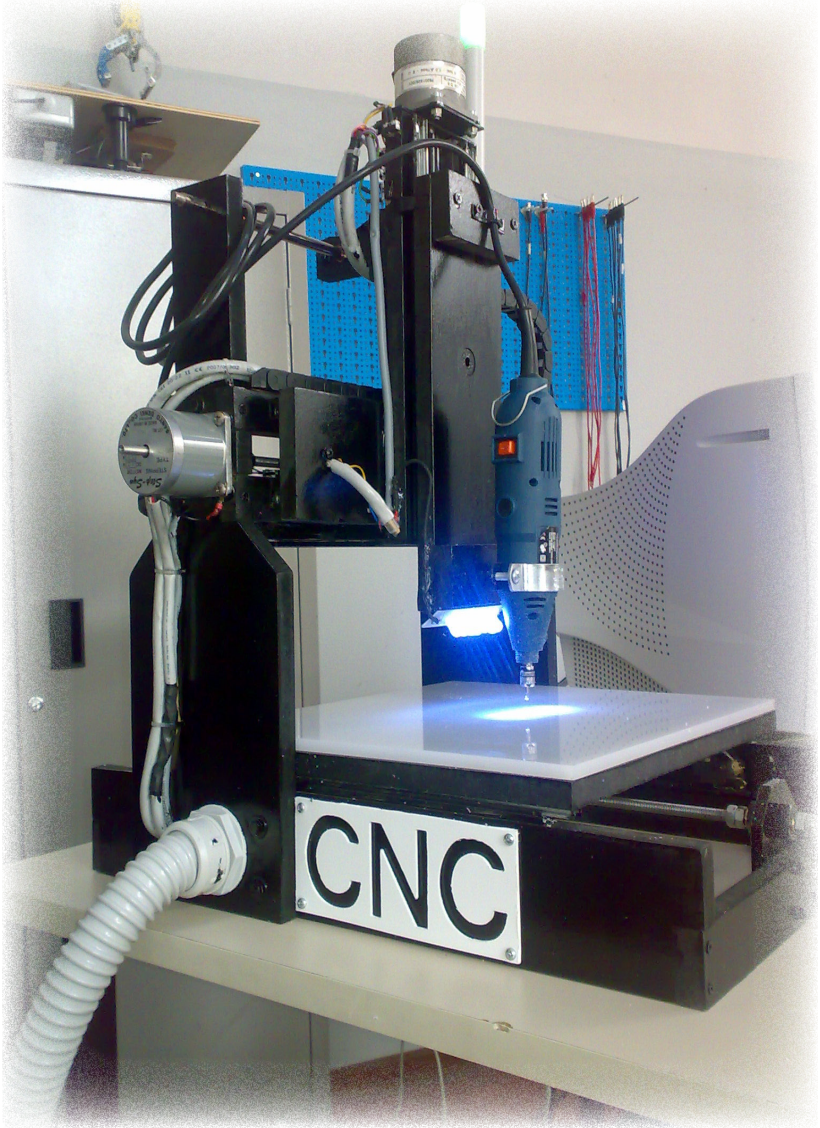
ISTITUTO TECNICO INDUSTRIALE STATALE "Primo Levi"

Via Matteotti - 30035 MIRANO - Venezia

Specializzazione: Elettronica e Telecomunicazioni

Ceccato Roberto - Nassuato Denis

5BET



CNC a 3 ASSI

⇨ Indice

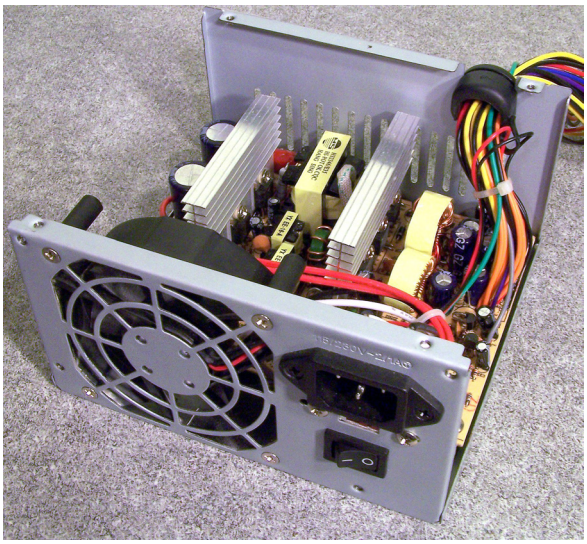
⇨ Switching o "a commutazione" - ATX

Possiedono circuiti più complessi rispetto ad un alimentatore tradizionale, ma hanno diversi vantaggi, tra cui un minore ingombro e peso a parità di potenza, un rendimento maggiore e quindi minore calore prodotto, sono meno adatti per l'uso in laboratorio, essendo caratterizzati: da un elevato ripple; e dalla generazione di componenti spurie ad alta frequenza che possono interferire nel funzionamento di alcune apparecchiature.

Il principio di funzionamento si basa sul fatto che un trasformatore è più efficiente, richiede un nucleo ferromagnetico più piccolo ed è molto più compatto, a parità di potenza, all'aumentare della frequenza operativa. Negli alimentatori elettronici vengono utilizzati particolari trasformatori fatti funzionare a frequenze di decine o centinaia di migliaia di Hertz invece dei 50 Hz della rete elettrica di distribuzione. Il nucleo di questo trasformatore è in Ferrite, materiale realizzato con polveri metalliche incollate, invece dei tradizionali lamierini di ferro, che alle alte frequenze comporterebbero una notevole perdita di energia.

In un alimentatore elettronico la tensione di rete viene per prima cosa raddrizzata e livellata con un condensatore. Successivamente un circuito oscillatore genera a partire da questa corrente continua, una corrente alternata di elevata frequenza, che viene applicata ai capi dell'avvolgimento primario del trasformatore; la tensione in uscita, presente ai capi dell'avvolgimento secondario del trasformatore, viene raddrizzata e livellata.

La funzione di stabilizzazione è solitamente ottenuta retroazionando l'errore del segnale in uscita sul regime di funzionamento dell'oscillatore. In pratica, un circuito misura la tensione di uscita, e se questa risulta troppo alta viene ridotta l'energia inviata dall'oscillatore al trasformatore, se invece la tensione scende, viene aumentato il flusso di energia. Grazie a questo sistema molti alimentatori switching sono in grado di accettare in ingresso un'ampia gamma di tensioni e frequenze. Per esempio gli alimentatori per Notebook spesso possono essere collegati sia alla rete europea a 230 V/50 Hz, sia a quella statunitense a 115 V/60 Hz.



Alimentatore elettronico ATX di un comune personal computer, privato del coperchio

L'apparato è reso più complesso dalla presenza di sistemi di protezione contro sovraccarichi e cortocircuiti, e da filtri necessari per evitare che il segnale ad alta frequenza si propaghi verso il carico oppure ritorni verso la rete elettrica.

• Il ripple

È uno dei parametri che caratterizzano la qualità di un alimentatore, la sua misura si effettua con l'oscilloscopio accoppiando l'ingresso in AC. Si tratta principalmente del rimanente residuo della componente alternata della rete elettrica, rettificata dai diodi e livellata dai condensatori di filtro; a questa piccola componente oscillatoria spuria, possono aggiungersi disturbi indotti dal carico o dovuti al funzionamento interno dell'alimentatore (presente soprattutto negli alimentatori a commutazione). Questo residuo è costituito da una lieve oscillazione della tensione di uscita avente, nel primo caso, andamento identico alla sinusoide di ingresso, in pratica questo piccolo disturbo si sovrappone alla tensione continua fornita in uscita. Per usi generici, questa componente residua risulta ininfluente sull'utilizzo, in altri casi, esempio fornire alimentazione ad un circuito in progetto, operante a bassi livelli di tensione o elevato guadagno, un ripple relativamente alto può essere di disturbo. Per questa ragione i migliori alimentatori da laboratorio sono in tecnologia lineare, con un ripple garantito su tutto il range di tensione/corrente, contenuto entro pochi millivolt.

⇨ Motori stepper

I motori passo-passo, spesso chiamati anche step o stepper, sono considerati la scelta ideale per tutte quelle applicazioni che richiedono precisione nello spostamento angolare e nella velocità di rotazione, quali la robotica, le montature dei telescopi ed i servomeccanismi in generale. Tuttavia ultimamente, per le applicazioni high-end, vengano spesso sostituiti da motori brushless o da attuatori voice-coil.
Indice

Vantaggi dei motori passo passo

- Quelli costruiti con tecnologia comune hanno un costo non elevato, relativamente ad altri tipi di motore con analoghe prestazioni.
- È possibile realizzare azionamenti di precisione controllati da computer in catena aperta, cioè senza utilizzare sensori di posizione o di velocità. Sono quindi utilizzabili con relativa semplicità e senza richiedere particolare potenza di calcolo.
- Hanno un'elevata robustezza meccanica ed elettrica: infatti non esistono contatti elettrici striscianti e, se necessario, possono essere realizzati anche in ambiente completamente stagno.
- È facile far compiere all'albero piccole rotazioni angolari arbitrarie in ambedue i versi e bloccarlo in una determinata posizione.
- La velocità di rotazione può essere molto bassa anche senza l'uso di riduttori meccanici.
- Hanno molto spesso momento d'inerzia piuttosto basso
- Sono molto stabili nella posizione a rotore bloccato e non presentano pendolamenti come nei sistemi brushless
- Se dimensionati bene non necessitano di alcuna taratura.

Difetti dei motori passo passo

- Richiedono sempre circuiti elettronici per il pilotaggio, in genere di tipo digitale.
- Hanno un funzionamento a scatti e producono vibrazioni, soprattutto ai bassi regimi e se si adottano le tecniche di pilotaggio più semplici.
- Il loro rendimento energetico dipende dalla tecnologia costruttiva adottata, la potenza meccanica espressa come coppia e misurata in Nm (Newton per metro), a parità di assorbimento in corrente, dipende spesso dal tipo di pilotaggio elettrico/elettronico adottato.
- Permettono una velocità di rotazione massima intorno a 1000-1500 rpm. Esistono tuttavia motori che raggiungono i 4000-5000 rpm tramite sistemi di retroazione ad anello chiuso. La loro caratteristica di coppia tuttavia scende quasi esponenzialmente al crescere della velocità.

Il principio di funzionamento

I motori passo-passo sono motori che, a differenza di tutti gli altri, hanno come scopo quello di mantenere fermo l'albero in una posizione di equilibrio: se alimentati si limitano infatti a bloccarsi in una ben precisa posizione angolare.

Solo indirettamente è possibile ottenerne la rotazione: occorre inviare al motore una serie di impulsi di corrente, secondo un'opportuna sequenza, in modo tale da far spostare, per scatti successivi, la posizione di equilibrio.

È così possibile far ruotare l'albero nella posizione e alla velocità voluta semplicemente contando gli impulsi ed impostando la loro frequenza, visto che le posizioni di equilibrio dell'albero sono determinate meccanicamente con estrema precisione.

La costruzione elettromeccanica

I motori passo-passo si dividono tradizionalmente in tre grandi gruppi: motori a magnete permanente, motori a riluttanza variabile e motori ibridi; questi ultimi sono i migliori. In realtà la quasi totalità di quelli oggi reperibili sono proprio del terzo tipo.
Struttura di un motore passo-passo

Un motore ibrido è costituito da un rotore e da uno statore; nella fotografia a lato è riportato un esemplare non particolarmente recente ma in cui si vede chiaramente la struttura.

Il rotore appare come una coppia di ruote dentate affiancate e solidali all'albero (i "denti" sono chiamati coppette) costituite da un nucleo magnetico (le due ruote sono permanentemente magnetizzate, una come NORD, l'altra come SUD) e le coppette in materiale ferromagnetico. Il numero di denti è variabile ma 50 è in assoluto il più frequente. Tra le due ruote è presente uno sfasamento esattamente pari ad 1/2 del passo dei denti: il dente di una delle due sezioni corrisponde quindi alla valle dell'altra. Nel rotore non sono presenti fili elettrici e quindi manca completamente ogni connessione elettrica tra la parte in movimento e quella fissa. In genere il rotore è montato su cuscinetti a sfera, anche nei modelli economici.

Lo statore appare come il classico insieme di avvolgimenti ed il circuito magnetico è costituito da 4 o, più frequentemente, 8 "espansioni polari" (otto in quello mostrato fotografia). All'interno dello statore sono presenti piccoli denti che si affacciano esattamente a quelli del rotore o meglio, sono esattamente affacciati al rotore solo il gruppo di denti appartenenti ad una espansione polare e a quella opposta; le altre coppie sono sfalsate rispettivamente di 1/4, 1/2 e 3/4 del passo dei denti. Avvolti intorno ai poli magnetici dello statore ci sono i fili che, opportunamente percorsi da corrente, generano il campo magnetico.

All'esterno sono evidentemente presenti le alimentazioni dei vari avvolgimenti; in pratica le fasi possono essere avvolte secondo due schemi:

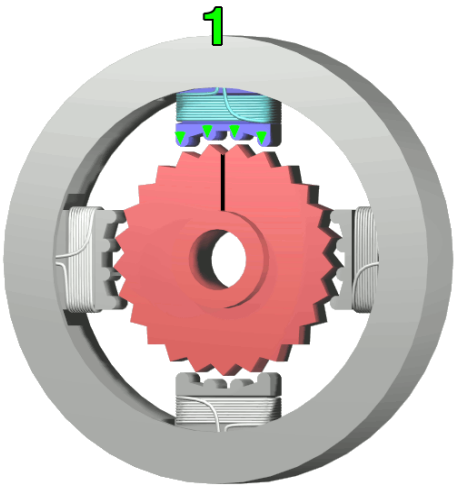
- Sono presenti due soli avvolgimenti (avvolti su più espansioni polari) e quindi all'esterno arrivano due sole coppie di fili: in questo caso si parla di motori bipolari in quanto la corrente dovrà percorrere le fasi nei due versi al fine di creare gli opportuni campi magnetici.
- Sono presenti quattro avvolgimenti avvolti a coppie, in antiparallelo, sulle espansioni polari; all'esterno arrivano almeno cinque fili (spesso sono infatti presenti delle connessioni interne al motore tra le varie fasi).

Si parla in questo caso di motori unipolari in quanto la corrente nella singola fase ha sempre lo stesso verso. È possibile creare due campi magnetici opposti semplicemente scegliendo in quale dei fili debba passare la corrente.

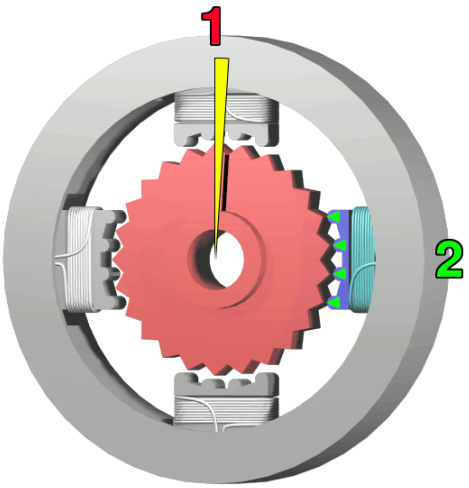
Una tipologia particolare di motore passo-passo è utilizzabile sia in configurazione unipolare che bipolare: si tratta di quelli a 6 o 8 fili.

Il numero di differenti posizioni di equilibrio presenti in una rotazione completa dell'albero è in genere indicato come passi per giro e dipende del numero dei denti del rotore e dai poli dello statore, non dal numero di fili uscenti o dal numero delle fasi, questo numero è spesso stampato sul contenitore ed espresso in gradi.

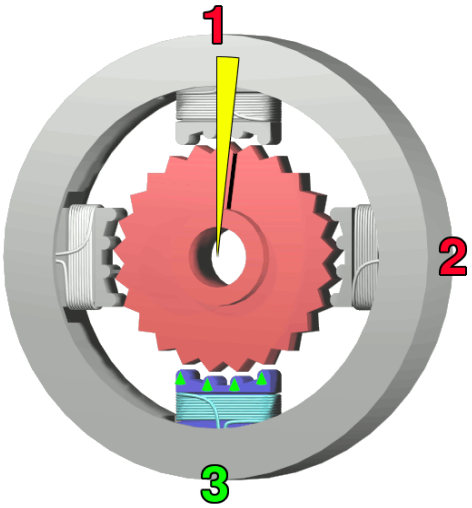
⇨ CONFIGURAZIONI A,B,C,D IN MODALITÀ PASSO INTERO



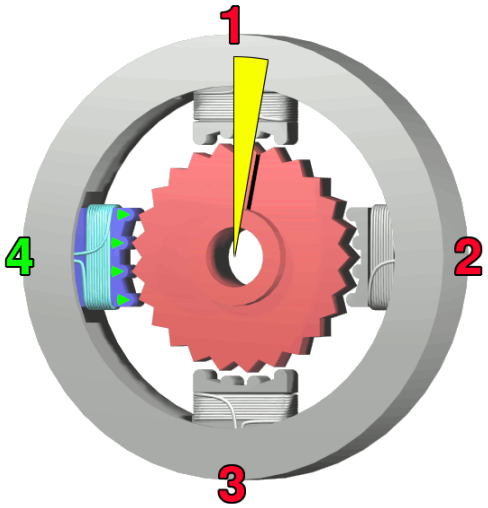
- PRIMA CONFIGURAZIONE



- SECONDA CONFIGURAZIONE



- TERZA CONFIGURAZIONE



- QUARTA CONFIGURAZIONE

⇨ Breve storia e introduzione al controllo numerico

Le macchine CNC, che fino agli anni ‘80 erano usate solo per lavorazioni ad alta precisione, sono oggi molto diffuse e impiegate in quasi ogni campo della meccanica. La tecnologia delle macchine CNC ha coperto un po’ tutti i rami della meccanica; le macchine a CNC più comuni sono presse piegatrici, punzonatrici, torni, fresatrici e macchine di taglio lamiera (laser, ossitaglio, plasma, waterjet ecc.). Esse rappresentano l’evoluzione delle macchine NC, perché permettono il controllo numerico diretto da un computer esterno (DNC).

CNC è l’abbreviazione di “Computer Numeric Control”, ossia una macchina per lavorazioni meccaniche, vedi tornio o fresa, la quale viene comandata da un computer per eseguire delle lavorazioni o pezzi.
Con un programma CAD (vedi Autocad), viene disegnato il particolare da realizzare, poi con un programma CAM (vedi Mastercam) viene generato il programma usato dal controllo del CNC.
Una macchina a CNC tipo hobbistico è generalmente costituito dalle seguenti parti:

- Computer
- Programma di controllo CNC
- Il file in formato Gcode del pezzo o lavorazione da eseguire
- Il controllo elettronico del CNC (Stepper drivers)
- Il gruppo alimentazione del controllo elettronico.
- I motori passo passo (step motor)
- I programmi CAD, CAM ed interpreti vari.
- La fresa che esegue le lavorazioni

Computer

Su questo computer è di norma installato il programma di controllo del CNC

●La porta parallela è il modo di comunicazione normalmente più usato tra computer ed il controllo elettronico.

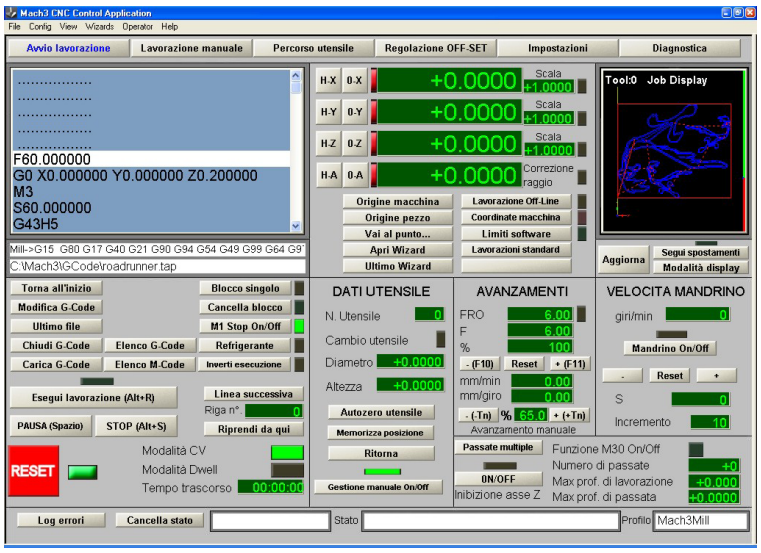
Programma di controllo CNC

●Nel nostro caso abbiamo utilizzato il software MACH3 della artsoft. Il suo compito è quello di trasformare il file in Gcode del pezzo da eseguire in una serie di segnali (passo e direzione), da inviare ai motori passo passo della fresa.

●Con questo programma è normalmente possibile eseguire spostamenti manuali degli assi della fresa, introducendo opportuni comandi sulla finestra video del programma di controllo (se presente).

●Con riferimento al programma MACH3, utilizzando opportuni tasti della tastiera, si ottiene manualmente il movimento assi della fresa.

●È inoltre possibile inserire direttamente e mandare in esecuzione comandi in Gcode.



Software mach3 utilizzato

● Su di una finestra del programma è possibile vedere il disegno del percorso utensile inerente il pezzo in lavorazione.

Una ulteriore finestra visualizza le coordinate dei tre assi e in altre finestre vengono visualizzati altre informazioni d’ impostazioni e lavorazioni in corso.

● Ogni programma di controllo ha una sezione dedicata ai parametri di settaggio che generalmente sono:

●Identificazione della porta parallela che comunica con il computer.

●Il settaggio dei pin di tale porta, con il quale si determina l’associazione del passo/direzione di ogni mo-

tore passo passo con il relativo pin e dei vari segnali di input e output usati.

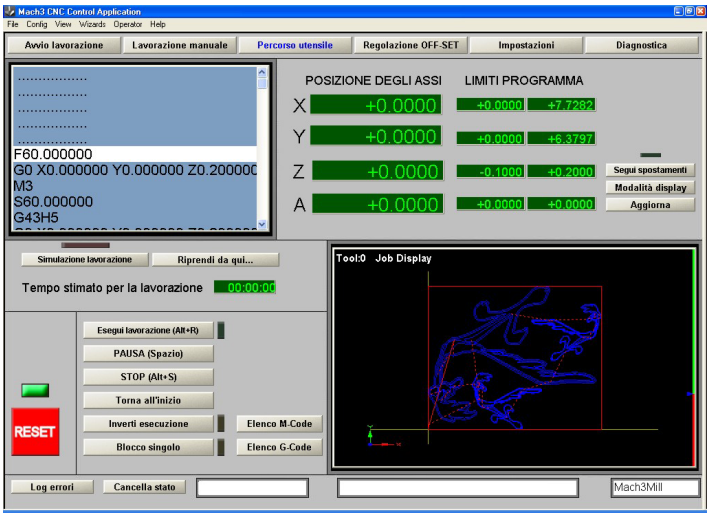
●L'impostazione del valore di spostamento per ogni passo di rotazione dei motori.

●Le massime velocità ed accelerazioni che i motori possono tenere.

●Eventuali compensazioni di gioco (backlash correction) che possono essere presenti tra le boccole e aste filettate degli assi di spostamento.

● Settaggi per eventuali fine corsa installati.

File della parte o lavorazione da eseguire



● È un file generalmente creato da un programma CAD/CAM, oppure può essere generato da un programma di conversione tipo ACE che lo ricava dal file di disegno in formato .dxf precedentemente realizzato con un programma CAD.

● Questo file è in formato ASCII e può essere editato con Notepad.

● Di norma questo file è scritto in linguaggio Gcode, alcune volte in HPGL (file usato per il plottaggio di disegni).

●Un file Gcode è composto da: Blocchi, Comandi e Coordinate.

●Blocchi: sono linee ASCII contenenti i comandi e le coordinate, sono praticamente uno o più comandi che devono essere eseguiti prima di passare al blocco successivo.

●Comandi: sono composti da una lettera e da un numero, i più comuni sono i seguenti:

- ⇨G00 Avanzamento rapido con interpolazione lineare
- ⇨G01 Interpolazione lineare (avanzamento con moto di lavoro)
- ⇨G02 Interpolazione circolare in senso orario
- ⇨G03 Interpolazione circolare in senso antiorario
- ⇨G04 Arresto temporizzato
- ⇨G17 Scelta del piano XY
- ⇨G18 Scelta del piano XZ
- ⇨G19 Scelta del piano YZ
- ⇨G70 Misure in pollici
- ⇨G71 Misure in mm
- ⇨o G90 Coordinate assolute
- ⇨G91 Coordinate relative.

● Vi sono molti altri comandi tipo M usati per controlli ausiliari in fase di lavorazione.

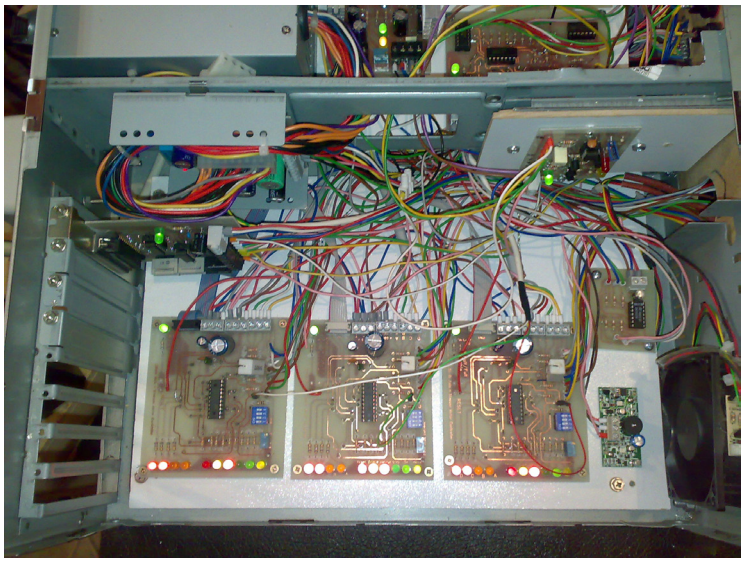
Il file in formato Gcode, pur essendo costruito con comandi standard ed unificati, può non essere letto da qualsiasi programma di controllo CNC.

A livello hobbistico, i programmi di controllo CNC che girano sotto DOS, riescono a riconoscere ed elaborare solo una piccola parte di questi comandi Gcode e generalmente sono sufficienti per la realizzazione di lavorazioni anche complesse.

Questo insieme di comandi che il nostro programma di controllo CNC riesce a riconoscere viene denominato "Post-processore".

Il programma CAM quindi che dovrà trasformare il disegno della parte da realizzare nel file Gcode, dovrà essere settato ed impostato in modo tale da usare il Post-processore del programma di controllo CNC che andrà a comandare la fresa, ossia dovrà impiegare solo un certo tipo di comandi Gcode che possono essere interpretati dal programma di controllo CNC.

Controllo elettronico (stepper drivers)



- È costituito da una o più schede elettroniche che alimentano e fanno muovere i motori passo passo nella giusta sequenza ed in maniera sincronizzata.
- Il controllo elettronico può essere generalmente di due tipi:

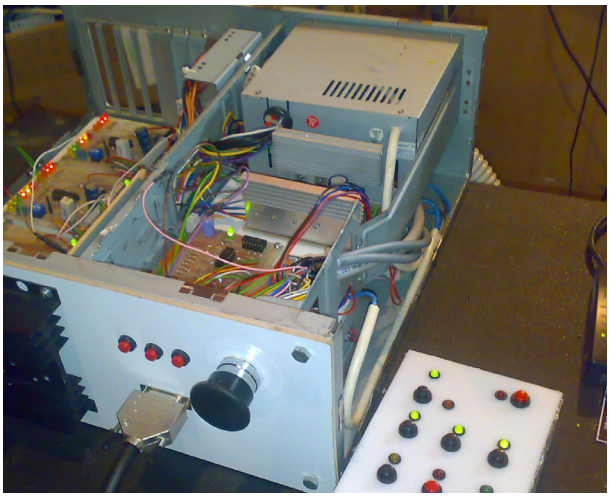
o Per motori unipolari, si riconosce da vistose resistenze presenti in vicinanza del connettore di collegamento dei motori.
Sono controlli non molto costosi.
I motori presentano solitamente 5 o 6 fili e non hanno molta potenza.

o Per motori bipolari che presentano normalmente 4,6 o 8 fili, mai 5.
Sono più performanti ed i componenti elettronici che compongono il controllo sono sofisticati per il loro funzionamento.

Per i motori bipolari, la tensione di alimentazione normalmente va da 5 a 20 volte il valore di targa riportato sul motore.

- Solitamente all'interno del box di tale controllo, risiede pure la parte di alimentazione di potenza.
- Per il collegamento del controllo elettronico al computer, normalmente si usa ol connettore DB25 (quello della porta parallela), ma anche via USB o RS 232.

Gruppo di alimentazione (Power Supply)



- Da questo gruppo viene prelevata l'alimentazione di potenza per i motori e per l'alimentazione delle schede.
- L'alimentazione di potenza, nel caso di drivers per motori unipolari, viene erogata da un alimentatore switching da computer che fornisce in uscita 5V con una corrente max di 20A. solitamente tale tensione viene raddrizzata da un ponte a diodi e livellata da un condensatore elettrolitico di circa 10.000mF.
- Per l'alimentazione delle schede normalmente si usa un secondo trasformatore toroidale che fornisce una tensione che viene poi stabilizzata a 5 Volt tramite un relativo alimentatore.

Motori passo passo (step motor)

- Per il nostro uso solitamente si impiegano motori di provenienza "surplus", ossia smontati da macchine dimesse (vedi stampanti), o da resti di magazzino non più usati.

Si hanno quindi prezzi più accessibili rispetto all'acquisto di motori nuovi.

- Per uso bipolare si usano motori con 4, 6 o 8 fili.
- Per uso unipolare da 5 6 o 8 fili.
- I motori sono classificati anche per il numero di passi che compiono per fare un giro, su frese CNC solitamente si impiegano motori da 200 passi (1,8 gradi).

motori da 48 passi sono poco costosi ma non molto adatti all'uso su frese CNC .

Programmi CAD, CAM ed interpreti vari.

- Con il CAD noi disegniamo la parte che poi viene lavorata.

Il più conosciuto è Autocad, ma molti altri CAD, anche migliori, possono essere impiegati, l'estensione del file è solitamente .dxf

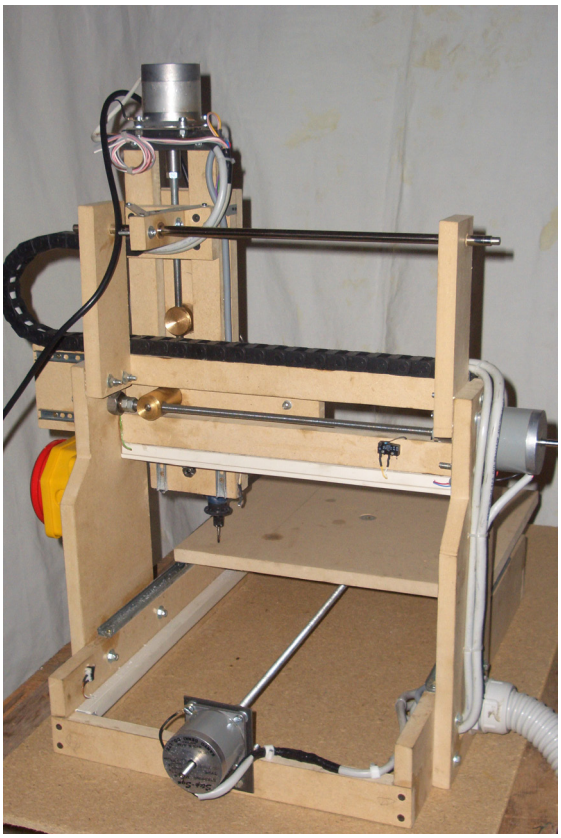
- Con il CAM noi trasformiamo il file del disegno da .dxf ad un file in G-code.

Sarà questo file G-code che verrà letto dal Programma di controllo CNC che permetterà alla nostra fresa di eseguire le lavorazioni.
Se non disponiamo di un programma CAM, possiamo utilizzare semplici programmi di conversione o interpreti che ci trasformano il file .dxf in G-code, il più conosciuto è ACE.

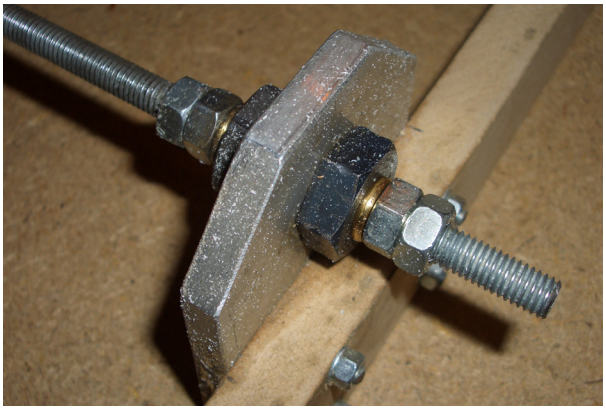
- Vi sono dei programmi di controllo fresa CNC che al loro interno hanno già un CAM, vedi DeskCNC, oppure posono convertire un file da .dxf in G-code, vedi Kcam4.

