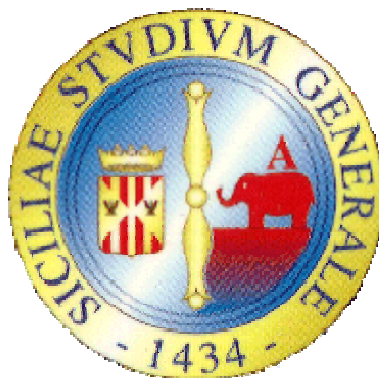


UNIVERSITA' DEGLI STUDI DI CATANIA



FACOLTA' DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA
DIPARTIMENTO DI INGEGNERIA INFORMATICA E DELLE TELECOMUNICAZIONI

TESI DI LAUREA

INTERFACCIA PER LA MICROMOVIMENTAZIONE DI UN SISTEMA ROTOTRASLATORIO

RELATORE:

Chiar.mo Prof. Michele Malgeri

CORRELATORE:

Chiar. Mo Dott. Paolo Finocchiaro

Carlotta Scirè Scappuzzo
616/000760

Anno Accademico 2006/2007

Dedicata ai miei genitori

INDICE

INDICE	3
INTRODUZIONE.....	5
CAPITOLO 1 – Contesto di Lavoro	8
1.1 ALLINEAMENTO.....	8
1.2 SISTEMI DI POSIZIONAMENTO	9
1.3 SISTEMA DI MOVIMENTAZIONE	11
1.4 STRUMENTI HARDWARE E SOFTWARE	13
1.5 ESP 6000 (Enhanced System Performance 6000).....	14
1.6 ERRORI DI STRUMENTAZIONE	17
1.7 INCERTEZZA.....	19
1.8 MISURAZIONI.....	20
1.9 FOTORIVELATORI	26
1.10 FOTODIODI.....	27
1.11 DLI (Deep Litography with Ions-Litografia profonda con Ioni).....	28
1.12 DISPOSITIVI PER DIAGNOSTICA.....	29
1.13 APPLICAZIONI FUTURE E IN CAMPO MEDICO.....	30
CAPITOLO 2 - LabVIEW	32
2.1 GENERALITA' SUL LabVIEW.....	32
2.2 ELEMENTI PRINCIPALI	36
2.2.1 Strutture	37
2.2.2 Variabili locali	39
2.2.3 Cluster	42
CAPITOLO 3 – Applicazione SW.....	45
3.1 PULSANTI ESTERNI.....	47
3.2 MOTOR ARRAY	47
3.3 AXIS ARRAY	49
3.4 BIT DI CONTROLLO.....	49
3.5 GRAFICO CAMPIONE	50
3.6 POSSIBILI APPLICAZIONI	53
3.6.1 Settare la posizione zero	53
3.6.2 Settare una nuova posizione	53
3.7 ARRAYS DELLE POSIZIONI.....	53
3.8 PROBLEMATICHE DEL TARGET.....	54
3.9 GRAFICO DEL COLLIMATORE	56
3.10 FILE DI INIZIALIZZAZIONE	57
3.11 CODICE.....	58
3.12 PAGINA 0	59
3.13 PAGINA 1	64
3.14 PAGINA 2	68
3.15 PAGINA 3	82

CAPITOLO 4 – Turno di Misura	85
4.1 PROPOSTA DI LAVORO	85
4.2 SETUP TURNO DI MISURA.....	86
4.3 ALLINEAMENTO	87
4.4 SFOP	90
4.5 SISTEMA DI MOVIMENTAZIONE	92
4.6 COLLEGAMENTO DA REMOTO	95
4.7 IRRAGGIAMENTO.....	96
4.8 PRIMO RIVELATORE.....	96
4.9 SECONDO RIVELATORE	100
4.10 INDUZIONE	103
 CONCLUSIONI	 107
 APPENDICE.....	 109
A.1 L’ACCELERATORE DI PARTICELLE: IL TANDEM	109
A.2 CAMERE A VUOTO, LA CAMERA 2000.....	111
 RINGRAZIAMENTI	 112
 BIBLIOGRAFIA	 113

INTRODUZIONE

Nel vasto campo della sperimentazione fisica, la diagnostica dei fasci di particelle si colloca in una posizione di rilievo, sia per i vantaggi che per le potenzialità che derivano dall'utilizzo di tecniche che ne fanno uso.

La presente tesi tratta dello sviluppo di un'applicazione software per la micro-movimentazione rototraslatoria di un sistema per l'irraggiamento di componenti elettronici e/o meccanici con fasci di particelle di alta energia, con particolare riguardo allo sviluppo e test di un prototipo di rivelatore al silicio per particelle cariche.

Il rivelatore, progettato e realizzato da un gruppo dei Laboratori Nazionali del Sud dell'Istituto Nazionale di Fisica Nucleare (LNS-INFN) in collaborazione con la ST Microelectronics di Catania, è in grado di produrre segnali in corrispondenza del passaggio di una particella carica.

L'analisi di tali segnali, una volta digitalizzati, dovrebbe permettere di identificare lo ione incidente e di determinare la posizione xy di impatto della particella stessa oltre che fornire importanti informazioni circa lo studio del rivelatore in base alla localizzazione spaziale dell'incidenza del fascio sulla sua superficie.

L'obiettivo dell'applicazione software sviluppata è stato quello di movimentare un sistema di posizionamento micrometrico su cui è stato posto il campione, fornendo un'interfaccia semplice e funzionale all'utente per renderlo in grado di pilotare il sistema durante la caratterizzazione del rivelatore con un fascio di ioni prodotto dall'acceleratore Tandem, presso i LNS-INFN.

La caratterizzazione consiste nell'irraggiamento del rivelatore con particelle note su posizioni predeterminate, e nella successiva analisi dei segnali prodotti, digitalizzati e registrati in maniera da verificare le proprietà previste del rivelatore stesso.

La problematica principale è data dal fatto che si parla di irraggiamenti di precisione su superfici estremamente ridotte, con spostamenti molto piccoli, per cui è necessaria una precisione micrometrica adeguata.

Nel corso della seguente tesi si cercherà di dare una visione generale delle tematiche che si sono affrontate, delle problematiche di controllo delle apparecchiature utilizzate che lavorano ad alta precisione e la loro applicazione per il posizionamento e la movimentazione del sistema in esame, degli strumenti hardware e software in dotazione e degli errori di strumentazione a cui sono sottoposti i dispositivi fisici, per i quali si sono effettuate misurazioni al fine di testare l'affidabilità del sistema trovando un riscontro in alcuni risultati sperimentali.

Inoltre si darà una panoramica dei campioni oggetto di studio, del loro funzionamento e di altre tecniche che adottano l'utilizzo di un fascio di particelle con alcune possibili applicazioni in molti altri campi oltre a quello fisico, come nel campo medico in cui l'utilizzo di particelle accelerate riscuote molto successo e accende un interesse sempre crescente verso l'applicazione della fisica avanzata a scopo terapeutico.

Introducendo le implementazioni software e l'ambiente di sviluppo scelto, il LabView, con le sue caratteristiche, si analizzerà la soluzione sw che si è proposta per affrontare il problema della movimentazione mettendo in rilievo lo studio dell'interfaccia utente, le considerazioni per una maggiore semplicità di uso e per massimizzare la quantità e la qualità delle informazioni fornite; infine verrà spiegato dettagliatamente il programma nelle sue funzionalità aprendo una finestra su possibili applicazioni future.

Una considerazione di rilievo merita l'esperienza del turno di misura che ha visto con l'applicazione del programma un riscontro tangibile delle funzionalità del programma stesso ed è stata un'esperienza estremamente formativa e stimolante perché mi ha permesso di entrare in contatto con strumentazioni e tecniche operative che hanno sempre esercitato un certo fascino su di me come su tutte quelle persone che

trovano nel mondo scientifico e soprattutto fisico una particolare attrattiva.

Il turno di misura si è svolto in sede dei LNS-INFN da giorno 2 dicembre a giorno 8 dicembre 2006 e hanno partecipato professionisti del settore che con la loro esperienza nel campo di altri esperimenti hanno contribuito in modo rilevante all'apprendimento dei vari aspetti tecnici e operativi dell'esperimento oltre che suscitare in me un interesse su ciò di cui ci si sarebbe dovuti occupare e ciò che ci si sarebbe aspettato dai test.

I risultati che si sono evinti a seguito dell'analisi dei segnali di risposta dei campioni sono stati soddisfacenti e verranno presto pubblicati.

Il lavoro è strutturato in quattro capitoli così organizzati:

- Il primo capitolo (Contesto di Lavoro) tratta dell'ambiente in cui si colloca la problematica.
- Il secondo capitolo (LabVIEW) introduce l'ambiente di sviluppo sw, il LabVIEW con il quale ho lavorato.
- Il terzo capitolo (Applicazione SW) descrive interamente il programma proposto, dall'interfaccia utente al codice grafico utilizzati.
- Il quarto capitolo (Turno di Misura) è dedicato all'esperimento in cui è stato utilizzato il programma.

Infine saranno esposte le conclusioni e le possibili future applicazioni del programma sviluppato.

CAPITOLO 1 – Contesto di Lavoro

Nel presente capitolo tratta del contesto nel quale si è svolto il progetto introducendo il problema pratico principale dell'applicazione, cioè l'allineamento del sistema e il giusto posizionamento del campione per poi passare alla panoramica dei sistemi e degli strumenti di cui si è fatto uso durante il periodo di tirocinio presso i LNS-INFN e delle nozioni teoriche a cui si è fatto riferimento per avere una visione più chiara della problematica da affrontare e per agire con le giuste conoscenze.

Infine una visione generale delle applicazioni che prevedono l'utilizzo di fasci di particelle accelerate, dallo studio di particolari rivelatori di fonti luminose estremamente sensibili, a tecniche per la creazione di componenti tecnologici di alta precisione, a dispositivi per lo studio di fasci, fino ad arrivare alle possibili applicazioni future.

1.1 ALLINEAMENTO

Le tecniche che adottano l'utilizzo di un fascio devono essere in grado di eseguire un'irradiazione accurata di un campione con un posizionamento ad alta precisione della porzione di fascio. Array di microlenti e di rivelatori per comunicazioni intercircuitali, array di microfori per connettori in fibra ottica 2D fanno riferimento ad una circonferenza del fascio di 200 μm , 125 μm e a volte ancora meno, ad esempio nel nostro caso il diametro del fascio che incide sul campione è di circa 50 μm x 100 μm .

Se vogliamo una precisa incidenza di queste particelle ad alta energia sul campione, il fascio deve essere parallelo il più possibile all'asse di riferimento e perpendicolare al campione.

Il modo più facile per avere questa condizione è diminuire il diametro del fascio di ioni e la sua divergenza utilizzando collimatori posti lungo la linea del fascio.

I collimatori consistono in fori dal diametro minore rispetto a quello del fascio e fanno passare solo la porzione di fascio che incide sul foro.

I collimatori inclinati o disallineati avranno come risultato una sezione più o meno ellittica del fascio, così le microlenti, i fori e le altre strutture ottenute risentiranno di un calo di efficienza e funzionalità nel caso in cui sia richiesta una perfetta circolarità del diametro; nel caso di studio preso in esame nella presente tesi ciò non è richiesto perché si preferisce ottenere un'alta precisione nella posizione del fascio, ma non necessariamente nella sua forma.

Quindi il campione e i collimatori necessitano di un sistema di posizionamento multiassiale accurato e ben controllato.

I sistemi di posizionamento micrometrici richiedono una particolare attenzione nel design e nella scelta dei materiali.

Un set di stage di traslazione dedicati è controllato per la corretta posizione del campione fissato sulla piastra.

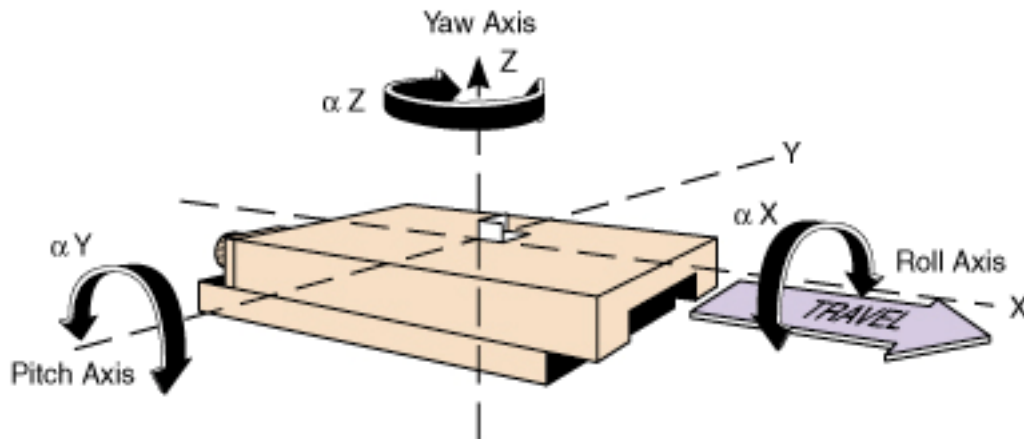
E' possibile l'irradiazione di un singolo punto ma anche l'irradiazione di strutture più complicate come linee muovendo il campione durante il processo di irradiazione con piccoli passi micrometrici.

1.2 SISTEMI DI POSIZIONAMENTO

Il posizionamento accurato del campione richiede una rigida costruzione di due o più stage di traslazione. Il fascio perpendicolare al campione e alla maschera implica che queste due devono essere parallele tra loro. Il parallelismo e la perpendicolarità del fascio possono essere cambiati sincronizzando i parametri di sistema dell'acceleratore di particelle Tandem [Appendice] e delle lenti magnetiche, mentre il parallelismo del campione con la maschera può essere impostato tramite gli stage di traslazione, rotazione e tip & tilt (inclinazione e ribaltamento) con i loro rispettivi gradi di libertà .

Ogni oggetto in movimento ha sei gradi di libertà disponibili: tre lungo i tre assi perpendicolari X, Y e Z e tre rotazioni attorno ognuno di questi tre assi αX , αY , αZ (fig 1.1).

Fig 1.1



Più stage sono usati e più gradi di libertà sono introdotti, più possibilità avrà l'utente di allineare individuali argomenti sulla linea del fascio.

Sfortunatamente questo introduce anche una maggiore varietà di errori di posizione, alcuni dei quali non possiamo compensare.

Due sono gli errori da considerare: il coseno, dovuto ad un disallineamento angolare tra il moto della tavola di posizione e l'accuratezza che determina l'elemento e l'errore Abbe, derivato dalla lunghezza finita dello stage.

La situazione diventa particolarmente evidente per un sistema multiassiale, dove l'insieme degli errori degli stage causa un peggioramento maggiore che nei sistemi a stage singolo.

Ci sono due possibili soluzioni per risolvere questo problema: un sistema a ciclo chiuso con encoder da posizionamento oppure un sistema chiamato Hexapod.

In generale gli stage di posizionamento sono montati su superfici piane di sezioni trasversali, bisogna fare attenzione che la superficie sia pulita e che nessuna particella di polvere sia presente sotto lo stage.

Quando gli assi lineari sono montati tra loro l'ordine di impilatura è meno importante, ma ciò cambia quando le tavole lineari e di rotazione sono mischiate.

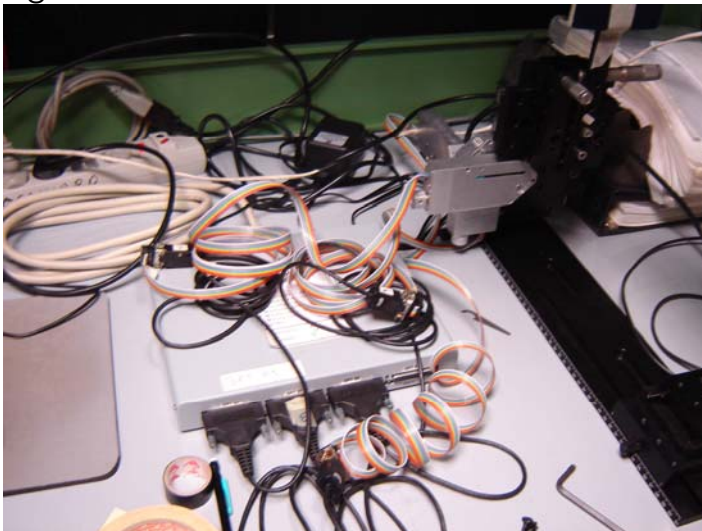
[Pedro Vynck, 2004]

1.3 SISTEMA DI MOVIMENTAZIONE

Un sistema di movimentazione controllato è costituito da: controllore, driver, vari dispositivi di movimentazione (motion device) e i cavi che servono per interconnettere i vari componenti.

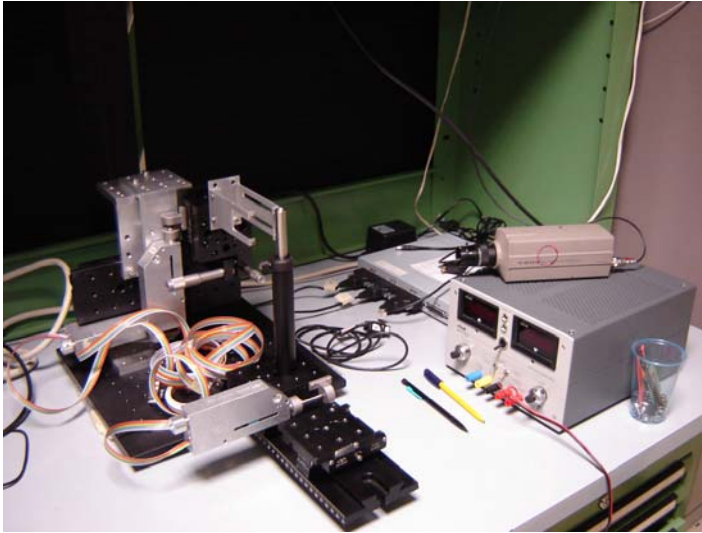
- Il controllore è un dispositivo elettronico che riceve comandi per movimentazione direttamente da un operatore o attraverso un computer, verifica la posizione reale del dispositivo e genera i segnali di controllo necessari.
- Il driver è un dispositivo elettronico che converte i segnali di controllo nel corretto formato supportato dai motorini e con la potenza necessaria per 'guidarli'.
- I dispositivi di movimentazione, i motorini o attuatori, sono dispositivi elettro-meccanici che possono muoversi ed essere caricati con le specifiche necessarie.

Fig 1.2



In figura 1.2 il controllore ESP 6000 con i 4 drivers e tre attuatori collegati tramite connettori e cavi.

Fig 1.3



In figura 1.3 l'intero sistema di movimentazione con attuatori e piastre, o stage lineari serie High Performance Low-Profile Crossed-Roller Bearing, modelli 426 e 436 con supporti a cilindri trasversali ai quali si possono collegare gli attuatori.

[Newport, 2006/2007]

L'utente inizia selezionando un motion device che risponde a certe specifiche di un'applicazione. Dopo si sceglie il controllore che può soddisfare le caratteristiche di movimentazione richieste.

La cosa importante da capire è che non importa come sembrano i componenti o quali sono le loro individuali specifiche, ciò che si vuole è essere sicuri che insieme essi svolgano con affidabilità il loro compito.

Questo vuol dire che un componente è buono in base a come si comporta nel sistema.

[Newport Corporation, 2000]

1.4 STRUMENTI HARDWARE E SOFTWARE

Gli strumenti hardware a disposizione sono : un controllore con 4 drivers indipendenti per 3 attuatori, due per le traslazioni lungo gli assi x e y del campione e uno per lo spostamento orizzontale del collimatore.

Gli attuatori sono della serie 850G, uno di questi è serie 850-HS, cioè a maggiore velocità. Questi attuatori sono alimentati da un motore DC a bassa inerzia per fornire un movimento dolce e scorrevole con basso rumore acustico e meccanico. La risoluzione sub-micrometrica è ottenuta con una vite madre di acciaio inossidabile arrotolata con precisione e lucidata elettricamente, guidata attraverso un meccanismo di scatto all'indietro (rinculo), con un passo di 0,8 mm.

Gli attuatori Serie 850G incorporano molte caratteristiche, come la variazione dei limiti di escursione e l'azionamento manuale.

Gli attuatori a velocità standard 850G, così come quelli più veloci serie 850HS, possono operare in modo ottimale anche in condizioni di vuoto.

L'interfaccia compatibile con gli attuatori Serie 850G è l'ESP Series Motion Controller/Drivers; l'attuatore 850G ha un piccolo chip dentro, che configura l'ESP controller/driver settando i parametri di default.

Quindi l'attuatore 850G è "pronto all'uso" (plug and play) per l'ESP controller/driver.

[Newport Corporation, 1999]

Il dispositivo di interfaccia, o DC Interface Box (DCIB), è utilizzato principalmente per motorini DC di piccole dimensioni che possono essere guidati direttamente dai moduli dell'ESP6000 Motion Controller o dai moduli DC del DCIB stesso.

Un'importante caratteristica del DCIB è il supporto per una manopola per il posizionamento e la correzione manuale.

[Newport Corporation, 1998]

Un primo approccio software per movimentare il sistema è stato il programma ESP 6000[] da cui si è partito per le prime misurazioni e da cui si sono presi degli spunti per lo sviluppo del programma futuro.

Lo scopo principale è stato quello di ricreare uno strumento sw di altrettanta potenza e flessibilità, ad alte prestazioni pur mantenendo un'interfaccia semplice.

Una volta scelto l'ambiente di sviluppo su cui lavorare, il Labview[], un altro importante contributo è stato dato da un programma per la movimentazione di un sistema per la creazione di elementi optoelettronici [Pedro Vynck, 2004], il cui studio ha fornito una base fondamentale da cui partire per portare modifiche e aggiungere funzionalità e controlli specifici in base alle esigenze riscontrate.

1.5 Enhanced System Performance 6000 (ESP 6000)

L'ESP6000 utilities è un programma Windows a 32 bit che permette all'utente di configurare e utilizzare facilmente sistemi per la movimentazione. Il programma garantisce massima flessibilità nell'impostazione e nelle operazioni grazie ad un menù guidato.

ESP6000 Windows utilities, chiamato anche ESP-util.exe, offre agli utenti proprietà e strumenti vantaggiosi tra cui l'abilità nell'utilizzare tutti e sei gli assi di moto, un'interfaccia grafica per servo-regolazione, schermi separati per jog (piccoli spostamenti) e movimento ciclico e per lo stato della posizione, capacità di monitorare il convertitore analogico/digitale etc.

ESP6000 è stato creato per quegli utenti con poca esperienza nel campo del controllo del moto e con media capacità di programmazione.

Gli utenti possono creare le loro movimentazioni desiderate con il controllore scrivendo un'applicazione software che richiama svariati comandi Application Programming Interface (API) contenuti in una Dynamic Link Library (DLL) fornita con il controllore.

Questa lista di comandi API di ESP6000 è stata sviluppata per rispondere ad una grande richiesta di movimentazioni come il semplice spostamento punto-punto o l'interpolazione lineare/circolare e il contorno.

ESP6000 ha un'architettura unica che abilita la comunicazione direttamente con l'host tramite il suo databus; il risultato è che l'utente

può mandare comandi binari o comandi ASCII al controllore tramite un'interfaccia PCI. L'utente può sviluppare la propria applicazione sw con programmi come VisualC/C++, Visual Basic o LabView, che emettono questi comandi al controllore tramite la DLL (dall'host).

Quando il sistema è inizializzato, la DLL fa una chiamata al device driver per aprire la comunicazione, questo risponde con un indirizzo di locazione di memoria.

Le librerie fornite devono essere utilizzate con Windows 95 e Windows NT. L'utente può fare una chiamata alla DLL allo stesso modo in cui potrebbe fare con qualsiasi altra funzione. La libreria dovrebbe essere linkata al programma dell'utente o utilizzando la funzione LoadLibrary() fornita dall'API Windows 95, oppure usando la libreria inclusa con la DLL.

[Newport Corporation, 2000]

L'architettura ESP consiste di: ESP6000 controller card, UniDrive 6000 universal motor driver e gli stage ESP compatibili.

L'ESP6000 controller card è designata per un'installazione conveniente sul PC dell'utente.

L'utilità Windows-based di ESP6000 fornisce un insieme pieno di funzioni per configurare e operare sugli assi, da uno a sei.

Il sistema è creato per operare con gli stage della Newport Corporation compatibili all'ESP, ma può essere configurato per funzionare con altri stage.

Le caratteristiche che rendono l'ESP particolarmente funzionale sono:

- Acquisizione di dati combinati e controllo della movimentazione.
- Controllore, drivers e setup degli stage "pronti all'uso".
- Configurazione per diversi tipi di motori: DC e stepper.
- Algoritmo di servo regolazione per un movimento fluido e preciso.
- Sincronizzazione multiasse.
- Potenti capacità di movimentazione in programmazione con linguaggi come Visual Basic, C e LabVIEW.
- Set di comandi forniti direttamente da Newport Corporation.

- Unità di spostamento selezionabili dall'utente.
- Configurazione per UniDrive 6000.

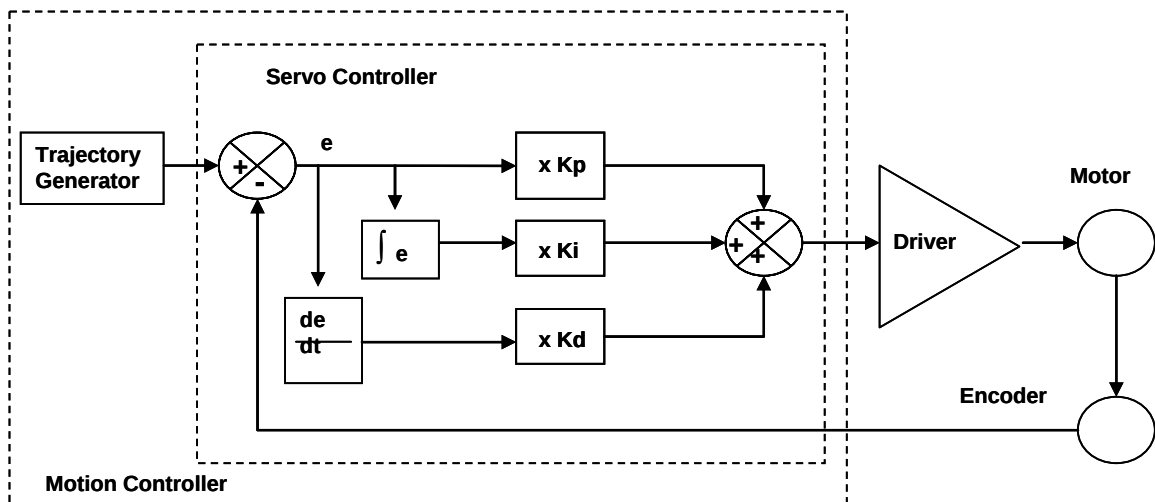
L'ESP6000 controller card utilizza due tipi di cavi di interfaccia, quello da 100 a 100 pin per connettersi ad UniDrive e quello da 100 a 68 pin per connettersi alla DCIB.

[Newport Corporation, 1998]

Il controllore ESP6000 ha un blocco funzionale PID per il funzionamento a ciclo chiuso, un sistema retroazionato che permette di minimizzare gli errori di posizionamento(fig 1.4).

Servo Tuning (sintonizzatore) setta i parametri proporzionali (K_p), integrativi (K_i), derivativi (K_d), accelerativi e decelerativi dell'algoritmo digitale PID, chiamato filtro PID.

Fig 1.4



Per avere le migliori prestazioni dinamiche possibili il sistema deve essere regolato per la specifica applicazione: caricamento, accelerazione, orientamento dell'apparato e prestazioni richieste.

Servo Tuning è generalmente utilizzato per ottenere migliori performance nella movimentazione o perché il sistema è malfunzionante.

Possiamo cambiare i parametri del PID e visualizzare gli effetti, come menzionato nel capitolo .

[Newport Corporation, 2000]

1.6 ERRORI DI STRUMENTAZIONE

L'obiettivo principale del controllore è assicurarsi che il dispositivo in moto segua il più possibile una traiettoria ideale nel tempo.

Nella realtà il dispositivo devierà dalla traiettoria ideale.

L'errore istantaneo, cioè la differenza tra la posizione attuale (indicata dalla posizione del feedback del dispositivo) e la posizione ideale, si chiama ' following error '.

L'errore invece è la differenza tra la posizione ideale e la posizione indicata da un dispositivo di misurazione di precisione esterno, quindi è l'errore percepito dall'utente.

[Newport Corporation, 2000]

Ecco alcuni parametri definiti nella *Norma UNI 4546*[UNI 4546]:

Accuratezza, è una misura statistica dell'errore di posizionamento punto a punto. La differenza tra il punto più alto e quello più basso è il maggiore errore che il dispositivo può avere, questa è l'accuratezza.

Sensibilità, capacità dello strumento di indicare quando cambia una misura.

Risoluzione, capacità dello strumento a risolvere stati diversi del misurando, cioè è il più piccolo incremento di posizione che il dispositivo può attuare.

Ripetibilità, è un errore relativo, attitudine di uno strumento a fornire valori di lettura poco differenti tra loro, in letture consecutive eseguite sullo

stesso misurando, dallo stesso operatore, nelle stesse condizioni per le grandezze d'influenza.

Il valore della ripetibilità è l'intervallo di valori di lettura entro il quale si prevede cade una percentuale assegnata dei valori di lettura.

Se non diversamente indicato, si intende che la percentuale assegnata a definire il valore della ripetibilità sia il 95%.

Isteresi, proprietà dello strumento di fornire valori di lettura diversi in corrispondenza di un medesimo misurando quando questo viene fatto variare per valori crescenti e per valori decrescenti.

Minimo incremento di posizione, è il più piccolo spostamento che il dispositivo può fare nella realtà, misurato da un dispositivo di precisione esterno.

Una volta definito questo parametro, si deve quantificare e non è molto semplice per la presenza dell'errore intrinseco di un piccolo spostamento. Infatti non è facile trovare una linea di separazione tra valori accettabili e valori non accettabili. Il valore più comune del massimo errore accettabile per piccoli spostamenti è il 20% secondo il Motion Control Tutorial [Newport Corporation, 2000], ma ogni applicazione ha i suoi standard.

Un modo di risolvere il problema è fare un grande numero di misurazioni (almeno qualche centinaio) per ogni grandezza di spostamento e rappresentare queste misurazioni in modo da permettere all'utente di determinare il minimo incremento di posizione secondo il suo standard, per esempio con l'uso di grafici.

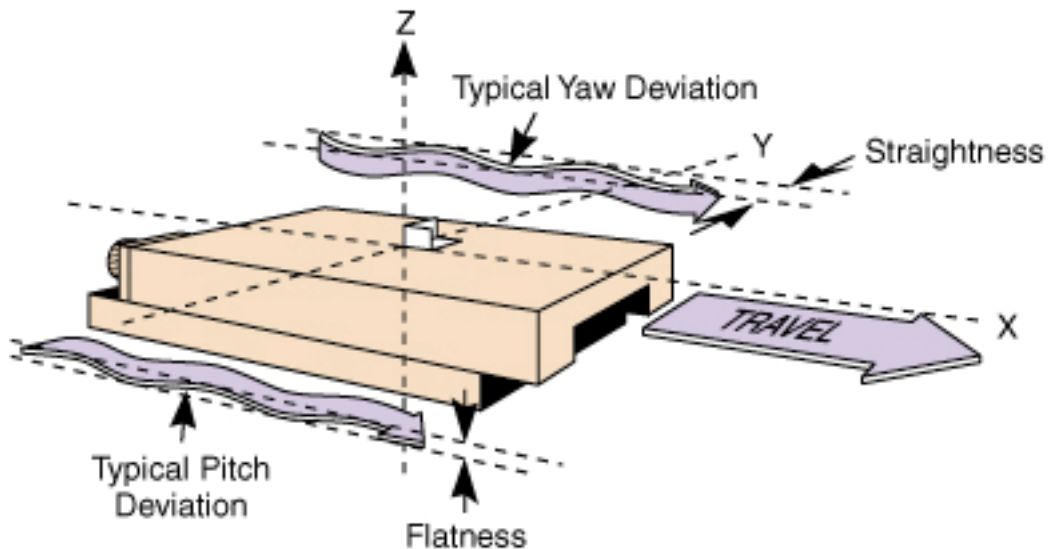
[Newport Corporation, 2000]

Pitch, Roll e Yaw sono i più comuni errori angolari associati agli stage di traslazione lineari, sono errori puramente meccanici e rappresentano l'errore rotazionale di un carico lungo i 3 assi (fig 1.5).

