Italy

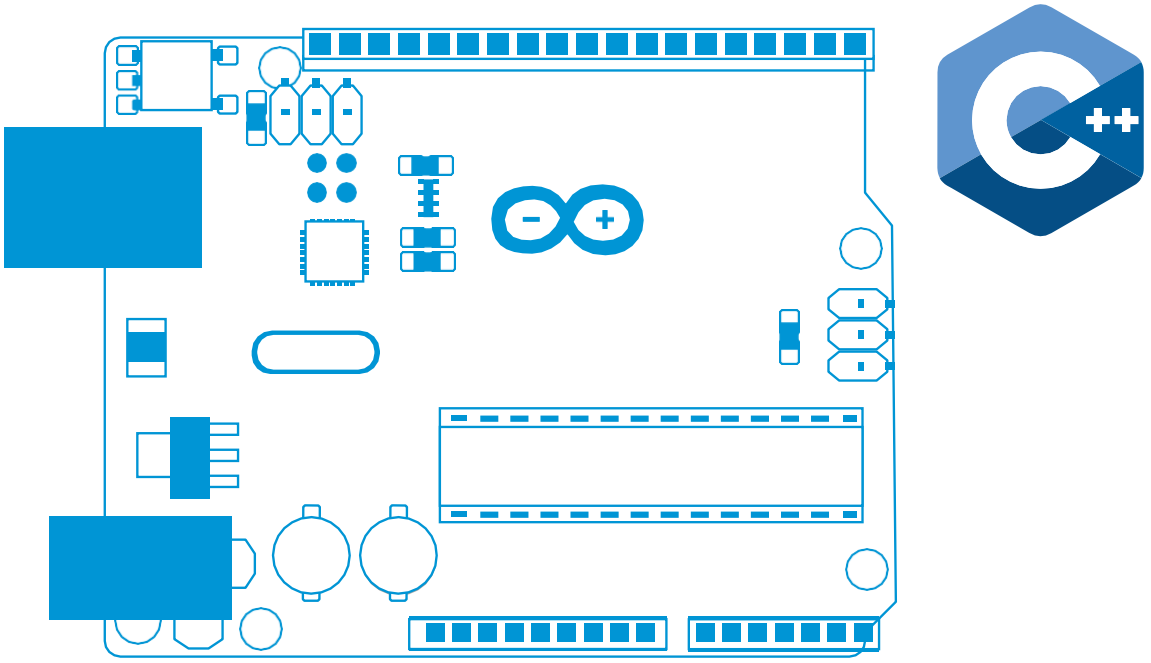
UNA COLLABORAZIONE TRA

Tecnologia | T04.1

A scuola con lo spazio

→ TI PRESENTO ARDUINO!

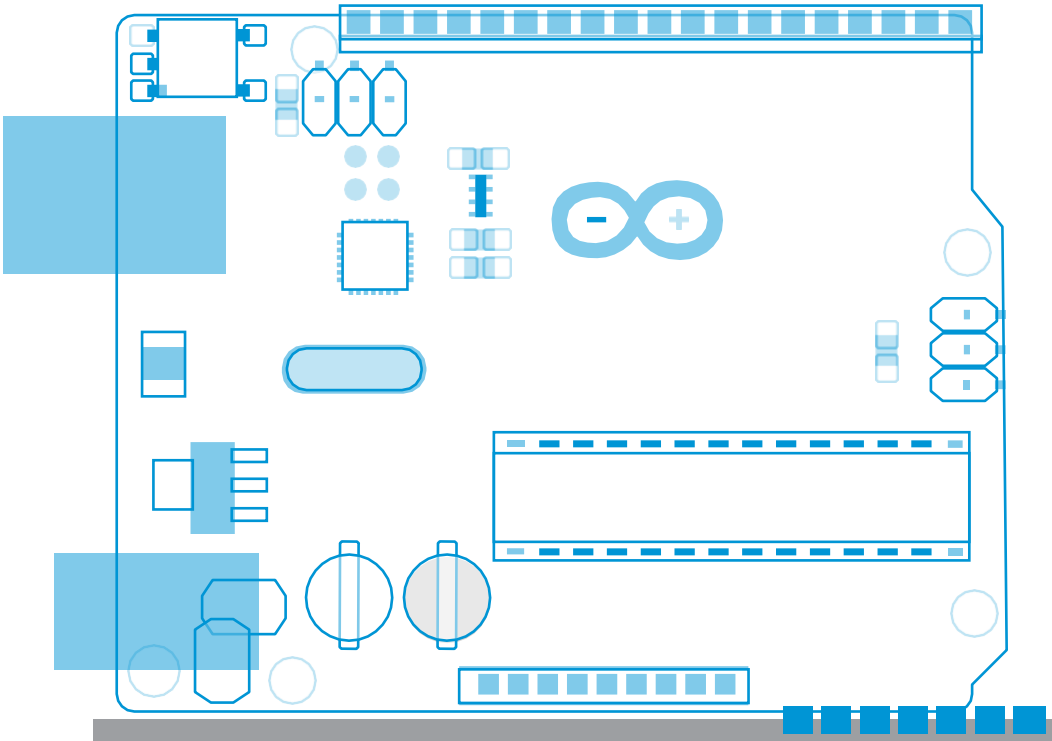
Introduzione alla programmazione di Arduino in C++



Guida insegnante & appunti studenti

Traduzione e adattamento da parte di ESERO Italia

CON



A scuola con lo spazio – incontrare arduino | T04.1
www.esa.int/education

L'ufficio ESA Education accoglie feedback e commenti a
education@esa.int

Una produzione ESA Education

→ TI PRESENTO ARDUINO!

Introduzione alla programmazione di Arduino in C++

Caratteristiche principali

Età: 14-20 anni

Argomenti curricolari: programmazione, elettronica

Difficoltà: intermedia

Durata: 90-120 minuti

Luogo: in classe

Comprende l'uso di: software e hardware Arduino

Parole chiave: Arduino, sensori, codice

Breve descrizione

Gli studenti esploreranno la tecnologia usata nello spazio attraverso la piattaforma di Arduino. Costruiranno circuiti per far lampeggiare un LED e per misurare temperatura, pressione e altezza. I concetti base di programmazione in C++ verranno introdotte nell'utilizzo della IDE (Integrated Developement Environment) di Arduino. Questo insieme di attività potrebbe essere il punto di partenza per partecipazioni future alla competizione CanSat.

Obiettivi di apprendimento

- Migliorare le capacità analitiche
- Capire un linguaggio di programmazione
- Capire la costruzione dei circuiti
- Capire come i sensori possano essere usati per raccogliere dati
- Migliorare le capacità di lavoro in gruppo

Attività - Contenuti e risultati					
	Title	**Description**	**Outcome**	**Requirements**	**Time**
0	Per cominciare	Un'introduzione ai componenti usati durante l'attività	Gli studenti prenderanno familiarità con i componenti necessari ed il loro funzionamento	Conoscenza di base delle componenti elettroniche	10 minuti
1	Fammi lampeggiare!	Gli studenti costruiranno il loro primo circuito con Arduino	Gli studenti saranno in grado di controllare un LED con il codice che hanno scritto	Aver completato l'attività precedente	15 minuti
2	Mandare un S.O.S	Gli studenti impareranno come mandare messaggi complessi usando un LED	Gli studenti saranno in grado di scrivere e programmare il LED a mandare un S.O.S.	Aver completato l'attività precedente	15 minuti
3	Misurare la temperatura	Gli studenti useranno un termistore per misurare la temperatura del loro ambiente	Gli studenti saranno in grado di costruire circuiti più complessi e di usare questi per fare misure di ambiente	Aver completato l'attività precedente	20 minuti
4	Misurare la pressione	Agli studenti è mostrato come usare un sensore di pressione per misurare la pressione dell'aria locale	Gli studenti saranno in grado di costruire circuiti più complessi e di usare questi per fare misure di ambiente	Aver completato l'attività precedente	20 minuti
5	Misurare l'altitudine	Agli studenti è mostrato come determinare l'altezza a partire da misure locali di pressione	Gli studenti saranno in grado di costruire circuiti con sicurezza e di manipolare il codice per prendere misure di ambiente	Aver completato l'attività precedente	20 minuti

→ Attività 0: Per cominciare

Introduzione

Arduino è una piattaforma open-source usata per costruire una vasta varietà di progetti elettronici, ciò include la possibilità di simulare missioni spaziali reali come ExoMars (http://exploration.esa.int/mars/).
Arduino incorpora sia una scheda fisica e programmabile (hardware) che del software (IDE) che gira su un computer.

La piattaforma hardware e software edi Arduino fu ideata per artisti, designers, amatori, hackers e persone interessate a creare oggetti ed ambienti interattivi. Arduino può essere usato per interagire con bottoni, LEDs, sensori, motori, casse audio, unità GPS, telecamere, internet, e anche i vostri smartphone o TV! Questa flessibilità, combinata con il fatto che il software di Arduino è gratuito; la scheda hardware ed i sensori relativamente economici; e sia il software che l'hardware intuitivi, ha portato ad una vasta comunità di utenti ((www.arduino.cc).

Hardware

Per questa attività useremo Arduino Uno (Figura A1, box numero h). Arduino può fornire voltaggi sui suoi pin (piedini metallici) e leggere voltaggi da essi. Notate i numeri stampati accanto ad ogni pin, con **GND** (terra) che si comporta come il polo negativo e **Vin o 5V** che si comporta come il polo positivo (come il positivo e negativo di una batteria).

Useremo le seguenti componenti mostrate nella figura A1:

a. **Breadboard**: Una struttura di supporto per connettere facilmente le componenti elettroniche.

b. **Sensore di pressione MPX4115A**: Un componente che misura la pressione assoluta di ambiente.

c. **Resistenze di 220 Ohm e 10 KOhm**: Per controllare il flusso di corrente. La resistenza si misura in ohms. Le resistenze hanno delle bande colorate su di esse che indicano il loro valore.

d. **Cavo USB**: Usato per connettere Arduino al computer o alla corrente.

