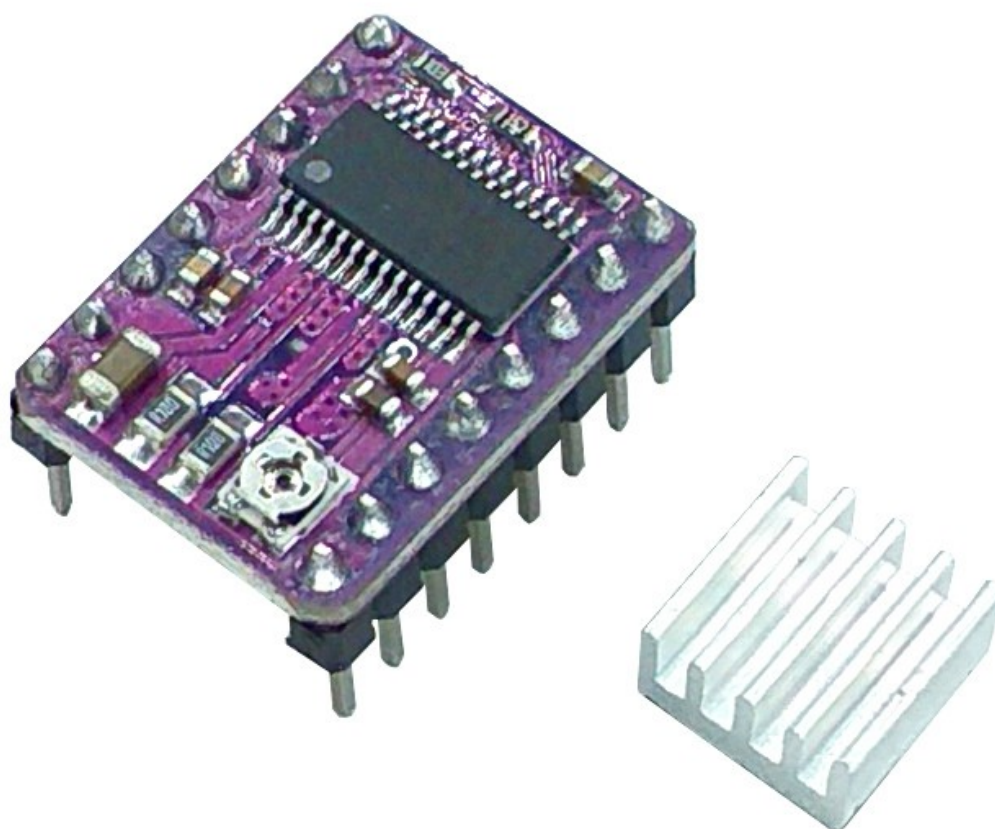




Benvenuto!

Grazie per aver acquistato il nostro Driver Per Motore Passo Passo DRV8825 AZ-Delivery. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!



Az-Delivery

Il motore passo-passo è un tipo di motore in cui l'albero del motore ruota a passi. Il motore passo-passo è un motore DC senza spazzole, vale a dire un motore DC brushless. Spostare l'albero a passi è molto utile perché l'albero può essere posizionato in modo molto preciso senza alcuna misurazione della posizione di feedback.

Tutti gli elettromotori sono costituiti da rotore e statore. Nei motori passo-passo, il rotore è generalmente un magnete permanente, che è circondato dalle bobine dello statore. Per ruotare un rotore, è necessario attivare o disattivare le bobine dello statore in un determinato ordine. Quando la corrente scorre attraverso le bobine dello statore, viene prodotto un campo elettromagnetico, che ruota il rotore o "magnete permanente" nella direzione del campo elettromagnetico.

Di costruzione ci sono tre diversi tipi di motori passo-passo:

- » motori passo-passo a magneti permanenti,
- » motori passo-passo a riluttanza variabile e
- » motori passo-passo ibridi sincroni.

(non tratteremo le costruzioni dei motori passo-passo in questo eBook)



Modalità di azionamento dei motori passo-passo

Per azionare il motore passo-passo, ci sono diverse modalità di azionamento o modalità di eccitazione:

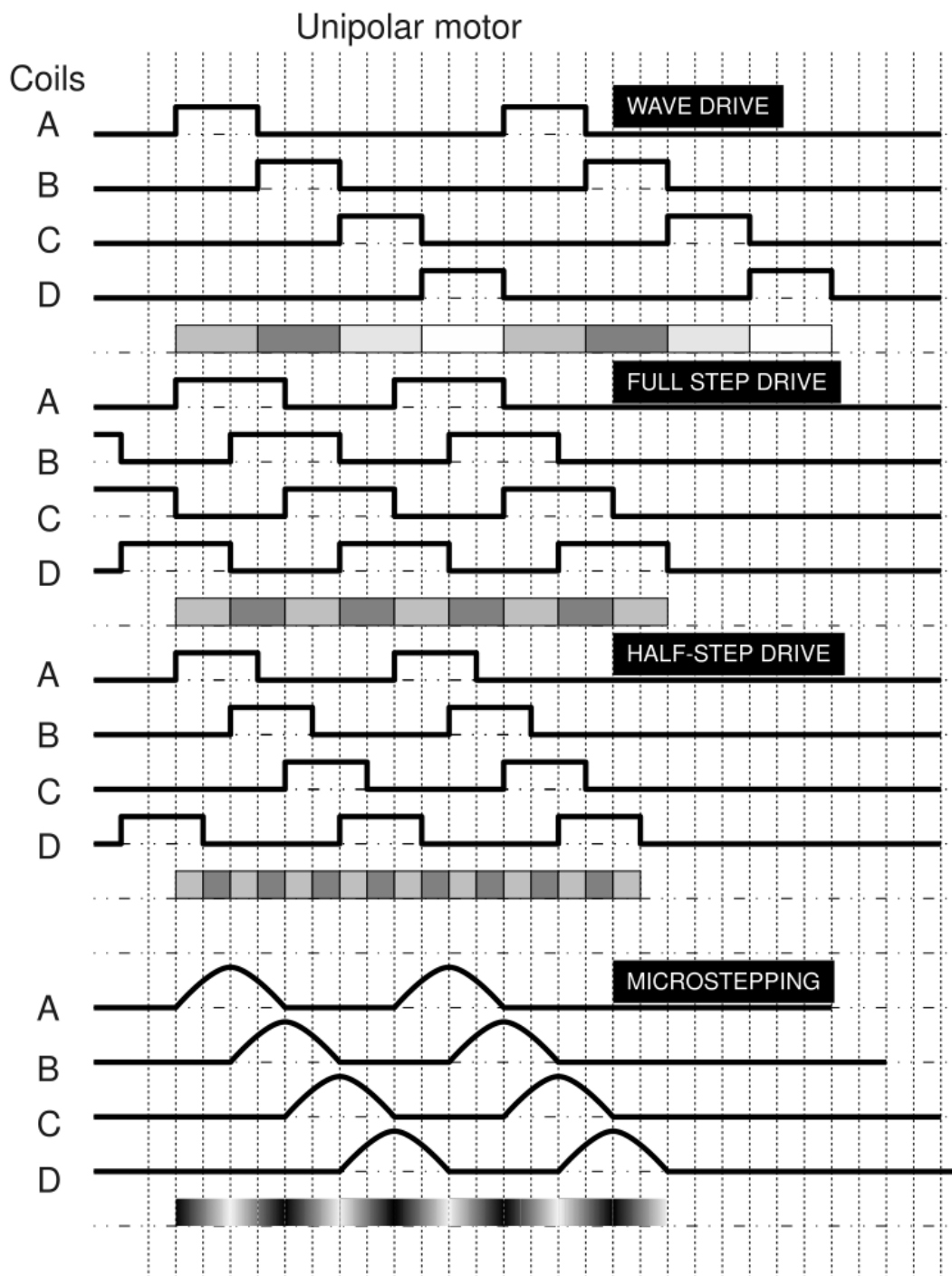
» **Modalità azionamento ad onda**, in questa modalità attiviamo solo una bobina dello statore alla volta, quindi la successiva e così via. In questa modalità, viene attivata solo una bobina e per spostare il rotore al passaggio successivo, accendiamo le bobine una alla volta consecutivamente. Quando accendiamo la seconda bobina, la prima bobina viene spenta, ecc.

» **Modalità azionamento a passo completo** fornisce una coppia molto più elevata perché abbiamo sempre due bobine attive in un dato momento. Questa modalità è meglio spiegata sull'immagine nella pagina successiva.

» **Modalità azionamento a mezzo passo** viene utilizzata per aumentare la risoluzione del motore passo-passo. Questa modalità deriva dalla combinazione delle due precedenti. Qui attiviamo una bobina e vicino alla fine dello stato attivo di quella bobina, attiviamo la bobina successiva. Quando è attivata la seconda bobina, spegniamo la prima bobina e così via. Con questa modalità otteniamo il doppio della risoluzione con la stessa struttura del motore.

» **Modalità microstepping** è il metodo più comune per controllare i motori passo-passo al giorno d'oggi. In questa modalità forniamo corrente variabile alle bobine sotto forma di onde sinusoidali. Ciò fornirà un movimento regolare del rotore, diminuirà lo stress delle parti e aumenterà la precisione del motore passo-passo.

Un altro modo per aumentare la risoluzione del motore passo-passo è aumentare il numero di poli del rotore e il numero di poli dello statore.



L'immagine è tratta dall'articolo di Wikipedia sui motori passo-passo.

https://en.wikipedia.org/wiki/Stepper_motor



Driver motori passo-passo

Se vuoi realizzare un dispositivo in cui dovrai controllare con precisione la rotazione dell'albero motore, come nelle stampanti 3D o in qualsiasi altra macchina CNC, o bracci robotici, ecc., avrai bisogno di un gruppo di motori passo-passo e, soprattutto, il driver per motori passo-passo.

