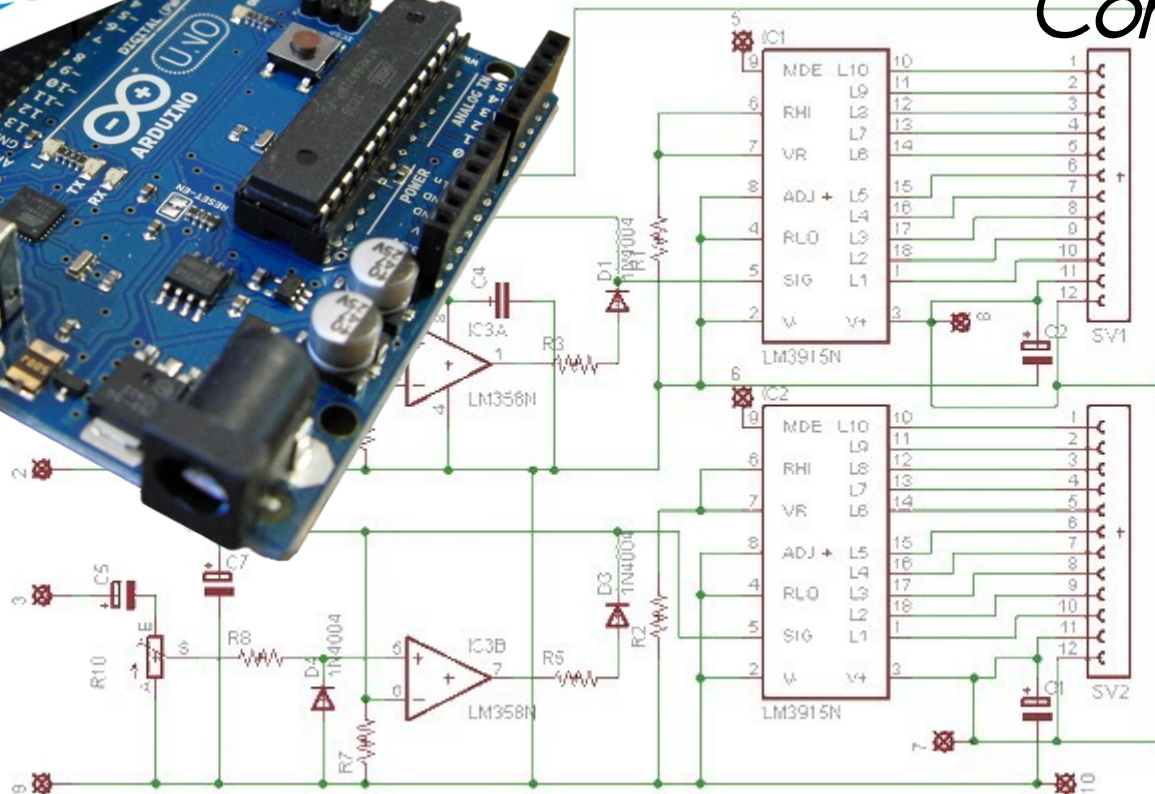
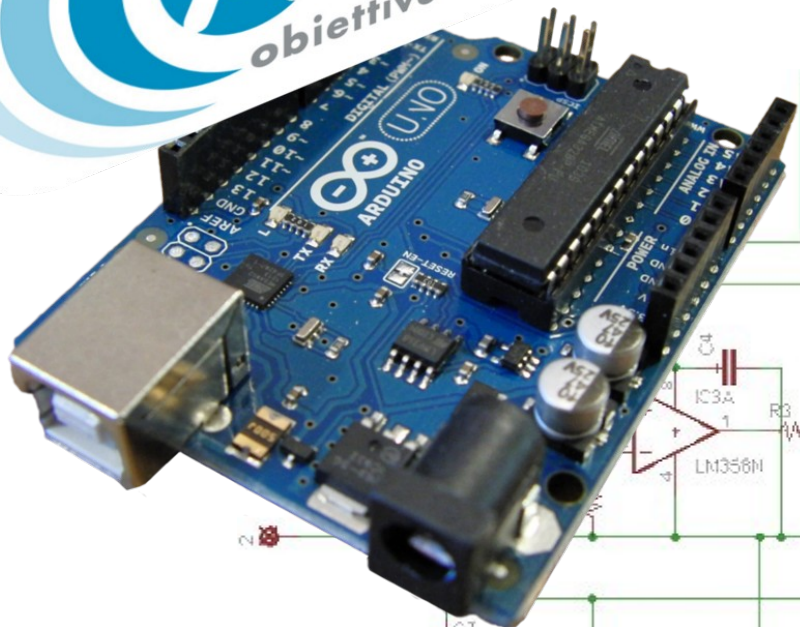


CORSO ARDUINO



Giulio Fieramosca
Stefano Panichi
Corso ASEV 2015

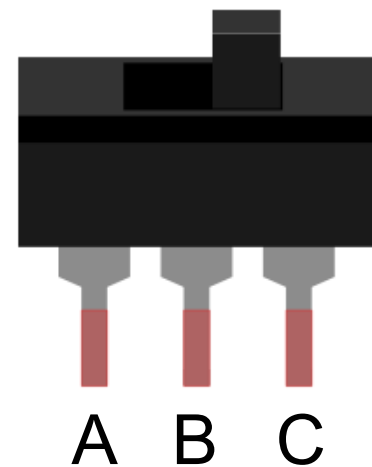
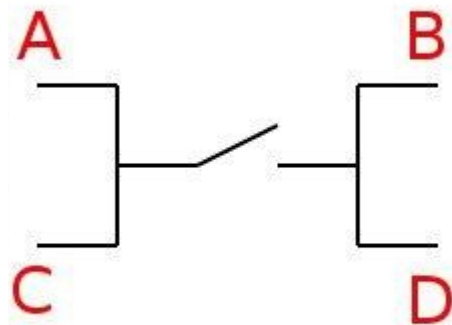


