

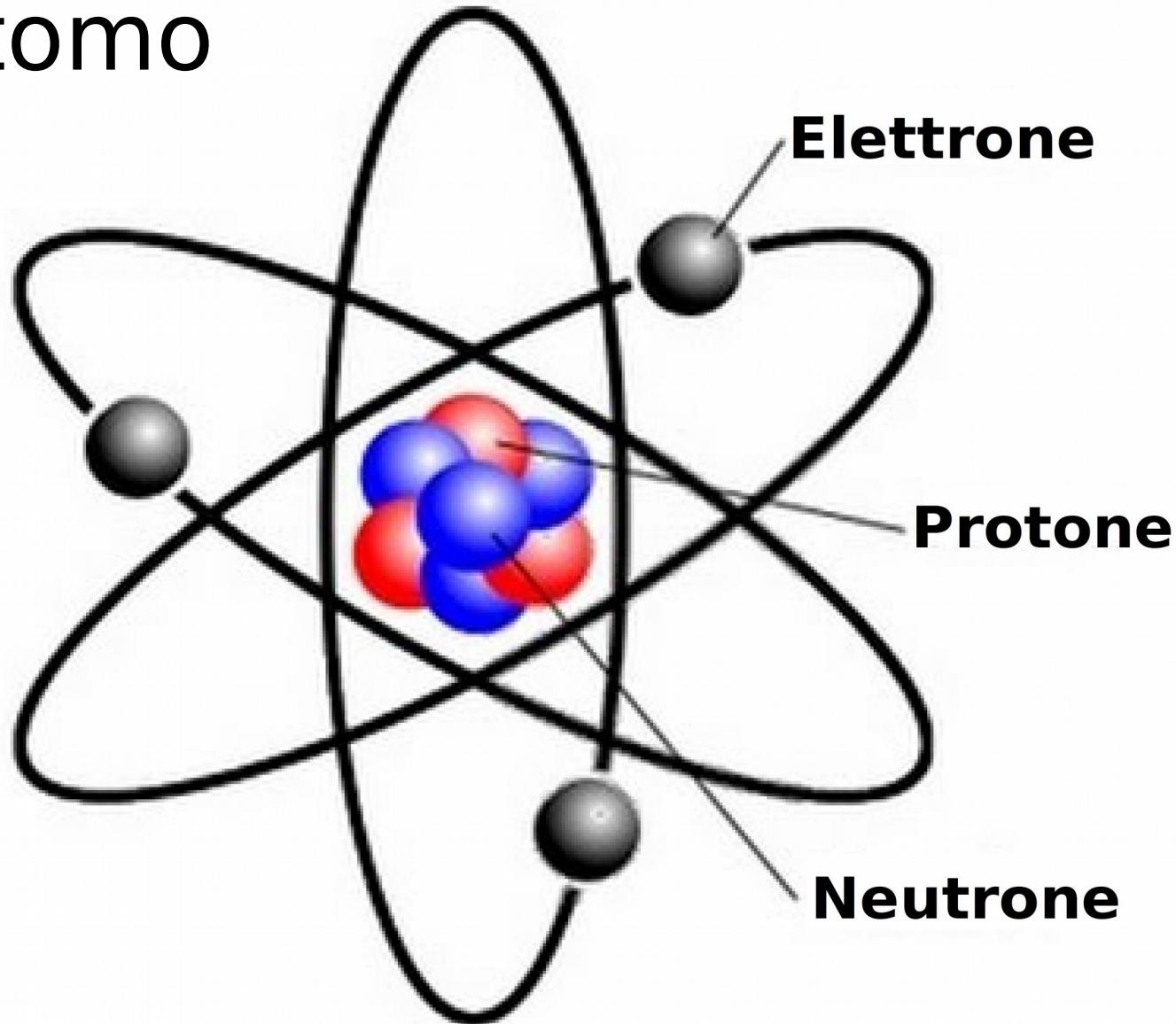
ELETTRONICA PRATICA E ARDUINO



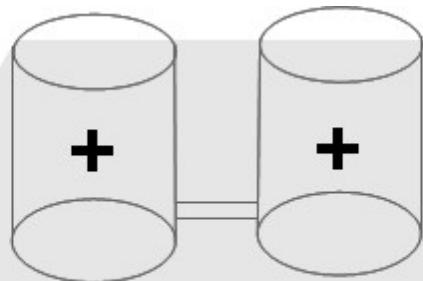
L'ELETTRICITÀ

9

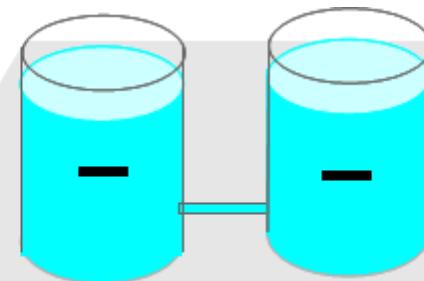
L'atomo



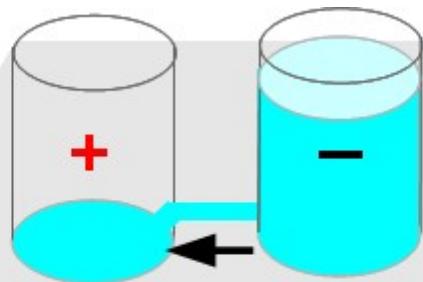
L' ELETTRICITÀ



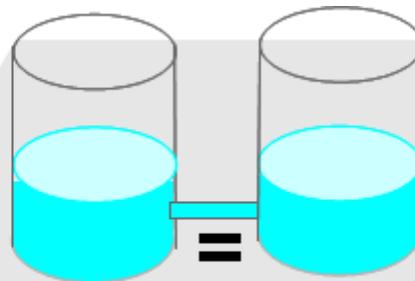
Nessun flusso



Nessun flusso



Flusso



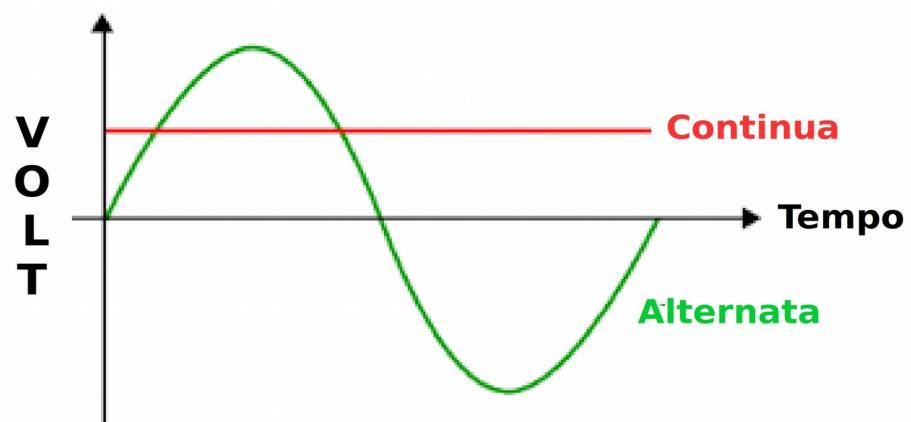
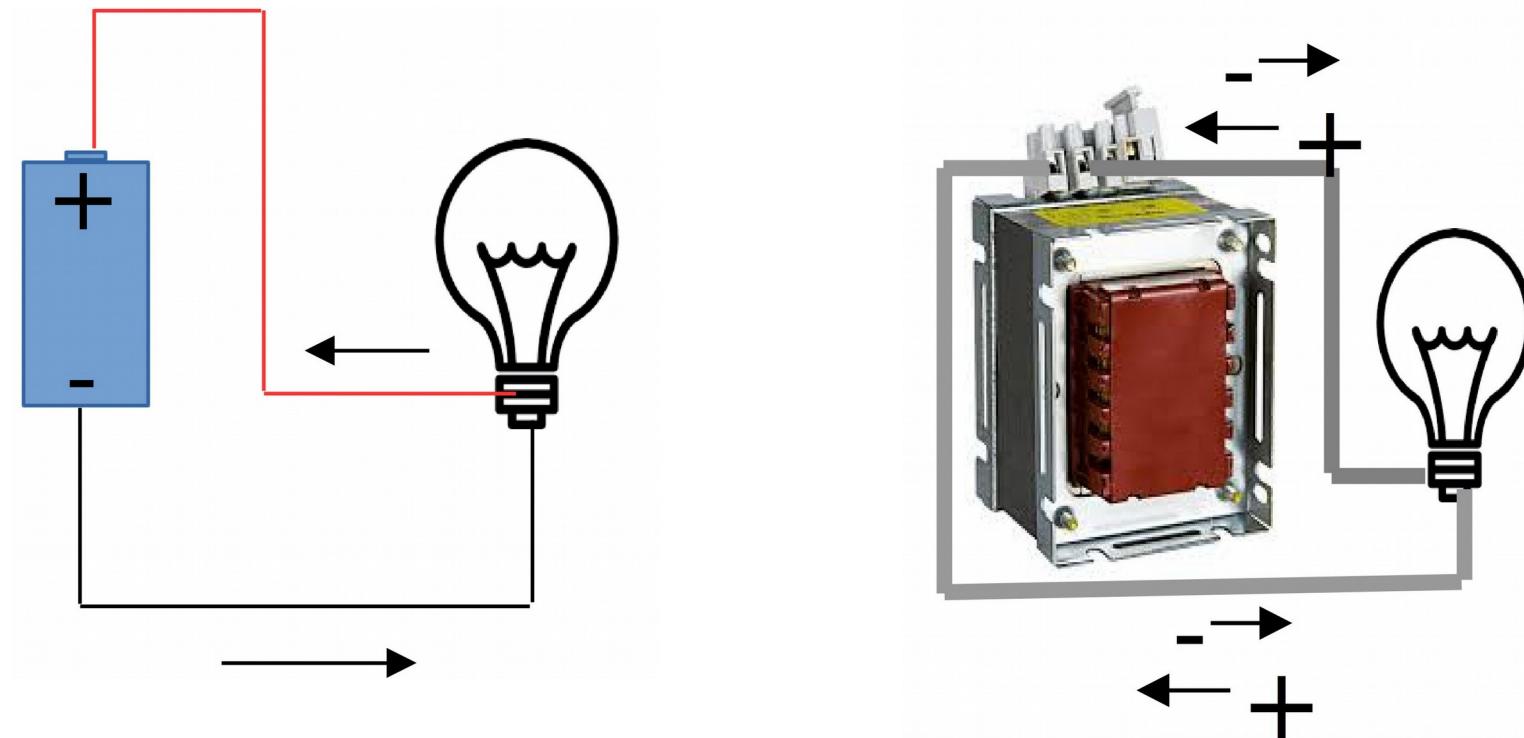
Nessun flusso

L'ELETTRICITÀ

11

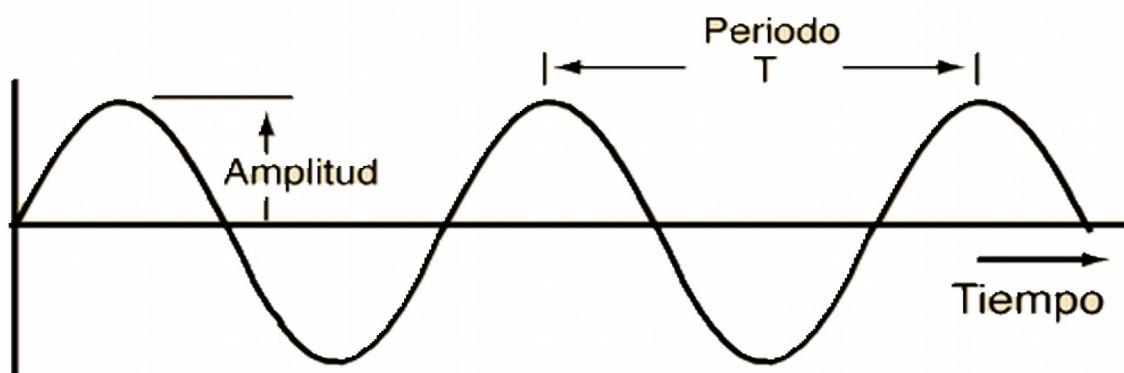
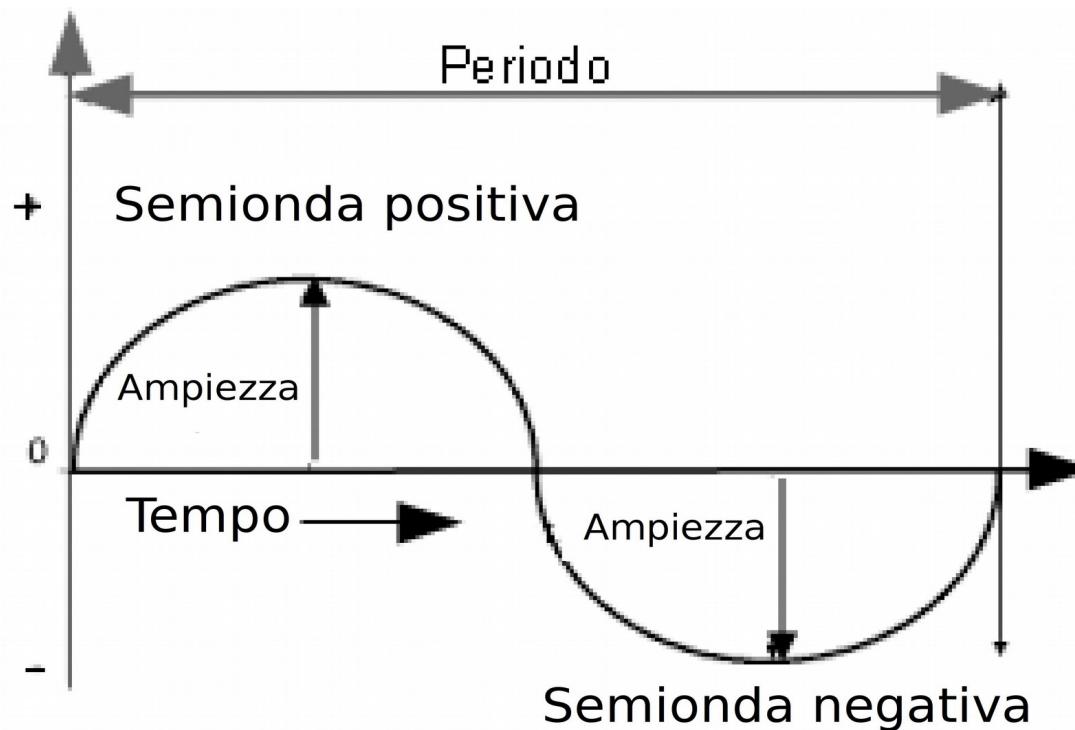


L'ELETTRICITÀ



L'ELETTRICITÀ

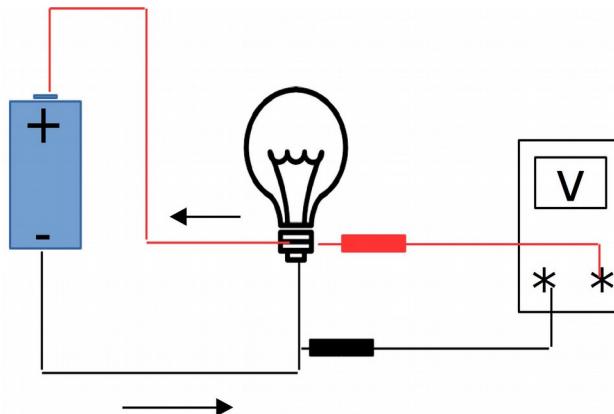
13



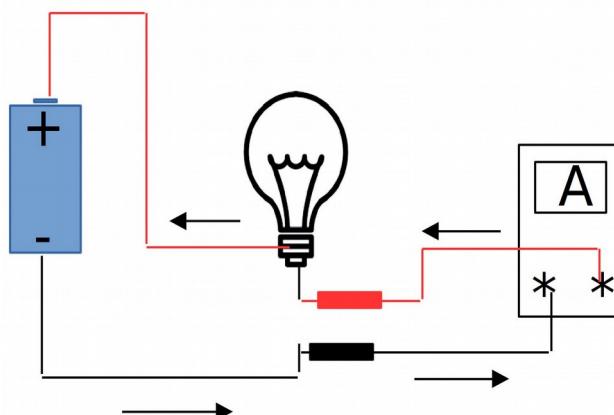
LA CORRENTE

15

La corrente elettrica si misura in Amper e nei suoi sottomultipli:
milli Amper (mA) = Amper / 1,000
micro Amper (μ A) = Amper / 1,000,000



Con un multmetro possiamo misurare la tensione direttamente in **parallelo** ai poli che ci interessano.



