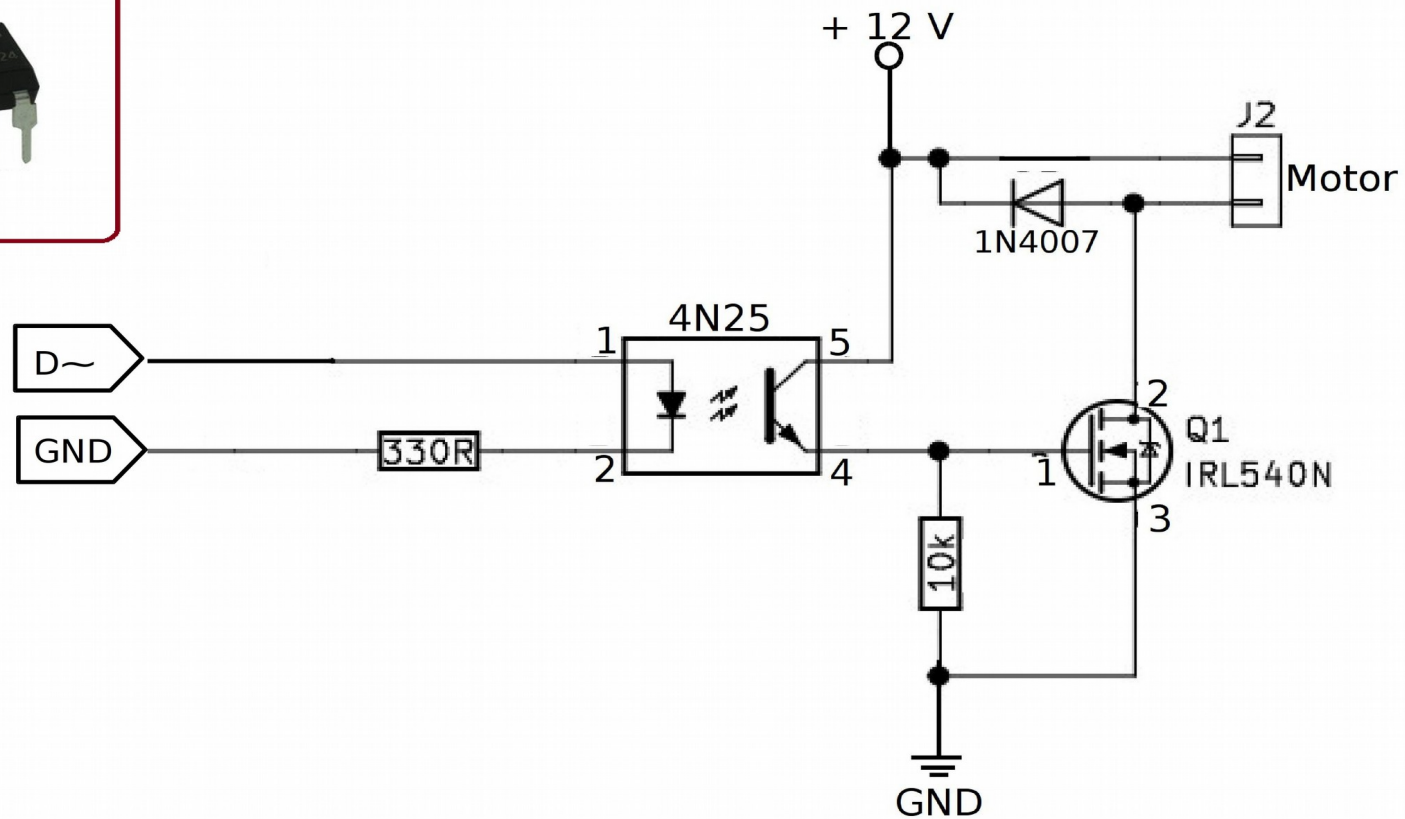
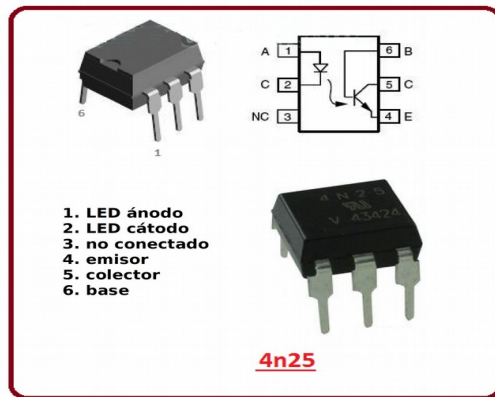


