

# Shield Arduino I/O expander

Prezzo: 27.87 €

Tasse: 6.13 €

Prezzo totale (con tasse): 34.00 €



Shield di espansione I/O per Arduino basato sul chip MCP23017 prodotto da Microchip. Dispone di 8 ingressi digitali, ognuno dei quali dispone di un diodo di protezione, una resistenza di pull-up ed un LED; 8 uscite a relè con relativo LED di stato. La comunicazione con l'MCP23017 è basata sull'I<sup>2</sup>C-Bus, pertanto la libreria utilizza "Wire.h" di Arduino. Il bus I<sup>2</sup>C è realizzato da 2 pin (SDA, ossia dati, ed SCL, cioè clock); sull'integrato sono disponibili sui pin 13 e 12, portati sui corrispondenti pin della scheda Arduino. Nello shield è prevista la possibilità di portare (tramite ponticello) il pin di interrupt dell'MCP23017 (INTB, pin 19) in ingresso su due differenti pin di Arduino (D2 o D4) in modo da adattarlo il più possibile al reale utilizzo. L'indirizzo I<sup>2</sup>C-Bus dell'MCP23017 viene selezionato tramite dip switch che permette di collegare fino a 8 diversi shield senza che interferiscano tra di loro. Il kit comprende tutti i componenti, la basetta forata e serigrafata e tutte le minuterie.



### Libreria Arduino I<sup>2</sup>C I/O Expander Shield

Abbiamo sviluppato una libreria software Arduino che fornisce tutte le routine necessarie alla rilevazione degli shield collegati e alla relativa gestione degli I/O (potete scaricare tale libreria da questa pagina). La comunicazione con l'MCP23017 è basata sull'I<sup>2</sup>C-Bus, pertanto la libreria utilizza "Wire.h" di Arduino; è presente una funzione "begin(int i2cAddress)" per inizializzare il singolo shield identificato tramite indirizzo I<sup>2</sup>C, ed anche una "init()" per programmare correttamente i registri interni del chip secondo le nostre necessità, ed infine una "pinMode(int pin, int mode)" per indicare se i singoli pin di I/O sono input o output. Oltre alle funzioni di inizializzazione, le istruzioni che in particolare ci interessano sono le "digitalRead(int pin)", "digitalWrite(int pin, int val)", "digitalWordRead()" e "digitalWordWrite(word w)" che, rispettivamente, permettono di leggere lo stato di un singolo pin di input, scrivere un singolo pin di output, leggere lo stato di tutti i pin di input (8 nel nostro caso) ed infine scrivere lo stato di tutti i pin di output (sempre 8). La gestione degli input può avvenire tramite interrupt; è pertanto presente un'apposita funzione "pinDisableINT(int pin)" che permette di configurare il singolo pin come generatore di interruzioni o meno (in particolari applicazioni potrebbe essere utile avere la possibilità di non impostare tutti i pin di input come generatori di interrupt, ma farlo selettivamente).

### Documentazione e link utili

- [Libreria IOExpanderMCP23017](#)