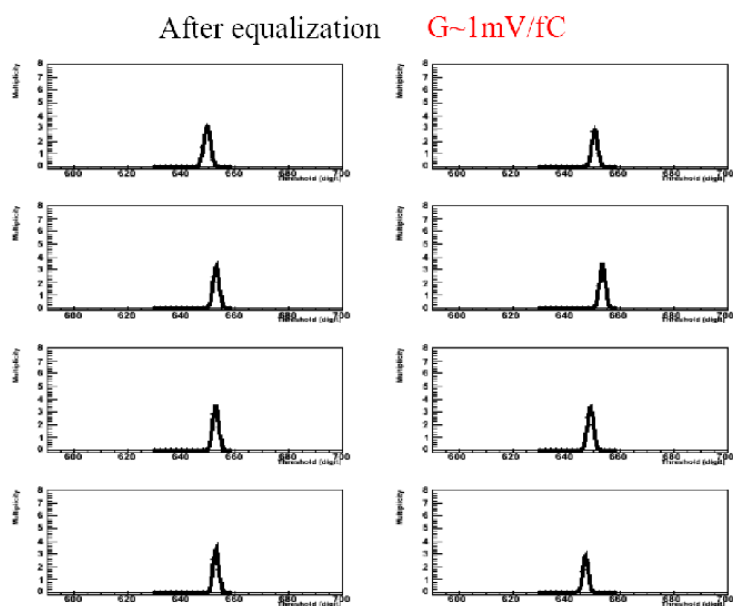


Figura 2.5: Equalizzazione della soglia e della baseline.

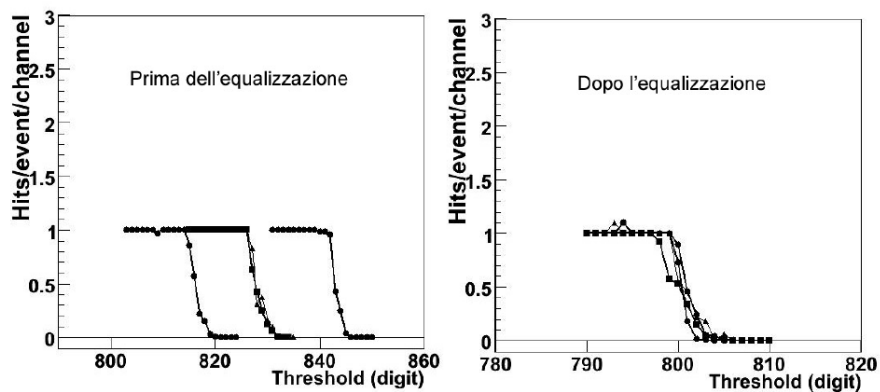


Figura 2.6: Scan di soglia in presenza di un segnale di test, prima e dopo l'equalizzazione. I grafici mostrano quattro canali contemporaneamente.

2.2.1 Guadagno

Guadagno del MAD4 Relativamente al guadagno, il chip MAD4 presenta un valore di ~ 3.5 mV/fC, valore troppo alto per i fotomoltiplicatori, che porta alla saturazione dell'amplificatore; a questo scopo, viene inserito un partitore di tensione per ottenere, in definitiva, un guadagno di circa 1.2 mV/fC.

Guadagno del CMAD Per quanto riguarda invece il CMAD, in questo caso è possibile programmare il guadagno impostando il valore di una resistenza e di una capacità; a questo scopo si hanno a disposizione 4 digit per la stringa di programmazione del resistore e del condensatore, e quindi si hanno $2^4 = 16$ valori possibili che vanno da 0 a 15 suddivisi in due differenti stadi di guadagno, che verranno denominati, rispettivamente, *Gain 1x* e *Gain 4x*. I codici che vanno da 0 a 7 fanno variare il guadagno nella modalità 1x (quindi da 1,2 a 0,4 mV/fC) mentre quelli che vanno da 8 a 15 fanno variare il guadagno nella seconda modalità, la 4x (da 4,8 a 1,6 mV/fC). Per mantenere un'opportuna costante di tempo si sceglie di impostare sempre lo stesso valore in digit per il resistore e per il condensatore. Il valore relativo al minimo guadagno è 0.4 mV/fC, mentre al guadagno massimo è 4.8 mV/fC.

Esaminiamo lo schema illustrativo; nel caso del guadagno *Gain 1x* avremo:

	BIT	GAIN	
Gain 1x	0	1.2 mV/fC	gain MAX per l'1x
	⋮		
	7	0.4 mV/fC	gain MIN per l'1x (guadagno MIN in assoluto)

Nel caso del *Gain 4x* si avrà:

	BIT	GAIN	
Gain 4x	8	$1.2 \cdot 4 \text{ mV/fC}$	gain MAX per l'4x
	⋮		(guadagno MAX in assoluto)
	15	$0.4 \cdot 4 \text{ mV/fC}$	gain MIN per l'4x

Come è possibile vedere dalle tabelle, la programmazione del guadagno avviene secondo una logica invertita, vale a dire che a bit meno significativi corrispondono guadagni maggiori. La risoluzione del guadagno è 0.4 mV/fC .

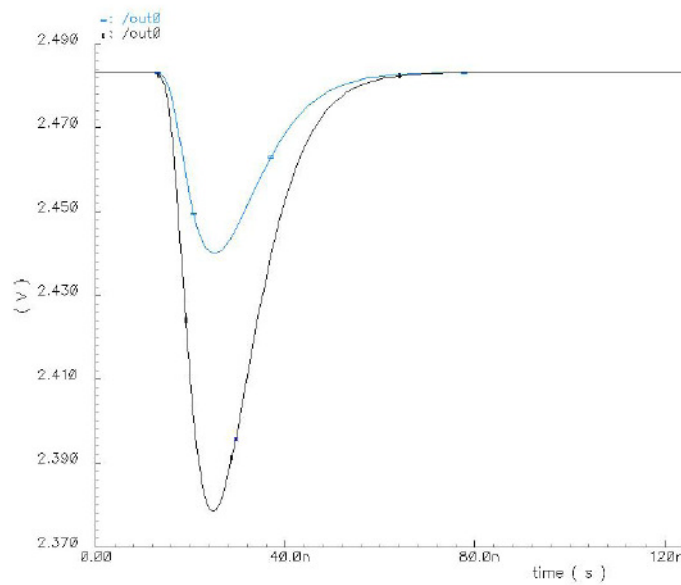


Figura 2.7: Simulazione del segnale in uscita. Confronto tra la modalità di guadagno 1x (in azzurro, in alto) e 4x (in nero, in basso).

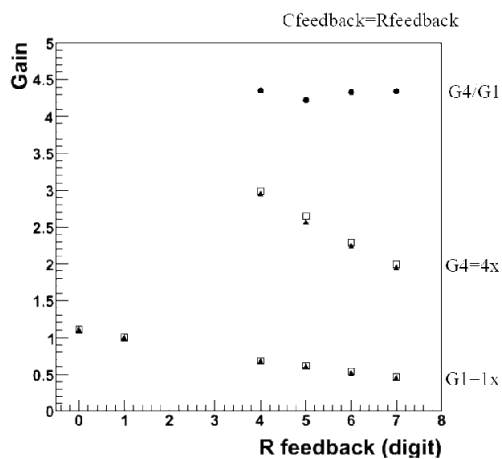


Figura 2.8: Linearità del guadagno 1x e 4x.

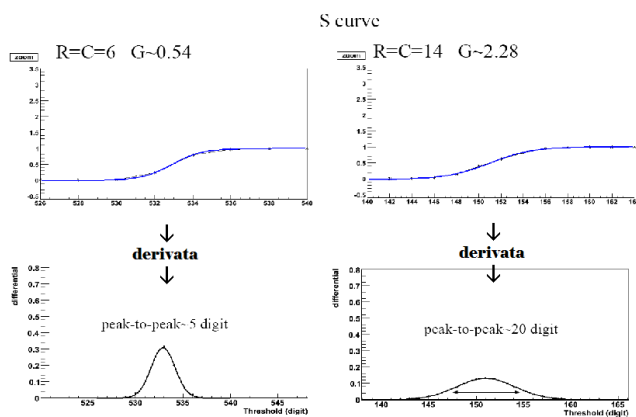


Figura 2.9: In figura possiamo notare la misura del rumore elettronico per due diversi tipi di guadagno. Sulla prima riga, curve ad *esse* relative al segnale di ingresso; in basso, le funzioni derivate delle curve ad *esse*: vediamo che il valore del *peak to peak* (larghezza) è direttamente proporzionale al fattore di guadagno.

2.2.2 Programmazione del livello di baseline e di threshold

Il CMAD è dotato di due DAC a dieci bit per canale, uno dedicato alla programmazione della soglia, l'altro alla programmazione della baseline. La programmazione della baseline permette di equalizzare la posizione del piedistallo canale per canale in modo tale da preservare il range dinamico completo su tutti i canali. Il decoding utilizzato per la programmazione di threshold e baseline è rappresentato nello schema di Figura 2.11. Come possiamo vedere, i tre bit più significativi, D_7 , D_6 , D_5 (si fa riferimento ad un ordinamento di tipo *big endian*¹) vengono utilizzati per comunicare il tipo di operazione da effettuare (programmazione della soglia o della baseline) mentre i cinque successivi (b_9 , b_8 , b_7 , b_6 , b_5 , e in seguito b_4 , b_3 , b_2 , b_1 , b_0) comunicano il valore di soglia o baseline che vogliamo impostare. Il valore comunicato di soglia o baseline occupa, quindi, un totale di 10 bit, il cui ordine è, come sopraccitato, comunicato tramite un byte di codice.

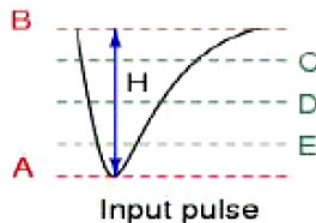


Figura 2.10: In figura è mostrata la forma del segnale a logica negativa, il livello della soglia (A, variabile) e il livello della baseline (B, mantenuto costante).

¹Ricordiamo che esistono due sistemi di ordinamento dei byte: il primo sistema, in cui la numerazione comincia a partire dall'estremo "più grande" (cioè dal byte più significativo) è chiamato *big endian*, in contrapposizione con il sistema *little endian*. Questi termini si devono a Jonathan Swift, che nel suo libro *I viaggi di Gulliver* satireggiava i politici che facevano la guerra per la disputa in cui rompere le uova sode: a partire dalla punta stretta (*little end*) o da quella più larga (*big end*)?. Nel campo dell'architettura degli elaboratori il termine è stato utilizzato per la prima volta in un articolo di Cohen del 1981.[14]

In riferimento alla Figura 2.10 si ricorda che il segnale prodotto dai fototubi è un segnale negativo; nella figura stessa, la linea B rappresenta la baseline mentre la linea A rappresenta il livello della soglia. Definiamo *livello della soglia* il livello di tensione della soglia, mentre per *soglia* si intende la differenza tra il livello della baseline e il livello della soglia. I livelli della soglia e della baseline, possono essere programmati con codici che variano da 0 a 1023 digit (vedi Fig. 2.12). Supponendo di mantenere costante il livello della baseline, ne consegue che il valore minimo di soglia (0 digit) rappresenta il valore massimo possibile di soglia. Il MAD4, invece, possiede un range dinamico della soglia di 256 digit (anzichè 1024)², ovvero da 0 a 255, oltre che un gain fissato a 3.5 mV/fC, baseline e threshold non programmabili.

Opcode

D ₇	D ₆	D ₅	Operation
0	0	0	D ₄ D ₃ D ₂ D ₁ D ₀ → b ₉ b ₈ b ₇ b ₆ b ₅ threshold DAC
0	1	0	D ₄ D ₃ D ₂ D ₁ D ₀ → b ₄ b ₃ b ₂ b ₁ b ₀ threshold DAC
0	0	1	D ₄ D ₃ D ₂ D ₁ D ₀ → b ₉ b ₈ b ₇ b ₆ b ₅ baseline DAC
0	1	1	D ₄ D ₃ D ₂ D ₁ D ₀ → b ₄ b ₃ b ₂ b ₁ b ₀ baseline DAC
1	0	0	D ₃ D ₂ D ₁ D ₀ → CAP control
1	1	0	D ₃ D ₂ D ₁ D ₀ → RES control
1	-	1	Invalid Opcode

Figura 2.11: Comunicazione a 10 bit.

Differenze col MAD4

Questo porta alla conseguenza che, mentre per il MAD4 non è possibile fare alcunché per modificare il livello di default della baseline, nel CMAD è possibile equalizzare la posizione del piedistallo in modo da farla coincidere con un determinato valore in digit, per poter sfruttare, per tutti i canali, il range dinamico desiderato.

²Si ricorda che un digit corrisponde, nel nostro caso, a 0.5 fC.

Nel nostro caso si è scelto di equalizzare la baseline al valore di 650 digit, non necessitando, per l'applicazione con i MAPMT, di aver a disposizione l'intero range dinamico.

Come si vede in Figura 2.12, la programmazione dei DAC nel CMAD avviene in logica negativa, dunque un alto valore di soglia in digit corrisponde ad una soglia bassa, e viceversa.

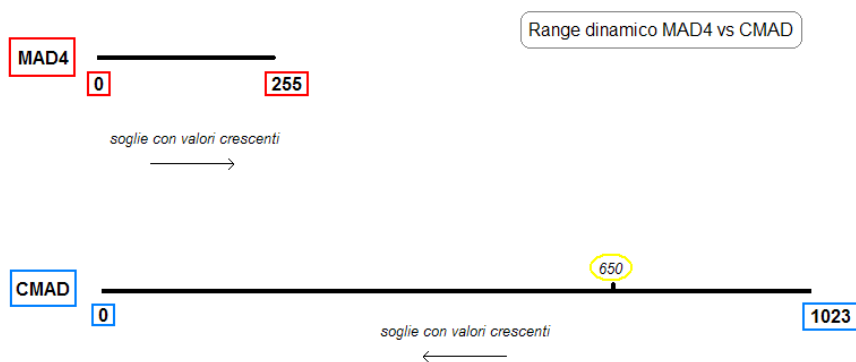


Figura 2.12: Range dinamici per il MAD4 e per il CMAD.

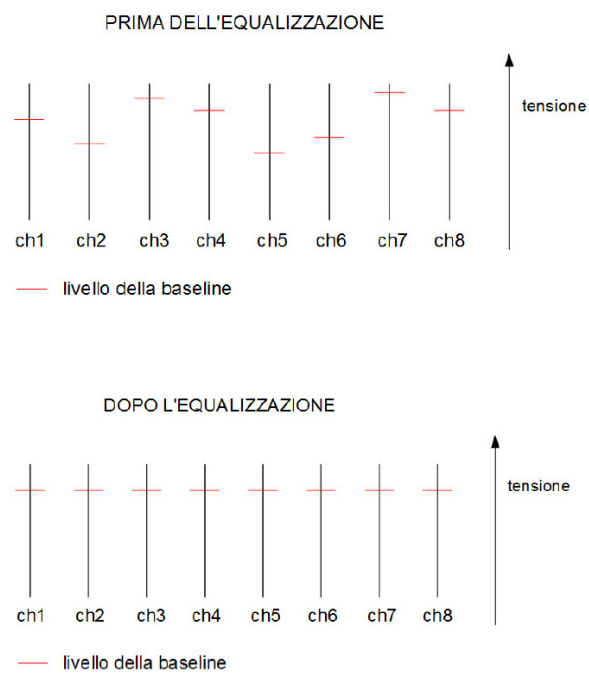


Figura 2.13: La figura mostra come si uniformino i livelli di tensione di *baseline* (pedistallo) dopo la procedura di equalizzazione (vedi anche Figure 2.4 e 2.5).

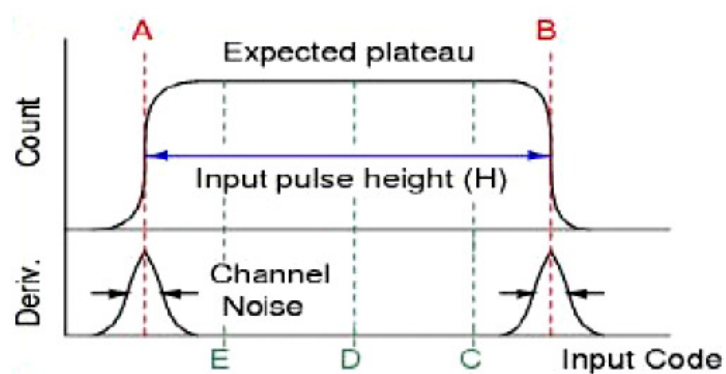


Figura 2.14: Parte superiore: scan di soglia, livello della soglia (in digit) sull'asse x e misura di molteplicità sull'asse y . Parte inferiore: derivata del grafico dello scan di soglia. Sino a quando il livello della soglia si trova al di sotto la linea A, la soglia è maggiore del segnale H, il comparatore non commuta e l'uscita è nulla. Quando il livello della soglia oltrepassa il livello del segnale, la soglia è minore del segnale stesso e il comparatore inizia a commutare. La commutazione continua fino a quando il livello della soglia non raggiunge la baseline B. L'uscita torna dunque ad essere nulla fino al termine dello scan di soglia.

2.2.3 Disposizione dei canali sulle schede

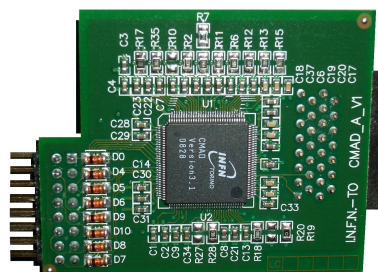


Figura 2.15: Scheda A con CMAD implementato.

