

Corso Arduino 2016

13 Maggio 2016

Matteo Michele Piazzolla

`matteo.piazzolla@gmail.com`

Lezione 2 - Sensori



POLITECNICO OPEN
unix LABS

Come hack with us.



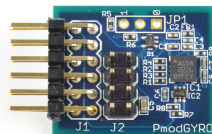
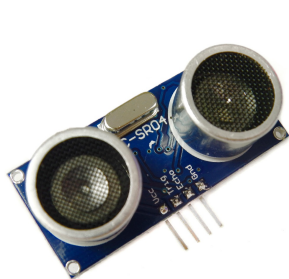
<https://tinyurl.com/poul-arduino>
queste slides si trovano all'indirizzo <https://slides.poul.org>

Il collegamento seriale (attraverso l'USB dell'Arduino) ci permette di creare un canale di comunicazione tra l'Arduino ed il PC

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int a=100;  
  Serial.print(a);  
  Serial.print(" Ciao Mondo\n");  
  Serial.println("Ciao Mondo!");  
}
```

- **Serial.begin(BPS)** Da effettuare nella funzione **setup** per inizializzare l'interfaccia seriale su un certo baud rate (BPS).
Generalmente **9600**
- **Serial.print[ln](da_stampare)** Quando vogliamo stampare qualcosa

I sensori sono gli strumenti che ci permettono di ricevere delle informazioni dal mondo esterno. Esistono tantissimi tipi di sensori per gli usi più disparati.

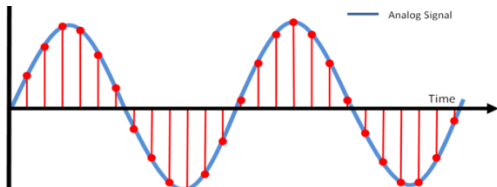


- Sensori di prossimità agli ultrasuoni che possono essere usati per esempio in un robot per riconoscere degli ostacoli
- Sensori di temperatura, di pressione, di luminosità
- Accelerometri, Giroscopi, Magnetometri (IMU)

Possiamo distinguere i sensori in due gruppi principali per la modalità con la quale leggiamo i loro dati

- Analogici
- Digitali

Sensori Analogici



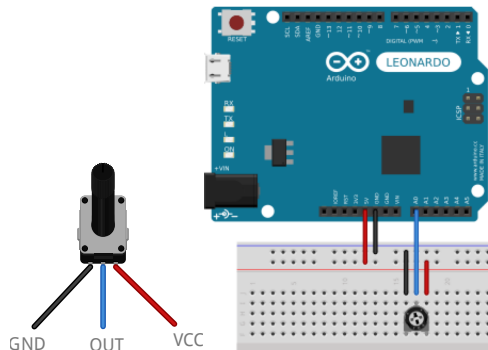
- Leggiamo quello che il sensore misura come **valore di tensione** su uno dei suoi pin
- L'**ADC** (ANALOG DIGITAL CONVERTER) discretizza il segnale continuo fornendoci un valore digitale
- l'Arduino Leonardo ha dentro di sé un ADC ad 10 bit che ci permette di leggere un valore da 0 a 1023 con Full Scale Range 0-5V
- E' necessario usare i **pin analogici** (A0-A5)

Potenziometro e Fotoresistore



- Usiamo un **potenziometro** per simulare un sensore analogico oppure un **fotoresistore**
- Il potenziometro ed il fotoresistore sono delle **resistenze variabili**
- Cambiando il valore della resistenza cambia anche il valore della tensione ai suoi capi

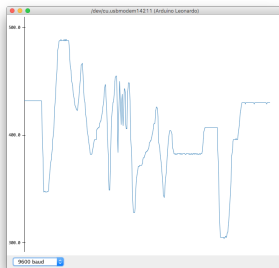
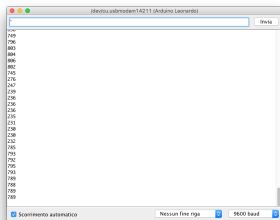
Potenziometro



Collegato in questo modo, facendolo ruotare da un estremo all'altro OUT varierà continuamente tra GND (0V) e VCC (3V3) (ADC [0 , **675**/1023]).