Uno stage perfetto non ruota attorno nessuno degli assi, quindi questi parametri dovrebbero essere zero.

[Newport Corporation, 2000]

Fig 1.5



1.7 INCERTEZZA

Con il termine 'incertezza di misurazione' si intende il dubbio circa la validità del risultato di una misurazione, cioè caratterizza la dispersione dei valori ragionevolmente attribuiti al misurando.

Il risultato di una misurazione è solo un'approssimazione o stima del valore del misurando ed è pertanto completo solamente quando sia accompagnato da una dichiarazione dell'incertezza di quella stima.

In molti casi il risultato di una misurazione è determinato sulla base di serie di osservazioni ottenute in condizioni di ripetibilità.

[ISO]

Nella maggioranza dei casi, la migliore stima dei valori di aspettazione di una grandezza q che varia casualmente, è la *media aritmetica* o *valore medio* $\tilde{q}$ delle n osservazioni:

$$\tilde{q} = \frac{1}{n} \sum q_k$$

dove q_k sono le n osservazioni indipendenti.

Le singole osservazioni q_k differiscono a causa di variazioni casuali delle grandezze d'influenza. La **varianza sperimentale** delle osservazioni è data da:

$$s^2(q_k) = \frac{1}{n-1} \sum (q_k - \tilde{q}_k)^2$$

Questa stima della varianza e la sua radice quadrata positiva $s(q_k)$, denominata **scarto tipo sperimentale**, caratterizzano la variabilità dei valori osservati q_k o la dispersione intorno alla media $\tilde{q}$.

[ISO]

Si definisce **errore assoluto** di misura la differenza tra il valore misurato della grandezza e quello vero:

$$e = Gm - Gv$$

La sola conoscenza dell'errore assoluto non è sufficiente per stabilire il grado di precisione della misura; assume fondamentale importanza l'**errore relativo** definito come il rapporto tra l'errore assoluto e il valore misurato della grandezza:

$$\zeta = \frac{e}{Gm}$$

1.8 MISURAZIONI

Gli strumenti in dotazione dei LNS sono due attuatori lineari Newport serie 850-G e uno serie 850-HS con le seguenti caratteristiche:

	850-G (Standard)	850-HS (High Speed)
<i>Risoluzione Encoder</i>	0,05101 μm	0,60514 μm
<i>Rapporto meccanico</i>	262:1	22:1
<i>Velocità massima</i>	500 $\mu\text{m/s}$	6000 $\mu\text{m/s}$

Sui suddetti attuatori si è eseguita una serie di misurazioni per determinare l'affidabilità del sistema; per ognuno sono state effettuate 100 letture per

spostamenti di 5 mm e 3 mm con l'ausilio di un calibro a cursore con risoluzione di 50 μm .

Fig 1.6



In figura 1.6 il calibro a cursore utilizzato durante l'intera misurazione e una delle piastre utilizzate.

Dallo studio statistico dei valori ottenuti è stato possibile valutare errori percentuali e risoluzioni di maggiore precisione dello strumento utilizzato per la misurazione.

	850-G (Standard)	850-HS (High Speed)
<i>Valore medio di lettura (5 mm)</i>	4,993333 mm	4,998 mm
<i>Varianza sperimentale</i>	$0,428 \cdot 10^{-3} \text{ mm}$	$0,119 \cdot 10^{-3} \text{ mm}$
Scarto tipo	$0,02068415 \text{ mm}^2$	$0,010928 \text{ mm}^2$
Errore %	0,413683	0,218556

Errore %	0,461074	0,350075
<i>Valore medio di lettura (3 mm)</i>	2,9985 mm	2,997833 mm
<i>Varianza sperimentale</i>	$0,191 \cdot 10^{-3} \text{ mm}$	$0,110 \cdot 10^{-3} \text{ mm}$
Scarto tipo	$0,01383221 \text{ mm}^2$	$0,010502 \text{ mm}^2$

L'attuatore serie 850-G risulta essere meno accurato rispetto quello serie 850-HS dopo avere analizzato i risultati ottenuti per spostamenti millimetrici; ma si è anche osservato che al ridursi della distanza il primo abbia un maggiore incremento dell'accuratezza rispetto al secondo.

Per ogni attuatore e per ogni ciclo di misure sono stati calcolati tre grafici con un foglio di lavoro Excel:

il primo riguardante le letture statistiche del misurando;

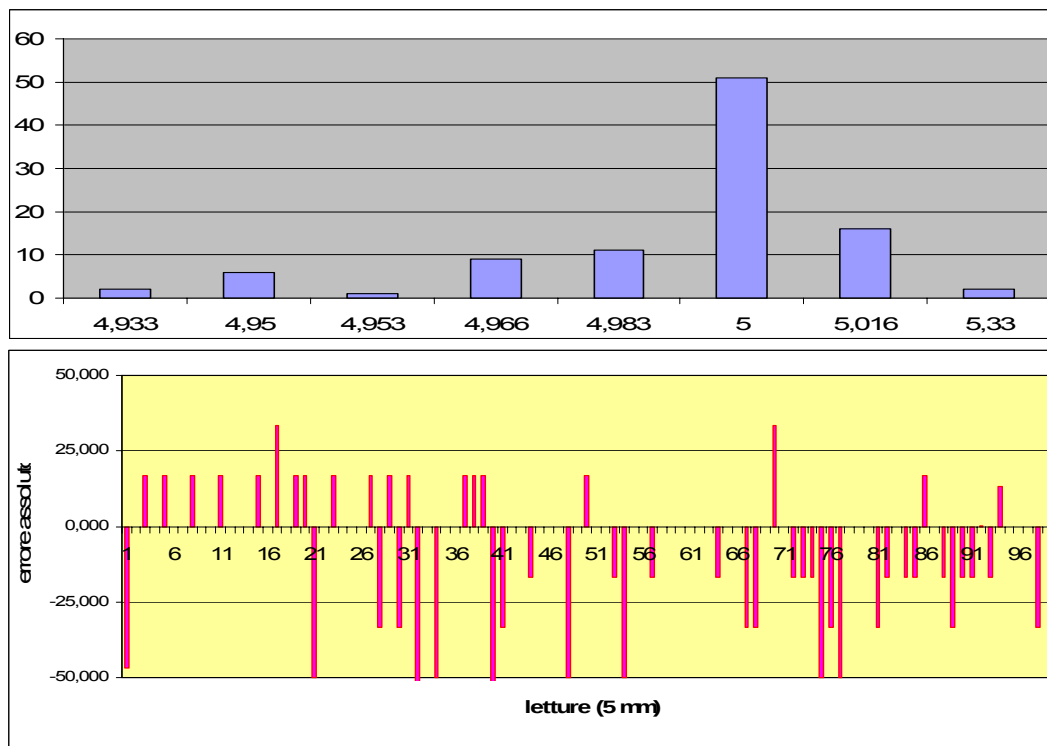
il secondo esprime l'Errore assoluto dato dalla differenza tra il valore letto e il valore ideale:

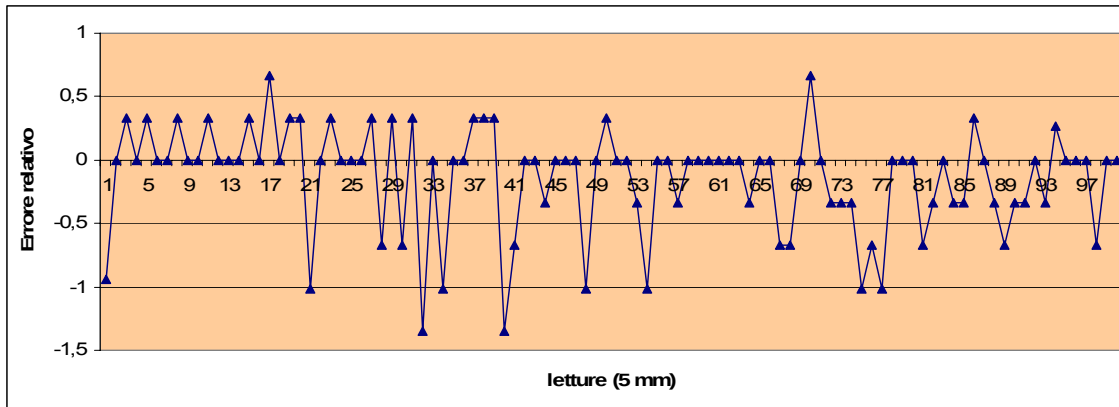
$$Err_{abs} = q_k - q;$$

il terzo rappresenta l'Errore relativo dato dal rapporto tra errore assoluto e il valore letto in %:

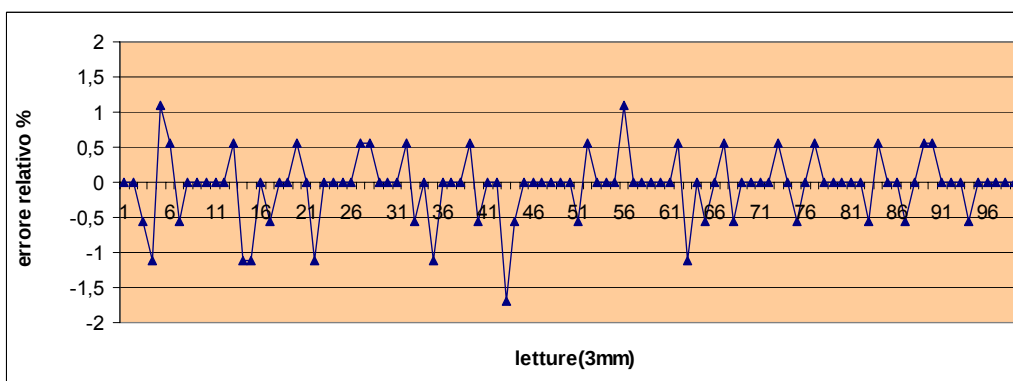
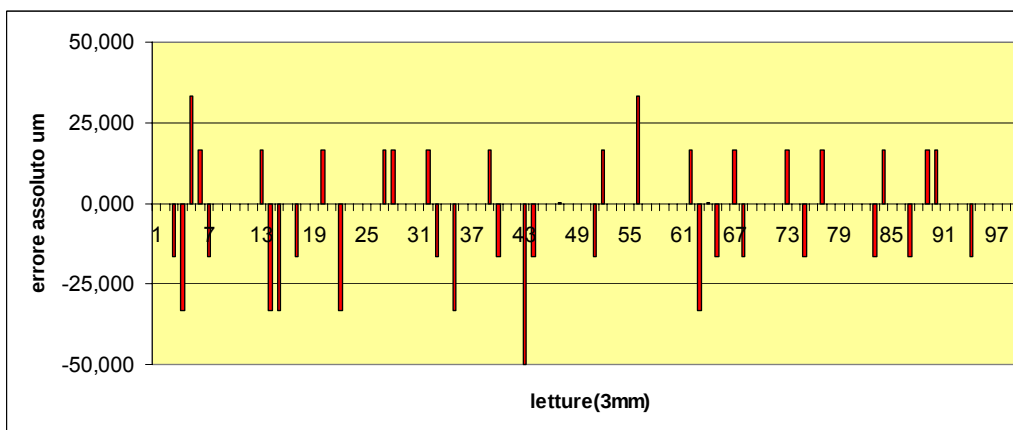
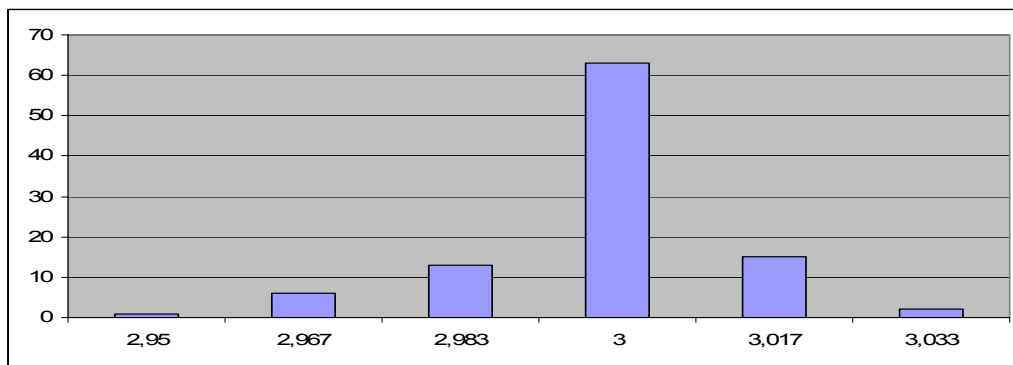
$$Err_{rel} = \frac{Err_{abs}}{q_k} * 100;$$

Di seguito i risultati statistici ottenuti per uno spostamento di 5mm con un attuatore serie 850-G:

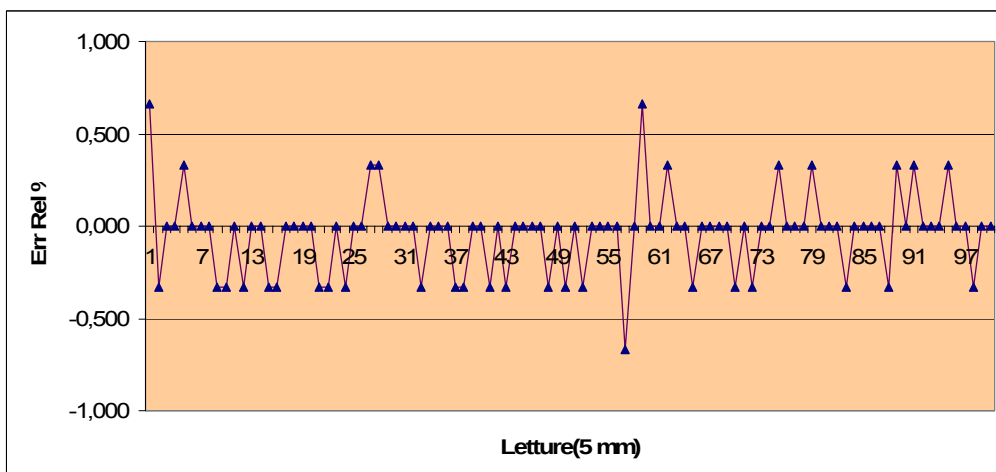
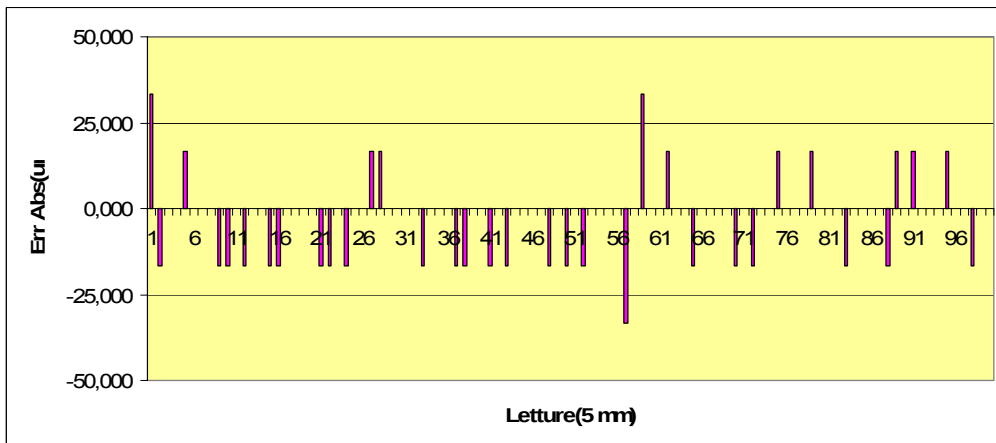
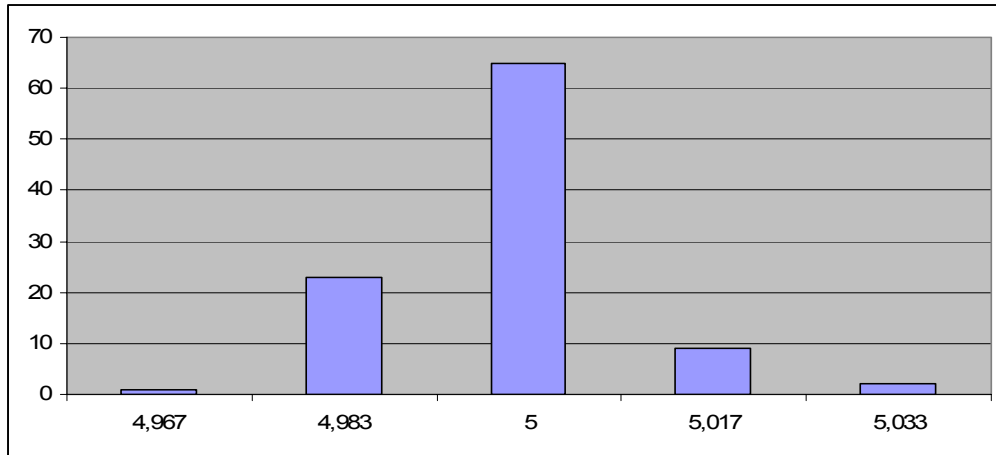




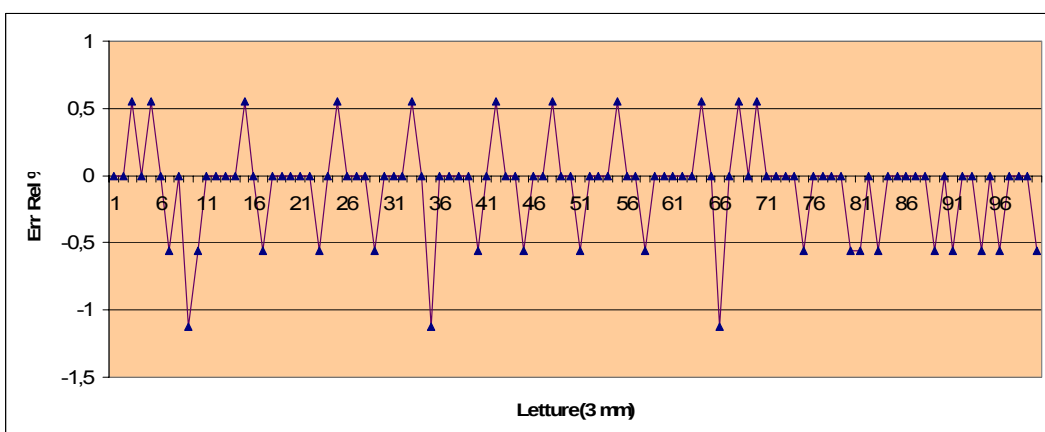
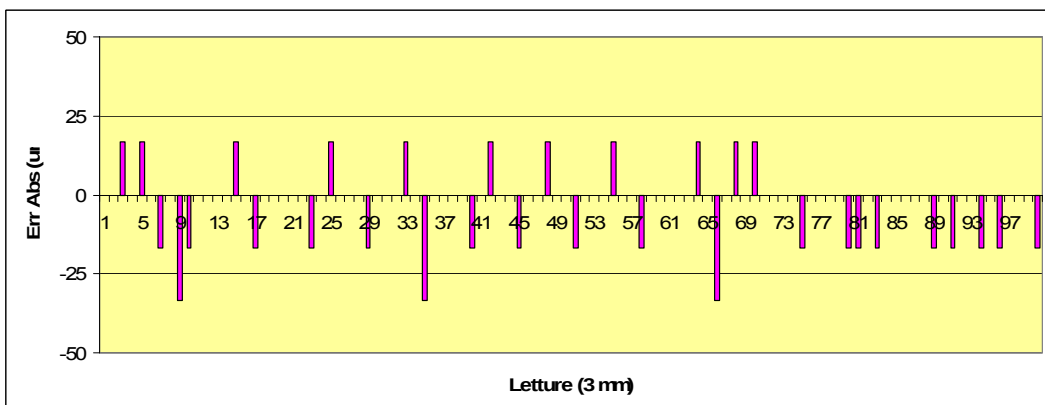
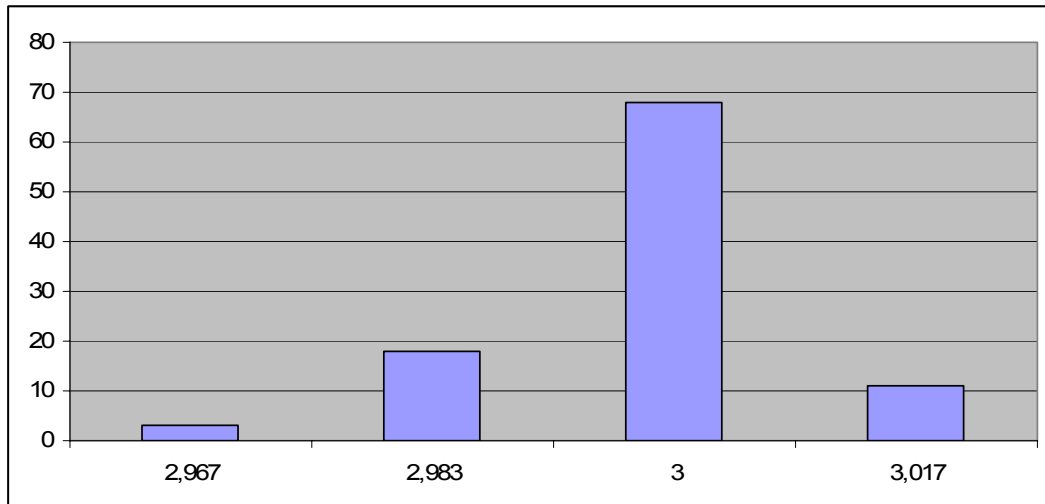
I seguenti grafici mostrano i dati raccolti per 100 misurazioni effettuate su uno spostamento di 3 mm per l'attuatore serie 850-G:



I seguenti grafici mostrano i dati raccolti per 100 misurazioni effettuate su uno spostamento di 5 mm per l'attuatore serie 850-HS:



I seguenti grafici mostrano i dati raccolti per 100 misurazioni effettuate su uno spostamento di 3 mm per l'attuatore serie 850-HS:



1.9 FOTORIVELATORI

Numerosi progressi nel campo della fotorivelazione hanno permesso la creazione di dispositivi in grado di fornire altissime prestazioni nella rivelazione di fotoni in tutto lo spettro luminoso.

I fotorivelatori sono sensori in grado di rivelare le radiazioni luminose con applicazione in molti settori scientifici come astrofisica, comunicazioni spaziali, medicina, fisica nucleare, telecomunicazioni, ottica, fotonica etc.

Oltre alle sorgenti di luce diretta come i laser, i fotorivelatori analizzano anche luce prodotta per interazioni di particelle (elettroni, protoni, raggi x, raggi gamma, raggi beta, neutroni).

Il fotorivelatore conserva varie informazioni riguardo all'energia emessa dalla sorgente luminosa, come intensità e tempo di arrivo dell'impulso luminoso, che possono essere analizzate dai segnali elettrici prodotti, proporzionali all'energia della sorgente in modo da poter ricostruire il profilo temporale dell'impulso luminoso originale.

L'effetto fotoelettrico è quello su cui si basano i fotorivelatori, cioè la capacità di alcuni materiali di convertire fotoni in portatori carichi elettricamente.

Nei fotorivelatori c'è una finestra in ingresso per la rivelazione della radiazione luminosa dove un'area sensibile assorbe la luce incidente e la converte in segnale elettrico.

Ci sono molti fattori che entrano in gioco nel processo di fotorivelazione come le caratteristiche intrinseche del materiale utilizzato (spessore, coefficiente di assorbimento) oppure le riflessioni sulla superficie.

Il fotorivelatore in risposta produce un impulso di segnale, corrente o tensione, che tipicamente viene amplificato.

1.10 FOTODIODI

I fotodiodi sono dispositivi a stato solido a semiconduttore per la rivelazione di singoli fotoni, quindi necessitano di alta sensibilità perché in presenza di intensità molto basse.

Una nuova generazione di fotodiodi sensibili al singolo fotone vede come rappresentante lo SPAD (Single Photon Avalanche Diode), dispositivo realizzato mediante tecnologia planare e basato sull'uso di giunzioni p-n al silicio con alta concentrazione di drogante e in grado di fornire in uscita segnali di corrente direttamente amplificati come risposta ad un singolo fotoelettrone prodotto. Lo SPAD funge da contatori di fotoni.

Lo SPAD è costituito da un sensore monolitico e un particolare circuito, di quenching, che permette di operare in una condizione di on-off sul singolo fotoelettrone, il cosiddetto "Geiger Mode".

La giunzione p-n viene polarizzata inversamente con una tensione superiore a quella nominale di breakdown in modo che quando un fotone colpisce la giunzione si produce una coppia elettrone/lacuna che si separa e ogni portatore si dirige nella sua zona maggioritaria. Nella zona di carica spaziale il portatore viene accelerato da una elevata caduta di campo elettrico innescando una valanga di cariche in tempi rapidissimi che produce in uscita un segnale di corrente dell'ordine del mA.

Dopo un certo tempo il circuito di quenching riporta il fotodiodo nella condizione di poter rivelare un altro fotone.

E' possibile la realizzazione di array e matrici di SPAD come oggetto di studi e ricerche di molti 'enti', tra cui l'LNS(Laboratori Nazionali del Sud), STM, VUB(Vrije Universiteit Brussel) e UCLA(University of California LosAngeles).

[Angelo Campisi 2004/2005]

1.11 DLI (Deep Litography with Ions-Litografia profonda con Ioni)

Durante gli ultimi anni si sono sviluppate varie tecnologie che si occupano della realizzazione e produzione di componenti di alta qualità, come micro-opto elementi e strutture micro-meccaniche come array di microlenti, microprismi e supporti in fibra con importanti applicazioni nel trasferimento ottico di dati e nelle telecomunicazioni.

Tra queste tecnologie c'è la Litografia Profonda con Ioni (DLI) che consiste nell'uso di un fascio microscopico di ioni di energia compresa tra i 10 MeV e i 20 MeV, collimato con maschere circolari di diametro compreso tra i 20 μm e 1 mm, che colpisce una determinata regione di un campione di PMMA (PoliMetilMetacrilato) che sarà poi trattato chimicamente (incisione o dilatazione) in base alla funzione desiderata.

L'irradiazione di un campione di PMMA avrà come risultato un'importante riduzione dello spessore molecolare del materiale nella zona irradiata, saranno creati radicali liberi con conseguente cambiamento delle proprietà del materiale bombardato rispetto al resto del campione.

Tale zona presenterà una più alta solubilità rispetto al resto del materiale e può essere incisa selettivamente tramite l'utilizzo di un solvente speciale permettendo la fabbricazione di array bidimensionali, di microfori, microspecchi, microprismi, strutture di supporto meccanico, dispositivi di precisione per l'allineamento di parti ottiche e fibre, etc.

Oppure si può dilatare la zona irradiata utilizzando un vapore di monomeri organici; infatti la velocità di diffusione per le molecole più piccole sarà molto più alta che quella del materiale non irradiato.

Questo processo di diffusione causerà un'espansione considerevole del volume dell'area irradiata con il risultato di superfici semisferiche e permette la realizzazione di microlenti sferiche, cilindriche ad alta precisione, microcavità etc.

Se necessario, entrambi i processi possono essere applicati a regioni differenti dello stesso campione.

La dose di irradiazione assorbita dal settore del campione di PMMA è un parametro cruciale per gli effetti dell'incisione e della dilatazione, inoltre è importante controllare l'omogeneità, la dimensione, la forma e il contorno del fascio di ioni accelerato, in modo veloce ed efficiente.

A questo scopo si utilizzano dei collimatori, aperture raffreddate ad acqua, che fermano la parte esterna del fascio lasciandone passare solo la parte centrale, che verrà analizzata e misurata per poter posizionare e sagomare il fascio.

[Articoli]

1.12 DISPOSITIVI PER DIAGNOSTICA

Sono state sviluppate due tecniche per formare e modellare un microscopico fascio di ioni, entrambe basate sull'uso di scintillatori e particolarmente adatte per applicazioni in DLI ed utilizzate con successo.

La prima utilizza una placca scintillatrice chiamata SFOP (Scintillanting Fiber Optic Plate), fatta da un fascio di fibre ottiche di trebio e osservato da una camera CCD con lenti adatte, ogni fibra ha il diametro di 10 μm .

Si parte dalla collimazione per ottenere la grandezza del fascio desiderata e si passa al posizionamento della maschera su cui è posto il campione perpendicolarmente al fascio; quindi lo SFOP e la camera CCD sono installate di fronte al fascio. Quando il fascio colpisce lo SFOP, questo produce luce scintillante che è rivelata dalla camera e mostrata sullo schermo di un computer.

Questa tecnica è molto più sensibile di quanto ci si aspettasse e i risultati sono così promettenti che contribuirà a migliorare l'accuratezza e la qualità con cui i componenti micro-ottici possono essere fabbricati tramite DLI.

La seconda tecnica utilizza un sensore scintillatore basato su sensori di fascio chiamato SBBS (Scintillator Based Beam-Sensor) che consiste di un oscillatore (CsI(Ti)) di $1*1*0.2 \text{ mm}^3$ accoppiato ad un fotomoltiplicatore .

La caratteristica che a volte la rende una migliore soluzione rispetto alla prima tecnica è l'utilizzo di uno 'stopper' del fascio ad alta precisione.

Sono stati sviluppati diversi sensori di diagnostica dei fasci basati sugli oscillatori, la loro sensibilità ha permesso di ottenere migliori prestazioni rispetto ai dispositivi elettrici usuali in termini di robustezza, affidabilità, facilità nell'implementazione e soprattutto economicità.

[Alfio Pappalardo 2001/2002]

1.13 APPLICAZIONI FUTURE E IN CAMPO MEDICO

Sempre maggiore il numero di applicazioni che vedono l'utilizzo di fasci accelerati di ioni, sia per scopo terapeutico (TIB) che per scopi di carattere scientifico e tecnologico. I dispositivi di diagnostica di fasci, per lo studio di posizione e profilo trasverso, sono sempre più richiesti e con una sensibilità sempre maggiore in modo da garantire una affidabilità adeguata e non meno importante, una semplicità di utilizzo.

Possibili applicazioni vanno dalla creazione di componenti opto-elettronici tramite DLI allo studio di campioni, come rivelatori, da impiegare in molti settori scientifici: medico, elettronico, astronomico e così via.

Un'applicazione che in modo particolare si vuole elogiare in questa tesi è la ricerca nel campo medico che negli ultimi anni ha riscontrato un particolare successo.

E' il caso del Centro di AdroTerapia e Applicazioni Nucleari Avanzate (CATANA) che è l'unico centro di adroterapia in Italia e in collaborazione con l'Istituto di Oftalmologia, l'Istituto di Radiologia e il Dipartimento di Fisica di Catania presso i LNS-INFN, si occupa di terapie di tumori oculari, tra cui i più comuni sono i melanomi alla coroide e all'iride, tramite l'irradiazione con protoni e riscontrando un successo sempre crescente soprattutto per quei tumori di minore entità.

La terapia è eseguita presso i LNS e stabilisce l'uso del fascio di protoni per le caratteristiche fisiche proprie dei protoni.

Quando i protoni si fermano ad una profondità prestabilita, ottenuta tramite variazione dell'entità del fascio, una quantità elevata di energia viene depositata nel corpo; è molto importante avere una conoscenza delle caratteristiche del tumore, come la sua estensione all'interno dell'occhio. Infatti il tumore è circoscritto tramite l'applicazione di 4 piccoli anelli sulla parte posteriore dell'occhio in fase di operazione, in modo da poter ben vedere il bordo con tecniche radiografiche.

Per ogni paziente si sceglie una profondità di penetrazione del fascio e un'estensione omogenea in modo individuale. Il paziente viene fatto accomodare su una sedia motorizzata e controllata da computer, quindi la testa e il collo sono immobilizzati. Poi, tramite immagini radiografiche, è possibile determinare la forma del tumore all'interno dell'occhio e la posizione ottimale per passare all'irradiazione dell'occhio facendo in modo che essa non colpisca in modo pesante i tessuti sani circostanti.

La terapia di protoni è la più precisa forma di trattamento avanzato con radiazioni perché minimizza i danni ai tessuti circostanti e sani.

Le cellule tumorali colpite dal fascio sono rese sterili in modo da non potersi riprodurre; ovviamente il risultato varia da soggetto a soggetto e anche in base alla grandezza della lesione ma gli sviluppi sono molto promettenti.

