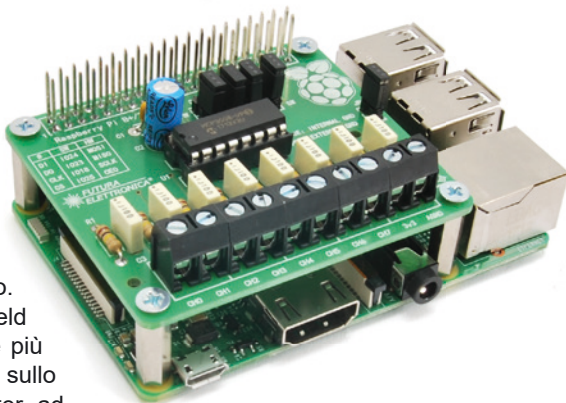


## Shield espansione ADC 8 canali per Raspberry Pi (cod. FT1306K)

### Il circuito

Modulo shield di espansione per Raspberry Pi, basato sul convertitore analogico-digitale a 10 bit MCP3008, che consente di collegare un massimo di 8 canali d'ingresso, ognuno dei quali corrispondente a un suo piedino. Sullo shield è possibile impilare shield di altro tipo (non si possono usare più shield di espansione analogica sullo stesso Raspberry), in modo da poter, ad esempio, interfacciare una gran quantità di sensori in applicazioni di misura, diagnostica, controllo ambientale e IoT. Risoluzione in tensione: 3,22 mV, range della tensione campionabile: 0÷3,3 V, frequenza di campionamento: 220 kpsps max. Si tratta di uno shield progettato per essere applicato al connettore di espansione della Raspberry Pi 3, che da un lato riporta il connettore femmina a passo 2,54x2,54 mm idoneo ad accoppiarsi a quello della Raspberry Pi e superiormente ha dei pin strip a doppia fila per ripetere le connessioni, in modo da poterle rendere accessibili ad altri circuiti ed eventuali ulteriori shield richiesti eventualmente dalle applicazioni.

Lo shield è stato specificatamente progettato per essere utilizzato con la Raspberry Pi versione 3, la quale è del tutto priva di porte ADC. Proprio per ovviare a questa mancanza, lo shield potrà essere montato sopra il connettore di espansione a 40 vie della Raspberry Pi 3, ottenendo



così 8 porte analogiche on-board, i cui segnali, opportunamente digitalizzati nel formato a 10 bit, saranno acquisiti sfruttando la porta SPI hardware della Raspberry Pi stessa, nel modo che vi spiegheremo più avanti.

Da notare che è stato previsto un ponticello siglato JP1, che permette di decidere se sfruttare come riferimento la massa interna al circuito, oppure la massa esterna. Se il ponticello verrà chiuso, il riferimento verrà dato dalla massa interna dello shield, pertanto all'ingresso sarà sufficiente collegare l'uscita del sensore e l'alimentazione per il sensore dovrà essere prelevata dal circuito stesso, sfruttando i morsetti "AGND" e "3,3V".

Qualora si disponga, invece, di sensori dotati di una propria massa perché provenienti da un circuito differente, si potrà lasciare aperto JP1, ma in questo caso la massa di questo secondo circuito dovrà essere connessa al morsetto AGND dello shield.

Lo stesso vale se bisogna acquisire tensioni provenienti da un circuito con una massa distinta da quella dello shield. Sono disponibili inoltre altri 4 Jumper (DI-DO-CLK-CS) in cui l'utente deve selezionare il tipo di SPI scegliendo tra HW o SW, avendo l'accortezza di spostare tutti i jumper su SW o su HW.

### Realizzazione pratica

Si montano dapprima le resistenze e lo zoccolo per l'integrato, quindi i condensatori (partendo dal non polarizzato e lasciando per ultimo l'elettrolitico), il jumper e il connettore doppio maschio/femmina a passo 2,54x2,54 mm, da montare lato saldature con le punte che sporgono dal lato componenti (le punte devono essere lunghe circa 15 mm). Le punte serviranno per applicare sopra il circuito eventuali ulteriori shield, mentre la parte femmina del connettore consentirà di innestare lo shield sulla Raspberry Pi 3.