I CMAD sono ospitati su schedine simmetricamente disposte su due file, uguali e speculari rispetto alle file stesse, collegate a gruppi di otto su schede Schede A e B Roof tramite connettori. Per distinguerle, chiameremo B quelle presenti sulla fila presso la quale vi è il connettore per l'alimentazione a 7 Volt (verrà preso questo come punto di riferimento) e A quelle presenti sull'altra fila speculare alla prima. Poniamo attenzione al fatto che i chip CMAD su entrambe le tipologie di schede sono *identici*: ciò che differisce nei due casi è la scheda (A o B) ospitante: le schede A e B sono identiche tra loro ma speculari rispetto all'asse passante longitudinalmente per il centro della Roof board; quindi le stesse differiscono anche nella connessione alla stessa Roof, e di conseguenza nella lettura dei canali. Come vedremo meglio nel Capitolo 3.3 la lettura avverrà in modo speculare, secondo gli schemi seguenti (i quali saranno ripresi, in seguito, nella lettura del segnale relativo ad ogni singolo canale):

Schede A (pari)

4	0
5	1
6	2
7	3

Schede B (dispari)

11	15
10	14
9	13
8	12

Schema della disposizione del canali sulle schede.

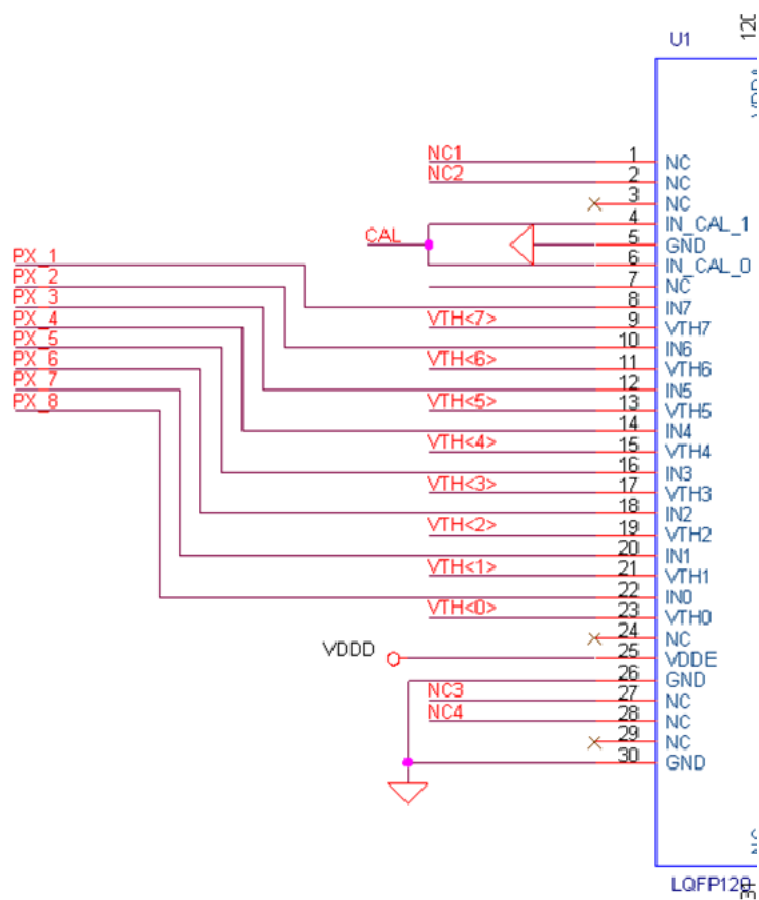


Figura 2.16: Schema input scheda A

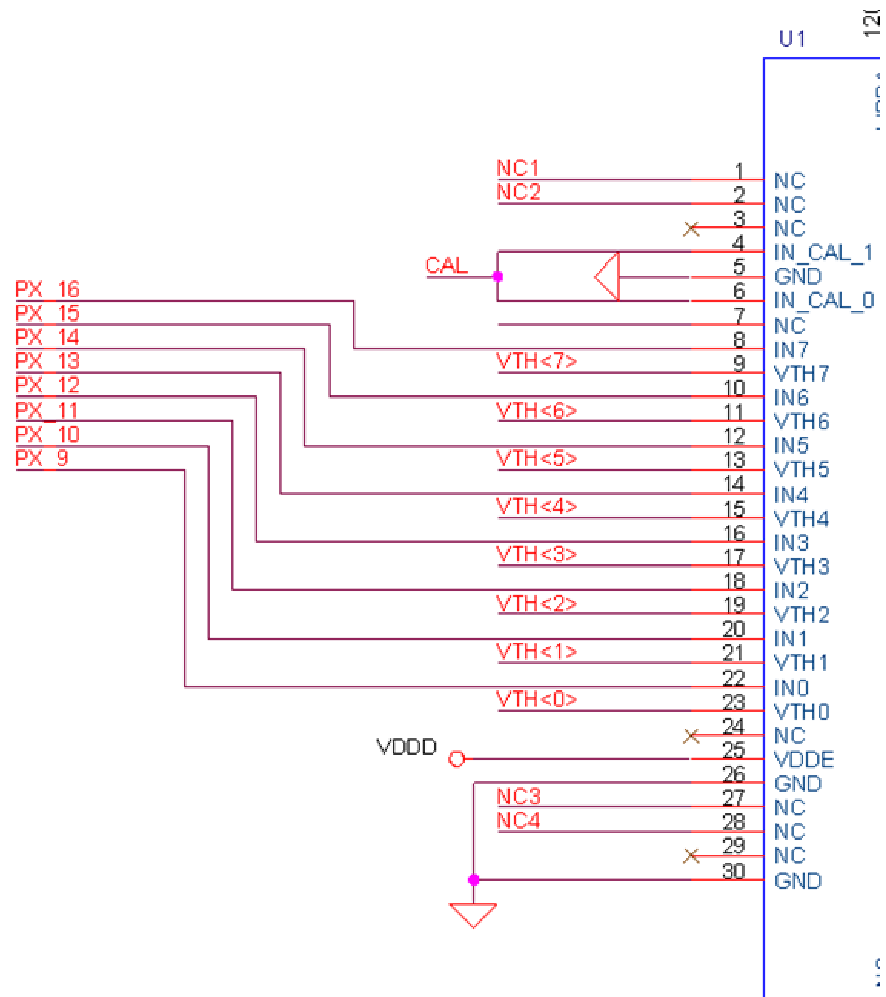


Figura 2.17: Schema input scheda B

Capitolo 3

Test sui chip e relativa analisi statistica

Il testing non può mai rivelare l'assenza di bug.

E. W. Dijkstra

3.1 Generale

Al fine di verificare il buon funzionamento dei chip CMAD montati sulle schede di front-end del RICH di COMPASS, si è proceduto all'utilizzo di programmi (scritti in linguaggio C++) già preesistenti e alla creazione di nuove macro di Root¹ (i codici delle quali sono presenti in Appendice .1), al fine evidenziare gli eventuali problemi presenti nelle schede, individuarne la frequenza e possibilmente la causa, e infine farne un'analisi statistica che determini l'andamento degli eventi. Le macro analizzano dati di test effettuati sia nei laboratori della sezione torinese dell'INFN, sia direttamente sull'esperimento COMPASS al CERN. I dati di entrambi gli ambienti sperimentali sono stati confrontati al

¹Root è un programma orientato ad oggetti distribuito dal CERN.

Provenienza dei dati:
laboratorio e ambiente
sperimentale

3.2. SISTEMA DI TEST

fine di rintracciare anomalie o differenze, e quindi portare al miglioramento le prestazioni dei chip presi in considerazione. Il lavoro è stato effettuato sostanzialmente su due macchine in ambiente Unix, una cpu VME (compass@tozvxl) e un pc (compass@cmps13xl).

3.2 Sistema di test

Apparato sperimentale Si ha a disposizione l'apparato sperimentale di elettronica necessario, ovvero l'architettura che comprende 8 CMAD + 1 Roof + 1 DREISAM collegate tramite fibra ottica all'HOT-CMC connesso al CATCH disposto nel crate VME, un alimentatore a bassa tensione e una cpu VME (compass@tozvxl).

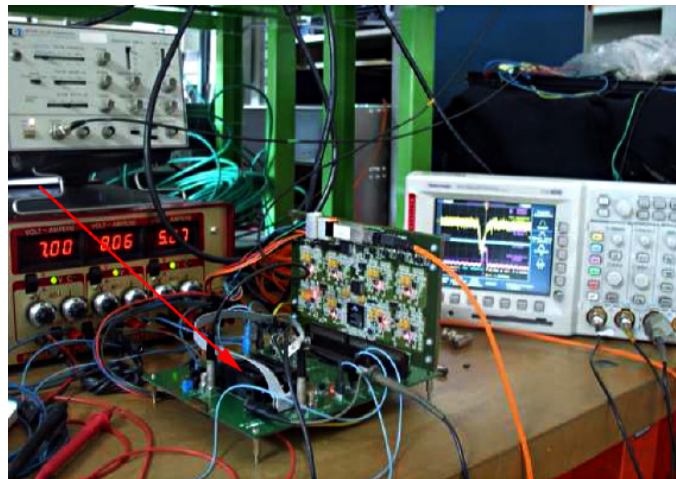


Figura 3.1: Apparato sperimentale nei laboratori di INFN (Torino).

Si è proceduto per fasi. Innanzitutto è necessario collegare la Roof completa di otto schede CMAD alla DREISAM, a sua volta collegata tramite fibra ottica (grazie alla quale avverrà il trasferimento dati) alla scheda HOT-CMC. Sistemato tutto ciò, è importante controllare, sullo schermo dell'alimentatore, che la corrente dissipata sia di circa 0.67 A, onde evitare corto circuito con conseguenti danni ai chip.

Inizio della fase di test A questo punto si può iniziare la fase di test. Essa avviene tramite l'ese-

cuzione di due programmi (sulla cpu VME compass@tozvxl), da terminale; il primo:

```
./test_console.s
```

e il secondo:

```
konsole -e./test_gui.s&
```

All'avvio del primo compare sulla finestra della console il seguente menu:

Main menu:

- 0: Exit this application
- 1: Configure ALL Catch FPGAs
- 2: Init TCS dummy boards
- 3: Print Catches status information
- 4: Read front-end mapping information
- 5: Reset front-end boards
- 6: Load front-end boards settings
- 7: Set common DAC
- 8: Set DAC channel
- 9: Set common threshold (fC)
- 10: Set pulse amplitude (digits)
- 11: Set pulse amplitude for DAC(digits)
- 12: Clear FIFO
- 13: Dump FIFO contents
- 14: Send sync/reset signal
- 15: Write patten to serial line
- 16: Toggle debug mode on/off [now off]
- 17: Print event information for one triggers
- 18: Calibrate one triplet
- 19: Calibrate all boards
- 20: Check sync/reset signal
- 21: Check front-end channels
- 22: Print geographical address

3.2. SISTEMA DI TEST

- 23: Set trigger parameters (latency & window)
- 24: Default setup

Enter choice: _

Inizializzazione del chip F1 è necessario quindi premere il tasto 6² e successivamente invio per inizializzare il chip F1 sulla DREISAM. Non appena compiuta questa azione, infatti, sulla DREISAM si accenderanno i led che indicano, appunto, l'avvenuta inizializzazione. È possibile quindi passare al secondo programma; si vede comparire un menu di comandi con, nell'ordine:

threshold scan I
threshold scan II
analysis
thr scan ExtPulse
exit

3.2.1 Procedura di test

Primo scan: threshold scan I Cliccando su:

threshold scan I

(questa operazione ci permette di fare uno scan di soglia per ogni canale nell'intervallo tra 630 e 658 digit prima dell'equalizzazione della baseline, che si farà al passo successivo) comparirà la richiesta del numero della scheda che si andrà a considerare:

enter board number:_

Una volta avviata l'acquisizione dati verrà salvato un file per ogni soglia con il valore di molteplicità (numero di hit per evento) per ogni canale.

²Si rammenta che questa operazione è da effettuare prima di ogni test.

Terminata la sessione, cliccando su:

`analysis`

nel menu sovracitato, avverrà la richiesta del nome della cartella in cui fare l'analisi (che sarà la stessa, ovviamente, in cui saranno stati salvati i dati non ancora equalizzati):

Enter directory name (Es: Analog_001; Analog_001/Gain_0):

basterà scrivere il percorso e premere invio.

Verrà inoltre richiesta la threshold minima e massima del nostro intervallo:

Enter minimum threshold value: _

Enter maximum threshold value: _

A questo punto si procede con l'equalizzazione delle soglie, utilizzando l'opzione: Secondo scan:

`threshold scan II`

threshold scan II

immediatamente dopo la quale verrà richiesto il valore del gain:

`enter gain:`

in cui si andrà ad inserire il valore di gain 0 (~ 1.2 mV/fC) o gain 8 (~ 4.8 mV/fC). Nel nostro caso si è usato il gain massimo in configurazione 1x e 4x.

Una volta terminata la sessione, premendo nuovamente:

`analysis`

si procede alla seconda analisi dei dati equalizzati.

Questo tipo di lavoro fornirà dei grafici (vedi Figura 3.2) e istogrammi (vedi Figura 3.3) che mostrano il valore medio (*mean*) e l'ampiezza a metà altezza (*rms*) delle curve gaussiane ottenute con gli scan di soglia che ci forniscono rispettivamente la posizione del piedistallo e il valore di rumore per ogni canale dopo l'equalizzazione, utili per le analisi successive.

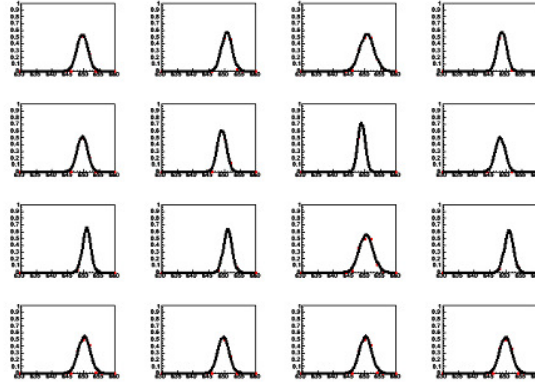
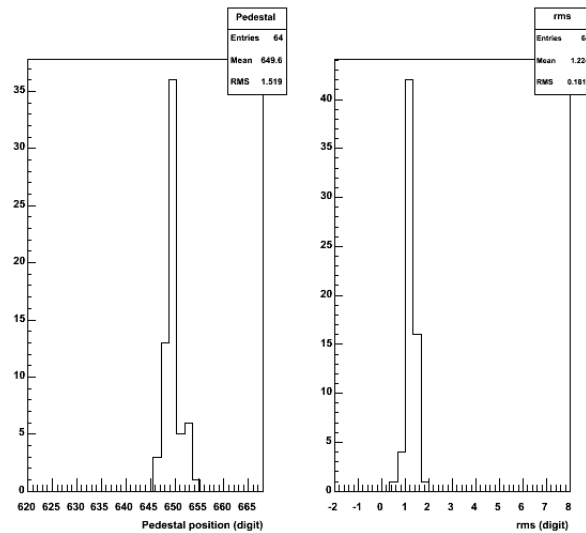


Figura 3.2: Curve di soglia dopo l'equalizzazione.

Figura 3.3: A sinistra, la distribuzione delle posizioni dei piedistalli; a destra, la distribuzione dei valori dell' rms .

3.3 Analisi dei dati

La nuova elettronica di front-end basata sul chip CMAD è stata montata sul RICH di COMPASS all'inizio del 2009. Per verificarne il corretto funzionamento abbiamo analizzato dati presi in ambiente sperimentale prima dell'inizio del *run* deol 2009. Per effettuare l'analisi si è utilizzata una macro di Root che permette di osservare in un'unica finestra una matrice di 64 plot, corrispondenti ai 64 canali di un modulo di front-end completo (8 CMAD + 1 Roof Board + 1 DREISAM). In particolare, per facilitare la lettura, è stato predisposto che nel grafico venisse evidenziata la prima colonna in azzurro, colonna corrispondente al canale 0 (vedi Figura 3.5). Lo scopo è quello di osservare che ciascun canale mostri una forma della curva simile ad una gaussiana, senza code, segnali interrotti o effetti di *crosstalk*.

Crosstalk

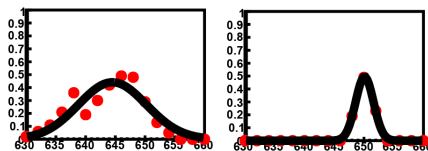


Figura 3.4: Un buon segnale (destra) e uno con effetto di *crosstalk* a sinistra.

3.3.1 Problemi rilevati

Da quest'analisi, è stato elaborato il seguente elenco di schede problematiche: Schede problematiche



Figura 3.5: Plot d'empio.

