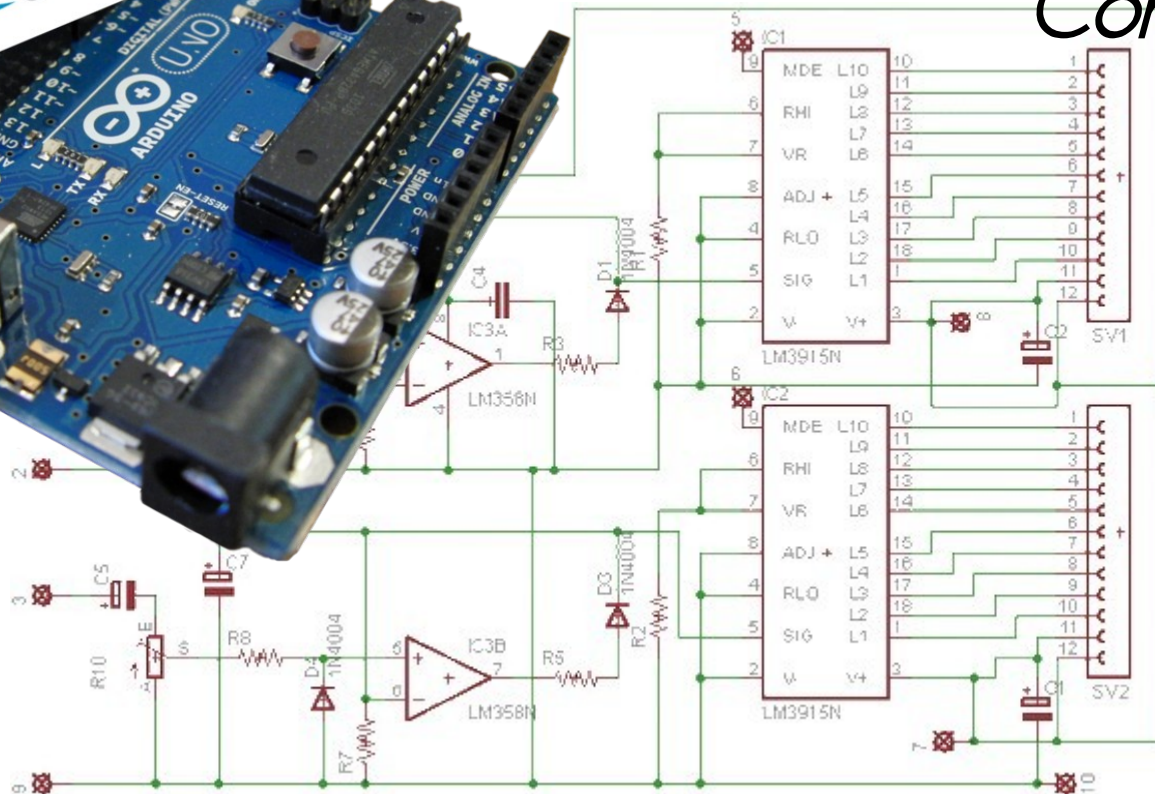
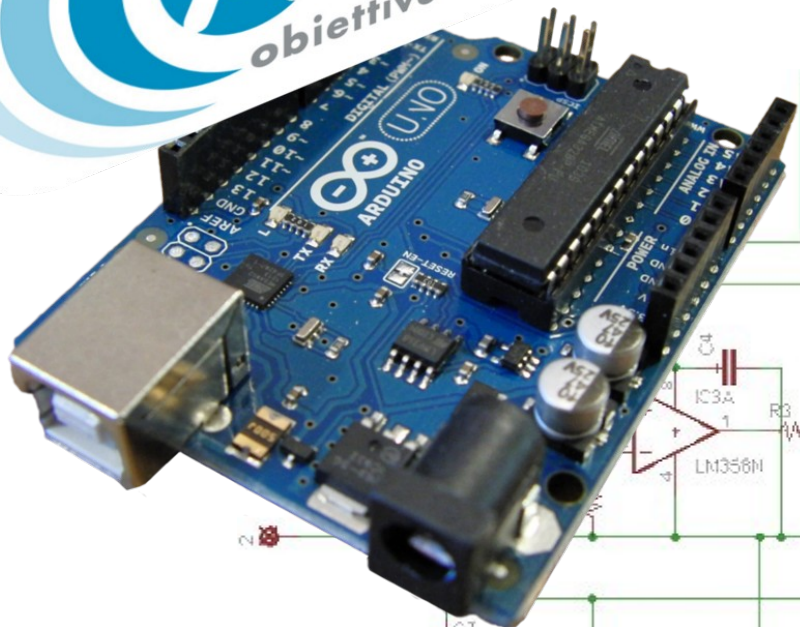


