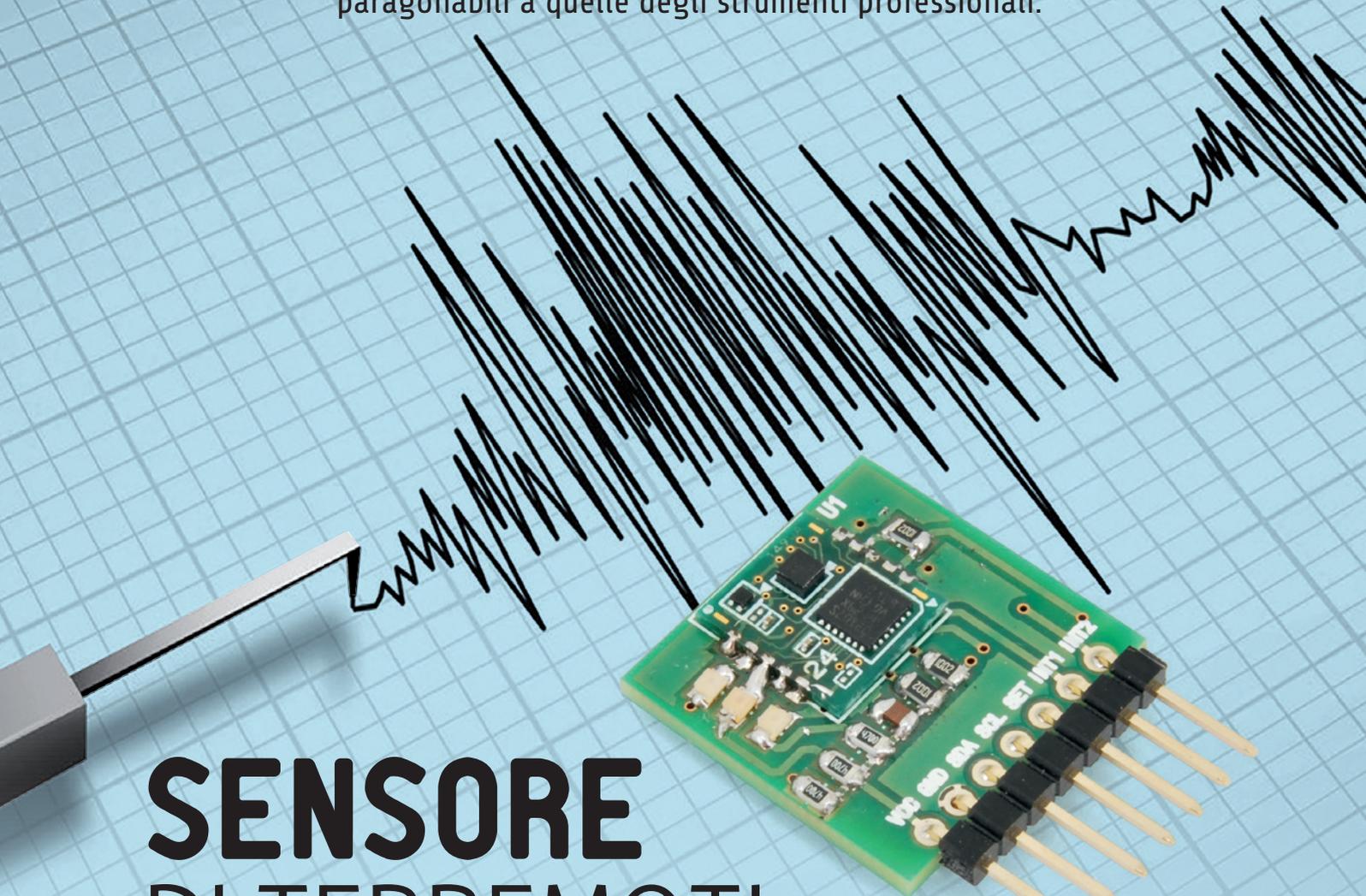


Sperimentiamo con il D75 della Omron, il più piccolo trasduttore sismico al mondo, realizzando un sismografo capace di sensibilità e precisione paragonabili a quelle degli strumenti professionali.



SENSORE DI TERREMOTI

di ALESSANDRO PASQUALINI

Ogni giorno il nostro pianeta è soggetto in media a migliaia di terremoti, solo che la maggior parte di essi è impercettibile da parte della popolazione (può essere rilevata solo dagli strumenti) e fortunatamente solo una piccola parte determina scosse abbastanza forti da essere percepite dall'uomo ed ancor meno da causare danni rilevanti. I terremoti hanno diverse origini, ma la causa più accreditata risiede nel modello della tettonica a zolle, secondo cui la Terra in origine era formata da un unico continente (ciò spiega la presenza di

specie animali uguali in continenti tra loro staccati) che si è poi separato dividendosi in quelli che oggi conosciamo e che sono cinque; tecnicamente si parla di "deriva dei continenti" ed è accertato che i continenti si muovono ancora oggi. Quando il continuo e impercettibile movimento delle placche si arresta improvvisamente, accumulando energia e tensione nelle zone coinvolte per decenni o addirittura centinaia di anni, ad un certo punto la tensione di rottura viene superata e tutta l'energia immagazzinata viene rilasciata nell'arco di qualche

secondo, generando un repentino scostamento della massa di roccia interessata e di conseguenza dando origine al fenomeno che noi e gli studiosi chiamiamo terremoto. Anche l'attività lavica sotto la crosta terrestre è collegata ai terremoti.

Il rilevamento dell'attività sismica, un tempo effettuato prettamente tramite apparati elettromeccanici detti sismografi (composti da una massa collegata a un pennino montato su un bilanciere, capace di tracciare

oscillazioni pari alla forza del movimento tellurico), negli ultimi tempi è stato affidato proficuamente all'elettronica, grazie all'esistenza di sensori come i MEMS, in grado di rilevare accelerazioni sui tre assi con estrema precisione. Sensori del genere hanno permesso la creazione di sismografi elettronici che possiamo considerare "allo stato solido" sebbene in realtà i chip su cui si basano integrano componenti che subiscono micromovimenti. Il progetto che descriviamo in

questo articolo è basato proprio su un sensore del genere e, nello specifico, su un componente della Omron per il quale abbiamo realizzato una breakout board di pronto utilizzo, ideale per prototipare applicazioni e per sperimentare con esso.