[G. A. Pablo Cirrone, V. Mongelli, 2003]

CAPITOLO 2 - LabVIEW

Il LabView è l'ambiente di sviluppo scelto per creare uno strumento software adeguato all'esigenza di movimentare un sistema rototraslatorio con una precisione micrometrica; la scelta è ricaduta su questo linguaggio sia per la curiosità di lavorare con uno strumento prevalentemente grafico e molto potente dal punto di vista della programmazione, infatti è possibile implementare una grande varietà di funzioni da utilizzare in applicazioni di ogni genere sia per l'esistenza di un altro programma [Pedro Vynck, 2004] utilizzato per altri scopi e con simili funzionalità che è stato molto utile in un primo momento per capire come affrontare il problema della movimentazione.

In questo capitolo si esporranno le caratteristiche generali del LabVIEW e i principali oggetti utilizzati nella creazione del programma.

2.1 GENERALITA' SUL LabVIEW

Il LabView (LV) è un sistema di sviluppo grafico dove il flusso di dati determina l'esecuzione del programma, in contrasto con quei programmi basati sui linguaggi di testo in cui le istruzioni ne determinano l'esecuzione.

E' un sistema di sviluppo disponibile per diverse piattaforme (MS Windows, Linux, Macintosh, Solaris). Con il LV si può facilmente costruire un'interfaccia utente utilizzando un set di strumenti e oggetti, si possono creare sistemi per l'acquisizione di dati e analisi di misure, strumenti di controllo, applicazioni di nuova generazione ed eseguibili stand-alone e librerie condivise, come le DLL; tutto ciò perché il LV è un vero compilatore a 32 bit .

Inoltre fornisce strumenti di debug molto potenti che aiutano nella risoluzione dei problemi del codice.

Il LV permette di collegare le proprie applicazioni al Web utilizzando il LabView Web Server e sw standard come reti TCP/IP e ActiveX.

LabVIEW può essere utilizzato per comunicare con un hardware come quello per l'acquisizione dei dati e la visione, con i dispositivi di controllo del movimento e con strumenti GPIB, PXI, VXI, RS-232 e RS-485.

Un programma in LabVIEW, o VI (Virtual Instrument), si sviluppa lavorando su due finestre: il *"Front Panel"* e il *"Block Diagram"* (in fig 2.4), ciascuno dei quali fornito di una propria paletta degli strumenti (*Tools Palette* in fig 2.1). Inoltre la *"Controls palette"* e la *"Functions palette"* forniscono gli oggetti necessari che possiamo selezionare e posizionare nel Front Panel e nel Block Diagram nel modo più opportuno (fig 2.2 e 2.3).

Fig 2.1



Fig 2.2

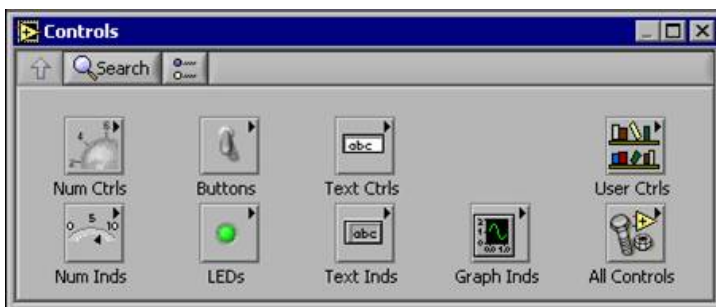


Fig 2.3

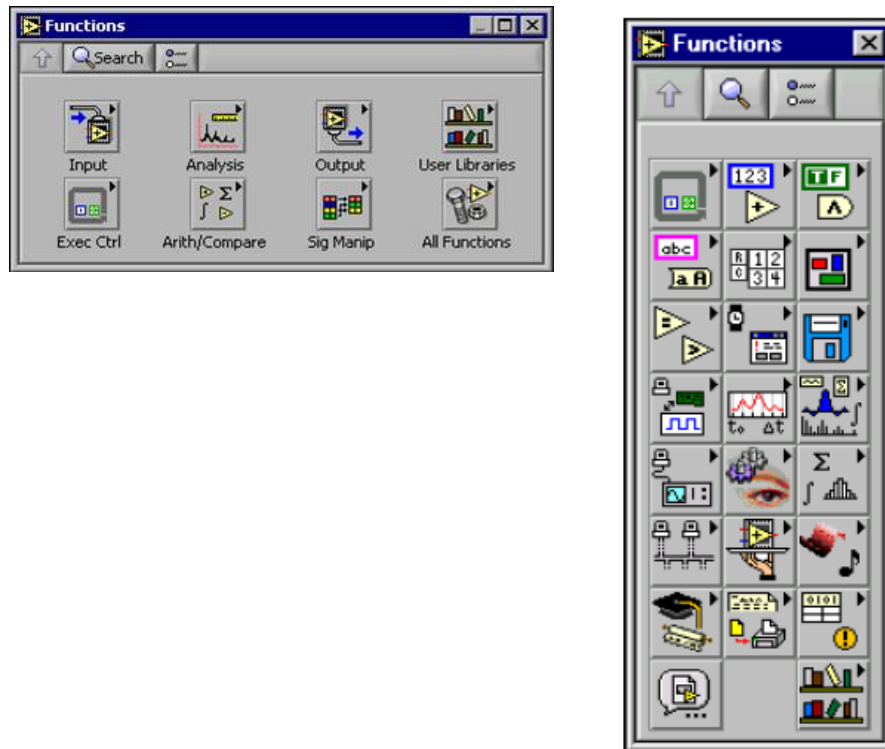
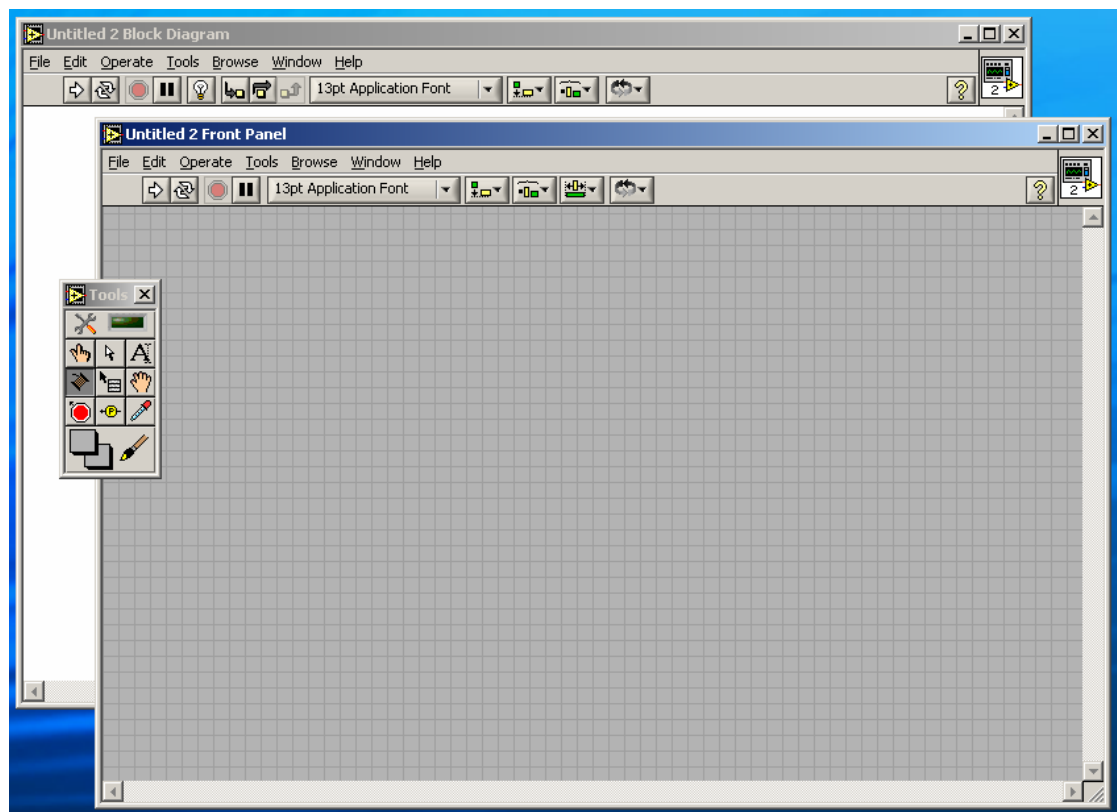


Fig 2.4



Per ogni oggetto inserito nel Front Panel, che è l'interfaccia utente, (indicatori numerici, pulsanti, grafici e controlli di ogni genere) viene automaticamente creato un terminale corrispondente nel Block Diagram, che può essere considerato come uno schema a blocchi nel quale effettivamente è contenuto il codice creato graficamente. I terminali hanno punti di input e output che opportunamente collegati rappresentano in forma grafica il flusso di dati tra un blocco struttura ed un altro.

I controlli sono le manopole (*knob*), i pulsanti da premere (*push button*), i selettori (*dial*) e altri dispositivi di input; i controlli simulano i dispositivi di input di uno strumento fisico e forniscono i dati allo schema a blocchi del VI.

Gli indicatori sono i grafici, i LED e altri display, essi simulano i dispositivi di output di uno strumento fisico e visualizzano i dati acquisiti o generati dallo schema a blocchi.

Una volta creata l'interfaccia utente, è possibile aggiungere il codice utilizzando VI e strutture che controllano gli oggetti del pannello frontale.

Lo schema a blocchi contiene il codice sorgente grafico grazie al quale il VI può essere eseguito. Gli oggetti del pannello frontale appaiono come terminali sotto forma di icona nello schema a blocchi. I fili connettono i terminali dei controlli e degli indicatori ai VI. I dati fluiscono dai controlli ai VI, da VI a VI e dai VI agli indicatori attraverso i fili.

Dalla combinazione di questi elementi è possibile sviluppare programmi di notevole complessità anche solo scrivendo poche righe di codice, bastano pochi click di mouse per avere sotto controllo tutto il programma.

Uno strumento di debug che si è rivelato particolarmente utile per la progettazione è l'esecuzione in modalità *highlight* che opportunamente combinata con punti di controllo (*breakpoint*) e indicatori (*probe*) permette di seguire l'esecuzione del codice e scovare gli errori. Infatti durante lo sviluppo del programma questo strumento ha permesso di individuare in breve tempo i problemi di progettazione che altrimenti sarebbero stati più difficili da trovare.

Una caratteristica importante è che è possibile apportare delle modifiche al volo ai programmi e mandarli in esecuzione senza averli prima salvati; in LabVIEW non esiste la compilazione tipica della maggior parte degli strumenti di sviluppo!

I VI possono contenere al loro interno dei sub-VI, dei piccoli programmi a se stanti, progettati anche dall'utente stesso, che permettono una maggiore astrazione nella programmazione.

Nel presente progetto si è utilizzato un set di sub-VI fornito a corredo dell'ESP6000 che contengono applicazioni costruite sulle funzioni scritte in codice C proprie del controllore stesso.

[National Instrument, 2000] [National Instrument, 2003]

2.2 ELEMENTI PRINCIPALI

La versione sw utilizzata è LabVIEW 7.1.

Nel programma sviluppato si utilizzano varie funzioni, costrutti e variabili che verranno spiegati in breve a seguito.

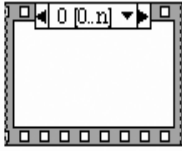
In questa tesi si assumerà per convenzione un colore appropriato per ciascun elemento del codice LabVIEW:

- Verde, per gli elementi booleani.
- Arancione, per gli elementi double.
- Blu, per gli elementi interi.
- Fucsia, per gli elementi composti come i cluster.

Strutture come array assumeranno come colore quello del tipo dei loro elementi.

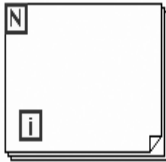
2.2.1 Strutture

Stacked Sequence



E' costituita da subdiagrams, chiamati frame, che eseguono sequenzialmente. E' possibile aggiungere o eliminare frame o creare sequenze locali per passare dati tra i frame facendo right-click sul bordo della struttura. Usando la stack sequence si decide di far eseguire i frame secondo un ordine di preferenza seguendo l'ordine numerico.

For Loop



Esegue i diagrammi che contiene n volte, dove n è il valore collegato al terminale contatore (N). Il terminale iterazione (i) fornisce il numero del corrente ciclo di iterazione, un numero che va da 0 a n-1.

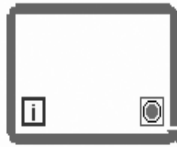
Case Structure



Contiene uno o due sottodiagrammi, o casi e solo uno di essi esegue quando la struttura è in esecuzione. Il valore collegato al terminale di

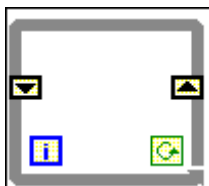
selezione (?) determina quale caso eseguire e può essere booleano, stringa, intero o di tipo enumerato.

While Loop



Ripete il sottodiagramma che contiene fino a quando il terminale della condizione, un terminale di input, riceve un particolare valore booleano. Dal valore booleano dipende il comportamento del ciclo while. Facendo right-click sul terminale di condizione e selezionando 'Stop if True' o 'Continue if True' dal menu si può scegliere la condizione da adottare. Il ciclo while esegue fino alla fine e il terminale di iterazione (i) fornisce il numero del ciclo di iterazione corrente, che è zero per la prima iterazione.

Shift Register



E' usato con i cicli while e for per trasferire dati da un'iterazione all'altra, si crea facendo right-click e selezionando Add Shift Register dal menu. Lo shift register possiede due terminali, le freccette ai due lati, di cui quella a destra immagazzina i dati fino alla fine dell'iterazione, poi li manda al terminale di sinistra all'inizio dell'iterazione successiva.

2.2.2 Variabili locali

Il meccanismo di comunicazione fra i diversi blocchi è basato sull'uso di variabili che possono essere utilizzate con un meccanismo di "passaggio per valore" che permette di creare nuove istanze, o copie, degli oggetti.

Le variabili locali e le loro istanze permettono di aggiornare o leggere un oggetto del front panel da diverse locazioni del block diagram e passare dati tra strutture che non possono essere collegate tra loro.

Fig 2.5

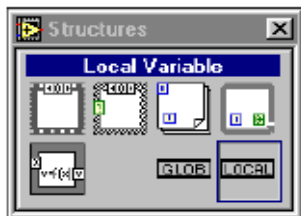
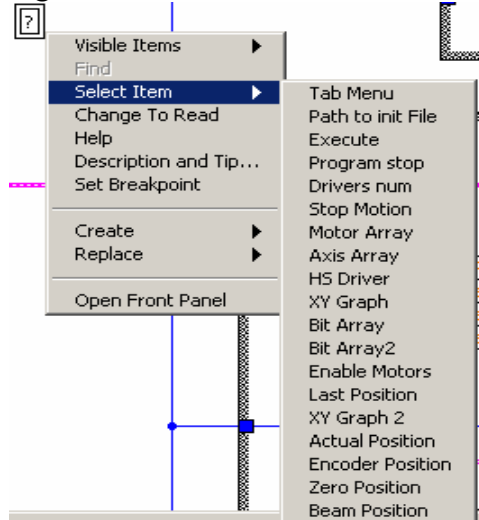


Fig 2.6



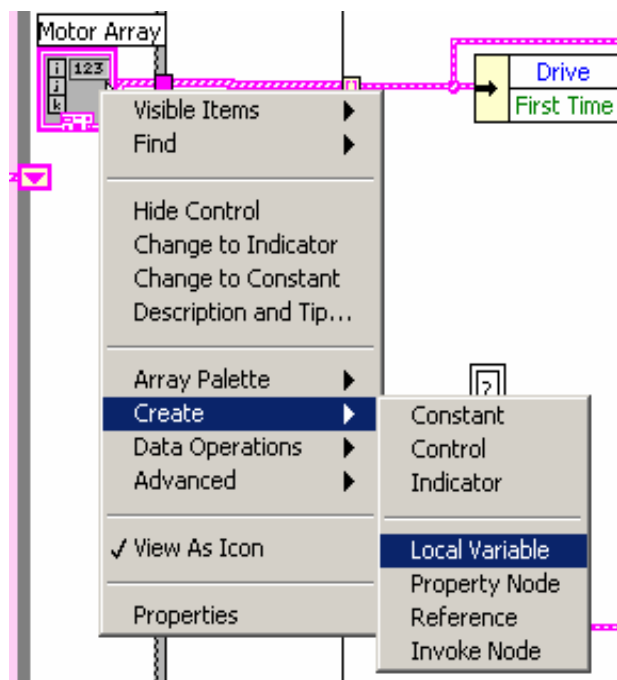
In figura 2.5 vediamo che si può creare una variabile locale selezionando LOCAL in modo da ottenere il nodo (?) in figura 2.6, a cui si associa l'elemento di cui si vuole creare una copia scegliendolo da Select Item (right click sul nodo figura 2.7).

Fig 2.7



Oppure la variabile si può creare direttamente dall'elemento specifico, facendo right-click e selezionando create-> local variable (figura 2.8).

Fig 2.8



Con una variabile locale si può sia leggere che scrivere dati, basta specificare come usarla; per default essa si trova allo stato di scrittura, visibile dal fatto che il bordo esterno è sottile, ma facendo right-click sulla variabile e selezionando 'Change To Read' essa diventa come una fonte di dati da cui è possibile leggere e il bordo esterno diventa spesso.

Si può ritornare allo stato di scrittura facendo right-click sulla variabile e selezionando 'Change To Write', questo le permette di cambiare configurazione e quindi poter solo ricevere dati.

Quando scrivi su una variabile locale, si aggiorna il suo corrispondente controllo o indicatore sul front panel.

Quando invece si legge da una variabile locale, si legge il valore corrente del suo corrispondente controllo o indicatore sul front panel.

Le variabili locali aumentano il numero di chiamate in memoria e rallentano la velocità di esecuzione, specialmente quando vi si accede nei cicli; per questo è consigliabile non farne un uso improprio. Un modo

semplice di ovviare a questo inconveniente è usare gli shift register , che permettono di fare uso di un minor numero di variabili locali.

Infatti nel corso della programmazione si sono riscontrati dei problemi negli aggiornamenti delle strutture di cui si sono create molte variabili locali, perchè le variabili sono entrate in conflitto in quanto utilizzate contemporaneamente da più cicli. Con l'utilizzo degli shift register si è superato apprezzabilmente il problema.

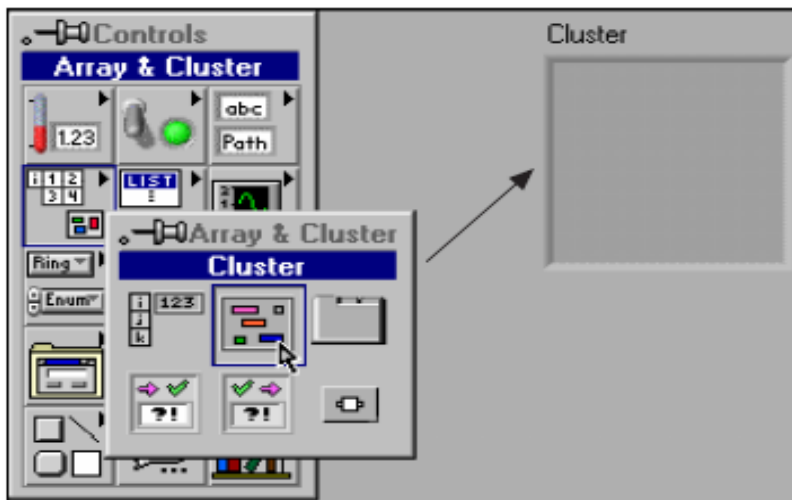
Tra le variabili locali utilizzate nel corso del programma:

- *Execute*, la variabile booleana istanza del pulsante *Execute* che svolge una funzione molto importante perché il suo valore determina il funzionamento del programma similamente ad un interruttore, se il suo valore è false non funziona niente, come se tutto fosse spento.
- *Stop Motion*, istanza dell'omonimo pulsante che permette di arrestare il movimento dei motorini in ogni istante.
- *Enable Motors*, istanza del pulsante *Enable Motors* che ha la funzione di abilitare e disabilitare i motorini.
- *Bit Array* e *Bit Array2* sono le variabili locali associate agli arrays dei bit di controllo.
- *Drivers Num*, rappresenta il numero dei drivers ed è utilizzato principalmente come il terminale (N) nei cicli for.
- *Motor Array* e *Axis Array* sono create per permettere di accedere agli elementi delle omonime strutture.
- *Encoder Position*, *Actual Position*, *Last Position* e *Zero Position* rappresentano le variabili locali associate agli arrays corrispondenti e sono utilizzate quando si vuole aggiornare o leggere un valore.

2.2.3 Cluster

Il cluster è una struttura dati che combina uno o più componenti in un nuovo tipo di dato (fig 2.8). I componenti che formano un cluster possono contenere diversi tipi di dato come booleano, stringa, numerico e può essere considerato come una *struct* del C o un *record* del Pascal.

Fig 2.8



Nel block diagram il filo che porta i dati immagazzinati nel cluster può essere considerato come un fascio di fili più piccoli, come un cavo telefonico, in modo da diminuire il numero di fili e di terminali connettori di cui si ha bisogno. Ogni filo nel cavo rappresenta uno dei componenti del cluster, ciò vuol dire che bisogna fare un'opera di 'unbundle' (fig 2.10) , letteralmente 'sfasciare', del cluster ogni volta che si vuole accedere ad uno dei suoi elementi e poi l'operazione inversa di 'bundle' (fig 2.9).

Fig 2.9

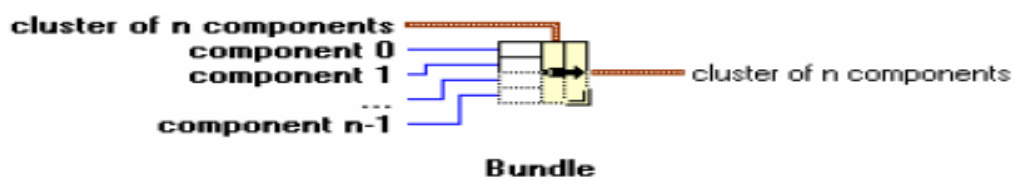
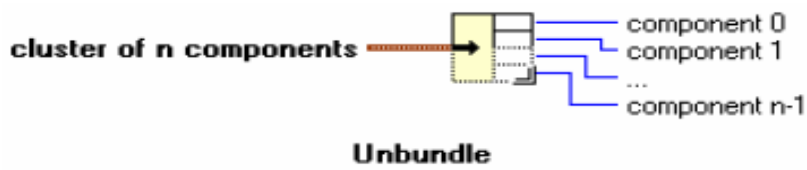


Fig 2.10



Bundle e unbundle possono essere dimensionati in base al numero (n) dei componenti del cluster, da un minimo di uno ad un max di n.

Bundle by name assegna un valore al parametro d'ingresso che si può selezionare facendo click destro -> Select Item (figura 2.11)e scegliendo tra gli elementi del cluster; ritorna in uscita il cluster con il valore assegnato.

Unbundle by name invece fornisce in uscita il valore di uno o più elementi, scelti sempre facendo click destro e selezionando tra tutti gli elementi del cluster (figura 2.12).

Fig 2.11

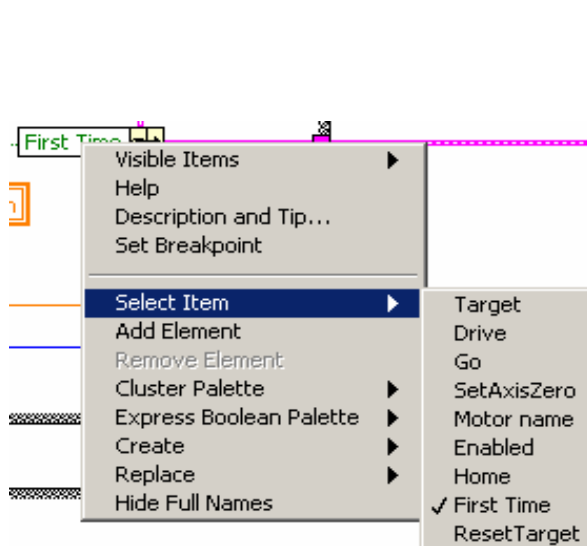
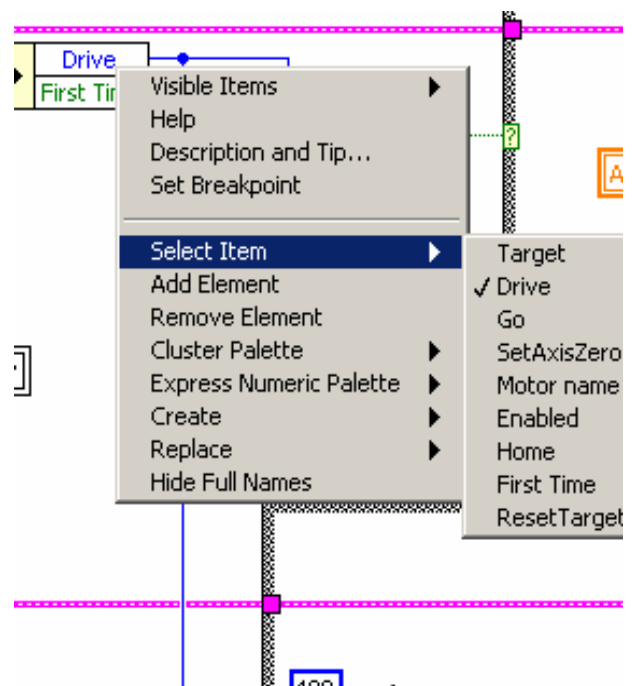


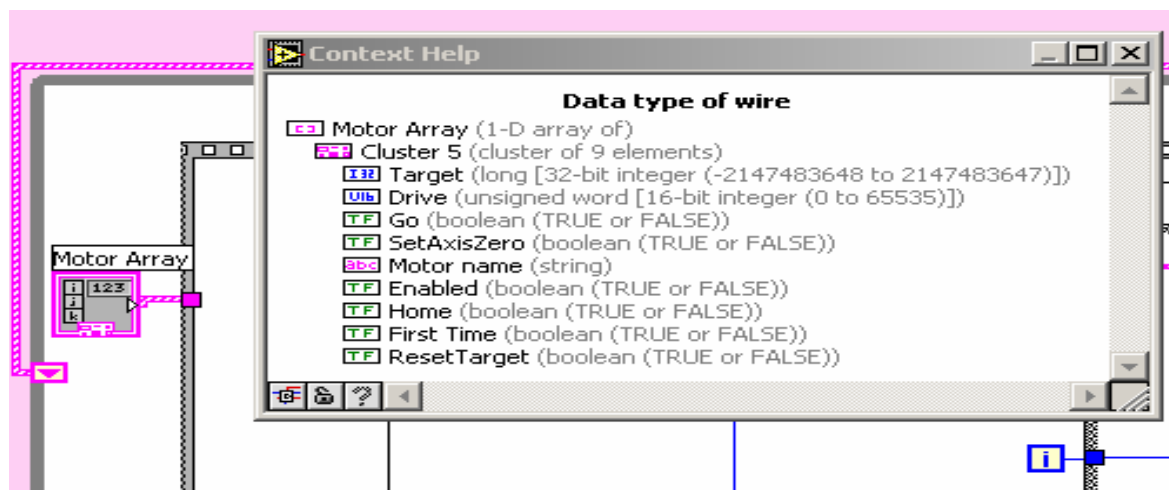
Fig 2.12



Axis Array e **Motor Array** sono arrays di cluster, ciò comporta che tutte le operazioni di lettura o aggiornamento dei loro elementi sono state eseguite tramite unbundle/bundle.

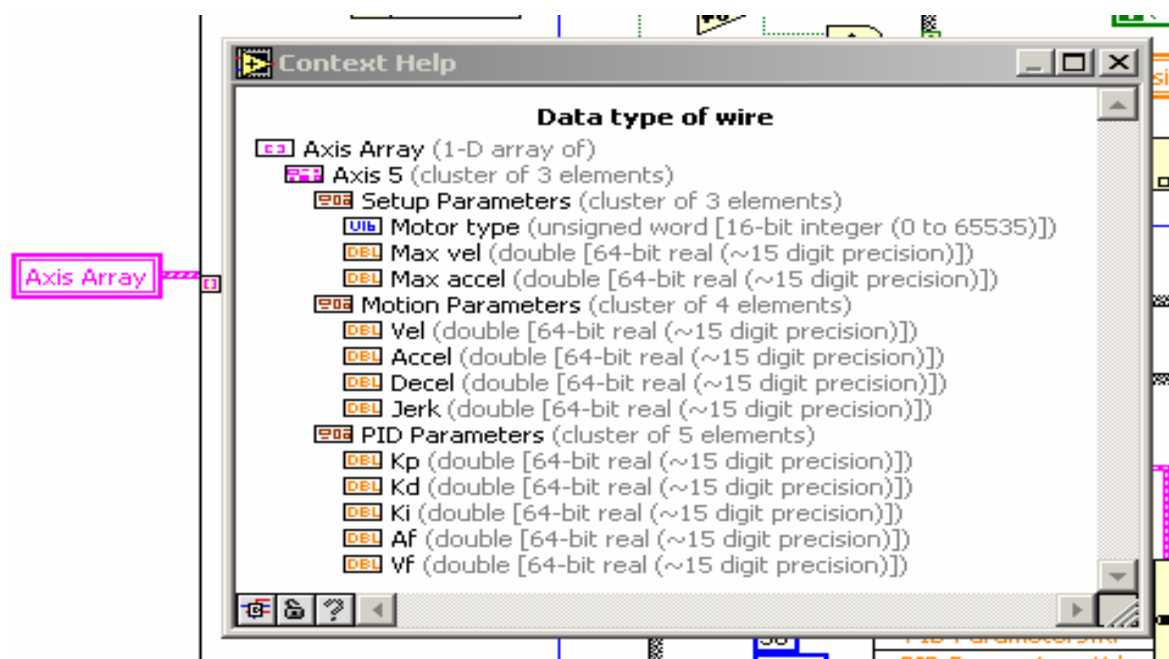
Motor Array è costituito dagli elementi in figura 2.13

Fig 2.13



Axis Array è costituito dagli elementi in figura 2.14

Fig 2.14



CAPITOLO 3 – Applicazione SW

Nel presente capitolo sarà spiegata ogni parte dell'interfaccia utente, come essa permette di dialogare con le componenti hardware e le funzionalità di ciascun elemento; sarà analizzato il codice grafico e la sua strutturazione in modo da rendere più semplice la comprensione dell'intera applicazione.

Nella realizzazione dell'interfaccia si è cercato di tenere conto della esigenza di semplicità e compattezza in modo da fornire all'utente uno strumento valido e di facile utilizzo per la movimentazione del sistema.

Il front panel rappresenta l'interfaccia utente ed è costituito da varie strutture che corrispondono a comandi, controlli, indicatori, grafici per la visualizzazione virtuale.

La scelta delle strutture, delle loro funzioni e della loro posizione, è stata fatta in modo da essere funzionale dal punto di vista operativo e permettere la concentrazione dei comandi principali in un'unica schermata; così, ad esempio, mentre un motorino si sta muovendo, si può vedere dal grafico il movimento del cursore e allo stesso tempo il cambiamento di posizione.