# CORSO ARDUINO



Giulio Fieramosca  
Stefano Panichi  
*Corso ASEV 2014*

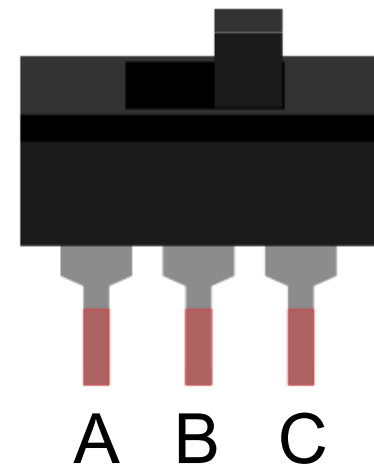
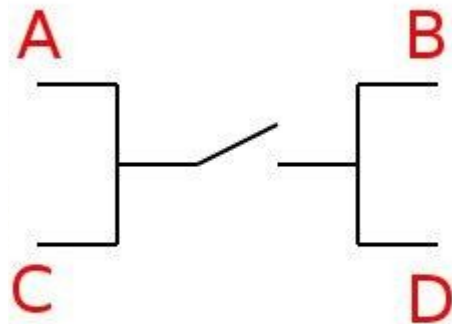


# Pulsanti e interruttori

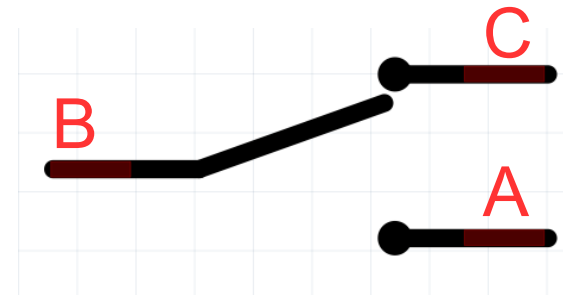
Pushbutton

Momentary  
button