Per misurare la corrente dobbiamo mettere il multmetro in **serie** a uno dei poli.

LA POTENCIA

LA POTENZA SI MISURA IN **WATT**

$$\text{Watt} = \text{Volt} * \text{Amper}$$

$$12 * 0.5 = 6$$

$$\text{Volt} = \text{Watt} / \text{Amper}$$

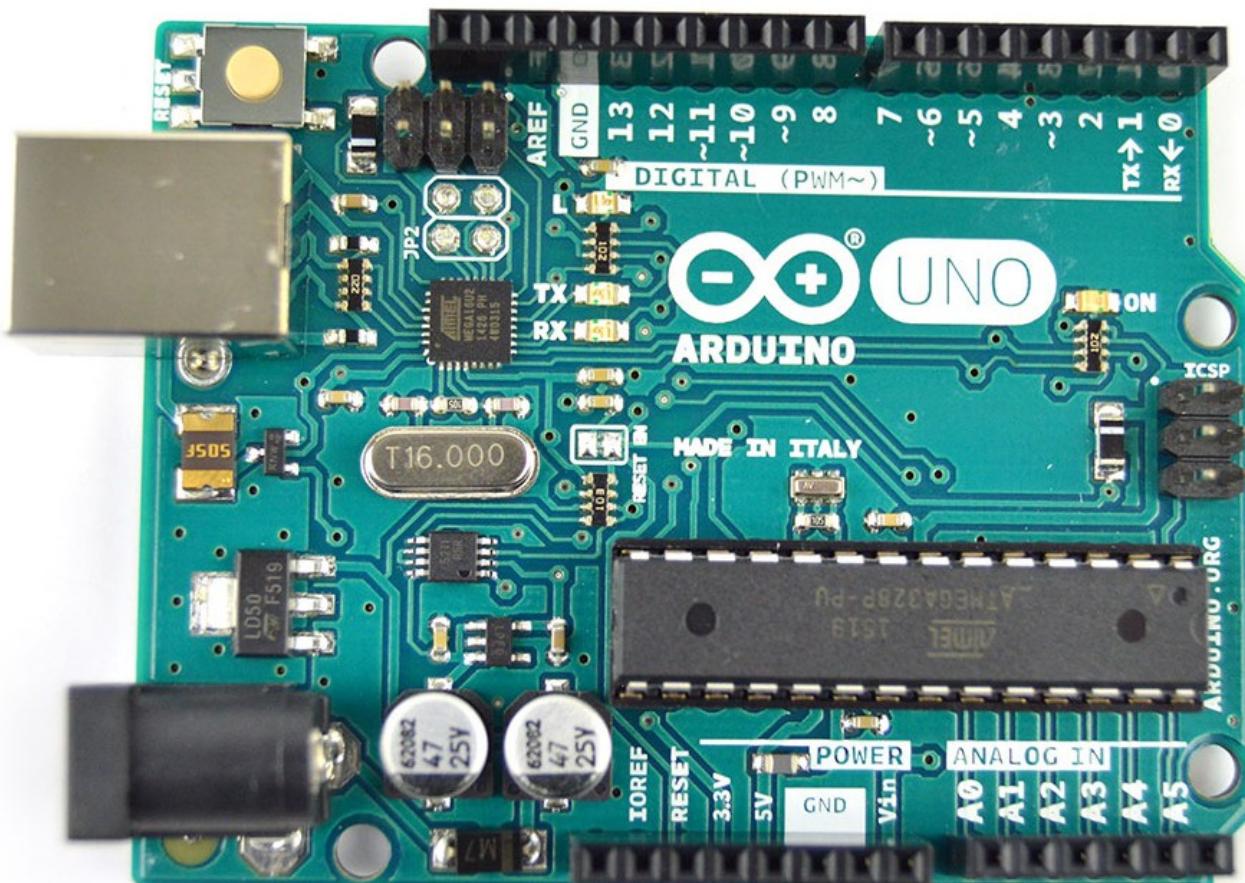
$$6 / 0.5 = 12$$

$$\text{Amper} = \text{Watt} / \text{Volt}$$

$$6 / 12 = 0.5$$

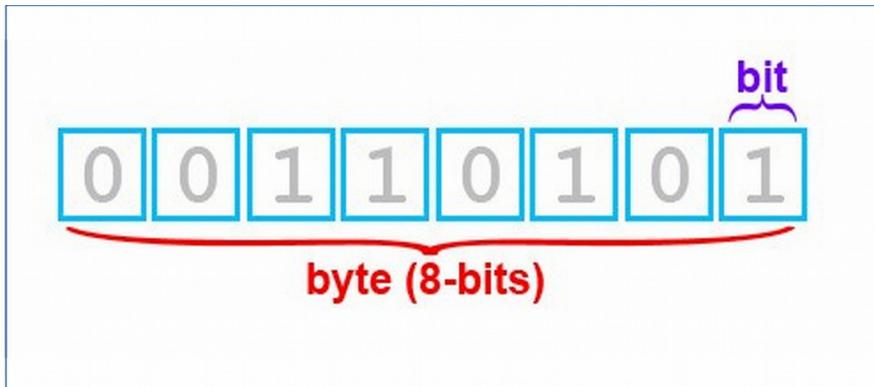
ARDUINO

17



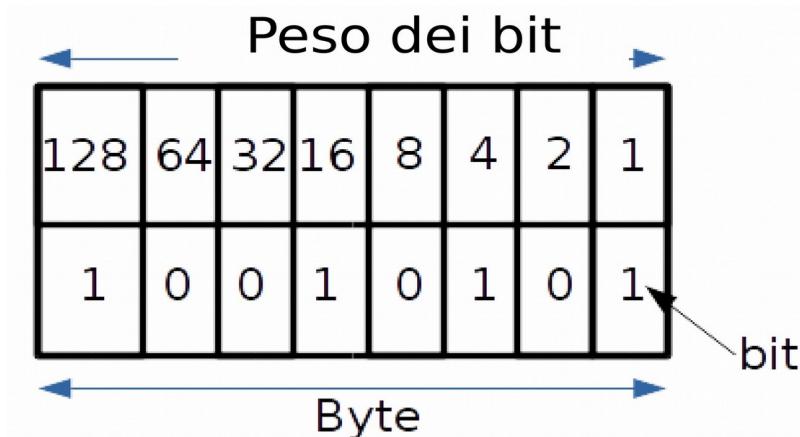
ARDUINO

20



00 / 01 / 02 / 03 / 04 / 05 / 06 / 07 / 08 / 09
10 / 11 / 12 / 13 / 14 / 15 / 16 / 17 / 18 / 19

0000 / 0001 / 0010 / 0011 / 0100 / 0101 / 0110 / 0111 / 1000 / 1001
0 1 2 3 4 5 6 7 8 9

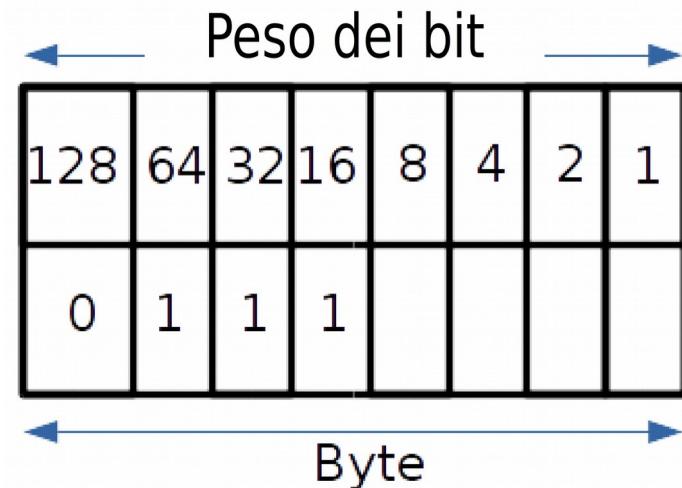
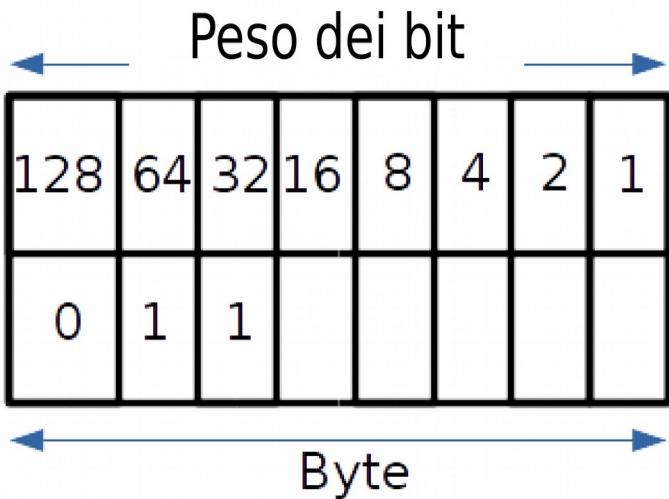
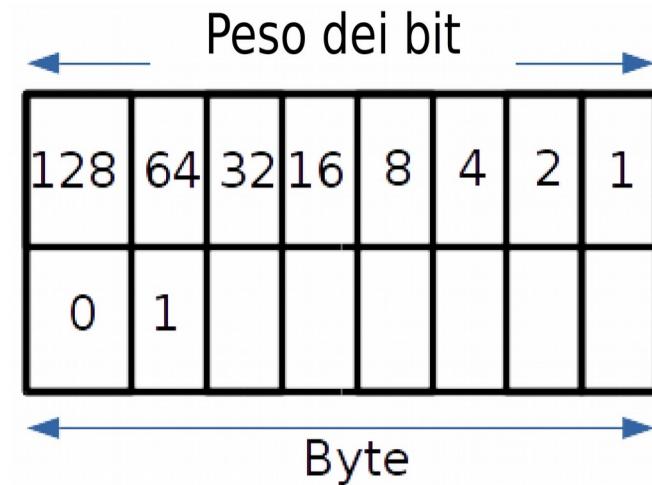
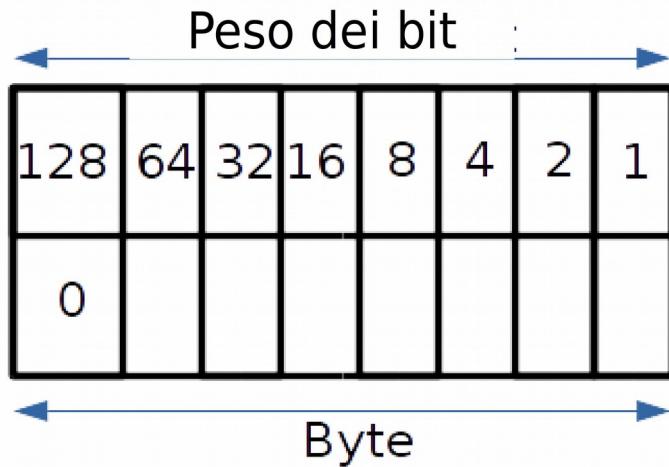


$$1+4+16+128=149$$

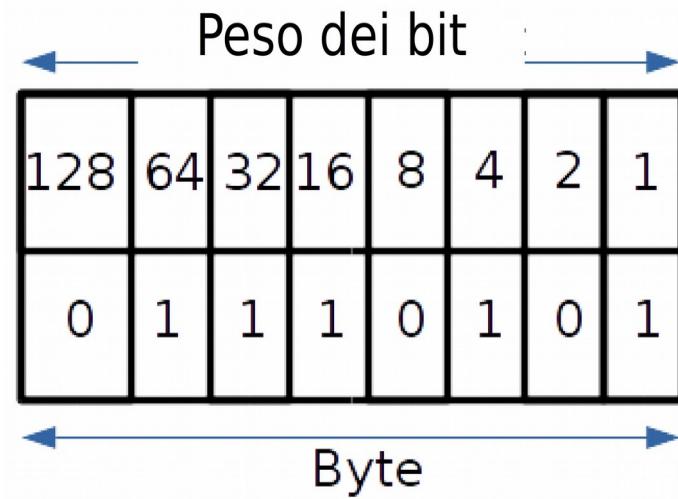
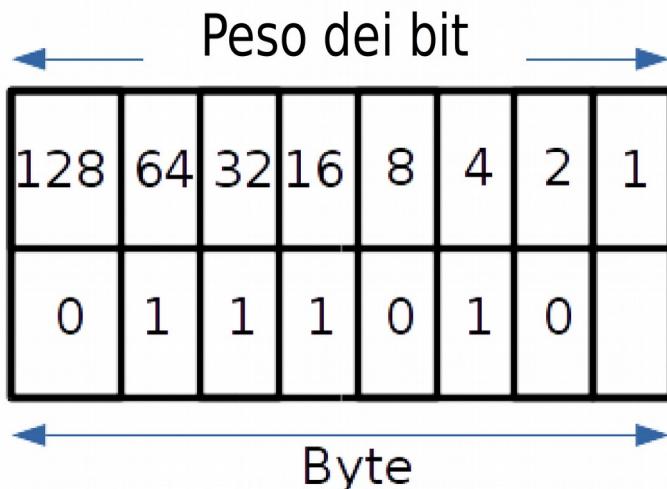
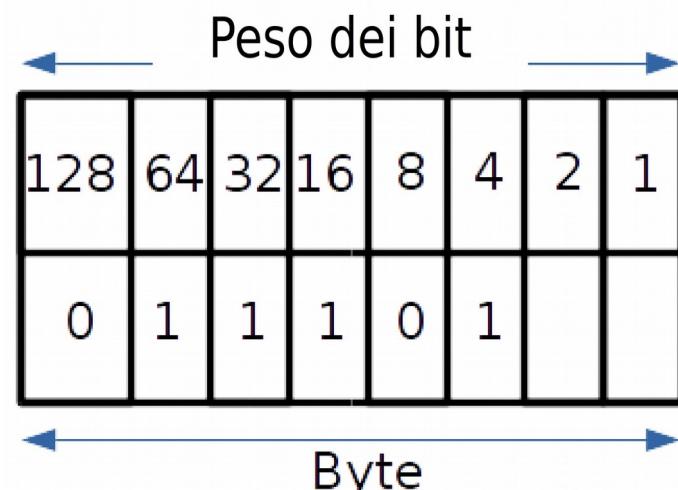