Le principali strutture che compongono l'interfaccia utente sono:

- I pulsanti esterni;
- Il 'tab' che rappresenta una tabella con due fogli, uno per l'array dei motorini, *Motor Array*, che comprende comandi e controlli per la movimentazione degli stessi e un array degli assi, *Axis Array*, che visualizza i parametri di motion e PID dei motorini;
- Una matrice di bit di controllo a fianco del tab;
- Un grafico principale che occupa buona parte della schermata, lo chiameremo "grafico del campione" perché rappresenta la posizione del campione rispetto al sistema di riferimento;
- Un grafico secondario, il "grafico del collimatore", che indica la posizione del collimatore finale rispetto all'asse del fascio;

- Tre arrays che indicano le posizioni dei motorini rispetto diversi punti di riferimento;

Fig 3.1

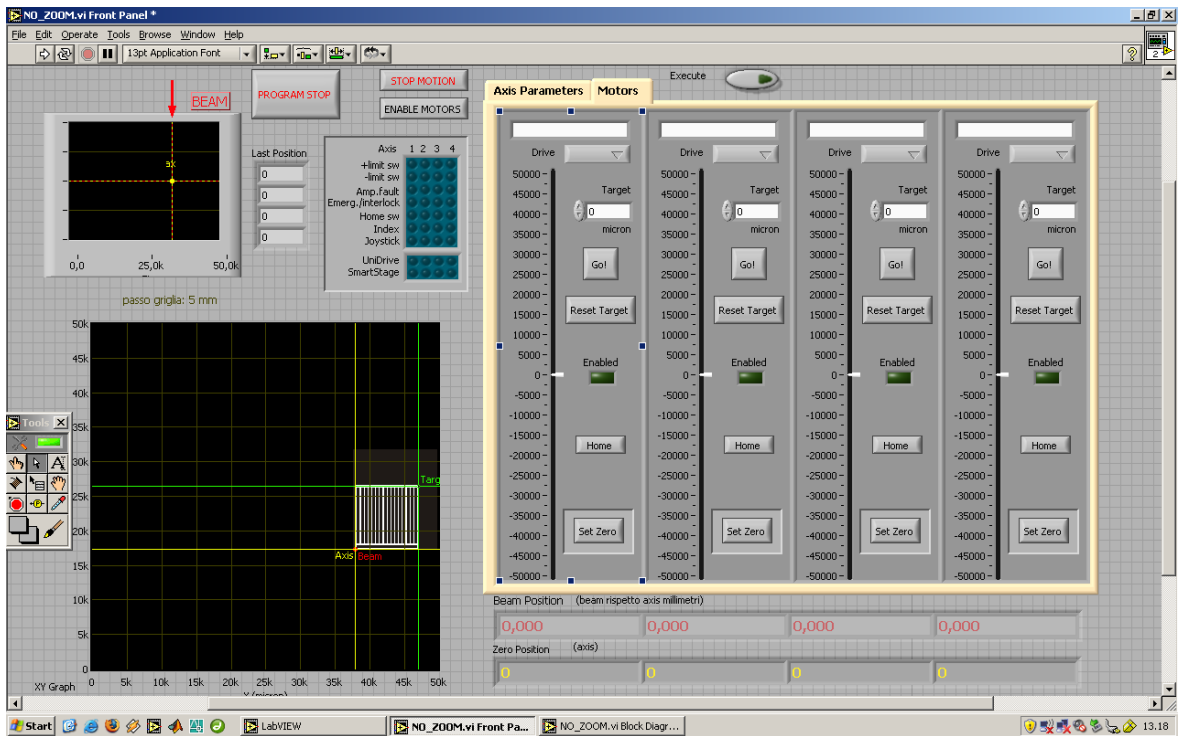
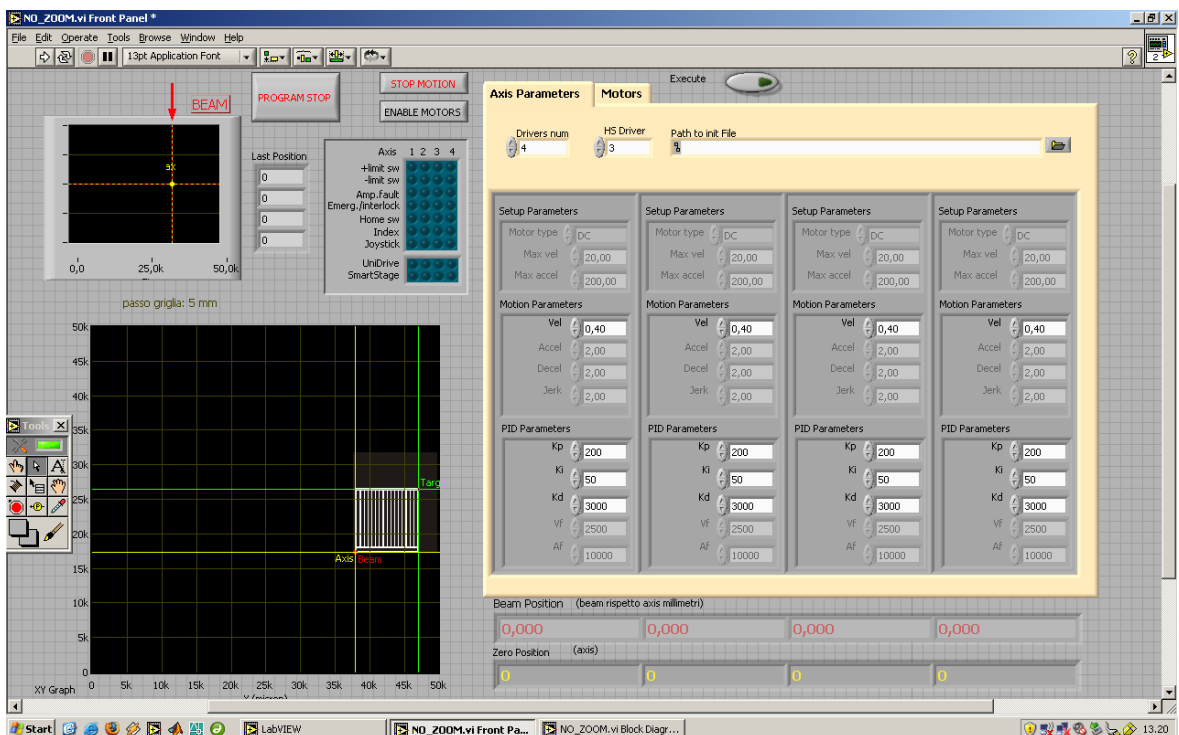


Fig 3.2



In figura 3.1 un'immagine del front panel con la tabella dei motorini, in figura 3.2 la stessa immagine con la tabella dei parametri associati ai motorini corrispondenti.

3.1 PULSANTI ESTERNI

Fig 3.3



In figura 3.3 i pulsanti esterni dell'interfaccia.

Il pulsante *Execute* funge da interruttore, deve essere acceso per abilitare il programma.

Program Stop è il comando che permette di uscire dal programma, disabilitare i motorini e salvare la configurazione corrente.

Stop Motion ha la funzione di bloccare istantaneamente il movimento dei motorini, da usare in caso di emergenza.

Il pulsante *Enable Motors* abilita i motorini e quando si accede al programma esso è allo stato false (Disable) perché i motorini vengono accesi con l'inizializzazione del controllore.

Sulla destra dell'interfaccia utente si presenta il tab con le due pagine introdotte prima.

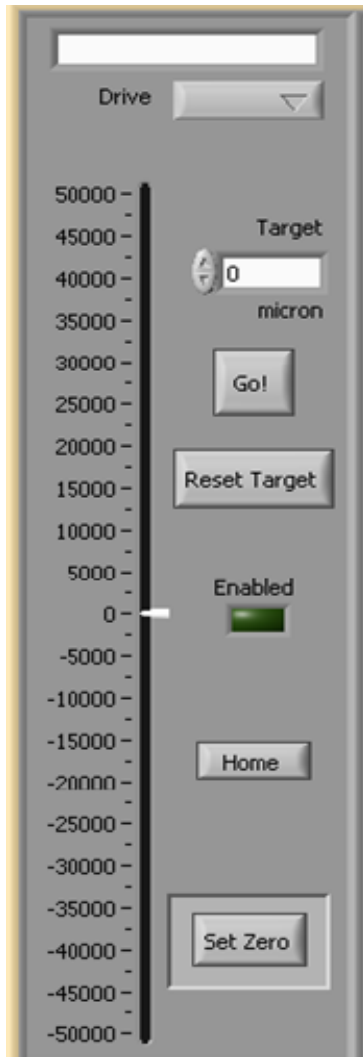
3.2 MOTOR ARRAY

In figura 3.4 uno dei cluster del *Motor Array* che rappresenta la parte principale dell'interfaccia utente perché da qui si può movimentare il sistema.

La targhetta in alto è quella in cui viene scritto il nome del motorino una volta letto il file di inizio.

Il selezionatore *Drive* permette di scegliere il motorino da assegnare al drive specifico, anche a questo viene dato per default il valore letto dal file di inizio.

Fig 3.4



Lo scroll a sinistra che si estende in verticale rappresenta il range relativo di valori che il target può assumere; da tenere in considerazione che se non si preme il tasto *Go* non avviene nessuna movimentazione.

Il valore del target può essere anche immesso direttamente digitandolo sulla targhetta, ciò permette anche di effettuare spostamenti 'fini' dell'ordine del singolo micron.

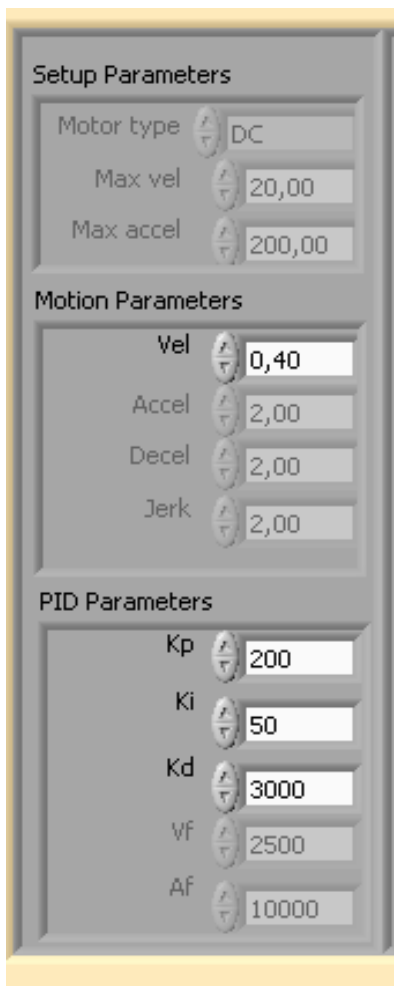
Il pulsante *Reset Target* ha la funzione di settare il target alla posizione corrente del fascio.

Il led *Enabled* indica se il motorino associato al driver è abilitato oppure no. Quando è abilitato il led si accende e assume un colore verde brillante, non a caso, per essere notato e fare in modo che in caso di spegnimento indesiderato e inaspettato del motorino, l'utente se ne accorga subito e provveda alla riabilitazione (spegnendo e accendendo tutti i motorini) oppure, in caso di raggiungimento di uno dei limiti hardware, riporti il sistema in una situazione di operatività.

Il pulsante *Home* ha il compito di riportare il target e il fascio alla posizione 'zero relativo' ed è molto utile anche perché in caso di errori nella manipolazione del target, permette di tornare ad una situazione di partenza ottimale senza compromettere l'integrità e la stabilità del sistema. Infine il pulsante *Set Zero* permette di settare la posizione di quello che sarà considerato come lo zero del sistema di riferimento e dal quale si prenderanno le posizioni relative di target e beam.

3.3 AXIS ARRAY

Fig 3.5



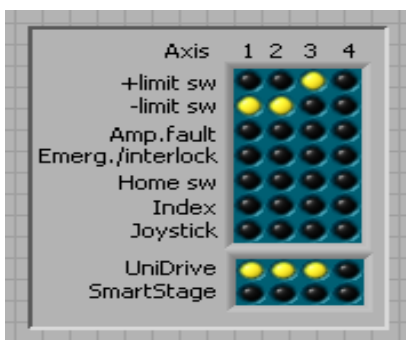
In figura 3.5 un cluster dell' **Axis Array** che contiene i parametri relativi ai motorini. I parametri di maggiore interesse sono la velocità, che nel caso dei motorini serie 850-G è posta a 0,4 $\mu\text{m/s}$ e i parametri PID Kp, Ki, Kd che, sempre nel caso del motorino serie 850-G, sono posti rispettivamente ai valori 200, 50, 3000 (per i parametri PID fare riferimento al paragrafo 1.5 del capitolo 1).

Nel caso del motorino serie 850-HS la velocità è posta a 3 $\mu\text{m/s}$, i parametri Kp, Ki, Kd invece sono posti rispettivamente a 500, 300, 1000.

Quando cambiamo il numero di motorino associato ad un driver automaticamente anche i parametri si aggiornano a quelli del nuovo motorino in modo che tutto funzioni correttamente con i propri valori ottimali.

3.4 BIT DI CONTROLLO

Fig 3.6



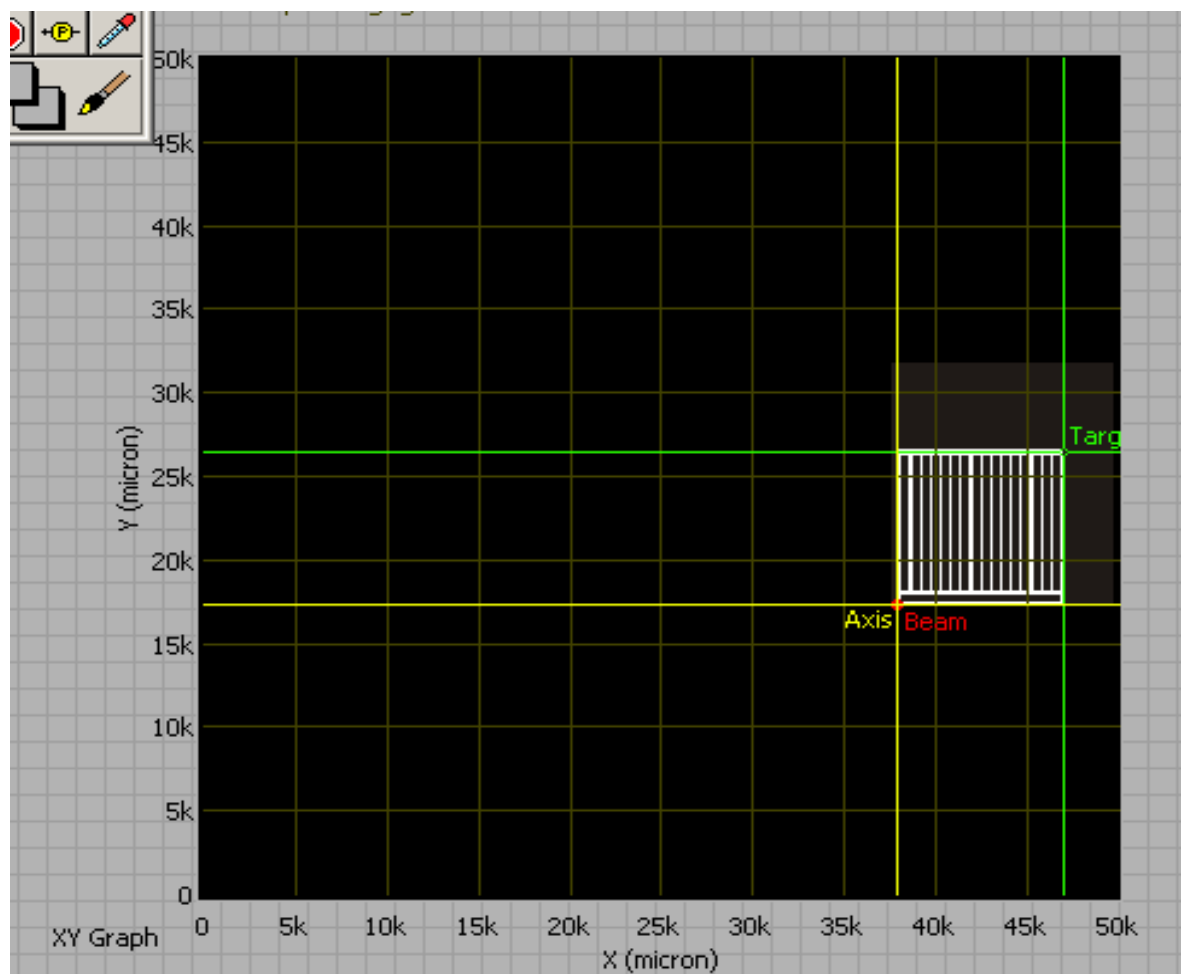
In figura 3.6 una matrice di bit che costituisce un importante indicatore dello stato dei motorini; le colonne rappresentano i motorini e le righe rappresentano i singoli controlli sui motorini.

I parametri che sono maggiormente utili e che danno un importante contributo ai fini del controllo, sono i primi due perché indicano quando il motorino corrispondente si trova al limite positivo (+limit sw) o al limite negativo (-limit sw).

Nel corso dei test per provare la validità del programma è stato di grande aiuto il poter tenere sotto controllo lo stato dei motorini direttamente dalla visualizzazione dei bit, che nel caso di allarme, assumono colore giallo.

3.5 GRAFICO CAMPIONE

Fig 3.7

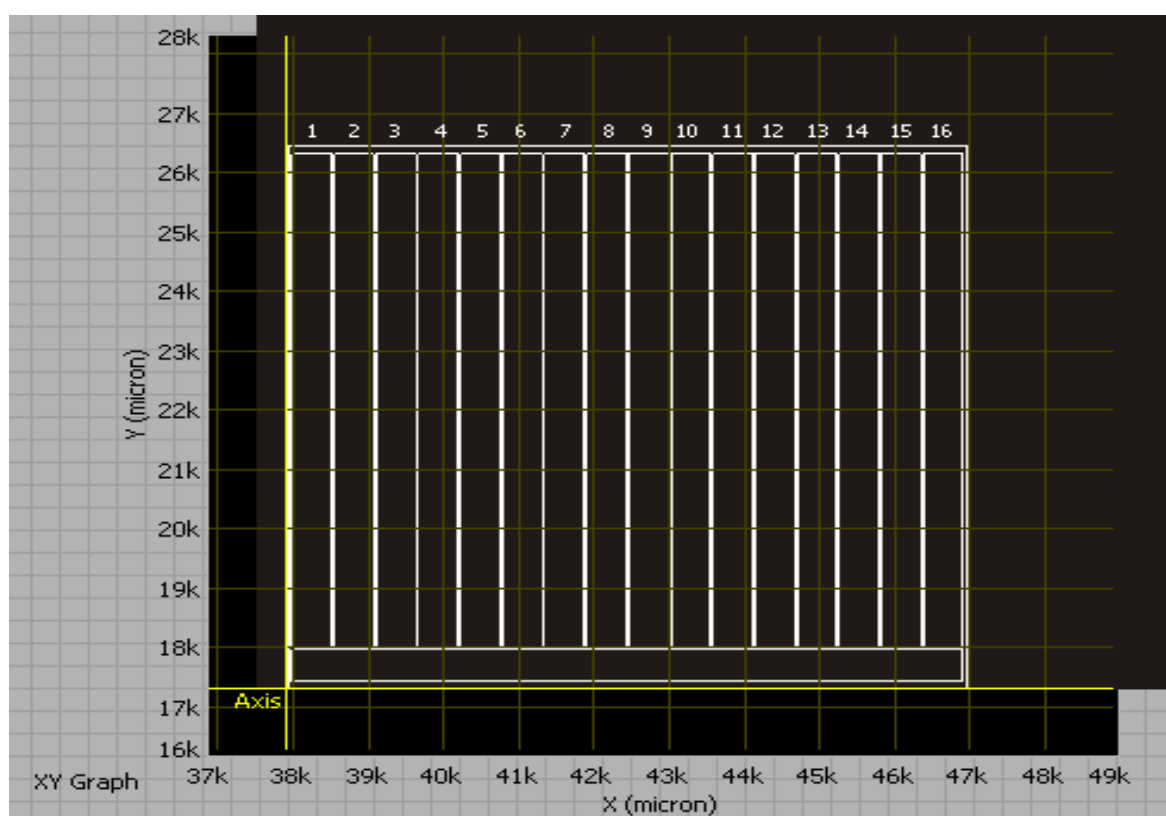


(passo di 5 mm)

In figura 3.7 una versione del grafico senza zoom che permette di fare un posizionamento preliminare dello zero relativo del sistema in modo da poter determinare dove inserire in sottofondo un'immagine in scala del campione.

Una volta determinata la posizione del campione in modo approssimativo si può chiudere il programma e salvare la configurazione relativa dei valori degli assi nel file di uscita che in seguito verrà aperto con la versione del programma con lo zoom.

Fig 3.8



(passo di 1 mm)

In figura 3.8 la versione del grafico con lo zoom che prende come coordinate degli assi i valori di X e Y che partono da 1mm in meno rispetto allo zero relativo che si è settato in precedenza.

Si dispone di una porzione ingrandita del grafico di partenza che permette di poter focalizzare l'attenzione sulla parte di interesse, in questo

caso la zona in cui si trova il campione perché è quella su cui si opererà durante il turno di misura.

Il posizionamento definitivo dell'immagine in sottofondo del campione verrà eseguita in modo preciso con l'ausilio di un laser o del fascio stesso nelle modalità spiegate più in dettaglio nel capitolo 4.

Il funzionamento del grafico è molto semplice, una volta capito il meccanismo che implica l'uso dei cursori.

Ci sono tre cursori di tre colori diversi e ognuno con il proprio nome e le proprie coordinate:

- Il cursore rosso è il beam, cioè rappresenta la posizione del fascio all'interno del sistema e le sue coordinate sono date dal valore fornito dall'array *Actual Position*.
- il cursore verde è il target, esso rappresenta lo spostamento relativo a partire dal beam e quindi le sue coordinate sono date dalla somma dei valori forniti dagli arrays *Zero Position* e *Last Position* e il valore inserito nella targhetta specifica del target.
- Il cursore giallo è quello che indica la posizione dello zero relativo ed è chiamato axis, le sue coordinate sono date dall'array *Zero Position* e sono settate quando si preme il pulsante *Set Zero*; una volta settato lo zero e sistemata la posizione definitiva del campione, queste coordinate rimarranno sempre le stesse per tutta la durata dell'applicazione.

Quando il programma viene aperto per la prima volta con il file di inizio "INIT X-Y.txt", con i motorini posti ai loro limiti negativi, tutti e tre i cursori si trovano alla posizione in alto a destra del sistema di riferimento.

Una considerazione molto importante per comprendere più a fondo il sistema è il fatto che il grafico rappresenta il campione in una posizione fissa rispetto ad un riferimento e il fascio, il cursore rosso, invece si muove posizionandosi sul punto da noi scelto. Nella realtà accade la cosa inversa, cioè il fascio è fermo ed è il campione che si muove in modo da trovarsi rispetto al fascio nella posizione corrispondente al grafico.

3.6 POSSIBILI APPLICAZIONI

A seguito alcune operazioni che convenzionalmente verranno eseguite durante l'utilizzo del programma.

3.6.1 Settare la posizione zero

Un primo passo è il posizionamento del target nel punto in basso a sinistra del campione, quindi si preme il pulsante **Go** e il beam si posiziona su target, infine si preme il pulsante **Set Zero** e lo zero relativo del nuovo sistema di riferimento sarà dato dalla posizione del beam e del target; graficamente ciò che si vede è che il cursore giallo si sposta sul punto in cui si trovano target e beam.

Da questo momento in poi si può lavorare solo con target e beam.

Un accorgimento per facilitare il lavoro è quello di salvare in uscita un file "INIT Y-X neg limit.txt" dove target e beam sono sul punto iniziale in alto a destra, ciò corrisponde al fatto che i motorini sono ai loro limiti negativi e axis settato alla posizione zero relativo.

3.6.2 Settare una nuova posizione

Muovendo il target tramite lo scrollino o inserendo un valore nell'apposita targhetta, si vede nel grafico lo spostamento del cursore verde; solo premendo **Go** il cursore rosso si sposterà sulla nuova posizione indicata dal target. Fatto questo è importante premere il pulsante **Reset Target** che resetta il valore dello spostamento relativo in modo che il beam possa posizionarsi nel modo corretto.

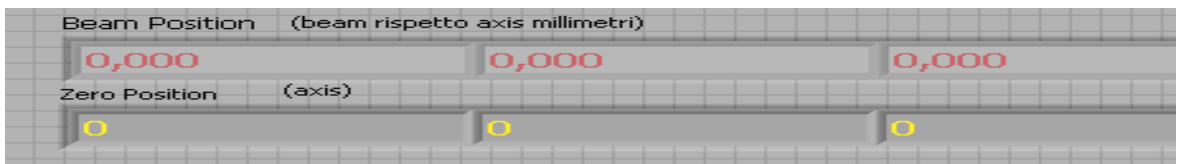
3.7 ARRAYS DELLE POSIZIONI

Fig 3.9



In figura 3.9 *Last Position*, è un array statico che conserva i valori di posizione del target fin quando non è premuto il pulsante *Go*, quando questo avviene i valori sono aggiornati alle coordinate correnti di target, che nel frattempo è cambiato.

Fig 3.10



In figura 3.10 *Zero Position*, è anch'esso un array statico e una volta settato, i valori rimangono gli stessi fino ad un nuovo comando "*Set Zero*".

Beam Position è l'array che indica la posizione del beam rispetto allo zero relativo.

3.8 PROBLEMATICA DEL TARGET

Il target, come detto in precedenza, è dato dalle coordinate fornite dagli arrays *Last Position* e *Zero Position*.

Se non si preme *Reset Target* il beam si sposterà di una posizione pari al valore del target, che non è stato posto a zero e si trova al valore precedentemente inserito, causando uno sfasamento del cursore verde rispetto a quello rosso.

Nella realtà ciò che importa maggiormente è sapere che la posizione del beam sia corretta perché, per ovvie ragioni, si sta considerando un fascio che incide su un campione e quindi non è tollerabile un errore di posizionamento dato da uno sbaglio grafico.

In questo caso quindi il problema dello sfasamento del target è solo di natura grafica, perché la posizione del beam rappresenta quella che effettivamente è la posizione del fascio nel sistema di riferimento, l'unico inconveniente è una complicazione nel fare assumere al beam la posizione desiderata dovuta all'impossibilità di fare coincidere i due cursori verde e rosso.

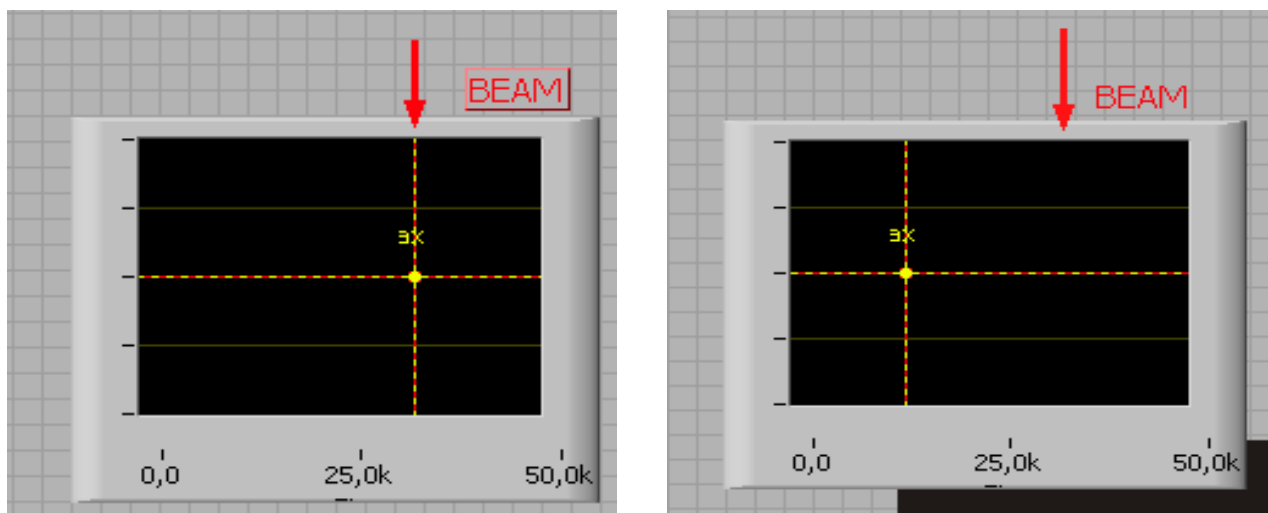
Per questo motivo è consigliato premere sempre il pulsante *Reset Target* dopo aver premuto *Go* ed essersi assicurati del corretto spostamento del beam sul target; da notare il fatto che premendo il pulsante *Set Zero* il target si resetta automaticamente senza dover ricorrere al comando specifico.

Nel caso in cui erroneamente si dimentica di premere *Reset Target* o per altri motivi si ha lo sfasamento dei cursori target e beam si può procedere in due modi: tramite lo scrollino o il valore inserito nella targhetta del target si sovrappone il cursore verde su quello rosso e si preme *Reset Target* (una soluzione che può essere definita 'grafica'), oppure una soluzione più drastica che prevede il riportare i motorini al limite negativo e chiudere il programma per poi riaprirlo con il file "INIT X-Y neg limit.txt"; il sistema è portato alla situazione di partenza nella quale target e beam si trovano sul punto in alto a destra del sistema di riferimento e lo zero relativo è già nella posizione che precedentemente è stata settata.

Per quanto riguarda la prima soluzione è utile notare che visto che la posizione del target è data dalla somma dei valori di *Last Position*, *Zero Position* e target, quando si ha uno sfasamento di target e beam i valori di *Last Position* e Beam Position differiscono di una quantità pari all'entità dello sfasamento, si può agire sulla targhetta di target e inserire il valore che permette di raggiungere in *Last Position* il valore di *Beam Position* dopo aver premuto *Reset Target*.

3.9 GRAFICO DEL COLLIMATORE

Fig 3.11



In figura 3.11 due versioni del grafico che rappresenta la posizione del collimatore finale rispetto al fascio indicato dalla freccia rossa, come in una visione dall'alto.

La linea orizzontale rappresenta il supporto del collimatore finale, la linea verticale corrisponde al punto in cui si ha il foro.

Se il fascio è allineato al foro, una parte di esso attraverserà il supporto e potrà così giungere al campione, come accade nel primo grafico, se invece non è allineato il collimatore finale fungerà da 'shutter', cioè da otturatore e non farà passare il fascio, come si può notare dal secondo grafico.

Durante un turno di misura può essere necessario interrompere temporaneamente o istantaneamente l'incidenza del fascio sul campione, per esempio quando il fascio è troppo intenso e potrebbe danneggiare un campione troppo sensibile; ciò che si fa in questi casi è movimentare il

motorino associato al supporto del collimatore in modo da 'stoppare' il fascio.

Per tenere conto delle esigenze operative sono state create due versioni del file di inizio, una con il fascio in posizione sul foro del collimatore: "INIT X-Y neg limit C in Beamline.txt" e una con il fascio completamente fuori dal sistema di movimentazione, che corrisponde al fatto che il terzo motorino si trova al suo limite positivo ("INIT X-Y neg limit.txt"). Se si vuole solo bloccare il fascio basta aprire la configurazione 'in Beamline' e spostare il motorino associato al collimatore di una posizione arbitraria a destra o a sinistra.

3.10 FILE DI INIZIALIZZAZIONE

Fig 3.12

[Section 0] Motor Number 1=1 Motor Number 2=2 Motor Number 3=3 Motor Number 4=4 [Section 1] Motor Name 1="X (Left-Right)" Motor Name 2="Y (Up-Down)" Motor Name 3=Collimator Motor Name 4=None [Section 2] Encoder Position 1=-12,1089 Encoder Position 2=-32,6986 Encoder Position 3=-18,3055 Encoder Position 4=0,0000 [Section 3] Zero Position 1=37,8911 Zero Position 2=17,3014 Zero Position 3=31,6945 Zero Position 4=50,0000	[Section 0] Motor Number 1=1 Motor Number 2=2 Motor Number 3=3 Motor Number 4=4 [Section 1] Motor Name 1="X (Left-Right)" Motor Name 2="Y (Up-Down)" Motor Name 3=Collimator Motor Name 4=None [Section 2] Encoder Position 1=-12,1089 Encoder Position 2=-32,6986 Encoder Position 3=0,0000 Encoder Position 4=0,0000 [Section 3] Zero Position 1=37,8911 Zero Position 2=17,3014 Zero Position 3=31,6945 Zero Position 4=50,0000
---	---

In figura 3.12 I due file di inizializzazione "INIT X-Y neg limit.txt" e "INIT X-Y neg limit C in Beamline.txt" rispettivamente.

La struttura del file di inizializzazione è molto semplice, ci sono 4 sezioni ognuna corrispondente ad un parametro che si vuole inizializzare con un valore ben preciso; la comodità sta nel fatto che si possono cambiare i valori direttamente sul file senza agire sul codice del programma.

Come si può ben notare il valore di *Encoder Position* 3 nel primo file è settato al valore che determina il limite positivo del terzo motorino, quindi il collimatore è completamente fuori dall'asse del fascio. Nel secondo file invece il valore di *Encoder Position* 3 è zero perché si è deciso di considerare come valore zero quello in cui il fascio è allineato al foro del collimatore.

3.11 CODICE

I seguenti paragrafi trattano dell'analisi passo per passo del codice del programma, dei vari costrutti che ne fanno parte, delle loro funzioni e come avviene lo scambio di dati tra essi. Inoltre per ogni sub.vi incontrato sarà fornita l'immagine e una breve spiegazione; come convenzione il simbolo @@ indicherà il percorso assoluto della cartella in cui sarà installato il programma.

Il block diagram è composto da una stack sequence di 4 pagine, numerate da 0 a 3.

Ogni pagina contiene al suo interno altre strutture raggruppate in sub sequenze, cicli while e for, case structure che permettono il fluire del programma in base all'ordine in cui si vuole, introducendo anche temporizzazioni e utilizzando una serie di funzioni e costrutti che permettono di implementare qualsiasi funzione occorre.

La prima pagina (la pagina 0) contiene la definizione delle strutture dati che saranno utilizzate nel corso del programma e due cicli per l'inizializzazione di array e strutture.

