*Bottoni Diego Simone Mazzotta 5D*

PROGETTO: Ascensore

*Indice:*

*1. L’automazione pag.1*

*2. Controlli automatici pag.2*

*3. Introduzione al progetto pag.3*

*4. Schema a blocchi pag.3*

*5. Osservazioni sul sistema pag.5*

*6. Specifiche di progetto pag.5*

*7. Analisi della realizzazione pag.10*

*8. Manuale per l’utilizzo pag.11*

*Allegato:*

*-Software Arduino*

*1. L’automazione*

**Che cos’è l’Automazione**

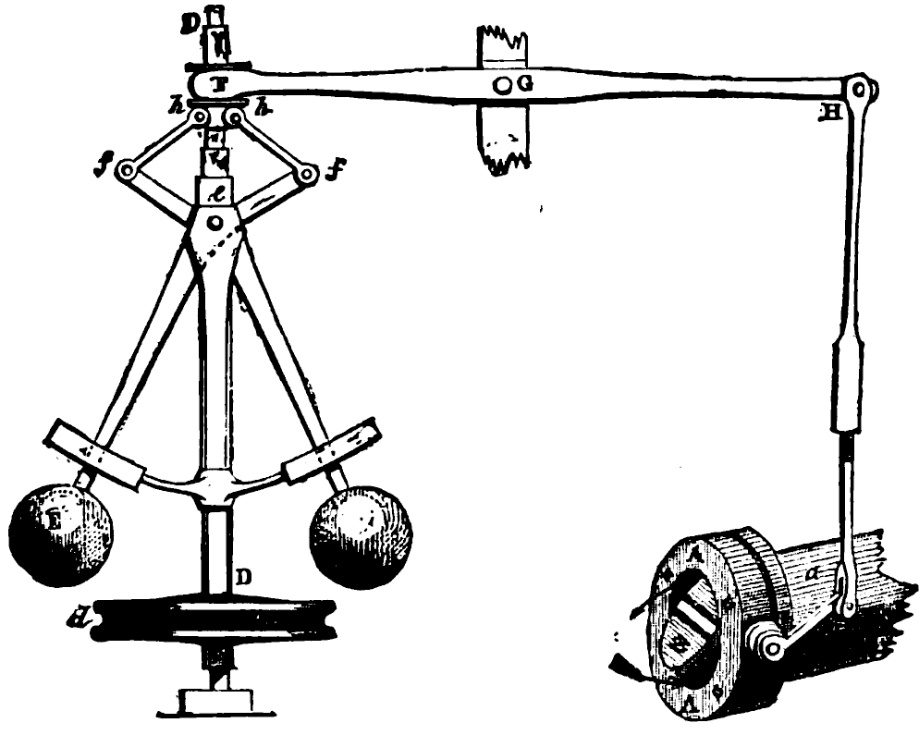
Il termine automazione nasce per identificare tutto ciò che è necessario per far funzionare una macchina (o un processo) in modo automatico, ossia senza l’intervento dell’uomo. L’automazione industriale, in particolare, sfrutta tecnologie meccaniche, elettroniche ed informatiche per il controllo dei processi produttivi industriali, governando flussi di energia, di materiali e di informazioni.

Dal punto di vista storico, l’automazione è nata con il principale scopo di sostituire l’uomo in compiti ripetitivi o nocivi, con apparecchiature in grado di operare in modo autonomo o con minimi interventi da parte dell’operatore umano.

Fondendo tecnologie industriali proprie dei processi di produzione e tecnologie informatiche, si propone di consentire la gestione efficiente delle informazioni, ponendosi come branca dell’ingegneria moderna che ha per obiettivo quello di ridurre o eliminare l’intervento dell’uomo nella produzione di beni e servizi.

**Il primo sistema di automazione**

Precursore dei sistemi di automazione può essere considerato il **regolatore di velocità di J. Watt** (fine del Settecento) per le locomotive a vapore, il cui scopo iniziale era di mantenerne la velocità costante, indipendentemente dal peso trainato o dalle pendenze della strada ferrata. Il regolatore, basandosi sulla velocità reale e confrontandola in modo meccanico con quella prestabilita, riusciva ad ottenere la potenza necessaria per variare la velocità.



