

Progetto finanziato dal **Ministero dell'Università e della Ricerca**

Enti promotori:
FabLab dell'Università di Trento
e **Level Up srl**

KIT MISURA

Programmare Arduino nel laboratorio di Fisica



KIT MISURA

Programmare Arduino nel laboratorio di Fisica

Questa scheda introduttiva accompagna il Kit Misura e fornisce gli elementi necessari per approcciare correttamente l'attività di laboratorio proposta, indicazioni sull'utilizzo degli strumenti di cui il kit è composto e spunti di riflessione.

Come nasce questo kit? Quale scopo si pone? Perché si pone l'accento sul concetto di misura? Quali sono le motivazioni che hanno spinto a realizzare un kit che renda necessario un dialogo serrato tra fisica e informatica? In questa introduzione risponderemo a tutte queste domande.

FISICA

In ogni scuola secondaria di secondo grado, lo studio della fisica parte dalla definizione o, meglio, dall'etimologia del nome: *Natura*. Un termine moderno e affascinante che, tuttavia, non sembra rispecchiare l'esperienza degli studenti, che si ritrovano a studiare per molti mesi il concetto di misura, l'incertezza e la sua propagazione, e le principali relazioni matematiche che intercorrono tra due o più grandezze fisiche: è difficile cogliere in tutto questo un nesso con la natura e il suo fascino. Anche l'approccio didattico spesso non è di aiuto: distante dalle "sensate esperienze"¹ di Galileo, il primo incontro con la fisica consiste solitamente nella lettura del libro di testo e nello studio prettamente teorico; le esperienze di laboratorio si riducono a poche occasioni nel corso dell'anno.

OBIETTIVI

Gli obiettivi del Kit Misura sono **(1)** riavvicinare lo studio della misura all'esperienza diretta dei fenomeni naturali e **(2)** permettere di operare in laboratorio con un approccio sperimentale, più vicino al metodo di lavoro di un ricercatore.

La misura

Per raggiungere il primo obiettivo, dobbiamo ricordare il motivo per cui si studiano la misura e l'incertezza in fisica. Questa disciplina, infatti, si occupa di tutti i fenomeni naturali che ci circondano, e in particolare della definizione delle procedure di misura di ciascun fenomeno rilevabile. Non solo: spesso, per poter raggiungere gli obiettivi di un esperimento di fisica, è necessario osservare fenomeni mai visti o analizzare i dettagli di un esperimento con una precisione mai raggiunta prima. L'attività del ricercatore si traduce quindi anche nell'ideazione, progettazione e costruzione di nuovi strumenti di misura – si pensi, per esempio, all'interferometro che ha rilevato le onde gravitazionali nel 2016. Inoltre, ogni altra disciplina scientifica si riferisce alle tecniche e alle metodologie sperimentali definite dalla fisica quando si tratta di effettuare una misura.

La misura, spesso trattata in modo nozionistico, è il perno su cui ruota tutto il metodo scientifico e separa fatti scientifici da temi pseudo-scientifici.

¹ Sono il nome che Galileo dava agli esperimenti, parte integrante del metodo scientifico, che avevano l'obiettivo di osservare i fenomeni scelti per poter formulare ipotesi sensate. Si osservano dei singoli fenomeni per trovare una legge che li accomuni tutti.

Tra i tanti aspetti fondamentali legati al concetto di misura citiamo i tre principali:

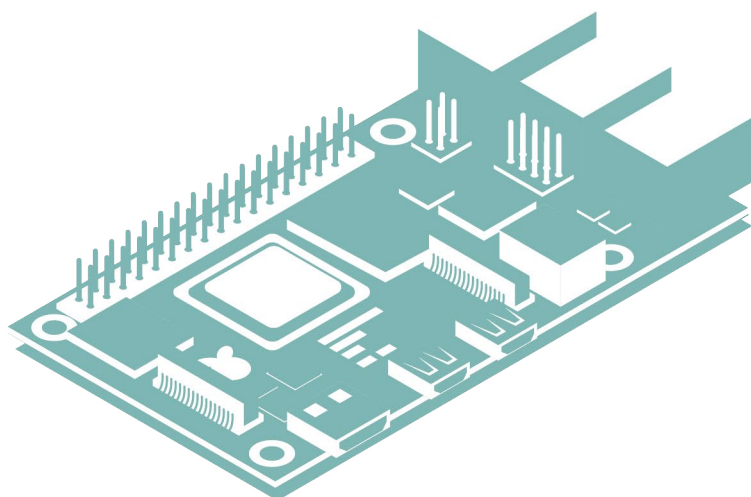
1. tutto ciò che non può essere misurato non appartiene al sapere scientifico;
2. una misura deve poter essere condotta da chiunque voglia formarsi sull'utilizzo di un determinato strumento; qualora esistano procedure di misura che solo una determinata persona può condurre, esse saranno probabilmente false, pseudo-scientifiche;
3. il valore espresso dallo strumento e la procedura adottata devono essere condivisi con tutta la comunità scientifica. Un dato sottoposto a segreto o non condivisibile è di per sé contraddittorio e indice di una procedura poco corretta.

Lavorare nel campo della ricerca

Per poter raggiungere il secondo obiettivo, presentiamo un approccio didattico pratico e *inquiry-based*, nel quale lo studente diventa il protagonista della lezione e il docente, con la sua esperienza, accompagna nell'individuazione degli elementi cardine e propone spunti di riflessione. Le procedure didattiche che verranno suggerite sono state pensate per accompagnare in un percorso il più possibile vicino al reale lavoro di un ricercatore in ambito scientifico. Spesso, infatti, ci si reca in laboratorio e si ricevono indicazioni precise su come lavorare; sulla scheda di laboratorio è presente una procedura chiara; il docente e il tecnico conoscono i dati attesi; il setup sperimentale è stato predisposto per il corretto svolgimento dell'esperimento: tutto questo non potrebbe essere più distante dal modo in cui opera un qualsiasi ricercatore scientifico.

Per questo, abbiamo pensato di fondare il nostro kit e tutte le attività su queste idee.

- Lo strumento di misura non è pronto all'uso, ma deve essere costruito e predisposto in un setup che ne permetta il corretto utilizzo.
- Nel momento in cui il setup sperimentale diventa funzionante, le condizioni di lavoro sono comunque caratterizzate da molte interferenze ambientali che complicano la misura. Non offriremo risposte o soluzioni alle problematiche che si incontreranno, ma spunti di riflessione su alcuni aspetti e possibili casi studio da valutare e applicare al proprio specifico contesto sperimentale. Si rifletterà sull'operato del gruppo e su ciò che è stato osservato durante l'esperienza;
- A ogni ricercatore di fisica sono richieste altre competenze trasversali – principalmente la conoscenza dell'inglese e dell'informatica. Per questo, proponiamo di costruire gli strumenti utilizzando il microprocessore Arduino Uno. Per poter effettuare le misure proposte, è possibile modificare o riscrivere un algoritmo e concretizzarlo in un semplice software, partendo da quelli proposti; questo aiuterà a sviluppare le proprie competenze di coding, sempre con l'aiuto delle indicazioni contenute in questo testo.



LO STRUMENTO DI MISURA

Come abbiamo visto, la procedura e lo strumento di misura sono i capisaldi del metodo scientifico e la fisica è la disciplina che se ne occupa. Non è scopo di questo scritto trattare in maniera esaustiva la teoria che riguarda questo tema; tuttavia, è necessario introdurre qui alcuni elementi fondamentali di uno strumento di misura. Invitiamo ad approfondire il tema in un'altra sede.

I parametri essenziali di un qualsiasi strumento di misura che vanno conosciuti al fine di valutare le sue potenzialità e i suoi limiti sono i seguenti:

- **Portata:** detto anche fondo scala, è il massimo valore della grandezza fisica in esame che è possibile misurare.
- **Sensibilità:** è la minima variazione della grandezza che è possibile misurare.
- **Taratura:** è la procedura che permette a uno strumento di essere coerente con l'unità di misura scelta. La procedura di taratura va effettuata paragonando la misura effettuata con lo standard di riferimento per la grandezza fisica in esame.
- **Prontezza:** è la velocità di risposta dello strumento, ossia il tempo che trascorre dal momento della misura a quello in cui lo strumento restituisce il valore corrispondente alla misura.
- **Accuratezza:** indica quanto la misura sia prossima al valore corretto. Si usa invece il termine precisione per indicare quanto più l'incertezza relativa è piccola.