Nella seconda pagina (pag 1) c'è la lettura dal file e la lettura delle posizioni dall'encoder.

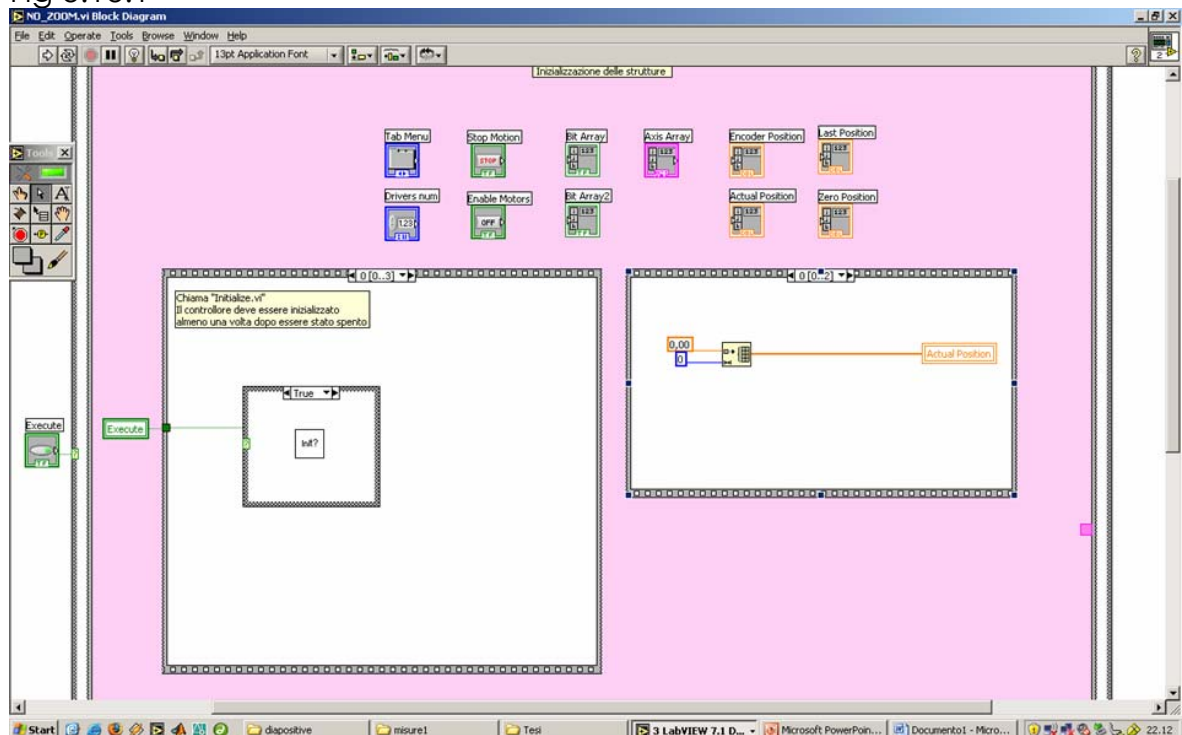
La pagina 3 (pag 2) è quella più onerosa dal punto di vista elaborativo perché comprende il cuore del programma, ossia la movimentazione vera e propria ed inoltre contiene il maggior numero di sotto strutture e di controlli.

La pagina 4 (pag 3) è quella che chiude il programma ed è costituita da due sottopagine.

A fine di analizzare in modo più comprensibile l'elaborato, alcune porzioni di codice grafico verranno espresse in pseudo codice C.

3.12 PAGINA 0

Fig 3.13.1



In figura 3.13.1 la pagina 0 della stack sequenze principale, di seguito le immagini delle sottostrutture contenute più nel dettaglio.

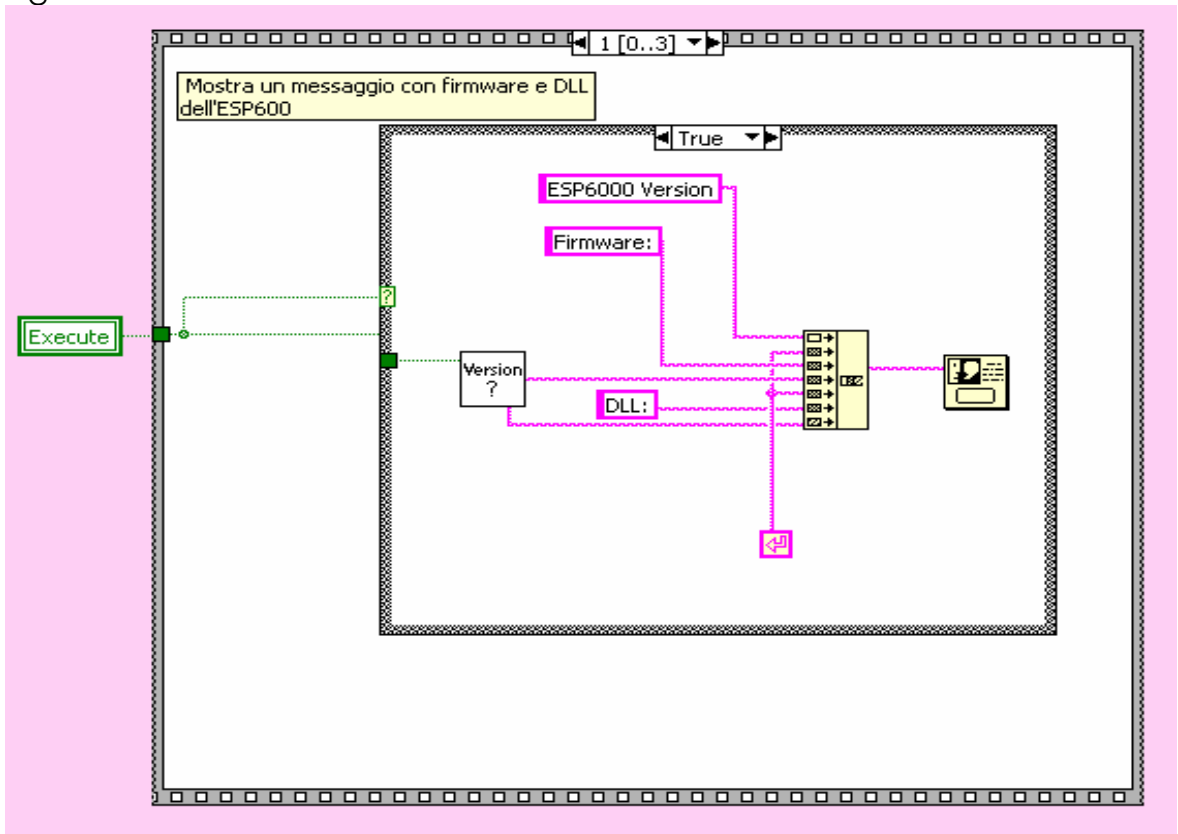


Init è il sub.vi che inizializza il controllore, questo deve essere fatto ogni volta che il controllore viene acceso.

Initialize-.vi

@@\LabVIEW7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Initialize-.vi

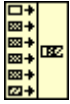
Fig 3.13.2



In questa pagina il sub.vi Version legge la versione del firmware e della dll e li fornisce in uscita alla finestra di dialogo tramite il 'concatenate string'.

Version, read.vi

@@\LabVIEW7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Version, read.vi

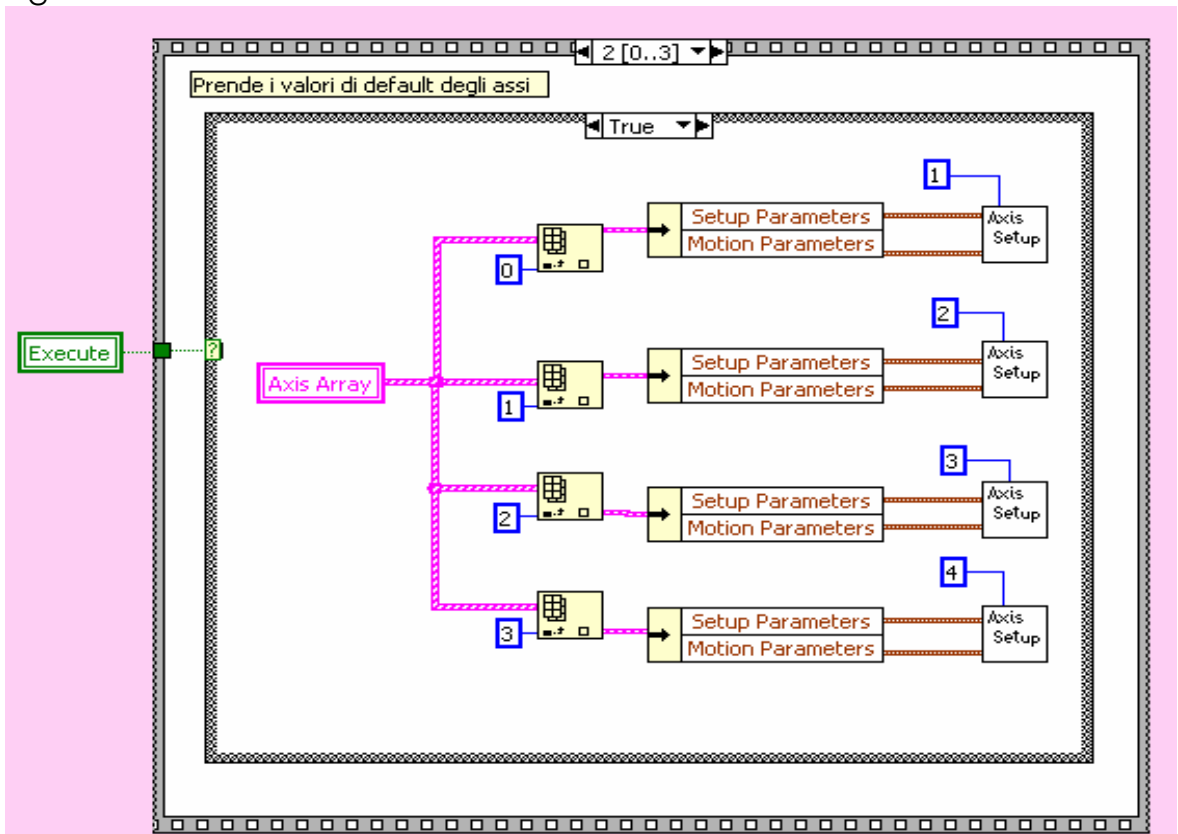


Concatenate Strings, prende in ingresso più stringhe e le concatena.



One Button Dialog contiene un messaggio e un pulsante singolo.

Fig 3.13.3



Inizializzazione dei parametri di ciascun motorino che compariranno nella tabella degli assi con i valori di default.



Index array ritorna in uscita l'elemento che ha come indice quello in ingresso; l'elemento in questo caso è il cluster da cui tramite l'unbundle by name si prendono i parametri di setup e motion.

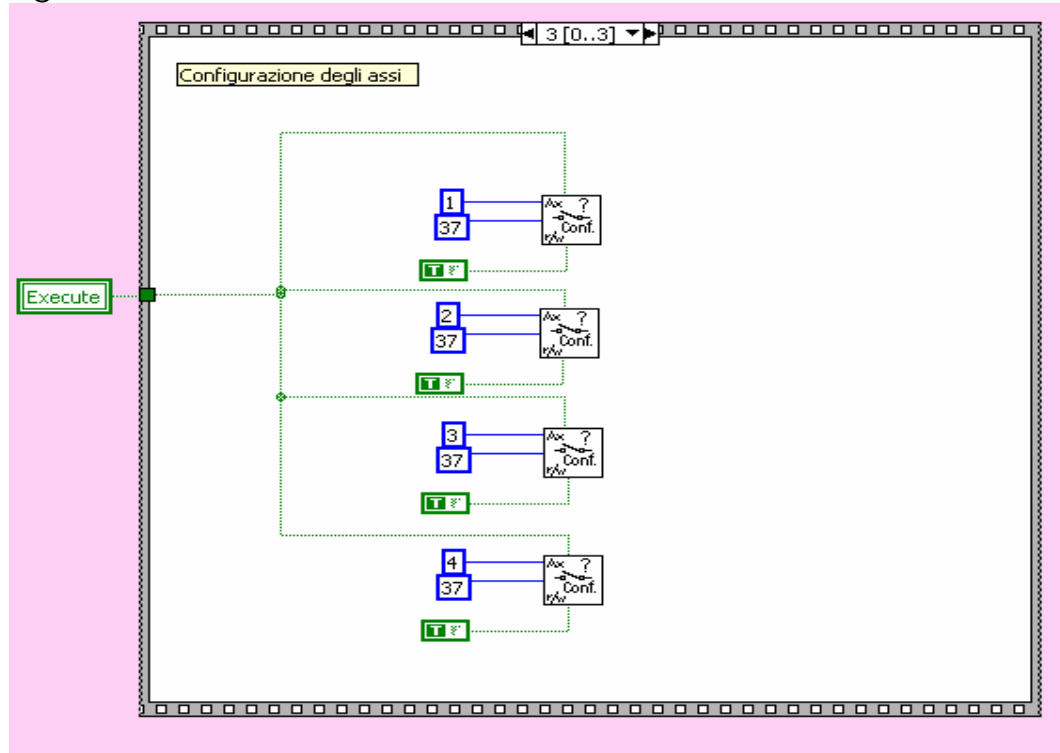


Axis setup è il sub.vi che registra i parametri di setup e di motion.

Axis, setup and motion.vi

@@\LabVIEW7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Axis,setup and motion.vi

Fig 3.13.4



Configurazione degli assi.

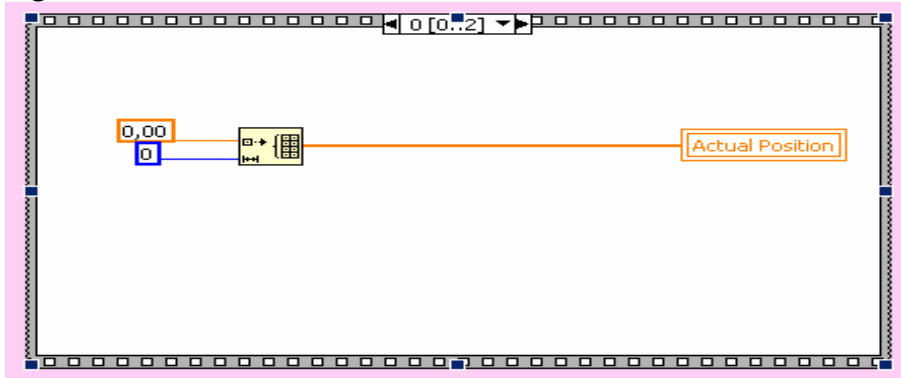


Hard-limit configuration.vi riguarda la configurazione dei limiti hardware, cioè come si comportano i motorini quando arrivano ai loro limiti.

Hard-limit configuration.vi

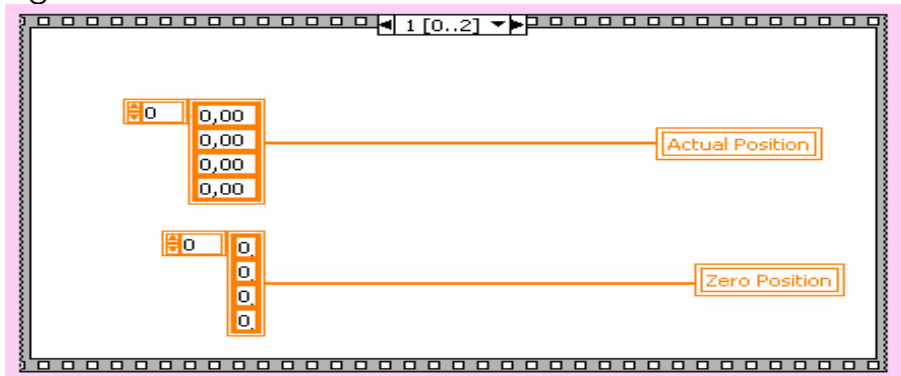
@@\LabVIEW7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Hard-limit configuration.vi

Fig 3.13.5



Initialize array crea un array di n elementi e li inizializza al valore 0.

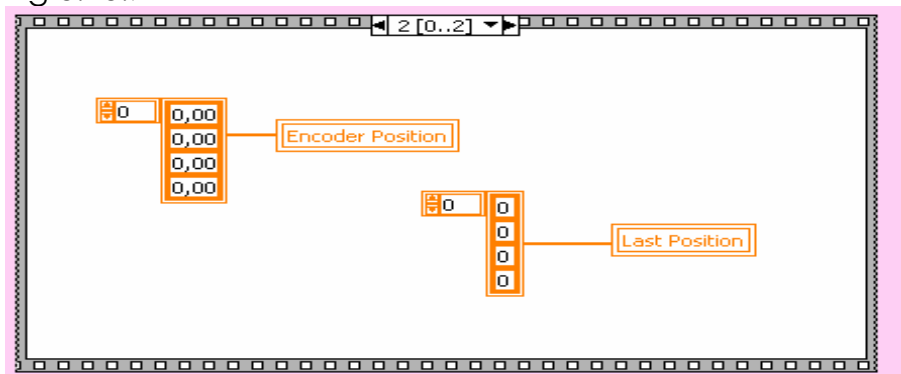
Fig 3.13.6



Gli arrays sono inizializzati al valore zero.

E' sempre importante inizializzare gli arrays in modo che abbiano un valore di partenza.

Fig 3.13.7



3.13 PAGINA 1

Fig 3.14a

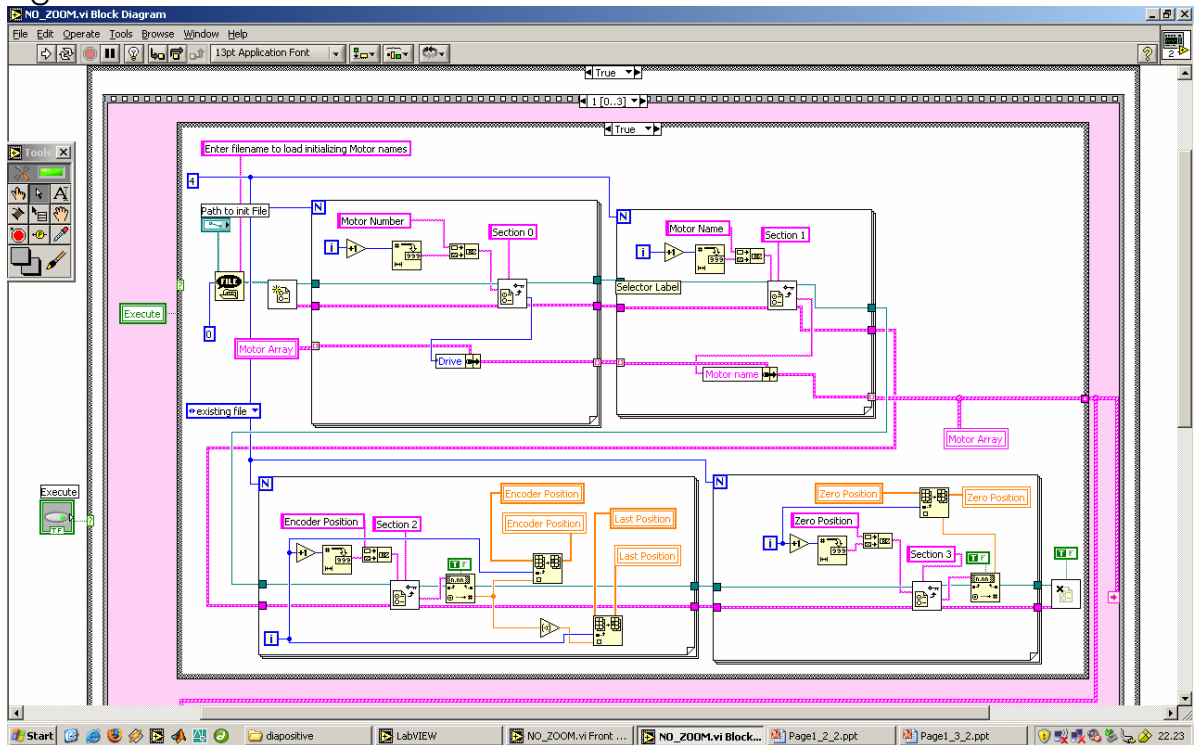
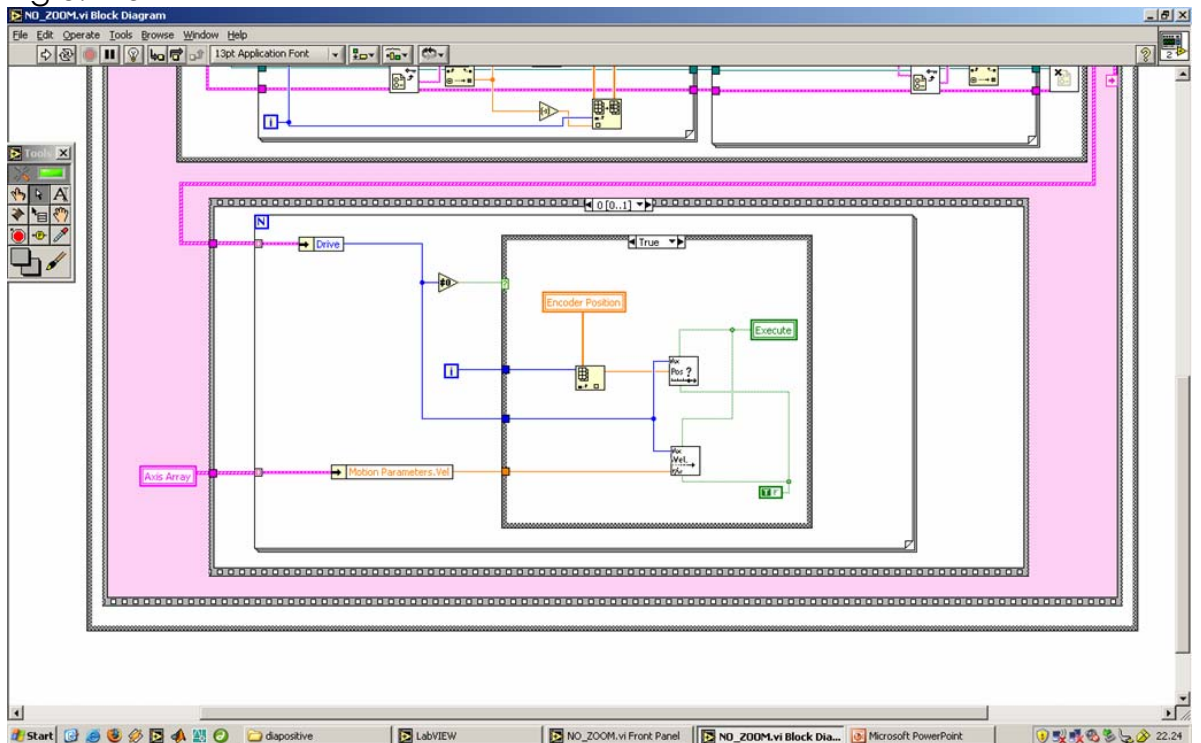
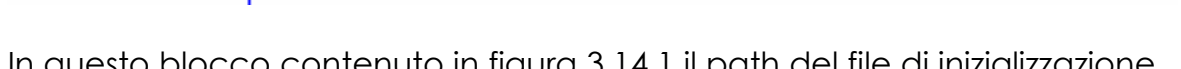


Fig 3.14b



Enter filename to save initializing Motor names



```
@@\LabVIEW 7.1\vi.lib\UTILITY\config.llb\Open Config Data.vi
```




Fract/Esp String To Number interpreta i caratteri da 0 a 9 , operatori somma e sottrazione, numero e, punti decimali in stringhe come un numero floating point in una notazione ingegneristica, esponenziale o in formato frazionario e lo ritorna in numero.



Replace Array Subset rimpiazza un elemento o un subarray da un array nel punto specificato dall'indice in ingresso con il valore in ingresso.

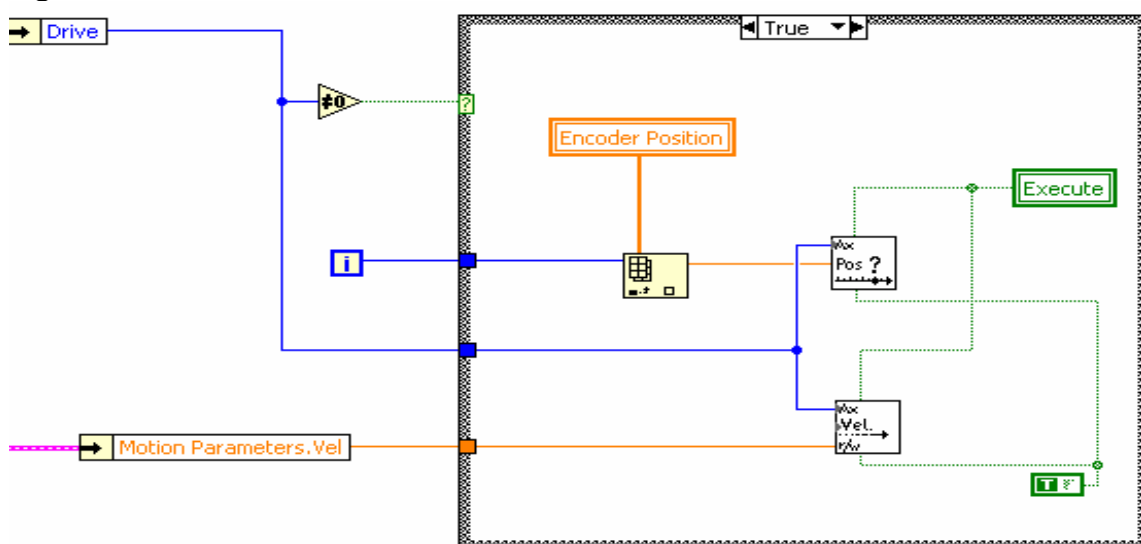


Close Config Data chiude la comunicazione con i dati di configurazione indicati da refnum (linea verde).

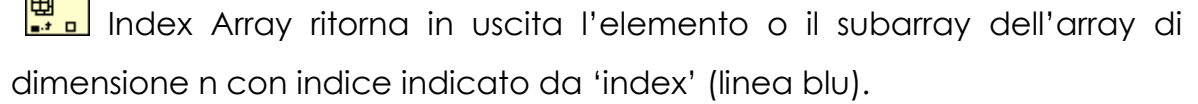
Close Config Data.vi

@\LabVIEW 7.1\vi.lib\utility\config.lib\Close Config Data.vi

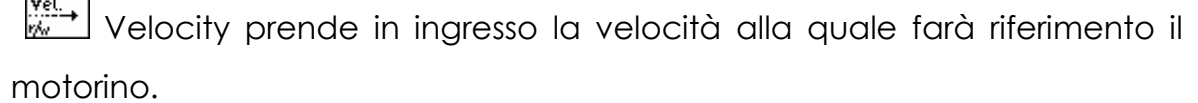
Fig 3.14.3



In figura 3.14.3 si vede che i parametri di velocità e posizione sono assegnati ad ogni motorino, grazie al fatto che ci troviamo all'interno di un ciclo for che va da 0 a 3. In ingresso all'array *Encoder Position* un indice permette di prelevare la giusta posizione per ogni motorino.



@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Position.vi



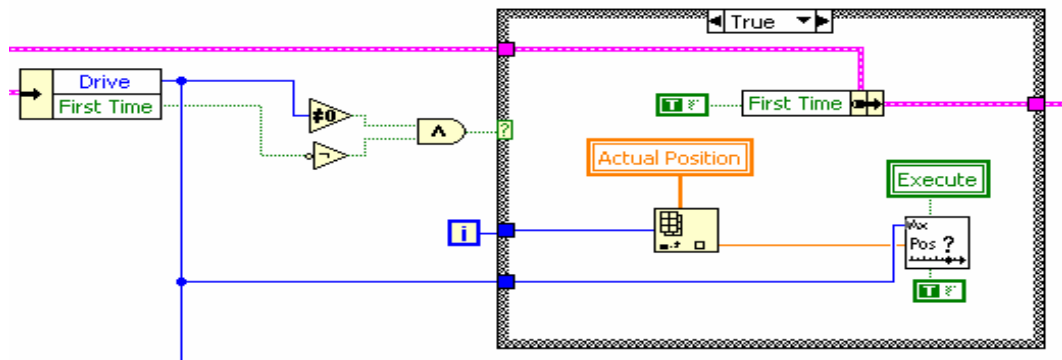
@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Velocity.vi

Fig 3.15



In figura 3.15 la terza pagina (pagina 2) è costituita da 9 sottopagine, la prima (fig 3.15.1) controlla se il numero del drive è cambiato rispetto al numero del motorino, questo garantisce che in caso si cambino i motorini rispetto alle posizioni del drive, verranno prese velocità, parametri pid e posizione del giusto motorino.

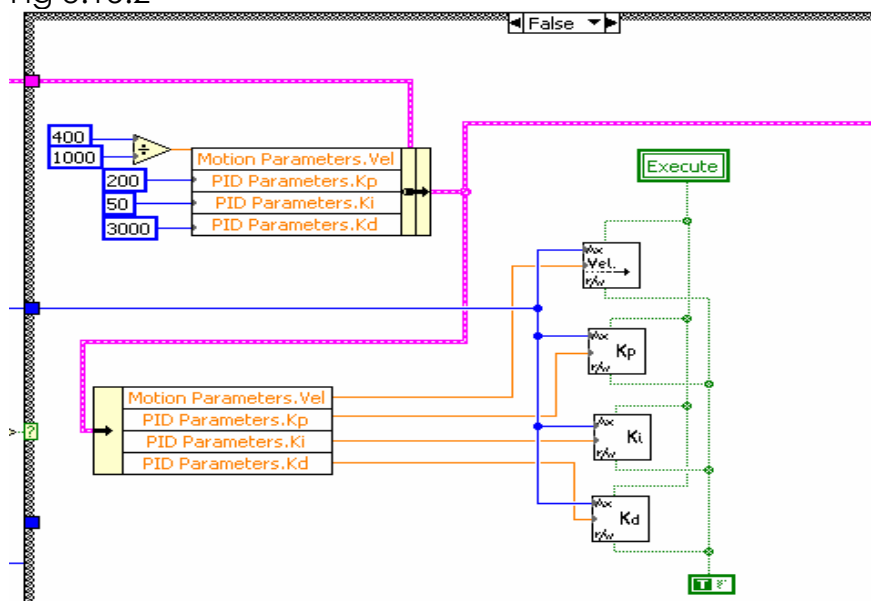
Fig 3.15.1



La variabile booleana *First Time* controlla se è la prima volta in cui al drive viene associato un determinato motorino, se non è la prima volta viene data al driver la posizione corrispondente e la variabile posta a true, altrimenti non si fa niente perché sono validi i valori preimpostati.

Però si deve fare un controllo circa il tipo di motorino dato che quelli a disposizione non sono tutti uguali.

Fig 3.15.2



In figura 3.15.2 il blocco in cui sono assegnati i parametri motion e PID ai corrispondenti ai motorini dopo il controllo di *First Time*.



Kp gain.vi, assegna il valore in ingresso (linea arancione) al parametro proporzionale Kp del motorino corrispondente al numero di asse indicato dalla linea blu in ingresso.



Ki gain.vi, assegna un valore al parametro integrativo Ki.



Kd gain.vi, assegna un valore al parametro derivativo Kd.

Di seguito una porzione di pseudo codice che implementa la pagina di figura 3.15.2.

```
[
    boolean First Time;
    if (First Time==TRUE){
        <prendi la posizione del nuovo motorino e mettila
        al posto indicato da
        drive>
    }
    else{
        <la posizione rimane la stessa di prima visto che il
        motorino non è cambiato>
    }

    if (HS Drive == Drive){          //HS Drive =3
        <cambia i valori di parametri PID e motion in quelli del
        motorino più veloce>
    }
    else{
```

```

    <mette i valori di parametri PID e motion corrispondenti
    del motorino più lento>
  }
]

```

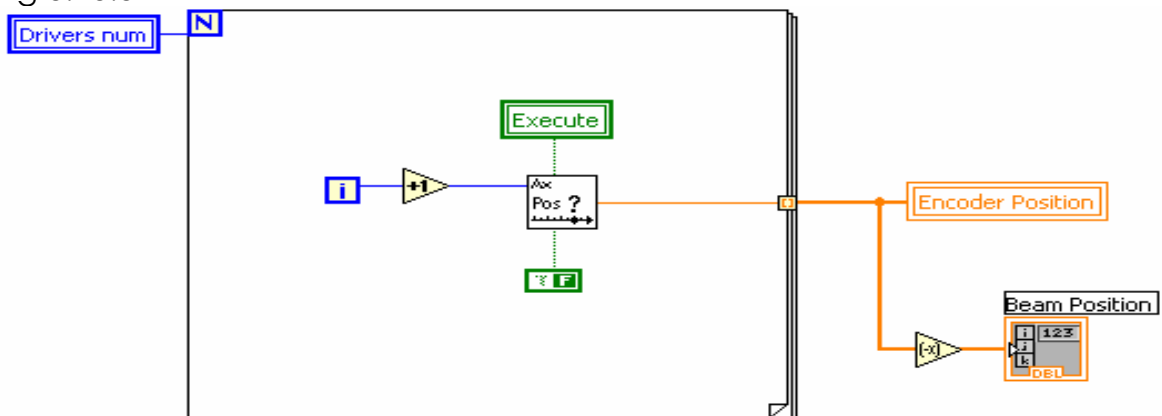
I parametri da considerare sono la velocità, i motorini più lenti hanno una velocità limite di 0.8 mm/s anche se in questo programma si adotta una velocità di 0.4 mm/s, il motorino veloce ha una velocità limite di 3 mm/s.

