

# CORSO DI LABORATORIO DI ELETTRONICA PER L'AUTOMAZIONE



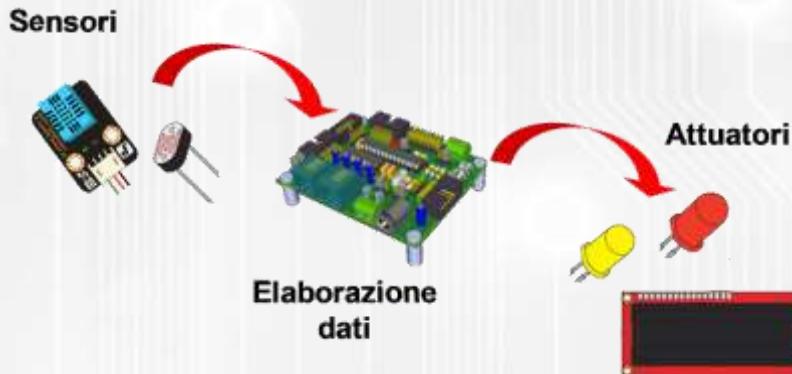
# Arduino

*Titolare del corso*  
*Prof. Andrea Cusano*

*Assistente:*  
*Ing. Patrizio Vaiano*

# *Sistema di automazione*

Il termine automazione identifica la tecnologia che usa sistemi di controllo per gestire macchine e processi, riducendo la necessità dell'intervento umano.

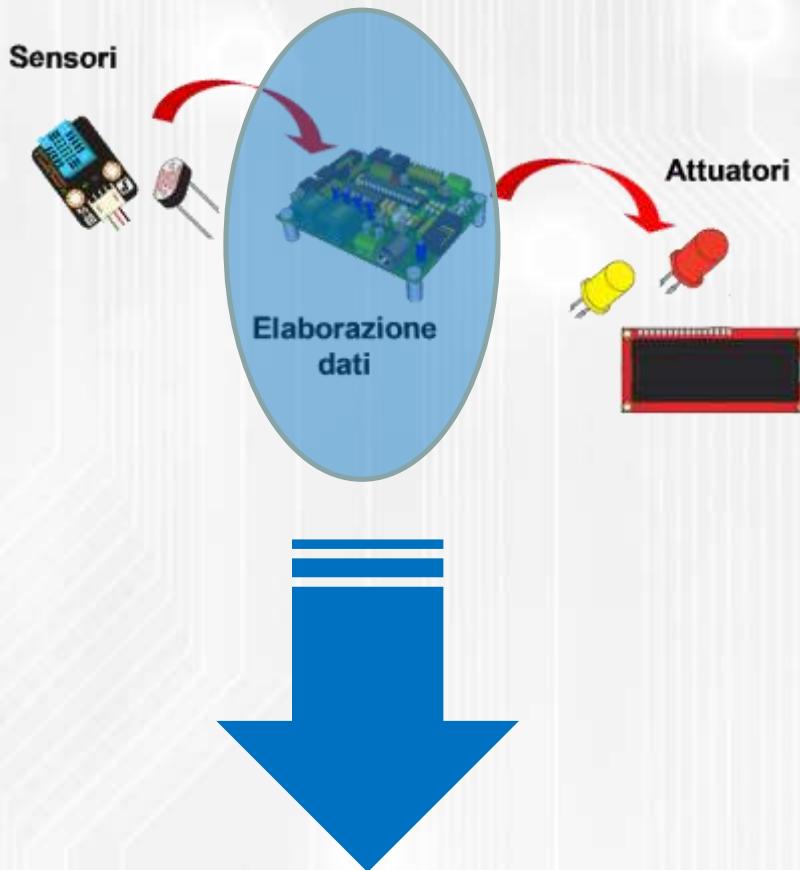


- ✓ Captare variazioni di parametri esterni e convertirli in segnali elettrici (sensori)
- ✓ Elaborare il segnale elettrico proveniente dai sensori tramite un ADC e generare segnali in uscita(elaborazione dati)
- ✓ Il segnale elettrico in uscita attiva dei dispositivi (attuatori)

Sistema di automazione, capace di captare quello che accade in un determinato ambiente e mediante l'elaborazione dei dati attivare degli attuatori o visualizzarlo a video

# *Sistema di automazione*

Il termine automazione identifica la tecnologia che usa sistemi di controllo per gestire macchine e processi, riducendo la necessità dell'intervento umano.



- ✓ Captare variazioni di parametri esterni e convertirli in segnali elettrici (sensori)
- ✓ Elaborare il segnale elettrico proveniente dai sensori tramite un ADC e generare segnali in uscita(elaborazione dati)
- ✓ Il segnale elettrico in uscita attiva dei dispositivi (attuatori)

# Cos' è Arduino?

- È una scheda elettronica di piccole dimensioni utile per *creare rapidamente prototipi* e per *scopi hobbistici e didattici*.
- È costituito da un *microcontrollore programmabile* e una *circuiteria di contorno* in grado di manipolare segnali digitali e analogici.
- Permette di realizzare velocemente piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, temperatura e umidità e molti altri progetti che utilizzano sensori, attuatori e comunicazione con altri dispositivi.
- È open-source: *software free* e *hardware riproducibile*.



# *La famiglia Arduino*



*Uno*



*Mega*



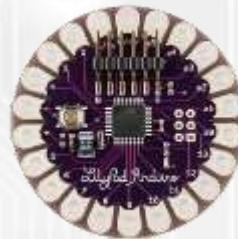
*Due*



*Leonardo*



*Yun*



*Lilypad*



*Nano*



*Ethernet*



*Esplora*



*Robot*



*Fio*



*Mini*



*Pro*

# *La famiglia Arduino*



*Uno*

*La versione più comune di Arduino è l' Arduino Uno, ed è di questa scheda di cui la maggior parte delle persone parla quando ci si riferisce ad un Arduino. Principali caratteristiche: Tensione operativa: 5VDC, 20 Pin I/O, Max 40mA di corrente erogabile da ogni pin, Max 200mA di corrente totale erogabile, Memoria flash di 32 kB, Frequenza del clock di 16 MHz.*



*Leonardo*

*La Leonardo è una scheda basata sul microcontrollore ATMEGA32U4. Dispone di 20 digitali pin di input/output (di cui 7 possono essere utilizzati come uscite PWM e 12 come ingressi analogici), un clock a 16 MHz, una connessione micro USB, un jack di alimentazione, un header ICSP e un pulsante di reset. Contiene tutto il necessario per supportare il microcontrollore, per collegarla a un computer con un cavo USB o alimentarla con un adattatore AC-DC o una batteria. Un'altra differenza importante è che l'ATMEGA32U4 è dotato di comunicazione diretta USB, quindi niente processore secondario: questo permette alla scheda di essere vista dal computer anche come una normale periferica.*

# *La famiglia Arduino*



*Due*

*La Due è una scheda basata sul microcontrollore Atmel SAM3X8E CPU ARM Cortex-M3, diventando così la prima scheda basata su un microcontrollore core ARM a 32-bit. Dispone di 54 digitali pin di ingresso/uscita (di cui 12 possono essere utilizzate come uscite PWM), 12 ingressi analogici, 4 UART (porte seriali hardware), un clock di 84 MHz, una OTG USB, 2 DAC (digitale a analogico), 2 TWI, un jack di alimentazione, un header SPI, un header JTAG, un pulsante di reset e un pulsante di cancellazione.*

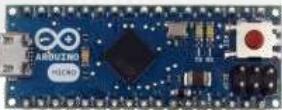
*A differenza di altre schede però la Due funziona a 3.3V, ed è anche la tensione massima che i pin possono sopportare.*



*Yun*

*La Arduino Yun è una scheda basata sul microcontrollore ATMEGA32U4 e monta anche un Atheros AR9331. L'Atheros supporta una distribuzione Linux basata su OpenWRT denominato Linino. La scheda ha integrato il modulo Ethernet e il modulo WiFi, un pin USB-A e uno slot per scheda micro-SD, 20 ingressi/uscite digitali (di cui 7 possono essere utilizzati come uscite PWM e 12 come analogici), un oscillatore a 16 MHz, una micro USB, un ICSP e un pulsante di reset. È una scheda molto completa, pronta alla connessione con il mondo, perchè ci permette di non dover acquistare Ethernet o wifi shield.*

# *La famiglia Arduino*



*Micro*



*Robot*

*La Arduino Micro è una scheda basata sul microcontrollore ATMEGA32U4, sviluppato in collaborazione con Adafruit. Dispone di 20 pin digitali di input/output (di cui 7 possono essere utilizzati come uscite PWM e 12 come ingressi analogici), un Clock a 16 MHz, una connessione micro USB, un header ICSP e un pulsante di reset. Come la Leonardo ha un microcontrollore che integra la comunicazione USB, eliminando la necessità di un processore secondario. E' fatta apposta per essere connessa ad una normale breadboard.*

*Un robot con due board basate sul microprocessore ATmega32u4: la Motor Board, dedicata esclusivamente al controllo dei due motori dell'Arduino Robot con cinque sensori pre-installati a infrarossi (e quattro pin digitali per collegare sensori supplementari, non poco assolutamente) e la Control Board con più elementi: otto pin analogici per sensori analogici, sei pin digitali collegabili ad altri sensori, un piccolo display LCD a colori, una bussola integrata, uno slot per schede di memoria SD e alcuni pulsanti di comando.*

*Si dovrebbe programmare solo la Control Board (simile alla Leonardo) mentre il firmware di default della Motor Board, attraverso una nuova libreria dedicata, ci evita la complessità della gestione dei motori e permette di fare tutto con dei semplici comandi come avvia i motori, ferma i motori, ruota di x gradi, eccetera.*

# *La famiglia Arduino*



*Esplora*

*La Esplora deriva direttamente dalla Leonardo. Si differenzia da tutte le precedenti schede perché reca già in sé alcuni sensori. E' progettata per chi vuole interagire direttamente con la scheda o utilizzarla come un joystick.*

*Presenta uscite per luci e suoni e come sensori in ingresso monta un joystick, un cursore, un sensore di temperatura, un accelerometro, un microfono e un sensore di luce. Per l'espansione può contare su due connettori di ingresso e di uscita Tinkerkit e una presa per uno schermo LCD a colori TFT.*



*Mega*

*Il Mega è la seconda versione che si incontra più facilmente in giro della famiglia: il Mega è come un fratello maggiore di Arduino Uno, più grosso e massiccio. Vanta 256 KB di memoria (8 volte più della Uno). Ha anche 54 pin di ingresso e di uscita di cui 16 analogici, 14 dei quali possono gestire la PWM. Questa scheda è adatta quando avete bisogno di una potenza che la scheda Uno non vi può dare. Una versione speciale della Mega è la ADK, fondamentalmente un Arduino Mega che è stata specificamente progettata per interfacciarsi con smartphone Android.*

# *La famiglia Arduino*

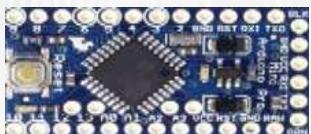


*Ethernet*

L'Arduino *Ethernet* è una scheda basata sul microcontrollore ATmega328; i suoi pin 10, 11, 12 e 13 sono riservati all'interfacciamento con il modulo Ethernet. Questo riduce il numero di pin disponibili a 9, di cui 4 utilizzabili anche come PWM.

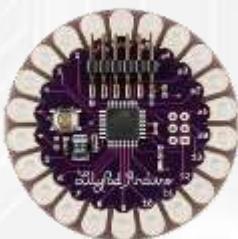
Ha integrato un lettore di schede microSD, perchè si possono memorizzare dei files per i vari web services che ci possiamo inventare.

Presenta un header seriale a 6 pin compatibile con l'adattatore seriale USB e con tutto il materiale Sparkfun e Adafruit che utilizza questo tipo di connessione. Per chi non possiede già una scheda e vuole la connessione Ethernet per i suoi progetti, è meglio acquistare questa configurazione piuttosto di una scheda e un ethernet shield a parte.



*Mini*

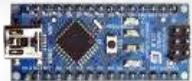
L'Arduino *Mini* è una piccola scheda di sviluppo originariamente basato sul ATmega328, pensata per le breadboard e per i progetti dove lo spazio proprio non c'è. Dispone comunque di 14 pin digitali di ingresso/uscita (6 PWM) e di 8 ingressi analogici.



*LilyPad*

Il *LilyPad* è una versione peculiare di Arduino ed è stato progettato per applicazioni indossabili e tessili! E' fatto apposta per essere cucito al tessuto e collegato ad altri componenti con del filo conduttivo, sapientemente nascosto ed integrato nelle pieghe del tessuto o in qualche altro modo.

# *La famiglia Arduino*



*Nano*

*Questa scheda ha più o meno le stesse caratteristiche e le stesse funzionalità della Duemilanove con ovviamente una configurazione e una dimensione differente. Non ha la presa di corrente e funziona con un cavo USB Mini-B al posto di uno standard.*



*Pro*

*La Pro è pensata per l'installazione semi-permanente o definitiva: è da usare quindi quando il vostro progetto non è più in fase di studio e si sta avviando verso una preserie o qualcosa di più. Per questo la scheda è senza pin premontati, permettendo l'uso di vari tipi di connettori o la saldatura diretta dei fili.*



*Fio*

*Questa scheda dispone di una connessione per una batteria ai polimeri di Litio corredata da un circuito di carica tramite USB. Un montaggio per XBee è disponibile sul fondo della scheda, quindi la Fio è destinata alle applicazioni wireless e superleggere. Utilizzando un adattatore modificato USB-to-XBee come XBee Explorer USB, possiamo caricare gli sketches in modalità wireless. La scheda è senza pin premontati, permettendo l'uso di vari tipi di connettori o di saldare direttamente i cavi risparmiando peso e spazio.*

*La Fio è stata progettata da Shigeru Kobayashi e prodotta da SparkFun Electronics.*

# *La famiglia Arduino – ultimi arrivati*



Arduino® UNO R4 WiFi



Arduino® UNO R4 Minima



Arduino Zero



Arduino GIGA R1 WiFi



ARDUINO UNO WiFi REV2

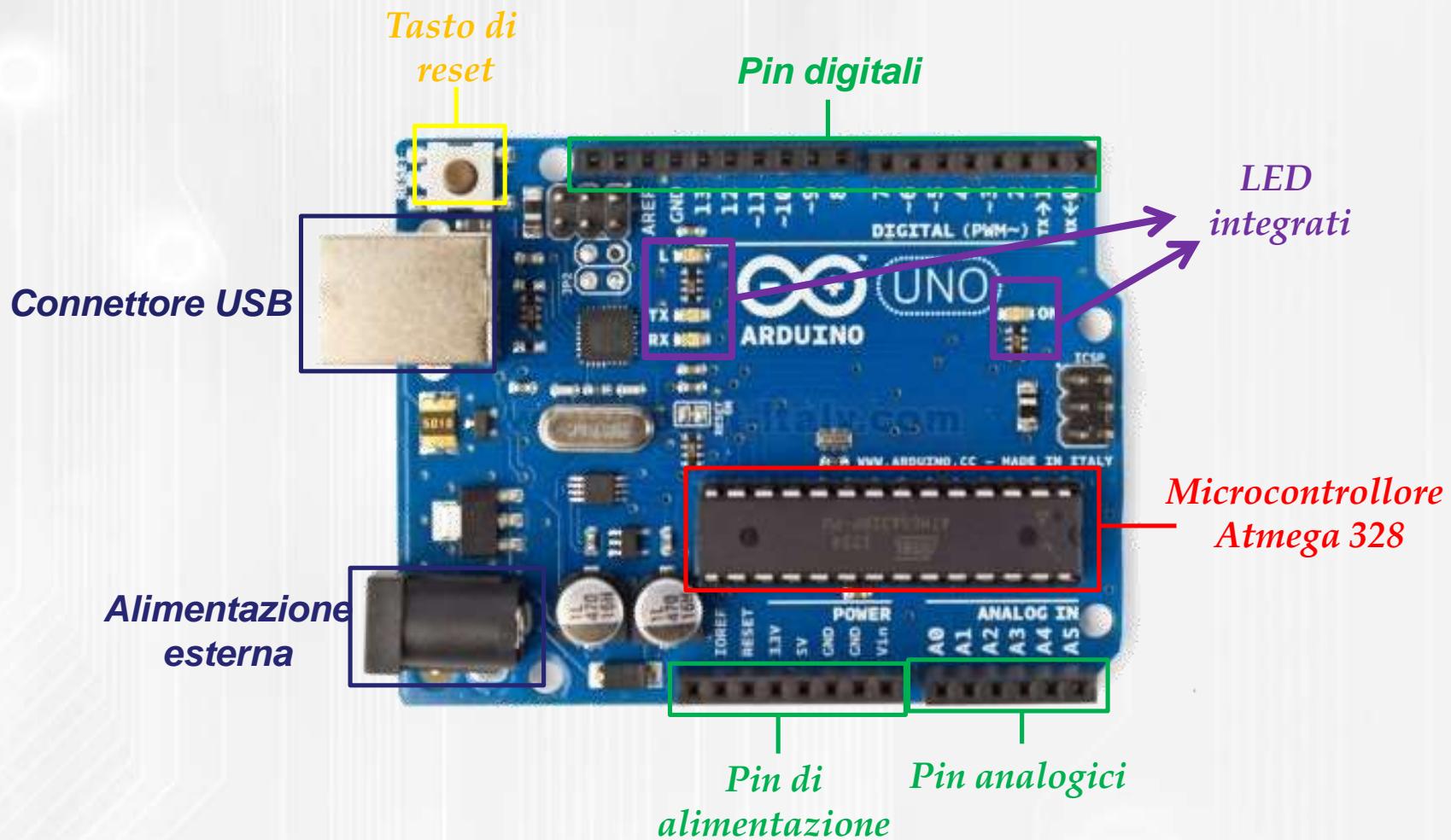
# *Quale scegliere?*

*Come ogni altro prodotto, anche una scheda Arduino va scelta in base al progetto da realizzare e non leggendo semplicemente le caratteristiche tecniche.*

*Prima di scegliere la scheda da acquistare, bisogna valutare i requisiti del progetto che si vuole sviluppare così da selezionare quella più adatta alla propria applicazione.*

- *Nr. di periferiche da controllare*
- *Mole di elaborazione*
- *Codice da sviluppare (le librerie non sono compatibili con tutte le board in commercio)*
- *Scenari applicativi (connessione ad internet, wi-fi , etc)*

# Arduino Uno



# Arduino Uno

## IL MICROCONTROLLORE

*...un po' di teoria*

*Un microcontrollore è un dispositivo di elaborazione dati simile al microprocessore.*

*Un microprocessore è costituito da:*

- *ALU*
- *Circuiti di controllo e temporizzazione*
- *Registri*
- *Bus*

*e per funzionare ha bisogno di:*

- *Memorie*
- *Interfacce I/O*
- *Alimentazione (anche decine di Watt)*

*Un microcontrollore è costituito da:*

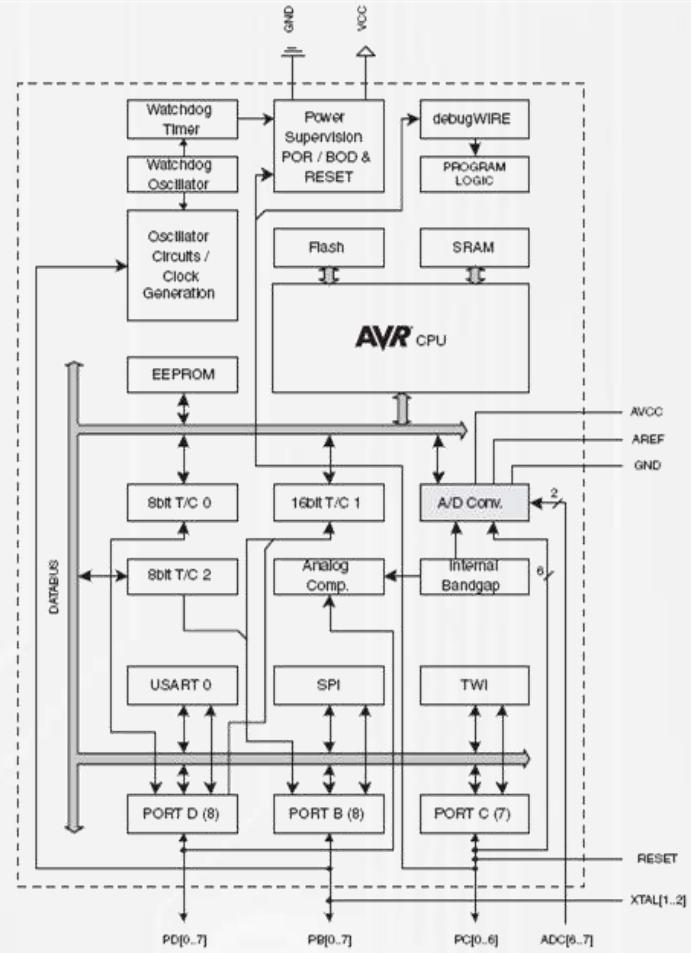
- *CPU RISC*
- *Registri*
- *Bus*
- *Piccole memorie*
- *Porte di I/O*
- *Alimentazione*
- *Contatori, timer, convertitore A/D.*
- *interfacce di comunicazione di vari tipi.*

# Arduino Uno

## IL MICROCONTROLLORE

### Microcontrollore ATmega328

- Tensione operativa: 5VDC
- 20 Pin I/O
- Max 40mA di corrente erogabile da ogni pin
- Max 200mA di corrente totale erogabile
- Memoria flash di 32 kB
- Frequenza del clock di 16 MHz



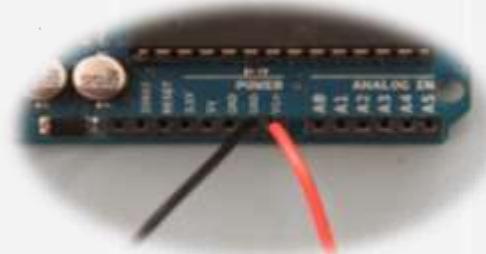
# Arduino Uno

## ALIMENTAZIONE

- da PC attraverso cavo USB.



- collegandolo direttamente al pin Vin o al pin 5V



- da rete attraverso il connettore esterno



# Arduino Uno

## TASTO DI RESET

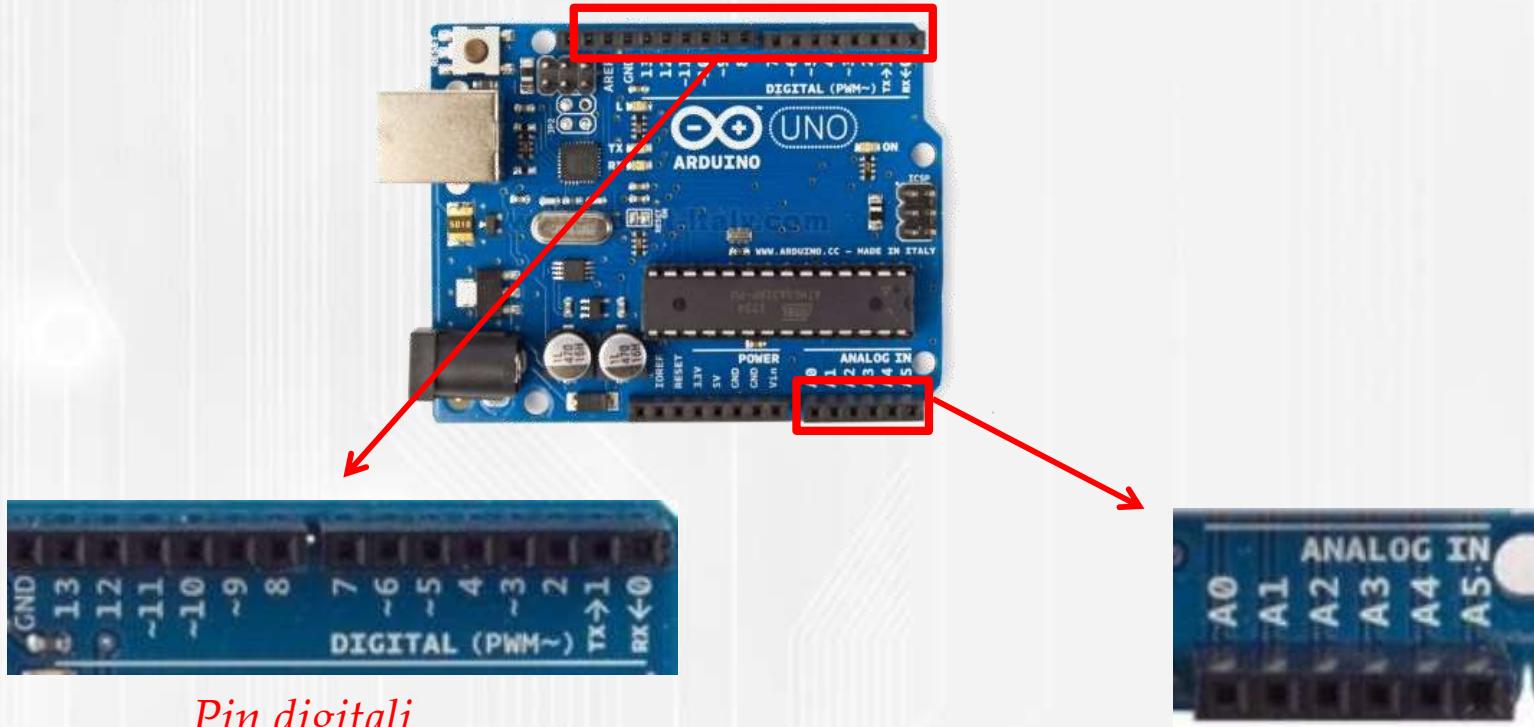
*Non cancella il programma nella memoria!*



*Schiacciando il tasto RESET l' Atmega esegue il codice indicato dall'indirizzo memorizzato nel vettore Reset, il quale controlla se c' é una richiesta load di un programma. Se non c' é incomincia ad eseguire il programma memorizzato.*

# Arduino Uno

## PIN DI I/O



*Pin digitali*

- pin digitali di I/O (2-13) di cui 6 PWM (3,5,6,9,10 e 11)
- Pin di RX e TX (0 e 1)
- Ground (GND)

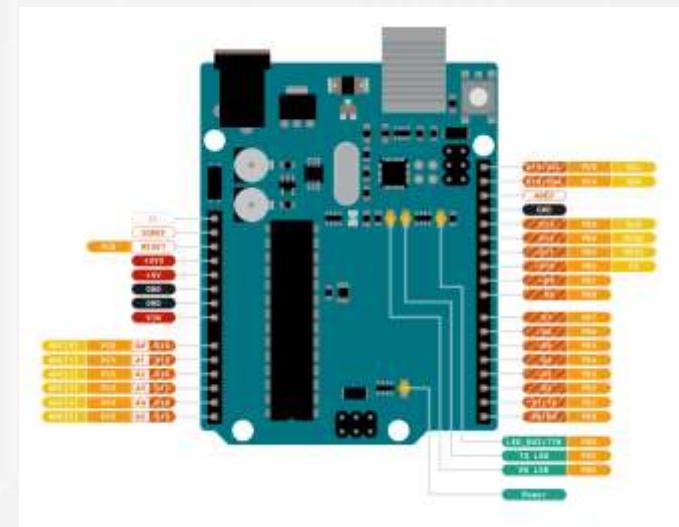
*Pin analogici*

- 6 pin di input analogico
- AREF (nella sezione DIGITAL)

# Arduino Uno

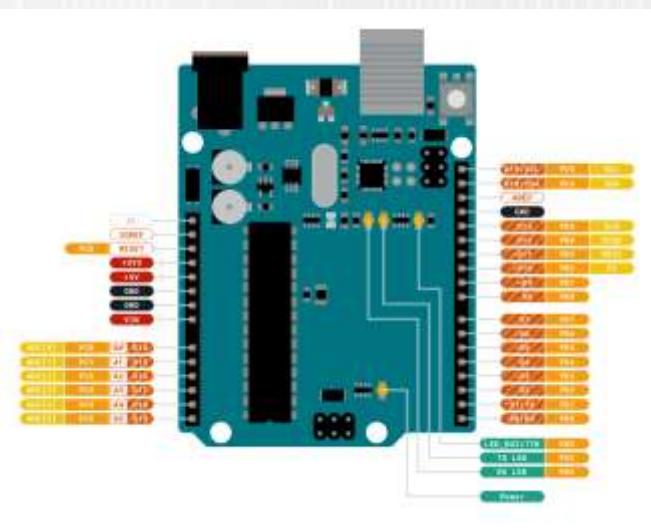
## PIN DI I/O

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line



# Arduino Uno

## PIN DI I/O



Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

# *Arduino Uno*

## **PIN DI I/O**

### *Pin digitali*

- *pin digitali di I/O (2-13) di cui 6 PWM (3,5,6,9,10 e 11).*

# *Arduino Uno*

## **PIN DI I/O**

### *Pin digitali*

- *pin digitali di I/O (2-13) di cui 6 PWM (3,5,6,9,10 e 11).*

### *PWM...un po' di teoria*

*La tecnica denominata P.W.M. ( Pulse Width Modulation ) consta essenzialmente nel trasmettere l'informazione attraverso un segnale impulsivo mediante la larghezza dell'impulso stesso. Questa particolare meccanismo viene usata prevalentemente nei controlli automatici nel blocco denominato "attuatore", ossia il blocco che comanda l'impianto sotto controllo.*

# Arduino Uno

## PIN DI I/O

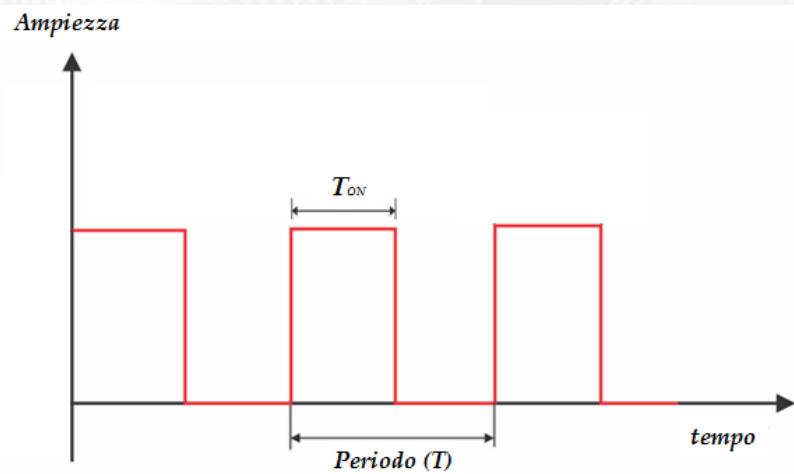
### Pin digitali

- pin digitali di I/O (2-13) di cui 6 PWM (3,5,6,9,10 e 11).

### PWM...un po' di teoria

Un segnale PWM è caratterizzato essenzialmente da due parametri:

- il periodo  $T$  (dal quale si ricava la frequenza  $f=1/T$ );
- il duty cycle  $\delta=T_{ON}/T$ .



Il valor medio (componente continua) del segnale PWM è il prodotto dell'ampiezza del segnale con il duty cycle.

# Arduino Uno

## PIN DI I/O

### Pin digitali

- pin digitali di I/O (2-13) di cui 6 PWM (3,5,6,9,10 e 11).

### PWM in Arduino Uno

- Ampiezza = 5 Volt
- Frequenza = 976Hz (pin 6 e 5) o 488Hz (pin 3, 9, 10 e 11)
- Duty cycle variabile tra 0 e 255 (8-bit)

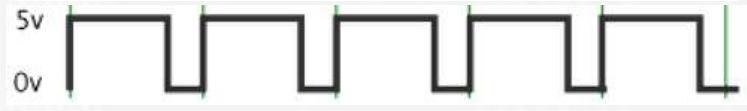
0 ---> 0% ( $T_{ON}=0 \text{ sec}$ )



255 ---> 100% ( $T_{ON}=T$ )



191 ---> 75% ( $T_{ON}=T/2$ )



64 ---> 25% ( $T_{ON}=T/4$ )

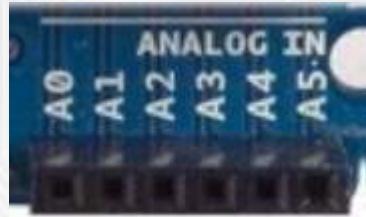


# Arduino Uno

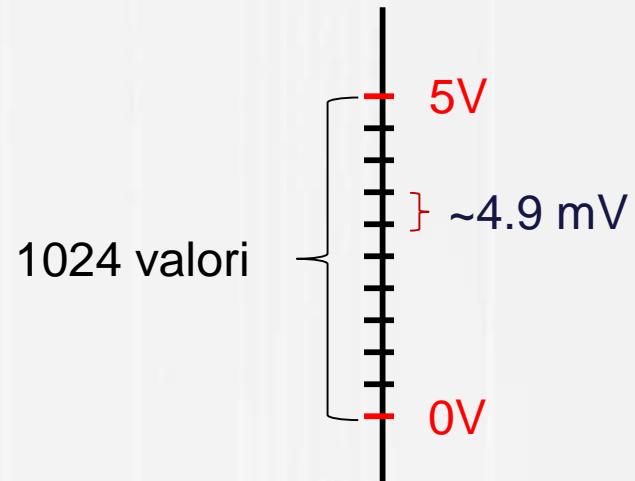
## PIN DI I/O

### Pin analogici

- 6 pin di **input** analogici (A0-A5)



- Range di valori in ingresso: 0-5V
- Risoluzione 10bit (0-1023)



Quindi ogni livello sarà di  $5V / 1023 = 4.88 mV$

# Arduino Uno

## PIN DI I/O

Come utilizzarli?



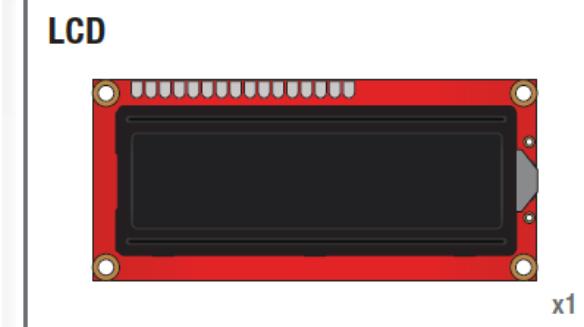
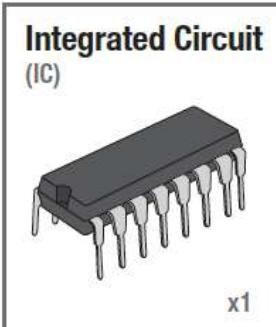
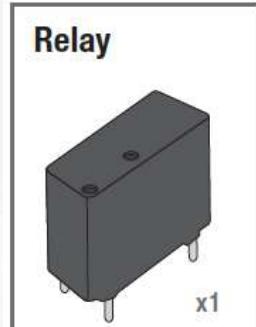
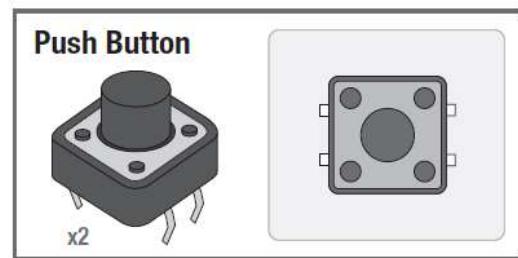
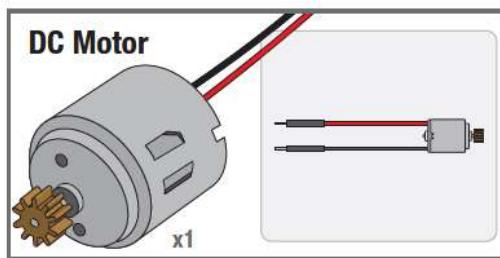
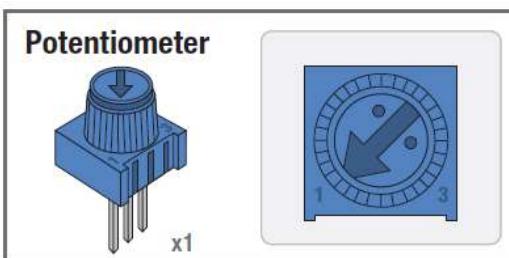
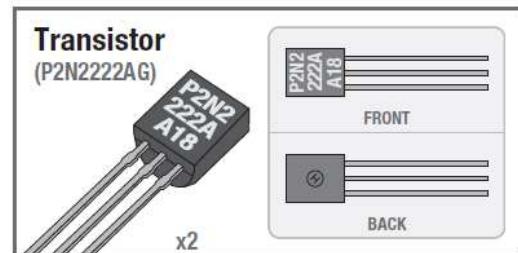
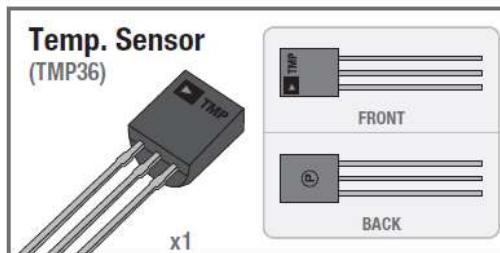
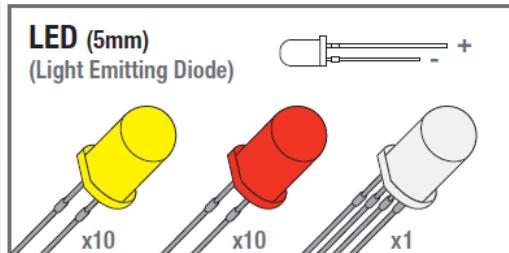
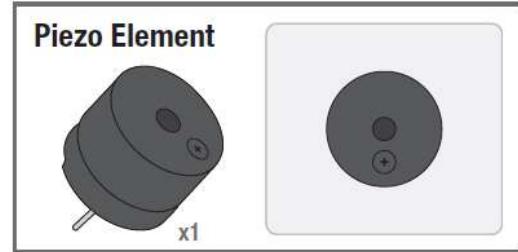
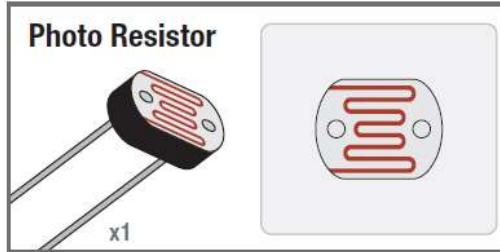
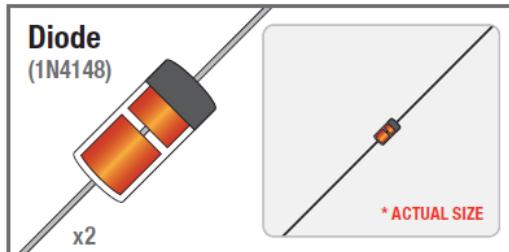
Per permettere ad Arduino di comunicare e interagire col «mondo esterno» c'è bisogno di:

- **SENSORI**: dispositivi fisici che leggono un segnale fisico dall'esterno (luce, pressione, temperatura, etc) e la convertono in un segnale elettrico
- **ATTUATORI**: dispositivo elettronico che trasformano un segnale elettrico in movimento o altre forme di segnali (luce, suono, etc)

# Arduino Uno

## PIN DI I/O

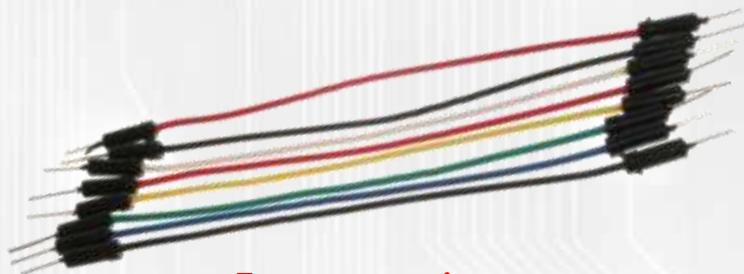
Come utilizzarli?



# Arduino Uno

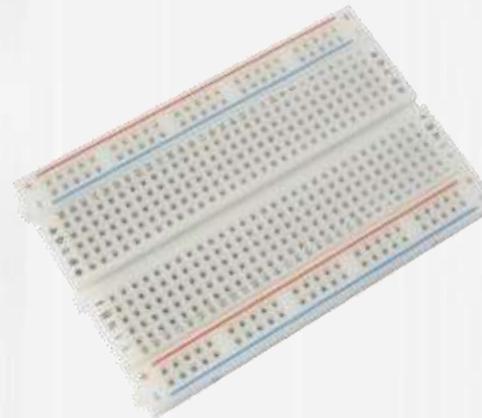
PIN DI I/O

Come collegarli?

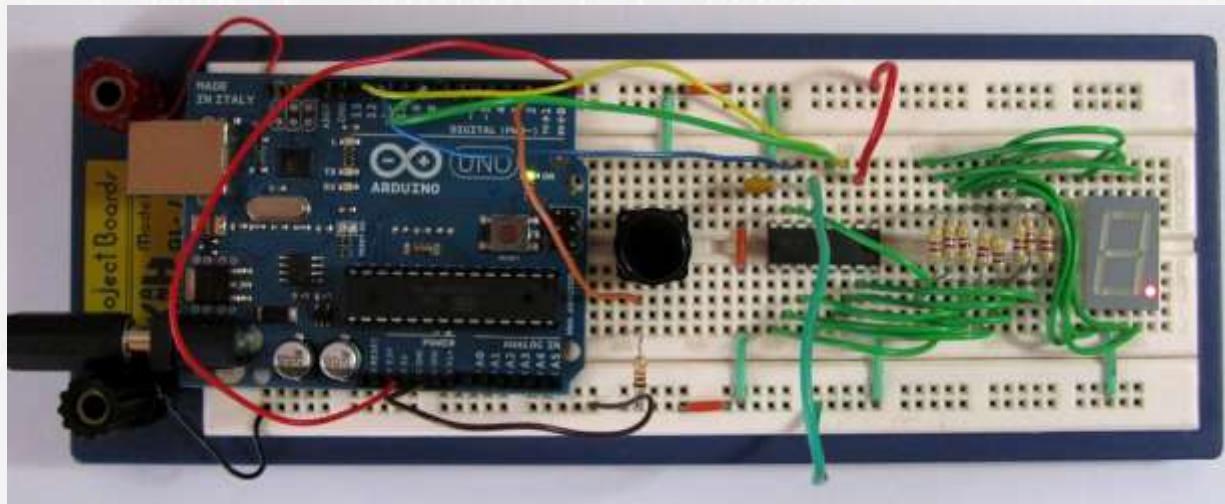


Jumper wires

+



Bread Board

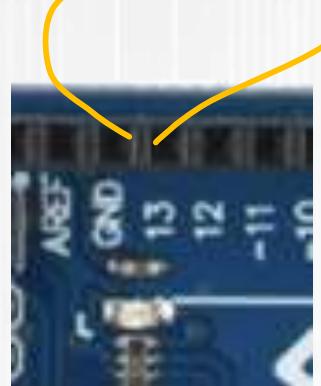
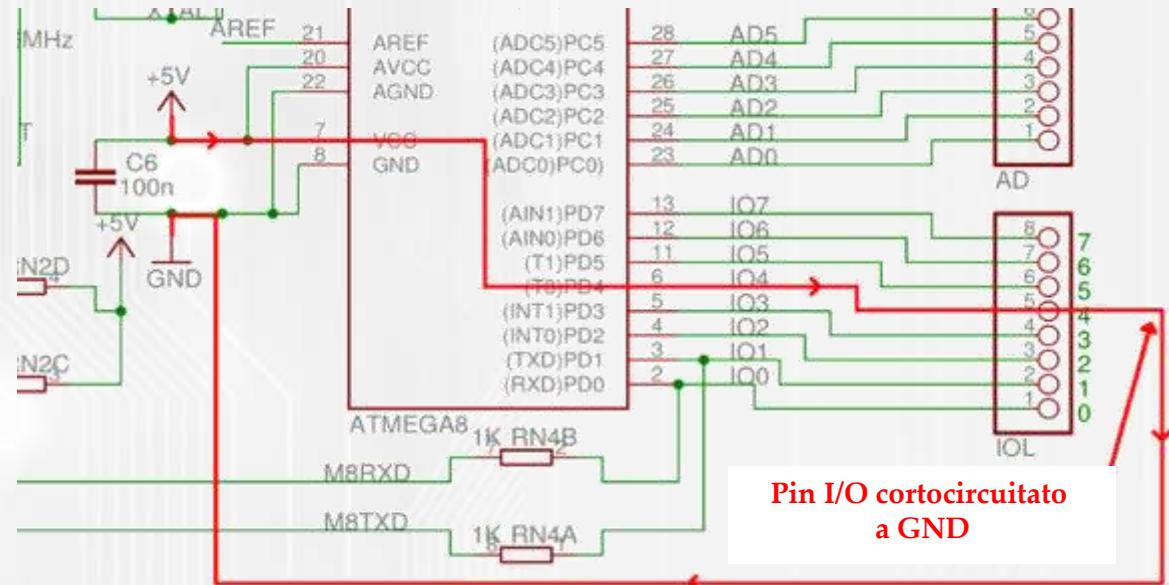


(solo per versioni prototipali)

# Arduino Uno

PIN DI I/O

*Attenzione ai collegamenti!*



# Arduino Uno



*La scheda tecnica del microcontrollore indica che 40 mA è il valore massimo di corrente per singolo PIN. Con una resistenza interna mediamente di soli 25Ω per PIN, un corto a massa può far passare fino a  $(5V / 25\Omega) = 200$  mA di corrente, più che sufficiente per distruggere il pin del microcontrollore.*

## PIN DI I/O

### *Attenzione ai collegamenti!*

#### Product Description

#### Additional Information

#### Reviews

#### Technical Details

Microcontroller: ATmega328

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

For the tutorial of installing Arduino IDE, refer to WIKI

#### Package included

1x SunFounder Uno R3 (Arduino compatible)

# *Arduino Uno*

*PIN DI I/O*



*Attenzione ai collegamenti!*

*...ancora...*

- *Applicando una tensione superiore a 5 V ad un qualunque PIN*

# Arduino Uno

*Per cominciare basta un PC, un cavo USB A-B e il nostro microcontrollore*



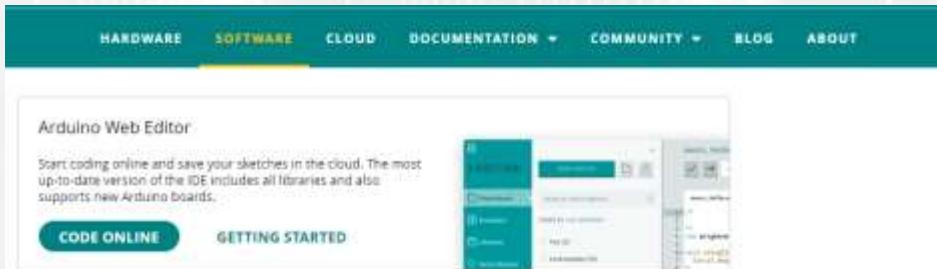
*Arduino può essere programmato mediante IDE*



# Arduino Uno

## Programmazione in IDE

- Ambiente di sviluppo free
- Compatibile con diversi sistemi operativi (Windows, Linux, Mac OS X )
- Diverse versioni con diverse librerie disponibili online



The screenshot shows the Arduino Web Editor homepage. At the top, there's a navigation bar with links for HARDWARE, SOFTWARE, CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. Below the navigation bar, the text "Arduino Web Editor" is displayed, followed by a brief description: "Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards." There are two prominent buttons: "CODE ONLINE" and "GETTING STARTED". To the right of these buttons is a small thumbnail image of the web editor interface.

## Downloads



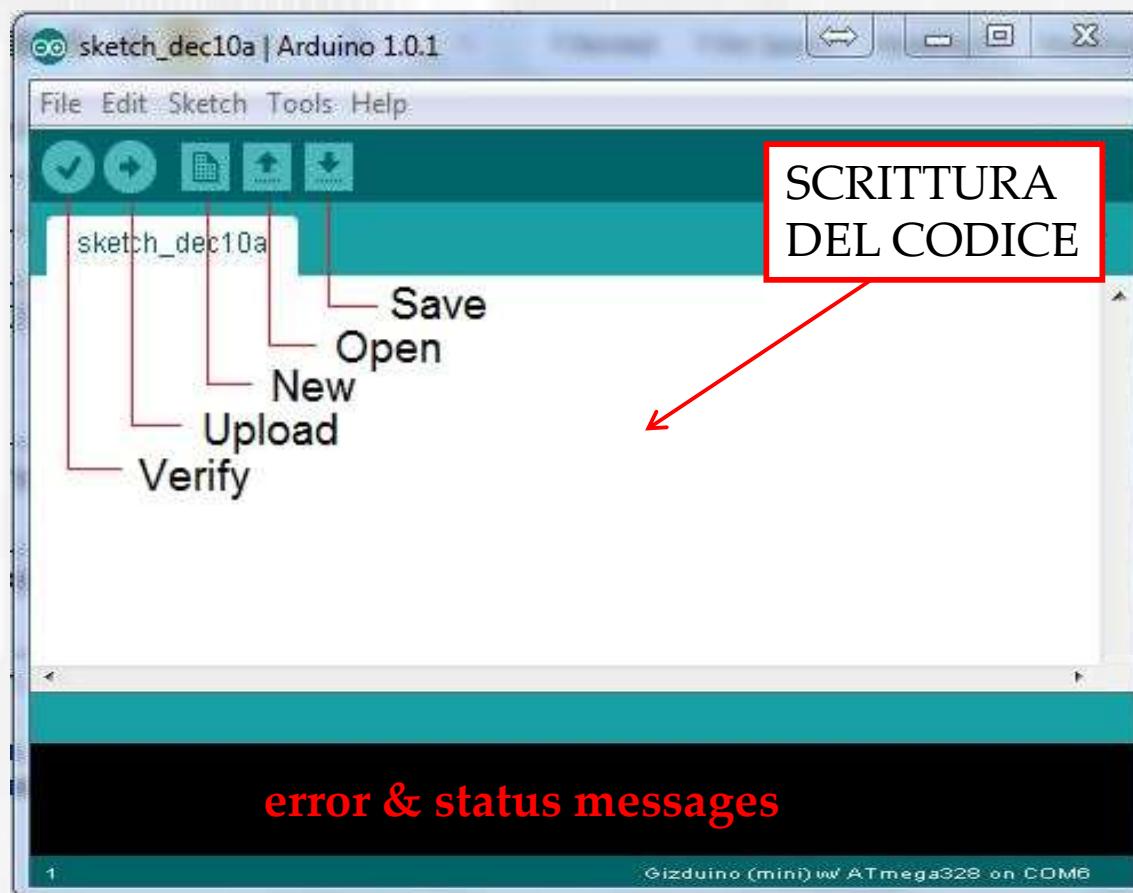
The screenshot shows the download page for Arduino IDE 2.1.0. On the left, there's a section for "Arduino IDE 2.1.0" featuring a logo, a brief description of the new features, and a link to the "Arduino IDE 2.0 documentation". On the right, there's a "DOWNLOAD OPTIONS" section with links for Windows (Win 10 and newer, 64 bits), Windows (MSI installer), Windows (ZIP file), Linux (AppImage 64 bits (x86-64)), Linux (ZIP file 64 bits (x86-64)), macOS (Intel, 10.14: "Mojave" or newer, 64 bits), and macOS (Apple Silicon, 11: "Big Sur" or newer, 64 bits). Below these links is a "Release Notes" link.

- ✓ Arduino IDE 1.8.x (classic)
- ✓ Arduino IDE 2.0.x (new)
- ✓ Arduino Web Editor (online)

# Arduino Uno

## Programmazione in IDE

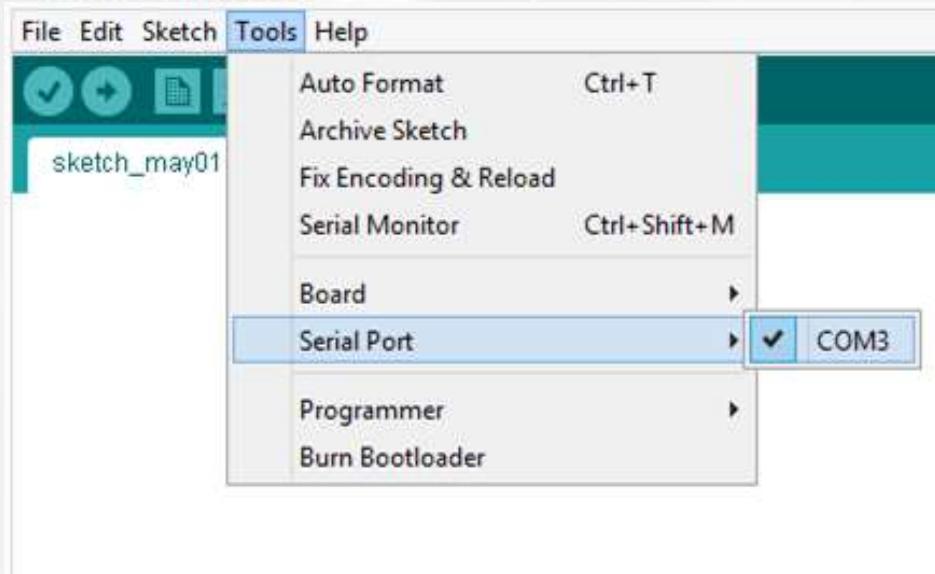
- Ambiente di sviluppo free
- Compatibile con diversi sistemi operativi (Windows, Linux, Mac OS X )
- Diverse versioni con diverse librerie disponibili online



# Arduino Uno

## Programmazione in IDE

*Il computer comunica con il microcontrollore attraverso una porta seriale.*

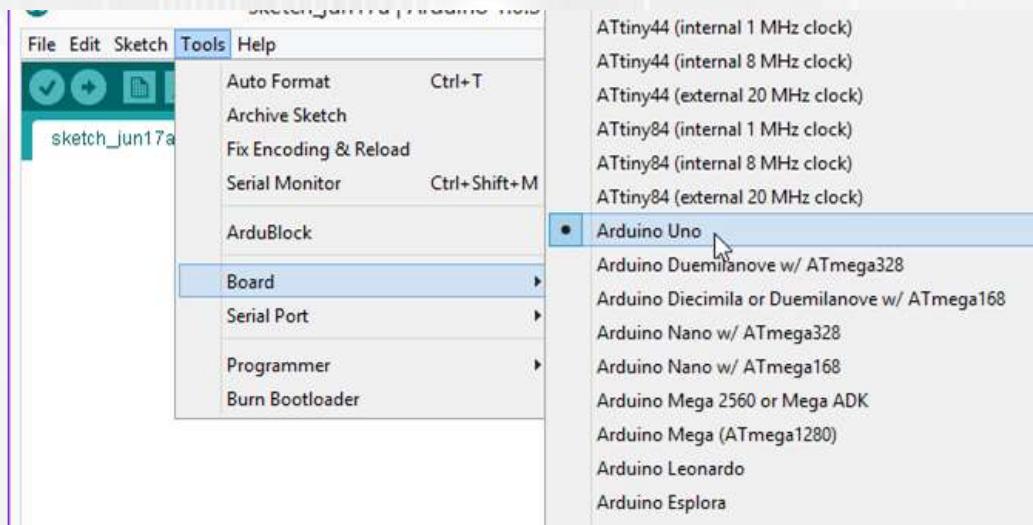


*Bisogna indicare al microcontrollore la porta seriale connessa del PC connessa ad Arduino*

# Arduino Uno

## Programmazione in IDE

Bisogna poi selezionare il tipo di scheda che si sta utilizzando.



Non tutte le librerie sono compatibili con tutte le schede Arduino!

# Arduino Uno

## Programmazione in IDE

The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, undo, redo, and upload. The main window displays a sketch titled "BareMinimum". The code consists of two functions: setup() and loop(). The setup() function contains a comment about running once. The loop() function contains a comment about running repeatedly.

```
File Edit Sketch Tools Help
BareMinimum
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

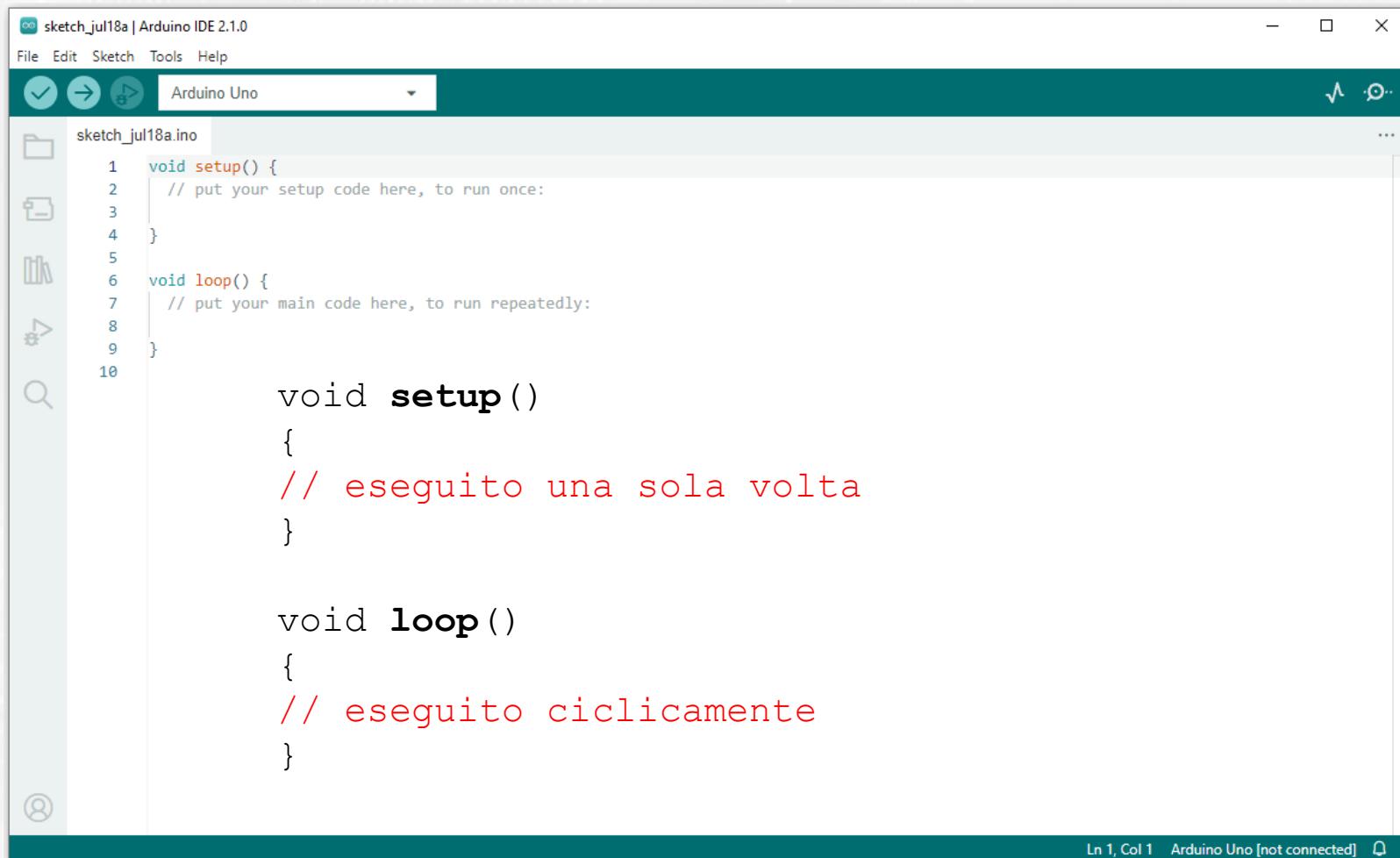
LilyPad Arduino w/ ATmega328 on COM28

```
void setup()
{
  // eseguito una
  sola volta
}
```

```
void loop()
{
  // eseguito
  ciclicamente
}
```

# Arduino Uno

## Programmazione in IDE



```
sketch_jul18a | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Arduino Uno
sketch_jul18a.ino
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
10

void setup() {
{
// eseguito una sola volta
}

void loop() {
{
// eseguito ciclicamente
}

Ln 1, Col 1  Arduino Uno [not connected] Q
```

# Arduino Uno

## WEB EDITOR

The screenshot shows the Arduino Web Editor interface. On the left is a sidebar with a teal header labeled 'EDITOR' containing links for Sketchbook, Examples, Libraries, Monitor, Reference, Help, Preferences, and Features usage. At the bottom of the sidebar is a 'CLOUD' button with a cloud icon. The main area has a light gray header with 'Untitled\_jun09a' and a 'NEW SKETCH' button. Below the header is a search bar for 'SEARCH SKETCHBOOK'. The central part of the screen is a code editor with tabs for 'Untitled\_jun09a.ino', 'ReadMe.adoc', and 'thingProperties.h'. The code editor displays the following C++ code:

```
1+ /*
2 Sketch generated by the Arduino IoT Cloud Thing "Untitled"
3 https://create.arduino.cc/cloud/things/ca587f5-994a-48ca-8f5b-c583c9fd481b
4
5 Arduino IoT Cloud Variables description
6
7 The following variables are automatically generated and updated when changes are made to the Thing
8
9 - No variables have been created, add cloud variables on the Thing Setup page
10 | to see them declared here
11
12 Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
13 which are called when their values are changed from the Dashboard.
14 These functions are generated with the Thing and added at the end of this sketch.
15 */
16 #include "thingProperties.h"
17
18 void setup() {
19 // Initialize serial and wait for port to open:
20 Serial.begin(9600);
21 // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
22 delay(1500);
23
24 // Defined in thingProperties.h
25 initProperties();
```

# *Arduino Uno*

## **Tipico workflow**

Per caricare il codice su una scheda Arduino utilizzando l'IDE, in genere si procede come segue:

1. Installa la tua scheda: significa installare il "pacchetto" (package) giusto per la tua scheda. L'installazione viene eseguita direttamente nell'IDE ed è un'operazione semplice e veloce.
2. Crea un nuovo sketch: uno sketch è il file di programma principale (main). Qui scriviamo una serie di istruzioni che vogliamo eseguire sul microcontrollore.
3. Compila il tuo sketch: il codice che scriviamo non è esattamente come appare quando viene caricato sul nostro Arduino: compilare il codice significa controllarlo per eventuali errori e convertirlo in un file binario (1 e 0). Se qualcosa va storto, è notificato nella console degli errori.

...

# *Arduino Uno*

## **Tipico workflow**

...

4. Carica il tuo sketch: una volta completata la compilazione, il codice può essere caricato sulla scheda. In questo passaggio collegiamo fisicamente la scheda al computer e selezioniamo la porta seriale corretta.
5. Monitor seriale (opzionale): per la maggior parte dei progetti Arduino, è importante sapere cosa sta succedendo sulla scheda. Lo strumento Serial Monitor disponibile in tutti gli IDE consente l'invio dei dati dalla scheda al computer.

# Arduino Uno

## Serial Monitor

The screenshot shows the Arduino IDE interface. The top menu bar displays "AnalogReadSerial | Arduino IDE 2.0.0-rc8". The main window shows a sketch named "AnalogReadSerial.ino" with the following code:

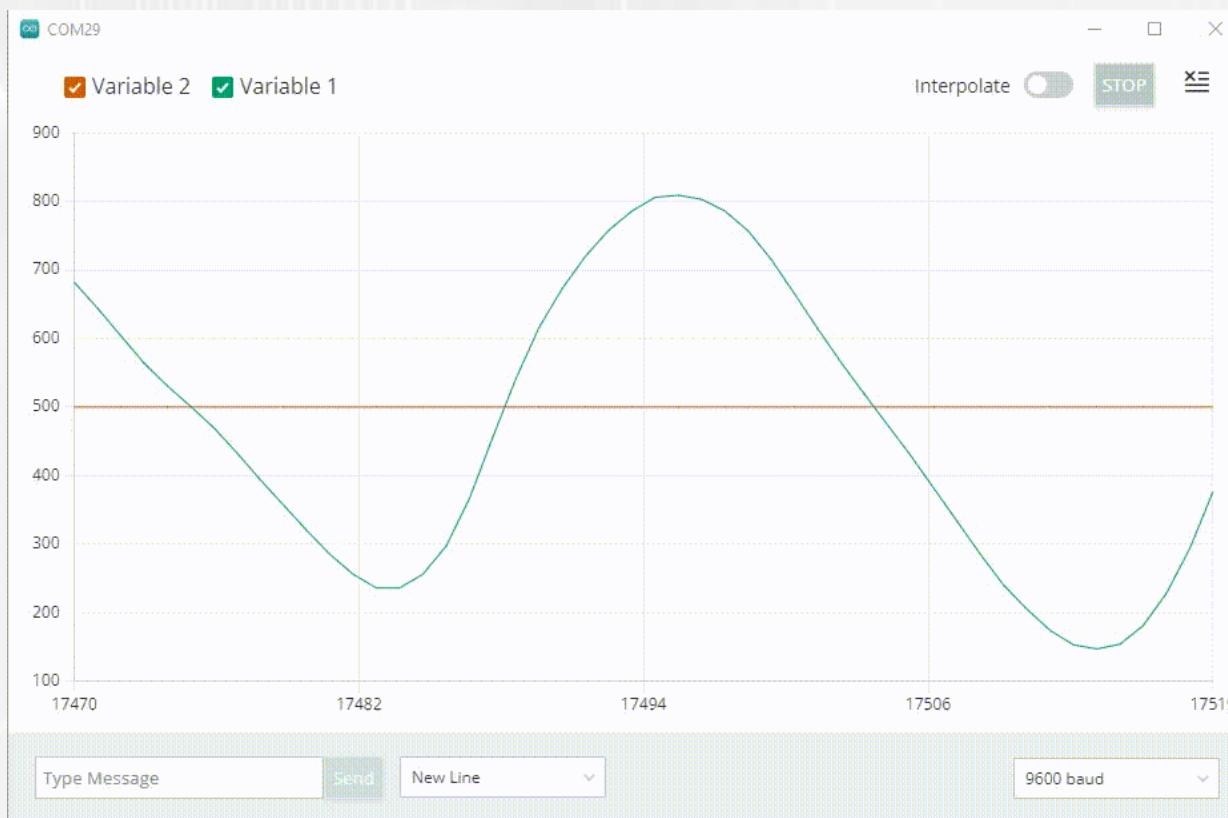
```
1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.print("Hello World!");
7   delay(1000);
8 }
```

The bottom half of the screen shows the "Serial Monitor" tab, which is currently active. It has a message input field at the top with the placeholder "Message (Esc + Enter to send message to Arduino Uno WiFi 101/102 on /dev/tty.usbmodem1421)". Below this, the output window displays the repeated text "Hello World!".

Serial.print() command

# *Arduino Uno*

## **Serial Plotter**



# Arduino Uno

## Debugging e autocompletion

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, Help, and a central tab labeled "Arduino Zero (Programming Port)". The left sidebar has sections for DEBUG, THREADS, CALL STACK, VARIABLES, WATCH, BREAKPOINTS, and CORTEX PERIPHERALS. Under BREAKPOINTS, two entries for "Blink.ino" are listed with checkboxes checked. The main code editor displays the "Blink.ino" sketch:

```
17 // modified 8 Sep 2011
18 by Colby Hansen
19
20 this example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by
36   delay(1000); // wait for a second
37 }
38
```

The bottom section is titled "Output" and contains the following log:

```
Info : accepting 'gdb' connection on tcp/50009
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x000002274 msp: 0x20007fb0
Info : SAMD MCU: SAMD21G18A (256KB Flash, 32KB RAM)
target halted due to debug-request, current mode: thread
```

The screenshot shows the Arduino IDE interface with a sketch titled "sketch\_jul5a.ino" running on an "Arduino MKR WiFi 1010". The code editor shows the following setup:

```
1 void setup() {
2
3 }
4
5 void loop() {
6   | digitalWrite
7   |
8 }
```

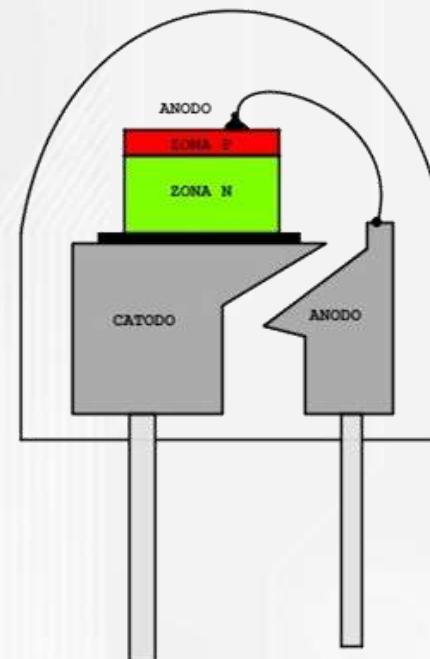
An autocompletion dropdown is open at the end of the line "digitalWrite", listing options such as "\*digitalWrite(pin\_size\_t pinNumber, PinStatus st...)" and "\*digitalWrite(pin\_size\_t pinNumber, int status)".

# Arduino Uno

## **Programmazione in IDE...facciamo lampeggiare un LED!**

*Il termine "LED" è un acronimo che sta per "Light Emitting Diode", ovvero "diodo che emette luce". I led sono costituiti da materiali in grado di emettere radiazioni luminose quando sono attraversati da una corrente elettrica. Per avere una luce apprezzabile senza distruggerlo il valore di corrente deve essere, per i piccoli LED, tra 5 e 20 mA circa.*

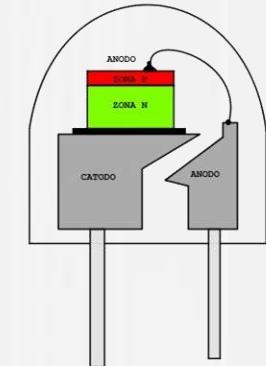
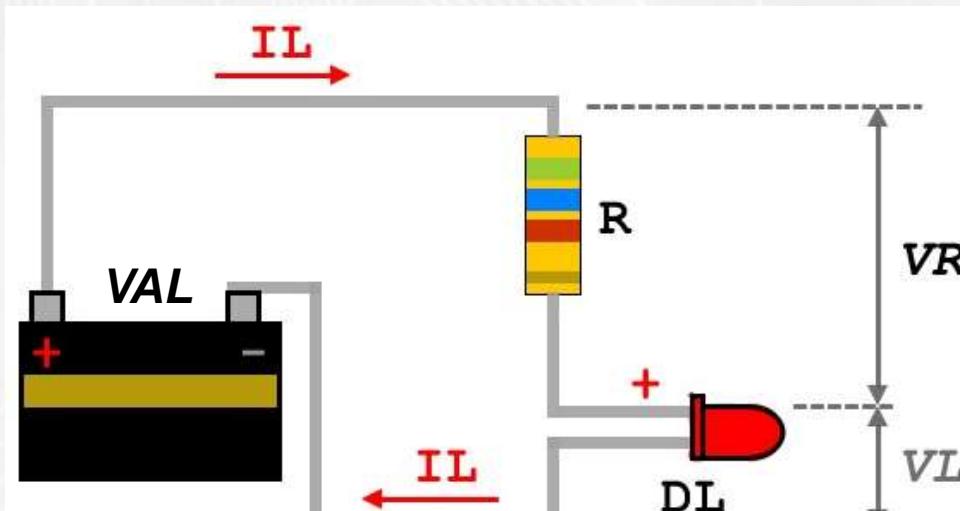
*I led hanno un terminale positivo chiamato anodo, ed uno negativo chiamato catodo, e per funzionare devono essere inseriti nel circuito rispettando tale polarità.*



# Arduino Uno

## Programmazione in IDE...facciamo lampeggiare un LED!

Quando si utilizza un led (**DL**), è necessario disporre un'opportuna resistenza (**R**) in serie ad esso, allo scopo di limitare la corrente (**IL**) che passa ed evitare che possa distruggersi; la caduta di tensione ai capi di un led (**VL**) può variare da 1 a 3.5 V circa, in funzione della lunghezza d'onda della radiazione emessa, ossia del colore.



Caratteristiche led

Colore	<b>VL</b>
Rosso	1.8V
Giallo	1.9 V
Arancio	2.0 V
Verde	2.3 V
Azzurro	3.0 V
Blu	3.5 V

Con **IL**=5-20mA

# Arduino Uno

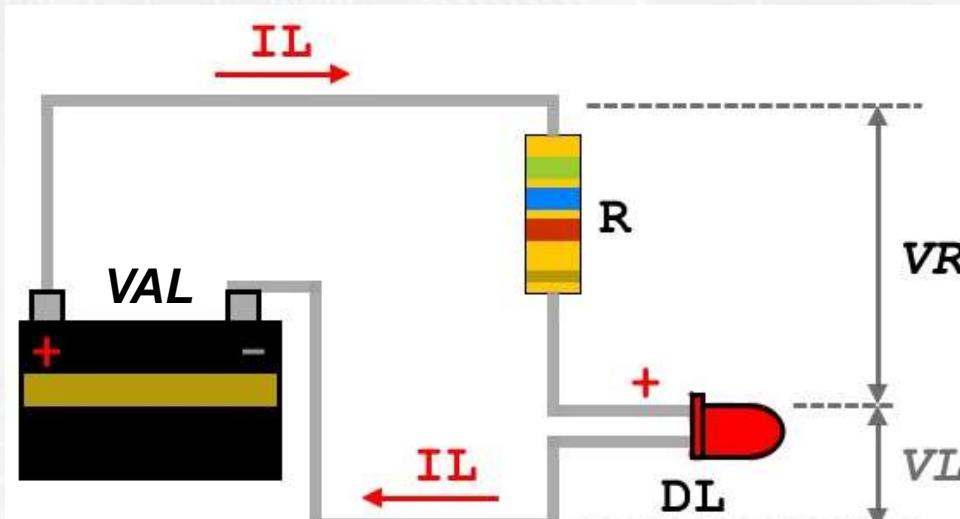
**Programmazione in IDE...facciamo lampeggiare un LED!**

$$VAL = VL + VR$$



$$IL = \frac{VAL - VL}{R}$$

$$VR = IL \times R$$



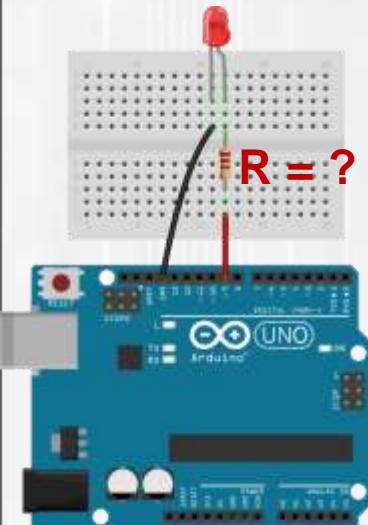
Caratteristiche led

Colore	VL
Rosso	1.8V
Giallo	1.9 V
Arancio	2.0 V
Verde	2.3 V
Azzurro	3.0 V
Blu	3.5 V
Con <b>IL</b> =5-20mA	

# Arduino Uno

## Programmazione in IDE...facciamo lampeggiare un LED!

Requisiti da tenere in conto (nel nostro caso) nella scelta della resistenza



$$IL = \frac{VAL - VL}{R} < 40 \text{ mA}$$

Product Description

Additional Information

### Technical Details

Microcontroller: ATmega328

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

For the tutorial of installing Arduino IDE, refer to WIKI

$$IL = \frac{VAL - VL}{R} < 20 \text{ mA}$$

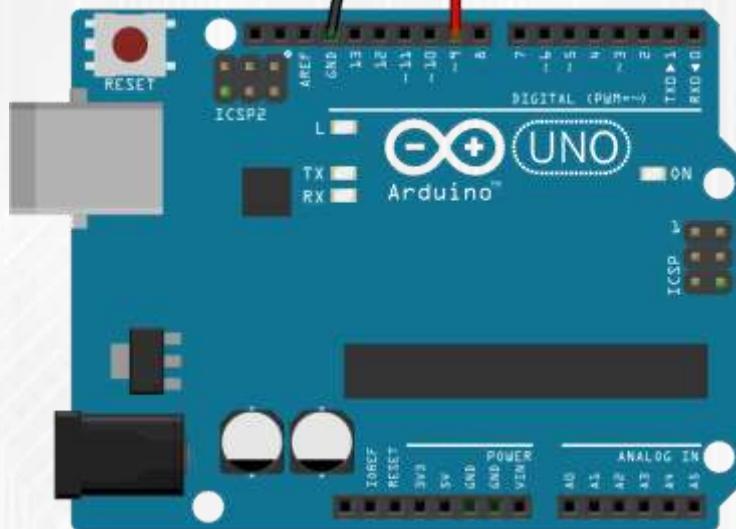
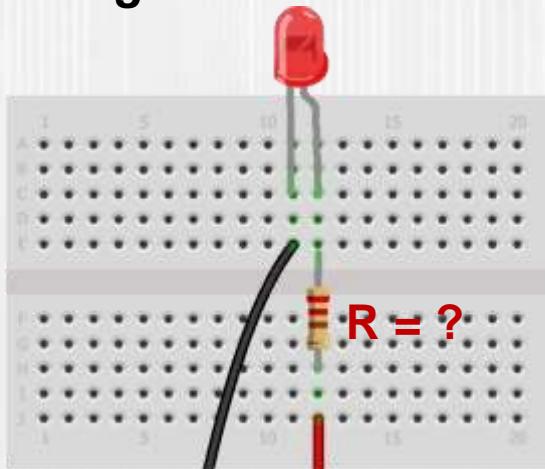
### Caratteristiche led

Colore	VL
Rosso	1.8V
Giallo	1.9 V
Verde	2.3 V

Con  $IL=20\text{mA}$

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**



Supposto che:

$$VAL = 5V$$

$$VL = 1.8 V \text{ (Led Rosso)}$$

$$IL = 20 mA$$

$$R = \frac{VAL - VL}{IL} = 160 \Omega$$

Quindi per non danneggiare il led:

$$R > 160 \Omega$$

$$R = \frac{VAL}{IL} \sim 250 \Omega$$

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);             // wait for half second
    digitalWrite(9, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);             // wait for half second
}
```

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT); ← Inizializza il pin digitale D9
}                                            come pin di uscita (saremo noi a impostarne il valore)

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);           // wait for half second
    digitalWrite(9, LOW);   // turn the LED off by making the voltage LOW
    delay(1000);           // wait for half second
}
```

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                   // wait for half second
    digitalWrite(9, LOW);          // turn the LED off by making the voltage LOW
    delay(1000);                   // wait for half second
}
```

Porta il valore del pin digitale D9 al livello logico alto (nel nostro caso 5 Volt)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for half second
    digitalWrite(9, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);              // wait for half second
}
```

Attende 1 secondo  
(tempo espresso in millisecondi)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);               // wait for half second
    digitalWrite(9, LOW);      // turn the LED off by making the voltage LOW
    delay(1000);               // wait for half second
}
```

Porta il valore del pin digitale D9  
al livello logico basso (0 Volt)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                  // wait for half second
    digitalWrite(9, LOW);           // turn the LED off by making the voltage LOW
    delay(1000);                  // wait for half second
}
```

Attende 1 secondo  
(tempo espresso in millisecondi)

# Arduino Uno

## GL5528 fotoresistore

LDR (Light Dependent Resistor) sono trasduttori sensibili alle radiazioni luminose, costituiti da materiale semiconduttore drogato con: Solfuro di Cadmio (CdS), Solfuro di Piombo (PbS), Selenio (Se) e Antimonio di Indio (InSb)

✓ Effetto fotoconduttivo, dove l'energia irradiante provoca la rottura dei legami covalenti nel semiconduttore, creando coppie lacune-elettroni e aumentando così la conducibilità del materiale, la cui resistenza varia al variare della radiazione luminosa che incide la superficie captante del trasduttore stesso

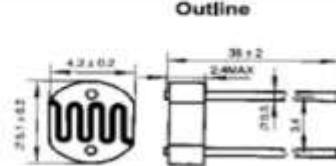
**CdS PHOTOCONDUCTIVE CELLS GL5528**



- ▲ Epoxy encapsulated
- ▲ Quick response
- ▲ Small size
- ▲ High sensitivity
- ▲ Reliable performance
- ▲ Good characteristic of spectrum

Light Resistance at 10Lux (at 25°C)	8~20kΩ
Dark Resistance at 0 Lux	1.0MΩ(min)
Gamma value at 100~1000lux	0.7
Power Dissipation(at 25°C)	100mW
Max Voltage (at 25°C)	150V
Spectral Response peak (at 25°C)	540nm
Ambient Temperature Range:	-30~+70°C

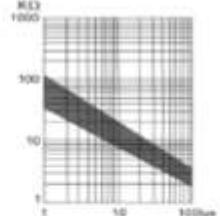
**Outline**



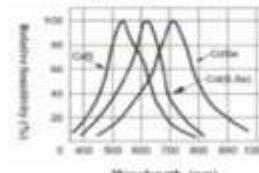
**Measuring Conditions**

1. Light Resistance:  
measured at 10 Lux with standard light A (2554c color temperature) and 2h pre-illumination at 400-500 lux prior to testing.
2. Dark Resistance:  
measured 10 seconds after pulsed 10 lux.
3. Gamma Characteristics:  
between 10 lux and 100 lux and given by:  
$$T = \frac{\log(R_{10}/R_{100})}{\log(10/1)} = \log(R_{10}/R_{100})$$
  
 $R_{10}, R_{100}$  cell resistance at 10 lux and 100 lux.  
The error of T is  $\pm 0.1$ .
4. Power:  
Max power dissipation at ambient temperature of 25°C.
5. Voltage:  
Max voltage in darkness that may be applied to the cell continuously.

**Illuminance Vs. Photo Resistance**



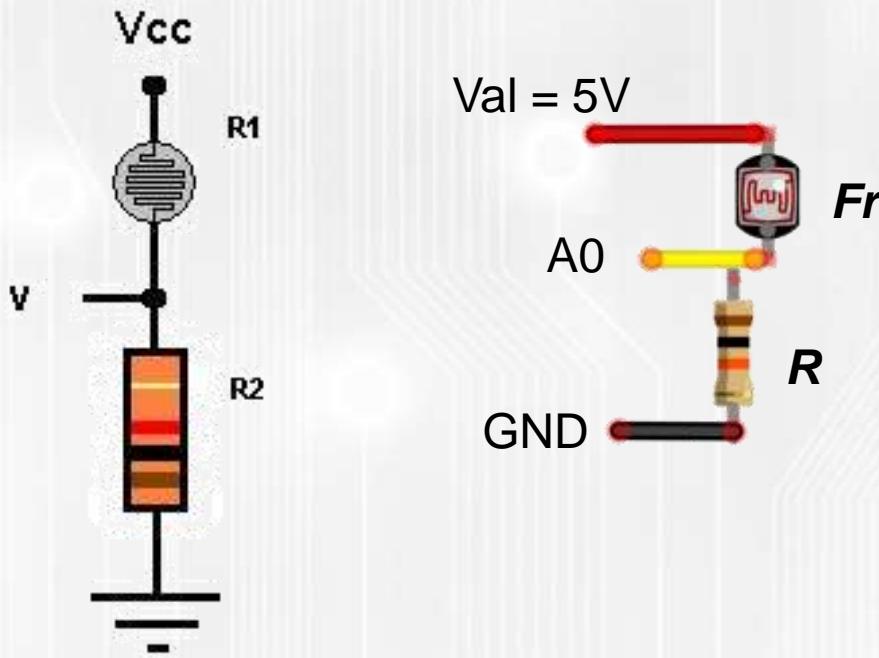
**Spectral Response**



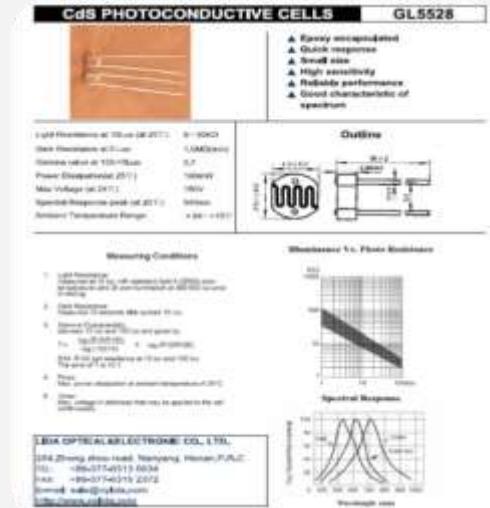
**LIDA OPTICAL&ELECTRONIC CO., LTD.**  
254,Zhong zhou road, Nanyang, Henan,P.R.C  
TEL: +86-377-6313 0034  
FAX: +86-377-6315 2372  
E-mail: sale@nylida.com  
<http://www.nylida.com>

# Arduino Uno

## GL5528 fotoresistore



La resistenza  $R_2$  ( $10K\Omega$ ) serve per creare un partitore di tensione. In base al valore che assumerà il Fotoresistore avremo, sul piedino della resistenza a cui è collegata, un valore che aumenta o diminuisce a seconda dell'intensità luminosa esterna.



$$V_{A0} = Val \cdot \frac{R}{R + Fr}$$

Un **incremento** della luminosità porta ad un incremento della tensione.

# Arduino Uno

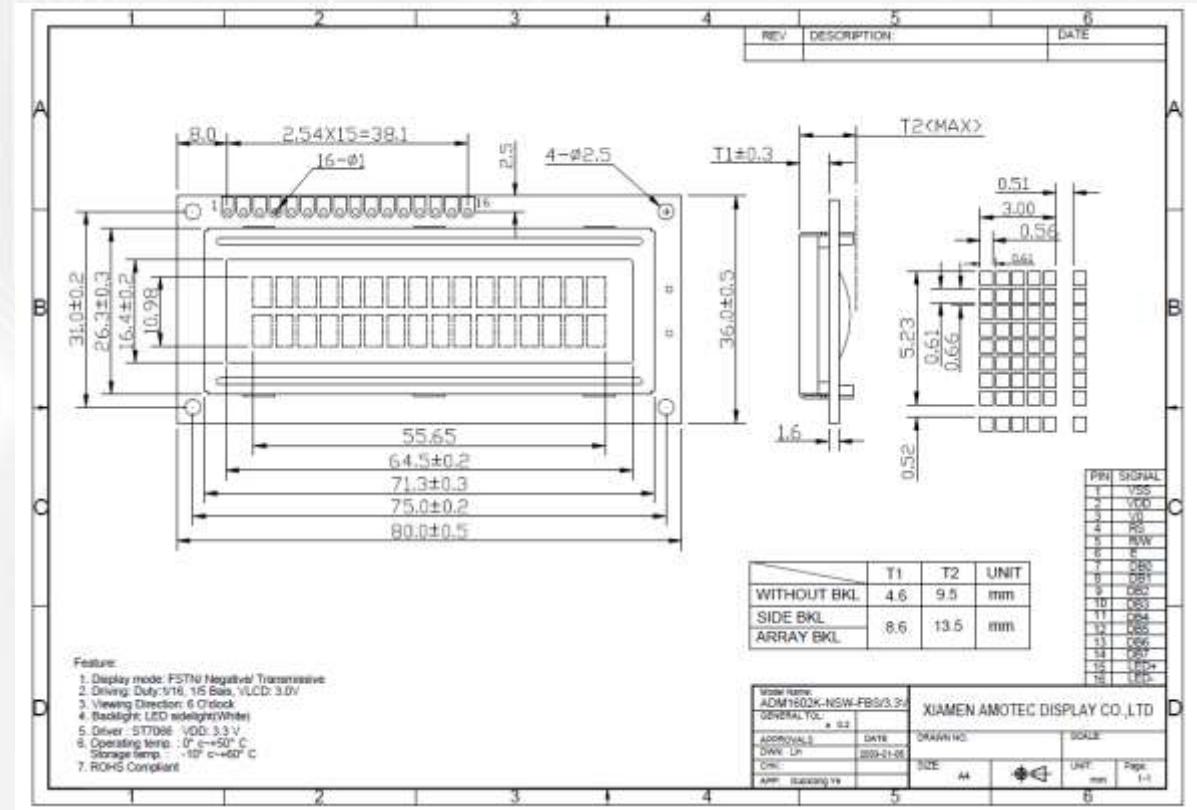
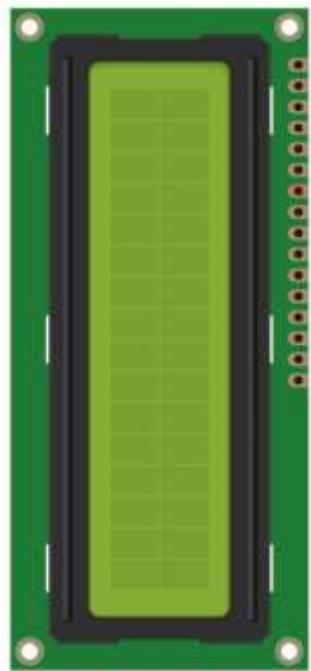
## DHT11 sensore di temperatura e umidità

- ✓ Per la misura della umidità utilizza il metodo resistivo
- ✓ Per la misura della temperatura utilizza un Negative Temperature Coefficient (NTC)
- ✓ Questi due sensori sono eccitati con un apposito circuito e la tensione da essi generata è convertita da un ADC ad 8 bit che permette di avere una risposta rapida e un segnale senza rumore.

Modello	DHT11
Alimentazione	3-5.5V DC
Segnale di uscita	digitale del segnale tramite single-bus
Elemento sensibile	Resistenza in Polimero
Campo di misura umidità	20-90% di umidità relativa, temperatura di 0-50 gradi Celsius
Precisione	umidità + -4% RH (Max + -5% di umidità relativa) temperatura +-2.0 Celsius
Risoluzione o la sensibilità	umidità 1% di umidità relativa, temperatura 0.1 Celsius
Ripetibilità umidità	+ -1% di umidità relativa temperatura +-1Celsius
Umidità isteresi	+ -1% RH
Stabilità a lungo termine	+ -0.5% UR / anno
Tempo di rilevazione	medio: 2s
Dimensioni	12 * 15,5 * 5,5 millimetri



# Arduino Uno LCD Module



# CORSO DI LABORATORIO DI ELETTRONICA PER L'AUTOMAZIONE



OPTOELECTRONICS group  
University of Sannio

## *Arduino: Esercitazione 1*

*Titolare del corso  
Prof. Andrea Cusano*

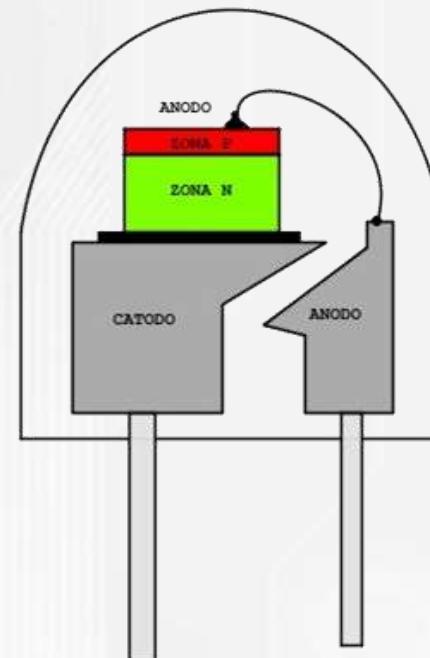
*Assistente:  
Ing. Patrizio Vaiano*

# Arduino Uno

## **Programmazione in IDE...facciamo lampeggiare un LED!**

*Il termine "LED" è un acronimo che sta per "Light Emitting Diode", ovvero "diodo che emette luce". I led sono costituiti da materiali in grado di emettere radiazioni luminose quando sono attraversati da una corrente elettrica. Per avere una luce apprezzabile senza distruggerlo il valore di corrente deve essere, per i piccoli LED, tra 5 e 20 mA circa.*

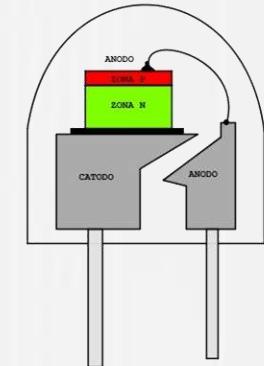
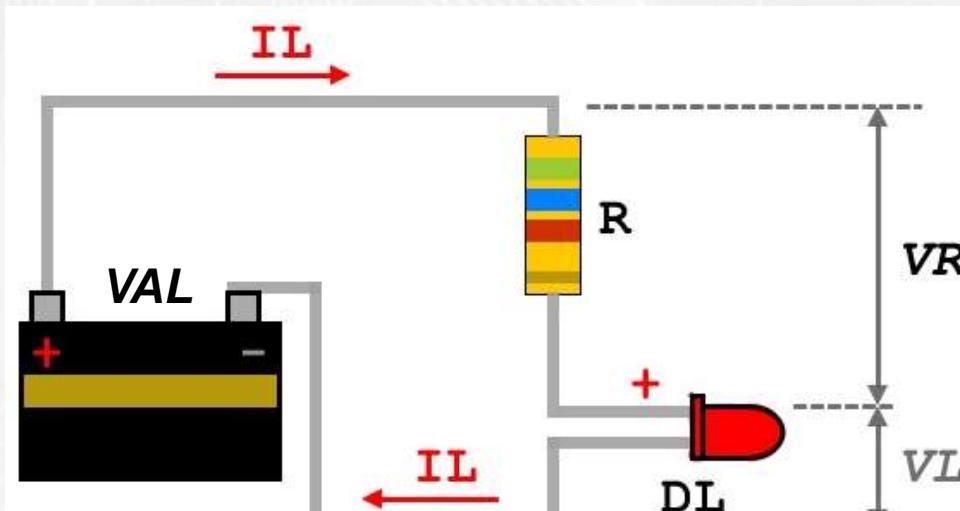
*I led hanno un terminale positivo chiamato anodo, ed uno negativo chiamato catodo, e per funzionare devono essere inseriti nel circuito rispettando tale polarità.*



# Arduino Uno

## Programmazione in IDE...facciamo lampeggiare un LED!

Quando si utilizza un led (**DL**), è necessario disporre un'opportuna resistenza (**R**) in serie ad esso, allo scopo di limitare la corrente (**IL**) che passa ed evitare che possa distruggersi; la caduta di tensione ai capi di un led (**VL**) può variare da 1 a 3.5 V circa, in funzione della lunghezza d'onda della radiazione emessa, ossia del colore.



Caratteristiche led	
Colore	<b>VL</b>
Rosso	1.8V
Giallo	1.9 V
Arancio	2.0 V
Verde	2.3 V
Azzurro	3.0 V
Blu	3.5 V

Con **IL**=5-20mA

# Arduino Uno

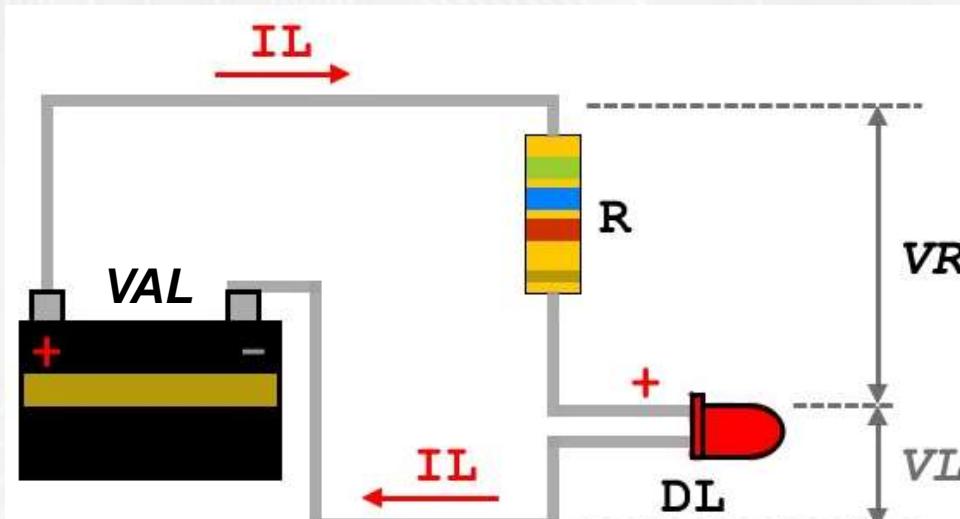
**Programmazione in IDE...facciamo lampeggiare un LED!**

$$VAL = VL + VR$$



$$IL = \frac{VAL - VL}{R}$$

$$VR = IL \times R$$



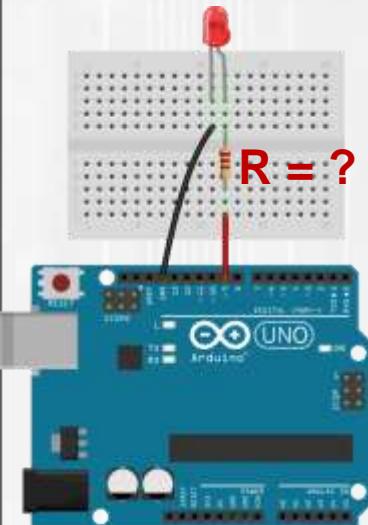
Caratteristiche led

Colore	<b>VL</b>
Rosso	1.8V
Giallo	1.9 V
Arancio	2.0 V
Verde	2.3 V
Azzurro	3.0 V
Blu	3.5 V
Con <b>IL</b> =5-20mA	

# Arduino Uno

## Programmazione in IDE...facciamo lampeggiare un LED!

Requisiti da tenere in conto (nel nostro caso) nella scelta della resistenza



$$IL = \frac{VAL - VL}{R} < 40 \text{ mA}$$

Product Description

Additional Information

### Technical Details

Microcontroller: ATmega328

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

For the tutorial of installing Arduino IDE, refer to WIKI

$$IL = \frac{VAL - VL}{R} < 20 \text{ mA}$$

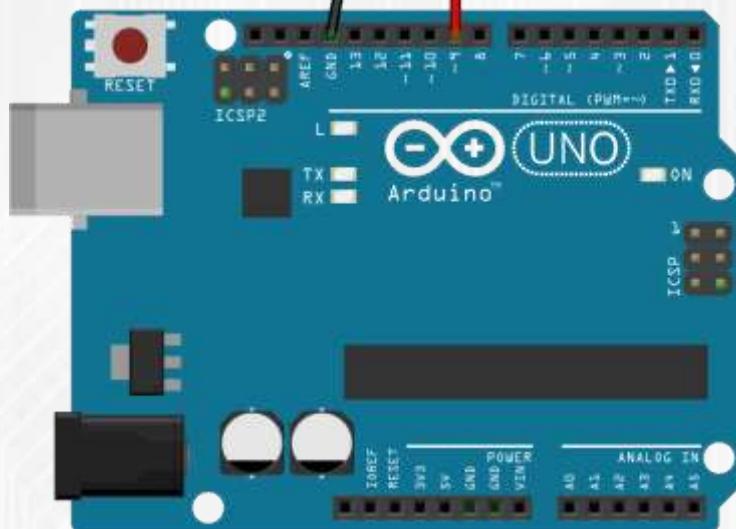
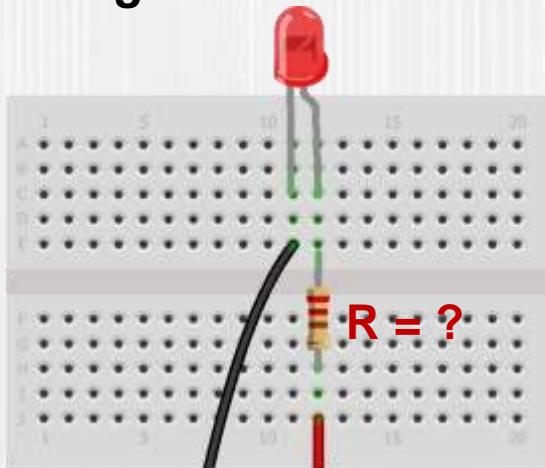
### Caratteristiche led

Colore	VL
Rosso	1.8V
Giallo	1.9 V
Verde	2.3 V

Con  $IL=20\text{mA}$

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**



Supposto che:

$$VAL = 5V$$

$$VL = 1.8 V \text{ (Led Rosso)}$$

$$IL = 20 mA$$

$$R = \frac{VAL - VL}{IL} = 160 \Omega$$

Quindi per non danneggiare il led:

$$R > 160 \Omega$$

$$R = \frac{VAL}{IL} \sim 250 \Omega$$

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);             // wait for half second
    digitalWrite(9, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);             // wait for half second
}
```

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT); ← Inizializza il pin digitale D9
}                                            come pin di uscita (saremo noi a impostarne il valore)

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);           // wait for half second
    digitalWrite(9, LOW);   // turn the LED off by making the voltage LOW
    delay(1000);           // wait for half second
}
```

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                   // wait for half second
    digitalWrite(9, LOW);          // turn the LED off by making the voltage LOW
    delay(1000);                   // wait for half second
}
```

Porta il valore del pin digitale  
D9 al livello logico alto  
(nel nostro caso 5 Volt)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for half second
    digitalWrite(9, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);              // wait for half second
}
```

Attende 1 secondo  
(tempo espresso in millisecondi)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);               // wait for half second
    digitalWrite(9, LOW);      // turn the LED off by making the voltage LOW
    delay(1000);               // wait for half second
}
```

Porta il valore del pin digitale D9  
al livello logico basso (0 Volt)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                  // wait for half second
    digitalWrite(9, LOW);           // turn the LED off by making the voltage LOW
    delay(1000);                  // wait for half second
}
```

Attende 1 secondo  
(tempo espresso in millisecondi)

# Arduino Uno

**Programmazione in IDE...facciamo lampeggiare un LED!**

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 9 as an output pin
    pinMode(9, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(9, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);               // wait for half second
    digitalWrite(9, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);               // wait for half second
}
```

Su quali istruzioni bisogna intervenire per modificare la frequenza di lampeggiamento?  
Qual è il duty-cycle? Come si modifica?

# Arduino Uno

## GL5528 fotoresistore

LDR (Light Dependent Resistor) sono trasduttori sensibili alle radiazioni luminose, costituiti da materiale semiconduttore drogato con: Solfuro di Cadmio (CdS), Solfuro di Piombo (PbS), Selenio (Se) e Antimonio di Indio (InSb)

✓ Effetto fotoconduttivo, dove l'energia irradiante provoca la rottura dei legami covalenti nel semiconduttore, creando coppie lacune-elettroni e aumentando così la conducibilità del materiale, la cui resistenza varia al variare della radiazione luminosa che incide la superficie captante del trasduttore stesso

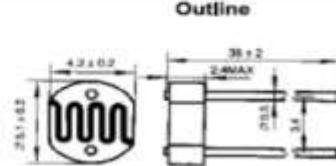
**CdS PHOTOCONDUCTIVE CELLS GL5528**



- ▲ Epoxy encapsulated
- ▲ Quick response
- ▲ Small size
- ▲ High sensitivity
- ▲ Reliable performance
- ▲ Good characteristic of spectrum

Light Resistance at 10Lux (at 25°C)	8~20kΩ
Dark Resistance at 0 Lux	1.0MΩ(min)
Gamma value at 100~1000lux	0.7
Power Dissipation(at 25°C)	100mW
Max Voltage (at 25°C)	150V
Spectral Response peak (at 25°C)	540nm
Ambient Temperature Range:	-30~+70°C

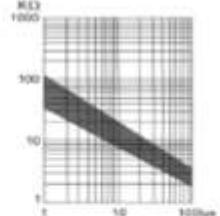
**Outline**



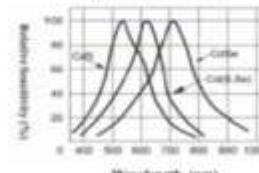
**Measuring Conditions**

1. Light Resistance:  
measured at 10 Lux with standard light A (2554c color temperature) and 2h pre-illumination at 400-500 lux prior to testing.
2. Dark Resistance:  
measured 10 seconds after pulsed 10 lux.
3. Gamma Characteristic:  
between 10 lux and 100 lux and given by:  
$$T = \frac{\log(R_{10}/R_{100})}{\log(10/10)} = \log(R_{10}/R_{100})$$
  
 $R_{10}, R_{100}$  cell resistance at 10 lux and 100 lux.  
The error of T is  $\pm 0.1$ .
4. Power:  
Max power dissipation at ambient temperature of 25°C.
5. Voltage:  
Max voltage in darkness that may be applied to the cell continuously.

**Illuminance Vs. Photo Resistance**



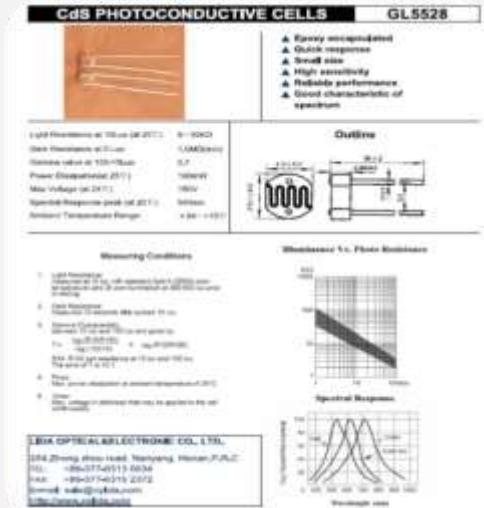
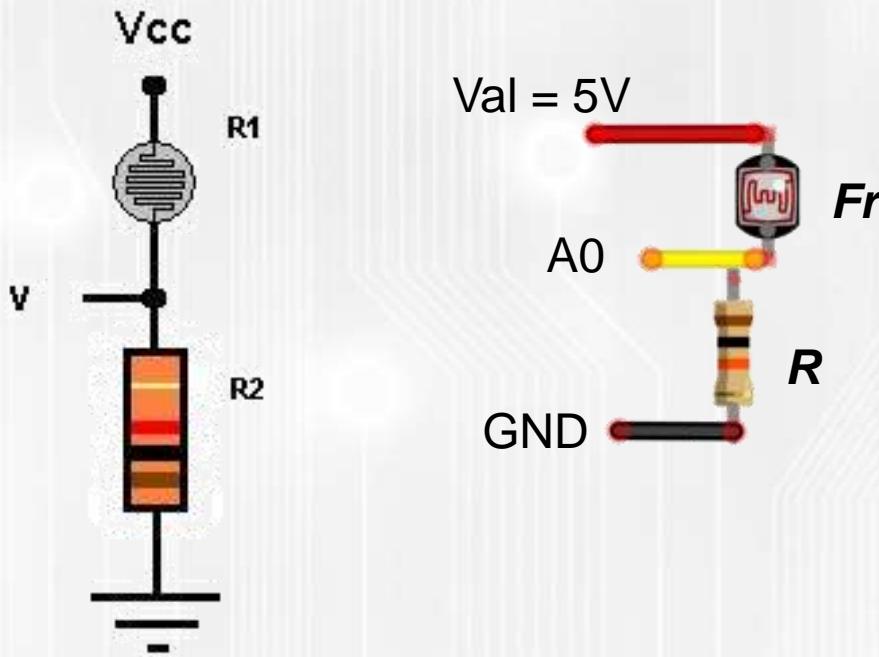
**Spectral Response**



**LIDA OPTICAL&ELECTRONIC CO., LTD.**  
254,Zhong zhou road, Nanyang, Henan,P.R.C  
TEL: +86-377-6313 0034  
FAX: +86-377-6315 2372  
E-mail: sale@nylida.com  
<http://www.nylida.com>

# Arduino Uno

## GL5528 fotoresistore



$$V_{A0} = Val \cdot \frac{R}{R + Fr}$$

La resistenza  $R_2$  ( $10K\Omega$ ) serve per creare un partitore di tensione. In base al valore che assumerà il Fotoresistore avremo, sul piedino della resistenza a cui è collegata, un valore che aumenta o diminuisce a seconda dell'intensità luminosa esterna.

Un **incremento** della luminosità porta ad un incremento della tensione.

# Arduino Uno

**...accendiamo un LED se si spegne la luce nella stanza!**

```
int luce; // Dichiara la variabile "luce", che indicherà i valori da 0 a  
1023 che saranno forniti dal pin analogico A0  
  
#define SOGLIA 600 // imposta una soglia costante (dipende dall'ambiente)  
  
void setup() {  
    // pin digitale indicato come output collegato al LED  
    pinMode(12,OUTPUT);  
    // inizializza la comunicazione seriale tra PC ed Arduino  
    Serial.begin(9600); //9600 è il baud rate, cioè la velocità di  
trasmissione dati, dipende dal dispositivo  
}  
  
void loop() {  
// . . .
```

# Arduino Uno

**...accendiamo un LED se si spegne la luce nella stanza!**

```
int luce; // Dichiara la variabile "luce", che indicherà i valori da 0 a  
1023 che saranno forniti dal pin analogico A0
```

```
#define SOGLIA 600 // imposta una soglia costante (dipende dall'ambiente)
```

```
void setup() {  
    // pin digitale indicato con  
    pinMode(12,OUTPUT);  
    // inizializza la comunicazione seriale tra PC ed Arduino  
    Serial.begin(9600); //9600 è il baud rate, cioè la velocità di  
    trasmissione dati, dipende dal dispositivo  
}
```

```
void loop() {  
// . . .
```

Si può in alternativa dichiarare  
una variabile int(era)

# Arduino Uno

**...accendiamo un LED se si spegne la luce nella stanza!**

```
int luce; // Dichiara la variabile "luce", che indicherà i valori da 0 a  
1023 che saranno forniti dal pin analogico A0

#define SOGLIA 600 // imposta una soglia costante (dipende dall'ambiente)

void setup() {  
    // pin digitale indicato come output collegato al LED  
    pinMode(12,OUTPUT);  
    // inizializza la comunicazione seriale tra PC ed Arduino  
    Serial.begin(9600); //9600 è il baud rate, cioè la velocità di  
// trasmissione dati, dipende dal dispositivo  
}  
  
void loop() {  
// . . .
```

Serve per usare il monitor seriale

# Arduino Uno

**...accendiamo un LED se si spegne la luce nella stanza!**

```
void loop() {
    // assegna al pin A0 il compito di leggere i valori d'ingresso (nodo tra
    FR e R)
    luce=analogRead(A0); // controllo se la tensione su A0 è al di sotto di un certo valore di
    soglia
    if(luce<SOGLIA){
        digitalWrite(12, HIGH); // d
    } else {
        digitalWrite(12, LOW);
    }
    // stampa il valore di tensione letto da A0 sul monitor seriale
    Serial.println(luce);
    // inserisce un ritardo tra due iterazioni successive
    delay(1000);
}
```

La tensione analogica (0-5V) è convertita in digitale (0-1023)

# Arduino Uno

**...accendiamo un LED se si spegne la luce nella stanza!**

```
void loop() {
    // assegna al pin A0 il compito di leggere i valori d'ingresso (nodo tra
    FR e R)
    luce=analogRead(A0);
    // controlla se la tensione su A0 è al di sotto di un certo valore di
    soglia
    if(luce<SOGLIA){
        digitalWrite(12, HIGH); // dà al pin D12 valore logico alto (LED acceso)
    } else {
        digitalWrite(12, LOW);
    }
    // stampa il valore di tensione letto da A0 sul monitor seriale
    Serial.println(luce);
    // inserisce un ritardo tra due iterazioni successive
    delay(1000);
}
```

Stampa sul monitor seriale la tensione  
letta da A0 convertita in digitale (0-1023)

# Arduino Uno

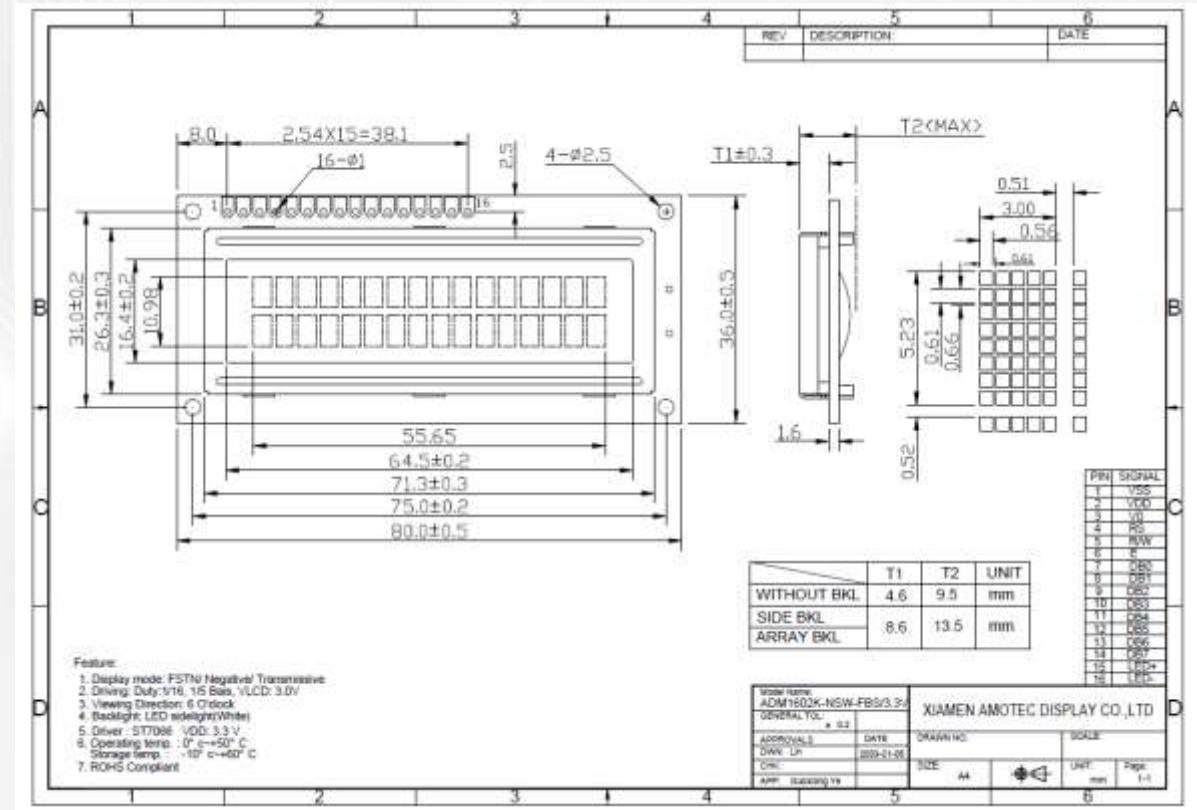
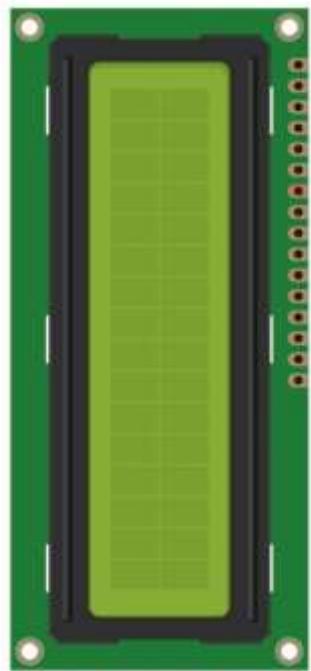
## DHT11 sensore di temperatura e umidità

- ✓ Per la misura della umidità utilizza il metodo resistivo
- ✓ Per la misura della temperatura utilizza un Negative Temperature Coefficient (NTC)
- ✓ Questi due sensori sono eccitati con un apposito circuito e la tensione da essi generata è convertita da un ADC ad 8 bit che permette di avere una risposta rapida e un segnale senza rumore.

