NASTRO TRASPORTATORE CON PIC

prof. Fusco Ferdinando

NASTRO TRASPORTATORE CON PIC

Il nastro trasportatore svolge le seguenti funzioni:

- Trasporto pezzi.
- Asciugatura pezzi.
- Scarto pezzi di dimensione superiore ad un valore prefissato.
- Conteggio pezzi.

Le componenti fondamentali:

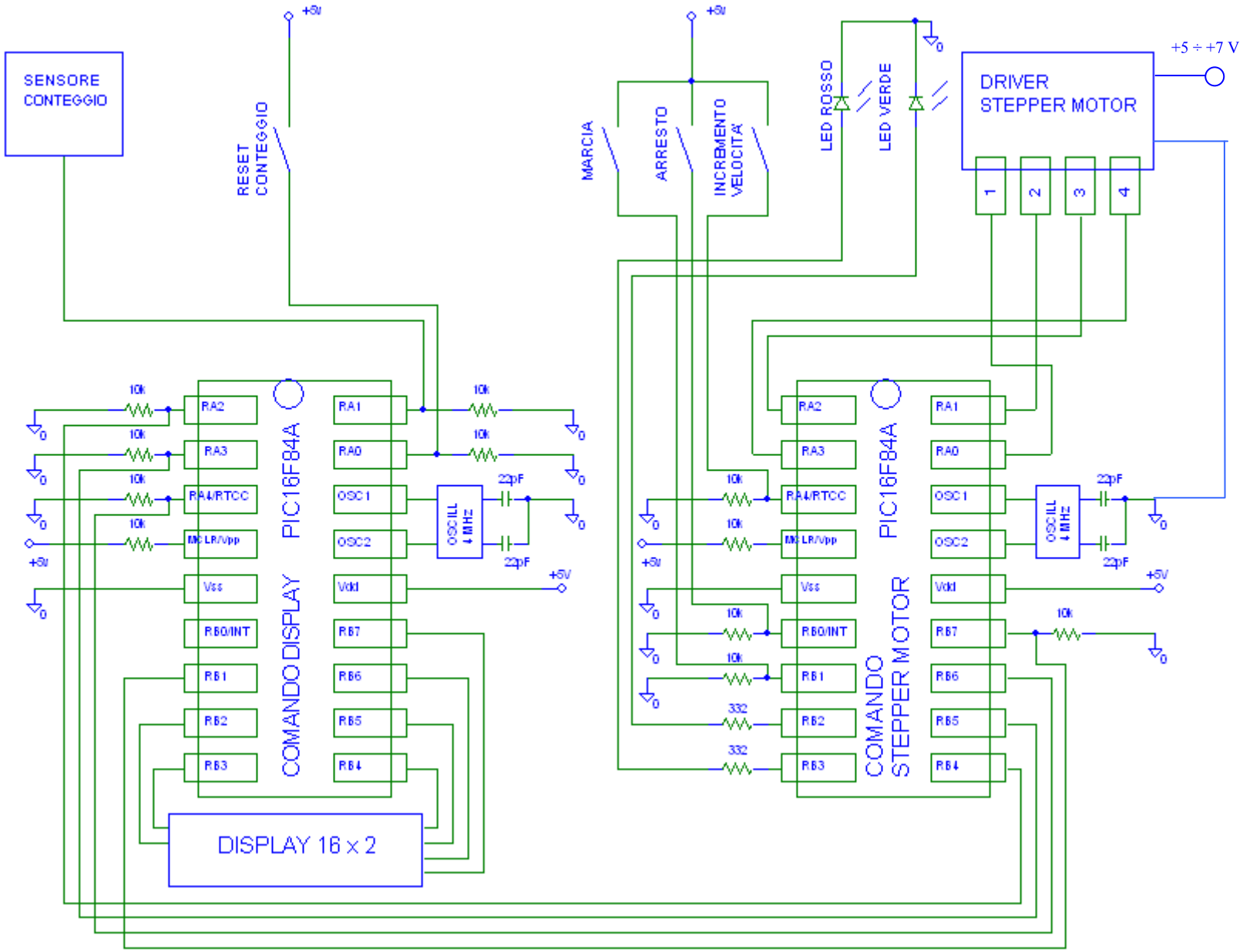
- Due controllori a interfaccia programmabile PIC 16F84A
- Motore passo passo per movimentazione nastro.
- Scheda operatore con display.
- Pistone elettromagnetico.
- Tre sensori IR
- Una ventola
- Scheda controllo del motore passo passo e controllo display.
- Scheda di potenza per azionamento motore.
- Scheda controllo pistone elettromagnetico.
- Scheda controllo ventola e blocco nastro.
- Alimentatori per le 4 schede
- Led di segnalazione

Dalla scheda operatore è possibile svolgere le seguenti funzioni:

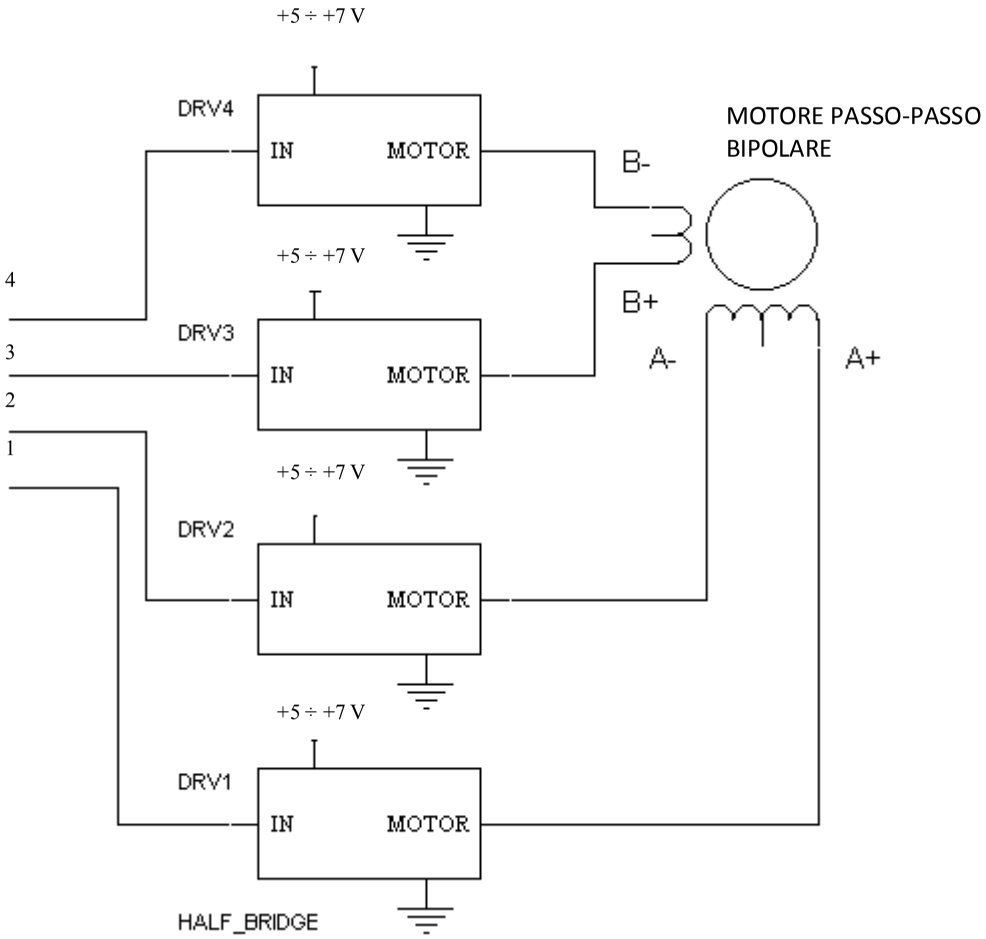
- Marcia del nastro.
- Arresto del nastro.
- Incremento velocità del nastro.
- Reset dei pezzi conteggiati.
  In particolare, quando vengono contati un numero di pezzi pari al numero impostato nel programma il nastro si ferma. Quindi, è necessario un reset del numero di pezzi.

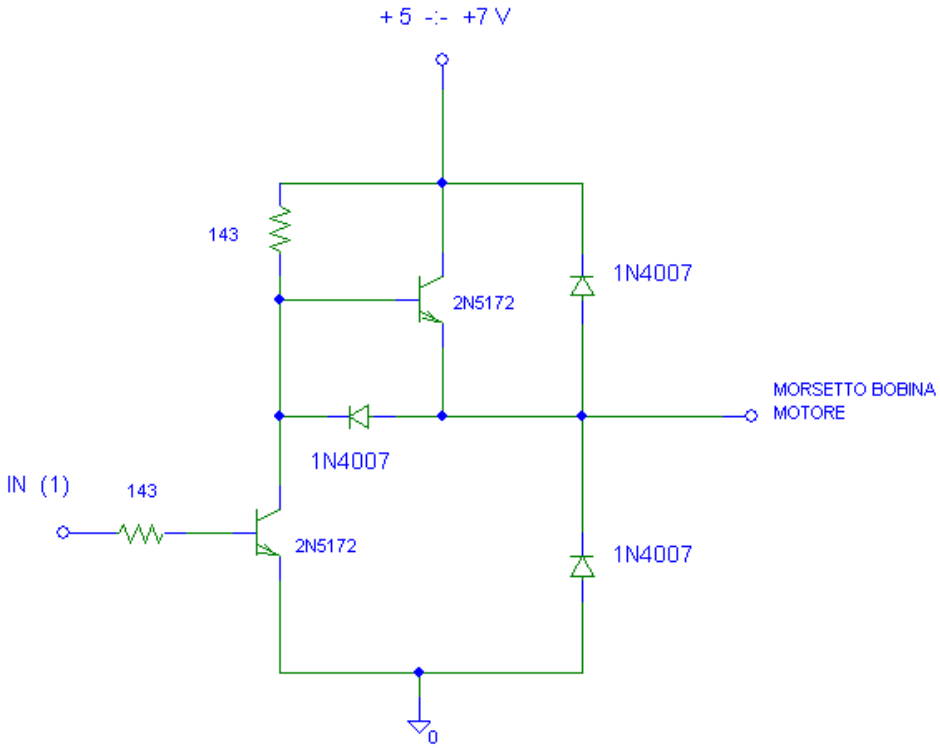
Lo scarto di pezzi con lunghezza superiore a quella prevista, dall'altezza del sensore IR a taglio di fascio, è automatico grazie al pistone elettromagnetico.
Si riportano di seguito le schede elettroniche realizzate, i programmi in linguaggio MikroBasic utilizzati per programmare i PIC e le foto del progetto finito.

SCHEDA CONTROLLO PIC



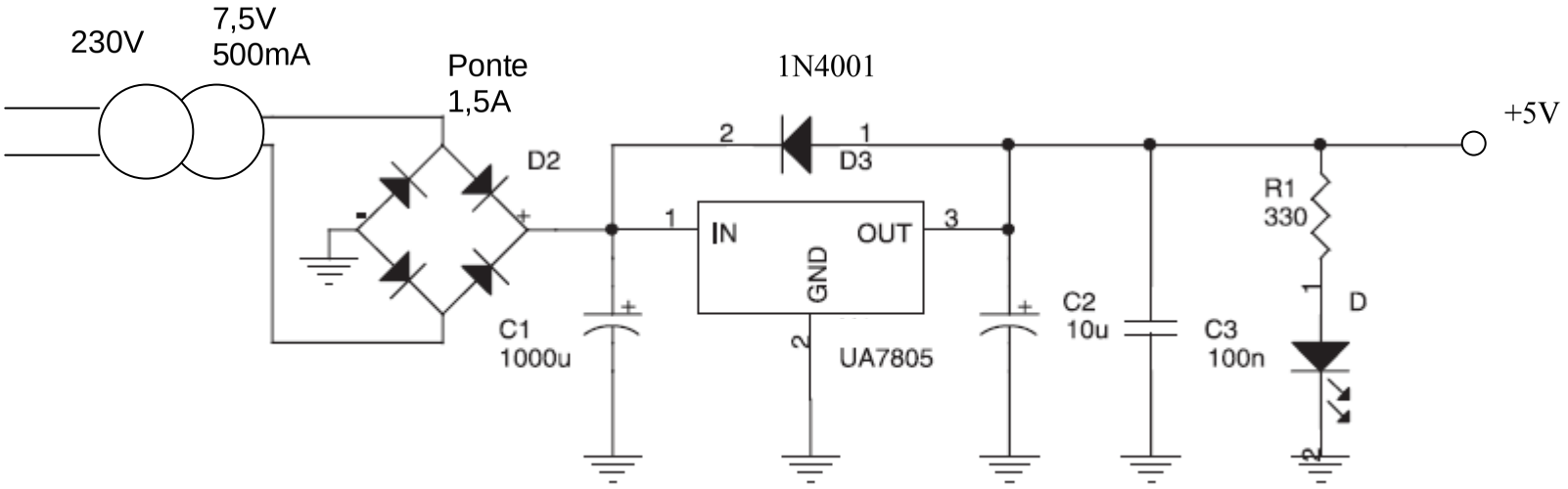
SCHEDA DRIVER PER CONTROLLO MOTORE PASSO PASSO





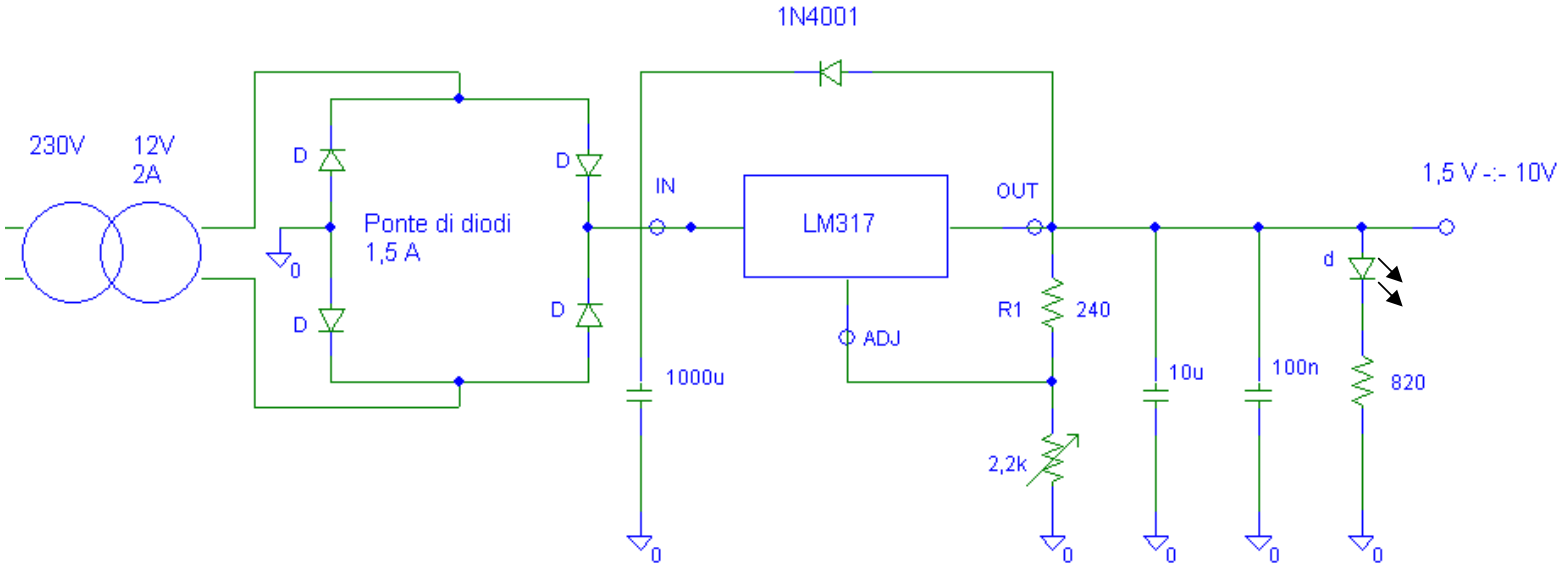
DRIVER SINGOLO
(Due PER BOBINA)

ALIMENTATORE PER SCHEDA PIC



ALIMENTATORE A TENSIONE VARIABILE (MAX 1A)
PER MOTORE STEPPER

Volendo variare la coppia (non la velocità) del motore passo passo si può agire sulla resistenza variabile 2,2k in modo da avere una regolazione sulla tensione d'uscita dell'alimentatore. Si consiglia una variazione tra i 5V e 10V.



PROGRAMMA PER IL PIC COMANDO MOTORE

```
program Nastrotrasportatore_mezzo

dim pulse as byte ' definizione variabile pulse

main:
portb=0    ' azzera tutte le porte
porta=0

trisa.0=0 'uscite per il circuito di potenza del motore stepper
trisa.1=0 '     "          "            "              "
trisa.2=0 '     "          "            "              "
trisa.3=0 '     "          "            "              "
trisa.4=1
trisb.0=1 ' porta B0 ferma tutto
trisb.1=1 ' porta B1 rotazione e incremento
trisb.2=0 ' led verde
trisb.3=0 ' led rosso
trisb.4=0
trisb.5=0
trisb.6=0
trisb.7=1   ' richiesta fermata PIC_display per numero pezzi=10

inizio:
portb=0    ' azzero tutte le porte quando c'è un comando di ferma tutto
porta=0
pulse=10    ' assegnato valore iniziale alla variabile pulse

portb.2=1 'led verde acceso
if portb.1=1 then
   portb.6=1           ' colloquia con PIC display per l'avvio
   delay_ms(100)
   portb.6=0

    portb.2=0 'led verde spento
```

```
    portb.3=1 'led rosso acceso

    while true

       if (porta.4=1)and(pulse>1) then ' come sotto
           pulse=pulse-1               ' come sotto
       portb.4=1               ' comunica al secondo pic il decremento di pulse
       while porta.4=1         'Routine Antirepeat
       wend
       portb.4=0
       end if

       porta.0=1               '1° mezzo passo
       Vdelay_ms(20*pulse)

       if (portb.0=1)then          'ferma tutto
        portb.5=1
        delay_ms(100)
        portb.5=0
       goto inizio
       end if

       porta.2=1               '2° mezzo passo
       Vdelay_ms(20*pulse)
                               ' porta.0=1

       if (portb.0=1)then         'ferma tutto
        portb.5=1
        delay_ms(100)
        portb.5=0
       goto inizio
       end if
```

```
        porta.0=0                '3° mezzo passo
        Vdelay_ms(20*pulse)
                                 ' porta.2=1


        if (portb.0=1)then           'ferma tutto
         portb.5=1
         delay_ms(100)
         portb.5=0
        goto inizio
        end if


        porta.1=1                '4° mezzo passo
        Vdelay_ms(20*pulse)
                                 ' porta.2=1


        if (portb.0=1)then           'ferma tutto
         portb.5=1
         delay_ms(100)
         portb.5=0
        goto inizio
        end if

        porta.2=0                '5° mezzo passo
        Vdelay_ms(20*pulse)
                                 ' porta.1=1

        if (portb.0=1)then          'ferma tutto
         portb.5=1
         delay_ms(100)
         portb.5=0
```

```
        goto inizio
        end if

        porta.3=1                '6° mezzo passo
        Vdelay_ms(20*pulse)
                                 ' porta.1=1


        if (portb.0=1) then          'ferma tutto
         portb.5=1
         delay_ms(100)
         portb.5=0
        goto inizio
        end if

        porta.1=0                '7° mezzo passo
        Vdelay_ms(20*pulse)

                                 ' porta.3=1

        if (portb.0=1)then           'ferma tutto
         portb.5=1
         delay_ms(100)
         portb.5=0
        goto inizio
        end if

        porta.0=1                '8° mezzo passo
        Vdelay_ms(20*pulse)
        porta.3=0
```

```
         if (portb.0=1) or (portb.7=1)then
          portb.5=1
          delay_ms(100)
          portb.5=0
         goto inizio
         end if

    wend

end if

goto inizio    ' nel caso non venga premuto alcun pulsante di inizio
end.
```

PROGRAMMA PER IL PIC COMANDO DISPLAY

```
program Nastrotrsportatore_schermo

dim pezzi as byte               ' variabile pezzi contati

dim pulse as byte               ' variabile del numero di giri del motore

dim velocita as byte            ' variabile velocità nastro

dim velocita1 as string[5]      ' stringa caratteri per visualizzare la velocità
                                ' con numeri decimali risparmiando
                                ' sulla memoria del PIC

dim txt_pezzi as char[3]        ' variabile per visualizzare il n. pezzi contati

main:
  trisa.0=1    ' Ingresso per pulsante azzera numero di pezzi

  trisa.1=1    ' Ingresso per sensore passaggio pezzi

  trisa.2=1    ' Ingresso collegato alla porta RB4 del PIC_movimento nastro
               ' per decremento variabile pulse (aumento di velocità)

  trisa.3=1    ' Ingresso collegato alla porta RB5 del PIC_movimento nastro
               ' per trasmettere l'avvenuta fermata del nastro

  trisa.4=1    ' Ingresso collegato alla porta RB6 del PIC_movimento nastro
               ' per trasmettere l'avvio del nastro a velocità minima

  trisb.1=0    ' Collegato alla porta RB3 del PIC_movimento nastro
               ' per trasmettere il comando d'arresto del nastro quando il
               ' a velocità minima

  portb=0      'Azzera portb
  porta=0      'Azzera porta
```

```
 
  pezzi=0
  velocita=0
  pulse=1000              ' si assegna inizialmente un valore alto per
                          ' ottenere 0,00 a display
 
  Lcd_Init(PORTB)         'Inizializza DISPLAY
  lcd_cmd(LCD_CLEAR)      'Invia comando di cancellazione
  lcd_cmd(LCD_CURSOR_OFF)'Spegne il cursore

while true                  ' ciclo continuo
 
      delay_ms(10)          ' pausa antirimbalzo
      if porta.1=0 then     ' passaggio pezzo tensione bassa
         pezzi=pezzi+1      ' Incrementa eventi passaggio pezzi
         while porta.1=0    ' Routine Antirepeat
         wend
 
      end if

      lcd_out(1,2,"PEZZI")
      bytetostr(pezzi,txt_pezzi)  ' conversione da byte a stringa del numero
                                  ' di pezzi
      lcd_out(1,9,txt_pezzi)      ' visualizzazione del n. pezzi sul display

      if pezzi>=20 then
         portb.1=1
         pulse=1000      ' alla fermata del nastro il display dovrà segnare
                         ' velocità nastro nulla.
         velocita=0
         delay_ms(100)
         portb.1=0
      end if
```

```
        if porta.0=1 then
           pezzi=0
        end if

        if(porta.4=1) then      ' attivazione velocità display
           pulse=10             ' massimo utile della variabile pulse
           while porta.4=1      ' Routine Antirepeat
           wend
        end if

        if (porta.2=1)then      ' dall'altro PIC
           pulse=pulse-1        ' aumenta velocità
           while (porta.2=1)    ' Routine Antirepeat
           wend
        end if

        velocita=164/pulse    ' velocità max del nastro è di 1,64 cm/s (si
                              ' moltiplica per 100
                              ' per poter ottenere tramite le operazioni sotto una
                              ' suddivisione in decimali senza usare una variabile
                              ' float che assorbirebbe tantissima memoria
        lcd_out(2,2,"Veloc")

        velocita1[0] = (velocita div 100)+48        ' conversione in stringa
        velocita1[1] = "."                            '''
        velocita1[2] = ((velocita div 10)mod 10)+48   '''
        velocita1[3] = (velocita mod 10)+48           '''
        velocita1[4] = 0                              '''

        lcd_out(2,9,velocita1)                      ' visualizzazione sul display
        lcd_out(2,13,"cm/s")

        if porta.3=1 then ' arresto nastro dall'altro PIC
```

```
           pulse=1000      ' alla fermata del nastro il display dovrà segnare
                           ' velocità nastro nulla.
           velocita=0
        end if
wend

end.
```

CALCOLO VELOCITA' DEL NASTRO TRASPORTATORE

La programmazione del PIC prevede, per l'azionamento del motore passo, un tempo regolabile per ogni singolo mezzo passo che può andare dai 200 ms ai 20 ms.

- **Motore stepper** utilizzato: numero di passi per giro = 96

- **Angolo per singolo passo**: $2\pi / 96 = 0{,}06545$ rad

- **Tempo minimo impiegato** per compiere un singolo passo: 0,04 s
  (il programma inserito nel PIC prevede 20 ms come tempo minimo per ogni mezzo passo e quindi 40 ms come tempo minimo per un passo).

- **La variabile PULSE** presente nel programma PIC può essere variata per interi da 1 a 10. Quindi come tempo per ogni passo si assumerà la funzione **t** ="0,04 · **PULSE**" espressa in secondi.

- **La velocità angolare del motore** sarà una variabile:

$$\boldsymbol{\omega = 0{,}06545 / t} = 0{,}06545 / (0{,}04 \cdot \mathbf{PULSE}) =$$

per **PULSE** = 1, $\boldsymbol{\omega =} 0{,}0818125$ rad/s

- **La velocità del nastro** V è legata alla velocità del motore secondo la relazione:

$$\mathbf{V = \omega \cdot R}$$

  con ω velocità angolare del motore e R raggio della puleggia.
  Per R = 1 cm: **V** = 0,06545 / (0,04 · PULSE) ≈ 1,64 / PULSE cm/s

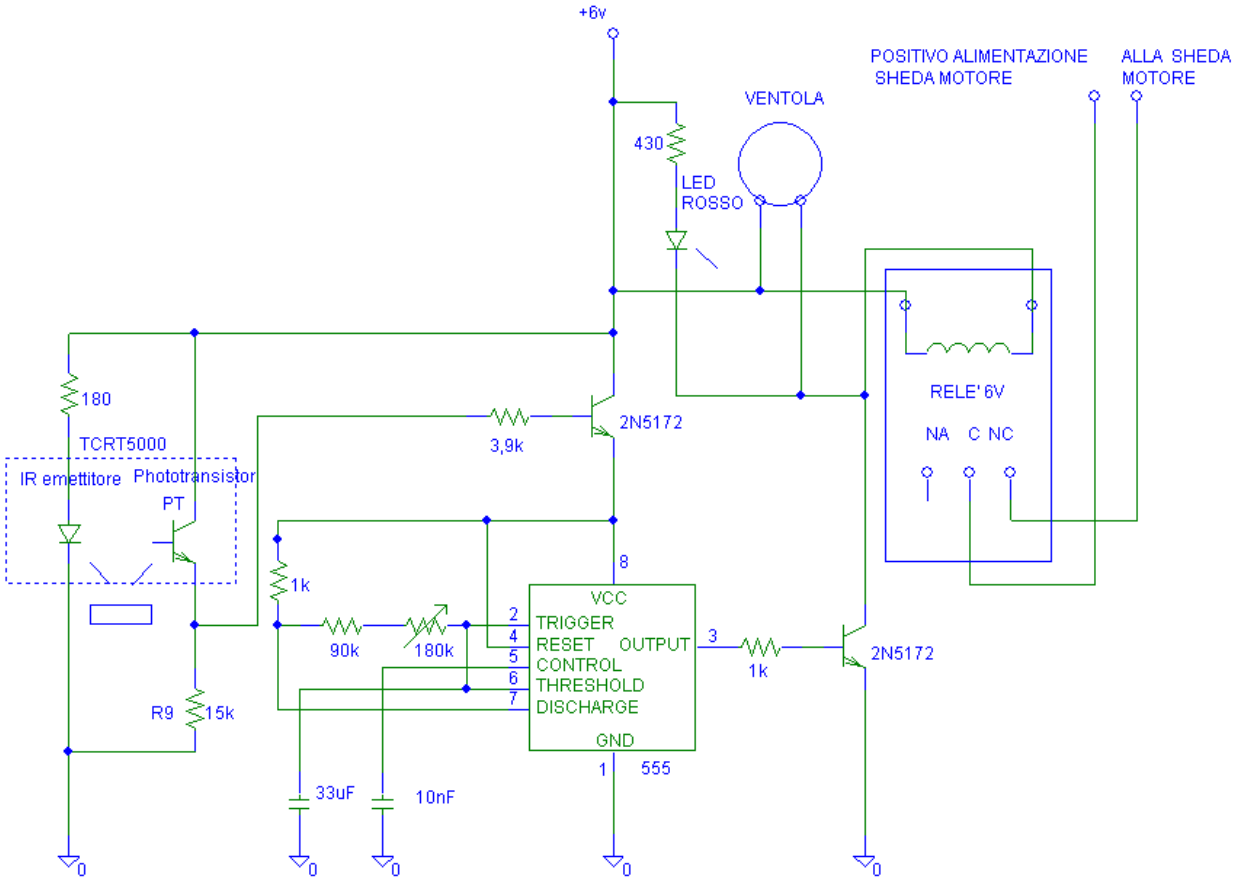
E' quindi 1,64 cm/s il valore di riferimento inserito nell'algoritmo di programmazione del PIC.

SCHEDA PER CONTROLLO VENTOLA

Al passaggio di un pezzo, si attiva, grazie al sensore IR a riflessione, la ventola e contemporaneamente si ferma il nastro.

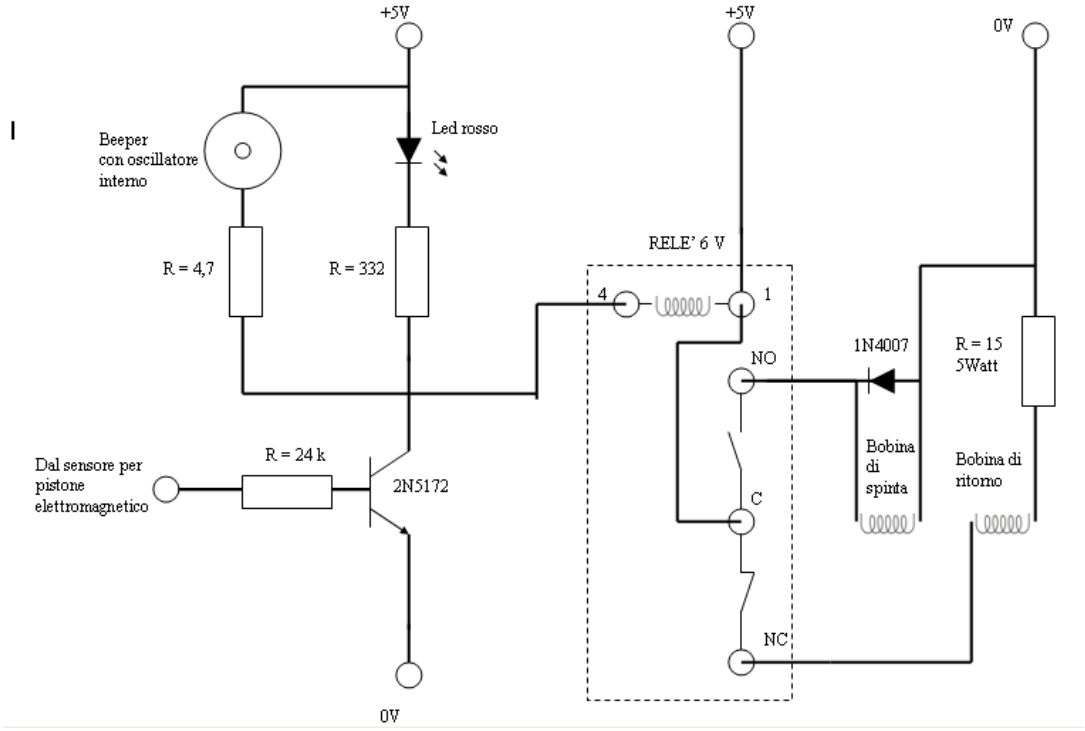
Il tempo di fermata e di azionamento ventola può essere variato, agendo sul trimmer nello schema seguente, in un intervallo di tempo tra i 2s e 6s circa.

Il nastro viene bloccato tagliando temporaneamente, grazie ad un relè, la tensione di alimentazione della scheda controllo motore (dato che tutti i pin ingressi/uscita del PIC controllo motore sono stati già utilizzati).

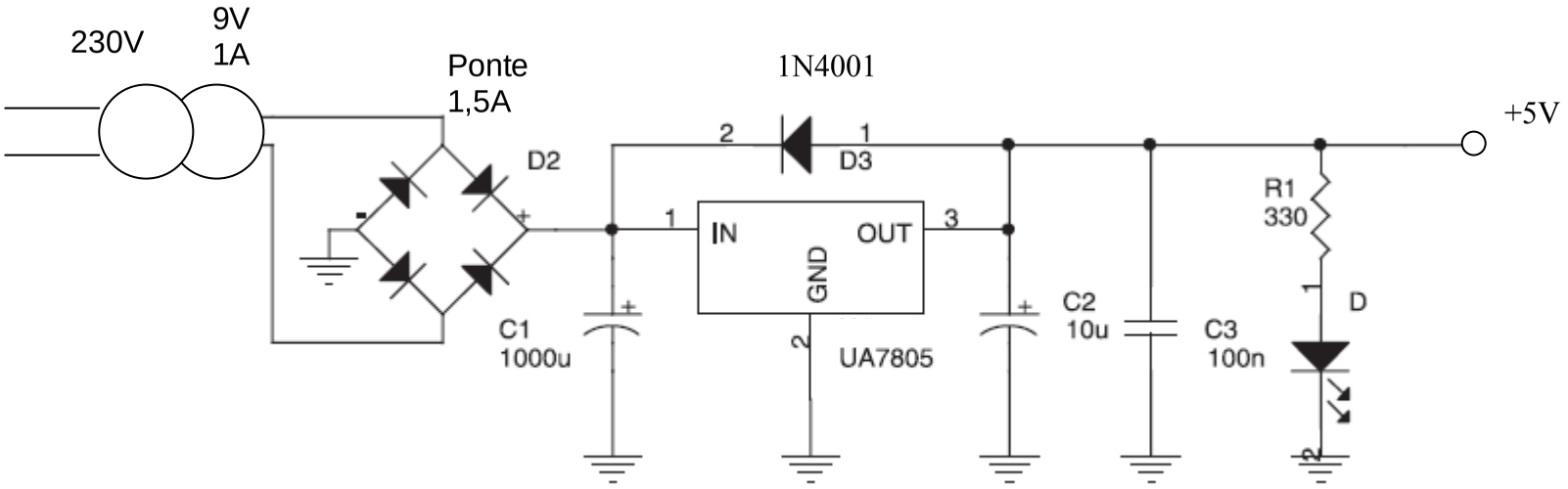


SCHEDA PER PISTONE ELETTROMAGNETICO

Al passaggio di un pezzo di altezza maggiore della misura massima, si attiva, grazie al sensore a taglio di fascio IR, il pistone elettromagnetico costituito da due bobine, affiancate con stessa direzione e mai alimentate insieme, che creano due campi elettromagnetici di verso opposto.
La parte mobile del pistone ha un'anima composta da una serie di magneti permanenti al neodimio.
Il campo elettromagnetico prodotto dalla bobina di ritorno tiene fermo il pistone nella posizione di riposo; esso è sempre attivo tranne che nell'intervallo di tempo in cui viene attivata la bobina di spinta al passaggio di un pezzo troppo alto.
Il campo elettromagnetico prodotto dalla bobina di spinta attiva il pistone nel suo massimo spostamento per espellere il pezzo fuori misura.
La resistenza in serie alla bobina di ritorno ha la funzione di limitare la corrente nella bobina e quindi il riscaldamento di quest'ultima, dato che questa è sempre alimentata.



ALIMENTATORE PER PISTONE ELETTROMAGNETICO



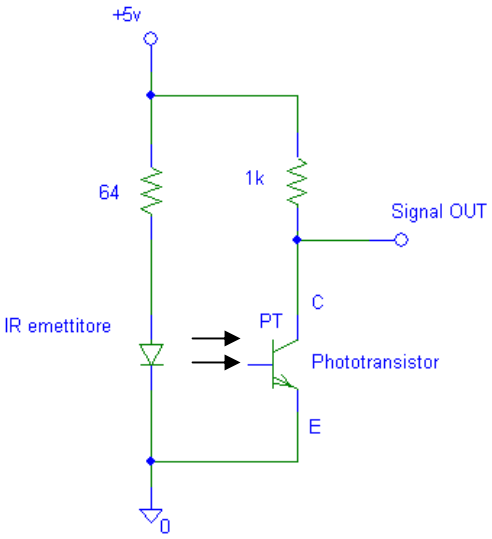
I SENSORI OTTICI IR

Il sensore ottico a riflessione (conta pezzi) comunica con il PIC collegato al Display.

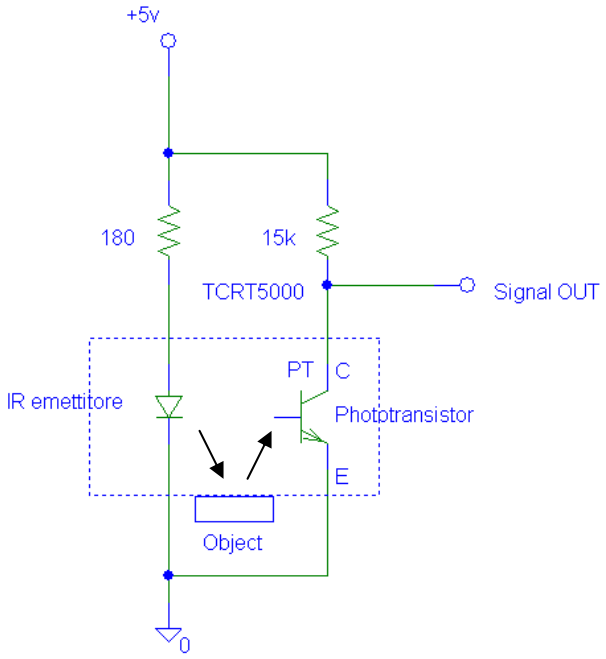
Il sensore ad interruzione di fascio, comunica con la scheda pistone elettromagnetico.

Per il sensore ad interruzione di fascio l'uscita a riposo sarà BASSA. Quando viene interrotto il fascio l'uscita sarà ALTA.

Per il conta pezzi il sensore a riposo darà un'USCITA ALTA. Viceversa, al passaggio di un pezzo, l'uscita sarà BASSA.



Fotosensore ad interruzione di fascio per azionamento pistone

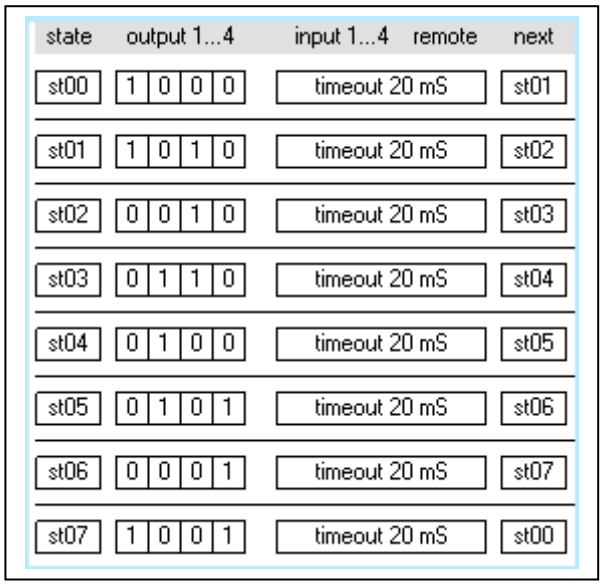


Fotosensore a riflessione

Tavola degli stati, pilotaggio *"half step"*

L'*half stepping* ha vantaggi e svantaggi:

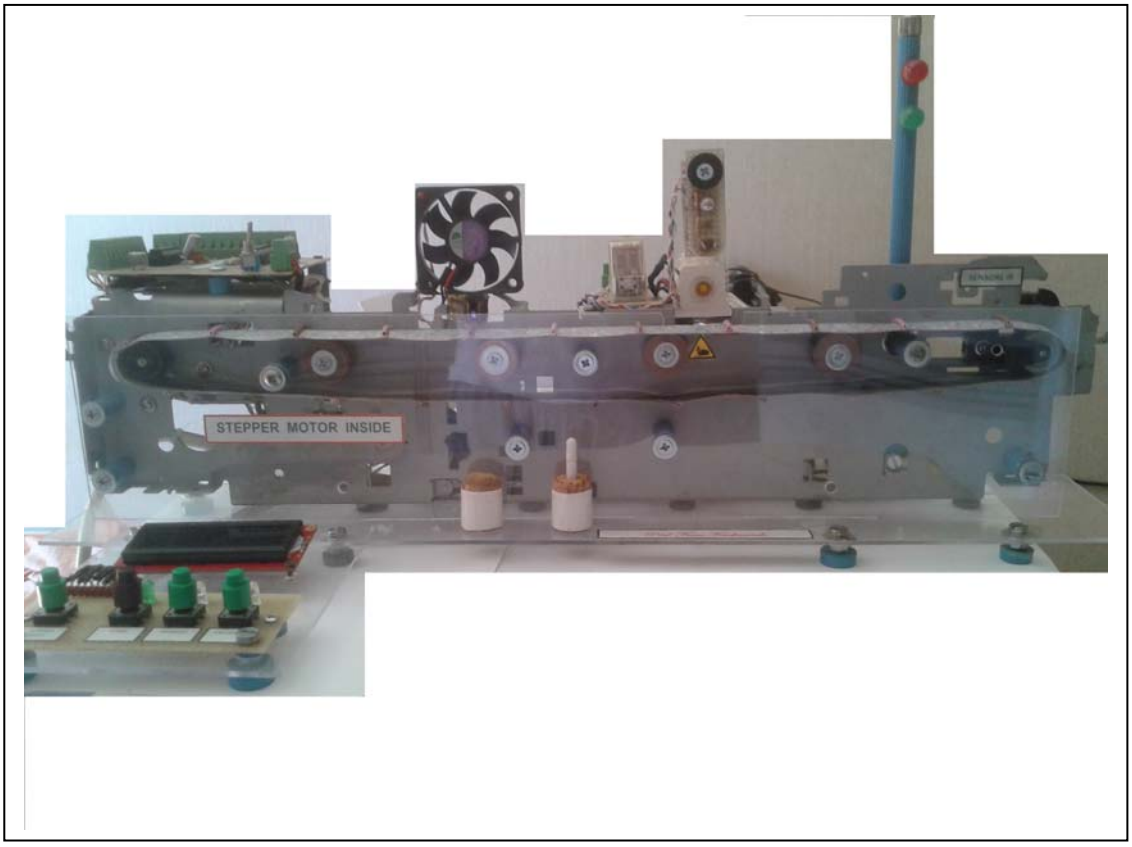
- **dà molta forza** alla rotazione (per metà del tempo si attivano 2 avvolgimenti anzichè 1)
- attenzione però a non **surriscaldare** il motore (che deve dissipare l'energia di 2 Avvolgimenti al posto di uno solo)
- la **velocità massima si dimezza** (è raddoppiato il numero di passi)
- la **precisione di movimento raddoppia** (facciamo due passi dove altrimenti se ne faceva uno).



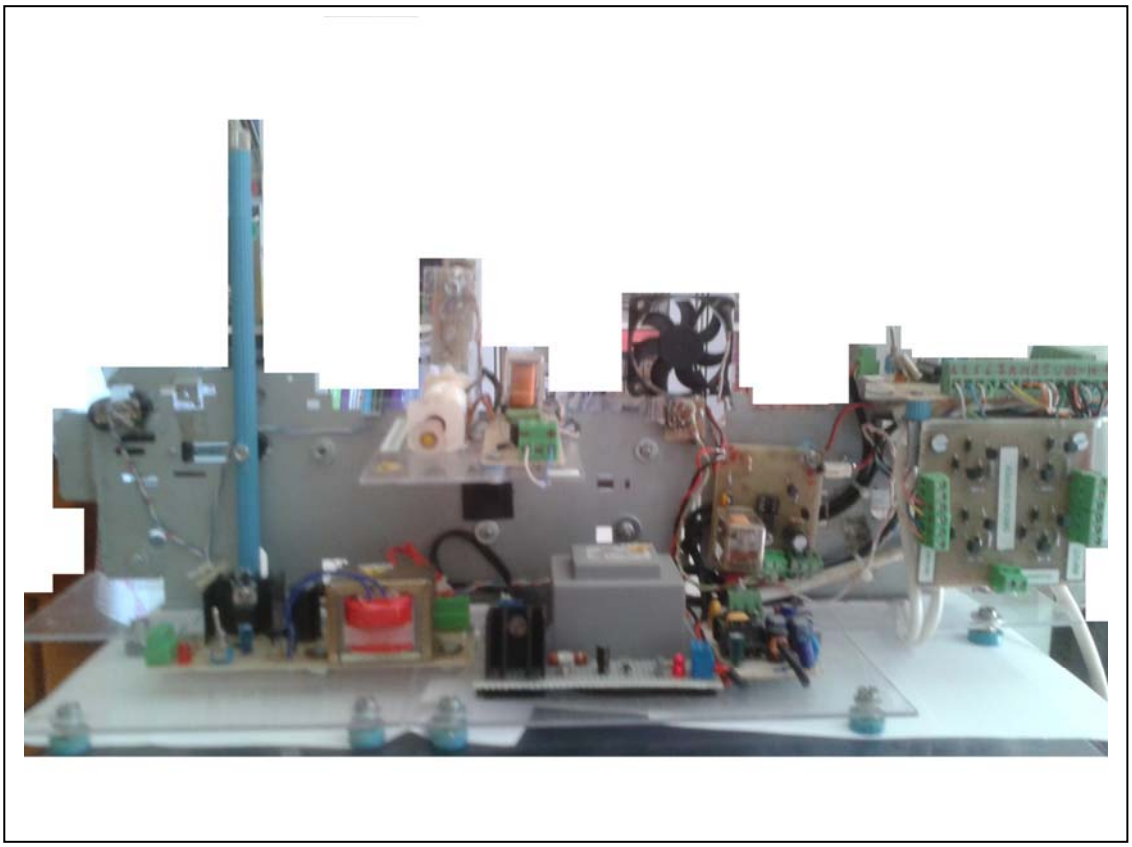
state	output 1...4	input 1...4	remote	next
st00	1 0 0 0	timeout 20 mS		st01
st01	1 0 1 0	timeout 20 mS		st02
st02	0 0 1 0	timeout 20 mS		st03
st03	0 1 1 0	timeout 20 mS		st04
st04	0 1 0 0	timeout 20 mS		st05
st05	0 1 0 1	timeout 20 mS		st06
st06	0 0 0 1	timeout 20 mS		st07
st07	1 0 0 1	timeout 20 mS		st00

Tavola della verità per azionare uno STEPPER MOTOR
con tecnica HALF STEP

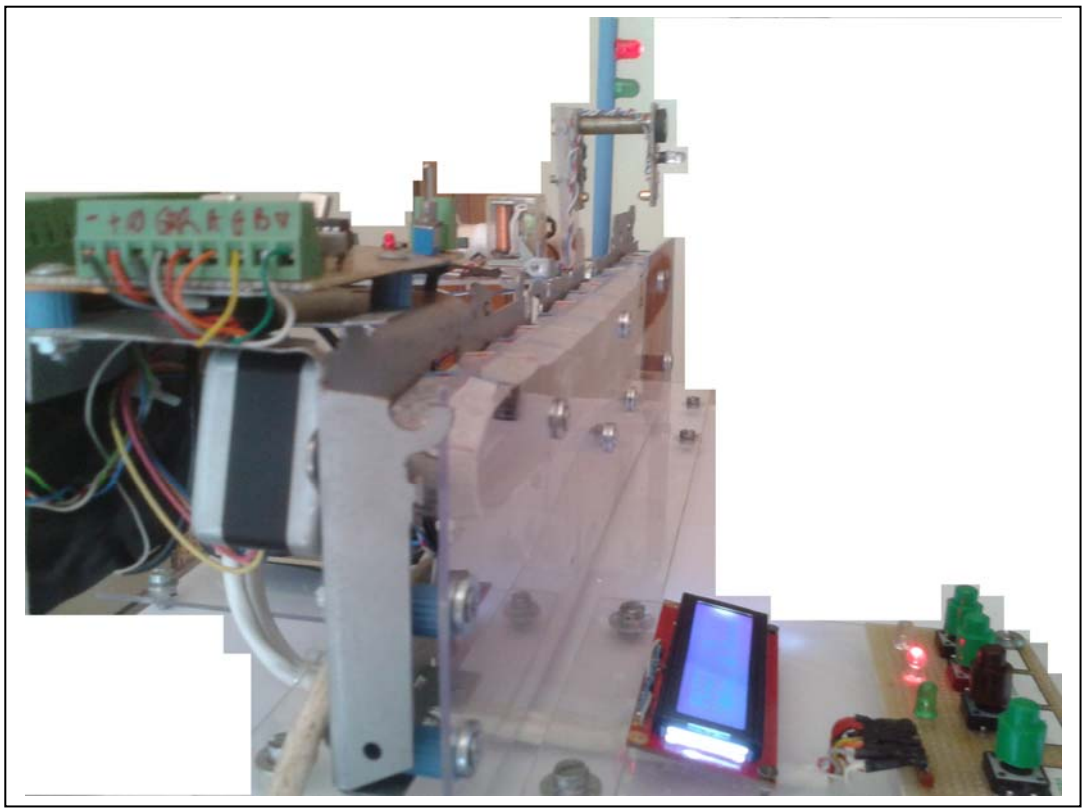
LE FOTO DEL PROGETTO
NASTRO TRASPORTATORE



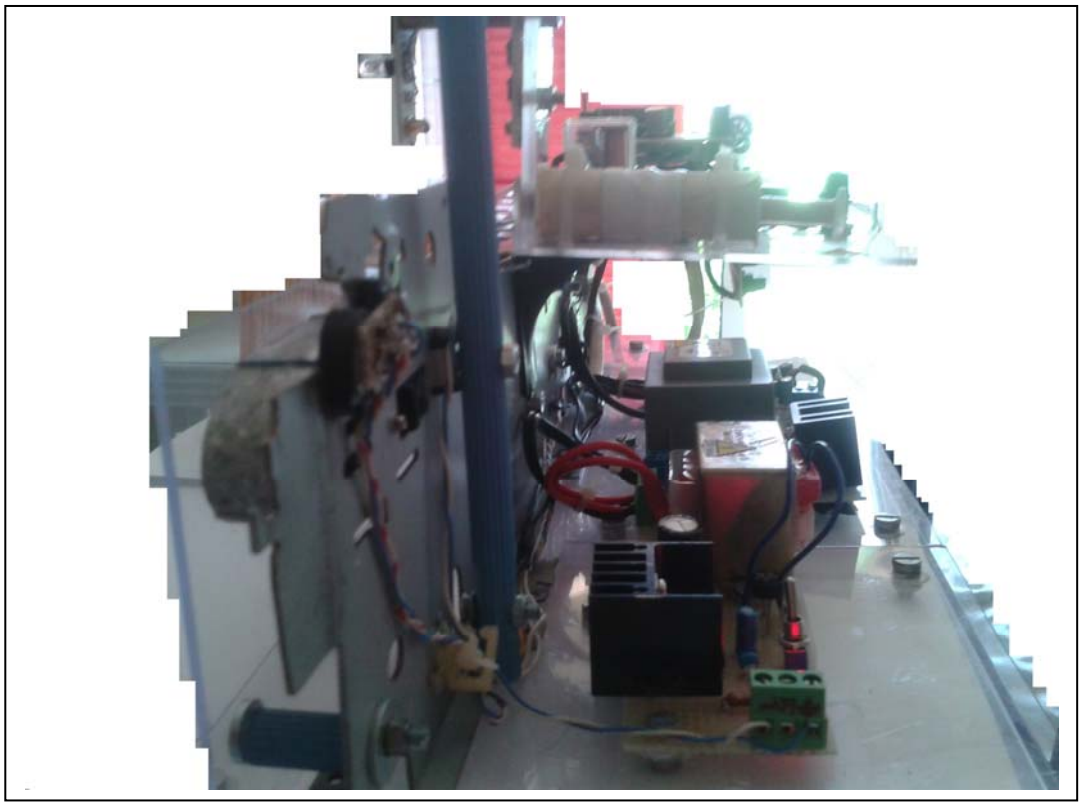
Vista frontale



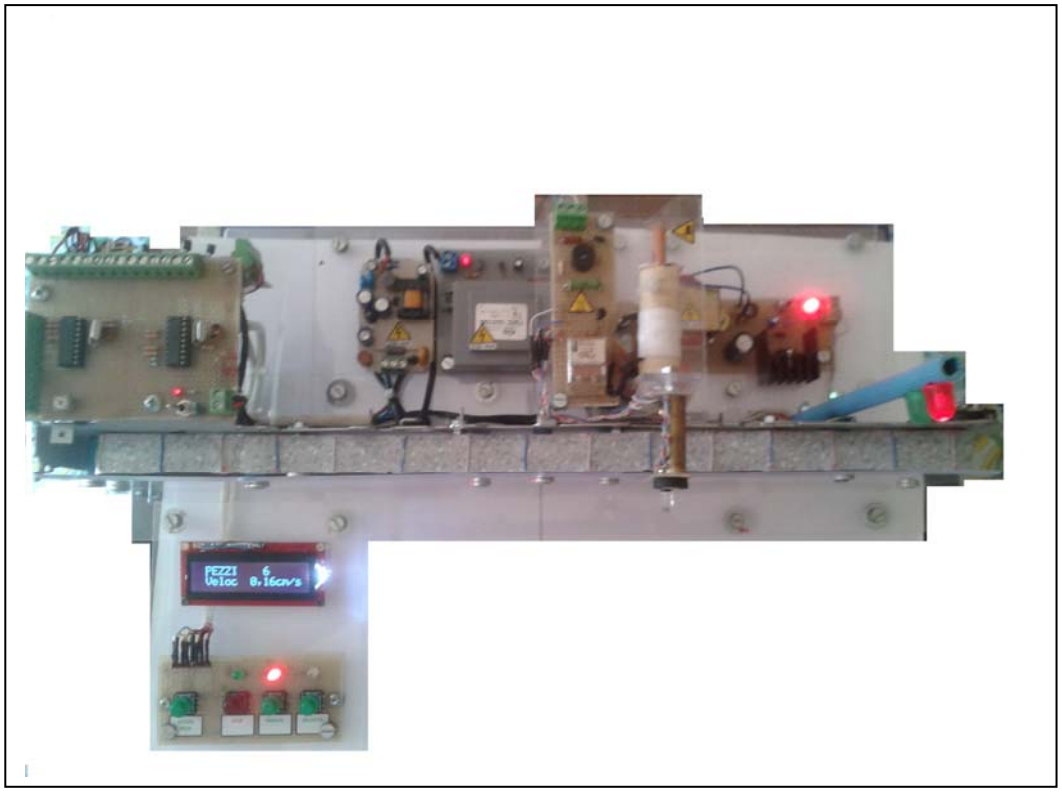
Vista posteriore



Vista laterale 1

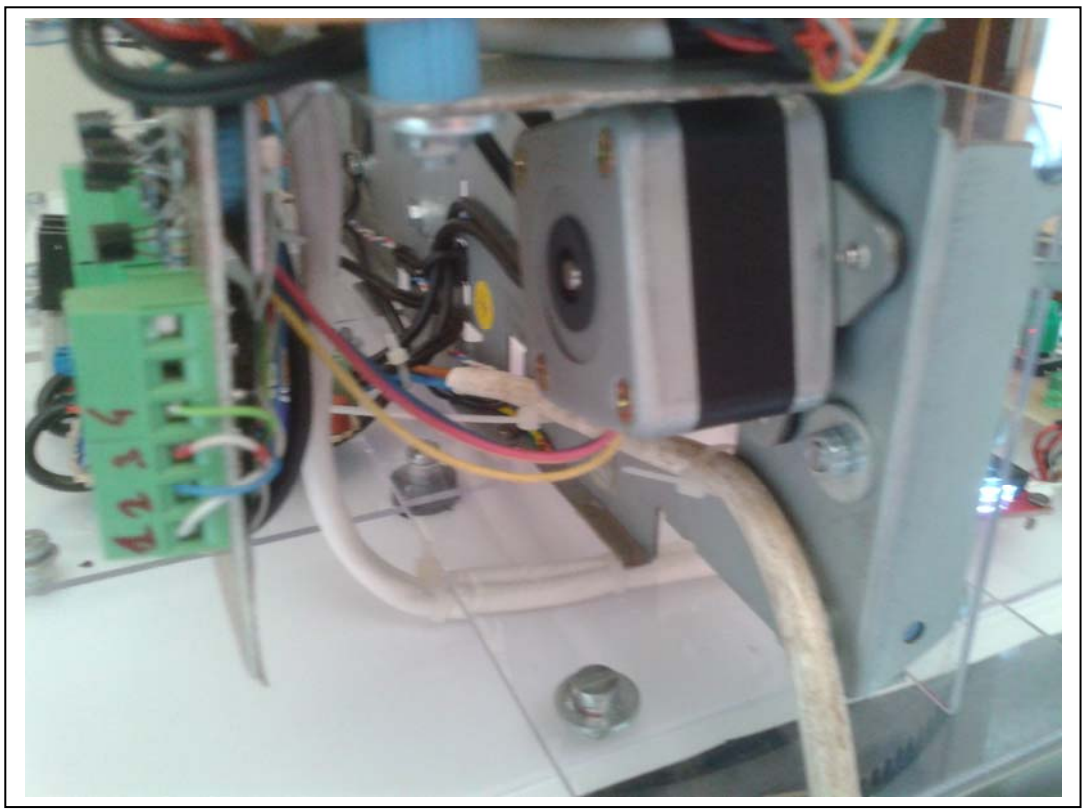


Vista laterale 2

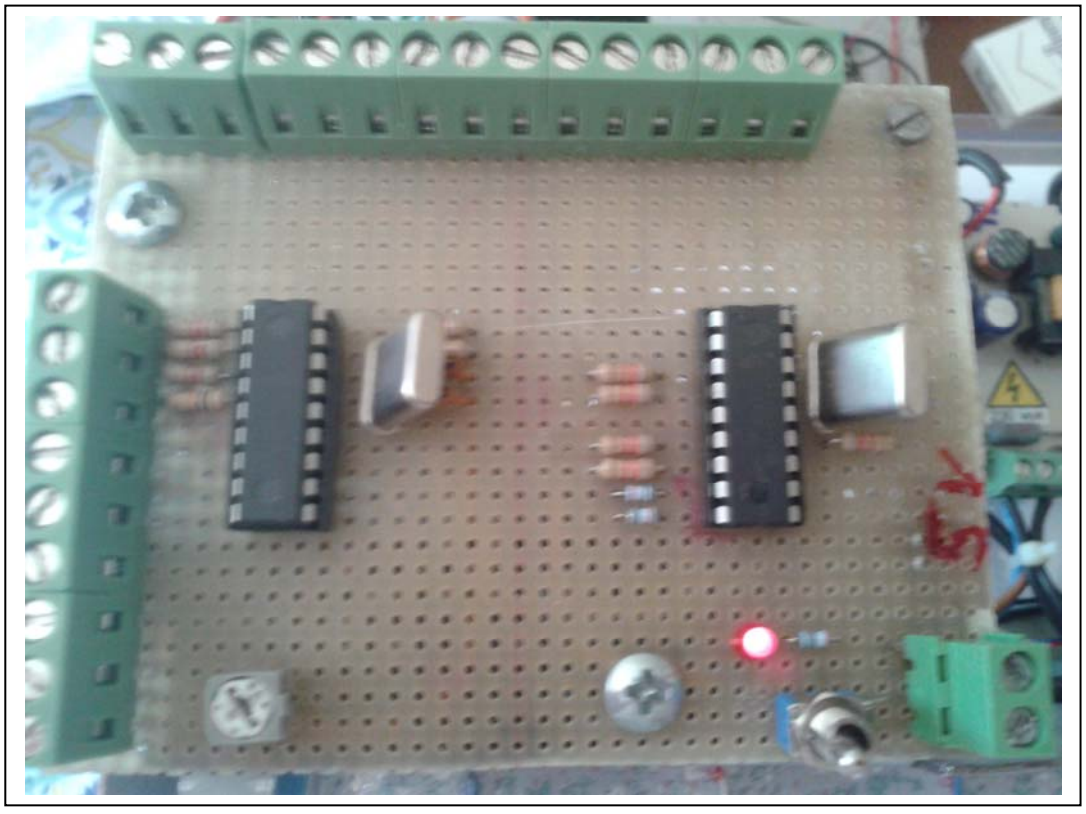


Vista dall'alto

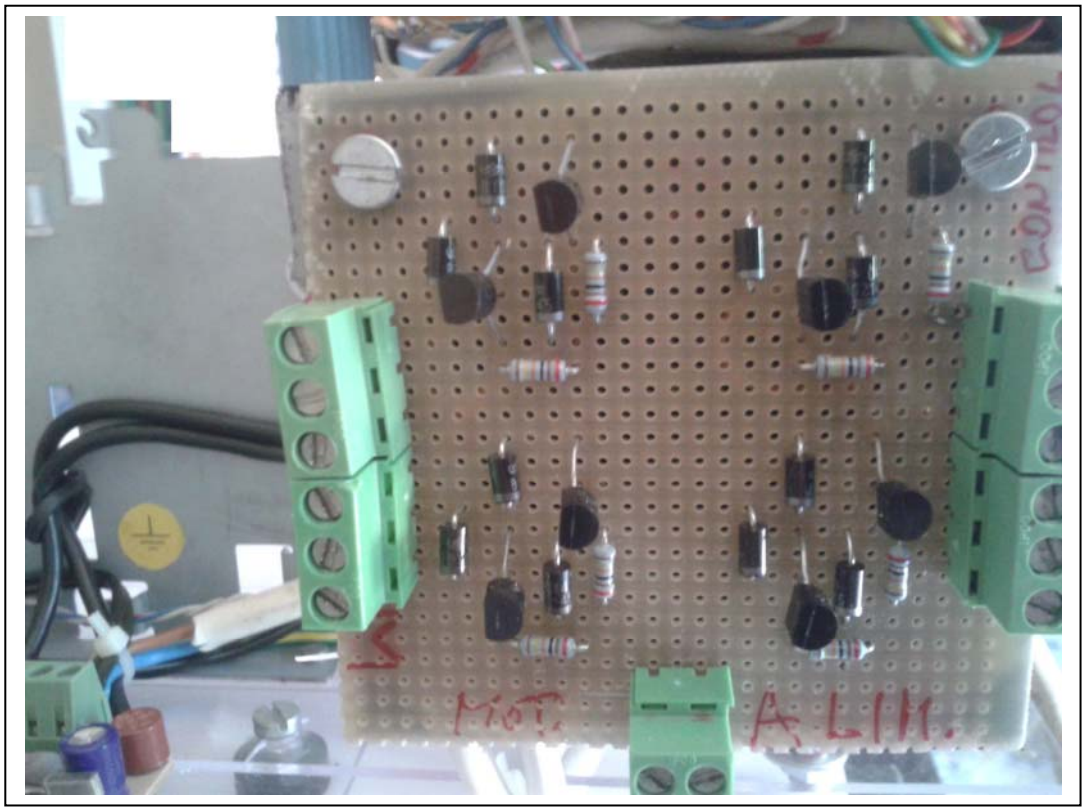
Particolare scheda operatore



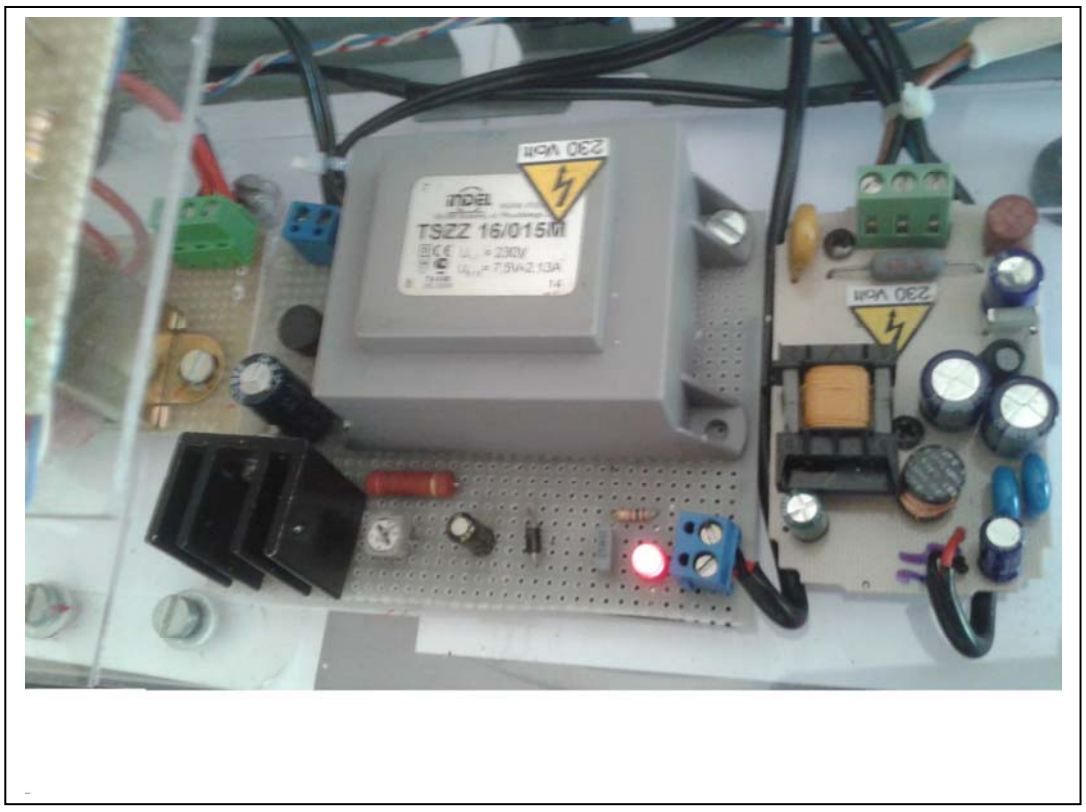
Particolare Stepper Motor