Queste caratteristiche spesso sono note e dichiarate dal produttore, ma quando i ricercatori devono inventare un nuovo strumento di misura devono procedere in autonomia alla loro valutazione, per testare i limiti della propria invenzione. Questo kit permetterà di esplorare proprio la fase fondamentale della realizzazione di uno strumento: la valutazione delle sue caratteristiche e il suo utilizzo.

L'incertezza

Qualsiasi operazione di misura condotta con qualsiasi strumento, sensibile a piacere, fornisce due valori: un numero corrispondente alla misura della grandezza in esame e un secondo valore, corrispondente all'incertezza sulla misura stessa.

L'operazione di stima dell'incertezza è molto complessa e richiede attenzione poiché deve tenere in considerazione vari fattori:

- la sensibilità dello strumento;
- la dispersione dei dati qualora si abbia a che fare con un insieme di misure;
- eventuali errori sistematici dovuti a difetti dello strumento o a procedure di misura errate;
- eventuali errori casuali legati al fenomeno fisico d'interesse o alla procedura.

Lasciamo ad altri testi, riportati in bibliografia, l'approfondimento sull'individuazione dell'incertezza e sulla sua propagazione nelle misure derivate.

IL KIT

Il Kit Misura vuole immergere in un'attività pratica dove si mescolano fisica e informatica, in modo da avvicinare in maniera il più possibile onesta e veritiera alla vita del ricercatore. Il linguaggio dell'informatica, infatti, si unisce a quello della matematica e alla lingua inglese e diventa un ulteriore strumento che ciascun ricercatore deve avere alla propria portata nella cassetta degli attrezzi.

Componenti

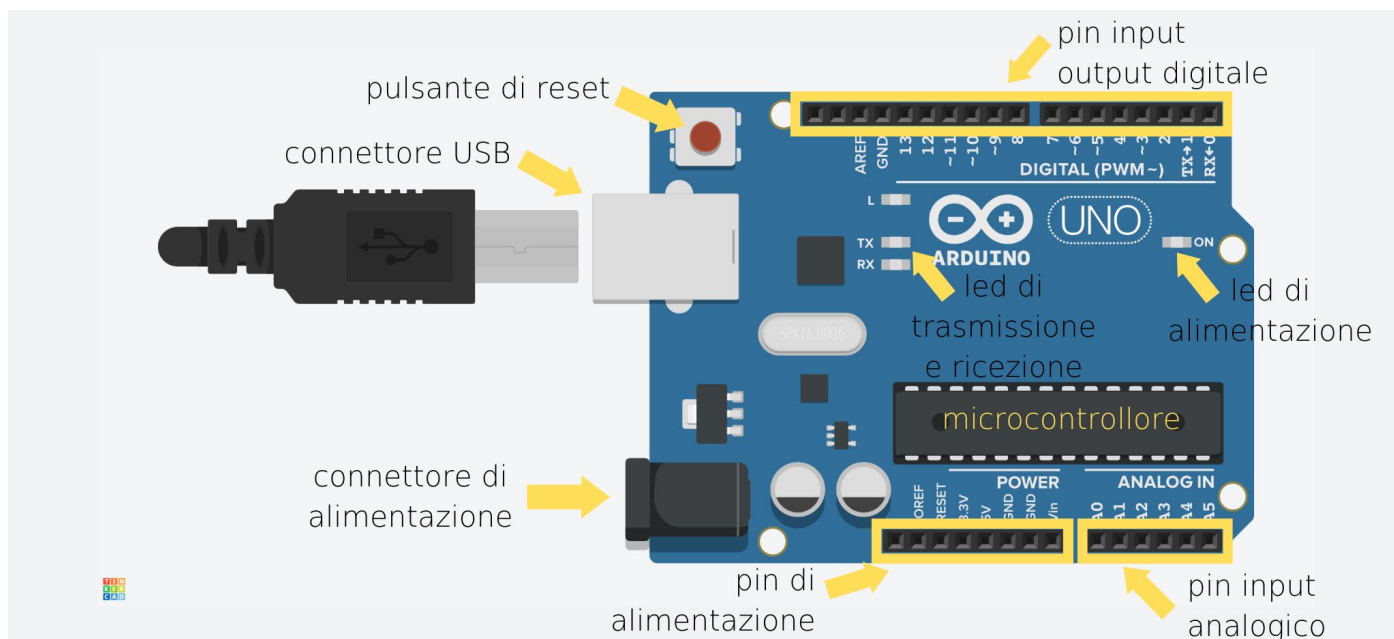
Il kit si compone di:

- 1 scheda Arduino Uno
- 1 cavo di connessione tra Arduino e il computer
- 1 breadboard
- 10 cavi jumper MM
- 4 cavi jumper MF
- 2 resistenze da 220 Ohm
(azzurre, conservate nel sacchetto dove si trova anche la fotoresistenza)
- 1 resistenza da 10 kOhm
(beige, conservata nel sacchetto dove si trova anche il sensore di peso)
- 1 supporto in legno per telaietti con filtro
- 1 supporto per breadboard
- 1 supporto in legno per cellulare
- 1 peso misterioso
- 8 telaietti con filtri colorati
- 1 piano inclinato in plastica trasparente
- 1 supporto per piano inclinato in plexiglass trasparente
- 3 strisce di diverso materiale per foderare il piano inclinato (legno, cartone e gomma)
- 1 oggetto per il piano inclinato (cubo di legno)
- 5 sensori:
 - sensore di peso FSR402 RFP602
 - sensore di distanza HY-SRF05
 - sensore di intensità luminosa LDR GL5516
 - sensore di temperatura TMP36
 - sensore di inclinazione MPU6050

Gli script necessari a svolgere le esperienze possono essere scaricati dalla pagina web <https://leveluptrento.com/kitmisura/> o inquadrando il QR code presente all'esterno della scatola.

GUIDA RAPIDA AL MICROPROCESSORE ARDUINO

Costruire un circuito con una scheda Arduino



L'oggetto in figura è una scheda Arduino Uno, il microprocessore che alimenta il circuito e che, se opportunamente programmato, è in grado di operare con varie tipologie di sensori, effettuare calcoli e compiere misurazioni anche molto precise. Le due file di fori neri in alto e in basso sono i **pin**, tramite i quali è possibile collegare la scheda a componenti esterne.

- Nella fila superiore i pin sono detti **digitali**, come indicato dalla scritta DIGITAL, e sono numerati progressivamente da 0 a 13. Alcuni numeri sono affiancati da una tilde (~), che significa che quei pin simulano l'attività dei pin analogici sotto determinate condizioni.
- Nella fila inferiore i pin sono divisi in due gruppi:
 - ◊ Quelli a sinistra sono identificati dalla scritta POWER e servono per fornire al circuito la corrente elettrica necessaria per il suo funzionamento. La corrente viene immessa a partire dai pin dove è segnata la tensione di alimentazione (5V o 3.3V) ed "esce" dal pin che rappresenta "lo zero" di riferimento per la tensione elettrica (GND). Esistono due fori identificati dal nome GND, identici come funzione; si può utilizzare indifferentemente uno dei due.
 - ◊ Nella parte destra della fila inferiore, invece, si trova un gruppo di pin identificati dalla scritta ANALOG IN, con nomi che vanno da A0 ad A6. Questi sono i pin analogici a cui è possibile collegare i sensori. Tramite questi pin, il sensore può comunicare i valori misurati ad Arduino, i quali possono poi essere utilizzati nei programmi caricati sul microprocessore.

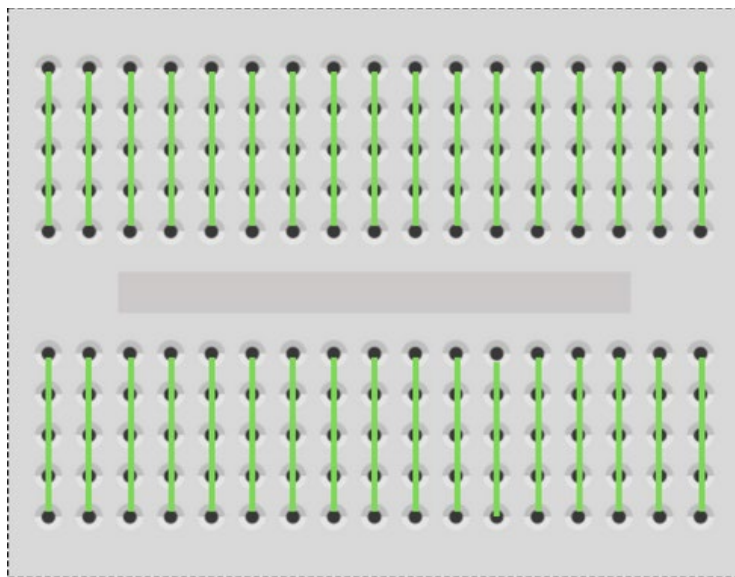
Attraverso questi pin è possibile realizzare circuiti elettrici che includono sensori e attuatori, sia di tipo digitale che analogico.

