

SMART WINDOW

LA VENEZIANA INTELLIGENTE

Progetto realizzato da:

- ❖ Anis Arfaoui
- ❖ Eduardo Gallina
- ❖ Riccardo Gatteri

IIS B. CASTELLI 2014/2015

1. INTRODUZIONE

- 1) DOMOTICA
 - Cos'è?
 - Vantaggi
 - Svantaggi
 - Aree di utilizzo
- 2) MOTIVAZIONE SCELTA DEL PROGETTO
- 3) WHY WE CHOSE THIS PROJECT?

2. SCHEMA A

BLOCCHI

- 1) RAPPRESENTAZIONE
- 2) SPIEGAZIONE SINGOLO BLOCCO
 - Arduino
 - Motore corrente continua
 - Servomotore
 - Controllo motore con arduino e L293D
 - Sensori
 - Visualizzazione (Display LCD)

3. FUNZIONAMENTO

- 1) SPIEGAZIONE HARDWARE
- 2) SPIEGAZIONE SOFTWARE

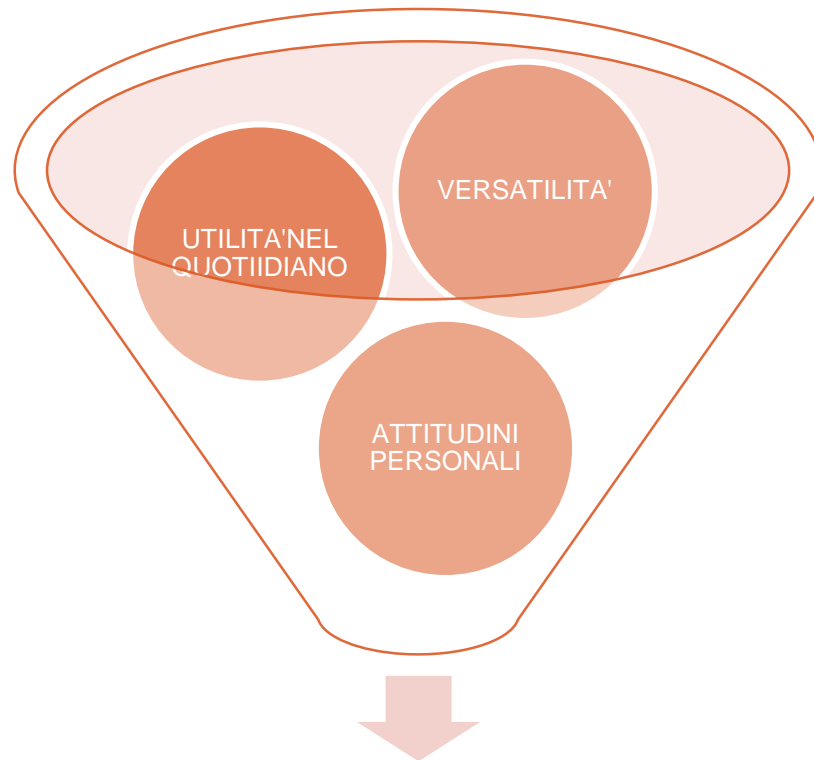
1. INTRODUZIONE

1) DOMOTICA

- Cos'è?
- La domotica è la scienza interdisciplinare che si occupa dello studio delle tecnologie atte a migliorare la qualità della vita nella casa. Quest'area richiede l'apporto di molte tecnologie e professionalità. L'obiettivo della domotica è il totale controllo di servizi e apparecchiature e la realizzazione di nuove operazioni altrettanto complesse.
- Vantaggi
- Versatilità: È possibile disporre i dispositivi di comando in zone non consentite dagli impianti tradizionali.
- Sicurezza: La riduzione della sezione del cablaggio comporta una conseguente riduzione del carico infiammabile, rendendo così l'intero sistema più sicuro.
- Adattabilità: Un impianto domotico è adatto a qualunque costruzione, che sia nuova o vecchia, ed è inoltre possibile utilizzare il cablaggio esistente modificandolo e integrandolo.
- Controllo remoto: È possibile gestire l'impianto domotico a distanza.
- Flessibilità: L'impianto è costruito in modo tale da funzionare comunque nonostante ci sia un guasto in uno dei dispositivi. Ciò dipende dal fatto che il sistema è costruito attraverso connessioni logiche.
- Risparmio energetico: Le automazioni consentono diversi modi per risparmiare energia gestendo l'efficienza dei dispositivi che compongono l'impianto. Ogni settore può essere tranquillamente disattivato o programmato (ad esempio per fasce orarie) per l'accensione o lo spegnimento. Inoltre, se è presente anche un impianto di produzione di energia, quale fotovoltaico, geotermico o altro, la Domotica può monitorarne l'efficienza e segnalarne qualunque anomalia.
- Svantaggi
- Costi: I dispositivi domotici sono naturalmente più alti rispetto a quelli dei dispositivi tradizionali. Però, la multifunzionalità di alcuni dispositivi, insieme alla riduzione del cablaggio, annullando in qualche modo la differenza. Inoltre, partendo dal presupposto che l'impianto domotico presenta vantaggi anche nell'ambito della gestione energia, è probabile che, nonostante il costo più elevato, l'impianto stesso permetta poi di guadagnare in qualche modo.
- Specializzazione addetti: Per quanto la Domotica sia in rapida espansione, è comunque vero che molti tecnici tra installatori e progettisti non conoscono l'ambito della "home automation".
- Disinformazione: Ancora non esiste una cultura domotica radicata nell'opinione pubblica. Pertanto, il progettista o installatore domotico deve mettere davanti al cliente i vantaggi e gli svantaggi del sistema, svolgendo un'azione educativa e spesso persuasiva.
- Aree di utilizzo
- Gestione dell'ambiente → Il controllo dell'ambiente viene automatizzato grazie alla presenza di un sistema di sensori e attuatori, questo permette la termoregolazione dei singoli locali abitativi in funzione dei cambiamenti ambientali con in più una costante verifica dei consumi energetici

- Gestione degli apparecchi → Il settore degli apparecchi domestici è quello in maggior crescita, grazie al miglioramento delle prestazioni dovuto all'introduzione di componenti elettronici di ultima generazione. Tutto ciò permette la telegestione e la telediagnostica manutentiva di ogni singolo apparecchio.
- Comunicazione e informazione → Un gateway fornisce una connessione a banda larga per permettere l'accesso ad internet alle apparecchiature domotiche per la trasmissione dei dati e per il controllo remoto (DSL, fibra ottica).
- Sicurezza → In quest'ultima categoria rientrano funzioni particolarmente richieste dall'utenza. La sicurezza si compone di security, ovvero sicurezza contro intrusioni non autorizzate o rapine, e di safety, ovvero sicurezza globale della casa contro fughe di gas, incendi, allagamenti o altri eventi dannosi. Degli esempi possono essere la gestione degli accessi, protezione antifurto e antincendio, videocontrollo, telesoccorso e teleassistenza.

2) MOTIVAZIONE SCELTA PROGETTO



SMART WINDOW

Si è deciso di realizzare la veneziana intelligente per tre motivi:

- VERSATILITA' → Il progetto realizzato può essere definito versatile in quanto è applicabile in diversi campi. Il principio di funzionamento della veneziana intelligente infatti può essere usato per altri dispositivi come cancelli automatici o serrande.
- UTILITA' NEL QUOTIDIANO → La smart window è utile perché facilita la vita quotidiana limitando gli sforzi fisici e perdite di tempo. Questo progetto ricalca pienamente le innovazioni nell'ambito della domotica.
- ATTITUDINI PERSONALI → Un altro motivo per cui si è deciso di progettare la veneziana intelligente è perché è adatto a noi e alle nostre capacità. Si è deciso dunque di realizzare la smart window in quanto sicuri di poter presentare un lavoro funzionante, essendo in grado di scriverne il software, di montare la parte hardware e di spiegarlo.

3)WHY WE CHOSE THIS PROJECT

This year for the state exam we had to to present a project.

Considering the skills we have developed over the secondary school years, we chose a project we knew we would be able to make on our own.

It was important to choose a makeable project because we would have to explain to our teachers how we had made it.

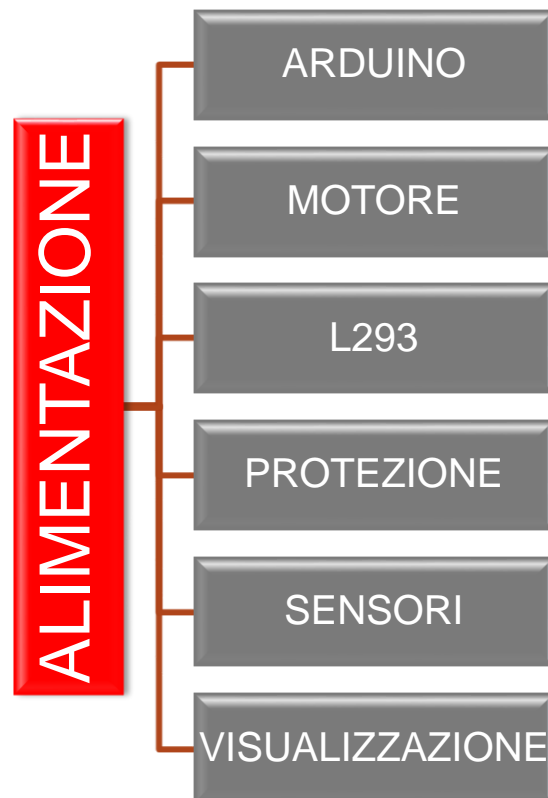
The project consists in a smart window which goes up and down in two modes: manual or automatic. The manual mode consists in pressing two buttons to open or close it; the automatic mode instead measures the light level and works according to it.

We chose Arduino as the microprocessor because it is easy to use and we had already used it.

The smart window is versatile because its program can control other things like gates or shutters. In addition, we can modify the function of the circuit changing the program written on the microprocessor Arduino. This means that we can add new functions giving instruction to the microprocessor, like making a diode flash when the motor is running.

We chose this project also because the smart window can be useful in our lives; as a matter of fact pressing a button to raise the windows is more convenient than doing it manually.

2. SCHEMA A BLOCCHI



1) ARDUINO

Per la realizzazione del circuito è stato utilizzato Arduino. Arduino è una scheda elettronica di piccole dimensioni con un microcontrollore e circuiteria di contorno, utile per creare rapidamente prototipi e per scopi hobbistici e didattici. Con Arduino si possono realizzare in maniera relativamente rapida e semplice piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, temperatura e umidità e molti altri progetti che utilizzano sensori, attuatori e comunicazione con altri dispositivi. È fornito di un semplice ambiente di sviluppo integrato per la programmazione. Tutto il software a corredo è libero, e gli schemi circuitali sono distribuiti come hardware libero.

Arduino comprende una piattaforma hardware, questo si basa su un circuito stampato che integra un microcontrollore con pin connessi alle porte I/O, un regolatore di tensione e quando necessario un'interfaccia USB che permette la comunicazione con il computer. A questo hardware viene affiancato un ambiente di sviluppo integrato (IDE) multipiattaforma (per Linux, Apple Macintosh e Windows). Questo software permette anche ai novizi di scrivere programmi con un linguaggio semplice e intuitivo derivato da C e C++ chiamato Wiring, liberamente scaricabile e modificabile. I programmi in Arduino vengono chiamati sketch. La piattaforma hardware Arduino è spesso distribuita agli hobbisti in versione pre-assemblata, acquistabile in internet o in negozi specializzati.

La particolarità del progetto è che le informazioni sull'hardware, e soprattutto i progetti, sono disponibili per chiunque: si tratta quindi di un hardware open source. Questa possibilità ha consentito lo sviluppo di prodotti Arduino compatibili da parte di piccole e medie aziende in tutto il mondo: è quindi divenuto possibile scegliere tra un'enorme quantità di schede Arduino-compatibili. Ciò che accomuna questi prodotti inerenti elettronica sperimentale e sviluppo è il codice sorgente per l'ambiente di sviluppo integrato e la libreria residente che sono resi disponibili, e concessi in uso, secondo i termini legali di una licenza libera.

Grazie alla base software comune ideata dai creatori del progetto, per la comunità Arduino è stato possibile sviluppare programmi per connettere, a questo hardware, più o meno qualsiasi oggetto elettronico, computer, sensori, display o attuatori. Dopo anni di sperimentazione, è oggi possibile fruire di un database di informazioni vastissimo.

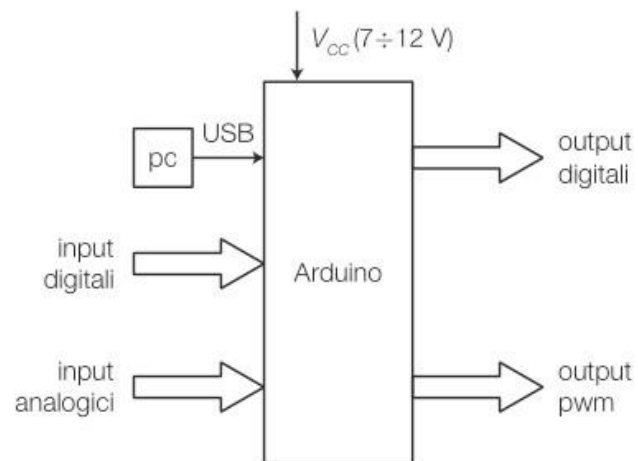
▪ I pregi principali di questo sistema sono:

- L'economicità dell'hardware e la gratuità del software, scaricabile dal sito web di Arduino
- La scheda contiene già tutto ciò che serve per un utilizzo di base.
- La semplicità di collegamento alla scheda, sia verso il computer (USB) sia verso gli altri circuiti elettronici (mediante pin messi a disposizione sulla scheda).
- La relativa semplicità della programmazione in C, che non richiede quindi la conoscenza del linguaggio assembler del microcontrollore.
- La grande disponibilità sul web di programmi (detti sketch), per svolgere le più svariate funzioni, o di progetti interamente sviluppati.
- La disponibilità di varie librerie di funzioni che semplificano il pilotaggio di display LCD, motori, trasduttori, servomotori, ecc.
- La disponibilità di schede aggiuntive (dette shield), realizzate da vari produttori, per interfacciare la scheda Arduino a motori, a carichi di potenza, a trasmettitori wireless, alla rete internet, ecc.