```
int sensorPin = A0;
int sensorValue = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  delay(100);
  Serial.println(sensorValue);
}
```

Monitor seriale



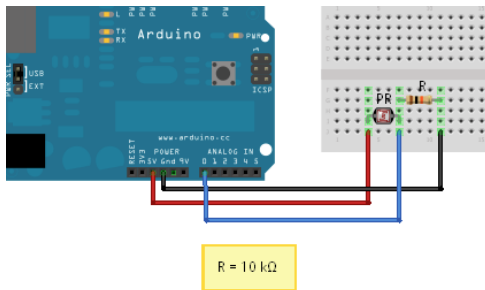
- **Monitor Seriale** stampa il valore numerico del sensore
- **Plotter Seriale** crea un grafico in tempo reale

Attenzione

Impostate nel menù a tendina lo stesso valore della seriale in bsp che avete scritto nel codice

Fotoresistore

Possiamo usare un fotoresistore come sensore di luminosità, infatti non è altro che una resistenza che cambia il suo valore in base alla luce che rileva.



Attenzione

collegate SEMPRE A 3,3 Volt



- Sensori generalmente più complessi e costosi
- I dati vengono richiesti al sensore utilizzando degli **indirizzi**
- Spesso hanno degli interrupt abilitabili (esempio FreeFall per un accelerometro)
- Vari protocolli di comunicazione
- Molte librerie già disponibili per Arduino

11.1 Register Overview

11.1.1 Common

Name	Address	Access	Default	Description
<u>WHO_AM_I</u>	0x20	R	10110100	Identifier of the product family.
<u>SILICON_REV_OTP</u>	0x21	R	00000101	Identifier of the silicon revision.
<u>EXT_STATUS</u>	0x22	R/W	00000000	Temperature data ready and error flags and bank selection.
<u>SYSTEM_STATUS</u>	0x23	R	00000000	Gyroscope and accelerometer data ready and error flags.
<u>GYRO_X_H</u>	0x24	R	00000000	Gyroscope X MSB.
<u>GYRO_X_L</u>	0x25	R	00000000	Gyroscope X LSB.
<u>GYRO_Y_H</u>	0x26	R	00000000	Gyroscope Y MSB.
<u>GYRO_Y_L</u>	0x27	R	00000000	Gyroscope Y LSB.
<u>GYRO_Z_H</u>	0x28	R	00000000	Gyroscope Z MSB.
<u>GYRO_Z_L</u>	0x29	R	00000000	Gyroscope Z LSB.

- Nel **datasheet** di ogni sensore ha una lista di indirizzi utilizzati per richiedere i dati al sensore o per una sua configurazione
- La maggior parte dei sensori ha un registro **WHO_AM_I** che risponde con un valore costante indicato nel datasheet utile per controllare che la comunicazione stia avvenendo col sensore richiesto

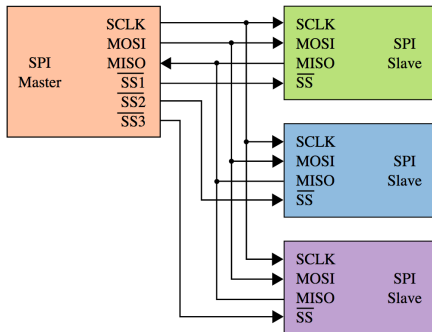
Esistono diversi protocolli di comunicazione utilizzati per dialogare con i sensori o con altre schede.

Voi per comunicare con l'Arduino dal PC state usando la "Seriale" chiamata UART o USART (Universal Synchronous/Asynchronous Receiver/Transmitter)

Altri esempi sono:

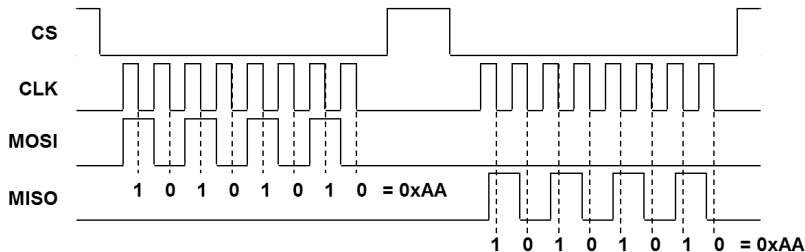
- **SPI** - Serial Peripheral Interface
- **I²C** - Inter Integrated Circuit
- **CAN** - Controlled Area Network (utilizzato tra le centraline delle automobili)
- **1-wire**
- **PCIe** (collegamenti tra le schede dei vostri pc)

Protocollo SPI - Serial Peripheral Interface



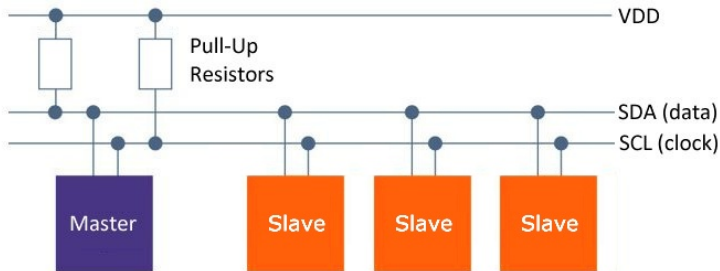
- Un **Master** e diversi **Slave**
- Bus con 4 fili
 - **SCLK** Clock
 - **MOSI** Master Out Slave In
 - **MISO** Master In Slave Out
 - **SS** o **CS** Slave Select o Chip Select per selezionare uno Slave

Protocollo SPI Bus Read



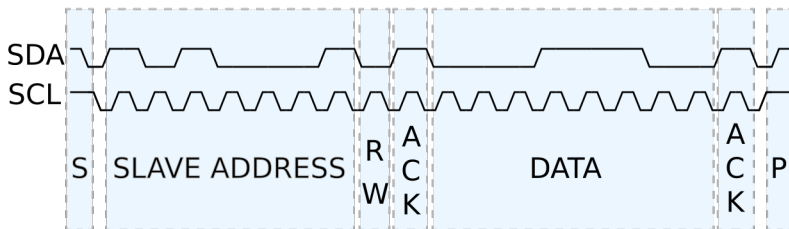
- Si abbassa o alza il Chip Select del sensore da contattare (GPIO LOW/GPIO HIGH)
- Il Master scrive sul Bus un **indirizzo** relativo al dato che vuole leggere dal sensore (nel datasheet del sensore si trova sempre una **mapa dei registri**)
- Lo Slave **risponde** con il dato richiesto

Protocollo I²C - Inter Integrated Circuit



- Un Master e diversi Slave
- Ogni Slave ha un suo **indirizzo hardware**
- Bus con 2 fili
 - **SDA** (Serial DAta) per i dati
 - **SCL** (Serial CLock) per il clock

Protocollo I²C Bus



- Il Master che impone il **clock** mette sul Bus **l'indirizzo del sensore** ed un bit che indica se vuole ricevere o inviare dati al sensore
- Il sensore Slave risponde con un **Ack** e si prepara a ricevere i dati oppure ad inviare quelli richiesti

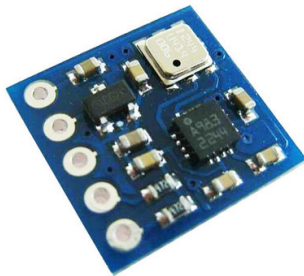
Ok Matteo basta con questi pipponi teorici!



Facci giocare con i sensori!

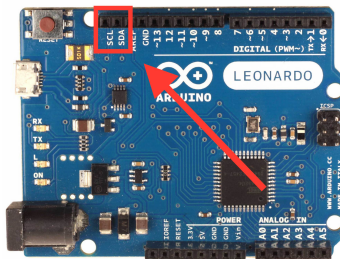
GY-652 Magnetometro e Barometro

GY-652 (BMP180 + HMC5883L) 2 sensori in una sola board



- **HMC5883L** - Magnetometro
- **BMP180** - Barometro
- Protocollo **I²C**

GY-652 - Circuito



- SDA
- SCL
- GND
- 3V3

Attenzione

Ripeto: collegatelo ai 3,3V non ai 5V!!!!

https:

[//slides.poul.org/2016/Corso_Arduino/CorsoArduino2016.zip](https://slides.poul.org/2016/Corso_Arduino/CorsoArduino2016.zip)

- Scaricate ed estraete lo **zip** dal link
- Copiate le cartelle delle librerie in "**Documenti/Arduino/libraries/**" o analogo per il vostro sistema operativo
- Nell'archivio trovate anche i **sorgenti** dei programmi che useremo

Includiamo le librerie necessarie

