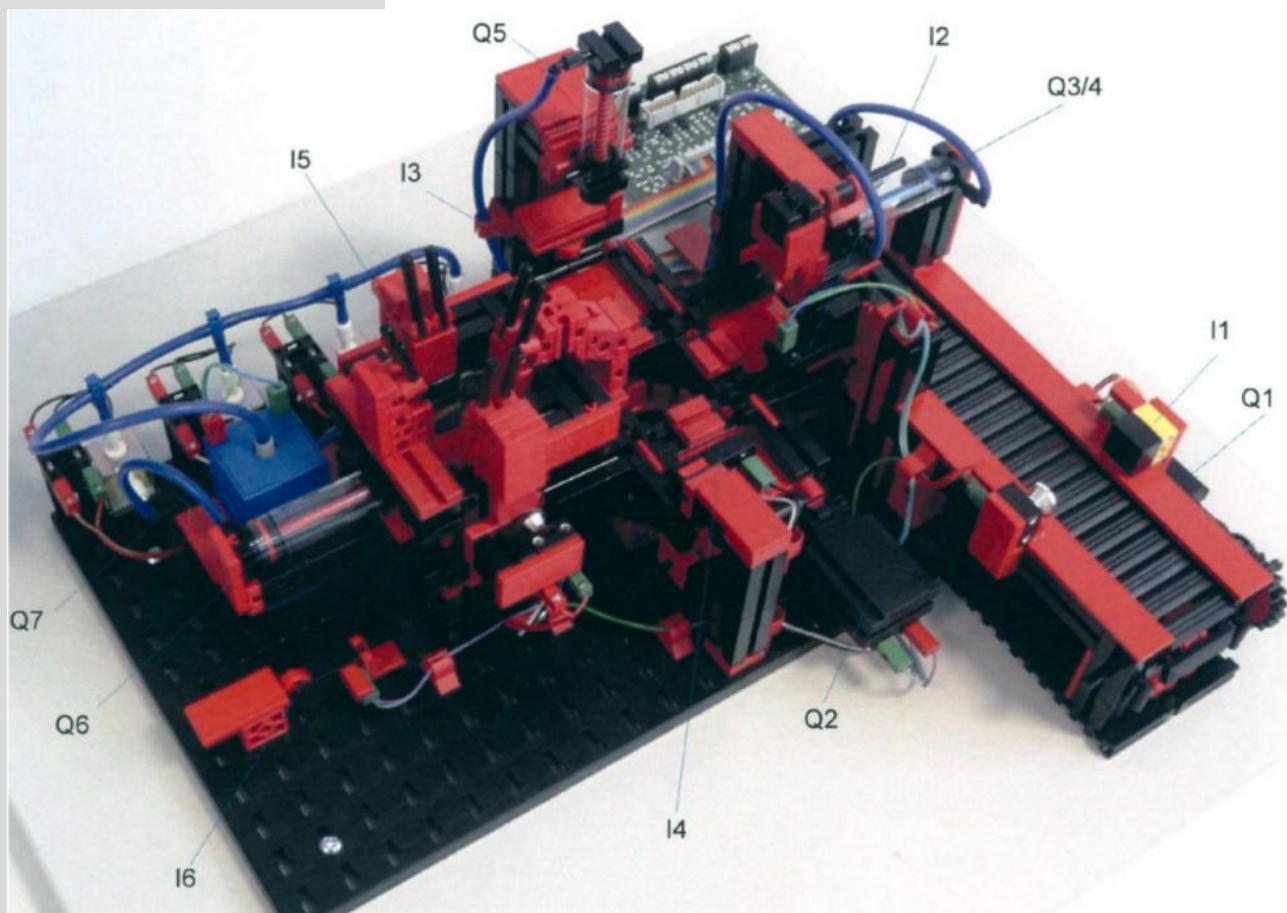


2022

Stazione Pneumatica



Nicola Laghi, Lucas Prati

4^A EN

09/04/2022

Stazione Pneumatica

Introduzione:

Nel seguente sistema automatico si vuole automatizzare la stazione pneumatica con etichettatrice mostrata in copertina. L'hardware è dotato di quattro ingressi di cui due finecorsa (attivi bassi) e due fotocellule (attive alte). All'inserimento di un pezzo, viene eccitata la prima fotocellula che avvia il processo di lavorazione. L'oggetto viene quindi trasportato da vari cilindri pneumatici, una piattaforma rotante e un nastro trasportatore fino a una stazione di etichettatura. Infine, il pezzo raggiunge l'ultima fotocellula, che arrestando il nastro trasportatore, segna la conclusione della lavorazione del pezzo. Questo progetto è stato realizzato tramite una scheda Arduino Mega e un programma in C.

