





Arduino & Potenziometro

Componenti utilizzati.

Hardware: Arduino 1, potenziometro, 3 cavi di collegamento, cavo USB

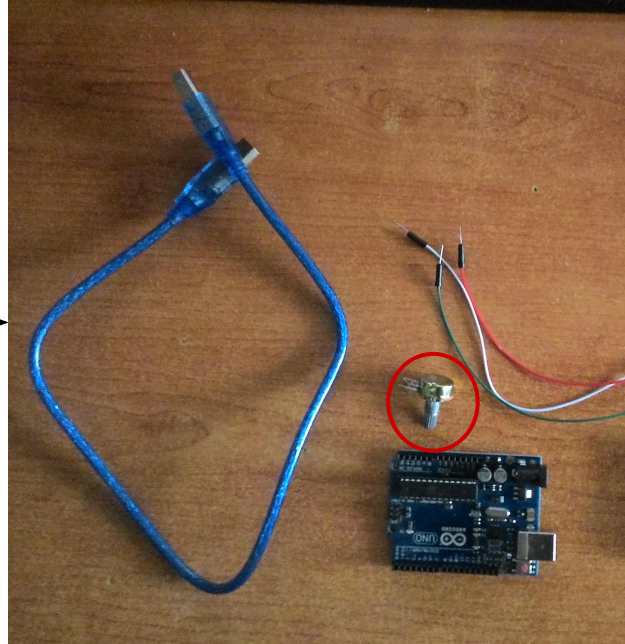
Software:  IDE Arduino: <https://www.arduino.cc/>
 CoolTerm: <http://freeware.the-meiers.org/>
 Processing: <https://processing.org/>
 MS Excel: <https://products.office.com/it-it/excel>

Hardware

Potenziometro: componente che permette di **variare la resistenza** durante il funzionamento in un circuito (cerchiato in rosso).

Arduino: piattaforma hardware low-cost programmabile, con cui è possibile creare circuiti "quasi" di ogni tipo per molte applicazioni, soprattutto in ambito di robotica ed automazione.