Pulsanti e interruttori

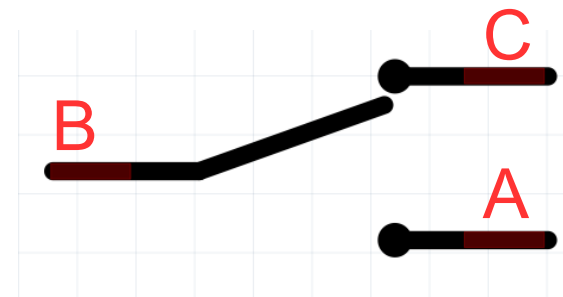
Pushbutton

Momentary
button

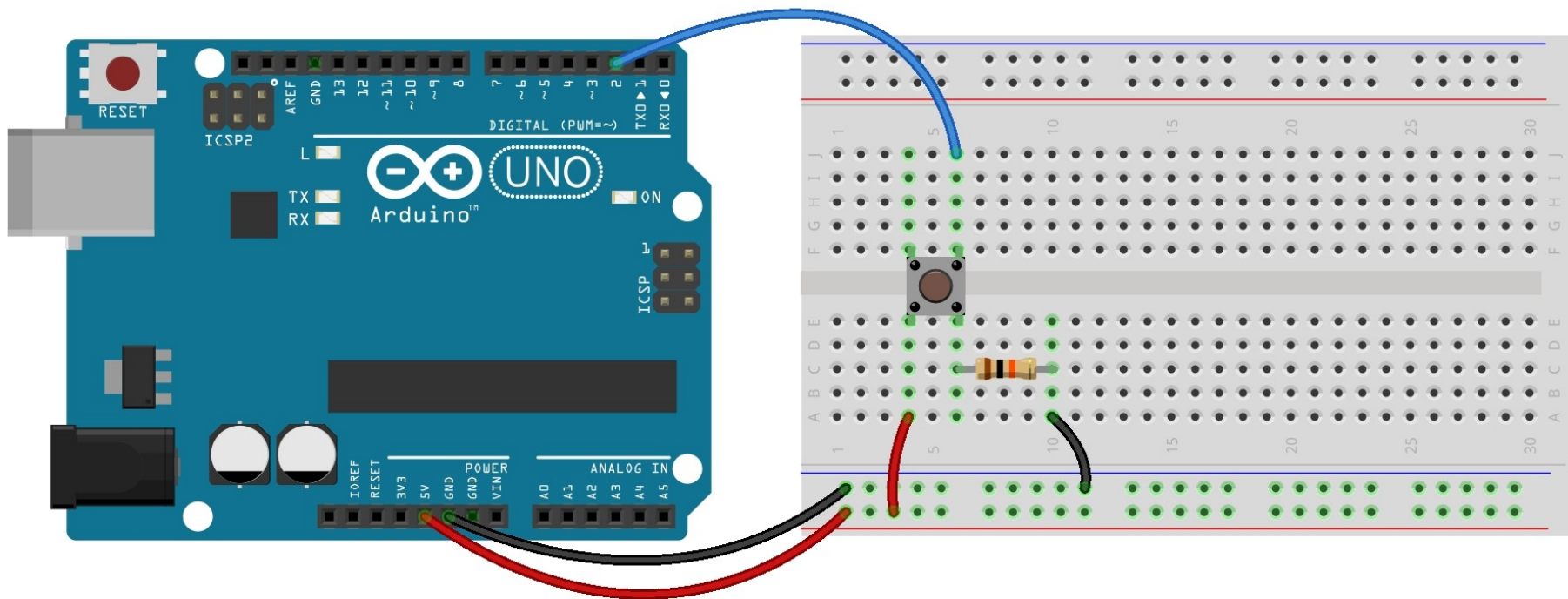
Pulsante
monostabile



Switch
Interruttore
bistabile

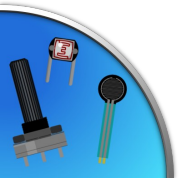
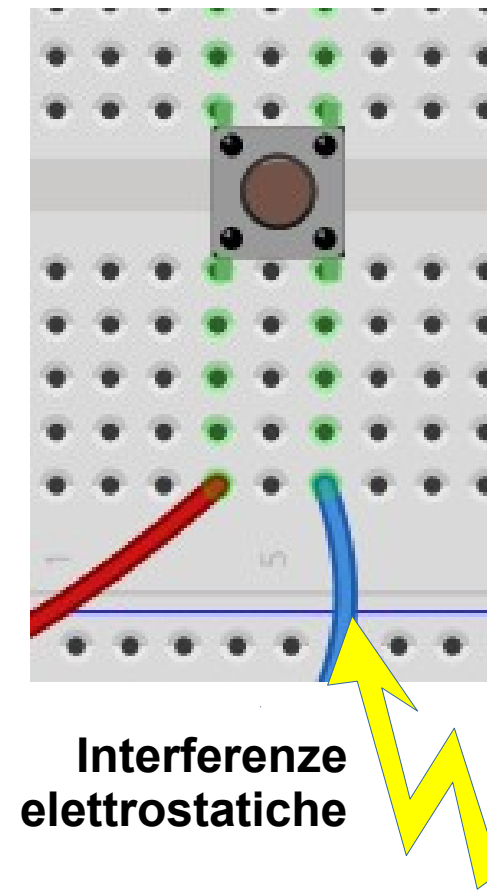
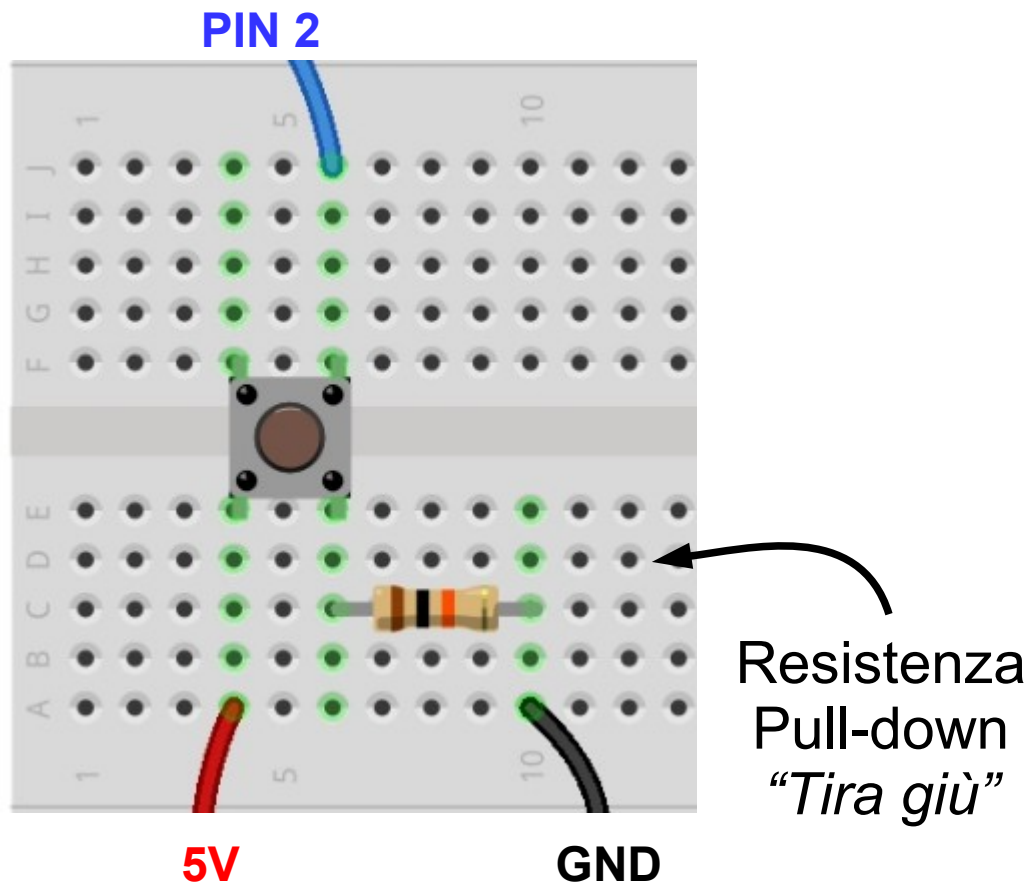