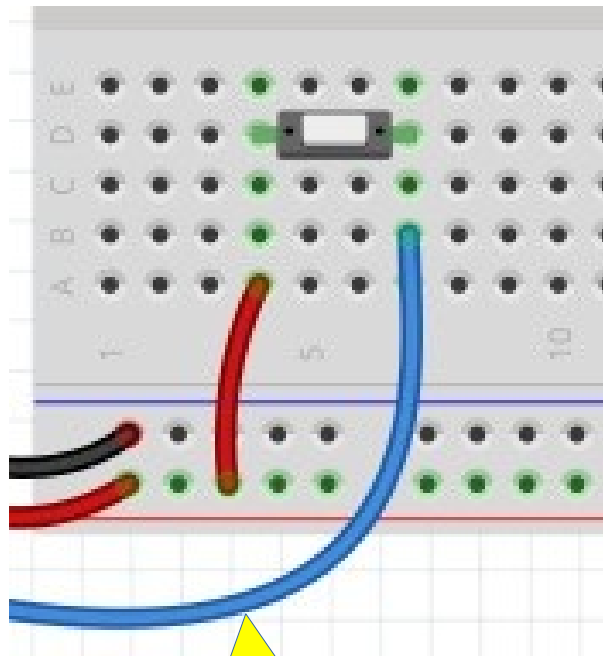
Pulsante  
monostabile



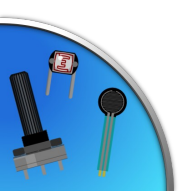
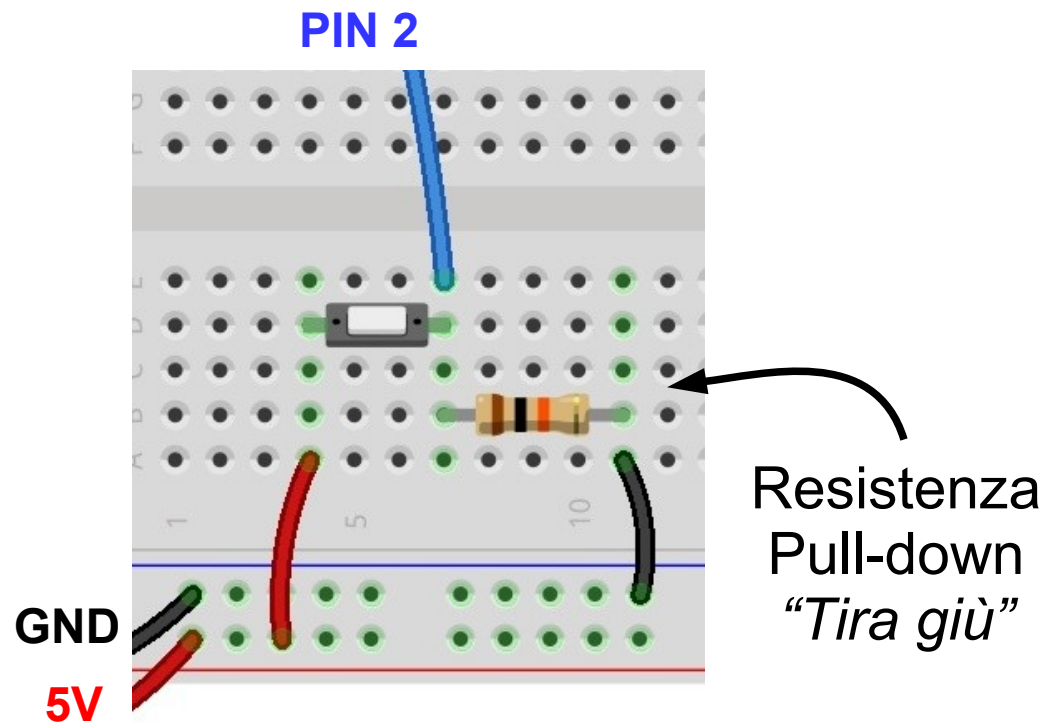
Switch  
Interruttore  
bistabile



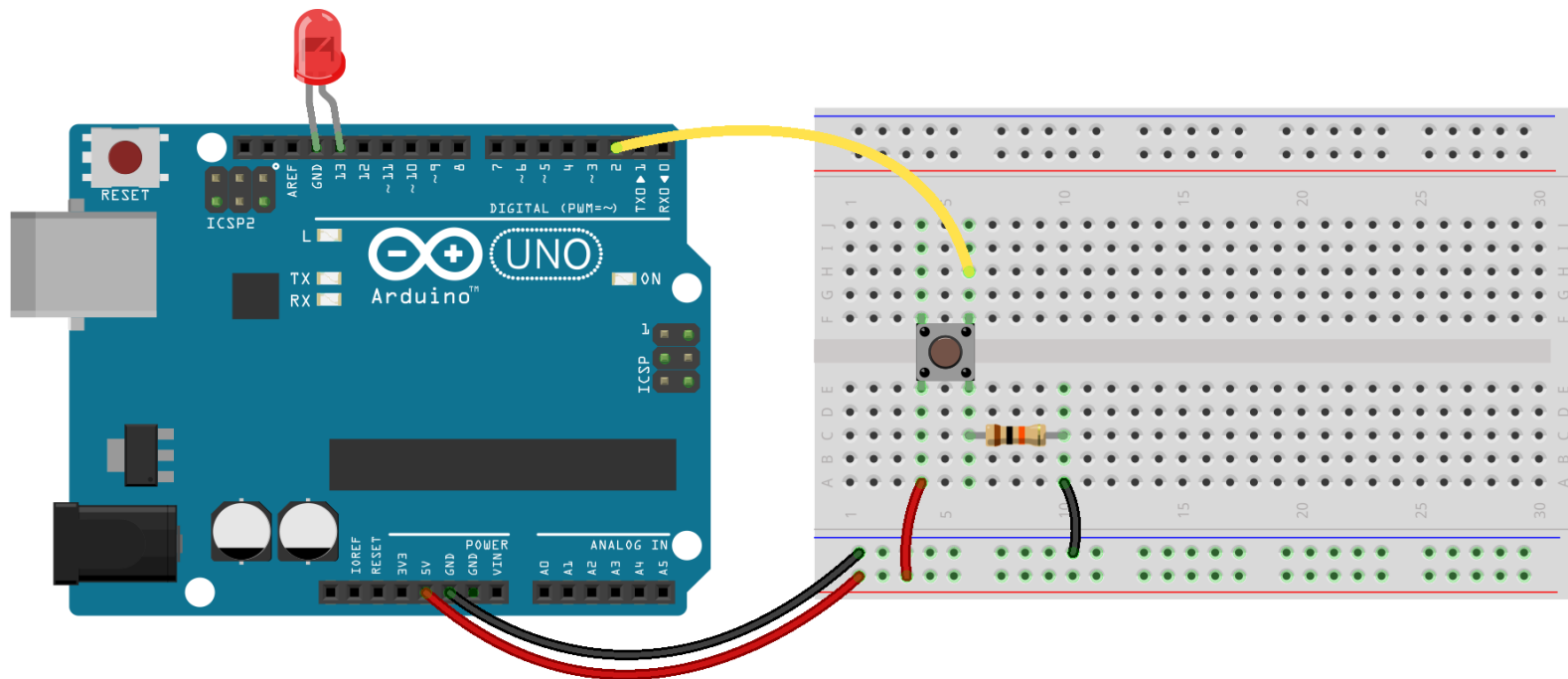
# Pull-down e cortocircuiti...