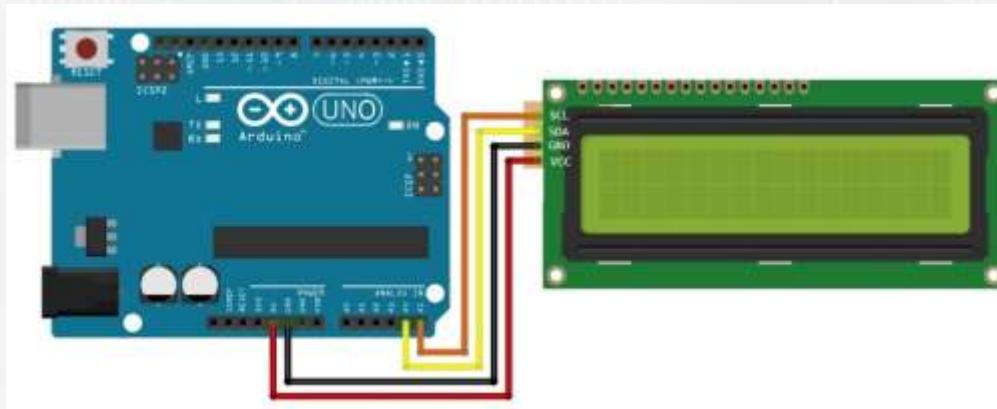
Modello	DHT11
Alimentazione	3-5.5V DC
Segnale di uscita	digitale del segnale tramite single-bus
Elemento sensibile	Resistenza in Polimero
Campo di misura umidità	20-90% di umidità relativa, temperatura di 0-50 gradi Celsius
Precisione	umidità + -4% RH (Max + -5% di umidità relativa) temperatura +-2.0 Celsius
Risoluzione o la sensibilità	umidità 1% di umidità relativa, temperatura 0.1 Celsius
Ripetibilità umidità	+ -1% di umidità relativa temperatura +-1Celsius
Umidità isteresi	+ -1% RH
Stabilità a lungo termine	+ -0.5% UR / anno
Tempo di rilevazione	medio: 2s
Dimensioni	12 * 15,5 * 5,5 millimetri



# Arduino Uno LCD Module



# *Arduino Uno LCD Module*



Comunicazione via bus I<sup>2</sup>C

GND	GND
5V	VCC
A4	SDA (serial data line)
A5	SCL (serial clock line)

# *Arduino Uno*

## *DHT11 sensore di temperatura e umidità*

Inclusione delle librerie  
&  
Inizializzazioni

```
#include <LiquidCrystal_I2C.h>

#include <DHT.h> // oppure "DHT.h"

// Oggetto della libreria DHT.h, il primo parametro tra parentesi
// rappresenta il pin a cui è collegato il sensore, il secondo parametro indica
// il tipo di sensore usato

#define DHTPIN 2      // digital pin 2
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);

// Oggetto della classe lcd
LiquidCrystal_I2C lcd(0x27, 16, 2); // indirizzo iniziale, 16 caratteri per
// riga, 2 righe
```

# *Arduino Uno*

## *DHT11 sensore di temperatura e umidità*

```
void setup() {  
  
    lcd.init(); // Inizializza LCD  
    lcd.backlight(); // attiva la retro illuminazione  
  
    dht.begin(); // Inizializza il sensore DHT  
  
    Serial.begin(9600); // inizializza la comunicazione seriale  
  
    delay(1000);  
}  
  
void loop() {
```

Inizializzazione del  
LCD, DHT e seriale

# Arduino Uno

## DHT11 sensore di temperatura e umidità

```
void loop() {  
  
    // Salva la temperatura nella variabile float t  
    float t = dht.readTemperature();  
    // Legge la temperatura in ° Fahrenheit (isFahrenheit = true)  
    // float f = dht.readTemperature(true);  
    // Salva l'umidità relativa nella variabile float h  
    float h = dht.readHumidity();  
  
    // Controlla se una lettura di h o t è fallita  
    if (isnan(h) || isnan(t)) {  
        Serial.println(F("Mancata lettura!"));  
    }  
  
    Serial.print("T=" + String(t) + "°C\n");  
    //Serial.println(t);  
    //Serial.print(F("°C\n"));  
    Serial.print(F("RH="));  
    Serial.print(h);  
    Serial.print(F("%\n"));  
}
```

Lettura di temperatura  
e umidità

# *Arduino Uno*

## *DHT11 sensore di temperatura e umidità*

```
void loop() {  
  
    // Salva la temperatura nella variabile float t  
    float t = dht.readTemperature();  
    // Legge la temperatura in ° Fahrenheit (isFahrenheit = true)  
    // float f = dht.readTemperature(true);  
    // Salva l'umidità relativa nella variabile float h  
    float h = dht.readHumidity();  
  
    // Controlla se una lettura di h o t è fallita  
    if (isnan(h) || isnan(t)) {  
        Serial.println(F("Mancata lettura!"));  
    }  
  
    Serial.print("T=" + String(t) + "°C\n");  
    //Serial.println(t);  
    //Serial.print(F("°C\n"));  
    Serial.print(F("RH="));  
    Serial.print(h);  
    Serial.print(F("%\n"));
```

Avverte se una lettura  
è fallita

# *Arduino Uno*

## *DHT11 sensore di temperatura e umidità*

```
void loop() {  
  
    // Salva la temperatura nella variabile float t  
    float t = dht.readTemperature();  
    // Legge la temperatura in ° Fahrenheit (isFahrenheit = true)  
    // float f = dht.readTemperature(true);  
    // Salva l'umidità relativa nella variabile float h  
    float h = dht.readHumidity();  
  
    // Controlla se una lettura di h o t è fallita  
    if (isnan(h) || isnan(t)) {  
        Serial.println(F("Mancata lettura!"));  
    }  
  
    Serial.print("T=" + String(t) + "°C\n");  
    //Serial.println(t);  
    //Serial.print(F("°C\n"));  
    Serial.print(F("RH="));  
    Serial.print(h);  
    Serial.print(F("%\n"));
```

Stampa su monitor seriale

# *Arduino Uno*

## *DHT11 sensore di temperatura e umidità*

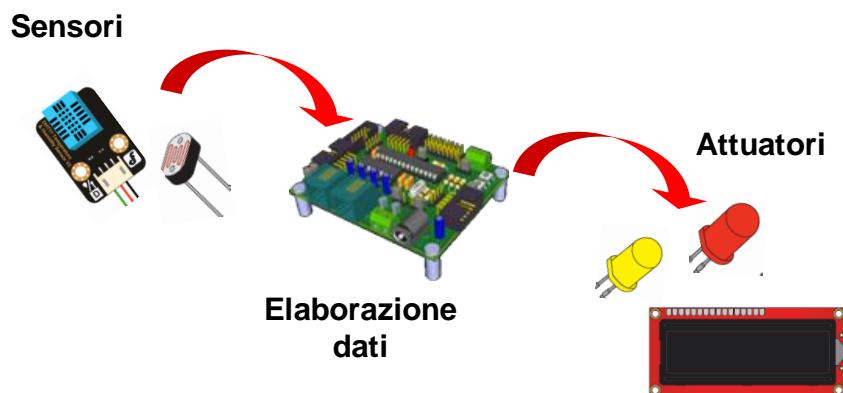
Stampa su LCD

```
// pulisce lo schermo  
lcd.clear();  
  
lcd.setCursor(0, 0); // part dal primo carattere  
  
// mostra la temperatura su LCD  
lcd.print("Temp. = " + String(t) + char(0xDF) + "C");  
  
// mostra l'umidità relativa su LCD  
lcd.setCursor(0, 1);  
lcd.print("RH = " + String(h) + " %");  
delay(1000);  
}
```

## ESERCITAZIONE ARDUINO

Scopo dell'esercitazione è realizzare sistemi che siano in grado di:

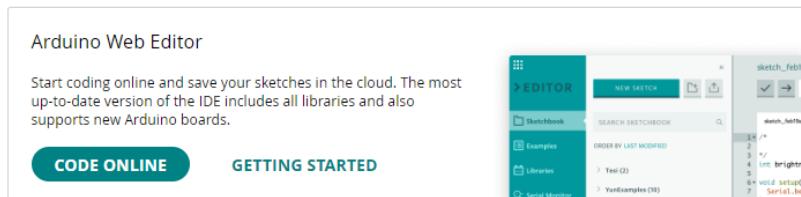
- 1) Sentire variazioni di parametri esterni e convertirle in variazioni di segnali elettrici (sensori);
- 2) Elaborare il segnale elettrico proveniente dai sensori e generare segnali in uscita (elaborazione dati);
- 3) Convertire il segnale elettrico in uscita dallo stato precedente in una particolare grandezza fisica (attuatori).



Per fare ciò una valida strategia che permette di elaborare via codice i segnali provenienti dai sensori e attivare opportuni attuatori è rappresentata da ARDUINO.

## Software ARDUINO IDE 2.2.1

- Step1: scaricare il software



## Downloads

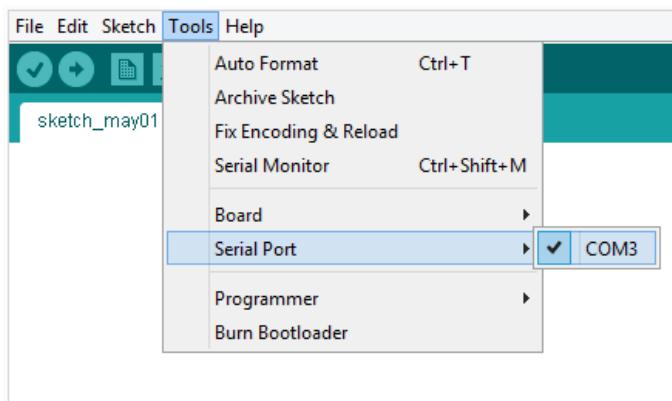
The page features a large image of the Arduino IDE logo and the text 'Arduino IDE 2.2.1'. Below this, there's a brief description of the release, a link to the documentation, and a note about nightly builds. On the right side, there's a 'DOWNLOAD OPTIONS' section with links for Windows, Linux, macOS, and source code.

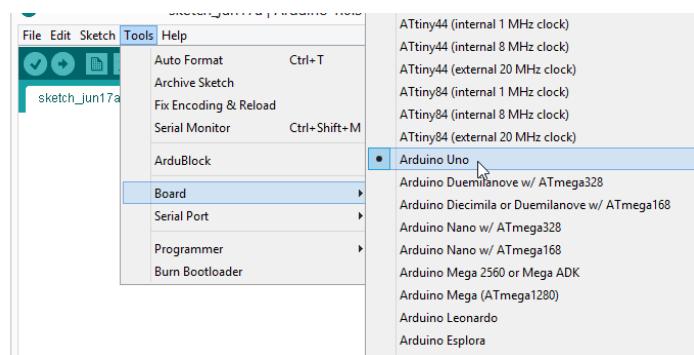
**DOWNLOAD OPTIONS**

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** Intel, 10.14: "Mojave" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

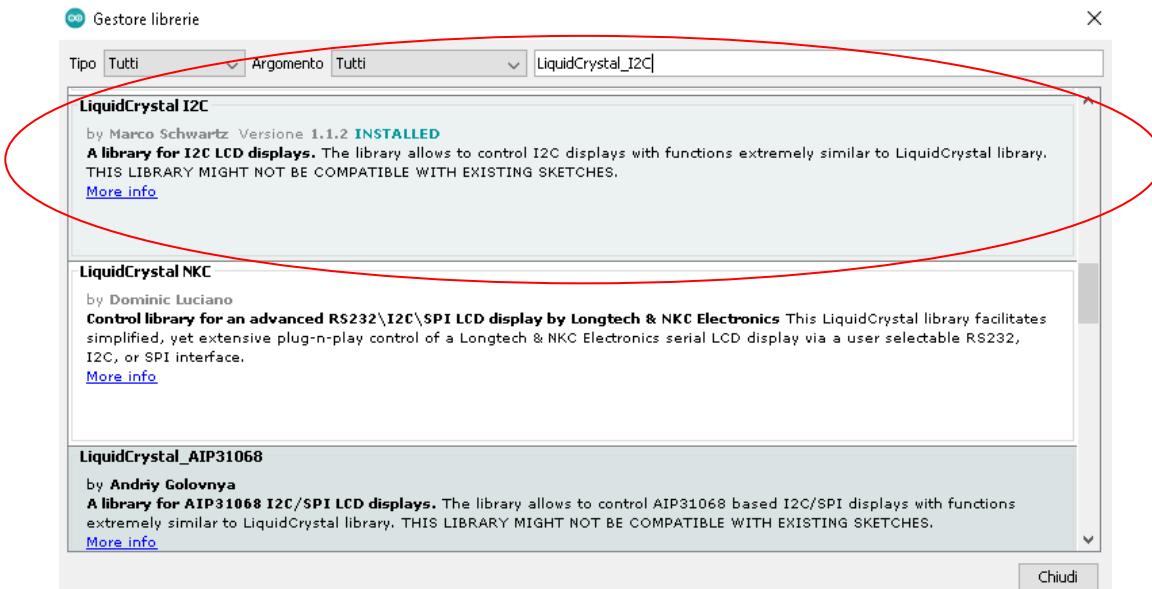
[Release Notes](#)

- Step 2: impostare la porta di comunicazione e il modello ARDUINO UNO che si sta utilizzando

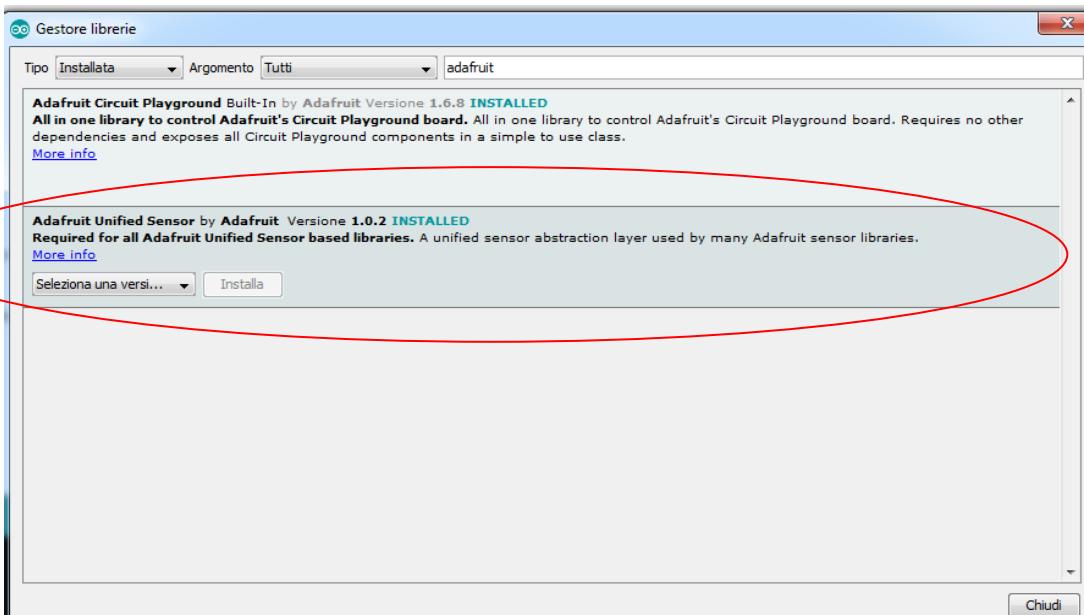
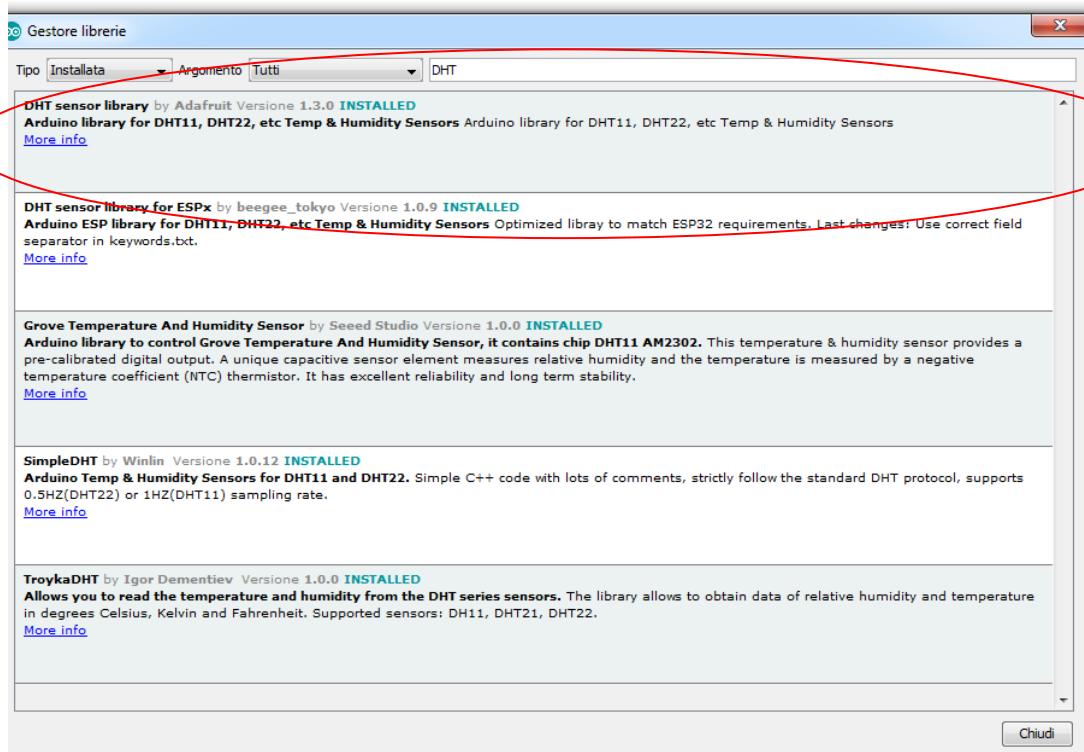




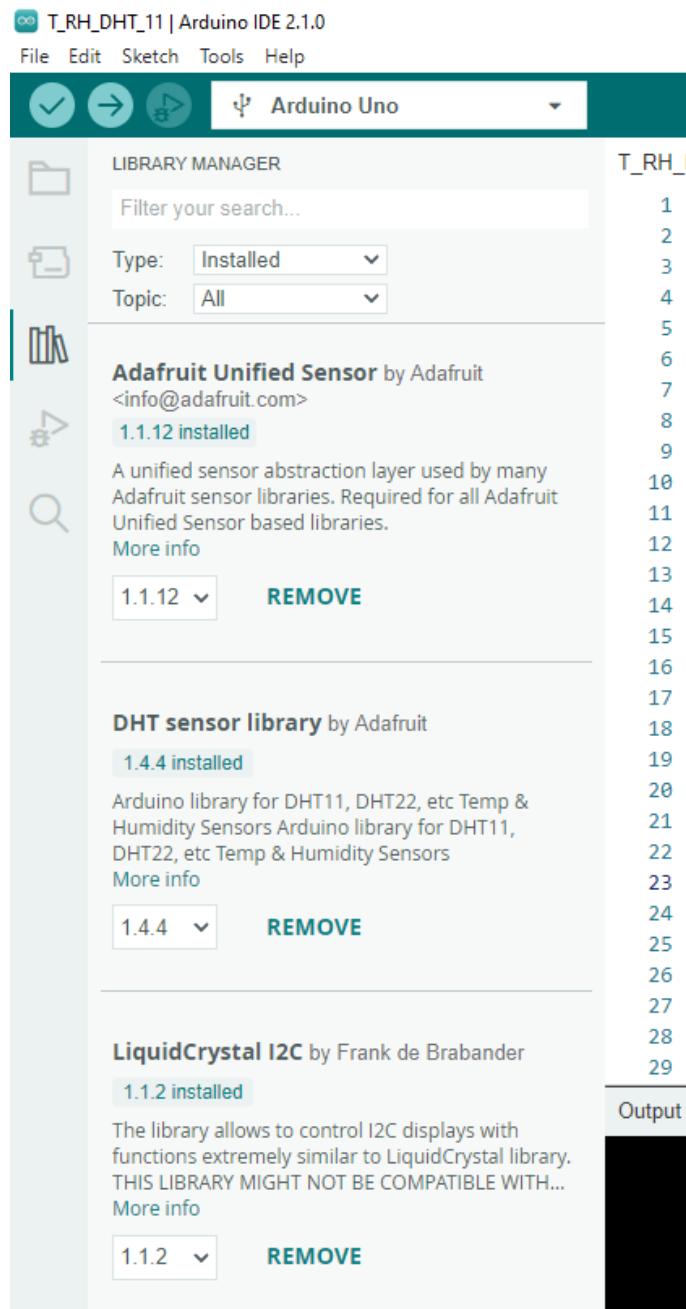
- Step 3: installazione delle librerie per utilizzare LCD (LiquidCrystal\_I2C) dalla sezione Strumenti/Gestione librerie.



- Step 4: installazione delle librerie per utilizzare DHT (DHT sensor library) e Adafruit Unified Sensor.



Se si utilizza la versione 2.1.0 o successive, le librerie installate / da installare sono accessibili nella parte sinistra della finestra.



## Lampeggiamento LED

Scopo di questa esercitazione è realizzare un sistema che faccia lampeggiare un led rosso.

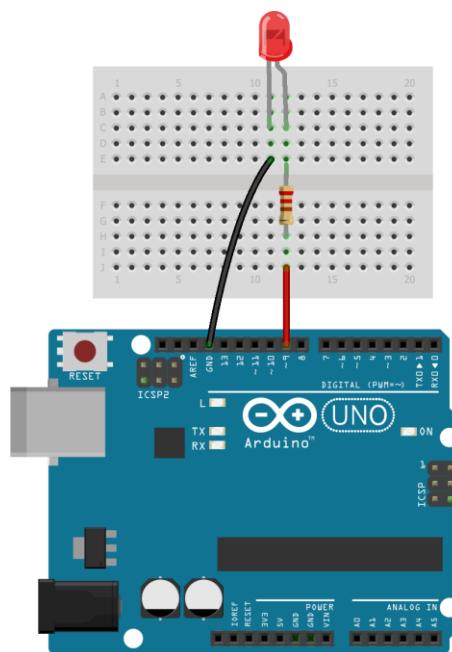
Strumentazione da laboratorio necessaria:

- Arduino UNO
- LED rosso
- Resistenza 220 Ω
- Cavetteria
- Bread-board

Testare 2 versioni differenti:

1. Il LED da far lampeggiare è integrato nella scheda (L).
2. Il LED da far lampeggiare è esterno alla scheda.

Cosa accadrebbe se si utilizzasse una resistenza molto inferiore a 220 Ω? Cosa accadrebbe invece se si usasse una resistenza molto maggiore di 220 Ω?



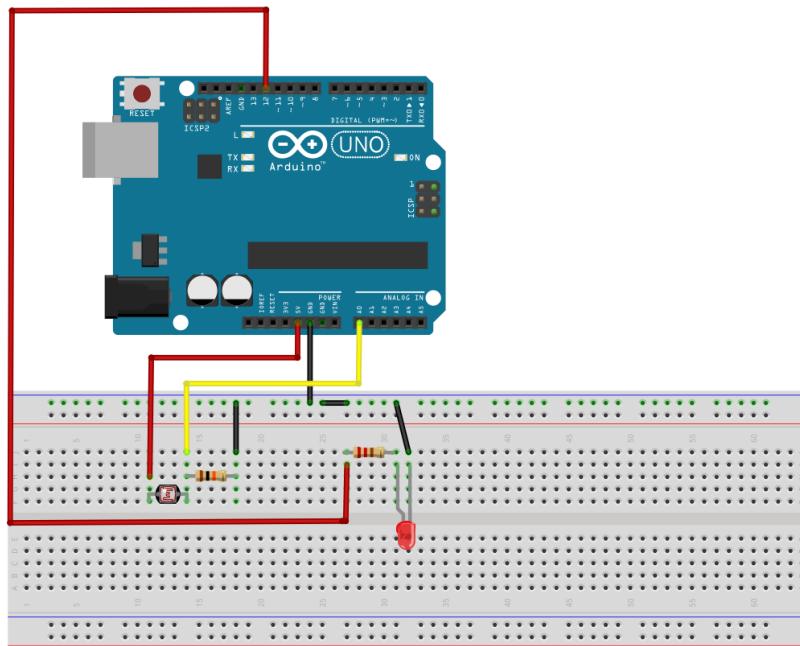
## Sistema automatico: Fotoresistore-LED

Scopo di questa esercitazione è realizzare un sistema che attivi un LED in caso di assenza di luce all'interno di un ambiente. In particolare, un fotoresistore rileverà la presenza o assenza di luce e il LED di sicurezza si attiverà nel momento in cui il fotoresistore sarà oscurato (impostare una soglia adeguata).

Strumentazione da laboratorio necessaria:

- Arduino UNO
- Fotoresistore
- LED rosso
- Resistenza 10 kΩ
- Resistenza 220 Ω
- Cavetteria
- Bread-board

Suggerimento: disegnare lo schema del circuito e riflettere sul partitore di tensione realizzato.

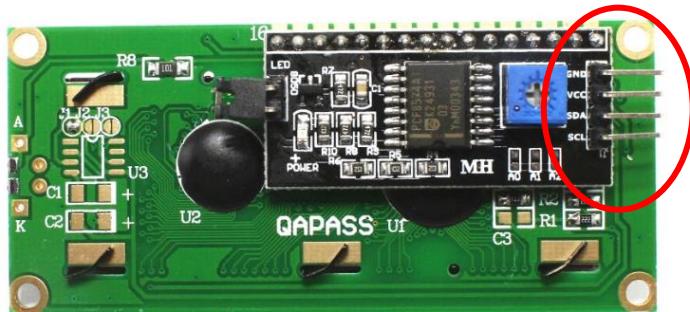
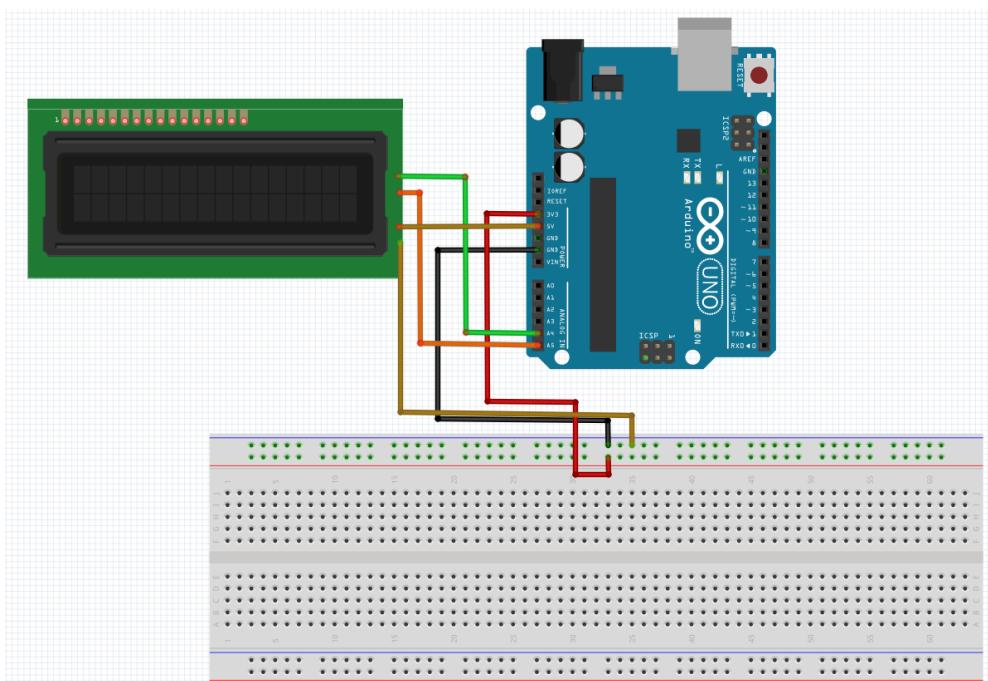


## Visualizzazione su display LCD

Scopo di questa esercitazione è realizzare un sistema che faccia comparire su un display LCD la scritta: "Ciao mondo". Servirsi della comunicazione I<sup>2</sup>C avendo cura di collegare SDA con il pin analogico A4 e SCL con A5.

Strumentazione da laboratorio necessaria:

- Arduino UNO
- Display LCD
- Cavetteria

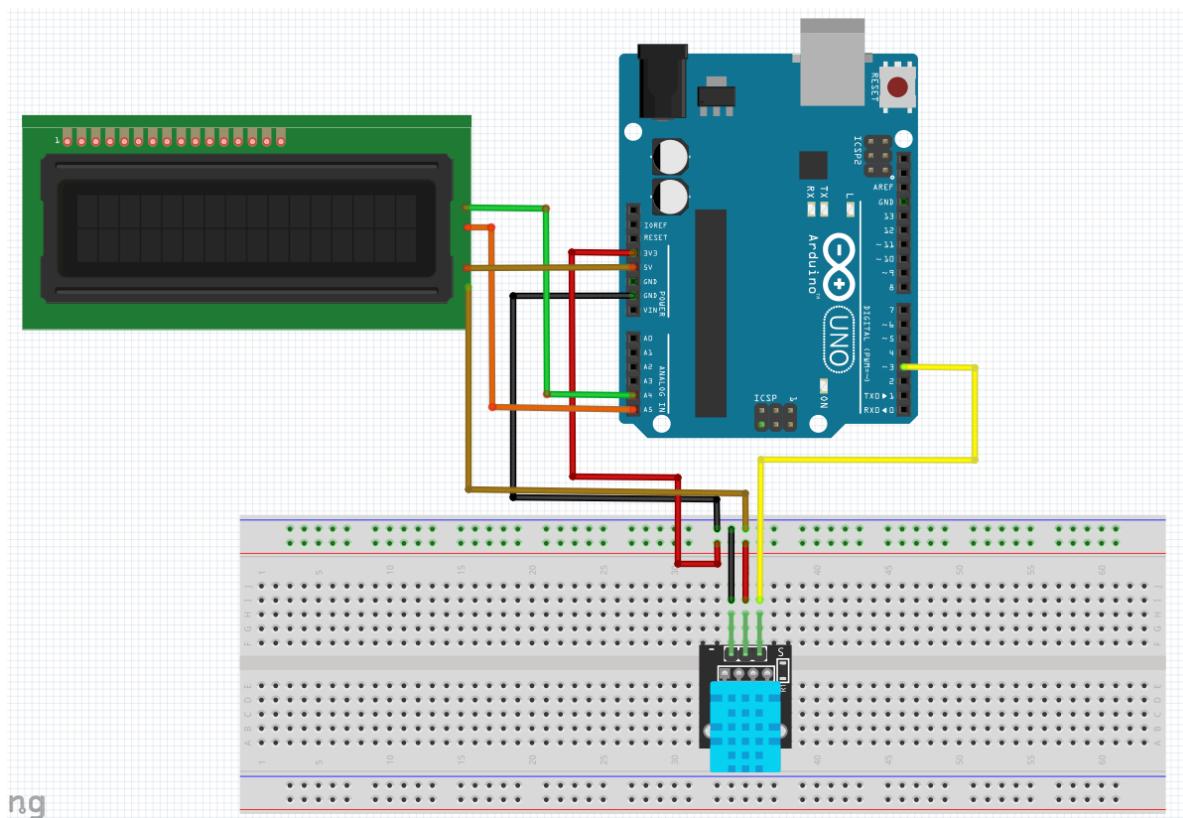


## Sistema per il controllo della temperatura

Scopo di questa esercitazione è realizzare un sistema che visualizzi a su display LCD la temperatura e l'umidità rilevata dal sensore DHT11. Prevedere la visualizzazione dei valori di temperatura e umidità relativa anche sul monitor seriale.

Strumentazione da laboratorio necessaria:

- Arduino UNO
- DHT11
- Display LCD
- Cavetteria



# *Corso di Laboratorio di Elettronica per l'Automazione*

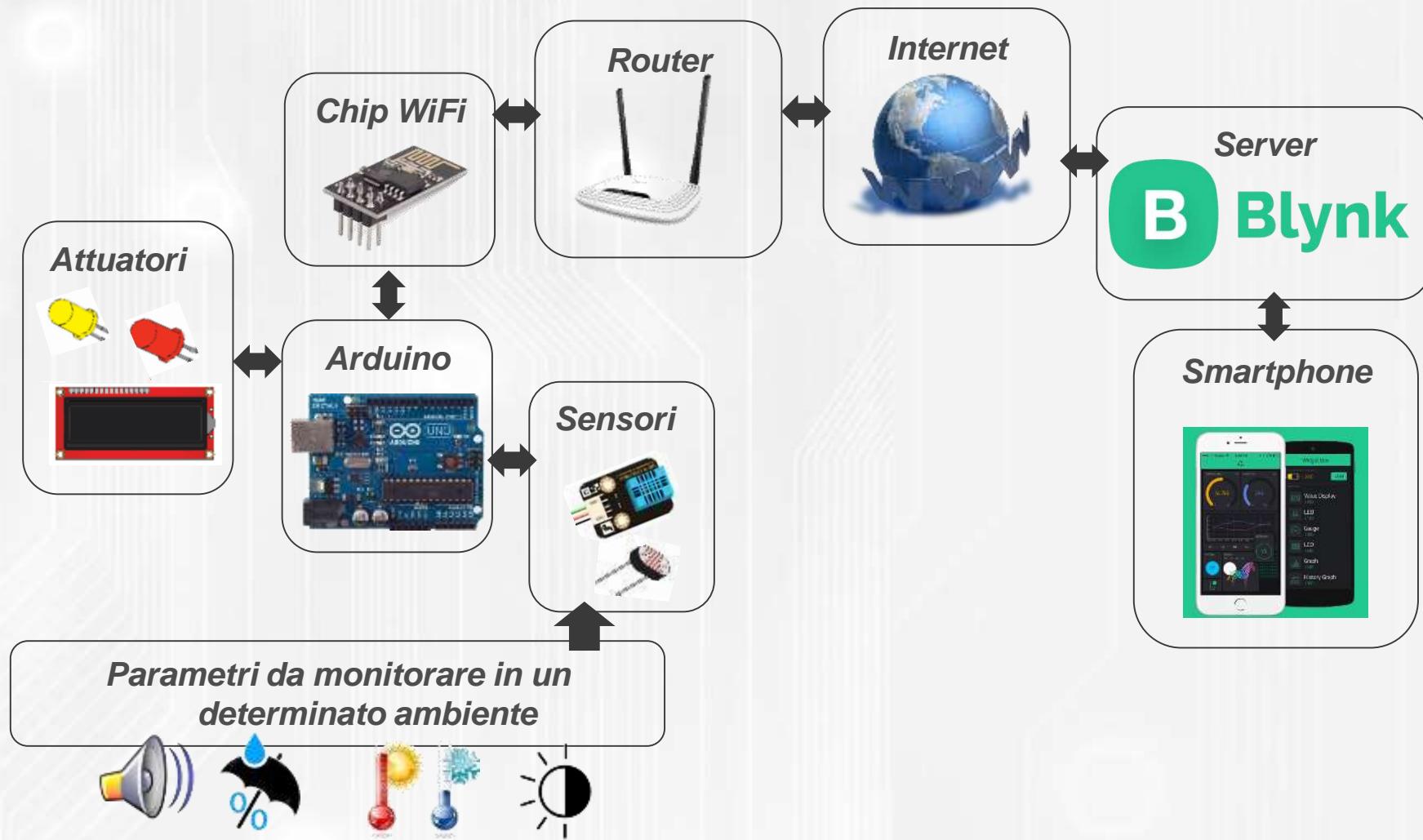


*Controllo sensori e attivazione attuatori  
da remoto con ARDUINO UNO*

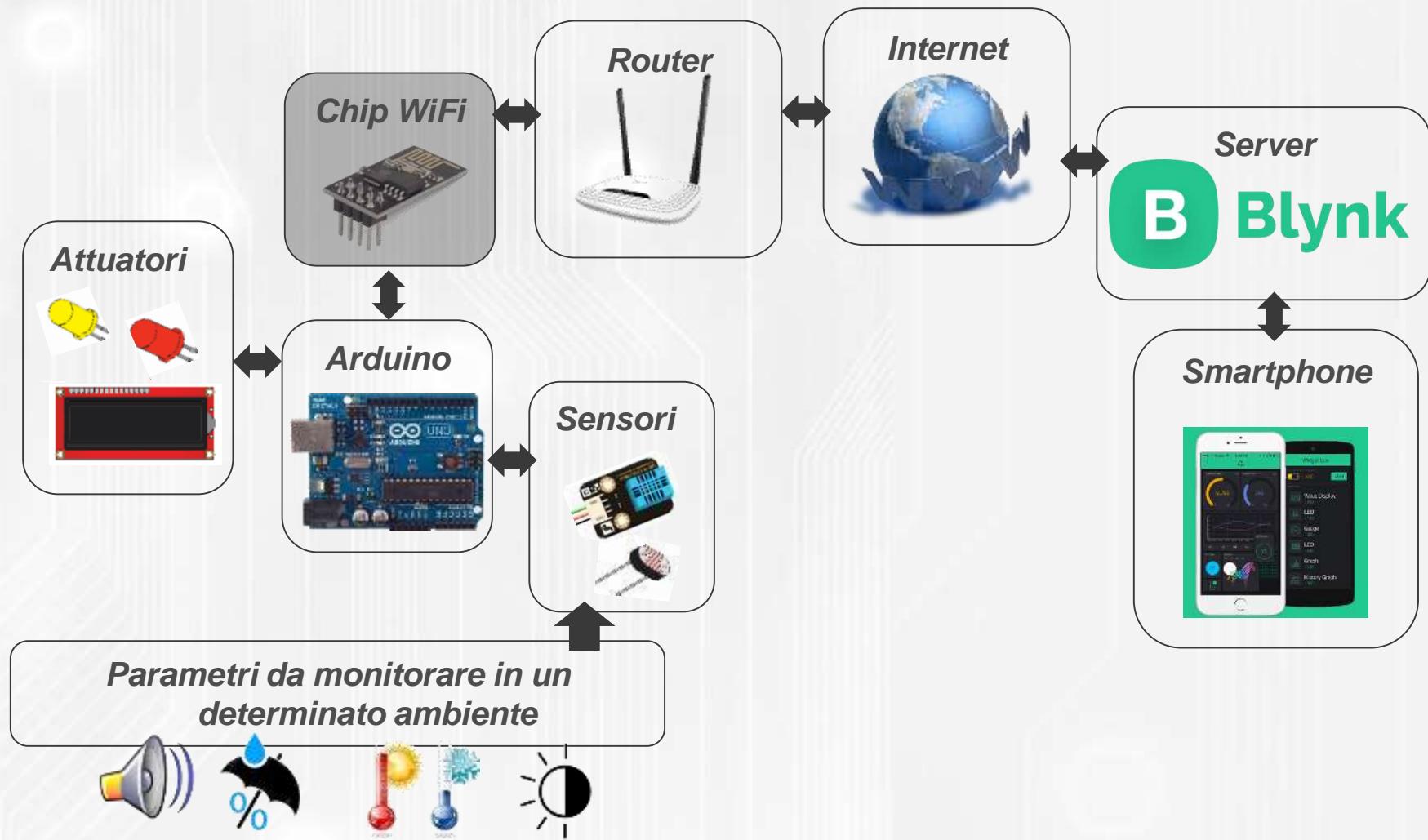
*Titolare del corso  
Prof. Andrea Cusano*

*Assistant:  
Ing. Patrizio Vaiano*

# Controllo sensori e attivazione attuatori da remoto con ARDUINO UNO



# Controllo sensori e attivazione attuatori da remoto con ARDUINO UNO



# Moduli Wireless ESP8266

Il modulo Wireless ESP8266 è basato su un SoC MCU a 32bit, con una discreta capacità di elaborazione, grazie alla quale è possibile gestire diverse periferiche.

Oltre a potere essere programmato come una vera e propria scheda di sviluppo, può essere utilizzato come adattatore Wireless-Seriale con un collegamento di tipo SPI/SDIO.

Risulta essere un ottimo punto di partenza per applicazioni di network.

I moduli ESP8266 sono disponibili in diversi formati denominati da ESP-01 a ESP-12.

La principale differenza tra le varie versioni dei moduli riguarda il loro formato e la quantità di memoria flash, ma anche la distribuzione dei pin GPIO, che non in tutte le versioni sono resi tutti disponibili.



# Modulo Seriale Wi-Fi ESP-01

Il Modulo Seriale Wi-Fi ESP-01 di Ai-Thinker si basa sul chip ESP8266 prodotto dalla Espressif  
È un sistema adatto a fornire connettività Wi-Fi a schede come Arduino e simili ma è possibile utilizzarlo direttamente per creare progetti IoT (Internet of Things) connessi alla rete

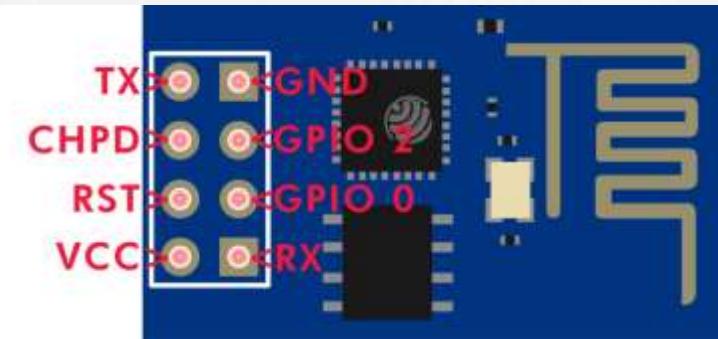
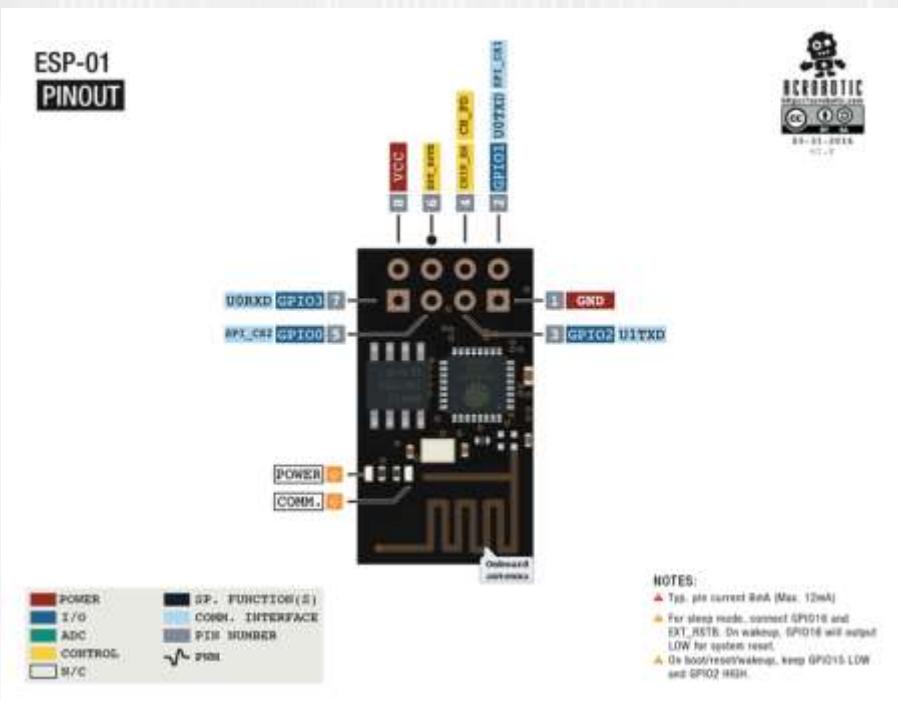


L'ESP-01 ha due modalità di avvio, a seconda di come sono impostati i pin GPIO-0 e GPIO-2 e può essere programmato tramite l'Arduino IDE:

1. **modalità programmazione**, che viene effettuata mediante l'interfaccia UART e permette di caricare sulla scheda il firmware e gli sketch;
2. **modalità di boot**, utilizzando la memoria flash esterna della scheda, ovvero per porre l'ESP in modalità esecuzione programmi.

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	TX Power	802.11 b: +20 dBm 802.11 g: +17 dBm 802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps) 802.11 g: -75 dbm (54 Mbps) 802.11 n: -72 dbm (MCS7)
Hardware	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

# Modulo Seriale Wi-Fi ESP-01

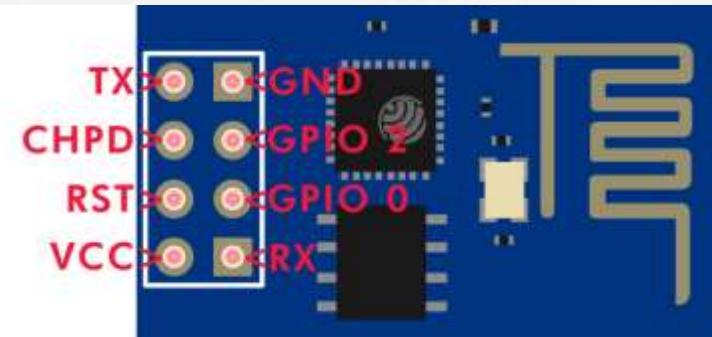
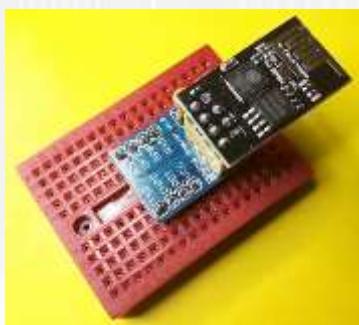
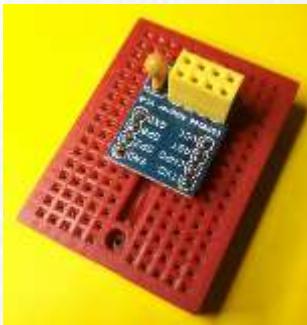
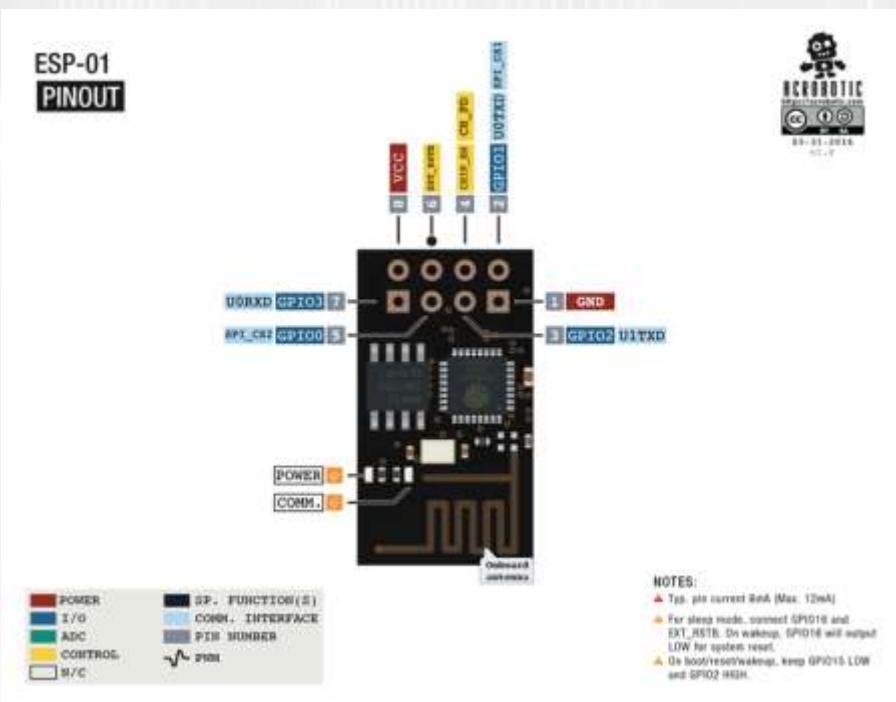


- VCC e GND
- TX e RX
- CH PD (EN)
- RST

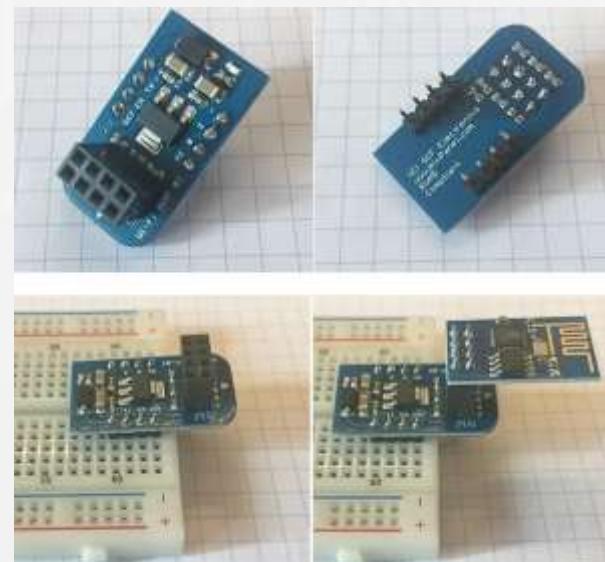
Il pin GPIO0 oltre ad essere un pin utilizzabile per controllare periferiche, è anche il pin che abilita alla programmazione e in fase di caricamento del programma deve essere collegato a GND. Il pin CHPD è il pin ENABLE (abilitazione).

Oltre ad essere un pin necessario per entrare in modalità di programmazione collegandolo a 3.3V, è un pin che in alcuni modelli deve rimanere collegato in ogni caso, anche dopo la programmazione, poiché permette di non mandare in “stand-by” il modulo ESP-01.

# Modulo Seriale Wi-Fi ESP-01



- VCC e GND
- TX e RX
- CH PD (EN)
- RST



# *Modulo Seriale Wi-Fi ESP-01*

*L'ESP8266 permette di interfacciare la nostra applicazione alla rete internet, quindi controllare degli oggetti, tramite i suoi pin I/O, sfruttando il collegamento Wi-Fi. L'ESP8266 può collegarsi ad una rete preesistente (client) o crearne una propria (server) alla quale è possibile collegarsi col PC, tablet o smartphone.*

*Passi per utilizzare ESP8266 per trasmissione dati:*

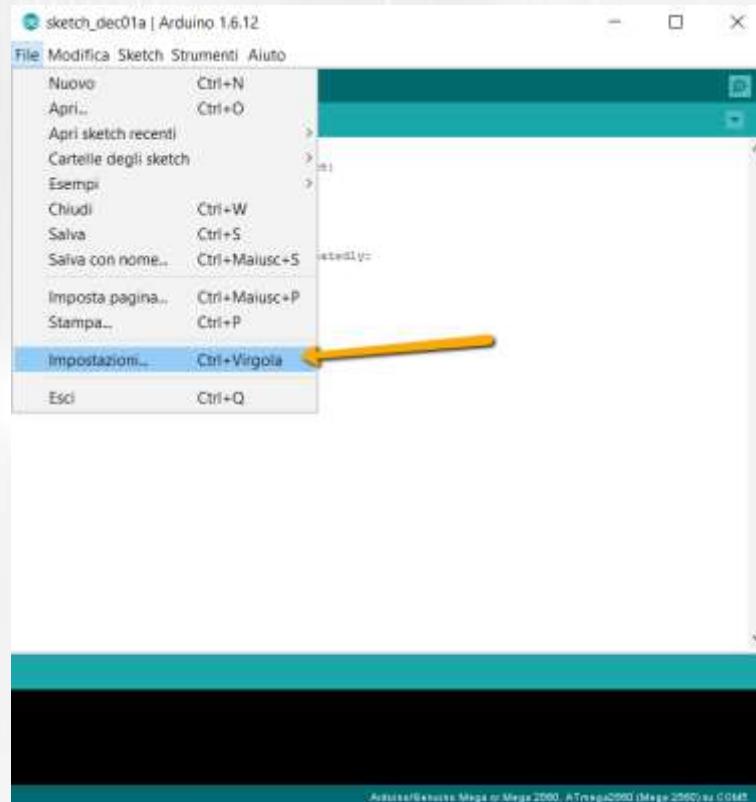
- *Scaricare librerie*
- *Caricare su ESP8266 il firmware*

*Modulo Seriale Wi-Fi ESP-01*

***Arduino IDE 1.x.x***

# Modulo Seriale Wi-Fi ESP-01

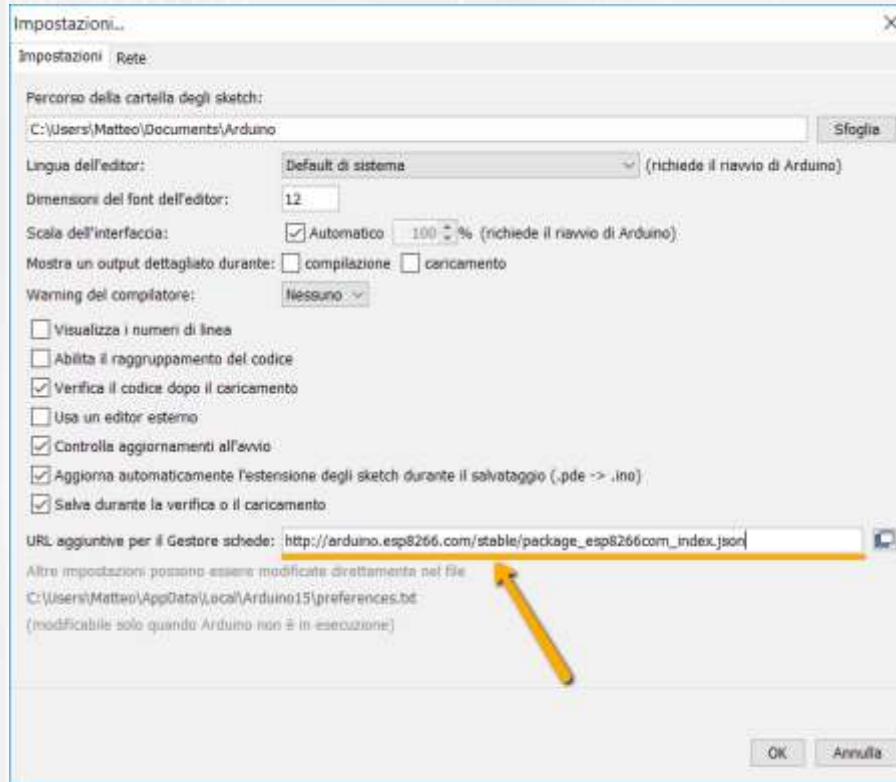
## Librerie



dalla schermata principale andare su **File>impostazioni**

# Modulo Seriale Wi-Fi ESP-01

## Librerie



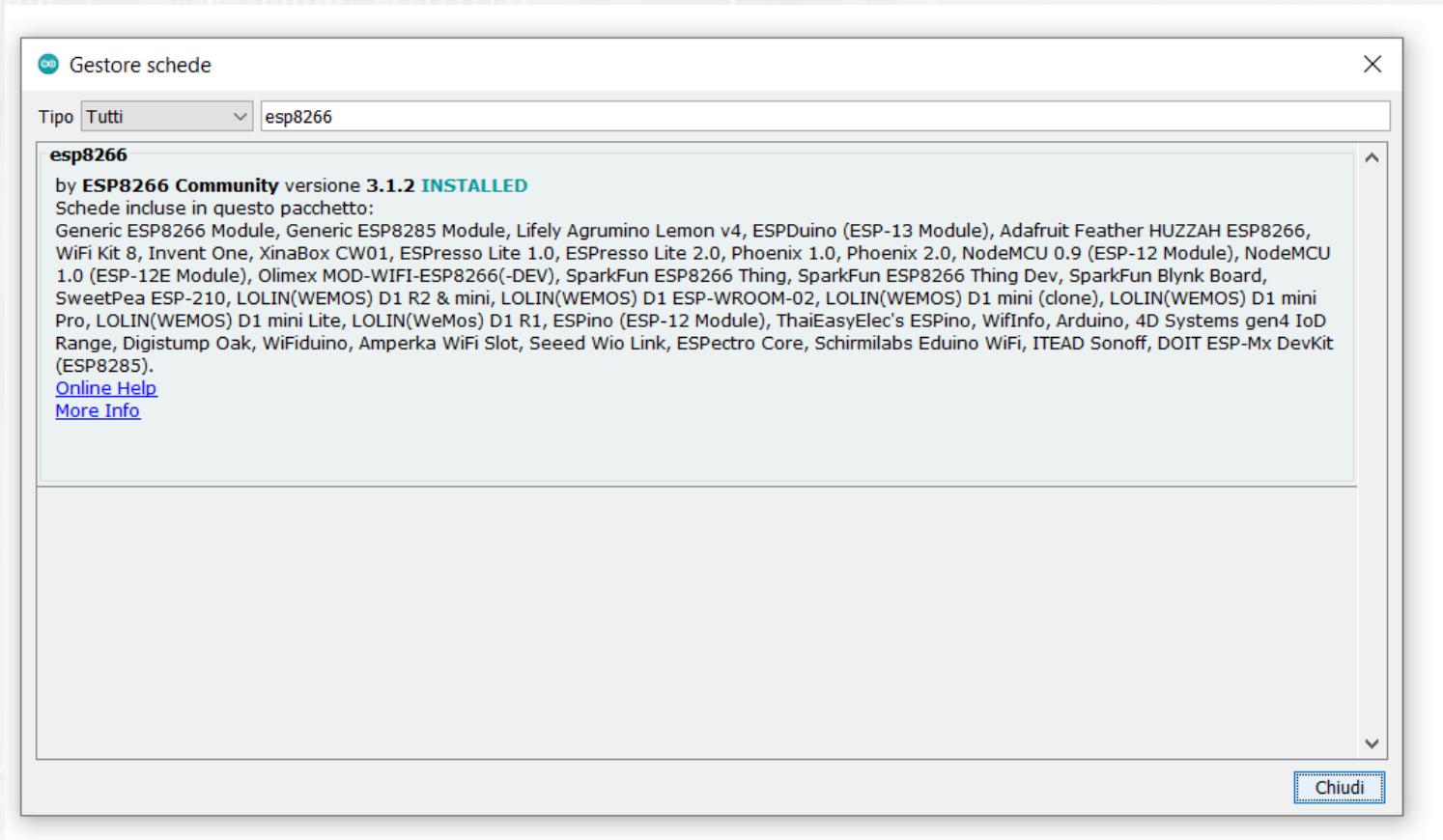
Inserire la stringa

**[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)**

nel campo "**URL aggiuntive per il Gestore Schede**". Quindi, fare clic sul pulsante "**OK**".

# Modulo Seriale Wi-Fi ESP-01

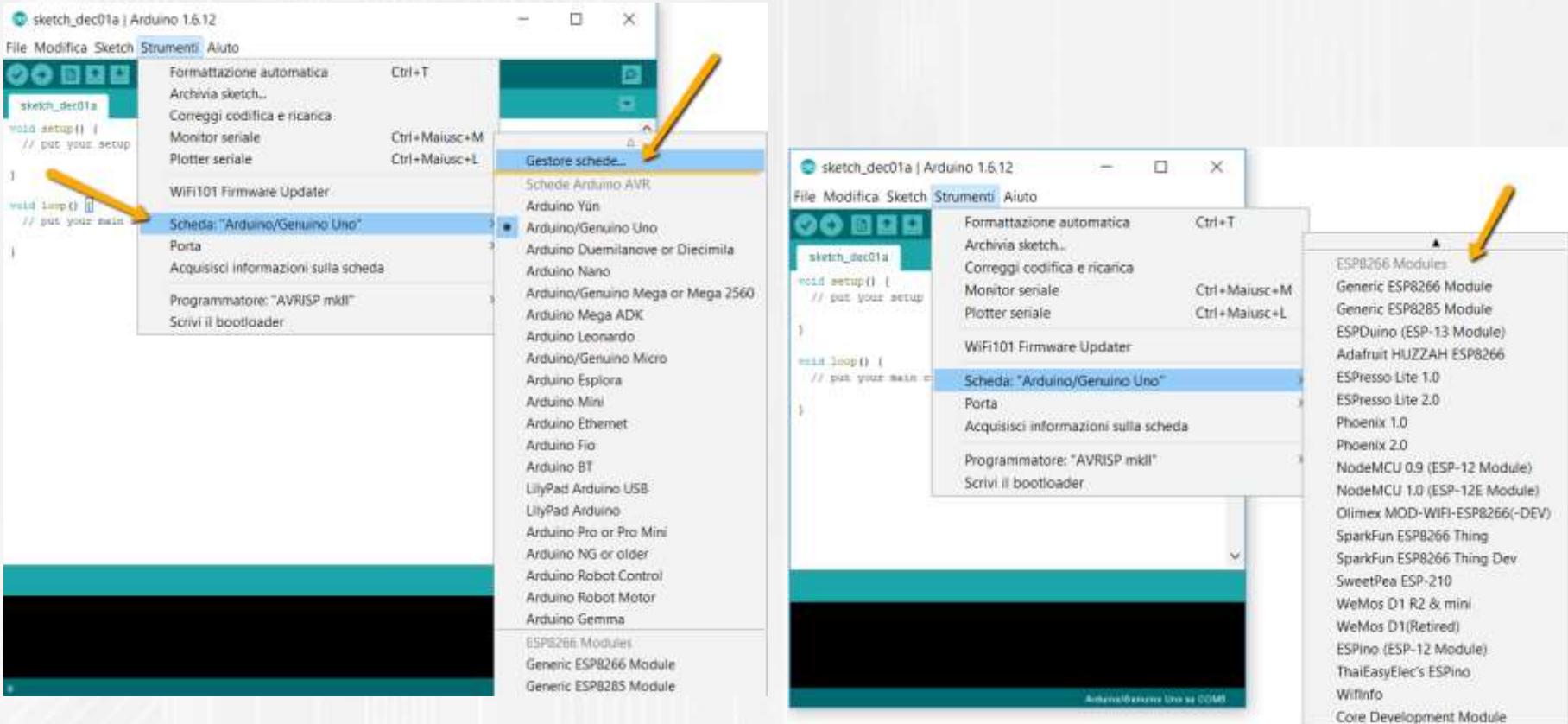
## Librerie



All'interno dell'IDE Arduino, recarsi in **strumenti>scheda>Gestore schede**, dove comparirà una schermata dalla quale è possibile selezionare **ESP8266** ed installare.

# Modulo Seriale Wi-Fi ESP-01

## Caricare su *ESP8266* il firmware



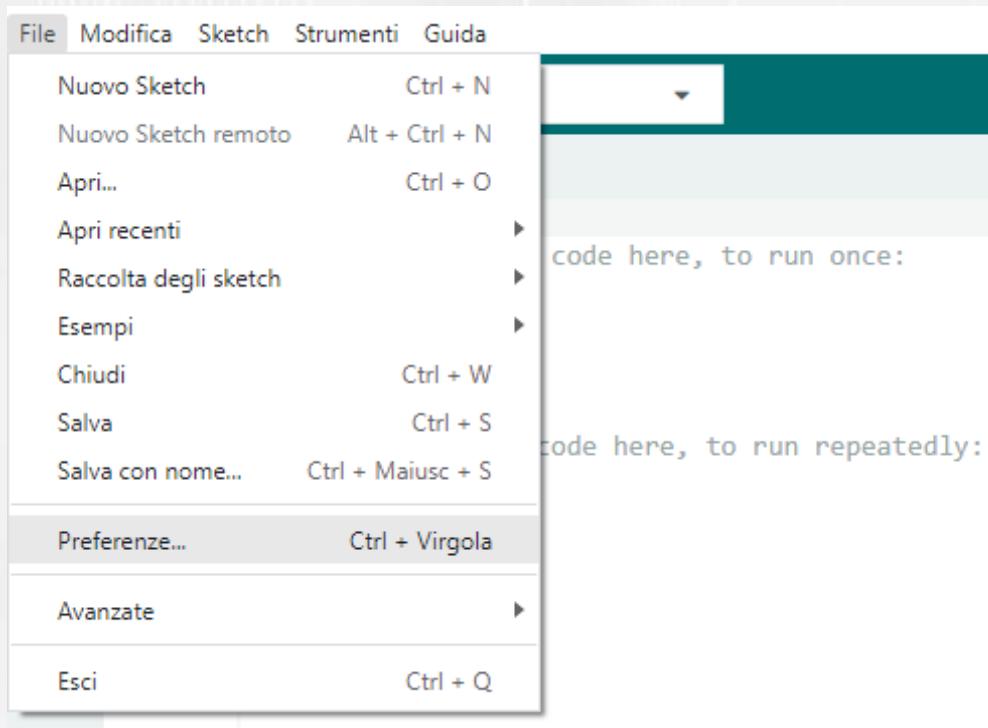
All'interno del percorso **strumenti>scheda** è possibile selezionare **Generic ESP8266 Module**. Selezionarlo per programmare le schede con **ESP8266**.

*Modulo Seriale Wi-Fi ESP-01*

***Arduino IDE 2.x.x***

# Modulo Seriale Wi-Fi ESP-01

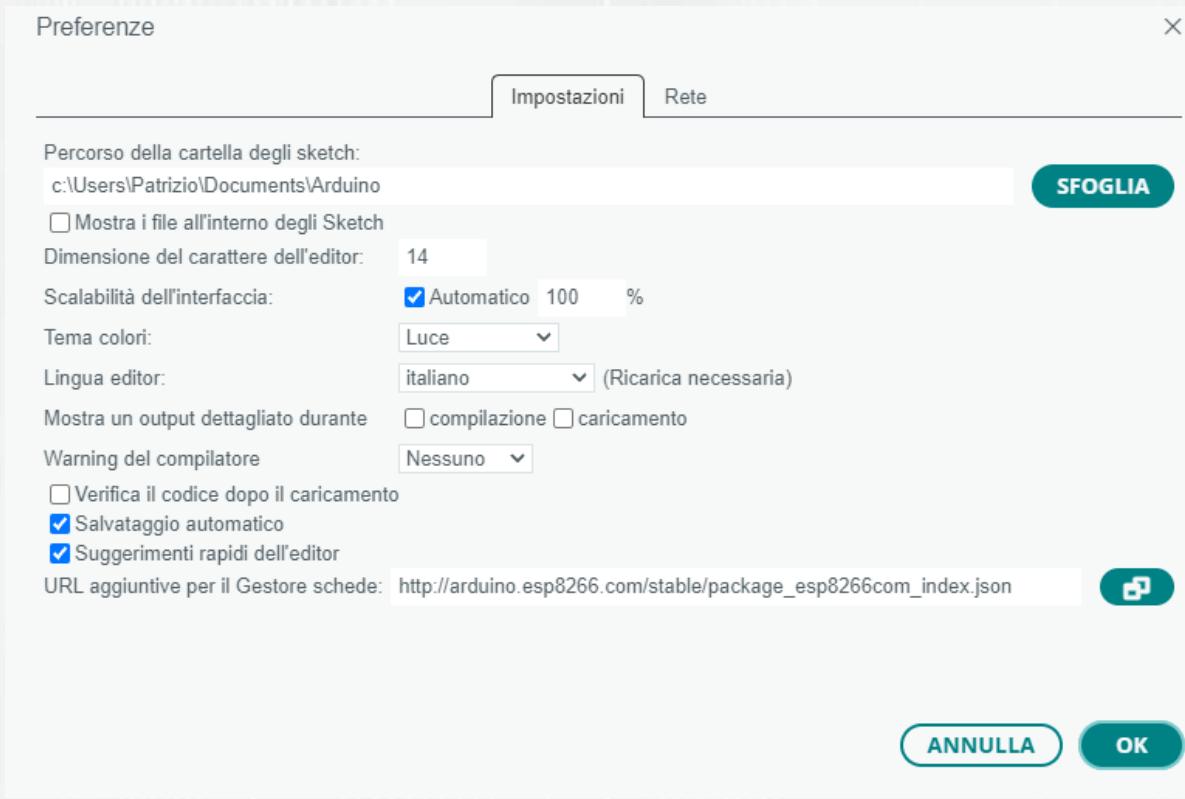
## Librerie



dalla schermata principale andare su **File>Preferenze**

# Modulo Seriale Wi-Fi ESP-01

## Librerie



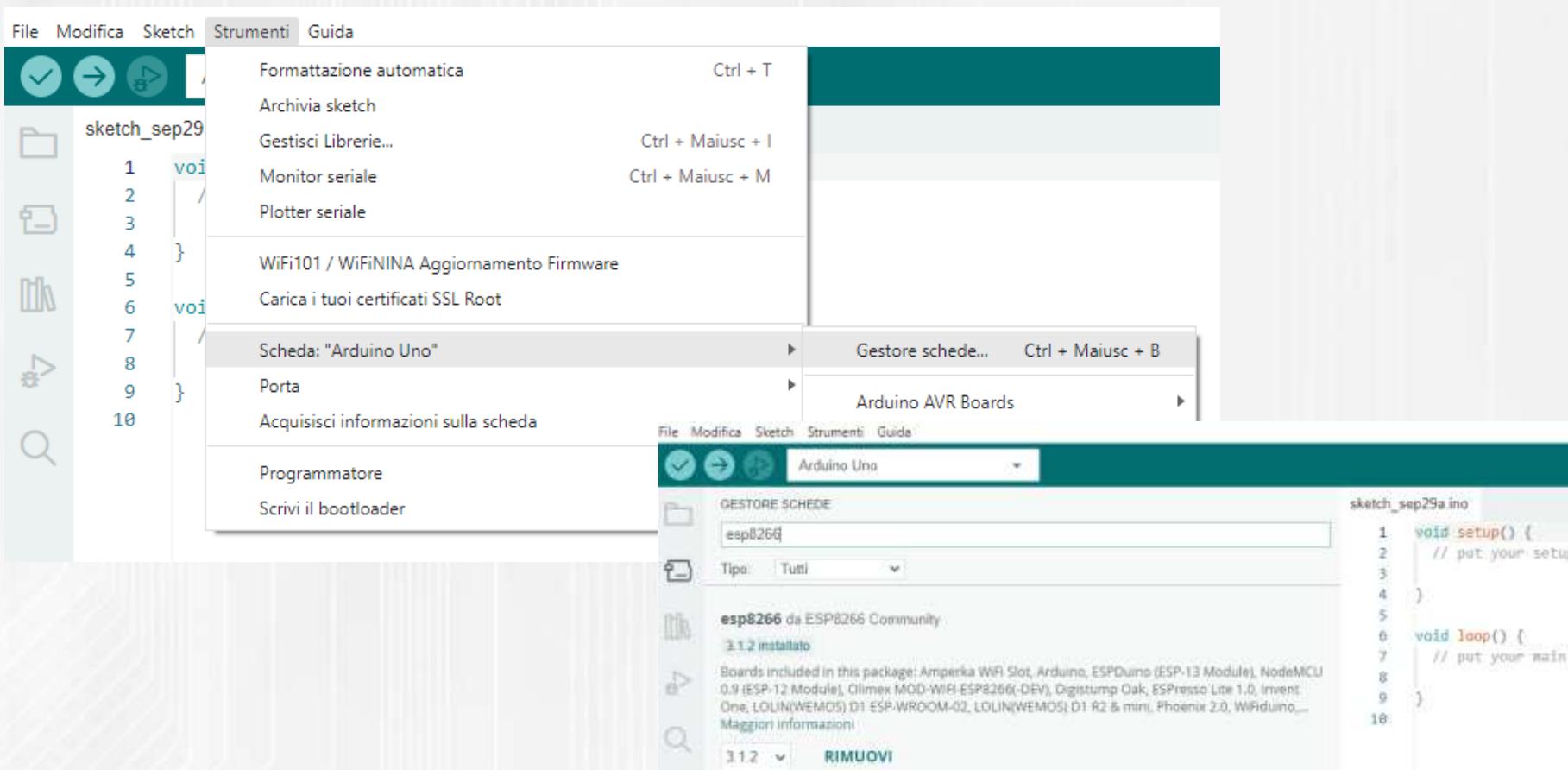
Inserire la stringa

**http://arduino.esp8266.com/stable/package\_esp8266com\_index.json**

nel campo "URL aggiuntive per il Gestore Schede". Quindi, fare clic sul pulsante "OK".

# Modulo Seriale Wi-Fi ESP-01

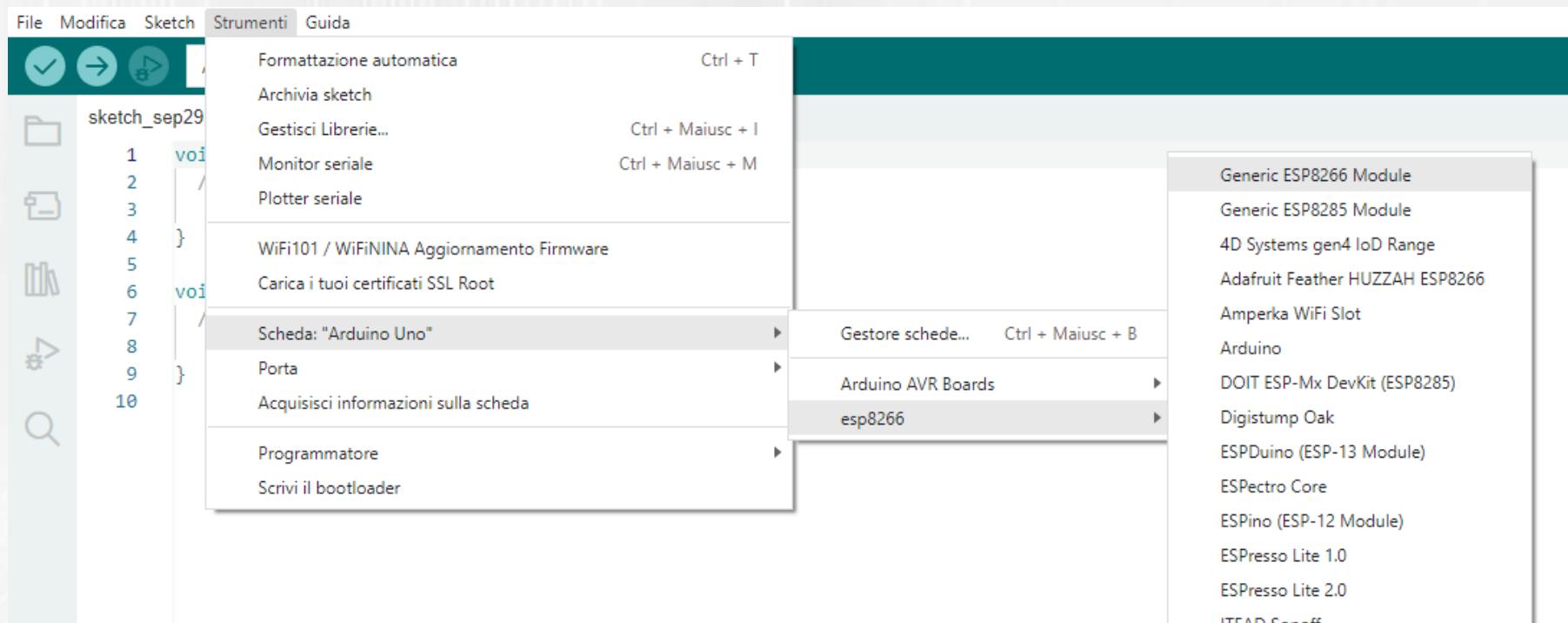
## Librerie



All'interno dell'IDE Arduino, recarsi in **strumenti>scheda>Gestore schede**, dove comparirà una schermata dalla quale è possibile selezionare **ESP8266** ed installare.

# Modulo Seriale Wi-Fi ESP-01

## Caricare su *ESP8266* il firmware

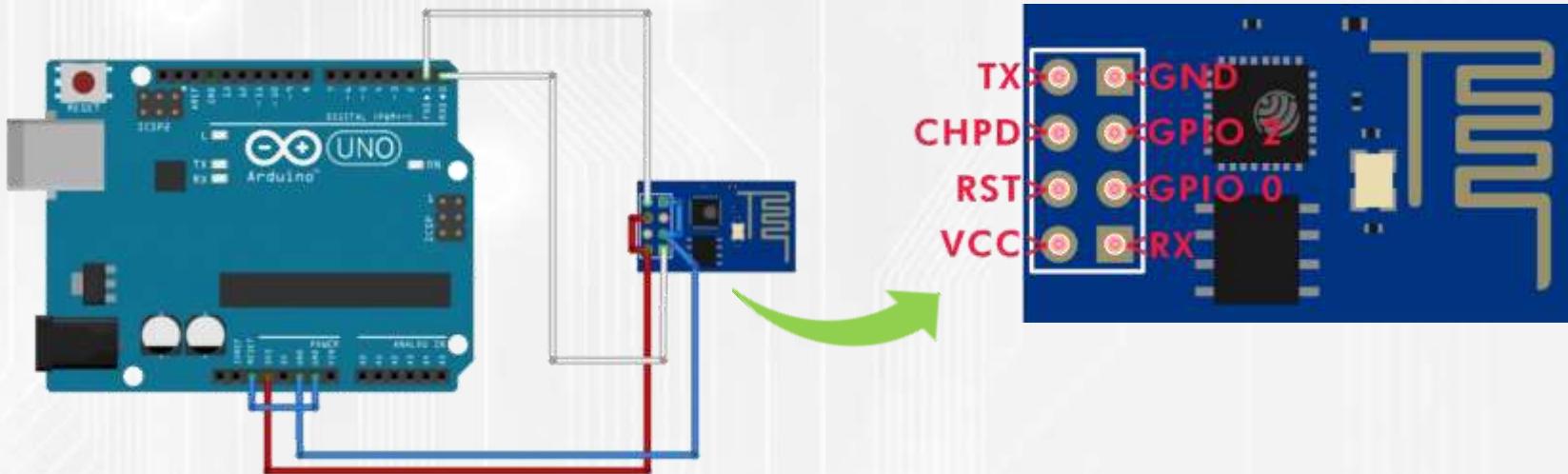


All'interno del percorso **strumenti>scheda** è possibile selezionare **Generic ESP8266 Module**. Selezionarlo per programmare le schede con **ESP8266**.

# Modulo Seriale Wi-Fi ESP-01

## Caricare su ESP8266 il firmware

Utilizzare Arduino UNO come adattatore seriale (USB), per caricare su ESP8266 il firmware.

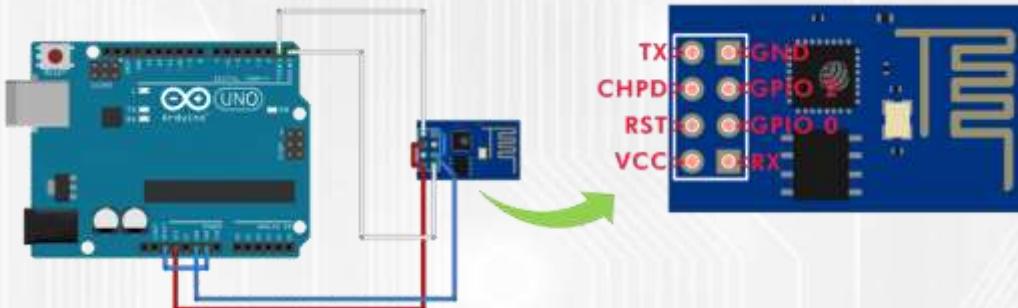


Circuito da utilizzare per caricare su ESP8266 il firmware

# Modulo Seriale Wi-Fi ESP-01

## Caricare su ESP8266 il firmware

Circuito da utilizzare per caricare  
su ESP8266 il firmware



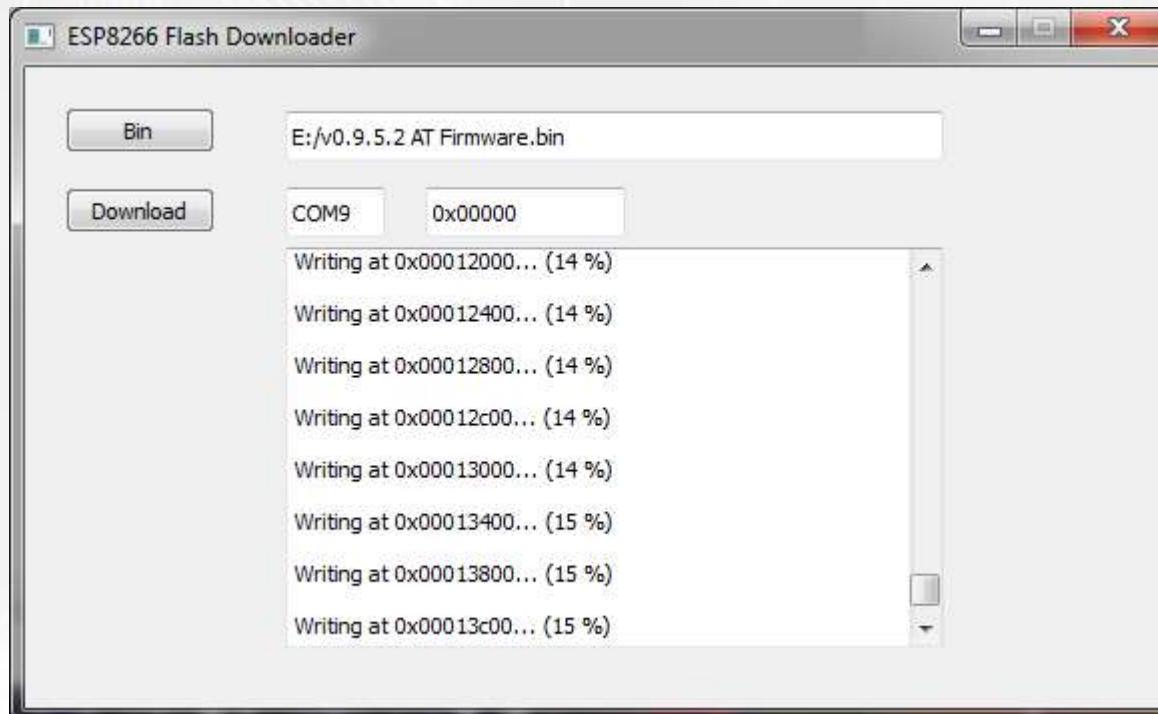
TX ESP01	TX ARDUINO
GND ESP01	GND ARDUINO
CHPD(EN) ESP01	3.3V ARDUINO
GPIO2 ESP01	
GPIO0 ESP01 (o per ATMODE, si per FLASH MODE)	GND ARDUINO (o per ATMODE, si per FLASH MODE)
RST ESP01	
3V3 ESP01	3.3V ARDUINO
RX ESP01	RX ARDUINO

- I pin GND e RESET di Arduino Uno devono essere connessi tra di loro, per permettere che Arduino funga da adattatore seriale;
- I pin TX e RX di Arduino Uno devono essere collegati rispettivamente al TX e RX di ESP-01;
- Il pin GPIO-0 di ESP-01 va connesso a massa;
- Il pin CH\_PD di ESP-01(o CH\_ENABLE) va connesso a 3.3V;
- Il pin VCC di ESP-01 va connesso a 3.3V;
- Il pin GND di ESP-01 va connesso a massa.

# Modulo Seriale Wi-Fi ESP-01

## Caricare su ESP8266 il firmware

- Scaricare il file .bin (firmware), ad esempio AT\_MODE\_ESP8266.zip
- Scaricare un'applicazione per caricare il firmware, ad esempio ESP8266 Flash Downloader oppure ESP8266Flasher (exe)
- Aprire ESP8266 Flash Downloader
- Caricare il firmware
- Selezionare la porta COM sul quale è presente Arduino e fare partire il Download



# *Modulo Seriale Wi-Fi ESP-01*

## *Verifica del corretto caricamento del firmware*

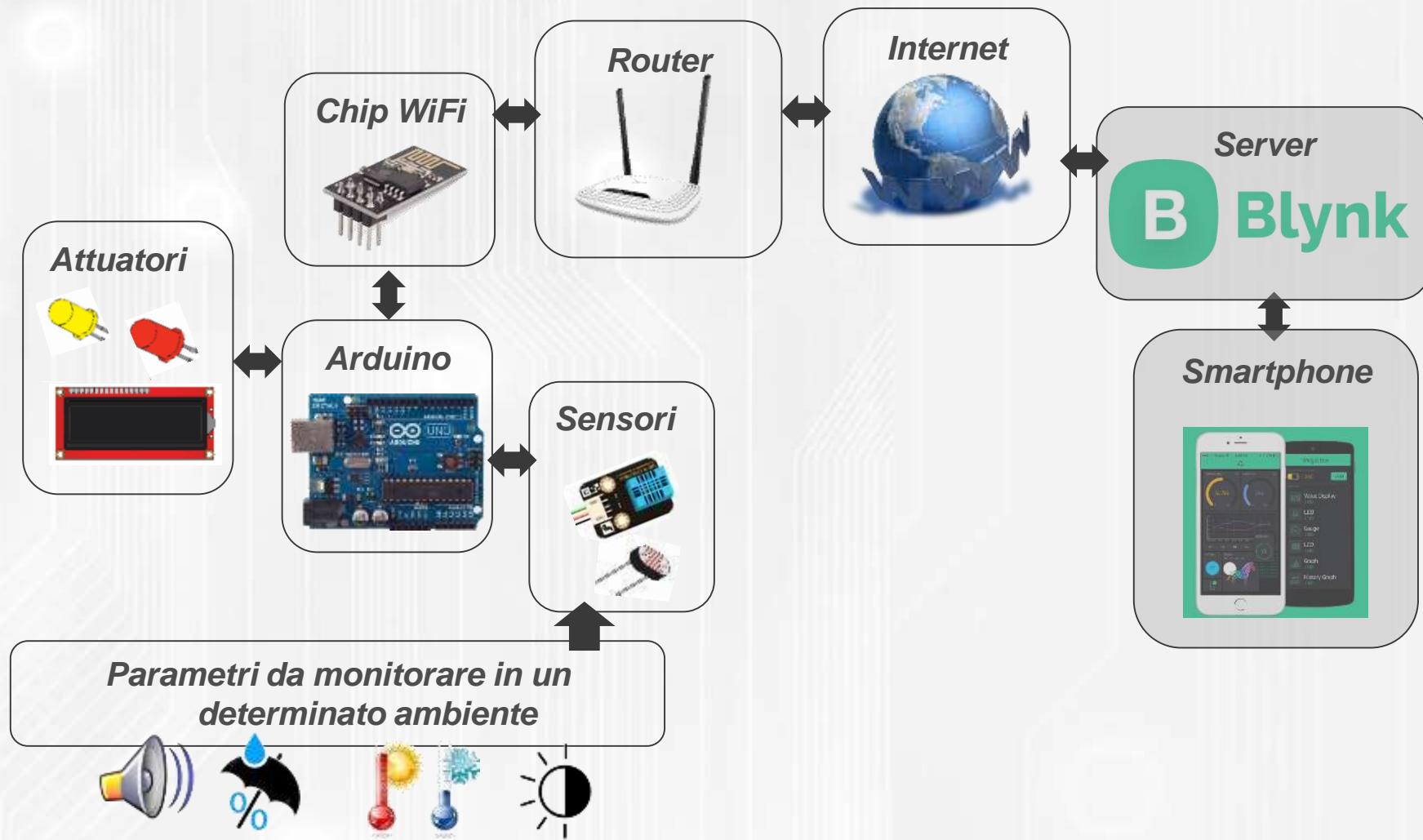
- Il pin GPIO-0 di ESP-01 va disconnesso dalla massa.
- Il pin GND di ESP-01 va disconnesso dalla massa (oppure scollegare il cavo USB).
- Il pin RESET di ESP-01 va connesso a massa per 2 secondi.
- Riconnettere il pin GND di ESP-01 a massa (oppure ricollegare il cavo USB).

# *Modulo Seriale Wi-Fi ESP-01*

## *Verifica del corretto caricamento del firmware*

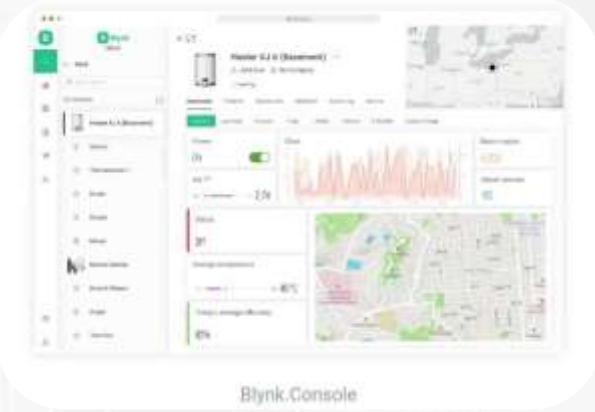
- Aprire il monitor seriale dell'Arduino IDE, impostando i parametri in basso a dx con i caratteri di controllo NL & CR selezionati, baudrate 115200.
- Digitare il comando AT e se il caricamento è avvenuto correttamente **si visualizzerà la risposta OK**.
- Riaprire il monitor seriale e digitare il comando  
**AT+UART\_DEF=9600,8,1,0,0** per far sì che Arduino comunichi con il modulo Wi-Fi, visto che hanno baudrate diversi, rispettivamente 9600 e 115200.
- Digitare nuovamente il comando AT per ottenere l'OK della fase precedente.

# Controllo sensori e attivazione attuatori da remoto con ARDUINO UNO



# Blynk

Blynk è una suite comprensiva di software che permette la prototipazione, lo sviluppo e la gestione remota di dispositivi elettronici connessi su qualunque scala. Blynk consente agli utenti di connettere i propri dispositivi hardware al cloud e di creare applicazioni OS, Android, e web, analizzare dati in tempo reale o storici dai dispositivi, di controllarli da remoto, di ricevere notifiche, etc.



Blynk.Console

# Blynk

Per configurare al meglio Blynk in modo tale da trasferire le informazioni dei sensori all'app ed eventualmente attivare gli attuatori bisogna eseguire le configurazioni su:

- o ARDUINO IDE



- o SMARTPHONE





# Configurazione di Blynk su ARDUINO IDE

Scaricare le librerie di Blynk collegandosi al sito: <https://blynk.io/en/getting-started>

The screenshot shows a web browser with three tabs open: "Get started with Blynk", "Install Blynk Library in Arduino", and "Release v1.3.2 - blynkk/blynk-ide". The main content area displays three sections:

- A Smartphone**: Shows an icon of a smartphone. Requirements: Android OS version 5+, iOS version 14.1+. Note: Blynk doesn't run on Windows Phones, Blackberries and other dead platforms. Sorry. You can also run Blynk on emulators.
- IoT Hardware**: Shows an icon of a cube. Blynk can run on over 400 hardware modules. Popular ones include ESP32, ESP8266, NodeMCU, Arduino (any model), Raspberry Pi (any model), and Particle (any model). [Full list of supported hardware →](#)
- Internet Connection**: Shows an icon of a network connection. To connect your hardware to the Internet, you can choose almost any module either built-in, or external shields. Supported connectivity includes WiFi, Ethernet, and Cellular (GSM, 2g, 3g, 4g, LTE).



# Configurazione di Blynk su ARDUINO IDE

Scaricare le librerie di Blynk collegandosi al sito: <https://blynk.io/en/getting-started>

The screenshot shows a web browser with three tabs open: "Get started with Blynk", "Install Blynk Library in Arduino IDE", and "Release v1.3.2 - blynkk/blynk-ide". The main content area displays the "Follow Quickstart guide" and "Install Blynk Library" sections. The "Install Blynk Library" section includes a description of the library's function, a note about its flexibility, and a link to a help article. To the right, a window of the Arduino IDE shows a snippet of C++ code for initializing the Blynk library.

C++ (Arduino)	Python	JavaScript	3rd party libraries
Most popular library for:	Python 2, 3, MicroPython	Works with Browsers, Node.js, Espruino,	Particle Node-RED

# Configurazione di Blynk su ARDUINO IDE

Seguendo il link indicato sono illustrati 3 metodi di installazione

The screenshot shows a web browser displaying the Blynk Documentation page for the 'Install Blynk Library in Arduino IDE'. The URL is <https://docs.blynk.io/en/blynk-library-firmware-api/installation/install-blynk-library-in-arduino-ide>. The page title is 'Install Blynk Library in Arduino IDE'. It includes a sidebar with navigation links like 'Introduction', 'GETTING STARTED', 'Supported Hardware', 'Quickstart', 'Device Activation Methods', 'Template Quick Setup', 'Send Data From Hardware To Blynk', 'Control Devices (GPIOs and beyond)', 'Events', 'Notifications (Alerts)', and 'Sign Up / Sign In'. The main content area starts with a heading 'Install Blynk Library in Arduino IDE' and a sub-section 'Step by step guide on how to install Blynk Library'. Below this, it says 'There are three ways to install Blynk Library for Arduino IDE:' followed by a numbered list: 1. Using built-in library manager in Arduino IDE, 2. Installing Blynk library as ZIP file in Arduino IDE, and 3. Manually install Blynk Library. A sub-section titled '1. Install Blynk Library using built-in library manager in Arduino IDE' provides instructions: 'To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.6.2). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.' At the bottom, there is a screenshot of the Arduino IDE interface showing the 'Sketch' menu open with 'Include Library' selected.

# Configurazione di Blynk su ARDUINO IDE

Scegliamo il primo metodo

Install Blynk Library in Arduino IDE | Release v1.3.2 - blynkk/blynk-ide | + https://docs.blynk.io/en/blynk-library-firmware-api/installation/install-blynk-library-in-arduino-ide Platform log in Company website Resources GitHub Search

**B Blynk Documentation**

Introduction  
GETTING STARTED  
Supported Hardware  
Quickstart  
Device Activation Methods  
Template Quick Setup  
Send Data From Hardware To Blynk  
Control Devices (GPIOs and beyond)  
Events  
Notifications (Alerts)  
Sign Up / Sign In

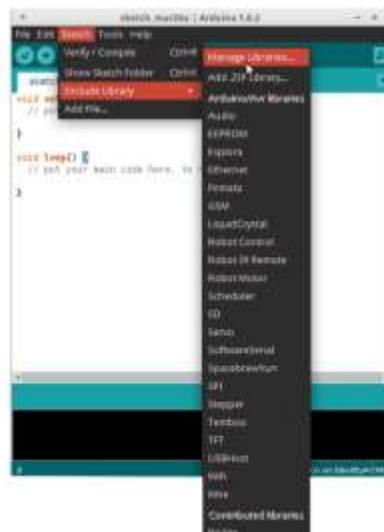
GENERAL CONCEPTS  
Developer Mode  
Device  
Device Template

ON THIS PAGE

1. Install Blynk Library using built-in library manager in Arduino IDE
2. Install Blynk as ZIP file
3. Install Blynk library manually
4. Troubleshooting

1. Install Blynk Library using built-in library manager in Arduino IDE

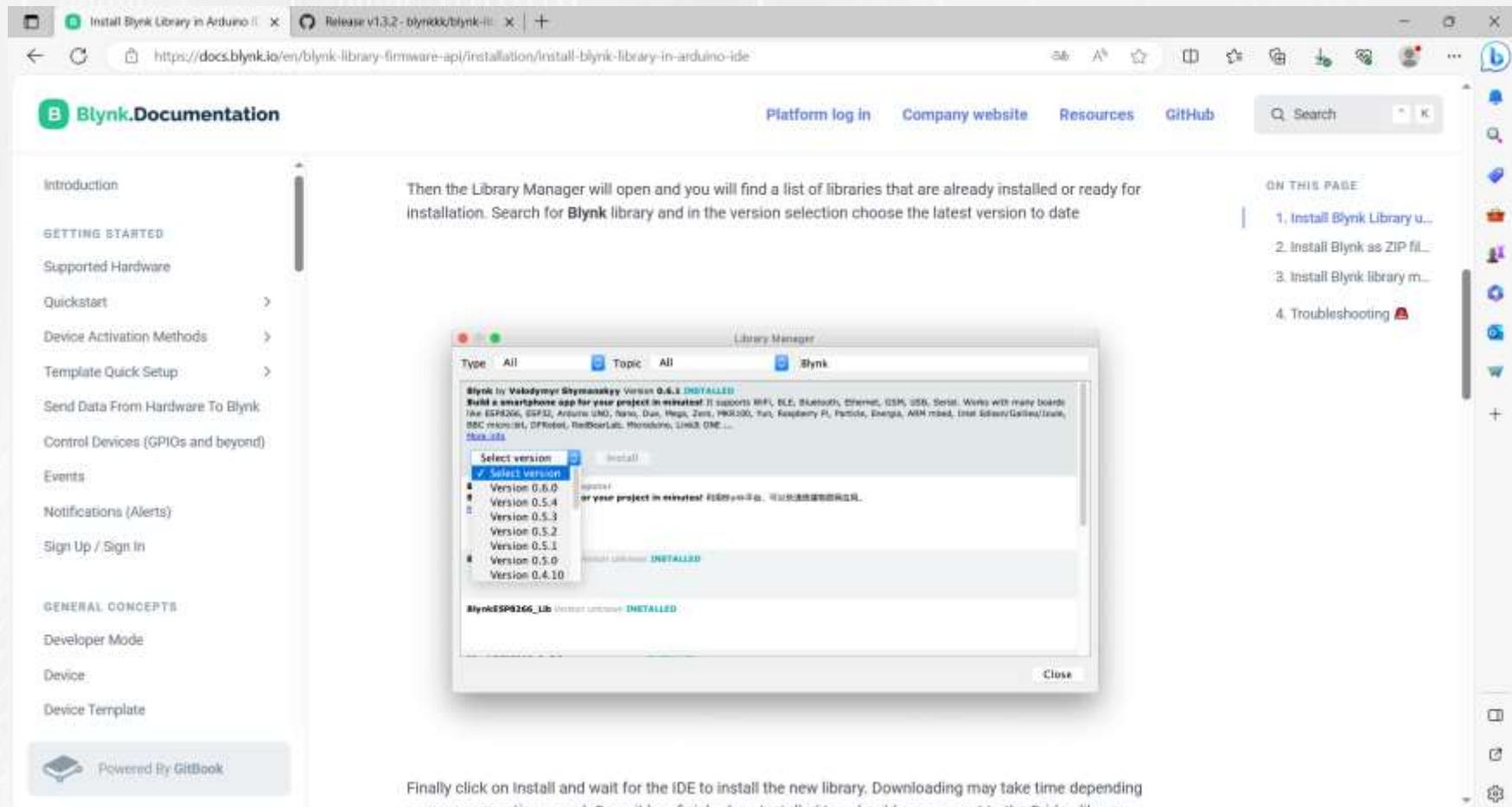
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.6.2). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



Powered By GitBook

# Configurazione di Blynk su ARDUINO IDE

Installiamo la libreria indicata (Blynk)



The screenshot shows a web browser displaying the Blynk Documentation page for installing the library in Arduino IDE. The URL in the address bar is <https://docs.blynk.io/en/blynk-library-firmware-api/installation/install-blynk-library-in-arduino-ide>. The main content area shows instructions for using the Library Manager to install the Blynk library. Below the instructions, a screenshot of the Arduino IDE Library Manager window is shown, displaying the Blynk library version 0.6.1 as installed. The sidebar on the left contains links to various Blynk documentation sections like Introduction, Getting Started, and Device Activation Methods. The right sidebar lists "ON THIS PAGE" with four numbered items: 1. Install Blynk Library using Library Manager, 2. Install Blynk as ZIP file, 3. Install Blynk library manually, and 4. Troubleshooting.

Then the Library Manager will open and you will find a list of libraries that are already installed or ready for installation. Search for **Blynk** library and in the version selection choose the latest version to date

ON THIS PAGE

1. Install Blynk Library using Library Manager
2. Install Blynk as ZIP file
3. Install Blynk library manually
4. Troubleshooting

Finally click on **Install** and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, as installed too should appear next to the Blynk library

# Configurazione di Blynk su ARDUINO IDE

In alternativa, è possibile seguire il secondo metodo

The screenshot shows a web browser displaying the Blynk Documentation page. The URL in the address bar is <https://docs.blynk.io/en/blynk-library-firmware-api/installation/install-blynk-library-in-arduino-ide>. The page content is as follows:

**2. Install Blynk as ZIP file in Arduino IDE**

Blynk library is available as a downloadable ZIP. Starting with Arduino IDE version 1.0.5, you can install 3rd party libraries in the IDE.

Download Blynk Library by clicking the button [Download Blynk Library](#)

**Do not unzip the downloaded library, leave it as is.**

In the Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library. At the top of the drop down list, select the option to "Add .ZIP Library".

1. Return to the Sketch > Include Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory.

The Library will be available to use in sketches, but with older IDE versions examples for the library will not be exposed in the File > Examples until after the IDE has restarted.

**3. Install Blynk library manually**

Download the latest [Blynk\\_Release\\_vXX.zip](#) file from the GitHub page: [Download Blynk Library](#)

1. Unzip the [Blynk\\_Release\\_vXX.zip](#) archive. You will notice that archive contains several folders and several libraries.

# Blynk

## Configurazione di Blynk su ARDUINO IDE

Seguendo il link indicato è possibile scaricare il file .zip

The screenshot shows a web browser window with the URL <https://github.com/blynkkk/blynk-library/releases/tag/v1.3.2>. The page displays the contents of the Blynk library release v1.3.2. It includes a logo, a message "With love from Ukraine" with heart icons, instructions on how to install the library, a section titled "In this release" with bullet points about Arduino Uno R4 Over the Air upgrade and bug fixes, and a note to update the IDE, Libraries, and Boards. There is also a GitHub star button and an "Assets" section containing three download links: "Blynk Release v1.3.2.zip", "Source code (.zip)", and "Source code (.tar.gz)". The "Assets" section also shows that 2 people reacted.

Install Blynk Library in Arduino IDE

Release v1.3.2 · blynkkk/blynk-library · Releases

v1.3.2

With love from Ukraine ❤️💛

How to install Blynk library: ↪ [link](#) ⇪

In this release

- Arduino UNO R4 Over the Air upgrade via [Blynk.NCP](#)
- Various bug fixes and improvements

Remember to regularly update your IDE, Libraries, and Boards!

Check out the full list of [supported hardware](#)

If you like Blynk, don't forget to give us a GitHub star! ⚡

Assets

<a href="#">Blynk Release v1.3.2.zip</a>	699 KB	3 weeks ago	
<a href="#">Source code (.zip)</a>		3 weeks ago	
<a href="#">Source code (.tar.gz)</a>		3 weeks ago	

2 people reacted

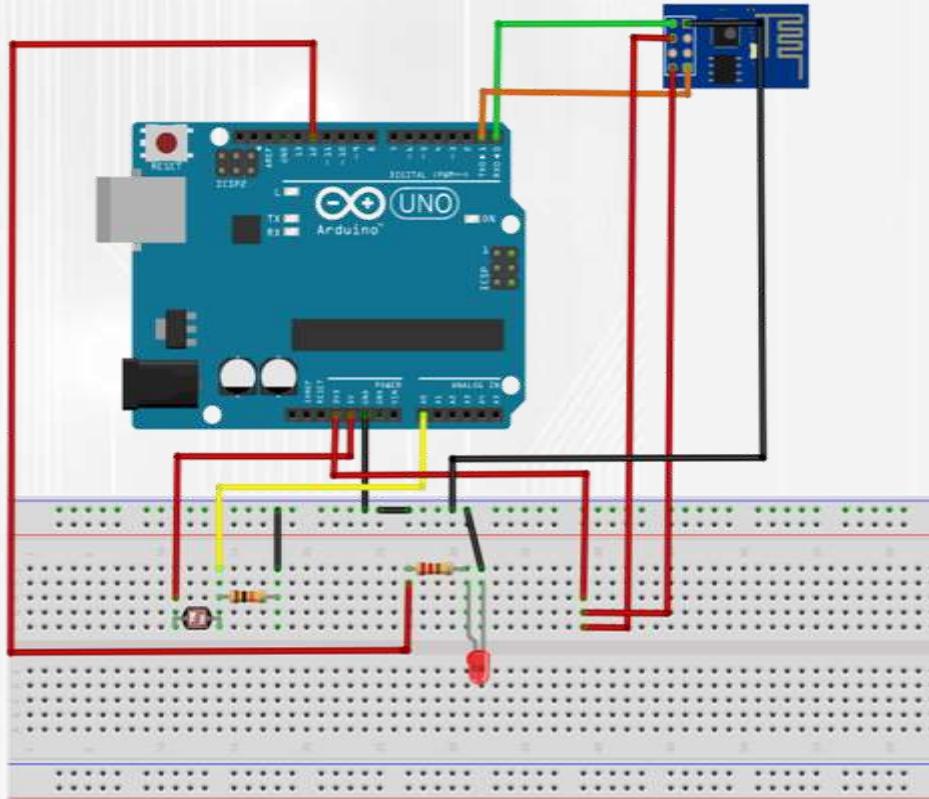


## Creazione algoritmo con IDE Arduino

Creazione di un'App per rilevamento dati da remoto del sensore e attivazione allarme con led e informazioni sull'allarme reperibili da terminale

# Blynk

## Circuito da realizzare



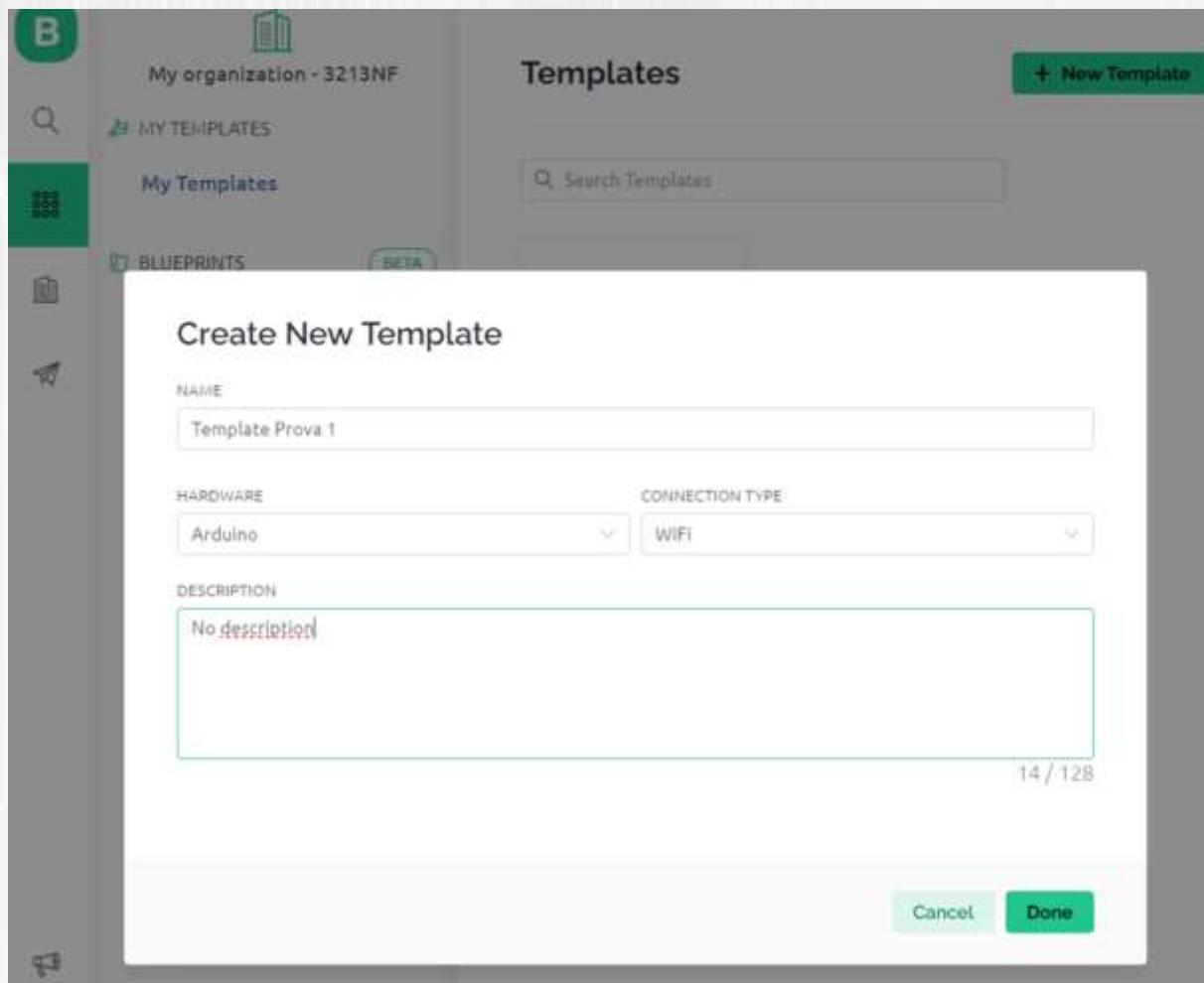
# *Blynk*

## *Configurazione di Blynk su PC*

- ✓ Andare sul sito: <https://blynk.cloud/dashboard/login>
- ✓ Creare Account
- ✓ Effettuare accesso su PC
- ✓ Esempio: creazione di un'App per rilevamento dati da remoto del sensore e attivazione allarme con led e informazioni sull'allarme reperibili da terminale

# Blynk

## Creazione del template



# Blynk

## Template

**B**

Template Prova 1

...

