

Corso Arduino Base

Servomotori

I servomotori, come qualunque altro dispositivo di uscita, possono essere comandati attraverso la scheda Arduino.



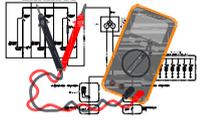
Servomotori utilizzati nella robotica e nel modellismo

I più comuni sono motori con angolo di rotazione 0° - 180° e vengono utilizzati quando serve pilotare un motore in posizione (cioè in angolo di rotazione) senza aggiungere ulteriori sensori.

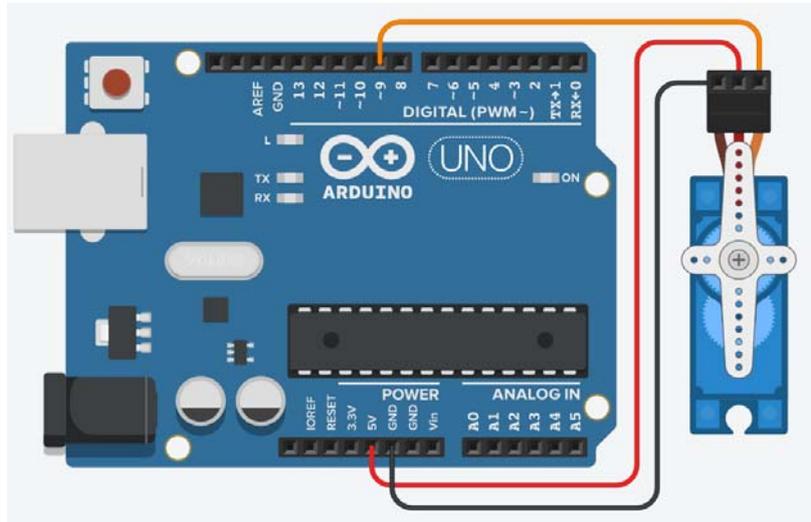
Il controllo della posizione può essere utilizzato in tantissime applicazioni (selettori che spostano oggetti, indici che puntano verso punti specifici, ecc...)

In caso di piccoli assorbimenti di potenza, oltre al comando, la scheda Arduino può fornire anche la potenza necessaria al suo funzionamento, pertanto sarà sufficiente verificare solamente che i livelli di tensione forniti da Arduino (0V - 5V) siano compatibili con la tensione nominale di alimentazione del servomotore (solitamente da 5V a 6V).

Il servomotore ha tre fili: uno di colore rosso da collegare alla sorgente di alimentazione (5V-6V), uno di colore marrone da collegare alla massa (GND) e quello giallo (arancione o bianco), di controllo (su cui transitano i segnali per il

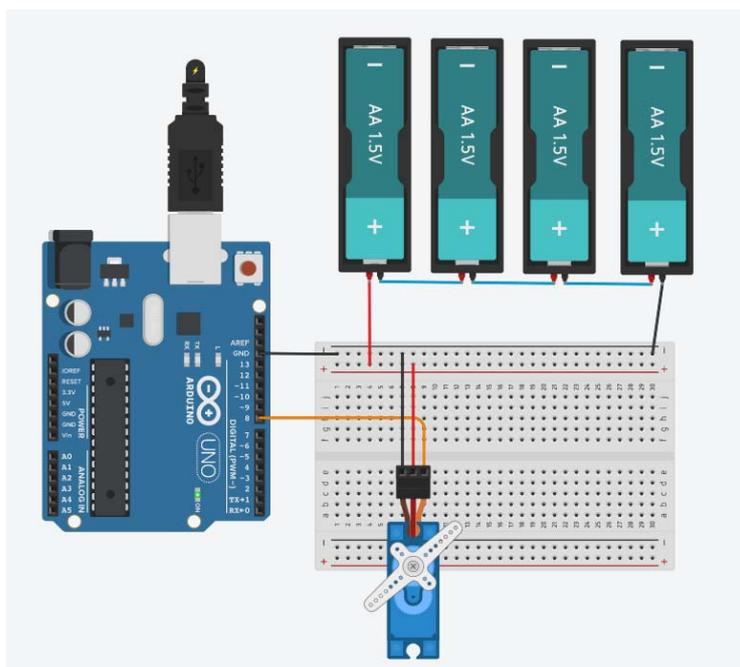


pilotaggio), da collegare ad una qualunque porta digitale di Arduino (non necessariamente PWM).

