

Capitolo 4

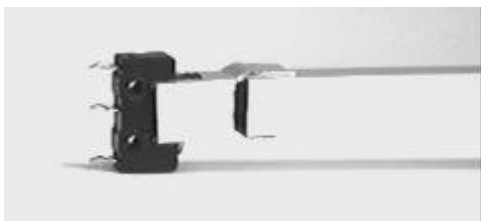
Realizzazione del sistema di controllo

4.1 Sensori

I sensori utilizzati sono di varia natura:

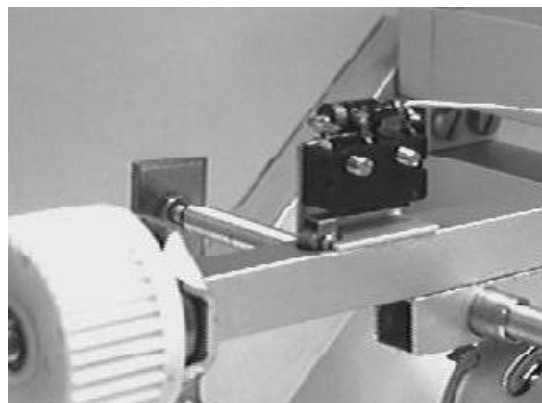
- elettronici
- elettromeccanici

I sensori elettromeccanici sono i più lenti tra i due, quindi sono usati per le rilevazioni lente. Sono stati utilizzati al nostro scopo nel segnalare i fine corsa o l'appoggio al terreno.

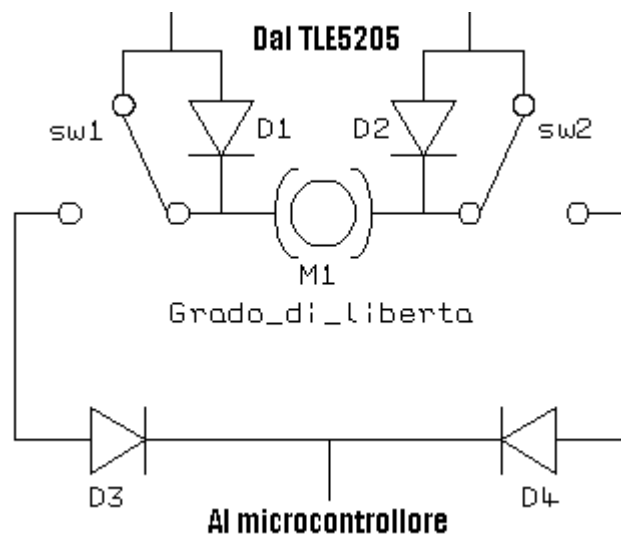


In queste foto si riesce a capire dove sono stati posti i sensori su menzionati, nella foto a sinistra è stato usato un sensore a levetta mentre a destra è stato usato uno a rotella.

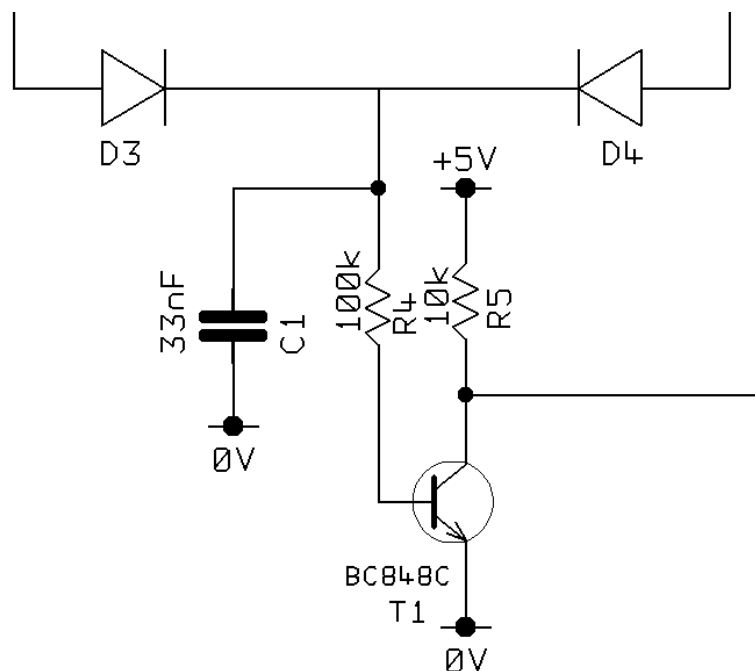
Il loro funzionamento consiste nello spostare il collegamento tra due lamelle ad una terza, scollegando una delle prime due; è stato sfruttato questo principio di funzionamento in unione al generatore PWM e due diodi (D1 D2),



(in figura si vedono i diodi D1 e D3 che sono saldati direttamente sui terminali dello switch per ridurre il n° di fili da cablare) per assicurare l'inversione della polarizzazione al motore, una volta raggiunto il termine fisico del movimento.



In questo modo anche se dovesse subentrare un errore, conseguentemente a un mal funzionamento del microcontrollore, una volta raggiunto la fine corsa, il diodo consentirebbe esclusivamente il passaggio di una corrente inversa, opposta alla precedente, quindi il moto sarà decelerato fino a fermarsi e costretto a cambiare verso, tutto questo se non prima, un comando proveniente dal micro impone lo spegnimento del PWM e corrispondentemente l'alimentazione, lasciando proseguire il moto per inerzia. L'operazione precedente rallenta il moto, evitando di far urtare con violenza l'estremità della corsa. A questo punto bisogna sfruttare lo stesso scatto dello switch per segnalare al microcontrollore che si è raggiunto la fine del movimento, ma non si può fare entrare nella porta del micro un onda quadra con ampiezza molto più ampia dell'alimentazione e in più per segnalare un evento costante, quindi bisogna filtrare il segnale.



Questa porzione di circuito mi permette di filtrare il segnale ad onda quadra proveniente dai diodi D3 D4, facenti parte del precedente circuito dei fine corsa.

Il condensatore C1, quando uno switch è premuto, si carica con la corrente proveniente dal corrispondente diodo (D3 o D4), di una tensione al massimo pari all'alimentazione (12V). Perché i picchi induttivi positivi sono tagliati sia dai diodi di protezione interni ai TLE5205 (i quali consentono un innalzamento fino a 12,7V) e ancor prima dai diodi D1 o D2 polarizzati inversamente. A questa tensione bisogna considerare la caduta sui diodi D3 o D4 di 0,7V. Per concludere, il range di tensione ai capi di C1 è compreso tra i 10V o 12V, in seguito alla tolleranza dei transistor di potenza interni ai TLE5205 e alla corrente che in essi circola.

Considerando costante V_{be} , C1 e R4 si comportano come una rete RC con costante di tempo

$$\tau = RC = 100K\Omega * 33nF = 3,3ms$$

la tensione V_c del condensatore si scarica su R4 con legge

$$V_c - V_{be} = (V_{co} - V_{be}) e^{-t/\tau}$$

La corrente che circola in base è uguale a quella in R4

$$I_{b(t)} = (V_{co} - V_{be}) e^{-t/\tau} / R_4$$

La corrente massima che può circolare nel collettore è limitata dalla resistenza R5 $\rightarrow I_{c_{max}} = 5V / 10K = 0,5mA$ da questo valore si può

ricavare il valore a cui deve scendere I_b affinché il transistor esca dalla saturazione. Usando i valori di guadagno h_{FE} $420 \div 800$ si ricava

$$I_b = 0,625\mu A \div 1,19\mu A$$

Ricavando il tempo (in cui impiega ad uscire dalla saturazione) dalla formula di $I_{b(t)}$ si ottiene la relazione

$$t = \tau \ln [(V_{co} - V_{be}) / (I_{b(t)} * R_4)]$$

imponendo la I_b risulta

$$t_{(0,625\mu A)} = 17ms$$

$$t_{(1,19\mu A)} = 15ms$$

da questo momento in poi il transistor entra in conduzione seguendo la relazione $I_c = I_b * h_{FE}$

$$V_{out(t)} = 5V - I_c * R_5 = 5V - [(V_{co} - V_{be}) * h_{FE} * R_5 / R_4] * e^{-t/\tau}$$

I valori TTL che a noi interessano sono 0.8V il livello basso, 2V il livello alto. Il tempo in cui il dispositivo è a questi valori lo ricaviamo dalla precedente formula.