Pulsanti e interruttori

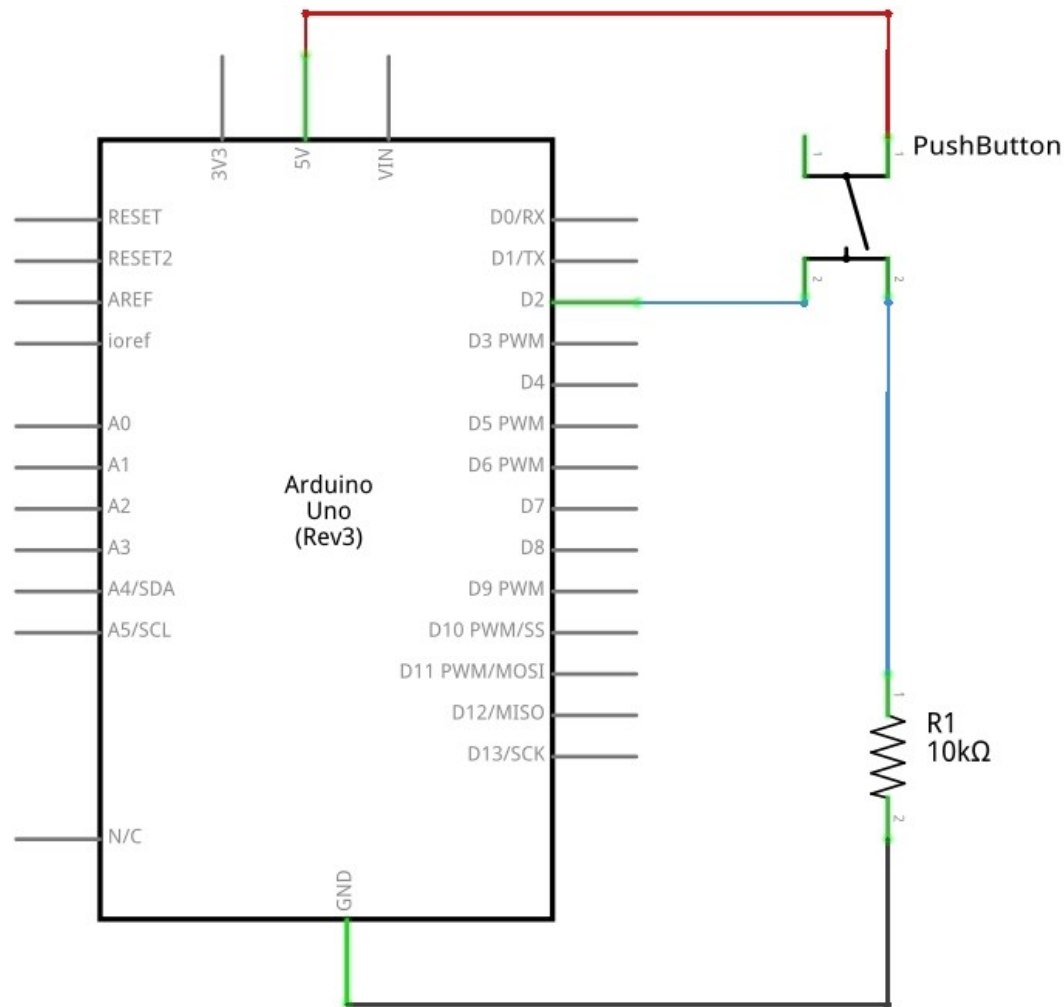


fritzing

Pull-down e cortocircuiti...



