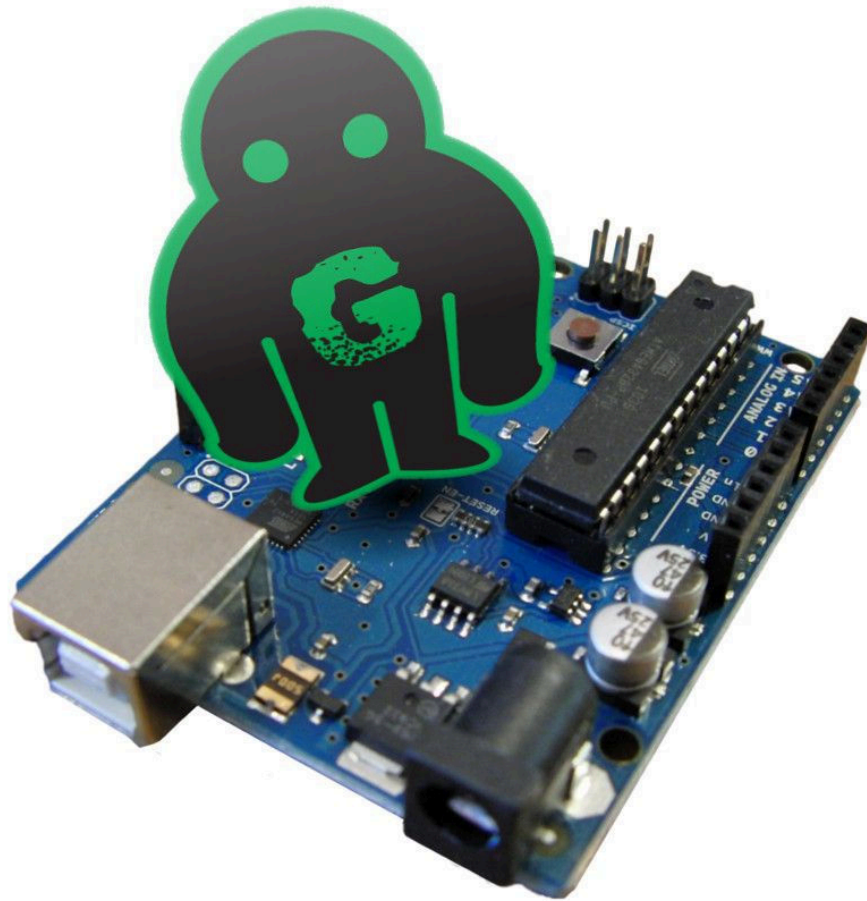


CORSO ARDUINO

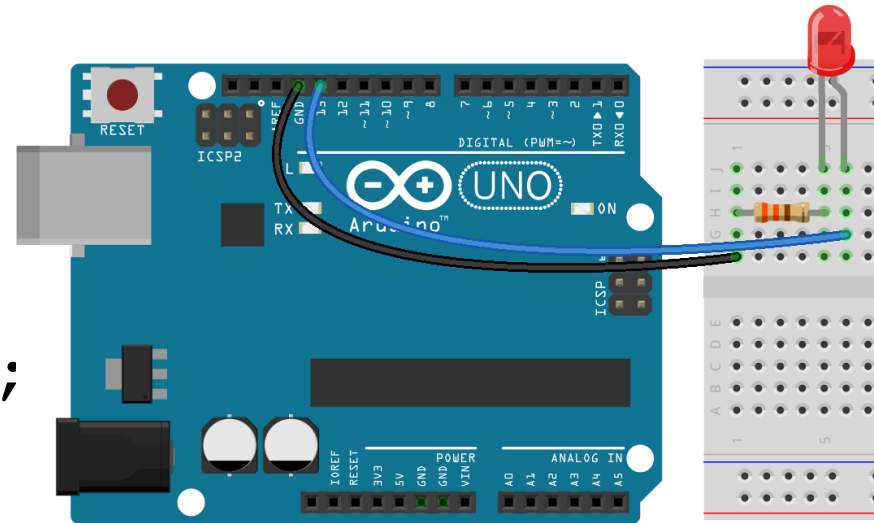


Jacopo Belli
Giulio Fieramosca
Luca Mattii
GOLEM 2016



Far lampeggiare un led

```
const byte LED = 13;  
void setup() {  
  pinMode(LED, OUTPUT);  
}  
void loop() {  
  digitalWrite(LED, HIGH);  
  delay(500);  
  digitalWrite(LED, LOW);  
  delay(500);  
}
```



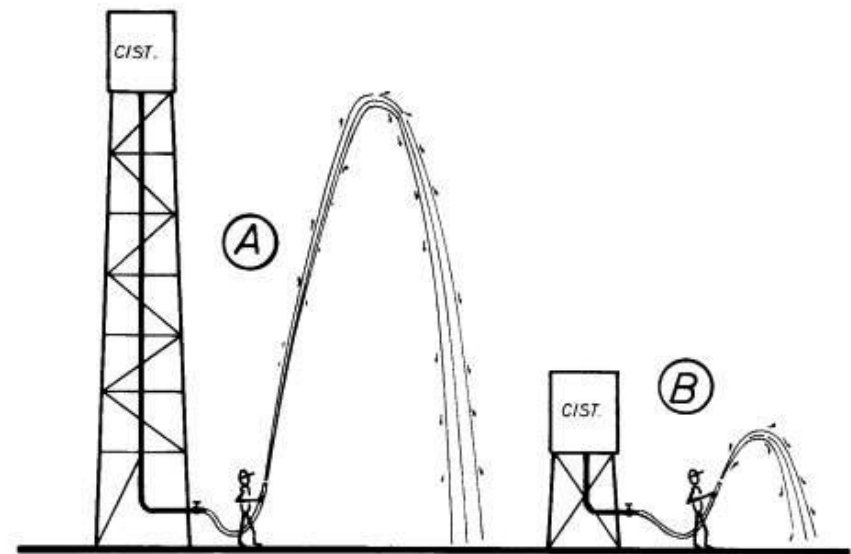
Elettronica di base

Differenza di potenziale (o tensione):

- Spinge gli elettroni a muoversi
- Si misura in Volt [V]

Intensità di Corrente:

- Flusso di elettroni, che si muovono da una zona di maggiore potenziale elettrico ad una zona di minore potenziale
- Si misura in Ampere [A]



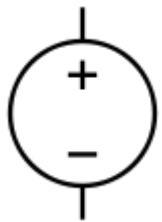
La tensione è come la
pressione che spinge
l'acqua nel tubo



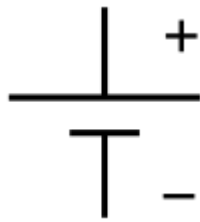
Generatori di tensione

Continua: pila, cella solare, alimentatore

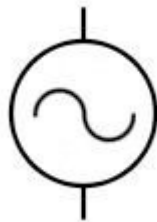
Alternata: alternatore, dinamo della bicicletta



Generatore
tensione
continua



Generatore
tensione
continua
(batteria)



Generatore
tensione
alternata

Simboli circuitali

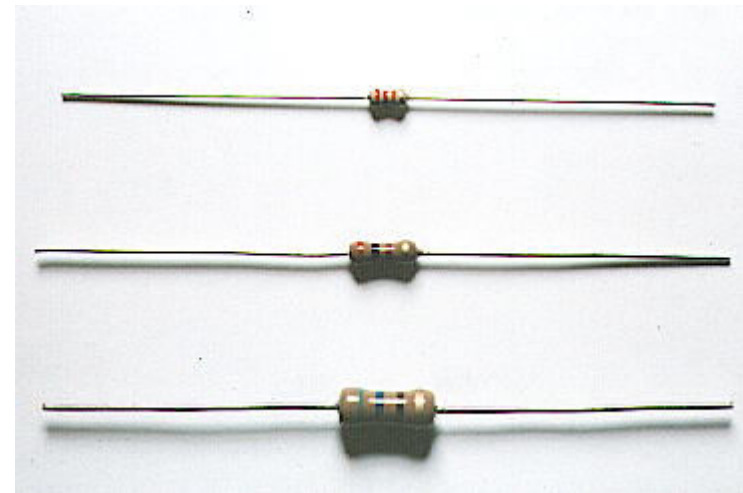


La resistenza

- Si oppone al flusso degli elettroni
- Si misura in ohm [Ω]
- È come un rubinetto che riduce il flusso d'acqua



