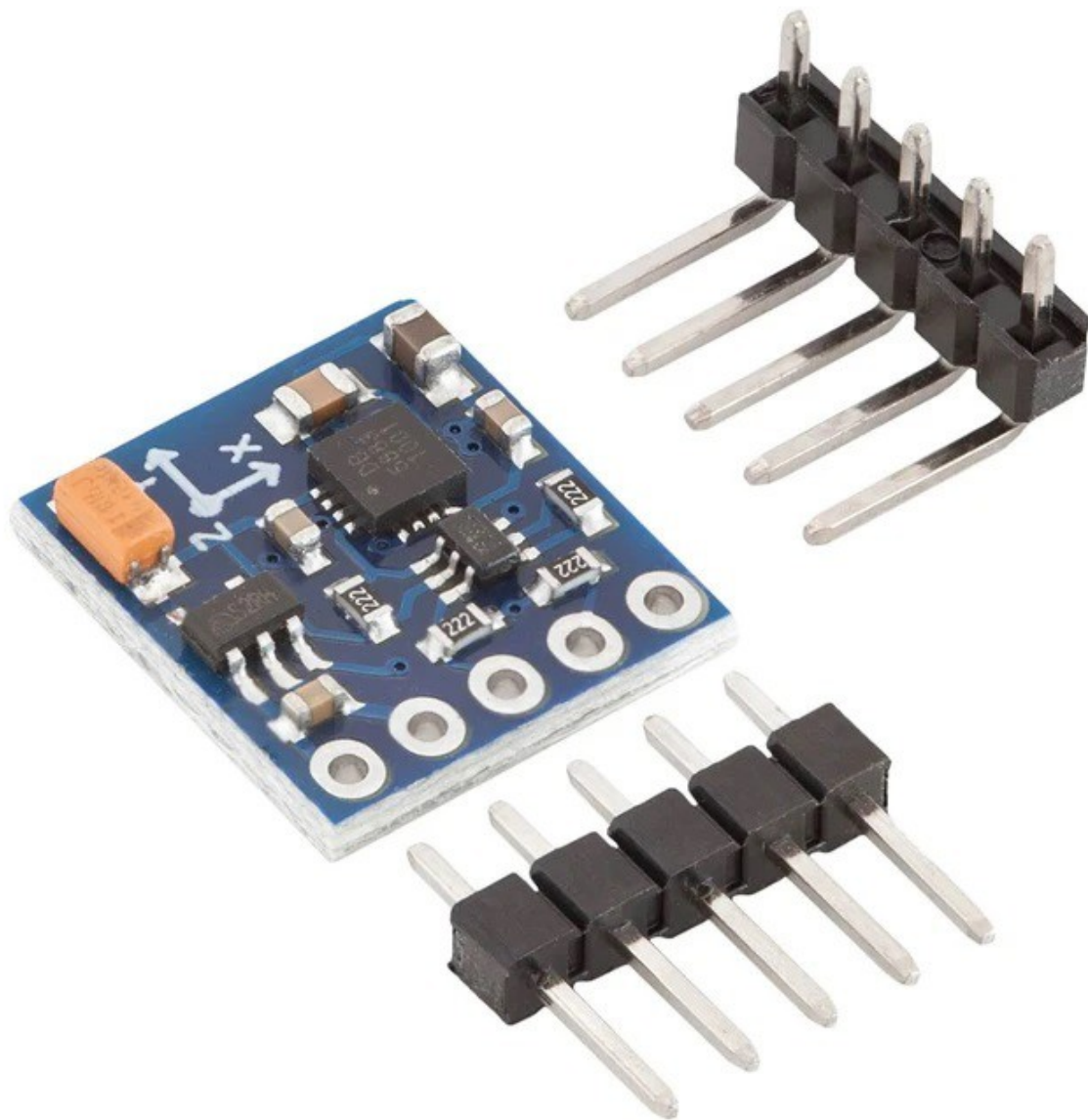


AZ-Delivery

Benvenuto!

Grazie per aver acquistato il nostro *Modulo Bussola Magnetica GY-271 AZ-Delivery*. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!