La fresa che esegue le lavorazioni
È la parte meccanica di tutto il sistema, su questa troviamo montati i motori passo passo e l'utensile fresa che eseguirà le lavorazioni.
La sua struttura è in funzione delle dimensioni dei pezzi da lavorare e del tipo di materiale che andremo a lavorare.
Materiali duri richiederanno una struttura della fresa robusta e rigida.
Solitamente gli assi di lavorazione sono tre: X,Y e Z.
Il sistema di guida dei tre assi è molto importante che sia preciso e scorrevole

Capacità delle macchine CNC



La maggior parte dei centri di lavoro controllati da computer sono dotati di movimento verticale del mandrino in grado di realizzare incisioni, sculture e lavorazioni di grandissima precisione. Lavorazioni realizzate su queste macchine possono anche arrivare ad essere indistinguibili ad un occhio non espertissimo rispetto a oggetti d'arte realizzati a mano. Se utilizzate con frese coniche o sferiche possono arrivare a creare superfici assolutamente lisce, di altissima precisione, in modo rapido, automatizzato e ad un costo estremamente contenuto. Le migliori macchine CNC arrivano a una precisione di un decimillesimo di millimetro (100 nanometri).
Le macchine CNC più avanzate sono dotate di testate orientabili (assi tilting) in grado di ruotare giroscopicamente lungo due assi (normalmente chiamati A, B, C o Q). Questo consente di inclinare l'utensile rispetto a tutti i piani di lavoro rendendo possibile realizzare figure molto complesse anche con forme di tipo organico e difficilmente ottenibili persino da parte di una lavorazione manuale (come ad esempio una cavità con un singolo foro molto stretto ma sufficiente per far passare la fresa. Le teste tilting permettono anche di ottimizzare la lavorazione avendo la possibilità di lavorare con l'utensile inclinato rispetto al piano d'avanzamento evitando di utilizzare la parte dell'utensile a velocità zero (centro di rotazione della fresa). Con i controlli numerici più recenti iniziano ad implementare funzioni di lavorazione avanzata delle superfici, permettendo di lavorare superfici 3D NURBS in modo nativo, senza bisogno di programmi CAM intermedi. Lo sfruttamento ottimale delle caratteristiche di queste macchine può avvenire o attraverso appositi accessori per copiatura e stampi, oppure via computer, attraverso una catena di programmi: prima si crea con il CAD un oggetto, che poi viene passato al CAM che si incarica di creare il programma per la realizzazione su una data macchina, e infine il programma

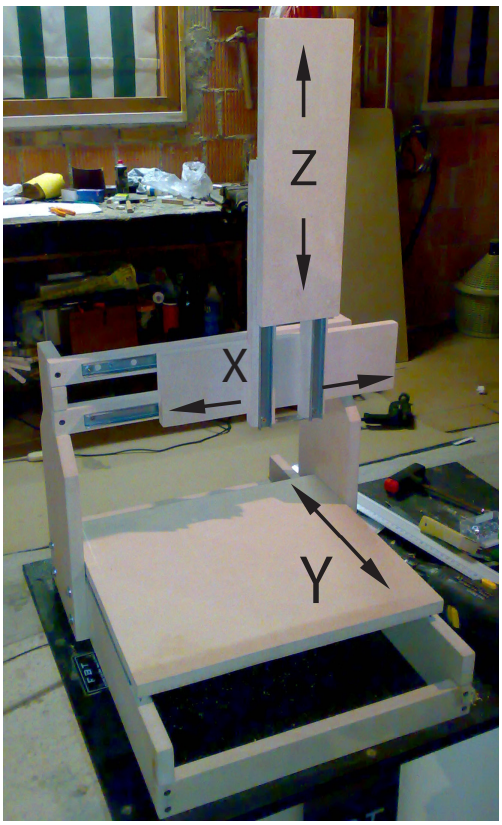


viene eseguito dalla macchina creando l'oggetto materiale. Praticamente tutte le macchine CNC moderne sono "a circuito chiuso" (a catena chiusa o retroazionato): ogni asse è dotato di encoder (encoder rotativi oppure righe ottiche) che permettono al controllo di conoscere in ogni istante la posizione della testa di lavorazione rispetto al pezzo. L'uso di righe ottiche permette il recupero completo dei giochi della trasmissione misurando direttamente e non attraverso i cinematismi la posizione dell'asse e quindi una miglior precisione di lavoro.

Assi

Caratteristica principale di queste macchine è il numero di gradi di libertà disponibili, detti assi della macchina. Per le frese sono generalmente 3, 4 o 5, per i torni si va da 2 a 4, le punzonatrici e le macchine per taglio hanno generalmente 2 o 3, mentre le pannellatrici (FINN-POWER, Salvagnini...) arrivano a 24 assi. Le piegatrici possono avere fino a 11 assi e sono le macchine "singole" più complesse; le macchine speciali combinate e i grandi portali di lavorazione possono arrivare ad avere anche decine di gradi di libertà o assi diversi. I tipi più comuni sono

- 2 assi: movimento solo su X e Y. Pezzi limitati a lavorazioni piane senza variazioni di profondità.
- 2.5 assi (due assi e mezzo): si tratta di macchine in grado di operare su tutti e tre gli assi, ma soltanto a passi discreti su uno di essi (generalmente l'asse Z), cioè gestiscono l'interpolazione soltanto su due assi. Sono oggi il tipo più economico.
- 4 e 5 assi: oltre al movimento sui tre assi X,Y e Z si aggiunge l'inclinazione (ed eventualmente la rotazione) del mandrino o di una tavola rotobasculante solidale al piano macchina. Queste macchine possono realizzare praticamente qualsiasi tipo di sagoma purché la forma delle concavità non vada in interferenza con il mandrino. Queste macchine sono molto costose, e vengono usate anche per lavorazioni su pezzi di grandissime dimensioni (turbine, eliche navali, scocche per motori, stampi...)



Nell'industria dei grandi trasporti si usano anche macchine dette "a 8 assi" o "a 10 assi": si tratta di centri di lavoro a due testate indipendenti, in grado di velocizzare le operazioni di lavorazione su pezzi di grandissime dimensioni come gli estrusi di alluminio e acciaio per le casse dei treni o le lamiere degli aerei.

⇨Il Nostro Progetto - CNC 3 assi

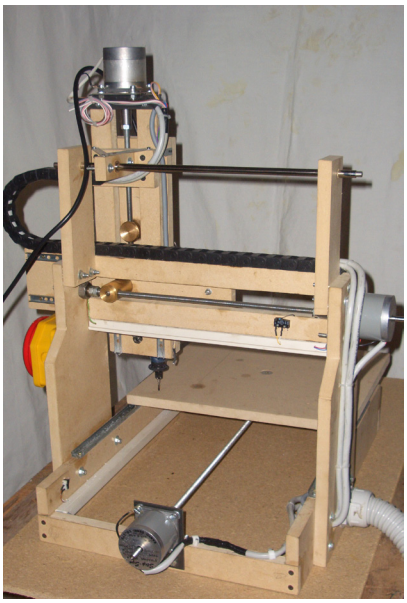
Prima realizzazione della struttura

Il nostro progetto è iniziato partendo dalla realizzazione della struttura della macchina in materiale MDF (Medium Density Fibreboard: “pannello di fibra a media densità”).

Il secondo passo è stato adattare i motori passo passo, attraverso una struttura meccanica composta da una vite senza fine e guida filettata in modo da ottenere un movimento lineare del relativo asse, in base alla rotazione della vite generata dal motore PP.

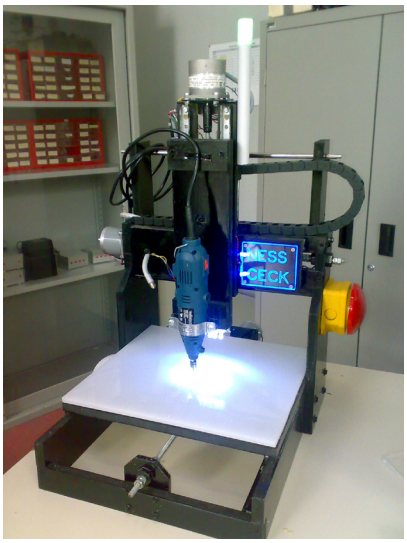
Successivamente, dopo aver apportato una modifica alla struttura della macchina introducendo una guida rialzata per stabilizzare e linearizzare il più possibile il movimento dell’asse X, abbiamo collegato i relativi finecorsa e switch di home rispettivamente ad ogni asse (X,Y,Z).

Dopo aver realizzato le principali schede per il movimento e controllo di tutto il sistema e verificato il suo corretto funzionamento, è stato realizzato un box (case orizzontale di un vecchio pc) per contenere tutti i circuiti di pilotaggio/gestione ed i gruppi di alimentazione.



Il software utilizzato per il funzionamento dell’intero sistema (Arcsoft - Mach3) è stato opportunamente impostato in tutti i suoi parametri, secondo quelli richiesti da tutta l’elettronica e dalla meccanica progettata, ad esempio, il numero dei passi che il motore doveva compiere per lo spostamento dell’asse di una unità (1mm); quindi, a sua volta, il numero di impulsi di passo che il pc doveva dare in uscita attraverso la porta parallela per effettuare un movimento.

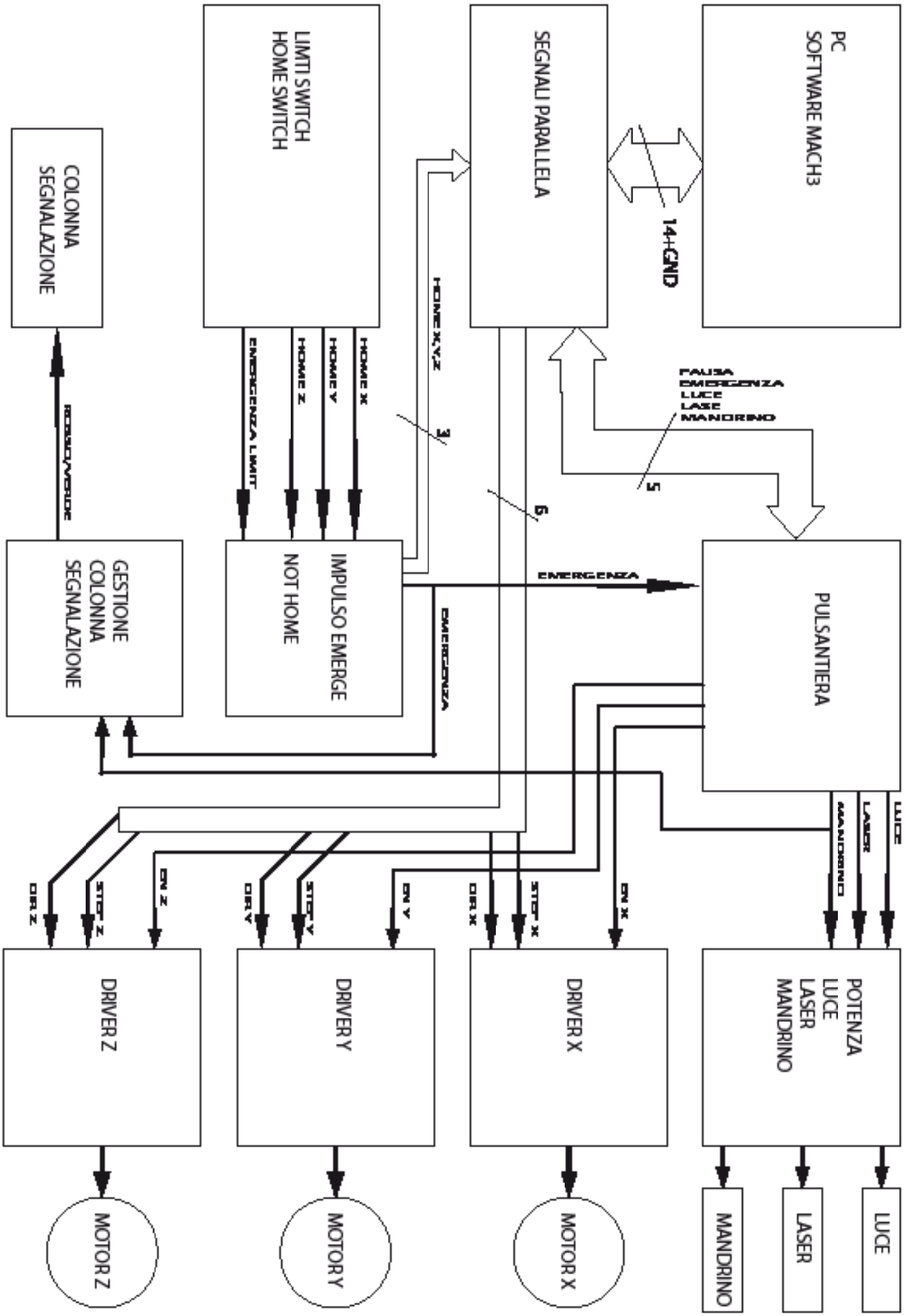
Come ultimo passaggio sono state effettuate una rifinitura meccanica (lubrificazione), una rifinitura estetica, come il cablaggio ordinato dei cavi, sia sulla macchina che all’interno del box, e una verniciatura di colore nero ai vari pezzi.



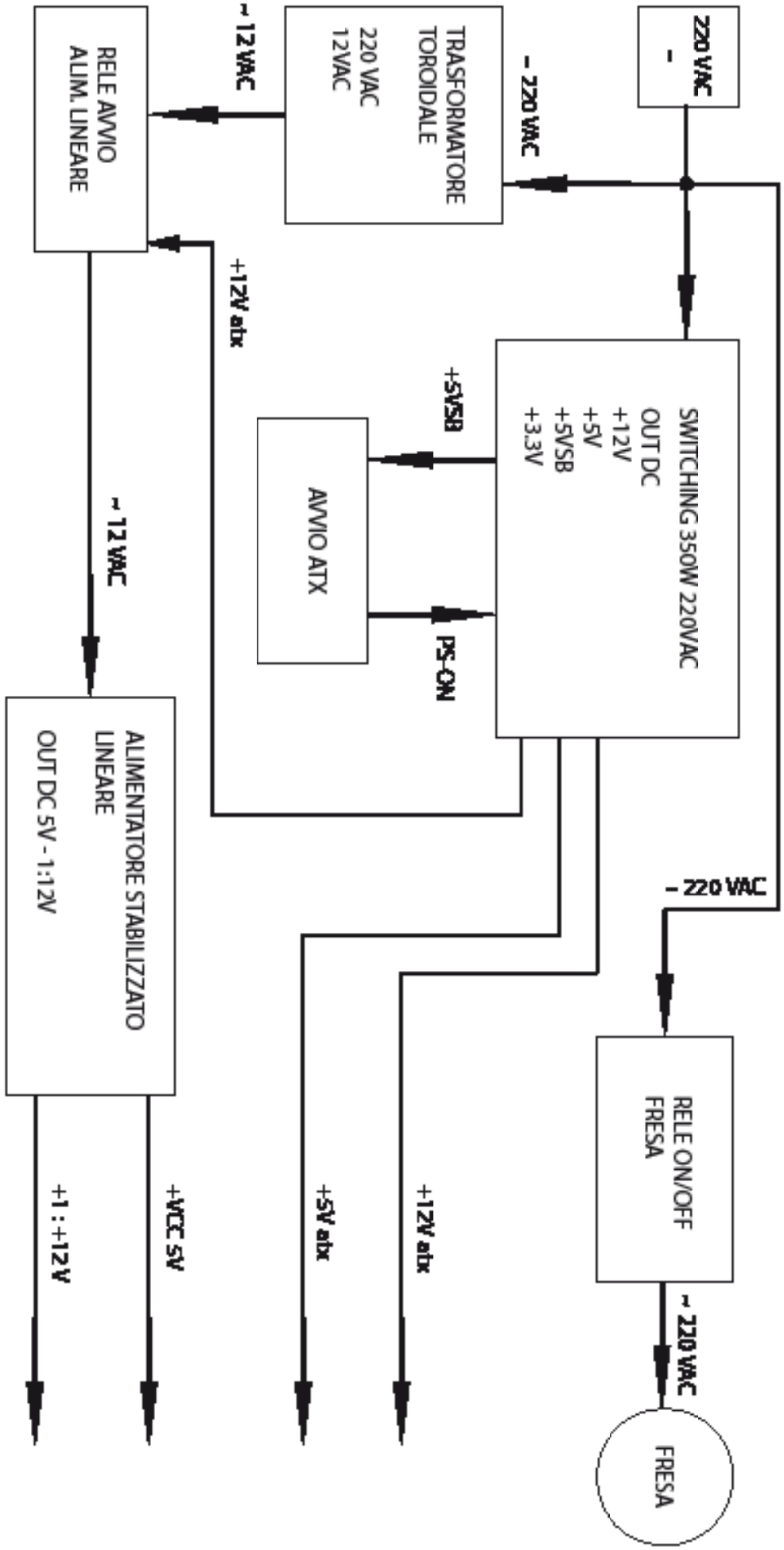
Macchina ultimata con tutte le rifiniture

Box di controllo ultimato

⇨ SCHEMA A BLOCCHI DEI COLLEGAMENTI TRA LE VARIE SCHEDE

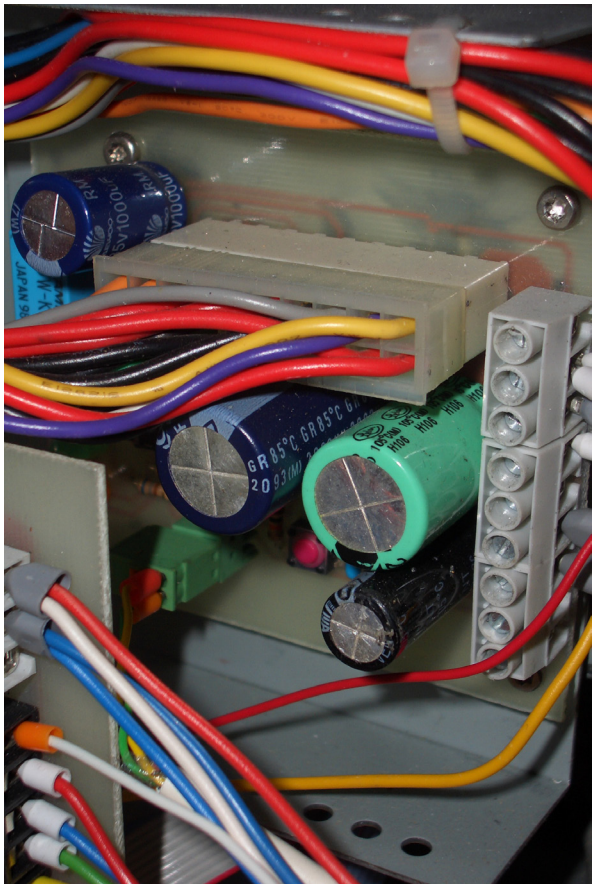


⇨ SCHEMA A BLOCCHI ALIMENTAZIONI



⇨ ACCENSIONE SPEGNIMENTO ALIMENTATORE ATX

► Descrizione Generale



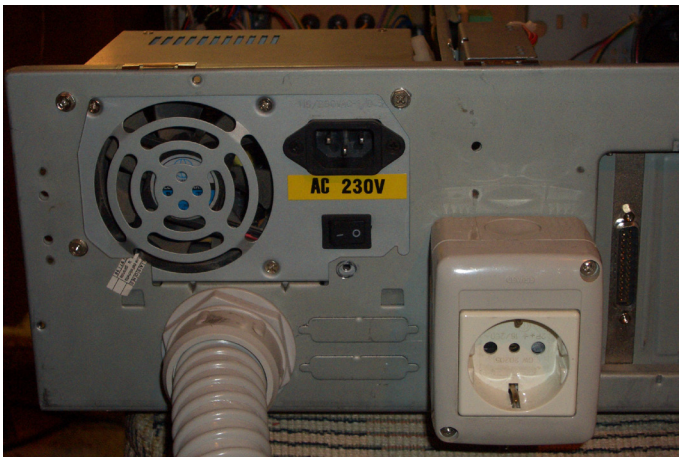
Il circuito ha il compito di avviare l'alimentatore da pc (ATX).

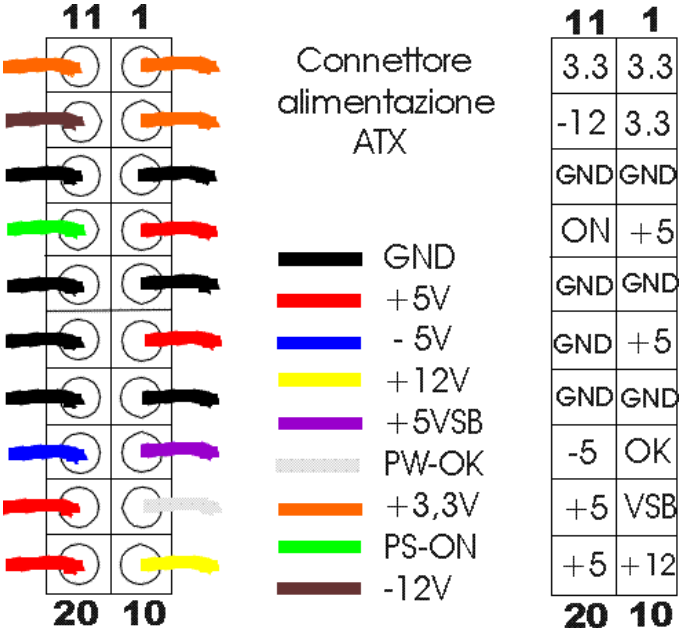
In modalità stanby l'alimentatore fornisce una sola tensione di 5V sul filo viola(+5VSB), qesta tensione va ad alimentare il flip-flop jk e il pulsante. Il flip-flop in configurazione toogle(J,K,preset e clear collegati a 5v), cambierà di stato la sua uscita ogni volta che viene premuto il pulsante o la chiave di avvio collegata come nello schema sul piedino di clock del FF.

quando viene attivata l'uscita verrà attivato anche un relè che collegherà a massa il filo verde(PS-ON) il quale causerà l'avvio dell'alimentatore.

La presenza dell'alimentazione a 5V in modalità stand-by è segnalata dall'accensione di un led giallo, la presenza della tensione a 5V ad alimentatore avviato è segnalata da un led verde. Un led arancio invece segnalerà il corretto avviamento dell'alimentatore e che le varie tensioni d'uscita(3.3V, +5V, +12V, -5V, -12V) sono presenti in maniera corretta(non sono presenti corto circuiti o altre cause di uno scorretto mal funzionamento).

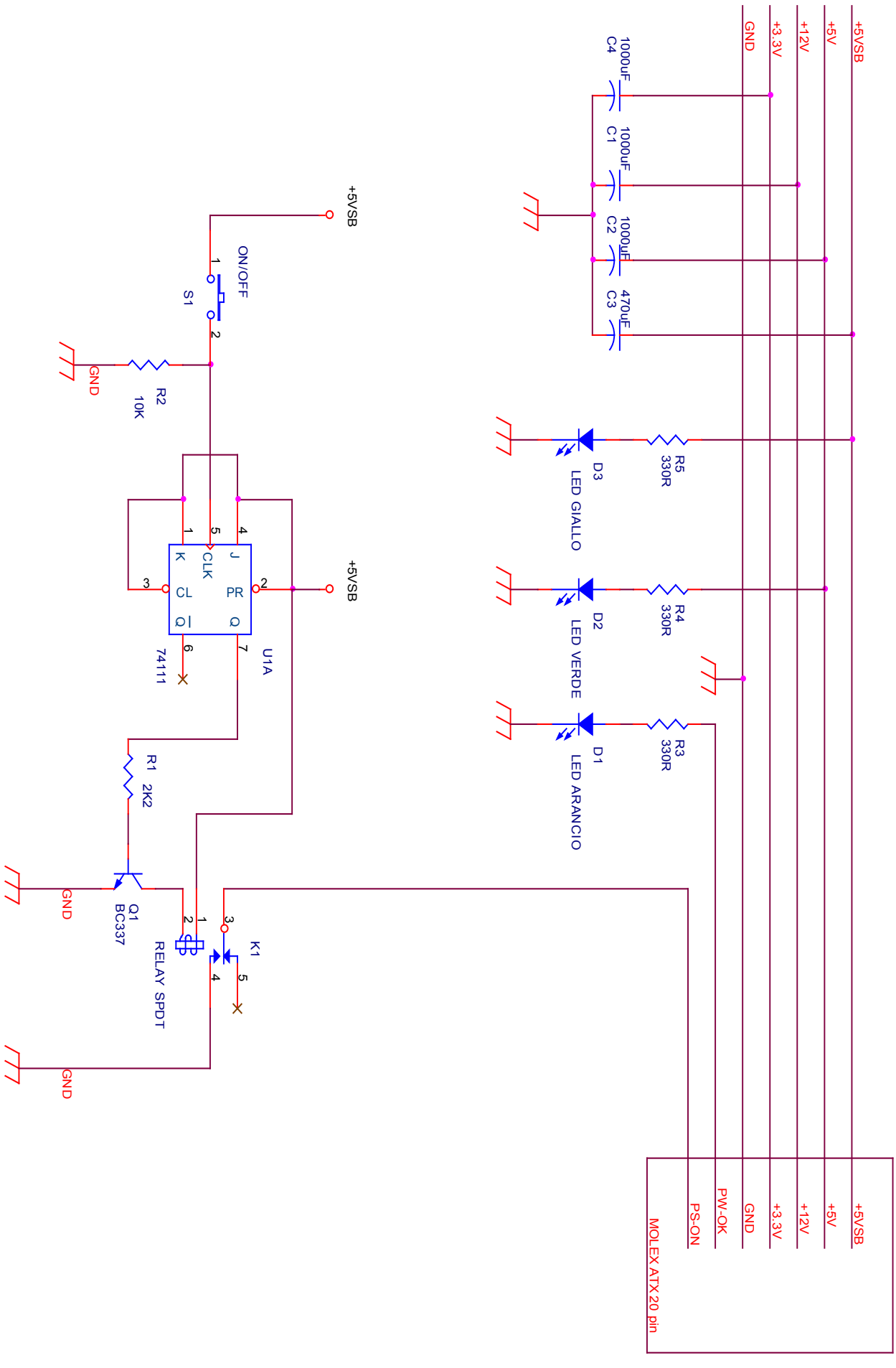
In caso contrario se è presente qualche anomalia il led arancio sarà spento.



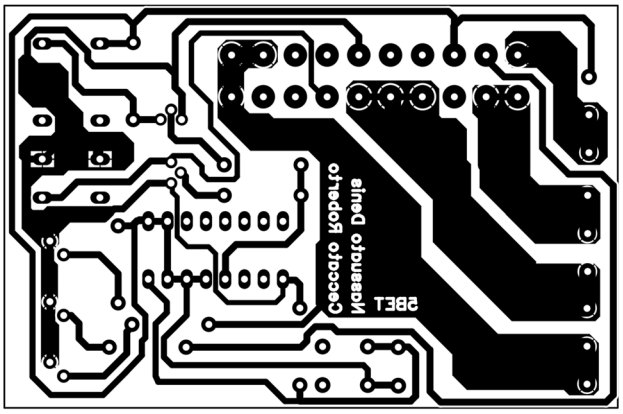


Piedinatura del connettore molex ATX a 20 pin.

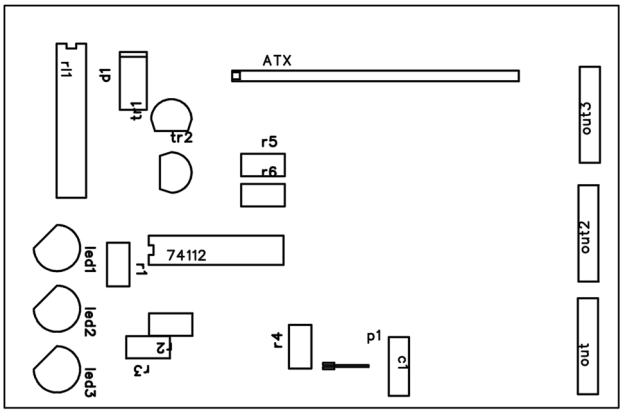
▶ SCHEMA ELETTRICO



► PCB



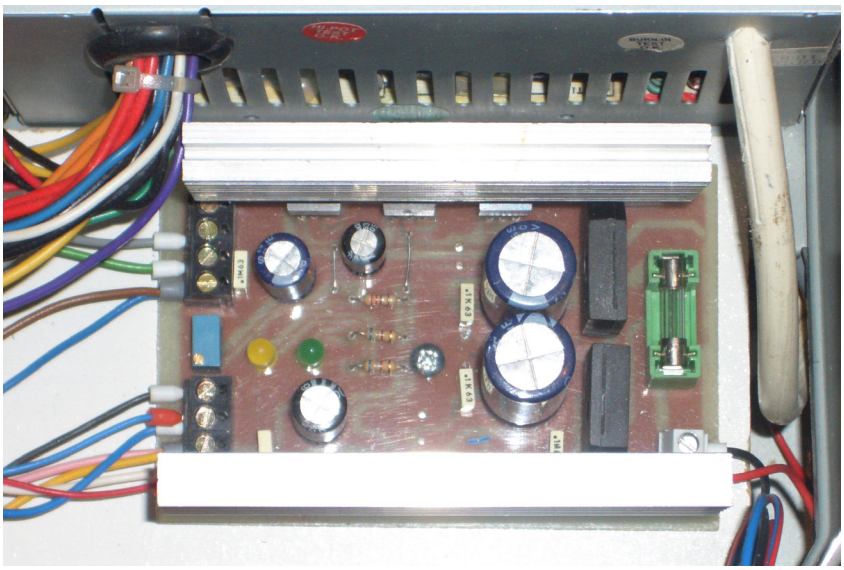
1. BOTTOM COPPER - Lato rame



2. SERIGRAFIA - Posizione componenti

⇨ Alimentatore stabilizzato 5Vdc - 1:12Vdc

► Descrizione Generale



Si tratta di una tecnologia estremamente semplice ed economica, largamente usata ove la potenza richiesta sia limitata ed il costo rappresenti un limite.

Un generico alimentatore lineare è idealmente (e spesso anche praticamente) composto dai seguenti elementi collegati in cascata:

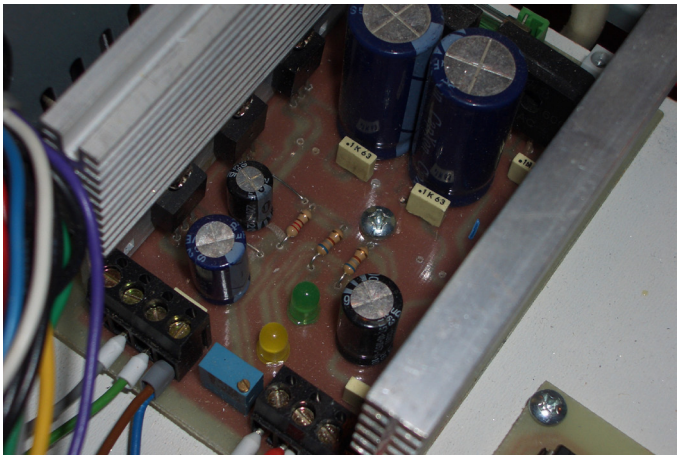
● Un trasformatore: provvede a ridurre (o in rari casi aumentare) la tensione proveniente dalla rete elettrica per avvicinarla al valore richiesto dal carico da servire.

● Un raddrizzatore: trasforma la corrente alternata fornita dal trasformatore in corrente continua. Può essere a diodo singolo o a ponte.

● Un filtro livellatore: livella la corrente unidirezionale pulsante uscente dal raddrizzatore in una corrente più uniforme e costante. Solitamente rappresentato semplicemente da un condensatore.

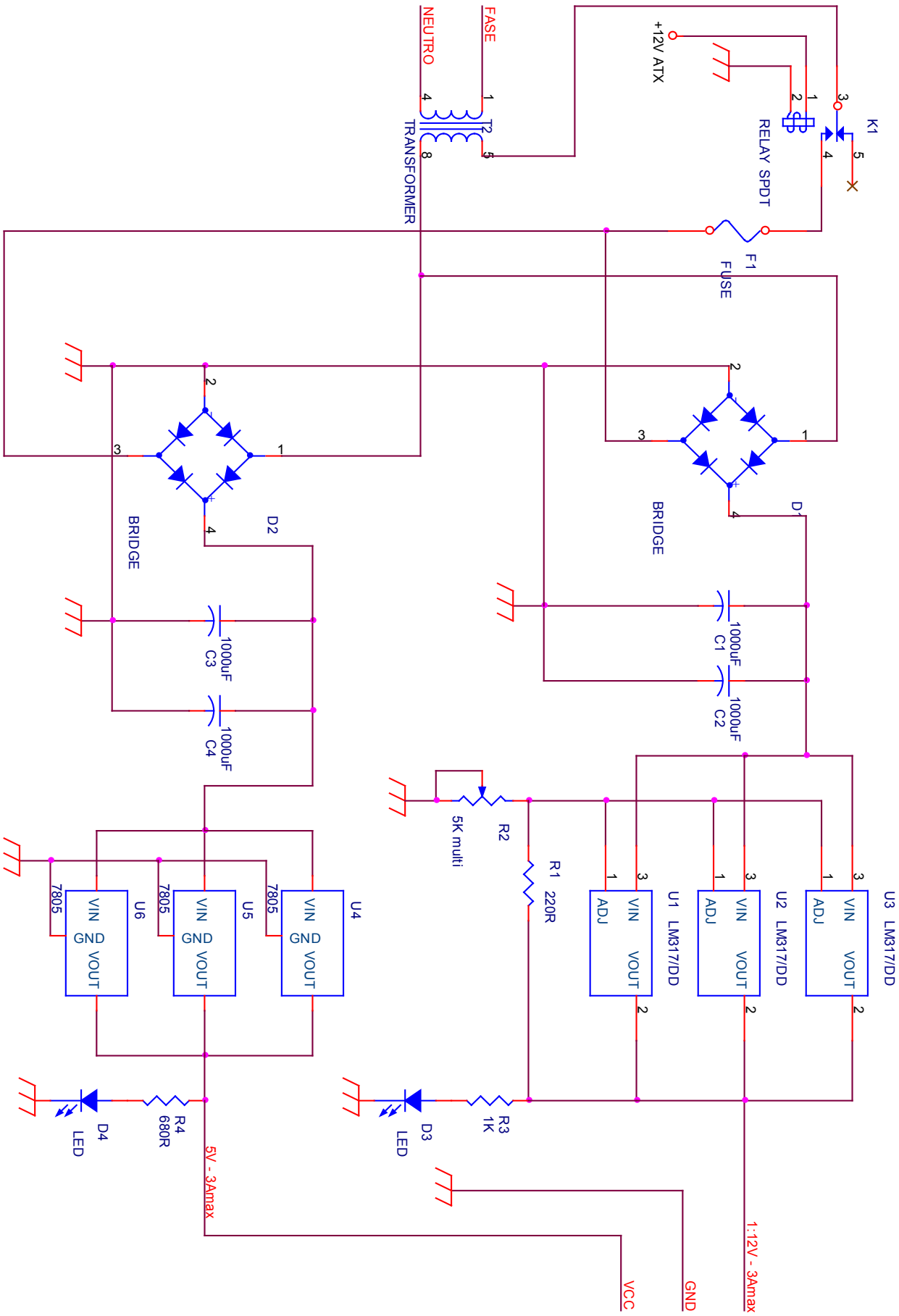
● Un circuito elettronico Stabilizzatore detto anche regolatore, che può spaziare da un semplice diodo zener ad un circuito integrato dedicato. Assicura che la tensione generata dall'alimentatore si mantenga costante nel tempo ed entro una stretta tolleranza rispetto al valore richiesto, al variare della tensione della rete elettrica e del carico applicato. Nel nostro caso per la tensione fissa di 5v come stabilizzatore è stato utilizzato il circuito integrato 7805 su contenitore TO-220, mentre per la tensione variabile da 1:12v il regolatore è l'integrato LM317 sempre in contenitore TO-220.

I principali limiti di questi alimentatori risiedono nel basso rendimento energetico, che comporta, nel caso di elevate potenze gestite, un consistente sviluppo di calore, che deve essere smaltito per evitare danni all'apparato.