CATCH	PORT	PLOT	ROOF	PROBLEM
501	8	8	125	<ul style="list-style-type: none"> • missing channel (known problem)
501	9	9	146	<ul style="list-style-type: none"> • strange shape - on 1 channel 52 (double pick?) → not present in test
503	9	42	7	<p>slightly larger crosstalk</p> <ul style="list-style-type: none"> - on channels 0 → not present in test <p>double pick on channel 4 (CMAD board 6)</p>
504	7	55	5	<ul style="list-style-type: none"> • double pick - on channel 0 (CMAD board 2) → not present in test
505	0	64	3	<ul style="list-style-type: none"> • double pick - on channel 0 (CMAD board 0) → not present in test - on channel 0 (CMAD board 2) → not present in test - on channel 0 (CMAD board 4) → not present in test
506	3	82	90	<ul style="list-style-type: none"> • (very) large crosstalk - on channel 0 (CMAD board 0) → still present in test, (strange shape, not very large) - on channel 4 (CMAD board 0) → not present in test

3.3. ANALISI DEI DATI

CATCH	PORT	PLOT	ROOF	PROBLEM
506	12	92	82	<ul style="list-style-type: none"> • large shape - on channel 0 (CMAD board 0),→ still present in test, (strange shape, not very large) - on channel 0 (CMAD board 2), → not present in test - on channel 0 (CMAD board 4),→ not present in test - on channel 0 (CMAD board 6),(double pick?)→ still present in test, (strange shape, not very large) • large shape - on channel 4 (CMAD board 0), → not present in test - on channel 4 (CMAD board 2), → not present in test - on channel 4 (CMAD board 4), → not present in test - on channel 4 (CMAD board 6), → not present in test

CATCH	PORT	PLOT	ROOF	PROBLEM
506	13	93	93	<ul style="list-style-type: none"> • large shape - on channel 0 (CMAD board 0), → still present in test (not large) - on channel 0 (CMAD board 2), → not present in test - on channel 0 (CMAD board 4), → not present in test - on channel 0 (CMAD board 6,) → still present in test (not large) • large shape - on channel 4 (CMAD board 0), → not present in test - on channel 4 (CMAD board 2), → not present in test - on channel 4 (CMAD board 4), → not present in test

3.3. ANALISI DEI DATI

CATCH	PORT	PLOT	ROOF	PROBLEM
507	8	104	87	<ul style="list-style-type: none"> • a very large crosstalk - on channel 0 (CMAD board 0), → still present in test (not large) • large shape - on channel 0 (CMAD board 2), → still present in test (not large) - on channel 0 (CMAD board 6), → still present in test (not large) • a very large crosstalk - on channel 4 (CMAD board 0), → not present in test • large shape - on channel 4 (CMAD board 2), → not present in test - on channel 4 (CMAD board 4), → not present in test - on channel 4 (CMAD board 6), → not present in test
507	9	105	79	<ul style="list-style-type: none"> • large shape - on channel 0 (CMAD board 0), →still present in test (not large) ★ NB: in test on channel 0 (board 4) and on channel 0 (CMAD board 6) you an see a strange effect, like crosstalk (but not very large...luckly)

CATCH	PORT	PLOT	ROOF	PROBLEM
507	10	106	80	<ul style="list-style-type: none"> • large crosstalk <ul style="list-style-type: none"> - on channel 0 (CMAD board 2), → not present in test - on channel 0 (CMAD board 6), → still present in test (like a crosstalk) • large shape <ul style="list-style-type: none"> - on channel 0 (CMAD board 0), → still present in test (like a large shape) - on channel 0 (CMAD board 4), → not present in test • strange crosstalk <ul style="list-style-type: none"> - on channel 4 (CMAD board 2), → not present in test - on channel 4 (CMAD board 6), → not present in test • large shape <ul style="list-style-type: none"> - on channel 4 (CMAD board 0), → not present in test - on channel 4 (CMAD board 4), → not present in test ★ NB: in test on channel 1, channel 2, channel 3 (CMAD board 6), there's a strange shape (like crosstalk)

3.3. ANALISI DEI DATI

CATCH	PORT	PLOT	ROOF	PROBLEM
508	14	126	46	<ul style="list-style-type: none"> • strange shape on - channel 0 (CMAD board 3), → not present in test
508	15	127	38	<ul style="list-style-type: none"> • very very large shape - on channel 0 (CMAD board 0), → not present in test - on channel 0 (CMAD board 4), → not present in test
509	2	130	116	<ul style="list-style-type: none"> • large crosstalk - on channel 0 (CMAD board 0) → not present in test - on channel 0 (CMAD board 4) → not present in test • large crosstalk - on channel 4 (CMAD board 4) → not present in test

CATCH	PORT	PLOT	ROOF	PROBLEM
509	3	131	113	<ul style="list-style-type: none"> • large crosstalk - on channel 0 (CMAD board 0) → not present in test - on channel 0 (CMAD board 4) → not present in test • large shape - on channel 0 (CMAD board 2) → not present in test - on channel 0 (CMAD board 6) → not present in test • strange and large shape - on channel 4 (CMAD board 0) → not present in test - on channel 4 (CMAD board 2) → not present in test - on channel 4 (CMAD board 4) → not present in test - on channel 4 (CMAD board 6) → not present in test

Come già si può intuire dai valori presenti in tabella, i canali che forniscono maggiori problemi sono per lo più i canali 0 e 4 delle schede pari, ovvero le A.

3.3.2 CMAD vs MAD4

A questo punto ci si è occupati di effettuare un confronto tra gli scan di soglia dei CMAD e dei precedenti MAD4 per valutare le differenti performance in ambiente sperimentale COMPASS. La nostra macro di Root ci fornisce, per ogni plot, due curve fittate sovrapposte: quella rossa è relativa al MAD4, quella in nero al CMAD:

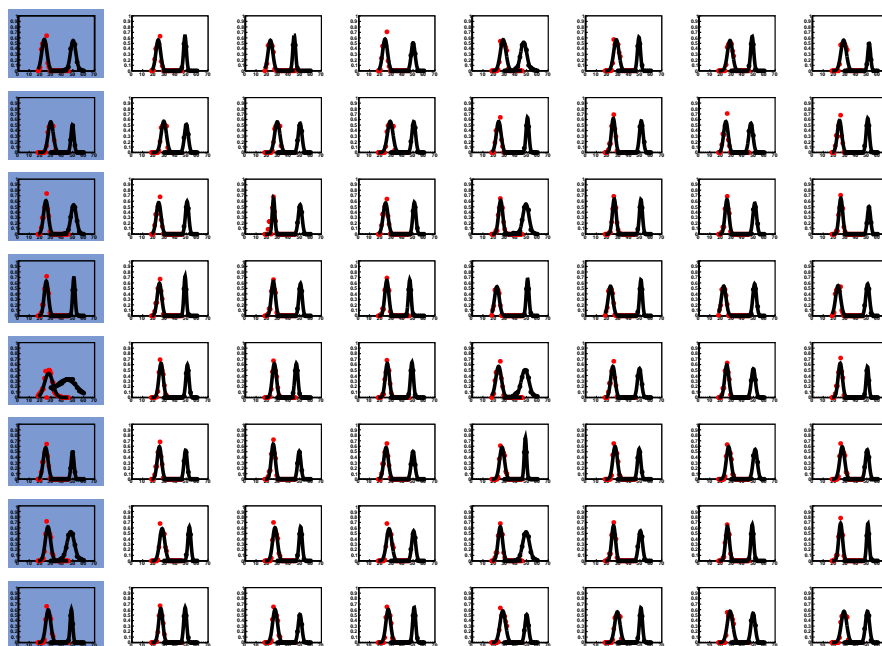


Figura 3.6: Plot CMAD vs MAD4.

Ovviamente, questa volta, all'interno della macro si dovranno prendere in considerazione i diversi range dinamici dei due chip; nello specifico:

threshold	MAD4	CMAD
MIN →	18	630
MAX →	50	660

Infatti, ricordiamo che il range dinamico del MAD4 va da 0 a 255 digit, con un'impostazione della baseline, di default, intorno ai 30 digit; mentre per il CMAD si ha un range da 0 a 1023 e si è deciso di programmare la posizione della baseline a 650 (circa a metà): da qui i valori di soglia, che andranno a

prendere in considerazione l'intervallo contenente il valore della baseline. Per una miglior visualizzazione dei grafici l'uno accanto all'altro, è stato sottratto 600 alla soglia del CMAD. Inoltre si vede anche dai grafici che gli scan per il MAD4 sono effettuati digit per digit, mentre quelli per il CMAD ogni due digit (per queste motivazioni la curva relativa al MAD4 sarà composta da un numero doppio di punti), e inoltre sono in logica invertita.

Logica invertita

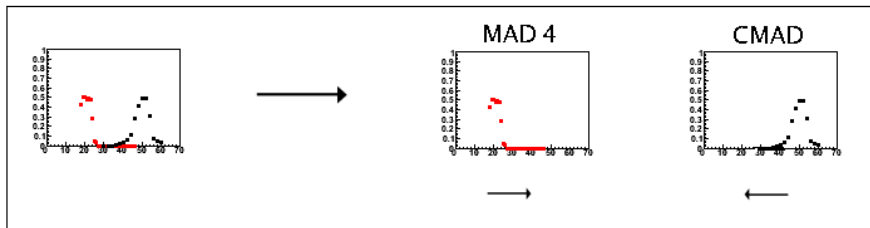


Figura 3.7: Logica invertita dei plot dei CMAD: possiamo vedere, separatamente, il plot del CMAD e del MAD4.

Tramite la creazione di una apposita macro, sono stati equiparati i dati di tutti i 144 moduli di front-end a 64 canali montati sul RICH di COMPASS, provenienti sia dai test sui vecchi chip MAD4, sia su quelli sui CMAD. Di seguito sono riportati grafici con in ascissa il numero del modulo di front-end (da 0 a 143) e in ordinata il relativo valore di *rms* (si è preso in considerazione il valore medio di tutti i canali per ogni grafico).

I simboli quadrati si riferiscono alle schede A, quelli triangolari alle B; inoltre quelli di colore rosso si riferiscono al MAD4, quelli in nero al CMAD. Vediamo immediatamente che i valori di *rms* per le CMAD A sono in generale i più elevati, mentre quelli col valore meno elevato sono i CMAD B.

3.3. ANALISI DEI DATI

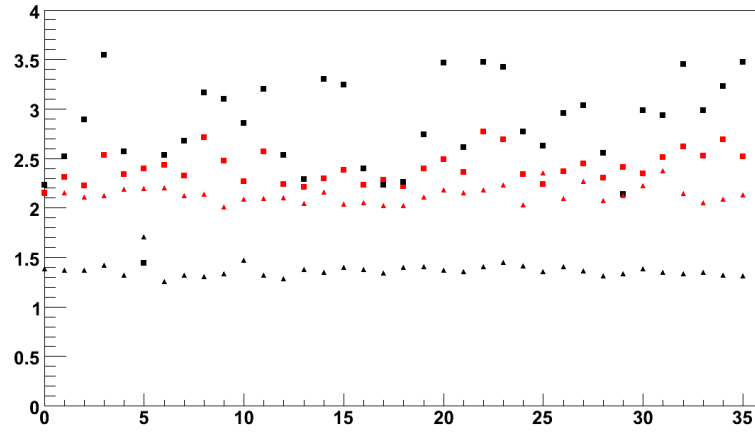


Figura 3.8: Valore medio su 64 canali di *rms* per i moduli di front-end da 0 a 35.

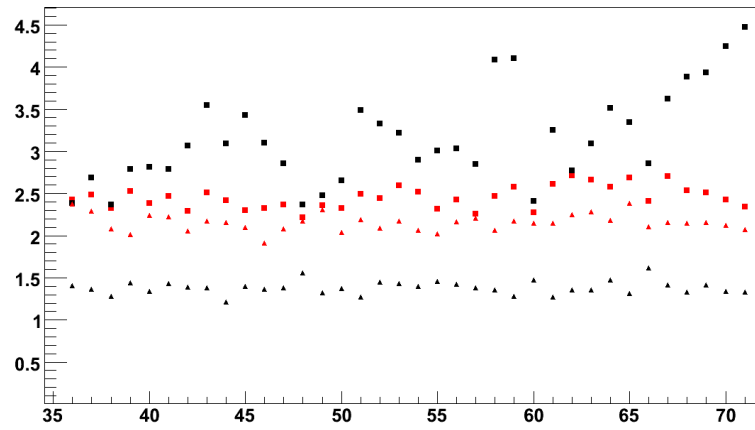


Figura 3.9: Valore medio su 64 canali di *rms* per i moduli di front-end da 36 a 71.

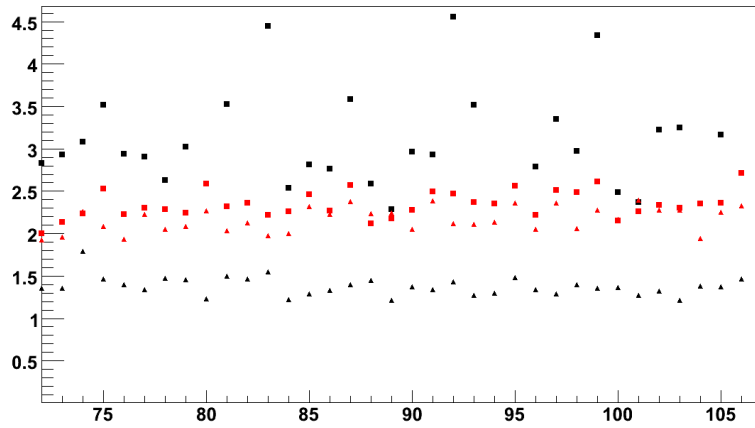


Figura 3.10: Valore medio su 64 canali di *rms* per i moduli di front-end da 72 a 107.

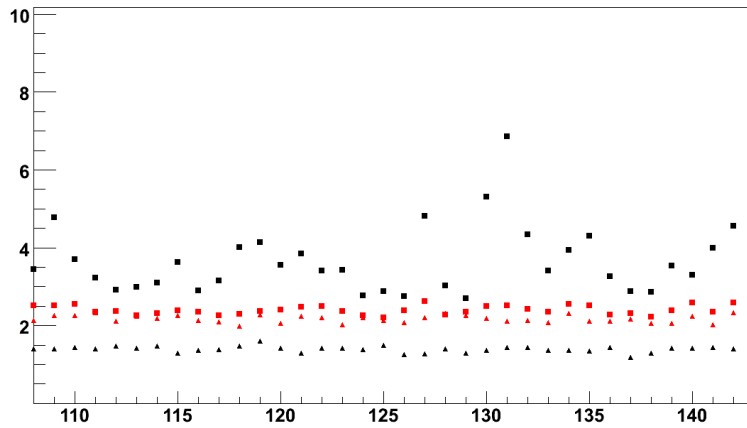


Figura 3.11: Valore medio su 64 canali di *rms* per i moduli di front-end da 108 a 143.

3.3.3 Schede con problemi maggiori

Durante la verifica del corretto funzionamento delle schede montate sul RICH si sono riscontrati alcuni malfunzionamenti. Le schede problematiche sono state rimosse, sostituite con altre e riportate a Torino per essere ricontrollate. In tabella, alcuni problemi riscontrati.

n. scheda	problema
104	crosstalk on channels 0 and 7
129	missing channel on canvas 1, channel 4
102	crosstalk on canvas 0 and 7, channel 0
84	crosstalk on canvas 0, 6, 2, 7 (very strong on 2, 7), channel 0
58	crosstalk on canvas 6, 7 (not very strong on canvas 4, 2), channel 0
128	crosstalk on canvas 0, 1, 7, channel 0
69	crosstalk on channel 0, channel 1 (only canvas 4, 5, 6), channel 3 (canvas 2, 6)
165	short circuit → changed B board; canvas 0 no signal (maybe >> offset); canvas 3, channel 6 no signal; canvas 5 no signal; canvas 7 large shape
145	Roof board changed

3.4 Creazione di una macro

per l'analisi dell'rms > 6 digit

Siccome lo scopo di questo studio è quello di confrontare i problemi di rumore, e quindi di relativo crosstalk, dei chip in sala e in laboratorio, si è approfondita l'analisi dei dati rilevati in ambiente sperimentale (CERN) per ottenerne un primo confronto, procedendo alla creazione di un duplice istogramma che ci permettesse di osservare in modo chiaro l'andamento dell'rms e della media (corrispondente alla posizione del piedistallo) dei fit gaussiani eseguiti sul segnale Rms e mean per ogni canale:

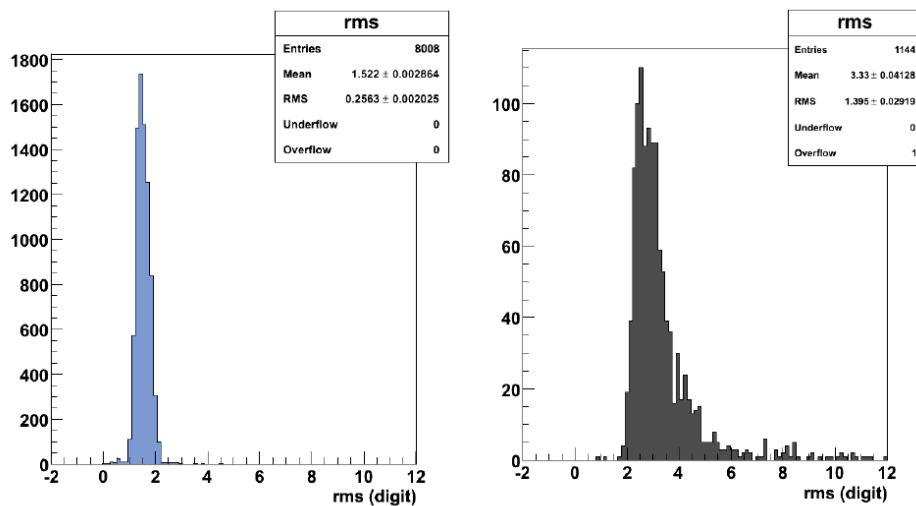


Figura 3.12: Distribuzione dell'rms del fit gaussiano al segnale di rumore per i canali 0 e 4 della scheda CMAD A (figura a destra) e per tutti gli altri canali sulle schede CMAD A e CMAD B (figura a sinistra).

3.4. CREAZIONE DI UNA MACRO CAPITOLO 3. TEST SUI CHIP E
PER L'ANALISI DELL'RMS > 6 DIGIT RELATIVA ANALISI STATISTICA

Siccome si è notato in precedenza quanto fossero più problematici in termini di rms troppo elevato i canali 0 e 4 delle schede CMAD A, si sono evidenziati i dati relativi ai canali 0 e 4 delle sole schede A; il risultato è mostrato in Figure 3.12 e 3.13.

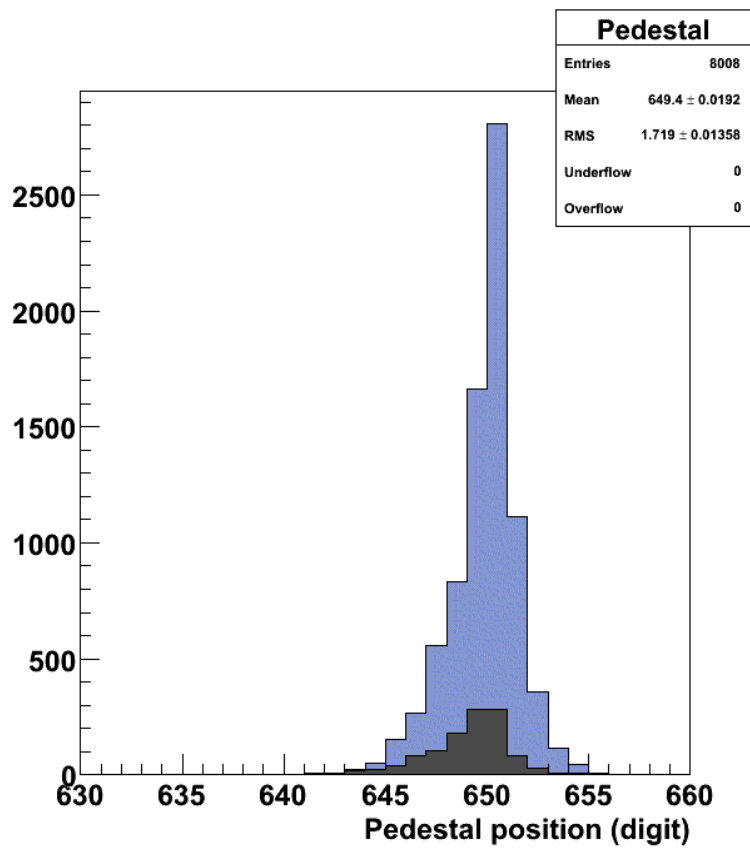


Figura 3.13: Canali 0 e 4 in grigio; i restanti in blu.

