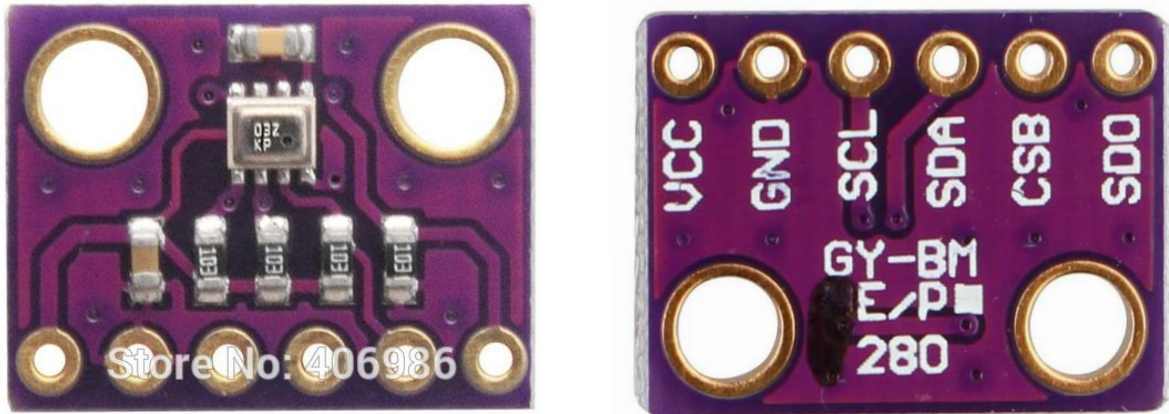


47- sensore di pressione GY-BM E/P 280 (BMP 280) GY-BM E/P 280 barometric sensor (some notes at end of this section)



Questo sensore consente di rilevare sia la temperatura che la pressione atmosferica (e da quest'ultima l'altitudine). Sembra che sia anche in grado di fornire indicazioni sul tasso di umidità, ma non sono state individuate librerie in grado di fornire questo dato. Le ridottissime misure di questo sensore (il sensore e' il microchip saldato sulla basetta) ne fanno un componente facilmente inseribile in apparecchi portatili, come orologi o telefoni.

Il sensore viene venduto come un prodotto di notevole precisione, ma quello utilizzato in questo progetto si e' dimostrato piuttosto impreciso (forse per una inadeguata taratura di fabbrica). La temperatura rilevata e' infatti di circa 0,8 gradi superiore alla temperatura effettiva mentre la pressione e' di circa 17 hPa inferiore.

Questa imprecisione e' stata "corretta" a programma, tramite alcune istruzioni che hanno aumentato di 17 hPa la pressione e tolto 0,8 gradi alla temperatura. In questo progetto il sensore barometrico e' stato utilizzato per costruire un altimetro "trasportabile" dotato di batterie, display e tasti per il settaggio.

Poiche' l'altezza calcolata in base alla pressione atmosferica risente pesantemente della situazione meteorologica del momento, il dispositivo e' dotato di due tasti mediante i quali e' possibile impostare l'altezza del punto di partenza (che ovviamente deve essere nota) in modo da poter poi raggiungere il punto di arrivo e calcolarne l'altezza, confidando che nel frattempo non siano variate le condizioni meteo.



[Qui](#) il filmato di questo progetto.

Prima di procedere alla compilazione del programma devono essere installate, se non gia' presenti, le seguenti librerie:

- LiquidCrystal_I2C.h <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>
- BMP280.h <http://www.marcelplace.nl/download/2011/bmp280.zip>

Per installare le librerie e' necessario seguire la procedura illustrata nei precedenti progetti, e sintetizzabile in:

Arduino: BMP280 - sensore di pressione – barometric sensor

- download della libreria in formato complesso
- installazione della libreria andando in IDE-> sketch-> includes Library-> add .zip library
- verifica di avvenuta installazione (andando in IDE-> sketch-> includes Library-> Contributed library)

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



This sensor can detect both the temperature and the atmospheric pressure (and by the latter, the altitude). It seems that is also able to provide indications on moisture, but we did not find the right library for this data. The small size of this sensor (the sensor is the microchip soldered on board) make it a component useful in to portable devices, such as watches or phones.