Si definiscono **sensori** quegli elementi in grado di determinare una o più caratteristiche dell'ambiente esterno e tradurle in un segnale elettrico che viene poi dato in ingresso ad Arduino. Per

questo motivo, si dice che i sensori forniscono un **input** al sistema. Si dicono invece **attuatori** quegli elementi che, ricevuto un segnale elettrico da Arduino, creano una variazione fisica nell'ambiente esterno. Per questo motivo, si dice che essi forniscono un **output**.

Grossolanamente possiamo dire che se un segnale, sia esso input od output, viene trasmesso soltanto in due modalità – acceso/spento, sì/no, alto/basso – allora si tratta di un segnale digitale. Se, al contrario, il segnale può variare in maniera graduale tra un valore minimo e un valore massimo, allora si parla di un segnale **analogico**.

Negli esperimenti proposti nelle schede successive l'input analogico dato ad Arduino è costituito dalle misurazioni effettuate dai sensori. I segnali analogici, a differenza di quelli digitali, trasmettono valori rappresentati da un numero intero che va da 0 a 1023. Questo perché sono espressi in unità di informazione, dette bit, ciascuna delle quali può assumere due possibili valori; le possibili combinazioni di 10 bit sono 1024, tante quante i numeri interi compresi tra 0 e 1023 (inclusi) che, quindi, possono essere espressi in questa modalità.



BREADBOARD

Per collegare fili, sensori, attuatori e resistenze tra loro nel modo corretto bisogna utilizzare una breadboard, ossia una tavola con dei fori collegati elettricamente secondo degli schemi ben precisi.

Esistono varie tipologie di breadboard; nel kit troverete delle **breadboard** uguali a quelle raffigurate a sinistra. I collegamenti tra i fori seguono le linee verdi.

Nel costruire i circuiti vanno tenute a mente anche alcune semplici regole.

- Un circuito, per poter funzionare, deve cominciare da Arduino e concludersi su di esso. Il punto di inizio del circuito è quello in cui viene fornita l'alimentazione, ossia dal quale si genera la corrente elettrica che scorre nel circuito: questo punto corrisponde al pin 5V di Arduino. Il punto in cui termina il circuito, ossia quello dove arriva la corrente elettrica, è invece il pin GND (si legge ground).
- Affinché la corrente elettrica scorra nel circuito senza sovraccaricarlo, questo deve contenere una **resistenza**, che agisce come la diga su un fiume rallentandone la portata. Molto spesso le resistenze sono già integrate nei sensori, ma vanno controllate bene le istruzioni date per ogni esperienza.
- I componenti del circuito vanno collegati alla breadboard tramite due fori che si trovano su file diverse, ossia non internamente connessi tra loro.
- Per evitare un cortocircuito, è sempre bene costruire il circuito **prima** di collegare Arduino al computer con l'apposito cavo, anche se le correnti in gioco sono molto basse.

I colori dei cavi nelle grafiche servono soltanto ad identificare meglio i vari cavi a che pin sono collegati, ma i cavi contenuti nel kit sono tutti identici e possono essere utilizzati colori diversi per scopi diversi senza fare alcun danno.

È oltre lo scopo di questo scritto approfondire l'elettronica che sta alla base della costruzione di circuiti; per approfondimenti e ulteriori spunti rimandiamo ad alcuni suggerimenti in bibliografia.

SOFTWARE E CARICAMENTO DEL CODICE

Per poter scrivere e caricare un programma sul microprocessore Arduino Uno, è necessario utilizzare un software apposito.

Programmazione con IDE di Arduino



Il programma IDE, identificato dall'icona visibile nella figura a sinistra, permette la scrittura di codice in linguaggio C++. È possibile utilizzare il programma in versione web <https://bit.ly/ArduinoEditorOnline>, in tal caso è necessario avere sempre una connessione internet attiva; altrimenti, è preferibile scaricare il programma sul proprio computer dal sito ufficiale <https://www.arduino.cc/en/software> (qui una guida per l'installazione: <https://www.arduino.cc/en/Guide>). Esistono due versioni del programma di cui daremo le istruzioni per caricare i codici nella maniera corretta.

Versione 1.8.*

Una volta aperto il file nel programma apposito, per caricarlo sulla scheda è indispensabile collegare Arduino al computer. Inoltre, nel menù "Strumenti" a tendina in alto (visibile in figura) è necessario controllare che alla voce "Scheda" sia selezionato "Arduino Uno"; alla voce "Porta", invece, si seleziona l'ingresso a cui è stato connesso il microprocessore.

```
dist_tantemedie_moto | Arduino 1.8.19
File Modifica Sketch Strumenti Aiuto
Formattazione automatica Ctrl+T
Archivia sketch...
Correggi codifica e ricarica
Gestione librerie... Ctrl+Maiusc+I
Monitor seriale Ctrl+Maiusc+M
Plotter seriale Ctrl+Maiusc+L
WiFi101 / WiFiNINA Firmware Updater
Scheda: "Arduino Uno"
Porta
Acquisisci informazioni sulla scheda
Programmatore: "AVRISP mkII"
Scrivi il bootloader

// Prima di c
// Poi selezi
// Caricare i
// Visualizza

// MEDIE DI M
// Questo pro
// In tutto v
// Le misure

#define TRIG
#define ECHO - // Al posto dei trattini, inserire

void setup() {
  Serial.begin(9600);
  analogReference(EXTERNAL);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  digitalWrite(TRIG, LOW);
  Serial.println("Tempo (s) Distanza (cm)");
  float distanza=0.;
  float d_somma=0.;
  float t_somma=0.;
  float t_somma=0.;
}
```

Una volta collegato correttamente il microprocessore, si può caricare il programma su di esso premendo la freccia che punta verso destra, posta in alto a sinistra nella barra degli strumenti. Il programma partirà nel momento in cui il caricamento è completo. Se ci dovessero essere errori nella scrittura del programma in fase di compilazione, appariranno delle indicazioni nella barra di stato in basso.



Per visualizzare i dati che il sensore sta misurando, si può aprire il monitor seriale partendo dal menù in alto: *Strumenti* > *Monitor seriale* (visibile nella figura precedente, 3 righe sopra quella evidenziata in azzurro), oppure cliccando il tasto con la lente d'ingrandimento in figura, in alto a destra sotto il tasto di chiusura della finestra.

Per visualizzare i dati sotto forma di grafico, esiste anche la modalità "Plotter" che si può aprire da *Strumenti* > *Plotter seriale*.

Versione 2.*.*

Una volta aperto il file nel programma apposito, per caricarlo sulla scheda è indispensabile collegare Arduino al computer. Nella parte in alto a sinistra del software si trova un menù a tendina che permette di indicare la porta e la scheda collegata. Il menù si aggiorna in automatico una volta che si è collegato Arduino Uno al pc con l'apposito cavo.

```
IntensitLuminosa_test | Arduino IDE 2.0.2
File Edit Sketch Tools Help

Select Board
  ↳ Arduino Uno COM6
  Select other board and port...

1 # _____ nome del pin selezionato che viene letto dal circuito
2 v
3 Select other board and port...
4 _____ la comunicazione con il computer
5 }
6 void loop() { //qui va inserito il codice che viene ripetuto in continuazione
7   int i=0; //definizione della variabile intera i, usata come contatore
8   for (i; i<400; i++) { //apertura del ciclo for che ripete per un numero massimo definit
9     int a0val = analogRead(pinIn); //al posto dei trattini va inserito nome del pin selez
10    delay(100); //attesa
11    Serial.println(a0val); //scrive sul monitor seriale il valore misurato
12  }
13  while (1) {
14    // non fa nulla, un modo per fermare le ripetizioni
15  }
```

Una volta collegato correttamente il microprocessore, si può caricare il programma su di esso premendo la freccia che punta verso destra, posta in alto a sinistra nella barra degli strumenti. Il programma partirà nel momento in cui il caricamento è completo. Se ci dovessero essere errori nella scrittura del programma in fase di compilazione, appariranno delle indicazioni nella barra di stato in basso.