*CAPITOLO 3. TEST SUI CHIP E 3.4. CREAZIONE DI UNA MACRO
RELATIVA ANALISI STATISTICA PER L'ANALISI DELL'RMS > 6 DIGIT*

Infine si è creato l'istogramma in Figura 3.14 prendendo in considerazione i dati con rms maggiore di 6 digit e plottando sull'asse delle ascisse il numero di canale per scheda (da 0 a 63), e su quella delle ordinate quante volte quel determinato canale presenta un rms > 6 digit, che equivale ad un rumore di circa 3 fC: Dati con rms > 6 digit

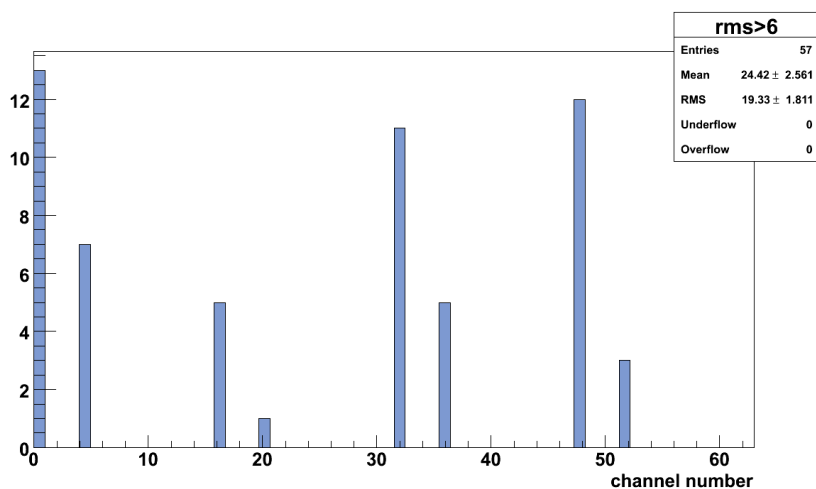


Figura 3.14: Numero di canali con rms maggiore di 6 digit.

Si vede che i canali problematici sono soprattutto, come prima si era già Canali problematici visto, lo 0 e il 4 di ogni scheda CMAD A ³; complessivamente, dei 64 canali si tratta dei canali 0, 4, 16, 20, 32, 36, 48, 52.

Nel complesso comunque il numero di canali con rms>6 è molto limitato, per la precisione si tratta di 57 su 9216 canali complessivi dell'elettronica di readout dei MAPMT, cioè 0.6% di tutti i canali.

Una volta ottenuti i numeri delle schede e dei canali con rms maggiore di 6 digit, si è completata una mappa (vedi Appendice .2, che contiene l'allocazione Mappa delle schede sull'esperimento, e i numeri di Roof e Catch di ciascuna).

³Vale a dire, rispettivamente, i canali 3 e 7 del chip.

3.5 Creazione di una macro

per l'analisi dell'rms > 3 digit

Dati con rms > 3 digit A questo punto si è passati ai dati raccolti in laboratorio in cui il *noise* tende ad essere in ogni caso inferiore a quello presente in sala sperimentale, e si sono voluti quindi discriminare i canali con rms superiore ai 3 digit anziché ai 6 digit (programma macro `histogram_test.C`, vedi Appendice .1), plottandone infine la distribuzione e richiedendo la stampa del nome del relativo file di calibrazione a cui corrisponde un dato numero di Roof board (vedi Appendice .3).

Confronto sala sperimentale - laboratorio Confrontiamo ora le schede più rumorose in laboratorio con quelle più rumorose in ambiente sperimentale e vediamo se esiste una correlazione.

In tabella riportiamo, dall'Appendice .2 i numeri delle schede Roof in cui compare almeno un canale con rms > 6. In grassetto sono segnalate quelle che sono presenti anche nell'elenco dell'Appendice .3 (laboratorio):

<i>3</i>	69	<i>9</i>	66	62	74	131	50	57
<i>13</i>	<i>70</i>	<i>32</i>	<i>70</i>	82	<i>92</i>	55	38	<i>40</i>
<i>14</i>	73	<i>34</i>	87	77	88	<i>45</i>	116	
<i>17</i>	74		<i>80</i>	91	90	<i>118</i>	113	
<i>28</i>	75		<i>89</i>			<i>112</i>		
	76							

Si riporta, anche per i dati presi in laboratorio, un istogramma analogo a quello di di Figura 3.14.

CAPITOLO 3. TEST SUI CHIP E 3.5. CREAZIONE DI UNA MACRO
 RELATIVA ANALISI STATISTICA PER L'ANALISI DELL'RMS > 3 DIGIT

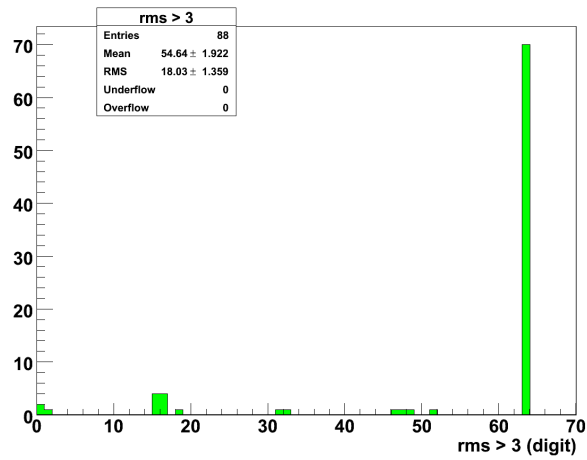


Figura 3.15: Canali con rms maggiore di 3 digit. Sull'asse delle ascisse si hanno i canali della Roof.

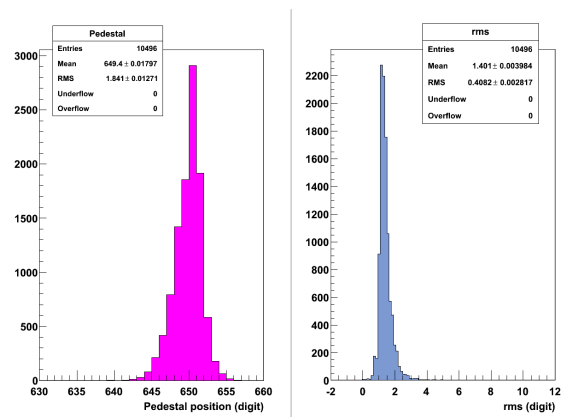


Figura 3.16: Posizione del piedistallo e distribuzione dell'*rms* (dati presi in laboratorio).

Chiaramente si nota quanto, in laboratorio, l'incidenza di rms maggiore di tre digit si abbia sul canale 63esimo.

3.5. CREAZIONE DI UNA MACRO CAPITOLO 3. TEST SUI CHIP E
 PER L'ANALISI DELL'RMS > 3 DIGIT RELATIVA ANALISI STATISTICA

I canali presenti con molteplicità non nulla nel grafico in Figura 3.15 sono quelli evidenziati in verde in tabella.

canale CMAD A	7	6	5	4	3	2	1	0
canale Roof	0	1	2	3	4	5	6	7
canale CMAD B	0	1	2	3	4	5	6	7
canale Roof	8	9	10	11	12	13	14	15
canale CMAD A	7	6	5	4	3	2	1	0
canale Roof	16	17	18	19	20	21	22	23
canale CMAD B	0	1	2	3	4	5	6	7
canale Roof	24	25	26	27	28	29	30	31
canale CMAD A	7	6	5	4	3	2	1	0
canale Roof	32	33	34	35	36	37	38	39
canale CMAD B	0	1	2	3	4	5	6	7
canale Roof	40	41	42	43	44	45	46	47
canale CMAD A	7	6	5	4	3	2	1	0
canale Roof	48	49	50	51	52	53	54	55
canale CMAD B	0	1	2	3	4	5	6	7
canale Roof	56	57	58	59	60	61	62	63

Dai dati mostrati in tabella si evince che, sistematicamente, il canale 7 di ogni chip è in genere il più rumoroso.

3.6 Creazione di una macro

per l'analisi dell'rms > 2 digit

In seguito, modificando opportunamente la macro precedente, ci si è interessati al plot dei valori con rms non più maggiore di tre digit ma, in modo più restrittivo, di due digit (~ 1 fC):

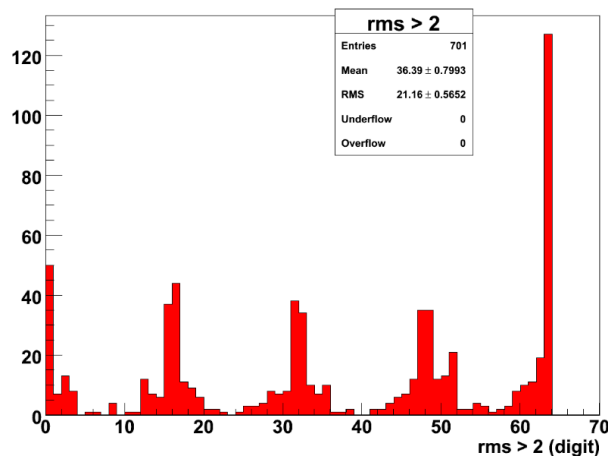


Figura 3.17: rms maggiore di 2 digit.

Questo plot risulta essere molto interessante per questa analisi, in quanto è un'ulteriore verifica del fatto che sia effettivamente il canale 7 di ogni chip ad influenzare i canali più prossimi (si vede chiaramente un effetto quasi di eco sui canali vicini ai settimi).

Che cos'ha di particolare in canale 7 del CMAD rispetto agli altri? Come si Rumore proveniente dalle
vede dagli schemi in Figure 2.16 e 2.17, il canale 7 è il primo vicino alle linee di calibrazione
delle linee di calibrazione IN_CAL_0 e IN_CAL_1. Ne deduciamo dunque
che, in particolari condizioni, disturbi esterni possono indurre del crosstalk sul
canale attraverso le porte di calibrazione.

3.7 Analisi del valor medio dell'rms del canale 7

Schede vs valor medio In ultimo è stata creata una macro per permettere di osservare la relazione tra i numeri delle 64 schede testate in laboratorio e il valore medio dell'rms del canale 7⁴ per ogni chip; il risultato è il seguente:

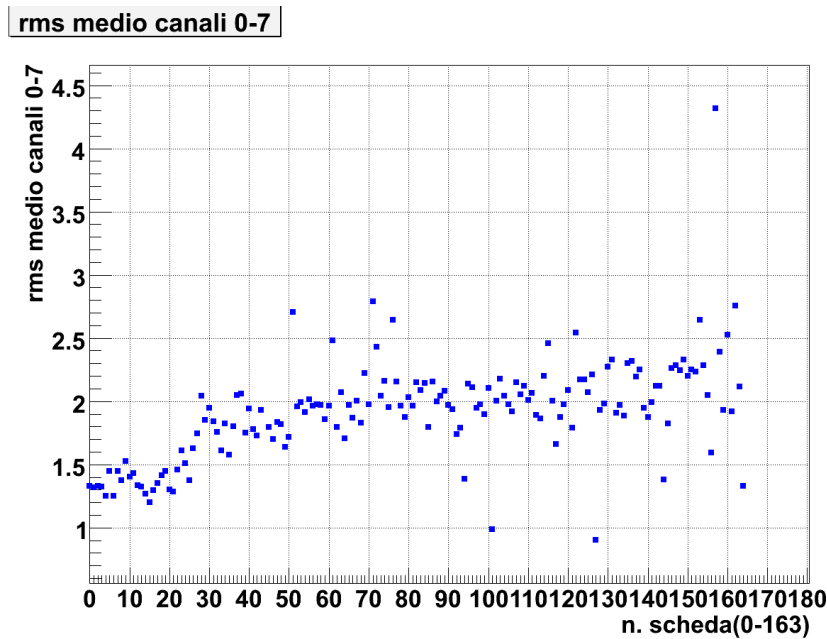


Figura 3.18: Rms medio dei canali 7 di ogni chip in funzione del numero di ogni scheda.

Si nota subito come le prime venticinque schede mostrino un rms basso (di circa 1.5 digit), mentre dalla venticinquesima in poi si ha un rms in generale maggiore, con picchi e dispersioni notevoli; questo significa che da quel momento in poi si è introdotta nel sistema di test una fonte non nota di rumore esterno, prima non presente.

La stessa figura è riproposta nelle pagine successive, ingrandita sull'asse delle ascisse, in modo da presentare 20 schede alla volta. In dettaglio:

⁴Un piccolo appunto sul numero del canale difettoso: sempre per i motivi sopracitati di lettura dei canali delle schede, si procederà in modalità bustrofedica e di conseguenza quelli che paiono, alternati, il canale 0 e il 7, sono esclusivamente e sempre settimi canali.

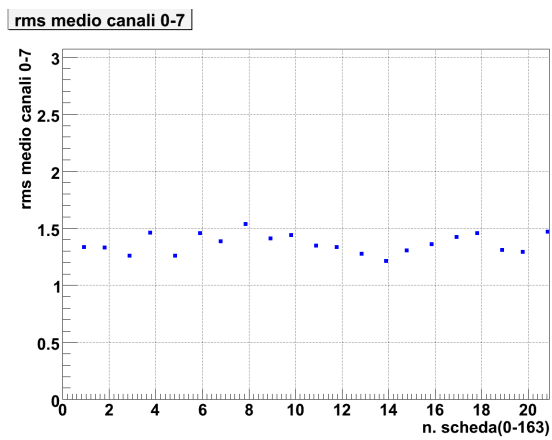


Figura 3.19: Schede da 0 a 20.

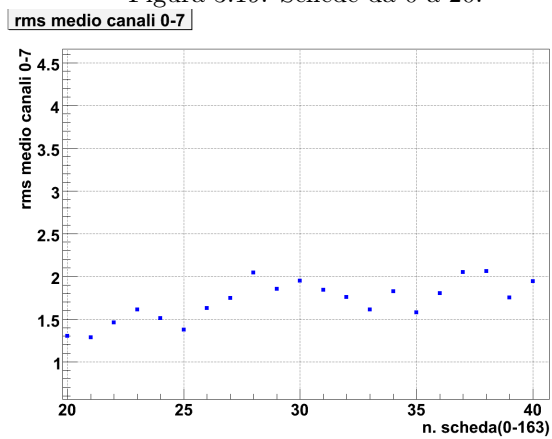


Figura 3.20: Schede da 20 a 40.

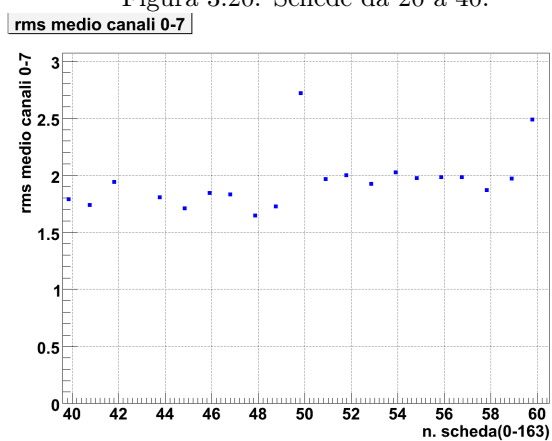


Figura 3.21: Schede da 40 a 60.

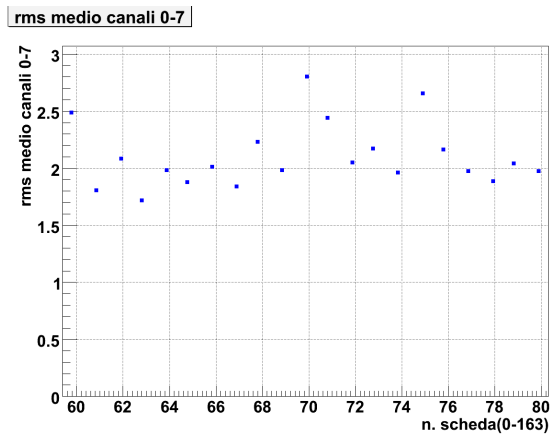


Figura 3.22: Schede da 60 a 80.

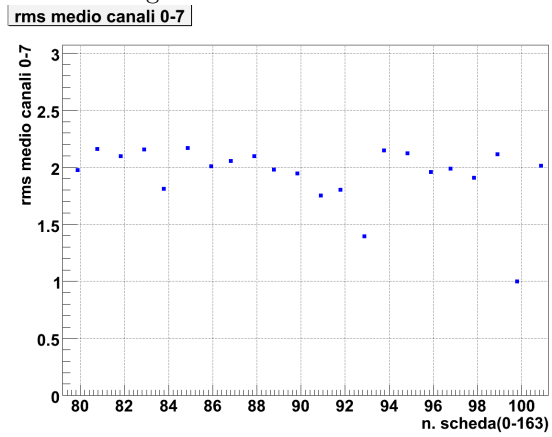


Figura 3.23: Schede da 80 a 100.

