MAGAZZINO AUTOMATICO

RELAZIONE DI AVANZAMENTO LAVORI

Tallino Giorgia 5AA a.s. 2018-2019

## INTRODUZIONE

Il progetto che stiamo realizzando è un magazzino automatico, che chiede all’utente determinati dati e agisce di conseguenza sull’immagazzinamento di alcuni pacchetti.

Il magazzino prevede 9 celle di immagazzinamento, sono quindi presenti 3 colonne e 3 righe, e una cella iniziale, la quale è l’unica a cui teoricamente l’utente può accedere.

Sono previste due modalità dii funzionamento:

-**modalità manuale:** l’utente fornisce le coordinate di una cella del magazzino che sono composte dal numero della colonna e della riga, se nella cella indicata è presente un pacchetto esso verrà prelevato dal magazzino e dato all’utente, posizionandolo nella cella iniziale, invece se la cella indicata è vuota significa che l’utente ha posizionato un pacchetto nella cella iniziale, esso verrà prelevato e depositato nella cella indicata;

-**modalità automatica:** i pacchetti vengono identificati ognuno da un numero da 1 a 9, l’utente deve indicare il numero del pacchetto a cui è interessato, se il numero è già presente, il pacchetto viene prelevato e depositato nella cella iniziale, se il numero non è ancora presente nel magazzino, significa che il pacchetto che l’utente ha posizionato nella cella iniziale verrà identificato dal numero inserito ed esso sarà prelevato e posizionato nella cella vuota più vicina.

La modalità automatica viene utilizzata per ottimizzare lo spazio nel magazzino, in quanto i pacchetti vengono posizionati sempre nella prima cella libera che si trova, quindi il più vicino possibile alla cella iniziale. Inoltre questa modalità è pensata considerando un magazzino reale in cui l’utente non deve sapere la posizione nel magazzino di un pacchetto, infatti gli è sufficiente conoscere il suo numero identificativo; nel caso il magazzino venisse spento e riacceso, l’utente potrebbe ancora prelevare il proprio pacchetto conoscendone solo il numero, questo perché ad ogni spostamento che avviene all’interno magazzino viene salvata nella EEPROM dell’Arduino la nuova disposizione dei pacchetti.

## Schema a blocchi

Lo schema a blocchi comprende:

-il blocco centrale di elaborazione, che è costituito dall’Arduino Mega che agisce come centralina del magazzino;

-dal blocco centrale partono due blocchi di gestione dei motori passo-passo;

-il blocco di condizionamento, per l’attuatore lineare, che lavora a 12V ma viene controllato attraverso l’Arduino che fornisce uscite a 5V.

-il blocco di output per l’utente, che è costituito dal display, utilizzato per mostrare la situazione interna del magazzino e chiedere informazioni all’utente;

-il blocco di input dall’utente, composto dal tastierino che permette all’utente di utilizzare il magazzino;

-il blocco di controllo del magazzino, che prevede un sensore di distanza, per rilevare la presenza nelle celle dei pacchetti, e 4 finecorsa per rilevare se il pezzo che muove i pacchetti è arrivato alla fine dei binari e bloccare i motori nel caso avvenisse.

## Schema elettrico

Lo schema elettrico comprende:

-i pin dell’Arduino Mega, che verrà inserito direttamente sul circuito stampato, i collegamenti ad esso quindi non saranno effettuati mediante cavi;

-i due ponti ad H per il controllo dei motori passo-passo e 2 morsettiere a 2 vie per ciascun motore;

-sono presenti 8 led per verificare le uscite dei ponti ad H (potrebbero essere rimossi nel progetto finale);

-i pin per collegare il display e il tastierino numerico alla scheda;

-una morsettiera a tre vie per collegare l’alimentazione (12V, 5V e GND), che viene fornita da un alimentatore per PC;

-sensore di distanza, con comunicazione I2C;

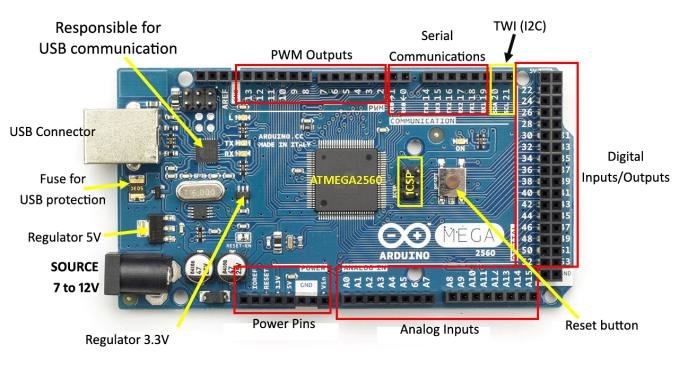
-2 pin per gestire l’attuatore;

-4 coppie di pin per collegare i finecorsa presenti nel magazzino, che sono costituiti da dei pulsanti collegati da una parte a massa dall’altra a degli input dell’Arduino collegati ai pull-up interni della scheda;