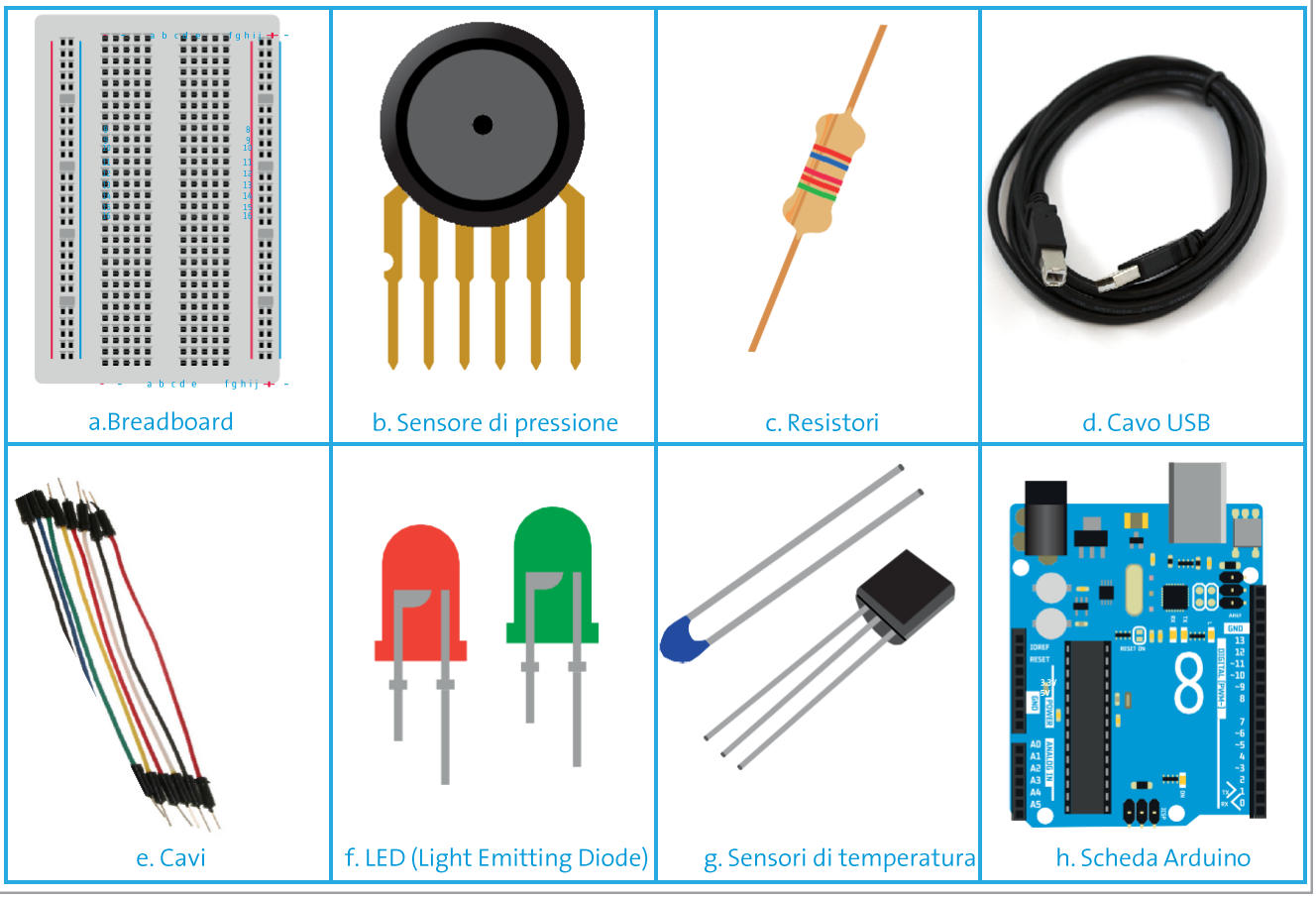
e. **Cavi**: usati per connettere elettricamente le componenti.

f. **LEDs (Light Emitting Diode)**: Componenti elettroniche che generano luce quando l'elettricità passa attraverso di essi

g. **Sensori di temperatura**: Un sensore analogico (come un LM35) e un termistore (una resistenza che è particolarmente sensibile alla temperatura).

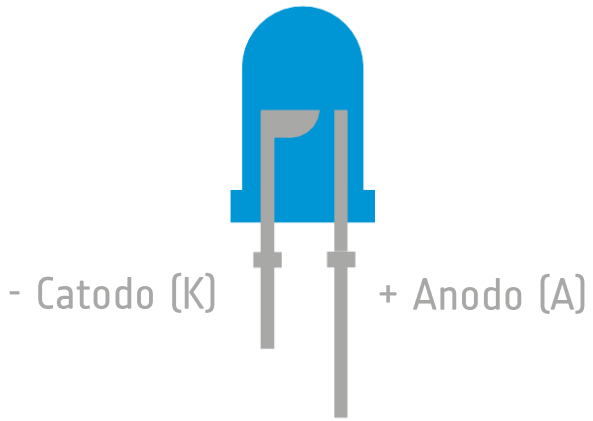
h. **Scheda Arduino**: Funziona come un semplice computer che contiene un processore, della memoria ed alcuni pin di input/output.

Figura A1



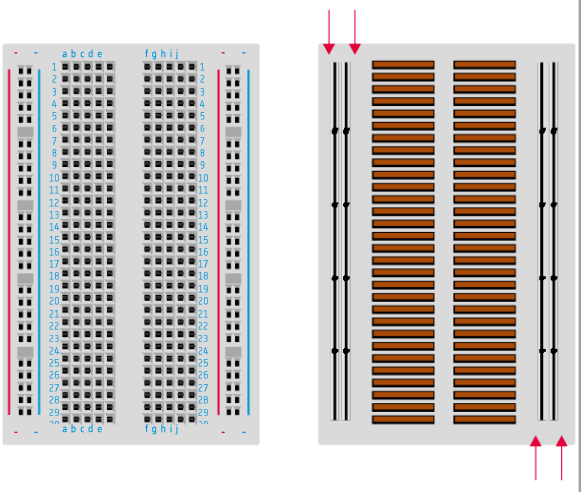
↑ Componenti kit base di Arduino

Figura A2



↑ LED: notare che il led ha una gamba lunga ed una corta. La gamba lunga è chiamato anodo ed la gamba corta catodo.

Figura A3



↑ Breadboard: notare che tutti i pin di ogni riga sono connessi

Quando connettete Arduino, il vostro computer dovrebbe riconoscerlo e inizializzare il processo di installazione del driver. Se avete problemi ad installare i driver potete trovare ulteriori istruzioni qui: www.arduino.cc/en/Guide/ArduinoUno

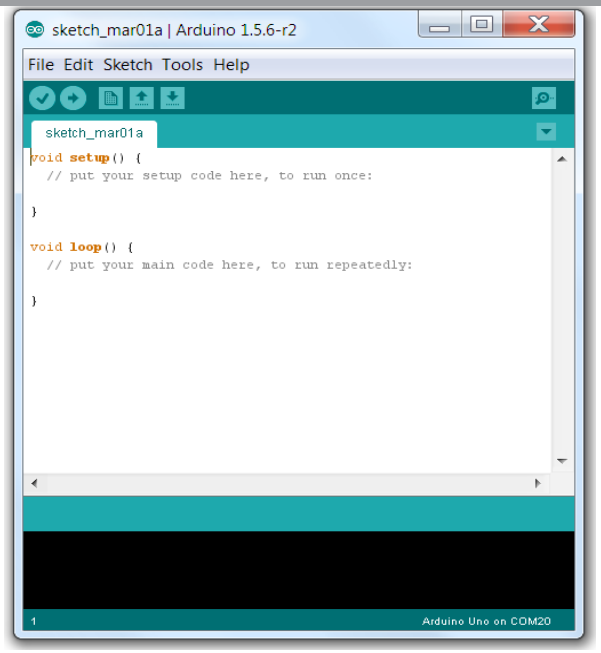
*Assicurarsi di selezionare la scheda giusta (provare Genuino Uno) e la corretta porta seriale (COM) nel menù Tools!

Software

Il software di Arduino può essere scaricato facilmente ed installato da qui: www.arduino.cc/en/Main/Software

I programmi scritti usando la IDE di Arduino sono chiamati sketch. Questi Sketch sono scritti nell'editor di testi e sono salvati con l'estensione .ino. L'editor ha possibilità di fare copia/incolla, e di fare ricerca/sostituzione del testo. L'area messaggi nella parte bassa (Figura A4) dà feedback mentre si salva e si esporta, serve anche a visualizzare gli errori. La scheda selezionata e la porta seriale sono visualizzate nell'angolo in basso a destra della finestra. I bottoni della toolbar permettono di verificare e caricare i programmi, creare, aprire e salvare gli sketches ed aprire il monitor seriale (per veder i dati raccolti dai sensori). La figura A4 è uno screenshot della IDE di Arduino.

Figura A4



↑ Interfaccia IDE di Arduino

Comandi Arduino - Tabella A1

	**Verify:** Controlla se ci sono errori nel codice durante la compilazione
	**Upload**: Compila il codice e lo carica sulla scheda configurata
	**New:** Crea un nuovo sketch
	**Open:** Presenta un menù di tutti gli sketch presenti nello sketchbook.
	**Save:** Salva lo sketch
	**Serial Monitor:** Apre il monitor seriale, permettendo di vedere i dati raccolti dai sensori

Ulteriori comandi si trovano nei cinque menù; File, Edit, Sketch, Tools e Help.

→ Attività 1: Fammi lampeggiare!

In questa attività gli studenti controlleranno un LED con Arduino. Lo scopo è controllare il lampeggiamento di un LED e capire come il codice è scritto/usato per controllare ogni LED. Più informazioni possono essere trovate attraverso una rapida ricerca su Google: Arduino LED blink

Materiale occorrente

- 1 Arduino Uno
- 1 breadboard
- 1 LED (verde)
- 1 resistenza da 220 Ohm (rosso/rosso/nero/nero/marrone)
- 2 cavetti

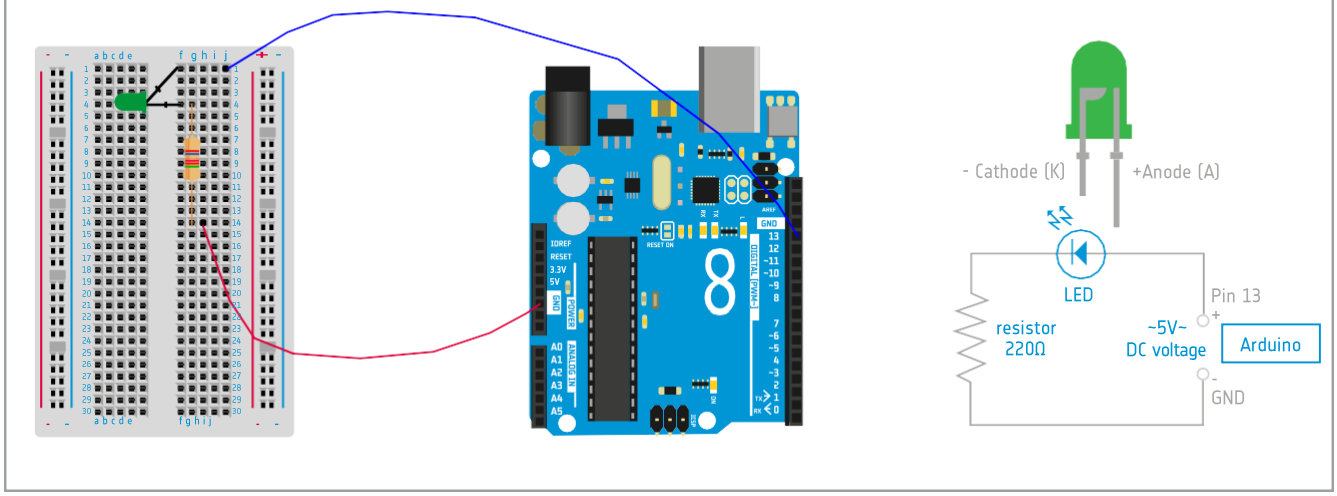
Esercizio

Setup

Il circuito è preparato come segue:

a. Connettere il LED verde nella breadboard come si vede in figura A5
b. Connettere la resistenza da 220 Ohm alla gamba corta del LED come mostrato in figura 5
c. Usando un cavetto, connettere la gamba lunga del LED al pin numero 13 di Arduino
d. Usando un cavetto, connettere la resistenza al pin GND di Arduino
e. Alla fine connettere Arduino al computer usando il cavo USB

