

Introduzione:

Paragrafo 1:

MACCHINE UTENSILIA CNC E G-CODE
[italiano-inglese]

Paragrafo 2:

PROGRAMMA PER ARDUINO

Paragrafo 3:

DEFINIZIONE MOTORE PASSO PASSO

Paragrafo 4:

ARDUINO

paragrafo 5:

DRIVER

Paragrafo 6:

CIRCUITO ELETTRICO

Paragrafo 7:

STRUTTURA

MACCHINE UTENSILI A CNC E G-CODE

[italiano]

Il progresso delle tecnologie elettroniche ha rivoluzionato l'industria metalmeccanica. Grazie alla macchina utensile a controllo numerico è stato possibile ridurre notevolmente i tempi di preparazione della macchina per la lavorazione di pezzi diversi con la semplice sostituzione di un preciso programma memorizzato su supporti magnetici.

La continua evoluzione della tecnologia elettronica non ha risparmiato la macchina utensile, passata dalle prime semplici lavorazioni da punto a punto all'esecuzione di qualsiasi profilo, anche il più complesso, grazie alla delega di tutte le operazioni di calcolo e verifica a un computer: si è così ottenuto il CNC (Controllo Numerico Computerizzato).

Ulteriori passi avanti sono stati fatti correlando all'unità di governo un video che permette di verificare le passate di lavorazione dell'utensile e dei suoi movimenti nel campo di lavoro, il profilo finale, le eventuali collisioni utensile-pezzo, il controllo delle tolleranze, la visualizzazione, all'occorrenza, di eventuali malfunzionamenti di qualche unità.

E' opportuno illustrare le differenze strutturali tra una macchina utensile tradizionale e una macchina utensile a controllo numerico.

In una macchina utensile tradizionale, quale il tornio, possiamo distinguere 3 sistemi principali:

- sistema di sostentamento, composto da basamento, bancale e relative guide;
- sistema di azionamento a mandrino: motore trifase, cinghie e gruppo testa motrice che racchiude tutti i meccanismi di sostentamento dell'albero del mandrino autocentrante;
- sistema di posizionamento dell'utensile: carro portautensili e relativi volantini di accostamento (con nonii graduati), vite madre e barra scanalata per il movimento uniforme per l'esecuzione della filettatura, per la sgrossatura o per la finitura e varie leve per il cambio di velocità del mandrino e degli avanzamenti dell'utensile.

Con la macchina utensile a controllo numerico computerizzato i primi due sistemi sono rimasti invariati; hanno invece subito una trasformazione elettronica e funzionale tutti i componenti elencati nel terzo.

La macchina utensile a CNC ha subito, cioè, una trasformazione elettromeccanica di tutto il sistema di posizionamento e di controllo dell'utensile e di controllo della rotazione del mandrino.

Il controllo numerico è una tecnica che consente di produrre pezzi finiti e/o semi finiti utilizzando macchine utensili asservite a un elaboratore di segnali: tale controllo è comunemente detto CNC.

Le funzioni principali di tale elaboratore di segnali sono:

- comando;
- controllo;
- regolazione

Il comando è rappresentato da una sequenza logica di codici alfanumerici che permettono agli organi di lavoro della macchina utensile di eseguire alcuni movimenti necessari alla lavorazione del pezzo.

L'insieme ordinato dei comandi determina un programma di lavoro.

Il CNC non ha rivoluzionato il modo di operare delle macchine utensili tradizionali e dei rispettivi utensili, ma ne ha notevolmente migliorato le prestazioni a vantaggio della precisione, della rapidità di posizionamento degli utensili specifici, dell'aumento della produttività e della ripetitività delle operazioni.