Cancel

Save

Home Datastreams Web Dashboard Automations Metadata Events Mobile

What's next?

- Configure template
- Set Up Datastreams
- Set up the Web Dashboard
- Add First Device

Template settings

Arduino, WiFi

Firmware configuration

Template ID and Template Name should be declared at the very top of the firmware code.

```
#define BLYNK_TEMPLATE_ID "TMPL402tWFg8D"
#define BLYNK_TEMPLATE_NAME "Template Prova
1"
```

# Blynk

## *Creazione Datastreams*

1. *Sensore* – serve a visualizzare il valore restituito dal sensore
2. *Terminale* – serve a visualizzare un messaggio di avviso per segnalare il superamento della soglia
3. *LED* – serve ad accendere una spia per segnalare il superamento della soglia

# Blynk

## Creazione Datastreams (Sensore)

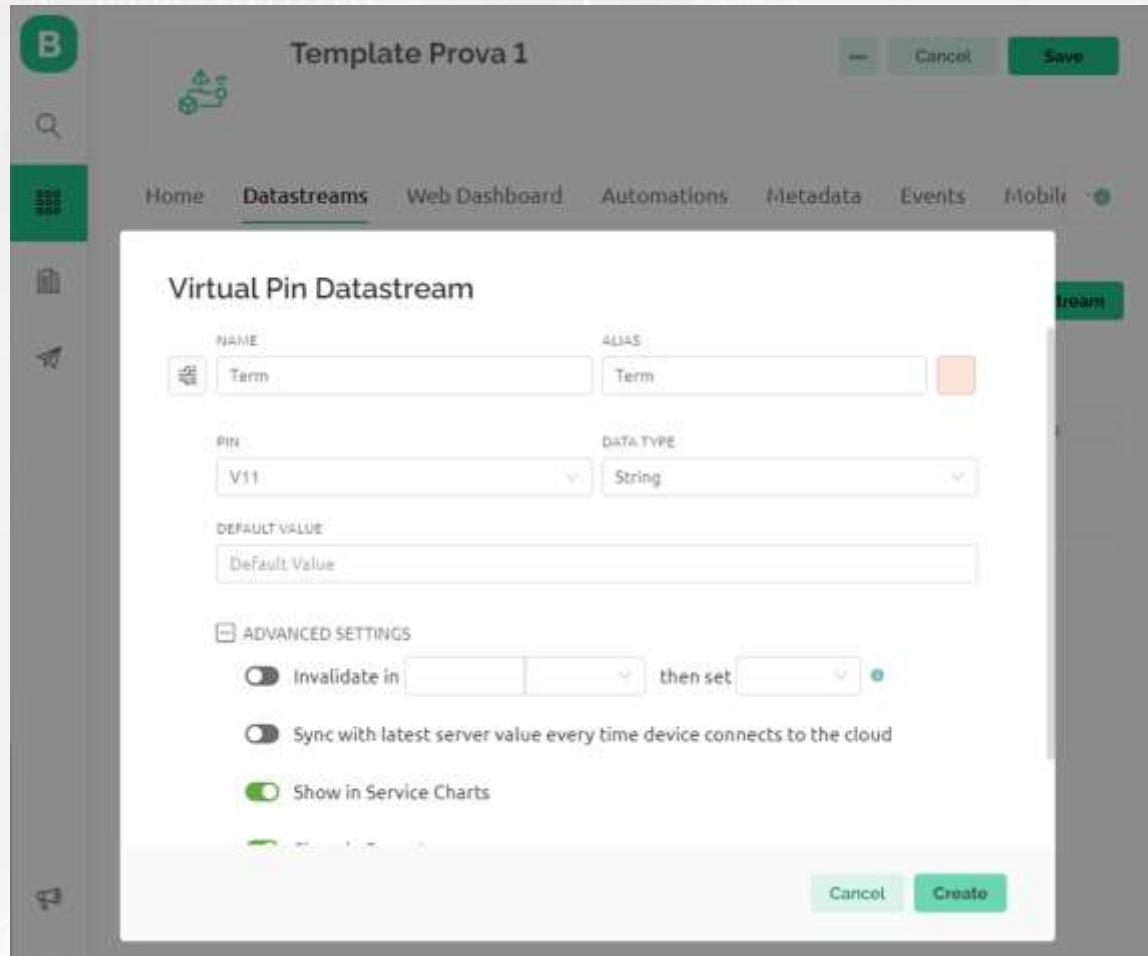
The screenshot shows the Blynk platform interface for creating a new Datastream. The title bar says "Template Prova 1". The main navigation bar includes Home, Datastreams (which is selected), Web Dashboard, Automations, Metadata, Events, and Mobile. On the left, there's a sidebar with icons for Home, Datastreams, Web Dashboard, Automations, Metadata, Events, and Mobile. The central modal window is titled "Virtual Pin Datastream". It contains the following fields:

- NAME:** Sens
- ALIAS:** Sens
- PIN:** V6
- DATA TYPE:** Integer
- UNITS:** Lux, lx
- MIN:** 0
- MAX:** 1000
- DEFAULT VALUE:** 0

At the bottom of the modal, there are "Cancel" and "Create" buttons.

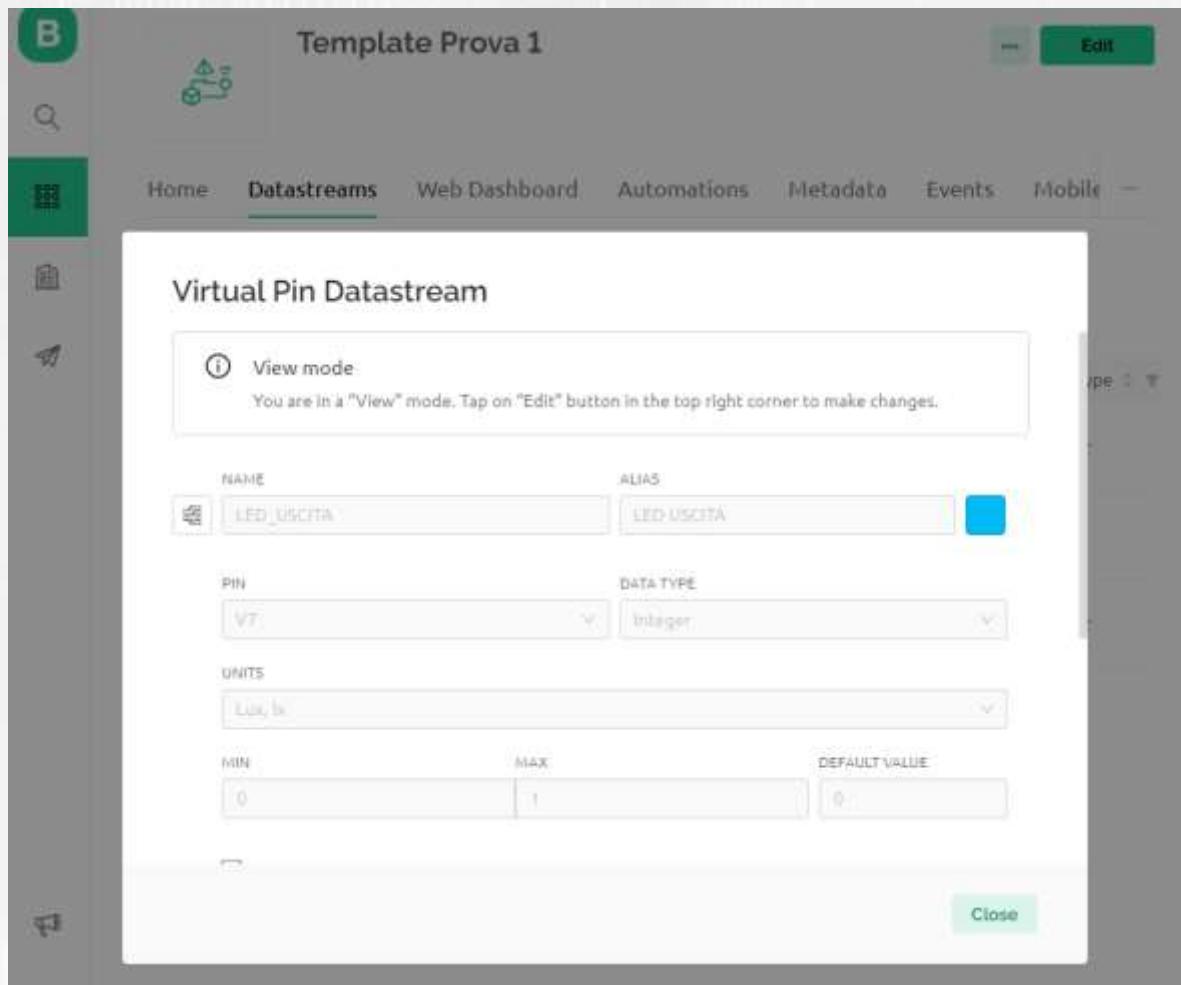
# Blynk

## Creazione Datastreams (terminale)



# Blynk

## Creazione Datastreams (LED)



# Blynk

## Datastreams

**B**

 Template Prova 1

Home Datastreams Web Dashboard Automations Metadata Events Mobile Dashboard

Search datastream

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw
1	SENS	SENS		V6	Integer	lx	false
2	TERM	TERM		V11	String		false
3	LED_USCITA	LED USCITA		V7	Integer	lx	false

# Blynk

## Datastreams

The screenshot shows the Blynk Datastreams interface. On the left is a sidebar with icons for Home, Search, Grid, and Recent. The main area has a header with a search bar and an 'Edit' button. Below is a table of datastreams:

ID	Name	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value
1	SENS	V6	Integer	lx	False	0	1000	--	0
2	TERM	V11	String		False			--	
3	LED_U	V7	Integer	lx	False	0	1	--	0

# Blynk

## Creazione Device

B

My organization - 9700AF

DEVICES

My Devices 1

All 1

LOCATIONS

My locations 0

All 0

USERS

My organization members 1

All 1

With no devices 0

Cancel

Region: fra1 Privacy Policy

### My Devices

+ New Device

#### New Device

Choose a way to create new device

From template

Scan QR code

Manual entry

Point on the cards to see instructions

# Blynk

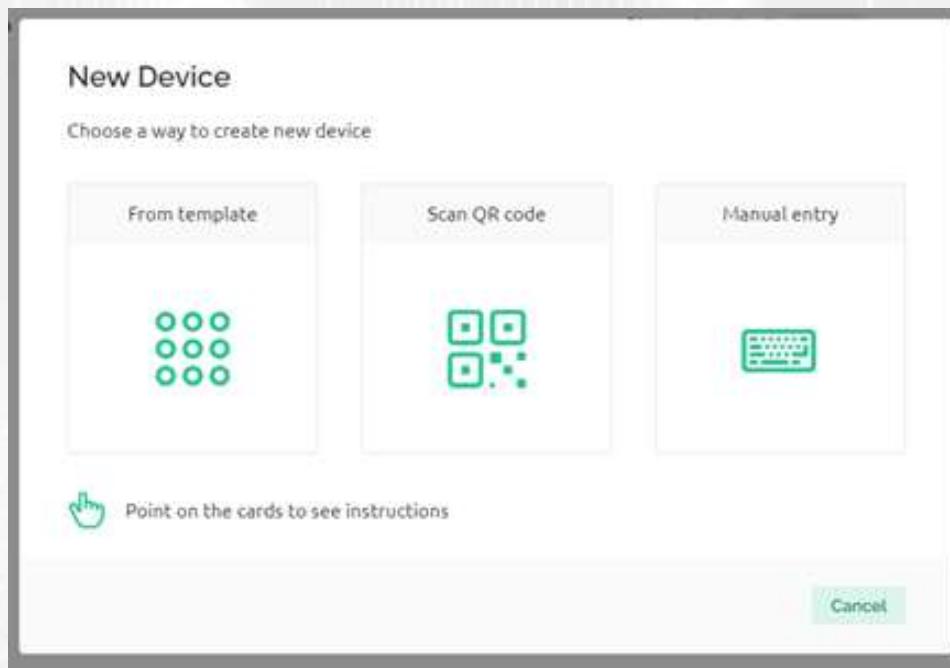
## Creazione Device

The screenshot shows the Blynk platform interface. On the left, there is a sidebar with navigation links: 'My organization - 3213NF' (with a blue 'B' icon), 'DEVICES' (selected, with a green 'D' icon), 'LOCATIONS' (with a blue location pin icon), and 'USERS' (with a user icon). The 'DEVICES' section shows 'My Devices' (1), 'All' (1), and a 'New Device' button. The main area is titled 'My Devices' and shows a table with one device entry:

Name	Auth Token	Device Owner	Status	Last Reported At	Organization	Actions
Template Prova 1	(you)	online	7:03 PM Sep 26, 2023	My organization		

# Blynk

## Creazione Device



The screenshot shows the 'New Device' configuration form. It starts with the instruction 'Create new device by filling in the form below'. There are two input fields: 'TEMPLATE' containing 'Template Prova 1' and 'DEVICE NAME' containing 'Template Prova 1'. Below the name field is the note 'Use letters, digits and spaces only'. At the bottom are 'Cancel' and 'Create' buttons.

New Device

Create new device by filling in the form below

TEMPLATE

Template Prova 1

DEVICE NAME

Template Prova 1

Use letters, digits and spaces only

Cancel Create

# Blynk

## Device

The screenshot shows the Blynk platform interface for managing devices. On the left, there's a sidebar with a search bar and a list of '1 Device'. The main area displays a device named 'Template Prova 1' with the status 'offline'. The 'Device Info' tab is selected, showing details like status (Offline), last online (7:05 PM Sep 28, 2023), device activated (5:26 PM Sep 26, 2023), auth token (Cy0W - \*\*\*\* - \*\*\*\* - \*\*\*\*), organization (My organization - 3213NF), and template name (Template Prova 1). A 'Actions' section includes a 'Click to copy Code' button, which reveals the following C code:

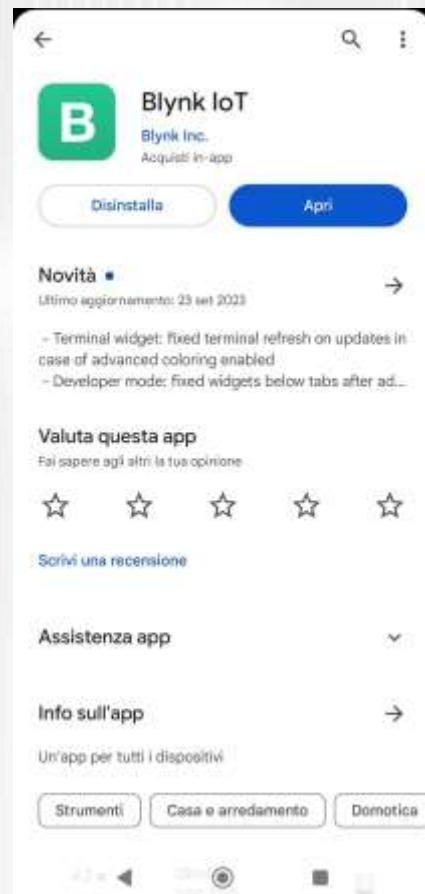
```
#define BLINK_TEMPLATE_ID "1"
#define BLINK_TEMPLATE_NAME "Template Prova 1"
#define BLINK_AUTH_TOKEN "Cy0W-****-****-****"
```

A note below the code states: 'Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.'

# Blynk

## Configurazione di Blynk su smartphone

Scaricare l'app  
Blynk IoT ed  
effettuare l'accesso



Caricare il  
template

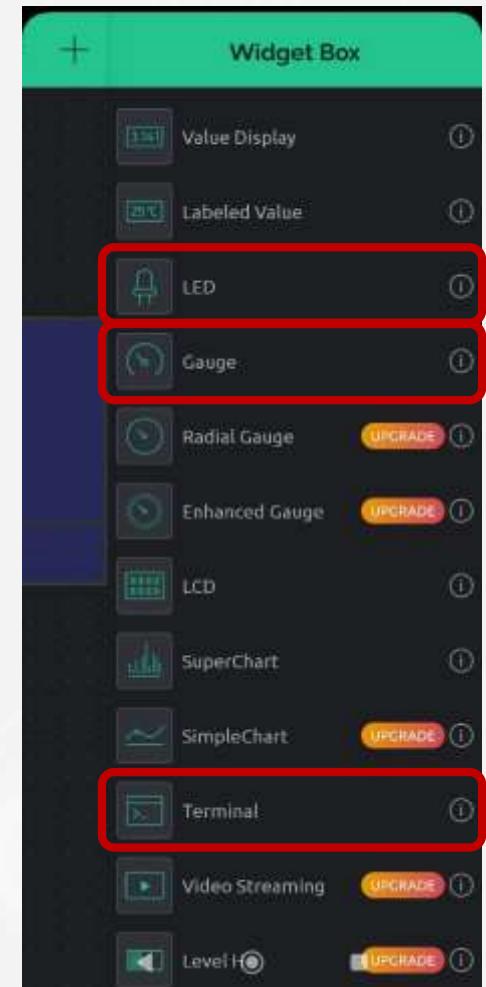


# Blynk

## Configurazione di Blynk su smartphone

Aggiungere i seguenti widget:

1. *Gauge - visualizzare il valore restituito dal sensore*
2. *Terminale - visualizzare un messaggio di avviso per segnalare il superamento della soglia*
3. *LED - accendere una spia per segnalare il superamento della soglia*

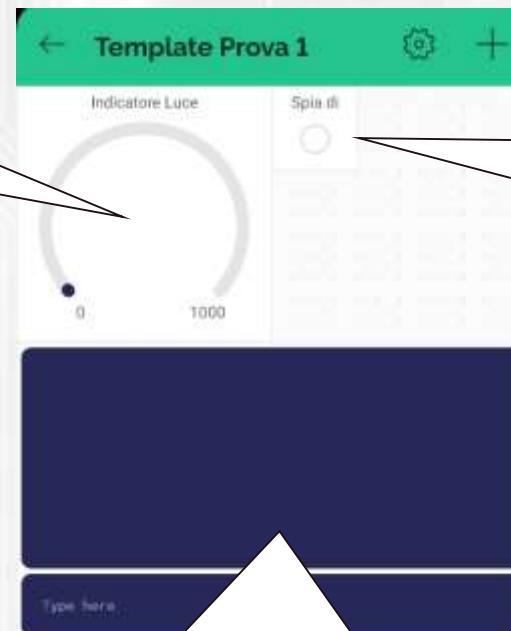


# Blynk

## Configurazione di Blynk su smartphone

Aggiungere i seguenti widget:

Gauge - visualizzare il valore restituito dal sensore



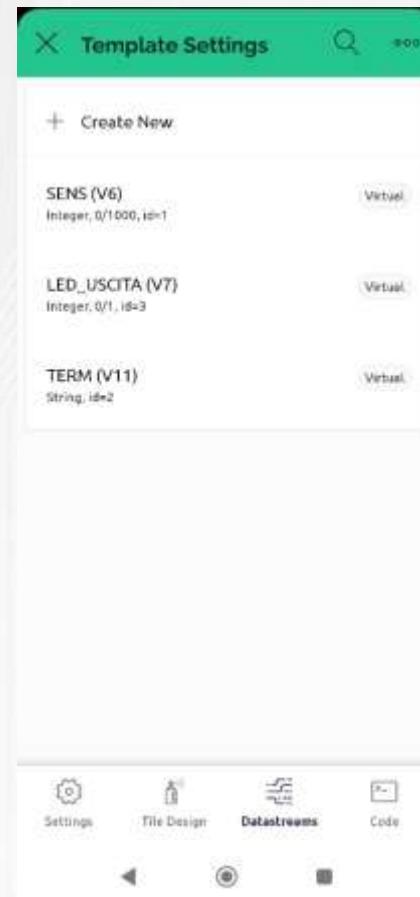
LED - accendere una spia per segnalare il superamento della soglia

Terminale - visualizzare un messaggio di avviso per segnalare il superamento della soglia



# Configurazione di Blynk su smartphone

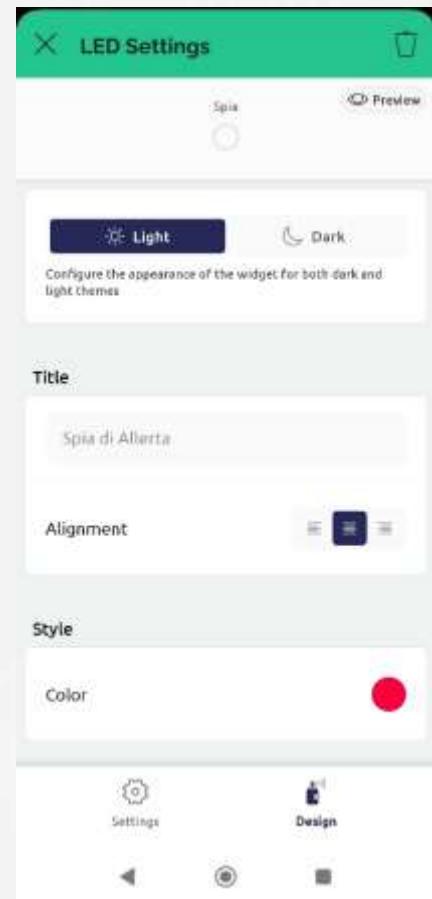
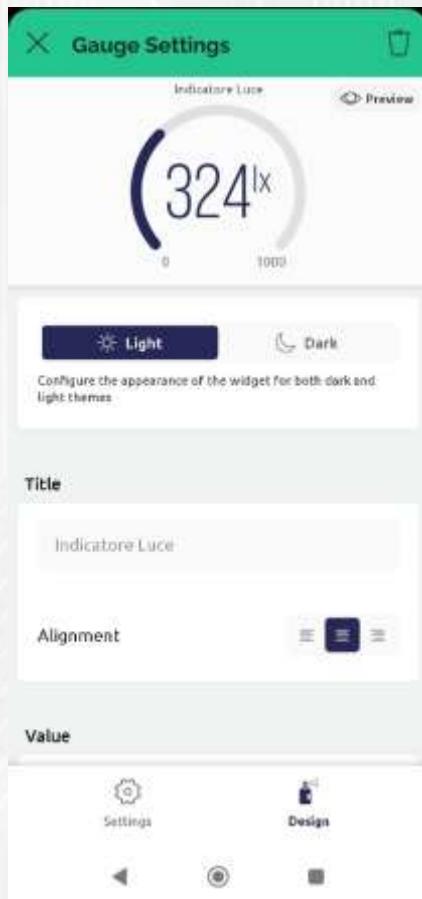
Per ogni widget selezionare il corrispondente datastream; è possibile anche gestire altre impostazioni (aspetto, colore, testo descrittivo, etc.)



# Blynk

## Configurazione di Blynk su smartphone

Per ogni widget selezionare il corrispondente datastream; è possibile anche gestire altre impostazioni (aspetto, colore, testo descrittivo, etc.)



## ESERCITAZIONE ARDUINO CON MODULO WIFI

Scopo dell'esercitazione è realizzare un sistema in grado di:

- 1) Sentire variazioni di parametri esterni e convertirli in segnali elettrici (sensore);
- 2) Elaborare il segnale elettrico proveniente dai sensori e generare segnali in uscita (elaborazione dati);
- 3) Convertire il segnale elettrico in uscita in una particolare grandezza fisica (attuatore-LED).
- 4) Mediante app Blynk IoT, visualizzare i parametri forniti dal sensore (Gauge), attivare un indicatore di allarme (LED) e visualizzare dei messaggi di allarme memorizzandone la tempistica (Terminale).



Per fare ciò una valida soluzione che permette di elaborare via codice i segnali provenienti dai sensori, attivare opportuni attuatori e la visualizzazione dei dati dei sensori sullo Smartphone è rappresentata dall'utilizzo di ARDUINO, del modulo ESP-01 e dell'applicazione su Smartphone Blynk IoT.

## Visualizzazione del livello di luminosità su App Blynk

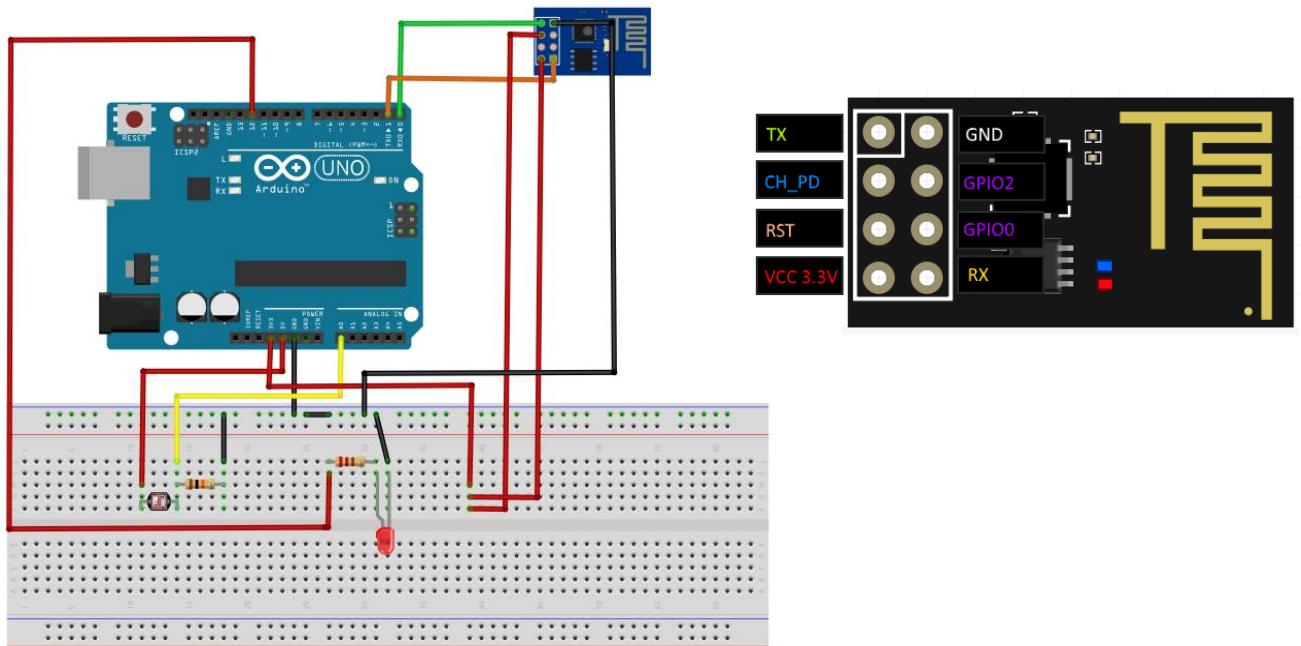
Scopo di questa esercitazione è realizzare un sistema che visualizzi su app Blynk IoT la luminosità rilevata dal fotoresistore e attivi l'accensione del LED (fisico e virtuale) quando la luminosità scende al di sotto di una soglia (fissata in base a verifica empirica). Oltre all'accensione del LED, sarà visualizzato un messaggio di allarme sul terminale con il valore del parametro luminosità - data - ora.

Software:

- Programma di base iniziale IDE
- Creazione del Template su sito Blynk.io
- Installazione app Blynk IoT

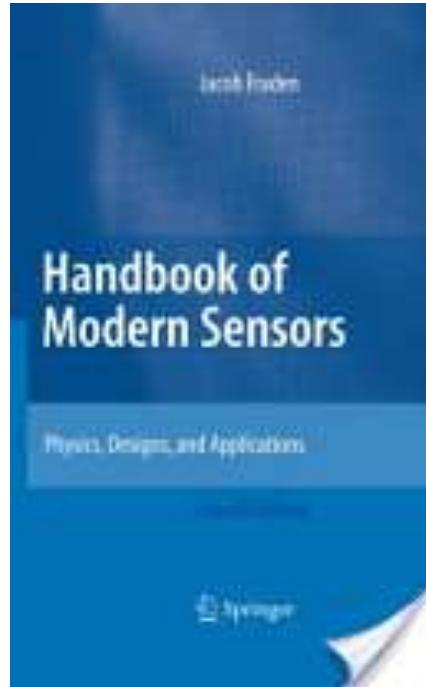
Strumentazione da laboratorio necessaria:

- Arduino UNO
- LED rosso
- Resistenza 220 Ω
- Fotoresistenza
- Cavetteria
- Breadboard
- Modulo WiFi ESP-01



**Elettronica per l'automazione  
anno accademico 2023/2024**

**Lezione  
Fondamenti di Sensori**

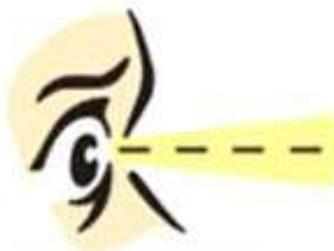


## ***Libri di testo***

- *Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications", Springer, 4th ed., 2010*
- *Jorg Haus, "Optical Sensors: Basics and Applications", Wiley-VCH Verlag GmbH, 2010*
- *J.W. Gardner, "Microsensors: Principles and Applications", Ed. John Wiley & Sons.*
- *R. Jaeger, Microelettronica II Edizione, McGraw-Hill Italia 2005*
- *A. Sedra, K.C. Smith, "Microelectronic Circuits - IV Edition", EdiSES*

# I 5 SENSI

I CINQUE SENSI / I LORO ORGANI



VISTA / OCCHIO



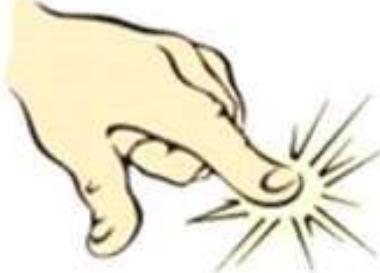
UDITO / ORECCHIO



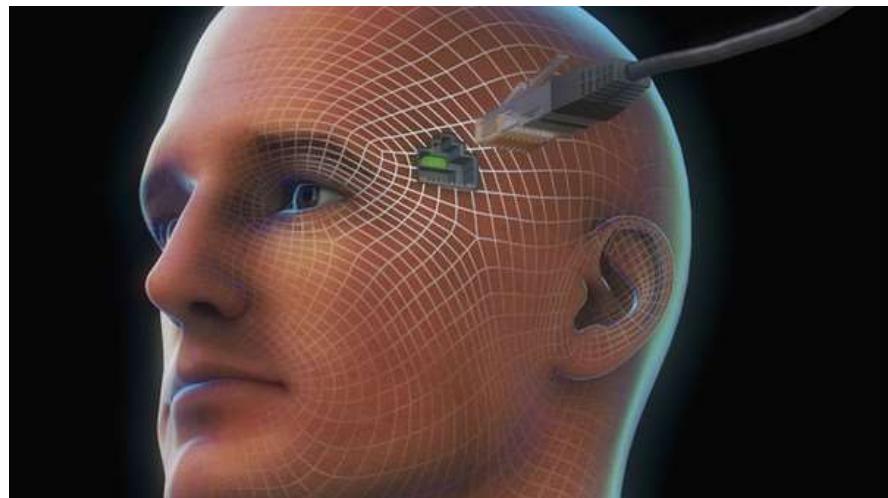
OLFATTO / NASO

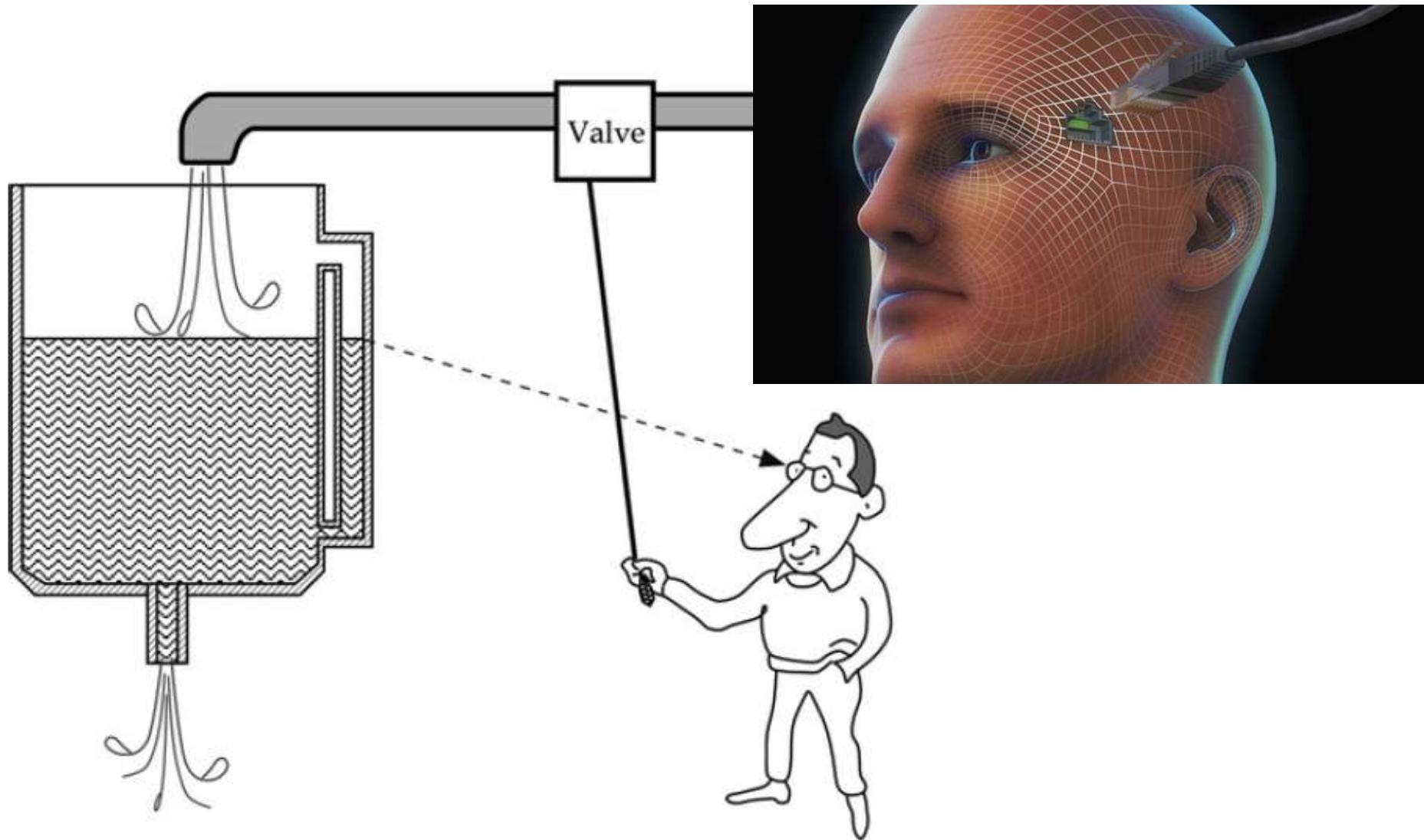


GUSTO / LINGUA



TATTO / PELLE





# **Sensore e/o Attuatore ???**

I **trasduttori**, i **sensori** e gli **attuatori** hanno la stessa funzione di base:

essi realizzano una trasformazione della grandezza fisica di ingresso per fornire in uscita una grandezza di tipo diverso.

**Conversione di energia** - **trasformazione di energia**

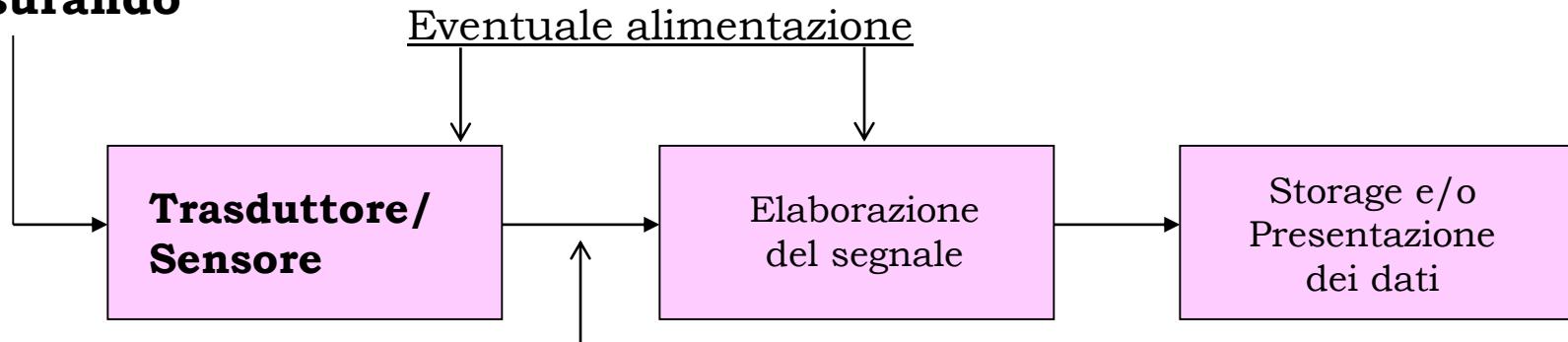
Perché allora sono stati utilizzati termini diversi?

Il **trasduttore** è un dispositivo che opera una trasformazione della grandezza fisica che ha in ingresso per restituire in uscita una grandezza fisica di tipo diverso.

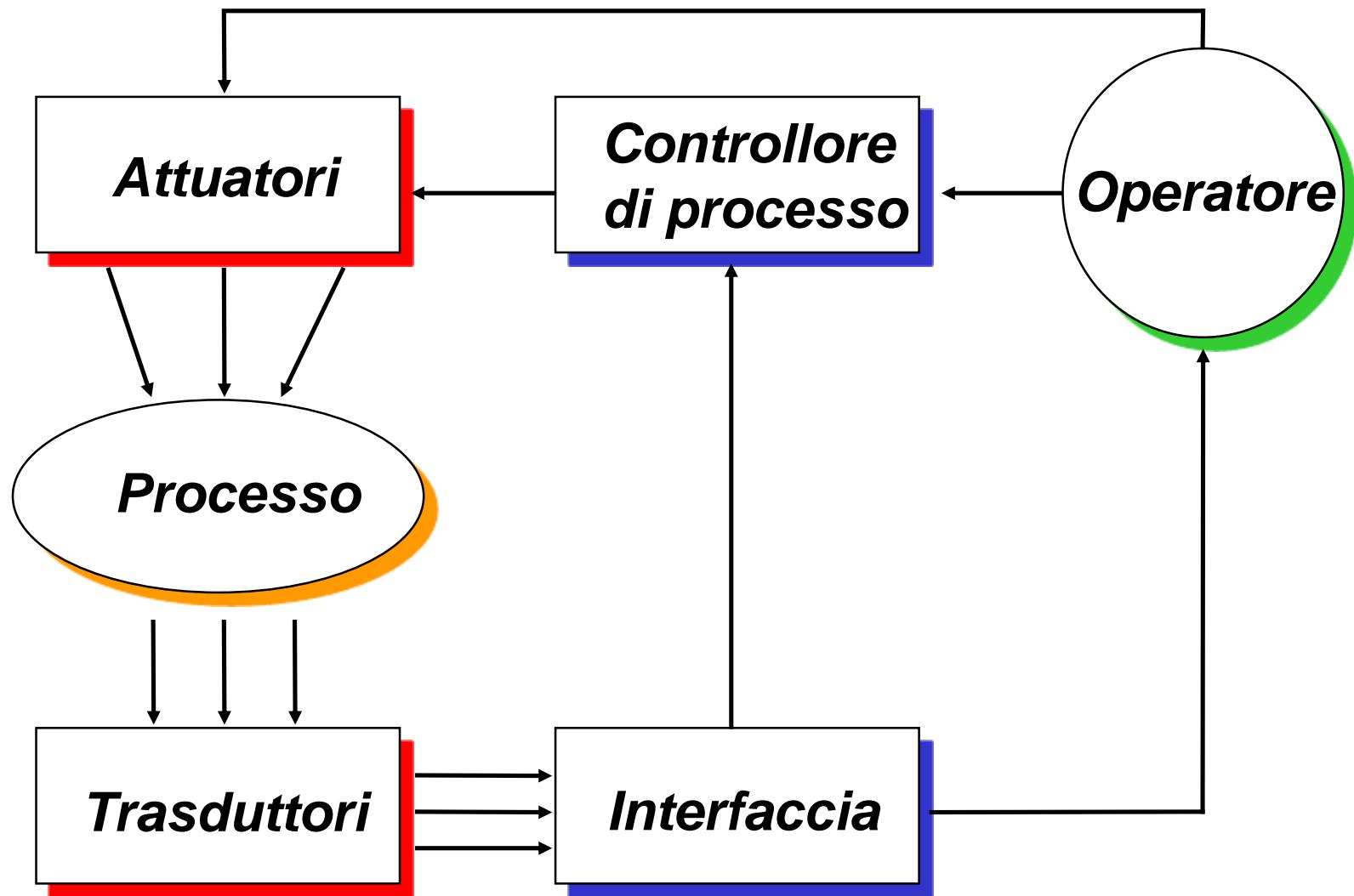
- il **sensore** è un: "particolare trasduttore che si trova in diretta interazione con il sistema misurato".
  - *È un dispositivo che converte una grandezza o uno stimolo in un'grandezza elettrica.*
  - *La grandezza di ingresso è di qualsiasi natura, mentre l'uscita è una grandezza elettrica*
- l'**attuatore** (può essere visto come l'opposto del sensore) converte un segnale elettrico in un segnale/grandezza non elettrica

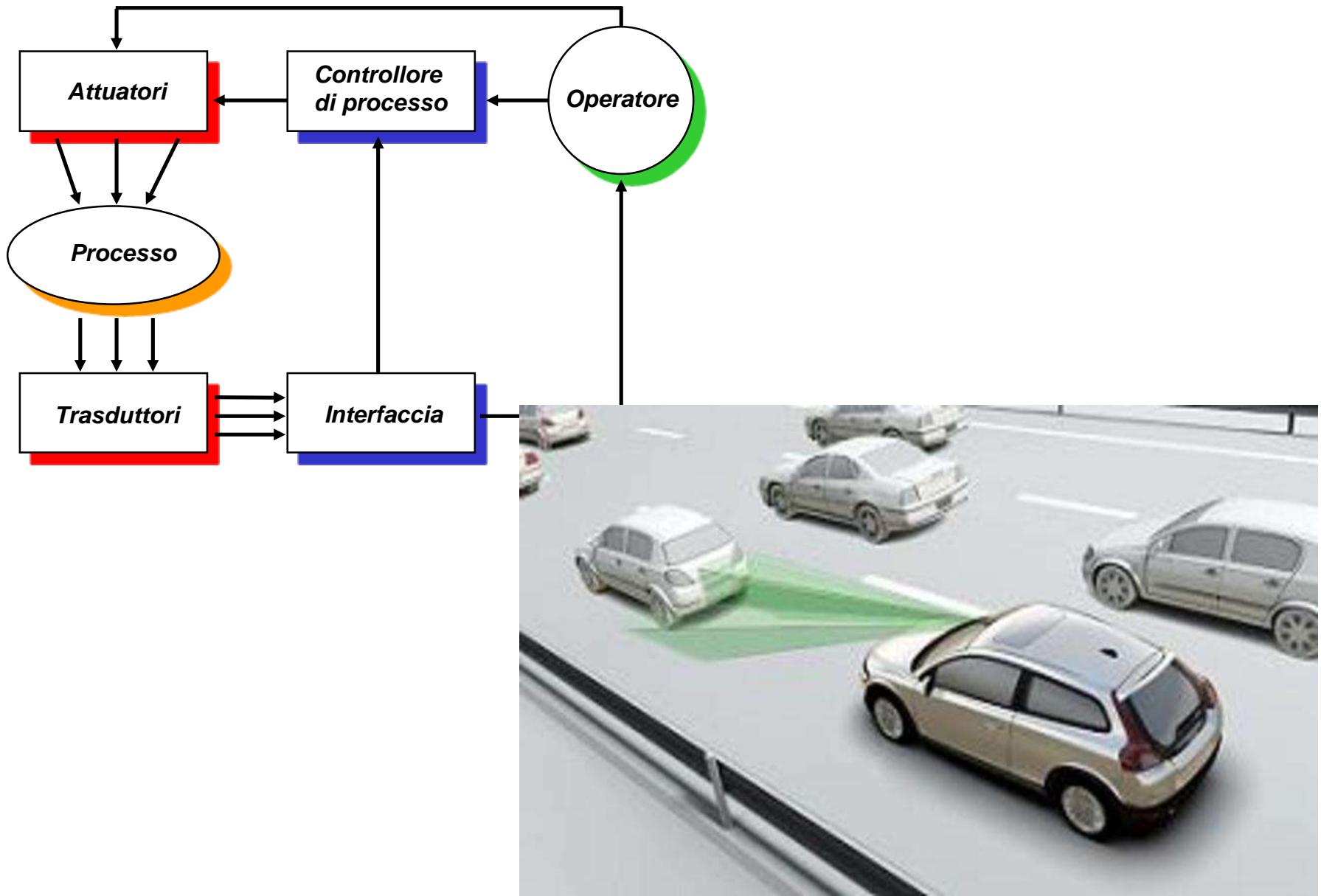
# Schema di una catena di misura

**Misurando**

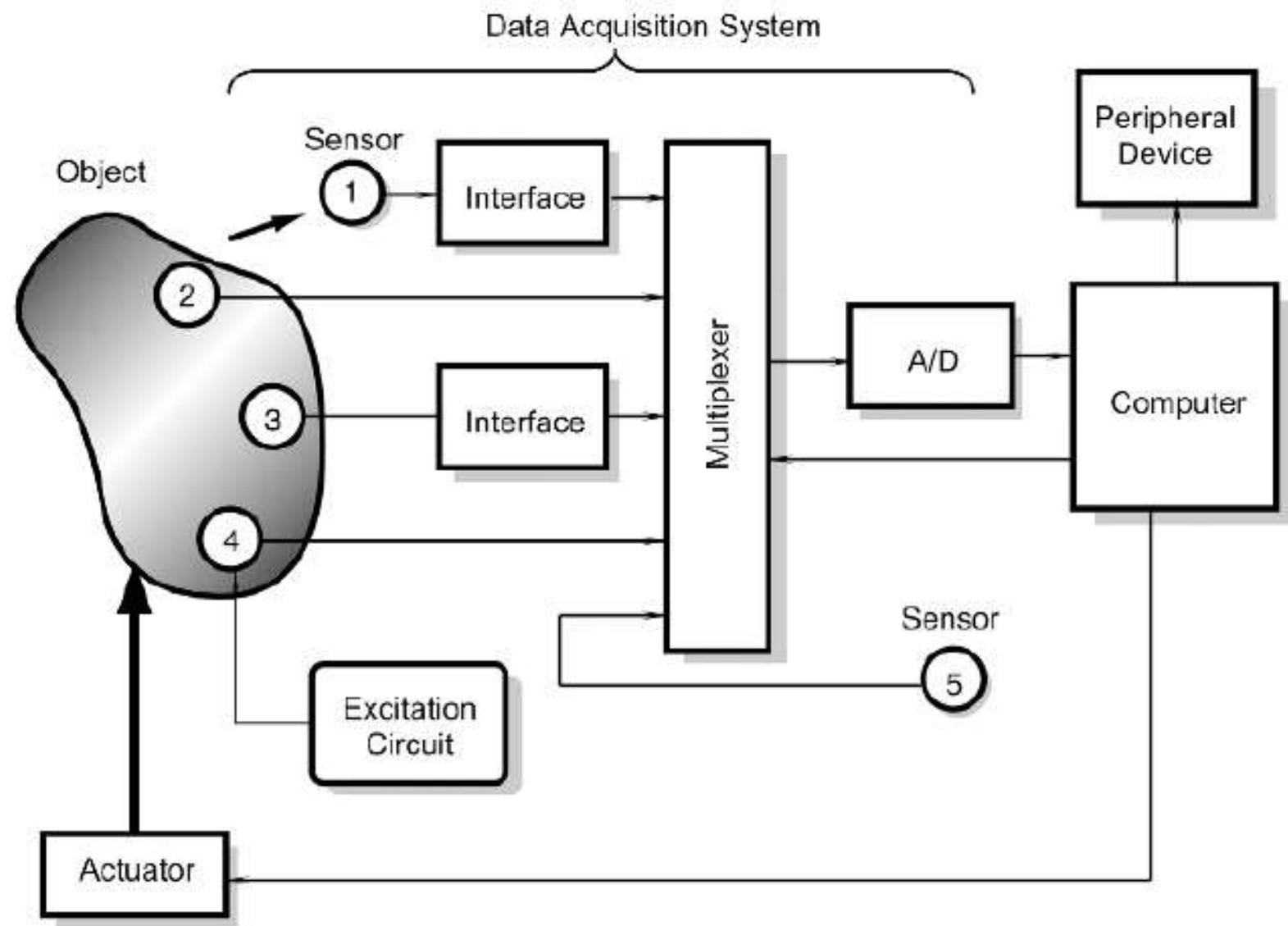


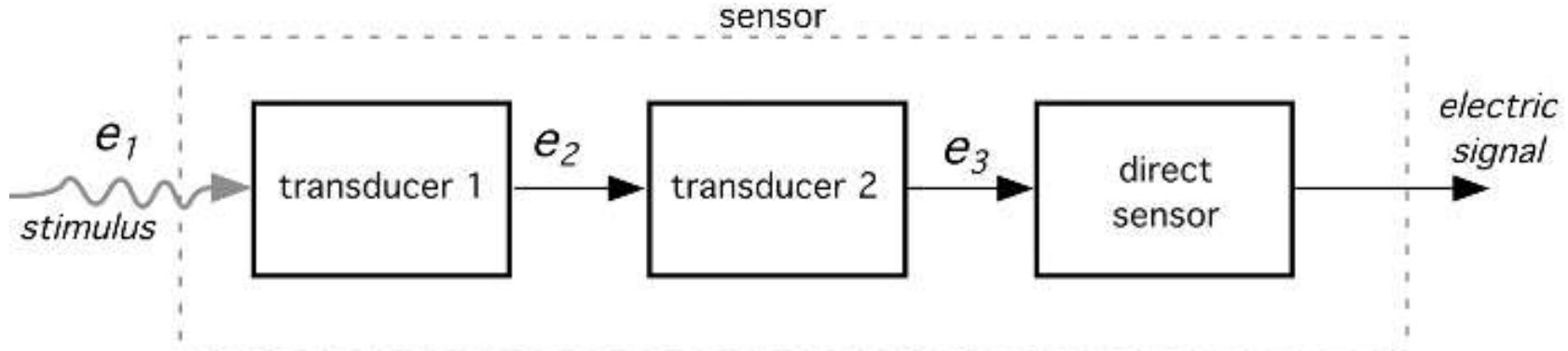
# Schema di una catena di regolazione





# Esempio: Classificazione dei sensori



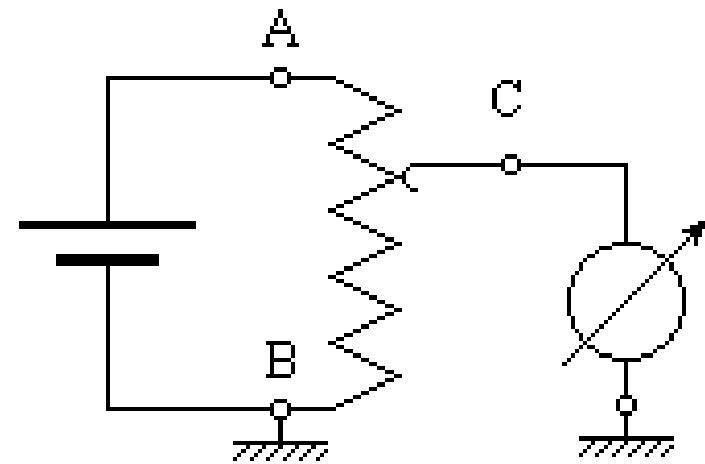
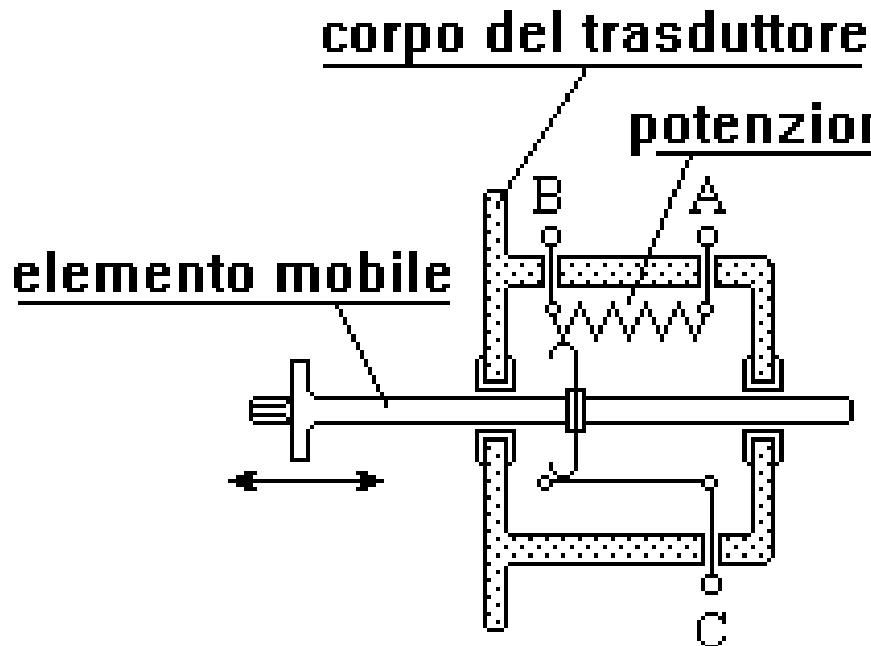


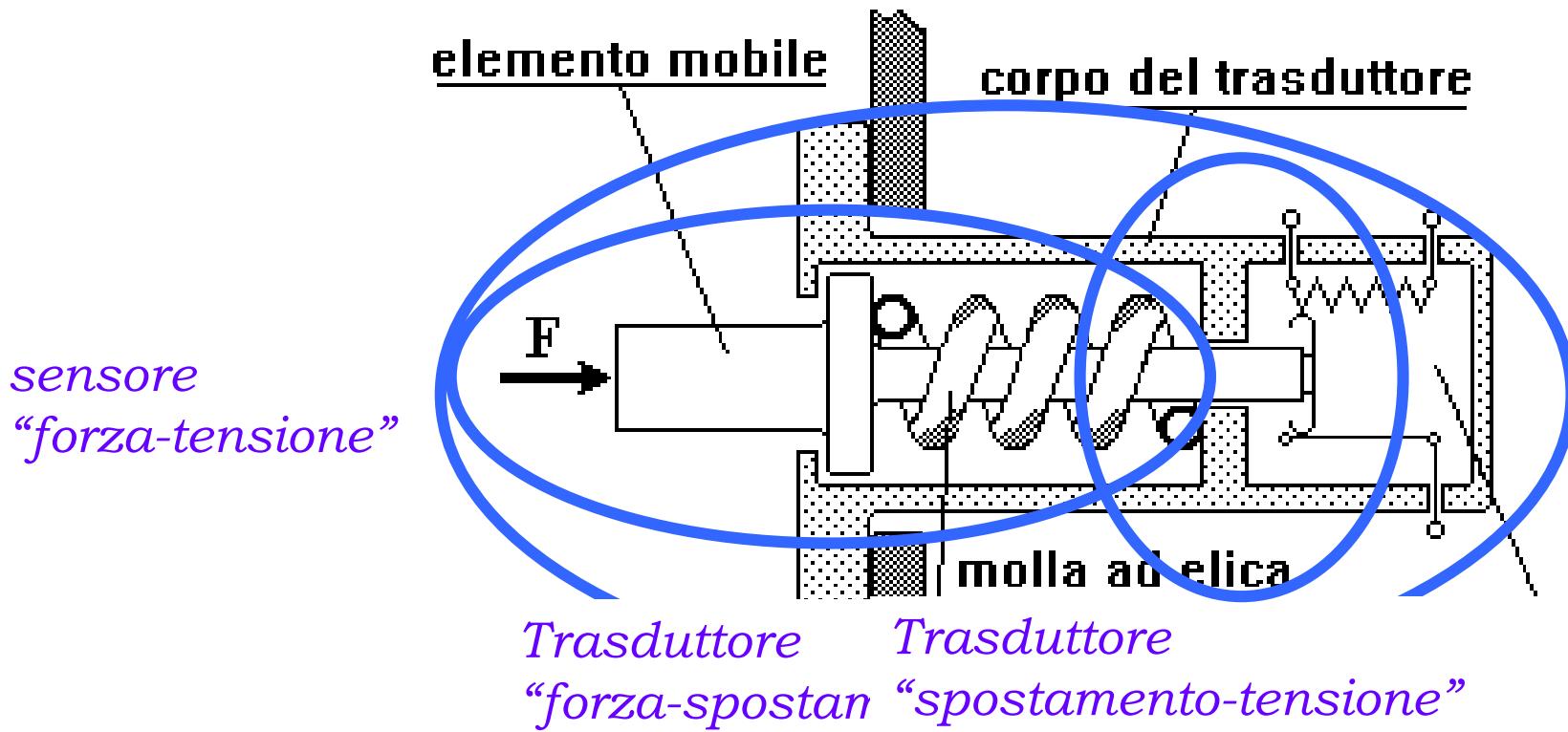
un trasduttore può essere parte di un sensore complesso

*A volte, per la complessità della trasformazione richiesta, il sensore può essere costituito da più trasduttori elementari messi l'uno in cascata all'altro*

# Sensore di posizione

## esempio sensore diretto





# **Sensori attivi e sensori passivi**

*I sensori attivi richiedono un eccitazione (alimentazione) per funzionare*

Un esempio di sensore attivo è il termistore ovvero un resistore sensibile alla temperatura

Ovviamente la temperatura modifica il valore di resistenza che per essere misurato ha bisogno di un circuito di alimentazione (alimentazione in corrente misura di tensione, alimentazione in tensione e misura di corrente)

Un altro esempio di sensore attivo è lo strain gauge resistivo ovvero una resistenza sensibile alla deformazione

Anche in questo caso è richiesto un circuito di alimentazione per la misura delle variazioni del valore della resistenza per ricavare il parametro di interesse ovvero la deformazione

# **Sensori attivi e sensori passivi**

I *sensori passivi* non richiedono la presenza di circuiti di alimentazione per funzionare e sono in grado di autogenerare un segnale elettrico in risposta ad uno stimolo esterno

Esempi di sensori passivi sono la termocoppia, il piezoelettrico, il fotodiodo.

Una termocoppia infatti autogenera una differenza di potenziale proporzionale alla differenza di temperatura per effetto Seebeck

# Tipi di sensori attivi

<b>Classi ed esempi</b>	<b>Natura del sensore</b>	<b>Misurando e applicazioni</b>
Resistore a filo	Resistenza variabile in un potenziometro	Dimensioni, spostamento
Strain-gage	Resistenza variabile con lo sforzo	Sforzi, coppie, forze, pressioni
Termometri a resistenza	Spira o termistore con variazioni di resistenza	Temperatura, effetti termici, calore irradiato
Cellule fotoconduttive	Resistenze variabili con radiazioni incidenti	Relè sensibili alla luce o a radiazioni infrarosse
Sensore a effetto Hall	Interazioni campo magnetico corrente	Forza del campo, correnti
Condensatore variabile	Variazioni di capacità per lunghezza o area	Spostamenti, pressioni
Microfono a condensatore	Variazioni capacità per pressione del suono	Voce, musica, rumori, vibrazioni

# Tipi di sensori passivi

<b>Classi ed esempi</b>	<b>Natura del sensore</b>	<b>Misurando e applicazioni</b>
Termocoppie	Metalli diversi a diverse temperature	Temperature, radiazioni, flussi di calore
Sensori piezoelettrici	Compressione del quarzo o altro cristallo	Vibrazioni, accelerazioni, suoni, pressioni
Cellule fotovoltaiche	Generazione in semiconduttori di tensione dalla luce solare	Esposizione, luce

# Classificazione dei sensori

- ✓ *Sulla base della tecnologia e del principio fisico utilizzato (ottico, piezoelettrico, ...)*
- ✓ *Funzione svolta e grandezza da misurare (m, K ...)*
- ✓ *Particolare settore cui sono destinati*
- ✓ *In base al loro comportamento energetico*

# Sensori assoluti e sensori relativi

An **absolute sensor** detects a stimulus in reference to an absolute physical scale that is independent on the measurement conditions, whereas a **relative sensor** produces a signal that relates to some special case.

- ❑ An absolute sensor is a thermistor: a thermocouple is a relative sensor. It produces an electric voltage that is function of a temperature gradient across the thermocouple wires.
- ❑ An absolute-pressure sensor produces signal in reference to vacuum. A relative-pressure sensor produces signal with respect to a selected baseline that is not zero pressure (e.g., to the atmospheric pressure).

# **Grandezze caratteristiche di un sensore**

- Grandezza Misurata**
- Principio di Funzionamento**
- Funzione di trasferimento
- Sensibilità
- Risoluzione
- Taratura
- Isteresi
- Linearità
- Tempo di risposta
- Vita utile
- Ripetibilità
- Stabilità  
(a breve e lungo termine)
- Accuratezza
- Affidabilità
- Range di ingresso
- Selettività
- Costo, dim. e peso
- Impedenza (in / out)
- Alimentazione
- Risposta in frequenza
- Fattori ambientali
  - ❖ temperatura max/ min
  - ❖ tempo di warm-up
  - ❖ umidità relativa
  - pressione max
  - ❖ presenza di gas. fumi.

# Caratteristiche di funzionamento di un sensore/trasduttore

- **Caratteristiche statiche:** descrivono le prestazioni e le condizioni ambientali per variazioni molto lente del misurando.
- **Caratteristiche dinamiche:** descrivono le prestazioni nel caso in cui il misurando subisca delle variazioni nel tempo.

# Funzione di trasferimento

La funzione di trasferimento è il legame ideale (teorico) tra stimolo fisico di input e segnale di output (segnale elettrico).

$$Y=f(s)$$

- La funzione di trasferimento può essere data in forma di tabella, di grafico o come equazione matematica
- La funzione di trasferimento può essere lineare (semplice) o non lineare (e.g., logaritmica esponenziale o ..)
- Esempi di relazioni (a, b e k parametri)

$$Y=a+b \cdot s$$

Lineare

$$Y = a+b \cdot \ln s$$

Logaritmica

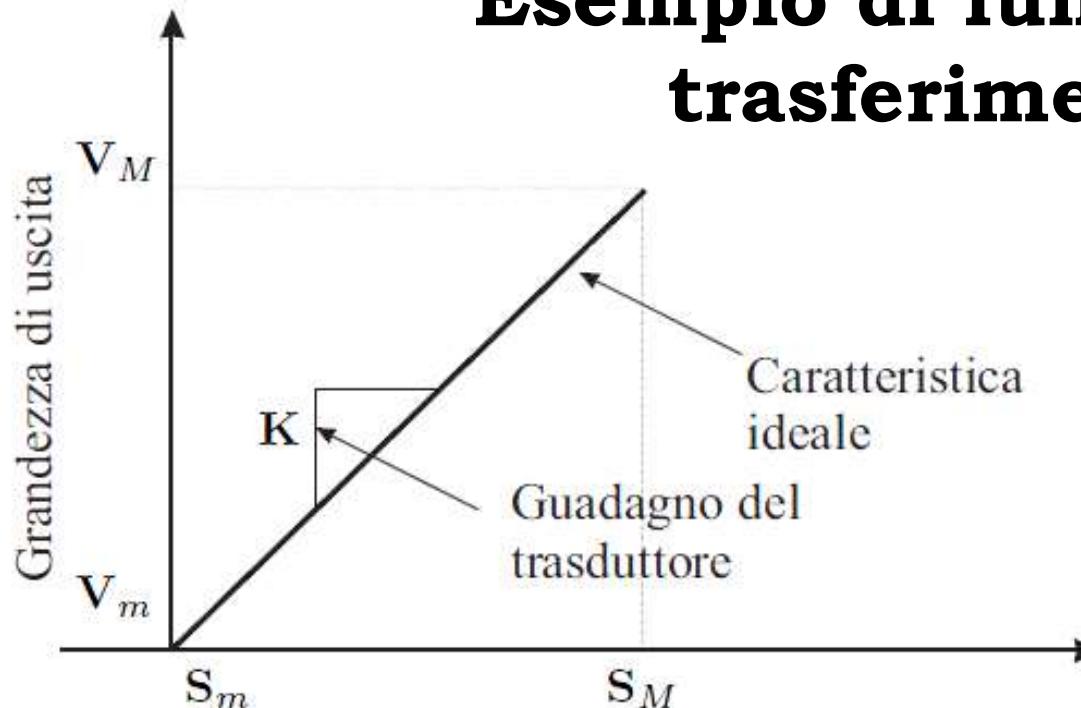
$$Y=a \cdot \exp(k \cdot s)$$

Esponenziale

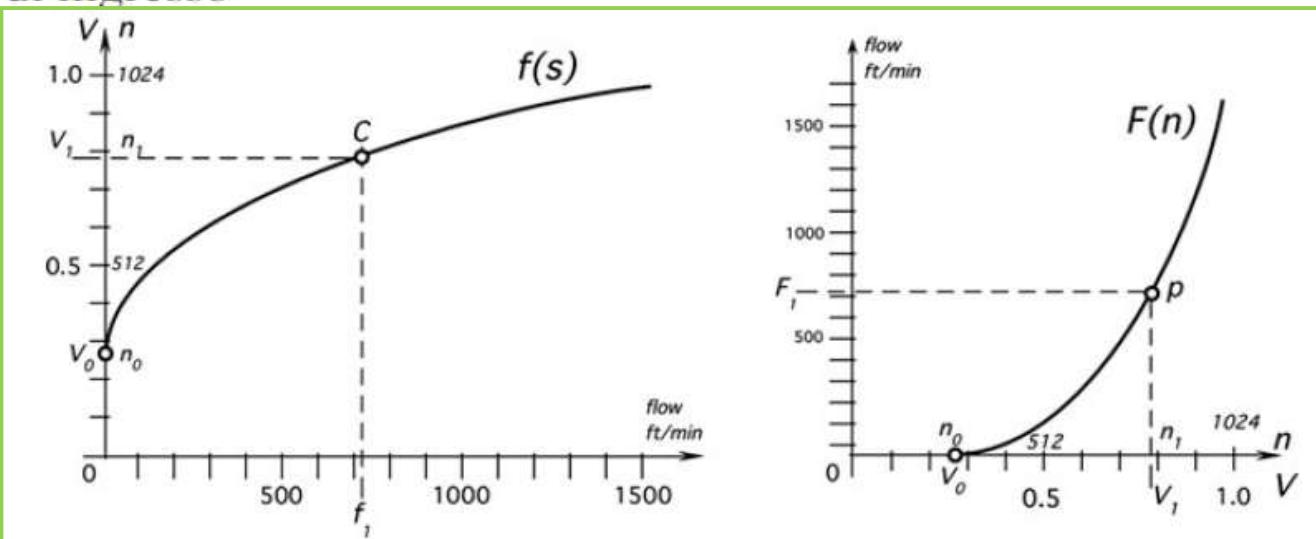
$$Y=a+b \cdot s^k$$

Potenza

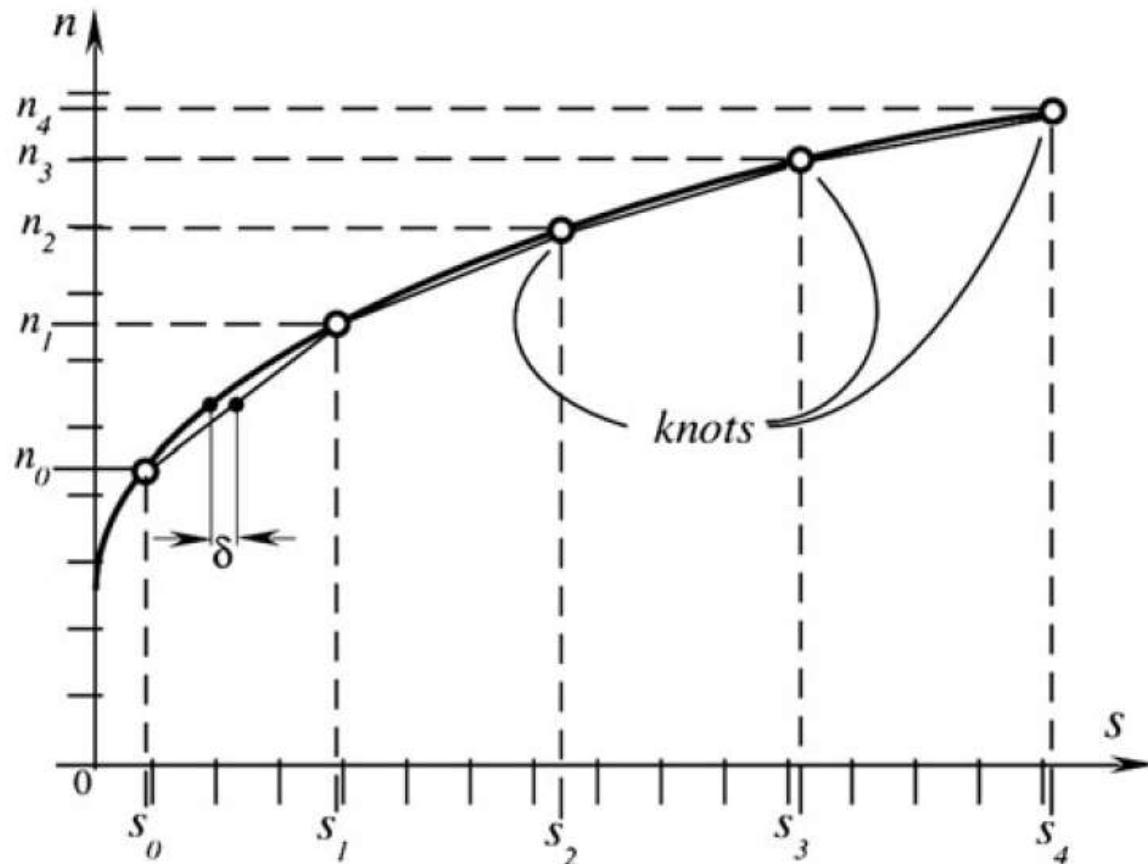
# Esempio di funzione di trasferimento



Grandezza di ingresso

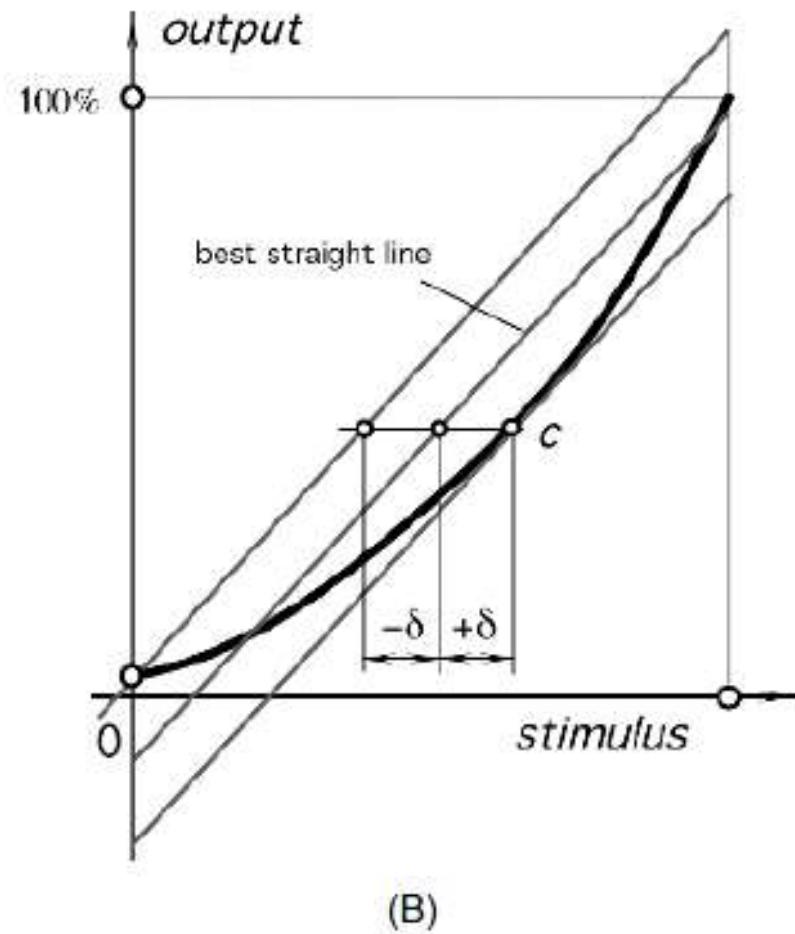
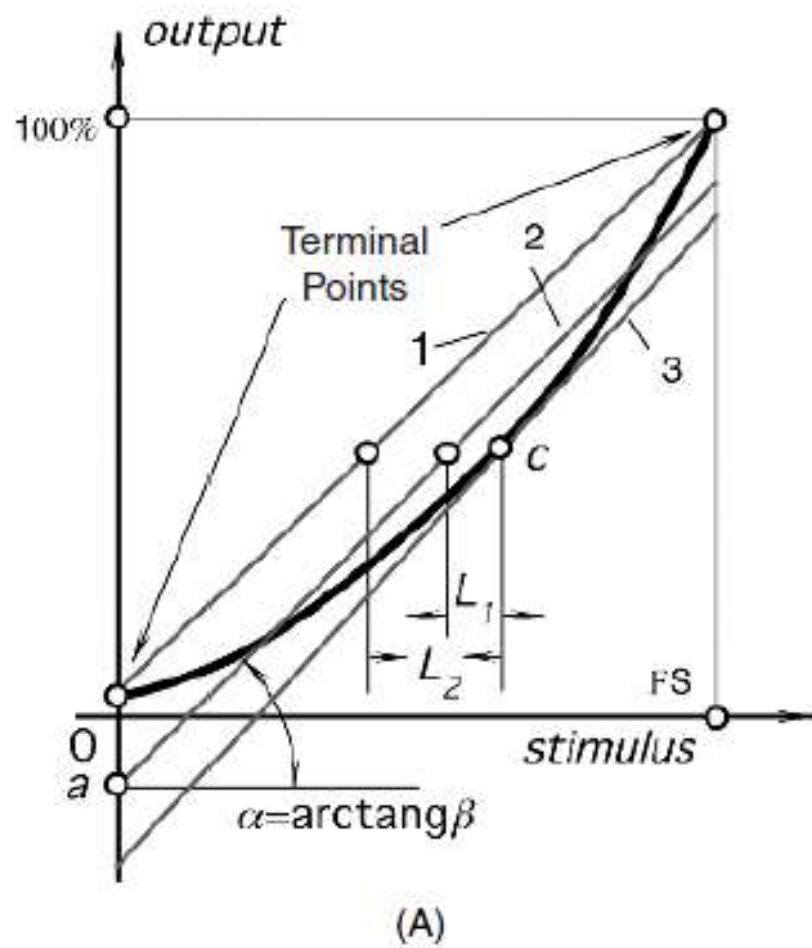


Qualche volta si utilizza una approssimazione lineare a tratti.



Linear piecewise approximation

Spesso la caratteristica reale viene approssimata con una caratteristica lineare diversa da quella teorica.



**Fig. 2.5.** Linear approximations of a nonlinear transfer function (A) and independent linearity (B).

## Esempio di funzione di trasferimento multiparametrica

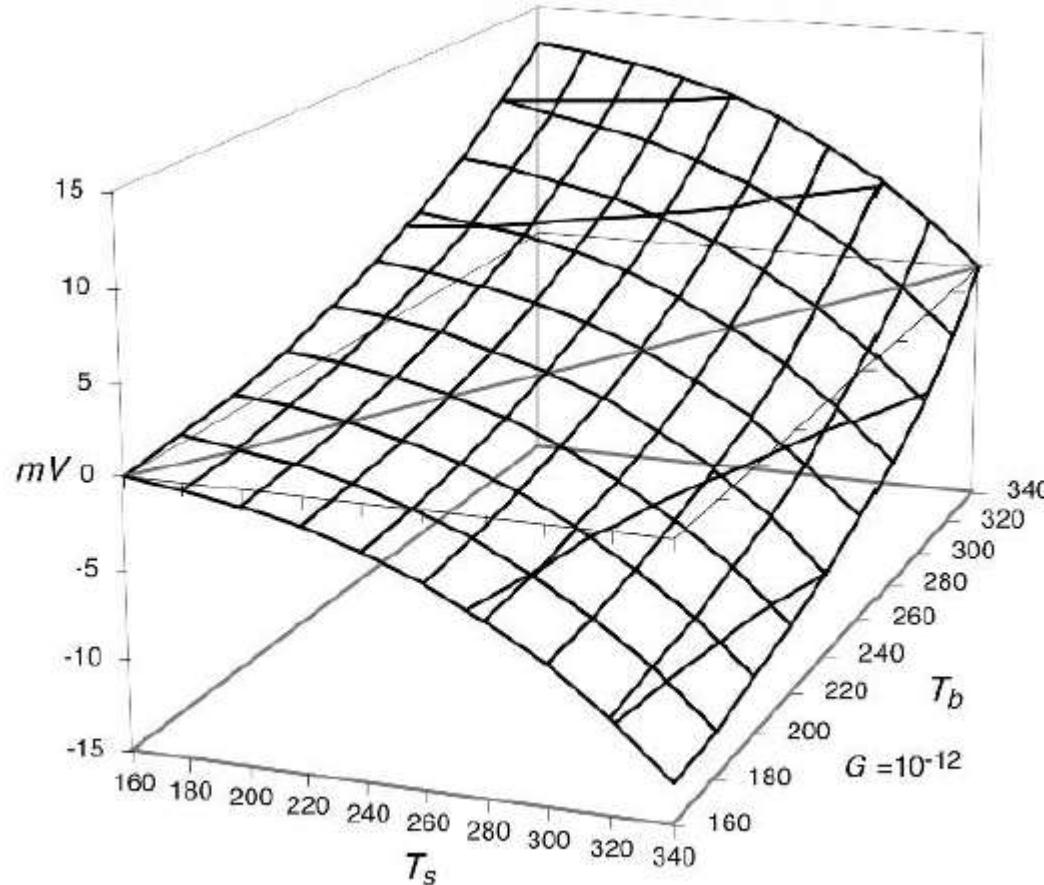


Fig. 2.1. Two-dimensional transfer function of a thermal radiation sensor.

$$V = G(T_b^4 - T_s^4),$$

# Sensibilità

La **sensibilità (sensitivity)** definisce il legame tra le variazioni dell'ingresso (misurando, stimolo) e le variazioni dell'uscita.

Una piccola variazione dell'input provoca una piccola variazione corrispondente nell'output.

**La SENSIBILITÀ è il rapporto tra la variazione dell'output e la variazione dell'input**

$$\text{Sensibilità} = \frac{dY}{ds} \Big|_{s=s_0}$$

# Caratteristica reale

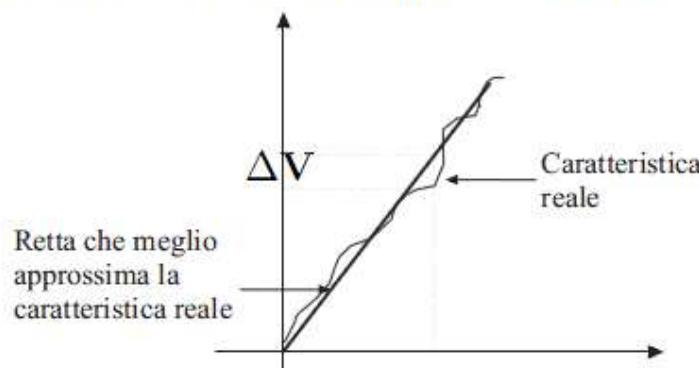
Come tutti i sistemi/dispositivi, i trasduttori/sensori reali hanno una caratteristica reale che si differenzia da quella ideale a causa di inevitabili imperfezioni costruttive.

La qualità di un sensore si misura in base a quanto la caratteristica reale si scosta da quella ideale. I costruttori di trasduttori forniscono alcuni indici che misurano la qualità del sensore.

**Errore di Linearità (linearity error).**

$$e_L \% = \frac{\Delta V}{V_M - V_m} 100$$

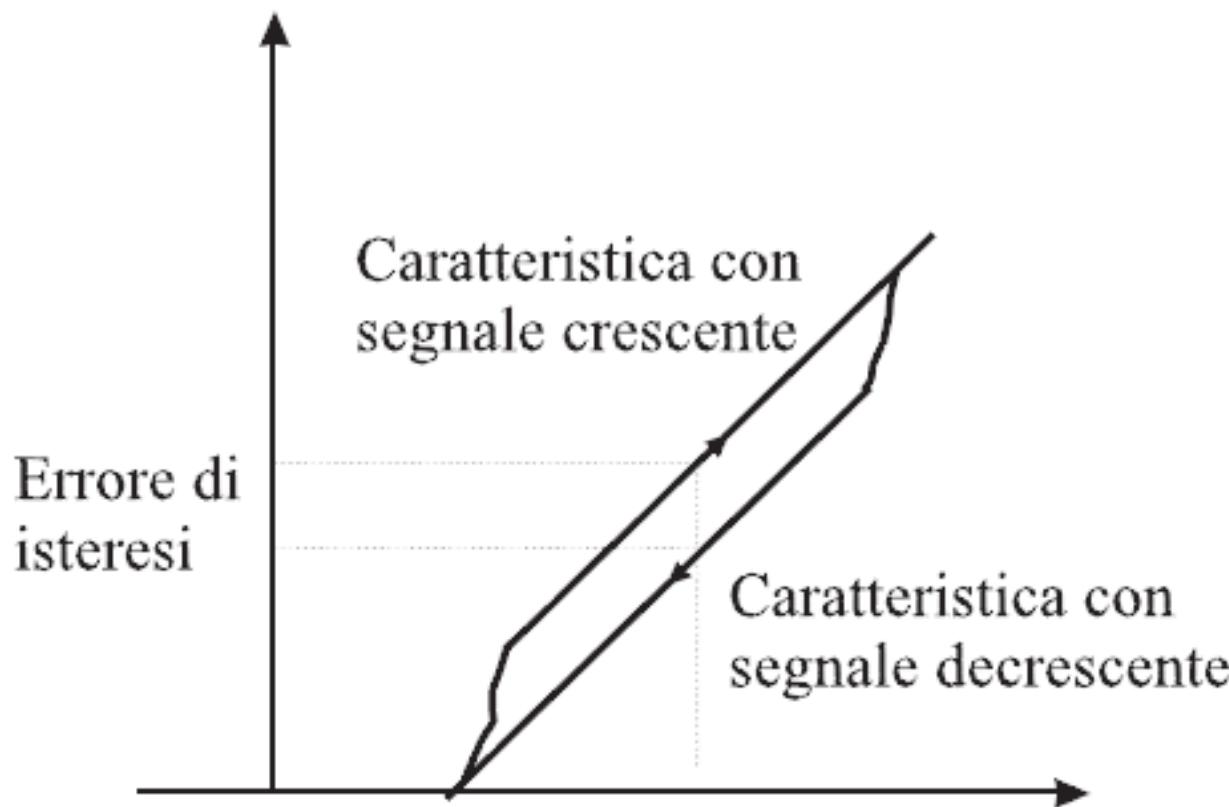
dove  $\Delta V$  è il massimo scostamento dalla caratteristica lineare



Per tener conto della caratteristica reale i singoli sensori devono essere calibrati

# Errore di isteresi (hysteresis)

Si parla di isteresi quando la caratteristica del trasduttore è differente nei casi in cui il segnale sia in crescita e quello in cui il sistema sia in diminuzione. L'errore di isteresi è la massima differenza tra le uscite nei due casi.



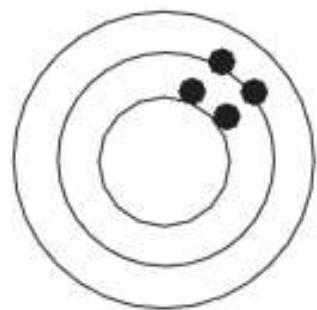
# Precisione e Accuratezza

**Precisione (precision, repeatability).** È una misura di quanto l'uscita del sensore è ripetibile su ripetuti esperimenti. In sostanza può accadere che ripetendo lo stesso esperimento più volte il sensore non fornisca l'identico risultato. La precisione di un sensore è in genere indicata fornendo un intervallo di errore massimo ( $\pm$  errore) oppure una percentuale di errore rispetto al campo di uscita:

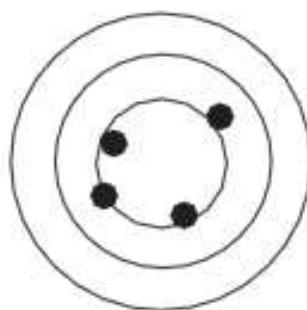
$$e_p \% = \frac{|e_{MAX}|}{V_M - V_m} 100$$

**Accuratezza (Accuracy).** È una misura di quanto l'uscita del trasduttore si scosta da quello che dovrebbe essere la sua uscita ideale, cioè quella uscita che avrebbe un trasduttore ideale che misure la stessa grandezza di quello reale.

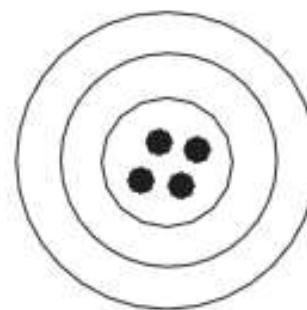
$$e_A \% = \frac{\text{errore}}{\text{valore vero}} 100$$



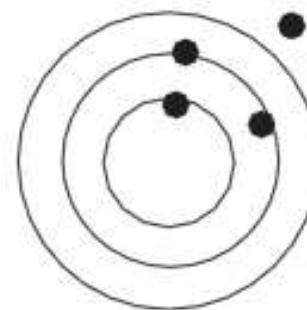
Preciso  
non accurato



Non preciso  
accurato



Preciso  
accurato



Non preciso  
non accurato

# Risoluzione

La **risoluzione** è la minima variazione nell'input che determina una variazione rilevabile nell'output del sensore.

La risoluzione dipende da:

- Sensibilità
- Sistema di lettura dell'uscita

# Span e Full Scale Output

SPAN  
[FS]

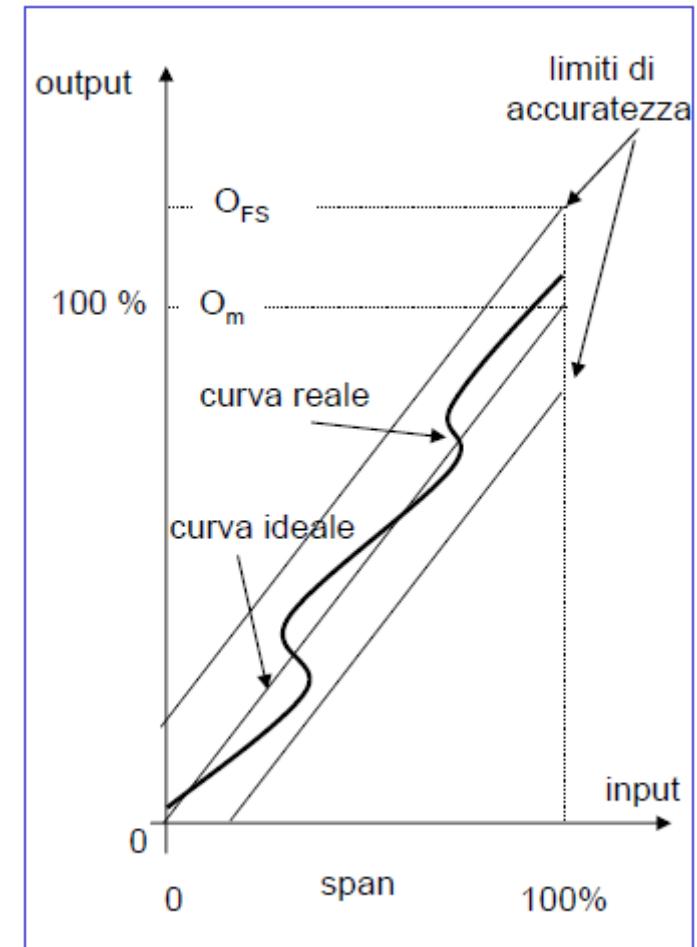
L'*input full scale* o *span* o *range* o portata è l'intervallo dinamico di ingresso che può essere convertito dal sensore con una incertezza prestabilita

## Note

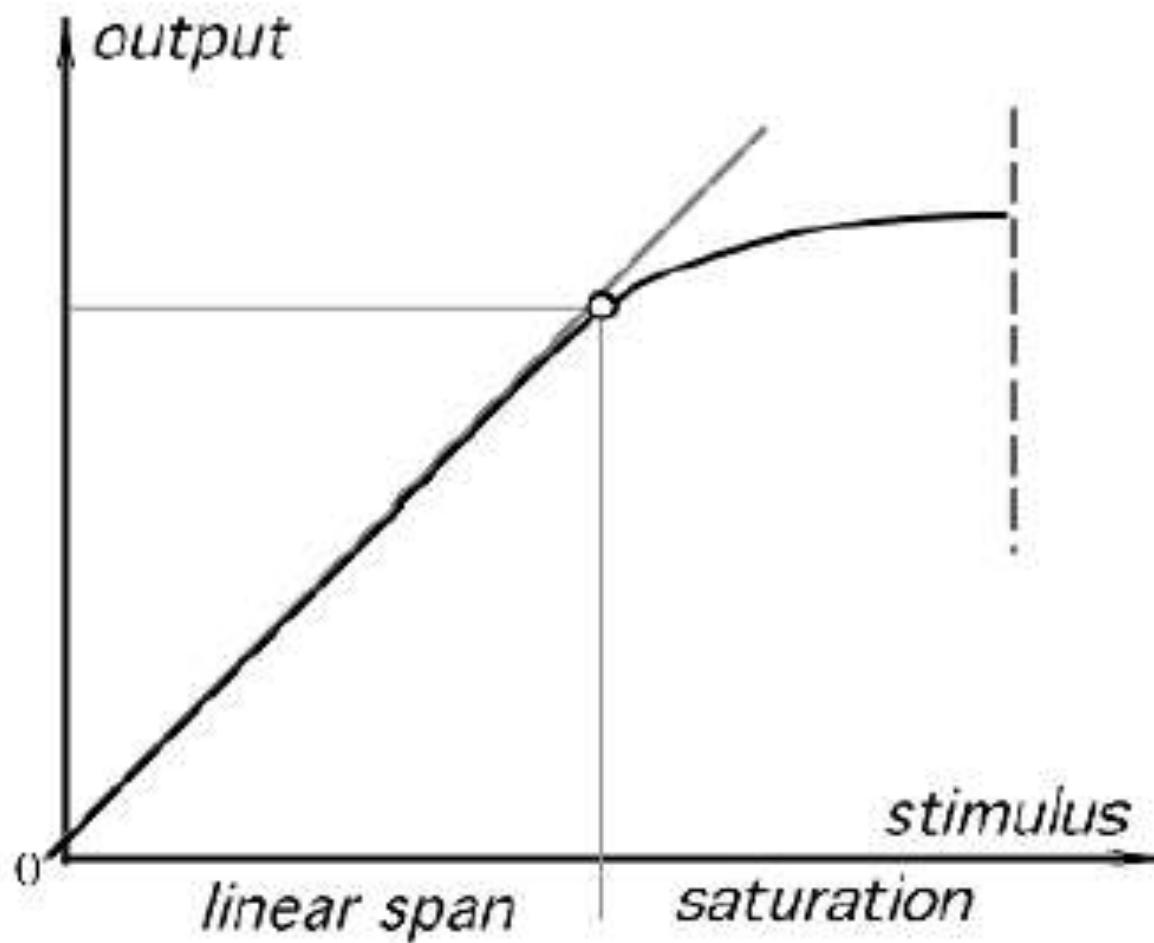
- L'*overrange* o *overload* è il massimo valore che si può applicare in ingresso al trasduttore senza comprometterne le sue prestazioni. Il tempo necessario al dispositivo per riprendere a funzionare entro le tolleranze specificate è detto *recovery time*.

Full Scale Output  
[FSO]

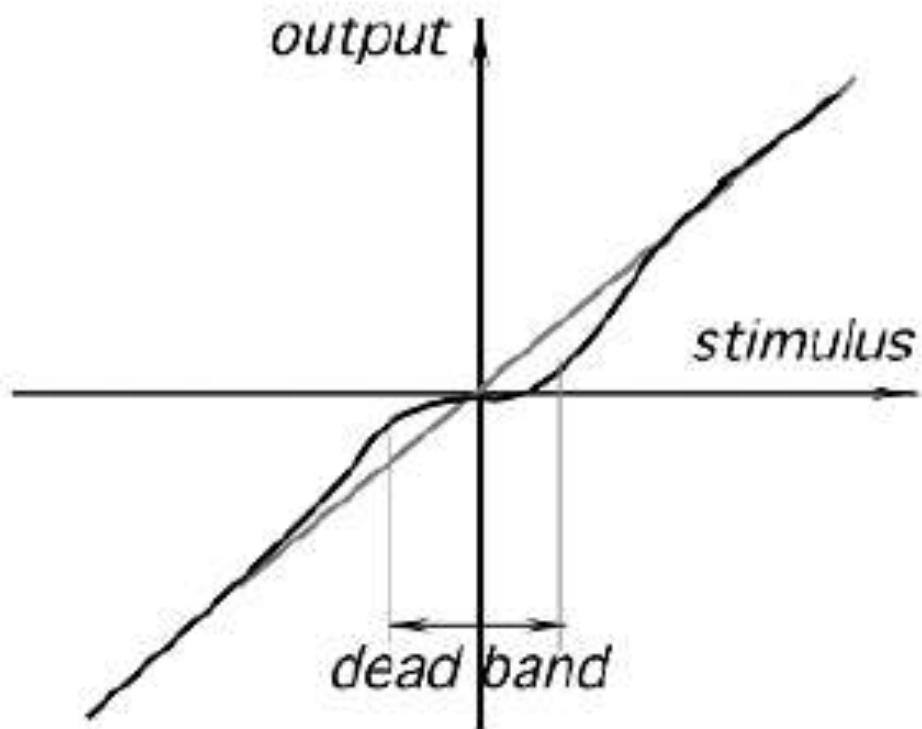
Il *full scale output* è la differenza fra i segnali di uscita del sensore quando ad esso sono applicati il più alto ed il più basso valore di ingresso



# Saturazione



# Dead band – banda morta



# Dipendenza da variabili ambientali e affidabilità

**Dipendenza da variabili ambientali.** Il trasduttore è progettato per rilevare una certa grandezza fisica, tuttavia risulta molto difficile costruire un dispositivo selettivo che non risulti influenzato da altre grandezze fisiche. Ad esempio la temperatura influenza le caratteristiche fisiche di quasi tutti i materiali, ed in particolare quelli a semiconduttore.

**Affidabilità.** L'affidabilità del trasduttore è di fondamentale importanza per il corretto funzionamento del sistema di controllo. L'affidabilità si misura in senso statistico fornendo il tempo medio che precede un guasto del dispositivo

# Caratteristiche dinamiche

Le *caratteristiche dinamiche* in un sensore descrivono il suo comportamento per variazioni rapide dell'ingresso con il tempo, in condizioni normali ed in assenza di sollecitazioni

Un sensore non sempre risponde ‘istantaneamente’ a stimoli di ingresso variabili nel tempo e questo determina un errore nella sua **risposta (errore dinamico)**. L'errore è dipendente dal tempo e può produrre delle oscillazioni se il sensore è inserito in un sistema di controllo.

Per i sensori occorre precisare l'intervallo di tempo che intercorre fra l'istante di applicazione del segnale di eccitazione e l'istante in cui inizia a funzionare entro i valori di accuratezza specificati. Questo intervallo di tempo va sotto il nome di *warm-up time*.

# Risposta in frequenza e nel tempo

## □ **risposta in frequenza [± .. % da ... a .. Hz]**

rappresenta

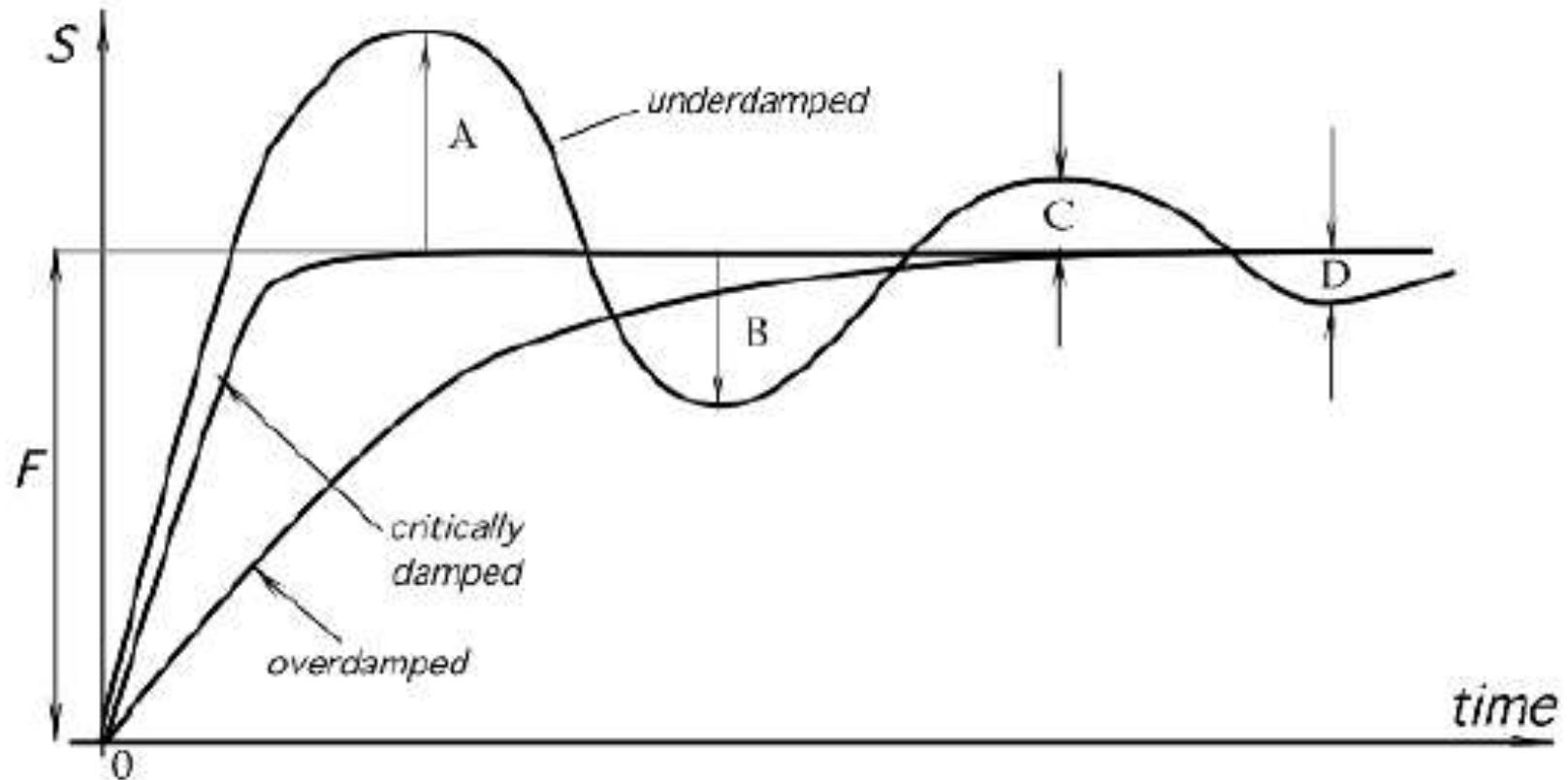
1. la variazione con la frequenza del rapporto fra ampiezza di uscita e ampiezza di ingresso, entro una prefissata banda.
2. la differenza al variare della frequenza fra la fase di ingresso e quella di uscita

## □ **risposta nel tempo [s]**

caratterizza la risposta del dispositivo ad una variazione dell'ingresso con il tempo.

- Il **tempo di risposta** è il tempo necessario affinché l'uscita raggiunga una specificata percentuale del valore finale (al 95% o al 98%).
- Il **tempo di salita** è il tempo necessario affinché l'uscita vada da un prefissato valore ad uno maggiore definiti in percentuale del valore finale ('10% - 90% oppure 5% - 90%)
- La **costante di tempo** ( $\tau$ ) è il tempo necessario affinché l'uscita raggiunga il 63% del valore finale.+

# Analisi nel dominio del tempo



# Parametri meccanici e chimici

- Sono parametri non fondamentali ai fini della misura, ma servono a contraddistinguere tipologia e modalità d'installazione:
  - ❖ *massa* (peso),
  - ❖ configurazione geometrica (*dimensioni* esterne, posizione ed orientamento delle *connessioni* meccaniche ed elettriche esterne, posizione di *montaggio*),
  - ❖ tipologia del materiale che è in contatto con il misurando; alternativamente si indicano le categorie di materiali che possono venire in contatto con il sensore, (ad es. liquidi e gas non corrosivi).
- I parametri d'identificazione (dati di targa) sono generalmente riportati sul contenitore del sensore (costruttore, modello, campo di misura, tipo di segnale d'eccitazione, uscita,.. ).

# Parametri economici

- Prezzo
- Costo e frequenza di manutenzione
- Costo taratura
- ...

Possono essere di secondaria importanza in relazione allo scopo cui è destinato il sensore.

# Laurea in Ingegneria Elettronica per l'Automazione e le Telecomunicazioni



## Laboratorio di Elettronica per l'Automazione Caso Studio: Termocoppia

**Docente del corso:**  
Prof. Andrea Cusano

**Assistente di laboratorio:**  
Ing. Giuseppe Quero  
Ing. Patrizio Vaiano

**Università degli Studi del Sannio**  
**Dipartimento di Ingegneria**  
**Gruppo di Optoelettronica**  
**Benevento**

# **Dispositivo sensoriale: Termocoppia**

**Parte 1:  
Teoria sul principio di funzionamento**

**Parte 2:  
Caratterizzazione metrologica**

# **Dispositivo sensoriale: Termocoppia**

**Parte 1:  
Teoria sul principio di funzionamento**

**Parte 2:  
Caratterizzazione metrologica**

# Definizione

La termocoppia è un SENSORE in grado di convertire la temperatura applicata ad una delle sue estremità in un segnale elettrico.

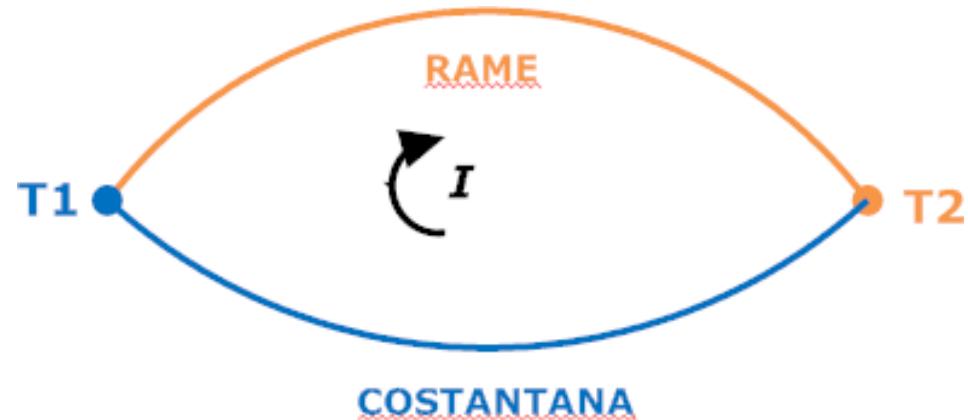
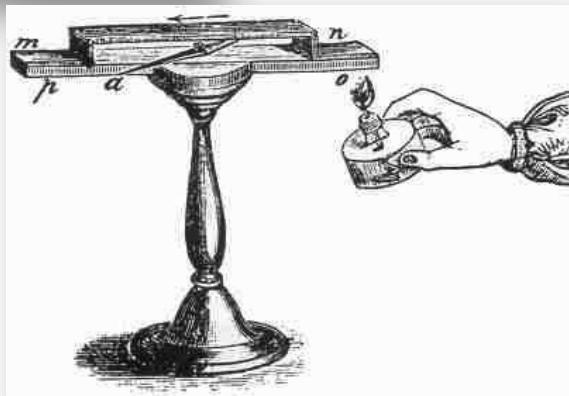


Le termocoppie sono ampiamente utilizzate perché economiche, facilmente sostituibili, standardizzate e possono misurare un ampio intervallo di temperature. Il loro limite più grande è l'accuratezza, infatti errori sistematici minori di un grado Celsius sono difficili da ottenere.

# Effetto Seebeck

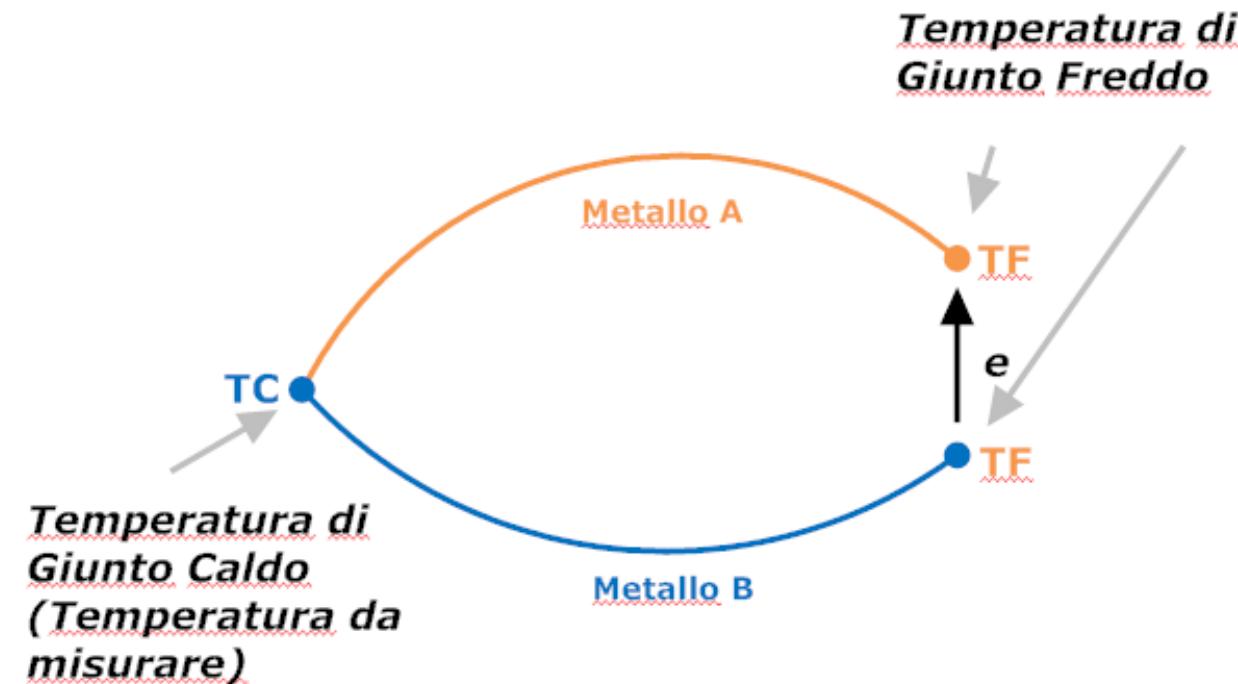
Le termocoppie sono sensori PASSIVI che generano una tensione per effetto termoelettrico.

Il principio di funzionamento è basato sull'effetto Seebeck: **quando due spire di metalli diversi sono congiunti tra loro e le giunzioni sono tenute a temperature differenti, si instaura una circolazione di corrente**



# Effetto Seebeck

Se la giunzione viene aperta si manifesta ai suoi capi una FORZA ELETTROMOTRICE INDOTTA (differenza di potenziale) e *che è approssimativamente proporzionale alla differenza di temperatura tra le due giunzioni*

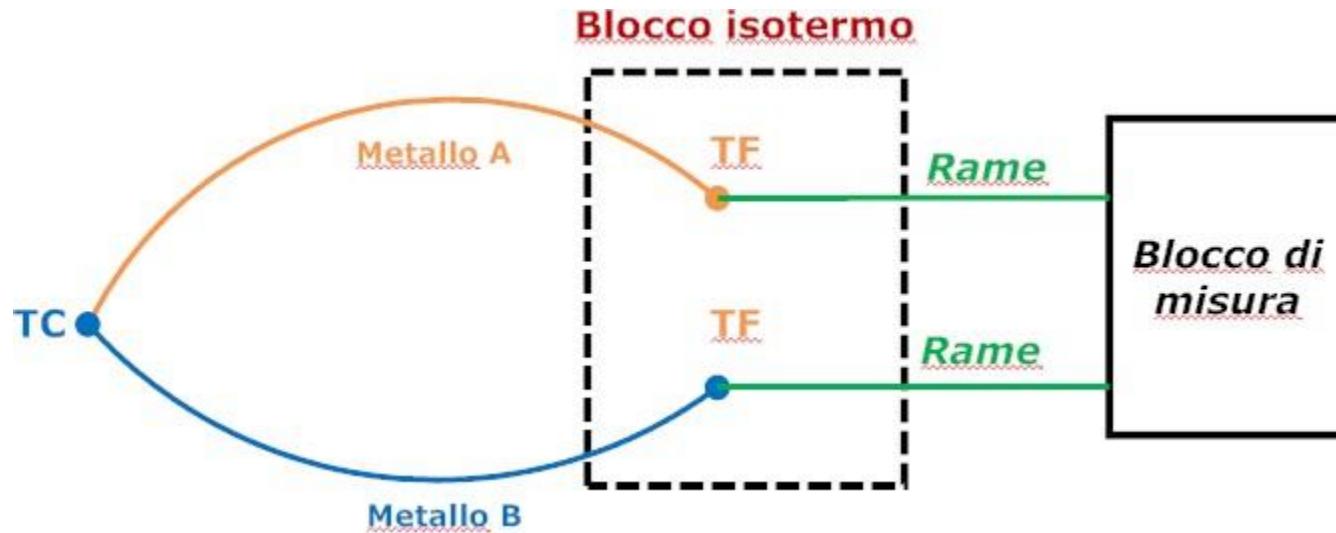


$$V = (S_B - S_A) * (T_C - T_F)$$

$S_A$  e  $S_B$  sono i coefficienti di Seebeck (o potere termoelettrico) relativi ai due metalli A e B,  $T_C$  e  $T_F$  sono le temperature delle due giunzioni.

# Effetto Seebeck

Misurando la tensione ai capi di **TF** sarà possibile identificare eventuali cambi di temperatura imposti al terminale **TC**



$$V = (S_B - S_A) * (T_C - T_F)$$

$S_A$  e  $S_B$  sono i coefficienti di Seebeck (o potere termoelettrico) relativi ai due metalli A e B,  $T_C$  e  $T_F$  sono le temperature delle due giunzioni.