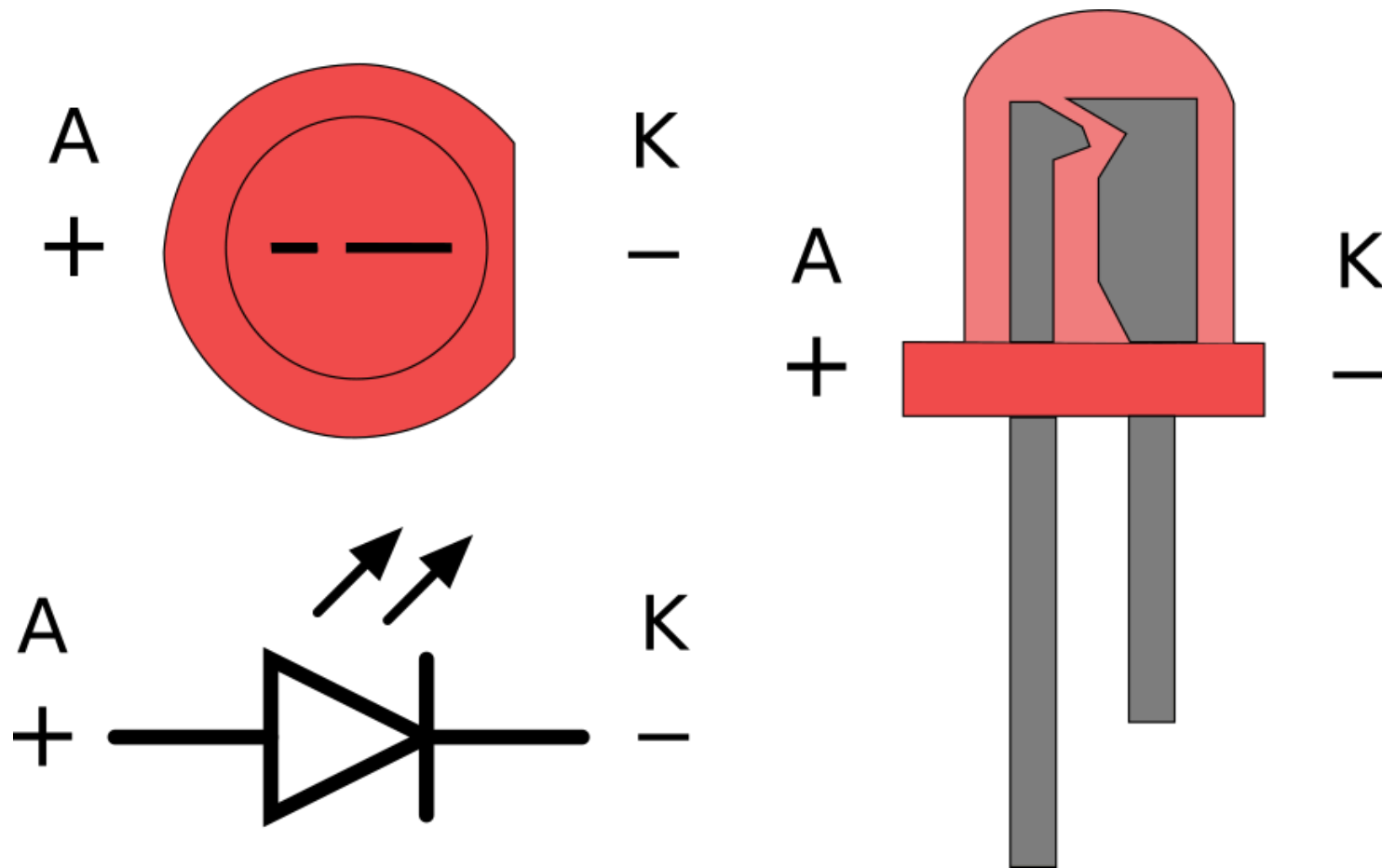
Simbolo su circuito



Alcune resistenze



LED (Light Emitting Diode)



Dimensionare una resistenza

$$V = I \times R$$

Legge di Ohm, lega
tensione (V), corrente (I)
e resistenza (R)



Datasheet

LED Rosso

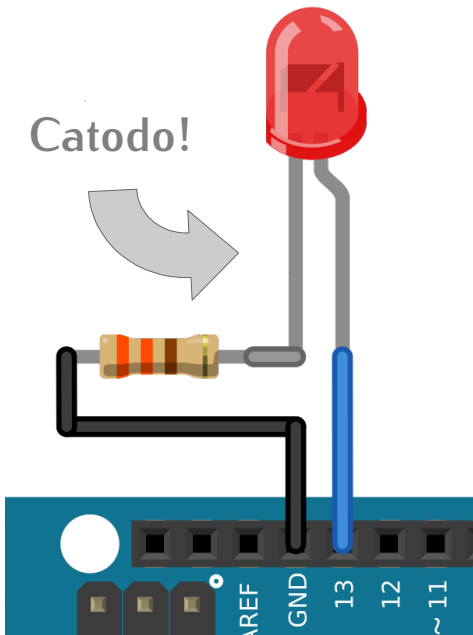
Caduta di tensione: 1,7V

Corrente di lavoro: 10mA

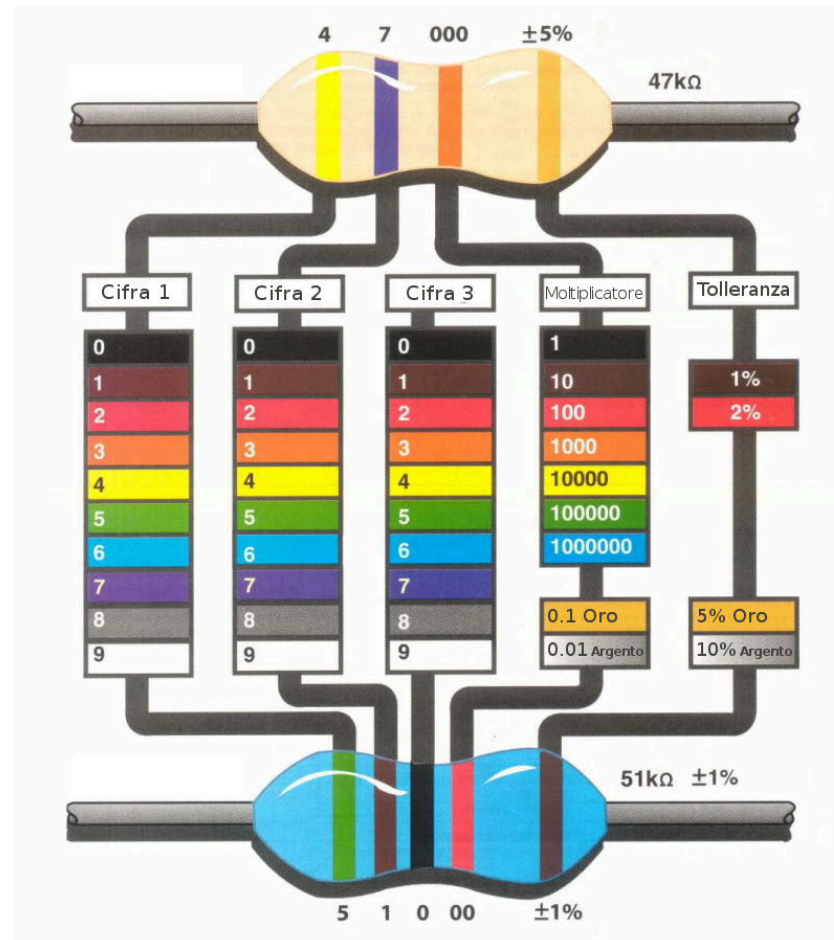
Tensione sulla resistenza = $5 - 1,7 = 3,3 \text{ V}$