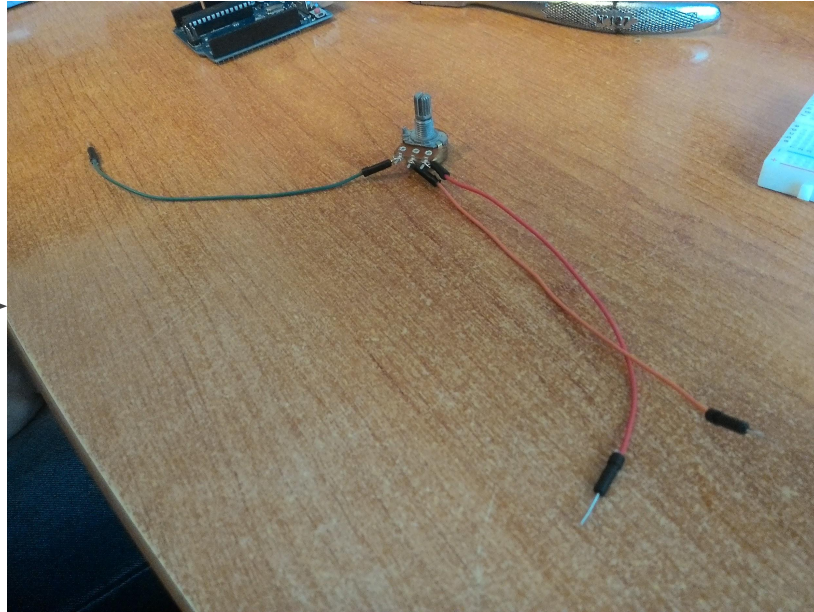
Componenti



Procedimento

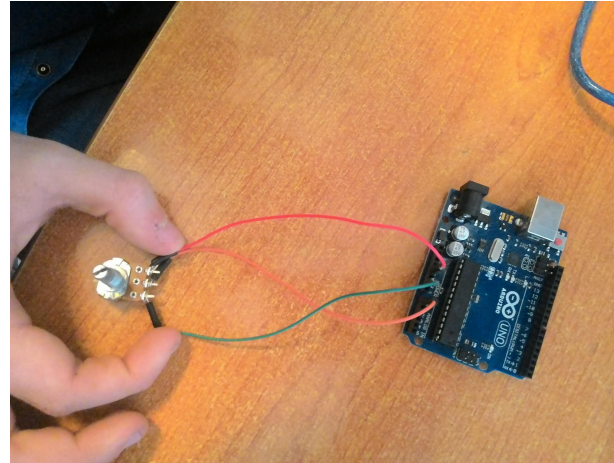
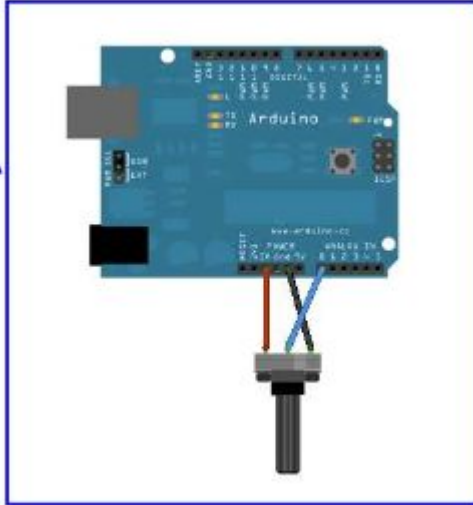
- 1) Abbiamo collegato i tre cavi alle 3 estremità del potenziometro.

**Potenziometro
e 3 cavi.**



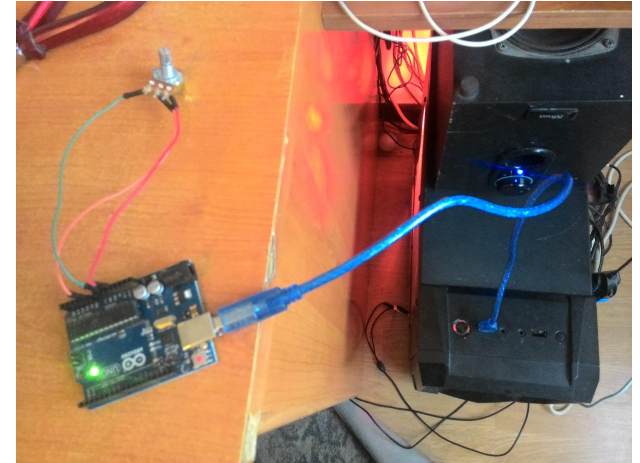
2) Abbiamo collegato i cavi dal potenziometro ad Arduino nel seguente modo: il primo cavo con la porta dei 5V , il secondo con la porta A0 e il terzo cavo con la porta GND. E poi tramite cavo USB l'abbiamo collegato al PC.

1. schema fatto con Virtual Breadboard



2. Abbiamo collegato i cavi del potenziometro ad Arduino

3. E poi al PC tramite USB



3) Abbiamo caricato nell'arduino il programma che registra i valori dati dal potenziometro.

Video YouTube: <https://www.youtube.com/watch?v=kw3jTeqvDa0>

```
AnalogReadSerial | Arduino 1.6.7
File Modifica Sketch Strumenti Aiuto
[Icons]
AnalogReadSerial
/*
 AnalogReadSerial
 Reads an analog input on pin 0, prints the result to the serial monitor.
 Graphical representation is available using serial plotter (Tools > Serial Plotter menu)
 Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

 This example code is in the public domain.
 */

// the setup routine runs once when you press reset:
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // print out the value you read:
 Serial.println(sensorValue);
 delay(1);        // delay in between reads for stability
}
```