# Tipologie

Tipo	Materiale	Isolamento dei conduttori	Guaina esterna	Temp. min	Temp. max	Differenza di potenziale min	Differenza di potenziale max
J	Ferro (Fe)	Nero	Nero	0 °C	750 °C	0mV	42,281mV
	Costantana (Cu-Ni)	Bianco					
T	Rame (Cu)	Marrone	Marrone	-250 °C	350 °C	-6,18mV	17,819mV
	Costantana (Cu-Ni)	Bianco					
K	Chromel (Ni-Cr)	Verde	Verde	-200 °C	1250 °C	-5,891mV	50,644mV
	Alumel (Ni-Al)	Bianco					
E	Chromel (Ni-Cr)	Viola	Viola	-200 °C	900 °C	-8,825mV	68,787mV
	Costantana (Cu-Ni)	Bianco					
N	Nicrosil(Ni-Cr-Si)	Rosa	Rosa	-260 °C	1300 °C	-4,336mV	47,513mV
	Nisil (Ni-Si)	Bianco					
S	Platino-13% Rodio	Arancione	Arancione	-50 °C	1768 °C	-0,236mV	18,693mV
	Platino (Pt)	Bianco					
B	Platino-30% Rodio	Grigio	Grigio	0 °C	1820 °C	0mV	13,82mV
	Platino-6% Rodio	Bianco					



# Il nostro sensore



ENGLISH

Datasheet

## IEC Glassfibre Exposed Welded Tip Thermocouples with Miniature Thermocouple Plug

Types K, J or N, 2 or 5 metres long with 're-wireable' miniature flat pin plug



- Thermocouple type K
- 10 metre spool length
- 7/0.2mm diameter conductor
- Operating range: -75°C to +250°C
- Teflon insulated single shot construction
- Resistant to chemicals, oils and fluids
- Suitable for aerospace applications and is widely used in autoclaves
- Colour coded to IEC-584-3

Thermocouple Type	K
Minimum Temperature Sensed	-60°C
Maximum Temperature Sensed	+350°C
Cable Length	2m
Termination Type	Miniature Plug
Response Time	Fast
Standards Met	IEC

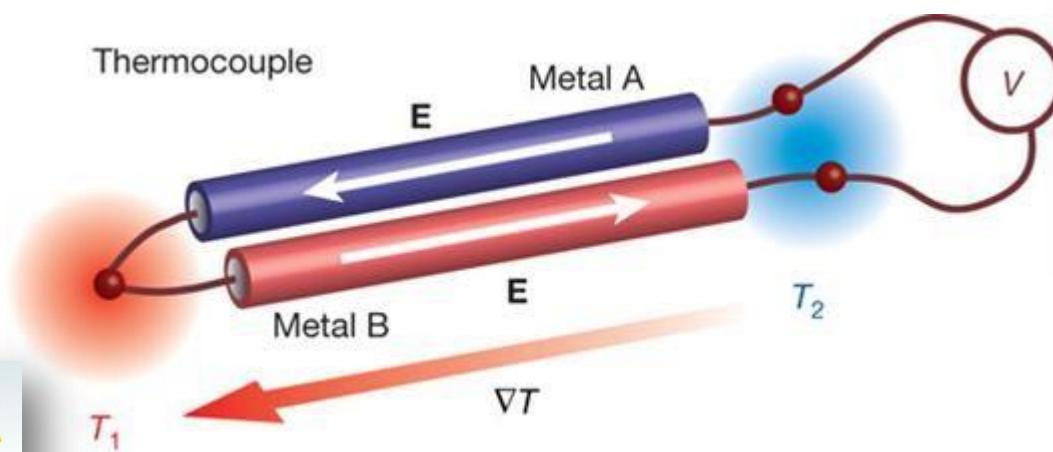
# **Dispositivo sensoriale: Termocoppia**

**Parte 1:**  
**Teoria sul principio di funzionamento**

**Parte 2:**  
**Caratterizzazione metrologica**

# Temperatura - Tensione

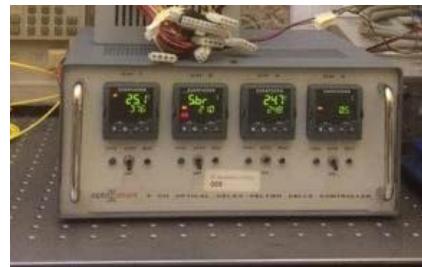
Temperatura → Trasduttore → Tensione



# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

- Termocoppia (T)
- Controllore di temperatura(C)
- Multimetro (M)



La suddetta strumentazione è collegata in modo opportuno come vedremo anche in laboratorio.



# Caratterizzazione

Caratterizzazione Tensione (V) – Temperatura (°C)

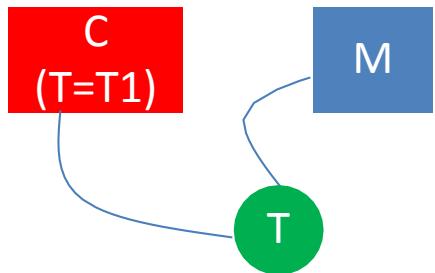


Sensibilità

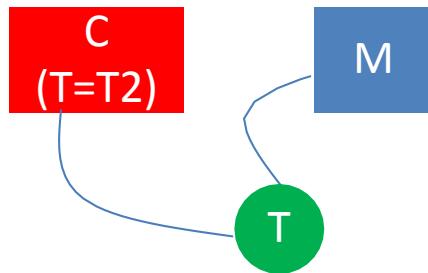
Operazione preliminare: collegare tra di loro i dispositivi in modo opportuno ed accendere la strumentazione (controllore di temperatura e multimetro)

1. leggere il valore in tensione sul multimento quando sul controllore viene raggiunta la temperatura  $T_1$
2. leggere il valore in tensione sul multimento quando sul controllore viene raggiunta la temperatura  $T_2 > T_1$
3. leggere il valore in tensione sul multimento quando sul controllore viene raggiunta la temperatura  $T_3 > T_2$
4. leggere il valore in tensione sul multimento quando sul controllore viene raggiunta la temperatura  $T_n > T_{n-1}$

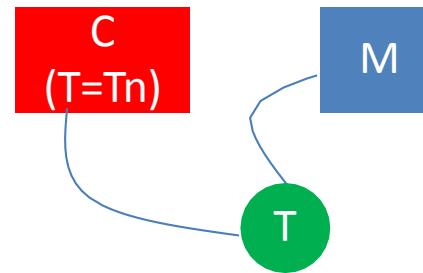
$$T = T_1 \rightarrow V = V_{out\ 1}$$



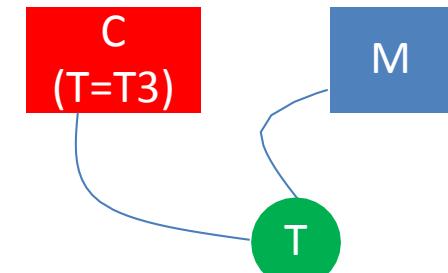
$$T = T_2 \rightarrow V = V_{out\ 2}$$



$$T = T_3 \rightarrow V = V_{out\ 3}$$



$$T = T_n \rightarrow V = V_{out\ n}$$



Calcolare le variazioni di tensione

$$\Delta V_1 = V_{out\ 2} - V_{out\ 1}$$

$$\Delta V_3 = V_{out\ 3} - V_{out\ 1}$$

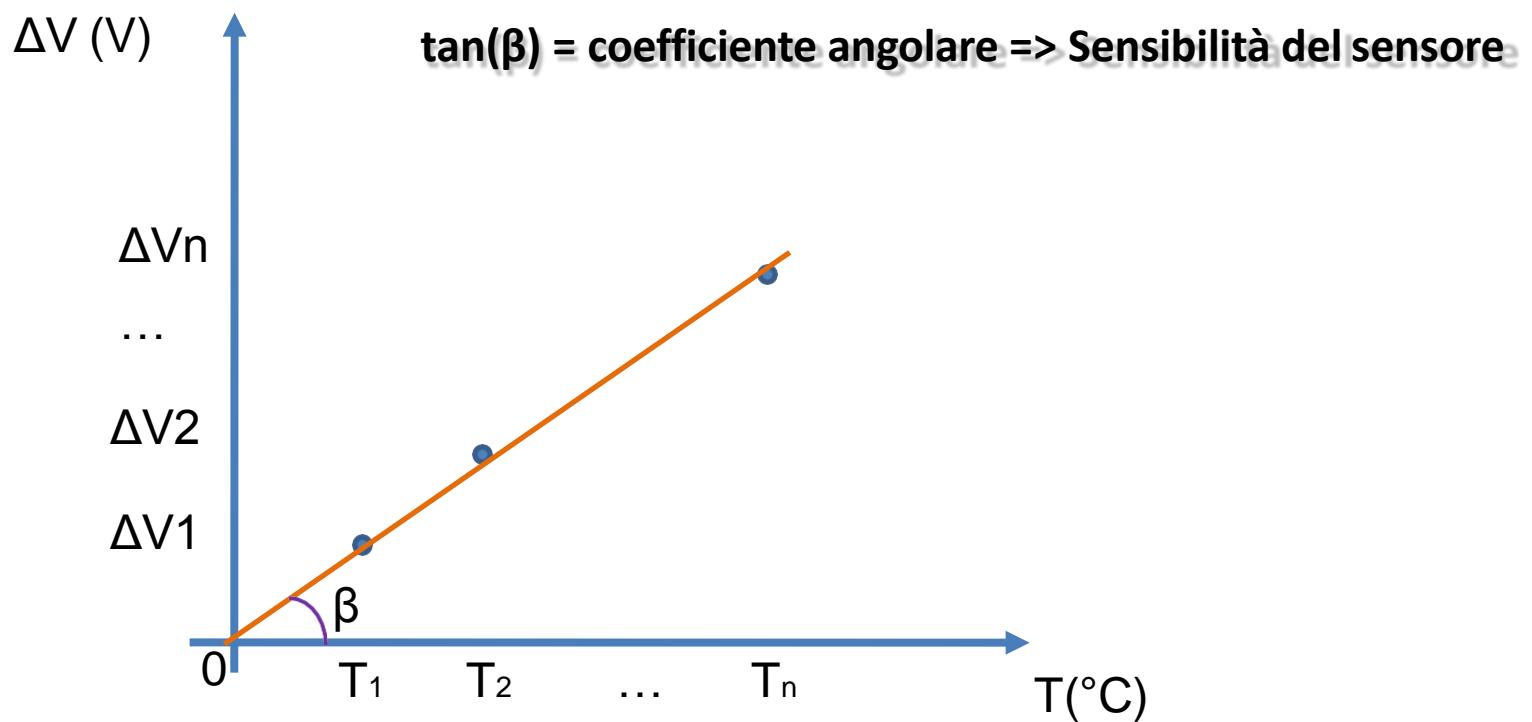
$$\Delta V_2 = V_{out\ 3} - V_{out\ 1}$$

$$\Delta V_n = V_{out\ n} - V_{out\ 1}$$

# Caratterizzazione

Riportare su un grafico:

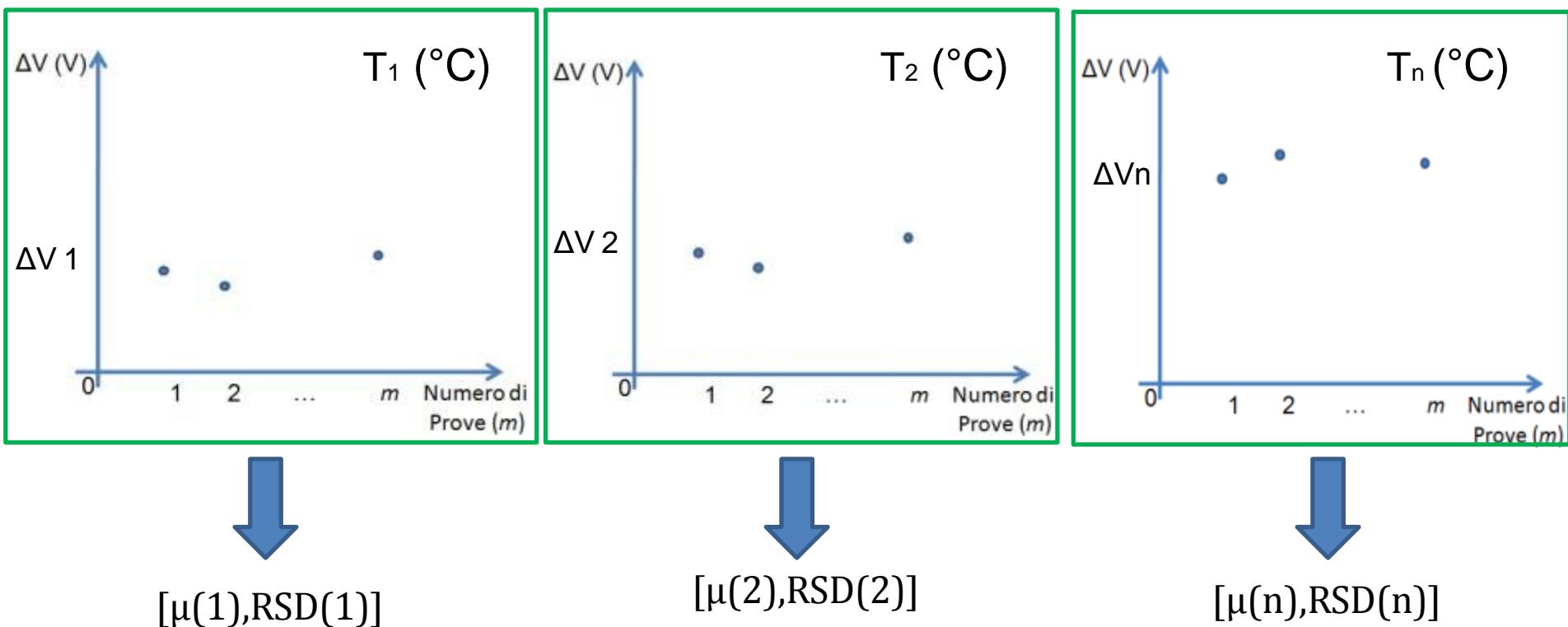
- le variazioni di tensione calcolate precedentemente rispetto ai valori di temperatura;
- l'interpolazione dei dati;
- calcolare il coefficiente angolare della retta di interpolazione



# Ripetibilità

Per ogni temperatura T (T<sub>1</sub>, T<sub>2</sub>... T<sub>n</sub>) applicata alla prima terminazione della termocoppia, calcolare la variazione di tensione per un numero di prove pari a m

Calcolare media ( $\mu$ ) e deviazione standard relativa percentuale (RSD) per ogni forza peso applicata dove  $RSD = 100 \times \text{dev.std.} / \text{media}$

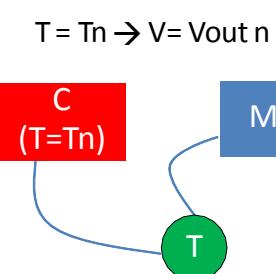
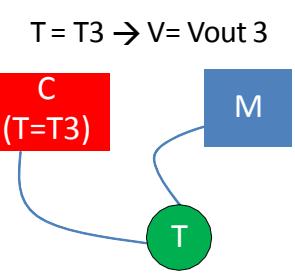
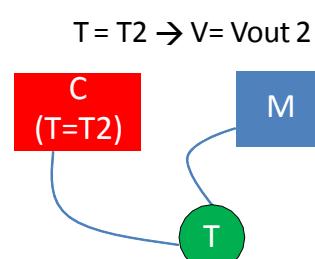
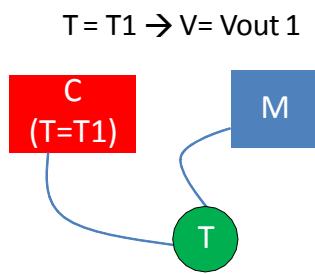


Calcolare la deviazione standard relativa percentuale (RSD) totale come:  $(1/n) \times \sum_{i=1:n} (\text{RSD}_i)$

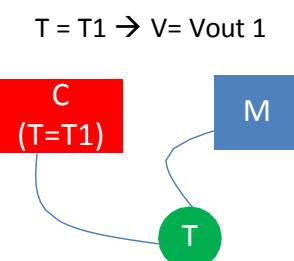
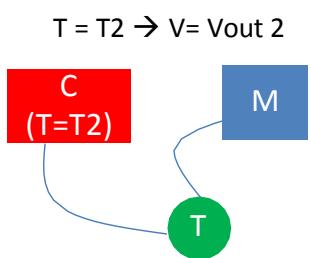
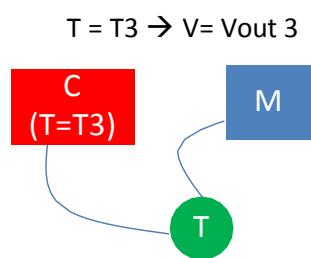
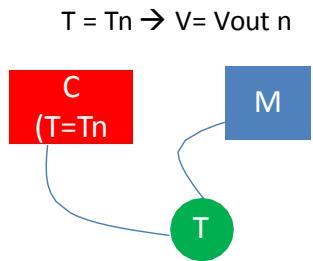
# Isteresi

Si parla di isteresi quando la caratteristica del trasduttore è differente nei casi in cui il segnale sia in crescita e quello in cui il segnale sia in diminuzione.

Aumento della  
temperatura  
(Salita)

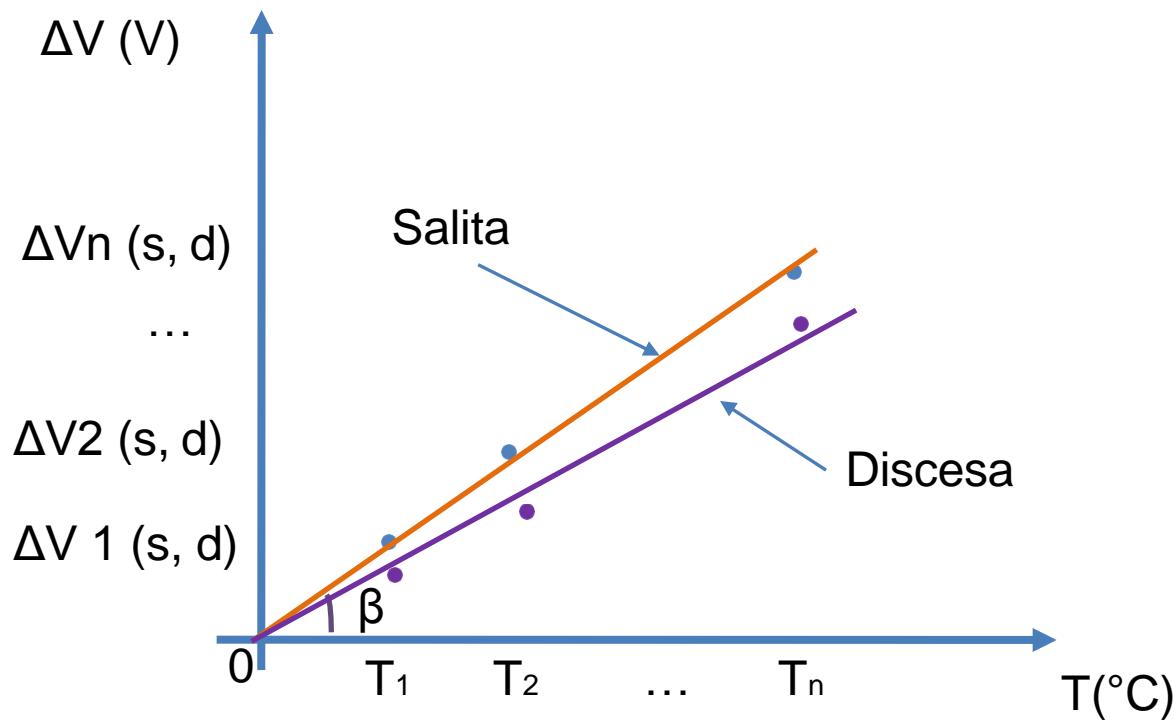


Diminuzione  
della temperatura  
(Discesa)



# Isteresi

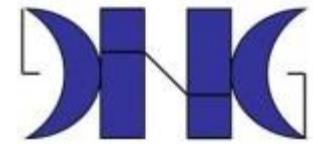
Si parla di isteresi quando la caratteristica del trasduttore è differente nei casi in cui il segnale sia in crescita e quello in cui il segnale sia in diminuzione.



Errore di isteresi in %

$$\frac{\Delta Vn(s) - \Delta Vn(d)}{Media(\Delta Vn(s), \Delta Vn(d))} * 100$$

# Laurea in Ingegneria Elettronica per l'Automazione e le Telecomunicazioni



## Corso di Elettronica per l'Automazione (9CFU)

### Caso Studio: Fotodiodo

**Docente del corso:**  
Prof. Andrea Cusano

**Assistenti di laboratorio:**  
Ing. Giuseppe Quero  
Ing. Federica Gambino

**Università degli Studi del Sannio**  
**Dipartimento di Ingegneria**  
**Gruppo di Optoelettronica**  
**Benevento**

# Sommario

## □ Introduzione

- Obiettivo dell'esercitazione
- Modalità di svolgimento

## □ Fotodiodo

- Principio di funzionamento
- DET 410

## □ Setup per la caratterizzazione

- DET 410
- Resistore variabile
- Sorgente laser
- Multimetro

## □ Fase operativa

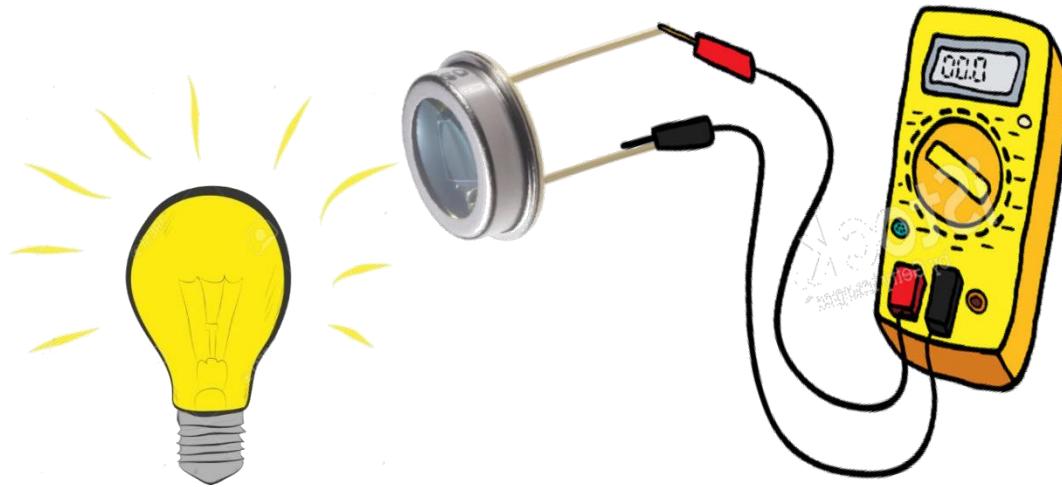
- Installazione sensore e connessioni varie
- Procedura di caratterizzazione ed elaborazione dati

# Obiettivo

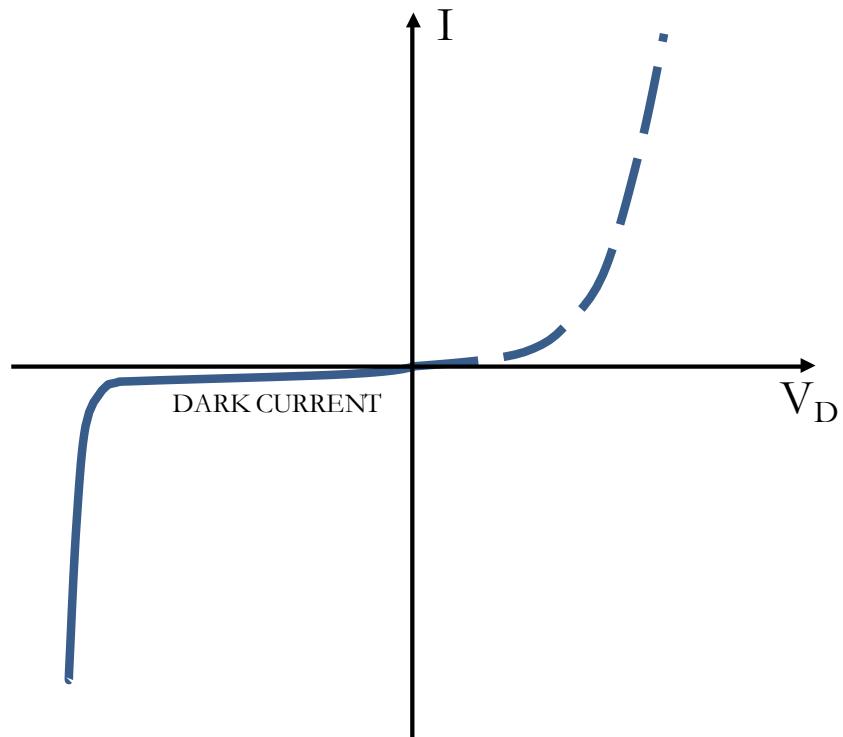
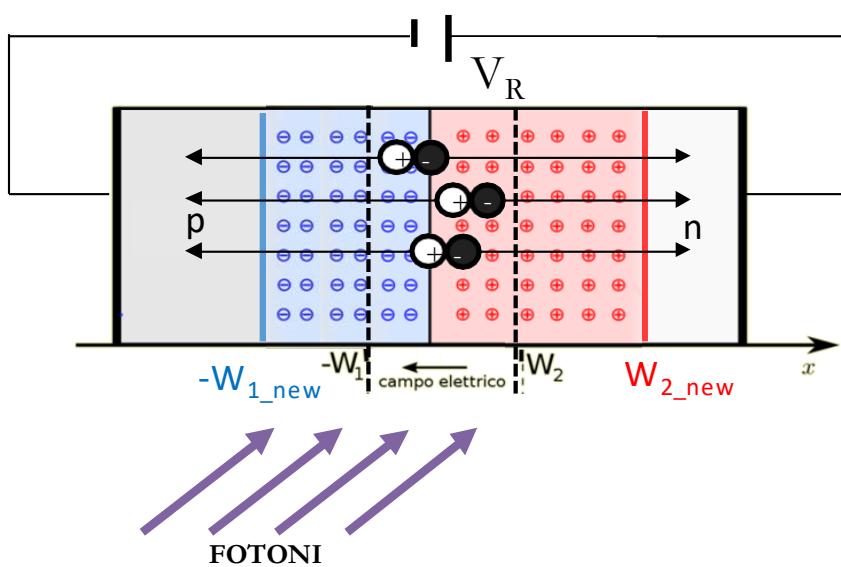
- **Caratterizzazione sperimentale di un fotodiodo**
- **Verifica delle prestazioni in termini di:**
  - Sensibilità
  - Ripetibilità
  - Isteresi
- **Modalità di svolgimento**
  - Descrizione del setup sperimentale
  - Descrizione del sensore
  - Procedura operativa di caratterizzazione
  - Elaborazione dati

# Definizione

Il fotodiodo è un sensore in grado di convertire un segnale ottico in un segnale elettrico



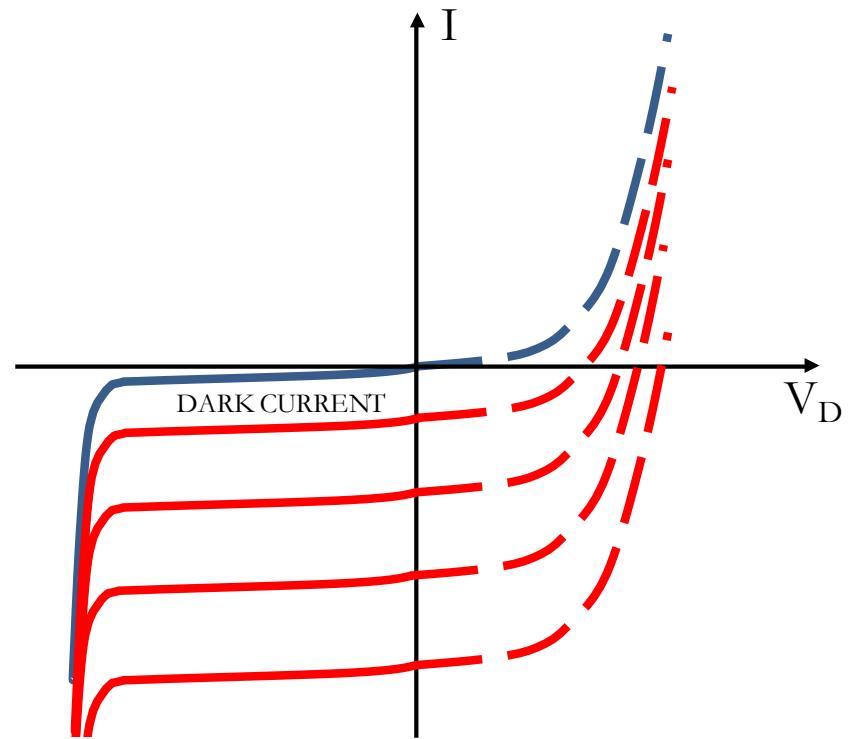
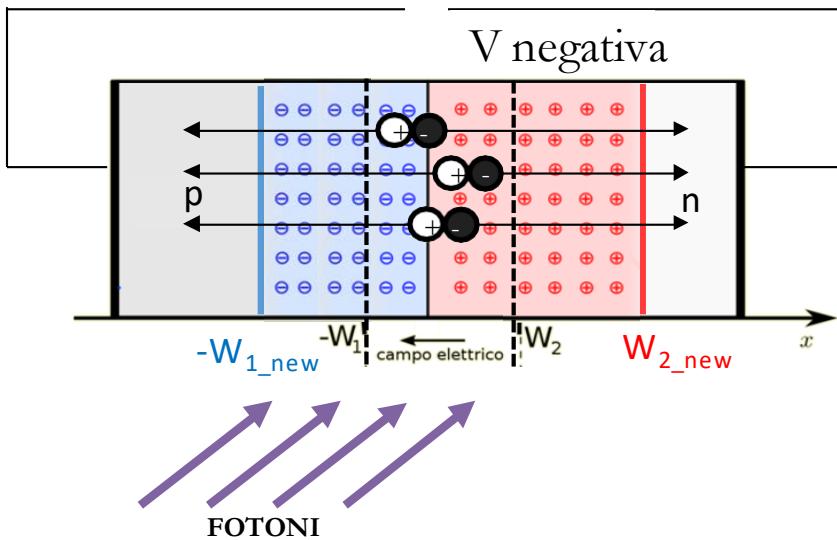
# Principio di funzionamento del fotodiodo



Nel momento in cui un fotone viene a contatto con la superficie della giunzione, cederà ad essa la sua energia dando vita a nuove coppie elettrone-lacuna.

Queste coppie a causa del campo elettrico alla giunzione cominceranno a muoversi dando luogo alla corrente fotogenerata che va dal lato n al lato p

# Principio di funzionamento

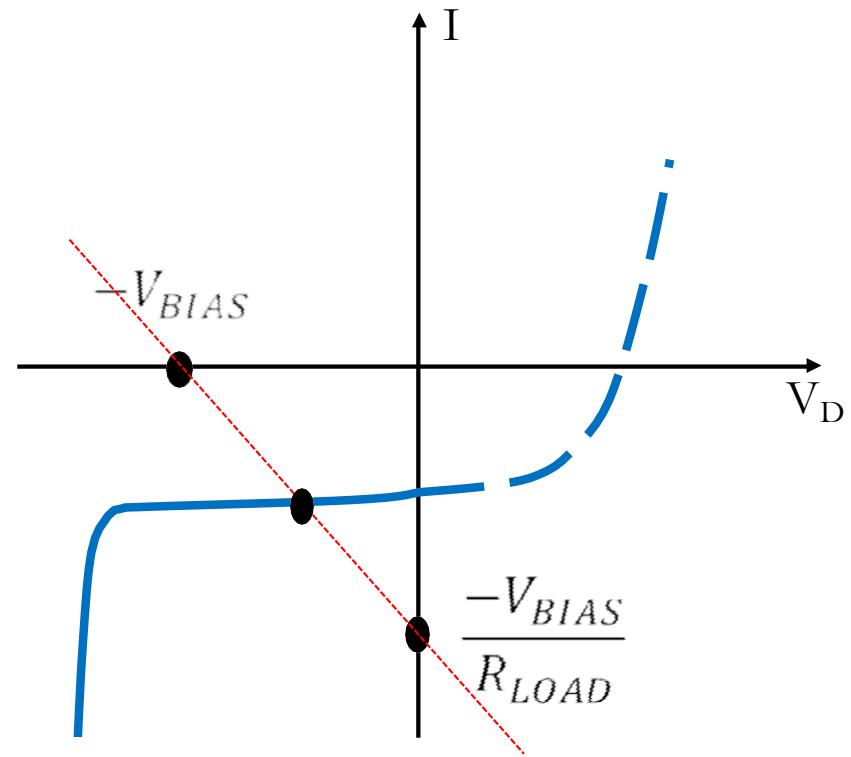
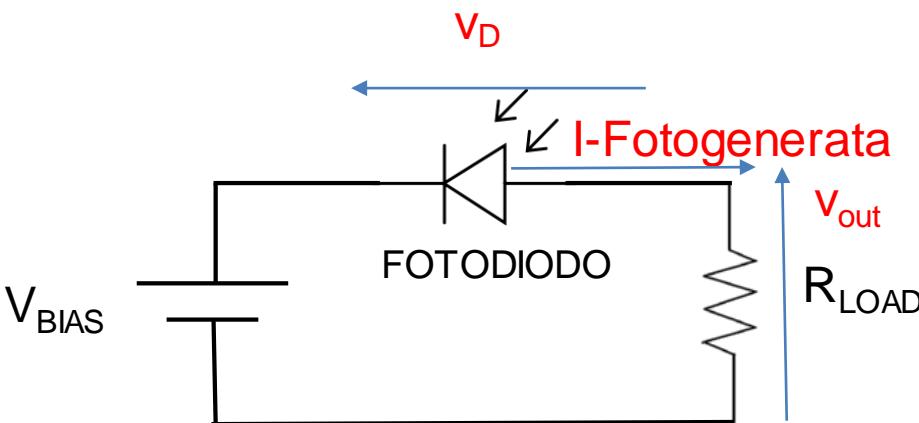


La generazione di una corrente foto generata modifica la caratteristica del diodo

# Schema di principio

Il diodo è polarizzato inversamente, quindi la sua corrente in assenza di radiazione luminosa è praticamente nulla

Quando la radiazione luminosa arriva sul fotodiodo, viene generata una corrente fotoindotta che circolerà nel diodo e fluirà nella resistenza generando una variazione della tensione di uscita



$$V_{BIAS} = V_D + RI_{FG}$$

$$V_{out} = R I_{FG}$$

# Il nostro sensore



435 Route 206 • P.O. Box 366  
Newton, NJ 07860-0366

[www.thorlabs.com](http://www.thorlabs.com)

PH. 973-579-7227  
FAX 973-300-3600  
[technicalsupport@thorlabs.com](mailto:technicalsupport@thorlabs.com)

## DET410 - HIGH-SPEED InGaAs DETECTOR

### **DESCRIPTION:**

Thorlabs' DET410 is a ready-to-use high-speed photo detector. The unit comes complete with a photodiode and internal 12V bias battery enclosed in a ruggedized aluminum housing. The head includes a removable 1" optical coupler (SM1T1), providing easy mounting of ND filters, spectral filters and other Thorlabs 1" stackable lens mount accessories. Also available are fiber adapters (SMA, FC and ST style). An #8-32 tapped hole is provided on the base of the housing to mount the detector directly to a Thorlabs' positioning device (1/2" post holder, mounting plates, etc.).

### **SPECIFICATIONS:**

<b>Detector:</b>	InGaAs PIN	<b>Housing:</b>	Black Anodized Aluminum
<b>Spectral Response:</b>	700-1800nm	<b>Size:</b>	Φ1.43" x 1.67"
<b>Peak Wavelength:</b>	1500nm +/- 50nm	<b>Output:</b>	BNC, DC-Coupled
<b>Rise/Fall Time<sup>1</sup>:</b>	5ns	<b>Bias:</b>	12V Battery (Type A23)
<b>Diode Capacitance:</b>	22pF	<b>Mounting:</b>	8-32 (M4) Tapped Hole
<b>NEP:</b>	$1 \times 10^{-14} \text{W}/\sqrt{\text{Hz}}$	<b>Diode Socket:</b>	TO-5, Anode Marked
<b>Dark Current:</b>	25nA	<b>Damage Threshold:</b>	100mW CW 0.5 J/cm <sup>2</sup> (10ns pulse)
<b>Active Area:</b>	Φ1mm (0.8mm <sup>2</sup> )	<b>Operating Temp:</b>	0 to 85°C
<b>Linearity Limit:</b>	1mW	<b>Storage Temp:</b>	-40 to 125°C
<b>Max Reverse Current:</b>	10mA		

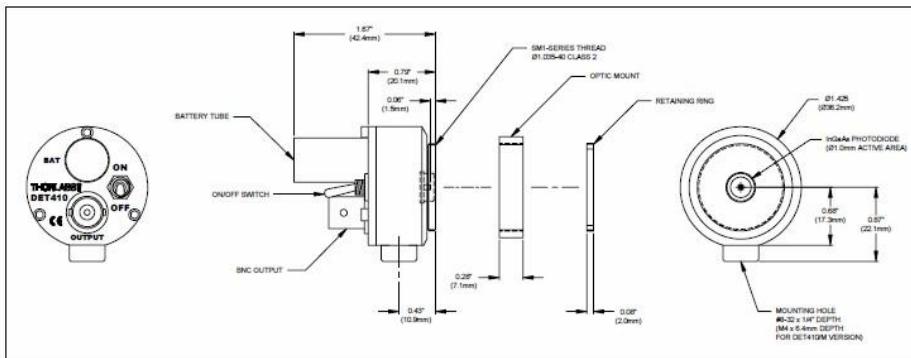
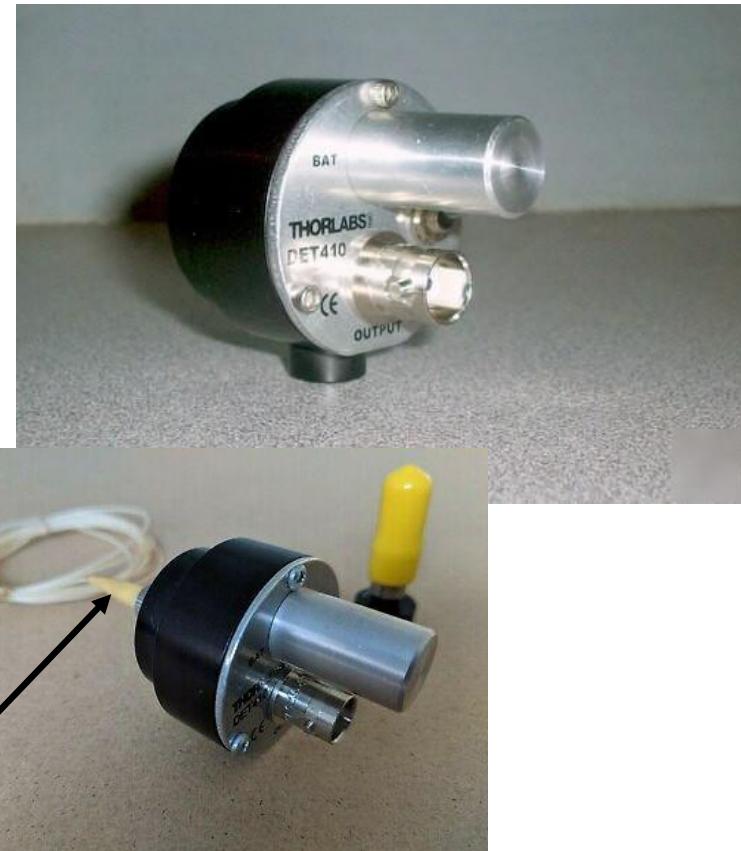


Figure 1. - Mechanical Dimensions

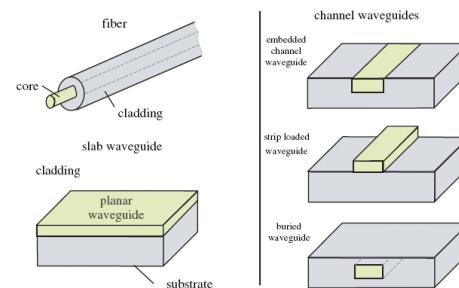
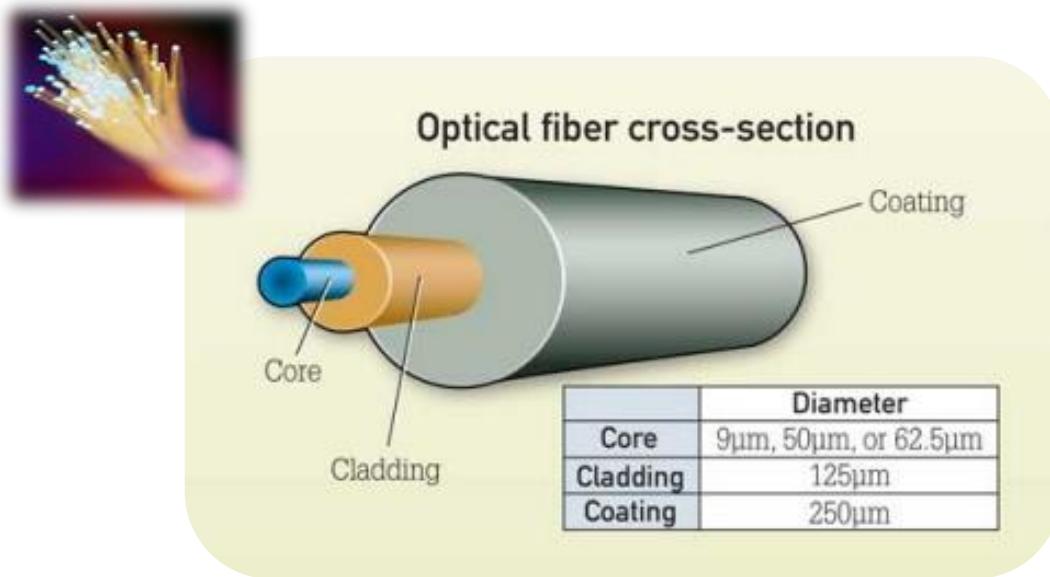


Ingresso in **fibra ottica**

# Fibra ottica

Le fibre ottiche sono delle guide d'onda di vetro formate da sottili fili di vetro ( $\text{SiO}_2$ )

La loro struttura è composta da due cilindri coassiali caratterizzati da diversi indici di rifrazione.



**Cladding (pure silica)**

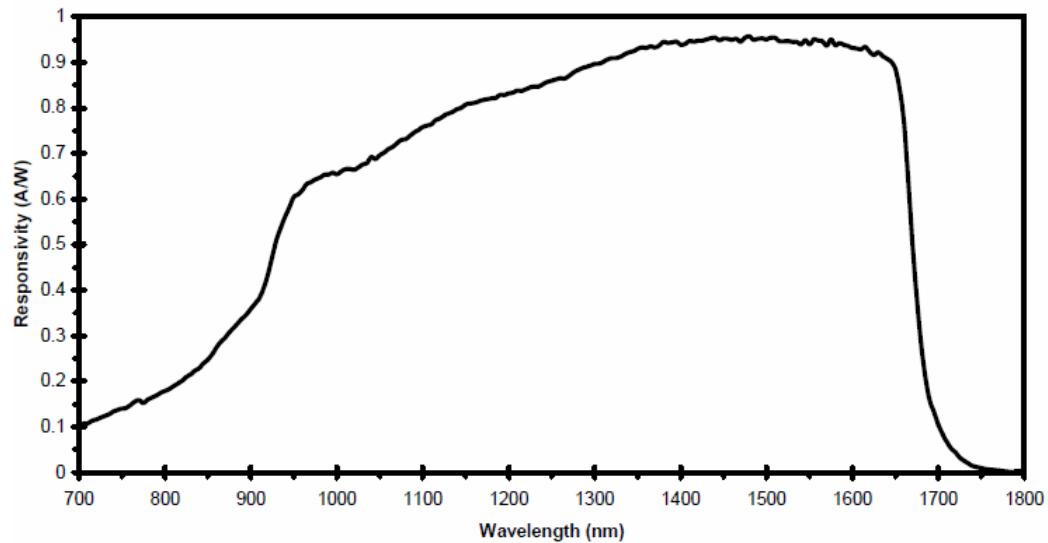
**Core (Germanium doped silica)**

Il cilindro interno è detto nucleo o core della fibra, mentre quello esterno è chiamato cladding. Nella struttura sono inoltre previsti rivestimenti (coating) aggiuntivi di materiale sintetico che hanno lo scopo di proteggere la guida d'onda, bloccando eventuali microfratture, e renderla più maneggevole e resistente.

# Il nostro sensore



Figure 2 - DET410 Spectral Responsivity Curve



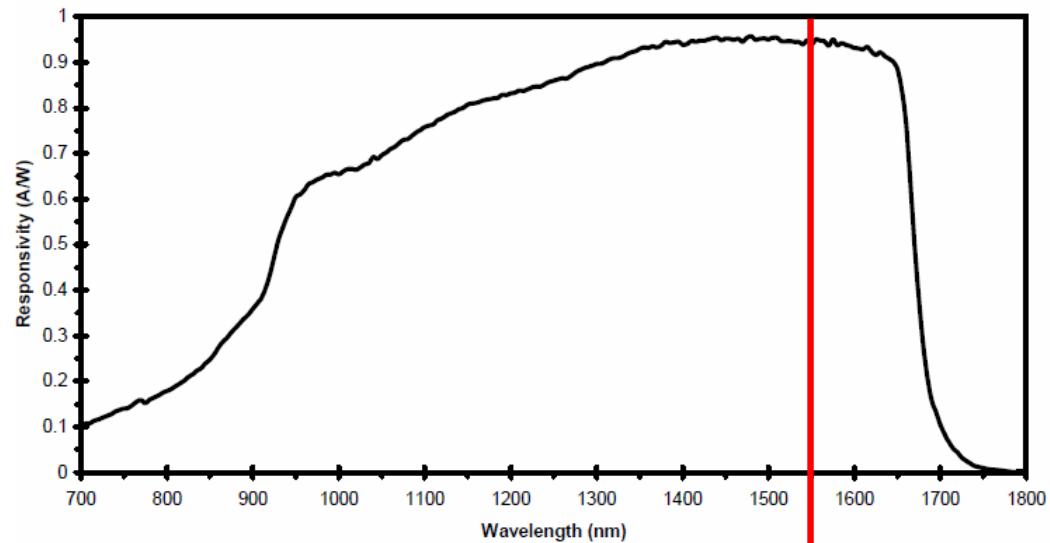
$$V_{out} = R_{LOAD} * I_{ph} = R_{LOAD} * Resp(\lambda) * P_{inc}$$

$$I_{ph} = Resp(\lambda) * P_{inc}$$

# Il nostro sensore

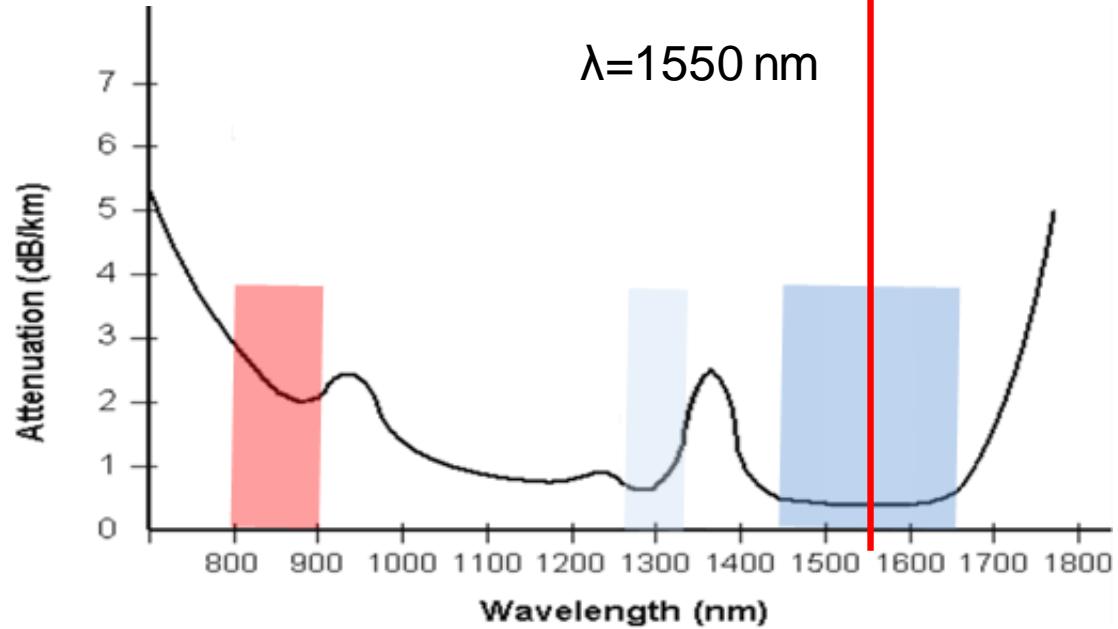


Figure 2 - DET410 Spectral Responsivity Curve



## Attenuazione in fibra ottica

Nominal spectrum (nm)	Fiber Types		
	Window	Multimode (MM)	Single Mode (SM)
850 ± 30 (short wavelength)	I	X	
1300 ± 30 (long wavelength)	II	X	X
1550 ± 30 (extra-long wavelength)	III		X



# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

- Fotodiodo (FD)

$$V_{out} = R_{LOAD} * \text{Resp}(\lambda) * P_{inc}$$



Collegamento strumentazione



FD

# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

- Fotodiodo (FD)

$$V_{out} = R_{LOAD} * Resp(\lambda) * P_{inc}$$

- Sorgente Laser (SL)



Collegamento strumentazione



SL

Cavo in fibra ottica



FD

# Setup sperimentale

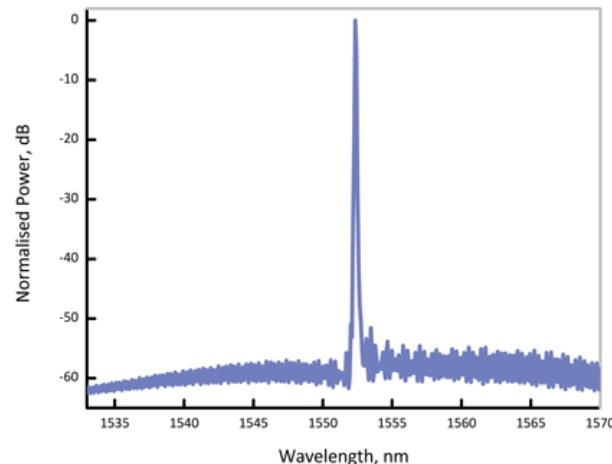
Il setup sperimentale è costituito dai seguenti dispositivi:

- Fotodiodo (FD)
- *Sorgente Laser (SL)*

**Tunable Laser Source  
AQ4321A**



Tipico spettro della sorgente laser a banda stretta



# Setup sperimentale

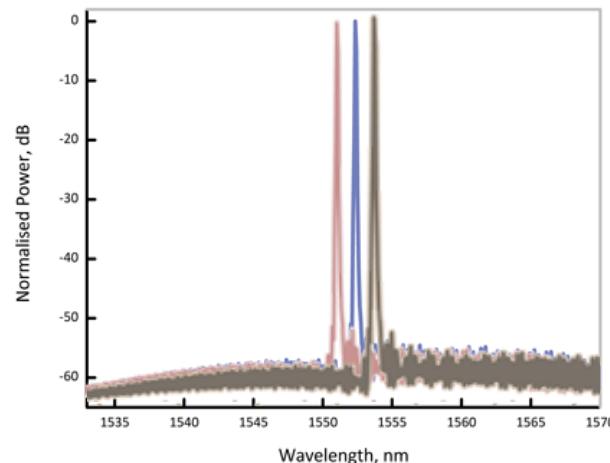
Il setup sperimentale è costituito dai seguenti dispositivi:

- Fotodiodo (FD)
- *Sorgente Laser (SL)*



Tipico spettro della sorgente laser a banda stretta

Variando la lunghezza d'onda  
di operazione



# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

- Fotodiodo (FD)

$$V_{out} = R_{LOAD} * Resp(\lambda) * P_{inc}$$

- Sorgente Laser (SL)

- **Resistore variabile (Rs)**



Collegamento strumentazione



SL

Cavo in fibra ottica



Resistore  
variabile

FD

Rs

# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

➤ Fotodiodo (FD)

$$V_{out} = \textcircled{R_{LOAD}} * Resp(\lambda) * P_{inc}$$

➤ Sorgente Laser (SL)

➤ Resistore variabile (Rs)



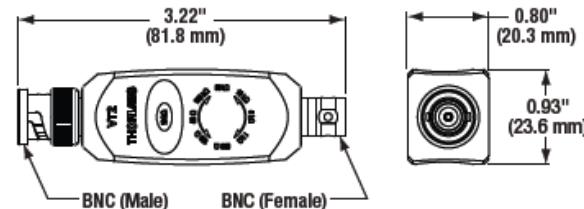
Electrical Specifications				
Nominal Value	Max Power Rating	Max DC Voltage	Value with 1 MΩ in Parallel	Value with 10 MΩ in Parallel
50 Ω	1 W	5 V	50 Ω	50 Ω
100 Ω	1 W	8 V	100 Ω	100 Ω
500 Ω	0.25 W	8 V	500 Ω	500 Ω
1 kΩ	0.25 W	12 V	1 kΩ	1 kΩ
5 kΩ	0.25 W	25 V	4.98 kΩ	4.99 kΩ
10 kΩ	0.125 W	25 V	9.90 kΩ	9.99 kΩ
50 kΩ	0.125 W	25 V	47.72 kΩ	49.75 kΩ
100 kΩ	0.125 W	25 V	90.91 kΩ	99.00 kΩ

General Specifications	
Resistance Tolerance	±5%
Storage Temperature	-30 to 60 °C
Operating Temperature	10 to 40 °C



VT2

## Drawing



# Setup sperimentale

Il setup sperimentale è costituito dai seguenti dispositivi:

➤ Fotodiodo (FD)

$$V_{out} = R_{LOAD} * Resp(\lambda) * P_{inc}$$

➤ Sorgente Laser (SL)



➤ Resistore variabile (Rs)



➤ **Multimetro (M)**



Collegamento strumentazione



SL

Cavo in fibra ottica



M

Resistore  
variabile

FD

Rs

Cavo elettrico

# Caratterizzazione

Caratterizzazione Potenza (W) – Tensione (V)



Sensibilità

Operazione preliminare: collegare tra di loro i dispositivi in modo opportuno ed accendere la strumentazione (controllore di temperatura e multimetro)

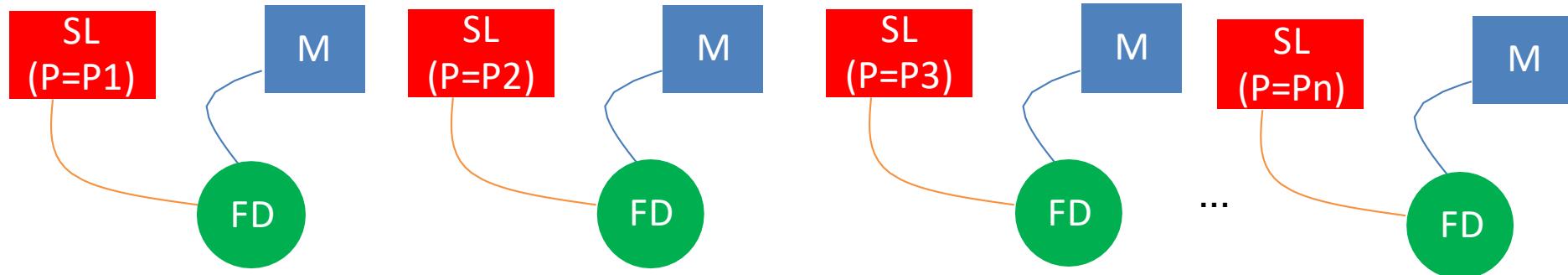
1. leggere il valore in tensione sul multimetro quando la potenza del laser viene fissata a P1
2. leggere il valore in tensione sul multimetro quando la potenza del laser viene fissata a P2>P1
3. leggere il valore in tensione sul multimetro quando la potenza del laser viene fissata a P3>P2
- ...
4. leggere il valore in tensione sul multimetro quando la potenza del laser viene fissata a Pn>Pn -1

$$P = P_1 \rightarrow V = V_{out\ 1}$$

$$P = P_2 \rightarrow V = V_{out\ 2}$$

$$P = P_3 \rightarrow V = V_{out\ 3}$$

$$P = P_n \rightarrow V = V_{out\ n}$$



Calcolare le variazioni di tensione

$$\Delta V_1 = V_{out\ 2} - V_{out\ 1}$$

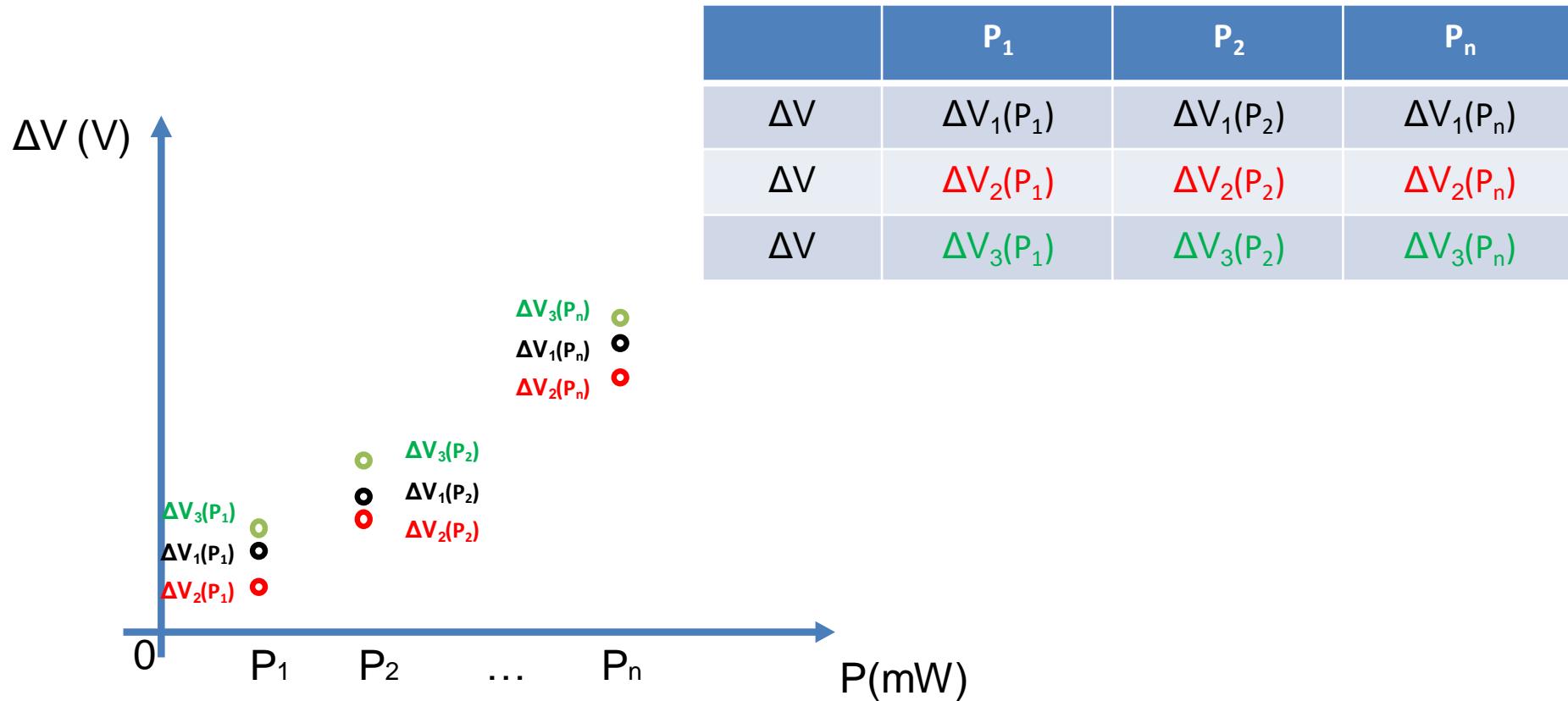
$$\Delta V_3 = V_{out\ 3} - V_{out\ 1}$$

$$\Delta V_2 = V_{out\ 3} - V_{out\ 1}$$

$$\Delta V_n = V_{out\ n} - V_{out\ 1}$$

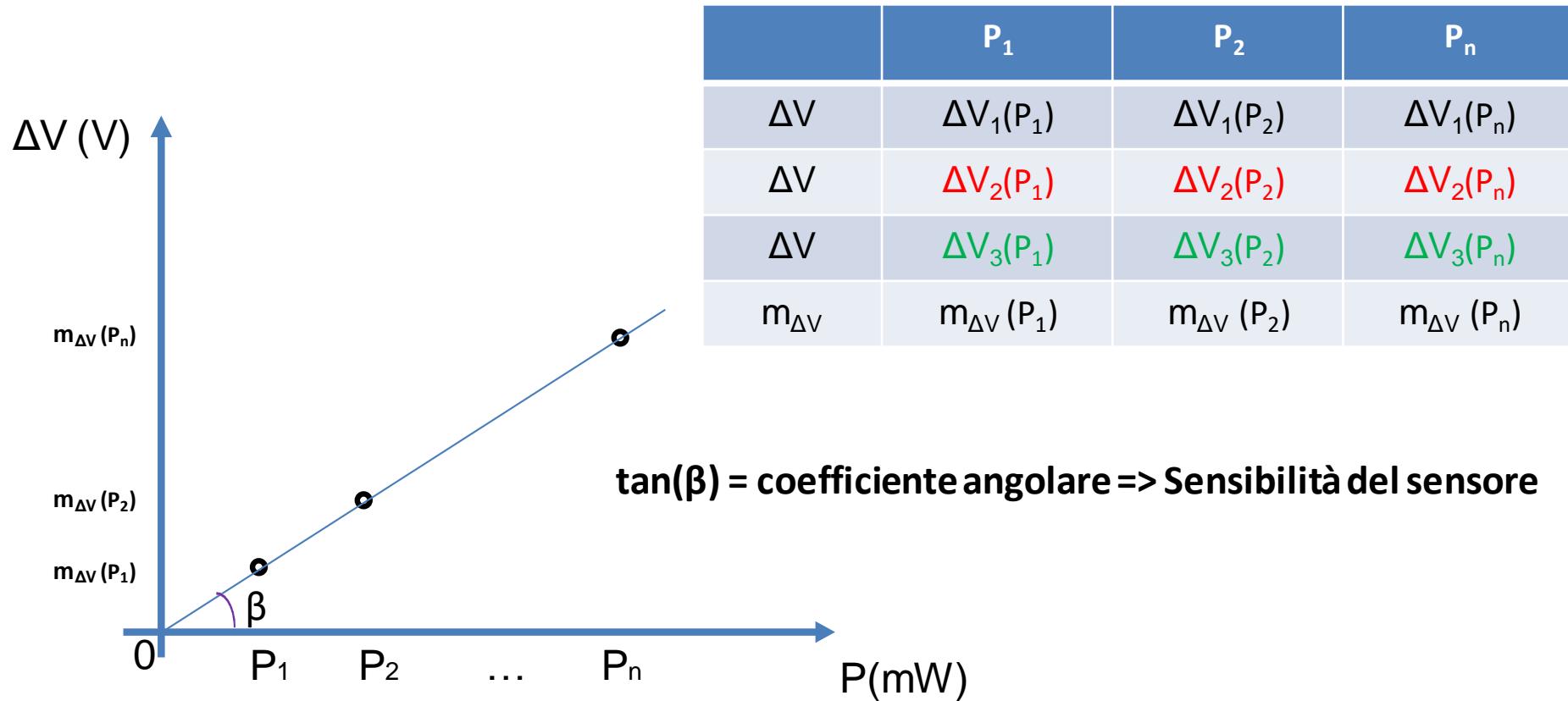
# Sensibilità

Tabellare e graficare le variazioni di tensione ( $\Delta V$ ) per ogni valore della potenza ottica applicata (P)



# Sensibilità

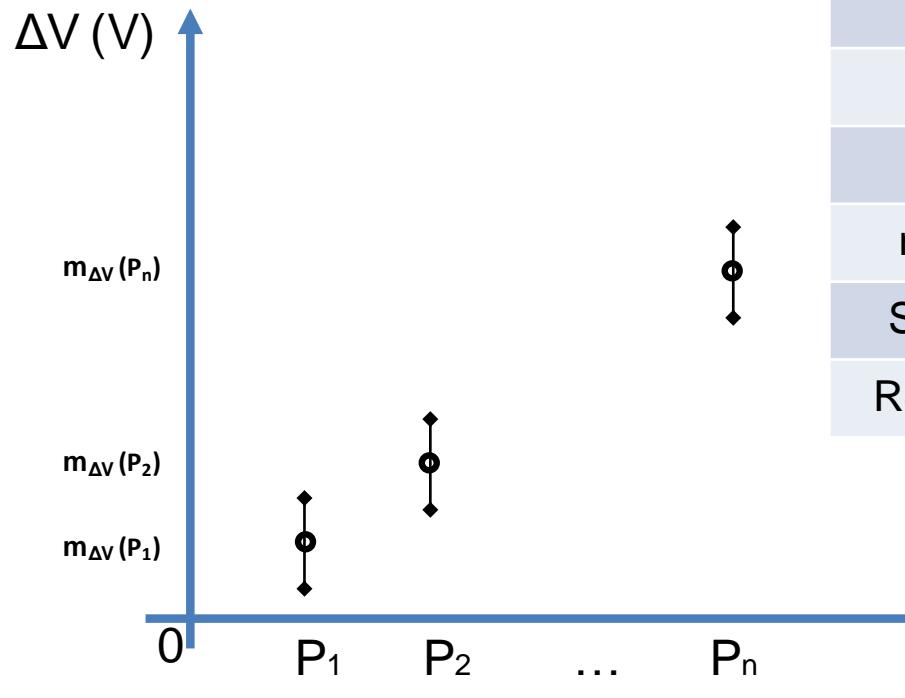
Tabellare e graficare le variazioni di tensione ( $\Delta V$ ) per ogni valore della potenza ottica applicata (P)



# Ripetibilità

Calcolare la deviazione standard relativa percentuale (RSD) per ogni valore della potenza ottica applicata (P) dove  $RSD_{\Delta V} = 100 * SD_{\Delta V} / m_{\Delta V}$

$SD_{\Delta V}$  = deviazione standard



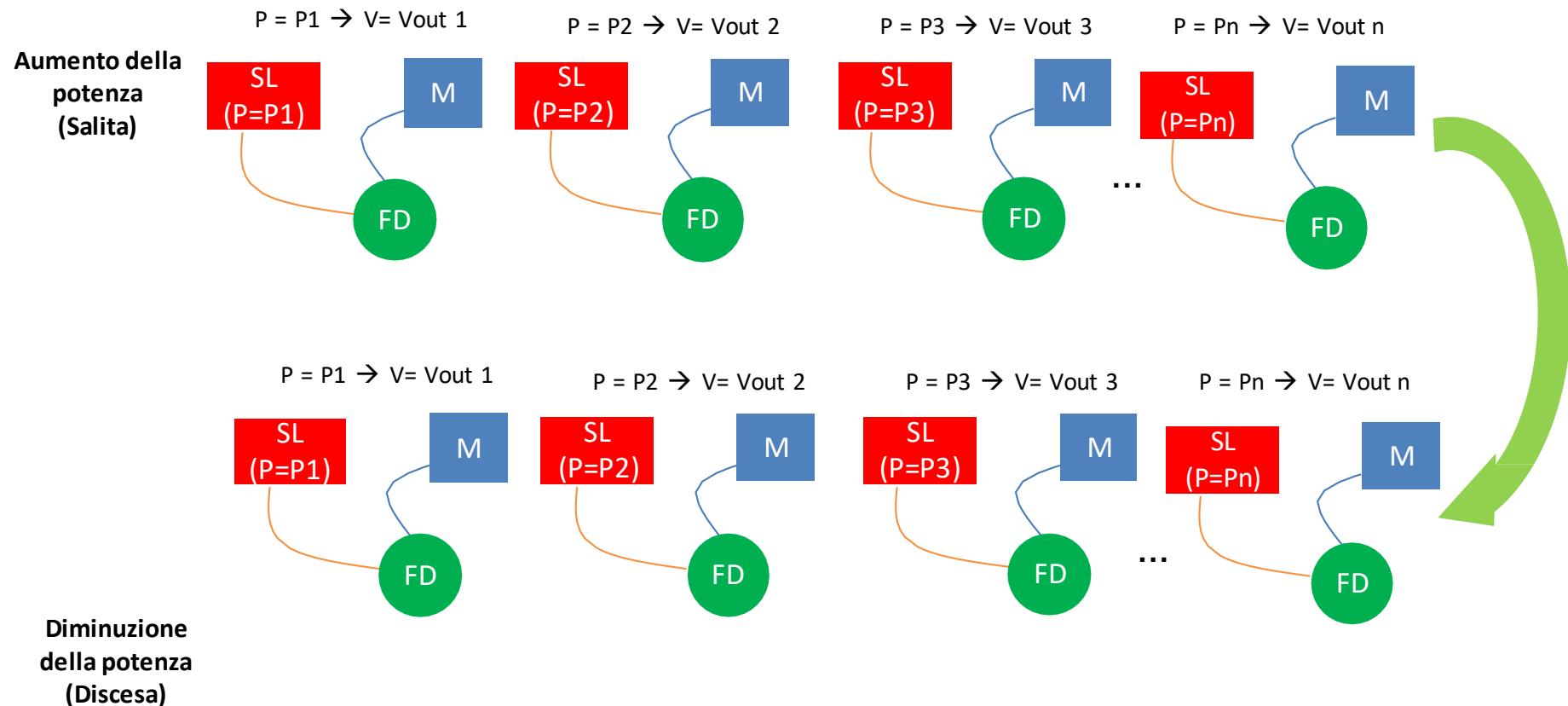
	$P_1$	$P_2$	$P_n$
$\Delta V$	$\Delta V_1(P_1)$	$\Delta V_1(P_2)$	$\Delta V_1(P_n)$
$\Delta V$	$\Delta V_2(P_1)$	$\Delta V_2(P_2)$	$\Delta V_2(P_n)$
$\Delta V$	$\Delta V_3(P_1)$	$\Delta V_3(P_2)$	$\Delta V_3(P_n)$
$m_{\Delta V}$	$m_{\Delta V}(P_1)$	$m_{\Delta V}(P_2)$	$m_{\Delta V}(P_n)$
$SD_{\Delta V}$	$SD_{\Delta V}(P_1)$	$SD_{\Delta V}(P_2)$	$SD_{\Delta V}(P_n)$
$RSD_{\Delta V}$	$RSD_{\Delta V}(P_1)$	$RSD_{\Delta V}(P_2)$	$RSD_{\Delta V}(P_n)$

Calcolare la deviazione standard relativa percentuale ( $RDS_{tot}$ ) totale come:

$$RDS_{tot} = \frac{\text{Somma}(RDS_{\Delta V}(P_1), RDS_{\Delta V}(P_2), \dots, RDS_{\Delta V}(P_n))}{n}$$

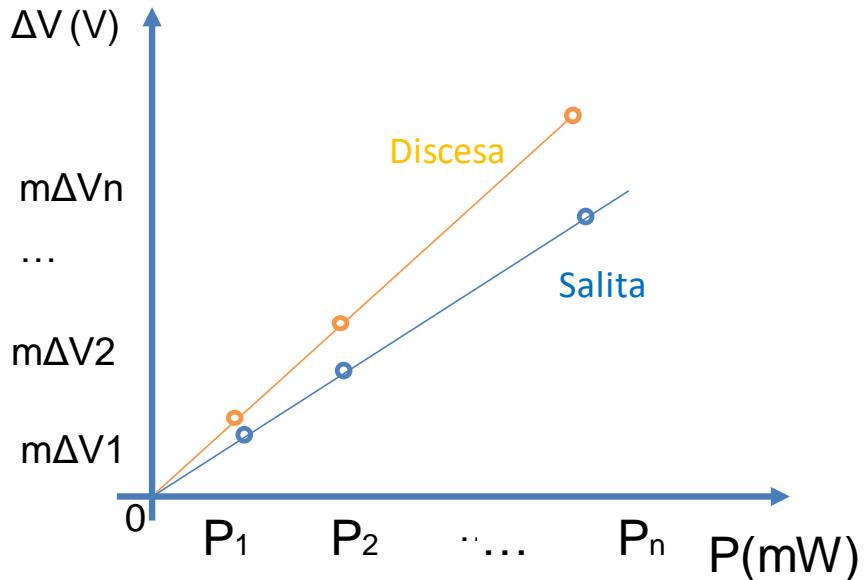
# Isteresi

Si parla di isteresi quando la caratteristica del sensore è differente nei casi in cui il segnale sia in crescita e quello in cui il segnale sia in diminuzione.



# Isteresi

Tabellare e graficare le variazioni di tensione ( $\Delta V$ ) per ogni valore della potenza ottica applicata (P)



	$P_1$	$P_2$	$P_n$
$\Delta V_s$	$\Delta V_{s1}$	$\Delta V_{s1}$	$\Delta V_{s1}$
$\Delta V_s$	$\Delta V_{s2}$	$\Delta V_{s2}$	$\Delta V_{s2}$
$\Delta V_s$	$\Delta V_{s3}$	$\Delta V_{s3}$	$\Delta V_{s3}$
$m_{\Delta V_s}$	$m_{\Delta V_s}(P_1)$	$m_{\Delta V_s}(P_2)$	$m_{\Delta V_s}(P_n)$
$\Delta V_d$	$\Delta V_{d1}$	$\Delta V_{d1}$	$\Delta V_{d1}$
$\Delta V_d$	$\Delta V_{d2}$	$\Delta V_{d2}$	$\Delta V_{d2}$
$\Delta V_d$	$\Delta V_{d3}$	$\Delta V_{d3}$	$\Delta V_{d3}$
$m_{\Delta V_d}$	$m_{\Delta V_d}(P_1)$	$m_{\Delta V_d}(P_2)$	$m_{\Delta V_d}(P_n)$
$IS_{\Delta V}(\%)$	$IS_{\Delta V}(P_1)$	$IS_{\Delta V}(P_2)$	$IS_{\Delta V}(P_n)$

Errore di isteresi percentuale per ogni valore di grandezza applicata

$$IS_{\Delta V}(\%) = \frac{|m_{\Delta V_s} - m_{\Delta V_d}|}{Media(m_{\Delta V_s}, m_{\Delta V_d})} * 100$$

Errore di isteresi percentuale totale

$$IS_{tot\Delta V}(\%) = Media (IS_{\Delta V}(P_1), IS_{\Delta V}(P_2), \dots, IS_{\Delta V}(P_n)) * 100$$

# Laboratorio di Elettronica per l'Automazione

Prof. Andrea Cusano

*Caratterizzazione di sensori per la misura  
dell'umidità relativa*

---

Ing. Marco Leone - Ing. Sofia Principe

# SOMMARIO

## □ Introduzione

- Obiettivo dell'esercitazione
- Modalità di svolgimento

## □ L'umidità atmosferica

- Descrizione e tipologie

## □ La misura dell'umidità relativa

- Sensori commerciali
- HIH 4000

## □ Setup per la caratterizzazione

- HIH 4000
- Alimentatore
- Multimetro
- Camera climatica
- Breadboard

## □ Fase operativa

- Installazione sensore su breadboard e connessioni varie
- Procedura di taratura ed elaborazione dati

# OBIETTIVO

- Caratterizzazione e calibrazione sperimentale di un sensore di umidità relativa**
- Verifica delle prestazioni in termini di:**
  - Sensibilità
  - Ripetibilità
  - Isteresi
- Modalità di svolgimento**
  - Descrizione del setup sperimentale
  - Descrizione del sensore
  - Procedura operativa di taratura
  - Elaborazione dati

# L'UMIDITÀ RELATIVA

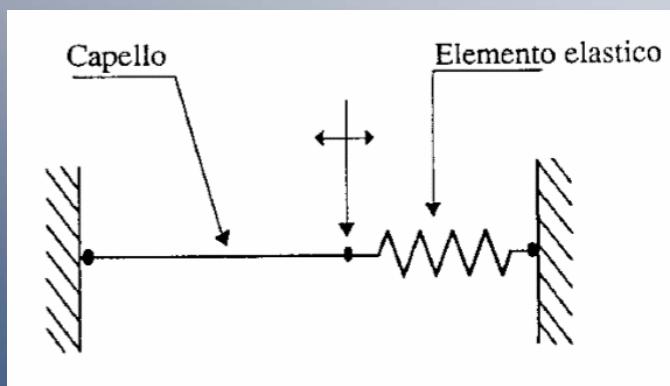
- PER **UMIDITÀ ATMOSFERICA** S'INTENDE LA QUANTITÀ DI VAPORE ACQUEO CONTENUTO NELL'ARIA E CHE PROVIENE DALL'EVAPORAZIONE DELL'ACQUA DALLA SUPERFICIE TERRESTRE PER AZIONE DELLA RADIAZIONE SOLARE E ANCHE DALLA TRASPIRAZIONE DELLE PIANTE.
- **UMIDITÀ ASSOLUTA, G/m<sup>3</sup>**
- L'UMIDITÀ ASSOLUTA INDICA QUANTI GRAMMI DI ACQUA SONO PRESENTI IN 1 METRO CUBO DI ARIA.
- **UMIDITÀ DI SATURAZIONE, G/m<sup>3</sup>**
- L'UMIDITÀ DI SATURAZIONE INDICA LA QUANTITÀ MASSIMA DI GRAMMI DI ACQUA CHE 1 METRO CUBO DI ARIA PUÒ CONTENERE AD UNA DATA TEMPERATURA.
- **UMIDITÀ RELATIVA, %RH**
- L'UMIDITÀ RELATIVA INDICA LA PERCENTUALE DELLA QUANTITÀ MASSIMA DI VAPORE ACQUEO CONTENUTA NELL'ARIA E SI ESPRIME:

$$RH \% = \frac{Umidità\ Assoluta}{Umidità\ di\ Saturazione} \times 100$$

L'UMIDITÀ RELATIVA SI RIFERISCE SEMPRE AD UNA TEMPERATURA.

# SENSORI DI RH E PRINCIPIO DI FUNZIONAMENTO

- **L'IGROMETRO A CAPELLO** È UNO DEI METODI PIÙ ANTICHI UTILIZZATI PER MISURARE L'UMIDITÀ. LA LUNGHEZZA DEL CAPELLO VARIA IN FUNZIONE DELL'UMIDITÀ AMBIENTE.
- NEI DISPOSITIVI AD USCITA ELETTRICA, LA VARIAZIONE DIMENSIONALE DEL MATERIALE IGROSCOPICO IN FUNZIONE DEL CONTENUTO DI VAPOR D'ACQUA ASSORBITO VIENE TRASDOTTA NELLA VARIAZIONE DI RESISTENZA ELETTRICA DI UN POTENZIOMETRO O DI UN ESTENSIMETRO.
- IL CAMPO DI MISURA TIPICO DI TALI STRUMENTI VA DAL **20% AL 90%**, PER TEMPERATURE COMPRESE TRA **0 °E 40°C** CON UNA INCERTEZZA DI MISURA RARAMENTE INFERIORE A **±5%**. IL SENSORE, INFATTI, MOSTRA UN COMPORTAMENTO NON LINEARE ED UNA SENSIBILITÀ RIDOTTA AGLI ESTREMI DEL CAMPO DI MISURA.



# SENSORE CAPACITIVO

- GLI IGROMETRI CAPACITIVI PRESENTANO UN FUNZIONAMENTO SIMILE AD UN CONDENSATORE IN CUI **IL DIELETTRICO È UN MATERIALE IGROSCOPICO** (SOLITAMENTE POLIMERICO O CERAMICO). GENERALMENTE UNO DEGLI ELETTRODI È PERMEABILE AL VAPOR D'ACQUA. L'EQUILIBRIO IGROMETRICO CHE SI STABILISCE TRA L'ISOLANTE E L'AMBIENTE, MODIFICA LA PERMITTIVITÀ RELATIVA DEL DIELETTRICO.
- **IL RISULTATO È UNA VARIAZIONE DELLA CAPACITÀ DELL'ELEMENTO SENSIBILE CHE SI TRASFORMA IN UN'INFORMAZIONE RAPPRESENTATIVA DELL'UMIDITÀ RELATIVA DELL'ARIA.**
- LA CURVA CARATTERISTICA DEI SENSORI CAPACITIVI PUÒ ESSERE DESCRIPTTA CON BUONA APPROSSIMAZIONE DA UNA RELAZIONE DEL TIPO:

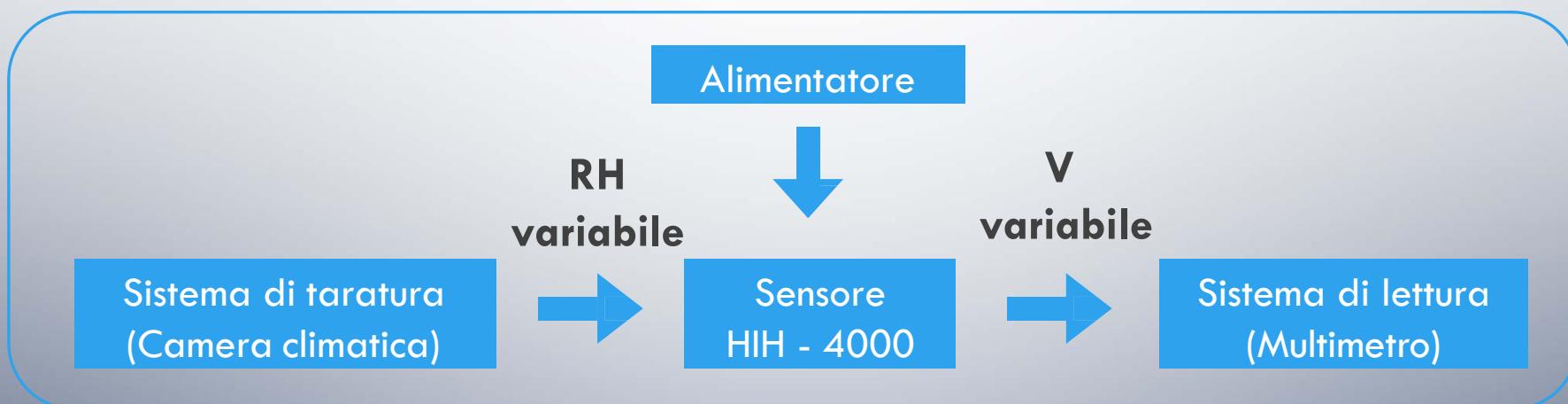
$$C = \left[ v \cdot \left( \epsilon_w^{1/3} - \epsilon_f^{1/3} \right) + \epsilon_f^{1/3} \right]^3 \cdot \epsilon_0 \frac{A}{L}$$

- AVENDO INDICATO CON  $\epsilon_f$ ,  $\epsilon_w$  e  $\epsilon_0$  LE COSTANTI DIELETTRICHE DELL'ACQUA, DEL FILM SENSIBILE E NEL VUOTO, V LA FRAZIONE VOLUMETRICA DELL'ACQUA ABSORBITA DAL SENSORE, A LA SUPERFICIE DEGLI ELETTRODI ED L LO SPESORE DEL DIELETTRICO.
- DA TALE RELAZIONE SI EVINCE CHE ALL'AUMENTARE DEL RAPPORTO TRA LA COSTANTE DIELETTRICA DELL'ACQUA E QUELLA DEL FILM SENSIBILE AUMENTA LA SENSIBILITÀ DEL SENSORE STESSO. INOLTRE LA CARATTERISTICA CAPACITÀ/UMIDITÀ È FUNZIONE DELLA FREQUENZA ECCITANTE.

# FASE Sperimentale

CALIBRAZIONE IN **UMIDITÀ RELATIVA** DI UN SENSORE COMMERCIALE DI TIPO HIH-4000

## Set-up sperimentale

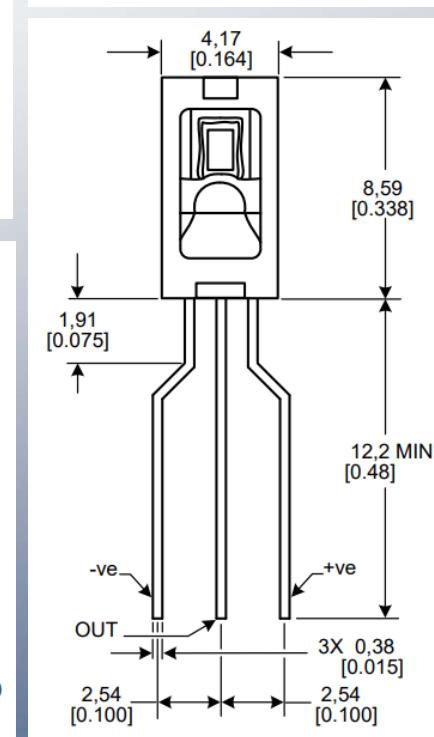
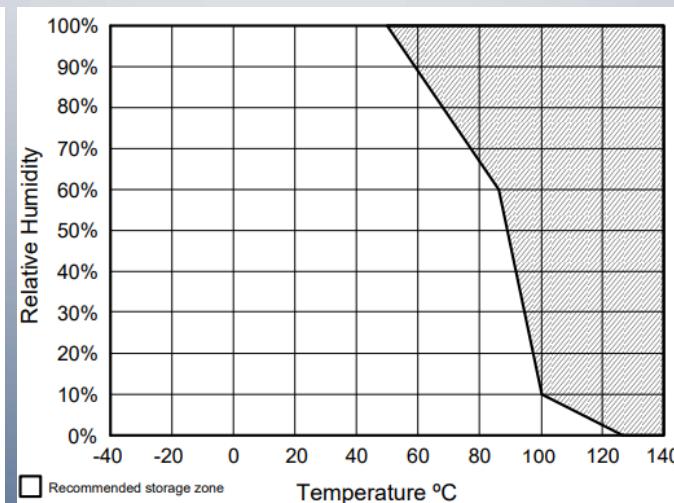
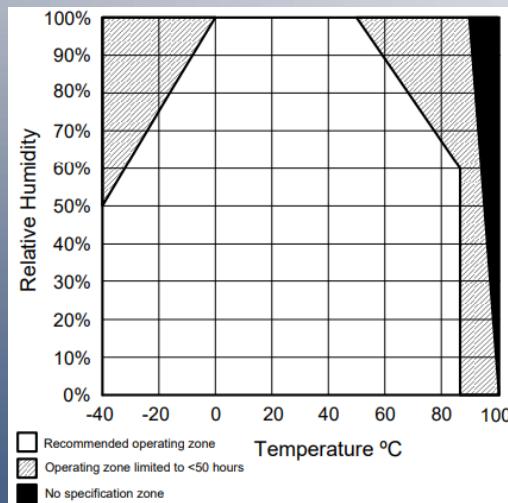


# SENSORE HONEYWELL HIH-4000-1

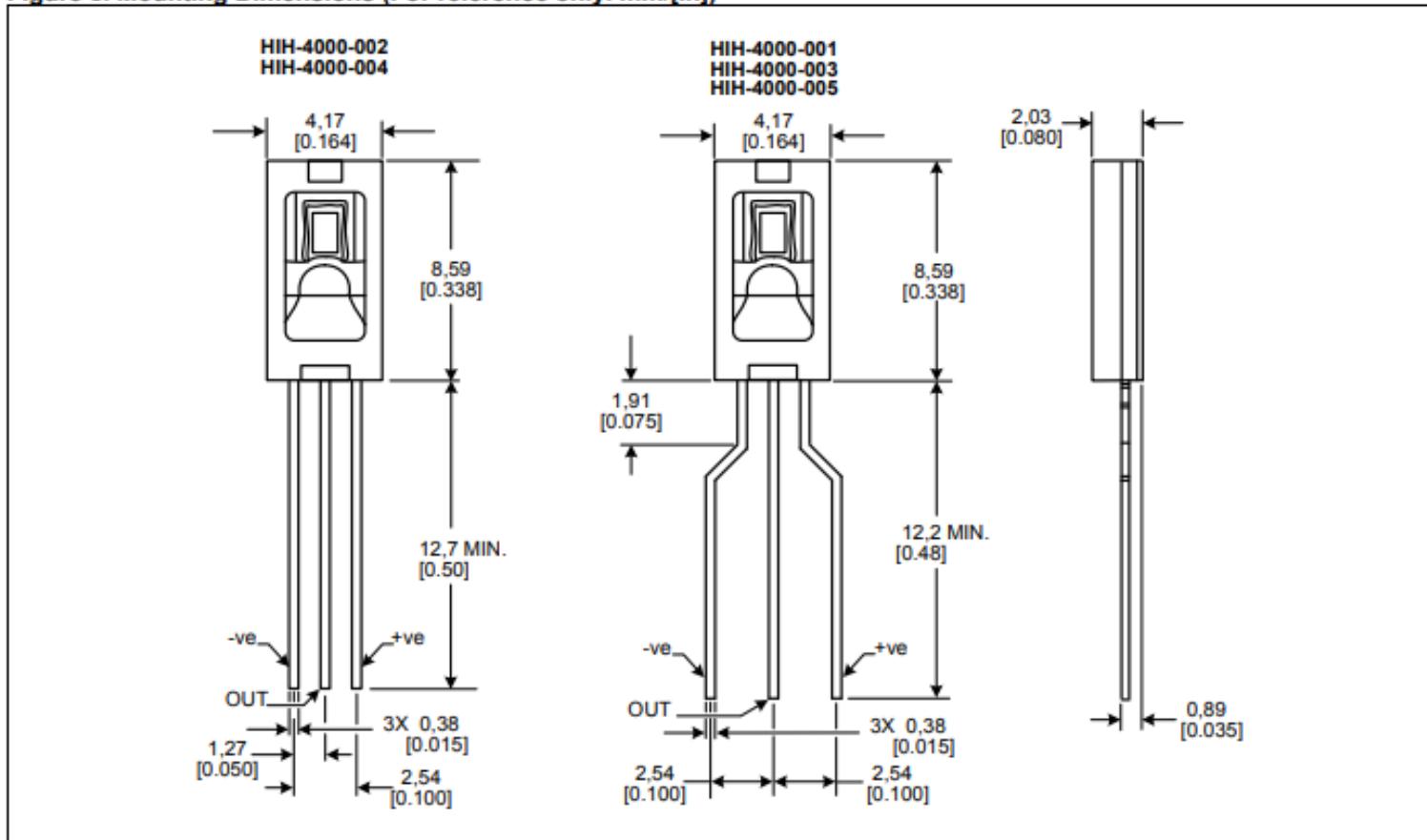
## RELATIVE HUMIDITY SENSOR

Table 2. Performance Specifications (At 5 Vdc supply and 25 °C [77 °F] unless otherwise noted.)

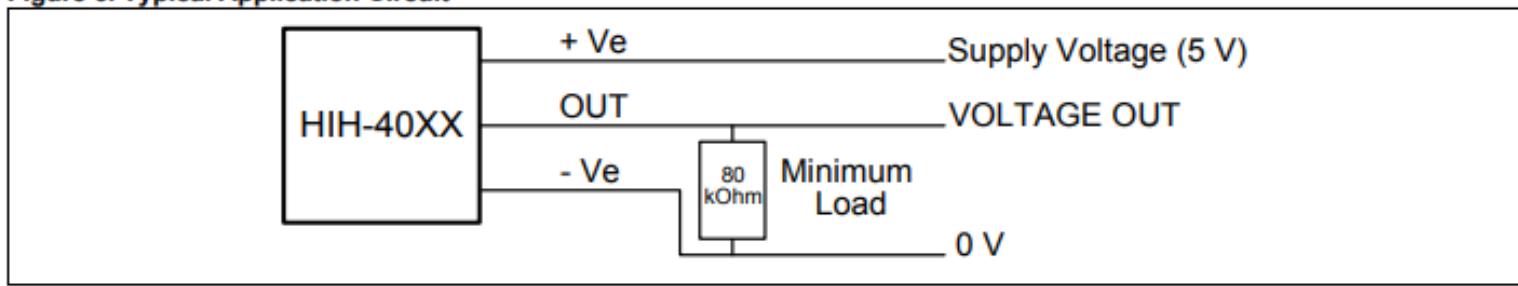
Parameter	Minimum	Typical	Maximum	Unit	Specific Note
Interchangeability (first order curve)	–	–	–	–	–
0% RH to 59% RH	-5	–	5	% RH	–
60% RH to 100% RH	-8	–	8	% RH	–
Accuracy (best fit straight line)	-3.5	–	+3.5	% RH	1
Hysteresis	–	3	–	% RH	–
Repeatability	–	±0.5	–	% RH	–
Settling time	–	–	70	ms	–
Response time (1/e in slow moving air)	–	5	–	s	–
Stability (at 50% RH)	–	1.2	–	% RH	–
Voltage supply	4	–	5.8	Vdc	2
Current supply	–	200	500	µA	–
Voltage output (1 <sup>st</sup> order curve fit)	$V_{OUT} = (V_{SUPPLY}) \cdot (0.0062(\text{sensor RH}) + 0.16)$ , typical at 25 °C				
Temperature compensation	True RH = (Sensor RH) / (1.0546 – 0.00216T), T in °C				
Output voltage temperature coefficient at 50% RH, 5V	–	-4	–	mV/°C	–
Operating temperature	-40[-40]	See Figure 1.	85[185]	°C[°F]	–
Operating humidity	0	See Figure 1.	100	% RH	3
Storage temperature	-50[-58]	–	125[257]	°C[°F]	–
Storage humidity	See Figure 2.			% RH	3



**Figure 5. Mounting Dimensions (For reference only. mm/[in])**



**Figure 6. Typical Application Circuit**



# CAMERA CLIMATICA: MEMMERT HPP110

## CARATTERISTICHE COSTRUTTIVE

- STRUTTURA ESTERNA, PORTA E CAMERA INTERNA IN ACCIAIO INOX, COMPLETAMENTE ISOLATA TERMICAMENTE
- PORTA IN ACCIAIO INOX SISTEMA DI RISCALDAMENTO AD ALETTE IRRAGGIANTI CON ALLOGGIAMENTO DELLE RESISTENZE ALL'INTERNO DI NERVATURE RIPARTITE SULLE 4 PARETI LATERALI DELLA CAMERA INTERNA PER UNA DISTRIBUZIONE OMOGENEA E UNIFORME DELLA TEMPERATURA

## CARATTERISTICHE TECNICHE

- CIRCOLAZIONE D'ARIA NATURALE
- 2 SENSORI DI TEMPERATURA PT100 A 4 FILI
- CAMPO DI TEMPERATURA [5, 70] °C
- CAMPO DI UMIDITÀ RELATIVA [10, 90] % RH
- DATA-LOGGER INTERNO
- ATMOCONTROL SOFTWARE WINDOWS PER SCARICO DATI A PC DAL DATA-LOGGER INTERNO, INVIO AUTOMATICO DI SEGNALI DI ALLARME A UNO O PIÙ INDIRIZZI EMAIL, **PROGRAMMAZIONE, IMPLEMENTAZIONE DEI PROFILI DI TEMPERATURA ED UMIDITÀ RELATIVA**



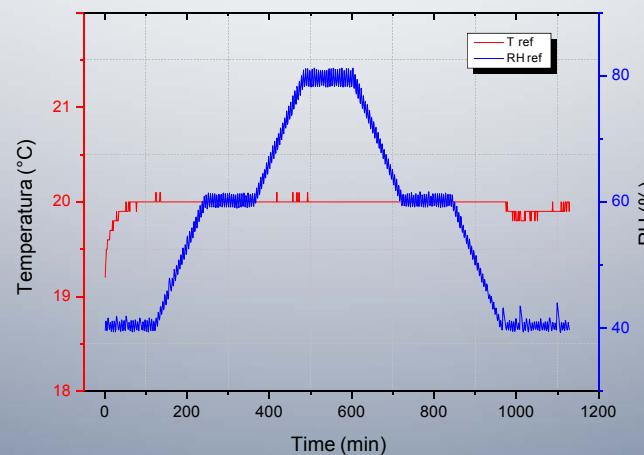
# TARATURA IN CAMERA CLIMATICA

1. TEST IN Umidità Relativa a Temperatura Costante:

- VARIAZIONE DI RH ALL'INTERNO DI UN RANGE PRESTABILITO;

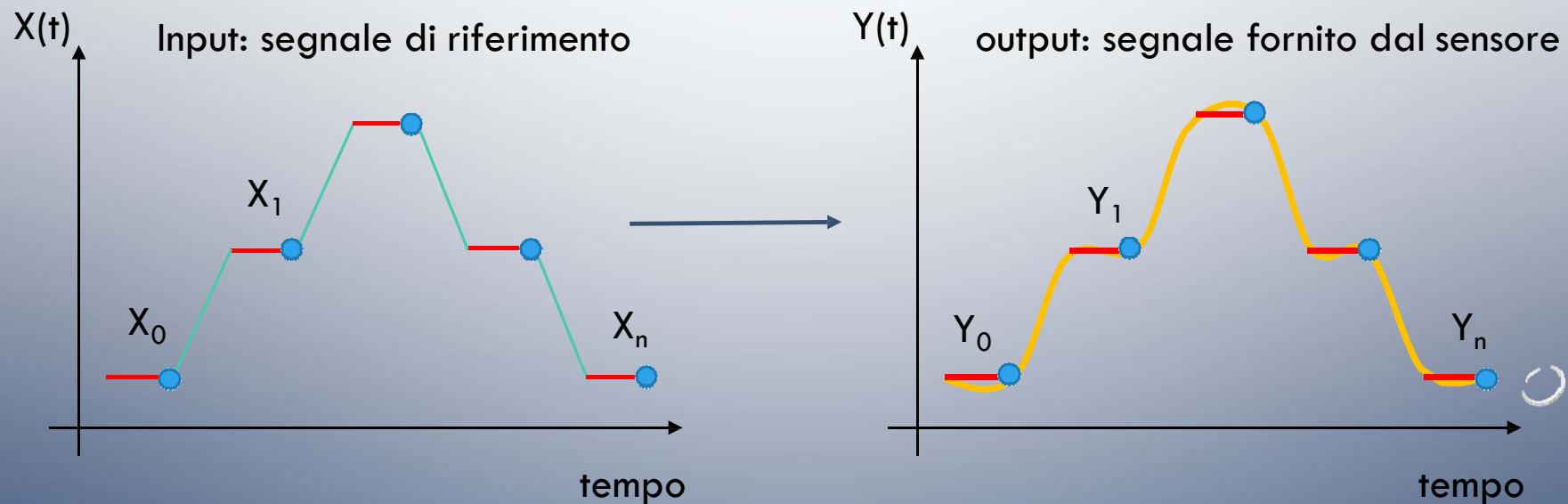
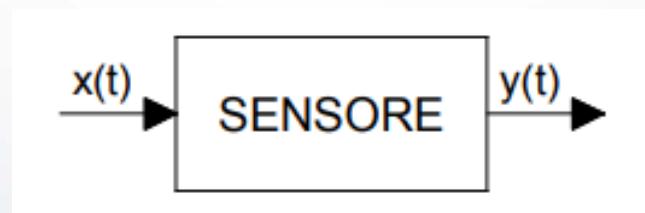
2. ACQUISIZIONE, ELABORAZIONE, INTERPRETAZIONE DEI DATI DELLE RISPOSTE SENSORIALI A SEGUITO DELLE VARIAZIONI IMPOSTE DAI CICLI PROGRAMMATI IN CAMERA CLIMATICA;

3. DERIVAZIONE DELLE CURVE DI CALIBRAZIONE E DELLE SENSIBILITÀ ALLA TEMPERATURA ED ALL'UMIDITÀ RELATIVA.



# PROCEDURA DI TARATURA: LA FUNZIONE DI TRASFERIMENTO

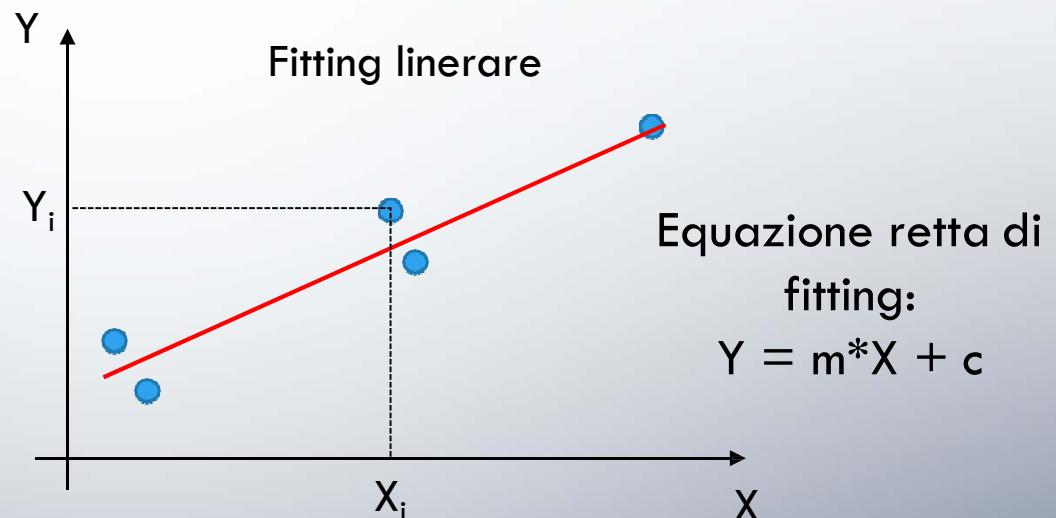
- LA **FUNZIONE DI TARATURA** (o **FUNZIONE di TRASFERIMENTO**), E' LA RELAZIONE CHE PERMETTE DI RICAVARE DA OGNI VALORE DELLA GRANDEZZA DI USCITA IL CORRISPONDENTE VALORE DEL MISURANDO.



# PROCEDURA DI TARATURA: LA SENSIBILITÀ

- S'INDIVIDUA UNA RELAZIONE ANALITICA CHE LEGA LE VARIAZIONE DELL'USCITA (Y) ALLE VARIAZIONI DELL'INGRESSO (X)

Ingresso X	Uscita Y
$X_0$	$Y_0$
$X_1$	$Y_1$
$X_2$	$Y_2$
$X_3$	$Y_3$
$X_4$	$Y_4$

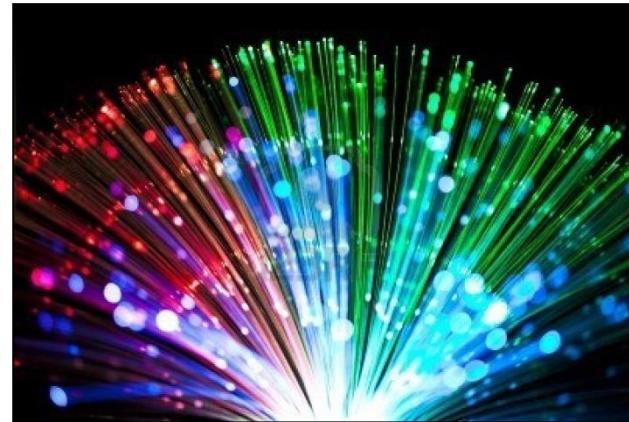


- SENSIBILITÀ:** È DEFINITA COME IL RAPPORTO FRA LA VARIAZIONE DEL SEGNALE DI USCITA SUL SEGNALE DI INGRESSO. SI DEDUCE CHE SE LA CARATTERISTICA È LINEARE, LA SENSIBILITÀ SI MANTIENE COSTANTE IN TUTTO IL CAMPO DI UTILIZZO E COINCIDE CON IL COEFFICIENTE ANGOLARE (PENDENZA) DELLA RETTA.

$$\text{Sensibilità} = m = \frac{\Delta Y}{\Delta X}$$



# Lesson : Fiber Optic Sensors





## Outline

- *Introduction*
- *Fiber Optic Sensors*
- *Fiber Bragg Grating Sensors*
- *Industrial Applications: Case Studies*



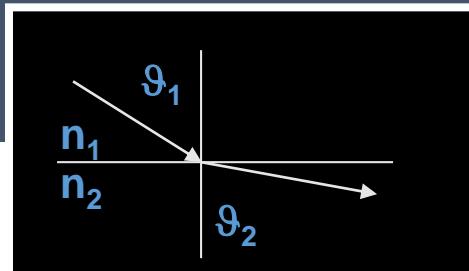
## Optical Fiber

- a filament of transparent dielectric material, glass or plastic
- usually cylindrical in shape
- a guidance system for light

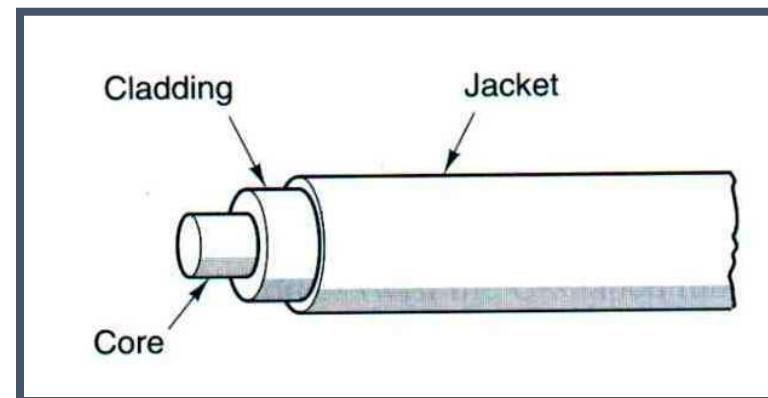
# Optical Fiber

## SNELL'S LAW:

$n_1 \sin \theta_1 = n_2 \sin \theta_2$   
where  $n$  is refractive index



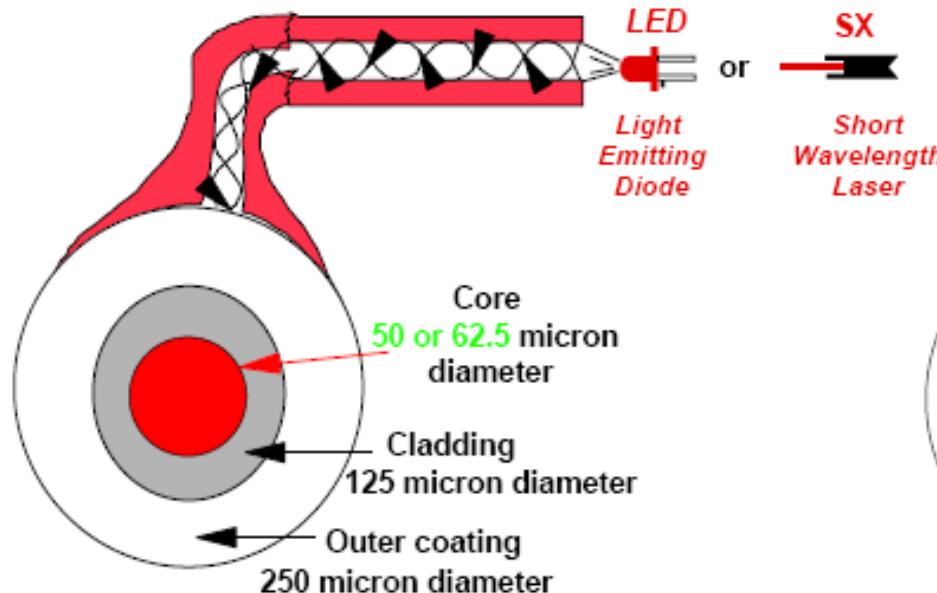
- Guidance is achieved through multiple reflections at the fiber walls.
- Core, transparent dielectric material, surrounded by another dielectric material with a lower refractive index called cladding. ( $n_1 > n_2$ )
- In practice, there is a third protective layer called jacket.