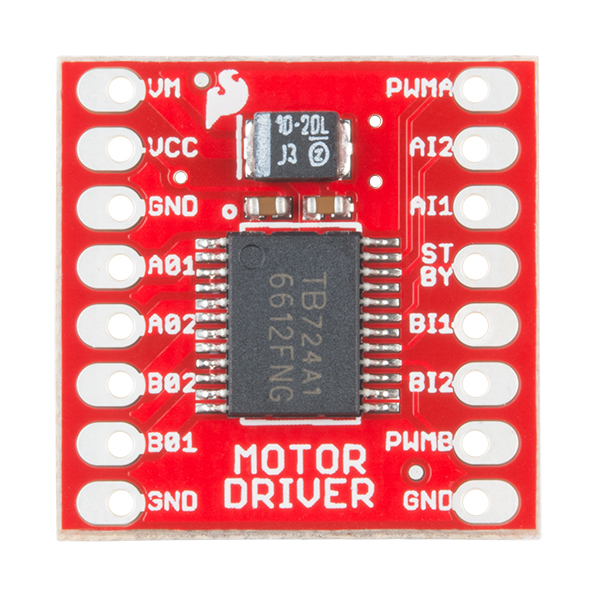
## componenti utilizzati

### Arduino MEGA 2560

È stato scelto questo microcontrollore per l’elevato numero di pin input/output che possiede.



### TB6612 TNG

Sono dei ponti ad H, sono stati scelti come sostituti degli L298, in quanto essi hanno un’efficienza maggiore: L298 eff.=70% TB6612 eff.=90%

Inoltre, a differenza degli L298, questi sono dei ponti ad H MOSFET.

Non necessitano quindi di un dissipatore.

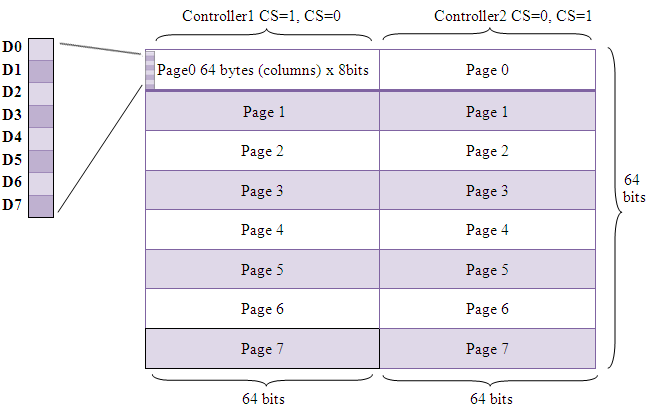
### Attuatore lineare

Attuatore lineare a 12V, con velocità 1cm/s, è provvisto di due finecorsa interni che fermano il motore in sicurezza alle due estremità della sua corsa.

### Display 128x64

|  |  |  |
| --- | --- | --- |
| Pin no. | **Function** | **Name** |
| 1 | Ground (0 V) | Vss |
| 2 | Supply voltage; 5V | Vcc |
| 3 | Contrast adjustment | Vo |
| 4 | High to display data; Low for instruction code | Register select (RS) |
| 5 | Low to write to the register; High to read from the register | Read/Write (R/W) |
| 6 | Reads data when high; Writes data at high to low transition (falling edge) | Enable (EN) |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Chip selection for IC1; Active high | CS1 |
| 16 | Chip selection for IC2; Active high | CS2 |
| 17 | Reset signal; Active low | RST |
| 18 | Output voltage for LCD driving | Vout |
| 19 | Backlight VCC (5V) | LED A |
| 20 | Backlight Ground (0V) | LED K |





Il display è costituito in totale da 1024 bit (64x128), che sono suddivisi in due metà che vengono selezionate mediante i pin CS, in ogni metà sono presenti 8 pagine ciascuna divisa in 64 colonne e 8 righe, come mostra la figura qui di fianco.

## software

Nel programma del nostro progetto abbiamo deciso di utilizzare il meno possibile librerie create da altri, quindi per riuscire a gestire il tutto abbiamo dovuto “spezzare” il programma principale in più parti utilizzando delle classi. Quelle attualmente complete, che sono state provate con i componenti e che funzionano sono la classe per gestire la rotazione dei motori passo-passo e quella per sapere che tasto viene premuto sul tastierino.

È in fase di sviluppo la classe per gestire i vari movimenti che il magazzino deve compiere per lo spostamento dei pacchi, per esempio lo spostarsi dalla cella iniziale ad una data cella, oppure l’azione di prelevare o depositare un pacchetto, ecc. Per poter completare questa parte del software è però necessaria la struttura completa del magazzino, la quale è ancora in lavorazione, soprattutto per mancanza di pezzi che sono stati ordinati e devono essere ancora consegnati.

Sarà prevista una classe anche per verificare la presenza nella cella di un pacchetto mediante l’uso di un sensore di distanza.

Per la gestione del display 64x128 è necessaria una libreria esterna, attualmente sono state utilizzate le librerie openGLCD e u8g2 per fare delle prove.

Infine il programma principale che raggruppa tutte le parti del codice è stato pensato e strutturato mediante il seguente flow chart: