



UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA MECCATRONICA

---

*TESI DI LAUREA MAGISTRALE*

PROGETTO E REALIZZAZIONE DI  
SISTEMI ELETTRONICI PER IL CONTROLLO  
DI LASER SINTONIZZABILI  
UTILIZZATI NEL PROGETTO SPES

*Relatore:* Ch.mo Prof. ROBERTO OBOE

*Correlatore:* Dott. ALBERTO ANDRIGHETTO

*Correlatore:* Ing. DANIELE SCARPA

*Laureando:* GIOVANNI CERONI

Matricola 1019807-IMC

ANNO ACCADEMICO 2014-2015



Alla mia famiglia



# Sommario

---

Nel laboratorio laser del progetto SPES sono in corso esperimenti di fotoionizzazione risonante, con lo scopo di ionizzare diversi tipi di atomi utilizzando combinazioni di fasci laser a diverse lunghezze d'onda; questi fasci vengono forniti da laser sintonizzabili. Volendo avere la possibilità di variare in modo continuo la lunghezza d'onda, è stato chiesto di comandare in modo accurato questi dispositivi che al momento possiedono solo un comando di tipo manuale. Questa tesi si occupa quindi della progettazione e realizzazione di un sistema di controllo e della sua interfaccia per la gestione di tre diversi dispositivi presenti nel laboratorio. Nella tesi verranno illustrate le caratteristiche dei singoli dispositivi; saranno descritti i criteri che hanno guidato la progettazione dell'hardware realizzato e ne sarà analizzato il software dedicato. L'ultima parte di questo lavoro di tesi evidenzia brevemente i risultati ottenuti con i test sperimentali per validare la bontà di quanto realizzato.



# Indice

---

<b>Sommario</b>	<b>v</b>
<b>Indice</b>	<b>vii</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Il progetto SPES ai Laboratori Nazionali di Legnaro[1][2][3]</b>	<b>3</b>
2.1 Introduzione . . . . .	3
2.2 Il decadimento radioattivo . . . . .	3
2.3 RIBs facilities . . . . .	5
2.3.1 ISOL facilities . . . . .	5
2.4 Il progetto SPES ai Laboratori Nazionali di Legnaro . . . . .	6
2.5 Configurazione della facility . . . . .	7
2.5.1 L'acceleratore primario . . . . .	7
2.5.2 Il Front-End . . . . .	9
2.5.3 Il target di produzione e il sistema di estrazione e di ionizzazione . . . . .	9
2.5.4 I separatori magnetici e la post accelerazione . . . . .	15
2.6 Applicazioni dei fasci di ioni esotici . . . . .	16
2.6.1 Applicazioni in fisica nucleare . . . . .	16
2.6.2 Applicazioni in fisica dello stato solido . . . . .	18
2.6.3 Applicazioni in medicina nucleare . . . . .	20
2.6.4 Applicazioni in astrofisica . . . . .	22
2.7 Conclusioni . . . . .	22
<b>3 Strumentazione per la fotoionizzazione Laser[4]</b>	<b>23</b>
3.1 Introduzione . . . . .	23
3.2 La fotoionizzazione . . . . .	23
3.3 L'emissione stimolata e la radiazione laser . . . . .	24
3.4 Principio di funzionamento dei dispositivi LASER . . . . .	24
3.4.1 Modalità Q-switch . . . . .	26
3.5 Laser a colorante (dye) . . . . .	27
3.5.1 Reticolo[5] . . . . .	28
3.5.2 Filtro Etalon . . . . .	29
3.5.3 Raddoppio in frequenza . . . . .	30
3.6 Motori passo-passo[6] . . . . .	31
3.6.1 Principio di funzionamento dei motori passo-passo . . . . .	31
3.7 Monocromatore . . . . .	32
3.8 Conclusioni . . . . .	33
<b>4 Progetto Scheda Elettronica</b>	<b>35</b>
4.1 Introduzione . . . . .	35
4.2 Richieste progettuali hardware . . . . .	35
4.2.1 Dispositivi hardware presenti . . . . .	35
4.2.2 Componenti hardware necessari . . . . .	37
4.3 Software utilizzato per la progettazione della scheda elettronica . . . . .	38
4.3.1 Orcad Capture . . . . .	38
4.3.2 Orcad Layout . . . . .	38
4.4 Struttura della scheda elettronica . . . . .	38

4.4.1	Microcontrollore PIC ed interfacciamento	39
4.4.2	Driver per controllo motori	41
4.4.3	Finecorsa	45
4.4.4	Pulsanti	48
4.4.5	Connettori dei dispositivi	50
4.5	Disegno della scheda elettronica	51
4.6	Realizzazione del prototipo	52
4.6.1	Stampaggio del circuito	52
4.6.2	Assemblaggio dei componenti	55
4.6.3	Alloggiamento per la scheda elettronica	57
4.7	Conclusioni	57
<b>5</b>	<b>Programma di controllo</b>	<b>59</b>
5.1	Introduzione	59
5.2	Analisi specifiche software richieste	59
5.2.1	Funzionalità richieste dal Laser FL2002	59
5.2.2	Funzionalità richieste dal Monocromatore JOBIN YVON	62
5.2.3	Funzionalità richieste dal Laser TDL	63
5.3	Software utilizzato per la programmazione	64
5.4	Nozioni base sulla programmazione di PIC	64
5.4.1	Porte I/O	65
5.4.2	Interrupt	65
5.4.3	Oscillatore	68
5.4.4	Modulo seriale (USART)	69
5.4.5	Convertitore A/D	69
5.5	Stesura del codice	69
5.5.1	Struttura generale	69
5.5.2	Operazioni preliminari	70
5.5.3	Funzioni di libreria	71
5.5.4	Routine di Interrupt (ISR)	75
5.5.5	Funzioni personalizzate	76
5.6	Programma principale	82
5.6.1	Inizializzazione del programma	82
5.6.2	Gestione degli avvisi interrupt	83
5.6.3	Conversione ADC e scelta Laser	83
5.6.4	Codice dedicato alla gestione del Laser FL2002	84
5.6.5	Codice dedicato alla gestione del Monocromatore	87
5.6.6	Codice dedicato alla gestione di un ulteriore dispositivo (Spare)	87
5.6.7	Codice dedicato alla gestione del Laser TDL50	88
5.7	Conclusioni	88
<b>6</b>	<b>Prove sperimentali</b>	<b>89</b>
6.1	Introduzione	89
6.2	Setup Sperimentale	89
6.2.1	Spettroscopia	89
6.3	Esperimento di fotoionizzazione laser	92
6.3.1	Ionizzazione dello stagno nella lampada a catodo cavo	92
6.4	Conclusioni	94
	<b>Conclusioni</b>	<b>95</b>
	<b>Ringraziamenti</b>	<b>97</b>



---

<b>Bibliografia</b>	<b>99</b>
<b>A Estratti dei manuali</b>	<b>101</b>
A.1 Estratti del manuale FL2002 . . . . .	101
A.2 Estratti del manuale TDL50 . . . . .	105
<b>B Schemi elettrici della scheda elettronica realizzata</b>	<b>107</b>
<b>C Codice</b>	<b>115</b>
C.1 Main program . . . . .	115
C.2 Funzioni . . . . .	127



## Introduzione

---

Il crescente interesse da parte della comunità scientifica verso lo studio di nuclei “esotici” non presenti in natura, sta favorendo la costruzione di appositi impianti (facilities) che siano in grado di produrli.

Presso i Laboratori Nazionali di Legnaro (LNL) si sta realizzando, con il progetto SPES, una facility di nuova generazione per la produzione di fasci radioattivi, con lo scopo di esplorare la “terra incognita” di nuovi nuclei instabili e di fornire un valido strumento per applicazioni di tipo medico ed industriale. La produzione di fasci di ioni radioattivi richiede l’impiego di tecnologie estremamente complesse ed innovative; il progetto SPES prevede infatti il bombardamento di un bersaglio (target) in carburo di uranio con un fascio di protoni, in modo da generare, tramite una reazione nucleare, specie radioattive all’interno del target. L’estrazione degli atomi prodotti viene effettuata ionizzando queste particelle ed estraendole con una differenza di potenziale; a seconda del tipo di elemento di cui si vuole produrre il fascio, vengono impiegate sorgenti di ionizzazione differenti: superficiali, al plasma, laser.

Prima che venga costruita ed avviata la facility vera e propria, all’interno dei LNL sono in corso studi tecnici e sperimentazioni per i vari componenti e settori che dovranno funzionare ed interagire per realizzare lo scopo finale. Il laboratorio laser del progetto SPES si occupa quindi di analizzare e verificare le procedure per riuscire ad ionizzare i vari elementi che si vorranno produrre. Uno strumento indispensabile a tal fine é un laser sintonizzabile che possa cioè variare, a seconda della richiesta, la lunghezza d’onda del fascio di luce laser che produce.

Questo tipo di dispositivi sono strumenti molto particolari e costosi, grazie però alla cooperazione con diversi enti ed università, il laboratorio é riuscito a dotarsi di due laser sintonizzabili: il TDL50 della Quantel e l’FL2002 della Lambda Physik. A differenza del piú recente TDL50, il laser FL2002 é un vecchio laser costruito negli anni ’80 e del quale non é piú disponibile il sistema di controllo, essenziale per il suo funzionamento.

Il presente lavoro di tesi s’inserisce in questo contesto ed ha, come scopo finale, la realizzazione di un sistema di controllo per questo Laser; sfruttando questa necessità si desidera inoltre che lo stesso strumento possa controllare anche altri dispositivi presenti nel laboratorio (Laser TDL50 e Monocromatore JOBIN YVON).

Il lavoro da svolgere é quindi la progettazione e realizzazione di una scheda elettronica contenente un microcontrollore che possa gestire i diversi dispositivi.

La tesi é suddivisa in cinque capitoli di cui si passa ora a riassumere brevemente i contenuti:

- **Capitolo 1.** In questo capitolo viene descritta la metodologia ISOL per la produzione di fasci di ioni radioattivi. In seguito vengono illustrate le principali caratteristiche del progetto SPES, in fase di sviluppo presso i Laboratori Nazionali di Legnaro. Infine, vengono elencate le principali applicazioni dei fasci di ioni radioattivi nei campi della fisica nucleare, dell’astrofisica, della scienza dei materiali e della medicina.
- **Capitolo 2.** In questo capitolo viene introdotto il concetto di radiazione laser, vengono esposti ed analizzati i principi fisici che stanno alla base di questo fenomeno e le tecniche costruttive utilizzate per ottenere dispositivi sempre piú versatili e

performanti. La descrizione si focalizza infine sui dispositivi Laser sintonizzabili e sulle loro caratteristiche peculiari.

- **Capitolo 3.** In questo capitolo viene descritto il progetto della scheda elettronica. Dopo aver analizzato i dispositivi dal punto di vista hardware, per individuare quali siano i segnali di uscita e quali quelli necessari al fine di ottenere un corretto funzionamento, si passa alla descrizione delle soluzioni scelte per realizzare le diverse funzionalità che la scheda deve garantire. Per finire viene descritta la procedura con la quale si è realizzato il prototipo della scheda e il suo assemblaggio.
- **Capitolo 4.** In questo capitolo, strettamente legato al precedente, vengono per prima cosa esaminate le funzionalità e le movimentazioni che si dovranno implementare; si passa poi ad un'introduzione sulla programmazione dei microcontrollori PIC valutando punto per punto le varie funzionalità che offrono. Viene infine descritto il programma che è stato realizzato per essere inserito nel microcontrollore e che gestisce tutta la scheda ed i sistemi ad essa collegati.
- **Capitolo 5.** In questo capitolo vengono descritti, in modo generale, gli esperimenti che è possibile effettuare nel laboratorio laser SPES, grazie all'ausilio dei Laser sintonizzabili. Vengono quindi riportati i risultati di alcuni test effettuati con il laser FL2002 pilotato dalla scheda elettronica che è stata progettata, realizzata e programmata per questo progetto di tesi.

## Il progetto SPES ai Laboratori Nazionali di Legnaro[1][2][3]

---

### 2.1 Introduzione

In questo primo capitolo viene illustrato il progetto SPES in fase di sviluppo presso i Laboratori Nazionali di Legnaro (LNL). Dopo un'introduzione sul decadimento radioattivo degli isotopi e la descrizione della metodologia ISOL per la produzione di fasci di ioni radioattivi, si passa ad illustrare nel dettaglio il progetto SPES, soffermandosi sui dispositivi principali ed i componenti che vengono utilizzati.

Per concludere vengono illustrate brevemente le molteplici applicazioni dei fasci di ioni radioattivi in vari campi: dalla fisica nucleare, alla scienza dei materiali e alla medicina.

### 2.2 Il decadimento radioattivo

Le proprietà fisiche e chimiche della materia sono legate alle caratteristiche degli atomi che la compongono, alla loro disposizione nello spazio e alle loro interazioni. Per questo motivo, per poter comprendere a fondo la materia, è necessario studiarla a partire dal livello atomico.

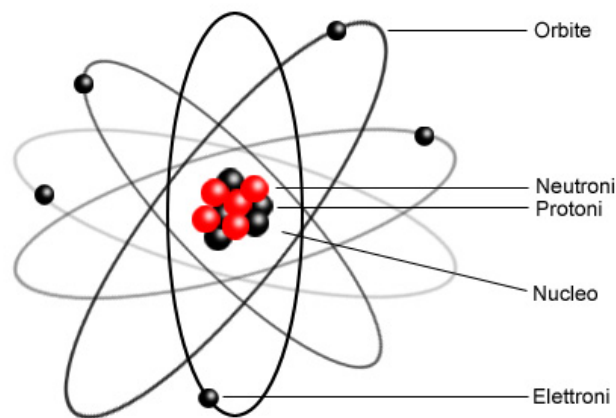


Fig. 2.1: Struttura dell'atomo.

L'atomo è costituito da particelle che possono essere cariche positivamente (protoni), neutre (neutroni) o con carica negativa (elettroni). Le prime due possiedono una massa di circa  $1,67 \cdot 10^{-27}$  kg e sono situate in un nucleo centrale in cui si concentra gran parte della massa atomica (più del 99.9% in peso), per questo motivo esso determina le principali caratteristiche dell'elemento d'appartenenza. Il numero ( $Z$ ) di protoni presenti nel nucleo determina l'elemento di appartenenza, il numero ( $N$ ) di neutroni è invece variabile. Atomi con ugual numero di protoni e diverso numero di neutroni sono detti isotopi dello stesso elemento e sono identificati mediante il numero di massa ( $A=Z+N$ ).

I protoni presenti nel nucleo tendono a respingersi tra loro a causa della forza elettrostatica, l'esistenza del nucleo e la sua stabilità è però resa possibile da una forza d'attrazione, denominata "nucleare forte", che agisce solo nelle distanze tipiche del nucleo e mantiene vicini protoni e neutroni. La forza attrattiva tra neutroni e protoni è leggermente superiore

a quella tra nucleoni dello stesso tipo, ed i nuclei con ugual numero di protoni e neutroni sono in genere stabili (non radioattivi); all'aumentare delle distanze però (in atomi con numero di massa  $A > 40$ ) la forza di Coulomb prende il sopravvento e sono necessari più neutroni per garantire la stabilità. La presenza della forza coulombiana determina comunque un limite superiore nel numero di particelle e quindi di atomi superpesanti: aumentando troppo la massa aumenta di conseguenza la distanza tra i nucleoni e la forza nucleare forte perde la sua efficacia. Nella carta dei nuclidi Fig. 2.2 sono riportati gli elementi e i loro isotopi riportando il numero atomico ( $Z$ ) in funzione del numero di neutroni ( $N$ ) ed evidenziando i tipi di decadimento radioattivo per i diversi isotopi.

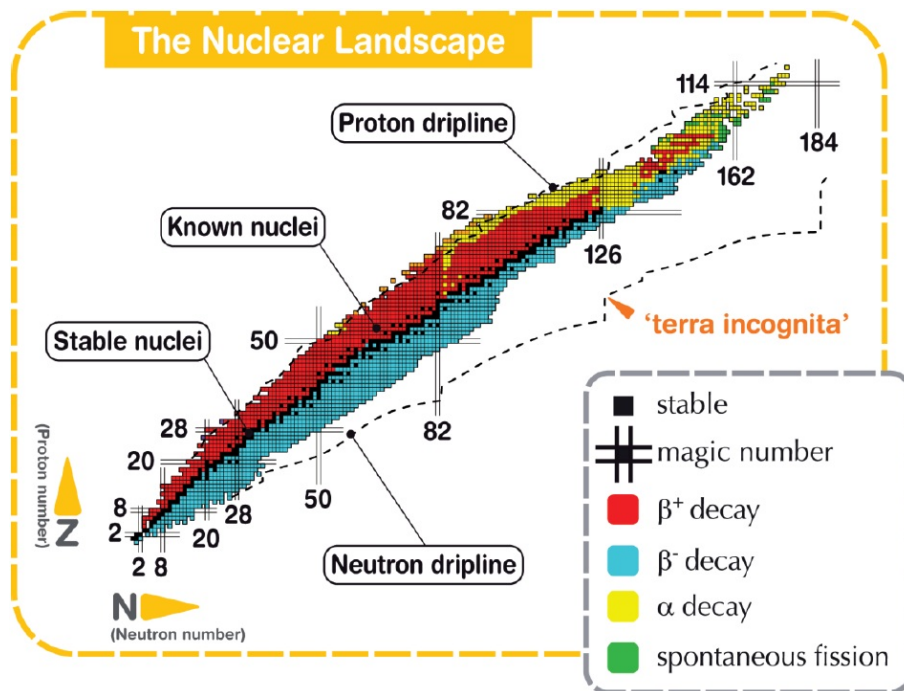


Fig. 2.2: Carta dei nuclidi.

I nuclei stabili sono segnati in colore nero e si può osservare che, almeno all'inizio, sono disposti su una retta inclinata a  $45^\circ$  ( $Z=N$ ). I nuclei con difetto di neutroni (proton-rich) o con eccesso di neutroni (neutron-rich) sono instabili e detti esotici (o radioattivi), poiché decadono emettendo particelle (alfa, beta, neutrini) e radiazioni elettromagnetiche, a seconda della tipologia di decadimento. Come detto in precedenza esistono dei limiti oltre i quali i nuclei sarebbero, a prescindere, troppo instabili e non potrebbero esistere, questi limiti sono delineati in Fig. 2.2 dalle driplines.

Attualmente sono state studiate le proprietà di circa 3600 nuclei che possono essere prodotti in laboratorio con più o meno facilità; si ritiene però, sulla base di considerazioni teoriche, che i nuclei esotici possano essere fino a 6000.

La carta dei nuclidi fornisce anche informazioni su particolari numeri di protoni o neutroni detti numeri magici. Questi numeri (per esempio 2,8,20,28,50,82,126) corrispondono alle configurazioni che rendono i nuclei particolarmente stabili; nuclei che possiedono sia  $Z$  sia  $N$  uguali ad un numero magico sono ancora migliori e quindi detti doppiamente magici. In natura gli elementi formati da nuclei doppiamente magici sono:  ${}^4\text{He}$  (2,2),  ${}^{16}\text{O}$  (8,8),  ${}^{40}\text{Ca}$  (20,20),  ${}^{208}\text{Pb}$  (82,126). La ricerca nel campo della fisica nucleare è ora molto focalizzata sui nuclei instabili, i quali hanno già permesso di confermare precedenti ipotesi di fondamentale importanza e di giungere ad importanti applicazioni nella fisica dello stato solido, in astrofisica ed anche in medicina nucleare.

## 2.3 RIBs facilities

I nuclei che si desidera studiare non sono presenti normalmente in natura (vengono infatti prodotti esclusivamente nelle fasi conclusive della vita delle stelle), per poterli studiare é quindi prima necessario produrli. Tra le tecniche per la produzione e lo studio degli isotopi instabili le RIBs (Radioactive Ion Beams) facilities hanno acquisito un grande interesse scientifico per la possibilitá di produrre un'ampia gamma di nuclei e la versatilitá del loro impiego per studi in diversi campi della scienza. In generale questi sistemi si caratterizzano per due funzioni principali: la produzione delle specie esotiche e il trasporto dei nuclei prodotti in diverse aree attrezzate per l'analisi sperimentale. Lo spostamento dei nuclei prodotti é reso possibile grazie alla loro ionizzazione ed é intervallato da una serie di operazioni di identificazione e purificazione degli stessi, la qualità del fascio di ioni radioattivi prodotti é infatti un fattore fondamentale per il successivo utilizzo.

### 2.3.1 ISOL facilities

Ad oggi le principali facilities per la produzione e il trasporto di ioni radioattivi utilizzando la tecnica ISOL (Isotope Separation On-Line), la quale consiste nella separazione degli isotopi in linea.

Questa tecnica richiede la seguente struttura:

- L'acceleratore primario
- Il complesso targhe-sistema di estrazione
- I separatori di massa isobari
- Il post acceleratore

Un fascio di particelle ad alta energia, proveniente dall'acceleratore primario, viene fatto collidere con un bersaglio (target) di materiale fissile; il risultato é la produzione degli isotopi radioattivi tramite reazioni nucleari (fissione, spallazione, frammentazione). Gli isotopi prodotti vengono quindi ionizzati per poter essere estratti e successivamente accelerati per differenza di potenziale. In particolare in un primo stadio si accelerano gli ioni e li si focalizza. Successivamente una serie di dispositivi elettrostatici permette il trasporto del fascio verso dei separatori magnetici di tipo Wien Filter all'interno dei quali il fascio viene opportunamente selezionato e purificato; i separatori sono quindi necessari se si vuole ottenere un fascio isobaricamente (massa) puro. Successivamente, gli ioni vengono post accelerati al livello di energia richiesto dal particolare esperimento. L'intensitá del fascio radioattivo prodotto é usualmente descritta dell'equazione:

$$I = \sigma \cdot \Phi \cdot N \cdot \varepsilon_1 \cdot \varepsilon_2 \cdot \varepsilon_3 \quad (2.3.1)$$

Dove:

- $\sigma$  é la sezione d'urto per le reazioni nucleari, ossia la possibilitá che avvenga una certa reazione nucleare.
- $\Phi$  é l'intensitá del fascio primario.
- $N$  é lo spessore del target.
- $\varepsilon_1$  é l'efficienza di rilascio del target.
- $\varepsilon_2$  é l'efficienza di ionizzazione.
- $\varepsilon_3$  é l'efficienza del percorso di estrazione e trasporto.

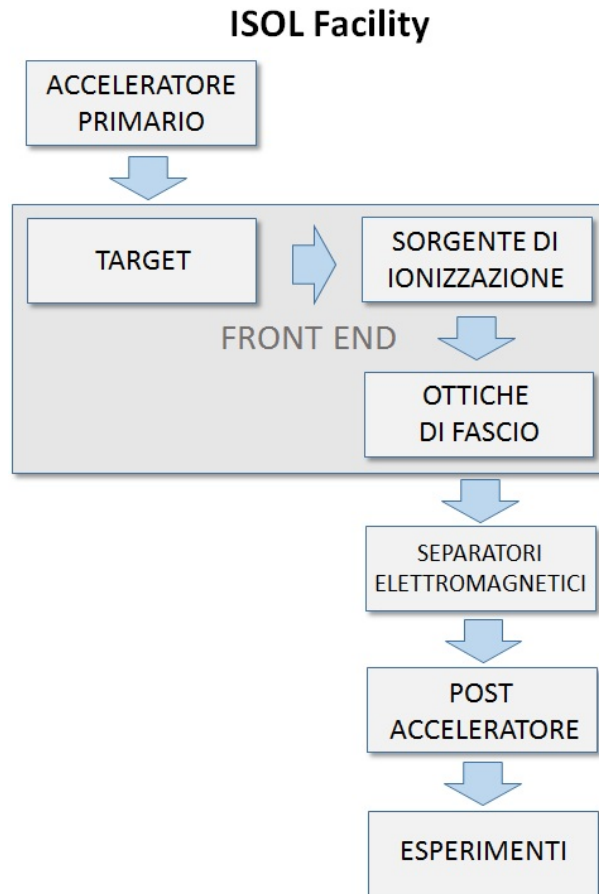


Fig. 2.3: Schema di una facility di tipo ISOL.

É chiaro quindi che per avere un funzionamento migliore possibile dell'intera facility, bisogna curare al meglio tutti questi aspetti, per poter cosí ottenere:

- La riduzione dei tempi di ritardo
- La massimizzazione della produzione senza perdere la purezza del fascio.

Il rilascio dei prodotti radioattivi dal target e l'estrazione dei nuclei esotici sono processi che dipendono fortemente dal livello termico: all'aumentare della temperatura si ottiene infatti una maggiore velocit  di diffusione delle particelle; questo risulta di particolare importanza soprattutto per atomi radioattivi caratterizzati da una breve emivita, poich  é facile che vengano "persi" per decadimento. Per questo motivo l'intero sistema viene mantenuto alla pi  alta temperatura possibile.

## 2.4 Il progetto SPES ai Laboratori Nazionali di Legnaro

Le grandi opportunit  scientifiche offerte dai RIB hanno spinto la comunit  scientifica a proporre la costruzione di una rete di facilities complementari, definite di intermedia generazione, con lo scopo di arrivare alla creazione di un'unica grande facilities europea di tipo ISOL, chiamata EURISOL. Vista la portata di tale progetto, sono coinvolti nell'iniziativa i principali laboratori nucleari europei con l'obiettivo dello studio e della progettazione di una struttura per la produzione di fasci di ioni radioattivi di qualit  significativamente superiore a quella attualmente disponibile. In tale direzione sono orientati anche i Laboratori Nazionali di Legnaro (LNL), presso i quali é in corso la costruzione di una facility



ISOL per la produzione di fasci di ioni esotici: il progetto SPES (Selective Production of Exotic Species). Tale progetto é coordinato a livello nazionale da INFN, ENEA (Bologna), Dipartimenti di Ingegneria Meccanica e di Scienze Chimiche dell'Università degli Studi di Padova e a livello internazionale prevede strette collaborazioni con CERN (Svizzera) ed i Laboratori di Oak Ridge (USA).

Il progetto ha come obiettivo la creazione di un apparato sperimentale in grado di produrre fasci di ioni radioattivi; al giorno d'oggi la maggior parte delle conoscenze che possediamo derivano dallo studio dei nuclei con eccesso di protoni, si pensa però che lo studio dei nuclei atomici con eccesso di neutroni, dove si trovano la maggior parte dei nuclei ancora incogniti, possa portare alla scoperta di nuove proprietà della struttura nucleare.

Il progetto SPES si sviluppa in quattro fasi:

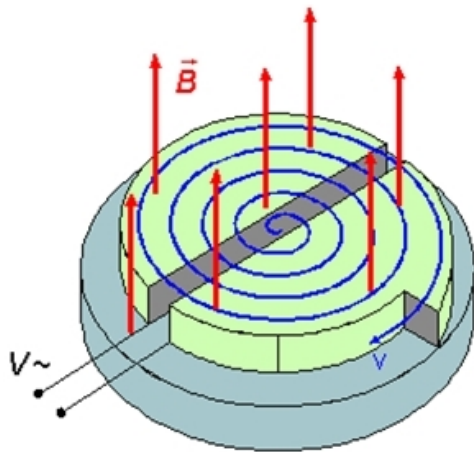
- SPES- $\alpha$ : in questa fase sono previste l'acquisto, l'installazione e la messa in funzione di un ciclotrone ad alte prestazioni: alta corrente d'uscita (circa 0.7mA) ed alta energia (fino a 70MeV). Il ciclotrone avrà una doppia uscita per consentire la duplice missione del laboratorio: la ricerca di base e le applicazioni tecnologiche. Uno dei due fasci servirá alla sezione di fisica nucleare che studierá la produzione di ioni con nucleo neutron-rich per mezzo di collisione di protoni su un target di UCx, l'altra per gli studi di fisica applicata.
- SPES- $\beta$ : in questa fase é prevista l'accelerazione di nuclei neutron-rich verso specifici target per la produzione di nuovi nuclei ancora neutron-rich caratterizzati da una brevissima vita e per questo non presenti normalmente in natura. Tali nuclei esistono solo negli ultimi istanti di vita delle stelle e si pensa serviranno per fornire informazioni sull'evoluzione stellare.
- SPES- $\gamma$ : in questa fase é previsto l'utilizzo del ciclotrone SPES- $\alpha$  per la produzione di nuclei radioattivi a scopo medico. Si vuole adottare questa nuova tecnica basata sull'acceleratore per produrre radionuclidi già conosciuti ma anche radiofarmaci innovativi (come quelli basati su  $^{82}\text{Sr}/^{82}\text{Rb}$  e  $^{68}\text{Ga}/^{68}\text{Ge}$ ).
- SPES- $\delta$ : in questa fase é previsto lo sviluppo di un'intensa sorgente neutronica dal ciclotrone e/o da un acceleratore lineare basato sulla RFQ (radio-frequency quadrupole technology). Tale sorgente verrà utilizzata dall'astrofisica nucleare, per la caratterizzazione dei rifiuti nucleari o per cure sperimentali per i tumori.

## 2.5 Configurazione della facility

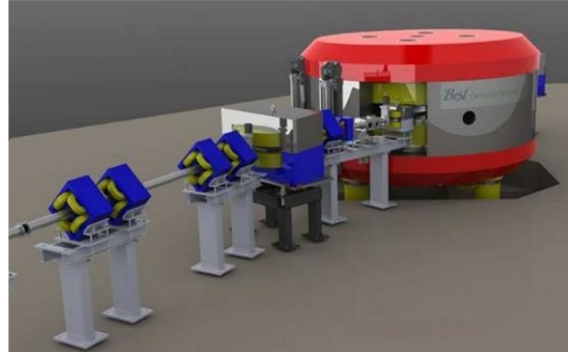
La facility é principalmente dedicata alla produzione di radioisotopi neutron-rich con masse da 80 a 160 una ottenuti a seguito di fissione di  $^{238}\text{U}$  con una resa massima prevista di  $10^{13}$  fissioni/s; la fissione é resa possibile dal bombardamento con un fascio protonico a 40 MeV, detto fascio primario, di un adeguato target costituito da sette dischi in carburo di uranio (UCx). Essendo previsto un elevato livello di radioattività nel target di produzione, é necessaria la progettazione di speciali infrastrutture in modo da garantire la sicurezza in termini di radioprotezione.

### 2.5.1 L'acceleratore primario

L'acceleratore primario produce il fascio di protoni che viene diretto verso il target per far avvenire la reazione nucleare. Per il progetto SPES é stato scelto come acceleratore primario un ciclotrone, un dispositivo che accelera particelle elettricamente cariche. Un vantaggio ulteriore é dato dalla possibilità di produrre con lo stesso ciclotrone due fasci di ioni indipendenti.



(a) Principio di funzionamento.



(b) Ciclotrone Best70p.

Fig. 2.4: Ciclotrone.

Come si osserva da Fig. 2.5 il suo funzionamento è dovuto ad una differenza di potenziale applicata tra due elettrodi semicilindrici associata ad un campo magnetico uniforme,  $B$  (T), ortogonale ad essi. La particella viene inizialmente messa in movimento dal campo elettrico, poi assume un'orbita circolare a causa della forza di Lorentz esercitata dal campo magnetico. Ogniqualevolta la particella passa nello spazio tra i due elettrodi viene accelerata aumentando così il proprio raggio di curvatura,  $r$  (m), sino ad un valore massimo  $R$  (m) per cui fuoriesce dal dispositivo. La tecnologia permette oggi di realizzare ciclotroni che producono fasci di intensità molto vicina a quella richiesta per il progetto SPES. Per questo motivo è stato scelto un dispositivo Best 70p sviluppato dall'azienda canadese Best Cyclotron System in grado di fornire una corrente massima di  $750 \mu A$  ripartita tra due fasci indipendenti di protoni aventi fino a 70 MeV di energia ciascuno. Tale dispositivo è da poco stato posizionato all'interno dell'edificio dedicato al progetto.



(a) Ciclotrone calato nel bunker.



(b) Posa del ciclotrone.

Fig. 2.5: Ciclotrone.

### 2.5.2 Il Front-End

Il Front-End, Fig. 2.6 è l'insieme di più elementi, aventi ciascuno una specifica funzione, che concorrono allo scopo di consentire la produzione e il trasporto dei fasci di ioni radioattivi. Si compone essenzialmente di due parti: il Front-End protonico (FEP), responsabile del trasporto del fascio protonico ed il Front-End radioattivo (FER) responsabile del trasporto del fascio radioattivo. Il Front-End svolge le seguenti funzioni:

- Garantisce sostegno e stabilità al canale protonico, alla camera target ed agli apparati necessari al trasporto del fascio radioattivo.
- Realizza un grado di vuoto sufficientemente spinto per il corretto funzionamento di tutti gli organi al suo interno e ad esso collegati.
- Consente un rapido aggancio della camera target.
- Provvede all'apporto verso la camera target di energia elettrica per il riscaldamento del blocco target e di acqua per il raffreddamento della camera stessa.

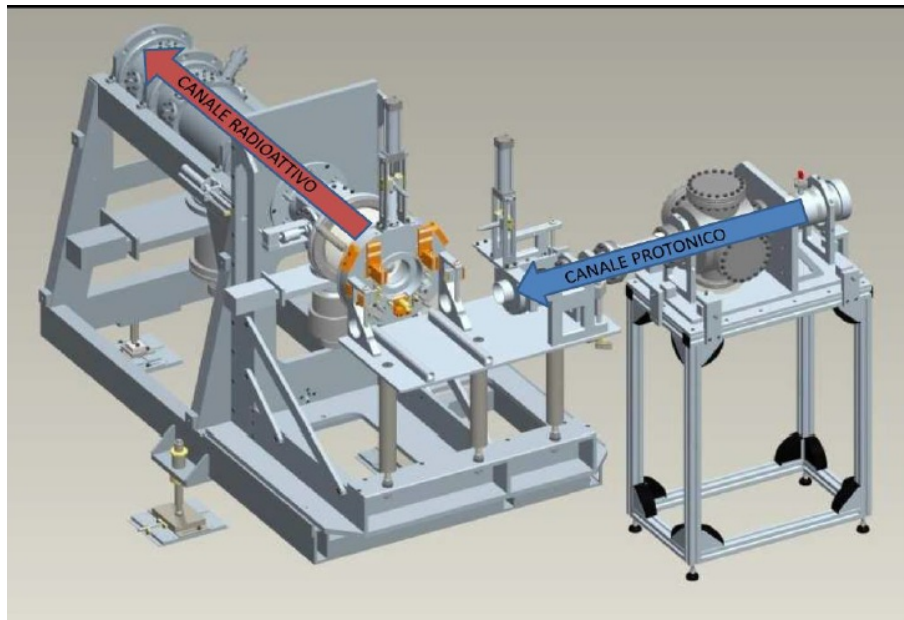


Fig. 2.6: Front End.

### 2.5.3 Il target di produzione e il sistema di estrazione e di ionizzazione

Sia il target di produzione dei radioisotopi, sia il sistema di estrazione e ionizzazione degli stessi, sono contenuti all'interno di una camera di forma cilindrica (camera target) Fig. 2.7, la quale viene raffreddata mediante un opportuno circuito dell'acqua per le elevate potenze in gioco. Sempre a causa delle alte temperature, per evitare l'ossidazione dei componenti presenti, l'interno della camera viene mantenuto in condizioni di alto vuoto (con pressioni dell'ordine dei  $10^{-6} mbar$ ); questa condizione è inoltre necessaria per aumentare il cammino libero medio delle particelle radioattive prodotte. Il volume della camera è delimitato da una flangia (plate) e da un coperchio (cover) a forma di pentola, entrambi realizzati in lega di alluminio, mentre la tenuta a vuoto è garantita da un O-Ring.

Il target è composto da sette dischi coassiali in UCx, aventi 40 mm di diametro e circa 1 mm di spessore ed opportunamente distanziati in direzione assiale al fine di dissipare,

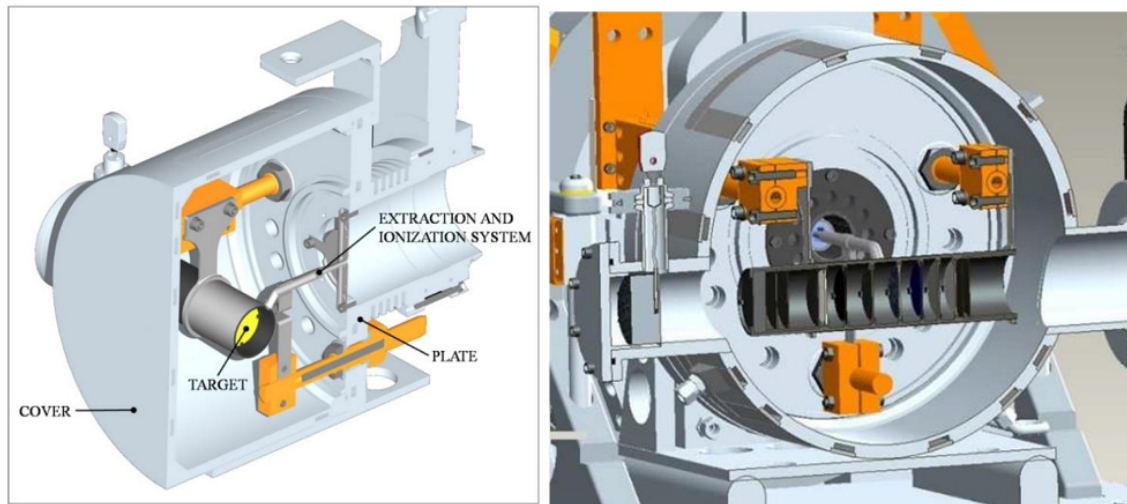


Fig. 2.7: Configurazione della camera target.

attraverso radiazione termica, la potenza sviluppata dal fascio di protoni. Essi sono contenuti in un tubo cavo di grafite, una scatola (box) avente un diametro esterno di 49 mm e una lunghezza di 200 mm (Fig. 2.8). Il fascio di protoni, prima di impattare contro i dischi, attraversa un sottile strato di grafite che non ne attenua l'energia, ma di fondamentale importanza poiché consente di schermare la zona attiva e di evitare un eccessivo raffreddamento del materiale in corrispondenza dell'ingresso del fascio. Dopo aver investito la finestra di grafite e i dischi di UCx, il fascio primario va ad impattare su tre dumper di grafite e sul fondo del tubo; questi elementi provvedono sia all'assorbimento definitivo del fascio (per evitare l'attivazione della camera esterna), sia ad evitare la fuoriuscita di particelle dal lato posteriore del target.

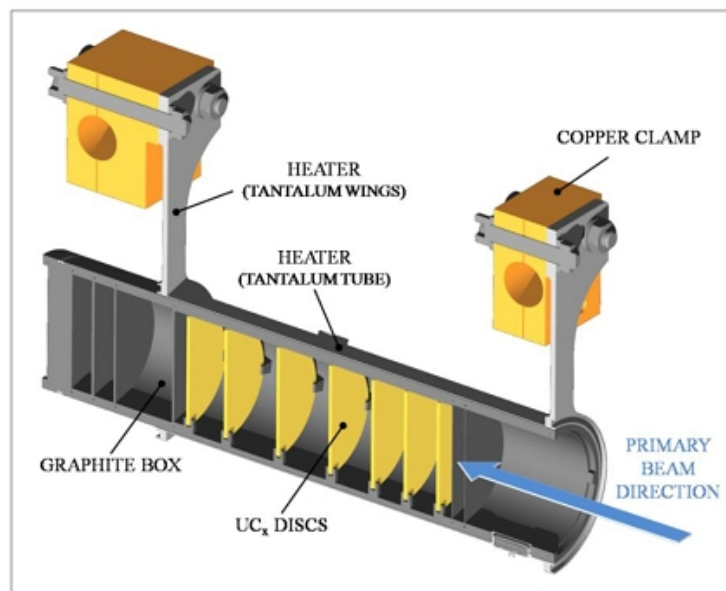


Fig. 2.8: Rappresentazione di un prototipo di bersaglio diretto del progetto SPES.

Come accennato precedentemente, la scatola deve mantenere la temperatura media di  $2000^{\circ}\text{C}$ , in modo da migliorare l'estrazione dei prodotti di fissione. Essendo la potenza del fascio di protoni non sufficiente a portare il target al livello di temperatura richiesto, è necessario introdurre un dispositivo indipendente avente la funzione di riscaldare e

schermare il target. Inoltre, il sistema di riscaldamento supplementare consente di evitare improvvisi sbalzi di temperatura molto pericolosi per l'integrità strutturale dei dischi. Il riscaldatore (heater) è composto da un tubo di spessore molto sottile e saldato ai bordi a due ali (wings) direttamente collegate a morsetti in rame (clamps); attraverso i morsetti è possibile far dissipare, per effetto Joule, il quantitativo di potenza necessario per mantenere il sistema scatola-dischi alla temperatura ottimale. Il materiale scelto per il riscaldatore è il tantalio: si tratta di un metallo altamente resistente alla corrosione, in grado di condurre energia elettrica e termica e di raggiungere temperature molto elevate.

Il processo di fissione nucleare, che si manifesta quando il fascio di protoni investe i sette dischi in carburo di uranio, produce nuclei radioattivi aventi massa compresa tra gli 80 ed i 160 uma; per la produzione di un RIB, la specie esotica desiderata deve però essere estratta dal target e ionizzata. Tale processo richiede del tempo e non può essere applicato ad isotopi aventi una vita media inferiore a poche decine di millisecondi. Il processo di estrazione avviene mediante la linea di trasferimento (transfer line), attraverso la quale gli isotopi in uscita dal target vengono indirizzati verso la sorgente di ionizzazione (ion source) dove sono ionizzati positivamente (ioni +1). Nell'attuale configurazione la linea di trasferimento (transfer line) è un sottile tubo di tantalio saldato al riscaldatore ad un'estremità e connesso meccanicamente alla sorgente di ionizzazione all'estremità opposta (Fig. 2.9). Come accade per il riscaldatore, anche il sistema linea di trasferimento-sorgente di ionizzazione viene riscaldato mediante dissipazione di potenza per effetto Joule; in questo modo la temperatura della sorgente arriva fino ai  $2400^{\circ}\text{C}$ .

La camera target è collegata al front-end e tra i due componenti è presente una differenza di potenziale pari a 30 kV; per questo è necessario, al fine di evitare il contatto diretto, interporre tra i due un isolante elettrico. La differenza di potenziale presente attira gli ioni radioattivi verso il front end; in particolare il componente che accoglie gli ioni in uscita dalla sorgente di ionizzazione è un elettrodo estrattore (electrode) realizzato in lega di titanio ( $\text{Ti}_6\text{Al}_4\text{V}$ ) e visibile in Fig. 2.9. In questo modo si forma il fascio di ioni radioattivi che verrà, dopo il passaggio attraverso i separatori elettromagnetici ed il post acceleratore, inviato alle sale sperimentali.

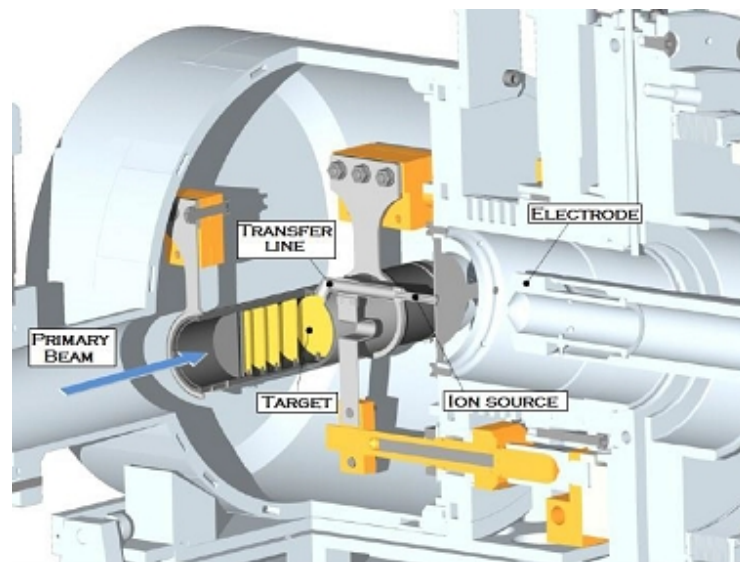


Fig. 2.9: Sistema di estrazione e ionizzazione.

Per poter ionizzare un atomo bisogna tener conto di una proprietà caratteristica di ogni elemento: il potenziale di ionizzazione; questo parametro definisce l'energia necessaria per portare un elettrone della sfera di valenza ad una distanza infinita dal nucleo. A secon-

da del potenziale di ionizzazione dell'elemento desiderato possono essere utilizzati diversi metodi di ionizzazione.

Nel Front-End SPES saranno disponibili i seguenti:

- Sorgente di ionizzazione superficiale (SIS - Surface Ionization Source), utilizzata per metalli alcalini e alcalino-terrosi che hanno bassi potenziali di ionizzazione.
- Sorgente di ionizzazione laser (RILIS - Resonant Ionization Laser Source), consente di ionizzare selettivamente elementi con medio potenziale di ionizzazione.
- Sorgente di ionizzazione al plasma (PIS - Plasma Ionization Source), in grado di ionizzare elementi con alto potenziale di ionizzazione ma senza poter operare una selezione.

Per le prime due tipologie di ionizzazione si fa riferimento alla medesima architettura, si tratta della sorgente basata sulla "MK1 ISOLDE" sviluppata al CERN.

Il componente principale di questa sorgente é la hot cavity: un tubicino di tantalio o renio all'interno del quale avviene la ionizzazione; é dotata di un opportuno invito conico e viene accoppiata per interferenza al foro centrale del supporto in tantalio. La gestione delle tolleranze é tale da impedire qualsiasi movimento relativo tra i due componenti. Dall'altra parte, il collegamento con la linea di trasferimento avviene tramite un tappo (bush) in tantalio, sul quale va in battuta la sorgente. Le elevate temperature raggiunte dalla sorgente di ionizzazione in condizioni di utilizzo, provocano una saldatura tra i componenti. Il collegamento tra il supporto e la flangia principale avviene mediante viti e tra questi due componenti viene interposto uno schermo in tantalio (screen), componente fondamentale per l'estrazione degli isotopi radioattivi.

**Surface Ion Source** La ionizzazione é causata dall'impatto degli isotopi radioattivi provenienti dal target sulla superficie interna della hot cavity. A causa di tale strisciamento gli isotopi cedono un elettrone e vengono quindi ionizzati positivamente (ioni +1). Questo é possibile se la minima energia necessaria per rimuovere un elettrone da una superficie (funzione di lavoro) é maggiore del potenziale di ionizzazione dell'isotopo. Nel caso in esame si riescono a produrre con elevata efficienza ioni positivi per elementi con potenziale di ionizzazione inferiore alla funzione di lavoro del tungsteno, pari a circa 5 eV. L'elevata temperatura a cui si trova la sorgente permette di avere un processo di ionizzazione efficiente; si favorisce infatti il moto Browniano e si incrementano i contatti isotopo-superficie. Lo svantaggio principale di tale metodo sta nel fatto che la ionizzazione non é selettiva, infatti non vengono ionizzati soltanto gli isotopi della specie desiderata. Si devono quindi predisporre dei separatori elettromagnetici in grado di selezionare, in base alla massa, le particelle presenti nel fascio. Questo però può non bastare per garantire la purezza del fascio; vi sono infatti isotopi di diversa specie tra loro isobari, aventi cioè lo stesso numero di massa atomica  $A$ , ma un diverso numero atomico  $Z$  (come ad esempio il  $^{132}\text{Cs}$  e lo  $^{132}\text{Sn}$ ). Per separare tali elementi sono necessari dei separatori isobari; bisogna però tener conto che tali dispositivi, oltre ad essere molto ingombranti e costosi, riducono notevolmente l'intensità del fascio.