IL SENSORE D7S

Il dispositivo che presentiamo è ad oggi il più piccolo sensore sismico al mondo, è prodotto dalla Omron ed è siglato D7S. Una delle sue caratteristiche più importanti è la segnalazione, attraverso il pin INT1, di eventi sismici che potrebbero avere effetti catastrofici sulle apparecchiature elettroniche; questa funzione permette ad esempio di spegnere le apparecchiature prima che le vibrazioni causate dal terremoto possano effettivamente causare tali danni. La salvaguardia delle apparecchiature è fondamentale per riuscire, dopo un evento sismico, ad avere ancora apparati funzionanti per prevenire ulteriori eventi dannosi. Peraltro la segnalazione ottenibile dal contatto INT1 potrebbe servire ad innescare allarmi e, perché no, ad attivare forme di protezione meccanica di strumenti di misura e locali che li ospitano.

I segnali che il sensore emette sono due: shutoff e collasso del terreno; la prima situazione si verifica se il terremoto è giudicato di intensità uguale o superiore a 5 nella scala di intensità del JMA (Japan Meteorological Agency) e rispetta le condizioni definite da JEW (Japan Electrolyzed Water Association standard JWA) nello standard JWDS 0007, appendice 2, mentre la seconda si verifica se il terreno subisce un'inclinazione approssimabile a 20°.

Il sensore possiede anche una memoria interna nella quale sono immagazzinati i dati relativi agli

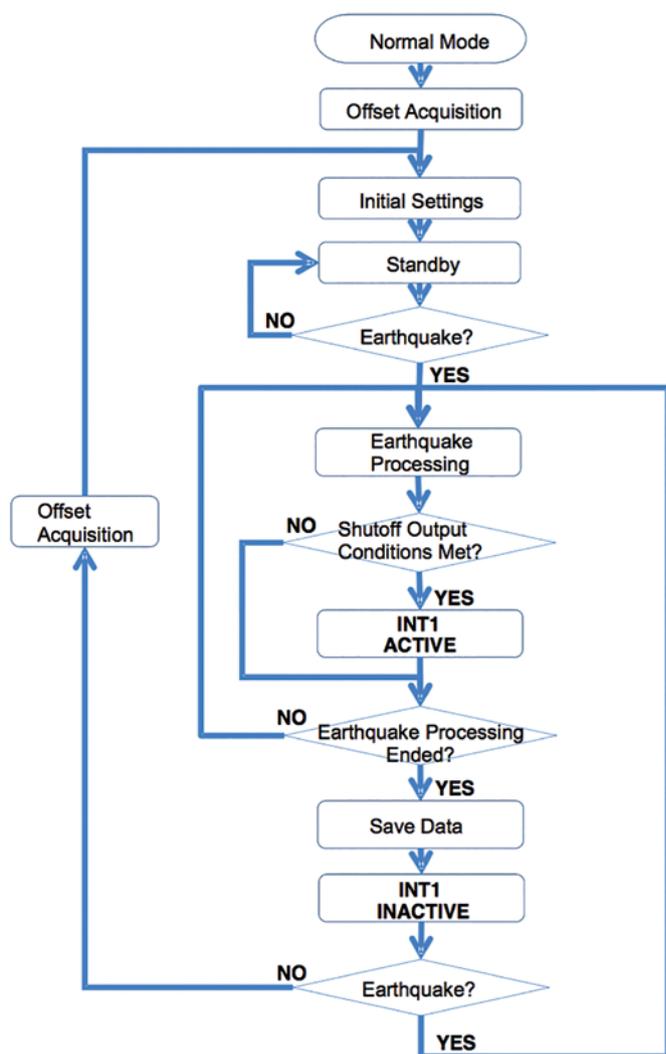


Fig. 1 - Schema del ciclo di rilevazione dei terremoti, dove normal mode corrisponde alla modalità di standby.

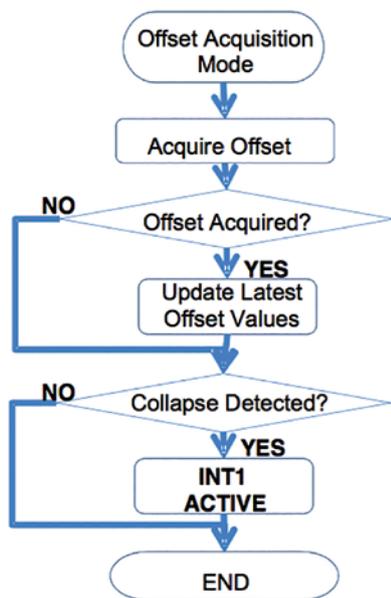


Fig. 2 - Flusso di determinazione della condizione di collasso forzando l'acquisizione degli offset.

ultimi cinque terremoti registrati e i cinque di più grande entità, oltre, naturalmente, a tutte le impostazioni di configurazione. Il D7S è composto da un accelerometro a tre assi, dei quali solamente due sono utilizzati durante la rilevazione di un sisma e sono selezionabili sia dall'utente sia automaticamente rispetto all'inclinazione del sensore. La presenza di un bus I²C permette di modificare le impostazioni del sensore, oppure leggere i dati relativi ai terremoti, da parte di qualsiasi microcontrollore che sia dotato di tale bus ed anche da Arduino. Il sensore D7S dispone in tutto di tre pin funzione, di cui due (INT1 e INT2) sono pin di segnalazione ed il terzo (SET) è una linea utilizzata per variare lo stato di funzionamento. Prima di utilizzare il sensore è necessario eseguire la procedura di installazione iniziale, ovvero il sensore deve rilevare gli offset degli assi selezionati e salvarne il valore nella memoria interna; questi offset verranno utilizzati per discriminare la condizione di collasso, confrontandoli con

quelli attuali al momento della rilevazione dell'evento sismico. Dopo la fase di installazione iniziale il sensore entra in standby fino all'inizio di un terremoto, quando inizia il calcolo dei dati relativi al sisma; rimane in questo stato fino a quando non è stato giudicato il termine del terremoto. A questo punto viene aggiornata la memoria interna con i dati di recente rilevazione. Ogni volta che viene alimentato, il sensore si porta in modalità di rilevazione degli offset e determina se la condizione di collasso si è verificata e, in caso affermativo, varia la condizione logica del pin INT1 portandola al livello basso. Se la condizione non è verificata, il sensore entra in modalità standby ed ha inizio il ciclo di