Per visualizzare i dati che il sensore sta misurando, si può aprire il monitor seriale partendo dal menù in alto: *Strumenti* > *Monitor seriale*, oppure cliccando il tasto con la lente d'ingrandimento in figura, che si trova in alto a destra sotto il tasto di chiusura della finestra.



Per visualizzare i dati sotto forma di grafico, esiste anche la modalità "Plotter" che si può aprire da *Strumenti* > *Plotter seriale* o dalla apposita icona.

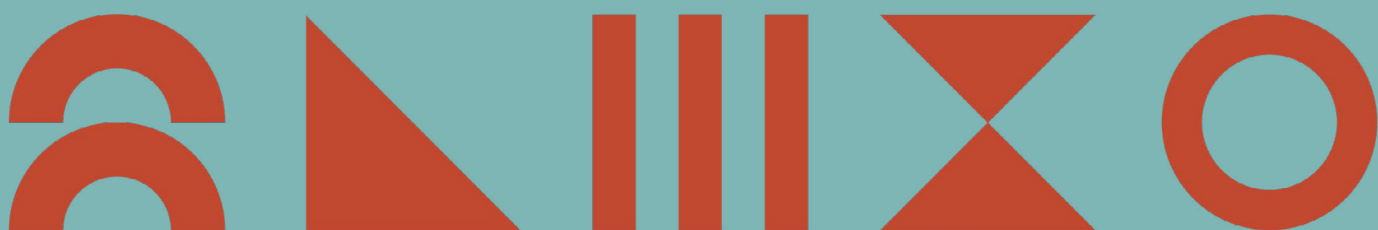
Alcuni comandi utili

<code>Serial.begin(9600)</code>	permette la comunicazione tra Arduino e il computer
<code>Serial.print()</code>	stampa il valore tra parentesi sul monitor seriale
<code>Serial.println()</code>	stampa sul monitor seriale e manda a capo
<code>analogRead()</code>	lettura del pin analogico
<code>delay()</code>	una pausa espressa in millisecondi
<code>s += a</code>	aggiunge al valore della variabile s il nuovo valore della variabile a (per le sommatorie)
<code>pinMode(numeroPin, in/out)</code>	comunica con un pin digitale definendone il numero e se è un pin di uscita o di entrata
<code>int</code>	definisce le variabili intere
<code>float</code>	definisce le variabili a virgola mobile
<code>digitalWrite(numeroPin, low/high)</code>	funzione che assegna ad un certo pin digitale (primo posto) un valore di alto o basso
<code>digitalRead(numeroPin)</code>	legge il valore di uno specifico pin digitale
<code>unsigned long</code>	definisce una variabile di lunghezza estesa per memorizzare un numero
<code>pulseIn(numeroPin, low/high)</code>	legge l'impulso in entrata di un pin, quando riceve il segnale specificato al secondo posto fa partire un timer in microsecondi e viene fermato quando il segnale cambia
<code>micros()</code>	Mostra il numero di microsecondi da quando Arduino ha fatto partire il programma, non servono parametri tra le parentesi
<code>analogReference(EXTERNAL)</code>	Configura il voltaggio di riferimento usato nell'input analogico, che può essere quindi anche diverso da 5V. Nel caso di EXTERNAL il voltaggio applicato al pin AREF sarà quello usato come riferimento e deve essere di un valore compreso tra 0 e 5 V.
<code>for (n=0; n<100; n++) {...}</code>	ciclo for con contatore n che parte da zero e arriva fino a 99, aggiungendo 1 ad ogni ripetizione

Per una lista più completa di comandi potete consultare il sito di Arduino:

<https://www.arduino.cc/reference/en/>

GUIDA ALL'USO DEI SENSORI



INTENSITÀ LUMINOSA



INTENSITÀ LUMINOSA

Materiali necessari:

- scheda Arduino Uno
- breadboard
- fotoresistenza
- resistenza da 220 Ohm
- cavi jumper MM
- cavo di collegamento tra computer e Arduino
- fonte di luce (torcia o cellulare)

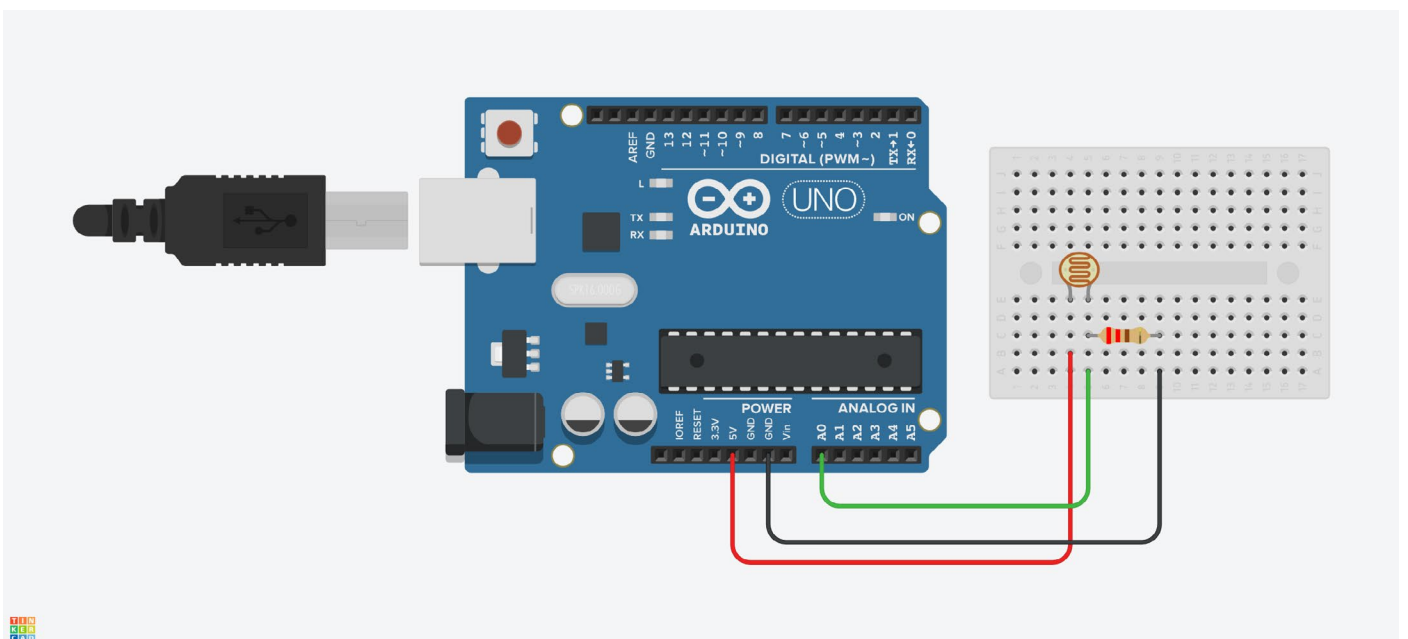


La fotoresistenza è il sensore che misura l'intensità luminosa ricevuta.

Il circuito da costruire per utilizzare la fotoresistenza è piuttosto semplice ed è rappresentato nella figura sotto:

1. si collega un cavo (rosso in figura) al pin 5V di Arduino e a un foro qualsiasi della breadboard;
2. si collega la fotoresistenza a un altro foro della stessa colonna (in figura la colonna numero 4) e al foro adiacente (avente il numero successivo 5);
3. si collega la resistenza da 220 Ohm a un altro foro di quest'ultima colonna (5) e a un foro di una terza colonna (per esempio 9). In questo modo, la corrente scorrerà prima attraverso la fotoresistenza e poi attraverso la resistenza: si dice quindi che la resistenza e la fotoresistenza sono collegate in serie;
4. si collega un cavo (nero in figura) a un altro foro della colonna dove arriva la resistenza (la 9 nel nostro caso) e al pin GND di Arduino;
5. si connette un ultimo cavo (verde in figura) alla colonna di fori dove sono collegate sia la resistenza che la fotoresistenza (in figura la 5) e a un qualsiasi pin analogico di Arduino (per esempio A0). Questo terzo collegamento è necessario per poter leggere ciò che la fotoresistenza misura.

