

Meccanica per Robot ARDUFILIPPO

Prezzo: 55.74 €

Tasse: 12.26 €

Prezzo totale (con tasse): 68.00 €



È un robot bipede i cui movimenti sono affidati a due soli servo, permettendo di cimentarsi con la robotica ed imparare l'uso dei servocomandi senza spendere molto. È in grado di camminare e di girare su se stesso, quindi potete dirigerlo in ogni direzione. Il suo assemblaggio è facilitato in quanto tutti i pezzi si incastrano gli uni negli altri e si possono saldare o incollare (con colla epossidica) per stabilizzare gli incastri. Il robot può essere controllato (nelle funzioni essenziali) a distanza tramite un telecomando IR (un normalissimo telecomando di quelli usati per i televisori o video-registratori). Il kit comprende tutti i particolari meccanici e i 2 servo. Non comprende la scheda di controllo Robot_Shield, la scheda ArduinoUnoREV3 e le batterie.

Cosa serve per completare il Robot

QUANTITÀ	PRODOTTO
1	8190-ROBOT_SHIELD "SHIELD DI CONTROLLO PER ARDUSPIDERIN, ARDUBIPE E ARDUFILIPPO"
1	7300-ARDUINOUNOREV3 "ARDUINO UNO REV3 CON ATMEGA328"
6-8	BATTERIE RICARICABILI NIMH O NICD FORMATO STILO

L'alimentazione e lo sketch

Per l'alimentazione è previsto vengano usate 6 o 8 batterie ricaricabili NiMh o NiCd formato stilo, che vanno inserite in due distinti portabatterie e collegate in serie.

Se optate per l'utilizzo di 6 celle dovrete usare due portabatterie a tre vani, mentre se usate otto celle dovrete usare due portabatterie da 4 vani. I portabatterie vanno posizionati uno a destra ed uno a sinistra dei due servo, collegandoli in serie con due clip per batterie da 9 volt (Fig. a sinistra).

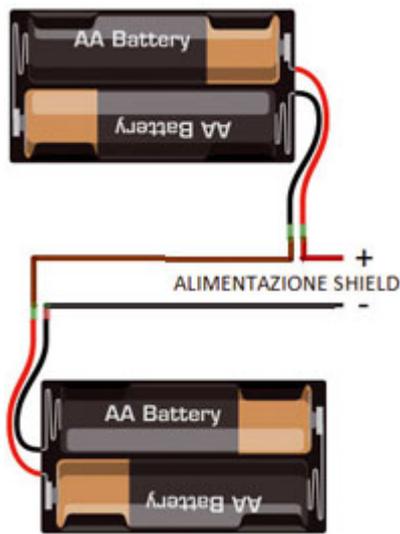
Consigliamo di programmare Arduino prima di collegare i servo, onde evitare che un eventuale programma precedentemente in memoria fornisca segnali che possono portare a movimenti scomposti.

Ricordatevi che se alimentate Arduino solo tramite la USB, per programmarlo, parte della tensione finirà sullo shield, quindi consigliamo di collegare il cavo USB solo quando è presente anche l'alimentazione principale sui punti BAT; diversamente, potrebbero esserci problemi con i servo.

Programmate Arduino con lo sketch *filippo.pde* ed avviate Serial Monitor, allorchè vi ritroverete una schermata con una serie di dati; quelli che vi interessano sono la tensione della batteria e la posizione di neutro dei servo.

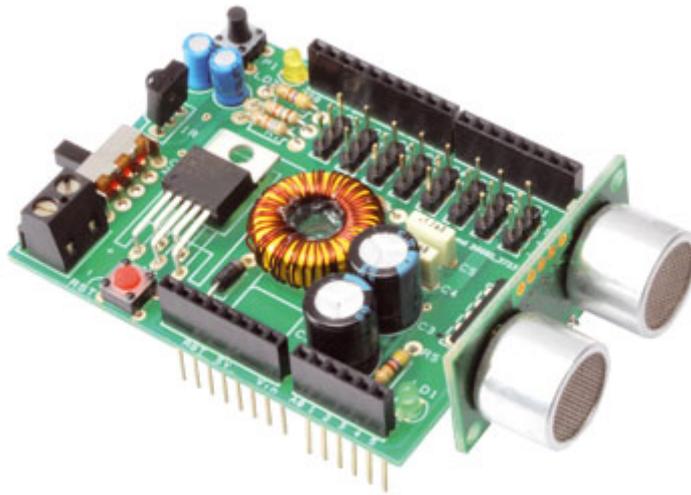
Inviare il comando "obs" per avere la lettura del sensore SRF05 ed il comando "lev" per verificare il livello della batteria. Assicuratevi che la tensione letta corrisponda al reale valore delle batterie; per accertare ciò è sufficiente confrontare i dati con una misura effettuata con un buon tester.

