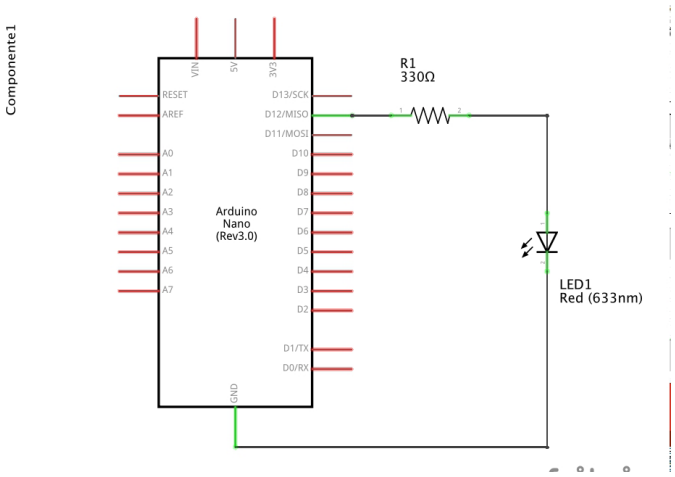
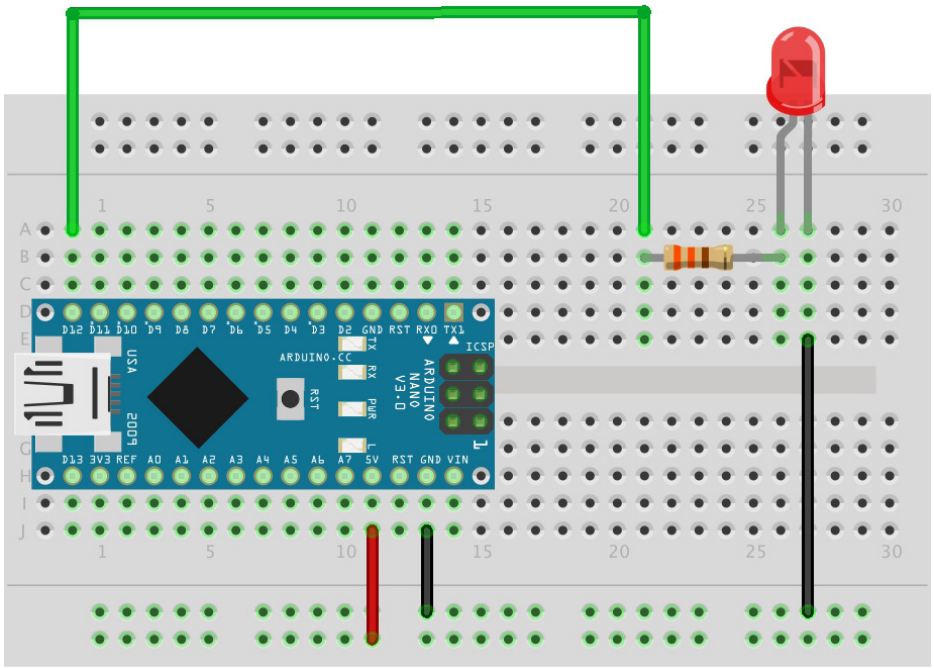
2. LED Esterno

Led Esterno





Aldo Pizzeghella - *www.aldopi.it - aldopi62@gmail.com*

Led Esterno

_02_led_esterno §

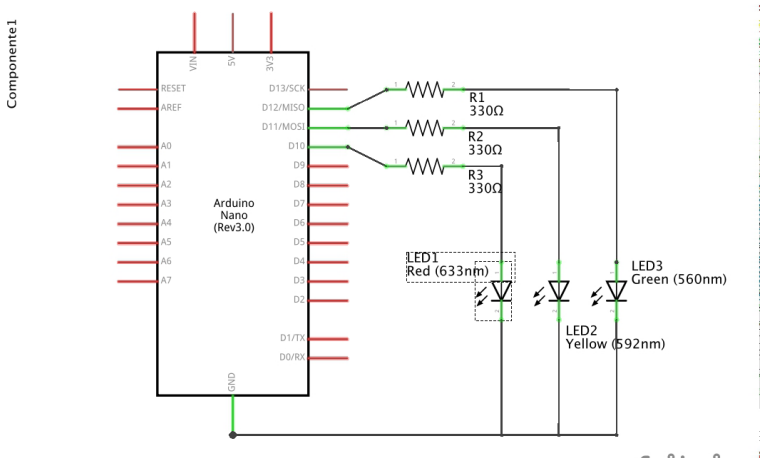
```
1  void setup() {
2    pinMode(12, OUTPUT);
3  }
4
5
6  void loop() {
7    digitalWrite(12, HIGH);
8    delay(1000);
9    digitalWrite(12, LOW);
10   delay(250);
11   digitalWrite(12, HIGH);
12   delay(250);
13   digitalWrite(12, LOW);
14   delay(250);
15   digitalWrite(12, HIGH);
16   delay(1000);
17   digitalWrite(12, LOW);
18   delay(1000);
19 }
```

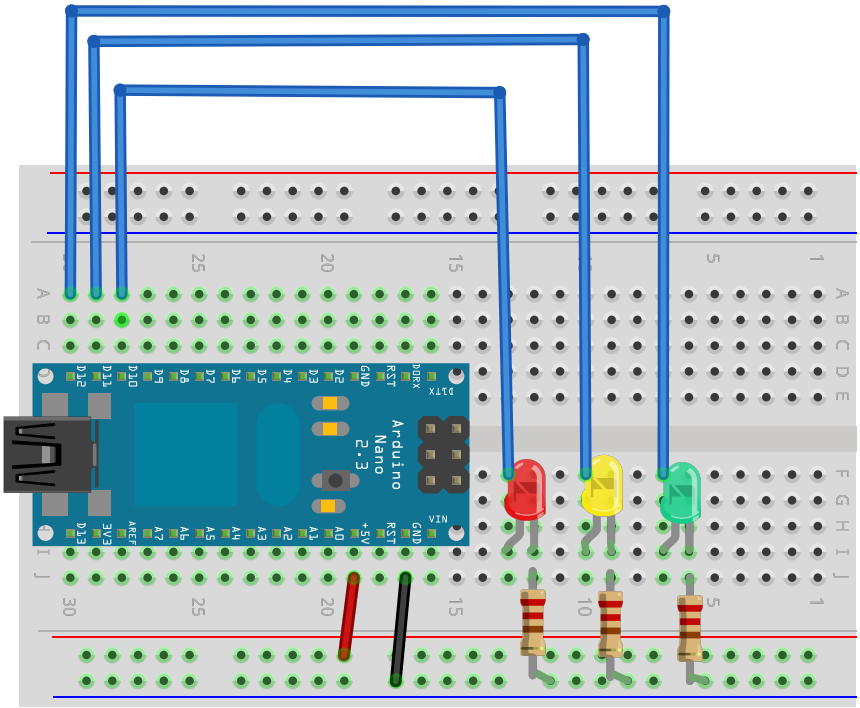
pinMode(pin);

digitalWrite(pin, stato);

delay(time);

Tre Leds


Componente1
Arduino
Nano
(Rev3.0)
R1
330Ω
R2
330Ω
R3
330Ω
LED1
Red (633nm)
LED2
Yellow (592nm)
LED3
Green (560nm)



Semaforo

a_04.1_tre_leds_-_semaforo §

```
/* ******************
 *  corso Arduino Base
 *
 */

#define ledrosso  12
#define ledgiallo 11
#define ledverde  10
#define tempo   3000

void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(ledgiallo, OUTPUT);
  pinMode(10, OUTPUT);
}


void loop() {
  digitalWrite(ledrosso, HIGH);
  delay(tempo);
  digitalWrite(ledrosso, LOW);
  digitalWrite(ledverde, HIGH);
  delay(tempo);
  digitalWrite(ledgiallo, HIGH);
  delay(tempo);
  digitalWrite(ledgiallo, LOW);
  digitalWrite(ledverde, LOW);
}
```

#define costante valore

Lampeggi

a_04.2_tre_leds_-_semaforo §

```
/* ******************
 *  corso Arduino Base
 *  by www.aldopi.it
*/

#define ledrosso  12
#define ledgiallo 11
#define ledverde  10
#define tempo   3000

int i = 0 ;  // dichiarazione variabile numero intero 2 byte 0..65535

void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(ledgiallo, OUTPUT);
  pinMode(10, OUTPUT);
}    // fine setup

void loop() {
  i = 0;
  while (i <10) {                // lampeggio led rosso 10 volte
     digitalWrite(ledrosso, HIGH);
     delay(tempo);
     digitalWrite(ledrosso, LOW);
     delay(tempo);
     i = i+1;
  }
    i = 0;
  while (i <5) {                 // lampeggio led verde 5 volte
     digitalWrite(ledverde, HIGH);
     delay(tempo);
     digitalWrite(ledverde, LOW);
     delay(tempo);
     i = i+1;
  }

  for (i=0, i<20, i=i+1) {    // ripeto per 20 volte lampeggio led giallo
     digitalWrite(ledgiallo, HIGH);
     delay(tempo);
     digitalWrite(ledgiallo, LOW);
     delay(tempo);
  }
}    // fine loop
```

```
while (condizione) {
}

for (condizione) {
}
```


Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

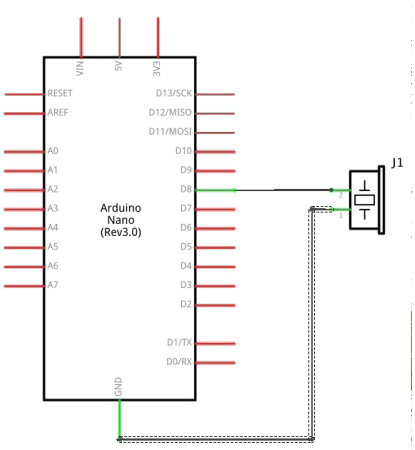
Scorrimento

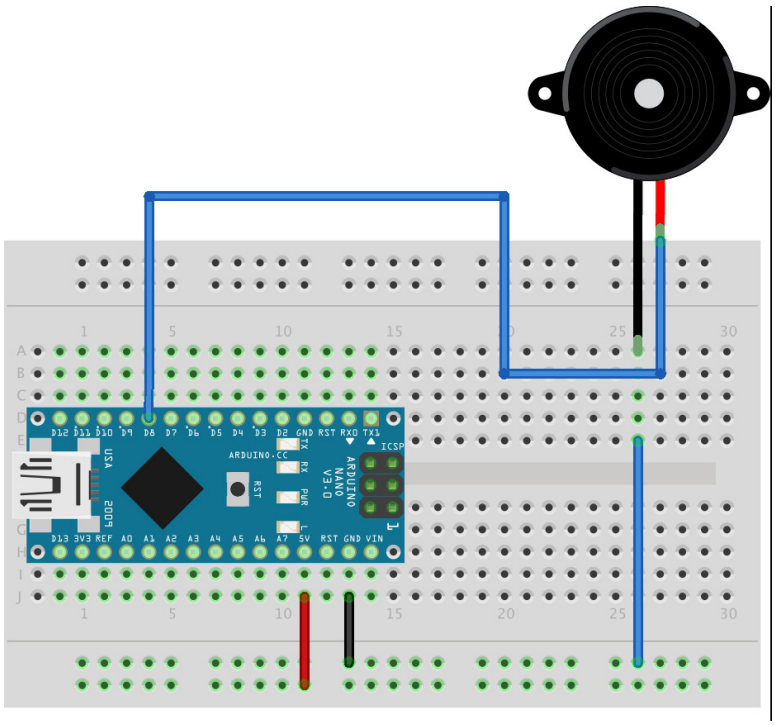
a_04.3_tre_leds_-_scorrimento §

```
/* ********************
 *  corso Arduino Base
 *  by www.aldopi.it
*/

#define ledrosso  12
#define ledgiallo 11
#define ledverde  10
#define tempo   3000

void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(ledgiallo, OUTPUT);
  pinMode(ledverde, OUTPUT);
}    // fine setup

void loop() {
 digitalWrite(ledrosso, HIGH);
  delay(tempo);
  digitalWrite(ledrosso, LOW);
  digitalWrite(ledgiallo, HIGH);
  delay(tempo);
  digitalWrite(ledgiallo, LOW);
  digitalWrite(ledverde, HIGH);
  delay(tempo);
  digitalWrite(ledverde, LOW);
  digitalWrite(ledgiallo, HIGH);
  delay(tempo);
  digitalWrite(ledgiallo, LOW);
}    // fine loop
```

3. - Buzzer - Avvisatore acustico

Il Buzzer è un piccolo dispositivo elettronico simile ad un altoparlante in grado di emettre un avviso sonoro.
Vi sono due tipi di buzzer:
- **ATTIVO** in grado di generare un suono di avviso autonomamente
- **PASSIVO** emette un suono se alimentato con una frequenza ben precisa
entrambi vengono alimentati con una tensione di 5V.
Lo schema di collegamento è identico per i due tipo.





Aldo Pizzeghella - *www.aldopi.it - aldopi62@gmail.com*

Buzzer Attivo

Questo dispositivo emette un suono di attenzione appena viene alimentato, non è possibile variare la nota emessa ne il volume ma è molto facile impiegarlo, basta collegare il suo pin (+ 5V) ad una porta digitale di Arduino e l'altro pin alla mazza. Quando si porta il pin al valore 1 il buzzer emette il suono.

Esempio 1:

```
void setup() {
    pinMode(8,OUTPUT);
    }
```

gruppo **setup**:
pinMode imposta il PIN_8 in uscita
(è da collegare al buzzer)

```
void loop() {
    digitalWrite(8, HIGT);
    delay(500);
    digitalWrite(8, LOW);
    delay(10000);
    }
```

gruppo **loop**:
porta a 1 (5V) il pin 8, accende il buzzer
pausa di 0,5 secondo
spengo il buzzer
pausa di 10 secondo

Esempio 2:

```
void setup() {
    pinMode(8,OUTPUT);
    }

void loop() {
    digitalWrite(8, HIGT);
    delay(500);
    digitalWrite(8, LOW);
    delay(1000);
    digitalWrite(8, HIGT);
    delay(500);
    digitalWrite(8, LOW);
    delay(1000);
    digitalWrite(8, HIGT);
    delay(500);
    digitalWrite(8, LOW);
    delay(1000);
    digitalWrite(8, HIGT);
    delay(100);
    digitalWrite(8, LOW);
    delay(1000);
    digitalWrite(8, HIGT);
    delay(100);
    digitalWrite(8, LOW);
    delay(1000);
    digitalWrite(8, HIGT);
    delay(100);
    digitalWrite(8, LOW);
    delay(10000);
    }
```

emette tre suoni brevi e tre suoni lunghi

Buzzer Passivo

questo dispositivo emette un suono se alimentato con una tensione alternata.
La nota che emette sarà porpozionale alla frequenza che riceve: più è alta la frequenza più acuto sarà il tono emesso.
In questo caso sarà arduino a emettere la frequenza necessaria.
I comandi da usare sono:

`tone(PIN, nota);` — **tone** fa emettere un suono della frequenza specificata con nota al buzzer collegato al piedino PIN.

`noTone(PIN);` — Interrompe la nota generata

Esempio:

```
void setup() {
  pinMode(8, OUTPUT);
}
```

gruppo **setup**:
pinMode imposta il PIN_8 in uscita
(è da collegare al buzzer)

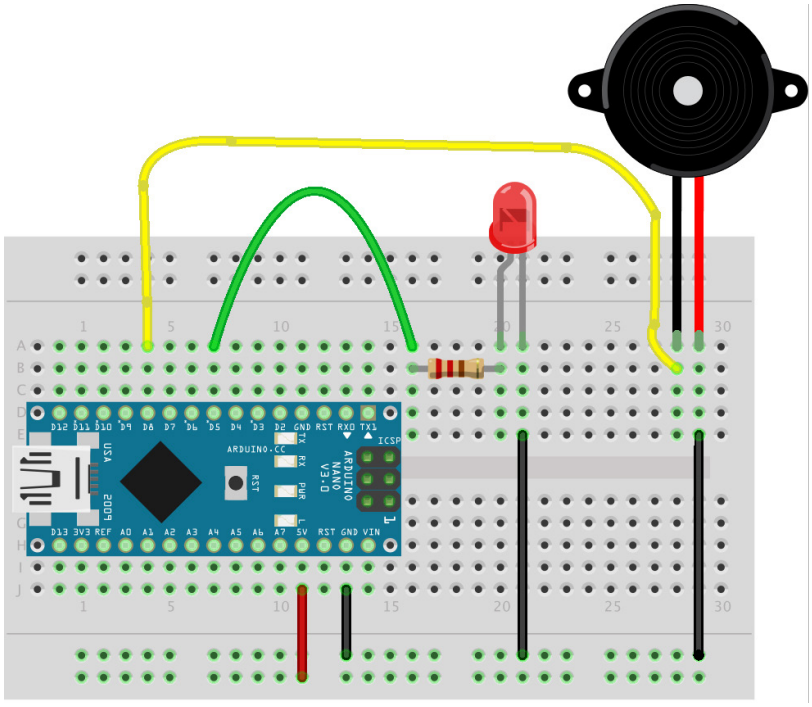
```
void loop() {
  tone(8, 1000);
  delay(500);
  tone(8, 2500);
  delay(1500);
  noTone(8);
  delay(10000);
}
```

gruppo **loop**:
emete una nota grave sul PIN_8
pausa di 0,5 secondi
emette una nota acuta sul PIN_8
pausa di 1,5 secondi
spegne il buzzer
pausa di 10 secondi

Emette:
un beep breve (0.5sec) grave,
un beep acuto lungo (1.5sec) e poi
una pausa lunga (10sec)

Buzzer Passivo + led esterno

colleghiamo il buzzer passivo ed un led



Esempio:

```
void setup() {
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  tone(8, 2000);
  digitalWrite(5,
HIGH);
  delay(500);
  noTone(8);
  digitalWrite(5, LOW);
  delay(5000);
}
```

gruppo **setup**:
buzzer PIN_8
led PIN_5

gruppo **loop**:
emete una nota sul PIN_8
accende il led esterno
pausa di 0.5 secondi
spegne il buzzer
spegne il led
pausa di 5 secondi

esempio utilizzo pulsante e led

pulsante_1 §

```
#define ledrosso 10
#define pulsante 3


void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(pulsante, INPUT_PULLUP);
}


void loop() {
  if (digitalRead(pulsante) == 0) {
    digitalWrite(ledrosso, HIGH);
    delay(200);
    digitalWrite(ledrosso, LOW);
    delay(200);
  }
}
```

pulsante_2 §

```
#define ledrosso 10
#define pulsante 3


void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(pulsante, INPUT_PULLUP);
}


void loop() {
  digitalWrite(ledrosso, HIGH);

  if (digitalRead(pulsante) == 0)  delay(200); else delay(500);

  digitalWrite(ledrosso, LOW);
  delay(200);
}
```


Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

esempio utilizzo pulsante e buzzer

pulsante_3

```
#define ledrosso 10
#define pulsante 3
#define buzzer 6


void setup() {
  pinMode(ledrosso, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(pulsante, INPUT_PULLUP);
}


void loop() {
  if (digitalRead(pulsante) == 0)  {
    digitalWrite(buzzer, HIGH);
    delay(300);
    digitalWrite(buzzer, LOW);
  }
  digitalWrite(ledrosso, HIGH);
  delay(200);
  digitalWrite(ledrosso, LOW);
  delay(200);
}
```