Una volta che il circuito è costruito e pronto, si può testare ed esplorarne il funzionamento utilizzando un breve programma.



Programmazione con IDE di Arduino

Collegandosi al sito <https://leveluptrento.com/kitmisura/> si possono scaricare alcuni script (file di codice) scritti in linguaggio C++ che possono essere usati per questo sensore, anche modificandoli a piacere.



Il primo script utile, chiamato *IntLum_tantemisure.ino*, permette di testare il funzionamento del circuito e capire come funziona il sensore. Il file va scaricato sul pc e aperto con il programma IDE di Arduino. Selezionando i parametri desiderati all'interno del programma, caricando il programma sulla scheda Arduino e aprendo il monitor seriale, si può vedere un certo numero di misure effettuate dal sensore in quell'istante. **Leggere attentamente i commenti del codice** per poter capire dove e cosa modificare in caso di necessità (per esempio, si può cambiare il nome del pin analogico a cui è collegata la fotoresistenza); i commenti sono le righe di codice precedute dalle doppie barre //.

Sensori

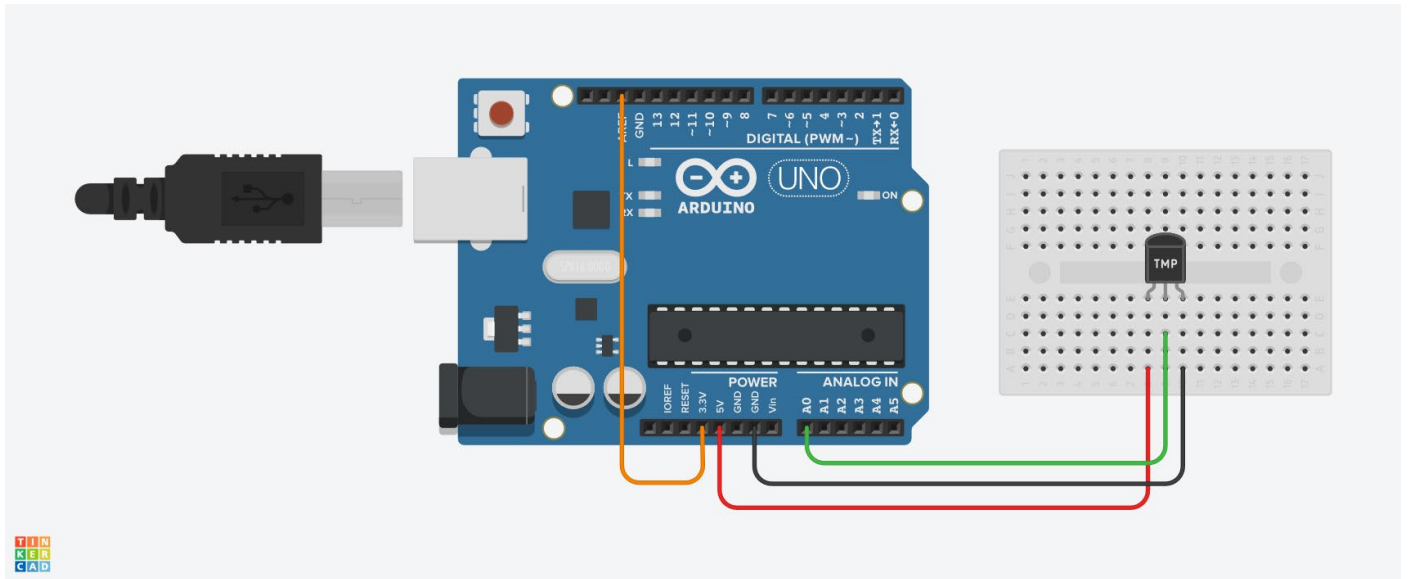
TEMPERATURA



TEMPERATURA

Materiali necessari:

- scheda Arduino Uno
- breadboard
- sensore di temperatura TMP36
- cavi jumper MM
- cavo di collegamento tra computer e Arduino



Il sensore di temperatura TMP36 è un sensore analogico simile a quello che vedete nella figura. Per costruire il circuito, che faccia funzionare il sensore di temperatura, bisogna seguire questi step:

1. si collegano i tre piedini del sensore a tre fori adiacenti della breadboard su tre colonne diverse (per esempio, 8, 9 e 10);
2. si collega un cavo (rosso in figura) al pin 5V di Arduino e a un foro nella stessa colonna del primo piedino del sensore (in figura la 8). Per “primo piedino” si intende quello più a sinistra quando il sensore è rivolto con la parte piatta di fronte a noi;
3. si collega un secondo cavo (verde in figura) a un pin analogico di Arduino (per esempio A0, ma può essere scelto liberamente, basta poi inserire il giusto numero identificativo nel codice) e a un foro collegato al piedino centrale del sensore, sulla stessa colonna (in figura la 9);
4. si collega un terzo cavo (nero in figura) al pin GND di Arduino e a un foro collegato all'ultimo piedino del sensore, sulla stessa colonna (in figura la 10);
5. si collega un ultimo cavo (arancione in figura) tra il pin 3.3V e il pin AREF di Arduino.
6. Una volta che il circuito è costruito e pronto, si può testare ed esplorarne il funzionamento utilizzando un breve programma.

Attenzione: è importante collegare il sensore nel verso corretto. Il piedino a cui va collegato il cavo che parte da 5V su Arduino deve essere quello di sinistra quando si osserva il sensore con la parte piatta rivolta verso di sé.

Programmazione con IDE di Arduino

Collegandosi al sito <https://leveluptrento.com/kitmisura/> si possono scaricare alcuni script (file di codice) scritti in linguaggio C++ che possono essere usati per questo sensore, anche modificandoli a piacere.



Il primo script utile, chiamato *Temp_tantemisure.ino*, permette di testare il funzionamento del circuito e capire come funziona il sensore. Il file va scaricato sul pc e aperto con il programma IDE di Arduino. Selezionando i parametri desiderati all'interno del programma, caricando il programma sulla scheda Arduino e aprendo il monitor seriale, si può vedere un certo numero di misure effettuate dal sensore in quell'istante. **Leggere attentamente i commenti del codice** per poter capire dove e cosa modificare in caso di necessità (per esempio, si può cambiare il nome del pin analogico a cui è collegata il sensore); i commenti sono le righe di codice precedute dalle doppie barre //.