Figure A5



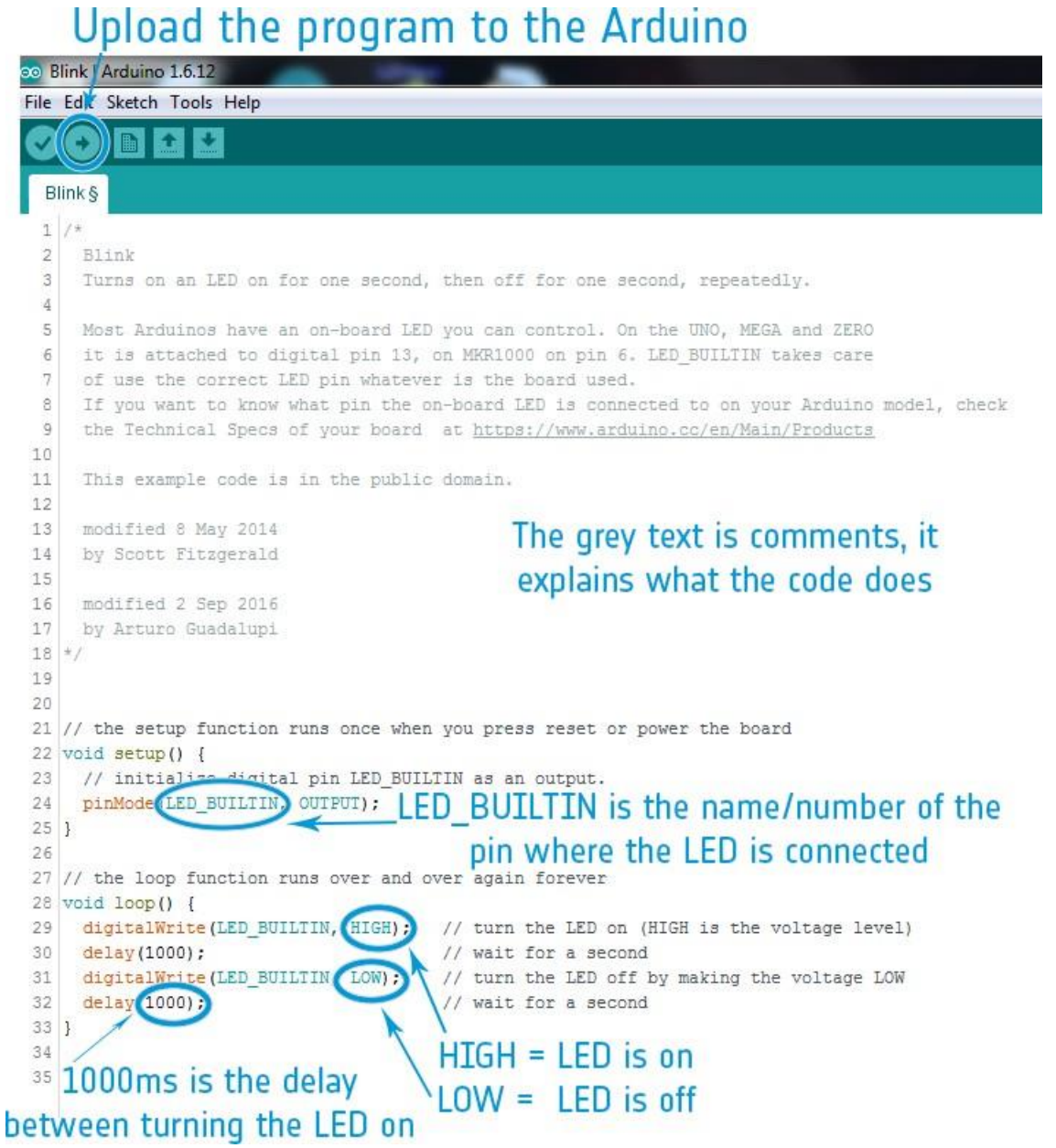
↑ Circuito Arduino pronto a far lampeggiare il LED verde

Codice

Gli studenti ora dovranno caricare un programma che controlli il LED. Useranno un codice di esempio che può essere trovato aprendo la IDE di Arduino e cliccando su:

File → Examples → Basics → Blink

Il codice necessario che apparirà sullo schermo viene mostrato nella seguente immagine. I commenti sono visualizzati in grigio dopo le doppie barre e spiegano ciascuna riga di codice.



Gli elementi principali del codice sopraindicato sono spiegati qui sotto:

```
void setup()
```

Questo comando è sempre scritto all'inizio, quando uno sketch ha inizio. Serve ad inizializzare le variabili che saranno usate nel resto dello sketch. Sarà eseguito una singola volta, dopo ogni accensione o reset dell'Arduino.

```
pinMode(LED_BUILTIN,OUTPUT)-> pinMode(13,OUTPUT)
```

Configura il pin specificato a comportarsi o come un input oppure come un output.
Nell'esempio in questione, sostituisci **LED_BUILTIN** con il numero del pin a cui è connesso il LED, ad esempio 13.

```
void loop()
```

Questo comando va precisamente quello che il suo nome suggerisce: ripete all'infinito tutti i comandi che contiene, permettendo al tuo programma di fare molteplici operazioni in successione.

```
digitalWrite(13,HIGH)
```

Accende il LED. HIGH imposta il voltaggio del pin 13 su 5V.

```
delay(1000)
```

Mette in pausa il programma per l'ammontare di tempo indicato (in millisecondi).

```
digitalWrite(13,LOW)
```

Spegne il LED abbassando il voltaggio. LOW significa avere 0V sul pin 13

Una lista completa di tutti i comandi può essere trovata tramite una rapida ricerca su Google: Arduino commands.

→ Attività 2: Mandare un S.O.S.!

In questa attività gli studenti usano il circuito costruito nell'attività 1. Controlleranno il LED verde con lo scopo di mandare in messaggio in codice Morse; questo per dimostrare come potrebbero comunicare con un rover su Marte. Per inviare una 'O', gli studenti dovrebbero cambiare il ritardo nel codice per aumentare il tempo in cui il LED resta acceso. In questo caso abbiamo cambiato da 1000 ms a 5000 ms, ma potrebbe essere un qualsiasi numero maggiore di 1000 ms.

Il codice è mostrato in figura A6:

Figure A6

```
void loop() {
  digitalWrite(13, HIGH);
  delay(5000);
  digitalWrite(13, LOW);
  delay(1000);

  digitalWrite(13, HIGH);
  delay(5000);
  digitalWrite(13, LOW);
  delay(1000);

  digitalWrite(13, HIGH);
  delay(5000);
  digitalWrite(13, LOW);
  delay(1000);

}
```

↑ Codice per inviare una O in Morse

→ Attività 3: Misurare la temperatura

In questa attività gli studenti misurano la temperatura usando Arduino. L'intento è quello di leggere la temperatura da un sensore di temperatura (termistore*) e di capire come il codice sia scritto/usato per controllare il sensore. Più informazioni possono essere trovate attraverso una rapida ricerca su Google: Arduino thermistor sensore.

Materiale occorrente

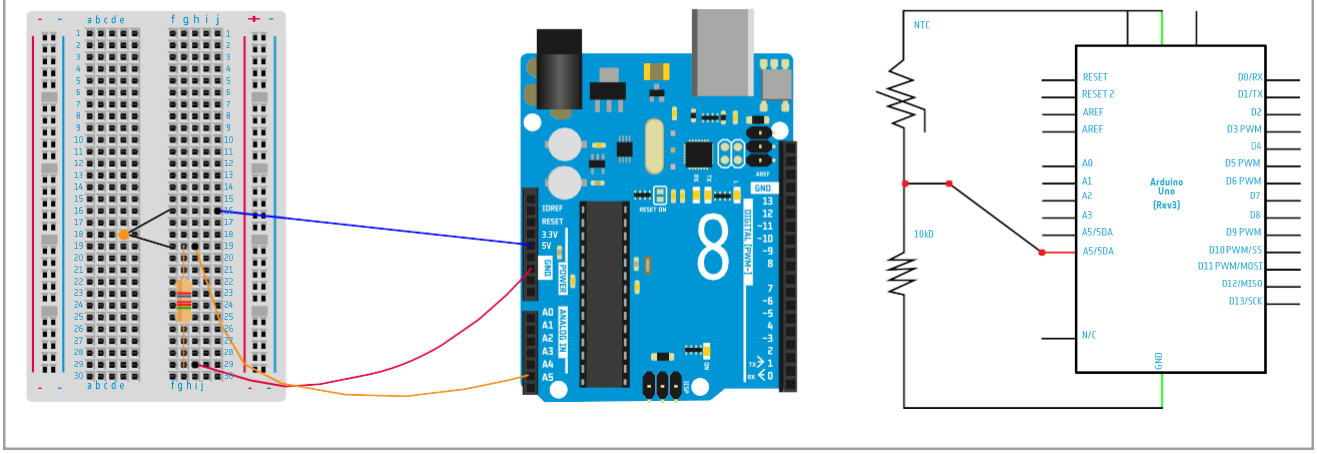
- 1 Arduino Uno
- 1 breadboard
- 1 sensore di temperature (termistore)
- 1 resistoreda 10 kOhm (marrone/nero/nero/marrone)
- 3 cavetti

Esercizio

Setup

a. Connettere il termistore alla breadboard come mostrato in figura A7
b. Connettere la resistenza 10 kOhm al termistore come mostrato in figura A7
c. Usando un cavetto, connettere il termistore al pin 5V di Arduino
d. Usando un cavetto, connettere la resistenza al pin GND di Arduino
e. Usando un cavetto, connettere il termistore e la resistenza al pin A5 di Arduino
f. Alla fine connettere Arduino al computer usando il cavo USB

Figura A7



↑ Circuito Arduino pronto per misurare la temperatura

* termistore: un resistore elettrico la cui resistenza è ridotta in modo significativo dal calore

Codice

Gli studenti scriveranno ora il loro codice. Una lista completa di tutte i comandi possibili può essere trovata con una rapida ricerca su Google: Arduino commands. La figura A8 illustra un esempio di codice che misura la temperatura. In questo esempio, la temperatura che viene letta dal sensore nel circuito è chiamata **AnalogT**, la sua unità di misura è espressa in volts poiché si tratta di un segnale elettrico; questo non ha però alcun significato fisico di temperatura. La temperatura che viene convertita in gradi Celsius (l'unità di misura della temperatura usata in Europa) è chiamata **AnalogTg**. Il nome del pin su Arduino è A5. Una volta che il codice è caricato e comincia a girare, aprire la finestra del Serial Monitor* per vedere i dati che arrivano.