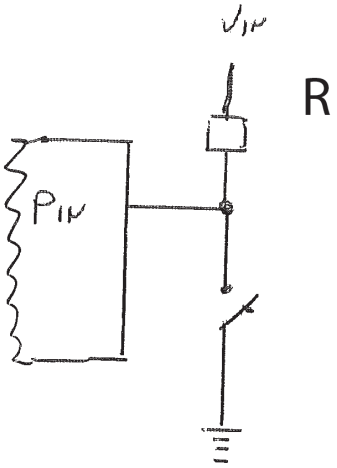
6.a - Pulsante - Seriale

```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura stato pulsante
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: pinMode(pin, INPUT);
 *             digitalRead(pin)      funzione per leggere le porte
 *             Serial                variabile che informa se è aperto il terminale
 *             ciclo attesa terminale seriale
*/

// ------------------------------  dichiarazioni globali
//                                 (le costanti scritte in MAIUSCOLO)
#define PULSANTE 4  //              porta collegata al pulsante
#define LED 13      //              porta collegata al led - interno o esterno


void setup() {  // ---------------  setup
     Serial.begin(9600);
         while (!Serial) {         // attesa che venga aperto il terminale seriale
     }
     Serial.println("test pulsante. Avvio...");
     pinMode(PULSANTE, INPUT);     // configuro pin INPUT per il pulsante
     pinMode(LED, OUTPUT);         // led interno un lampeggio di benventuo
     digitalWrite(LED, HIGH);
     delay(500);
     digitalWrite(LED, LOW);
}  // ============================  setup fine


void loop() {  // ----------------  loop
     Serial.print("pulsante: ");
     Serial.println(digitalRead(PULSANTE));   // invio sul terminale lo stato del pulsante
     delay(250);
}  // ============================  loop fine


// fine programma
```

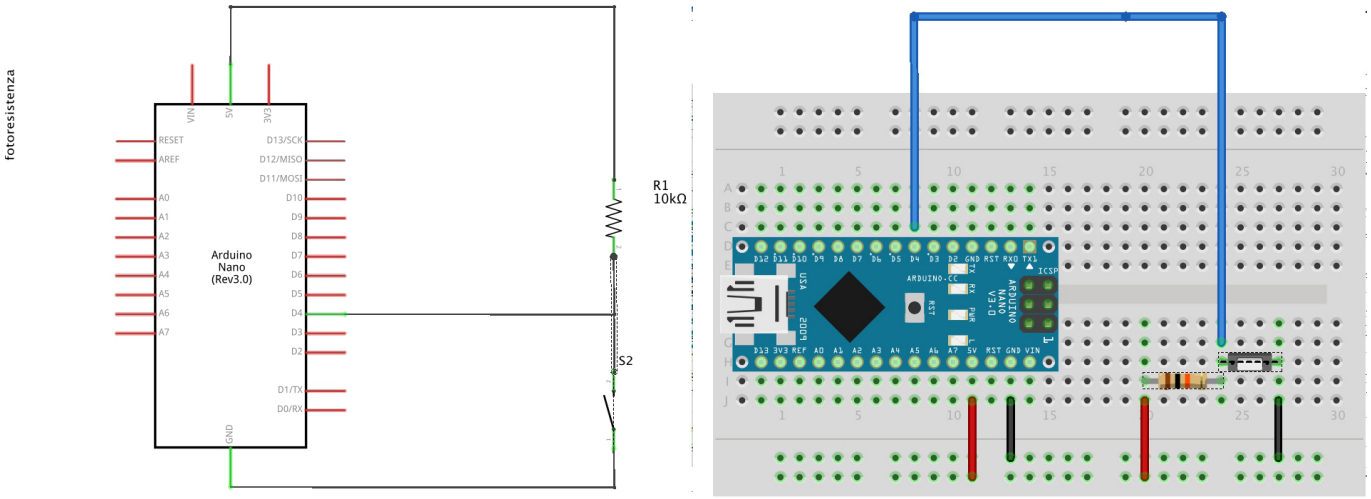

Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

6. - Input digitale - Pulsante

Arduino può leggere lo stato di un pulsante o di un interruttore.
In questo caso di parla di valori digitali: 0 o 1; ON o OFF.
Il circuito necessario è questo:



La resistenza serve per completare il circuito:
quando l'interruttore è aperto Arduino legge il valore Vin (1)
quando l'interruttore è chiuso Arduino legge il valore GND (0).
La resistenza si chiama PULL-UP.
Il valore della resistenze è 10KΩ (marron - nero - arancione - oro).

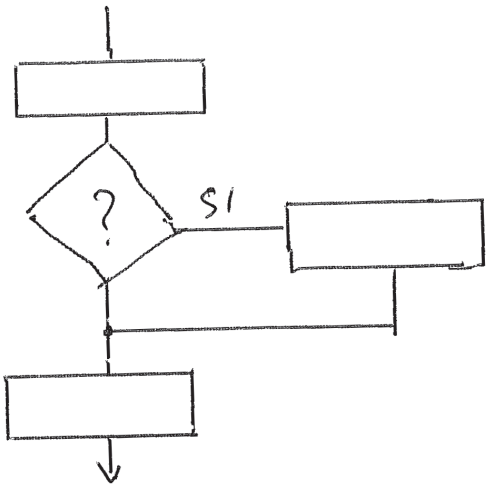


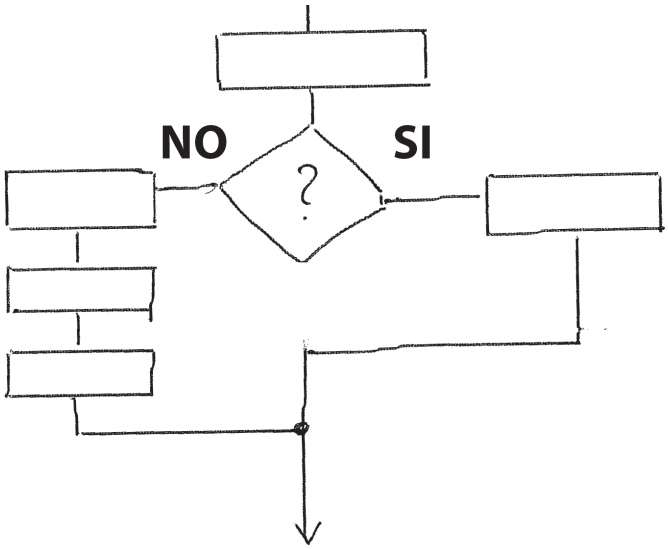
i comandi a disposizione sono:

`pinMode(pin, INPUT);`	prepara il pin (0-12) nella modalità di ingresso, per leggere il pulsante.
`pinMode(pin, INPUT_PULLUP);`	prepara il pin con la resistenza già inserita all'interno di Arduino
`digitalRead(pin);`	Legge il valore e restituisce 0 se il pulsante è chiuso restituise 1 se è aperto

istruzione: if

l'istruzione IF ci permette di far prendere delle decisioni ad Arduino.
Quando di confrontano due valori tra loro il risultato può essere VERO o FALSO
Se il risultato del confronto è VERO verranno eseguite specifiche operazioni.
La rappresentazione grafica di questa istruzione è:





i termini di controno sono:

==	uguale
>	maggiore
>=	maggiore e uguale
<	minore
<=	minore e uguale
!=	non uguale

Esempio:

```
if (bottone == 1) istruzione;
```
verrà eseguita l'istruzione solo se il valore memorizzato in bottore è 1

```
if (numero != 5) istruzione;
```
verrà eseguita l'istruzione solo se il valore memorizzato in numero non è 5

```
if (digitalRead(10) == 1) {
     istruzione1;
     istruzione2;
}
```
verrà letto il valore del **pin** nr 10,
se è collegato a V+ il risultato sarà 1
il contronto sarà **TRUE,** verranno eseguite le istruzioni

6.b - Pulsante + Seriale

```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura stato pulsante
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: dichiarazione LOCALI o GLOBALI
*
*/

// ------------------------------  dichiarazioni globali
//                                 (le costanti scritte in MAIUSCOLO)
#define PULSANTE 4  //              porta collegata al pulsante
#define LED 13      //              porta collegata al led - interno o esterno

int stato = 0;      //              variabile indica stato pulsante


void setup() {  // ---------------  setup
     Serial.begin(9600);
         while (!Serial) {         // attesa che venga aperto il terminale seriale
     }
     Serial.println("test pulsante. Avvio...");
     pinMode(PULSANTE, INPUT);     // configuro pin INPUT per il pulsante
     pinMode(LED, OUTPUT);         // led interno un lampeggio di benventuo
     digitalWrite(LED, HIGH);
     delay(500);
     digitalWrite(LED, LOW);
}  // ============================  setup fine


void loop() {  // ----------------  loop
     stato = digitalRead(PULSANTE);   // lettura stato pulsante
     Serial.print("pulsante: ");
     if (stato ==0 ) Serial.println("rilasciato");  // verifica stato pulsante
     else            Serial.println("premuto");
     delay(250);
}  // ============================  loop fine


// fine programma

```

6.c - Pulsante + Seriale

```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura stato pulsante
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: IF () ELSE
 *             ciclo attesa rilascio pulsante
*/


// ------------------------------  dichiarazioni globali
#define PULSANTE 4  //              porta collegata al pulsante
#define LED 13      //              porta collegata al led - interno o esterno

int stato = 0;      //              variabile indica stato pulsante


void setup() {  // ---------------  setup
     Serial.begin(9600);
         while (!Serial) {         // attesa che venga aperto il terminale seriale
     }
     Serial.println("test pulsante. Avvio...");
     pinMode(PULSANTE, INPUT);     // configuro pin INPUT per il pulsante
     pinMode(LED, OUTPUT);         // led interno un lampeggio di benventuo
     digitalWrite(LED, HIGH);
     delay(500);
     digitalWrite(LED, LOW);
}  // ============================  setup fine


void loop() {  // ----------------  loop
     stato = digitalRead(PULSANTE);   // lettura stato pulsante
     Serial.print("pulsante: ");
     if (stato == 1 ) Serial.println("rilasciato");  // verifica stato pulsante
     else             Serial.println("premuto");

     while (!digitalRead(PULSANTE)) {  // ciclo a vuoto in attesa che venga rilasciato il
          delay(100);
     }
     delay(250);
}  // ============================  loop fine

// fine programma
```


Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

6.d - Pulsante e Leds

```
/****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura stato pulsante accensione led
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 ****************************************
*/

/* istruzioni:
*/

// ------------------------------  dichiarazioni globali
#define PULSANTE 4       //         porta collegata al pulsante
#define LEDBUILT 13      //         porta collegata al led - interno o esterno
#define LEDVERDE 12
int key = 0;

void setup() {  // ---------------  setup
     pinMode(PULSANTE, INPUT);     // configuro pin INPUT per il pulsante
     pinMode(LEDVERDE, OUTPUT);
     pinMode(LEDBUILT, OUTPUT);    // led interno un lampeggio di benventuo
     digitalWrite(LEDBUILT, HIGH);
     delay(500);
     digitalWrite(LEDBUILT, LOW);
}  // ============================  setup fine

void loop() {  // ----------------  loop
     key = digitalRead(PULSANTE);   // lettura stato pulsante
     if (key == 0 )  digitalWrite(LEDVERDE, HIGH);
     else            digitalWrite(LEDVERDE, LOW);

     delay(250);

}  // ============================  loop fine

// fine programma
```

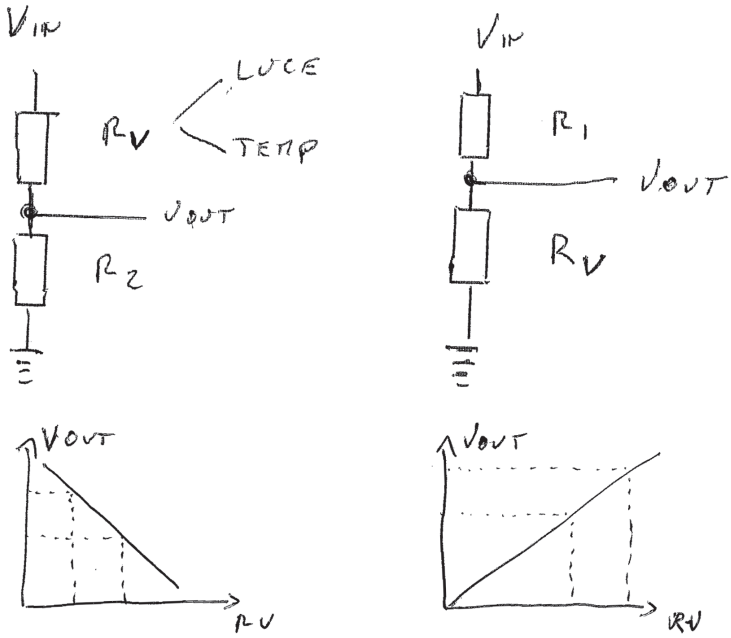
6.e - Pulsante - Leds

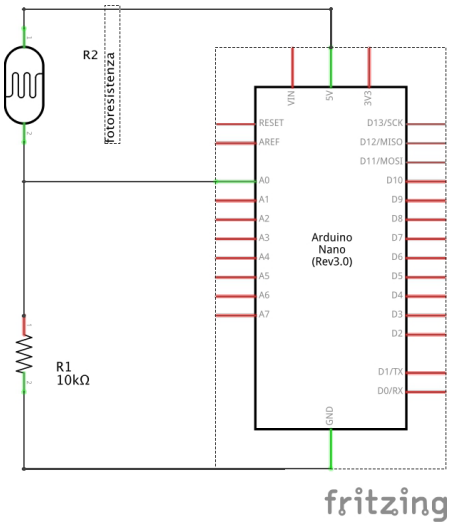
```
switch (variabile) {
        case x:
        case y:
        default:
}

break;
```

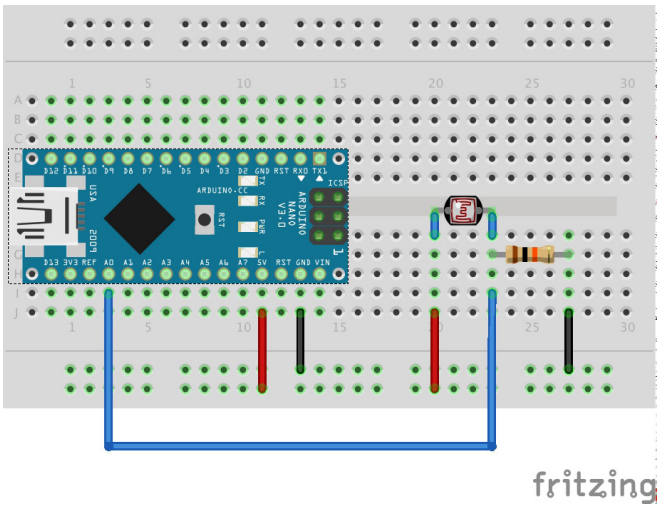
```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura stato pulsante accensione le
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: switch - case
*/

// ------------------------------  dichiarazioni globali
#define PULSANTE 4        //         porta collegata al pulsante
#define LEDBUILT 13       //         porta collegata al led - interno o esterno
#define LEDROSSO 11
#define LEDVERDE 12
int stato = 0;            //         variabile indica stato pulsante
int key = 0;

void setup() {  // --------------  setup
     pinMode(PULSANTE, INPUT);     // configuro pin INPUT per il pulsante
     pinMode(LEDROSSO, OUTPUT);
     pinMode(LEDVERDE, OUTPUT);
     pinMode(LEDBUILT, OUTPUT);    // led interno un lampeggio di benventuo
     digitalWrite(LEDBUILT, HIGH);
     delay(500);
     digitalWrite(LEDBUILT, LOW);
}  // ============================  setup fine

```

```
30
31 void loop() {  // ---------------  loop
32
33      key = digitalRead(PULSANTE);   // lettura stato pulsante
34      if (key == 0 ) {
35           stato = stato + 1; if (stato >3) stato = 0;  // incremento key 0-1-2-3
36      }
37      switch (stato) {
38           case 0:
39                digitalWrite(LEDROSSO, LOW);
40                digitalWrite(LEDVERDE, LOW);
41                break;
42           case 1:
43                digitalWrite(LEDROSSO, HIGH);
44                digitalWrite(LEDVERDE, LOW);
45                break;
46           case 2:
47                digitalWrite(LEDROSSO, LOW);
48                digitalWrite(LEDVERDE, HIGH);
49                break;
50           case 3:
51                digitalWrite(LEDROSSO, HIGH);
52                digitalWrite(LEDVERDE, HIGH);
53                break;
54      }
55
56      while (!digitalRead(PULSANTE)) {   // ciclo vuoto attesa rilascio tasto
57        delay(100);                      // attesa anti rimbalzo
58      }
59      delay(250);
60 }  // ============================  loop fine
61
62 // fine programma
63
```

7.1 - Fotoresistenza







7.a - Fotoresistenza + Seriale

analogRead(pin)

```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura analogica fotoresistenza
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: analogRead(pin);
 *
*/

// ------------------------------  dichiarazioni globali
#define FOTORESISTENZA 0 //         porta analogica A0 collegata alla fotoresistenza
#define LEDBUILT 13      //         porta collegata al led - interno o esterno
int valore = 0;
int numero = 0;

void setup() {  // ---------------  setup
     Serial.begin(9600);
     while (!Serial) {         // attesa che venga aperto il terminale seriale
     }
     Serial.println("test lettura Fotoresistenza. Avvio...");

     pinMode(LEDBUILT, OUTPUT);    // led interno un lampeggio di benventuo
     digitalWrite(LEDBUILT, HIGH);
     delay(500);
     digitalWrite(LEDBUILT, LOW);
}  // ============================  setup fine

void loop() {  // ----------------  loop

     valore = analogRead(FOTORESISTENZA);   // lettura stato pulsante
                                            // valore tra 0 e 1023
     numero = valore /4;
     Serial.print(valore);
     Serial.print("\t");
     Serial.println(numero);

     delay(500);
}  // ============================  loop fine

// fine programma
```

7.b - Fotoresistenza + Led

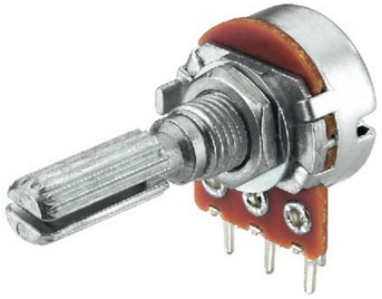
```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: lettura analogica fotoresistenza
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: analogRead(pin);
 *
*/

// ------------------------------  dichiarazioni globali
#define FOTORESISTENZA 0 //         porta analogica A0 collegata alla fotoresistenza
#define LEDRED 12        //         porta collegata al led esterno
#define LEDBUILT 13
int valore = 0;
int numero = 0;

void setup() {  // ---------------  setup
     pinMode(LEDBUILT, OUTPUT);    // led interno un lampeggio di benventuo
     digitalWrite(LEDBUILT, HIGH);
     delay(500);
     digitalWrite(LEDBUILT, LOW);
     pinMode(LEDRED, OUTPUT);
}  // ============================  setup fine

void loop() {  // ----------------  loop

     valore = analogRead(FOTORESISTENZA);   // lettura stato pulsante
                                            // valore tra 0 e 1023
     numero = valore *2;                    // calcolo tempo lampeggio led
     digitalWrite(LEDRED, HIGH);
     delay(numero);
     digitalWrite(LEDRED, LOW);
     delay(numero);
}  // ============================  loop fine

// fine programma
```

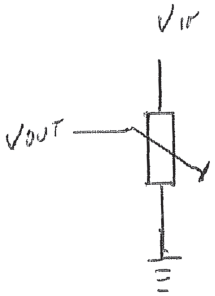

Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

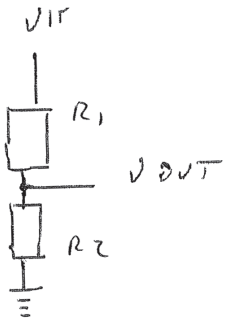
7.c - Fotoresistenza + Buzzer

```
/*****************************************
 *    progetto: corso BASE Arduino
 * descrizione: fotoresistenza buzzer
 *      autore: www.AldoPi.it
 *        data: 3 novembre 2016
 *    versione: 1.0
 *****************************************
*/

/* istruzioni: analogRead(pin);
 *
*/

// ------------------------------  dichiarazioni globali
#define FOTORESISTENZA 0 //         porta analogica A0 collegata alla fotoresistenza
#define BUZZER 4         //         porta collegata al BUZZER
#define LEDBUILT 13
int valore = 0;
int numero = 0;

void setup() {  // ---------------  setup
     pinMode(LEDBUILT, OUTPUT);    // led interno un lampeggio di benventuo
     digitalWrite(LEDBUILT, HIGH);
     delay(500);
     digitalWrite(LEDBUILT, LOW);
     pinMode(LEDRED, OUTPUT);
}  // ============================  setup fine

void loop() {  // ----------------  loop

     valore = analogRead(FOTORESISTENZA);   // lettura stato pulsante
                                            // valore tra 0 e 1023
     numero = valore *3;                    // calcolo tempo frequenza suono
     tone(BUZZER, numero);
     delay(250);

}  // ============================  loop fine

// fine programma
```