Anche per quanto riguarda i parametri PID c'è una sostanziale differenza, dovuta al fatto che ogni motorino ha le proprie caratteristiche e quindi un assetto diverso che può essere impostato modificando i valori dei parametri PID e facendo varie prove.

In questo programma si è scelto di utilizzare i parametri PID di default del controllore dopo aver testato che effettivamente tali sono i valori più adatti in seguito a più prove effettuate cambiando i valori.

Infatti per valori diversi da quelli di default si è notato che il motorino ha difficoltà a trovare la posizione corretta e continua ad oscillare attorno alla posizione di arrivo.

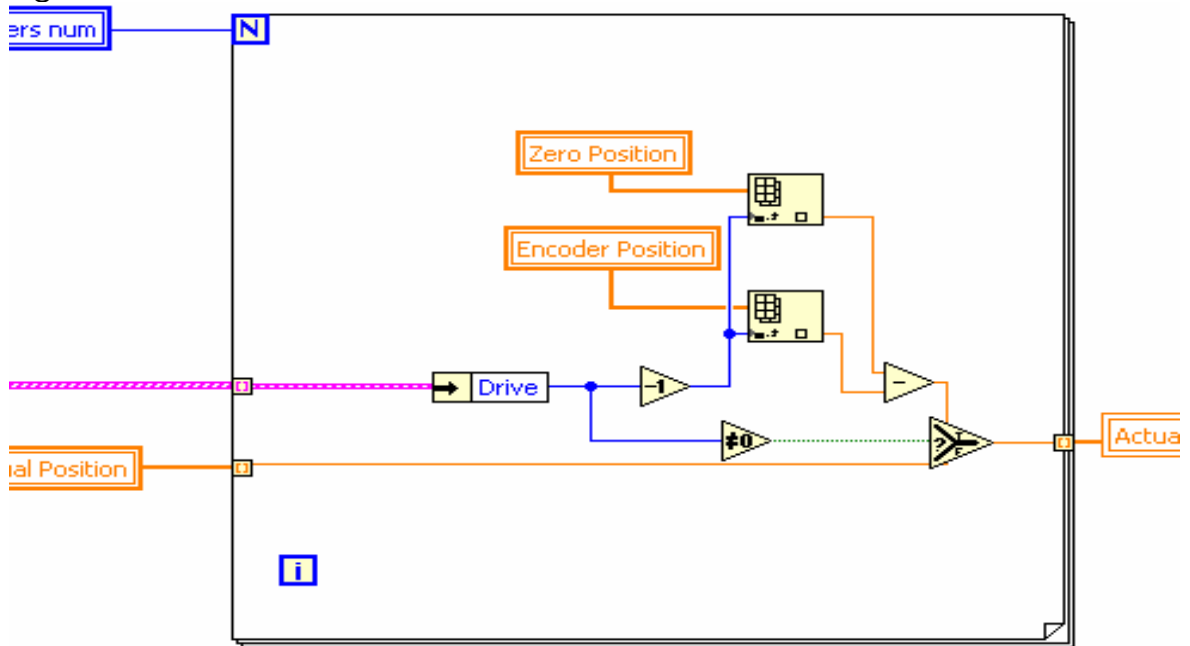
Fig 3.15.3



In figura 3.15.3 la seconda sottopagina del frame 2.

Per ogni driver la posizione attuale del motorino è inserita negli arrays *Encoder Position* e *Beam Position*, in questo ultimo la posizione è negata per esigenze grafiche che si spiegheranno in dettaglio più avanti.

Fig 3.15.4

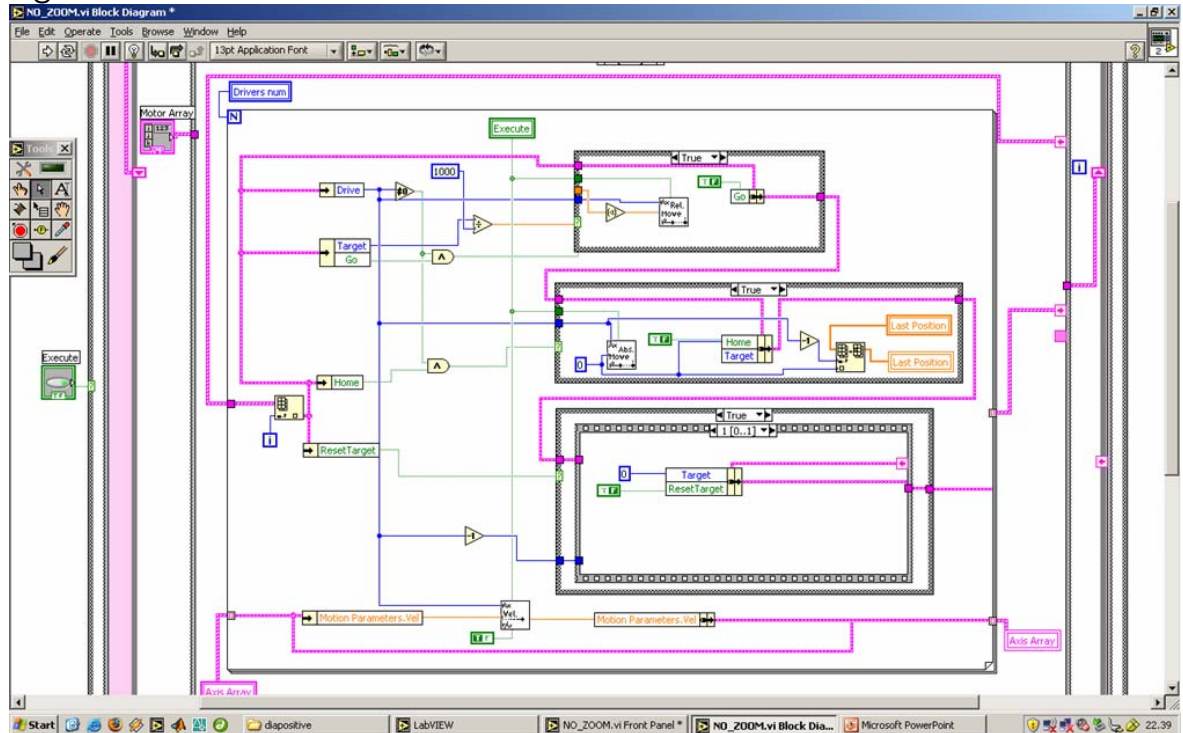


Nel blocco contenuto in figura 3.15.4, nell'array *Actual Position* viene assegnata la posizione data da una combinazione tra la posizione dello zero e quella data dall'encoder.

Questo è dovuto al fatto che *Actual Position* è un array dinamico che cambia ogni volta che un motorino si muove, di pari passo all'array *Encoder Position* che prende il suo valore direttamente dall'encoder e quindi cambia in real time. Gli arrays *Zero Position* e *Last Position* invece sono statici, essi assumono un valore determinato e possono essere modificati solo all'occorrenza di un particolare evento come la premitura di un pulsante.

L'array *Beam Position* è anch'esso dinamico e i suoi valori sono dati dal negato dei valori dell'array *Encoder Position*.

Fig 3.15.5



In figura 3.15.5 sono definite le operazioni che vengono eseguite quando si premono i pulsanti *Go*, *Home* e *Reset Target*.

Il pulsante *Go* permette di spostare il motorino di un valore pari a quello inserito dall'utente sulla targhetta corrispondente dell'interfaccia.



Move to relative position.vi

Move to relative position.vi

@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Move to relative position.vi

Il pulsante *Home* ha la funzione di far tornare i motorini alla posizione zero.



Move to absolute position.vi

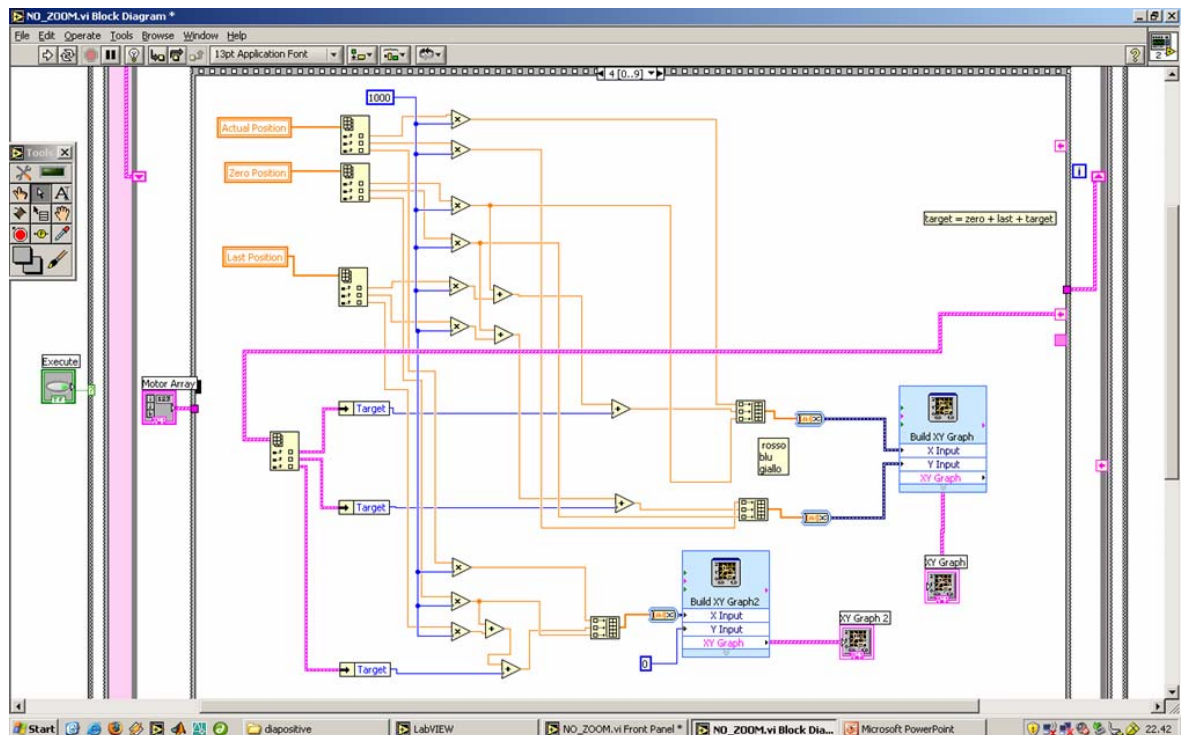
Move to absolute position.vi

@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Move to absolute position.vi

Il pulsante *Reset Target* resetta il target al valore corrente del *Beam Position*.

Tutti i pulsanti sono di tipo booleano, in partenza sono nello stato logico false, quando si preme passa allo stato true e attiva la case structure corrispondente, all'interno della quale viene di nuovo posto a false.

Fig 3.15.6



Nella pagina di figura 3.15.6 sono inseriti i due grafici che sono presenti nel programma, il primo riguarda la posizione del campione nel sistema di riferimento e presenta 3 plot, il primo per la posizione del fascio rispetto allo zero precedentemente fissato che prende le coordinate dall'array *Actual Position* ed è rappresentato dal cursore rosso; il secondo per il target che rappresenta lo spostamento relativo dalla posizione del fascio ed è dato dalla somma del valore inserito nella targhetta corrispondente, la posizione dello zero e l'ultima posizione assunta dal fascio ed è rappresentato dal cursore verde; il terzo per la posizione che si sceglie di settare come posizione zero del sistema di riferimento relativo che è dato dal valore fornito dall'array *Zero Position* ed è rappresentato dal cursore giallo.

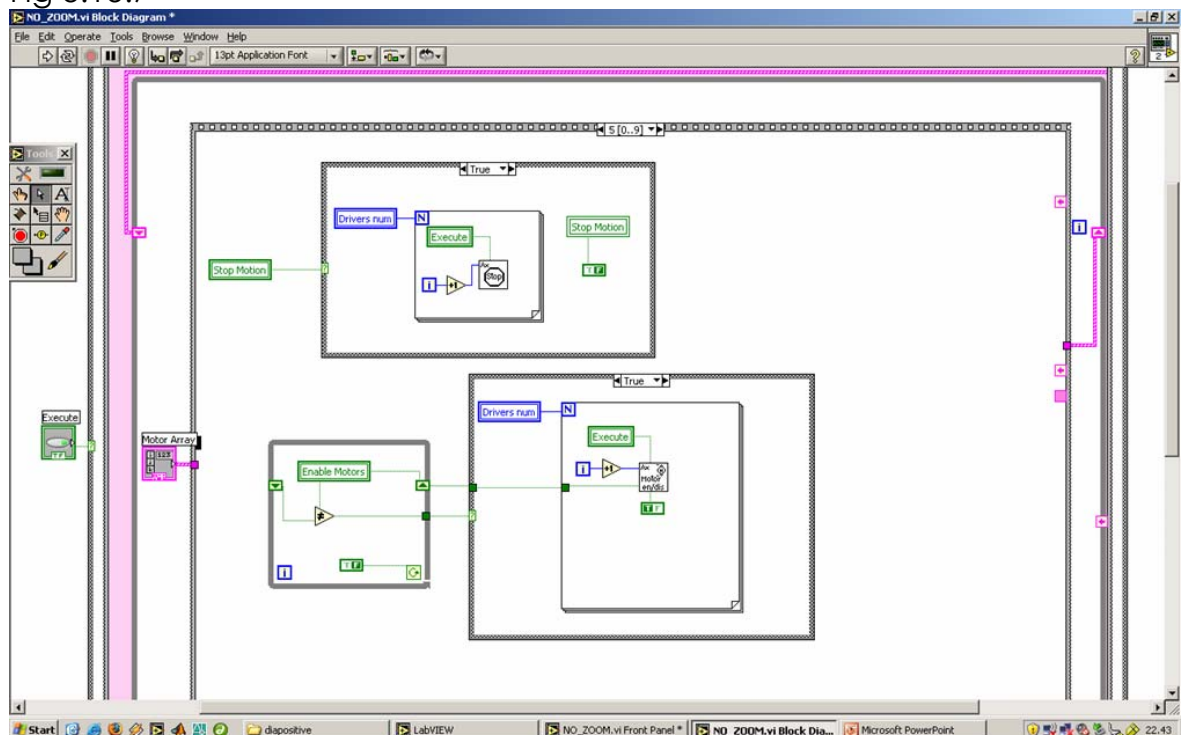
Per questo grafico si è creata un'altra versione del programma che permette di fare uno zoom nella regione interessata, facendo partire il grafico da una posizione poco antecedente rispetto allo zero relativo del sistema ().

Il secondo grafico rappresenta la posizione del collimatore finale rispetto alla posizione del fascio.

 Build Array concatena array multipli o aggiunge elementi ad un array di dimensione n.

 Convert To Dynamic Data converte dati di tipo numerico, booleano, array e funzioni in dati di tipo dinamico da utilizzare nei grafici.

Fig 3.15.7



Nella figura 3.15.7 il primo blocco permette di stoppare i motorini durante il movimento in qualsiasi momento premendo il pulsante *Stop Motion*.

Gli altri due blocchi riguardano il controllo dello stato dei motorini, se sono accesi anche i led corrispondenti lo saranno.



Stop motion.vi serve per stoppare il movimento quando si vuole.

Stop motion.vi

@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Stop motion.vi

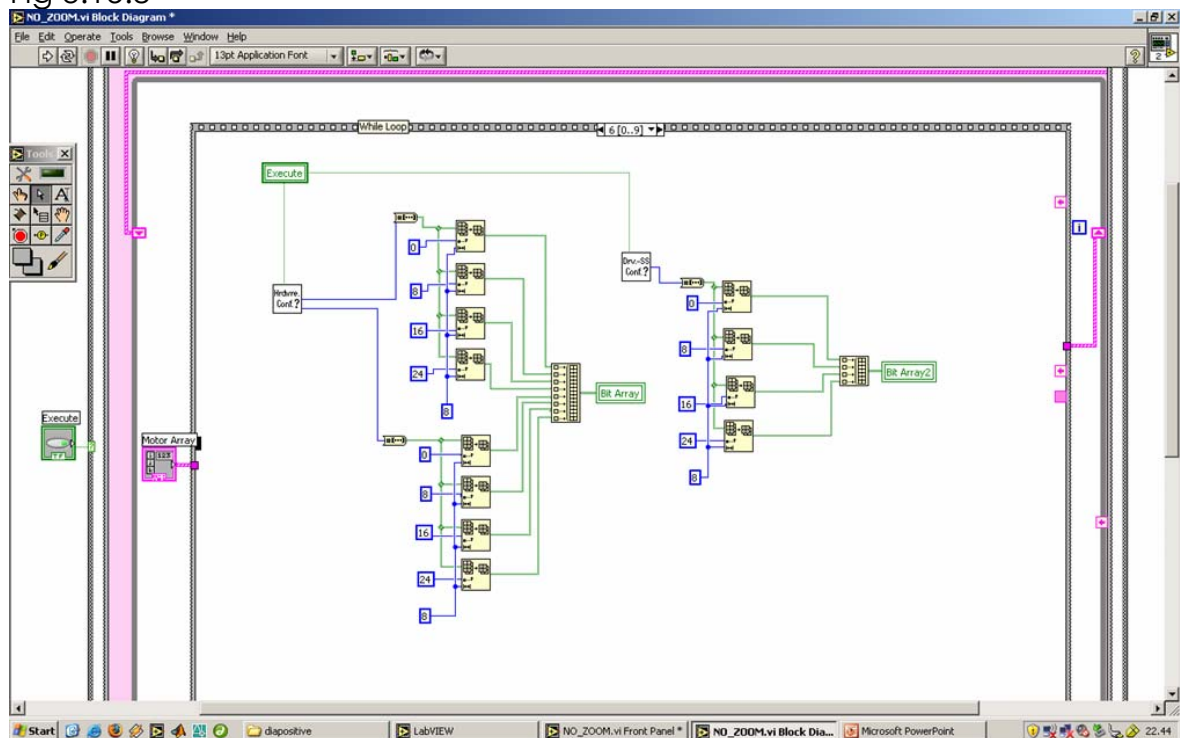


Motor, enable-disable.vi

Motor, enable-disable.vi

@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Motor, enabledisable.vi

Fig 3.15.8



Questa parte del programma in figura 3.15.8 svolge un'importante funzione di informazione perché attraverso dei bit di controllo che dialogano direttamente con il driver è possibile vedere ad esempio se un motorino ha raggiunto uno dei suoi limiti (superiore o inferiore) o se è effettivamente presente.



@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\Hardware status.vi



@@\LabVIEW 7.1\user.lib\Esp6000(LV4)\Esp6000(LV4)\System configuration vi

#[...]

In a

Fig 3.15.10

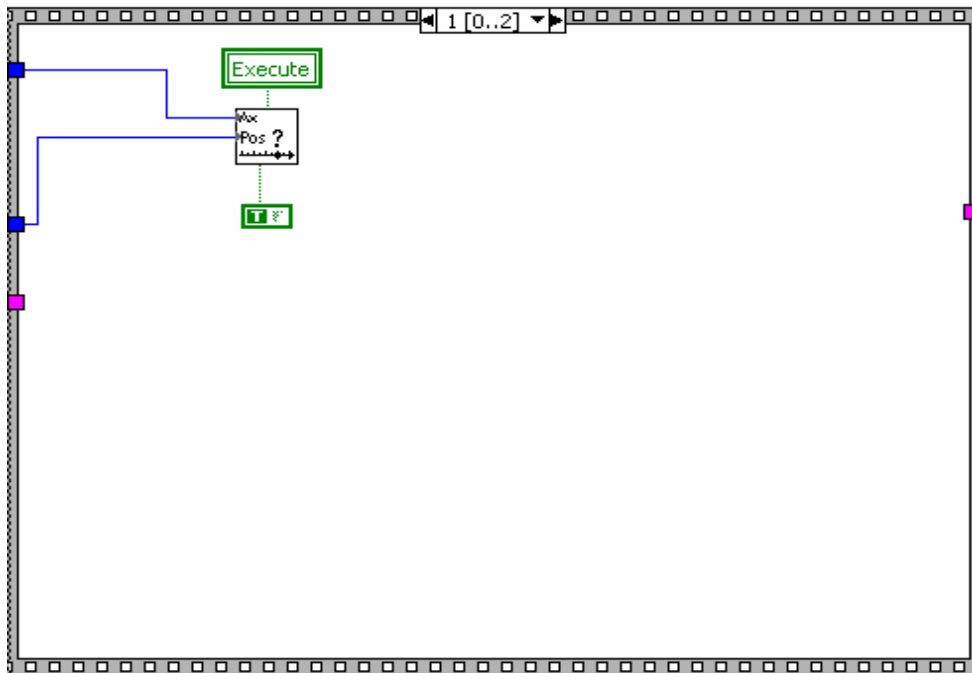
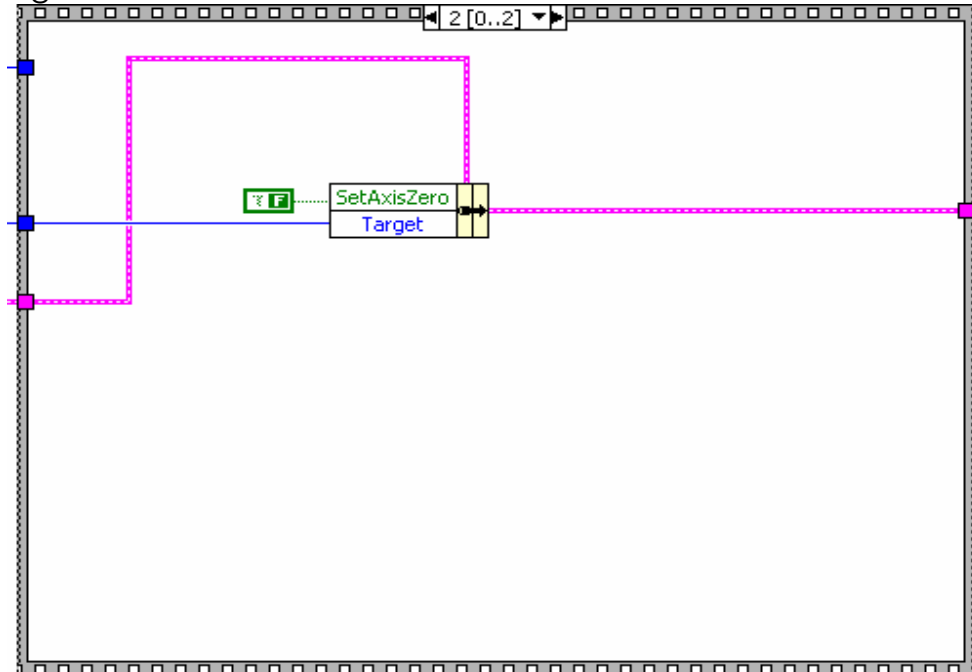
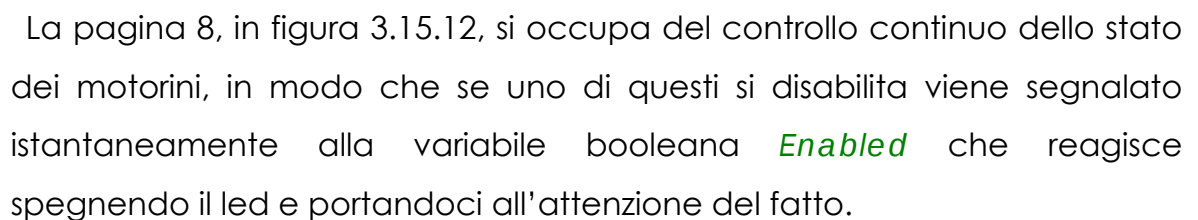


Fig 3.15.11



Di conseguenza sia la posizione del beam che il target si portano alla posizione zero e da quel momento il valore della posizione del beam sarà

Fig 3.15.12



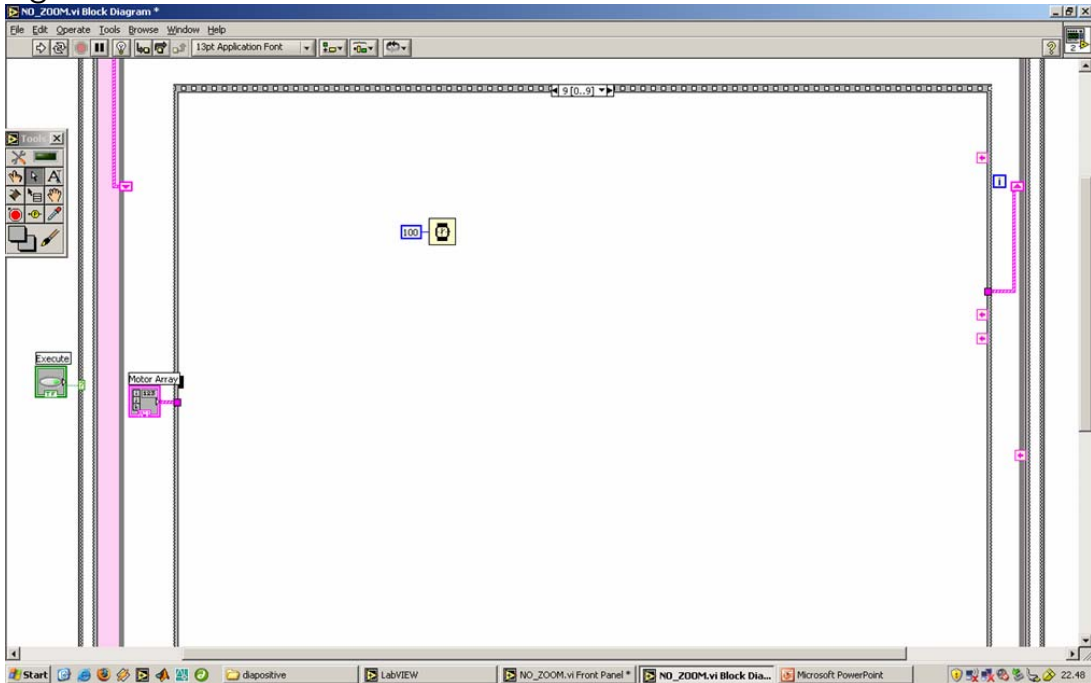
79

chiamate convenzionali. Ci sono paia di terminali di input e output, si può usare uno o entrambi i terminali. Ogni paio di terminali che viene aggiunto corrisponde ad un parametro della lista dei parametri della funzione che si vuole chiamare in ordine dall'alto in basso. Facendo click destro sul nodo e selezionando 'Configure' dal menu compare la finestra 'Call Library Function', in cui si può specificare il nome della libreria o il path, il nome della funzione , i parametri le chiamate convenzionali e il valore di ritorno del nodo.

Di seguito la porzione di pseudo codice che rappresenta il ciclo for di figura 3.15.12:

```
[
    boolean led[N-1];
    for(i=0; i<N; i++){
        <controlla lo stato di ciascun motorino>
        if (motorino[i]=enabled)
            <il led corrispondente si accende>
            led[i]=ON;
        else
            <il led corrispondente si spegne>
            led[i]=OFF;
    }
]
```

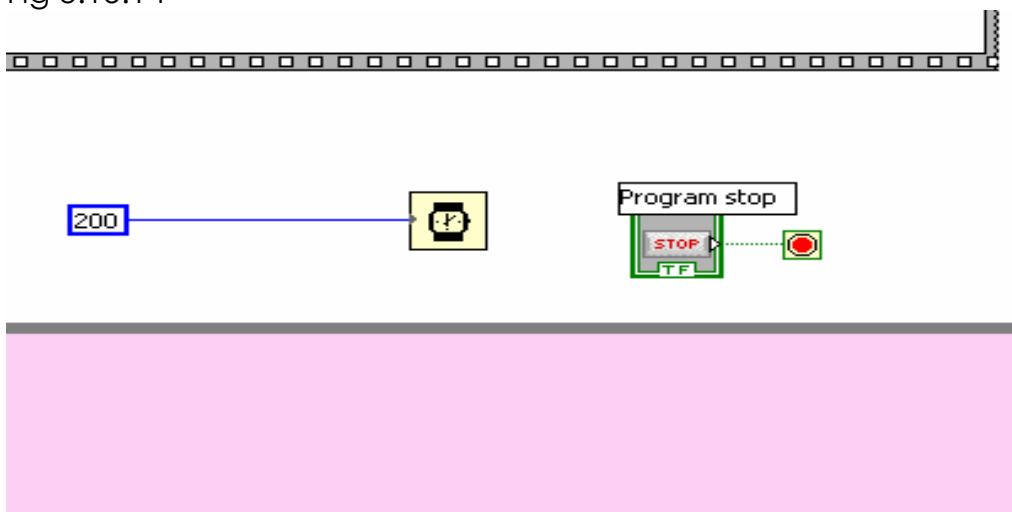
Fig 3.15.13



Quest'ultima parte del frame 2 (figura 3.15.13) introduce una temporizzazione, cioè si impone che prima di ripetere il ciclo debbano passare 100 microsecondi.



Fig 3.15.14



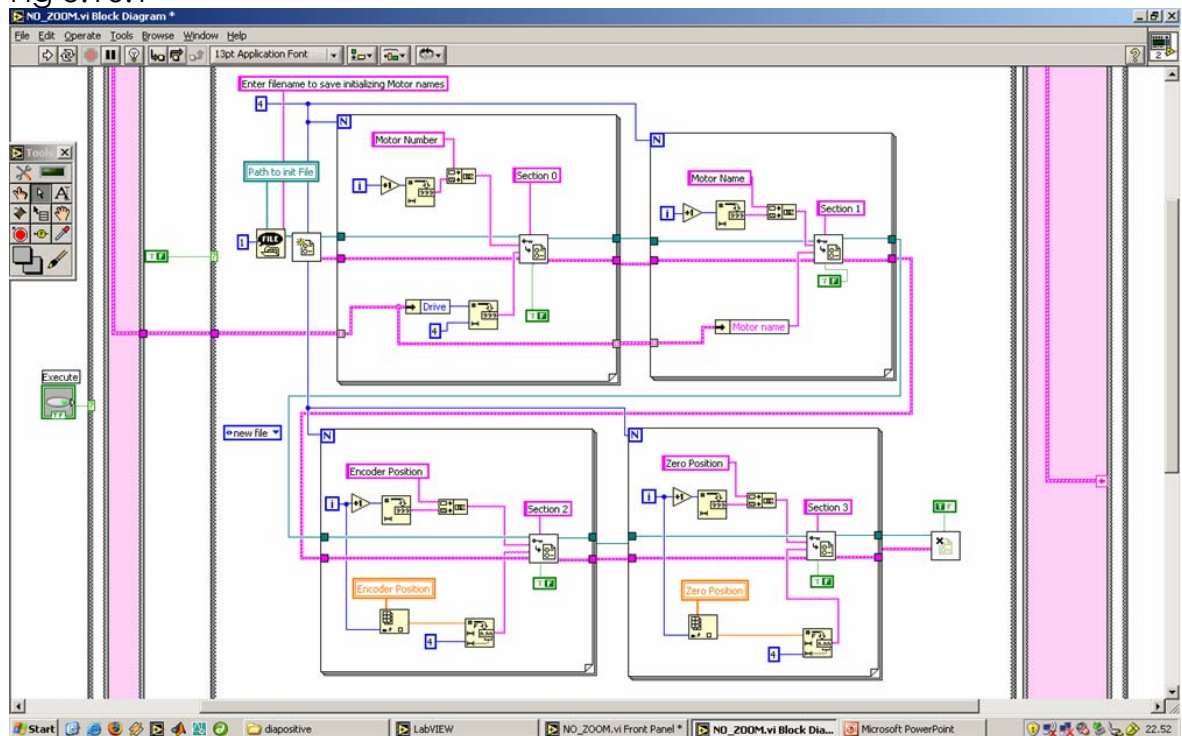
Quando si preme il pulsante *Program Stop*, questo viene riportato al valore false.