▪ Dal punto di vista funzionale la scheda Arduino presenta:

- La presa USB per il collegamento al computer, durante la programmazione e la messa a punto del software.
- La presa per l'alimentazione (7 -12 V), per il funzionamento autonomo (stand-alone) una volta che la scheda viene scollegata dal computer e inserita nel circuito a cui è destinata; un regolatore di tensione all'interno produce la 5 V stabilizzata necessaria ai vari integrati.
- Ingressi digitali (a livelli TTL) individuati, mediante opportuna programmazione, tra i 14 pin digital.
- Ingressi analogici (analog in): le tensioni presenti su questi sei ingressi sono acquisite e convertite in digitale (a 10 bit) mediante opportune istruzioni.
- Uscite digitali individuate, mediante opportuna programmazione, tra i 14 pin digital.
- Uscite PWM (Pulse Width Modulation): alcuni dei pin digital (3, 5, 6, 9, 10, 11), se definiti come uscite, possono produrre segnali PWM mediante opportune istruzioni, per il controllo di alcuni attuatori come motori, lampade, riscaldatori, servomotori, ecc. La modulazione a larghezza d'impulso (PWM) modifica il duty-cycle di un'onda rettangolare (dallo 0% al 100%) in base al valore della grandezza modulante (in questo caso un numero compreso tra 0 e 255). L'informazione contenuta in un segnale PWM è di tipo analogico, per questo l'istruzione che imposta il segnale su uno dei pin dedicati è "analogWrite", nonostante tali pin siano tra quelli definiti "digital".

- Qua di seguito vengono rappresentati i principali collegamenti di Arduino:



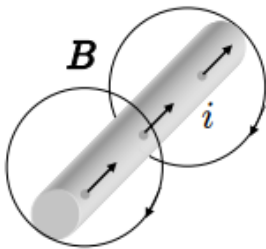
2) MOTORE

Prima di iniziare a descrivere il funzionamento dei motori in corrente continua è necessario descrivere i principi fisici utili per comprendere questo argomento.

- Legge di Lenz → Se in un circuito il flusso concatenato varia nel tempo si genera una tensione

$$V = -\frac{d\Phi}{dt}$$

- Legge di Biot-Savart → Un conduttore percorso in intensità i genera un campo magnetico di intensità B



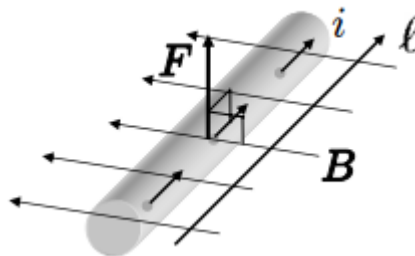
$$B = \frac{\mu_0 i}{2\pi r}$$

$\mu_0 = 4\pi 10^{-7}$
permeabilità magnetica nel vuoto

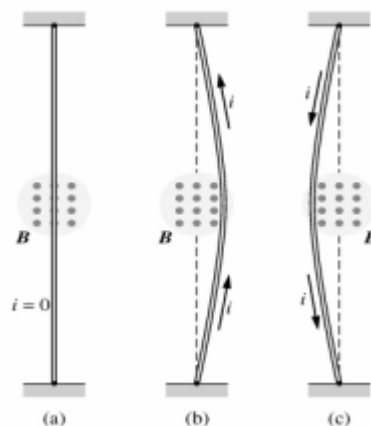
$$\left(\underline{B} = \frac{\mu_0 i}{2\pi r^2} \underline{s} \wedge \underline{r} \right)$$

Unità di misura di B nel sistema SI
tesla ($T=NA^{-1}m^{-1}$)

- Legge di Faraday → Una carica q in moto con velocità v e immersa in un campo magnetico di intensità B è sottoposta ad una forza $F = qv \wedge B$



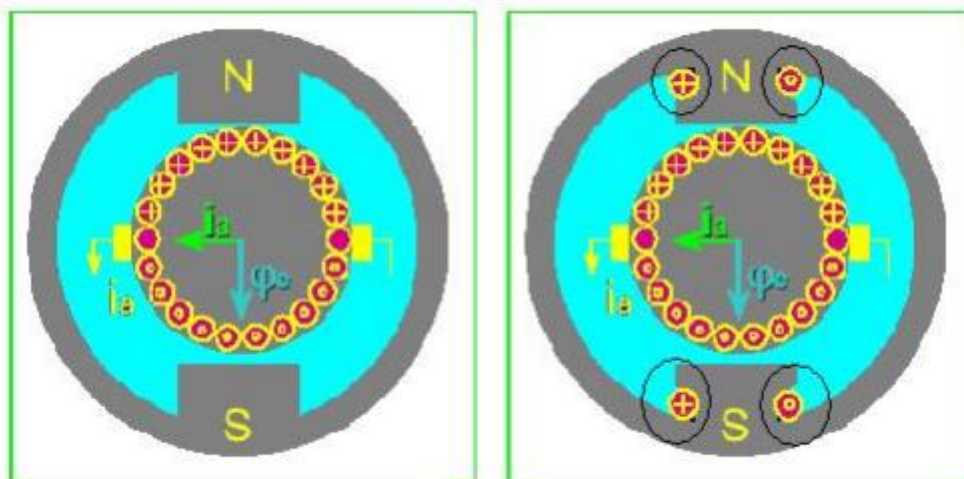
- Un conduttore di lunghezza l percorso da corrente i e immerso in un campo magnetico di intensità B è sottoposto ad una forza $F = li \wedge B$



INTRODUZIONE

Dopo aver fatto dei piccoli cenni di fisica generale è possibile iniziare a descrivere i motori a corrente continua.

La macchina a corrente continua trasforma energia elettrica sotto forma di corrente continua, in energia meccanica o viceversa. Essa ha la struttura di una macchina rotante ed è composta da uno statore esterno fisso e da un rotore che ruota all'interno dello statore.



statore a magnete permanente

statore alimentato

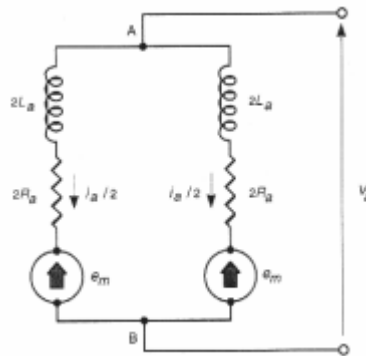
Questa tipologia di motori può avere due diversi modelli di statore, si distinguono infatti gli statori a magneti permanenti e gli statori alimentati. Entrambi hanno come scopo fondamentale quello di creare attorno a sé un campo magnetico costante. Cos'è un campo magnetico? Un campo magnetico è un campo vettoriale solenoidale generato nello spazio dal moto di una carica elettrica o da un campo elettrico variabile nel tempo.

Con il primo tipo di statore, ovvero quello a magneti permanenti il valore del campo magnetico che si andrà a generare non è modificabile in alcun modo in quanto sono esclusivamente i due magneti presenti a creare il campo magnetico stesso. La seconda tipologia è differente dalla prima, anch'essa dev'essere alimentata con una corrente continua per fare in modo che si crei comunque un campo magnetico costante, però in questo caso è possibile regolare il campo magnetico stesso che viene a crearsi (il circuito statorico deve essere avvolto tra su blocchi di materiale ferromagnetico). I valori del campo magnetico possono essere modificati regolando la tensione ai capi del circuito statorico.

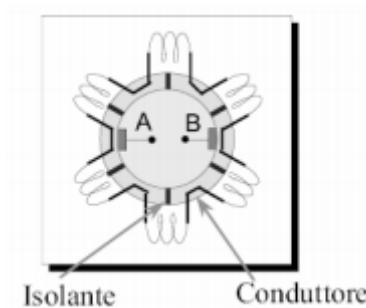
La macchina è reversibile, cioè la stessa macchina può funzionare indifferentemente nei due sensi di trasformazione energetica: quando trasforma energia elettrica in meccanica assume il nome di motore, mentre quando trasforma energia meccanica in elettrica è detta dinamo. Nel funzionamento come motore si fornisce una potenza elettrica ai morsetti della macchina, mentre dall'albero si preleva una potenza meccanica.

Per quanto riguarda la dinamo l'albero viene mantenuto in rotazione da un motore primo (come un diesel o una turbina) che fornisce la potenza meccanica, mentre dai morsetti della macchina si preleva potenza meccanica.

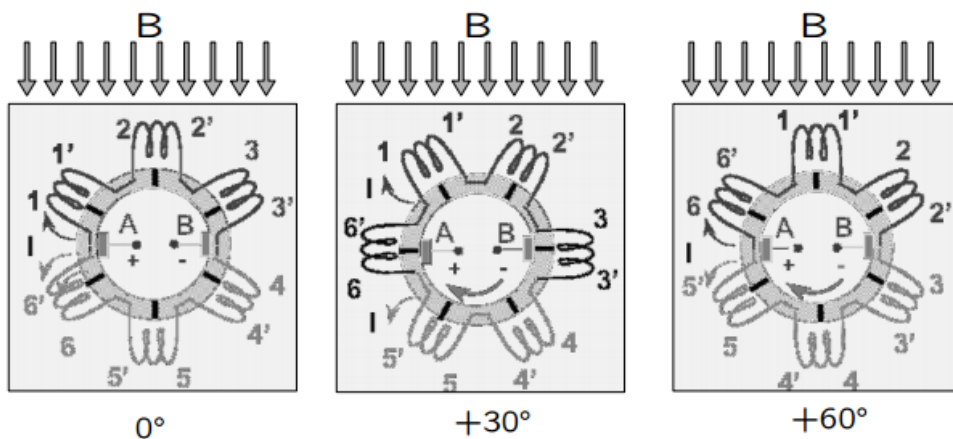
Attraverso le spazzole A e B il circuito viene percorso da due correnti circa uguali in modulo (determinano la corrente complessiva di armatura).



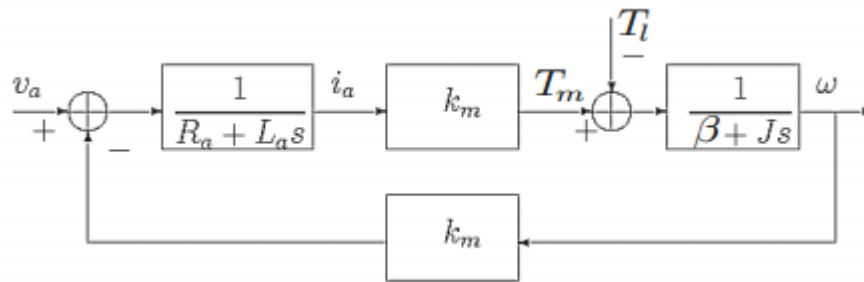
Il rotore è un cilindro di materiale ferromagnetico laminato e sagomato in modo tale da poter presentare una serie di cave disposte assialmente, all'interno di queste cave sono collocati avvolgimenti che prendono il nome di armature e vengono collegate in serie l'una con l'altra tramite l'uso di un dispositivo detto collettore. Il tutto funziona grazie al campo magnetico di eccitazione che si concatena con quello di armatura producendo una coppia che muove il rotore.



Grazie alla figura proposta di seguito, è possibile notare i collegamenti degli avvolgimenti di rotore e l'azionamento delle spazzole e del collettore.



Qui di seguito si è voluto proporre una spiegazione del funzionamento dei motori a collettore in corrente continua utilizzando una schema a blocchi.



La prima cosa che si nota è che c'è la presenza di una retroazione negativa interna al motore, cui utilità è quella di stabilizzare la velocità del motore. Questo semplifica notevolmente il progetto di un sistema di controllo di velocità.

CIRCUITO MAGNETICO DELLA MACCHINA A C.C.

Nella macchina a corrente continua il campo magnetico è generato dallo statore. I poli sono costituiti da magneti permanenti, che molte volte viene abbreviato con PM (dall'inglese "permanent magnet"), nelle macchine di minor potenza, oppure dai nuclei in ferro eccitati da apposite bobine di campo nelle macchine di potenza maggiore.

PARTI CHE COMPONGONO UN MOTORE IN C.C. A MAGNETI PERMANENTI

- Rotore
- Magnete
- Spazzola
- Scudo o flangia
- Cuscinetto
- Albero
- Collettore
- Portaspazzola
- Saldatura tra i capi delle bobine e la lamella del collettore
- Avvolgimento
- Lamierini di rotore
- Bobina appartenente dell'avvolgimento
- Corona di statore

Tutti i conduttori sono collegati tra loro in serie in modo da formare un circuito chiuso. Sulla superficie cilindrica esterna del collettore sono appoggiate le due spazzole fisse sullo statore che hanno il compito di portare all'esterno la tensione generata nell'avvolgimento. Il contatto tra il collettore e le spazzole è strisciante ed è sede di notevoli sollecitazioni sia meccaniche che elettriche, a causa della elevata velocità relativa e del contemporaneo passaggio di corrente tra le due superfici.

Il contatto spazzola-collettore è quindi un'area molto critica nelle macchine a corrente continua e dev'essere curato con particolare attenzione

CALCOLO DELLA TENSIONE GENERATA E COSTANTE DI TENSIONE

La tensione generata da un singolo conduttore vale, come già si sa:

- $e = B \times l \times v \rightarrow$ La velocità periferica v del rotore data dal prodotto:
- $v = w \times r \rightarrow$ Dove w è la velocità angolare in rad/s ed è r è il raggio del rotore in metri. La tensione generata in un singolo conduttore risulta dunque espressa dalla formula
- $e = B \times l \times r \times w$

La tensione totale ai capi delle spazzole è ottenuta come somma dei contributi della metà dei conduttori attivi in parallelo all'altra metà.

Poiché le tensioni nei diversi conduttori attivi non sono esattamente uguali tra loro e nei conduttori in zona neutra sono addirittura nulle, dobbiamo utilizzare un valore medio della tensione di conduttore, indicata con e_m , generata da un valore medio di induzione al traferro che si indica con B_m .

La tensione generata risulta direttamente proporzionale al flusso di macchina e al numero di spire, che sono costanti costruttive della dinamo, e inoltre è proporzionale alla velocità angolare del rotore, variabile a seconda delle necessità.

Le costanti costruttive della dinamo vengono raggruppate in un unico fattore K_e , detto costante di tensione:

- $K_e = N \times B_m \times l \times r$

Di conseguenza la tensione E è misurata in V e w è misurata in rad/s, la costante di tensione risulta espressa in $\frac{Wb}{rad}$

N.B. → La costante di tensione è un parametro fondamentale della macchina a c.c. legato al valore di flusso. Nelle macchine a magneti permanenti il flusso è effettivamente costante.

La dinamo equivale a un generatore reale avente tensione a vuoto pari a E e resistenza pari a R . La caratteristica corrente-tensione della dinamo assume il nome di caratteristica esterna.

Tanto la tensione a vuoto quanto la corrente di cortocircuito $I_{cc} = \frac{K_e}{R} \times w$ sono direttamente proporzionali a w .

COPPIA RESISTENTE E COSTANTE DI COPPIA

La corrente I erogata dalla dinamo si suddivide nei due rami in parallelo dall'avvolgimento rotorico, ciascuno dei quali viene così percorso da $\frac{I}{2}$.

Poiché i conduttori percorsi da questa corrente sono immersi in un campo magnetico, vengono sottoposti ciascuno a forza elementare f , che per la regola della mano sinistra risulta tangenziale al rotore e opposta al senso di rotazione.

Il valore medio della forza elementare vale:

- $f_m = Bm \times l \times \frac{I}{2}$

Il contributo di tutti i conduttori produce la coppia complessiva:

- $C = N \times Bm \times l \times r \times I$

E si ha Bm che è l'induzione media, l la lunghezza ed r il raggio del rotore espressi in metri, mentre C è misurata in $N \times m$. La coppia, proporzionale alla corrente e diretta in senso opposto a w frena il rotore.