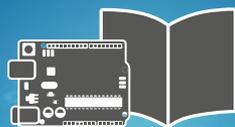
rilevazione dei terremoti (Fig. 1). È bene specificare che la discriminazione della condizione di collasso non avviene solamente all'accensione del sensore ma ogniqualvolta il sensore entra in standby; rimane, infatti, possibile forzare il controllo della condizione di collasso portando il sensore nella modalità di acquisizione offset (Fig. 2). I dati che il sensore calcola per ogni evento sismico sono PGA (Peak Ground Acceleration), SI (Spectral Intensity) e la temperatura ambientale media alla quale l'evento si è verificato. Durante il calcolo, ovvero nel corso di un sisma, è possibile leggere il valore istantaneo di PGA e SI accedendo ad alcuni specifici registri. All'interno del sensore è stata

| Registro | Accesso | Descrizione | Valori |
|------------------------|---------|---|---|
| STATE (0x1000) | R | Stato corrente | <ul style="list-style-type: none"> - 0x00 standby - 0x01 rilevazione terremoti - 0x02 installazione iniziale - 0x03 acquisizione offset - 0x04 autodiagnostica |
| AXIS_STATE (0x1001) | R | Assi utilizzati per la rilevazione | <ul style="list-style-type: none"> - 0x00 assi YZ - 0x01 assi XZ - 0x02 assi XY |
| EVENT (0x1002) | R | Eventi generati | Bit 0 1 se viene generato shutdown Bit 1 1 se viene generato collasso Bit 2 1 se si verifica un errore in autodiagnostica Bit 3 1 se si verifica errore in acquisizione offset |
| MODE (0x1003) | R/W | Modo corrente | <ul style="list-style-type: none"> - 0x01 standby - 0x02 installazione iniziale - 0x03 acquisizione offset - 0x04 autodiagnostica È possibile cambiare il modo corrente solamente se il sensore si trova nello stato standby. |
| CTRL (0x1004) | R/W | Assi da utilizzare durante la rilevazione | Bit da 4 a 6 <ul style="list-style-type: none"> - 0x00 usa assi YZ - 0x01 usa assi XZ - 0x02 usa assi XY - 0x03 scegli automaticamente gli assi da utilizzare - 0x04 scegli automaticamente gli assi da utilizzare durante l'installazione iniziale |
| | | Soglia per il segnale di shutdown | Bit 3 <ul style="list-style-type: none"> - 0 per la soglia alta - 1 per la soglia bassa |
| CLEAR_COMMAND (0x1005) | R/W | Comandi di cancellazione della memoria | Bit 0 Se 1 vengono cancellati dati dei terremoti Bit 1 Se 1 vengono cancellate le informazioni di autodiagnostica Bit 2 Se 1 vengono cancellate le ultime informazioni relative agli offset Bit 3 Se 1 vengono cancellate le informazioni relative all'installazione iniziale |

Tabella 1 - Descrizione riassuntiva dei registri di stato e configurazione.

Vuoi entrare nell'affascinante mondo di **ARDUINO**
e imparare a utilizzarlo al meglio?

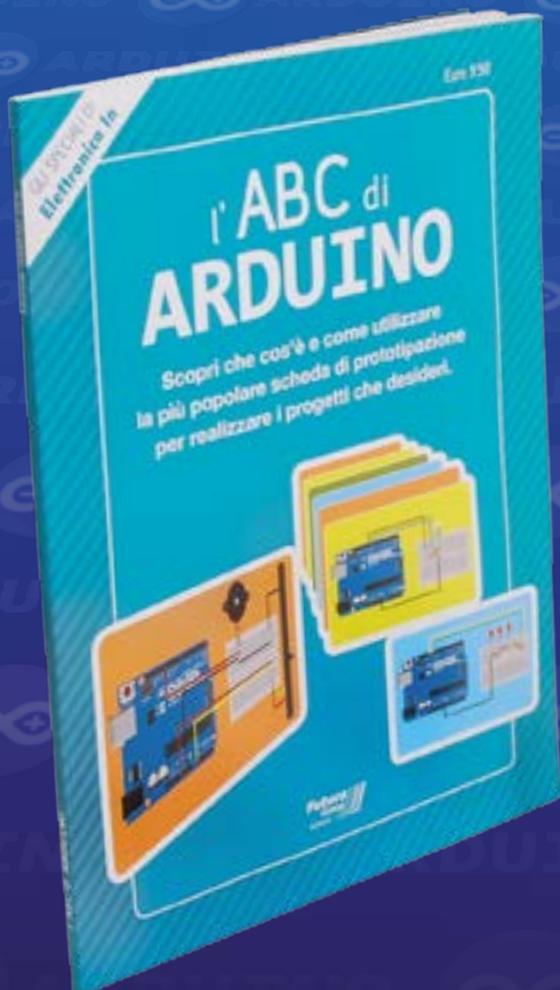
Allora non perderti questa
FANTASTICA OFFERTA!



B&B

Board & Book

Acquista insieme libro & scheda:
avrà uno **sconto di 4,50 euro!**



IVA INCLUSA

€ **29,90**

anziché

~~€ 34,40~~

cod. ARDUKITBOOK1



| # | Indirizzo (ultimi 8 bit) | Nome | Accesso | Descrizione |
|----|--------------------------|-----------------|---------|---|
| 1 | 0x00 | MAIN_OFFSET_X_H | R | Offset sull'asse X. |
| 2 | 0x01 | MAIN_OFFSET_X_L | R | |
| 3 | 0x02 | MAIN_OFFSET_Y_H | R | Offset sull'asse Y. |
| 4 | 0x03 | MAIN_OFFSET_Y_L | R | |
| 5 | 0x04 | MAIN_OFFSET_Z_H | R | Offset sull'asse Z. |
| 6 | 0x05 | MAIN_OFFSET_Z_L | R | |
| 7 | 0x06 | MAIN_T_AVE_H | R | Temperatura a cui si è verificato il sisma. |
| 8 | 0x07 | MAIN_T_AVE_L | R | |
| 9 | 0x08 | MAIN_SI_H | R | Valore di SI registrato. |
| 10 | 0x09 | MAIN_SI_L | R | |
| 11 | 0x0A | MAIN_PGA_H | R | Valore di PGA registrato. |
| 12 | 0x0B | MAIN_PGA_L | R | |

Tabella 2 - Descrizione riassuntiva del blocco di registri contenenti le informazioni sui terremoti.

implementata una funzione di autodiagnostica, molto utile per verificare se il D7S funziona correttamente, ma la stessa deve essere attivata manualmente attraverso la scrittura di un registro usando il bus I²C. Ad ogni operazione di acquisizione degli offset, il sensore determina automaticamente se l'operazione è andata a buon fine, altrimenti aggiorna uno specifico registro, segnalando di conseguenza il guasto.

I pin funzionali del sensore sono INT1, INT2 e SET: il primo corrisponde, come già evidenziato, alla segnalazione delle condizioni di shutoff e collasso mentre INT2 permette di conoscere se il sensore si trova in standby, se il sensore è in fase di rilevazione di un terremoto, se è in corso un'operazione di acquisizione degli offset oppure se è stata attivata la funzione di autodiagnostica. L'ultimo pin, SET, permette di portare il sensore, attraverso un impulso esterno, in modalità di installazione iniziale, senza dover necessariamente ricorrere all'utilizzo del bus I²C.