Indice dei Contenuti

Introduzione.....	3
Specifiche.....	5
Modalità di funzionamento.....	7
La piedinatura.....	10
Come configurare l'Arduino IDE.....	11
Come configurare il Raspberry Pi e il Python.....	15
Collegamento del sensore con Uno.....	16
Libreria per Arduino IDE.....	17
Esempio di sketch.....	18
Collegamento del sensore con Raspberry Pi.....	23
Abilitazione dell'interfaccia I2C.....	24
Librerie e strumenti per Python.....	25
Script Python.....	27
Calcolo della declinazione.....	31



Introduzione

Il cuore di un modulo GY-271 è il chip *QMC5883L*. Il modulo è destinato ad applicazioni di alta precisione come la bussola, la navigazione e il gioco in droni, robot, dispositivi portatili mobili e personali. Il chip è in realtà un circuito sensoriale multichip, magnetoresistivo, composto da sensori triassiali e application specific integrated circuits (ASIC) per misurare il campo magnetico. Si basa su una tecnologia magnetoresistiva ad alta risoluzione con licenza *Honeywell AMR*. Insieme al convertitore analogico-digitale (ADC) a 16 bit progettato su misura, offre i vantaggi del basso rumore, dell'alta precisione, del basso consumo energetico, della cancellazione dell'offset e della compensazione della temperatura. Il sensore consente una precisione di puntamento della bussola da 1° a 2°.

Con un'alimentazione DC applicata ai morsetti del sensore, il sensore converte qualsiasi campo magnetico incidente nelle direzioni sensibili dell'asse in un'uscita di tensione differenziale. L'ASIC amplifica ed elabora il segnale per avere un'uscita digitale. Il dispositivo ha una funzione di cancellazione degli offset per eliminare gli offset del sensore e dell'ASIC. Inoltre applica un campo magnetico auto-allineato per ripristinare lo stato magnetico prima di ogni misurazione per garantire un'elevata precisione. Grazie a queste caratteristiche, il *QMC5883L* non necessita di calibrazione ad ogni esecuzione nella maggior parte delle situazioni applicative. Potrebbe essere necessario calibrarlo una volta in un nuovo sistema o quando il sistema passa a una nuova batteria.

Az-Delivery

Questo dispositivo è collegato ad un'interfaccia seriale, ma come dispositivo slave sotto il controllo di un dispositivo master, come il microcontrollore. Il controllo di questo dispositivo avviene tramite l'interfaccia I2C. Questo dispositivo supporta i modalità di velocità standard e veloce, rispettivamente *100kHz* e *400kHz*. Per supportare tutte queste modalità sono necessarie resistenze esterne di pull-up.

Specifiche

- » Campo di tensione di esercizio: da 3.3V a 5V DC
- » Basso Consumo Energetico: 75 μ A
- » Interfaccia di comunicazione: I2C (modalità standard e veloce)
- » Indirizzo I2C di default: 0x0D
- » Precisione della bussola: da 1° a 2°
- » Campo di temperatura di esercizio: da -40°C a +85°C
- » Campo Magnetico: ± 8 Gauss
- » Risoluzione da analogico a digitale: 16-bit
- » ADC con sensori AMR a basso rumore
- » Uscita temperatura
- » Uscita Dati a Temperatura Compensata
- » Dimensioni: 13 x 14 x 3mm [0.51 x 0.55 x 0.12in]

Il dispositivo è dotato di un orologio interno per le funzioni logiche digitali interne e la gestione dei tempi. Questo orologio non è disponibile per uso esterno.

Sono necessari alcuni periodi di tempo affinché il dispositivo sia completamente funzionante dopo l'accensione. L'alimentazione esterna richiede un periodo di tempo per l'aumento della tensione, che è tipicamente di 50 millisecondi. Tuttavia, non è controllata dal dispositivo. Il periodo di tempo del Power On Reset (POR) comprende il tempo per resettare tutte le logiche, caricare i valori nella memoria non volatile su appositi registri, entrare in modalità stand-by e prepararsi per le misurazioni (circa 450 microsecondi).

Az-Delivery

Quando il dispositivo è acceso, tutti i registri vengono resettati dal POR, quindi il dispositivo passa alla modalità stand-by e attende ulteriori comandi.