Il prodotto Ke per la corrente erogata è pari alla coppia da applicare all'albero della dinamo per tenerla in rotazione a velocità costante sotto carico (Ke misurata in $N \times m/A$).

Quando la costante Ke viene utilizzata per il calcolo della coppia, viene detta costante di coppia e viene indicata con Kt , quindi:

- $C = Kt \times I$

FUNZIONAMENTO A VUOTO DELLA MACCHINA A CORRENTE CONTINUA COME MOTORE

La macchina è ferma e non genera alcuna tensione al proprio interno una corrente nel suo avvolgimento la quale vale:

- $I_s = \frac{V}{R}$, dove R è la resistenza dell'avvolgimento e la I_s è detta corrente di spunto

Come già si sa la corrente e il campo magnetico generano una coppia che allo spunto risulta pari a

- $C_s = Kt \times I_s$

Per effetto di questa coppia il rotore viene sottoposto ad una accelerazione ed entra in rotazione con velocità crescente. L'avvolgimento presente all'interno del motore si muove a causa della presenza del campo magnetico in modo tale da generare la tensione E , la quale sarà direttamente proporzionale alla velocità w .

Il senso di rotazione è sempre tale da produrre una tensione E che si oppone alla tensione V . Quando il rotore raggiunge una certa velocità angolare w la corrente che entra nella macchina sarà pari a

- $I = \frac{V - K \times w}{R}$

Osservando la seguente formula si può affermare che la corrente I è più piccola della corrente I_s , ma nonostante ciò è ancora sufficiente a fornire una coppia che accelera ulteriormente il rotore. Se la macchina funziona a vuoto, ovvero che non è applicata alcuna coppia frenante, l'accelerazione prosegue fino a raggiungere una velocità elevata a sufficienza perché la tensione E assuma un valore pari a V .

Quindi, ponendo $E = K_e \times \omega_0 = V$, si può ricavare:

$$\omega_0 = \frac{V}{K_e}$$

E' stato così introdotto il funzionamento della macchina a corrente continua come motore. Il funzionamento del motore è esattamente inverso a quello della dinamo essendo la stessa macchina in esame reversibile. Si considerano dunque tre condizioni:

- Funzionamento allo spunto → Il motore è fermo e gli viene applicata una tensione.
- Funzionamento in transitorio → Durante la fase di accelerazione
- Funzionamento a vuoto → Quando il motore raggiunge la velocità ω_0 e coppia ed accelerazione si annullano.

FUNZIONAMENTO SOTTO CARICO E CARATTERISTICA ESTERNA

Se si considera la macchina elettrica funzionante a vuoto, la macchina ruota con coppia e corrente nulla (non c'è scambio di potenze tra macchina e l'esterno). Se viene applicata una coppia frenante chiamata C_L il rotore subisce un rallentamento, dunque la tensione E si riduce e circola una corrente $I = \frac{V-E}{R}$ che genera una coppia motrice C , la quale si oppone a C_L .

Il motore si stabilizza a quella velocità che gli permette di generare una coppia motrice esattamente uguale e opposta a C_L . Per permettere che la coppia motrice sia pari alla coppia frenante dev'essere presente una corrente pari a $I = \frac{C_L}{K_T}$.

La velocità corrispondente sarà dunque uguale a

$$\omega = \omega_0 - \frac{R}{K_e^2} \text{ dove } \frac{R}{K_e^2} \text{ è una costante costruttiva della macchina che prende il nome di}$$

costante del motore e viene indicata con H . Se H è troppo alto è necessario utilizzare valori più elevati di induzione (magneti permanenti di miglior qualità oppure scegliere un motore di maggiori dimensioni) perché altrimenti viene prodotta una elevata perdita di velocità all'aumentare della coppia resistente.

RENDIMENTO DEL MOTORE A CORRENTE CONTINUA A MAGNETI PERMANENTI

Come già si sa la formula del rendimento è:

$$\eta = \frac{P_u}{P_i}$$

Sapendo che P_i è pari a VI e che $\frac{R}{V}$ è uguale a $\frac{1}{I_s}$ si ottiene

$$\eta = \left(1 - \frac{I}{I_s}\right) \left(1 - \frac{I_0}{I}\right)$$

Attraverso l'analisi matematica si ricava che il valore massimo del rendimento è ottenuto per quel valore di corrente indicato con $I_{\eta M}$. Sostanzialmente è la media tra la corrente a vuoto e quella di cortocircuito.

$$I_{\eta M} = \sqrt{I_0 I_s}$$

CORRENTE E COPPIA NOMINALE

Anche per il motore in corrente continua il fattore che limita le prestazioni è la temperatura: a questo proposito valgono le stesse considerazioni riguardo le classi di isolamento, la sovratemperatura ΔT , la resistenza termica K e le perdite totali. Come nella dinamo, la corrente nominale deve essere limitata a un valore tale da contenere le perdite complessive e la conseguente sovratemperatura entro livelli accettabili.

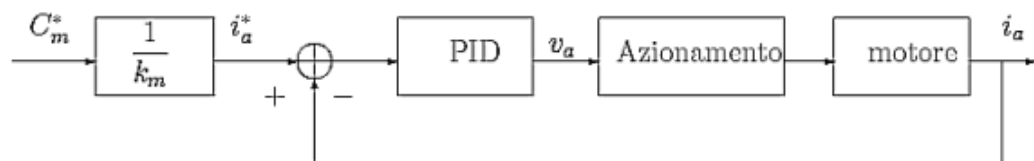
Poiché la macchina è caratterizzata da un ben preciso valore di costante di coppia K_t , determinare la corrente nominale equivale a determinare la coppia nominale.

$$C_n = \frac{1}{2\pi} N \Phi I_n$$

Le dimensioni del circuito magnetico stabiliscono il valore del flusso Φ , mentre lo spazio disponibile per l'avvolgimento stabilisce il valore della corrente concatenata NI_n . Si conclude che le dimensioni fisiche della macchina determinano il prodotto $NI_n \Phi$ cioè il valore della coppia nominale C_n .

VANTAGGI MOTORE CONTINUA

- Facilità di regolazione (controllo in tensione di armatura o in tensione del circuito di eccitazione). Il modello matematico del motore a corrente continua è molto semplice: lineare tempo invariante



SVANTAGGI MOTORE CONTINUA

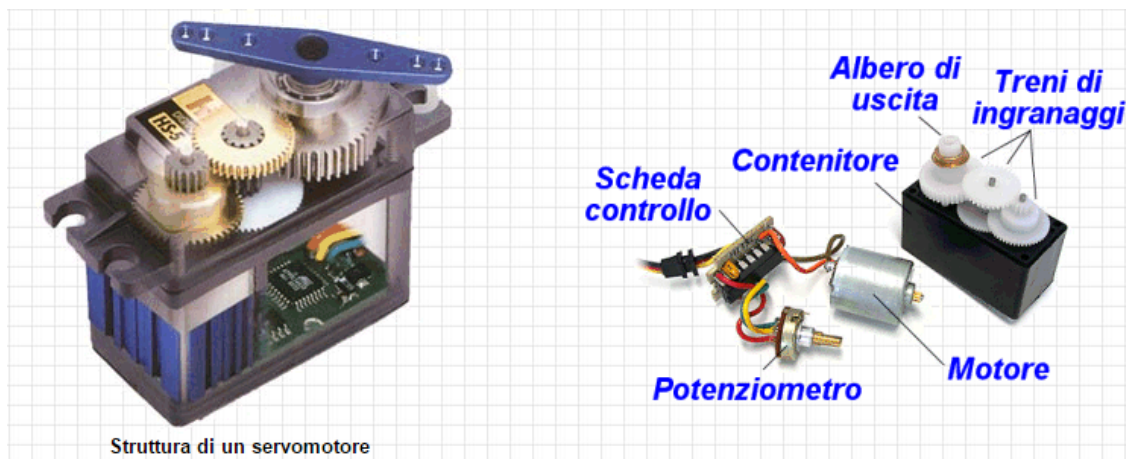
- Perdite elettriche per cadute di tensione nel contatto spazzola-collettore
- Perdite meccaniche dovute ad attrito durante la commutazione
- Produzione di scintille di commutazione (dovuta all'autoinduzione)

3) SERVOMOTORE

- Cos'è un servomotore?

Il servomotore è un particolare tipo di motore (può essere elettrico, pneumatico, idraulico) generalmente di piccola potenza, le cui condizioni operative, a differenza dei motori tradizionali, sono soggette ad ampie e spesso repentine variazioni sia nel campo della velocità che della coppia motrice, alle quali si deve adattare con la massima rapidità e precisione. Da un servomotore elettrico è generalmente richiesta bassa inerzia, elevata linearità tensione/velocità e corrente/coppia, rotazione (o traslazione) uniforme con bassa oscillazione di coppia angolare (o forza) senza posizioni preferenziali e capacità di sopportare picchi impulsivi di potenza.

I servomotori trovano applicazione nei controlli di posizione, sistemi automatici di regolazione e nelle periferiche di sistema, come stampanti e plotter. Nella robotica per gli azionamenti, sono molto utilizzati i servomotori. Di solito questi si presentano come piccoli contenitori di materiale plastico da cui fuoriesce un perno in grado di ruotare in un angolo compreso tra 0 e 180° mantenendo stabilmente la posizione raggiunta. Per ottenere la rotazione del perno è utilizzato un motore a corrente continua e un meccanismo di demoltiplica che consente di aumentare la coppia in fase di rotazione. La rotazione del motore è effettuata tramite un circuito di controllo interno in grado di rilevare l'angolo di rotazione raggiunto dal perno tramite un potenziometro resistivo e bloccare il motore sul punto desiderato.



Struttura di un servomotore

- Come si usa un servomotore?

I Servomotori sono concepiti per essere pilotati nel modo più semplice possibile, eseguendo la movimentazione senza l'ausilio di circuiterie troppo complesse o l'uso di sistemi a microprocessore. Un servomotore dispone solitamente di soli tre fili attestati ad un connettore femmina per pin strip a passo 2.54mm.

Due di questi fili sono riservati all'alimentazione in corrente continua. Il positivo è di colore rosso, il negativo di colore nero, il terzo filo, normalmente di colore bianco, è riservato per il controllo del posizionamento.

Tramite il filo del controllo è necessario applicare un segnale impulsivo o PWM (Pulse Wave Modulation) le cui caratteristiche sono "quasi" univoche per qualsiasi servomotore disponibile in commercio. Per essere sicuri di riuscire a pilotare qualsiasi servomotore il nostro circuito di pilotaggio dovrà essere in grado di trasmettere al servomotore circa 50 impulsi positivi al secondo (uno ogni 20 ms) di durata variabile, in un intervallo massimo compreso tra 0.25ms e 2.75ms.

Generalmente con un impulso di durata pari a 1.5ms il perno del servomotore si pone esattamente al centro del suo intervallo di rotazione. Da questo punto, il perno può ruotare in senso antiorario se l'impulso fornito ha una durata inferiore a 1.5ms e in senso orario se l'impulso fornito ha durata superiore a 1.5ms.

Il rapporto esatto tra la rotazione del perno e la larghezza dell'impulso fornito può variare tra i vari modelli di servomotore.

Il motore e il potenziometro sono collegati al circuito di controllo e l'insieme di questi tre elementi definisce un sistema di feedback ad anello chiuso. Il circuito e il motore vengono alimentati da una tensione continua stabilizzata, in genere di valore compreso tra 4,8 V e 6,0 V, anche se molti motori sono in grado di accettare input di alimentazione fino a 7,2 V. Per far girare il motore dovete inviare un segnale digitale al circuito di controllo. In questo modo si attiva il motore che, attraverso una serie di ingranaggi, è collegato al potenziometro. La posizione dell'albero del potenziometro indica una misura della posizione dell'albero motore del servo. Quando il potenziometro raggiunge la posizione desiderata, il circuito di controllo spegne il motore.

È facile dedurre che i servomotori vengono progettati in genere per effettuare una rotazione parziale piuttosto che impostare un moto rotatorio continuo, come nel caso di un motore in continua o passo-passo. Anche se è possibile configurare un servo R/C perché ruoti in modo continuo, l'impiego fondamentale di un servo consiste nel raggiungere una posizione accurata dell'albero del motore, con movimenti compresi nell'intervallo tra 90° e 180°. Anche se questo movimento non sembra considerevole, può risultare più che sufficiente per manovrare un robot, per sollevare e abbassare le gambe, per ruotare un sensore che deve esaminare ciò che lo circonda e molto altro ancora. La rotazione precisa di un angolo da parte di un servo in risposta a determinati segnali digitali rappresenta una delle funzionalità più sfruttate in tutti i campi della robotica.

- Servo e modulazione PWM

L'albero del motore di un servo R/C viene posizionato utilizzando una tecnica PWM (Pulse Width Modulation). In un sistema di questo tipo, il servo risponde alla durata di un segnale definito all'interno di un treno di impulsi a frequenza fissa, simile a quello rappresentato nella Figura. In particolare, il circuito di controllo risponde a un segnale digitale i cui impulsi hanno una durata variabile da circa 1 ms a circa 2 ms. Questi impulsi vengono trasmessi alla velocità di 50 al secondo. La durata esatta di un impulso, espressa in frazioni di millisecondo, stabilisce la posizione del servo.

Alcuni servo consentono di variare la frequenza del segnale PWM, altri invece non funzionano correttamente oppure "tremano" nel caso in cui gli impulsi vengano inviati a frequenze diverse da 50 al secondo. Per garantire il corretto funzionamento di un servo, verificate sempre che ci siano 20 ms tra l'inizio di un impulso e quello successivo.