The sensor is sold as a high precision product, but the one used in this project is rather inaccurate (perhaps because of inadequate factory setting). The detected temperature is, in fact, about 0.8 degrees higher than the actual temperature while the pressure is about 17 hPa lower.

This imprecision has been "corrected" by program, through some instructions that increased pressure by 17 hPa and removed 0.8 temperature degrees.

In this project, the barometric sensor was used to build a "portable" altimeter, with batteries, display and setting keys.

Since the elevation calculated starting from atmospheric pressure is heavily influenced by weather conditions, the device is equipped with two keys by which you can set the starting point elevation (which of course must be known) so it can then calculate the end elevation point, hoping that, in meantime, the weather is not changed. [Here](#) the movie of this project.

Before proceeding to program compilation must be installed, if not already done, the libraries:

- LiquidCrystal_I2C.h <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>
- BMP280.h <http://www.marcelplace.nl/download/2011/bmp280.zip>

For installation, see process shown in previous exercises, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

Before proceeding to program compilation must be installed, if not already done, the libraries:

- LiquidCrystal_I2C.h found [here](#)

For library installation, see process shown in previous examples, and summarized in:

- library download in compressed form;

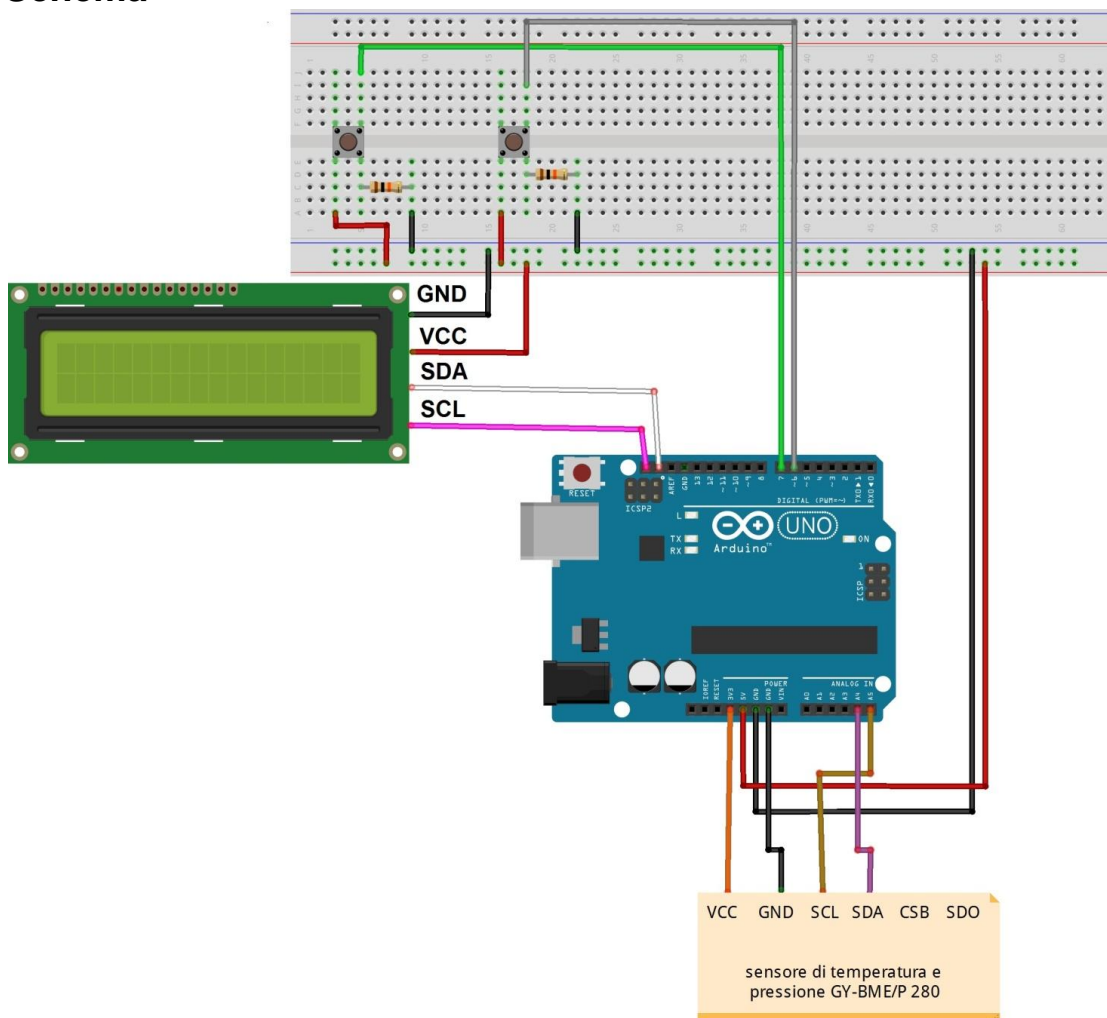
Arduino: BMP280 - sensore di pressione – barometric sensor

- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

Materiali

- Un sensore GY BM E/P 280
- Un display a cristalli liquidi del tipo 2004
- Due pulsanti
- Una breadboard (anche piccola)
- Due resistenze da 220 ohm

Schema



fritzing

Programma

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo.
 * Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T.
 *
 * Prima di compilare il programma e' necessario installare (vedi le note sulla scheda tecnica):
 * - la libreria BMP280.h, reperibile qui: http://www.marcelsplice.nl/download/2011/bmp280.zip
 * - la libreria NewLiquidCrystal : https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
 *
 * I risultati del sensore utilizzato in questo esercizio sono alquanto imprecisi (probabilmente il
 * sensore utilizzato nel test non e' stato correttamente tarato in fabbrica) per cui il programma
 * aggiunge alla temperatura ed alla pressione delle costanti, al fine di rendere i due valori per
 * quanto possibile vicini alla realta'
 *
```

Arduino: BMP280 - sensore di pressione – barometric sensor

```
* Piu' in dettaglio, sono stati aggiunti:
* -0,8 gradi alla temperatura
* 17 hpa alla pressione atmosferica
*
* Per compensare la variazioni di pressione dovute alla situazione meteo del momento, il programma
* prevede nei primi 5 secondi di funzionamento, l'uso di due pulsanti per modificare
* l'altezza iniziale. Tale modifica viene quindi memorizzata e riportata poi su ogni successiva
* misurazione.
*
* Allo scopo di ridurre il consumo di energia, l'illuminazione si spegne dopo 15 secondi
* dall'inizio delle misurazioni. Per riattivarla premere il pulsante collegato al pin 8
*
* -----
* Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
*
* Before compiling program must be installed (see also notes on progrma data sheet):
* - The library BMP280.h, available here: http://www.marcelsplice.nl/download/2011/bmp280.zip
* - The library NewLiquidCrystal: https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
*
* sensor used in this project is somewhat imprecise (probably has not been properly calibrated at
* the factory), so program adds to temperature and pressure some constants, in order to make the
* two values as much as possible accurate.
*
* in detail, have been added:
* -0.8 degrees to temperature
* 17 hPa to atmospheric pressure
*
* To compensate pressure changes due to the current weather situation, the program provides, in the
* first 5 seconds of operation, the possibility of acting on two buttons to change the initial
* altitude. This change is then stored and reproduced for each subsequent measurement.
*
* In order to reduce power consumption, the display backlight turns off after 15 seconds from the
* beginning of the measurements. To reactivate it press button connected to pin 8
* -----
*/
#include "BMP280.h"
#define P0 1013.25 // pressione standard, a livello del mare
#define su 7 // il pulsante "su" e' collegato alla porta 7
#define giu 6 // il pulsante "giu" e' collegato alla porta 6
BMP280 bmp;
#include <Wire.h> // libreria wire presente, di fault, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
//. . . . . addr, en, rw, rs, d4, d5, d6, d7, bl, blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce la tipologia del
display
double T = 0; // variabile per la memorizzazione della temperatura
double P = 0; // variabile per la memorizzazione della pressione
double A = 0; // variabile per la memorizzazione dell'altezza
char risultatomisura = 0; // variabile di stato della misurazione di pressione e temperatura
int statusu = 0; // variabile per la memorizzazione dello stato del pulsante su
int statogiu = 0; // variabile per la memorizzazione dello stato del pulsante giu
int correttore = 0; // variabile per la memorizzazione del correttore di altezza a compensazione
delle condizioni meteo
long tempinizio = 0; // variabile di memorizzazione del momento di inizio del conteggio (in
millisecondi)
long tempcorrente = 0; // variabile di memorizzazioen del momento corrente (in millisecondi)
int alt = 0; // variabile di memorizzazione dell'altezza, senza decimali
//
// ***** routine di verifica della preessione ed esposizione dei dati *****
// ***** checks temperature and pressure, then shows data*****
//
void verificaedesponi (void)
{
    risultatomisura = bmp.startMeasurement();
    if (risultatomisura != 0)
    {
        delay(risultatomisura);
        risultatomisura = bmp.getTemperatureAndPressure(T, P);
        if (risultatomisura != 0)
        {
            A = bmp.altitude(P, P0); //calcola l'altezza slm
            alt = A + correttore; // modifica altezza per compensare le influenze meteo
            P = P + 17; // compensa errore del sensore - sensor error compensation
            T = T - 0.8; // compensa errore del sensore - sensor error compensation
            lcd.clear (); // pulisce lo schermo
            lcd.print ("temp. ");
            lcd.print (T);
            lcd.print (" c");
        }
    }
}
```

Arduino: BMP280 - sensore di pressione – barometric sensor