Pulsanti e interruttori



fritzing

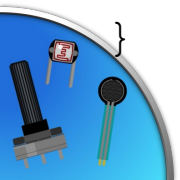
Pulsanti e interruttori - Listato

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

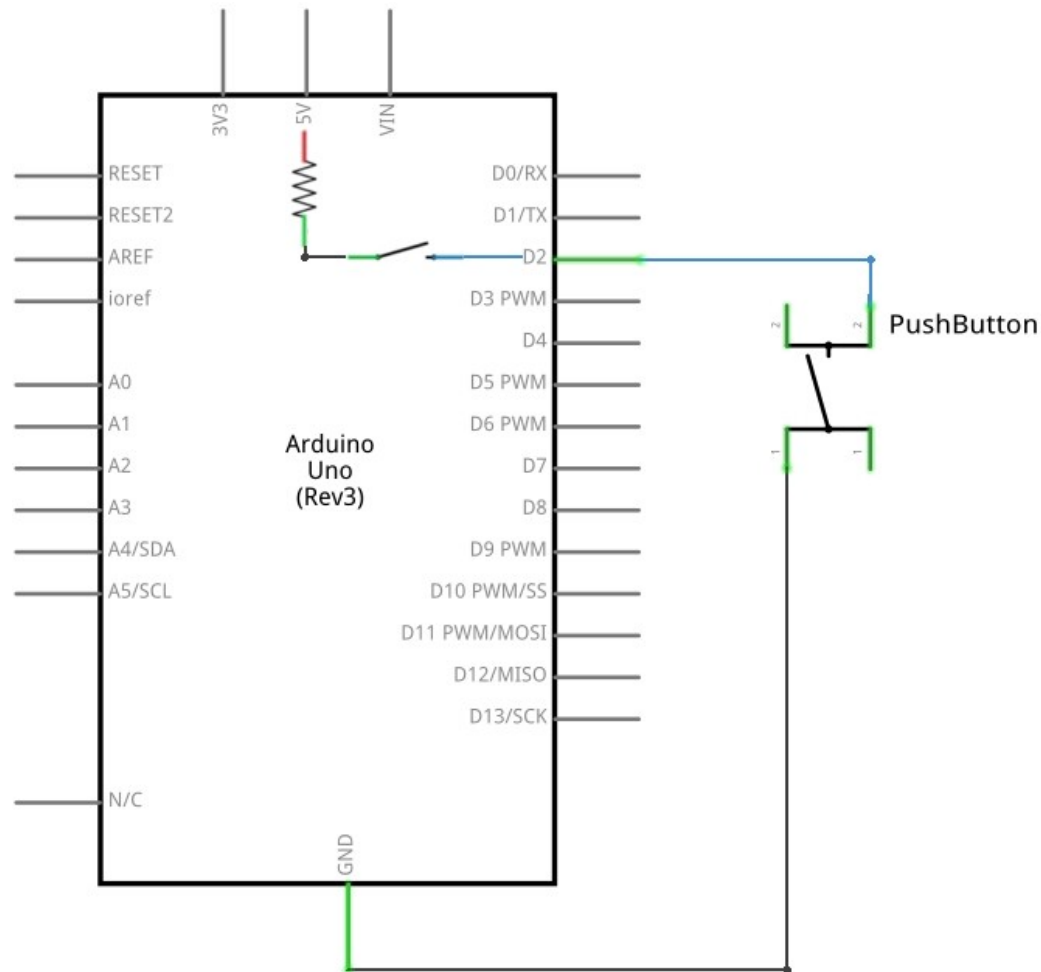
void setup() {
  pinMode(PIN_LED, OUTPUT);           // il pin del LED è in OUTPUT
  pinMode(PIN_BOTTONE, INPUT);        // il pin del bottone è in INPUT
}