$$1+2+4+8+16+32+64+128=255$$

SCRIVERE IL NUMERO 117



SRIVERE IL NUMERO 117



ARDUINO

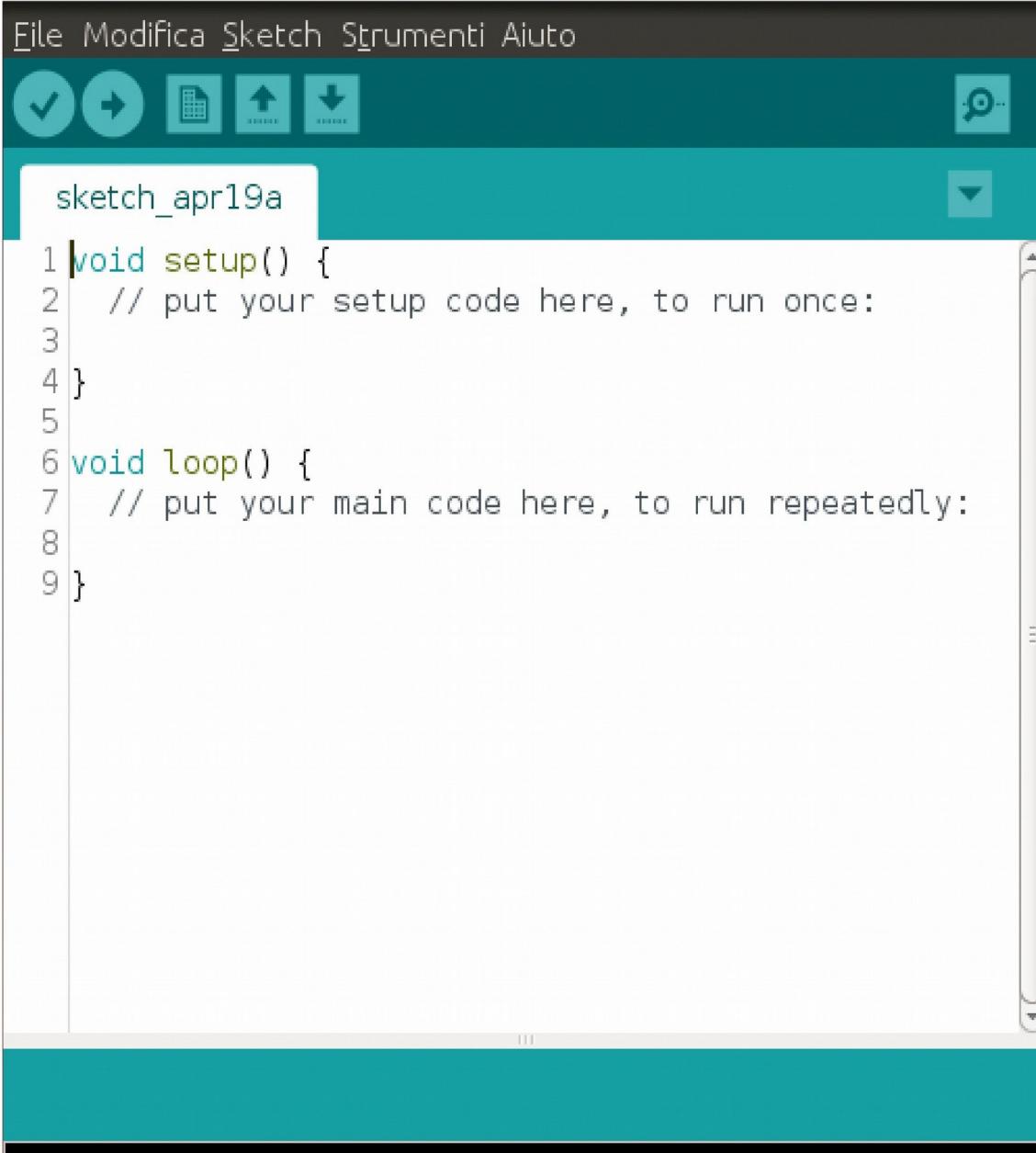
22



65.535

PROGRAMMAZIONE

25



The image shows a screenshot of the Arduino IDE interface. At the top, there is a menu bar with options: File, Modifica, Sketch, Strumenti, Aiuto. Below the menu is a toolbar with several icons: a checkmark, a right arrow, a file folder, an upload arrow, a download arrow, and a magnifying glass. The main workspace is titled "sketch_apr19a". It contains the following code:

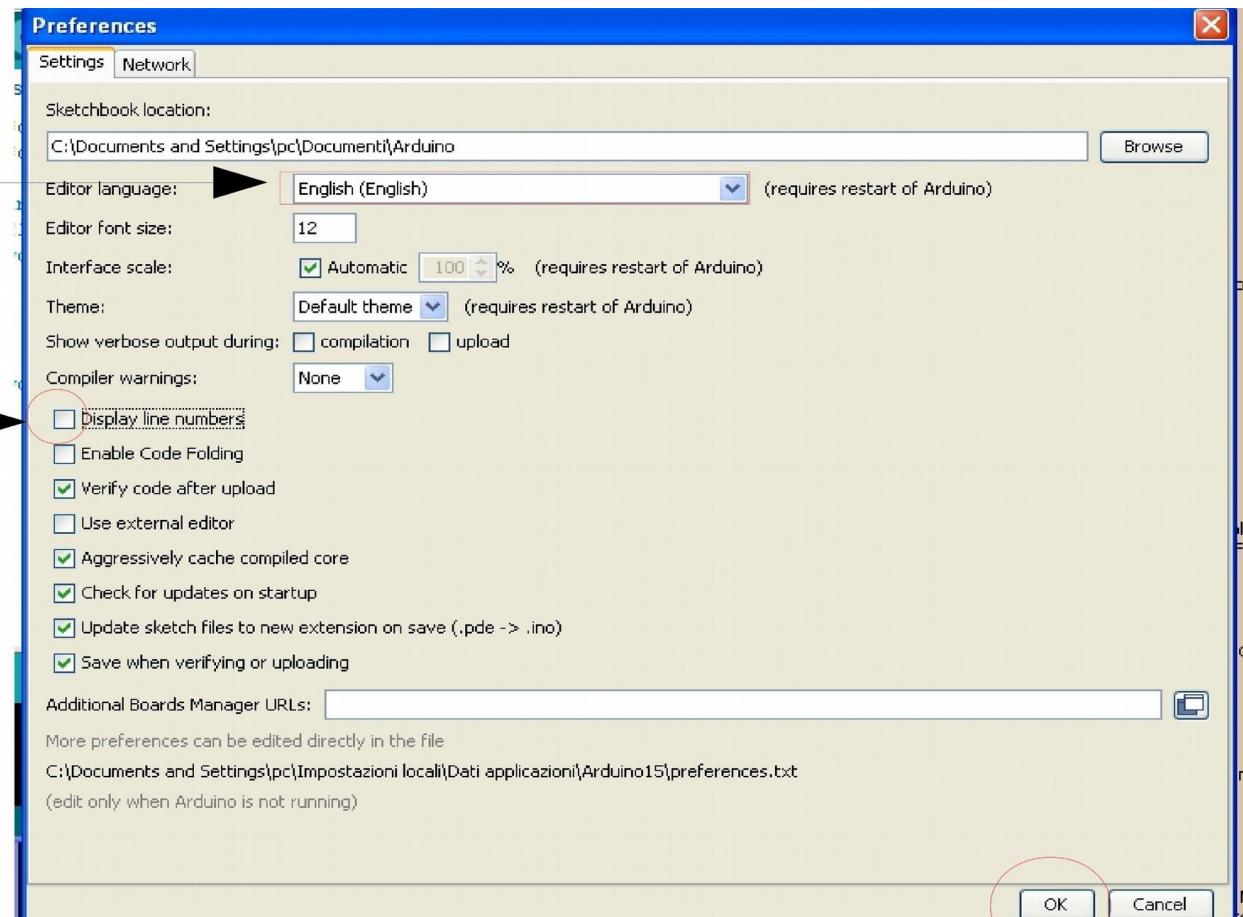
```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8 }  
9 }
```

PROGRAMMAZIONE

26

Qui scegliamo
la lingua

Qui inseriamo i
numeri di linea



Per terminare

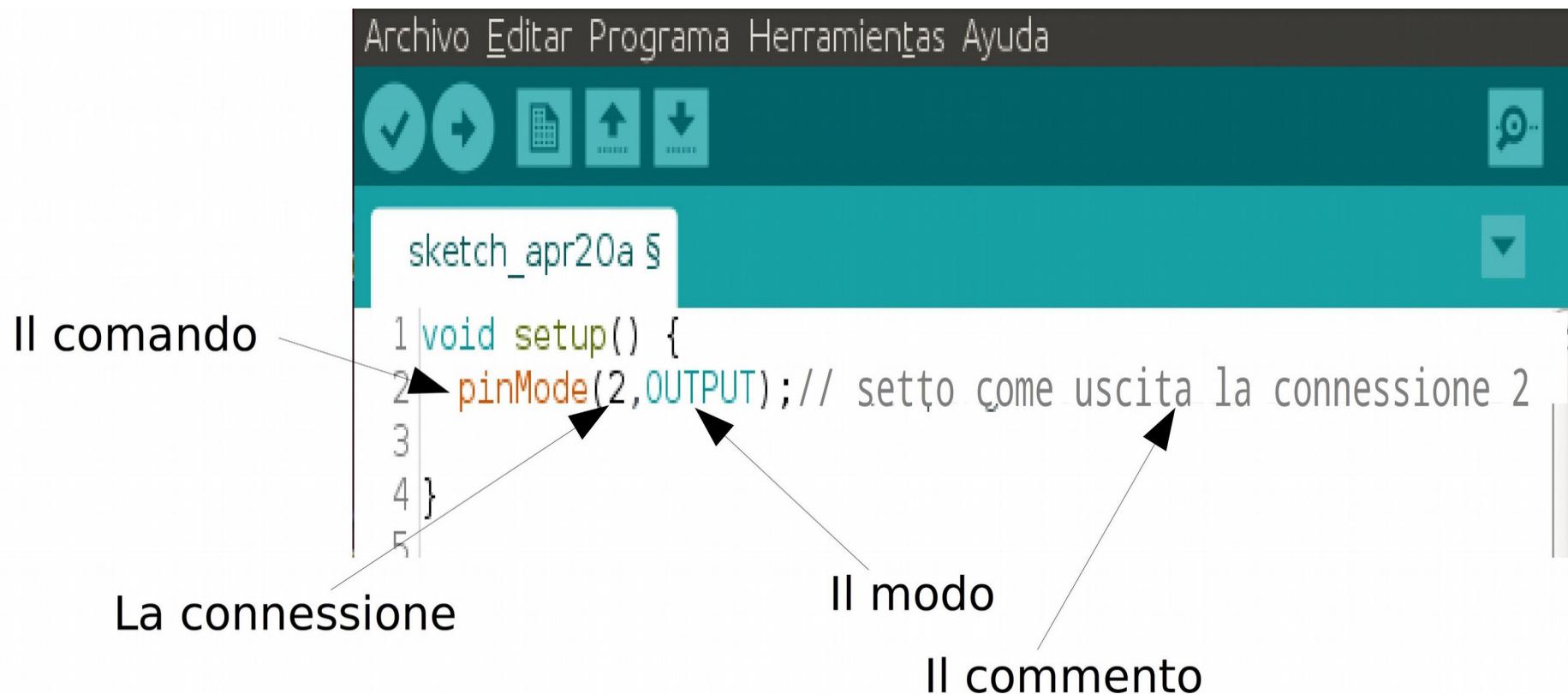
PROGRAMMAZIONE

27



PROGRAMMAZIONE

28



PROGRAMMAZIONE

29

```
5  
6 void loop() {  
7   digitalWrite(2,HIGH); // accendo il led  
8 }  
9  
10}
```

Il comando
La connessione
Il modo
Il commento

```
5  
6 void loop() {  
7   digitalWrite(2,HIGH); // accendo il led  
8   delay(1000); // aspetto un secondo  
9  
10}
```

La funzione
Il valore
Il commento

PROGRAMMAZIONE

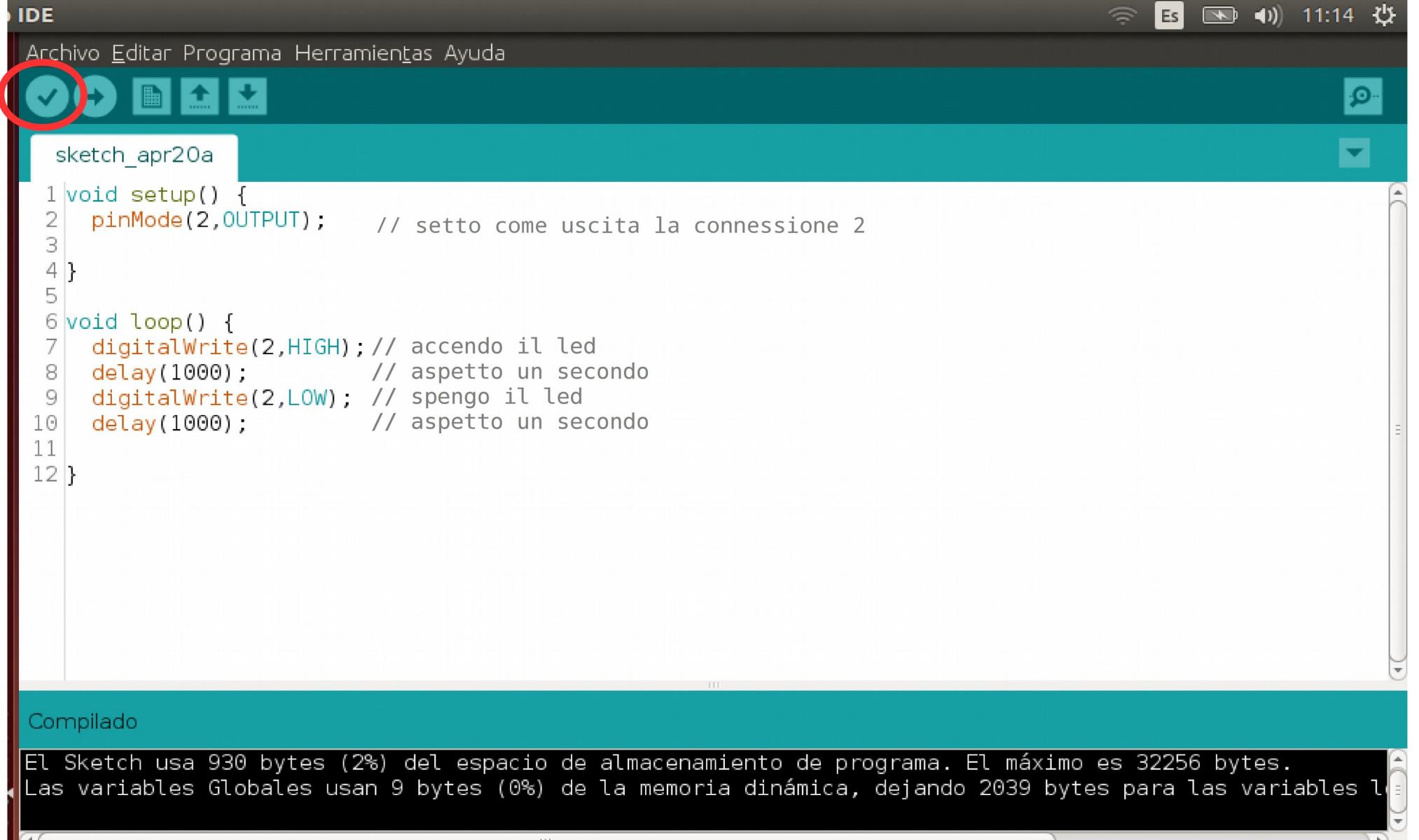
30

```
5
6 void loop() {
7   digitalWrite(2,HIGH); // accendo il led
8   delay(1000); // aspetto un secondo
9   digitalWrite(2,LOW); // spergo il led
10
11 }
```

Il comando
La connessione
Il modo
Il commento

PROGRAMMAZIONE

31



The screenshot shows the Arduino IDE interface. The title bar reads "IDE". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar features icons for file operations (New, Open, Save, Save All, Find, Print) and a checkmark icon, which is circled in red. The code editor displays the following sketch:

```
1 void setup() {  
2   pinMode(2,OUTPUT); // setto come uscita la connessione 2  
3 }  
4  
5 void loop() {  
6   digitalWrite(2,HIGH); // accendo il led  
7   delay(1000); // aspetto un secondo  
8   digitalWrite(2,LOW); // spengo il led  
9   delay(1000); // aspetto un secondo  
10 }  
11  
12 }
```

The status bar at the bottom shows the message: "El Sketch usa 930 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las variables l".