Completare lo shield con le morsettiere; sui morsetti è stato riportato gli ingressi degli 8 canali (CH0÷CH8) oltre alla massa comune degli input (GND) e al positivo di alimentazione 3,3 volt, anche questo riferito al contatto GND.

Alla fine inserire l'integrato nel proprio zoccolo, badando che la sua tacca di riferimento sia rivolta come indicato, ossia

dal lato opposto a quello dove si trova il jumper JP1 (che tra l'altro dev'essere chiuso). Per assemblare l'unità avvitarne nei quattro fori da 3 mm agli angoli della Raspberry Pi 3 altrettante colonnine esagonali (o tonde) in metallo o in plastica, purché alte 15 mm, sulle quali appoggiare, mentre lo innestate nel connettore di espansione, lo shield, fissandolo poi con le viti del caso (3MA).

Fatto questo, il sistema è pronto per essere configurato, caricandovi la libreria e copiando il software sulla SD-Card.

Ricordate che per l'alimentazione dovete utilizzare un alimentatore 220Vca/5Vcc stabilizzato dotato di cavetto terminante con un connettore microUSB (perché la Raspberry Pi 3 ha a bordo un microUSB da cui prende anche l'alimentazione) e capace di erogare una corrente di almeno 1,5 ampere. Lo shield non richiede alimentazione perché ad esso provvede la Raspberry Pi.

### La libreria

Tutto il controllo dello shield ADC per Raspberry Pi è stato demandato ad una libreria già realizzata da terze parti: nel nostro caso si tratta di Adafruit. La libreria prevede la gestione della SPI sia hardware che software. Prima di entrare nel dettaglio della libreria e di spiegare come installarla, riportiamo i pin già previsti nella libreria e interessati dall'applicazione, in modo che l'hardware potrà essere costruito intorno a questi pin, senza dover poi effettuare grossi interventi sul codice della libreria.

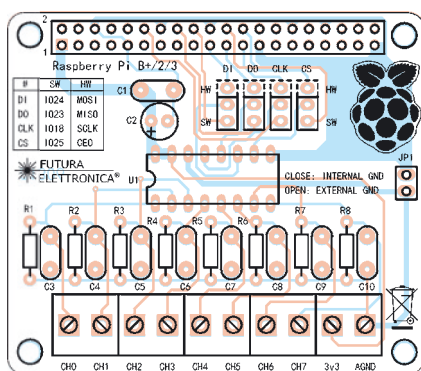
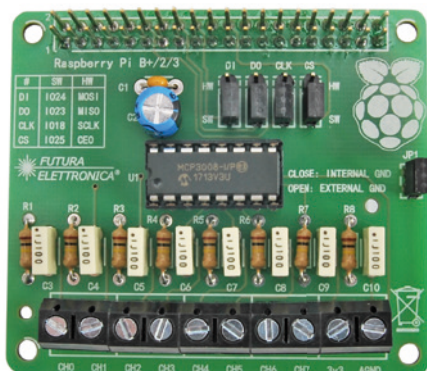
### SPI Software

MCP3008 CLK > Raspberry Pi GPIO18  
 MCP3008 DOUT > Raspberry Pi GPIO23  
 MCP3008 DIN > Raspberry Pi GPIO24  
 MCP3008 CS/SHDN > Raspberry Pi GPIO25

## CARATTERISTICHE TECNICHE

- **Compatibile con Raspberry Pi 3**
- **Ingressi ADC: 8**
- **Risoluzione ADC: 10 bit**
- **Risoluzione in tensione: 3,22 mV**
- **Range della tensione campionabile: 0÷3,3 V**
- **Frequenza di campionamento: 220 ksp/s max.**

## Piano di montaggio



### Elenco Componenti:

C1: 100 nF ceramico  
 C2: 47 µF 63 VI elettrolitico  
 C3: 100 nF 100 VL poliestere  
 C4: 100 nF 100 VL poliestere  
 C5: 100 nF 100 VL poliestere  
 C6: 100 nF 100 VL poliestere  
 C7: 100 nF 100 VL poliestere  
 C8: 100 nF 100 VL poliestere  
 C9: 100 nF 100 VL poliestere  
 C10: 100 nF 100 VL poliestere  
 R1+R8: 100 kohm  
 U1: MCP3008

Varie:

- Zoccolo 8+8
- Strip maschio 2 vie
- Strip maschio 3 vie (4 pz.)
- Jumper (5 pz.)
- Connettore Raspberry Pi 20+20
- Morsetto 2 vie passo 5.08mm (5 pz.)
- Torretta F/F 15 mm (4 pz.)
- Vite 8 mm 3 MA (8 pz.)
- Circuito stampato S1306 (65x56 mm)

### SPI hardware

MCP3008 CLK > Raspberry Pi SCLK  
 MCP3008 DOUT > Raspberry Pi MISO  
 MCP3008 DIN > Raspberry Pi MOSI  
 MCP3008 CS/SHDN > Raspberry Pi CEO

### Installazione della libreria

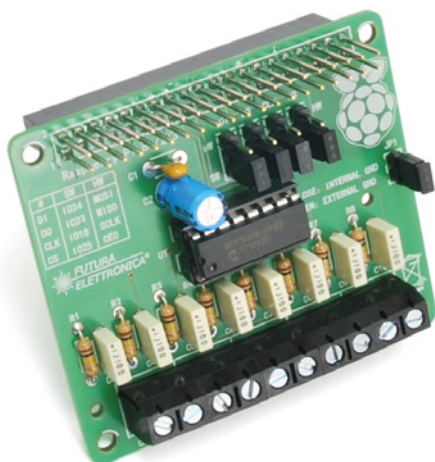
Come prima cosa è necessario caricare all'interno della Raspberry Pi il sistema operativo.

A questo proposito si ricorda che è possibile scaricare l'immagine del sistema desiderato direttamente dal sito

<https://www.raspberrypi.org/> attraverso la sezione "Download".

Tra i sistemi operativi proposti per Raspberry Pi, nel sito vi consigliamo NOOBS o Raspbian, in quanto si adattano perfettamente senza particolari difficoltà all'esecuzione del Terminale per poter installare e gestire la libreria di gestione del nostro shield.

Una volta creata la SD e avviato il sistema operativo, potremo procedere con il download ed installazione della libreria Adafruit attraverso GitHub.



Come prima cosa dovete assicurarvi che Raspberry Pi disponga di una connessione Internet, semplicemente aprendo il Browser e provando a navigare in Internet. Se la navigazione risulta funzionante, allora si potrà aprire il terminale e digitare i comandi nella sequenza indicata, seguiti da "invio":

1. `sudo apt-get update`
2. `sudo apt-get install build-essential python-dev python-smbus git`
3. `cd ~`
4. `git clone https://github.com/adafruit/Adafruit_Python_MCP3008.git`
5. `cd Adafruit_Python_MCP3008`
6. `sudo python setup.py install`

L'installazione della libreria dovrebbe essere eseguita senza particolari problemi, ma se si riscontrassero errori, si consiglia di provare ad eseguire nuovamente il comando o di riprendere dall'inizio la procedura appena esposta.

La **Fig. 1** propone la schermata che mostra i comandi impartiti, tutti con esito positivo.

### Utilizzo della libreria

A installazione terminata, si può quindi aprire l'esempio incluso nella libreria stessa, in questo modo si potrà intervenire sulla sua configurazione e provare immediatamente ad effettuare una lettura dei canali dell'MCP3008.

Sempre da terminale bisogna digitare:

1. `cd ~/Adafruit_Python_MCP3008/examples`
2. `nano simpletest.py`

Mediante i comandi indicati verrà aperto l'esempio "simpletest.py" in modalità modifica. In questo modo sarà possibile decidere quale SPI usare (hardware o software); scorrete quindi la pagina mostrata fino a trovare le dichiarazioni CLK, MISO, MOSI e CS.

Queste sono le dichiarazioni per l'uso della SPI software, pertanto se lo scopo è utilizzare proprio questa, non saranno necessarie modifiche; in caso contrario, se vorrete utilizzare la porta SPI hardware, vi sarà necessario commentare la riga 2, 3, 4, 5, 6 nell'esempio sotto riportato, antepoendo il simbolo "#" ad inizio riga, mentre dovrà essere tolto questo simbolo dalle righe inerenti alla seriale hardware quali SPI\_PORT (riga 8), SPI\_DEVICE (riga 9) e mcp (riga 10):