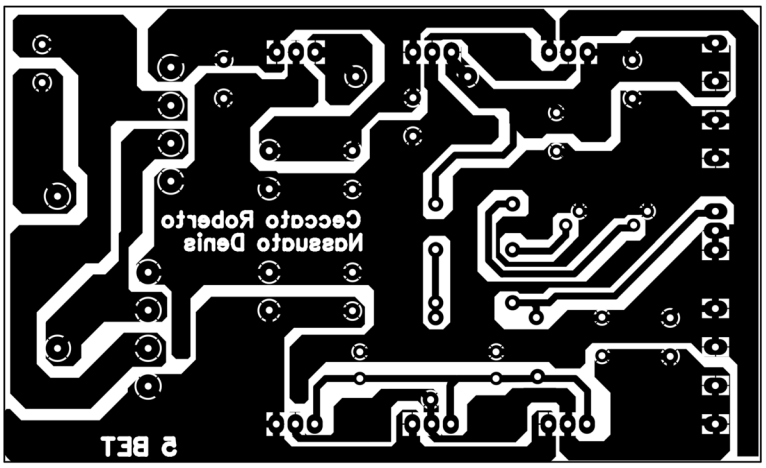
Un altro limite è nell'eccessivo incremento di dimensioni e peso all'aumentare della potenza di un alimentatore analogico.

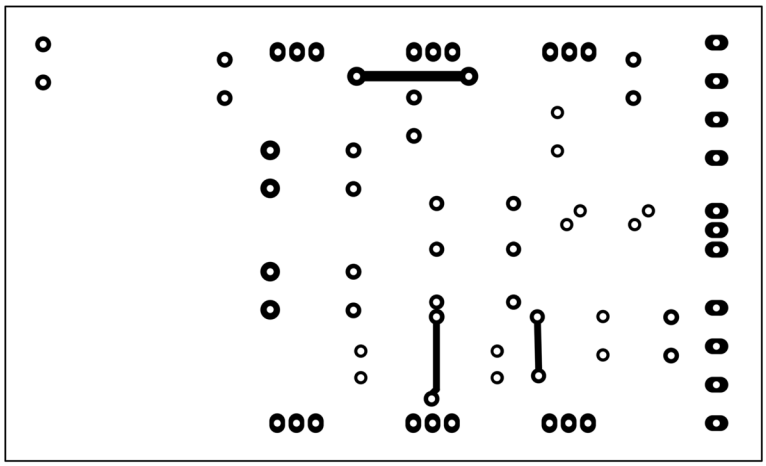
▶ SCHEMA ELETTRICO



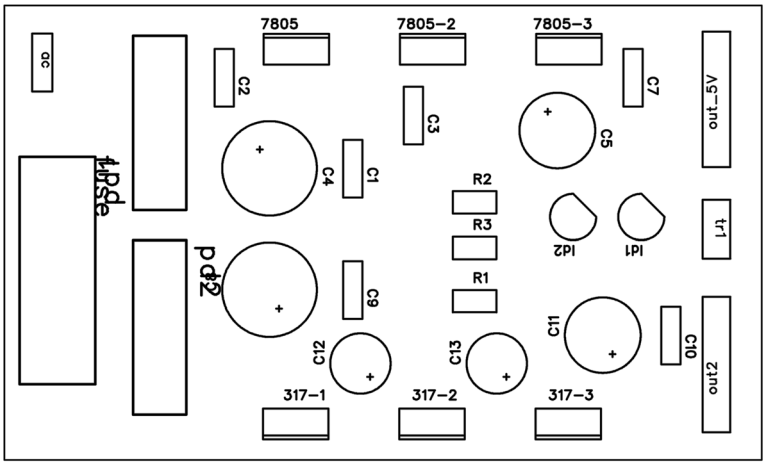
► PCB

1. TOP COPPER - Lato componenti





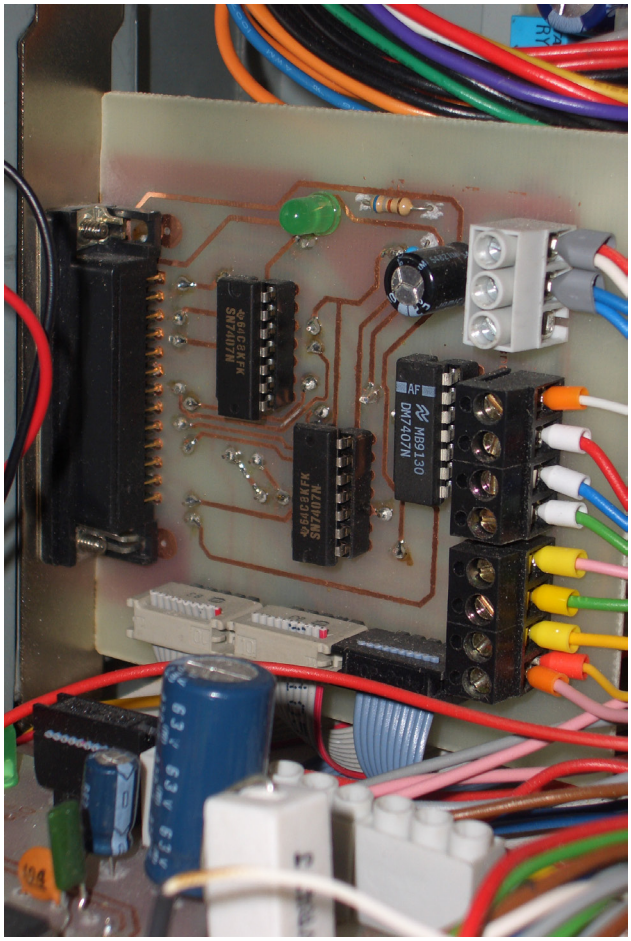
2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

⇨ Porta parallela (LPT)

► Descrizione Generale



Il circuito disaccoppia i segnali provenienti da e verso il pc tramite la porta parallela.

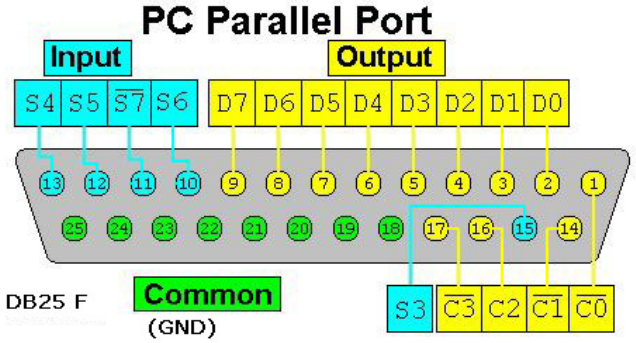
Il disaccopiamento avviene attraverso degli Hex buffer (7407) i quali portano in uscita lo stesso livello (1 o 0) che si presenta al suo ingresso.

I segnali verso la macchina e provenienti dal pc sono:

- STEP ASSE X
- STEP ASSE Y
- STEP ASSE Z
- DIREZIONE ASSE X
- DIREZIONE ASSE Y
- DIREZIONE ASSE Z
- CONTROLLO ON/OFF MANDRINO
- CONTROLLO ON/OFF LASER
- CONTROLLO ON/OFF LUCE LAV.

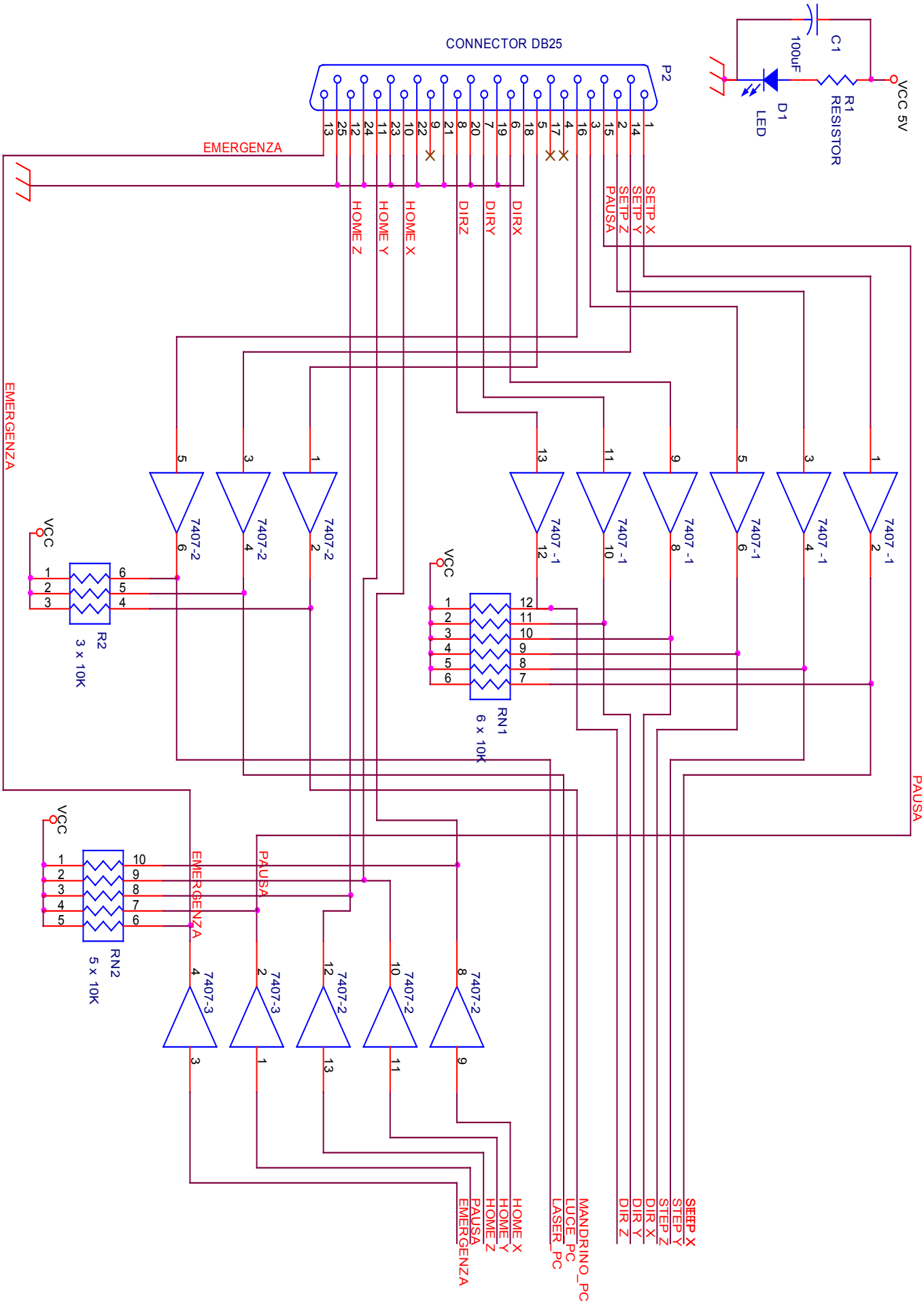
Mentre i segnali provenienti dalla macchina e diretti verso il pc sono:

- EMERGENZA GENERALE
- PAUSA LAVORAZIONE
- HOME ASSE X
- HOME ASSE Y
- HOME ASSE Z

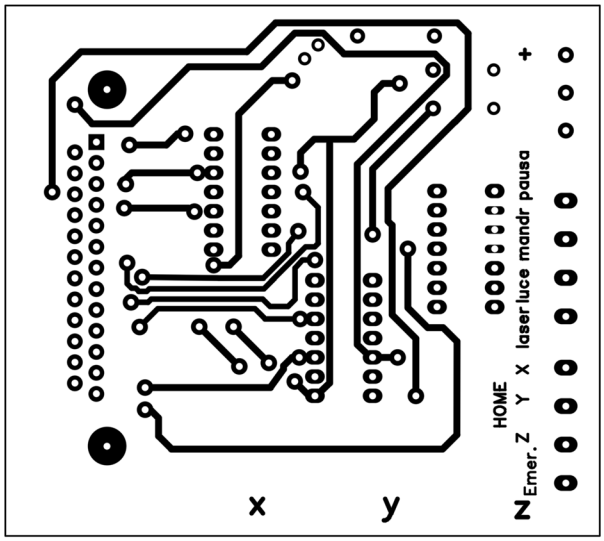


D0 ⇨ STEP X	D6 ⇨ DIR Z	S6 ⇨ HOME Y
D1 ⇨ STEP Y	D7 ⇨ NC	S7 ⇨ HOME X
D2 ⇨ STEP X		C0 ⇨ NC
D3 ⇨ MANDRINO	S3 ⇨ PAUSA	C1 ⇨ LUCE
D4 ⇨ DIR X	S4 ⇨ EMERGEN.	C2 ⇨ LASER
D5 ⇨ DIR Y	S5 ⇨ HOME Z	C3 ⇨ NC

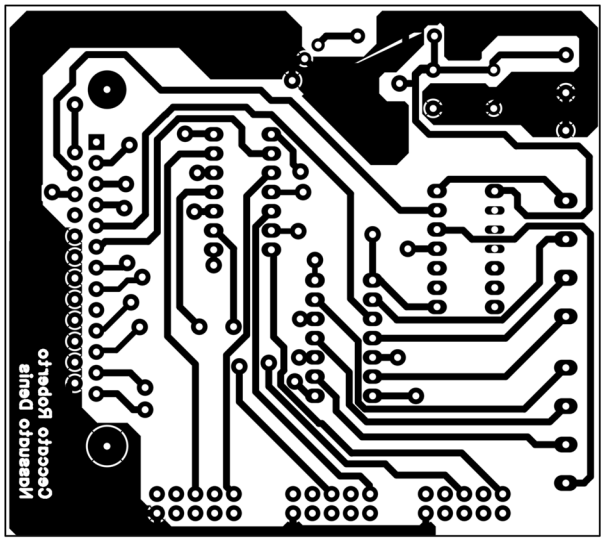
▶ SCHEMA ELETTRICO



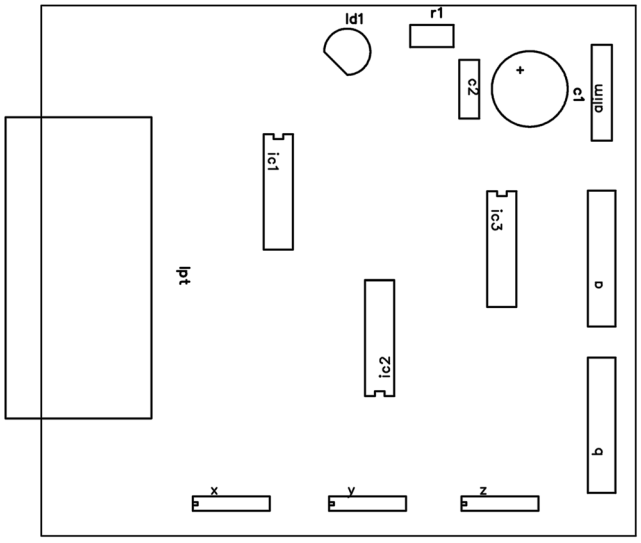
► PCB



1. TOP COPPER - Lato componenti



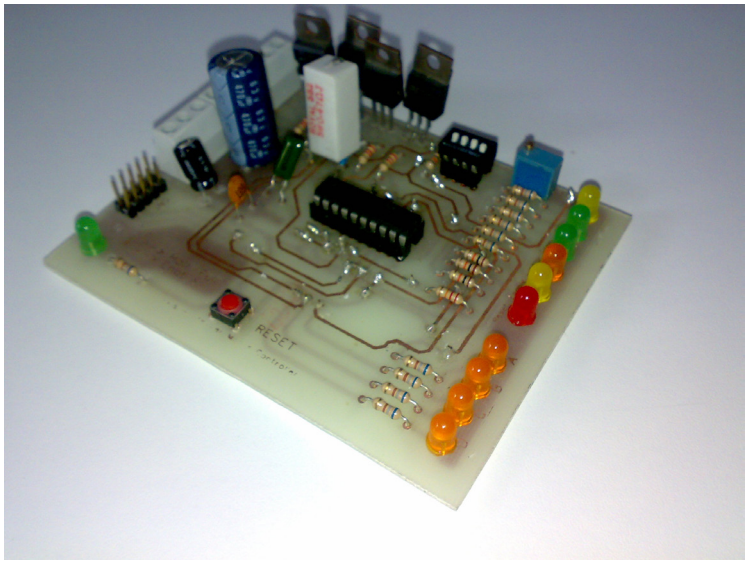
2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

⇨ Driver Motore stepper

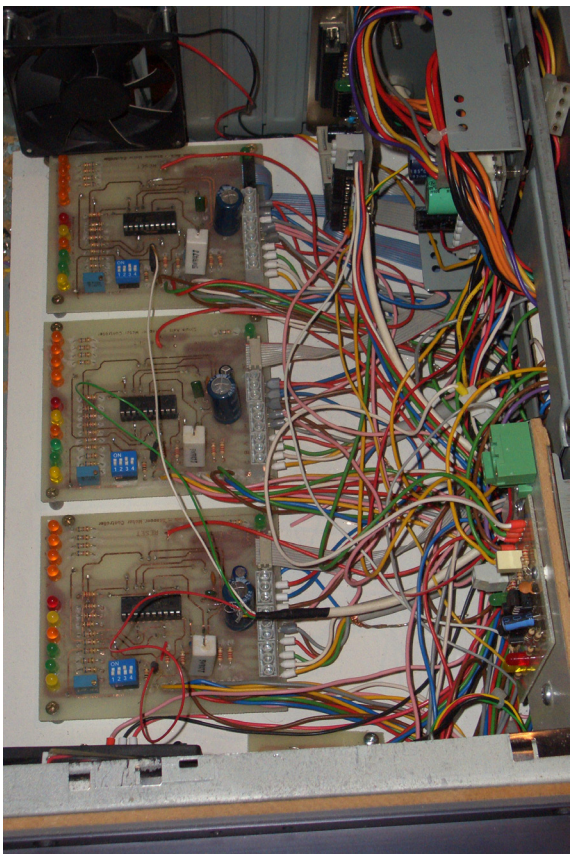
► Descrizione Generale



Il circuito è in grado di gestire e pilotare un singolo motore passo passo (stepper) del tipo unipolare. Il cuore di questo driver sta nell'integrato L297, il quale in base ad alcuni segnali di controllo in ingresso quali:

- STEP: impulso di passo;
- CW/CCW: segnale di direzione della rotazione;
- HALF/FULL: tipo di passo, mezzo passo o passo intero;
- ENABLE: abilitazione del driver.
- RESET: configurazione iniziale delle fasi del motore.

è in grado di gestire direttamente le quattro basi dei quattro transistor darlington, i quali a loro volta pilotano ciascuna delle quattro fasi del motore.



► Funzionamento

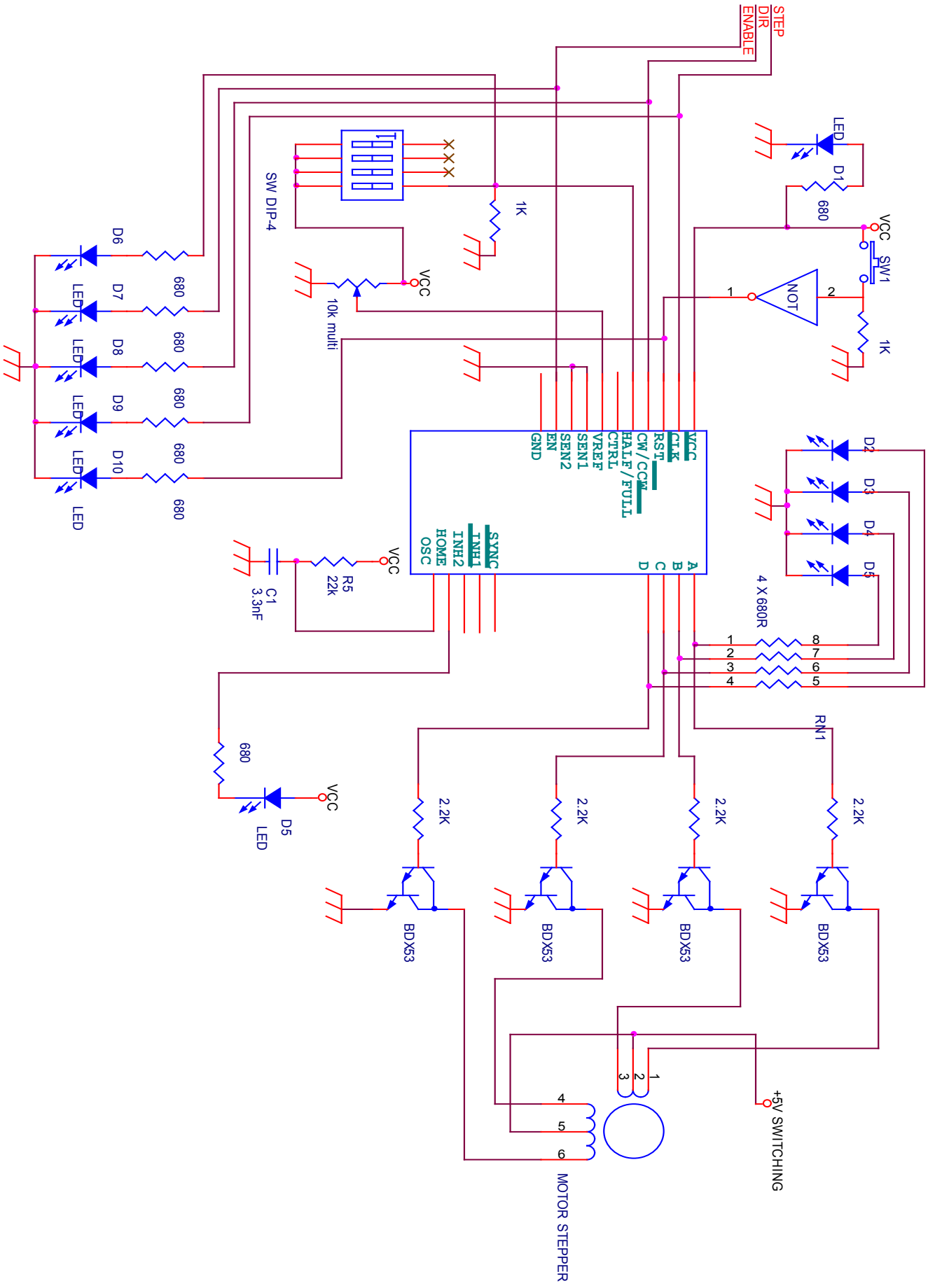
Per attivare la scheda è necessario abilitare il segnale di Enable del relativo asse tramite la pulsantiera di comando, a questo punto il driver manterrà attive le prime due fasi(A,B) del motore, se all'ingresso STEP si presenta un treno di impulsi(in questo caso proveniente dalla porta parallela del PC), l'integrato inizierà la codifica e il pilotaggio delle varie fasi del motore per farlo ruotare in base anche allo stato del segnale di direzione.

La velocità di direzione è proporzionale alla frequenza degli impulsi si step.

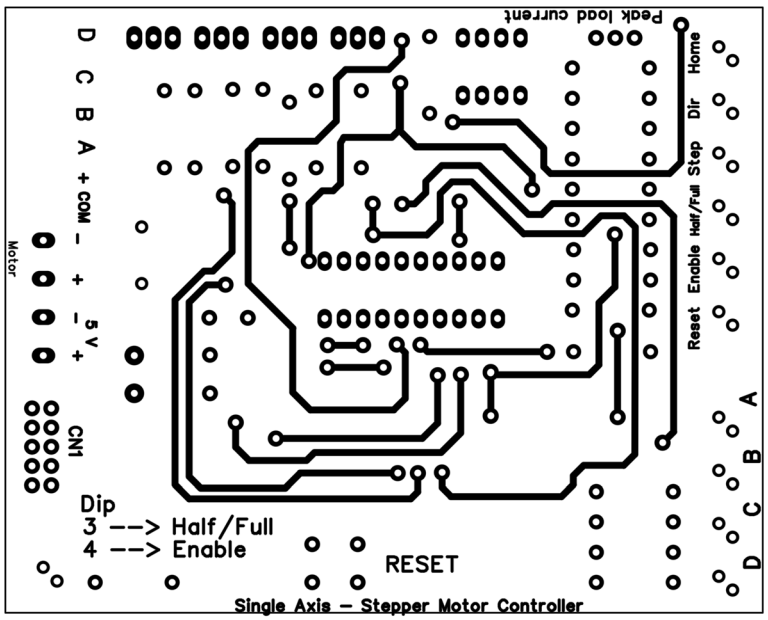
Lo stato di tutti i segnali sia di controllo che delle fasi è visualizzato tramite il relativo led presente sulla scheda.

PER IL COMANDO SONO PRESENTI TRE DRIVER DI QUESTO TIPO, UNO PER CIASCUN MOTORE DEL RELATIVO ASSE (X,Y,Z). TUTTI I SEGNALI DI ABILITAZIONE SONO GESTITI DALLA PULSANTIERA DI COMANDO, I SEGNALI DI STEP E DIREZIONE DAL PC, MENTRE IL TIPO DI PASSO È IMPOSTATO PER TUTTE E TRE LE SCHEDE IN MODALITÀ MEZZO PASSO TRAMITE IL DIP-SWITCH PRESENTE SULLA SCHEDA

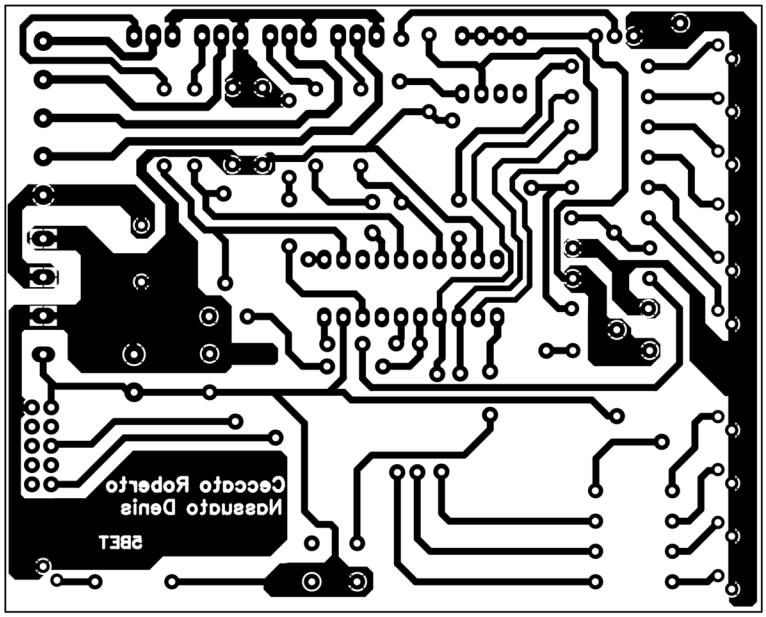
▶ SCHEMA ELETTRICO



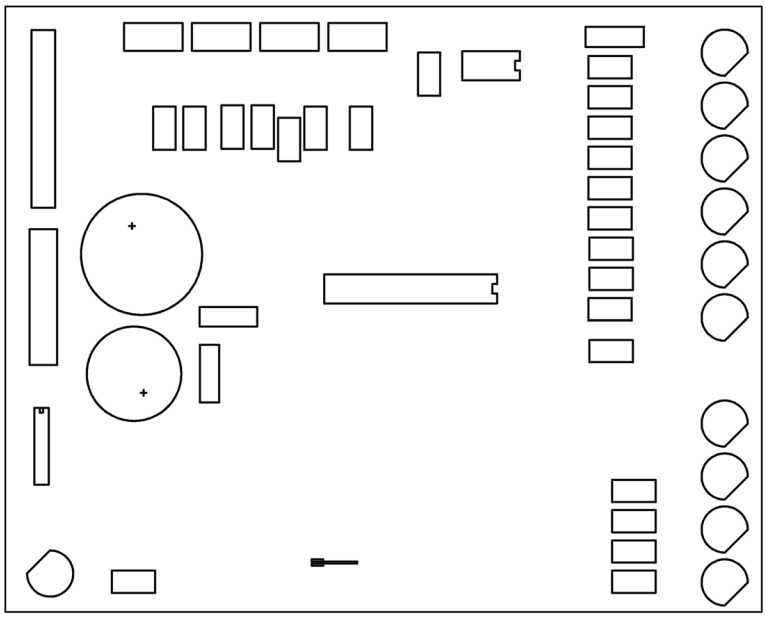
► PCB



1. TOP COPPER - Lato componenti



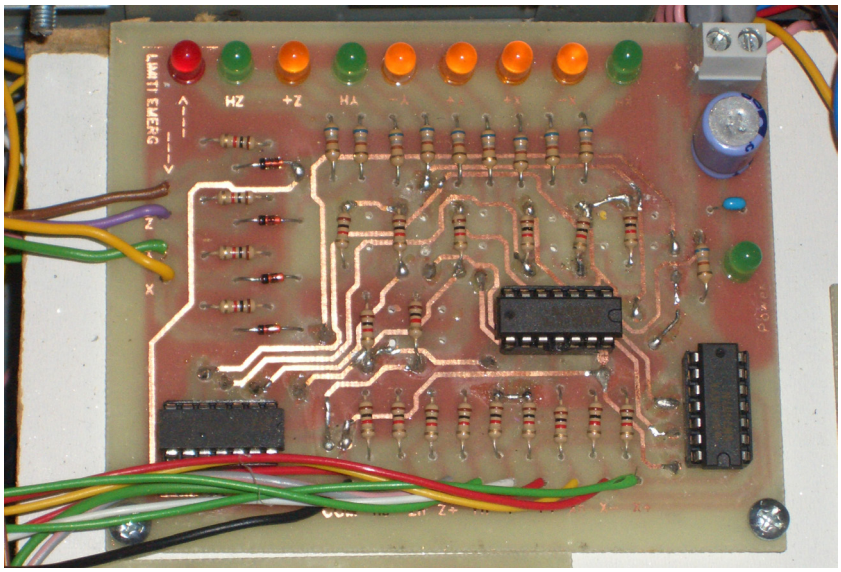
2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

⇨ Controllo limit switch e home switch

► Descrizione Generale



Il circuito disaccoppia il segnale di ciascun finecorsa o home switch posizionati su ogni asse, in modo da eliminare i possibili disturbi provenienti dai cavi dei motori, i quali viaggiano paralleli a quelli dei finecorsa fino alla macchina.

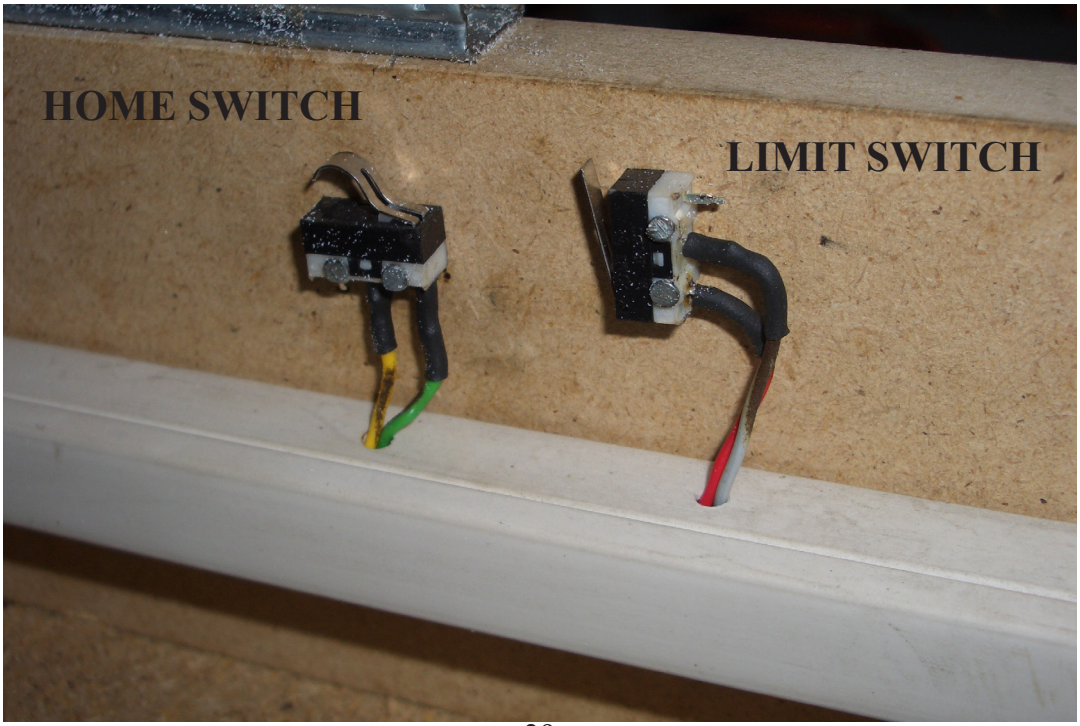
► Funzionamento

I segnali disaccopiati degli home switch vengono passati direttamente al PC, dove il software usa i medesimi segnali per referenziare la macchina a coordinate 0,0,0.