Usare Microcontrollore per controllare un gruppo di motori passo-passo è poco pratico e nella maggior parte dei casi impossibile. Quindi abbiamo bisogno di un circuito elettronico del driver per ogni motore passo-passo.

In questo eBook tratteremo uno di questi dispositivi, chiamato "*driver del motore passo-passo DRV8825*". Questo dispositivo può controllare la velocità di rotazione e la direzione di rotazione dell'albero motore e può guidare il motore passo-passo in diverse modalità di eccitazione. Possiamo persino usare questo dispositivo per le modalità di eccitazione microstepping. Con questo driver per motori passo passo possiamo controllare solo motori passo passo bipolari.

Questo è un driver per microstepping, con chip principale come circuito integrato chiamato "DRV8825" e prodotto da "Texas Instruments". Il driver ha un traduttore integrato per un facile utilizzo. Ciò riduce il numero di pin di controllo a solo due, uno per il controllo dei passi e l'altro per il controllo della direzione di rotazione. Il driver offre sei diverse risoluzioni di passo o modalità di eccitazione: passo completo e quattro modalità di microstepping: mezzo passo, quarto passo, ottavo passo, sedicesimo passo e passo a trenta secondi.

Az-Delivery

Il driver del motore passo-passo DRV8825 ha una capacità di azionamento in uscita fino a 45V e consente di controllare un motore passo-passo bipolare con una corrente di uscita fino a 2,2 A per bobina.

Specifiche

- » Tensione di uscita min-max. del motore: 8.2V – 45V
- » Voltaggio logico: 3.3V - 5V
- » Corrente nominale per fase: 1.5A
- » Corrente massima per fase: 2.2A con raffreddamento passivo, dissipatore di calore in alluminio
- » Dimensioni: 16 x 21mm

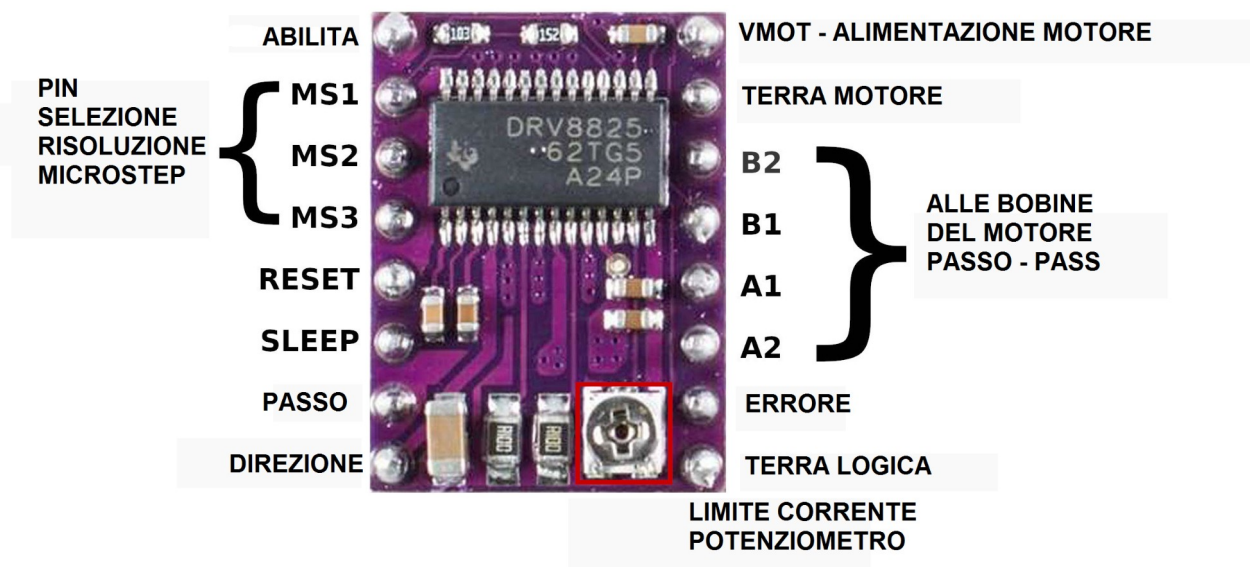
Il driver richiede solo una connessione di alimentazione. Questo modulo non ha alcun pin di alimentazione logico poiché il chip "*DRV8825*" viene alimentato dal regolatore di tensione interno da 3,3V. Tuttavia, è necessario collegare la terra del microcontrollore con il pin *GND LOGIC*.

VMOT e *GND MOT* vengono utilizzati per alimentare il motore, da 8,2V a 45V. Secondo la scheda tecnica, l'alimentazione del motore richiede un adeguato condensatore di disaccoppiamento vicino alla scheda, in grado di sostenere correnti fino a 4A.

ATTENZIONE: questo driver ha a bordo condensatori ceramici a basso ESR, il che lo rende vulnerabile a picchi di tensione. In alcuni casi, questi picchi possono superare i 35V (tensione nominale massima dell' A4988), che può potenzialmente danneggiare in modo permanente la scheda e/o il motore passo-passo.