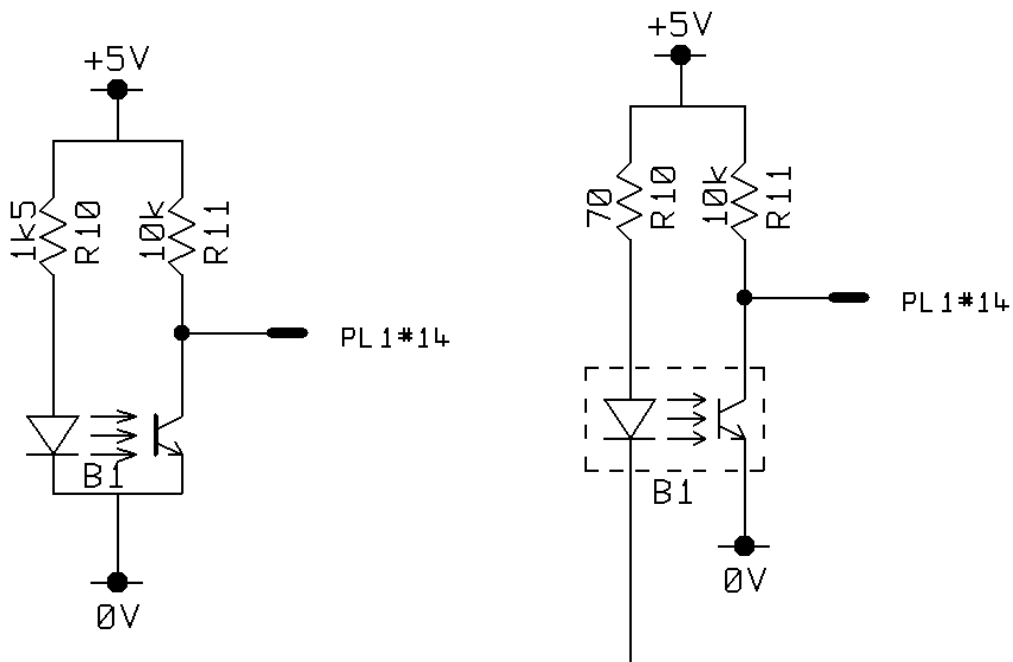
$$t = \tau \ln \{ [(V_{co} - V_{be}) * h_{FE} * R_5 / R_4] / (5V - V_{out(t)}) \}$$

imponendo i due precedenti valori di tensione di uscita, per i due possibili valori estremi di h_{FE} si ricava:

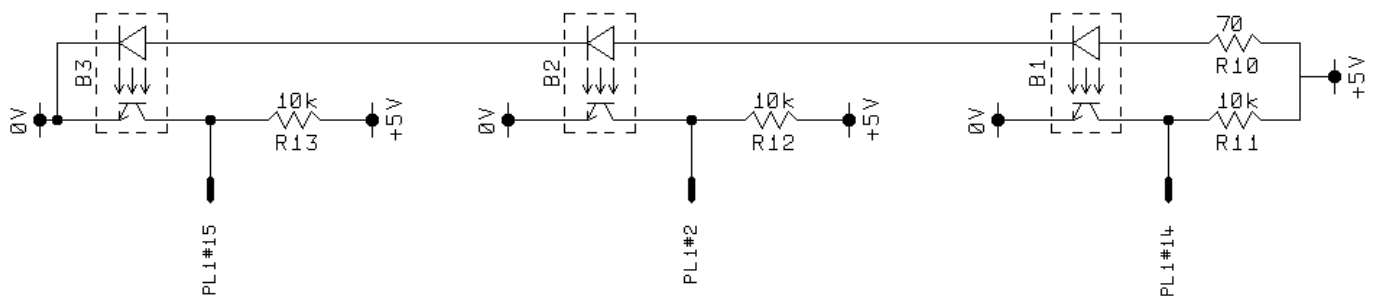
	$h_{FE} = 420$	$H_{FE} = 800$
$V_{out(t)} = 0.8V$	$t = 15,6ms$	$t = 17,7ms$
$V_{out(t)} = 2V$	$t = 16,7ms$	$t = 18,8ms$

Questi tempi sono di gran lunga superiori del periodo di onda quadra generato dal PWM, quindi tra due onde non c'è il rischio di commutazione. Invece riferendo i tempi al ciclo meccanico, sono molto minori del tempo che si impiega a raggiungere il centro del movimento, circa mezzo secondo, dopo il quale il software considererebbe un valore basso come il raggiungimento del fine corsa.

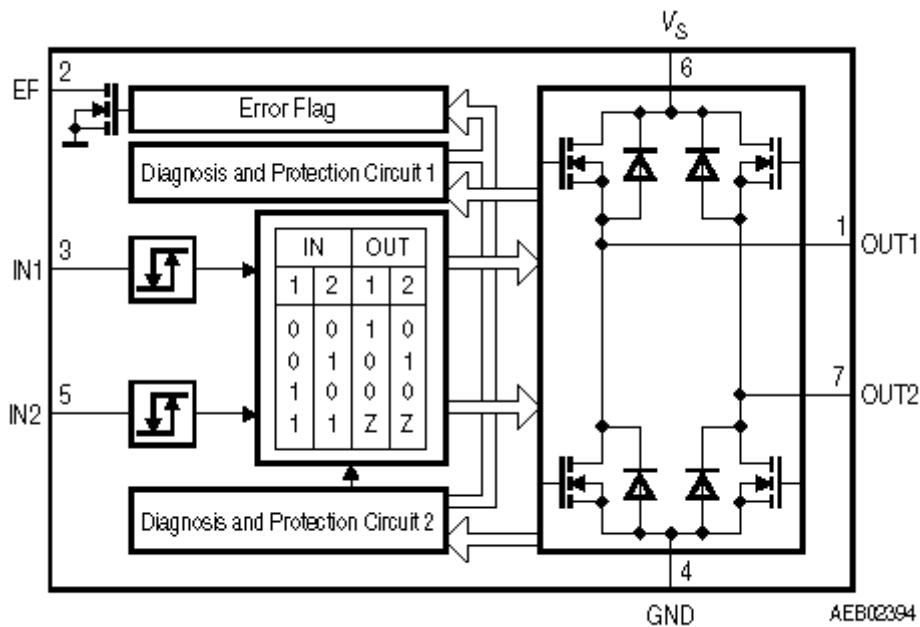
Un altro sensore utilizzato è quello che si occupa del conteggio del numero di giri della barra filettata o della bussola filettata. La velocità richiesta è superiore alle precedenti, all'incirca dell'ordine della decina di milli secondi. A seguito di questa maggiore velocità mi sono orientato a sensori ottici e ho scelto quelli a infrarosso. Inizialmente avevo pensato a un circuito di polarizzazione come qui sotto a sinistra:



Dopo prove di laboratorio, ho notato che la corrente che passava attraverso il diodo, non dava buoni risultati. In quanto la luce riflessa che eccitava il transistor non portava in sufficiente conduzione quest'ultimo, in modo da avere una tensione di uscita su PL1#14 sotto i 0.8V. Per aumentare la conduzione del transistor ho dovuto aumentare la corrente di polarizzazione del diodo a infrarossi, fino a 20mA. Questo valore per 18 elementi avrebbero consumato 360mA, non indifferente nel conto totale dell'autonomia della batteria. Per migliorare la situazione ho pensato di mettere in serie i tre diodi relativi alla stessa zampa, riducendo di $2/3$ il consumo di questi ultimi, secondariamente ma non meno importante, riducendo il n° di fili che dovevano scorrere sulle zampe quindi facilitando il cablaggio.