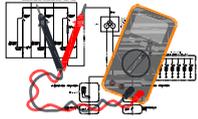


Servo controllato e alimentato da Arduino

In caso di assorbimenti di potenza non trascurabili, invece, la scheda Arduino si limiterà ad inviare il segnale di comando al servomotore che riceverà la potenza necessaria da alimentazione esterna (batterie, alimentatore, ecc...) ricordandosi di collegare la massa di Arduino in comune con il negativo dell'alimentazione.



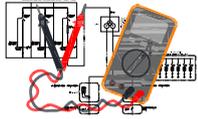
Servo controllato da Arduino e alimentato da batterie esterne



Il controllo del servomotore viene effettuato tramite l'invio di impulsi di larghezza temporale variabile: a seconda della durata dell'impulso il servomotore potrà girare in senso orario, antiorario oppure fermarsi. Tipicamente viene inviato un impulso ogni 20ms (cioè 50 impulsi al secondo): se la durata dell'impulso è pari a 1ms l'albero si muove in senso antiorario, mentre se la durata dell'impulso è pari a 2ms si muove in senso opposto; impostando la durata dell'impulso a 1,5ms il motore si ferma.

Utilizzando le funzioni della libreria "Servo" è possibile pilotare facilmente fino a 12 servomotori, senza curarsi dell'invio degli impulsi di comando, ma l'uso di questa libreria disabilita la funzionalità `analogWrite()` sui pin 9 e 10 (che pertanto perdono temporaneamente la possibilità di comandare un dispositivo di uscita analogico potendo fornire solo un segnale digitale On/Off), indipendentemente dal fatto che sia presente o meno un servo su tali pin.

Di seguito un codice di esempio che utilizza le funzioni `attach()` e `write()` della libreria `Servo.h`, per pilotare un servomotore da 0° a 180° con Arduino.



CODICE piattaforma Arduino

```
/* Programma per testare un servomotore attraverso le funzioni contenute nella  
libreria "Servo.h" */
```

```
// includiamo nel nostro programma la libreria "Servo.h"
```

```
// per poter utilizzare le sue funzioni
```

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

```
// creiamo un oggetto di tipo "Servo" per controllare un servomotore
```

```
// con Arduino Uno si possono creare fino a 12 oggetti "Servo"
```

```
Servo servo1;
```

```
int pos1 = 0; // variabile che memorizzerà la posizione di servo1
```

```
void setup(){
```

```
servo1.attach(8); // il servo1 sarà comandato attraverso il pin n.8
```

```
servo1.write(0); // porto il servo alla posizione zero gradi
```

```
// con apposita funzione contenuta nella libreria Servo.h
```

```
delay(2000); // attendo 2 secondi
```

```
servo1.write(90); // porto il servo alla posizione 90 gradi
```

```
delay(2000); // attendo 2 secondi
```

```
servo1.write(180); // porto il servo alla posizione 180 gradi
```

```
delay(2000); // attendo 2 secondi
```

```
}
```

```
void loop(){
```

```
for(pos1 = 0; pos1 <= 180; pos1+=20) { // la var pos1 va da 0 a 180 gradi
```

```
// in step di 20 gradi
```

```
servo1.write(pos1); // porto servo1 alla posizione 'pos1'
```

```
delay(500); // aspetto 500ms affinché servo1 raggiunga la posizione
```

```
} // chiudo il ciclo for
```

```
for(pos1 = 180; pos1>=0; pos1-=20) { // la var pos1 va da 180 a 0
```

```
// in step di 20 gradi
```

```
servo1.write(pos1); // porto servo1 alla posizione 'pos1'
```

```
delay(500); // aspetto 500ms affinché servo1 raggiunga la posizione
```

```
} // chiudo il ciclo for
```

```
}
```