Interferenze elettrostatiche



# Pulsanti e interruttori



fritzing

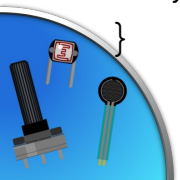
# Pulsanti e interruttori - Listato

```
const byte PIN_BOTTONE = 2;           // pin del bottone
const byte PIN_LED = 13;              // pin del led
byte statoBottone = 0;               // variabile di stato del bottone

void setup() {
  pinMode(PIN_LED, OUTPUT);          // il pin del LED è in OUTPUT
  pinMode(PIN_BOTTONE, INPUT);       // il pin del bottone è in INPUT
}

void loop() {
  statoBottone = digitalRead(PIN_BOTTONE); // legge lo stato
                                              // del bottone

  if (statoBottone == HIGH) {         // se è HIGH
    digitalWrite(PIN_LED, HIGH);      // accende il led
  }
  else {
    digitalWrite(PIN_LED, LOW);       // altrimenti lo spegne
  }
}
```



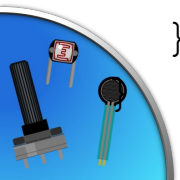
# Pulsanti e interruttori - Listato

```
const byte PIN_BOTTONE = 2;           // pin del bottone
const byte PIN_LED = 13;              // pin del led
byte statoBottone = 0;                // variabile di stato del bottone

void setup() {
  pinMode(PIN_LED, OUTPUT);           // il pin del LED è in OUTPUT
  pinMode(PIN_BOTTONE, INPUT_PULLUP); // il pin del bottone è in INPUT, con pullup interno!
}

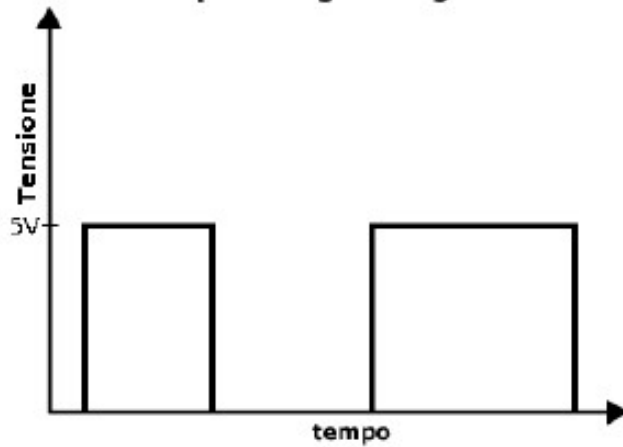
void loop() {
  statoBottone = digitalRead(PIN_BOTTONE); // legge lo stato
                                              // del bottone

  if (statoBottone == LOW) {             // se è LOW
    digitalWrite(PIN_LED, HIGH);        // accende il led
  }
  else {
    digitalWrite(PIN_LED, LOW);         // altrimenti lo spegne
  }
}
```

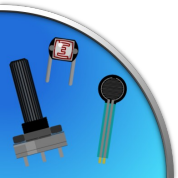
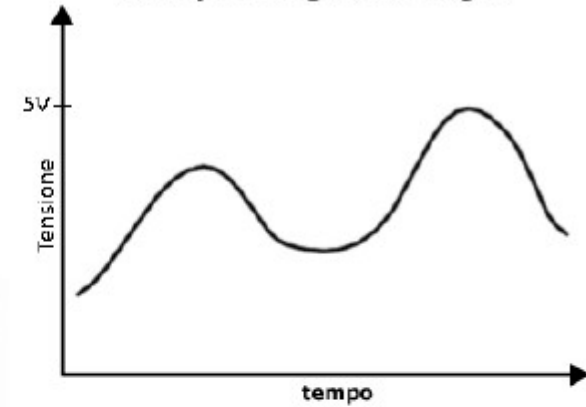