$10 \text{ mA} = (10 / 1000) \text{ A} = 0,01 \text{ A}$

$R = V / I = 3,3 / 0,01 = 330 \Omega$



Valori delle resistenze



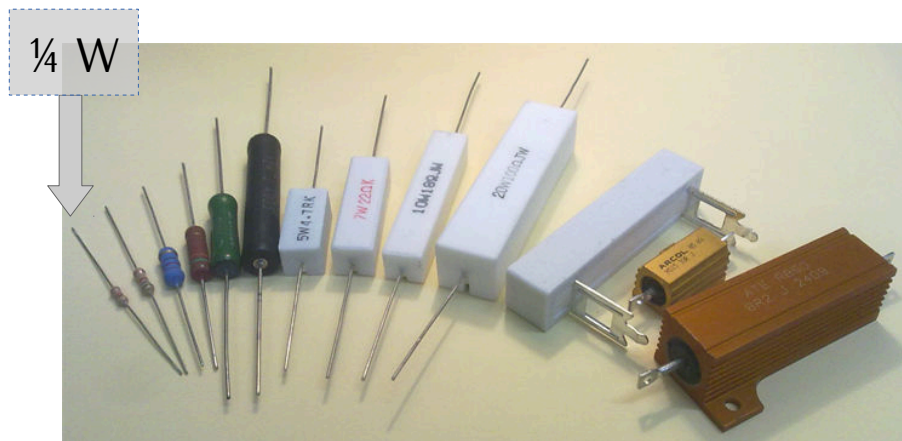
Esempio:



$$33 \times 10 \, \Omega \longrightarrow 330 \, \Omega (\pm 5\%)$$

Approfondimento: Potenza

- Quanto scalda la resistenza (*potenza dissipata*)?
 $P = V \times I = I^2 \times R$ – *Legge di Joule*
- Quanto può scaldare prima di rompersi (*potenza massima*)?
Dipende da che forma ha...



Esempio:

$$I = 10\text{mA}$$

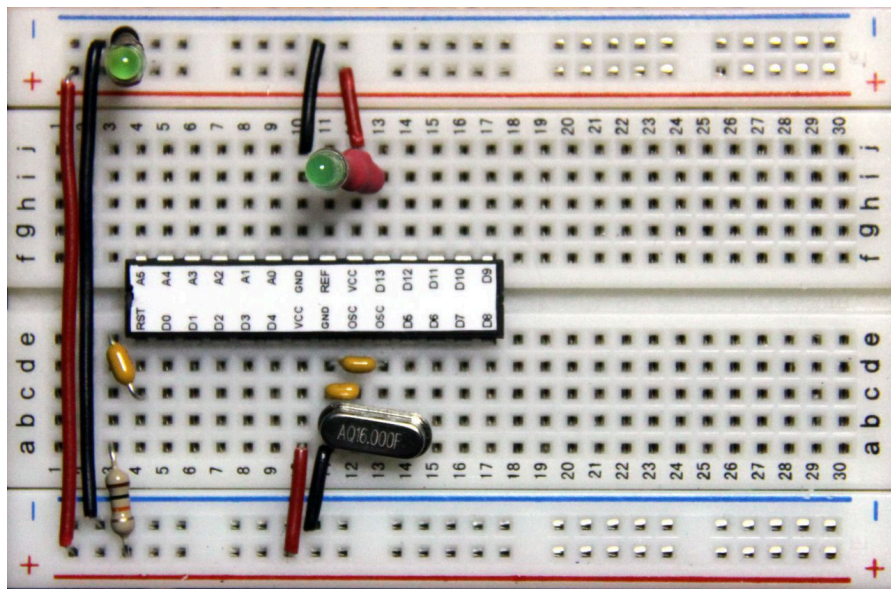
$$R = 330\Omega$$

Quindi...

$$P = (0.01)^2 \times 330 = 0,033 \text{ W}$$

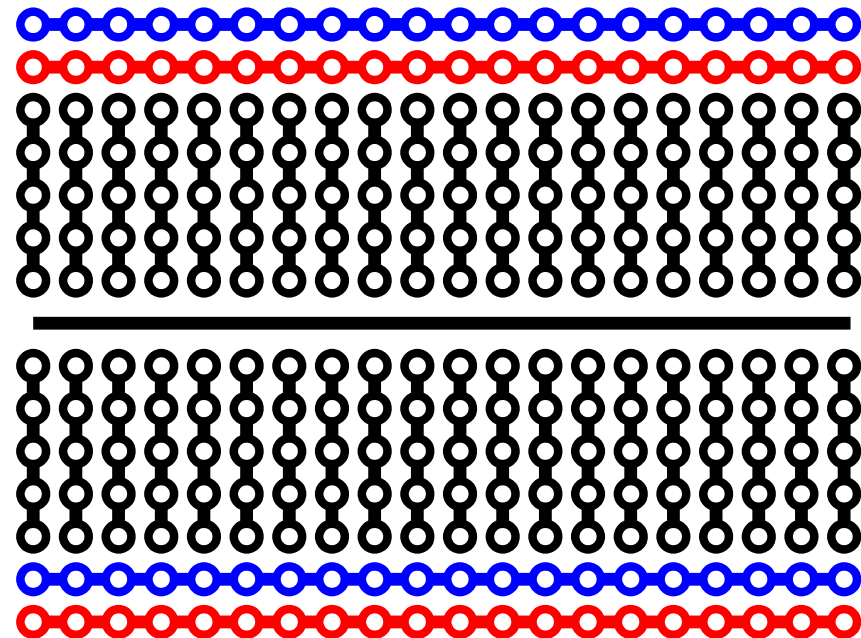
$$\frac{1}{4} \text{ W} = 0,25 \text{ OK!}$$

Collegare più componenti insieme: La breadboard



Alcuni componenti su
una breadboard

Struttura della breadboard



Alcuni componenti:

Condensatori



Accumula e rilascia energia (come un serbatoio)

Si usa per *livellare* le tensioni variabili (stabilizzazione) o in generale per *rallentarle* (filtraggio)



Condensatore



Condensatore
elettrolitico



Condensatore
Variabile



Alcuni componenti: Sensori ad interruttore

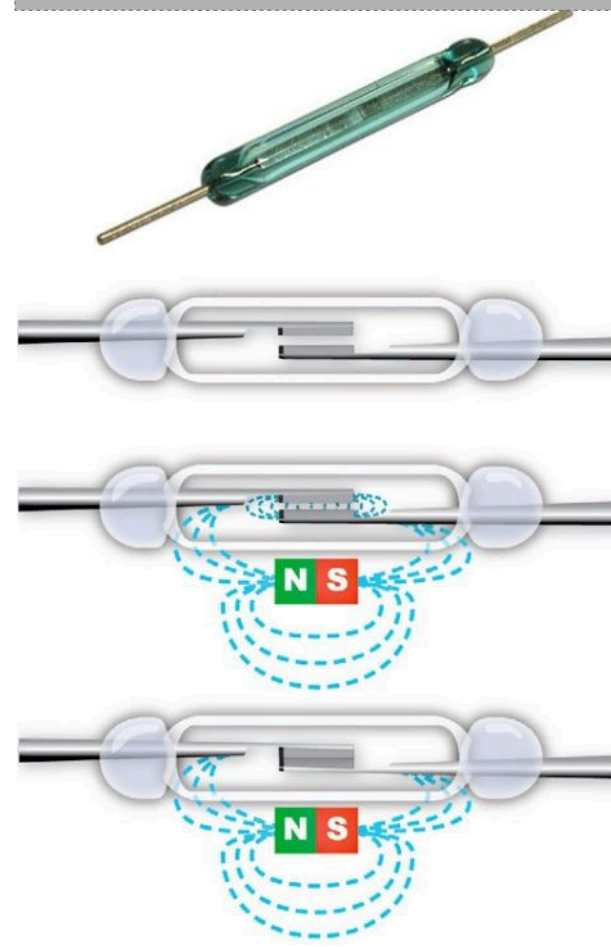


Pushbutton, switch e finecorsa



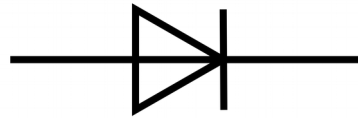
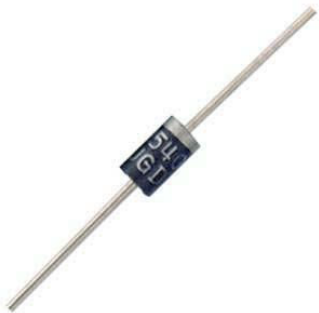
Tilt, l'accelerometro dei poveri