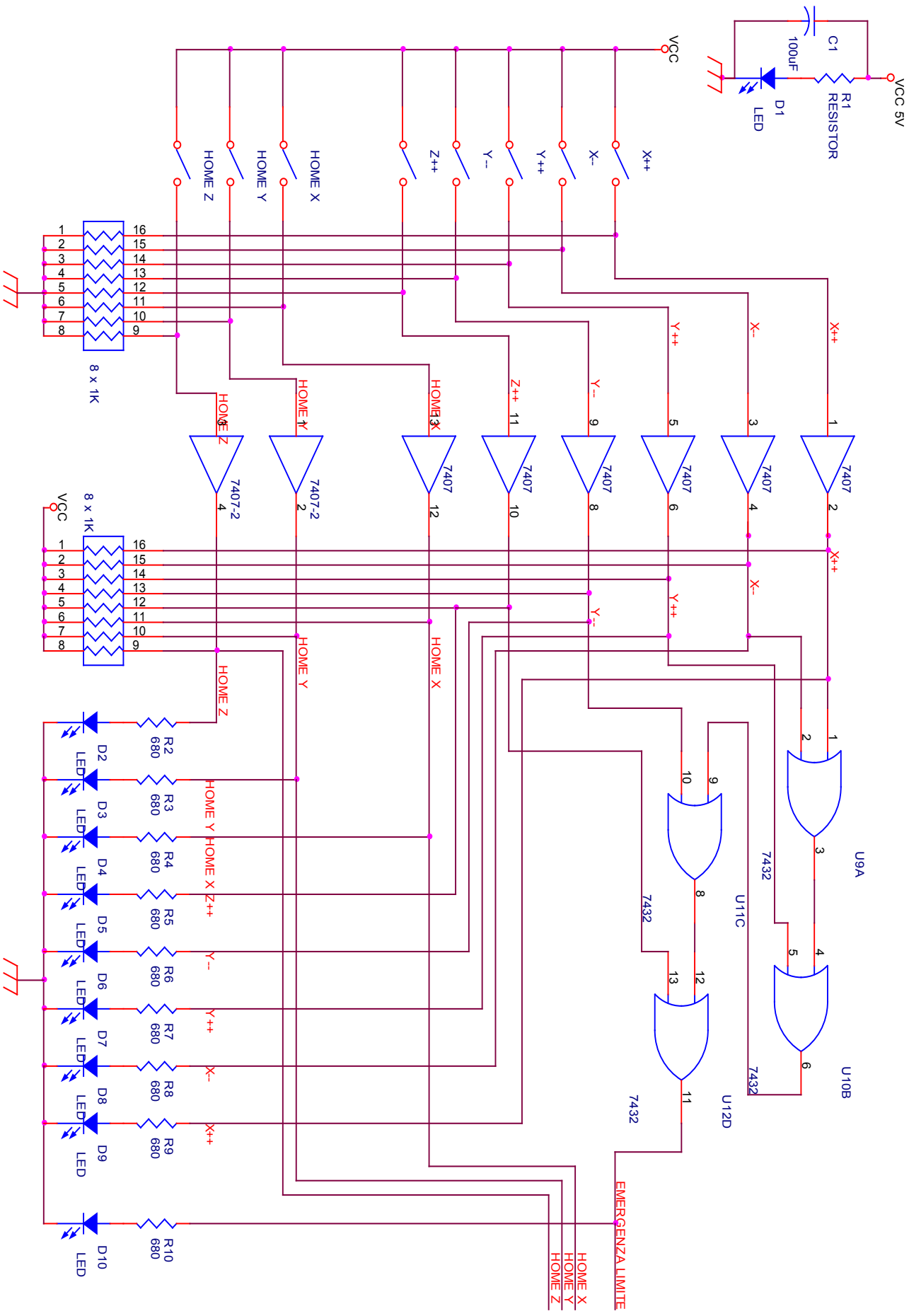
Invece i segnali provenienti dai finecorsa di limite, che sono due per l'asse X, due per l'asse Y e solamente uno per l'asse Z, vengono mandati tutti in OR logico tramite una rete logica(a porte OR), in modo che quando qualsiasi finecorsa di limite viene attivato venga a sua volta attivato un segnale unico di emergenza.

Ad ogni limit switch è collegato un led arancio, per ogni home switch è collegato un led verde mentre per l'uscita della rete a porte or, cioè l'emergenza generale relativa ai limit switch è collegato un led rosso.

Quindi ad ogni attivazione di qualsiasi switch sulla macchina comporterà l'accensione del relativo led di stato.

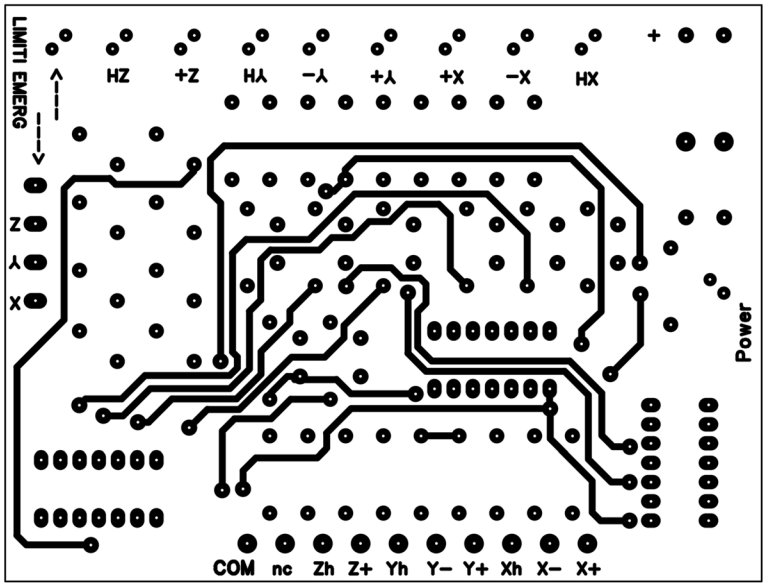


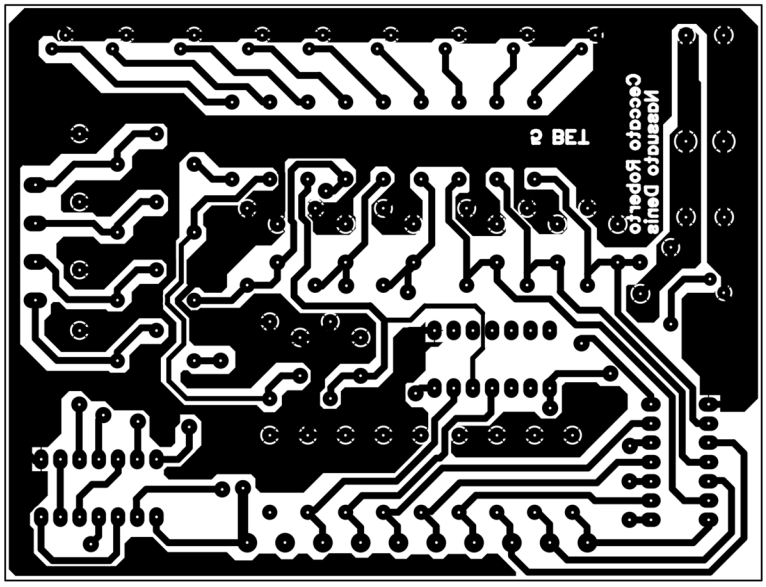
▶ SCHEMA ELETTRICO



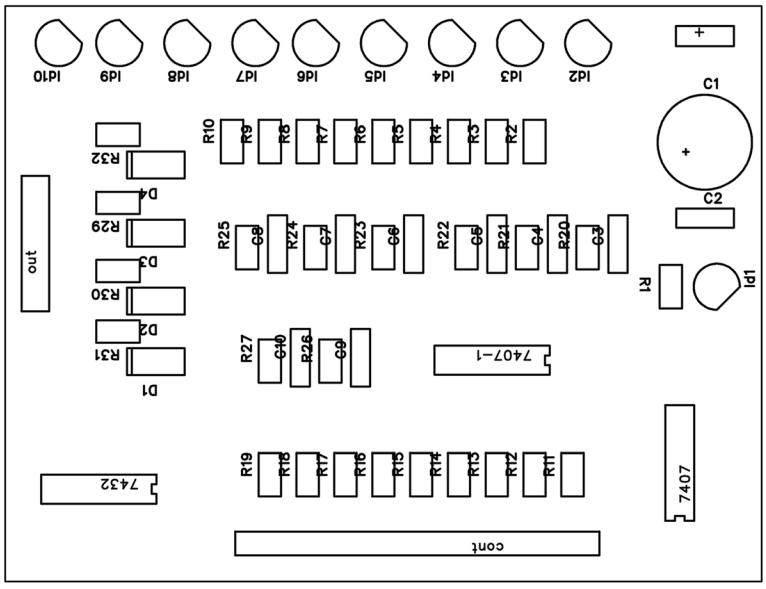
► PCB

1. TOP COPPER - Lato componenti





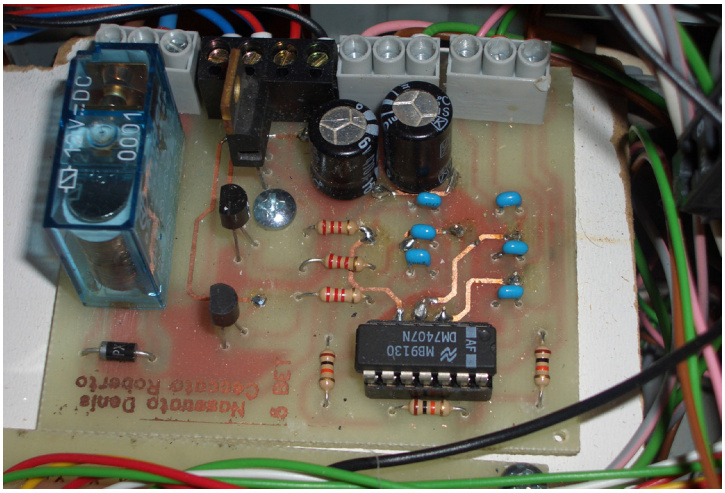
2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

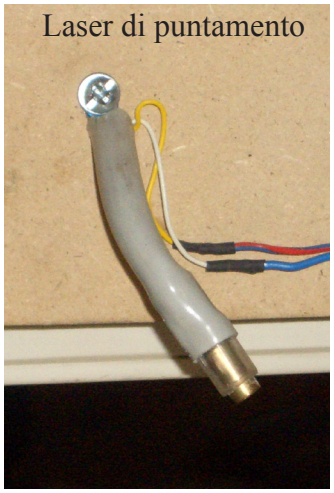
⇨ Potenza dispositivi luce, laser, fresa

► Descrizione Generale



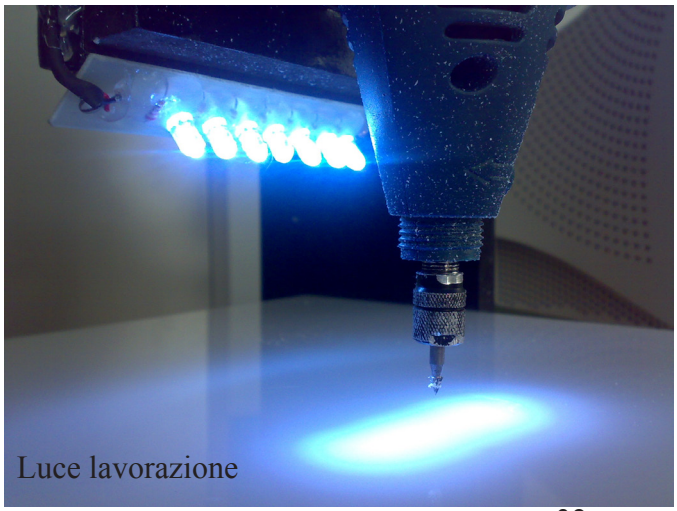
Il circuito è stato realizzato per riuscire a pilotare tramite ingressi digitali (TTL), 3 dispositivi (luce lavorazione, laser di puntamento e mandrino fresa), mediante l'utilizzo di un transistor (bc337) per i led ad alta luminosità della zona lavoro; un altro transistor per pilotare il laser di puntamento posizione utensile ed infine un terzo transistor (BDX53) per azionare un relè il quale provvede all'interruzione dell'alimentazione del mini trapano di precisione.

Per avere un'ampia illuminazione del piano di lavoro, sono stati collegati 7 led del tipo ad alta luminosità (10000 mcd) al di sotto dell'asse verticale Z.



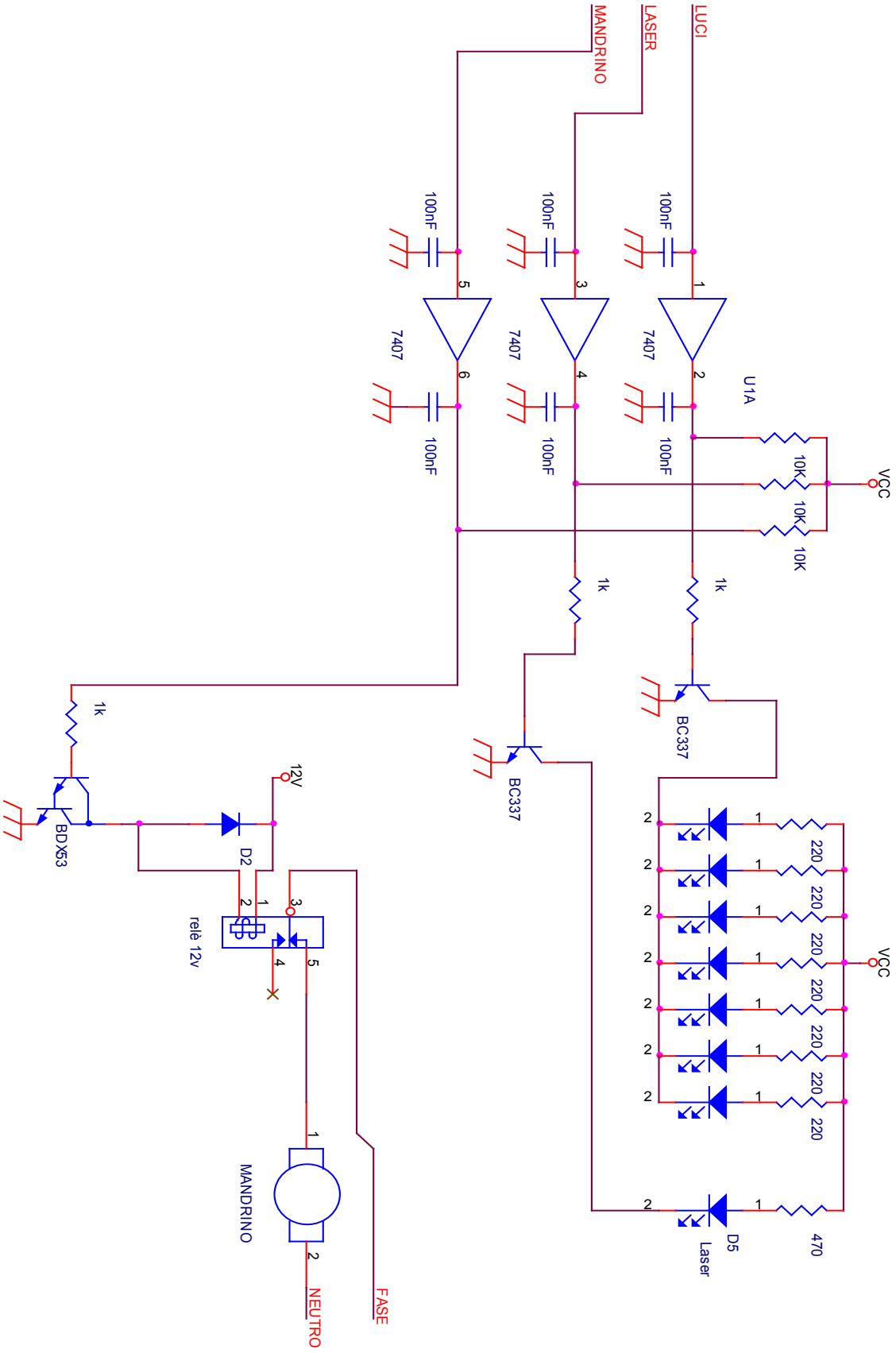
Laser di puntamento

Fresa

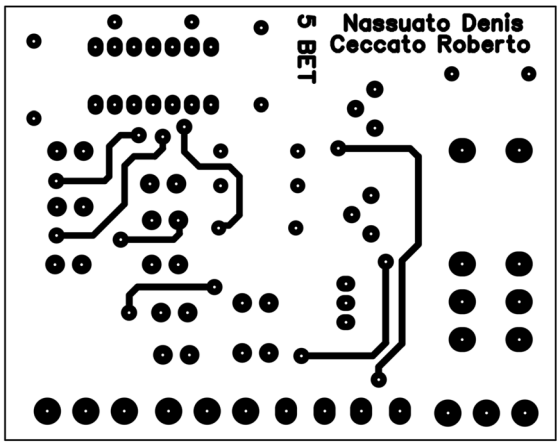


Luce lavorazione

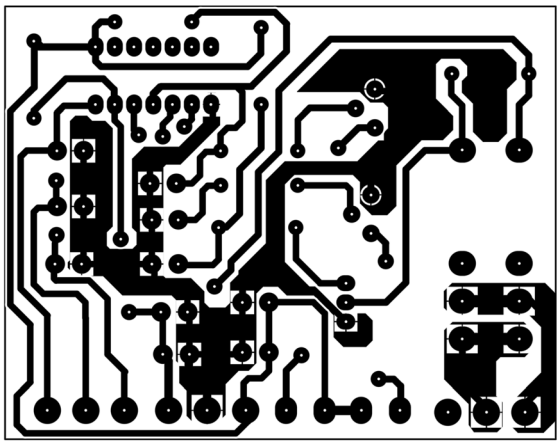
▶ Schema Elettrico



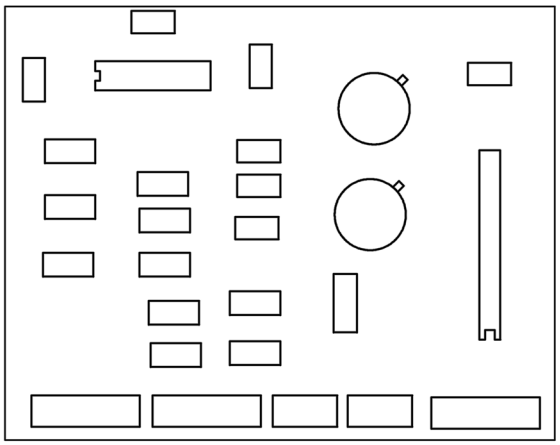
▶ PCB



1. TOP COPPER - Lato componenti



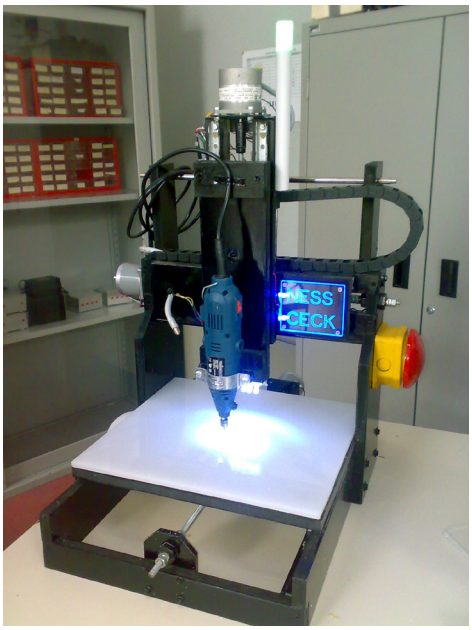
2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

⇨ Colonna di segnalazione visiva e segnalazione acustica

► Descrizione Generale



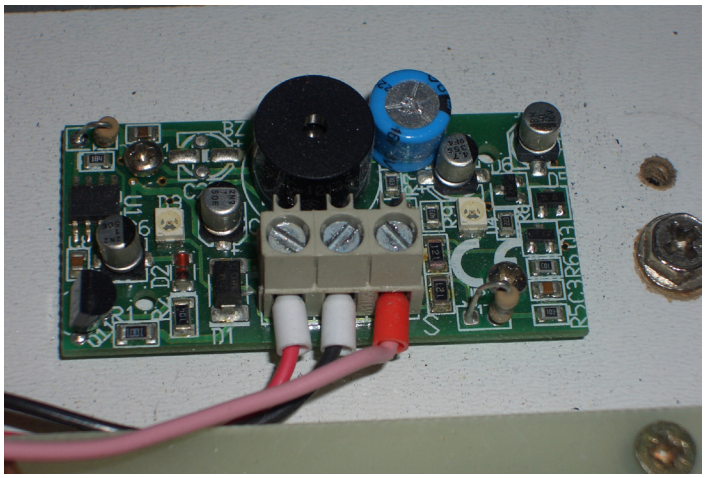
Le segnalazioni luminose e sonore sono un elemento fondamentale nella sicurezza di un impianto.

Per evitare errate interpretazioni è stata introdotta una normativa europea che attribuisce un significato univoco ai segnali visivi o sonori.

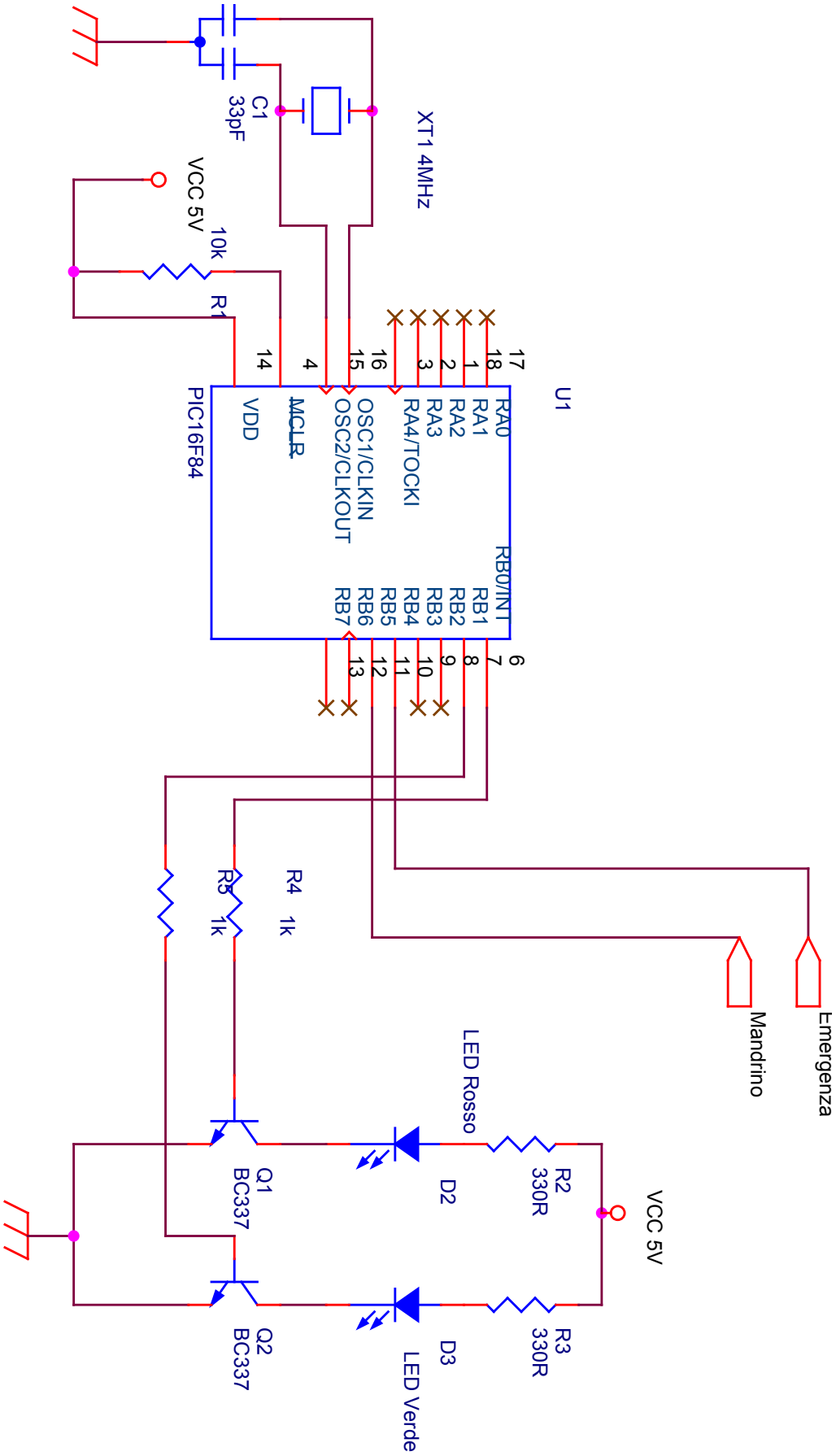
STATO MACCHINA OK

STATO MACCHINA EMERGENZA

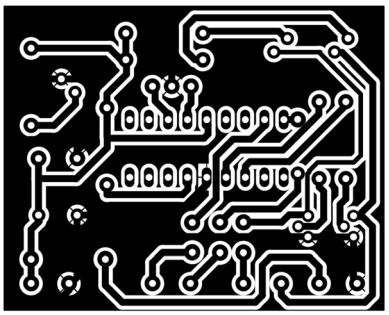
SIRENA ATTIVATA IN CASO DI EMRGENZA



▶ Schema Elettrico

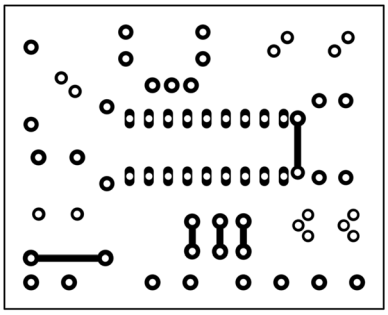


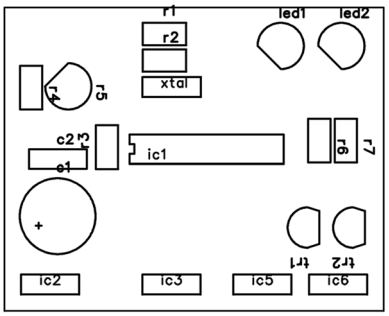
► PCB



1. BOTTOM COPPER - Lato rame

2. TOP COPPER - Lato componenti





3. SERIGRAFIA - Posizione componenti

```
;*******************************************************************************
;*                          CECCATO ROBERTO - NASSUATO DENIS                   *
;*                                                                             *
;*                                   5 BET                                     *
;*                                                                             *
;*                              SOFTWARE PIC 16F84A                            *
;*                  PER GESTIONE PULSANTIERA CONTROLLO CNC A 3 ASSI             *
;*                                                                             *
;*******************************************************************************

;       SOFTWARE DI GESTIONE COLONNINA LUMINOSA DI SEGNALAZIONE PER:

;       1- STATO MACCHINA OK (LED VERDE FISSO)
;       2- STATO MACCHINA OK E MANDRINO ATTIVO (LED VERDE LAMPEGGIANTE)
;       3- STATO MACCHINA EMERGENZA (LED ROSSO FISSO)
;       4- STATO MACCHINA EMERGENZA CON MADRINO
;          ANCORA ATTIVO (ROSSO/VERDE ALTERNATI)


;       TABELLA PIEDINATURA INGRESSI USCITE PIC

;       RA0   -->     NC
;       RA1   -->     NC
;       RA2   -->     NC
;       RA3   -->     NC
;       RA4   -->     NC
;       ------------------------------------------------
;       RB0   -->     SEGNALE DI ATTIVAZIONE LED ROSSO
;       RB1   -->     SEGNALE DI ATTIVAZIONE LED VERDE
;       RB2   -->     NC
;       RB3   -->     NC
;       RB4   -->     INGRESSO SEGNALE ATTIVAZIONE MANDRINO
;       RB5   -->     INGRESSO SEGNALE EMERGENZA MACCHINA
;       RB6   -->     NC
;       RB7   -->     NC
;       ------------------------------------------------


;----------------------- DIRETTIVE -------------------------

                PROCESSOR 16F84A
                RADIX DEC
                INCLUDE "P16F84A.INC"
                __CONFIG    3FF1H

                ;WatchDog Timer                         OFF
                ;Power Up Timer                         OFF
                ;Data Code Protect                      OFF
```

```
;---------------------- Registri temporanei -------------------------

Count  EQU        20H                  ;registro contatore delay
Count1 EQU        21H                  ;registro contatore delay

test   EQU        22H                  ;registro per il nà di ritardi da eseguire per attendere il
tempo
                                       ;minimo di attesa per l'avvio del test uscite

;----------------------------- Reset Vector -----------------------
;---------------------- Start point at CPU reset --------------------

           ORG     00H

;---------------------------- INIZIO -------------------------------

INIZIO     BSF     STATUS,RP0     ;Bank 1

           MOVLW   00H            ;imposto PORTA
           MOVWF   TRISA          ;con RA0:RA5 uscite
           MOVLW   0F0H           ;imposto PORTB
           MOVWF   TRISB          ;con RB0,RB1,RB3,RB7 uscite e RB2,RB4:RB6 in-
gressi

           BCF     STATUS,RP0     ;Bank 0

           CLRF    PORTB                ;uscite di PORTB a zero
           CLRF    PORTA                ;uscite di PORTA a zero


;--------------------------- PROGRAM START --------------------------

           BCF     PORTB,0        ;spengo led rosso se attivo
           BCF     PORTB,1        ;spengo led verde se attivo

MAIN       BTFSC   PORTB,5        ;testo ingresso emergenza
           GOTO    EME            ;se attivo salto al controllo mandrino
                                  ;e attivazione led rosso o rosso/verde
           BTFSS   PORTB,5        ;altrimenti testo nuovamente l'emergenza
           GOTO    V              ;se non è attivo salto al controllo mandrino
                                  ;e attivazione led verde o verde lampeggiante


           GOTO    MAIN           ;torna al programma principale MAIN LOOP

;---------------------- STATO EMERGENZA -----------------------

EME        BTFSC   PORTB,4        ;testo se è attivo il mandrino
           GOTO    RV             ;se si lampeggio rosso/verde
           BCF     PORTB,1        ;se no spengo led verde
           BSF     PORTB,0        ;accendo led rosso
           GOTO    MAIN           ;torna al MAIN LOOP
```

```
;--------------------------- STATO OK --------------------------

V       BTFSC   PORTB,4      ;testo se è attivo il mandrino
        GOTO    VL           ;se si lampeggio verde
        BCF     PORTB,0      ;se no spengo led rosso
        BSF     PORTB,1      ;accendo led verde
        GOTO    MAIN         ;torna al MAIN LOOP

;------------------- EMERGE E MANDRINO ATT ---------------------

RV      BCF     PORTB,0      ;spengo led rosso
        BCF     PORTB,1      ;spengo led verde
        BSF     PORTB,1      ;accendo led verde
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY        ;attendo circa 800ms
        CALL    DELAY
        BCF     PORTB,1      ;spengo led verde
        BSF     PORTB,0      ;accendo led rosso
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY        ;attendo circa 800ms
        CALL    DELAY
        BCF     PORTB,0      ;spengo led rosso
        GOTO    MAIN         ;torna al MAIN LOOP

;----------------------- MANDRINO ATTIVO ----------------------

VL      BCF     PORTB,0      ;spengo led rosso
        BCF     PORTB,1      ;accendo led verde
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY        ;attendo circa 800ms
        CALL    DELAY
        BSF     PORTB,1      ;accendo led verde
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY        ;attendo circa 800ms
        CALL    DELAY
        BCF     PORTB,1      ;spengo led verde
        GOTO    MAIN         ;torna al MAIN LOOP
```

```
;---------------------------- DELAY circa 200 ms--------------------------------

DELAY

        CLRF    Count
        CLRF    Count1

DELAYLOOP2

        DECFSZ  Count,1
        GOTO    DELAYLOOP2

        DECFSZ  Count1,1
        GOTO    DELAYLOOP2

                RETURN


;------------------------- FINE PROGRAMMA -----------------------------

                END
```

⇨ Pulsantiera di Controllo

► Descrizione Generale

La pulsantiera di controllo ha il compito di gestire un totale di 8 segnali di comando per la macchina. Essa è gestita internamente da un microcontrollore Microchip, un PIC 16F872. Il µC opera ad una frequenza di clock dettata dal quarzo a 4MHz.

I segnali nello specifico sono:

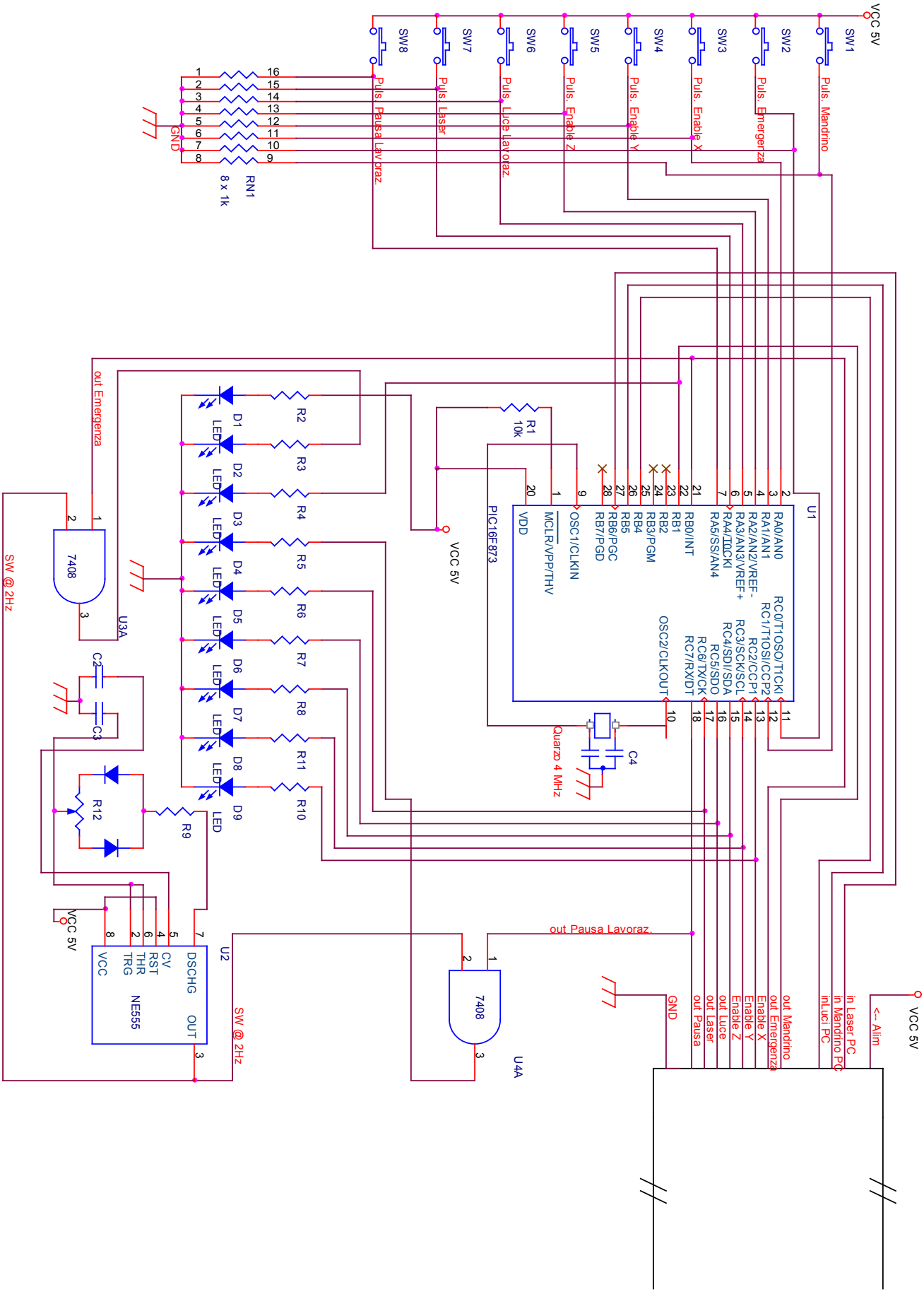
- Enable drive motore asse X
- Enable drive motore asse Y
- Enable drive motore asse Z
- Luce piano di lavorazione
- Laser di punatamento posizione utensile
- Mandrino fresa
- Segnale di pausa lavorazione
- Segnale di emergenza generale macchina

Ogni segnale può essere attivato o disattivato tramite il relativo pulsante, allo stesso tempo un led segnalerà lo stato della linea.