Il funzionamento del regolatore di Watt si basa sugli effetti della forza centrifuga sulle masse E.  
Se la velocità della macchina aumenta esse si allargano e, per mezzo di leve, fanno chiudere un po' la valvola a farfalla. La quantità di vapore che giunge nel cilindro diminuisce e la macchina rallenta. Se la macchina ritarda succede esattamente il contrario.

*2. Controlli automatici*

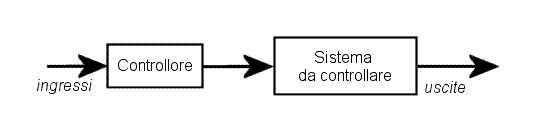
La teoria dei controlli è quella branca della [scienza](https://it.wikipedia.org/wiki/Scienza) ed [ingegneria](https://it.wikipedia.org/wiki/Ingegneria) che studia il comportamento di un [sistema](https://it.wikipedia.org/wiki/Sistema_%28fisica%29) le cui grandezze siano soggette a variazioni nel [tempo](https://it.wikipedia.org/wiki/Tempo). Questa scienza, che ha un vastissimo campo di applicazione, è nata nell'ambito dell'[elettronica](https://it.wikipedia.org/wiki/Elettronica) [industriale](https://it.wikipedia.org/wiki/Industria) e dell'automazione.

Il controllo può avvenire solo in un regime temporale. Spesso lo studio matematico con [modelli matematici](https://it.wikipedia.org/wiki/Modello_matematico) nel [dominio](https://it.wikipedia.org/wiki/Dominio_%28matematica%29) del tempo diventa molto difficile, causa la necessità di risolvere [equazioni differenziali](https://it.wikipedia.org/wiki/Equazione_differenziale). Quindi attraverso delle trasformazioni, le trasformate di cui le più famose sono quelle di [Fourier](https://it.wikipedia.org/wiki/Trasformata_di_Fourier) e quelle di [Laplace](https://it.wikipedia.org/wiki/Trasformata_di_Laplace), si studia lo stesso sistema con tecniche di tipo [algebrico](https://it.wikipedia.org/wiki/Algebra) nel dominio della frequenza e una volta ottenuto il risultato si antitrasforma per tornare nel dominio del tempo.

Un sistema automatico di controllo può funzionare essenzialmente in due modi: come *controllo ad anello aperto* o come *controllo in retroazione*.

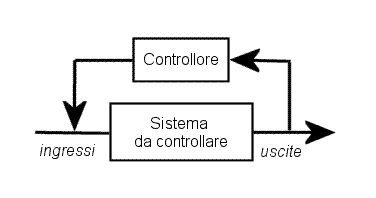
Il **controllo ad anello aperto** (o **in avanti** o **predittivo** o ***feedforward***) si basa su una elaborazione degli ingressi eseguita senza conoscere il valore dell'uscita del sistema controllato, essendo note alcune proprietà del sistema da controllare.

In questo caso è fondamentale avere un buon modello matematico che descriva con buona precisione il comportamento del sistema. Tanto più il modello matematico su cui si basa l'azione del controllo feedforward è esatto, tanto più questo tipo di controllo è affidabile.



Il **controllo ad anello chiuso** (o **retroazionato** o **all'indietro** o ***feedback***), più complesso ma molto più flessibile del primo, può rendere [stabile](https://it.wikipedia.org/wiki/Stabilit%C3%A0_%28teoria_dei_sistemi%29) un sistema che di per sé non lo è affatto.

In questo caso l'anello di controllo riporta all'ingresso del processo che si vuole controllare o rendere stabile una funzione dell'uscita che va sommata algebricamente al segnale già presente in ingresso.



*3. Introduzione al progetto*

Il nostro obiettivo era quello di realizzare un prototipo funzionante che possa svolgere le funzioni di un reale ascensore. Attraverso un sistema di controllo della posizione, il prototipo deve controllare la cabina sull’asse verticale, muoverla rispettando le chiamate interne (dalla cabina stessa) e esterne (dai piani).