4.2 Stadio pilota motori



Lo stadio che pilota i motori è realizzato con un componente prodotto dalla Siemens chiamato TLE5205, realizzato con un processo in multi tecnologia “SPT” che permette la coesistenza sulla stessa struttura monolitica: tecnologia di controllo bipolare e CMOS, dispositivo di potenza DMOS.

4.2.1 Caratteristiche

- Eroga più di 5A in continua e 6A di picco
- Ottimizzato per applicazioni di controllo motori corrente continua
- Opera con alimentazione fino a 40V
- Bassa $R_{DS\ ON}$; tipica 200m Ω a 25°C per switch
- Uscite protette contro tutti i cortocircuiti
- Protezione surriscaldamento con isteresi e diagnosi
- Diagnosi cortocircuiti e carico aperto, con flag errore open drain
- Spegnimento con isteresi se sottoalimentato
- Ingresso con isteresi compatibile CMOS/TTL
- Assente corrente di crossover
- Presenti diodi di protezione per carichi induttivi
- Ampio range di temperatura; $-40^{\circ}\text{C} < T_j < 150^{\circ}\text{C}$

Pin	Simboli	Funzioni
1	Out 1	Uscita del canale 1: Protezione contro i cortocircuiti; diodi integrati per carichi induttivi

2	EF	Flag errore: Uscita TTL/CMOS compatibile, per la determinazione di errori; (open drain)
3	In 1	Ingresso di controllo 1: TTL/CMOS compatibile
4	GND	Massa: Internamente connesso alla linguetta
5	In2	Ingresso di controllo 2: TTL/CMOS compatibile
6	V_s	Alimentazione: Rispetto massa
7	Out2	Uscita del canale 2; Protezione contro i cortocircuiti; diodi integrati per carichi induttivi

4.2.2 Descrizione circuito

1. Circuito d'ingresso: è un Trigger di Schmitt con isteresi, compatibile TTL/CMOS, seguito dal controllo dello stadio d'uscita.
2. Stadio d'uscita: è un H-bridge DMOS. Il circuito integrato protegge l'uscita contro cortocircuiti a massa o verso alimentazione, e anche per picchi di tensione Positivi e negativi, che si verificano quando il carico induttivo sta cambiando, questi picchi sono limitati dai diodi integrati di protezione. In più un circuito monitore verifica l'attivazione di ogni transistor d'uscita, se attivo impedisce la

conduzione del corrispondente transistor sorgente, facendo cedere l'operazione al transistor complementare, quindi non può esserci corrente di crossover.

3. Funzione monitoraggio alimentazione: quando V_S raggiunge $V_{S\ ON}$ il circuito integrato diviene attivo con un isteresi. Tutti i transistor di uscita saranno spenti se l'alimentazione diminuisce al di sotto del valore $V_{S\ OFF}$.
4. Funzione di protezione: diversi errori come cortocircuiti a $+V_S$, massa o attraverso il carico sono rilevati. Tutti gli errori spengono gli stadi di uscita come risultato, con un ritardo di $50\mu s$ e mettendo a massa EF (= flag errore). Cambiando l'ingresso si resetta il flag errore.
 - A) Rilevato cortocircuito a massa; se transistor in alto è acceso e l'uscita è a massa, la corrente in uscita è limitata internamente. Dopo un ritardo di $50\mu s$ tutte le uscite saranno spente e il flag errore sarà settato.
 - B) Rilevato cortocircuito a $+V_S$; se transistor in basso è acceso e l'uscita è a $+V_S$, la corrente in uscita è limitata internamente. Dopo un ritardo di $50\mu s$ tutte le uscite saranno spente e il flag errore sarà settato.
 - C) Rilevato sovraccarico; se la corrente attraverso i transistor del lato basso, eccedono il punto di escursione I_{SDL} , tutte le uscite saranno spente e il flag errore sarà settato.

- D) Rilevato innalzamento temperatura; a una temperatura di giunzione più alta di 150°C l'interruttore termico spegne tutte quattro le uscite e setta il flag errore con un ritardo.
- E) Rilevato carico aperto; l'OUT1 ha una resistenza da $10\text{K}\Omega$ pull-up mentre l'OUT2 ne ha una da $10\text{K}\Omega$ pull-down. Se i due ingressi sono alti, le uscite sono spente. Nel caso non ci sia carico tra le uscite, la tensione su OUT1 è $+V_S$ e OUT2 è a massa. Questo stato sarà rilevato da due comparatori e il flag errore sarà settato dopo un tempo di ritardo di $50\mu\text{s}$. Cambiando l'ingresso l'errore si resetta.

Maggiori dettagli si possono trovare in appendice, dove sono presenti

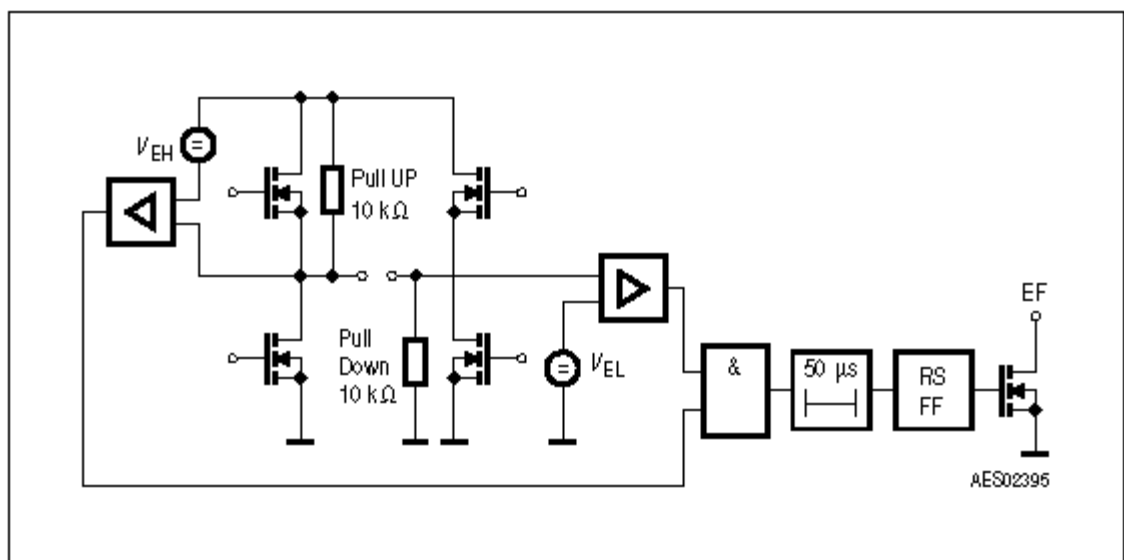
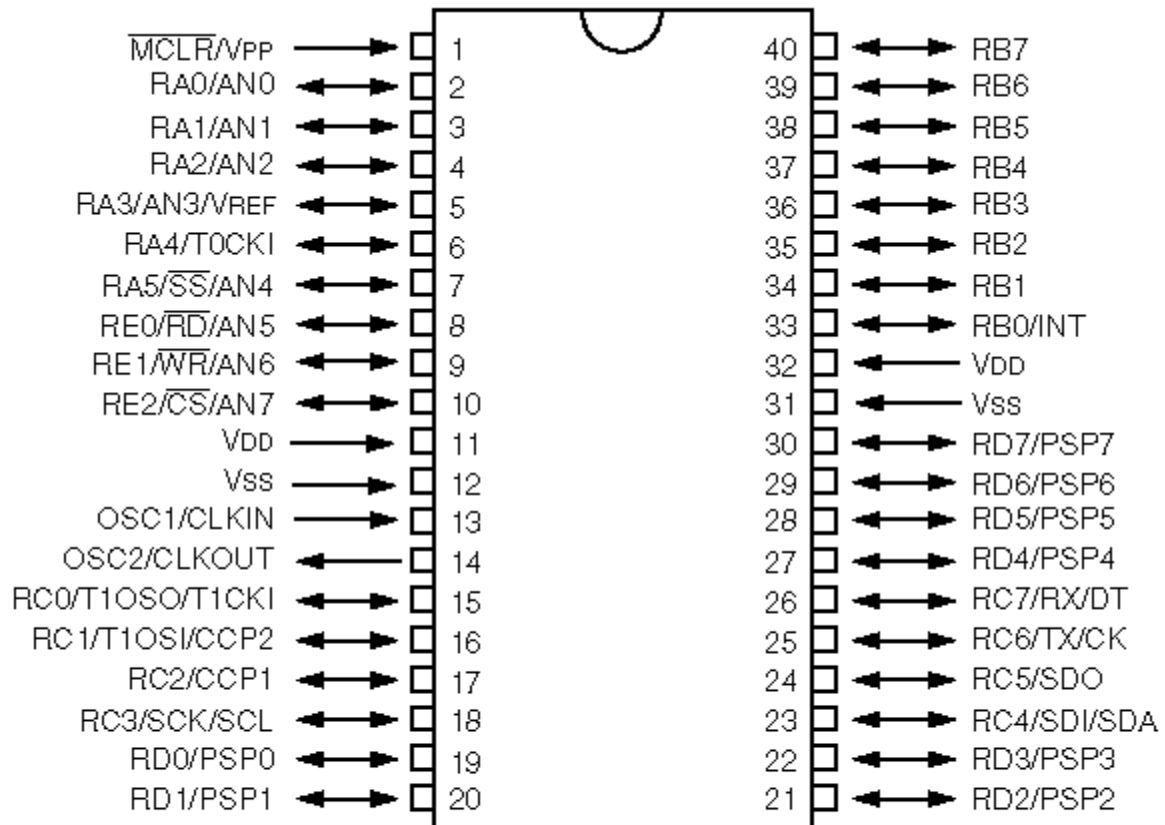