Quando il pin INT1 viene portato al livello logico basso da un evento shutoff o collasso, il valore predefinito, ovvero l'1 logico, può essere ripristinato solamente leggendo il registro EVENT (vedi prossimo paragrafo), eseguendo la procedura di installazione

iniziale oppure togliendo l'alimentazione al sensore.

LA MEMORIA INTERNA

Per poter utilizzare il D7S è necessario conoscere come programmare i principali parametri di configurazione, oltre che l'ubicazione dei dati che il sensore calcola per ogni terremoto. La memoria interna è configurata in registri di 8 bit indirizzabili

con indirizzi di 16 bit. Dunque, utilizzando, ad esempio, la libreria Wire di Arduino, è necessario inviare prima gli 8 bit più significativi dell'indirizzo, seguiti dagli 8 bit meno significativi.

Non tutti i registri risultano accessibili e il produttore dichiara esplicitamente di fare attenzione a non accedere a quelli il cui accesso è proibito per evitare di pregiudicare il corretto funzionamento del sensore stesso.

I primi tre registri che si incontrano nella memoria sono STATE, AXIS_STATE ed EVENT (accessibili solamente in lettura) e contengono le informazioni sullo stato corrente del sensore e le sue attuali configurazioni: il registro STATE contiene nei suoi 3 bit meno significativi lo stato, il registro AXIS_STATE è composto da 2 bit in cui sono indicati gli assi utilizzati per le rilevazioni, mentre il registro EVENT è

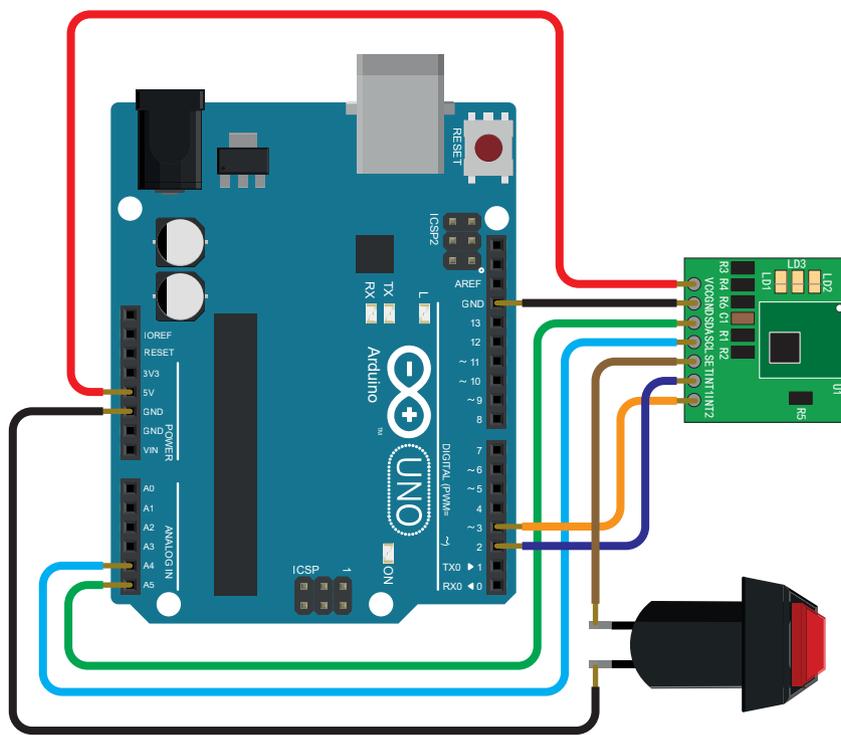


Fig. 3 - Schema di collegamento del sensore D7S con Arduino UNO. Il pulsante (di tipo normalmente aperto) è opzionale.

| # | Colore del LED | Descrizione |
|---|----------------|--|
| 1 | VERDE | Indica che il sensore è alimentato. |
| 2 | ARANCIONE | Si accende quando il sensore è in fase di registrazione di un terremoto, acquisizione offset e autodiagnostica. Il led rimane spento quando il sensore è in standby. |
| 3 | ROSSO | Si accende quando si verifica la condizione di shutoff o di collasso. |

Tabella 3 - Descrizione riassuntiva dei LED presenti nella breakout.

formato da 4 bit, ognuno relativo ad un possibile evento che può verificarsi.

I successivi tre registri sono MODE, CTRL e CLEAR_COMMAND, accessibili sia in lettura che in scrittura e consentono di modificare il comportamento del sensore: il primo registro MODE consente di cambiare lo stato del sensore, il successivo CTRL permette di selezionare gli assi da utilizzare e la soglia di shutoff, mentre CLEAR_COMMAND permette, portando ad 1 i suoi bit, l'azzeramento di determinate porzioni della memoria.

Nella **Tabella 1** sono riportati sinteticamente tutti i registri di configurazione e sono esposti i possibili valori che possono assumere.

Nei registri dall'indirizzo 0x2000 a 0x2003 si trovano le informazioni istantanee dell'evento sismico in corso (se il sensore è in standby i registri sono azzerati) e sono così suddivisi: i primi due registri, rispettivamente 0x2000 e 0x2001, contengono il valore di SI diviso in parte alta e parte bassa, mentre nei restanti registri 0x2002 e 0x2003 è presente il valore di PGA, sempre diviso in parte alta e parte bassa.

Successivamente sono presenti dieci blocchi di registri, posizionati dall'indirizzo 0x30XX a 0x39XX, e contengono i dati dei terremoti: i primi cinque corrispondono agli ultimi cinque eventi sismici registrati, mentre gli ultimi cinque sono i terremoti di più forte intensità.

Tutti i blocchi sono suddivisi come mostrato nella **Tabella 2**, dove i primi sei registri corrispondono a informazioni di

offset, due registri contengono la temperatura alla quale l'evento sismico si è verificato, due contengono il valore di SI e, infine, due registri memorizzano il valore di PGA registrato. L'indirizzo dello specifico registro deve essere costruito sommando l'indirizzo del blocco all'indirizzo del registro all'interno del blocco.

Infine sono presenti dei blocchi di registri relativi alle informazioni di installazione iniziale,

degli ultimi offset registrati e dell'autodiagnostica. Questi dati non sono strettamente attinenti all'applicazione, quindi rimandiamo, per chi fosse interessato, al datasheet del D7S.

LA NOSTRA SCHEDA

Il progetto che vi proponiamo è quello di un sismografo che si basa su una breakout board a bordo della quale si trova il sensore integrato; la scheda ne rende accessibile il bus di comunicazione attraverso comodi pin-strip, in modo da semplificare notevolmente l'utilizzo del sensore ed anche l'integrazione in apparati esistenti. Sul connettore troviamo,

Le onde generate dal terremoto

Ad un terremoto sono associate principalmente due onde chiamate P e S generate nell'ipocentro della scossa tellurica.