**Schermata dell'IDE
Arduino con già il
programma
inserito.**

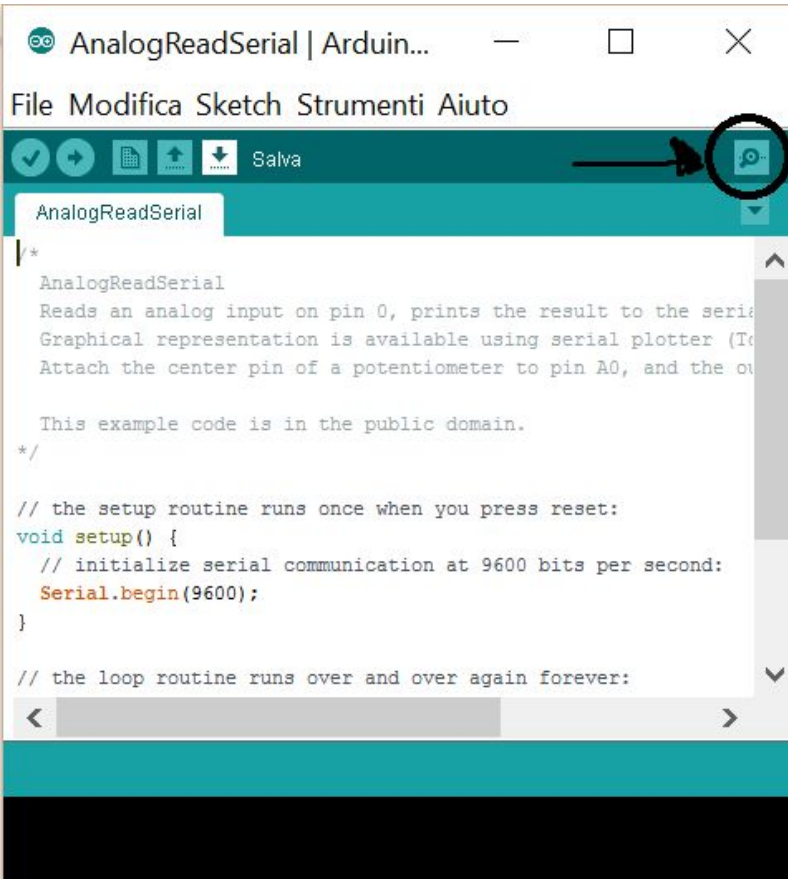


Il programma da inserire nel IDE Arduino:

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(1);  
}
```



Cliccando sulla piccola icona in alto a destra, inoltre, si potranno visualizzare i valori del potenziometro in tempo reale.



**Schermata dell'IDE
Arduino con già il
programma inserito.**

588
597
607
616
629
638
650
661
671
684
697
712
722
734
745
758
767
776
786
796
802
809
815
822
825
828
831
833
834
835
837
837
837
837
835
834
832
830
824
823
814
809
801
796
787
780
771
763
755
747
735
727
720
712

Scorrimento automatico

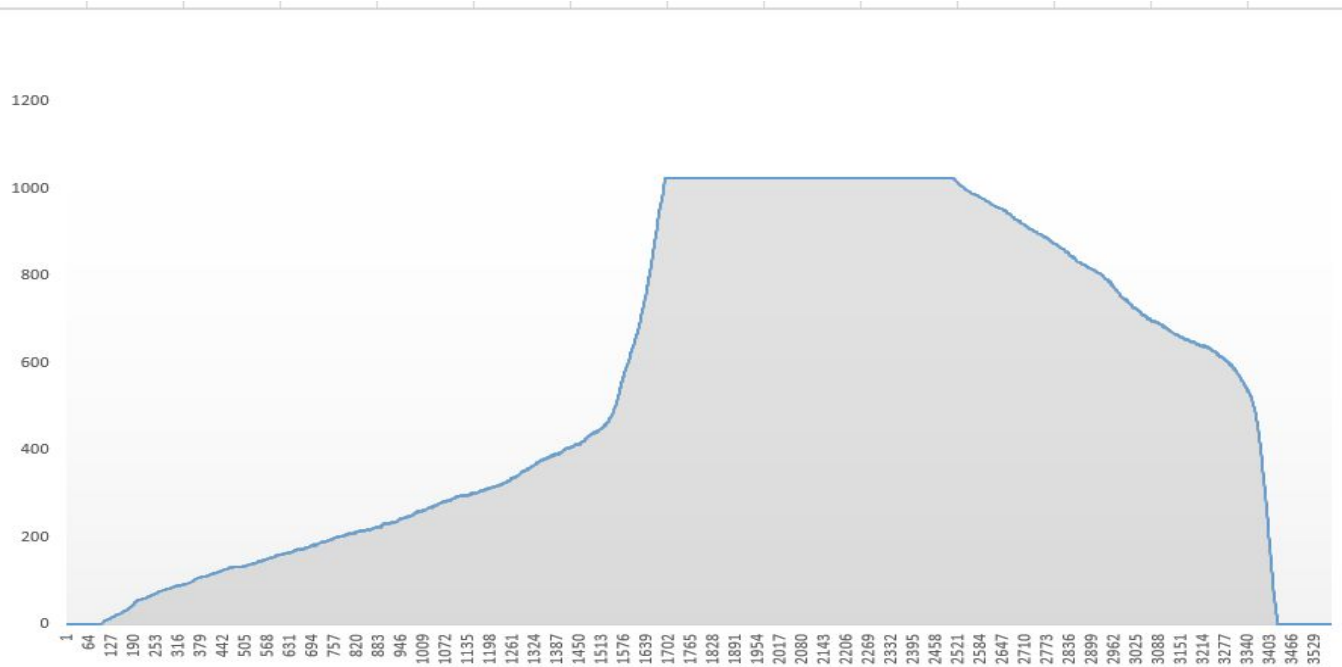
I valori del
potenziometro variano
da 0 a 1023.

**Il monitor
seriale che
si apre.**



4) Poi, tramite il programma CoolTerm abbiamo salvato i dati per poi caricarli su Excel, in modo da poterne ricavare un grafico.

Da notare, gli stacchi di velocità effettuati ruotando la manopola (più rapidi a metà e alla fine).



**Grafico
Escel.**


5) Dopo aver preso i dati e fatto il grafico su excel di questi ultimi, possiamo utilizzare il programma “Processing” per controllare graficamente e in tempo reale il nostro potenziometro.

Dopo aver avviato Processing bisogna inserire il programma per visualizzare i valori.

Inserito il programma, bisognerà premere carica per caricare il programma sull'arduino.

Si aprirà un'altra schermata dove sarà possibile, ruotando la manopola del potenziometro, vedere graficamente come i valori salgono e scendono in base alla rotazione della manopolina.

In alcuni casi bisognerà caricare sull'IDE Arduino il programma usato precedentemente per poter vedere graficamente i valori.




Il programma da inserire in Processing per ottenere un grafico isometrico:

```
import processing.serial.*;
Serial myPort;
int xPos = 1;
float inByte=0;
void setup () {
  size(800, 600);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[0], 9600);
  myPort.bufferUntil('\n');
  background(0);
}
void draw () {
  stroke(255);
  line(xPos, height, xPos, height - inByte);
```

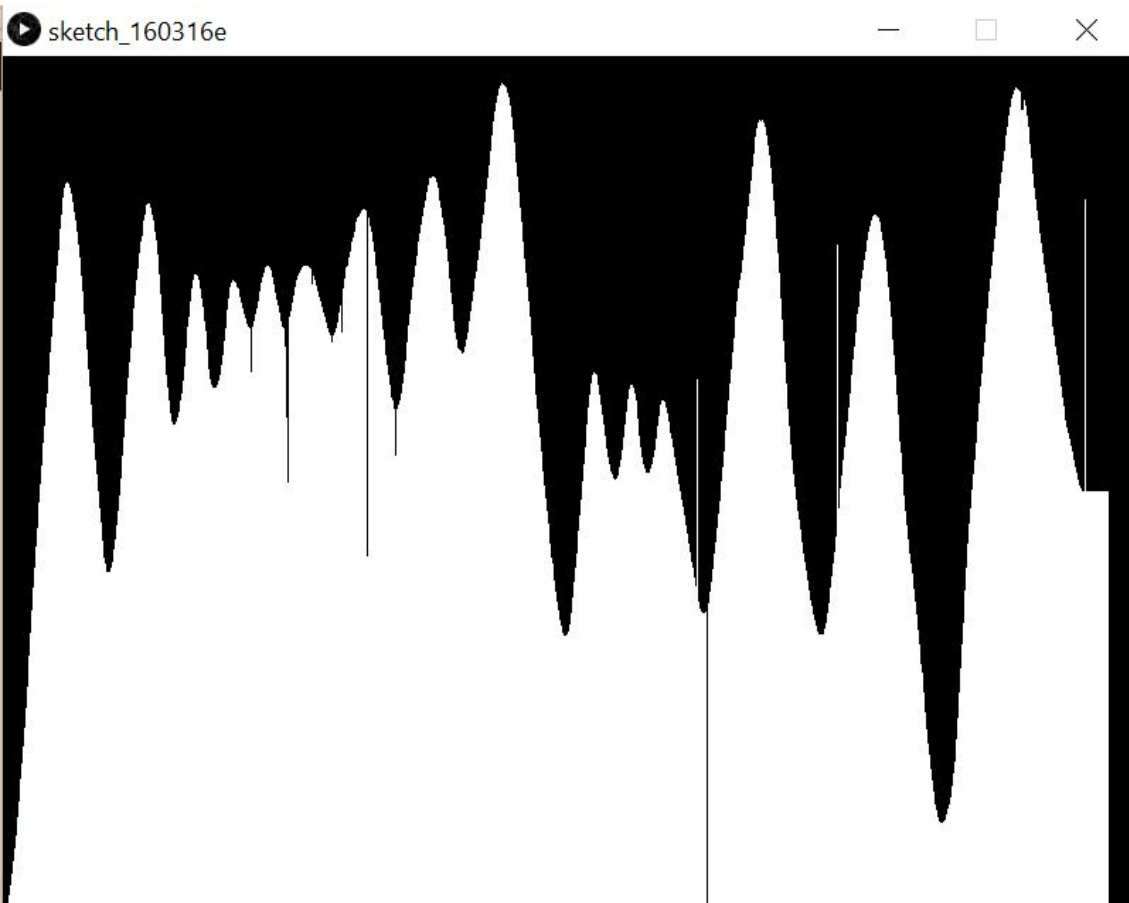
**Il codice
procede nella
slide successiva.**

