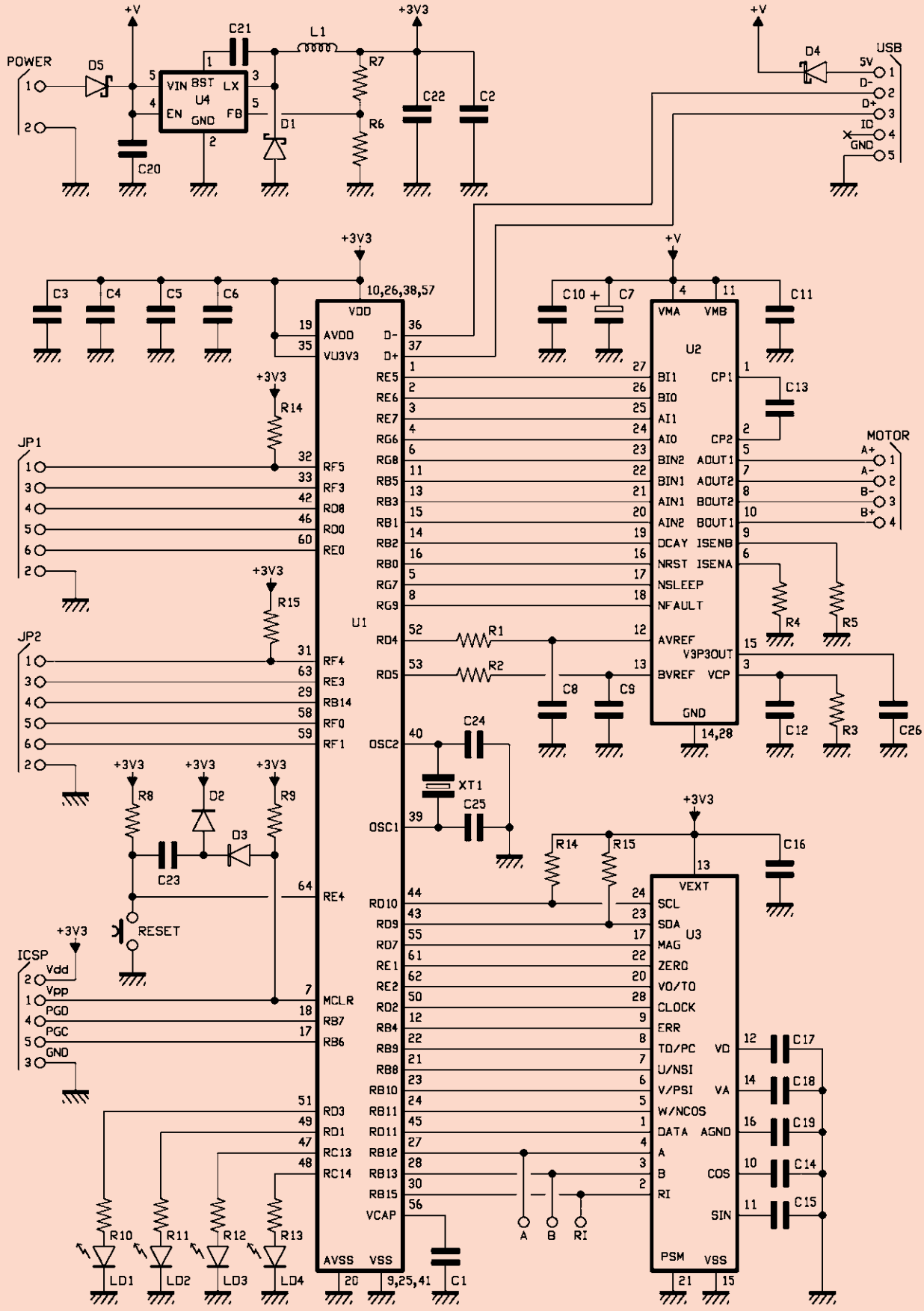
MOTORFISH:
LO STEPPER NON SBAGLIA UN PASSO

Controller con retroazione per motore passo-passo in grado di verificare dinamicamente il posizionamento dell'albero e prevenire eventuali perdite di passo.