1. # Software SPI configuration:
2. CLK = 18
3. MISO = 23
4. MOSI = 24
5. CS = 25
6. mcp = Adafruit\_MCP3008.  
MCP3008(clk=CLK, cs=CS,  
miso=MISO, mosi=MOSI)
7. # Hardware SPI configuration:
8. # SPI\_PORT = 0
9. # SPI\_DEVICE = 0
10. # mcp = Adafruit\_MCP3008.  
MCP3008(spi=SPI.SpiDev(SPI\_PORT,  
SPI\_DEVICE))

Nel nostro caso è stato scelto di utilizzare la SPI hardware, pertanto il codice risultante sarà:

```
1. # Software SPI configuration:
2. #CLK = 18
3. #MISO = 23
4. #MOSI = 24
5. #CS = 25
6. #mcp = Adafruit_MCP3008.
   MCP3008(clk=CLK, cs=CS,
   miso=MISO, mosi=MOSI)
7. # Hardware SPI configuration:
8. SPI_PORT = 0
9. SPI_DEVICE = 0
10. mcp = Adafruit_MCP3008.
    MCP3008(spi=SPI.SpiDev(SPI_
    PORT, SPI_DEVICE))
```

Effettuate le dovute modifiche, potete salvare il file premendo “**Ctrl-o**” e di seguito

“**Invio**” per salvare il file, quindi “**Ctrl-x**” per uscire.

Ora che il file è stato salvato dovete digitare, sempre da terminale, la seguente stringa per eseguire l’esempio appena modificato e permettere la lettura delle porte analogiche.

```
sudo python simpletest.py
```

Come mostrato nella **Fig. 2**, verrà avviato il semplice programma di test che permette la lettura di tutti e 8 i canali del convertitore A/D.

Di norma, se all’ingresso non sarà collegato nulla, si dovrà vedere “0” su ogni canale, oppure un valore differente, risultato della conversione analogico/digitale se vi è un sensore connesso all’ingresso o un segnale applicato allo stesso.

```
pi@raspberrypi: ~/Adafruit_Python_MCP3008
fruit_MCP3008
byte-compiling build/bdist.linux-armv6l/egg/Adafruit_MCP3008/__init__.py to __init__.pyc
byte-compiling build/bdist.linux-armv6l/egg/Adafruit_MCP3008/MCP3008.py to MCP3008.pyc
creating build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_MCP3008.egg-info/PKG-INFO -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_MCP3008.egg-info/SOURCES.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_MCP3008.egg-info/dependency_links.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_MCP3008.egg-info/requires.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_MCP3008.egg-info/top_level.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating 'dist/Adafruit_MCP3008-1.0.0-py2.7.egg' and adding 'build/bdist.linux-armv6l/egg' to it
removing 'build/bdist.linux-armv6l/egg' (and everything under it)
Processing Adafruit_MCP3008-1.0.0-py2.7.egg
Copying Adafruit_MCP3008-1.0.0-py2.7.egg to /usr/local/lib/python2.7/dist-packages
Adding Adafruit-MCP3008-1.0.0 to easy-install.pth file

Installed /usr/local/lib/python2.7/dist-packages/Adafruit_MCP3008-1.0.0-py2.7.egg
Processing dependencies for Adafruit-MCP3008==1.0.0
Searching for Adafruit-GPIO==0.9.3
Best match: Adafruit-GPIO 0.9.3
Processing Adafruit_GPIO-0.9.3-py2.7.egg
Adafruit-GPIO 0.9.3 is already the active version in easy-install.pth

Using /usr/local/lib/python2.7/dist-packages/Adafruit_GPIO-0.9.3-py2.7.egg
Searching for spidev==3.1
Best match: spidev 3.1
Processing spidev-3.1-py2.7-linux-armv6l.egg
spidev 3.1 is already the active version in easy-install.pth

Using /usr/local/lib/python2.7/dist-packages/spidev-3.1-py2.7-linux-armv6l.egg
Finished processing dependencies for Adafruit-MCP3008==1.0.0
pi@raspberrypi:~/Adafruit_Python_MCP3008 $
```