Il chip *QMC5883L* ha anche un sensore di temperatura integrato che può fornire la lettura della temperatura per altre applicazioni.

La compensazione della temperatura dei dati magnetici misurati è abilitata di default al fattore. La temperatura misurata dal sensore di temperatura integrato sarà utilizzata per compensare le variazioni di sensibilità magnetica del sensore dovute alla variazione di temperatura. I dati del sensore magnetico compensati vengono inseriti automaticamente nei registri dei dati di uscita.



Modalità di funzionamento

Modalità di Misurazione Continua

Durante la modalità di misurazione in continuo, il sensore magnetico effettua continuamente le misurazioni e inserisce i dati misurati nei registri di uscita dei dati. Nella modalità di misurazione in continuo, i dati del sensore magnetico vengono compensati automaticamente per gli effetti di offset e temperatura. I guadagni sono calibrati in fabbrica.

Sequenza di Lettura Normale

La lettura normale dei dati del magnetometro può essere effettuata come segue:

- » Controllare il pin *DRDY* (o sondando *DRDY* bit nel registro *06H*)
- » Leggere *DRDY* bit nel registro *06H* (per il sondaggio non è necessario)
 - DRDY*: Dati pronti ("1"), o no ("0").
 - DOR*: Sono mancati alcuni o tutti i dati ("1"), o no ("0")
- » Leggere i dati misurati; se si accede a uno dei sei registri di dati, *DRDY* e *DOR* passano a "0".

Se si accede a uno dei sei registri di dati, inizia la protezione dei dati. Durante il periodo di protezione dei dati, il registro dei dati non può essere aggiornato fino a quando non sono stati letti gli ultimi bit del registro *05H*.

Az-Delivery

Durante la misurazione è possibile leggere il registro dei dati che conserva i dati precedentemente misurati. Pertanto, non verrà impostato nessun interrupt (*DRDY* bit) se la lettura dei dati avviene a metà della misura.

Se i dati *N*-th vengono saltati, i dati attuali vengono lavati via dai prossimi dati in arrivo. In questo caso, l'interrupt (bit *DRDY*) si mantiene alto fino alla lettura dei dati. Il bit *DOR* è impostato su "1", che indica la perdita di una serie di dati di misura. Esso (bit *DOR*) passerà a "0" una volta che si accede al registro 06H nella successiva operazione di lettura dei dati.

Il blocco dei dati si attiva quando si accede a uno qualsiasi dei registri di dati. Se il registro 05H (sblocco dati) non è accessibile fino alla fine delle prossime misurazioni, i dati correnti bloccano i dati successivi per aggiornare il registro dati. In questo caso, anche il bit *DOR* è impostato su "1" fino a quando non si accede al registro 06H nella successiva lettura dei dati.

Ogni volta che si misura il campo magnetico viene generato un interrupt sul pin *DRDY*. L'interrupt può essere disabilitato impostando: $0AH[0] = 1$.

Modalità stand-by

La modalità stand-by è uno stato di default del chip, al Reset all'Accensione (POR) o al reset soft. In questa modalità vengono attivati solo alcuni blocchi funzione che mantengono il consumo di energia il più basso possibile. In questo stato, i valori del registro sono tenuti ON da una potenza ultra-bassa. L'interfaccia I2C può essere svegliata leggendo i valori del registro che sono mantenuti ON o scrivendo su qualsiasi registro. Non si verifica alcuna misurazione magnetometrica in modalità stand-by. Anche l'orologio interno viene tenuto fermo.

Esempio di misurazione

- » Controlla il registro di stato $06H[0]$; "1" significa pronto.
- » Leggi i registri dei dati da $00H$ tramite $05H$.

Esempio di Standby

- » Scrivi il valore $0x00$ nel registro $09H$.

