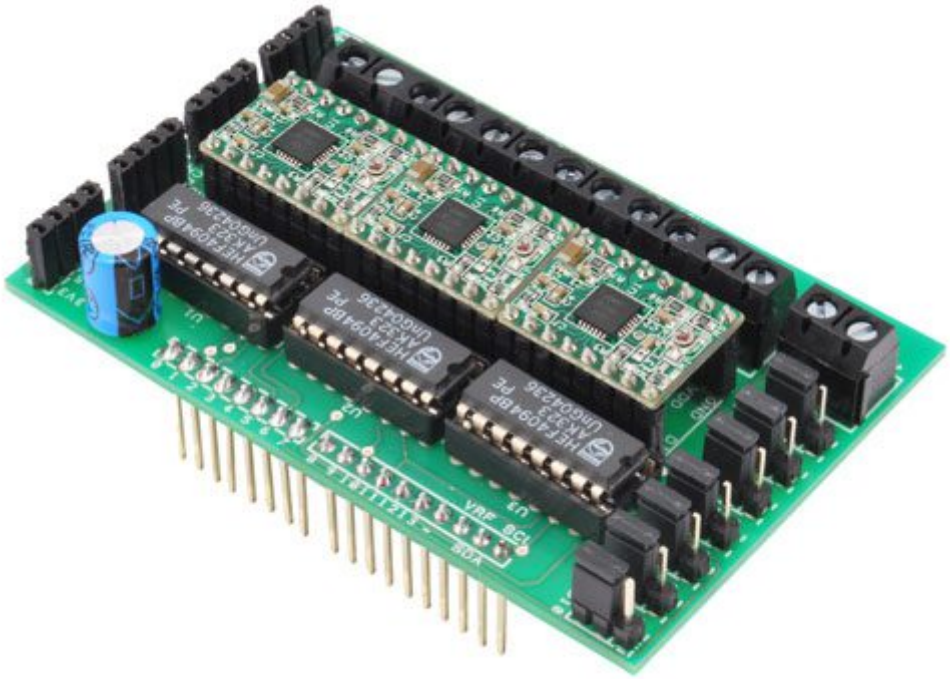
Shield per Arduino motori stepper - in kit da saldare