**Resonant Ionization Laser Ion Source** Il metodo RILIS é oggi il piú potente strumento per la produzione di fasci di ioni radioattivi per le infrastrutture di tipo ISOL, in quanto fornisce un processo di ionizzazione selettiva e garantisce la soppressione di contaminazioni indesiderate a livello della sorgente di ionizzazione. Utilizza la stessa architettura della tecnica di ionizzazione superficiale e prevede di proiettare un laser all'interno della hot cavity (Fig. 2.11). Il laser é caratterizzato dalla sovrapposizione di piú raggi aventi

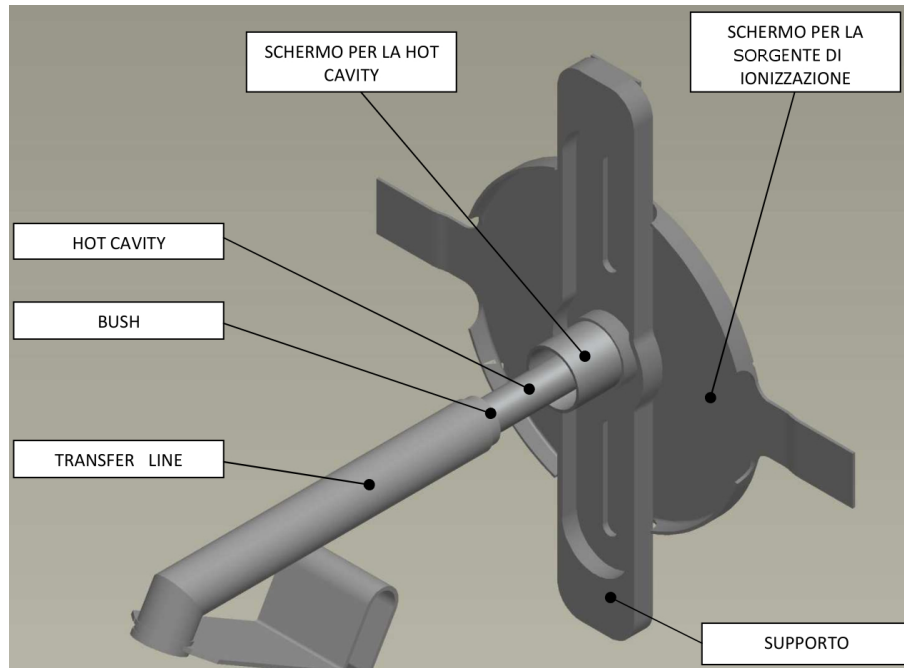


Fig. 2.10: Struttura usata per la ionizzazione superficiale (SIS).

lunghezza d'onda differente (2 o 3 a seconda dell'elemento da ionizzare), che corrispondono a diversi valori energetici ( $E = h \cdot f$ , dove  $h$  é la costante di Planck e  $f$  la frequenza d'onda). Tale laser irradia gli isotopi e fornisce loro l'energia necessaria per far compiere all'elettrone piú esterno dei salti quantici verso orbitali ancor piú lontani dal nucleo, fino alla separazione dall'atomo. In questo modo si ha formazione di uno ione positivo  $1+$ . La metodologia RILIS permette di ionizzare soltanto ioni radioattivi della specie di interesse producendo un fascio nel quale le contaminazioni sono minime. Rimane comunque la possibilitá che alcuni elementi (in particolare il cesio) vengano ionizzati per ionizzazione superficiale, provocando cosí contaminazione nel fascio. Per risolvere l'inconveniente é possibile sostituire il materiale con cui é realizzata la hot cavity: utilizzando il niobio la ionizzazione superficiale viene notevolmente attenuata.

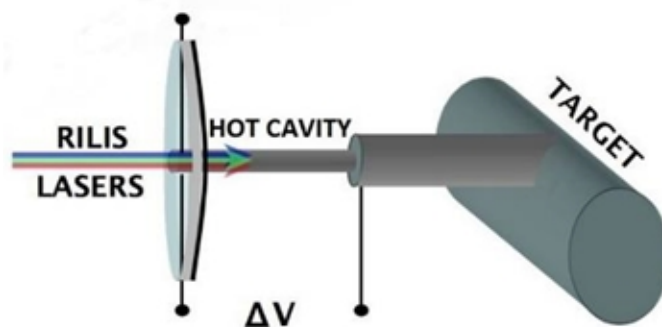


Fig. 2.11: Configurazione della sorgente RILIS.

Per avere un'efficienza di ionizzazione elevata é di fondamentale importanza limitare e controllare il disallineamento della hot cavity causato dall'espansione termica; se la hot cavity si disallinea viene a ridursi la zona di azione del laser e di conseguenza anche l'efficienza di ionizzazione (Fig. 2.12)

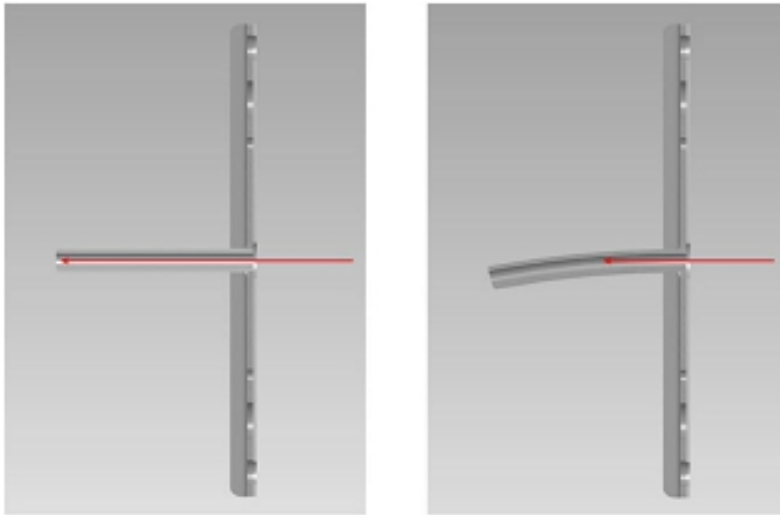


Fig. 2.12: Allineamento della hot cavity.

**Plasma ION source** Questa tecnica é in grado di ionizzare qualsiasi tipo di elemento, anche se con efficienze minori rispetto alle tecniche laser e superficiale. Il progetto SPES si basa su una sorgente di ionizzazione al plasma usata nell'infrastruttura ISOLDE del CERN di Ginevra, chiamata MK5. Un prototipo di questa sorgente é stato realizzato, testato e ottimizzato dal punto di vista elettrotermico dal gruppo di lavoro SPES.

In Fig. 2.13 vengono visualizzati gli elementi e il rispettivo metodo di ionizzazione. Gli elementi di transizione da  $Z = 40$  a  $Z = 46$  sono prodotti nel target, ma non vengono estratti a causa della loro bassa volatilità. In Fig. 2.14 sono invece rappresentati i radionuclidi che verranno prodotti nel progetto SPES.

		<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="width: 15px; height: 15px; background-color: yellow; border: 1px solid black; margin-right: 5px;"></div> surface ionization mechanism  <div style="width: 15px; height: 15px; background-color: lightgreen; border: 1px solid black; margin-right: 5px;"></div> laser ionization mechanism  <div style="width: 15px; height: 15px; background-color: red; border: 1px solid black; margin-right: 5px;"></div> electron impact ionization mechanism  <div style="width: 15px; height: 15px; background-color: lightblue; border: 1px solid black; margin-right: 5px;"></div> not extracted         </div>																							
1	1	H																	2	He					
2	3	Li	4																	5	6	7	8	9	10
																				13	14	15	16	17	
3	11	Na	12	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18						
																				Al	Si	P	S	Cl	Ar
4	19	K	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36						
											Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr						
5	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54							
6	55	56	57	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86							
7	87	88	89	104	105	106	107	108	109	110	Main fission ( $p \rightarrow ^{238}\text{U}$ ) fragments														

Fig. 2.13: Tavola periodica degli elementi e metodi di ionizzazione.



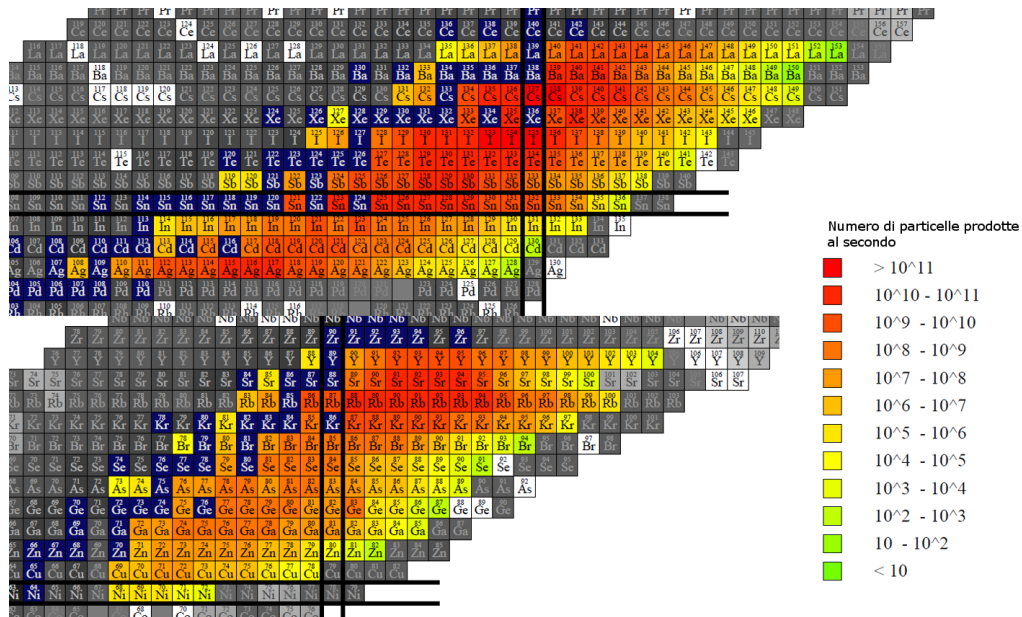


Fig. 2.14: Radionuclidi che verranno prodotti nel progetto SPES.

### 2.5.4 I separatori magnetici e la post accelerazione

Il fascio estratto dalla sorgente di ionizzazione è sottoposto ad una prima fase di purificazione nella quale, attraverso un separatore di massa, viene intrappolata una grande quantità di particelle non desiderate. La risoluzione di tale dispositivo ( $M/\Delta M$ , dove per  $M$  si intende la massa dell'isotopo) è pari a circa 300; si riescono così a separare in base alla massa i vari elementi. In Fig. 2.15 una rappresentazione dei moduli che compongono il front-end SPES.

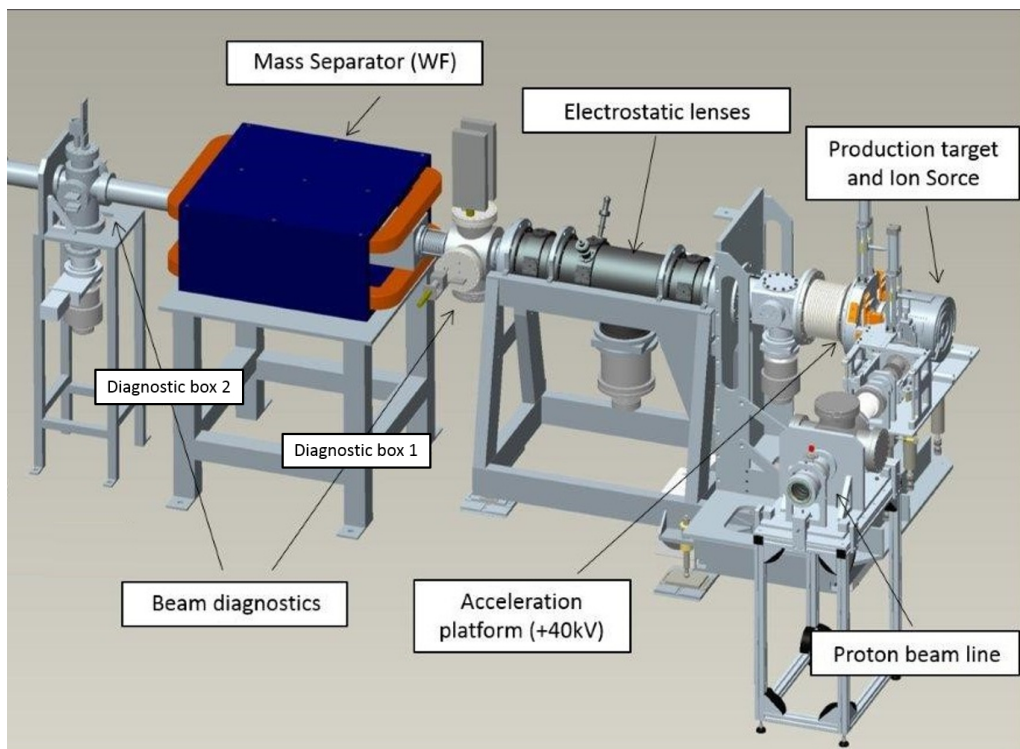


Fig. 2.15: Composizione del Front End.

Il separatore di massa é seguito da un separatore isobaro, avente una risoluzione pari a 20000, e che permette quindi di separare gli isotopi come il  $^{132}\text{Cs}$  e il  $^{132}\text{Sn}$  tra loro isobari. A questo punto il fascio radioattivo é purificato e può essere direttamente convogliato alle sale sperimentali per essere utilizzato in esperimenti che richiedono fasci radioattivi di bassissima energia, in alternativa può essere post accelerato per altri tipi di esperimenti. La post accelerazione viene ottimizzata mediante l'impiego di un Charge Breeder, un dispositivo in grado di incrementare la carica degli ioni del fascio, il quale passa successivamente in ALPI (acceleratore LINAC superconduttore). Questo acceleratore é da molti anni in funzione ai LNL, ma negli ultimi tempi é stato sottoposto ad un numero significativo di miglioramenti. In particolare, mentre prima consentiva la produzione di fasci di ioni stabili, ora può essere impiegato come un acceleratore di RIB.



Fig. 2.16: Post acceleratore ALPI.

## 2.6 Applicazioni dei fasci di ioni esotici

I fasci di ioni esotici hanno suscitato, nel corso degli anni, un crescente interesse, dovuto alle loro molteplici applicazioni non solo nel campo della fisica nucleare ma anche in astrofisica, medicina e fisica dello stato solido.

Nei prossimi paragrafi viene data una breve panoramica delle applicazioni nei vari ambiti di ricerca.

### 2.6.1 Applicazioni in fisica nucleare

Sono elencate di seguito le principali applicazioni nell'ambito della fisica nucleare.

#### Miglioramento e verifica del modello standard

L'attuale modello su cui si basa la fisica nucleare comprende tre delle quattro interazioni fondamentali, cioè l'interazione nucleare forte, l'interazione nucleare debole e l'elettromagnetismo (queste ultime talvolta identificate con il termine interazione elettrodebole) e descrive la funzione e le proprietà di tutte le particelle (note ed osservate) che costituiscono la materia. Sebbene tale modello abbia avuto un notevole successo, l'aspetto poco soddisfacente é che é stato costruito facendo uso di alcune assunzioni ad hoc non ancora pienamente verificate. Per chiarire l'origine e confermare la validità di tali assunzioni sono stati ideati complessi esperimenti di fisica nucleare, suggeriti da convincenti basi di natura teorica. La messa in opera di tali procedure sperimentali comporterebbe l'effettuazione di

misure precise sulle proprietà di decadimento di alcuni isotopi, che possono essere ricavate utilizzando come sorgente pura di ioni i fasci radioattivi prodotti nelle facilities.

### Studio della struttura di nuclei complessi

I nucleoni (protoni e neutroni) sono a loro volta costituiti da subparticelle dette quark. Esse esercitano un'azione fisica che va oltre i nucleoni nei quali sono confinate. Si osserva in particolare che tale interazione si manifesta in maniera diversa tra nucleoni liberi e nucleoni appartenenti allo stesso nucleo, in quanto influenzata dalla densità di protoni e neutroni associata al particolare tipo di nucleo. Attualmente non è stata ancora identificata una formula generale in grado di quantificare l'entità delle interazioni nucleari per tutti i tipi di nuclei, in quanto calcoli di natura quantomeccanica sono applicabili solamente ai nuclei più leggeri. La fisica nucleare quindi si è posta l'obiettivo di ottenere una trattazione unitaria in grado di:

- Permettere di determinare l'effettiva interazione tra le particelle nucleari.
- Eliminare le incongruenze presenti nel modello corrente.
- Essere applicabile anche ai nuclei esotici, ossia quelli con rapporto protoni/neutroni estremo.

A questo proposito i fasci di ioni radioattivi possono fornire un prezioso contributo.

### Misure della dimensione del nucleo: i nuclei halo

Le dimensioni del nucleo sono direttamente correlate al numero di nucleoni che lo costituiscono, ovvero il numero di massa ( $A$ ). Tale correlazione è espressa dalla formula:  $R = R_0 \cdot A^{\frac{1}{3}}$  dove con  $R$  è espresso il raggio del nucleo e  $R_0$  rappresenta una costante pari a 1.2 fermi ( $1 \text{ fermi} = 10^{-15} \text{ m}$ ).

Tuttavia in particolari condizioni, lontane dalla stabilità, si possono incontrare notevoli deviazioni dalla legge poiché le interazioni tra i nucleoni possono diventare così piccole da non garantire la formazione di un nucleo sotto forma di aggregato. Si possono quindi formare nuclei ad anello, detti nuclei "halo". Nuclei di questo tipo presentano una diversa distribuzione dei neutroni; si tratta in particolare di nuclei con sovrabbondanza di neutroni rispetto al corrispondente isotopo stabile, inoltre uno o due di questi neutroni risentono di una debole interazione con il nucleo, e vi orbitano intorno rimanendo quindi separati da esso (neutroni di valenza). Un esempio di nucleo "halo" è l'isotopo  $^{11}\text{Li}$ , il cui nucleo presenta due neutroni di valenza. In questo nuclide la dimensione media del nucleo è paragonabile a quella dell'isotopo  $^{48}\text{Ca}$ , se però si considera anche l'alone racchiuso dalle orbite dei due neutroni di valenza le dimensioni diventano confrontabili con quelle del nucleo molto più massivo dell'isotopo  $^{208}\text{Pb}$ .

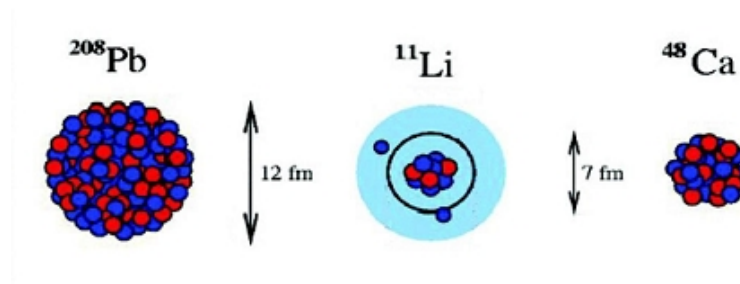


Fig. 2.17: Paragone tra la dimensione del nucleo di  $^{11}\text{Li}$  e quella di altri nuclei più massivi.

Il nucleo  $^{11}\text{Li}$  é un particolare sistema a tre corpi (i due neutroni ed il nucleo centrale), infatti rappresenta un esempio naturale di sistema borromeico. In topologia si definisce borromeico un sistema costituito da tre anelli tra loro legati inscindibilmente in maniera tale che tuttavia la soppressione di uno comporta la separazione degli altri due.



Fig. 2.18: Gli anelli Borromeici.

I nuclei ad anello sono detti anche borromeici perché infatti se il costituente di uno degli anelli viene in qualche modo rimosso, l'intero sistema diviene immediatamente instabile e i due anelli rimanenti si separano e si allontanano facilmente. Attualmente é possibile determinare la distribuzione dei protoni con esperimenti di spettroscopia atomica, utilizzando fasci radioattivi a bassa energia e luce laser. Se si cerca invece di determinare la distribuzione di tutti i nucleoni é necessario disporre di fasci ad alta energia, come quelli prodotti nelle facilities.

### Produzione di elementi superpesanti

In natura si ritrovano all'incirca 90 elementi, dall'idrogeno all'uranio. Grazie a reazioni di fusione nucleare negli ultimi anni é stato possibile sintetizzare elementi con elevato numero di massa estendendo il numero dei costituenti della tavola periodica fino ad elementi con numero atomico pari a 112, prospettando una futura estensione a 116. In particolare partendo da osservazioni preliminari si ritiene che questi elementi detti superpesanti, in cui la repulsione coulombiana assume valori elevati, riescano a formarsi intorno alla cosiddetta isola di stabilità, una configurazione formata da 114 protoni e 184 neutroni che sembra garantire la stabilità del nucleo. Le tecnologie più recenti consentono la disponibilità di fasci intensi, costituiti da nuclei instabili ricchi di neutroni (n-rich); se questi vengono accoppiati a target stabili, anch'essi ricchi di neutroni, potrebbero crearsi le condizioni adatte per approfondire tale fenomeno e aprire definitivamente la strada ad uno studio più accurato di questa realtà.

### 2.6.2 Applicazioni in fisica dello stato solido

La tecnica Radiotracer Diffusion, nata nel 1920, consiste nell'impiantare dei nuclei radioattivi all'interno di un sistema solido e poi studiarne il decadimento rilevando le particelle o la radiazione gamma da essi emessa. Tale tecnica consente di captare segnali anche da pochissimi atomi e rappresenta uno dei procedimenti più comuni per analizzare i metodi di diffusione atomica nei solidi. Il sistema ospitante può essere drogato con i radioisotopi "sonda" per diffusione, tramite reazione nucleare, oppure per impianto ionico; la scelta dell'atomo radioattivo da utilizzare per un determinato esperimento, viene fatta in base alla natura chimica e alle proprietà nucleari di quest'ultimo.

L'uso della tecnica Radiotracer Diffusion consente di:

- osservare tramite i prodotti di decadimento l'interazione tra l'atomo sonda e l'intorno reticolare che lo circonda
- informazioni riguardanti il campo elettrico e magnetico all'interno del cristallo
- studiare i processi diffusivi e le interazioni tra gli atomi sonda
- indagare i tipi di difetti presenti nel cristallo

### Drogaggio dei semiconduttori

Lo sviluppo dei semiconduttori di piccole dimensioni, aventi caratteristiche ottiche ed elettriche ottimali, richiede un controllo completo dei difetti che governano tali proprietà; possono essere sia intrinseci (come le vacanze interstiziali) che estrinseci (come i droganti e le impurità atomiche). Per tale motivo sia la ricerca di base che quella applicata stanno concentrando notevoli sforzi nello studio dei difetti e dell'attivazione elettrica dei droganti in diversi semiconduttori. Analogamente agli isotopi stabili, gli isotopi radioattivi influenzano le proprietà elettroniche ed ottiche dei semiconduttori in base alla loro natura chimica ed alla loro posizione all'interno del reticolo cristallino. Le proprietà elettroniche e ottiche dei semiconduttori dipendono dalla tipologia di semiconduttore e dalle sue dimensioni; in semiconduttori molto piccoli tali proprietà possono essere sensibilmente alterate da un difetto presente con concentrazione minore di  $10^{12}$  atomi/cm<sup>3</sup>. Per controllare in maniera affidabile le prestazioni dei semiconduttori sono quindi necessarie tecniche sperimentali che combinino un'alta sensibilità chimica con un'alta sensibilità per la determinazione di basse concentrazioni di difetti. Per decenni la principale tecnica di rilevazione delle impurità all'interno di un cristallo è stata il channeling: in tale tecnica un fascio di ioni viene guidato lungo le righe atomiche o lungo i piani del cristallo (canali), tuttavia non è possibile determinare concentrazioni di difetti inferiori a  $10^{18}$  atomi/cm<sup>3</sup>. La sensibilità di tale tecnica può essere profondamente aumentata impiantando all'interno del cristallo impurezze radioattive che emettono particelle cariche (emission channeling, Fig. 2.19). La misura dell'emissione lungo differenti direzioni cristalline, permette la determinazione del sito cristallografico dell'atomo emittente con un'accuratezza di pochi decimi di angstrom.

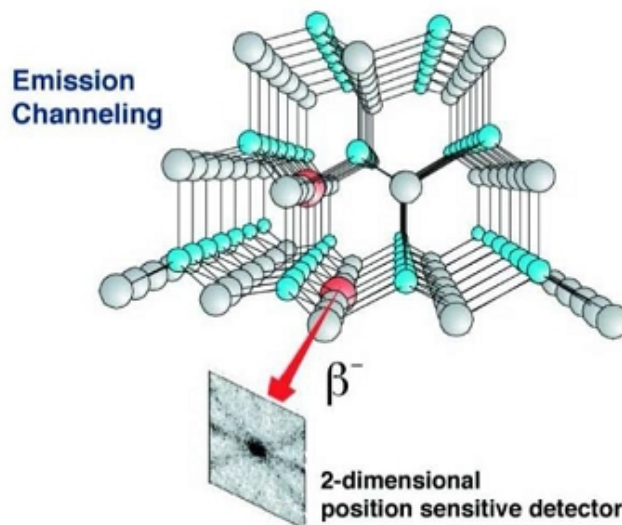


Fig. 2.19: Emission channeling degli elettroni emessi da atomi radioattivi.

### 2.6.3 Applicazioni in medicina nucleare

I fasci di ioni radioattivi possono avere interessanti applicazioni mediche sia nel campo della diagnosi sia nel campo del trattamento di patologie tumorali. Di seguito sono illustrate le principali applicazioni in campo medico che attualmente sono state prese in considerazione.

#### La tomografia ad emissione positronica

La tomografia ad emissione positronica (PET) é una tecnica di diagnostica usata in medicina nucleare che sfrutta l'emissione di positroni per realizzare immagini tridimensionali o mappe ad alta risoluzione degli organi interni dei pazienti.

Il positrone, detto anche antielettrone, é l'equivalente di antimateria dell'elettrone e ha carica elettrica pari a  $+1$ . L'antimateria é la materia composta da particelle aventi la stessa massa ma caratteristiche opposte rispetto a quelle che costituiscono la materia ordinaria. Quando un positrone incontra un elettrone, essi si annichiliscono e la loro massa si converte in energia tramite emissione di due fotoni nella banda dei raggi gamma. Un positrone può essere generato per decadimento radioattivo o all'interazione con la materia di fotoni aventi energia superiore a  $1.022$  MeV. Anche se viene principalmente utilizzata per studiare le interazioni tra particelle elementari, l'antimateria trova applicazione tecnologica nella PET. Tale procedura consiste nell'iniezione nel soggetto da analizzare di un isotopo tracciante di breve vita media, legato chimicamente ad una molecola attiva a livello metabolico (spesso uno zucchero). Dopo aver atteso il tempo necessario perché la molecola raggiunga la giusta concentrazione all'interno dei tessuti da analizzare, il soggetto viene posizionato nello scanner (Fig. 2.20).



Fig. 2.20: Scanner utilizzato nella tecnica di rilevazione PET.

In seguito l'isotopo decade emettendo un positrone il quale, dopo un percorso al massimo di pochi millimetri, si annichilisce con un elettrone producendo una coppia di fotoni che si muovono in direzioni opposte tra loro (sfasate di  $180^\circ$ ). Questi ultimi vengono rilevati dal dispositivo di scansione grazie all'impiego di speciali tubi fotomoltiplicatori. La rilevazione deve essere simultanea per entrambi i fotoni: perciò quelli che non raggiungono il rivelatore in coppia entro pochi nanosecondi non vengono presi in considerazione. La direzione di ogni coppia di fotoni individua una retta, quindi tramite due coppie é possibile individuare il punto di emissione dei fotoni. Lo scanner utilizza la rilevazione delle

coppie di fotoni per realizzare una mappa della densità dell'isotopo nel corpo che mostra la diversa concentrazione della molecola campione nei tessuti.

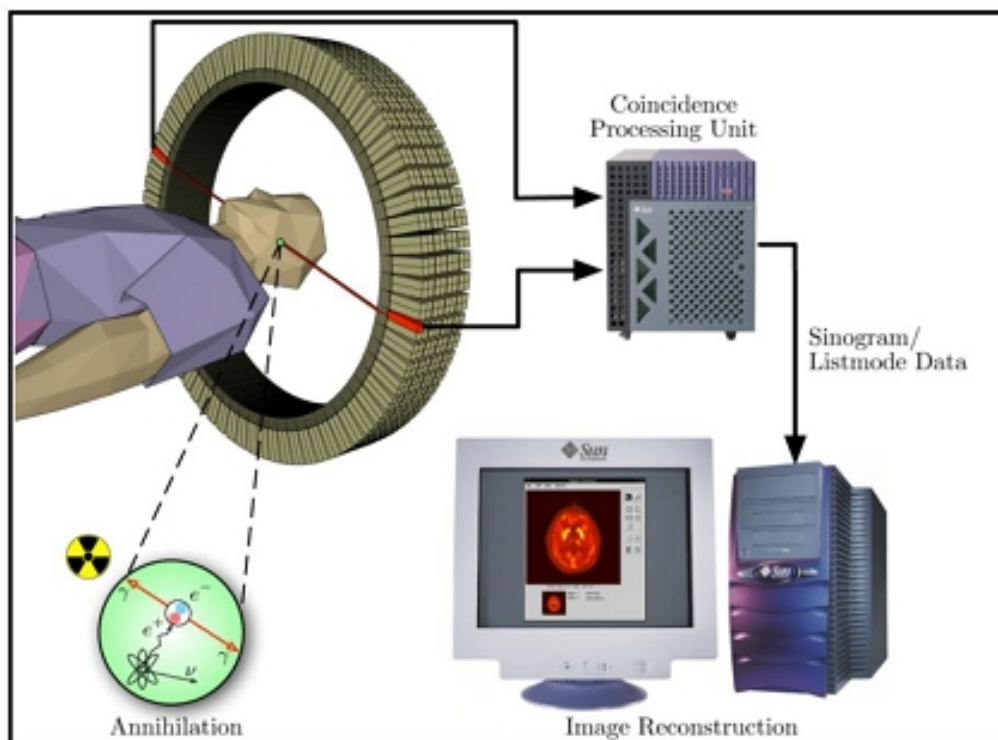


Fig. 2.21: Schema riassuntivo della tecnica di analisi PET e modalità di acquisizione e presentazione dei dati..

Tale mappa viene poi letta ed interpretata da uno specialista in medicina nucleare o radiologia al fine di determinare una diagnosi. Mentre altri metodi di scansione, come la TAC (Tomografia Assiale Computerizzata), permettono di identificare alterazioni organiche e anatomiche del corpo umano, la PET è in grado di rilevare anche alterazioni a livello biologico-molecolare che spesso precedono l'alterazione anatomica. Ciò è possibile tramite l'uso di opportuni marcatori che presentano una diversa velocità di assorbimento a seconda del tessuto interessato. Inoltre, con una scansione PET è possibile visualizzare e quantificare con discreta precisione il cambio di afflusso sanguigno nelle varie strutture anatomiche.

La PET viene utilizzata estensivamente negli ambiti di oncologia clinica per l'analisi dei tumori e la ricerca i metastasi, e nelle ricerche cardiologiche e neurologiche. In particolare, questa tecnica gioca un ruolo sempre maggiore nella verifica della risposta alla terapia, specialmente per le terapie anticancro. Si prospettano quindi ulteriori applicazioni e sviluppi di questa tecnologia medica. I radionuclidi utilizzati nella scansione PET sono generalmente isotopi con breve tempo di dimezzamento, come  $^{11}\text{C}$  ( 20 min),  $^{13}\text{N}$  ( 10min),  $^{15}\text{O}$  ( 2 min) e  $^{18}\text{F}$  ( 110 min). Per via del loro basso tempo di dimezzamento, i radioisotopi devono essere prodotti da un ciclotrone posizionato in prossimità dello scansionatore PET. La PET gioca un ruolo sempre maggiore nella verifica della risposta alla terapia, specialmente in particolari terapie anti-cancro; si prospettano dunque per essa sempre maggiori applicazioni e sviluppi.

### La produzione di radiofarmaci

Nell'ambito del progetto SPES è in fase di studio la possibilità di produzione di radioisotopi utili per applicazioni di medicina nucleare. In tale ambito la ricerca è molto viva e

nell'ottica di una collaborazione tra i laboratori di fisica nucleare e gli ospedali, si stanno studiando i possibili radiofarmaci producibili. Saranno prodotti soprattutto radioisotopi neutron-rich-emitters, dalle applicazioni in campo terapeutico molto promettenti. Ciò che rende questo studio significativo è la possibilità di produrre radiofarmaci ad altissima purezza, privi di contaminanti. Attualmente è stata condotta con successo una sperimentazione pilota con  $^{89}\text{Sr}$ , uno dei radioisotopi più promettenti, per la verifica della realizzabilità del progetto.

#### 2.6.4 Applicazioni in astrofisica

L'astrofisica nucleare ha come scopo la comprensione della struttura, dell'evoluzione e della composizione dell'universo e di ciò che lo costituisce.

Le stelle generano energia tramite reazioni nucleari che coinvolgono sia nuclei stabili sia radioattivi. Tali reazioni talvolta procedono stabilmente e durano miliardi di anni, altre volte hanno un carattere esplosivo e una durata di pochi minuti o secondi. Nelle diverse fasi di vita delle stelle vengono sintetizzati nuovi elementi chimici sia tramite processi di nucleosintesi, che seguono strettamente la valle di stabilità, sia attraverso processi che si svolgono nella terra incognita.

Per sviluppare un modello che descriva il meccanismo di nucleosintesi, è necessario misurare le rese delle reazioni nucleari relative ai principali cicli astrofisici e le caratteristiche di decadimento di molti nuclei tuttora sconosciuti. Queste informazioni essenziali includono i tempi di vita, le masse ed i principali canali di decadimento dei nuclei instabili e possono essere misurate unicamente se si ha a disposizione un fascio radioattivo. Perciò, si prevede che la nuova generazione di infrastrutture per la produzione di fasci radioattivi, risulterà di fondamentale importanza per la comprensione della sintesi elementare dell'universo.

## 2.7 Conclusioni

Negli ultimi anni ha assunto sempre maggiore importanza lo studio dei fasci di ioni radioattivi, RIBs, in particolare dei meno noti isotopi neutron-rich al di sotto della valle di stabilità. Ciò ha spinto la comunità scientifica internazionale a costruire numerose RIBs facilities per la loro produzione. Il progetto SPES in fase di realizzazione presso i Laboratori Nazionali di Legnaro costituisce uno stadio intermedio per la costruzione di un'unica grande facility europea di tipo ISOL ad alte prestazioni denominata EURISOL. Tale progetto prevede l'allestimento di una facility di seconda generazione che verrà suddivisa in diverse sezioni sperimentali con aree dedicate a seconda del loro ambito di ricerca. La sezione SPES- $\beta$  prevede la produzione dei fasci con una resa di  $10^{13}$  fissioni/s all'interno un target formato da sette dischi coassiali in carburo di uranio bombardato da un fascio protonico con corrente massima di 0.7 mA a 70 MeV. Le applicazioni sperimentali di questi fasci spaziano dalla fisica teorica a quella applicata estendendosi ai campi della fisica nucleare, dell'astrofisica, della fisica dello stato solido e della medicina.



## Strumentazione per la fotoionizzazione Laser[4]

### 3.1 Introduzione

Questo capitolo si pone l'obiettivo d'introdurre il concetto di radiazione Laser, soffermandosi in particolare sul principio fisico che ne sta alla base (emissione stimolata dei fotoni) e sulla struttura interna dei dispositivi in grado di produrla. Viene posta particolare attenzione sui dispositivi Laser a dye, sulla loro struttura (che utilizza un reticolo di diffrazione) ed il loro funzionamento; vengono anche analizzati alcuni componenti che sono spesso abbinati a questi dispositivi: il filtro Etalon ed il cristallo di seconda armonica. Viene infine descritto il funzionamento di un monocromatore, il quale fa un particolare uso del reticolo di diffrazione.

### 3.2 La fotoionizzazione

Un atomo in condizioni "normali" possiede una carica elettrica complessiva pari a zero, in esso ci sono cioè un ugual numero di protoni ed elettroni. La ionizzazione é il fenomeno che consiste nello "strappare" via uno o piú elettroni da un atomo ed ottenerne quindi uno con carica elettrica positiva (chiamato ione  $1+$ ).

Come noto, gli elettroni orbitano attorno al nucleo separati da precise "distanze" caratterizzate da altrettanto precisi livelli di energia. Per poter spostare un elettrone da un livello al successivo, occorre fornire esattamente la differenza di energia necessaria per il salto; in particolare per allontanare definitivamente un elettrone si deve fornire una quantità di energia detta di ionizzazione, che varia a seconda della specie atomica. Un metodo per far questo é quello d'inviare fotoni (fotoionizzazione) che possedano tale energia, il fotone verrà assorbito e l'elettrone compirà il "salto"; se l'elettrone non viene "strappato" ma solo spostato, dopo un certo tempo l'atomo tenderá a tornare allo stato fondamentale: l'elettrone tornerà "giú" di un livello e la differenza di energia verrà ri-emessa sotto forma di fotone.

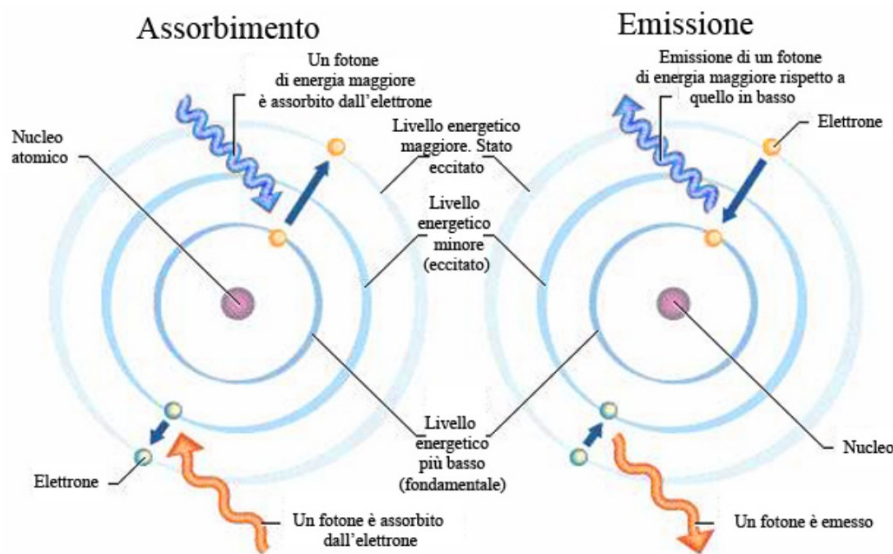


Fig. 3.1: Assorbimento ed eccitazione di un fotone.

Come si può intuire, questa tecnica consente un'altissima selettività di ionizzazione, agendo cioè su un vapore di più elementi, si possono ricavare ioni esclusivamente di un preciso elemento, fornendogli la specifica energia di ionizzazione. Si rende quindi necessario un fascio di fotoni aventi tutti la stessa energia che, com'è noto, è legata alla lunghezza d'onda (e quindi al colore).

### 3.3 L'emissione stimolata e la radiazione laser

Normalmente, se la luce attraversa un materiale, viene assorbita da esso man mano che avanza, questo accade perché ogni fotone che impatta contro un atomo, viene assorbito e gli cede la propria energia eccitandolo. Può avvenire però anche che il fotone impatti contro un atomo già eccitato, in questo caso avviene una diseccitazione e la conseguente emissione di energia (un fotone alla stessa frequenza e nella stessa direzione di quello incidente), in questo caso si parla di emissione stimolata.

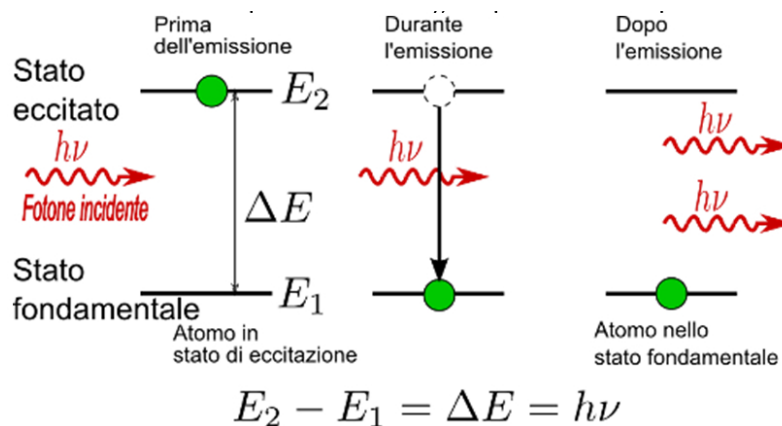


Fig. 3.2: Emissione stimolata di un fotone.

Se quindi s'interviene dall'esterno eccitando gli atomi del materiale, la probabilità che i fotoni, attraversandolo, generino emissione stimolata anziché essere assorbiti, aumenta all'aumentare del rapporto tra atomi eccitati ed atomi non eccitati. Operando un'inversione di popolazione, cioè la presenza di più atomi eccitati che atomi normali, la luce che attraversa il materiale guadagnerà potenza invece di perderla.

Utilizzando questo fenomeno, nel corso del XX secolo si è concretizzata la possibilità di ottenere un'amplificazione di un fascio di luce attraverso la costruzione di dispositivi denominati laser (Light Amplification by Stimulated Emission of Radiation)

### 3.4 Principio di funzionamento dei dispositivi LASER

Un Laser è costituito, in prima approssimazione, da un materiale (mezzo attivo) eccitato esternamente, posizionato tra due specchi; tale configurazione prende il nome di cavità risonante e crea un cammino chiuso per il fascio di luce, il quale, ogni volta che attraversa il mezzo attivo, guadagna sempre più fotoni.

Affinché il materiale diventi "attivo" è necessario che ci sia un sistema di pompaggio (una lampada flash o un altro laser) che dia vita costantemente ad un'inversione di popolazione. In Fig. 3.3 è schematizzata la struttura di un laser mentre in Fig. 3.4 si può osservare come sia possibile ottenere inversione di popolazione: il sistema di pompaggio porta gli elettroni nel livello più alto, a questo punto essi tendono a scendere compiendo due transizioni. È evidente che siano necessari materiali che possiedano almeno tre livelli: se ce ne fossero solo due la radiazione di pompaggio agirebbe infatti sugli elettroni facendoli costantemente

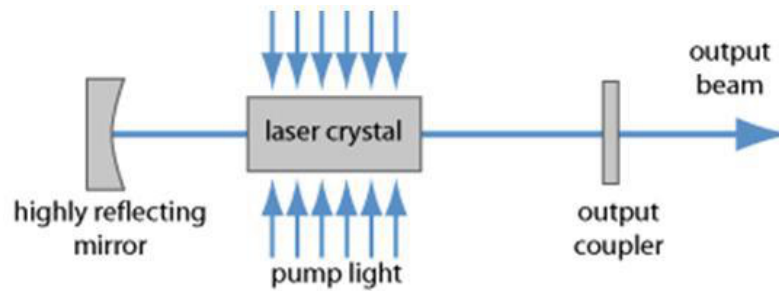


Fig. 3.3: Rappresentazione schematica di una cavità risonante laser.

salire e scendere senza ottenere l'effetto voluto; per questo motivo si cerca anche di ottenere materiali nei quali il tempo di permanenza nel secondo livello sia molto breve, per ottenere così un accumulo di elettroni nel primo livello che man mano scendano allo zero emettendo la luce desiderata.

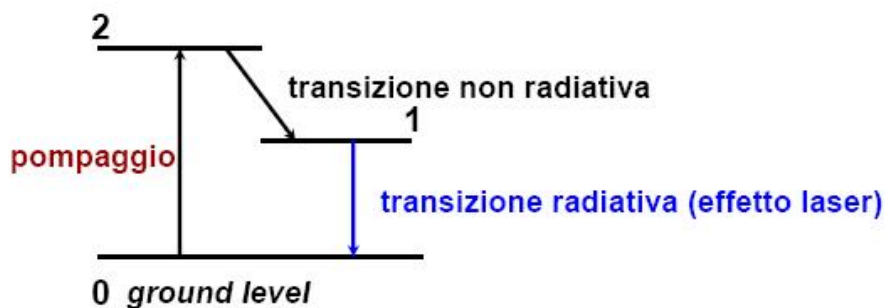


Fig. 3.4: Livelli di transizione degli elettroni.

L'estrazione di energia dalla cavità avviene mediante la riduzione della riflettività di uno dei due specchi che diventa quindi l'uscita per la radiazione laser. Chiaramente solo una parte della radiazione fornita dalla pompa viene assorbita dal materiale attivo e convertita in emissione laser, il resto viene perso o convertito in calore. Considerando inoltre le altre perdite si può capire come l'efficienza di questi dispositivi sia generalmente molto bassa. Per avere azione laser devono quindi agire in accordo due attori principali: la cavità ed il mezzo attivo.

A riguardo della cavità, pensando al fascio di luce come tante onde a diversa frequenza, si capisce come il continuo rimbalzare possa dare luogo ad un'interferenza costruttiva oppure distruttiva: se il fascio di partenza e quello che torna nello stesso punto dopo aver compiuto un intero "giro", hanno la stessa fase, si ha interferenza costruttiva e la radiazione a quella specifica lunghezza d'onda non viene trasmessa. Per i fasci che si propagano lungo l'asse della cavità la condizione costruttiva si verifica quando viene soddisfatta l'equazione

$$m \frac{\lambda}{2} = nL \quad (3.4.1)$$

dove  $m$  è un numero intero,  $L$  è la distanza tra gli specchi ed  $n$  rappresenta l'indice di rifrazione del mezzo attivo. Ad ogni lunghezza di cavità corrispondono quindi delle precise lunghezze di risonanza.

La distribuzione, spaziale e temporale, del campo elettromagnetico corrispondente ad ogni frequenza di risonanza costituisce un "modo" di oscillazione della cavità; dall'espressione (3.4.1) si ricava che la distanza in frequenza tra due modi adiacenti è:

$$\Delta\nu = \frac{c}{n2L} \quad (3.4.2)$$

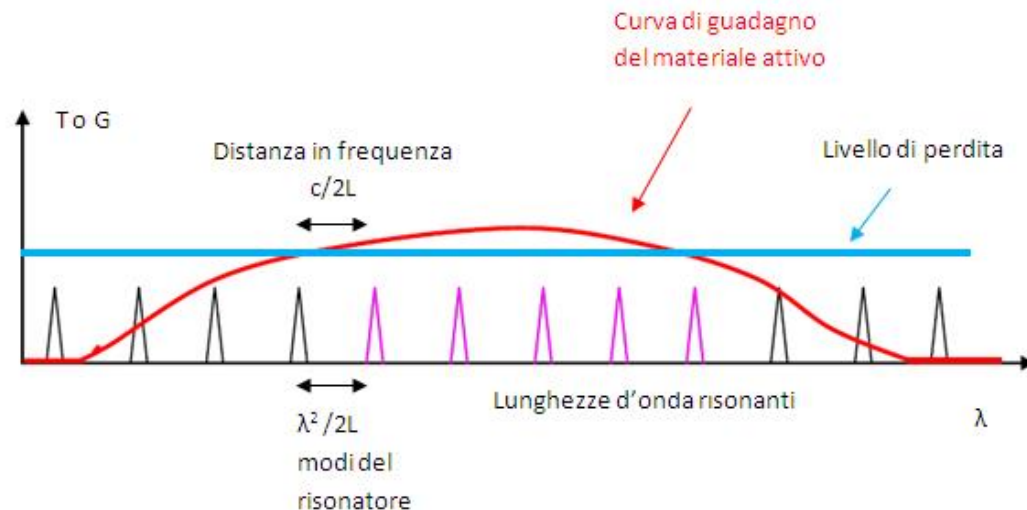


Fig. 3.5: Amplificazione dei modi longitudinali laddove il guadagno supera le perdite.

Guardando al mezzo attivo, esso è in grado di amplificare la radiazione all'interno di un determinato intervallo di frequenze, tale curva dipende dalle proprietà del materiale e pertanto cambia a seconda del materiale utilizzato.

All'interno della cavità oscilleranno quindi tutti i modi che soddisfano la condizione di fase derivante dalle equazioni 3.4.1 e 3.4.2 ma nel tempo guadagneranno energia solo quelli che rientrano nell'azione amplificatrice del mezzo attivo.

Per poter ottenere l'isolamento di un particolare modo si possono seguire diverse strade:

- utilizzare un mezzo attivo con larghezza di guadagno sufficientemente stretta
- inserire un filtro all'interno della cavità
- aumentare la distanza in frequenza dei modi diminuendo la lunghezza di cavità

Grazie alla sua struttura, la radiazione laser ha le seguenti importanti caratteristiche:

- **Direzionalità:** al contrario delle radiazioni elettromagnetiche convenzionali, la luce laser viene emessa in un'unica direzione.
- **Monocromaticità:** come già accennato, a seconda degli atomi presenti nel mezzo attivo, la radiazione laser possiede una o poche lunghezze d'onda.
- **Brillanza:** come diretta conseguenza delle due caratteristiche precedenti, la radiazione laser ha una quantità di energia emessa per unità di angolo solido, enormemente maggiore rispetto alle sorgenti tradizionali.
- **Coerenza:** nell'emissione stimolata ogni fotone ha la stessa fase del fotone che ha indotto l'emissione, quindi la fase viene mantenuta nel tempo e nello spazio.

### 3.4.1 Modalità Q-switch

Spesso è necessario disporre di un laser che produca impulsi di breve durata ed elevata intensità, ciò si può ottenere con la tecnica del Q-switching che consiste nel modificare il fattore di qualità Q della cavità risonante. Tale fattore è adimensionale ed indica la qualità dell'oscillatore, esso è uguale a  $2\pi$  volte il rapporto tra l'energia immagazzinata

e l'energia dissipata in un ciclo d'oscillazione; alti valori di  $Q$  corrispondono ad un basso tasso di perdita energetica in relazione all'energia immagazzinata all'interno della cavità risonante.