Reed, interruttore
magnetico



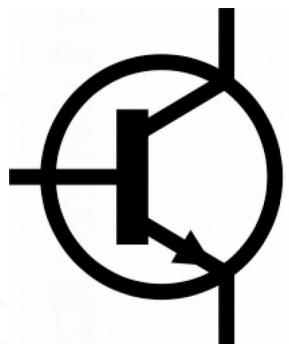
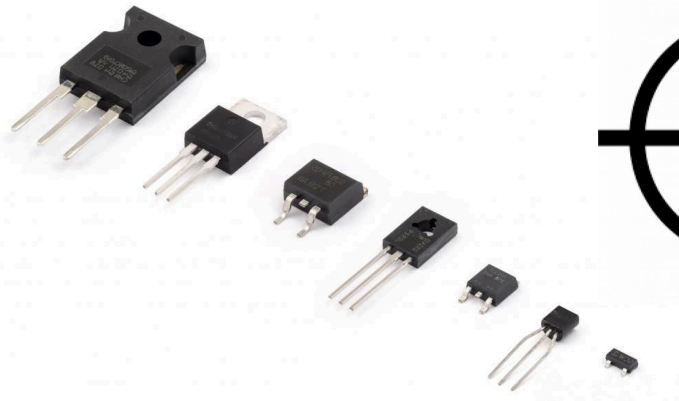
Alcuni componenti:

Diodi e transistor



Diodo

- La corrente può scorrervi in un solo verso;
- Il LED è un tipo particolare di diodo.



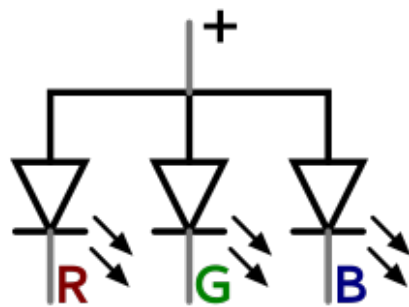
Transistor

Utilizzati come *interruttori elettronici* o *amplificatori*.

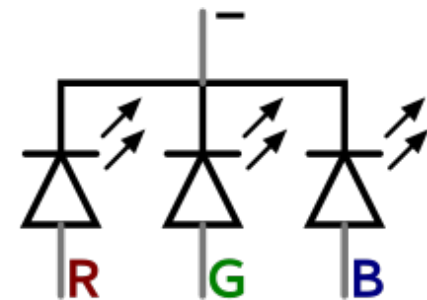


Alcuni componenti:

LED RGB



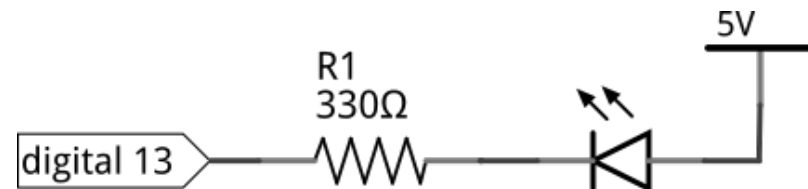
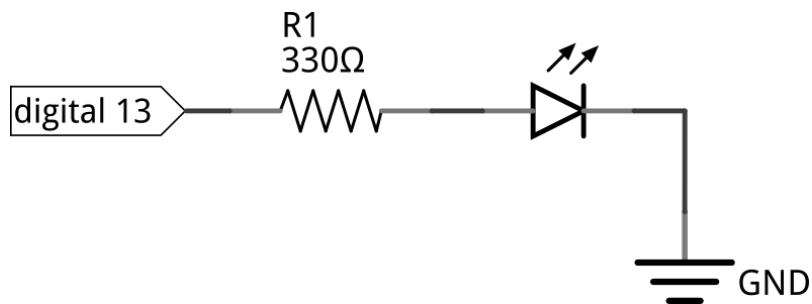
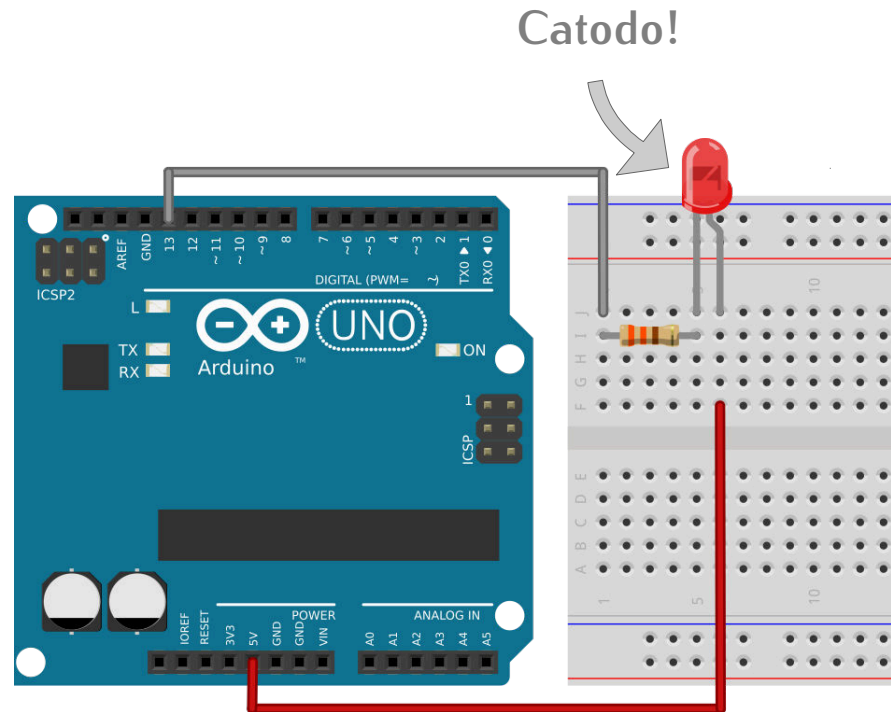
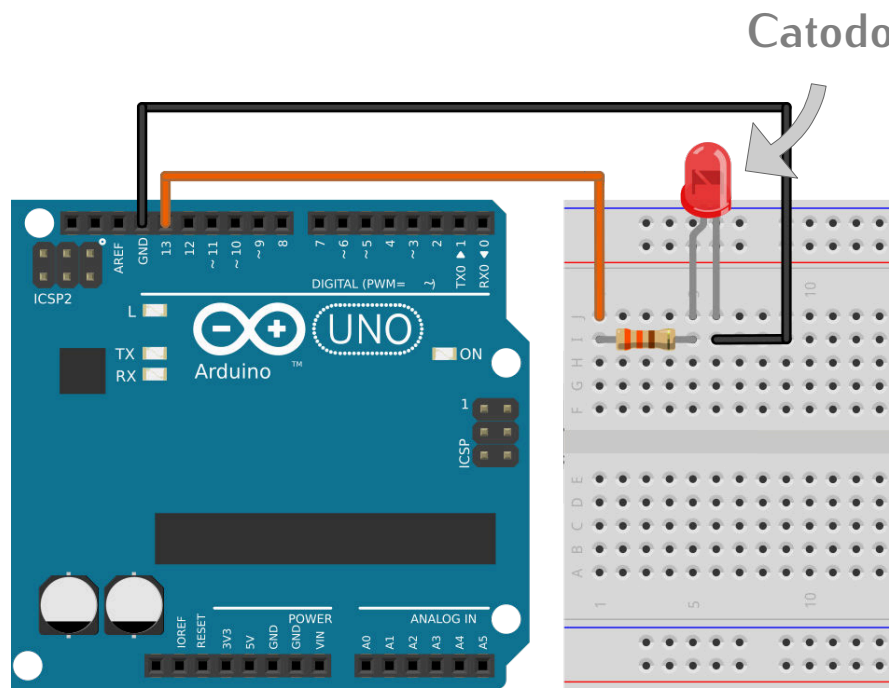
Anodo comune