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
  }
  delay(10);
}
```

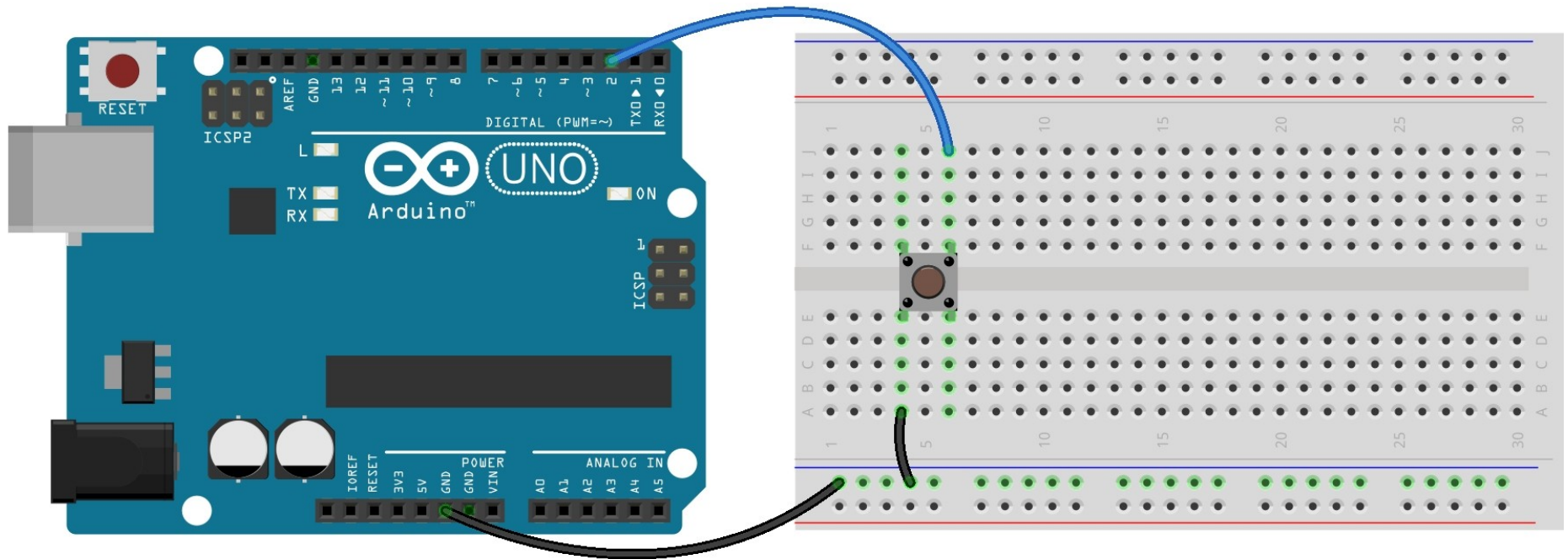


PullUp Interno

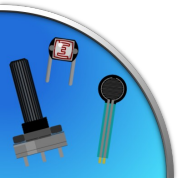


fritzing

PullUp Interno



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_PULLUP); // il pin del bottone è in INPUT, con pullup interno!
}

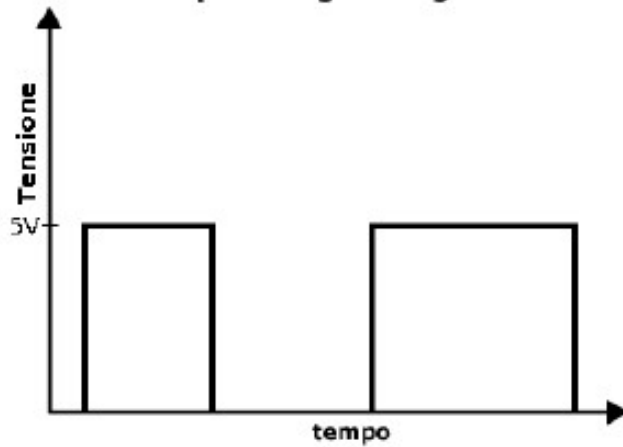
void loop() {
  statoBottone = digitalRead(PIN_BOTTONE); // legge lo stato
                                              // del bottone

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

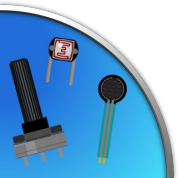
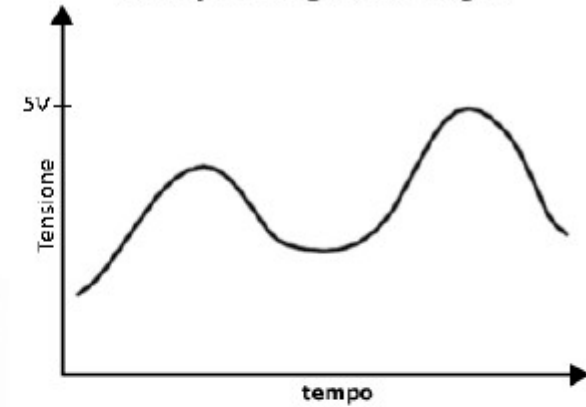


Segnali digitale e analogico

esempio di segnale digitale



esempio di segnale analogico

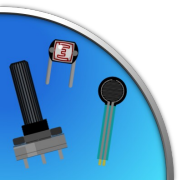


Sensori Analogici



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



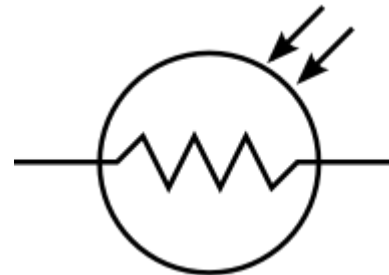
Potenziometro:

Si varia la resistenza ruotando una manopola