Le onde P sono longitudinali e sono caratterizzate dal fatto che imprimono alle particelle di roccia un'oscillazione uguale al moto di propagazione dell'onda. Esse generano compressioni e rarefazioni nel materiale che attraversano. Queste onde si propagano velocemente e attraversano tutti i tipi di materiale. Sono anche chiamate onde principali. Le onde S sono onde secondarie, in quanto viaggiano ad

una velocità inferiore delle corrispondenti onde P. Sono onde trasversali e di conseguenza possono propagarsi solamente attraverso materiale roccioso e non possono attraversare né fluidi né gas (Se ad esempio il terremoto avesse origine nell'oceano si avrebbe la generazione solamente delle onde P, in quanto le onde S non possono propagarsi in acqua). Queste onde sono utilizzate per determinare il valore di intensità secondo le varie scale esistenti. Ad esempio il sensore D7S utilizza le onde S per determinare PGA e SI di ogni terremoto.

| # | Nome pin | Direzione/Funzione | Descrizione |
|---|----------|--------------------|--|
| 1 | VCC | Alimentazione | Da 3.3V a 5V |
| 2 | GND | Alimentazione | / |
| 3 | SDA | I2C | Linea dati del bus I2C. |
| 4 | SCL | I2C | Linea clock del bus I2C |
| 5 | SET | IN | Quando viene portato al valore basso cambia lo stato del sensore portandolo in fase di <i>installazione iniziale</i> . Quando alto, il sensore è in modalità di funzionamento normale. |
| 6 | INT1 | OUT | Assume il valore basso quando si verifica la condizione di shutoff o di collasso. Mantiene il valore alto altrimenti. |
| 7 | INT2 | OUT | Assume il valore basso durante la fase di rilevazione di un terremoto, acquisizione offset e autodiagnostica. Mantiene il valore basso quando il sensore è in standby. |

Tabella 4 - Descrizione della pin-out della breakout.

VUOI SVILUPPARE LE TUE APPLICAZIONI CON ARDUINO?

Da noi trovi tutto quello che ti serve!



L'ABC di ARDUINO

cod. ABCARDU

€ 9,90



Arduino e le tecniche di programmazione dei microcontrollori Atmel

cod. ATPROMA

€ 15,00



ARDUINO UNO Programmazione avanzata e librerie di sistema

cod. ARDUADVANCED

€ 22,00

Arduino Uno Rev3

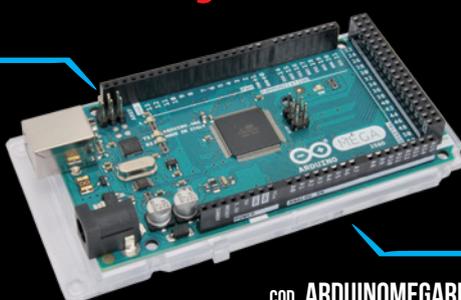
Arduino Mega

Arduino Due



cod. ARDUINOUNOREV3

€ 24,50



cod. ARDUINOMEGAREV3

€ 43,00



ARDUINODUE

€ 44,00

Prezzi IVA inclusa

kit V5 con Arduino Uno REV3

Set contenente tutti i componenti necessari per realizzare gli esperimenti descritti nel libro "L'ABC di Arduino".



cod. ARDUKITV5

€ 59,00

Starter kit per robot con Arduino Uno

Set contenente le parti necessarie per realizzare il Robot descritto nel libro incluso.

€ 155,00

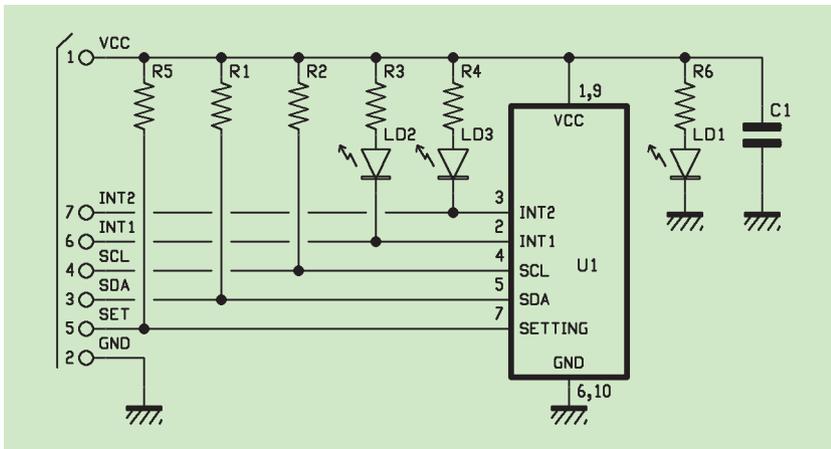


cod. ARDUKITV4

FUTURA ELETTRONICA®
www.futurashop.it

Futura Group srl • via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Caratteristiche tecniche di questi prodotti e acquisti on-line su www.futurashop.it



provvisti di opportuni resistori di pull-up, i pin SDA, SCL, SET, INT1 e INT2 del D7S. In più abbiamo deciso di aggiungere tre LED: uno è connesso all'alimentazione e consente di verificare la presenza della tensione di alimentazione del sensore, mentre

gli altri due sono applicati ai pin INT1 e INT2. La funzione dei tre diodi luminosi è descritta nella **Tabella 3**.

Il pin SET può agevolmente essere collegato a un pulsante, senza doversi preoccupare della resistenza di pull-up perché già

predisposta nella breakout, e può essere utilizzato per portare il sensore in modalità di installazione iniziale semplicemente premendolo, evitando, di conseguenza, di dover utilizzare il bus I²C. Il collegamento con Arduino è anch'esso molto semplice, infatti basta semplicemente connettere i pin di alimentazione del sensore (VCC e GND) con i pin 5V e GND di Arduino e i pin SDA e SCL (in Arduino UNO i pin adibiti al bus I²C sono rispettivamente A4 e A5).