```
        lcd.setCursor (0, 1);
        lcd.print ("altezza: ");
        lcd.print (alt);
        lcd.print (" m. slm");
        lcd.setCursor (0, 2);
        lcd.print ("press.: ");
        lcd.print (P);
        lcd.print (" hPa");
        lcd.setCursor (0, 3);
    }
    else
    {
        lcd.clear ();
        lcd.print("Errore");
    }
}
else
{
    lcd.clear ();
    lcd.print ("Errore");
}
}
//
//
void setup()
{
    Serial.begin(9600);
    pinMode (su, INPUT); // porta collegata al pulsante accendispegni
    pinMode (giu, INPUT); // porta collegata al pulsante canalesu
    lcd.begin(20, 4); // inizializza il display e accende l'illuminazione dello sfondo
    lcd.clear ();
    if (!bmp.begin()) // verifica se il barometro e' operativo
    {
        lcd.print("inizial.fallita!"); // initialization fail
        lcd.print("verifica collegam."); // check wiring
        while (1);
    }
    else lcd.print("inizializzazione ok"); // initialization ok
    bmp.setOversampling(4);
    delay (2000);
    verificaedesponi ();
    lcd.print ("5 sec. x comp. alt."); // 5 second to compensate altitude
    tempoinizio = (millis); // memorizza il tempo di inizio compensazione
    tempocorrente = (millis); // memorizza il momento corrente
    while ((tempocorrente - tempoinizio) < 5000)
    {
        statusu = digitalRead (su);
        statogiu = digitalRead (giu);
        if (statusu == HIGH)
        {
            correttore++; // aumenta di 1 metro l'altezza
            tempoinizio = millis(); // resetta tempoinizio, per offrire altri 5 secondi
            delay (200);
        }
        if (statogiu == HIGH)
        {
            correttore--; // diminuisce di 1 metro l'altezza;
            tempoinizio = millis(); // resetta tempoinizio, per offrire altri 5 secondi
            delay (200);
        }
        lcd.setCursor (9, 1);
        alt = A + correttore;
        lcd.print (alt);
        lcd.print ("      ");
        tempocorrente = millis ();
    }
    tempoinizio = millis ();
}
//
//
void loop()
{
    statusu = digitalRead (su);
    if (statusu == HIGH)
    {
        tempoinizio = millis ();
        lcd.backlight(); // accende la luce del display LCD
    }
}
```

Arduino: BMP280 - sensore di pressione – barometric sensor

```
tempocorrente = millis ();
if (tempocorrente - tempoinizio > 15000) // se sono passati piu' di 15 secondi
  lcd.noBacklight(); // spegne l'illuminazione del display
  verificaedesponi (); // lancia la routine di rilevazione ed esposizione dati
  delay (2000); // attende due secondi prima di fare un nuovo controllo
}
```