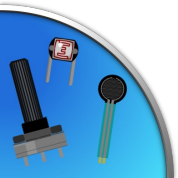
Termistore (PTC o NTC):

Varia la resistenza con la temperatura

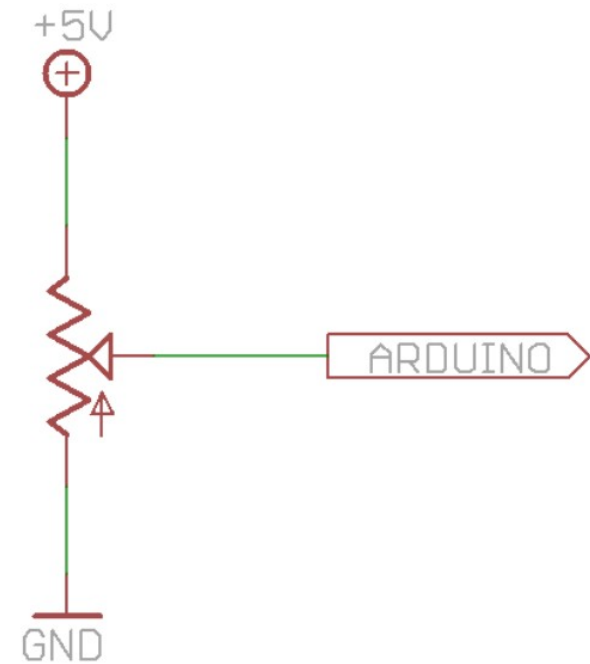
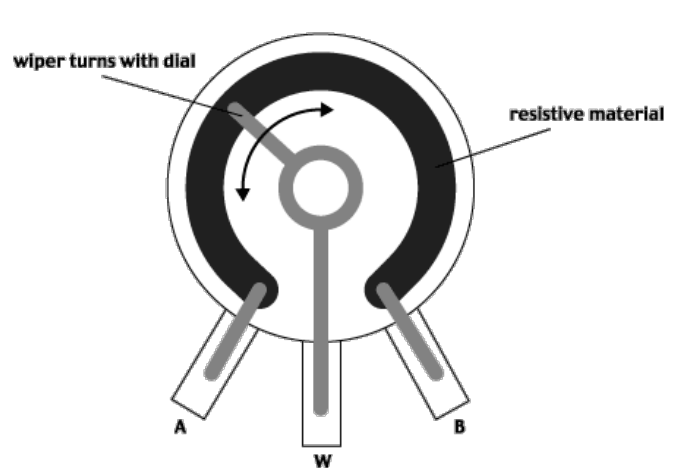


Fotoresistenza:

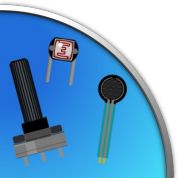
Varia la resistenza con la luminosità



Potenzziometro



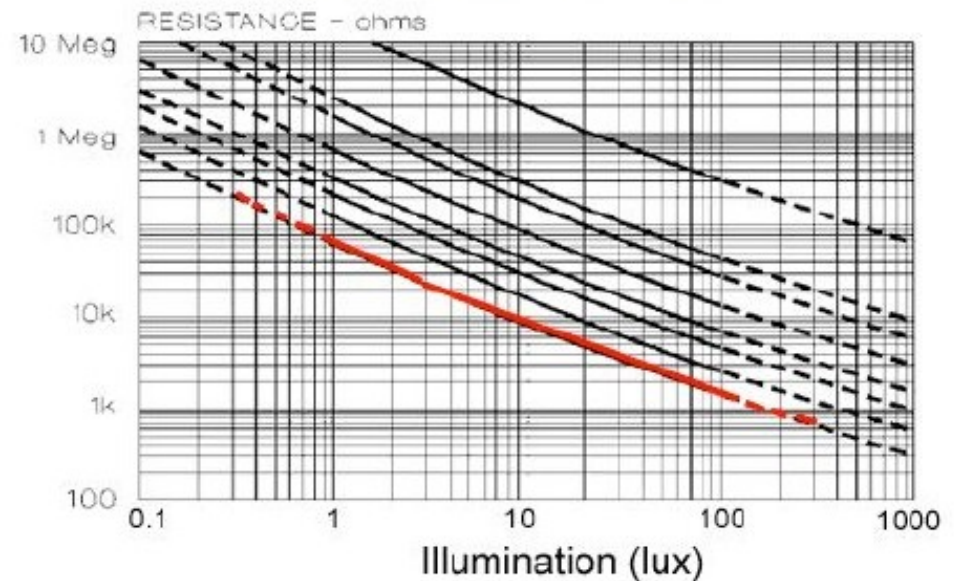
Funzionamento di un
potenziometro



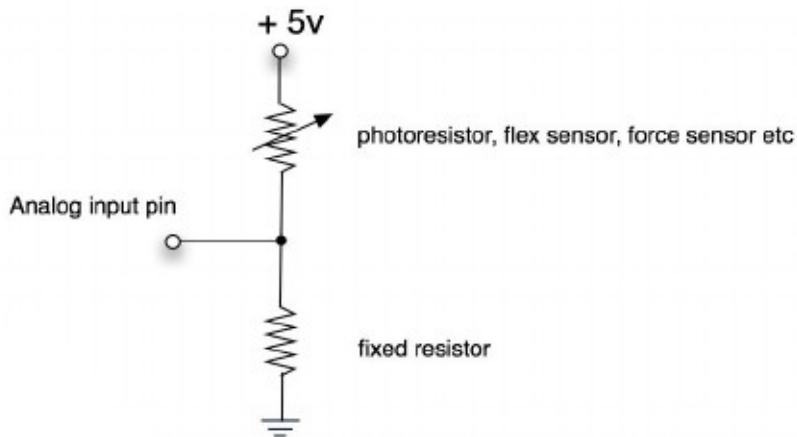
Partitore di tensione con fotoresistenza



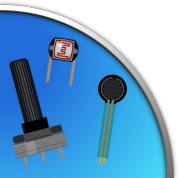
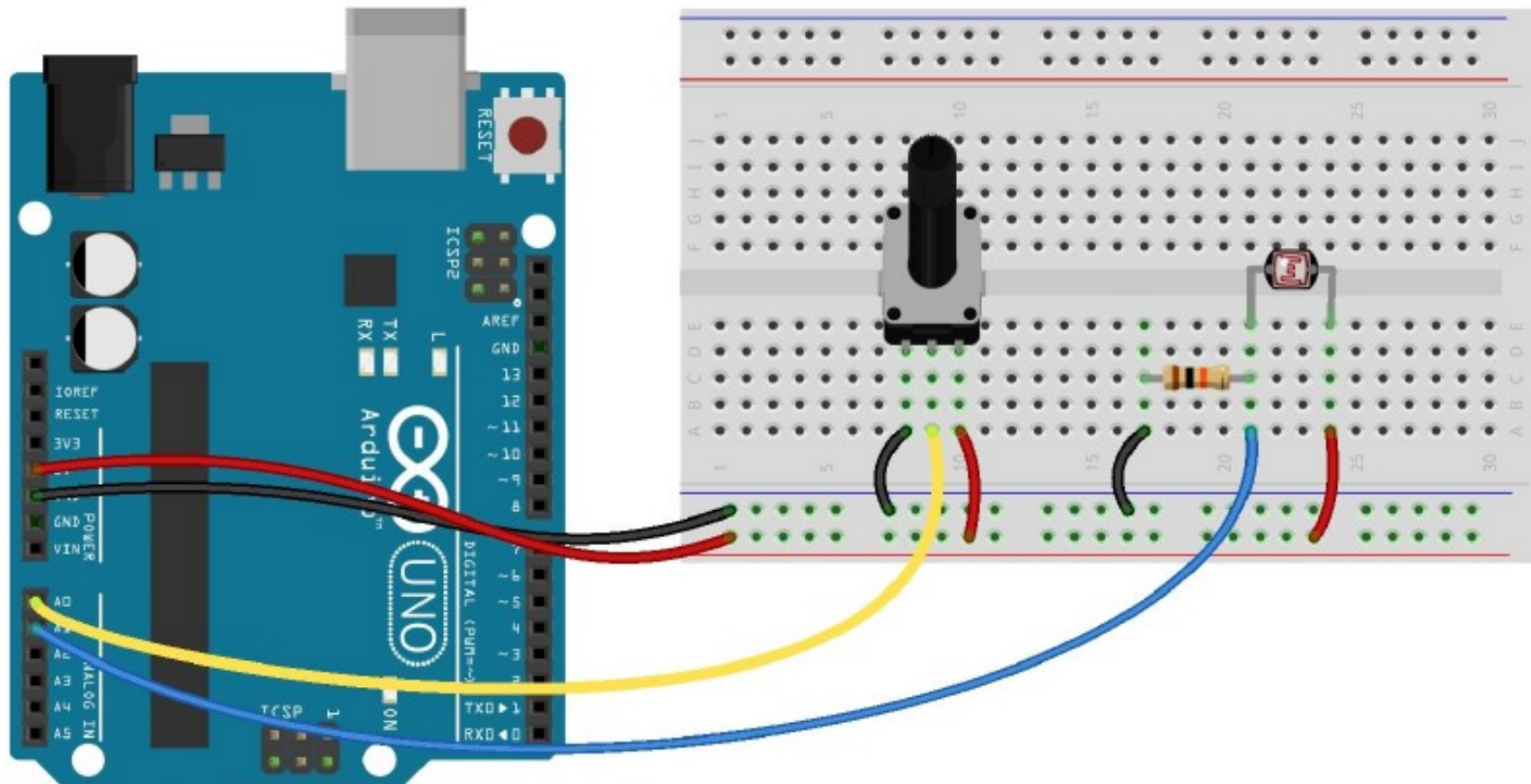
Resistance vs. Illumination



Variable resistor connected to the analog input of the arduino



Leggi il sensore - Schema

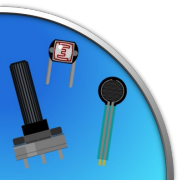


Leggi il sensore - Listato

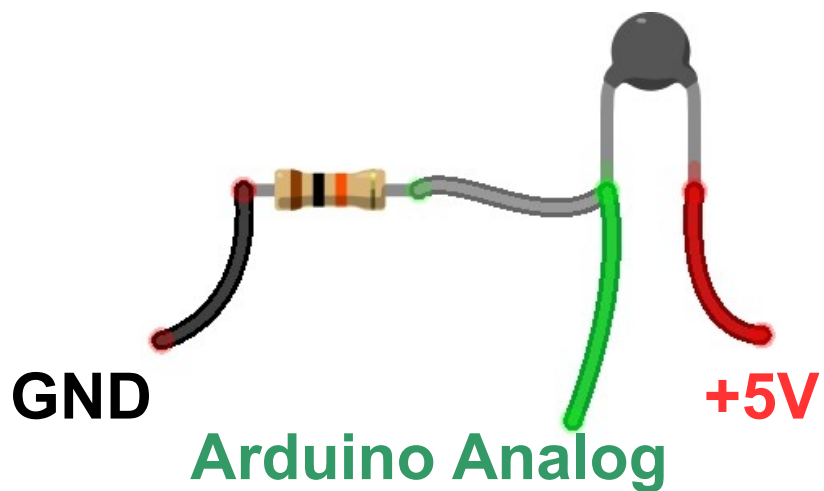
```
const byte PIN_POTENZIOMETRO = A0; // Pin del potenziometro
const byte PIN_FOTORESISTENZA = A1; // Pin della fotoresistenza
int valoreSensore = 0; // Variabile con il valore letto