Il sistema è dotato inoltre di un pulsante di ALT (che possa interrompere la salita o la discesa della cabina, di un allarme antincendio (che interrompa le normali funzioni dell’ascensore, portandolo al piano terra e inibendo tutte le chiamate) e di attuatori luminosi (Diodi LED e Display a 7 segmenti) che permettono all’operatore di visualizzare la posizione e lo stato della cabina.

Questo sistema di controllo della posizione, viene interfacciato con il microcontrollore che, a sua volta, è interfacciato al motore tramite una scheda di potenza.

*4. Schema a blocchi*

****

**Analisi dei singoli blocchi**

-Microcontrollore: il microcontrollore da noi utilizzato è Arduino Mega. Esso controlla il funzionamento dell’intero sistema seguendo le istruzioni del programma che è stato caricato al suo interno. Possiamo dare istruzioni al microcontrollore tramite il computer a cui è interfacciato e, utilizzando la porta seriale, possiamo comunicare in tempo reale con il programma al suo interno.

-Sensore di posizione: al fine di controllare in tempo reale la posizione della cabina è stato predisposto un potenziometro il cui rotativo è stato accoppiato al motore attraverso una puleggia che, trasferisce il moto della corda (che sostiene la cabina) al perno del potenziometro; così facendo una variazione della posizione della cabina si trasforma in una variazione della resistenza ai capi del potenziometro.

-Pulsanti di chiamata: inviano al microcontrollore la richiesta di inviare la cabina al piano di destinazione scelto.

-Allarme incendio: quando è attivo costringe la cabina ad andare e restare al piano terra, declinando ogni segnalazione di chiamata e tenendo accesi tutti i led rossi.

-Pulsante di ALT: quando premuto, il microcontrollore riceve l’istruzione di arrestare il motore, fermando la cabina nella posizione in cui si trova.

-Scheda di potenza: i segnali di uscita dei pin di Arduino non hanno potenza sufficiente per pilotare direttamente il motore, in quanto esso ha bisogno di molta più corrente che il microcontrollore può fornire. La scheda di potenza risolve questo problema, in quanto essendo collegata con il generatore (alimentazione) può tranquillamente pilotare il motore della cabina, prendendo tutta l’energia necessaria da esso. Nonostante ciò essa è sempre pilotata da Arduino, quindi è come se fosse il microcontrollore stesso a pilotare direttamente l’attuatore.

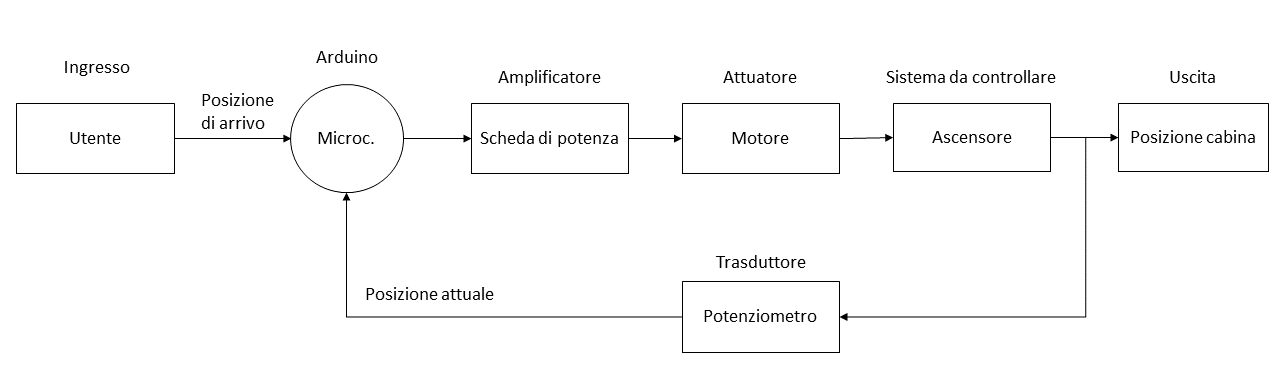
-Motore: viene utilizzato per trasformare l’energia elettrica in energia meccanica. Essendo un motore in continua può ruotare in entrambi i versi (orario e antiorario) a seconda della polarità con cui lo si alimenta. In questo modo può pilotare la cabina in salita e in discesa.