# Fiber Optic Technology

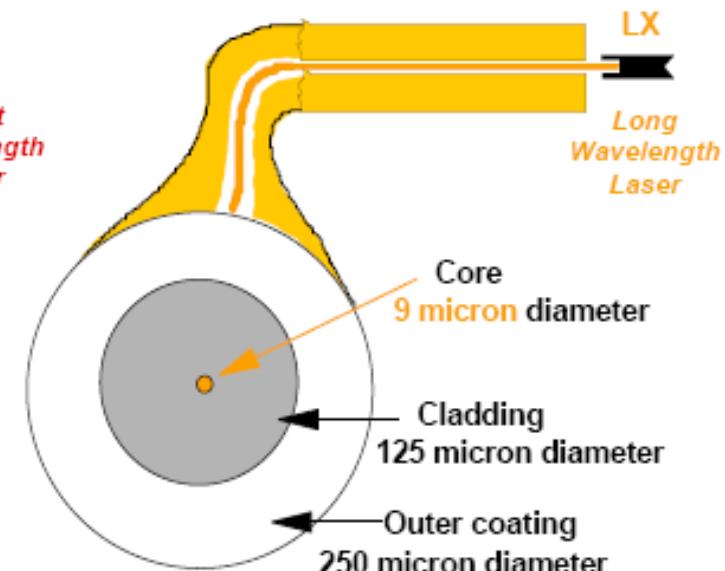
## Multimode (MM) fiber

" Multiple paths" for the light to travel

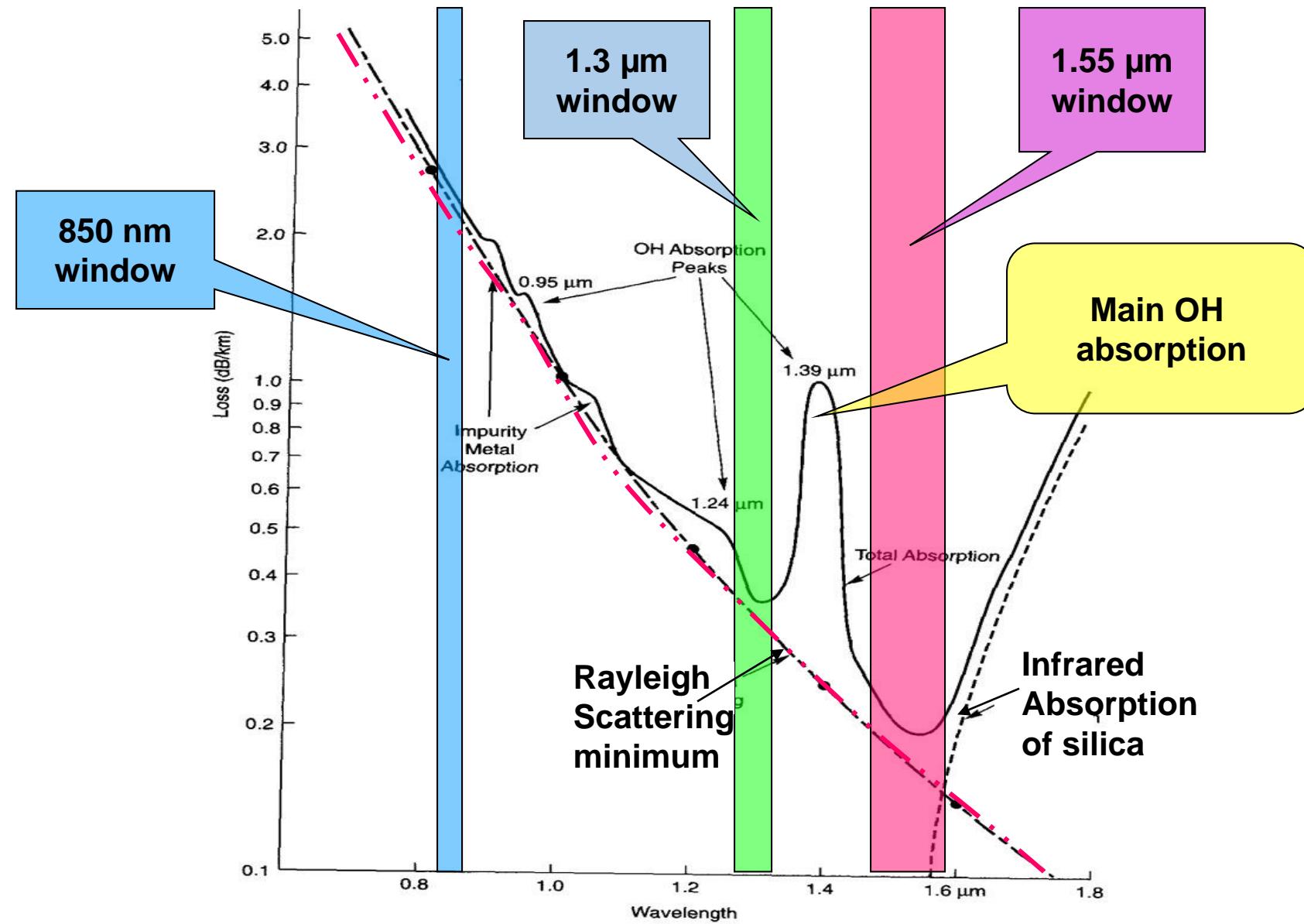


## Single mode (SM) fiber

"Single path" for the light to travel

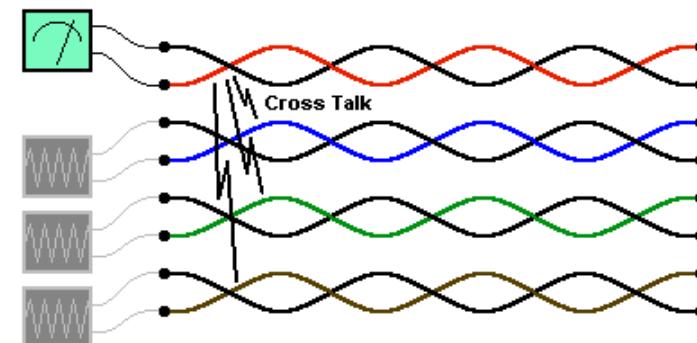


For comparison purposes  
this is the relative size of a  
human hair (@ 70 microns)



- Capacity: much wider bandwidth
- Crosstalk immunity
- Immunity to static interference
  - Lightening
  - Electric motor
  - Fluorescent light
- Higher environment immunity
  - Weather, temperature, etc.
- Safety: Fiber is non-metallic
  - No explosion, no chock
- Longer lasting
- Security: tapping is difficult
- Economics: Fewer repeaters
  - Low transmission loss (dB/km)
  - Fewer repeaters
  - Less cable

## Optical Fiber Advantages



# Why Fiber Optic Sensors?



Source: NASA Langley  
Composite wing IVHM testing



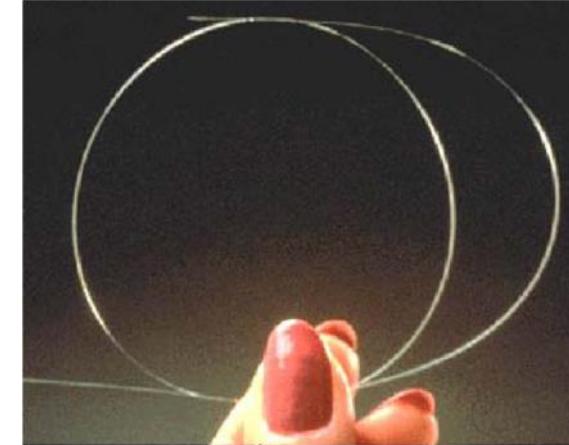
400 conventional strain gauge sensors



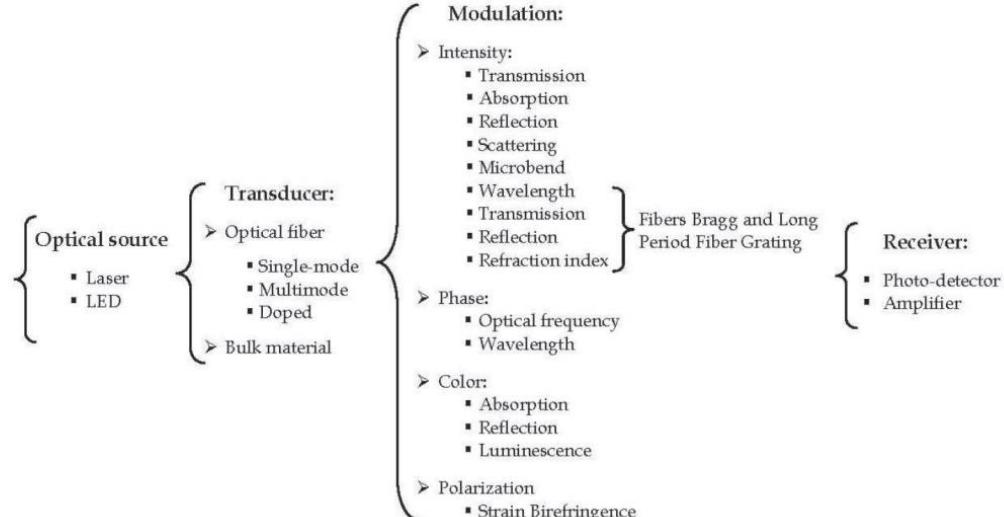
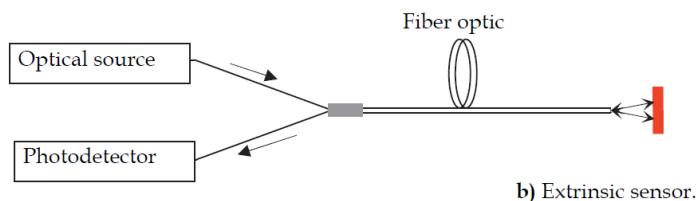
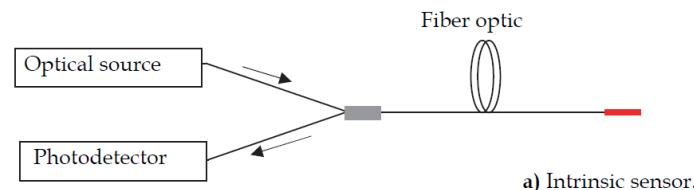
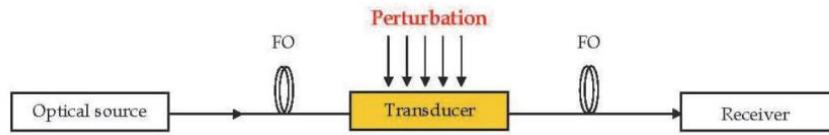
3000 fiber optic FBG strain sensors

# Advantages of Fiber Optic Sensors

- *Galvanic isolation*
- *EMI immunity*
- *Intrinsically safe*
- *Passive: no need for electrical power*
- *Possibility of remote, multiplexed operation*
- *Small size and lightweight*
- *Integrated telemetry: fiber itself is a data link*
- *Wide bandwidth*
- *High sensitivity*

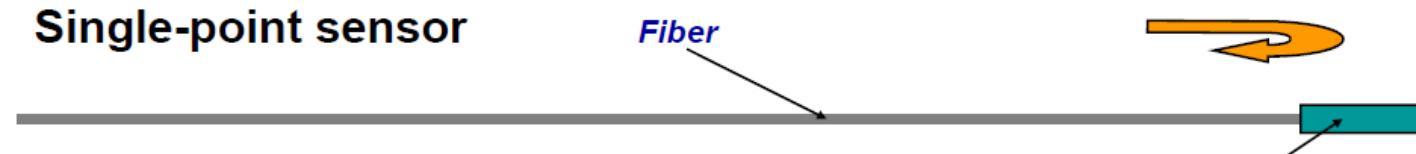


# Fiber Optic Sensors: Fundamentals

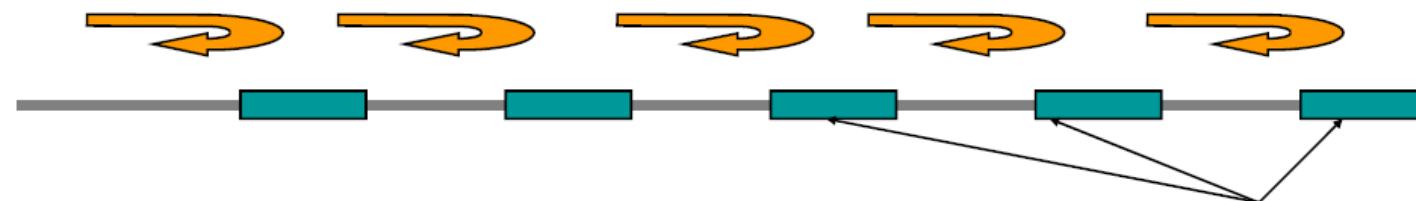


# Fiber Sensors Configurations

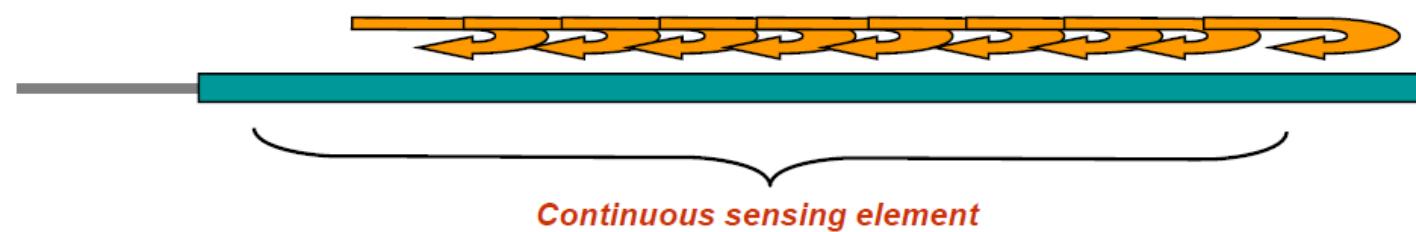
**Single-point sensor**



**Multi-point (quasi-distributed) sensor**



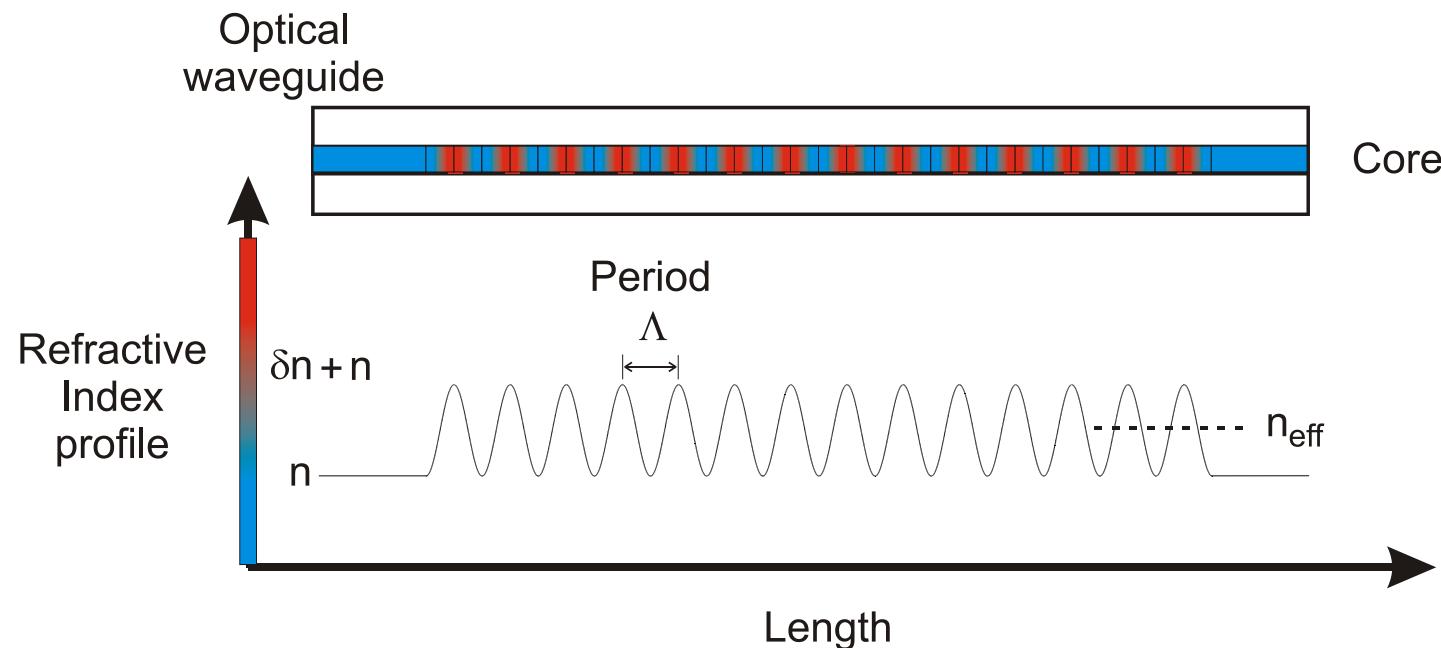
**Distributed sensor**



# ~ Bragg grating fundamentals ~

What is a Bragg grating?.

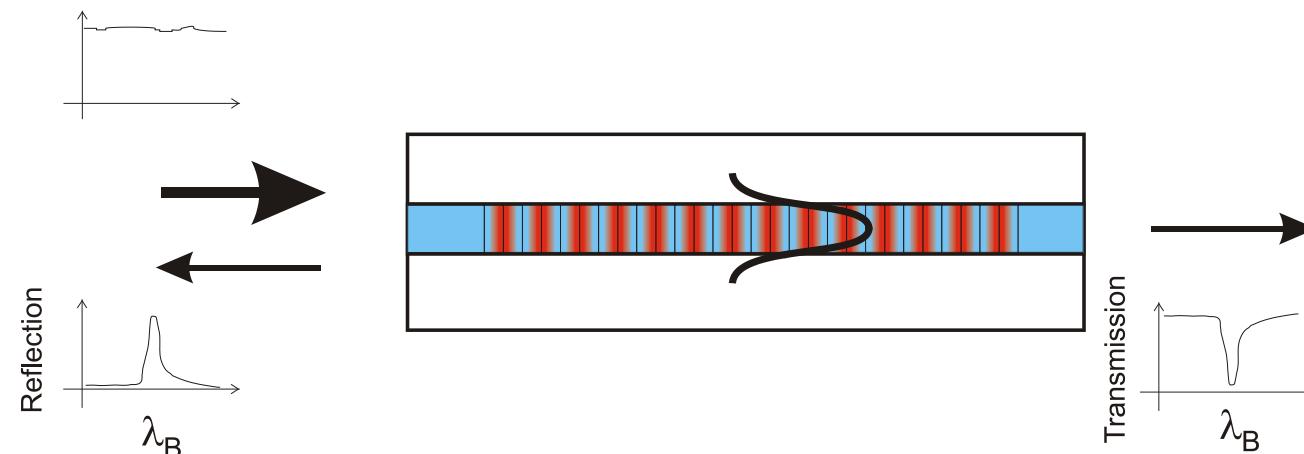
- A periodic or almost periodic structure consisting of a variation of for example the refractive index along the length of a waveguide.



# ~ Bragg grating fundamentals ~

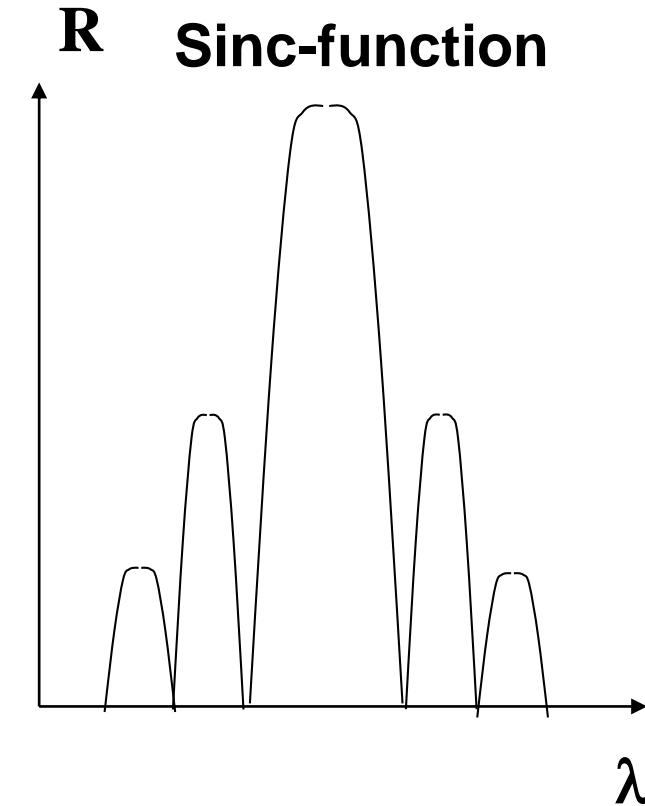
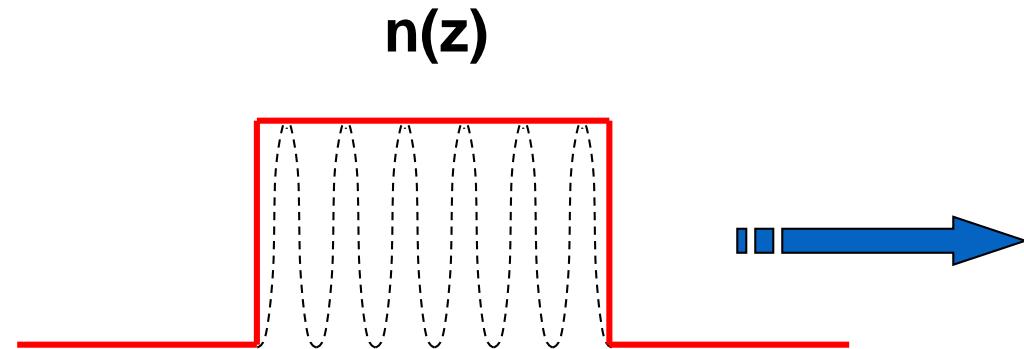
What does it do?.

- Coupling of a forward propagating core-mode to a backward propagating core-mode.
- Acts as a band-rejection filter passing all wavelengths that are not in resonance with the grating and reflecting wavelengths that satisfies the Bragg condition.



# BRAGG GRATINGS APPROXIMATION

- Reflection spectrum a F. T. of the index profile
- Grating starts and stops abruptly
- F. T. of square function is the sinc function





# Achieving Refractive Index Changes

- Create defects in glass
  - Increase absorption (in UV)
  - Affect the refractive index  $n$  through Kramers-Kronig relations
- Densification of glass
  - Increase the refractive index  $n$  directly

Both strategies can be implemented using light exposure



# Photosensitivity in Germania doped silica fibers

Non linear effect which permits the index of refraction in the core of the fiber to be increased by exposure to intense laser radiation

## Without Hydrogen

Index change: up to  $10^{-3}$

*Defects absorb light  
and form new bonds*

*Writing by pumping defect bands:  
240 nm, 330 nm, also 193 nm*

## Hydrogen loaded

Index change: up to  $10^{-2}$

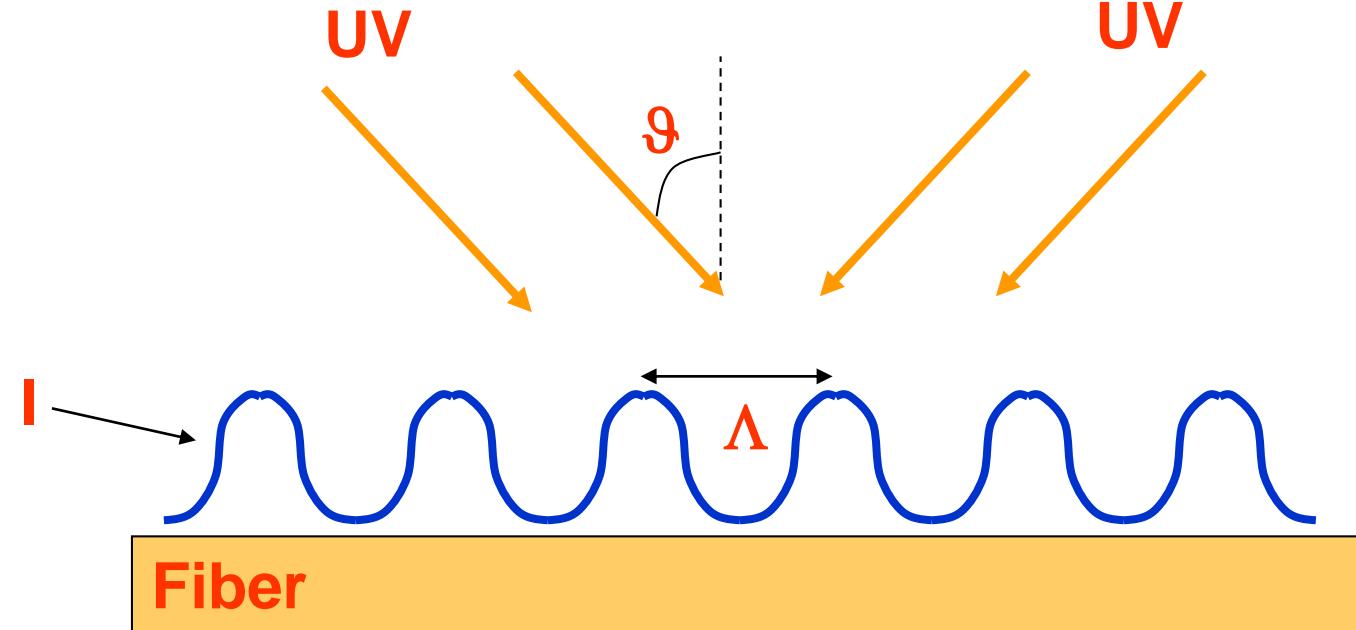
*Photochemical reaction of  
hydrogen with Ge-O bonds*

*Edge of fundamental absorption:  
<310 nm (the shorter the better)*

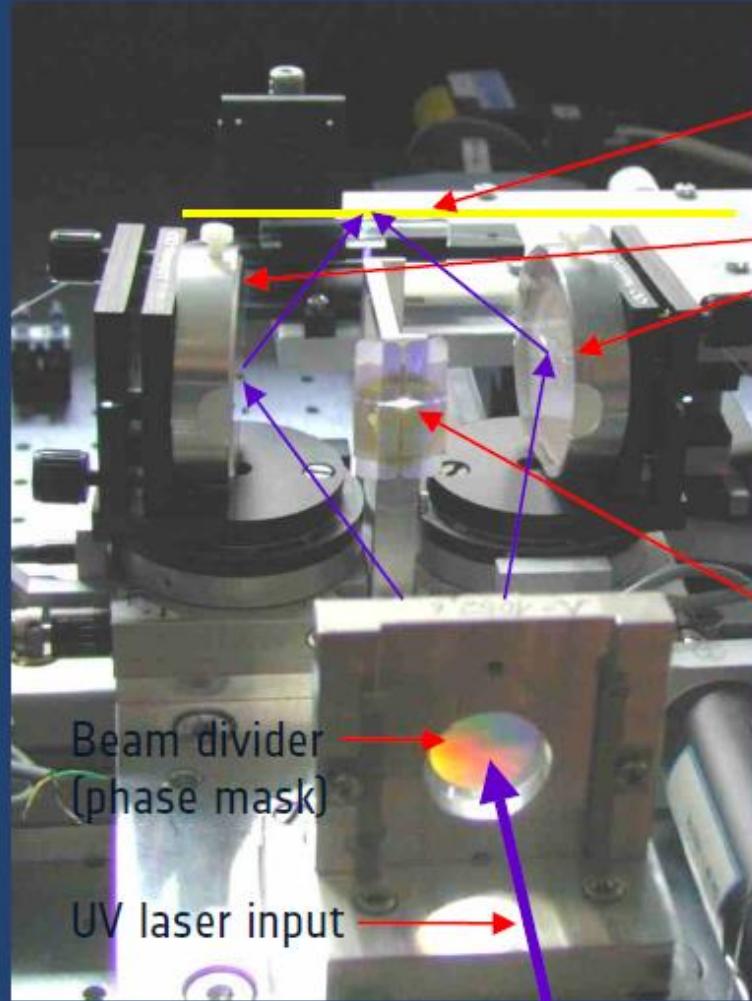
# Gratings Manufacturing: The Holographic Method

- Interfering UV beams:
  - Sinusoidal  $\Delta n$

$$\Lambda = \frac{\lambda_{uv}}{2n_{uv} \sin \theta}$$



# Talbot Interferometer



Optical fibre

2 mirrors (adjustable),  
directing diffracted beams  
of +1<sup>st</sup> and of -1<sup>st</sup> order,  
for superposition  
at optical fibre.

Blocking of 0<sup>th</sup> order

Beam divider  
(phase mask)

UV laser input

# FABRICATION : Phase Mask

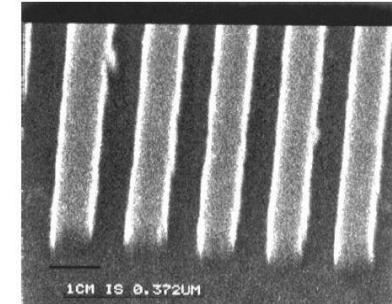
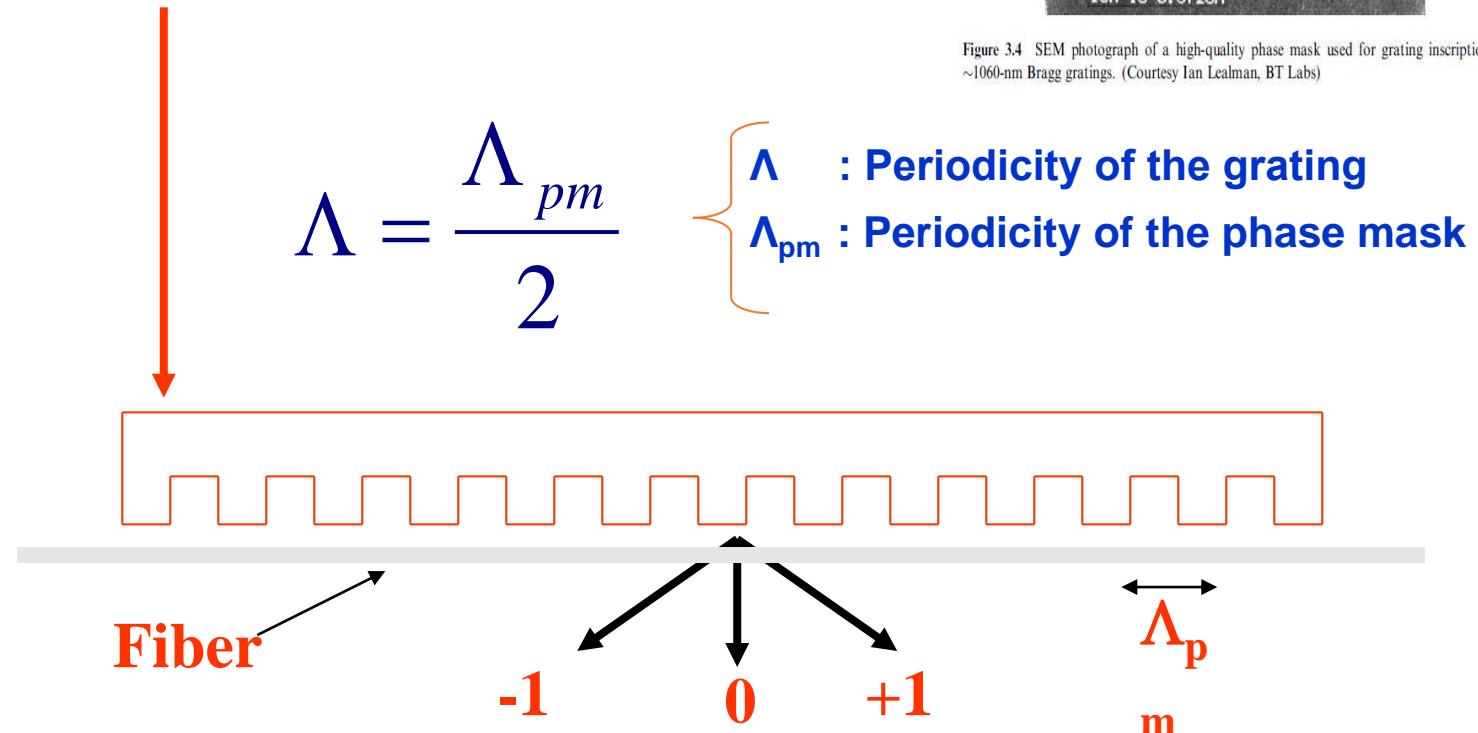
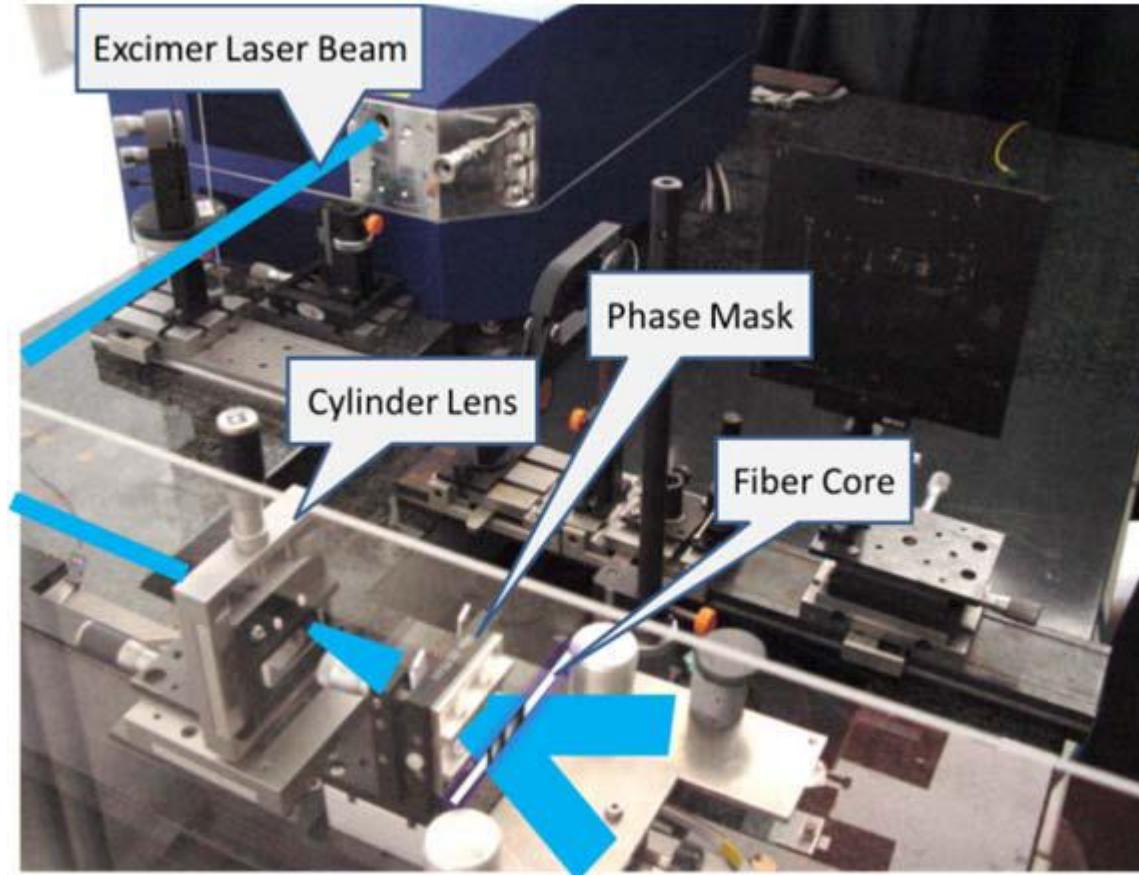


Figure 3.4 SEM photograph of a high-quality phase mask used for grating inscription of ~1060-nm Bragg gratings. (Courtesy Ian Lealman, BT Labs)



## Phase Mask: A practical Set-up

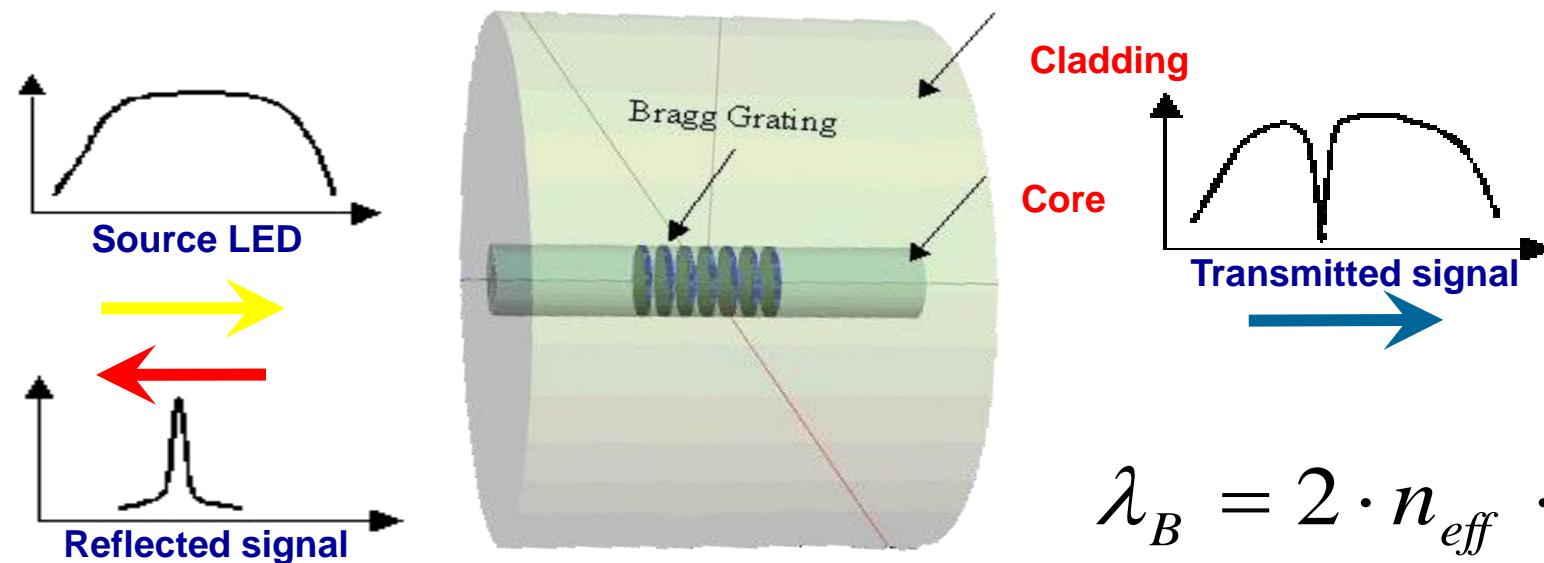


# Fiber Bragg gratings in Single Mode Optical Fibers

## Parameters related to a Bragg grating.

- $n_{\text{eff}} \sim 1.455$  in silica.
- “Short” period grating to operate in lowest order mode ( $m=1$ ) with Bragg wavelength  $\lambda_B \sim 1550\text{nm}$ ,  $\Lambda \sim 500\text{nm}$ .
- Typical index changes,  $\delta n \sim 10^{-5} - 10^{-3}$ .
- Typical lengths, 1cm – 10cm, some types ~1m.

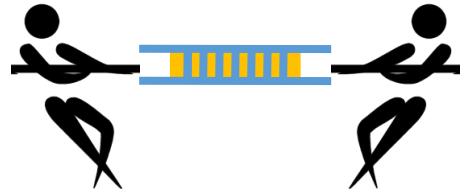
A. Cusano et al., «Fiber Bragg Grating Sensors: Recent Advancements, Industrial Applications and Market Exploitation», Bentham Science Publishers, 2011  
ISBN: 160805084X,



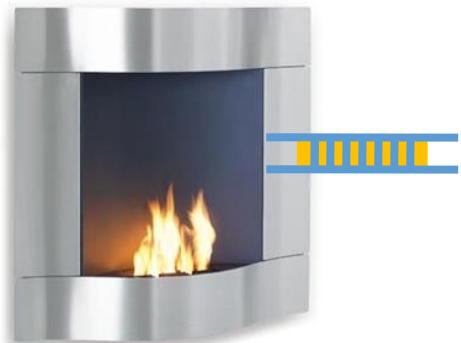
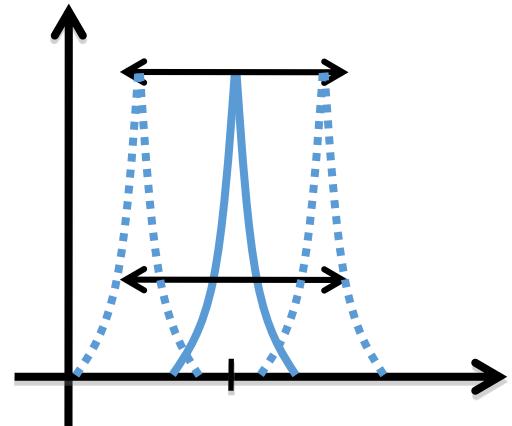
$$\lambda_B = 2 \cdot n_{\text{eff}} \cdot \Lambda$$



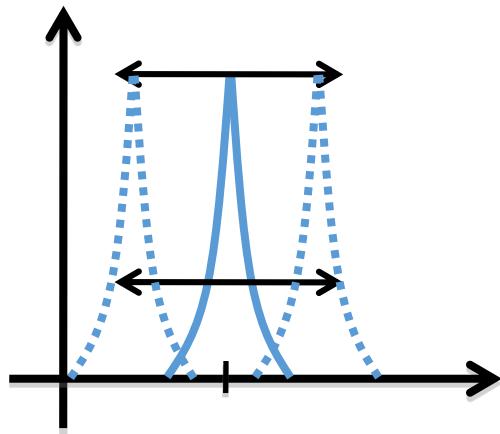
# Temperature & Strain



Strain  
Changes  $\Lambda$  via  
direct  
deformation  
and  $n$  via  
elasto optic  
effect



Temperature  
Change  $n$  via  
thermo optic  
effect and  $\Lambda$   
via thermal  
expansion



# Temperature Sensitivity

$$\left( \frac{\Delta\Lambda}{\Lambda} \right)_T = \alpha \Delta T$$

**Thermal  
Expansion Effect**

$$\left( \frac{\Delta n}{n} \right)_T = \frac{1}{n} \cdot \frac{\partial n}{\partial T} \Delta T$$

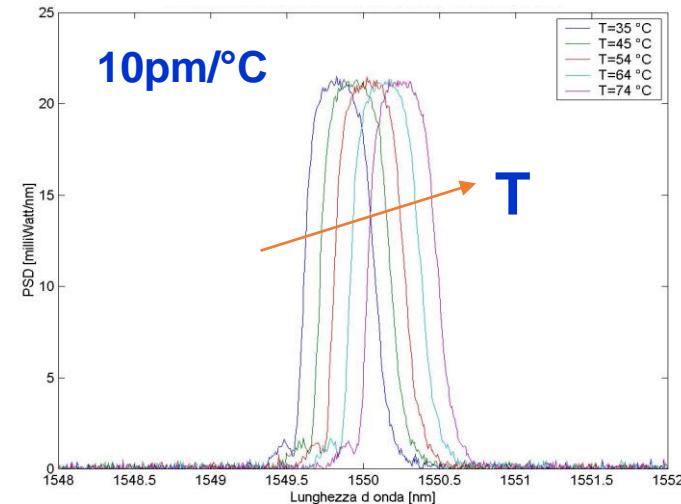
**Thermo Optic  
Effect**

$$\left( \frac{\Delta\lambda_B}{\lambda_B} \right)_T = \left( \frac{\Delta\Lambda}{\Lambda} \right)_T + \left( \frac{\Delta n}{n} \right)_T = \left( \alpha + \frac{1}{n} \cdot \frac{\partial n}{\partial T} \right) \Delta T = S_T \Delta T$$

**$S_T$ : Temperature Sensitivity**

With :

- α : Optical fiber core temperature expansion coefficient
- Λ : Periodicity of the grating
- n : Optical fiber core refractive index



# STRAIN SENSITIVITY

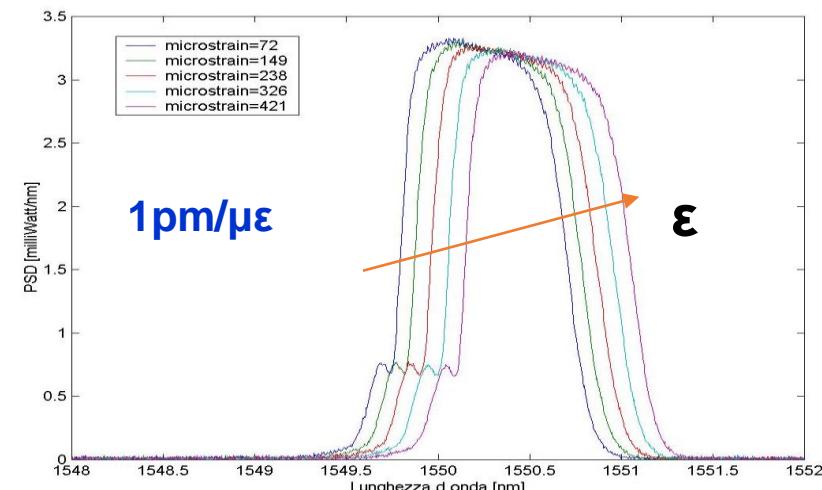
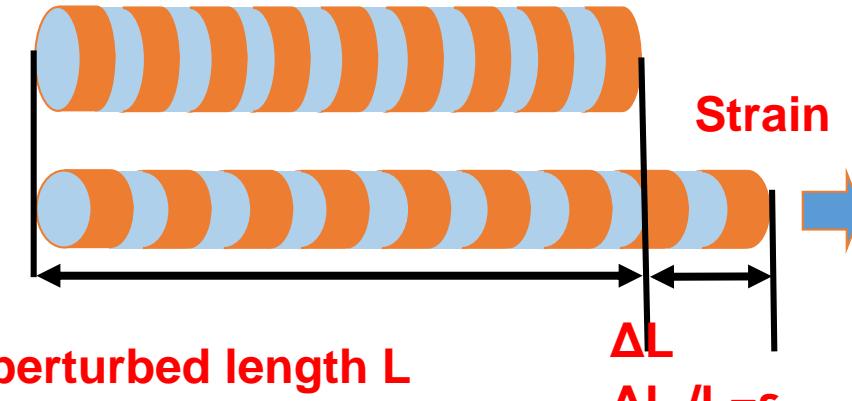
## Axial Strain

$$\left\{ \begin{array}{l} \varepsilon_1 = \varepsilon \\ \varepsilon_2 = \varepsilon_3 = -\nu\varepsilon \\ \nu: \text{Poisson ratio} \end{array} \right.$$



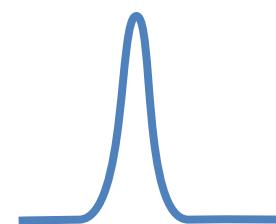
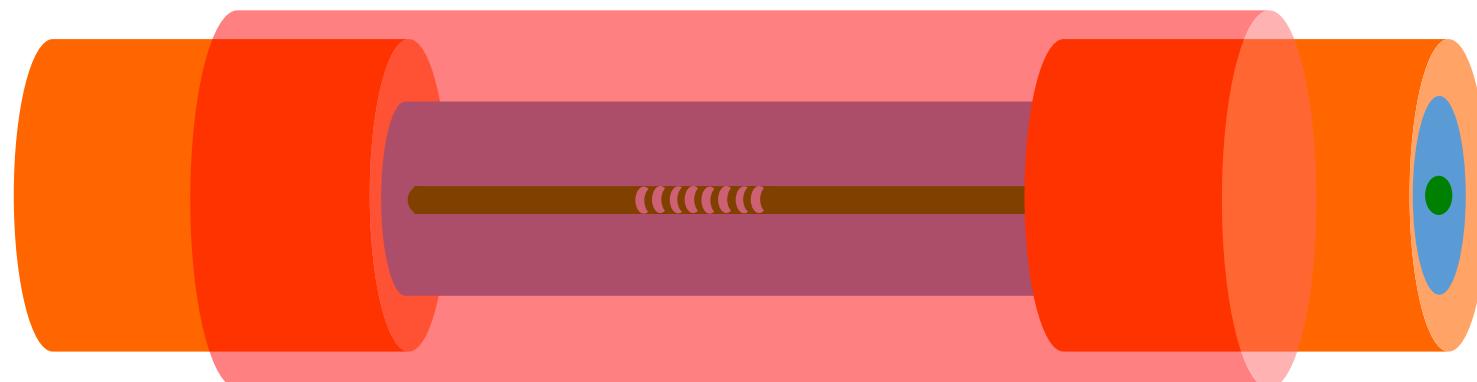
$$\left( \frac{\Delta\lambda_B}{\lambda_B} \right)_\varepsilon = \left( \frac{\Delta\Lambda}{\Lambda} \right)_\varepsilon + \left( \frac{\Delta n}{n} \right)_\varepsilon =$$

$$\left\{ 1 - \frac{n^2}{2} [p_{12} - \nu \cdot (p_{11} + p_{12})] \right\} \varepsilon_1 = S_\varepsilon \varepsilon$$

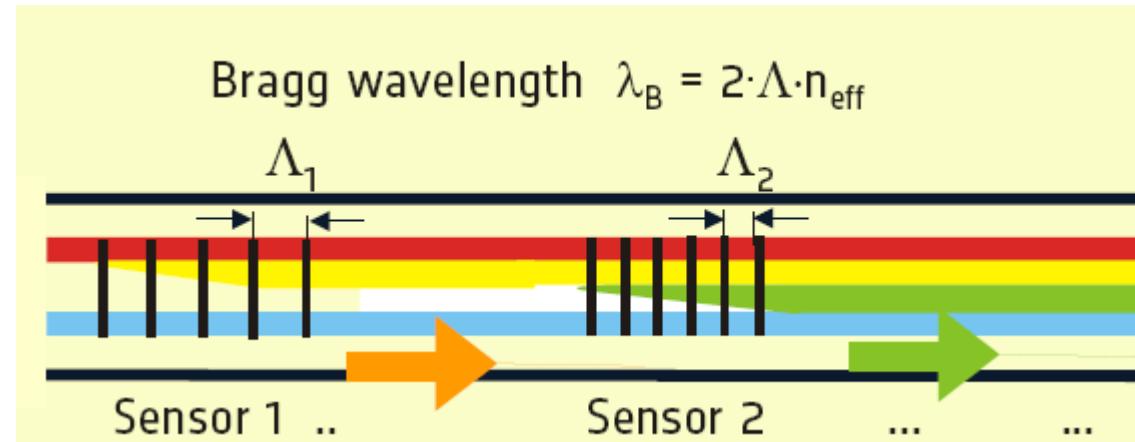


# FBGs AS MULTIFUNCTIONAL SENSORS

- **Functionalization:** integration with appropriate materials and suitable packaging (i.e. mechanical packaging, magnetostRICTIVE, polymeric coatings) to measure a number of physical, chemical and biological parameters (i.e. magnetic field, humidity, cryo temperatures, acoustic waves, weight, chemical and biological analytes)



# Fibre Bragg Grating Sensor and Multiplexing Principle



Hundreds of FBG sensors can be realized within a single optical fibers by changing the grating period

$$\Delta\lambda_{B-i} = 2n_i \Delta\Lambda_i$$

Maximum number of sensors is ruled by taking into account spectral range of the light source and as well as the maximum wavelength shift of each sensor according to:

$$N < \Delta\lambda_{\text{Source}} / \Delta\lambda_{\text{BMAX}}$$

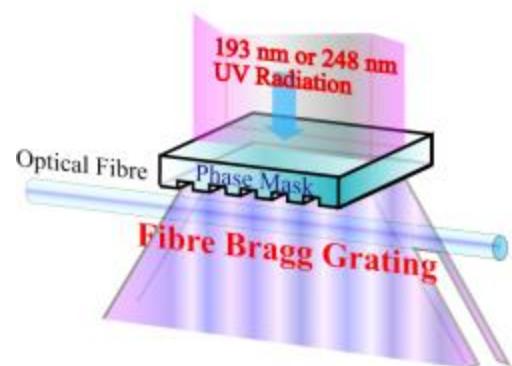
# FBG Sensors: A Standardised Technological Platform

... An attractive sensing solution:

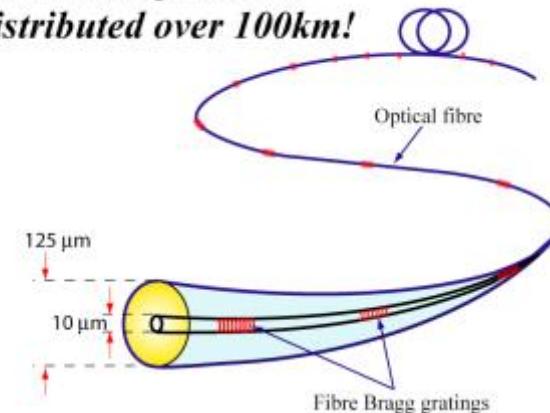
- *Wavelength Encoded*
- *Self Referencing*
- *Linear Output*
- *Small and Lightweight*
- *WDM & TDM Multiplexing*
- *Mass Producible*

*Reflective & Transmission Operation*

- *Single & Multi Point Sensing*
- *Multi Parameter Sensing*
- *Long Range*
- *Reasonable Cost*
- *Durable*



*Hundreds of sensors distributed over 100km!*





## FBG TECHNOLOGY EVOLUTION: Major Milestones

- 
- The timeline chart illustrates the evolution of FBG technology through a series of major milestones, each marked by a red vertical line and a corresponding year and description.
- 1978: Discovery of photosensitivity in fibers (K.O. Hill)
  - 1989: UV side-writing technique (Meltz, Morey & Glenn)
  - 1993: Mask writing technique (K. O. Hill)
    - Hydrogen loading photosensitization (Lemaire)
  - 1995: Commercial FBG production (3M, Bragg Photonics, Innovative Fibers, Indx, etc...)
  - 1997: First commercial FBG interrogator (Electro-Photonics)
    - Deployment of FBGs in WDM systems (Ciena)
  - 2003: Commercial reel-to-reel FBG arrays (LxSix, Sabeus)
  - 2007: First pre-packaged FBG strain gauge (Micron Optics)
    - First strain gauge company to offer FBG sensors (HBM)
  - 2009: Standardization efforts (W. R. Habel et al. BAM Institute-Jena Germany)
  - 2010: First FBG sensors installation at CERN (University of Sannio & Optosmart)
  - 2013: First Certified FBG Sensing System for Weighing in Motion (Ansaldo STS, University of Sannio & Optosmart)

# FBG-Based Sensors & Arrays:

## Multiple Parameters & Suppliers



Accelerometer



Displacement meter



Strain meter



Pressure meter



Thermometer



Incline meter

Reliability is the main  
commercial issue



# Applications: Segment Areas



## Oil & Gas

- Reservoir monitoring
- Downhole P/T sensing
- Seismic arrays



## Energy Industry

- Power plants
- Boilers & Steam turbines
- Power cables
- Turbines
- Refineries

## Aerospace

- Jet engines
- Rocket & propulsion systems
- Fuselages



## Underwater

- Leaks in subsea pipeline monitoring
- Flood detection
- Hydrophone



## Civil

- Bridges
- Dams
- Road
- Tunnels
- Land slides

## Transportation

- Rail monitoring
- Weight in motion
- Carriage safety





# Smart Railways Using Fiber Optic Technology



# Needs and Requirements

- Increasing need for improved **safety**, **reliability** and **efficiency** in the worldwide railway industry;
- Monitoring of railway traffic and in particular rail truck integrity;
- Require the Development of **smart monitoring systems** for real-time and continuous monitoring of the structural and operational conditions of rail tracks;
- Advantages for railway industry: reduced maintenance costs, optimized performance and capacity;
- Railway monitoring requires **extensive sensor networks** (1,000s of sensors) for multifunctional measurements (strain, vibration, velocity, temperature, acceleration, **weighing in motion**, traffic status, structural health ... ).



- Since 2007: exclusive agreement with Ansaldo STS (originally with Finmeccanica group, currently part of Hitachi group) for the development of trackside monitoring systems based on the **use of fiber optic sensors** to improve safety and security levels of the national transportation system.



Early warning for overloads and imbalances

- ✓ Reduce tracks damaging
- ✓ Reduce on board failures

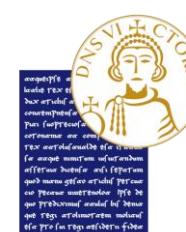


Early Warning for Defected Wheels

- ✓ On condition maintenance
- ✓ Reduce infrastructure damaging

- Year 2015: we developed a **Novel and Smart monitoring System** entirely based on optical fiber technology and fiber optic sensors, able to perform **weighing in motion (WIM)** as well as **wheel impact load detection (WILD)** and wheel defect identification in railways assets.

 **Hitachi Rail STS**

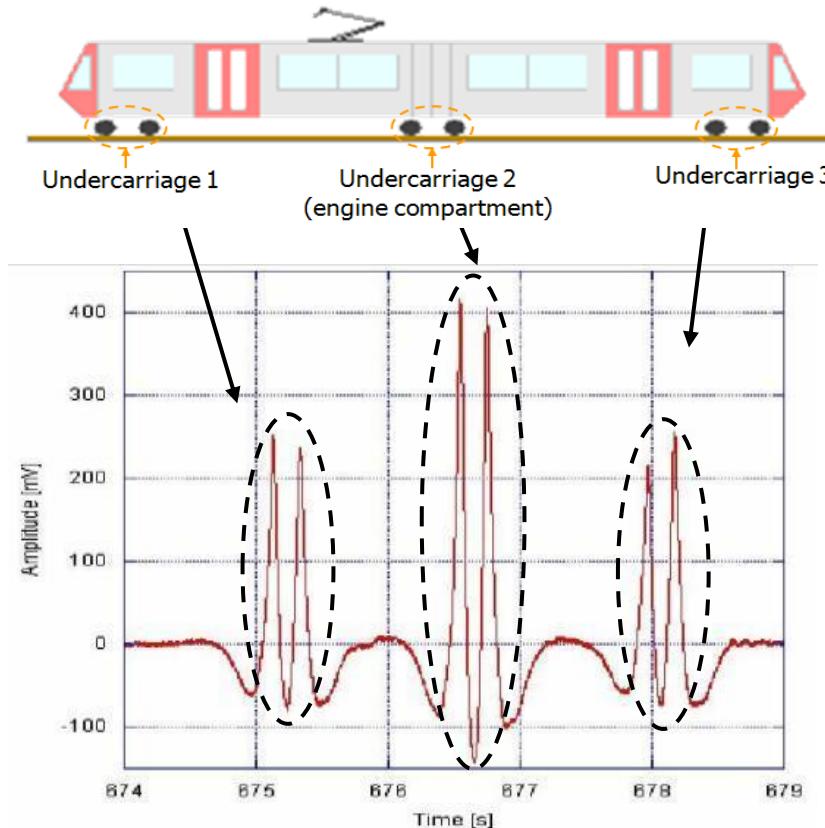
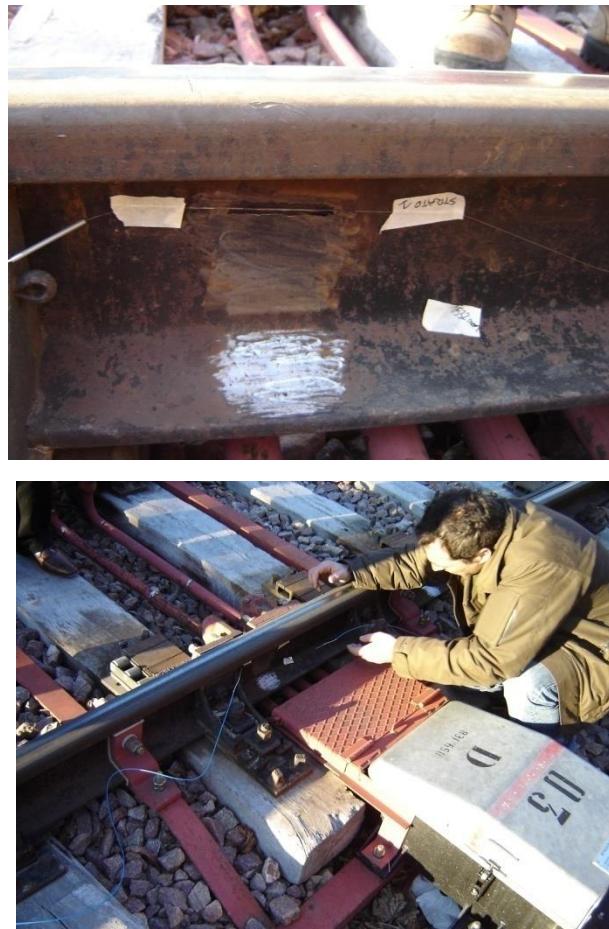


  
**optosmart**  
OPTOELECTRONIC AND SMART SYSTEMS

- Year 2021: The WIM/WILD system, presented by Optosmart, was selected among the 100 best industrial innovations (on 3 thousand projects) for international technology transfer within the prestigious China Zhongguancun (ZGC) Forum 2021.

# Feasibility: Year 2007 “Tel Station (Bz)

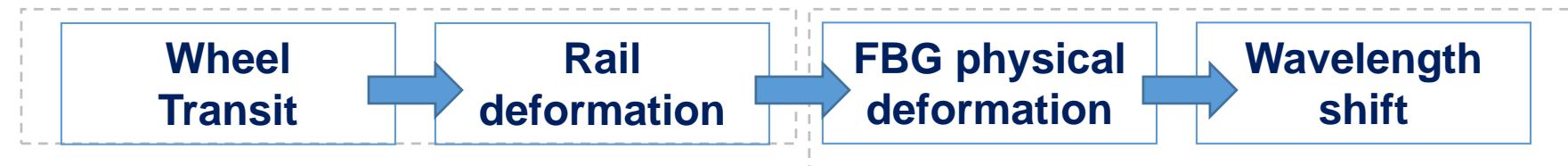
“Bare FBG” glued to the rail



A Single FBG glued to the rail, can provide useful information about:

- occupation state, train, velocity, acceleration, axle counting, weighing in motion,

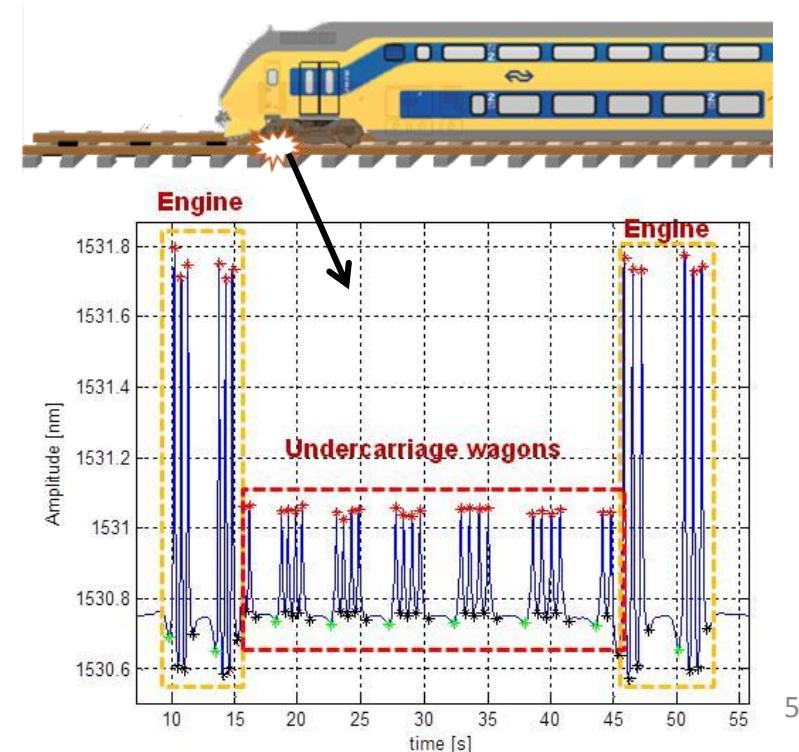
The single FBG sensor, installed under the rail foot (between two sleepers), detects the vertical forces generated by the wheel/rail contact, and returns an output signal in the time domain composed by several peaks.

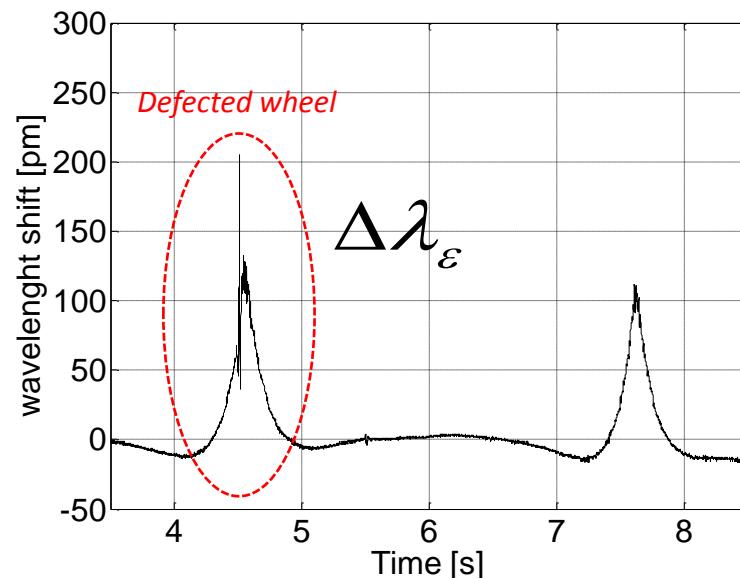


- The single peak is related to the single wheel passing over the sensing region,
- The amplitude of the peak retrieves information about the weight of the associated train wheel.

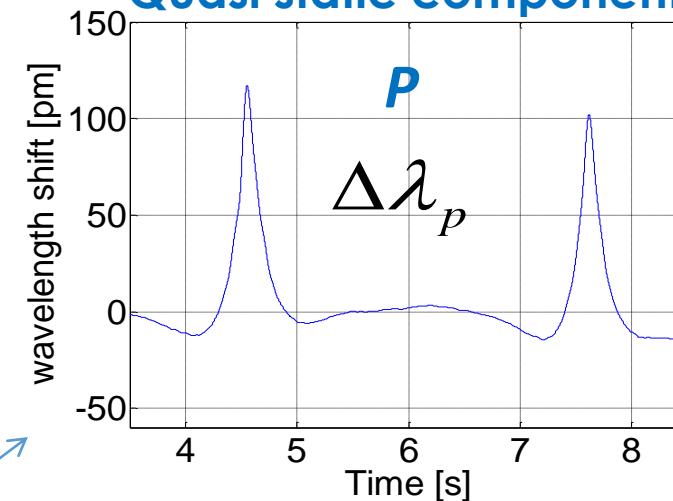
$$P[kg] = \alpha * \Delta\lambda_{\varepsilon} \Rightarrow F[kN] = g * P$$

$\alpha$  is the FBG calibration coefficient [kg/pm] measured one time at the end of the installation (or on demand), by using Engine or test train with known weight. ( about 7 kg/pm, 70 N/pm on UNI50 rail type)

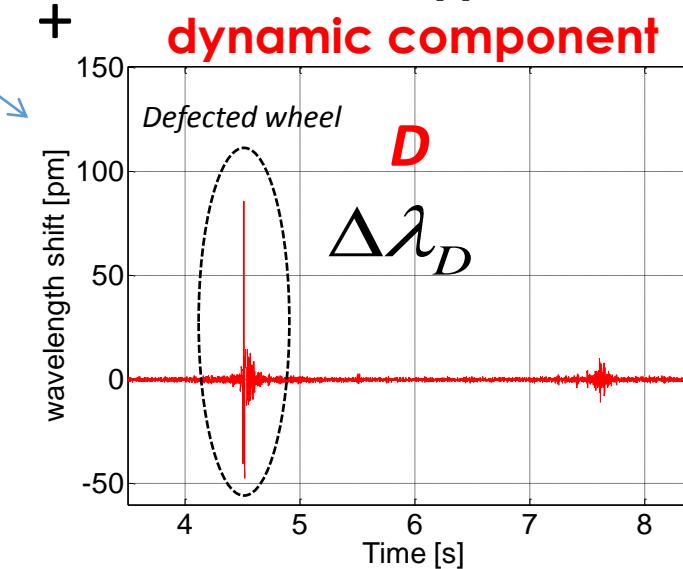




## Quasi static component



Measuring the weight of each wheel



Measuring the rolling surface's quality of each wheel

$$\Delta\lambda_e = \Delta\lambda_p + \Delta\lambda_D = \Delta\lambda_{LPF} + (\Delta\lambda_e - \Delta\lambda_{LPF})$$

→

$$F_p [kN] = g * \alpha * \Delta\lambda_p$$

$$F_D [kN] = g * \alpha * \Delta\lambda_D$$

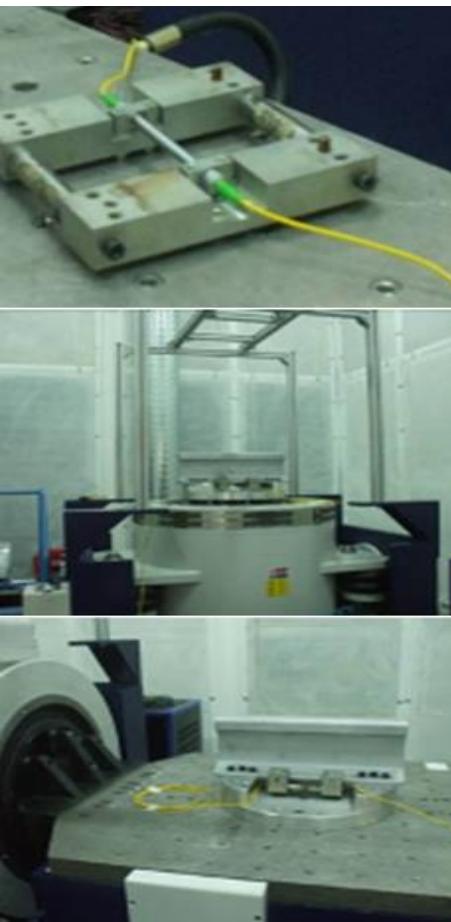
It's composed by a FBG strain sensor integrated in a metallic package (in order to improve robustness features) and a customized stainless steel package designed in order to clamp the sensor under the rail foot (between two sleepers)



- ✓ **Fast and noninvasive installation:** no gluing, no welding and no drilling operations on rail truck are required. They can be easily mounted and removed on the rail in few minutes, **during the normal train service**, and without the interruption of the traffic,
- ✓ Designed for the UIC60 and UNI50 rails, but can be **customized upon request for any kind of rail**,
- ✓ The **repeatability** of the installation is guaranteed (use of dynamometric key to tighten the 8 screws),
- ✓ The sensor is clamped to the point with **maximum deformation of the track**,
- ✓ **No electric power supply and electric equipment near the rail** (fiber optic cabling),
- ✓ In **compliance with the standards EN50125-3 and IS402**,
- ✓ the same sensor retrieves both WIM and WILD measurements,
- ✓ fiber optic and sensors **multiplexing**,
- ✓ **wide working temperature range** [-40,+85] °C.

# Certification Tests

## Mechanical tests



- Vibrations: up to  $280 \text{ m/s}^2$
- Shock:  $2500 \text{ m/s}^2/1\text{ms}$

## Solar irradiation tests



Maximum level of solar radiation :  $1120 \text{ W/m}^2$ . Temperature:  $60^\circ\text{C}$

## Climatic tests



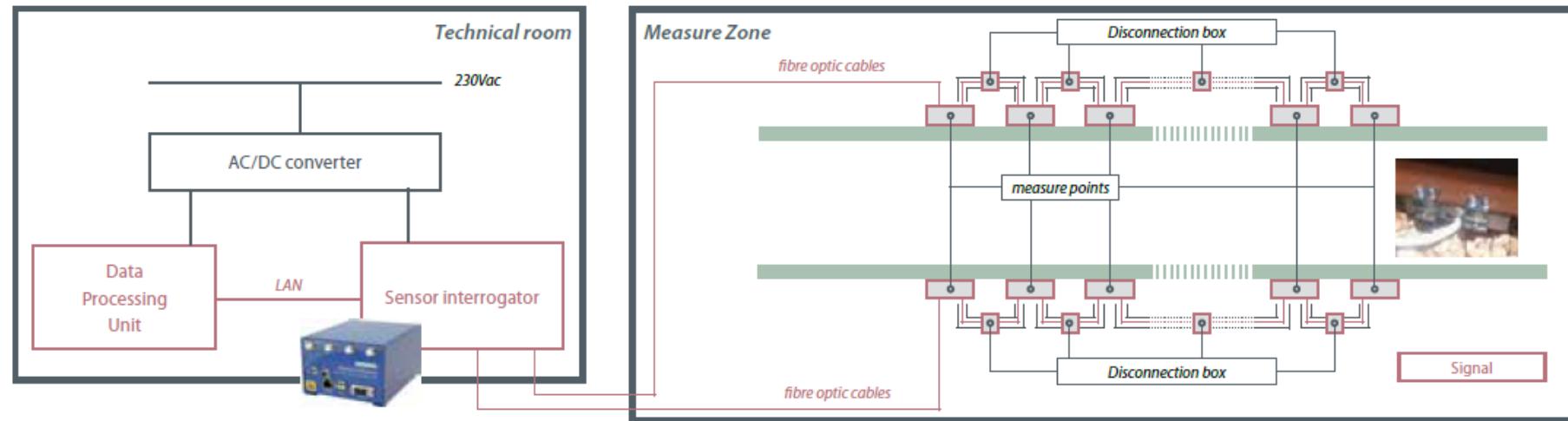
- Temperature changes :  $-40^\circ\text{C} \div +85^\circ\text{C}$
- Hot dry test:  $70^\circ\text{C}$ , U.R.  $\leq 60\%$
- Hot wet test:  $55^\circ\text{C}$ , U.R.  $95\%$
- Cold test:  $-25^\circ\text{C}$

## Immersion tests in sand and water



300 liters of water ( h 50 cm) – 75 kg of sand (h 30 cm)

- The sensors are installed in parallel along the two rails ("Measure Zone"), and the single one is provided with fiber optic cables equipped with connectors. An IP67 box (Disconnection Box) is used to protect the optical connections and allow their maintenance,
- The "Technical Room" is the place in which the optoelectronic interrogation unit (frequency sample: 2.5 kHz, repeatability: less than 1 pm), the acquisition unit and the power supply unit are installed,



- **In WIM configuration:** Train and axle speed, axle counting, Train composition, weight per wheel/axle/wagons/train, transversal and longitudinal unbalance. Alarms for overloads and unbalance conditions.
- **In WILD configuration:** Static and Dynamic impact forces, Peak force, Ratio factor Peak Force/Static force, Ratio factor Dynamic Force/Static Force, Defected wheels identification. Defect classification (on going). Warning for defected wheels, Alarms for high dynamic and peak Forces, Alarms on ratio factors.
  - **WIM & WILD accuracy:** less than 2%
  - **Speed range:** 5-120Km/h (expected **speed range of 5-350 km/h**)
  - **WIM & WILD resolution:** less than 65 kg (0.6 kN).

**COMPLETED**

## METRO NAPOLI (1 WIM/WILD SYSTEM)



**COMPLETED**

## METRO MILANO (1 WIM/WILD SYSTEM)



**ATM**  
AZIENDA TRASPORTI MILANESE S.p.A.



1 year ago!

## ABU DHABI STAGE 1 (3 WIM SYSTEMS INSTALLED AND CURRENTLY IN NORMAL OPERATION)

**COMPLETED**



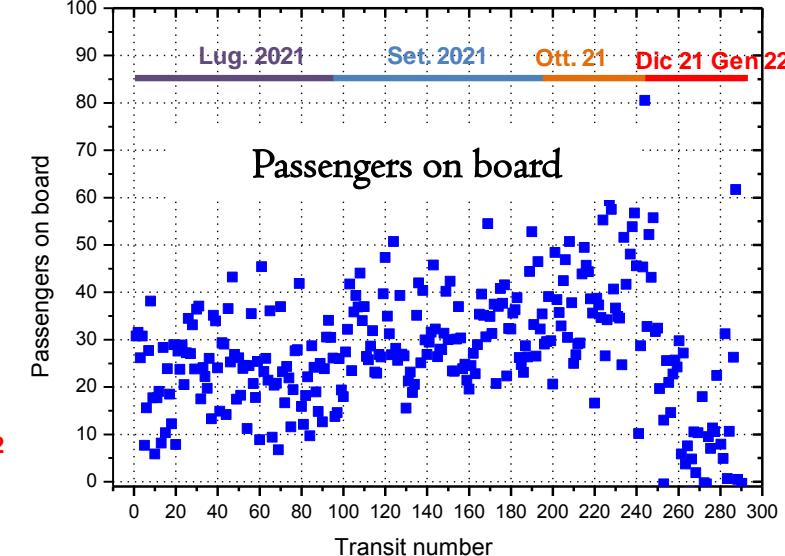
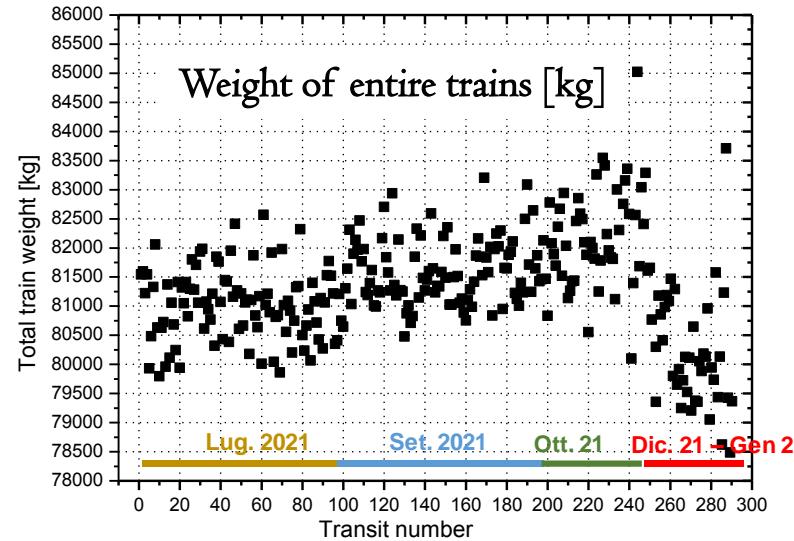
## ABU DHABI STAGE 2 (4 WIM/WILD + 9 WIM SYSTEMS)



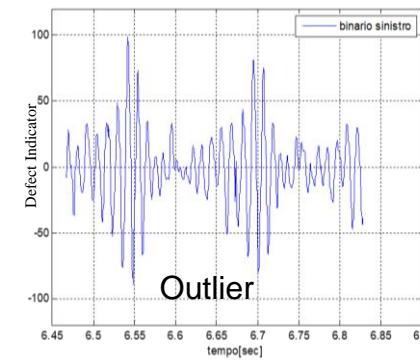
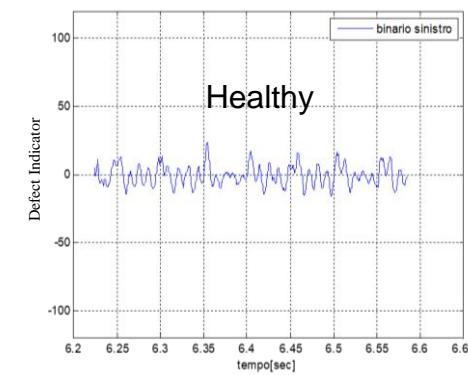
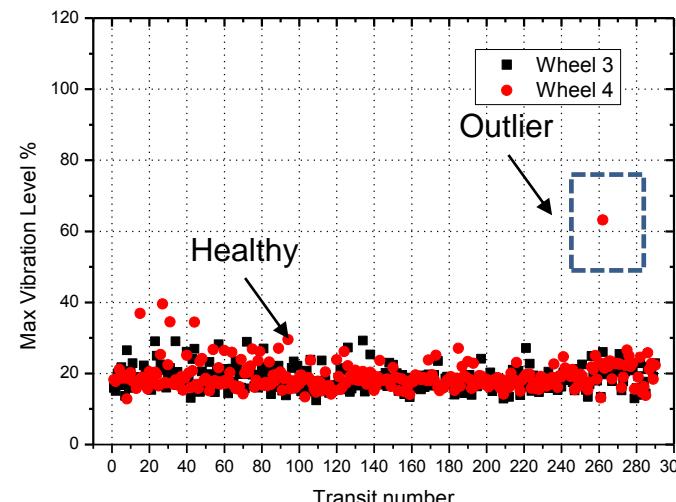
## EAV - NAPOLI (1 WIM/WILD SYSTEM)



✓ WIM results for **290 passenger trains** (selected in seven months), the single one with a nominal empty load of approx. 78 tons, and passengers on board estimation (70 kg per passenger)



✓ WILD results for **about 6000 wheels selected and analysed** in terms of vibration patterns in order to identify healthy and outlier wheels with high vibration levels. Only 54 on 5800 measured wheels returned an out-of-statistical defect indicator.



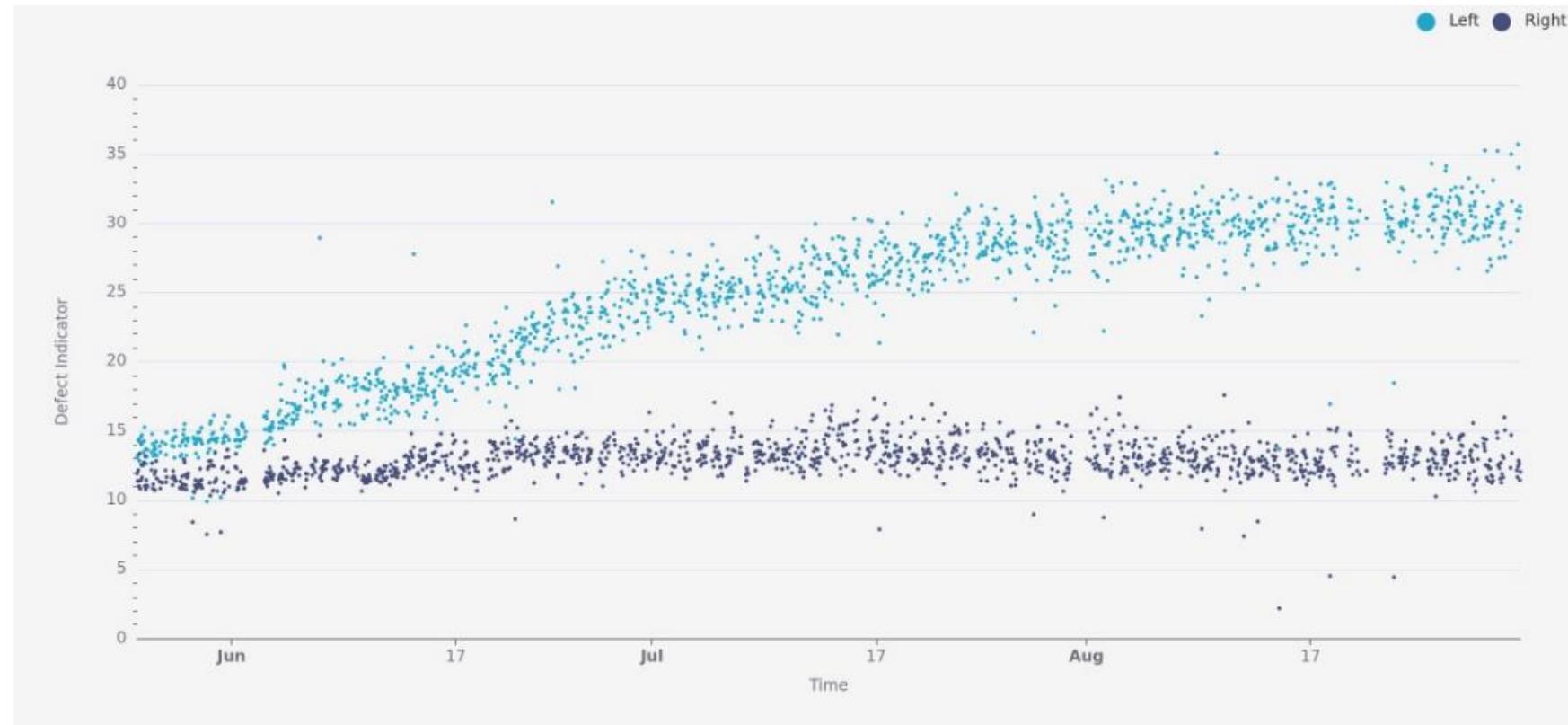
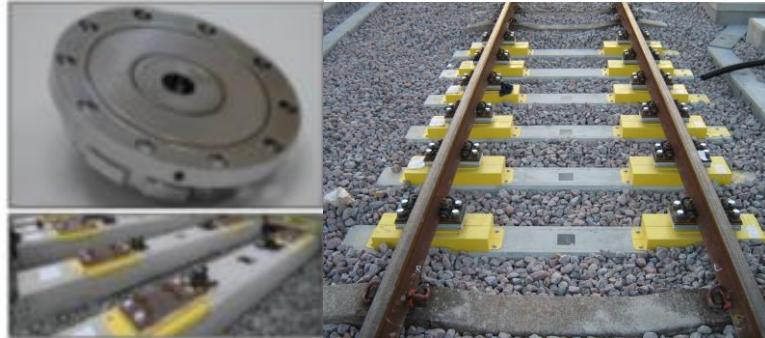


Figure 43 Load of Axle 9 for train "015" over the time

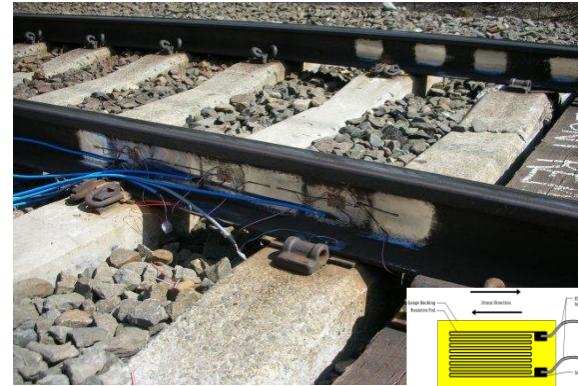
## Load cells



### Svantaggi:

- Area di ingombro elevata;
- Realizzazione multiplexing complessa;
- Connessioni elettriche;
- Sensibilità alle interferenze elettromagnetiche.

## Strain gauge



### Svantaggi:

- Realizzazione multiplexing complessa;
- Connessioni elettriche;
- Sensibilità alle interferenze elettromagnetiche.

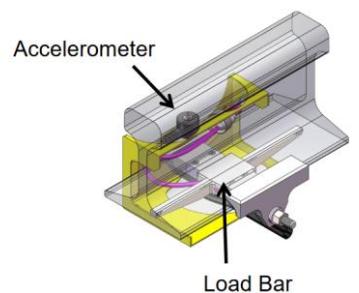
## Accelerometers



### Svantaggi:

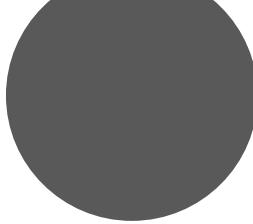
- Realizzazione multiplexing complessa;
- Connessioni elettriche;
- Sensibilità alle interferenze elettromagnetiche;
- Installazione invasiva.

- ATLAS (**Strain gauges**)
- Multirail Wheel Scan (MWS) (**Load cells**)
- Wheel Condition Monitoring (WCM) Track IQ (**Load cells** and **Accelerometers**)

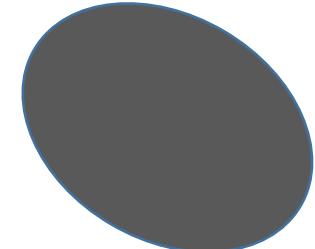


Un sistema WILD deve essere in grado di **individuare, classificare** ed eventualmente **quantizzare** un difetto della ruota.

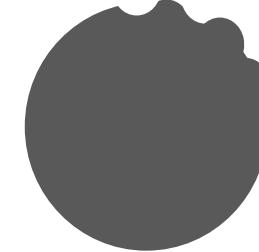
**FLAT**



**OUT of ROUNDNESS**



**TREAD**



### La calibrazione sperimentale di un sistema WILD non è semplice:

richiede un numero elevato di misurazioni, facendo transitare ruote con diversi difetti, ognuno di essi di diversa entità.

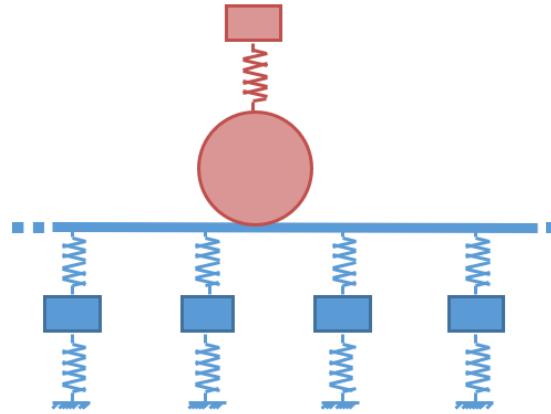
Tale problematica **può essere superata utilizzando un tool di simulazione** opportunamente validato e calibrato, che consenta di individuare il legame tra i segnali forniti dai sensori e le caratteristiche del difetto.

Pertanto nasce l'esigenza di sviluppare un modello che consente di simulare l'interazione ruota rotaia con affidabilità.



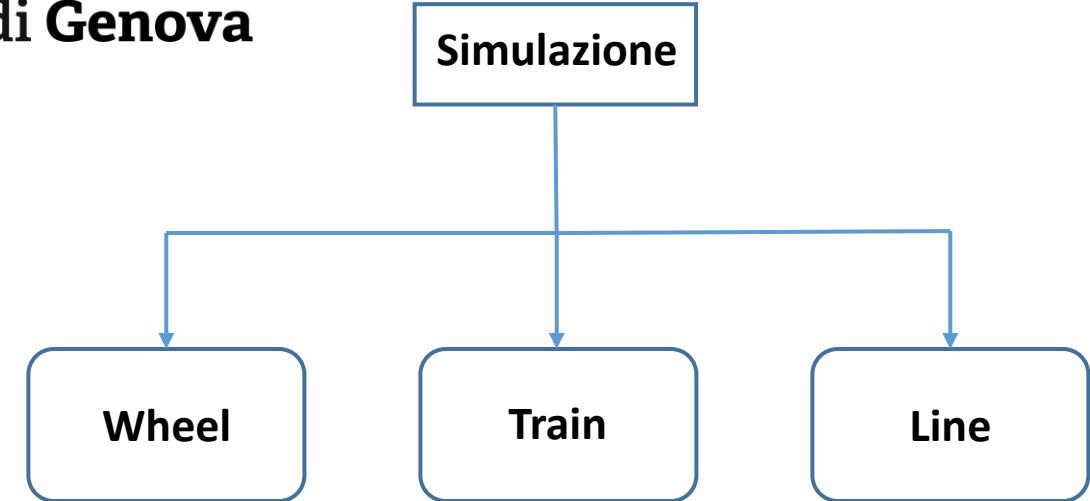
**Hitachi Group** in collaborazione con l'**Università di Genova** ha sviluppato un modello in grado di simulare l'interazione ruota-rotaia, che consente di tenere in conto anche di diverse tipologie di difetto.

Il simulatore è stato sviluppato in ambiente MATLAB a partire da tale modello e particolareggiato in modo da fornire in uscita la risposta di un sensore di deformazione posto al di sotto della rotaia su cui transita la ruota e si compone di 3 principali script: **wheel**, **train** e **line**.



Hitachi Rail STS

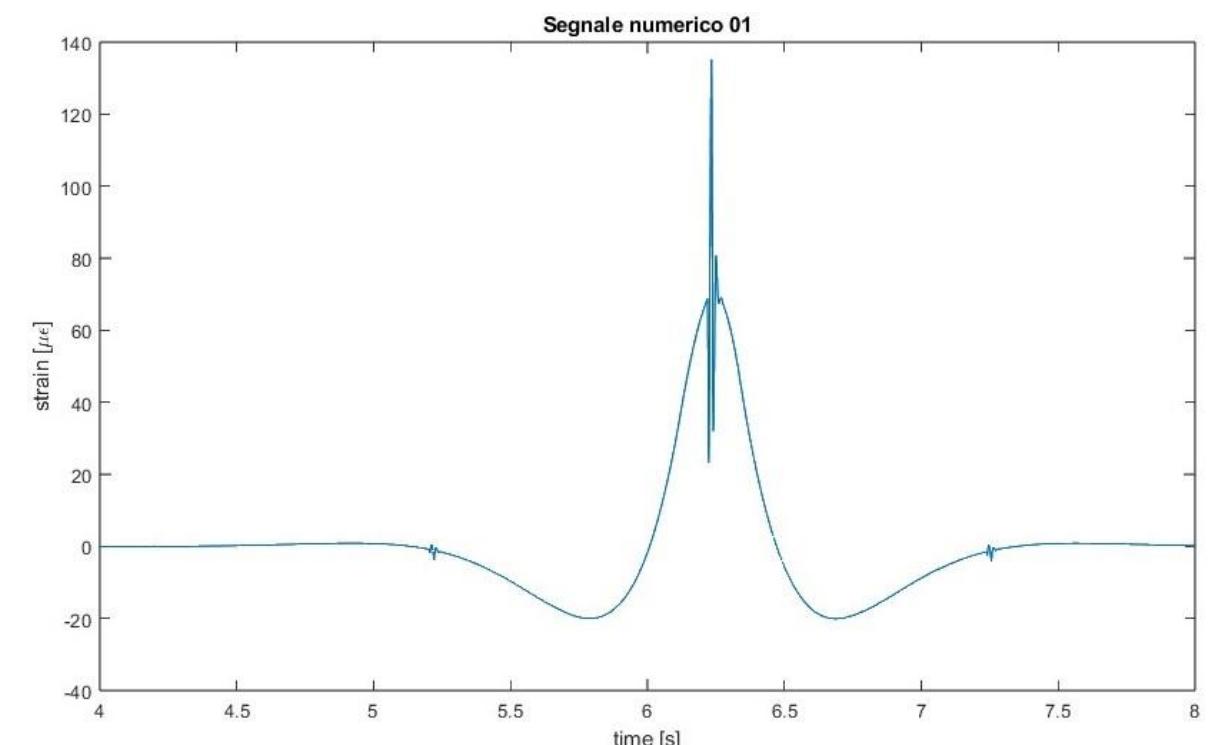
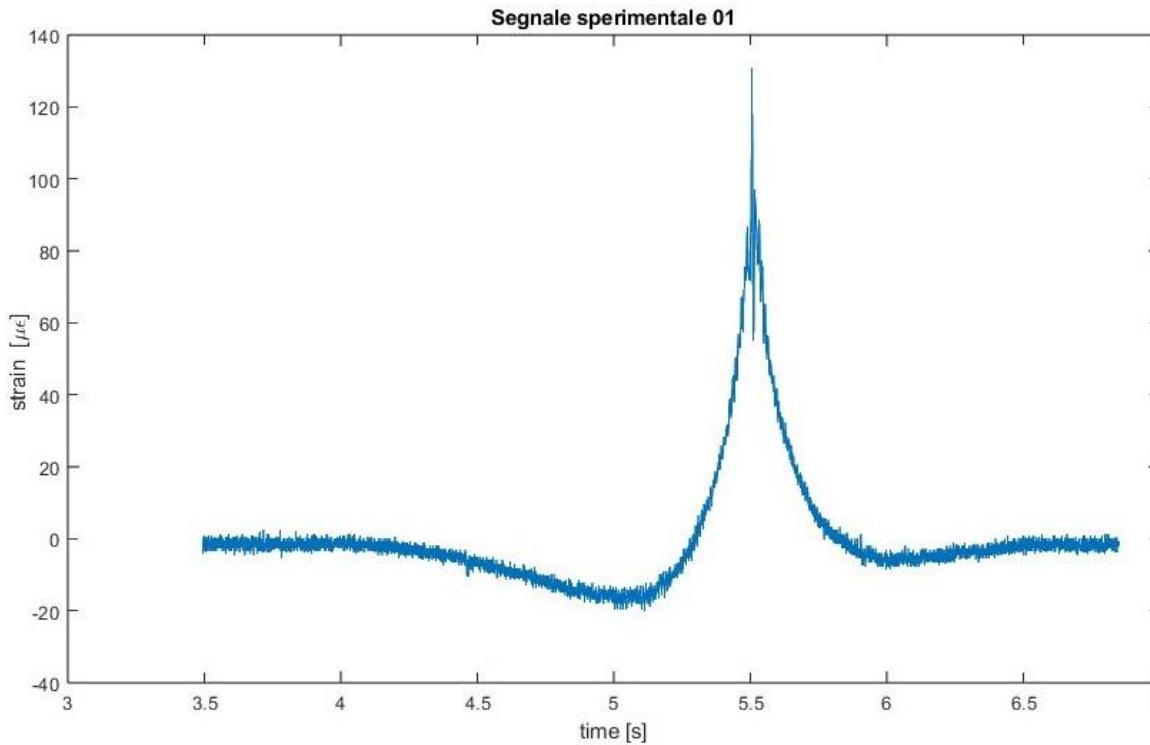
Università  
di Genova



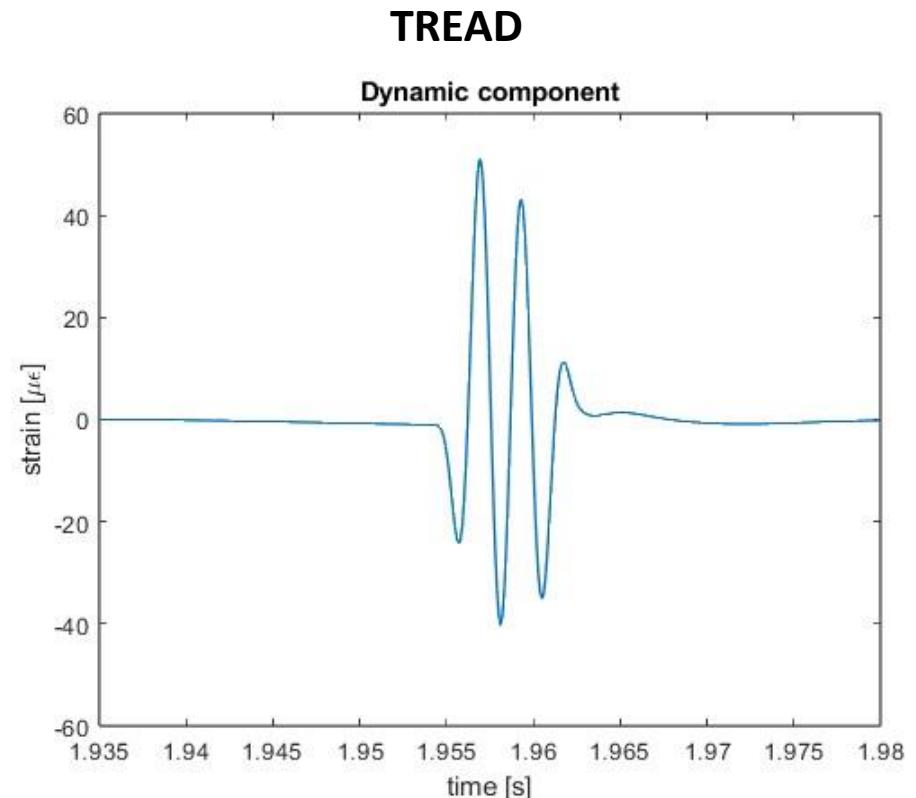
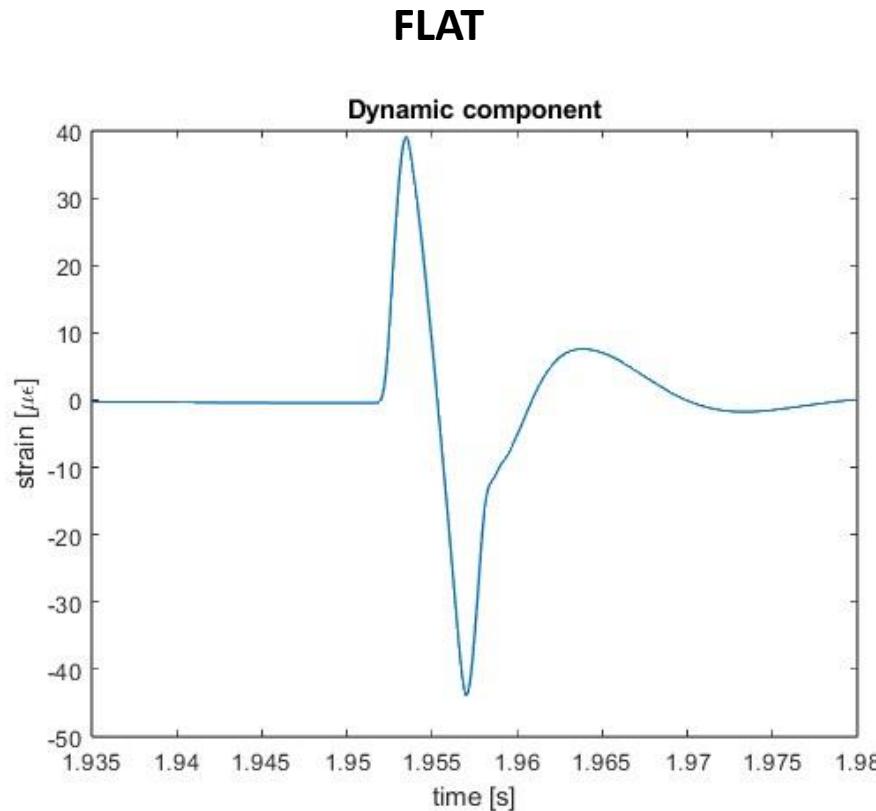
Al fine di poter effettivamente utilizzare questo modello per calibrare un sistema di classificazione del difetto è necessaria una sua validazione mediante dei dati sperimentali.

I dati sperimentali sono stati prelevati attraverso una stazione di misura installata presso un binario dello scalo merci di Marcianise:

- Ruote con difetto FLAT di diversa entità: 39mm e 76mm;
- Diverso peso: 3250 e 5750 kg;
- Diverse velocità: 10, 20, 40 km/h.



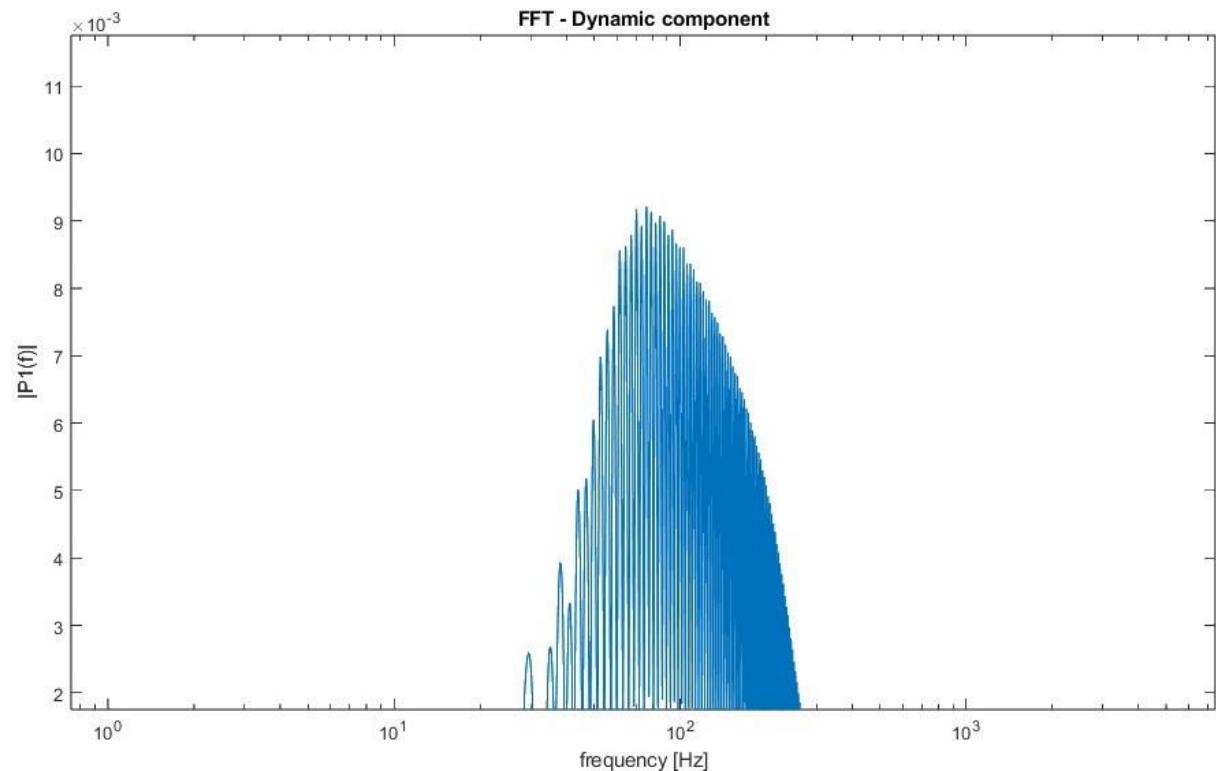
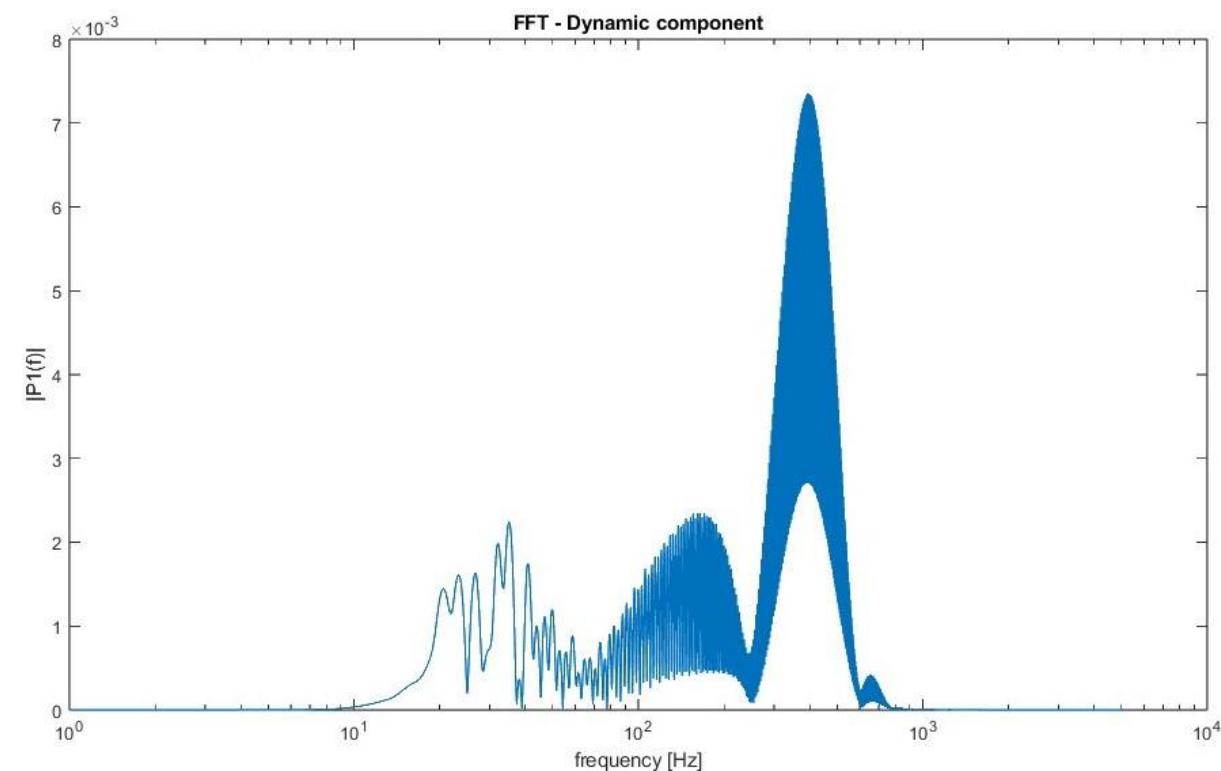
- Analisi: confronto tra le componenti dinamiche di flat e tread: dominio del tempo



$v = 30 \text{ [km/h]}$  |  $m = 5750 \text{ [kg]}$  | flat = 39 [mm] |  $\theta_0 = -0.3278 \text{ [rad]}$

$v = 30 \text{ [km/h]}$  |  $m = 5750 \text{ [kg]}$  | extent = 0.02 [m] |  $\sigma = 10^{-4} \text{ [m]}$  |  $\theta_0 = -0.3927 \text{ [rad]}$

- **Analisi 3: confronto tra le componenti dinamiche di flat e tread: dominio della frequenza**

**FLAT****TREAD**

# Sviluppi software e risultati ottenuti (2)

## 2) Estensione del simulatore di interazione ruota-rotaia per generazione data-set

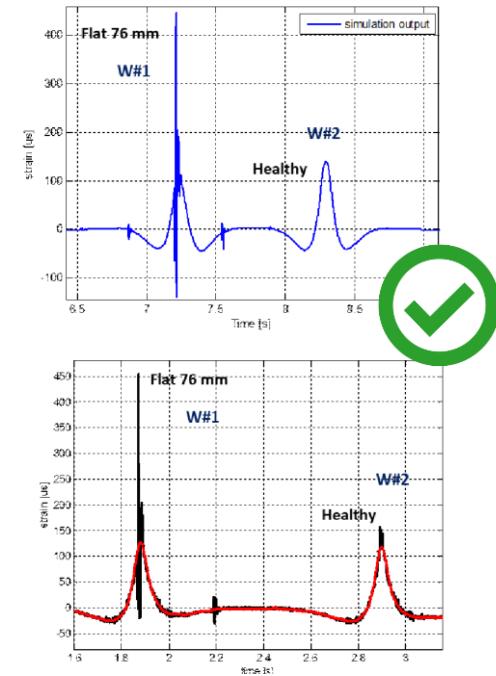
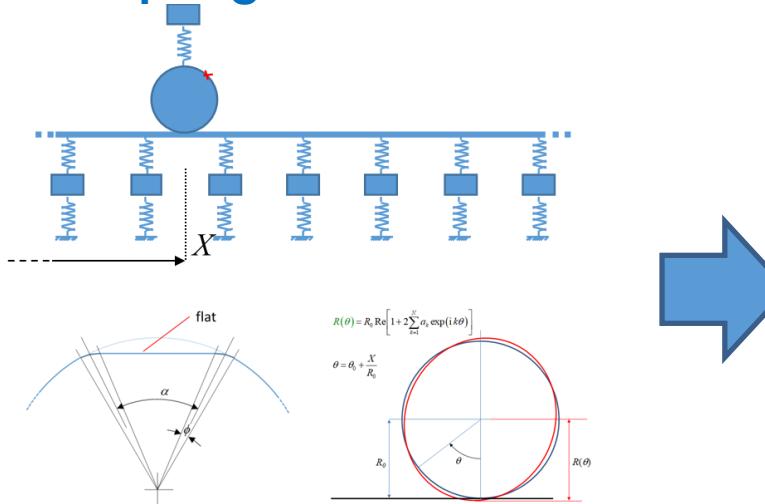
Il **simulatore** si basa sull'ipotesi di perfetto contatto ruota-rotaia (nessun distacco), e simula l'interazione ruota-rotaia in termini di risposta meccanica statica e dinamica del binario (deformazione e flessione sotto il piede del binario) al transito di ruote in movimento, permettendo di predire il comportamento dell'insieme binario e treno in transito in particolari condizioni.