Un modo per proteggere il driver da tali picchi è quello di inserire un grande condensatore elettrolitico da $100\mu F$ (o almeno $47\mu F$) attraverso i pin di alimentazione del motore.

Piedinatura del DRV8825





Pin Selezione Microstep

Il driver DRV8825 consente il microstepping consentendo posizioni di passo più piccole. Ciò si ottiene alimentando le bobine con corrente variabile. Ad esempio, se si sceglie di azionare il motore passo-passo “*NEMA17*”, ha un angolo di passo di $1,8^\circ$ o 200 passi per giro; in modalità a un quarto di passo, il motore fornirà 800 microstep per giro completo.

Il driver DRV8825 ha tre pin di input del selettore di risoluzione microstep:

- » MS1
- » MS2
- » MS3

Impostando i livelli logici appropriati su questi pin possiamo impostare la modalità di azionamento del motore su una di queste cinque modalità:

M0	M1	M2	Risoluzione Microstep
LOW	LOW	LOW	Passo pieno
HIGH	LOW	LOW	Mezzo passo
LOW	HIGH	LOW	Quarto di passo
HIGH	HIGH	LOW	Ottavo di passo
LOW	LOW	HIGH	Sedicesimo passp
HIGH	LOW	HIGH	1/32 di passo
HIGH	HIGH	HIGH	1/32 di passo
HIGH	HIGH	HIGH	1/32 di passo

Questi tre pin di selezione microstep sono tirati in *LOW* da resistenze pull-down interne, quindi se li lasciamo tutti scollegati, il motore funzionerà in modalità full step.



Differenza tra modalità microstepping e full step

Le modalità di eccitazione del microstepping sono tutte modalità in cui l'albero del motore si sposta tra i passi hardware. Queste modalità posizionano l'albero del motore tra i passi, creando più passi e un movimento regolare dell'albero.

La modalità di eccitazione a mezzo passo è una combinazione di passo completo e azionamento ad onda. Ciò si traduce in metà dell'angolo di passo base. Questo angolo di passo più piccolo offre un funzionamento più fluido grazie alla maggiore risoluzione dell'angolo. Il mezzo passo produce circa il 15% di coppia in meno rispetto al passo intero, tuttavia il mezzo passo modificato elimina la diminuzione della coppia aumentando la corrente applicata al motore quando una singola bobina viene eccitata.

Il microstepping può dividere il passo base di un motore fino a 256 volte, rendendo più piccoli i passi piccoli. Un azionamento microstep utilizza due onde sinusoidali a corrente variabile distanti 90° , ciò è perfetto per consentire un funzionamento regolare del motore. Noterai che il motore funziona in modo silenzioso e senza alcuna reale azione di passo rilevabile.

Controllando la direzione e l'ampiezza del flusso di corrente in ciascuna bobina dello statore, la risoluzione aumenta e le caratteristiche del motore migliorano, con meno vibrazioni e un funzionamento più fluido. Poiché le onde sinusoidali lavorano insieme, c'è una transizione graduale da una bobina all'altra. Quando aumenta la corrente in una bobina, la corrente nella bobina successiva diminuisce, il che si traduce in una progressione graduale del passo e nel mantenimento della coppia erogata.



Pin di passo e direzione

Il pin di **PASSO** controlla i passi del motore. Ogni impulso HIGH inviato a questo pin fa avanzare il motore in base al numero di microstep impostati dai pin di selezione microstep. Più veloci sono gli impulsi, più veloce ruoterà il motore.

Il pin **DIR** controlla il senso di rotazione dell'albero motore. Tirandolo in *HIGH* si aziona l'albero in senso orario e tirandolo in *LOW* si aziona l'albero in senso antiorario. Se si desidera solo ruotare l'albero in una sola direzione, è possibile collegare il pin *DIR* direttamente a *VCC* o *GND* a seconda.

NOTA: i pin **STEP** e **DIR** non sono collegati internamente a una particolare tensione, quindi non lasciarli fluttuanti nell'applicazione.



Pin di abilitazione dell'alimentazione

Pin **EN** , se tirato su *LOW* il driver è abilitato. Per impostazione predefinita, questo pin è *LOW* in modo che il driver sia sempre abilitato, a meno che non lo si tenga esplicitamente *HIGH* per disabilitare il driver.

Pin **SLP** , tirando questo pin *LOW* si mette il driver in modalità sleep, riducendo al minimo il consumo di energia. È possibile utilizzarlo quando il motore non è in uso, per risparmiare energia.

Pin **RST**, quando viene tirato *LOW*, tutti i pin *STEP* vengono ignorati, fino a quando non si tira su *HIGH*. Ripristina anche il driver impostando il traduttore interno su uno stato "home" predefinito. Lo stato "Home" è la posizione iniziale da cui parte il motore e differisce in base alla risoluzione del microstep.

NOTA: se non si utilizza il pin *RST*, è possibile collegarlo al pin *SLEEP* per portarlo *HIGH* e abilitare il driver.