# Il segnale analogico

esempio di segnale digitale



esempio di segnale analogico

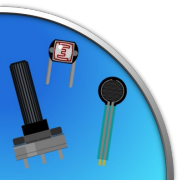


# Sensori Analogici



**Resistenze variabili**, vanno combinati con altri componenti per leggere una variazione di tensione

**Sensori già pronti**, che forniscono in output una tensione variabile, di solito compresa fra lo 0 e i 5v





# Resistenze variabili



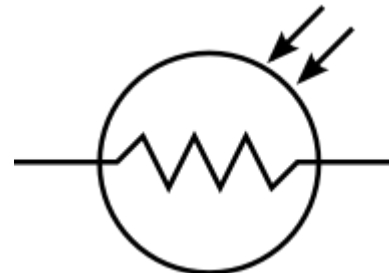
## Potenziometro:

Si varia la resistenza ruotando una manopola



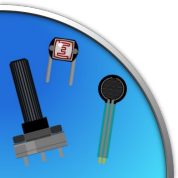
## Termistore (PTC o NTC):

Varia la resistenza con la temperatura

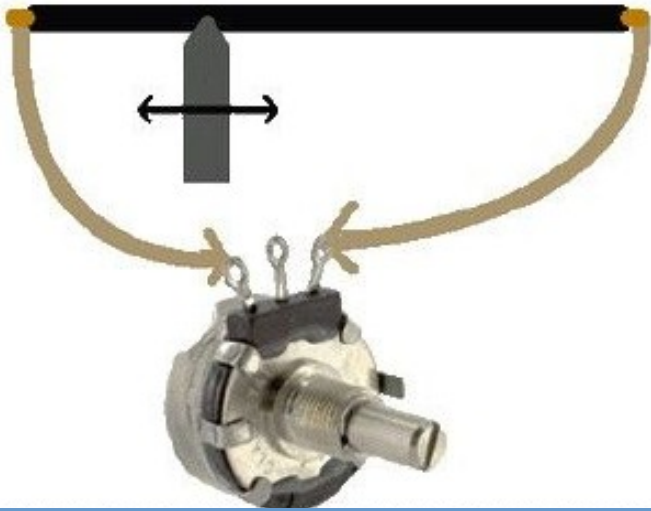


## Fotoresistenza:

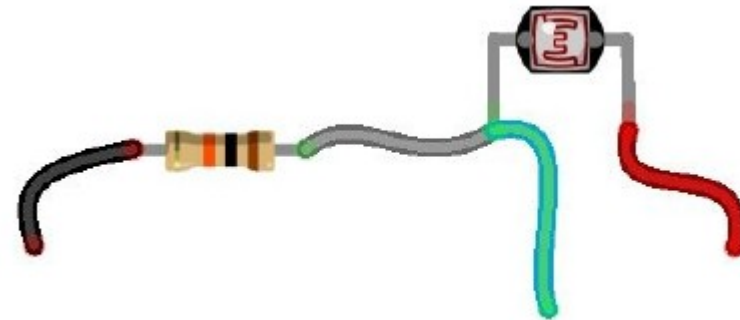
Varia la resistenza con la luminosità



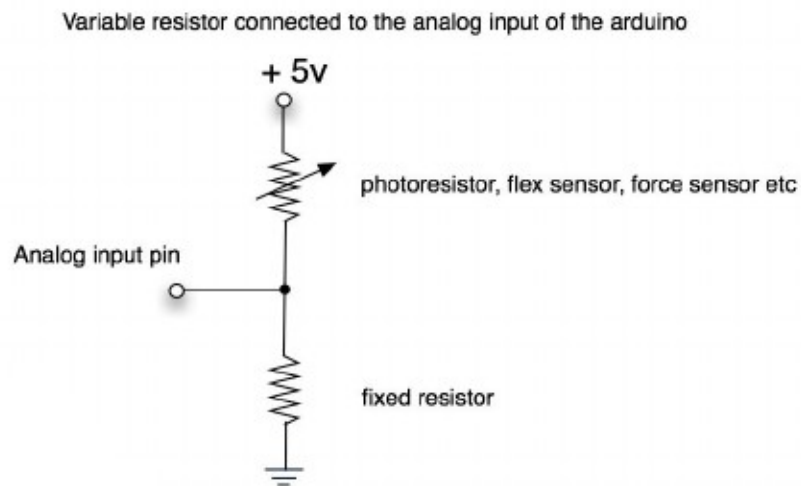
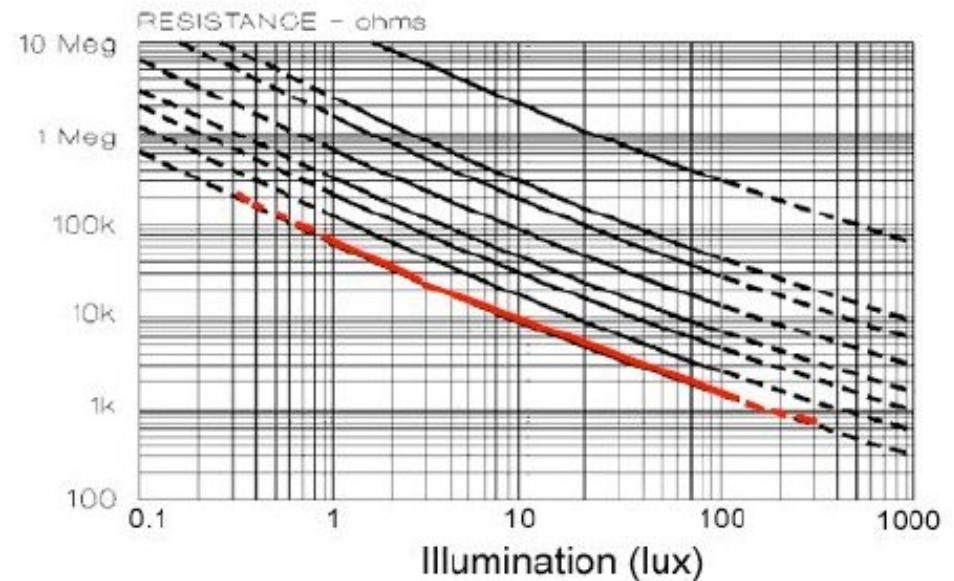
# Partitori di tensione