Catodo comune

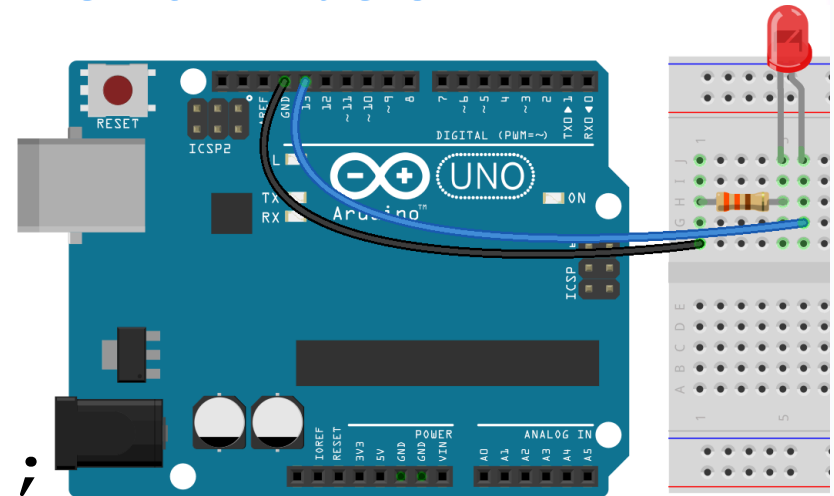


Nota: connessione inversa



Far lampeggiare un led

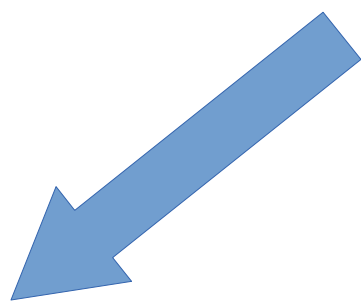
```
const byte PINLED = 13;
void setup() {
  pinMode(PINLED, OUTPUT);
}
void loop() {
  digitalWrite(PINLED, HIGH);
  delay(500);
  digitalWrite(PINLED, LOW);
  delay(500);
}
```



Esercizio 1: Lampeggio Easy

```
const byte PINLED = 13;
void setup() {
    pinMode(PINLED, OUTPUT);
}

void loop() {
    digitalWrite(PINLED, !digitalRead(PINLED));
    delay(500);
}
```



Commenti

```
// Commento su una riga
/* Commento
 * su
 * più
 * righe
 */
```



- Sono righe di codice che vengono **ignorate** dal compilatore;
- Servono per **chiarire** cosa viene fatto nel programma, per questo è bene farne uso, soprattutto in punti dove vengono effettuate operazioni particolarmente **criptiche**.

Variabili

- Contenitori di informazioni, contraddistinti da un'etichetta (**identificatore**) e dal genere di informazione che possono contenere (**tipo**);
- L'**identificatore** può contenere *underscore*, ma **non** può iniziare con un *numero*. Vedremo che anche gli identificatori delle *funzioni* seguono la stessa norma;



Identificatori OK

numeroScarpe

_var

Identificatori ERRATI

2Led

1023

Tipi di variabili

byte

int

long int

unsigned int

unsigned long int

Numeri interi...

positivi e piccoli ($0 \div 255$), occupano 1 byte, appunto;
compresi fra -32,768 e 32,767, occupa 2 byte;
occupando 4 byte arriva fino a $\pm 2,147,483,647$
occupano rispettivamente quanto int e long int,
ma sono solo positivi.

char

È un numero intero, ma viene codificato per farlo corrispondere ad un carattere, secondo la convenzione ASCII. Occupa 1 byte.

float

Numeri con virgola mobile (*fluttuante*), occupano 4 byte;

boolean

Valori di verità, binari (*vero* o *falso*). Occupano comunque 1 byte, poiché questa è la dimensione delle celle di memoria su Arduino.



- I numeri con virgola si indicano col **punto** (1.2 e non 1,2);
- Conviene utilizzare il tipo float solo se serve davvero, perché le operazioni su tali numeri sono più lente, essendo più complessi da gestire.

Dichiarazione e inizializzazione

```
int a; // Dichiarazione di una variabile intera
float Temperatura = 1.2; // Inizializzazione di una variabile reale
char carattere = '3'; // I caratteri si racchiudono fra gli apici
boolean acceso = true;
```

- **Dichiarando** una variabile, le si riserva un adeguato spazio in memoria;
- **Inizializzando** una variabile le si assegna anche un valore iniziale.



```
const byte LEDPIN = 13;
```

- Una **costante** è una variabile inizializzata e non più modificabile;
- Si crea specificando **const** nella riga di inizializzazione

Costrutto condizionale: if

```
if (condizione) {  
    Blocco da eseguire  
}
```

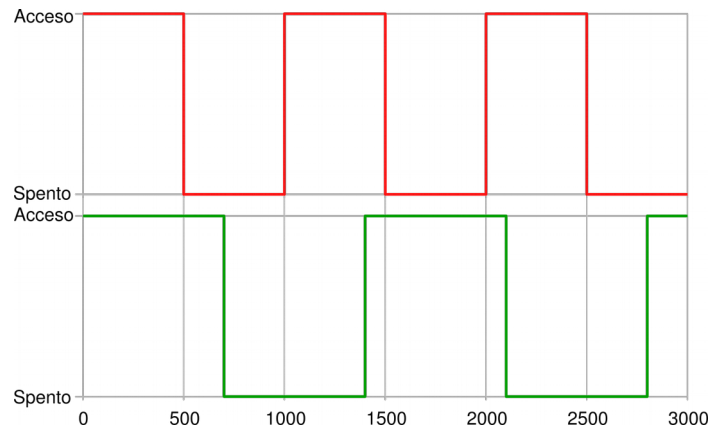
Se la condizione viene verificata, il blocco viene eseguito.
Altrimenti il programma prosegue saltandolo.

La condizione si può esprimere usando variabili e **operatori logici**:

<code>!=</code> diverso	<code>==</code> uguale
<code>></code> maggiore	<code><</code> minore
<code>>=</code> maggiore o uguale	<code><=</code> minore o uguale

