Servomotori come e quando usarli

<https://www.gandotech.net/servomotori-come-quando-usarli/>

[30 settembre 2014](https://www.gandotech.net/servomotori-come-quando-usarli/) • [6 Comments](https://www.gandotech.net/servomotori-come-quando-usarli/#comments)

Una guida completa sui servomotori, per capirne il funzionamento e imparare a scegliere come e quando impiegarli. Include esempi per Arduino e microcontrollori PIC.

Ero indeciso se creare una nuova categoria “Robotica” o se pubblicare questa guida nel modo tradizionale. Alla fine ho deciso di pubblicarla nelle sezioni guide e elettronica, perché altrimenti finisco col creare troppe categorie distinte con relativamente poco materiale.

Comunque, questa settimana si parla di attuatori, e nello specifico i **servomotori**.  
Questi motori sono anche conosciuti come ***servocomandi***, ***motori servo,*** o più concisamente ***servo***.  
In questo articolo voglio illustrare questi dispositivi in modo semplice e completo, dalla descrizione del funzionamento, alle applicazioni fino ad arrivare ad indicazioni precise ed esempi di pilotaggio con Arduino e microcontrollori PIC.

Che le danze abbiamo inizio.

Cosa sono i servomotori ?

[](https://www.gandotech.net/wp-content/uploads/2014/09/Servo1.jpg)I motori servo, sono attuatori speciali muniti di un sistema di feedback che permette di controllarne la posizione angolare.

In altre parole, sono un tipo particolare di motori, che generalmente non ruotano in modo continuo (sebbene esistano alcuni modelli capaci di ruotare di 360°), e che è possibile controllare in modo da ruotarli in una posizione specifica e mantenerla fino a che lo si desidera.

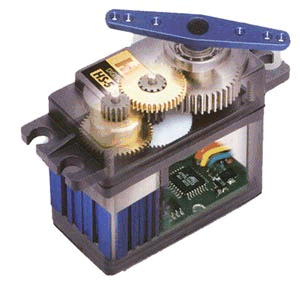
Contengono tutta l’elettronica che serve per comandarli: sistema di feedback, logica e stadio di potenza.

Possono ruotare in entrambi i sensi, e l’elettronica  che li comanda è in grado di variarne la velocità in modo efficiente per garantire una buona precisione nel posizionamento.

La maggior parte dei servomotori può ruotare di 180°, ma esistono modelli da 45°, 90° ed anche 360°.

Ne esistono di tutte le taglie, dai micro servocomandi per modellismo a servo industriali capaci di spostare un cammion.

Come funzionano i servomotori ?

[](https://www.gandotech.net/wp-content/uploads/2014/09/Struttura_Servo.gif)All’interno dei servomotori si trovano diverse parti meccaniche ed elettroniche. Eccone un elenco :

* Motore DC
* Un potenziometro
* Elettronica di controllo (sistema retroazionato)
* Motor driver (stadio di potenza)
* Un set di ingranaggi e un meccanismo di demoltiplica

Il potenziometro è collegato all’albero di uscita così da ottenere un valore resistivo proporzionale all’angolo della posizione attuale.

L’elettronica di controllo utilizza il valore resistivo per decidere la direzione di rotazione e fermarla quando raggiunge la posizione desiderata. Sempre questo sistema è responsabile della scelta della velocità, che è proporzionale alla distanza da percorrere.

Lo stadio di potenza è una parte di elettronica che si occupa di alimentare il motore in modo corretto a seconda della direzione, ed è in grado di fornire e dissipare tutta la potenza necessaria all’operazione.

Il set di ingranaggi e il meccanismo di demoltiplica servono ad aumentare la precisione, ridurre l’inerzia ed aumentare la coppia. Grazie a questa meccanica integrata i servomotori sono in grado di esercitare forze considerevoli utilizzando un motore poco potente e mantenendo i consumi bassi.

L’elettronica di controllo viene pilotata da un apposito segnale che descriverò nel prossimo paragrafo.  
Questo segnale può essere generato da un multivibratore astabile oppure, come ci piace tanto, da un microcontrollore. Sì perchè il bello dei servocomandi è quello di essere impiegati con i microcontrollori per creare automatismi, robot, o giocattoli sfiziosi ed interessanti.

Il segnale di controllo

Il segnale di controllo è tutto quello che abbiamo a disposizione per comandare il servocomando ed ottenere il comportamento che vogliamo.  
Si tratta di un segnale PWM a 50Hz con impulsi lunghi da un minimo di 1mS ad un massimo di 2mS dove 1mS corrisponde a  0° e 2mS a 180°.