Figura A8

```
1  void setup() {
2    // put your setup code here, to run once:
3  Serial.begin(9600);
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8  float AnalogT;
9  float AnalogTf;
10 AnalogT= float(analogRead(A5));
11 AnalogTf = (-40000/AnalogT) + 100;
12 Serial.println();
13 Serial.print("Temperature: ");
14 Serial.print(AnalogTf);
15 Serial.print("C");
16 delay(1000);
17
18 }
```

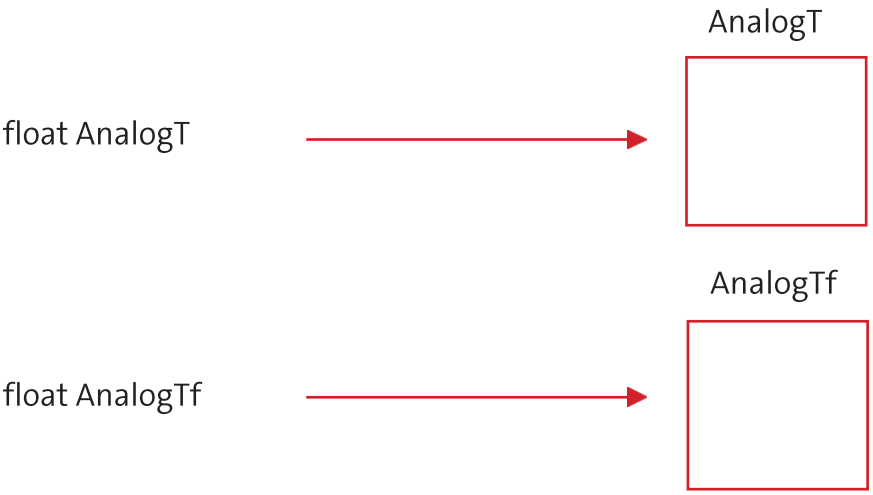
↑ Il codice di Arduino per misurare la temperatura

*guardare Tabella A1: comandi Arduino.

Per la spiegazione di void setup e void loop si veda l'attività 1 esercizio 3.

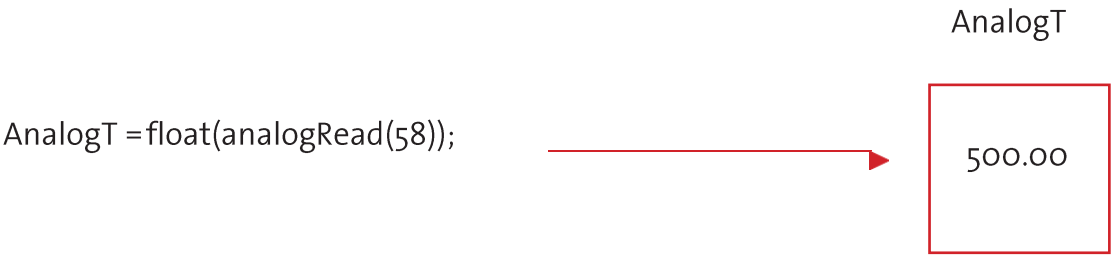
Riga 3: Siccome la temperatura sarà stampata sullo schermo del computer, è necessario predefinire il rate dei dati (9600 bits/secondo) per la trasmissione degli stessi tra Arduino e il computer.

Righe 8-9: Queste due righe di codice sono usate per definire le variabili chiamate ANalogT ed AnalogTf. Questi nomi sono stati scelti dal programmatore. Per modellare questo processo useremo una spiegazione grafica. Quando si scrive float AnaloT, dentro la memoria del computer si crea una scatola o uno spazio vuoto chiamato AnalogT che ha una grandezza specifica indicata dalla parola 'float'. Float significa che il numero dentro la scatola avrà una parte decimale.



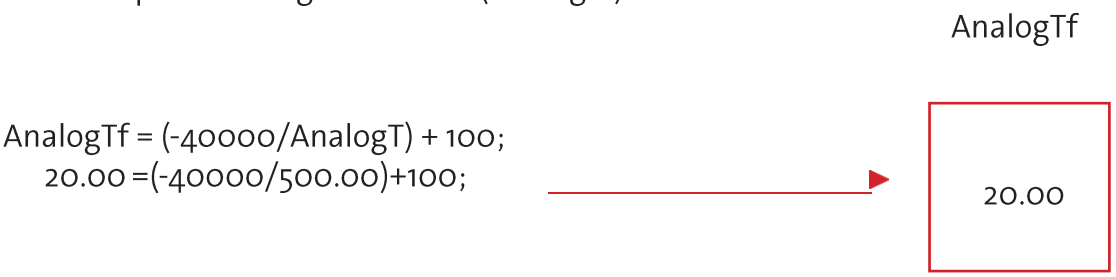
Riga 10: AnalogT = float(analogRead(A5));

Il computer scriverà il numero letto dal pin A5 su Arduino (es. 500.00) dentro la scatola **AnalogT**. Si usa la funzione analogRead() per leggere da ogni pin di Arduino.



Riga 11: AnalogTf = (-40000/AnalogT) + 100;

In questa riga c'è l'operazione matematica che calibra e converte la temperatura in volt (AnalogT) nella temperatura in gradi Celsius (AnalogTf).



Righe 12-15: Per mostrare la misure della temperatura sullo schermo, occorre usare la funzione print. Questa funzione recupera il numero dentro ad una scatola e lo stampa nel Serial monitor. In questo esempio la funzione stamperà il numero che si trova dentro la scatola chiamata AnalogTf. Tutte le parole dentro ".." sono stampate direttamente sul serial monitor.

Riga 16: La funzione di delay è usata per aspettare 1000 millisecondi prima di visualizzare la temperatura sul serial monitor.

Discussione

Gli studenti dovranno probabilmente calibrare il loro sensore di temperatura. Possono usare un altro termometro per misurare la temperatura reale come riferimento per calibrare il loro sensore. Per calibrarlo in maniera approssimata, potrebbero far girare il codice così come scritto precedentemente, poi se la temperatura misurata dal sensore non fosse la stessa temperatura di riferimento misurata dal termometro possono provare a cambiare l'ultimo numero (+100) della riga 11 (AnalogTf) del codice in figura A8.

Pratica

Si chiede agli studenti di calibrare il sensore di temperatura come se fossero sulla superficie di Marte. Dal momento che la temperatura media sulla superficie Marziana è di circa -60°C mentre quella terrestre è di circa +14°C, si chiede agli studenti di calcolare uno spostamento di -74 gradi. Il codice potrebbe essere adattato in questo modo:

AnalogTf = (-40000/AnalogT) + 100 – 74;

→ Attività 4: Misurare la pressione

In questa attività, gli studenti misurano la pressione ambientale con Arduino. Lo scopo è leggere la pressione dal sensore (MPX4115A) e capire come il codice è scritto/usato per controllare il sensore. Maggiori informazioni possono essere trovate attraverso una rapida ricerca su Google: Arduino MPX4115A.n this activity, students measure the ambient pressure with the Arduino.

Materiale occorrente

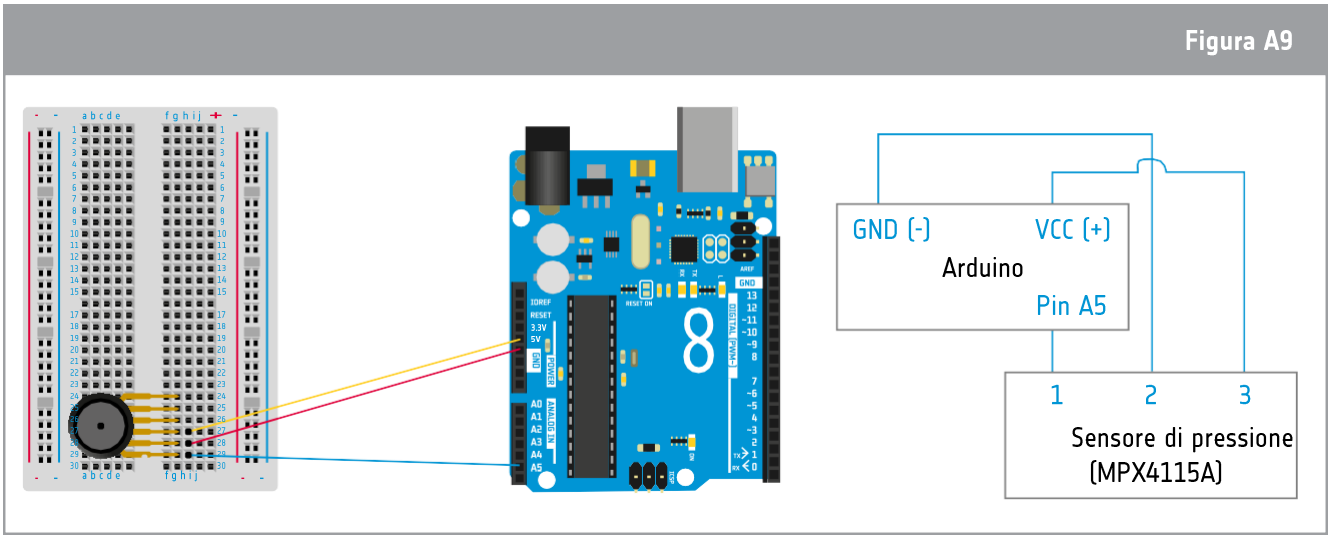
- 1 Arduino Uno
- 1 breadboard
- 1 sensore di pressione
- 3 cavetti

Esercizio

Setup

a. Connettere il sensore di pressione alla breadboard
b. Usando un cavetto, connettere la gamba 1 del sensore di pressione (contrassegnato) al pin A5
c. Usando un cavetto, connettere la gamba 2 del sensore di pressione al pin GND di Arduino
d. Usando un cavetto, connettere la gamba 3 del sensore di pressione al pin 5V di Arduino
e. Alla fine connettere Arduino al computer usando il cavo USB.