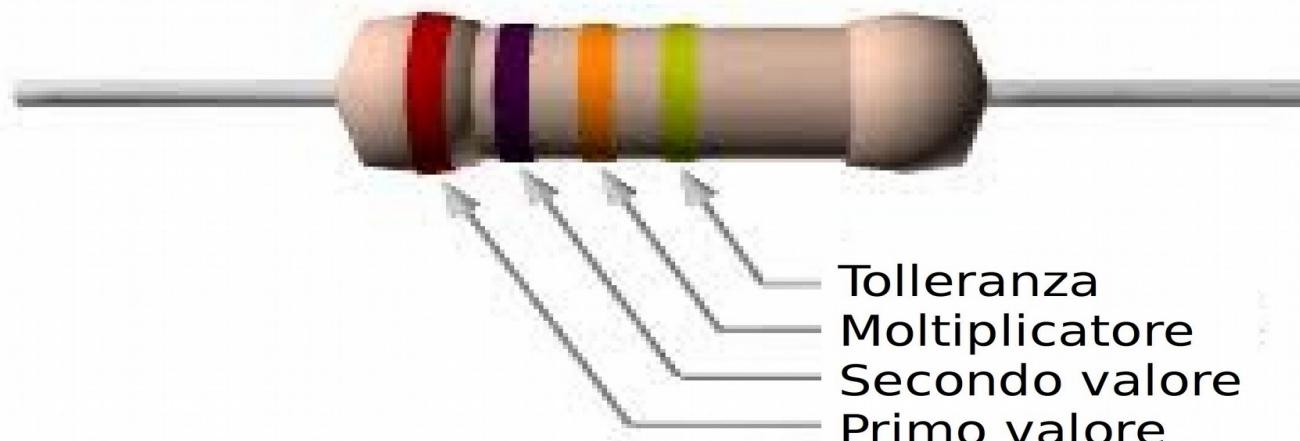
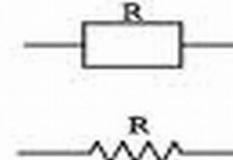
ELETTRONICA

32

LA RESISTENZA SI MISURA IN OHM



simbolo elettrico Resistore



Tolleranza
Moltiplicatore
Secondo valore
Primo valore

ELETTRONICA

33

Resistenze con 4 anelli

colore	1° anello	2° anello	3° anello	4° anello
	1° cifra	2° cifra	moltiplicatore	tolleranza
nero	.	0	x1	-
marrone	1	1	x10	-
rosso	2	2	x100	-
arancione	3	3	x1000 (1KΩ)	-
giallo	4	4	x10000 (10KΩ)	-
verde	5	5	x100000 (100KΩ)	-
blu	6	6	x1000000 (1MΩ)	-
viola	7	7	x10000000 (10MΩ)	-
grigio	8	8	x100000000	-
bianco	9	9	-	5%
oro	-	-	:10	10%
argento	-	-	:100	20%

Resistenze con 5 anelli

colore	1° anello	2° anello	3° anello	4° anello	5° anello
	1° cifra	2° cifra	3° cifra	moltiplicatore	tolleranza
nero	0	0	0	x1	-
marrone	1	1	1	x10	±1%
rosso	2	2	2	x100	±2%
arancione	3	3	3	x1000 (1KΩ)	-
giallo	4	4	4	x10000 (10KΩ)	-
verde	5	5	5	x100000 (100KΩ)	±0,5%
blu	6	6	6	x1000000 (1MΩ)	±0,25%
viola	7	7	7	x10000000 (10MΩ)	±0,1%
grigio	8	8	8	x100000000	±0,025%
bianco	9	9	9	x1000000000	-
oro	-	-	-	:10	±5%
argento	-	-	-	:100	±10%

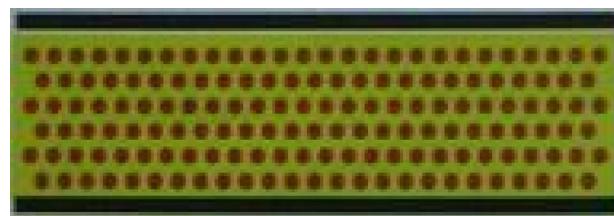
ELETTRONICA

34

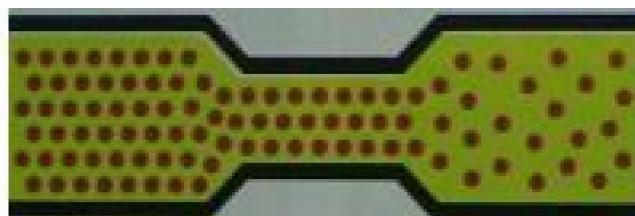
												1 Ω
1,2 Ω	1,5 Ω	1,8 Ω	2,2 Ω	2,7 Ω	3,3 Ω	3,9 Ω	4,7 Ω	5,6 Ω	6,8 Ω	8,2 Ω	10 Ω	
12 Ω	15 Ω	18 Ω	22 Ω	27 Ω	33 Ω	39 Ω	47 Ω	56 Ω	68 Ω	82 Ω	100 Ω	
120 Ω	150 Ω	180 Ω	220 Ω	270 Ω	330 Ω	390 Ω	470 Ω	560 Ω	680 Ω	820 Ω	1K Ω	
1,2K Ω	1,5K Ω	1,8K Ω	2,2K Ω	2,7K Ω	3,3K Ω	3,9K Ω	4,7K Ω	5,6K Ω	6,8K Ω	8,2K Ω	10K Ω	
12K Ω	15K Ω	18K Ω	22K Ω	27K Ω	33K Ω	39K Ω	47K Ω	56K Ω	68K Ω	82K Ω	100K Ω	
120K Ω	150K Ω	180K Ω	220K Ω	270K Ω	330K Ω	390K Ω	470K Ω	560K Ω	680K Ω	820K Ω	1M Ω	
1,2M Ω	1,5M Ω	1,8M Ω	2,2M Ω	2,7M Ω	3,3M Ω	3,9M Ω	4,7M Ω	5,6M Ω	6,8M Ω	8,2M Ω	10M Ω	

ELETTRONICA

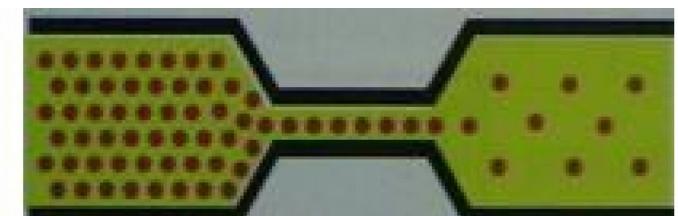
35



Senza resistenza



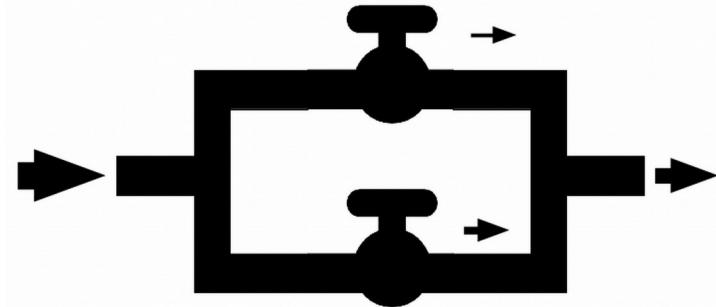
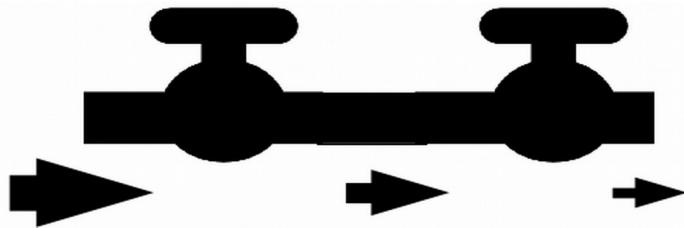
Minima resistenza



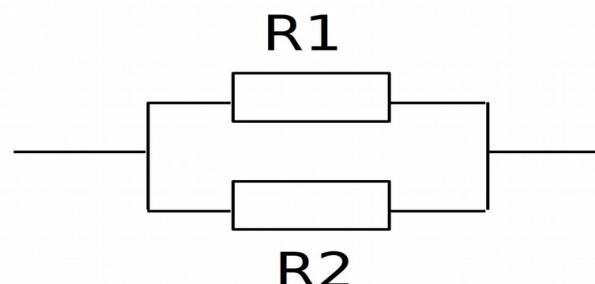
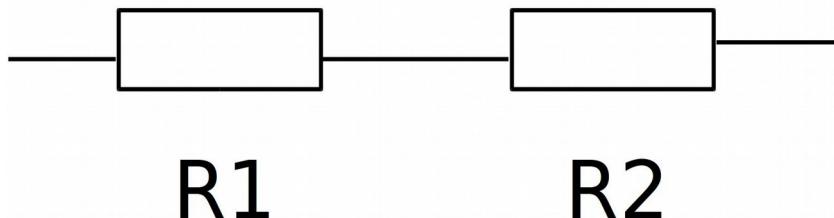
Massima resistenza

ELETTRONICA

36



$$\text{Ohmios} = R_1 + R_2$$

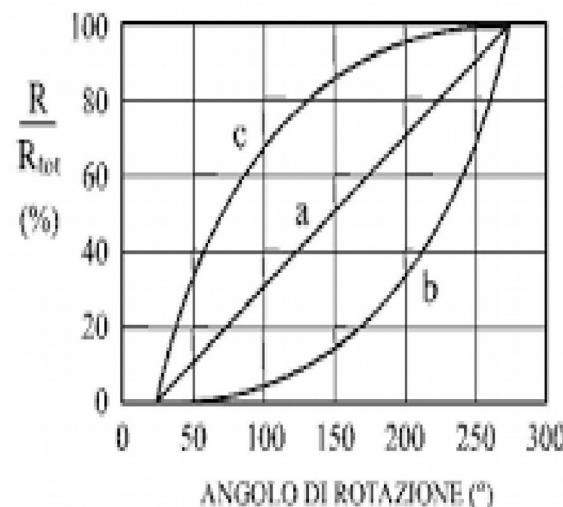
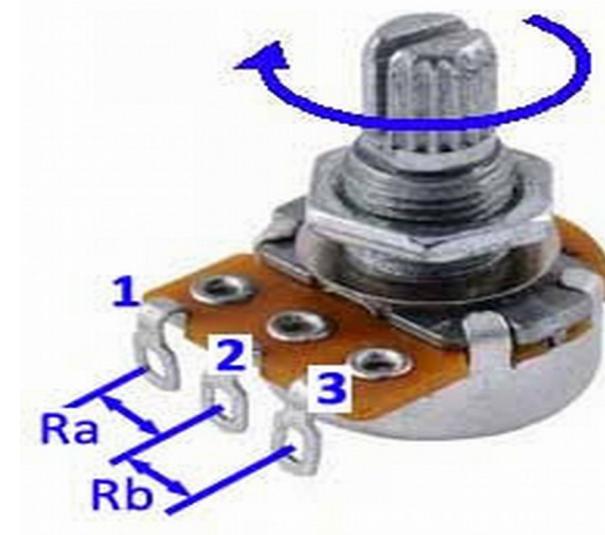
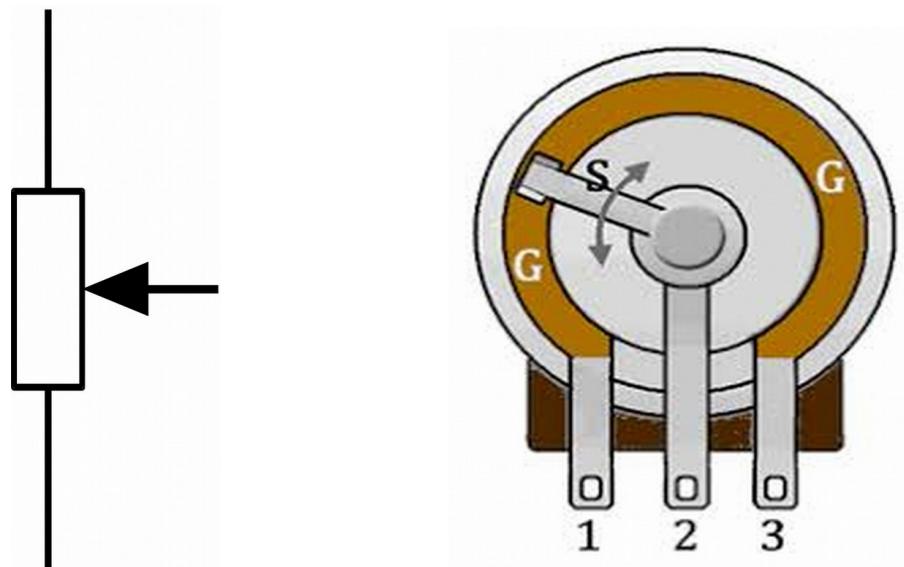


$$1,200 + 1,500 = 2,700 \text{ Ohmios}$$

$$\frac{1,200 * 1,500}{1,200 + 1,500} = 666.66 \text{ Ohmios}$$

ELETTRONICA

37



- a - lineare
- b - logaritmico
- c - esponenziale

A - Lineare

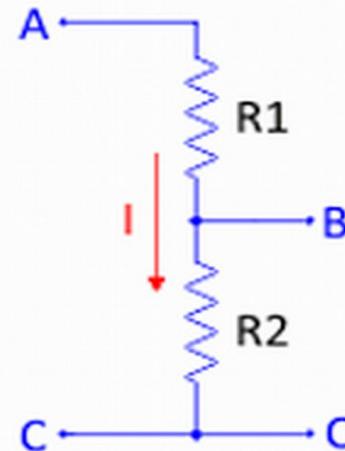
B - Logaritmico

C - esponenziale

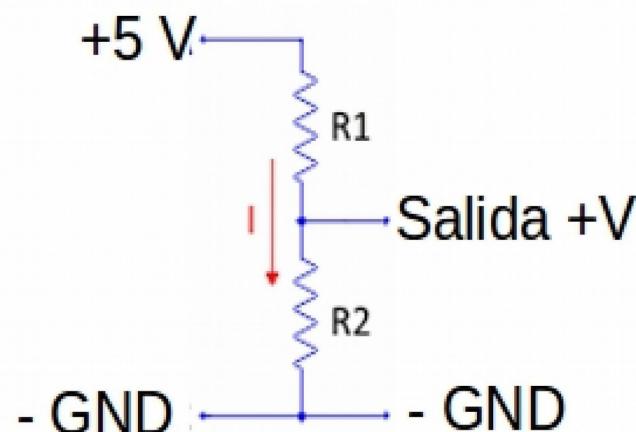
Angolo di rotazione

ELETTRONICA

39



$$V_{BC} = V_{AC} \cdot \frac{R2}{R1+R2}$$



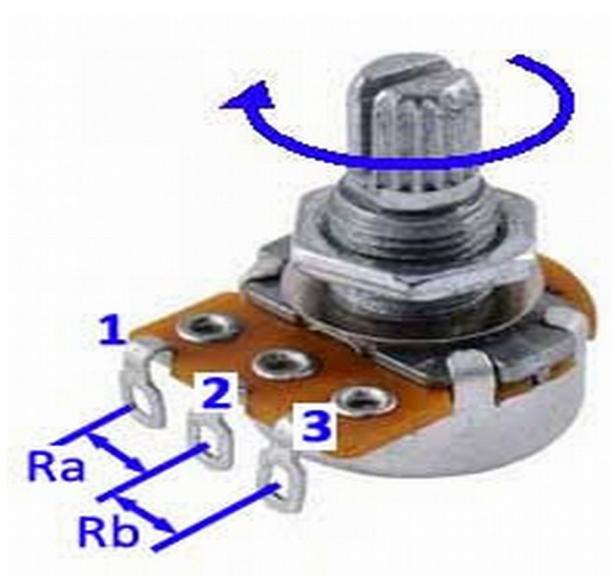
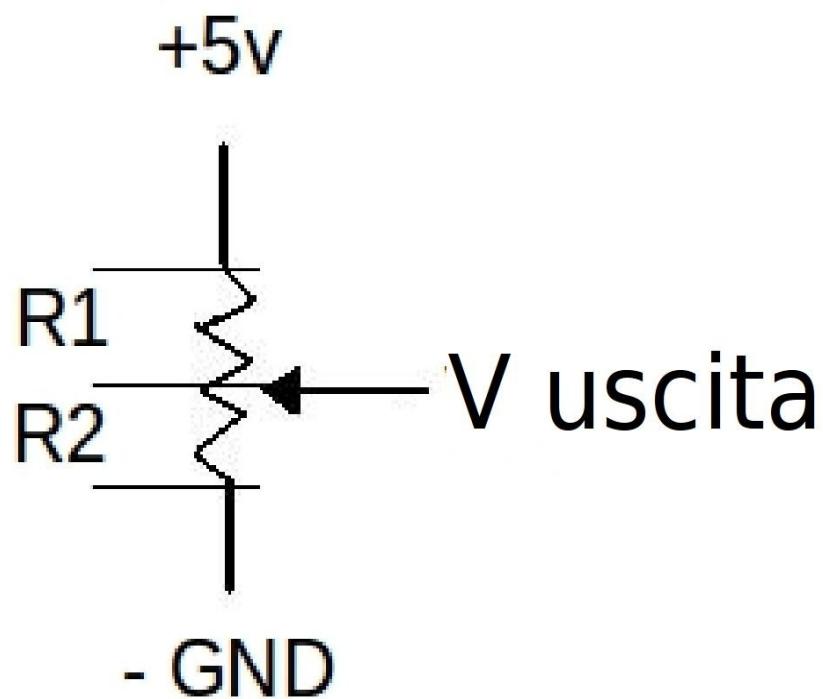
$$V(\text{salida}) = V(\text{entrada}) * \frac{R2}{R1+R2}$$