I codici a cui fanno riferimento i programmatori sono definiti dalla normativa ISO o più comunemente detti g-code e sono riportati nelle tabelle seguenti:

Significato

A Coordinata angolare attorno all'asse X

B Coordinata angolare attorno all'asse Y

C Coordinata angolare attorno all'asse Z

F Velocità di avanzamento

G Funzione preparatoria

M Funzione ausiliare

N Numero di blocco

S Velocità di rotazione del mandrino

T Utensile

% Inizio programma

X Dimensione del diametro

Z Dimensione della lunghezza

Funzioni ausiliari

M00 Arresto programmato

M01 Arresto opzionale

M02 Fine programma

M03 Rotazione mandrino in senso orario

M04 Rotazione mandrino in senso antiorario

M05 Arresto mandrino

M08 Avvio refrigerante

M09 Arresto refrigerante

M19 Orientamento mandrino

M30 Fine programma

M98 Richiamo sottoprogramma

M99 Fine sottoprogramma o ritorno a inizio programma

Funzioni preparatorie

- G00** Posizionamento in rapido assi
- G01** Interpolazione lineare (a velocità di avanzamento)
- G02** Interpolazione circolare in senso orario
- G03** Interpolazione circolare in senso antiorario
- G04** Sosta temporizzata
- G10** Impostazione dati
- G17** Selezione piano di lavoro XY
- G18** Selezione piano di lavoro ZX
- G19** Selezione piano di lavoro YZ
- G20** Programmazione in pollici
- G21** Programmazione in millimetri
- G22** Attivazione zone di controllo
- G23** Disattivazione zone di controllo
- G25** Disattivazione controllo di velocità mandrino
- G26** Attivazione controllo di velocità mandrino
- G27** Controllo del ritorno al punto di riferimento
- G28** Ritorno al punto di riferimento
- G30** Ritorno al secondo punto di riferimento
- G31** Salto della lavorazione
- G33** Movimento di filettatura
- G34** Movimento di filettatura a passo variabile
- G36** Compensazione automatica lunghezza utensile in X
- G37** Compensazione automatica lunghezza utensile in Z
- G40** Cancellazione correzione raggio utensile
- G41** Compensazione raggio utensile a sinistra (pezzo a destra)
- G42** Compensazione raggio utensile a destra (pezzo a sinistra)
- G92** Limitazione velocità massima mandrino
- G52** Impostazione sistema di coordinate locali
- G53** Selezione coordinate macchina
- G54** Cancellazione correzione raggio utensile
- G55** Compensazione raggio utensile a sinistra (pezzo a destra)
- G56** Compensazione raggio utensile a destra (pezzo a sinistra)
- G57** Limitazione velocità massima mandrino
- G58** Impostazione sistema di coordinate locali

- G59** Selezione coordinate macchina
- G65** Richiamo macro
- G66** Richiamo macro modale
- G67** Cancellazione richiamo macro modale
- G68** Attivazione specularità asse Z (seconda torretta)
- G69** Disattivazione specularità asse Z (seconda torretta)
- G70** Ciclo di finitura
- G71** Ciclo di sgrossatura in tornitura (direzione Z)
- G72** Ciclo di sgrossatura in sfacciatura (direzione X)
- G73** Ciclo di ripetizione del profilo
- G74** Foratura a tratti lungo asse Z
- G75** Foratura a tratti con utensili motorizzati lungo asse X o gole
- G76** Ciclo di filettatura in più passate
- G80** Cancellazione cicli fissi
- G83** Ciclo fisso di foratura frontale
- G84** Ciclo fisso di maschiatura frontale
- G86** Ciclo fisso di alesatura frontale
- G87** Ciclo fisso di foratura laterale
- G88** Ciclo fisso di maschiatura laterale
- G89** Ciclo fisso di alesatura laterale
- G77** Ciclo di lavorazione diametrale esterno/interno
- G78** Ciclo di filettatura
- G79** Ciclo di tornitura frontale
- G96** Velocità di taglio costante
- G97** Velocità di rotazione costante
- G94** Avanzamento in unità al giro
- G95** Avanzamento in unità al minuto
- G90** Programmazione in coordinate assolute
- G91** Programmazione in coordinate relative
- G98** Ritorno al livello iniziale
- G99** Ritorno al livello R

[inglese]

The progress of electronic technology has revolutionized the metalworking industry. Thanks to the machine tool with numerical control has been possible to reduce considerably the time of preparation of the machine for the processing of different pieces with the simple substitution of a specific program stored on magnetic media. The continuing evolution of electronic technology has not spared the machine tool, passed from the first simple machining from point to point to the execution of any profile, even the most complex, thanks to the delegation of all the operations of calculation and verification of a computer: it has obtained the CNC (Numerical Control Computerized).

It's appropriate to illustrate the structural differences between a traditional machine tool and a machine tool with numerical control.