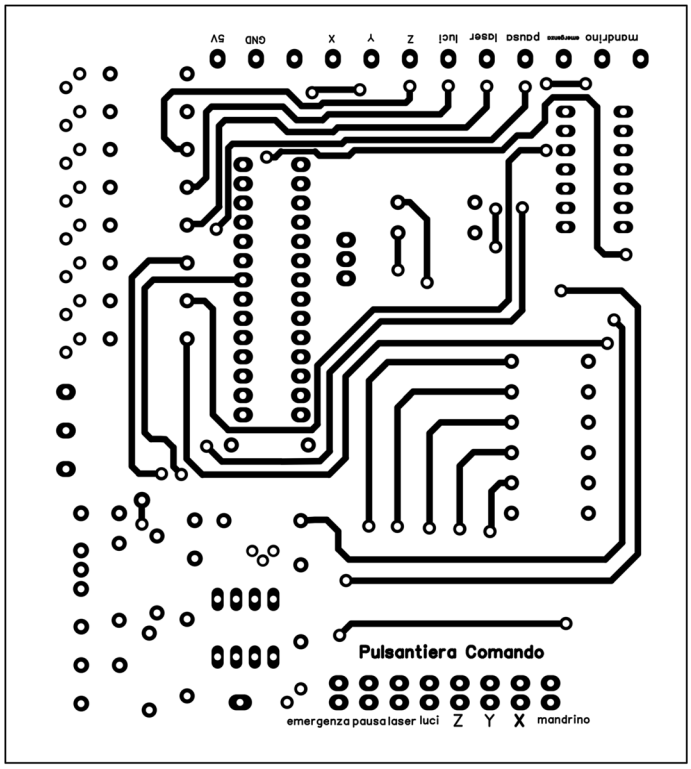
► Funzionamento

Nel momento in cui la macchina viene accesa il microcontrollore di gestione manterrà attivo il segnale di emergenza e non è possibile attivare nessun comando affinchè non verrà premuto il pulsante di emergenza posto sulla pulsantiera stessa o sulla macchina. A questo punto, dopo aver disattivato l'emergenza, verranno abilitati i 3 drive dei motori e inoltre sarà possibile gestire tutte le altre linee di controllo.

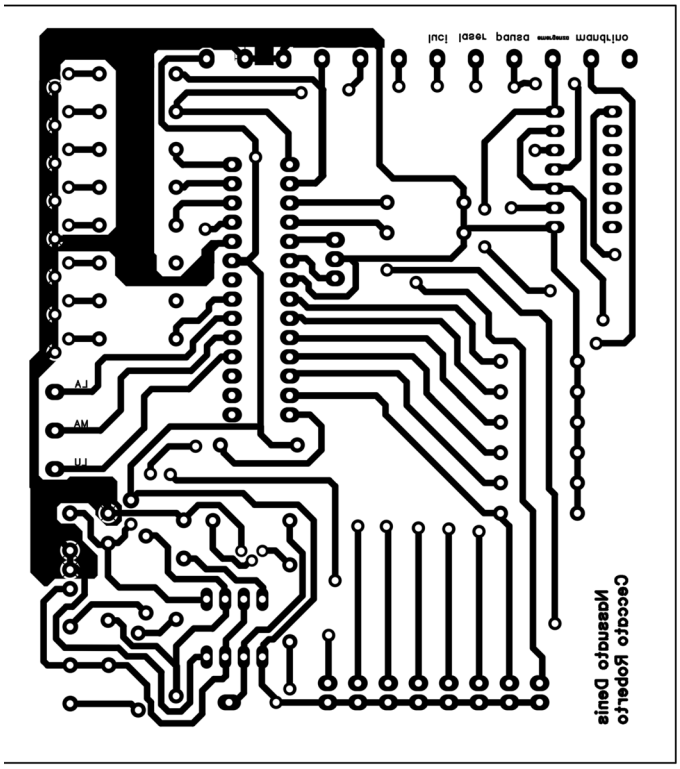
▶ SCHEMA ELETTRICO



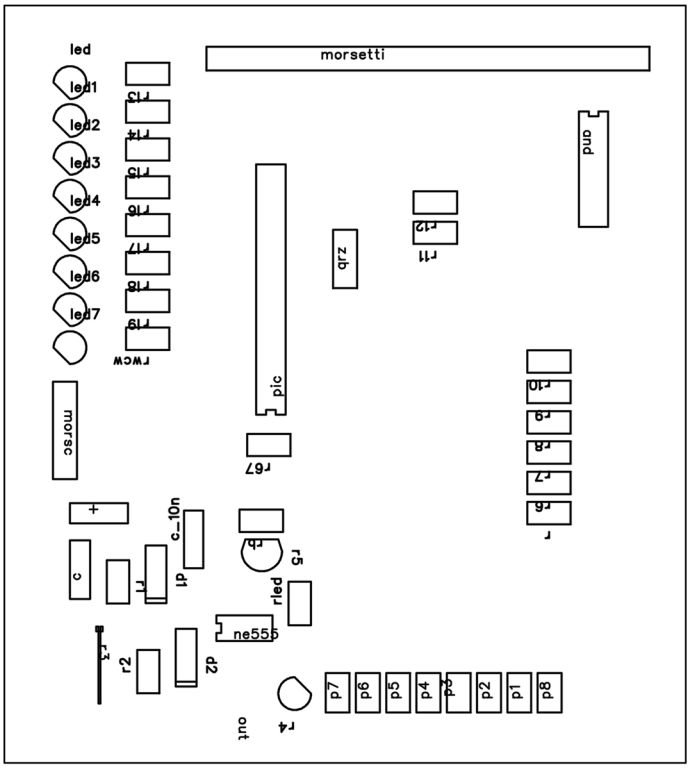
▶ PCB



1. TOP COPPER - Lato componenti



2. BOTTOM COPPER - Lato rame



3. SERIGRAFIA - Posizione componenti

```
;*******************************************************************************
;*                                                                             *
;*                          CECCATO ROBERTO - NASSUATO DENIS                   *
;*                                                                             *
;*                                                                             *
;*                                   5 BET                                     *
;*                                                                             *
;*                                                                             *
;*                          SOFTWARE PIC 16F87X (16F872)                       *
;*                  PER GESTIONE PULSANTIERA CONTROLLO CNC A 3 ASSI             *
;*                                                                             *
;*******************************************************************************

;       SOFTWARE DI GESTIONE PULSANTIERA MACCHINA CNC A 3 ASSI
;       LA PULSANTIERA COMANDA 8 SEGNALI DIVERSI PER IL CONTROLLO DI:

;       1- ABILITAZIONE SCHEDA MOTORE ASSE X
;       2- ABILITAZIONE SCHEDA MOTORE ASSE Y
;       3- ABILITAZIONE SCHEDA MOTORE ASSE Z
;       4- ON/OFF LUCE LAVORAZIONE TRAMITE PULSANTE O ESTERNO
;       5- ON/OFF LASER PUNTAMENTO POSIZIONE REALE UTENSILE SUL PIANO DI LAVORO
;       6- ON/OFF MANDRINO UTENSILE
;       7- ON/OFF SEGNALE DI PAUSA DURANTE LA LAVORAZIONE (USATO PER PIC O ALTRO)
;       8- SEGNALE FUNGO DI EMERGENZA

;       TABELLA PIEDINATURA INGRESSI USCITE PIC

;       RA0   ⇨   PULSANTE ENABLE ASSE X
;       RA1   ⇨   PULSANTE ENABLE ASSE Y
;       RA2   ⇨   PULSANTE ENABLE ASSE Z
;       RA3   ⇨   PULSANTE ON/OFF LUCI LAVORAZIONE
;       RA4   ⇨   PULSANTE ON/OFF LASER PUNTAMENTO
;       RA5   ⇨   PULSANTE ON/OFF PAUSA LAVORAZIONE
;       ------------------------------------------------
;       RB0   ⇨   USCITA SEGNALE EMERGENZA
;       RB1   ⇨   USCITA ON/OFF MANDRINO
;       RB2   ⇨   LIBERO
;       RB3   ⇨   LIBERO
;       RB4   ⇨   INGRESSO ESTERNO ON/OFF LUCE LAVORAZIONE x interrupt
;       RB5   ⇨   INGRESSO ESTERNO ON/OFF MANDRINO UTENSILE x interrupt
;       RB6   ⇨   INGRESSO ESTERNO ON/OFF LASER PUNTAMENTO x interrupt
;       RB7   ⇨   LIBERO
;       ------------------------------------------------
;       RC0   ⇨   PULSANTE EMERGNEZA
;       RC1   ⇨   PULSANTE ON/OFF MANDRINO UTENSILE
;       RC2   ⇨   USCITA ABILITAZIONE ASSE X
;       RC3   ⇨   USCITA ABILITAZIONE ASSE Y
;       RC4   ⇨   USCITA ABILITAZIONE ASSE Z
;       RC5   ⇨   USCITA ON/OFF LUCI LAVORAZIONE
;       RC6   ⇨   USCITA ON/OFF LASER PUNTAMENTO
;       RC7   ⇨   USCITA ON/OFF PAUSA LAVORAZIONE
;-------------------------------------------------
```

```
;----------------------- DIRETTIVE -------------------------

                PROCESSOR 16F872
                RADIX DEC
                INCLUDE "P16F872.INC"
                __CONFIG    3FF9H

                ;WatchDog Time                          OFF
                ;Power Up Timer                        ON
                ;Brown Out Reset                       ON
                ;Low Voltage Programming               ON
                ;Data Code Protect                     OFF
                ;Flash Program Memory Write Enable     ON
                ;Background Debugger Mode              OFF

;----------------------- Registri temporanei -------------------------

Count  EQU      20H             ;registro contatore delay
Count1 EQU      21H             ;registro contatore delay

test    EQU     22H             ;registro per il nà di ritardi da eseguire per attendere il tempo
                                ;minimo di attesa per l'avvio del test uscite
;------------------------------ Reset Vector -------------------------
;----------------------- Start point at CPU reset --------------------

                ORG     00H

                GOTO    INIZIO

;------------------ Start point at INTERRUPT EVENT -------------------

                ORG     04H

                BTFSS   INTCON,RBIF     ;verifico se è avvenuto l'interrupt
                                        ; sul cambiamento di livello RB4:RB7
                RETFIE

                BCF     INTCON,RBIF     ;se si:

                CALL    DELAY           ;attende circa 1 ms

                BTFSC   PORTB,6
                CALL    INT_LSR         ;gestione accensione laser da pc
                CALL    DELAY

                BTFSC   PORTB,5
                CALL    INT_MAND        ;gestione accensione mandrino da pc
                CALL    DELAY

                BTFSC   PORTB,4
                CALL    INT_LUCI        ;gestione accensione luce da pc
                CALL    DELAY
                RETFIE
```

```
;---------------------------- INIZIO -------------------------------

INIZIO      BSF         STATUS,RP0      ;Bank 1
            BCF         STATUS,RP1


            MOVLW       06H             ;Configura i pin di PORTA
            MOVWF       ADCON1          ;come ingressi digitali

            MOVLW       03FH            ;imposto PORTA  '00111111'
            MOVWF       TRISA           ;con RA0:RA5 ingressi

            MOVLW       074H            ;imposto PORTB  '01110100'
            MOVWF       TRISB           ;con RB0,RB1,RB3,RB7 uscite e RB2,RB4:RB6 in-
gressi

            MOVLW       003H            ;imposto PORTC  '00000011'
            MOVWF       TRISC           ;con RC0,RC1 ingressi e RC2:RC7 uscite

            BCF         STATUS,RP0
            BCF         STATUS,RP1      ;Bank 0

            CLRF        PORTB           ;uscite di PORTB a zero
            CLRF        PORTC           ;uscite di PORTC a zero

            BSF         INTCON,GIE      ;abilito interrupt generale
            BSF         INTCON,RBIE     ;abilito interrupt sul cambiamento di livello RB4:RB7

;-------------------------- PROGRAM START -------------------------

START       BSF         PORTB,0         ;attivo EMERGENZA
            CALL        DELAY_200ms     ;attende 200ms

WAIT        BTFSS       PORTC,0         ;rimane in attesa
            GOTO        WAIT            ;della pressione del tasto di emergenza
            CALL        EM              ;se viene premuto cambia stato all'uscita portandola a 0

            CALL  DELAY                 ;attende circa 1 ms

            BSF         PORTC,2         ;abilita l'asse X
            CALL        DELAY_200ms     ;attende 200ms
            BSF         PORTC,3         ;abilita l'asse Y
            CALL        DELAY_200ms     ;attende 200ms
            BSF         PORTC,4         ;abilita l'asse Z
            CALL        DELAY_200ms     ;attende 200ms

            GOTO        MAIN            ;salta al programma principale
                                        ;di gestione della pulsantiera
```

```
;---------------------------- MAIN ---------------------------------

MAIN
            BTFSC       PORTA,0         ;testa il pulsante di abilitazione asse X se premuto
            CALL        ENX             ;cambia di stato la relativa uscita
            BTFSC       PORTA,1         ;altrimenti testa il pulsante di abilitazione
                                        ;dell'asse Y se premuto
            CALL        ENY             ;cambia di stato la relativa uscita
            BTFSC       PORTA,2         ;altrimenti testa il pulsante di abilitazione
                                        ;dell'asse Z se premuto
            CALL        ENZ             ;cambia di stato la relativa uscita
            BTFSC       PORTA,3         ;altrimenti testa il pulsante di on/off luci se premuto
            CALL        LUCI            ;cambia di stato la relativa uscita
            BTFSC       PORTA,4         ;altrimenti testa il pulsante di on/off laser se premuto
            CALL        LASER           ;cambia di stato la relativa uscita
            BTFSC       PORTA,5         ;altrimenti testa il pulsante di pausa se premuto
            CALL        PAUSA           ;cambia di stato la relativa uscita
            BTFSC       PORTC,0         ;altrimenti testa il pulsante di emergenza se premuto
            CALL        EM              ;cambia di stato la relativa uscita
            BTFSC       PORTC,1         ;altrimenti testa il pulsante di
                                        ;on/off mandrino utensile se premuto
            CALL        MANDR           ;cambia di stato la relativa uscita

            GOTO        MAIN            ;ricomincia la sequenza di controllo pulsantiera

;------------------- SUBROUTINE GESTIONE SEGNALI ---------------------

;-------------------- abilitazione scheda asse X ---------------------

ENX         BTFSC       PORTA,0         ;se pulsante ancora premuto
            GOTO        ENX             ;rimane in attesa
            CALL        DELAY           ;altrimenti
            BTFSC       PORTC,2         ;controlla se l'uscita è a livello basso
            GOTO        ELSE_ENX        ;se no viene posta a 0 con salto all'else
            BSF         PORTC,2         ;se si viene posta a 1
            GOTO        ENDIF_ENX       ;salta alla fine
ELSE_ENX
            BCF         PORTC,2         ;else per la disattivazione dell'uscita se a valore 1
ENDIF_ENX
            CALL        DELAY

            RETURN                      ;ritorna la programma princiaple
```

```
;-------------------- abilitazione scheda asse Y ---------------------

ENY             BTFSC       PORTA,1             ;se pulsante ancora premuto
                GOTO        ENY                 ;rimane in attesa
                CALL        DELAY               ;altrimenti
                BTFSC       PORTC,3             ;controlla se l'uscita è a livello basso
                GOTO        ELSE_ENY            ;se no viene posta a 0 con salto all'else
                BSF         PORTC,3             ;se si viene posta a 1
                GOTO        ENDIF_ENY           ;salta alla fine
ELSE_ENY
                BCF         PORTC,3             ;else per la disattivazione dell'uscita se a valore 1
ENDIF_ENY
                CALL        DELAY

                RETURN                          ;ritorna la programma princiaple

;-------------------- Abilitazione scheda asse Y ---------------------

ENZ             BTFSC       PORTA,2             ;se pulsante ancora premuto
                GOTO        ENZ                 ;rimane in attesa
                CALL        DELAY               ;altrimenti
                BTFSC       PORTC,4             ;controlla se l'uscita è a livello basso
                GOTO        ELSE_ENZ            ;se no viene posta a 0 con salto all'else
                BSF         PORTC,4             ;se si viene posta a 1
                GOTO        ENDIF_ENZ           ;salta alla fine
ELSE_ENZ
                BCF         PORTC,4             ;else per la disattivazione dell'uscita se a valore 1
ENDIF_ENZ
                CALL        DELAY

                RETURN                          ;ritorna la programma princiaple

;------------------------ ON OFF LUCI -----------------------------

LUCI            BTFSC       PORTA,3             ;se pulsante ancora premuto
                GOTO        LUCI                ;rimane in attesa
                BTFSC       PORTB,4             ;se già attivo da pc esce
                RETURN                          ;tornando al programma pricipale
                CALL        DELAY               ;altrimenti
                BTFSC       PORTC,5             ;controlla se l'uscita è a livello basso
                GOTO        ELSE_LUCI           ;se no viene posta a 0 con salto all'else
                BSF         PORTC,5             ;se si viene posta a 1
                GOTO        ENDIF_LUCI          ;salta alla fine
ELSE_LUCI
                BCF         PORTC,5             ;else per la disattivazione dell'uscita se a valore 1
ENDIF_LUCI
                CALL        DELAY

                RETURN                          ;ritorna la programma princiaple
```

```
;------------------------ ON OFF LASER -----------------------------

LASER       BTFSC       PORTA,4         ;se pulsante ancora premuto
            GOTO        LASER           ;rimane in attesa
            BTFSC       PORTB,2         ;se già attivo da pc esce
            RETURN                      ;tornando al programma pricipale
            CALL        DELAY           ;altrimenti
            BTFSC       PORTC,6         ;controlla se l'uscita è a livello basso
            GOTO        ELSE_LSR        ;se no viene posta a 0 con salto all'else
            BSF         PORTC,6         ;se si viene posta a 1
            GOTO        ENDIF_LSR       ;salta alla fine
ELSE_LSR
            BCF         PORTC,6         ;else per la disattivazione dell'uscita se a valore 1
ENDIF_LSR
            CALL        DELAY

            RETURN                      ;ritorna la programma principle


;------------------------ ON OFF MANDRINO --------------------------

MANDR       BTFSC       PORTC,1         ;se pulsante ancora premuto
            GOTO        MANDR           ;rimane in attesa
            BTFSC       PORTB,6         ;se già attivo da pc esce
            RETURN                      ;tornando al programma pricipale
            CALL        DELAY           ;altrimenti
            BTFSC       PORTB,1         ;controlla se l'uscita è a livello basso
            GOTO        ELSE_MAN        ;se no viene posta a 0 con salto all'else
            BSF         PORTB,1         ;se si viene posta a 1
            GOTO        ENDIF_MAN       ;salta alla fine
ELSE_MAN
            BCF         PORTB,1         ;else per la disattivazione dell'uscita se a valore 1
ENDIF_MAN
            CALL        DELAY

            RETURN                      ;ritorna la programma principle


;------------------------ ATTIVAZIONE EMERGENZA -----------------------------

EM          BTFSC       PORTC,0         ;se pulsante ancora premuto
            GOTO        EM              ;rimane in attesa
            CALL        DELAY           ;altrimenti
            BTFSC       PORTB,0         ;controlla se l'uscita è a livello basso
            GOTO        ELSE_EM         ;se no viene posta a 0 con salto all'else
            BSF         PORTB,0         ;se si viene posta a 1
            GOTO        ENDIF_EM        ;salta alla fine
ELSE_EM
            BCF         PORTB,0         ;else per la disattivazione dell'uscita se a valore 1
ENDIF_EM
            CALL        DELAY

            RETURN                      ;ritorna la programma principle
```

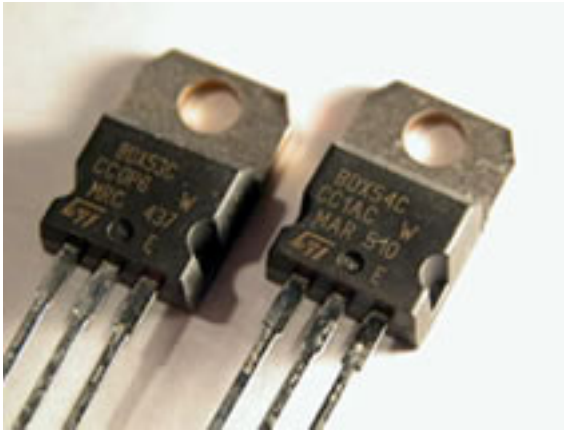
```
;------------------------ ON OFF PAUSA -----------------------------

PAUSA       BTFSC       PORTA,5             ;se pulsante ancora premuto
            GOTO        PAUSA               ;rimane in attesa
            CALL        DELAY               ;altrimenti
            BTFSC       PORTC,7             ;controlla se l'uscita è a livello basso
            GOTO        ELSE_PAUSA          ;se no viene posta a 0 con salto all'else
            BSF         PORTC,7             ;se si viene posta a 1
            GOTO        ENDIF_PAUSA         ;salta alla fine
ELSE_PAUSA
            BCF         PORTC,7             ;else per la disattivazione dell'uscita se a valore 1
ENDIF_PAUSA
            CALL        DELAY

            RETURN                          ;ritorna la programma principle

;-------------------- ON OFF LUCI FROM INTERRUPT --------------------

INT_LUCI
            BTFSS       PORTB,4             ;controlla se l'ingresso è a livello alto
            GOTO        ELSE_EXT_LUCI       ;se no viene posta a 0 con salto all'else
            BSF         PORTC,5             ;se si viene posta a 1
            GOTO        ENDIF_EXT_LUCI      ;salta alla fine
ELSE_EXT_LUCI
            BCF         PORTC,5             ;else per la disattivazione dell'uscita
ENDIF_EXT_LUCI

            RETURN

;------------------ ON OFF MANDRINO FROM INTERRUPT ------------------

INT_MAND
            BTFSS       PORTB,5             ;controlla se l'ingresso è a livello alto
            GOTO        ELSE_EXT_MAND       ;se no viene posta a 0 con salto all'else
            BSF         PORTB,1             ;se si viene posta a 1
            GOTO        ENDIF_EXT_MAND;salta alla fine
ELSE_EXT_MAND
            BCF         PORTB,1             ;else per la disattivazione dell'uscita
ENDIF_EXT_MAND

            RETURN

;------------------- ON OFF LASER FROM INTERRUPT -------------------

INT_LSR
            BTFSS       PORTB,6             ;controlla se l'ingresso è a livello alto
            GOTO        ELSE_EXT_LSR        ;se no viene posta a 0 con salto all'else
            BSF         PORTC,6             ;se si viene posta a 1
            GOTO        ENDIF_EXT_LSR       ;salta alla fine
ELSE_EXT_LSR
            BCF         PORTC,6             ;else per la disattivazione dell'uscita
ENDIF_EXT_LSR
            RETURN
```

```
;---------------------------- DELAY 0.765 ms--------------------------------

DELAY
        CLRF        Count
        CLRF        Count1

DELAYLOOP

        DECFSZ      Count,1
        GOTO        DELAYLOOP

        ;DECFSZ     Count1,1
        ;GOTO       DELAYLOOP

        RETURN

;---------------------------- DELAY circa 200 ms--------------------------------

DELAY_200ms

        CLRF        Count
        CLRF        Count1

DELAYLOOP2

        DECFSZ      Count,1
        GOTO        DELAYLOOP2

        DECFSZ      Count1,1
        GOTO        DELAYLOOP2

        RETURN


;------------------------- FINE PROGRAMMA -----------------------------

                END
```

DATASHEET

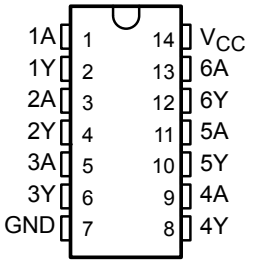
DEI PRINCIPALI COMPONENTI UTILIZZATI

- 7407
- 7408
- 7432
- L297
- BDX 53
- PIC 16F872
- PIC 16F84A

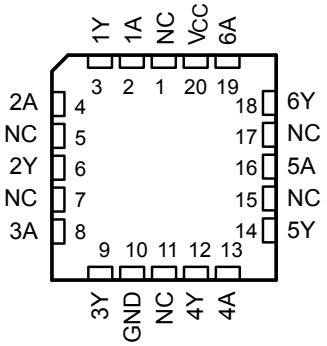


- **Converts TTL-Voltage Levels to MOS Levels**
- **High Sink-Current Capability**
- **Input Clamping Diodes Simplify System Design**
- **Open-Collector Driver for Indicator Lamps and Relays**
- **Package Options Include "Small Outline" Packages, Ceramic Chip Carriers, and Standard and Ceramic 300-mil DIPs**

SN54LS07 . . . J PACKAGE
SN74LS07, SN74LS17 . . . D OR N PACKAGE
(T0P VIEW)



SN54LS07 . . . FK PACKAGE
(T0P VIEW)



NC – No internal connection

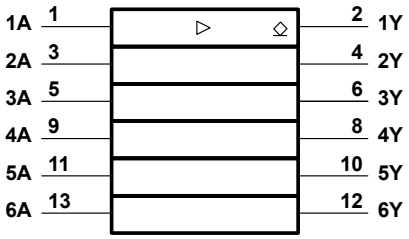
description

These monolithic hex buffers/drivers feature high-voltage open-collector outputs to interface with high-level circuits or for driving high-current loads. They are also characterized for use as buffers for driving TTL inputs. The ′LS07 has a rated output voltage of 30 V and the ′LS17 has a rated output voltage of 15 V. The maximum sink current is 30 mA for the SN54LS07 and 40 mA for the SN74LS07 and SN74LS17.

These circuits are compatible with most TTL families. Inputs are diode-clamped to minimize transmission-line effects, which simplifies design. Typical power dissipation is 140 mW and average propagation delay time is 12 ns.

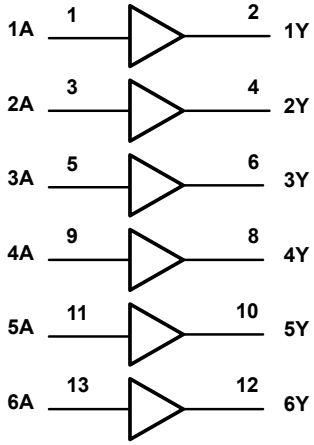
The SN54LS07 is characterized over the full military temperature range of –55°C to 125°C. The SN74LS07 and SN74LS17 are characterized for operation from 0°C to 70°C.

logic symbol†

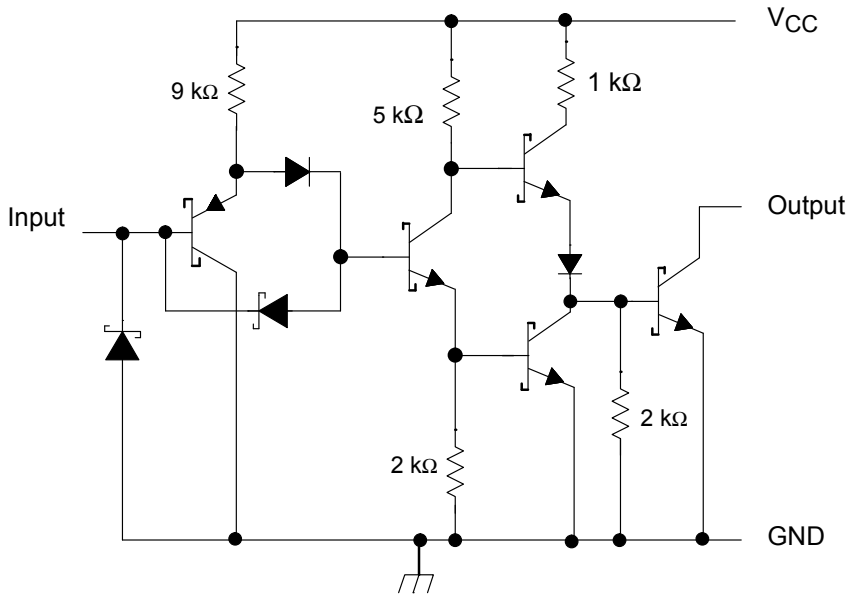


† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for D, J, and N packages.

logic diagram (positive logic)



schematic (each gate)



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC} . . . . . 7 V
Input voltage, V_I (see Note 1) . . . . . 5.5 V
Output voltage, V_O (see Notes 1 and 2): SN54LS07, SN74LS07 . . . . . 30 V
SN74LS17 . . . . . 15 V
Operating free-air temperature range: SN54LS07 . . . . . −55°C to 125°C
SN54LS07, SN74LS17 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Voltage values are with respect to network ground terminal.
2. This is the maximum voltage that should be applied to any output when it is in the off state.

recommended operating conditions

			SN54LS07			SN74LS07 SN74LS17			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage		2			2			V
V_{IL}	Low-level input voltage				0.8			0.8	V
V_{OH}	High-level output voltage	'LS07			30			30	V
		'LS17						15	
I_{OL}	Low-level output current				30			40	mA
T_A	Operating free-air temperature		−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†		SN54LS07			SN74LS07 SN74LS17			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	V_{CC} = MIN, I_I = –12 mA				–1.5			–1.5	V
I_{OH}	V_{CC} = MIN, V_{IH} = 2 V	'LS07, V_{OH} = 30 V			0.25			0.25	mA
		'LS17, V_{OH} = 15 V			0.25			0.25	
V_{OL}	V_{CC} = MIN, V_{IL} = 0.8 V	I_{OL} = 16 mA			0.4			0.4	V
		I_{OL} = MAX§			0.7			0.7	
I_I	V_{CC} = MAX, V_I = 7 V				1			1	mA
I_{IH}	V_{CC} = MAX, V_I = 2.4 V				20			20	μA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V				–0.2			–0.2	mA
I_{CCH}	V_{CC} = MAX				14			14	mA
I_{CCL}	V_{CC} = MAX				45			45	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ I_{OL} = 30 mA for SN54 series parts and 40 mA for SN74 series parts.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	R_L = 110 Ω, C_L = 15 pF		6	10	ns
t_{PHL}					19	30	

NOTE 3: Load circuit and voltage waveforms are shown in Section 1 of *TTL Logic Data Book,* 1988.

National Semiconductor

March 1993

54AC/74AC08 • 74ACT08
Quad 2-Input AND Gate

General Description

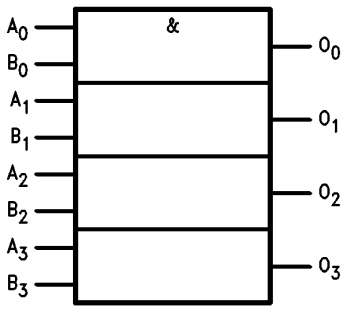
The 'AC/'ACT08 contains four, 2-input AND gates.

Features

- I_{CC} reduced by 50% on 54AC/74AC only
- Outputs source/sink 24 mA
- 'ACT08 has TTL-compatible inputs
- Standard Military Drawing (SMD)
  — 'AC08: 5962-87615
- For Military 54ACT08 device, see 54ACTQ08

Logic Symbols

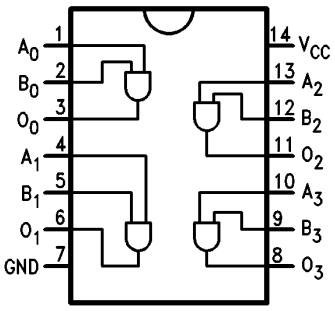
IEEE/IEC



TL/F/9914–1

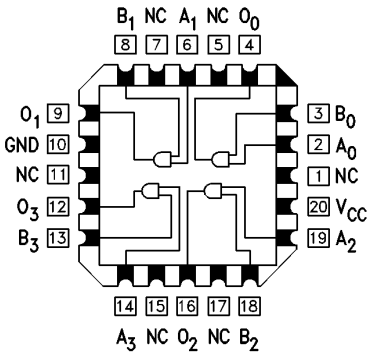
Connection Diagrams

Pin Assignment for DIP, Flatpak and SOIC



TL/F/9914–3

Pin Assignment for LCC



TL/F/9914–2

Pin Names	Description
A_n, B_n	Inputs
O_n	Outputs

FACT™ is a trademark of National Semiconductor Corporation.

Absolute Maximum Rating (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Diode Current (I_{IK})	
V_I = −0.5V	−20 mA
V_I = V_{CC} + 0.5V	+20 mA
DC Input Voltage (V_I)	−0.5V to V_{CC} + 0.5V
DC Output Diode Current (I_{OK})	
V_O = −0.5V	−20 mA
V_O = V_{CC} + 0.5V	+20 mA
DC Output Voltage (V_O)	−0.5V to V_{CC} + 0.5V
DC Output Source or Sink Current (I_O)	±50 mA
DC V_{CC} or Ground Current per Output Pin (I_{CC} or I_{GND})	±50 mA
Storage Temperature (T_{STG})	−65°C to +150°C
Junction Temperature (T_J)	
CDIP	175°C
PDIP	140°C

Note 1: Absolute maximum ratings are those values beyond which damage to the device may occur. The databook specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. National does not recommend operation of FACT™ circuits outside databook specifications.

Recommended Operating Conditions

Supply Voltage (V_{CC})	
'AC	2.0V to 6.0V
'ACT	4.5V to 5.5V
Input Voltage (V_I)	0V to V_{CC}
Output Voltage (V_O)	0V to V_{CC}
Operating Temperature (T_A)	
74AC/ACT	−40°C to +85°C
54AC	−55°C to +125°C
Minimum Input Edge Rate (ΔV/Δt) 'AC Devices V_{IN} from 30% to 70% of V_{CC} V_{CC} @ 3.3V, 4.5V, 5.5V	125 mV/ns
Minimum Input Edge Rate (ΔV/Δt) 'ACT Devices V_{IN} from 0.8V to 2.0V V_{CC} @ 4.5V, 5.5V	125 mV/ns

DC Characteristics for 'AC Family Devices

Symbol	Parameter	V_{CC} (V)	74AC T_A = +25°C	74AC T_A = −55°C to +125°C	54AC T_A = −40°C to +85°C	74AC T_A = −40°C to +85°C	Units	Conditions
			Typ	Guaranteed Limits				
V_{IH}	Minimum High Level Input Voltage	3.0	1.5	2.1	2.1	2.1	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
		4.5	2.25	3.15	3.15	3.15		
		5.5	2.75	3.85	3.85	3.85		
V_{IL}	Maximum Low Level Input Voltage	3.0	1.5	0.9	0.9	0.9	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
		4.5	2.25	1.35	1.35	1.35		
		5.5	2.75	1.65	1.65	1.65		
V_{OH}	Minimum High Level Output Voltage	3.0	2.99	2.9	2.9	2.9	V	I_{OUT} = −50 μA
		4.5	4.49	4.4	4.4	4.4		
		5.5	5.49	5.4	5.4	5.4		
		3.0		2.56	2.4	2.46	V	*V_{IN} = V_{IL} or V_{IH} I_{OH} −12 mA
		4.5		3.86	3.7	3.76		I_{OH} −24 mA
		5.5		4.86	4.7	4.76		−24 mA
V_{OL}	Maximum Low Level Output Voltage	3.0	0.002	0.1	0.1	0.1	V	I_{OUT} = 50 μA
		4.5	0.001	0.1	0.1	0.1		
		5.5	0.001	0.1	0.1	0.1		
		3.0		0.36	0.5	0.44	V	*V_{IN} = V_{IL} or V_{IH} I_{OL} 12 mA
		4.5		0.36	0.5	0.44		I_{OL} 24 mA
		5.5		0.36	0.5	0.44		24 mA
I_{IN}	Maximum Input Leakage Current	5.5		±0.1	±1.0	±1.0	μA	V_I = V_{CC}, GND

*All outputs loaded; thresholds on input associated with output under test.