Pin rilevazione errori

Il driver dispone anche di un pin di uscita *FAULT*. Questo pin è *LOW* ogni volta che i FET H-bridge interni all'interno del chip "*DRV8825*" sono disabilitati a causa di una protezione da sovracorrente o arresto termico.

In realtà il pin *FAULT* è in cortocircuito sul pin *SLEEP*, quindi, ogni volta che il pin *FAULT* è *LOW*, l'intero chip è disabilitato. E rimane disabilitato fino a quando non viene rimosso o riapplicato il *RESET* o il voltaggio motore sul pin *VMOT*.

Pin di uscita

I pin di uscita del driver sono *A1*, *A2*, *B1* e *B2*. Ogni pin di uscita può fornire correnti fino a 2,2A. Tuttavia, la quantità di corrente fornita al motore dipende dall'alimentazione, dal sistema di raffreddamento e dalle impostazioni di limitazione della corrente.

Se non si conosce la piedinatura del motore passo-passo bipolare, è possibile testarlo utilizzando il multimetro. Posizionare l'interruttore sul multimetro per il test di continuità (posizione etichettata con il segno sonoro). Innanzitutto, verifica se il multimetro funziona collegando entrambi i nodi del multimetro insieme. Dovresti sentire un segnale acustico se funziona. Il motore passo-passo bipolare ha due bobine e quattro fili; due fili per bobina (uno per l'ingresso e uno per l'uscita della bobina). Una bobina è solo un filo avvolto, quindi controlliamo la continuità tra due fili del motore.

Az-Delivery

Basta collegarsi con un nodo del multimetro a qualsiasi filo del motore e collegare il secondo nodo a qualsiasi altro filo del motore. Se si sente un segnale acustico, quei due fili fanno parte di una bobina e gli altri due fili fanno parte della seconda bobina.

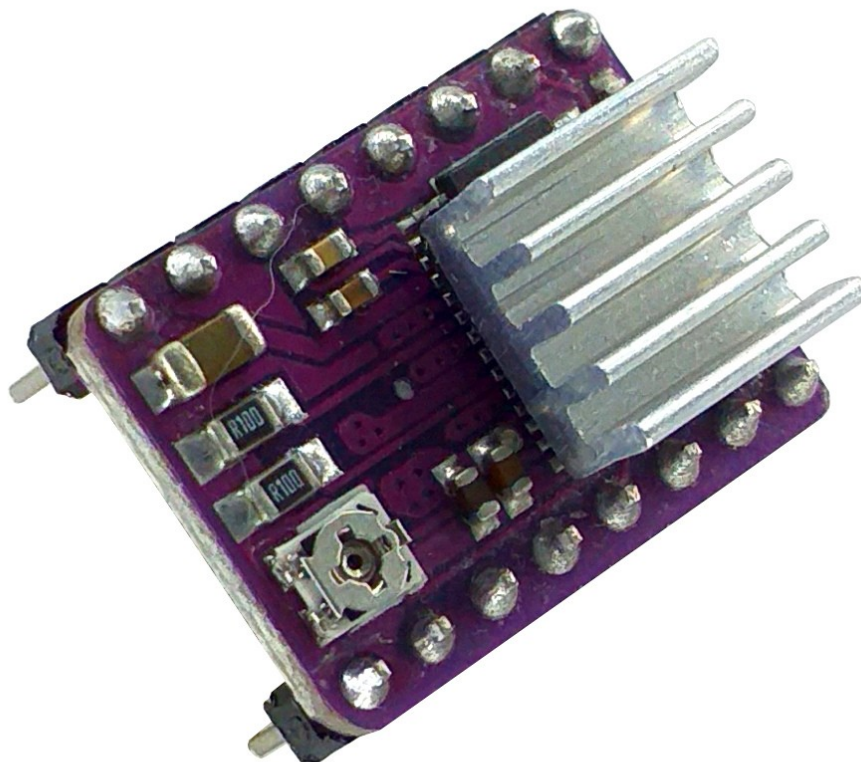
I motori passo-passo in generale hanno nomi simili per i loro fili. Ma i nomi dei fili sono diversi per i diversi produttori. Ecco alcuni esempi di nomi di fili diversi e come questi fili sono collegati con il driver A4988:

Pin motore passo-passo					>	Pin driver
A'	A	1	A1	A+	>	A1
A''	C	3	A2	A-	>	B1
B'	B	2	B1	B+	>	A2
B''	D	4	B2	B-	>	B2

(Per questo driver, i pin A1 e A2 sono per la prima bobina e i pin B1 e B2 sono per la seconda bobina)

Sistema di raffreddamento - Dissipatore di calore in alluminio

L'eccessiva dissipazione di potenza del chip del driver provoca un aumento della temperatura che può andare oltre la capacità del chip, probabilmente danneggiandolo. Anche se l'IC del driver ha una corrente nominale massima di 2,2 A per bobina, il chip può fornire solo correnti intorno a 1,5A per bobina senza surriscaldarsi. Per raggiungere più di 1,5A per bobina, è necessario un dissipatore di calore o un altro metodo di raffreddamento. Il modulo driver DRV8825 viene fornito con un dissipatore di calore in alluminio. Si consiglia di installarlo prima di utilizzare il modulo.