Si segnalano di seguito altre caratteristiche del sistema:

- L'impianto è alimentato a 24V e lo si vuole controllare con una scheda Arduino Mega, con il microcontrollore (MCU) ATmega2560.
- La scheda Arduino Mega è alimentata a 5V tramite cavo USB collegato al PC, ma può essere alimentata anche con alimentatore esterno 9V/12V collegato al jack 2,1mm della scheda.
- Per far comunicare la scheda con l'impianto sono presenti opportuni circuiti di interfaccia che hanno lo scopo di far passare le tensioni da 24V a 5V e viceversa e, nel caso degli attuatori elettrici e pneumatici, di fornire anche la corrente richiesta per la loro attuazione.
- Il sistema è inoltre dotato di un sensore di temperatura analogico e di un display LCD per il monitoraggio della temperatura, del tempo di lavoro e del numero di pezzi lavorati.

Descrizione degli I/O dell'impianto:

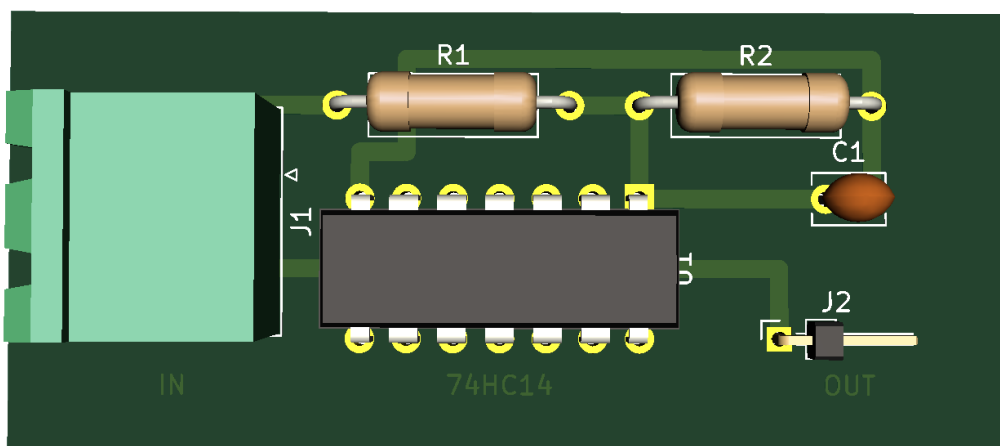
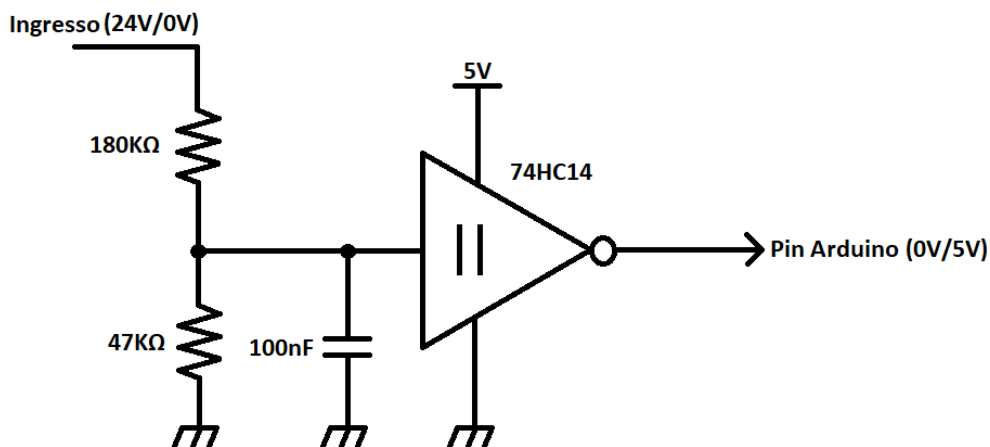
Ingressi	Pin Arduino Mega	Riferimento simbolico	
I1	29	FUsc	Fotocellula di uscita dal nastro (attiva alto)
I4	35	Allineamento	Allineamento della piattaforma rotante (attivo basso)
I5	37	FIng	Fotocellula di ingresso (attiva alta)
I6	39	PStop	Accensione/Spegnimento (attivo bassa)

Uscite	Pin Arduino Mega	Riferimento simbolico	
Q1	6	Nastro	Motore nastro trasportatore
Q2	2	Rotazione	Motore per ruotare la piattaforma
Q3	36	Spin2_Av	Cilindro avanti per inserire il pezzo nel nastro
Q4	40	Spin2_Ind	Cilindro indietro
Q5	32	Etichettatrice	Etichettatrice
Q6	4	Spin1	Cilindro di ingresso nella stazione rotante
Q7	44	Compressore	Compressore

NB: Tutte le uscite della centralina di controllo (Arduino Mega) sono attive basse, perché il circuito driver a relè utilizzato è di tipo invertente (il relè viene eccitato quando l'uscita della centralina vale 0). Il primo cilindro e quello dell'etichettatrice sono a singolo effetto con molla di richiamo, il cilindro che inserisce il pezzo sul nastro è invece a doppio effetto (si usano Q3 e Q4).