-Led RGB: vengono accesi in caso di attivazione dell’allarme antincendio o del pulsante ALT

-Led Piani: sono presenti 2 led per piano: uno verde, che rappresenta la disponibilità della cabina al piano e uno rosso, il quale segnala che l’ascensore è occupato

-Display 7 segmenti: visualizzano in tempo reale il piano più vicino al quale si trova la cabina

*5. Osservazioni sul sistema*

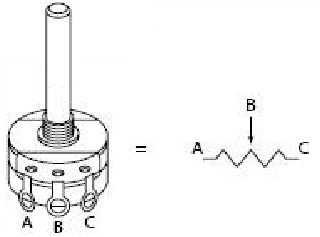


Per quanto riguarda il ramo che comanda l’attuatore, possiamo considerare questo schema equivalente che, utilizzando il microcontrollore come nodo sommatore della controreazione, controlla sulla catena diretta il movimento del motore e sulla catena di controreazione il controllo della posizione della cabina. In questo modo il sistema si autogestisce durante la fase di scorrimento della cabina tra un piano ed un altro.

*6. Specifiche di progetto:*

**Il controllo di posizione**

Abbiamo utilizzato un potenziometro multi-giri con una resistenza variabile tra 0 e 10KΩ. Il componente è fornito di tre terminali, ai capi dei due esterni è presente una resistenza fissa pari al valore specificato (10 KΩ), mentre tra il centrale e uno qualsiasi degli altri due è presente una resistenza variabile proporzionale alla posizione del perno ruotante.

[](http://www.google.it/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjX16TPltnUAhXEshQKHTZ9AUUQjRwIBw&url=http%3A%2F%2Farduinotutorial.altervista.org%2F2011%2F03%2F&psig=AFQjCNFDT8Op0-FYaRlDXXzIQ0DD0xgrog&ust=1498486297873039)

Come si nota in figura, fra i terminali A e C vi è la resistenza fissa, mentre tra A e B e tra B e C, si può prelevare la resistenza variabile.

Da tutto ciò risulta la formula Rab + Rbc = Rac = 10KΩ.



Applicando una tensione continua ai capi dei terminali A e C (Vac), si otterranno due tensioni Vab e Vbc. Per il secondo principio di Kirchoff (la somma algebrica delle tensioni in una maglia è nulla) possiamo scrivere: Vab + Vbc = Vac.

Considerando il potenziometro come un partitore di tensione la quale presa ai capi del terminale centrale e uno dei due esterni, si avrà un valore di tensione proporzionale alla posizione del rotativo. Accoppiando il moto del motore al moto del perno riusciamo a far variare la posizione dello stesso per avere una variazione di tensione precisa che corrisponde all’esatta variazione di posizione.

In questo modo possiamo affermare che: ogni variazione di posizione compiuta dal motore, ripercuotendosi sul valore della resistenza, quindi su quello di tensione in uscita, può essere acquisito dal nostro Arduino.

**Conversione analogico-digitale del valore del potenziometro**

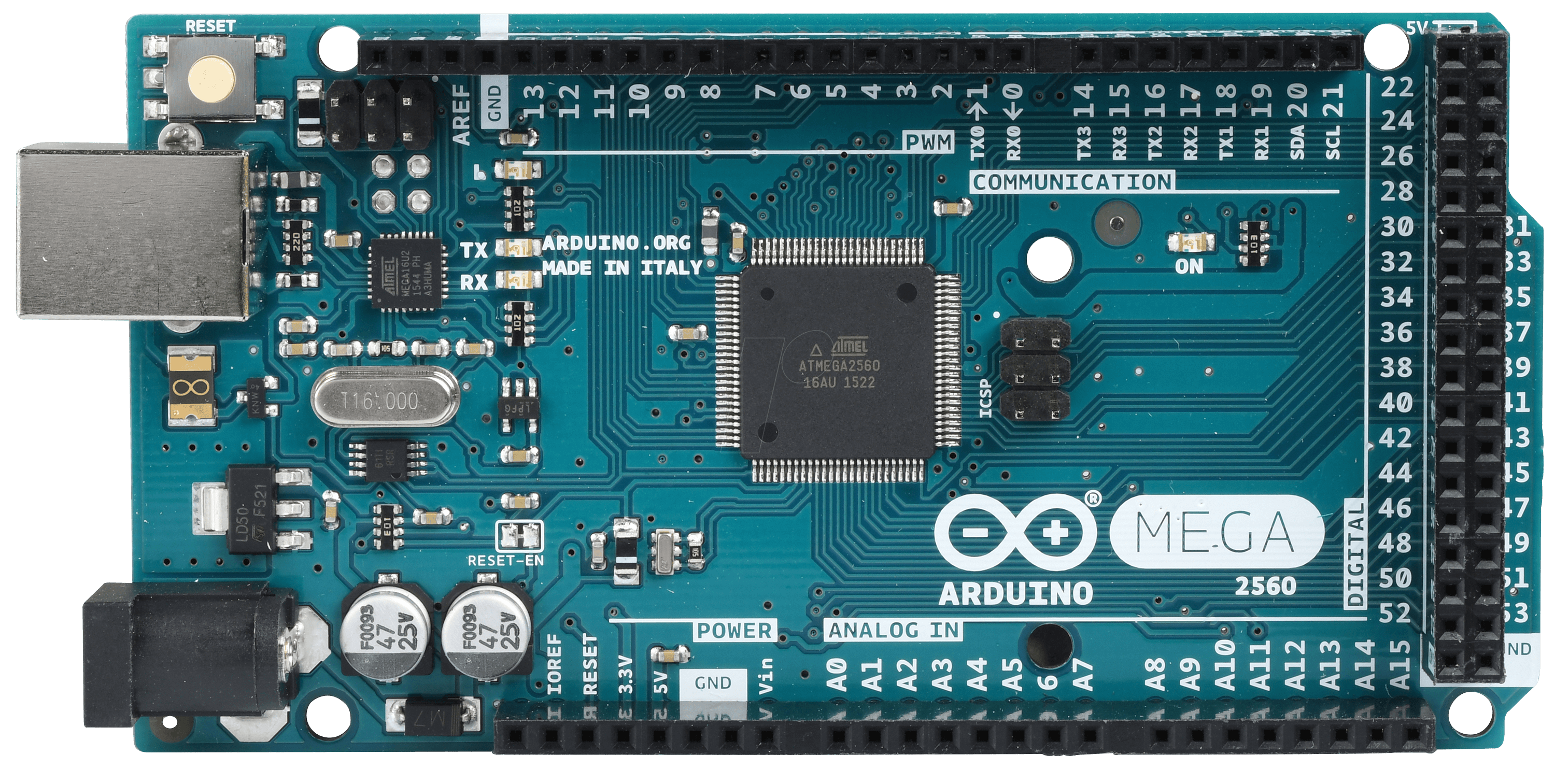
Per essere elaborato dal microcontrollore, il segnale di uscita dal potenziometro, ha bisogno di essere convertito in digitale. Il microcontrollore che abbiamo impiegato per questo progetto è Arduino Mega AT2560, il quale ha già al suo interno un ADC a 10 bit di risoluzione. Questo convertitore trasforma il segnale analogico che riceve in ingresso (in questo caso un segnale in tensione) in un opportuno segnale digitale.

Il range di valori che questo ADC può convertire è compreso tra 0 e 5V, per questo il circuito su cui si basa il controllo della posizione rispetta questo range; così facendo ogni possibile valore di tensione fornito dal potenziometro può essere convertito in digitale e acquisito da Arduino.

Come abbiamo già accennato la risoluzione di questo dispositivo è 10 bit, ciò significa che il valore digitale verrà convertito su un bus di 10 bit; possiamo calcolare il numero di combinazioni possibili elevando 2 al numero di bit di risoluzione in questo caso avremo 210 = 1024 combinazioni.