tabelle con dati e grafici con tempi di risposta.

4.3 Microprocessori

Sull'Esapode ci sono due microcontrollori, uno si occupa del movimento di tre zampe (anteriore e posteriore sullo stesso lato, in più il centrale del fianco opposto) mentre l'altro si occupa del movimento delle tre zampe complementari al primo. Questo microprocessore è il PIC16C74A. I PIC sono una famiglia di circuiti integrati prodotti dalla Microchip Technology Inc, che appartengono alla categoria dei microcontroller, ovvero quei componenti che integrano in un unico dispositivo, tutti i circuiti necessari a realizzare un completo sistema digitale programmabile.

In figura sono mostrate le funzioni dei piedini.



PIC16C74 PIC16C74A

i PIC (in questo caso un PIC16C74A) si presentano esternamente come dei normali circuiti integrati TTL o CMOS, ma internamente dispongono di tutti dispositivi tipici di un sistema a microprocessore, ovvero:

- Una CPU (**C**entral **P**rocessor **U**nit ovvero unità centrale di elaborazione) il cui scopo è interpretare le istruzioni di programma.
- Una memoria EPROM (**E**rasable **P**rogrammable **R**ead **O**nly **M**emory ovvero memoria programmabile a sola lettura), in cui sono memorizzate in maniera permanente le istruzioni del programma da eseguire.
- Una memoria RAM (**R**andom **A**ccess **M**emory ovvero memoria ad accesso casuale) utilizzata per memorizzare le variabili utilizzate dal programma.
- Una serie di LINEE DI I/O per pilotare dispositivi esterni o ricevere impulsi da sensori, pulsanti, ecc.
- Una serie di dispositivi ausiliari al funzionamento, quali generatori di clock, bus, contatori, ecc.

La presenza di tutti questi dispositivi in uno spazio veramente contenuto, consente al progettista di avvalersi degli enormi vantaggi derivanti dall'uso di un sistema a microprocessore, anche in quei circuiti, che fino a poco tempo fa erano destinati ad essere realizzati con circuiterie tradizionali.

Per il progetto ho scelto il PIC16C74A. Esso è dotato di 40 pin, di cui ben 33 disponibili per l'I/O, ovvero per i collegamenti al resto del circuito e di alcune caratteristiche che lo rendono maggiormente adatto alle esigenze del progetto.

4.3.1 Caratteristiche del nucleo nel microcontrollore PIC16C74.

- CPU Risc ad alte performance.
- Solo 35 istruzioni a singole parole da imparare.
- Tutte le istruzioni sono a singolo ciclo, ad eccezione dei salti che ne hanno due.
- Velocità operativa: 20MHz clock d'ingresso, 200ns ciclo d'istruzione.
- Memoria programma 4K x 14bit a parola (EPROM).
- Memoria dati 192 Byte (RAM).
- Capacità d'interrupt.
- Profondità stack otto livelli hardware.
- Modo d'indirizzamento diretto, indiretto e relativo modo d'indirizzamento.
- Reset d'accensione (POR).
- Temporizzatore d'accensione (PWRT).
- Temporizzatore di partenza oscillatore (OST).
- Temporizzatore Watchdog (WDT), con il suo oscillatore RC sul chip, per rendere l'operazione di supervisione affidabile.

- Protezione del codice, programmabile.
- Risparmio d'energia usando il modo SLEEP.
- Scelta dell'oscillatore.
- Tecnologia CMOS-EPROM, basso consumo, alta velocità.
- Progetto interamente statico:
 - tipico $15\mu\text{A}$ @ 3V, 32KHz.
 - tipico $<1\mu\text{A}$ corrente d'attesa.

4.3.2 Caratteristiche delle periferiche.

- Timer 0: temporizzatore 8-bit con 8-bit di prescaler.
- Timer 1: temporizzatore 16-bit con prescaler, può incrementare anche durante lo SLEEP, attraverso un clock esterno.
- Timer 2: temporizzatore 8-bit con registro periodo 8-bit, prescaler e postscaler.
- Moduli, CAPTURE, COMPARE e PWM.
 - CAPTURE 16-bit, risoluzione max. è 12.5ns
 - COMPARE 16-bit, risoluzione max. è 200ns
 - PWM risoluzione max. è 10-bit
- Convertitore analogico digitale multicanale a 8-bit.

- Porta seriale sincrona (SSP) con protocollo SPI e I²C.
- Ricevitore trasmettitore sincrono asincrono universale (USART/SCI).
- Porta parallela slave (PSP) 8-bit, con esterni RD (leggi), WR (scrivi) e CS (chip selezionato) di controllo.
- Circuito di rilevazione abbassamento alimentazione per Brown-out Reset (BOR).

4.3.3 Descrizione delle periferiche

Il registro TMR0 è un contatore, ovvero un particolare tipo di registro il cui contenuto viene incrementato con cadenza regolare e programmabile direttamente dall'hardware del PIC. In pratica, a differenza di altri registri, il TMR0 non mantiene inalterato il valore che gli viene memorizzato, ma lo incrementa continuamente, se ad esempio scriviamo in esso il valore 10 con le seguenti istruzioni:

```
movlw 10
```

```
movwf TMR0
```

dopo un tempo pari a quattro cicli macchina, il contenuto del registro comincia ad essere incrementato a 11, 12, 13 e così via con cadenza costante e del tutto indipendente dall'esecuzione del resto del programma.