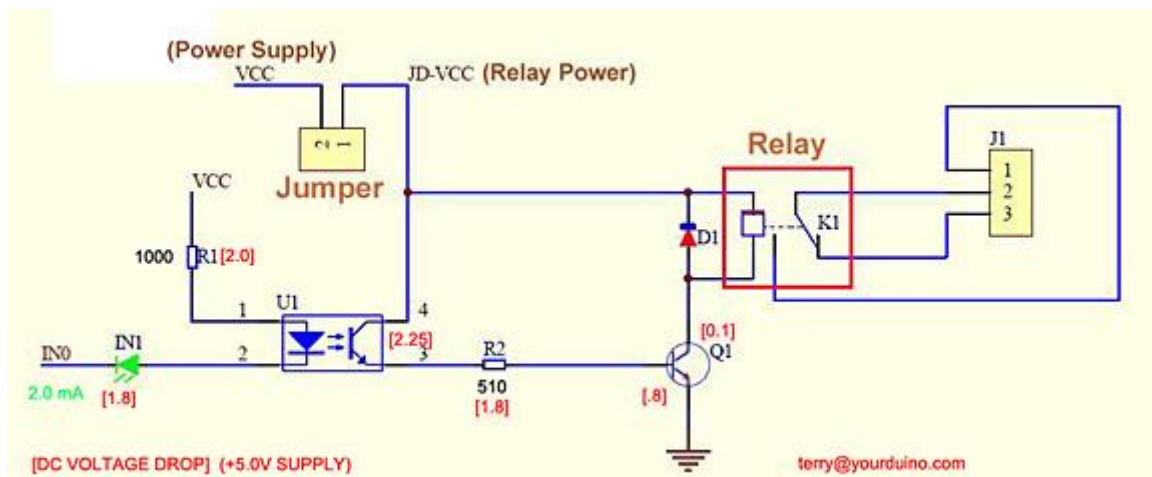
- **Scheda Arduino Mega (centralina di controllo):** Arduino Mega 2560 è una scheda microcontrollore basata sull'ATmega2560. Il microcontrollore a 8 bit dispone di 54 pin di ingresso/uscita digitali (di cui 15 utilizzabili come uscite PWM), 16 ingressi analogici e 4 porte seriali; un oscillatore al quarzo da 16 MHz, una connessione USB, un jack di alimentazione, e un pulsante di reset per riavviarlo. Per l'alimentazione è sufficiente collegare la scheda a un computer con un cavo USB o alimentarla con un adattatore AC-DC o una batteria.

- Impianto pneumatico:** L'impianto è composto da 2 fotocellule, un finecorsa, un pulsante, due cilindri pneumatici con molla di richiamo, un cilindro a doppio effetto e 2 motori elettrici, uno per la piattaforma rotante e uno per il nastro trasportatore. I cilindri sono azionati da elettrovalvole. L'impianto è alimentato a 24V tramite un alimentatore da banco regolabile.
- Circuiti di condizionamento degli ingressi:** La centralina alimentata a 5V deve leggere gli ingressi digitali dell'impianto (stato fotocellule e fine corsa) che hanno livelli di tensione 0/24V. Per non danneggiare la scheda si è realizzato un semplice circuito di condizionamento basato su un partitore resistivo in grado di fornire 5V quando in ingresso ci sono 24V, cui è stato aggiunto un condensatore per filtrare i rimbalzi meccanici associati ai fine corsa. Il segnale filtrato che varia lentamente è inviato all'integrato 74HC14, che non è altro che una porta logica NOT con doppia soglia di scatto (con isteresi). La porta logica NOT è quella che inverte la logica degli ingressi: gli ingressi che per l'impianto sono attivi bassi per la scheda diventano attivi alti, e viceversa. In figura è mostrato lo schema elettrico del circuito di condizionamento:



- **Circuiti driver delle uscite:** Le uscite della centralina a 5V non sono in grado di azionare le elettrovalvole e i motori elettrici dell'impianto per 2 motivi:
 - 5V non sono sufficienti, ne servono 24
 - La corrente massima che una singola uscita della centralina può fornire è pari a 40mA, mentre per azionare valvole e motori ne servono 200 circa.

Come circuiti driver per pilotare le uscite si è utilizzata una scheda elettronica a relè commerciale già disponibile, ogni canale della scheda ha lo schema elettrico sottostante:



VCC e JD-VCC sono collegate all'alimentazione 5V della scheda Arduino, GND è collegato a GND della scheda Arduino.