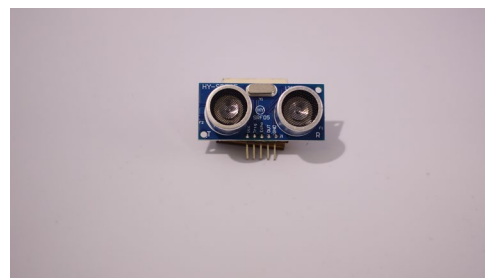
DISTANZA



DISTANZA

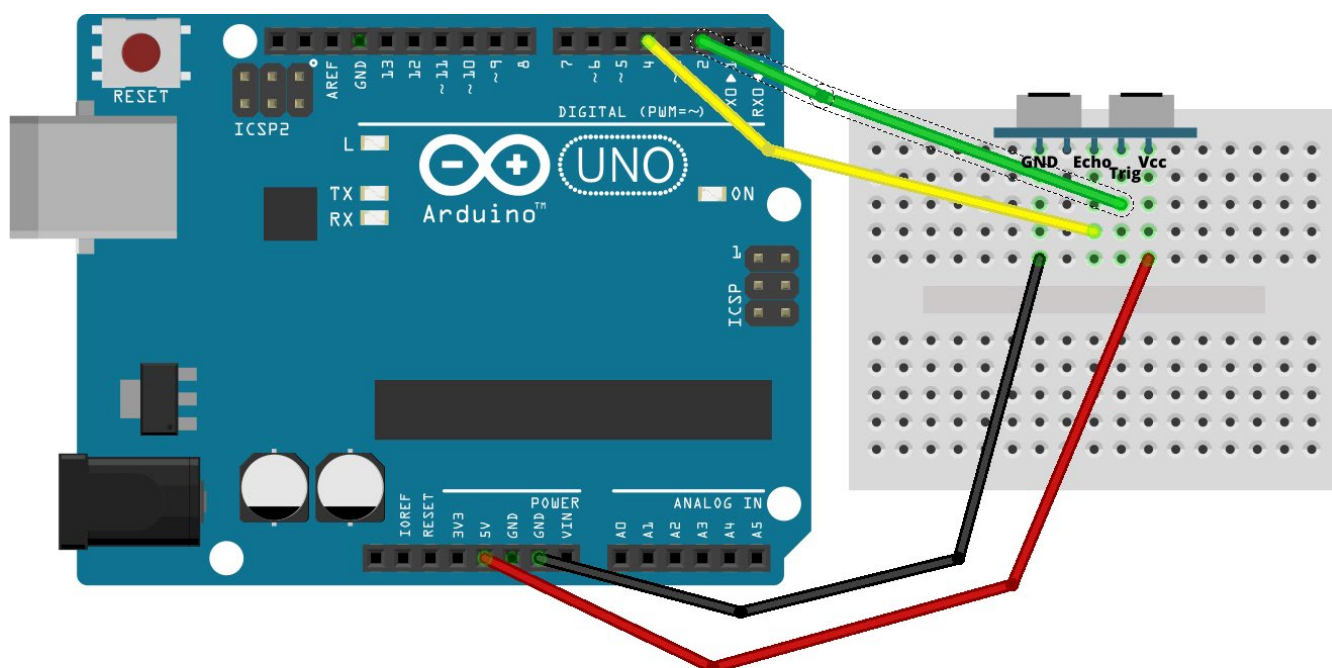
Materiali necessari:

- scheda Arduino Uno
- breadboard
- sensore ad ultrasuoni HC-SR04
- cavi jumper MM
- cavo di collegamento tra computer e Arduino



Il sensore a ultrasuoni è uno strumento utilizzato per calcolare la distanza dagli oggetti tramite le onde sonore, emesse a una frequenza non percepibile dall'orecchio umano. Il sensore ha 5 uscite, VCC, TRIG, ECHO, OUT e GND, ognuna delle quali corrisponde a un piedino. Per costruire il circuito che faccia funzionare il sensore a ultrasuoni, rappresentato nella figura sotto, bisogna seguire questi step:

1. Si collegano i piedini ai fori della breadboard, in modo che si trovino tutti sulla stessa riga e su cinque colonne adiacenti (per esempio, 7, 8, 9, 10 e 11).
2. Dopodiché si collega un primo cavo (rosso in figura) al pin 5V di Arduino e a un foro sulla stessa colonna (in figura la colonna 11) di VCC, il quale dà l'energia necessaria al funzionamento del sensore.
3. Un secondo cavo (verde in figura) si collega a uno dei pin digitali della scheda e a un foro nella stessa colonna (in figura la 10) del TRIG, il trigger, che riceve il segnale per far partire l'onda sonora.
4. A un altro dei pin digitali va collegato il terzo cavo (giallo in figura), la cui altra estremità va connessa a un foro sulla stessa colonna (in figura la 9) di ECHO, che manda il segnale di interruzione del cronometro quando l'onda ritorna alla sorgente.
5. Un ultimo cavo (nero in figura) va collegato al pin GND della scheda Arduino e, con l'altra estremità, a un foro sulla stessa colonna (in figura la 7) di GND, in modo da chiudere il circuito e permettere il passaggio della corrente.
6. La colonna dove è collegato all'uscita denominata OUT del sensore (in figura la colonna 8) va lasciata libera. Non ci sono quindi collegamenti tra Arduino e questa colonna.

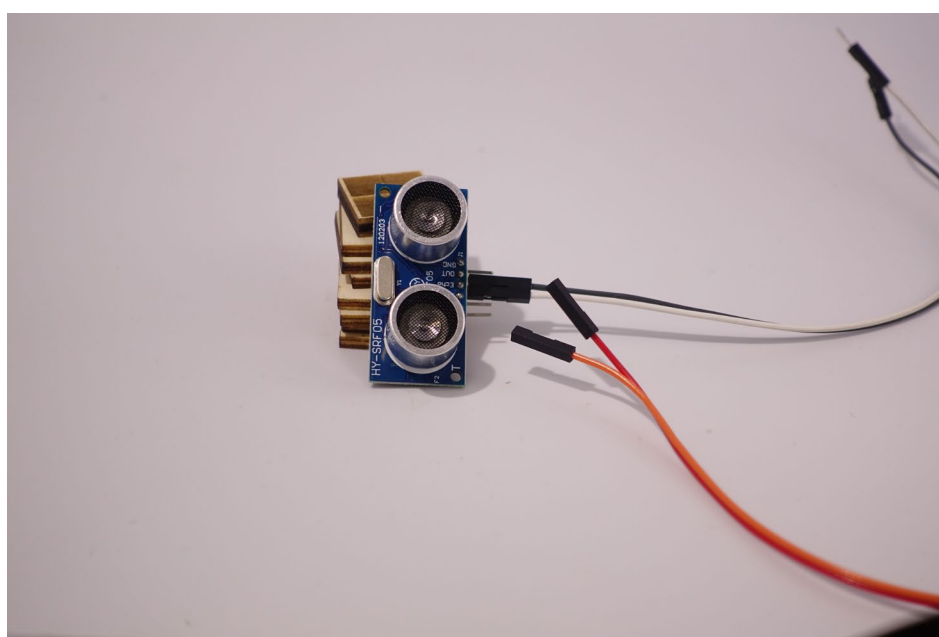


Una volta che il circuito è costruito e pronto, si può testare ed esplorarne il funzionamento utilizzando un breve programma.

N.B: Evitare di usare i pin digitali 1 e 2. Qui sotto una tabella che mostra i collegamenti corretti con ogni pin.

SENSORE	Vcc	Trig	Echo	OUT	GND
ARDUINO	5V	pin digitale qualsiasi	pin digitale qualsiasi	-	GND

Per alcune esperienze potrebbe essere richiesto di collegare il sensore di distanza con cavi MF, senza collegare il sensore alla breadboard, ma collegando le uscite direttamente ai pin di Arduino. In figura un esempio di questa configurazione.



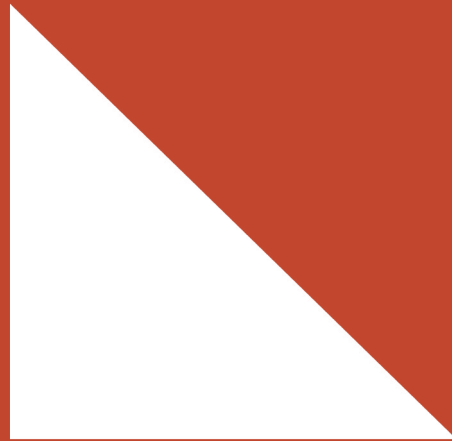
Programmazione con IDE di Arduino

Collegandosi al sito <https://leveluptrento.com/kitmisura/> si possono scaricare alcuni script (file di codice) scritti in linguaggio C++ che possono essere usati per questo sensore, anche modificandoli a piacere.



Il primo script utile, chiamato *Dist_tantemisure.ino*, permette di testare il funzionamento del circuito e capire come funziona il sensore. Il file va scaricato sul pc e aperto con il programma IDE di Arduino. Selezionando i parametri desiderati all'interno del programma, caricando il programma sulla scheda Arduino e aprendo il monitor seriale, si può vedere un certo numero di misure effettuate dal sensore in quell'istante. **Leggere attentamente i commenti del codice** per poter capire dove e cosa modificare in caso di necessità (per esempio, si può cambiare il nome dei pin analogici a cui è collegato il sensore); i commenti sono le righe di codice precedute dalle doppie barre //.