Una volta raggiunto il valore 255 il registro TMR0 viene azzerato automaticamente riprendendo quindi il conteggio non dal valore originariamente impostato ma da zero.

La frequenza di conteggio è direttamente proporzionale alla frequenza di clock applicata al chip e può essere modificata programmando opportunamente alcuni bit di configurazione.

In più si può agire sul PRESCALER che consiste in un divisore programmabile a 8 bit da utilizzare nel caso la frequenza di conteggio inviata al contatore TMR0 sia troppo elevata per i nostri scopi.

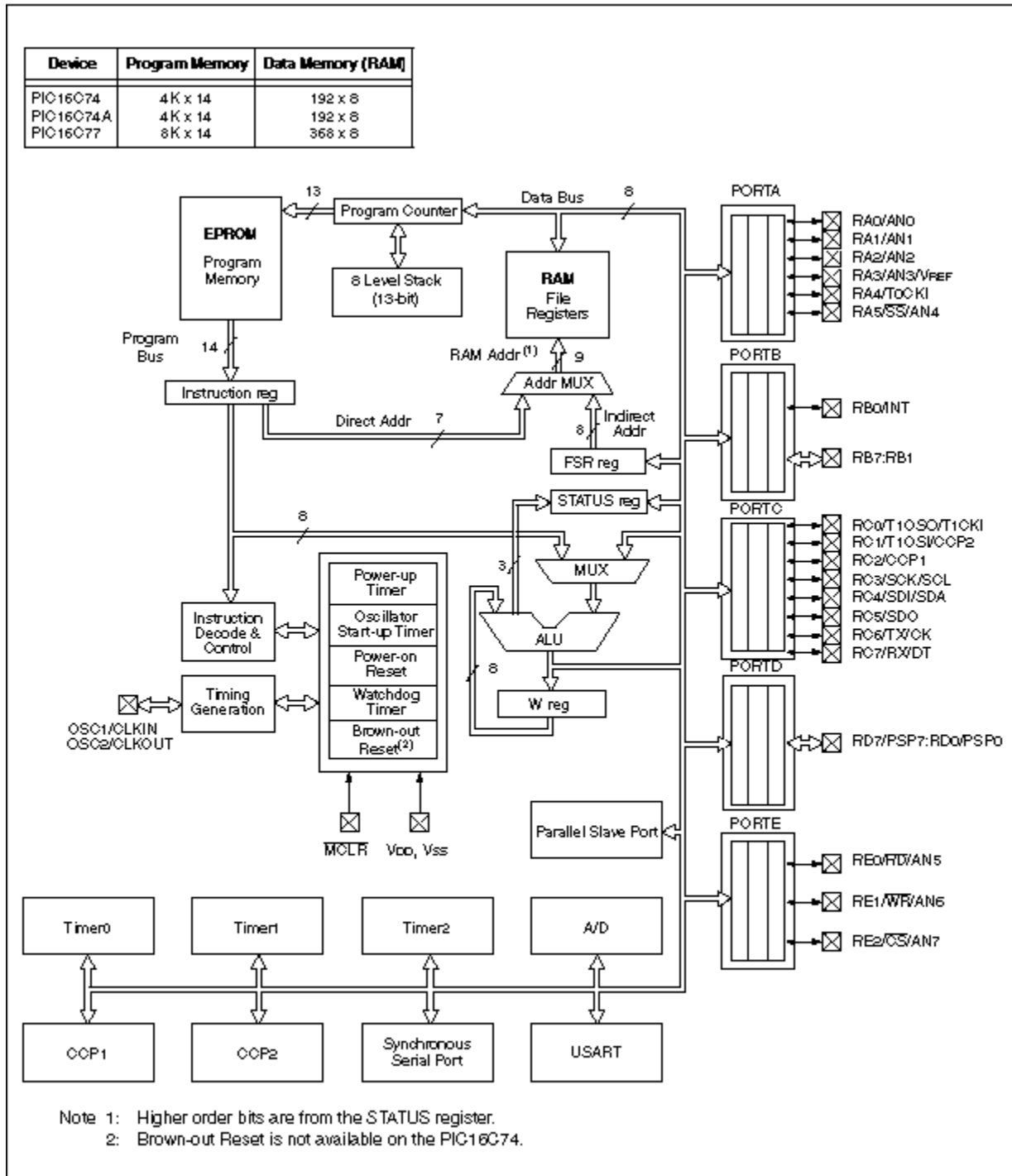
Con l'uso del PRESCALER possiamo dividere ulteriormente la frequenza $F_{osc}/4$ configurando opportunamente i bit **PS0**, **PS1** e **PS2** del registro **OPTION**.

L'interrupt è una particolare caratteristica dei microprocessori in generale che consente di intercettare un evento esterno, interrompere momentaneamente il programma in corso, eseguire una porzione di programma specializzata per la gestione dell'evento verificatosi e riprendere l'esecuzione del programma principale.

Volendo fare un paragone con il mondo reale possiamo dire che l'interrupt rappresenta per il PIC quello che per noi rappresenta ad esempio la suoneria del telefono.

EPROM - Erasable Programmable Read Only Memory (memoria a sola lettura programmabile, cancellabile) è una memoria PROM che può essere cancellata se esposta alla luce di una lampada ad ultravioletti. Per questo motivo i dispositivi dotati di EPROM hanno una finestra trasparente da cui è possibile far raggiungere il chip interno al dispositivo dai raggi UV.

Diagramma a blocchi



Descrizione dei Pin d'uscita.

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS ⁽⁴⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
RA0/AN0	2	3	19	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 RA3 can also be analog input3 or analog reference voltage RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2	4	5	21	I/O	TTL	
RA3/AN3/VREF	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. Interrupt on change pin. Interrupt on change pin. Interrupt on change pin. Serial programming clock. Interrupt on change pin. Serial programming data.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6	39	43	16	I/O	TTL/ST ⁽²⁾	
RB7	40	44	17	I/O	TTL/ST ⁽²⁾	

Legend: I = input O = output I/O = Input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
 Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.
 Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
 Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Descrizione dei Pin d'uscita.