I tre terminali di ogni relè sono così collegati:

- 1) 24V dell'alimentatore da banco.
- 2) Uscita Q dell'impianto.
- 3) NC, Not Connected.

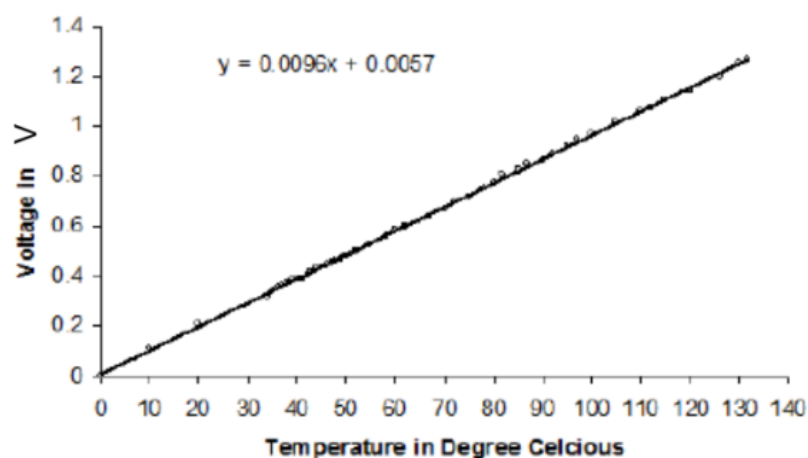
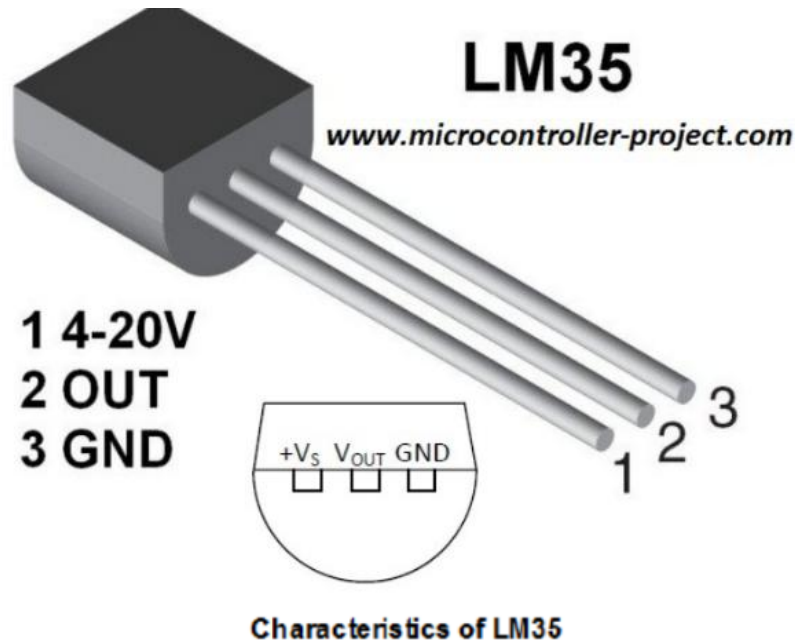
Quando il relè non è eccitato, il pin 2 è collegato al pin 3, ossia l'uscita Q dell'impianto è flottante/NC: l'elettrovalvola o il motore corrispondenti ad essa non sono attivati. Quando il relè è eccitato, il pin 2 è collegato al pin 1, ossia l'uscita Q è collegata a 24V: l'elettrovalvola o il motore corrispondenti ad essa non sono attivati.

L'ingresso IN0 della scheda a relè viene collegato al pin di uscita della scheda Arduino corrispondente all'uscita dell'impianto che si vuole pilotare.

Dallo schema si nota che il relè è eccitato se l'uscita della scheda Arduino vale 0V. Infatti, il relè si eccita se il transistor Q1 è acceso. Ma Q1 si accende se il diodo del fotoaccoppiatore è acceso. Il diodo del fotoaccoppiatore si accende solo se l'uscita della scheda Arduino è 0V, mentre è spento se l'uscita della scheda Arduino è 5V. Questo è il motivo per cui tutte le uscite della scheda Arduino sono attive basse.

- **Sensore di temperatura:**

Il sensore di temperatura utilizzato è LM35, è un sensore lineare a semiconduttore, presenta una sensibilità di $10\text{mV}/^{\circ}\text{C}$, ossia fornisce una tensione di uscita che varia di 10mV ogni variazione di 1°C della temperatura ambiente



Collegando il pin 2 del sensore a un ingresso analogico della scheda Arduino (ad esempio A0), possiamo convertire il valore della tensione fornita dal sensore in un numero compreso tra 0 e 1023.

La funzione `analogRead(A0)` fornisce un valore numerico compreso tra 0 e 1023 ($2^{10}-1$) proporzionale alla tensione analogica fornita all'ingresso A0 della scheda; Nel caso in cui all'ingresso A0 fosse collegato il pin 2 del LM35 avremmo:

- 0 se in ingresso ci sono 0V (ossia 0°C).
- $1 \cdot 1023 / 5 = 205$ se in ingresso ci sono 1V (ossia 100°C).

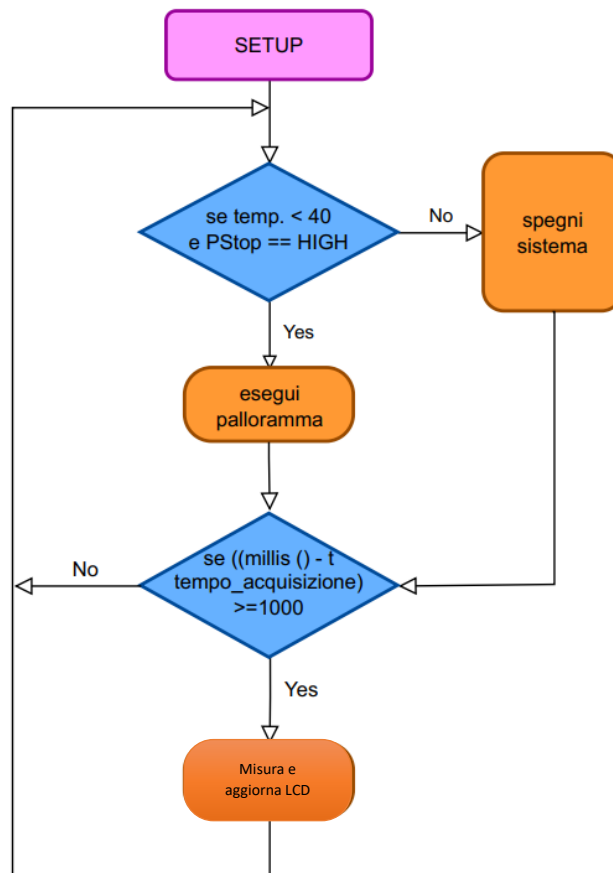
Per convertire i valori del sensore in dati utili per il programma si assegna alla variabile: “`analogRead(A0)*0.488`”, così facendo si otterrà il numero di gradi ambientali. L'alimentazione del sensore è a 5V DC.

- **Display LCD 16x2 I2C:**

Dato che la visualizzazione dei dati su monitor seriale richiede che la scheda Arduino sia collegata al PC tramite cavo USB quando ciò non è possibile abbiamo utilizzato un display LCD 16x2 I2C alimentato a 5V che servirà per indicare il numero di pezzi lavorati, la temperatura ambientale letta dal sensore e il tempo di lavoro. Utilizzando un display LCD seriale con interfaccia I2C sono sufficienti 2 pin di una scheda Arduino per pilotare il display, insieme alla libreria `LyquidCrystal I2C` (l'indirizzo del display, individuato con un apposito codice dato, è 0x3F). I comandi utilizzati sono i seguenti:

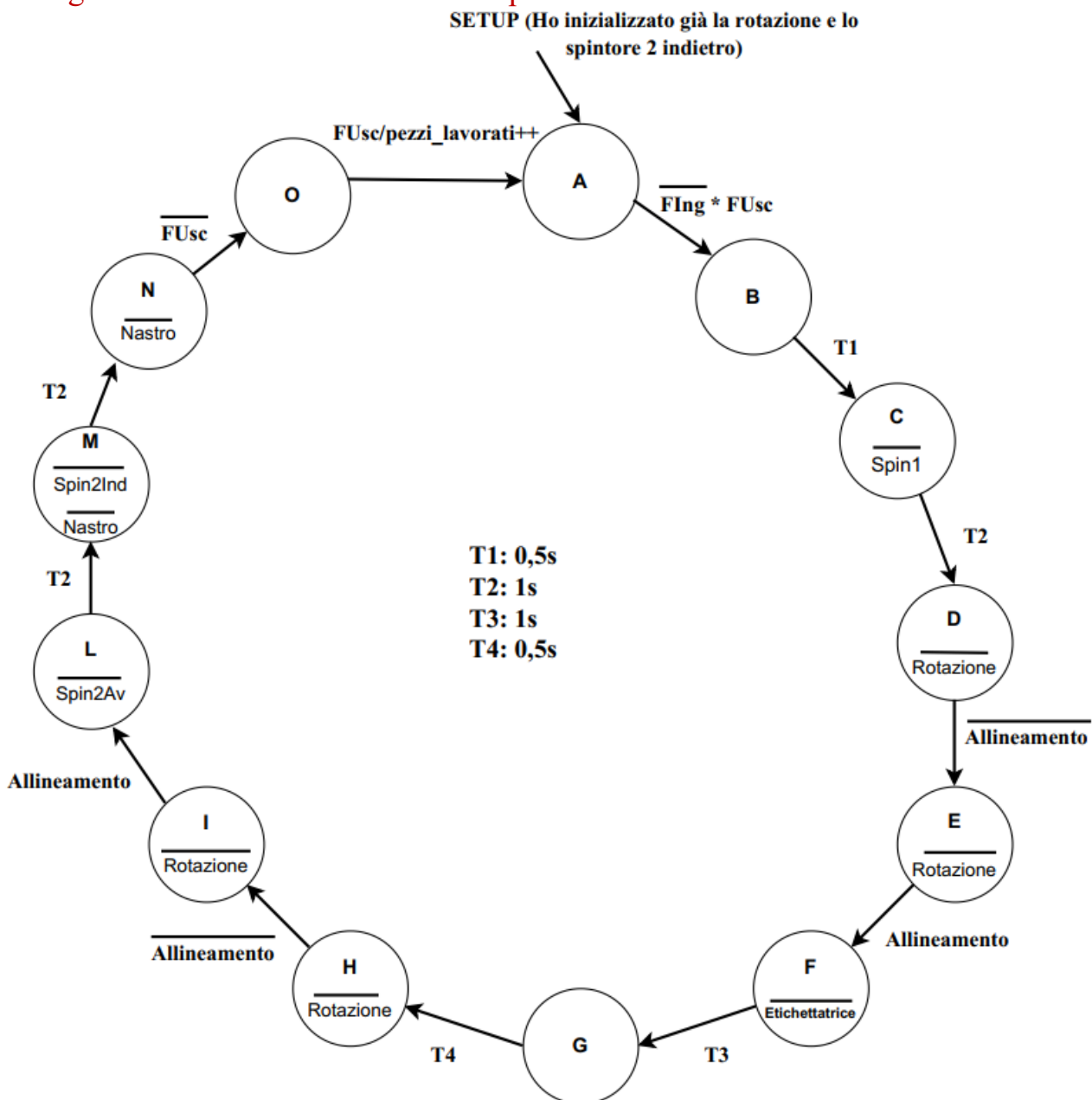
- `lcd.init()`; per inizializzare il display.
- `lcd.backlight()`; per accendere la luce di sfondo.
- `lcd.setCursor()`; per impostare il cursore in un determinato punto del display.
- `lcd.print()`; per scrivere nel display.

Diagramma di flusso dell'algoritmo di gestione del sistema:



Dopo aver inizializzato il sistema impostando pin di ingresso e uscita e inizializzando lo stato dell'automa e le uscite del sistema, nel ciclo programma si controlla lo stato dei pulsanti di accensione e della temperatura: se questi sono idonei si esegue il codice relativo al pallogramma, in caso contrario si disattivano tutte le uscite e si resetta lo stato del sistema. Inoltre, ogni secondo se il programma non è stato interrotto, viene aggiornato il display LCD con il numero dei pezzi lavorati il tempo di lavoro e la temperatura ambientale letta dal sensore.

Pallogramma relativo all'automa monopezzo:



Sono stati utilizzati 4 temporizzatori i cui valori sono riportati al centro del pallogramma. Il programma è composto da 13 stati.

Descrizione dettagliata del pallogramma monopezzo:

Si inizializza il programma, in questo processo accendiamo il compressore e ci assicuriamo che il cilindro 2 sia indietro in posizione di attesa attivando “Spin2Ind” per 0,5 secondi, inoltre allineiamo la piattaforma rotante.

Dopo l’inizializzazione il programma è pronto per accogliere il primo pezzo, il quale occupa la prima fotocellula (FIng=0), inoltre il sistema si assicura che non vi sia nessun pezzo alla fine della lavorazione (FUsc=1) prima di proseguire.

Dopo mezzo secondo dall’inserimento (T1[0,5s]), il pezzo viene spinto dal primo cilindro (Spin1=0) sulla piattaforma rotante.

Dopo un secondo (T2[1s]) la piattaforma viene azionata (Rotazione=0).

Se la piattaforma non è allineata (Allineamento=0) allora continua la rotazione, mentre se è allineata (Allineamento=1) il sistema attiva l’etichettatrice (Etichettatrice=0), che terminerà la lavorazione dopo un secondo (T3[1s]).

Dopo mezzo secondo (T4[0,5s]) aziono la rotazione (Rotazione=0).

Se la piattaforma non è allineata (Allineamento=0) allora continua la rotazione, mentre se è allineata (Allineamento=1) il sistema attiva il cilindro finale in avanti (Spin2Av=0).

Dopo un secondo (T2[1s]) attivo il nastro finale (Nastro=0) e attivo il cilindro finale in posizione indietro per metterlo in posizione di attesa di un altro pezzo (Spin2Ind=0).

Dopo un secondo (T2[1s]) disattivo il cilindro finale (Spin2Ind=1) e quando il pezzo arriva nella fotocellula finale (FUsc=0) disattivo il nastro trasportatore (Nastro=1).

Quando tolgo il pezzo dalla fotocellula finale (FUsc=1) la variabile pezzi_lavorati viene aumentata e verrà aggiornato di conseguenza il display.

Codice/FW di gestione del sistema:

Lo sketch Arduino, ossia il FW (FirmWare) di controllo della scheda, è allegato nel file “Stazione_Pneumatica_V5.ino”.

Il FW si caratterizza per la seguente occupazione di memoria programma e di memoria dati:

Lo sketch usa 8542 byte (3%) dello spazio disponibile per i programmi. Il massimo è 253952 byte. Le variabili globali usano 605 byte (7%) di memoria dinamica, lasciando altri 7587 byte liberi per le variabili locali. Il massimo è 8192 byte.

Tutte le temporizzazioni sono realizzate in background sfruttando la funzione millis().

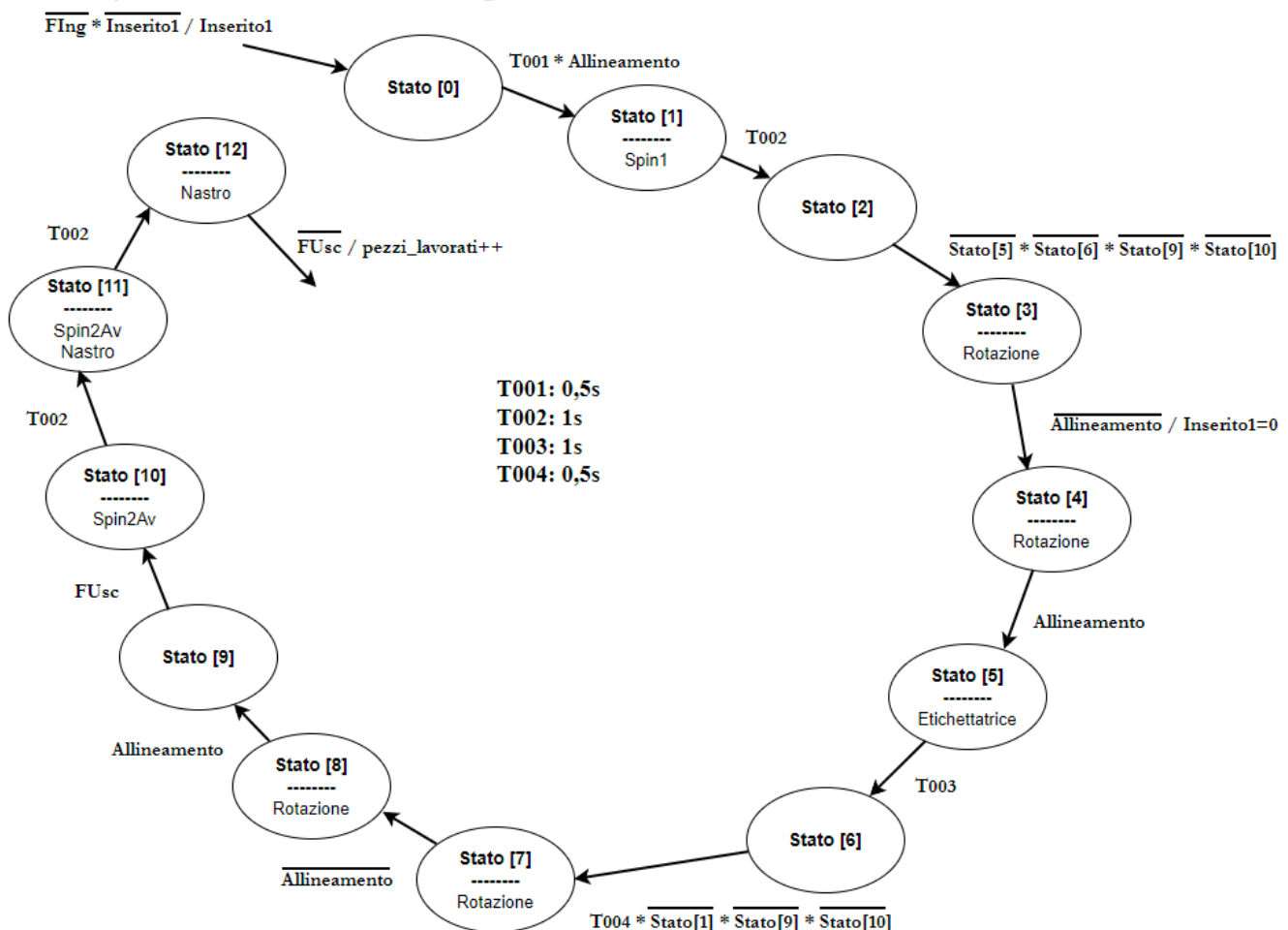
millis() viene utilizzata anche per la misura del tempo di lavoro. La gestione del display LCD avviene in contemporanea alla misura del tempo di lavoro, e con essa anche la misurazione della temperatura attuale.