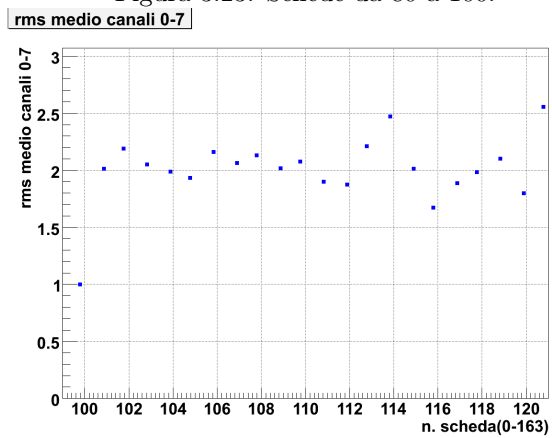


Figura 3.24: Schede da 100 a 120.

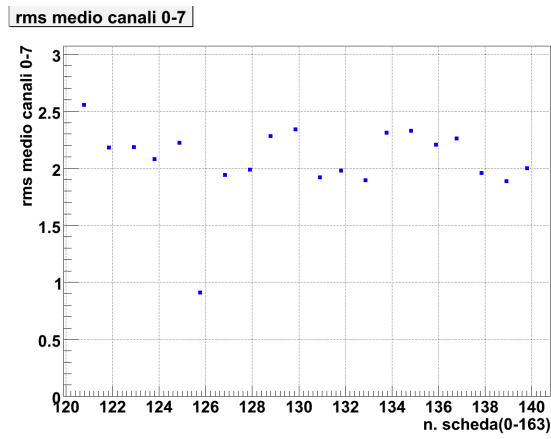


Figura 3.25: Schede da 120 a 140.

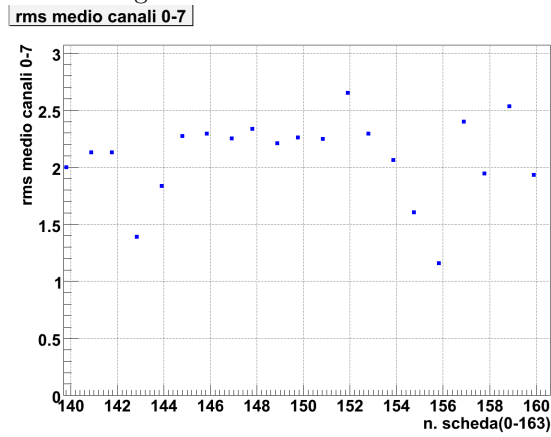


Figura 3.26: Schede da 140 a 160.

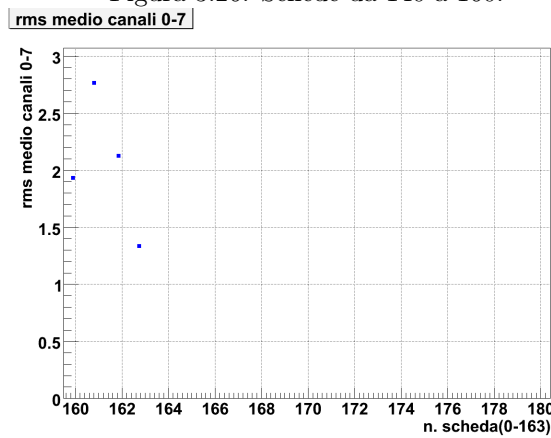


Figura 3.27: Schede da 160 a 180.

Per cercare di comprendere quale sia la sorgente del rumore osservato sono state effettuate ulteriori misure effettuate in diverse condizioni:

1. con resistenza da 50Ω sul connettore lemo sul connettore dell'ingresso del segnale di calibrazione IN_CAL_0 e IN_CAL_1 ;
2. senza resistenza da 50Ω sul connettore lemo;
3. con Roof prototipo (i files relativi a questo test sono preceduti da *prototype*);
4. con ventola di raffreddamento della DREISAM accesa;
5. con ventola spenta.

cercando di comprendere se la causa dei problemi di crosstalk riscontrati fosse adducibile ad una qualunque di queste modifiche, per così dire, *ambientali*.

Tuttavia i risultati più diversificati (e migliori) si sono avuti nell'ultima prova, quella che non prevedeva l'accensione della ventola sottostante la DREISAM posta al fine di raffreddare il circuito, ma anche in queste particolari condizioni la riduzione di *crosstalk* è molto limitata.

Considerazioni finali In conclusione non è possibile individuare la causa effettiva esterna del *crosstalk*, in quanto le migliori in questa particolare situazione sono in ogni caso trascurabili.

3.8 Impostazione della soglia

Come ultima parte ci si è curati dell'impostazione della soglia ottimale per tutti i canali dell'elettronica di front-end montata su RICH, andando a considerare, come sempre, quelli maggiormente problematici e creando macro di Root che mostrassero una statistica dei problemi rilevati.

La nostra richiesta è quella di impostare una soglia posizionata a sinistra del valore medio in logica invertita, in modo tale che il rumore (in questo caso, il valore di molteplicità dei punti di ordinata) non superi una determinata percentuale (che è stata impostata in un primo momento al 2%, in seguito al 5%) per ogni canale. Effettuare ciò consiste operativamente nel "tagliare fuori" eventuali fenomeni di *pulse* nella zona appena seguente alla "discesa" della curva.

Per ottenere un valore indicativo riguardo la posizione in cui impostare la soglia, si è definito il parametro $abs(x_point - (mean - 4 \cdot rms))$ che in seguito, per semplicità, chiameremo semplicemente *abs*.

In dettaglio, l' x_point è l'array di punti sull'asse delle ascisse minori del *mean*, il *mean* è il valore medio della gaussiana a cui viene tolto (in questo caso) quattro volte l'*rms*, e l'*rms* è l'ampiezza della curva a metà altezza; del tutto se ne considera il valore assoluto (*abs*). In Figura 3.28 è possibile vedere schematizzata l'operazione appena sopra descritta.

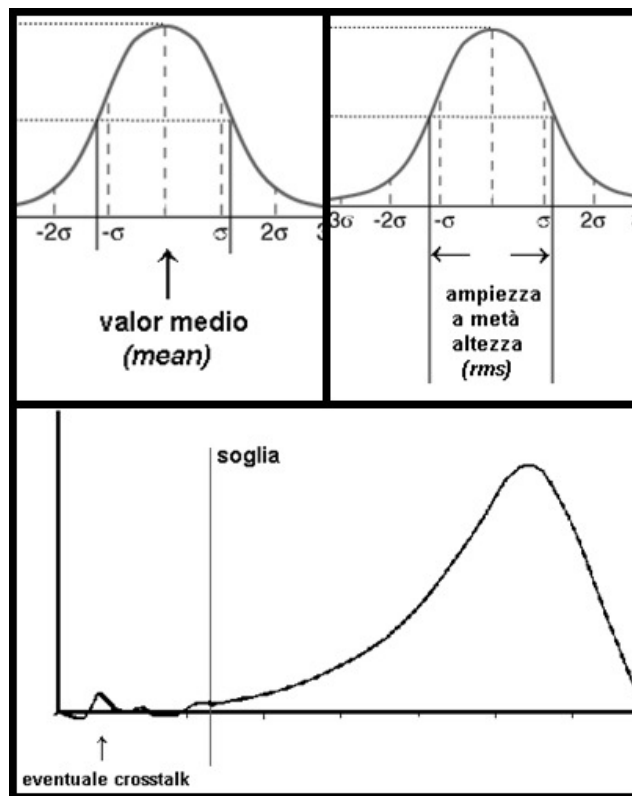


Figura 3.28: Nei due riquadri in alto, vediamo indicati il valor medio (*mean*) e l'*rms* delle curve. Nella figura in basso, vediamo un esempio di impostazione della soglia in un punto ottimale per la curva: essa è infatti sistemata alla fine della "discesa" e "taglia fuori" gli eventuali effetti di crosstalk.

CAPITOLO 3. TEST SUI CHIP E
RELATIVA ANALISI STATISTICA 3.8. IMPOSTAZIONE DELLA SOGLIA

In seguito, per affinare la ricerca, si sono effettuate diverse prove per scegliere un fattore moltiplicativo dell' *rms* (inizialmente è stato utilizzato il 4; in seguito, si è concluso che il valore ottimale è di 2,5), in modo tale da migliorarne il posizionamento.

I dati analizzati sono quelli da riferirsi a tre diversi scan effettuati con diversi valori di guadagno in ambiente sperimentale (vedi Appendice .2).

In primo luogo si è cercato innanzitutto di eliminare una parte di dati ponendo $abs(x_point - (mean - 4 \cdot rms)) > 4$ (valore ragionevole per l'andamento delle nostre curve), andando a riempire un istogramma che ci mostrasse la molteplicità delle entrate ad $abs > 4$ e, parallelamente, creando un grafico con in ascissa il numero dei moduli di elettronica, e in ordinata il valore di $abs > 4$ (Figura 3.29). Da questo è stato possibile ricavare un elenco di $abs > 4$ coi corrispondenti numeri di modulo di elettronica di front-end (vedi tabelle).

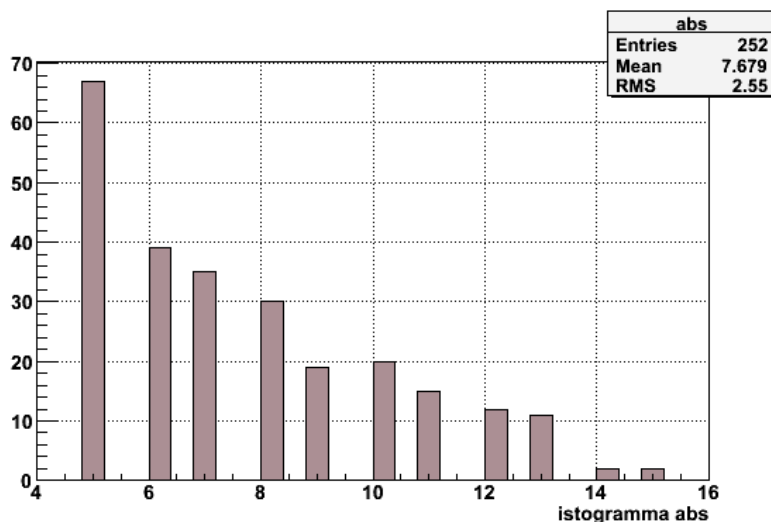


Figura 3.29: Istogramma riempito coi valori maggiori di 4 digit per il parametro *abs*.

3.8. IMPOSTAZIONE DELLA SOGLIA RELATIVA ANALISI STATISTICA

modulo 0 → 10

abs > 4	modulo
12	2
6	3
6	3
5	3
5	3
9	4
5	4
11	4
5	4
9	5
7	6
5	7
13	7
7	7
5	8
8	9
8	9
12	9
11	9
13	9
11	9
8	9
7	9
12	9
6	10

modulo 11 → 20

abs > 4	modulo
8	11
7	11
9	11
6	11
7	11
7	13
5	14
5	14
7	14
11	15
6	15
8	16
5	16
8	16
10	17
5	20
13	20
5	20
5	20
13	20
5	20
13	20
9	20
12	20
5	20

CAPITOLO 3. TEST SUI CHIP E

RELATIVA ANALISI STATISTICA 3.8. IMPOSTAZIONE DELLA SOGLIA

modulo 21 → 33

abs > 4	modulo
7	21
8	21
11	21
7	22
10	22
6	22
5	23
5	26
5	26
12	26
7	26
13	27
5	28
12	28
5	28
5	31
5	31
10	31
11	31
7	31
5	32
9	32
6	33
7	33
5	33

modulo 33 → 37

abs > 4	modulo
11	33
7	33
5	33
13	33
9	33
7	33
15	33
6	33
5	34
5	34
12	35
9	35
12	35
10	35
10	35
10	35
10	35
11	35
10	35
5	35
10	35
8	35
11	35
10	36
8	37

3.8. IMPOSTAZIONE DELLA SOGLIA RELATIVA ANALISI STATISTICA

modulo 38 → 40

abs > 4	modulo
6	38
9	38
5	38
7	39
5	39
7	39
5	39
9	39
7	39
5	39
13	39
11	39
9	39
6	39
7	39
9	39
5	39
7	39
7	39
6	39
5	40
8	40
13	40
14	40
12	40

modulo 41 → 45

abs > 4	modulo
8	41
9	41
6	41
7	41
9	41
8	41
7	41
5	41
5	41
5	41
8	42
6	43
11	43
9	43
9	43
12	43
10	43
10	43
8	43
8	43
8	43
6	44
5	44
11	44
5	45

CAPITOLO 3. TEST SUI CHIP E

RELATIVA ANALISI STATISTICA 3.8. IMPOSTAZIONE DELLA SOGLIA

modulo 45 → 51

abs > 4	modulo
6	45
8	46
5	47
5	47
6	47
6	47
5	48
7	48
5	48
13	48
5	49
5	49
8	50
6	50
9	50
8	50
5	50
7	50
6	50
8	50
6	50
7	50
5	51
6	51
8	51

modulo 52 → 58

abs > 4	modulo
6	52
10	52
7	52
9	52
5	52
5	52
7	53
10	53
6	53
8	54
6	54
5	54
7	54
7	55
5	55
5	56
5	56
7	56
7	56
8	57
8	57
7	57
5	57
5	57
8	58

3.8. IMPOSTAZIONE DELLA SOGLIA RELATIVA ANALISI STATISTICA

modulo 59 → 67

abs > 4	modulo
5	59
15	59
6	59
14	61
6	62
5	62
12	62
9	62
11	62
11	62
10	62
11	62
5	63
5	64
6	64
6	64
8	65
10	65
9	65
6	66
8	66
6	66
6	66
7	66
5	67
5	67

modulo 67 → 127

abs > 4	modulo
6	67
5	68
7	68
5	68
12	69
10	69
6	69
10	69
10	69
6	69
13	69
1	69
6	69
5	70
7	70
6	71
8	71
8	71
5	88
5	98
8	99
6	99
5	99
6	117
5	118
6	127

In seguito, prendendo in considerazione la serie di dati fornitici a guadagno 0 e guadagno 3, è stata fatta un'ulteriore analisi. In primis, come in precedenza, è stato creato un istogramma che mostrasse l'andamento della molteplicità del parametro *abs* (Figure 3.30 e 3.31).

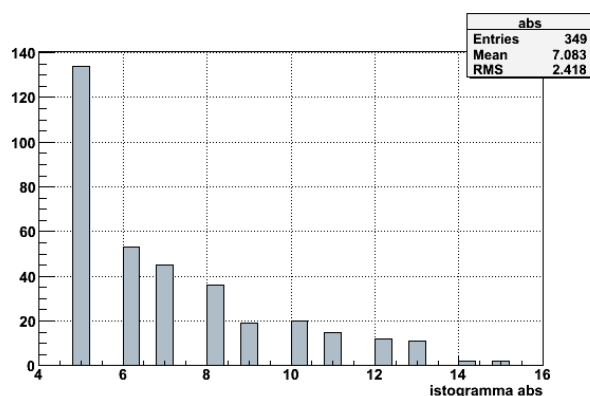


Figura 3.30: Molteplicità dei dati ottenuti a guadagno 0. In ordinata per *istogramma abs* si intende il valore sovracitato $abs(x_point - (mean - 4 \cdot rms))$.

3.8.1 Scelta del fattore numerico moltiplicativo per l'rms

Si è proceduto quindi ad una verifica qualitativa per la ricerca del fattore moltiplicativo dell'rms da inserire all'interno del parametro *abs_new*⁵ che permettesse una più ottimale possibile scelta della *threshold* per il maggior numero di canali. Si sono plottati i seguenti istogrammi⁶, andando a scegliere quello che mostrasse l' *rms* minore e un valore di *mean* il più possibile centrato sullo zero; la scelta è caduta sui 2,5 *rms*.

⁵Da questo momento, per osservare meglio l'andamento sia dei valori positivi che dei valori negativi, chiameremo il nuovo parametro *abs_new*, ricordandoci però che non si tratterà più di valore assoluto come il precedente *abs*.

⁶Gli istogrammi riportati fanno riferimento ai dati ottenuti con guadagno impostato a 0.

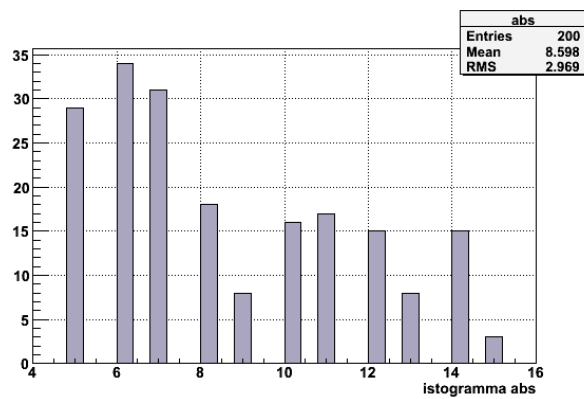


Figura 3.31: Molteplicità dei dati ottenuti a guadagno 3. In ordinata per *istogramma abs* si intende il valore sovracitato $abs(x_point - (mean - 4 \cdot rms))$.

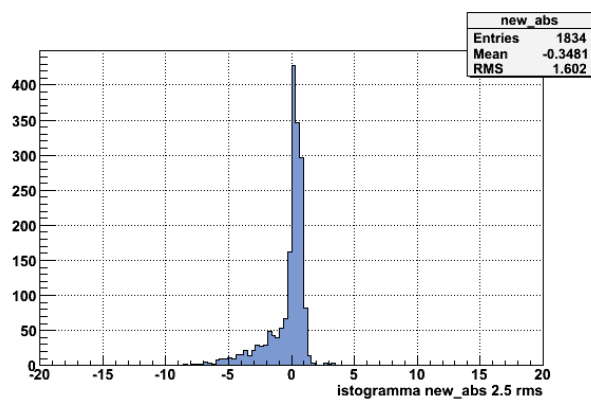


Figura 3.32: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 2.5, scelto per le nostre analisi.

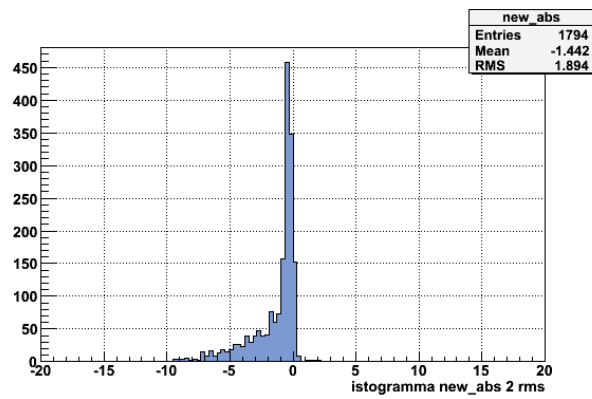


Figura 3.33: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 2.