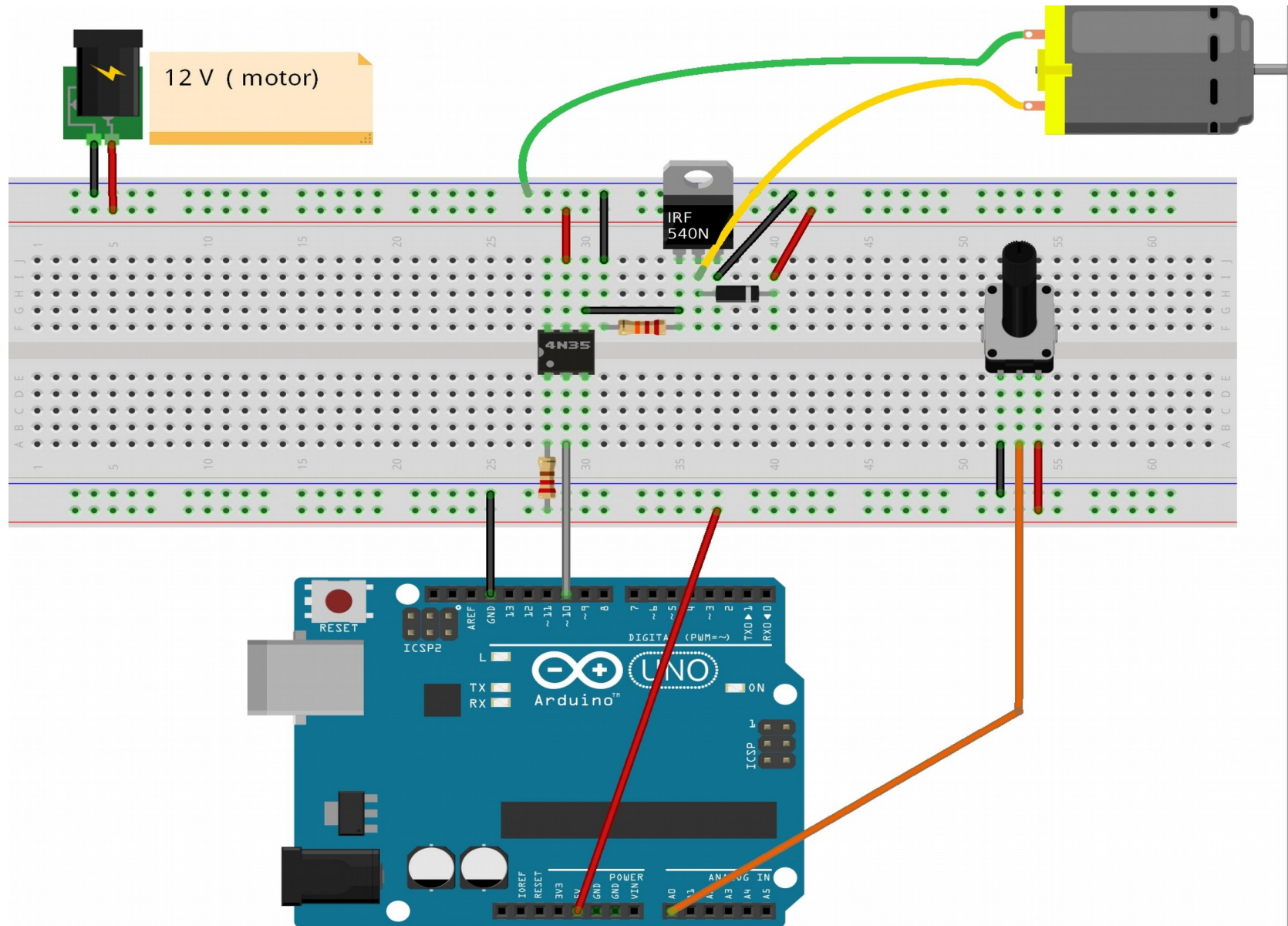
## Il foto accoppiatore

155



# ARDUINO

157



# ARDUINO

## IL PWM SEGUE:

158

| Timer | Frequenza di base | Comparatore di uscita |      | Uscita di Arduino | Frequenza automatica |
|-------|-------------------|-----------------------|------|-------------------|----------------------|
| TCCR0 | 62,500 Hz         | OC0                   | OC0A | 6                 | 976 Hz               |
|       | 62,500 Hz         |                       | OC0B | 5                 | 976 Hz               |
| TCCR1 | 31,250 Hz         | OC1                   | OC1A | 9                 | 488 Hz               |
|       | 31,250 Hz         |                       | OC1B | 10                | 488 Hz               |
| TCCR2 | 31,250 Hz         | OC2                   | OC2A | 11                | 488 Hz               |
|       | 31,250 Hz         |                       | OC2B | 3                 | 488 Hz               |
|       |                   |                       |      |                   |                      |