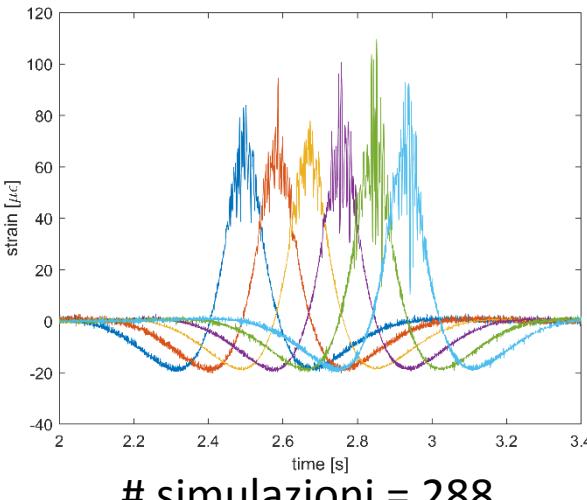
UNIVERSITÀ DEGLI STUDI  
DI GENOVA



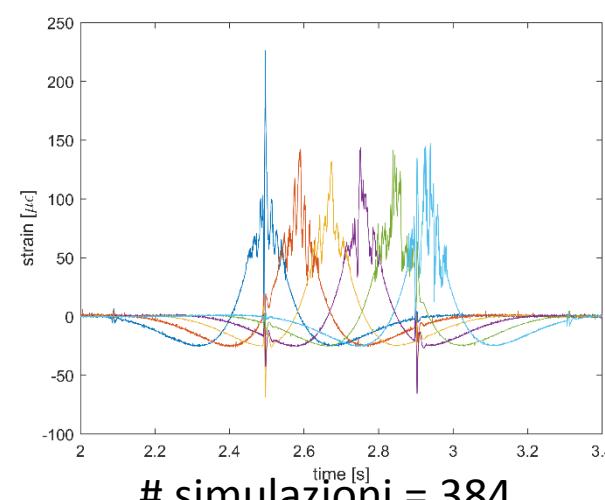
Data-set di tracciati di ruote generati con il tool esteso e «sporcati» con rumore additivo estratto da segnali sperimentali, per il training ed il test del tool di classificazione delle ruote ferroviarie (sviluppato da Intelligentia)



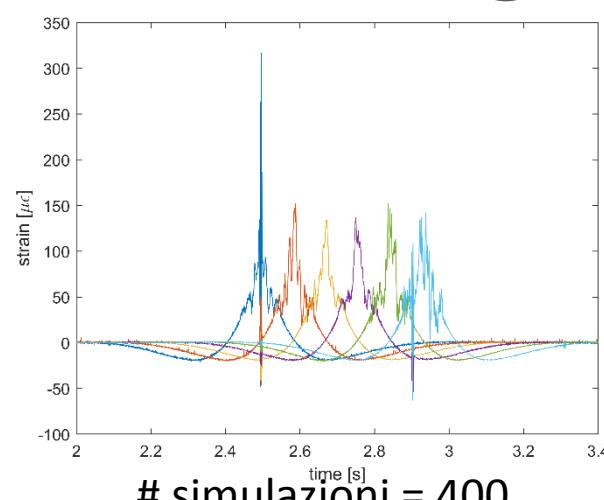
HEALTHY



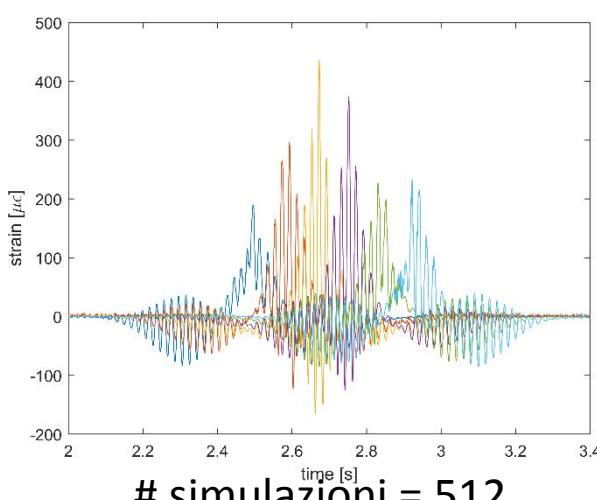
FLAT



TREAD

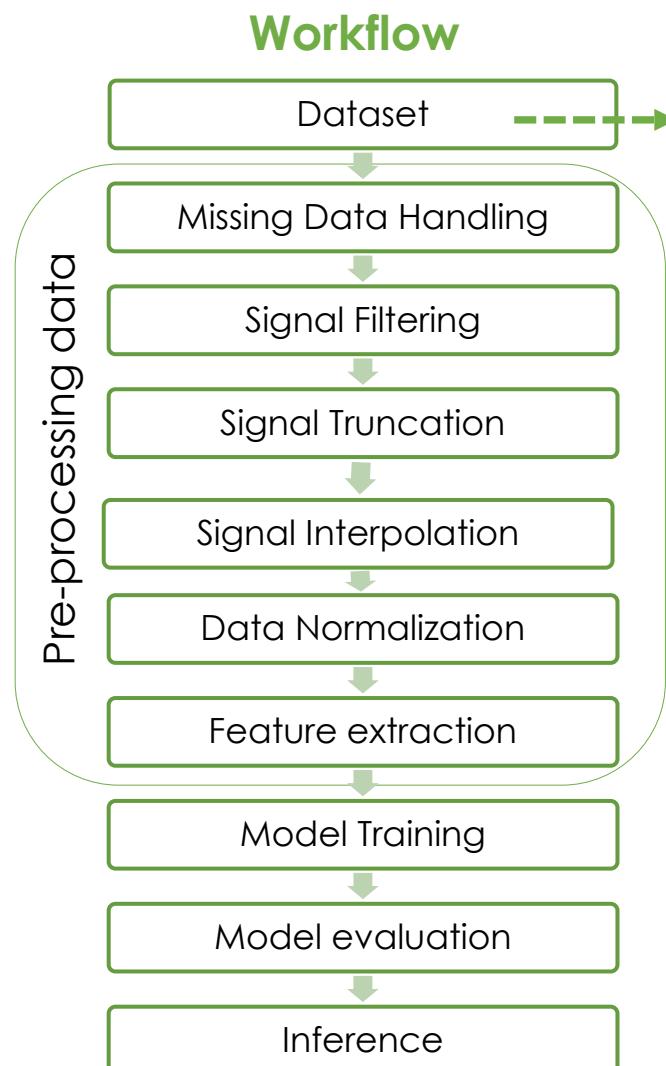


OOR



# Sviluppi software e risultati ottenuti

3) Sviluppo di un algoritmo basato su tecniche di Intelligenza Artificiale e/o Machine Learning in grado di classificare la tipologia dei difetti distinguendoli in Flat, Out Of Roundness (OOR), Treads (Spalling o Shelling).



Class	Instances
0 Healthy	288
1 Flat	384
2 Tread	400
3 OOR	512

## Metrics

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

TP = True Positive  
 TN = True Negative  
 FP = False Positive  
 FN = False Negative

## Results

### Evaluation on Test Data

**Accuracy Score: 0.987**

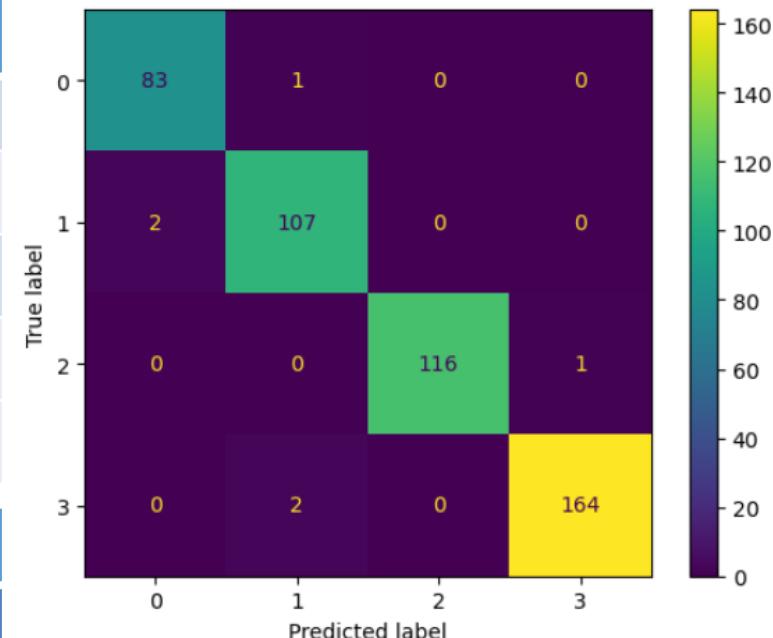
Classes	Precision	Recall	F1 score	Support
0 Healthy	0.98	0.99	0.98	84
1 Flat	0.97	0.98	0.98	109
2 Tread	1.00	0.99	1.00	117
3 OOR	0.99	0.99	0.99	166

**Pre-processing Time: 0.025 sec**

**Prediction Time: 0.0009 sec**



## Confusion matrix





# Development and In-field demonstration of an intrusion detection system using Fiber Bragg Grating (FBG) sensors

**HITACHI**  
Inspire the Next



Optoelectronics Division,  
Engineering Department,  
University of Sannio,  
Benevento, Italy



E.A.V.

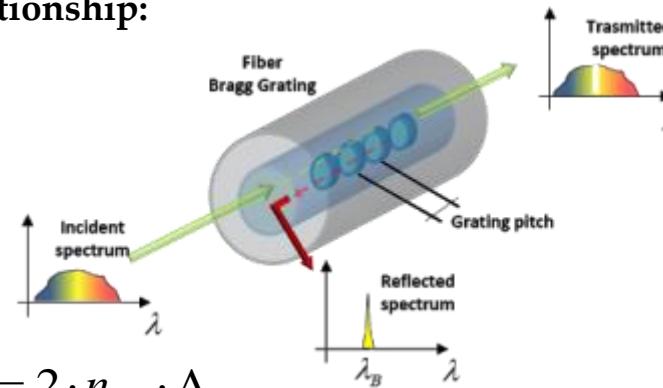


# Intrusion detection systems

- Increasing demand for improved security, safety and efficiency in the worldwide railway industry;
- Increasing need for improved the monitoring and the *protection of railway tunnels, level crossings, train depots or common scenario in the railway environment*;
- Reducing of theft, vandalism and graffiti

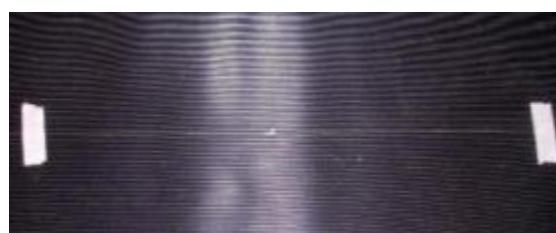


Fiber Bragg Grating (FBG) is a type of distributed Bragg reflector constructed in a short segment of optical fiber that reflects particular wavelengths of light and transmits all others. The reflected wavelength ( $\lambda_B$ ), called the Bragg wavelength, is defined by the relationship:

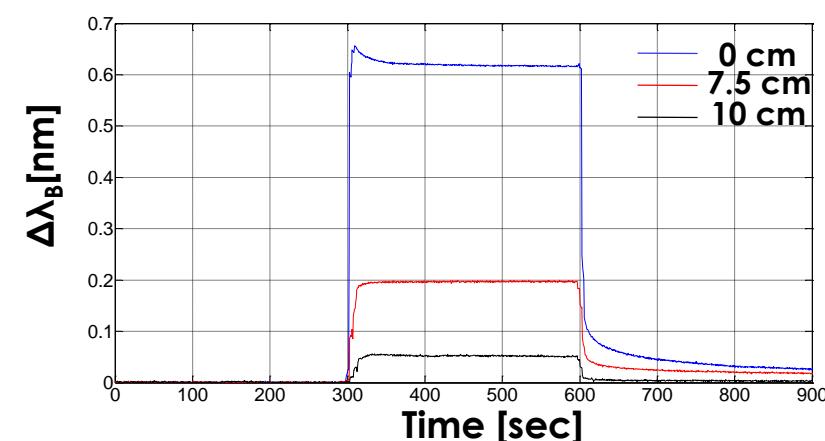
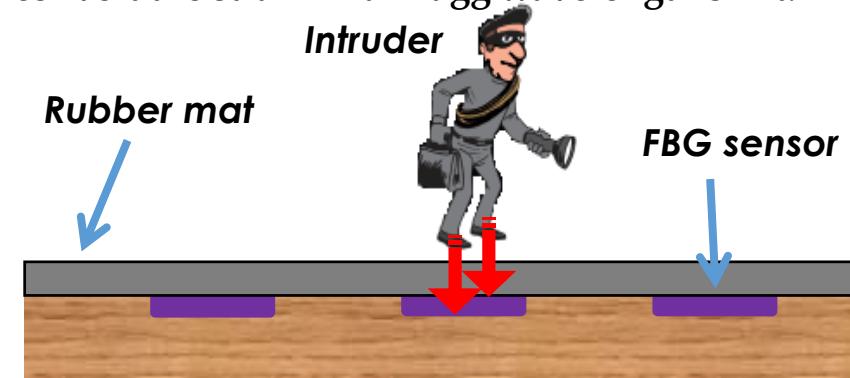


$$\lambda_B = 2 \cdot n_{eff} \cdot \Lambda$$

$n_{eff}$  is the effective refractive index of the fiber,  
 $\Lambda$  is the grating pitch  
 $\lambda_B$  is the reflected Bragg wavelength.



An intruder, walking on the mat, represents a load on the upper surface of the soft rubber mat, endowed by a low young modulus. The FBGs sense the strain associated to the load and convert the strain in a Bragg wavelength shift.



# Field Demonstration



Sensorized mat



Optical Fiber Bundle



Fiber Optical Couple Box



Interrogation unit



Server Personal PC



# Experimental results

- Experimental results reveal that Intrusion Detection System, with FBGs sensors, enable the detection of intruder's break-in.
  - Intrusion Detection System is able to detect and localize the intruder's path walking on the mat.
  - Technology Readiness Level = 6



Frame 1									
5	27	3	4	9	6	7	8	9	10
26	18	16	17	16	15	14	13	12	11
25	22	23	24	26	28	27	26	28	29
48	19	20	27	26	25	24	23	22	21
41	42	43	44	45	46	47	48	49	50
50	58	59	57	56	55	54	53	52	51
79	74	72	72	71	70	69	68	67	66
51	62	63	64	65					

Frame 2									
5	27	3	4	9	6	7	8	9	10
26	19	18	17	16	15	14	13	12	11
25	22	23	24	26	28	27	26	28	29
48	19	20	27	26	25	24	23	22	21
41	42	43	44	45	46	47	48	49	50
50	58	59	57	56	55	54	53	52	51
79	74	73	72	71	70	69	68	67	66
51	62	63	64	65					

Frame 3									
5	27	3	4	9	6	7	8	9	10
26	19	18	17	16	15	14	13	12	11
25	22	23	24	26	28	27	26	28	29
48	19	20	27	26	25	24	23	22	21
41	42	43	44	45	46	47	48	49	50
50	58	59	57	56	55	54	53	52	51
79	74	72	72	71	70	69	68	67	66
51	62	63	64	65					



1	2	3	4	5	6	7	8	9	10
19	19	19	17	19	19	14	13	12	11
29	29	29	24	25	26	27	28	26	24
49	49	49	37	36	36	34	33	32	31
49	49	49	44	45	45	47	49	49	50
59	59	59	57	59	59	54	53	53	51
59	59	59	72	72	72	69	68	67	69
69	69	69	64	68					
1	2	3	4	5	6	7	8	9	10
19	19	19	17	19	19	14	13	12	11
29	29	29	24	25	26	27	28	26	24
49	49	49	37	36	36	34	33	32	31
49	49	49	44	45	45	47	49	49	50
59	59	59	57	59	59	54	53	52	51
59	59	59	72	72	72	69	68	67	69
69	69	69	64	68					

A. Catalano, F. A. Bruno, C. Galliano, M. Pisco, G. V. Persiano, A. Cutolo, A. Cusano, "An optical fiber intrusion detection system for railway security", Sensors and Actuators A: Physical, Volume 253, 1 January 2017, Pages 91-100, ISSN 0924-4247, <http://dx.doi.org/10.1016/j.sna.2016.11.026>.

A.Catalano, F. A. Bruno, M. Pisco, A. Cutolo, A. Cusano, "An intrusion detection system for the protection of railway assets using Fiber Bragg Grating sensors", Sensors 2014, 14(10), 18268-18285, doi:10.3390/s141018268



# Photonic sensing system for aeronautic application



MAGNAGHI AERONAUTICA SpA



**Aim:** development of a novel aircraft “Weight and Balance” measurement system,  
totally based on Fiber Bragg Grating (FBG) sensors technology, to be installed on the  
aircraft landing gears. TRL 7.

# Needs of Weight and Balance measurements

- The accurate estimation of the Weight and Centre of Gravity location improves:

Safety



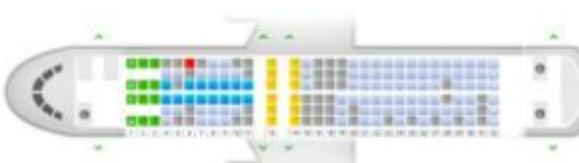
Stability



Fuel consumption & gas emission



- Currently, the estimations of the W&B consists of assigning to each passenger the same weight (hand baggage included) and placing baggage in a dedicated compartment at the after-body zone of the airplane.



✓ Unreliable

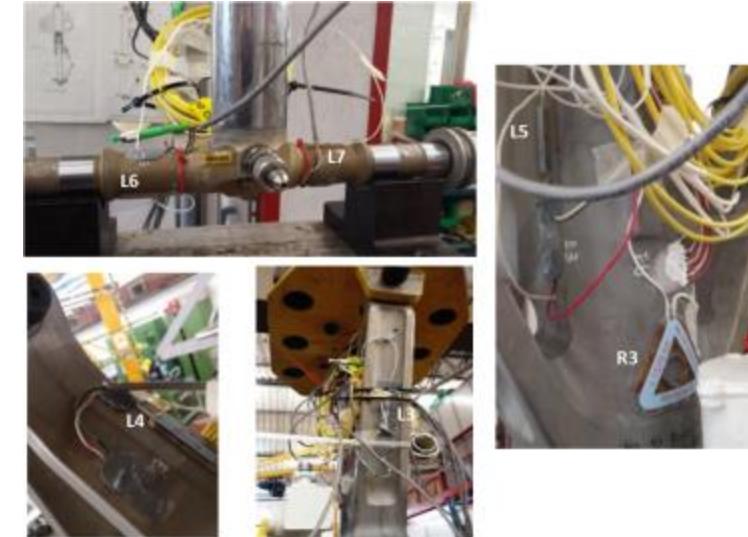
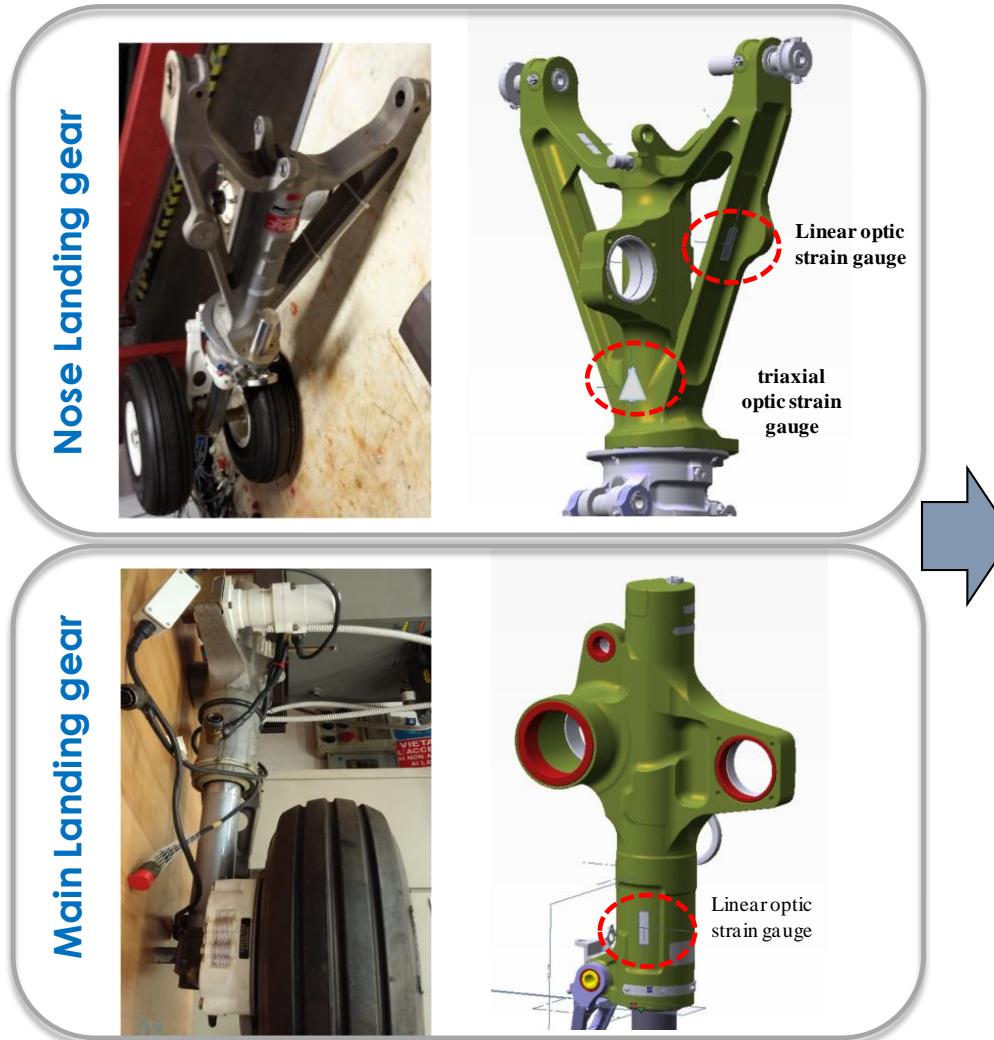
✓ Imprecise

✓ Wasteful



1. Time
2. Fuel
3. Gas Emissions

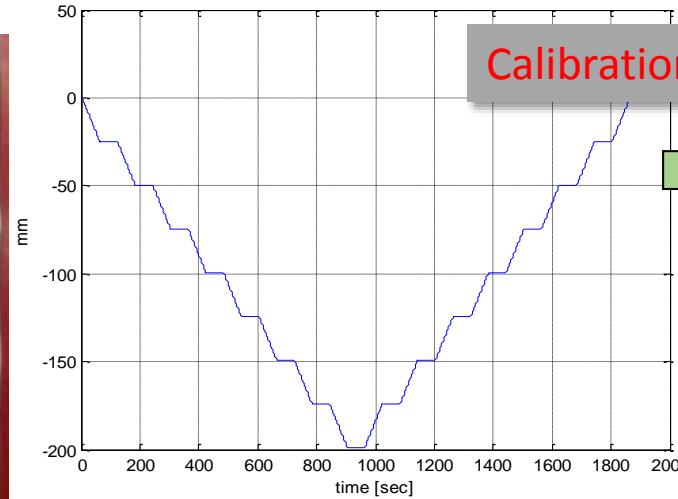
# Integration of FBG sensors network on landing gears



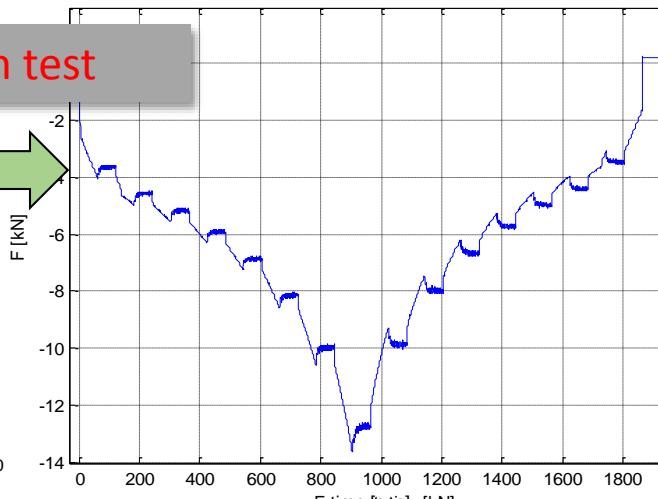
In order to compare and validate the optical technology, Magnaghi Aeronautica has installed electrical strain gauges near the optical sensors

Experimental tests in a static configuration, by varying the shock travel of the landing gear under test

### shock travel

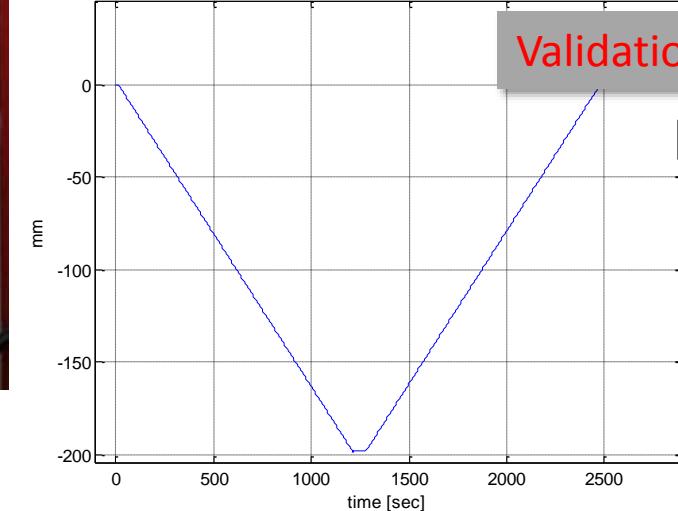


Calibration test

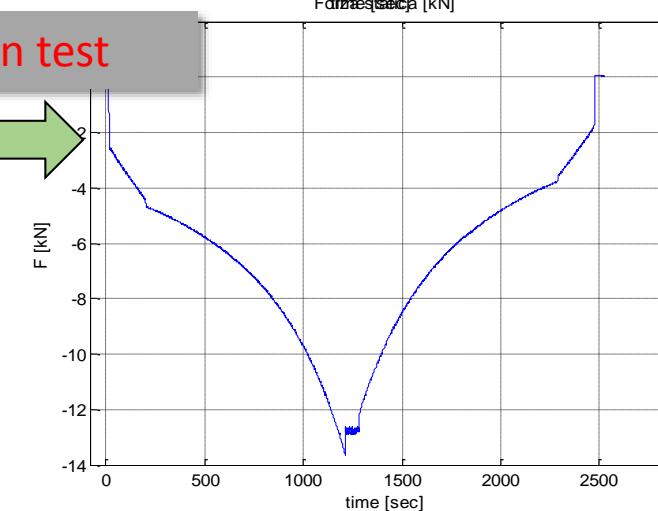


### Force applied

Forza statica [kN]



Validation test





# Ballistic Tests on Naval Panels

**FINCANTIERI**



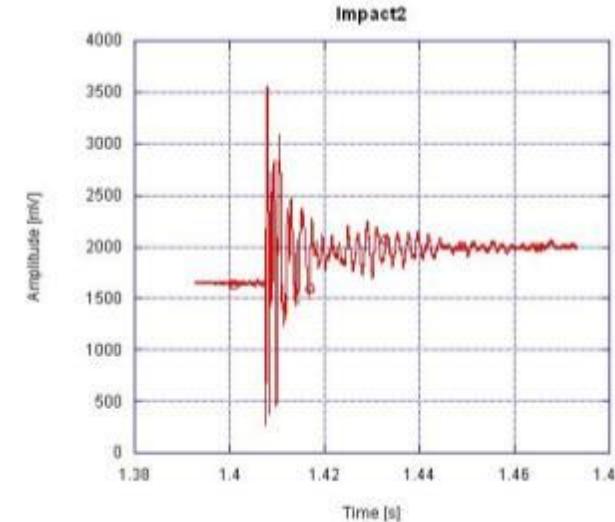
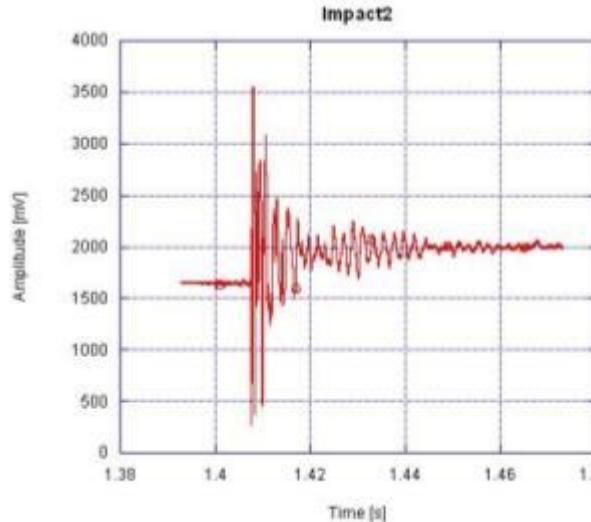
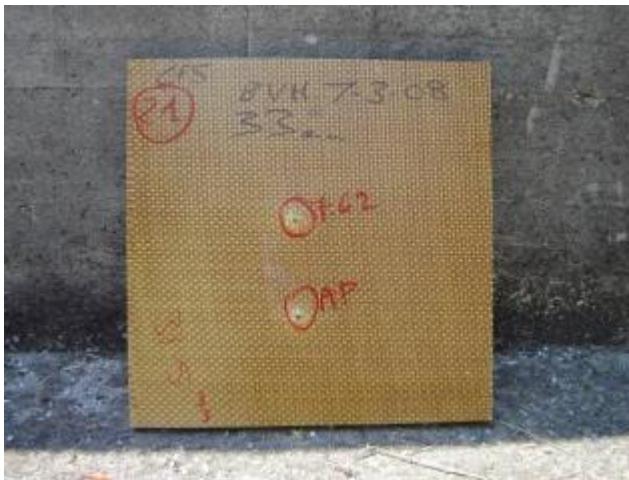
Aim: Use of FBGs sensors for monitoring of ballistic impacts on composite panels for Fincantieri

# Ballistic Tests on Naval Panels

## Ballistic Equipment



Composite Panel after shooting



Test	Panel thickness (mm)	Bullet	Speed (m/sec)	Punching
1	33 mm	7.62*51 AP	825.8	Si
2	33 mm	7.62*51 Nato	833.7	No



# Vibrational Spectroscopy on Naval Engines

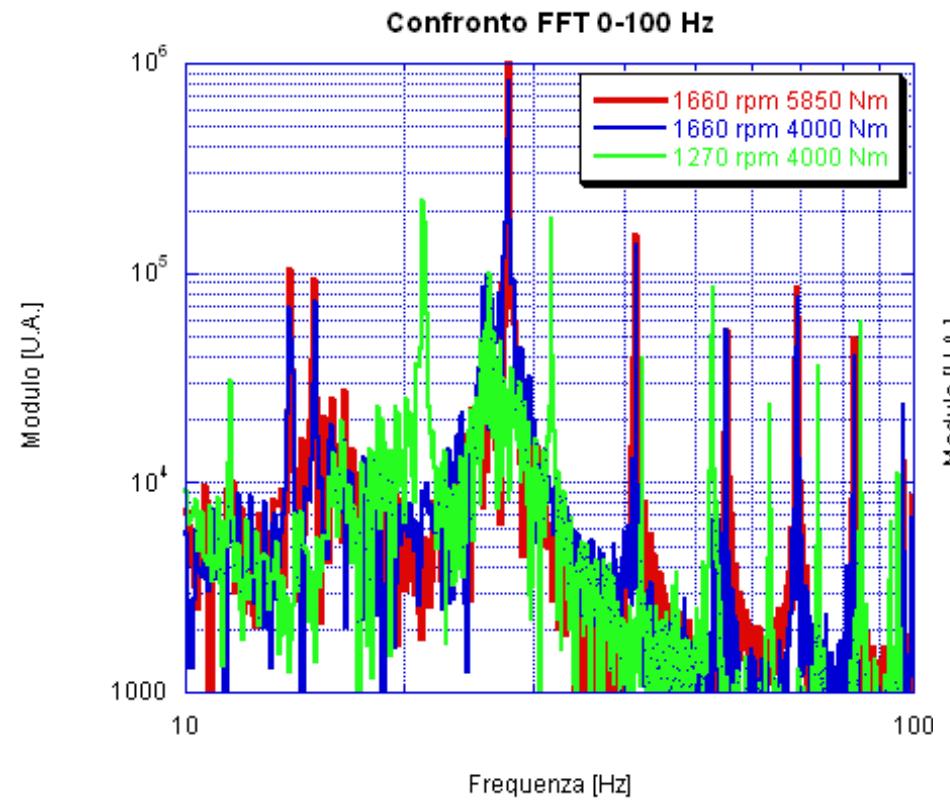


Tests conducted on  
MTU Engines used by  
Grimaldi Ferries

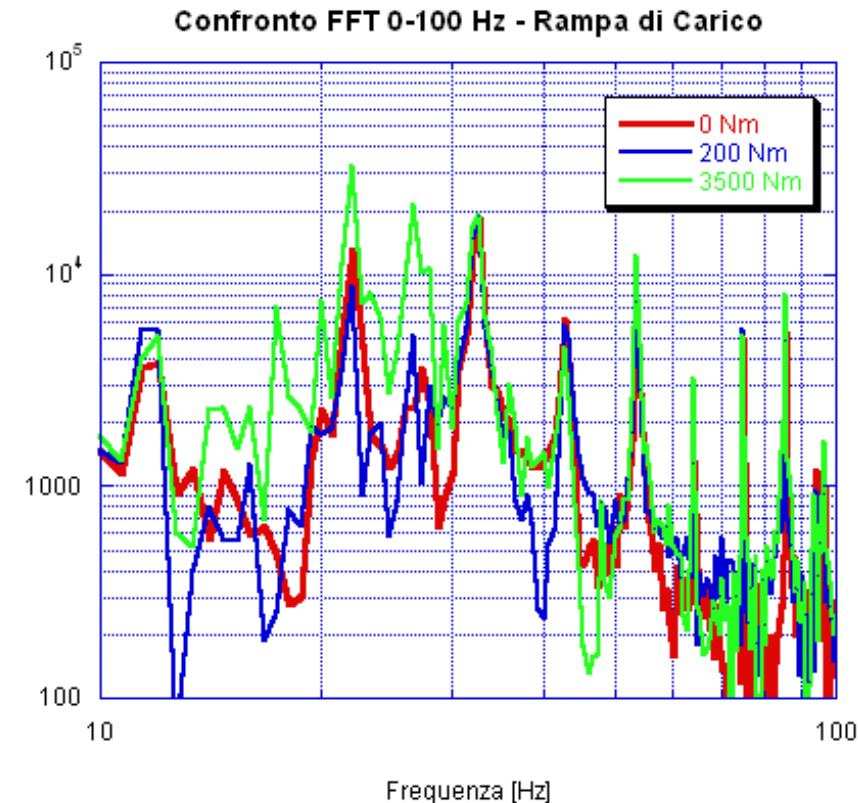


Aim: Use of FBGs sensors for vibration monitoring in marine engines for early fault detection

# Vibrational Spectroscopy on Naval Engines



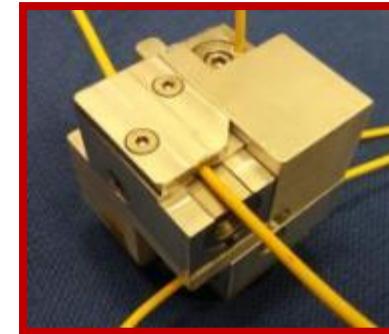
The retrieved spectra contains resonant frequencies typical of the subcomponents included in the marine engine



Operative conditions or damages of these components significantly changes the vibration spectra



# Triaxial Fiber optic Magnetic Field Sensor for MRI applications



Magnetic Resonance Imaging - MRI



OPTOELECTRONICS group  
University of Sannio

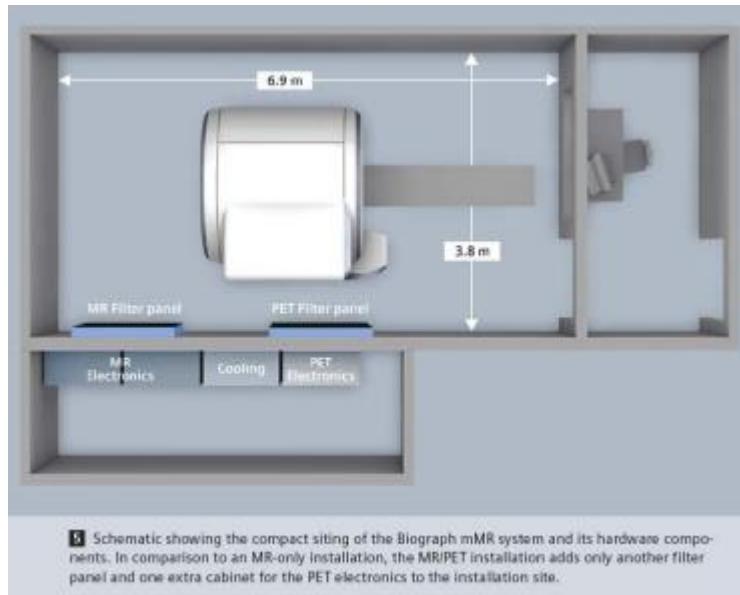


Aim: Fiber optic magnetic field sensing system for continuous real time monitoring  
in MRI diagnostic centers

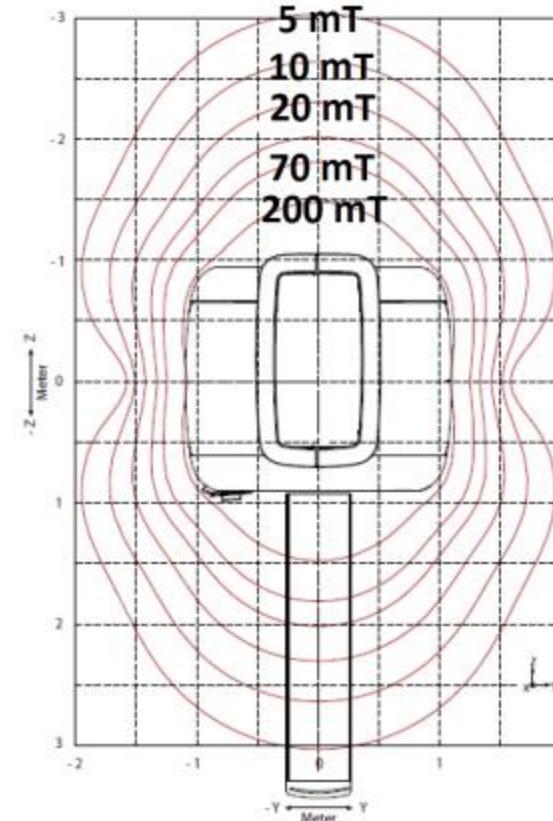
# Needs

## Quality assurance tests

**Acceptance tests at MR installations and periodic quality assurance tests in MRI facilities must be performed to monitor the static magnetic field for staff safety.**



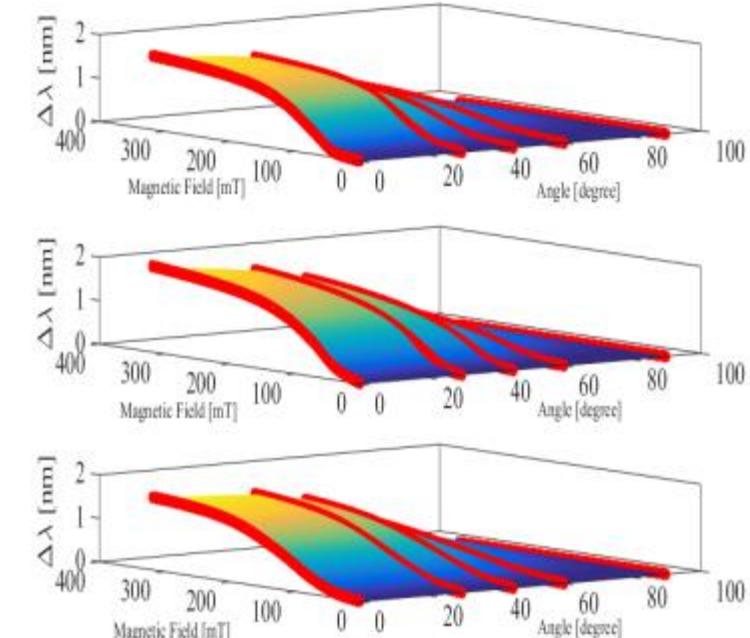
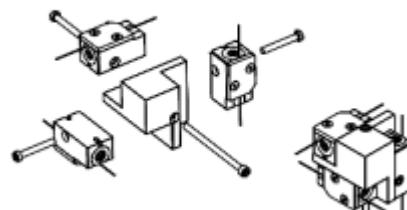
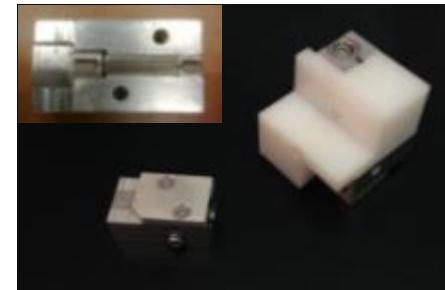
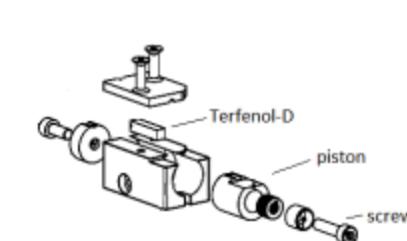
## MRI/PET: Magnetic Resonance Imaging & Positron Emission Tomography



Exposure rates lower than 1 hour/day are recommended in very high magnetic field zones (>200 mT)

# Triaxial fiber optic magnetic field sensor

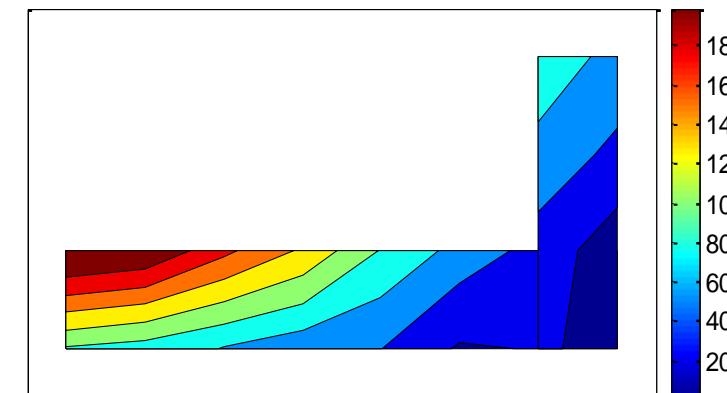
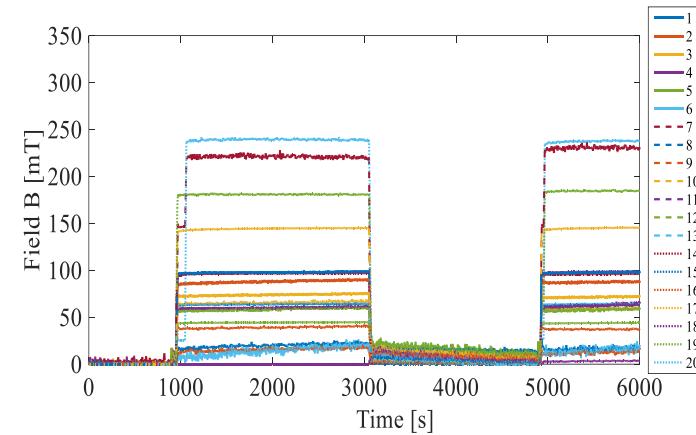
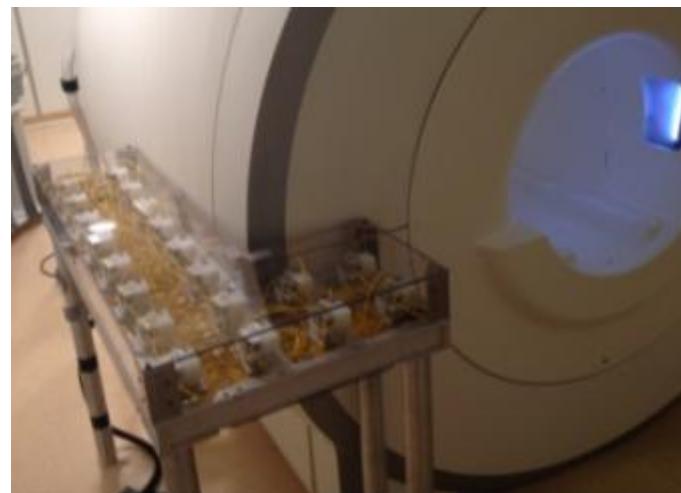
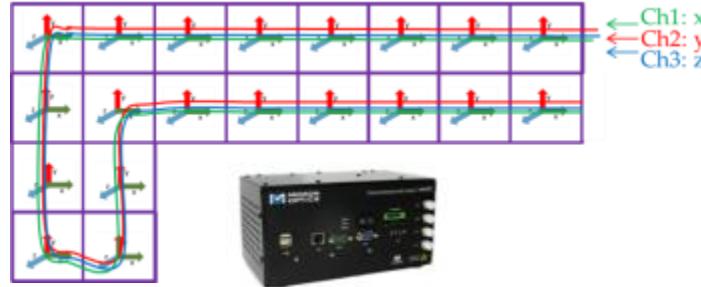
*The sensor is based on the integration of fibre optic strain sensors (fibre Bragg gratings-FBGs) with a magnetostrictive material (Terfenol-D)*



Full Characteristic of the triaxial sensor :  
«wavelength vs (angle, magnetic field)»

# Sensing system for MRI application

A sensors network has been implemented to map the magnetic field in a continuous manner and in real time in a diagnostic MRI room without interfering with the normal operations of the diagnostic centre



A patch cable brings the fiber bundle to the interrogator system located in a safe control room out from the MRI diagnostic room



# Fiber Optic Hydrophones based on Coated Fiber Bragg Gratings



A Finmeccanica Company



OPTOELECTRONIC AND SMART SYSTEMS



Aim: Design and development of High-performance opto-acoustic antennas,  
based on Fiber Bragg Grating (FBG) technology, for military, environmental and  
industrial underwater applications

### Traditional hydrophones

- Electromagnetic sensitivity
- Water seepages
- Heavy and bulky



### Fiber optic hydrophones

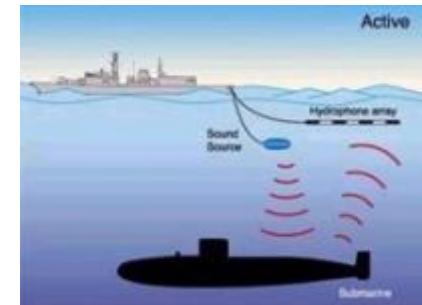
- Immunity to Electromagnetic Interferences
- Light and small
- No electric connections
- Remote monitoring
- Multiplexing capability



## Marine environmental monitoring



## Military applications



## Medical



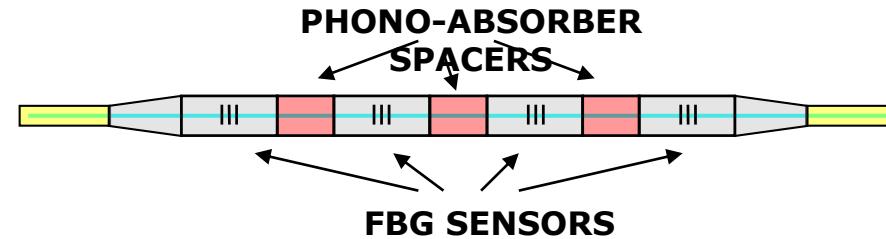
# Sensor Array: Fabrication

Flexible array :

Material : Damival

Array : 4x1

Sensors : D-5



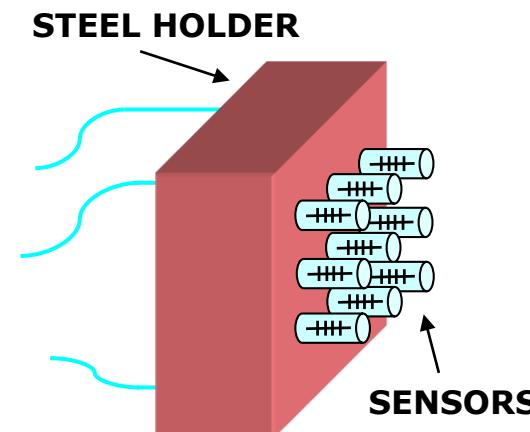
Phono-absorber spacers keep FBG sensors mechanically and acoustically separated

Rigid array :

Material : Araldite

Array : 4x4

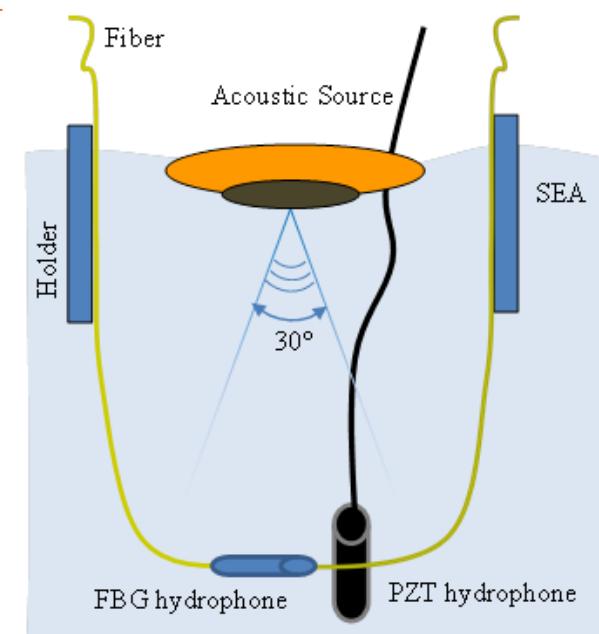
Sensors : A-5



# Offshore Preliminary Test

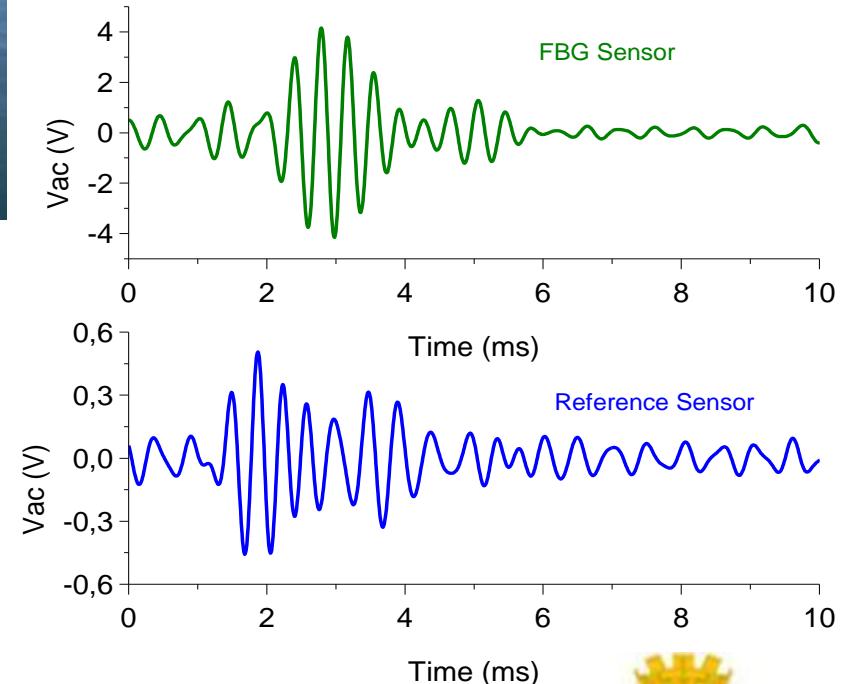


PZT



- Port of Baia, Napoli
- FBG Sensor: D-5
- Sea environment with high noise

Sensor time response



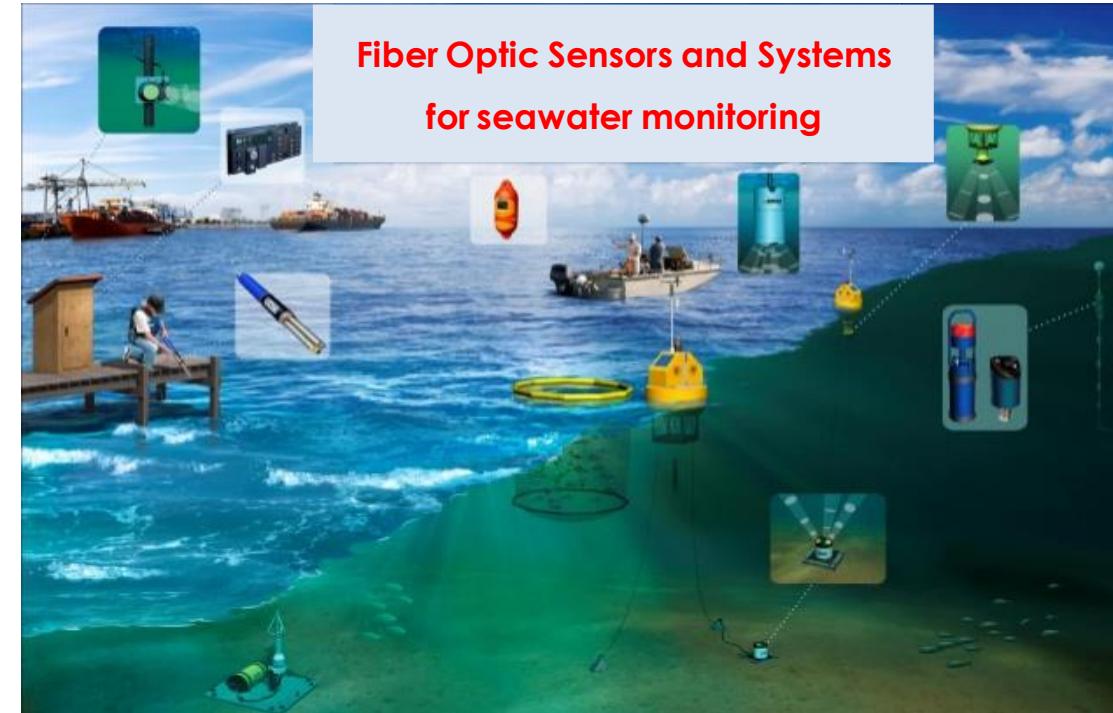
Technological Transfer to



MARINA MILITARE



# Fiber Optic Sensors and Systems for seawater monitoring

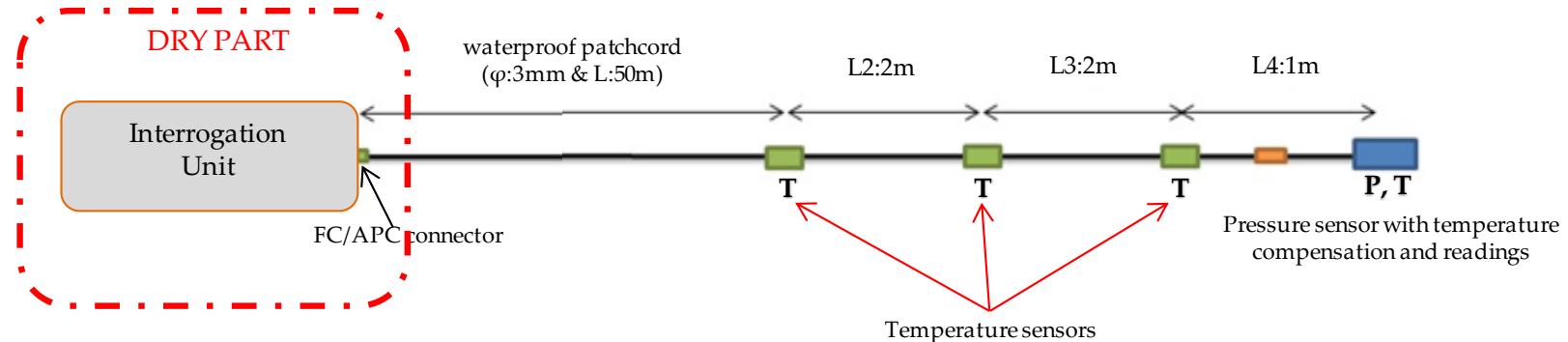
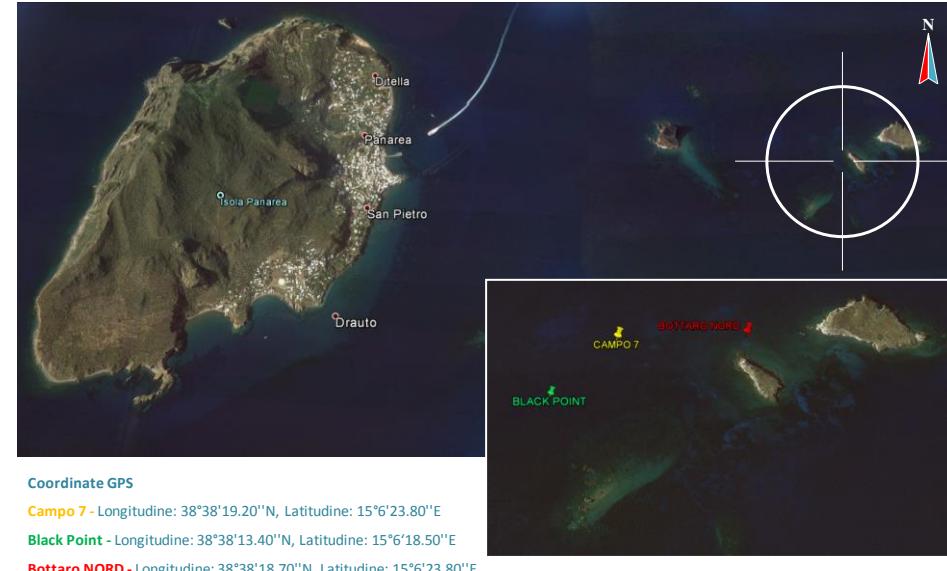


Temperature and Pressure monitoring



## Needs:

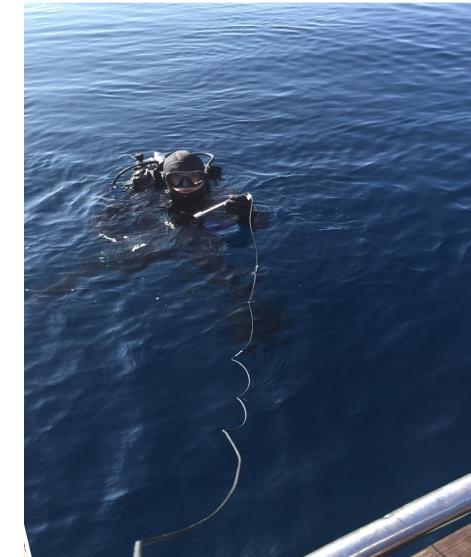
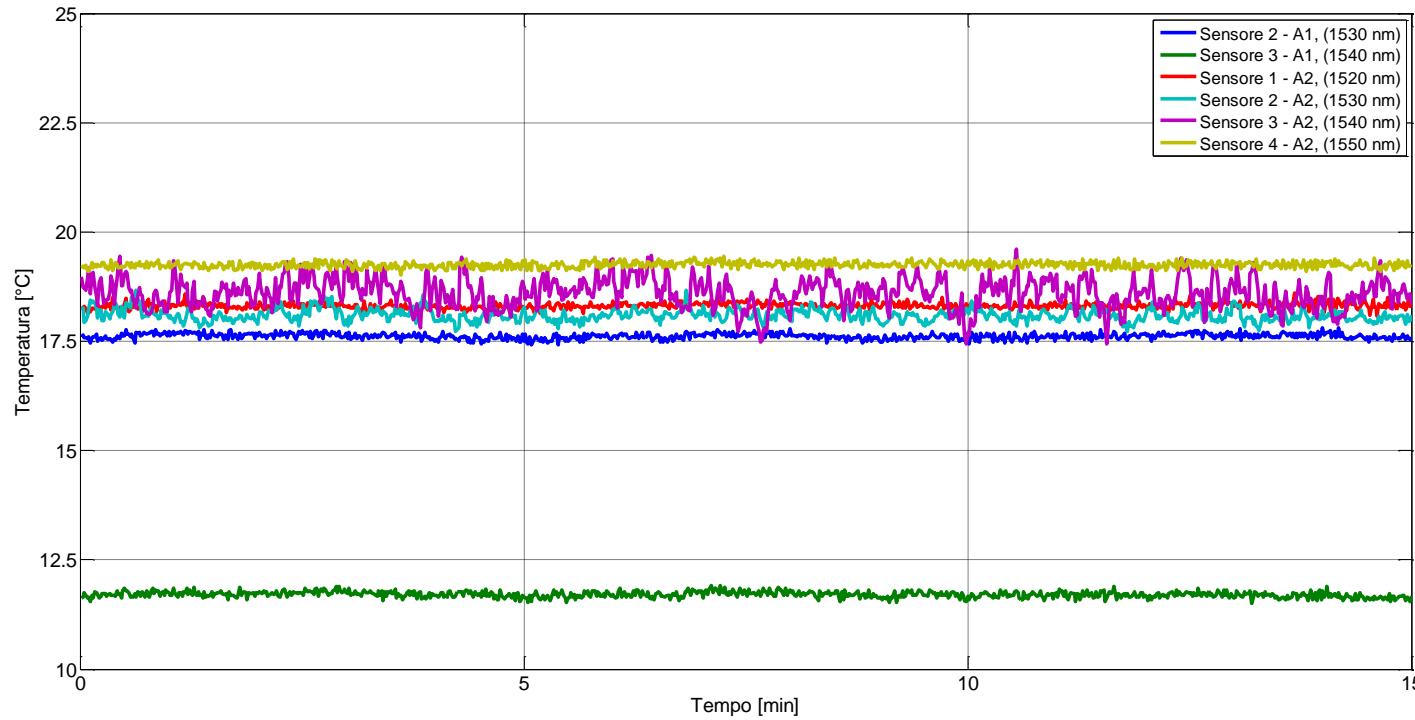
- 2 Fiber optic sensors array composed of 3 T sensors and 1 co-located P&T sensor;
- Waterproof sensors able to correctly operate underwater (down to 30m );



# Results

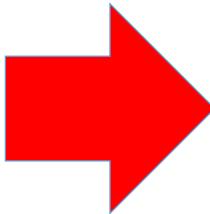


BOTTARO NORD





# Fiber Optic Sensors for High Energy Physics Applications



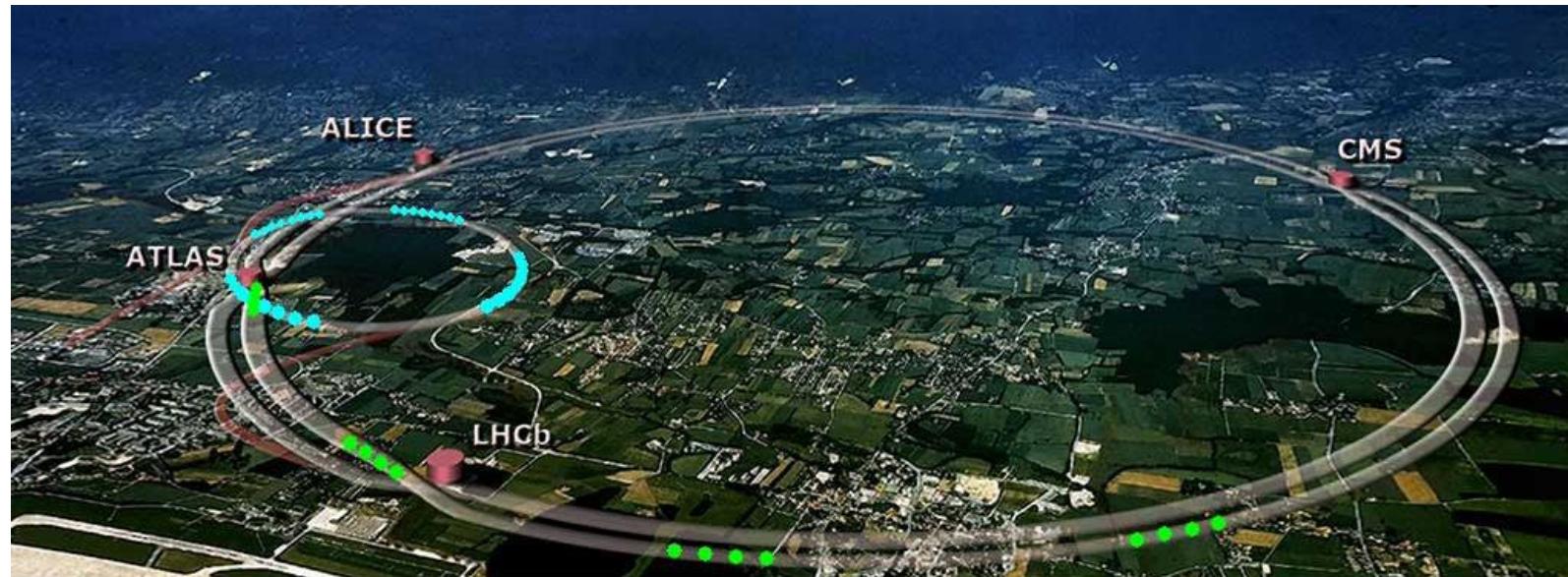
**Fiber Optic Sensor  
Community  
(academic and  
industrial partners)**



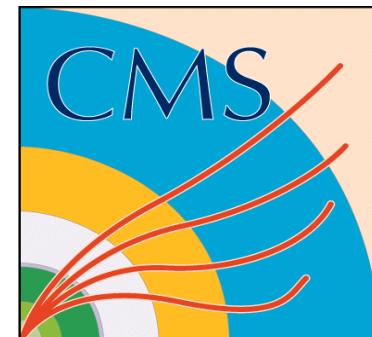
**CMS Collaboration @ CERN**

**FOS4HEP Project**

# High Energy Physics at CERN

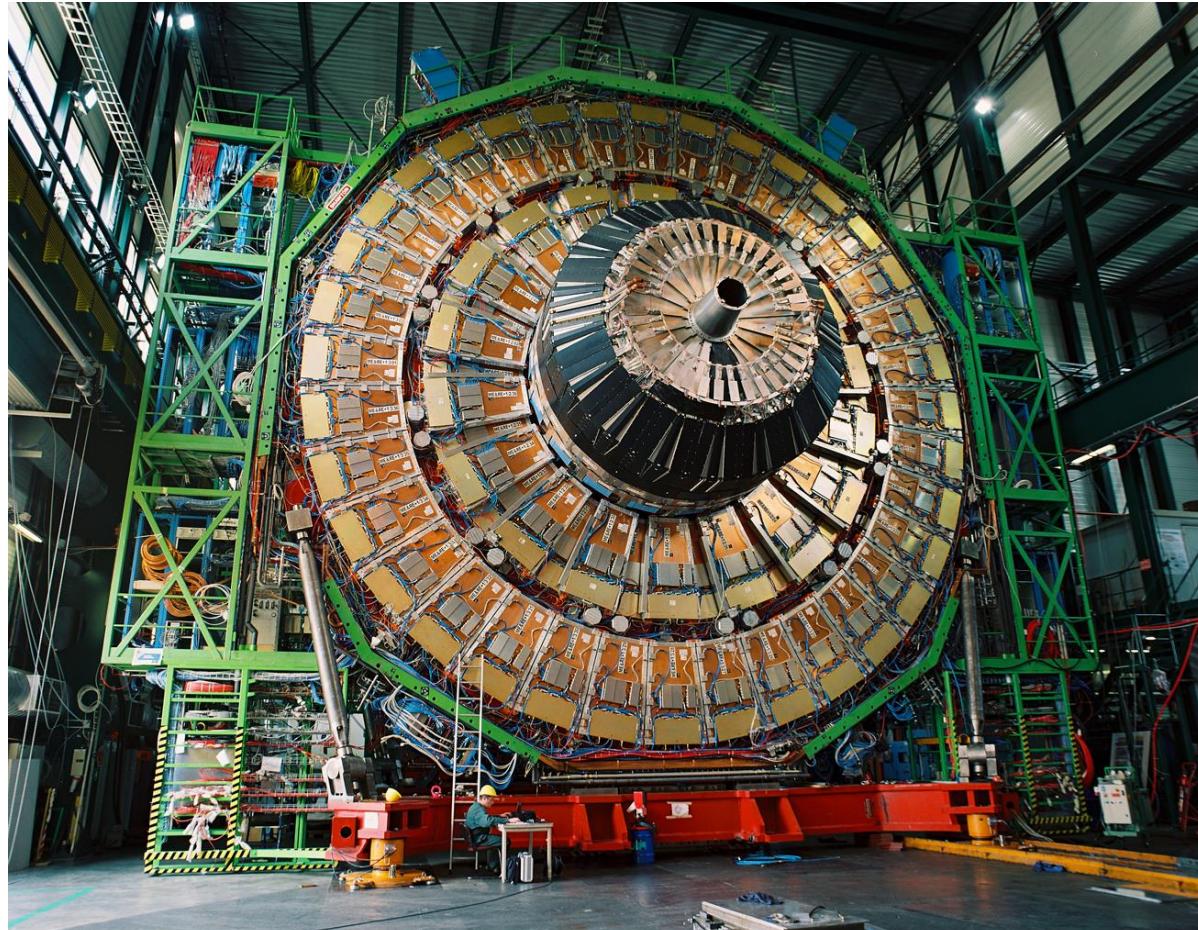


CERN (European Organization for Nuclear Research) from 1954  
Large Hadron Collider (LHC) housed in a 27 km tunnel and 100 m underground  
4 main experiments running



# CMS: Needs and Requirements

**Compact** because it is “small” for its enormous weight,  
**muon** for one of the particles it detects, and **solenoid** for the  
coil inside its huge superconducting magnet.



## Needs:

Thousands of sensors for  
the monitoring and control  
of CMS operations

(SAFETY and SECURITY are  
fundamental in these types  
of experiments operating in  
extreme conditions )

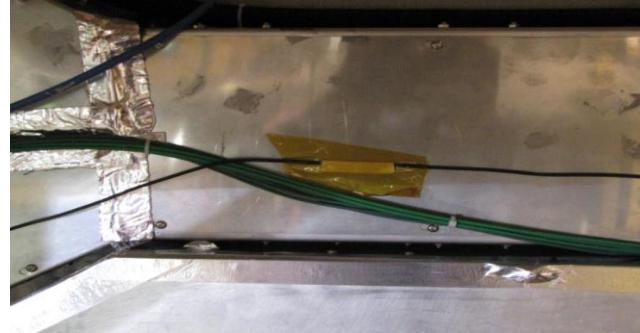
## Parameters to be detected:

Temperature, Humidity,  
pressure, magnetic induced  
strain, displacements  
magnetic field, gases, cryo  
sensors, and many others

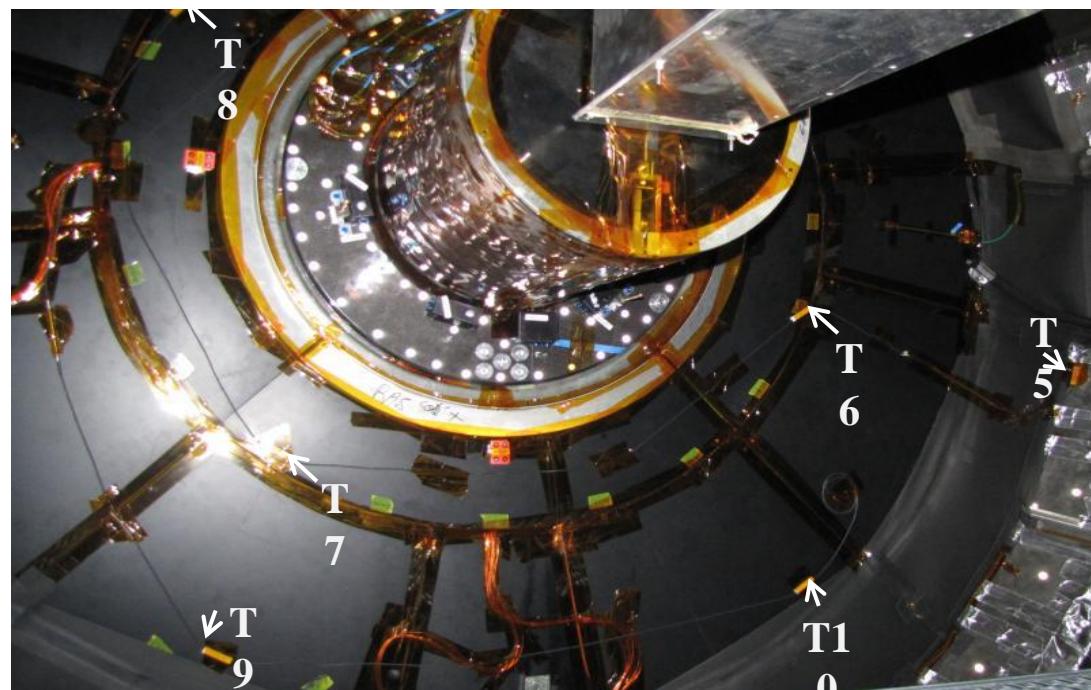
## Requirements:

Reduced cabling  
Radiation Hardness  
Electromagnetic immunity

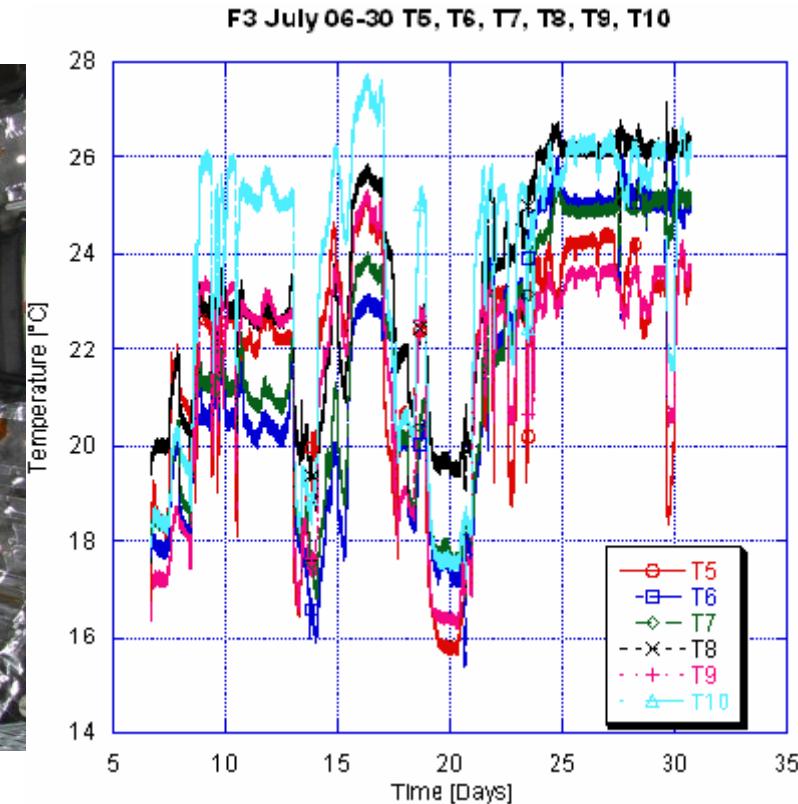
# Fiber Optic Temperature Sensors at CMS



Ceramic packaging for radiation hardness  
Coating absence on FBGs to avoid radiation induced cracking and aging



A. Saccomanno, A. Cusano et al, IEEE Sensors Journal, 12(12), 3392- 3398 (2012)

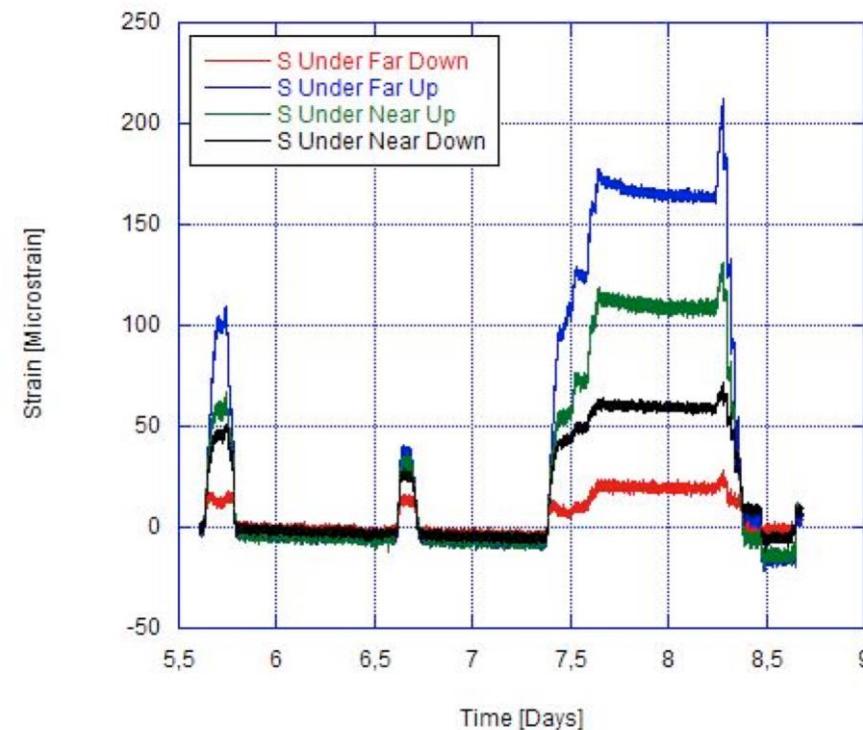


# Strain Monitoring at CMS



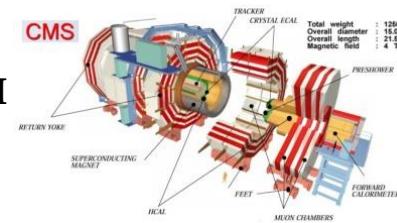
High B field causes strain within iron structures leading to possible structure re-shaping

This effect is very dangerous since it can cause the damaging of the beam pipe leading to catastrophic explosions



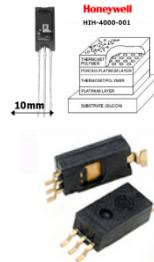
# Humidity Sensors Issues in HEP Trackers

- ✓ Low mass and Small dimensions
- ✓ Insensitivity to magnetic field
- ✓ Operation at temperature down to -40°C and response to the full range [0, 100]% RH
- ✓ Reduced number of wires
- ✓ High long term stability and/or possibility of an easy remote recalibration
- ✓ Radiation resistance to dose up to 1 Mgy



## Miniaturized humidity sensors used at CERN

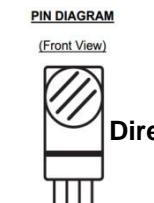
### HIH 4000 series by Honeywell



Honeywell  
HIH-4000-001  
Capacitive  
Small Size  
Inexpensive  
3 wires for each RH measuring point  
Accuracy of 3,5%RH  
Response time 15s  
Minimum operation temperature -40°C  
NOT Radiation hard



### Precon HS2000



PIN DIAGRAM  
(Front View)  
Capacitive  
Small Size  
Easy in use  
Direct reading of Dew Point  
NOT radiation hard  
Sensitive to B fields

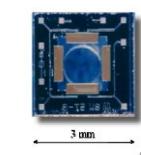
Pin # 1 2 3 4

Pin 1	Temperature Out (0 to Vsupply)
Pin 2	Power (2 to 5.5 volt)
Pin 3	RH Out (0 to Vsupply)
Pin 4	Ground



### Hygrometrix HMX

Resistive  
Small size  
Tolerant to radiations  
Extremely low response  
Complex chain of read-out  
over long distances  
Highly not linear signal  
NOT AVAILABLE on the market any more



For HEP detectors, multi-point distributed measurements and insensitivity to electromagnetic noise and resistance to ionizing radiations are a MUST.

NOWADAYS THERE IS NO MINIATURIZED HUMIDITY SENSOR ON THE MARKET  
WELL SUITED FOR HEP DETECTOR APPLICATIONS

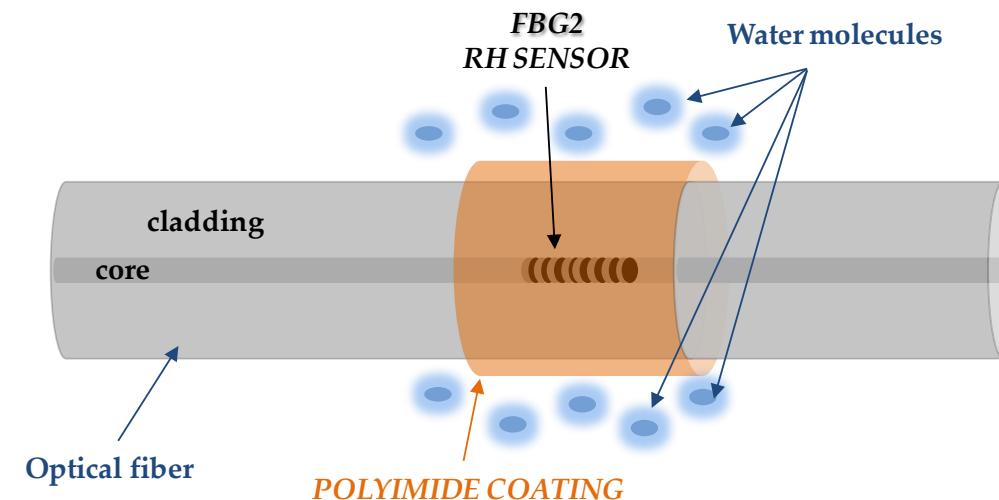
# FBG as Humidity Sensor

- Bare FBGs are insensitive to humidity.
- Use of a sensitive material as coating of the FBG to induce a mechanical effect.
- Hygroscopic polymers swell upon adsorption of water molecules.



## Development of humidity sensor by coating the FBG with a polyimide film.

Water molecules absorbed by the hygroscopic coating → Coating expansion ("Swelling") → Strain induced on the FBG → Bragg wavelength shift ( $\Delta\lambda_B$ )

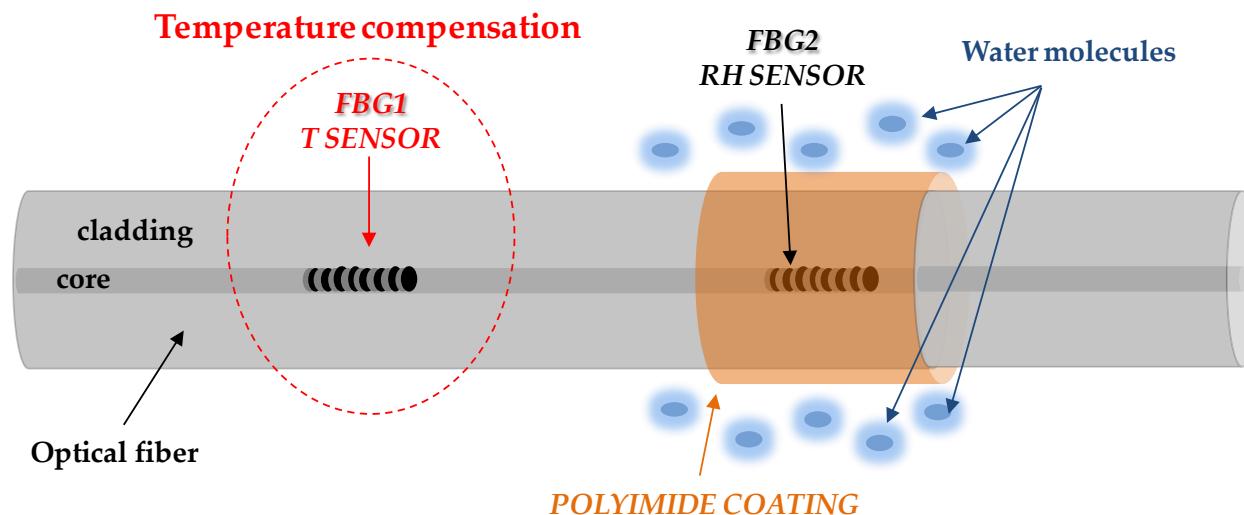


# Temperature compensation

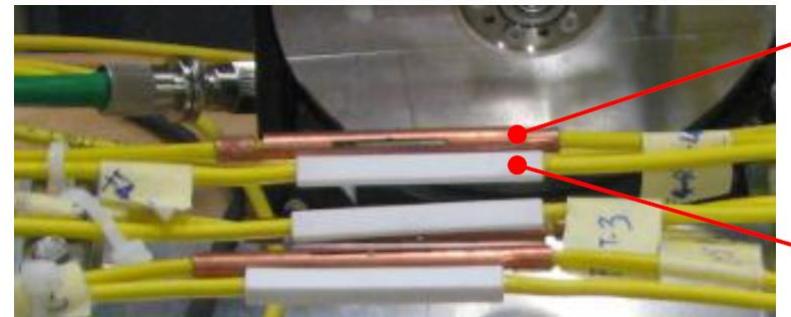
Polyimide-coated FBGs are also sensitive to temperature:

$$\Delta\lambda_B = f(\Delta T, \Delta RH) = S_T(T, RH) \cdot \Delta T + S_{RH}(T, RH) \cdot \Delta RH$$

- A temperature compensation scheme is required to extract RH measurements from the sensor readings, that can be simply accomplished by using an uncoated FBG (on the same fiber)



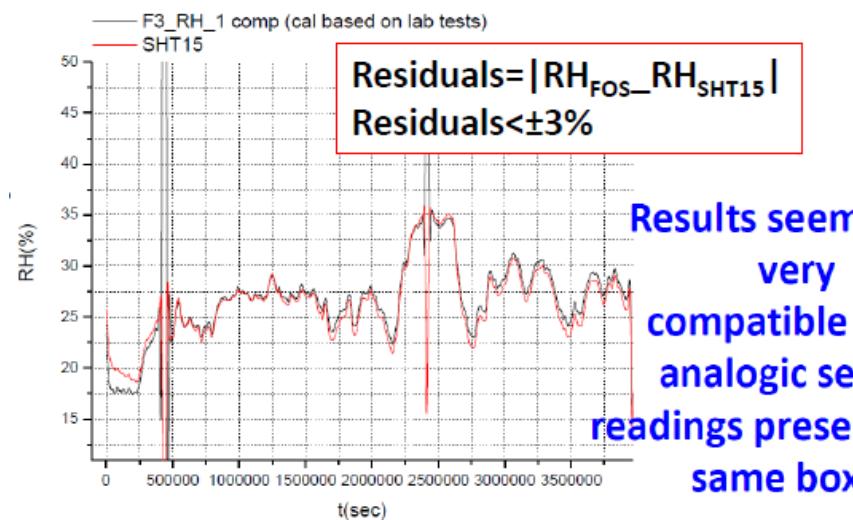
## R&D PROGRAMME@CERN



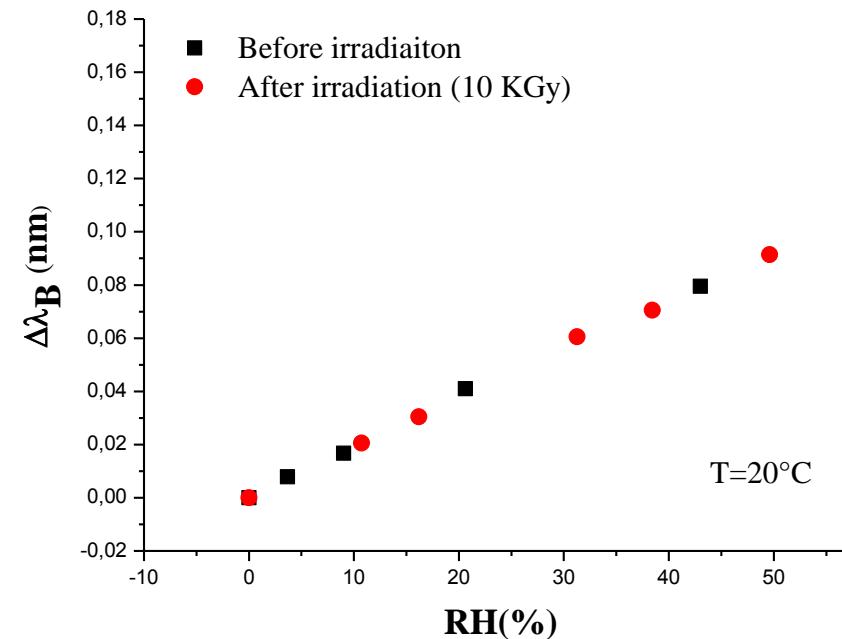
RH-FOS SENSOR

T-FOS SENSOR

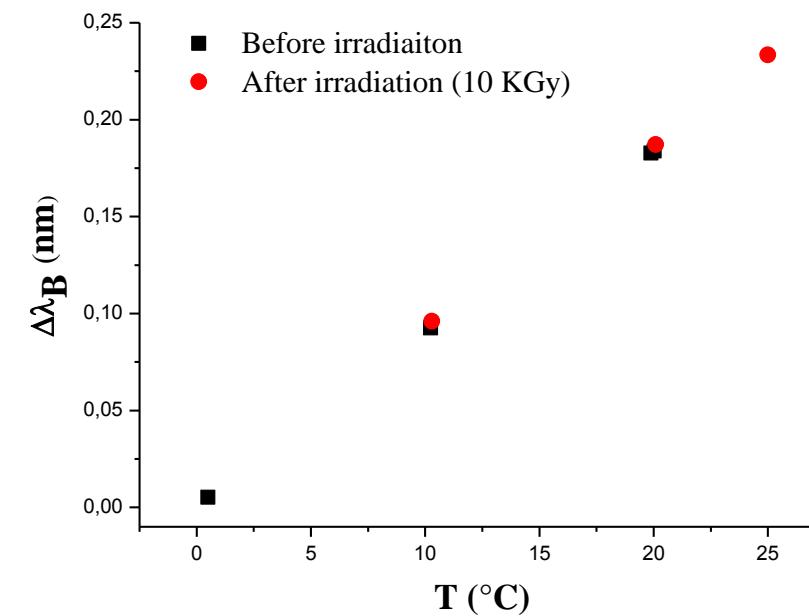
- Arrays pre-irradiated at 10 kGy ionizing irradiation dose
- FBG conveniently protected in a special casing
- Calibration of the arrays performed in the climatic chamber using a special cylindrical support
- After the calibration, arrays installed in CMS as first application of this technology to the “real world”



## Pre and Post irradiation characteristic curves (after a 10kGy dose)

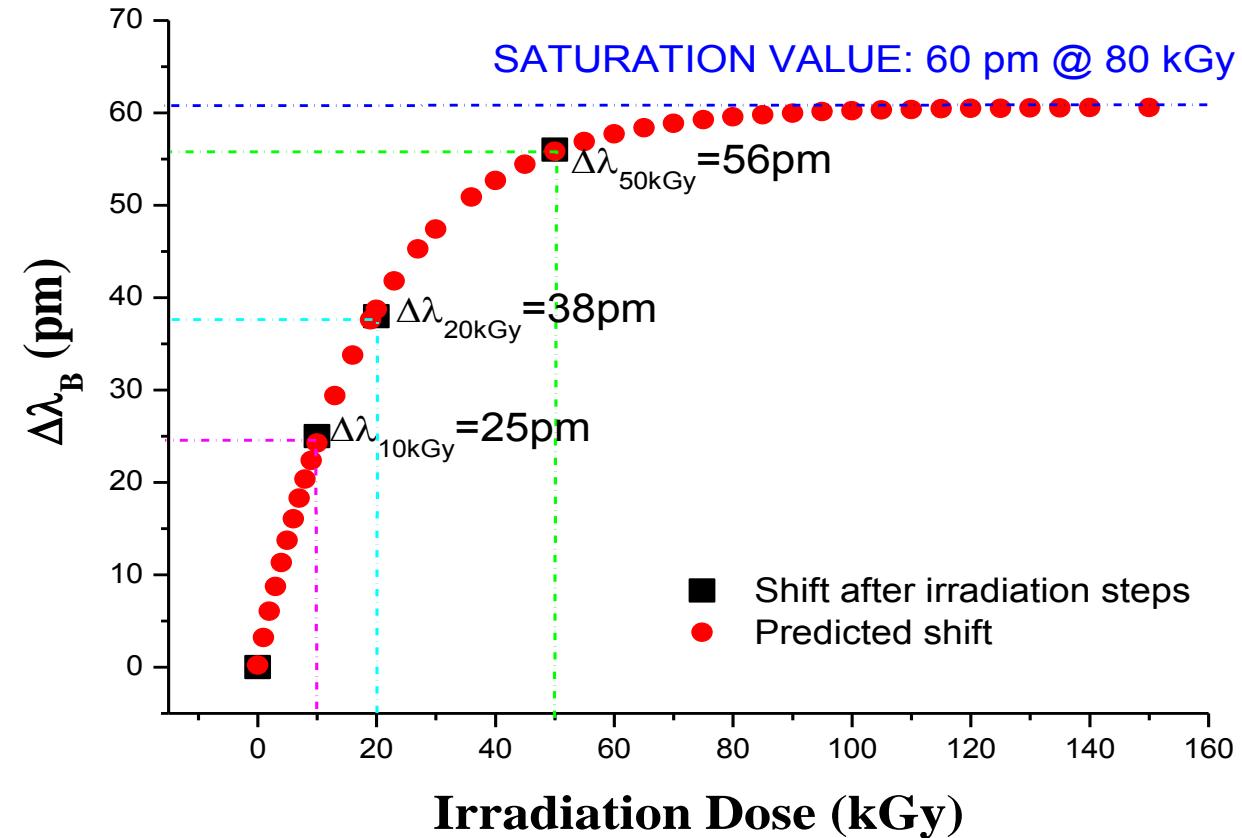


$\Delta\lambda_B$  vs RH before and after the irradiation process



$\Delta\lambda_B$  vs T before and after the irradiation process

## $\Delta\lambda_B$ vs Irradiation dose



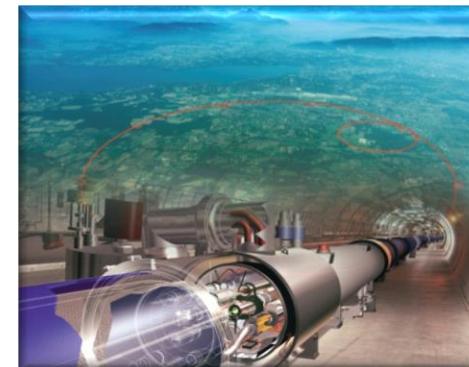
Commercial electronic hygrometer resulted definitely damaged after the 10 kGy dose.

# Cryo-Sensors for superconducting magnets

## Needs and requirements

A large number of sensors and electronic devices located inside the 27- km LHC tunnel able to withstand the **radiation** and **cryogenic** environment:

**Cryogenic Thermometers**  
to monitor and to control the temperature in the wide range 1.9 - 300 K



**Electrical Strain Gauges**  
to monitor and to control the stress profile from the assembly to the operation of the magnet

**A lot of wires needed!**

### *Potentials for the use of fibers for temperature and strain*

- Immunity to electromagnetic interference in high field environment
- reduction of instrumentation (wires, connections, read out system )
- low heat in-leak into the bath
- improvement of workplace safety in the upgraded tunnel by monitoring the temperature over long length (Wavelength Division Multiplexing)

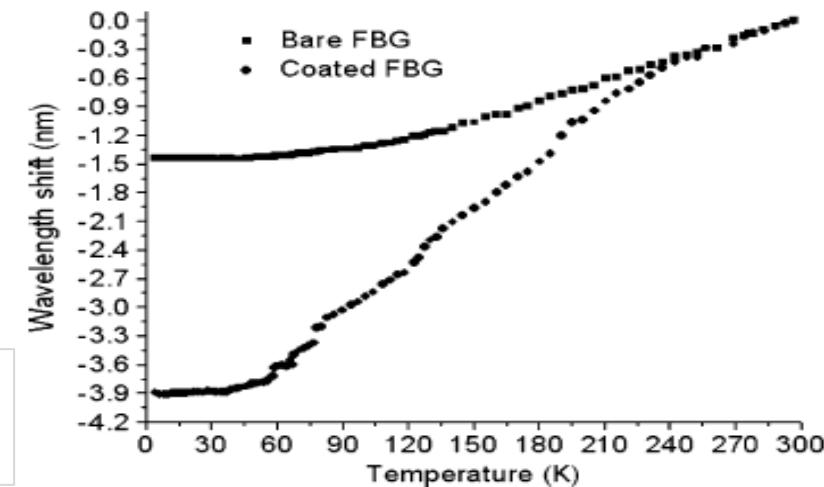
# Temperature sensors: working principle

Poor FBG sensitivity at Cryo

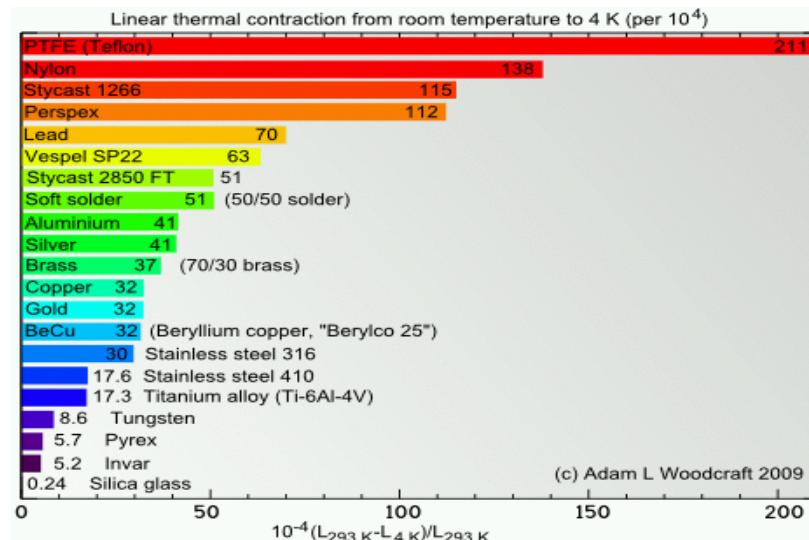
Temperatures

8 - 12 pm/K @ room T  
< 0.1 pm/K below 40 K

Hongjie Zhang, Fanping Deng, Qiliang Wang, Luguang Yan, Yingming Dai, and Keeman Kim, Development of Strain Measurement in Superconducting Magnet Through Fiber Bragg Grating, IEEE TRANSACTIONS ON APPLIED SUPERCONDUCTIVITY, VOL. 18, NO. 2, JUNE 2008;



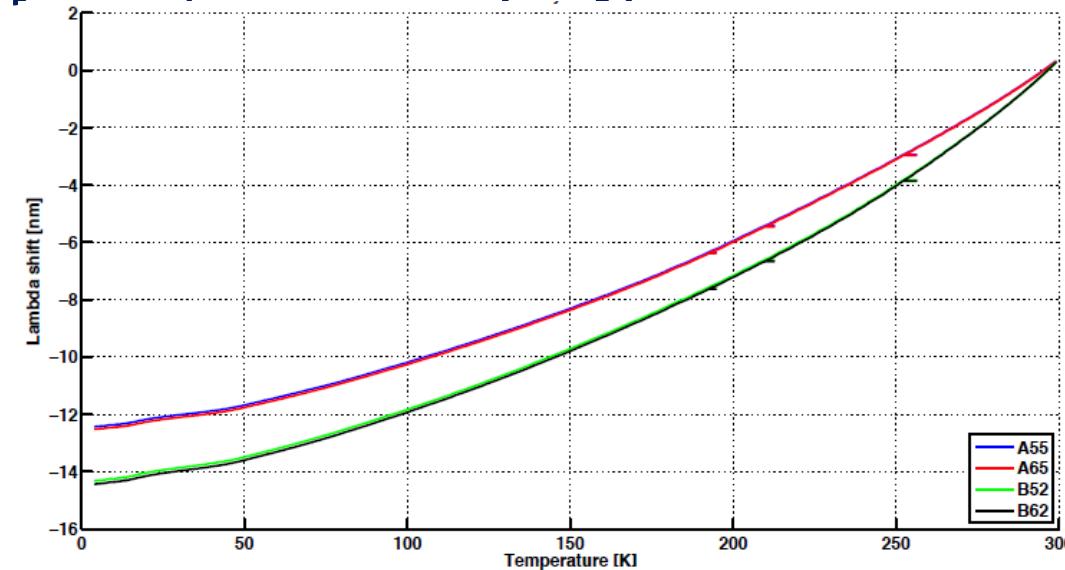
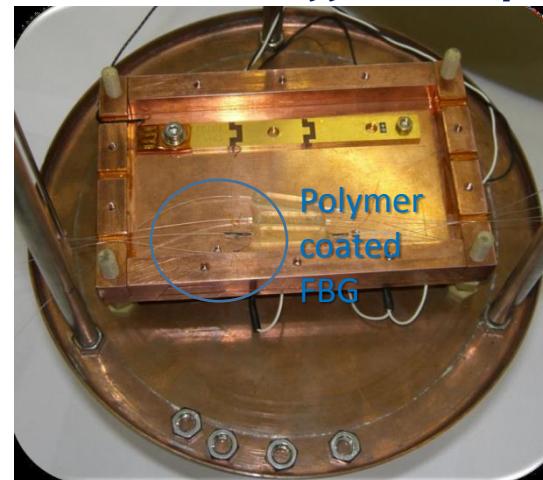
A suitable material with a large thermal expansion coefficient at cryogenic temperature is needed



# Embedded Temperature sensors

Polymer coating can be easy and cheap, it makes the FBG sensors very competitive in terms of fabrication and production

*Tests at 4.2 K have been carried out on FBGs coated with different polymers (PMMA and Epoxy)*



- The analysis shows a strong non linearity within the range 4-10 K but a good feasibility of the coating technique up to 10 K

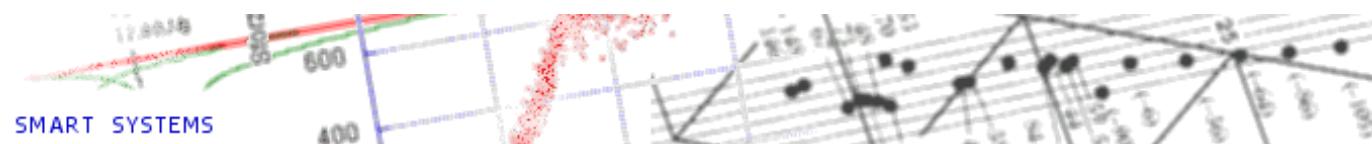


# FBG Health Monitoring Feasibility for Solid Rocket Motor Composite Case

**optoXsmart**  
OPTOELECTRONIC AND SMART SYSTEMS

**Mavio**  
propulsione aerospaziale

**optoXsmart**  
OPTOELECTRONIC AND SMART SYSTEMS



# Solid Rocket Motors Monitoring

- **Aim: Structural Health Monitoring (SHM) of composite cases of Solid Rocket Motors (SRM)**
- Structural behavior of composite part during mechanical testing
- Damage identification during mechanical tests
- Monitoring and identification of impact event during integration and structure transportation



# FBG sensors installation

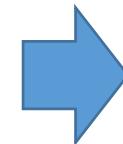
**Sensorization of a small scale vessel:** 5 FBGs have been embedded into the breadboard during the filament winding process

FBGs were installed in the middle part of the cylinder: 3 of them along the hoop layer ( $120^\circ$  each others), 2 of them along the helix path ( $180^\circ$  each others)

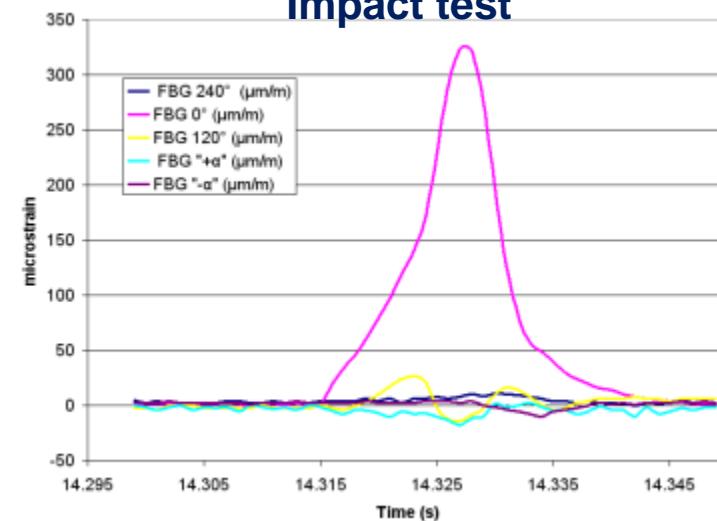


Before testing the item was instrumented by additional 4 strain gauges located on the external surface of the vessel.

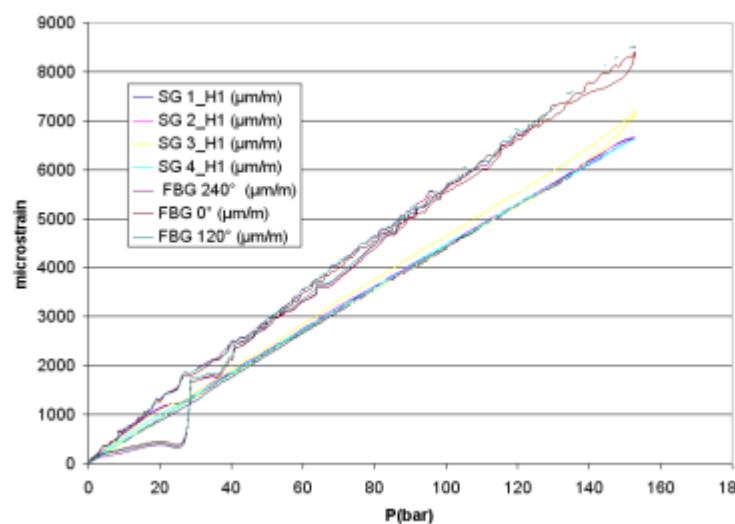
# FBGs validation



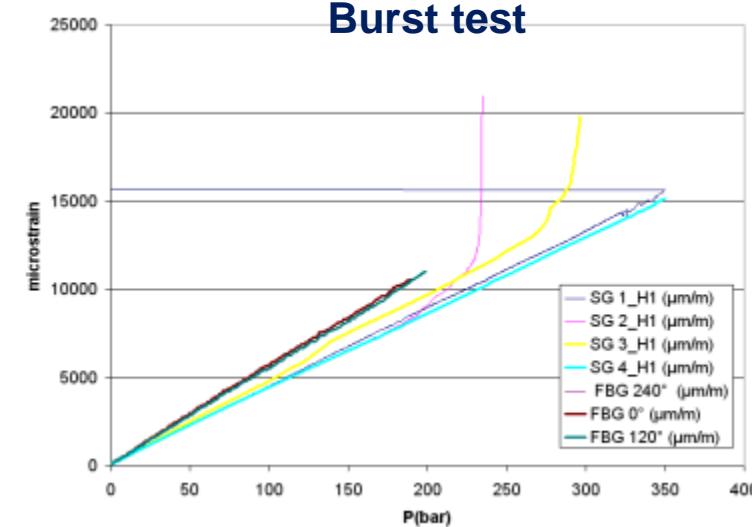
## Impact test



## Proof test

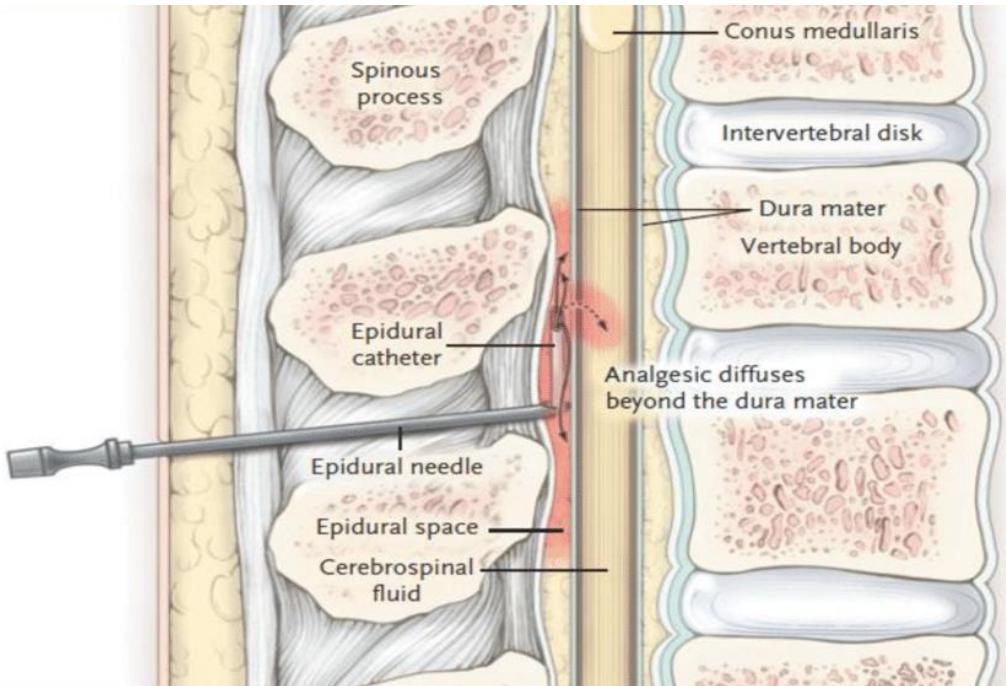


## Burst test

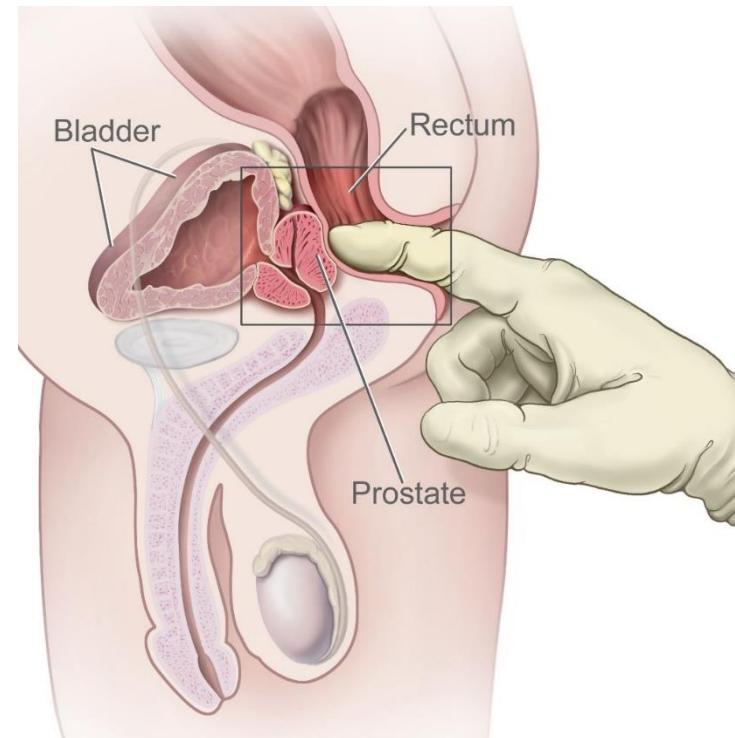


# Smart Needles based on Optical fiber sensors for:

## Epidural anesthesia

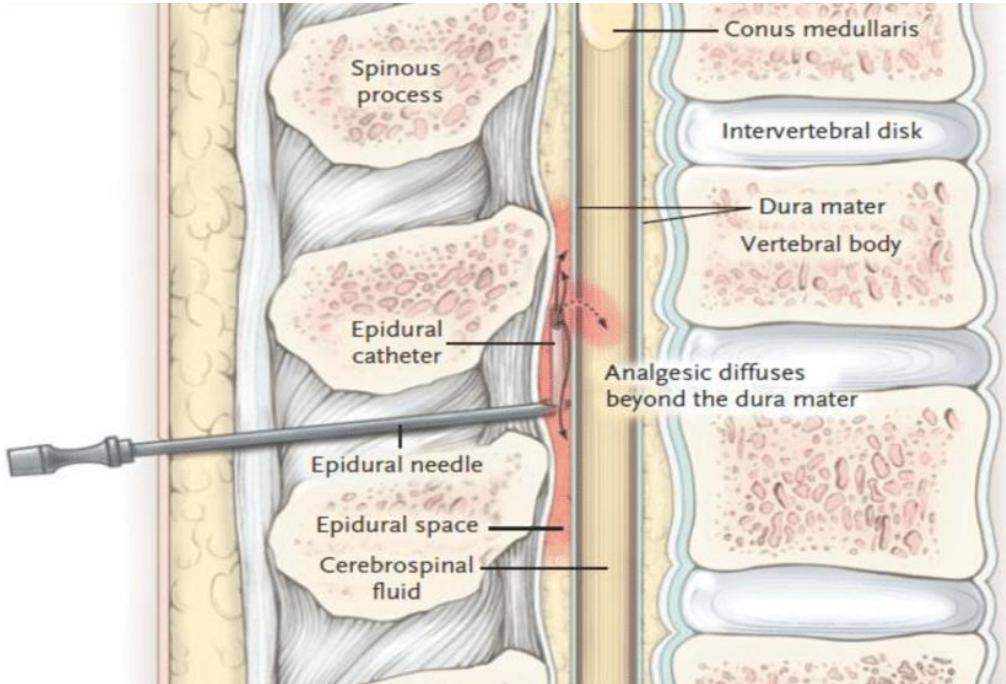


## Prostate cancer screening

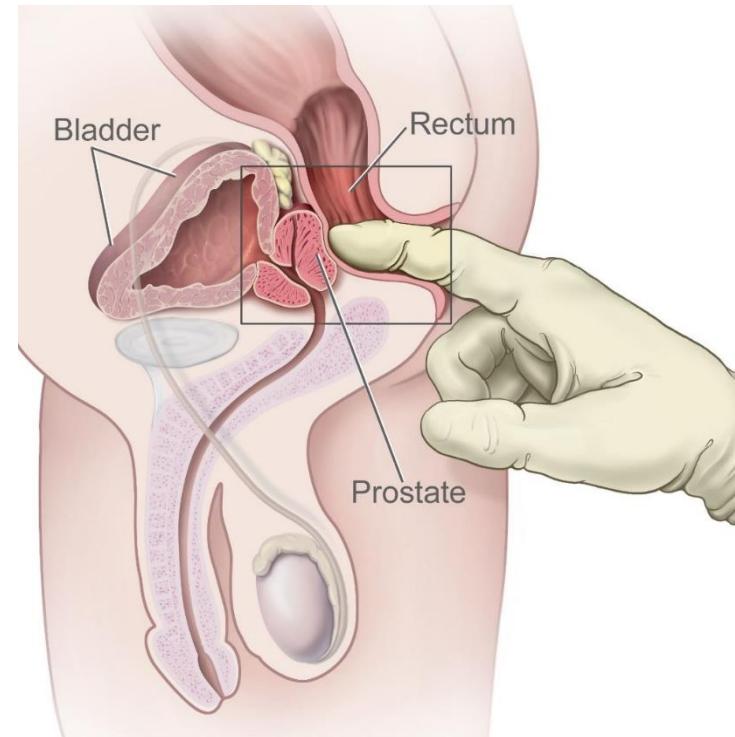


# Smart Needles based on Optical fiber sensors for:

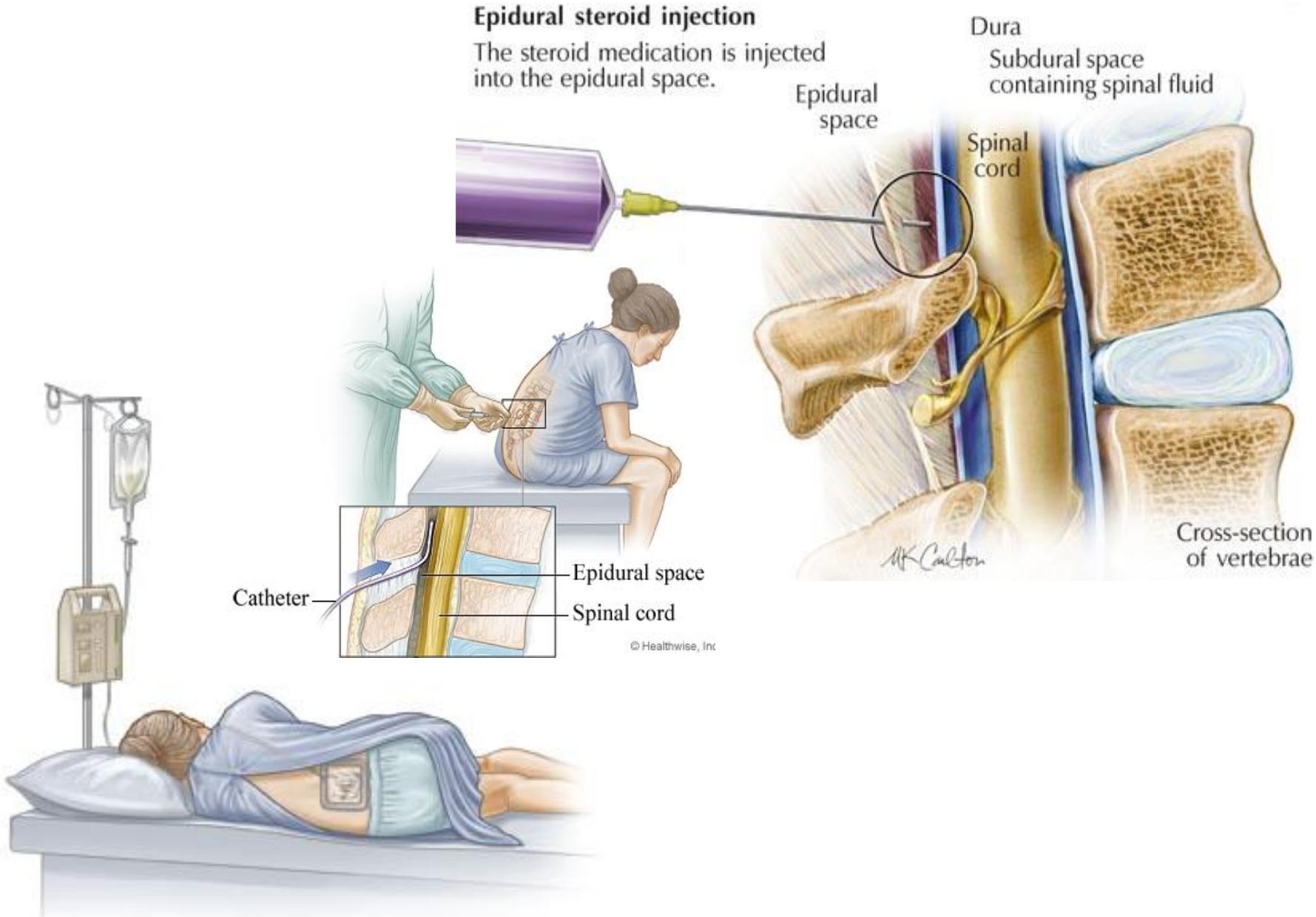
## Epidural anesthesia



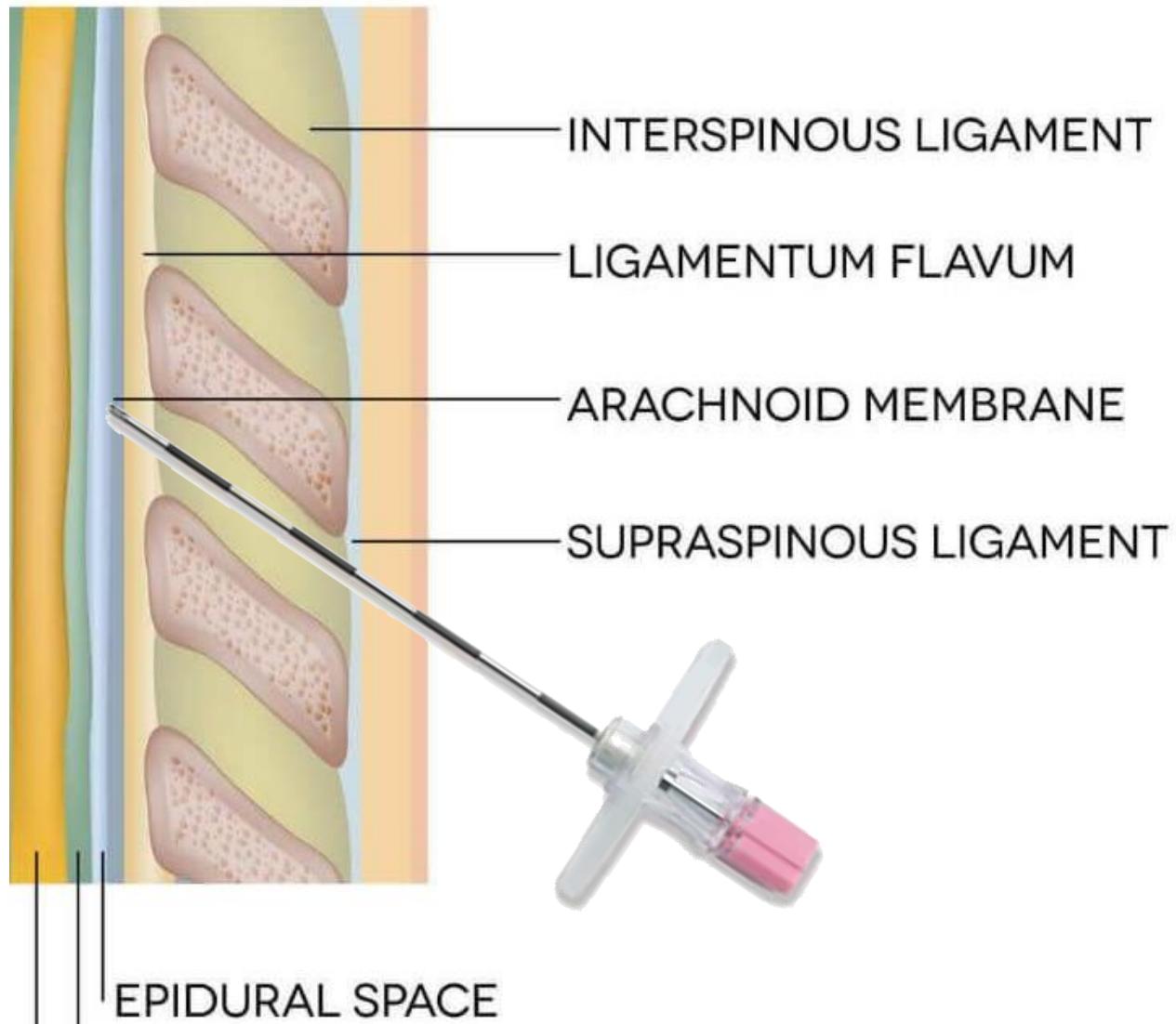
## Prostate cancer screening



# Epidural anesthesia



# The Epidural space



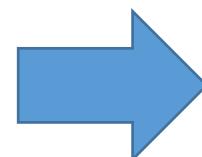
## The standard of care: LOR (Loss of Resistance)

During the LOR technique, 2 to 5 mL of air or saline solutions is drawn into the syringe, and continuous or intermittent pressure is applied to the plunger as the epidural needle is advanced toward the epidural space. On entry into the epidural space, the syringe contents are injected due to a loss in resistance



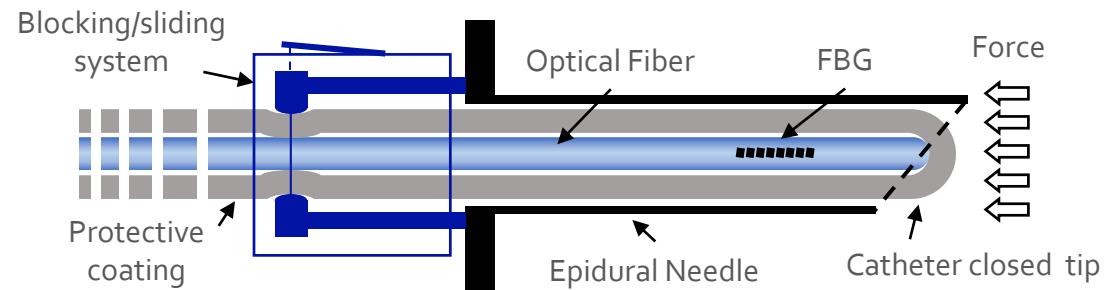
# Our idea....