**ATTENZIONE** : *1 e 2 millisecondi sono valori standard, ma poi ogni apparecchio ha la sua tolleranza, alcuni potrebbero differire abbastanza da questi valori, anche se dichiarano loro stessi di gradire questo segnale di controllo. Consiglio quindi di testare i servo ed aggiustare questi valori per il proprio apparecchio in modo da ottenere la massima precisione.*

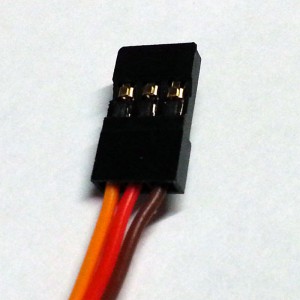
Ciò significa che per ottenere un segnale a 50Hz avremo un treno di impulsi lunghi da 1 a 2mS a 20mS di distanza gli uni dagli altri. Questo segnale può essere ottenuto in vari modi, usando i delay (busy waiting) oppure sfruttando interrupt del timer o appositi circuiti PWM.

Quando la scheda di controllo del servo riceve un treno di impulsi di 1mS (o il valore che hai testato sperimentalmente per il tuo dispositivo) fa ruotare il servo a 0°.  
Per ottenere il posizionamento a 90° basta fornire un treno di impulsi da 1.5mS, mentre per ruotarlo a 180° si fornisce in ingresso un treno di impulsi da 2mS.

Il segnale di controllo deve essere mantenuto almeno fino a quando il motore è completamente in posizione. Non fa male lasciare il segnale sempre applicato, infatti l’elettronica non muoverà il motore se riceve l’ordine di muoversi alla posizione attuale, anzi mantenere il segnale assicura che nessuna forza esterna alteri la posizione.

Al contrario, interrompere gli impulsi in modo prematuro comporterà un posizionamento non accurato, e la possibilità che l’albero possa ruotare per forze esterne.

Come cablare i servomotori

[](https://www.gandotech.net/wp-content/uploads/2014/09/Servo3.jpg)

Il connettore dei servomotori è composto da 3 fili, due sono per l’alimentazione ed uno è per il segnale di controllo.

Sembra scontato dirlo, ma l’alimentazione dei servo va rispettata. Non è possibile invertire massa e positivo perchè vengono usati per alimentare l’elettronica di controllo.

Inoltre, ogni dispositivo viene prodotto per lavorare in un determinato range di tensione, non bisogna eccedere nè sottoalimentare il servo.

Un’altra osservazione che sembra superflua ma è doveroso fare è che i motori consumano una corrente che non è bene ottenere ad esempio da Arduino alimentato dalla porta USB. Questo dipende molto dal servo utilizzato, però in generale, è meglio alimentare il tutto con un alimentatore esterno in grado di erogare le correnti necessarie.

I colori dei fili cambiano da modello a modello, consulta il datasheet del tuo attuatore oppure guarda se trovi la marca in questa tabella :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo servo** | **Segnale (S)** | **Alimentazione (+)** | **Alimentazione (-)** |
| **Futaba** | **Bianco** | **Rosso** | **Nero** |
| **JR** | **Arancio** | **Rosso** | **Marrone** |
| **Hitec** | **Giallo** | **Rosso** | **Nero** |
| **Airtronics** | **Arancio** | **Rosso** | **Nero** |
| **Airtronics** | **Bianco** | **Rosso** | **Nero** |
| **Airtronics** | **Nero** | **Rosso** | **Nero** |
| **Airtronics – Z** | **Blu** | **Rosso** | **Nero** |
| **KO** | **Bianco** | **Rosso** | **Nero** |
| **DAGU** | **Arancio** | **Rosso** | **Marrone** |

Possibili applicazioni dei servomotori

Questi comodi e sfiziosi attuatori possono essere impiegati in numerosi scenari. Da piccoli automatismi, a giocattoli, a veri e propri robot.  
Possono essere impiegati per operare a distanza in ambienti pericolosi, oppure per scopi ludici.

Sono molto spesso inclusi in modellini di vario tipo, come macchine (per lo sterzo), elicotteri, navi e barche (per il timone) , deltaplani, aerei a motore (flap) etc.  
Ovunque serva un attuatore in grado di produrre una rotazione controllata e mantenerla, i servo trovano applicazione.

Esistono poi servomotori potenti e costosi, per uso industriale, che possono muovere macchine imponenti.