In a conventional machine tool, such as the lathe, we can distinguish 3 systems main:

- Support system, consisting of basement, work bench and their guides;
- Spindle drive system: three-phase motor, drive belts and head assembly that encompasses all the mechanisms of maintenance of the spindle shaft chuck;
- Positioning system tool: saw carriage and related leaflets approach (with graduated verniers), lead screw and grooved bar for movement uniform for the execution of the thread, for the roughing or finishing and various levers for the change of speed of the mandrel and advances the tool.

With the machine tool computer numerical control the first two systems have remained unchanged; instead have undergone processing electronics and functional all the components listed in the third.

The CNC machine tool has undergone, ie, a transformation of the whole electromechanical positioning system and control of the tool and for controlling the rotation of the spindle.

The control is a technique that allows the production of finished parts and / or semi finished using machine tools to a slave processor of signals that control is commonly called CNC. The main functions of the signal processor are:

- Command;
- Control;
- Adjustment;

The command is represented by a logical sequence of alphanumeric codes which allow the working bodies of the machine tool to perform some of the movements necessary for the machining of the workpiece.

The ordered set of commands determines a work program.

The CNC has revolutionized the way we operate the machines and traditional tools of their tools, but it has greatly improved performance for the sake of accuracy, speed, positioning of specific tools, productivity gains and the repetitiveness of the operations.