Esiste il modo per cambiare la frequenza del PWM che si scrive nel “setup()”:

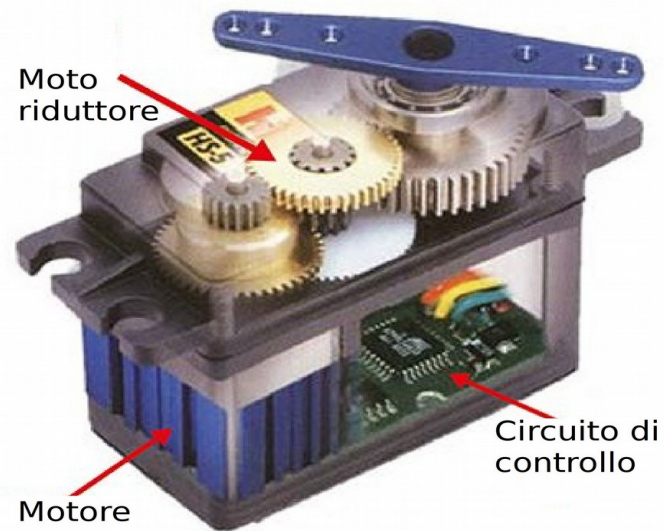
```
setPwmFrequency(10,8);
```

Il timer zero è anche utilizzato dalla funzione “millis()” e “delay()” e da alcune librerie come la “Servo” ed è meglio non cambiarlo.

# ARDUINO

## Il servomotore

159



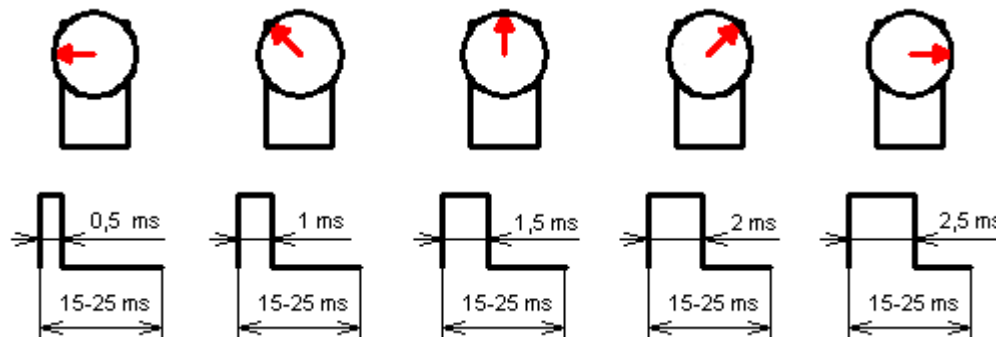
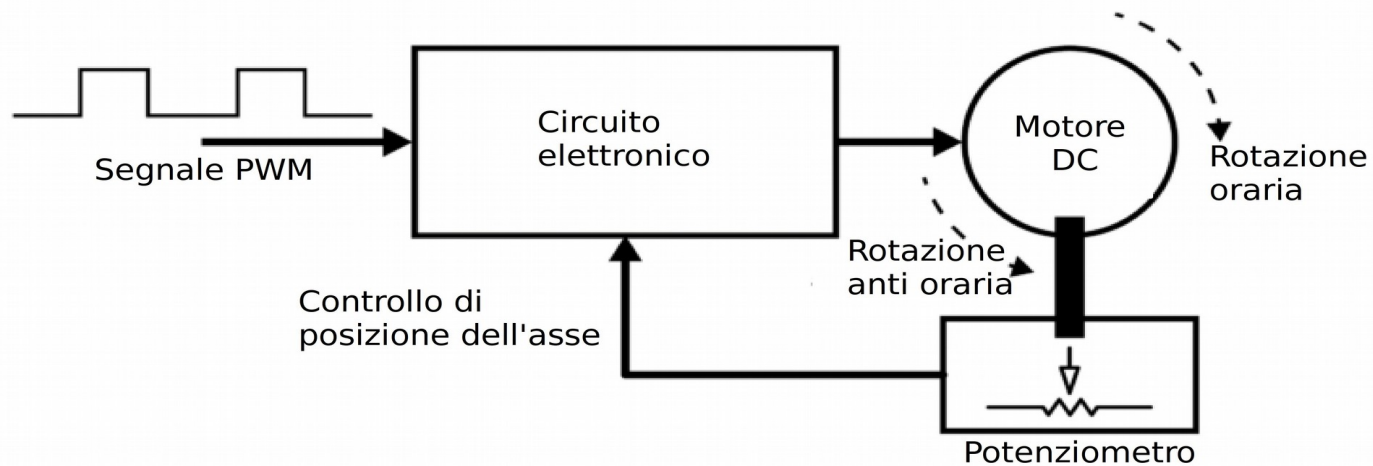
| Positivo di alimentazione | Negativo di alimentazione | Segnale di controllo         |
|---------------------------|---------------------------|------------------------------|
| Red                       | Black                     | Brown, Yellow, White, Orange |