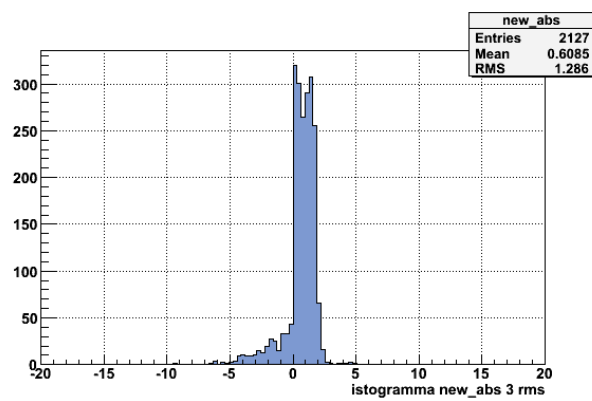


Figura 3.34: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 3.

CAPITOLO 3. TEST SUI CHIP E
3.8. IMPOSTAZIONE DELLA SOGLIA RELATIVA ANALISI STATISTICA

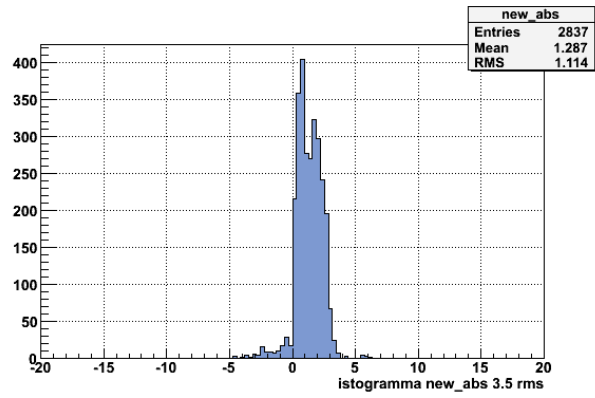


Figura 3.35: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 3.5.

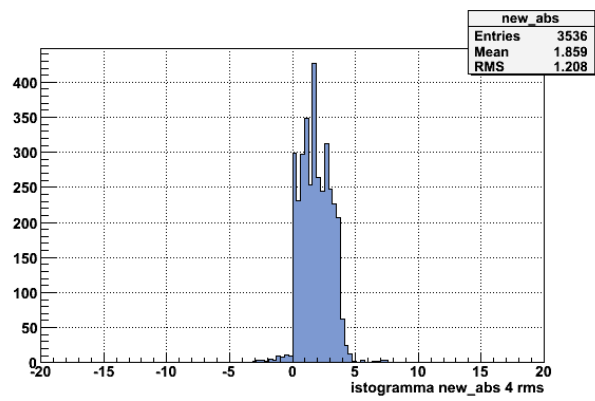


Figura 3.36: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 4.

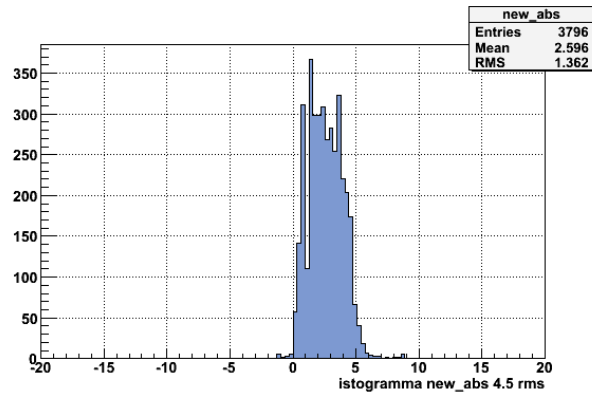


Figura 3.37: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 4.5.

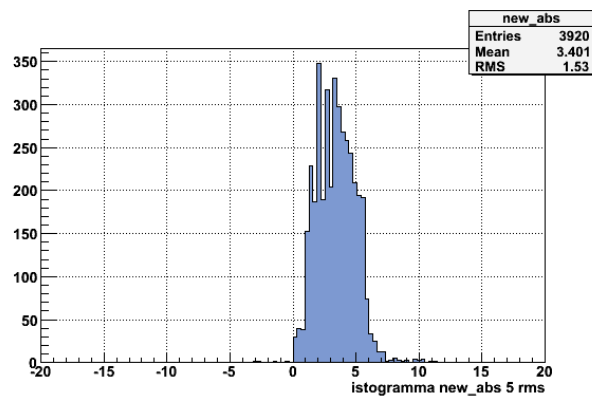


Figura 3.38: Istogramma per i dati relativi al fattore moltiplicativo per l'rms pari a 5.

Scelto, quindi, come fattore moltiplicativo per l'rms, il valore di 2.5, si è proceduto in una veloce verifica dei canali più rumorosi e si sono cercati, parallelamente, i corrispettivi, già più rumorosi, con $rms > 6$, sul file di mapping (vedi Appendice .2), che sono risultati essere i moduli di front-end appartenenti alla sezione Bottom Saleve, Catch 505, Plot 69-70-71. I valori confrontati sono i seguenti:

SCHEDE CON RMS < 6 (sala sperimentale)^{7 8}

BOTTOM SALEVE

catch	505	505	505
porta	5	6	7
plot	69	70	71
canale	48	48	32
	52	52	36

GAIN 3

plot	canale
69	52 *
70	52 *
71	16 *
	32 *
	52 *

GAIN 0

plot	canale
69	4
70	(nessuno)
71	16
	20

Mentre i grafici di riferimento sono i seguenti:

⁷Considero i canali con $*abs_new_ < -5$

⁸I valori segnati con la (*) sono quelli già presenti nel file di mapping.

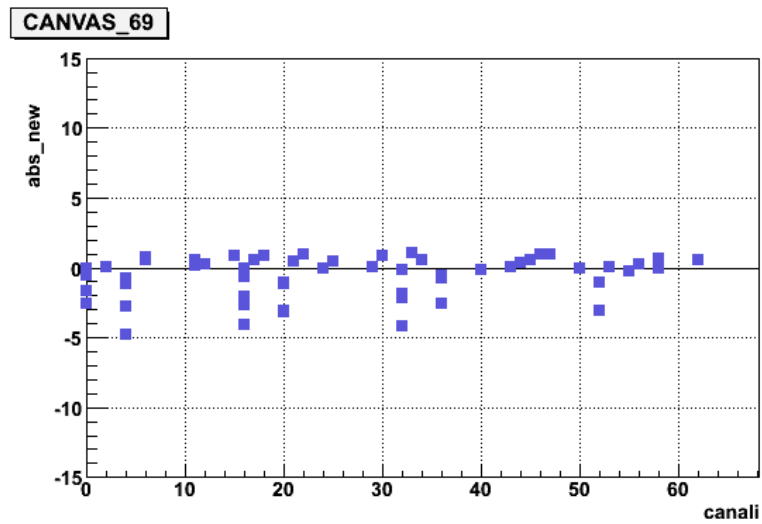


Figura 3.39: Istogramma per i dati relativi a guadagno 0 per il grafico 69.

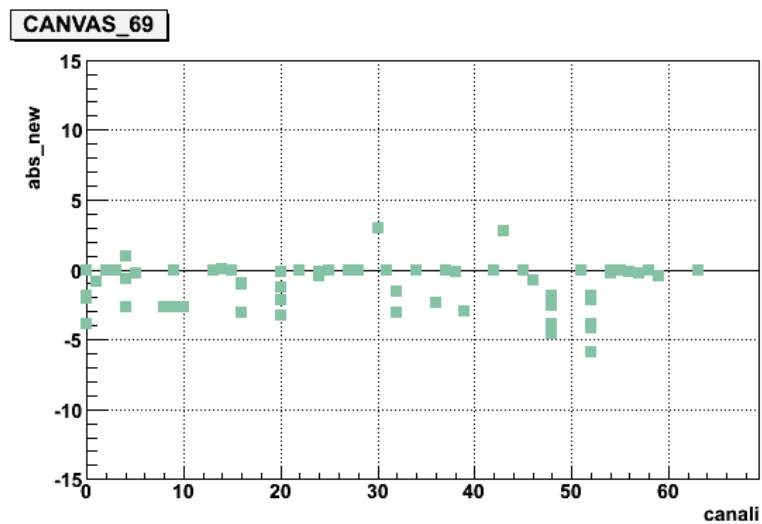


Figura 3.40: Istogramma per i dati relativi a guadagno 3 per il grafico 69.

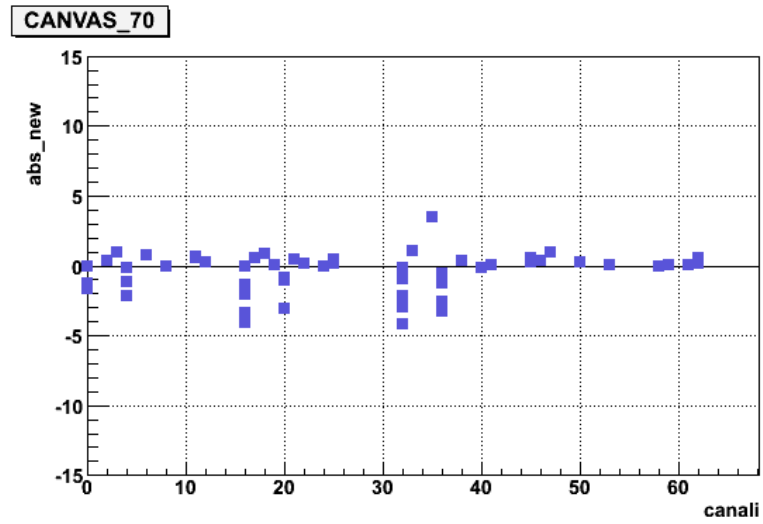


Figura 3.41: Istogramma per i dati relativi a guadagno 0 per il grafico 70.

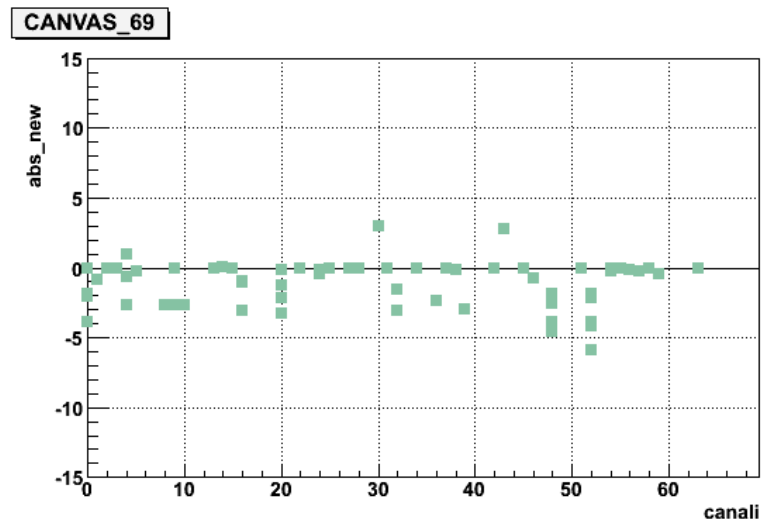


Figura 3.42: Istogramma per i dati relativi a guadagno 3 per il grafico 70.

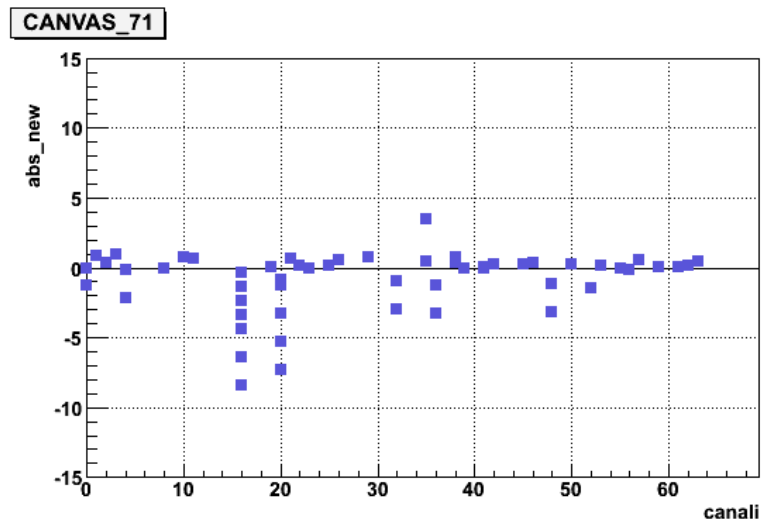


Figura 3.43: Istogramma per i dati relativi a guadagno 0 per il grafico 71.

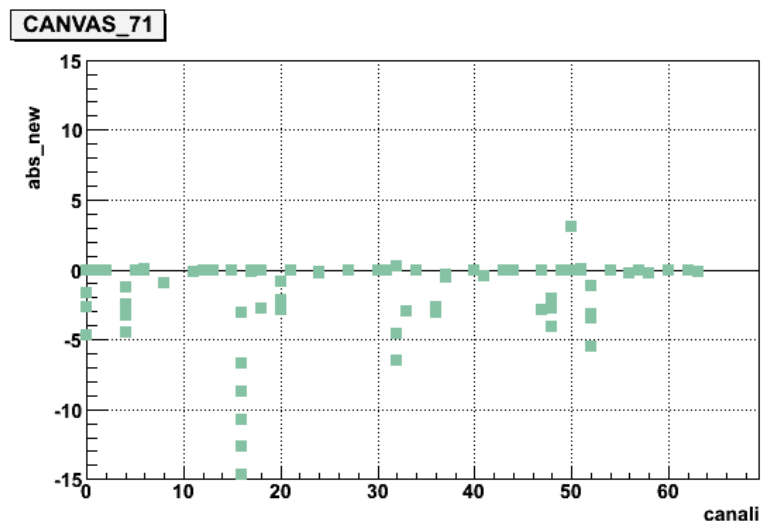


Figura 3.44: Istogramma per i dati relativi a guadagno 3 per il grafico 69.

3.8.2 Impostazione finale della soglia

sui 144 moduli di front-end

A parte è stata eseguita un'analisi punto per punto dei vari fit su tutti i canali, andando ad osservare l'andamento crescente o decrescente della curva, impostando la soglia a 4 rms per i canali più rumorosi. Tramite questa prima impostazione, sono stati ricavati tre files di dati: i primi due, facenti riferimento all'analisi dei settantadue moduli dei rivelatori del Bottom Saleve e Jura (dallo 0 al 71) a guadagno 0 e a guadagno 3, mentre il terzo comprende l'estensione generale a tutti i moduli, ovvero dallo 0 al 143, a guadagno 0. Grazie a questa serie di dati, ci è fornito il valore della soglia ottimale impostata per ciascun canale ed è stato possibile ottenere il seguente istogramma (macro in Appendice .1.3 e .1.4), che mette in evidenza il parametro *mean-soglia*, ribattezzata *soglia definitiva*, che sancisce la distanza della soglia impostata e il valor medio della curva.

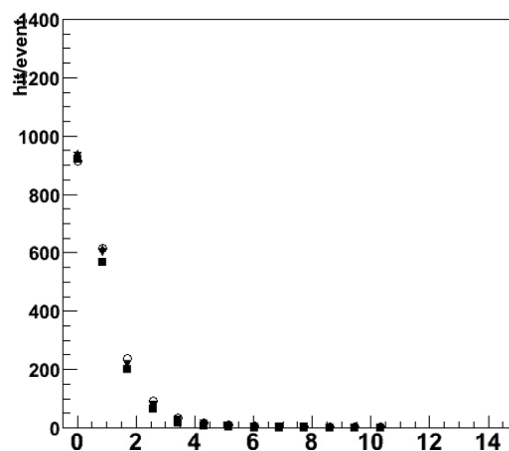


Figura 3.45: La figura mostra lo scan di soglia finale, in sala sperimentale: molteplicità media per ogni soglia sui 230 canali di ognuno dei 4 quadranti (i diversi simboli, cerchietto, stellina, quadratino, triangolino, sono relativi ai diversi quadranti Top e Bottom Saleve, Top e Bottom Jura).

CAPITOLO 3. TEST SUI CHIP E

RELATIVA ANALISI STATISTICA 3.8. IMPOSTAZIONE DELLA SOGLIA

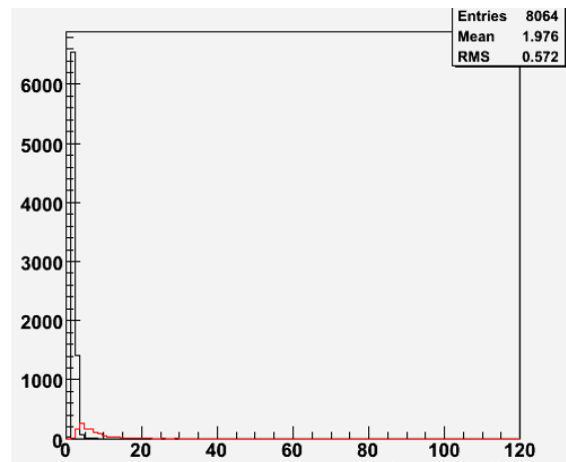


Figura 3.46: In figura, la distribuzione delle soglie finali: in rosso i canali 0 e 4 delle schede CMAD A, in nero tutti gli altri.

Conclusioni

Al termine delle analisi illustrate in questa tesi è possibile trarre alcune importanti conclusioni.

Il chip CMAD offre grandi versatilità e flessibilità d'uso, mostrando un netto miglioramento rispetto al precedente MAD4, e parallelamente anche un ottimo comportamento per quanto riguarda la soppressione del rumore. Dagli studi effettuati, infatti, si evince che il rumore è quantitativamente e quantitativamente trascurabile in rapporto al valore di rumore intrinseco del chip e se si considerano le effettive esigenze dell'apparato sperimentale.

Al momento della stesura della tesi i chip prodotti in serie e testati sono già stati installati sul rivelatore e funzionanti.