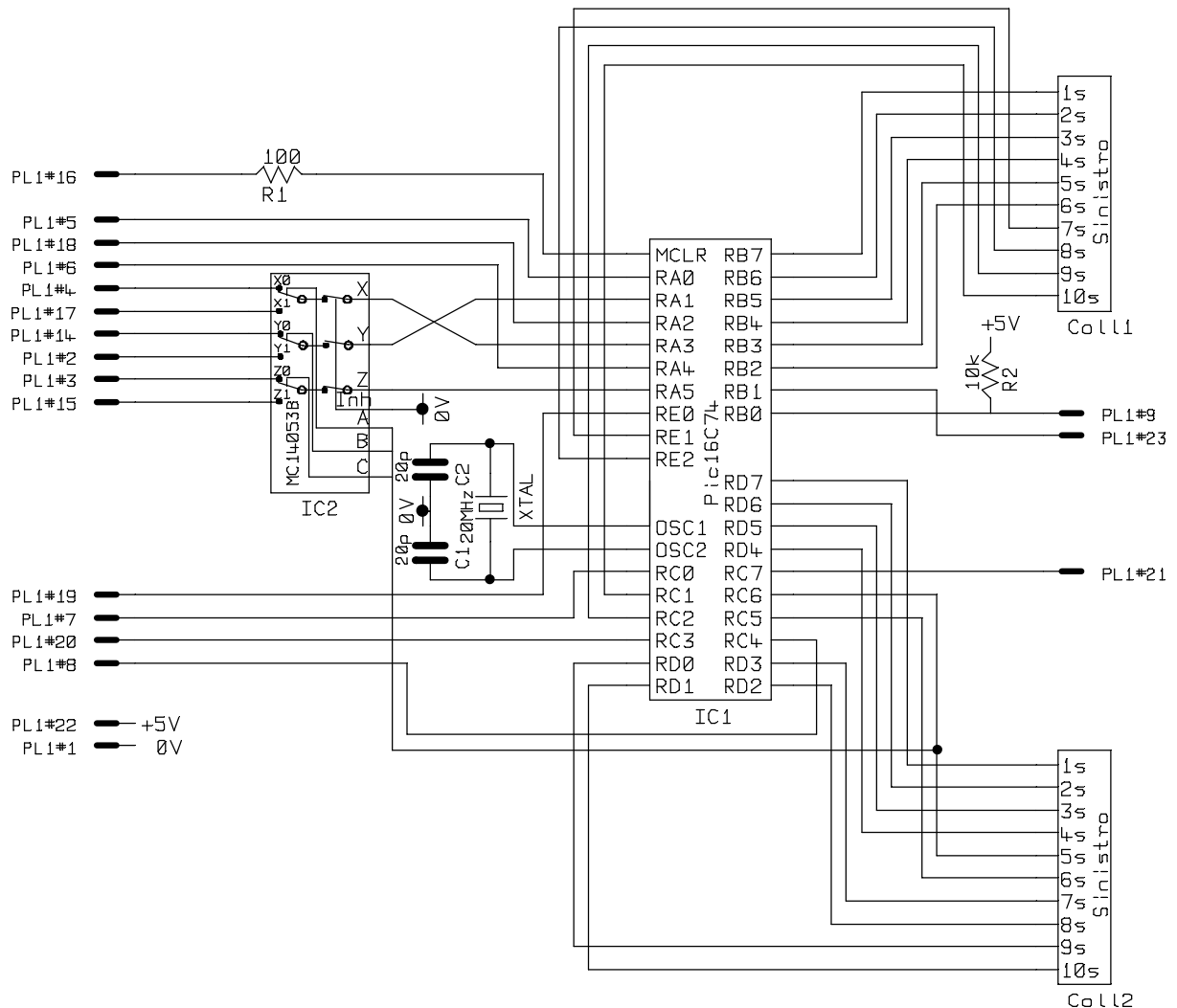
Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RD0/PSP0	19	21	38	I/O	ST/TTL ⁽³⁾	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	23	40	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	24	41	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	30	2	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	31	3	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	32	4	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	33	5	I/O	ST/TTL ⁽³⁾	
RE0/RD/AN5	8	9	25	I/O	ST/TTL ⁽³⁾	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5.
RE1/WF/AN6	9	10	26	I/O	ST/TTL ⁽³⁾	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/CS/AN7	10	11	27	I/O	ST/TTL ⁽³⁾	RE2 can also be select control for the parallel slave port, or analog input7.
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
Vdd	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
 2: This buffer is a Schmitt Trigger input when used in serial programming mode.
 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

4.4 Schemi

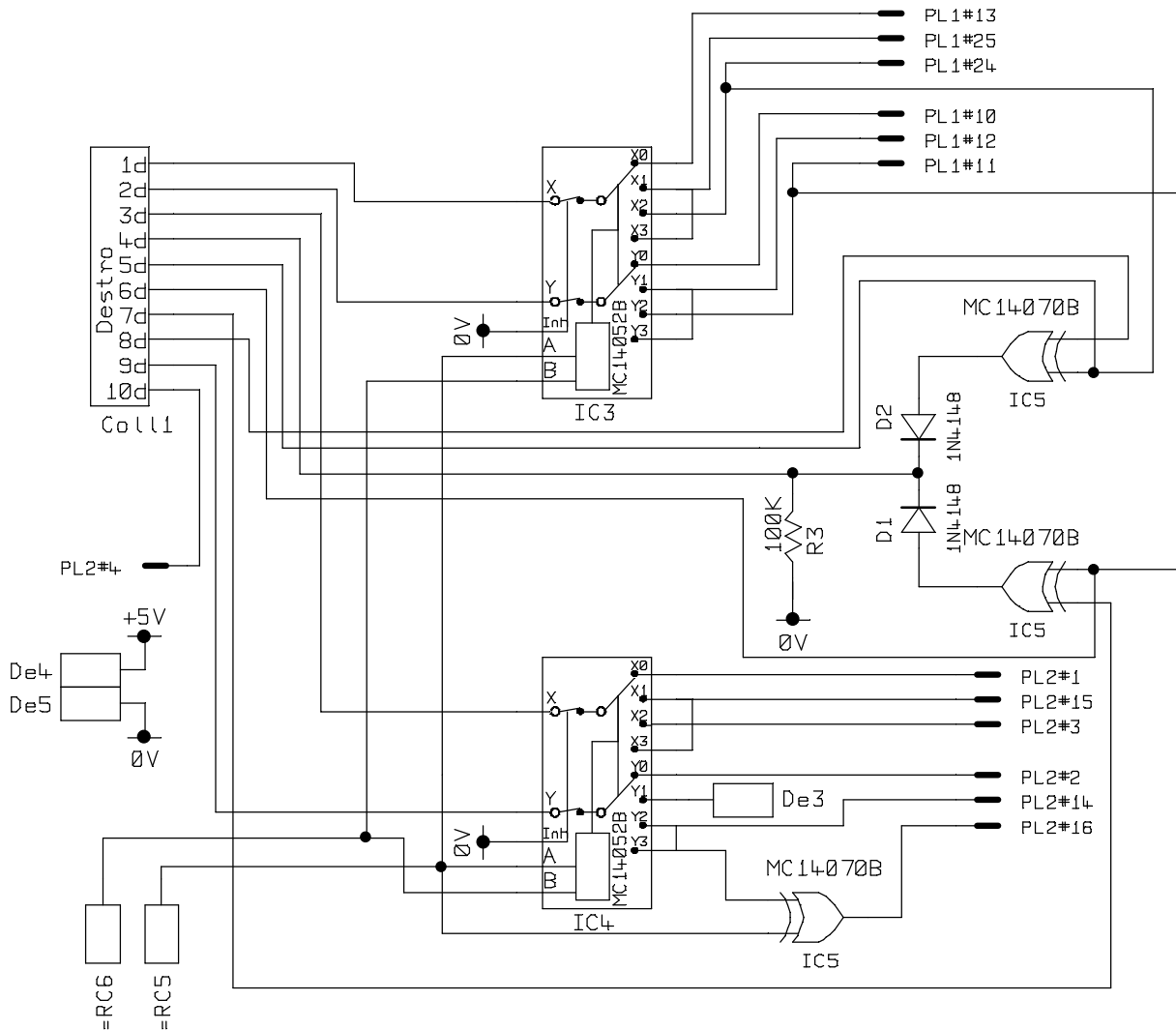
Si inizia con lo schema elettrico delle connessioni al microcontrollore, per come le varie porte sono collegate esternamente.



I connettori virtuali “Coll1” e “Coll2” sono stati utilizzati per modulare in tre parti il circuito, in modo da consentire una rappresentazione più grande.