```
#include <stdio.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_HMC5883_U.h>
Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345);
```

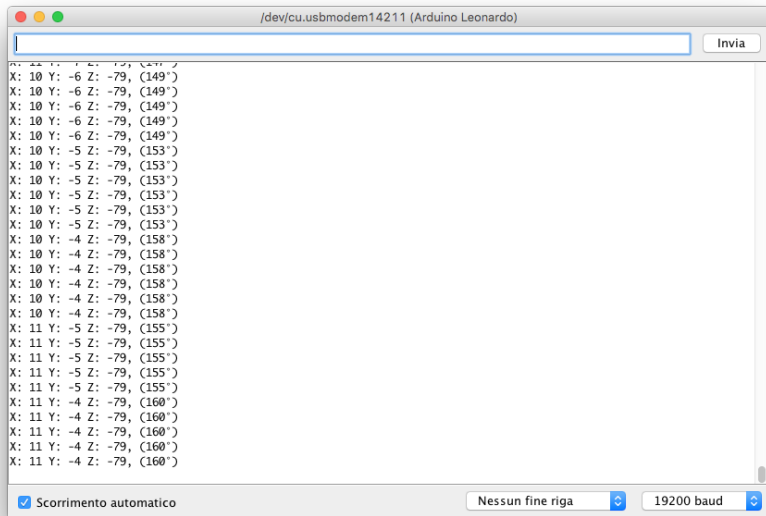
Inizializziamo il sensore

```
void setup(void) {  
    Serial.begin(115200);  
    /* Initialise the sensor */  
    if(!mag.begin())    {  
        Serial.println("No HMC5883 detected!");  
        while(1);  
    }  
}
```