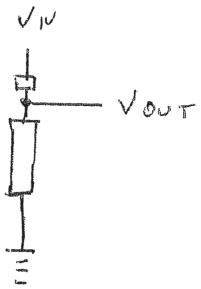
7.2 - Potenziometro

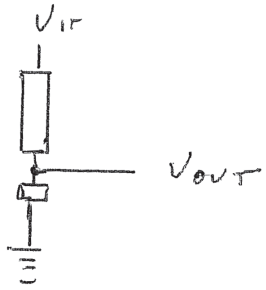


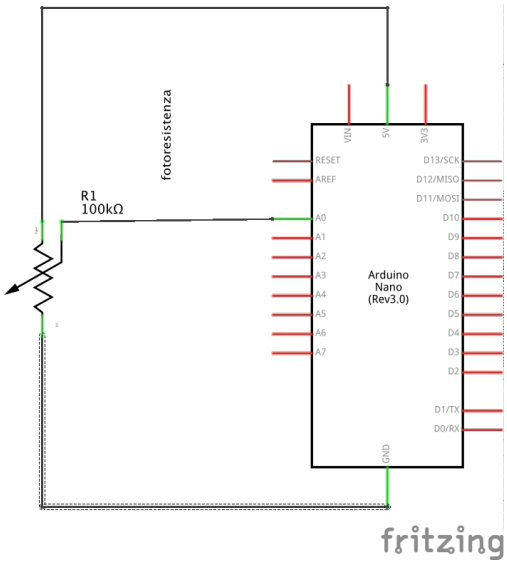


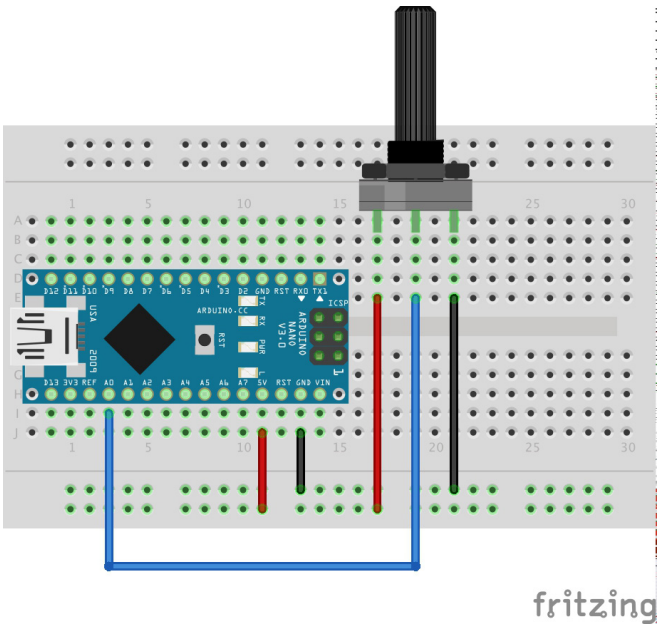


potenziometro - 10Kohm - Rtot = R1 + R2 = 10Kohm









5.a - Output - Monitor Seriale

_05_terminale_seriale §

```
1  void setup() {
2    Serial.begin(9600);
3    Serial.println("uso terminale... avvio");
4    Serial.println("");
5  }
6
7
8  void loop() {
9    Serial.println("numerazione... avvio");
10
11   Serial.print("numero: ");
12   Serial.println("1");
13
14   Serial.print("numero: ");
15   Serial.println("2");
16
17   Serial.print("numero: ");
18   Serial.println("3");
19
20   Serial.print("numero: ");
21   Serial.println("4");
22
23   Serial.print("numero: ");
24   Serial.println("5");
25
26   Serial.println("------------------ fine");
27   Serial.println("");
28   delay(5000);
29
30 }
```

Serial.begin(velocità);

Serial.print(valore);

Serial.println(valore);

5.b - Output - Plotter Seriale

_06_plotter_seriale §

```
1  void setup() {
2    Serial.begin(9600);
3  }
4
5
6  void loop() {
7    Serial.println("numerazione... avvio");
8
9    Serial.println("1");    delay(250);
10   Serial.println("2");    delay(250);
11   Serial.println("3");    delay(250);
12   Serial.println("4");    delay(250);
13   Serial.println("5");    delay(250);
14   delay(2000);
15
16 }
```

_06_plotter_seriale_random §

```
1  void setup() {
2    Serial.begin(9600);
3  }
4
5
6  void loop() {
7        Serial.println("numerazione... avvio");
8        for (int i=0; i<10; i++) {
9          Serial.print(i);
10         Serial.print("\t");
11         Serial.println(round(0, 50));
12       }
13 }
```

Output - Plotter Seriale

```
Serial.begin(Velocità);
```

predispone la porta seriale (pin 0 e 1) per comunicare con l'esterno. es. terminale
velocità esprime la velocità della comiunicazione (9600, ecc)

```
Serial.print(valore);              Serial.println(valore);
Serial.print(valore, formato);     Serial.println(valore, formato);
```

invia il valore specificato in:
valore: valore da trasmetter sul terminale
formato: [facoltativo] come configurare il dato in uscita: HEX, DEC, OTT, BIN

esempio1:

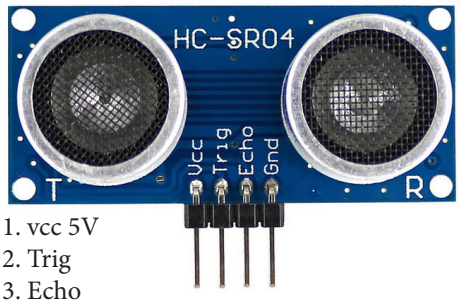
```
void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println("inizio lavoro...");
    delay(1000);
    Serial.print("porta numero 10: ");
    Serial.println(LOW);
    Serial.println("");
    delay(9999);      // attesa lunga
}
```

esempio 2:

```
void setup() {
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(100);
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Metro Elettronico

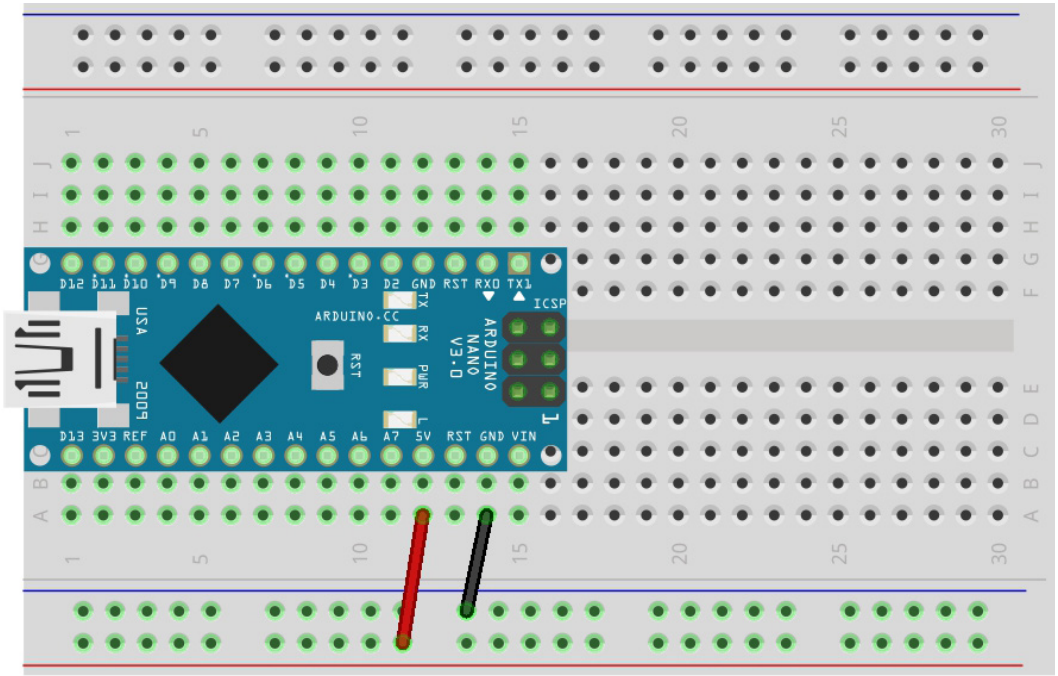
Sensore distanze.
misura la distanza utilizzando gli ultrasuoni, invia un impulso e misura il tempo di ritorno.



1. vcc 5V
2. Trig
3. Echo
4. gnd

Display 8 cifre

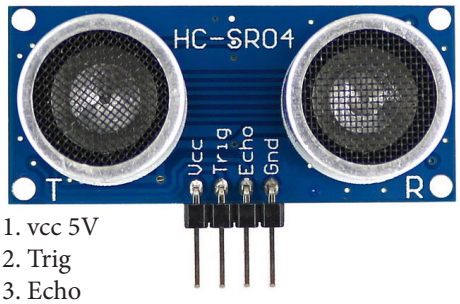
1. vcc 5V
2. Din
3. Load
4. clock
5. gnd



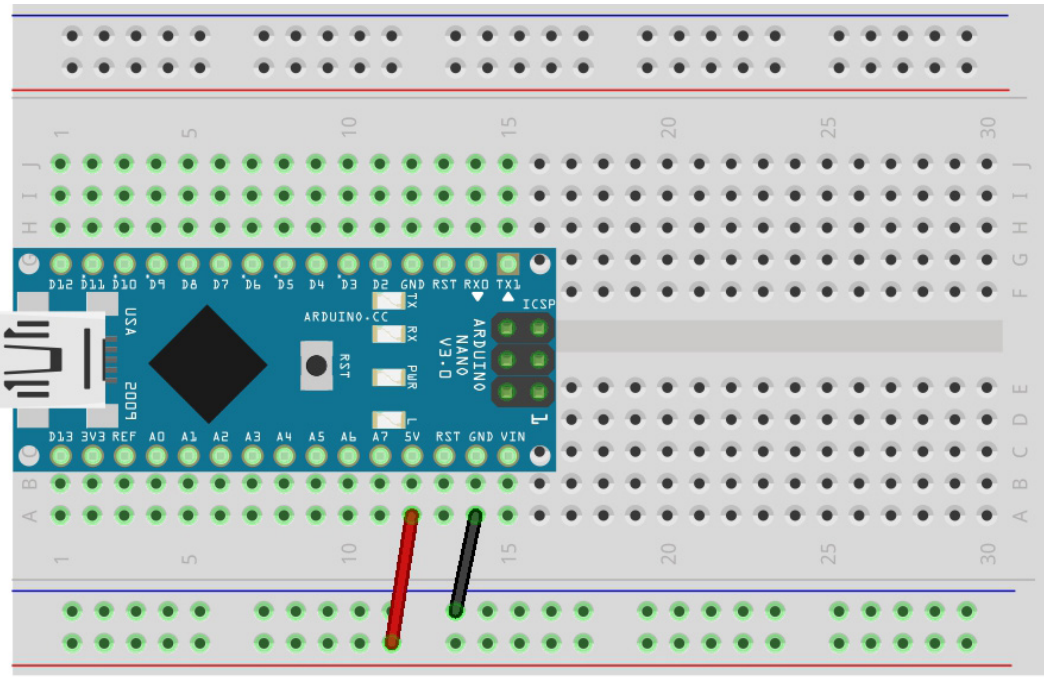
```
#define PIN_START 4
#include "LedControl.h"
#define lc_datain 12
#define lc_clock   10
#define lc_cs              11
LedControl lc = LedControl(lc_datain, lc_clock, lc_cs, 1);