Dovendo suddividere un segnale con Vmax = 5V su 1024 combinazioni avremo un LSB (bit meno significativo) di circa 5mV, il quale è anche il valore della sensibilità.

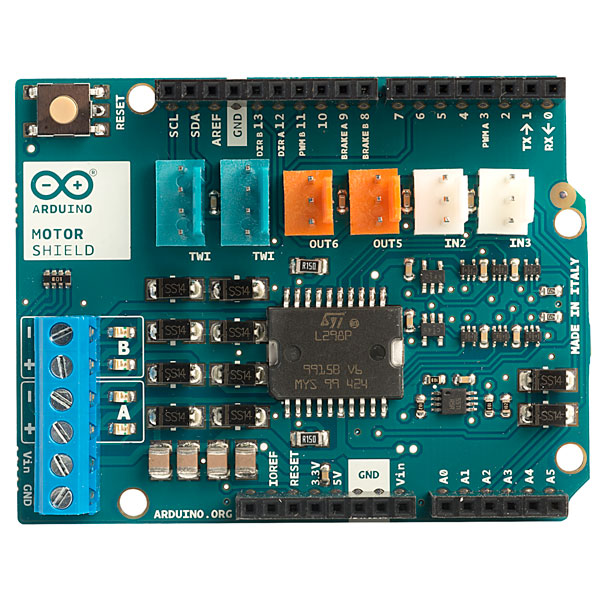
**Il microcontrollore**

[](https://www.google.it/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwi2paK3kNnUAhVLkRQKHeewBQkQjRwIBw&url=https%3A%2F%2Fwww.reichelt.com%2FSingle-board-microcontroller%2FARDUINO-MEGA%2F3%2Findex.html%3FACTION%3D3%26GROUPID%3D6667%26ARTICLE%3D119696&psig=AFQjCNE8BElRWmOW2rdxQvo_tp-LlEcoqQ&ust=1498484638394344)

Arduino è una piattaforma hardware low-cost programmabile, con cui è possibile creare circuiti "quasi" di ogni tipo per molte applicazioni, soprattutto in ambito di robotica ed automazione. Si basa su un Microcontrollore della ATMEL, l'ATMega168/328. C'è da aggiungere un particolare molto importante, ogni programma che si scrive su Arduino sarà naturalmente avviato a loop() finché non si toglie l'alimentazione dal dispositivo.

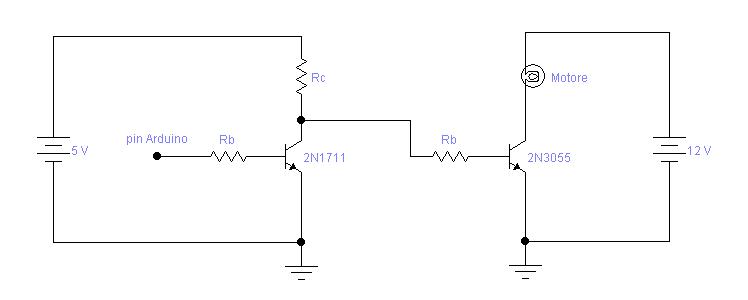
Questo framework è studiato per artisti, designers, hobbiesti e chiunque sia interessato a creare oggetti interattivi. La scheda Arduino è in grado di interagire con l'ambiente in cui si trova ricevendo informazioni da una grande varietà di sensori. Ma non si parla solo di sensori, Arduino può comandare luci, LED, motori e altri attuatori. Il linguaggio di programmazione è basato su Wiring (un ambiente di programmazione Open-Source pensato per una facile applicazione per semplificare la programmazione in C e C++) e sull'interfaccia Processing. Sono inoltre disponibili commercialmente molte schede applicative [plug-in](https://it.wikipedia.org/wiki/Plugin_(informatica)), note come "shields".