$$5V * \frac{10000}{10000 + 10000} = 0.5$$

$$5V * 0.5 = 2.5V$$

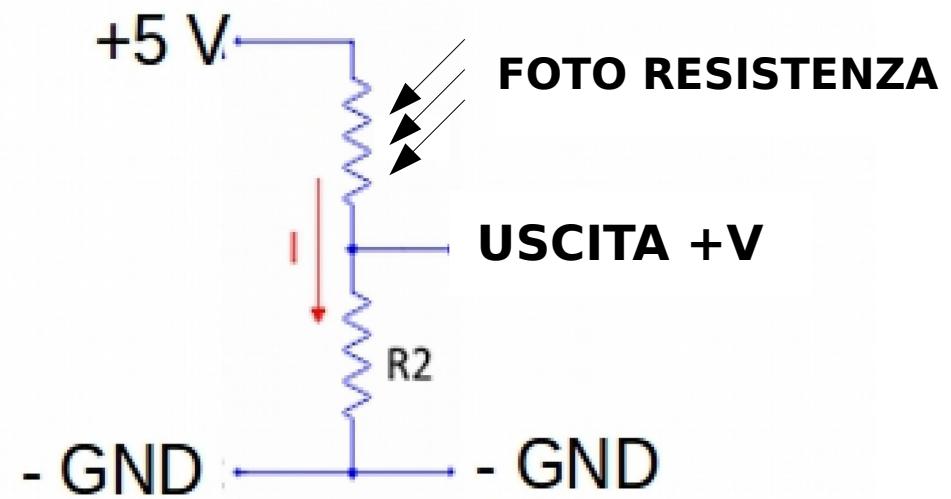
ELETTRONICA

40



ELETTRONICA

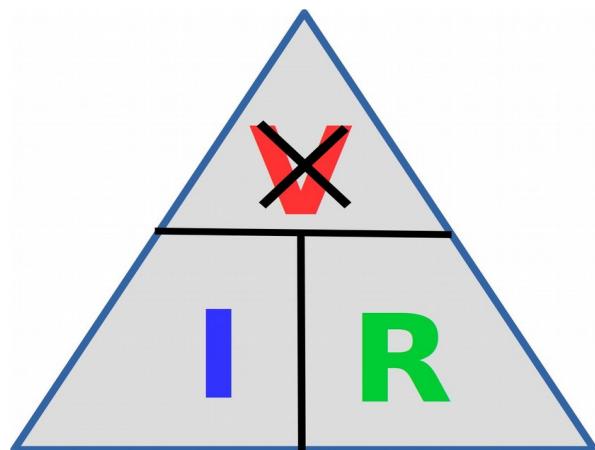
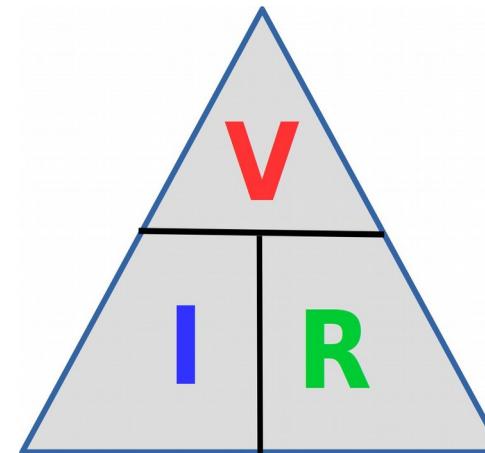
41



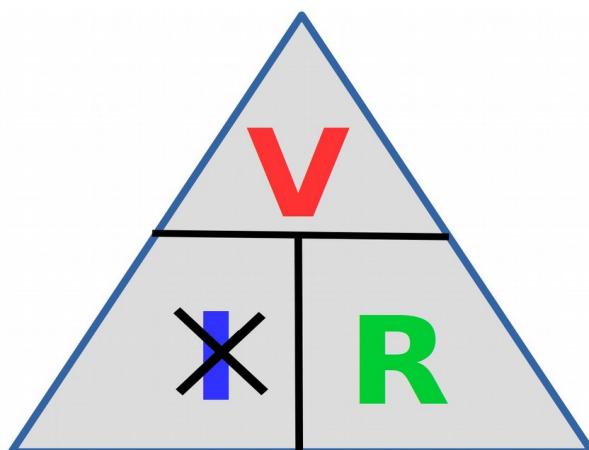
ELETTRONICA

42

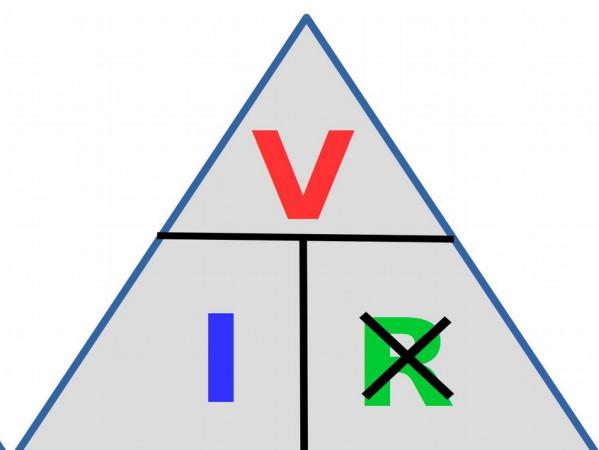
LA LEGGE DI OHM



$$V = I \cdot R$$



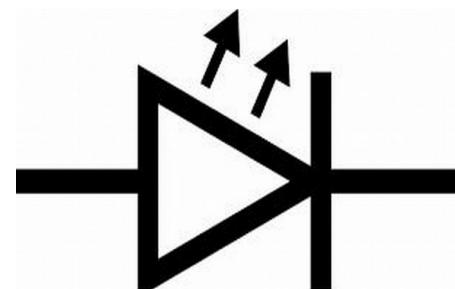
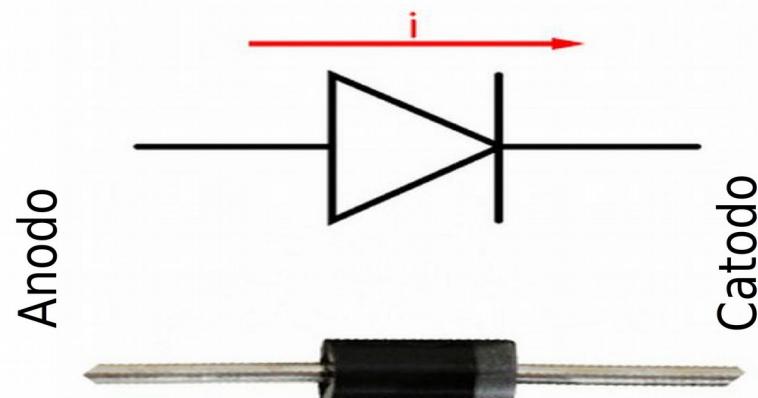
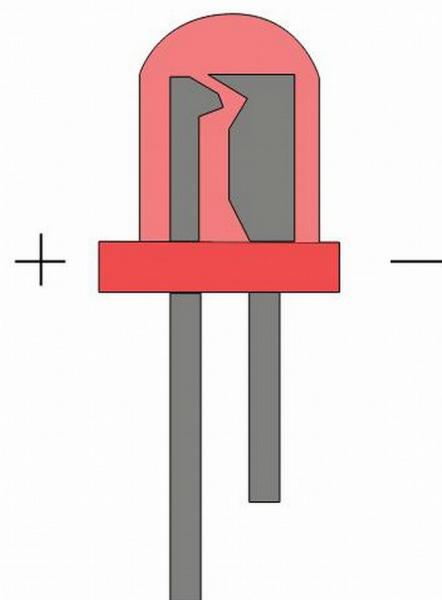
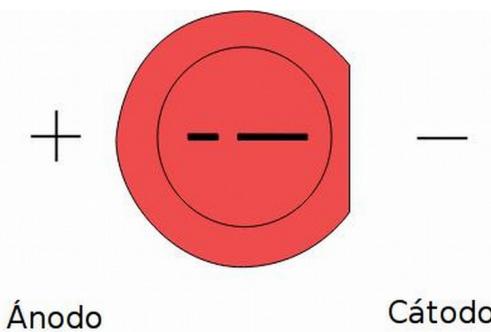
$$I = V / R$$



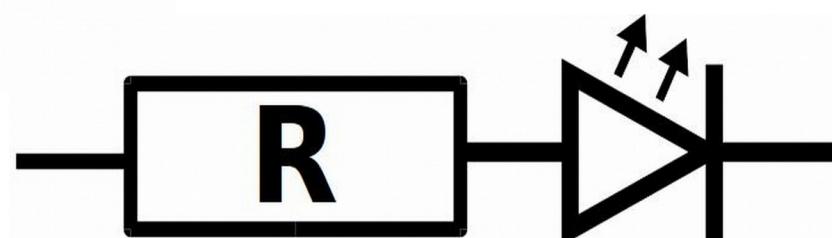
$$R = V / I$$

ELETTRONICA

43

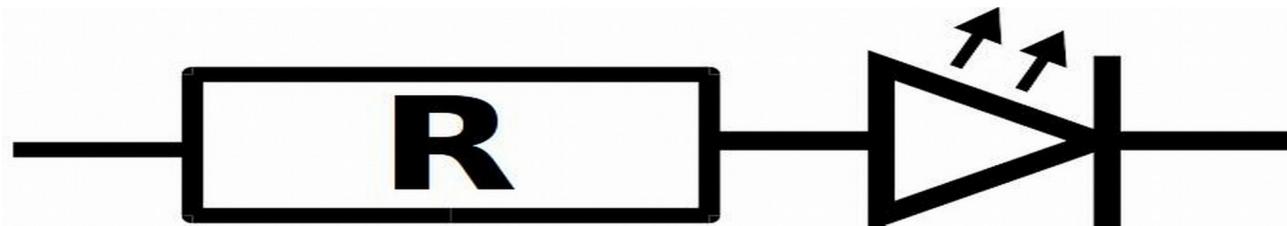


Símbolo del LED



ELETTRONICA

44



$$\Omega = V : A.$$

$$\Omega = (V - 1.5) / 0.016$$

dove:

Ω è il valore della resistenza

V è la tensione di uscita di Arduino o dell'alimentazione
In questo caso 5 Volt.

1.5 è la caduta di tensione dentro il LED

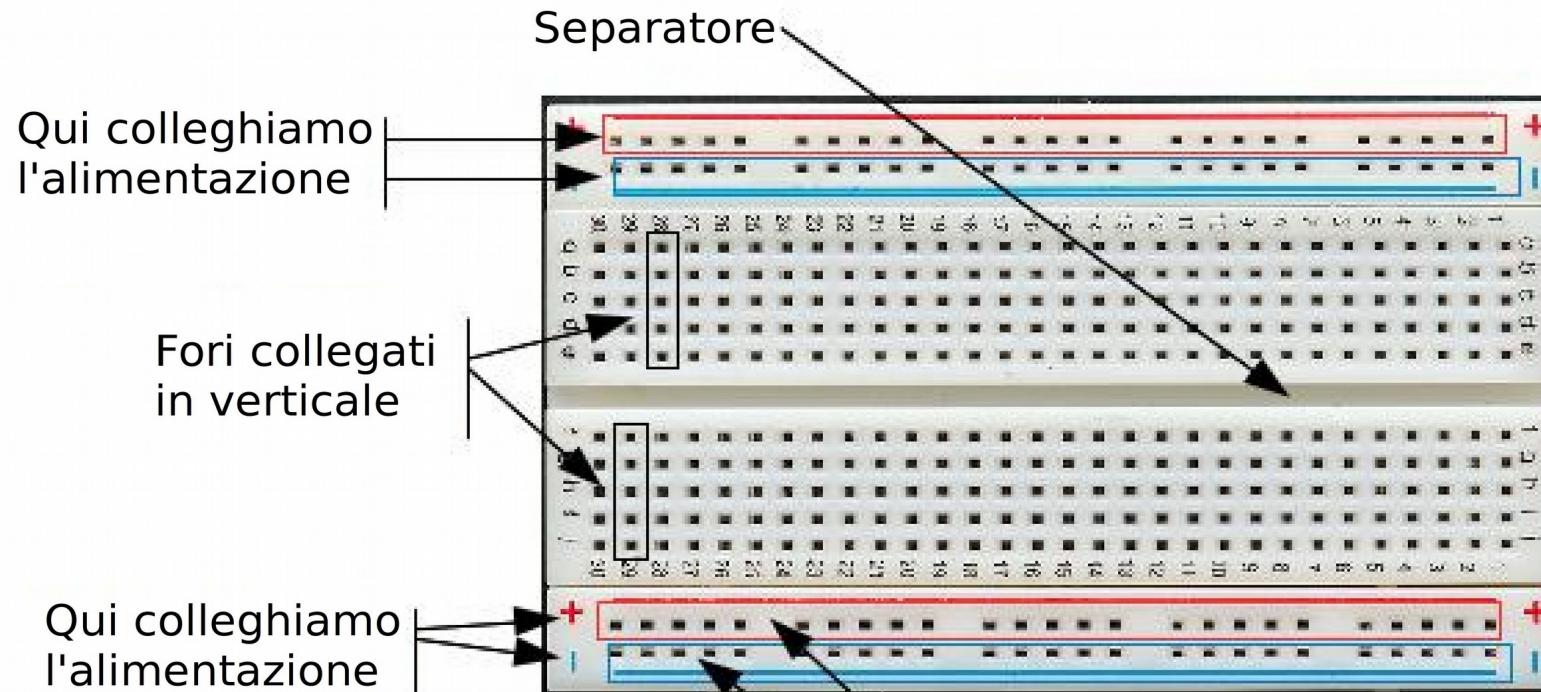
0,016 è la corrente in Amper che esige il LED

$$(5 - 1.5) / 0.016 = 218.75$$

Il valore della resistenza 218.75 Ω non esiste pertanto scegliamo il valore più vicino di **220 Ohmios**.