DC Characteristics for 'AC Family Devices (Continued)

Symbol	Parameter	V_{CC} (V)	74AC $T_A = +25°C$		54AC $T_A = -55°C$ to $+125°C$	74AC $T_A = -40°C$ to $+85°C$	Units	Conditions
			Typ	Guaranteed Limits				
I_{OLD}	†Minimum Dynamic Output Current	5.5			50	75	mA	V_{OLD} = 1.65V Max
I_{OHD}		5.5			−50	−75	mA	V_{OHD} = 3.85V Min
I_{CC}	Maximum Quiescent Supply Current	5.5		2.0	40.0	20.0	μA	$V_{IN} = V_{CC}$ or GND

†Maximum test duration 2.0 ms, one output loaded at a time.

Note: I_{IN} and I_{CC} @ 3.0V are guaranteed to be less than or equal to the respective limit @ 5.5V V_{CC}.
I_{CC} for 54AC @ 25°C is identical to 74AC @ 25°C.

DC Characteristics for 'ACT Family Devices

Symbol	Parameter	V_{CC} (V)	74ACT $T_A = +25°C$		74ACT $T_A = -40°C$ to $+85°C$	Units	Conditions
			Typ	Guaranteed Limits			
V_{IH}	Minimum High Level Input Voltage	4.5 5.5	1.5 1.5	2.0 2.0	2.0 2.0	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
V_{IL}	Maximum Low Level Input Voltage	4.5 5.5	1.5 1.5	0.8 0.8	0.8 0.8	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
V_{OH}	Minimum High Level Output Voltage	4.5 5.5	4.49 5.49	4.4 5.4	4.4 5.4	V	I_{OUT} = −50 μA
		4.5 5.5		3.86 4.86	3.76 4.76	V	*V_{IN} = V_{IL} or V_{IH} I_{OH} −24 mA −24 mA
V_{OL}	Maximum Low Level Output Voltage	4.5 5.5	0.001 0.001	0.1 0.1	0.1 0.1	V	I_{OUT} = 50 μA
		4.5 5.5		0.36 0.36	0.44 0.44	V	*V_{IN} = V_{IL} or V_{IH} I_{OL} 24 mA 24 mA
I_{IN}	Maximum Input Leakage Current	5.5		±0.1	±1.0	μA	$V_I = V_{CC}$, GND
I_{CCT}	Maximum I_{CC}/Input	5.5	0.6		1.5	mA	$V_I = V_{CC}$ − 2.1V
I_{OLD}	†Minimum Dynamic Output Current	5.5			75	mA	V_{OLD} = 1.65V Max
I_{OHD}		5.5			−75	mA	V_{OHD} = 3.85V Min
I_{CC}	Maximum Quiescent Supply Current	5.5		4.0	40.0	μA	$V_{IN} = V_{CC}$ or GND

*All outputs loaded; thresholds on input associated with output under test.

†Maximum test duration 2.0 ms, one output loaded at a time.

National Semiconductor

June 1996

54AC/74AC32 • 74ACT32
Quad 2-Input OR Gate

General Description

The 'AC/'ACT32 contains four, 2-input OR gates.

Features

- I_{CC} reduced by 50% on 54AC/74AC only
- Outputs source/sink 24 mA
- 'ACT32 has TTL-compatible inputs
- Standard Military Drawing (SMD)
  — 'AC32: 5962-87614
- For Military 54ACT32, see the 54ACTQ32

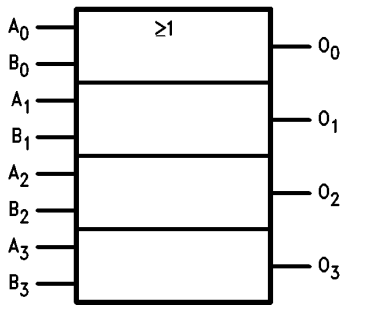
Commercial	Military	Package Number	Package Description
74ACT32PC		N14A	14-Lead Molded Dual-In-Line (0.300″ Wide)
74ACT32SC (Note 1)		M14A	14-Lead Molded Small Outline (0.150″ Wide), JEDEC
74ACT32MTC (Note 1)		MTC14	14-Lead Molded Thin Shrink Small Outline Package, JEDEC
	54ACT32DM (Note 2)	J14A	14-Lead Ceramic Dual-In-Line
	54ACT32FM (Note 2)	W14B	14-Lead Cerpak
	54ACT32LM (Note 2)	E20A	20-Lead Ceramic Leadless Chip Carrier, Type C

Note 1: Devices also available in 13″ Tape and Reel. Use suffix SCX, SJX, and MTCX.

Note 2: Military grade device with environmental and burn-in processing, use suffix DMQB, FMQB and LMQB.

Logic Symbol

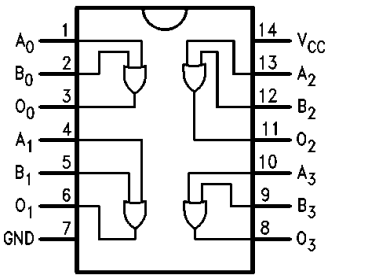
IEEE/IEC



TL/F/9919–1

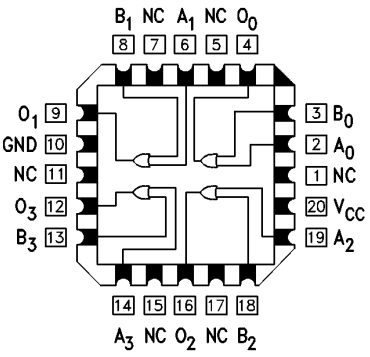
Connection Diagrams

Pin Assignment for DIP, Flatpak, SOIC and TSSOP



TL/F/9919–3

Pin Assignment for LCC



TL/F/9919–2

Pin Names	Description
A_n, B_n	Inputs
O_n	Outputs

FACT™ is a trademark of National Semiconductor Corporation.

Absolute Maximum Rating (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Diode Current (I_{IK})	
V_I = −0.5V	−20 mA
V_I = V_{CC} + 0.5V	+20 mA
DC Input Voltage (V_I)	−0.5V to V_{CC} + 0.5V
DC Output Diode Current (I_{OK})	
V_O = −0.5V	−20 mA
V_O = V_{CC} + 0.5V	+20 mA
DC Output Voltage (V_O)	−0.5V to V_{CC} + 0.5V
DC Output Source or Sink Current (I_O)	±50 mA
DC V_{CC} or Ground Current per Output Pin (I_{CC} or I_{GND})	±50 mA
Storage Temperature (T_{STG})	−65°C to +150°C
Junction Temperature (T_J)	
CDIP	175°C
PDIP	140°C

Note 1: Absolute maximum ratings are those values beyond which damage to the device may occur. The databook specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. National does not recommend operation of FACT™ circuits outside databook specifications.

Recommended Operating Conditions

Supply Voltage (V_{CC})	
'AC	2.0V to 6.0V
'ACT	4.5V to 5.5V
Input Voltage (V_I)	0V to V_{CC}
Output Voltage (V_O)	0V to V_{CC}
Operating Temperature (T_A)	
74AC/ACT	−40°C to +85°C
54AC	−55°C to +125°C
Minimum Input Edge Rate (ΔV/Δt) 'AC Devices V_{IN} from 30% to 70% of V_{CC} V_{CC} @ 3.3V, 4.5V, 5.5V	125 mV/ns
Minimum Input Edge Rate (ΔV/Δt) 'ACT Devices V_{IN} from 0.8V to 2.0V V_{CC} @ 4.5V, 5.5V	125 mV/ns

DC Characteristics for 'AC Family Devices

Symbol	Parameter	V_{CC} (V)	74AC T_A = +25°C	74AC / 54AC T_A = −55°C to +125°C	74AC T_A = −40°C to +85°C	Units	Conditions
			Typ	Guaranteed Limits			
V_{IH}	Minimum High Level Input Voltage	3.0	1.5	2.1	2.1	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
		4.5	2.25	3.15	3.15		
		5.5	2.75	3.85	3.85		
V_{IL}	Maximum Low Level Input Voltage	3.0	1.5	0.9	0.9	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
		4.5	2.25	1.35	1.35		
		5.5	2.75	1.65	1.65		
V_{OH}	Minimum High Level Output Voltage	3.0	2.99	2.9	2.9	V	I_{OUT} = −50 μA
		4.5	4.49	4.4	4.4		
		5.5	5.49	5.4	5.4		
		3.0		2.56	2.4	2.46	V	*V_{IN} = V_{IL} or V_{IH} I_{OH} −12 mA
		4.5		3.86	3.7	3.76		I_{OH} −24 mA
		5.5		4.86	4.7	4.76		−24 mA
V_{OL}	Maximum Low Level Output Voltage	3.0	0.002	0.1	0.1	0.1	V	I_{OUT} = 50 μA
		4.5	0.001	0.1	0.1	0.1		
		5.5	0.001	0.1	0.1	0.1		
		3.0		0.36	0.5	0.44	V	*V_{IN} = V_{IL} or V_{IH} I_{OL} 12 mA
		4.5		0.36	0.5	0.44		I_{OL} 24 mA
		5.5		0.36	0.5	0.44		24 mA
I_{IN}	Maximum Input Leakage Current	5.5		±0.1	±1.0	±1.0	μA	V_I = V_{CC}, GND

*All outputs loaded; thresholds on input associated with output under test.

DC Characteristics for 'AC Family Devices (Continued)

Symbol	Parameter	V_{CC} (V)	74AC $T_A = +25°C$		54AC $T_A = -55°C$ to $+125°C$	74AC $T_A = -40°C$ to $+85°C$	Units	Conditions
			Typ	Guaranteed Limits				
I_{OLD}	†Minimum Dynamic Output Current	5.5			50	75	mA	V_{OLD} = 1.65V Max
I_{OHD}		5.5			−50	−75	mA	V_{OHD} = 3.85V Min
I_{CC}	Maximum Quiescent Supply Current	5.5		2.0	40.0	20.0	μA	$V_{IN} = V_{CC}$ or GND

†Maximum test duration 2.0 ms, one output loaded at a time.

Note: I_{IN} and I_{CC} @ 3.0V are guaranteed to be less than or equal to the respective limit @ 5.5V V_{CC}.
I_{CC} for 54AC @ 25°C is identical to 74AC @ 25°C.

DC Characteristics for 'ACT Family Devices

Symbol	Parameter	V_{CC} (V)	74ACT $T_A = +25°C$		74ACT $T_A = -40°C$ to $+85°C$	Units	Conditions
			Typ	Guaranteed Limits			
V_{IH}	Minimum High Level Input Voltage	4.5 5.5	1.5 1.5	2.0 2.0	2.0 2.0	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
V_{IL}	Maximum Low Level Input Voltage	4.5 5.5	1.5 1.5	0.8 0.8	0.8 0.8	V	V_{OUT} = 0.1V or V_{CC} − 0.1V
V_{OH}	Minimum High Level Output Voltage	4.5 5.5	4.49 5.49	4.4 5.4	4.4 5.4	V	I_{OUT} = −50 μA
		4.5 5.5		3.86 4.86	3.76 4.76	V	*V_{IN} = V_{IL} or V_{IH} I_{OH} −24 mA I_{OH} −24 mA
V_{OL}	Maximum Low Level Output Voltage	4.5 5.5	0.001 0.001	0.1 0.1	0.1 0.1	V	I_{OUT} = 50 μA
		4.5 5.5		0.36 0.36	0.44 0.44	V	*V_{IN} = V_{IL} or V_{IH} I_{OL} 24 mA I_{OL} 24 mA
I_{IN}	Maximum Input Leakage Current	5.5		±0.1	±1.0	μA	$V_I = V_{CC}$, GND
I_{CCT}	Maximum I_{CC}/Input	5.5	0.6		1.5	mA	$V_I = V_{CC}$ − 2.1V
I_{OLD}	†Minimum Dynamic Output Current	5.5			75	mA	V_{OLD} = 1.65V Max
I_{OHD}		5.5			−75	mA	V_{OHD} = 3.85V Min
I_{CC}	Maximum Quiescent Supply Current	5.5		4.0	40.0	μA	$V_{IN} = V_{CC}$ or GND

*All outputs loaded; thresholds on input associated with output under test.

†Maximum test duration 2.0 ms, one output loaded at a time.

SGS-THOMSON MICROELECTRONICS

BDX53A/53B/53C
BDX54A/54B/54C

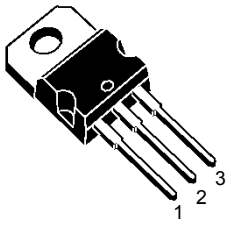
COMPLEMENTARY SILICON POWER DARLINGTON TRANSISTORS

- BDX53B, BDX53C, BDX54B AND BDX54C ARE SGS-THOMSON PREFERRED SALESTYPES

DESCRIPTION

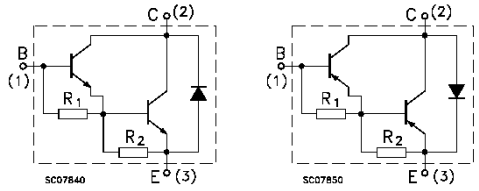
The BDX53A, BDX53B and BDX53C are silicon epitaxial-base NPN power transistors in monolithic Darlington configuration and are mounted in Jedec TO-220 plastic package. They are intented for use in hammer drivers, audio amplifiers and other medium power linear and switching applications.

The complementary PNP types are the BDX54A, BDX54B and BDX54C respectively.



TO-220

INTERNAL SCHEMATIC DIAGRAM



R_1 Typ. = 10 KΩ R_2 Typ. = 150 Ω

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter		Value			Unit
		NPN	BDX53A	BDX53B	BDX53C	
		PNP	BDX54A	BDX54B	BDX54C	
V_{CBO}	Collector-Base Voltage (I_E = 0)		60	80	100	V
V_{CEO}	Collector-Emitter Voltage (I_B = 0)		60	80	100	V
V_{EBO}	Emitter-base Voltage (I_C = 0)		5			V
I_C	Collector Current		8			A
I_{CM}	Collector Peak Current (repetitive)		12			A
I_B	Base Current		0.2			A
P_{tot}	Total Dissipation at $T_c \leq 25\ ^oC$		60			W
T_{stg}	Storage Temperature		-65 to 150			^oC
T_j	Max. Operating Junction Temperature		150			^oC

October 1995

THERMAL DATA

$R_{thj\text{-}case}$	Thermal Resistance Junction-case	Max	2.08	°C/W
$R_{thj\text{-}amb}$	Thermal Resistance Junction-ambient	Max	70	°C/W

ELECTRICAL CHARACTERISTICS (T_{case} = 25 °C unless otherwise specified)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
I_{CBO}	Collector Cut-off Current (I_E = 0)	for **BDX53A/54A** for **BDX53B/54B** for **BDX53C/54C**	V_{CB} = 60 V V_{CB} = 80 V V_{CB} = 100V			0.2 0.2 0.2	mA mA mA
I_{CEO}	Collector Cut-off Current (I_B = 0)	for **BDX53A/54A** for **BDX53B/54B** for **BDX53C/54C**	V_{CB} = 30 V V_{CB} = 40 V V_{CB} = 50V			0.5 0.5 0.5	mA mA mA
I_{EBO}	Emitter Cut-off Current (I_C = 0)	V_{EB} = 5 V				2	mA
$V_{CEO(sus)}$*	Collector-Emitter Sustaining Voltage (I_B = 0)	I_C = 100 mA	for **BDX53A/54A** for **BDX53B/53B** for **BDX53C/54C**	60 80 100			V V V
$V_{CE(sat)}$*	Collector-emitter Saturation Voltage	I_C = 3 A	I_B =12 mA			2	V
$V_{BE(sat)}$*	Base-emitter Saturation Voltage	I_C = 3 A	I_B =12 mA			2.5	V
h_{FE}*	DC Current Gain	I_C = 3 A	V_{CE} = 3 V	750			
V_F*	Parallel-diode Forward Voltage	I_F = 3 A I_F = 8 A			1.8 2.5	2.5	V V

* Pulsed: Pulse duration = 300 μs, duty cycle 1.5 %
For PNP types voltage and current values are negative.

PIC16F8X

18-pin Flash/EEPROM 8-Bit Microcontrollers

Devices Included in this Data Sheet:

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84
- Extended voltage range devices available (PIC16**LF**8X, PIC16**LCR**8X)

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single cycle except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle

Device	Program Memory (words)	Data RAM (bytes)	Data EEPROM (bytes)	Max. Freq (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

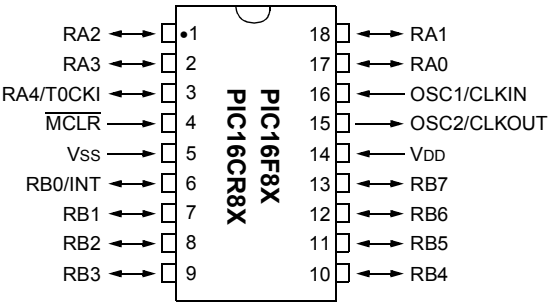
- 14-bit wide instructions
- 8-bit wide data path
- 15 special function hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on change
  - Data EEPROM write complete
- 1000 erase/write cycles Flash program memory
- 10,000,000 erase/write cycles EEPROM data memory
- EEPROM Data Retention > 40 years

Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagrams

PDIP, SOIC



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) - via two pins (ROM devices support only Data EEPROM programming)
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Flash/EEPROM Technology:

- Low-power, high-speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 6.0V
  - Industrial: 2.0V to 6.0V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 2V, 32 kHz
  - < 1 μA typical standby current @ 2V

PIC16CXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register), and the other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram for the PIC16F8X is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

FIGURE 3-1: PIC16F8X BLOCK DIAGRAM

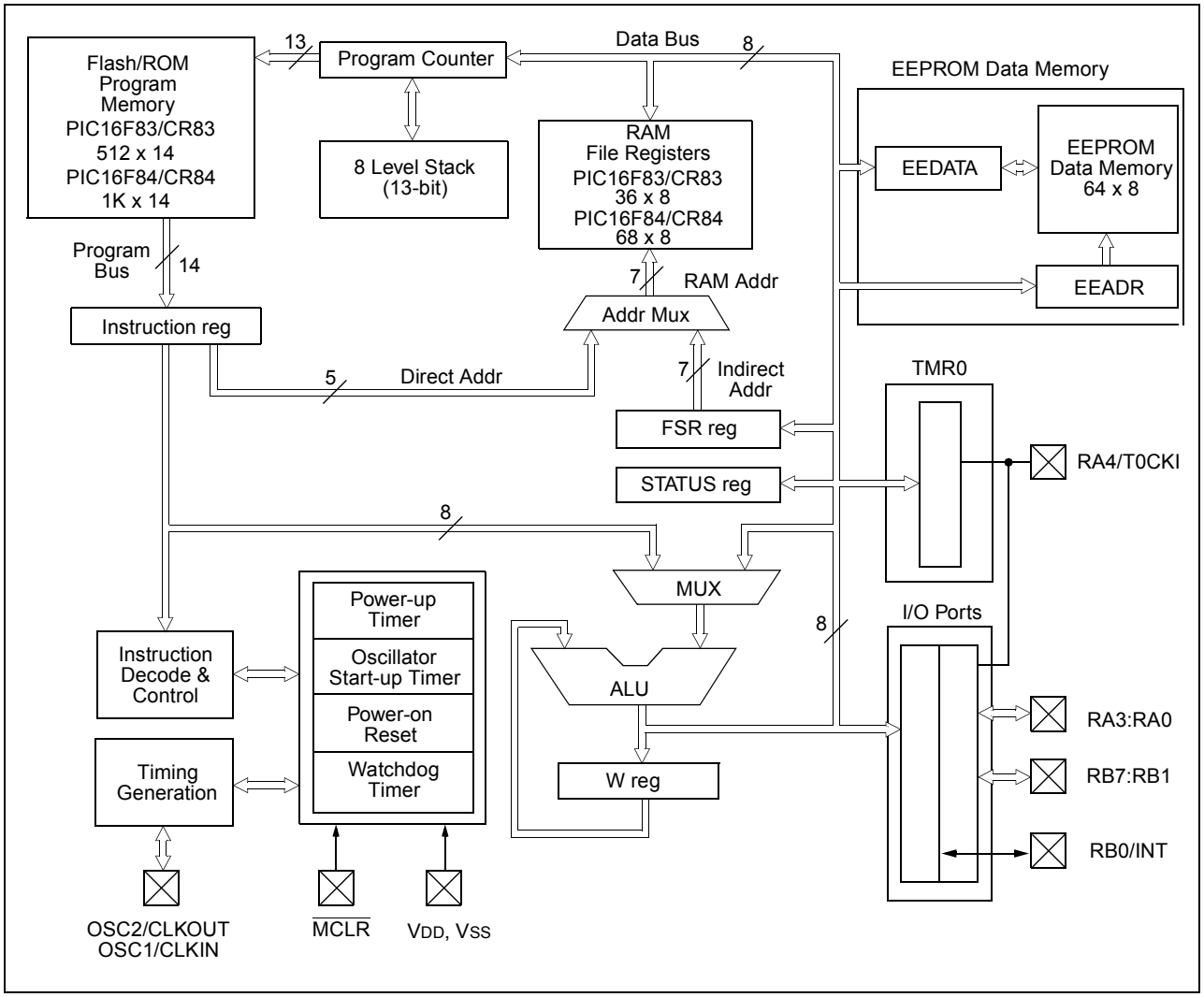


TABLE 3-1 PIC16F8X PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS (3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	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.
$\overline{MCLR}$	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0	17	17	I/O	TTL	
RA1	18	18	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL/ST (1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST (2)	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST (2)	Interrupt on change pin. Serial programming data.
VSS	5	5	P	—	Ground reference for logic and I/O pins.
VDD	14	14	P	—	Positive supply for logic and I/O pins.

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 the 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 in RC oscillator mode and a CMOS input otherwise.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

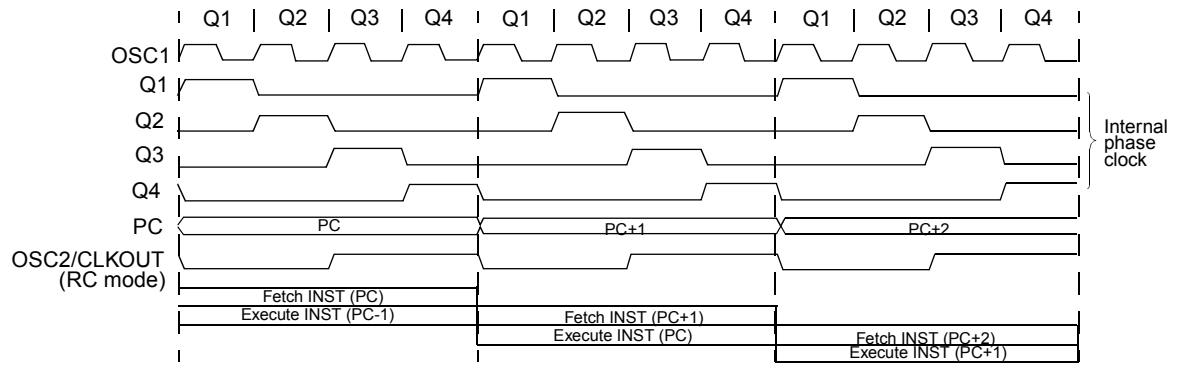
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`) then two cycles are required to complete the instruction (Example 3-1).

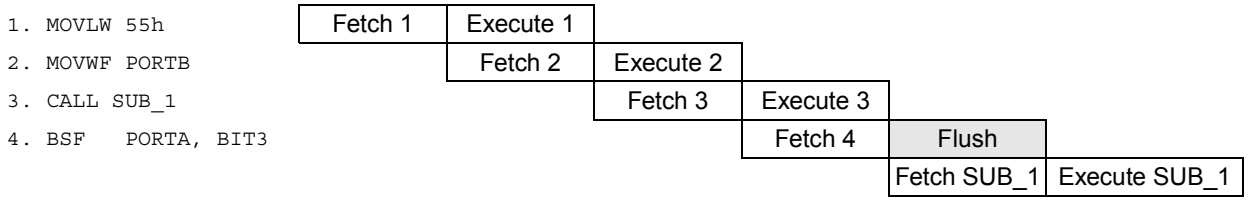
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

FIGURE 4-1: REGISTER FILE MAP - PIC16F83/CR83

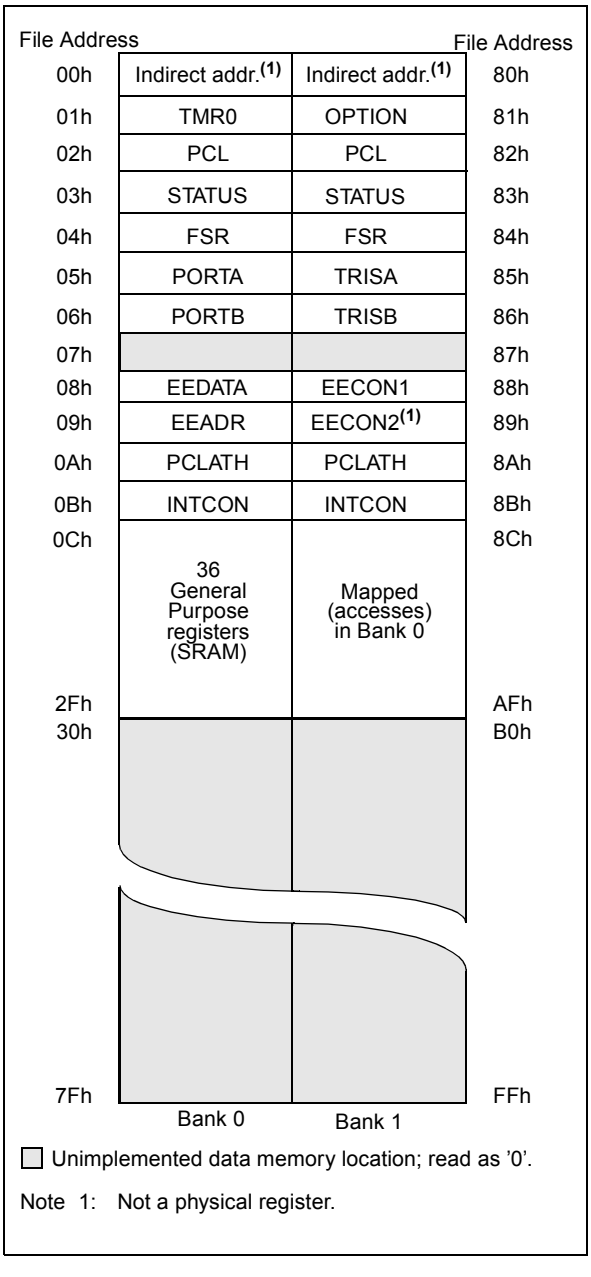


FIGURE 4-2: REGISTER FILE MAP - PIC16F84/CR84

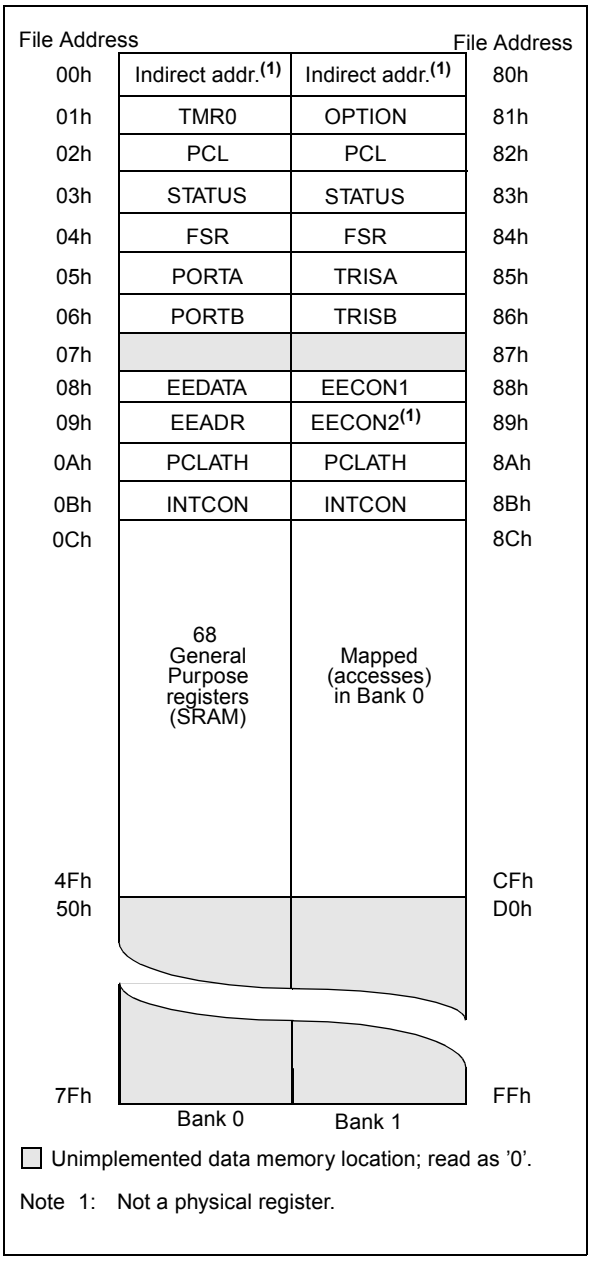


TABLE 4-1 REGISTER FILE SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note3)
Bank 0											
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of the Program Counter (PC)								0000 0000	0000 0000
03h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h		Unimplemented location, read as '0'								---- ----	---- ----
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (1)					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	0000 0000
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	PORTA data direction register					---1 1111	---1 1111
86h	TRISB	PORTB data direction register								1111 1111	1111 1111
87h		Unimplemented location, read as '0'								---- ----	---- ----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2 (not a physical register)								---- ----	---- ----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (1)					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> is never transferred to PCLATH.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in the STATUS register are not affected by a $\overline{MCLR}$ reset.

3: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer Reset.

5.0 I/O PORTS

The PIC16F8X has two ports, PORTA and PORTB. Some port pins are multiplexed with an alternate function for other features on the device.

5.1 PORTA and TRISA Registers

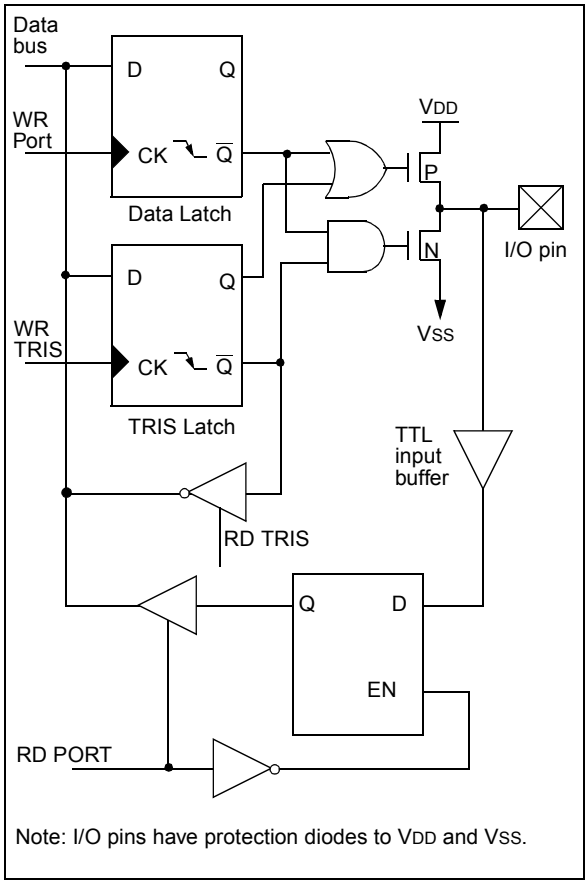
PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

Setting a TRISA bit (=1) will make the corresponding PORTA pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., put the contents of the output latch on the selected pin.

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The RA4 pin is multiplexed with the TMR0 clock input.

FIGURE 5-1: BLOCK DIAGRAM OF PINS RA3:RA0

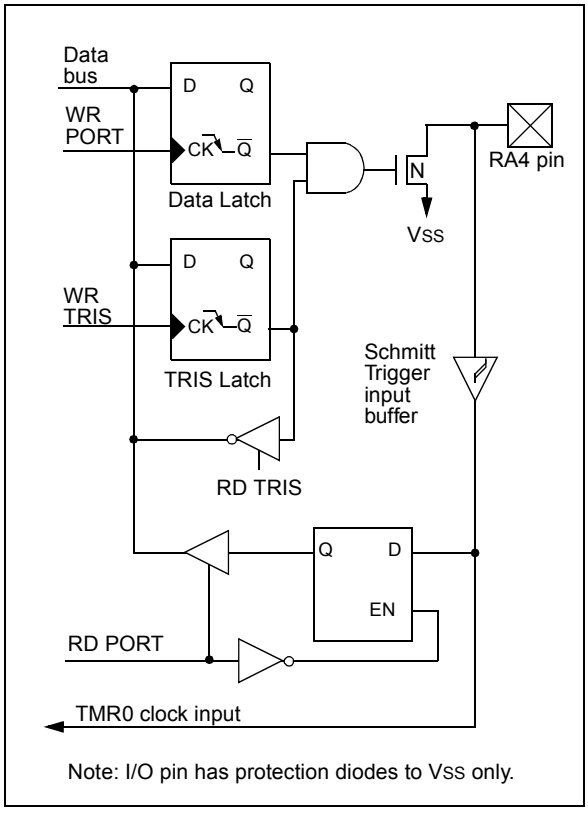


Note: I/O pins have protection diodes to VDD and VSS.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF    PORTA       ; Initialize PORTA by
                    ; setting output
                    ; data latches
BSF     STATUS, RP0 ; Select Bank 1
MOVLW   0x0F        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISA       ; Set RA<3:0> as inputs
                    ; RA4 as outputs
                    ; TRISA<7:5> are always
                    ; read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF PIN RA4



Note: I/O pin has protection diodes to VSS only.