Idea relativa al sistema multipezzo:

Per il sistema multipezzo si pensa di utilizzare un vettore booleano per contenere il valore dei singoli stati, così da poterli gestire individualmente e in contemporanea. Si vogliono usare più variabili “unsigned long int” per tenere traccia del tempo trascorso, una per ogni settore del sistema. Si vuole utilizzare una variabile booleana: “Inserito1”, per la gestione dell’ingresso dei pezzi nel sistema.

NB: come idea di diagramma di flusso si vuole utilizzare la stessa tipologia fornita per il controllo del sistema in monopezzo.

Pallogramma del sistema multipezzo:



Sono stati utilizzati 4 temporizzatori i cui valori sono riportati al centro del pallogramma. Il programma è composto da 13 stati.

Descrizione dettagliata del pallogramma multipezzo:

Si inizializza il programma, in questo processo accendiamo il compressore e ci assicuriamo che il cilindro 2 sia indietro in posizione di attesa attivando “Spin2Ind” per 0,5 secondi, inoltre allineiamo la piattaforma rotante.

Quando un pezzo viene inserito (FIng=0) e la variabile d’appoggio Inserito1 vale 0 allora il sistema avvia la lavorazione e mette a uno la variabile inserito 1.

Dopo mezzo secondo (T001[0,5s]) e se la piattaforma è allineata (Allineato=1) attivo il primo cilindro (Spin1=1).

Dopo un secondo (T002[1s]) disattivo il primo cilindro (Spin1=0).

Dopo controllo gli stati [5] [6] [9] [10] e se non sono attivi, attivo la rotazione del piatto (Rotazione=1).

Se la piattaforma non è allineata (Allineamento=0) e la variabile inserito vale 0 continuo la rotazione del piatto.

Quando la piattaforma è allineata (Allineamento= 1) fermo la rotazione del piatto (Rotazione = 0) e attivo l’etichettatrice (Etichettatrice=1).

Dopo un secondo (T003[1s]) disattivo l’etichettatrice.

Dopo mezzo secondo (T004[0,5]) controllo gli stati [1] [9] [10] e se rispettano le loro descrizione attivo la rotazione del piatto (Rotazione=1).

Se la piattaforma non è allineata (Allineamento=0) continuo la rotazione del piatto.

Se la variabile di appoggio (Allineamento= 1) fermo la rotazione del piatto (Rotazione = 0).

Se FUsc non legge alcun pezzo (FUsc=1) attivo il cilindro finale (Spin2Av=1).

Dopo un secondo (T002[1s]) attivo il cilindro indietro (Spin2Ind=1) e il nastro finale (Nastro=1).

Dopo un altro secondo disattivo il cilindro finale (Spin2Ind=0).

Quando la fotocellula finale legge il pezzo (Fusc=0) fermo il nastro finale (Nastro=0).

Quando tolgo il pezzo dalla fotocellula il sistema potrà continuare e la variabile pezzi_lavorati viene aumentata e verrà aggiornato di conseguenza il display.

Codice/FW di gestione del sistema:

Lo sketch Arduino, ossia il FW (FirmWare) di controllo della scheda, è allegato nel file “Stazione_Pneumatica_V4.ino”.

Il FW si caratterizza per la seguente occupazione di memoria programma e di memoria dati:

Lo sketch usa 8928 byte (3%) dello spazio disponibile per i programmi. Il massimo è 253952 byte. Le variabili globali usano 626 byte (7%) di memoria dinamica, lasciando altri 7566 byte liberi per le variabili locali. Il massimo è 8192 byte.

Tutte le temporizzazioni sono realizzate in background sfruttando la funzione millis().

millis() viene utilizzata anche per la misura del tempo di lavoro. La gestione del display LCD avviene in contemporanea alla misura del tempo di lavoro, e con essa anche la misurazione della temperatura attuale.

Conclusioni:

Da questo progetto abbiamo imparato la differenza fra la gestione monopezzo e quella multipezzo del sistema utilizzando una scheda Arduino e il linguaggio strutturato, abbiamo imparato a gestire in maniera simultanea più stati e condizioni di un sistema, il controllo di sicurezza, la gestione di un sensore di temperatura e di un display LCD 16x2 I2C.