Le possibilità di controllo e gestione di guadagno, baseline e soglia, unite al sostenimento di grandi frequenze di eventi, fanno sì che il CMAD possa essere impiegato con ottimi risultati anche su diversi altri apparati, rendendolo largamente utilizzabile per scopi di ricerca scientifica e industriali.

Capitolo 4

Appendice

.1 Codici

.1.1 Codice di histogram_test.C

```
{
int thr_min=630;
int thr_max=660;
int canvas_min=0;
int canvas_max=10;
char filename[40];
int mean;
double rms;
char filename[40];
double threshold[30];
char * sep = " to ";
    TList * filelist = new TList();

    FILE * fin = fopen("lista2_serena.dat","r");
    while ( fscanf (fin,"%s\n", filename)!= EOF){
        TObjectString * s = new TObjectString(filename);
        TString name = s->GetString();
        filelist -> Add (s);
```

```
    }  
    TH1F *fhisto_pedestal2 = new TH1F  
        ("Pedestal","",30,thr_min,thr_max);  
    TH1F *fhisto_noise2 = new TH1F("rms","",100,-2,12);  
    TH1F *fhisto_rms3 = new TH1F("rms > 3","",70,0,70);  
  
    TIter next(filelist);  
    TObjString * str;  
    int j=0;  
    int p=0;  
    char str2[1000];  
    int i=0  
  
    while ( str = (TObjString * ) next() ) {  
        cout << str->GetString() <<endl;  
        char * fname = str->GetString();  
        fstream fddata;  
        fddata.open(fname,ios::in);  
        int nchannel = 0;  
        while (1){  
            char line[256];  
            fddata.getline(line,256);  
            if (fddata.eof() ) break;  
            stringstream fddatastr;  
            fddatastr << line;  
            fddatastr >> mean;  
            fddatastr >> rms;  
  
            if (rms >=3){  
                cout << "calibration file: " << str->GetString() << endl;  
                cout << "nchannel: " << nchannel << endl;  
                cout << " line: " << line << endl;  
                fhisto_rms3->Fill(nchannel);
```

```
}
fhisto_pedestal2->Fill(mean);
fhisto_noise2->Fill(rms);
    line++;
    nchannel++;
    }
}
TCanvas *canvas_histo = new TCanvas("Histogram",
                                   "Histogram",172,64,577,543);
canvas_histo->Divide(2,1);
fhisto_pedestal2->GetXaxis()
    ->SetTitle("Pedestal position (digit)");
fhisto_noise2->GetXaxis()->SetTitle("rms (digit)");
fhisto_pedestal2->SetFillColor(6);
fhisto_noise2->SetFillColor(38);
TCanvas *canvas_rms3 = new TCanvas
    (" rma > 3 ", " rms > 3 ",172,64,577,543);
fhisto_rms3->GetXaxis()->SetTitle("rms > 3 (digit)");
fhisto_rms3->SetFillColor(3);
    gROOT->SetStyle("Plain");
    gStyle->SetOptStat(299584);
canvas_histo->cd(1);
fhisto_pedestal2->Draw();
canvas_histo->cd(2);
fhisto_noise2->Draw();
canvas_rms3->cd();
fhisto_rms3->Draw();
}
```

.1.2 Codice di graph_rms3.C

```
{
char filename[40];
int mean;
double rms;
double threshold[30];
char * sep = " to ";
double somma[200];
double media[200];
for(int i=0; i<200; i++)
{
somma[i]=0;
media[i]=0;
cout << "somma: " << somma[i] << "media: " << endl;
cout << media[i] << endl;
}
// *****
//                               DATI DA LEGGERE
// *****

TList * filelist = new TList();
FILE * fin = fopen("lista2_serena.dat","r");

while ( fscanf (fin,"%s\n", filename)!= EOF){
    TObjString * s = new TObjString(filename);
    TString name = s->GetString();
    filelist -> Add (s);
}
TIter next(filelist);
TObjString * str;
int i=0;
int k=0;
char str2[1000];
int nscheda_array[200];
```



```
TCanvas *c1 = new TCanvas("c1","GRAFICO",200,10,600,400);
c1->SetFillColor(10);
c1->SetGrid();

while ( str = (TObjString * ) next() ) {
    cout << str->GetString() <<endl;

char * fname = str->GetString();
fstream fddata;
    fddata.open(fname,ios::in);
    int nchannel=0;

while (1){
    char line[256];
    fddata.getline(line,256);
    if (fddata.eof() ) break

        // apre un datastream
stringstream fddatastr;
    fddatastr << line;
    fddatastr >> mean;
    fddatastr >> rms;

// *****
//                               CICLO PER L'RMS
// *****

if(nchannel==0 || nchannel==15 || nchannel==16 ||
    nchannel==31 || nchannel==32 || nchannel==47 ||
        nchannel==48 || nchannel==63 )
{
    cout << " canale: " << nchannel << endl;
```

```
        " rms: " << rms << endl;
    somma[k]+=rms;
}
line++;

if (rms>=3){
cout << " canale scelto: " << nchannel << endl;
    " rms maggiore o uguale 3: " << rms << endl;
}
nchannel++;
//media[k] = somma[k]/8;
cout <<" somma[k]" << somma[k] << endl;
    }
// *****
//             OTTENIMENTO NUMERO SCHEDA
// *****

    char board_number[3];
cout << "fname: " << str->GetString() << endl;
int bn=atoi(board_number);
fn_board=bn;
cout << " fn_board " << fn_board << endl;
    char fname_array[40];
    for (int i=0; i<40; i++){
        fname_array[i]=fname[i];
    }
cout << " lunghezza array: " << sizeof(fname_array) << endl;
cout << " carattere 33°: " << fname_array[i-7] << endl;
    if(fname_array[38])
cout << " carattere 34°: " << fname_array[i-6] << endl;
    if(fname_array[39])
cout << " carattere 35°: " << fname_array[i-5] << endl;
char K;
```

```
u_array[0] = fname_array[i-7];
    if(fname_array[38])
u_array[1] = fname_array[i-6];
else
    u_array[1]=K;
    if(fname_array[39])
u_array[2] = fname_array[i-5];
else
    u_array[2]= K;

    int numeroscheda = atoi(u_array);
cout << " numero scheda (in numero): " << numeroscheda << endl;
    nscheda_array[k]=numeroscheda;

    cout << " numero scheda array: " << nscheda_array[k] << endl;
    cout << " k: " << k << endl;
k++;
}

Double_t x["nscheda_array"], y["valore medio canale 7"];
for (Int_t i=0; i<k; i++) {

x[i] = nscheda_array[i];
y[i] = somma[i]/8;

    cout << " x = " << x[i] << " y = " << y[i] << endl;
    }
// *****
//
//          PARTE GRAFICA
// *****

gr = new TGraph(k,x,y);
// gr->SetLineColor(2);
```

```
//gr->SetLineWidth(1);  
    gr->SetMarkerColor(4);  
    gr->SetMarkerStyle(21);  
    gr->SetTitle("rms medio canali 0-7");  
    gr->GetXaxis()->SetTitle("n. scheda(0-163)");  
    gr->GetYaxis()->SetTitle("rms medio canali 0-7");  
    gr->Draw("AP");  
}
```

.1.3 Codice di lettura _dati.C

```
{
char filename[40];
double ordinata;
double ascissa;
int canvas;
int canale;
double meanmeno5rms;
double mean;
double rms;
double abs;
double abs_array[10000];
double canale_array[10000];
double canvas_array[10000];
double rms_array[10000];
double elenco_canali_array[10000];
//double new_abs_array[1000];
double elenco_separato_array_1[10000];
double elenco_separato_array_2[10000];
double elenco_separato_array_3[10000];
double canale_array_1[1000][1000];
double nuovo_abs_array[1000][1000];
double new_abs_array[10000];
char canvas_name[200];
    int k=0;
    int j=0;
    int m=0;
    int n=0;

// *****

fstream fddata;
```

```
// fddata.open("scan_data_CMAD.txt",ios::in);
fddata.open("scan_data_23Mar09_G0.txt",ios::in);
//fddata.open("scan_data_24Mar09_G3.txt",ios::in);

//TH1F * fhisto_abs= new TH1F("abs","",30,3,15);
TH1F * fhisto_canale= new TH1F("canale","",126, -1,63);
TH1F * fhisto_canvas= new TH1F("canvas","",142,0,71);
TH1F * fhisto_doppiocanale= new TH1F("doppio canale","",126,-1,63);
//TH1F * fhisto_rms= new TH1F("rms","",20,0,10);
TH1F * fhisto_new_abs= new TH1F("new_abs","",126,-20,20);

// *****

while (1){
    char line[37];
    fddata.getline(line,37);
    if (fddata.eof() ) break;

    stringstream fddatastr;
    fddatastr << line;
    fddatastr >> ordinata;
    fddatastr >> ascissa;
    fddatastr >> canvas;
    fddatastr >> canale;
    fddatastr >> meanmeno5rms;
    fddatastr >> mean;
    fddatastr >> rms;
    fddatastr >> abs;

    //cout << " ordinata: " << ordinata << " ascissa: " << ascissa << endl;
    //cout << " canvas: " << canvas << " canale: " << canale << endl;
    //cout << " mean-3*rms: " << meanmeno5rms << " mean: " << mean << endl;
    //cout << " rms: " << rms << " abs: " << abs << endl;
```

```
// *****  
//  
//                                RIEMPIMENTO ARRAY  
// *****  
  
//cout << " abs: " << abs << endl;  
//if(abs > 3)  
//cout << " abs > 3: " << abs << " canvas: " << canvas << endl;  
//cout << " canale: " << canale << endl;  
//int abs_molteplicita=abs;  
//int canale_molteplicita==canale;  
    //cout << " new_abs = " << new_abs << endl;  
double new_abs=(ascissa-(mean-(2.5)*rms));  
//int canalepercanvas= canale*canvas;  
// for(int j=0; j<72; j++){  
    //}  
if((new_abs <= 0 && ordinata >= 0.01)|| (new_abs > 0 && ordinata < 0.01)){  
  
rms_array[k]=rms;  
abs_array[k]=new_abs;  
canale_array[k]=canale;  
canvas_array[k]=canvas;  
elenco_canali_array[k]=k;  
new_abs_array[k]=new_abs;  
  
    for (int b=0; b<72; b++){  
  
        if(canvas==b){  
nuovo_abs_array[b][j]=new_abs;  
canale_array_1[b][j]=canale;  
        cout << " canale_array_1[b][j]: " << canale_array_1[b][j] <<endl;  
        cout << " canvas: " << canvas << endl;  
        j++;  
    }  
    }  
}
```

```
}  
}  
  
// if(k<64){  
// canale_array_1[k]=canale;  
// cout << " canale_array_1[k]: " << canale_array_1[k] << endl;  
// cout << " canvas: " << canvas << endl;  
// }  
// cout << " k = " << k << endl;  
// cout << " rms = " << rms << endl;  
// cout << " ordinata: " << ordinata << endl;  
// cout << " abs:      " << abs << endl;  
// cout << " canvas:   " << canvas << endl;  
// cout << " canale:   " << canale << endl;  
// //cout << "   abs_array[k]: " << abs_array[k] << endl;  
// //cout << "   || canale_array[k]: " << canvas_array[k] << endl;  
// cout << " rms < n : " << rms << " canale: " << canale << endl;  
// cout << " -----" << endl;  
// cout << "new_abs = " << new_abs << endl;  
// cout << " -----" << endl;  
  
//fhisto_abs->Fill(abs_array[k]);  
fhisto_canale->Fill(canale_array[k]);  
fhisto_canvas->Fill(canvas_array[k]);  
//fhisto_rms->Fill(rms_array[k]);  
fhisto_new_abs->Fill(new_abs);  
  
k++;  
  
}  
}  
  
// *****
```



```
for(int l=0;l<k;l++){

    // cout << canale_array[l] << "    " << l << endl;

    if(canale_array[l]==canale_array[l+1]){

        // cout << canale_array[l] << "    " << l << endl;

        if(canale_array[l]==canale_array[l+1]){

            // cout << " canale_array[l]: " << canale_array[l] << endl;
            // cout <<" canale_array[l+1]: " << canale_array[l+1] << endl;
            // cout << " canvas_array[l]: " << canvas_array[l] << endl;
            // cout << " canvas_array[l+1]: " << canvas_array[l+1] << endl;
            // cout << " ***** " << endl;

            fhisto_doppiocanale->Fill(canale_array[l]);

        }
    }
}

if(canvas==b){

    canale_array_1[b][j]=canale;
    cout << " canale_array_1[b][j]: " << canale_array_1[b][j] << endl;
    cout << " canvas: " << canvas << endl;

    j++;

}
}
```

```
// cout << " canale_array[1]: " << canale_array[1] << endl;
// cout <<" canale_array[1+1]: " << canale_array[1+1] << endl;
// cout << " canvas_array[1]: " << canvas_array[1] << endl;
// cout << " canvas_array[1+1]: " << canvas_array[1+1] << endl;
// cout << " ***** " << endl;

fhisto_doppiocanale->Fill(canale_array[1]);

}

// *****
//                                     GRAFICO
// *****

TCanvas *c1 = new TCanvas("c1","abs vs canvas",200,10,600,400);
c1->SetFillColor(10);
c1->SetGrid();

gr = new TGraph(k,canvas_array,abs_array);
// gr->SetLineColor(2);
// gr->SetLineWidth(1);
gr-> SetMinimum(-8);
    gr->SetMarkerColor(36);
    gr->SetMarkerStyle(21);
    gr->SetTitle(" ");
    gr->GetXaxis()->SetTitle("canvas");
    gr->GetYaxis()->SetTitle("");
    gr->Draw("AP");

// *****

// TCanvas *c2 = new TCanvas("c2","istogramma abs",200,10,600,400);
```

```
// c2->SetFillColor(10);
// c2->SetGrid();
// fhisto_abs->GetXaxis()->SetTitle("istogramma abs");
// fhisto_abs->SetFillColor(40);
// fhisto_abs->Draw();

// *****

TCanvas *c3 = new TCanvas("c3","istogramma canale",200,10,600,400);
c3->SetFillColor(10);
c3->SetGrid();
fhisto_canale->GetXaxis()->SetTitle("istogramma canale");
fhisto_canale->SetFillColor(46);
fhisto_canale->Draw();

// *****

TCanvas *c4 = new TCanvas("c4","istogramma canvas",200,10,600,400);
c4->SetFillColor(10);
c4->SetGrid();
fhisto_canvas->GetXaxis()->SetTitle("istogramma canvas");
fhisto_canvas->SetFillColor(41);
fhisto_canvas->Draw();

// *****

TCanvas *c5 = new TCanvas("c5","istogramma canale doppio",200,10,600,400);
c5->SetFillColor(10);
c5->SetGrid();
fhisto_doppiocanale->GetXaxis()->SetTitle("istogramma doppio canale");
fhisto_doppiocanale->SetFillColor(30);
fhisto_doppiocanale->Draw();
```

```
// *****

// TCanvas *c6 = new TCanvas("c6","istogramma rms >= 2.5",200,10,600,400);
// c6->SetFillColor(10);
// c6->SetGrid();
// fhisto_rms->GetXaxis()->SetTitle("istogramma rms >= 2.5");
// fhisto_rms->SetFillColor(29);
// fhisto_rms->Draw();

// *****

TCanvas *c7 = new TCanvas("c7","istogramma new_abs 2.5 rms ",200,10,600,400);
c7->SetFillColor(10);
c7->SetGrid();
fhisto_new_abs->GetXaxis()->SetTitle("istogramma new_abs 2.5 rms");
fhisto_new_abs->SetFillColor(38);
fhisto_new_abs->Draw();

// *****

TCanvas *canvas_plot[72];

for(int b=0; b<72; b++){

    sprintf(canvas_name,"CANVAS_%d",b);

    canvas_plot[b] = new TCanvas(canvas_name, canvas_name,200,10,600,400);

    canvas_plot[b]->SetFillColor(10);
    canvas_plot[b]->SetGrid();

    gr = new TGraph(j,canale_array_1[b],nuovo_abs_array[b]);
    // gr->SetLineColor(2);
```

```
// gr->SetLineWidth(1);
gr-> SetMinimum(-15);
gr-> SetMaximum(+15);
    gr->SetMarkerColor(30);
    gr->SetMarkerStyle(21);
    gr->SetTitle(canvas_name);
    gr->GetXaxis()->SetTitle("canali");
    gr->GetYaxis()->SetTitle("abs_new_boh");
    canvas_plot[b]->cd();
    gr->Draw("AP");
}

// *****

cout << " mappa di lettura " << endl;
int w=0;
for(int n=0; n<72; n++){
    for(int m=0; m<64; m++){
        cout << "canale: " << m << " canvas: " << n << " w: " << w << endl;
        w++;
    }
}
}
```

.1.4 Codice di lettura_dati_2.C

```
{
char filename[40];
double ordinata;
int canvas;
int canale;
double mean;
double rms;
double abs;
int k=0;
double sogliaeffettiva_array[10000];
double soglia;
double soglia2;
double parametro_a;
double parametro_b;

// *****

fstream fddata;

fddata.open("scan_data_23Mar09-0-71.txt",ios::in);
// fddata.open("scan_data_24Mar09_G3-0-71.txt",ios::in);

//fddata.open("scan_data_Dec08_G0.txt",ios::in);

TH1F * fhisto_sogliaeffettiva= new TH1F("mean meno soglia","",23,0,23);

// *****

while (1){

    char line[50];
    fddata.getline(line,50);
```

```
if (fddata.eof() ) break;

stringstream fddatastr;

// // dati per scan_data_Dec08_G0.txt

// fddatastr << line;
//     fddatastr >> ordinata;
//     fddatastr >> soglia;
//     fddatastr >> mean;
//     fddatastr >> canvas;
//     fddatastr >> canale;
//     fddatastr >> soglia2;
//     fddatastr >> rms;
//     fddatastr >> parametro_a;
//     fddatastr >> parametro_b;

// dati per scan_data_23Mar09-0-71.txt

fddatastr << line;
    fddatastr >> ordinata;
    fddatastr >> soglia;
    fddatastr >> mean;
    fddatastr >> canvas;
    fddatastr >> canale;
    fddatastr >> soglia2;
    fddatastr >> rms;

double sogliaeffettiva=(mean-soglia);

// if(ordinata!=0.01 && ordinata!=0){
//     cout << " ordinata diversa da 0.01: " << ordinata << endl;
```

```
// cout << " canvas: " << canvas << " canale: " << canale << endl;
// }

if(canvas<72){
cout << " soglia effettiva " << sogliaeffettiva << endl;
cout << " canvas " << canvas << endl;

sogliaeffettiva_array[k]=sogliaeffettiva;

fhisto_sogliaeffettiva->Fill(sogliaeffettiva_array[k]);

k++;

}
}
// *****
//          GRAFICO
// *****

TCanvas *c8 = new TCanvas("c8","istogramma soglia effettiva",200,10,600,400);

c8->SetFillColor(10);
c8->SetGrid();

fhisto_sogliaeffettiva->GetXaxis()->SetTitle("istogramma soglia effettiva");
fhisto_sogliaeffettiva->SetFillColor(6);
fhisto_sogliaeffettiva->Draw();

}
```


.2 File di mapping

.2.1 RICH NEW ELECTRONIC CMAD

Bottom Saleve

1

V Roof	110	92	32	100	156	27	39	9	115	89	94	83
N Roof	134	26	23	12	3	13	4	1	17	14	28	31
Fiber	65	66	67	68	69	70	71	72	73	74	75	76
Catch	504	504	504	504	505	505	505	505	505	505	505	505
Num Port	12	13	14	15	0	1	2	3	4	5	6	7
rms>6 ch					0	48			48	48	48	32
										52	52	36

V Roof	114	107	98	90	88	104	87	97	102	96	105	93
N Roof	29	24	20	9	0	10	8	5	11	18	32	22
Fiber	53	78	55	56	57	58	59	60	61	62	63	64
Catch	504	504	504	504	504	504	504	504	504	504	504	504
Num Port	0	1	2	3	4	5	6	7	8	9	10	11
rms>6 ch				0							16	48

V Roof	86	113	85	103	101	108	99	91	117	120	119	118
N Roof	27	21	6	15	16	7	2	97	96	98	25	100
Fiber	77	42	43	44	45	46	47	48	49	50	51	52
Catch	503	503	503	503	503	503	503	503	503	503	503	503
Num Port	4	5	6	7	8	9	10	11	12	13	14	15
rms>6 ch												

¹Il monte *Saleve* e la catena dello *Jura*, in Svizzera, alle pendici dei quali si trova la città di Ginevra, vengono utilizzati in questo frangente come punti d'orientamento della posizione degli apparati elettronici su RICH.