Siccome il segnale di controllo è PWM si può provare a trasmettere il segnale di controllo usando uno dei moduli radio economici che ho presentato in questa serie di [articoli](https://www.gandotech.net/moduli-radio-economici/). Se il segnale viene trasmesso continuamente non ocorre un microcontrollore lato ricevitore, anche se così facendo si ottengono risultati molto approssimativi e appena il trasmettitore viene spento il servo impazzisce; l’utilizzo di un microcontrollore lato ricevitore è neessario se si vuole realizzare qualcosa di più di una semplice prova di concetto.

Pilotare i servomotori con Arduino

Per pilotare i servomotori con arduino esiste una libreria preconfezionata che semplifica molto il lavoro.

Sul sito di arduino si trova ovviamente la [documentazione](https://arduino.cc/en/reference/servo)della libreria, con l’elenco delle funzioni disponibili, i dettagli e gli esempi.

Per usare un servo con arduino basta quindi :

1. Importare la libreria
2. Istanziare un oggetto di tipo servo
3. Collegare l’oggetto ad un pin di arduino usando il metodo ***attach(****pin****)***
4. Usare i metodi write o writeMicroseconds per controllare il servo

Perchè due metodi distinti ?

Il metodo write accetta come parametro un angolo, mentre l’altro metodo accetta un valore in microsecondi che indica la durata dell’impulso.

Questa scelta permette di pilotare in modo semplice i servo standard (che sono quindi perfettamente conformi al segnale di controllo compreso tra 1mS e 2mS). Tuttavia, siccome esistono modelli che sono lievemente differenti, esiste la possibilità che in determinate applicazioni, l’utilizzo del metodo write risulti inefficace.  
In quei casi, è possibile usare il metodo writeMicroseconds per posizionare ‘manualmente’ il servo, provando i valori in microsecondi ed uscendo dall’intervallo standard.

**ATTENZIONE :** *Se quando fai ruotare il servo ad una posizione, questo non interrompe la rotazione del motore e emette rumore e vibrazioni vuol dire che è arrivato a fine corsa. Probabilmente devi ridurre l’angolo richiesto o provare sperimentalmente per trovare i valori in microsecondi che corrispondono a 0° e 180° perchè non sono quelli standard.****NON****devi mantenere il servo nell’impossibilità di ruotare, cunsuma un sacco di corrente e persistere può danneggiare il dispositivo in vari punti.*

Esempi di utilizzo della libreria sono inclusi nell’IDe di Arduino e anche nella [documentazione](https://arduino.cc/en/reference/servo).

Pilotare i servomotori con i PIC

Per i PIC non ho trovato una libreria abbastanza astratta da funzionare sicuramente su tutto l’HW a disposizione.  
Questo dipende dal fatto che mentre per il mondo di Arduino esistono un paio di microcontrollori, quando si progetta qualcosa con i PIC si ha a disposizione un arsenale di hardware, tutto diverso, con limiti e caratteristiche diverse.

Molto spesso, se si desidera realizzare un progetto complesso, i PIC danno più scelta per comprare solo HW che serve, senza sprecare un microcontrollore con 8 timer e 32Kb di memoria programma per far lampeggiare un led.

Il rovescio della medaglia è che ci si deve rimboccare le maniche e fare quelli che si scrivono le procedure, invece che usare la pappa preparata da altri.

Come ho già accennato è possibile ottenere il segnale di controllo usando diversi metodi, alcuni sono :

* Busy waiting
* Interrupt del timer
* Hardware per PWM incluso nel micro o esterno

Busy waiting :

Questo metodo è il più semplice, e funziona su qualsiasi PIC perchè usa eclusivamente un pin digitale e tempo macchina.

Gli svantaggi sono che generare il segnale è un’operazione critica, non si possono sprecare cicli cpu e quindi farlo impegna il microcontrollore che non può fare altro che questo per tutta la durata necessaria.

Si può usare a seconda delle applicazioni, se ad esempio gli interrupt sono sporadici e quando accadono provocano quasi certamente una variazione del segnale di controllo questa tecnica risulta efficace.

Se al contrario, mentre la cpu sta generando il segnale di controllo vengono eseguiti molti interrupt il segnle di controllo non sarà preciso, e il servo risponderà muovendosi attorno alla zona richiesta senza mai raggiungerla.