```
void loop(void) {  
    sensors_event_t event;  
    mag.getEvent(&event);  
    int x = event.magnetic.x;  
    int y = event.magnetic.y;  
    int z = event.magnetic.z;  
    float heading = atan2(y, x);  
    int angle = int(heading * (180 / M_PI)+180);  
    char str[32];  
    sprintf(str, "X: %d Y: %d Z: %d, (%d°)", x, y, z, angle);  
    Serial.println(str);  
    delay(10);  
}
```

Trovate il sorgente in "esempi/Magnetometro_WITH_LIB"



- Misura **pressione** e **temperatura**
- Con una opportuna conversione è possibile ottenere **l'altitudine**
- Trovate il sorgente in **"esempi/Barometro_WITH_LIB"**

Se avete sensori poco comuni senza librerie già disponibili per Arduino dovreste scriverne una voi.

- Useremo la classe **Wire** di Arduino che gestisce il collegamento I²C per scrivere un **driver** del magnetometro

Setup Libreria Wire

Includiamo l'header della libreria

```
#include<Wire.h>
```

Setup:

- `Wire.begin();` Inizializza la libreria

Spesso è possibile cambiare il comportamento dei sensori scrivendo degli appositi valori dentro i loro registri di configurazione

- `Wire.beginTransmission(indirizzo_sensore);`
 - Avvia la trasmissione, l'argomento è l'**indirizzo del sensore**
 - Si usa solo prima di una **Wire.write()** e non per una `Wire.read()`
- `Wire.write(registro);`
 - Prima scrittura sul Bus, l'argomento è il **registro** del sensore dove vogliamo scrivere
- `Wire.write(valore);`
 - Seconda scrittura, l'argomento è il **valore** che vogliamo scrivere
- `Wire.endTransmission();`
 - Conclude la trasmissione
 - Si usa solo dopo una `Wire.write()` e non per una `Wire.read()`

Leggiamo le misure di un sensore

- `Wire.beginTransmission(indirizzo_sensore);`
 - Avvia la trasmissione, l'argomento è l'**indirizzo del sensore**
- `Wire.write(registro);`
 - Prima ed unica scrittura sul Bus, l'argomento è il **registro** del sensore che vogliamo leggere
- `Wire.endTransmission();`
 - Conclude la scrittura
- `Wire.requestFrom(indirizzo_sensore, N);`
 - Richiede al sensore N byte
- `uint8_t dato = Wire.read();`
 - Legge 8 bit dal Bus e li restituisce
 - Per leggere più byte bisogna chiamare più volte la `read()`

Dati divisi in più registri

Nella maggior parte dei casi i dati da leggere sono divisi in 2 o più registri come parte **alta** (HIGH) del dato e parte **bassa** (LOW)

```
uint8_t x_H = Wire.read();  
uint8_t x_L = Wire.read();  
int x = (int16_t)(x_L | ((int16_t)x_H << 8));
```

- Leggiamo i 2 registri
- **Shiftiamo** a sinistra di 8 bit il registro contenente la parte alta (x_H) del nostro dato
- e lo mettiamo in OR con la parte bassa
- Questo ci permette di "concatenare" i 2 registri da 8 bit ottenendo un dato a **16 bit**

Driver Magnetometro senza libreria

Nella cartella degli esempi trovate il driver per leggere dal magnetometro i valori raw di x, y, e z **"esempi/Magnetometro_WITHOUT_LIB"**

- Prima di comprare un sensore cercate e leggete **TUTTO** il suo datasheet, ci sono sensori scarsamente documentati. Magari spendendo un euro in più vi risparmiate dei gran mal di testa
- Non sempre le librerie che trovate sono abbastanza efficienti o fanno quello che volete voi, non abbiate paura di modificarle
- Se state scrivendo a mano un driver attenzione ai **valori di default** dei registri e controllate sempre di aver settato tutti i registri necessari

Domande?

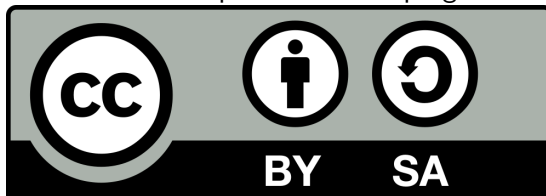
Se vi vengono in mente più tardi, fatele sul gruppo oppure non fatevi problemi e contattatemi:

email: matteo.piazzolla@gmail.com

facebook: Matteo Michele Piazzolla

twitter: @teo_piaz

In bocca al lupo con i vostri progetti!



Queste slides sono licenziate Creative Commons Attribution-ShareAlike 3.0 Unported si ringrazia **rbino** per i precedenti corsi Arduino e per le sue slide che ho usato come template

<https://www.poul.org>