Il metodo consiste nel prevenire temporaneamente le oscillazioni laser tenendo molto alte le perdite nella cavità: l'emissione stimolata é soppressa e si ottiene un'elevatissima inversione di popolazione. Al termine di questo periodo l'energia fornita dalla pompa al mezzo attivo si é accumulata nel mezzo stesso. A questo punto si riducono drasticamente le perdite e s'innesca quindi un'emissione stimolata molto forte che genera un impulso corto ma molto potente che riporta il guadagno al livello stazionario.

### 3.5 Laser a colorante (dye)

Come già accennato, la lunghezza d'onda della radiazione laser dipende dal mezzo attivo che viene utilizzato, per questo motivo ci sono diverse tipologie di mezzi attivi che possono essere materiali solidi (es. Nd:YAG), gas (es. XeCl) o liquidi (rodamina 6g, )

I laser a colorante (laser dye) sono le sorgenti di radiazioni laser piú versatili in assoluto, essi utilizzano, come mezzi attivi, soluzioni di particolari coloranti. Rispetto ai laser che utilizzano sostanze allo stato solido o gassoso, la caratteristica piú significativa di tali dispositivi é l'elevata larghezza di banda di guadagno (decine di nanometri); utilizzando lo stesso colorante si puó quindi scegliere tra un'ampia gamma di lunghezze d'onda. Inoltre utilizzando lo stesso laser fisico (macchinario), si puó cambiare colorante per avere a disposizione le lunghezze d'onda che servono in un particolare momento o esperimento, dall'ultravioletto al vicino infrarosso. In Fig. 3.6 sono raccolte, a titolo di esempio, le diverse lunghezze d'onda, espresse in nm, ottenibili con l'utilizzo di diversi coloranti.

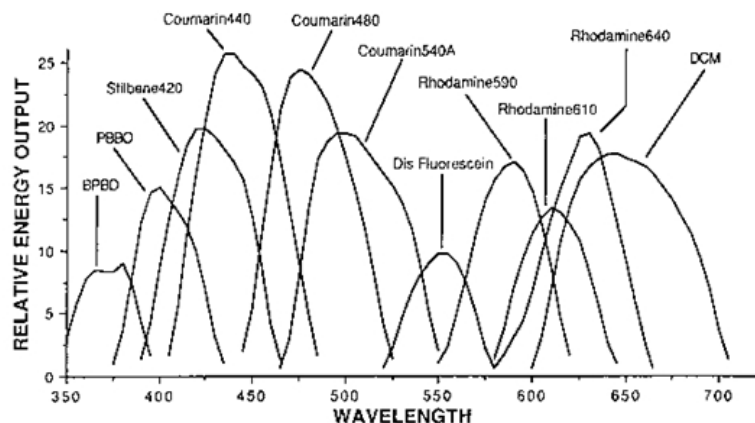


Fig. 3.6: Rappresentazione dell'ampio campo di  $\lambda$  ottenibile con diversi coloranti.[7]

Molti dei coloranti utilizzati sono organici e poiché tali composti tendono a degradarsi per effetto della luce a cui sono sottoposti, la soluzione viene costantemente reintegrata da un serbatoio esterno: il liquido viene cioè fatto circolare in un apposito contenitore di vetro facendo uso di una pompa esterna. Grazie alla notevole larghezza di banda di emissione che caratterizza i laser a dye, essi si prestano molto bene ad essere utilizzati in moltissime applicazioni; nella maggior parte dei casi é però necessario poter cambiare frequentemente e velocemente la lunghezza d'onda selezionata, per far questo ci si puó avvalere di laser "sintonizzabili".

Con un'opportuna modifica della cavità risonante, si ottiene la possibilità di restringere enormemente la larghezza di banda e poter selezionare a piacimento una specifica lunghezza-

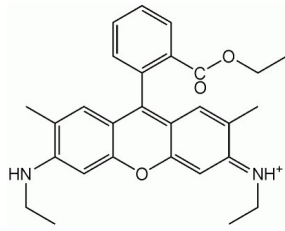


Fig. 3.7: Struttura di uno dei pigmenti piú utilizzati come materiale laser: la Rodamina 6G.

za d'onda. Uno dei metodi piú comunemente utilizzati consiste nel sostituire lo specchio concavo, con una particolare struttura chiamata reticolo, esso ha la proprietá di riflettere lunghezze d'onda diverse a seconda dell'angolo con cui il fascio incide su di esso; ruotando il reticolo si puó quindi sintonizzare l'azione su una ristrettissima larghezza di banda. Per poter ottenere un'unica lunghezza d'onda é comunque necessario inserire dei filtri nella cavità. Particolare attenzione andrà quindi posta alla movimentazione di tutti gli elementi aggiuntivi (etalon, cristallo) che dovranno anch'essi ruotare per mantenersi sintonizzati; ogni componente infatti ha bisogno che il raggio incida con un angolo che dipende dalla lunghezza d'onda desiderata.

### 3.5.1 Reticolo[5]

Il reticolo a riflessione é costituito da uno specchio riflettente sul quale vengono ricavati tanti piccoli solchi che non consentono la riflessione della radiazione, si creano cioé tanti diversi piani riflettenti inclinati di un certo angolo (Fig. 3.8); inviando luce bianca sul reticolo, i raggi di luce riflessa dalle singole facce percorrono ognuno rispetto al successivo, una maggior distanza data dalla somma algebrica delle parti di traiettoria in grassetto comprese tra i piedi delle perpendicolari tratteggiate e i punti d'incidenza. Ponendo la distanza  $d$  tra due facce inclinate consecutive confrontabile con  $\lambda$ , la differenza di cammino ottico sará dello stesso ordine di grandezza di  $\lambda$  e potrà dare dei ritardi di fase tra 0 e  $2\pi$  o multipli secondo la formula

$$m\lambda = d(\sin \alpha \pm \sin \beta)^1 \quad (3.5.1)$$

Dove  $\alpha$  é l'angolo d'incidenza del fascio rispetto alla normale del reticolo,  $d$  é la distanza tra due solchi consecutivi (passo del reticolo),  $\lambda$  é la lunghezza d'onda della luce diffratta,  $m$  l'ordine di diffrazione,  $\beta$  l'angolo di diffrazione.

Tra tutte le lunghezze d'onda della luce bianca ce ne sará una per la quale, a quel particolare angolo d'incidenza, lo sfasamento tra due raggi consecutivi é esattamente  $2\pi$  (cioé la differenza di cammino é pari esattamente a una o piú intere lunghezze d'onda); solo per quella  $\lambda$  si avrá quindi una fortissima interferenza costruttiva e sará come aver riflesso solo quella lunghezza d'onda.

<sup>1</sup>Il segno + é da considerarsi se  $\alpha$  e  $\beta$  si trovano dalla stessa parte rispetto alla normale, altrimenti si deve utilizzare il segno -.

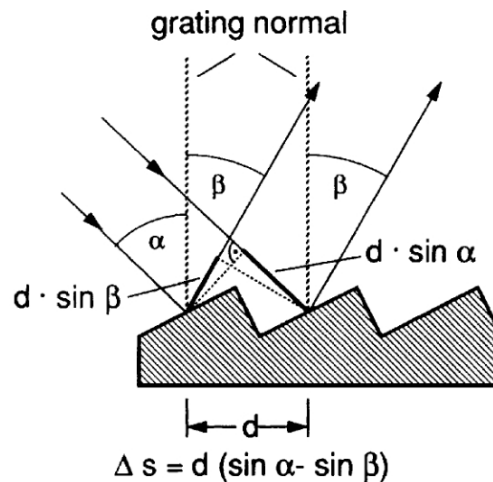


Fig. 3.8: Ingrandimento di un reticolo di diffrazione.

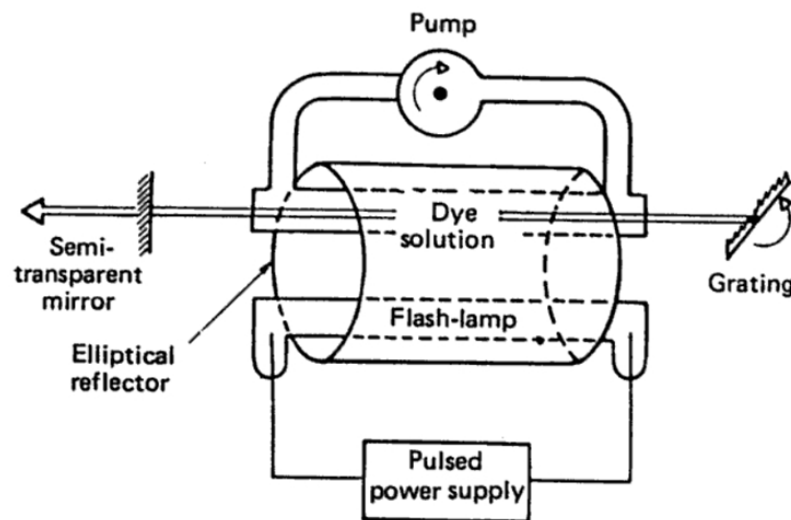


Fig. 3.9: Struttura di un Laser a dye.

### 3.5.2 Filtro Etalon

Nel caso in cui lo spettro in uscita non fosse sufficientemente stretto, molti laser consentono l'aggiunta di un filtro etalon per ridurre drasticamente le lunghezze d'onda presenti nel fascio.

L'etalon é un particolare tipo di filtro ottico che, grazie alla sua struttura, riesce ad impedire il passaggio a tutte le lunghezze d'onda diverse da quella desiderata; esso utilizza il principio di funzionamento dell'interferometro di Fabry-Perrot.

Due specchi piani sono posti parallelamente tra loro e lo spazio racchiuso costituisce la cavità; in questo caso, a differenza della cavità laser, entrambi gli specchi sono semiriflettenti e quindi ad ogni riflessione una parte della radiazione passa e la restante (maggioranza) viene riflessa. Come descritto in precedenza, anche in questo caso si possono ottenere i due casi d'interferenza costruttiva o distruttiva; si ottiene quindi che solo alla giusta lunghezza d'onda i fasci di andata e ritorno si rinforzeranno a vicenda, rendendo praticamente nulla la riflessione sul secondo specchio.

Come abbiamo già visto con la formula (3.4.2), essendo l'etalon molto sottile (soprattutto paragonato con la lunghezza della cavità risonante del laser), i modi risonanti dell'etalon sono molto più distanti in frequenza rispetto a quelli della cavità laser rendendo possibile

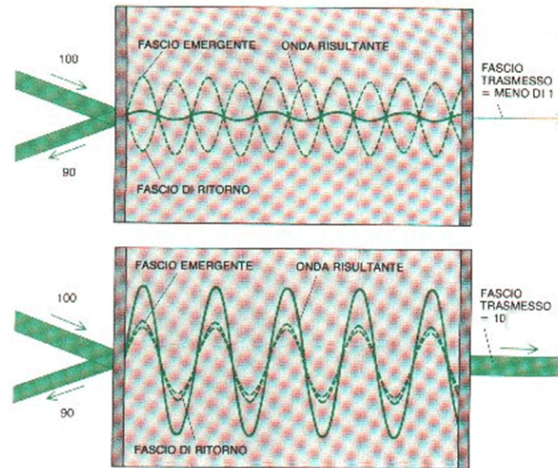


Fig. 3.10: Funzionamento filtro di Fabry-Perrot.

una selezione molto accurata; inclinando leggermente il filtro rispetto al fascio laser si può inoltre variare leggermente la lunghezza di percorso e quindi le frequenze di risonanza ottica. In Fig. 3.11 è riportato il risultato dell'aggiunta di un filtro etalon nella cavità laser.

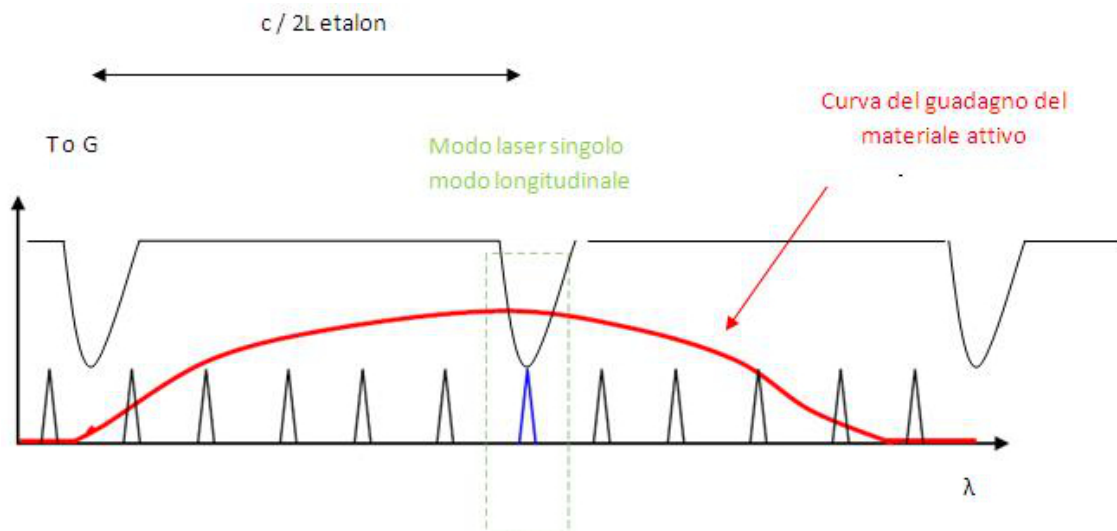


Fig. 3.11: Amplificazione del singolo modo longitudinale grazie all'inserimento di un etalon.

### 3.5.3 Raddoppio in frequenza

Un'ulteriore modifica alla struttura base di Fig. 3.3 può essere l'aggiunta di un cristallo in prossimità dell'uscita del laser.

Quando la luce si propaga in un mezzo trasparente, il suo campo elettrico induce nel mezzo una polarizzazione elettrica, che al primo ordine sarà lineare con il campo applicato; se però il mezzo è otticamente non lineare, la risposta può diventare non lineare. La non linearità al secondo ordine provoca la generazione di una seconda armonica quando il fascio laser attraversa il mezzo, si ottiene così un fascio di luce a frequenza doppia rispetto a quello entrante (e quindi con una lunghezza d'onda dimezzata).

Solitamente, per compiere questa operazione, i materiali ottici non lineari utilizzati sono dei cristalli di LBO, KDP oppure  $\text{LiNbO}_3$ .





Fig. 3.12: Effetto di un cristallo non lineare su un raggio laser.

## 3.6 Motori passo-passo[6]

Come accennato nel paragrafo precedente, i laser sintonizzabili hanno necessità di movimentare, con estrema precisione, alcuni componenti al loro interno; per operare questi spostamenti in maniera accurata vengono solitamente impiegati dei motori elettrici di tipo passo-passo i quali, date le loro caratteristiche, sono particolarmente indicati per questo tipo di applicazioni.

I principali vantaggi rispetto ad altri tipi di azionamenti sono:

- Possibilità di compiere accurati posizionamenti in catena aperta, cioè senza aver bisogno di sensori di posizione o di velocità.
- Sono molto stabili nella posizione a rotore bloccato.
- L'errore di posizione non è cumulativo.
- La velocità di rotazione può essere molto bassa anche senza l'uso di riduttori meccanici.
- Particolarmente semplici ed adatti al controllo digitale.

### 3.6.1 Principio di funzionamento dei motori passo-passo

I motori a passo attuano una conversione elettromeccanica dell'energia che, a seconda della tipologia costruttiva, sfrutta il principio dei sistemi a riluttanza variabile, quello dei sistemi elettrodinamici o una combinazione di entrambi (si parla in quest'ultimo caso di motori a passo ibridi).

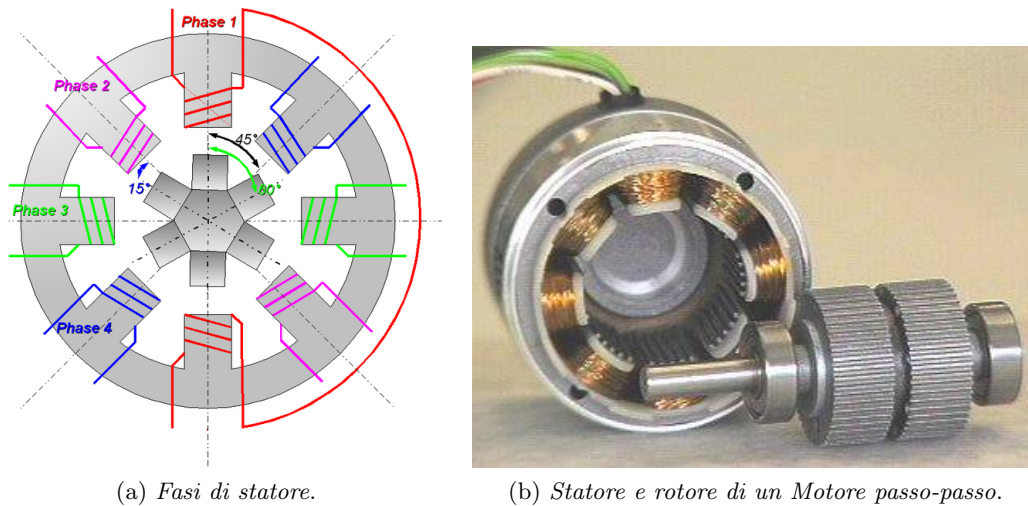
Nei motori ibridi lo statore è realizzato in materiale ferromagnetico e presenta una marcata anisotropia radiale (vedi Fig. 3.13), attorno a queste scanalature si avvolgono i fili conduttori che vanno a formare le fasi di statore (ogni fase è composta da due avvolgimenti disposti in posizioni diametralmente opposte). Ogni scanalatura può presentare a sua volta ulteriori piccole scanalature (denti).

Il rotore ha invece un nucleo costituito da un magnete permanente cilindrico che produce un campo magnetico assiale unipolare (Fig. 3.14). Su ciascun polo è allocata una struttura dentata ed i denti delle due sezioni sono disallineati tra loro di mezzo passo di dentatura. Alimentando opportunamente le fasi di statore, la corrente che circola in esse produce un campo magnetico ed i denti del rotore vengono quindi attratti o respinti dai denti di statore a seconda della loro posizione relativa iniziale (Fig. 3.14).

Alternando in modo circolare l'alimentazione delle fasi si ottiene quindi la rotazione continua del rotore; mantenendo invece l'alimentazione su una sola fase, si mantiene una determinata posizione.

La rotazione compiuta dal rotore ad ogni passo è detta angolo di passo  $\alpha_p$  ed il corrispondente numero di passi/giro è dato da

$$Np = \frac{2\pi}{\alpha_p} \quad (3.6.1)$$



(a) Fasi di statore.

(b) Statore e rotore di un Motore passo-passo.

Fig. 3.13: Motore passo passo.

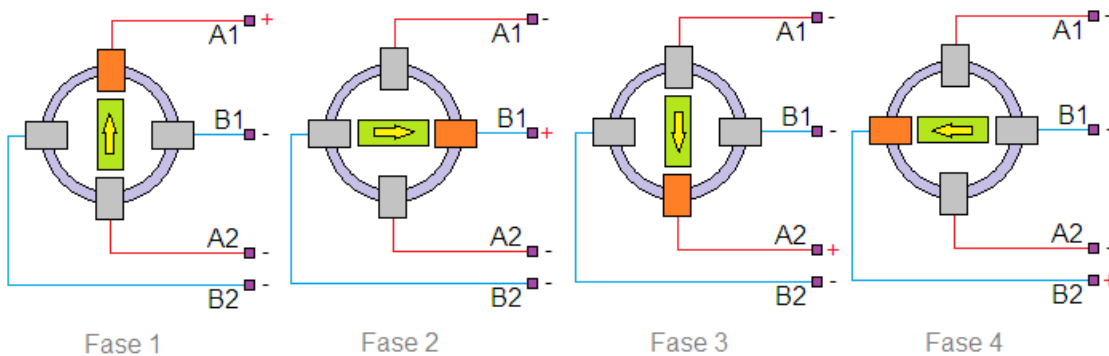


Fig. 3.14: Alimentazione circolare delle fasi di statore.

$Np$  é quindi un importante parametro nei motori a passo, in quanto indicativo della risoluzione angolare.

Oltre ad alimentare una fase per volta ed ottenere cosí uno spostamento del rotore pari ad un passo (full step), é possibile alimentare due fasi adiacenti in modo da posizionare il dente di rotore a metà strada, ottenendo cosí uno spostamento di mezzo passo (half step). Con lo stesso principio, si può ottenere una serie ampia a piacere di posizioni intermedie tra due step, inviando due correnti con diverso modulo nelle due fasi adiacenti (microstep). Visto il suo principio di funzionamento, il motore a passo si presta particolarmente per essere controllato in modo digitale, l'alimentazione delle fasi con i tempi ed i moduli corretti é quindi affidata ad un pc/plc/microcontrollore.

### 3.7 Monocromatore

Il monocromatore é un dispositivo in grado di scomporre un fascio di luce policromatica in piú fasci di luce monocromatica e restituire in uscita uno solo di essi. La scomposizione del fascio é effettuata tramite un prisma o un reticolo a riflessione. Nei monocromatori con reticolo si mantiene fissa la direzione della luce in ingresso e si ruota il reticolo per avere in uscita la lunghezza d'onda desiderata.

Costruttivamente il monocromatore ha una fenditura d'ingresso su cui viene inviata la luce da analizzare e che funge da sorgente puntiforme del sistema. Essa illumina uno specchio concavo che trasforma il fascio divergente in un fascio parallelo che illumina uni-

formemente il reticolo; la luce riflessa segue un percorso ottico inverso ed i raggi vengono fatti convergere in un'altra piccola fenditura che funge da punto d'uscita. Naturalmente un monocromatore é tanto piú efficace quanto piú riesce a separare due lunghezze d'onda vicine, questo é dato dalla grandezza degli specchi concavi, dalla lunghezza del cammino ottico che si crea e dal numero di scanalature del reticolo.

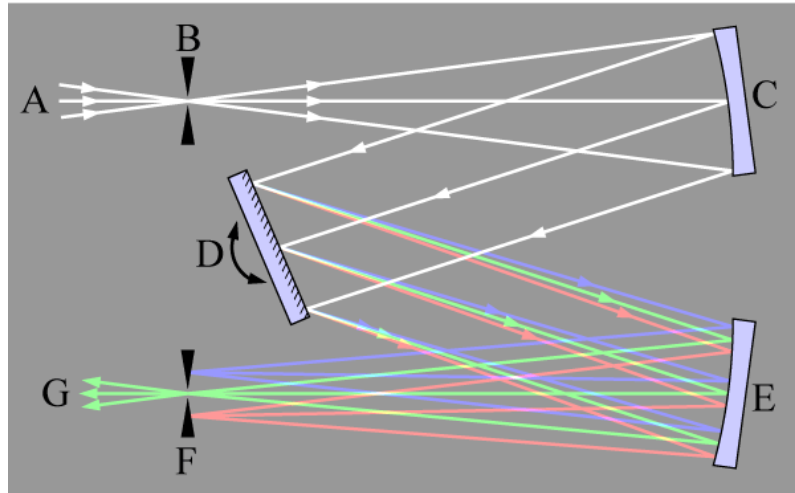


Fig. 3.15: Funzionamento del monocromatore.

### 3.8 Conclusioni

La fotoionizzazione Laser risonante é una tecnica indispensabile se si ha la necessitá di ottenere una forte selezione giá in fase di ionizzazione. Per poterla effettuare é necessario disporre di fasci di luce laser (una luce monocromatica, unidirezionale e coerente) in grado di eccitare gli elettroni di un preciso elemento.

Per ottenere un fascio laser é necessario utilizzare una cavità risonante al fine di portare al massimo il fenomeno dell'emissione stimolata dei fotoni. Dispositivi molto particolari sono i Laser a dye, in grado di cambiare la lunghezza d'onda del fascio prodotto; questo, nel nostro caso particolare, é reso possibile da un reticolo di diffrazione posto all'interno della cavità. In questi dispositivi, in aggiunta alla cavità laser, si possono utilizzare elementi aggiuntivi: un filtro Etalon per filtrare ulteriormente lo spettro in uscita o un cristallo non lineare per dimezzare la lunghezza d'onda.



## Progetto Scheda Elettronica

---

### 4.1 Introduzione

Il presente capitolo si occupa della progettazione della scheda elettronica (PCB: Printed Circuit Board); progettata e realizzata per il controllo di alcuni dispositivi all'interno del laboratorio Laser del progetto SPES. Il primo passo é quello di verificare l'hardware presente e decidere, sulla base delle funzioni che dovrà garantire la scheda, come strutturare la PCB. Il capitolo prende in esame separatamente le diverse funzioni e propone un circuito che possa svolgerle al meglio. Dopo aver esposto l'intero progetto si passa ad illustrare il procedimento per la realizzazione della PCB (stampaggio, saldatura ecc.), per terminare infine con l'assemblaggio nel contenitore dedicato.

### 4.2 Richieste progettuali hardware

Lo scopo di di questa tesi é progettare e realizzare un sistema di controllo per il laser sintonizzabile FL2002 e per altri dispositivi presenti nel laboratorio laser del progetto SPES.

Il laser FL2002 é un dispositivo realizzato negli anni '80 ma solo da poco di proprietà dell'INFN, essendo un laser sintonizzabile é stato costruito con tutto il necessario per poter essere programmato ed effettuare scansioni ed analisi a diverse lunghezze d'onda; al momento però solo i motori elettrici sono fisicamente disponibili, il sistema di controllo é infatti inutilizzabile. Per poter essere utilizzato occorre quindi dotarlo di un nuovo sistema di controllo realizzato ad hoc. Nello stesso laboratorio e per i medesimi esperimenti vengono utilizzati altri due dispositivi: il laser TDL50 e il monocromatore JOBIN YVON, si desidera quindi poter controllare tutti 3 i dispositivi con lo stesso strumento.

Per ottenere tutto ciò é necessario realizzare una scheda elettronica che sia in grado gestire le informazioni necessarie e di comandare in modo adeguato i motori nei diversi dispositivi.

#### 4.2.1 Dispositivi hardware presenti

Per capire come strutturare la scheda, bisogna innanzitutto sapere quali comandi dovrà fornire e quali saranno i segnali di controllo e i sensori a disposizione; per capire questo si é fatto uso, dove presenti, dei manuali, le cui pagine d'interesse sono riportate in appendice [A](#). Da queste e dall'osservazione degli strumenti ricaviamo le seguenti informazioni:

#### **FL2002**

Questo Laser offre la possibilità di eseguire scansioni utilizzando un reticolo di diffrazione (Grating), per stringere la riga spettrale di emissione si può inoltre inserire un filtro etalon mentre un cristallo di seconda armonica (SHG, Second Harmonic Generation) permette la duplicazione della fondamentale; questi 3 componenti, come già visto nel [Cap. 3](#), devono essere movimentati tramite motori elettrici di tipo passo passo. Per adattarsi al sistema esistente il collegamento tra la PCB e il laser deve avvenire tramite un connettore a 60 pin il quale si suddivide poi in 3 connettori come riportato in [tab 4.1](#)

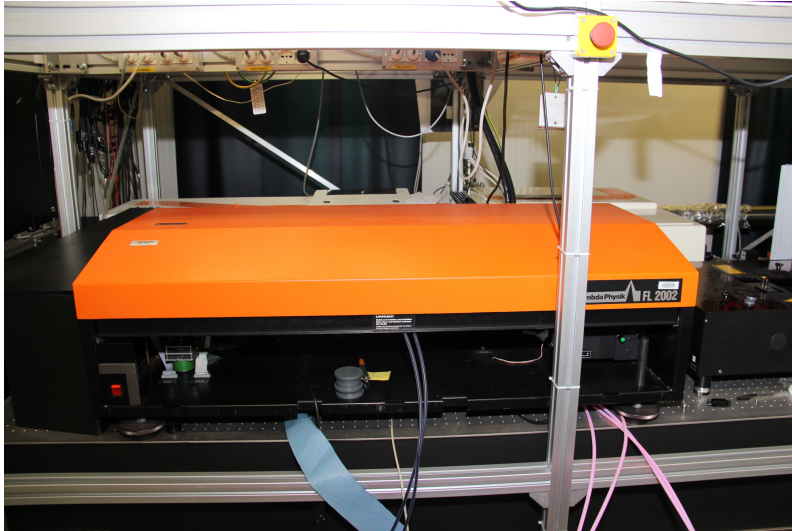


Fig. 4.1: Laser FL2002 presente in laboratorio.

Pin	Motore
1-14	Grating
15-28	Etalon
43-60	Cristallo

Tab. 4.1: Divisione connettori FL2002.

Per ognuno di questi tre gruppi si ha:

- un motore da pilotare
- due finecorsa che avvertono quando la movimentazione ha raggiunto una delle due estremità di movimento
- un led che avvisa l'evento finecorsa
- una levetta per il controllo manuale

Inoltre, solo per il cristallo, sono presenti alcuni pin collegati ad un circuito interno che mantiene la temperatura del cristallo ad un livello costante; la funzione del cristallo è infatti fortemente dipendente dalla temperatura. Bisognerà fornire quindi la tensione richiesta dal circuito (vedere Fig. A.3)

## TDL50

Anche questo è un laser sintonizzabile ed è già dotato di un proprio sistema di controllo, il quale offre la possibilità d'impostare e controllare questo dispositivo da un pc, comunicando tramite una porta RS-232 (ed il protocollo standard ad essa associato); per questo dispositivo è quindi necessario predisporre sulla PCB una porta per la comunicazione seriale dedicata.



Fig. 4.2: Laser TDL50 presente in laboratorio.

### Monocromatore

Questo dispositivo ha al suo interno un solo motore passo-passo che viene utilizzato per cambiare angolo d'inclinazione del reticolo, i cavi di alimentazione di questo motore sono predisposti ad un collegamento mediante una porta DB9. Per la gestione del moto sono presenti anche in questo caso due finecorsa.

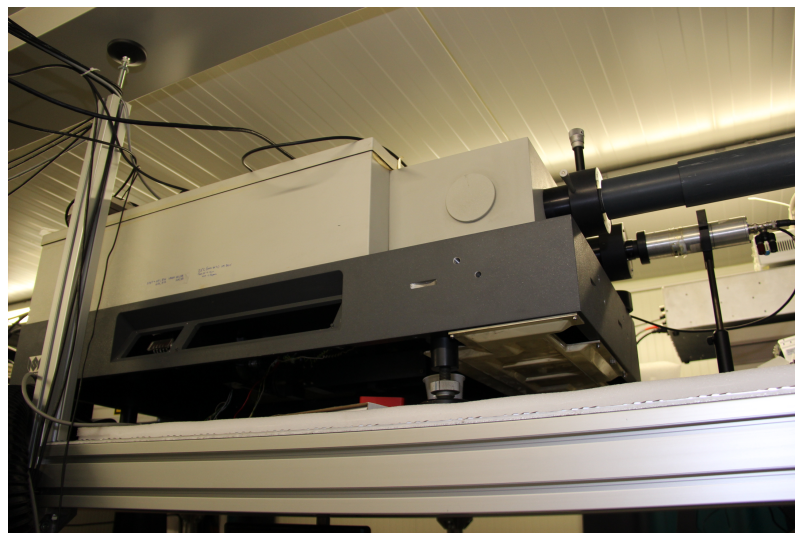


Fig. 4.3: Monocromatore presente in laboratorio.

#### 4.2.2 Componenti hardware necessari

Oltre a questi dati ricavati dall'hardware presente, bisogna tener conto che la scheda deve poter essere gestita da un operatore tramite un pannello frontale oppure tramite un pc collegato ad essa; bisogna quindi prevedere l'inserimento di:

- pulsanti per poter comunicare alla scheda le impostazioni desiderate

- un display lcd per visualizzare le informazioni date dal programma
- una porta seriale RS-232 per la comunicazione con un pc
- led di avvertimento

### 4.3 Software utilizzato per la progettazione della scheda elettronica

Per realizzare il progetto si é fatto uso di alcuni software che rendono il lavoro piú intuitivo, semplice e soprattutto aiutano a prevenire alcuni errori in cui ci si potrebbe imbattere.

#### 4.3.1 Orcad Capture

Questo software consente di realizzare uno schema elettrico del circuito che si desidera creare. Il primo passo é quello di posizionare su un “foglio di lavoro” i componenti elettronici necessari, questi sono visualizzati in modo schematico, ponendo in particolare evidenza i pin di collegamento; le librerie disponibili per questo software comprendono la quasi totalitá dei componenti piú comunemente utilizzati, in caso di necessitá é comunque possibile utilizzare l’editor apposito e creare i componenti necessari.

Ad ogni componente bisogna poi assegnare la rispettiva piedinatura fisica (footprint), data dalle misure reali del componente e dalle distanze tra i pin; i footprint si trovano sulle librerie del software “Orcad Layout” che anche in questo caso comprendono quasi tutte le possibilitá ed é eventualmente disponibile un editor per i casi particolari.

Il passo successivo é quello di collegare i vari pin dei componenti mediante delle linee grafiche. Importante ricordare che in questa fase la cosa importante é esclusivamente la numerazione dei pin di ogni componente ed i collegamenti tra i vari pin, non hanno alcuna rilevanza invece le posizioni, le misure e le proporzioni di componenti, pin e vie di collegamento (le quali verranno gestite e impostate successivamente con “Orcad Layout”).

#### 4.3.2 Orcad Layout

Questo software, prendendo le informazioni dallo schema elettrico precedentemente creato, permette di disegnare la scheda vera e propria.

All’avvio, il programma visualizza i footprint di tutti i componenti presenti nel progetto, dopo aver definito il bordo della scheda si puó quindi passare a posizionare i componenti, tenendo conto della loro funzione e dei collegamenti di cui necessitano. Il programma in questo caso risulta molto utile perché disegna delle sottili linee rette tra i pin che ricordano all’utente come bisogna collegare i vari pin. Una volta posizionati i componenti si passa a definire le larghezze delle piste ed a trovare i percorsi piú adatti e convenienti per ogni collegamento.

### 4.4 Struttura della scheda elettronica

Dopo aver identificato a grandi linee quali dovranno essere i componenti presenti e le funzionalitá da fornire, si passa ora ad esaminare come poter realizzare ogni operazione. Lo schema elettrico deciso viene quindi disegnato con il software “Orcad Capture”.

É bene ricordare che la quasi totalitá dei circuiti integrati per PCB richiede, come alimentazione, una tensione di circa 5V; di conseguenza, essendo circuiti digitali, comunicano tra loro e con il mondo esterno tramite livelli logici: alto (o 1) che sará 5V e basso (o 0) che sará a 0V. Per semplicitá di trattazione si é scelto di suddividere concettualmente la scheda in macro-aree.



#### 4.4.1 Microcontrollore PIC ed interfacciamento

##### PIC

Il cuore di tutta la PCB é un microcontrollore in grado di essere programmato e di gestire un certo numero di ingressi ed uscite.

Un microcontrollore é un dispositivo elettronico che integra in un unico chip: un processore, una memoria permanente, una memoria volatile e dei canali (PIN) di I/O; questa struttura li rende particolarmente adatti per lo sviluppo di applicazioni specifiche di controllo digitale dove il basso costo, le ridotte dimensioni e i bassi consumi, sono spesso condizioni importantissime.

Il PIC18F452 (prodotto dalla Microchip), già utilizzato in altri progetti ai laboratori di Legnaro, é stato scelto per la sua già testata e confermata affidabilità, il costo contenuto ed un adeguato numero di pin dedicati ad ingressi e periferiche.

Questo componente possiede diverse funzionalità, tra le quali, che interessano maggiormente a questo progetto, ci sono:

- memoria ram: 32 Kbytes per il programma e 1,5 Kbytes per allocazione dinamica delle variabili
- 33 pin di I/O
- predisposizione alla comunicazione con porta seriale RS232
- oscillatore interno predisposto
- 3 pin per la ricezione di interrupt esterni
- ADC
- timer

Per poter utilizzare il PIC bisogna fornirgli, oltre alla tensione di alimentazione ( $V_{cc}=5V$ ) ed alla massa, un segnale di clock con il quale il suo processore possa operare; al suo interno il PIC possiede già un oscillatore predisposto per poter lavorare abbinato ad un quarzo esterno che lo stabilizzi in modo preciso ad una specifica frequenza (FOSC). Per questo progetto si é scelto un quarzo da 8Mhz, esso ha due estremitá che vanno collegate da una parte al PIC (OSC1 ed OSC2) e dall'altra a massa tramite due condensatori da 22pF.

##### LCD

Per poter interagire con il sistema, nel caso di un controllo da pannello, é necessario predisporre un piccolo display LCD; non essendo necessario riportare disegni o grafici si é scelto un LCD 16x2 che dá la possibilità di scrivere due righe da sedici caratteri alfanumerici. Il display é basato sul chip HD44780U, la tipologia piú comunemente utilizzata e dotata di librerie pronte per la programmazione.

L'alimentazione di tale componente viene effettuata collegando l'apposito pin 15 a  $V_{cc}$  tramite una resistenza da 10  $\Omega$ , il contrasto é impostato invece collegando il pin 3 ad un potenziometro da 10  $k\Omega$  avente le soglie da  $V_{cc}$  a GND. Il controllo e la gestione dei caratteri viene invece gestito dai pin 4,6,11,12,13,14 che devono quindi essere collegati al PIC (i pin scelti sono quelli della porta C da 0 a 5).

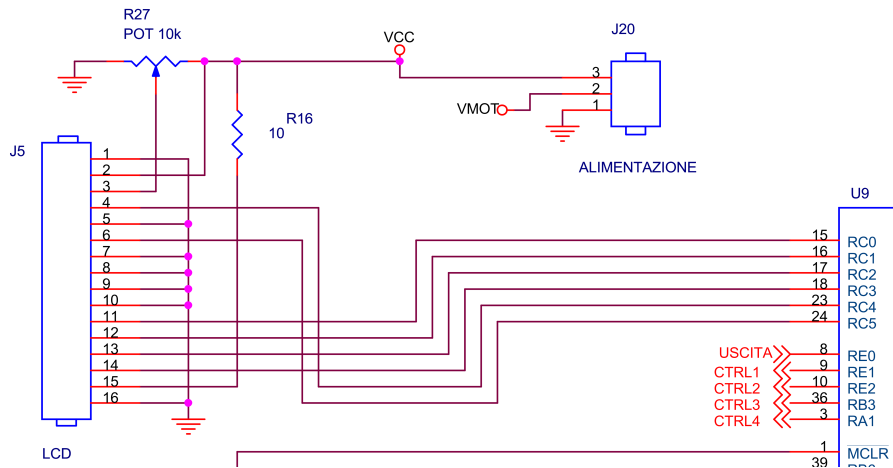


Fig. 4.4: Schema di collegamento del display LCD

### Porta seriale

La comunicazione seriale é una modalit  di comunicazione tra dispositivi digitali, essa consiste nell'invio di una sequenza di bit dal mittente al ricevente tramite un unico cavo elettrico, pertanto l'ordine della sequenza viene mantenuto. I dispositivi che comunicano in questo modo devono per forza utilizzare lo stesso procedimento di codifica e decodifica del segnale, per questo motivo é stato creato uno standard riconosciuto a livello mondiale e utilizzato in moltissimi campi di applicazione, esso prende il nome di EIA RS-232; questo standard definisce dei protocolli meccanici, elettrici ed informatici che rendono possibile lo scambio d'informazioni tra dispositivi. Il connettore che pi  comunemente viene utilizzato (specialmente per le comunicazioni con i PC) é quello a 9 pin (DB9) dove per , per un utilizzo di semplice invio e ricezione, i pin utilizzati sono solo 3: uno per la massa, uno per l'invio (TX) ed uno per la ricezione (RX). I segnali utilizzati nel protocollo RS-232 sono

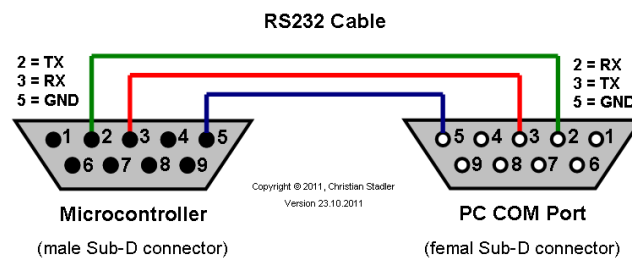


Fig. 4.5: Collegamento maschio-femmina nella porta RS-232.

approssimativamente  $\pm 7.5$  V, il pic é per  in grado di fornire segnali tra 0V e 5V, per questo motivo bisogna operare l'opportuna conversione; il MAX232 é il chip generalmente utilizzato per questo scopo e quindi, anche per questo progetto, ci si é affidati a questo componente. Per poter operare queste conversioni in modo corretto ad esso vanno inoltre collegati dei condensatori e delle resistenze. Il chip MAX232 permette di controllare fino a due porte seriali, per questo progetto baster  quindi un integrato per interfacciarsi sia con il pc che con il laser TDL50. Il Pic18F452 possiede due pin dedicati all'utilizzo della comunicazione seriale (C6 e C7) essi andranno quindi collegati rispettivamente ai pin di trasmissione (T1in) e ricezione (R1out) il primo canale del MAX232; avendo bisogno di un secondo canale si sono scelti altri due pin del pic (B4 e B5) che andranno collegati ai pin del secondo canale del MAX232 (T2in e R2out). Il risultato finale é quindi riportato in Fig. 4.6.

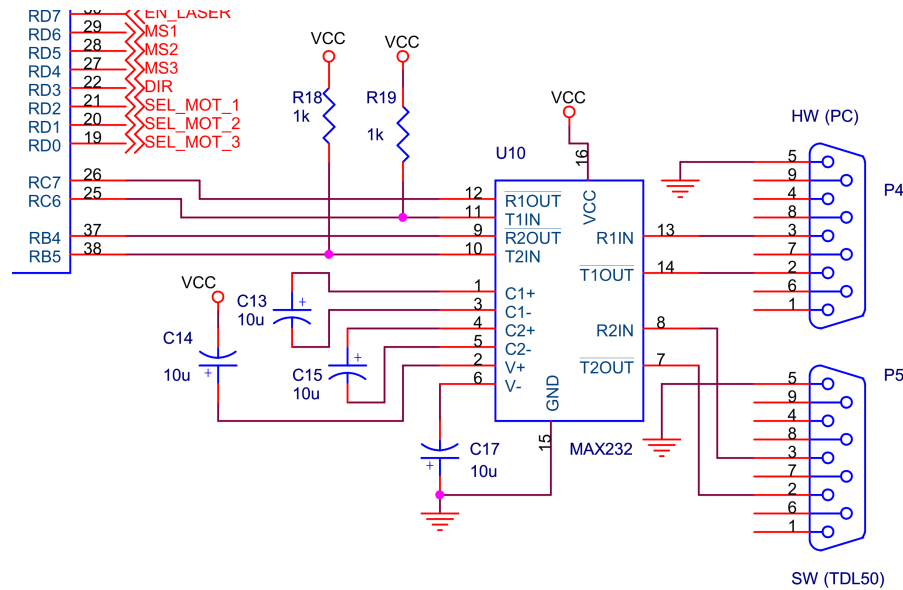


Fig. 4.6: Schema di collegamento del chip MAX232.

### Potenzimetro

Per avere la possibilità di variare la velocità dei motori durante l'esecuzione del programma, si è pensato di utilizzare un potenziometro di tipo rotativo da porre sul pannello frontale per il controllo manuale; le due estremità della resistenza vanno chiaramente collegate una a Vcc e l'altra a massa, il pin centrale invece, essendo quello variabile, dovrà essere letto dal PIC ed è quindi stato collegato ad esso tramite il pin RA0 (questo pin è infatti collegato internamente con il circuito dedicato alla conversione analogico/digitale del PIC). Per filtrare eventuali rumori è stato collegato un condensatore (100nF) per operare un filtraggio di tipo passa-basso.

### Ventole

Il datasheet dei driver per motori specificano chiaramente che, a causa dell'elevata potenza gestita, è consigliato l'utilizzo di dissipatori e/o ventole per il raffreddamento; per questo motivo si è deciso di considerare l'inserimento di due ventole all'interno del contenitore dove alloggerà la PCB. Si è scelto di utilizzare ventole alimentate a 5V per utilizzare direttamente la Vcc già fornita per tutti gli altri componenti.

#### 4.4.2 Driver per controllo motori

Come già accennato nel capitolo precedente, i motori utilizzati nella costruzione di questi dispositivi Laser sono motori passo passo, il manuale del FL2002 ne dà conferma e osservando le Fig. A.2 e A.3 Vediamo che sono di tipologia bipolare (4 fili per pilotare le fasi di statore).

Per controllare questo tipo di motori ci si affida generalmente a dei circuiti integrati appositamente studiati per fornire alle fasi di statore la giusta tensione con un'opportuna sequenza; ciò è reso possibile in particolar modo dalla presenza al loro interno di transistor in configurazione di ponte ad H. Questi integrati, per poter funzionare, hanno bisogno di un certo numero di componenti aggiuntivi (resistenze, condensatori); per questo motivo esistono in commercio diverse soluzioni preassemblate che consentono di acquistare direttamente un piccolo circuito già pronto e predisposto per essere inserito in un progetto di PCB per svolgere la funzione di Motor Driver.

A seconda della tensione di alimentazione dei motori, della corrente necessaria e della risoluzione degli step, ci sono moltissimi Driver disponibili sul mercato; per le nostre necessità é stato scelto l'A4988 Stepper motor della Pololu corporation, il quale fornisce le seguenti caratteristiche:

- segnali utilizzati dal chip: 3V-5.5V
- tensione operativa: 8V-35V
- Corrente di Output per bobina: 1,2A continua 2A massima
- 5 differenti risoluzioni per gli step
- Protezione di sottotensione, di crossover-corrente e massima temperatura
- Circuito di protezione per inversione di polarità
- potenziometro per la regolazione della corrente sul motore
- PCB multistrato a 4 livelli per una maggiore capacità dissipativa

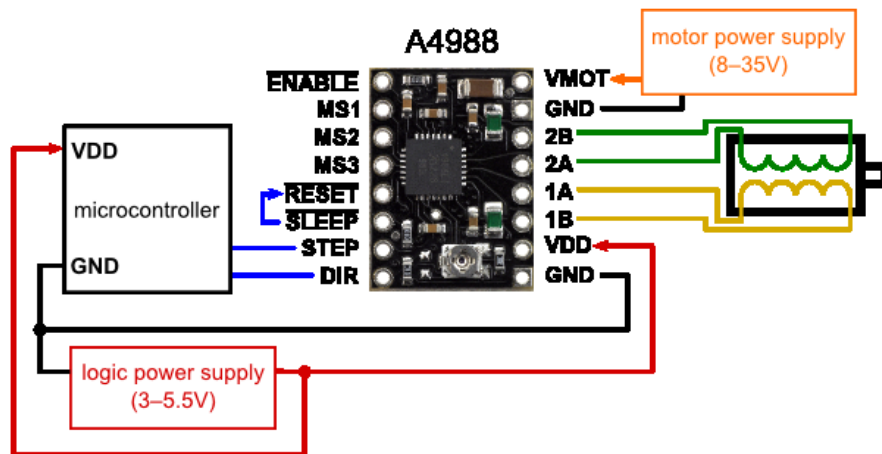


Fig. 4.7: Pin del chip A4988 e principio di collegamento.

Come mostrato in Fig. 4.7 il circuito possiede 16 pin le cui funzioni sono ora descritte:

- $\overline{ENABLE}$ : a livello logico basso abilita le uscite, a livello logico alto le disabilita, in ogni caso gli altri parametri rimangono impostati.
- MS1,MS2,MS3: definiscono la risoluzione degli step secondo la Tab.4.2

MS1	MS2	MS3	Microstep Resoution
L	L	L	Full Step
H	L	L	Half Step
L	H	L	Quarter Step
H	H	L	Eighth Step
H	H	H	Sixteenth Step

Tab. 4.2: Definizione Microstep.

- $\overline{RESET}$ : a livello logico basso resetta le impostazioni del chip e finché resta basso, le uscite sono disabilitate.
- $\overline{SLEEP}$ : a livello logico basso l'integrato è quasi del tutto disabilitato, ciò permette di risparmiare energia.
- STEP: una transizione da livello logico basso ad alto fa compiere uno step al motore (se le uscite sono abilitate).
- DIR: il livello logico alto o basso determina la direzione di rotazione del motore.
- VMOT: tensione di alimentazione del motore.
- GND: collegamento a massa.
- 1B,1A: uscite del circuito, tensioni della fase 1 del motore.
- 2B,2A: uscite del circuito, tensioni della fase 2 del motore.
- VDD: tensione di alimentazione del circuito.