di **MASSIMO DEL FEDELE**

Chi non è mai capitato di lanciare una stampa 3D complessa e lunga, magari un lavoro di ore o decine di ore, assentarsi e scoprire che, a lavoro quasi terminato, i motori della stampante hanno perso qualche passo col risultato di avere l'oggetto rovinato stampato male con degli strati non allineati tra loro? Purtroppo non è infrequente e la cosa sorprendente (o irritante...) è che succede quasi sempre verso la fine della stampa, causando notevole spreco di materiale e di tempo.
Siccome la stragrande maggioranza delle stampanti 3D in commercio utilizza dei motori passo-passo controllati ad anello aperto, la cosa è inevitabile perché il controller non ha modo di accertare se l'ordine impartito viene correttamente eseguito dal motore, ovvero se ad ogni impulso inviato il rotore avanza effettivamente di un passo.
Per sopperire questa carenza abbiamo deciso di realizzare un controller, chiamato Motorfish, in grado di gestire localmente (a stretto contatto...) un motore passo-passo dotato di encoder, quindi di operare un controllo a reazione (o retroazione) in grado di verificare che la posizione angolare dell'albero dello stepper motor sia coerente con gli impulsi di pilotaggio.
Prima di analizzare lo schema elettrico è opportuno chiarire come funzionano i due tipi di controllo di posizione dei motori, che sono ad anello aperto e ad anello chiuso.
Nel primo caso, anche noto come open loop, il controller invia un comando al motore senza conferma dello spostamento effettivo; se il motore per un qualsiasi motivo non è stato in grado di spostarsi (ad esempio se la testina di stampa ha incontrato un ostacolo oppure se la velocità di spostamento richiesta è troppo elevata) l'errore viene mantenuto e si accumula con eventuali altri errori successivi.
L'elettronica non ha alcun modo di conoscere la reale posizione del motore, col risultato di avere delle stampe magari "sfalsate" da metà lavoro in poi.
Nel controllo ad anello chiuso (closed loop, **Fig. 1**) viene impartito un comando di spostamento al motore, ma in questo caso un sensore di posizione sul medesimo è in grado di fornire una "conferma" dello spostamento avvenuto o un segnale di errore se questo è impossibile. Il controller può quindi decidere se fermare tutto, riprovare oppure andare avanti cercando di correggere il problema in seguito.
Il sistema ad anello chiuso ha parecchi vantaggi sul primo; innanzitutto, lo spostamento non dipende più dalla tipologia del motore, ma soltanto dalla risoluzione del sensore di


POWER
D5
+V
C21
L1
+3V3
R7
C22
C2
U4
VIN BST LX
EN GND FB
C20
D1
R6
D4
USB
5V
D-
D+
ID
GND
C3
C4
C5
C6
10,26,38,57
VDD
AVDD
VU3V3
D-
D+
U1
C10
C7
C11
VMA
VMB
U2
BI1
BI0
AI1
AI0
BIN2
BIN1
AIN1
AIN2
DCAY
NRST
NSLEEP
NFAULT
AVREF
BVREF
CP1
CP2
AOUT1
AOUT2
BOUT2
BOUT1
ISENB
ISENA
V3P3OUT
VCP
GND
14,28
C13
MOTOR
A+
A-
B-
B+
R14
R15
JP1
JP2
RE5
RE6
RE7
RG6
RG8
RB5
RB3
RB1
RB2
RB0
RG7
RG9
RD4
RD5
R1
R2
C8
C9
R4
R5
C12
R3
C26
C24
XT1
C25
OSC2
OSC1
R8
D2
D3
R9
C23
RESET
ICSP
Vdd
Vpp
PGD
PGC
GND
RE4
MCLR
RB7
RB6
RD3
RD1
RC13
RC14
R10
R11
R12
R13
LD1
LD2
LD3
LD4
AVSS
VSS
20
9,25,41
C1
RD10
RD9
RD7
RE1
RE2
RD2
RB4
RB9
RB8
RB10
RB11
RD11
RB12
RB13
RB15
VCAP
U3
VEXT
SCL
SDA
MAG
ZERO
VO/TO
CLOCK
ERR
TD/PC
U/NSI
V/PSI
W/NCOS
DATA
A
B
RI
VD
VA
AGND
COS
SIN
PSM
VSS
21
15
C16
C17
C18
C19
C14
C15
A
B
RI