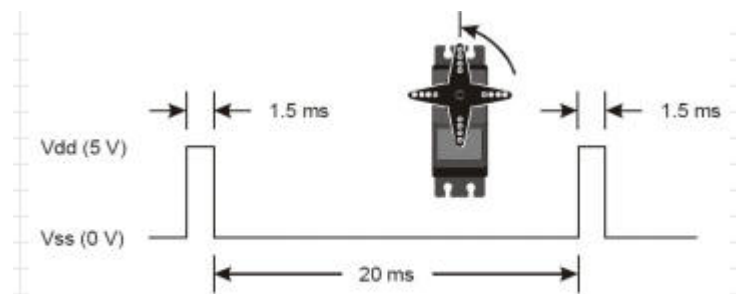


Diagramma temporizzazione per posizionamento al centro

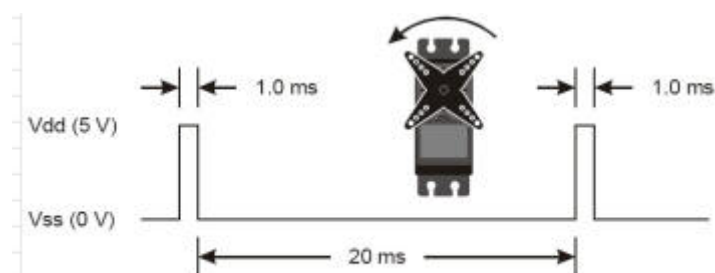


Diagramma temporizzazione per rotazione antioraria

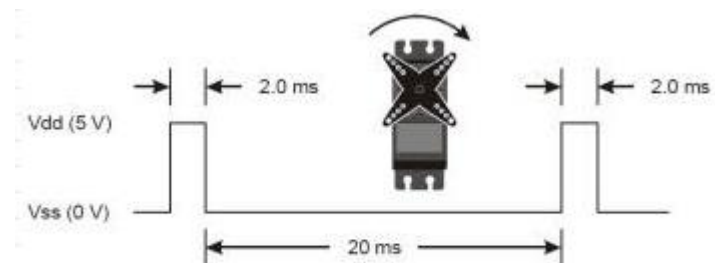


Diagramma temporizzazione per rotazione oraria

Alla durata di 1 ms, il servo viene comandato di ruotare completamente in una direzione, per esempio in senso antiorario; a 2 ms il servo ruota completamente nella direzione opposta. Di conseguenza, un impulso di 1,5 ms comanda il servo in modo da posizionarlo nella sua posizione centrale, o di riposo. È già stato detto che la posizione angolare del servo è stabilita dalla larghezza (più precisamente, dalla durata, dell'impulso di comando. Questa tecnica ha assunto negli anni parecchi nomi. Uno dei nomi più diffusi è probabilmente quello di segnale digitale proporzionale, in quanto il movimento del servo è proporzionale al segnale digitale con il quale viene attivato.

L'alimentazione fornita al motore all'interno del servo è anche proporzionale alla differenza tra la posizione attuale dell'albero e la posizione che deve raggiungere.

Se il servo deve effettuare un movimento breve per raggiungere la nuova posizione, il motore viene guidato con una velocità di rotazione bassa. In questo modo si garantisce che il motore non superi la posizione desiderata. Al contrario, se il servo deve effettuare un movimento più accentuato per raggiungere la nuova posizione, il motore viene pilotato alla massima velocità consentita, in modo da arrivare appena possibile. Il motore rallenta quando il servo si avvicina alla posizione finale. Questo processo in apparenza abbastanza complesso avviene in un breve intervallo di tempo; in genere un servo di medie dimensioni è in grado di percorrere un angolo di 60° in un tempo variabile da un quarto a mezzo secondo.

- Il ruolo del potenziometro

Il potenziometro del servo svolge un ruolo fondamentale che consente di stabilire l'istante in cui il motore ha impostato l'albero nella posizione desiderata. Questo potenziometro è fisicamente collegato all'albero di uscita del motore; in alcuni servo, l'albero del potenziometro coincide con l'albero stesso del motore. In questo modo, la posizione del potenziometro coincide precisamente con quella dell'albero del motore. Nei servo il potenziometro è configurato come un partitore di tensione e fornisce al circuito di controllo una tensione che varia in funzione della variazione dell'uscita del servo.

Il circuito di controllo del servo mette in relazione questa tensione con la temporizzazione degli impulsi digitali di ingresso e genera un segnale di errore nel caso in cui debba correggere la tensione da inviare al motore. Questo segnale di errore è proporzionale alla differenza rilevata tra la posizione del potenziometro e la temporizzazione definita dal segnale in ingresso. Per compensare questa differenza, il circuito di controllo applica al motore un segnale che tiene conto di questo errore.

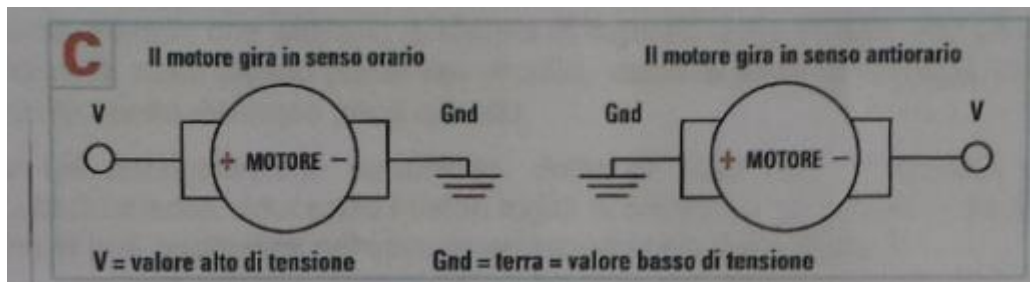
Quando la tensione del potenziometro e la temporizzazione degli impulsi digitali coincidono, il segnale di errore viene annullato e il motore si ferma.

- Tipi di collegamento

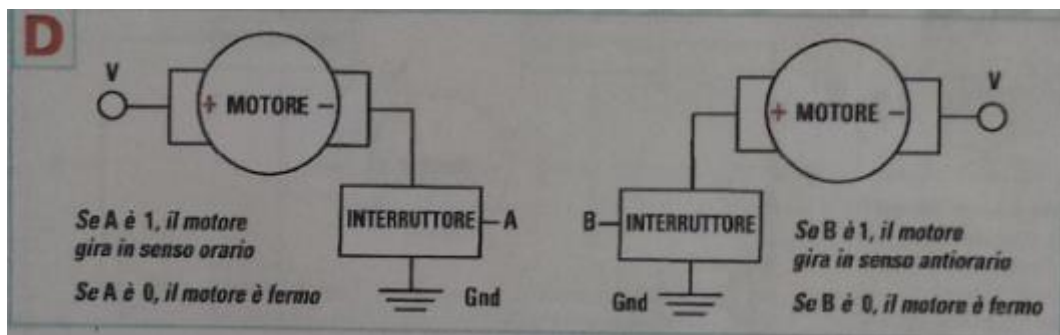
Anche se molti aspetti dei servo possono essere considerati standard, esistono molte differenze tra i prodotti di case diverse per quanto riguarda la forma e i contatti elettrici dei convertitori impiegati per collegare il servo a un ricevitore. Nonostante queste differenze, in genere potete collegare i servo a un unico convertitore a tre terminali, da saldare su un circuito stampato oppure da inserire in una breadboard. Se la presenza del convertitore originale dovesse rappresentare un problema, tagliate i fili di collegamento del servo e collegateli direttamente al circuito di controllo. Questa soluzione può essere valida come le altre, anche se rende più difficile l'eventuale sostituzione del servo, se necessaria.

4) CONTROLLO MOTORI CON ARDUINO E INTEGRATO CON L293D

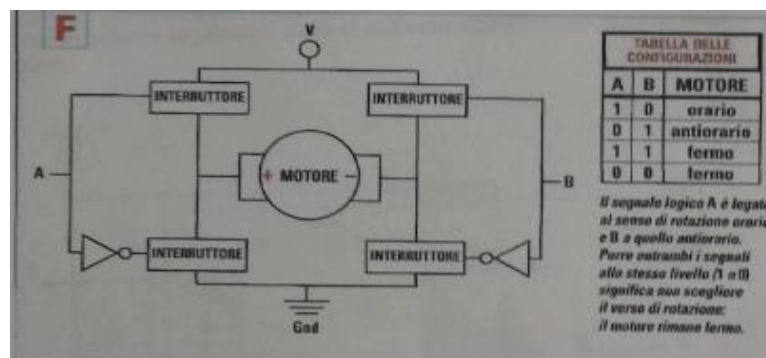
Il motore in corrente continua che è stato utilizzato per la realizzazione del nostro circuito è in grado di invertire il senso di marcia a seconda della polarità dell'alimentazione (questo è un principio comune di tutti i motori in corrente continua che si trovano solitamente nei vari laboratori di elettronica).



Per quanto riguarda il comando del motore stesso si possono utilizzare dei semplici interruttori come viene dimostrato nella foto successiva



Per poter semplificare questi due circuiti in un unico solo complessivo, si potrebbero utilizzare quattro interruttori e un motore in grado di girare in entrambi i sensi per realizzare così un circuito chiamato "a ponte H". Questa modifica viene anche fatta per evitare il "fuse test", il circuito dunque verrà modificato così:

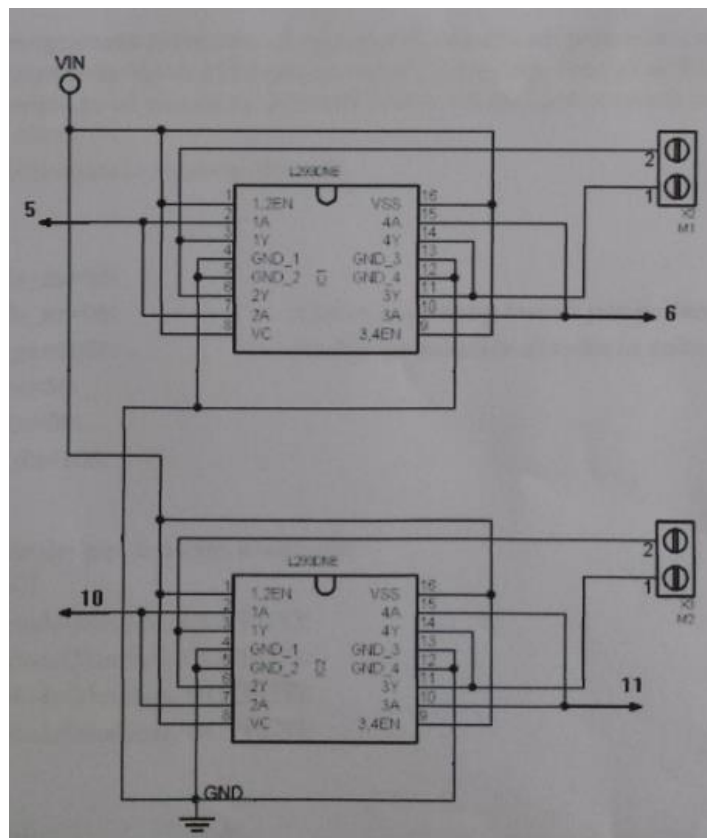


Si nota che è presente una porta NOT, la quale ha la funzione di negare ed è in grado di garantire che gli ingressi degli interruttori posti sullo stesso lato abbiano sempre e solo valori opposti.

L293D è uno degli elementi che compongono la scheda di controllo motori e che funge da intermediario tra i motori che devono essere posti in movimento e il microcontrollore che invia i comandi necessari. I driver L293 sono costituiti da 4 mezzi ponti H e sono progettati per fornire corrente bidirezionale fino a 600mA a tensioni tra 4,5V e 36V.

E' anche progettato per fare in modo che possa pilotare carichi induttivi come relè o motori in corrente continua (dispositivi ad alta I o ad alta V con alimentazione non alternata).

Questo chip può lavorare tra 0° e 70° e può essere compatibile con la logica TTL.

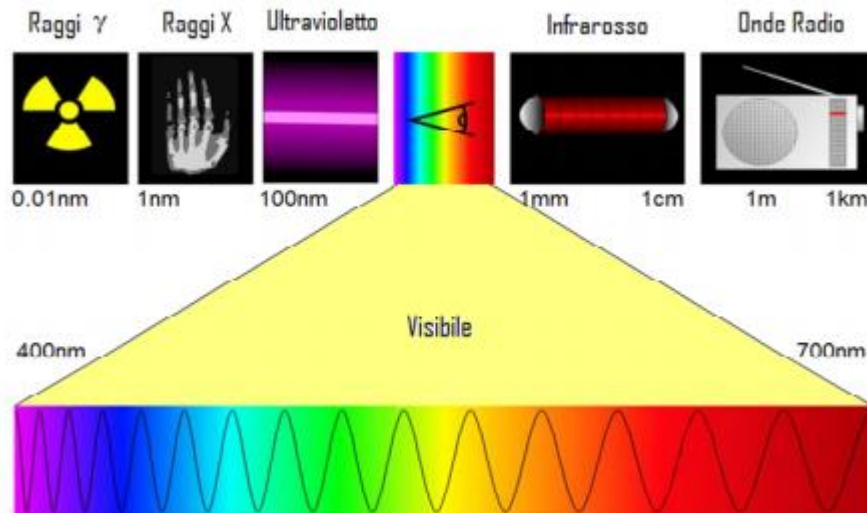


5) SENSORI

Nel circuito sono stati utilizzati dei sensori optoelettronici, come fotodiodi, fototransistor, fotoresistenze, diodi LED e display.

SENSORI OPTOELETTRONICI

Con il termine luce si intende comunemente la radiazione elettromagnetica di lunghezza d'onda tra 0.4 μm e 0.7 μm . Questa è la radiazione cui è sensibile l'occhio umano.



La lunghezza d'onda è una funzione inversa della frequenza, si ha infatti:

$$\lambda = \frac{c}{f} \text{ dove:}$$

λ = lunghezza d'onda

c = velocità della luce = 3×10^8 m/s

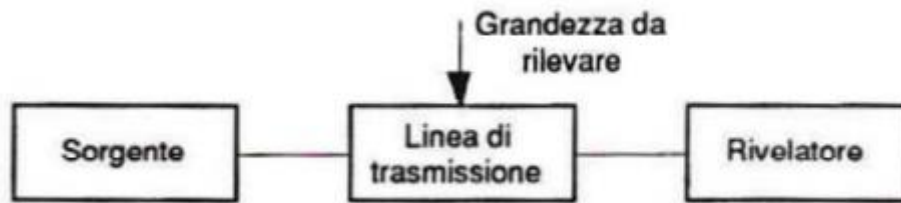
f = frequenza dell'onda

I sensori optoelettronici permettono di rilevare la presenza di oggetti di qualunque materiale, purchè interrompano il flusso luminoso. I sensori optoelettronici possono essere del tipo:

- A diffusione – il sensore risponde a qualunque oggetto che riflette la luce.
- A retroriflessione - Il sensore risponde solo alla luce riflessa da una ben determinata superficie detta riflettore detta riflettore.
- A sbarramento – Il sensore risponde a qualunque interruzione del flusso luminoso.

Gli elementi fondamentali di un sensore ottico sono:

- La sorgente (o emettitore) → nelle prime realizzazioni era una lampada ad incandescenza, oggi però è sostituito da dispositivi policromatici (a emissione non coerente) a stato solido.
- La linea di trasmissione → è realizzato con fotodiodi o con fotoresistenze, dispositivi sensibili ad uno spettro di frequenze molto ampio.
- Il fotorivelatore (o fotoricevitore) → può essere libera oppure guidata come nel caso delle fibre ottiche.



FOTOEMETTITORI

- Diodo LED

L'acronimo LED significa "Light Emetting Diode". E' costituito da una giunzione P-N, realizzata con arseniuro di gallio o con fosforo di gallio, in grado di emettere radiazioni luminose quando è attraversata da una corrente dell'ordine delle decine di mA.

Il funzionamento del LED si basa sul fenomeno detto "elettroluminescenza", dovuto alla emissione di fotoni (nella banda del visibile o dell'infrarosso) prodotti dalla ricombinazione degli elettroni e delle lacune allorché la giunzione è polarizzata in senso diretto. Poiché la luce emessa da un LED è direttamente proporzionale alla corrente che lo attraversa, tale dispositivo risulta particolarmente adatto alla trasmissione di segnali analogici tramite modulazione dell'intensità luminosa. Diversamente dalle comuni lampade a filamento un LED emette luce a bassa temperatura e può lampeggiare a frequenze superiori al Mhz.

Esistono due tipologie di LED, gli stessi LED con emissioni all'interno della fascia visibile e quelli IRED (Infra-Red Emitting Diode), con emissione nel campo infrarosso.

FOTORIVELATORI

Un fotorivelatore converte un segnale luminoso in un segnale elettrico, in genere in una tensione. Esistono fotorivelatori che sfruttano diversi principi fisici.

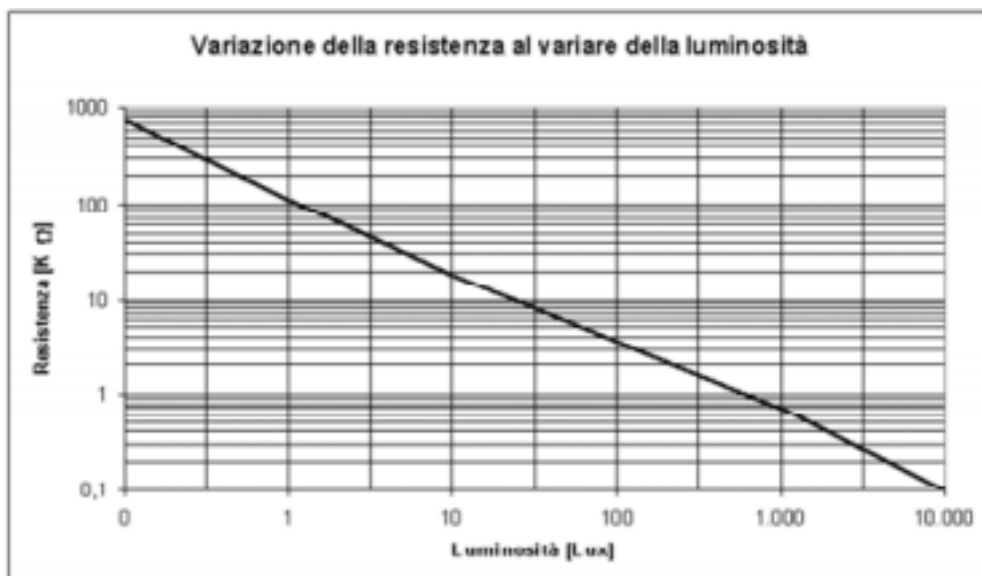
Sensore	Principio fisico
Termopile Cristalli piroelettrici	Conversione dell'energia dei fotoni in calore
Fotoresistenze Fotodiodi Fototransistori CCD	Effetto fotoelettrico interno
Diodi a vuoto Fotomoltiplicatori	Effetto fotoelettrico esterno (Emissione di elettroni per assorbimento di fotoni nei metalli)

I fotorivelatori più comuni sfruttano i primi due principi fisici, si hanno quindi:

- Fotorivelatori termici → sfruttano la conversione dell'energia dei fotoni in calore e sono caratterizzati da una sensibilità costante dall'infrarosso all'ultravioletto. Poiché si comportano da filtro passa-alto (con frequenza di taglio tra 1 Hz e 100 Hz), devono operare con luce intermittente.
- Fotorilevatori a semiconduttore → sfruttano l'effetto fotoelettrico interno, cioè la variazione di conduttività dovuta alla creazione di coppie elettrone-lacuna nelle giunzioni di semiconduttore colpite da fotoni. Hanno sensibilità variabile in funzione della frequenza, in quanto non tutta la luce incidente è in grado di produrre effetto fotoelettrico, ma solo quella di lunghezza d'onda inferiore ad un valore λ_s , dipendente dal materiale, detto soglia fotoelettrica.

FOTORESISTENZE

La fotoresistenza è un dispositivo a semiconduttore la cui conducibilità elettrica aumenta quando è colpito dalla luce. Infatti, un fotone che colpisce un elettrone impegnato in un legame covalente cede a questo la sua energia. Se l'energia ceduta è sufficiente a liberare l'elettrone dal legame si origina nel semiconduttore una coppia elettrone-lacuna in grado di partecipare alla conduzione elettrica.

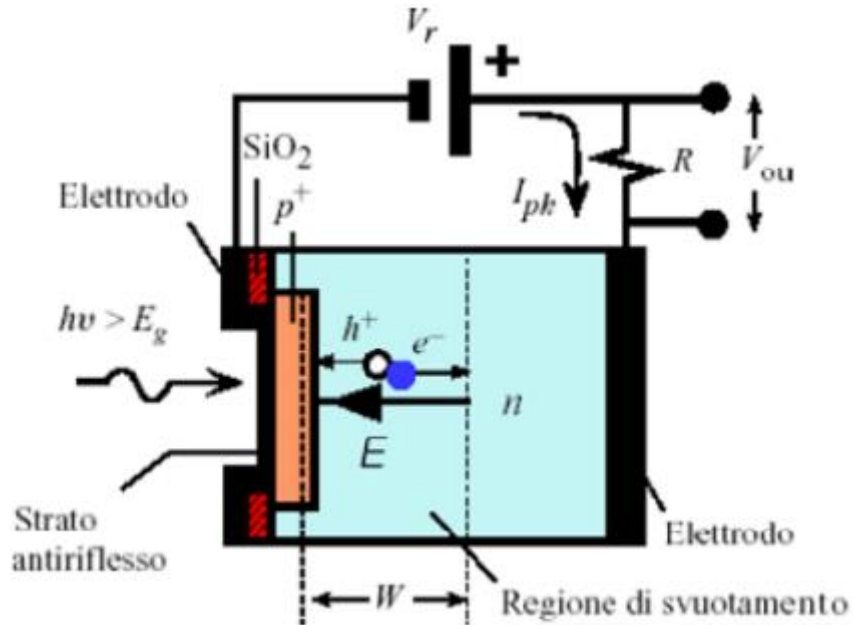


- Svantaggio:
 - Bassa velocità di ricombinazione delle coppie lacuna-elettrone generata dalla radiazione elettromagnetica.

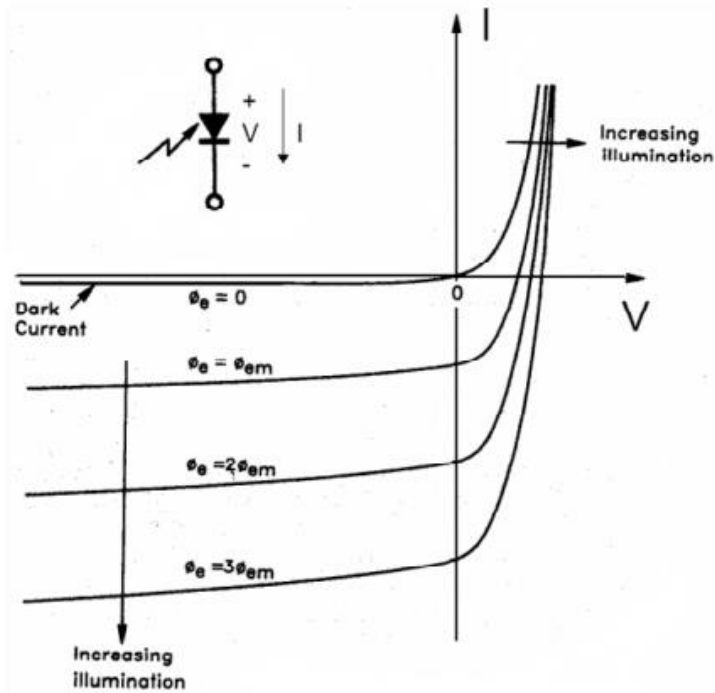
Le fotoresistenze commerciali hanno una resistenza di buio dell'ordine dei MΩ ed una resistenza pari a 100Ω quando è sottoposta a forti luminosità. Possono dissipare potenze di circa 1W o 2W.

FOTODIODI

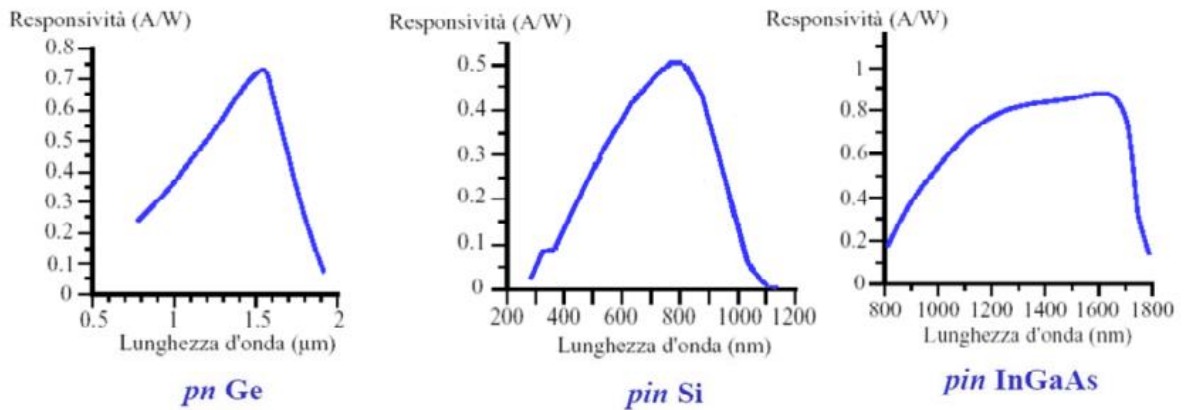
Un fotodiode è una giunzione PN drogata asimmetricamente. La zona P, infatti, è molto più drogata della zona N, è molto sottile, ed è disposta molto vicino alla struttura esterna.



L'intensità della corrente inversa generata per effetto fotoelettrico dipende dall'intensità della radiazione luminosa e dalla tensione presente ai capi del fotodiode.

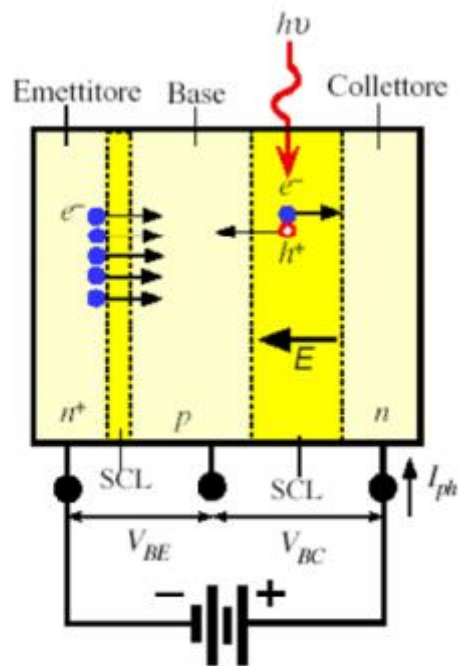


Le caratteristiche di un fotodiiodo dipendono fortemente dal materiale utilizzato.



FOTOTRANSISTOR

Un fototransistor è un transistor bipolare a giunzione con che la base lasciata non connessa. Poiché il fototransistor viene incasottato in un contenitore trasparente in modo che la luce possa raggiungere la viene incasottato in un contenitore trasparente in modo che la luce possa raggiungere la giunzione del collettore di base la fotocorrente primaria agisce come corrente di base.



$$I_{ph} \approx \beta I_{ph0}$$

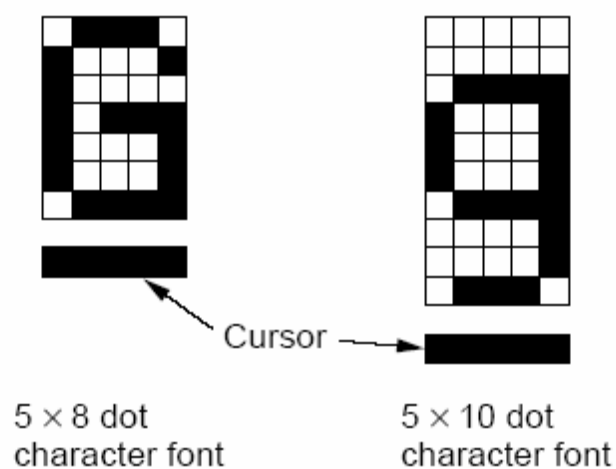
Il fototransistor ha prestazioni simili a quelle del fotodiode, ma con una sensibilità alla luce molto maggiore, grazie al fatto che gli elettroni che sono generati dai fotoni nella giunzione del collettore di base sono iniettati nella base e la corrente così prodotta viene successivamente amplificata. Per contro un fototransistor ha una risposta nel tempo molto meno rapida del fotodiode.

6) VISUALIZZAZIONE

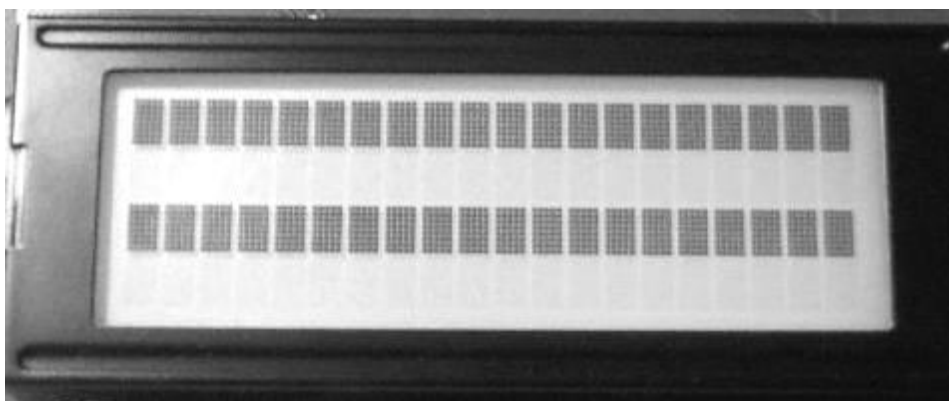
Per la visualizzazione si è deciso di utilizzare un display LCD.

I Display o visualizzatori LCD a matrice di punti sono chiamati così perché l'informazione è affidata allo stato (acceso o spento 1/0) di numerosi piccoli punti chiamati pixel organizzati su righe e colonne (matrice) successive. Il visualizzatore può essere:

- Alfanumerico: in questo caso i punti sono predisposti in gruppi di 5x10 o 5x8 pixel, quindi, accendendo opportunamente i puntini, ogni gruppo può assumere l'aspetto di un carattere; di solito questi visualizzatori mostrano una o più righe di caratteri, fino a 40 caratteri per riga.