Questi driver sono in grado di controllare un singolo motore, per questo motivo nella PCB servirà un driver per ogni motore: 3 per il FL2002, 1 per il Monocromatore, 2 di Riserva se in futuro ci sarà necessità d'inserire un nuovo dispositivo da controllare (Spare).

Sappiamo già che la PCB non dovrà controllare più dispositivi contemporaneamente e nel caso del FL2002 i motori andranno azionati uno alla volta, per questo motivo si è scelto di collegare assieme tutti i pin MS1, MS2, MS3, RESET e DIR a 5 rispettivi pin del PIC (rispettivamente D6,D5,D4, A2,D3); ottimizzando in questo modo il numero di pin utilizzati.

La decisione di far compiere un passo ad uno dei motori viene imposta portando a valore logico 1 il pin STEP del Driver ad esso collegato; anche in questo caso non è possibile utilizzare un diverso pin del PIC per ogni STEP dei driver, per ottimizzare il numero si è deciso in questo caso di utilizzare un demultiplexer. Questo è un circuito integrato avente  $k$  ingressi e  $2^k$  uscite ed in grado di selezionare, per ogni combinazione d'ingressi, una sola delle uscite; avendo bisogno di 6 segnali è stato scelto un demultiplexer con 3 pin d'indirizzo ed 8 uscite, il 74HCT238M96. In Fig. 4.8 è riportata una tabella del suo funzionamento.

INPUTS						OUTPUTS							
ENABLE			ADDRESS										
E3	$\overline{E2}$	$\overline{E1}$	A2	A1	$\overline{A0}$	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	L	L	L	L	L	L	L	L
L	X	X	X	X	X	L	L	L	L	L	L	L	L
X	H	X	X	X	X	L	L	L	L	L	L	L	L
H	L	L	L	L	L	H	L	L	L	L	L	L	L
H	L	L	L	L	H	L	H	L	L	L	L	L	L
H	L	L	L	H	L	L	L	H	L	L	L	L	L
H	L	L	L	H	H	L	L	L	H	L	L	L	L
H	L	L	H	L	L	L	L	L	L	H	L	L	L
H	L	L	H	H	L	L	L	L	L	L	L	H	L
H	L	L	H	H	H	L	L	L	L	L	L	L	H

NOTE: H = High Voltage Level, L = Low Voltage Level, X = Don't Care

Fig. 4.8: Tavola della verità del chip 74HCT238M96.

Non avendo bisogno di tutte 8 le uscite, si è deciso di mantenere costantemente a livello

alto il pin 6 e costantemente al livello basso i pin 4 e 5; in questo modo agendo sui pin 1, 2, 3 si ottiene un'uscita tra Q1,Q2,Q3,Q4,Q5,Q6. Abbiamo così la possibilità di scegliere uno tra i 6 driver necessari, agendo solo su 3 pin del PIC.

L'abilitazione al funzionamento di questi Driver è invece stata affidata ad un selettore che, a seconda della posizione inserita, abilita un gruppo di driver: FL, Monocromatore o Spare, collegando i relativi pin *ENABLE* a massa.

Per far questo si è scelto di utilizzare un selettore a due canali, questo componente è costituito da due selettori che lavorano in modo parallelo perché collegati al medesimo switch fisico; in questo modo, collegandolo opportunamente, saremo in grado di abilitare il dispositivo desiderato ed allo stesso tempo poter sapere qual è il dispositivo abilitato in un dato momento (questo per poter disporre di un controllo di sicurezza nel caso di distrazione dell'operatore).

Il modello scelto è il commutatore rotativo CK1030 a 6 vie e 2 poli riportato in Fig. 4.9.

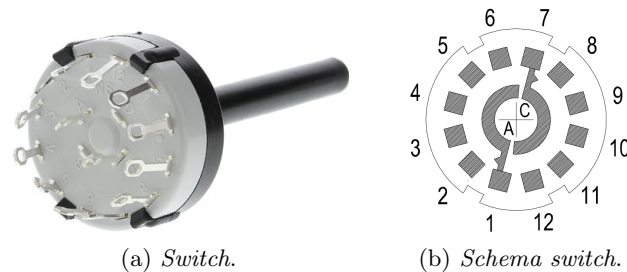


Fig. 4.9: Commutatore rotativo CK1030 a 6 vie e 2 poli.

La porzione di circuito riguardante il selettore è riportata in Fig. 4.10

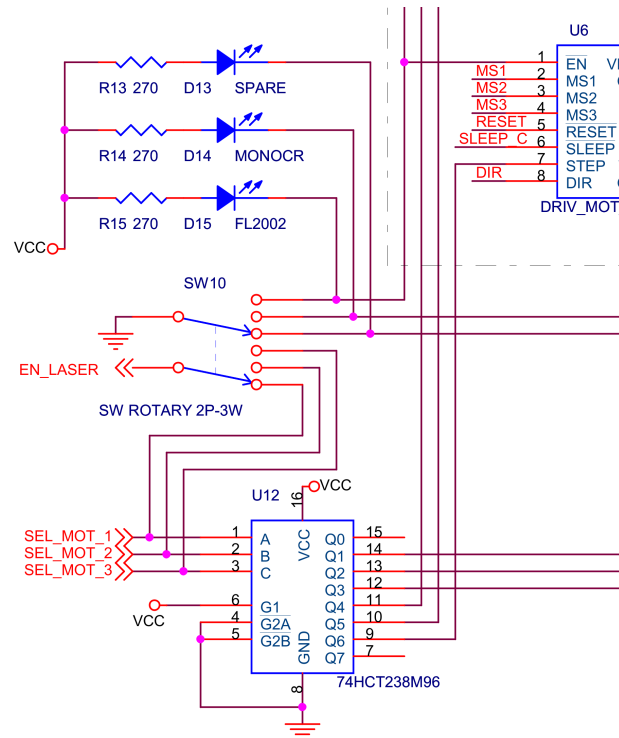


Fig. 4.10: Schema di collegamento del chip 74HCT238M96.

Uno dei due contatti é collegato a massa e 3 delle sue 6 possibili posizioni sono collegate ai 3 gruppi di driver per i tre dispositivi, in questo modo a seconda della posizione scelta dall'operatore una sola delle posizioni andrà collegata a massa ed abiliterá quindi il rispettivo dispositivo.

Per avere un riscontro visivo di quale dispositivo sia selezionato si é pensato di predisporre dei led, questi hanno i catodi collegati alla rispettiva posizione del selezionatore e l'anodo collegato a Vcc tramite una resistenza calcolata opportunamente per ottenere la corrente necessaria all'accensione. Questo collegamento consente anche di ottenere un livello di tensione diverso da zero sulle posizioni che rimangono non collegate e quindi di disabilitare i due dispositivi non selezionati.

L'altro contatto é invece collegato ad un pin RD7 del pic e 3 delle sue 6 posizioni sono collegate ai pin di selezione del chip 74HCT238M96; questo per rendere possibile, al programma che implementeremo, l'identificazione del laser selezionato.

#### 4.4.3 Finecorsa

La scheda dovrà essere in grado di rilevare un finecorsa ed identificare quale di essi sia stato attivato. Questo riconoscimento può essere compiuto solo dal PIC ma anche in questo caso non é conveniente utilizzare pin diversi per ogni finecorsa; si é deciso quindi di dedicare un solo pin del pic alla rilevazione di avvenuto finecorsa, successivamente, mediante un circuito dedicato, il programma andrà ad analizzare tutti i finecorsa per trovare quale sia stato attivato.

Per prima cosa osserviamo quali siano i finecorsa presenti nei dispositivi e come collegarli.

#### FL2002

In Fig. 4.11 (particolare della Fig. A.2) si può vedere che i finecorsa sono di tipo ottico, costituiti cioè da un fotodiode e da un transistor fotosensibile; in condizioni normali il diodo illumina il transistor, il quale si comporta come un interruttore chiuso (cortocircuito tra collettore ed emettitore), quando invece tra i due s'interpone una barriera, il transistor, non capta più luce e si comporta come un interruttore aperto. Nel nostro caso i tre motorini hanno due placchette nere poste alle estremitá del loro possibile movimento meccanico, in questo modo si può avere un segnale di avvertimento ed evitare di arrivare ad un contatto fisico che potrebbe rovinare lo strumento data la sua fragilitá e la sua grande precisione.

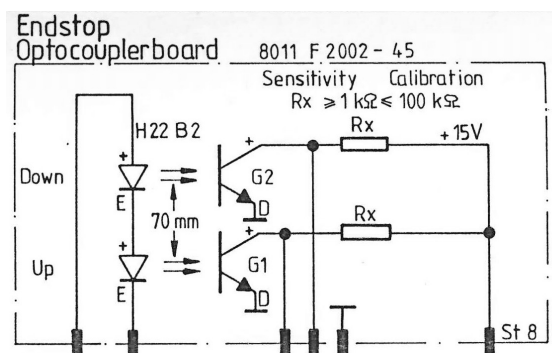


Fig. 4.11: Collegamenti interni dei finecorsa nel laser FL2002[7].

Osservando gli schemi elettrici si ricava che, nei tre connettori da 20 pin, i collegamenti riguardanti i finecorsa devono essere i seguenti:

- pin 5: a massa
- pin 6: a  $V_{cc}$  mediante un'opportuna resistenza
- pin 7-8: alla porzione di circuito dedicata ai finecorsa
- pin 14: a  $V_{mot}$
- pin 13: avrebbe bisogno, per accendere il led di finecorsa, di una tensione  $V_{mot}$  che passi a 0V quando il finecorsa viene attivato. per com'è stata pensata la scheda questo segnale non è disponibile se non aggiungendo ulteriori componenti elettronici; non essendo questo led necessario si è deciso di lasciarlo sempre spento collegando questo pin sempre a  $V_{mot}$ .

Effettuando questi collegamenti, finché un finecorsa non viene attraversato dalla propria placchetta, sui collettori dei transistor è presente un potenziale nullo (essendo gli emettitori collegati a massa), nel momento in cui una placchetta interrompe il fascio di luce all'interno di un finecorsa, il collettore del relativo transistor passa ad un potenziale positivo dato dalla tensione che si applica ai pin 14 del connettore (vedi Fig. A.2).

Osservando la Fig. A.3 si nota come il cristallo ha bisogno di alcuni collegamenti aggiuntivi, questo dispositivo, per funzionare correttamente, deve infatti essere mantenuto ad una temperatura costante e controllata; il circuito che si occupa di questo è già presente nel laser ed è sufficiente alimentarlo con le tensioni descritte in figura (pin 17-18).

### Monocromatore

Nel monocromatore sono invece presenti dei finecorsa meccanici; in condizioni normali il tasto non è premuto e i pin 1-2 risultano di conseguenza collegati, nel momento in cui invece il tasto viene premuto, il collegamento avviene tra i pin 3-4.

Anche in questo caso il motore aziona una vite che fa traslare una slitta, su questa è fissata una sporgenza che andrà a premere i finecorsa quando lo spazio di movimentazione sarà esaurito.

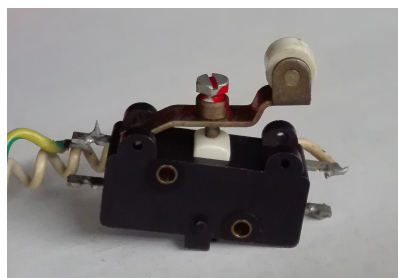


Fig. 4.12: Uno dei due finecorsa presenti nel monocromatore.

### Circuito utilizzato per la rilevazione dei finecorsa

Prendendo atto di queste due tipologie si vuole realizzare un circuito in grado di rilevare se uno tra i pin 7-8 dei 3 connettori del FL2002 passa da una tensione pari a 0 ad una tensione positiva, oppure se uno dei due finecorsa del monocromatore ha collegato i pin 3-4. Tale rilevazione andrà chiaramente compiuta dal PIC, un aspetto che bisogna però considerare è che il PIC ammette in ingresso solo tensioni tra 0-5V mentre i segnali



provenienti dal FL2002 sono anche di 15V; per questo motivo é stato necessario progettare un circuito in grado di leggere le tensioni in gioco in maniera compatibile e sicura. Per fare questo si é scelto d’inserire dei transistor mosfet (BS170) ed utilizzarli in modalit  ON/OFF.

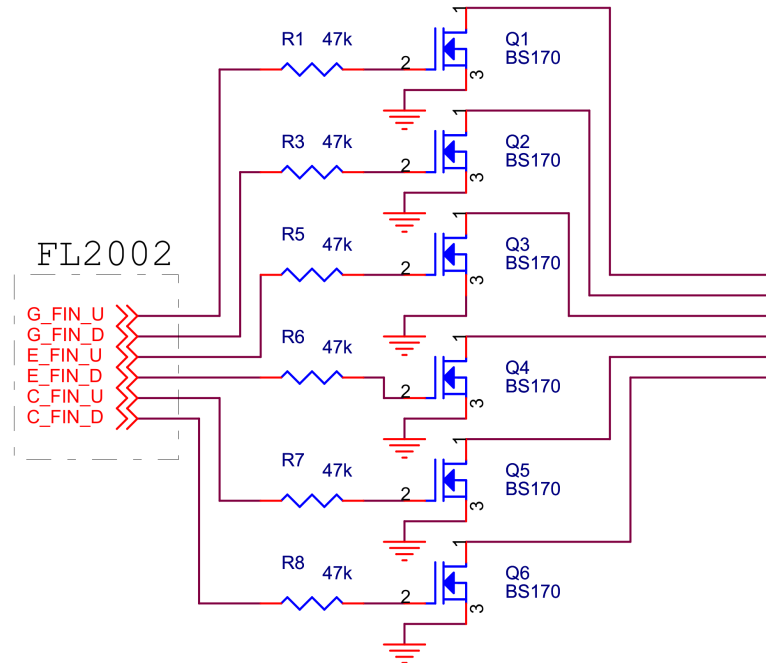


Fig. 4.13: Schema di collegamento del circuito per il rilevamento dei finecorsa.

Lo schema riportato in Fig. 4.13 ha il seguente funzionamento: finch  i contatti riportanti le tensioni dei pin 7-8 sono a massa, la corrente di gate sui transistor   nulla e quindi essi sono come interruttori aperti, in questo caso attraverso le resistenze di pull-up<sup>1</sup> (R4) tutti i drain sono portati alla tensione di 5V; nel momento in cui uno dei fili (finecorsa) si porta ad una tensione positiva, la corrente di gate che inizia a circolare attiva il transistor che si “chiude” portando il drain a massa. Con questo circuito abbiamo quindi posto a 5V tutti i contatti da analizzare e nel momento in cui un finecorsa sar  attivato, il relativo contatto scender  a 0V.

I finecorsa del monocromatore non pongono vincoli dati da circuiti interni, grazie a ci  si pu  decidere di verificarli alla stessa maniera, collegando gli switch in modo che l’uscita sia sempre a 5V e scenda a 0V nel caso in cui il finecorsa venga azionato. In questo modo la procedura per verificare un avvenuto finecorsa sar  identica, con un risparmio dal punto di vista dei componenti hardware necessari e minor complessit  nel software da realizzare.

Avendo previsto la possibilit  di aggiungere altri due motori si sono inoltre predisposti i collegamenti anche per altri 4 finecorsa (due per motore).

Per poter ottenere l’informazione di avvenuto finecorsa si   deciso di collegare, in modo appropriato, tutti i contatti di finecorsa ad un unico pin del PIC abilitato all’utilizzo degli interrupt (RB0). Per poter far s  che tale pin resti a livello alto, a meno che uno dei finecorsa non sia azionato, si sono utilizzati dei diodi collegati come si vede in Fig. B.3; in questo modo la resistenza di pull-up tiene il pin a 5V finch  tutti i finecorsa sono a 5V, nel momento in cui uno di questi scende a 0V anch’esso scende, la presenza dei diodi evita

<sup>1</sup>Le resistenze sono calcolate in modo da ottenere una minima corrente sufficiente per portare il segnale al PIC o ai diversi chip, con una tensione di 5V si sono quindi utilizzate resistenze da 47k 

che vadano a 0V anche tutti gli altri finecorsa.

Per avere un riscontro dell'avvenuto finecorsa si é collegato un led al pin di interrupt secondo lo schema riportato in Fig. 4.14

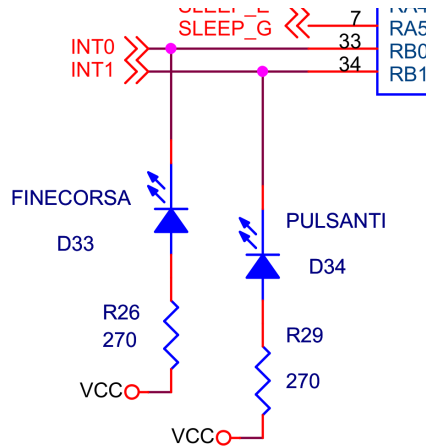


Fig. 4.14: Schema di collegamento led di interrupt.

A questo punto bisogna poter controllare uno per uno i contatti, esaminando le loro tensioni, per avere la possibilità di capire quale finecorsa sia stato attivato; non potendo dedicare un pin del PIC per ogni finecorsa, si é pensato di utilizzare un circuito integrato con funzione di multiplexer. La funzione di questo dispositivo é quella di selezionare un singolo segnale elettrico fra diversi posti in ingresso, la selezione avviene in base al valore dei pin di selezione.

L'integrato scelto é il HEF4067B a 16 ingressi, il suo funzionamento permette di selezionare un singolo segnale tra i diversi posti in ingresso, la selezione avviene in base al valore dei 4 pin di selezione; é possibile inoltre abilitarne o disabilitarne l'uso mediante un pin di Enable.

I 12 segnali di finecorsa vanno quindi collegati ai pin d'ingresso dell'integrato ed avendo la necessità di verificare quando uno di essi passa da Vcc a massa, i restanti 4 pin devono essere collegati direttamente a Vcc.

Il pin d'uscita é quello dove si dovrà effettuare la lettura e quindi va collegato direttamente al PIC (RE0). I 4 pin di selezione dovranno essere gestiti dal pic e vengono quindi collegati ad esso tramite (RE1, RE2, RB3, RA1). Infine, si é pensato di utilizzare anche il pin di enable, il motivo di questa scelta e lo schema di collegamento verranno descritti nel Par. 4.4.4

#### 4.4.4 Pulsanti

Per quanto riguarda i pulsanti fisici ci sono due diverse categorie:

- quelli inseriti nel progetto per poter impostare il sistema di controllo
- quelli già presenti sui dispositivi per pilotare “manualmente” i diversi motori.

Per semplificare la scrittura del programma si é pensato di utilizzare lo stesso concetto utilizzato per i finecorsa, facendo uso dello stesso integrato e di un circuito molto simile. In Fig. B.4 si può vedere come siano stati collegati i pulsanti che andranno inseriti nel

Control	Address				Channel ON
	A3	A2	A1	A0	
L	L	L	L	L	Y0 = Z
L	L	L	L	H	Y1 = Z
L	L	L	H	L	Y2 = Z
L	L	L	H	H	Y3 = Z
L	L	H	L	L	Y4 = Z
L	L	H	L	H	Y5 = Z
L	L	H	H	L	Y6 = Z
L	L	H	H	H	Y7 = Z
L	H	L	L	L	Y8 = Z
L	H	L	L	H	Y9 = Z
L	H	L	H	L	Y10 = Z
L	H	L	H	H	Y11 = Z
L	H	H	L	L	Y12 = Z
L	H	H	L	H	Y13 = Z
L	H	H	H	L	Y14 = Z
L	H	H	H	H	Y15 = Z
H	X	X	X	X	none

[1] H = HIGH voltage level; L = LOW voltage level; X = don't care.

Fig. 4.15: Tavola della verità del chip HEF4067B.

pannello frontale: un'estremità collegata a massa e l'altra all'ingresso del multiplexer, inoltre, come per i finecorsa, ogni ingresso è tenuto a livello alto tramite resistenze di pull-up; nel momento in cui un pulsante viene premuto effettua un cortocircuito e porta il relativo ingresso del multiplexer a massa. Per evitare errori dovuti a falsi contatti e/o rimbalzi di tensione si è inoltre deciso di posizionare un condensatore per ogni ingresso al fine di stabilizzare la tensione.

Per quanto riguarda i pulsanti già presenti, l'unico dispositivo dove sono predisposti dei controlli manuali è l'FL2002: in prossimità di ogni motorino è infatti presente una levetta a due posizioni e uno zero normale.

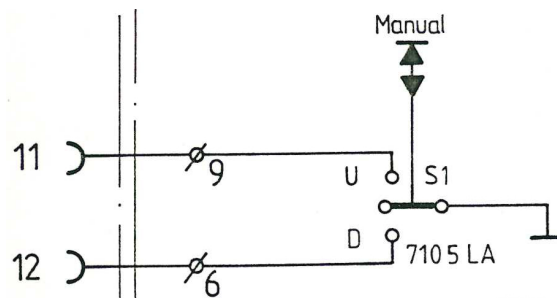


Fig. 4.16: Collegamento interno delle levette presenti sul laser FL2002[7].

In Fig. 4.16 è riportato un particolare dello schema di collegamento riguardante la levetta per il controllo manuale dell'Etalon (ma il collegamento è identico anche per le altre due); vediamo che per tutti tre i motori la levetta è collegata a 3 pin del connettore e funziona come un interruttore: un'estremità ha un collegamento fisso mentre la parte mobile, normalmente libera ed isolata, viene mossa manualmente verso una delle due posizioni possibili. Volendo unificare il metodo d'identificazione con gli altri pulsanti si è deciso di collegare l'estremità fissa di ogni levetta a massa e le due posizioni vengono collegate agli ingressi del multiplexer, in questo modo tutti i collegamenti dei possibili pulsanti con il multiplexer saranno normalmente a livello alto ed andranno a massa una volta premuti. Allo stesso modo dei finecorsa, tutti gli ingressi del multiplexer vengono collegati tramite diodi ad pic mediante il pin RB1 anch'esso adibito alla gestione degli interrupt.

Avendo utilizzato lo stesso tipo di circuito per finecorsa e pulsanti ed essendo due tipi di eventi indipendenti, il programma non dovrà gestire entrambi i circuiti contemporaneamente e si può quindi pensare di utilizzare gli stessi pin del pic per entrambi multiplexer, i pin per la selezione saranno cioè collegati ad entrambi gli integrati per la selezione dell'ingresso, il programma si occuperà poi, volta per volta, di leggere solo il multiplexer necessario. Per poter far questo occorre utilizzare il pin di enable presente nei multiplexer per poter abilitare solo l'uscita desiderata. Anche in questo caso si è pensato di risparmiare un pin ed utilizzare (RB2) per l'abilitazione di uno o dell'altro multiplexer: facendo uso di un transistor in modo molto simile a quello illustrato per i finecorsa, si può creare una biforcazione che produca il livello logico alto su un ramo e basso sull'altro o viceversa. Lo

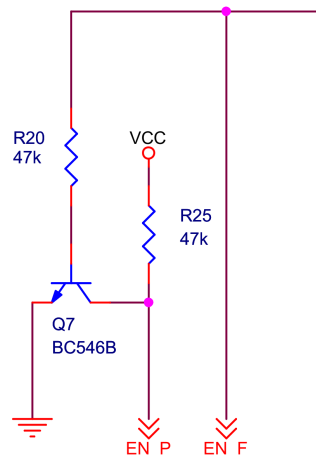


Fig. 4.17: Schema di collegamento per l'abilitazione alternata dei due multiplexer.

schema riportato in Fig. 4.17 ha il seguente funzionamento: nel momento in cui il pin è impostato a valore logico 0, la corrente di base è nulla e il transistor funziona da circuito aperto, sul collettore si trova quindi la tensione Vcc; nel momento in cui il pin è invece a 5V il transistor "si chiude" ed il collettore si trova a massa.

#### 4.4.5 Connettori dei dispositivi

##### FL2002

Come già accennato, l'FL2002 utilizza un connettore da 60 pin, riassumendo ciò che è stato detto nelle sezioni precedenti, i collegamenti di questo connettore sono riportati nella Fig. 4.18

dove G, E e C stanno per Grating, Etalon, Cristallo; 1A, 2A, 1B, 2B sono le fasi dei motori; FIN sono i collegamenti per i finecorsa; LEV indica i collegamenti per una levetta ed infine U, D indicano il verso up o down.

##### Monocromatore

Il monocromatore necessita invece di molti meno collegamenti ed utilizza una porta seriale da 9 pin (DB9) che viene collegata come mostrato in Fig. 4.19

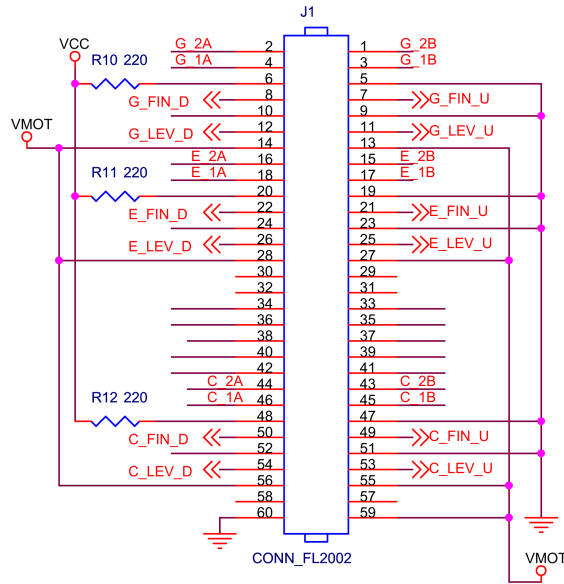


Fig. 4.18: Schema di collegamento per il connettore del laser FL2002.

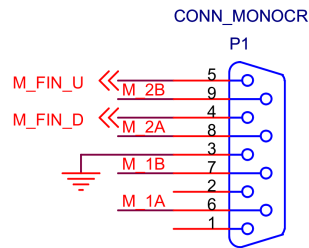


Fig. 4.19: Schema di collegamento per il connettore del Monocromatore.

## Spare

Per i due motori addizionali si é deciso di utilizzare la stessa tipologia di collegamento usata dal monocromatore, per semplicitá di layout i collegamenti risultano come da Fig. 4.20

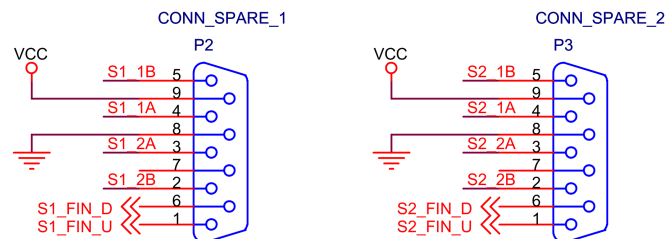


Fig. 4.20: Schema di collegamento per i motori di spare.

## 4.5 Disegno della scheda elettronica

Una volta completato lo schema elettrico si puó passare all'uso del software "Orcad Layout" per il disegno della PCB. Dopo aver importato il progetto, osservando l'ingombro dei vari componenti, si ha una prima idea sulle dimensioni effettive che potrà assumere la

scheda. Teoricamente il bordo della scheda é definibile a piacere, bisogna però considerare che, per il metodo di stampa su scheda fotoresistiva, bisogna utilizzare delle schede commerciali, le quali hanno delle misure standard, nel nostro caso la misura più vicina alla nostra necessità é di 200x150mm.

Per il posizionamento dei componenti si é seguita una scala di priorità, cercando di minimizzare la lunghezza delle piste di collegamento.

1. considerando che le porte seriali DB9, il connettore da 60 pin ed il connettore per la tensione di alimentazione sono i componenti che vanno collegati con i cavi provenienti dall'esterno, si é deciso di posizionarli ai bordi della scheda per facilitarne il collegamento.
2. Vicino alle due porte dedicate alla comunicazione mediante protocollo seriale si sono quindi posizionati il MAX232 ed i componenti necessari al suo funzionamento.
3. I driver dei motori sono stati posizionati il più possibile vicino ai rispettivi connettori.
4. Il selezionatore ed il demultiplexer sono stati posizionati ad una distanza media tra i driver.
5. Tutto il necessario per la rilevazione dei finecorsa é stato avvicinato il più possibile al connettore del FL2002 ed a quello del monocromatore.
6. Il pic é stato posizionato in una zona più centrale possibile, essendo collegato a tutte le altre parti di circuito; vicino ad esso é stato posto chiaramente l'oscillatore.
7. Il connettore dell'LCD, essendo molto lungo é stato posizionato lungo un bordo della scheda e vicino al PIC in prossimitá dei pin ad esso collegati.
8. I connettori dei pulsanti sono stati posizionati a bordo scheda e con loro tutti i componenti necessari alla loro gestione.
9. I restanti componenti (programmatore, potenziometro, ventola) sono stati posizionati nelle zone rimaste libere e sempre cercando di minimizzare la lunghezza dei collegamenti da effettuare.

Il collegamento di tutti questi componenti risulta impossibile da realizzare utilizzando solo una faccia della scheda, in molti casi ci si trova infatti ad affrontare il problema di piste che s'intersecano. Per ovviare a questo problema si può far uso di una scheda a doppia faccia, nella quale le piste possono passare da sopra (top) a sotto (bottom) mediante dei fori di collegamento (via). Il software aiuta molto in questa parte di progetto poiché ricorda continuamente quali pin siano da collegare ed inoltre, mediante dei parametri preimpostati, consente di tracciare le piste della dimensione corretta, avvisando nel caso si creino dei cortocircuiti involontari tra piste diverse.

## 4.6 Realizzazione del prototipo

### 4.6.1 Stampaggio del circuito

Una volta terminato il progetto si é pronti per la realizzazione fisica della PCB; esistono diversi modi far questo ed i due più comunemente utilizzati sono:

- inviare il progetto ad una ditta specializzata in stampaggio PCB
- stampare il circuito mediante fotoincisione

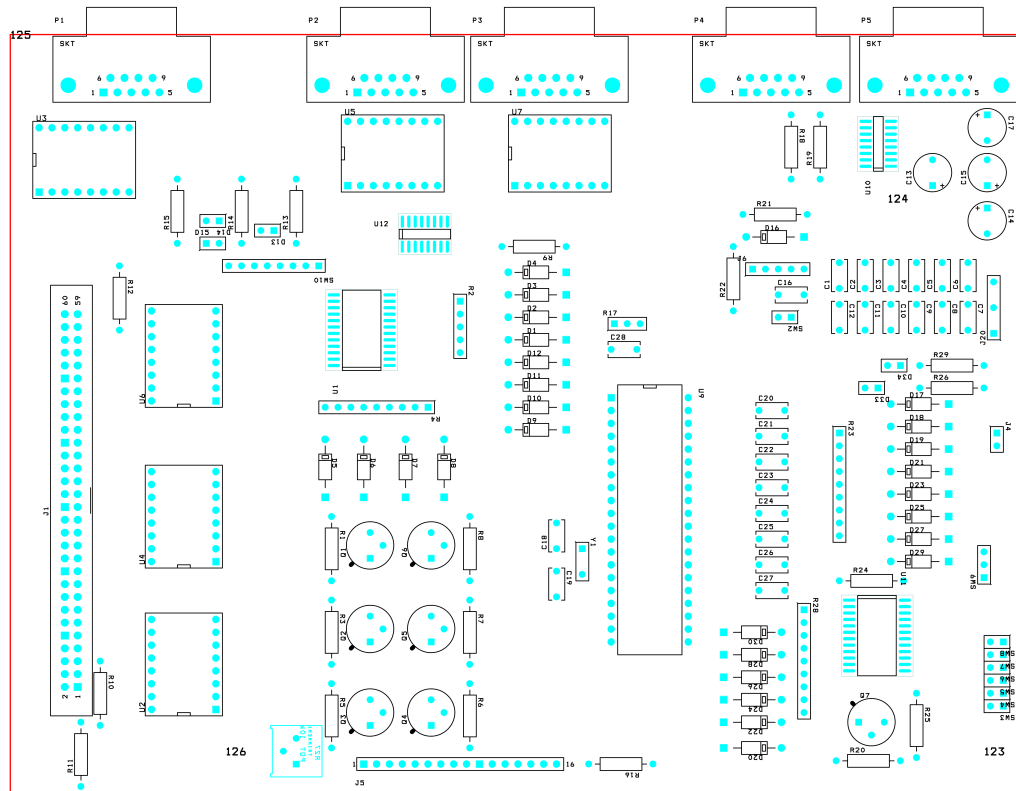


Fig. 4.21: Posizionamento dei componenti all'interno della PCB.

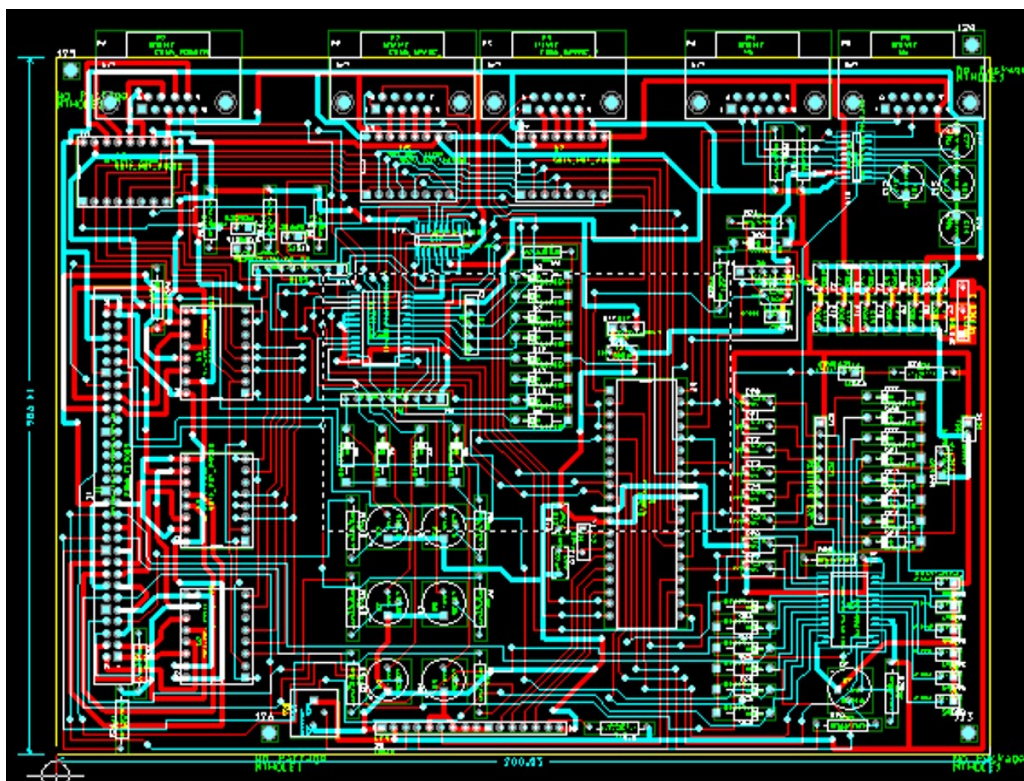


Fig. 4.22: Piste di collegamento da i componenti della PCB: blu (top) e rosso(bottom).

Il primo metodo è sicuramente il più comodo, sicuro ed efficiente; i macchinari utilizzati dalle aziende specializzate consentono infatti di realizzare le piste con una precisione molto alta ed è quindi possibile miniaturizzare tutta la PCB.

Il secondo metodo, al contrario, è molto grossolano e per questo motivo bisogna considerare un adeguato margine di sicurezza.

Affidare il lavoro ad un'azienda è piuttosto costoso e sicuramente indicato nel caso si abbia necessità di più copie della stessa scheda, bisogna essere quindi sicuri che tutto funzioni correttamente; se possibile è quindi sempre meglio effettuare la prima prova utilizzando il secondo metodo per poter eventualmente apportare delle modifiche.

### Fotoincisione<sup>[12]</sup>

Il metodo della fotoincisione consente di “disegnare” sul rame le piste usando la luce e particolari sostanze chimiche.

Le basette in rame necessarie per questa procedura sono normali basette per circuiti stampati<sup>2</sup> sulla quale è stesa, in modo omogeneo, una pellicola fotosensibile; illuminando questa bassetta con luce ultravioletta, il polimero che costituisce la pellicola diventa solubile in una soluzione basica e quindi può essere facilmente rimosso. Schede di questo tipo sono facilmente reperibili in commercio e vengono vendute con una pellicola adesiva a protezione dello strato fotosensibile.

Il primo passo per eseguire questa procedura è quello di creare un master costituito da un disegno in scala 1:1 delle piste su un supporto che sia trasparente alla luce ultravioletta, in questo modo l'inchiostro proteggerà le zone in cui si desidera che rimanga il rame; nel nostro caso si è stampato il circuito su fogli lucidi utilizzando una comune stampante laser.

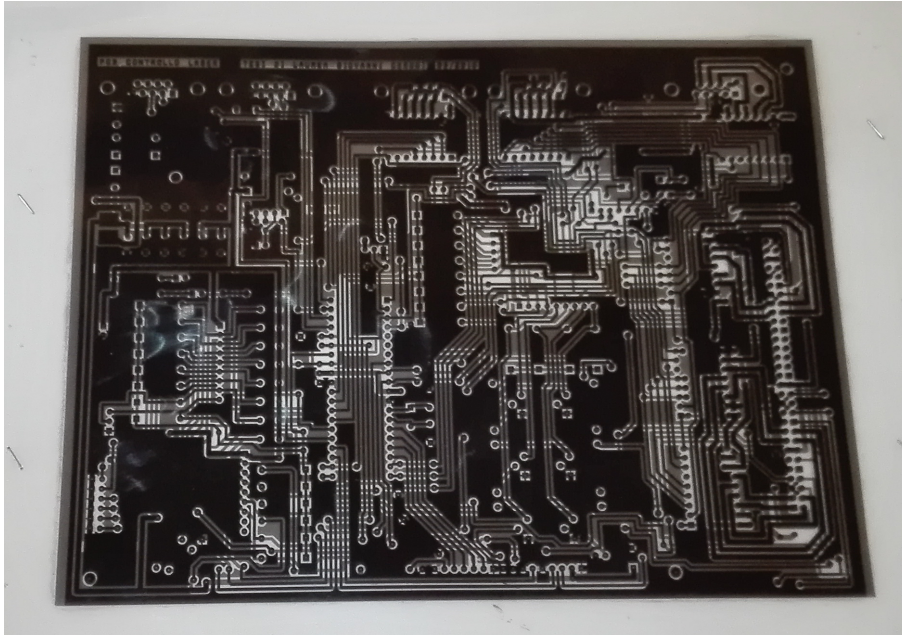


Fig. 4.23: Lucidi per il trattamento della scheda fotosensibile.

A questo punto bisogna togliere la pellicola adesiva e appoggiare il master sopra alla faccia di rame della bassetta (questo procedimento va compiuto più velocemente possibile per evitare che la parte da proteggere venga illuminata). La luce ultravioletta necessaria è poi solitamente fornita da un bromografo, strumento appositamente studiato per questa

<sup>2</sup>piastra di materiale isolante sulla quale è steso uno strato di rame



funzione.



Fig. 4.24: Bromografo utilizzato per il trattamento della scheda.

Terminata l'esposizione é quindi possibile evidenziare il disegno utilizzando una soluzione alcalina che scioglierá la vernice fotosensibile nelle zone illuminate dagli UV lasciando intatte le zone rimaste in ombra. Come soluzione si usa solitamente della comune soda caustica (NaOH) preparata sciogliendone circa 10g per litro d'acqua. Immergendo la basetta la soluzione fa effetto in circa un minuto, dopo il quale bisogna accuratamente lavarla con acqua per eliminare ogni traccia di soda che altrimenti continuerebbe ad agire aggredendo nel tempo anche la zona protetta dalla pellicola.

L'ultimo passaggio che resta da compiere é quello di eliminare il rame rimasto privo della pellicola protettiva, questa operazione viene effettuata immergendo la basetta in una soluzione di cloruro ferrico ( $\text{FeCl}_3$ ) per la sua capacità di corrodere il rame.

Per ultimare la preparazione della PCB non resta che praticare tutti i fori necessari (componenti a foro passante e vie).

#### 4.6.2 Assemblaggio dei componenti

Tutti i componenti necessari (eccetto i driver per i motori) sono stati scelti dal catalogo della ditta RS (per comodità e velocità di consegna); La quasi totalità dei componenti sono a foro passante ma alcuni sono stati scelti in versione SMD, per una questione di spazio e comodità nella gestione delle piste di collegamento.

Nella lista sono compresi i connettori a 2,3,5,8 pin necessari per il collegamento dei componenti fissati al pannello frontale; per evitare errori nella connessione si sono scelti connettori di tipo Molex che, essendo asimmetrici, permettono l'accoppiamento maschio-femmina esclusivamente in un verso, impedendo in questo modo un collegamento errato che danneggerebbe il circuito ed i componenti

L'assemblaggio nella PCB viene effettuato mediante saldatura a stagno, la maggior parte dei componenti viene saldata direttamente alla scheda, quelli per cui é meglio preve-

dere una possibile sostituzione (pic e driver motori) sono invece stati accoppiati ad appositi zoccoli.

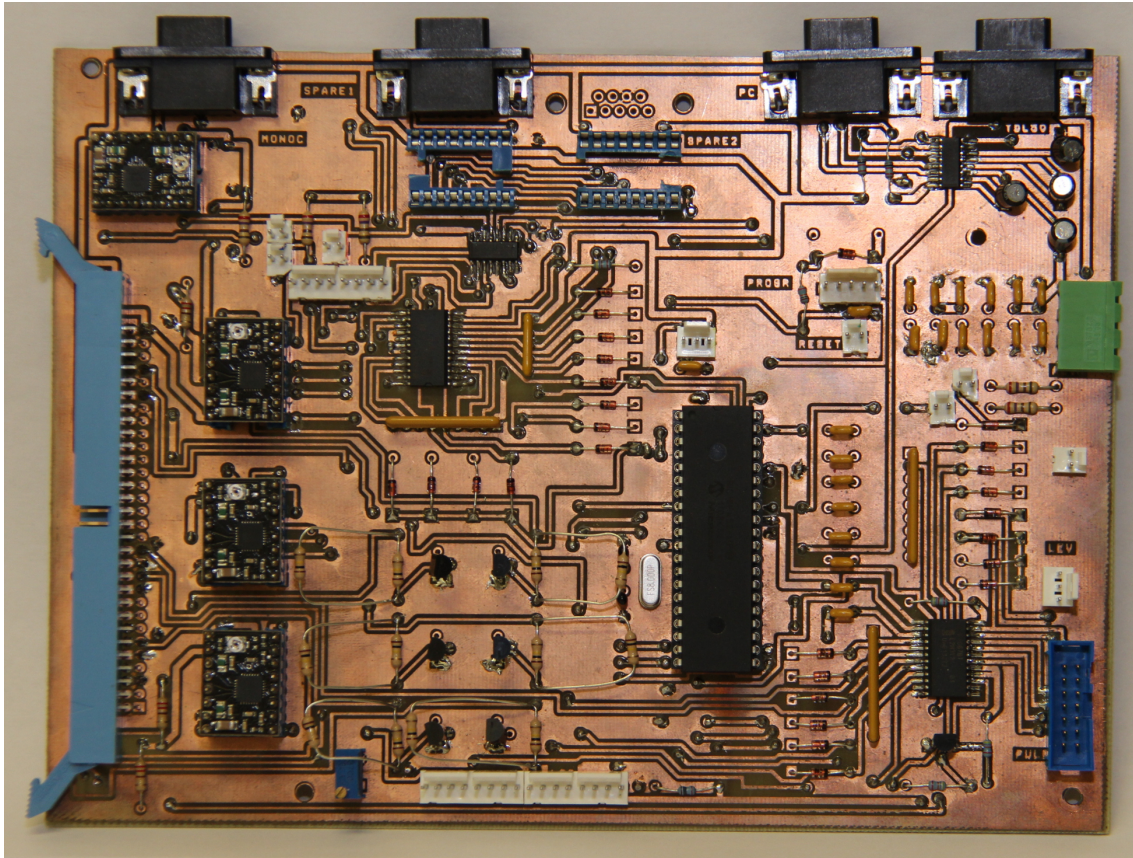


Fig. 4.25: PCB completa di tutti i componenti saldati.

### Alimentatore

L'alimentatore é il dispositivo che si occupa di convertire la tensione di rete nella tensione continua necessaria al funzionamento della PCB. Per questo progetto é necessario un alimentatore che fornisca due tensioni diverse:

- 5V per il funzionamento dei circuiti integrati.
- 12V per l'alimentazione dei motori.

Per una stima della potenza necessaria bisogna considerare il caso in cui sia massima la richiesta di corrente ed utilizzare la formula  $P = V \cdot I$ .

La corrente necessaria per il funzionamento dei vari integrati e dell'LCD non é facilmente calcolabile, le correnti in gioco sono molte ma sappiamo che saranno tutte nell'ordine dei mA; le correnti necessarie al funzionamento dei motori sappiamo invece poter arrivare ad un picco di 2A per driver.

Con queste conoscenze possiamo dire che la potenza derivante dai 5V non sará piú di qualche watt mentre la parte assolutamente predominante dipenderá dalla potenza assorbita dai motori. Abbiamo giá visto nel Par. 4.4.2 che i motori utilizzati contemporaneamente potranno essere al massimo 3, la potenza necessaria nel caso massimo é quindi calcolata con  $P_{mot} = (12 \cdot 2) \cdot 3 = 72W$

Considerando un po' di margine di sicurezza, l'alimentatore scelto é il TXL 100-0512DI della Tracopower il quale garantisce 100W fornendo: 12A per i 5V e 7A per i 12V.



Fig. 4.26: Alimentatore TRACOPOWER TXL 100-0512DI da 100W

### 4.6.3 Alloggiamento per la scheda elettronica

La scheda, una volta ultimata, dev'essere collocata in un alloggiamento che renda possibile l'installazione dello schermo LCD, i pulsanti e gli altri componenti per l'interfacciamento con l'utente; deve inoltre essere considerato lo spazio per l'alimentatore e le ventole. A questo scopo si é scelto un case in formato rack, di dimensioni 88.1x482.6x334mm.

Per preparare il pannello frontale si sono prese le misure di tutti i componenti da inserire e si sono praticati i fori necessari.

Oltre al pannello, frontale si sono praticati dei fori su quello inferiore, per poter fissare con delle viti sia la scheda che l'alimentatore <sup>3</sup>. Il risultato dell'assemblaggio finale é riportato in Fig. 4.27



Fig. 4.27: PCB nel suo rack di alloggiamento.

## 4.7 Conclusioni

La scheda elettronica progettata e realizzata, permette il controllo di tre dispositivi: il laser FL2002, il laser TDL50 ed il Monocromatore JOBIN YVON; il cuore di questa scheda é il microcontrollore PIC18F452 i cui pin sono collegati a diverse parti della scheda adibite a specifiche funzioni. La scheda puó essere gestita da pannello, tramite appositi pulsanti o da

<sup>3</sup>la scheda é stata fissata con dei distanziali in plastica per evitare di creare collegamenti elettrici con la scatola di metallo

remoto, tramite un pc collegato con un cavo seriale; l'interfacciamento con l'utente avviene a monitor e su un display LCD posto a pannello. La necessità di utilizzare un limitato numero di pin per un più grande numero di segnali ha reso necessaria la realizzazione di diversi circuiti facenti uso di un demultiplexer (per la selezione di uno solo tra i 6 motori disponibili) e di due mutiplexer per poter ispezionare uno alla volta i molti segnali provenienti da finecorsa e pulsanti). Lo schema elettrico preparato ha reso possibile la realizzazione della PCB mediante il procedimento di fotoincisione, si é successivamente provveduto alla saldatura dei componenti ed al posizionamento nel case predisposto.

## Programma di controllo

---

### 5.1 Introduzione

In questo capitolo viene descritto il programma scritto per essere inserito nel PIC che andrà a gestire l'intera scheda elettronica. Per prima cosa si analizzano i dispositivi che verranno collegati alla scheda, per capire quali siano i comandi che dovranno essere forniti per effettuare le movimentazioni richieste. Vengono successivamente dedicate alcune pagine alle nozioni fondamentali da conoscere per poter programmare un PIC, vengono cioè analizzate nel dettaglio le diverse funzioni che può svolgere e descritta la sintassi necessaria per il loro utilizzo. Una buona parte del programma principale si affida a funzioni scritte ad hoc per quest'applicazione, sono quindi riportate tutte le funzioni con una descrizione concettuale delle operazioni svolte; il capitolo si conclude con la descrizione del main program.

### 5.2 Analisi specifiche software richieste

Dopo aver preparato la scheda elettronica sulla base delle specifiche hardware, bisogna ora capire quali siano le specifiche software per procedere con la scrittura del programma da inserire nel PIC; si vuole cioè capire quali tipi di movimentazioni potranno essere richieste al programma e come dovranno essere gestite le diverse informazioni fornite. Vengono quindi ora esaminati più approfonditamente i tre dispositivi ponendo particolare attenzione alla movimentazione che verrà loro richiesta.