ELETTRONICA

45



ELETTRONICA

46

Resistenze con 4 anelli

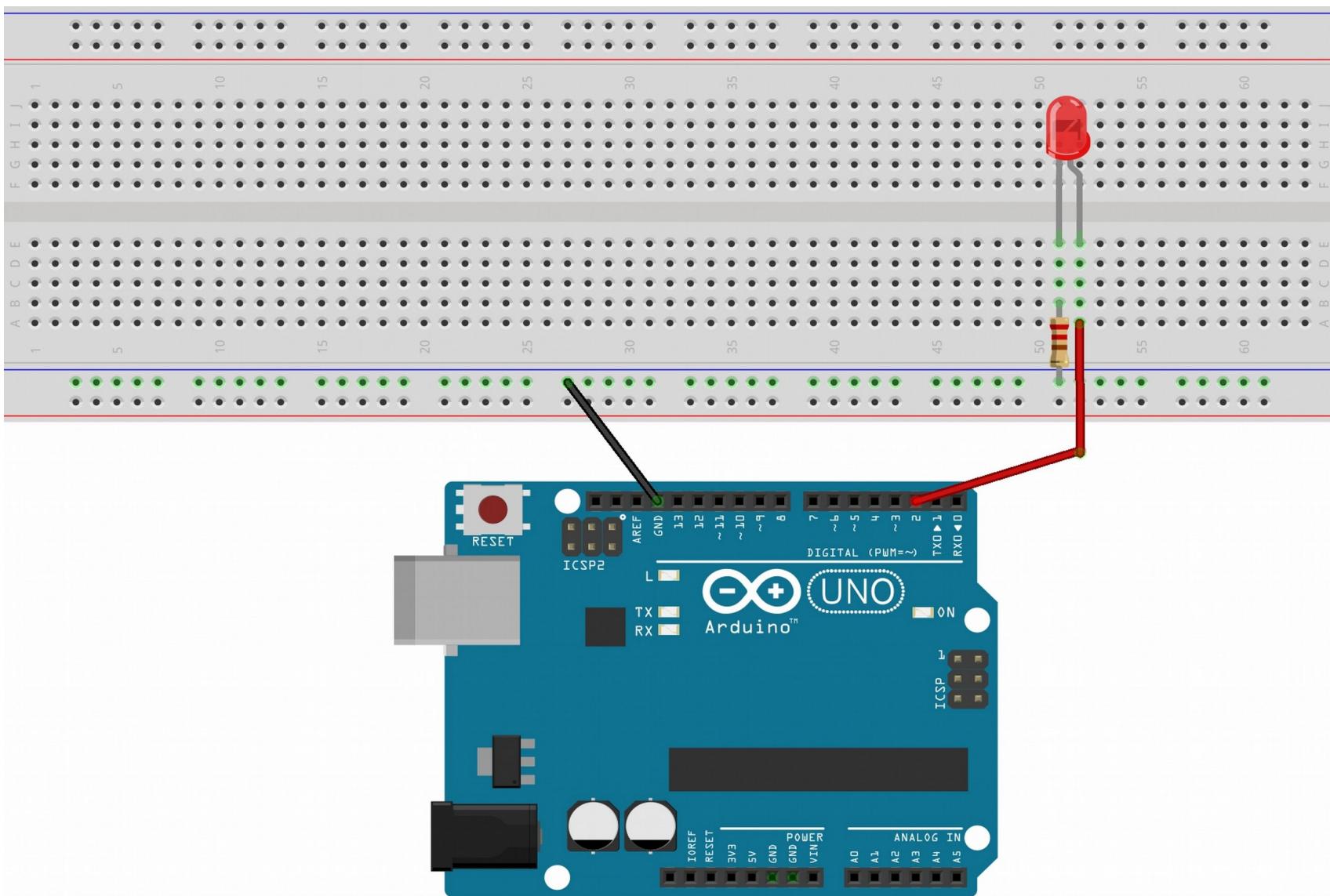
colore	1° anello	2° anello	3° anello	4° anello
	1° cifra	2° cifra	moltiplicatore	tolleranza
nero	.	0	x1	-
marrone	1	1	x10	-
rosso	2	2	x100	-
arancione	3	3	x1000 (1KΩ)	-
giallo	4	4	x10000 (10KΩ)	-
verde	5	5	x100000 (100KΩ)	-
blu	6	6	x1000000 (1MΩ)	-
viola	7	7	x10000000 (10MΩ)	-
grigio	8	8	x100000000	-
bianco	9	9	-	5%
oro	-	-	:10	10%
argento	-	-	:100	20%

Resistenze con 5 anelli

colore	1° anello	2° anello	3° anello	4° anello	5° anello
	1° cifra	2° cifra	3° cifra	moltiplicatore	tolleranza
nero	0	0	0	x1	-
marrone	1	1	1	x10	±1%
rosso	2	2	2	x100	±2%
arancione	3	3	3	x1000 (1KΩ)	-
giallo	4	4	4	x10000 (10KΩ)	-
verde	5	5	5	x100000 (100KΩ)	±0,5%
blu	6	6	6	x1000000 (1MΩ)	±0,25%
viola	7	7	7	x10000000 (10MΩ)	±0,1%
grigio	8	8	8	x100000000	±0,025%
bianco	9	9	9	x1000000000	-
oro	-	-	-	:10	±5%
argento	-	-	-	:100	±10%

ARDUINO

47



fritzing

ARDUINO

The image shows two screenshots of the Arduino IDE interface. Both screenshots feature a dark-themed interface with a light-colored code editor area.

Screenshot 1 (Top): The Arduino IDE with the Tools menu open. The menu bar includes File, Modifica, Sketch, Strumenti (selected), and Aiuto. The Tools menu contains several options:

- Formattazione automatica (Ctrl+T)
- Archivia sketch...
- Correggi codifica e ricarica
- Gestione librerie... (Ctrl+Maiusc+I)
- Monitor seriale (Ctrl+Maiusc+M)
- Plotter seriale (Ctrl+Maiusc+L)
- WiFi101 / WiFiNINA Firmware Updater
- Scheda: "Arduino/Genuino Uno"** (highlighted in blue)
 - Gestore schede...
 - Schede Arduino AVR
 - Arduino Yún
 - Arduino/Genuino Uno** (highlighted in blue)
 - Arduino Duemilanove or Diecimila
- Porta
- Acquisisci informazioni sulla scheda
- Programmatore: "AVRISP mkII"** (highlighted in blue)
 - Scrivi il bootloader

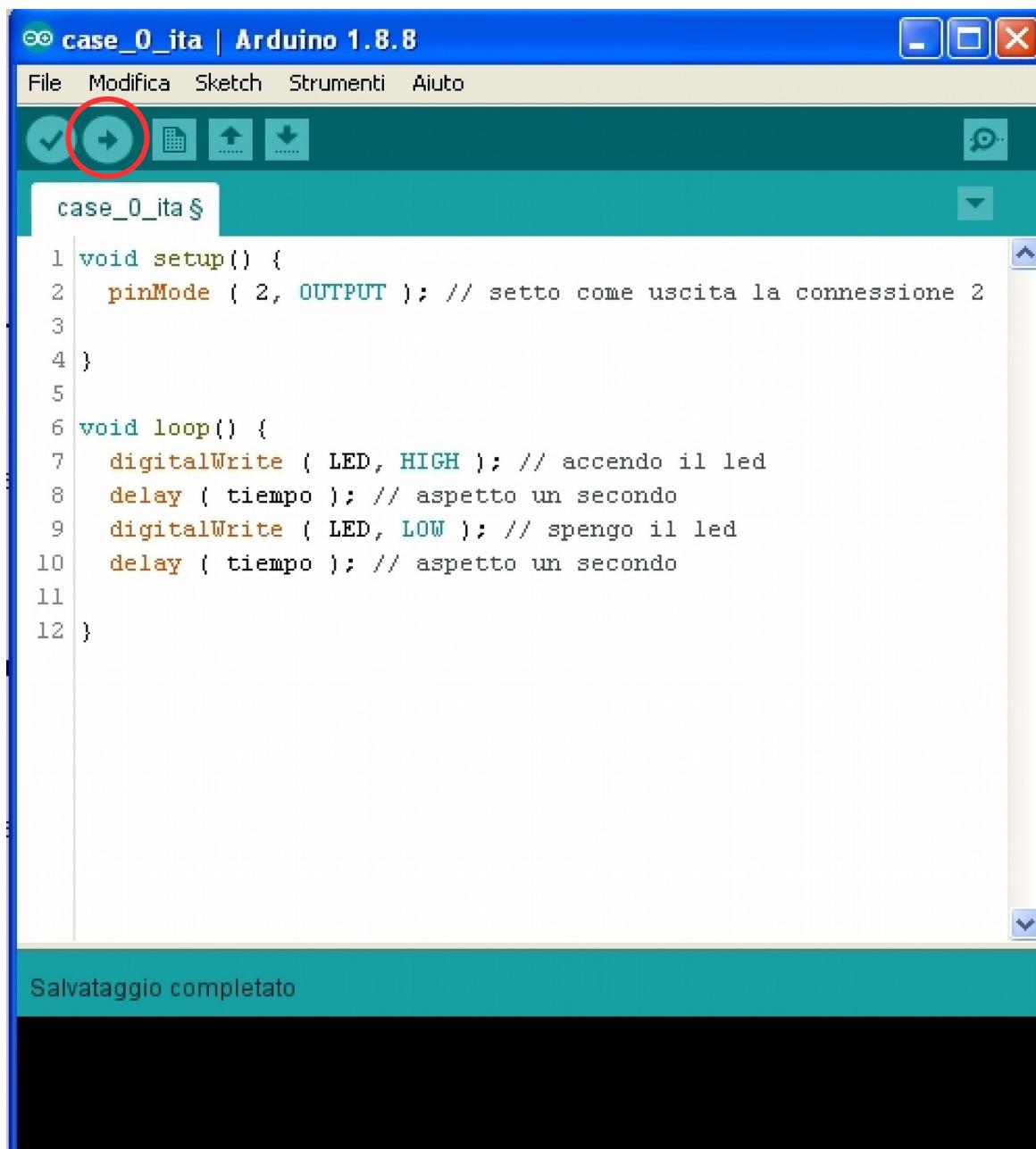
Screenshot 2 (Bottom): The Arduino IDE with the Tools menu open, specifically showing the Port selection. The menu bar and code editor are identical to the top screenshot.

The Tools menu shows the current port configuration:

- Porta: "COM13" (highlighted in blue)
 - Porte seriali
 - COM10
 - COM13** (highlighted in blue)
 - COM3

PROGRAMMAZIONE

49



PROGRAMMAZIONE

50

Le variabili sono:

byte

int

long

float

unsigned int

unsigned long

boolean

char

PROGRAMMAZIONE

52



The screenshot shows the Arduino IDE interface. At the top, there is a toolbar with icons for checkmark, refresh, file, upload, download, and search. Below the toolbar, the sketch name 'sketch_apr20a' is displayed. The main area contains the following C++ code:

```
1 void setup() {
2   pinMode(2,OUTPUT);      // setto come uscita la connessione 2
3
4 }
5
6 void loop() {
7   digitalWrite(2,HIGH); // accendo il led
8   delay(1000);          // aspetto un secondo
9   digitalWrite(2,LOW);  // spengo il led
10  delay(1000);         // aspetto un secondo
11
12 }
```

At the bottom left, the word 'Compilado' is visible.

PROGRAMACIÓN

53

```
1 int tempo = 1000; // memorizza la frequenza del lampeggio
2
3 void setup() {
4     pinMode ( 2, OUTPUT ); // setto come uscita la connessione 2
5
6 }
7
8 void loop() {
9     digitalWrite ( 2, HIGH ); // accendo il led
10    delay ( tempo ); // aspetto un secondo
11    digitalWrite ( 2, LOW ); // spengo il led
12    delay ( tempo ); // aspetto un secondo
13
14 }
```

PROGRAMMAZIONE

54

```
1 int tempo = 1000; // memorizza la frequenza del lampeggio
2 byte LED = 2; // la variabile LED memorizza la connessione del led
3
4 void setup() {
5     pinMode ( LED, OUTPUT ); // setto come uscita la connessione LED
6
7 }
8
9 void loop() {
10    digitalWrite ( LED, HIGH ); // accendo il led
11    delay ( tempo ); // aspetto un tempo
12    digitalWrite ( LED, LOW ); // spengo il led
13    delay ( tempo ); // aspetto un tempo
14
15 }
```

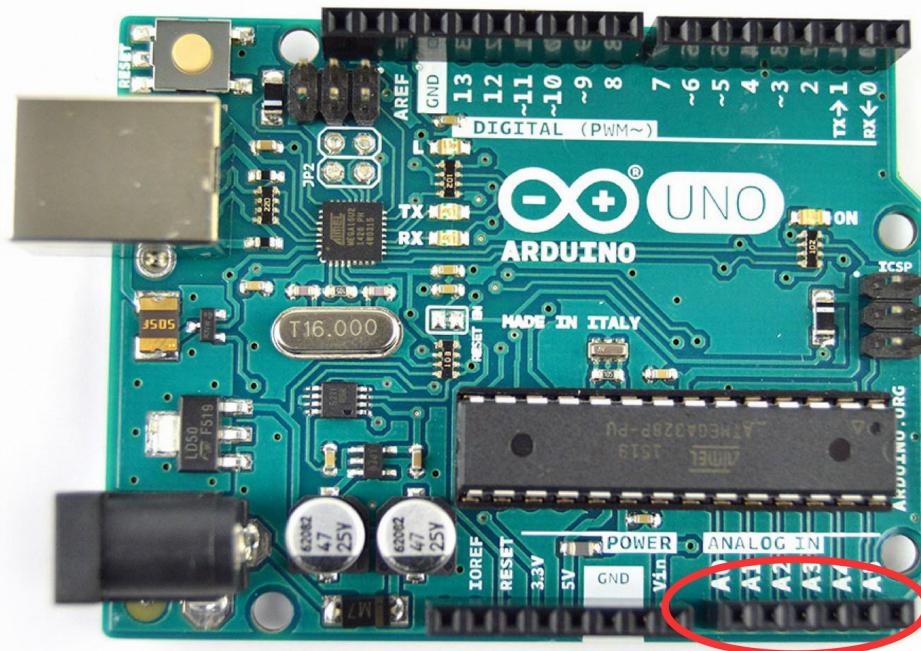
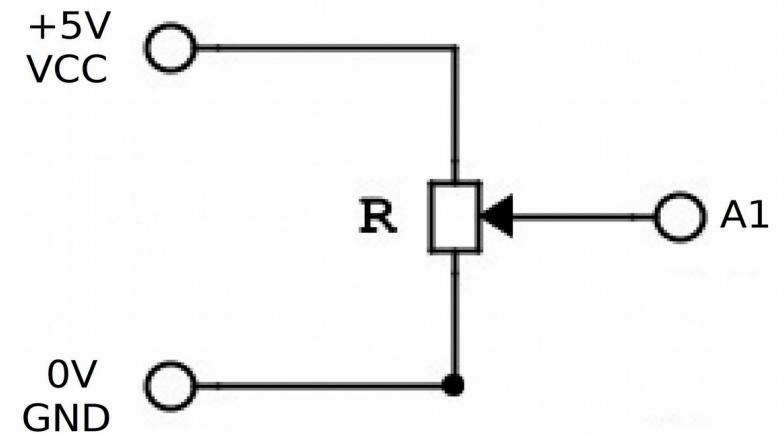
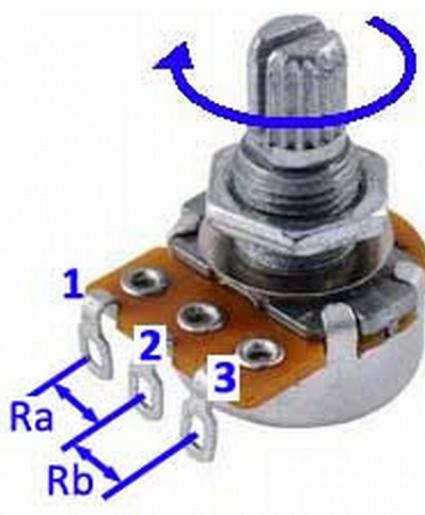
PROGRAMMAZIONE

55

```
1 #define LED 2 // è la connessione del led
2
3 int tempo = 1000; // memorizza la frequenza del lampeggio
4
5
6 void setup() {
7     pinMode ( LED, OUTPUT ); // setto come uscita la connessione LED
8
9 }
10
11 void loop() {
12     digitalWrite ( LED, HIGH ); // accendo il led
13     delay ( tempo ); // aspetto un tempo
14     digitalWrite ( LED, LOW ); // spengo il led
15     delay ( tempo ); // aspetto un tempo
16
17 }
```