INCLINAZIONE



INCLINAZIONE

Materiali necessari:

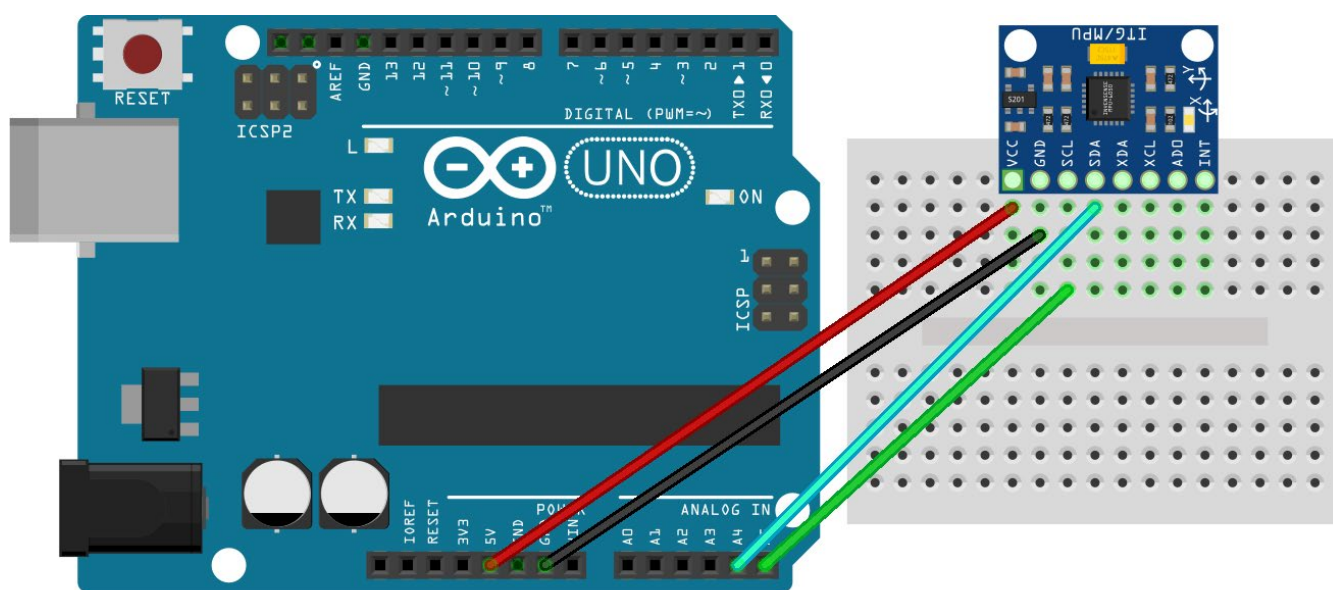
- scheda Arduino Uno
- breadboard
- MPU6050
- cavi jumper MF
- cavo di collegamento tra computer e Arduino



La MPU6050 è un sensore che ha molteplici utilizzi in un singolo modulo: un accelerometro ed un giroscopio a 3 assi. Con il giroscopio si può misurare l'accelerazione angolare di un corpo, mentre con l'accelerometro si può misurare l'accelerazione di un corpo lungo una direzione ben definita. In questo kit verrà usato soltanto come inclinometro, ossia come strumento per misurare l'angolo di inclinazione di una superficie rispetto al terreno.

Per poterlo mettere in funzione il circuito da costruire è il seguente:

1. Si collegano i piedini ai fori della breadboard, in modo che si trovino tutti sulla stessa riga e su otto colonne adiacenti (per esempio, 6, 7, 8, 9, 10, 11, 12 e 13).
2. Dopodiché si collega un primo cavo (rosso in figura) al pin 5V di Arduino e a un foro sulla stessa colonna (in figura la 6) di VCC, il quale dà l'energia necessaria al funzionamento del sensore.
3. Un secondo cavo (nero in figura) va collegato al pin GND della scheda Arduino e, con l'altra estremità, a un foro sulla stessa colonna (in figura la 7) di GND, in modo da chiudere il circuito e permettere il passaggio della corrente.
4. Un terzo cavo (verde in figura) si collega al pin analogico A5 della scheda e a un foro nella stessa colonna (in figura la 8) chiamato SCL.
5. Il pin digitale A4 invece va collegato tramite il quarto cavo (azzurro in figura), la cui altra estremità va connessa a un foro sulla stessa colonna (in figura la 9) chiamato SDA.
6. Le altre quattro uscite della MPU6050 vanno lasciate libere, senza cavi di collegamento con Arduino (in figura le colonne 10, 11, 12 e 13).



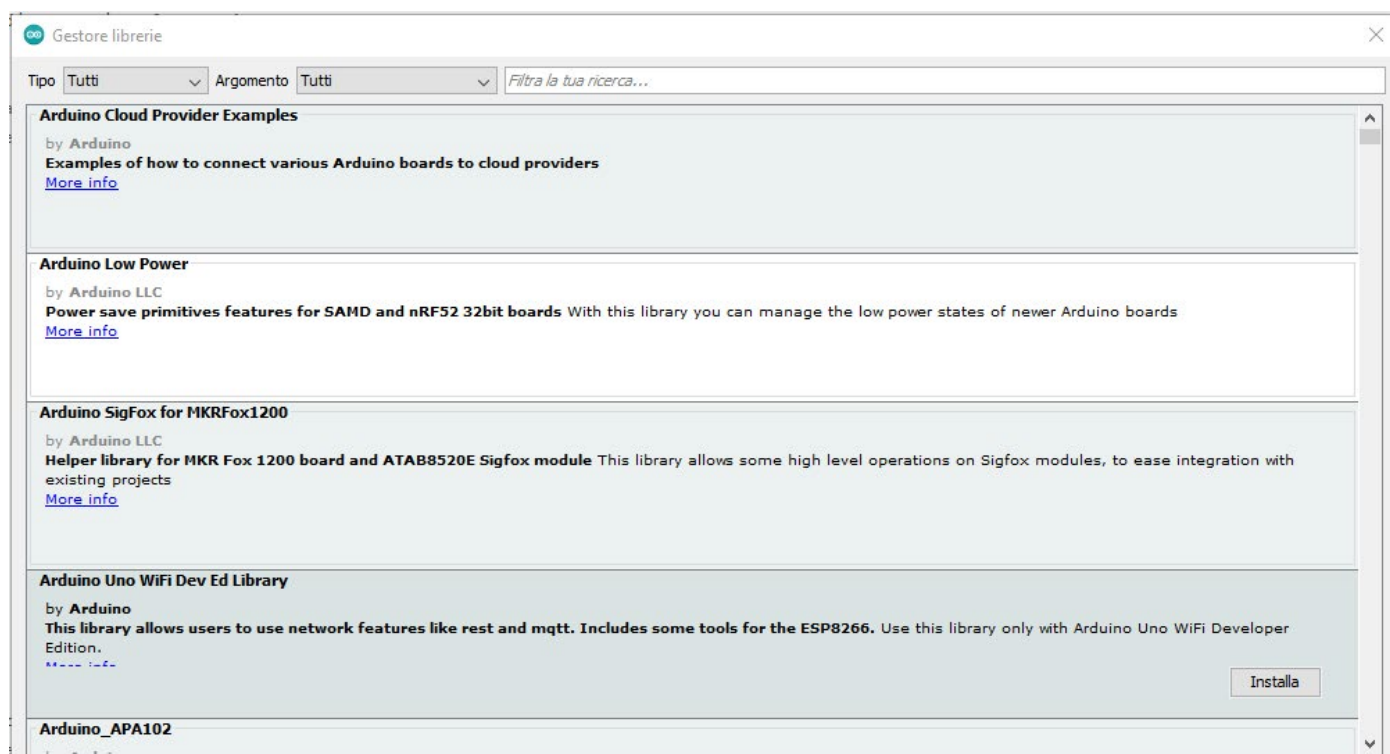
Una volta che il circuito è costruito e pronto, si può testare ed esplorarne il funzionamento utilizzando un breve programma.

Per alcune esperienze potrebbe essere richiesto di collegare il sensore di inclinazione con cavi MF, come per il sensore a ultrasuoni spiegato nelle pagine precedenti.

Programmazione con IDE di Arduino

Per poter utilizzare questo sensore è necessario scaricare alcune librerie. Per fare ciò si procede aprendo nella barra in alto il menù *Strumenti* > *Gestione librerie...* e si apre una seconda finestra, come in figura.

Si attende che il catalogo delle librerie venga aggiornato e nella barra di ricerca in alto a destra digitare il nome della libreria *MPU6050_tockn* by *tockn*, versione 1.5.2 e si scarica schiacciando il pulsante **Installa** che compare nel riquadro della libreria scelta in basso a destra. Successivamente si cerca la libreria *OneWire* by *Paul Stoffregen*, versione 2.3.7 e si installa come fatto precedentemente.