Prezzo: 43.44 €

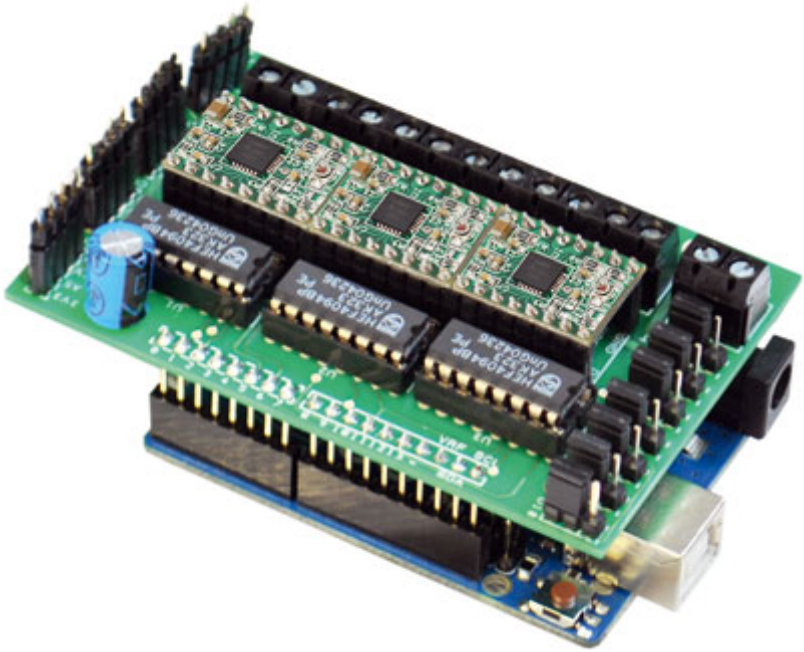
Tasse: 9.56 €

Prezzo totale (con tasse): 53.00 €

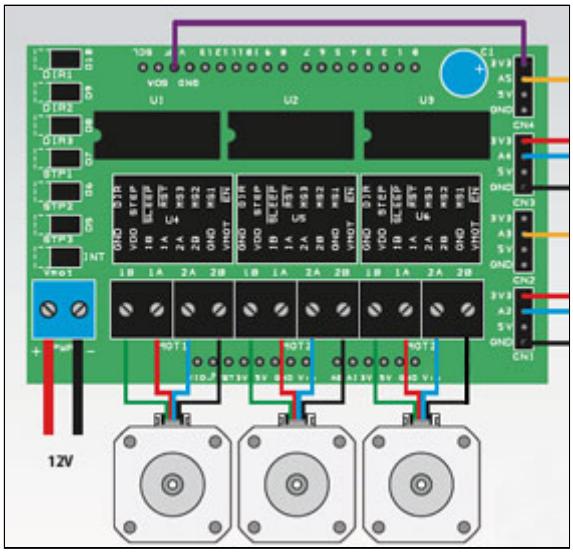


Shield per Arduino UNO o Mega, che permette di comandare tre motori passo-passo bipolari da 2A. Il controllo dei motori è gestito da un driver (MD09B oppure 3DDRIVER - uno per motore) prodotto dalla Pololu. Ogni driver contiene un doppio ponte ad H a MOSFET e può essere impostato per gestire sia la direzione, sia il numero di gradi che il rotore del motore deve compiere alla ricezione di ogni comando; in altre parole, possiamo decidere se quando lo comandiamo, il modulo deve far ruotare l'albero di uno step alla volta, oppure di 1/2, 1/4, 1/8 o 1/16, in base all'accuratezza che si desidera ottenere. Sulla shield sono presenti anche 4 ingressi analogici. Alimentazione: 12 Vdc, dimensioni: 86x56 mm. **N.B.** la confezione comprende la scheda da saldare e i 3 driver (già montati).

MONTATA SU ARDUINO...



COLLEGAMENTI DEI MOTORI



IL DRIVER PER I MOTORI MD09B

Il controllo dei motori passo-passo è operato da moduli Pololu (www. pololu.com) basati sull'integrato A4983 della Allegro e dotati ciascuno di un doppio driver a ponte ad H governato da una logica che controlla l'accensione e permette di far compiere al motore passi ridotti rispetto a quelli previsti dalla sua parte elettromeccanica; ad esempio, per ogni impulso di comando inviato a STEP (il contatto che riceve gli impulsi di azionamento del motore) lo stepper-motor può compiere un passo intero oppure frazioni da 1/2 a 1/6, a seconda della combinazione logica presentata ai piedini MS1, MS2, MS3, secondo la tabella riportata in questo riquadro. La direzione del movimento conseguente ad ogni impulso ricevuto sul contatto STEP dipende dal livello logico applicato all'ingresso DIR (1logico forza il verso antiorario e 0 quello orario). Ogni ponte può erogare una corrente di 2 A, mentre il modulo si alimenta in continua con tensioni fino a 35 V; un piccolo trimmer posto sulla basetta e connesso al pin REF (17) dell'A4983, permette di regolare la corrente erogabile dai ponti.