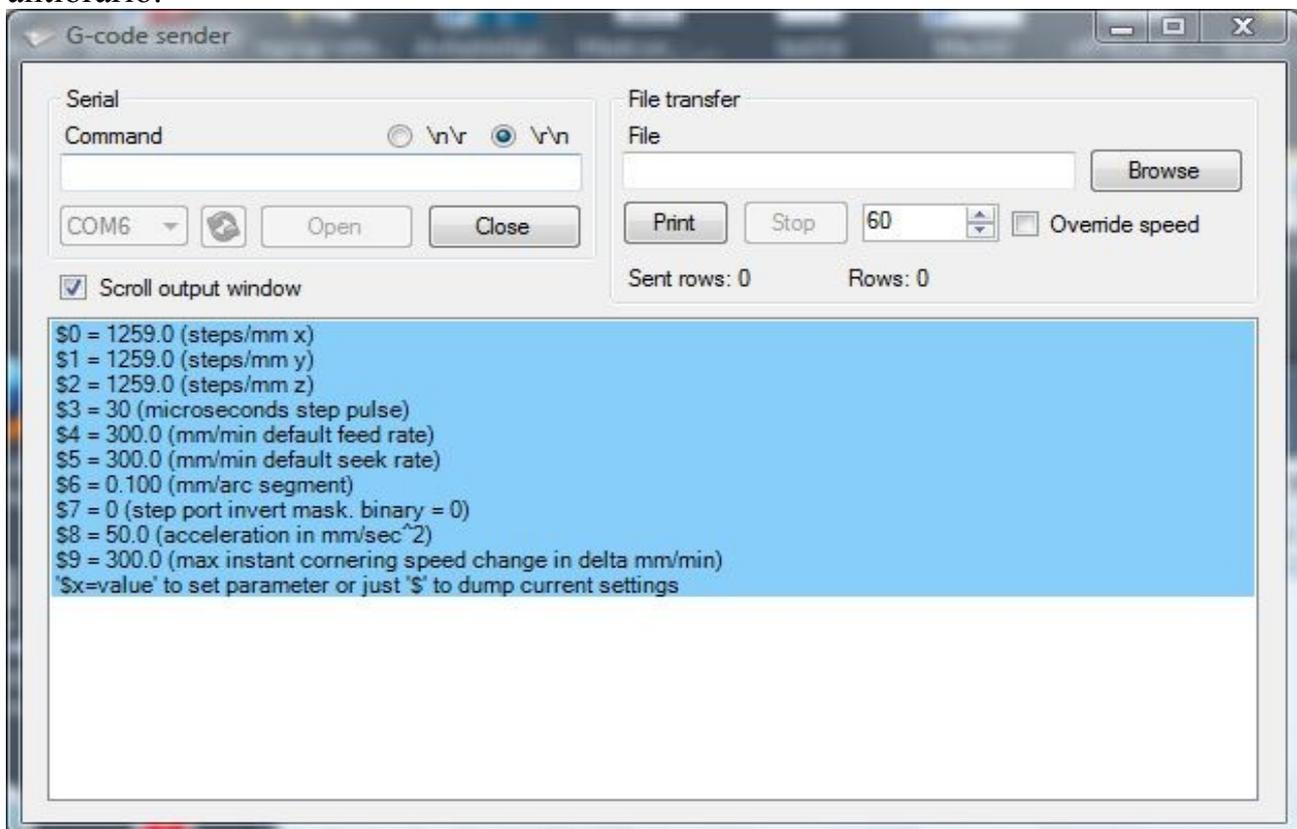
PROGRAMMA PER ARDUINO

IL nome del programma che è stato utilizzato per arduino è GRBL.

Questo programma è in grado di interpretare il g code(linguaggio macchina) e trasformarlo in una serie di input che una volta mandati ad arduino,riesce ad interpretarli ed a dare istruzioni dettagliate ai driver che piloteranno i motori utilizzando la modulazione PWM.

Queste istruzioni in particolare si chiamano STEP e DIR .

STEP sta ad indicare il numero dei passi che il motore deve compiere, ma non solo , indica pure la velocità con cui si deve muovere. Infatti se si applica un'oscilloscopio a questa uscita si noterà che esce un'onda quadra a diverse frequenze, infatti la frequenza è quella che *gestisce* la velocità. DIR invece serve a gestire la direzione del motore, anch'essa da arduino, con impulsi 1/0 che stanno ad indicare: livello logico alto(1) il motore gira in senso orario, livello logico basso (0) il motore girerà in senso antiorario.



Configurazione GRBL . Grbl ha bisogno di sapere quanto ogni passo sarà lo strumento nella realtà. Le impostazioni vengono salvate in eeprom e sarà mantenuto fino alla successiva modifica.

\$ 0, \$ 1 e \$ 2 - Passi mm /

\$ 0 set di assi X, \$ 1 set asse Y, insieme \$ 2 Asse Z

\$ 3 - Microsecondi / step impulsi.

I driver passo passo sono tarati per una certa lunghezza di impulso passo minimo. Se gli impulsi sono troppo lunghi si potrebbe incorrere in problemi al funzionamento del sistema a elevate velocità di avanzamento. Generalmente qualcosa tra 20 e 50

microsecondi.

\$ 4 - velocità di avanzamento di default.

Durante l'esecuzione di G-code questa impostazione di solito non entrano in scena come tutte le velocità di avanzamento saranno specificati nel file.

\$ 5 – indica la risoluzione.

\$ 6 - mm / segmento di arco.

Grbl rende cerchi e archi in linee sottili. Non sarà probabilmente mai necessario modificare questo valore - ma se si scopre che i cerchi sono troppo rudimentali si può cambiare questa impostazione. I valori più bassi offre una maggiore precisione, ma può portare a problemi di prestazioni.

\$ 7 - invert mask.

Il sistema di controllo può avere bisogno dei suoi valori invertiti. Linee di segnale sono normalmente alto e passa basso per un paio di microsecondi per segnalare un evento. Per raggiungere questo obiettivo ho bisogno di invertire alcuni dei bit. Questo valore è un byte che sono sottoposti a XOR con il passo-dati prima che viene inviato alla porta passo.

In questo modo è possibile utilizzare questo sia per invertire impulsi di passo (come faccio io) oppure di invertire una o più delle direzioni degli assi. I bit in questo byte corrisponde ai pin assegnati al passo in config.h. Per i bit di default vengono assegnate come segue:

```
# define X_STEP_BIT 0
# define Y_STEP_BIT 1
# define Z_STEP_BIT 2
# define X_DIRECTION_BIT 3
# define Y_DIRECTION_BIT 4
# define Z_DIRECTION_BIT 5
```

Se si vuole invertire la direzione X e Y in questa configurazione si dovrebbe calcolare un valore come questo (nel proprio ambiente di calcolo preferito):

>> (1 <Che è uguale a 24 in modo da eseguire questo comando li avrebbe invertire:

\$ 7 =24.

\$ 8 – accelerazione.

Questo è l'accelerazione in mm secondo per secondo. Non c'è bisogno di capire cosa significhi, basti dire che un valore inferiore dà l'accelerazione fluida, mentre un valore più alto produce più imprecisioni.