Two Button Dialog mostra una finestra di dialogo che contiene un messaggio ('Do you want to disable all motors? ') e due pulsanti (Yes, No). Yes implica la disabilitazione di tutti i motorini.

```
[
    char c;
    char motorino[N-1];
    printf("Do you want to disable all motors?(YES/NO) : ");
    scanf("%c", &c);
    if (c==YES){
        for (i=0; i<N; i++){          //N=Drivers num
            <I motorini sono disabilitati>
            motorino [i]=disabled;
        }
    }
    else
    {
        <non fa niente>
    }
]
```

Fig 3.16.1



Nella seconda pagina (figura 3.16.1) si conservano i valori del numero e del nome dei motorini e delle posizioni nel file di uscita.

Il file di uscita quindi conserva i valori di stato del programma nel momento in cui chiudiamo l'applicazione; ciò vuol dire che in qualsiasi momento possiamo ripartire dallo stato in cui si era lasciato.



Number To Fractional String converte i numeri in notazione frazionale o stringhe floating point.



Write Key

Write Key.vi

@@\LabVIEW 7.1\vi.lib\UTILITY\config.llb\Write Key.vi

<VI Revision History>

CAPITOLO 4 – Turno di Misura

Nel presente capitolo si espone lo sviluppo del progetto in sede della preparazione e dello svolgimento del turno di misura che ha visto la prima applicazione del programma di movimentazione sviluppato e si chiude con una breve illustrazione dei risultati ottenuti.

4.1 PROPOSTA DI LAVORO

Il titolo della proposta di lavoro è : Characterization of new position sensitive monolithic silicon telescopes.

I rivelatori in esame, di tipo semiconduttore, costituiscono un particolare tipo di telescopio monolitico sensibile alla posizione (dal progetto MONTE Monolithic Telescope) e sono costituiti da due giunzioni p-n chiamate ΔE ed E integrate in un comune chip di silicio.

Quando le particelle accelerate colpiscono la regione di svuotamento delle giunzioni, dall'interazione elettromagnetica si produce energia che rompe i legami covalenti permettendo la creazione di coppie elettrone/lacuna e una variazione di potenziale che viene acquisita, quantificata tramite convertitori ADC e digitalizzata.

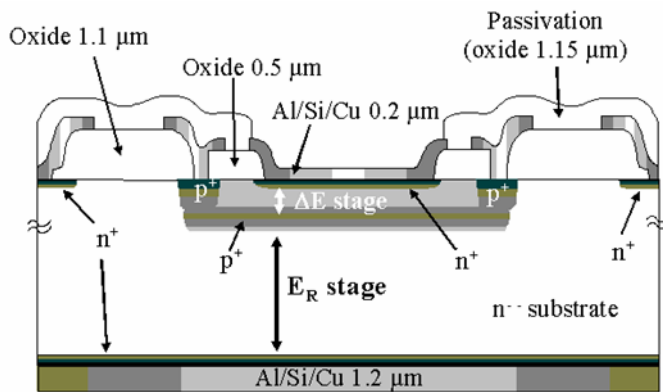
L'energia che passa dalla regione ΔE ha una piccola perdita che dipende dallo spessore di suddetta regione, in E invece essa viene interamente assorbita.

La regione ΔE dei campioni in esame è molto sottile, circa $2\text{ }\mu\text{m}$ ed è ottenuta tramite impiantazione ionica. Una regione di tipo p+ costituisce l'elettrodo ground comune alle due differenti regioni ΔE ed E .

L'aspetto di primaria importanza è l'effetto induttivo che si crea dalla zona E alle strip ΔE e dovuto alla resistenza finita dell' elettrodo ground in comune.

I nuovi rivelatori sensibili alla posizione consistono di 16 strip ΔE verticali impiantate in una piattaforma comune E .

Fig 4.1



La posizione x è fornita dalla strip irradiata, mentre la posizione y dovrebbe essere fornita dal segnale di induzione lungo ogni strip.

Sono stati creati tre prototipi di rivelatori con differente geometria e parametri di produzione, con lo scopo di trovare il miglior modo per fornire il segnale di posizione.

L'obiettivo del turno di misura è quello di testare le caratteristiche dei rivelatori e il loro generale comportamento, controllando la risposta di ogni strip irradiata in diversi punti da un fascio collimato, per dare un'accurata risposta di posizione ed escludere una dipendenza del segnale da qualsiasi altro fattore.

E' stato proposto un microfascio di ^{12}C a 50 MeV per irradiare 16 punti diversi per ogni strip in esame.

[Francesca Amorini, 2005]

4.2 SETUP TURNO DI MISURA

In un primo momento ci si è occupati della preparazione della camera 2000 [Appendice A] e degli strumenti necessari.

Il sistema di movimentazione con piastre, attuatori e supporti è stato posto tra i due bracci meccanici internamente alla camera e sottoposti a livellazione (figura 4.2).

Fig 4.2



4.3 ALLINEAMENTO

Una delle cose più delicate e di fondamentale importanza per la riuscita di qualsiasi esperimento di precisione è l'allineamento del fascio. L'asse del fascio deve essere perfettamente allineato all'asse geometrico del sistema altrimenti non arriverà niente a destinazione, considerando anche che il fascio deve passare attraverso dei collimatori per essere ridotto. Inoltre il fascio è soggetto a deviazioni e divergenza e per essere focalizzato passa attraverso quadrupoli, che sono particolari lenti adatte a questo scopo.

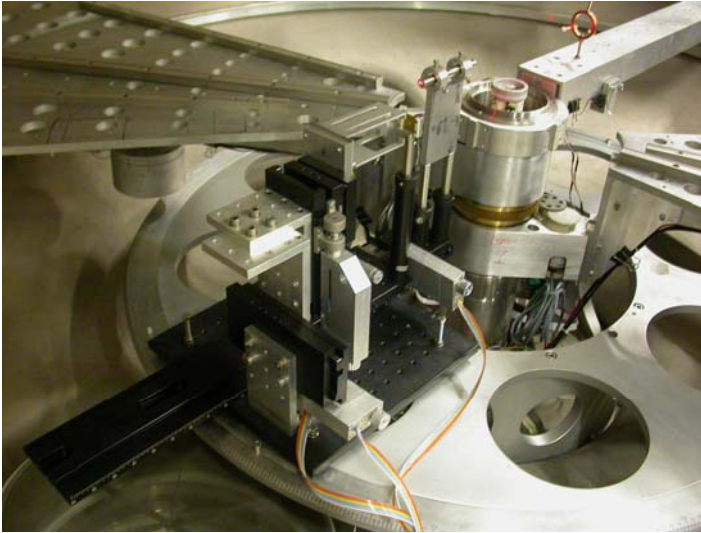
Fig 4.3



In figura 4.3 il laser è posizionato sulla linea del fascio.

Si è effettuato un allineamento preliminare con il laser; anche qui si presenta il problema del posizionamento corretto del laser con la linea di fascio e l'allineamento con il precollimatore di lunghezza 10 cm e diametro 1 mm.

Fig 4.3.1



In figura 4.3.1 si può ben vedere il puntino rosso prodotto dal laser che passa attraverso il collimatore di 1 mm.

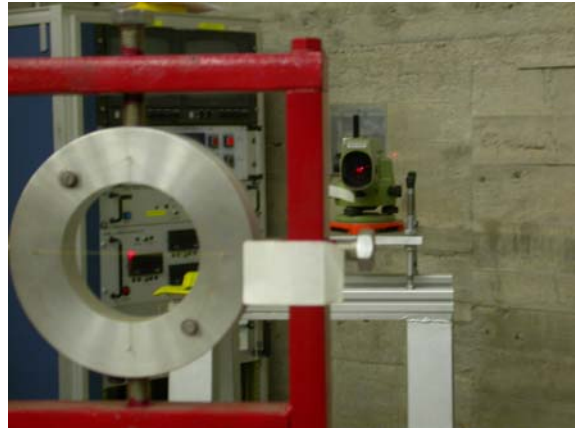
Tra gli strumenti adottati per un più corretto allineamento ci sono il teodolite e una croce di riferimento anteposta; come si può ben notare nelle figure 4.3.2 e 4.3.3 il laser per essere allineato deve passare al centro della crocetta ed essere acquisito dal teodolite.

Il teodolite è un particolare cannocchiale allineato alla crocetta in modo da vedere al centro dello schermo il centro della croce, il tutto perfettamente in asse con la linea del fascio.

Fig 4.3.2



Fig 4.3.3



Dopo aver accertato la corretta posizione del laser al centro della crocetta si è passati al montaggio del cannocchiale interno contenente due collimatori di 2 mm ciascuno e fissato ad un'estremità alla linea di fascio e all'altra estremità con un anello metallico e tre viti, una metallica e due di plastica (figura 4.3.4).

Quindi un nuovo allineamento per approssimazioni successive del laser che passa attraverso cannocchiale, precollimatore e collimatore finale dal diametro irregolare che va dai 100 ai 60 μm (perché fatto con uno spillo).

Fig 4.3.4

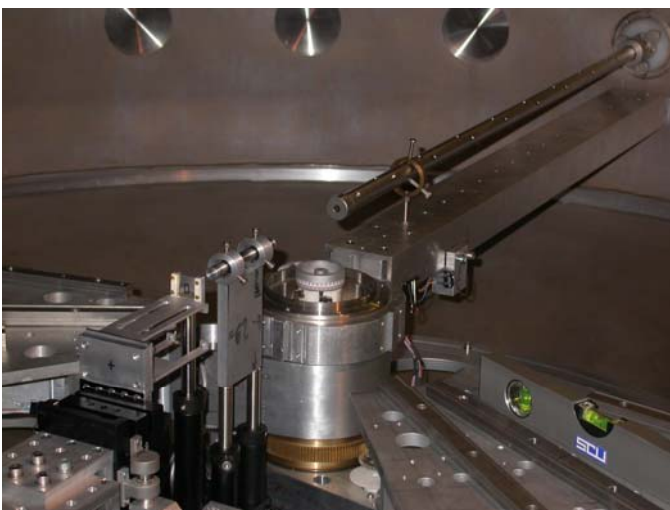


Fig 4.3.5



In figura 4.3.5 il laser attraversa precollimatore e collimatore finale.

4.4 SFOP

Dopo aver effettuato l'allineamento si passa al posizionamento dello SFOP (figura 4.4) in uscita al precollimatore e visualizzazione dello spot prodotto tramite telecamera.

Fig 4.4



Fig 4.4.1



In seguito lo SFOP è posto in uscita al collimatore finale e vediamo che lo spot che si produce è più piccolo e anche meno intenso, dovuta al fatto che quando una luce, un fascio di particelle, colpisce un foro delle dimensioni paragonabili alla sua lunghezza d'onda si crea l'effetto della diffrazione.

Infatti quando il fascio, che è intenso, passa attraverso l'ultimo collimatore, la maggior parte delle particelle è diretta, altre sono deflesse di vari angoli, in base alla distanza. Noi cerchiamo di ottenere deflessione zero, cioè minima dispersione del fascio, in modo che al rivelatore arrivino solo le particelle dirette.

Le particelle poi saranno conteggiate tramite convertitori ADC.

In figura 4.4.1 una visione dall'esterno della camera 2000 dello SFOP.

4.5 SISTEMA DI MOVIMENTAZIONE

Dopo l'installazione della flangia (figure 4.5 e 4.5.1) con connettori per collegare all'esterno gli attuatori, abbiamo collegato questi con il driver a sua volta collegato al PC e posto tutto esternamente alla camera.

Fig 4.5

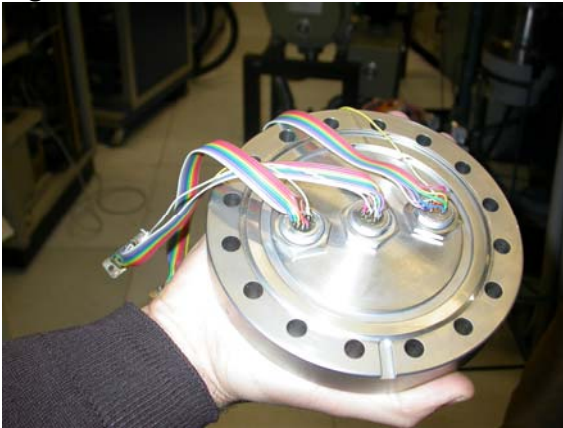
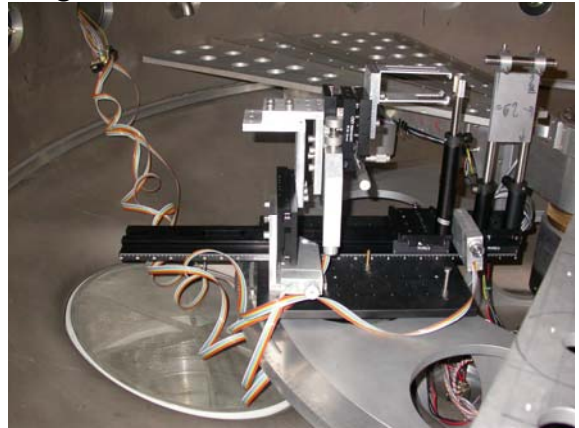


Fig 4.5.1



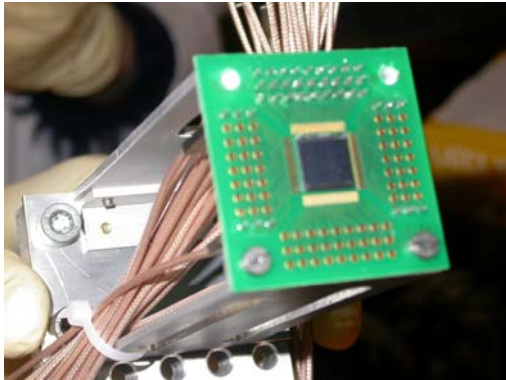
Una volta iniziato ad effettuare spostamenti per verificare la validità del sistema, si è cercato di valutare nel miglior modo possibile la posizione iniziale da cui partiranno le piastre e quindi il campione facendo sempre considerazioni sull'escursione massima dei motorini e sul range di nostro interesse.

Per prima cosa si è fissato un punto al centro del campione e uno in basso a sinistra in modo approssimativo. Salvate le configurazioni in modo da potersi portare direttamente sui punti fissati si è appurato che il secondo punto è quello che effettivamente ci torna utile.

Quindi creata una versione del programma con lo zoom, in modo da avere in evidenza la parte di interesse e sulla quale si sta agendo. Sullo sfondo del grafico è posta un'immagine in scala del campione in modo da vedere come si muove il fascio in modo virtuale tra le strip.

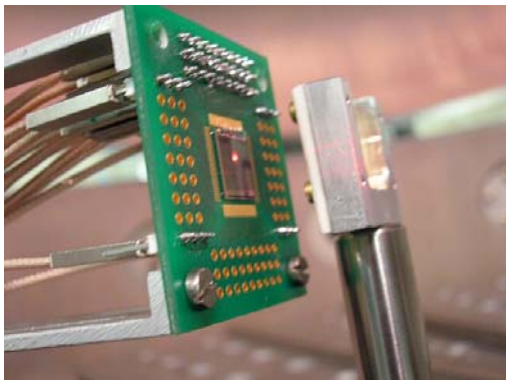
In un primo momento si presenta un problema, la basetta su cui è posto il campione non è perfettamente simmetrica (figura 4.5.2).

Fig 4.5.2



Con l'aiuto del laser si cerca di allineare il campione ad occhio irradiando lungo il bordo esterno della prima strip e in caso di errori evidenti apportando correzioni manualmente, agendo sulle viti che sostengono la basetta su cui è fissato il campione.

Fig 4.5.3



La telecamera infine, posta in modo da poter visualizzare il campione durante l'irraggiamento da laser, permette di essere più precisi ed effettuare un pennellamento dei bordi orizzontale e verticale (figura 4.5.4).

In figura 4.5.3 è ben visibile il laser sul campione.

Fig 4.5.4

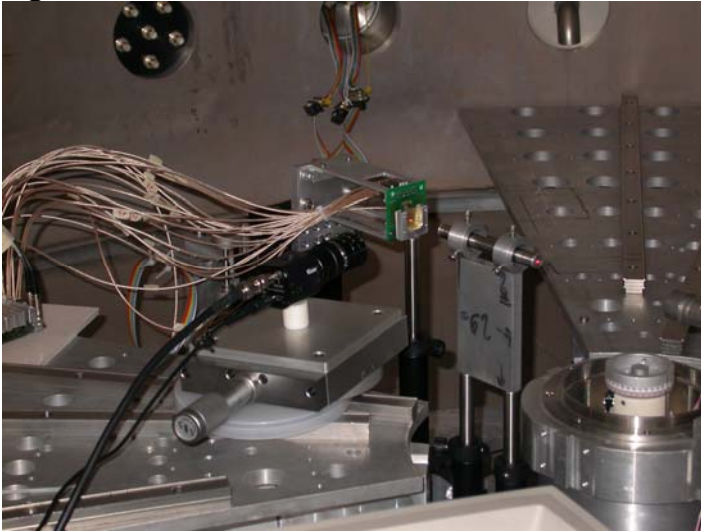


Fig 4.5.5



Una volta spostati collimatore finale e basetta in modo da non intercettare il fascio, si effettua un'ultima visualizzazione dello spot prodotto dal laser con la telecamera (figura 4.5.5) e lo SFOP.

Il laser viene rimosso e l'asse del fascio ripristinato, come si vede in figura 4.5.6, prima di effettuare il vuoto nella camera2000 .

Fig 4.5.6



4.6 COLLEGAMENTO DA REMOTO

Poco vicino alla camera2000 un PC collegato al driver dei motorini contenente il programma (figura 4.6) è collegato in remoto con il PC che verrà utilizzato per effettuare la movimentazione del campione durante l'irraggiamento.

Fig 4.6



4.7 IRRAGGIAMENTO

Prodotto il fascio di carbonio 12 con il tandem, tramite lo SFOP e la telecamera si può osservare la sua intensità. Ad una prima visualizzazione risulta troppo intenso per il nostro scopo.

Infatti il fascio, soprattutto nel nostro caso, non deve essere troppo intenso perché il segnale di risposta potrebbe alterarsi a causa di un secondo segnale che si sovrappone alle alte frequenze e alla lunga il rivelatore potrebbe danneggiarsi o anche bruciarsi.

Tramite convertitori, il segnale sulle strip viene conteggiato e un fascio non troppo intenso permette di avere un numero adeguato di particelle da conteggiare in modo da poter meglio discriminare il segnale.

Il fascio visto senza collimatore finale ha la forma ovale dovuta al fatto che il cannocchiale non è perfettamente in piano ma un po' inclinato.

4.8 PRIMO RIVELATORE

Il primo rivelatore in esame è stato il R328-3, di 100 mm² con 16 strip verticali di area 536 μm * 8255 μm in quattro gruppi con resistenze diverse, una strip orizzontale delle stesse dimensioni posta inferiormente, contatto di massa posto superiormente. [Sensors Product Development Group, 2006]

Fig 4.7

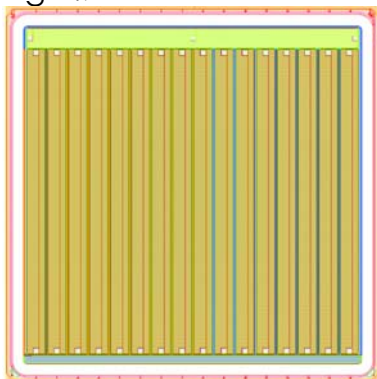
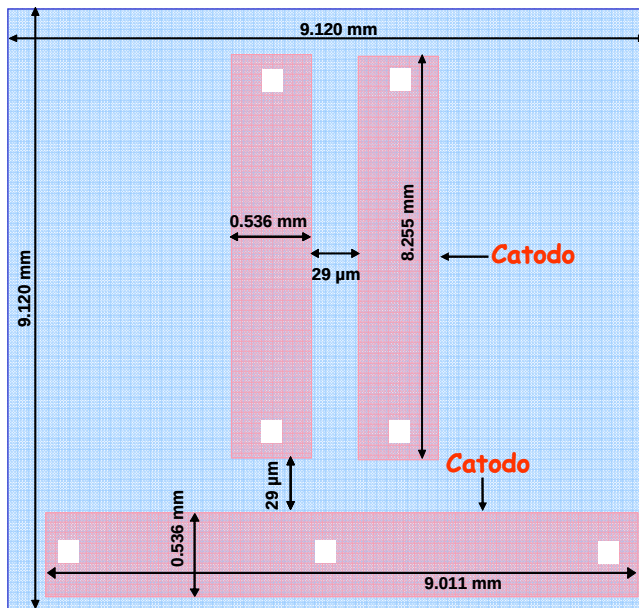


Fig 4.7.1



In un primo momento è stato necessario aggiustare il grafico nel programma perché da un conteggio preliminare non corrispondevano le strip reali con quelle virtuali; posizionati sulla prima strip e passando alla 16 si è verificata la corrispondenza per poi controllare inizio, fine e centro strip.

Per un allineamento del grafico più preciso, ci si è posizionati al centro virtuale di una strip, facendo piccoli spostamenti di 100 μm, prima verso destra e poi verso sinistra conteggiando ogni volta il segnale di uscita e facendo la media; dai dati ottenuti si è costruito il grafico della dispersione che ha fornito il centro effettivo dato dal punto in cui i conteggi sono più alti.

Fig 4.7.2

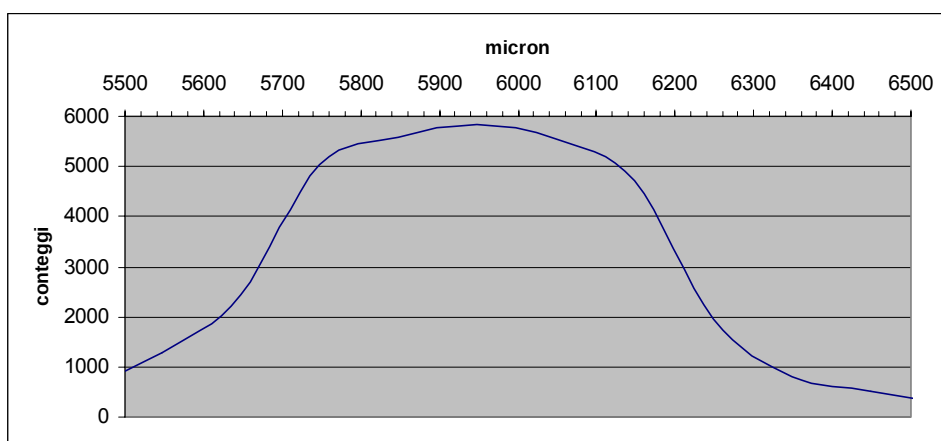
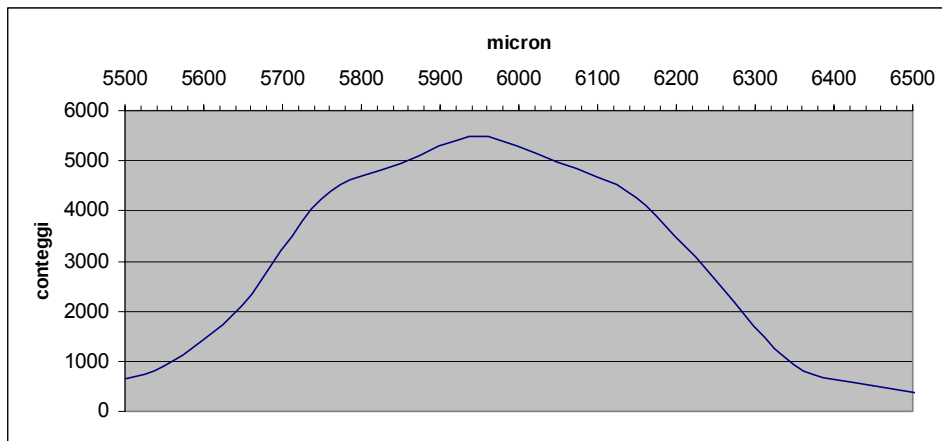


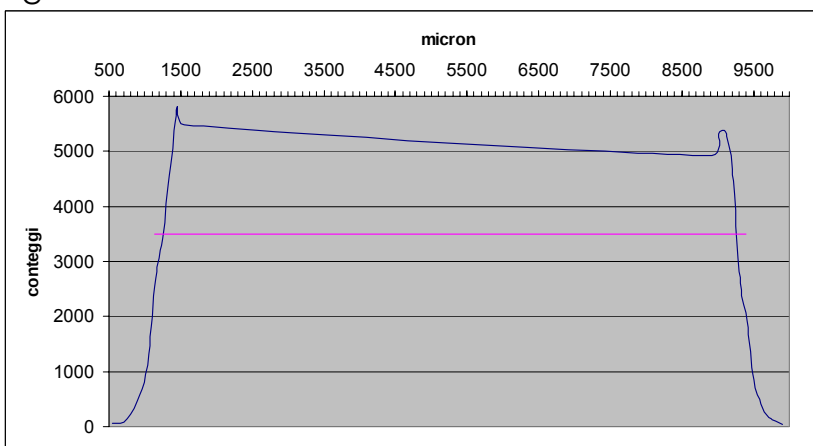
Fig 4.7.3



Si è ripetuta questa operazione in altri punti lungo la strip per vedere se il centro coincide.

Per verificare l'allineamento in lunghezza, ci si posiziona prima a circa 500 μm sopra al bordo superiore e scendendo a piccoli passi di 100 μm in giù fino ad arrivare a circa 500 μm sotto il bordo. All'interno della strip i conteggi devono essere all'incirca uguali, dai risultati è stato possibile fare un grafico dal quale si evince l'angolo che rappresenta l'ingresso nella zona attiva della strip; si è fatto lo stesso per il bordo inferiore, posizionandoci a circa 500 μm al di sotto del bordo e andando su a piccoli passi di 100 μm e conteggiando il segnale.

Fig 4.7.4



Mettendo insieme i due grafici ottenuti dai conteggi dei due bordi, si è ottenuto un unico grafico da cui imponendo una retta della lunghezza della strip (8255 μm) si è potuto definire i due limiti superiore e inferiore.

Svolte queste operazioni si è posizionato il grafico tenendo conto dei risultati (il centro risultava spostato di 112 μm rispetto al valore che prima si era considerato) e settato definitivamente l'origine del nostro grafico in modo da poter lavorare in modo virtuale ma avvicinandoci nel modo migliore possibile alla realtà.

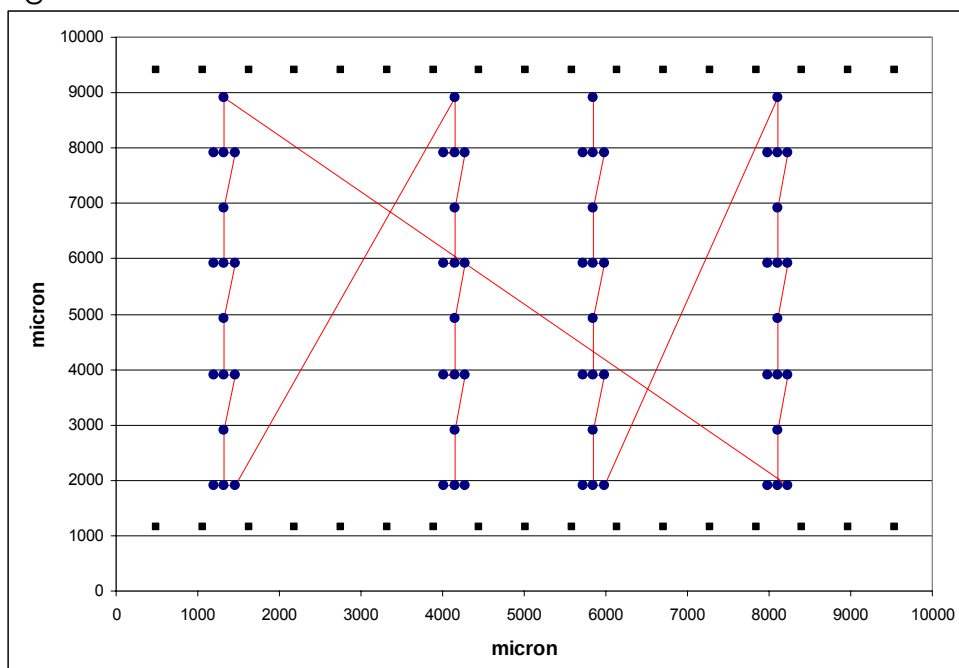
Fatta una tabella di marcia, si è iniziato il lavoro.

Si sono scelte le strip 2-7-10-14 e per ognuna 16 punti, 8 in verticale e in quattro di questi due ai lati.

Scelto l'ordine di irraggiamento si è fatta una tabella dei run, distribuita tra i gruppi di lavoro, in cui ogni punto è stato corrispondente ad un determinato codice, in modo da poter lavorare con un certo sincronismo e facendo affidamento su un comune riferimento; una volta posizionati su ciascun punto abbiamo potuto effettuare l'acquisizione.

In fig 4.7.5 un'immagine del percorso che si è fatto tra le strip, i cui margini sono rappresentati dai puntini in alto nel grafico.

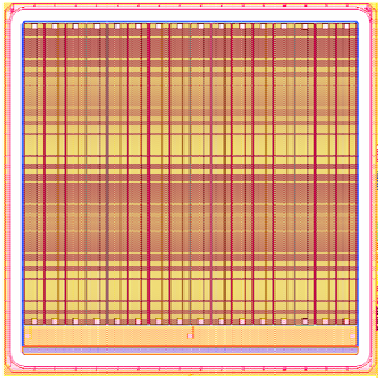
Fig 4.7.5



4.9 SECONDO RIVELATORE

Il secondo rivelatore preso in esame è il R328-1 con 16 strip orizzontali di area $536 \text{ um} \times 8255 \text{ um}$ di uguale resistenza, una orizzontale posta inferiormente di area $537 \text{ um} \times 9012 \text{ um}$; il contatto di massa posto inferiormente.

Fig 4.8



In questo caso non abbiamo potuto fare un allineamento preliminare con il laser perché avremmo dovuto riaprire la linea del fascio e posizionare il laser correttamente, quindi abbiamo ripetuto l'operazione di allineamento tramite conteggi; si è presentato un problema in più rispetto al primo campione, l'inclinazione poteva essere non trascurabile.

L'allineamento del rivelatore è stato controllato sotto fascio contando le particelle che colpivano la strip al variare della posizione in orizzontale in due punti (in fig 4.8.1 e 4.8.2) ed in verticale (in fig 4.8.3).