Arduino permette anche la gestione di interrupt, funzionalità molto utile da abbinare con il sensore presentato perché permette di reagire istantaneamente agli eventi generati dal D7S. Per

```

/dev/cu.usbmodem1421 (Arduino/Genuino Uno)
Starting D75 communications (it may take some time)...STARTED
--- LASTEST EARTHQUAKES MEASURED ---
Earthquake n. 1
  SI: 0.20 [m/s]
  PGA (Peak Ground Acceleration): 9.90 [m/s^2]
  Temperature: 22.00 [°C]
Earthquake n. 2
  SI: 0.36 [m/s]
  PGA (Peak Ground Acceleration): 11.73 [m/s^2]
  Temperature: 21.00 [°C]
Earthquake n. 3
  SI: 0.23 [m/s]
  PGA (Peak Ground Acceleration): 9.51 [m/s^2]
  Temperature: 21.00 [°C]
Earthquake n. 4
  SI: 0.00 [m/s]
  PGA (Peak Ground Acceleration): 0.00 [m/s^2]
  Temperature: 0.00 [°C]
Earthquake n. 5
  SI: 0.00 [m/s]
  PGA (Peak Ground Acceleration): 0.00 [m/s^2]
  Temperature: 0.00 [°C]
  
```

Fig. 4 - Output del serial monitor di Arduino IDE eseguendo il Listato 1.

Listato 1

```
#include <D7S.h>

void setup() {
  Serial.begin(9600);
  while (!Serial)
    ;
  //Inizializziamo la comunicazione con il sensore D7S
  Serial.print("Starting D7S communications (it may take some time)...");
  D7S.begin();
  while (!D7S.isReady()) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("STARTED");

  //Preleviamo i dati dal sensore
  Serial.println("--- LATEST EARTHQUAKES MEASURED ---\n");
  for (int i = 0; i < 5; i++) { //gli indici variano da 0 a 4 (5 indici in totale)
    Serial.print("Earthquake n. ");
    Serial.println(i+1);
    //Stampa SI
    Serial.print("\tSI: ");
    Serial.print(D7S.getLastestSI(i));
    Serial.println(" [m/s]");
    //Stampa PGA
    Serial.print("\tPGA (Peak Ground Acceleration): ");
    Serial.print(D7S.getLastestPGA(i));
    Serial.println(" [m/s^2]");
    //Stampa temperatura
    Serial.print("\tTemperature: ");
    Serial.print(D7S.getLastestTemperature(i));
    Serial.println(" [°C]\n");
  }
}

void loop() {
}
```

utilizzarla è sufficiente collegare i pin 2 e 3 di Arduino con i pin INT1 e INT2 e registrare l'esecuzione dell'ISR (Interrupt Service Routine) sul pin 2 quando si verifica l'evento FALLING (ovvero la transizione da valore logico alto a valore logico basso), mentre quella sul pin 3 al verificarsi di CHANGE (entrambe le transizioni logiche alto-basso e basso-alto). Nella **Tabella 4** viene dettagliata la piedinatura del connettore della breakout board (fate riferimento al piano di montaggio visibile in queste pagine): il primo gruppo è formato dai pin di alimentazione, il secondo dai pin del bus I²C, mentre gli ultimi tre sono i pin funzionali del sensore. Le connessioni del caso sono illustrate nella **Fig. 3**, che riporta l'hardware completo del progetto; la realizzazione si effettua semplicemente cablando median-

Misurare l'intensità del terremoto

Nel tempo, per lo studio e la classificazione dei terremoti sono state introdotte diverse scale di misura dell'intensità tra le quali le più importanti e significative sono magnitudo, Mercalli e PGA (Peak Ground Acceleration). La scala magnitudo è una misura dell'energia meccanica sprigionata da un evento sismico all'epicentro. Nella definizione proposta da Charles Richter (magnitudo Richter), e maggiormente utilizzata nel modo, il magnitudo viene definito su base puramente strumentale, evitando quindi gli errori soggettivi dovuti all'osservatore. La relativa intensità viene definita su scala logaritmica ed è messa in diretta relazione con la quantità di TNT (tritololo) equivalente per generare la stessa

quantità di energia sprigionata. La scala Mercalli, invece, è una soluzione empirica che valuta l'intensità di un terremoto sulla base degli effetti che produce su persone, cose ed edifici; ciò implica che due eventi sismici aventi stessa intensità magnitudo ma localizzati uno nel deserto, ovvero una zona disabitata, e l'altro in un centro abitato, avranno due valori differenti secondo la scala Mercalli. Il PGA è la misura della massima accelerazione del suolo indotta dal terremoto e misurata da accelerometri. Proprio per la sua natura essa è facilmente registrabile e fornisce un dato strumentale con una stretta relazione con l'equivalente valore della scala Mercalli.

Fishino

Design With Simplicity

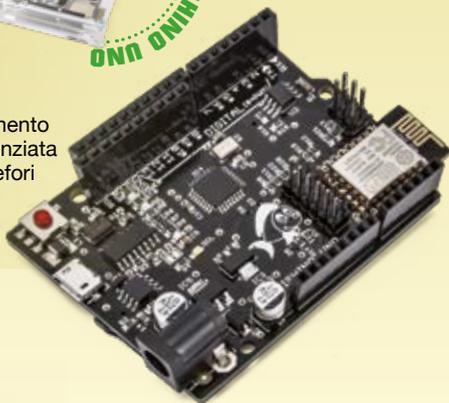
I TUOI PROGETTI DIVENTANO WIRELESS

FISHINO UNO

- Alimentazione:
 - da 7 a 12 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Compatibile al 100% con Arduino UNO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Sezione di alimentazione a 3,3 V potenziata
- Compatibile con shield e schede millefori
- Peso 25 g
- Dimensioni 76,5 x 53,5 x 14 mm



cod. FISHINOUNO
€ 36,00



cod. FISHINOMEGA
€ 49,90



FISHINO MEGA

- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino MEGA
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Stadio di alimentazione switching (5 Vdc e 3,3 Vdc)
- Compatibile con shield e schede millefori
- Peso 37 g
- Dimensioni 101,5 x 53,5 x 15 mm



FISHINO32

- Controller PIC32 bit
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug o sull'ingresso Vin)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile con Arduino UNO
- Modulo WiFi
- Modulo RTC con batteria di mantenimento
- Clock 120 MHz, riducibile via software
- Interfaccia per scheda microSD
- Codec audio stereo
- Peso 29 g
- Dimensioni 76,5 x 53,5 x 14 mm

cod. FISHINO32
€ 59,00



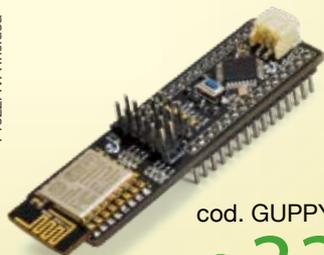
Fishino UNO, Fishino MEGA, Fishino GUPPY, Fishino32 e Fishino PIRANHA sono schede di sviluppo Arduino-compatibile dotate di WiFi e lettore di schede microSD. Grazie ad un completo pacchetto di librerie (scaricabile dal sito www.fishino.it), si ha la possibilità di gestire tutti i propri progetti dal web in maniera semplice e veloce!