#### 5.2.1 Funzionalità richieste dal Laser FL2002

Questo laser sintonizzabile ha la possibilità di lavorare con l'aggiunta di un filtro Etalon o di un Cristallo SHG, per questo motivo l'utente dovrà specificare in quale modalità lavorare: solo Grating, con Etalon o con Cristallo. Queste tre modalità dovranno permettere di eseguire delle movimentazioni in modo automatico sulla base dei dati forniti dall'operatore. Si vuole inoltre che il programma fornisca la possibilità di un controllo manuale, l'operatore dovrà cioè poter muovere i motori in modo indipendente azionando le levette adibite a tale scopo e regolandone la velocità mediante il potenziometro rotativo posto sul pannello frontale.

#### Utilizzo del Grating

Nel caso si scelga la modalità con il solo Grating, l'operatore dovrà avere la possibilità di eseguire una scansione in lunghezza d'onda, ciò vuol dire poter scegliere un valore di partenza e uno di arrivo che siano compatibili con il range offerto dal reticolo. Per far questo il reticolo dovrà muoversi dall'angolazione associata alla lunghezza d'onda iniziale fino a quella finale ad una velocità adeguata all'esperimento che l'utente vorrà effettuare; bisogna quindi conoscere con esattezza come opera il reticolo al muoversi del motore al quale è collegato.

Il reticolo è connesso al rispettivo motore tramite cinghie ed ingranaggi che convertono il moto circolare del motore nel movimento angolare necessario al reticolo per riflettere lunghezze d'onda diverse. Data la grande precisione richiesta, l'angolo d'inclinazione deve variare molto lentamente, per questo motivo un movimento relativamente ampio del

motore si riflette in un movimento estremamente ridotto nel reticolo grazie all'utilizzo di numerose ruote dentate che fungono da riduttori di velocità.

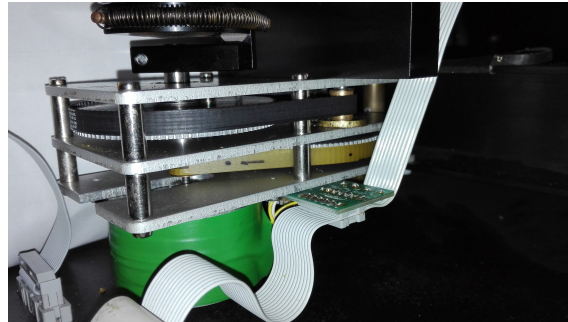


Fig. 5.1: Ingranaggi di collegamento tra motore e reticolo nel FL2002.

Non essendo presenti encoder di alcun tipo, il programma che andremo a scrivere non avrà modo di ricavare informazioni sulla posizione dei motori; per questo motivo sarà necessario impostare manualmente la posizione di partenza. Questo è reso possibile dal fatto che il Laser è dotato di un contatore numerico analogico che riporta quale sia la lunghezza d'onda d'esercizio istante per istante (con una precisione fino al centesimo di Angstrom). Un particolare da tenere in considerazione per questo motore è la presenza di un gioco abbastanza significativo nel contatore analogico, se si effettua un cambio di direzione nella rotazione, la lunghezza d'onda riportata si sfasa infatti di circa  $0,1 \text{ \AA}$



Fig. 5.2: Contatore analogico del laser FL2002.

Per conoscere la variazione di lunghezza d'onda corrispondente ad un passo del motore bisogna utilizzare la formula 5.2.1 ripresa dalla pagina di manuale riportata in Fig. A.4:

$$\frac{0,0042nm}{Ordine} \quad (5.2.1)$$

dove Ordine è un valore legato alla lunghezza d'onda in uscita secondo il grafico riportato in Fig. 5.3; il range in cui sarà necessario operare per gli esperimenti da svolgere nel laboratorio laser di SPES sarà sempre tra 460 e 640 nm, per questo motivo l'ordine sarà sempre pari a 5.

A questo punto utilizzando la formula 5.2.1 si ricava che ad un passo del motore collegato al grating, la lunghezza d'onda in uscita cambierà di  $0.0084 \text{ \AA}/passo$

In questa modalità è inoltre utile avere a disposizione la possibilità di posizionare il reticolo ad una specifica lunghezza d'onda, fornendo le informazioni necessarie al programma che si occuperà di eseguire la movimentazione in modo automatico.

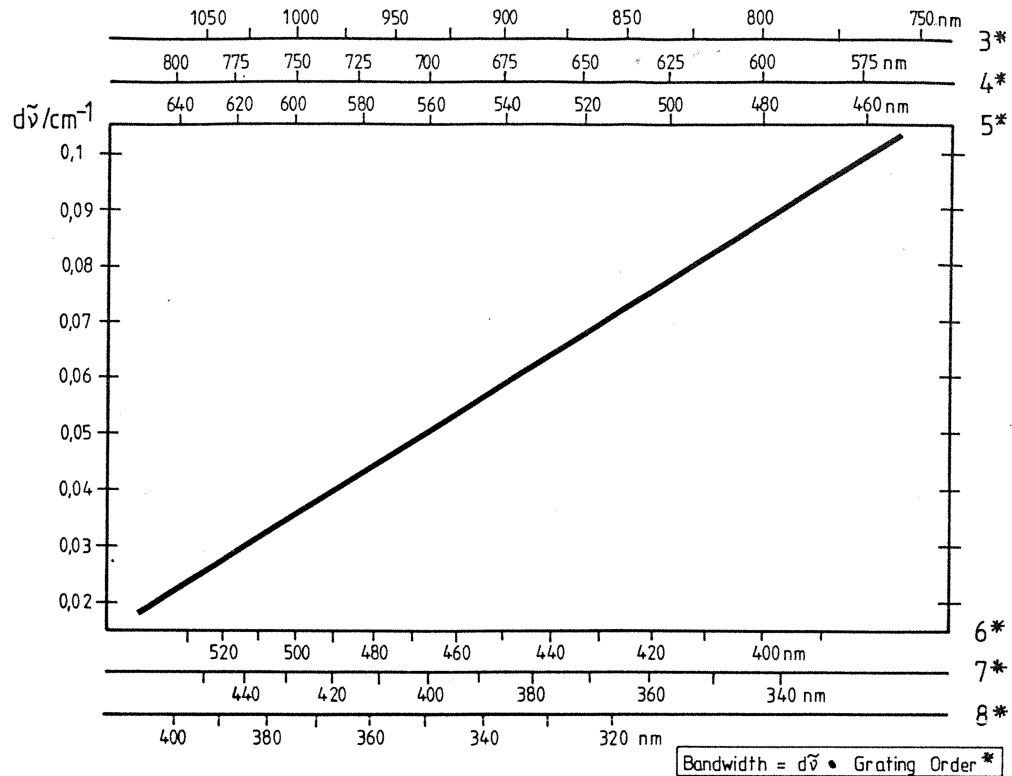


Fig. 5.3: Grafico per la determinazione dell'Ordine sulla base della lunghezza d'onda utilizzata. [7]

### Utilizzo del filtro Etalon

Nel caso si scelga la modalità con l'etalon, l'operatore dovrà avere esclusivamente la possibilità di eseguire una scansione in lunghezza d'onda. Come già visto nella Sez. 3.5.2, l'etalon permette di filtrare un fascio di luce e far passare una ristrettissima banda di frequenze, per poter far questo deve però essere posizionato con un'inclinazione molto accurata che dipende dalla lunghezza d'onda d'uscita desiderata. Ogni etalon ha una propria funzione che lega l'angolo del fascio entrante alla lunghezza d'onda d'uscita, tale funzione tiene conto principalmente della larghezza del filtro (distanza tra i due specchi) e del mezzo contenuto nella cavità. Per capire come inclinare il filtro per seguire il cambiamento della lunghezza d'onda, bisogna far riferimento alle seguenti formule:

$$\Delta\lambda = -\lambda_0 \cdot \alpha^2 / 2n^2 \quad (5.2.2)$$

dove:

$\lambda_0$  é la lunghezza d'onda di partenza

$n$  é l'indice di rifrazione del mezzo contenuto nella cavità.

$$\alpha = 1.25 \cdot 10^{-5} \cdot ZE \quad [rad] \quad (5.2.3)$$

dove:

$ZE$  sono i passi effettuati dal motore.

L'equazione 5.2.2 indica quanta sia la differenza in lunghezza d'onda se il filtro viene inclinato di un angolo  $\alpha$  ricavabile dall'equazione 5.2.3.

Utilizzando queste equazioni il programma potrà sapere quando muovere il reticolo ed il filtro etalon.

### Utilizzo del Cristallo SHG

Nel caso si scelga la modalità con il cristallo, l'operatore avrà esclusivamente bisogno di eseguire una scansione in lunghezza d'onda. Come già visto nella Sez. 3.5.3, il cristallo permette di dimezzare la lunghezza d'onda del fascio che lo attraversa; come per l'etalon ogni lunghezza d'onda entrante ha bisogno di un preciso angolo d'incidenza sul cristallo. Per poter operare una scansione in questa modalità bisognerà quindi muovere in modo opportuno il cristallo perché segua il reticolo.

A differenza dell'etalon l'equazione che lega lunghezza d'onda e angolo d'incidenza, varia per ogni tipo di cristallo ed è sempre una relazione polinomiale di grado superiore al secondo; è quindi molto complicato determinare istante per istante come posizionare il cristallo. La tipologia di esperimenti che si dovranno effettuare in questa modalità prevede tuttavia range di scansione molto limitati e quindi spostamenti molto piccoli, per questo motivo si può considerare la dipendenza come lineare e calcolare di volta in volta, a seconda del range di scansione, la relazione lineare che lega il movimento del reticolo a quello del cristallo.

La procedura per poter ricavare questa relazione, è quindi di determinare alcuni punti all'interno del range di scansione del reticolo in cui calibrare manualmente la posizione del cristallo; partendo da questi punti si potrà poi effettuare l'interpolazione lineare. Si è pensato di suddividere il range di scansione in 4 parti uguali, in questo modo, oltre al punto di partenza, si avranno altri 4 punti a disposizione. Le coppie di punti verranno memorizzate tenendo conto di quanti passi sono stati effettuati di volta in volta dal reticolo e dal cristallo. Con l'equazione della retta così ottenuta, il programma saprà come muovere il cristallo perché resti sempre agganciato al grating, mantenendo in questo modo costante la potenza del fascio in uscita.

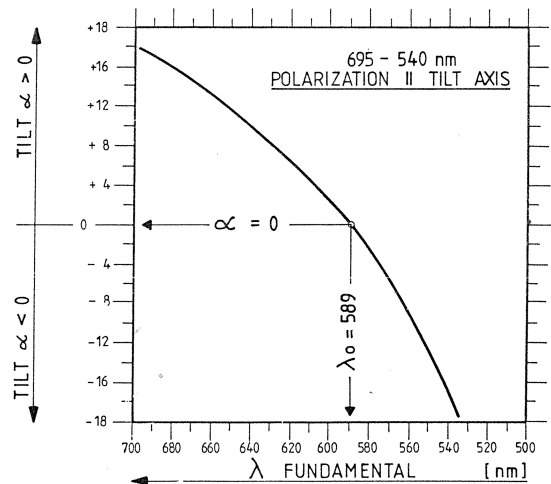


Fig. 5.4: Grafico della funzione che lega l'inclinazione del cristallo alla lunghezza d'onda in uscita.[7]

### 5.2.2 Funzionalità richieste dal Monocromatore JOBIN YVON

Come già illustrato nella Sez. 3.7 il monocromatore ha al suo interno un reticolo di diffrazione, per questo motivo, da un punto di vista software, il suo controllo è del tutto simile ad un laser sintonizzabile che utilizzi il solo grating. Il reticolo è collegato anche in questo caso al suo motore tramite molti ingranaggi (Fig. 5.6) e la sua tipologia di movimentazione sarà del tutto analoga a quella del reticolo nel FL2002.





Fig. 5.5: Ingranaggi di collegamento tra motore e reticolo nel Monocromatore.

Come per l'FL2002 anche il monocromatore JOBIN YVON possiede un contatore numerico analogico che riporta la lunghezza d'onda d'uscita istante per istante; tale contatore ha la precisione del decimo di Angstrom e servirá anche in questo caso per inizializzare il programma inserendo la posizione iniziale del reticolo.

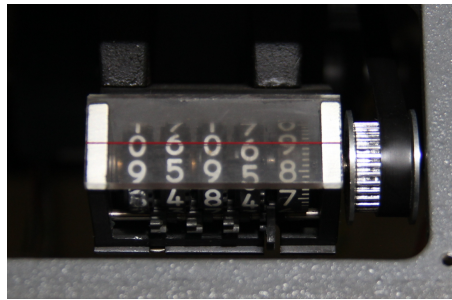


Fig. 5.6: Contatore analogico del monocromatore.

Tramite misure sperimentali si ricava che, per ogni passo del motore collegato al reticolo, la lunghezza d'onda varia di  $0.02 \text{ \AA}/\text{passo}$ .

Anche nell'utilizzo di questo dispositivo puó essere utile la possibilitá di posizionarsi ad una specifica lunghezza d'onda in modalitá automatica oltre a dotare il dispositivo di una movimentazione manuale che verrá affidata alla levetta posizionata nel pannello frontale del case di alloggiamento per la PCB.

### 5.2.3 Funzionalitá richieste dal Laser TDL

Questo laser é giá totalmente indipendente e dotato del proprio controllore, scopo della PCB é esclusivamente quello di permettere la comunicazione tramite la porta seriale per il controllo da pc.

Il sistema di controllo utilizza un'apposita sintassi che é riportata in appendice (Fig. A.5 e A.6); ció che viene chiesto é quindi la possibilitá di utilizzare la sintassi predefinita del dispositivo.

La sintassi prevede sia per la trasmissione che per la ricezione, una stringa di caratteri alfanumerici che termini sempre con il carattere di fine messaggio (carriage return). Tutto questo dovrá essere gestito non dal modulo USART hardware ma attraverso il modulo software che verrá descritto nelle prossime sezioni.

### 5.3 Software utilizzato per la programmazione

Programmare un microcontrollore significa fornirgli delle istruzioni in un linguaggio specifico (linguaggio macchina) che lo guidino passo passo verso l'obiettivo desiderato. Il linguaggio macchina é il piú sintetico ed essenziale possibile; per questo motivo risulta molto difficile da gestire per la persona che vuole scrivere un programma. Ogni casa produttrice fornisce quindi dei software che rendano la stesura del programma piú intuitiva; in particolare la Microchip rende disponibile per gli utilizzatori dei suoi prodotti, il software "MPLAB X IDE", il quale permette di scrivere un programma in linguaggio C ed attraverso un compilatore (XC8) tradurlo in linguaggio macchina. Una volta preparato il codice e compilato, non resta che programmare il PIC servendosi di un software di programmazione che, collegandosi fisicamente con il PIC (pin  $\overline{MCLR}$ , RB6, RB7), trasferisce il codice nella memoria al suo interno.

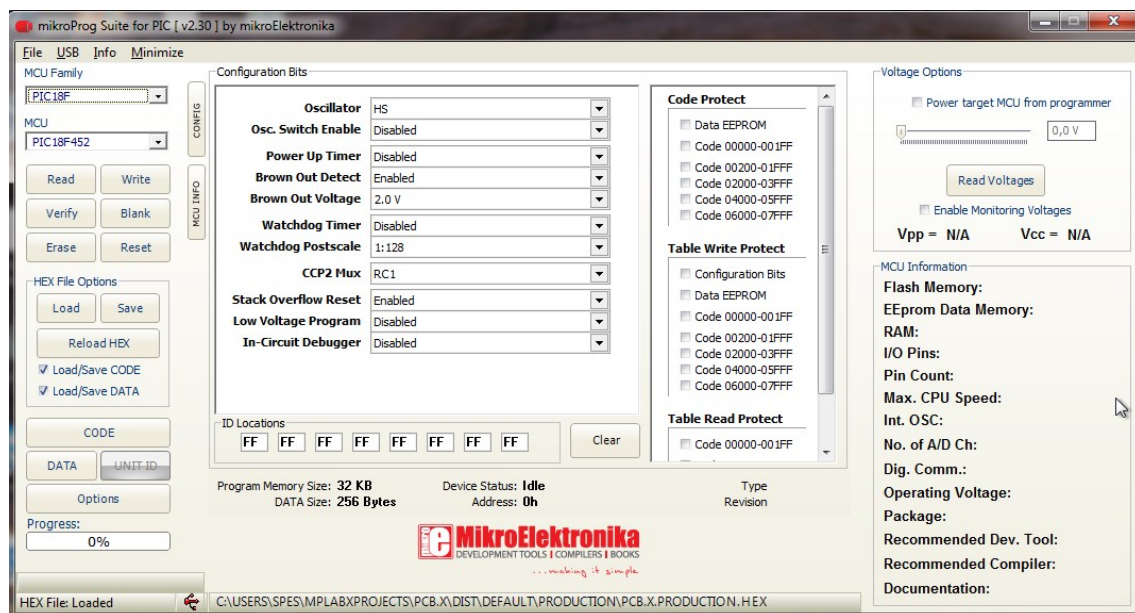


Fig. 5.7: Schermata programmatore "mikroProg Suite For PIC".

### 5.4 Nozioni base sulla programmazione di PIC

Prima d'iniziare a scrivere il programma, é necessario sapere come si programmi un PIC, quali funzioni possa svolgere e come poterle implementare.

Come già accennato, ogni PIC interagisce con il mondo esterno tramite dei pin di collegamento ed ognuno di questi può svolgere diverse funzioni; per impostare il funzionamento di ogni pin e tutti i parametri ad esso collegati, sono presenti delle zone di memoria (registri) suddivisi in base ai diversi pin e/o alle diverse funzioni. Essendo il PIC una "macchina" binaria, utilizza un linguaggio basato esclusivamente su due livelli (alto e basso<sup>1</sup>); ogni registro conterrà quindi delle "celle" da un bit, ognuna delle quali riferita ad uno scopo ben preciso. Ogni registro é composto da otto celle, questo per avere ogni registro della dimensione di 1 byte e poterlo definire in modo piú veloce utilizzando la notazione esadecimale; l'impostazione di ogni registro dev'essere effettuata per mezzo del programma che si va ad implementare.

<sup>1</sup>il PIC18F452 utilizza come livello alto,  $V_{cc}=5V$ , e come livello basso,  $GND=0V$ .

### 5.4.1 Porte I/O

Nel PIC18F452, i pin disponibili ad essere utilizzati come I/O sono raggruppati in “porte” da 8 pin, denominate A, B, C, D ed una porta da 4 denominata E; questi pin non sono tutti uguali, alcuni di loro sono infatti predisposti per un uso particolare (come già accennato nel Par. 4.4.1).

Ogni porta utilizza tre registri per il suo interfacciamento:

- registro TRIS va utilizzato per impostare la funzione di un pin come ingresso (1) o uscita (0).
- registro LAT per impostare lo stato del pin come alto (1) o basso (0).
- registro PORT per leggere lo stato del pin.

Essendo i pin raggruppati in porte da 8, la sintassi per questi tre comandi offre la possibilità d'impostare con un unico comando tutta la porta utilizzando la notazione esadecimale per riempire un byte di memoria dedicato alla funzione dei pin o al loro stato; `TRISA=0x3B` sta per esempio a significare che gli 8 bit che impostano la funzione I/O per la porta A vanno riempiti con 00111011, il che significa che i pin A2,A6,A7 sono da considerare uscite, gli altri invece come ingressi.

Se si vuole invece impostare un solo pin, la sintassi da utilizzare é per esempio `LATD-bits.LATD6=1` per impostare il pin RD6 al livello alto.

### 5.4.2 Interrupt

Un programma che gestisce una situazione fisica reale, deve spesso trattare degli eventi asincroni, dato che il programma principale non può restare in attesa di un segnale che non si sa quando arriverá, queste situazioni sono generalmente affrontate fornendo la possibilità d'interrompere il codice in qualsiasi momento per eseguire parti di codice specifiche; data questa loro caratteristica questi eventi prendono il nome di “interrupt”. Il PIC preso in considerazione é in grado di gestire due livelli di priorità per gli interrupt (alta o bassa), in questo modo, nell'eventualità che accada un interrupt mentre é già in atto una routine per un interrupt precedente, il PIC é in grado d'interrompere anche questa procedura ed attivare quella per il nuovo interrupt nel caso questo sia piú prioritario; una volta gestito l'evento ad altra priorità ritornerà all'interrupt precedente ed infine al normale flusso di programma.

#### Funzionamento

Un interrupt hardware (un segnale che viene generato da una periferica esterna al PIC) deve avere la possibilità di essere intercettato ed analizzato immediatamente, per questo motivo i pin adibiti alla gestione di interrupt hanno una struttura diversa dai comuni pin di I/O. Il particolare collegamento interno verso il PIC, permette a quest'ultimo di vedere istantaneamente una variazione di stato in questi pin ed attivare la specifica routine (ISR Interrupt Service Routine) preparata per quel particolare evento. Una volta che viene rilevato un interrupt il programma dev'essere quindi in grado di:

1. interrompere il normale svolgimento del programma principale
2. riconoscere quale periferica ha generato l'interrupt
3. eseguire la rispettiva ISR
4. riprendere il programma principale da dov'era stato interrotto

I punti 1 e 4 sono completamente gestiti dal PIC e dal compilatore utilizzando una zona di memoria per salvare e ricordare il punto dove si é interrotto il normale flusso del programma, per poi porlo riprendere; il resto invece é a carico di chi scrive il programma, bisogna infatti fare in modo che il PIC riconosca quale interrupt si sia verificato e scrivere tutte le ISR necessarie.

### Registri interrupt

Per la gestione degli interrupt ci sono dieci registri da considerare: RCON, INTCON, INTCON2, INTCON3, PIR 1, PIR2, PIE 1, PIE2, IPR1, IPR2.

Ogni interrupt necessita in linea generale di tre bit per poter essere gestito e controllato:

- bit “flag” per indicare che l’interrupt é avvenuto
- bit “enable” per abilitare o meno l’esecuzione della routine dedicata.
- bit “priority” per impostare il tipo di prioritá

vediamo ora dove sono collocate queste informazioni.

**RCON** Il registro RCON contiene i bit utilizzati per abilitare l’uso della prioritá degli interrupt

R/W-0	U-0	U-0	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

Fig. 5.8: Registro RCON.

In questo registro il bit utilizzato é:

IPEN:

- 1= le prioritá vengono abilitate
- 0= le prioritá non vengono disabilitate.

**INTCON** I registri INTCON contengono alcuni bit di flag e di prioritá

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMROIE	INTOIE	RBIE	TMROIF	INTOIF	RBIF
bit 7							bit 0

Fig. 5.9: Registro INTCON.

In questo registro i bit utilizzati sono:

GIE/GIEH:

quando  $IPEN=0$

- 1=abilita tutti gli interrupt
- 0=disabilita tutti gli interrupt

quando  $IPEN=1$

- 1=abilita tutti gli interrupt ad alta prioritá
- 0=disabilita tutti gli interrupt

PEIE/GIEL:

quando  $IPEN=0$

- 1=abilita tutti gli interrupt

0=disabilita tutti gli interrupt  
 quando IPEN=1  
 1=abilita tutti gli interrupt a bassa priorit a  
 0=disabilita tutti gli interrupt a bassa priorit a  
 INT0IE:  
 1=abilita la ricezione dell'interrupt sul pin INT0  
 0=disabilita la ricezione dell'interrupt sul pin INT0  
 INT0IF:  
 1= segnala che l'interrupt INT0   avvenuto (dev'essere poi portato a 0 dal software)  
 0= segnala che l'interrupt INT0 non   avvenuto

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

Fig. 5.10: Registro INTCON2.

In questo registro i bit utilizzati sono:

INTEDG0:  
 1=interrupt INT0 avviene sul fronte di salita  
 0=interrupt INT0 avviene sul fronte di discesa  
 INTEDG1:  
 1=interrupt INT1 avviene sul fronte di salita  
 0=interrupt INT1 avviene sul fronte di discesa

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

Fig. 5.11: Registro INTCON3.

In questo registro i bit utilizzati sono:

INT1IP:  
 1=l'interrupt INT1   di alta priorit a  
 0=l'interrupt INT1   di bassa priorit a  
 INT1IE:  
 1=l'interrupt INT1   abilitato  
 0=l'interrupt INT1   disabilitato  
 INT1IF:  
 1= l'interrupt INT1   avvenuto (dev'essere poi portato a 0 dal software)  
 0= l'interrupt INT1 non   avvenuto

**PIR** I registri PIR contengono i bit di segnalazione per gli interrupt delle periferiche

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

Fig. 5.12: Registro PIR1.

In questo registro il bit utilizzato é:

RCIF:

1=il buffer di ricezione della USART é pieno (viene azzerato una volta letto)

0=il buffer di ricezione della USART é vuoto

**PIE** I registri PIE contengono i bit di abilitazione per gli interrupt delle periferiche

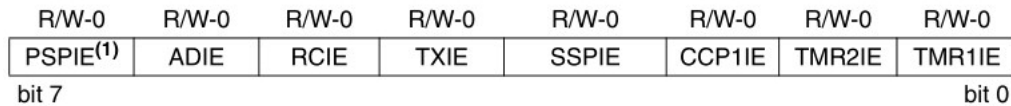


Fig. 5.13: Registro PIE1.

In questo registro il bit utilizzato é:

RCIE:

1=l'interrupt dalla periferica USART é abilitato

0=l'interrupt dalla periferica USART é disabilitato

**IPR** I registri IPR contengono i bit di prioritá per gli interrupt delle periferiche

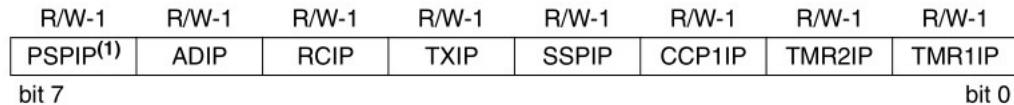


Fig. 5.14: Registro IPR1.

In questo registro il bit utilizzato é:

RCIP:

1=l'interrupt di USART é di alta prioritá

0=l'interrupt USART é di bassa prioritá

### 5.4.3 Oscillatore

Il PIC18F452 necessita di un segnale di clock di riferimento con il quale lavorare, per questo motivo ha al suo interno un oscillatore che, in questo caso, viene stabilizzato da un quarzo esterno da 8Mhz; oltre a collegare fisicamente il quarzo ai pin predisposti, bisogna però dichiarare in fase di programmazione alcuni parametri per configurare correttamente anche questo aspetto.

Bisogna innanzitutto dichiarare la tipologia di oscillatore utilizzato, osservando il manuale vediamo che siamo nel caso di un High Speed Crystal; tale dichiarazione viene effettuata configurando in modo appropriato i bit FOSC2, FOSC1 e FOSC0 nel registro CONFIG1H; in alternativa il programma MPLAB offre la possibilitá di dichiarare in modo molto semplice quale oscillatore si stia utilizzando (con la riga di comando `#pragma config OSC=HS`), provvedendo poi lui ad impostare correttamente i registri.

La definizione esatta della frequenza dell'oscillatore dev'essere invece dichiarata utilizzando la riga di comando:

```
#define _XTAL_FREQ 8000
```

#### 5.4.4 Modulo seriale (USART)

La comunicazione seriale é affidata nel PIC, al modulo USART (Universal Synchronous Asynchronous Receiver Transmitter) configurabile in diversi modi a seconda delle particolari necessità di ogni progetto.

Il modulo USART dev'essere configurato in modo da interfacciarsi correttamente con la periferica alla quale viene collegato (un pc nella maggior parte dei casi), una volta ricevuto un messaggio, questo viene memorizzato in una zona di memoria dedicata in attesa che ne venga letto il contenuto.

La configurazione serve per assicurare gli stessi tempi di trasmissione e ricezione tra i dispositivi; se questi non utilizzano lo stesso baud rate, cioè la velocità di trasmissione e ricezione, il modulo USART non sarà in grado d'interpretare correttamente i caratteri inviati.

La gestione del modulo USART é affidata alle funzioni contenute nella libreria "usart.h" fornito da MPLAB ed utilizzabile con il compilatore C18

#### 5.4.5 Convertitore A/D

La possibilità di convertire una tensione analogia in un riferimento numerico digitale é un aspetto spesso di fondamentale importanza, il PIC18F452 rende possibile questa conversione utilizzando un circuito interno dedicato. Anche questo modulo necessita di essere configurato per poter effettuare la conversione in modo corretto. La gestione del convertitore A/D avviene utilizzando le funzioni contenute nella libreria "adc.h"

### 5.5 Stesura del codice

Il software che si vuole realizzare per programmare il PIC, deve tener conto di tutto quello visto nei capitoli precedenti e gestire in modo appropriato l'hardware a disposizione sulla base delle richieste da soddisfare.

#### 5.5.1 Struttura generale

Il PIC sono pensati e costruiti per eseguire il programma caricato nella loro memoria in modo ciclico, dal momento in cui vengono accesi a quando vengono spenti. Sulla base di questa premessa il programma dev'essere strutturato in modo da compiere, ad ogni ciclo, esclusivamente le operazioni necessarie in quel dato momento, tralasciando la maggior parte del codice; in questo modo si ottiene un programma molto reattivo, pronto ad ogni possibile evento ed in grado d'agire di conseguenza.

Questo tipo di struttura viene creata utilizzando delle "aree" di codice alle quali il programma può accedere esclusivamente quando sono soddisfatte particolari condizioni; il programma valuterá se entrare in ogni specifica area, verificando il valore di particolari variabili utilizzate per tale scopo (variabile di Flag).

Il primo importante aspetto da considerare per questo progetto, é la necessità di rilevare eventi esterni (finecorsa, pulsanti ecc.), per i quali dev'essere possibile l'identificazione immediata senza la necessità di attendere l'inizio di un nuovo ciclo. Per questo motivo, come già visto in 5.4.2, sono disponibili le funzionalità di interrupt; la rilevazione di un interrupt consente d'interrompere il normale scorrere del ciclo per effettuare delle operazioni dedicate (ISR), al termine di queste il ciclo ricomincia da dov'era stato interrotto.

Come già accennato, all'interno del ciclo principale, bisogna organizzare delle aree specifiche per ogni operazione. Seguendo quanto visto finora la struttura generale del codice si può quindi inizialmente vedere come in Fig. 5.15.

Per consentire al programma di poter eseguire le porzioni di codice dedicate agli interrupt, esso deve ripetersi ciclicamente il più spesso possibile, passando quindi per la sezione di gestione degli interrupt; per garantire questo, si è deciso di utilizzare meno cicli (for e while) possibili, gestendo tutte le situazioni con verifiche di tipo if, in questo modo si ottiene un programma che non rimane bloccato in un ciclo per troppo tempo.

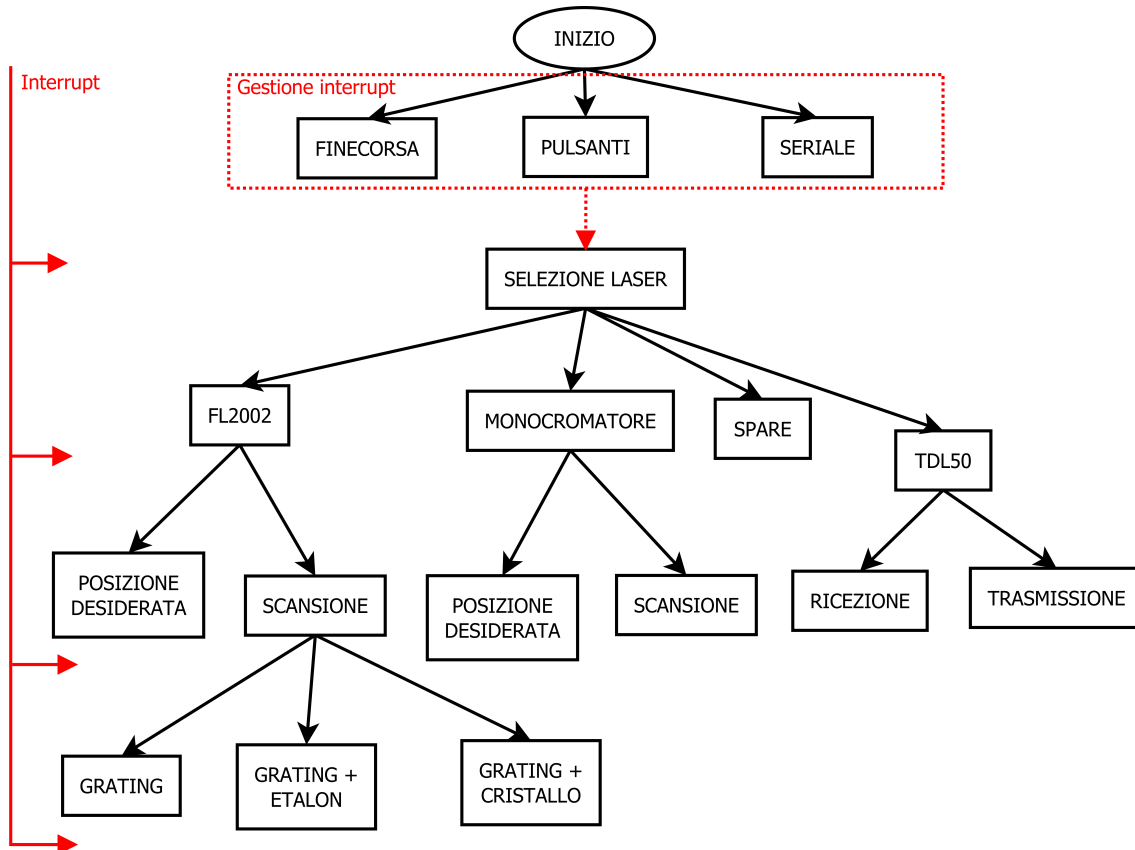


Fig. 5.15: Funzionamento del monocromatore.

### 5.5.2 Operazioni preliminari

Il linguaggio C utilizza delle regole ben precise per la scrittura di un programma, alcune istruzioni devono infatti essere scritte prima del programma principale per consentire al compilatore d'interpretare correttamente il codice. In questo caso vengono forniti:

1. librerie da importare
2. definizioni macro utilizzate per snellire il programma e renderlo più leggibile
3. impostazioni per il programmatore
4. prototipi delle funzioni
5. definizione delle variabili utilizzate
6. routine di interrupt (ISR)



Dopo aver fornito queste informazioni si é pronti per la stesura del main program, il quale contiene al suo interno molte funzioni che possono essere invocate ogniqualvolta sia necessario, moltissime di queste (le piú comunemente utilizzate) sono già scritte e pronte all'utilizzo; esse sono suddivise per categorie e raggruppate in “librerie”. Tutto ciò che non si trova in queste librerie e di cui si ha necessità dev'essere realizzato ad hoc ed inserito in modo opportuno nel codice. Prima di analizzare il main program si propone quindi una panoramica delle funzioni utilizzate per la stesura del codice; la maggior parte di esse sono di uso comune e si riporta un veloce riassunto del loro scopo, una descrizione piú accurata viene invece riservata alle funzioni molto particolari o che hanno necessità di essere modificate.

### 5.5.3 Funzioni di libreria

Le librerie preesistenti per la programmazione in C devono essere la prima cosa riportata nel codice, la sintassi da utilizzare é:

```
#include <nomelibreria.h>
```

Le librerie utilizzate<sup>2</sup> per questo programma sono:

#### pic18f452.h

Questa libreria rende possibile l'interfacciamento con il PIC utilizzato, grazie ad essa é possibile utilizzare una sintassi molto intuitiva che verrà poi convertita in numeri di registri e linguaggio macchina per il PIC.

#### delays.h

Questa libreria contiene le funzioni necessarie ad inserire delle pause all'interno del programma. Le funzioni a disposizione consentono di inserire pause della durata di n-istruzioni elementari, dove 1 istruzione elementare equivale a 4 colpi di clock. Le funzioni disponibili sono riportate in Fig. 5.16

Function	Description
Delay1TCY	Delay one instruction cycle.
Delay10TCYx	Delay in multiples of 10 instruction cycles.
Delay100TCYx	Delay in multiples of 100 instruction cycles.
Delay1KTCYx	Delay in multiples of 1,000 instruction cycles.
Delay10KTCYx	Delay in multiples of 10,000 instruction cycles.

Fig. 5.16: Elenco delle funzioni disponibili nella libreria “delays.h”.

La x, dove presente, é il fattore moltiplicativo e può essere un numero tra 1 e 255.

#### adc.h

Questa libreria contiene le funzioni necessarie alla gestione del modulo A/D; in Fig. 5.17 sono riportate tutte le funzioni disponibili.

Tra queste funzioni, l'unica che necessita di una spiegazione approfondita é OpenADC, utilizzata per configurare il modulo ADC.

<sup>2</sup>oltre alle classiche (stdlib.h,stdio.h,math.h,string.h)

Function	Description
BusyADC	Is A/D converter currently performing a conversion?
CloseADC	Disable the A/D converter.
ConvertADC	Start an A/D conversion.
OpenADC	Configure the A/D convertor.
ReadADC	Read the results of an A/D conversion.
SetChanADC	Select A/D channel to be used.

Fig. 5.17: Elenco delle funzioni disponibili nel pacchetto “adc.h”.

**OpenADC** Il prototipo di questa funzione é:

```
void OpenADC(unsigned char config, unsigned char config2);
```

Nell’argomento **config** si devono specificare le seguenti categorie:

- A/D velocità di conversione rispetto al clock
- giustificazione della parola risultato della conversione (a destra o a sinistra)
- pin utilizzato per la tensione d’ingresso

Nell’argomento **config2** si devono specificare le seguenti categorie:

- canale utilizzato come ingresso
- abilitazione dell’interrupt

Nel nostro caso la funzione viene invocata nel seguente modo:

```
OpenADC(ADC_FOSC_2&ADC_RIGHT_JUST&ADC_1ANA_0REF, ADC_CH0&ADC_INT_OFF);
```

dove:

- ADC\_FOSC\_2 é la frequenza utilizzata vale FOSC/2
- ADC\_RIGHT\_JUST la conversione é giustificata a destra
- ADC\_1ANA\_0REF l’unico pin d’ingresso analogico é A0 e non ci sono riferimenti esterni, la conversione avviene cioè portando il range 0-5V nel range 0-1023
- ADC\_CH0 il canale utilizzato é il canale 0
- ADC\_INT\_OFF interrupt dell’ADC disabilitati

## LCD44780.h[11]

Questa libreria contiene le funzioni necessarie alla gestione di un display LCD basato sul chip HD44780U , le funzioni di questa libreria che sono state utilizzate sono:

- OpenLCD (); per inizializzare il display, non occorre fornire alcun parametro.
- ClearLCD (); per cancellare tutto ciò che é scritto sul display
- CursorLCD (char1,char2); se char1=1 apre il cursore, se =0 non apre, se char2 =1 il cursore lampeggia, se =0 non lampeggia.
- WriteVarLCD(char \*buffer); per scrivere una stringa sul display
- HomeLCD (); per posizionare il cursore nel primo carattere della prima riga.
- Line2LCD (); per posizionare il cursore nel primo carattere della seconda riga.
- ShiftCursorLCD (char); se char=0 il cursore si sposta di un carattere a sinistra, se =1 si sposta a destra.

**usart.h**

Questo pacchetto contiene le funzioni necessarie alla gestione del modulo USART; in Fig. 5.18 sono riportate tutte le funzioni disponibili.

Function	Description
BusyUSART	Is the USART transmitting?
CloseUSART	Disable the USART.
DataRdyUSART	Is data available in the USART read buffer?
getcUSART	Read a byte from the USART.
getsUSART	Read a string from the USART.
OpenUSART	Configure the USART.
putcUSART	Write a byte to the USART.
putsUSART	Write a string from data memory to the USART.
putrsUSART	Write a string from program memory to the USART.
ReadUSART	Read a byte from the USART.
WriteUSART	Write a byte to the USART.
baudUSART	Set the baud rate configuration bits for enhanced USART.

Fig. 5.18: Elenco delle funzioni disponibili nel pacchetto “usart.h”.

**OpenUSART** Il prototipo di questa funzione é:

```
void OpenUSART(unsigned char config, unsigned int spbrg);
```

Nell'argomento **config** si devono specificare le seguenti categorie:

- abilitazione interrupt in trasmissione
- abilitazione interrupt in ricezione
- modo di utilizzo del modulo (Sincrono, Asincrono)
- lunghezza carattere (8 o 9 bit)
- selezione del metodo (slave o master) <sup>3</sup>
- modo di ricezione (singolo o continuo)
- Tipologia di baud rate (alto o basso)

L'argomento **spbrg** é un valore legato al baud rate al quale lavora la USART; tale valore é da calcolare utilizzando la formula la formula 5.5.1.

$$\text{baud rate} = \frac{FOSC}{64 * (spbrg + 1)} \quad (5.5.1)$$

dove:

- il baud rate scelto per quest'applicazione é pari a 1200, tale valore é stato scelto perché già utilizzato dal laser TDL50.
- FOSC si ricorda essere la frequenza dell'oscillatore che vale 8 Mhz.

<sup>3</sup>solo per il modo sincrono

Il valore di `spbrg` vale quindi, in questo caso, 103.

Nel nostro caso la funzione viene invocata nel seguente modo:

```
OpenUSART(USART_TX_INT_OFF&USART_RX_INT_OFF&USART_ASYNC_MODE&
          USART_EIGHT_BIT&USART_CONT_RX&USART_BRGH_LOW, 103);
```

dove:

- `USART_TX_INT_OFF` interrupt in trasmissione disabilitato
- `USART_RX_INT_OFF` interrupt in ricezione disabilitato
- `USART_ASYNC_MODE` modo Asincrono
- `USART_EIGHT_BIT` lunghezza di carattere 8 bit
- `USART_CONT_RX` metodo di ricezione continuo
- `USART_BRGH_LOW` baud rate basso
- 103 ottenuto tramite l'equazione 5.5.1

### sw\_uart

Il PIC18F452 ha al suo interno un solo modulo USART, ciò vuol dire che può gestire una sola porta seriale, avendo la necessità di utilizzare due porte si è scelto di utilizzare il pacchetto “sw\_uart.h” che di fatto simula un modulo USART chiamato quindi “software UART” (la S viene persa perché in questo caso non è possibile utilizzare la modalità sincrona).

In Fig. 5.19 sono riportate tutte le funzioni disponibili nella libreria sw\_uart.h.

Function	Description
getcUART	Read a byte from the software UART.
getsUART	Read a string from the software UART.
OpenUART	Configure I/O pins for use as a UART.
putcUART	Write a byte to the software UART.
putsUART	Write a string to the software UART.
ReadUART	Read a byte from the software UART.
WriteUART	Write a byte to the software UART.

Fig. 5.19: Elenco delle funzioni disponibili nel pacchetto “sw\_uart.h”.

Osservando la lista delle funzioni, si nota subito che, al contrario del modulo USART hardware, non è possibile verificare la presenza di informazioni in arrivo; i due moduli dovranno quindi essere gestiti in modo diverso.

Anche la configurazione avviene in modo diverso, utilizzando la funzione `OpenUART()`; le informazioni necessarie non sono fornite come argomenti alla funzione ma devono essere inserite negli appositi file: “sw\_uart.c” e “sw\_uart.h”. I pin di trasmissione e ricezione non sono vincolati come nel modulo hardware e possono essere scelti tra tutti i pin di semplice I/O, altri parametri da impostare sono alcuni tempi di attesa che le funzioni utilizzano per riuscire a leggere e scrivere in modo corretto, le formule da utilizzare per il calcolo di questi tempi sono riportate in Fig. 5.20

Durante la programmazione si è però riscontrato che i risultati dati delle formule non sempre producevano un corretto funzionamento del modulo, per questo motivo si è resa necessaria una taratura manuale dei ritardi facendo uso di un oscilloscopio.

Function	Behavior
DelayTXBitUART	Delay for: $(((((2 * F_{osc}) / (4 * \text{baud})) + 1) / 2) - 12)$ cycles
DelayRXHalfBitUART	Delay for: $(((((2 * F_{osc}) / (8 * \text{baud})) + 1) / 2) - 9)$ cycles
DelayRXBitUART	Delay for: $(((((2 * F_{osc}) / (4 * \text{baud})) + 1) / 2) - 14)$ cycles

Fig. 5.20: Formule per il calcolo dei ritardi per il modulo UART.

### 5.5.4 Routine di Interrupt (ISR)

Nel momento in cui si verifica un interrupt, il normale flusso del programma s'interrompe e vengono avviate le funzioni ISR. La sintassi per queste particolari funzioni prevede la stesura di una funzione per la gestione degli interrupt a bassa priorit  ed una per quelli ad alta priorit ; viste le necessit  del programma da implementare, si   deciso d'inserire la rilevazione dei pulsanti premuti, tra gli interrupt a bassa priorit  mentre i finecorsa ed il modulo USART, in quelli ad alta (data la loro importanza, questi due eventi devono infatti avere la priorit  su tutto).

Le funzioni ISR non devono essere troppo complicate e non devono contenere richiami ad altre funzioni complesse, per questo motivo si   scelto di far modificare alla funzione una variabile assegnandole un numero a seconda dell'interrupt trovato, all'inizio di ogni ciclo il programma controller  tale variabile ed invocherà le funzioni necessarie.

#### low\_priority

L'unico interrupt che questa funzione deve gestire   la pressione di uno dei pulsanti i quali sono collegati al pin B1 che corrisponde all'interrupt 1. Quando un pulsante viene premuto, il bit di flag associato all'interrupt 1 viene portato a livello logico 1, la funzione controlla quindi questo evento verificando la variabile associata ad esso con il comando `if(INTCON3bits.INT1IF==1)`. Nel caso questa condizione sia vera bisogner  capire successivamente quale pulsante sia stato premuto, per questo motivo la funzione imposta la variabile `q` al valore 1 e resetta il bit di flag impostandolo al valore 0.

#### high\_priority

Questa funzione deve gestire due diversi tipi di interrupt e quindi   divisa in due parti, la prima parte verifica se   in arrivo un carattere dal modulo USART mentre la seconda verifica se si   verificato un finecorsa.

1. La prima verifica viene effettuata sul bit di flag relativo al modulo USART (`PIR1bits.RCIF`), se tale bit   a livello alto il PIC legge il carattere preparato in memoria e lo salva nella variabile *message*. A questo punto si passa a verificare quale carattere sia stato premuto; la maggior parte di essi vanno infatti a formare le stringhe di comunicazione, alcuni saranno invece associati a richieste particolari ed andranno a modificare la variabile `q` con un valore che avr  un significato particolare nel programma principale (Tab. 5.1). Il bit di flag viene infine resettato.
2. La seconda verifica viene effettuata sul bit di flag relativo ai finecorsa (`INTCONbits.INTOIF`), se tale bit   a livello alto vuol dire che un finecorsa   appena scattato quindi per prima cosa la funzione abilita il reset dei driver per impedire a qualsiasi cosa di muoversi, l'analisi di quale sia il finecorsa attivato verr  fatta successivamente, per questo motivo la variabile `q` viene impostata al valore 2 ed infine il bit di flag viene resettato.

carattere	comando
q	reset abilitato
i	q=4
r	reset disabilitato
w	q=3
INVIO	parola_pronta=1
altro carattere	memorizza il carattere

Tab. 5.1: Funzione eseguita alla pressione di un carattere.

### 5.5.5 Funzioni personalizzate

Vengono ora descritte le funzioni create ad hoc per il progetto, riportando una spiegazione concettuale del loro funzionamento ed i passaggi principali.

#### **trova\_finecorsa**

Come descritto in 4.4.3, quando si verifica un evento “finecorsa”, é necessario capire quale in particolare sia stato attivato; per far questo é necessario impostare i pin di selezione del multiplexer per verificare uno alla volta tutti gli ingressi.

All’interno di un ciclo while che continua finché non é stato trovato il finecorsa, é implementato un ciclo di tipo switch-case che ad ogni ciclo imposta la combinazione di selezione corrispondente ad un preciso finecorsa e scrive quest’ultimo nella variabile *find\_finecorsa*; prima di ricominciare il ciclo while, viene effettuata una verifica sul pin d’uscita, in caso questo sia uguale a zero la ricerca é terminata e la funzione restituisce in uscita il numero corrispondente al finecorsa trovato.

#### **trova\_pulsante**

Come per i finecorsa, é necessaria una funzione per capire quale pulsante/levetta sia stato attivato. Concettualmente bisogna eseguire gli stessi passaggi descritti per i finecorsa, con la differenza sostanziale che quando un pulsante viene premuto, il segnale ad esso collegato e quello di interrupt scendono a livello logico zero per il solo tempo in cui questo resta premuto, per questo motivo nel ciclo while dev’essere considerato che B1 sia ancora uguale a zero, altrimenti il ciclo continuerebbe a cercherebbe inutilmente. Al termine della ricerca la funzione restituisce in uscita il numero corrispondente al pulsante premuto. Il passaggio per tutti i possibili pulsanti avviene in brevissimo tempo grazie alla velocità del PIC ma può essere che a volte il pulsante resti premuto per un tempo troppo breve per essere rilevato, in questo caso la funzione esce dal ciclo senza aver trovato una corrispondenza ed é come se non si fosse premuto il tasto.