\$ 9 - max cambio istantaneo di velocità in curva.

Questo è il cambiamento immediato, cioè la velocità massima che il gestore di accelerazione consente. I valori più alti sono meno precisi. I valori più bassi rende il manager di accelerazione più attento e un lavoro migliore.

MOTORE PASSO-PASSO

Il motore passo-passo spesso chiamato anche step o stepper è un motore elettrico sincrono in corrente continua senza spazzole che può suddividere la propria rotazione in un grande numero di step.

La posizione del motore può essere controllata accuratamente senza dover ricorrere al controllo ad anello chiuso (feedback) se la taglia ed il tipo di motore sono scelti in modo adeguato all'applicazione.

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.

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. Un motore ibrido è costituito da un ROTORE e da uno STATORE.

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 vall dell'altra. Nel rotore non sono presenti fili elettrici e quindi manca completamente ogni connessione elettrica tra la parte in movimento e quella fissa.

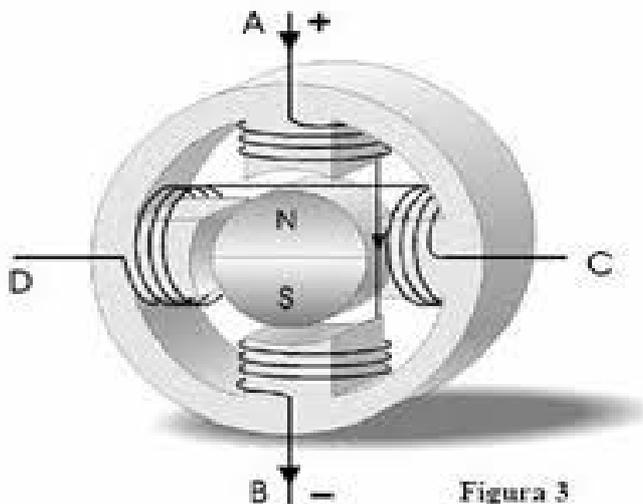


Figura 3

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". 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.





ARDUINO

L'hardware originale Arduino è realizzato dalla italiana *Smart Projects*. Alcune schede Arduino sono state progettate dalla statunitense *SparkFun Electronics*. Una scheda Arduino tipica consiste in un microcontroller a 8-bit AVR prodotto dalla Atmel, con l'aggiunta di componenti complementari per facilitarne l'incorporazione in altri circuiti. In queste schede sono usati chip della serie megaAVR - nello specifico i modelli ATmega8, ATmega168, ATmega328, ATmega1280 e ATmega2560.

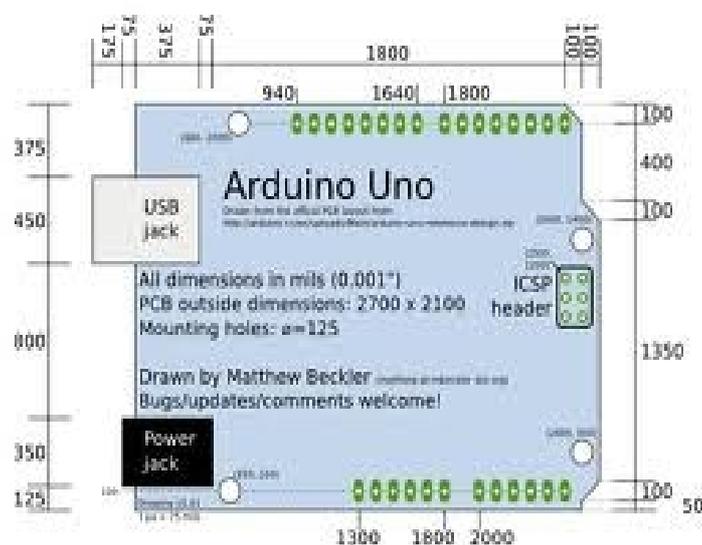
La programmazione del microcontroller

Inoltre, il microcontroller della scheda in alcuni casi è pre-programmato con un bootloader che semplifica il caricamento dei programmi sulla memoria flash incorporata nel chip. A livello concettuale, tutte le schede sono programmate attraverso una porta seriale RS-232, ma il modo in cui questa funzionalità è implementata nell'hardware varia da versione a versione. Le schede seriali Arduino contengono un semplice circuito inverter che permette la conversione tra il livello della RS-232 e il livello dei segnali TTL.