# ARDUINO

## Il servomotore

160

DIAGRAMMA A BLOCCHI DI UN SERVO MOTORE



# PROGRAMMAZIONE

161

La classe “Servo” si trova nelle librerie del IDE e si carica come una libreria normale.

Questa classe aggiunge alcune funzioni che ci facilitano la vita:

**attach()** nella parentesi scriviamo dove è collegato il servomotore

**attached()** controlla se un servomotore è collegato

**detach()** scollega un servomotore

**read()** legge la posizione angolare del servomotore scritta col “write”

**write()** scrive la posizione del servomotore

**writeMicrosecond()** è la velocità del servomotore da 1,000 a 2,000

# PROGRAMMAZIONE

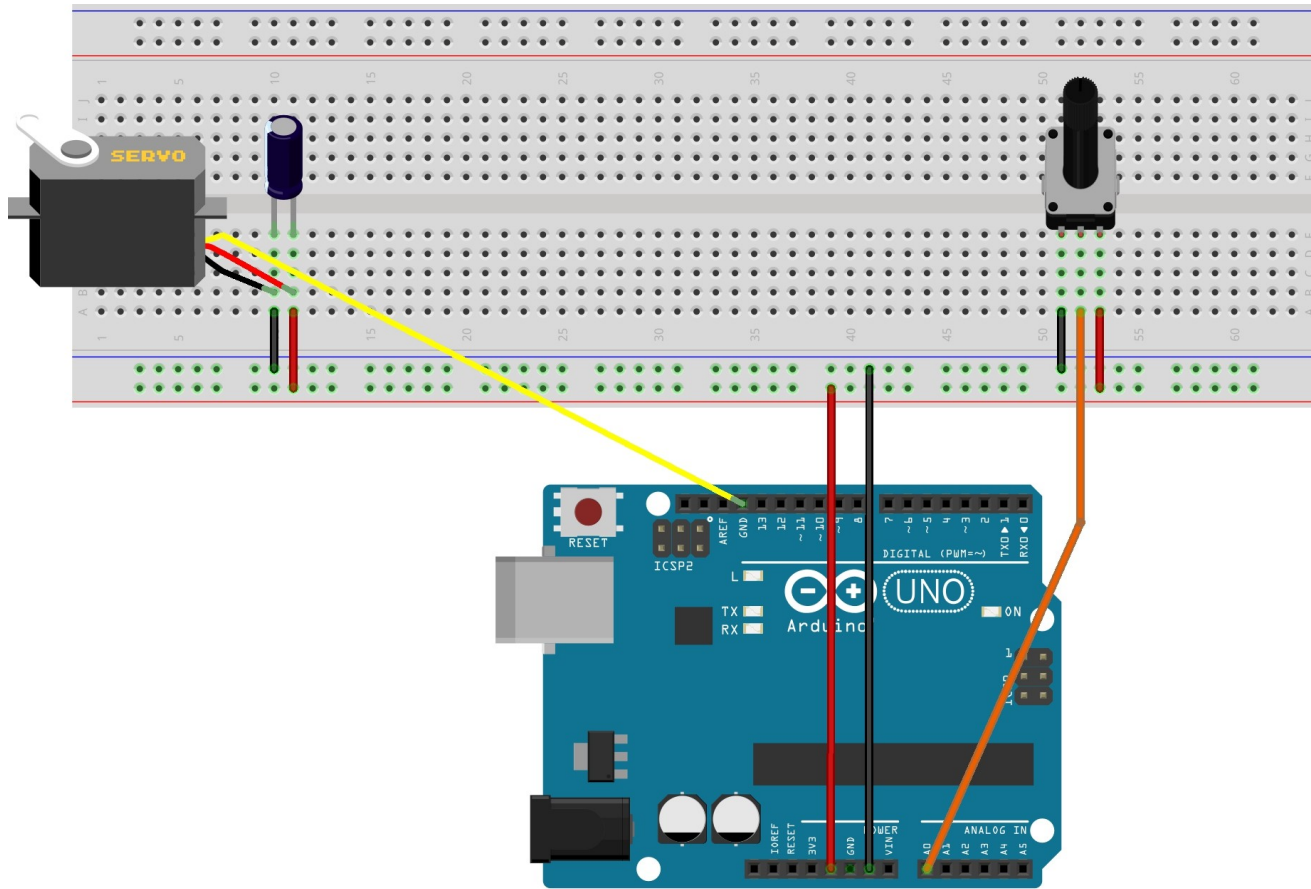
161

```
1 #include <Servo.h>
2 Servo servomotore1; // nomino il mio servo "servomotore1"
3 int gradi=0;        // memorizzo i gradi dell'asse del servomotore
4
5 void setup() {
6   servomotore1.attach(11); // scrivo dove ho collegato il servomotore1
7
8 }
9
10 void loop() {
11   gradi=analogRead(0)/3.8; // leggo il potenziometro e divido per 3.8 per sapere i gradi
12                           // di giro del potenziometro
13   gradi = gradi - 45;      // tolgo 45 gradi
14   if (gradi < 0) gradi=0; // se i gradi sono minori di zero metto a zero i gradi
15   if (gradi > 180) gradi = 180; // se i gradi sono maggiori di 180 metto a 180 i gradi
16   servomotore1.write(gradi); // scrivo i gradi nel servomotore1
17   delay(15); // aspetto un tempo per permettere al servomotore di posizionarsi
18
19 }
```



