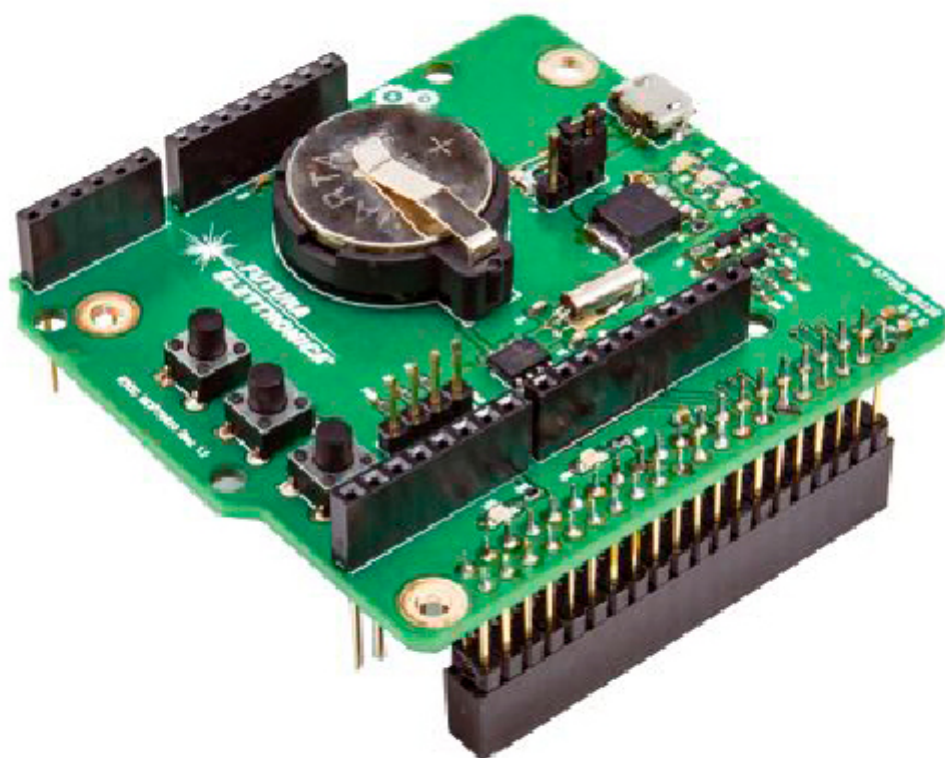


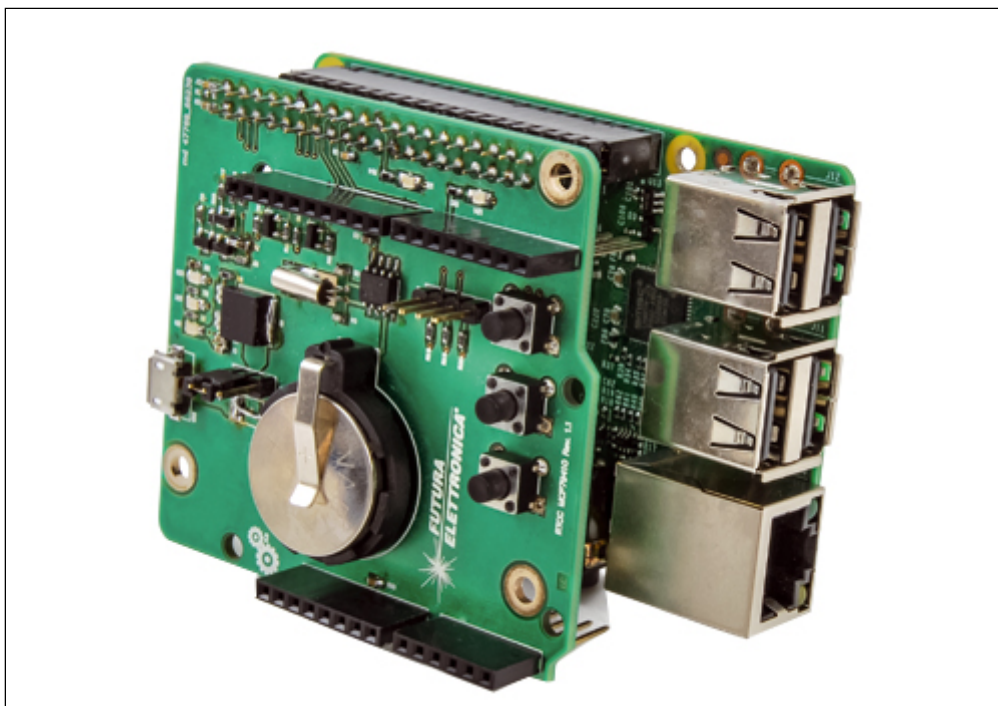
Shield RTC per Raspberry e Arduino

Prezzo: 13.93 €

Tasse: 3.07 €

Prezzo totale (con tasse): 17.00 €





Nuovo shield Real Time Clock applicabile sia a una scheda Arduino Uno R3 che a una Raspberry Pi 2 / 3B / 3B+. Particolarmente indicata in tutte quelle applicazioni che richiedono l'informazione oraria, che viene ottenuta localmente mediante circuiti chiamati RTC (Real Time Clock) o RTCC (Real Time Clock Calendar); tra esse le radiosvegliie, i sistemi di registrazione degli accessi del personale, quelli di accensione/spegnimento a tempo ecc. L'RTCC in questo shield è basato sull'integrato MCP79410, che mette a disposizione: configurazione di ore, minuti e secondi sia nel formato 24h che nel 12h (AM/PM); configurazione di giorno, mese, anno e giorno della settimana; gestione automatica degli anni bisestili; oscillatore a 32.768 Hz; calibrazione/regolazione interna digitale con risoluzione di ± 1 ppm (range massimo ± 129 ppm); due allarmi programmabili; TimeStamp sia su power-up che su power-down; 64 Byte di memoria SRAM tamponata; 128 Byte di memoria EEPROM con possibilità di scrittura paginata a 8 byte alla volta; 8 Byte di memoria EEPROM protetta, per scrivere nella quale (si può un solo byte per volta) bisogna eseguire una sequenza di sblocco; interfaccia di comunicazione I2C fino a 400 kHz. **La scheda può anche essere impostata per accendere Arduino o Raspberry Pi ad uno specifico orario. N.B.** Arduino e Raspberry Pi non sono compresi. [Clicca qui per leggere l'articolo completo presentato su Elettronica In n. 204 - Aprile 2016 e n. 205 Maggio 2016](#)

Libreria per MCP79410

La libreria permette di configurare e gestire l'integrato MCP79410 ed è divisa in tre file; il primo contiene le funzioni sviluppate per l'occasione e ha estensione .cpp (lo abbiamo chiamato MCP79410.cpp). Il secondo ha estensione .h e contiene le dichiarazioni di funzione del precedente più tutte le variabili e le strutture dati necessarie (MCP79410.h). Il terzo (keywords.txt) è un file di testo contenente le parole chiave delle funzioni pubbliche da utilizzare negli sketch Arduino. Tutti questi devono essere raggruppati sotto una cartella comune, all'interno della cartella di installazione dell'IDE Arduino, nominata MCP79410. Così facendo, comparirà la libreria nello IDE Arduino sotto la voce di menu Sketch>Include Library. Quindi supponendo che lo IDE Arduino sia stato installato sotto C:\Program Files (x86)\Arduino la nostra libreria dovrà essere salvata nel seguente percorso: C:\Program Files (x86)\Arduino\libraries\MCP79410. Oltre ai file di libreria è consuetudine aggiungere una cartella con degli sketch di esempio: nel nostro caso abbiamo creato la sotto-cartella Examples ? MCP79410_SAdvancedSettings nella quale trovano posto i file del nostro sketch che ci permettono di configurare i registri di MCP79410.

Sketch

Lo sketch è suddiviso in sei file di cui il principale è "MCP79410_AdvancedSettings.ino", quindi con un doppio clic su questo si apre l' IDE Arduino e tutti i file ad esso associati. I file dello sketch sono:

- MCP79410_AdvancedSettings ? File principale contenente le funzioni "setup()" e "loop()" tipiche di uno sketch Arduino, nonché le dichiarazioni di variabile, le costanti stringa, le costanti di tempo, le dichiarazioni delle macchine a stati ecc;

- MCP79410_AdvancedSettings > File principale contenente le funzioni "setup()" e "loop()" tipiche di uno sketch Arduino, nonché le dichiarazioni di variabile, le costanti stringa, le costanti di tempo, le dichiarazioni delle macchine a stati ecc;
- DigitalInput > File di gestione degli ingressi digitali, ovvero i pulsanti P1, P2 e P3;
- DigitalOutput > File di gestione delle uscite digitali (il LED D4 e l'uscita "ForceON");
- RTCC_Management > File di gestione dell'integrato MCP79410 e relativa configurazione;
- RTCC_Settings > file per la programmazione dell'integrato MCP79410 TimeKeeper e allarmi;
- TimersInt > File di gestione dell'interrupt per la gestione dei timer.

Documentazione e link utili

- [Sketch](#)