**La scheda di potenza**

[](https://www.google.it/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwj7k5qhkNnUAhVJOxQKHSBZDiMQjRwIBw&url=https%3A%2F%2Fwww.rapidonline.com%2Farduino-a000079-motor-shield-rev3-73-4455&psig=AFQjCNEndl992xXuIuo7QN_JZDUVuhBUGw&ust=1498484555784904)

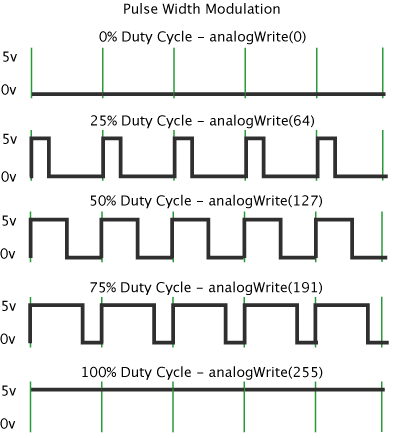
Questo motor shield per Arduino è basato sull' L298: un completo ponte H doppio progettato per il pilotaggio di carichi induttivi come relè, solenoidi e motori. Lo shield permette di controllare 2 motori tramite Arduino in velocità e direzione, in maniera indipendente l'uno dall'altro. E' anche possibile misurare l'assorbimento in corrente di ogni motore. Il motor shield deve essere alimentato necessariamente con una tensione esterna.

In alternativa alla motor shield di Arduino, sarebbe possibile utilizzare una scheda di potenza a BJT, con un primo transistor che, riceve i segnali dai pin del microcontrollore e che, a sua volta, pilota un secondo transistor di potenza collegato al motore. In questo modo anche un pin di Arduino che non può erogare più di 25-40mA, può pilotare un motore che necessita di 1° per funzionare. L’unico inconveniente di questo tipo di scheda di potenza è che non sarebbe possibile effettuare l’inversione della polarità con altrettanta facilità rispetto a quella che abbiamo con la motor shield.



**Il controllo PWM**

Il funzionamento di un motore in continua (DC) può essere gestito in due modi: a regime statico oppure a regime dinamico. La differenza sta nel controllo di velocità che si effettua; nel primo caso, la tensione di alimentazione è costante, nel secondo, essa è variabile nel tempo. Possiamo facilmente ed efficacemente controllare la velocità di un motore con un circuito chiamato PWM (pulse width modulation).

[](http://www.google.it/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiPjMygqdnUAhVFnRQKHc0jCwwQjRwIBw&url=http%3A%2F%2Fwww.mcmajan.com%2Fmcmajanwpr%2Fblog%2F2016%2F09%2F17%2Ftrasformare-luscita-pwm-in-una-tensione-analogica-i-filtri-rc-e-i-timers-it%2F&psig=AFQjCNGlEQ-CQyl1w26N2dbQrkEKp1aFEA&ust=1498491292793168)

Come si vede dall’immagine, se alimentassimo il motore con un segnale a frequenza e tensione fisse, ma facendone variare la durata del periodo positivi (duty cycle), avremmo una conseguente variazione della velocità. La frequenza, inoltre, non può essere troppo bassa in quanto si rischierebbe di accendere e spegnere il motore invece di farne variare l’andatura. Utilizzando questo sistema, abbiamo un modo funzionale per pilotare qualsiasi motore in continua con un preciso controllo di velocità.

*7. Analisi della realizzazione:*

Siamo partiti dall’idea di controllare un sistema con una catena diretta e un feedback: quale miglior esempio se non il controllo di un ascensore? In questo caso dobbiamo tener conto di molti fattori, tra cui gli input ricevuti dall’esterno per la selezione del piano di destinazione, la necessità di un motore che permetta il movimento della cabina, il controllo continuo che deve essere effettuato sulla posizione della cabina e la visualizzazione della stessa attraverso attuatori luminosi.

Traendo spunto da un modello già esistente, ma non funzionante, abbiamo disegnato una prima bozza del nuovo prototipo, cercando di eliminare i difetti da quello precedente, ottimizzandone le prestazioni utilizzando il minor spazio possibile.

Una volta realizzato lo chassis, più compatto e funzionale, abbiamo scelto i componenti più adatti in base alle nostre esigenze. Riguardo la scelta dell’attuatore di movimento, abbiamo optato per un motore in continua rispetto a un passo-passo; in quanto, il fatto di doversi muovere verso l’alto e verso il basso con frequenti cambi di verso e, di essere interfacciato con la logica di controllo e la relativa scheda di potenza, lo hanno reso il più viabile fra tutti. In particolare abbiamo scelto un motore di un alzacristalli di un’auto e, dopo averne apportato alcune modifiche al fine di adattarlo al funzionamento del prototipo, lo abbiamo fissato al telaio.