Potenziometro limite di corrente

Prima di utilizzare il motore, è necessario apportare una piccola modifica. Dobbiamo limitare la quantità massima di corrente che fluisce attraverso le bobine del motore passo-passo e impedire che superi la corrente nominale del motore. Sul driver A4988 è presente un piccolo potenziometro per trimmer che può essere utilizzato per impostare il limite di corrente. Per impostare il limite corrente, è necessario seguire i passaggi successivi:

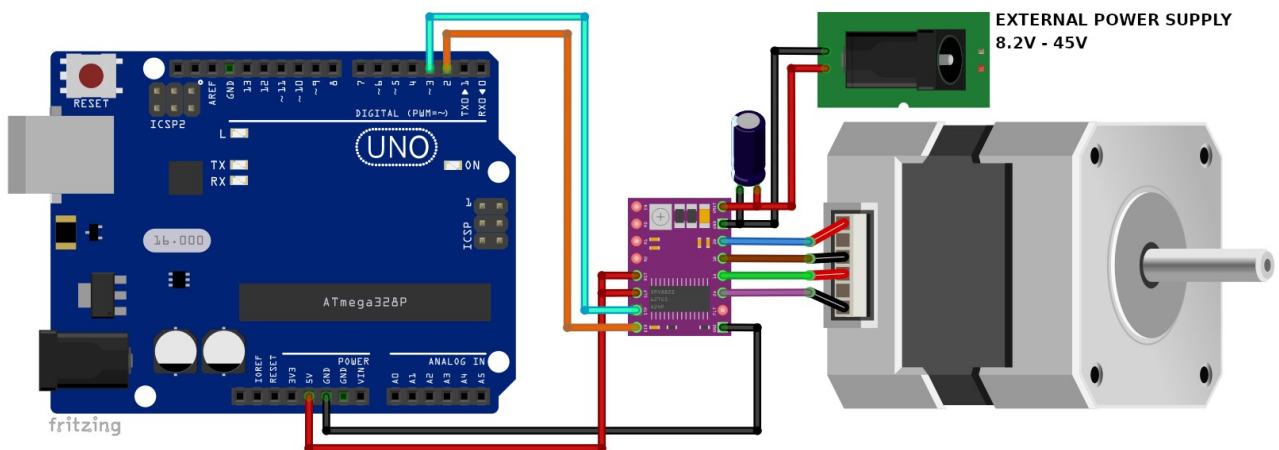
- » dai un'occhiata alla scheda tecnica del tuo motore passo-passo. Annotare la corrente nominale del motore. Nel nostro caso stiamo usando *"NEMA17" 200 steps/rev, 12V, 350mA*
- » mettere il driver in modalità full step lasciando scollegati i tre pin di selezione microstep
- » mantenere il motore in posizione fissa NON sincronizzando l'ingresso *STEP*. Non lasciare fluttuante l'ingresso *STEP*, collegarlo all'alimentazione logica (5V per Microcontrollore o 3,3V per Raspberry Pi)
- » posizionare l'amperometro in serie con una delle bobine sul motore passo-passo e misurare la corrente effettiva
- » prendere un piccolo cacciavite e regolare il potenziometro fino a raggiungere il valore di corrente nominale dal foglio dati del motore

NOTA: sarà necessario eseguire nuovamente questa regolazione se si cambia la tensione logica (*VDD*).

ATTENZIONE: Il collegamento o lo scollegamento di un motore passo-passo mentre il driver è alimentato può distruggere il driver!!!

Collegamento del driver con Microcontrollore

Collega il driver al tuo Microcontrollore come nello schema di collegamento in basso:



Pin driver > Pin Microcontrollore

RESET > 5V

Filo rosso

SLEEP > 5V

Filo rosso

GND LOGIC > GND

Filo nero

STEP > D3

Filo ciano

DIR > D2

Filo arancio

VMOT > + dell'alimentazione esterna

Filo rosso

GND MOT > GND dell'alimentazione esterna

Filo nero

Az-Delivery

RICORDA di mettere un grande condensatore elettrolitico di disaccoppiamento da 100 μ F sui pin di alimentazione del motore, il più vicino possibile alla scheda, come nello schema di collegamento sopra.

Mantenere i pin di selezione del microstep scollegati per far funzionare il motore in modalità full step o collegare il pin *MS* appropriato alla tensione *VCC*, per operare in diverse modalità di eccitazione, come abbiamo discusso. Ma se lo fai, l'albero motore per lo stesso clock *STEP* funzionerebbe più lentamente se stai usando una modalità di eccitazione diversa dalla modalità a passo completo. Questo perché in altre modalità di eccitazione stiamo usando più passi, più passi equivalgono a più tempo per superarli. Quindi, per utilizzare un motore con la stessa velocità di rotazione, è necessario aumentare il clock sul pin *STEP*. Con il motore passo-passo "*NEMA17*", abbiamo usato *950us* per lo stato ON e *950us* per lo stato OFF per il segnale di clock sul pin *STEP*. Questo è il minimo nella modalità a passo completo (tutti i pin *MS* sono scollegati). Ma con la modalità 1/32 di passo (tutti e tre i pin *MS* collegati al *VCC*), questi valori arrivano fino a 18us. Lo vedrai nel codice

Non abbiamo bisogno di librerie per far funzionare questo driver. Creeremo il nostro sketch.