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

void loop() {
  //Legge il valore del sensore, e lo stampa
  valoreSensore = analogRead(PIN_POTENZIOMETRO);
  Serial.print("potenziometro = ");
  Serial.println(valoreSensore);
  valoreSensore = analogRead(PIN_FOTORESISTENZA);
  Serial.print("fotoresistenza = ");
  Serial.println(valoreSensore);
  delay(1000); // 1 secondo di pausa per poter
               // leggere i valori
}
```

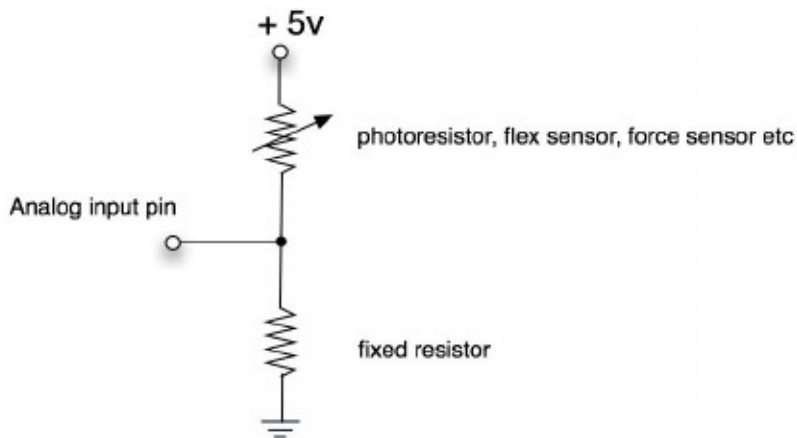


Partitore di tensione con termoresistenza



$$R = 10k \cdot \left(\frac{1024}{\text{Valore Letto}} - 1 \right)$$

Variable resistor connected to the analog input of the arduino



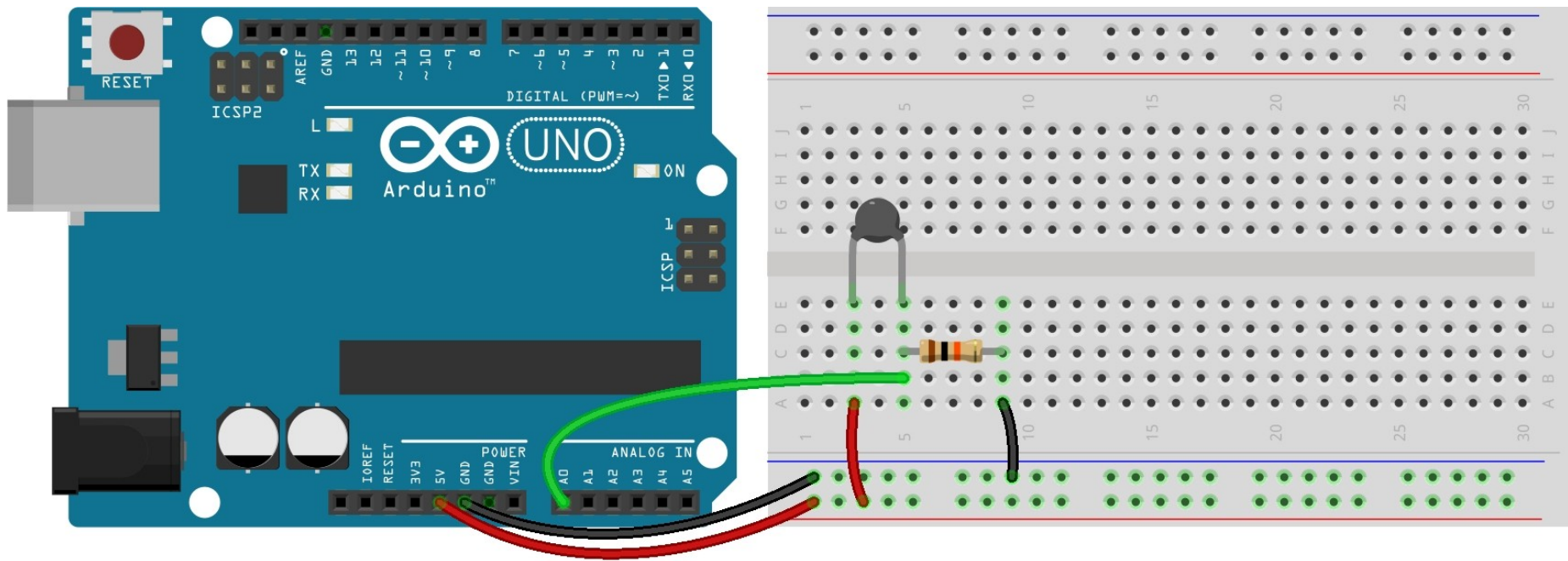
Parametri necessari:

Resistenza nominale: 10k

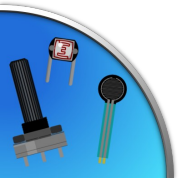
Temperatura nominale: 25°C → 298,15K

Coefficiente B: 3435

Sensore di temperatura: conversione in gradi centigradi



fritzing

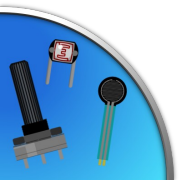


Sensore di temperatura: conversione in gradi centigradi

```
const int TMP_NOMINALE = 25;
const int RESISTENZA_SERIE = 10; // In kilohm
const int RESISTENZA_NOMINALE = 10; // In kilohm
const int COEFFICIENTE_B = 3435;
const byte PIN_TERMOMETRO = A0;

float ottieniTemperatura (const float resistenza) {
    float temperatura = log10(resistenza/RESISTENZA_NOMINALE);
    temperatura /= COEFFICIENTE_B;
    temperatura += 1.0 / (TMP_NOMINALE + 273.15);
    temperatura = 1.0 / temperatura;
    return ( temperatura - 273.15 );
}