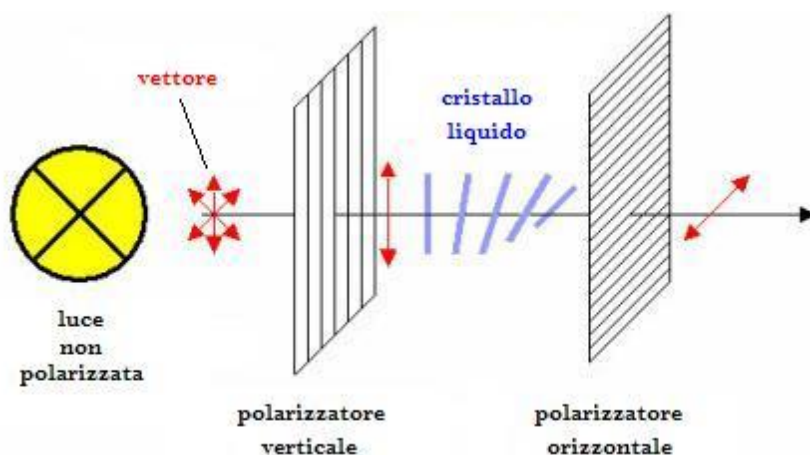
Esempio di un display alfanumerico 20x4 con 5x10 gruppi di pixel



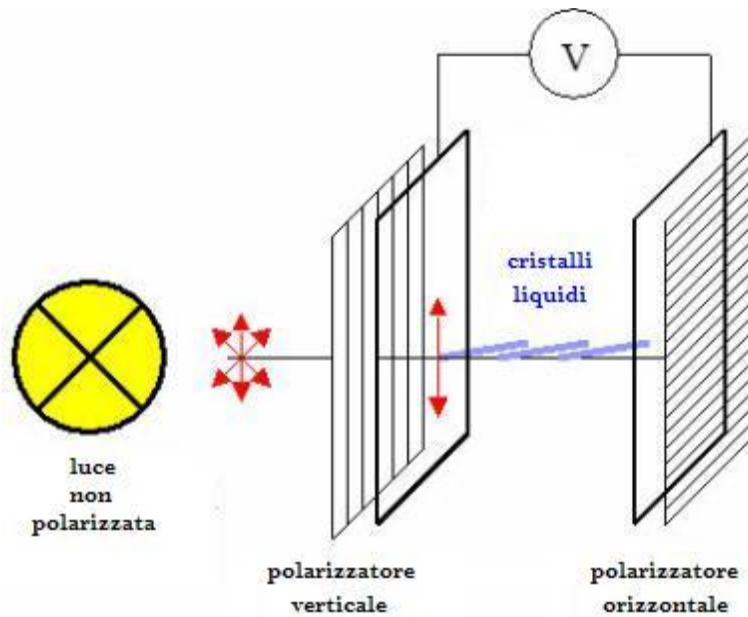
FUNZIONAMENTO

Il funzionamento di ogni LCD è basato sulle proprietà di particolari sostanze denominate cristalli liquidi. Tale liquido è intrappolato fra due superfici vetrose provviste di numerosi contatti elettrici con i quali poter applicare un campo elettrico al liquido contenuto. Ogni contatto elettrico comanda una piccola porzione del pannello identificabile come un pixel. Sulle facce esterne dei pannelli vetrosi sono collocati due filtri polarizzatori disposti su assi perpendicolari tra loro. La particolarità naturale dei cristalli liquidi è torcere di 90° la polarizzazione della luce che arriva da uno dei due filtri, permettendole quindi di attraversare il secondo polarizzatore. In assenza di campo elettrico, la luce può passare attraverso l'intera struttura e, trascurando la porzione di luce assorbita dai polarizzatori, l'apparecchio risulta trasparente. In presenza di un campo elettrico ortogonale ai piani delle due lastre di vetro, invece, le molecole del liquido si allineano parallelamente al campo, limitando la rotazione della luce entrante. Se i cristalli sono completamente allineati con il campo elettrico, la luce che vi passa attraverso è polarizzata perpendicolarmente al secondo polarizzatore e viene quindi bloccata del tutto facendo apparire il pixel non illuminato. Controllando la torsione dei cristalli liquidi in ogni pixel, tramite un terzo filtro, si è in grado di regolare quanta luce far passare. Si noti che in questo modo un pixel guasto apparirà sempre illuminato. Nella realtà alcune tipologie di pannelli funzionano all'opposto, ossia sono trasparenti se accesi ed opachi se spenti. In tal caso un pixel danneggiato risulta sempre opaco. Un LCD non emette luce: ecco il motivo per cui questi schermi necessitano della retroilluminazione (back light). La luce emessa da questo sistema attraversa il cristallo liquido e vi esce colorata dal sistema di filtraggio. L'attivazione di ogni singolo punto del display viene comandata da un meccanismo di indicizzazione "riga per colonna", per cui ogni pixel si attiva quando passa corrente in entrambi gli elettrodi (anteriore e posteriore) che lo riguardano. Nel display TFT (chiamati anche a matrice attiva mentre quelli descritti fino ad ora sono chiamati a matrice passiva) i singoli pixel sono attivati da un apposito transistor. Quindi non è più necessario porre davanti al video una serie di elettrodi: è sufficiente la presenza di un'unica lastra trasparente da impiegare come massa.

Principio di funzionamento di un cristallo liquido:



Principio di funzionamento di un cristallo liquido con applicata la tensione:

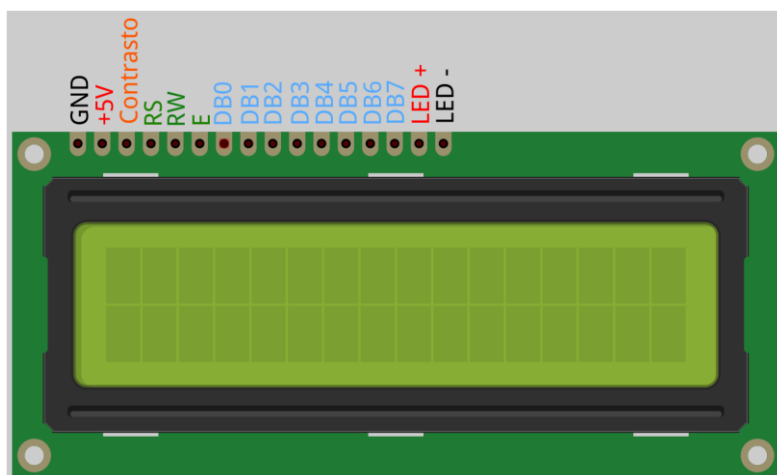


DIFFERENZA TRA MATRICE ATTIVA E PASSIVA

Un LCD può essere realizzato utilizzando una griglia di visualizzazione a matrice attiva (conosciuti come TFT) o passiva. Nella matrice attiva ad ogni intersezione di pixel è posizionato un transistor, di conseguenza richiede meno energia per controllare la luminosità dei pixel. Per questo motivo, la corrente nel display a matrice attiva può essere attivata e disattivata più spesso, migliorando così la frequenza di aggiornamento dello schermo (il mouse su uno schermo si muove velocemente grazie a questa possibilità di aggiornamento rapida). Un LCD a matrice passiva è formato da una griglia di conduttori con pixel posizionati ad ogni intersezione (esempio: calcolatrici, sveglie, orologi).

FUNZIONAMENTO E STRUTTURE FISICHE DI UN LCD 16X2

Nel nostro caso specifico avremo 2 righe nelle quali possiamo scrivere un massimo di 16 caratteri, la retroilluminazione utilizzata dal nostro LCD è con lo sfondo verde e le lettere nere.



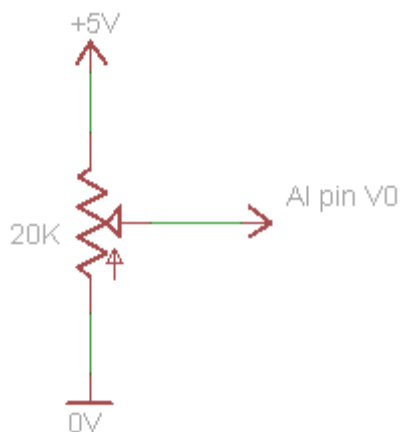
PIN

Pin	Denominazione	Descrizione
1	Vss o Gnd	Massa
2	Vcc	+5V
3	Vo	Regolazione contrasto
4	RS	Register Select (selezione dati/istruzioni)
5	R/W	Read/Write
6	E	Enable (Abilitazione)
7	D0	Linea Dati, bit 0
8	D1	Linea Dati, bit 1
9	D2	Linea Dati, bit 2
10	D3	Linea Dati, bit 3
11	D4	Linea Dati, bit 4
12	D5	Linea Dati, bit 5
13	D6	Linea Dati, bit 6
14	D7	Linea Dati, bit 7
15	A	Anodo Retroilluminazione (+4.2V)
16	K	Catodo Retroilluminazione (Massa)

1-GND

2-VCC 5V è importante non invertire la polarità perché il display si rovinerebbe in maniera irreparabile.

3-VO è il pin dedicato alla regolazione del contrasto, su di esso dovranno arrivare tensioni tra 0v e 5v in modo da regolare la luminosità dei vari pixel. Per regolare tale tensione si utilizzano dei trimmer da 10k collegati nel seguente modo:



4-RS register select. Se RS è a livello logico alto vuol dire che stiamo inviando dati che dobbiamo scrivere mentre se poniamo RS a livello logico basso significa che stiamo dando comandi da eseguire come per esempio la cancellazione dei caratteri presenti a video.

5-R/W read/write. Se posto a zero vogliamo scrivere (modalità scrittura) un dato mentre se posto ad uno vogliamo leggerlo (modalità lettura)

6-E enable. È il pin sul quale viene mandato il segnale di sincronismo, se poniamo E a livello logico alta significa che il microcontrollore esegue la lettura del dato sulla linea dati, lo interpreta e lo scrive a video. Perciò quando dovremo scrivere sul display prima predisponiamo la linea dati e successivamente poniamo E a livello logico alto. Una volta che il controller ha letto il dato E va riportato a livello logico zero.

7 ÷ 14 – D0 ÷ D7 linea dati. Qui vengono inviati i comandi da eseguire o i caratteri da scrivere. Per risparmiare pin sul microcontrollore si può utilizzare la modalità 4 bit cioè utilizzano solo 4 pin e inviamo i dati in due blocchi al posto che in un blocco unico.

Questi LCD hanno due tipi di memorie: la **CGROM**, che contiene i "font" dei caratteri, e la **DDRAM**, che ha il compito di memorizzare cosa va scritto sul display. Quando inviamo i dati sulla linea dati i caratteri saranno scritti uno dopo l'altro da un contatore sulla DDRAM. Ogni posizione dei caratteri sul display sono legate alla DDRAM quindi le posizioni in memoria delle due righe sono : -prima riga: 0x00 ÷ 0x0f -seconda riga: 0x40 ÷ 0x 4f. quindi per esempio nella locazione decimale 16 non produco nessun carattere quindi quando scrivo devo stare attento a dove posiziono i caratteri(questo lavoro solitamente è svolto in apposite librerie). La CGROM contiene i font presenti nel display che cambiano leggermente da display a display, i principali sono i seguenti:

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG ROM (1)			0	@	P	`	P				-	9	3	α	ρ	
xxxx0001	(2)		!	1	A	Q	a	q				。	ア	子	厶	ã	q
xxxx0010	(3)		"	2	B	R	b	r				「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s				」	ウ	フ	ε	ε	∞
xxxx0100	(5)		\$	4	D	T	d	t				、	エ	ト	フ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u				・	オ	ナ	1	ε	Ü
xxxx0110	(7)		&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w				ア	キ	ヌ	ラ	g	π
xxxx1000	(1)		<	8	H	X	h	x				イ	ク	ネ	リ	♪	×
xxxx1001	(2)		>	9	I	Y	i	y				ウ	ケ	ル	ル	”	γ
xxxx1010	(3)		*	=	J	Z	j	z				エ	コ	ン	レ	j	〒
xxxx1011	(4)		+	;	K	L	k	l				オ	サ	ヒ	ロ	*	π
xxxx1100	(5)		,	<	L	¥	l	l				ホ	シ	フ	フ	φ	円
xxxx1101	(6)		-	=	M	J	m	j				ユ	ズ	ハ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	^				ヨ	セ	ホ	ン	ん	
xxxx1111	(8)		/	?	O	_	o	_				ツ	ツ	マ	°	ö	■

Il funzionamento è semplice, per visualizzare per esempio la lettera A inviamo sulla linea dati 0100 0001 (si incrocia la colonna con la riga). I caratteri dal 32(00100000b) al 125(01111101) rispecchiamo il codice ascii. Quando scriveremo nel software ci basterà inviare "A" oppure usare il carattere binario o esadecimale corrispondente, in questo modo ci sarà molto facilitato scrivere delle stringhe sul display.

Il quadratino su un display per scrivere un carattere è di 5x8 pixel, nella tabella sono da 5x7 perché le ultime due sono utilizzate per il cursore presente sul display ma nulla ci vieta di utilizzarle complete quindi eliminare il cursore dal display. Abbiamo inoltre 8 locazioni vuote dove possiamo costruirci noi dei caratteri (prime otto locazioni in formato 5x8).

Il controller dispone di due registri a 8 bit: L'IR (**instruction register**) e DR (**data register**). Il primo è a sola scrittura e memorizzazione dei comandi e quindi le informazioni riguardanti l'indirizzamento della DDRAM e della CGROM. Il secondo è un registro di transito e memorizza le informazioni che dovranno essere poi scritte sulla DDRAM o sulla CGROM. Questi due registri collaborano con il pin RS in questo modo:

-RS=0 utilizzo IR (comando da eseguire)

-RS=1 utilizzo DR (carattere da visualizzare)

Supponendo di utilizzare il display in modalità lettura (cosa da noi non fatta) utilizzeremo i sei bit meno significativi per sapere la posizione attuale del cursore (AC – **address counter**) e quindi sapere il valore del contatore. Il bit più significativo (D7) sarà usato come enable(chiamato **busy flag**) cioè quando il controller non può leggere sarà posto a 1 mentre quando il controller potrà leggere verrà posto a 0.