**Fig. 1** - Comandi impartiti e loro esito.

## Listato 1

```

1. print('Reading MCP3008 values, press Ctrl-C to quit...')
2. # Print nice channel column headers.
3. print('\n {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(*range(8)))
4. print('\n' * 57)
5. # Main program loop.
6. while True:
7. # Read all the ADC channel values in a list.
8. values = [0]*8
9. for i in range(8):
10. # The read_adc function will get the value of the specified channel (0-7).
11. values[i] = mcp.read_adc(i)
12. # Print the ADC values.
13. print('\n {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(*values))
14. # Pause for half a second.
15. time.sleep(0.5)

```

Attenzione: ricordate che la tensione applicata in ingresso allo shield non deve superare i 3,3V, pena il danneggiamento dell'integrato ADC. Proprio perché il limite è 3,3V (stabilito da VREF che è posto a livello della tensione di alimentazione) il risultato della conversione con tale tensione sarà 1.023, ovvero ogni punto della conversione sarà equivalente a circa 3,22 mV di tensione.

Ad esempio, rifacendoci all'esempio sotto riportato, il valore 534 corrisponderà a  $3,22 \times 534 = 1.719$  mV mentre il valore 452 corrisponderà a 1.455 mV.

Se ad esempio volessimo effettuare un test veloce per verificare la correttezza della conversione, potremmo prendere un potenziometro, il cui centrale andrà con-

nesso ad un canale tra gli 8 disponibili, mentre le due estremità andranno collegate l'una a 3,3 V e l'altra a GND, quindi ruotarne il perno del cursore: si potrà osservare la variazione costante del valore risultante dall'A/D da 0 a 1.023.

Nell'esempio proposto nella **Fig. 2**, ogni colonna rappresenta il canale (da 0 a 7), mentre ogni riga espone il risultato del campionamento, quindi il risultato dell'operazione di conversione.

Per uscire dal programma di test dovete premere "Ctrl + c".

Per farvi comprendere meglio la logica del codice, quindi il modo in cui avviene la conversione, possiamo prendere in esame la parte di codice riportata nel **Listato 1**, ovvero la parte inclusa nel pro-

```

pi@raspberrypi: ~/Adafruit_Python_MCP3008/examples
pi@raspberrypi:~/Adafruit_Python_MCP3008/examples $ sudo python simpletest.py
Reading MCP3008 values, press Ctrl-C to quit...
 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
-----
534 | 453 | 0 | 0 | 0 | 0 | 0 | 0 |
533 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
533 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
533 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
533 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 453 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 453 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
534 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |
533 | 452 | 0 | 0 | 0 | 0 | 0 | 0 |

```

**Fig. 2** - Il programma di test che esegue la lettura degli 8 canali.

gramma “simpletest.py” da noi realizzato. Il codice riportato è abbastanza semplice e non fa altro che effettuare la lettura sequenziale degli 8 ingressi (da 0 a 7) mediante l’istruzione:

```
values[i] = mcp.read_adc(i)
```

che richiama la funzione **read\_adc()** della libreria dell’MCP3008, la quale restituirà il valore corrispondente alla conversione per ciascuno degli ingressi, correlandola

con il numero dell’ingresso interessato. Bene, con questo abbiamo detto tutto; ora sta a voi implementare l’applicazione desiderata e integrare la libreria nel vostro software.

L’articolo completo del progetto è stato pubblicato su Elettronica In n. 214

**A tutti i residenti nell'Unione Europea. Importanti informazioni ambientali relative a questo prodotto**



Questo simbolo riportato sul prodotto o sull'imballaggio, indica che è vietato smaltire il prodotto nell'ambiente al termine del suo ciclo vitale in quanto può essere nocivo per l'ambiente stesso. Non smaltire il prodotto (o le pile, se utilizzate) come rifiuto urbano indifferenziato; dovrebbe essere smaltito da un'impresa specializzata nel riciclaggio. Per informazioni più dettagliate circa il riciclaggio di questo prodotto, contattare l'ufficio comunale, il servizio locale di smaltimento rifiuti oppure il negozio presso il quale è stato effettuato l'acquisto.

Prodotto e distribuito da:

**FUTURA GROUP SRL**

**Via Adige, 11 - 21013 - Gallarate (VA)**

**Tel. 0331-799775 Fax. 0331-778112**

**Web site: [www.futurashop.it](http://www.futurashop.it)**

**Info tecniche: [www.futurashop.it/Assistenza-Tecnica](http://www.futurashop.it/Assistenza-Tecnica)**