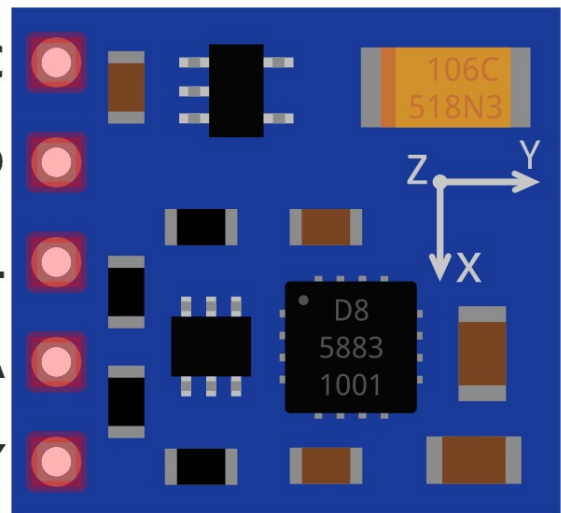
Esempio di Reset Soft

- » Scrivi il valore $0x80$ nel registro $0AH$.

La piedinatura

Il modulo bussola magnetico GY-271 ha cinque pin. La piedinatura è mostrata nell'immagine seguente:

Power supply - VCC
Ground - GND
I2C Serial clock line - SCL
I2C Serial data line - SDA
Data Ready pin - DRDY



L'alimentazione e i pin logici funzionano su tensioni in un range da 3,3V fino a 5V.

Il pin *DRDY* viene utilizzato per indicare che la misurazione è terminata. Si può usare come interrupt, come segnale per iniziare a leggere i dati dal sensore.

Come configurare l'Arduino IDE

Se non hai già installato Arduino IDE, vai al link:

<https://www.arduino.cc/en/Main/Software>

e scarica il file di installazione del sistema operativo scelto.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: **ARDUINO 1.8.9**, followed by a paragraph describing the IDE as open-source software that runs on Windows, Mac OS X, and Linux. Below this, it states that the software can be used with any Arduino board and refers to the 'Getting Started' page for installation instructions. On the right side of the page, there is a teal sidebar with links for different operating systems: **Windows** (Installer for Windows XP and up, and ZIP file for non admin install), **Windows app** (Requires Win 8.1 or 10, with a 'Get' button), **Mac OS X** (10.8 Mountain Lion or newer), and **Linux** (32 bits, 64 bits, ARM 32 bits, ARM 64 bits). At the bottom of the sidebar, there are links for Release Notes, Source Code, and Checksums (sha512).

Per gli utenti *Windows*, fare doppio clic sul file .exe scaricato e seguire le istruzioni nella finestra di installazione..

Az-Delivery

Per gli utenti Linux, scaricare un file con estensione *.tar.xz*, che è necessario estrarre. Quando lo si estrae, andare nella directory estratta, e aprire il terminale in quella directory. È necessario eseguire due script *.sh*, il primo chiamato *arduino-linux-setup.sh* e il secondo chiamato *install.sh*.

Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:

sh arduino-linux-setup.sh user_name

user_name - è il nome di un superutente nel sistema operativo Linux. All'avvio del comando deve essere inserita una password per il superutente. Aspettate qualche minuto che lo script completi tutto.

Dopo l'installazione del primo script, deve essere utilizzato il secondo script, chiamato *install.sh*. Nel terminale (directory estratta), eseguire il seguente comando: **sh install.sh**

Dopo l'installazione di questi script, andare su *Tutte le App*, dove troverai l'*Arduino IDE* installato.

Az-Delivery

Quasi tutti i sistemi operativi sono dotati di un editor di testo preinstallato (ad esempio Windows viene fornito con Notepad, Linux Ubuntu viene fornito con Gedit, Linux Raspbian viene fornito con Leafpad, ecc.). Tutti questi editor di testo sono perfettamente adatti allo scopo dell'eBook.

La prossima cosa da fare è controllare se il PC è in grado di rilevare la scheda Arduino. Aprite l'Arduino IDE appena installato e andate su:

Strumenti > Scheda > {your board name here}

{your board name here} dovrebbe essere l' *Arduino/Genuino Uno*, come si può vedere nella seguente immagine:

Deve essere selezionata la porta alla quale è collegata la scheda Arduino. Vai su: *Strumenti > Porta > {port name goes here}*

e quando la scheda Arduino è collegata alla porta USB, il nome della porta è visibile nel menu a tendina dell'immagine precedente.