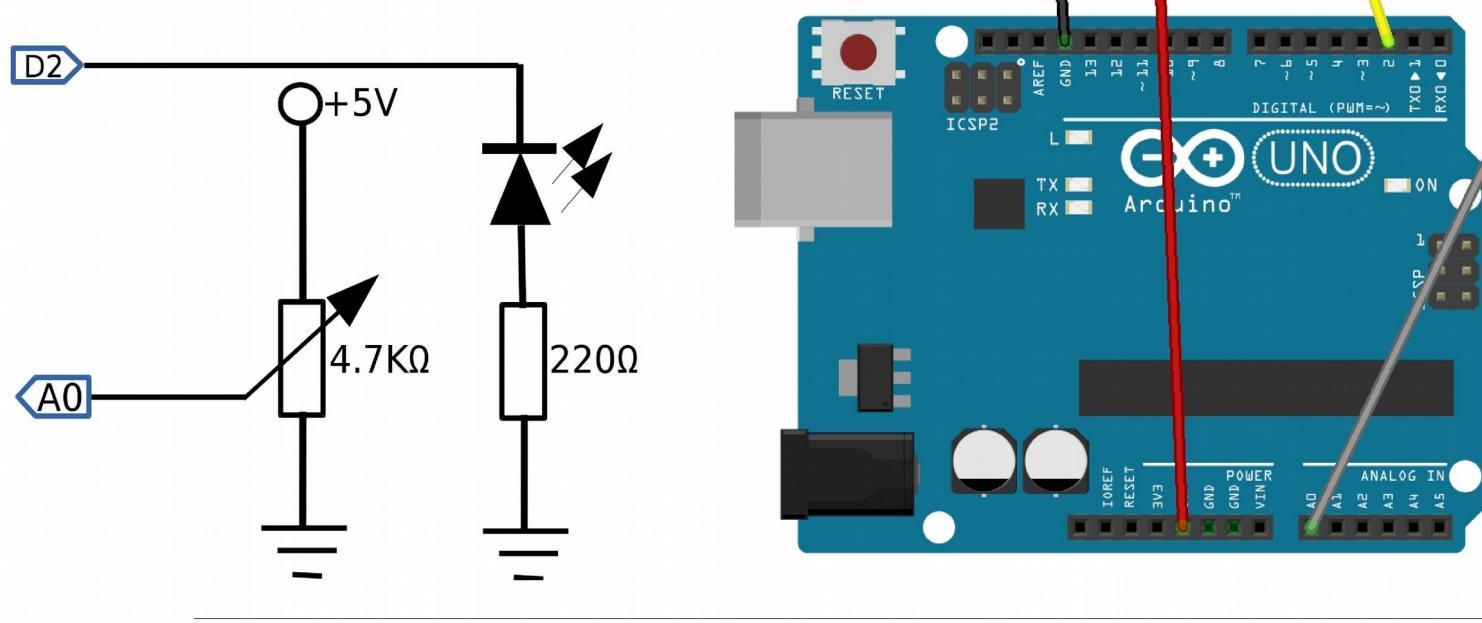
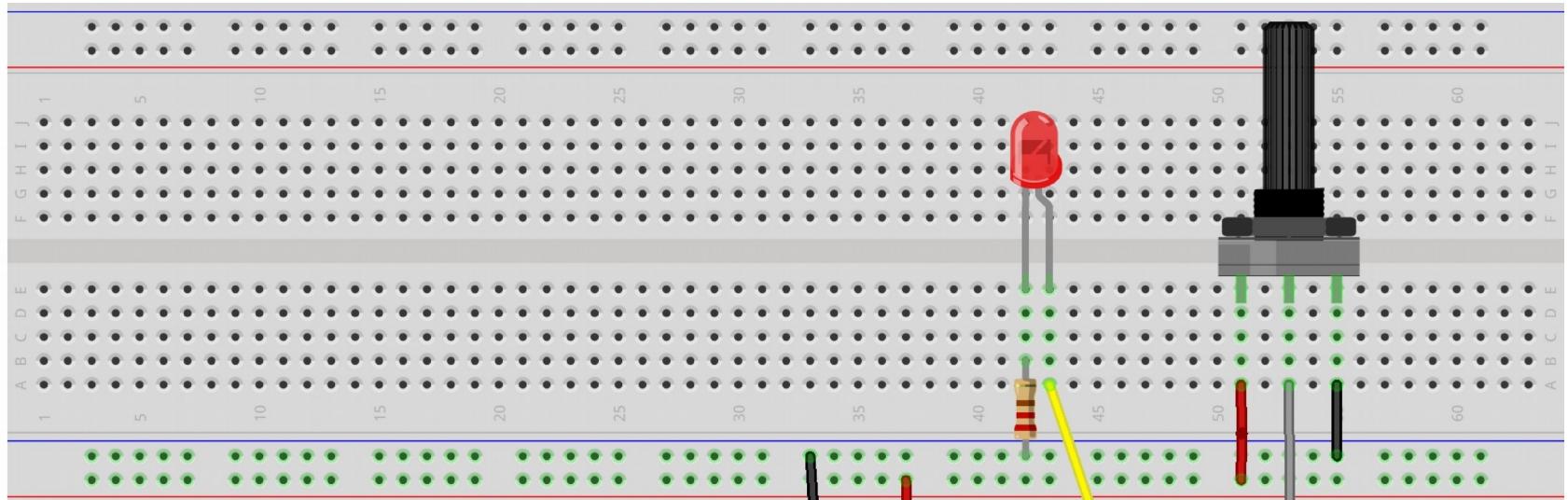
ELETTRONICA

58



ELETTRONICA

59



PROGRAMMAZIONE

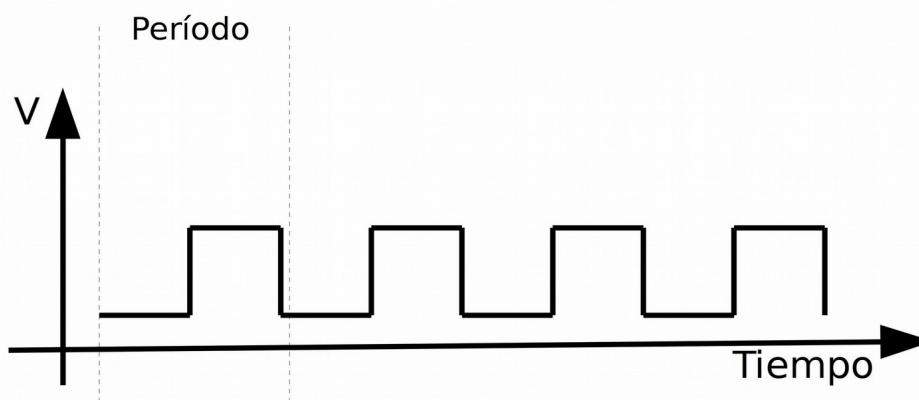
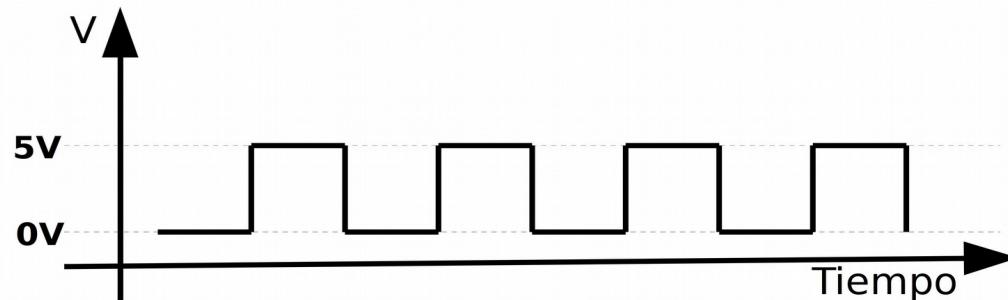
60

```
1 #define LED 2    // è la connessione del LED
2
3 int tempo = 1000; // memorizza la frequenza del lampeggio
4
5 void setup() {
6     pinMode ( 2, OUTPUT ); // setto come uscita la connessione 2
7 }
8
9 void loop() {
10    tempo = analogRead(0); // leggo il valore di A0 e lo metto in "tempo"
11    digitalWrite ( 2, HIGH ); // accendo il led
12    delay ( tempo ); // aspetto un secondo
13    digitalWrite ( 2, LOW ); // spengo il led
14    delay ( tempo ); // aspetto un secondo
15
16 }
```

ARDUINO

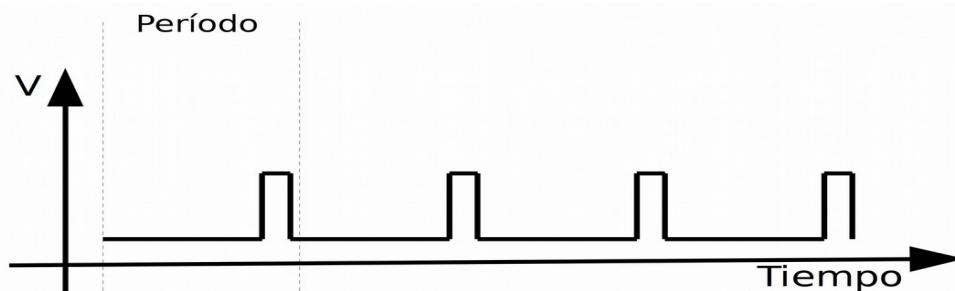
62

PWM

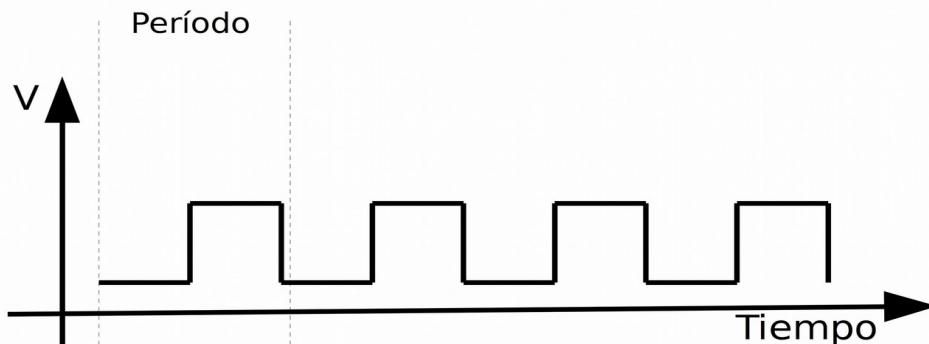


ARDUINO

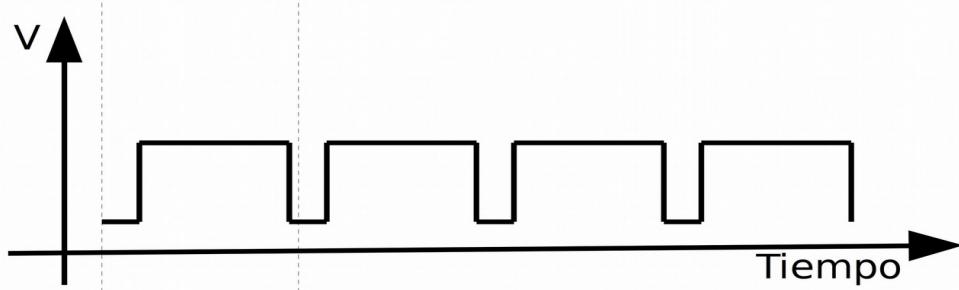
63



$\text{PWM} = 60$



$\text{PWM} = 128$



$\text{PWM} = 200$

analogWrite (uscita, valore);

PROGRAMMAZIONE

66

```
1 #define LED 3 // è la connessione del led
2 #define POTENZIOMETRO 0 // è la connessione del potenziometro
3
4 int valore = 0; // in "valore" memorizzo la lettura del potenziometro
5
6 void setup() {
7   pinMode ( LED, OUTPUT ); // configuro la connessione LED come uscita
8 }
9
10 void loop() {
11   valore = analogRead ( POTENZIOMETRO ); // in "valore" metto la lettura del potenziometro
12   valore = valore / 4; // divido la variabile
13   analogWrite ( LED, valore ); // scrivo sull'uscita analogica del LED il valore della variabile
14
15 }
```

PROGRAMMAZIONE

Il monitor seriale

68

Serial.begin (9600);

Apre la comunicazione

La velocità dei dati si misura in **Baud**

Serial.print ();

Scrive i dati

Serial.print (valore);

Scrive il valore

Serial.print ("Buon giorno");

Scrive il testo

Serial.println (valore); Scrive a capo

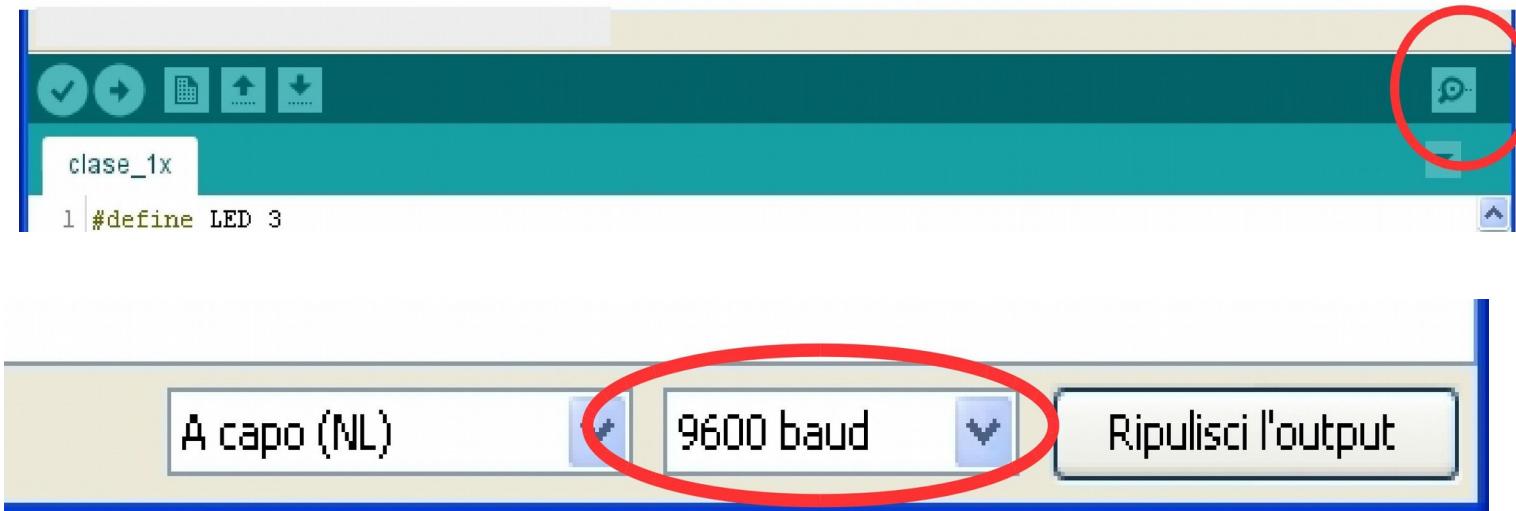
PROGRAMMAZIONE

69

```
1 #define LED 3 // è la connessione del led
2 #define POTENZIOMETRO 0 // è la connessione del potenziometro
3
4 int valore = 0; // in "valore" memorizzo la lettura del potenziometro
5
6 void setup() {
7     pinMode ( LED, OUTPUT ); // configuro la connessione LED come uscita
8     Serial.begin ( 9600 ); // attivo la porta di comunicazione
9 }
10
11 void loop() {
12     valore = analogRead ( POTENZIOMETRO ); // in "valore" metto la lettura del potenziometro
13     Serial.println ( valore ); // scrivo il valore sul monitor
14     delay ( 500 ); // aspetto mezzo secondo
15     valore = valore / 4; // divido la variabile
16     analogWrite ( LED, valore ); // scrivo sull'uscita analogica del LED il valore della variabile
17
18 }
```

PROGRAMMAZIONE

70



```
512  
503  
500  
500  
506  
518  
514  
506  
505  
512  
511  
503  
---
```