Ecco una possibile implementazione di questo metodo :

void sendDeg(unsigned grad) {

unsigned short i = 0;

unsigned upTime = 500 + ( (10 \* grad) );

unsigned downTime = 20000 - upTime;

GIE\_bit = 0;

for (i = 0; i > 15; i++) {

RA0\_bit = 1;

VDelay\_us ( upTime );

RA0\_bit = 0;

VDelay\_us ( downTime );

}

GIE\_bit = 1;

}

La procedura riportata sopra è realizzata per inviare treni da minimo 15 impulsi di durata aggiustata per il mio servomotore.  
Il segnale di controllo viene generato su RA0. Puoi modificare la matematica di uptime per ottenere l’impulso della durata corretta se questo non va bene per il tuo servomotore, il valore downtime viene calcolato automaticamente in funzione di uptime, e garantisce un segnale a 50Hz.

Interrupt del Timer

Quì dipende da quanti e quali timer si hanno a disposizione, la cosa da tenere in mente è che il segnale deve essere a 50Hz, poi all’interno della funzione di interrupt è possibile usare vari contatori per ottenere il timing corretto.

Non riporto una soluzione per questo caso, perchè è abbastanza inutile. A seconda del microcontrollore usato si hanno timer con capacità diversa, e diversa capacità o possibilità di usare contatori a più byte.

Una soluzione che risulta facile da realizzare, e che comporta poco lavoro a livello di configurazione del timer è quella di predisporlo per avere un interrupt ogni microsecondo o ogni 10 microsecondi, e poi usare un contatore per calcolare la durata dell’impulso.

Le soluzioni basate sul timer sono migliori, sia come precisione, sia per il fatto che il microcontrollore è in grado di svolgere altre azioni e di essere interrotto solo quando necessario pur generando un segnale preciso.

Circuiti PWM

Esistono moduli interni ad alcuni PIC in grado di generare segnali PWM in modo autonomo, quando possibile, utilizzare questi moduli è la soluzione migliore sotto ogni punto di vista.

Non tutti i moduli PWM sono però in grado di generare segnali PWM con frequenze basse, come purtroppo occorre per i servomotori (parlo dei moduli interni ai PIC).

Occorre quindi verificare le capacità del proprio HW, nel caso si desideri utilizzare un PIC sprovvisto di tale modulo, esistono circuiti integrati che possono generare per noi il segnale. Questi sono l’ideale se si vuole utilizzare un microcontrollore modesto e si devono controllare molti servomotori.

Moduli PWM basati su circuiti integrati del genere sono reperibili su Adafruit o Sparkfun e in genere utilizzano una porta I2C del microcontrollore per comandare fino a 16 canali PWM a 12bit di risoluzione.

Link

Un breve video dimostrativo : <https://www.youtube.com/watch?v=ln5VuzEHR-g>

Servomotori su Wikipedia : <https://it.wikipedia.org/wiki/Servocomando>

Documentazione libreria Arduino :[https://arduino.cc/en/reference/servo](https://arduino.cc/en/reference/servo" \t "_blank)

Servomotori su Ebay : [Servomotori](https://rover.ebay.com/rover/1/724-53478-19255-0/1?icep_ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&icep_uq=servomotor&icep_sellerId=&icep_ex_kw=&icep_sortBy=12&icep_catId=&icep_minPrice=&icep_maxPrice=&ipn=psmain&icep_vectorid=229494&kwid=902099&mtid=824&kw=lg)https://rover.ebay.com/roverimp/1/724-53478-19255-0/1?ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&uq=servomotor&mpt=%5bCACHEBUSTER%5d

Arduino su Ebay : [Arduino nano](https://rover.ebay.com/rover/1/724-53478-19255-0/1?icep_ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&icep_uq=arduino+nano&icep_sellerId=&icep_ex_kw=&icep_sortBy=12&icep_catId=&icep_minPrice=&icep_maxPrice=&ipn=psmain&icep_vectorid=229494&kwid=902099&mtid=824&kw=lg)https://rover.ebay.com/roverimp/1/724-53478-19255-0/1?ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&uq=arduino+nano&mpt=%5bCACHEBUSTER%5d

PIC su Ebay : [PIC microcontroller](https://rover.ebay.com/rover/1/724-53478-19255-0/1?icep_ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&icep_uq=pic+microcontroller&icep_sellerId=&icep_ex_kw=&icep_sortBy=12&icep_catId=&icep_minPrice=&icep_maxPrice=&ipn=psmain&icep_vectorid=229494&kwid=902099&mtid=824&kw=lg)https://rover.ebay.com/roverimp/1/724-53478-19255-0/1?ff3=9&pub=5575090064&toolid=10001&campid=5337506914&customid=&uq=pic+microcontroller&mpt=%5bCACHEBUSTER%5d