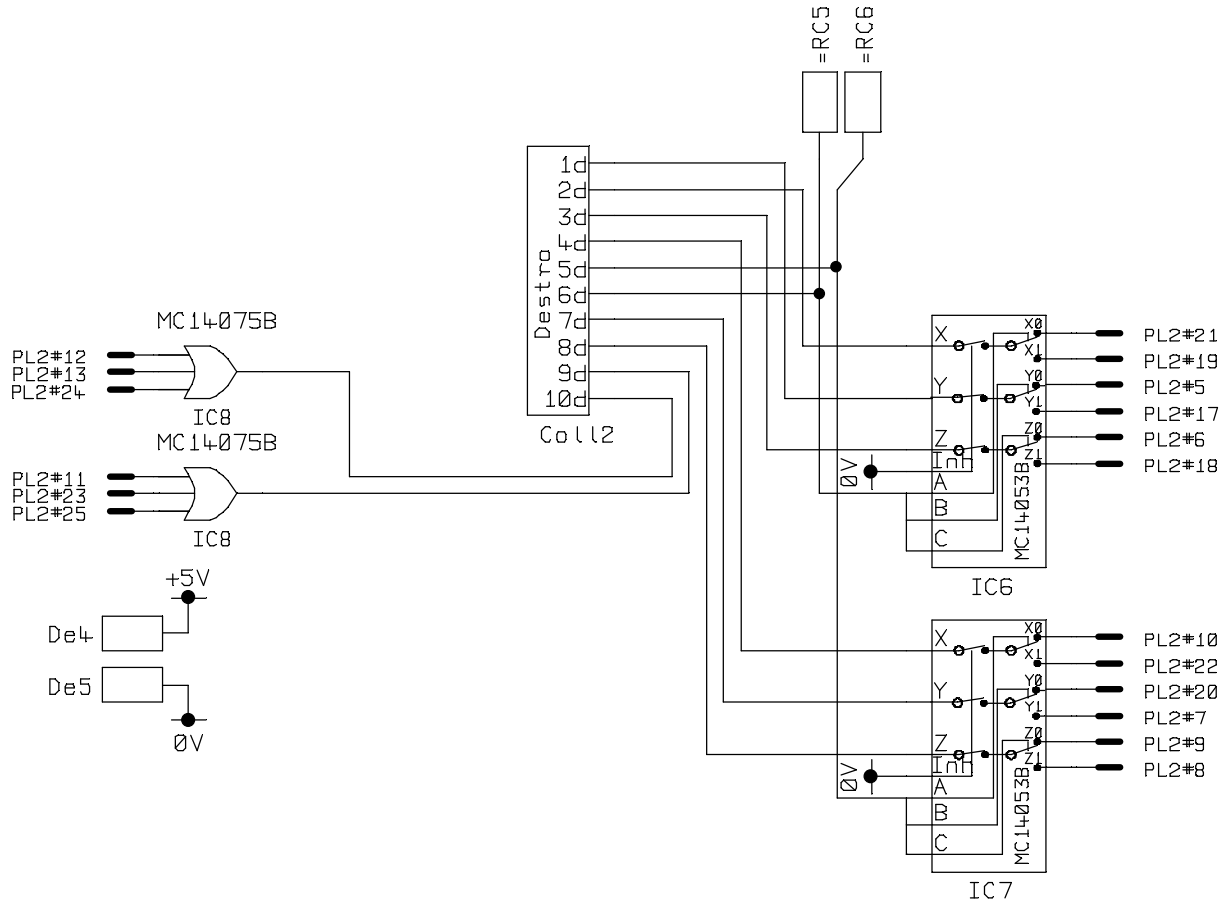
Nota: tutti i disegni sono stati ricavati utilizzando “Vutrax”, versione free scaricata da www.elettroshop.com

Il successivo schema è collegato alla parte alta del precedente (sinistro collegato con destro), con in più altri collegamenti virtuali, sempre per semplificare la rappresentazione.

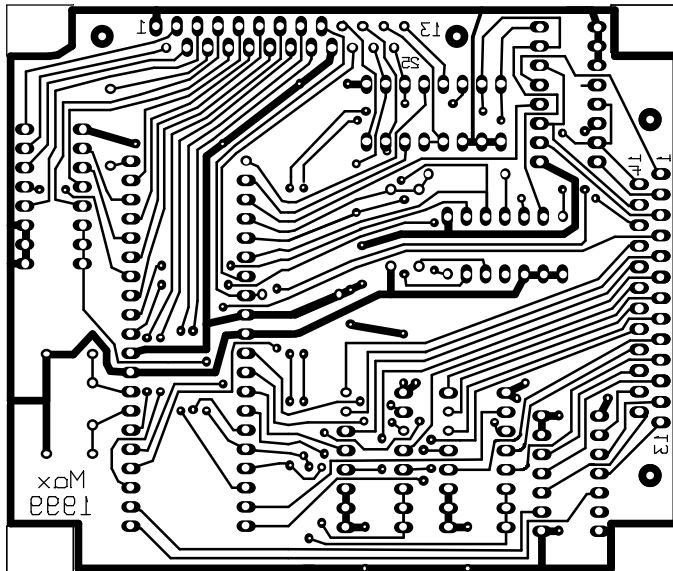


I nomi “PL1#**” e “PL2#**” sono relativi ai due connettori Canon 25 poli.

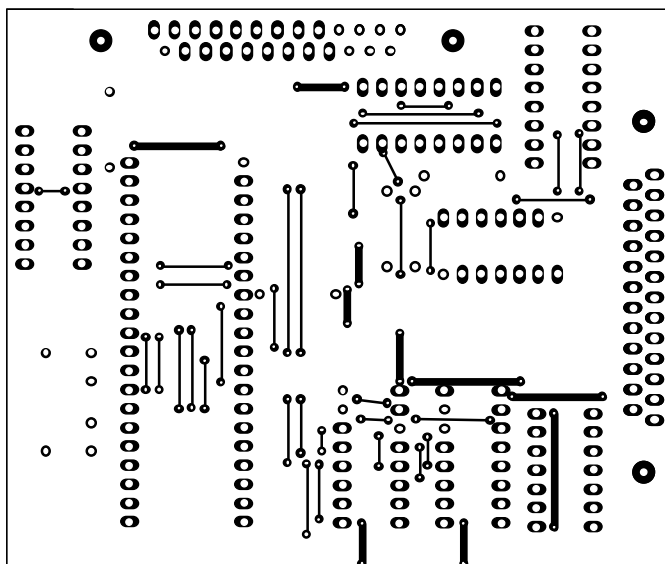
Il successivo schema è collegato alla parte bassa del primo (sinistro collegato con destro), con in più altri collegamenti virtuali collegati con il secondo, sempre per semplificare la rappresentazione.



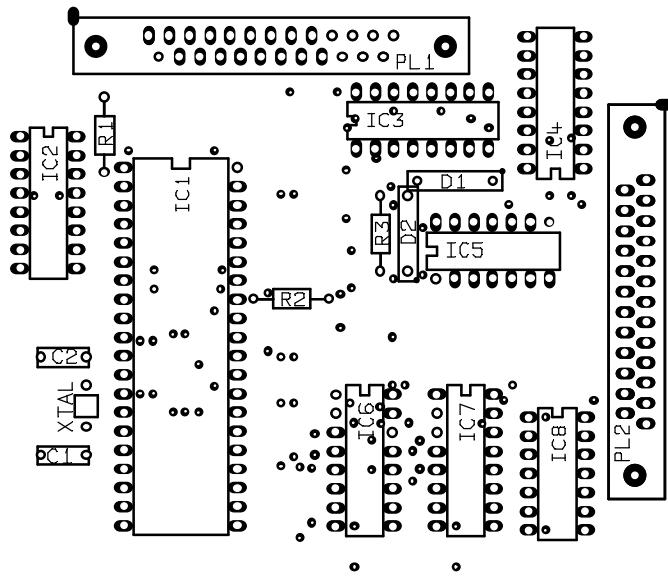
L'insieme dei tre schemi precedenti sono stati collegati su un unica basetta per realizzare il master qui sotto, la vista è il lato componenti.