TABLE 5-1 PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open drain type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 5-2 SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are unimplemented, read as '0'

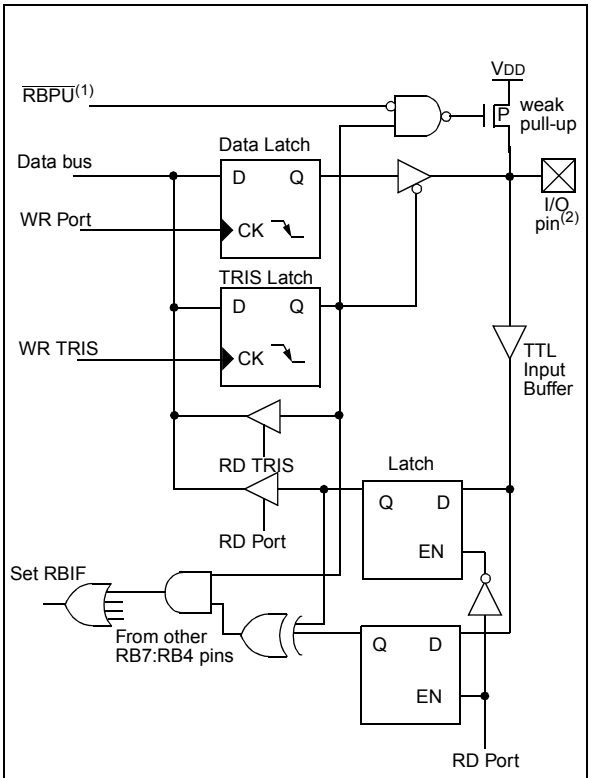
5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' on any bit in the TRISB register puts the corresponding output driver in a hi-impedance mode. A '0' on any bit in the TRISB register puts the contents of the output latch on the selected pin(s).

Each of the PORTB pins have a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION_REG<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The pins value in input mode are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of the pins are OR'ed together to generate the RB port change interrupt.

FIGURE 5-3: BLOCK DIAGRAM OF PINS RB7:RB4



Note 1: TRISB = '1' enables weak pull-up (if $\overline{\text{RBPU}}$ = '0' in the OPTION_REG register).

2: I/O pins have diode protection to VDD and VSS.

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Read (or write) PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

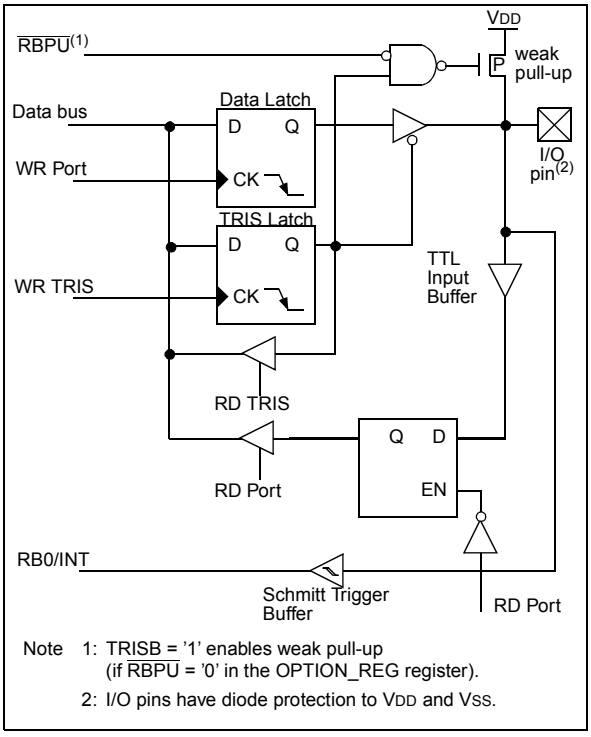
A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression (see AN552 in the Embedded Control Handbook).

Note 1: For a change on the I/O pin to be recognized, the pulse width must be at least TCY (4/fOSC) wide.

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-4: BLOCK DIAGRAM OF PINS RB3:RB0



Note 1: TRISB = '1' enables weak pull-up (if $\overline{\text{RBPU}}$ = '0' in the OPTION_REG register).

2: I/O pins have diode protection to VDD and VSS.

EXAMPLE 5-1: INITIALIZING PORTB

```
CLRF    PORTB        ; Initialize PORTB by
                     ; setting output
                     ; data latches
BSF     STATUS, RP0  ; Select Bank 1
MOVLW   0xCF         ; Value used to
                     ; initialize data
                     ; direction
MOVWF   TRISB        ; Set RB<3:0> as inputs
                     ; RB<5:4> as outputs
                     ; RB<7:6> as inputs
```

TABLE 5-3 PORTB FUNCTIONS

Name	Bit	Buffer Type	I/O Consistency Function
RB0/INT	bit0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST(2)	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST(2)	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger.

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4 SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

8.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

Address 2007h is beyond the user program memory space and it belongs to the special test/configuration memory space (2000h - 3FFFh). This space can only be accessed during programming.

To find out how to program the PIC16C84, refer to *PIC16C84 EEPROM Memory Programming Specification* (DS30189).

FIGURE 8-1: CONFIGURATION WORD - PIC16CR83 AND PIC16CR84

R-u	R-u	R-u	R-u	R-u	R-u	R/P-u	R-u	R-u	R-u	R-u	R-u	R-u	R-u
CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
bit13													bit0

R = Readable bit
P = Programmable bit
- n = Value at POR reset
u = unchanged

bit 13:8 **CP**: Program Memory Code Protection bit
1 = Code protection off
0 = Program memory is code protected

bit 7 **DP**: Data Memory Code Protection bit
1 = Code protection off
0 = Data memory is code protected

bit 6:4 **CP**: Program Memory Code Protection bit
1 = Code protection off
0 = Program memory is code protected

bit 3 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
1 = Power-up timer is disabled
0 = Power-up timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1:0 **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

FIGURE 8-2: CONFIGURATION WORD - PIC16F83 AND PIC16F84

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
bit13													bit0

R = Readable bit
P = Programmable bit
- n = Value at POR reset
u = unchanged

bit 13:4 **CP**: Code Protection bit
1 = Code protection off
0 = All memory is code protected

bit 3 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
1 = Power-up timer is disabled
0 = Power-up timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1:0 **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

8.2 Oscillator Configurations

8.2.1 OSCILLATOR TYPES

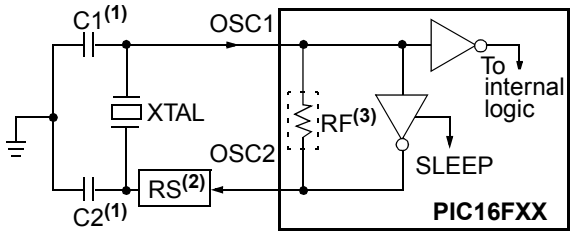
The PIC16F8X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

8.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 8-3).

FIGURE 8-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note1: See Table 8-1 for recommended values of C1 and C2.
2: A series resistor (RS) may be required for AT strip cut crystals.
3: RF varies with the crystal chosen.

The PIC16F8X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 8-4).

8.9 Interrupts

The PIC16F8X has 4 sources of interrupt:

- External interrupt RB0/INT pin
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB7:RB4)
- Data EEPROM write complete interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also contains the individual and global interrupt enable bits.

The global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. Bit GIE is cleared on reset.

The "return from interrupt" instruction, `RETFIE`, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

When an interrupt is responded to; the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. For external interrupt events, such as the RB0/INT pin or PORTB change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 8-17). The latency is the same for both one and two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

FIGURE 8-16: INTERRUPT LOGIC

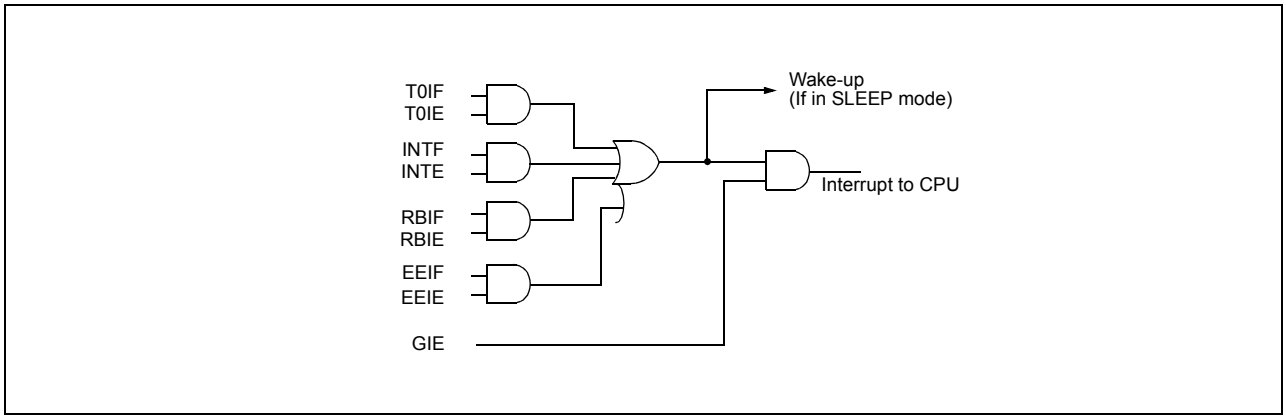
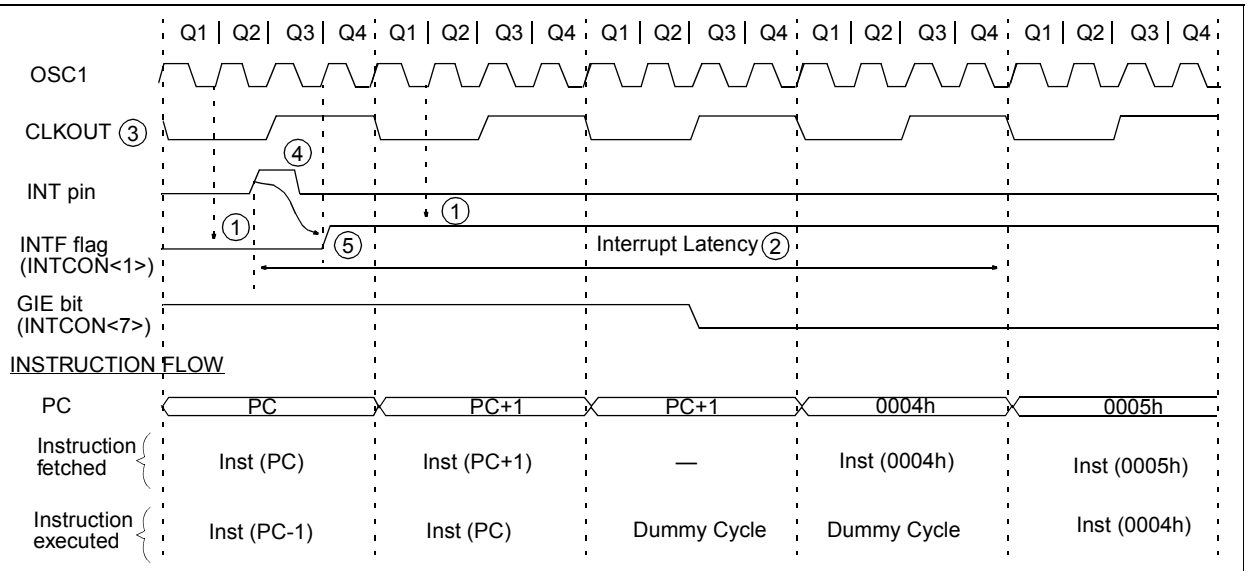


FIGURE 8-17: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4Tcy where Tcy = instruction cycle time.
Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in RC oscillator mode.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

8.9.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION_REG<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing control bit INTE (INTCON<4>). Flag bit INTF must be cleared in software via the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake the processor from SLEEP (Section 8.12) only if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether the processor branches to the interrupt vector following wake-up.

8.9.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in TMR0 will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 6.0).

8.9.3 PORT RB INTERRUPT

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<3>) (Section 5.2).

Note 1: For a change on the I/O pin to be recognized, the pulse width must be at least TCY wide.

9.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 9-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 9-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 9-1 OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 9-2 lists the instructions recognized by the MPASM assembler.

Figure 9-1 shows the general formats that the instructions can have.

Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 9-1: GENERAL FORMAT FOR INSTRUCTIONS

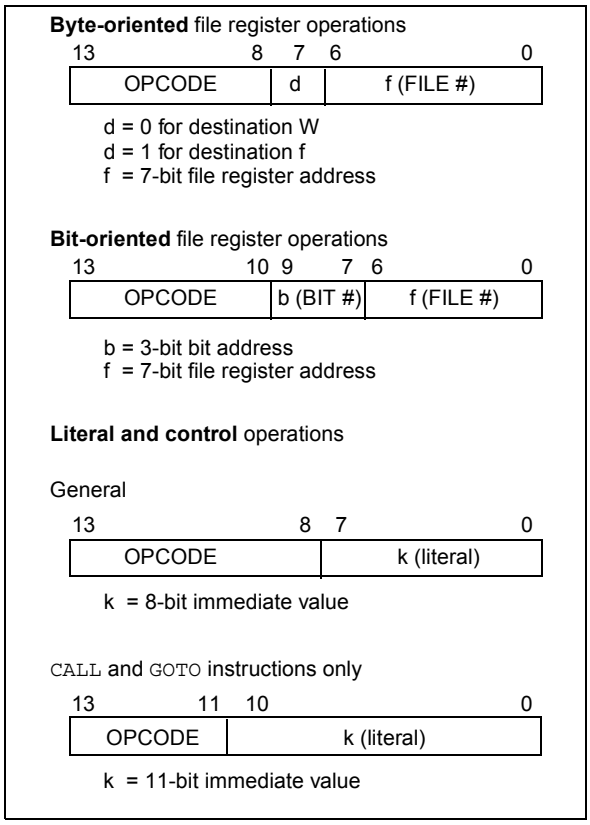


TABLE 9-2 PIC16FXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb	LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	**f, d**	Add W and f	1	00 0111 dfff ffff		C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	00 0101 dfff ffff		Z	1,2
CLRF	**f**	Clear f	1	00 0001 lfff ffff		Z	2
CLRW	**-**	Clear W	1	00 0001 0xxx xxxx		Z	
COMF	**f, d**	Complement f	1	00 1001 dfff ffff		Z	1,2
DECF	**f, d**	Decrement f	1	00 0011 dfff ffff		Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff			1,2,3
INCF	**f, d**	Increment f	1	00 1010 dfff ffff		Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00 1111 dfff ffff			1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	00 0100 dfff ffff		Z	1,2
MOVF	**f, d**	Move f	1	00 1000 dfff ffff		Z	1,2
MOVWF	**f**	Move W to f	1	00 0000 lfff ffff			
NOP	**-**	No Operation	1	00 0000 0xx0 0000			
RLF	**f, d**	Rotate Left f through Carry	1	00 1101 dfff ffff		C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	00 1100 dfff ffff		C	1,2
SUBWF	**f, d**	Subtract W from f	1	00 0010 dfff ffff		C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00 1110 dfff ffff			1,2
XORWF	**f, d**	Exclusive OR W with f	1	00 0110 dfff ffff		Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	**f, b**	Bit Clear f	1	01 00bb bfff ffff			1,2
BSF	**f, b**	Bit Set f	1	01 01bb bfff ffff			1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff			3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff			3
LITERAL AND CONTROL OPERATIONS							
ADDLW	**k**	Add literal and W	1	11 111x kkkk kkkk		C,DC,Z	
ANDLW	**k**	AND literal with W	1	11 1001 kkkk kkkk		Z	
CALL	**k**	Call subroutine	2	10 0kkk kkkk kkkk			
CLRWDT	**-**	Clear Watchdog Timer	1	00 0000 0110 0100		$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10 1kkk kkkk kkkk			
IORLW	**k**	Inclusive OR literal with W	1	11 1000 kkkk kkkk		Z	
MOVLW	**k**	Move literal to W	1	11 00xx kkkk kkkk			
RETFIE	**-**	Return from interrupt	2	00 0000 0000 1001			
RETLW	**k**	Return with literal in W	2	11 01xx kkkk kkkk			
RETURN	**-**	Return from Subroutine	2	00 0000 0000 1000			
SLEEP	**-**	Go into standby mode	1	00 0000 0110 0011		$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11 110x kkkk kkkk		C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	11 1010 kkkk kkkk		Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

PIC16F872

28-Pin, 8-Bit CMOS FLASH Microcontroller

Devices Included in this Data Sheet:

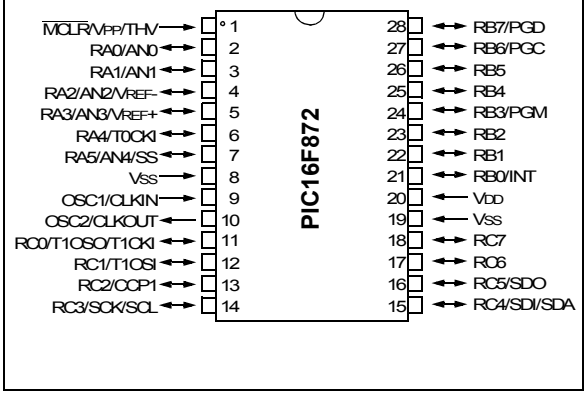
- PIC16F872

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory
  128 x 8 bytes of Data Memory (RAM)
  64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C72A
- Interrupt capability (up to 10 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram

DIP, SOIC, SSOP



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I^2C™ (Master/Slave)
- Brown-out detection circuitry for Brown-out Reset (BOR)

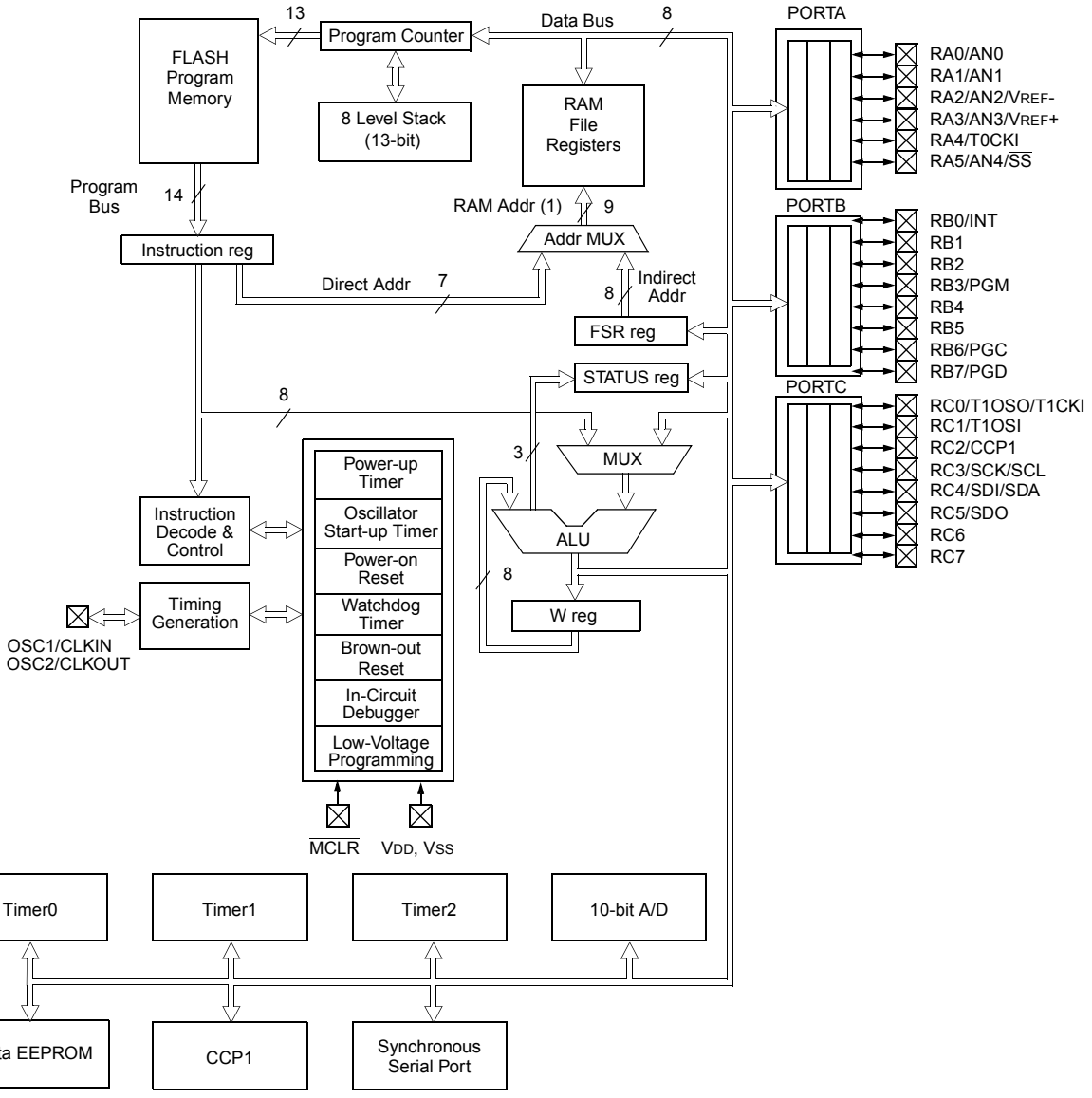
1.0 DEVICE OVERVIEW

This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

This data sheet covers the PIC16F872 device. The PIC16F872 is a 28-pin device and its block diagram is shown in Figure 1-1.

FIGURE 1-1: PIC16F872 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F872	2K	128 Bytes	64 Bytes



Note 1: Higher order bits are from the STATUS register.

TABLE 1-1: PIC16F872 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL/ST(1)	RB3 can also be the low voltage programming input.
RB4	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	I/O	TTL	Interrupt on change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6	17	17	I/O	ST	
RC7	18	18	I/O	ST	
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

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 the external interrupt or LVP.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of these PICmicro® MCUs. The Program Memory and Data Memory have separate buses, so that concurrent access can occur, and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

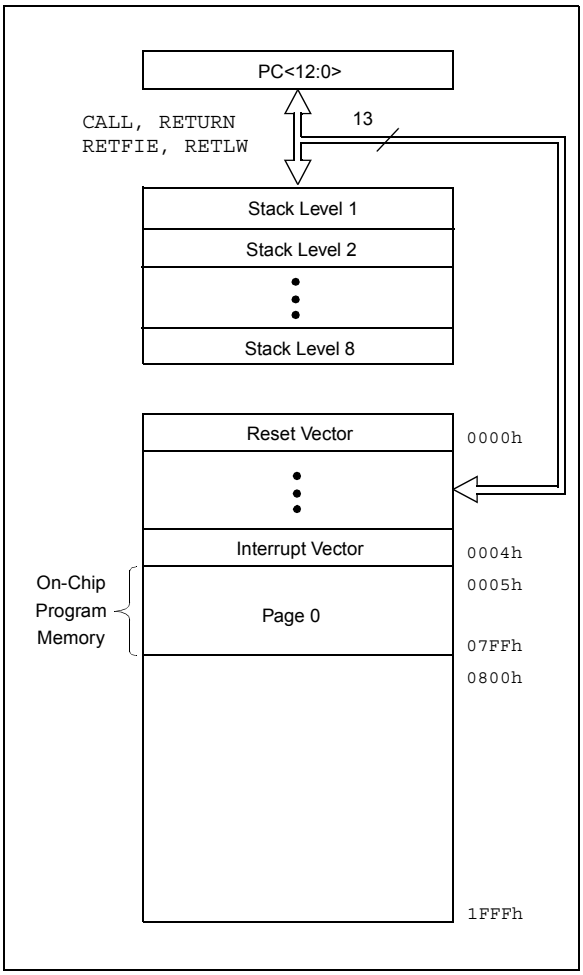
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16F872 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F872 device has 2K x 14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wrap-around.

The reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F872 PROGRAM MEMORY MAP AND STACK



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1(STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP<1:0>	Bank
00	0
01	1
10	2
11	3

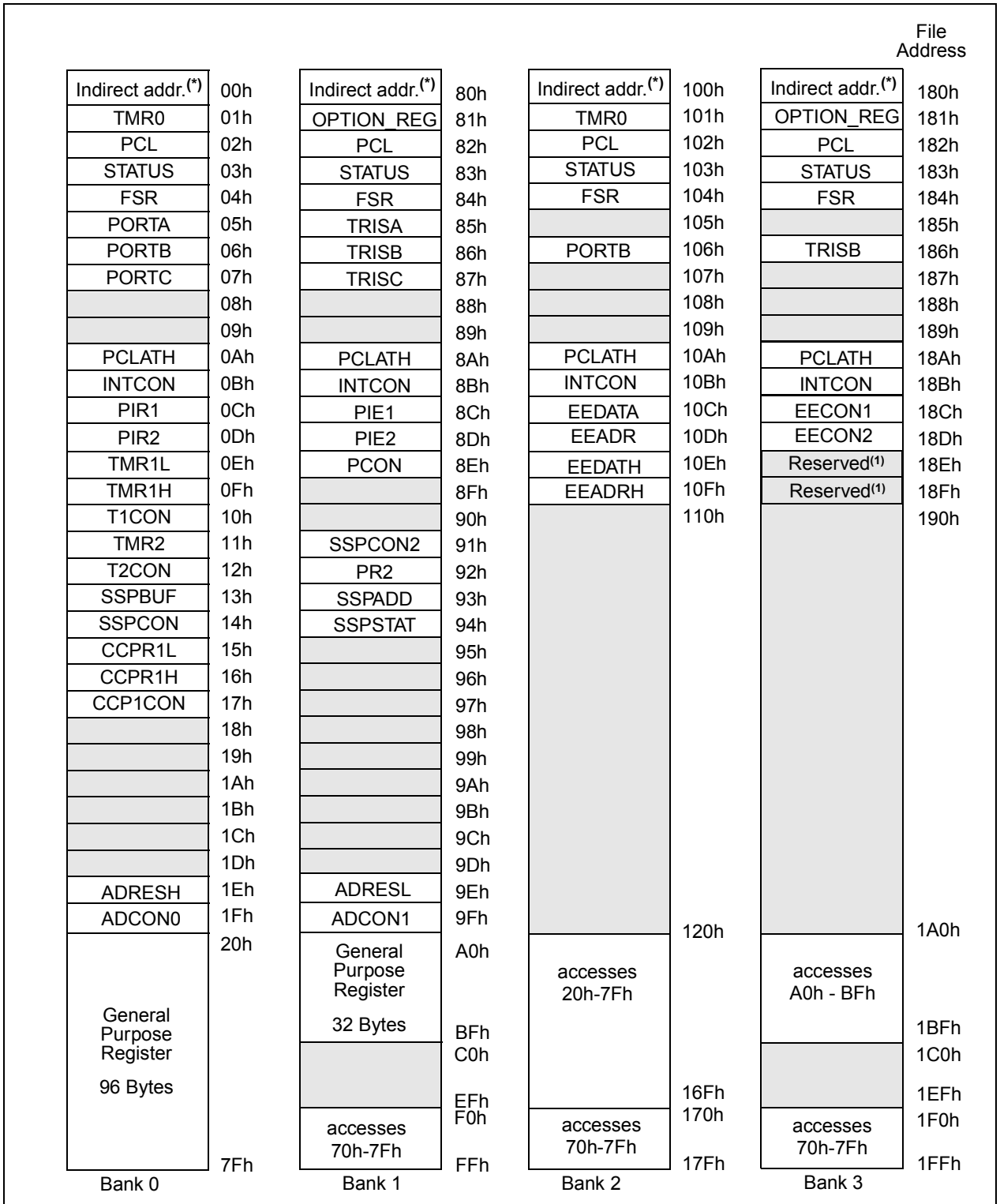
Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some “high use” Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Note:	EEPROM Data Memory description can be found in Section 4.0 of this Data Sheet

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR.

FIGURE 2-2: PIC16F872 REGISTER FILE MAP



Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are reserved; maintain these registers clear.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 0											
00h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h(3)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	(4)	ADIF	(4)	(4)	SSPIF	CCP1IF	TMR2IF	TMR1IF	r0rr 0000	r0rr 0000
0Dh	PIR2	—	(4)	—	EEIF	BCLIF	—	—	(4)	-r-0 0--r	-r-0 0--r
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: These registers can be addressed from any bank.

4: These bits are reserved; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 1											
80h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h(3)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	**(4)**	ADIE	**(4)**	**(4)**	SSPIE	CCP1IE	TMR2IE	TMR1IE	r0rr 0000	r0rr 0000
8Dh	PIE2	—	**(4)**	—	EEIE	BCLIE	—	—	**(4)**	-r-0 0--r	-r-0 0--r
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	BOR	---- --qq	---- --uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: These registers can be addressed from any bank.

4: These bits are reserved; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 2											
100h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h(3)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h(3)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high byte						xxxx xxxx	uuuu uuuu
10Fh	EEADRH	—	—	—	EEPROM address register high byte					xxxx xxxx	uuuu uuuu
Bank 3											
180h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h(3)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h(3)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

3: These registers can be addressed from any bank.

4: These bits are reserved; always maintain these bits clear.

2.2.2.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits, see the "Instruction Set Summary."

Note:	The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n= Value at POR reset

bit 7: **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h - 1FFh)
`0` = Bank 0, 1 (00h - FFh)

bit 6-5: **RP<1:0>**: Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h - 1FFh)
`10` = Bank 2 (100h - 17Fh)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes

bit 4: **$\overline{TO}$**: Time-out bit
`1` = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3: **$\overline{PD}$**: Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2: **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
(for $\overline{\text{borrow}}$ the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the most significant bit of the result occurred
`0` = No carry-out from the most significant bit of the result occurred
Note: For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

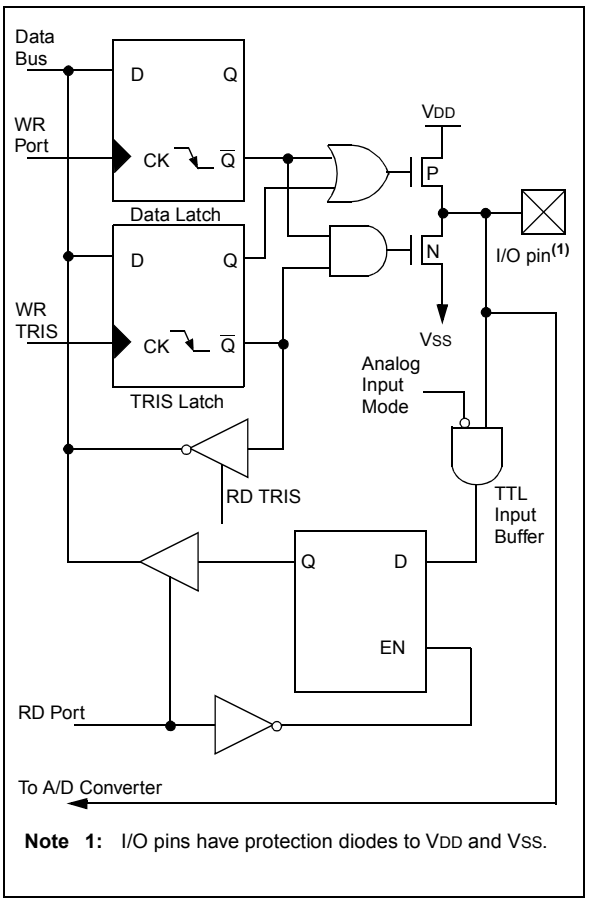
Note:	On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