#### **passo**

Il comando per eseguire un passo viene dato ad un motore attraverso il demultiplexer, il quale abilita un particolare driver sulla base della combinazione dei suoi pin di selezione. Si é deciso di creare una funzione per ogni motore, la quale, ogni volta che viene invocata, faccia fare un passo al motore corrispondente; a tal scopo le funzioni, una volta invocate, impostano inizialmente la combinazione relativa al motore corrispondente(vedi Fig. 4.8) e successivamente quella associata a nessun motore. La sequenza con cui i pin vengono

impostati é di fondamentale importanza, i pin infatti non vengono cambiati tutti e tre simultaneamente, se dovesse capitare che per raggiungere una determinata sequenza, si passa per la sequenza relativa ad un altro motore, anche quest'ultimo eseguirá un passo.<sup>4</sup> La funzione viene invocata prendendo in ingresso un parametro (*velocit*) corrispondente ad un numero intero, tale numero serve per generare una piccola pausa tra un passo e l'altro; dopo la definizione di ogni terna di selezione, segue infatti un tempo di attesa generato dal comando `Delay1KTCYx(velocit)` in questo modo, al variare della variabile *velocit* si puó controllare la velocità di rotazione del motore.

Per poter conoscere la posizione del motore durante l'esecuzione del programma, ogni funzione tiene infine conto di quanti passi sono stati fatti dal proprio motore incrementando o decrementando la variabile *passi* ad esso associata.

Nel caso dei motori del grating e del monocromatore, oltre al numero di passi é bene tener conto della lunghezza d'onda associata alla posizione corrente, questo viene fatto aggiungendo o sottraendo alla lunghezza d'onda di partenza, la variazione della lunghezza d'onda d'uscita provocata da un passo del motore (visto in 5.2).

Per favorire l'utilizzo della scheda con programmi Matlab già in uso nel laboratorio, le funzioni passo inviano tramite porta seriale un carattere "U" ogni volta che eseguono un passo.

### taratura

Durante l'esecuzione del programma, l'operatore puó prendere delle decisioni od impostare alcuni parametri direttamente dal pannello mediante l'utilizzo dei pulsanti (su, giù, destra, sinistra, ok); si é deciso di fare in modo che tutte queste possibilità siano riconducibili alla modifica di un numero. É quindi necessaria una funzione da invocare all'interno del codice quando é richiesto l'intervento dell'operatore.

La funzione riceve in ingresso un puntatore che punta alla variabile numerica da modificare<sup>5</sup>, una volta invocata, la funzione ha al suo interno una serie di condizioni che verificano se é stato premuto uno tra i pulsanti su, giù, destra, sinistra. La funzione utilizza due variabili dedicate: *n\_riga* per memorizzare quale posizione (cifra) del numero sia al momento selezionata, *n* per memorizzare quale numero si voglia inserire in tale posizione.

Se il tasto premuto é destra o sinistra, la funzione incrementa o decrementa la variabile *n\_riga* per spostare la posizione da modificare, allo stesso tempo per facilitare la comprensione dell'operatore, il cursore lampeggiante del display si sposta di conseguenza.

Se invece il tasto premuto é su o giù la funzione incrementa o decrementa la variabile *n* e copia tale valore nella posizione attualmente selezionata. Al termine della funzione la variabile *aggiornato* viene impostata al valore 1 per permettere al programma principale di capire che una modifica é avvenuta.

### scrivere LCD

Avendo la necessità di scrivere spesso sul display LCD, si é deciso di utilizzare una funzione, in questo modo il programma diventa piú leggibile ma soprattutto si risparmiano righe di codice e di conseguenza spazio di memoria nel PIC.

La funzione riceve in ingresso due stringhe (*a*, *b*) di lunghezza non definita, come prima cosa si occupa di cancellare le precedenti scritte sull'LCD e di posizionarsi all'inizio della prima riga, a questo punto scrive la stringa *a* sulla prima riga e la *b* sulla seconda.

<sup>4</sup>Questo vale solo per i motori che sono abilitati contemporaneamente

<sup>5</sup>i puntatori sono strumenti preziosi nel linguaggio C, in questo caso permettono alla funzione di modificare una variabile esterna che non é sempre la stessa ma cambia a seconda di dove viene invocata la funzione

## levette

La possibilità di muovere i motori in modo manuale è offerta dalle tre levette presenti sul FL2002 e dalla levetta posizionata sul pannello frontale del case contenente la PCB; in più parti del codice servirà quindi abilitare la possibilità di controllo manuale che verrà gestita da questa funzione.

Essendo in alcuni casi necessario impedire il movimento di alcuni motori, la funzione riceve come parametri d'ingresso quattro numeri (0 o 1) che segnalano l'abilitazione o no alla movimentazione manuale del rispettivo motore. Ogni motore è associato ad una levetta che può avere due posizioni, la funzione vede quindi quale pulsante sia stato attivato e, nel caso il rispettivo motore sia abilitato, entra in una delle otto sezioni in cui è suddivisa. Ognuna di queste sezioni è organizzata allo stesso modo:

- imposta la direzione di rotazione
- fa compiere al motore un passo<sup>6</sup>
- entra in un ciclo while in cui il motore continua a compiere passi finché la levetta rimane premuta (`PRTBbits.RB1==0`). Fintanto che il programma rimane in questo ciclo viene invocata anche la funzione accelerazione, la quale diminuisce in modo graduale il tempo di attesa tra un passo e l'altro fino ad ottenere la velocità massima.
- imposta la variabile pulsante al valore 15 che corrisponde a “nessun pulsante”

## accelerazione

Quando si utilizzano le levette per la movimentazione manuale, è necessario poter disporre di una movimentazione molto lenta (per sistemare al meglio la posizione desiderata muovendo un passo alla volta) ma anche molto veloce (per arrivare velocemente nei pressi della posizione desiderata); non è però consigliato partire subito con una velocità troppo elevata, alcuni motori (in particolare quelli del grating e del monocromatore) rischiano infatti di “perdere passi”, per questo motivo è meglio iniziare la movimentazione in modo lento per poi aumentare gradualmente. Questa funzione viene inoltre utilizzata nella parte di codice in cui il motore deve muoversi ad una velocità sostenuta.

La funzione viene invocata passandole un puntatore alla variabile utilizzata per impostare la velocità del motore. Tale variabile viene quindi diminuita ad intervalli regolari basandosi sui passi effettuati, fino ad arrivare ad un valore che produce la velocità massima raggiungibile.

## massimizza

Nella modalità in cui si utilizza anche il cristallo è necessario massimizzare più volte l'uscita di quest'ultimo, questo viene effettuato dall'operatore manualmente azionando la levetta e trovando il giusto angolo d'incidenza. Per effettuare questo procedimento è stata creata una funzione che, quando viene invocata, chiede all'operatore di massimizzare il cristallo, abilita la levetta corrispondente e nel momento in cui viene data la conferma (con INVIO/OK) memorizza nelle due variabili *calibrazione* il numero di passi effettuati da grating e cristallo.

---

<sup>6</sup>questo serve per consentire all'operatore di compiere un passo alla volta, mettendo anche questo passo all'interno del ciclo while si rischia infatti che la levetta, azionata per un tempo molto breve, non venga identificata dalla funzione `trova_pulsante`



### interpolazione

La funzione che lega il movimento del cristallo a quello del reticolo dev'essere di volta in volta calcolata dal programma, per far questo si é scritta la funzione interpolazione che, prendendo i punti definiti dalla funzione massimizza, esegue l'interpolazione lineare e calcola i coefficienti della retta utilizzando il codice seguente.

```
for(i=0;i<5;i++){
    x=x+x_calibrazione[i];
    y=y+y_calibrazione[i];
    sqrt_xy=sqrt_xy+x_calibrazione[i]*y_calibrazione[i];
    sqrt_x=sqrt_x+x_calibrazione[i]*x_calibrazione[i];
}
//termine noto
b=((sqrt_x*y)-(sqrt_xy*x))/(5*sqrt_x-(x*x));
//coefficiente moltiplicativo di x
a=((5*sqrt_xy)-(x*y))/(5*sqrt_x-(x*x));
```

### scelta

Come già visto nella Sez. 5.2, sia nel caso si voglia utilizzare l'FL2002 nella modalità con solo il grating sia che si voglia utilizzare il monocromatore, si può voler raggiungere una determinata lunghezza d'onda oppure effettuare una scansione.

La funzione "scelta" viene utilizzata all'interno del programma per operare la scelta tra la modalità "raggiungere" e quella "scansione". Il programma associa a questa scelta la variabile *n\_operazione2* (valore uguale a 1 se si vuole fare una scansione, 2 se si vuole raggiungere una posizione), la funzione inizia quindi con lo scrivere sul display la scelta da operare ed il valore attuale della variabile; essendo il programma ciclico anche questa funzione viene interrogata continuamente finché la scelta non é stata effettuata; per non avere un continuo refresh del display si é quindi deciso che questo venga aggiornato solo nel caso in cui la variabile *operazione2* sia stata in qualche modo modificata dall'utente. La modifica di questa variabile e quindi la scelta di quale operazione selezionare, é attuabile in due modalità: attraverso la comunicazione seriale oppure attraverso i pulsanti.

Nel caso la scelta venga inviata dalla tastiera del pc, il numero, gestito inizialmente come stringa, viene convertito e copiato nella rispettiva variabile numerica; nel caso invece venga premuto un pulsante per impostare il valore da pannello, viene invocata la funzione "taratura" sulla variabile *operazione2* (e quindi aggiornato il display) e solo nel caso venga premuto il tasto OK, il valore viene convertito e copiato nella variabile numerica. Una volta operata la scelta, il programma é pronto per ricevere dall'utente le informazioni piú specifiche sulle movimentazioni da effettuare e viene invocata la funzione "inizializzazione".

### inizializzazione

Questa funzione serve per fornire al programma tutte le informazioni necessarie per poter eseguire le movimentazioni richieste dall'utente.

Per prima cosa il programma deve capire dove sia posizionato il reticolo (del FL2002 o del monocromatore), la prima informazione da fornire é quindi la lunghezza d'onda visualizzata sul contatore del dispositivo da utilizzare. La funzione scrive a monitor e sul display la domanda "dove sono?" ed attende un numero da inserire nella variabile *n\_onda\_ adesso*. Anche in questo caso, tale numero può essere inserito digitando sulla tastiera del pc collegato, oppure attraverso i pulsanti; la variabile viene gestita anche in questo caso con la funzione "taratura" illustrata in precedenza. Nel momento in cui si é inserita la lunghezza d'onda di partenza, la funzione, nel caso si voglia effettuare una scansione, chiede con lo stesso metodo utilizzato per la variabile precedente, le lunghezze d'onda di partenza (*n\_onda\_inizio*) e finale (*n\_onda\_fine*). Se si é invece nel caso di voler solo raggiungere

una certa lunghezza d'onda, la funzione chiede quale sia la destinazione e riempie con un simbolico 1 la variabile *n\_onda\_fine*<sup>7</sup>.

A questo punto la funzione deve calcolare quanti passi occorrono al reticolo per effettuare la movimentazione necessaria; la funzione continua quindi calcolando l'intervallo (in lunghezza d'onda) per raggiungere la prima posizione, e quello per effettuare la scansione.

```
n_onde_ragg=n_onda_inizio-n_onda_adesso;
n_onde_sc=n_onda_fine-n_onda_inizio;
```

Conoscendo di quanto varia la lunghezza d'onda ad ogni passo (memorizzato nella variabile *ONDSTEP*) si ricavano quindi i passi necessari (arrotondati all'intero piú vicino) per i due intervalli.

```
result=round(n_onde_ragg/ONDSTEP);
n_passi_ragg=trunc(result);
```

```
result=round(n_onde_sc/ONDSTEP);
n_passi_sc=trunc(result);
```

Per concludere, la funzione determina in quale verso dovranno ruotare i motori (UP o DOWN) basandosi sul valore positivo o negativo delle variabili *n\_passi\_ragg* e *n\_passi\_sc*.

### muovere

Si é già detto che la gestione della movimentazione per il caso dell'FL2002 con solo grating o del monocromatore sono identiche, per questo motivo si é riunita la parte di codice necessaria, nella funzione "muovere"; tale funzione é invocata passando un numero come parametro: 1 per il FL2002 o 0 per il monocromatore.

Innanzitutto, essendo la funzione destinata alla movimentazione automatica, viene disabilitato l'utilizzo delle levette per il controllo manuale impostando la variabile *NOlevette* al valore 1. Avendo la necessità di tenere il conto dei passi fatti e dovendo utilizzare la funzione per due diversi dispositivi, é necessario inizialmente definire, tramite un puntatore, a quale variabile specifica (*passi\_G* o *passi\_M*) collegare la piú generica *passi\_fatti*. A questo punto, la funzione verifica se per raggiungere il primo obiettivo, il motore debba girare in senso orario o antiorario, a seconda che la variabile *n\_passi\_ragg* sia positiva o negativa. Il motore viene quindi azionato ed utilizzando la funzione "accelerazione" aumenta via via la velocità di rotazione fino al massimo consentito per arrivare in breve tempo alla lunghezza d'onda desiderata; la variabile di conteggio dei passi viene azzerata ed il procedimento si ripete allo stesso modo fino ad arrivare alla lunghezza d'onda finale. Questa seconda movimentazione, che si effettua solo nel caso si voglia eseguire una scansione, avviene sempre alla velocità minima, per facilitare il salvataggio dei dati durante gli esperimenti.

Terminata la scansione, può essere necessario ripetere lo stesso identico esperimento per acquisire un'altra serie di dati, per questo motivo, é preferibile avere la possibilità di ripetere la movimentazione senza dover inserire ogni volta i dati. La funzione invoca quindi la funzione "ripetere".

---

<sup>7</sup>questo perché per passare alle fasi successive, la funzione verifica che le variabili diventino man mano diverse da zero

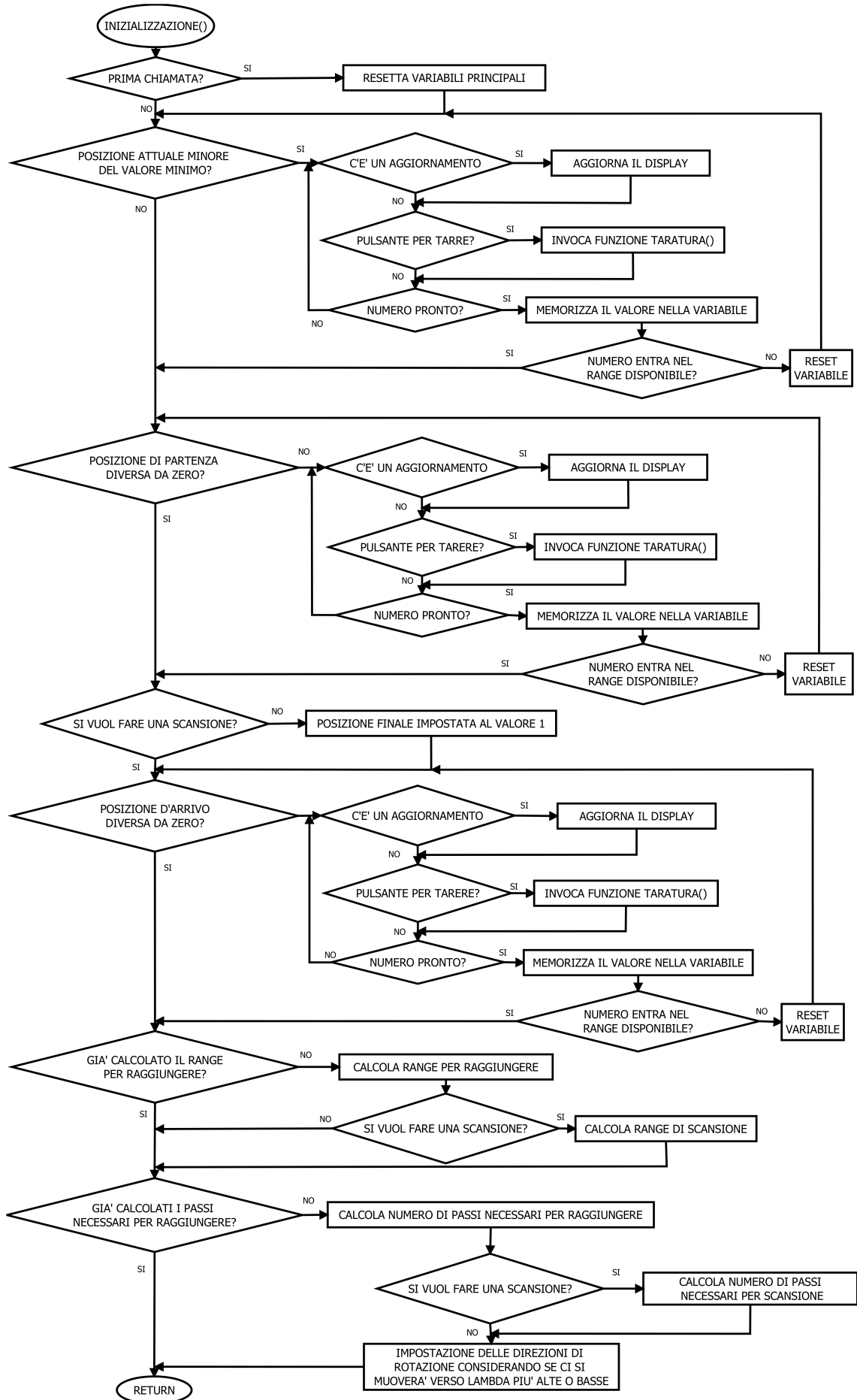


Fig. 5.21: Diagramma a blocchi della funzione inizialezione.

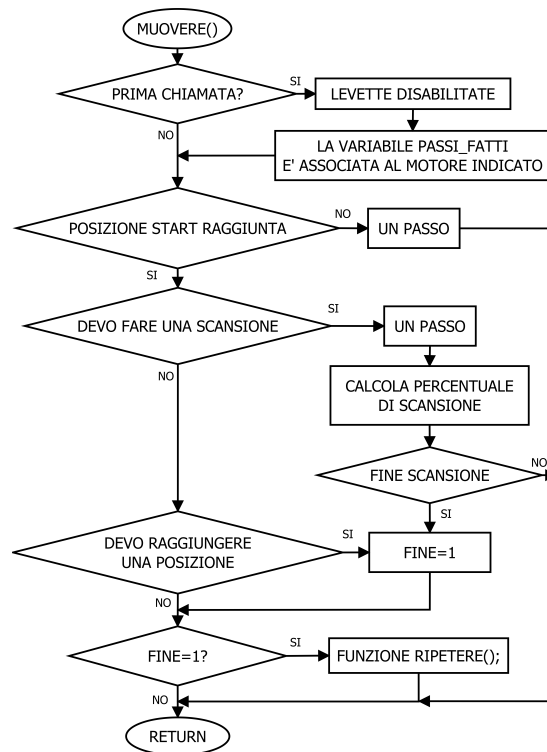


Fig. 5.22: Diagramma a blocchi della funzione muovere.

### percentuale

Per fornire all'operatore un'informazione sullo stato della scansione, si é deciso di comunicare la percentuale di movimentazione effettuata.

La funzione calcola la percentuale basandosi sui passi da compiere ( $n\_passi\_sc$ ) e su quelli effettuati ( $passi\_fatti^*$ ); il risultato viene quindi scritto sul display aggiornandolo solo nel caso il numero percentuale sia cambiato.

### ripetere

La funzione inizia con lo scrivere a monitor e sul display se si voglia ripetere l'operazione, nel caso venga premuto INVIO/OK il programma imposta quindi la variabile *rip* al valore 1 e salva nella variabile *tornare* tutti i passi che sono stati effettuati per la sola scansione, questi dati vengono immediatamente utilizzati per portare i motori nelle posizioni di partenza. A causa del gioco presente negli ingranaggi del grating, i passi per il ritorno vengono aumentati di 30 per permettere poi all'operatore di posizionarsi correttamente con l'utilizzo della levetta.

## 5.6 Programma principale

Il programma principale é suddiviso in 2 parti: una d'inizializzazione, che viene eseguita una sola volta ed una seconda parte contenuta in un ciclo while, che viene ripetuta continuamente.

### 5.6.1 Inizializzazione del programma

Il programma principale ha bisogno, prima di tutto, di definire ed inizializzare alcuni parametri, vengono quindi:

- definite le porte del PIC come ingressi o uscite a seconda delle necessità viste in precedenza
- impostati i valori di alcuni PIN (direzione, reset, selezione microstep e selezionatori dei multiplexer e demultiplexer)
- inizializzati i moduli (ADC, LCD, USART, UART)
- inizializzati i registri per abilitare e configurare correttamente gli interrupt.

Questa prima parte del programma é eseguita all'accensione del PIC ma poi, se si vorrá ricominciare da capo, sará sufficiente ripartire da questo punto, per questo motivo si é inserita qui l'etichetta INIZIO associata alla funzione goto. La prima cosa da fare a questo punto é resettare le variabili principali<sup>8</sup>; il programma passa poi a verificare se ci sia un finecorsa attivo, in caso affermativo sarebbe pericoloso procedere con le movimentazioni, il programma quindi s'interrompe e viene comunicato sul display e sul monitor quale finecorsa é attivato. Essendo il raggiungimento di un finecorsa un evento generalmente non voluto, si é scelto che sia l'operatore, con un suo intervento manuale ad alimentazione scollegata, a portare il sistema fuori dalla condizione non permessa.

La porzione di codice descritta finora dev'essere eseguita solo una volta, il resto del codice che viene ora descritto é quindi inserito in un grande ciclo while che si ripeterá fino allo spegnimento del dispositivo.

### 5.6.2 Gestione degli avvisi interrupt

La prima cosa che viene fatta all'inizio di ogni ciclo é la gestione degli avvisi interrupt, a seconda del valore assunto dalla variabile q il programma puó ora invocare le funzioni necessarie per raggiungere la particolare richiesta; il ciclo while inizia quindi con uno switch case che verifica la variabile q eventualmente modificata da uno degli interrupt.

A seconda del valore assunto dalla variabile il programma capisce quale interrupt sia avvenuto ed applica la procedura dedicata, terminata questa la variabile q viene resettata al valore 0.

Il valore della variabile q puó assumere i seguenti valori che corrispondono ad un preciso evento:

- 1: un pulsante é stato premuto, per capire quale viene invocata la funzione "trova\_pulsante" ed il numero fornito come risultato viene salvato nella variabile *pulsante*.
- 2: un finecorsa é stato raggiunto, per capire quale viene invocata la funzione "trova\_finecorsa" ed il numero fornito come risultato viene salvato nella variabile *finecorsa*.
- 3 si vuole ripartire dall'inizio, utilizzando la funzione "goto" si porta quindi il programma all'etichetta INIZIO posizionata prima delle inizializzazioni delle variabili.
- 4 si desidera avere informazioni sullo stato del sistema, in particolare su quale lunghezza d'onda sia attualmente impostato, viene quindi stampata a schermo una stringa riportante la variabile *n\_onda\_adesso*.

### 5.6.3 Conversione ADC e scelta Laser

Dopo essersi occupato degli interrupt il programma passa ad eseguire la conversione di tensione mediante il modulo ADC collegato al potenziometro, il valore della conversione viene quindi memorizzato nella variabile *vel\_pot*; ruotando il potenziometro tale valore

<sup>8</sup> questo per riportarle al valore iniziale nel caso si arrivi in questo punto rieseguendo il programma

aumenta o diminuisce, determinando la velocità dei motori (quando consentito). La porzione di codice che segue serve invece a verificare in che posizione sia stato messo il selezionatore e comunicare all'utente quale dispositivo sia al momento selezionato; il dispositivo può essere cambiato ruotando il selezionatore e premendo infine INVIO o il tasto OK per confermare la selezione; una volta effettuata la scelta questa parte non viene più ripetuta. In Fig. 5.23 è rappresentata la struttura generale del main program.

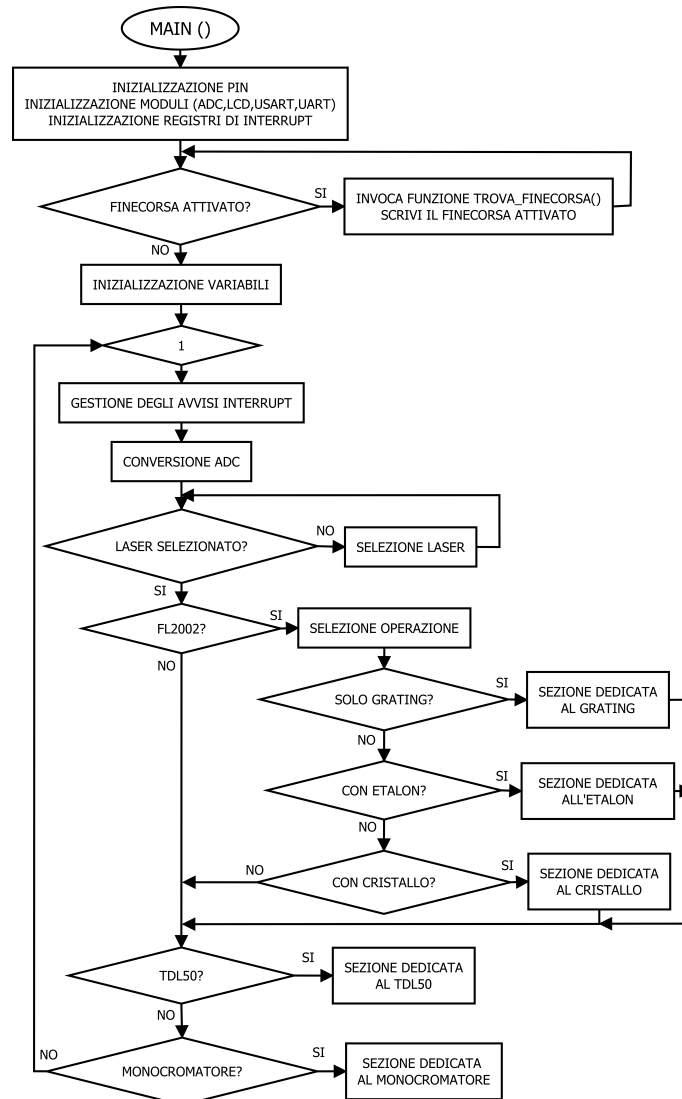


Fig. 5.23: Diagramma a blocchi del codice main.

#### 5.6.4 Codice dedicato alla gestione del Laser FL2002

Se viene selezionato il laser FL2002 il programma salva innanzitutto nella variabile OND-STEP la variazione di lunghezza d'onda dovuta ad un passo del motore del grating (0.84 centesimi di Angstrom), successivamente viene chiesto di scegliere, tramite un numero, quale modalità si vuole utilizzare (1: solo grating, 2: con etalon, 3: con cristallo). La scelta può essere fatta tramite pc o tramite pannello frontale (nel secondo caso viene utilizzata la funzione "taratura" descritta precedentemente); il numero corrispondente alla scelta viene salvato nella variabile *n\_operazione*.



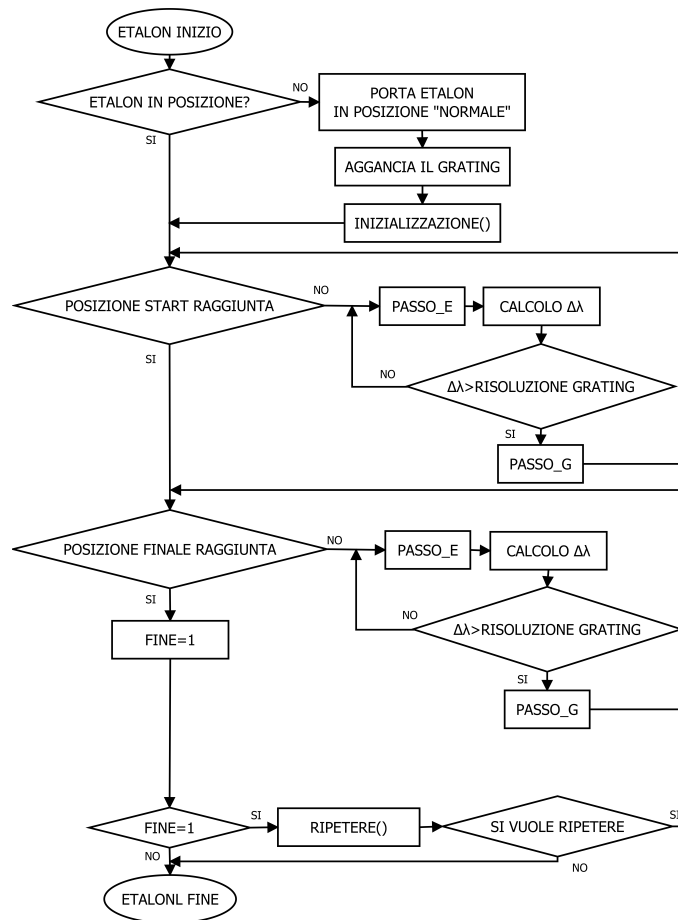


Fig. 5.25: Diagramma a blocchi della sezione di codice dedicata all'etalon.

## Cristallo

Se si sceglie la modalità con il cristallo, il programma, dopo aver operato l'inizializzazione, porta il reticolo alla posizione per iniziare la scansione ed inizia il procedimento per il calcolo della "funzione di cristallo". Il primo dei cinque punti dev'essere registrato nel punto iniziale, per questo motivo viene invocata la funzione "massimizza"; per i punti successivi il reticolo viene automaticamente ruotato in modo da compiere di volta in volta un quarto dell'intervallo di scansione, ad ogni intervallo viene quindi invocata la funzione "massimizza", per memorizzare il numero di passi effettuati dal grating e dal cristallo nella variabile  $x\_calibrazione$  e  $y\_calibrazione$ ; terminata l'acquisizione dell'ultimo punto viene invocata la funzione "interpolazione" che calcola la retta passante per i cinque punti.

Il programma a questo punto deve ritornare alla situazione di partenza per poter effettuare la scansione, per far questo porta indietro il cristallo e il reticolo tenendo conto dei passi fatti precedentemente; ricordando che il contatore del grating é soggetto ad un errore di gioco quando si effettua un'inversione del senso di rotazione, al reticolo vengono fatti fare 30 passi in piú per poi chiedere all'operatore di posizionarlo correttamente.

La scansione viene in questo caso effettuata muovendo il reticolo e calcolando, tramite la funzione appena trovata, di quanto si deve muovere il cristallo, in particolare il programma esegue il calcolo ad ogni passo del reticolo e valuta se il risultato (arrotondato) coincide con il precedente, nel caso sia diverso (sará sempre diverso al massimo di 1 poiché il cristallo deve muoversi molto meno del reticolo) viene fatto fare un passo anche al cristallo.



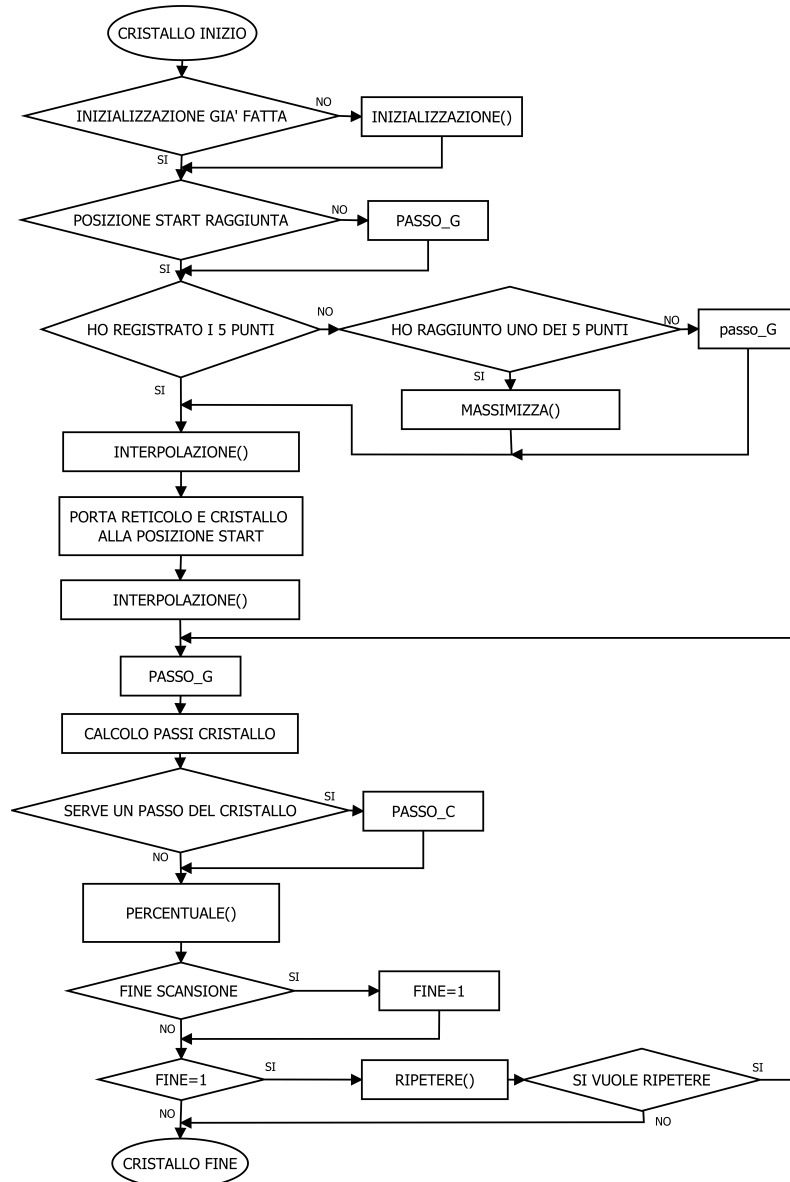


Fig. 5.26: Diagramma a blocchi della sezione di codice dedicata al cristallo.

### 5.6.5 Codice dedicato alla gestione del Monocromatore

Se viene selezionato il Monocromatore il programma abilita l'utilizzo della levetta dedicata ed imposta la variabile ONDSTEP con la variazione di lunghezza d'onda dovuta ad un passo del motore nel monocromatore (2 centesimi di Angstrom). Con la funzione "scelta" viene successivamente chiesto all'utente di selezionare se vuole effettuare una scansione o raggiungere una lunghezza d'onda; la funzione "inizializzazione" viene poi invocata per inserire le informazioni necessarie. Terminata questa procedura il programma avvia la funzione "muovere" al cui termine l'operatore può decidere se rieffettuare l'operazione.

### 5.6.6 Codice dedicato alla gestione di un ulteriore dispositivo (Spare)

Il codice per un eventuale spare non è stato chiaramente ancora scritto, la possibilità di utilizzo è però garantita e si potrà prendere spunto da ciò che è stato scritto per gli altri dispositivi. Si fa presente però che, con il programma al momento caricato nel PIC, non rimane molto spazio in memoria; alcune funzioni (passi da effettuare, riconoscimento,

finecorsa) sono già state implementate ma se fosse necessario scrivere parecchie altre righe di codice, probabilmente bisognerà anche cambiare microcontrollore, utilizzandone uno con lo stesso numero di pin e compatibile con l'attuale ma con maggiore memoria.

### 5.6.7 Codice dedicato alla gestione del Laser TDL50

Se viene selezionato il laser TDL il programma entra nella sezione di codice dedicata, la quale è suddivisa in due parti, una parte per la trasmissione di caratteri dal pc al laser e l'altra per la ricezione dal dispositivo al pc. La trasmissione avviene come negli altri casi: con la pressione del tasto INVIO la variabile *parola\_pronta* segnala che una stringa è disponibile e viene trasmessa carattere per carattere, al termine della stringa viene inviato il carattere di fine messaggio (carriage return).

La ricezione invece è gestita in modo diverso rispetto al modulo USART, in questo caso infatti non c'è un interrupt che interviene per avvisare dell'arrivo di un carattere, il programma controlla quindi costantemente il pin di ricezione B4 e nel caso cambi il suo valore, viene immediatamente utilizzata la funzione "ReadUART" per memorizzare il carattere nella variabile *c* e la lettura prosegue finché non viene letto il carattere di fine messaggio o il carattere Z (lettera che il TDL50 invia dopo aver eseguito un reset). Al termine della lettura tutta la stringa ricevuta viene scritta sul monitor utilizzando la funzione "putsUSART".

## 5.7 Conclusioni

In questo capitolo è stato descritto il programma realizzato per il controllo della scheda elettronica. I dispositivi da controllare dovranno effettuare scansioni in lunghezza d'onda, questo comporta la rotazione a velocità costante del reticolo di diffrazione. Nel caso particolare del laser FL2002, assieme al reticolo possono essere utilizzati un filtro etalon o un cristallo SHG; in questo caso bisognerà ruotare anche questi in modo da mantenerli "agganciati" durante la scansione. Nel caso dell'etalon bisogna utilizzare la sua formula caratteristica, nel caso del cristallo si effettua una linearizzazione della curva di conversione attorno al punto di lavoro.

Il programma principale è infine strutturato in unzioni, per facilitarne la lettura, risparmiare memoria nel caso di funzioni invocate in più punti e semplificare le operazioni di debug.

## Prove sperimentali

---

### 6.1 Introduzione

In questo capitolo si vogliono verificare la scheda elettronica realizzata ed il programma implementato in essa. Si effettuerá quindi un esperimento caratteristico del laboratorio Laser, per vedere come si comporta il sistema di controllo ed accertarsi che produca dei risultati compatibili con quelli reperibili in letteratura. Dopo una breve introduzione sul setup sperimentale e sulle basi teoriche che stanno alla base dell'esperimento, vengono quindi riportati ed analizzati i dati prodotti.

### 6.2 Setup Sperimentale

All'interno del laboratorio laser del progetto SPES viene studiata ed analizzata la tecnica della fotoionizzazione laser risonante, vengono cioè simulate le condizioni che si verificheranno nella sorgente LIS (Laser Ion Source) e si analizzano le procedure per riuscire ad ionizzare in modo corretto gli elementi d'interesse. In particolare si vogliono verificare i dati contenuti in letteratura sulle lunghezze d'onda necessarie alla ionizzazione e quali siano le combinazioni piú efficienti; per far questo si eseguono specifici esperimenti di spettroscopia.

#### 6.2.1 Spettroscopia

Con spettroscopia s'intende lo studio di uno spettro (in questo caso luminoso), per vedere quali siano le lunghezze d'onda di cui é composto. Come si é già descritto nel Cap. 3, gli atomi di ogni elemento possono assorbire/emettere solo delle specifiche lunghezze d'onda; sfruttando questa proprietá si puó, osservando lo spettro emesso da una particolare sostanza, capire da quali elementi sia composta.

#### Lampada a catodo cavo

La lampada a catodo cavo (HCL: Hollow Cathod Lamp) é uno strumento costituito da un bulbo in vetro, riempito di gas inerte (Ar o Ne), al suo interno si trovano inoltre un catodo ed un anodo; il catodo é costituito da una capsula, la quale, nel fondo, contiene l'elemento caratterizzante della lampada.

Una grande differenza di potenziale tra anodo e catodo, ionizza gli atomi di gas che vengono accelerati verso il catodo e lo colpiscono. A causa di queste continue collisioni, una parte degli atomi che compongono il catodo vengono staccati (azione di sputtering) e si mescolano al gas; questi atomi collidendo con altre particelle si eccitano e poi tornano al loro stato fondamentale emettendo fotoni alla propria lunghezza d'onda caratteristica.

Questo tipo di lampada é in grado di produrre quindi una luce che contiene le righe spettrali dell'elemento catodico e risulta utile in molti esperimenti, in particolare viene utilizzata per la spettroscopia di assorbimento atomico: la luce della lampada viene fatta passare attraverso un gas o un liquido, analizzando lo spettro in uscita si puó capire se ci sia presenza o meno di atomi in grado di assorbire la lunghezza d'onda del fascio entrante.

Nel laboratorio Laser del progetto SPES la lampada viene utilizzata per un altro tipo di esperimento, si vogliono infatti verificare i processi di fotoionizzazione laser risonante.

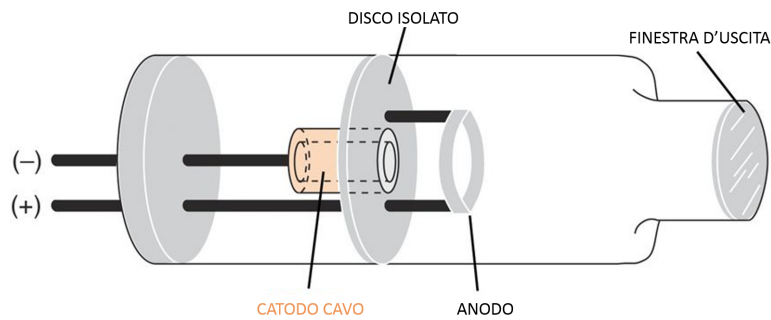


Fig. 6.1: Rappresentazione schematica di una lampada a catodo cavo (HCL: Hollow Cathode Lamp).

Conoscendo l'elemento caratterizzante della lampada, viene introdotto un fascio di luce laser ad una lunghezza d'onda che possa eccitarne ed ionizzarne gli atomi presenti. Per un effetto detto "optogalvanico", una radiazione elettromagnetica che investa un gas in una scarica autosostenuta, provoca un cambiamento nelle sue proprietà elettriche (generalmente variandone la conduttività); per questo motivo, osservando la corrente che scorre nella lampada, si può vedere un aumento di quest'ultima nel momento in cui la luce laser viene assorbita e produce un aumento di ioni.

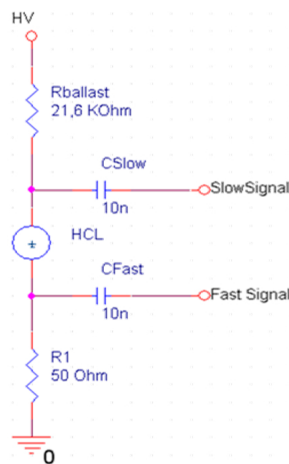


Fig. 6.2: Misurazione della tensione per ricavare i segnali slow e fast.

Utilizzando atomi caratterizzati da più livelli energetici, si possono effettuare due diversi tipi di esperimenti:

- Slow signal: viene introdotta una sola lunghezza d'onda, la quale fa compiere uno dei salti energetici agli elettroni: l'introduzione di questa energia che poi viene distribuita tra tra gli atomi a causa degli urti, provoca un leggero aumento degli ioni che si riflette in un segnale di picco della durata di alcuni  $\mu s$ .
- Fast signal: vengono introdotte due o più lunghezze d'onda (utilizzando chiaramente laser diversi ma sincronizzati) le quali fanno compiere salti diversi agli elettroni fino a raggiungere la ionizzazione, questo improvviso aumento di ioni si riflette in un segnale di picco della durata di alcuni  $ns$ .

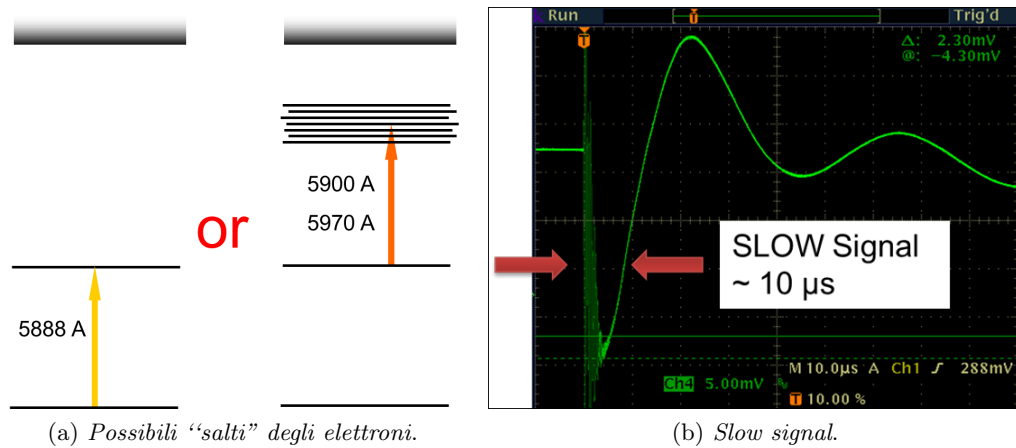


Fig. 6.3: Utilizzo di radiazione laser per eccitare gli elettroni per far compiere loro un "salto" al livello energetico successivo.

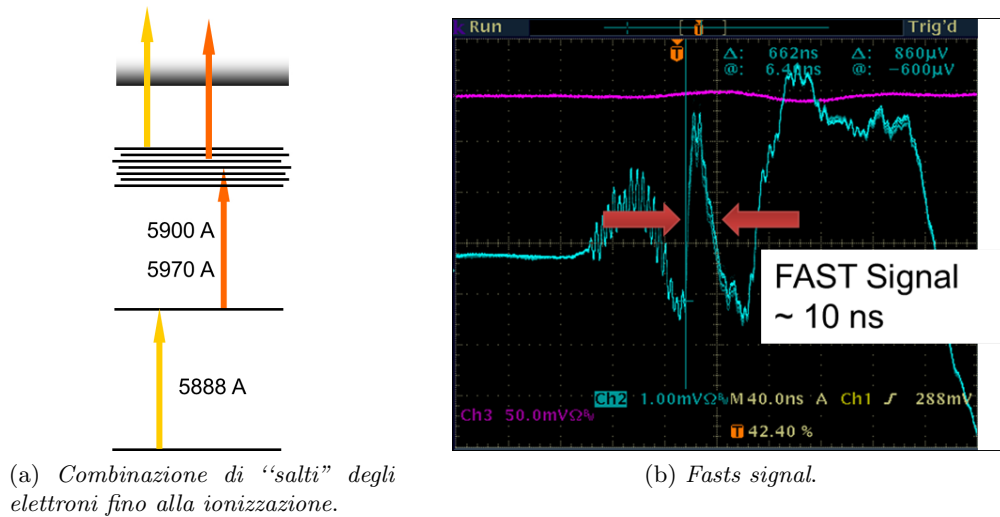


Fig. 6.4: Utilizzo di radiazione laser per eccitare gli elettroni per far compiere loro due "salti" fino ad arrivare alla ionizzazione.

Grazie a questi esperimenti si può verificare innanzitutto la capacità di sintonizzarsi sulle lunghezze d'onda associate a particolari salti energetici (utilizzando lo slow signal) ed anche l'efficienza di ionizzazione (utilizzando il fast signal) sperimentando diverse combinazioni come si può osservare nell'esempio riportato in Fig. 6.5.

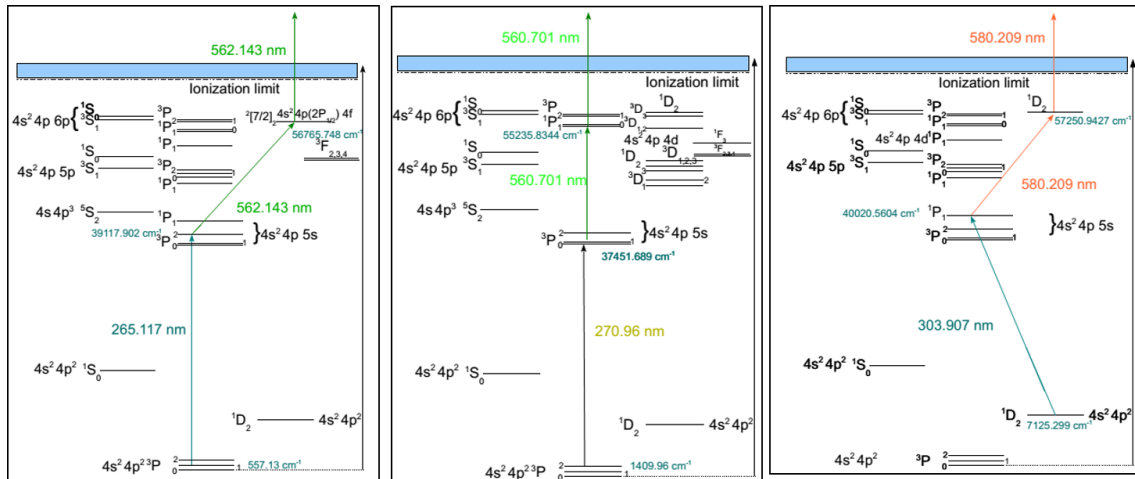


Fig. 6.5: Diverse combinazioni di salti quantici in un atomo di germanio.