*the standard epidural kit*



*Our prototype*

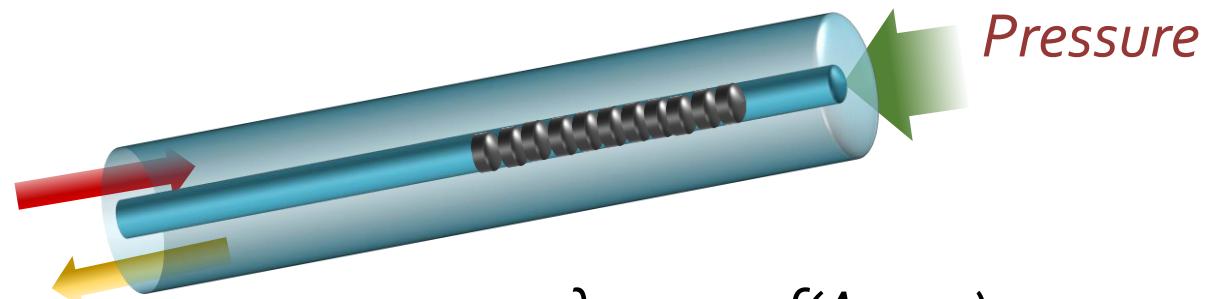
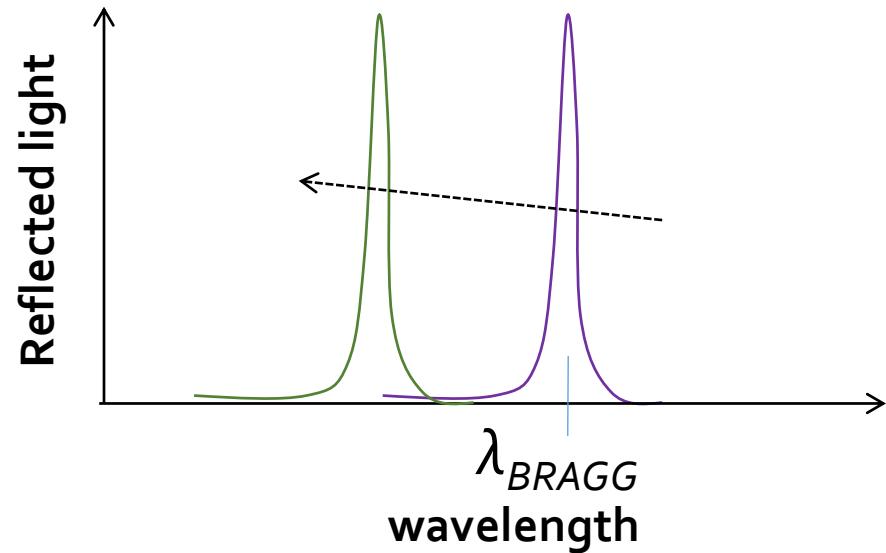
The Fiber Bragg Grating (**FBG**) is integrated inside the Epidural Needle lumen



# The Fiber Bragg Grating

Fiber Bragg Grating: periodic modulation ( $\Lambda$  is the pitch) of the refractive index of the fiber core.  
It acts like a **wavelength selective band-pass filter** in reflection

*compression*

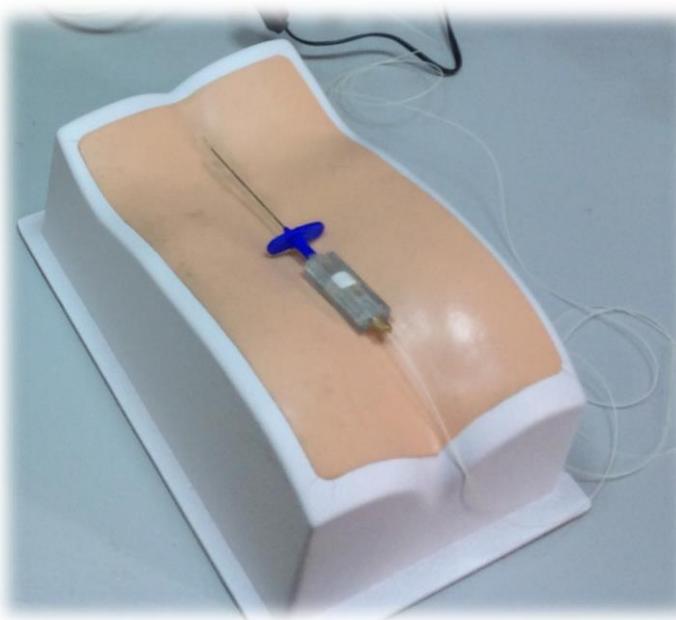


$$\lambda_{BRAGG} = f(\Lambda, n_{eff})$$

$\lambda_B$ =Bragg wavelength  
 $p_e=0.22$  (photoelastic constant)  
 $F$ =force applied  
 $Af$ =probe cross section area  
 $E_{tot}$  = whole Young Modulus  
 $P$ =pressure

$$\frac{\Delta \lambda_B}{\lambda_B} = -(1 - p_e) \cdot \frac{F}{Af \cdot E_{tot}} = -(1 - p_e) \cdot \frac{P}{E_{tot}}$$

# The device development roadmap



1. Tests on silicone membranes



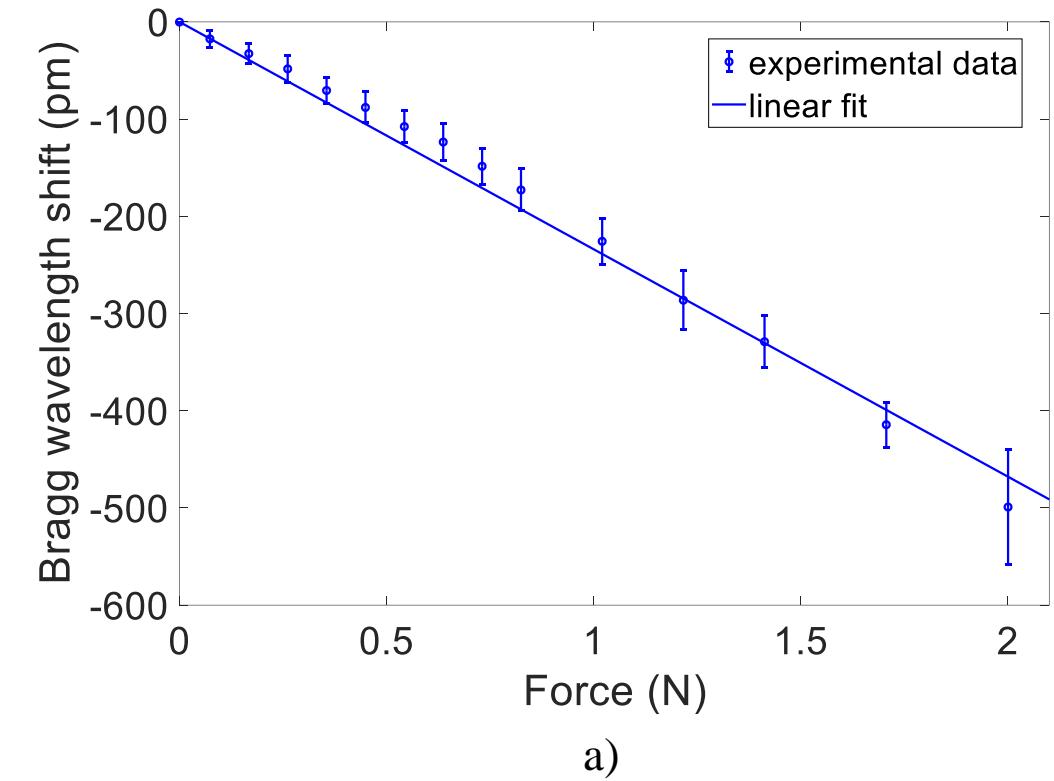
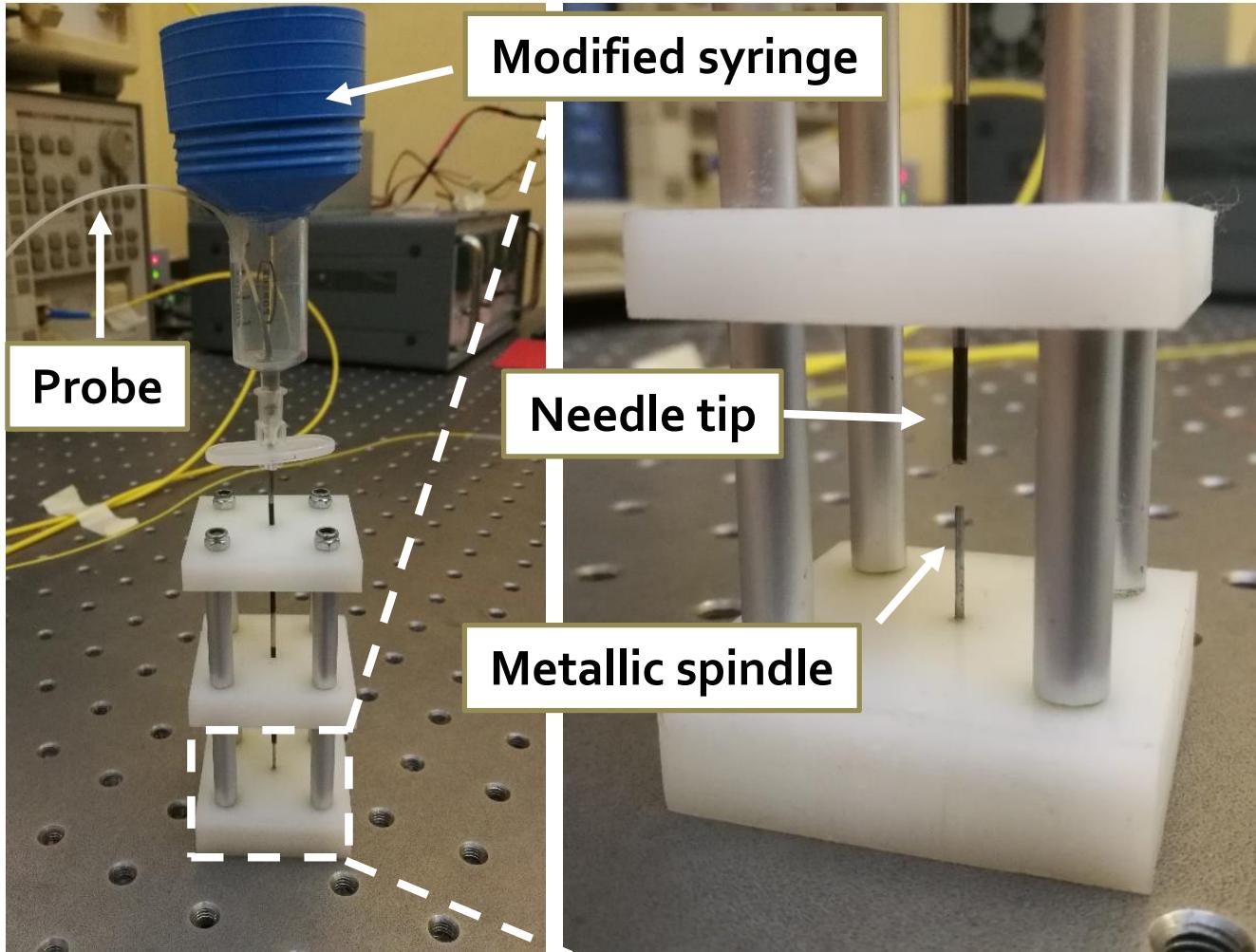
2. Tests on phantom tissue



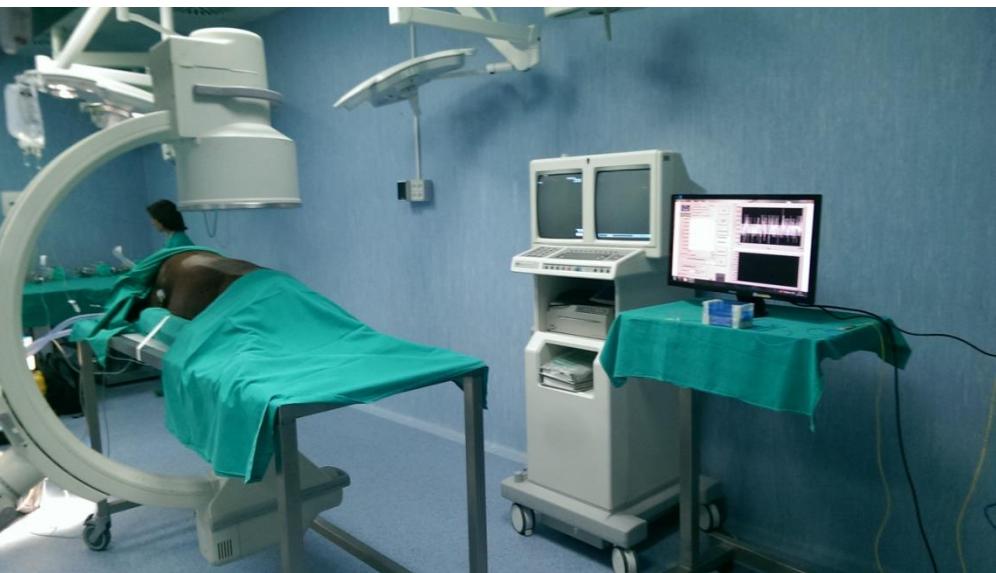
3. Test in vivo (animal study)

# The calibration setup

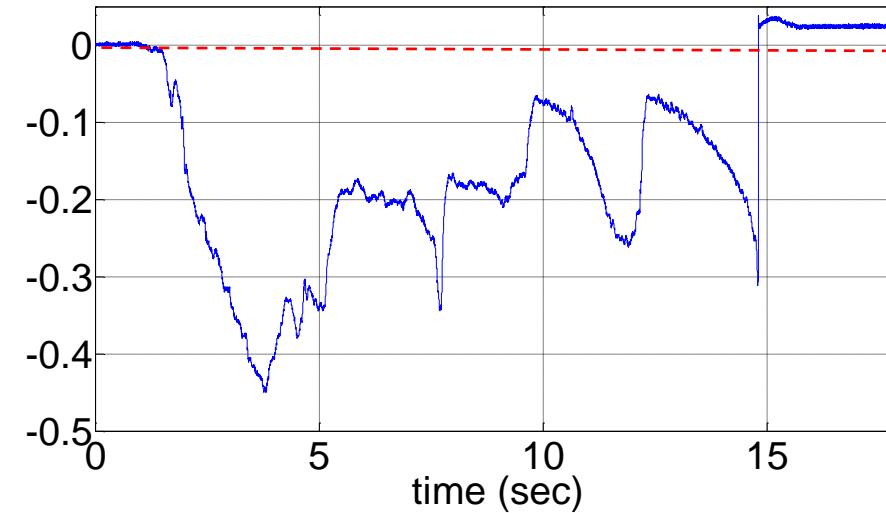
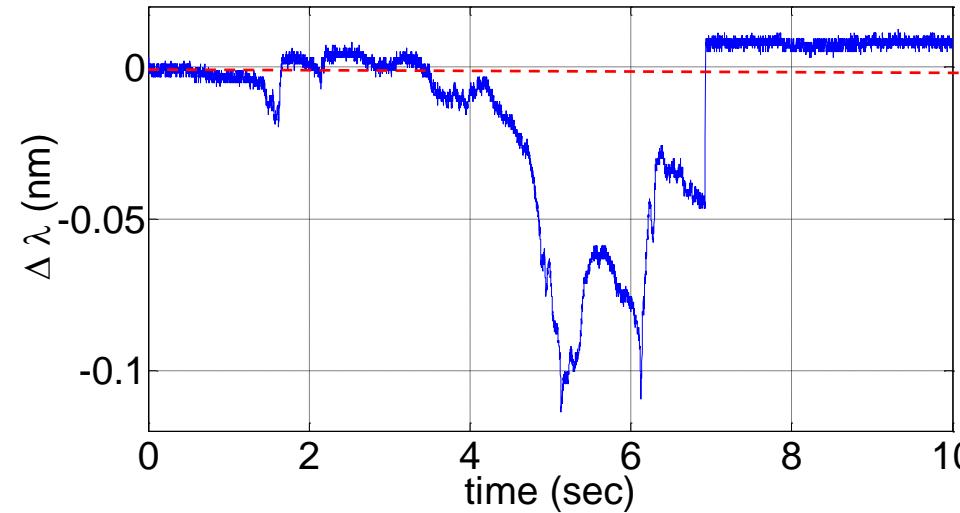
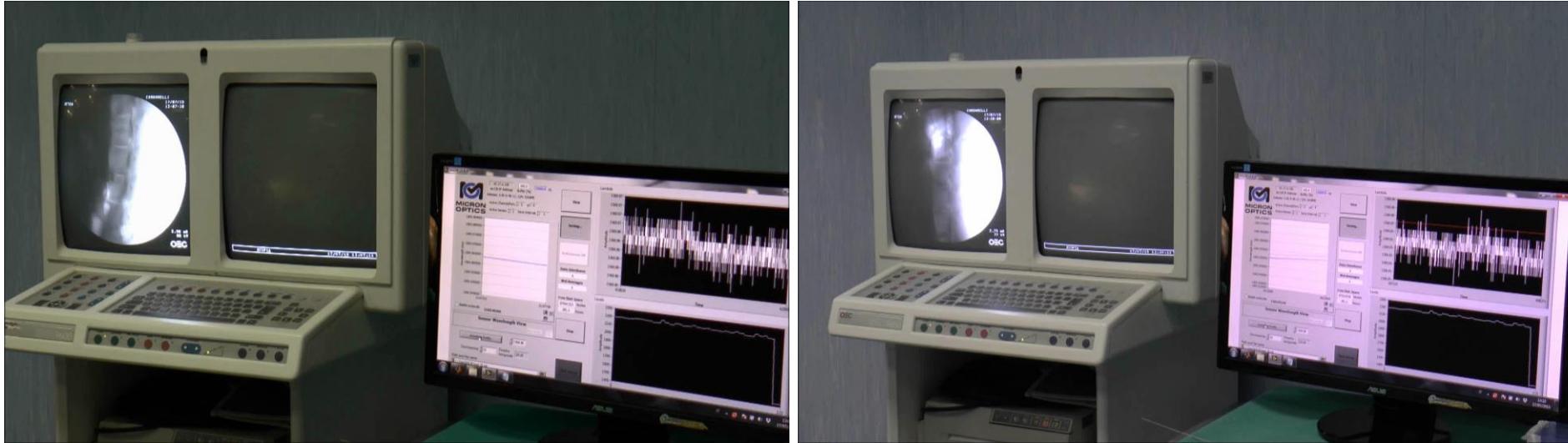
Correlation between the Bragg wavelength shifts and the force experienced by the needle tip



# In vivo test: animal study



# Results (Epidural space identification)



# Il Progetto B.A.R.T.O.LO.

*Bioptic Adavanced Robotic Technologies in OncoLOgy*



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE CAMPANIA

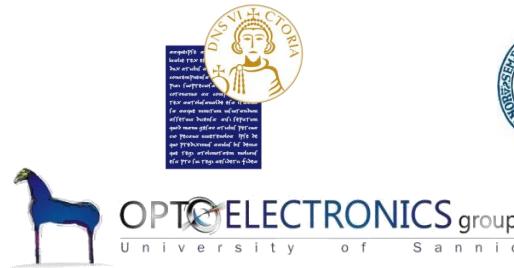


POR CAMPANIA

FESR

2014-2020

PO FESR 2014-2020 – Obiettivo specifico 1.2 - MANIFESTAZIONE DI INTERESSE PER LA REALIZZAZIONE DI TECHNOLOGY PLATFORM NELL'AMBITO DELLA LOTTA ALLE PATOLOGIE ONCOLOGICHE”



**□ Obiettivo generale del progetto:** Sviluppo ed applicazione di nuove tecnologie robotiche per la diagnosi bioptica del tumore prostatico

**□ Ruolo di Unisannio:** Sviluppo di **un probe in fibra ottica**, che sia in grado di rilevare e trasmettere ad un software user friendly, informazioni relative alla rigidità del tessuto prostatico (in termini di composizione elastica).

# Elasticity as a biomarker for prostate cancer: a systematic review

Daniel W. Good\*, Grant D. Stewart\*, Steven Hammer†, Paul Scanlan†, Wenmiao Shu†,  
Simon Phipps\*, Robert Reuben† and Alan S. McNeill\*

\*Edinburgh Urological Cancer Group, University of Edinburgh, Department of Urology, Western General Hospital and

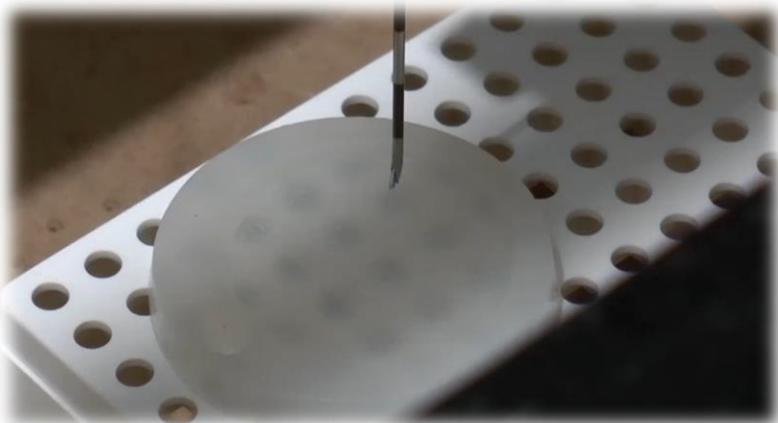
†School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK

- To systematically review the range of methods available for assessing elasticity in the prostate and to examine its use as a biomarker for prostate cancer.
- A systematic review of the electronic database PubMed was performed up to December 2012.
- All relevant studies assessing the use of elasticity as a biomarker for prostate cancer were included except those not studying human prostates or reporting a sensitivity, specificity or quantitative elasticity value.
- There has been much interest in the use of elasticity in the detection of prostate cancer and there have been many publications using various methods of detection. The most common method of assessment is an imaging method, called sonoelastography. Further imaging methods include ultrasound (US), three-dimensional US and magnetic resonance elastography. These methods are reviewed for sensitivity and specificity.
- The other method of assessment is the mechanical method. These use quantitative elasticity values to differentiate benign from malignant areas of the prostate. This method of assessment has shown that the elasticity changes for differing Gleason grades and T stages of disease within the prostate. Quantitative elasticity values offer the potential of using 'threshold' elasticity values under which the prostate is benign.
- Tissue elasticity has great potential as a diagnostic and prognostic biomarker for prostate cancer and can be assessed using various methods. Currently transrectal sonoelastography has the most evidence supporting its use in clinical practice.

## Keywords

elasticity, biomarker, prostate cancer, sonoelastography

# La roadmap dello sviluppo del dispositivo



1. Test su membrane siliconiche



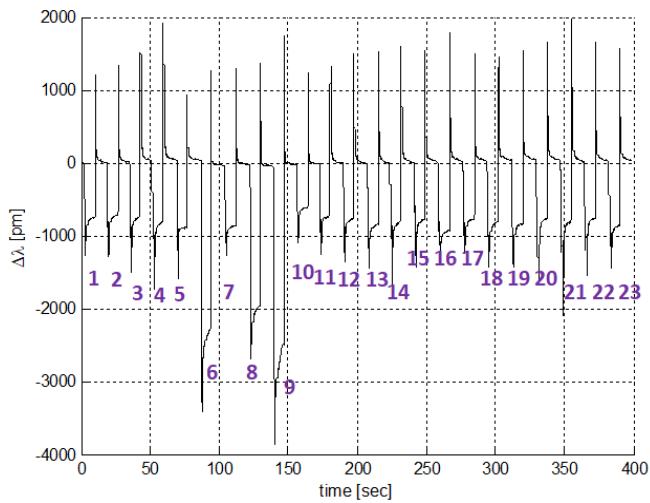
2. Test su phantom tissue



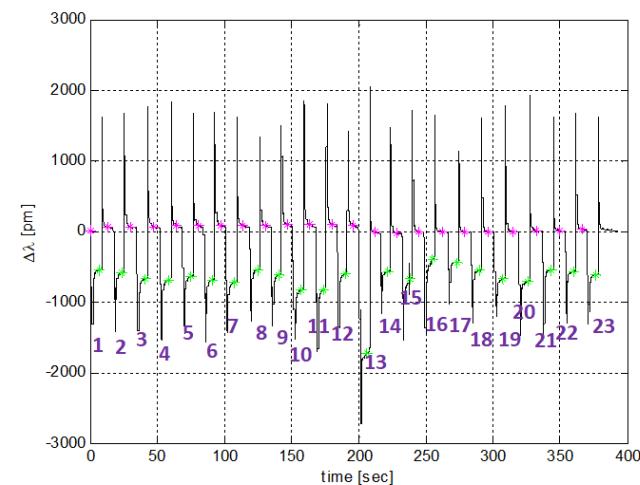
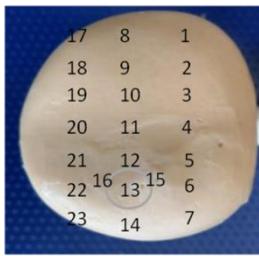
3. Test Ex-vivo

# Test su Phantom tissues

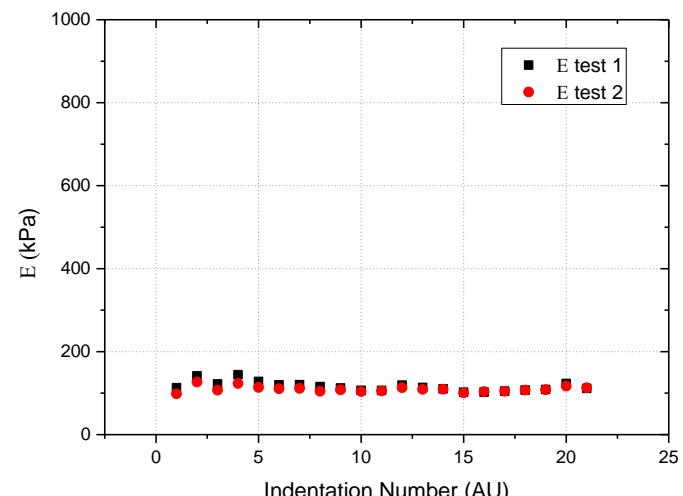
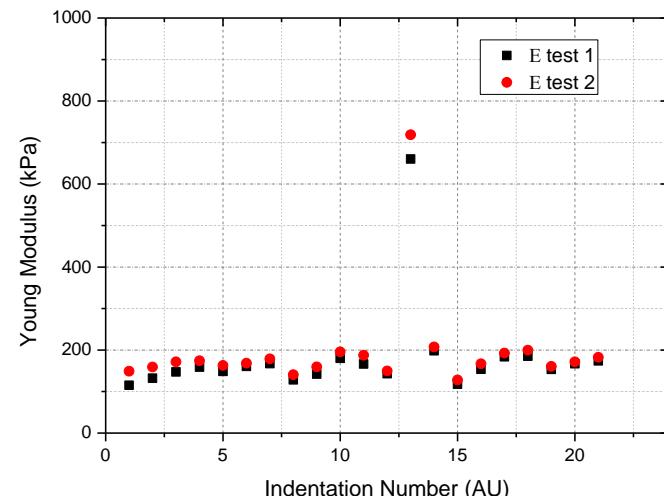
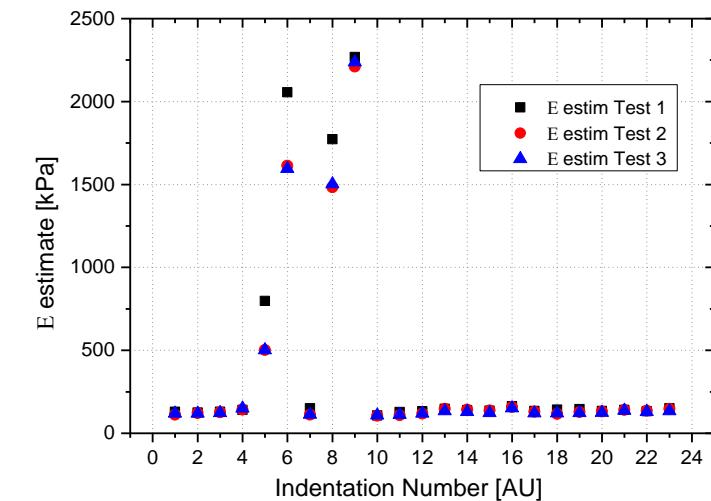
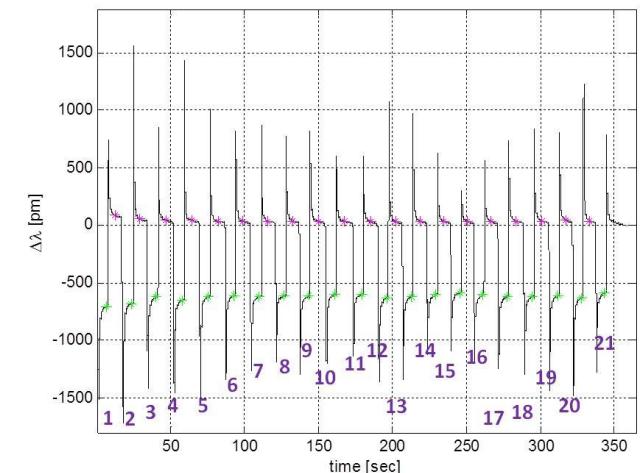
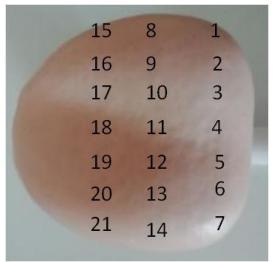
**Prostata #1**



**Prostata #2**



**Prostata #3**



# Test di punzonatura di prostate ex – vivo asportate a pazienti

**Prostata “malata” in posizione frontale**



**Immagine del probe vincolato al robot  
durante un test di punzonatura**

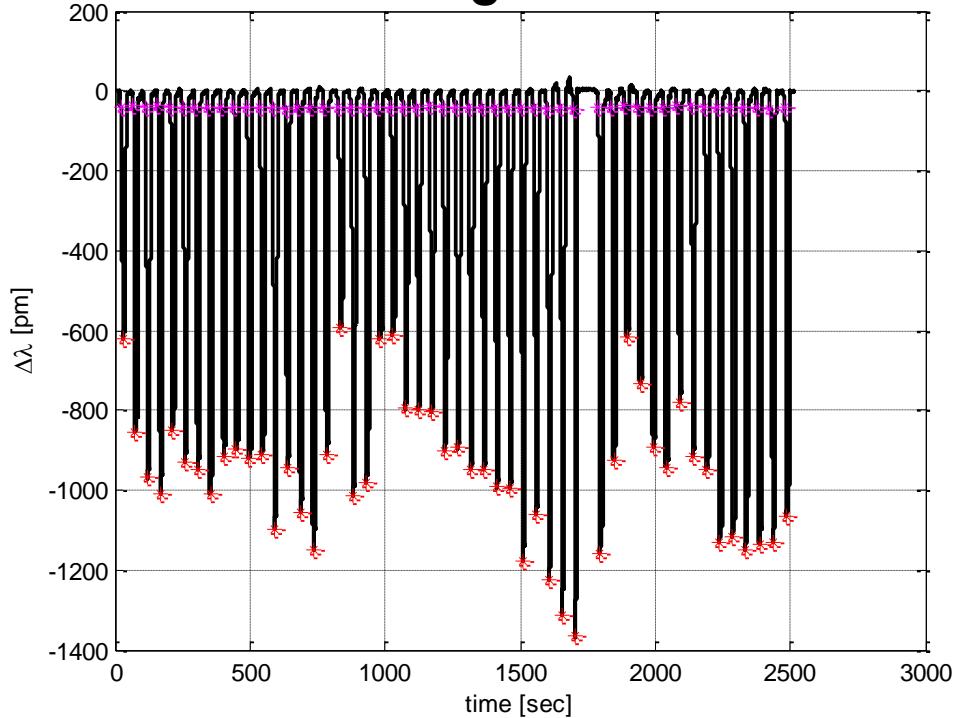


**Prostata “sana” in posizione frontale**

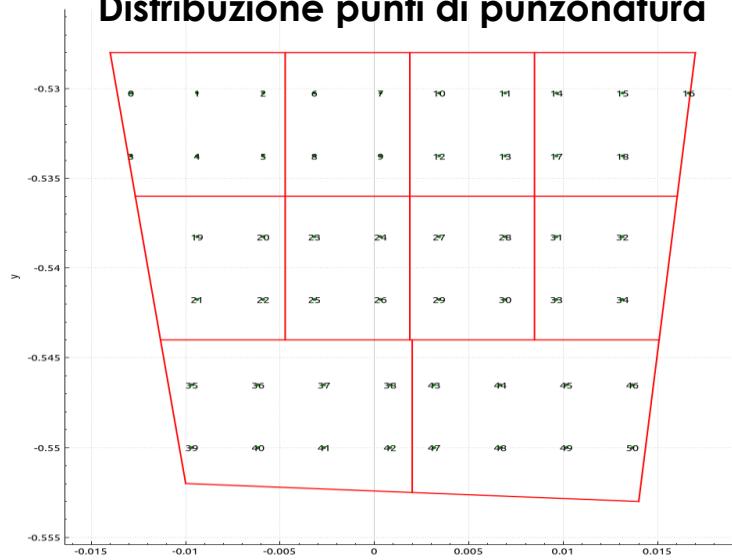


# Test di punzonatura di prostate ex – vivo asportate a pazienti

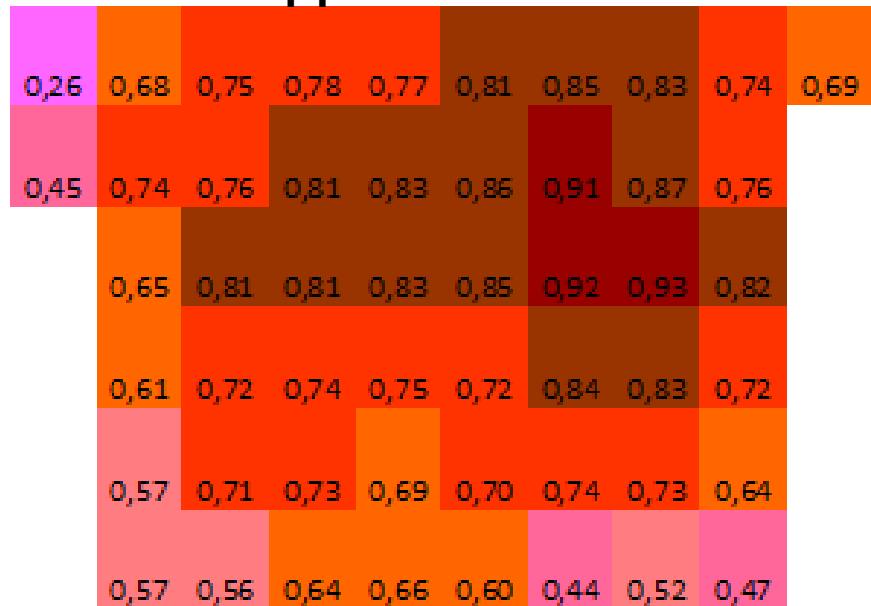
## Sensorgramma



## Distribuzione punti di punzonatura



## Mappa cromatica



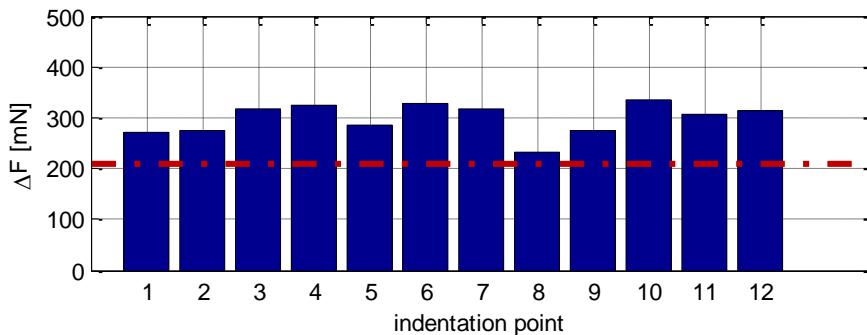
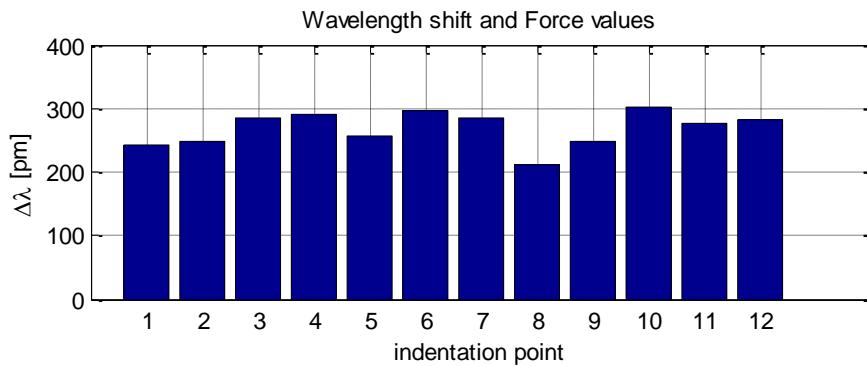
## Immagine della prostata



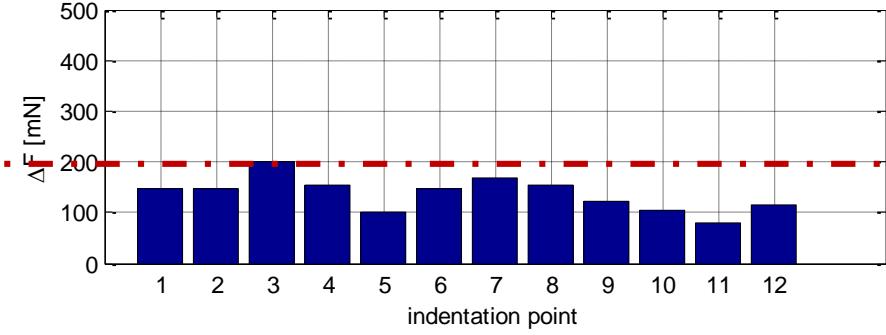
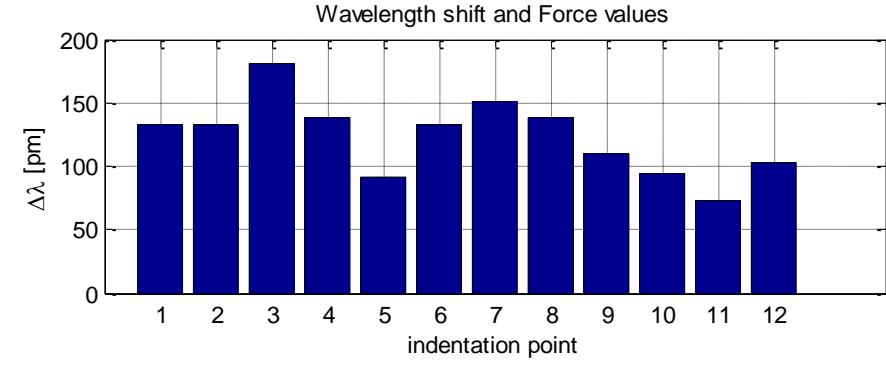
# Test di punzonatura di prostate ex – vivo asportate a pazienti

## Prostata sana Vs prostata malata

### $\Delta\lambda$ (pm) e F (N) – prostata malata



### $\Delta\lambda$ (pm) e F (N) – prostata sana



- Le punzonature della prostata “malata” rispetto a quella “sana” fanno registrare un incremento della livello di forza misurata da probe, che va da un minimo del 52% ad un massimo del 277%.

Ind. #	1	2	3	4	5	6	7	8	9	10	11	12
F [mN] Prostata sana	147	147	200	153	101	147	167	153	122	105	82	114
F [mN] Prostata malata	270	276	317	323	286	328	317	233	275	336	308	315
Δ %	84	88	58	110	183	124	89	52	126	221	277	176

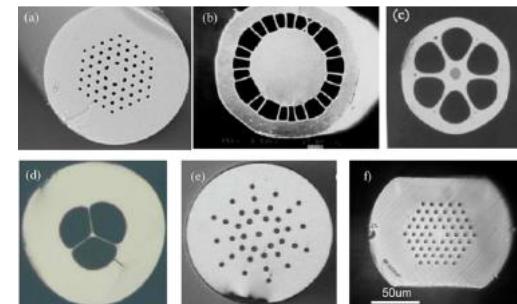
# **ADDING FUNCTIONALITIES**

## ***Microstructured Fiber Bragg Gratings Sensors for Chemical and Biological Applications***

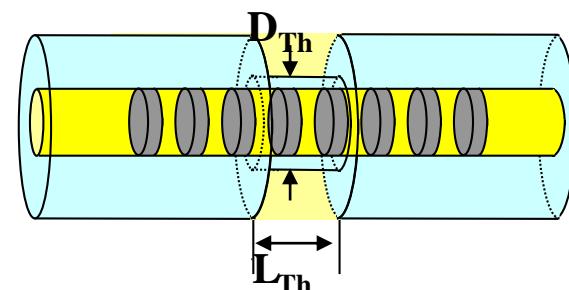
### **1) FBG realization within unconventional fiber structures**

- Microstructured Optical Fiber (MOFs)

J. Canning *et al.*, Laser Chemistry 2008, 239417 (2008)



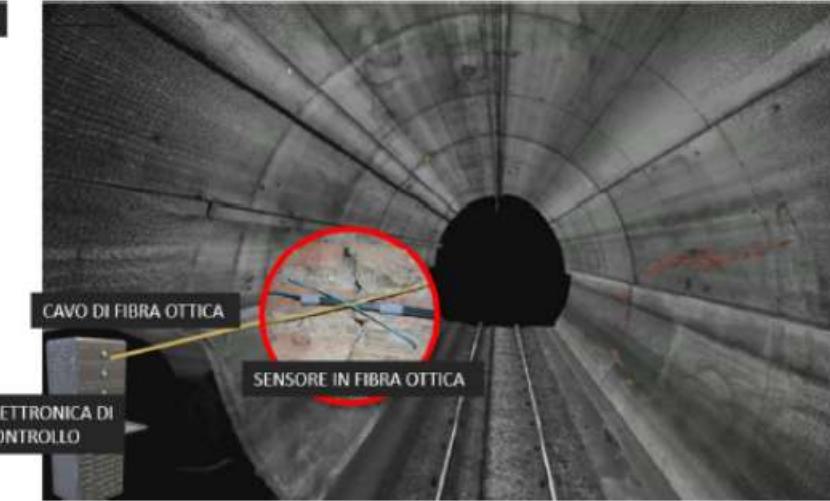
### **2) Conventional FBGs post-processed for structuring the host fiber in localized micro-sized regions: sensitization to surroundings**



A. Cusano *et al.*, Journal of Lightwave Technology 27, 1663 (2009)

**SCOPO:** monitoraggio due gallerie

SISTEMA REMOTO DI ELABORAZIONE DEI DATI



5 sezioni per galleria da monitorare

per ogni sezione devono essere installati al massimo 5 sensori -> 25 sensori

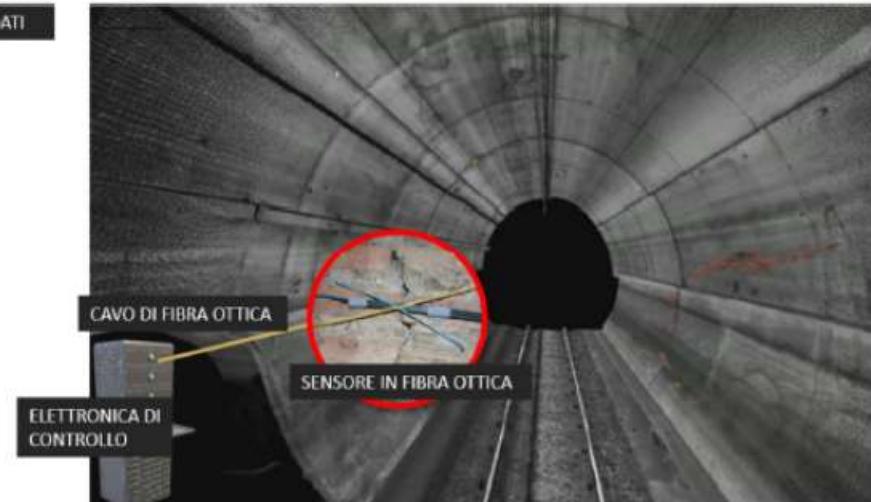
devono essere in grado di rilevare variazioni dell'ordine di  $\pm 5 \text{ mm}$

7 sensori di spostamento in fibra per monitorare l'ampiezza delle fessure

devono essere in grado di rilevare variazioni dell'ordine di  $\pm 1500 \mu\epsilon$

Per il monitoraggio delle ampiezze delle fessure, sarà necessario installare un secondo sistema di monitoraggio "tradizionale" che non richieda alimentazione.

SISTEMA REMOTO DI ELABORAZIONE DEI DATI



## Osservabili

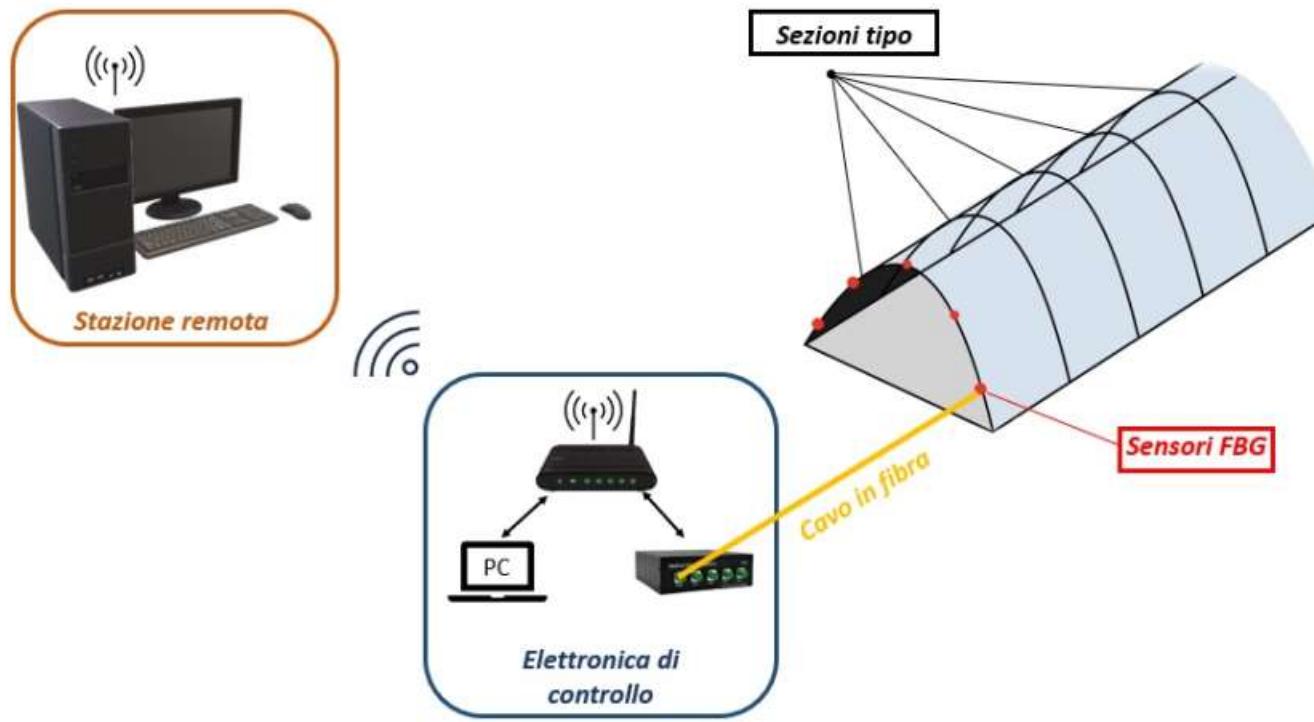
1. Convergenza
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

Per garantire l'accuratezza delle misurazioni, i sensori di deformazione in fibra ottica devono essere accoppiati a sensori per la compensazione termica, in modo che le variazioni di temperatura non influiscano sulle misurazioni.

I dati **acquisiti** devono essere **elaborati** da un software di elaborazione associato alla sensoristica, installato sull'unità centrale, che applica modelli matematici per ricavare i parametri richiesti;

Il software di **visualizzazione** dei risultati deve consentire all'utente di interrogare in tempo reale il sistema, di estrarre report relativi ai dati rilevati in un determinato arco temporale e di segnalare il superamento di valori soglia;

La **remotizzazione**, mediante modem wi-fi, delle misure trasferite via cavo ottico ad un'elettronica di controllo in connessione con una postazione centrale.



Il sistema di monitoraggio è costituito da un insieme di componenti

1. Interrogatore;
2. Cavo multifibra rinforzato;
3. Sensori di deformazione in fibra di tipo «short-gage»;
4. Sensori di deformazione in fibra di tipo «long-gage»;
5. Sensori di spostamento in fibra e flessurimetri tradizionali;
6. Un pc locale ed un modem.

# Interrogatore



## FBG Wavelength Interrogator

The FBG Wavelength Interrogator AGSWA is designed to be a robust, reliable and long-term field operation for the FBG sensing system. Based on a built-in wavelength analysis module, AGSWA can update the data in high accuracy and high frequency.

AGSWA responds directly to the user commands of the optical interrogator core and outputs sensor wavelength data. It provides a single suite of tools for data acquisition, computation, and analysis of optical sensor networks.

 Get a Quote

## Specification Of FBG Wavelength Interrogator

Optical Properties		
Number of Optical channels	1, 2, 4, 8, 16, 24, 32 or 36	Customized
Scan Frequency	5000 Hz 500Hz	Single Channel Four Channels
Wavelength Range	1510nm-1590nm	
Wavelength Repeatability	+/-3pm	
Wavelength Resolution	1 pm	
Dynamic Range	>30 dB	
Standard number of FBG Per Channel	40	
Connector	FC/APC	Customized

# Interrogatore

## Data Processing Capabilities

Interface	Ethernet
Software Features	For spectral analysis, peak detection, data logger, peak tracking, user can read out in temperature, pressure, or strain, etc, save data to excel document
Software support	Windows

## Mechanical, Environmental, Electrical Properties

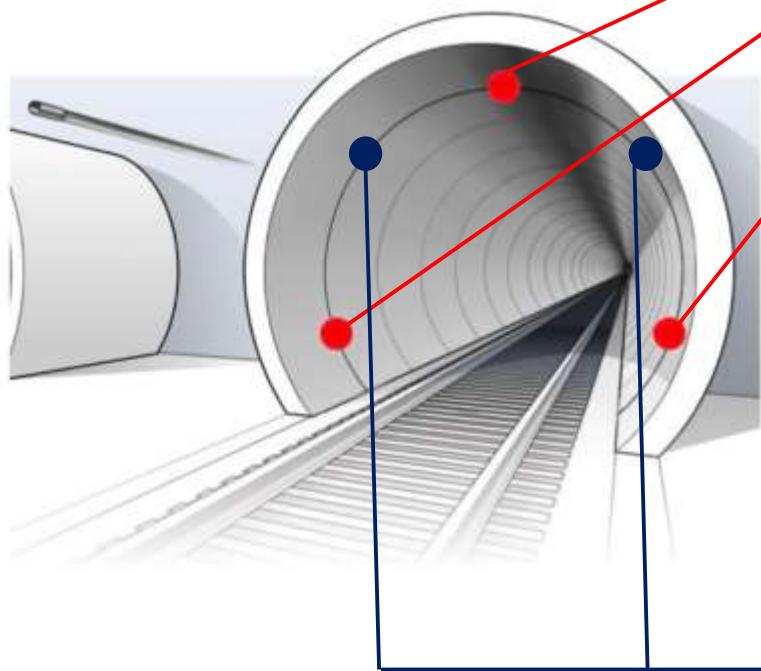
Dimension	260mm*210mm*62mm
Operation Humidity	0%-80%RH
Operation Temperature	-5° to +50° C
Storage Temperature	-20° to +60° C
Input Voltage	DC 12V
Maximum Power	30W

Per effettuare il collegamento tra il set di sensori e l'interrogatore, posto all'imbocco della galleria, viene utilizzato un cavo multifibra rinforzato.



# MONITORAGGIO DELLA SEZIONE

Sensori di deformazione in fibra compensati in temperatura di tipo *short-gage*: **FBG MST-01**



Sensori di deformazione in fibra compensati in temperatura di tipo *long-gage*: **os3610**



# MONITORAGGIO DELLA SEZIONE



***short-gage: FBG MST-01***



Parametri	Unità di misura	Valore
Lunghezza d'onda centrale	nm	1460 ~ 1610
Intervallo	$\mu\epsilon$	$\pm 1500$
Risoluzione	$\mu\epsilon$	0,01% F.S.
Precisione	$\mu\epsilon$	0,1% F.S.
Compensazione di temperatura	-	Incorporata
Materiale	-	Acciaio inossidabile
Dimensioni	mm	$\phi 12 \times 126$
Installazione	-	Saldatura
Connettore ottico	-	FC/APC, o personalizzabile
Temperatura di esercizio	°C	-20 ~ +80

# MONITORAGGIO DELLA SEZIONE



**Long-gage: os361**



Parametri	Valore
Sensibilità allo strain e temperatura	$\sim 1.2 \text{ pm}/\mu\epsilon$ ; $22 \text{ pm}/^\circ\text{C}$
Compensazione di temperatura	Integrata
Lunghezze disponibili	25 o 100 cm
Temperatura di esercizio	-40 ~ 80°C
Limite di deformazione	$5000\mu\epsilon$ (Punto di zero regolabile)
Tempo di vita	$> 1 \times 10^8$ cicli @ $\pm 2000\mu\epsilon$
Peso	190g (25 cm); 384g (100 cm)
Materiale	Acciaio inossidabile/Teflon
Tipo di cavo	Cavo armato da 3 mm
Raggio di curvatura del cavo	$\geq 17\text{mm}$
Metodi di ancoraggio	Solo per montaggio su superficie
Connettore ottico	FC/APC
Riflettività di picco (Rmax)	> 70%
FWHM (punto a -3dB)	0.25nm ( $\pm .05\text{nm}$ )
Isolamento	> 15 dB
Spaziatura tra reticolni	4nm

# Monitoraggio delle variazioni di ampiezza delle fessure

Per monitorare le variazioni di ampiezza delle fessure, è stato scelto il sensore di spostamento in fibra FBG DST-01, fornito da ATgrating:



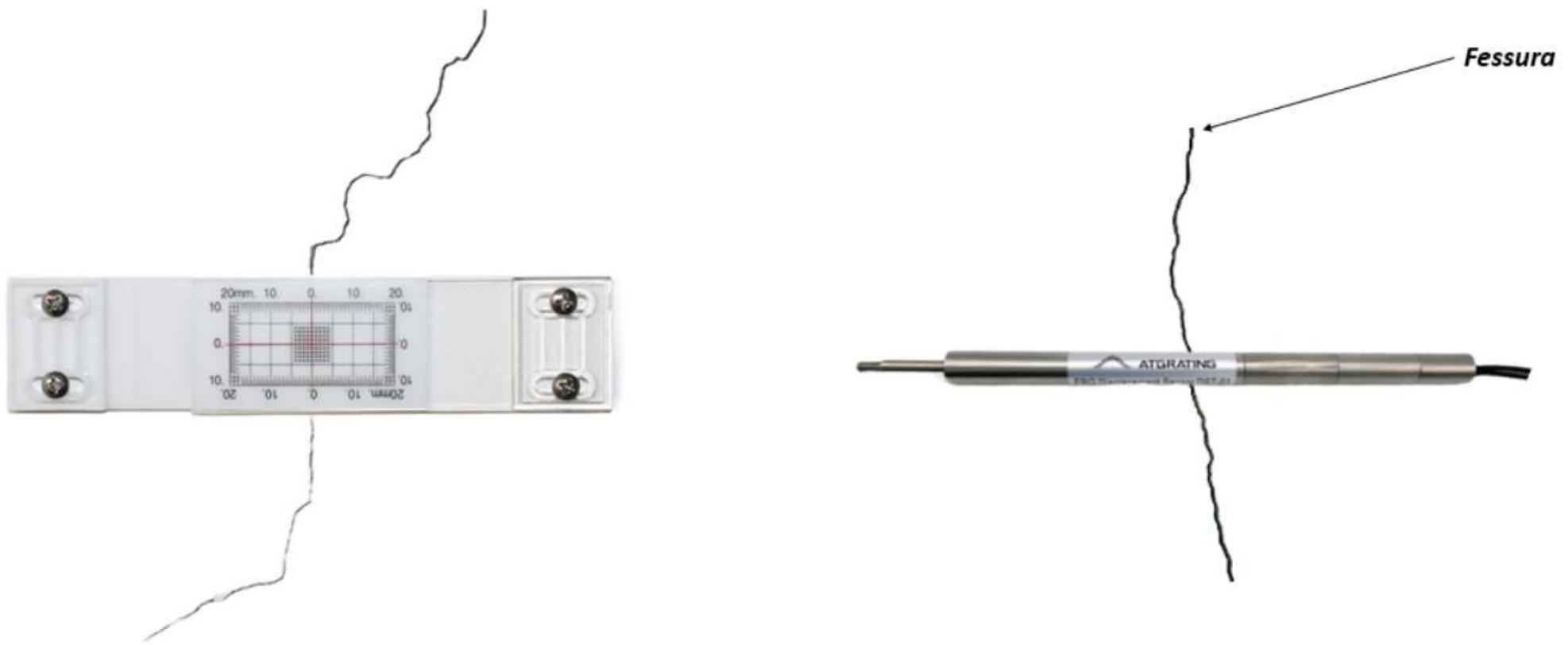
Per l'installazione, il sensore viene montato ortogonalmente alla fessura.



Parametri	Unità di misura	Valore
Lunghezza d'onda centrale	nm	1460 ~ 1610
Intervallo	mm	10 ~ 150
Risoluzione	mm	0,05% F.S.
Precisione	mm	0,5% F.S.
Compensazione di temperatura	-	Incorporata
Materiale	-	Acciaio inossidabile
Dimensioni	mm	φ22 x 234
Installazione	-	Fissaggio meccanico
Connettore ottico	-	FC/APC, o personalizzabile
Temperatura di esercizio	°C	-20 ~ +80

# Monitoraggio delle variazioni di ampiezza delle fessure

Per valutare le prestazioni dei sensori ottici in relazione alle fessure, vengono impiegati anche sensori tradizionali, in particolare i fessurimetri manuali.



# **Numero di sensori per canale dell'interrogatore e numero di canali necessari**

È importante considerare alcuni fattori fondamentali:

- È necessario ricordare che ogni sensore di deformazione e spostamento ha due reticolli a causa della compensazione integrata della temperatura;
- È importante prendere in considerazione il massimo campo operativo di ogni sensore richiesto dalla nostra applicazione, ovvero  $\pm 1550 \mu\epsilon$ ;
- È necessario includere una banda di guardia per evitare sovrapposizioni degli spettri dei sensori allocati alle lunghezze d'onda adiacenti.

Complessivamente sono presenti

- 25 sensori (50 reticolli) per la misurazione della deformazione:
  - 15 di tipo short-gage (30 reticolli);
  - 10 sono di tipo long-gage (20 reticolli);
- 7 crackmetri (14 reticolli) per la misurazione dello spostamento.

Dalle specifiche dell'interrogatore si evince che, per ogni canale, la banda di lunghezza d'onda supportata è compresa tra 1510-1590 nm:

**Banda disponibile per canale: 80 nm**

# Occupazione in banda dei sensori

Compensati in temperatura -> 2 FBG  
La larghezza di spettro dei picchi di ognuno è di 0.5 nm spaziati di 4 nm.  
La larghezza di ogni spettro è di 0.5 nm.

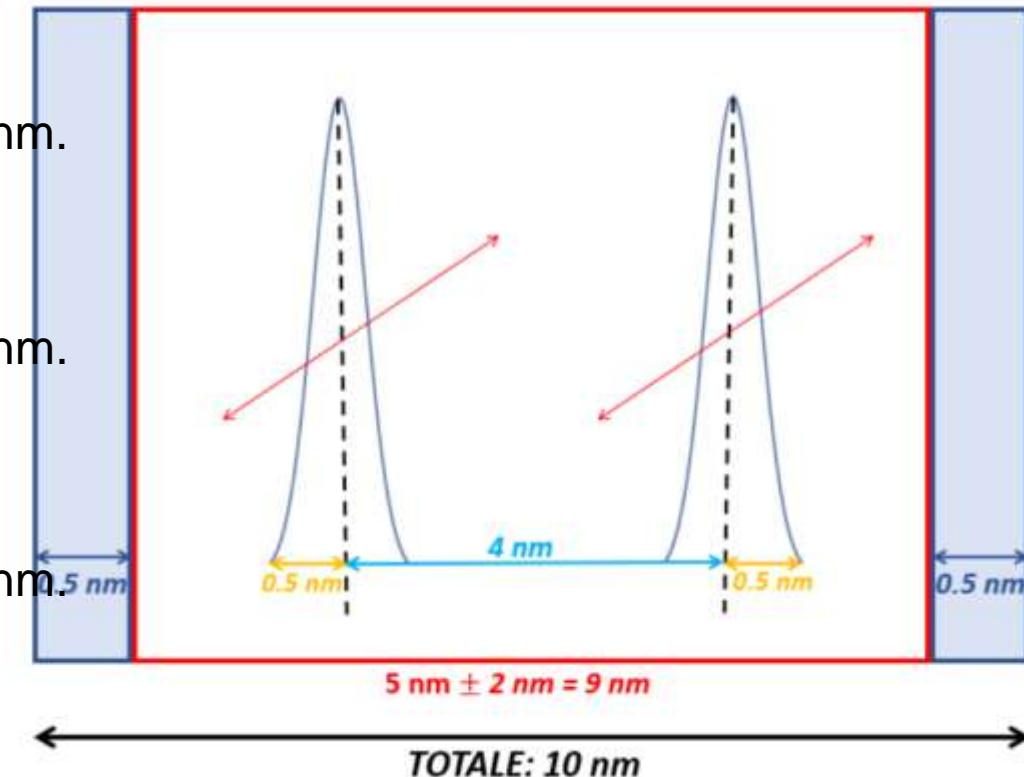
Massimo campo operativo  $\pm 1550 \mu\epsilon$  con una sensibilità pari a  $1 \text{ pm}/\mu\epsilon \rightarrow 1.5 \text{ nm}$   
La banda totale  $5 \text{ nm} \pm 2 \text{ nm}$ .

Aggiungendo una banda di guardia di 0.5 nm sia a destra che a sinistra.

Occupazione in banda: 5 nm.

Occupazione in banda: 3 nm.

Occupazione in banda: 1 nm.



**L'occupazione di banda di un singolo sensore di risulta essere di 10 nm.**

# Occupazione in banda dei sensori

Sensore	Quantità	Occupazione in banda totale
Sensore di deformazione short-gage	15	150 nm
Sensore di deformazione long-gage	10	100 nm
Sensore di spostamento	7	70 nm
<b>TOTALE</b>	<b>32</b>	<b>320 nm</b>

- **Canale 1:** questo canale può ospitare otto sensori di deformazione short-gage, i quali occupano l'intera banda disponibile di 80 nm.
- **Canale 2:** in questo canale è possibile collocare sette sensori di tipo short-gage e un sensore di tipo long-gage, per un totale di 80 nm di banda impiegata, come nel primo canale.
- **Canale 3:** questo canale può ospitare otto sensori di tipo long-gage, che occupano complessivamente 80 nm di banda.
- **Canale 4:** infine, questo canale può contenere un sensore di tipo long-gage e sette sensori di spostamento, con una banda totale di 80 nm

# Occupazione in banda dei sensori



Bisogna ora associare ad ogni sensore la corrispondente lunghezza d'onda di centro banda. Siccome i canali hanno lo stesso numero di sensori e la stessa occupazione di banda, eseguiremo questa operazione su un solo canale.

Considerando che l'interrogatore supporta un range di lunghezze d'onda che va da 1510 nm a 1590 nm, possiamo assegnare le lunghezze d'onda centrali ai 8 sensori come segue:

Sensore	Lunghezza d'onda centrale
1° Sensore	1515 nm
2° Sensore	1525 nm
3° Sensore	1535 nm
4° Sensore	1545 nm
5° Sensore	1555 nm
6° Sensore	1565 nm
7° Sensore	1575 nm
8° Sensore	1585 nm

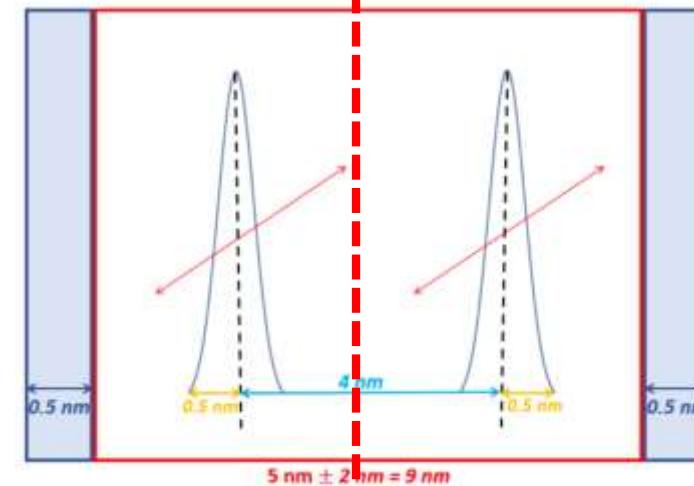
# Occupazione in banda dei sensori

Sensore	Lunghezza d'onda centrale
1° Sensore	1515 nm
2° Sensore	1525 nm
3° Sensore	1535 nm
4° Sensore	1545 nm
5° Sensore	1555 nm
6° Sensore	1565 nm
7° Sensore	1575 nm
8° Sensore	1585 nm

Lunghezza d'onda centrale  
1515 nm

## ATTENZIONE

Lunghezza d'onda centrale fa riferimento alla lunghezza d'onda di centro banda non al singolo picco dei due FBG per sensore.



Lunghezza d'onda centrale

Lunghezza d'onda FBG 1 -> 1513 nm

Lunghezza d'onda FBG 2 -> 1517 nm

# **OSSERVABILI**

1. Convergenza
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento



- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

La convergenza, indicata con il simbolo  $\delta$ , è data dalla variazione del raggio della sezione.

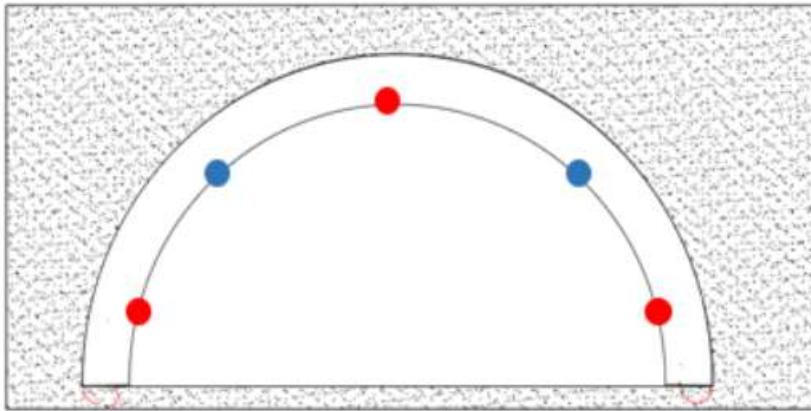
$$\delta = R_1 - R_2$$

$R_1$  -> Raggio iniziale della sezione

$R_2$  -> Raggio finale a seguito di una deformazione

Configurazione dei sensori montati sulla sezione

- short-gage
- long-gage



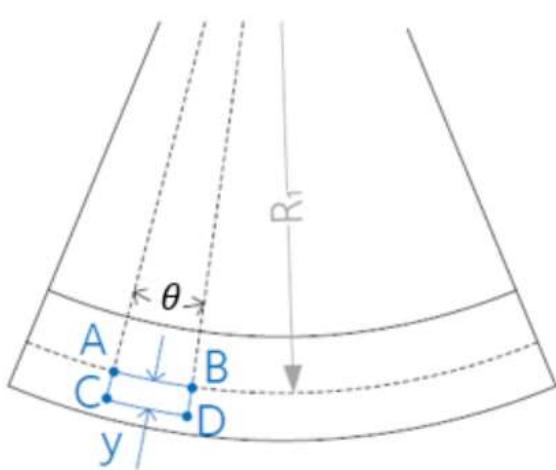
Esempio di convergenza



- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

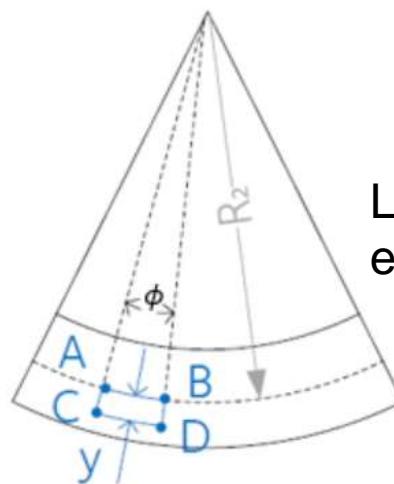
Per ogni sezione, i 5 sensori ci forniscono informazioni sulla deformazione circonferenziale, che rappresentiamo con il si

### Modello matematico: Teoria della Trave Inflessa



Determiniamo la deformazione come:

$$\epsilon_{CD} = \frac{y(R_1/R_2\theta - \theta)}{(R_1 - y)\theta} = \frac{y(R_1 - R_2)}{R_2(R_1 + y)}$$



Lo spostamento radiale è calcolato mediante la seguente equazione:

$$\delta = R_1 - R_2 = \frac{\epsilon_{CD}R_1(y + R_1)}{\epsilon_{CD}R_1 + y(1 + \epsilon_{CD})}$$

- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

Per ogni sezione, i 5 sensori ci forniscono informazioni sulla deformazione circonferenziale, che rappresentiamo con il si

### **Modello matematico:** Teoria della Trave Inflessa

$$\delta = R_1 - R_2 = \frac{\epsilon_{CD} R_1 (y + R_1)}{\epsilon_{CD} R_1 + y(1 + \epsilon_{CD})}$$

La distanza tra il nostro sensore e l'asse neutro, rappresentata da  $y$ , viene stimata utilizzando dati geotecnici che descrivono, sinteticamente, la qualità del terreno circostante la galleria.

Il raggio iniziale della sezione,  $R_1$ , invece, può essere calcolato a partire dalle coordinate dei sensori.

$\epsilon_{CD}$  rappresenta ciò che abbiamo definito come  $\epsilon_\theta$ , che ricaviamo dai nostri sensori.

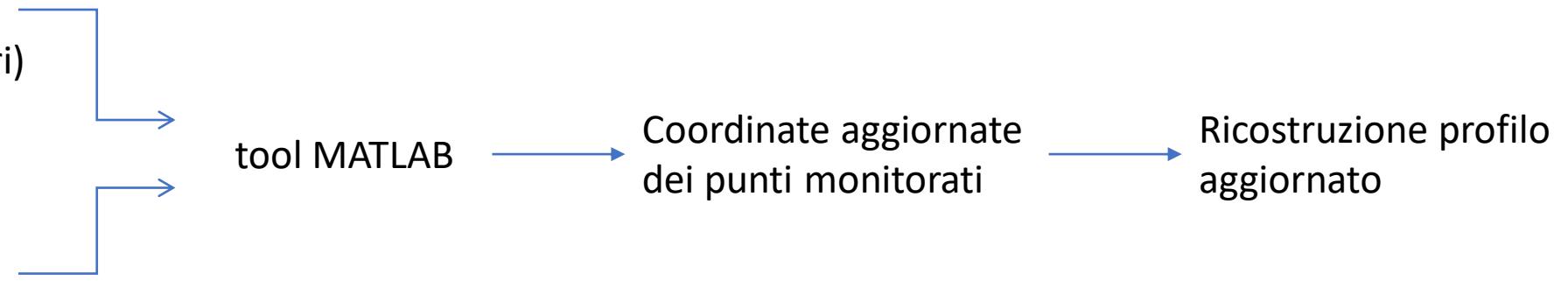
Per ogni sensore, possiamo calcolare la variazione di raggio in quel punto, cioè la convergenza ( $\delta$ ).

**Con 5 valori di deformazione per ogni sezione, possiamo calcolare la convergenza in 5 punti per ogni sezione, ottenendo, quindi, 5 valori di  $\delta$ .**

- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

Coordinate iniziali dei punti monitorati  
(dove sono posizionati i sensori)

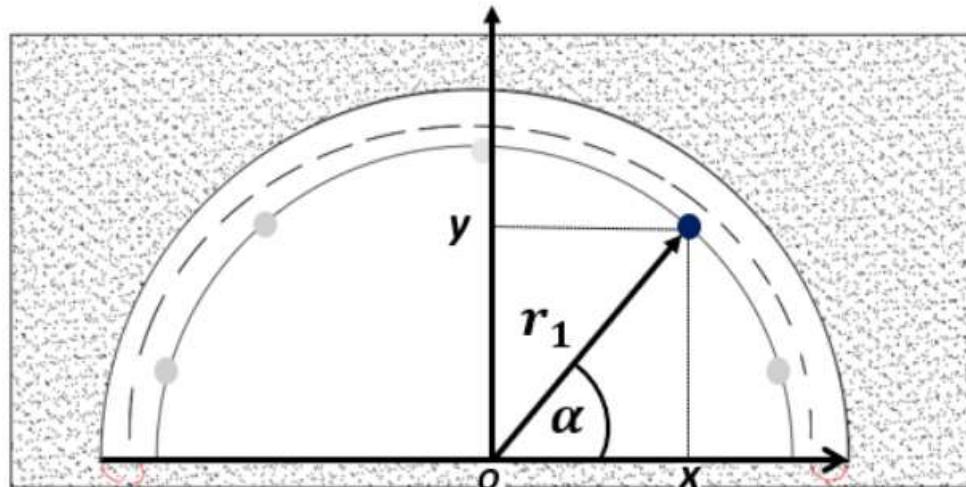
Misura dei  $\delta$  per ciascun punto di misura



- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

## Procedura di calcolo che consente di determinare le coordinate aggiornate dei punti a valle dei 5 δ misurati

Supponiamo di conoscere le coordinate iniziali ( $x, y$ ) della posizione dei sensori e il raggio iniziale della sezione, indicato con  $r_1$ . Per il calcolo delle nuove coordinate, è necessario calcolare gli angoli tra l'asse  $x$  e i diversi punti monitorati, che indichiamo con  $\alpha$



$$\begin{cases} r_1 \cos(\alpha) = x \\ r_1 \sin(\alpha) = y \end{cases} \quad \alpha = \arccos\left(\frac{x}{r_1}\right)$$

Nel caso in cui il centro della circonferenza non si trovi nel punto (0,0):

$$\alpha = \arccos\left(\frac{x - x_c}{r_1}\right)$$

Una volta determinati gli angoli  $\alpha$  per ogni punto, possiamo calcolare le nuove coordinate. In particolare, la convergenza è data dalla differenza tra il raggio originale  $r_1$  e il raggio a seguito di convergenza, che denotiamo  $r_2$ :

$$r_2 = r_1 - \delta$$

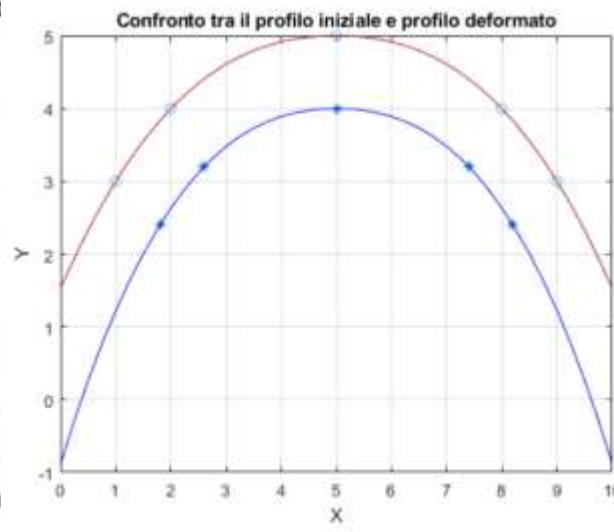
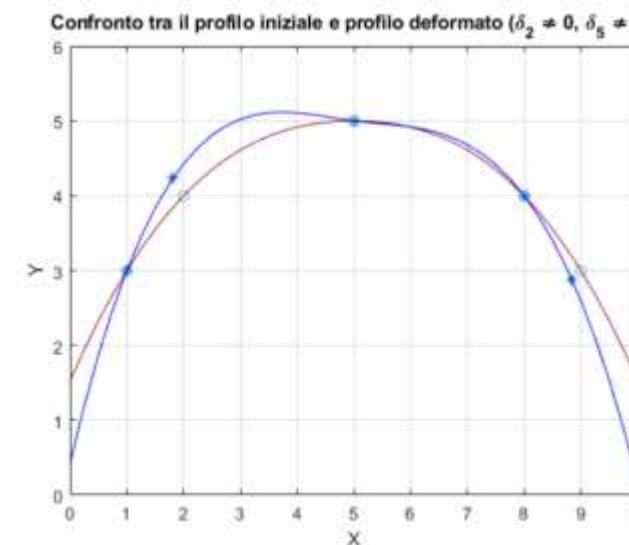
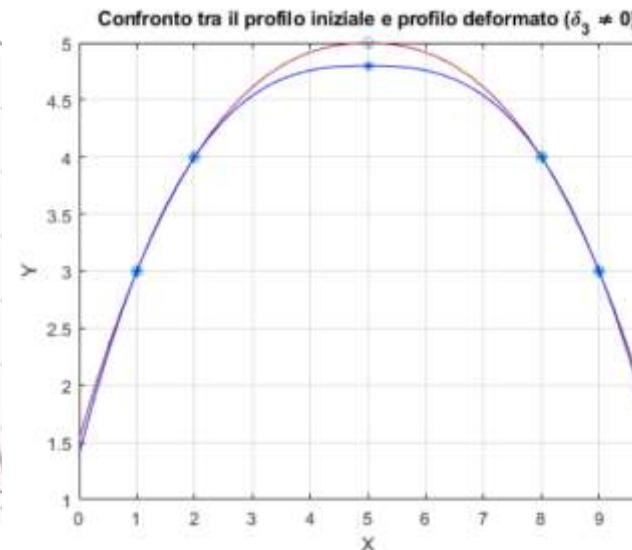
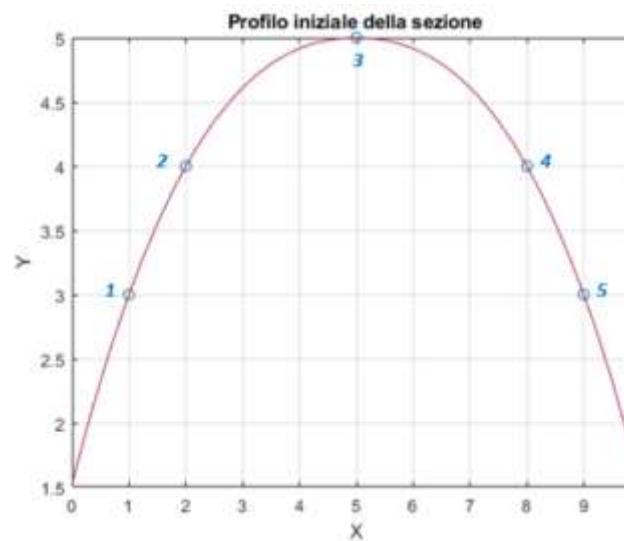
- 1. Convergenza**
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

## Procedura di calcolo che consente di determinare le coordinate aggiornate dei punti a valle dei 5 $\delta$ misurati

Conoscendo, quindi, il raggio  $r_2$ , le coordinate del centro  $(x_c, y_c)$  e l'angolo  $\alpha$ , possiamo ricalcolare le nuove coordinate  $(x', y')$  per ogni punto:

$$\begin{cases} x_c + (r_1 - \delta) \cos(\alpha) = x' \\ y_c + (r_1 - \delta) \sin(\alpha) = y' \end{cases}$$

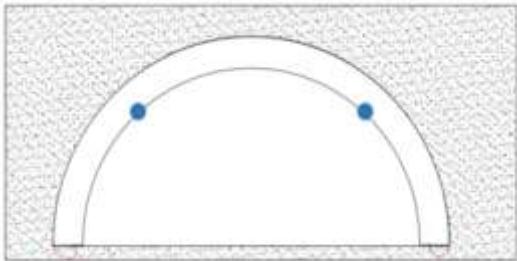
### Alcuni esempi





1. Convergenza
2. **Stato di deformazione longitudinale del cavo**
3. Stato di sollecitazione delle centine
4. Ampiezza delle fessure del rivestimento

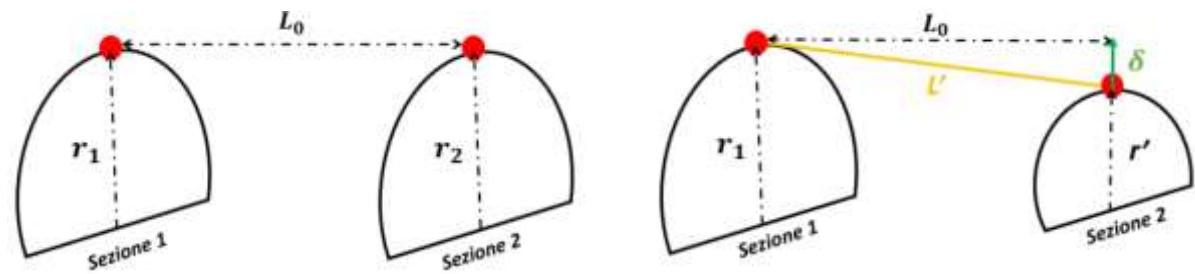
Per la valutazione della deformazione longitudinale del cavo, si possono utilizzare due metodi di analisi differenti.



I sensori di deformazione long-gage, infatti, sono orientati con un angolo di 45° rispetto al vettore normale alla sezione.

Pertanto, a partire dalla deformazione rilevata dai sensori, è possibile calcolare la deformazione longitudinale  $\epsilon_l$  moltiplicando il valore letto dal sensore per il coseno di 45°, ovvero  $\sqrt{2}/2$ .

Come ulteriore analisi è possibile valutare la deformazione longitudinale del cavo tramite convergenza. Dunque, si considerino due sezioni adiacenti



$$L' = \sqrt{L_0^2 + \delta^2}$$

SEZIONI	PARAMETRI MISURATI	OUTPUT
Sezione 1	$\epsilon_{l\_Right1}, \epsilon_{l\_Left1}$	$\epsilon_{l1}$
Sezione 2	$\epsilon_{l\_Right2}, \epsilon_{l\_Left2}$	$\epsilon_{l2}$
Sezione 3	$\epsilon_{l\_Right3}, \epsilon_{l\_Left3}$	$\epsilon_{l3}$
Sezione 4	$\epsilon_{l\_Right4}, \epsilon_{l\_Left4}$	$\epsilon_{l4}$
Sezione 5	$\epsilon_{l\_Right5}, \epsilon_{l\_Left5}$	$\epsilon_{l5}$

1. Convergenza
2. Stato di deformazione longitudinale del cavo
- 3. Stato di sollecitazione delle centine**
4. Ampiezza delle fessure del rivestimento

**Centine coperte** è possibile valutare al massimo la tensione della struttura di rivestimento

In questo caso, non è possibile calcolare direttamente lo stress della centina, ma possiamo valutare lo stato di sollecitazione (o stress) del materiale di rivestimento (ad esempio calcestruzzo o mattoni) nei 5 punti monitorati.

Partendo dal valore di deformazione circonferenziale letto dai sensori, possiamo calcolare lo stress in ogni punto mediante la Legge di Hooke, che lega la deformazione allo stress mediante il modulo elastico del materiale. In funzione del materiale di rivestimento, si fissa il corrispondente modulo elastico e si stima lo stato tensionale circonferenziale secondo la relazione:

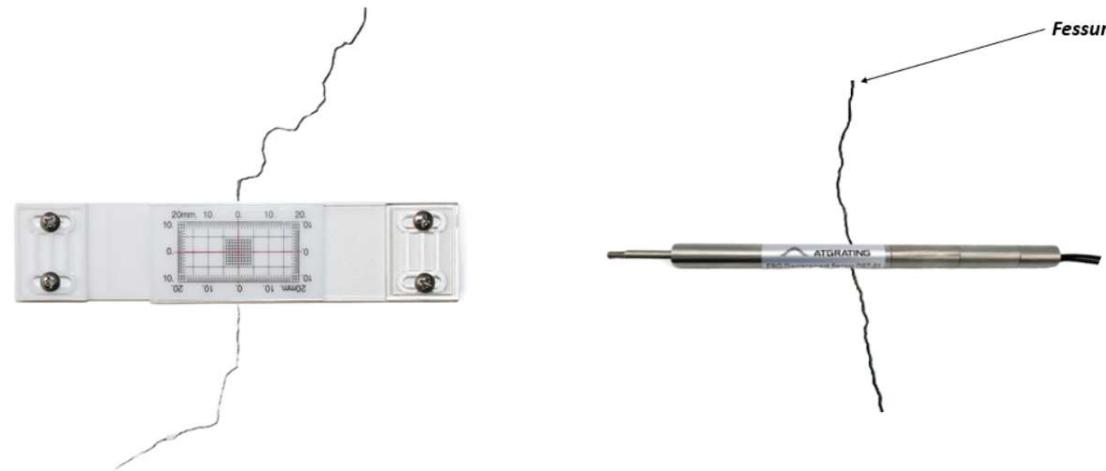
$$\sigma = E \cdot \epsilon$$

1. Convergenza
2. Stato di deformazione longitudinale del cavo
3. Stato di sollecitazione delle centine
- 4. Ampiezza delle fessure del rivestimento**



Non è necessario applicare alcun modello per ottenere il parametro di interesse dalla lettura dei sensori, poiché la lettura è diretta e immediata.

Dopo aver posizionato il sensore di spostamento e aver determinato il valore di riferimento che corrisponde all'ampiezza iniziale della fessura, ogni variazione  $\Delta L$  che viene rilevata dal sensore corrisponde direttamente alla variazione dell'ampiezza della fessura stessa.



# SOMMARIO

## CONVERGENZA

Per ogni sezione, riusciamo a determinare mediante i sensori di deformazione short-gage e la componente circonferenziale dei sensori di deformazione long-gage:

1. Cinque valori di  $\delta$ , uno per ogni punto monitorato.
2. Il profilo aggiornato della sezione, il quale tiene conto dell'eventuale convergenza registrata nei punti monitorati.

## DEFORMAZIONE LONGITUDINALE DEL CAVO

Per quanto riguarda la deformazione longitudinale del cavo, riusciamo a determinare:

1. La misura della deformazione longitudinale del cavo tra due sezioni adiacenti, ottenuta mediante la valutazione della convergenza.

## STATO DI SOLLECITAZIONE DELLE CENTINE

Per quanto riguarda lo stato di sollecitazione delle centine, riusciamo a determinare:

1. Lo stato tensionale della centina nel caso in cui questa sia scoperta da eventuale materiale di rivestimento.
2. Lo stato tensionale del materiale di rivestimento (calcestruzzo o mattoni) nel caso in cui la centina sia coperta.

## AMPIEZZA DELLE FESSURE

Per quanto riguarda l'ampiezza delle fessure, siamo in grado di determinarlo direttamente attraverso i crackmetri. Questi strumenti ci consentono di misurare la variazione di ampiezza delle fessure rispetto allo zero, cioè l'ampiezza della fessura nel momento in cui la strumentazione viene installata

# Railway or Road Tunnels Monitoring: pics from real applications

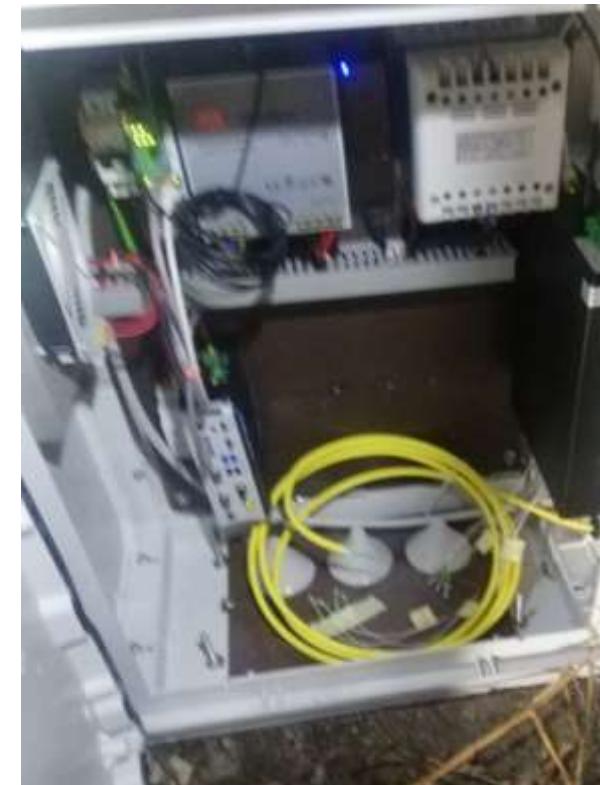
Strain Sensor



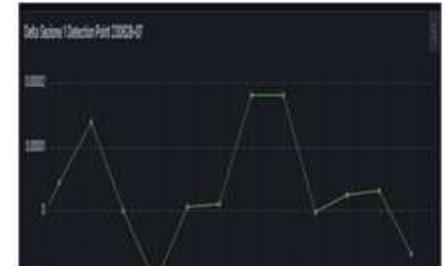
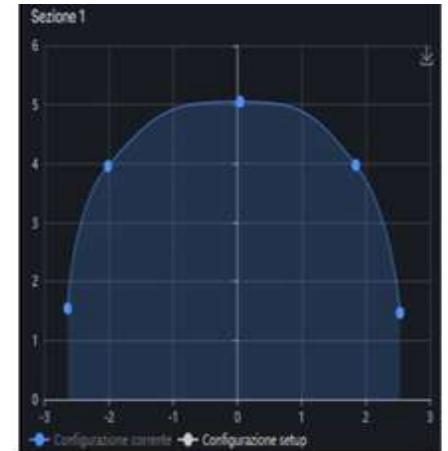
Displacement Sensor



Field Cabinet



Remote Monitoring



# Confronto costi monitoraggio

F.O. vs Wireless



# INDICE

<u>Obiettivo</u>	3
<u>Sensoristica</u>	4-5
<u>Confronto costi</u>	6
<u>Risultati</u>	7-8



# Obiettivo

## IL CASO STUDIO



L'obiettivo di questo studio è effettuare un'analisi dei **costi di installazione e gestione a vita intera** di due sistemi di monitoraggio strutturale di ponti e viadotti messi a confronto per individuare il sistema più vantaggioso:

- Il primo relativo ad una tipologia esistente e di tipo puramente **meccanico** a trasmissione **Wireless**;
- Il secondo di tipo **elettromeccanico** realizzato tramite sensori in **Fibra Ottica**.



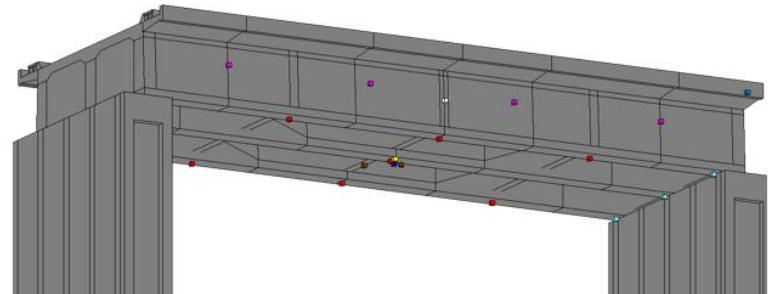
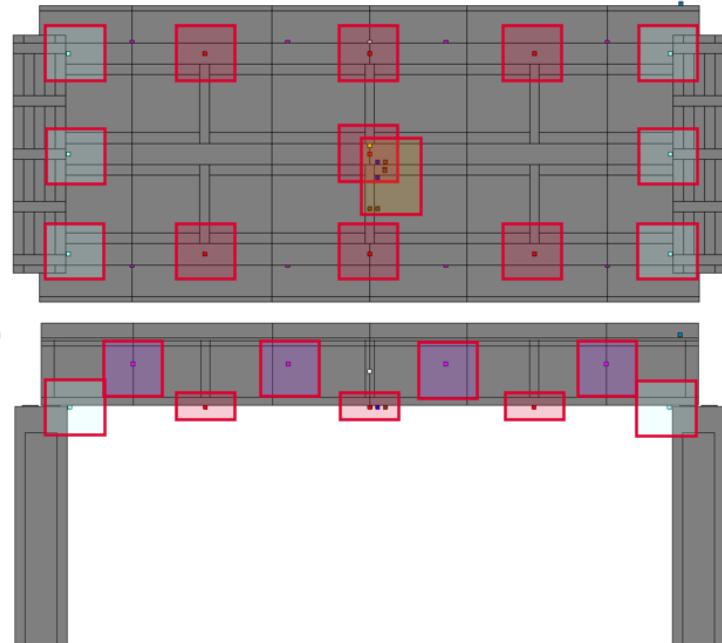
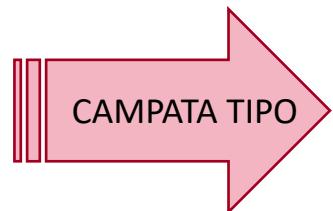
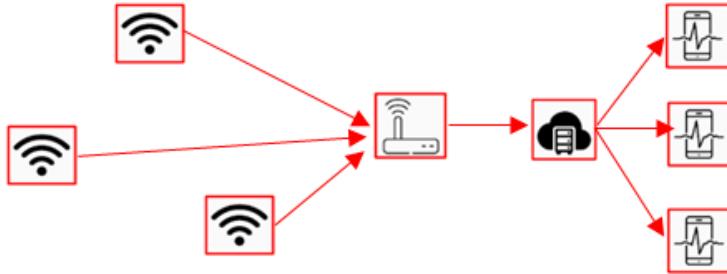
Caso studio è il **Viadotto del Borratino** (linea DD.MA Roma – Firenze), su cui è già avviata la sperimentazione del sistema Wireless.

Il Borratino è un viadotto caratterizzato da **54 campate** a doppio binario costituito da **travate appoggiate di 25 metri di lunghezza** (asse pila) in c.a.p.

In entrambe le valutazioni è stata considerata una delle campate maggiormente attrezzate (n° 17) di lunghezza pari a 25 metri in semplice appoggio.

# La sensoristica

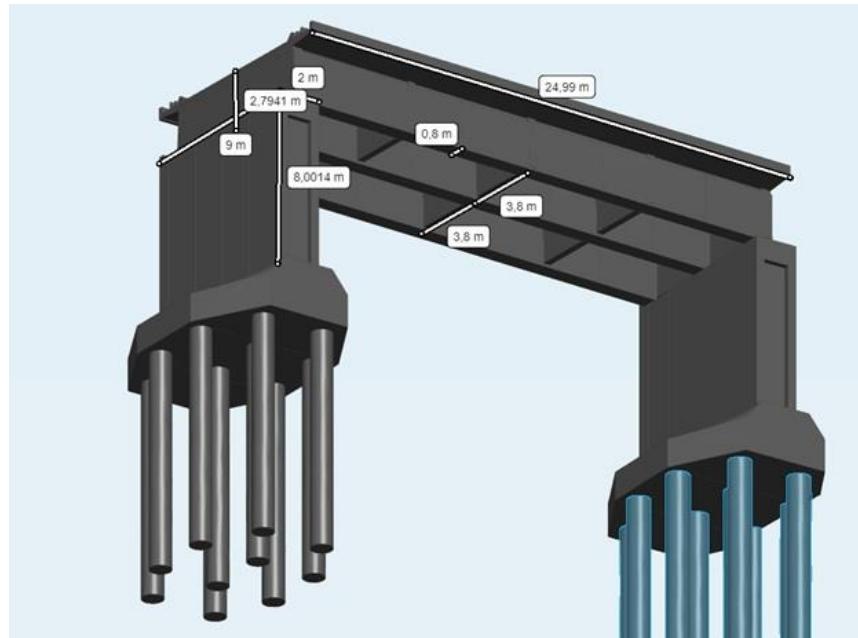
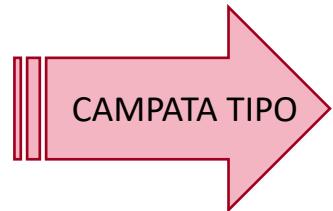
## SENSORI WIRELESS



- 9 Accelerometri triassiali asincroni
- 7 Accelerometri triassiali sincronizzati
- 6 Fessurimetri con corsa 2.5 cm
- 6 Fessurimetri con corsa 20 cm
- 15 Inclinometri
- 2 Sensori di temperatura
- 1 Sensore di umidità.

# La sensoristica

## SENSORI IN F.O.



- 9 Accelerometri triassiali asincroni
- 7 Accelerometri triassiali sincronizzati
- 6 Fessurimetri con corsa 2.5 cm
- 1 Fessurimetro con corsa 15 cm
- 5 Flessimetri
- 15 Inclinometri
- 2 Sensori di temperatura
- 1 Sensore di umidità.

Per garantire coerenza e confrontabilità tra i due sistemi, tra due scenari proposti si è scelto di prendere in considerazione quello che prevede l'utilizzo di **6 fessurimetri da 2,5 cm + 1 fessurimetro da 15 cm affiancato da 5 flessimetri**.