COME INVIARE COMANDI AD UN LCD

Istruzione	RS	D7	D6	D5	D4	D3	D2	D1	D0	Descrizione
Pulisci display (Clear Display)	0	0	0	0	0	0	0	0	1	Cancella il contenuto del display e pone l'Address Counter a 0 (il cursore torna quindi in posizione 1 alla prima riga)
Home	0	0	0	0	0	0	0	1	...	Pone l'Address Counter a 0 (riporta il cursore ad inizio riga)
Modalità immissione (Entry Mode Set)	0	0	0	0	0	0	1	I/D	S	Imposta l'incremento della posizione nella DDRAM (I/D = Increment/Decrement) e lo spostamento del display (S = Shift display)
Controllo display (Display on/off control)	0	0	0	0	0	1	D	C	B	Spegne/accende il display (D = Display), visualizza/nasconde il cursore (C = Cursor), fa lampeggiare/rimanere fisso il cursore (B = Blinking)
Spostamento cursore o Display (Cursor or Display shift)	0	0	0	0	1	S/C	R/L	Imposta il movimento del cursore o del display (S/C = Shift display/Cursor) e lo spostamento del display (R/L = Right/Left) senza cambiare il contenuto della DDRAM
Impostazione funzioni (Function Set)	0	0	0	1	DL	N	F	Imposta la lunghezza dati dell'interfaccia (8 o 4 bit) (DL = Data Length), numero di righe del display (N = Number), grandezza del font da utilizzare (F = Font)
Indirizzo CGROM (Set CGROM Address)	0	0	1	*	*	*	*	*	*	Imposta il punto (l'indirizzo) della CGROM in cui scrivere o da cui leggere. (al posto degli asterischi va impostato l'indirizzo: es: inviamo 0100 0000 => setta l'indirizzo della CGROM al primo carattere). Il dato sarà disponibile (o verrà scritto) dopo questa impostazione. Questa funzione è utile per impostare i caratteri custom.
Indirizzo DDRAM (Set DDRAM Address)	0	1	*	*	*	*	*	*	*	Imposta il punto (l'indirizzo) della DDRAM in cui scrivere o da cui leggere. (al posto degli asterischi va impostato l'indirizzo: es: inviamo 1000 0000 => setta l'indirizzo della DDRAM al primo carattere, 1100 0000 => setta l'indirizzo della DDRAM al primo carattere della seconda riga). Il dato sarà disponibile (o verrà scritto) dopo questa impostazione

Dove c'è riportato "..." significa che avere 0 o 1 non cambia. Ora riportiamo il significato delle lettere riportate nella tabella precedente:

Simbolo del bit	se lo imposto a 0	se lo imposto a 1
I/D	La posizione all'interno della DDRAM decrementa	La posizione incrementa
S	Il testo rimane fermo	Il testo si sposta
D	Display spento	Display acceso
C	Cursore invisibile	Cursore visibile
B	Il cursore non lampeggia	Il cursore lampeggia
S/C	Si sposta il cursore	Si sposta il display
R/L	Spostamento a sinistra	Spostamento a destra
DL	Interfaccia a 4 bit	Interfaccia ad 8 bit
N	Il display ha una sola riga	Il display ha 2 o più righe
F	Font 5×8	Font 5×10

Dobbiamo ricordarci che quando mettiamo dei dati sulla linea dati il pin di E deve essere messo a 1 e poi a 0 così da fare la sincronizzazione.



LCD a retroilluminazione blu con pin disposti in parallelo.

3. FUNZIONAMENTO

1) SPIEGAZIONE HARDWARE

Il progetto può essere visto come la composizione di due ulteriori macro-blocchi che hanno reso realizzabile il corretto funzionamento dello stesso. E' suddiviso quindi in una parte hardware e una parte software

Hardware:

Quando si parla di hardware si intende la parte fisica di un determinato progetto o circuito, ovvero tutte quelle parti elettriche, elettroniche, magnetiche, ottiche e meccaniche che ne permettono il funzionamento.

Nel nostro caso con il termine hardware si intendono quindi sia i componenti elettronici utilizzati quali: circuiti integrati, resistori, condensatori, dispositivi optoelettronici, sensori, ma anche parti di tipo meccaniche come l'utilizzo di motori in corrente continua.

Per la realizzazione del progetto si è proceduto passo dopo passo, partendo da una base solida e continuando ad incrementare il circuito con delle idee innovative che potessero rendere la nostra veneziana reale, cioè applicabile fin da subito alla vita reale portando costi inferiori sia dal punto di vista costruttivo, ma anche nel risparmio energetico.

Software:

Quando si parla di software si intende la parte di logica e più nello specifico nel nostro caso della programmazione in linguaggio C++ ma non puro, dato che il microcontrollore che viene utilizzato, Arduino, utilizza un linguaggio di programmazione leggermente diverso dal C++ comune.

La veneziana realizzata può funzionare in due modalità: una manuale e una automatica. Tutto ciò è reso possibile grazie alla selezione di un deviatore che invia un segnale di tipo analogico al pin 10 di Arduino. Questo dato può assumere +5v, livello HIGH, o 0V, livello LOW. Fisicamente il deviatore è collegato sulla scheda millefori con i due terminali rispettivamente a +Vcc, 5V, GND e il pin centrale "invia" il segnale al microprocessore. Nel caso questo fosse posizionato in posizione LOW, il circuito funziona in modalità manuale mentre se fosse in posizione di HIGH lavora in modalità automatica.

MODALITA' LOW: MANUALE

Selezionando il deviatore in posizione LOW questo invia un segnale al pin 10 di Arduino, attraverso il centrale del componente stesso e viene rilevato dal microprocessore attraverso il comando di digitalRead (nella parte della programmazione si possono osservare il listato e i comandi utilizzati).

Dal punto di vista hardware i componenti elettronici utilizzati sono due pulsanti normalmente aperti, due finecorsa, resistori, diodi LED, un display LCD, un fotodiodo emettitore e uno rivelatore.

I due pulsanti normalmente aperti sono collegati attraverso una resistenza di pull down verso massa e con l'altro terminale a +Vcc. Tra la resistenza e il pin del pulsante viene prelevato un valore di tensione dal microprocessore e letto dallo stesso attraverso il comando di digitalRead.

Se non vi è nessuna azione di pressione, il pulsante essendo normalmente aperto, viene rilevato un valore di tensione LOW. Nel caso vi fosse un'azione di pressione sul pulsante, si chiuderebbe il circuito e verrebbe rilevato da Arduino un valore di +5V (HIGH). La stessa modalità viene utilizzata per i finecorsa che vengono posizionati sulla struttura in modo da determinare la massima apertura e la massima chiusura della veneziana, intervenendo direttamente sulla rotazione dei motori nel caso fossero premuti, in modo da non danneggiarla e da non rovinare i motori che continuerebbero a ruotare a vuoto, utilizzando potenza inutile e danneggiando la struttura.

I valori rilevati dal microprocessore dei due pulsanti e dai due finecorsa vengono confrontati e il funzionamento è regolato da una tabella di verità che viene poi tradotta in termini di linguaggio di programmazione attraverso il comando if.

In base ai dati ricevuti, Arduino determina che uscita erogare, se di livello alto o basso.

Questa istruzione viene inviata da due pedini del microprocessore al circuito integrato L293, rispettivamente ai pin 2 e 7 di questo ultimo: "ingessi" del ponte.

Il componente L293 essendo un ponte ad H consente ai motori, collegati con i rispettivi terminali a pin 3 e 6 dell'integrato cioè alle "uscite" del ponte ad H, di poter ruotare sia in senso orario che antiorario, determinando così il moto di salita e di discesa della veneziana. Il corretto funzionamento può essere visualizzato osservando l'accensione dei due diodi LED che sono dichiarati nel ciclo di setup del listato come periferiche di OUTPUT, ricevendo così in base al moto dei motori, un segnale alto o basso dal microprocessore: giallo salita e verde discesa, e sul display LCD dove apparirà la scritta dello stato della veneziana.

Nel progetto vi è una simulazione di un impianto di sicurezza composto da un fotodiodo emettitore, un fotodiodo rivelatore e da un buzzer. Il fotodiodo emettitore, polarizzato direttamente, è sempre attivo e eroga un raggio di luce infrarosso non visibile ad occhio nudo, ma attraverso la fotocamera di un cellulare. Questo raggio viene diretto ad un fotodiodo rivelatore polarizzato inversamente, collegato con il catodo a +Vcc e con l'anodo a massa attraverso una resistenza di carico da 180KΩ, in modo tale che possa condurre solo nel momento in cui il raggio infrarosso viene interrotto. Nel caso il fotodiodo ricevitore funzionasse, secondo la caratteristica di un diodo, come un "circuito aperto", non ci sarebbe una corrente che circola nella resistenza da 180KΩ e di conseguenza non vi è una caduta di potenziale sul componente passivo. Questo valore viene inviato al microprocessore, la veneziana continuerebbe nella sua azione e il buzzer rimane disattivato (il raggio infrarosso non è interrotto). Al contrario del caso precedente, se il raggio fosse interrotto dal passaggio di qualcosa o di qualcuno, il fotodiodo rivelatore si comporterebbe da "corto circuito", si genera quindi a questo punto una caduta di tensione sulla resistenza di carico, in quanto vi è iniziata a circolare una corrente inversa. Il valore di tensione sulla resistenza di carico viene inviato all'Arduino che arresterà immediatamente la salita o la discesa della veneziana e azionerà il buzzer e inizierà a suonare finché non viene rimosso l'ostacolo che interrompe il flusso del raggio infrarosso.

MODALITA' HIGH: AUTOMATICA

Questa modalità è diversa rispetto a quella presentata precedentemente in quanto la logica di funzionamento si basa sulla rivelazione della luce solare nell'ambiente e in base ad essa si determina il movimento di salita o di discesa della veneziana.

Tutto ciò è reso possibile grazie ad un sensore optoelettronico di tipo rivelatore, una fotoresistenza.

Questo componente è in grado di modificare il proprio valore resistivo in base al valore di luce solare che colpisce la parte fotosensibile del sensore.

Un reoforo della fotoresistenza è collegato a +Vcc, l'altro ad una resistenza collegata verso massa. Nel nodo tra i due componenti viene prelevato un valore di tensione che varia proporzionalmente al cambiamento del valore resistivo.

Quest'istruzione viene inviata al microprocessore Arduino ad uno degli ingressi analogici, nel nostro caso A0, che viene letto attraverso il comando analogRead, convertito in valori decimali dividendo il valore di tensione in fasce da 1024 bit in modo tale da poter lavorare molto più facilmente, determinando così le rotazioni di salita e discesa del motore e fissando il valore di tensione che corrisponde ad un determinato valore di illuminazione in cui avviene la commutazione o il cambio di rotazione.

Come nel caso precedente vi sono i due LED, uno giallo e uno verde, che aiutano nella visualizzazione dei due moti della veneziana cioè quello di salita e di discesa.

Nel progetto vi è anche un display LCD che indica, oltre ai tre stati della veneziana, il livello di illuminazione nell'ambiente.

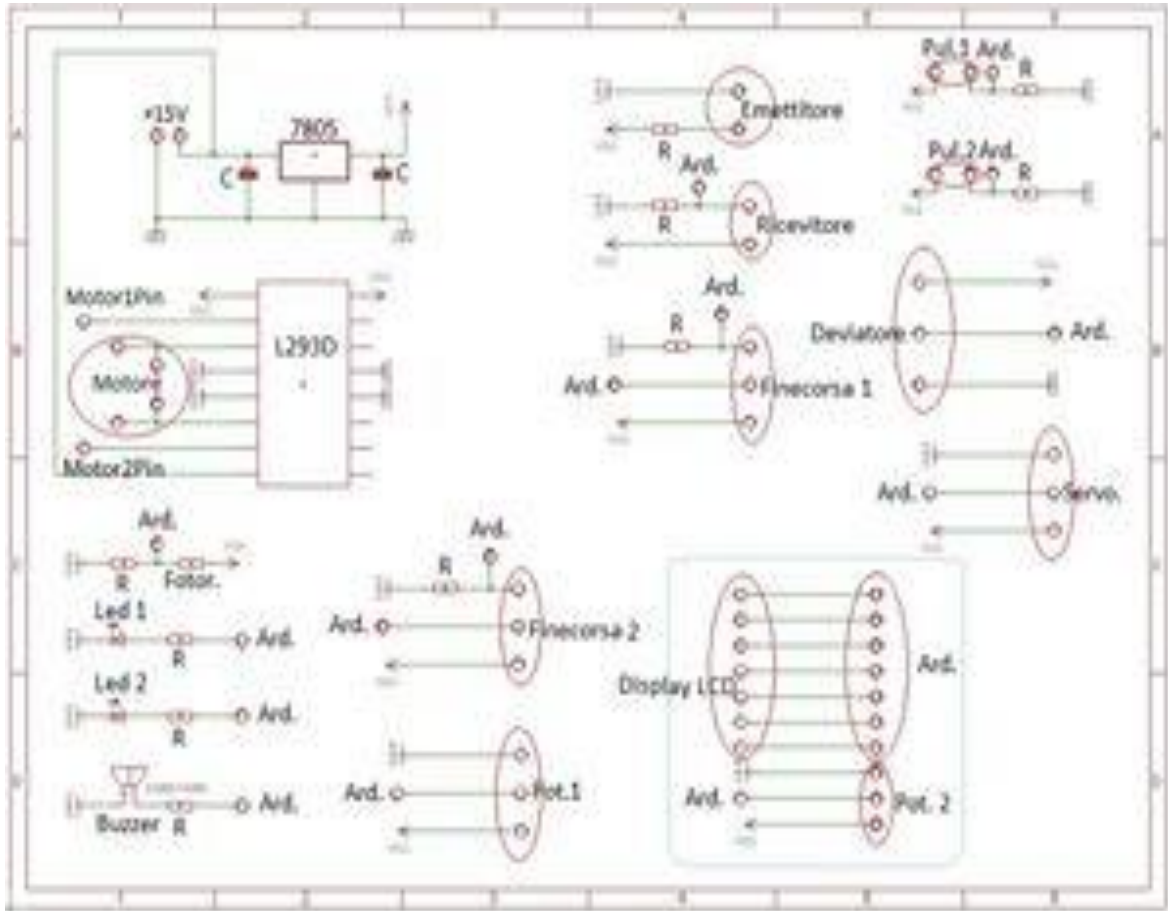
Questo è possibile grazie all'istruzione lcd.print la quale consente infatti di poter stampare qualsiasi dato sul display. Tutto questo deve essere però inizializzato sia dalla dichiarazione dei pin di Arduino collegati ai rispettivi ingressi del display e dal comando lcd.begin (16,2) che permette di poter riempire entrambe le righe del componente.