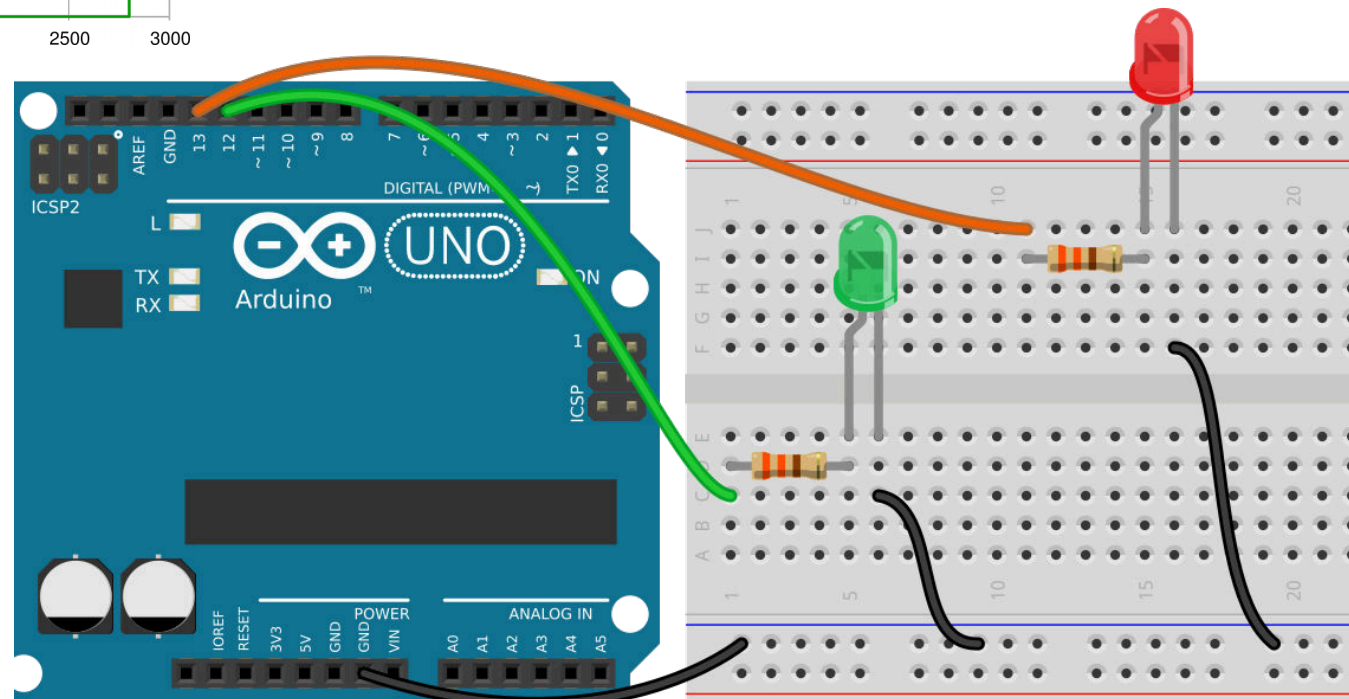


Esercizio 2: Lampeggio asincrono



Led Rosso: 500ms

Led Verde: 700ms



Esercizio 2: Lampeggio asincrono

// Variabili e costanti "globali":
servono in tutto il programma

```
const int TEMPO_ROSSO      = 500;
```

```
const int TEMPO_VERDE     = 700;
```

```
int contatore_rosso       = 0;
```

```
int contatore_verde       = 0;
```

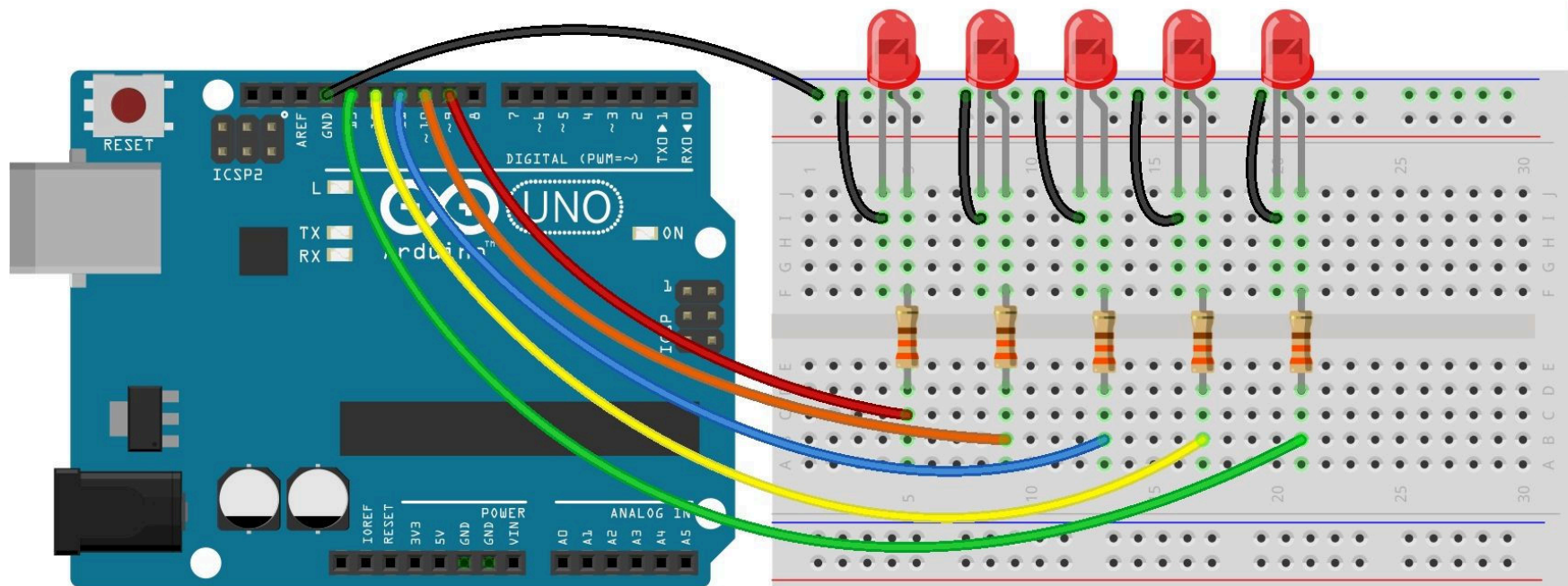


Esercizio 2: Lampeggio asincrono

```
void loop() {  
    delay(1);  
    contatore_rosso++;  
    if (contatore_rosso == TEMPO_ROSSO) {  
        digitalWrite(PIN_ROSSO, !digitalRead(PIN_ROSSO));  
        contatore_rosso = 0;  
    }  
    ...  
}
```



Esercizio 3: Fila di Luci – Board



fritzing

Esercizio 3: Fila di Luci – Sketch

```
// I LED sono collegati ai pin 9, 10, 11, 12, 13
```

```
const byte PIN_PRIMO_LED = 9;
```

```
const byte PIN_ULTIMO_LED = 13;
```

```
byte pin_led_attuale = PIN_PRIMO_LED;
```

```
void setup() {
```

```
    pinMode(PIN_PRIMO_LED, OUTPUT);
```

```
    pinMode(PIN_PRIMO_LED + 1, OUTPUT);
```

```
    pinMode(PIN_PRIMO_LED + 2, OUTPUT);
```

```
    pinMode(PIN_PRIMO_LED + 3, OUTPUT);
```

```
    pinMode(PIN_PRIMO_LED + 4, OUTPUT);
```

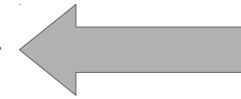
```
    digitalWrite(pin_led_attuale, HIGH);
```

```
    // Per default gli altri pin sono spenti!
```

```
}
```



Costanti importanti



Variabile *contatore*

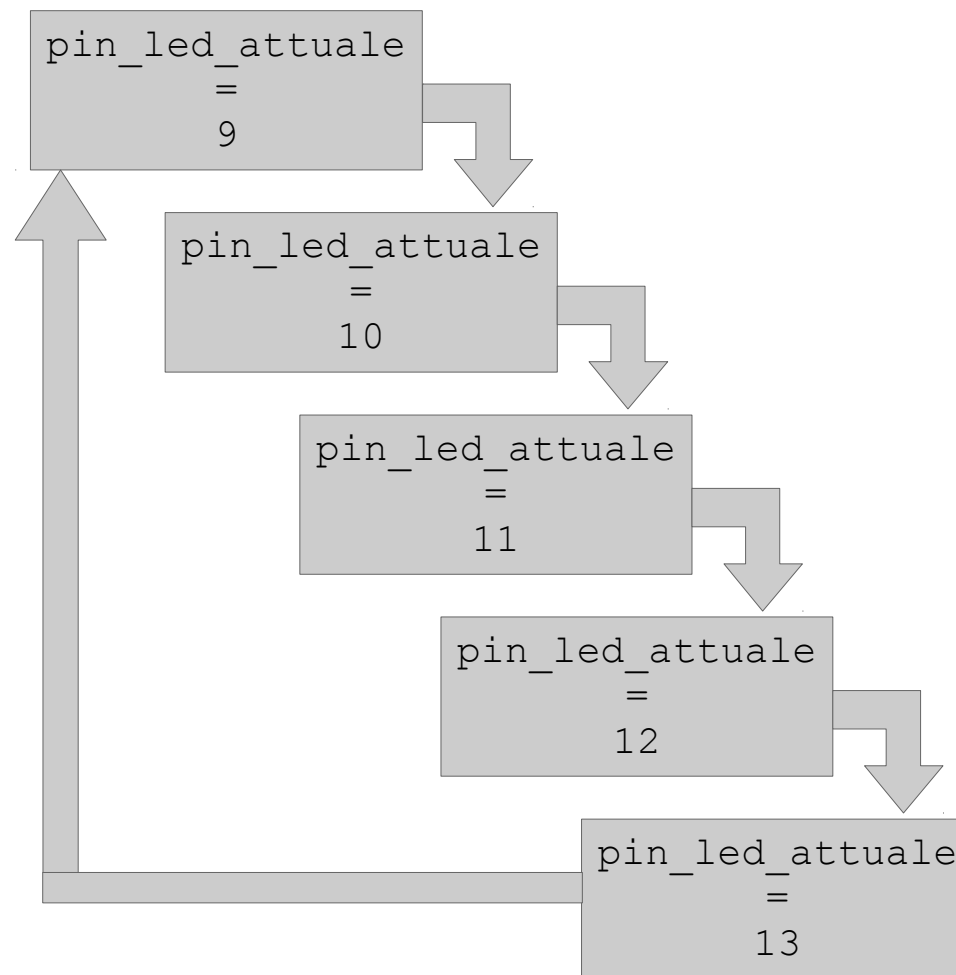


Esercizio 3: Fila di Luci – Sketch

```
void loop() {  
    delay(200);  
    // Spengo il led attualmente acceso (led_attuale)  
    digitalWrite(pin_led_attuale, LOW);  
    // Passo al led successivo  
    pin_led_attuale++;  
    // Ma se il LED successivo non esiste? ricomincio dal primo  
    if ( pin_led_attuale > PIN_ULTIMO_LED ) {  
        pin_led_attuale = PIN_PRIMO_LED;  
    }  
    // Accendo definitivamente il LED "successivo"  
    digitalWrite(pin_led_attuale, HIGH);  
}
```