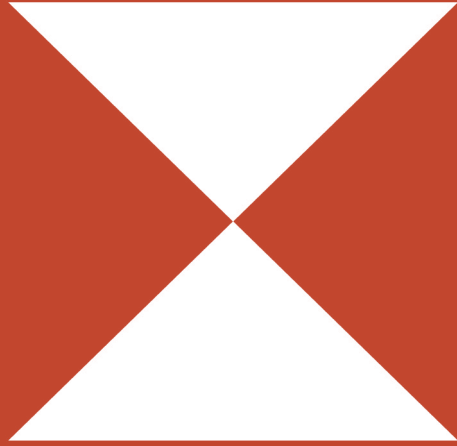


Collegandosi al sito <https://leveluptrento.com/kitmisura/> si possono scaricare alcuni script (file di codice) scritti in linguaggio C++ che possono essere usati per questo sensore, anche modificandoli a piacere.



Il primo script utile, chiamato *Giro_tantemisure.ino*, permette di testare il funzionamento del circuito e capire come funziona il sensore. Il file va scaricato sul pc e aperto con il programma IDE di Arduino. Selezionando i parametri desiderati all'interno del programma, caricando il programma sulla scheda Arduino e aprendo il monitor seriale, si può vedere un certo numero di misure effettuate dal sensore in quell'istante. **Leggere attentamente i commenti del codice** per poter capire dove e cosa modificare in caso di necessità. Il programma stampa a schermo le misure di inclinazione già espresse in gradi, che corrispondono alla rotazione lungo l'asse indicato come y sul sensore stesso.

PESO

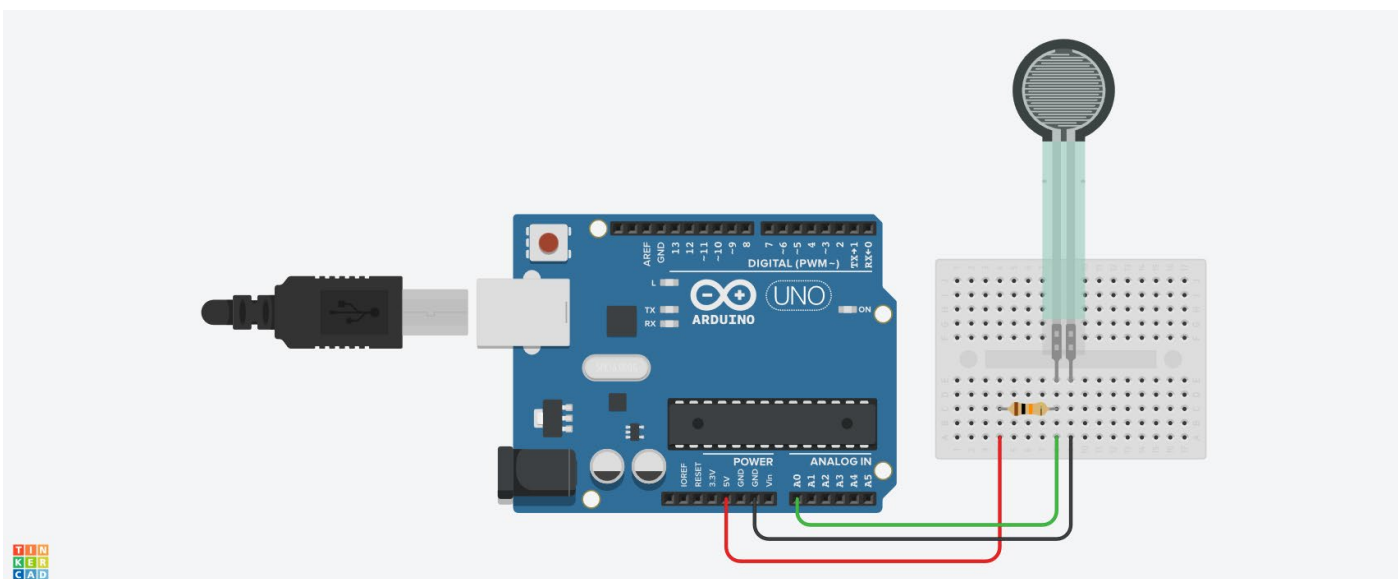


Materiali necessari:

- scheda Arduino Uno
- breadboard
- sensore di peso
- resistenza da 10 kOhm
- cavi jumper MM



Il sensore di peso è una resistenza che cambia valore in base alla quantità di pressione che viene applicata sulla sua superficie e perciò è in grado di dare una stima del peso dell'oggetto che viene appoggiato sulla superficie sensibile.



Il circuito da costruire è il seguente:

1. si collega un cavo al pin 5V di Arduino e a un foro della breadboard (in figura la colonna 4);
2. si collega la resistenza da 10kΩ a un altro foro nella stessa colonna (in figura la 4) e al foro di un'altra colonna (in figura la colonna 8);
3. si collega il sensore di peso a due fori in due colonne adiacenti (in figura le colonne 8 e 9). In questo modo, la corrente scorrerà prima attraverso la resistenza e poi attraverso il sensore di peso: si dice quindi che sono collegati in serie;
4. si collega un ultimo cavo a un altro foro della colonna dove arriva il sensore di peso e al pin GND di Arduino (in figura la 9);
5. si connette un ultimo cavo alla colonna di fori dove sono collegate sia la resistenza che il sensore di peso (in figura la colonna 8) e a un qualsiasi pin analogico di Arduino (ad esempio A0). Questo terzo collegamento è necessario per poter leggere ciò che il sensore misura.

I collegamenti sono mostrati nella figura sopra.

Una volta che il circuito è costruito e pronto, si può testare ed esplorarne il funzionamento utilizzando un breve programma.

Per alcune esperienze potrebbe essere richiesto di collegare il sensore di peso con cavi MF, come per il sensore di distanza o il sensore di inclinazione, spiegati nelle pagine precedenti.

ATTENZIONE: il sensore di peso ha una superficie di funzionamento molto piccola, per questo si

consiglia di utilizzare un bicchiere e del materiale che può essere versato al suo interno in maniera graduale; come sabbia, zucchero o farina. Posizionare il bicchiere sopra il supporto in legno che si trova nel kit, facendo attenzione a farlo aderire per bene alla superficie del sensore. Mano a mano che il bicchiere viene riempito e misurato con il sensore si può confrontare il peso con una bilancia da cucina, più è alta la sua precisione meglio sarà.

Non superare i 9 kg di peso sul sensore.

Programmazione con IDE di Arduino

Collegandosi al sito <https://leveluptrento.com/kitmisura/> si possono scaricare alcuni script (file di codice) scritti in linguaggio C++ che possono essere usati per questo sensore, anche modificandoli a piacere.



Il primo script utile, chiamato *Peso_tantemisure.ino*, permette di testare il funzionamento del circuito e capire come funziona il sensore. Il file va scaricato sul pc e aperto con il programma IDE di Arduino. Selezionando i parametri desiderati all'interno del programma, caricando il programma sulla scheda Arduino e aprendo il monitor seriale, si può vedere un certo numero di misure effettuate dal sensore in quell'istante. **Leggere attentamente i commenti del codice** per poter capire dove e cosa modificare in caso di necessità (per esempio, si può cambiare il nome del pin analogico a cui è collegato il sensore); i commenti sono le righe di codice precedute dalle doppie barre //.

BIBLIOGRAFIA

John R Taylor "Introduzione all'analisi degli errori. Lo studio delle incertezze nelle misure fisiche" Zanichelli editore, 1999

Paolo Aliverti "Il manuale di Arduino. Guida completa" Edizioni LSWR, 2016

Maurizio Giaffredo "Elettronica a scuola con Arduino e Tinkercad" Rizzoli Education, 2022

Charles K. Alexander, Matthew N. O. Sadiku, Giambattista Gruosso, Giancarlo Storti Gajani "Circuiti elettrici" McGraw Hill Education, 2017

Giovanni Organtini "Fisica con Arduino" Zanichelli editore, 2021

Massimiliano Virdis "Laboratorio di fisica con Arduino", 2020

Sito ufficiale Arduino: Progetti di Fisica con arduino Uno
<https://create.arduino.cc/projecthub/projects/tags/physics>

Data sheet dei vari sensori:

sensore di peso FSR402 RFP602

<https://www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR402-Layout2.pdf>

sensore di distanza HY-SRF05

<https://datasheetspdf.com/datasheet/HY-SRF05.html>

sensore di intensità luminosa LDR GL5516

<https://datasheetspdf.com/datasheet/GL5516.html>

sensore di temperatura TMP36

https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

sensore di inclinazione MPU6050

<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>