Funzionamento di un potenziometro

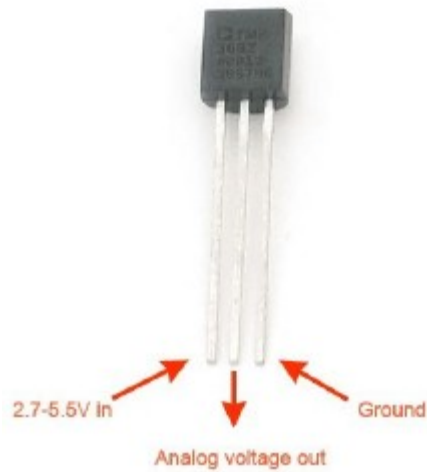


## Resistance vs. Illumination



# Sensori integrati

Temperatura *TMP36*

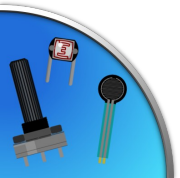
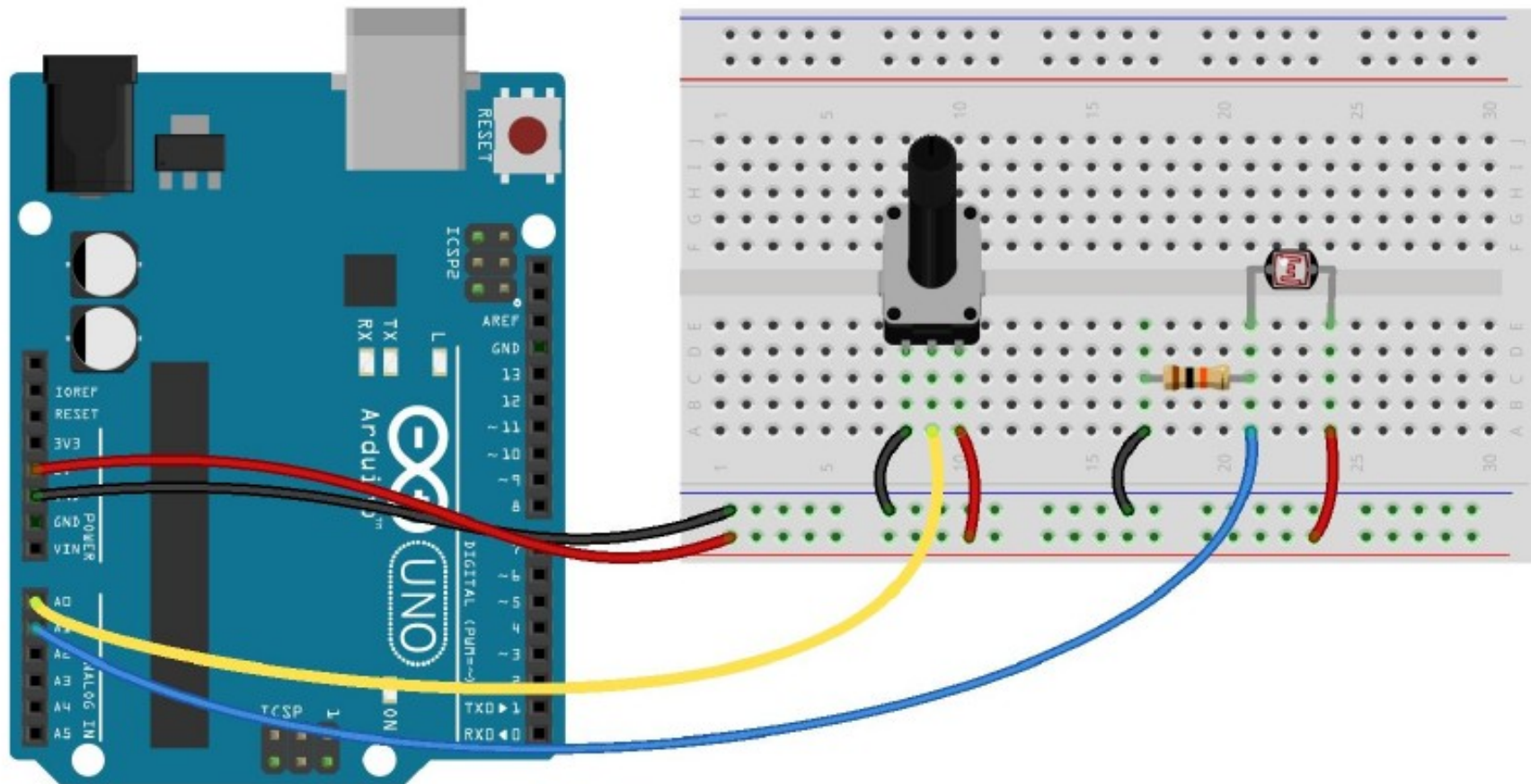


Accelerometro



Distanza

# Leggi i sensori - Schema

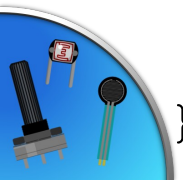


# Leggi i sensori - Listato

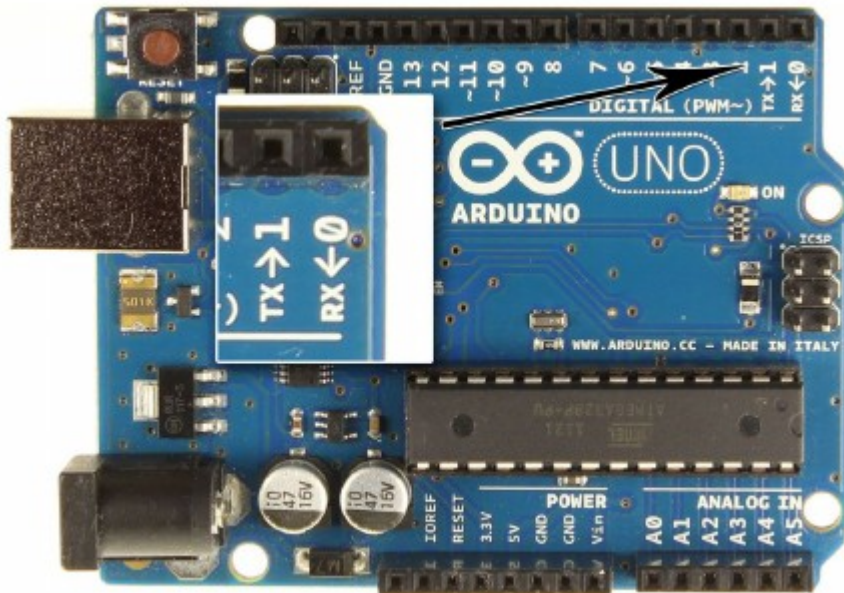
```
const byte PIN_POT = A0;    // Pin del potenziometro
const byte PIN_FOTORES = A1; // Pin della fotoresistenza
int valoreLetto = 0; // Variabile con il valore letto

void setup() {
    Serial.begin(9600);      // Avvia la comunicazione
                             // seriale a 9600 baud
}

void loop() {
    valoreLetto = analogRead(PIN_POT);
    //Legge il valore del sensore, e lo stampa
    Serial.print("valore Potenziometro = ");
    Serial.println(valoreLetto);
    valoreLetto = analogRead(PIN_FOTORES);
    //Legge il valore del sensore, e lo stampa
    Serial.print("valore Fotoresistenza = ");
    Serial.println(valoreLetto);
    delay(1000); // 1 secondo di pausa per poter
                 // leggere i valori
}
```