Se si utilizza l'Arduino IDE su Windows, i nomi delle porte sono i seguenti:

Per gli utenti Linux, il nome della porta è `/dev/ttyUSBx` per esempio, dove `x` rappresenta un numero intero compreso tra 0 e 9.



Come configurare il Raspberry Pi e il Python

il sistema operativo, tutto deve essere impostato in modo da poter essere utilizzato in modalità Headless. La modalità Headless consente la connessione remota al Raspberry Pi, senza la necessità di uno schermo di un PC, mouse o tastiera. Le uniche cose di cui avete bisogno per questa modalità sono il Raspberry Pi, l'alimentazione e la connessione internet. Tutto questo è spiegato in dettaglio nell'eBook gratuito:

Raspberry Pi Quick Startup Guide

che si trova al seguente link:

<https://www.az-delivery.de/products/raspberry-pi-kostenfreies-e-book?ls=en>

Il sistema operativo Raspbian viene fornito con il Python preinstallato.

Collegamento del sensore con Uno

Collegare il modulo bussola magnetico GY-271 con Uno come indicato nel seguente schema di collegamento:

Pin sensore > **Pin Uno**

VCC > 5V

Filo rosso

GND > GND

Filo nero

SCL > A5 [SCL]

Filo blu

SDA > A4 [SDA]

Filo verde

DRDY > non collegato

Libreria per Arduino IDE

Per poter utilizzare il sensore con Uno, si raccomanda di scaricare e installare una libreria esterna per esso. La libreria utilizzata in questo eBook si chiama QMC5883L. Per scaricarla e installarla, aprire l'Arduino IDE e andare su:

Strumenti > Gestione Librerie

Quando si apre una nuova finestra, digitare qmc nella casella di ricerca e installare la libreria QMC5883LCompass realizzata da MRPrograms, come mostrato nella seguente immagine:

Diversi esempi di sketch vengono forniti con la libreria. Il codice di diversi esempi di sketch viene modificato per creare l'esempio di sketch nel prossimo capitolo per mostrare come utilizzare tutte le funzioni che vengono fornite con la libreria.

Az-Delivery

Esempio di sketch

```
#include <QMC5883LCompass.h>
QMC5883LCompass compass;
void setup() {
    Serial.begin(9600);
    compass.init();
}
void loop() {
    compass.read(); // Read compass values
    byte a = compass.getAzimuth();
    // Return Azimuth reading
    Serial.print("Azimuth: ");
    Serial.println(a);
    byte d = compass.getBearing(a);
    // Output is a value from 0 - 15
    // based on the direction of the bearing / azimuth
    Serial.print("Direction: ");
    Serial.println(d);
    char compassLetters[3];
    compass.getDirection(compassLetters, a);
    Serial.print(compassLetters[0]);
    Serial.print(compassLetters[1]);
    Serial.println(compassLetters[2]);
    delay(1000);
}
```

Az-Delivery

Caricare lo sketch su Uno e aprire il Monitor Seriale (Strumenti > Monitor Seriale). Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:

Per ottenere valori simili, ruotare la scheda della bussola attorno ad uno dei suoi assi.

Az-Delivery

Lo sketch inizia con l'importazione di una libreria QMC5883LCompass.

Successivamente, viene creato un oggetto chiamato compass, che rappresenta il modulo stesso. Questo oggetto viene utilizzato per inizializzare e leggere i dati del modulo.

Nella funzione setup() la comunicazione seriale viene avviata con un baud rate di 9600bps. Inoltre, l'oggetto compass viene inizializzato con la seguente riga di codice: `compass.init();`

All'inizio della funzione loop() vengono letti i dati dal modulo con la seguente riga di codice: `compass.read()`

Successivamente, le letture vengono convertite in valori Azimut (ciò che i valori Azimut sono, non è contemplato in questo eBook) con la seguente riga di codice:
`byte a = compass.getAzimuth();`

La funzione chiamata getAzimuth() non ha argomenti e restituisce un valore. Il valore restituito è un numero intero che rappresenta il valore Azimuth.

Dopo di che, i dati con il valore Azimuth vengono visualizzati nel Monitor Seriale.