PROGRAMMAZIONE



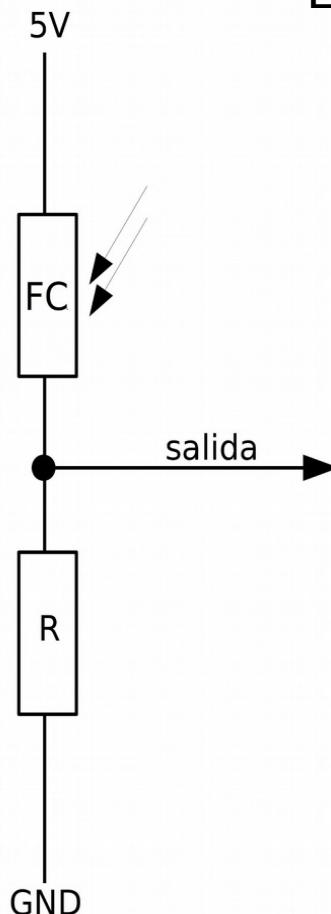
PROGRAMMAZIONE

72

```
1 #define LED 3 // è la connessione del led
2 #define POTENZIOMETRO 0 // è la connessione del potenziometro
3
4 int valore = 0; // in "valore" memorizzo la lettura del potenziometro
5 float VOLT = 0; // in volt memorizzo la tensione che ho nell'ingresso analogico
6 void setup() {
7     pinMode ( LED, OUTPUT ); // configuro la connessione LED come uscita
8     Serial.begin (9600); // attivo la porta di comunicazione
9 }
10
11 void loop() {
12     valore = analogRead ( POTENZIOMETRO ); // in "valore" metto la lettura del potenziometro
13     VOLT = valore * 0.0048; // moltiplico per il valore della risoluzione del convertitore
14     Serial.println ( VOLT ); // scrivo i volt sul monitor
15     delay ( 500 ); // aspetto mezzo secondo
16     valore = valore / 4; // divido la variabile
17     analogWrite ( LED, valore ); // scrivo sull'uscita analogica del LED il valore della variabile
18 }
19 }
```

ELETTRONICA

73



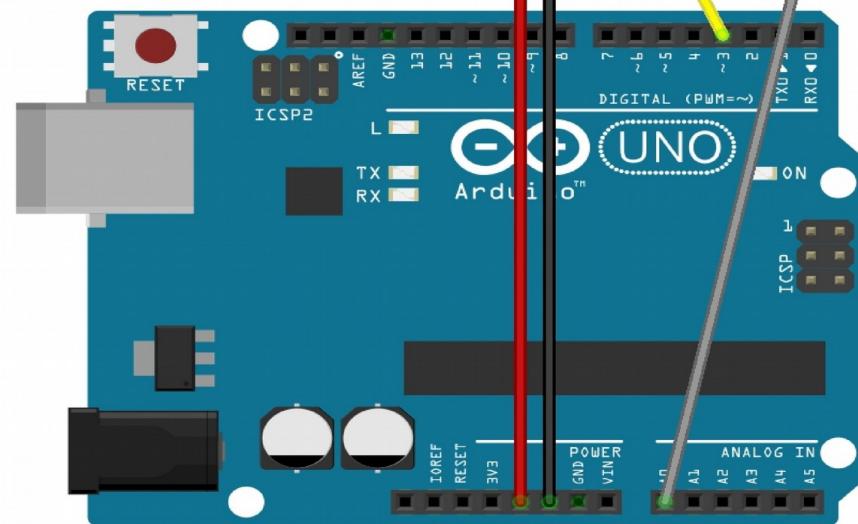
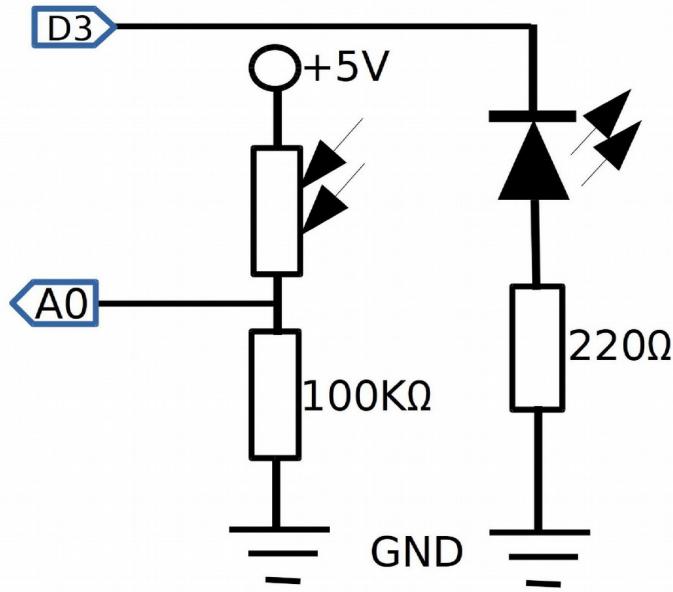
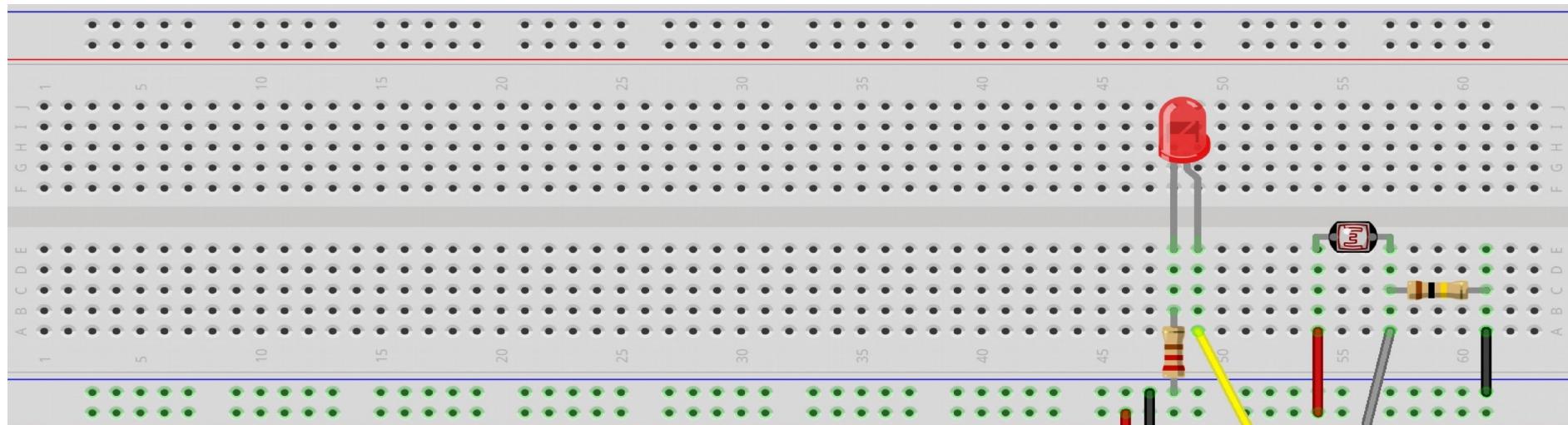
$$\text{Luce bassa } 5 \text{ V} * \frac{100,000}{1,000,000 + 100,000} = 0.45 \text{ V}$$

$$V_{\text{salida}} = V_{\text{entrada}} * \frac{R}{FC + R}$$

$$\text{Luce forte } 5 \text{ V} * \frac{100,000}{10 + 100,000} = 4.99 \text{ V}$$

ARDUINO

74



fritzing

PROGRAMMAZIONE

Il controllo di flusso

76

Arduino può decidere se un valore è:

- Maggiore >
- Minore <
- Uguale ==
- Diverso !=
- Maggiore o uguale >=
- Minore o uguale <=

```
If ( variabile == 300){  
    digitalWrite ( LED, HIGH);  
}
```

“Se il valore della variabile
è uguale a 300 spegni il LED”.

```
else {  
    digitalWrite ( LED, LOW );  
}
```

Se no accendi il LED”.

PROGRAMMAZIONE

78

```
1 #define LED 3
2 #define FOTORESISTENZA 0
3
4 int valore = 0; // in valore metto la lettura della fotoresistenza
5 void setup() {
6     pinMode ( LED, OUTPUT ); // configuro la connessione LED come uscita
7
8 }
9
10 void loop() {
11     valore = analogRead ( FOTORESISTENZA ); // in "valore" metto la lettura della fotoresistenza
12     valore = valore / 4; // divido la variabile
13     if ( valore < 100 ) { // se il valore della variabile "valore" è minore di 100
14         valore = 100; // metto a 100 la variabile
15     }
16     analogWrite ( LED, valore ); // scrivo sull'uscita analogica del LED il valore della variabile
17
18 }
```

PROGRAMMAZIONE

79

```
1 #define LED 3
2 #define FOTORESISTENZA 0
3
4 int valore = 0; // in valore metto la lettura della fotoresistenza
5 void setup() {
6     pinMode ( LED, OUTPUT ); // configuro la connessione LED come uscita
7
8 }
9
10 void loop() {
11     valore = analogRead ( FOTORESISTENZA ); // in "valore" metto la lettura della fotoresistenza
12
13     if ( valore < 500 ) { // se il valore della variabile "valore" è minore di 500
14         digitalWrite ( LED, HIGH ); // accendo il LED
15     }
16     else { // se no
17         digitalWrite ( LED, LOW ); // spengo il LED
18     }
19
20 }
```

PROGRAMMAZIONE

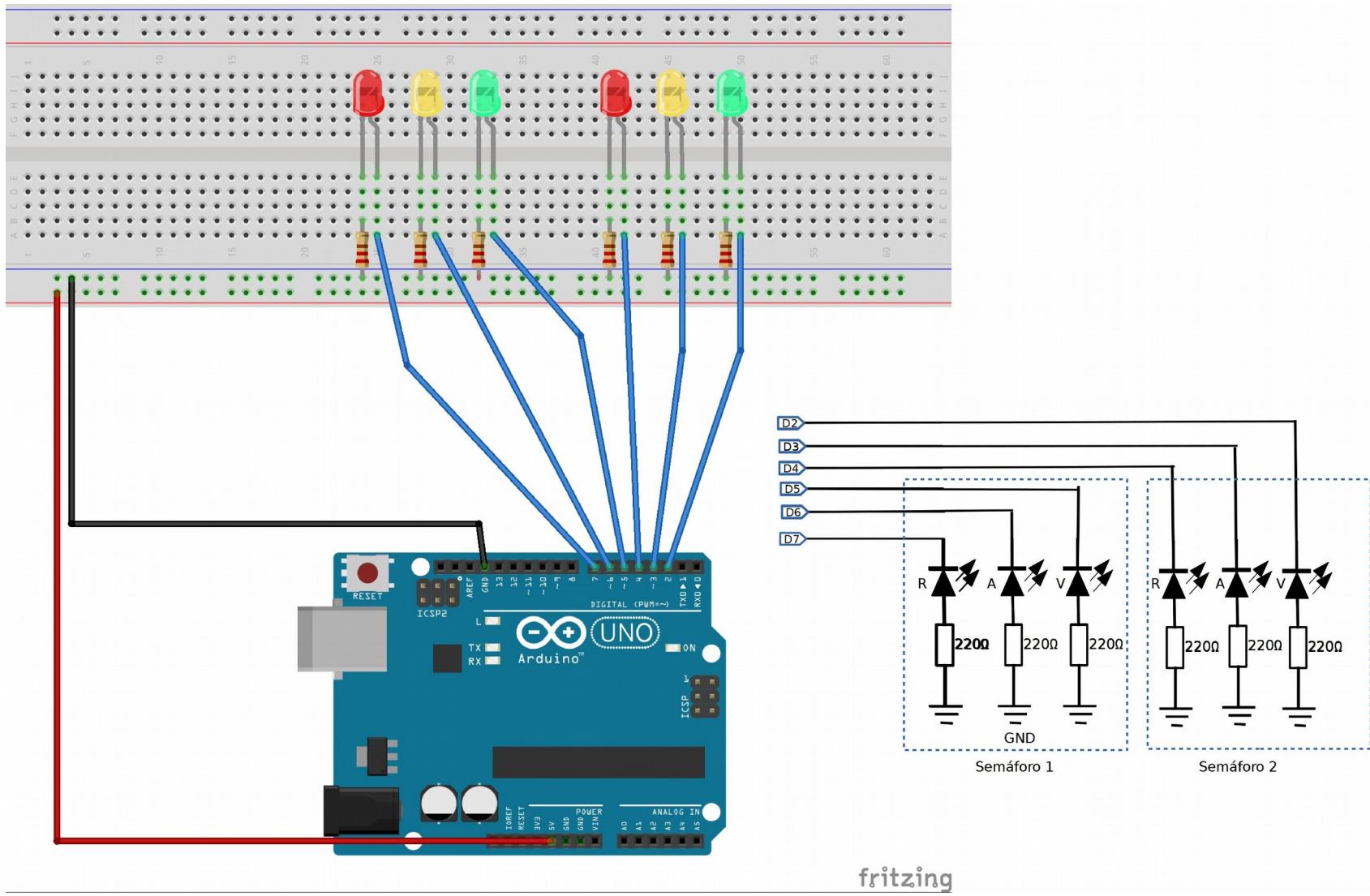
Il controllo di flusso

80

```
for ( int i=0; i<10; i++){  
    // istruzioni  
}
```

PROGRAMMAZIONE

82



fritzing

PROGRAMMAZIONE

83

```
1 #define LED_verde_1 2      // collego il LED verde del semaforo 1 a la connessione 2
2 #define LED_giallo_1 3    // collego il LED giallo del semaforo 1 a la connessione 3
3 #define LED_rosso_1 4     // collego il LED rosso del semaforo 1 a la connessione 4
4 #define LED_verde_2 5     // collego il LED verde del semaforo 2 a la connessione 5
5 #define LED_giallo_2 6    // collego il LED giallo del semaforo 2 a la connessione 6
6 #define LED_rosso_2 7     // collego il LED rosso del semaforo 2 a la connessione 7
7
```

```
7
8 void setup() {
9   pinMode ( LED_verde_1, OUTPUT );    // imposto come uscita la connessione del LED_verde_1
10  pinMode ( LED_giallo_1, OUTPUT ); // imposto come uscita la connessione del LED_giallo_1
11  pinMode ( LED_rosso_1, OUTPUT );   // imposto come uscita la connessione del LED_rosso_1
12  pinMode ( LED_verde_2, OUTPUT );   // imposto come uscita la connessione del LED_verde_2
13  pinMode ( LED_giallo_2, OUTPUT ); // imposto come uscita la connessione del LED_giallo_2
14  pinMode ( LED_rosso_2, OUTPUT );   // imposto come uscita la connessione del LED_rosso_2
15
16 }
17
```

```
17
18 void loop() {
19   digitalWrite( LED_rosso_1,HIGH); // accendo il rosso del semaforo 1
20   digitalWrite( LED_rosso_2,HIGH); // accendo il rosso del semaforo 2
21   delay(5000);                  // aspetto 5 secondi
22   digitalWrite( LED_rosso_1,LOW); // spengo il rosso del semaforo 1
23   digitalWrite( LED_verde_1,HIGH); // accendo il verde del semaforo 1
24   delay(15000);                 // aspetto 15 secondi
```

PROGRAMMAZIONE

84

```
25  for (int i=0; i<4; i++){          // spengo e accendo il verde del semaforo 1 quattro volte
26      digitalWrite(LED_verde_1,LOW);
27      delay(250);
28      digitalWrite(LED_verde_1,HIGH);
29      delay (250);
30  }

31  digitalWrite( LED_verde_1,LOW); // spengo il verde del semaforo 1
32  digitalWrite( LED_giallo_1,HIGH); // accendo il giallo del semaforo 1
33  delay(2000);                  // aspetto 2 secondi
34  digitalWrite( LED_giallo_1,LOW); // spengo il giallo del semaforo 1
35  digitalWrite( LED_rosso_1,HIGH); // accendo il rosso del semaforo 1
36  delay (5000);                // aspetto 5 secondo
37  digitalWrite( LED_rosso_2,LOW); // spengo il rosso del semaforo 2
38  digitalWrite( LED_verde_2,HIGH); // accendo il verde del semaforo 2
39  delay(15000);                // aspetto 15 secondo
40  for (int i=0; i<4; i++){      // spengo e accendo il verde del semaforo 2 quattro volte
41      digitalWrite(LED_verde_2,LOW);
42      delay(250);
43      digitalWrite(LED_verde_2,HIGH);
44      delay (250);
45  }
46  digitalWrite( LED_verde_2,LOW); // spengo il verde del semaforo 2
47  digitalWrite( LED_giallo_2,HIGH); // accendo il giallo del semaforo 2
48  delay(2000);                  // aspetto 2 secondi
49  digitalWrite( LED_giallo_2,LOW); // spengo il led giallo del semaforo 2
50
51 }
```