Qui si può vedere che tipo di collegamenti sono stati fatti nella parte hardware del progetto:

SCHEMA CONNETTORI (COLLEGAMENTI)

N°	Connettori verticali	N°	Connettori orizzontali	N°	Connettori display
1	Led verde (discesa)	1	N.A	1	RS
2	Led giallo (salita)	2	N.A	2	RW
3	Pulsante 1 (discesa)	3	N.A	3	E
4	Pulsante 2 (salita)	4	N.A	4	D4
5	Fine corsa discesa (Vcc)	5	N.A	5	D5
6	Fine corsa discesa (Gnd)	6	N.A	6	D6
7	Fine corsa discesa (arduino)	7	N.A	7	D7
8	Fine corsa salita (Vcc)	8	N.A	8	A
9	Fine corsa salita (Gnd)	9	N.A		
10	Fine corsa salita (arduino)	10	N.A		
11	Buzzer	11	N.A		
		12	Fotoresistenza		
		13	N.A		
		14	Display (Gnd)		
		15	Display (Vcc)		
		16	Deviatore		
		17	Ricevitore		
		18	N.A		
		19	N.A		
		20	N.A		
		21	N.A		
		22	Pin abilitazione motore 1		
		23	Pin abilitazione motore 2		
		24	Motore 1 pin 1		
		25	Motore 2 pin 2		
		26	Motore 1 pin 1		
		27	Motore 2 pin 2		

Qui di seguito possiamo osservare come è stata realizzata la parte hardware del progetto:



2) SPIEGAZIONE SOFTWARE

```
/*Si dichiarano le librerie presenti nel programma*/

#include <LiquidCrystal.h>//Libreria del display LCD
#include <Servo.h>//Libreria del servo motore

/*Si specificano i vari pin di Arduino*/

LiquidCrystal lcd(52,53,50,45,44,43,42);//Pin che corrispondono al display LCD
"LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)"
Servo servoMotor;//Servomotore
int valore;//Valore potenziometro del servomotore
int backLight = 41;//Retroilluminazione del display
int pulsante_d=12;//Pulsante discesa
int pulsante_s=30;//Pulsante salita
int fine_d=22;//Finecorsa discesa
int fine_s=28;//Finecorsa salita
int motor1Pin=5;//Abilitazione motore 1
int motor2Pin=9;//Abilitazione motore 2
int Led_s=7; //Led verde (salita)
int Led_d=8; //Led giallo (discesa)
int generale=10;//Deviatore (modalità automatica/manuale)
int fotoresistenza=A0;//Fotoresistenza
int fotodiodo=34;//Fotodiodo
int buzzer=13;//Buzzer

/*Si imposta la tipologia dei pin di Arduino (INPUT,OUTPUT/HIGH,LOW)*/

void setup() { //Stringa per l'impostazione
  pinMode(backLight, OUTPUT);//Retroilluminazione pin d'uscita
  digitalWrite(backLight, HIGH);//Retroilluminazione normalmente attiva
  pinMode(pulsante_s, INPUT);//Pulsante salita pin d'ingresso
  pinMode(pulsante_d, INPUT);//Pulsante discesa pin d'ingresso
  pinMode(fine_s, INPUT);//Finecorsa salita pin d'ingresso
  pinMode(fine_d, INPUT);//Finecorsa discesa pin d'ingresso
  pinMode(motor1Pin, OUTPUT);//Abilitazione motore 1 pin d'uscita
  pinMode(motor2Pin, OUTPUT);//Abilitazione motore 2 pin d'uscita
  pinMode(Led_s, OUTPUT);//Led salita pin d'uscita
  pinMode(Led_d, OUTPUT);//Led discesa pin d'uscita
  pinMode(generale, INPUT);//Deviatore pin d'ingresso
  pinMode(fotoresistenza, INPUT);//Fotoresistenza pin d'ingresso
  pinMode(fotodiodo, INPUT);//Fotodiodo pin d'ingresso
  Serial.begin(9600);//Si instaura il collegamento con la porta USB
  pinMode (buzzer, OUTPUT);//Buzzer pin d'uscita
  servoMotor.attach(13);//Si dichiara a quale pin corrisponde il servomotore
```

```

}

/*Da qui inizia il "vero" programma, ovvero cosa si vuole eseguire*/

void loop() { //Stringa per l'impostazione
  valore = analogRead(A1); //Valore potenziometro
  valore = map(valore, 0, 1023, 0, 180); //Si trasforma il valore in bit del potenziometro in
  gradi (mappaggio)
  servoMotor.write(valore); //Si fa assumere al potenziometro il valore mappato
  delay(15); //Ritardo minimo della rotazione (per questioni di sicurezza)

/* Si inseriscono i codici per far visualizzare in modalità manuale il valore di luce sul
displa LCD*/

  int val=analogRead(fotoresistenza); //Valore analogico fotoresistenza
  Serial.println(val, DEC); //Attraverso il monitor seriale si visualizza il valore della
fotoresistenza in decimale

  if (val<500) //Se il valore è maggiore di 500
  {
    lcd.begin(16,2); //Si imposta la grandezza del display
    lcd.setCursor(0,1); //Si imposta su quale riga si visualizzerà la frase scelta
    lcd.print("LUCE BASSA"); //Si visualizza "LUCE BASSA" sul display
  }
  else //Altrimenti (se il valore è minore di 500)
  {
    lcd.begin(16,2); //Si imposta la grandezza del display
    lcd.setCursor(0,1); //Si imposta su quale riga si visualizzerà la frase scelta
    lcd.print("LUCE BASSA"); //Si visualizza "LUCE ALTA" sul display
  }

  if(digitalRead(fotodiodo) == HIGH) { //Se il fotodiodo è alto, quindi conduce (non ci sono
ostacoli sul percorso della veneziana), allora...
    if(digitalRead(generale) == LOW) { //Se il deviatore è basso, quindi impostato verso
massa, allora...

/*In questa parte del programma si visualizza il funzionamento in automatico, il quale
dipende dai due finecorsa e dai due pulsanti (si ricorda che il fotodiodo è attivo)*/

/*Se tutto è inattivo, la veneziana è ferma e sul display si visualizza "Mantenimento"*/

    if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW) {
      digitalWrite(motor1Pin, LOW); //Si dichiara se l'abilitazione del motore 1 è attiva o
disattiva

```

```

    digitalWrite(motor2Pin, LOW);//Si dichiara se l'abilitazione del motore 2 è attiva o
disattiva
    digitalWrite(Led_s, LOW);//Si dichiara se il led di salita è acceso o spento
    digitalWrite(Led_d, LOW);//Si dichiara se il led di discesa è acceso o spento
    lcd.begin(16,2);//Si dichiara la grandezza del display LCD (lo spazio in cui si
collocano le scritte)
    lcd.clear();//Si dichiara che il display LCD è pronto a ricevere il segnale
    lcd.setCursor(0,0);// Si dichiara su quale riga del display si vuole visualizzare la frase
    lcd.print("MANTENIMENTO");//Stringa per scrivere sul display

}
else{

/*Se tutto è inattivo, tranne il finecorsa di discesa,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

    if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){//2
        digitalWrite(motor1Pin, LOW);
        digitalWrite(motor2Pin, LOW);
        digitalWrite(Led_s, LOW);
        digitalWrite(Led_d, LOW);
        lcd.begin(16,2);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("MANTENIMENTO");

    }
    else{

/*Se tutto è inattivo, tranne il finecorsa di salita,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

        if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){//3
            digitalWrite(motor1Pin, LOW);
            digitalWrite(motor2Pin, LOW);
            digitalWrite(Led_s, LOW);
            digitalWrite(Led_d, LOW);
            lcd.begin(16,2);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("MANTENIMENTO");

        }
        else{

```

```
/*Se tutto è inattivo, tranne i due finecorsa,la veneziana è ferma e sul display si visualizza "Mantenimento"*/
```

```
    if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == LOW && digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){//4
        digitalWrite(motor1Pin, LOW);
        digitalWrite(motor2Pin, LOW);
        digitalWrite(Led_s, LOW);
        digitalWrite(Led_d, LOW);
        lcd.begin(16,2);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("MANTENIMENTO");

    }
    else{
```

```
/*Se tutto è inattivo, tranne il pulsante di discesa,la veneziana scende e sul display si visualizza "Discesa"*/
```

```
    if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == HIGH && digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW){//5
        digitalWrite(motor1Pin, LOW);
        digitalWrite(motor2Pin, HIGH);
        digitalWrite(Led_s, LOW);
        digitalWrite(Led_d, HIGH);
        lcd.begin(16,2);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("DISCESA");

    }
    else{
```

```
/*Se tutto è inattivo, tranne il finecorsa e il pulsante di discesa,la veneziana è ferma e sul display si visualizza "Mantenimento"*/
```

```
    if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == HIGH && digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){//6
        digitalWrite(motor1Pin, LOW);
        digitalWrite(motor2Pin, LOW);
        digitalWrite(Led_s, LOW);
        digitalWrite(Led_d, LOW);
        lcd.begin(16,2);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("MANTENIMENTO");
```



```

    }
    else{

/*Se tutto è inattivo, tranne il finecorsa di salita ed il pulsante di discesa,la veneziana
scende e sul display si visualizza "Discesa"*/

        if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){//7
            digitalWrite(motor1Pin, LOW);
            digitalWrite(motor2Pin, HIGH);
            digitalWrite(Led_s, LOW);
            digitalWrite(Led_d, HIGH);
            lcd.begin(16,2);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("DISCESA");

        }
        else{

/*Se tutto è attivo, tranne il pulsante di salita, la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

            if(digitalRead(pulsante_s) == LOW && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){//8
                digitalWrite(motor1Pin, LOW);
                digitalWrite(motor2Pin, LOW);
                digitalWrite(Led_s, LOW);
                digitalWrite(Led_d, LOW);
                lcd.begin(16,2);
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("MANTENIMENTO");

            }
            else{

/*Se tutto è inattivo, tranne il pulsante di salita,la veneziana sale e sul display si
visualizza "Salita"*/

                if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW){//9
                    digitalWrite(motor1Pin, HIGH);
                    digitalWrite(motor2Pin, LOW);
                    digitalWrite(Led_s, HIGH);
                    digitalWrite(Led_d, LOW);
                    lcd.begin(16,2);
                    lcd.clear();

```

```

    lcd.setCursor(0,0);
    lcd.print("SALITA");

    }
    else{

/*Se tutto è inattivo, tranne il finecorsa di discesa e il pulsante di salita,la veneziana
sale e sul display si visualizza "Salita"*/

        if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){//10
            digitalWrite(motor1Pin, HIGH);
            digitalWrite(motor2Pin, LOW);
            digitalWrite(Led_s, HIGH);
            digitalWrite(Led_d, LOW);
            lcd.begin(16,2);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("SALITA");

        }
        else{

/*Se tutto è inattivo, tranne il finecorsa e il pulsante di discesa,la veneziana è ferma e
sul display si visualizza "Mantenimento"*/

            if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){//11
                digitalWrite(motor1Pin, LOW);
                digitalWrite(motor2Pin, LOW);
                digitalWrite(Led_s, LOW);
                digitalWrite(Led_d, LOW);
                lcd.begin(16,2);
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("MANTENIMENTO");

            }
            else{

/*Se tutto è attivo, tranne il pulsante di salita,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

                if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == LOW &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){//12
                    digitalWrite(motor1Pin, LOW);
                    digitalWrite(motor2Pin, LOW);
                    digitalWrite(Led_s, LOW);

```

```

    digitalWrite(Led_d, LOW);
    lcd.begin(16,2);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANTENIMENTO");

}
else{

/*Se tutto è inattivo, tranne i due pulsanti,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

    if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW){//13
        digitalWrite(motor1Pin, LOW);
        digitalWrite(motor2Pin, LOW);
        digitalWrite(Led_s, LOW);
        digitalWrite(Led_d, LOW);
        lcd.begin(16,2);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("MANTENIMENTO");

    }
    else{

/*Se tutto è attivo, tranne il finecorsa di salita,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

        if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){//14
            digitalWrite(motor1Pin, LOW);
            digitalWrite(motor2Pin, LOW);
            digitalWrite(Led_s, LOW);
            digitalWrite(Led_d, LOW);
            lcd.begin(16,2);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("MANTENIMENTO");

        }
        else{

/*Se tutto è attivo, tranne il finecorsa di discesa,la veneziana è ferma e sul display si
visualizza "Mantenimento"*/

            if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){//15

```

```

    digitalWrite(motor1Pin, LOW);
    digitalWrite(motor2Pin, LOW);
    digitalWrite(Led_s, LOW);
    digitalWrite(Led_d, LOW);
    lcd.begin(16,2);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANTENIMENTO");

    }
    else{

/*Se tutto è attivo, la veneziana è ferma e sul display si visualizza "Mantenimento"*/

        if(digitalRead(pulsante_s) == HIGH && digitalRead(pulsante_d) == HIGH &&
digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){//16
            digitalWrite(motor1Pin, LOW);
            digitalWrite(motor2Pin, LOW);
            digitalWrite(Led_s, LOW);
            digitalWrite(Led_d, LOW);
            lcd.begin(16,2);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("MANTENIMENTO");

            }}}}}}}}}}}}}}}}}}}
        }

/*A questo punto inizia il programma riguardante la modalità automatica*/

        else{//Altrimenti (se il deviatore è altro, quindi collegato a Vcc)
            int val=analogRead(fotoresistenza);
            Serial.println (val, DEC);

            if((val<500) && digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW){//sale
                digitalWrite(motor1Pin, HIGH);
                digitalWrite(motor2Pin, LOW);
                digitalWrite(Led_s, HIGH);
                digitalWrite(Led_d, LOW);
                lcd.begin(16,2);
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("SALITA");
                lcd.setCursor(0,1);
                lcd.print("LUCE BASSA");

            }
            else{

```

```

if((val<500) && digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){

digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, LOW);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, LOW);
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("MANTENIMENTO");
lcd.setCursor(0,1);
lcd.print("LUCE BASSA");
}
else{
if((val<500) && digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){

digitalWrite(motor1Pin, HIGH);
digitalWrite(motor2Pin, LOW);
digitalWrite(Led_s, HIGH);
digitalWrite(Led_d, LOW);
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SALITA");
lcd.setCursor(0,1);
lcd.print("LUCE BASSA");

}
else{
    if((val<500) && digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){

digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, LOW);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, LOW);
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("MANTENIMENTO");
lcd.setCursor(0,1);
lcd.print("LUCE BASSA");
}
else{
    if((val>500) && digitalRead(fine_s) == LOW && digitalRead(fine_d) == LOW) {
digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, HIGH);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, HIGH);

```

```

lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("DISCESA");
lcd.setCursor(0,1);
lcd.print("LUCE ALTA");

}
else{

if((val>500) && digitalRead(fine_s) == LOW && digitalRead(fine_d) == HIGH){

digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, LOW);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, LOW);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MANTENIMENTO");
  lcd.setCursor(0,1);
  lcd.print("LUCE ALTA");
}
else{
  if((val>500) && digitalRead(fine_s) == HIGH && digitalRead(fine_d) == LOW){
digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, HIGH);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, HIGH);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DISCESA");
  lcd.setCursor(0,1);
  lcd.print("LUCE ALTA");

}
else{
  if((val>500) && digitalRead(fine_s) == HIGH && digitalRead(fine_d) == HIGH){
digitalWrite(motor1Pin, LOW);
digitalWrite(motor2Pin, LOW);
digitalWrite(Led_s, LOW);
digitalWrite(Led_d, LOW);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MANTENIMENTO");
  lcd.setCursor(0,1);

```

```
lcd.print("LUCE ALTA");  
}
```

```
} } } } } }
```

/*Questa parte del programma riprende il fotodiode, i codici seguenti sono nel caso in cui al ricevitore non arriva la radiazione luminosa, quindi si ha un ostacolo durante il moto della veneziana-*/

```
else {
```

```
digitalWrite(motor1Pin, LOW);  
digitalWrite(motor2Pin, LOW);  
digitalWrite(Led_s, LOW);  
digitalWrite(Led_d, LOW);
```

```
lcd.begin(16,2);
```

```
lcd.clear();
```

```
lcd.setCursor(0,0);
```

```
lcd.print("MANTENIMENTO");
```

```
tone(buzzer,4400,500);//Il buzzer suona. Con la funzione "tone" si dichiara: il pin  
scelto, la frequenza, ed il ritardo con cui il suono viene trasmesso
```

```
delay (1500);//Ritardo impostato
```

```
}
```

```
}
```