Le versioni attuali di Arduino sono gestite via usb: la versione, “**uno**”, utilizza un microcontrollore Atmega8U2 programmato come convertitore USB-seriale mentre le precedenti versioni *diecimila* e *duemilanove* usavano chip adattatori USB-seriale, come gli FT232 di FTDI alcune varianti, come la Arduino Mini e la versione non ufficiale *Boarduino*, usano una scheda o un cavo adattatore USB-seriale staccabile.

Funzionalità di Input/Output

Per implementare il comportamento interattivo, Arduino è fornita di funzionalità di Input/Output, grazie alle quali essa riceve i segnali raccolti da sensori esterni. In base a tali valori, il comportamento della scheda è gestito dal microcontroller, in base alle decisioni determinate dal particolare programma in esecuzione in quel momento sulla scheda. L'interazione con l'esterno avviene attraverso attuatori pilotati dal programma attraverso i canali di output in dotazione.



Alimentazione elettrica

L'alimentazione della scheda può avvenire attraverso la porta USB del computer, o attraverso la maggior parte degli alimentatori USB, oppure attraverso un adattatore in corrente continua a 9 volt, con connettore cilindrico (diametro 2,1 mm e positivo centrale). In quest'ultimo caso, la scheda commuta automaticamente sull'alimentazione esterna quando il connettore dell'alimentatore esterno è inserito, mentre commuta autonomamente sull'alimentazione USB in caso di disconnessione del connettore. La Arduino-NG e la Arduino Diecimila, versioni meno recenti, necessitano di essere commutate a mano, azionando uno switch ubicato tra la porta USB e l'ingresso dell'alimentazione esterna.

La programmazione del microcontroller

La programmazione può avvenire tramite il protocollo ISP e un'altra scheda Arduino utilizzata come programmatore, tramite la usb (in alcuni casi il microcontrollore contiene usb hardware) oppure grazie a un programmatore esterno dedicato.



DRIVER

I driver sono dei circuiti elettrici in grado di interpretare gli impulsi STEP e DIR . Infatti,all'uscita,dove verranno collegati i motori , avremo una sorta di contatore ad anello che dando la fase alternatamente a tutte le induttanze, il motore girerà secondo il codice macchina,eseguendo così il lavoro.

Quello montato sulla scheda della mia cnc e costituito da:

un CONTATORE binario a otto bit (74191).

un DEMULTIPLEXER(74139).

quattro porte NOT(7408).

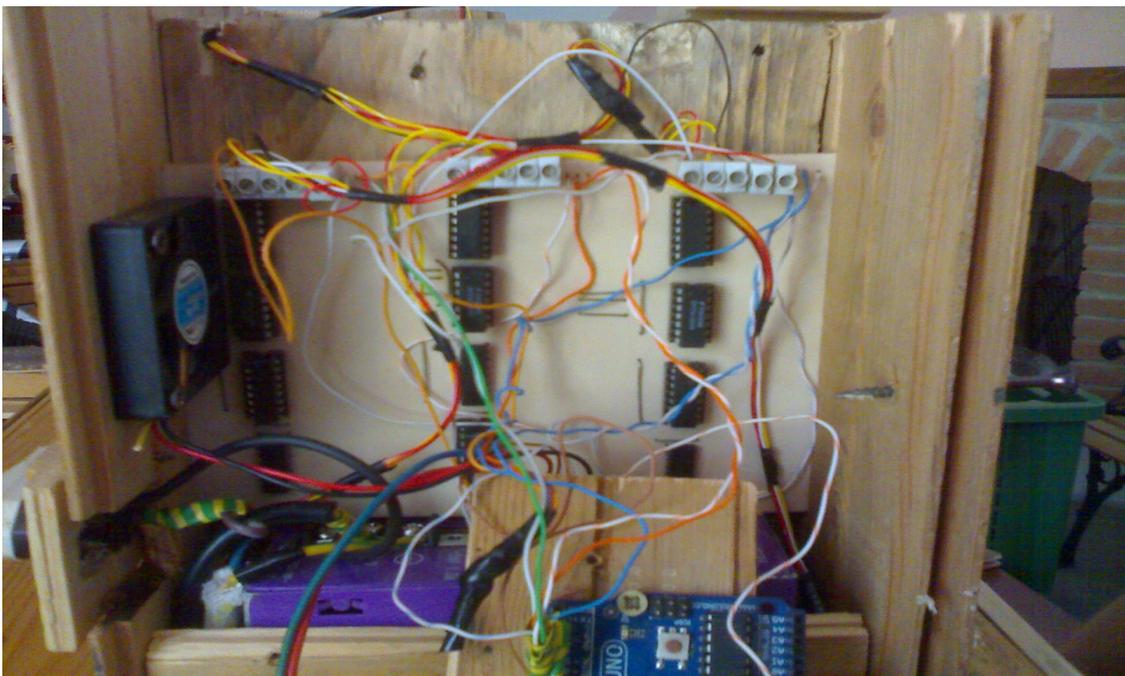
e da quattro transistor che ci forniscono le uscite di potenza(uln2003).