# Confronto costi

## COSTI HARDWARE

Sensoristica di campo					
	Fibra Ottica		Wireless		
	n°	Tot	n°	tot	Differenza %
Accelerometro	16	50.000,00 €	16	10.880,00 €	360%
Inclinometro	15	15.000,00 €	15	7.200,00 €	108%
Fessurimetri	12	17.000,00 €	12	5.400,00 €	215%
Umidità	1	400,00 €	1	300,00 €	33%
Temperatura	2	400,00 €	2	240,00 €	67%
	<b>46</b>	<b>82.800,00 €</b>	<b>46</b>	<b>24.020,00 €</b>	<b>245%</b>

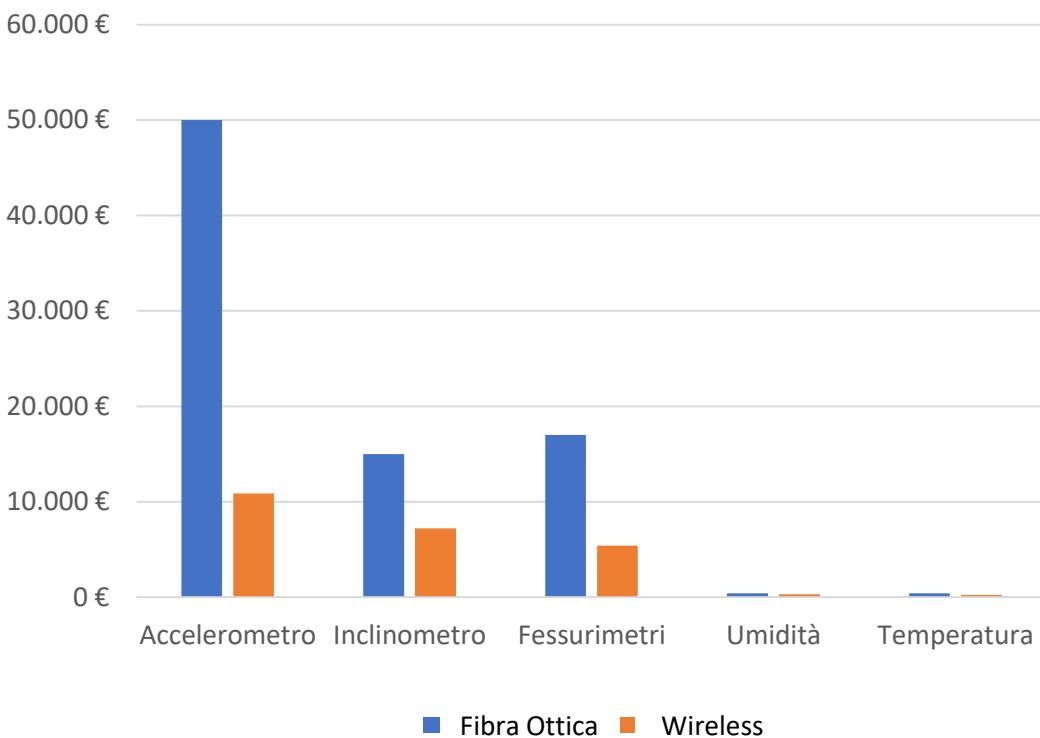
Apparecchiature di sistema					
	Fibra Ottica		Wireless		
	n°	tot	n°	tot	Differenza %
Nodo comunicativo	-	- €	5	2.600,00 €	-
Interrogatore	1	50.000,00 €	-	- €	-
Gateway Nord	-	- €	1	4.870,90 €	-
Gateway Sud	-	- €	1	4.434,00 €	-
Gateway Sentinella	-	- €	1	4.872,20 €	-
Sensore deck	-	- €	5	5.600,00 €	-
Accessori	1	10.960,00 €	-	- €	-
	<b>2</b>	<b>60.960,00 €</b>	<b>13</b>	<b>22.377,10 €</b>	<b>172%</b>

<b>TOTALE</b>	<b>48</b>	<b>143.760,00 €</b>	<b>59</b>	<b>46.397,10 €</b>	<b>210%</b>
---------------	-----------	---------------------	-----------	--------------------	-------------

N.B. per differenza % positiva si intende un costo maggiore per la soluzione in Fibra Ottica

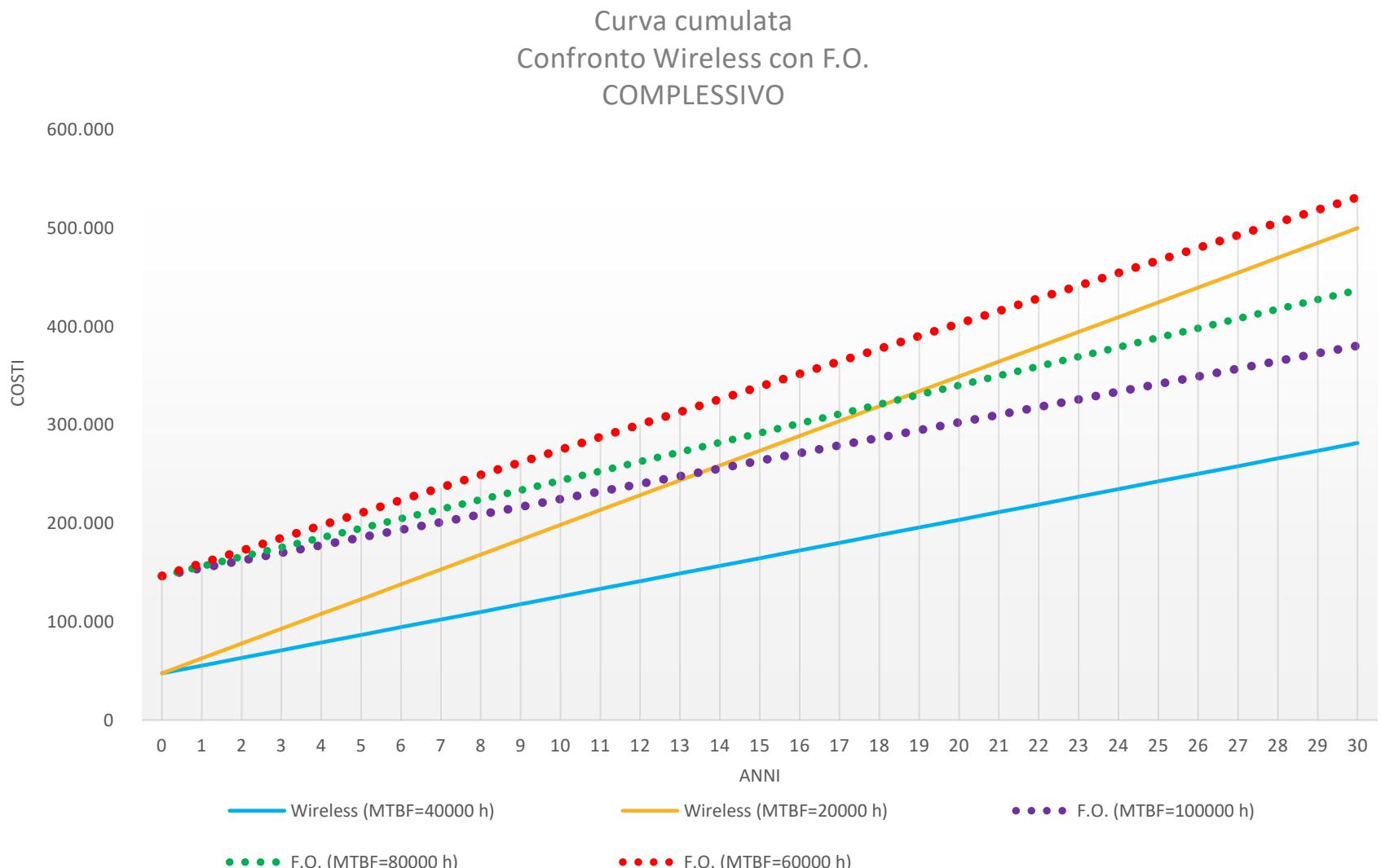
Confronto costi hardware di campo



# I risultati

## COSTI CUMULATI A 30 ANNI – IPOTESI COSTO COMPONENTI INVARIANTE NEL TEMPO

STIMA TEMPO DI RITORNO INVESTIMENTO		
MTBF (h)		ANNI
F.O.	Wireless	
80.000	20.000	<b>18</b>
100.000	20.000	<b>14</b>



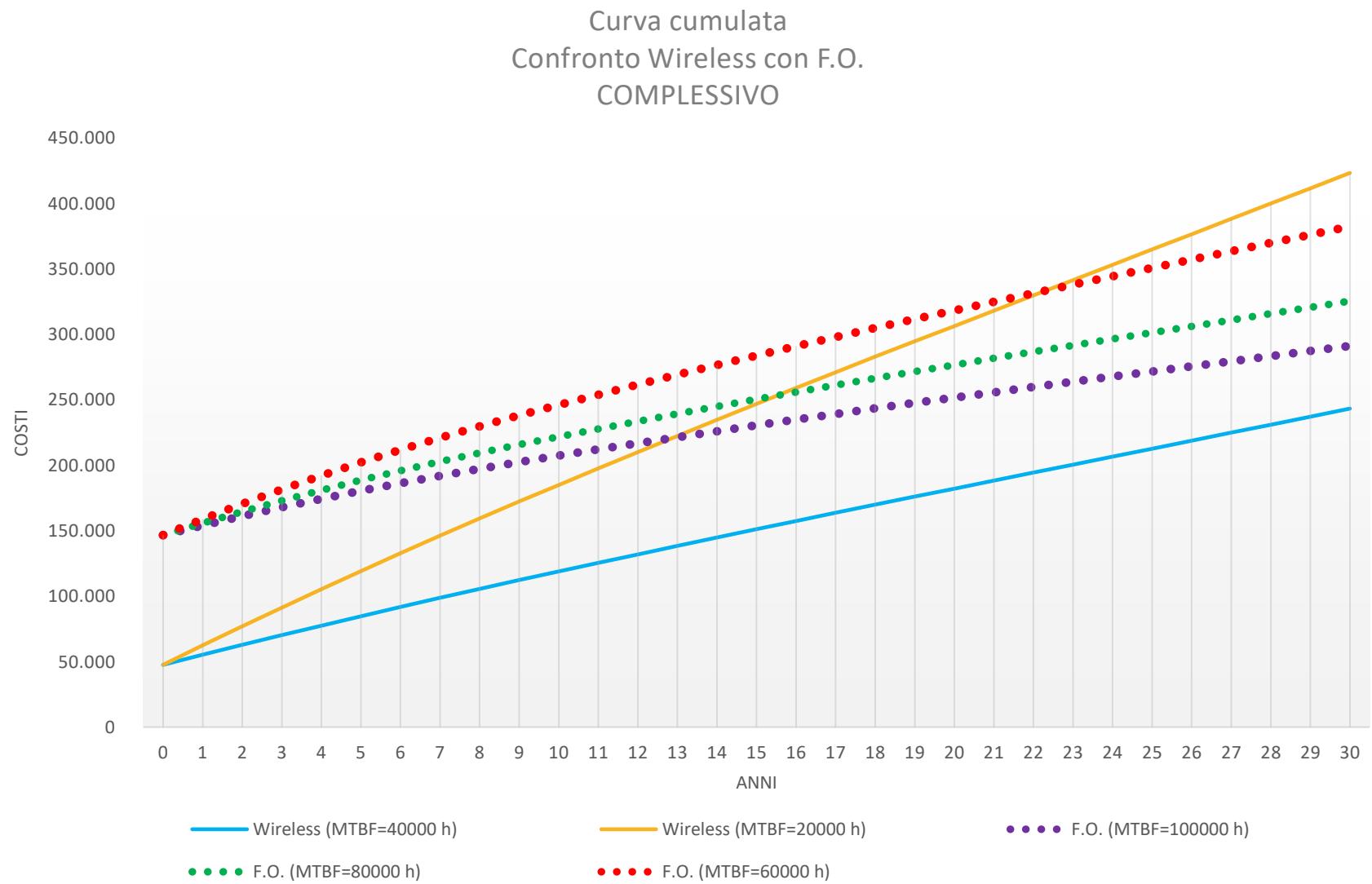
# I risultati

## COSTI CUMULATI A 30 ANNI – IPOTESI RIDUZIONE COSTO COMPONENTI NEL TEMPO

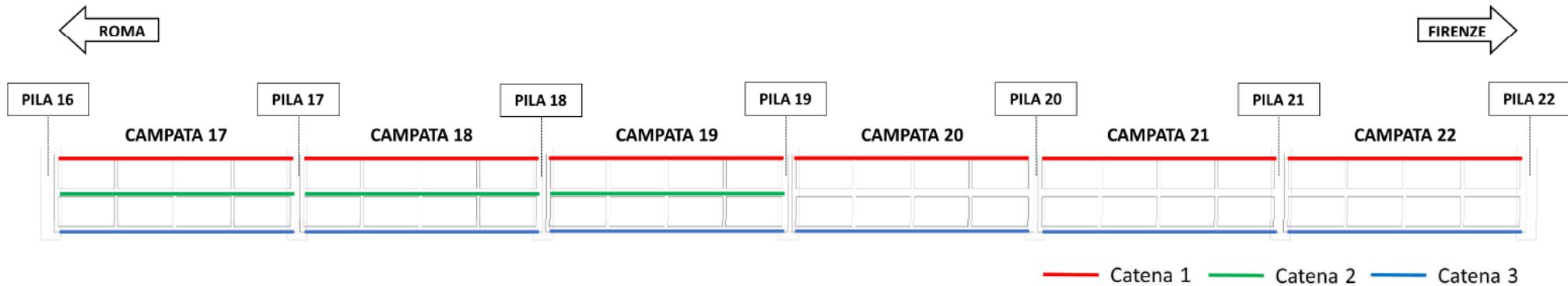
<b>Fibra Ottica</b>			
Riduzione costo/anno			
PRIMI 10 ANNI	DA 11 A 20	DA 20 A 25	>25
5%	2%	1%	0%

<b>Wireless</b>		
Riduzione costo/anno		
PRIMI 10 ANNI	DA 11 A 20	>20
2%	1%	0%

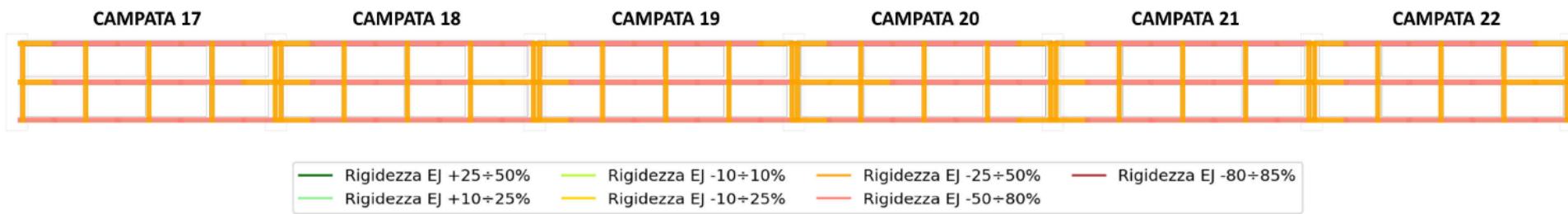
<b>STIMA TEMPO DI RITORNO INVESTIMENTO</b>		
MTBF (h)		ANNI
F.O.	Wireless	
60.000	20.000	<b>22</b>
80.000	20.000	<b>15</b>
100.000	20.000	<b>13</b>



# Attrezzaggio Borratino



Pianta model updating



## Attrezzaggio Borratino

- Monitorare l'affidabilita' del sistema di campo (Sensori+Interrogatore)
- Testare la lunghezza massima senza amplificazione di segnale con il numero massimo di sensori
- Aumentare l'affidabilita' di sistema con la chiusura ad anello della fibra
- Modalita' di montaggio e di intervento
- Skill necessarie per la gestione del sistema (Manutenibilita')
- Modalita' di ingaggio

# Grazie



# Laurea in Ingegneria Elettronica per l'Automazione e le Telecomunicazioni



## Corso di Laboratorio di Elettronica per l'Automazione Caratterizzazione metrologica dei sensori Elaborazione dei dati in Matlab

**Docente del corso:**

Prof. Andrea Cusano

**Assistant:**

Ing. Patrizio Vaiano

Dott.ssa Federica Piccirillo



Università degli Studi del Sannio

Dipartimento di Ingegneria

Gruppo di Optoelettronica

Benevento

ANNO ACCADEMICO 2023/2024

# Parte 1

## Caratterizzazione metrologica dei sensori

# Sensore

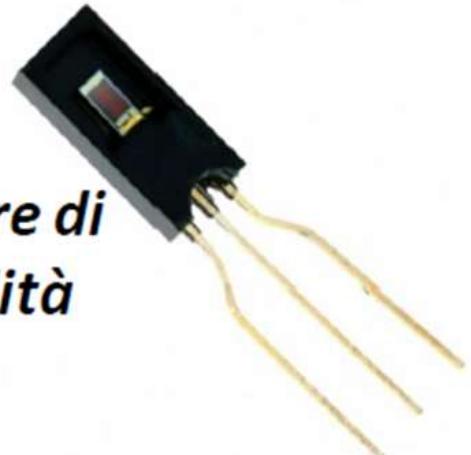
- È un dispositivo che converte una variazione di una grandezza (stimolo) in una variazione di una grandezza elettrica.
- La grandezza di ingresso è di qualsiasi natura, mentre l'uscita è una grandezza elettrica (es. tensione, corrente).

*Fotodiodo*



*Termocoppia*

*Sensore di  
umidità*



# **Caratterizzazione metrologica dei sensori**

**Caratterizzazione e Sensibilità**

**Ripetibilità**

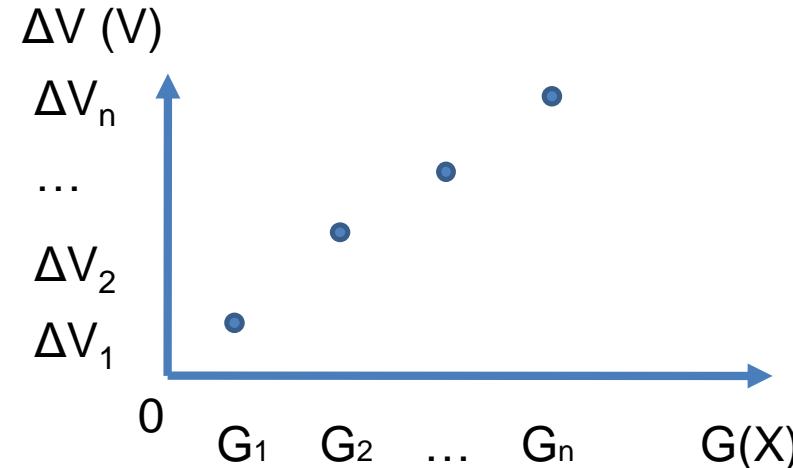
**Isteresi**

# Caratterizzazione e Sensibilità

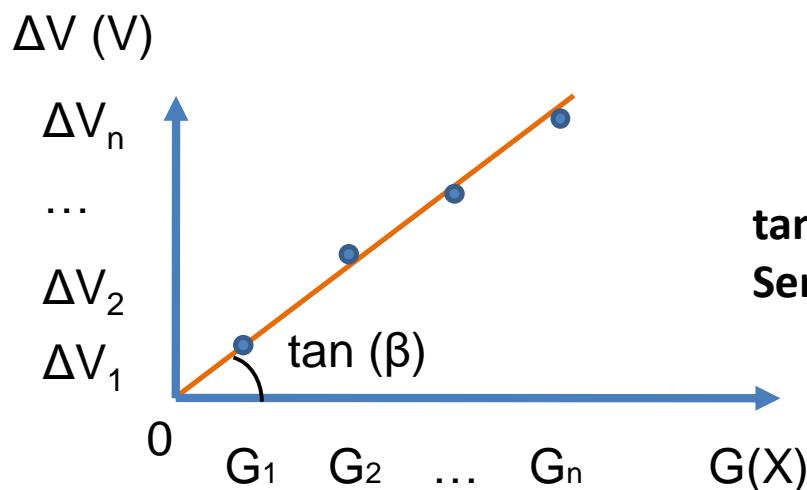
1. Riportare in una tabella le variazioni di tensione ( $\Delta V_k$ ,  $k = 1, \dots, n$ ) corrispondenti a ciascun valore della grandezza applicata ( $G_k$ ,  $k = 1, \dots, n$ )

$G(\text{in})$	$\Delta V(\text{out})$
$G_1$	$\Delta V_1$
$G_2$	$\Delta V_2$
$G_3$	$\Delta V_3$
...	...
$G_n$	$\Delta V_n$

2. Riportare tali valori su un grafico



3. Interpolazione (curva di fitting) e sensibilità



$\tan(\beta)$  = coefficiente angolare =  
Sensibilità del sensore

# Ripetibilità

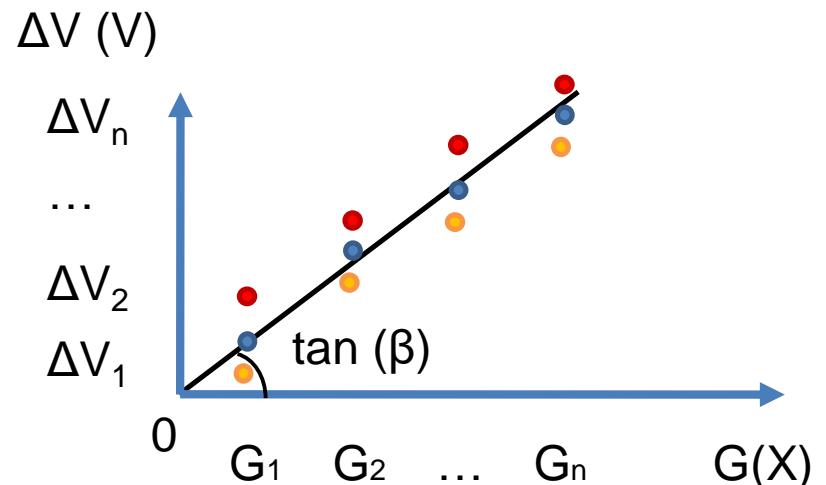
1. Ripetere m volte le misure di tensione ( $\Delta V_{k,l}$ ,  $k = 1, \dots, n$ ,  $l = 1, \dots, m$ ) per ciascun valore della grandezza applicata ( $G_k$ ,  $k = 1, \dots, n$ ) e annotare i valori ottenuti in una tabella.

	M1	M2	M...	Mm	Media
G(in)	$\Delta V(\text{out})$	$\Delta V(\text{out})$	...	$\Delta V(\text{out})$	$\Delta V(\text{out})$
$G_1$	$\Delta V_{1,1}$	$\Delta V_{1,2}$	...	$\Delta V_{1,m}$	$\Delta V_{1,\text{mean}}$
$G_2$	$\Delta V_{2,1}$	$\Delta V_{2,2}$	...	$\Delta V_{2,m}$	$\Delta V_{2,\text{mean}}$
$G_3$	$\Delta V_{3,1}$	$\Delta V_{3,2}$	...	$\Delta V_{3,m}$	$\Delta V_{3,\text{mean}}$
...	...	...	...	...	...
$G_n$	$\Delta V_{n,1}$	$\Delta V_{n,2}$	...	$\Delta V_{n,m}$	$\Delta V_{n,\text{mean}}$

2. Calcolare la media delle variazioni di tensione ( $\Delta V_{k,\text{mean}}$ ,  $k = 1, \dots, n$ ) per ogni valore della grandezza applicata ( $G_k$ ) e la deviazione standard relativa  $\text{RSD}_k = 100 \times \text{SD}_k / \Delta V_{k,\text{mean}}$ ,  $k = 1, \dots, n$ .

3. Valutare la sensibilità relativa alla media delle misure di tensione effettuate e rappresentare graficamente la media e la deviazione standard relativa.
4. Calcolare infine la deviazione standard totale, come media delle deviazioni standard su tutte le misure eseguite:

$$(1/n) \times \sum_{i=1:n} (\text{RSD}_i)$$



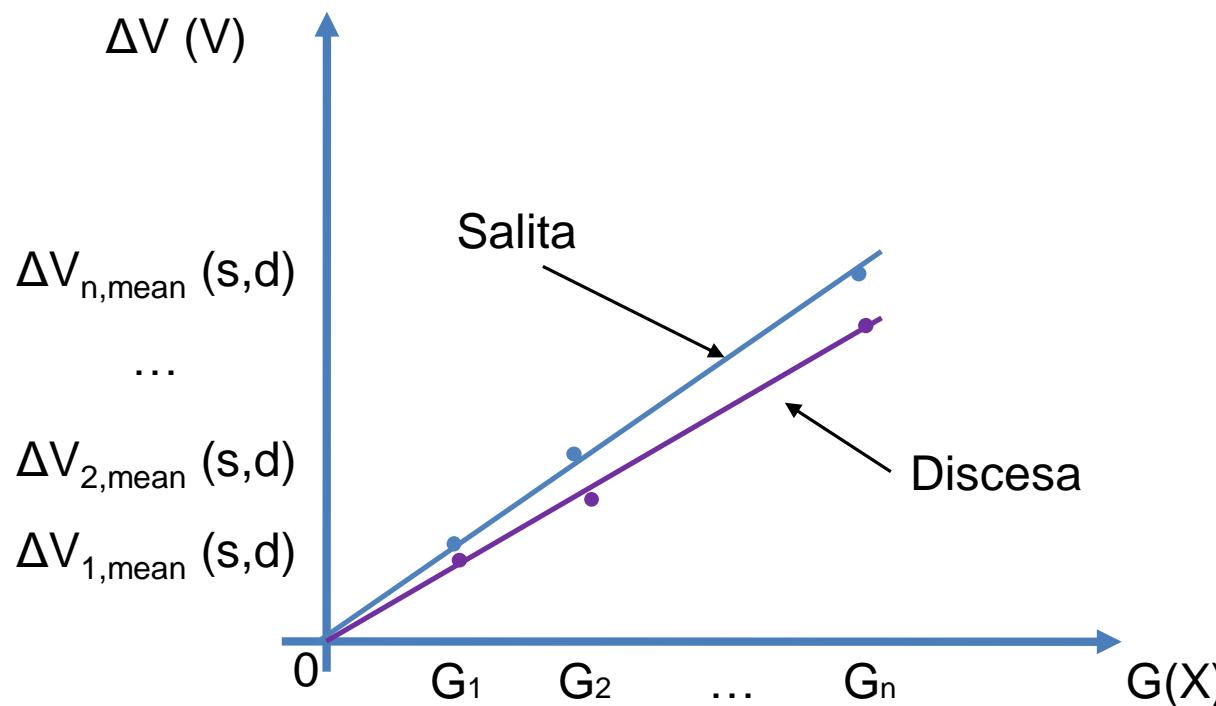
# Isteresi

1. Ripetere  $m$  volte le misure di tensione in salita e in discesa per ciascun valore della grandezza applicata e annotare i valori ottenuti in una tabella.
2. Valutare la media delle misure in salita e in discesa e rappresentarle graficamente.
3. Valutare per ogni valore della grandezza applicata l'errore di isteresi (%) secondo la formula:

$$I_k = 100 \times |\Delta V_{k,\text{mean}(s)} - \Delta V_{k,\text{mean}(d)}| / \text{mean}[\Delta V_{k,\text{mean}(s)}, \Delta V_{k,\text{mean}(d)}]$$

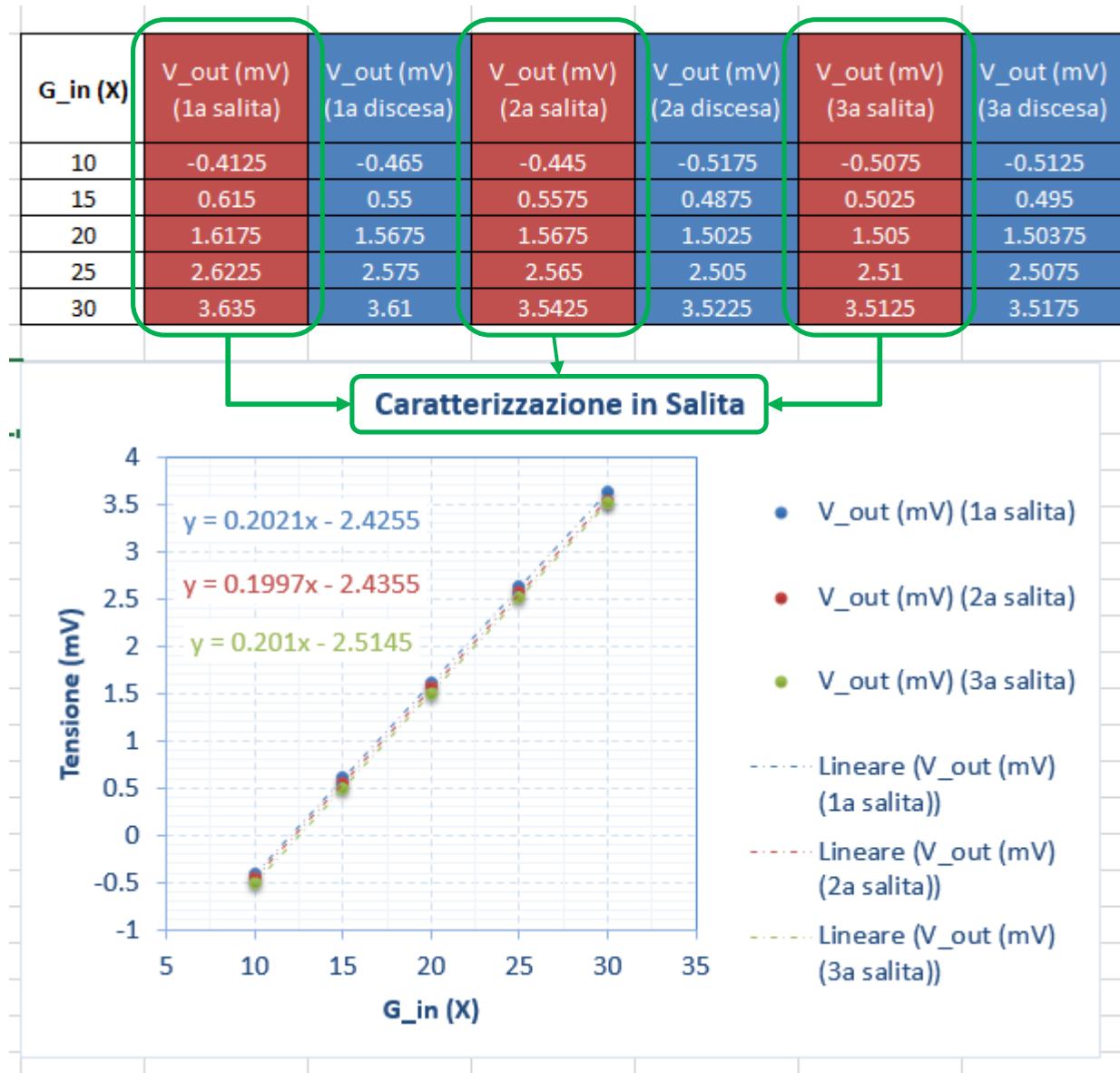
4. Valutare l'errore di isteresi totale:

$$I_{\text{tot}} = (1/n) \times \sum_{i=1:n} (I_i)$$



	Media Salita	Media Discesa	Isteresi
G(in)	ΔV(out)	ΔV(out)	ΔV(out)
G1	ΔV <sub>1,mean(s)</sub>	ΔV <sub>1,mean(d)</sub>	ΔV <sub>1,hys</sub>
G2	ΔV <sub>2,mean(s)</sub>	ΔV <sub>2,mean(d)</sub>	ΔV <sub>2,hys</sub>
G3	ΔV <sub>3,mean(s)</sub>	ΔV <sub>3,mean(d)</sub>	ΔV <sub>3,hys</sub>
...	...	...	...
Gn	ΔV <sub>n,mean(s)</sub>	ΔV <sub>n,mean(d)</sub>	ΔV <sub>n,hys</sub>

# Esempio: misure ripetute in salita e discesa, fitting



1. Ripetere m volte le misure di tensione ( $\Delta V_{k,l}$ ,  $k = 1, \dots, n$ ,  $l = 1, \dots, m$ ) per ciascun valore della grandezza applicata ( $G_k$ ,  $k = 1, \dots, n$ ) e annotare i valori ottenuti in una tabella

2. Riportare tali valori su un grafico

3. Interpolazione (curva di fitting) e sensibilità

Analogamente per la discesa

# Esempio: media e deviazione standard

G_in (X)	V_out (mV) (1a salita)	V_out (mV) (1a discesa)	V_out (mV) (2a salita)	V_out (mV) (2a discesa)	V_out (mV) (3a salita)	V_out (mV) (3a discesa)	V_out_media (mV) (Salita)	V_out_media (mV) (Discesa)
10	-0.4125	-0.465	-0.445	-0.5175	-0.5075	-0.5125	-0.455	-0.498
15	0.615	0.55	0.5575	0.4875	0.5025	0.495	0.558	0.511
20	1.6175	1.5675	1.5675	1.5025	1.505	1.50375	1.563	1.525
25	2.6225	2.575	2.565	2.505	2.51	2.5075	2.566	2.529
30	3.635	3.61	3.5425	3.5225	3.5125	3.5175	3.563	3.550

1. Ripetere m volte le misure di tensione ( $\Delta V_{k,l}$ ,  $k = 1, \dots, n$ ,  $l = 1, \dots, m$ ) per ciascun valore della grandezza applicata ( $G_k$ ,  $k = 1, \dots, n$ ) e annotare i valori ottenuti in una tabella
2. Calcolare la media delle variazioni di tensione ( $\Delta V_{k,\text{mean}}$ ,  $k = 1, \dots, n$ ) per ogni valore della grandezza applicata ( $G_k$ ) e la deviazione standard relativa  $RSD_k = 100 \times SD_k / \Delta V_{k,\text{mean}}$ ,  $k = 1, \dots, n$ .
3. Calcolare infine la deviazione standard totale, come media delle deviazioni standard su tutte le misure eseguite:

$$(1/n) \times \sum_{i=1:n} (RSD_i)$$

Dev Std Salita	Dev Std Discesa	RSD (%) Salita	RSD (%) Discesa
0.048	0.029	10.612	5.814
0.056	0.034	10.075	6.680
0.056	0.037	3.605	2.438
0.056	0.040	2.192	1.570
0.064	0.052	1.792	1.465

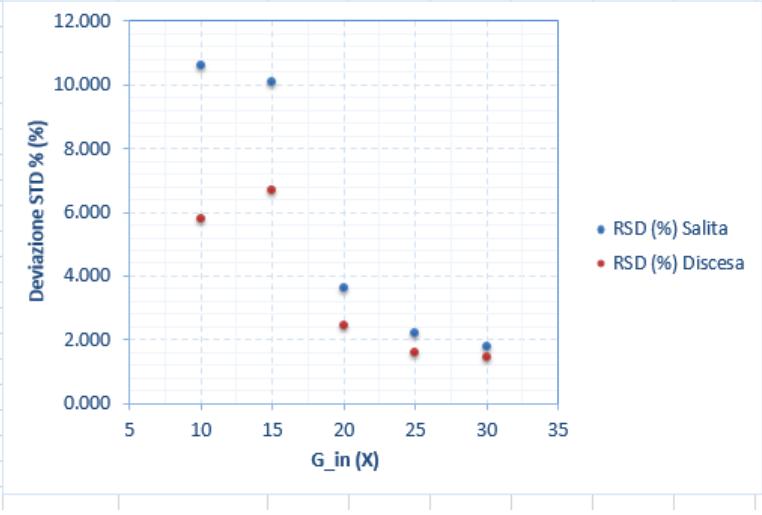
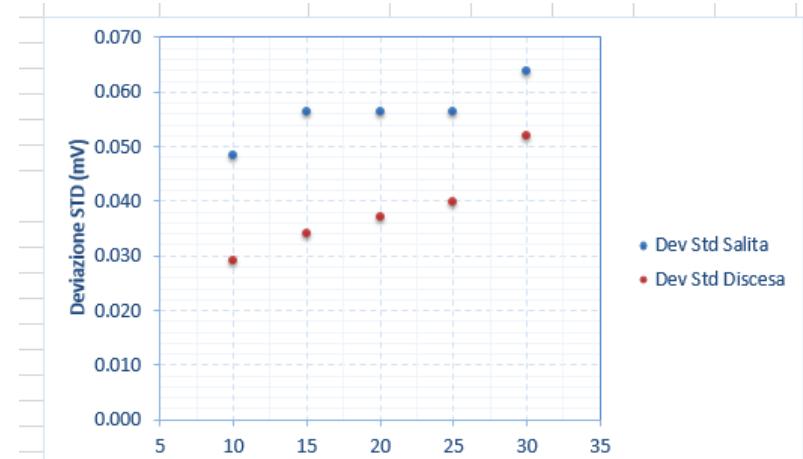
Analogamente per la discesa

# Esempio: deviazione standard



Dev Std Salita	Dev Std Discesa
0.048	0.029
0.056	0.034
0.056	0.037
0.056	0.040
0.064	0.052

RSD (%) Salita	RSD (%) Discesa
10.612	5.814
10.075	6.680
3.605	2.438
2.192	1.570
1.792	1.465



# Esempio: media complessiva e isteresi

V_out_media (mV) (Salita)	V_out_media (mV) (Discesa)
-0.455	-0.498
0.558	0.511
1.563	1.525
2.566	2.529
3.563	3.550

G_in (X)	Media Totale (mV)
10	-0.477
15	0.535
20	1.544
25	2.548
30	3.557

Isteresi (%)
9.091
8.885
2.510
1.439
0.375

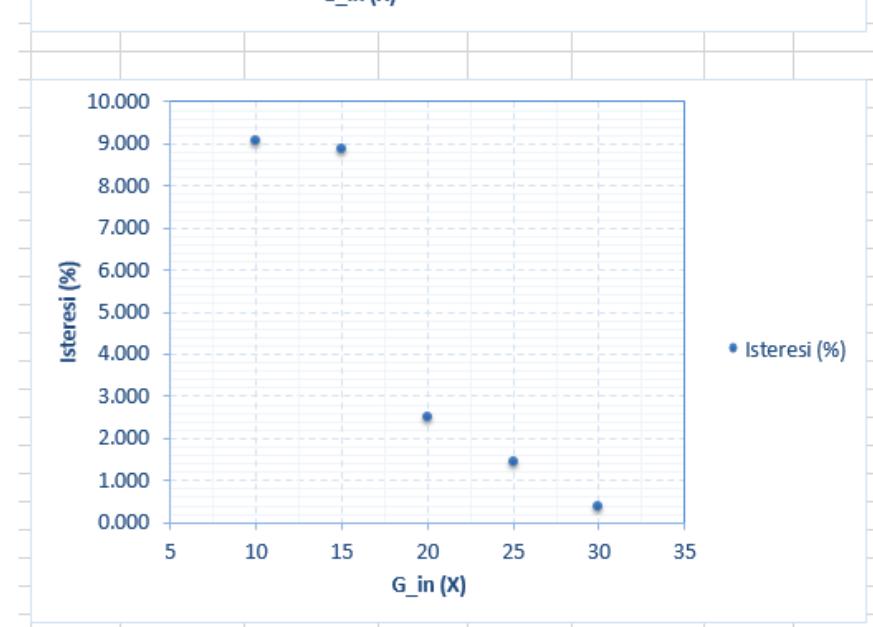
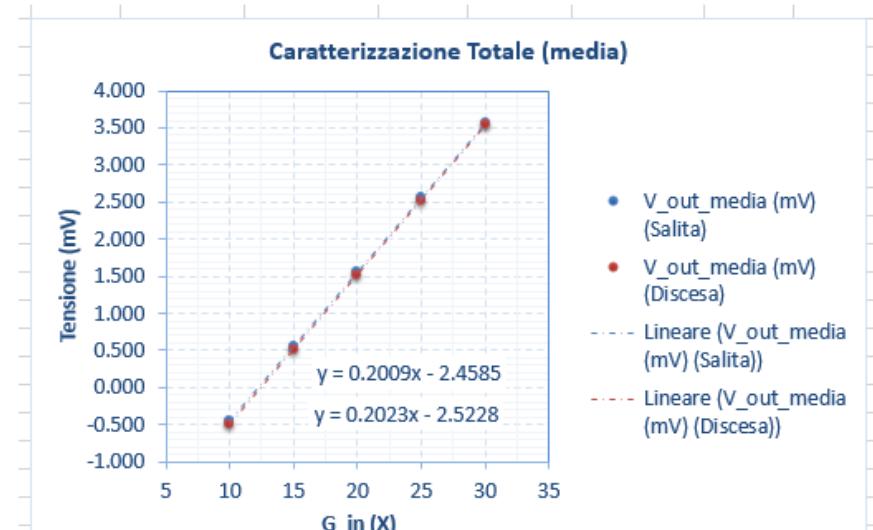
1. Valutare la media delle misure in salita e in discesa e rappresentarle graficamente.

2. Valutare per ogni valore della grandezza applicata l'errore di isteresi (%) secondo la formula:

$$I_k = 100 \times |\Delta V_{k,\text{mean}(s)} - \Delta V_{k,\text{mean}(d)}| / \text{mean}[\Delta V_{k,\text{mean}(s)}, \Delta V_{k,\text{mean}(d)}]$$

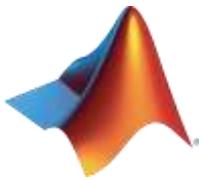
3. Valutare l'errore di isteresi totale:

$$I_{\text{tot}} = (1/n) \times \sum_{i=1:n} (I_i)$$



## Parte 2

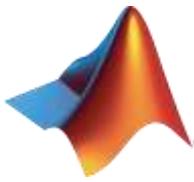
# Introduzione all'uso di Matlab



# Introduzione all'uso di Matlab

- Matlab per uso accademico attraverso email unisannio
- Getting started with Matlab
- Definizione di variabili, vettori, matrici
- Operazioni tra vettori e matrici
- Cicli if, for, while
- Come importare file

# Matlab for academic use



<https://it.mathworks.com/>

## MathWorks Account Creation

- Creare un MathWorks account

## Associate with the license

- Una volta inserite le proprie credenziali cliccare su “Manage Licenses” e poi su “Add License”:
- Selezionare “Activation Key” e cliccare su Continue - Enter the Activation Key

A questo punto è possibile utilizzare Matlab online e scaricare i file di installazione:

Download the products

- Nella pagina del proprio account cliccare su “Get Licensed Product and Updates”.

The screenshot shows the MathWorks homepage with the navigation menu: Products & Services, Solutions, Academia (which is highlighted), Support, User Community, Events, and Company. The top right corner features a "Create Account" button with a red border and a "Log In" link.

Usare l'email dell'università ([n.cognome@studenti.unisannio.it](mailto:n.cognome@studenti.unisannio.it)) e selezionare l'opzione per uso accademico

The screenshot shows a dropdown menu for selecting software usage. The options are: select how you will use your software – Commercial use, Government use, and Academic use (including campus/site license use). The "Academic use" option is selected and highlighted with a blue background. A red arrow points to this option with the text "Select this option". Below the menu, it says "Student use of MATLAB® and Simulink® Student Version".

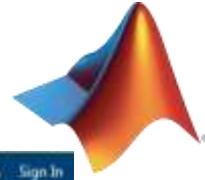
- Online Documentation:

<http://www.mathworks.com/help/>

- MATLAB Answers:

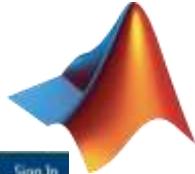
<http://www.mathworks.com/matlabcentral/answers/>

# AVVIO DI MATLAB



The image shows the MATLAB interface. At the top is the menu bar with tabs: HOME, PLOTS, APPS, FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, and RESOURCES. The FILE tab is selected, showing options like New Script, New Live Script, New, Open, Find Files, Import Data, Workspace, New Variable, Open Variable, Favorites, Analyze Code, Preferences, Run and Time, Clear Commands, Simulink, Layout, Set Path, Add-Ons, Help, Parallel, Community, Request Support, Learn MATLAB, and Sign In. Below the menu is the toolbar with icons for New, Open, Save, Import Data, Workspace, New Variable, Open Variable, Favorites, Analyze Code, Preferences, Run and Time, Clear Commands, Simulink, Layout, Set Path, Add-Ons, Help, Parallel, Community, Request Support, Learn MATLAB, and Sign In. The main workspace includes the Current Folder browser (C:\Users\FedericaP\Documents\MATLAB), Command Window (with a message about getting started and a prompt f1 >>), and Workspace browser (empty). A Details browser at the bottom left says "Select a file to view details". The status bar at the bottom indicates "Ready".

# AVVIO DI MATLAB



A screenshot of the MATLAB desktop interface. The menu bar includes HOME, PLOTS, APPS, FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, and RESOURCES. The toolbar below the menu bar includes icons for New Script, New Live Script, New, Open, Find Files, Import Data, Workspace, New Variable, Open Variable, Favorites, Analyze Code, Run and Time, Clear Commands, Simulink, Layout, Preferences, Set Path, Add-Ons, Help, Parallel, Community, Request Support, Learn MATLAB, and Sign In. The main workspace contains three windows: 'Current Folder' (blue border), 'Command Window' (orange border), and 'Workspace' (green border). The 'Current Folder' window shows the path C:\Users\ FedericoP\Documents\ MATLAB. The 'Command Window' has the text 'f1 &gt;&gt;' and a message 'New to MATLAB? See resources for Getting Started.' The 'Workspace' window shows a table with columns 'Name' and 'Value'. A status bar at the bottom indicates 'Ready'.

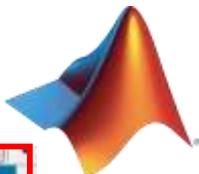
## CURRENT FOLDER:

percorso utilizzato da Matlab per risalire ai programmi da eseguire

**COMMAND WINDOW:**  
area dove si possono inserire i comandi da far eseguire

## WORKSPACE:

spazio in cui vengono salvate le variabili definite in fase di lavoro



# GETTING STARTED

Documentation Search Help

All Examples Functions

## Get Started with MATLAB

The Language of Technical Computing

Millions of engineers and scientists worldwide use MATLAB® to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems.

### Tutorials

**Desktop Basics**  
Enter statements at the command line and view results.

**Matrices and Arrays**  
MATLAB operates primarily on arrays and matrices, both in whole and in part. A matrix is a two-dimensional array often used for linear algebra.

**Array Indexing**  
Variables in MATLAB are typically arrays that can hold many numbers. When you want to access selected elements of an array, use indexing.

**Workspace Variables**  
The workspace contains variables that you create within or import into MATLAB from data files or other programs.

**Text and Characters**  
Create string arrays for text, or create character arrays for data.

**Calling Functions**  
MATLAB provides a large number of functions that perform computational tasks. To call a function, enclose its input arguments in parentheses.

**2-D and 3-D Plots**  
Graphics functions include 2-D and 3-D plotting functions to visualize data and communicate results.

**Programming and Scripts**  
The simplest type of MATLAB program is called a script. A script contains a sequence of commands and function calls.

**Help and Documentation**  
All functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.

### Online Learning

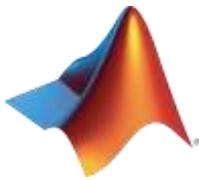
**MATLAB Onramp**  
Free two-hour online MATLAB course

**Videos**

**Getting Started with MATLAB**  
Get an overview of MATLAB, the language of technical computing.

**Working in the Development Environment**  
Access tools such as the command history workspace browser and variable editor, save and load your workspace data, and manage windows and desktop layout.

`>> help`



# Variabili

```
>> ans =
```

```
10
```



Variabile generata automaticamente quando un'espressione  
non è assegnata ad una variabile definita dall'utente

- In Matlab, per definire una nuova variabile basta decidere il nome e assegnarle un valore numerico:

```
>> x = 10;
```

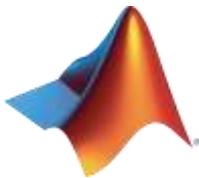
- Matlab è case-sensitive (quindi attenzione a nomi che usate) e si possono usare massimo 31 caratteri.
- Mettendo il ; non compare il valore sulla command window ma solo nel workspace.
- Matlab esegue i calcoli e memorizza le variabili (scalari, vettori o matriciali) in doppia precisione.

```
>> who
```

Elenca i nomi delle variabili attualmente presenti nel workspace

```
>> whos
```

All'elenco delle variabili nel workspace, aggiunge info sul tipo di variabile e sull'occupazione di memoria



# Operazioni tra scalari

```
>> a = 3;  
>> b = 8;
```

**Somma**

```
>> c = a + b
```

c =

11

**Differenza**

```
>> d = a - b
```

d =

-5

**Prodotto**

```
>> e = a * b
```

e =

24

**Rapporto**

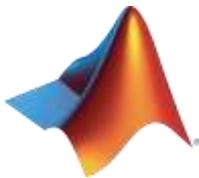
```
>> f = a/b
```

f =

0.3750

**Priorità**

- 1. Espressioni contenute tra parentesi
- 2. Elevamento a potenza (istruzione:  $\wedge$ )
- 3. Moltiplicazioni e divisioni (valutandole da sinistra a destra)
- 4. Somme e sottrazioni (valutandole da sinistra a destra)



# Funzioni & Comandi

## Funzioni

Necessita di parametri in  
Ingresso tra parentesi tonde

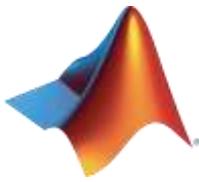
## Comandi

Utilizzato con almeno uno spazio  
tra il nome  
del comando stesso e il  
suo argomento

Funzione	Significato
sin	seno
cos	coseno
asin	arcoseno
acos	arcocoseno
tan	tangente
atan	arcotangente
exp	esponenziale
log	Logaritmo naturale, base e
log2	Logaritmo in base 2
log10	Logaritmo in base 10
sqrt	Radice quadrata
abs	Valore assoluto
sign	La funzione segno

- **clear all:** pulisce il Workspace, cancellando tutte le variabili salvate;
- **close all:** chiude i grafici aperti;
- **clc:** pulisce i comandi scritti nella Command Window.
- Il simbolo **%** o **%%** permette di leggere ciò che lo segue non come comando da eseguire ma come commento, ad esempio unità di misura.
- **ctrl+r:** permette di commentare una riga di testo in uno script;
- **ctrl+t:** permette di de-commentare una riga di testo in uno script;
- **ctrl+c:** permette di interrompere l'esecuzione del programma.

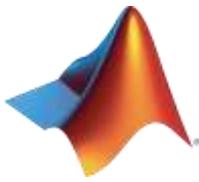
**N.B.:** quando si inizia a risolvere un nuovo problema è buona norma cancellare tutte le variabili presenti nel Workspace per evitare inutili errori di riutilizzo o sovrapposizione di variabili già definite precedentemente.



# M-files

Gli M-file hanno estensione .m e sono utilizzati per scrivere sequenze di comandi (script e funzioni); essendo file ASCII (American Standard Code for Information Interchange) possono essere scritti utilizzando qualsiasi editor di testi. Matlab, tuttavia, prevede un editor molto utile che include funzioni di correzione (debug).

1. **Script:** Sequenza di comandi con variabili globali (restano nel workspace)
2. **Funzione:** Sequenza di comandi con variabili locali (i loro valori sono disponibili solo all'interno della funzione), tipicamente usate per I/O



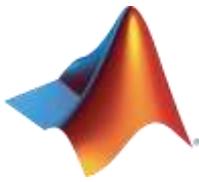
# File Script

I valori delle variabili create durante la loro esecuzione sono caricate nella sessione di lavoro e pertanto rimangono in memoria fino a quando non vengono esplicitamente cancellate attraverso un comando. Tali variabili sono definite “globali”. Il comando **whos** consente di visualizzare le variabili nel workspace.

## ATTENZIONE!

Tutte le variabili create in un file di script vengono aggiunte al workspace. Questo può avere effetti indesiderati, perché le variabili già esistenti nel workspace possono essere sovrascritte.

Di conseguenza, è meglio codificare le applicazioni più complesse usando la funzione piuttosto che lo script.



# Array & Tipi di dati in Matlab

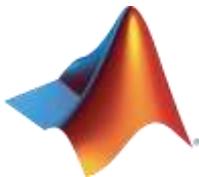
Ogni variabile è per Matlab una struttura di **tipo array**, dove per array si intende un insieme di valori ordinati, secondo uno o più indici, cui si riferisce un singolo nome variabile. (*Gli scalari sono matrici 1x1*).

*Array ad un indice -> vettore*

*Array a due indici -> matrice*

I tipi di variabile supportati in Matlab sono:

Type	Description
<code>char</code>	Character array
<code>complex</code>	Complex data. Cast function takes real and imaginary components
<code>double</code>	Double-precision floating point
<code>int8, int16, int32, int64</code>	Signed integer
<code>logical</code>	Boolean true or false
<code>single</code>	Single-precision floating point
<code>struct</code>	Structure
<code>uint8, uint16, uint32, uint64</code>	Unsigned integer
Fixed-point	Fixed-point data types



# Vettori

- Un **vettore riga** può essere definito tra parentesi quadre separando i singoli elementi da uno spazio oppure da una virgola:

```
>> v_riga = [1 3 5 7 9 11 13 15];
```

- Un **vettore colonna** può essere definito tra parentesi quadre separando i singoli elementi da un punto e virgola:

```
>> v_colonna = [1; 3; 5; 7; 9; 11; 13; 15];
```

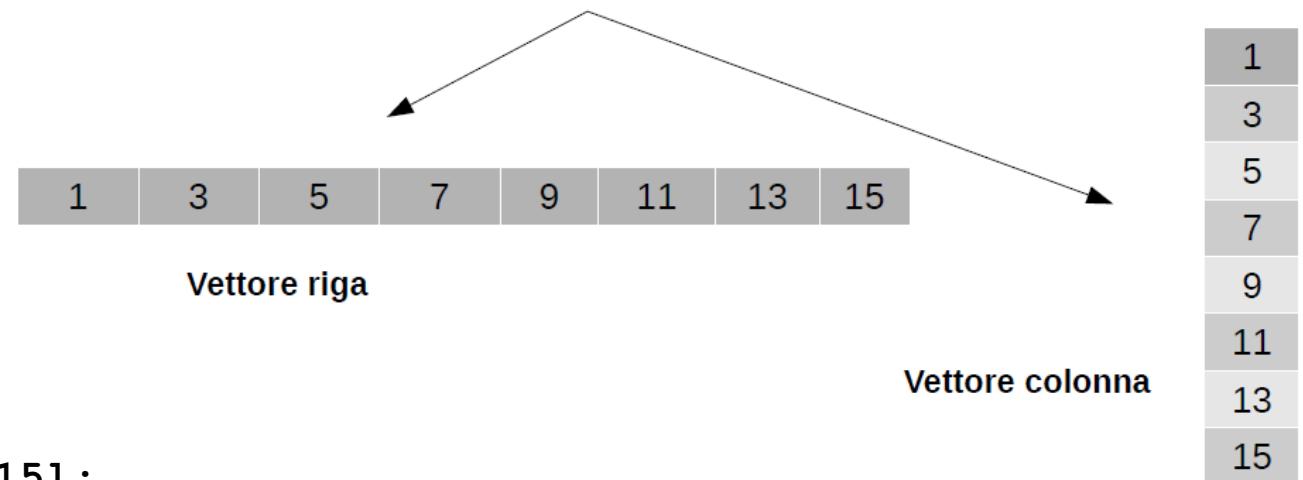
- È possibile trasformare un vettore riga in un vettore colonna usando l'apice ' (trasposizione):

```
>> v_colonna = v_riga';
```

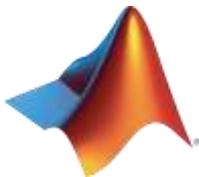
- Tramite l'uso degli indici è possibile accedere ad un dato valore al suo interno: >> v\_riga(3)

In Matlab gli indici iniziano da 1 e non da 0.

Rappresentazione in termini di array ad un solo indice in due modi



ans =



# Operazioni tra vettori

```
>> v1 = [1 2 3];  
>> v2 = [4 6 8];
```

## Somma

```
>> v1 + v2
```

```
ans =
```

```
5 8 11
```

## Differenza

```
>> v1 - v2
```

```
ans =
```

```
-3 -4 -5
```

Per determinare **massimo**: `>> max(v2)`

```
ans =  
8
```

Per determinare **minimo**: `>> min(v3)`

```
ans =  
4
```

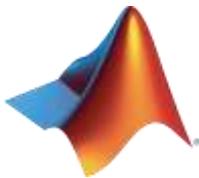
Operatori puntuali	Significato
<code>.*</code>	Moltiplicazione termine a termine di vettori
<code>./</code>	Divisione termine a termine di vettori
<code>.^</code>	Elevamento a potenza termine a termine di vettori
Prodotto	Rapporto
<code>&gt;&gt; v1.*v2</code>	<code>&gt;&gt; v1./v2</code>
<code>ans =</code>	<code>ans =</code>
<code>4 12 24</code>	<code>0.2500 0.3333 0.3750</code>

Altrimenti Matlab esegue il prodotto (o rapporto) riga per colonna, restituendo un solo numero.

È necessario che gli operandi abbiano le stesse dimensioni.

Per determinare la **lunghezza** di un vettore: `>> length(v2)`

```
ans =  
3
```



# Definire un vettore

- Per ottenere un vettore di punti equispaziati secondo un intervallo dx, definito tra un estremo inferiore (xStart) e uno superiore (xEnd):  
`>> x = 1:2:10`

**x = xStart:dx:xEnd**

Valore iniziale  
del vettore      incremento  
relativo (se  
omesso = 1)      Valore finale  
del vettore

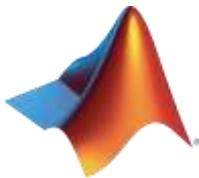
**x =**  
1      3      5      7      9

- In alternativa invece di specificare la spaziatura dei singoli elementi, è possibile definire il numero di elementi contenuti dal vettore tramite il comando linspace in cui i primi due input corrispondono agli estremi del vettore desiderato, mentre il terzo input corrisponde al numero totale di elementi (num) che si desiderano nel vettore:

**x = linspace(xStart, xEnd, num)**

Valore iniziale  
del vettore      Valore finale  
del vettore      Numero di punti  
(opzionale, se  
omesso = 100)

`>> x = linspace(1,10,5)`  
**x =**  
1.0000      3.2500      5.5000      7.7500      10.0000



# Matrici

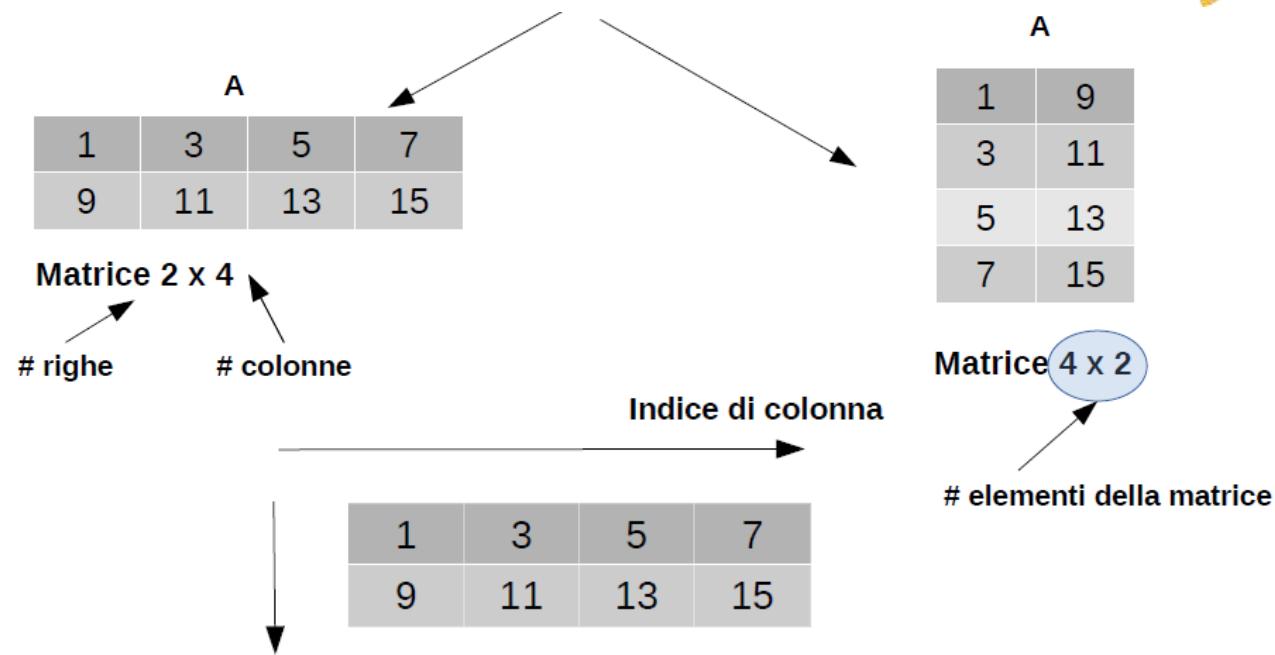
- Un **matrice** può essere definita tra parentesi quadre: nelle varie righe gli elementi sono separati mediante degli spazi o delle semplici virgole, mentre le varie righe vengono distinte con dei punti e virgola:

```
>> A = [1 3 5 7; 9 11 13 15];
```

Inizio nuova riga

- Per trasporre una matrice (scambiare le righe con le colonne) si usa sempre l'apice: `>> B = A'`
- Per accedere ad un elemento della matrice bisogna usare gli appositi indici (dove i corrisponde alla riga e j alla colonna) `>> A(i, j)`
- Per considerare tutti gli elementi di una riga (o colonna) basta specificare la riga (o colonna) da considerare e usare `(:)` `>> A(1, :)` `>> A(:, 3)`

Una sequenza di numeri può essere memorizzata anche come una matrice A (array a due indici)



**A(2,4) → 15** Restituisce il valore contenuto nella seconda riga e nella quarta colonna

**ans =**

1

3

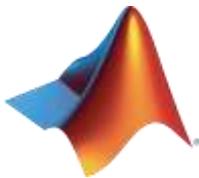
5

7

**ans =**

5

13



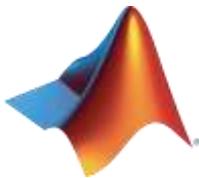
# Matrici

- Per determinare le dimensioni di una matrice si può usare il comando `size`, che restituisce come output due valori, corrispondenti rispettivamente al numero di righe e colonne della matrice:

Funzione	Azione
<code>zeros</code>	Matrice contenente solo elementi uguali a zero
<code>ones</code>	Matrice contenente solo elementi uguali a uno
<code>rand</code>	Matrice contenente numero pseudo-casuali in [0, 1]
<code>eye</code>	Matrice identità
<code>diag</code>	Matrice diagonale

```
>> M = [1 3 5 7; 9 11 13 15];  
>> size(M)  
  
ans =  
    2 4
```

*Funzioni Matlab per la costruzione di particolari matrici*



# Matrici

- Per determinare le dimensioni di una matrice si può usare il comando `size`, che restituisce come output due valori, corrispondenti rispettivamente al numero di righe e colonne della matrice:

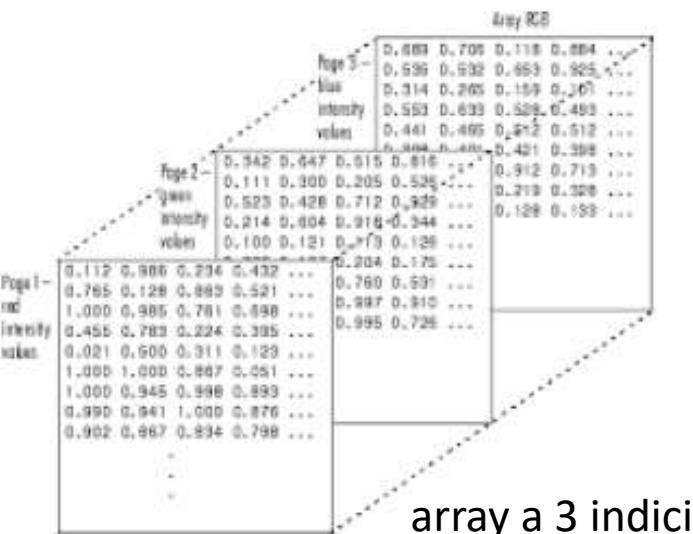
Funzione	Azione
<code>zeros</code>	Matrice contenente solo elementi uguali a zero
<code>ones</code>	Matrice contenente solo elementi uguali a uno
<code>rand</code>	Matrice contenente numero pseudo-casuali in [0, 1]
<code>eye</code>	Matrice identità
<code>diag</code>	Matrice diagonale

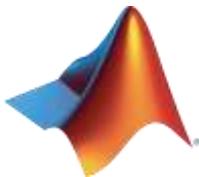
- È possibile utilizzare array a 3 indici (MATRICI TRIDIMENSIONALI):

```
>> A = [1 2 3; 4 5 6; 7 8 9];  
  
>> B(2,:,:)=A;  
  
>> C(:,2,:)=A;  
  
>> D(:,:,2)=A;
```

```
>> M = [1 3 5 7; 9 11 13 15];  
  
>> size(M)  
  
ans =  
2 4
```

*Funzioni Matlab per la costruzione di particolari matrici*





# Matrici tridimensionali

```
>> A = [1 2 3; 4 5 6; 7 8 9];
```

```
>> B(:,:,:) = A
```

```
B(:,:,1) =
```

0	0	0
1	4	7

```
C(:,:,1) =
```

0	1
0	4
0	7

```
>> C(:,:,2) = A
```

```
D(:,:,1) =
```

0	0	0
0	0	0
0	0	0

```
B(:,:,2) =
```

```
C(:,:,2) =
```

```
D(:,:,2) =
```

0	0	0
2	5	8

0	2
0	5
0	8

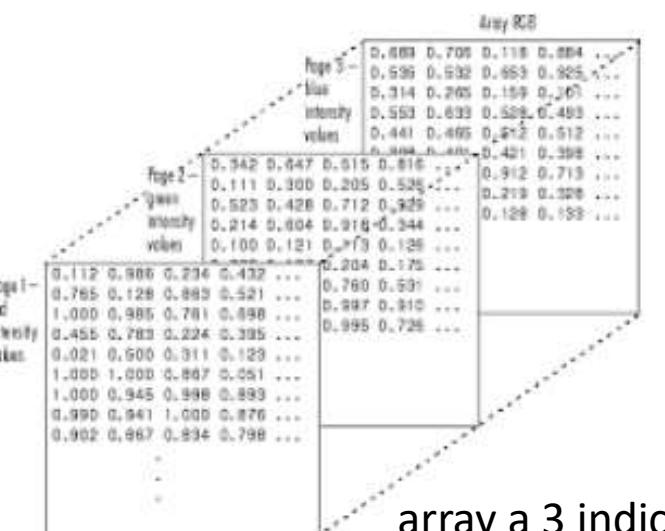
1	2	3
4	5	6
7	8	9

```
B(:,:,3) =
```

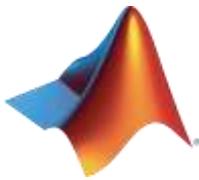
```
C(:,:,3) =
```

0	0	0
3	6	9

0	3
0	6
0	9



array a 3 indici



# Operazioni tra Matrici

- Due matrici possono essere sommate o sottratte se e solo se hanno le stesse dimensioni.
- Per sommare tutti gli elementi che compaiono in un vettore o in una matrice si può usare il comando sum. Nel caso di un vettore l'output è la somma di tutti i suoi elementi, mentre nel caso di una matrice in uscita viene ottenuto un vettore riga (di lunghezza pari al numero di colonne della matrice) i cui elementi sono la somma degli elementi presenti nella stessa colonna.

```
>> v = [1 2 3];  
  
>> somma = sum(v)  
  
somma =  
6
```

```
>> M = [1 2 3; 4 5 6; 7 8 9];  
  
>> somma = sum(M)  
  
somma =  
12      15      18
```

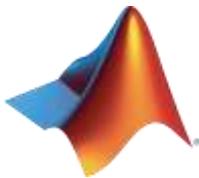
- Il prodotto tra matrici è diverso dal prodotto elemento per elemento.  
**N.B.** Il prodotto tra matrici è possibile solo se il numero delle colonne della prima matrice è uguale al numero di righe della seconda matrice.

```
>> C = A * B;
```

Prodotto tra matrici in senso classico (righe per colonne)

```
>> C = A .* B;
```

Prodotto elemento per elemento.



# Cicli if, for, while

N.B. Ciascun ciclo viene concluso mediante un end !

- Il **ciclo if** permette di eseguire delle istruzioni diverse in funzione della casistica esaminata, può essere necessario eseguire un certo blocco di istruzioni solo nel caso in cui una particolare istruzione risulti essere verificata ed eseguire un diverso blocco di istruzioni in caso contrario.

```
if condizione1
    blocco istruzioni 1
else if condizione2
    blocco istruzioni 2
...
else
    blocco istruzioni alternative
end
```

**Operatori di confronto:**

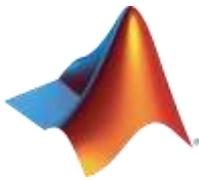
Operatore	Significato
<	minore
<=	Minore o uguale
==	uguale
>=	Maggiore o uguale
>	maggior
~=	Diverso da

Esempio istruzioni con if; vengono creati 2 array e vengono comparati usando `isequal` (l'operatore `==` risulterebbe in errore visto che gli array hanno dimensioni diverse). Se le dimensioni sono le stesse, i due array vengono concatenati altrimenti viene mostrato a video un warning e restituito un array vuoto.

**Operatori logici:**  
Gli operatori relazioni possono essere combinati mediante operatori logici.

Operatore	Significato
&	and
	or
~	not
xor	or esclusivo

a	b	a&b	a   b	~a	xor(a,b)
0	0	0	0	1	0
1	0	0	1	0	1
0	1	0	1	1	1
1	1	1	1	0	0



# Cicli if, for, while

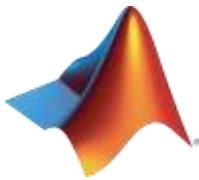
**N.B.** Ciascun ciclo viene concluso mediante un end !

- Il **ciclo if** permette di eseguire delle istruzioni diverse in funzione della casistica esaminata, può essere necessario eseguire un certo blocco di istruzioni solo nel caso in cui una particolare istruzione risulti essere verificata ed eseguire un diverso blocco di istruzioni in caso contrario.

```
A = ones(2,3);
B = rand(3,4);

if isequal(size(A), size(B))
    C = [A; B];
else
    disp('A e B hanno dimensioni diverse')
    C = [];
end
```

Esempio istruzioni con if; vengono creati 2 array e vengono comparati usando isequal (l'operatore == risulterebbe in errore visto che gli array hanno dimensioni diverse). Se le dimensioni sono le stesse, i due array vengono concatenati altrimenti viene mostrato a video un warning e restituito un array vuoto.



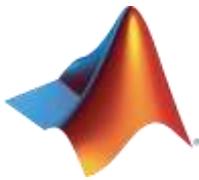
# Cicli `if`, `for`, `while`

**N.B.** Ciascun ciclo viene concluso mediante un `end` !

- Il **ciclo `for`** permette di effettuare una serie di operazioni, in ciascun ciclo all'inizio si specificano gli estremi tra cui varia l'indice. Ad esempio con il ciclo for è possibile effettuare operazioni per ogni elemento di un vettore, facendo variare un indice dopo ogni operazione.

```
for i=1:nEnd  
    blocco istruzioni  
end
```

Loop attraverso la matrice per assegnare un nuovo valore a ciascun elemento (in particolare il valore 2 viene assegnato ad ogni elemento sulla diagonale, -1 sulle due diagonali adiacenti e tutti gli altri elementi sono impostati a zero).



# Cicli `if`, `for`, `while`

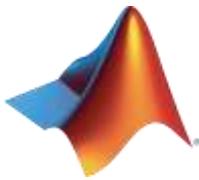
**N.B.** Ciascun ciclo viene concluso mediante un `end` !

- Il **ciclo `for`** permette di effettuare una serie di operazioni, in ciascun ciclo all'inizio si specificano gli estremi tra cui varia l'indice. Ad esempio con il ciclo for è possibile effettuare operazioni per ogni elemento di un vettore, facendo variare un indice dopo ogni operazione.

```
nrows = 4; ncols = 6;  
A = ones(nrows, ncols);  
  
for c=1:ncols  
    for r=1:nrows  
        if r == c  
            A(r,c)=2;  
        else if abs(r-c) == 1  
            A(r,c) = -1;  
        else  
            A(r,c) = 0;  
        end  
    end  
end
```

Loop attraverso la matrice per assegnare un nuovo valore a ciascun elemento (in particolare il valore 2 viene assegnato ad ogni elemento sulla diagonale, -1 sulle due diagonali adiacenti e tutti gli altri elementi sono impostati a zero).

**N.B.** Evitare l'suo eccessivo di cicli for in Matlab



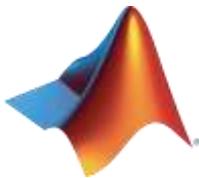
# Cicli `if`, `for`, `while`

**N.B.** Ciascun ciclo viene concluso mediante un `end`!

- Il **ciclo `while`** permette di continuare a fare una serie di operazioni (eseguire una serie di comandi) fino a quando la sua condizione risulta vera. (La condizione da soddisfare contiene generalmente un operatore relazionale).

```
while condizione  
    blocco istruzioni  
end
```

Data la serie armonica (somma degli inversi dei numeri naturali), calcolare quanti termini sono necessari prima di raggiungere almeno il valore 5 e mostrare il numero a video.



# Cicli `if`, `for`, `while`

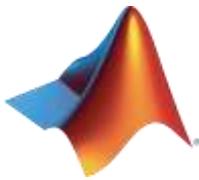
**N.B.** Ciascun ciclo viene concluso mediante un `end` !

- Il **ciclo `while`** permette di continuare a fare una serie di operazioni (eseguire una serie di comandi) fino a quando la sua condizione risulta vera. (La condizione da soddisfare contiene generalmente un operatore relazionale).

```
n = 0;  
somma = 0.;  
  
while somma < 5.  
    n = n + 1;  
    somma = somma + 1./n;  
end  
  
disp(['n=' , num2str(n)]);
```

Data la serie armonica (somma degli inversi dei numeri naturali), calcolare quanti termini sono necessari prima di raggiungere almeno il valore 5 e mostrare il numero a video.  
Con l'istruzione `num2str` è possibile convertire numeri in caratteri per la rappresentazione.

**N.B.** Evitare l'suo eccessivo di cicli `while` in Matlab



# Importare dati

Supported File Formats for Import and Export (Matlab documentation)

Inoltre si può usare anche **Import tool** per importare i dati da un file (spreadsheet files, delimited text files, fixed-width text files). Mostra una preview e permette di selezionare i dati da importare in modo interattivo. Per usarlo scrivere nel command prompt:

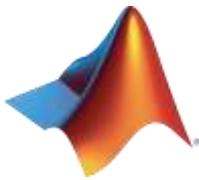
```
>> uiimport(nome_file)
```

Alcuni dei comandi più comuni per importare dati:

- **load(nome\_file)** carica le variabili dal file nel workspace;
- **num = xlsread(nome\_file)** legge Excel spreadsheet file e restituisce i dati numerici in una matrice;
- **A = importdata(nome\_file)** carica i dati dal file in un array A;
- **M = readmatrix(nome\_file)** legge una matrice da file (column-oriented);
- **T = readtable(nome\_file)** carica una tabella dal file (column-oriented);

# Parte 3

## Elaborazione dei dati in Matlab



# Caratterizzazione Metrologica di un sensore ed Elaborazione Dati in Matlab

- Importazione dei dati da file Excel (o da file di testo)
- Creazione delle variabili G (Vettore), Tensione (Matrici)
- Calcolo della Media e della Deviazione Standard (Assoluta, Relativa e Totale)
- Rappresentazione Grafica delle Tensioni (Plot multipli, Subplot, Settings, Errorbar, Salvataggio ed esportazione)
- Fitting dei dati (Basic Fitting, Fit, Polyfit, CFTool)
- Media Globale e Calcolo dell'Isteresi

# Caratterizzazione Metrologica di un sensore

- Importazione dei dati da file Excel (o da file di testo)
- Creazione delle variabili G (Vettore), Tensione (Matrici)
- Calcolo della Media e della Deviazione Standard (Assoluta, Relativa e Totale)
- Rappresentazione Grafica delle Tensioni (Plot multipli, Subplot, Settings, Errorbar, Salvataggio ed esportazione)
- Fitting dei dati (Basic Fitting, Fit, Polyfit, CFTool)
- Media Globale e Calcolo dell'Isteresi

**Doppio Codice per  
Salita e Discesa?**

# Funzioni definite dall'utente (M-File)

Le funzioni sono programmi (o routine) che accettano argomenti di input e restituiscono argomenti di output. Ogni funzione ha la propria area di lavoro, separata dall'area di lavoro di base MATLAB.

Una funzione inizia con una riga di definizione che contiene l'elenco delle variabili di input e output ed il nome della funzione:

```
function [variabili_output] = nome_funzione (variabili_input)
```

ATTENZIONE: il nome del file deve essere uguale al nome della funzione!

E' utile scrivere una riga di commento, contenente le parole chiave della funzione, immediatamente dopo la riga di definizione della variabile per facilitare le ricerche con il comando **lookfor**.

# Struttura di una funzione

1	Riga di definizione	var. output, nome funzione, var. input
2	Prima riga di intestazione	Riga descrittiva sintetica visualizzabile con l'help e utile per la ricerca con lookfor
3	Help text	Descrizione dettagliata della funzione
4	Corpo della funzione	Codice vero e proprio

# Funzioni definite dall'utente (Esempio)

Scrivere una funzione per calcolare la tensione di uscita di un sensore noto il valore G della grandezza applicata in ingresso.

```
% MAIN
Gin = 25;
Vout = Sensore (Gin);

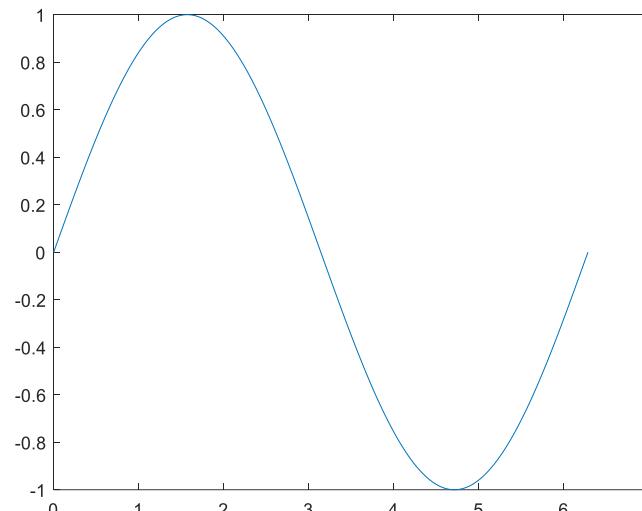
% FUNZIONE
function [y] = Sensore (x)
m = 0.0409;
q = -0.8066;
y = m*x+q;
```

# Basic Plotting

La procedura di base per la rappresentazione grafica di MATLAB, ad esempio in 2D, consiste nel prendere un vettore di ascisse,  $x = (x_1, \dots, x_N)$  e un vettore di ordinate,  $y = (y_1, \dots, y_N)$ , individuare i punti  $(x_i, y_i)$ , con  $i = 1, 2, \dots, n$  e quindi unirli con linee rette. È necessario preparare  $x$  e  $y$  in una forma di matrice identica; vale a dire,  $x$  e  $y$  sono entrambi array di righe o array di colonne della stessa lunghezza.

Esempio:

```
x = 0 : pi/100 : 2*pi;  
y = sin (x);  
figure  
plot (x,y)
```

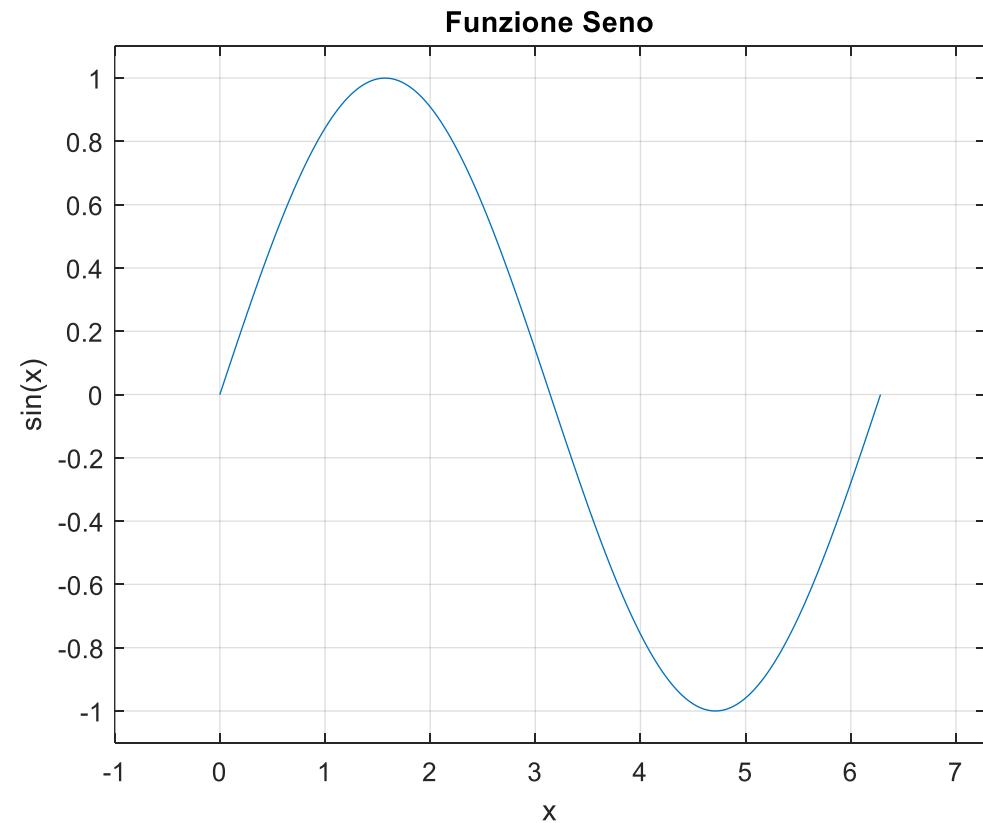


# Basic Plotting

Aggiunta di elementi grafici, es. titolo, etichette sugli assi, annotazioni ...

Esempio:

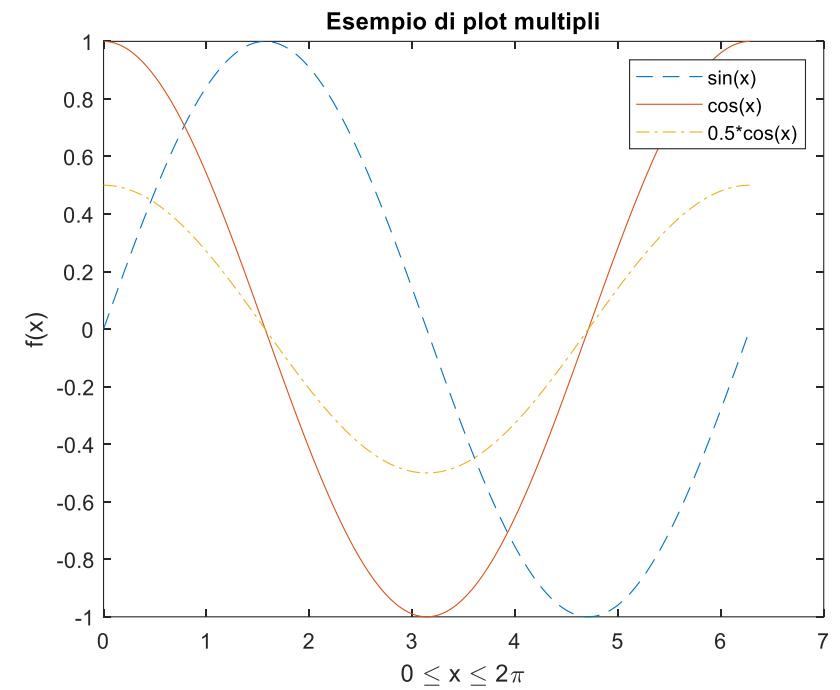
```
xlabel ('x')
ylabel ('sin(x)')
title('Funzione Seno')
grid on
box on
axis ([-1 2*pi+1 -1.1 1.1])
```



# Plot sovrapposti

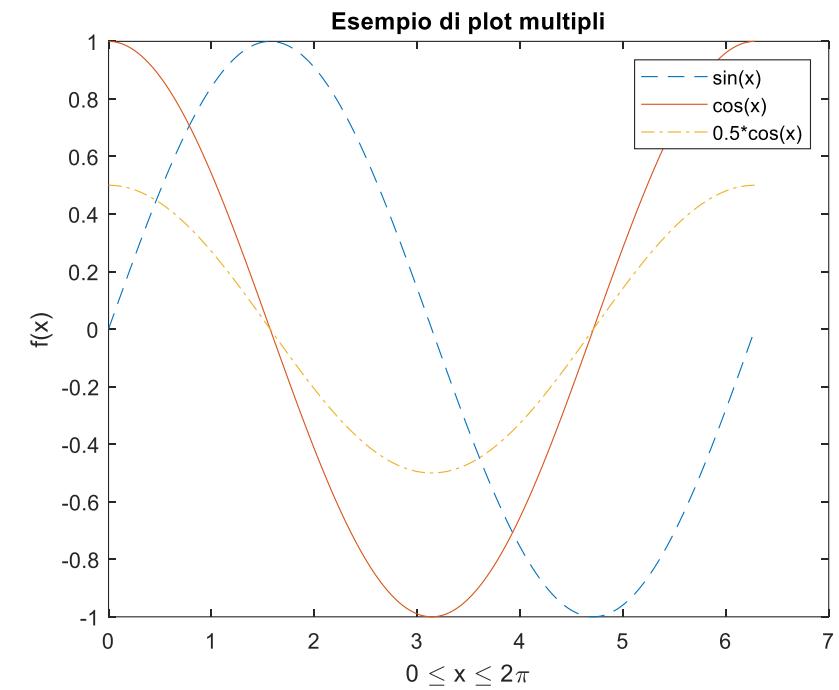
Esempio:

```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = 0.5*cos(x);
figure
plot(x,y1,'--',x,y2,'-',x,y3,'-.')
xlabel('0 \leq x \leq 2\pi')
ylabel('f(x)')
legend('sin(x)', 'cos(x)', '0.5*cos(x)')
title('Esempio di plot multipli')
```



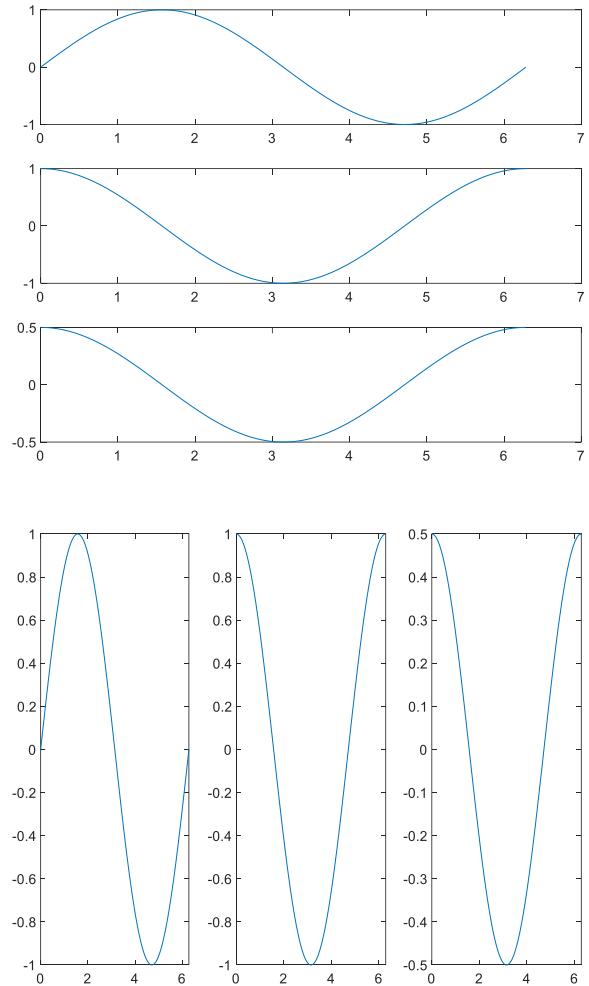
# Plot sovrapposti

```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = 0.5*cos(x);
figure
plot(x,y1,'--')
hold on
plot(x,y2,'-')
plot(x,y3,'-.')
xlabel('0 \leq x \leq 2\pi')
ylabel('f(x)')
legend('sin(x)', 'cos(x)', '0.5*cos(x)')
title('Esempio di plot multipli')
```



# Plot multipli

```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = cos(x);
y3 = 0.5*cos(x);
figure
subplot (3,1,1), plot (x,y1), title ('sin(x)')
subplot (3,1,2), plot (x,y2), title ('cos(x)')
subplot (3,1,3), plot (x,y3), title ('0.5*cos(x)')
figure
subplot (1,3,1), plot (x,y1), title ('sin(x)')
subplot (1,3,2), plot (x,y2), title ('cos(x)')
subplot (1,3,3), plot (x,y3), title ('0.5*cos(x)')
```



# Fitting dei dati

Il problema dell'approssimazione di dati consiste nel voler determinare una funzione  $g(x)$  che possa caratterizzare l'andamento di un dato insieme di punti  $(x_0, y_0), \dots, (x_N, y_N)$

## Interpolazione polinomiale

$g(x_i) = y_i, i = 0, \dots, N$  dove  $g$  deve passare per tutti i punti dati

**Non sempre questo è il modo migliore di procedere!**

Con dati sperimentali affetti da vari errori l'interpolazione perde senso!

# Fitting dei dati – Sensore

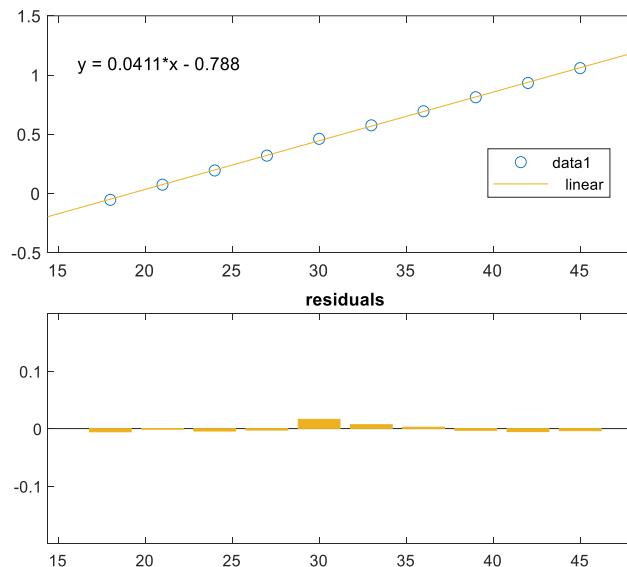
**Problema:** approssimare la tensione in uscita dal sensore nella prima rampa di salita

1. Realizzare un plot con i dati (x,y)
  
2. Scegliere un metodo per il fitting:
  - A. Basic Fitting
  - B. Fit / PolyFit / FitType / ...
  - C. CFTool

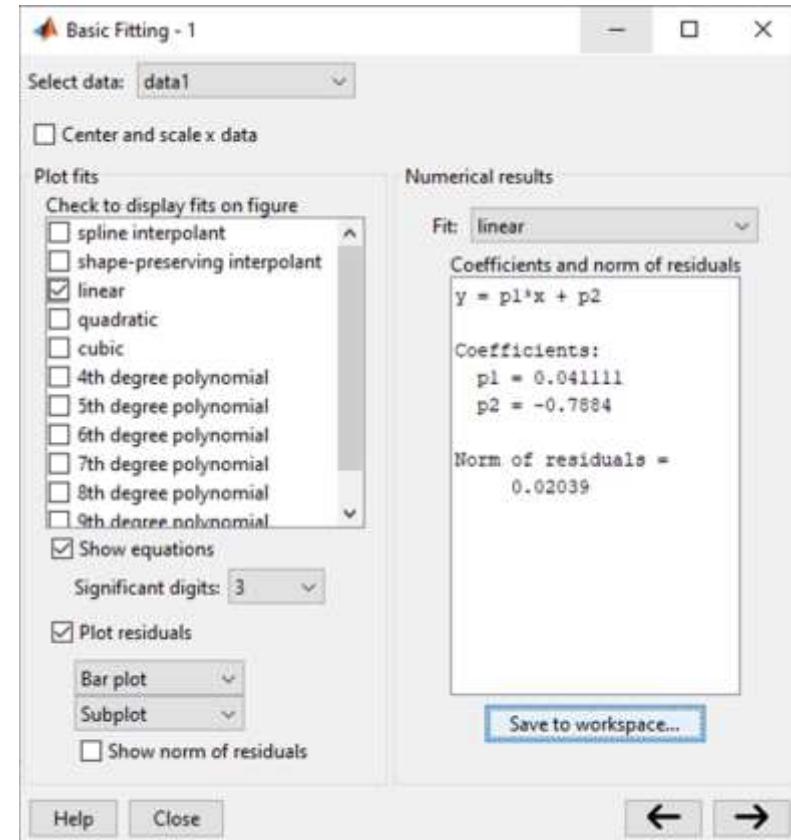
# Fitting dei dati – Basic Fitting

Dopo aver creato la figura con il plot dei dati, è possibile utilizzare lo strumento Basic Fitting per eseguire l'approssimazione dei dati e visualizzarla direttamente sul grafico.

- Immediato utilizzo
- Spline, shape-preserving interpolant
- Fitting polinomiali (fino al 10°)
- Equazione sul grafico
- Plot dei residui
- Salvataggio nel workspace dei coefficienti, residui e dei valori della funzione in un intervallo specificato dall'utente



esempio



# Fitting dei dati – Funzione Fit

**fit** Fit a curve or surface to data.

**FO = fit(X, Y, FT)** creates a fit object, FO, that encapsulates the result of fitting the model specified by the fittype FT to the data X, Y.

-- X must be a matrix with either one (curve fitting) or two (surface fitting) columns. For surface fitting, if your data is in separate vectors, then you can use the following syntax:

**fo = fit([x, y], z, ft)**

-- Y must be a column vector with the same number of rows as X.

-- FT is a string or a FITTYPE specifying the model to fit.

# Fitting dei dati – Funzione Fit

-- FT is a string or a FITTYPE specifying the model to fit.

If FT is a string, then it may be:

FITTYPE	DESCRIPTION
'poly1'	Linear polynomial curve
'poly11'	Linear polynomial surface
'poly2'	Quadratic polynomial curve
'linearinterp'	Piecewise linear interpolation
'cubicinterp'	Piecewise cubic interpolation
'smoothingspline'	Smoothing spline (curve)
'lowess'	Local linear regression (surface)

or any of the names of library models described in "List of Library Models for Curve and Surface Fitting" in the documentation.

# Fitting dei dati – Funzione Fit

**[FO, G] = fit(X, Y, ...)** returns appropriate goodness-of-fit measures, for the given inputs, in the structure G. G includes the fields:

- SSE sum of squares due to error
- R2 coefficient of determination or  $R^2$
- adjustedR2 degree of freedom adjusted  $R^2$
- stdError fit standard error or root mean square error

**[FO, G, O] = fit(X, Y, ...)** returns a structure, O, with output values appropriate for the given inputs. For example, for nonlinear fitting, O contains the number of iterations, number of model evaluations, an exitflag denoting convergence, the residuals, and the Jacobian.

# Fitting dei dati – Funzione Fit

**polyfit** Fit polynomial to data.

**P = polyfit(X,Y,N)** finds the coefficients of a polynomial P(X) of degree N that fits the data Y best in a least-squares sense. P is a row vector of length N+1 containing the polynomial coefficients in descending powers,  $P(1)*X^N + P(2)*X^{N-1} + \dots + P(N)*X + P(N+1)$ .

# Fitting dei dati - Curve Fitting Toolbox

Adattamento di curve e superfici ai dati usando regressione, interpolazione e linearizzazione

Curve Fitting Toolbox™ fornisce un'applicazione e una serie di funzioni per l'adattamento di curve e superfici ai dati. Il toolbox consente di eseguire analisi esplorative, pre-elaborare e post-elaborare dati, confrontare modelli candidati e rimuovere valori anomali. È possibile condurre analisi di regressione utilizzando la libreria di modelli lineari e non lineari forniti o specificare le proprie equazioni personalizzate. La libreria fornisce condizioni iniziali e parametri del solutore ottimizzati per migliorare la qualità degli adattamenti. Il toolbox supporta anche tecniche di modellazione non parametriche, come spline, interpolazione e linearizzazione.

Dopo aver creato un adattamento, è possibile applicare una varietà di metodi di post-elaborazione per il plottaggio, l'interpolazione e l'estrapolazione, la stima degli intervalli di confidenza e il calcolo di integrali e derivati.

<https://it.mathworks.com/products/curvefitting.html#resources>

# Caratterizzazione di un sensore – Codice MATLAB

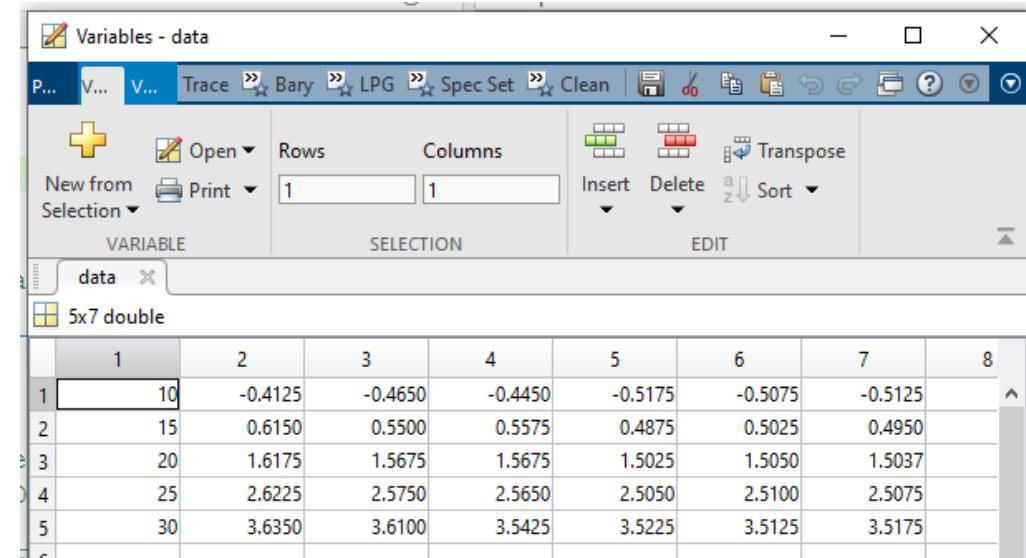
## Importazione dei dati da Excel

A	B	C	D	E	F	G	
1	G_in (X)	V_out (mV) (1a salita)	V_out (mV) (1a discesa)	V_out (mV) (2a salita)	V_out (mV) (2a discesa)	V_out (mV) (3a salita)	V_out (mV) (3a discesa)
2	10	-0.4125	-0.465	-0.445	-0.5175	-0.5075	-0.5125
3	15	0.615	0.55	0.5575	0.4875	0.5025	0.495
4	20	1.6175	1.5675	1.5675	1.5025	1.505	1.50375
5	25	2.6225	2.575	2.565	2.505	2.51	2.5075
6	30	3.635	3.61	3.5425	3.5225	3.5125	3.5175
7							

```
%% Import Data from File

clc; % Clean command window
clear; % Clear all variables in the workspace
close all; % Close all the figures

data = xlsread('Cartell1.xlsx',1,'A2:G6'); % Import data
from Excel
```

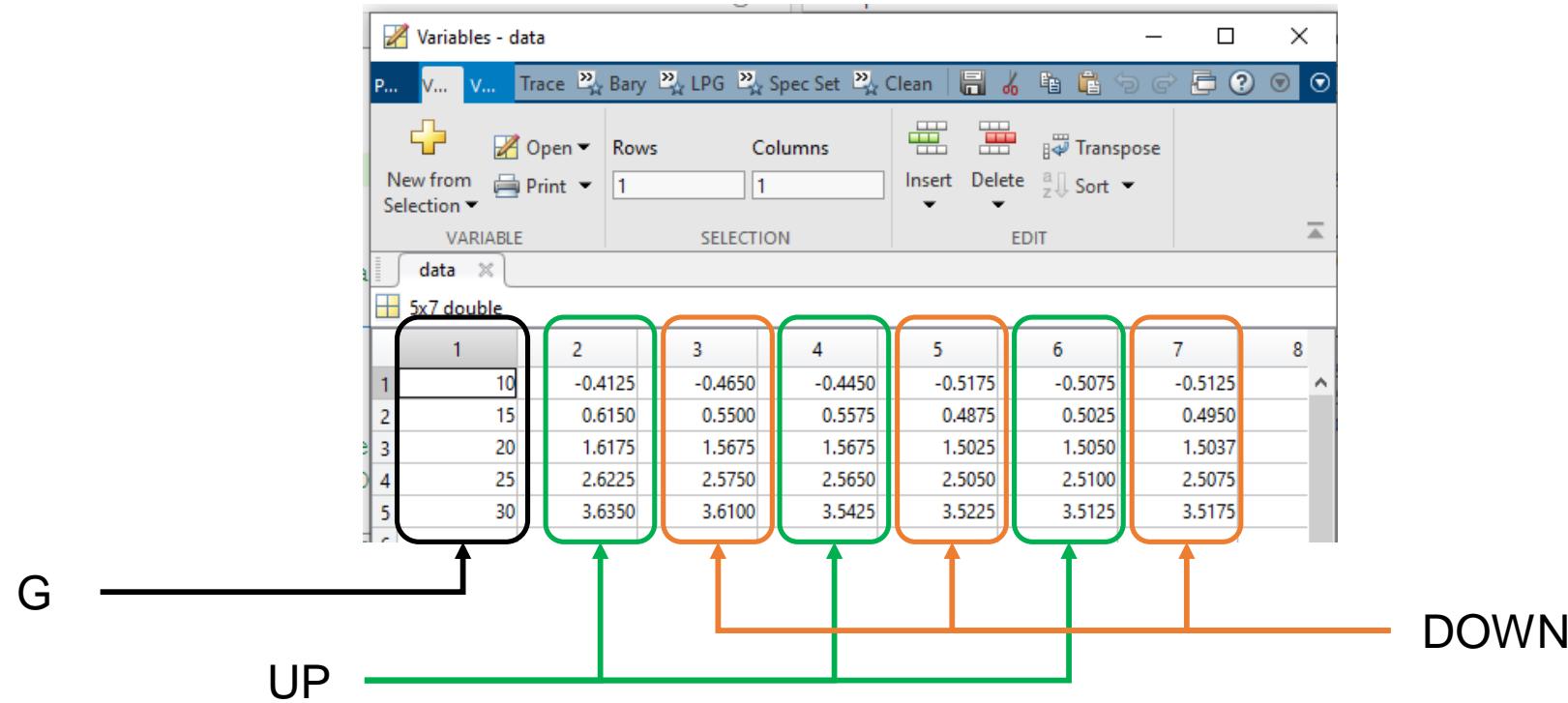


# Caratterizzazione di un sensore – Codice MATLAB

## Creazione di vettori e matrici

```
%% Creation of the Arrays G, UP & DOWN

G = data (:,1); % applied quantity G
UP = [data (:,2),data (:,4),data (:,6)]; % Ramps UP (Increasing)
DOWN = [data (:,3),data (:,5),data (:,7)]; % Ramps DOWN (Decreasing)
```



# Caratterizzazione di un sensore – Codice MATLAB

## Chiamata alla funzione di elaborazione

```
%% Call to Function for Elaboration of Up & Down Data Sets  
  
[M_UP] = Elab_data (G,UP,'Rising G'); % Call for ramps UP  
[M_DOWN] = Elab_data (G,DOWN,'Falling G'); % Call for ramps DOWN
```

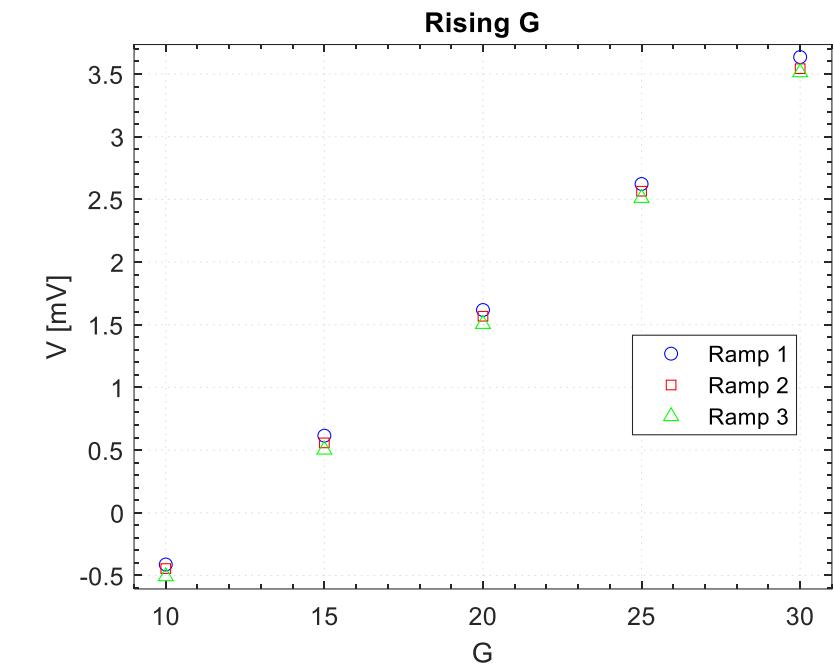
Poiché buona parte del codice si ripete per l'analisi dei dati in salita e in discesa...

# Caratterizzazione di un sensore – Codice MATLAB

## Grafico delle tre rampe (es. di salita)

```
%% Elaboration Data from the sensor (UP or DOWN)

function [M] = Elab_data (G,V,Title)
% Plot three ramps in the same graph
figure, plot (G,V(:,1), 'bo', G,V(:,2), 'rs', G,V(:,3), 'g^')
xlabel ('G'), ylabel ('V [mV]')
title (Title)
set (gca, 'XMinorTick', 'on', 'XMinorGrid', 'off', 'YMinorTick', 'on',...
'YMinorGrid', 'off', 'GridLineStyle', ':');
grid on; box on; set (gca, 'FontSize', 12);
set (gca, 'XLim', [G(1)-1,G(end)+1], 'YLim', [min(min(V))-0.1,max(max(V))+0.1])
l = legend ('Ramp 1','Ramp 2','Ramp 3'); set (l, 'location', 'Best')
% saveas (gcf,strcat>Title, '.fig')) % Save as MATLAB figure
% saveas (gcf,strcat>Title, '.png')) % Save as PNG image
```



# Caratterizzazione di un sensore – Codice MATLAB

## Calcolo della media, deviazione standard e fitting

```
%% Mean & Standard Deviation

M = mean (V,2); % Mean along the 2nd dimension of V (columns)
SD = std (V,0,2); % Standard Deviation with N-1 along the 2nd dimension
RSD = abs(100*SD./M); % Relative Standard Deviation
RSD_tot = mean (RSD); % Total Relative Standard Deviation
disp(['Total RSD (',Title,',') = ',num2str(RSD_tot), ' %']) % Display in command window

%% Fitting & Sensitivity with R-square

[F,E] = fit (G,M,'Poly1');
% disp (F)
F1 = polyfit (G,M,1); % polynomial fitting, first order; F(x) = p1*x+p2; F1 = [p1,p2]
disp(['Sensitivity (',Title,',') = ',num2str(F1(1)), ' mV/[G]'])
disp(['R^2 (',Title,',') = ',num2str(E.adjrsquare)])
```

### Command Window:

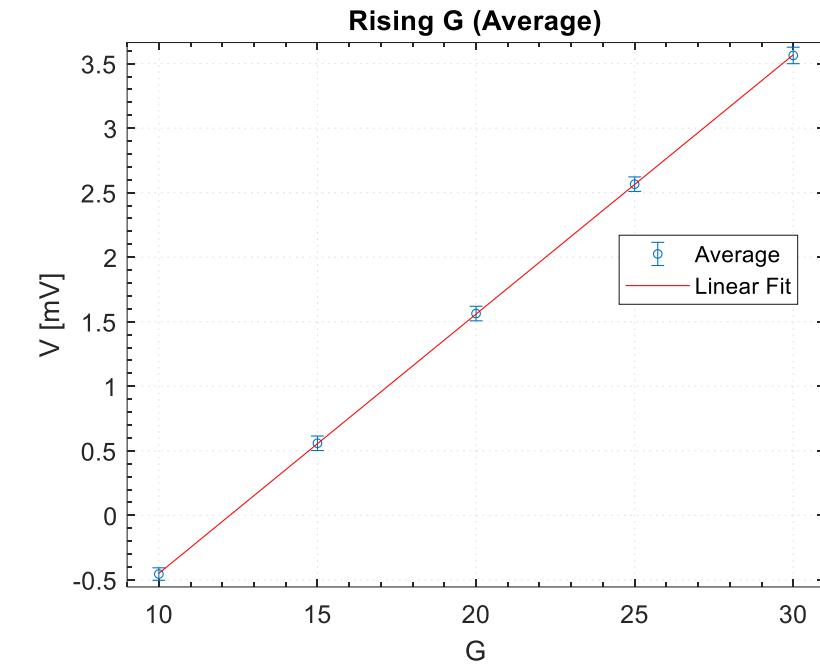
Total RSD (Rising G) = 5.6554 %  
Sensitivity (Rising G) = 0.20088 mV/[G]  
R^2 (Rising G) = 0.99999

# Caratterizzazione di un sensore – Codice MATLAB

## Grafico della media, barre di errore e fitting

```
%% Plots Average & Fitting

figure, errorbar (G,M,SD, 'o', 'MarkerSize', 4)
hold all, plot (F)
xlabel ('G'), ylabel ('V [mV]')
title ([Title, ' (Average)'])
set (gca,'XMinorTick','on','XMinorGrid','off','YMinorTick','on',...
'YMinorGrid','off','GridLineStyle',':');
grid on; box on; set (gca,'FontSize',12);
set (gca,'XLim',[G(1)-1,G(end)+1], 'YLim',[min(M)-0.1,max(M)+0.1])
l = legend ('Average', 'Linear Fit'); set (l,'location','Best')
% saveas (gcf,strcat>Title, ' Average.fig'))
% saveas (gcf,strcat>Title, ' Average.png'))
```

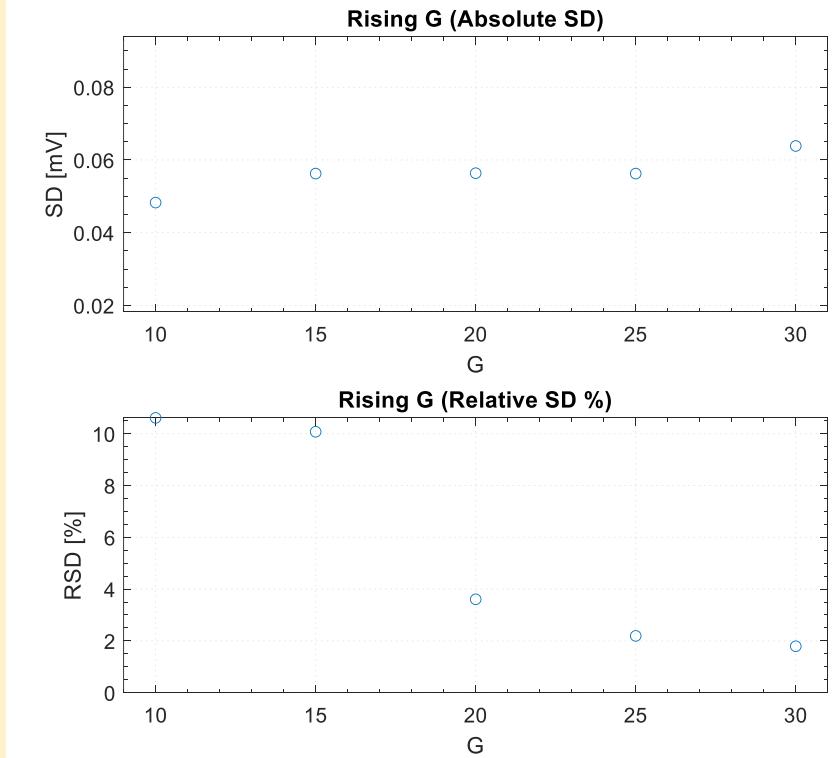


# Caratterizzazione di un sensore – Codice MATLAB

## Grafico della deviazione standard (assoluta e relativa)

```
%% Plots Standard Deviation

figure, subplot (2,1,1), plot (G,SD, 'o')
ylabel ('SD [mV]')
title ([Title, ' (Absolute SD)'])
set (gca,'XLim',[G(1)-1,G(end)+1], 'YLim', [min(SD)-0.03,max(SD)+0.03])
subplot (2,1,2), plot (G,RSD, 'o')
ylabel ('RSD [%]')
title ([Title, ' (Relative SD %)'])
set (gca,'XLim',[G(1)-1,G(end)+1])
for i = 1 : 2
    subplot (2,1,i)
    xlabel ('G'),
    set (gca,'XMinorTick','on','XMinorGrid','off','YMinorTick','on',...
        'YMinorGrid','off','GridLineStyle',':');
    grid on; box on; set (gca,'FontSize',12);
end
set (gcf,'OuterPosition',[200,100,700,700])
% saveas (gcf,strcat>Title, ' SD.fig'))
% saveas (gcf,strcat>Title, ' SD.png'))
```



# Caratterizzazione di un sensore – Codice MATLAB

## Calcolo dell’Isteresi e fitting totale

```
%% Hysteresis

M = mean ([M_UP,M_DOWN],2); % Average over the two mean vectors
H = 100*abs((M_UP-M_DOWN)./M); % Hysteresis
H_tot = mean (H); % Total Hysteresis
disp(['Total Hysteresis = ',num2str(H_tot),'%']); % Display Total Hysteresys

%% Global Fitting & Sensitivity with R-square

[F,E] = fit (G,M,'Poly1');
% disp (F)
F1 = polyfit (G,M,1); % polynomial fitting, first order; F(x) = p1*x+p2; F1 = [p1,p2]
disp(['Sensitivity (Global) = ',num2str(F1(1)), ' mV/[G]'])
disp(['R^2 (Global) = ',num2str(E.adjrsquare)])
```

### Command Window:

```
Total Hysteresis = 4.4601 %
Sensitivity (Global) = 0.20159 mV/[G]
R^2 (Global) = 1
```

# Caratterizzazione di un sensore – Codice MATLAB

## Grafico della caratteristica totale e dell'isteresi

```
%% Plots Global Average & Hysteresis

figure, subplot (2,1,1), plot (G,M, 'o')
hold all, plot (F)
ylabel ('V [mV]')
title ('Global Average')
set (gca,'XLim',[G(1)-1,G(end)+1], 'YLim', [min(M)-0.1,max(M)+0.1])
l = legend ('Average','Linear Fit'); set (l,'location','Best')
subplot (2,1,2), plot (G,H, 'o')
ylabel ('Hysteresis [%]')
title ('Hysteresis %')
set (gca,'XLim',[G(1)-1,G(end)+1], 'YLim', [min(H)-5,max(H)+5])
for i = 1 : 2
    subplot (2,1,i)
    xlabel ('G'),
    set (gca,'XMinorTick','on','XMinorGrid','off','YMinorTick','on',...
        'YMinorGrid','off','GridLineStyle',':');
    grid on; box on; set (gca,'FontSize',12);
end
set (gcf,'OuterPosition',[200,100,700,700])
```