#define trigPin 5
#define echoPin 6

void setup() {
      lc.shutdown(0, false);
      lc.setIntensity(0, 3);
      lc.clearDisplay(0);

      pinMode(trigPin, OUTPUT);
      pinMode(echoPin, INPUT);
      timer = millis() / 10;
      pinMode(PIN_START, INPUT_PULLUP);
}

void loop() {
      word nn = distanza();

      if (nn > 999) lc.setDigit(0, 7, (nn / 1000) % 10, false);
      else lc.setRow(0, 7, 0);
      if (nn > 99) lc.setDigit(0, 6, (nn / 100) % 10, false);
      else lc.setRow(0, 6, 0);
      if (nn > 9) lc.setDigit(0, 5, (nn / 10) % 10, false);
      else lc.setRow(0, 5, 0);
      lc.setDigit(0, 4, (nn / 1) % 10, false);

      delay(250);
}

word distanza() {
      long duration;
      float distance;
      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      digitalWrite(trigPin, HIGH);
      delayMicroseconds(10);
      digitalWrite(trigPin, LOW);
      duration = pulseIn(echoPin, HIGH);
      distance = (duration / 2) / 2.91;
      if (distance < 40) {
            return -1.0;
      } else {
            return distance;
      }
}
```

Sensore di parcheggio

Sensore distanze.
misura la distanza utilizzando gli ultrasuoni, invia un impulso e misura il tempo di ritorno.



1. vcc 5V
2. Trig
3. Echo
4. gnd



Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

```
#define buzzer 4

#define trigPin 5
#define echoPin 6

void setup() {
      pinMode(buzzer, OUTPUT);
      pinMode(trigPin, OUTPUT);
      pinMode(echoPin, INPUT);
}


void loop() {
      digitalWrite(buzzer, 0);
      word nn = distanza();
      if (nn < 50) {
            digitalWrite(buzzer, 1);
            delay(200);
      }
      if (nn < 100) {
            digitalWrite(buzzer, 1);
            delay(100);
      }
      delay(150);
}


word distanza() {
      long duration;
      float distance;
      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      digitalWrite(trigPin, HIGH);
      delayMicroseconds(10);
      digitalWrite(trigPin, LOW);
      duration = pulseIn(echoPin, HIGH);
      distance = (duration / 2) / 2.91;
      if (distance < 40) {
            return 10;
      } else {
            return distance;
      }
}
```

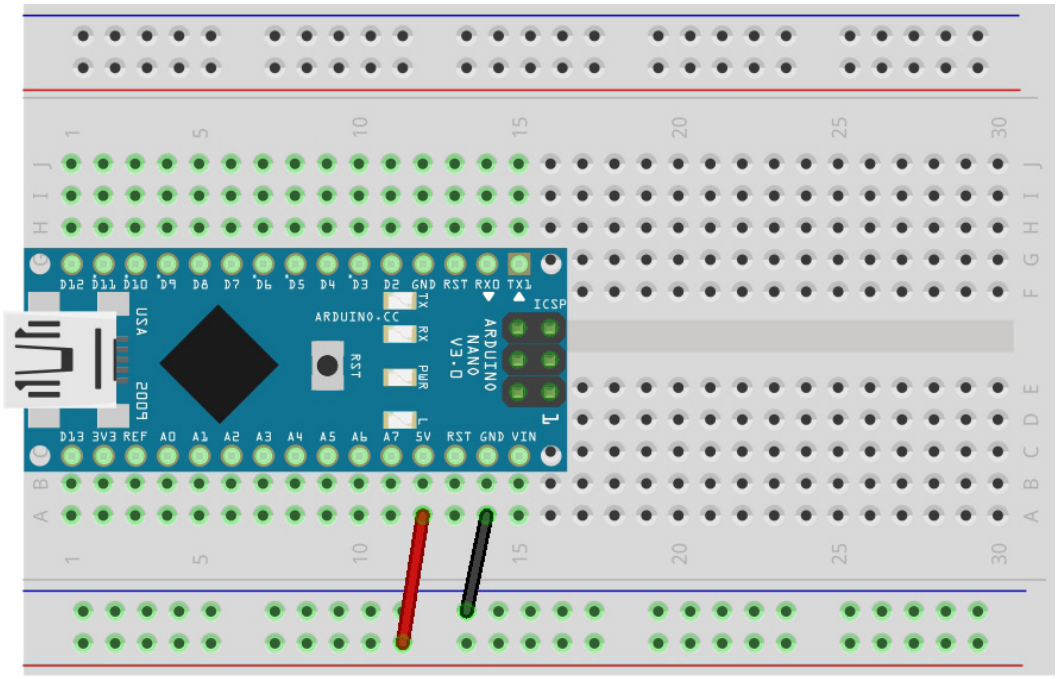
Aldo Pizzeghella - *www.aldopi.it - aldopi62@gmail.com*

```
/*****************************************
    progetto: corso BASE Arduino
  descrizione: orologio ore.miniti.secondi
     autore: www.AldoPi.it
       data: 27 dicembre 2016
    versione: 1.0
     shields: max7219 + 8 x display 7 segmenti
 *****************************************
*/

#include "LedControl.h"
#define lc_datain 12
#define lc_cs     11
#define lc_clock  10

LedControl lc = LedControl(lc_datain, lc_clock,
lc_cs, 1);

byte ora = 17, minuti = 24, secondi = 0, centesimi =
0, i;
unsigned long   timer = 0;
int attesa = 10;
unsigned long numero = 0;


void setup() {
 lc.shutdown(0, false);
 lc.setIntensity(0, 3);
 lc.clearDisplay(0);
 for (int y = 9; y > -1; y--) {
  lc.setDigit(0, 0, y, false);
  lc.setDigit(0, 1, y, false);
  lc.setDigit(0, 2, y, false);
  lc.setDigit(0, 3, y, false);
  lc.setDigit(0, 4, y, false);
  lc.setDigit(0, 5, y, false);
  lc.setDigit(0, 6, y, false);
  lc.setDigit(0, 7, y, false);
  delay(200);
 }
 delay(250);
 lc.clearDisplay(0);
 timer = millis() / 1000;
}


void loop() {
 i = millis() / 1000 - timer;
 if (i > 0) {
  timer = millis() / 1000;
  incremento(i);
 }
}

void incremento(int i) {
 secondi += i;
 if (secondi == 60) {
  secondi = 0;
  minuti++;
  if (minuti == 60) {
   minuti = 0;
   ora++;
   if (ora == 100)
    ora = 0;
  }
 }
 visualizza();
}


void visualizza() {
 if (((ora / 10) % 10) == 0) lc.setRow(0, 7, 0);
 else lc.setDigit(0, 7, (ora / 10) % 10, false);
 lc.setDigit(0, 6, ora % 10, false);
 if (((minuti / 10) % 10) == 0) lc.setRow(0, 4, 0);
 else lc.setDigit(0, 4, (minuti / 10) % 10, false);
 lc.setDigit(0, 3, minuti % 10, false);
 if (((secondi / 10) % 10) == 0) lc.setRow(0, 1, 0);
 else lc.setDigit(0, 1, (secondi / 10) % 10, false);
 lc.setDigit(0, 0, secondi % 10, false);
 if (secondi % 2) {
  lc.setRow(0, 5, B10000000);
  lc.setRow(0, 2, B10000000);
 } else {
  lc.setRow(0, 5, 0);
  lc.setRow(0, 2, 0);
 }