Fig 4.8.1

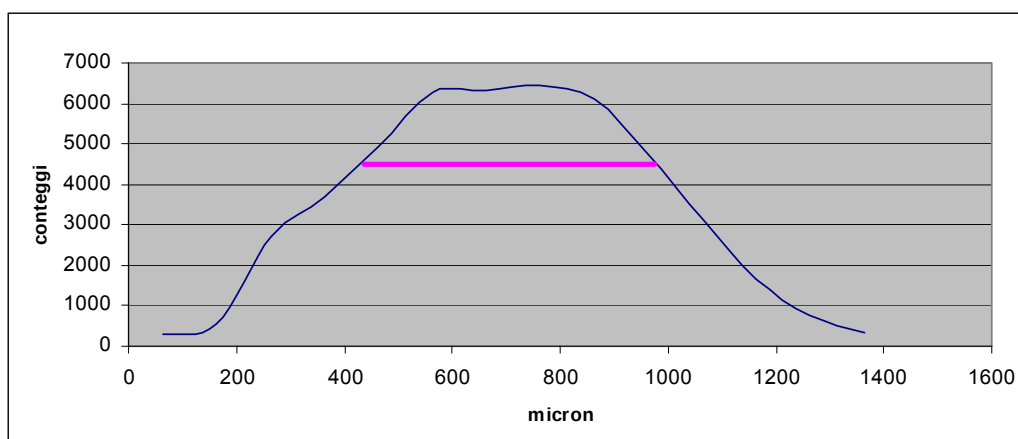


Fig 4.8.2

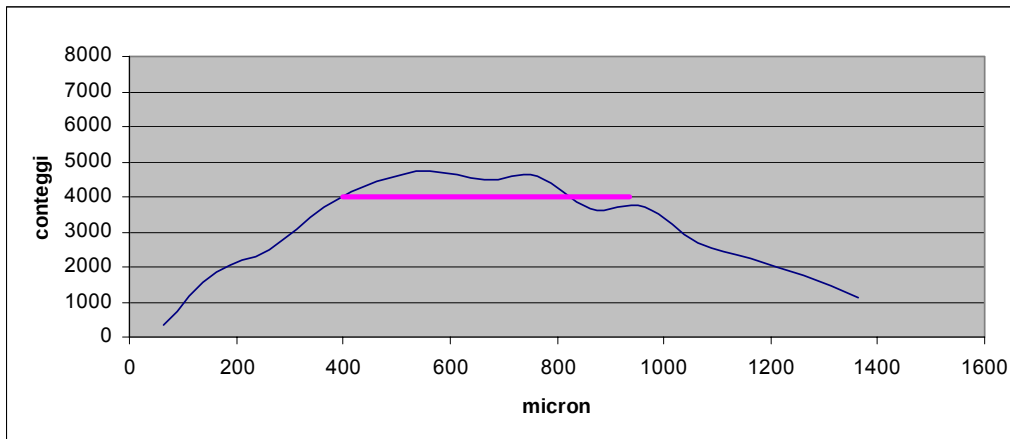
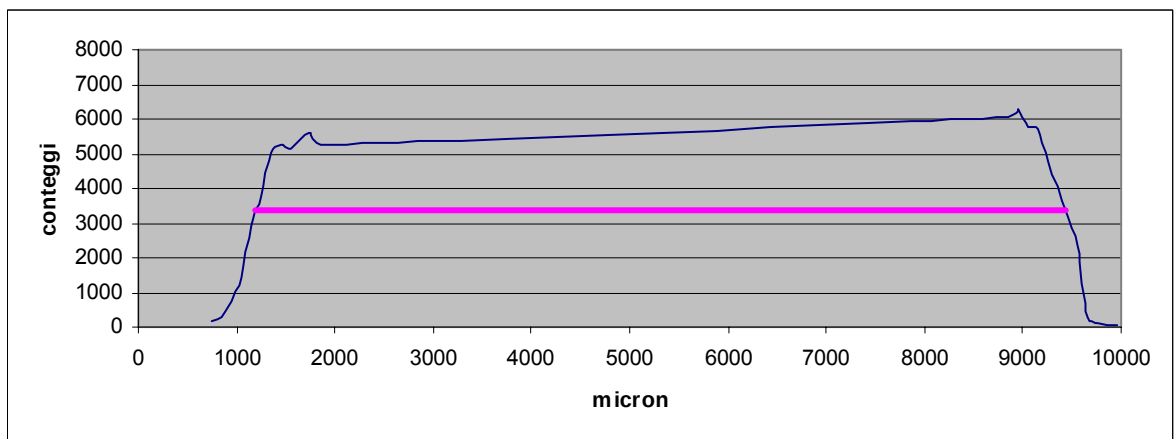


Fig 4.8.3



L'inclinazione del rivelatore, fornita dai precedenti calcoli è stata di $0,6^\circ$ e alla luce di questo si è fatto uso delle formule di rototraslazione :

$$X' = a + x \cos(\delta) - y \sin(\delta)$$

$$Y' = b + x \sin(\delta) + y \cos(\delta)$$

Dove δ è l'angolo d'inclinazione, a e b sono gli scostamenti lineari dal sistema di riferimento originale trovati sperimentalmente con i calcoli precedenti.

In fig 4.8.4 il run completo del secondo rivelatore, si sono irradiate in ordine le strip 2, 7, 10 e 15.

In fig 4.8.5 e 4.8.6 le strip 2 e 15 sono state irradiate in condizioni diverse rispetto alle condizioni del precedente run, senza collimatore finale e con il fascio ad un'intensità minore.

Fig 4.8.4

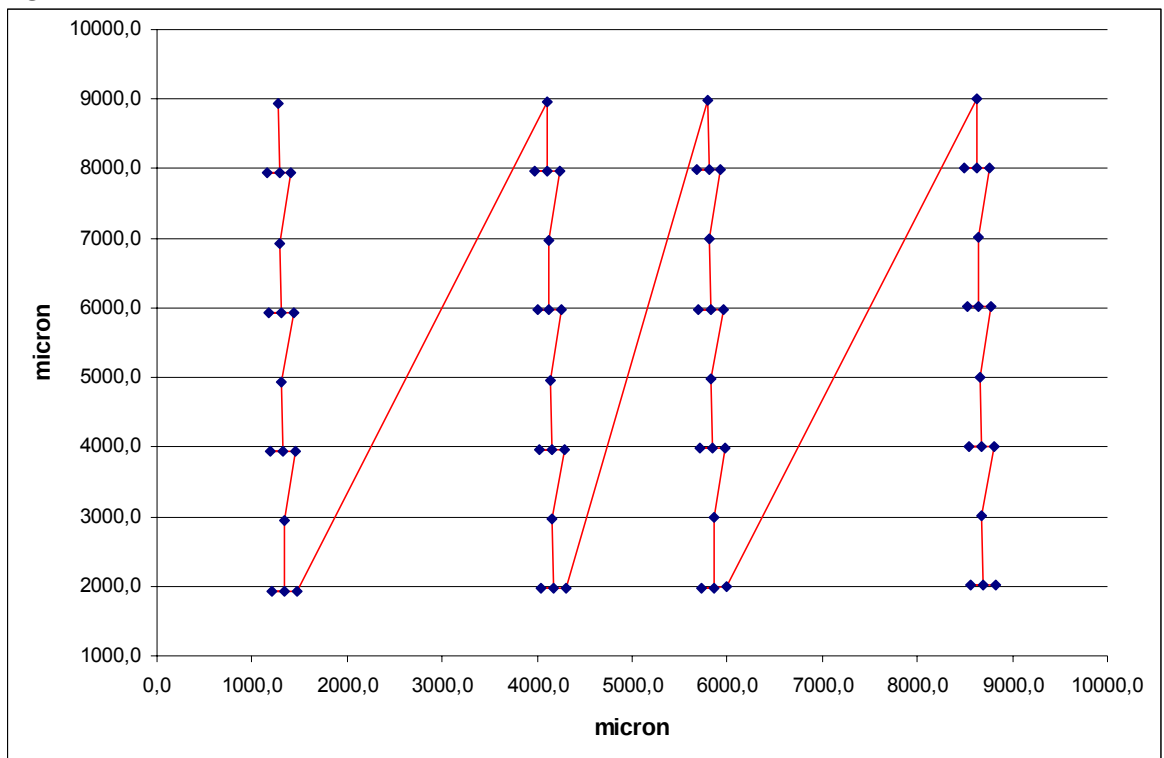


Fig 4.8.5

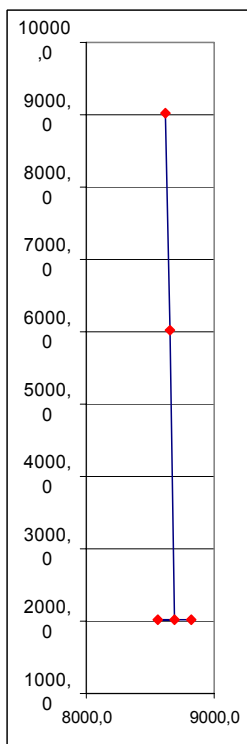
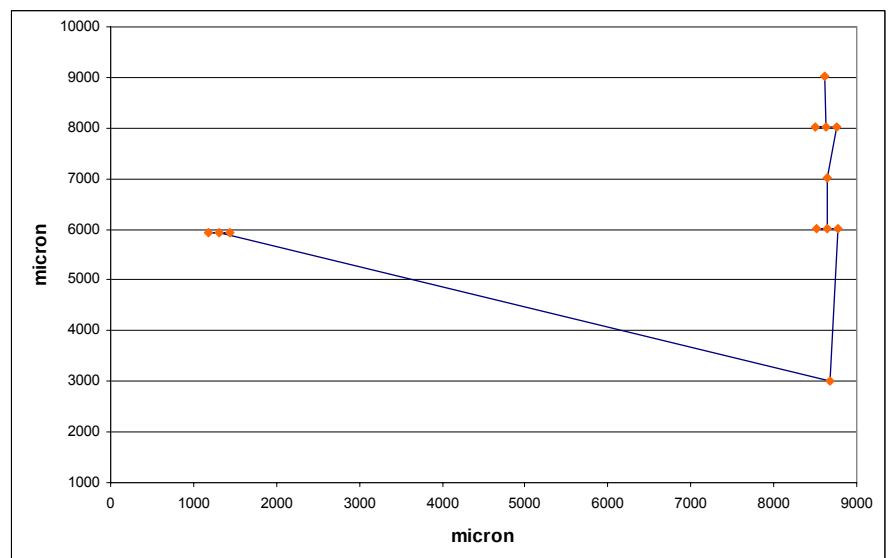


Fig 4.8.6



4.10 INDUZIONE

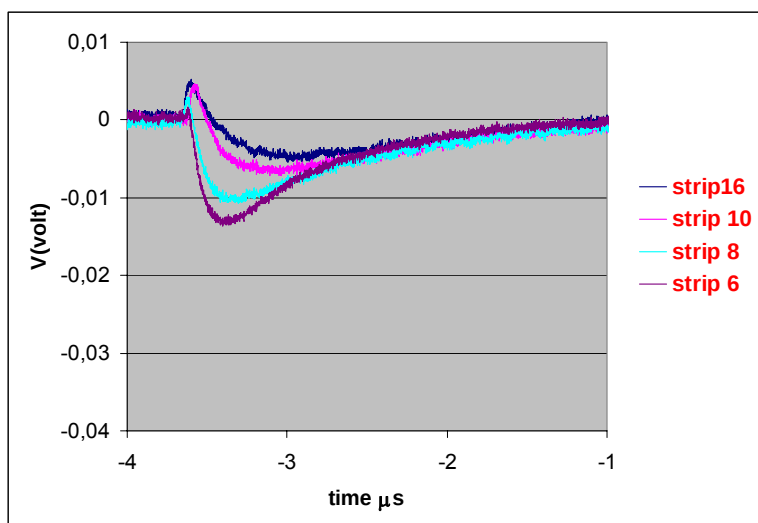
La carica creata nella regione E collegata tramite l'elettrodo ground, cambia il campo elettrico della zona ΔE inducendo un segnale più grande del segnale reale. L'induzione è dovuta alla resistenza dell'elettrodo ground, dell'ordine di qualche $K\Omega$. Viene quindi misurata l'ampiezza del segnale di induzione di una strip quando ne è irradiata un'altra.

Ci si aspetterebbe una dipendenza lineare dell'induzione rispetto alla posizione della particella urtata, ma i risultati sperimentali dimostrano che per questo rivelatore ciò non accade.

L'andamento dei picchi di induzione osservati mostra bene la notevole dipendenza dell'induzione dalla posizione per le posizioni più vicine al ground.

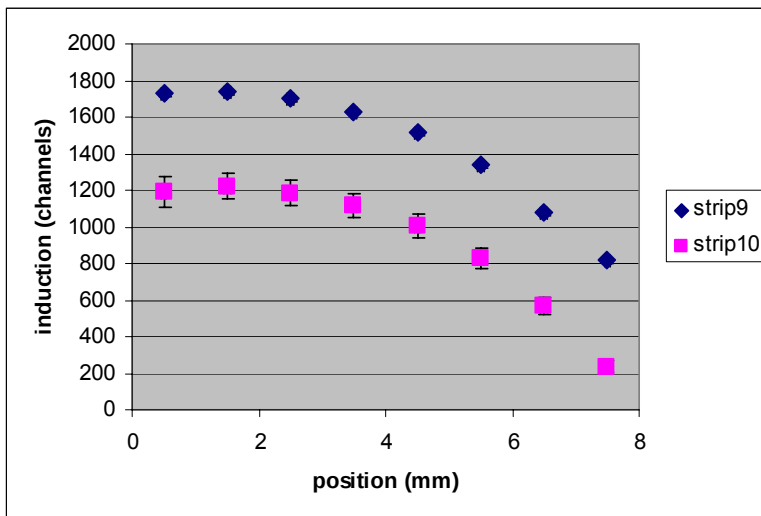
L'induzione viene vista da tutte le strip e si vede molto meglio nelle strip non colpite. In fig 4.9 un esempio dei segnali misurati nella strip 2 irradiando le strip dalla 16 alla 1 e si può vedere che l'induzione è più rapida nelle strip vicine.

Fig 4.9



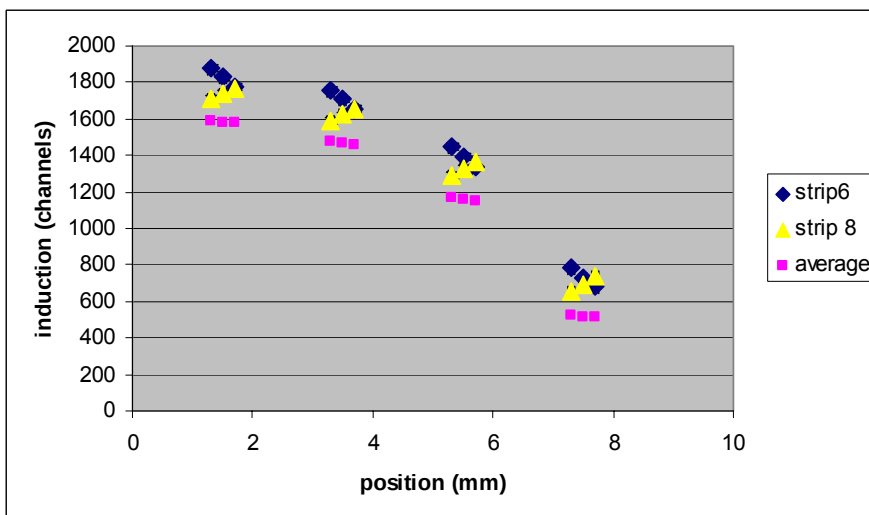
I punti mostrati nel grafico in fig 4.9.1 sono i punti centrali delle strip, si può ben notare che la dipendenza dell'induzione dalla posizione non è lineare.

Fig 4.9.1



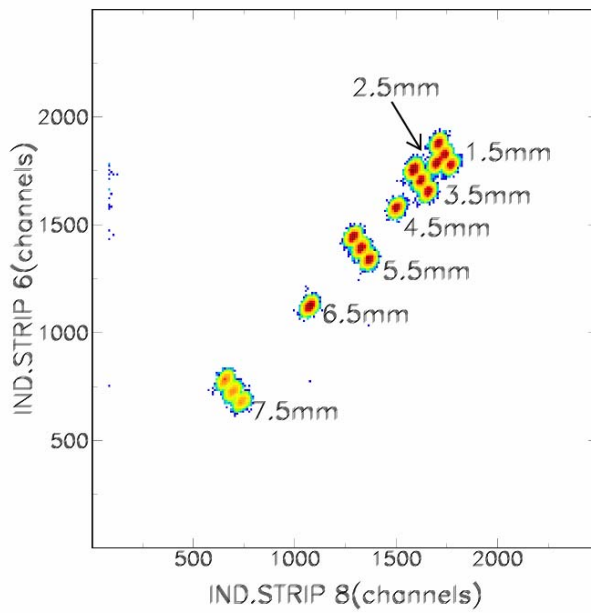
In figura 4.9.2 i 4 punti in è stata irradiata la strip 7 al centro strip e a -130 e + 130 micron. Sono riportate le induzioni misurate nelle strip adiacenti e la media (evento per evento) dell'induzione nelle due strip. Da notare come l'induzione sia anticorrelata nelle due strip se cresce nella 6 diminuisce nella 8.

Fig 4.9.2



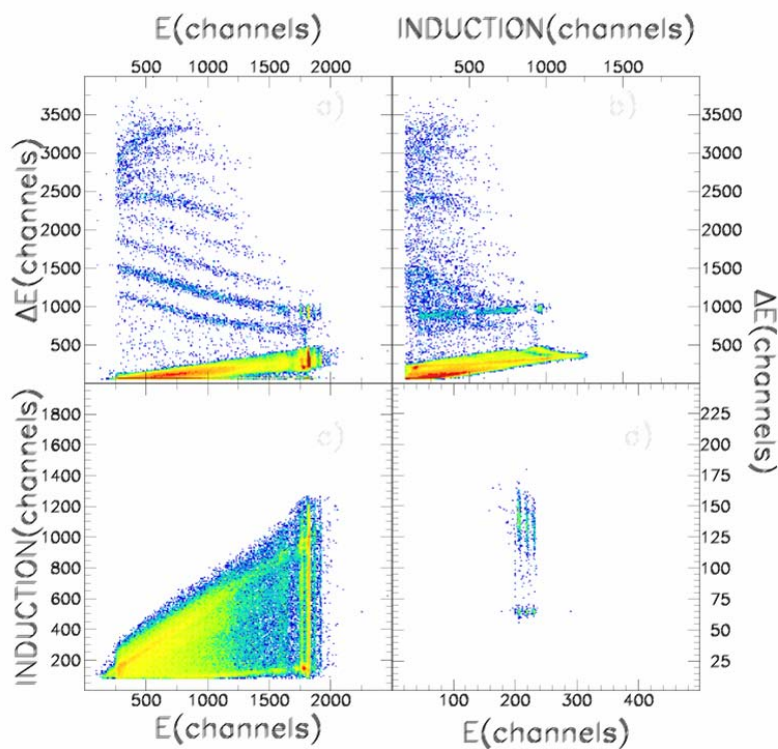
Combinando le due induzioni misurate nelle strip adiacenti è possibile ottenere l'immagine della strip irradiata, come fare una vera propria fotografia della strip con ottima risoluzione, come si evince dalla figura 4.9.3.

Fig 4.9.3



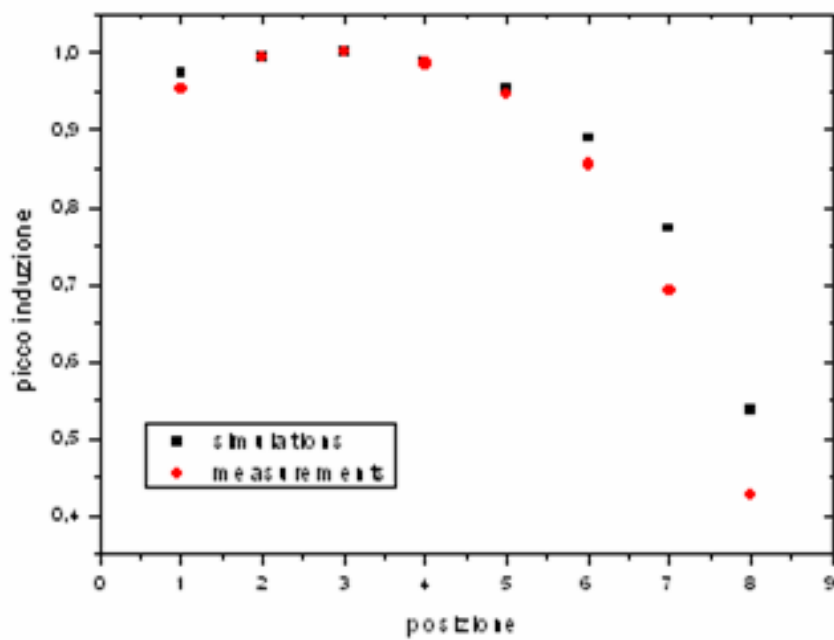
Nella figura 4.9.4 delle immagini DE-Induzione e Induzione-E in cui l'induzione E ha il tipico andamento a vela dato dalla proporzionalità dell'induzione dall'energia E.

Fig 4.9.4



Nel grafico in fig 4.9.5 i risultati ottenuti tramite simulazioni permettono di avere un buon metodo per studiare varianti del rivelatore atte a migliorarne la prestazioni.

Fig 4.9.5



CONCLUSIONI

Nella presente tesi si è affrontato il problema della movimentazione di un sistema di posizionamento e dello sviluppo del sw di interfaccia utente, della verifica e del funzionamento.

Dalla descrizione della problematica che ha visto la nascita del programma, cioè la movimentazione di un sistema rototraslatorio sul quale è stato posto un campione da caratterizzare con un fascio di particelle cariche prodotto presso i LNS-INFN, alla visione delle apparecchiature e tecnologie necessarie per il fine proposto, alla panoramica delle possibili applicazioni correnti e future di utilizzo di fasci accelerati, ci si è fatti un'idea dell'ambiente in cui si è svolta la formulazione di una possibile soluzione sw.

Quindi la presentazione dell'ambiente di sviluppo scelto, il LabVIEW, di cui si sono discusse le principali strutture e funzionalità per dare al lettore che non conosce il linguaggio la possibilità di comprendere il programma sviluppato, che è stato poi analizzato nella sua interfaccia e nel suo codice prevalentemente grafico nel capitolo 3.

Infine, ma non meno importante, una particolare attenzione al capitolo 4 dove si parla del turno di misura che ha visto una soddisfacente applicazione del programma, che ha riscosso un certo successo sia perché ha funzionato senza problemi e ha permesso una certa facilità nel posizionamento del campione nei punti desiderati, sia perché è stato piacevole avere un riscontro grafico 'virtuale' che ha permesso di avere una maggiore consapevolezza del lavoro che si stava svolgendo.

Inoltre i test effettuati sui campioni hanno fornito i risultati sperati e avere contribuito alla realizzazione di un qualcosa di concreto e utile per lo studio di rivelatori particolarmente sensibili e in generale aver collaborato con professionisti del settore, rende il lavoro svolto più appagante e stimolante verso nuove future possibilità di collaborazione.

Il programma allo stato attuale presenta delle imperfezioni, come il problema del target che deve essere sempre settato e degli aggiornamenti delle schermate che causano un asincronismo dei comandi; in parole povere, spesso capita che quando si inserisce il valore di target nella targhetta corrispondente, questo non viene preso in ingresso e lo si deve riscrivere fin quando ciò non avviene.

E', inoltre, necessario l'inserimento di un certo numero di controlli che rendano ancora più immediato l'uso del sistema e la cui importanza è risultata evidente durante l'uso del sistema in particolare nei turni di prova.

E' possibile migliorare la fruibilità dell'interfaccia prevedendo dei 'raster', cioè dei percorsi prestabiliti, come il movimento zig-zag o obliquo, da fare compiere al campione in esame tramite la movimentazione del sistema rototraslatorio e questo è uno degli obiettivi proposti per l'ottimizzazione del programma.

Una futura applicazione del progetto sviluppato potrà essere quella del suo utilizzo in altri turni di misura e in altri casi in cui sarà necessario movimentare un sistema con precisione micrometrica.

APPENDICE

A.1 L'ACCELERATORE DI PARTICELLE: IL TANDEM

Un acceleratore di particelle è una macchina che genera fasci di particelle cariche (elettroni, protoni o ioni pesanti) ad alta energia cinetica da impiegare nel campo della ricerca scientifica, fisica nucleare, terapie mediche, applicazioni industriali e altre ancora.

Al giorno d'oggi, ci sono due principali tipi di acceleratori di particelle ad alta energia : acceleratori lineari e acceleratori circolari. Questi a loro volta possono suddividersi in un'ampia gamma di sottospecie, la cui descrizione esula dall'ambito di questa tesi. In particolare presso i LNS sono disponibili due acceleratori: uno circolare, detto Ciclotrone Superconduttore, e uno lineare elettrostatico, detto Tandem. Quest'ultimo, in quanto è quello che ho avuto modo di utilizzare nel corso di questo lavoro, è brevemente descritto qui di seguito.

Il tandem può accelerare ioni negativi all'interno di un tubo in cui si è fatto il vuoto per mezzo di una differenza di potenziale dell'ordine di diversi MV ($+4\text{MV} < \Delta V < +15\text{MV}$), generata per via elettrostatica.

Per prima cosa un elettrone viene attaccato ad un atomo neutro, facendolo diventare uno ione negativo. Il fascio di ioni negativi è iniettato tramite una differenza di potenziale di qualche centinaio di kV all'ingresso del Tandem. L'elevata differenza di potenziale tra il punto di ingresso (massa) e il terminale (al centro della macchina) fa sì che gli ioni negativi vengano fortemente accelerati verso il terminale stesso, acquistando energia cinetica pari a $e \cdot \Delta V$ (nel caso massimo 15MV). Al terminale è posizionato una sottile lamina di carbonio, detto stripper, che attraversato dagli ioni accelerati strappa a questi diversi elettroni trasformandoli in ioni positivi. Essi sono dunque accelerati una seconda volta (questo è il motivo per cui si chiama Tandem) poichè oltrepassato il terminale gli ioni, adesso positivi, vedono una differenza di potenziale ancora favorevole (da

terminale verso l'uscita c'è tensione $-\Delta V$). Il Tandem può produrre fasci molto stabili ad energia variabile con continuità. E' limitato ad energie abbastanza basse a causa del suo principio di funzionamento elettrostatico, che impedisce di creare differenze di potenziale al di sopra dei 15MV.

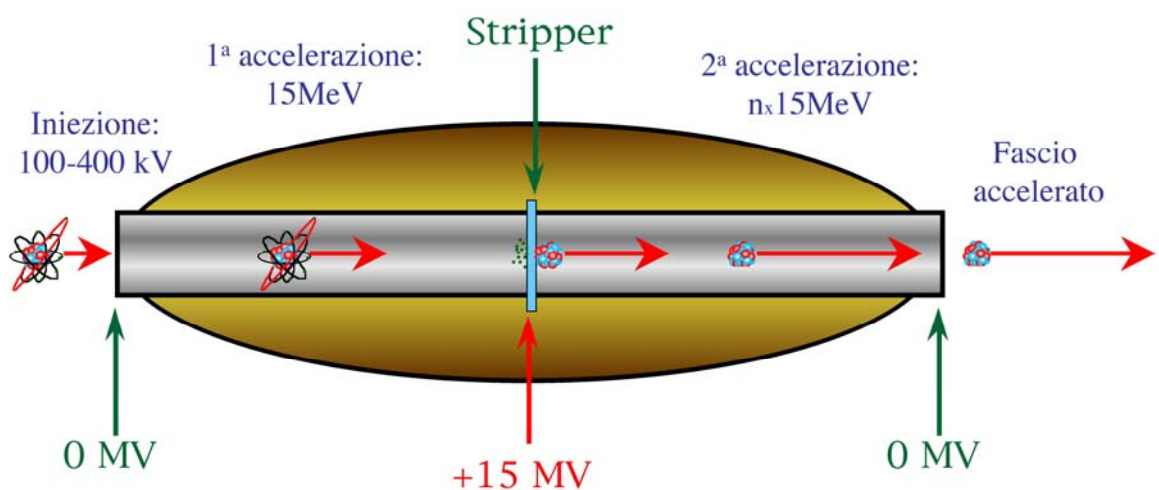
Fig 1 Il tandem



Fig 2 Interno del tandem



Fig 3 Il principio di funzionamento del tandem



A.2 CAMERE A VUOTO, LA CAMERA 2000

Con il termine “vuoto” si indica lo stato di un gas quando la densità delle particelle è più bassa della pressione atmosferica sulla superficie terrestre.

Di una camera in cui viene effettuato il vuoto è importante determinare la grandezza e la forma per poi poter specificare il modo corretto la pressione e la composizione del gas residuo.

Rimuovere il gas atmosferico da un contenitore permette di creare un ambiente differente in molti aspetti da quello abituale: meccanico per la pressione esercitata nelle pareti, chimico e fisico per la conducibilità termica ed elettrica.

La camera 2000, è una camera a vuoto con il diametro di 2 m con bracci meccanici e supporti interni motorizzati, posizionata in modo da poter essere intercettata dal fascio.

Nel caso specifico essa si trova a 60° rispetto alla linea del fascio.

Fig 4 La camera 2000

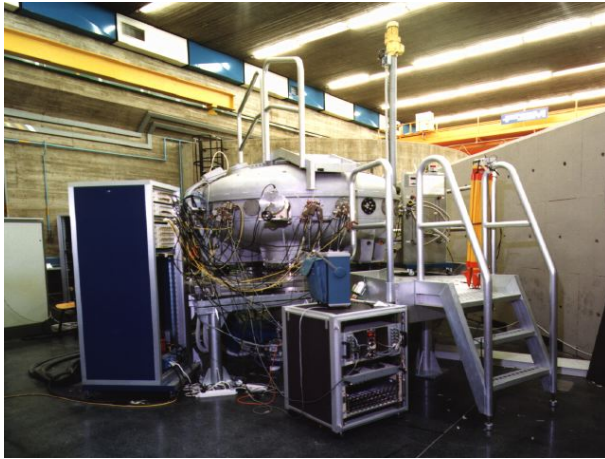
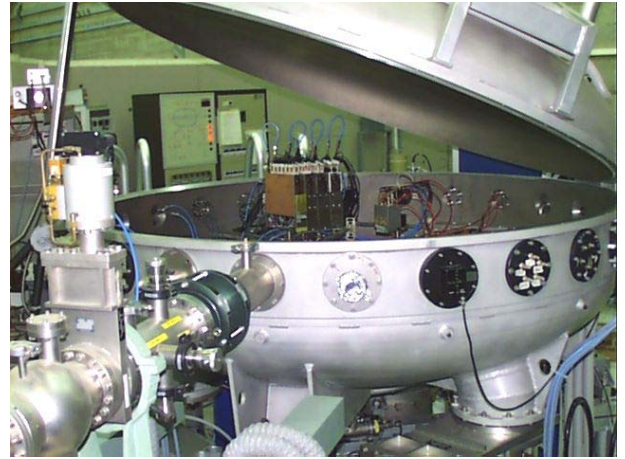


Fig 5 Interno della camera 2000



RINGRAZIAMENTI

Ringraziamenti particolari a tutti coloro che mi sono stati vicini in questo periodo, che mi hanno aiutata e sostenuta:

in particolare al dott. Paolo Finocchiaro (LNS-INFN) e al prof. Michele Malgeri (DIIT), che mi hanno seguita costantemente nella preparazione della presente tesi e sono stati sempre pronti a darmi spiegazioni e suggerimenti;

ai dott. Luigi Cosentino, dott. Alfio Pappalardo e dott. Angelo Campisi (LNS-INFN) che sono stati sempre disponibili a dare il loro aiuto e mettere a disposizione le loro conoscenze;

a coloro con cui ho condiviso con molto piacere l'esperienza del turno di misura presso l'INFN dott.ssa Francesca Amorini, prof. Giuseppe Cardella, prof.ssa Francesca Rizzo, dott. PierPaolo Figuera;

ai miei genitori e ai miei fratelli che mi hanno dato supporto morale e mi hanno sempre incoraggiata e stimolata, soprattutto a mio fratello Sergio con cui ho condiviso l'esperienza del tirocinio;

ai miei amici e in particolar modo al mio ragazzo Antonio, che mi hanno sostenuta e a cui voglio bene.

BIBLIOGRAFIA

Francesca Amorini, 2005

Characterization of new position sensitive Monolithic Silicon Telescopes

Angelo Campisi, 2004/2005

Caratterizzazione di matrici di Fotodiodi a valanga per la rivelazione di singoli fotoni

G. A. Pablo Cirrone, V. Mongelli, 2003

www.lns.infn.it/CATANA

Articoli:

1 L.Cosentino, P.Finocchiaro, A.Pappalardo, P.Vynck, B.Volckaerts, A.Hermanne, M.Vervaeke, H.Thienpont

- Ion micro-beam diagnostics with photodetectors
- Diagnostic Tools For Low Intensity Ion Micro-Beams

2 L.Cosentino, P.Finocchiaro, A.Pappalardo, P.Vynck, B.Volckaerts, A.Hermanne, M.Vervaeke, H.Thienpont, H.Ottevaere, P.Tuteleers

- Beam monitoring enhances Deep Proton Lithography: towards high-quality micro-optical components

3 L.Cosentino, P.Finocchiaro

- Ion beam imaging at very low energy and intensity (2003 ELSEVIER)

ISO Guida all'espressione dell'incertezza nella misurazione

National Instrument, 2000

LabVIEW 6 Basics Course Manual I / II

National Instrument, 2003

LabView Manuale Italiano

Newport Corporation, 1998

DC Interface Box Operator's Manual/ Section 1

Newport Corporation, 1999

Newport Corporation, 2000

Motion Control Tutorial/ Section 6

www.newport.com

Newport Corporation, 2006/2007

The Newport Resource 2006/2007- Solutions to Make, Manage & Measure Light

Alfio Pappalardo 2001/2002

Nuove tecniche di microdiagnostica con scintillatori per fasci ionici di bassa intensità

Sensors Product Development Group, 2006

DE-E position sensitive detector: R328

www.st.com

Pedro Vynck, 2004

Development of an Irradiation Set-up for Deep Lithography with Protons

UNI 4546 Misure e misurazioni