# PROGRAMMAZIONE

163



fritzing



## I CONDENSATORI

164

La capacità di carica dei condensatori si misura in Farad che si scrive con una "F" maiuscola.

micro Farad (**μF**) = 1F / 1,000,000

nano Farad (**nF**) = 1F / 1,000,000,000

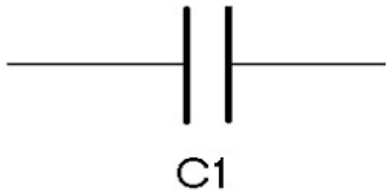
pico Farad (**pF**) = 1F / 1,000,000,000,000

**pF** \* 1000 = **nF**

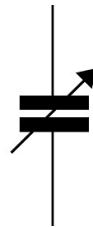
**nF** \* 1000 = **μF**

**μF** / 1000 = **nF**

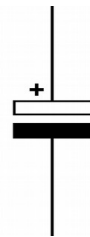
**nF** / 1000 = **pF**



Simbolo elettrico del  
condensatore non polarizzato



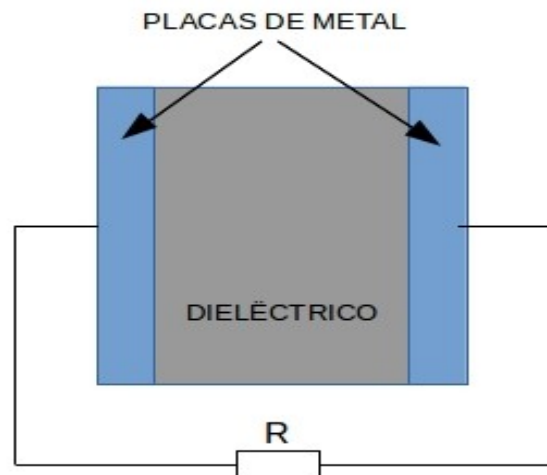
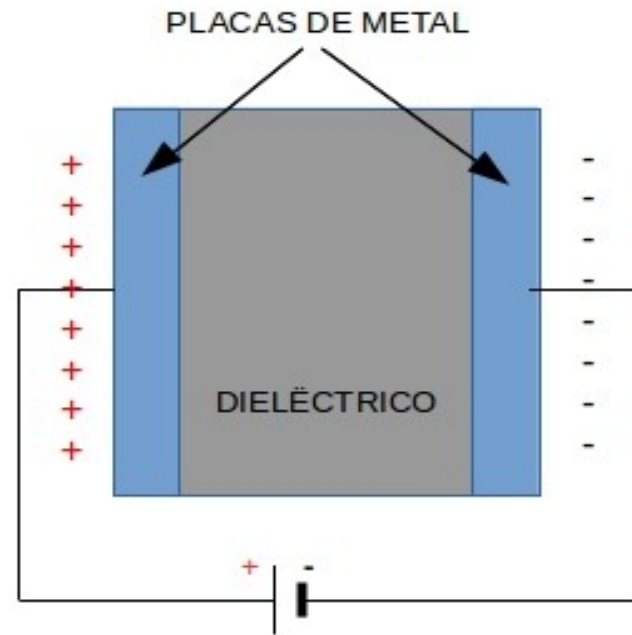
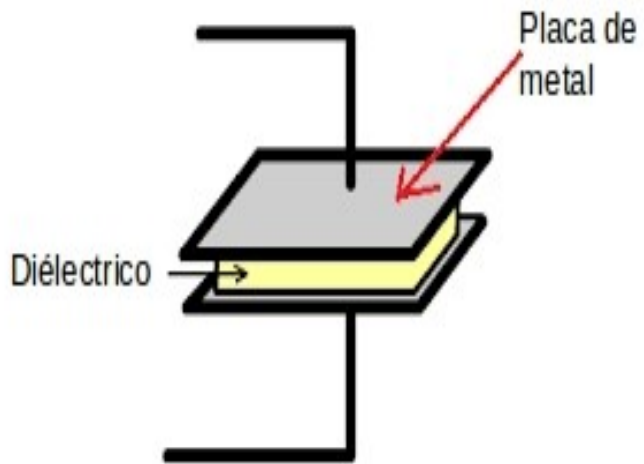
Simbolo elettrico del  
condensatore variabile