Figura A9



↑ Il circuito Arduino pronto per misurare la pressione

Codice

In questo caso, gli studenti usano gli stessi comandi visti nell'attività precedente sulla temperatura; di nuovo si concentreranno sulla calibrazione del sensore. Una cosa importante ed utilissima per l'utilizzo del sensore è guardare il datasheet; questo infatti contiene informazioni dettagliate riguardo il funzionamento del sensore. In questo caso, possiamo facilmente trovare le informazioni utili con una rapida ricerca su Google, cercando MPX4115A datasheet.

Il sensore di pressione converta la pressione misurata in un voltaggio. Questo è un segnale analogico che sarà tradotto in un segnale digitale. Per tradurre il voltaggio misurato in valori di pressione ambientale (la cui corretta unità di misura del S.I. è il Pascal), occorre la funzione di trasferimento del sensore. La funzione descrive la relazione matematica tra il voltaggio in output del sensore e la pressione equivalente. Questa funzione può essere trovate nel datasheet del sensore. Detta funzione può cambiare tra sensori differenti.

Se diamo un'occhiata al datasheet, troveremo la seguente funzione di trasferimento: $\mathbf{V_{out}}$ = **Vs(Px0.009-0.095)**. Dobbiamo quindi isolare il valore di P (pressione) e considerare che Vs = 1024 (il voltaggio letto è suddiviso in 1024 scalini). $\mathbf{V_{out}}$ = valore analogico che sarà letto dal pin **A5**.

Di seguito vediamo un esempio del codice che stampa i valori della pressione. Usatelo come riferimento.

Figura A10

```
1  void setup() {
2    // put your setup code here, to run once:
3     Serial.begin(9600);
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8     float AnalogP;
9     float AnalogPf;
10    AnalogP= float(analogRead(A5));
11    AnalogPf = ((AnalogP/1024)+0.095)/0.009;
12    Serial.println();
13    Serial.print("Pressure: ");
14    Serial.print(AnalogPf);
15    Serial.print("kPa");
16    delay(1000);
17
18 }
```

↑ Codice Arduino per misurare la pressione

Il nome dato alla pressione in volt è ANalogP e per la pressione in pascal è AnalogPf. Il nome del pin dove è connesso il sensore di pressione è A5.
Per calibrare il sensore di pressione gli studenti possono controllare la pressione ambientale del loro luogo su internet. Per una grossolana calibrazione possono far girare il codice come scritto sopra, e se il loro risultato non è lo stesso della pressione di riferimento, possono aggiustare la riga 11 (**analogPf**) del codice.

→ Attività 5: Misurare l'altitudine

La pressione varia con la quota, essa può quindi essere usata per misurare la quota. Per fare questo gli studenti devono aggiungere questa parte di codice all'interno del loop del codice di misura di pressione dell'attività precedente.

Segue un esempio del codice che stampa la quota in metri. Usatelo come riferimento.

Figura A11

```
float Altitude;
float Altitudef;
  Altitude = pow(AnalogPf/101.325,0.1903);
  Altitudef = (1-Altitude)*44300 + 100;
Serial.print("Altitude: ");
Serial.print(Altitudef); Serial.print ("meters");
```

↑ Codice Arduino per misurare l'altitudine

Il nome dato alla quota in metri è Altitudef. La pressione in unità di Kpa è data da AnalogPf in questo esempio.

Dall'aiuto fornito nel box "sapevi che", gli studenti dovrebbero rendersi conto del fatto che al crescere della quota la temperatura dell'atmosfera cala. Questa relazione dovrebbe poter essere visibile nel loro grafico di quota e temperatura.

Come usare Arduino in classe

Arduino è uno strumento eccellente per introdurre gli studenti alle basi della programmazione e alla vasta gamma di applicazioni possibili. Può anche essere usato per insegnare ed imparare materie STEM. Alcuni esempi sono:

- **Introduzione ai circuiti base** - La prima attività può essere usata per introdurre gli studenti a semplici circuiti.
- **Introduzione alla analisi dei dati** - Gli studenti raccolgono dati dal mondo reale (temperatura e/o pressione) e li analizzano usando grafici. Gli studenti leggono le informazioni e giungono a conclusioni.
- **Introduzione ad elementi di meteorologia e a come vengono raccolti i dati** - Gli studenti possono sviluppare in autonomia stazioni meteorologiche e fare previsioni su giornate soleggiate o nuvolose.

→ TI PRESENTO ARDUINO!

Introduzione alla programmazione di Arduino in C++

Sapevi che...?

La missione del 2020 del programma ExoMars ha portato un rover europeo con un contenuto scientifico russo sulla superficie di Marte per verificare se mai sia esistita la vita su Marte!

→ Attività 0: Per cominciare

Tutti noi, tutti i giorni, usiamo la tecnologia! Nelle missioni spaziali, gli ingegneri adoperano la tecnologia per comandare robot e comunicare con loro al fine di espandere la nostra conoscenza scientifica. In questa attività diventerai un ingegnere che studia Marte, il Pianeta Rosso, usando un attrezzo tecnologico chiamato Arduino. Vediamo come identificare i componenti di Arduino!!

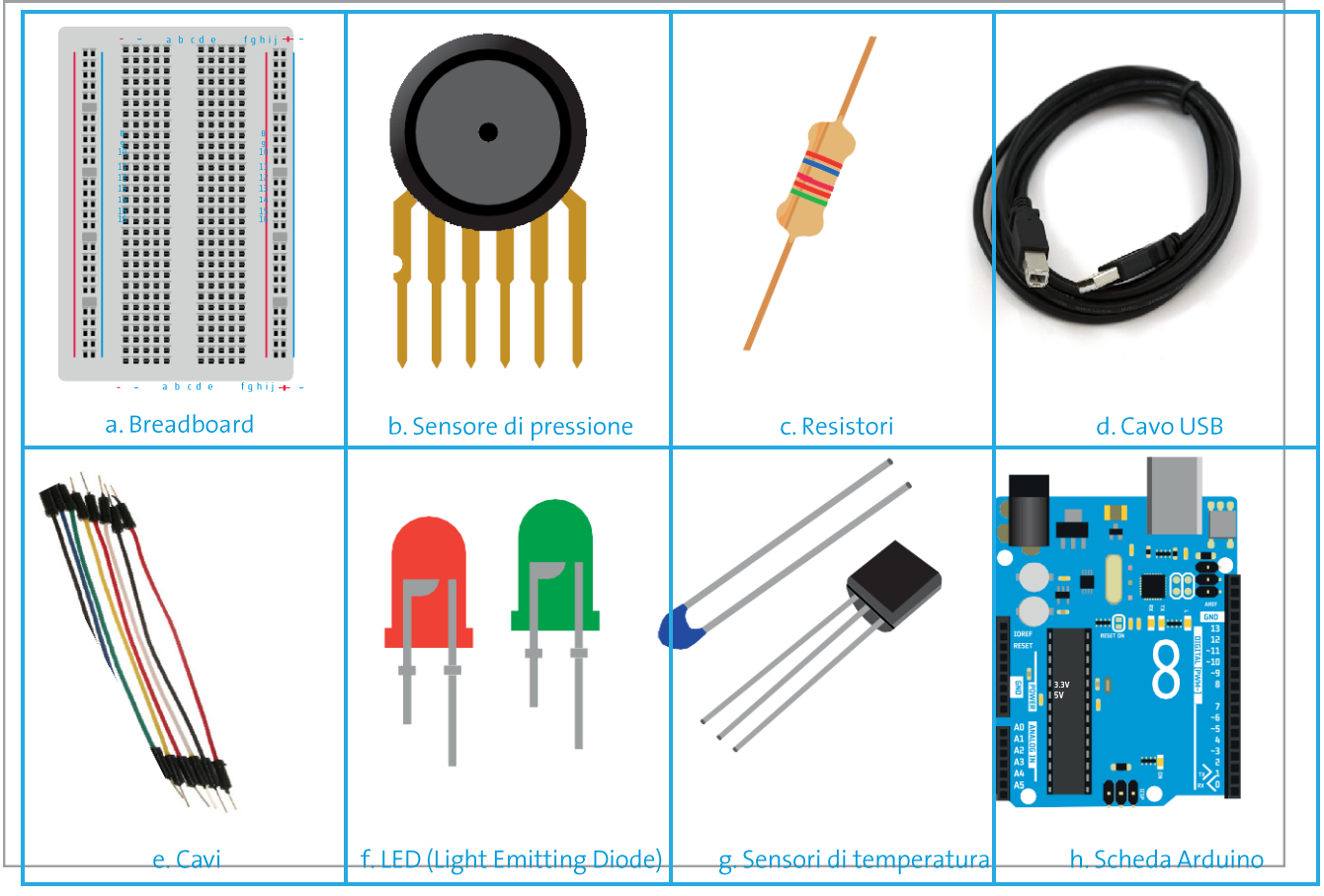
Materiale occorrente

- Kit base di Arduino

Componenti a terra:

1. Lavora in coppia. Apri il tuo kit Arduino e confronta i componenti con le descrizioni nella prossima pagina.

Figura A1



a. Breadboard
b. Sensore di pressione
c. Resistori
d. Cavo USB
e. Cavi
f. LED (Light Emitting Diode)
g. Sensori di temperatura
h. Scheda Arduino