Az-Delivery

Quindi, i valori di Azimuth vengono convertiti in numeri di direzione e visualizzati nel Monitor Seriale.

Ci sono otto direzioni sulla bussola:

- » Nord
- » Nord-Est
- » Est
- » Sud-Est
- » Sud
- » Sud-Ovest
- » Ovest
- » Nord-Ovest

Come si può vedere sul cerchio della bussola, ci sono 16 diverse aree, come nell'immagine seguente:

Az-Delivery

La funzione chiamata `getBearing()` ha un solo argomento e restituisce un valore. L'argomento è un numero intero, che rappresenta il valore Azimuth. Anche il valore restituito è un numero intero, nell'intervallo da 0 a 15, che rappresenta uno dei numeri di direzione dell'immagine del cerchio della bussola.

Dopo di che i numeri di direzione vengono convertiti in lettere di direzione con le seguenti righe di codice:

```
char compassLetters[3];  
compass.getDirection(compassLetters, a);
```

Dove “a” è il valore Azimuth. Qui viene creato un array char con tre elementi. Poi si usa la funzione `getDirection()` per convertire il valore Azimuth nella lettera di direzione. La funzione ha due argomenti e non restituisce alcun valore. Il primo argomento è l'array char, in cui la funzione memorizza le lettere di direzione, e il secondo argomento è il valore Azimuth.

Alla fine della funzione `loop()` vengono visualizzati i dati delle lettere di direzione e con la funzione `delay()` si crea una pausa di 250 millisecondi tra due loop della funzione `loop()`.

Collegamento del sensore con Raspberry Pi

Collegare il modulo magnetico della bussola GY-271 con il Raspberry Pi come indicato nel seguente schema di collegamento:

Pin sensore > Pin Raspberry Pi

VCC	>	3V3	[pin 1]	Filo rosso
GND	>	GND	[pin 9]	Filo nero
SCL	>	GPIO3	[pin 5]	Filo blu
SDA	>	GPIO2	[pin 3]	Filo verde



Abilitazione dell'interfaccia I2C

Per poter utilizzare il sensore con Raspberry Pi, l'interfaccia I2C sul Raspberry Pi deve essere abilitata. Per farlo, vai su:

Menu Applicazione > Preferiti > Configurazione Raspberry Pi

Quando si apre una nuova finestra, cerca la scheda Interfacce. Attivare quindi il radio button I2C e clicca su OK, come mostrato nell'immagine seguente:

Librerie e strumenti per Python

Per poter utilizzare il modulo con il Raspberry Pi, si raccomanda di scaricare e installare una libreria esterna per esso. La libreria utilizzata in questo eBook si chiama gy271compass. Prima di essere utilizzata, devono essere installate diverse librerie e strumenti.

Installare innanzitutto il git, python-smbus and i2c-tools. Per fare ciò aprire il terminale ed eseguire i seguenti comandi, uno per uno:

```
sudo apt-get update
```

```
sudo apt-get install -y git python3-smbus i2c-tools
```

Poi, controllare l'indirizzo I2C del modulo, eseguendo il seguente comando:
`i2c detect -y 1`

Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:

Dove 0x0d è l'indirizzo I2C del modulo.

Az-Delivery

Se l'interfaccia I2C del Raspberry Pi non è abilitata, e il comando precedente viene eseguito, viene generato il seguente errore:

Per scaricare la libreria esterna, salvare lo script `gy271compass.py` dal seguente link:
<https://github.com/Slaveche90/gy271compass>

Oppure utilizzare il git per clonare il repository, eseguendo il seguente comando nel terminale:

`git clone https://github.com/Slaveche90/gy271compass`

Questo comando creerà una directory chiamata `gy271compass`. Aprila e copia lo script `gy271compass.py` nella stessa directory dove viene salvato lo script del prossimo capitolo.

