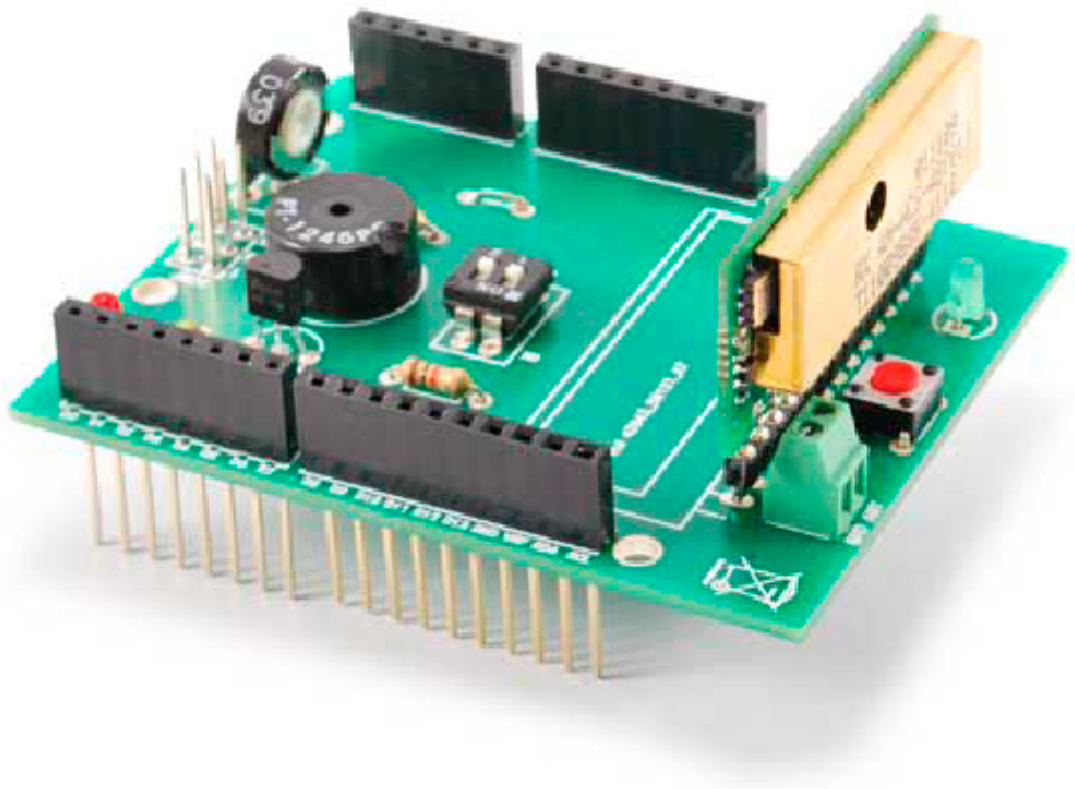
Shield HCS per Arduino - in kit da saldare

Prezzo: 22.95 €

Tasse: 5.05 €

Prezzo totale (con tasse): 28.00 €



Basato sul modulo RX-4MHCS-4B dell'Aurel, un ibrido supereterodina contenente un ricevitore a 433,92 MHz completo di decodifica HCS, l'RF Shield consente di realizzare un ricevitore a 8 canali basato su Arduino, estendibile a 12 e controllabile tramite apposito radiocomando siglato TX-12CH. Lo sketch caricato su Arduino permette di leggere, elaborare i dati in arrivo e usarli per generare i comandi I²C-Bus con i quali gestisce lo shield a relé 7100-FT1079K (non compreso). Lo shield dispone di un buzzer, un trimmer (alimentato dagli stessi 5 volt che alimentano l'ibrido), di cui Arduino legge il cursore tramite la linea analogica A2 (allo scopo di impostare il tempo di ritardo in rilascio nella modalità monostabile ritardata) e un LED (LD1) che viene acceso dalla linea D2 di Arduino quando viene ricevuto un codice riconosciuto valido, un dip switch a 2 poli con il quale si impostano le modalità di funzionamento del ricevitore (bistabile o astabile). Il modulo RX-4MHCS-4B viene alimentato tramite il 5V di Arduino ed è contornato dal pulsante P1 e dal LED LD2; i contatti d'antenna finiscono su una morsettiera miniatura che permette di connettere l'antenna ricevente, sia essa uno stilo lungo 17 cm (in questo caso si collega al solo morsetto ANT) o un'antenna esterna (nel qual caso alla morsettiera si collega il cavo schermato, con lo schermo a GND e il polo caldo ad ANT). **N.B.** La scheda Arduino Uno e lo Shield Arduino I/O expander non sono compresi.

Librerie Arduino

Per gestire lo shield I/O Expander è prevista una libreria Arduino che fornisce tutte le routine necessarie alla rilevazione degli shield collegati e alla relativa gestione degli I/O (potete scaricare tale libreria dal nostro sito www.elettronicain.it). La comunicazione con l'MCP23017 è basata sull'I²C-Bus, pertanto la libreria utilizza "Wire.h" di Arduino; è presente una funzione "begin(int i2cAddress)" per inizializzare il singolo shield identificato tramite indirizzo I²C, 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). Quanto allo shield HCS, non richiede librerie ma viene gestito direttamente dallo sketch.

Documentazione e link utili

- Sketch