FISHINO PIRANHA

- Alimentazione:
 - 3,7 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug)
 - da 3,5 a 20 Volt sull'ingresso Vin
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino MKR1000
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Peso 10 g
- Dimensioni 62 x 25 x 16 mm



Prezzi IVA inclusa



cod. GUPPY
€ 33,90

FISHINO GUPPY

- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 6,5 a 20 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino NANO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Peso 10 g
- Dimensioni 75 x 20 x 18,5 mm

te jumper il circuito riportato. Quanto alla breakout board, è possibile autocostruirla riferendosi al piano di montaggio visibile in queste pagine: una volta inciso e forato il PCB (si parte dalle sue tracce lato rame che potete scaricare dal nostro sito www.elettronica.in insieme ai file del progetto) disponetevi i componenti a partire dall'integrato D7S, rispettando il verso indicato dal piano di montaggio. Trattandosi di componenti SMD è opportuna una certa precisione, nonché l'utilizzo di un saldatore a punta finissima e potenza di 20W massimi, oltre che di filo di lega saldante da 0,5 mm di diametro, pasta flussante e una lente d'ingrandimento per verificare le saldature e l'eventuale cortocircuito tra pad vicini. Affinché il sensore possa rilevare fedelmente i sismi, è opportuno che il circuito stampato sia avvitato saldamente a una struttura rigidamente ancorata al pavimento dell'edificio o a una base in cemento stabile al suolo, ovvero a una staffa metallica

Listato 2

```
#include <D7S.h>
#define INT1_PIN 2 //Pin di Arduino collegato al pin INT1 del sensore D7S
#define INT2_PIN 3 //Pin di Arduino collegato al pin INT2 del sensore D7S

//Funzione per gestire lâ€™evento di inizio di un terremoto
void startEarthquakeHandler() {
    Serial.println("----- EARTHQUAKE STARTED! -----\n");
}

//Funzione per gestire lâ€™evento di fine di un terremoto
void endEarthquakeHandler(float si, float pga, float temperature) {
    Serial.println("----- EARTHQUAKE ENDED! -----");
    //Stampa SI
    Serial.print("\tSI: ");
    Serial.print(si);
    Serial.println(" [m/s]");
    //Stampa PGA
    Serial.print("\tPGA (Peak Ground Acceleration): ");
    Serial.print(pga);
    Serial.println(" [m/s^2]");
    //Stampa temperatura
    Serial.print("\tTemperature: ");
    Serial.print(temperature);
    Serial.println(" [Ã°C]\n");
    //resettiamo gli eventi
    D7S.resetEvents();
}

//Funzione per gestire lâ€™evento shutoff
void shutoffHandler() {
    Serial.println("----- SHUTOFF! -----\n");
    Serial.println("Shutting down all device!");
    //stop all device
    while (1)
        ;
}

//Funzione per gestire lâ€™evento collasso del terreno
void collapseHandler() {
    Serial.println("----- COLLAPSE! -----\n");
}

void setup() {
    Serial.begin(9600);
}
```

(Continua)

Fig. 5 - Output del serial monitor di Arduino IDE eseguendo il Listato 2.

Listato 2 - Segue

```
while (!Serial)
;
//Inizializziamo la comunicazione con il sensore D7S
Serial.print("Starting D7S communications (it may take some time)...");
D7S.begin();
while (!D7S.isReady()) {
    Serial.print(".");
    delay(500);
}
Serial.println("STARTED");
//Il sensore sceglier  automaticamente gli assi da usare durante la fase di installazione
Serial.println("Setting D7S sensor to switch axis at initialization time.");
D7S.setAxis(SWITCH_AT_INSTALLATION);
//Abilitiamo gli interrupt
D7S.enableInterruptINT1(INT1_PIN);
D7S.enableInterruptINT2(INT2_PIN);
//Registriamo gli handler degli eventi
D7S.registerInterruptEventHandler(START_EARTHQUAKE, &startEarthquakeHandler);
D7S.registerInterruptEventHandler(END_EARTHQUAKE, &endEarthquakeHandler);
D7S.registerInterruptEventHandler(SHUTOFF_EVENT, &shutoffHandler);
D7S.registerInterruptEventHandler(COLLAPSE_EVENT, &collapseHandler);
//Inizializziamo il sensore
Serial.println("Initializing the D7S sensor in 2 seconds. Please keep it steady.");
delay(2000);
Serial.print("Initializing...");
D7S.initialize();
while (!D7S.isReady()) {
    Serial.print(".");
    delay(500);
}
Serial.println("INITIALIZED!");
//verifichiamo che non ci siano stati collassi
if (D7S.isInCollapse()) {
    collapseHandler();
}
//resettiamo gli eventi (in modo da determinare gli eventi giusti)
D7S.resetEvents();
//iniziamo la gestione degli eventi tramite interrupt
D7S.startInterruptHandling();
Serial.println("\nListening for earthquakes!");
}

void loop() {
}
```

avvitata al muro; evitate l’inserzione tramite il pin-strip (perch  l’ancoraggio meccanico non   rigido abbastanza) ma semmai collegate la scheda mediante fili.

LA LIBRERIA PER ARDUINO

Abbiamo sviluppato una libreria per utilizzare il sensore con Arduino, evitando di dover gestire direttamente i registri con il bus I²C e la libreria Wire. La libreria permette sia di modificare la configurazione, sia di leggere i dati dei terremoti dalla memoria del sensore; provvede anche alla gestione degli eventi interrupt generati dal sensore D7S e il suo utilizzo   mostrato nel **Listato 2**. Il **Listato 1** mostra come poter

accedere alla memoria interna del sensore per prelevare i dati degli ultimi cinque terremoti registrati. Nella funzione setup   necessario eseguire una chiamata al metodo **begin()** che inizializza la comunicazione con il sensore. Prima di iniziare ad inviare comandi al sensore   necessario attendere che il D7S sia pronto, ovvero che si trovi in standby. A questo punto   possibile accedere alla memoria del sensore utilizzando il metodo **getLastestSI(index)**, **getLastestPGA(index)**, **getLastestTemperature(index)**, in cui “index” indica l’indice del terremoto di cui vogliamo prelevare i dati (i terremoti salvati sono cinque, dunque l’indice

pu  variare da 0 a 4). La libreria implementa anche i metodi equivalenti **getRankedSI(index)**, **getRankedPGA(index)**, **getRankedTemperature(index)** per accedere ai dati relativi ai cinque sismi di maggiore entit  registrati in memoria.