Az-Delivery

Codice Arduino-IDE:

```
uint8_t stepPin = 2;
uint8_t dirPin = 3;
int steps = 1000;    // you should increase this if you are using
                     // some of microstepping modes
int usDelay = 950;   // minimal is 950 for full step mode and NEMA15 motor
                     // minimal is 35 for sixteenth step mode

void setup() {
    Serial.begin(9600);
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
}

void loop() {
    digitalWrite(dirPin, HIGH); // motor direction cw
    for(int x = 0; x < steps; x++) {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(usDelay);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(usDelay);
    }
    delay(1000);
    digitalWrite(dirPin, LOW); // motor direction ccw
    for(int x = 0; x < steps; x++) {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(usDelay);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(usDelay);
    }
    delay(1000);
}
```

Az-Delivery

Iniziamo lo sketch definendo i pin *STEP* e *DIR* collegati al nostro Microcontrollore. Definiamo una variabile chiamata *step* che usiamo per il numero di passi per l'albero motore. Nella funzione di configurazione, dichiariamo i pin *STEP* e *DIR* come uscite digitali. Nella sezione loop giriamo il motore in senso orario e poi lo giriamo in senso antiorario ad un intervallo di due secondi

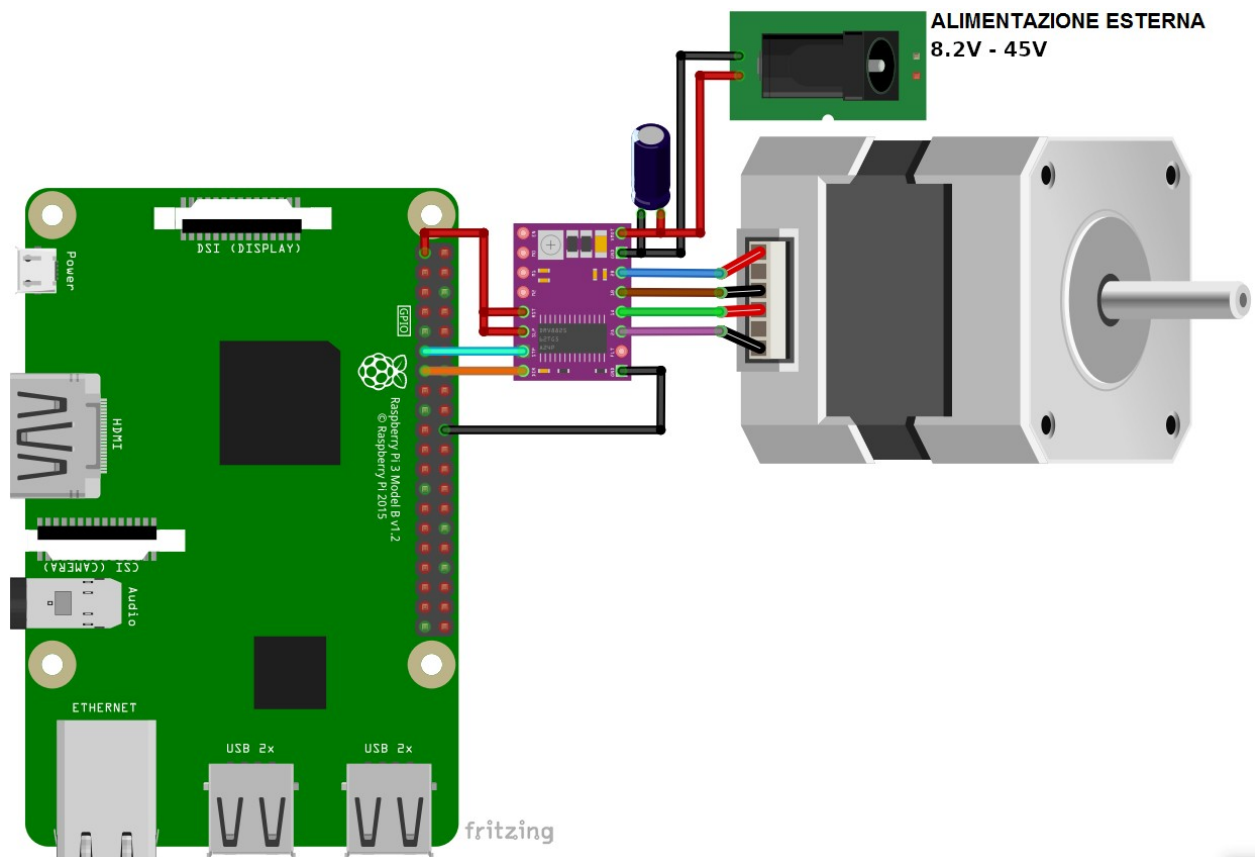
Per controllare la direzione di rotazione di un motore, impostiamo il pin *DIR* su *HIGH* o *LOW*. L'ingresso *HIGH* gira il motore in senso orario e *LOW* lo gira in senso antiorario.