}

// fine programma
```

Orologio

Display 8 cifre

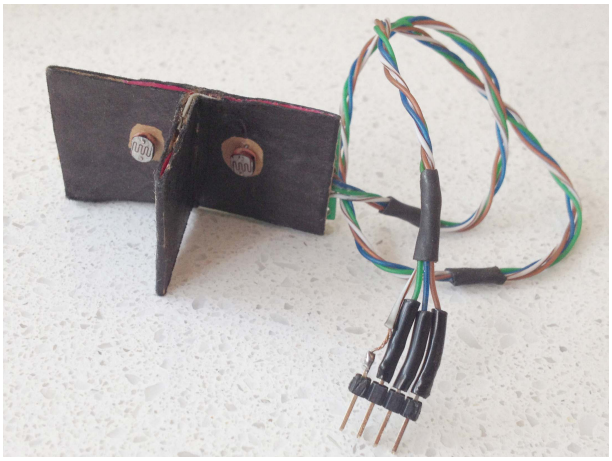
1. vcc 5V
2. Din
3. Load
4. clock
5. gnd



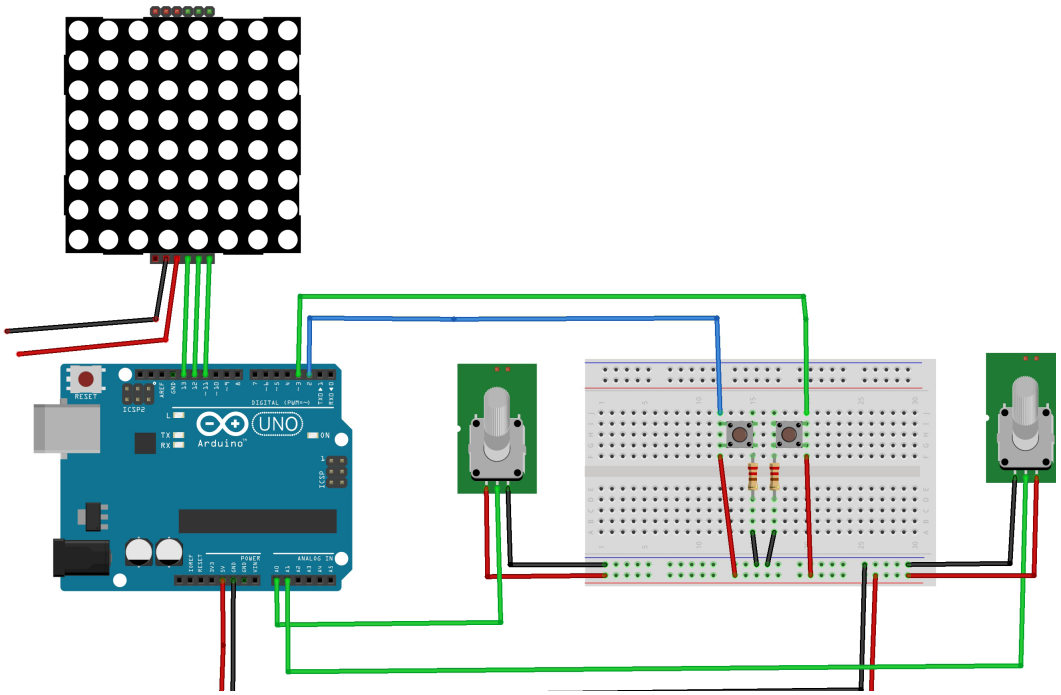
```
#define PIN_START 4
#include “LedControl.h”
#define lc_datain 12
#define lc_clock   10
#define lc_cs              11
LedControl lc = LedControl(lc_datain, lc_clock, lc_cs, 1);

byte ora = 0, minuti = 0, secondi = 0, centesimi = 0, i;
unsigned long      timer = 0;
int attesa = 10;
unsigned long numero = 0;


void setup() {
      lc.shutdown(0, false);
      lc.setIntensity(0, 3);
      lc.clearDisplay(0);
      for (int y = 9; y > -1; y--) {
      timer = millis()/10;
      pinMode(PIN_START,INPUT_PULLUP);
}


void loop() {
      i = millis()/10 - timer;
      if (i > 0) {
             timer = millis()/10;
             incremento(i);
      }
}


void incremento(int i) {
      if (digitalRead (4)==0) centesimi += i;
      if (centesimi > 99) {
            centesimi = 0;
            secondi++;
            if (secondi == 60) {
                  secondi = 0;
                  minuti++;
                  if (minuti == 60) {
                        minuti = 0;
                        ora++;
                        if (ora == 100)
                              ora = 0;
                  }
            }
      }
      visualizza();
}


void visualizza() {
      if (((ora / 100) % 10) == 0) lc.setRow(0, 7, 0);
      else lc.setDigit(0, 7, (ora / 100) % 10, false);
      lc.setDigit(0, 6, ora % 10, false);
      if (((minuti / 10) % 10) == 0) lc.setRow(0, 5, 0);
      else lc.setDigit(0, 5, (minuti / 10) % 10, false);
      lc.setDigit(0, 4, minuti % 10, false);
      if (((secondi / 10) % 10) == 0) lc.setRow(0, 3, 0);
      else lc.setDigit(0, 3, (secondi / 10) % 10, false);
      lc.setDigit(0, 2, secondi % 10, false);
      lc.setDigit(0, 1, (centesimi / 10) % 10, false);
      lc.setDigit(0, 0, centesimi % 10, false);
}
```

Inseguitore Luce

marron: gnd
rosso: Vcc +5
giallo: comando

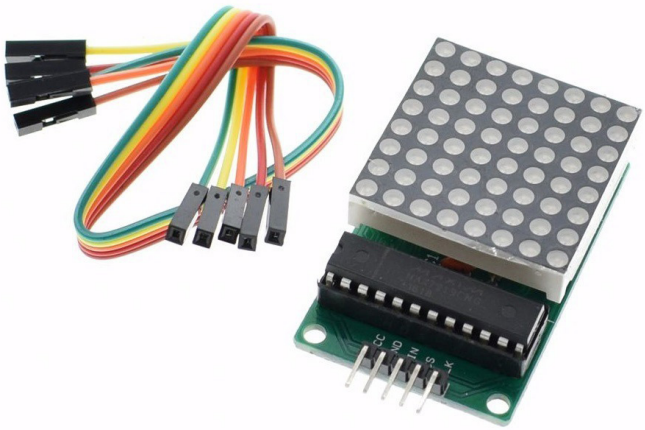


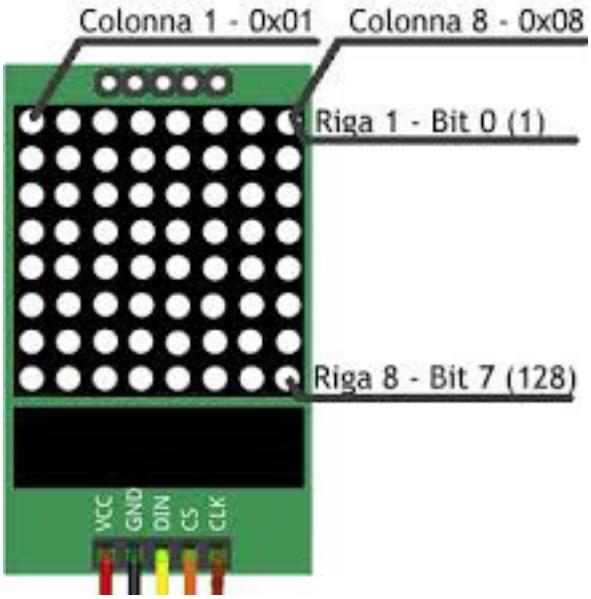
bianco: gnd
verde: photo 1
blu: photo 2
marron : Vcc 5V



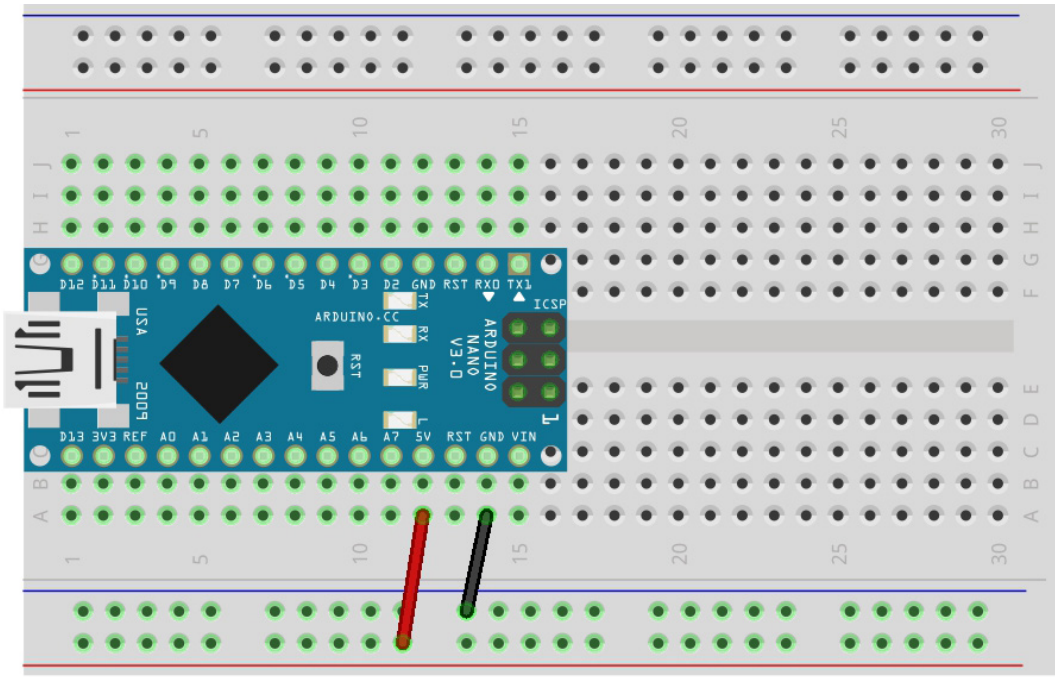
```
#include <Servo.h>
Servo servoMotor;