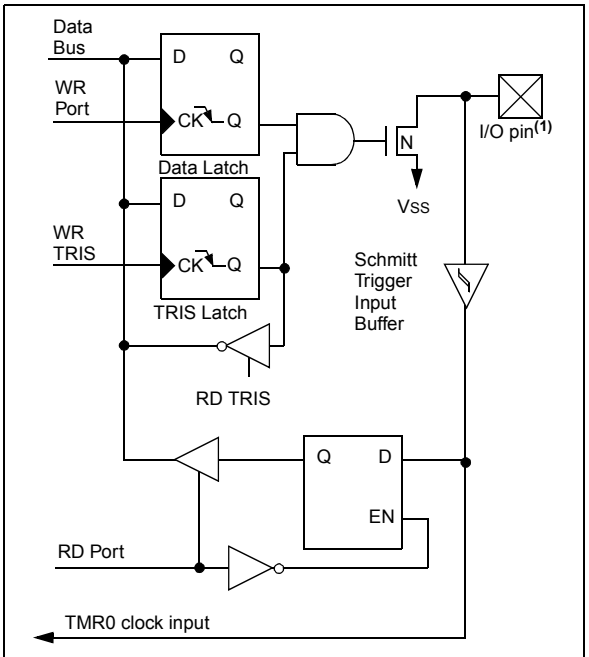
```
BCF     STATUS, RP0     ;
BCF     STATUS, RP1     ; Bank0
CLRF    PORTA           ; Initialize PORTA by
                        ; clearing output
                        ; data latches
BSF     STATUS, RP0     ; Select Bank 1
MOVLW   0x06            ; Configure all pins
MOVWF   ADCON1          ; as digital inputs
MOVLW   0xCF            ; Value used to
                        ; initialize data
                        ; direction
MOVWF   TRISA           ; Set RA<3:0> as inputs
                        ; RA<5:4> as outputs
                        ; TRISA<7:6> are always
                        ; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA<3:0> AND RA5 PINS



Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



Note 1: I/O pin has protection diodes to VSS only.

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input
RA1/AN1	bit1	TTL	Input/output or analog input
RA2/AN2	bit2	TTL	Input/output or analog input
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0 Output is open drain type
RA5/$\overline{SS}$/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input

Legend: TTL = TTL input, ST = Schmitt Trigger input.

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

Note: When using the SSP module in SPI slave mode and $\overline{SS}$ enabled, the A/D converter must be set to one of the following modes where PCFG<3:0> = 0100,0101, 011x, 1101, 1110, 1111.

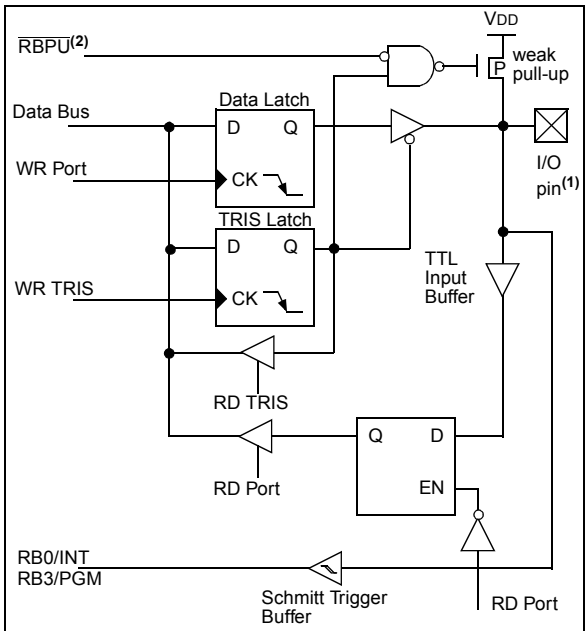
3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (=1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISB bit (=0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Three pins of PORTB are multiplexed with the Low Voltage Programming function; RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 3-3: BLOCK DIAGRAM OF RB<3:0> PINS



Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB<7:4>) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

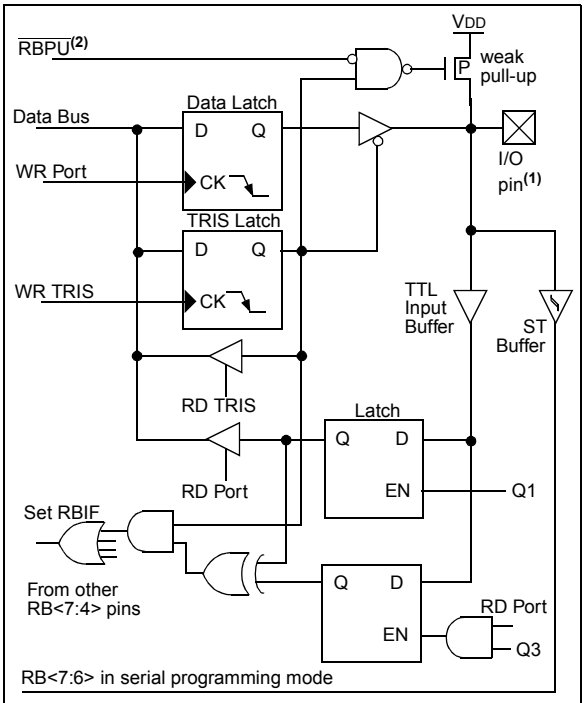
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key-depression. Refer to the Embedded Control Handbook, *"Implementing Wake-Up on Key Stroke"* (AN552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

RB0/INT is discussed in detail in Section 11.10.1.

FIGURE 3-4: BLOCK DIAGRAM OF RB<7:4> PINS



Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION_REG<7>).

Note: When using Low Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.

TABLE 3-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM	bit3	TTL/ST(1)	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST(2)	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST(2)	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input.

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt or LVP mode.

2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

3.3 PORTC and the TRISC Register

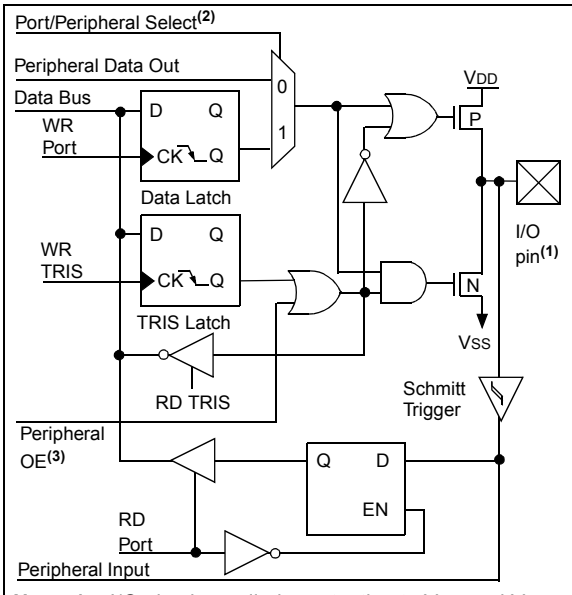
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

When the I^2C module is enabled, the PORTC (3:4) pins can be configured with normal I^2C levels or with SMBUS levels by using the CKE bit (SSPSTAT<6>).

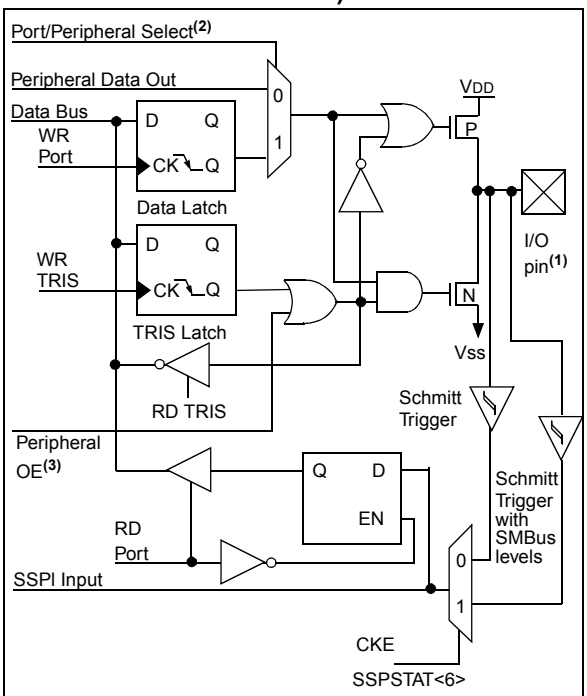
When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (`BSF, BCF, XORWF`) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<0:2> RC<5:7>



Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<3:4>



Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

TABLE 3-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI	bit1	ST	Input/output port pin or Timer1 oscillator input.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6	bit6	ST	Input/output port pin.
RC7	bit7	ST	Input/output port pin.

Legend: ST = Schmitt Trigger input.

TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

REGISTER 11-1: CONFIGURATION WORD

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	F0SC1	F0SC0
bit13													bit0

Register: CONFIG
Address 2007h

bit 13-12:

bit 5-4: **CP<1:0>:** Flash Program Memory Code Protection bits (2)
11 = Code protection off
10 = 0000h to 06FFh code protected
01 = 0000h to 03FFh code protected
00 = 0000h to 07FFh code protected

bit 11: **DEBUG:** In-Circuit Debugger Mode
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins.
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.

bit 10: **Unimplemented: Read as '1'**

bit 9: **WRT:** Flash Program Memory Write Enable
1 = Unprotected program memory may be written to by EECON control
0 = Unprotected program memory may not be written to by EECON control

bit 8: **CPD:** Data EE Memory Code Protection
1 = Code protection off
0 = Data EEPROM memory code protected

bit 7: **LVP**: Low Voltage In-Circuit Serial Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3 is digital I/O, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 6: **BODEN**: Brown-out Reset Enable bit (1)
1 = BOR enabled
0 = BOR disabled

bit 3: **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit (1)
1 = PWRT disabled
0 = PWRT enabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.
2: All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.

TABLE 11-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

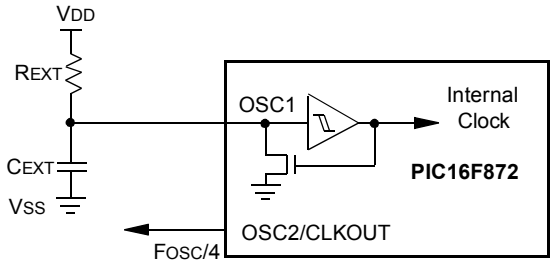
3: Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

4: When migrating from other PICmicro devices, oscillator performance should be verified.

11.2.3 RC OSCILLATOR

For timing insensitive applications, the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C_{EXT} values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 11-3 shows how the R/C combination is connected to the PIC16F872.

FIGURE 11-3: RC OSCILLATOR MODE



Recommended values: $3\ k\Omega \le R_{EXT} \le 100\ k\Omega$
$C_{EXT} > 20pF$

11.10 Interrupts

The PIC16F872 has 10 sources of interrupt. The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

Note:	Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set regardless of the status of the GIE bit. The GIE bit is cleared on reset.

The "return from interrupt" instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables interrupts.

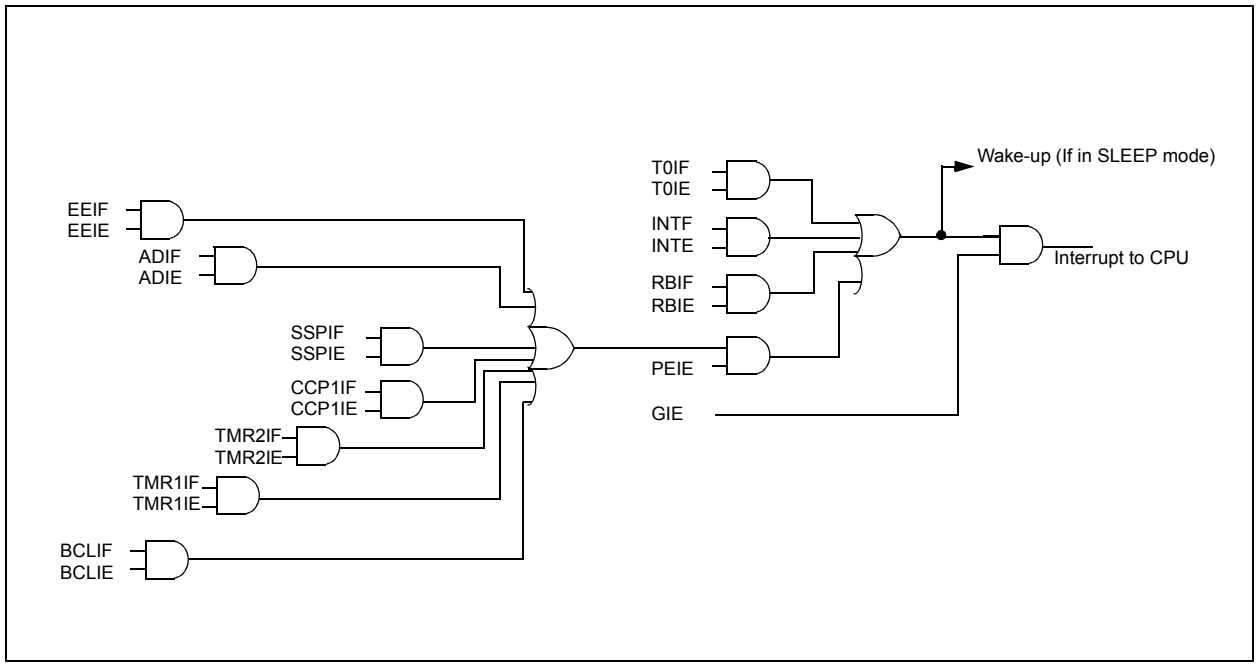
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in Special Function Registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in Special Function Register INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs. The latency is the same for one or two cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit

FIGURE 11-9: INTERRUPT LOGIC



12.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 12-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 12-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 12-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 12-2 lists the instructions recognized by the MPASM assembler.

Figure 12-1 shows the general formats that the instructions can have.

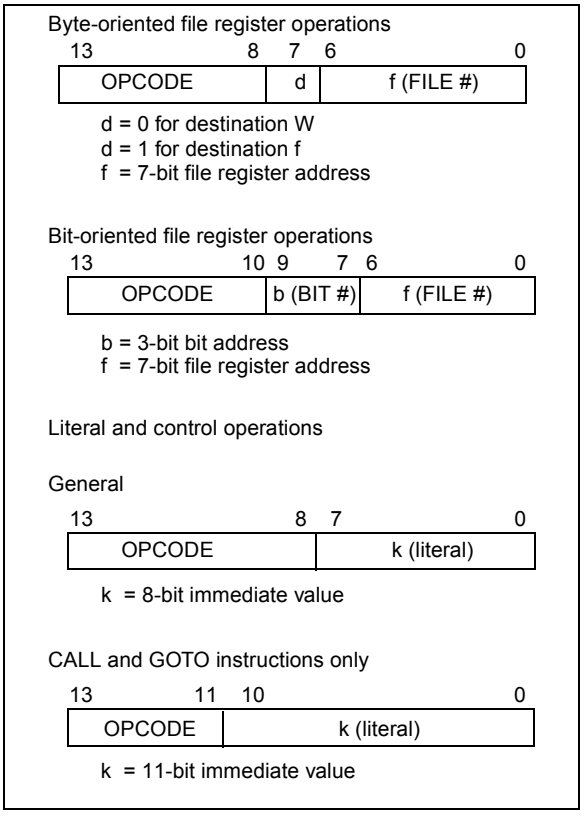
Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

TABLE 12-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

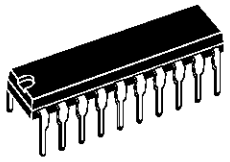
Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

SGS-THOMSON MICROELECTRONICS

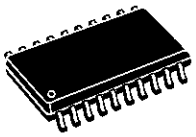
L297
L297D

STEPPER MOTOR CONTROLLERS

- NORMAL/WAWE DRIVE
- HALF/FULL STEP MODES
- CLOCKWISE/ANTICLOCKWISE DIRECTION
- SWITCHMODE LOAD CURRENT REGULATION
- PROGRAMMABLE LOAD CURRENT
- FEW EXTERNAL COMPONENTS
- RESET INPUT & HOME OUTPUT
- ENABLE INPUT



DIP20



SO20

ORDERING NUMBERS : L297 (DIP20)
L297D (SO20)

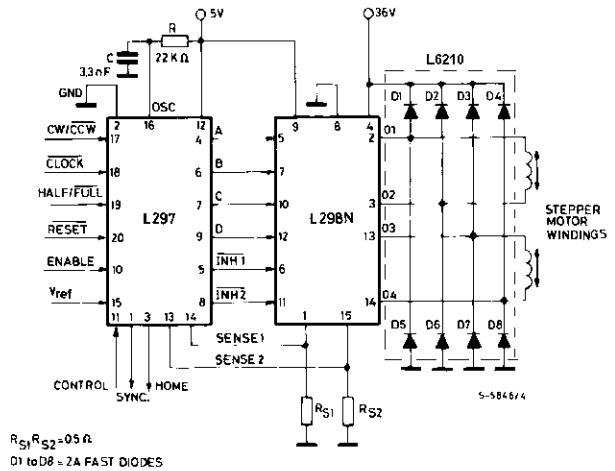
DESCRIPTION

The L297/A/D Stepper Motor Controller IC generates four phase drive signals for two phase bipolar and four phase unipolar step motors in microcomputer-controlled applications. The motor can be driven in half step, normal and wawe drive modes and on-chip PWM chopper circuits permit switchmode control of the current in the windings. A feature of this device is that it requires only clock, direction and mode input signals. Since the phase are generated internally the burden on the microprocessor, and the programmer, is greatly reduced. Mounted in DIP20 and SO20 packages, the L297 can be used with monolithic bridge drives such as the L298N or L293E, or with discrete transistors and darlingtons.

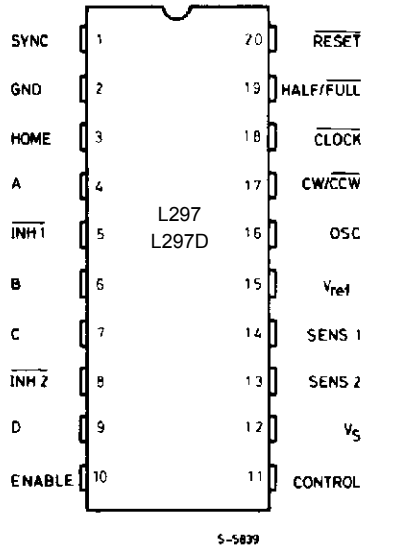
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Supply voltage	10	V
V_i	Input signals	7	V
P_{tot}	Total power dissipation (T_{amb} = 70°C)	1	W
T_{stg}, T_j	Storage and junction temperature	-40 to + 150	°C

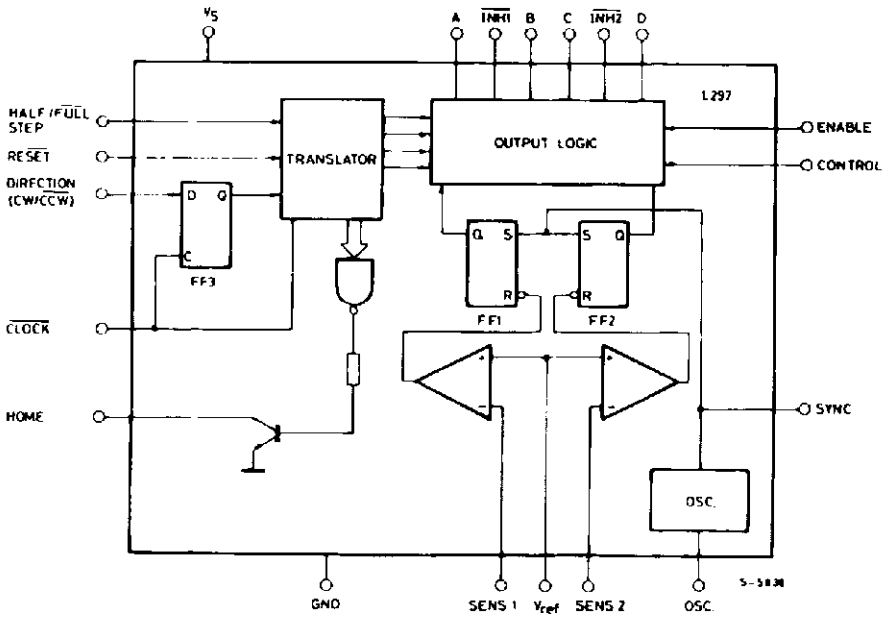
TWO PHASE BIPOLAR STEPPER MOTOR CONTROL CIRCUIT



PIN CONNECTION (Top view)



BLOCK DIAGRAM (L297/L297D)



PIN FUNCTIONS - L297/L297D

N°	NAME	FUNCTION
1	SYNC	Output of the on-chip chopper oscillator. The SYNC connections The SYNC connections of all L297s to be synchronized are connected together and the oscillator components are omitted on all but one. If an external clock source is used it is injected at this terminal.
2	GND	Ground connection.
3	HOME	Open collector output that indicates when the L297 is in its initial state (ABCD = 0101). The transistor is open when this signal is active.
4	A	Motor phase A drive signal for power stage.
5	$\overline{INH1}$	Active low inhibit control for driver stage of A and B phases. When a bipolar bridge is used this signal can be used to ensure fast decay of load current when a winding is de-energized. Also used by chopper to regulate load current if CONTROL input is low.
6	B	Motor phase B drive signal for power stage.
7	C	Motor phase C drive signal for power stage.
8	$\overline{INH2}$	Active low inhibit control for drive stages of C and D phases. Same functions as INH1.
9	D	Motor phase D drive signal for power stage.
10	ENABLE	Chip enable input. When low (inactive) INH1, INH2, A, B, C and D are brought low.
11	CONTROL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
12	V_s	5V supply input.
13	$SENS_2$	Input for load current sense voltage from power stages of phases C and D.
14	$SENS_1$	Input for load current sense voltage from power stages of phases A and B.
15	V_{ref}	Reference voltage for chopper circuit. A voltage applied to this pin determines the peak load current.
16	OSC	An RC network (R to V_{CC}, C to ground) connected to this terminal determines the chopper rate. This terminal is connected to ground on all but one device in synchronized multi - L297 configurations. f $\cong$ 1/0.69 RC
17	CW/$\overline{CCW}$	Clockwise/counterclockwise direction control input. Physical direction of motor rotation also depends on connection of windings. Synchronized internally therefore direction can be changed at any time.
18	$\overline{CLOCK}$	Step clock. An active low pulse on this input advances the motor one increment. The step occurs on the rising edge of this signal.

PIN FUNCTIONS - L297/L297D (continued)

N°	NAME	FUNCTION
19	HALF/$\overline{\text{FULL}}$	Half/full step select input. When high selects half step operation, when low selects full step operation. One-phase-on full step mode is obtained by selecting FULL when the L297's translator is at an even-numbered state. Two-phase-on full step mode is set by selecting FULL when the translator is at an odd numbered position. (The home position is designate state 1).
20	$\overline{\text{RESET}}$	Reset input. An active low pulse on this input restores the translator to the home position (state 1, ABCD = 0101).

THERMAL DATA

Symbol	Parameter		DIP20	SO20	Unit
$R_{th\text{-}j\text{-}amb}$	Thermal resistance junction-ambient	max	80	100	°C/W

CIRCUIT OPERATION

The L297 is intended for use with a dual bridge driver, quad darlington array or discrete power devices in step motor driving applications. It receives step clock, direction and mode signals from the systems controller (usually a microcomputer chip) and generates control signals for the power stage.

The principal functions are a translator, which generates the motor phase sequences, and a dual PWM chopper circuit which regulates the current in the motor windings. The translator generates three different sequences, selected by the HALF/FULL input. These are normal (two phases energised), wave drive (one phase energised) and half-step (alternately one phase energised/two phases energised). Two inhibit signals are also generated by the L297 in half step and wave drive modes. These signals, which connect directly to the L298's enable inputs, are intended to speed current decay when a winding is de-energised. When the L297 is used to drive a unipolar motor the chopper acts on these lines.

An input called CONTROL determines whether the chopper will act on the phase lines ABCD or the inhibit lines INH1 and INH2. When the phase lines are chopped the non-active phase line of each pair (AB or CD) is activated (rather than interrupting the line then active). In L297 + L298 configurations this technique reduces dissipation in the load current sense resistors.

A common on-chip oscillator drives the dual chopper. It supplies pulses at the chopper rate which set the two flip-flops FF1 and FF2. When the current in a winding reaches the programmed peak value the voltage across the sense resistor (connected to one of the sense inputs $SENS_1$ or $SENS_2$) equals V_{ref} and the corresponding comparator resets its flip flop, interrupting the drive current until the next oscillator pulse arrives. The peak current for both windings is programmed by a voltage divider on the V_{ref} input.

Ground noise problems in multiple configurations can be avoided by synchronising the chopper oscillators. This is done by connecting all the SYNC pins together, mounting the oscillator RC network on one device only and grounding the OSC pin on all other devices.

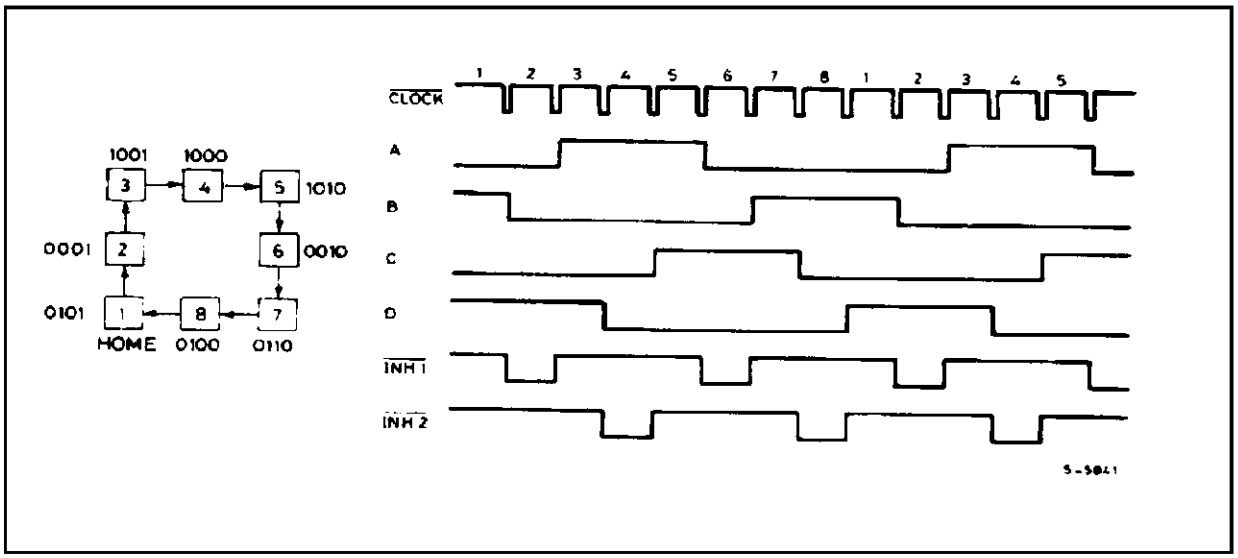
MOTOR DRIVING PHASE SEQUENCES

The L297's translator generates phase sequences for normal drive, wave drive and half step modes. The state sequences and output waveforms for these three modes are shown below. In all cases the translator advances on the low to high transistion of $\overline{CLOCK}$.

Clockwise rotation is indicate; for anticlockwise rotation the sequences are simply reversed $\overline{RESET}$ restores the translator to state 1, where ABCD = 0101.

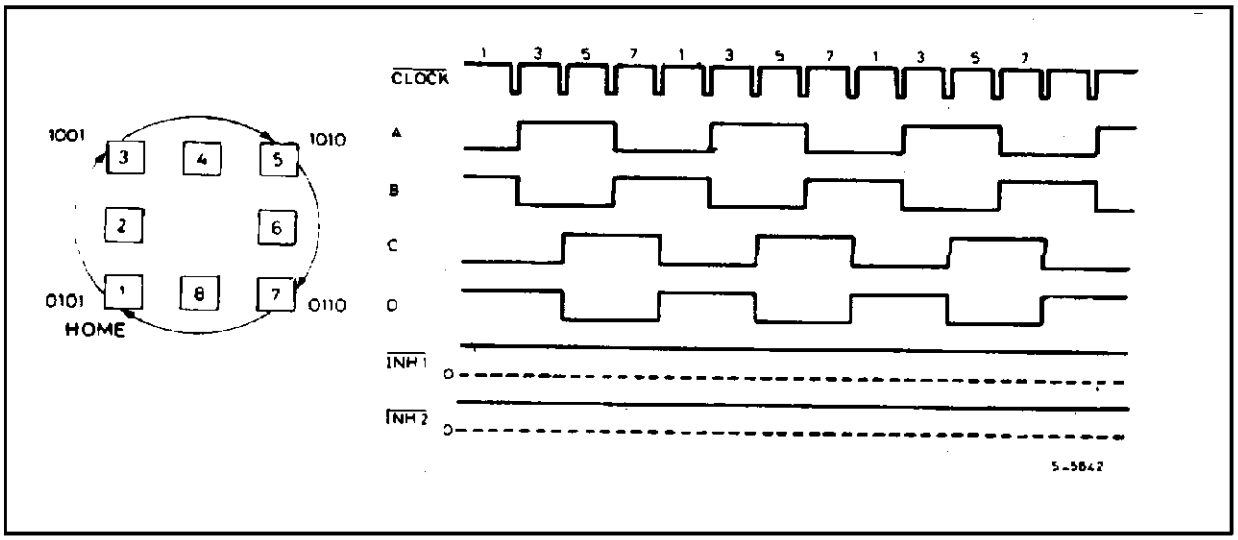
HALF STEP MODE

Half step mode is selected by a high level on the HALF/$\overline{FULL}$ input.



NORMAL DRIVE MODE

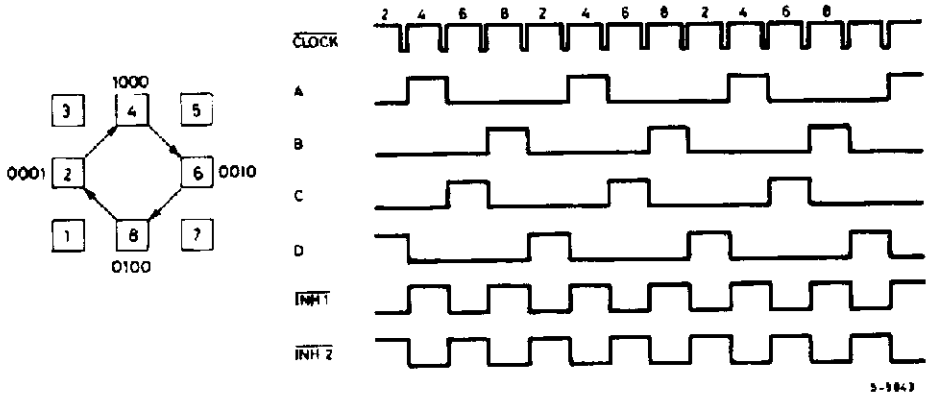
Normal drive mode (also called "two-phase-on" drive) is selected by a low level on the HALF/$\overline{FULL}$ input when the translator is at an odd numbered state (1, 3, 5 or 7). In this mode the $\overline{INH1}$ and $\overline{INH2}$ outputs remain high throughout.



MOTOR DRIVING PHASE SEQUENCES (continued)

WAVE DRIVE MODE

Wave drive mode (also called "one-phase-on" drive) is selected by a low level on the HALF/$\overline{FULL}$ input when the translator is at an even numbered state (2, 4, 6 or 8).



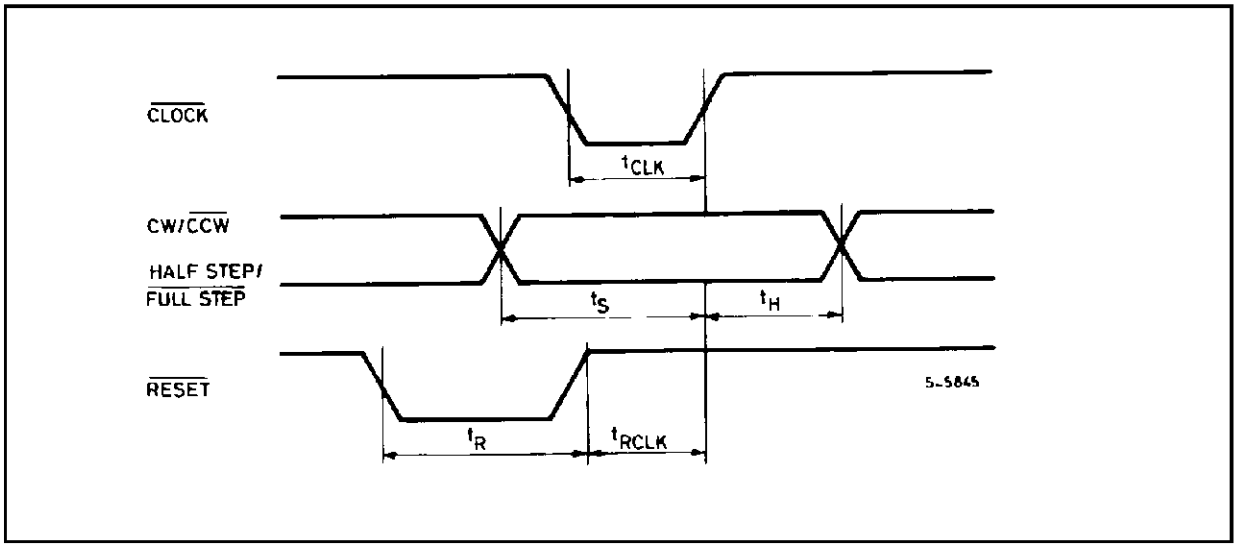
ELECTRICAL CHARACTERISTICS (Refer to the block diagram T_{amb} = 25°C, V_s = 5V unless otherwise specified)

Symbol	Parameter	Test conditions		Min.	Typ	Max.	Unit
V_s	Supply voltage (pin 12)			4.75		7	V
I_s	Quiescent supply current (pin 12)	Outputs floating			50	80	mA
V_i	Input voltage (pin 11, 17, 18, 19, 20)		Low			0.6	V
			High	2		V_s	V
I_i	Input current (pin 11, 17, 18, 19, 20)		V_i = L		100	µA	
			V_i = H			10	µA
V_{en}	Enable input voltage (pin 10)		Low			1.3	V
			High	2		V_s	V
I_{en}	Enable input current (pin 10)		V_{en} = L			100	µA
			V_{en} = H			10	µA
V_o	Phase output voltage (pins 4, 6, 7, 9)	I_o = 10mA	V_{OL}			0.4	V
		I_o = 5mA	V_{OH}	3.9			V
V_{inh}	Inhibit output voltage (pins 5, 8)	I_o = 10mA	$V_{inh\ L}$			0.4	V
		I_o = 5mA	$V_{inh\ H}$	3.9			V
V_{SYNC}	Sync Output Voltage	I_o = 5mA	$V_{SYNC\ H}$	3.3			V
		I_o = 5mA	$V_{SYNC\ V}$			0.8	

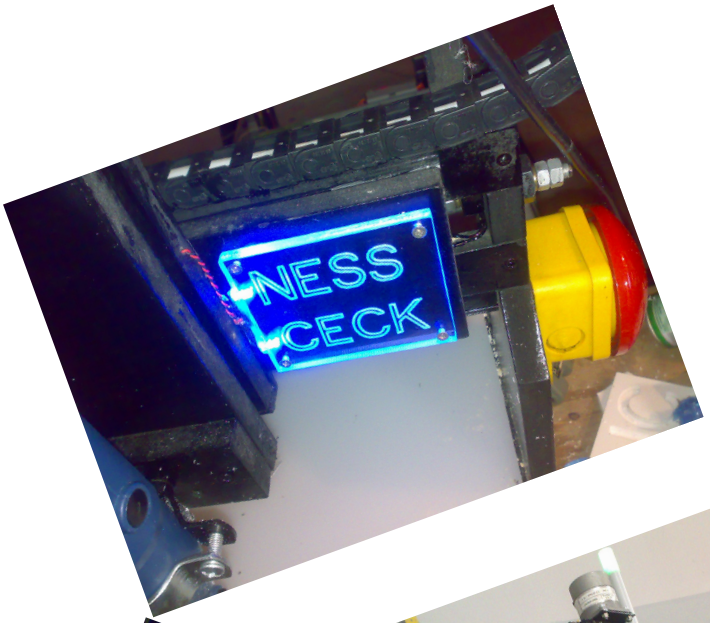
ELECTRICAL CHARACTERISTICS (continued)

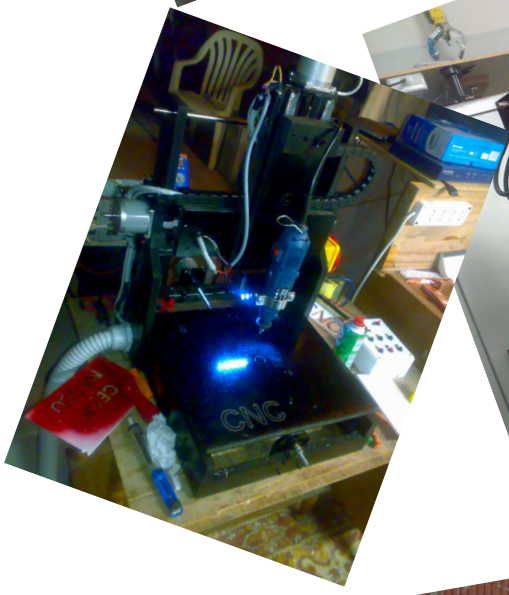
Symbol	Parameter	Test conditions	Min.	Typ	Max.	Unit
I_{leak}	Leakage current (pin 3)	V_{CE} = 7 V			1	µA
V_{sat}	Saturation voltage (pin 3)	I = 5 mA			0.4	V
V_{off}	Comparators offset voltage (pins 13, 14, 15)	V_{ref} = 1 V			5	mV
I_o	Comparator bias current (pins 13, 14, 15)		-100		10	µA
V_{ref}	Input reference voltage (pin 15)		0		3	V
t_{CLK}	Clock time		0.5			µs
t_S	Set up time		1			µs
t_H	Hold time		4			µs
t_R	Reset time		1			µs
t_{RCLK}	Reset to clock delay		1			µs

Figure 1.



⇨FOTO REALIZZAZIONE





Gli Autori

ITIS LEVI
CHIMICA ELETTRONICA E TELECOMUNICAZIONI INFORMATICA

Ceccato Roberto

Nassuato Denis