La velocità di rotazione di un albero motore è determinata dalla frequenza degli impulsi (segnale di clock) che inviamo al pin *STEP*. Più alti sono gli impulsi, più veloce è il motore. Gli impulsi non fanno altro che tirare l'uscita *HIGH*, aspettando un po' poi tirandola *LOW* e aspettando di nuovo. Questo periodo di attesa è definito dalla variabile *usDelay*. Modificando il ritardo tra i due impulsi, si modifica la frequenza di tali impulsi e quindi la velocità di rotazione dell'albero motore.

Abbiamo testato questo codice con il motore passo-passo *NEMA17*. Nella modalità passo-passo il valore minimo della variabile *usDelay* è 950. Se si scende al di sotto di questo valore, l'albero del motore non si sposterà. Ciò significa che il motore non è in grado di gestire frequenze di clock più elevate. Se si imposta la modalità di eccitazione 1/32, si ottengono trenta-due volte più passi rispetto alla modalità full step. Quindi, al fine di spostare l'albero del motore nella stessa posizione come nella modalità full step, dobbiamo aumentare i passi variabili trenta-due volte. E per ottenere la stessa velocità di rotazione della modalità full step (*usDelay* = 950) dobbiamo cambiare il valore di *usDelay* su 18 (che è minimo per questa modalità). Se si imposta questa variabile su un valore inferiore a 18, l'albero del motore non si sposterà

Coolegamento del driver con Raspberry Pi

Collega il driver al tuo Raspberry Pi come nello schema di collegamento seguente:



Pin driver > Pin Raspberry Pi

VDD > 3.3V [pin 17]

GND > GND [pin 30]

STEP > GPIO17 [pin 11]

DIR > GPIO27 [pin 15]

Filo rosso

Filo nero

Filo ciano

Filo arancio

VMOT > + dell'alimentazione esterna

GND > GND dell'alimentazione esterna

Filo rosso

Filo nero

Az-Delivery

Collega il pin *RST* e *SLEEP* alla 3.3V (**Filo rosso**) per tenere abilitato il driver.

RICORDA di posizionare un grande condensatore elettrolitico di disaccoppiamento da $100\ \mu F$ il più vicino possibile alla scheda, come nello schema di collegamento sopra.

Come per Arduino, non è necessario installare alcuna libreria per questo modulo driver. Creeremo solo il nostro script Python. Ecco il codice:

```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
STEP = 17 # step pin
DIR = 27 # direction pin
EN = 23 # enable pin
GPIO.setup(STEP, GPIO.OUT)
GPIO.setup(DIR, GPIO.OUT)
GPIO.setup(EN, GPIO.OUT)
steps = 5000 # number of steps
usDelay = 950 # number of microseconds
uS = 0.000001 # one microsecond
GPIO.output(EN, GPIO.LOW)
print("[press ctrl+c to end the script]")
try: # Main program loop
    while True:
        GPIO.output(DIR, GPIO.HIGH) # cw direction
        for i in range(steps):
            GPIO.output(STEP, GPIO.HIGH)
            sleep(uS * usDelay)
            GPIO.output(STEP, GPIO.LOW)
            sleep(uS * usDelay)
        sleep(2)
        GPIO.output(DIR, GPIO.LOW) # ccw direction
        for i in range(steps):
            GPIO.output(STEP, GPIO.HIGH)
            sleep(uS * usDelay)
```

Az-Delivery

```
GPIO.output(STEP, GPIO.LOW)
sleep(uS * usDelay)
sleep(2)
# Scavenging work after the end of the program
except KeyboardInterrupt:
    GPIO.output(EN, GPIO.HIGH)
```

Abbiamo appena trasformato lo sketch di Arduino in codice Python. L'unica differenza è il codice per il pin EN. Questo è il pin per abilitare il driver. Se è tirato *LOW*, il driver è abilitato e se è tirato *HIGH* il driver è disabilitato. Ne abbiamo bisogno perché, i pin *DIR* e *STEP* fluttuano e se terminiamo lo script senza collegare questi pin a *GND* o *VCC*, il driver impazzirebbe e l'albero del motore inizierebbe a muoversi in modo strano. Quindi dobbiamo disabilitare il driver al termine dello script. Lo facciamo nel blocco *except* del codice.

Il resto del codice è lo stesso dello sketch di Arduino, quindi non è necessario spiegarlo.

Ce l'hai fatta, ora puoi usare il tuo modulo per i tuoi progetti.



E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet.

Se stai cercando dei prodotti di alta qualità per il tuo Arduino e Raspberry Pi, AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.

<https://az-delivery.de>

Buon divertimento!

Impressum

<https://az-delivery.de/pages/about-us>