Esercizio 3: Fila di Luci - Note



Compiti per casa

- **Luci di SuperCar:** accendi 5 led da sinistra a destra e poi da destra a sinistra;
- **RGB:** capire come collegare un LED RGB, quindi farlo lampeggiare un colore alla volta, poi 2 colori alla volta e alla fine tutti e 3 i colori insieme;



Realizzazione di circuiti:

Saldatore a stagno



Stazione saldante



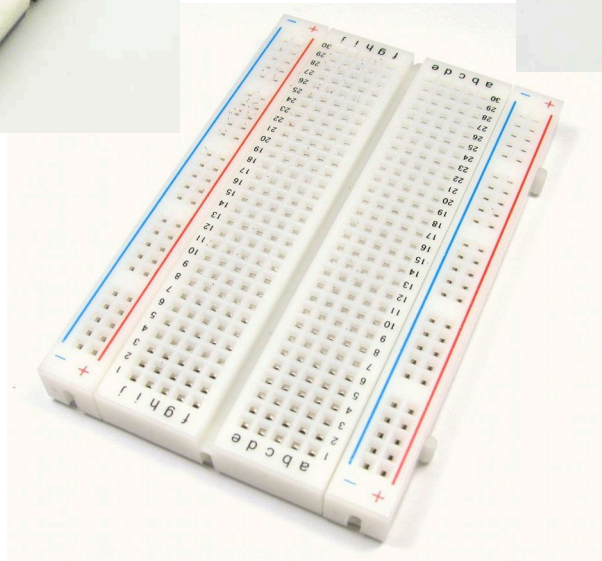
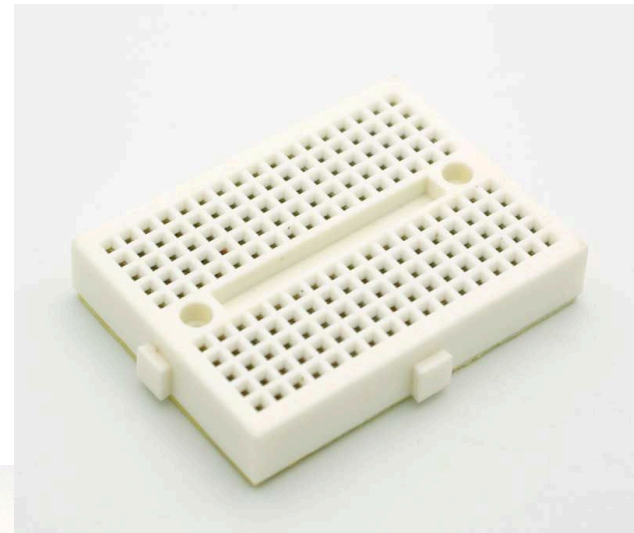
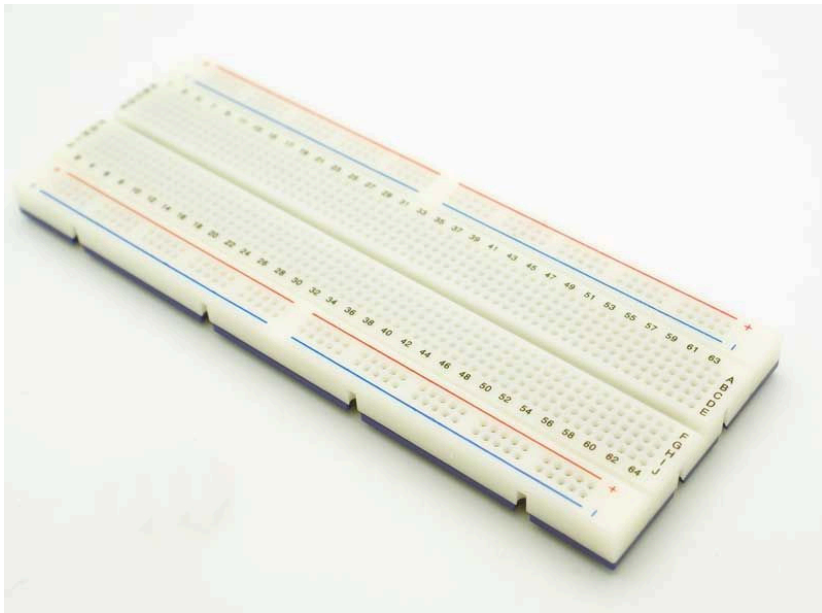
Saldatore a stilo



Stagno
60/40

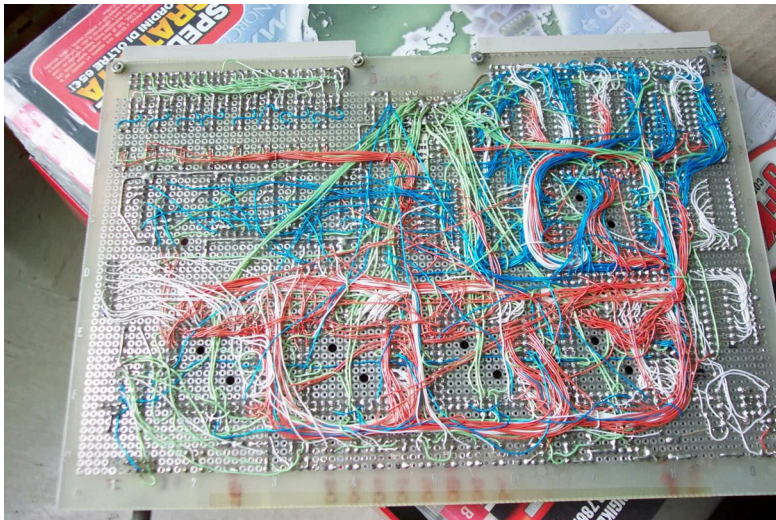
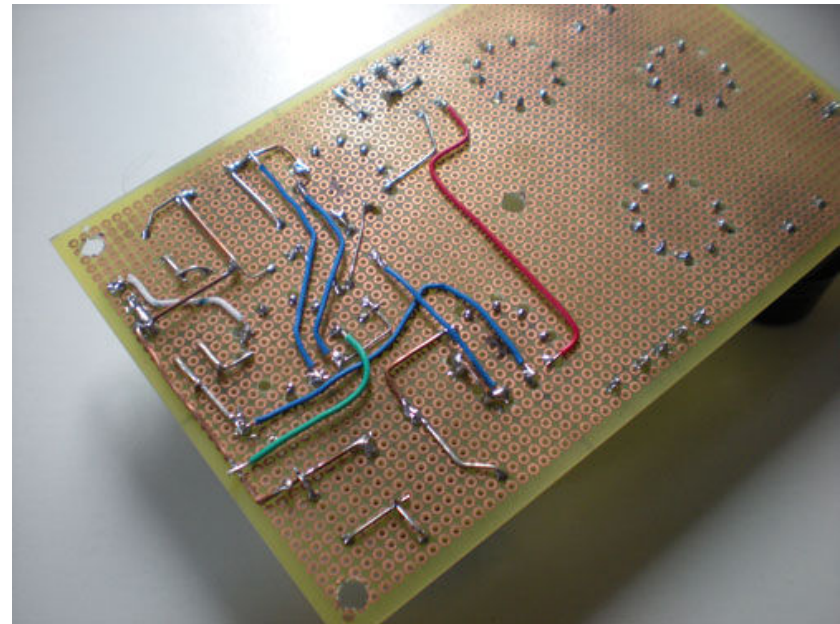
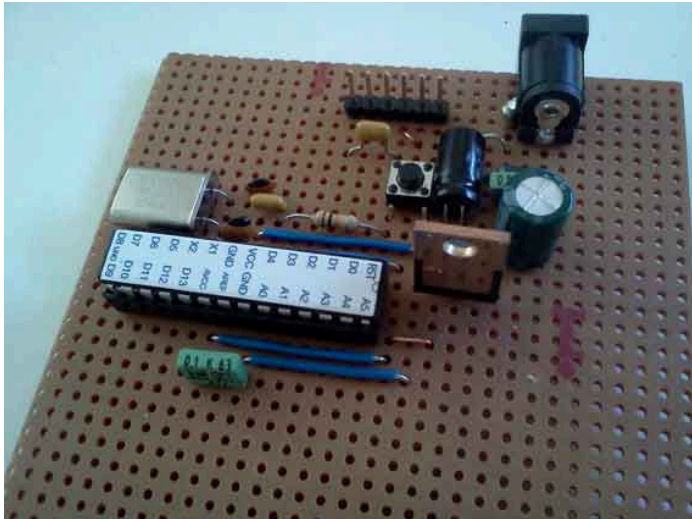