↑ Componenti kit base Arduino

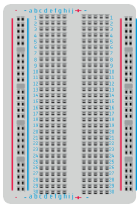
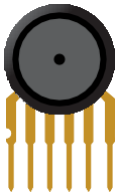
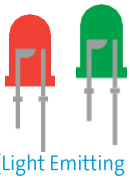
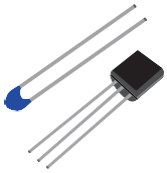
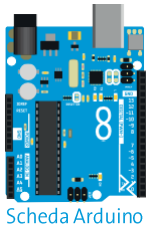
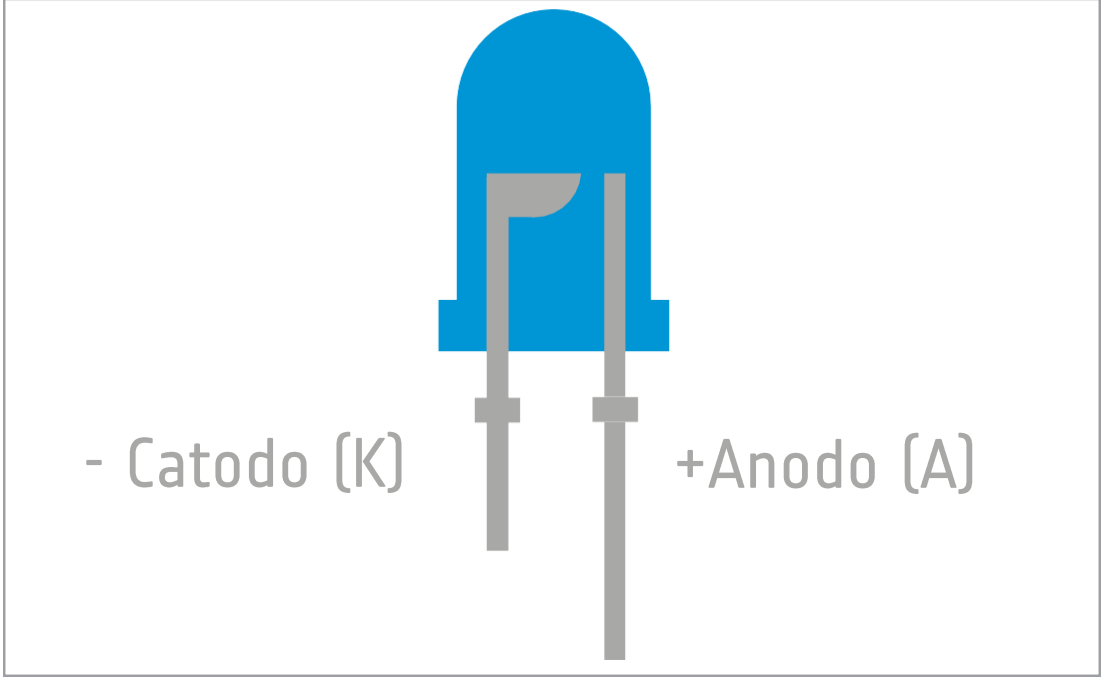
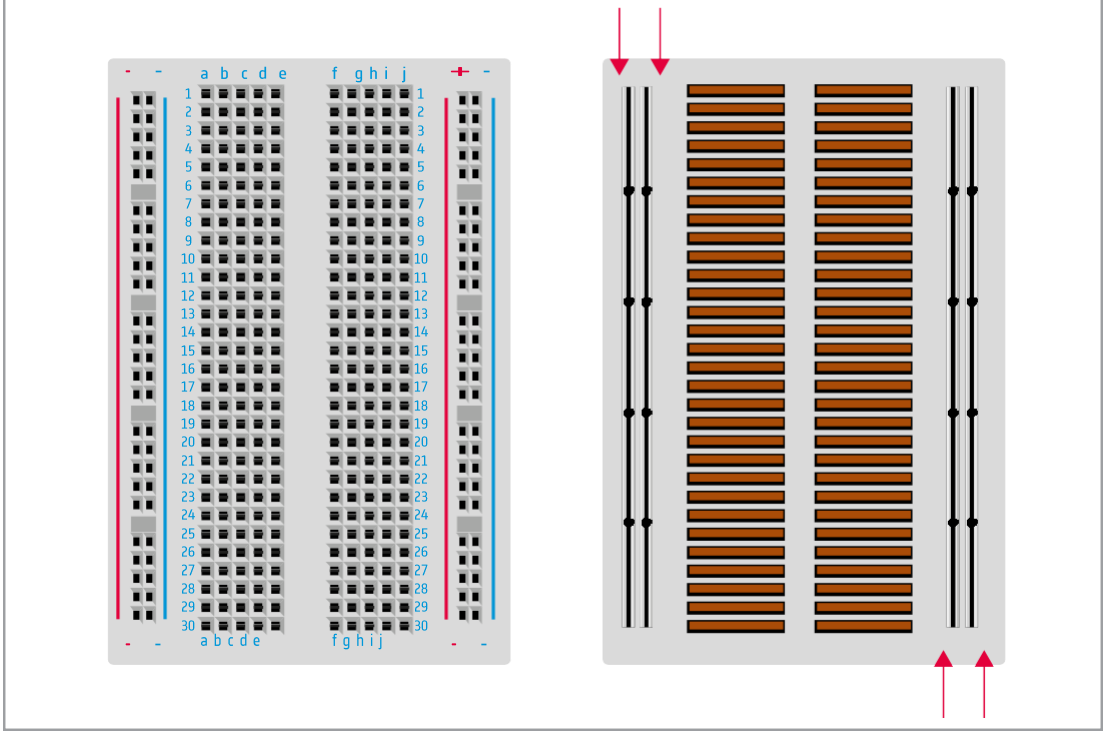
COMPONENTI	DESCRIZIONE
 Breadboard	Ha forma circolare e misura la pressione ambiente
 Sensore di pressione	Ha pin di input ed output e funziona come un semplice computer
 Resistori	È un supporto bianco usato per connettere facilmente componenti di elettronica
 Cavo USB	Ha due lunghe gambe collegate ad una testa blu e misura la temperatura
 Cavi	Di forma cilindrica con bande colorate, riduce la corrente che circola in un circuito
 LED (Light Emitting Diode)	Sono rossi e verdi ed emettono luce quando della corrente elettrica passa attraverso di loro
 Sensori di Temperatura	Hanno molti colori e sono usati per connettere i componenti e per condurre l'elettricità
 Scheda Arduino	Un lungo cavo nero usato per connettere Arduino al computer

Figura A2



↑ LED: notare che il led ha una gamba lunga ed una corta. La gamba lunga è chiamato anodo ed la gamba corta catodo.

Figura A3



↑ Breadboard: notare che tutti i pin di ogni riga sono connessi.

→ Attività 1: Fammi lampeggiare!

Per comandare oggetti interattivi hai bisogno di usare un attrezzo come un Arduino, fatto di diversi componenti elettronici. Dal momento che sei tu l'ingegnere con l'incarico di studiare il Pianeta Rosso, hai bisogno di comunicare con il rover, di dargli comandi e di ricevere dati. In questa attività imparerai il linguaggio di programmazione C++ che ti servirà per comunicare con il tu Arduino Uno, ed imparerai come si controlla una luce!

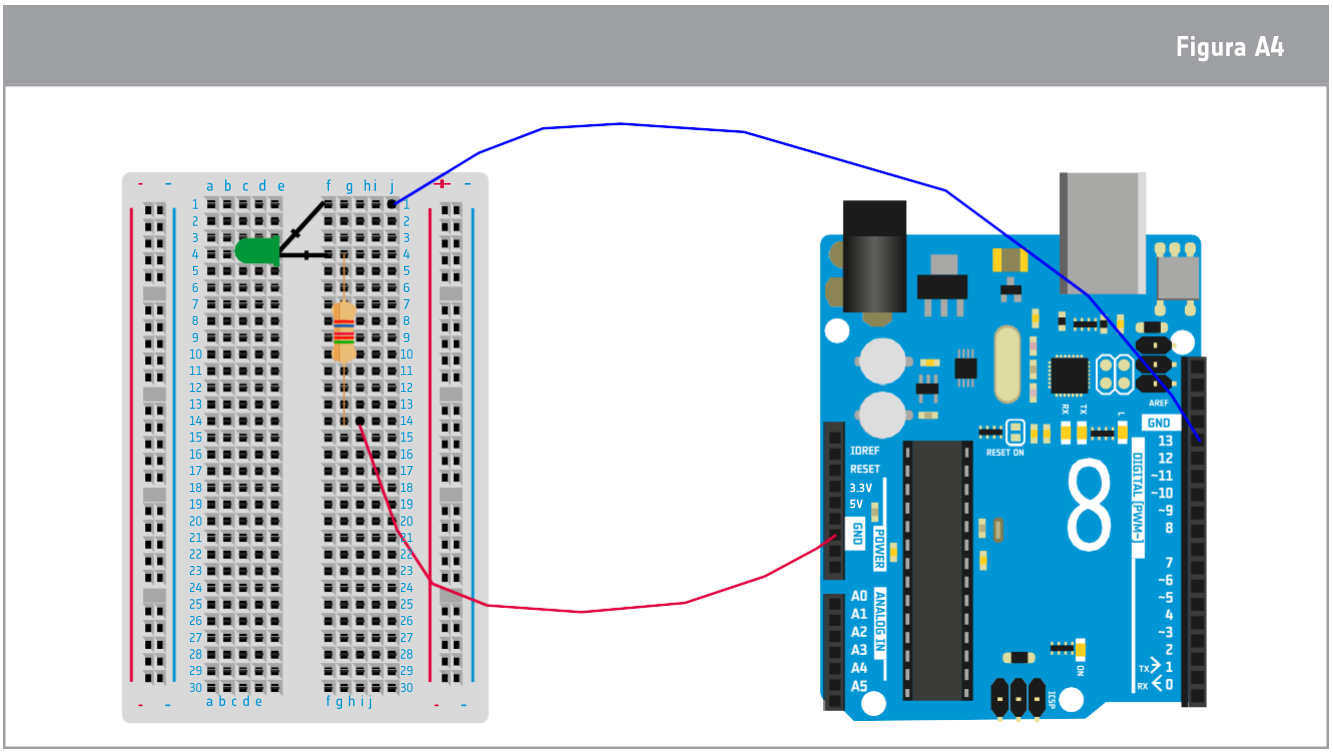
Materiale occorrente

- 1 Arduino Uno
- 1 breadboard
- 1 LED (verde)
- 1 resistenza da 220 Ohm (rosso/rosso/nero/nero/marrone)
- 2 cavetti

Esercizio

1. Segui attentamente queste istruzioni per preparare la breadboard e controllare il LED verde:
   a. Connettere il LED (plastica trasparente) nella breadboard come si vede in figura A4
   b. Connettere la resistenza da 220 Ohm alla gamba corta del LED come mostrato in figura A4
   c. Usando un cavetto, connettere la gamba lunga del LED al pin numero 13 di Arduino
   d. Usando un cavetto, connettere la resistenza al pin GND di Arduino
   e. Alla fine connettere Arduino al computer usando il cavo USB