Concentratevi ora sull'impostazione del neutro dei servo, indispensabile in quanto il robot, per funzionare correttamente, deve essere perfettamente centrato, cosa non sempre possibile a livello meccanico, in quanto i braccetti dei servo possono avere solo posizioni predefinite.

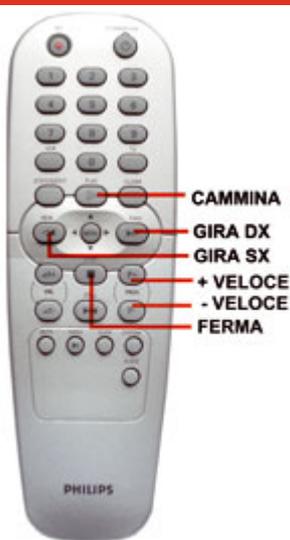


[Guarda il video](#)

[La scheda di controllo "ROBOT_SHIELD"](#)



Il controllo con un telecomando IR



Lo sketch è previsto per funzionare con telecomandi che usano la codifica Philips, che è la più utilizzata nei televisori; se usate un trasmettitore di marca differente i codici dei pulsanti non corrispondono e dovrete modificare lo sketch, inserendo i codici che Serial Monitor mostra sullo schermo del PC quando trasmettete col vostro telecomando (Fig.8 e Fig.9).

Figura 8

Figura 9

```

COM6
FILIPPO Arduino V1.0
Vbat= 7.8900001144
Livello batteria OK.
Recupero dati posizione neutro servi...
Neutro dei Servi: 87 90
Inizializzo Servi...
Servi inizializzati.
Invia 'obs' per testare il sensore di ostacoli
Invia 'lev' per testare il livello della batteria
Sono in Standby...
IR code= 1
TimeOneStep= 2000
IR code= 801
TimeOneStep= 2000
Autoscroll No line ending 9600 baud
  
```

```

COM6
FILIPPO Arduino V1.0
Vbat= 7.8600001335
Livello batteria OK.
Recupero dati posizione neutro servi...
Neutro dei Servi: 87 90
Inizializzo Servi...
Servi inizializzati.
Invia 'obs' per testare il sensore di ostacoli
Invia 'lev' per testare il livello della batteria
Sono in Standby...
IR code= 20DF02FD
TimeOneStep= 2000
Autoscroll No line ending 9600 baud
  
```

Ad esempio, se utilizzate un telecomando LG noterete che il codice inviato è unico indipendentemente da come premete i pulsanti. Volendo, ad esempio, assegnare il pulsante “freccia in su” della navigazione a menù alla funzione di camminata, non dovrete fare altro che premerlo, leggere il codice IR ricevuto e trascriverlo nella riga dello sketch che assegna il codice per la camminata:

```
#define WALK_CODE1 0x20DF02FD  
#define WALK_CODE2 0x20DF02FD
```

I parametri che possono essere modificati da programma per adattare i movimenti sono:

```
int TimeOneStep = 2000;  
int AmpPasso = 30;  
int IncPasso = 15;  
int AmpRuota = 30;  
int IncRuota = 18;
```

TimeOneStep è il valore iniziale del tempo impiegato per eseguire un passo, può essere modificato da telecomando da un minimo di 1 ad un massimo di 4 secondi.

AmpPasso è la massima ampiezza (in gradi) dell’angolo formato dalle gambe durante la camminata; il suo valore può variare da un minimo di 5 ad un massimo di 40.

IncPasso è la massima inclinazione del robot durante la camminata; il suo valore può variare da un minimo di 5 ad un massimo di 20 e deve garantire al robot di rimanere in equilibrio su una gamba mentre compie un passo.

AmpRuota è la massima ampiezza in gradi dell’angolo formato dalle gambe durante la rotazione; il suo valore può variare da un minimo di 5 ad un massimo di 40.

IncRuota è la massima inclinazione del robot durante la rotazione; il suo valore può variare da un minimo di 5 ad un massimo di 20.

Documentazione e link utili

- [SKETCH](#)