Simbolo elettrico del  
condensatore elettrolitico

# ELETRONICA

165



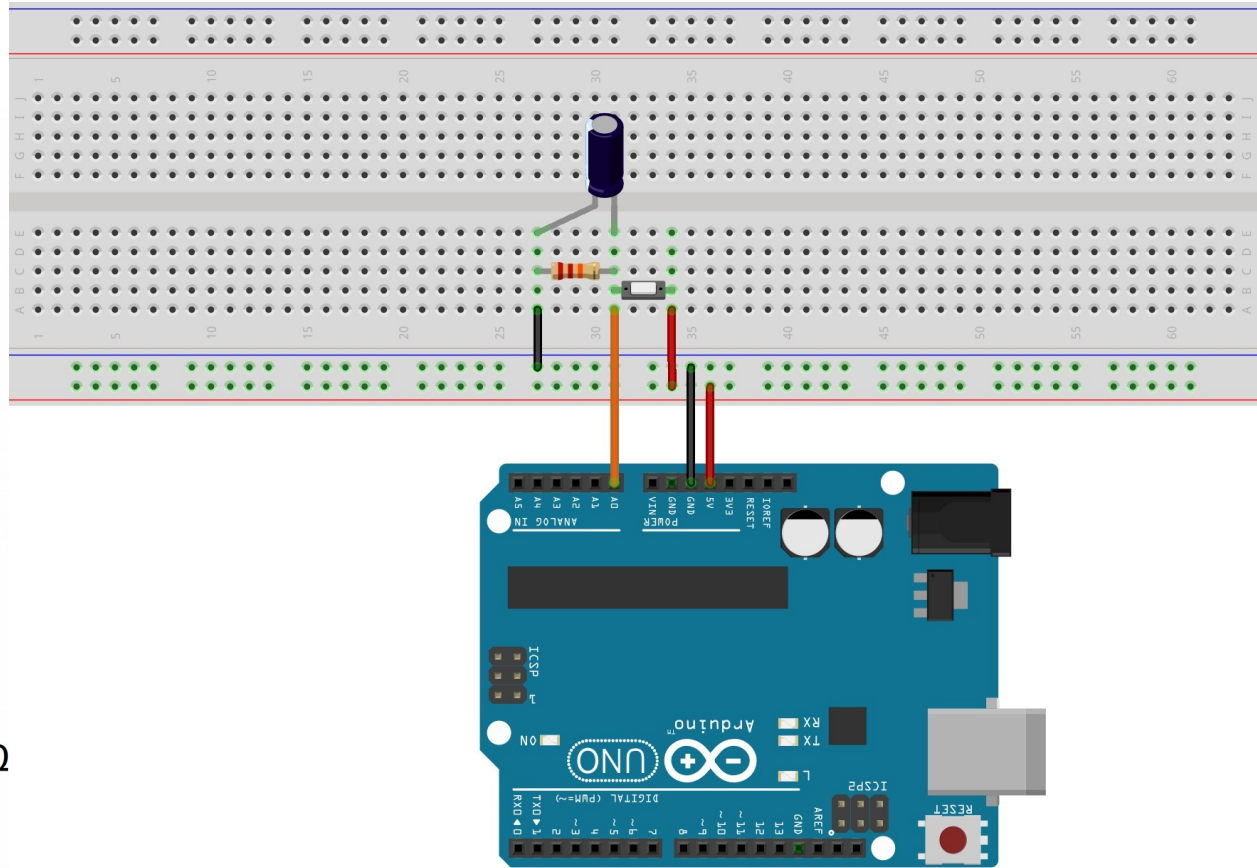
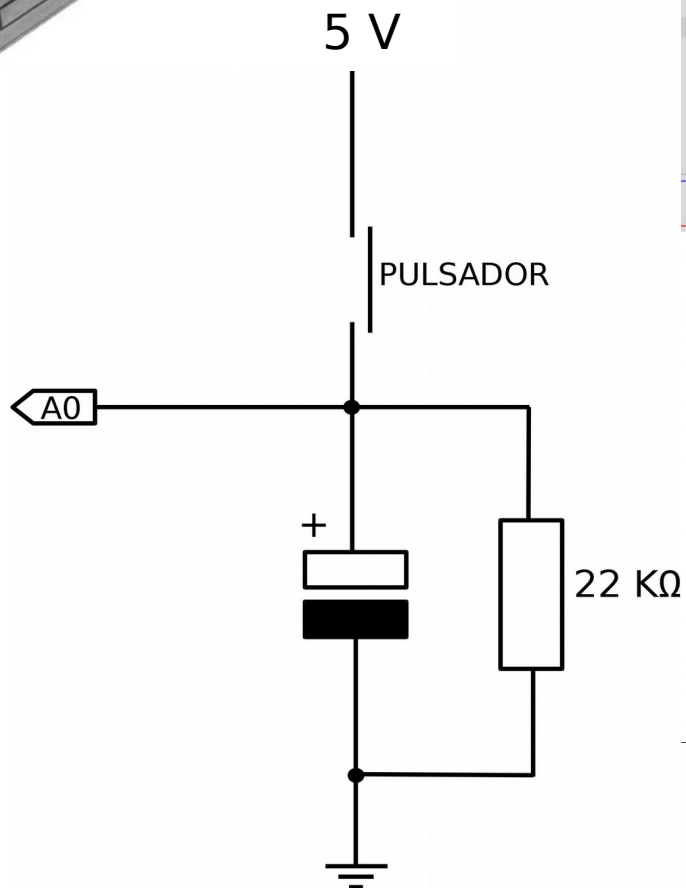
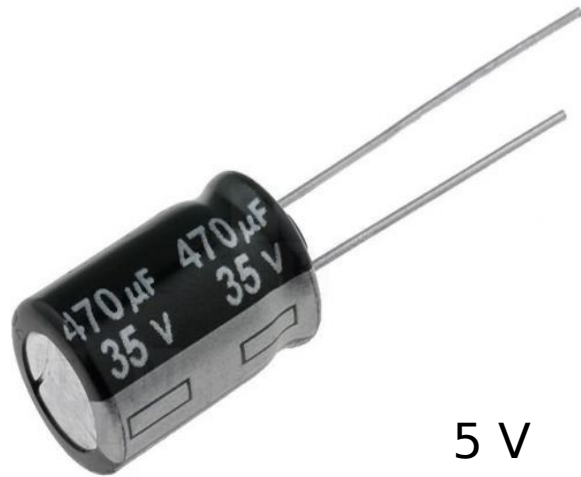
# ELETRONICA