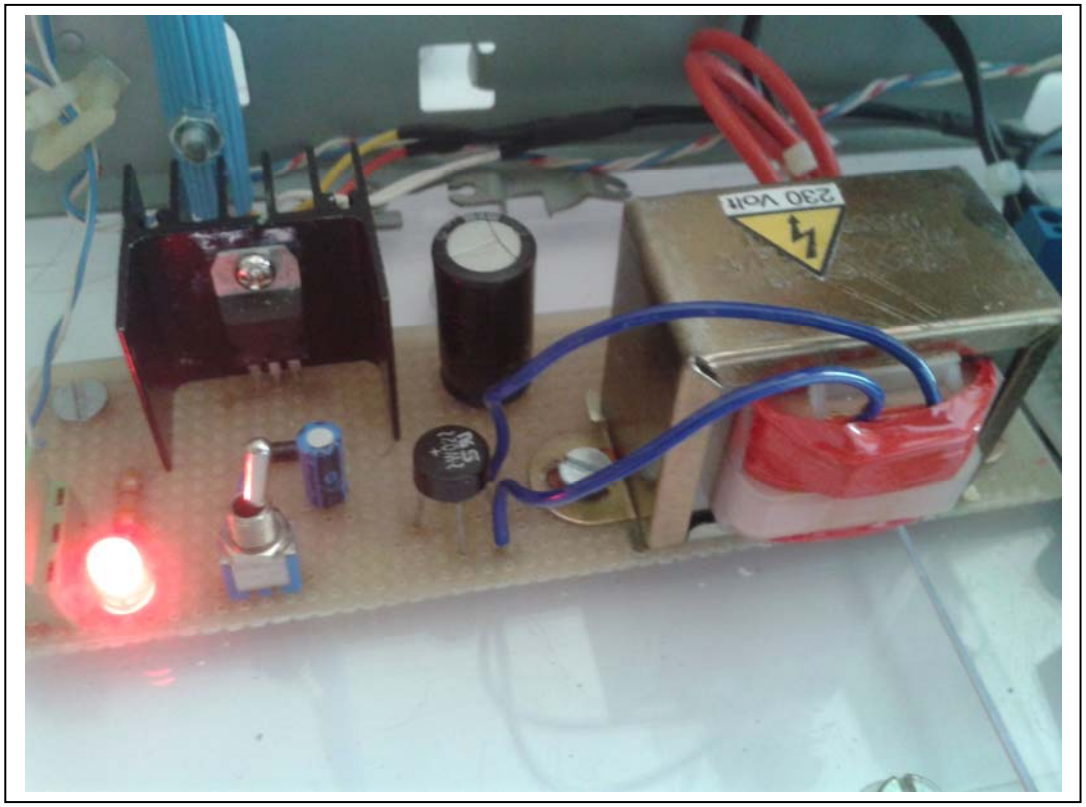
Particolare scheda PIC



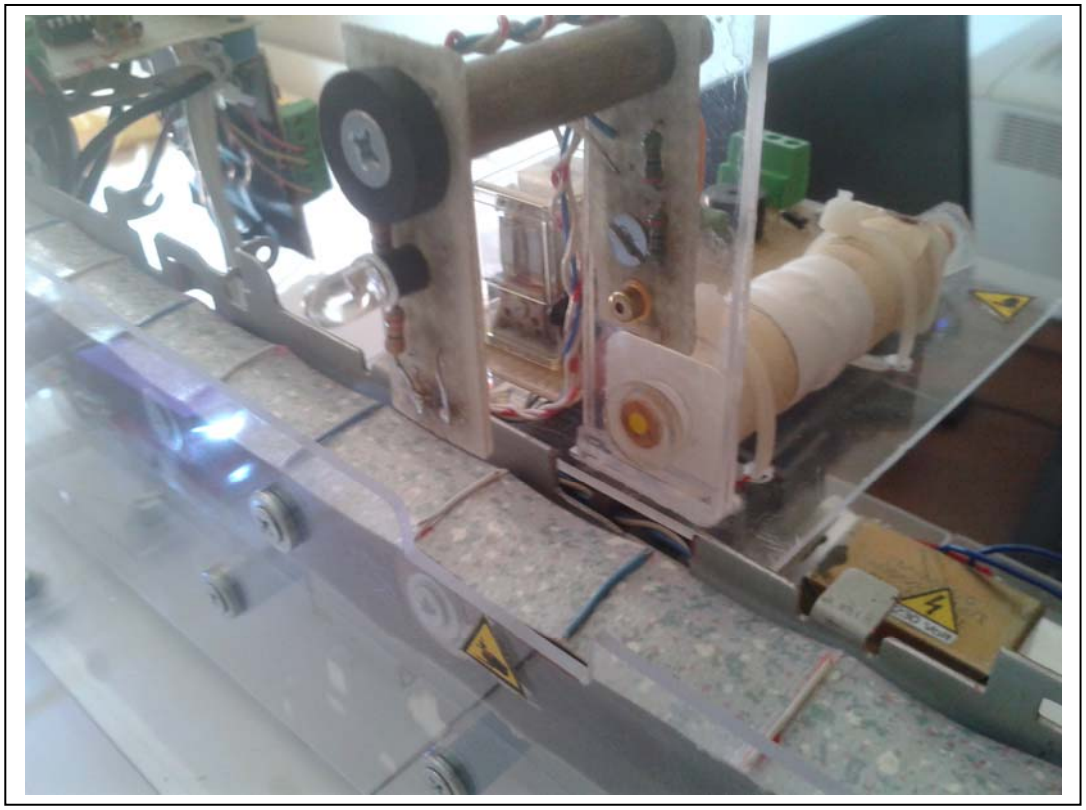
Scheda controllo motore passo passo



Alimentatore motore passo passo e alimentatore scheda PIC



Alimentatore scheda controllo pistone elettromagnetico



Particolare del sensore IR e del pistone elettromagnetico

Particolare del sensore IR a riflessione

Particolare del display