.2.2 RICH NEW ELECTRONIC CMAD**Bottom Jura**

V Roof		79	47	54	7	42	75	158	157	159	160	2
N Roof	122	105	106	107	109	110	111	154	152	151	155	153
Fiber												
Catch	502	502	502	502	502	502	502	502	503	503	503	503
Num Port	8	9	10	11	12	13	14	15	0	1	2	3
rms>6 ch												

V Roof	76	74	45	36	10	15	30	162	162	154	84	116
N Roof	121	149	140	144	148	139	143	147	124	101	103	108
Fiber												
Catch	501	501	501	501	502	502	502	502	502	502	502	502
Num Port	12	13	14	15	0	1	2	3	4	5	6	7
rms>6 ch												

V Roof	77	81	73	18	17	29	51	16	52	31	80	78
N Roof	130	137	135	136	133	123	142	127	125	146	141	150
Fiber												
Catch	501	501	501	501	501	501	501	501	501	501	501	501
Num Port	0	1	2	3	4	5	6	7	8	9	10	11
rms>6 ch												

.2.3 RICH NEW ELECTRONIC CMAD**Top Jura**

V Roof	3	35	43	56	4	139	147	122	67	5	62	57
N Roof	61	66	68	70	71	73	99	132	87	79	80	89
Fiber	25	26	27	28	29	30	31	32	33	34	35	36
Catch	507	507	507	507	507	507	507	507	507	507	507	507
Num Port	0	1	2	3	4	5	6	7	8	9	10	11
rms>6 ch		32		16					0		16	32
									4		48	36

V Roof	26	46	49	12	44	137	143	152	123	68	69	72
N Roof	62	64	67	138	75	85	81	94	82	93	77	91
Fiber	13	14	15	16	17	18	19	20	21	22	23	24
Catch	506	506	506	506	506	506	506	506	506	506	506	506
Num Port	4	5	6	7	8	9	10	11	12	13	14	15
rms>6 ch	32								48		0	0
											4	4
											20	
											32	
											36	

V Roof	40	13	38	53	19	141	109	148	124	65	71	63
N Roof	65	63	76	72	74	86	95	83	92	78	88	90
Fiber	1	2	3	4	5	6	7	8	9	10	11	38
Catch	505	505	505	505	505	505	505	505	506	506	506	506
Num Port	8	9	10	11	12	13	14	15	0	1	2	3
rms>6 ch					0				0		0	32
									32		4	
									48			

.2.4 RICH NEW ELECTRONIC CMAD

Top Saleve

V Roof	60	132	128	152	151	150	144	138	135	24	34	28
N Roof	131	51	55	45	41	42	39	37	33	118	115	112
Fiber	65	66	67	68	69	70	71	72	73	74	75	76
Catch	509	509	509	509	509	509	509	509	509	509	509	509
Num Port	4	5	6	7	8	9	10	11	12	13	14	15
rms>6 ch	32 48		48	0 4						16		0 4 48 52

V Roof	61	130	127	8	155	146	136	133	48	25	6	33
N Roof	59	50	52	54	43	48	46	38	35	119	116	113
Fiber	53	54	55	56	57	58	59	60	61	62	63	64
Catch	508	508	508	508	508	508	508	508	509	509	509	509
Num Port	8	9	10	11	12	13	14	15	0	1	2	3
rms>6 ch		32						0 32			32 36	0 4 32 36

V Roof	131	129	126	153	149	145	142	134	22	41	21	50
N Roof	60	57	49	56	120	47	44	40	36	34	117	114
Fiber	41	42	43	44	45	46	47	48	49	50	51	52
Catch	507	507	507	507	508	508	508	508	508	508	508	508
Num Port	12	13	14	15	0	1	2	3	4	5	6	7
rms>6 ch		0 48						16				

.3 Files di calibrazione con rms > 3

2

(★)	CALIBRATION FILE	CH	MEAN	RMS
	baseline-Gain_0-CMAD/Calibration_100.txt	63	647	3.61
	baseline-Gain_0-CMAD/Calibration_103.txt	63	648	4.18
	baseline-Gain_0-CMAD/Calibration_104.txt	63	652	4.10
	baseline-Gain_0-CMAD/Calibration_106.txt	63	647	3.28
	baseline-Gain_0-CMAD/Calibration_107.txt	63	650	3.95
	baseline-Gain_0-CMAD/Calibration_109.txt	63	644	3.05
	baseline-Gain_0-CMAD/Calibration_110.txt	63	649	3.25
	baseline-Gain_0-CMAD/Calibration_111.txt	63	650	3.34
★	baseline-Gain_0-CMAD/Calibration_113.txt	15	652	3.50
★	baseline-Gain_0-CMAD/Calibration_113.txt	46	652	3.08
★	baseline-Gain_0-CMAD/Calibration_113.txt	63	649	4.53
	baseline-Gain_0-CMAD/Calibration_114.txt	63	651	3.01
	baseline-Gain_0-CMAD/Calibration_115.txt	15	663	4.42
	baseline-Gain_0-CMAD/Calibration_115.txt	63	650	3.15
★	baseline-Gain_0-CMAD/Calibration_116.txt	63	649	3.33
	baseline-Gain_0-CMAD/Calibration_117.txt	1	650	3.02
	baseline-Gain_0-CMAD/Calibration_117.txt	15	650	3.44
	baseline-Gain_0-CMAD/Calibration_117.txt	18	649	3.30
	baseline-Gain_0-CMAD/Calibration_117.txt	31	648	3.31
	baseline-Gain_0-CMAD/Calibration_119.txt	63	649	3.02
	baseline-Gain_0-CMAD/Calibration_120.txt	63	650	3.94
	baseline-Gain_0-CMAD/Calibration_122.txt	16	649	3.06
	baseline-Gain_0-CMAD/Calibration_122.txt	63	651	3.80

²I files con la stellina ★ sono quelli che hanno l'rms anche maggiore di 6 digit.

.3. FILES DI CALIBRAZIONE CON RMS > 3CAPITOLO 4. APPENDICE

(★)	CALIBRATION FILE	CH	MEAN	RMS
	baseline-Gain_0-CMAD/Calibration_124.txt	47	649	3.35
	baseline-Gain_0-CMAD/Calibration_124.txt	63	649	3.95
	baseline-Gain_0-CMAD/Calibration_125.txt	63	650	3.53
	baseline-Gain_0-CMAD/Calibration_130.txt	63	648	3.84
★	baseline-Gain_0-CMAD/Calibration_131.txt	63	647	5.36
	baseline-Gain_0-CMAD/Calibration_133.txt	63	650	4.11
	baseline-Gain_0-CMAD/Calibration_135.txt	63	650	4.03
	baseline-Gain_0-CMAD/Calibration_136.txt	63	646	5.47
	baseline-Gain_0-CMAD/Calibration_137.txt	63	649	3.63
	baseline-Gain_0-CMAD/Calibration_138.txt	63	650	4.03
	baseline-Gain_0-CMAD/Calibration_142.txt	63	656	4.93
	baseline-Gain_0-CMAD/Calibration_143.txt	63	648	3.69
	baseline-Gain_0-CMAD/Calibration_144.txt	16	651	3.66
	baseline-Gain_0-CMAD/Calibration_146.txt	63	649	3.74
	baseline-Gain_0-CMAD/Calibration_147.txt	2	690	7.22
	baseline-Gain_0-CMAD/Calibration_147.txt	63	650	3.00
	baseline-Gain_0-CMAD/Calibration_148.txt	63	650	3.05
	baseline-Gain_0-CMAD/Calibration_149.txt	63	650	3.36
	baseline-Gain_0-CMAD/Calibration_151.txt	63	650	3.28
	baseline-Gain_0-CMAD/Calibration_152.txt	63	651	3.22
	baseline-Gain_0-CMAD/Calibration_153.txt	63	650	4.60
	baseline-Gain_0-CMAD/Calibration_154.txt	63	647	3.45
	baseline-Gain_0-CMAD/Calibration_158.txt	63	650	4.37
	baseline-Gain_0-CMAD/Calibration_160.txt	16	649	3.69
	baseline-Gain_0-CMAD/Calibration_160.txt	63	649	3.98
	baseline-Gain_0-CMAD/Calibration_161.txt	63	649	3.09
	baseline-Gain_0-CMAD/Calibration_162.txt	16	648	4.03
	baseline-Gain_0-CMAD/Calibration_162.txt	32	649	3.07
	baseline-Gain_0-CMAD/Calibration_162.txt	48	649	3.34

CAPITOLO 4. APPENDICE3. FILES DI CALIBRAZIONE CON RMS > 3

(*)	CALIBRATION FILE	CH	MEAN	RMS
	baseline-Gain_0-CMAD/Calibration_162.txt	63	650	3.86
	baseline-Gain_0-CMAD/Calibration_163.txt	63	649	3.43
	baseline-Gain_0-CMAD/Calibration_37.txt	15	646	3.11
*	baseline-Gain_0-CMAD/Calibration_38.txt	63	650	3.11
	baseline-Gain_0-CMAD/Calibration_49.txt	63	649	3.43
*	baseline-Gain_0-CMAD/Calibration_50.txt	63	649	3.88
	baseline-Gain_0-CMAD/Calibration_51.txt	63	651	4.48
*	baseline-Gain_0-CMAD/Calibration_55.txt	63	648	3.21
	baseline-Gain_0-CMAD/Calibration_56.txt	63	647	3.44
*	baseline-Gain_0-CMAD/Calibration_57.txt	63	651	3.42
	baseline-Gain_0-CMAD/Calibration_61.txt	63	650	4.40
*	baseline-Gain_0-CMAD/Calibration_62.txt	0	649	3.40
	baseline-Gain_0-CMAD/Calibration_63.txt	63	647	3.18
*	baseline-Gain_0-CMAD/Calibration_66.txt	63	649	3.94
	baseline-Gain_0-CMAD/Calibration_67.txt	63	650	3.70
*	baseline-Gain_0-CMAD/Calibration_69.txt	63	652	3.47
	baseline-Gain_0-CMAD/Calibration_71.txt	0	650	3.39
	baseline-Gain_0-CMAD/Calibration_71.txt	63	648	5.16
	baseline-Gain_0-CMAD/Calibration_72.txt	63	652	5.98
*	baseline-Gain_0-CMAD/Calibration_73.txt	63	649	4.27
*	baseline-Gain_0-CMAD/Calibration_74.txt	63	650	4.89
*	baseline-Gain_0-CMAD/Calibration_75.txt	63	647	3.26
*	baseline-Gain_0-CMAD/Calibration_76.txt	15	649	3.38
*	baseline-Gain_0-CMAD/Calibration_76.txt	63	648	4.91
*	baseline-Gain_0-CMAD/Calibration_77.txt	63	647	3.19
*	baseline-Gain_0-CMAD/Calibration_80.txt	63	649	4.51
	baseline-Gain_0-CMAD/Calibration_81.txt	63	648	3.24
*	baseline-Gain_0-CMAD/Calibration_82.txt	63	650	5.25
	baseline-Gain_0-CMAD/Calibration_83.txt	63	649	4.75

.3. FILES DI CALIBRAZIONE CON RMS > 3CAPITOLO 4. APPENDICE

(★)	CALIBRATION FILE	CH	MEAN	RMS
	baseline-Gain_0-CMAD/Calibration_85.txt	50	667	3.98
★	baseline-Gain_0-CMAD/Calibration_87.txt	63	650	3.44
★	baseline-Gain_0-CMAD/Calibration_88.txt	63	649	4.39
★	baseline-Gain_0-CMAD/Calibration_90.txt	63	647	3.10
★	baseline-Gain_0-CMAD/Calibration_91.txt	63	650	3.57
	baseline-Gain_0-CMAD/Calibration_94.txt	51	645	3.48
	baseline-Gain_0-CMAD/Calibration_95.txt	63	648	3.27
	baseline-Gain_0-CMAD/Calibration_96.txt	63	650	3.18
	baseline-Gain_0-CMAD/Calibration_97.txt	63	649	3.42
	baseline-Gain_0-CMAD/Calibration_98.txt	63	650	3.25

Bibliografia

- [1] P. Abbon, et al., The COMPASS Experiment at CERN, CERN-PH-EP (2007) 001.
- [2] P. Abbon, et al., The COMPASS Collaboration et al., *Nucl. Instrum. Meth. A*, 577 (2007) 455.
- [3] P. Abbon, et al., A New Analogue Sampling Readout System for the COMPASS RICH-1 Detector, *Nucl. Instrum. Meth., A* 589 (2008) 362-369.
- [4] P. Abbon et al., The Fast Readout System for the MAPMTs of COMPASS RICH-1, *Nucl. Instrum. Meth. A* 587 (2008) 371.
- [5] P. Abbon, et al., The COMPASS RICH-1 Fast Photon Detection System. *Nucl. Instrum. Meth. A*595:23-26, 2008.
- [6] P. Abbon, et al., The Fast Readout System for the MAPMTs of COMPASS RICH-1, *Nucl. Instrum. Meth. A*595:204-207, 2008.
- [7] P. Abbon et al., The COMPASS RICH-1 Detector Upgrade, *Eur. Phys. J.ST* 162 (2008) 251.
- [8] P. Abbon et. al., Particle Identification with the Fast COMPASS RICH-1 Detector, *Nucl. Instrum. Meth. A* (2010), doi: 10.1016/j.nima.2010.02.238.
- [9] P. Abbon et. al., Design and construction of the Fast Photon Detection System for COMPASS RICH-1, *Nucl. Instrum. Meth. A* 616 (2010) 21-37.
- [10] G. Baum et al., COMPASS: A proposal for a Common Muon and Proton Apparatus for Structure and Spectroscopy. CERN-SPSLC-96-14.

- [11] G. Baum et al., The COMPASS *Addendum 1*, CERN/SPSLC 96-30, SPSC/P 297 Add. 1, May 20, 1996.
- [12] G. Baum et al., The COMPASS RICH Project, *Nucl. Instrum. Meth.*, A 443 (1999) 207-211.
- [13] G. Baum et al., The COMPASS RICH-1 Detector, *Nucl. Phys.B (Proc. Suppl.)* 443 (1999) 354-359.
- [14] A. S. Tanenbaum, *Architettura dei calcolatori: un approccio strutturale*, Cap. 2.2.3, Pearson Prentice Hall, 2006.
- [15] V. Barone, *Relatività - Principi e applicazioni*, Cap. 7.13.5, Bollati Boringhieri, Torino, 2004.
- [16] Ö. Çobanolu, et al., "CMAD" a Full Custom ASIC for the Upgrade Of COMPASS RICH-1.
- [17] E. Albrecht et al., COMPASS RICH-1. *Nucl. Instrum. Meth.*, A 504 (2003) 354-355.
- [18] E. Albrecht et al., Results from COMPASS RICH-1. *Nucl. Instrum. Meth.*, A 535 (2004) 448-451.
- [19] E. Albrecht et al., First performances of COMPASS RICH-1. *Nucl. Instrum. Meth.*, A 518 (2004) 586-589.
- [20] E. Albrecht et al., Status and Characterisation of COMPASS RICH-1. *Nucl. Instrum. Meth.*, A 502 (2004) 586.
- [21] E. Albrecht et al., Status and Characterisation of COMPASS RICH-1. *Nucl. Instrum. Meth.*, A 553 (2005) 215.
- [22] F. Bradamante, The COMPASS Experiment at CERN, *Nucl. Phys. A* 662 (1997) c-65c.
- [23] F. Bradamante, The COMPASS Experiment at CERN: present and future, *Nucl. Phys.B (Proc. Suppl.)* 154 (2006) 86-98.