166

|        |       |        |          |           |            |            |
|--------|-------|--------|----------|-----------|------------|------------|
| 1,0 pF | 10 pF | 100 pF | 1.000 pF | 10.000 pF | 100.000 pF | 1,0 microF |
| 1,2 pF | 12 pF | 120 pF | 1.200 pF | 12.000 pF | 120.000 pF | 1,2 microF |
| 1,5 pF | 15 pF | 150 pF | 1.500 pF | 15.000 pF | 150.000 pF | 1,5 microF |
| 1,8 pF | 18 pF | 180 pF | 1.800 pF | 18.000 pF | 180.000 pF | 1,8 microF |
| 2,2 pF | 22 pF | 220 pF | 2.200 pF | 22.000 pF | 220.000 pF | 2,2 microF |
| 2,7 pF | 27 pF | 270 pF | 2.700 pF | 27.000 pF | 270.000 pF | 2,7 microF |
| 3,3 pF | 33 pF | 330 pF | 3.300 pF | 33.000 pF | 330.000 pF | 3,3 microF |
| 3,9 pF | 39 pF | 390 pF | 3.900 pF | 39.000 pF | 390.000 pF | 3,9 microF |
| 4,7 pF | 47 pF | 470 pF | 4.700 pF | 47.000 pF | 470.000 pF | 4,7 microF |
| 5,6 pF | 56 pF | 560 pF | 5.600 pF | 56.000 pF | 560.000 pF | 5,6 microF |
| 6,8 pF | 68 pF | 680 pF | 6.800 pF | 68.000 pF | 680.000 pF | 6,8 microF |
| 8,2 pF | 82 pF | 820 pF | 8.200 pF | 82.000 pF | 820.000 pF | 8,2 microF |