Il contatore è collegato direttamente ad arduino che gli fornisce il clock (riferito a STEP) e l'UP/DOWN (riferito a DIR).

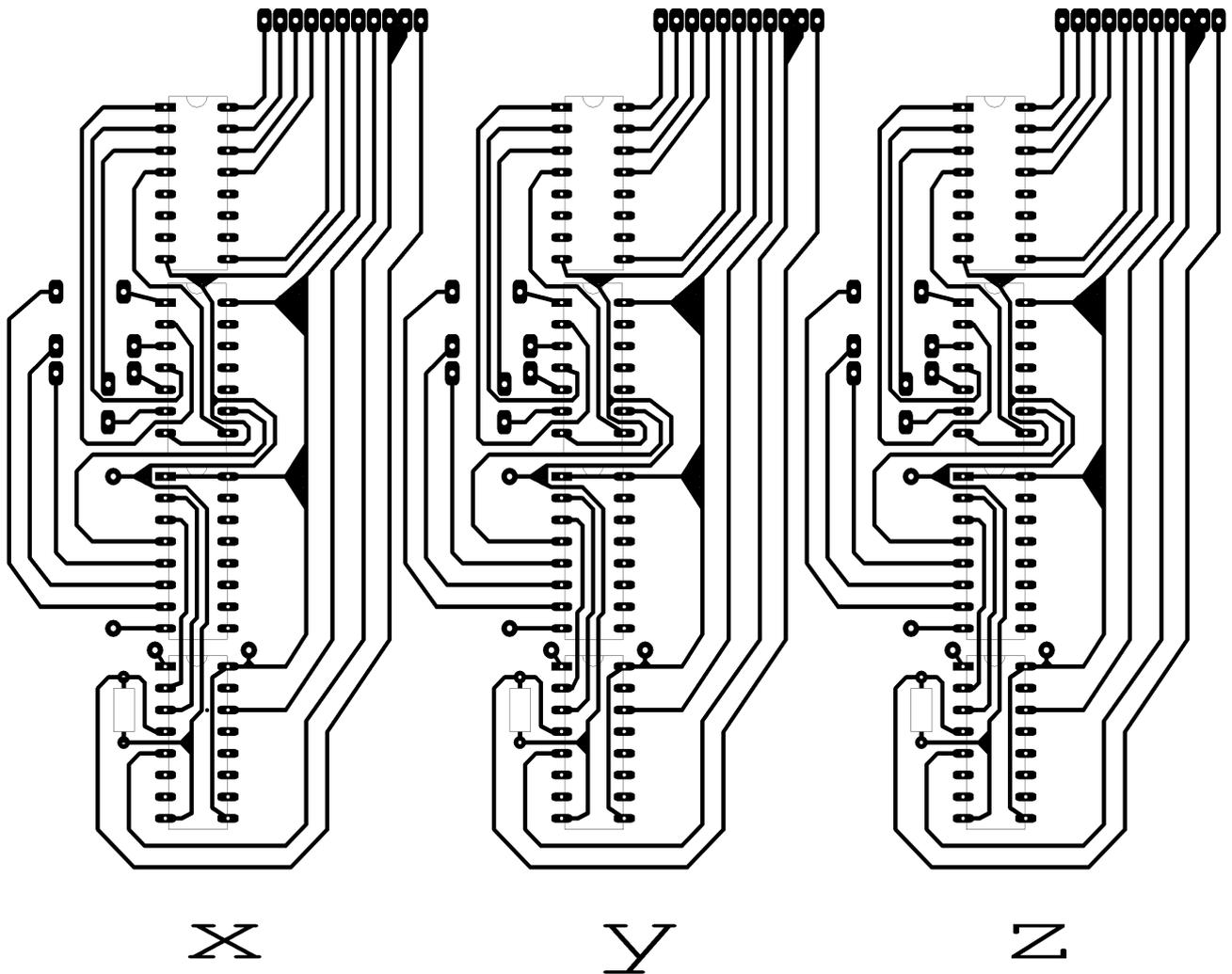
Ricevendo il segnale step inizia a formare quattro combinazioni binare (0000, 0001, 0010, 0011), che vengono mandate in modo seriale al demultiplexer, che leggendole abilita una per volta le sue uscite in parallelo, formando così una sorta di contatore ad anello. Arrivati a questo punto non possiamo collegare subito l'uscita di potenza, perchè abbiamo un segnale di questo tipo (0111), che non va bene per le mie esigenze di pilotaggio a passo intero. Dunque, per ottenere il segnale voluto introduciamo delle porte NOT che ci invertono gli uni coi zeri e viceversa, ottenendo un segnale di questo tipo (1000).