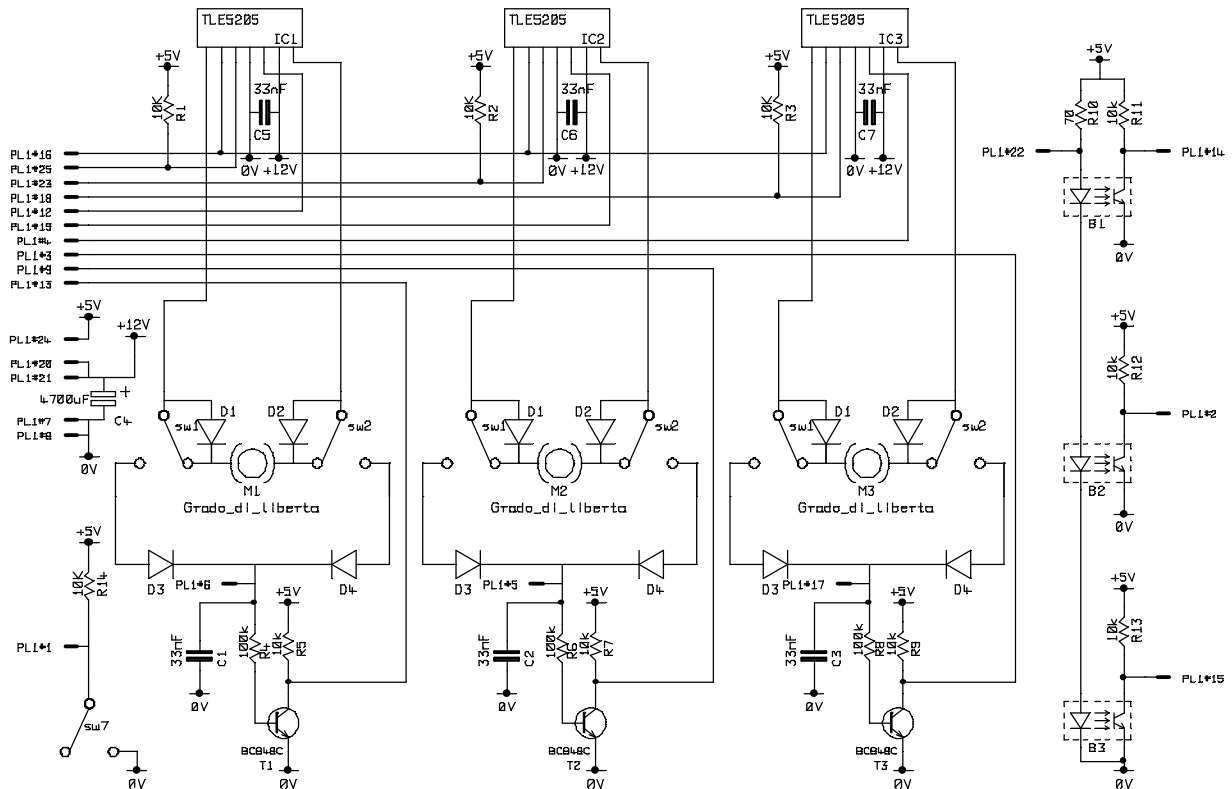
Qui sotto si possono vedere i ponticelli che sono stati necessari per stampare il master in singola faccia.



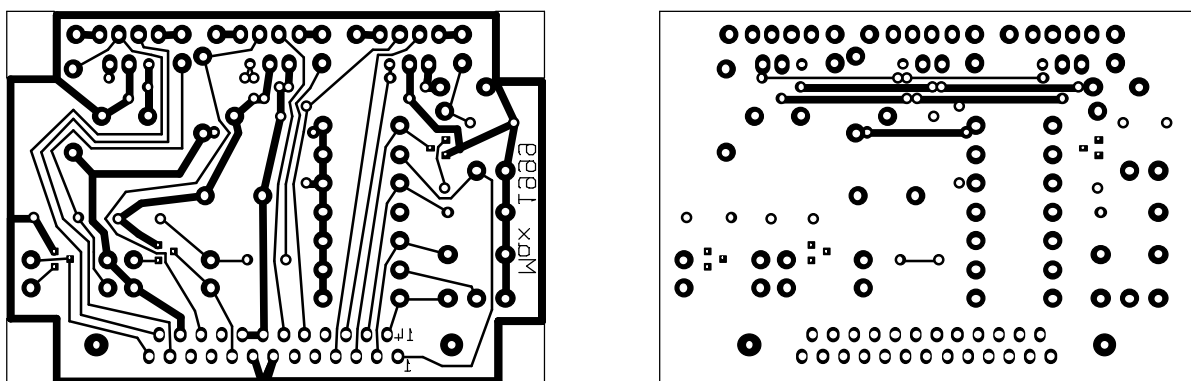
In questo disegno si può vedere il lato componenti e la loro disposizione.



Dopo la scheda per il microcontrollore si è realizzata la scheda per i drivers di potenza. Lo schema elettrico si può vedere qui sotto.



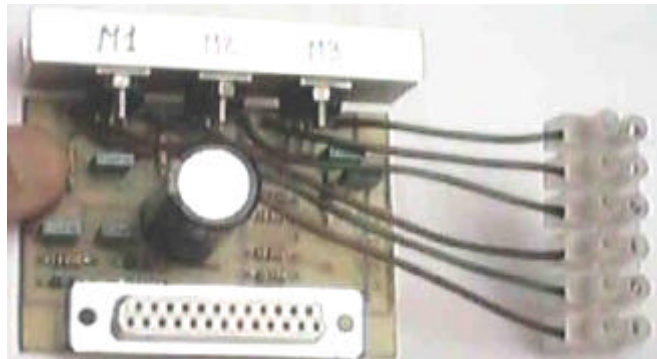
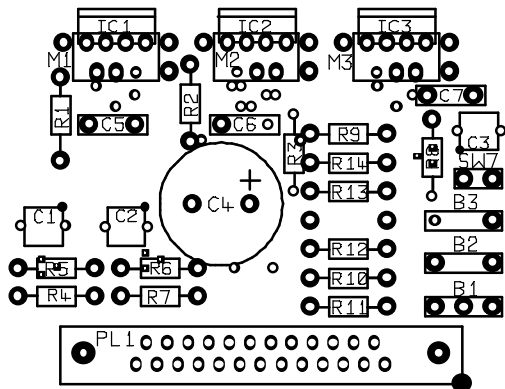
Con questo schema elettrico è stato realizzato il seguente master



Si osservi che i transistor usati sono del tipo surface mount quindi non si sono praticati dei fori e le saldature sono state eseguite sulle piccole piazzuole quadrate stagnando per prima la piazzuola successivamente è

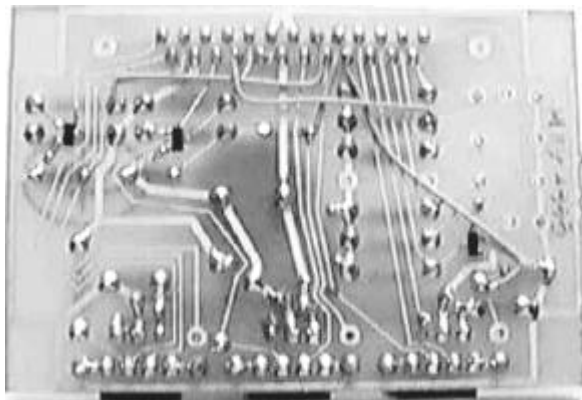
stato appoggiato il transistor al quale con una piccola punta di saldatore è stato scaldato il terminale.

Di seguito si può vedere la disposizione dei componenti nel lato superiore.



Dopo la realizzazione sono state apportate delle modifiche collegando i sensori attraverso PL1 (nello schema elettrico precedente si possono vedere in quali pin) tramite quattro ponticelli da aggiungere e le uscite di potenza attraverso una morsettiera a vite, per consentire il distacco della scheda senza bisogno di dissaldare.

Di seguito si mostra la foto relativa ai ponticelli lato rame.



Per finire si mostrano i collegamenti tra le schede e le zampe.