FUNZIONAMENTO

Il comando del movimento può essere ottenuto in due modi: nel primo, Arduino fornisce direttamente gli impulsi per impostare il verso di rotazione dell'albero dello stepper-motor e per far ruotare di uno step ad impulso; nel secondo, ossia quello da noi impiegato e previsto nel firmware, Arduino imposta il verso di rotazione e invia impulsi che determinano rotazioni di uno step o frazione di esso, a seconda dell'impostazione fatta all'inizializzazione e conservata da opportuni shift-register. La differenza tra le due modalità è che la prima è più impegnativa della seconda e costringe Arduino a lavorare più velocemente perché deve aggiornare più in fretta lo stato dei motori passo passo. Infatti nel primo modo di comando, che si ottiene spostando i jumper DIR ed STP (quindi DIR1/STP1, DIR2/STP2, DIR3/ STP3) verso le linee di Arduino, è proprio Arduino che imposta la posizione dell'albero del motore passo per passo; ipotizzando che il motore sia collegato all'uscita MOT1, ciò significa che se il motore è un 360 passi ed il suo albero deve ruotare di 180° in due secondi, occorre inviare 90 impulsi al secondo al piedino D7 (se la velocità angolare deve aumentare, ad esempio per far rispondere il labirinto più velocemente, la frequenza degli impulsi di step deve crescere di conseguenza). Prima, però, Arduino deve impostare la direzione con l'opportuno livello logico sul piedino D10, che decide la direzione mediante il jumper DIR. Nella seconda modalità, ci avvaliamo delle prerogative del controller Pololu per impostare preventivamente il numero di step o frazioni da compiere ogni volta che Arduino invia un impulso di rotazione e imposta il verso; questa modalità si ottiene spostando i jumper DIR e STP verso i pin Q5 e Q6 dei rispettivi shift register. In questo caso, ci basta definire il fattore di divisione una sola volta e poi per variare la velocità di risposta si gioca sulla frequenza degli impulsi giunti da Arduino. Analizziamo dunque questa modalità, per comprendere la quale dobbiamo spendere qualche altra parola sul modulo controller per stepper-motor prodotto dalla Pololu, ovvero sulla sua interfaccia di comando. Il modulo, siglato MD09B, consta di un doppio ponte ad H governato da un'elettronica che permette di impostare il verso di rotazione del campo elettromagnetico e quindi dell'albero dello stepper motor, oltre a frazionare gli step. Ogni volta che sul piedino STEP giunge un impulso (la durata minima ammissibile è 1 ?s) a livello alto, a meno di diversa impostazione le uscite 1A, 1B, 2A e 2B forniscono gli impulsi per comandare lo spostamento del rotore del motore di uno step. Ciò vale se gli ingressi MS1, MS2, MS3 sono tutti a livello basso; in caso contrario, si ottiene la rotazione di 1/2, 1/4 ecc. Sempre, perché il controller funzioni bisogna che il contatto EN sia a livello basso; ponendolo a 1 il modulo va in standby e i motori non ricevono corrente. Il contatto DIR decide il verso di rotazione: posto a livello logico basso impone il verso orario, mentre nello stato alto imposta l'antiorario. SLEEP attiva la modalità di riposo (logica attiva e driver spenti) mentre RESET azzera il controllore che governa i driver e pone le uscite di comando dei motori (1A, 1B, 2A, 2B) a zero logico anche se STEP continua a ricevere impulsi. La logica è TTL-compatibile, ma comunque il modulo accetta agli ingressi di controllo tensioni fino a 0,7 volte quella applicata tra il contatto 2 (Vcc) e massa (GND); il livello basso equivale a non più di 0,3xVcc. Detto ciò, vediamo che per impostare il fattore di divisione degli step, Arduino deve definire la combinazione delle linee MS1, MS2, MS3 di ogni modulo; per farlo distintamente e liberarsi del problema, nello shield sono previsti tre shift-register che caricano l'impostazione di ciascun controller individualmente e sequenzialmente. Una volta configurati i controller, le impostazioni possono rimanere quelle fin quando non serve modificarle e comunque non bisogna aggiornarle ad ogni impulso di comando di STEP; è questo che sgrava Arduino da parte del lavoro. L'aggiornamento delle impostazioni dei controller si effettua caricando sequenzialmente i dati tramite la linea D13 di Arduino, che finisce in parallelo sui pin 2 (DATA) dei registri a scorrimento; ogni volta che è stata inviata la configurazione di tutti e tre i controller, Arduino attiva la linea di STROBE (piedini 1 di U1, U2, U3, ovvero I/O D2 della Arduino) e trasferisce sugli MD09B i rispettivi stati, quindi riporta D2 a zero logico. Insieme alla configurazione degli MS1, MS2, MS3 (che può essere diversa da U1 a U3) i dati inviati da Arduino riguardano lo stato di SLEEP e RESET, che sono impostati uguali per tutti i moduli e valgono 1 logico fisso. Ogni controller alimenta direttamente il motore e viene a sua volta alimentato da una linea comune che, tramite il jumper siglato Vmot, può prendere tensione da Arduino o da un alimentatore esterno mediante l'apposita morsettiera PWR.

Documentazione e link utili

- Sketch di esempio