Figura A4



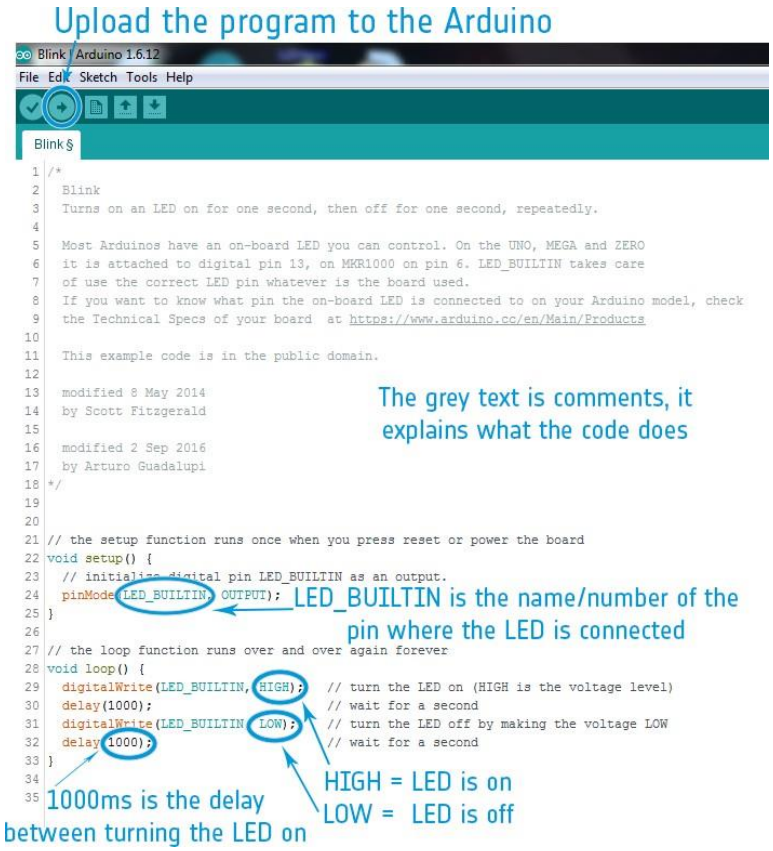
↑ Circuito Arduino pronto a far lampeggiare il LED verde

2. Nel seguente box disegna lo schema elettrico del circuito corrispondente alle connessione elettriche della figura A4: Usa questi simboli:

Resistore

3. Ora che la breadboard è pronta, hai bisogno di mandare istruzioni all'Arduino per far lampeggiare il LED. Apri l'IDE di Arduino sul computer e clicca su File → Examples → Basics → Blink.

Il codice Blink apparirà sullo schermo come mostrato nella seguente figura. Leggi attentamente la spiegazione per capire a fondo.

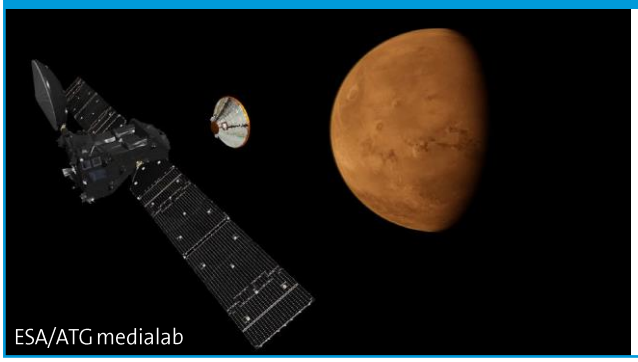


a. Prima di caricare il codice su Arduino, devi accertarti che il nome del pin del codice corrisponda con il numero del pin di Arduino dove è collegato il led verde. Spiega che cosa hai cambiato nel codice per controllare il led verde:

b. Carica il codice su Arduino cliccando nel pulsante in alto a sinistra nello schermo.

Congratulazioni, il led sta lampeggiando!

Sapevi che...?


ESA/ATG medialab

Marte è così lontano che i segnali impiegano abbastanza tempo per viaggiare da una navicella o da un rover fino alla Terra. Questo ritardo nelle comunicazioni rende la comunicazione con il rover o una rapida reazione nel caso succeda qualcosa di inaspettato, una vera sfida.

→ Attività 2: Mandare un S.O.S.!

Ora che sai come far lampeggiare un LED usando Arduino, questa attività ti insegnerà a comunicare usando un LED ed a mandare un messaggio al rover su Marte. Effettivamente una fortissima tempesta di sabbia è prevista a breve sul rover! Per evitare di danneggiare il rover fagli inviare un S.O.S.!

Sapevi che...?

NASA/JPL-Caltech/MSSS

Questa immagine di una tempesta di sabbia su Marte mostra le nuvole che si formano dalle variazioni di pressione, temperatura e quota dovute alla disposizione verticale: come quando il vento soffia sopra una montagna o le mura di un cratere.

Materiale occorrente

- Il circuito costruito nell'attività 1

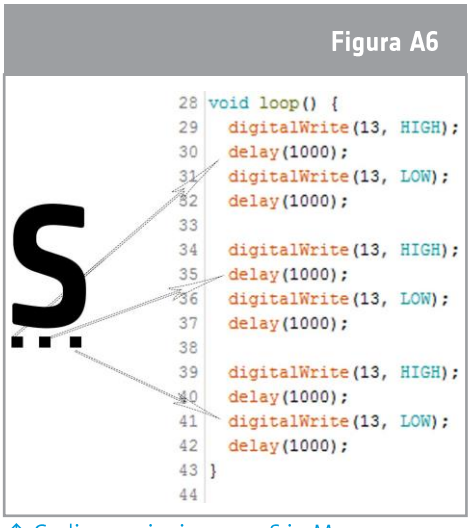
Esercizio

Usa il codice Morse per far lampeggiare il LED verde. Nella figura A5, un punto indica un segnale corto; questo terrà acceso il LED per un tempo breve. Una linea indica un segnale lungo; questo terrà acceso il LED per un tempo lungo. Il codice per mandare una S è dato come esempio nella figura A6.

Figura A5

↑ SOS in codice Morse

Figura A6



↑ Codice per inviare una S in Morse

Spiega cosa ti serve cambiare nel codice per inviare una O:

Adatta il codice per mandare un S.O.S., poi carica il programma su Arduino e provalo!

→ Attività 3: Misurare la temperatura

La tecnologia di Arduino ti permette di controllare un LED collegato ad Arduino usando il codice scritto su un portatile. Quando carichi il codice, questo manda le istruzioni ad Arduino ed attiva le componenti elettroniche. Nella missione ExoMars, la tecnologia sarà utilizzata per indagare l'ambiente di Marte grazie ad alcuni sensori. In questa attività scoprirai come misurare la temperatura della classe e simulare la temperatura su Marte!

Materiale occorrente

- 1 Arduino Uno
- 1 breadboard
- 1 sensore di temperatura (termistore)
- 1 resistore 10kΩ (marrone/nero/nero/marrone)
- 3 wires

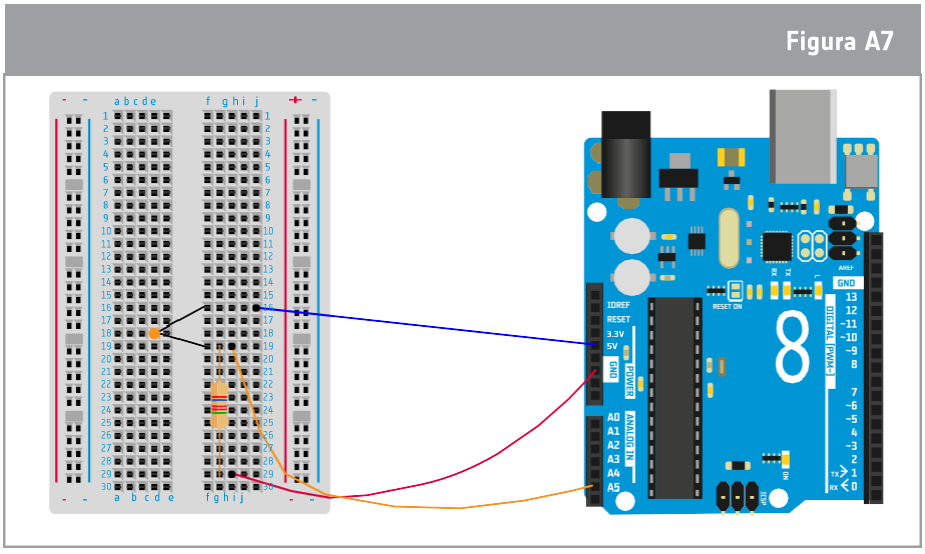
Esercizio

1. Per misurare la temperatura della classe occorre costruire il circuito sulla breadboard come segue.

   Segui attentamente queste istruzioni:

   a. Connettere il termistore alla breadboard come mostrato in figura A7
   b. Connettere la resistenza 10 kOhm al termistore come mostrato in figura A7
   c. Usando un cavetto, connettere il termistore al pin 5V di Arduino
   d. Usando un cavetto, connettere la resistenza al pin GND di Arduino
   e. Usando un cavetto, connettere il termistore e la resistenza al pin A5 di Arduino
   f. Alla fine connettere Arduino al computer usando il cavo USB

Figura A7



↑ Circuito Arduino pronto per misurare la temperatura

2. La figura A8 illustra il codice incompleto per misurare la temperatura. Tre valori, mostrati dai box (rosso, verde e blu) devono essere riempiti per dare le istruzioni corrette ad Arduino per misurare la temperatura. Vediamo come riempire questo codice.

Figura A8

```
1
2  void setup() {
3    Serial.begin(9600);
4  }
5
6  void loop() {
7
8    float [                    ];
9    float [                    ];
10
11       [                    ] = float(analogRead([   ]));
12       [                    ] = (-40000/[                    ])+100;
13
14   Serial.println();
15   Serial.print("Temperature in classroom: ");
16   Serial.print([                    ]);
17   Serial.print("C");
18   delay(1000);
19
20 }
```

↑ Codice Arduino per misurare la temperatura

a. Quale unità è usata per determinare la temperature in Europa?

Il sensore di temperatura collegato ad Arduino legge in primo luogo la temperatura usando un altra unità di misura: il volt. Per trasformare questo numero da volt a gradi Celsius devi usare una formula scritta nella riga 12 di figura A8. Alla fine stamperai la temperatura su schermo usando la funzione Serial.print nel codice.

b. Definisci la variabile che contiene la temperatura volts: [ ]

Definisci la variabile che contiene la temperatura in gradi Celsius: [ ]

Cerca il pin di Arduino in cui è connesso il sensore di temperatura: [ ]

c. Sei pronto a scrivere il codice sul tuo portatile. Apri l'IDE di Arduino e scrivi il codice in figura A8, inserisci al posto dei box le variabili che hai definito per i box colorati. Stai attento a sostituire ogni box colorato con la sua rispettiva variabile!!

Carica il codices u Arduino Uno cliccando sul pulsante 

d. Per vedere la misura di temperatura, occorre che apri una finestra specifica sulla IDE di Arduino chiamata Serial monitor. Cerca per questo simbolo  sulla barra degli strumenti per avere accesso.

Discussione

1. Parla dell'importanza delle misure di temperatura coi tuoi compagni. Come puoi verificare che il sensore misura accuratamente la temperatura della tua classe?