Realizzazione di circuiti: La breadboard



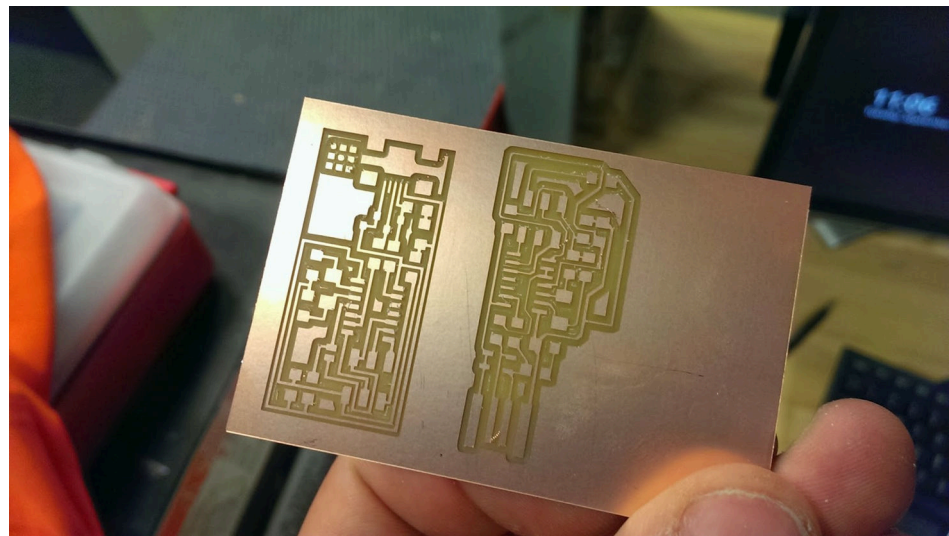
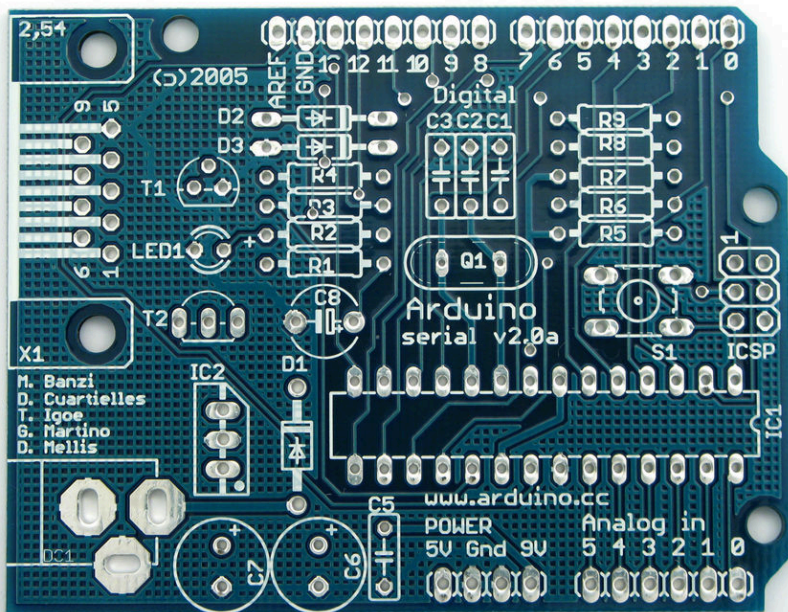
Realizzazione di circuiti: millefori

Arduino assemblato su millefori



Realizzazione di circuiti: PCB

PCB di Arduino Serial



PCB realizzato con una fresa circuiti

Strumentazione: multimetro

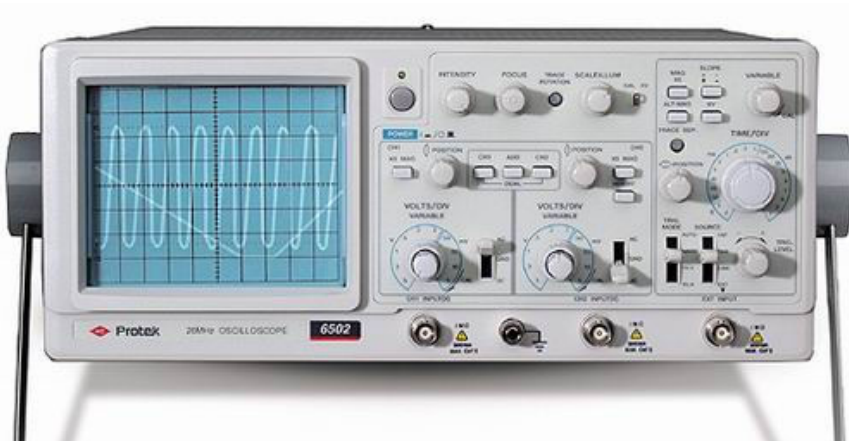


Misurazione di:

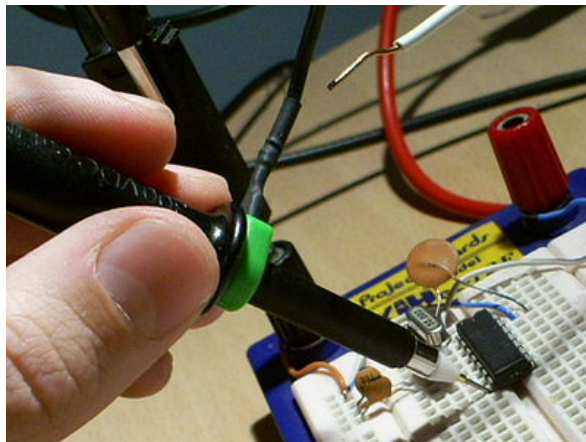
- Tensione
- Corrente
- Resistenza
- Caduta sui diodi e test di conduzione

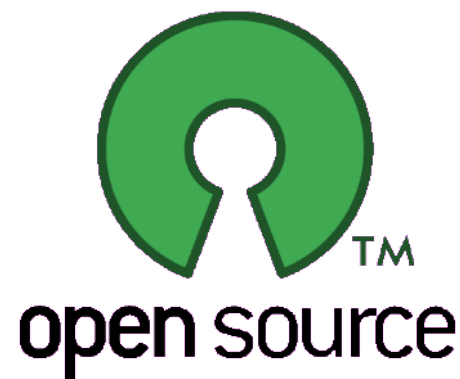
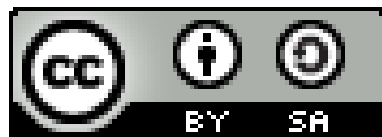


Strumentazione: oscilloscopio



Analisi delle variazioni
periodiche di tensione
nel corso del tempo





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*