#define PHOTO_1 0  // porte analogiche
#define PHOTO_2 1  // porte analogiche
#define SOGLIA 20
#define SENSIBILITA 1
#define SERVOPIN 8 // porta digitale

int posizione = 90;
int i=0;
long media=0;
int val=0;
int differenza=0;


void setup() {
     servoMotor.attach(SERVOPIN);
     servoMotor.write(5);
     delay(750);
     servoMotor.write(175);
     delay(750);
     servoMotor.write(90);
     delay(2000);
}


void loop(){
     differenza = sonde();
     if (differenza > 200)        posizione = posizione + 5;
     else if (differenza > 0)     posizione = posizione + 1;
     else if (differenza < -200)  posizione = posizione - 5;
     else if (differenza < 0)     posizione = posizione - 1;
     if (posizione<5) posizione=5;
     else if (posizione >175) posizione = 175;
     servoMotor.write(posizione);
     delay(100);
}


int sonde() {
     int dif = analogRead(PHOTO_1)/SENSIBILITA - analogRead(PHOTO_2)/SENSIBILITA;
     if (dif<SOGLIA*SENSIBILITA and dif>-SOGLIA*SENSIBILITA) dif=0;
     return dif;
}
```

Display 8x8





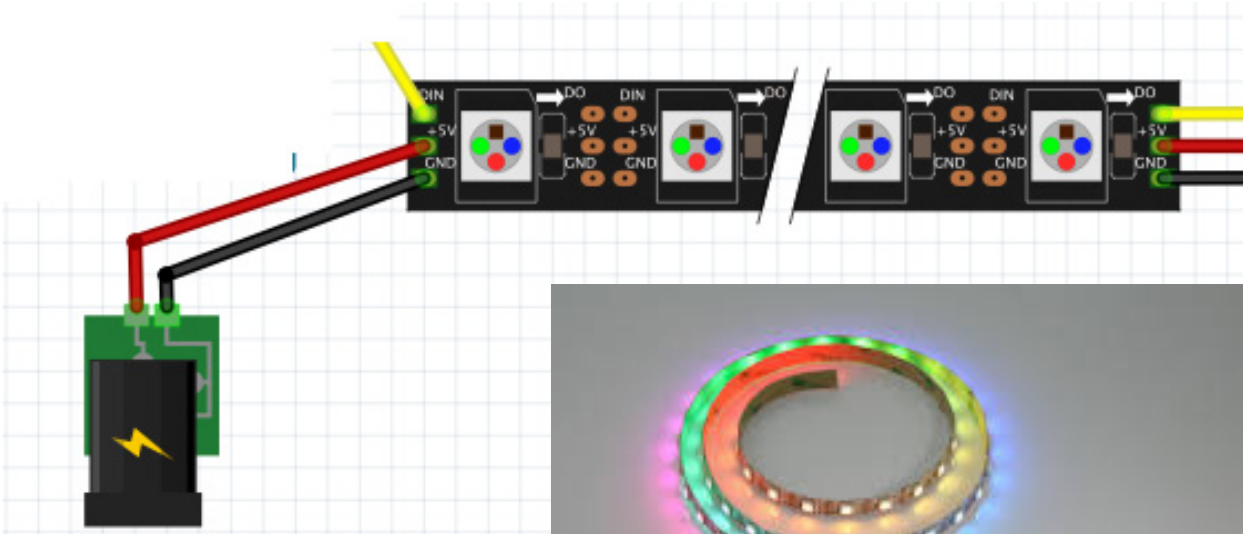
1. Vcc
2. GND
3. Din
4. CS
5. CLK



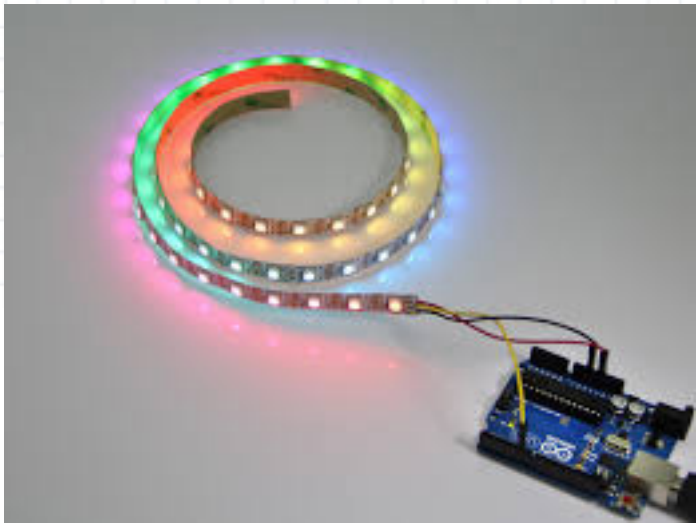
```
#define attesa  10        // tempo in mSec di attesa
#define numero  20        // numero di led accesi contemporaneamente

#include “LedControl.h”        // importa libreria
LedControl  led=LedControl(12,10,11,4);    // pin: 12-DATA 11-CLOCK 10-CS
int x[numero + 1];                      // VETTORE posizione X dei led accesi
int y[numero + 1];                      // VETTORE posizione Y dei led accesi
int i;


void setup() {
      led.shutdown(0,false);                // accendo la madrice
      led.setIntensity(0,2);                // regolo la luminosit√†
      led.clearDisplay(0);                  // cancello la matrice
      for (i=1; i<numero; i++) {            // accendo N led in posizione casuale
            x[i] = random(0,8);
            y[i] = random(0,8);
          led.setLed(0, x[i], y[i], true); // accendo il led
      }
      i=1;
}


void loop() {  // ----------------  loop
      led.setLed(0, x[i], y[i], false);
      x[i] = random(0,8);
      y[i] = random(0,8);
      led.setLed(0, x[i], y[i], true);
      if (++i > numero) i=1;
      delay(attesa);
}
```

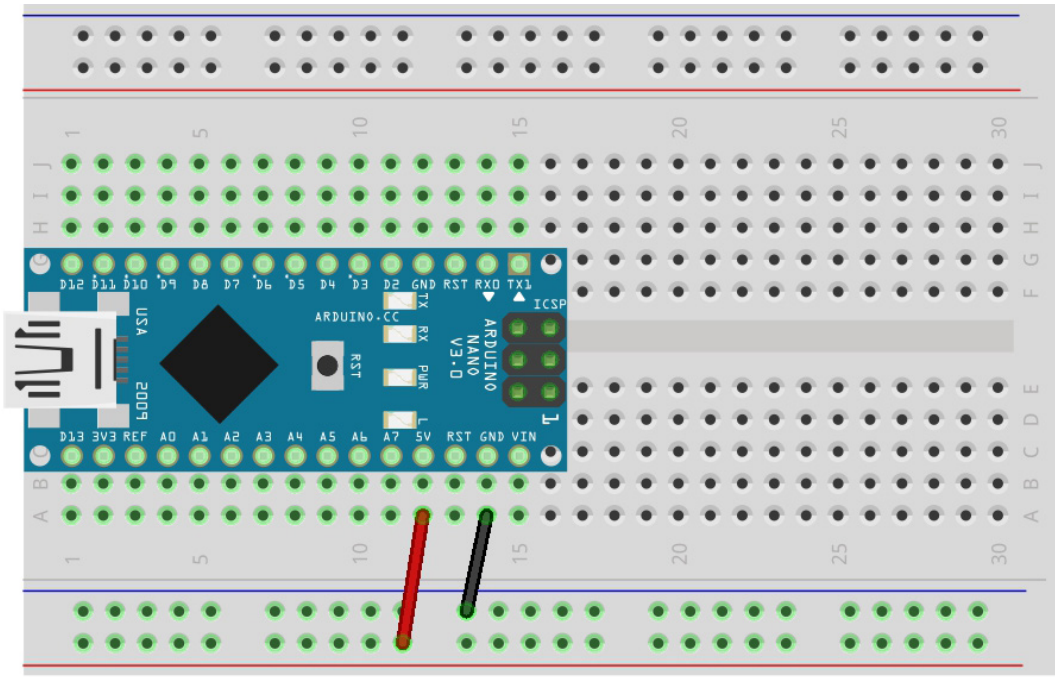

Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

Striscia led RGB



1. Din
2. Vcc 12V
3. GND





```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

#define PIN 12
#define NLED 40

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NLED, PIN, NEO_GRB + NEO_KHZ800);

int val, led;
void setup() {
      led = 1;
      strip.begin();
      strip.show();
      randomSeed(analogRead(0));
}
void loop() {
      flash(60);
      delay(50000);
}

void flash(int tempo) {
      uint32_t inizio = millis() / 1000;
      do {
            delay(random(100, 1000));
            led = random(0, NLED);
            strip.setPixelColor(led, strip.Color(random(0, 150), random(0, 150),
                                                                  random(0,
150)));
            strip.show();
            delay(50);
            strip.setPixelColor(led, strip.Color(0, 0, 0));
            strip.show();
      } while ((millis() / 1000 - inizio) < tempo);
}
```


Aldo Pizzeghella - www.aldopi.it - aldopi62@gmail.com

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

#define LED_PIN    12
#define LED_NUM    40

uint32_t colore = 0;
int passo, velocita, durata1;
uint32_t  ciclo_tempo  = 0;

Adafruit_NeoPixel strip = Adafruit_NeoPixel(LED_NUM, LED_PIN, NEO_RGB + NEO_KHZ800);

void setup() {
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
}

void loop() {
  bandiera(10, 50);
}


void bandiera(uint8_t velocita, int ddd) {
  int y;
  unsigned long ttt;
  do {
    for (y = 1; y < LED_NUM - 12; y++) {
      strip.setPixelColor(y - 1, strip.Color(0, 0, 0));
      strip.setPixelColor(y, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 1, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 2, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 3, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 4, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 5, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 6, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 7, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 8, strip.Color(0, 50, 0));
      strip.setPixelColor(y + 9, strip.Color(0, 50, 0));
      strip.setPixelColor(y + 10, strip.Color(0, 50, 0));
      strip.setPixelColor(y + 11, strip.Color(0, 50, 0));
      strip.show();
      delay(velocita);
    }
    delay(velocita);
    for (y = LED_NUM - 5; y >= 0; y--) {
      strip.setPixelColor(y + 6, strip.Color(0, 0, 0));
      strip.setPixelColor(y, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 1, strip.Color(50, 0, 0));
      strip.setPixelColor(y + 2, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 3, strip.Color(40, 40, 40));
      strip.setPixelColor(y + 4, strip.Color(0, 50, 0));
      strip.setPixelColor(y + 5, strip.Color(0, 50, 0));
      strip.show();
      delay(velocita);
    }
    delay(velocita);
  } while ((millis() / 100 - ciclo_tempo)  < ddd);
}
```