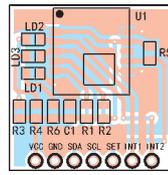
Nel **Listato 2** vediamo il firmware che realizza un sismografo utilizzando gli interrupt per la gestione degli eventi interrupt generati dal sensore D7S. Per poter utilizzare questo esempio bisogna collegare i pin 2 e 3 di Arduino rispettivamente ai pin INT1 e INT2 del sensore e registrare le ISR contenute nella libreria e chiamando i due metodi **enableInterruptINT1(pin)** e **enableInterruptINT2(pin)**,

Elenco Componenti:

C1: 100 nF ceramico (0805)
 R1, R2, R5: 10 kohm (0805)
 R3, R4, R6: 470 ohm (0805)
 U1: D7S-A0001
 LD1: LED verde (0805)
 LD2: LED rosso (0805)
 LD3: LED giallo (0805)

Varie:

- Pin strip 90° 7 vie
- Circuito stampato S1342 (22x22 mm)



dove il parametro "pin" indica il pin di Arduino cui sono collegati INT1 e INT2..

Per utilizzare questa funzionalità bisogna registrare le ISR contenute nella libreria e per fare ciò è necessario chiamare i due metodi `enableInterruptINT1(pin)` e `enableInterruptINT2(pin)`, dove il parametro "pin" indica il pin di Arduino cui sono collati INT1 e INT2. Queste chiamate consentono alla libreria di gestire automaticamente gli eventi interrupt e di chiamare gli handler definiti dall'utente. Per registrare gli handler si esegue una chiamata al metodo `registerInterruptEventHandler(event, handler)` dove "event" è l'evento generato dal sensore che si vuole gestire (START_EARTHQUAKE, END_EARTHQUAKE, SHUTOFF_EVENT o COLLAPSE_EVENT) e "handler" è il puntatore alla funzione che deve gestire l'evento. Particolare attenzione va posta alla funzione per l'evento END_EARTHQUAKE che necessita di una firma diversa dalle altre, in quanto alla sua invocazione vengono passati come parametri i tre dati del terremoto appena terminato (SI, PGA e temperatura). A questo punto è necessario inizializzare il sensore, ovvero eseguire la procedura di installazione iniziale, per registrare gli

offset di riferimento degli assi dell'accelerometro. Si può notare nel **Listato 2** che il D7S viene configurato in modo tale da scegliere gli assi da utilizzare al momento dell'installazione iniziale, chiamando il metodo di configurazione `setAxis(SWITCH_AT_INSTALLATION)`; le altre possibili impostazioni sono FORCE_YZ, FORCE_XZ, FORXE_XY e AUTO_SWITCH che corrispondono alle varie possibilità elencate in **Tabella 1**. Terminata la fase di inizializzazione è possibile abilitare la gestione degli eventi interrupt (che per impostazione predefinita è disabilitata per evitare dati inconsistenti), chiamando il metodo `startInterruptHandling()`; ma prima di ciò è necessario resettare il registro eventi del sensore, per evitare di gestire eventi generati in terremoti precedenti, chiamando `resetEvents()`

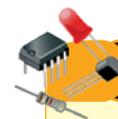
L'utilizzo del **Listato 2** merita una nota: per come è costruito il sensore D7S una volta che viene generato l'evento shutoff o collasso il LED collegato al pin INT1 dovrebbe accendersi, ma gli interrupt ne impediscono l'accensione perché gestiscono istantaneamente l'evento (è possibile comunque che il LED effettui un lampeggio singolo). La libreria prevede anche i metodi `selftest()` e `acquireOffset()` per eseguire

l'autodiagnostica e forzare l'acquisizione degli offset e i metodi associati `getSelftestResult()` e `getAcquireOffsetResult()` per ottenere il risultato di entrambe le operazioni.

I metodi `clearEarthquakeData()`, `clearInstallationData()`, `clearLastestOffsetData()`, `clearSelftestData()`, `clearAllData()`, permettono di azzerare la memoria. All'interno della libreria sono comunque disponibili ulteriori esempi per dimostrare il funzionamento di tutte le funzionalità messe a disposizione.

CONCLUSIONI

In questo articolo vi abbiamo presentato il più piccolo sensore sismico al mondo, descrivendone le caratteristiche, il funzionamento, l'interfacciamento e l'utilizzo, fornendovi tutte le informazioni affinché possiate sviluppare autonomamente progetti che lo impiegano. Abbiamo altresì proposto un'applicazione in abbinamento all'immancabile Arduino, con sketch per la gestione dell'integrato D7S in modo trasparente (l'apposita libreria solleva dalla gestione dei registri dell'integrato). ■



per il MATERIALE

La breakout con sensore di terremoti (cod. BREAKOUT019) viene fornita montata e collaudata ed è disponibile presso Futura Elettronica al prezzo di Euro 48,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:
 Futura Elettronica, Via Adige 11,
 21013 Gallarate (VA)
 Tel: 0331-799775
<http://www.futurashop.it>

Realizza facilmente un robot e controllalo con Arduino e Raspberry Pi!

Montaggio semplice e veloce!



€ 59,00

Cod. **ALPHABOT**

(PER IL MODELLO ALPHABOT LE BOARD RASPBERRY PI E ARDUINO NON SONO COMPRESI)



BOARD UNO COMPRESA



€ 79,00

Cod. **ALPHABOT2**

AlphaBot e AlphaBot2 piattaforme robotiche - in kit

AlphaBot e **AlphaBot2** sono piattaforme di sviluppo robotiche controllate da Raspberry Pi e Arduino (ALPHABOT) e Arduino (ALPHABOT2). I robot possono essere comandati tramite telecomando (incluso) o smartphone (iOS o Android). Sono in grado di evitare ostacoli ed inseguire una linea, ma grazie agli accessori opzionali è possibile aumentarne le funzioni e le prestazioni.

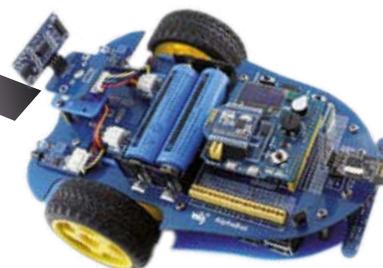
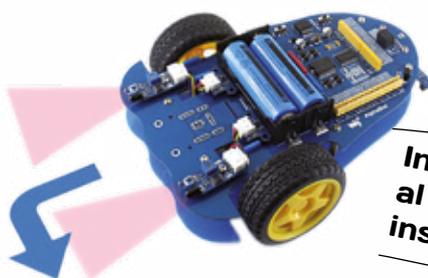
Evitano gli ostacoli grazie ai sensori IR



CONTROLLABILI CON TELECOMANDO O SMARTPHONE



Inseguono una linea grazie al modulo sensore per inseguimento linea



Prezzi IVA inclusa

FUTURA ELETTRONICA®

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Caratteristiche tecniche e vendita on-line su:

www.futurashop.it