Il segnale così modulato può essere mandato ai transistor, utilizzati come switch e non come amplificatori, in modo da chiuderci il contatto con la corrente dei motori (12V-2A), e collegare il tutto alle fasi del motore.

Il circuito sopra descritto può pilotare solamente un motore, quindi avremo bisogno di crearne tre, uno per motore.



CIRCUITO ELETTRICO



Il circuito rappresentato in figura, rappresenta il DRIVER precedentemente descritto. Nella parte di sopra sono rappresentati i contatti dove verranno collegate le correnti (5-12V) ed i motori passo passo (step 200).

Il primo integrato ci fornisce le uscite di POTENZA, il secondo le porte NOT, il terzo il DEMULTIPLEXER ed il quarto il CONTATORE BINARIO.

STRUTTURA CNC

Una volta completata l'elettronica, bisogna pensare alla parte meccanica, a cui i motori trasmettono il moto.

La struttura è formata da un tavolo a croce, che utilizza due viti senza fine per muovere la struttura avanti-dietro

/ destra-sinistra formando gli assi X ed Y. I motori sono attaccati ad esse attraverso delle boccole, formate da gomma dura e da due fascette d'acciaio regolabili, e alloggiati in scheletri d'alluminio per dissipare meglio il calore da loro prodotto.

Il tavolo è composto invece da alluminio, con tre feritoie per far scorrere due piccoli morsetti che serviranno poi a tenere bel saldo il pezzo da lavorare al tavolo. L'asse Z è invece formato:

da due pistoncini che fungono da liste assicurandoci un andamento rettilineo una vite senza fine collegata al motore sempre attraverso le boccole di gomma e l'alloggio in legno della fresa, che collega il tutto.

Il motore passo passo è invece alloggiato alla base della struttura, con quattro viti collegate ad uno scatolato d'alluminio, che ne fa il cuore della struttura.

La parte elettrica è posizionata nella parte sottostante del tavolo, racchiusa da un infisso di legno per coprire il tutto.

Una ventola poi si occupa di far circolare l'aria all'interno, in modo da far fuoriuscire il calore provocato dalle uscite di potenza che riscaldano notevolmente.

Per quanto riguarda la parte della sicurezza è stato montato un coperchio in plexiglas per garantire la protezione del gestore da parte di eventuali rotture di parti in movimento, infatti se durante la lavorazione questo viene aperto i motori cesseranno di lavorare. Per assicurarci del corretto funzionamento del macchinario sono state aggiunte due spie; la spia rossa segnala un problema di qualsiasi entità impedendo quindi al macchinari di lavorare, la verde invece segnala che è possibile effettuare la lavorazione.



Scuola superiore “Alessandro volta”

Nuoro

Tesina di maturità

CNC:
Control
Numeric
Computer

Nome:Paolo Piras

Classe:5°B

Anno scolastico 2011/2012