# ELETRONICA

165



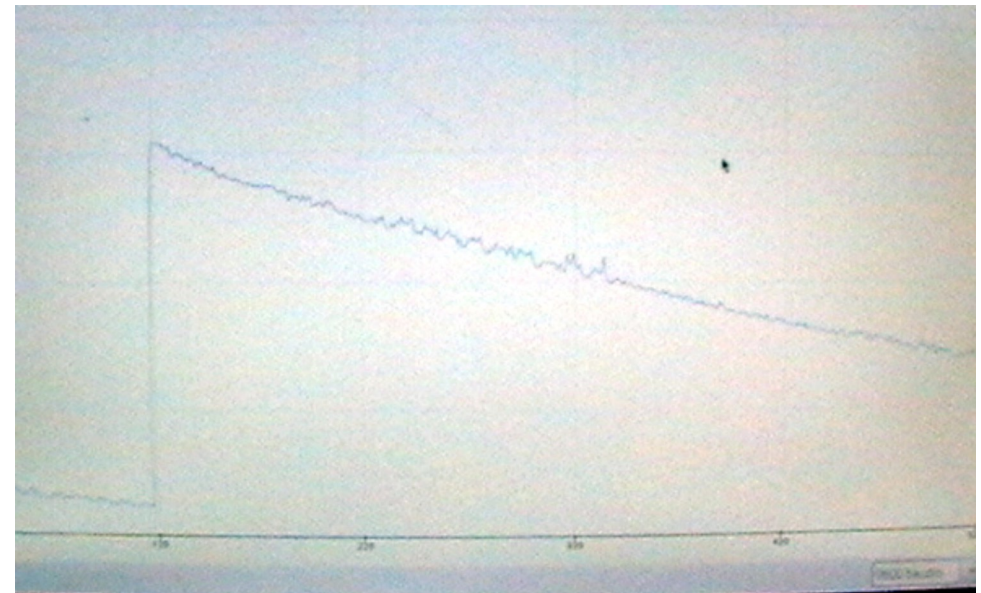
fritzing



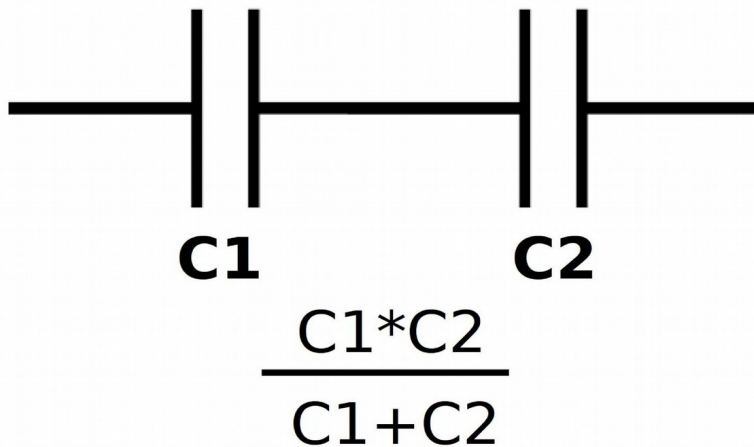
# ELETRONICA

167

```
1 void setup() {  
2   Serial.begin(9600);  
3  
4 }  
5  
6 void loop() {  
7   Serial.println(analogRead(0));  
8   delay(30);  
9  
10 }
```

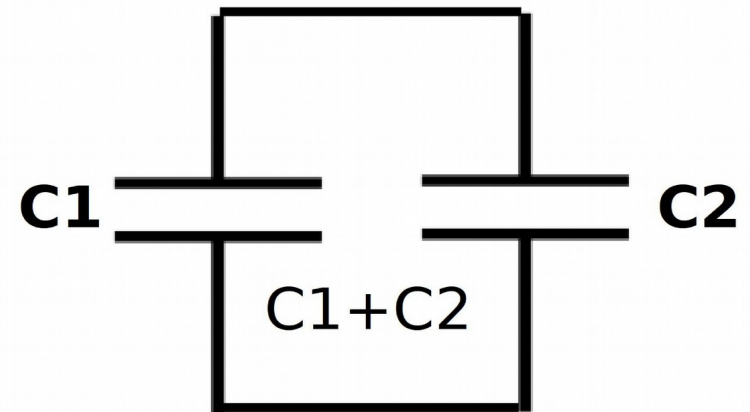


## IN SERIE



Mettendo in serie due condensatori otteniamo un valore di capacità minore del condensatore più piccolo.

## IN PARALLELO



Mettendo in parallelo due condensatori la capacità si somma ma la tensione di lavoro si dimezza.

# ARDUINO

## Il servomotore

168