# Comunicazione seriale



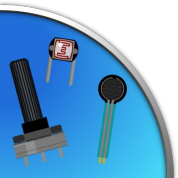
```
Serial.begin(9600);
```

```
Serial.available()
```

```
var = Serial.read();
```

```
Serial.print("stringa");
```

```
Serial.println(var);
```





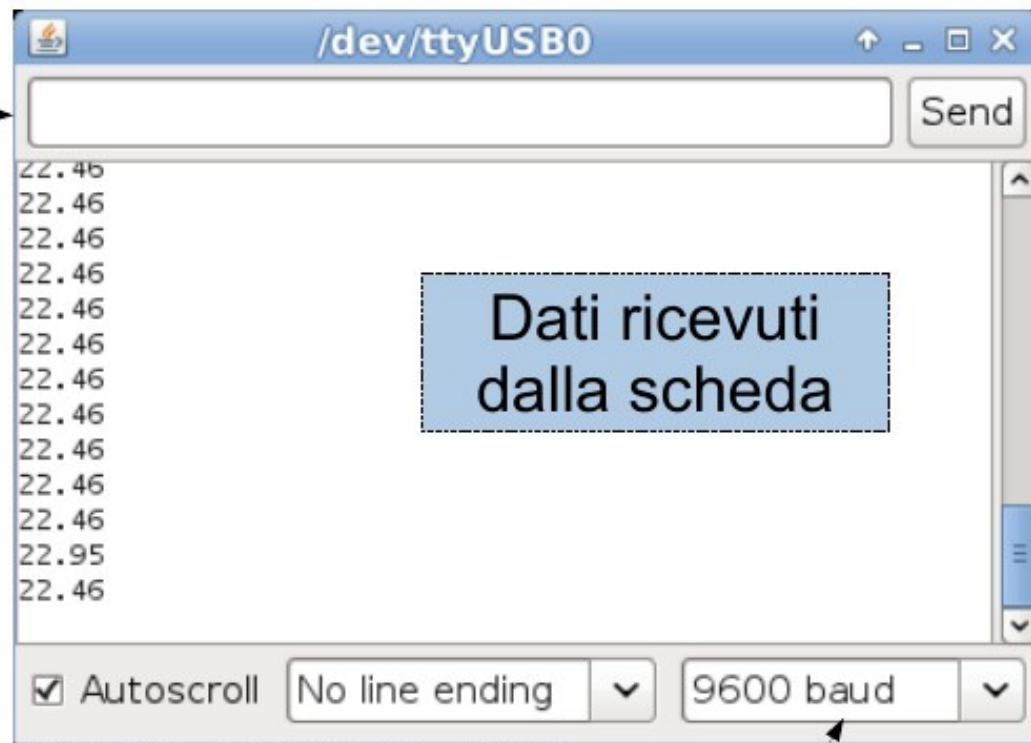
# Monitor Seriale

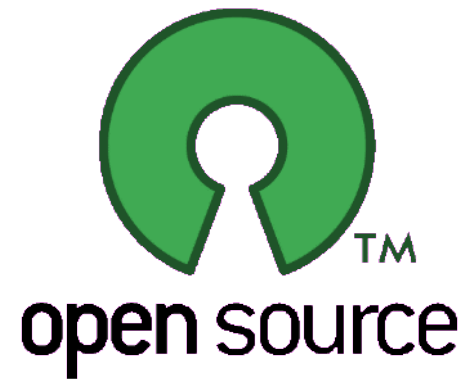
Bottone per aprire il serial monitor

Inviare dati alla scheda

Dati ricevuti dalla scheda

Velocità di trasmissione (default 9600)





Presentazione realizzata con software open source  
(LibreOffice Impress, Gimp, Arduino, Fritzing)

Quest'opera è distribuita con Licenza **CC-BY-SA**  
e realizzata da *Stefano Panichi* e *Giulio Fieramosca*