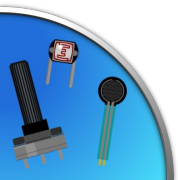
float ottieniResistenza (const byte pin) {
    float resistenza = analogRead(pin);
    resistenza = 1024.0 / resistenza;
    resistenza--;
    resistenza *= RESISTENZA_SERIE;
    return resistenza;
}
```



Sensore di temperatura: conversione in gradi centigradi

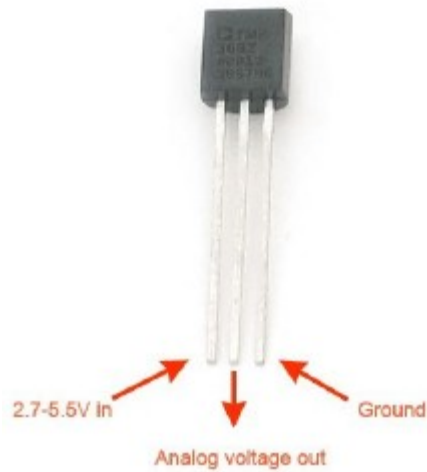
```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  float r = ottieniResistenza(PIN_TERMOMETRO);  
  Serial.print("Resistenza: ");  
  Serial.println(r);  
  r = ottieniTemperatura(r);  
  Serial.print("Temperatura: ");  
  Serial.println(r);  
  Serial.print(""); // Riga vuota  
  delay(500);  
}
```



Sensori integrati

Temperatura *TMP36*

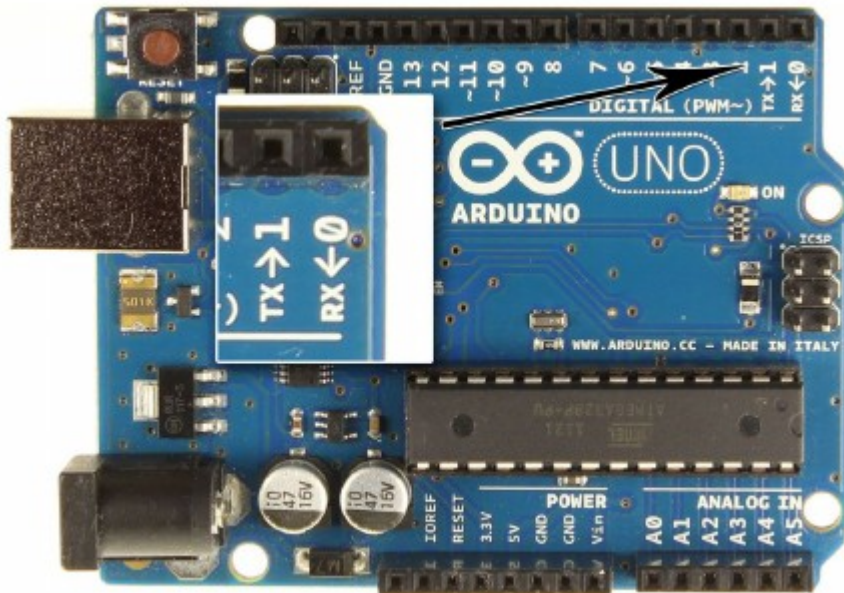


Accelerometro



Distanza

Comunicazione seriale



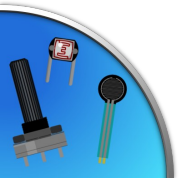
```
Serial.begin(9600);
```

```
Serial.available();
```

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

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

```
Serial.println(c);
```





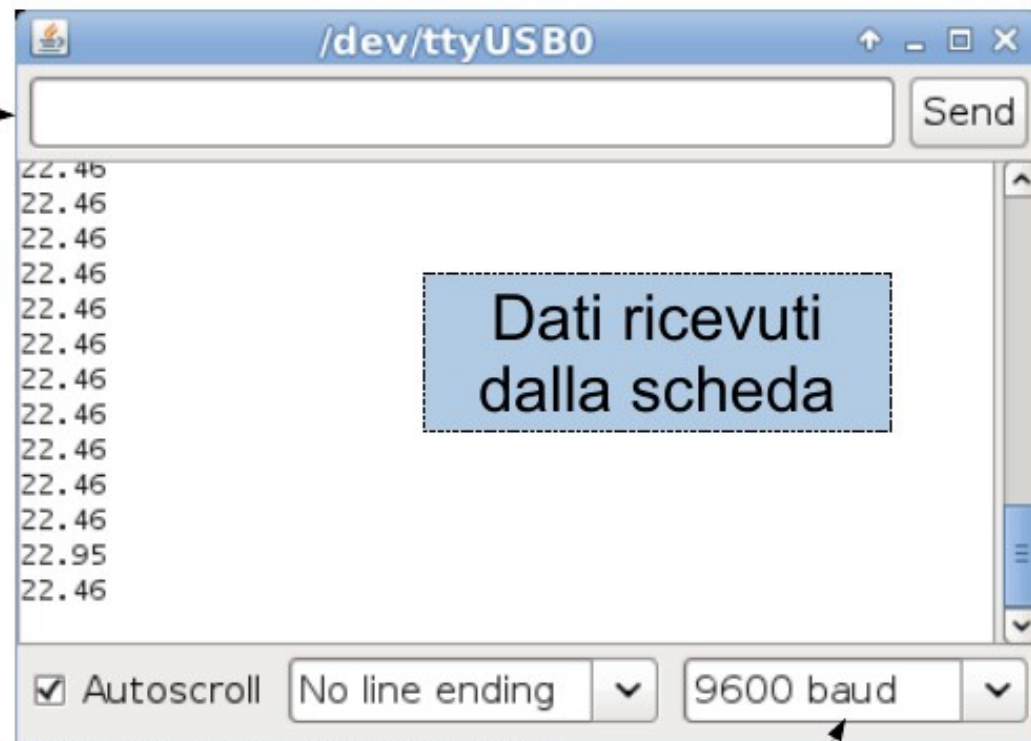
Monitor Seriale

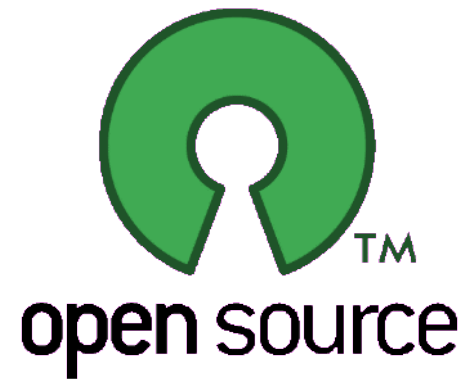
Bottone per aprire il serial monitor

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*