### 6.3 Esperimento di fotoionizzazione laser

La comodità di avere a disposizione un laser sintonizzabile, è la possibilità di effettuare scansioni in lunghezza d'onda per verificare (o trovare) quale sia quella che massimizza il risultato desiderato. Le informazioni riportate in letteratura sono infatti riferite a condizioni molto particolari, il semplice variare della temperatura o di altri parametri modificano leggermente i livelli quantici all'interno degli atomi. Per questo motivo utilizzando questo tipo di strumentazione, si può ottenere un grafico riportante il segnale di assorbimento o ionizzazione, in funzione della lunghezza d'onda.

#### 6.3.1 Ionizzazione dello stagno nella lampada a catodo cavo

Il test viene effettuato utilizzando un catodo avente lo stagno (Sn) come elemento caratterizzante. Viene prima effettuata una misura di slow signal, variando la lunghezza d'onda del fascio entrante nella HCL, per verificare il primo dei salti quantici necessari. Una volta trovato e verificato l'intervallo di scansione che permette di visualizzare il picco di assorbimento, viene effettuata la misura di fast signal combinando all'interno della HCL due fasci laser: uno fisso proveniente dal laser TDL50 ed uno variabile proveniente dal laser FL2002.

#### Segnale slow signal

I dati riportati in letteratura dicono che lo stagno può assorbire una lunghezza d'onda pari a 614.96035 nm, per questo motivo viene effettuata una scansione nel range 614.84 - 615.04. L'oscilloscopio memorizza il picco minimo di slow signal con una frequenza di 10Hz e tutti i dati sono poi riportati successivamente in un grafico.

In Fig. 6.6 è riportato il risultato ottenuto con questo esperimento, si può vedere chiaramente come il massimo venga raggiunto con un andamento "a campana" e si verifichi per una lunghezza d'onda pari a 614.937 che corrisponde ad una staratura del contatore analogico.

#### Segnale fast signal

A questo punto si può procedere con la verifica di ionizzazione, che viene convalidata da un'analisi del segnale fast.

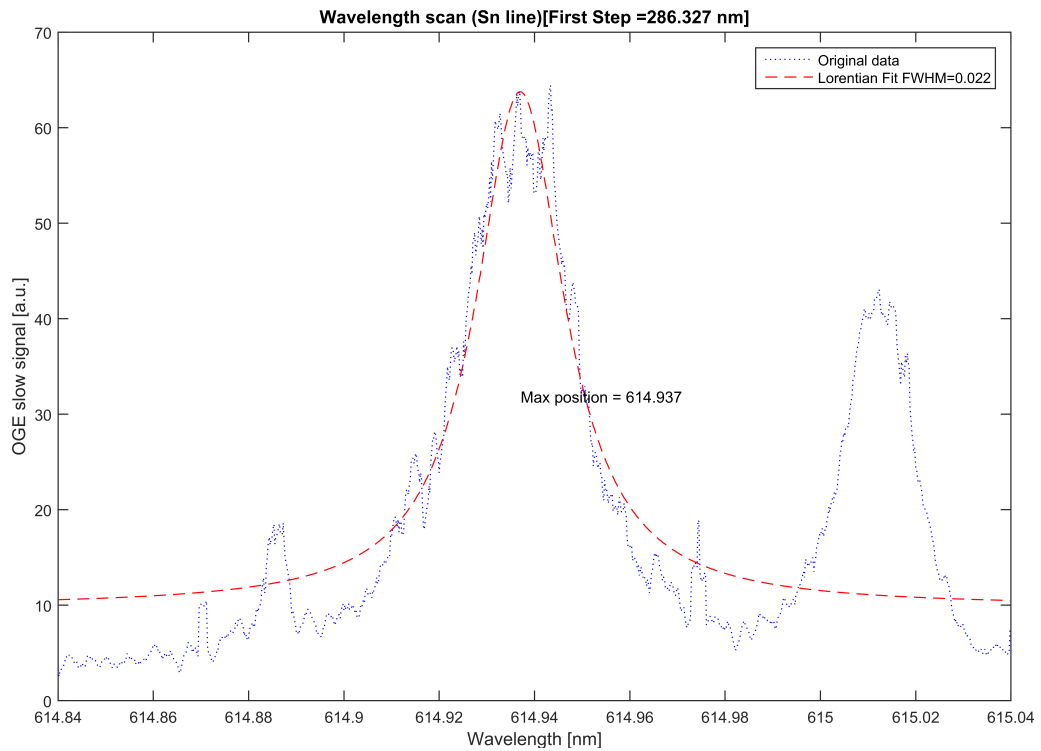


Fig. 6.6: Scansione per verificare un salto quantico nello stagno presente nella lampada HCL.

Anche in questo caso l'esperimento viene effettuato con una scansione da parte di uno dei due laser (mentre l'altro rimane ad una lunghezza d'onda fissa) ed il risultato è riportato in Fig. 6.7

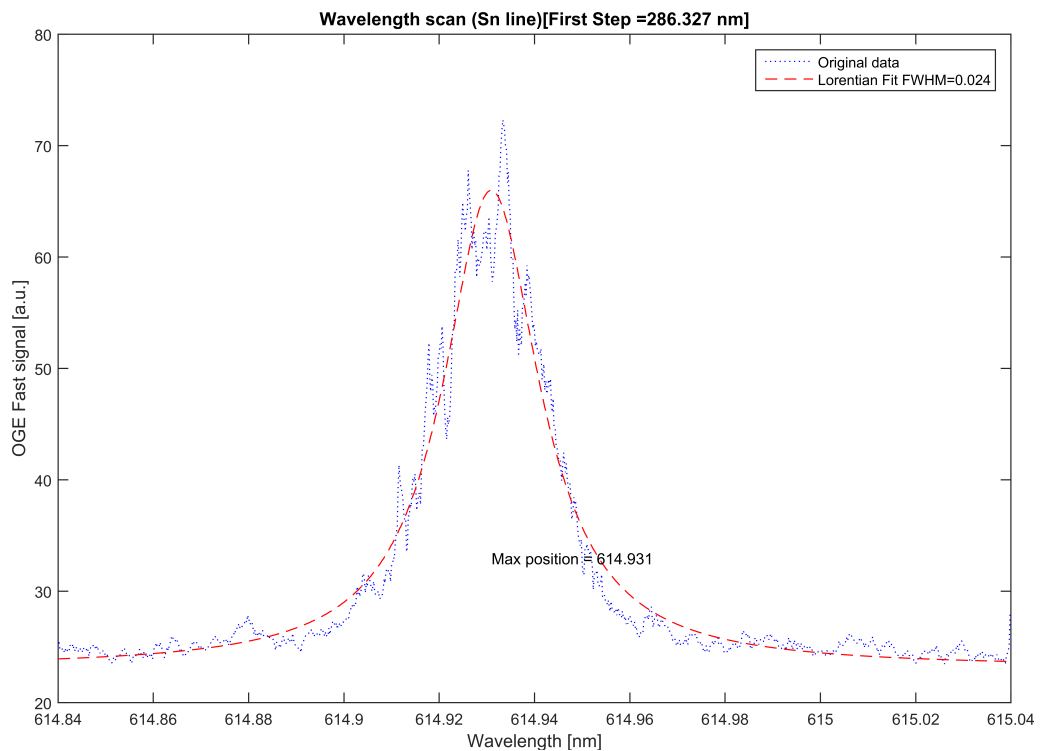


Fig. 6.7: Scansione per verificare la ionizzazione dello stagno presente nella lampada HCL.

Il grafico mostra chiaramente come la combinazione dei due laser sia riuscita a compiere la ionizzazione dello stagno presente nella lampada.

É interessante vedere come il grafico del segnale slow presenti un secondo picco a circa 615.015 nm, osservando i dati presenti in letteratura si capisce che questo segnale é dovuto all'eccitazione del gas neon (Ne) presente nella lampada, il quale risulta assorbire la lunghezza d'onda 615.02985 nm. Questo segnale sparisce però nel segnale fast che riporta infatti solo la completa ionizzazione dell'elemento.

## 6.4 Conclusioni

In questo capitolo si é voluta testare la scheda elettronica progettata e la corretta funzionalità del programma scritto. Il setup sperimentale consiste nell'utilizzo di due laser, uno dei quali viene pilotato dalla PCB per effettuare una scansione in lunghezza d'onda; i fasci laser sono fatti entrare all'interno di una lampada a catodo cavo (HCL) per verificare prima l'assorbimento ed il relativo salto quantico (osservando il segnale slow dalla lampada) e successivamente il doppio assorbimento e la conseguente ionizzazione (osservando il segnale fast).



# Conclusioni

---

Nel presente lavoro di tesi é stato progettato, realizzato e verificato il sistema di controllo per alcuni dispositivi presenti nel laboratorio Laser del progetto SPES. Fino a questo momento infatti, il laboratorio non sfruttava appieno le proprie potenzialit , non essendo possibile utilizzare il laser sintonizzabile FL2002 per effettuare scansioni automatiche; il controllo che é stato implementato, ha avuto quindi, come obiettivo principale, la messa in funzione di questo dispositivo.

Essendo presenti in laboratorio altri dispositivi simili, si é pensato di realizzare un sistema di controllo che riuscisse ad interfacciarsi anche con:

- il Laser sintonizzabile TDL50, che necessita di una comunicazione seriale per ricevere istruzioni da pc.
- il Monocromatore JOBIN YVON, che essendo anch'esso, come l'FL2002, pilotato da motori Passo-Passo, utilizza la stessa tecnica di controllo

Per raggiungere tale scopo, si é deciso di realizzare una scheda elettronica che, utilizzando un microcontrollore, sia in grado d'interfacciare l'operatore con questi tre dispositivi.

Sulla base dell'hardware presente e delle funzioni richieste al sistema di controllo, si é eseguito uno studio per realizzare un circuito che possa far fronte alle diverse criticit  nel miglior modo possibile; la scheda é stata quindi prima progettata utilizzando il software Orcad Capture, poi disegnata col software Orcad Layout ed infine realizzata utilizzando la tecnica della fotoincisione.

Terminata la realizzazione dell'hardware, si é passati alla programmazione del software da inserire nel microcontrollore: facendo particolare attenzione ai segnali (forniti in uscita e richiesti in ingresso) dei diversi dispositivi, si sono realizzate varie funzioni in linguaggio C (ognuna legata ad un diverso aspetto/necessit ) che unite in modo appropriato danno vita al programma principale.

La scheda, ora pronta all'uso, é stata utilizzata in alcuni esperimenti per verificarne l'affidabilit  e la bont  del controllo. I risultati sono pi  che buoni, confermando che il sistema di controllo realizzato soddisfa completamente le specifiche che erano state richieste.

Il software realizzato é stato pensato per soddisfare le necessit  odierne nel laboratorio Laser del progetto SPES; se per  in futuro dovessero presentarsi casistiche particolari che la scheda non é in grado d'affrontare, sar  possibile apportare piccole o grandi modifiche hardware e/o software. Ricordando che la memoria del PIC dedicata al programma é quasi esaurita, si prevede la necessit  di effettuare un upgrade del microcontrollore, utilizzando un PIC compatibile con la piedinatura utilizzata ma con maggior memoria.



# Ringraziamenti

---

Al termine di questo lavoro di tesi, mi sento di ringraziare innanzitutto il Prof. Roberto Oboe, relatore di questa tesi, per tutto quello che é riuscito a trasmettermi in questi anni attraverso il suo lavoro e la sua personalit .

Ringrazio il Dott. Alberto Andrighetto, coordinatore del progetto SPES, per avermi dato l'opportunit  di conoscere, lavorare e far parte di questo progetto.

Un grande grazie all'Ing. Daniele Scarpa, che con tanta pazienza e cortesia,   stato sempre disponibile a condividere la sua esperienza e le sue conoscenze, con consigli e chiarimenti che mi hanno aiutato ad affrontare ogni tipo di dubbio ed ostacolo.

Un ringraziamento anche a tutti gli altri ragazzi e ragazze del progetto SPES, per i bei momenti trascorsi insieme in questi mesi, che hanno reso quest'esperienza indimenticabile.

Un ringraziamento speciale ai miei genitori, Francesco e Paola, per avermi dato la possibilit  d'iniziare e completare questo percorso universitario e per avermi sempre incoraggiato e sostenuto. Un grazie ai miei fratelli: Maria Teresa, Filippo, Caterina, Rachele e Beatrice, che mi hanno sopportato con pazienza in tutti questi anni anche nei momenti difficili e stressanti.

Infine un ringraziamento a tutti i miei amici e compagni di studio, per aver condiviso le interminabili ore in biblioteca e per essere stati sempre presenti, nei momenti felici ma anche in quelli pi  difficili.



# Bibliografia

---

- [1] C. Anelli, Impiego di fibre polimeriche lunghe per la sintesi di target refrattari ad elevata porosità e permeabilità per il progetto SPES, A.A. 2014-2015
- [2] E. Boratto, Progettazione ed ingegnerizzazione del Front-End protonico del progetto SPES, A.A. 2014-2015
- [3] M. Ballan, Progettazione e test di componenti e sistemi per l'operazione e la sicurezza del sistema target-sorgente del progetto SPES, A.A. 2013-2014
- [4] M. Bruschi, Prove di fotoionizzazione selettiva mediante laser sintonizzabile in vapori atomici di alluminio, A.A. 2008-2009.
- [5] W.Demtröder, Laser Spectroscopy vol 1 Basic Principles Fourth edition.
- [6] M. Zigliotto, Dispense di Fondamenti di macchine ed azionamenti elettrici, 2009
- [7] Manuale Laser FL2002 Lambda Physik
- [8] Microchip, MPLAB C18 C COMPILER LIBRARIES
- [9] Microchip, PIC18FXX2 DataSheet
- [10] Allegro, A4988 DataSheet
- [11] M. Laurenti, LCD44780.h, libreria C per controllo display LCD.
- [12] <http://www.vincenzov.net/>
- [13] [www.wikipedia.it](http://www.wikipedia.it)



## Estratti dei manuali

### A.1 Estratti del manuale FL2002

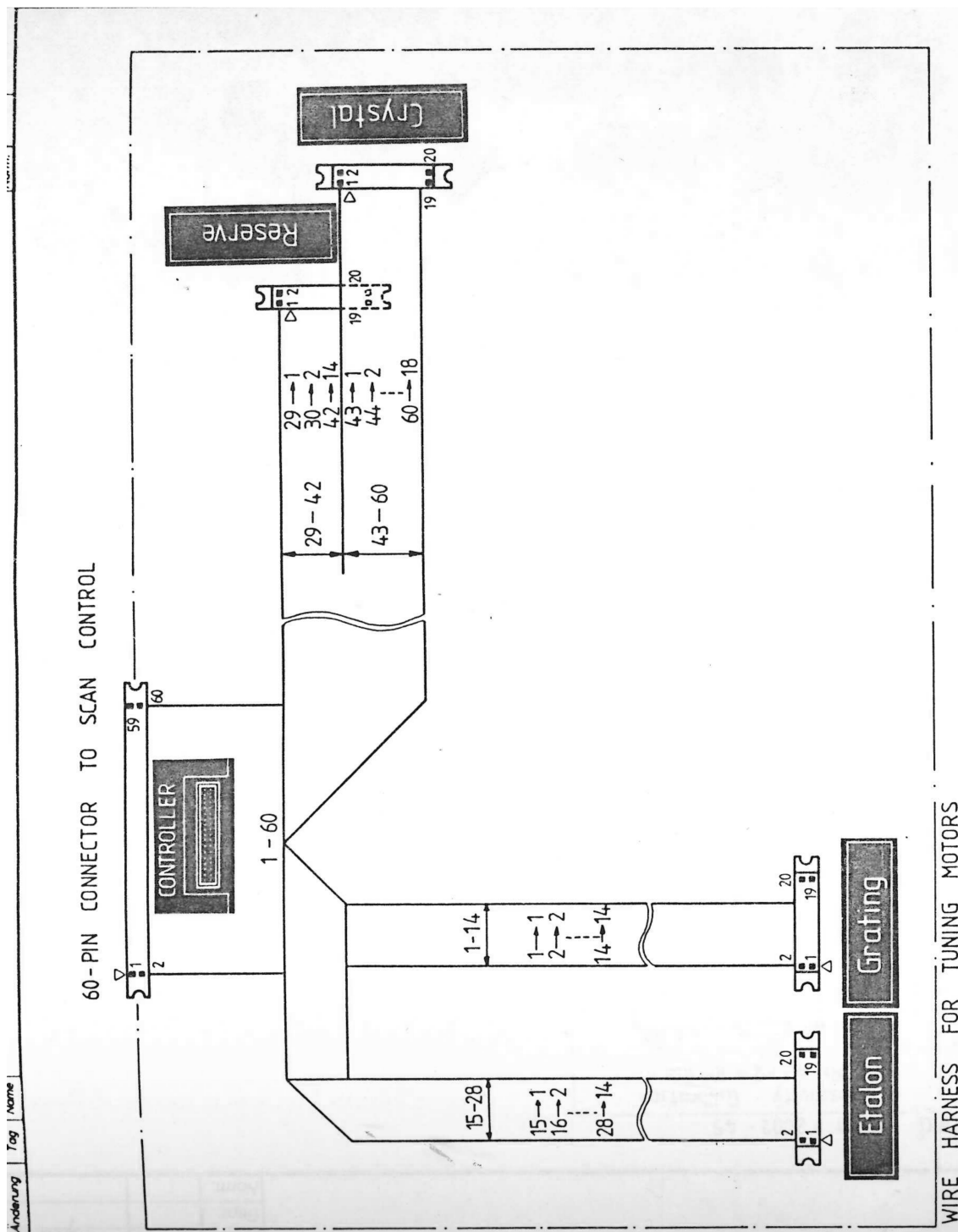


Fig. A.1: Suddivisione pin del connettore per FL2002.

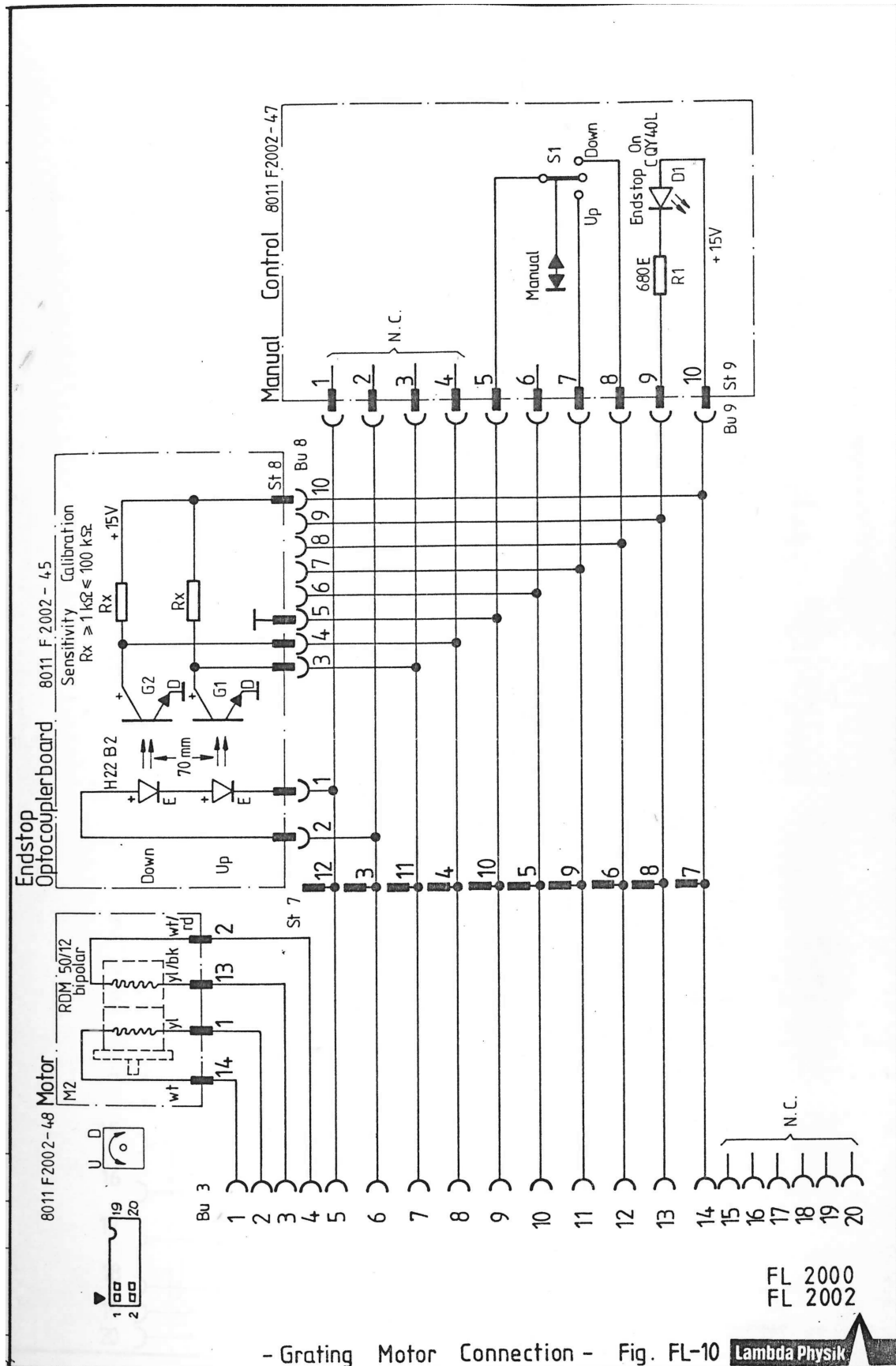


Fig. A.2: Schema di collegamento interno del grating.



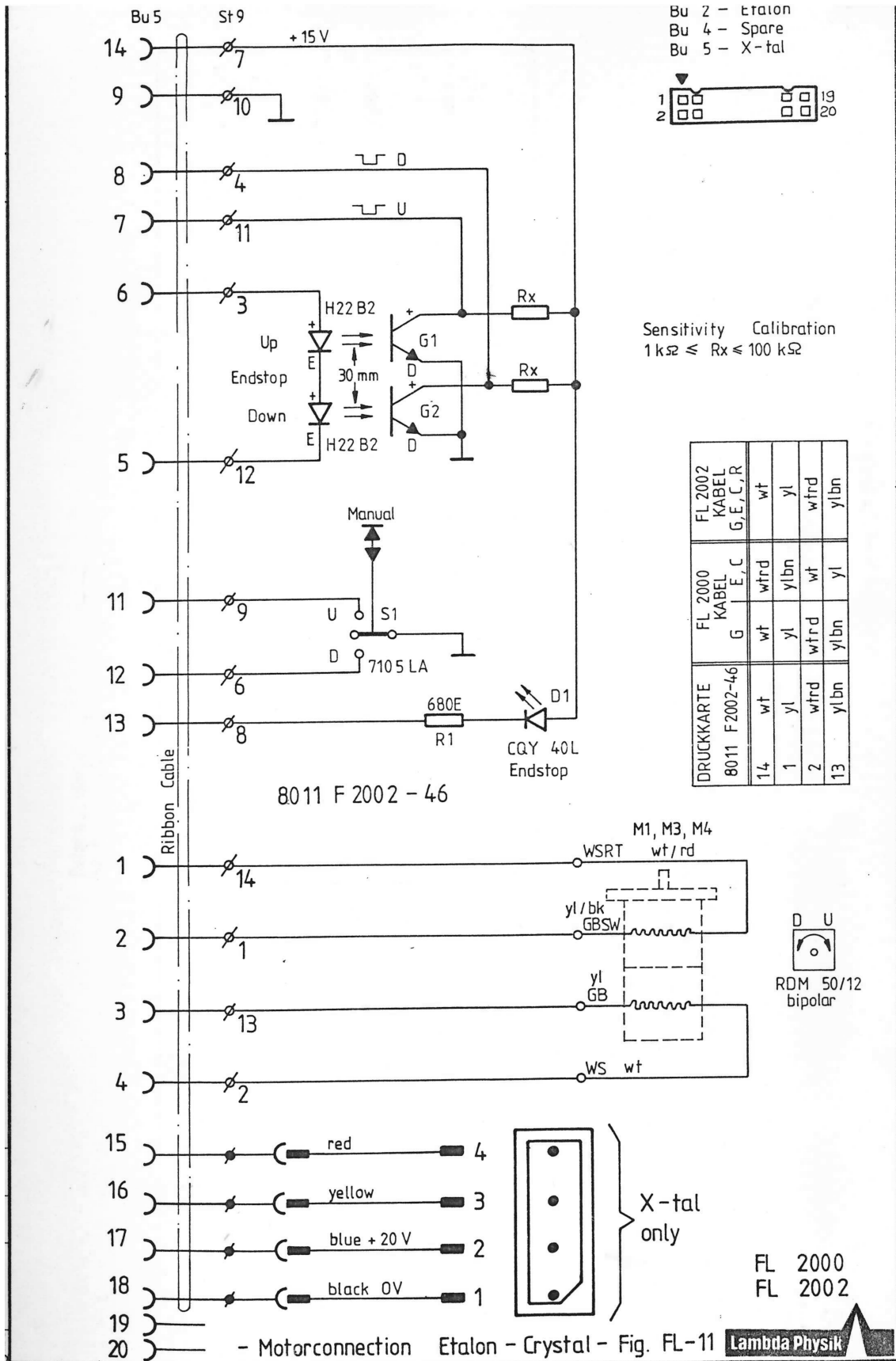


Fig. A.3: Schema di collegamento interno dell'etalon/cristallo.

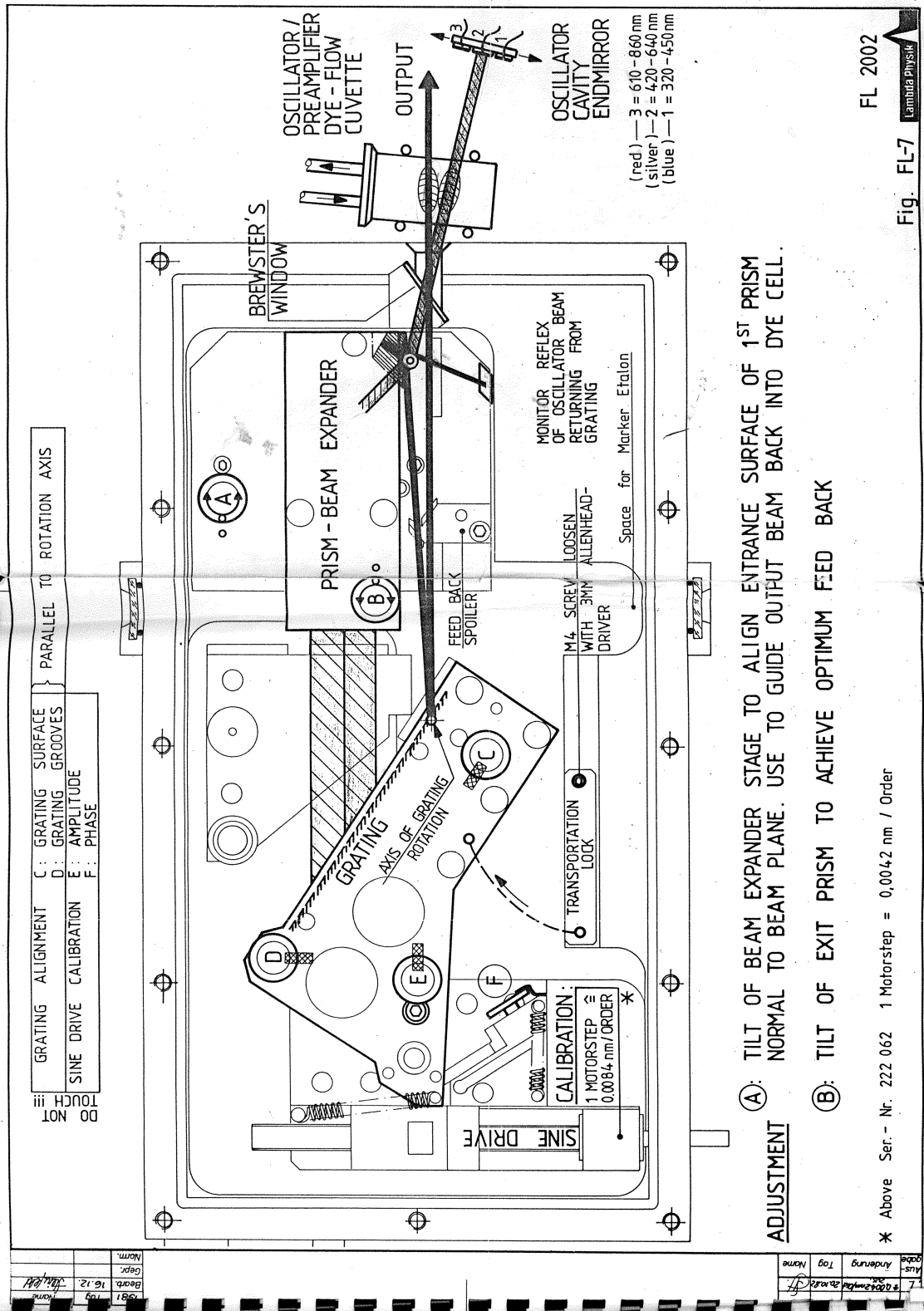


Fig. A.4: Percorso del fascio all'interno del Laser FL2002.

## A.2 Estratti del manuale TDL50

## SCAN PARAMETERS

Input Code		Number of characters	Function	Conditions/Notes
Letter	ASCII hex			
(B)	42H	6	Set BEGIN	$375.000 \leq \lambda_B < \lambda_E \leq 750.000$
(E)	45H	6	Set END	$375.000 \leq \lambda_B < \lambda_E \leq 750.000$
(P)	50H	6	Set INIT	$375.000 \leq \lambda_I \leq 750.000$
(S)	53H	2	Set slow motor speed	$01 \leq V_s \leq 06$
(N)	4EH	2	Set number of scans	$01 \leq N_{bs} \leq 99$
(F)	46H	2	rate of reporting current $\Delta$ to RS232 monitor	F=00 $\rightarrow$ no report F=01 $\rightarrow \lambda_I + 1n$ report at $\Delta\lambda = 1$ F=02 $\rightarrow \lambda_I + 10n$ report at $\Delta\lambda = 10$ F=03 $\rightarrow \lambda_I + 100n$ report at $\Delta\lambda = 100$

TABLE I

Fig. A.5: Parametri per la scansione nel laser TDL50.

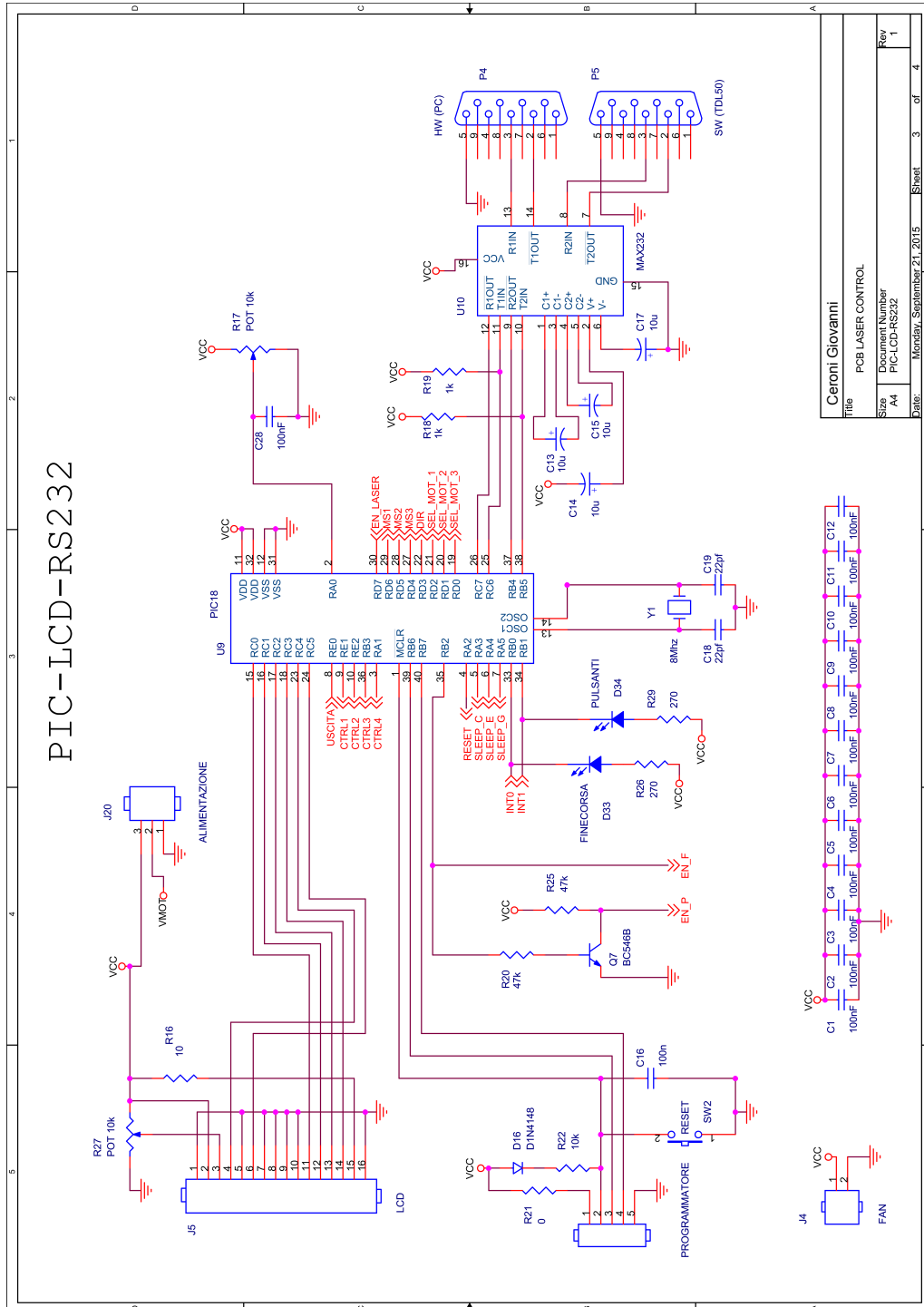
## COMMANDS

Input code		Number of characters	Functions	Conditions/Notes
Letter	ASCII hex			
H	48 <sub>H</sub>	0	halt motor drive	(manual or automatic)
C	43 <sub>H</sub>	0	continue	continue or resume motor drive
Z	5A <sub>H</sub>	0	reset	reset (from RB02 or RS232)
G	47 <sub>H</sub>	0	go auto	format: Bxxxxxx Exxxxxx Sxx Nxx---Fxx or Pxxxxxx Bxxxxxx Exxxxxx Sxx Nxx---Fxx

TABLE II

Fig. A.6: Comandi per il laser TDL50.

## Schemi elettrici della scheda elettronica realizzata



Ceroni Giovanni	
Title	PCE LASER CONTROL
Sheet	Document Number
A4	PIC-LCD-RS232
Date:	Monday, September 21, 2015
Sheet	3 of 4
Rev	1

Fig. B.1: Schema elettrico di collegamento del PIC-LCD-RS232.



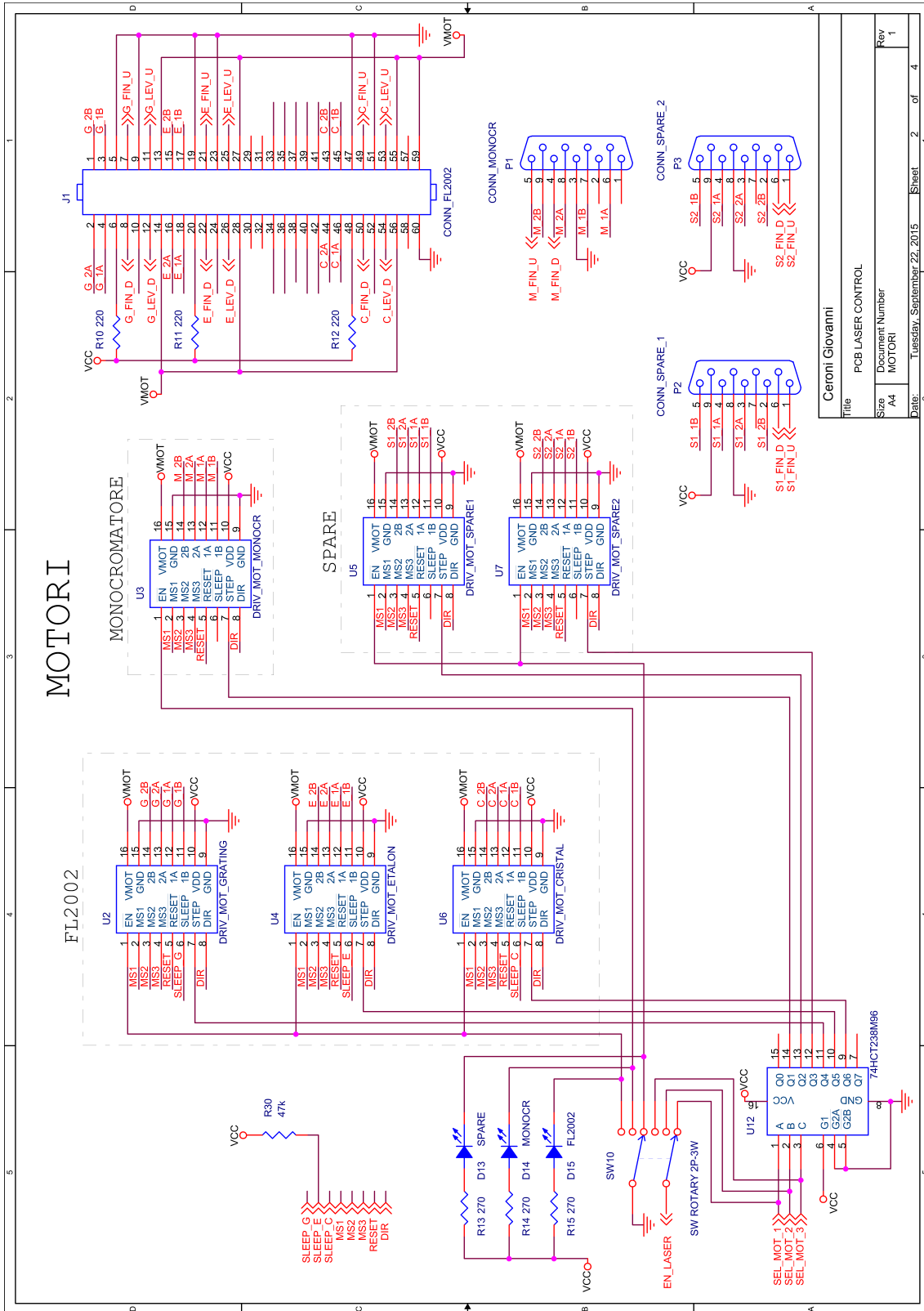
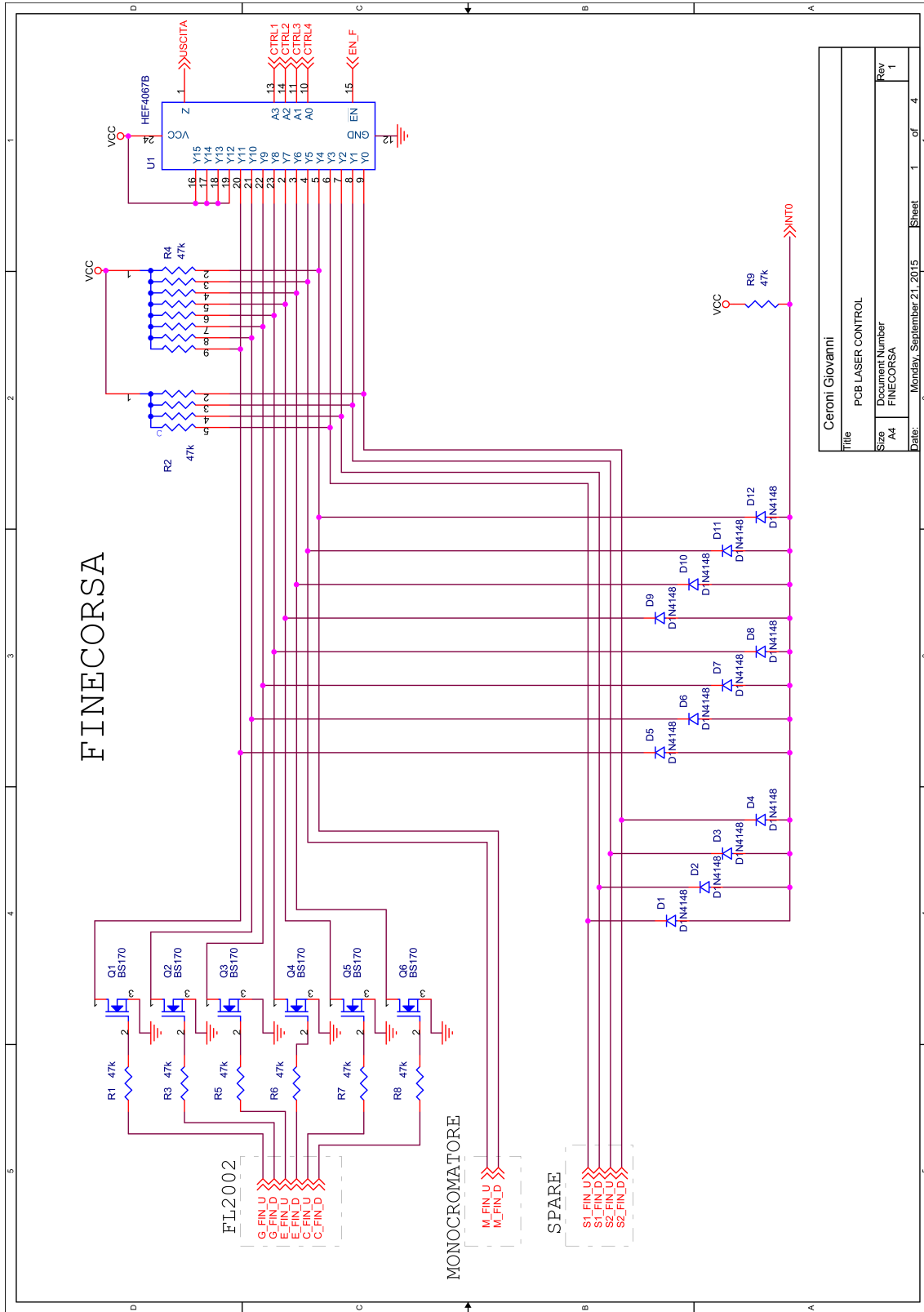


Fig. B.2: Schema elettrico di collegamento dei driver per i motori.

Cerioni Giovanni			
PCB LASER CONTROL			
Title	Document Number	Sheet	Rev
Size	MOTORI	2	1
Date	Tuesday, September 22, 2015	of	4







Ceroni Giovanni	
Title PCB LASER CONTROL	
Size A4	Document Number FINECORS
Date: Monday, September 21, 2015	Rev 1
Sheet 1	of 4

Fig. B.3: Schema elettrico di collegamento dei finecorsa.



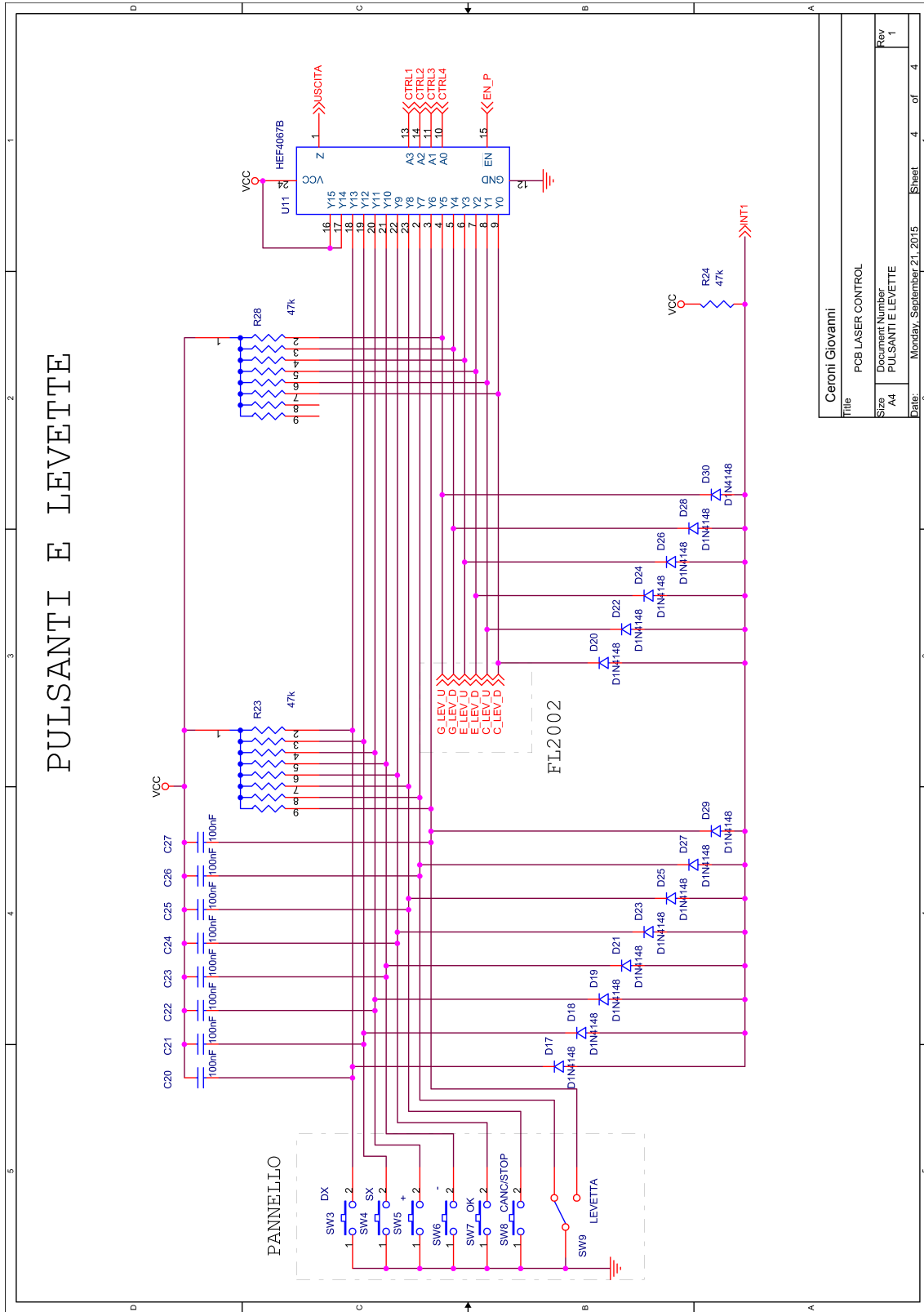


Fig. B.4: Schema elettrico di collegamento dei pulsanti.