Az-Delivery

Script Python

```
import gy271compass as GY271
from time import sleep

sensor = GY271.compass(address=0x0d)

print('[Press CTRL + C to end the script!]\n')
try:
    while True:
        angle = sensor.get_bearing()
        temp = sensor.read_temp()

        print('Heading Angle = {}°'.format(angle))
        print('Temperature = {:.1f}°C'.format(temp))
        sleep(1)

except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Salvare lo script con il nome `gy271.py` nella stessa directory in cui è stato salvato lo script `gy271compass`. Per eseguire lo script `gy271.py` aprire il terminale dello script `gy271.py` nella directory dove è stato salvato lo script ed eseguire il seguente comando: `python3 gy271.py`

Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:

Per fermare lo script premere CTRL + C sulla tastiera.

Az-Delivery

Lo script inizia con l'importazione della libreria di classe esterna `gy271compass` e della libreria dell'orario.

Successivamente, l'oggetto chiamato `compass` viene creato utilizzando la seguente riga di codice:

```
sensor = GY271.compass(address=0x0d)
```

dove `0x0d` è l'indirizzo I2C del modulo.

Poi si crea il blocco `try-except`. Nel blocco `try` del codice si crea un loop indefinito (`while True:`). Per leggere l'angolo di direzione nel loop indefinito si utilizza la seguente riga di codice:

```
angle = sensor.get_bearing()
```

e per leggere i dati di temperatura, si utilizza la seguente riga di codice:

```
temp = sensor.read_temp()
```

I dati dell'angolo di puntamento sono memorizzati nella variabile chiamata `angle` e i dati di temperatura sono memorizzati nella variabile chiamata `temp`. Queste due variabili vengono poi utilizzate per visualizzare i dati nel terminale.

Quando si preme il `CTRL + C` sulla tastiera, avviene un interrupt della tastiera. Nell'interrupt della tastiera viene eseguito il blocco di codice `except` e un messaggio `Script end!` viene visualizzato.

Az-Delivery

Il modulo ha diverse opzioni che possono essere impostate alla creazione dell'oggetto sensore. È possibile passare i seguenti argomenti al costruttore `compass()`:

- | | |
|-----------------------------|---|
| <code>address=0x0d</code> | - the I2C address of the module |
| <code>mode=MODE_CONT</code> | - Continuous or stand-by mode:
<i>MODE_CONT</i> or <i>MODE_STBY</i> |
| <code>odr=ODR_10Hz</code> | - Output Data Rate, with values:
<i>ODR_10Hz</i> ,
<i>ODR_50Hz</i> ,
<i>ODR_100Hz</i> or
<i>ODR_200Hz</i> |
| <code>sens=SENS_2G</code> | - Sensitivity: <i>SENS_2G</i> or <i>SENS_8G</i> |
| <code>osr=OSR_512</code> | - OverSampling Rate, with values:
<i>OSR_512</i> ,
<i>OSR_256</i> ,
<i>OSR_128</i> or
<i>OSR_64</i> |
| <code>d=0.0</code> | - Dati di declinazione (spiegati nel prossimo capitolo) |

Utilizzo (queste sono le impostazioni predefinite):

```
sensor = GY271.compass(address=0x0d, mode=MODE_CONT,  
                        odr=ODR_10Hz, sens=SENS_2G, osr=OSR_512, d=0.0)
```



Calcolo della declinazione

Per calcolare il valore di declinazione magnetica, andare al seguente link:

<https://www.magnetic-declination.com/YUGOSLAVIA/NEOPLANTA/2970848.html>

e trovare la propria posizione. Successivamente, cliccare con il mouse del PC sulla posizione e verrà visualizzato un pop-up. Copiare i seguenti dati:

E utilizzare l'angolo minuti al convertitore decimale, per esempio:

<https://www.rapidtables.com/convert/number/degrees-minutes-seconds-to-degrees.html>

per ottenere il valore del punto flottante dell'angolo.

In questo esempio: +3 gradi e 55 minuti = $+3^{\circ} 55'$ = $+3.916667^{\circ}$

`sensor = GY271.compass(address=0x0d, d=3.916667)`

NOTA: Non dimenticare il segno, perché la declinazione magnetica può essere + o -.

Ce l'hai fatta!

Ora puoi usare il tuo modulo per i tuoi progetti.

AZ-Delivery

E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet..

Se stai cercando dei prodotti di alta qualità per il tuo Arduino e Raspberry Pi, AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.

<https://az-delivery.de>

Buon divertimento!

Impressum0

<https://az-delivery.de/pages/about-us>