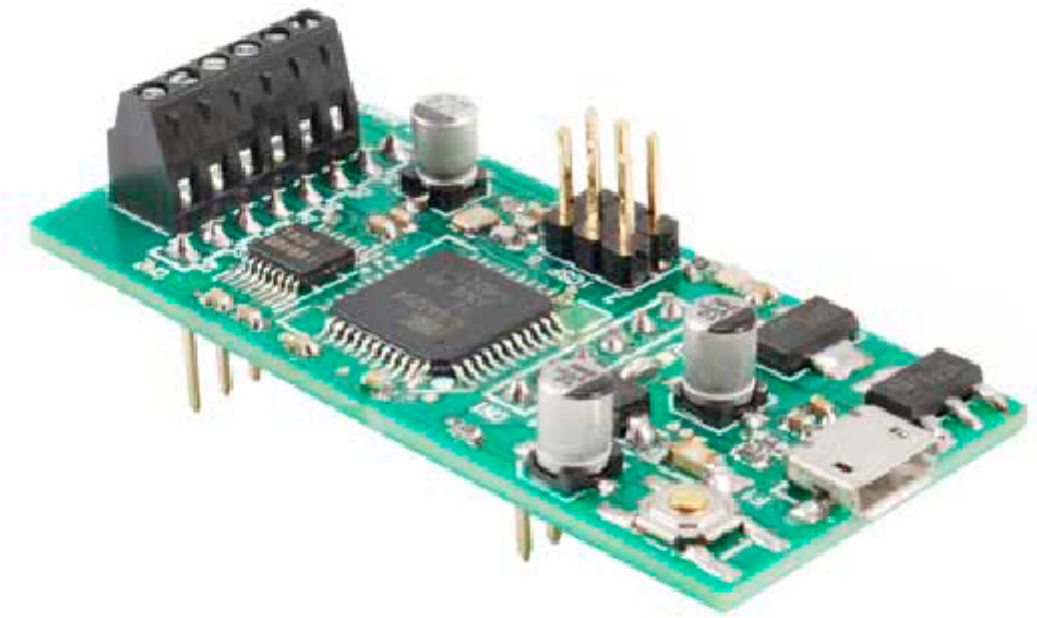
Open Motor Control - montato

Prezzo: 27.87 €

Tasse: 6.13 €

Prezzo totale (con tasse): 34.00 €



Basato sul microcontrollore ATmega32U4, questo modulo è in grado di pilotare 2 motori DC o 1 motore passo-passo. L'alimentazione per i motori DC può essere compresa tra 3 e 15 Vdc con una corrente di uscita di 1,4 A (2,5 A di picco) per motore. Dispone di protezione contro sovracorrenti e sovratemperatura. Il driver può essere comandato via USB, seriale o I2C. Il firmware di gestione è stato realizzato con l'ambiente di sviluppo di Arduino (basato su Arduino Leonardo), dando così la possibilità all'utente di effettuare le personalizzazioni desiderate.

Caratteristiche tecniche

- Può controllare due motori DC o uno stepper motor
- Alimentazione motori: da 3 a 15 Volt
- Alimentazione logica di controllo: da 3 a 5 Volt (selezionabile interna o esterna)
- Corrente di uscita: 1,4 A continua (2,5 A di picco) per motore
- Tre possibili modalità d'interfaccia: USB, Seriale e I2C
- Ingressi compatibili 3,3V e 5V

Gestione via Seriale

Se vogliamo gestire il controller tramite una scheda Arduino conviene utilizzare una seriale software, utilizzando una sola linea digitale da connettere alla linea RX, in questo modo la porta seriale hardware rimarrà libera di comunicare con il PC. Lo sketch di esempio è riportato nel Listato 4. In questo esempio viene mostrato come sia possibile costruire il byte di controllo (byte2) partendo dalle impostazioni dei singoli bit di direzione e frenatura. Sarà la riga di codice int mode = (modeM2<<3) + (modeM1<<2) + (dirM2<<1) + dirM1 a ricostruire il byte completo traslando e sommando i singoli bit. Utilizzeremo successivamente quattro comandi Serial.write per inviare i singoli byte contenenti il carattere di inizio e i tre byte di dati.

Gestione tramite I²C

Se volete gestire la scheda OMC21 (Open Motor Control) tramite la porta TWI di Arduino non dovete far altro che connettere le due linee SDA e SCL di entrambe le porte tra loro e connettere i GND di entrambe le schede. Non sono necessarie le resistenze di pull-UP per il semplice motivo che sono già interne sia su Arduino UNO che su 32U4 ed abilitate in automatico non appena si richiama la funzione Wire.begin(). Per lo sketch da usare su Arduino UNO per comandare il controller fare riferimento al Listato 5, che si presenta molto simile ai listati precedenti. C'è però una importante differenza con il modo di comunicare tramite seriale che riguarda il sistema di indirizzamento delle periferiche. Mentre in una comunicazione seriale lo scambio di dati avviene punto a punto, ovvero solo due periferiche dialogano tra loro, nel BUS I2C vi possono coesistere sino a 127 periferiche di tipo slave tutte in comunicazione con l'unico master. Il master in questo caso è Arduino UNO (o la logica di comando) mentre gli slave sono le schede Open Motor Control; il vantaggio evidente è quello di poter utilizzare più controller e quindi più motori tutti gestiti da un'unica scheda Arduino utilizzando una sola porta TWI. Ricordiamo che in Arduino la porta TWI utilizza a tutti gli effetti il protocollo I2C che però è proprietario della Philips, che lo ha inventato. Nel Listato 5 è infatti presente la riga di codice const int OMC21address=4 che ha lo scopo di specificare verso quale slave avviene la comunicazione; ovviamente sul BUS TWI potranno coessistere anche altri slave non necessariamente controller motori OMC21. Nello sketch OMC21.ino è stato specificato come indirizzo di periferica di default il valore 4 ma a piacimento questo può essere modificato se intendente usare sul BUS una perifierica che ha già questo indirizzo. Se usate più controller OMC21 sarà necessario che ciascuno abbia un indirizzo univoco, dovrete quindi caricare su ciascun controller uno sketch in cui avrete specificato un indirizzo differente. Nella stesura del programma di controllo dovrete poi tenere conto dei vari indirizzi assegnati, per assicurare il corretto transito dei dati.

Documentazione e link utili

- **Firmware e software per OMC21**