## Codice

---

### C.1 Main program

```
#include <xc.h> //libreria per il compilatore xc8
#include <adc.h>
#include <p18C452.h>
#include <stdlib.h>
#include <string.h>
#include <delays.h>
#include <usart.h>
#include <stdio.h>
#include "LCD44780.h"
#include <math.h>
#include <sw_uart.h>

//DEFINIZIONI PER UNA PIU' FACILE COMPRESIONE DEL CODICE
#define USCITA PORTEbits.RE0 //uscite dei mux finecorsa/pulsanti
#define CTRL1 LATEbits.LATE1 //combinazione per scelta ingresso dei mux
#define CTRL2 LATEbits.LATE2 //combinazione per scelta ingresso dei mux
#define CTRL3 LATBbits.LATB3 //combinazione per scelta ingresso dei mux
#define CTRL4 LATAbits.LATA1 //combinazione per scelta ingresso dei mux
#define RESET_DRV_MOT LATAbits.LATA2 //se =0 reset SI se =1 reset NO
#define EN_LASER PORTDbits.RD7 //se =0 enable SI se =1 enable NO
#define SLEEP_C PORTAbits.RA3 //pin di sleep nel driver del cristallo
#define SLEEP_E PORTAbits.RA4 //pin di sleep nel driver dell'etalon
#define SLEEP_G PORTAbits.RA5 //pin di sleep nel driver del grating
#define MS1 LATDbits.LATD6 //pin per la definizione tipo di step
#define MS2 LATDbits.LATD5 //pin per la definizione tipo di step
#define MS3 LATDbits.LATD4 //pin per la definizione tipo di step
#define DIR LATDbits.LATD3 //pin per il verso di rotazione
#define SEL_MOT_1 LATDbits.LATD2 //A0 NEL DEMUX
#define SEL_MOT_2 LATDbits.LATD1 //A1 NEL DEMUX
#define SEL_MOT_3 LATDbits.LATD0 //A2 NEL DEMUX
#define LETTURA LATBbits.LATB2 //se =0 leggo finecorsa, se =1 leggo pulsanti

//IMPOSTAZIONI PER IL PROGRAMMATORE
#pragma config OSC=HS //utilizzo di oscillatore esterno
#pragma config WDT=OFF //watchdog disabilitati
#pragma config LVP=OFF //Low voltage program disabilitato

//PROTOTIPI DELLE FUNZIONI
int trova_finecorsa(void);
int trova_pulsante(void);
void passo_E(int velocit);
void passo_C(int velocit);
void passo_G(int velocit);
void passo_M(int velocit);
void passo_SP1(void);
void taratura(char *onda);
void scrivereLCD(char a[],char b[]);
void levette (int en_G,int en_E,int en_C,int en_M);
void inizializzazione (int tipo);
void massimizza();
void interpolazione ();
void muovere (int motore);
void accelerazione (int *vel_iniziale);
```



```

double b=0,a=0;
long n_cristallo[2]={0,0};
float ONDSTEP;
int n_operazione2=0;
int laser[2]={0,0};
char operazione[2]="1";
char operazione2[2]="1";
int ADC_ris=0;
int NOlevette=0;
int percento[2]={0,0};
int vel_pot=0;
int velocit=0;
int vel_min_G=250;
int vel_min_M=200;
int dir_ragg=0;
int dir_sc=0;

//FUNZIONI DI RITARDO PER IL CORRETTO FUNZIONAMENTO DEL SOFTWARE UART
void DelayRXBitUART (){
Delay10TCYx(151);
}
void DelayRXHalfBitUART(){
    Delay10TCYx(72);
}

void DelayTXBitUART(){
    Delay10TCYx(150);
}

//ISR INTERRUPT
void interrupt __low_priority eventoL(void){
    if(INTCON3bits.INT1IF==1){          //se viene premuto un pulsante
        q=1;
        INTCON3bits.INT1IF=0;
    }
}

void interrupt __high_priority eventoH(void)
{
if(PIR1bits.RCIF==1){                //se viene attivata la comunicazione seriale
    getsUSART(message,1);
    if(message[0]=='q')
        RESET_DRV_MOT = 0; //reset SI
    if(message[0]=='i')
        q=4;
    if(message[0]=='r')
        RESET_DRV_MOT = 1; //reset NO
    if(message[0]=='w')
        q=3;
    if(message[0]!='q'&&message[0]!='r'&&message[0]!='w'){
        if(message[0]!=13){
            temp[g]=message[0];
            g++;
        }
        else{
            temp[g]='\r';
            g=0;
            strcpy(parola,temp);
            strcpy(temp,vuoto);
            parola_pronta=1;
        }
    }
}
PIR1bits.RCIF = 0; //reset del bit di flag

```

```

    }
    if(INTCONbits.INTOIF==1){          //se avviene un finecorsa
        RESET_DRV_MOT = 0; //reset
        q=2;
        INTCONbits.INTOIF=0;
    }
}
int main(int argc, char** argv) {

//DEFINIZIONI DELLE PORTE
TRISA=0x01;    //i pin della porta B sono tutti uscite tranne A0
TRISB=0;      //i pin della porta C sono tutti uscite
TRISC=0;      //i pin della porta C sono tutti uscite
TRISD=0;      //i pin della porta C sono tutti uscite
TRISDbits.TRISD7 = 1; //EN_LASER è un ingresso
TRISBbits.TRISB4 = 1; //ingresso sw_uart
TRISEbits.TRISE0 = 1;
TRISEbits.TRISE1 = 0;
TRISEbits.TRISE2 = 0;

//SELEZIONE TIPO DI STEP
MS1 = 0;      //MS1
MS2 = 0;      //MS2
MS3 = 0;      //MS3

DIR = 0;      //dir (0->Down 1->Up)

//INIZIALIZZAZIONE SCELTA MUX E DEMUX
CTRL1 = 1;
CTRL2 = 1;
CTRL3 = 1;
CTRL4 = 1;

SEL_MOT_1 = 0;
SEL_MOT_2 = 0;
SEL_MOT_3 = 0;

LETTURA = 0; //abilitata lettura finecorsa

ADCON1=0x07; //inizializzazione registro degli interrupt
//configurazione ADC
OpenADC(ADC_FOSC_2&ADC_RIGHT_JUST&ADC_1ANA_0REF,ADC_CHO&ADC_INT_OFF);
//CONFIGURAZIONE LCD
OpenLCD ();
//CONFIGURAZIONE USART
OpenUSART(USART_TX_INT_OFF&USART_RX_INT_OFF&USART_ASYNC_MODE&
    USART_EIGHT_BIT&USART_CONT_RX&USART_BRGH_LOW, 103); // 103 per boud 1200;12 per boud 9600 8Mhz
//CONFIGURAZIONE UART
OpenUART();
Delay1KTCYx(4);

//CONFIGURAZIONE REGISTRI INTERRUPT
RCONbits.IPEN=1;    //utilizzo delle priorità abilitato
INTCONbits.GIEH=1;  //inetrrupt ad alta priorità abilitati
INTCONbits.GIEL=1;  //inetrrupt a bassa priorità abilitati
INTCONbits.INTOIE=1; //INT0 è abilitato
INTCON3bits.INT1E=1; //INT1 è abilitato
INTCON2bits.INTEDG0=0; //INT0 viene "visto" sul fronte di discesa
INTCON2bits.INTEDG1=0; //INT1 viene "visto" sul fronte di discesa
INTCON3bits.INT1IP=0; //INT1 è a bassa priorità
PIE1bits.RCIE=1;    //L'interrupt USART è abilitato
IPR1bits.RCIP=1;    //L'interrupt USART è ad alta priorità

```



```

message[0]='\0'; //non dovrebbe servire
//INIZIO PROGRAMMA CHE VIENE EFFETTUATO UNA VOLTA SOLA A MENO CHE NON SI
//A MENO CHE NON SI TORNI TRAMITE GOTO
INIZIO:
putsUSART("restart:");
SLEEP_C=0;
SLEEP_E=0;
SLEEP_G=0;

//VERIFICO CHE NON SIANO PRESENTI FINECORSIA
while(PORTBbits.RB0==0){
    finecorsa=trova_finecorsa();
    sprintf(riga,"end-stop %s",find_finecorsa);
    scrivereLCD(riga,"");
    putsUSART(riga);
    Delay1TCYx(100);
}
//RESET DELLE VARIABILI PRINCIPALI
i=0;
parola_pronta=0;
strcpy(parola,vuoto);
//scrivo=0;
sel_laser=0;
n_operazione=0;
n_operazione2=0;
sequenza=0;
strcpy(onda_ adesso,zeri);
strcpy(onda_inizio,zeri);
strcpy(onda_fine,zeri);
n_onda_ adesso=0;
n_onda_fine=0;
n_onda_inizio=0;
n_onda_ragg=0;
n_onda_sc=0;
n_passi_sc=0;
n_passi_ragg=0;
primo=1;
fine=0;
lasertrovato=0;
aggiornato=1;
laser[0]=0;
n=0;
cristallo_primo=1;
NOlevette=0;
g=0;
temp[0]='\0';
strcpy(parola,temp);
strcpy(temp,vuoto);
//INIZIO CICLO WHILE PRIMARIO
while(1){
    switch(q){
        case 1: //pulsanti
            pulsante=trova_pulsante();
            q=0;
            break;
        case 2: //finecorsa
            finecorsa=trova_finecorsa();
            q=0;
            break;
        case 3:
            q=0;
            goto INIZIO;
            break;
    }
}

```

```

    case 4:
        sprintf(riga, "lambda: %f steps: %i passi", n_onda_ adesso, passi_G);
        putsUSART(riga);
        q=0;
    }
    if(pulsante==8){
        sequenza--;
        canc=0;
    }
    //CONVERSIONE ADC
    ConvertADC(); // inizio conversione
    while( BusyADC() ); // aspetta completamento
    ADC_ris = ReadADC(); // legge il risultato
    vel_pot=trunc(ADC_ris/7)+4; //conversione
    if(sequenza==0 && message[0]!=13 && pulsante!=9){
        RESET_DRV_MOT = 0; //reset abilitato per non far partire i motori ora
        lasertrovato=0;
        SEL_MOT_1 = 1; //cambio il primo pin di selezione
        if(EN_LASER==1){ //se vedo abilitato il laser SP ho finito di cercare
            sprintf(rigaLCD,"SP");
            laser[1]=2; //memorizzo il numero corrispondente
            lasertrovato=1;
        }
        if(lasertrovato==0){
            SEL_MOT_2 = 1; //cambio il secondo pin di selezione
            if(EN_LASER==1){//se vedo abilitato il laser SP ho finito di cercare
                sprintf(rigaLCD,"MO");
                laser[1]=3; //memorizzo il numero corrispondente
                lasertrovato=1;
            }
        }
        if(lasertrovato==0){
            SEL_MOT_3 = 1; //cambio il terzo pin di selezione
            if(EN_LASER==1){//se vedo abilitato il laser SP ho finito di cercare
                sprintf(rigaLCD,"FL");
                laser[1]=1; //memorizzo il numero corrispondente
                lasertrovato=1;
            }
        }
        }
    //RESETTO I PIN DI SELEZIONE PER POTER POI COMANDARLI CORRETTAMENTE
    SEL_MOT_1 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_3 = 0;
    if(laser[1]!=laser[0]){ //se il selettore è stato girato cambia il laser
        sprintf(riga,"selected %s",rigaLCD);
        putsUSART(riga);
        putsUSART("if you want TDL select spare and press 4\r\n");
        scrivereLCD(riga,"press enter/OK");
        laser[0]=laser[1];
    }
}
else{
    if(sequenza==0&&message[0]!='4')
        sel_laser=laser[1];
    if(sequenza==0 && parola[1]=='4'&&laser[1]==2)
        sel_laser=4;
    if(sequenza==0){
        sequenza++;
        parola_pronta=0;
        strcpy(parola,vuoto);
        message[0]='0';
        pulsante=15;
        RESET_DRV_MOT = 1; //reset disabilitato
    }
}

```

```

        lasertrovato=0;
    }
}
//////////
//FL2002//
//////////
if(sel_laser==1&&RESET_DRV_MOT == 1){
    if (NOlevette==0){//si possono usare le 3 levette presenti sul laser
        levette(1,1,1,0);
    }
    if (sequenza==1){
        putsUSART("1=G 2=E 3=C 4=Ma 5=Sc \r\n");
        sequenza++;
    }
    if(n_operazione==0 && aggiornato==1){
        ONDSTEP=0.84;          //centesimi di A per passo
        sprintf(riga,"5=Sc?  %s",operazione);
        scrivereLCD("1=G 2=E 3=C 4=Ma",riga);
        ShiftCursorLCD(0);
        CursorLCD (0,1);
        aggiornato=0;
    }
    if(n_operazione==0&&parola_pronta==1){//se è pronta una str da tastiera
        n_operazione=atoi(parola);          //converto la stringa in numero
        parola_pronta=0;                      //resetto la variabile
        aggiornato=1;                        //avviso che il numero è stato aggiornato
    }
    if(n_operazione==0&&(pulsante==10||pulsante==11||pulsante==12||
        pulsante==13))          //se è premuto un tasto tra su/giù/dx/sx
        taratura(&operazione);
    if(n_operazione==0&&pulsante==9){          //se è premuto il tasto ok
        n=0;
        n_riga=0;
        n_operazione=atoi(operazione); //converto la stringa in numero
        aggiornato=1;                    //avviso che il numero è stato aggiornato
        pulsante=15;
    }
}
if(n_operazione!=0&&n_operazione!=1&&n_operazione!=2&&n_operazione!=3){
    n_operazione=0;
    putsUSART("ERROR, TRY AGAIN");
}
//SOLO GRATING
if(n_operazione==1){ //se è stata scelta la modalità con solo grating
    SLEEP_G=1;
    if(primo==1&&sequenza==2){          //solo la prima volta
        putsUSART("1=scan 2=go to\r\n");
        primo++;
    }
    if(sequenza==2||sequenza==3)
        scelta();
    if((sequenza>=4||rip==1)){
        if(rip==1){          //se si sta ripetendo l'operazione
            levette(1,0,0,0); //si può muovere solo il grating
            if(scritto==0){
                scrivereLCD("adjust grating","press enter/OK");
                putsUSART("adjust grating and press enter/OK");
                scritto=1;
            }
        }
        if(message[0]==13||pulsante==9)//se viene premuto INVIO o OK
            vai=1;          //abilito l'esecuzione della movimentazione
        primo=1;
        message[0]='0';
        pulsante=15;
    }
}

```

```

    }
  }
  if(rip==0||vai==1) //se devo ripetere la movimentazione
    muovere(1);
}

}
//CON ETALON
if(n_operazione==2){ //se è stata scelta la modalità con l'etalon
  if(sequenza==2){
    SLEEP_G=1; // "sveglia" etalon e Grating
    SLEEP_E=1;
    scrivereLCD("move Et in norm","press enter/OK");
    putsUSART("move Etalon in normal position and press enter/OK");
    Delay1KTCYx(20);
    levette(0,1,0,0); //abilito solo la levetta dell'etalon
    sequenza++;
  }
  if(sequenza==3 && (parola_pronta==1 || pulsante==9)){
    parola_pronta=0;
    if(message[0]==13||pulsante==9){
      sequenza++;
      message[0]='0';
      pulsante=15;
    }
  }
  if(sequenza==4){
    DIR=0; //muovo in down
    for(passi=0;passi<=201;passi++){
      passo_E(vel_pot);
      sequenza++;
    }
    if(passi>=201&&sequenza==5){
      scrivereLCD("couple Grating","press enter/OK");
      putsUSART("couple Grating and press enter/OK");
      sequenza++;
    }
  }
  if(sequenza==6)
    levette(1,0,0,0); //abilita solo levetta grating
  if(sequenza==6 && (parola_pronta==1 || pulsante==9)){
    parola_pronta=0;
    if(message[0]==13||pulsante==9){ //se invio/ok
      sequenza++;
      message[0]='0';
      pulsante=15;
      primo=1;
      n_onda_adesso=0;
    }
  }
  if(sequenza==7)
    inizializzazione(1);
  if(n_passi_ragg!=0&&passi_G!=labs(n_passi_ragg)&&sequenza==8){
    DIR=abs(dir_ragg-1); //valuto da che parte devo girare
    passo_E(vel_pot);
    alfa=0.0000125*passi_E;
    delta_lambda=-(n_onda_adesso*alfa*alfa/2);
    if(fabs(delta_lambda)>(ONDSTEP/3600)){ //cent di A
      DIR=dir_ragg;
      passo_G(vel_pot);
      accelerazione(vel_pot);
      passi_E=0;
      Delay10KTCYx(100);
    }
  }
}

```

```

        if(abs(passi_G)>=labs(n_passi_ragg)){
            passi_G=0;
            scrivereLCD ("going to start","");
            putsUSART("going to start \r\n");
            Delay10KTCYx(200);
            Delay10KTCYx(200);
            Delay10KTCYx(200);
            sequenza++;
        }
    }
    if((sequenza==9 && fine==0)){
        DIR=abs(dir_sc-1);
        passo_E(vel_min_G);
        alfa=0.0000125*passi_E;
        delta_lambda=-((n_onda_adesso*alfa*alfa)/2);
        if(fabs(delta_lambda)>(ONDSTEP/3600)){ //cent di A
            DIR=dir_sc;
            passo_G(vel_min_G);
            percentuale(passi_G);
            tornare[1]=tornare[1]+passi_E;
            passi_E=0;
            alfa=0;
            delta_lambda=0;
            Delay10KTCYx(100);
        }
        if(sequenza==9 && abs(passi_G)>=labs(n_passi_sc)){
            passi_E=tornare[1];
            fine=1;
            scrivereLCD ("end","");
            sprintf(riga,"end \r\n");
            putsUSART(riga);
            sequenza++;
        }
    }
    if(sequenza==10&&rip!=1){
        ripetere();
    }
    if(sequenza==10&&rip==1){ //se si sta ripetendo l'operazione
        levette(1,0,0,0); //si può muovere solo il grating
        if(scritto==0){
            scrivereLCD("adjust grating","press enter/OK");
            putsUSART("adjust grating and press enter/OK");
            scritto=1;
        }
        if(message[0]==13||pulsante==9){ //se viene premuto INVIO o OK
            sequenza=9;
            rip=0;
            message[0]='0';
            pulsante=15;
            passi_E=0;
            passi_G=0;
            delta_lambda=0;
            alfa=0;
        }
    }
}
//CRISTALLO
if(n_operazione==3){
    SLEEP_C=1; // "sveglia" cristallo e Grating
    SLEEP_G=1;
    if(sequenza==2)
        inizializzazione(1);
    //SE NON SONO ANCORA ARRIVATO A "RAGGIUNGERE"
}

```

```

if(n_passi_ragg!=0&&abs(passi_G)<labs(n_passi_ragg)&&sequenza==3){
    DIR=dir_ragg;
    passo_G(vel_pot);
    accelerazione(vel_pot);
}
//QUANDO SONO ARRIVATO A "RAGGIUNGERE"
if(n_passi_ragg!=0&&abs(passi_G)>=labs(n_passi_ragg)&&sequenza==3){
    massimizza(); //massimizzo il primo punto (0;0)
}
//RIPETO PER GLI ALTRI 4 PUNTI
if(sequenza==4 && div_sc<5){
    if(fatto==0){
        DIR=dir_sc; //nella direzione data dalla scansione
        if(i<passi_div_sc){
            passo_G(10);
            i++;
        }
        else
            fatto=1;
    }
    else
        massimizza(); //massimizzo
}
if(div_sc==5&&sequenza==4){ //fine memorizzazione punti
    interpolazione(); //eseguo l'interpolazione
    sequenza++;
}
//TORNO INDIETRO PER COMINCIARE LA SCANSIONE
if(sequenza==5){
    DIR=abs(dir_sc-1);
    for(i=1;i<=(labs(n_passi_sc)+30);i++)//+30 per poi essere nel
        //verso giusto col grating che ha gioco
        passo_G(10);
    for(i=1;i<=abs(y_calibrazione[4]);i++)
        passo_C(10);
    sequenza++;
}
CR: //da qui riparto se voglio ripetere la scansione
if(sequenza==6){
    scrivereLCD("adjust grating","press enter/OK");
    putsUSART("adjust grating and press enter/OK");
    sequenza++;
}
if(sequenza==7){
    levette(1,0,0,0); //abilita solo levetta grating
    if(message[0]==13|pulsante==9){ //quando è premuto invio/OK
        sequenza++;
        message[0]=' ';
        pulsante=15;
        passi_G=0;
        passi_C=0;
        i=0;
        NOlevette=1;
    }
}
//INIZIA SCANSIONE
if(sequenza==8){
    DIR=dir_sc;
    passo_G(vel_min_G); //un passo del grating
    n_cristallo[1]=round(a*i+b);//calcolo la posizione del cristallo
    i++;
    if(n_cristallo[1]!=n_cristallo[0]){ //sarà sempre diverso al
        //massimo di 1 perchè il cristallo si muove molto meno

```

```

        DIR=dir_sc;
        passo_C(vel_min_G); //un passo del cristallo
        n_cristallo[0]=n_cristallo[1];
    }
    percentuale(passi_G); //calcolo a che punto sono
    //SE HO RAGGIUNTO LA FINE DELLA SCANSIONE
    if(i>labs(n_passi_sc)){
        sequenza++;
        scrivereLCD("end","");
        sprintf(riga,"end\r\n");
        putsUSART(riga);
        fine=1;
    }
}
if(fine==1){
    ripetere(); //si vuole ripetere?
    if (rip==1){
        sequenza=6; //sequenza impostata per poter ripetere
        goto CR;
    }
}
}
}
//////////
////TDL50//
//////////
if(sel_laser==4){
    DelayTXBitUART();
    //-----TRASMISSIONE-----//
    if(parola_pronta==1){
        i=0;
        //SCRIVO UN CARATTERE PER VOLTA FINO ALLA FINE DELLA PAROLA
        while(parola[i]!='\r'&&parola[i]!=' '){
            WriteUART(parola[i]);
            i++;
        }
        sprintf(riga,"\r"); //alla fine dev'esserci il carattere di CR
        WriteUART(riga[0]);
        parola_pronta=0;
    }
    i=1;
    //-----RICEZIONE-----//
    if(PORTBbits.RB4==0){ //se sta arrivando una stringa
        c[0] = ReadUART(); //leggo immediatamente un carattere
        //inizio a leggere e continuo finchè non arriva il carattere di fine
        while(c[i-1]!='\r'&&c[i-1]!='Z'){
            c[i] = ReadUART(); //memorizzo il carattere in c
            i++;
        }
        putsUSART(c); //scrivi a monitor la stringa che è arrivata
        for(i=0;i<=30;i++) //resetto in contenuto della variabile c
            c[i]='\0';
    }
}
//MONOCROMATORE
if(sel_laser==3&&RESET_DRV_MOT == 1){
    levette(0,0,0,1); //abilito la levetta del monocromatore
    if(sequenza==1){
        ONDSTEP=2; //centesimi di A per passo
        putsUSART("1=scan 2=go to\r\n");
        sequenza++;
    }
    if(sequenza==2||sequenza==3)

```

```
        scelta(); //opero la scelta
MON:
if((sequenza>=4||rip==1)){ //dopo inizializzazione o per ripetere
    muovere(0);
    //ESCO DAL CICLO GOTO SE SONO STATI PREMUTI TASTI
    if(rip==1&&message[0]!='0')
        goto MON;
}
}
}
```



## C.2 Funzioni

```
int trova_finecorsa(){
    LETTURA = 0;
    int ver_fin=1;
    int f=0;
    while(ver_fin==1){ //continuo finchè non ho trovato il finecorsa
        f=f+1;
        switch(f){
            case 1: //etalon down
                CTRL1 = 1;
                CTRL2 = 0;
                CTRL3 = 0;
                CTRL4 = 0;
                sprintf(find_finecorsa,"E-D");
                break;
            case 2: //etalon up
                CTRL1 = 1;
                CTRL2 = 0;
                CTRL3 = 0;
                CTRL4 = 1;
                sprintf(find_finecorsa,"E-U");
                break;
            case 3: //cristallo down
                CTRL1 = 0;
                CTRL2 = 1;
                CTRL3 = 1;
                CTRL4 = 0;
                sprintf(find_finecorsa,"C-D");
                break;
            case 4: //cristallo up
                CTRL1 = 0;
                CTRL2 = 1;
                CTRL3 = 1;
                CTRL4 = 1;
                sprintf(find_finecorsa,"C-U");
                break;
            case 5: //grating down
                CTRL1 = 1;
                CTRL2 = 0;
                CTRL3 = 1;
                CTRL4 = 0;
                sprintf(find_finecorsa,"G-D");
                break;
            case 6: //grating up
                CTRL1 = 1;
                CTRL2 = 0;
                CTRL3 = 1;
                CTRL4 = 1;
                sprintf(find_finecorsa,"G-U");
                break;
            case 7: //monoc UP
                CTRL1 = 0;
                CTRL2 = 1;
                CTRL3 = 0;
                CTRL4 = 1;
                sprintf(find_finecorsa,"M-U");
                break;
            case 8: //monoc down
                CTRL1 = 0;
                CTRL2 = 1;
                CTRL3 = 0;
                CTRL4 = 0;
        }
    }
}
```

```

        sprintf(find_finecorsa,"M-D");
        break;
    default:
        RESET_DRV_MOT = 1;
        ver_fin=0;
        break;
    }
    if(USCITA==0){ //se il pin di uscita è a massa ho trovato il finecorsa
        //scrivo quale finecorsa ho trovato
        scrivereLCD ("end-stop",find_finecorsa);
        sprintf(riga,"end-stpp %i",find_finecorsa);
        putsUSART(riga);
        ver_fin=0; //avviso che ho trovato un finecorsa
    }
}
return f; //comunico il numero corrispondente
}

```

```

-----
int trova_pulsante(){
    LETTURA = 1;
    int ver_pulsante=1;
    int p=-1;
    //continuo finchè non trovo il pulsante o il pin di interrupt torna a 1
    while(ver_pulsante==1 && PORTBbits.RB1==0){
        p=p+1;
        switch(p){
            case 0: //C_LEV_D
                CTRL1 = 0;
                CTRL2 = 0;
                CTRL3 = 0;
                CTRL4 = 0;
                Delay1TCYx(1);
                break;
            case 1: //C_LEV_U
                CTRL1 = 0;
                CTRL2 = 0;
                CTRL3 = 0;
                CTRL4 = 1;
                Delay1TCYx(1);
                break;
            case 2: //E_LEV_D
                CTRL1 = 0;
                CTRL2 = 0;
                CTRL3 = 1;
                CTRL4 = 0;
                Delay1TCYx(1);
                break;
            case 3: //E_LEV_U
                CTRL1 = 0;
                CTRL2 = 0;
                CTRL3 = 1;
                CTRL4 = 1;
                Delay1TCYx(1);
                break;
            case 4: //G_LEV_D
                CTRL1 = 0;
                CTRL2 = 1;
                CTRL3 = 0;
                CTRL4 = 0;
                Delay1TCYx(1);
                break;
            case 5: //G_LEV_U
                CTRL1 = 0;

```

```
        CTRL2 = 1;
        CTRL3 = 0;
        CTRL4 = 1;
        Delay1TCYx(1);
        break;
    case 6:          //LEV_D
        CTRL1 = 0;
        CTRL2 = 1;
        CTRL3 = 1;
        CTRL4 = 0;
        Delay1TCYx(1);
        break;
    case 7:          //LEV_U
        CTRL1 = 0;
        CTRL2 = 1;
        CTRL3 = 1;
        CTRL4 = 1;
        Delay1TCYx(1);
        break;
    case 8:          //CANC
        CTRL1 = 1;
        CTRL2 = 0;
        CTRL3 = 0;
        CTRL4 = 0;
        Delay1TCYx(1);
        break;
    case 9:          //OK
        CTRL1 = 1;
        CTRL2 = 0;
        CTRL3 = 0;
        CTRL4 = 1;
        Delay1TCYx(1);
        break;
    case 10:         //-
        CTRL1 = 1;
        CTRL2 = 0;
        CTRL3 = 1;
        CTRL4 = 0;
        Delay1TCYx(1);
        break;
    case 11:         //+
        CTRL1 = 1;
        CTRL2 = 0;
        CTRL3 = 1;
        CTRL4 = 1;
        Delay1TCYx(1);
        break;
    case 12:         //sx
        CTRL1 = 1;
        CTRL2 = 1;
        CTRL3 = 0;
        CTRL4 = 0;
        Delay1TCYx(1);
        break;
    case 13:         //dx
        CTRL1 = 1;
        CTRL2 = 1;
        CTRL3 = 0;
        CTRL4 = 1;
        Delay1TCYx(1);
        break;
    default:
        scrivereLCD("ERROR case","press again");
```

```

        ver_pulsante=0;
        break;
    }
    if(USCITA==0){
        ver_pulsante=0;
    }
}
if(ver_pulsante==1)
    p=15;
return p;
}

```

```

-----
void passo_E(velocit){
    Delay1KTCYx(velocit);
    SEL_MOT_1 = 1;
    SEL_MOT_2 = 0;
    SEL_MOT_3 = 1;
    Delay1KTCYx(velocit);
    SEL_MOT_3 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_1 = 0;
    if(DIR==0)
        passi_E--;
    else
        passi_E++;
    putsUSART("U");
}

```

```

-----
void passo_C(velocit){
    Delay1KTCYx(velocit);
    SEL_MOT_1 = 0;
    SEL_MOT_2 = 1;
    SEL_MOT_3 = 1;
    Delay1KTCYx(velocit);
    SEL_MOT_3 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_1 = 0;
    if(DIR==0)
        passi_C--;
    else
        passi_C++;
    putsUSART("U");
}

```

```

-----
void passo_G(velocit){
    Delay1KTCYx(velocit);
    Delay1KTCYx(velocit);
    SEL_MOT_1 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_3 = 1;
    Delay1KTCYx(velocit);
    Delay1KTCYx(velocit);
    SEL_MOT_3 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_1 = 0;
    if(DIR==0){
        passi_G--;
        n_onda_adesso=n_onda_adesso-0.84;
    }
    else{
        passi_G++;
        n_onda_adesso=n_onda_adesso+0.84;
    }
}

```

```

    putsUSART("U");
}
-----
void passo_M(velocit){
    SEL_MOT_1 = 1;
    SEL_MOT_2 = 0;
    SEL_MOT_3 = 0;
    Delay1KTCYx(velocit);
    SEL_MOT_1 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_3 = 0;
    Delay1KTCYx(velocit);
    if(DIR==0){
        passi_M--;
        n_onda_ adesso=n_onda_ adesso-ONDSTEP;
    }
    else{
        passi_M++;
        n_onda_ adesso=n_onda_ adesso+ONDSTEP;
    }
    putsUSART("U");
}
-----
void passo_SP1(){
    SEL_MOT_3 = 0;
    SEL_MOT_2 = 1;
    SEL_MOT_1 = 0;
    Delay1KTCYx(15);
    SEL_MOT_3 = 0;
    SEL_MOT_2 = 0;
    SEL_MOT_1 = 0;
    Delay1KTCYx(15);
}
-----
void taratura(char *onda){
    if(pulsante==13){ //pulsante destra
        n=0;
        n_riga++; //incremento la variabile usata per posizionarsi sul carattere
        ShiftCursorLCD(1); //sposto il cursore a destra di una posizione
        pulsante=15; //resetto la variabile pulsante al valore "nullo"
    }
    if(pulsante==12){ // sinistra
        n=0;
        n_riga--;
        ShiftCursorLCD(0);
        pulsante=15;
    }
    if(pulsante==11){ //pulsante su
        n++; //incremento di 1 il numero da inserire
        sprintf(numero,"%i",n); //lo memorizzo come caratt nella var numero
        *(onda+n_riga)=numero[0]; //lo copio nella giusta posiz nella stringa
        pulsante=15;
    }
    if(pulsante==10){ //pulsante giù
        n--;
        sprintf(numero,"%i",n);
        *(onda+n_riga)=numero[0];
        pulsante=15;
    }
    aggiornato=1;
}
-----

```

```

void scrivereLCD(char a[],char b[]){
    ClearLCD ();    //resetto il display
    HomeLCD ();    //porto in posizione in alto a sinistra
    WriteVarLCD(a); //scrivo la prima stringa
    Line2LCD();    //mi porto nella seconda riga
    WriteVarLCD(b); //scrivo la seconda stringa
}
-----
void levette (int en_G,int en_E,int en_C,int en_M){ //1=abilitato    0=disabilitato
    v=0;
    if(pulsante==0&&en_C==1){ //se la levetta premuta è abilitata
        DIR=0;    //definisco la direzione in cui girare
        passo_C(vel_pot);
        //FINCHE' RIMANE PREMUTA LA LEVETTA
        while(PORTBbits.RB1==0&&RESET_DRV_MOT==1){
            accelerazione(vel_pot);
            passo_C(vel_pot);
        }
        pulsante=15;    //resetto la variabile pulsante
    }
    if(pulsante==1&&en_C==1){
        DIR=1;
        passo_C(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_C(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==2&&en_E==1){
        DIR=0;
        passo_E(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_E(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==3&&en_E==1){
        DIR=1;
        passo_E(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_E(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==4&&en_G==1){
        DIR=0;
        passo_G(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(&vel_pot);
            passo_G(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==5&&en_G==1){
        DIR=1;
        passo_G(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_G(vel_pot);
        }
    }
}

```

```

        pulsante=15;
    }
    if(pulsante==6&&en_M==1){
        DIR=0;
        passo_M(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_M(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==6&&en_G==1){
        DIR=0;
        passo_G(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_G(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==7&&en_M==1){
        DIR=1;
        passo_M(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_M(vel_pot);
        }
        pulsante=15;
    }
    if(pulsante==7&&en_G==1){
        DIR=1;
        passo_G(vel_pot);
        while(PORTBbits.RB1==0){
            accelerazione(vel_pot);
            passo_G(vel_pot);
        }
        pulsante=15;
    }
}

-----
//DECREMENTA LA VARIABILE CHE VIENE PASSATE E CHE VERRA' USATA COME TEMPO DI
//ATTESA TRA DUE PASSI
void accelerazione (int *vel_iniziale){
    v++;
    if(v==10&&*vel_iniziale>=5)
        *vel_iniziale=*vel_iniziale-2;
    if(*vel_iniziale<4)
        *vel_iniziale=4;
    if(v>=10)
        v=0;
}

-----
void percentuale(long passi_fatti){
    percento[1]=abs(trunc(abs(passi_fatti)*100/abs(n_passi_sc)));
    //SCRIVO SOLO SE LA PERCENTUALE E' CAMBIATA
    if (percento[1]!=percento[0]){
        sprintf(riga,"%i %%",percento[1]);
        scrivereLCD("rate",riga);
        percento[0]=percento[1];
    }
}
}

```

```

-----
//FUNZIONE PER SCEGLIERE SE FARE SCANSIONE O RAGGIUNGERE UNA POSIZIONE, QUESTA
//POI INVoca ALLA FINE LA FUNZIONE INIZIALIZZAZIONE
void scelta(){
    //SE NON E' ANCORA SCELTO E SE C'E' STATO UN AGGIORNAMENTO
    if(n_operazione2==0&&aggiornato==1){
        sprintf(riga,"2 go to %s",operazione2); //aggiorno il display
        scrivereLCD("1 scan",riga);
        ShiftCursorLCD(0);
        CursorLCD (0,1);
        aggiornato=0;
    }
    //SE E' STATO SCRITTO CON LA TASTIERA
    if(n_operazione2==0&&parola_pronta==1){
        n_operazione2=atoi(parola); //converto il carattere in numero
        parola_pronta=0;
        aggiornato=1;
        sequenza++;
        message[0]='0'; //aggiunto a settembre serve?
    }
    //SE E' PREMUTO UN TASTO DA PANNELLO (-,+ ,SX ,DX)
    if(n_operazione2==0&&(pulsante==10|pulsante==11|pulsante==12|pulsante==13))
        taratura(&operazione2); //invoco taratura
    if(n_operazione2==0&&pulsante==9){ //se è premuto il tasto OK
        n=0;
        n_riga=0;
        n_operazione2=atoi(operazione2); //converto il carattere in numero
        sequenza++;
        aggiornato=1;
        pulsante=15;
    }
    //SE IL NUMERO NON E' COMPATIBILE CON LA SCELTA
    if(n_operazione2!=0&&n_operazione2!=1&&n_operazione2!=2){
        n_operazione2=0;
        putsUSART("ERROR, TRY AGAIN");
        sequenza--;
    }
    //SE LA SCELTA E' STATA FATTA SI PUO' PARTIRE CON L'INIZIALIZZAZIONE
    if(sequenza==3 && n_operazione2!=0)
        inizializzazione(n_operazione2);
}
-----
//QUESTA FUNZIONE DOMANDA LE INFORMAZIONI SULLA MOVIMENTAZIONE DESIDERATA
//E CALCOLA I PASSI DA FAR COMPIERE AL RETICOLO
void inizializzazione (int tipo){//tipo=1 per scansione-- tipo=2 per raggiungere
    if(primo==1){
        primo++;
        passi_G=0;
        passi_E=0;
        passi_C=0;
        passi_M=0;
        n_riga=0;
        CursorLCD(0,1);
    }
    ONDAADESSO: //da qui ricomincia se il primo inserimento non è corretto
    //SE LA PRIMA LAMBDA NON ENTRA NEL RANGE
    if(n_onda_ adesso<459999){
        if(aggiornato==1){ //se c'è stato un aggiornamento
            //AGGIORNO DISPLAY
            scrivereLCD("lambda now?",onda_ adesso);
            Line2LCD();
            if(primo==2){
                putsUSART("lambda now? ");
            }
        }
    }
}

```



```

        primo++;
    }
    for(i=1;i<=n_riga;i++) //il cursore si posiziona sotto il numero
        ShiftCursorLCD(1);
    aggiornato=0;
}
//SE VIENE PREMUTO UN TASTO PER LA TARATURA DA PANNELLO
if(n_onda_ adesso==0&&(pulsante==10|pulsante==11|pulsante==12|pulsante==13))
    taratura(&onda_ adesso);
//SE VIENE PREMUTO IL TASTO OK
if(pulsante==9){
    n_onda_ adesso=atof(onda_ adesso); //converto la stringa in numero
    pulsante=15;
    aggiornato=1; //avviso che c'è stato un aggiornamento
    n_riga=0;
}
//SE E' STATO PREMUTO INVIO
if(parola_pronta==1){
    strcpy(onda_ adesso,parola); //copio la stringa inserita
    parola_pronta=0;
    n_onda_ adesso=atof(onda_ adesso); //converto la stringa in numero
    primo=3;
    n_riga=0;
    message[0]='0';
}
//VERIFICA CHE IL NUMERO SIA COERENTE E NON CI SIANO ERRORI
if(n_onda_ adesso!=0&&(460000>n_onda_ adesso|| n_onda_ adesso>640000)){
    strcpy(onda_ adesso,zeri); //resetto la stringa
    n_onda_ adesso=0; //resetto il numero
    primo=2;
    goto ONDAADESSO; //torno indietro
}
}
ONDAINIZIO: //da qui ricomincia se il secondo inserimento non è corretto
if(n_onda_ adesso!=0&&n_onda_ inizio==0){
    if(primo==3){
        //SE SI VUOLE FARE SCANSIONE
        if(tipo==1){
            sprintf(riga,"lambda start?");
            putsUSART("lambda start?");
        }
        //SE SI VUOLE RAGGIUNERE UNA LUNGHEZZA D'ONDA
        else{
            sprintf(riga,"lambda arrive?");
            putsUSART("lambda arrive?");
        }
        primo++;
    }
}
//SE C'E' UN AGGIORNAMENTO AGGIORNO IL DISPLAY
if(aggiornato==1){
    scrivereLCD(riga,onda_ inizio);
    Line2LCD();
    for(i=1;i<=n_riga;i++) //il cursore si posiziona sotto il numero
        ShiftCursorLCD(1);
    aggiornato=0;
}
if(n_onda_ inizio==0&&(pulsante==10|pulsante==11|pulsante==12|pulsante==13))
    taratura(&onda_ inizio);
if(pulsante==9){
    n_onda_ inizio=atof(onda_ inizio);
    pulsante=15;
    aggiornato=1;
    n_riga=0;
}

```

```

    }
    if(parola_pronta==1){
        strcpy(onda_inizio,parola);
        parola_pronta=0;
        n_onda_inizio=atof(onda_inizio);
        aggiornato=1;
        n_riga=0;
        message[0]='0';
    }
    //VERIFICA CHE IL NUMERO SIA COERENTE E NON CI SIANO ERRORI
    if(n_onda_inizio!=0&&(460000>n_onda_inizio|| n_onda_inizio>640000)){
        strcpy(onda_inizio,zeri);
        n_onda_inizio=0;
        primo=3;
        goto ONDAINIZIO;
    }
}
//NEL CASO SI VOGLIA SOLO RAGGIUNERE, NON SERVE CHIEDERE onda_fine
//CHE VIENE IMPOSTATA AL VALORE 1 PER FARE CHE SIA COMUNQUE DIVERSA DA 0
if(n_onda_inizio!=0&&tipo==2)
    n_onda_fine=1;
//DA QUI RICOMINCIA SE IL TERZO INSERIMENTO NON E' CORRETTO
ONDAFINE:
//SE DEVO FARE UNA SCANSIONE
if(n_onda_inizio!=0&&n_onda_fine==0&& tipo==1){
    if(aggiornato==1){
        scrivereLCD("lambda end?",onda_fine);
        Line2LCD();
        if(primo==4){
            putsUSART("lambda end?");
            primo++;
        }
        for(i=1;i<=n_riga;i++) //il cursore si posiziona sotto il numero
            ShiftCursorLCD(1);
        aggiornato=0;
    }
    if(n_onda_fine==0&&(pulsante==10||pulsante==11||pulsante==12||pulsante==13))
        taratura(&onda_fine);
    if(pulsante==9){
        n_onda_fine=atof(onda_fine);
        pulsante=15;
        aggiornato=1;
        n_riga=0;
    }
    if(parola_pronta==1){
        strcpy(onda_fine,parola);
        parola_pronta=0;
        n_onda_fine=atof(onda_fine);
        aggiornato=1;
        n_riga=0;
        message[0]='0';
    }
    //VERIFICA CHE IL NUMERO SIA COERENTE E NON CI SIANO ERRORI
    if(n_onda_fine!=0&&(460000>n_onda_fine|| n_onda_fine>640000)){
        strcpy(onda_fine,zeri);
        n_onda_fine=0;
        primo=4;
        goto ONDAFINE;
    }
}
//SE HO LE INFORMAZIONI NECESSARIE E NON HO ANCORA CALCOLATO I PASSI
if(n_onda_fine!=0&&n_onde_ragg==0){
    //CALCOLO IL RANGE PER ARRIVARE ALL'INIZIO DELLA SCANSIONE (0 RAGGIUNGERE)

```

```

    n_onda_ragg=n_onda_inizio-n_onda_adesso;
    //SE DEVO FARE SCANSIONE CALCOLO IL RANGE DI SCANSIONE
    if(tipo==1)
        n_onda_sc=n_onda_fine-n_onda_inizio;
    Delay10KTCYx(200);
    Delay10KTCYx(200);
}
if(n_onda_ragg!=0 && n_passi_ragg==0){
    result=round(n_onda_ragg/ONDSTEP);
    n_passi_ragg=trunc(result); //calolo quanti passi per raggiungere
    if(tipo==1){
        result=round(n_onda_sc/ONDSTEP);
        n_passi_sc=trunc(result); //calcolo quanti passi per scansione
        sprintf(riga,"%li passi per scansione \r\n",n_passi_sc);
        putsUSART(riga);
    }
    if(n_passi_ragg>=0) //valuto in quale direzione girare per raggiungere
        dir_ragg=1;
    else
        dir_ragg=0;
    if(n_passi_sc>=0) //valuto in quale direzione girare per scansione
        dir_sc=1;
    else
        dir_sc=0;
    sequenza++;
    primo=1;
}
}
}
-----
//QUESTA FUNZIONE VIENE INVOCATA PER MASSIMIZZARE L'USCITA DEL CRISTALLO
void massimizza(){
    if(primo==1){
        sprintf(riga,"maximizes the cristal and press enter/OK \r\n");
        putsUSART(riga);
        scrivereLCD("maximizes crist","press enter/ok");
        primo=0;
    }
    levette(0,0,1,0); //abilito solo la levetta del cristallo
    //QUANDO PREMO INVIO/OK POSSO MEMORIZZARE LA POSIZIONE
    if(message[0]==13||pulsante==9){
        if(cristallo_primo==1){
            passi_C=0;
            passi_G=0;
            x_calibrazione[0]=0; //il primo punto è (0;0)
            y_calibrazione[0]=0;
            passi_div_sc=labs(trunc(n_passi_sc/4));
            sprintf(riga,"passipolinomio %li",passi_div_sc);
            putsUSART(riga);
            sequenza++;
            primo=1;
            cristallo_primo=2; //il prossimo massimizza non deve passare di qua
        }
        //GLI ALTRI 4 PUNTI VENGONO MEMORIZZATI TENENDO CONTO DEI PASSI EFFETTUATI
        else{
            x_calibrazione[div_sc]=passi_G;
            y_calibrazione[div_sc]=passi_C;
        }
        message[0]='0';
        pulsante=15;
        div_sc++; //passo al punto successivo
        fatto=0;
        primo=1;
        i=1;
    }
}

```

```

    }
}
-----
//QUESTA FUNZIONE SERVE PER CALCOLARE LA RETTA D'INTERPOLAZIONE USANDO
//I 5 PUNTI MEMORIZZATI CON LA FUNZIONE MASSIMIZZA
void interpolazione (){
    for(i=0;i<5;i++){
        x=x+x_calibrazione[i];
        y=y+y_calibrazione[i];
        sqrt_xy=sqrt_xy+x_calibrazione[i]*y_calibrazione[i];
        sqrt_x=sqrt_x+x_calibrazione[i]*x_calibrazione[i];
    }
    b=((sqrt_x*y)-(sqrt_xy*x))/(5*sqrt_x-(x*x)); //termine noto
    a=((5*sqrt_xy)-(x*y))/(5*sqrt_x-(x*x)); //coefficiente moltiplicativo di x
    sprintf(riga,"a %f",a);
    putsUSART(riga);
    sprintf(riga,"b %f",b);
    putsUSART(riga);
}
-----
//QUESTA FUNZIONE VIENE UTILIZZATA PER EFFETTUARE LE MOVIMENTAZIONI NEL CASO
//DI FL2002 CON SOLO GRATING O DI MONOCROMATORE
void muovere (int motore){ //motore -> 1=Grating 0=Monocor
    //SE DEVO ANCORA RAGGIUNGERE LA POSIZIONE
    if(primo==1 && passaggio==0){
        NOlevette=1; //levette disabilitate
        passi_E=0;
        passi_C=0;
        i=0;
        sprintf(riga,"going to start");
        putsUSART(riga);
        scrivereLCD(riga,"");
        //VALUTO QUALE MOTORE MUOVERE ED ASSEGNO AD UN PUNTATORE LA RISPETTIVA
        //VARIABILE PASSI
        if(motore==1){
            passi_fatti=&passi_G;
            *passi_fatti=0;
        }
        if(motore==0){
            passi_fatti=&passi_M;
            *passi_fatti=0;
        }
        DIR=dir_ragg; //utilizzo la direzione calcolata in inizializzazione
        primo++;
    }
    //FACCIO ESEGUIRE UN PASSO AL MOTORE INDICATO
    if(passaggio==0){
        accelerazione(vel_pot);
        if(motore==1&&RESET_DRV_MOT==1){
            passo_G(vel_pot);
        }
        if(motore==0&&RESET_DRV_MOT==1){
            passo_M(vel_pot);
        }
        //QUANDO I PASSI EFFETTUATI SONO UGUALI A QUELLI CALCOLATI PER RAGGIUNG
        if(abs(*passi_fatti)>=labs(n_passi_ragg)){
            passaggio=1; //vado al passaggio dopo
            primo=1;
        }
    }
}
if(n_operazione2==1 && passaggio==1){
    if(primo==1){
        *passi_fatti=0; //per contare solo quelli della scansione
    }
}

```

```

        DIR=dir_sc; //uso la direzione per la scansione
        primo++;
    }
    if(passaggio==1){
        //FACCIO ESEGUIRE UN PASSO AL MOTORE INDICATO
        if(motore==1&&RESET_DRV_MOT==1){
            passo_G(vel_min_G);
            Delay1KTCYx(vel_min_G);
        }
        if(motore==0&&RESET_DRV_MOT==1)
            passo_M(vel_min_M);
        percentuale(*passi_fatti);//calcolo la percentuale
        //SE SONO ARRIVATO ALLA FINE DELLA SCANSIONE
        if(abs(*passi_fatti)>=labs(n_passi_sc)){
            passaggio=2;
            fine=1; //indico che ho finito
            primo=1;
        }
    }
}
//SE DOVEVO SOLO RAGGIUNGERE NON SERVE CONTINUARE
if(n_operazione2==2&&passaggio==1){
    sprintf(riga,"finish");
    putsUSART(riga);
    scrivereLCD(riga,"");
    fine=1; //dico che ho finito
    primo=1;
}
if(fine==1) //se ho finito
    ripetere(); //chiedo se si voglia ripetere
}
-----
//QUESTA FUNZIONE CHIEDE SE SI VOGLIA RIPETERE LA SCANSIONE E PREPARA IL
//DISPOSITIVO IN POSIZIONE
void ripetere(){
    scritto=0;
    vai=0;
    NOlevette=0;
    rip=0;
    //IN TUTTI I CASI POSSIBILI TRANNE PER IL SEMPLICE RAGGIUNGERE
    if(primo==1 && (n_operazione2==1||n_operazione==2||n_operazione==3)){
        putsUSART("press enter/OK for repeat");
        scrivereLCD("press enter/OK"," for repeat");
        primo++;
        message[0]='0';
    }
    if(message[0]==13||pulsante==9){ //se viene premuto invio/OK
        rip=1; //segnalo che si vuole ripetere
        message[0]='0';
        pulsante=15;
    }
    if(rip==1){ //se si vuole ripetere salvo i passi fatti
        tornare[0]=passi_G;
        tornare[1]=passi_E;
        tornare[2]=passi_C;
        tornare[3]=passi_M;
        DIR=abs(dir_sc-1);
        for(i=1;i<=(abs(tornare[0])+30);i++){//+30 per poi essere nel verso
            //giusto col grating che ha gioco
            passo_G(10);
        }
        for(i=1;i<=abs(tornare[3]);i++){
            passo_M(10);

```

```
    }
    for(i=1;i<=abs(tornare[2]);i++){
        passo_C(10);
    }
    DIR=dir_sc;
    for(i=1;i<=abs(tornare[1]);i++){
        passo_E(10);
    }

    passaggio=1;
    fine=0; //non serve più entrare nella funzione "ripetere"
    primo=1;
}
}
```