I primi problemi che abbiamo riscontrato hanno riguardato il trasferimento del moto del motore al rotativo del potenziometro; questo collegamento è alla base di questo progetto, in quanto tutto si basa sul controllo in feedback della posizione della cabina tramite la variazione di resistenza. Con un circuito alimentato a 5V (direttamente da Arduino) con unico utilizzatore lo stesso potenziometro, possiamo trasformare la variazione di resistenza in una variazione di tensione. A tal proposito, abbiamo pensato di inserire una puleggia tra il motore e il potenziometro che, bloccata insieme al rotativo del potenziometro, lo faccia ruotare seguendo l’andamento della corda alla quale è attaccata la cabina.

Oltre alla messa a punto della parte hardware è stata sviluppata anche la parte software, ovvero tutta la parte di programmazione che deve essere caricata sul microcontrollore. Non avendo ancora pronta la parte meccanica sul nostro prototipo (puleggia e motore), abbiamo potuto testare il programma mano a mano che veniva sviluppato. Essendo 2 prototipi analoghi per funzionamento e struttura (sistema di acquisizione della posizione), lo stesso software è compatibile con entrambi, cambiano solamente i valori di tensione (quindi di posizione) dei piani; problema ovviabile inserendo delle opportune variabili che possono essere modificate anche per effettuare la calibrazione della cabina ai piani.

Dopo aver ultimato tutti i collegamenti tra i singoli componenti e il microcontrollore, siamo passati alla fase di controllo: abbiamo caricato il firmware finale su Arduino e abbiamo verificato che il nostro prototipo riproducesse fedelmente le funzioni più comuni degli attuali ascensori.

Cercando nuove idee da sviluppare per migliorare questo progetto e traendo spunto dai modelli reali più moderni, abbiamo pensato di sviluppare un nuovo sistema di gestione delle chiamate ai piani, eliminando la pulsantiera all’interno della cabina, scegliendo direttamente la destinazione dall’esterno, tramite dei tablet che andrebbero a sostituire gli ormai obsoleti pulsanti. Non c’è limite alle possibili implementazioni da apportare. Basta la fantasia.

*8. Manuale per l’utilizzo:*

Alimentare il prototipo collegando Arduino e la sua scheda di potenza al generatore di tensione (12V - scheda di potenza / 5V - Arduino). Il sistema rileverà subito la posizione della cabina e la mostrerà grazie ai display. Attraverso la pressione di un qualsiasi pulsante, il sistema risponderà al comando accendendo i led rossi (cabina occupata) e inviando la cabina al piano di destinazione. Una volta raggiunto il piano desiderato si spegneranno i led rossi e si accenderà il led verde (presente al piano) corrispondente. Durante la salita o la discesa, è possibile monitorare costantemente la posizione della cabina attraverso i display, i quali vengono aggiornati dal software ad ogni cambio di piano.

Mentre la cabina si muove, è possibile premere il pulsante di ALT per fermarla e poter così scegliere una nuova destinazione. Per quanto riguarda il test dell’allarme antincendio, azionando il relativo deviatore verrà lanciata la sequenza di allarme: verranno accesi tutti i led rossi e sui display verrà mostrato il relativo simbolo di pericolo, la cabina sarà inviata al piano terra e, fino al rientro dell’allarme non sarà possibile utilizzare l’ascensore in alcun modo.

Come ulteriore funzionalità, è possibile interfacciare Arduino con un PC per visualizzare il valore della posizione, attraverso valori decimali su una scala da 0 a 1023 che, è il risultato della conversione analogico digitale del valore di tensione sul potenziometro, compiuta dall’ADC interno di Arduino (1024 valori in quanto abbiamo 10 bit di risoluzione) e comandare direttamente il sistema di movimento scegliendo il piano di destinazione direttamente da tastiera.