2. Quale riga di codice della figura A8 puoi modificare per calibrare il sensore?

3. Per esercitarti un po' di più con Arduino, calibra il sensore di temperature come se si trovasse sulla superficie di Marte. Considera che la temperatura media sulla superficie marziana è di circa -60° Celsius.

   a. Temperatura media sulla Terra= _____

   b. Scrivi la nuova formula di calibrazione del sensore di temperatura qui sotto:

→ Attività 4: Misurare la pressione

In questa attività imparerai come misurare la pressione atmosferica usando Arduino e un sensore di pressione.

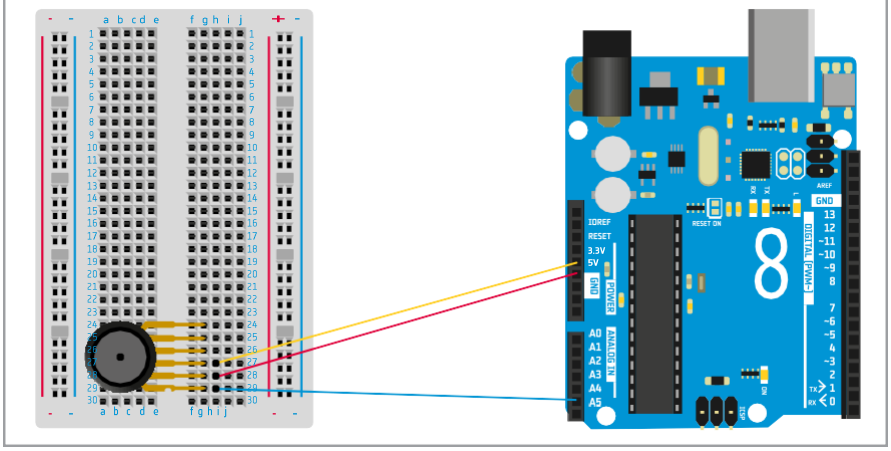
Materiale occorrente

- 1 Arduino Uno
- 1 breadboard
- 1 sensore di pressione
- 3 cavetti

Esercizio

1. Per misurare la pressione nella classe devi prima costruire il circuito sulla breadboard (Figura A9). Segui attentamente queste istruzioni (Nota: ogni pin nel sensore di pressione si comporta in modo diverso, quindi assicurati che il sensore sia orientato come in figura):

Figura A9



↑ Circuito Arduino pronto per misurare la pressione

2. La a figura A10 mostra un codice incompleto per misurare la pressione. Tre variabili mostrate come box (rosso, verde e blu), devono essere riempite per dare le istruzioni corrette ad Arduino per riuscire a misurare la pressione. Trova un modo per completare il codice.

Figura A10

```
1  void setup() {
2
3    Serial.begin(9600);
4  }
5  void loop() {
6
7    float [          ] ;
8    float [          ] ;
9
10     [          ] = float(analogRead([ ]));
11     [          ] =(([          ] /1024)+0.095)/0.009;
12   Serial.println();
13   Serial.print("Pressure in classroom: ");
14   Serial.print([          ]);
15   Serial.print("kPa");
16   delay(1000);
17
18 }
```

↑ Codice Arduino per misurare la pressione

a. Quale unità di misura viene usata per misurare la pressione nel S.I.?

---

b. Definisci la variabile che contiene la pressione in volt:

Definisci la variabile che contiene la pressione in pascal:

Individua il pin di Arduino in cui è connesso il sensore di pressione:

Il sensore di pressione connesso ad Arduino leggerà per prima cosa la pressione in un'altra unità di misura: il Volt, che è l'unità di misura del potenziale elettrico. Per trasformare questo numero da volt a pascal si utilizza questa formula dal datasheet del sensore: Vout = Vs(Px0.009-0.095)

Infine mostra la misura di pressione sullo schermo usando la funzione Serial.print nel codice

c. Usando la formula Vout = Vs(Px0.009-0.095), calcola la formula di calibrazione per completare la riga 12 della figura A10. Siccome il voltaggio letto è diviso in 1024 parti, considera Vs = 1024.

---

d. Sei pronto a scrivere il codice sul tuo portatile. Apri l'IDE di Arduino e scrivi il codice in figura A8, inserisci al posto dei box le variabili che hai definito per i box colorati. Stai attento a sostituire ogni box colorato con la sua rispettiva variabile!

Carica il codice di Arduino cliccando sul pulsante 

e. Per vedere la misura di temperatura, occorre che apri una finestra specifica sulla IDE di Arduino chiamata Serial monitor. Cerca per questo simbolo  sulla barra degli strumenti per avere accesso.

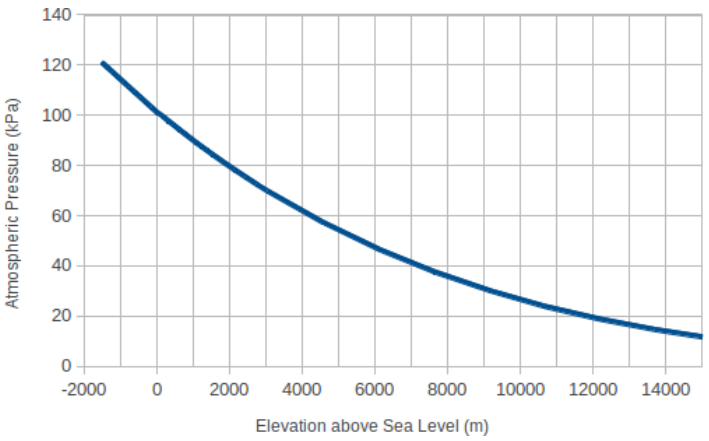
Discussione

Parla dell'importanza delle misure di temperatura coi tuoi compagni. Come puoi verificare che il sensore misuri accuratamente la temperatura della tua classe?

Quale riga di codice della figura A10 puoi modificare per calibrare il sensore?

→ Attività 5: Misurare l'altitudine

La pressione dell'aria carica con la quota. Per esempio la pressione a livello del mare è maggiore che la pressione in montagna. Puoi vedere come varia nel seguente grafico. Con un po' di matematica ed una basilare comprensione della fisica dei gas, possiamo calcolare esattamente la relazione tra pressione e quota. In questo momento non occorre preoccuparsi di ciò. In questa attività userai le tue misurazioni di pressione per determinare la tua quota usando una versione semplificata della relazione pressione-quota conosciuta come la formula barometrica.



Materiale occorrente

- Circuito Arduino costruito nell'attività 4
- Codice per misurare la pressione dell'attività 4

Esercizio

1. Apri il codice per la pressione che hai scritto nell'attività 4 nella IDE di Arduino. Aggiungi questo codice incompleto dentro il main loop:

Figura A11

```
1  float                    ;
2  float                    ;
3
4                  = pow(                  /101.325,0.1903);
5                  =(1-                    )*44300+100;
6
7  Serial.println();
8  Serial.print("Altitude in classroom: ");
9  Serial.print(                );
10 Serial.print("meters");
11 delay(1000);
```

↑ Codice Arduino per misurare l'altitudine

2. Per determinare la quota, dovrai usare le misure del sensore di pressione una formula scritta nella riga 4 della figura A11. Come hai scoperto negli esercizi precedenti, un sensore connesso ad Arduino leggerà dapprima le misure in volts. Per trasformare questo numero da volt a metri avrai bisogno della formula scritta nella riga 5 della figura A11. Alla fine i dati relative alla quota possono essere mostrati usando la funzione Serial.print nel codice. Se la lettura non è calibrata, può essere sistemata modificando la riga 5 del codice.

3. a. Definisci la variabile contenente la quota in volt:

Definisci la variabile contenente la quota in metri:

Cerca il nome che hai dato al box verde che contiene la pressione in pascal (guarda l'attività 4):

b. Completa il codice con le variabili che hai definito nelle box gialla, viola e verde. Stai attento a sostituire ogni box colorato con la sua rispettiva variabile! Poi carica il codice.

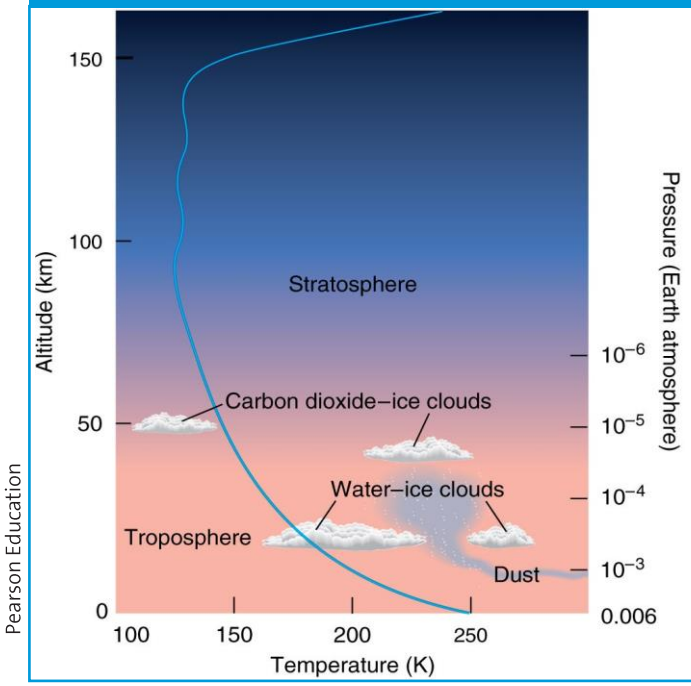
c. Per vedere la misura di quota, occorre che apri una finestra specifica sulla IDE di Arduino chiamata Serial monitor.

4. Combina entrambi i codici di temperatura e quota in un singolo sketch e caricalo su Arduino.

Raccogli la temperatura e la quota dal pavimento al tetto della tua classe e disegna un grafico che mostri i risultati.

Puoi spiegare la forma del grafico?

Sapevi che...?



Pearson Education

L'atmosfera marziana è un "foglio" sottilissimo di gas, principalmente diossido di carbonio, che si estende sulla superficie di Marte fino al bordo dello spazio. Il sole riscalda la superficie di Marte e parte di questo calore riscalda il gas atmosferico vicino alla superficie. Il gas riscaldato quindi si diffonde o sale per convezione attraverso l'atmosfera. Per ciò la temperatura del gas è maggiore vicino alla superficie e cala man mano che si sale di quota.

→ LINK

Arduino blog, utile per trovare le ultime novità su Arduino:

https://blog.arduino.cc/

Guide per l'utilizzo delle varie componenti di Arduino
https://quarkstream.wordpress.com/

Molti progetti basati su Arduino che puoi provare a casa possono essere trovati su Instructables:
http://www.instructables.com/howto/arduino/

HackADay ospita molti altri progetti Arduino:
https://hackaday.io/list/3611-arduino-projects