movimento (un **encoder**, abitualmente). Quindi non saremo più obbligati a lavorare per passi discreti (step) o micropassi (microstep) ma potremo avere un funzionamento continuo del motore, con minori vibrazioni meccaniche.
Un secondo vantaggio è che, non essendo a rischio di perdita di posizione, possiamo "spingere" il motore al massimo senza preoccuparci di nulla o quasi; se il motore non riesce a seguire i comandi l'elettronica se ne accorge ed è in grado di rimediare, sia riducendo velocità e/o accelerazione sia correggendo gli errori.
Il terzo vantaggio è un ridotto consumo di corrente. In un motore passo-passo a ciclo aperto siamo infatti obbligati a mantenere una corrente elevata negli avvolgimenti anche a motore fermo, per evitare movimenti indesiderati dovuti magari a masse in rapida decelerazione oppure a manomissioni involontarie del piatto della stampante; nel motore a ciclo chiuso, quando non è in movimento, possiamo ridurre a zero la corrente negli avvolgimenti, sicuri che in caso di spostamenti involontari l'elettronica sarà in grado di correggere immediatamente la posizione.
Ultimo, ma non in ordine di importanza, è la possibile semplificazione del motore stesso. Non sarà il nostro caso, visto che utilizzeremo comunque un motore passo-passo, ma volendo è possibile sostituirlo con un qualsiasi altro tipo di motore, anche a spazzole o brushless, cambiando semplicemente l'elettronica e/o il software di controllo.
Come vedremo in seguito, il nostro MotorFish è in grado di controllare anche un motore a spazzole (brushed) modificando semplicemente il software del controller. I motori brushed utilizzano un solo "avvolgimento" (non è proprio così, ma questo è ciò che appare all'esterno), ed il nostro driver è in grado di pilotarne fino a 2; purtroppo non sarà possibile pilotare motori a 3 avvolgimenti.

SCHEMA ELETTRICO

Dopo il lungo preambolo sul controllo a retroazione passiamo a vedere come viene implementato, analizzando il circuito che lo realizza.
Iniziamo con la descrizione della sezione di alimentazione; per evitare di "portarci in giro" troppi fili abbiamo infatti previsto un piccolo alimentatore switching sulla scheda, in grado di fornire i 3,3 volt necessari all'elettronica digitale direttamente dalla tensione elevata utilizzata per pilotare il motore.
I motori passo-passo, infatti, per compensare i ritardi dovuti all'elevata induttanza degli avvolgimenti, sono solitamente pilotati da driver in modalità PWM con tensioni piuttosto elevate. Il nostro alimentatore è in grado di funzionare con tensioni in ingresso da circa 5-6 Volt fino a 42 volt massimi, coprendo quindi la gamma delle tensioni utilizzate abitualmente nei motori passo-passo.

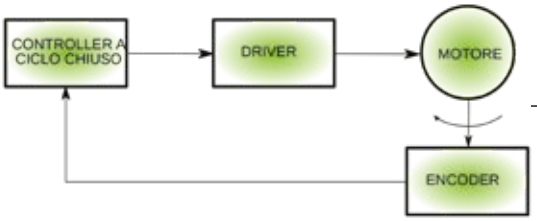


Fig. 1 Schematizzazione del controllo ad anello chiuso.

Si tratta di un semplice convertitore buck (abbassatore), costituito da un integrato LMR14206, una piccola induttanza, un diodo e pochi altri componenti passivi.
Abbiamo scelto una circuitazione switching poiché con una differenza elevata tra tensione in ingresso ed in uscita uno schema con regolatore lineare sarebbe molto inefficiente e richiederebbe un dissipatore di calore di elevate dimensioni, mentre nel nostro caso si raggiunge un'efficienza vicina al 90% con un riscaldamento praticamente nullo.
L'integrato lavora a 1,2 MHz, cosa che permette di utilizzare componenti di dimensioni ridottissime, soprattutto per quanto riguarda l'induttanza; per contro, è necessaria una cura meticolosa nello studio del circuito stampato, utilizzando collegamenti corti, ampie zone di massa e componenti molto vicini al circuito integrato.
La regolazione della tensione in uscita avviene attraverso il pin di feedback (FB), il quale è impostato internamente per una tensione di 0,765 volt; occorre quindi un partitore resistivo che fornisca quel valore quando la tensione in uscita raggiunge i 3,3 volt richiesto.
Il nostro partitore, costituito da R7 (4,4 kohm) e R6 (1,02 kohm) consente di ottenere una tensione in uscita pari a:

$$V_o = V_{FB} \cdot \frac{R_6 + R_7}{R_6} = 0.765\,V \cdot \frac{3.4\,K + 1.02\,K}{1.02\,K} = 3.315\,V$$

quindi molto vicina ai 3,3 V.

IL CONTROLLER

Per gestire tutti i componenti (che come vedremo a breve non sono pochi!) con la necessaria velocità di risposta abbiamo optato per un controller a 32 bit, per la precisione una PIC della serie 32MX, dotata di interfaccia USB per facilitarne la programmazione e la gestione tramite un PC.

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

Elettronica In

oltre l'elettronica

www.elettronicain.it