```
if (xPos >= width) {  
    xPos = 0;  
    background(0);  
} else { xPos++; }  
}
```

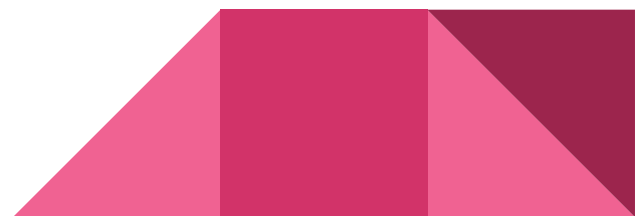
```
void serialEvent (Serial myPort) {  
    String inString = myPort.readStringUntil('\n');  
    if (inString != null) {  
        inString = trim(inString);  
        inByte = float(inString);  
        println (inByte);  
        inByte = map(inByte, 0, 1023, 0, height);  
    }  
}
```



Il grafico .



**Grafico
isometrico.**



Il programma da inserire in Processing per ottenere un tachimetro molto semplice:

```
import processing.serial.*;
Serial myPort;
float inByte;
void setup () {
  size(257, 257);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[0], 9600);
  myPort.bufferUntil('\n');
  background(0);
}
```

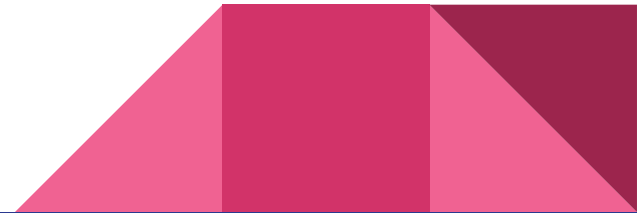
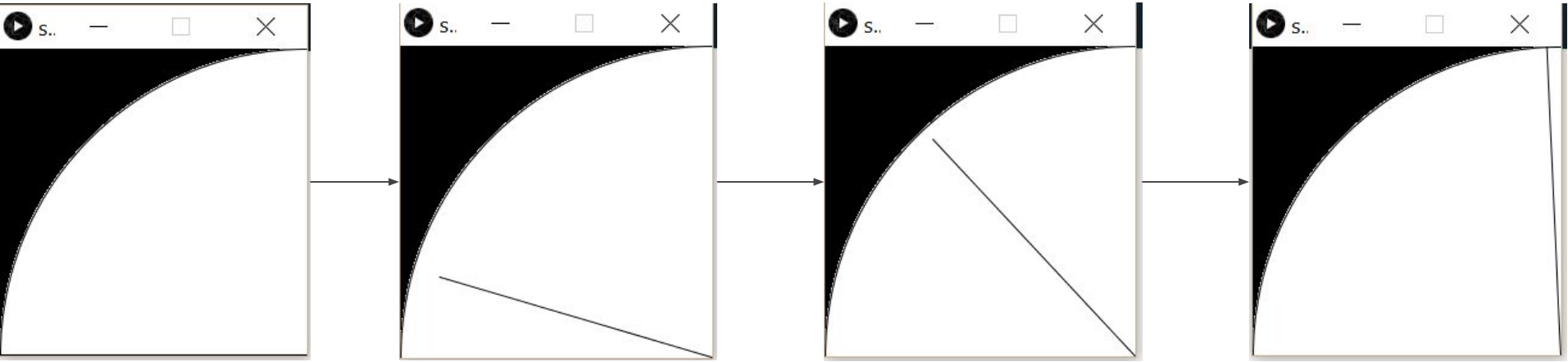
**Il codice procede
nella slide successiva.**

```
void draw () {  
  ellipse (256, 256, 512, 512);  
  line (256, 256, inByte/2, 256-inByte);  
}  
  
void serialEvent (Serial myPort) {  
  String inString = myPort.readStringUntil('\n');  
  if (inString != null) {  
    inString = trim(inString);  
    inByte = float(inString);  
    println (inByte);  
    inByte = map(inByte, 0, 512, 0, height);  
  }  
}
```



Il grafico .

Il Tachimetro



Conclusioni

Analog Read Serial permette di prelevare dati dall'Arduino in poco tempo.

Si può usare non solo per prelevare i dati da un potenziometro, ma anche per prelevare, ad esempio, la temperatura, la profondità e altri dati.

Conti Christian & Vergassola Nicolò 3Q

